

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 996 269**

51 Int. Cl.:

H04N 19/146 (2014.01)

H04N 19/17 (2014.01)

H04N 19/167 (2014.01)

H04N 19/42 (2014.01)

H04N 19/98 (2014.01)

G06T 1/20 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **01.03.2018** **E 18159605 (7)**

97 Fecha y número de publicación de la concesión europea: **11.09.2024** **EP 3389273**

54 Título: **Procesamiento basado en regiones**

30 Prioridad:

10.04.2017 US 201715483854

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

12.02.2025

73 Titular/es:

**INTEL CORPORATION (100.00%)
2200 Mission College Blvd.
Santa Clara, CA 95054, US**

72 Inventor/es:

**KAMBHATLA, SRIKANTH;
PARIKH, KUNJAL;
WANG, CHANGLIANG y
SMITH, GARY**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 996 269 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

Procesamiento basado en regiones

5 **CAMPO TÉCNICO**

Las realizaciones generalmente se refieren a arquitecturas de procesamiento de gráficos. Más en particular, las realizaciones se refieren al procesamiento basado en regiones en arquitecturas de procesamiento gráfico.

10 **ANTECEDENTES DE LA DESCRIPCIÓN**

En los entornos de visualización inalámbrica, una plataforma de origen (por ejemplo, consola de juegos, ordenador) puede codificar (por ejemplo, comprimir) una señal de vídeo tal como una película o un juego tridimensional (3D) para su transmisión inalámbrica a una pantalla. Cada fotograma de la señal de vídeo puede codificarse de manera uniforme y colocarse en una memoria intermedia de fotogramas, que se sincroniza con la salida de la pantalla. La codificación uniforme de los fotogramas puede aumentar la utilización del ancho de banda del enlace inalámbrico, el consumo de potencia y/o el coste, en particular si la señal de vídeo contiene contenido de alto rango dinámico (HDR).

El documento WO2013144809 A2 divulga un codificador de imágenes para codificar una imagen de una escena de alto rango dinámico, que comprende: una unidad de codificación de textura de píxeles, dispuesta para codificar colores de píxeles de la imagen con una representación de imagen que comprende palabras de código de N bits; una unidad de análisis de imagen dispuesta para determinar y emitir a un valor de gris diferenciador de región, que es un valor luma que demarca por debajo del mismo luma de todos los píxeles de un primer objeto en al menos un bloque de la imagen, y por encima del mismo luma de todos los píxeles de un segundo objeto en el al menos un bloque de la imagen; y un formateador dispuesto para codificar conjuntamente en una señal de imagen de salida la representación de imagen y el valor de gris diferenciador de región, que hace posible una utilización muy versátil de las imágenes que codifican escenas HDR, en particular codificándolas en formatos de imagen compatibles con las normas de codificación de vídeo MPEG existentes.

El artículo "Region-based tone mapping for efficient High Dynamic Range video coding" de P. Lauga y col., European workshop on visual information processing, del 10 de junio de 2013 divulga un método de codificación de vídeo de alto rango dinámico (HDR) que segmenta la imagen HDR en regiones de bajo y alto rango que se correlacionan por tonos de manera independiente.

La invención se define en las reivindicaciones independientes. Las realizaciones de la invención se definen en las reivindicaciones dependientes.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Las diversas ventajas de las realizaciones se volverán 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:

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, según una realización;

las Figuras 3A-3B son diagramas de bloques de multiprocesadores de gráficos, según las realizaciones;

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

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

la Figura 6A es una ilustración de un ejemplo de imagen de alto rango dinámico (HDR) que está dividida en una pluralidad de regiones según una realización;

la Figura 6B es una ilustración de un ejemplo de una imagen HDR que está dividida en una región en primer plano y una región de fondo según una realización;

la Figura 7 es un diagrama de bloques de un ejemplo de una arquitectura de procesamiento basada en regiones según una realización;

la Figura 8A es un diagrama de flujo de un ejemplo de un método de funcionamiento de un aparato de paquete semiconductores según una realización;

la Figura 8B es un diagrama de flujo de un ejemplo de un método de selección de cantidades de codificación basándose en el nivel de luminancia según una realización;

5 la Figura 8C es un diagrama de flujo de un ejemplo de un método de selección de cantidades de codificación para regiones en primer plano y de fondo según una realización;

la Figura 9 es un diagrama de bloques de un ejemplo de un sistema informático según una realización;

10 la Figura 10 es una ilustración de un ejemplo de un aparato de paquete de semiconductores según una realización;

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

15 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;

20 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

25 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;

30 las Figuras 16-18 son diagramas de bloques de un ejemplo de una vista general de un sistema de procesamiento de datos según una realización;

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

35 las Figuras 20-22 son diagramas de bloques de ejemplos de unidades de ejecución según una realización;

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

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

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

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

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

50

DESCRIPCIÓN DETALLADA

En la siguiente descripción, se exponen numerosos detalles específicos para facilitar una comprensión más completa de la presente invención. Sin embargo, será evidente para un experto en la materia que la presente invención se puede poner en práctica sin uno o más de estos detalles específicos. En otros casos, no se han descrito características bien conocidas para evitar complicar la presente invención.

55

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 comunican 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 un conjunto de chips o puede estar integrado dentro de uno o más 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

60
65

5 puede permitir que el sistema informático 100 reciba la entrada de uno o más dispositivos de entrada 108. Adicionalmente, el concentrador de E/S 107 puede permitir a un controlador de visualización, que puede estar incluido en el uno o varios procesadores 102, proporcionar salidas a uno o varios 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.

10 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, aunque no de forma limitativa, PCI Express, o puede ser una interfaz de comunicaciones o estructura de comunicaciones específica del proveedor. En una realización, el uno o más procesadores paralelos 112 forman un sistema de procesamiento paralelo o vectorial enfocado informáticamente 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 a través del concentrador de E/S 107. El uno o más procesadores paralelos 112 pueden incluir también un controlador de visualización y una interfaz de visualización (no mostrados) para permitir una conexión directa a uno o más dispositivos de visualización 110B.

20 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. Se puede utilizar 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 inalámbrica 119 que pueden estar integrados en la plataforma, y varios otros dispositivos que pueden añadirse mediante uno o más dispositivos suplementarios 120. El adaptador de red 118 puede ser un adaptador Ethernet u otro adaptador de red cableado. El adaptador de red inalámbrica 119 puede incluir uno o más de un dispositivo de red Wi-Fi, Bluetooth, de comunicación de campo cercano (NFC) o de otro tipo que incluye una o más radios inalámbricas.

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

35 En una realización, el uno o más procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de gráficos y vídeo, incluyendo, por ejemplo, circuitería de salida de vídeo, y constituye una unidad de procesamiento de gráficos (GPU). En otra realización, el uno o más procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de propósito general, mientras conservan la arquitectura de cálculo 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 de otros elementos de sistema en un único circuito integrado. Por ejemplo, el uno o varios procesadores en paralelo 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 integrarse 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 se puede integrar en un módulo de múltiples chips (MCM), que se puede interconectar con otros módulos de múltiples chips en un sistema informático modular.

50 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, incluyendo el número y disposición de puentes, el número de procesador(es) 102, y el número de procesador(es) paralelo(s) 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(es) 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 integrarse en un único chip. Algunas realizaciones pueden incluir dos o más conjuntos de procesador(es) 102 conectados mediante múltiples tomas, que pueden acoplarse con dos o más instancias del procesador o procesadores paralelos 112.

60 Algunos de los componentes particulares mostrados en el presente documento son opcionales y pueden no incluirse en todas las implementaciones del sistema informático 100. Por ejemplo, se puede admitir cualquier número de tarjetas o periféricos suplementarios, o pueden eliminarse algunos componentes. Asimismo, algunas arquitecturas pueden utilizar una terminología diferente para componentes similares a los ilustrados en la Figura 1. Por ejemplo, el concentrador de memoria 105 puede denominarse puente norte en algunas arquitecturas, mientras que el concentrador de E/S 107 puede denominarse puente sur.

La Figura 2A ilustra un procesador paralelo 200, según una realización. Los diversos componentes del procesador paralelo 200 pueden implementarse utilizando uno o más dispositivos de circuito integrado, tales como procesadores programables, circuitos integrados específicos de la aplicación (ASIC) o matrices de puertas programables en campo (FPGA). El procesador paralelo 200 ilustrado es una variante del uno o más procesadores paralelos 112 mostrados en la Figura 1, según 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 permite la comunicación con otros dispositivos, incluyendo otras instancias de la unidad de procesamiento paralelo 202. La unidad de E/S 204 puede conectarse directamente a otros dispositivos. En una realización, la unidad de E/S 204 se conecta con otros dispositivos utilizando 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 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 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 comandos mediante la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para ejecutar esos comandos a un extremo frontal 208. En una realización, el extremo frontal 208 se acopla con un planificador 210, que está configurado para distribuir comandos u otros elementos de trabajo a una matriz de agrupaciones de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupaciones 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 agrupaciones de procesamiento 212. En una realización, el planificador 210 se implementa mediante una lógica de firmware que se ejecuta en un microcontrolador. El planificador implementado por microcontrolador 210 es configurable para realizar operaciones complejas de planificación y distribución de trabajo con granularidad gruesa y fina, permitiendo una rápida preferencia y cambio 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 la planificación en la matriz de procesamiento 212 mediante uno de múltiples timbres de procesamiento de gráficos. Las cargas de trabajo pueden entonces distribuirse automáticamente a lo largo de la matriz de procesamiento 212 mediante la lógica del planificador 210 dentro del microcontrolador planificador.

La matriz de agrupaciones de procesamiento 212 puede incluir hasta "N" agrupaciones de procesamiento (por ejemplo, de la agrupación 214A, agrupación 214B a la agrupación 214N). Cada agrupación 214A a 214N de la matriz de agrupaciones 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 agrupaciones de procesamiento 212 utilizando varios algoritmos de planificación y/o distribución de trabajo, que pueden variar dependiendo de la carga de trabajo que surja para cada tipo de programa o cálculo. La planificación puede ser manejada dinámicamente por el planificador 210, o puede ser asistida en parte por la lógica del compilador durante la compilación de la lógica del programa configurada para su ejecución por la matriz de agrupaciones de procesamiento 212. En una realización, se pueden asignar diferentes agrupaciones 214A a 214N de la matriz de agrupaciones de procesamiento 212 para procesar diferentes tipos de programas o para realizar diferentes tipos de cálculos.

La matriz de grupo de procesamiento 212 se puede configurar para realizar varios tipos de operaciones de procesamiento en paralelo. En una realización, la matriz de grupo de procesamiento 212 está configurada para realizar operaciones de cálculo paralelas de propósito general. Por ejemplo, la matriz de agrupaciones de procesamiento 212 puede incluir lógica para ejecutar tareas de procesamiento que incluyen filtración de datos de video y/o de audio, realización de operaciones de modelado, que incluyen operaciones físicas y realización de transformaciones de datos.

En una realización, la matriz de agrupaciones de procesamiento 212 está configurada para realizar operaciones de procesamiento de gráficos paralelo. En las realizaciones en las que el procesador en paralelo 200 está configurado para realizar operaciones de procesamiento de gráficos, la matriz de agrupaciones de procesamiento 212 puede incluir lógica adicional para admitir la ejecución de dichas operaciones de procesamiento de gráficos, incluyendo, aunque no de forma limitativa, una lógica de muestreo de textura para realizar operaciones de textura, así como lógica de teselado y otra lógica de procesamiento de vértices. Adicionalmente, la matriz de agrupaciones de procesamiento 212 puede configurarse para ejecutar programas de sombreado relacionados con el procesamiento de gráficos tales como, aunque no de forma limitativa, 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 del sistema mediante la unidad de E/S 204 para su procesamiento. Durante el procesamiento, los datos transferidos se pueden almacenar en la memoria en chip (por ejemplo, la memoria del procesador paralelo 222) durante el procesamiento, y luego escribirse nuevamente en la memoria del sistema.

En una realización, cuando la unidad de procesamiento paralelo 202 se utiliza para realizar el procesamiento de gráficos, el planificador 210 puede configurarse 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 agrupaciones de procesamiento 212. En algunas realizaciones, porciones

de la matriz de agrupaciones de procesamiento 212 pueden configurarse para realizar diferentes tipos de procesamiento. Por ejemplo, una primera porción puede configurarse para realizar un sombreado de vértices y una generación de topología, una segunda porción puede configurarse para realizar sombreado de teselación y de geometría, y una tercera porción puede configurarse 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 agrupaciones de procesamiento 212 puede recibir tareas de procesamiento a ejecutar a través del planificador 210, que recibe comandos que definen tareas de procesamiento desde el extremo frontal 208. Para las operaciones de procesamiento de gráficos, las tareas de procesamiento pueden incluir índices de datos que se procesarán, por ejemplo, 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 se procesarán los datos (por ejemplo, qué programa se ejecutará). El planificador 210 puede estar configurado para extraer los índices que corresponden a las tareas o puede recibir los índices desde el extremo frontal 208. El extremo frontal 208 se puede configurar para garantizar que la matriz de agrupaciones de procesamiento 212 esté configurada en un estado válido antes de que se inicie la carga de trabajo especificada por las memorias intermedias de comandos entrantes (por ejemplo, memorias intermedias de lotes, memorias intermedias de inserción, etc.).

Cada una de la una o más instancias de la unidad de procesamiento paralelo 202 se puede acoplar con la memoria de procesador paralelo 222. Se puede acceder a la memoria de procesador paralelo 222 mediante la barra transversal de memoria 216, que puede recibir solicitudes de memoria de la matriz de agrupaciones 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 (por ejemplo, unidad de subdivisión 220A, unidad de subdivisión 220B, hasta la unidad de subdivisión 220N) pudiendo cada una emparejarse a una parte (por ejemplo, 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 primera unidad de memoria 224A correspondiente, una segunda unidad de subdivisión 220B tiene una unidad de memoria 224B correspondiente y una enésima unidad de subdivisión 220N tiene una enésima unidad de memoria 224N correspondiente. 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 gráfica de acceso aleatorio, tal como una memoria gráfica de acceso aleatorio sincrónica (SGRAM), que incluye una memoria gráfica de doble tasa de datos (GDDR). En una realización, las unidades de memoria 224A-224N también pueden incluir memoria apilada 3D, incluyendo, aunque no de forma limitativa, 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 a partir de uno de diversos diseños convencionales. Los objetivos de representación, tales como las memorias intermedias de fotograma o los mapas de textura pueden almacenarse a lo largo 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 utilizar de manera eficiente 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 junto con memoria caché local.

En una realización, cualquiera de las agrupaciones 214A-214N de la matriz de agrupaciones 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 se puede configurar para transferir la salida de cada agrupación 214A-214N a cualquier unidad de subdivisión 220A-220N o a 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 varios 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 en paralelo 222, lo que permite que las unidades de procesamiento dentro de las diferentes agrupaciones de procesamiento 214A a 214N se comuniquen con la memoria de sistema u otra memoria que no sea local a la unidad de procesamiento en paralelo 202. En una realización, la barra transversal de memoria 216 puede utilizar canales virtuales para separar flujos de tráfico entre las agrupaciones 214A a 214N y las unidades de subdivisión 220A a 220N.

Aunque se ilustra una única instancia de la unidad de procesamiento paralelo 202 dentro del procesador en paralelo 200, se puede incluir 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 suplementaria, o pueden interconectarse múltiples tarjetas suplementarias. Las diferentes instancias de la unidad de procesamiento paralelo 202 pueden configurarse 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 varias instancias de la unidad de procesamiento en paralelo 202 o el procesador en paralelo 200 se pueden implementar en una diversidad de configuraciones y factores de forma, incluyendo, aunque no de forma limitativa, 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, según 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 fotogramas 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 de almacenamiento recibidas desde la barra transversal de memoria 216 y la ROP 226. Los errores de lectura y las peticiones urgentes de escritura son enviadas por la memoria caché L2, 221, a la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. También pueden enviarse actualizaciones a la memoria intermedia de fotogramas mediante la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. En una realización, la interfaz de memoria intermedia de fotogramas 225 interacciona con una de las unidades de memoria en memoria de procesador paralelo, tales 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. A continuación, la ROP 226 proporciona datos de gráficos procesados que se almacenan en la memoria gráfica. En algunas realizaciones, la ROP 226 incluye lógica de compresión para comprimir datos de profundidad o color que se escriben en la memoria y descomprimir datos de profundidad o color que se leen de la memoria. La lógica de compresión puede ser lógica de compresión sin pérdidas que utiliza 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 una compresión de color delta sobre datos de profundidad y de color por mosaico.

En algunas realizaciones, la ROP 226 está incluida dentro de cada agrupación de procesamiento (por ejemplo, la agrupación 214A-214N de la Figura 2) en lugar de dentro de la unidad de subdivisión 220. En tal realización, se transmiten solicitudes de lectura y de escritura de datos de píxel a través de la barra transversal de memoria 216 en lugar de datos de fragmento de píxel. Los datos de gráficos procesados se pueden visualizar en un dispositivo de visualización, tal como uno del uno o más dispositivos de visualización 110 de la Figura 1, enrutados para su posterior procesamiento por el procesador o procesadores 102, o enrutados para su posterior procesamiento por una de las entidades de procesamiento dentro del procesador en 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, según 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 configurarse 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 utilizan técnicas de emisión de instrucciones de instrucción única, múltiples datos (SIMD) para admitir la ejecución paralela de una gran cantidad de hilos sin proporcionar múltiples unidades de instrucción independientes. En otras realizaciones, se utilizan técnicas de instrucción única e hilos múltiples (SIMT) para admitir la ejecución en paralelo de un gran número de hilos generalmente sincronizados, utilizando una unidad de instrucción común configurada para emitir instrucciones a un conjunto de motores de procesamiento dentro de cada uno de los grupos 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 comprenderán que un régimen de procesamiento SIMD representa un subconjunto funcional de un régimen de procesamiento SIMT.

El funcionamiento de la agrupación de procesamiento 214 puede controlarse mediante un gestor de canalizaciones 232 que distribuye tareas de procesamiento a procesadores paralelos 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 textura 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. Una o más instancias del multiprocesador de gráficos 234 pueden incluirse dentro de una agrupación de procesamiento 214. El multiprocesador de gráficos 234 puede procesar datos y una barra transversal de datos 240 se puede utilizar para distribuir los datos procesados a uno de los múltiples destinos posibles, incluyendo otras unidades de sombreado. El gestor de canalizaciones 232 puede facilitar la distribución de los datos procesados mediante la especificación de destinos para los datos procesados que se distribuirán a través de 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 (por ejemplo, unidades aritmético-lógicas, unidades de carga-almacenamiento, etc.). La lógica de ejecución funcional se puede configurar de una manera canalizada en la que pueden emitirse nuevas

instrucciones antes de que se hayan completado instrucciones previas. La lógica de ejecución funcional admite una diversidad de operaciones que incluyen aritmética de números enteros y de coma flotante, operaciones de comparación, operaciones booleanas, desplazamiento de bits y de cálculo de diversas funciones algebraicas. En una realización, se puede aprovechar el mismo hardware de unidad funcional 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 en el 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 estar inactivo durante los ciclos en los que se está procesando ese grupo de hilos. Un grupo de hilos también puede incluir 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, se puede realizar el procesamiento a través de ciclos de reloj consecutivos. En una realización, se pueden ejecutar múltiples grupos de hilos simultáneamente 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 renunciar a una memoria caché interna y utilizar una memoria caché (por ejemplo, la memoria caché L1, 308) dentro del agrupación de procesamiento 214. Cada multiprocesador de gráficos 234 también tiene acceso a memorias caché L2 dentro de las unidades de subdivisión (por ejemplo, las unidades de subdivisión 220A a 220N de la Figura 2) que se comparten entre todas las agrupaciones de procesamiento 214 y se pueden utilizar para transferir datos entre hilos. El multiprocesador de gráficos 234 puede acceder también a una memoria global fuera de chip, que puede incluir una o más de una memoria de procesador paralelo local y/o una memoria de sistema. Se puede utilizar 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 se pueden almacenar en la memoria caché L1, 308.

Cada agrupación de procesamiento 214 puede incluir una MMU 245 (unidad de gestión de memoria) que está configurada para correlacionar direcciones virtuales con 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) utilizadas para correlacionar una dirección virtual con una dirección física de una tesela (más información sobre la generación de teselas) y, opcionalmente, un índice de línea de memoria caché. La MMU 245 puede incluir memorias intermedias de conversión anticipada de direcciones (TLB) o memorias caché que pueden residir dentro del multiprocesador de gráficos 234 o de la memoria caché L1 o la agrupación de procesamiento 214. La dirección física se procesa para distribuir la localidad de acceso a los datos de superficie para permitir un intercalado eficiente de solicitudes entre unidades de subdivisión. El índice de líneas de caché se puede utilizar para determinar si una solicitud de una línea de caché es un acierto o un error.

En aplicaciones gráficas y de computación, una agrupación de procesamiento 214 puede configurarse de manera que cada multiprocesador de gráficos 234 esté acoplado a una unidad de textura 236 para realizar operaciones de mapeo de textura, por ejemplo, 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 (por ejemplo, las unidades de subdivisión 220A a 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. Cualquier número de unidades de procesamiento, por ejemplo, multiprocesador de gráficos 234, unidades de textura 236, preROP 242, etc., pueden incluirse dentro de una agrupación de procesamiento 214. Además, aunque solo se muestra un grupo de procesamiento 214, una unidad de procesamiento paralela como se describe en el presente documento puede incluir cualquier número de instancias del grupo de procesamiento 214. En una realización, cada agrupación de procesamiento 214 puede estar configurada para operar independientemente de otras agrupaciones de procesamiento 214 utilizando unidades de procesamiento separadas y distintas, memorias caché L1, etc.

La Figura 2D muestra un multiprocesador de gráficos 234, según 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 incluyendo, aunque no de forma limitativa, una memoria caché de instrucciones 252, una unidad de instrucciones 254, una unidad de correlación de direcciones 256, un archivo de registros 258, uno o más núcleos de unidad de procesamiento de gráficos de propósito general (GPGPU) 262 y una o más unidades de carga/almacenamiento 266. Los núcleos de GPGPU 262 y las unidades de carga/almacenamiento 266 se acoplan 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 ejecutar desde el gestor de canal 232. Las instrucciones se almacenan en memoria caché en la caché de instrucciones 252 y se envían para su ejecución por la unidad de instrucciones 254. La unidad de instrucciones 254 puede enviar instrucciones como grupos de hilos (por ejemplo, urdimbres), 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 de un espacio de direcciones local, compartido o global especificando una dirección dentro de un espacio de direcciones unificado. Se puede utilizar la unidad de correlación de direcciones 256 para traducir direcciones en el espacio de direcciones unificado a una dirección de memoria distinta a la que pueden acceder 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 registros 258 proporciona almacenamiento temporal para operandos conectados a las rutas de datos de las unidades funcionales (por ejemplo, núcleos de GPGPU 262, 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 tal manera que a cada unidad funcional se le asigna una porción dedicada del archivo de registro 258. En una realización, el archivo de registros 258 se divide entre los diferentes conglomerados que son ejecutados por el multiprocesador de gráficos 324.

Los núcleos de GPGPU 262 pueden incluir cada uno unidades de coma flotante (FPU) y/o unidades lógicas aritméticas de números enteros (ALU) que se utilizan para ejecutar instrucciones del multiprocesador de gráficos 324. Los núcleos de GPGPU 262 pueden ser similares en arquitectura o pueden diferir en arquitectura, según 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 una aritmética de coma flotante o permitir una 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 copiar rectángulo o de mezcla de píxeles. En una realización, uno o más de los núcleos de GPGPU puede incluir también una lógica de función fija o especial.

En una realización, los núcleos GPGPU 262 incluyen una lógica SIMD capaz de realizar una única instrucción en múltiples conjuntos de datos. En una realización, los núcleos GPGPU 262 pueden ejecutar físicamente instrucciones SIMD4, SIMD8 y SIMD16 y ejecutar lógicamente instrucciones SIMD1, SIMD2 y SIMD32. Las instrucciones SIMD para los núcleos GPGPU pueden ser generadas en tiempo de compilación por un compilador sombreador o pueden generarse automáticamente cuando se ejecutan programas escritos y compilados para arquitecturas de único programa - múltiples datos (SPMD) o de SIMT. Se pueden ejecutar múltiples hilos de un programa configurado para el modelo de ejecución SIMT por medio de una única instrucción SIMD. Por ejemplo, y en una realización, se pueden ejecutar en paralelo ocho hilos SIMT que realizan operaciones iguales o similares mediante una única unidad lógica SIMD8.

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 memoria 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 registros 258. El archivo de registro 258 puede funcionar a la misma frecuencia que los núcleos de GPGPU 262, por tanto, la transferencia de datos entre los núcleos de GPGPU 262 y el archivo de registro 258 tiene una latencia muy baja. Se puede utilizar la memoria compartida 270 para posibilitar la comunicación entre hilos que se ejecutan en las unidades funcionales dentro del multiprocesador de gráficos 234. La memoria caché 272 se puede utilizar como una memoria caché de datos, por ejemplo, para almacenar en memoria caché datos de textura comunicados entre las unidades funcionales y la unidad de textura 236. La memoria compartida 270 también se puede utilizar como memoria caché gestionada por programas. Los hilos que se ejecutan en los núcleos de GPGPU 262 pueden almacenar, de manera programática, datos 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, según 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 configurarse como un multiprocesador de transmisión 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 según 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 instrucción 332A-332B, del archivo de registro 334A-334B y de 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 (por ejemplo, núcleo de GPGPU 336A-336B, núcleo de GPGPU 337A-337B, 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, una memoria caché de textura y/o de datos 342 y una memoria compartida 346.

Los diversos componentes pueden comunicarse mediante una estructura de interconexión 327. En una realización, la estructura 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, la estructura de interconexión 327 es una capa de tejido de red de alta velocidad separada sobre la que se apila cada componente del multiprocesador de gráficos 325. Los componentes del multiprocesador de gráficos 325 se comunican con componentes remotos mediante la estructura 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 la estructura de interconexión 327. La estructura de interconexión 327 puede arbitrar la comunicación dentro del multiprocesador de gráficos 325 para garantizar una asignación de ancho de banda equitativa entre los componentes.

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

Los expertos en la materia entenderán que la arquitectura descrita en las Figuras 1, 2A a 2D y 3A-3B es descriptiva y no limitativa 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 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 en 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 acoplarse de manera comunicativa al procesador/núcleos de anfitrión a través de un bus u otra interconexión (por ejemplo, una interconexión de alta velocidad tal como PCIe o NVLink). En otras realizaciones, la GPU puede integrarse en el mismo paquete o chip que los núcleos y acoplarse 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 que se conecta la GPU, los núcleos del procesador pueden asignar trabajo a la GPU en forma de secuencias de comandos/instrucciones contenidas en un descriptor de trabajo. Luego, la GPU utiliza circuitos/lógica dedicados para procesar eficientemente estos comandos/instrucciones.

Técnicas de interconexión de GPU y procesador anfitrión

La Figura 4A ilustra una arquitectura ilustrativa en la que una pluralidad de 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 (por ejemplo, buses, interconexiones de punto a punto, etc.). En una realización, los enlaces de alta velocidad 440-443 admiten un caudal de comunicación de 4 GB/s, 30 GB/s, 80 GB/s o mayor, dependiendo de la implementación. Se pueden utilizar varios protocolos de interconexión, incluyendo, aunque no de forma limitativa, 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 utilizando los mismos o diferentes protocolos/enlaces que los utilizados para los 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 multiprocesador simétrico (SMP) que operan a 20 GB/s, 30 GB/s, 120 GB/s o superior. Como alternativa, toda la comunicación entre los diversos componentes de sistema mostrados en la Figura 4A puede conseguirse utilizando los mismos

protocolos/enlaces (por ejemplo, a través de una estructura de interconexión común). Sin embargo, como se menciona, los principios subyacentes de la invención no están limitados a ningún tipo particular de tecnología de interconexión.

5 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 o diferentes tecnologías de acceso a la memoria. A modo de ejemplo, y no de limitación, las memorias de procesador 401-402 y las memorias de GPU 420-423 pueden ser memorias volátiles, tales como memorias de acceso aleatorio dinámico (DRAM) (incluyendo DRAM apiladas), SDRAM DDR de gráficos (GDDR) (por ejemplo, GDDR5, GDDR6), o memoria de alto ancho de banda (HBM) y/o pueden ser memorias no volátiles, tales como 3D XPoint o Nano-Ram. En una realización, una parte de las memorias puede ser memoria volátil y otra parte puede ser memoria no volátil (por ejemplo, utilizando una jerarquía de memoria de dos niveles (2LM)).

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

20 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 a través del enlace de alta velocidad 440. Como alternativa, el módulo de aceleración de gráficos 446 puede integrarse en el mismo paquete o chip que el procesador 407.

25 El procesador 407 ilustrado incluye una pluralidad de núcleos 460A-460D, cada uno con una memoria intermedia de conversión anticipada 461A-461D y una o varias memorias caché 462A-462D. Los núcleos pueden incluir diversos otros componentes para ejecutar instrucciones y procesar datos que no se ilustran para evitar complicar los principios subyacentes de la invención (por ejemplo, unidades de extracción de instrucciones, unidades de predicción de rama, decodificadores, unidades de ejecución, memoria intermedia de reordenamiento, etc.). Las cachés 462A-462D pueden comprender cachés de nivel 1 (L1) y de nivel 2 (L2). Además, una o más cachés compartidas 426 pueden incluirse en la jerarquía de almacenamiento en caché y ser compartidas por conjuntos de 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

30 Se mantiene la coherencia para datos e instrucciones almacenados en las diversas cachés 462A-462D, 456 y la memoria de sistema 441 mediante una comunicación entre núcleos mediante 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 memoria caché son ampliamente conocidas por los expertos en la técnica y no se describirán en detalle en el presente documento para no complicar los principios subyacentes de la invención.

35 En una realización, un circuito proxy 425 acopla en comunicación el módulo de aceleración de gráficos 446 al bus de coherencia 464, permitiendo al módulo de aceleración de gráficos 446 participar en el protocolo de coherencia de caché como un par de los núcleos. En particular, una interfaz 435 proporciona conectividad al circuito proxy 425 a través del enlace de alta velocidad 440 (por ejemplo, un bus PCIe, NVLink, etc.) y una interfaz 437 conecta el módulo de aceleración de gráficos 446 al enlace de alta velocidad 440.

40 En una implementación, un circuito de integración de acelerador 436 proporciona servicios de gestión de caché, de acceso a 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. Los motores de procesamiento de gráficos 431, 432, N pueden comprender cada uno 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 (por ejemplo, codificadores/decodificadores de vídeo), muestreadores y motores 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 GPU individuales integradas en un paquete, tarjeta de línea o chip común.

- 5 En una realización, el circuito de integración del acelerador 436 incluye una unidad de gestión de memoria (MMU) 439 para realizar varias funciones de gestión de memoria, tal como conversiones de memoria virtual a física (también denominadas conversiones de memoria efectiva a real) y protocolos de acceso a memoria para acceder a la memoria del sistema 441. La MMU 439 también puede incluir una memoria intermedia de traducción adelantada (TLB) (no mostrada) para almacenar en caché las traducciones de dirección virtual/efectiva a física/real. En una implementación, una caché 438 almacena comandos y datos para un acceso eficiente por parte de 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 proxy 425 que toma parte en el mecanismo de coherencia de caché en beneficio de la caché 438 y las memorias 433-434, N (por ejemplo, enviando actualizaciones a la caché 438 relacionadas con modificaciones/accesos de líneas de caché en las cachés de procesador 462A-462D, 456 y recibiendo actualizaciones desde la caché 438).
- 10
- 15 Un conjunto de registros 445 almacena 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 guardado y restauración para guardar y restaurar contextos de los diversos hilos durante cambios de contexto (por ejemplo, cuando se guarda un primer hilo y se almacena un segundo hilo de modo que el segundo hilo puede ser ejecutado por un motor de procesamiento de gráficos). Por ejemplo, en una conmutación de contexto, el circuito de gestión de contexto 448 puede almacenar valores de registro actuales en una región designada en memoria (por ejemplo, identificada por un puntero de contexto). Luego, puede restaurar los valores de registro cuando se vuelve al contexto. En una realización, un circuito de gestión de interrupciones 447 recibe y procesa interrupciones recibidas de dispositivos del sistema.
- 20
- 25 En una implementación, las direcciones virtuales/efectivas de un motor de procesamiento de gráficos 431 son traducidas a direcciones reales/físicas en la memoria del sistema 411 por la MMU 439. Una realización del circuito de integración de acelerador 436 admite múltiples (por ejemplo, 4, 8, 16) módulos de aceleración de gráficos 446 y/u otros dispositivos de aceleración. El módulo 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 virtualizados 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.
- 30
- 35 Por 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 memoria caché de sistema. Además, el circuito de integración de acelerador 436 puede proporcionar instalaciones de virtualización para que el procesador anfitrión gestione la virtualización de los motores de procesamiento de gráficos, las interrupciones y la gestión de memoria.
- 40
- 45 Como los recursos de hardware de los motores de procesamiento de gráficos 431-432, N se correlacionan explícitamente al espacio de direcciones real visto por el procesador anfitrión 407, cualquier procesador anfitrión puede direccionar estos recursos directamente utilizando un valor de dirección efectiva. 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 para que aparezcan ante el sistema como unidades independientes.
- 50
- 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 procesa 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 las DRAM apiladas), memoria GDDR (por ejemplo, GDDR5, GDDR6) o HBM, y/o pueden ser memorias no volátiles tales como 3D XPoint o Nano-Ram.
- 55
- En una realización, para reducir el tráfico de datos a través del enlace 440, se utilizan técnicas de desvío para garantizar que los datos almacenados en las memorias de gráficos 433-434, M sean datos que serán utilizados con mayor frecuencia por los motores de procesamiento de gráficos 431-432, N y preferentemente no serán utilizados por los núcleos 460A-460D (al menos no con frecuencia). De manera similar, el mecanismo de desvío intenta mantener los 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 de la memoria de sistema 411.
- 60
- 65 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, pueden 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 con un caudal superior dada su estrecha proximidad al bus de coherencia 462 y a las memorias caché 462A-462D, 426.

Una realización admite diferentes modelos de programación, incluyendo un modelo de programación de proceso dedicado (sin virtualización del módulo de aceleración de gráficos) y modelos de programación compartidos (con virtualización). Esta última puede incluir modelos de programación que están controlados por el circuito de integración del acelerador 436 y modelos de programación que están controlados por el módulo de aceleración de gráficos 446.

En una realización del modelo de proceso dedicado, los motores de procesamiento de gráficos 431-432, N, están dedicados a 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, proporcionando virtualización dentro de una VM/subdivisión.

En los modelos de programación de proceso dedicado, los motores de procesamiento de gráficos 431-432, N, pueden estar compartidos por múltiples particiones de VM/aplicación. Los modelos compartidos requieren un hipervisor de sistema para virtualizar los motores de procesamiento de gráficos 431-432, N para permitir el acceso de cada sistema operativo. En sistemas de subdivisión única sin un hipervisor, los motores de procesamiento de gráficos 431-432, N pertenecen al 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 431-432, N individual selecciona un elemento de proceso utilizando un manejador de proceso. En una realización, los elementos de proceso se almacenan en la memoria de sistema 411 y se pueden direccionar utilizando las técnicas de conversión de dirección efectiva 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 a 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. Tal y como se utiliza en el presente documento, un "corte" comprende una parte específica de los recursos de procesamiento del circuito de integración de acelerador 436. El espacio de direcciones efectivas de aplicación 482 dentro de la memoria de sistema 411 almacena elementos de proceso 483. En una realización, los elementos de proceso 483 se almacenan en respuesta a invocaciones de GPU 481 desde aplicaciones 480 ejecutadas en el procesador 407. Un elemento de proceso 483 contiene el estado de proceso para la aplicación correspondiente 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 de una cola de trabajos. En el último caso, el WD 484 es un puntero de 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 431-432, N individuales pueden ser compartidos por todos, o por 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 iniciar un trabajo en un entorno virtualizado.

En una implementación, el modelo de programación de procesos dedicados 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 431 individual. Debido a que el módulo de aceleración de gráficos 446 es propiedad de un único proceso, el hipervisor inicializa el circuito de integración de acelerador 436 para la subdivisión propietaria y el sistema operativo inicializa el circuito de integración de acelerador 436 para el proceso de propiedad en el momento en que se asigna el módulo de aceleración de gráficos 446.

En funcionamiento, una unidad de extracción de WD 491 en el corte de integración del acelerador 490 extrae el siguiente WD 484 que incluye una indicación del trabajo que debe realizar uno de los motores de procesamiento de gráficos del módulo de aceleración de gráficos 446. Los datos del WD 484 pueden ser almacenados en los registros 445 y utilizados 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 recorrido de segmentos/páginas para acceder a las tablas de segmentos/páginas 486 dentro del espacio de direcciones virtuales de SO 485. El circuito de gestión de interrupciones 447 puede procesar eventos de interrupción 492 recibidos desde el módulo de aceleración de gráficos 446. Al realizar operaciones de gráficos, una dirección efectiva 493 generada por un motor de procesamiento de gráficos 431-432, N se traduce a una dirección real mediante la MMU 439.

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

Tabla 1 - Registros inicializados por el hipervisor

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

En la Tabla 2 se muestran registros ilustrativos que pueden ser inicializados por el sistema operativo.

5 Tabla 2 - Registros inicializados por el sistema operativo

| | |
|---|--|
| 1 | Identificación de procesos e hilos |
| 2 | Puntero de guardado/restauración de contexto de dirección efectiva (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 |

10 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.

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

20 Los modelos de programación compartida permiten que todos o un subconjunto de procesos de todas o un subconjunto de 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 es compartido por múltiples procesos y subdivisiones: compartido en cortes de tiempo y compartido dirigido a gráficos.

25 En este modelo, el hipervisor del sistema 496 posee el 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 admita la virtualización por parte del hipervisor del sistema 496, el módulo de aceleración de gráficos 446 puede adherirse a los siguientes requisitos: 1) La solicitud de trabajo de una aplicación debe ser autónoma (es decir, no es necesario mantener el estado entre trabajos), o el módulo de aceleración de gráficos 446 debe proporcionar un mecanismo de guardado y restauración de contexto. 2) El módulo de aceleración de gráficos 446 garantiza que la solicitud de trabajo de una aplicación se completará en una cantidad específica de tiempo, incluyendo cualquier error de traducción o el módulo de aceleración de gráficos 446 proporciona la capacidad de adelantarse al procesamiento del trabajo. 3) Se ha de garantizar al módulo de aceleración de gráficos 446 la equidad entre procesos durante el funcionamiento en el modelo de programación compartido dirigido.

35 En una realización, para el modelo compartido, se requiere que la aplicación 480 realice una llamada de sistema al 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 a área de guardado/restauración de contexto (CSRP). El tipo del módulo de aceleración de gráficos 446 describe la función de aceleración dirigida como objetivo para la llamada de sistema. El tipo del módulo de aceleración de gráficos 446 puede ser un valor específico del 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 efectiva a una estructura definida por el usuario, un puntero de dirección efectiva 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 a utilizar para el proceso actual. El valor pasado al sistema operativo es similar a que una aplicación establezca el AMR. Si las implementaciones del circuito de integración del acelerador 436 y del módulo de aceleración de gráficos 446 no admiten un registro de anulación de máscara de autoridad de usuario (UAMOR), el sistema operativo puede

5 aplicar el valor de UAMOR actual al valor de AMR antes de pasar el AMR en la llamada del hipervisor. El hipervisor 496 puede aplicar opcionalmente el valor actual del registro de anulación de máscara de autoridad (AMOR) antes de colocar el AMR en el elemento de proceso 483. En una realización, el CSRFP es uno de los registros 445 que contienen la dirección efectiva de un área en el espacio de direcciones 482 de la aplicación para que el módulo de aceleración de gráficos 446 guarde y restablezca el estado de contexto. Este puntero es opcional si no es necesario guardar un estado entre trabajos o cuando se adelanta un trabajo. El área de guardado/restauración del contexto puede estar anclada en la memoria del sistema.

10 Tras recibir la llamada de sistema, el sistema operativo 495 puede verificar que la aplicación 480 se ha registrado y que se le ha dado la autoridad para utilizar 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 del SO al hipervisor

| | |
|---|---|
| 1 | Un descriptor de trabajo (WD) |
| 2 | Un valor de registro de máscara de autoridad (AMR) (potencialmente enmascarado). |
| 3 | Un puntero a área de guardado/restauración de contexto (CSRFP) de dirección efectiva (EA) |
| 4 | Un ID de proceso (PID) y un 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) |

15 Al recibir la llamada del hipervisor, el hipervisor 496 verifica que el sistema operativo 495 se haya registrado y se le haya otorgado autoridad para utilizar el módulo de aceleración de gráficos 446. El hipervisor 496 pone, a continuación, el elemento de proceso 483 en la lista vinculada de elementos de proceso para el tipo del módulo de aceleración de gráficos 446 correspondiente. El elemento de proceso puede incluir la información mostrada en la Tabla 4.

20 Tabla 4 - Información de elemento de proceso

| | |
|----|---|
| 1 | Un descriptor de trabajo (WD) |
| 2 | Un valor de registro de máscara de autoridad (AMR) (potencialmente enmascarado). |
| 3 | Un puntero a área de guardado/restauración de contexto (CSRFP) de dirección efectiva (EA) |
| 4 | Un ID de proceso (PID) y un 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 identificador 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 del corte de integración de acelerador 490.

25 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 utilizado 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/efectivo para acceder a las memorias de procesador 401-402 y viceversa, simplificando de esta manera la programabilidad. En una realización, una primera porción del espacio de direcciones virtual/efectiva 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/efectiva total (en ocasiones denominado el espacio de direcciones efectiva) 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 correlacionada con esa memoria.

35 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 memorias caché de los procesadores de anfitrión (por ejemplo, 405) y las GPU 410-413 e implementa técnicas de desvío que indican las memorias físicas en las que deben almacenarse

ciertos tipos 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.

5 Una realización permite que la memoria conectada a la GPU 420-423 se mapee como parte de la memoria de sistema, y que se acceda a ella utilizando 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 acceder a la memoria conectada a la GPU 420-423 como memoria del sistema sin una onerosa sobrecarga de coherencia de caché proporciona un entorno operativo beneficioso para la descarga de la GPU. Esta disposición permite que el software del procesador anfitrión 405 configure operandos y acceda a los resultados de los cálculos, sin la sobrecarga de las copias de datos DMA de E/S tradicionales. Estas copias tradicionales implican llamadas de controlador, interrupciones y accesos de E/S correlacionadas en la memoria (MMIO) que son todos ineficientes en relación con los accesos a la memoria simple. Al mismo tiempo, la capacidad de acceder a la memoria conectada a la GPU 420-423 sin sobrecargas de coherencia de memoria caché puede ser crucial para el tiempo de ejecución de un cálculo descargado. En casos con un tráfico sustancial en la memoria de escritura de transmisión por flujo continuo, 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 eficiencia del establecimiento de operandos, la eficiencia del acceso a resultados y la eficiencia del cómputo de GPU desempeñan, todas ellas, un papel en la determinación de la eficacia de la descarga de GPU.

20 En una implementación, la selección entre el desvío de GPU y el desvío de procesador anfitrión está determinada por una estructura de datos de rastreador de desvío. Se puede utilizar una tabla de desvío, por ejemplo, que puede ser una estructura granular a nivel de página (es decir, controlada en la granularidad de una página de memoria) que incluye 1 o 2 bits por página de memoria conectada a la GPU. La tabla de desvíos puede implementarse en un intervalo de memoria robado de una o más memorias conectadas a la GPU 420-423, con o sin caché de desvío en la GPU 410-413 (por ejemplo, para almacenar en memoria caché entradas utilizadas de manera frecuente/reciente de la tabla de desvíos). Como alternativa, la tabla de desvío completa puede mantenerse dentro de la GPU.

En una implementación, se accede a la entrada de la tabla de desvío asociada con cada acceso a la memoria conectada a la GPU 420-423 antes del acceso real a la memoria de la 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 la GPU se reenvían directamente a una memoria de GPU correspondiente 420-423. Las solicitudes locales de la GPU que encuentran su página en el desvío del anfitrión se reenvían al procesador 405 (por ejemplo, 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 sesgada por GPU pueden reenviarse a la GPU 410-413. La GPU puede hacer, a continuación, que la página realice una transición a un desvío de procesador de anfitrión si actualmente no está utilizando la página.

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

Un mecanismo para cambiar el estado de desvío emplea una llamada de API (por ejemplo, OpenCL), que, a su vez, llama al controlador de dispositivos de la GPU que, a su vez, envía un mensaje a (o pone en cola un descriptor de comandos para) la GPU que le indica que cambie el estado de desvío y, para algunas transiciones, que realice una operación de vaciado de caché en el anfitrión. La operación de vaciado de caché es necesaria para una transición del desvío de procesador anfitrión 405 al desvío de GPU, pero no es necesaria para la transición opuesta.

En una realización, la coherencia de caché se mantiene haciendo que temporalmente las páginas desviadas por la GPU no puedan ser almacenadas en la caché por el procesador 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 tanto, para reducir la comunicación entre el procesador 405 y la GPU 410 es beneficioso garantizar que las páginas con desvío por la GPU sean aquellas requeridas por la GPU, pero no por el procesador anfitrión 405 y viceversa.

Canalización de procesamiento de gráficos

La Figura 5 ilustra una canalización de procesamiento de gráficos 500, según una realización. En una realización, un procesador gráfico puede implementar la canalización de procesamiento de gráficos 500 ilustrada. El procesador gráfico puede incluirse dentro de los subsistemas de procesamiento en paralelo como se describe en el presente documento, tal como el procesador en paralelo 200 de la Figura 2, que, en una realización, es una variante del procesador o procesadores en paralelo 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 (por ejemplo, la unidad de procesamiento paralelo 202 de la Figura 2) como se describe en el presente documento. Por ejemplo, una unidad sombreadora (por ejemplo, el multiprocesador de gráficos 234 de la Figura 3) puede configurarse para realizar las funciones de una o más de una unidad de procesamiento de vértices

504, una unidad de procesamiento de control 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 de rasterización 526 también pueden realizarse por otros motores de procesamiento dentro de una agrupación de procesamiento (por ejemplo, la agrupación de procesamiento 214 de la Figura 3) y una unidad de subdivisión correspondiente (por ejemplo, la unidad de subdivisión 220A-220N de la Figura 2). La canalización de procesamiento de gráficos 500 también puede implementarse utilizando unidades de procesamiento especializadas para una o más funciones. En una realización, una o más porciones de la canalización de procesamiento de gráficos 500 pueden realizarse mediante lógica de procesamiento paralelo dentro de un procesador de propósito general (por ejemplo, CPU). En una realización, una o más porciones de la canalización de procesamiento de gráficos 500 pueden acceder a una memoria en chip (por ejemplo, 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. A continuación, el ensamblador de datos 502 envía los datos de vértices, incluyendo los atributos de vértice, 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 sombreado de vértices, iluminando y transformando datos de vértices según lo especificado por los programas de sombreado de vértices. La unidad de procesamiento de vértices 504 lee datos que se almacenan en caché, en memoria local o de sistema para su utilización al procesar los datos de vértices y puede estar programada para transformar los datos de vértices de una representación de coordenadas basada en objeto a un espacio de coordenadas del espacio mundial o un espacio de coordenadas de dispositivo normalizado.

Una primera instancia de un ensamblador de primitivas 506 recibe atributos de vértices 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 lo admitido por varias interfaces de programación de aplicaciones (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 del parche (por ejemplo, las bases del parche) a una representación que sea adecuada para su utilización en la evaluación de la superficie 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 se cuantifica un nivel de detalle dependiente de la vista 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 primitivas de línea, de triángulo o cuadriláteros, que se transmiten a una unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de evaluación de teselación 512 realiza operaciones en coordenadas parametrizadas del parche subdividido para generar una representación de superficie y atributos de vértice para cada vértice asociado a 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 sombreado de geometría para transformar primitivas de gráficos recibidas del ensamblador de primitivas 514 según lo especificado por los programas de sombreado 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 los parámetros utilizados para rasterizar las primitivas de gráficos nuevas.

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 envía los parámetros y vértices que especifican nuevas primitivas de gráficos al ensamblador de primitivas 518. El ensamblador de primitivas 518 recibe los parámetros y vértices de la unidad de procesamiento de geometría 516 y construye primitivas de gráficos para su procesamiento por una unidad de escala, selección y recorte de ventana gráfica 520. La unidad de procesamiento de geometría 516 lee datos que se almacenan en memoria de procesador paralelo o memoria de sistema para su utilización en el procesamiento de datos de geometría. La unidad de escala, selección y recorte de ventana gráfica 520 realiza el recorte, la selección y el escalado de la ventana gráfica y envía primitivas de gráficos procesadas 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 una conversión de exploración en las nuevas primitivas de gráficos para generar fragmentos y emitir esos 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

5 programable que está configurada para ejecutar programas sombreadores de fragmentos o programas sombreadores de píxeles. La unidad de procesamiento de fragmentos/píxeles 524 transforma fragmentos o píxeles recibidos desde el rasterizador 522, según lo especificado por los programas de sombreado de fragmentos o píxeles. Por ejemplo, la unidad de procesamiento de fragmentos/píxeles 524 puede programarse para realizar operaciones que incluyen, aunque no de forma limitativa, mapeo de texturas, sombreado, combinación, corrección de texturas y corrección de perspectiva para producir fragmentos o píxeles sombreados que se envían a una unidad de operaciones de rasterización 526. La unidad de procesamiento de fragmentos/píxeles 524 puede leer datos que están almacenados bien en la memoria del procesador en paralelo o bien en la memoria del sistema para su utilización cuando se procesan los datos de fragmentos. Los programas de sombreado de fragmentos o píxeles se pueden configurar para sombrear muestras, píxeles, teselas u otras granularidades dependiendo de la velocidad de muestreo configurada para las unidades de procesamiento.

15 La unidad de operaciones de rasterización 526 es una unidad de procesamiento que realiza operaciones de rasterización que incluyen, aunque no de forma limitativa, 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 (por ejemplo, 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.

Procesamiento basado en regiones

25 Pasando ahora a la Figura 6A, se muestra una imagen de alto rango dinámico (HDR) 600 (por ejemplo, imagen fija, fotograma de vídeo). La imagen HDR 600, que también puede denominarse imagen de rango dinámico mejorado (EDR) y/o de rango dinámico amplio (WDR), generalmente puede formar parte de un contenido visual presentado a un usuario junto con un juego tridimensional (3D), un entorno de realidad virtual (RV), un entorno de realidad aumentada (RA), etcétera. Por ejemplo, la imagen HDR 600 se puede transmitir desde un dispositivo fuente a través de un enlace inalámbrico (por ejemplo, Bluetooth, WiFi) a un sistema de pantalla montada en la cabeza (HMD) que lleve el usuario puesto para proporcionar una experiencia de RV inmersiva en tiempo real. En otro ejemplo, la imagen HDR 600 se transmite a través de un enlace por cable (por ejemplo, bus serie universal/USB) a un sistema de visualización fijo que está viendo el usuario para reproducir una película. Puede considerarse que la imagen 600 está formateada en HDR en la medida en que reproduce un rango dinámico de luminosidad mayor que el que es posible con las técnicas fotográficas o de imagen digital estándar. En consecuencia, la imagen HDR 600 puede tener un intervalo de luminancia similar al experimentado a través del sistema visual humano, que se ajusta constantemente para adaptarse a un amplio rango de luminancia presente en el entorno. El cerebro puede interpretar continuamente esta información para que el usuario/espectador pueda ver en una amplia gama de condiciones de luz.

40 En el ejemplo ilustrado, el dispositivo fuente divide la imagen HDR 600 en una pluralidad de regiones, en donde se puede medir automáticamente un nivel de luminancia para cada región. Por ejemplo, se puede determinar que una primera región 602 y una segunda región 604 tienen un nivel de luminancia relativamente alto, mientras que se puede determinar que una o más regiones restantes 606 tienen un nivel de luminancia relativamente bajo. Como se analizará con más detalle, las regiones 602, 604, 606 pueden codificarse independientemente (es decir, comprimirse) antes de su transmisión a través del enlace basándose en sus respectivos niveles de luminancia. Más particularmente, las regiones más brillantes tales como las regiones 602, 604 pueden recibir menos codificación/compresión con el fin de preservar el contenido HDR. Por el contrario, las regiones más oscuras, tales como las regiones restantes 606, pueden recibir más codificación/compresión para reducir el consumo de ancho de banda del enlace y/o el consumo de energía. Cabe señalar, en particular, que el aumento de la cantidad de codificación en las regiones restantes 606 relativamente oscuras puede tener un impacto poco o nada perceptible en la calidad de la imagen debido a la falta de contenido HDR en las regiones restantes 606.

55 La Figura 6B muestra otra imagen HDR 620 que también puede formar parte del contenido visual presentado a un usuario en un juego 3D, entorno de RV, entorno de RA, etc. En el ejemplo ilustrado, la imagen HDR está dividida por el dispositivo fuente en una región en primer plano 622 y una región de fondo 624. Las regiones 622, 624 pueden codificarse independientemente antes de su transmisión por el enlace basándose en su profundidad en la escena (por ejemplo, estado de primer plano/fondo). De manera más particular, las regiones más cercanas, tal como la región en primer plano 622, pueden recibir menos codificación/compresión para garantizar la calidad visual de las porciones más importantes de la escena. Por el contrario, las regiones más alejadas, tales como las regiones de fondo 624, pueden recibir más codificación/compresión para reducir el ancho de banda del enlace y/o el consumo de energía.

60 De hecho, el enfoque ilustrado de primer plano/fondo se puede utilizar junto con el enfoque de luminancia ilustrado en la Figura 6A. Por ejemplo, la región de fondo 624 puede codificarse/comprimirse a un nivel superior incluso si se detecta una luminancia relativamente alta en la región de fondo 624. Del mismo modo, la región en primer plano 622 puede codificarse/comprimirse a un nivel inferior incluso si se detecta una luminancia relativamente baja en la región en primer plano 622.

La Figura 7 muestra una arquitectura de procesamiento 700 que utiliza el procesamiento basado en regiones como se describe en el presente documento. En el ejemplo ilustrado, un controlador de dispositivo 702 recibe una imagen HDR 704 (por ejemplo, una imagen fija, un fotograma de vídeo), divide la imagen HDR 704 en una pluralidad de regiones y determina, por región, un nivel de luminancia de la imagen HDR 704. El controlador de dispositivo 702 también puede determinar si cada región está en una región en primer plano y/o de fondo. El controlador de dispositivo 702 ilustrado también envía una cantidad de codificación específica de una región 706 a una canalización de medios 708 (por ejemplo, de un procesador gráfico, no mostrado) que utiliza un códec 710 para codificar las regiones de la imagen HDR 704 de acuerdo con la cantidad de codificación específica de una región 706. El controlador de dispositivo 702 también puede activar una o más interrupciones del procesador gráfico para cambiar el códec 710 entre esquemas de codificación. Los datos codificados se pueden enviar a un transmisor 712, que envía los datos codificados a una pantalla inalámbrica 714 (por ejemplo, incluyendo un receptor, no mostrado). Como alternativa, los datos codificados se pueden enviar mediante un enlace cableado a una pantalla no inalámbrica (no mostrada). En cualquier caso, la arquitectura ilustrada 700 puede reducir el consumo de ancho de banda del enlace, el consumo de energía y el coste, manteniendo al mismo tiempo la calidad de la imagen desde la perspectiva del usuario (por ejemplo, el sistema visual humano). Las operaciones del controlador de dispositivo 702 pueden ser realizadas por uno o más de otros componentes tales como, por ejemplo, un canal de visualización (por ejemplo, incluyendo un componente de mejora de contraste adaptativo local/LACE, no mostrado).

La Figura 8A muestra un método 800 de funcionamiento de un aparato de paquete de semiconductores. En particular, el método 800 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 tal 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 utiliza tecnología de circuito tal como, por ejemplo, tecnología de circuito integrado de aplicación específica (ASIC), de semiconductor complementario de óxido metálico (CMOS) o de lógica transistor-transistor (TTL), o cualquier combinación de las mismas.

Por ejemplo, el código de programa informático para llevar a cabo las operaciones mostradas en el método 800 puede escribirse en cualquier combinación de uno o más lenguajes de programación, incluido un lenguaje de programación orientado a objetos tal como JAVA, SMALLTALK, C++ o similares y lenguajes de programación procedimentales convencionales, tal como el lenguaje de programación "C" o lenguajes de programación similares. Adicionalmente, las instrucciones lógicas pueden incluir instrucciones de ensamblador, instrucciones de arquitectura de conjunto de instrucciones (ISA), instrucciones de máquina, instrucciones dependientes de la máquina, microcódigo, datos de configuración de estado, datos de configuración para circuitos integrados, información de estado que personaliza circuitería electrónica y/u otros componentes estructurales que son nativos del hardware (por ejemplo, procesador de anfitrión, unidad central de procesamiento/CPU, microcontrolador, etc.).

El bloque lógico configurable 802 ilustrado provee la división de una imagen HDR en una pluralidad de regiones. El bloque 802 puede incluir la recuperación de datos de subdivisión desde un registro de control u otra ubicación de almacenamiento adecuada, la generación de los datos de subdivisión, etc., por imagen/fotograma o para múltiples imágenes/fotogramas. El espaciado/tamaño de las subdivisiones puede variar dependiendo de las circunstancias. Adicionalmente, en el bloque 804 se puede determinar un nivel de luminancia de la imagen HDR por región. El bloque 804 puede incluir la generación de un histograma de luminancia para la imagen HDR. De manera más particular, se pueden definir intervalos de luminancia, en donde se puede determinar el porcentaje de la región que cae en cada contenedor. Por ejemplo, el 75 % de una primera región puede estar comprendida en un contenedor de luminancia de X candelas por metro cuadrado ($X \text{ cd/m}^2$) y el 25 % de la primera región puede estar comprendida en un contenedor de luminancia de $Y \text{ cd/m}^2$, en donde $Y > X$. Por el contrario, el 5 % de una segunda región puede caer en un contenedor de luminancia X, mientras que el 95 % de la segunda región cae en el contenedor de luminancia Y. En tal ejemplo, se puede considerar que la primera región contiene contenido HDR, mientras que puede considerarse que la segunda región no contiene contenido HDR. El bloque ilustrado 806 selecciona, por región, una cantidad de codificación para cada región de la pluralidad de regiones basándose en el nivel de luminancia.

La Figura 8B muestra un método 820 de selección de cantidades de codificación basándose en el nivel de luminancia. Por lo tanto, el método 820 se puede incorporar en el bloque 806 (Figura 8A), ya analizado. El método 820 se puede implementar en general en una arquitectura de procesamiento tal como, por ejemplo, la arquitectura 700 (Figura 7). De manera más particular, el método 820 se puede implementar 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 RAM, ROM, PROM, firmware, memoria flash, etc., en lógica configurable tal como, por ejemplo, PLA, FPGA, CPLD, en lógica de hardware de funcionalidad fija que utiliza tecnología de circuito tal como, por ejemplo, tecnología ASIC, CMOS o TTL, o cualquier combinación de las mismas.

El bloque de procesamiento 822 ilustrado puede proporcionar la inicialización de un primer umbral y un segundo umbral, en donde el segundo umbral es mayor o igual que el primer umbral. En el bloque 824 se puede determinar si el nivel de luminancia de la región en cuestión supera el segundo umbral. El nivel de luminancia puede incluir, por ejemplo, el contenedor de luminancia que tenga el porcentaje más alto, una media ponderada de los contenedores de

luminancia, y así sucesivamente. Si se supera el segundo umbral, el bloque 826 puede disminuir la cantidad de codificación, en donde la disminución de la cantidad de codificación disminuye la compresión de la imagen (por ejemplo, y a su vez aumenta la calidad de la imagen). El bloque 826 puede incluir la selección de un esquema de codificación (por ejemplo, H.264, H.265, etc.) y/o un parámetro de cuantificación (QP) que dé como resultado la cantidad de codificación seleccionada. Por tanto, la disminución puede ser conforme a los niveles de cuantificación disponibles y/o a otras restricciones de codificación. El bloque ilustrado 828 codifica la región con la cantidad de codificación seleccionada. En un ejemplo, el bloque 828 incluye dar instrucciones a un códec para que codifique la región.

Si en el bloque 824 se determina que no se ha superado el segundo umbral, el bloque 830 se puede proveer la determinación de si el nivel de luminancia (por ejemplo, el contenedor de luminancia que tiene el porcentaje más alto, una media ponderada de los contenedores de luminancia, etc.) no supera el primer umbral. Si el nivel de luminancia no supera el primer umbral, el bloque ilustrado 832 aumenta la cantidad de codificación, lo que aumenta la compresión de la imagen. Aumentar la compresión de la imagen también puede disminuir la calidad de la imagen (por ejemplo, si se utiliza compresión con pérdida). El bloque 832 también puede incluir la selección de un esquema de codificación y/o QP que dé como resultado la cantidad de codificación seleccionada, en donde el aumento puede ser conforme a los niveles de cuantificación disponibles y/o a otras restricciones de codificación. Una vez aumentada la cantidad de codificación, el bloque ilustrado 828 codifica la región con la cantidad de codificación seleccionada. Adicionalmente, el bloque 828 puede codificar una primera región de la pluralidad de regiones con una primera cantidad de codificación y codificar una segunda región de la pluralidad de regiones con una segunda cantidad de codificación. Inicializar el segundo umbral a un valor que es mayor que el valor del primer umbral puede proporcionar un efecto de tipo histéresis que elimina la conmutación continua entre el control de alta luminancia y el control de baja luminancia.

La Figura 8C muestra un método 840 de selección de cantidades de codificación para las regiones de fondo y primer plano de una imagen HDR. Por lo tanto, el método 840 se puede incorporar en el bloque 806 (Figura 8A), ya considerado, junto con, por ejemplo, el método 820 (Figura 8B), ya analizado. De manera más particular, el método 840 se puede implementar 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 RAM, ROM, PROM, firmware, memoria flash, etc., en lógica configurable tal como, por ejemplo, PLA, FPGA, CPLD, en lógica de hardware de funcionalidad fija que utiliza tecnología de circuito tal como, por ejemplo, tecnología ASIC, CMOS o TTL, o cualquier combinación de las mismas.

El bloque de procesamiento 842 ilustrado provee la detección de una región en primer plano de la pluralidad de regiones. La determinación del bloque 842 puede hacerse basándose en información del sistema operativo (SO) (por ejemplo, "pistas" del área de enfoque, datos de seguimiento ocular, etc.) y/o en otra información, dependiendo de las circunstancias. El bloque ilustrado 844 disminuye la cantidad de codificación en respuesta a la detección de la región en primer plano, en donde la disminución de la cantidad de codificación disminuye la compresión de la imagen (por ejemplo, y a su vez aumenta la calidad de la imagen). Una vez disminuida la cantidad de codificación, el bloque 850 puede codificar la región con la cantidad de codificación seleccionada. En un ejemplo, el bloque 850 incluye dar instrucciones a un códec para que codifique la región.

Si no se detecta una región en primer plano en el bloque 842, en el bloque 846 se puede determinar si la región en cuestión es una región de fondo. Si es así, el bloque 848 puede aumentar la cantidad de codificación, en donde el aumento de la cantidad de codificación aumenta la compresión de la imagen. Como ya se ha señalado, aumentar la compresión de la imagen también puede disminuir la calidad de la imagen (por ejemplo, si se utiliza compresión con pérdida). En este sentido, aumentar la cantidad de codificación puede tener un impacto insignificante en la calidad de imagen percibida porque la región está en el fondo (por ejemplo, no es el centro de enfoque). Una vez aumentada la cantidad de codificación, el bloque ilustrado 850 codifica la región con la cantidad de codificación seleccionada.

La Figura 9 muestra un sistema informático de rendimiento mejorado 900. En el ejemplo ilustrado, un procesador de anfitrión 902 incluye un controlador de memoria integrado (IMC) 904 que se comunica con una memoria de sistema 906 (por ejemplo, DRAM). El procesador de anfitrión 902 puede estar acoplado a un procesador gráfico 908 (es decir, mediante un bus de interconexión de componentes periféricos/PCI) y a un módulo de entrada/salida (ES) 910. El módulo de ES 910 se puede acoplar a un controlador de red 912 (por ejemplo, inalámbrica y/o cableada), una pantalla 914 (por ejemplo, pantalla de cristal líquido/LCD fija o montada en la cabeza, diodo emisor de luz/pantalla LED, etc., para presentar visualmente una escena tridimensional/3D) almacenamiento masivo 918 (por ejemplo, memoria flash, disco óptico, unidad de estado sólido/SSD). El procesador gráfico 908 ilustrado incluye una canalización de medios 920 y está acoplado a una memoria de gráficos 916 (por ejemplo, RAM dedicada para gráficos).

La canalización de medios 920 puede incluir un códec tal como, por ejemplo, el códec 710 (Figura 7). Adicionalmente, la memoria de sistema 906 y/o el almacenamiento masivo 918 pueden incluir un conjunto de instrucciones 922 (por ejemplo, instrucciones de controlador de dispositivo), que cuando son ejecutadas por el procesador de anfitrión 902 y/o el procesador gráfico 908, hacen que el sistema 900 implemente uno o más aspectos del método 800 (Figura 8A), del método 820 (Figura 8B) y/o del método 840 (Figura 8C), ya analizados. Por tanto, la ejecución de las instrucciones 922 puede hacer que el sistema 900 divida una imagen HDR en una pluralidad de regiones, determine, por región, un nivel de luminancia de la imagen HDR y seleccione, por región, una cantidad de codificación para cada región de la

pluralidad de regiones basándose en el nivel de luminancia. Adicionalmente, la ejecución de las instrucciones 922 puede hacer que el sistema 900 codifique selectivamente regiones en primer plano y de fondo.

La Figura 10 muestra un aparato de paquete de semiconductores 1000 (por ejemplo, chip) que incluye un sustrato 1002 (por ejemplo, silicio, zafiro, arseniuro de galio) y una lógica 1004 (por ejemplo, matriz de transistores y otros componentes de circuito integrado/CI) acoplados al sustrato 1002. La lógica 1004, que puede implementarse, por ejemplo, en una lógica configurable y/o lógica hardware de funcionalidad fija, puede implementar en general uno o más aspectos del método 800 (Figura 8A), del método 820 (Figura 8B) y/o del método 840 (Figura 8C), ya analizados. Por tanto, la lógica 1004 puede dividir una imagen HDR en una pluralidad de regiones, determinar, por región, un nivel de luminancia de la imagen HDR y seleccionar, por región, una cantidad de codificación para cada región de la pluralidad de regiones basándose en el nivel de luminancia. Adicionalmente, la lógica 1004 puede codificar selectivamente regiones en primer plano y de fondo.

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 gráfico (por ejemplo, unidad de procesamiento de gráficos/GPU) que procesa datos de gráficos y genera las imágenes (por ejemplo, fotogramas de vídeo, imágenes fijas) mostradas en el panel LCD 1150. Por otra parte, 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 OPENGL u otra arquitectura adecuada. Adicionalmente, el procesador 1110 puede estar conectado a un procesador de anfitrión (por ejemplo, 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 utiliza 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. Adicionalmente, 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 utilizar 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 (por ejemplo, 25 %, 50 %) normalmente hacen que los elementos emisores de luz se enciendan durante una porción 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. Por otra parte, 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.

- 5 Como alternativa, se puede utilizar un sistema de visualización emisivo en el que el panel LCD 1150 sería sustituido por un panel de visualización emisivo (por ejemplo, diodo emisor de luz orgánico/OLED), se omitiría la retroiluminación 1160 y los controladores de fila y columna 1152 y 1154, respectivamente, se podrían utilizar para modular directamente el color y el brillo de los píxeles.

10 **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 (por ejemplo, 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 (por ejemplo, controlador integrado, microcontrolador, procesador de anfitrión, procesador gráfico) acoplado a una memoria 1222, que puede incluir ubicaciones de almacenamiento que son direccionables a través del procesador 1224. Como se analizará 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. Por otra parte, la configuración de resolución de visualización 1216 puede definirse en términos de unidad de distancia o área (por ejemplo, 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, introducir 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 utiliza 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 utilizar 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 (por ejemplo, fotogramas de vídeo, imágenes fijas) al sistema 1300 mediante 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) utilizada, 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 utilizar 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 utilizar para generar una presentación de pantalla ancha (por ejemplo, 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 (por ejemplo, cristal líquido, matriz activa, plasma, etc.), HMD, dispositivos de proyección de vídeo, etc., que se coordinan entre sí para producir la presentación 1440. Por otra parte, 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 como 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. Adicionalmente, el módulo de monitorización de señales 1432 puede monitorizar 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. Por otra parte, el módulo de monitorización de señales 1432 puede detectar automáticamente cambios en el conjunto de unidades de visualización 1430 según 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 admiten 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 (por ejemplo, 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 gráfico a múltiples clientes simultáneamente (por ejemplo, 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 gráfico (por ejemplo, GPU) 1530, un procesador de anfitrión (por ejemplo, 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 gráfico se recupere de la memoria mediante 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 gráfico 1530. Los comandos de representación pueden hacer que el procesador gráfico 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 gráfico 1530 ilustrado incluye un renderizador de gráficos 1532 que ejecuta procedimientos de renderizado según los comandos de renderizado generados por el procesador de anfitrión 1524. La salida del procesador gráfico 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 utilizar 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), el estándar de codificación de vídeo avanzada (AVC) MPEG4 de la Organización Internacional para el 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 ha señalado, 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 mediante 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, según una realización. En diversas realizaciones, el sistema 1600 incluye uno o más procesadores 1602 y uno o más procesadores gráficos 1608, y puede ser un sistema de escritorio de procesador único, un sistema de estación de trabajo 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 utilización en dispositivos móviles, portátiles o integrados.

Una realización del sistema 1600 puede incluir o incorporarse dentro de una plataforma de juegos basada en servidor, una consola de juegos, incluyendo 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, teléfono inteligente, dispositivo informático de tableta o dispositivo de Internet móvil. El sistema de procesamiento de datos 1600 también puede incluir, acoplarse o estar integrado dentro de un dispositivo portátil, tal como un dispositivo ponible tipo 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 decodificador que tiene uno o más procesadores 1602 y una interfaz gráfica generada por uno o más procesadores gráficos 1608.

En algunas realizaciones, cada uno del uno o más procesadores 1602 incluye uno o más núcleos de procesador 1607 para procesar instrucciones que, cuando son ejecutadas, realizan operaciones para software de usuario y sistema. En algunas realizaciones, cada uno del 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 la computación de conjunto de instrucciones complejas (CISC), la computación de conjunto de instrucciones reducidas (RISC) o la computación mediante una palabra de instrucción muy larga (VLIW). Múltiples núcleos de procesador 1607 pueden procesar cada uno un conjunto de instrucciones 1609 diferente, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo del 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 está compartida entre varios componentes del procesador 1602. En algunas realizaciones, el procesador 1602 también utiliza una memoria caché externa (por ejemplo, una memoria caché de nivel 3 (L3) o una memoria caché de último nivel (LLC)) (no se muestra), que puede compartirse entre los núcleos de procesador 1607 utilizando técnicas de coherencia de caché conocidas. Adicionalmente, se incluye un archivo de registros 1606 en el procesador 1602, que puede incluir diferentes tipos de registros para almacenar diferentes tipos de datos (por ejemplo, registros de números enteros, registros de coma flotante, registros de estado y un registro de puntero a instrucción). Algunos registros pueden ser registros de propósito general, mientras que otros registros pueden ser específicos para el diseño del procesador 1602.

En algunas realizaciones, el procesador 1602 está acoplado a 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 del sistema 1600. En una realización, el sistema 1600 utiliza una arquitectura de sistema de 'concentrador' ilustrativa, incluyendo un concentrador de controladores de memoria 1616 y un concentrador de controladores de entrada/salida (E/S) 1630. Un concentrador de controladores de memoria 1616 facilita la comunicación entre un dispositivo de memoria y otros componentes del sistema 1600, mientras que un concentrador de controladores de E/S (ICH) 1630 proporciona conexiones a dispositivos de E/S por medio de un bus de E/S local. En una realización, la lógica del concentrador de controladores de memoria 1616 está integrada dentro del procesador.

El dispositivo de memoria 1620 puede ser un dispositivo de memoria de acceso aleatorio dinámico (DRAM), un dispositivo de memoria de acceso aleatorio estático (SRAM), un dispositivo de memoria flash, un dispositivo de memoria de cambio de fase o algún otro dispositivo de memoria que tenga un rendimiento adecuado para servir como memoria de proceso. En una realización, el dispositivo de memoria 1620 puede funcionar como memoria del sistema para el sistema 1600, para almacenar datos 1622 e instrucciones 1621 para utilizarse cuando uno o más procesadores 1602 ejecutan una aplicación o proceso. El concentrador de controladores de memoria 1616 también está acoplado a un procesador gráfico externo opcional 1612, que puede comunicarse con los uno o más procesadores gráficos 1608 de los procesadores 1602 para realizar operaciones de gráficos y de medios.

En algunas realizaciones, el ICH 1630 permite que dispositivos periféricos se conecten al dispositivo de memoria 1620 y al procesador 1602 por medio de un bus de E/S de alta velocidad. Los dispositivos periféricos de E/S incluyen, aunque no de forma limitativa, un controlador de audio 1646, una interfaz de firmware 1628, un transceptor inalámbrico 1626 (por ejemplo, Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1624 (por ejemplo, unidad de disco duro, memoria flash, etc.) y un controlador de E/S heredado 1640 para acoplar dispositivos heredados (por ejemplo, un sistema personal 2 (PS/2)) al sistema. Uno o varios controladores 1642 de bus serie universal (USB) conectan dispositivos de entrada, tales como combinaciones de teclado y ratón 1644. Un controlador de red 1634 también puede acoplarse al 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 se

pueden utilizar otros tipos de sistemas de procesamiento de datos que están configurados de manera diferente. Por ejemplo, el concentrador del controlador de E/S 1630 puede estar integrado dentro del uno o más procesadores 1602, o el concentrador del controlador de memoria 1616 y el concentrador del controlador de E/S 1630 pueden estar integrados en un procesador gráfico externo discreto, tal como el procesador gráfico 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 gráfico integrado 1708. Aquellos elementos de la Figura 17 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra Figura del presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, aunque no de forma limitativa. 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 memoria caché de instrucciones y datos dentro de cada núcleo de procesador y uno o más niveles de memoria caché de nivel medio compartida, tal como un Nivel 2 (L2), Nivel 3 (L3), Nivel 4 (L4) u otros niveles de caché, donde el nivel más alto de memoria caché antes de la memoria externa se clasifica como LLC. En algunas realizaciones, la lógica de coherencia de memoria caché mantiene la coherencia entre las diversas unidades de memoria 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. Las una o más unidades de controlador de bus 1716 gestionan un conjunto de buses de periféricos, tales como uno o más buses de interconexión de componentes periféricos (por ejemplo, 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 varios dispositivos de memoria externa (no se muestran).

En algunas realizaciones, uno o varios de los núcleos de procesador 1702A a 1702N incluyen la admisión de múltiples hilos simultáneos. En una realización de este tipo, el núcleo de agente de sistema 1710 incluye componentes para coordinar y hacer funcionar 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 energía (PCU), que incluye una lógica y componentes para regular el estado de energía de los núcleos de procesador 1702A-1702N y el procesador gráfico 1708.

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

En algunas formas de realización, se utiliza una unidad de interconexión basada en anillo 1712 para acoplar los componentes internos del procesador 1700. Sin embargo, se puede utilizar una unidad de interconexión alternativa, tal como una interconexión de punto a punto, una interconexión conmutada u otras técnicas, incluyendo técnicas ampliamente conocidas en la técnica. En algunas realizaciones, el procesador gráfico 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, incluyendo una interconexión de E/S de paquete, que facilita la comunicación entre diversos componentes de procesador y un módulo de memoria embebida 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 gráfico 1708 utiliza módulos de memoria embebida 1718, tal como una memoria caché compartida de último nivel.

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.

5 La Figura 18 es un diagrama de bloques de un procesador gráfico 1800, que puede ser una unidad de procesamiento de gráficos discreta o puede ser un procesador gráfico integrado con una pluralidad de núcleos de procesamiento. En algunas realizaciones, el procesador gráfico se comunica mediante una interfaz de E/S de memoria asignada a registros en el procesador gráfico y con comandos colocados en la memoria de procesador. En algunas realizaciones, el procesador gráfico 1800 incluye una interfaz de memoria 1814 para acceder a la memoria. La interfaz de memoria
10 1814 puede ser una interfaz con la memoria local, una o más memorias caché internas, una o más memorias caché externas compartidas y/o con la memoria del sistema.

En algunas realizaciones, el procesador gráfico 1800 también incluye un controlador de visualización 1802 para dirigir 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 gráfico 1800 incluye un motor de códec de vídeo 1806 para codificar, decodificar o transcodificar medios hacia, desde o entre uno o más formatos de codificación de medios, incluyendo, aunque no de forma limitativa, los formatos del Grupo de expertos en imágenes en movimiento (MPEG) tales como MPEG-2, los formatos de codificación de vídeo avanzada (AVC) tales como H.264/MPEG-4 AVC, así como los formatos de la Sociedad de ingenieros de imágenes en movimiento y televisión (SMPTE) 421M/VC-1 y formatos del Grupo conjunto de expertos en fotografía (JPEG) tales como JPEG y Motion JPEG (MJPEG).

En algunas realizaciones, el procesador gráfico 1800 incluye un motor de transferencia de imágenes en bloques (BLIT) 1804 para realizar operaciones de rasterización bidimensionales (2D) incluyendo, por ejemplo, transferencias de bloques con límites de bits. Sin embargo, en una realización, las operaciones de gráficos 2D se realizan utilizando uno o varios 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 gráficas, incluyendo operaciones gráficas tridimensionales (3D) y operaciones multimedia.

En algunas realizaciones, el GPE 1810 incluye una canalización 3D 1812 para realizar operaciones en 3D, tales como representar imágenes y escenas tridimensionales utilizando funciones de procesamiento que actúan sobre formas de primitivas en 3D (por ejemplo, rectángulo, triángulo, etc.). La canalización 3D 1812 incluye elementos de función programable y fija que realizan diversas tareas dentro del elemento y/o generan hilos de ejecución en un subsistema 3D/de medios 1815. Si bien la canalización 3D 1812 se puede utilizar para realizar operaciones de medios, una realización de GPE 1810 también incluye una canalización de medios 1816 que se utiliza específicamente para realizar operaciones de medios, tales como postprocesamiento de vídeo y mejora de imágenes.

En algunas realizaciones, la canalización de medios 1816 incluye unidades de lógica programable o de función fija para realizar una o más operaciones de medios especializadas, tales como aceleración de descodificación de vídeo, desentrelazado de vídeo y aceleración de codificación de vídeo en lugar o en nombre 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 varias 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 los 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 multimedia. En algunas realizaciones, el subsistema 3D/de medios 1815 incluye una o más memorias caché internas para instrucciones de hilos y datos. En algunas realizaciones, el subsistema también incluye una memoria compartida, incluyendo registros y una 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 gráfico 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 del presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte del presente documento, pero no se limitan a tal cosa.

En algunas realizaciones, el GPE 1910 se acopla con un transmisor por flujo continuo de comandos 1903, que proporciona un flujo de comandos al GPE 3D y a las canalizaciones de medios 1912, 1916. En algunas realizaciones, el transmisor por flujo continuo de comandos 1903 está acoplado a la memoria, que puede ser una memoria de sistema

o una o más de memoria de caché interna y memoria de caché compartida. En algunas realizaciones, el transmisor 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 mediante 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 (por ejemplo, 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 una 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 (por ejemplo, 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 gráfico utiliza el motor de estimación de movimiento de vídeo 1934 para realizar operaciones en vídeo a nivel de macrobloque que, de lo contrario, podrían 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 gráfico 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 la memoria. En algunas realizaciones, el puerto de datos 1944 facilita el acceso a la memoria para operaciones que incluyen escrituras de objetivo de representación gráfica, 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 (por ejemplo, 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 mediante 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 gráfico 2000. Los elementos de la Figura 20 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra Figura del presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte del presente documento, pero no se limitan a tal cosa.

En algunas realizaciones, el procesador gráfico 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 gráfico a otras unidades de procesamiento, incluyendo otros

procesadores gráficos o uno o varios núcleos de procesador de propósito general. En algunas realizaciones, el procesador gráfico es uno de muchos procesadores integrados dentro de un sistema de procesamiento de múltiples núcleos.

5 En algunas realizaciones, el procesador gráfico 2000 recibe lotes de comandos mediante la interconexión en anillo 2002. Los comandos entrantes son interpretados por un transmisor en continuo de comandos 2003 en el extremo frontal de la canalización 2004. En algunas realizaciones, el procesador gráfico 2000 incluye una lógica de ejecución escalable para realizar procesamiento de geometría 3D y procesamiento de medios a través del núcleo o núcleos de gráficos 2080A-2080N. Para los comandos de procesamiento de geometría 3D, el transmisor en 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 al extremo frontal de vídeo 2034, que se acopla a 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 e imágenes 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, tanto la canalización de geometría 2036 como el motor de medios 2037 generan hilos de ejecución para los recursos de ejecución de hilos proporcionados por al menos un núcleo de gráficos 2080A.

20 En algunas realizaciones, el procesador gráfico 2000 incluye recursos de ejecución de hilos escalables que presentan 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 gráfico 2000 puede tener cualquier número de núcleos de gráficos 2080A a 2080N. En algunas realizaciones, el procesador gráfico 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 gráfico es un procesador de baja potencia con un único subnúcleo (por ejemplo, 2050A). En algunas realizaciones, el procesador gráfico 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/textura 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 caché compartida y lógica de operación de píxeles. También se pueden incluir otros recursos compartidos en las diversas realizaciones del procesador gráfico.

35 La Figura 21 ilustra una 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 del presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte del presente documento, pero no se limitan a tal cosa.

40 En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un sombreador de píxeles 2102, un despachador de hilos 2104, una memoria 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 memoria caché de datos 2112 y un puerto de datos 2114. En una realización, los componentes incluidos están interconectados mediante una estructura 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 (por ejemplo, 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.

55 En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N se utiliza 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 soporte nativo para muchas instrucciones de sombreador de gráficos 3D convencional, de tal manera que los programas de sombreador de las bibliotecas de gráficos (por ejemplo, Direct 3D y OpenGL) se ejecuten con una conversión mínima. Las unidades de ejecución admiten procesamiento de vértices y geometría (por ejemplo, programas de vértices, programas de geometría, sombreadores de vértices), procesamiento de píxeles (por ejemplo, sombreadores de píxeles, sombreadores de fragmentos) y procesamiento de propósito general (por ejemplo, sombreadores de computación y medios).

60 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 a elementos de datos, enmascaramiento y control de flujo dentro de las instrucciones. El número de canales puede ser independiente del número de unidades

65

aritméticológicas (ALU) o unidades de coma flotante (FPU) físicas de un procesador gráfico en particular. En algunas realizaciones, las unidades de ejecución 2108A-2108N admiten tipos de datos de números enteros y de coma flotante.

5 El conjunto de instrucciones de la unidad de ejecución incluye instrucciones de datos múltiples de una sola instrucción (SIMD). Los diversos elementos de datos se pueden almacenar como un tipo de datos encapsulado en un registro y la unidad de ejecución procesará los diversos elementos en función del tamaño de los datos de los elementos. Por ejemplo, cuando se realizan operaciones con un vector de 256 bits de ancho, los 256 bits del vector se almacenan en un registro y la unidad de ejecución realiza operaciones con 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 octeto (B)). Sin embargo, son posibles diferentes anchos de vector y tamaños de registro.

15 Una o varias memorias caché de instrucciones internas (por ejemplo, 2106) se incluyen en la lógica de ejecución de hilos 2100 para almacenar en la caché instrucciones de hilos para las unidades de ejecución. En algunas realizaciones, una o más cachés de datos (por ejemplo, 2112) están incluidas en datos de hilo de caché 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.

25 Durante la ejecución, las canalizaciones de gráficos y medios envían solicitudes de iniciación de hilos a la lógica de ejecución de hilos 2100 mediante la lógica de generación y 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 (por ejemplo, 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 también puede procesar solicitudes de generación de hilos en tiempo de ejecución desde los programas de sombreado en ejecución.

35 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 (por ejemplo, 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 se van a interpolar a lo largo 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 (por ejemplo, 2108A) mediante el despachador de hilos 2104. En algunas realizaciones, el sombreador de píxeles 2102 utiliza 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 del procesamiento posterior.

45 En algunas realizaciones, el puerto de datos 2114 proporciona un mecanismo de acceso a la memoria para que la lógica de ejecución de hilos 2100 envíe datos procesados a la memoria para su procesamiento en una canalización de salida de procesador gráfico. En algunas realizaciones, el puerto de datos 2114 incluye o se acopla a una o más memorias caché (por ejemplo, la caché de datos 2112) para almacenar en caché datos para un acceso a la memoria por medio del puerto de datos.

50 La Figura 22 es un diagrama de bloques que ilustra unos formatos de instrucciones de procesador gráfico 2200 según algunas realizaciones. En una o más realizaciones, las unidades de ejecución de procesador gráfico admiten un conjunto de instrucciones que tiene instrucciones en múltiples formatos. Los recuadros de líneas continuas ilustran los componentes que se incluyen generalmente en una instrucción de unidad de ejecución, mientras que las líneas discontinuas incluyen componentes que son opcionales o que sólo se incluyen en un subconjunto de las instrucciones. En algunas realizaciones, el formato de instrucción 2200 descrito e ilustrado son macroinstrucciones, en el sentido de que son instrucciones suministradas a la unidad de ejecución, en oposición a microoperaciones resultantes de la decodificación de instrucciones una vez que se procesa la instrucción.

60 En algunas realizaciones, las unidades de ejecución de procesador gráfico admiten de manera nativa instrucciones en un formato de 128 bits 2210. Un formato de instrucción compacta de 64 bits 2230 está disponible para algunas instrucciones basándose en la instrucción, las opciones de instrucción y el número de operandos seleccionados. 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 según la realización. En algunas realizaciones, la instrucción se compacta en parte utilizando 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 en función de los valores de índice y utiliza las salidas de tabla de compactación para reconstruir una instrucción nativa en el formato de 128 bits 2210.

5 Para cada formato, el código de operación de instrucción 2212 define la operación que debe realizar la unidad de ejecución. Las unidades de ejecución ejecutan cada instrucción en paralelo a lo través 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 en cada canal de color que representa un elemento de textura o un 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 instrucciones 2214 permite controlar determinadas opciones de ejecución, tales como selección de canales (por ejemplo, predicación) y orden de canales de datos (por ejemplo, reorganización de vectores (*swizzle*)). 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 utilización en el formato de instrucción compacto de 64 bits 2230.

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

20 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 utiliza el modo de direccionamiento de registro directo o el modo de direccionamiento de registro indirecto. Cuando se utiliza el modo de direccionamiento de registro directo, la dirección de registro de uno o más operandos viene proporcionada directamente por bits en la instrucción 2210.

25 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 es para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones admiten 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 utilizar direccionamiento alineado de bytes para los operandos de origen y destino y cuando está en un segundo modo, la instrucción 2210 puede utilizar direccionamiento alineado de 16 bytes para todos los operandos de origen y destino.

30 En una realización, la porción de modo de dirección del campo de modo de acceso/dirección 2226 determina si la instrucción va a utilizar direccionamiento directo o indirecto. Cuando se utiliza 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 utiliza un modo de direccionamiento de registro indirecto, la dirección de registro de uno o más operandos puede calcularse basándose en un valor de registro de dirección y un campo inmediato de dirección en la instrucción.

35 En algunas realizaciones, las instrucciones se agrupan basándose en los campos de bits del código de operación 2212 para simplificar la decodificación de 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 precisa del código de operación que se muestra es simplemente un ejemplo. En algunas realizaciones, un grupo de código de operación de movimiento y lógica 2242 incluye instrucciones lógicas y de movimiento de datos (por ejemplo, mover (*mov*), comparar (*cmp*)). En algunas realizaciones, el grupo de movimiento y lógica 2242 comparte los cinco bits más significativos (MSB), donde las instrucciones de movimiento (*mov*) tienen la forma de 0000xxxxb y las instrucciones lógicas tienen la forma de 0001xxxxb. Un grupo de instrucciones de control de flujo 2244 (por ejemplo, llamar, saltar (*jmp*)) incluye instrucciones en la forma de 0010xxxxb (por ejemplo, 0x20). Un grupo de instrucciones misceláneas 2246 incluye una mezcla de instrucciones, incluyendo instrucciones de sincronización (por ejemplo, esperar, enviar) en forma de 0011xxxxb (por ejemplo, 0x30). Un grupo de instrucciones matemáticas en paralelo 2248 incluye instrucciones aritméticas a nivel de componente (por ejemplo, sumar, multiplicar (*mul*)) en forma de 0100xxxxb (por ejemplo, 0x40). El grupo de cálculo matemático paralelo 2248 realiza las operaciones aritméticas en paralelo a través de los canales de datos. El grupo de cálculo matemático vectorial 2250 incluye instrucciones aritméticas (por ejemplo, dp4) en forma de 0101xxxxb (por ejemplo, 0x50). El grupo de cálculo matemático vectorial realiza operaciones aritméticas, tales como cálculos de producto escalar en operandos vectoriales.

60 **Canalización de gráficos**

65 La Figura 23 es un diagrama de bloques de otra realización de un procesador gráfico 2300. Los elementos de la Figura 23 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra Figura del presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte del presente documento, pero no se limitan a tal cosa.

En algunas realizaciones, el procesador gráfico 2300 incluye una canalización de gráficos 2320, una canalización de medios 2330, un motor de visualización 2340, una lógica de ejecución de hilos 2350 y una canalización de salida de representación 2370. En algunas realizaciones, el procesador gráfico 2300 es un procesador gráfico dentro de un sistema de procesamiento de múltiples núcleos que incluye uno o más núcleos de procesamiento de propósito general.

5 El procesador gráfico está controlado por escrituras de registro en uno o varios registros de control (no mostrados) o mediante comandos emitidos al procesador gráfico 2300 mediante una interconexión en anillo 2302. En algunas realizaciones, la interconexión en anillo 2302 acopla el procesador gráfico 2300 a otros componentes de procesamiento, tales como otros procesadores gráficos o procesadores de propósito general. Los comandos desde la interconexión en anillo 2302 son interpretados por un transmisor por flujo continuo de comandos 2303, que suministra instrucciones a componentes individuales de la canalización de gráficos 2320 o la canalización de medios 2330.

10 En algunas realizaciones, el transmisor en 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 de espacio de coordenadas y de iluminación para 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.

15 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 memoria caché L1, 2351, conectada que es específica para cada matriz o compartida entre las matrices. La memoria caché se puede configurar como una memoria caché de datos, una memoria caché de instrucciones o una memoria caché única subdividida para contener datos e instrucciones en diferentes particiones.

20 En algunas realizaciones, la canalización de gráficos 2320 incluye componentes de teselación para realizar una teselación acelerada por hardware de objetos 3D. En algunas realizaciones, un sombreador de casco programable 2311 configura las operaciones de teselación. Un sombreador de dominio 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 propósito 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 utiliza la teselación, se pueden omitir los componentes de teselación 2311, 2313, 2317.

25 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 sobre objetos geométricos enteros, en lugar de vértices o parches de vértices como en fases previas de la canalización de gráficos. Si la teselación está inhabilitada, el sombreador de geometría 2319 recibe una entrada desde el sombreador de vértices 2307. En algunas realizaciones, el sombreador de geometría 2319 se puede programar mediante un programa de sombreado de geometría para realizar la teselación de geometría si las unidades de teselación están inhabilitadas.

30 Antes de la rasterización, un recortador 2329 procesa los datos de vértices. El recortador 2329 puede ser un recortador de función fija o un recortador programable que tenga funciones de recorte y sombreador de geometría. En algunas realizaciones, un rasterizador 2373 (por ejemplo, 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.

35 El procesador gráfico 2300 tiene un bus de interconexión, una estructura de interconexión o algún otro mecanismo de interconexión que permite el paso de datos y mensajes entre los componentes principales del procesador. En algunas realizaciones, las unidades de ejecución 2352A, 2352B y la memoria caché o memorias caché asociadas 2351, el muestreador de textura y de medios 2354 y la memoria 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, cachés 2351, 2358 y unidades de ejecución 2352A, 2352B tienen rutas de acceso a memoria separadas.

40 En algunas realizaciones, la canalización de salida de representación 2370 contiene un rasterizador 2373 que convierte objetos basados en vértices en una representación asociada basada en píxeles. En algunas realizaciones, la lógica del rasterizador incluye una unidad generadora de ventanas/enmascaradora para realizar una rasterización de triángulo y de línea de función fija. En algunas realizaciones también están disponibles una caché de representación 2378 y una caché de profundidad 2379 asociadas. Un componente de operaciones de píxeles 2377 realiza operaciones basadas en píxeles sobre los datos, aunque, en algunas instancias, las operaciones de píxeles asociadas con operaciones 2D (por ejemplo, transferencias de imagen de bloque de bits con mezcla) se realizan por el motor 2D

45

50

55

60

65

2341, o son sustituidas en el momento de la visualización por el controlador de visualización 2343 utilizando planos de visualización de superposición. En algunas realizaciones, una memoria caché L3 compartida 2375 está disponible para todos los componentes gráficos, permitiendo compartir datos sin utilizar la memoria principal del sistema.

5 En algunas realizaciones, la canalización de medios 2330 del procesador gráfico 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 transmisor en continuo de comandos 2303. En algunas realizaciones, la canalización de medios 2330 incluye un transmisor en continuo de comandos independiente. En algunas realizaciones, el extremo frontal de vídeo 2334 procesa los comandos de medios antes de enviar el comando al motor de medios 2337. En algunas
10 realizaciones, el motor de medios 2337 incluye una funcionalidad de generación de hilos para generar hilos para su despacho a la lógica de ejecución de hilos 2350 a través del distribuidor 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 está acoplado al procesador gráfico mediante la interconexión en anillo 2302 o algún otro bus o estructura 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 una lógica de propósito especial capaz de funcionar 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 el sistema, tal como en un ordenador portátil, o un dispositivo de visualización externo conectado mediante un conector de dispositivo de visualización.
15
20

En algunas realizaciones, la canalización de gráficos 2320 y la canalización de medios 2330 se pueden configurar para realizar operaciones basadas en múltiples interfaces de programación de gráficos y medios y no son específicas de ninguna interfaz de programación de aplicaciones (API). En algunas realizaciones, el software de controlador para el procesador gráfico traduce llamadas de API que son específicas de una biblioteca de medios o de gráficos particular a comandos que pueden procesarse por el procesador gráfico. En algunas realizaciones, se proporciona compatibilidad con la biblioteca Open graphics (OpenGL) y Open computing language (OpenCL) del grupo Khronos, la biblioteca Direct3D de Microsoft Corporation, o se puede proporcionar soporte tanto para OpenGL como para D3D. También puede facilitarse compatibilidad con la biblioteca de visión por ordenador de código abierto (OpenCV). También se admitiría 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 gráfico.
25
30

Programación de canalización de gráficos

35 La Figura 24A es un diagrama de bloques que ilustra un formato de comandos de procesador gráfico 2400 según algunas realizaciones. La Figura 24B es un diagrama de bloques que ilustra una secuencia de comandos de procesador gráfico 2410 según una realización. Los recuadros en líneas continuas de la Figura 24A ilustran los componentes que están incluidos por lo general en un comando de gráficos, mientras que las líneas discontinuas incluyen componentes que son opcionales o que están incluidos únicamente en un subconjunto de los comandos de gráficos. El formato de comandos de procesador gráfico 2400 ilustrativo de la Figura 24A incluye campos de datos para identificar un cliente objetivo 2402 del comando, un código de operación del comando (cód. de ope.) 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.
40
45

En algunas realizaciones, el cliente 2402 especifica la unidad cliente del dispositivo de gráficos que procesa los datos de comando. En algunas realizaciones, un analizador de comandos de procesador gráfico examina el campo cliente de cada comando para condicionar el procesamiento posterior del comando y enrutar los datos del comando a la unidad cliente apropiada. En algunas realizaciones, las unidades cliente del procesador gráfico incluyen una unidad de interfaz de memoria, una unidad de renderizado, una unidad 2D, una unidad 3D y una unidad multimedia. Cada unidad cliente tiene una canalización de procesamiento correspondiente que procesa los comandos. Una vez que el comando es recibido por la unidad cliente, la unidad cliente lee el código de operación 2404 y, si está presente, el subcódigo de operación 2405 para determinar la operación a realizar. La unidad cliente ejecuta el comando utilizando la información del campo de datos 2406. Para algunos comandos se espera un tamaño de comando explícito 2408 para especificar 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.
50
55

El diagrama de flujo de la Figura 24B muestra una secuencia de comandos de procesador gráfico 2410 ilustrativa. En algunas realizaciones, el software o firmware de un sistema de procesamiento de datos que presenta una realización de un procesador gráfico utiliza 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 únicamente con fines ilustrativos, ya que las realizaciones no están limitadas a estos comandos específicos o para esta secuencia de comandos. Por otra parte, los comandos se pueden emitir como un lote de comandos en una secuencia de comandos, de tal forma que el procesador gráfico procesará la secuencia de comandos al menos parcialmente en concurrencia.
60
65

En algunas realizaciones, la secuencia de comandos del 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 de 3D 2422 y la canalización de medios 2424 no operan al mismo tiempo. El vaciado de la canalización se lleva a cabo para hacer que la canalización de gráficos activo complete cualesquiera comandos pendientes. En respuesta a un vaciado de canalización, el analizador de comandos para el procesador gráfico pausará el procesamiento de comandos hasta que los motores de dibujo activos completen las operaciones pendientes y se invaliden las memorias caché de lectura relevantes. Opcionalmente, cualquier dato en la memoria caché de renderizado que esté marcado como "sucio" se puede descargar a la memoria. En algunas realizaciones, el comando de vaciado de canalización 2412 se puede utilizar para la sincronización de canalización o antes de colocar el procesador gráfico en un estado de baja potencia.

En algunas realizaciones, se utiliza un comando de selección de canalización 2413 cuando una secuencia de comandos requiere que el procesador gráfico cambie explícitamente entre canalizaciones. En algunas realizaciones, un comando de selección de canalización 2413 se requiere sólo una vez dentro de un contexto de ejecución antes de emitir comandos de canalización a menos que el contexto sea para emitir comandos para ambas canalizaciones. En algunas realizaciones, se requiere un comando de vaciado de canalización 2412 inmediatamente antes de una conmutación 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 utiliza 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 la canalización para la canalización activa. En una realización, el comando de control de canalización 2414 se utiliza para la sincronización de la canalización y para borrar datos de una o más memorias caché dentro de la canalización activa antes de procesar un lote de comandos.

En algunas realizaciones, los comandos de estado de la memoria intermedia de retorno 2416 se utilizan 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 varias memorias intermedias de retorno en las que las operaciones escriben datos intermedios durante el procesamiento. En algunas realizaciones, el procesador gráfico también utiliza una o más memorias intermedias de retorno para almacenar datos de salida y para realizar una comunicación entre hilos. En algunas realizaciones, el estado de memoria intermedia de retorno 2416 incluye seleccionar el tamaño y número de memorias intermedias de retorno que utilizar para un conjunto de operaciones de canalización.

Los comandos restantes de la secuencia de comandos difieren en función de la canalización activa para las operaciones. Basándose en una determinación de canalización 2420, la secuencia de comandos se adapta a la canalización 3D 2422 comenzando con el estado de canalización 3D 2430, o a la canalización de medios 2424 comenzando en el estado de canalización de medios 2440.

Los comandos para el estado de canalización 3D 2430 incluyen comandos de ajuste de estado 3D para 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 utilización. En algunas realizaciones, los comandos de estado de canalización 3D 2430 también pueden inhabilitar o eludir selectivamente ciertos elementos de la canalización si esos elementos no se utilizarán.

En algunas realizaciones, el comando de primitiva 3D 2432 se utiliza para enviar primitivas 3D para que sean procesadas por la canalización 3D. Los comandos y parámetros asociados que se pasan al procesador gráfico a través del comando de primitivas 3D 2432 se reenvían a la función de extracción de vértices en la canalización de gráficos. La función de extracción de vértices utiliza los datos del 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, el comando de primitivas 3D 2432 se utiliza para realizar operaciones de vértices en primitivas 3D por medio de sombreadores de vértices. Para procesar sombreadores de vértices, la canalización 3D 2422 despacha hilos de ejecución de sombreadores a unidades de ejecución de procesador gráfico.

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 la ejecución de un comando. En algunas realizaciones, la ejecución se desencadena mediante un comando "ir" o "patear" en la secuencia de comandos. En una realización, la ejecución del comando se desencadena utilizando 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 se han completado las operaciones, los objetos geométricos resultantes se rasterizan y el motor de píxeles da color a los píxeles resultantes. También pueden incluirse comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo trasero de píxeles para esas operaciones.

En algunas realizaciones, la secuencia de comandos de procesador gráfico 2410 sigue la ruta de canalización de medios 2424 cuando se realizan operaciones de medios. En general, la utilización específica y la forma de programación de la canalización de medios 2424 depende de las operaciones de medios o de cálculo a realizar. Se pueden descargar operaciones de decodificación de medios específicas a la canalización de medios durante la decodificación de medios. En algunas realizaciones, la canalización de medios también se puede omitir y la decodificación de medios se puede realizar en su totalidad o en parte utilizando recursos proporcionados por uno o varios núcleos de procesamiento de propósito general. En una realización, la canalización de medios también incluye elementos para operaciones de unidad de procesador gráfico de propósito general (GPGPU), donde el procesador gráfico se utiliza para realizar operaciones vectoriales SIMD utilizando programas de sombreado de cálculo que no están explícitamente relacionados con el renderizado de primitivas de gráficos.

En algunas realizaciones, la canalización de medios 2424 se configura de forma similar a 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 descodificación de vídeo y de codificación de vídeo en la canalización de medios, tal como el formato de codificación o de descodificación. En algunas realizaciones, los comandos de estado de canalización de medios 2440 también admiten la utilización 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 para procesar. En algunas realizaciones, todos los estados de canalización de medios deben ser válidos antes de emitir un comando de objeto de medios 2442. Una vez que se configura el estado de la canalización y se ponen en cola los comandos de objeto de medios 2442, la canalización de medios 2424 se desencadena mediante un comando de ejecución 2444 o un evento de ejecución equivalente (por ejemplo, escritura de registro). La salida de la canalización de medios 2424 se puede procesar posteriormente mediante operaciones proporcionadas por la canalización 3D 2422 o la canalización de medios 2424. En algunas realizaciones, las operaciones de GPGPU se configuran y ejecutan de manera similar a 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 según 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 gráfico 2532 y uno o más núcleos de procesador de propósito general 2534. Cada uno de la aplicación de gráficos 2510 y el sistema operativo 2520 se ejecutan 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 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 de OpenGL (GLSL). La aplicación incluye también instrucciones ejecutables 2514 en un lenguaje máquina adecuado para su ejecución por el núcleo de procesador de propósito general 2534. La aplicación también incluye objetos gráficos 2516 definidos por datos de vértices.

En algunas realizaciones, el sistema operativo 2520 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo tipo UNIX propietario o un sistema operativo tipo UNIX de código abierto que utiliza una variante del núcleo de Linux. Cuando se utiliza la API Direct3D, el sistema operativo 2520 utiliza un compilador de sombreador de interfaz 2524 para compilar cualquier instrucción de sombreador 2512 en HLSL en un lenguaje de sombreador de nivel inferior. La compilación puede ser una compilación de tipo justo a tiempo (JIT) o la aplicación puede realizar una compilación previa 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 en modo de usuario 2526 contiene un compilador de sombreado de extremo trasero 2527 para convertir las instrucciones de sombreado 2512 en una representación específica de hardware. Cuando se utiliza la API de OpenGL, las instrucciones de sombreado 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 utiliza 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 gráfico 2532 para despachar comandos e instrucciones.

Implementaciones de núcleos IP

Uno o más aspectos de al menos una realización se pueden implementar mediante un código representativo almacenado en un medio legible por máquina que representa y/o define la 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 genere la lógica para realizar las técnicas descritas en el presente documento. Dichas representaciones, conocidas como "núcleos IP", son unidades de lógica reutilizables para un circuito integrado que pueden almacenarse en un medio tangible legible por máquina tal 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 tal manera que el circuito realice 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 se puede utilizar para fabricar un circuito integrado para realizar las operaciones según una realización. El sistema de desarrollo de núcleo de IP 2600 se puede utilizar para generar diseños modulares reutilizables que pueden incorporarse en un diseño más grande o utilizarse para construir un circuito integrado completo (por ejemplo, un circuito integrado 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 (por ejemplo, C/C++). La simulación de software 2610 se puede utilizar para diseñar, probar y verificar el comportamiento del núcleo de IP. A continuación, se puede crear o sintetizar 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, incluyendo la lógica asociada realizada utilizando las señales digitales modeladas. Además de un diseño RTL 2615, también pueden crearse, diseñarse o sintetizarse diseños de nivel inferior a nivel lógico o a nivel de transistor. Por lo tanto, los detalles particulares del diseño inicial y la simulación pueden variar.

El diseño RTL 2615 o equivalente puede ser sintetizado adicionalmente por la instalación de diseño en 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 se puede simular o probar más para verificar el diseño del núcleo IP. El diseño de núcleo de IP puede almacenarse para su entrega a una instalación de fabricación de terceros 2665 utilizando memoria no volátil 2640 (por ejemplo, disco duro, memoria flash o cualquier medio de almacenamiento no volátil). Como alternativa, el diseño de núcleos IP puede transmitirse (por ejemplo, a través de Internet) a través de una conexión por cable 2650 o una conexión inalámbrica 2660. La instalación de fabricación 2665 puede entonces fabricar un circuito integrado que está basado, al menos en parte, en el diseño de núcleos IP. El circuito integrado fabricado se puede configurar para llevar a cabo operaciones de acuerdo con al menos una realización descrita en la presente memoria.

La Figura 27 es un diagrama de bloques que ilustra un circuito integrado de sistema en un chip 2700 ilustrativo que se puede fabricar utilizando uno o más núcleos de IP, según una realización. El circuito integrado ilustrativo incluye uno o más procesadores de aplicación 2705 (por ejemplo, las CPU), al menos un procesador gráfico 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 una lógica de periférico o de bus que incluye un controlador de USB 2725, un 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 lo puede proporcionar un subsistema de memoria flash 2760 que incluye memoria flash y un controlador de memoria flash. La interfaz de memoria se puede proporcionar mediante un controlador de memoria 2765 para acceder a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad integrado 2770.

Adicionalmente, 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 propósito general.

En un ejemplo, el procesador 1110 (Figura 11) envía imágenes HDR codificadas a la pantalla 1120 (Figura 11) como se ha descrito con respecto a las Figuras 6A-10 y en los siguientes Ejemplos.

El término "acoplado" se puede utilizar 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., se pueden utilizar 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 utiliza 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. No obstante, los expertos en la materia comprenderán que pueden introducirse diversas modificaciones y cambios en la misma.

REIVINDICACIONES

1. Un sistema informático que comprende:

5 un transmisor (712) para emitir una imagen de alto rango dinámico, HDR (704);
 una memoria que incluye un conjunto de instrucciones; y
 un procesador, en donde cuando son ejecutadas por el procesador, las instrucciones hacen que el sistema:
 divida (802) la imagen HDR en una pluralidad de regiones que incluyen regiones más brillantes y regiones más
 10 oscuras,
 determine (804), por región, un nivel de luminancia de la imagen HDR, y
 seleccione (806), por región, una cantidad de codificación para cada región de la pluralidad de regiones, incluyendo
 un aumento de la cantidad de codificación si el nivel de luminancia no supera un primer umbral, en donde el aumento
 de la cantidad de codificación es para aumentar la compresión de la imagen, y
 15 una disminución de la cantidad de codificación si el nivel de luminancia supera un segundo umbral, en donde el
 segundo umbral debe ser mayor que el primer umbral, en donde la disminución de la cantidad de codificación es para
 disminuir la compresión de la imagen, en donde las cantidades de codificación de las regiones más brillantes son
 menores que las cantidades de codificación de las regiones más oscuras.

20 2. El sistema de la reivindicación 1, en donde las instrucciones, cuando son ejecutadas, hacen que el sistema
 informático:

inicialice el primer umbral y el segundo umbral.

25 3. El sistema de la reivindicación 1, que incluye además un códec para codificar una primera región de la pluralidad
 de regiones con una primera cantidad de codificación y codificar una segunda región de la pluralidad de regiones con
 una segunda cantidad de codificación.

30 4. El sistema de la reivindicación 1, en donde las instrucciones, cuando son ejecutadas, hacen que un sistema
 informático:

detecte una o más regiones en primer plano de la pluralidad de regiones; y
 disminuya la cantidad de codificación en la una o más regiones en primer plano, en donde la disminución de la cantidad
 de codificación es para disminuir la compresión de la imagen.

35 5. El sistema de la reivindicación 1, en donde las instrucciones, cuando son ejecutadas, hacen que el sistema
 informático:

40 detecte una o más regiones de fondo de la pluralidad de regiones; y
 aumente la cantidad de codificación en la una o más regiones de fondo, en donde el aumento de la cantidad de
 codificación es para aumentar la compresión de la imagen.

6. El sistema de una cualquiera de las reivindicaciones 1 a 5, en donde las instrucciones, cuando son ejecutadas,
 hacen que el sistema informático genere un histograma de luminancia para la imagen HDR.

45 7. Un método, que comprende:

dividir (802) la imagen HDR en una pluralidad de regiones que incluyen regiones más brillantes y regiones más
 oscuras;
 50 determinar (804), por región, un nivel de luminancia de la imagen HDR; y
 seleccionar (806), por región, una cantidad de codificación para cada región de la pluralidad de regiones basándose
 en el nivel de luminancia, incluyendo
 un aumento de la cantidad de codificación si el nivel de luminancia no supera un primer umbral, en donde el aumento
 de la cantidad de codificación es para aumentar la compresión de la imagen, y
 55 una disminución de la cantidad de codificación si el nivel de luminancia supera un segundo umbral, en donde el
 segundo umbral debe ser mayor que el primer umbral, en donde la disminución de la cantidad de codificación es para
 disminuir la compresión de la imagen, en donde las cantidades de codificación de las regiones más brillantes son
 menores que las cantidades de codificación de las regiones más oscuras.

60 8. El método de la reivindicación 7, en donde la selección del nivel de codificación incluye:
 inicializar (822) el primer umbral y el segundo umbral.

9. El método de la reivindicación 7, que comprende, además:

65 codificar una primera región de la pluralidad de regiones con una primera cantidad de codificación; y
 codificar una segunda región de la pluralidad de regiones con una segunda cantidad de codificación.

10. El método de la reivindicación 7, que comprende, además:

5 detectar (842) una o más regiones en primer plano de la pluralidad de regiones; y
disminuir (844) la cantidad de codificación en la una o más regiones en primer plano, en donde la disminución de la
cantidad de codificación disminuye la compresión de la imagen.

11. El método de la reivindicación 7, que comprende, además:

10 detectar (846) uno o más regiones de fondo de la pluralidad de regiones; y
aumentar (848) la cantidad de codificación en una o más regiones de fondo, en donde el aumento de la cantidad de
codificación aumenta la compresión de la imagen.

12. El método de una cualquiera de las reivindicaciones 7 a 11, en donde la determinación del nivel de luminancia
15 incluye generar un histograma de luminancia para la imagen HDR.

13. Al menos un medio de almacenamiento no transitorio legible por ordenador que comprende un conjunto de
instrucciones que, cuando son ejecutadas por un dispositivo informático, hacen que el dispositivo informático lleve a
cabo el método según una cualquiera de las reivindicaciones 7 a 12.

20 14. Un aparato que comprende un procesador (908) y una memoria (916), estando el procesador configurado para
llevar a cabo el método según una cualquiera de las reivindicaciones 7 a 12.

25 15. Un aparato de paquete de semiconductores que comprende un sustrato, y una lógica acoplada al sustrato, en
donde la lógica se implementa en una o más de una lógica configurable o lógica de hardware de funcionalidad fija,
estando la lógica configurada para llevar a cabo el método según una cualquiera de las reivindicaciones 7 a 12.

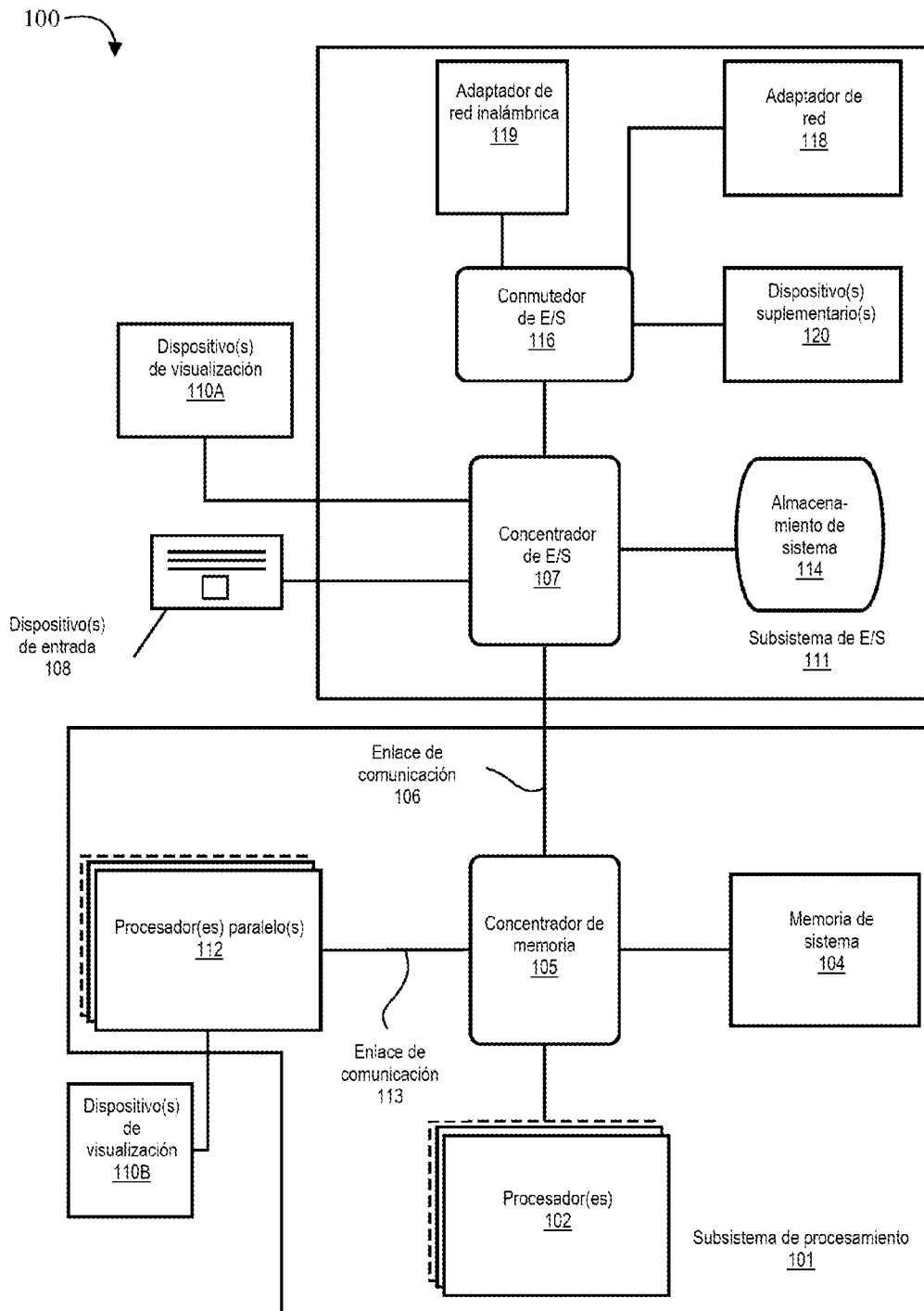


FIG. 1

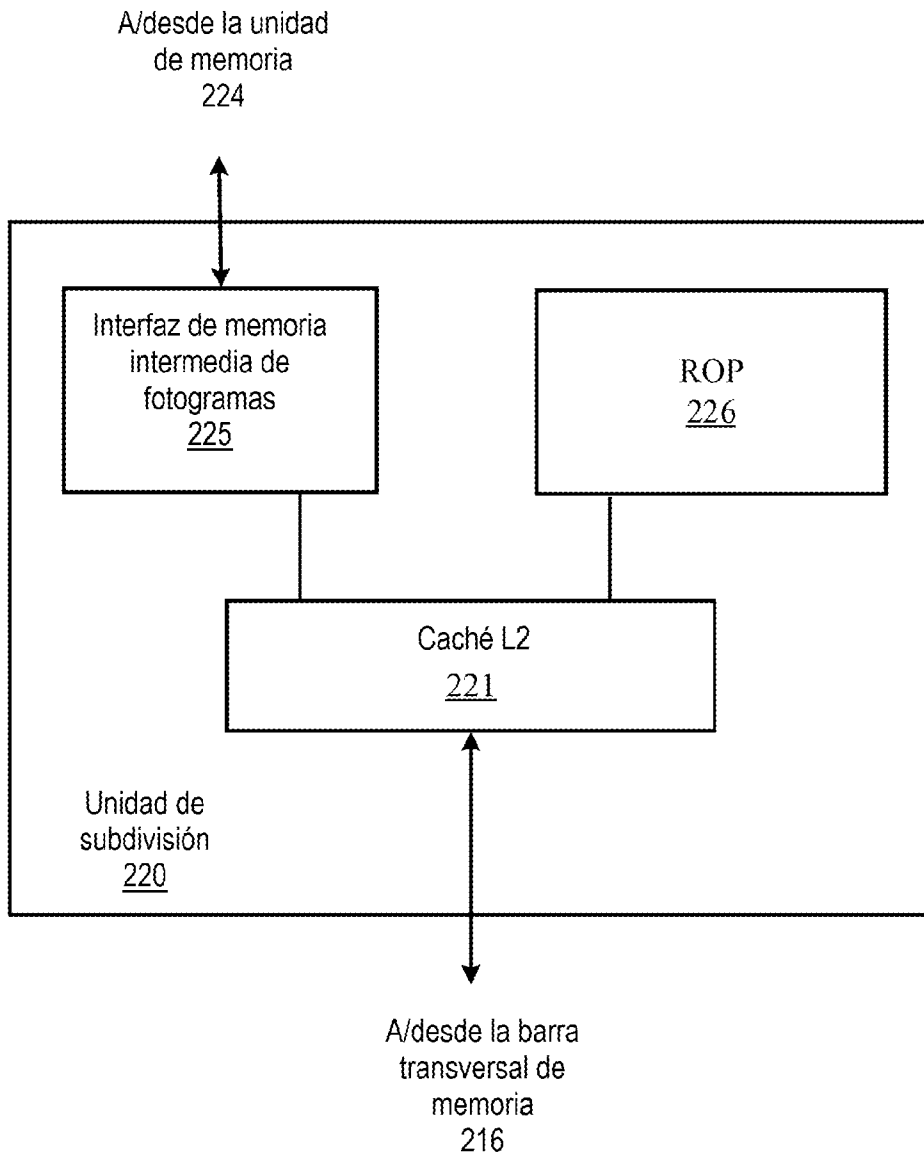


FIG. 2B

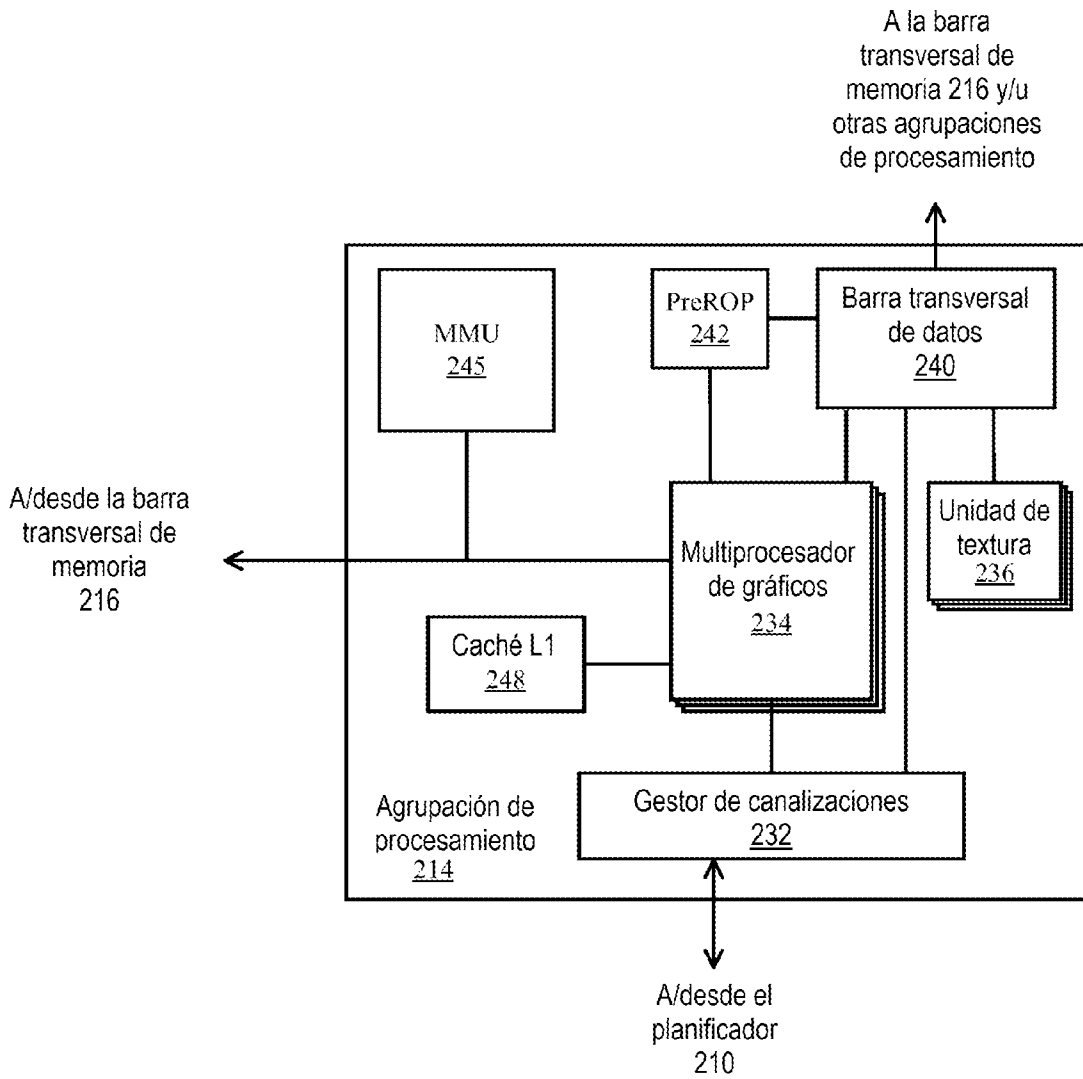


FIG. 2C

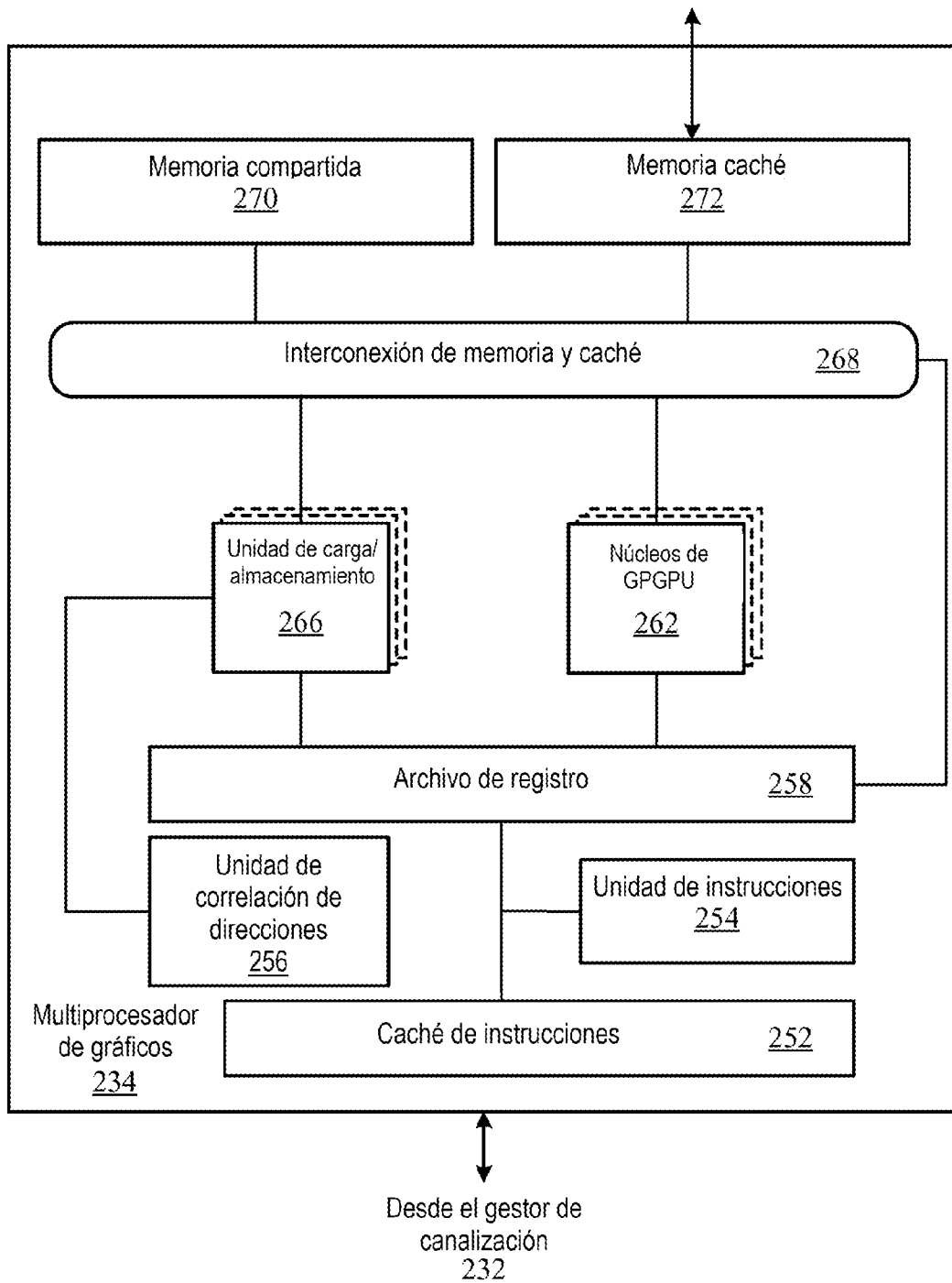


FIG. 2D

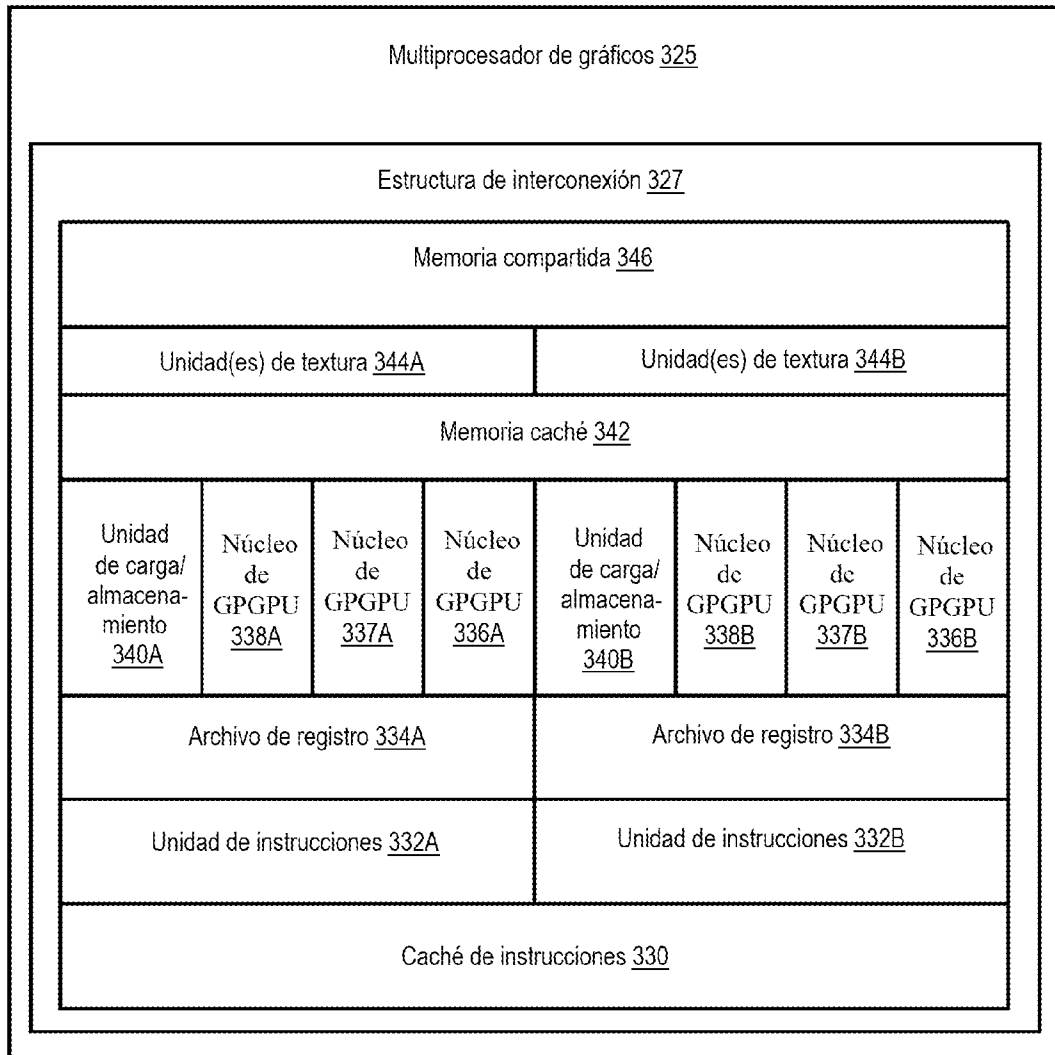


FIG. 3A

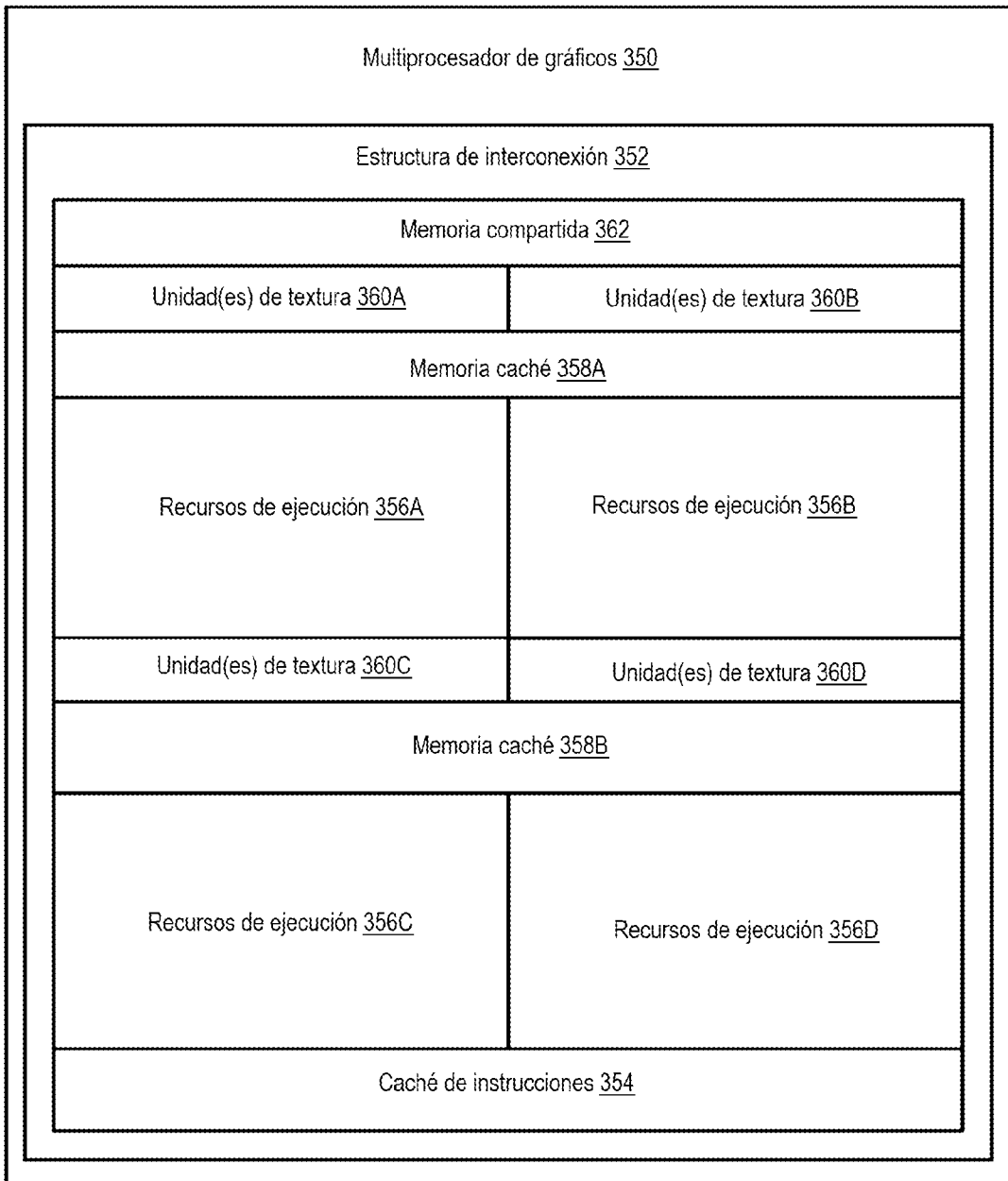


FIG. 3B

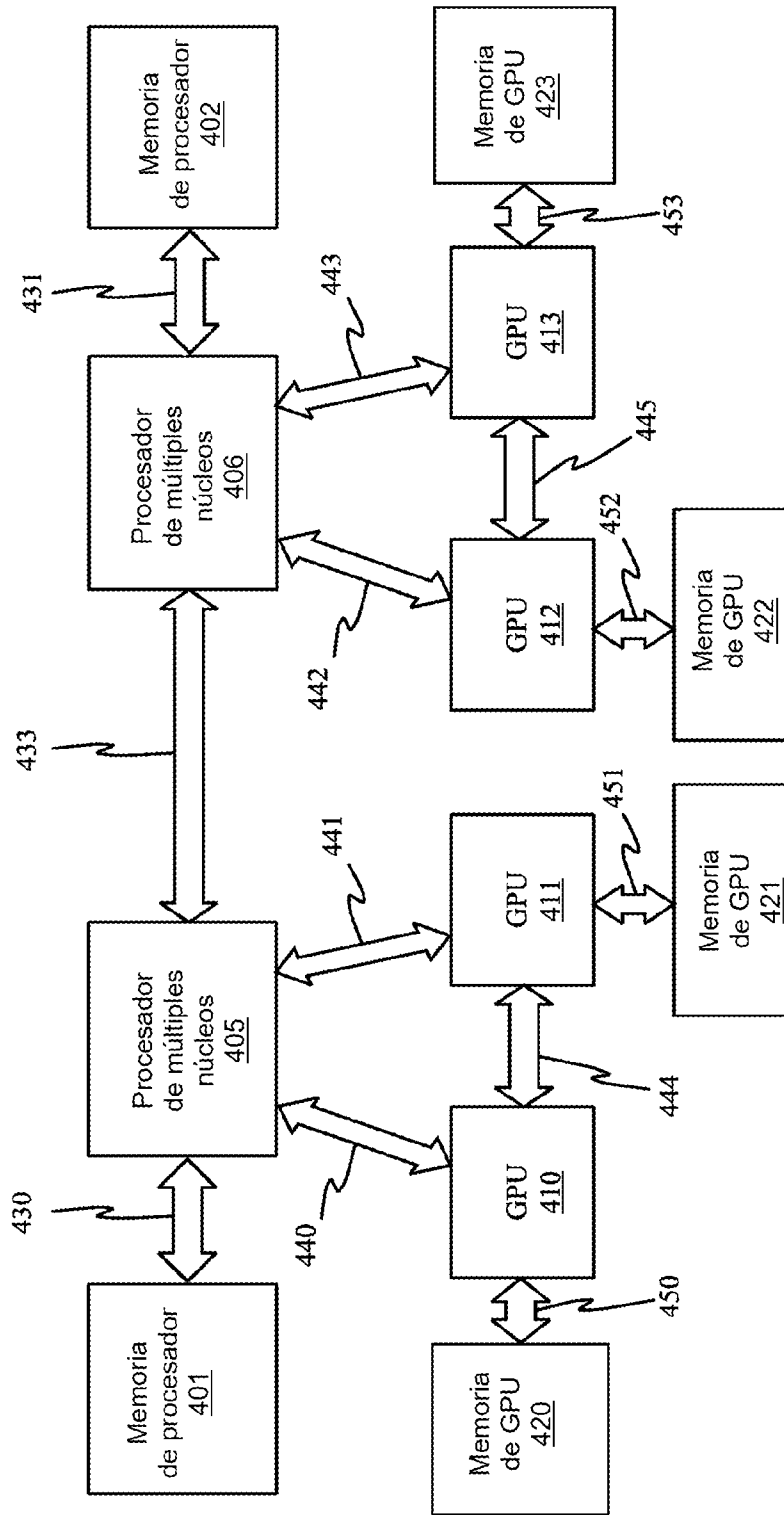


FIG. 4A

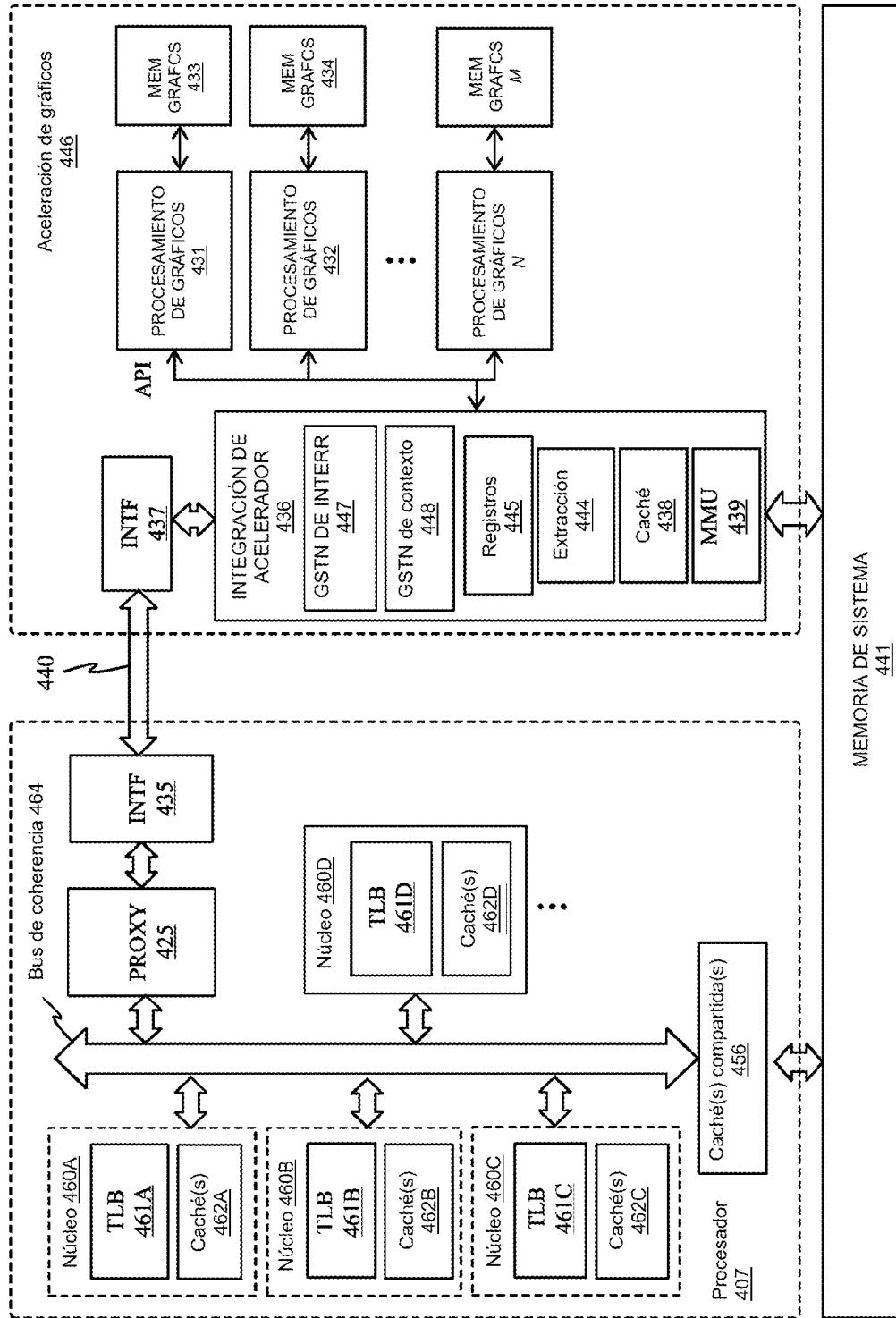


FIG. 4B

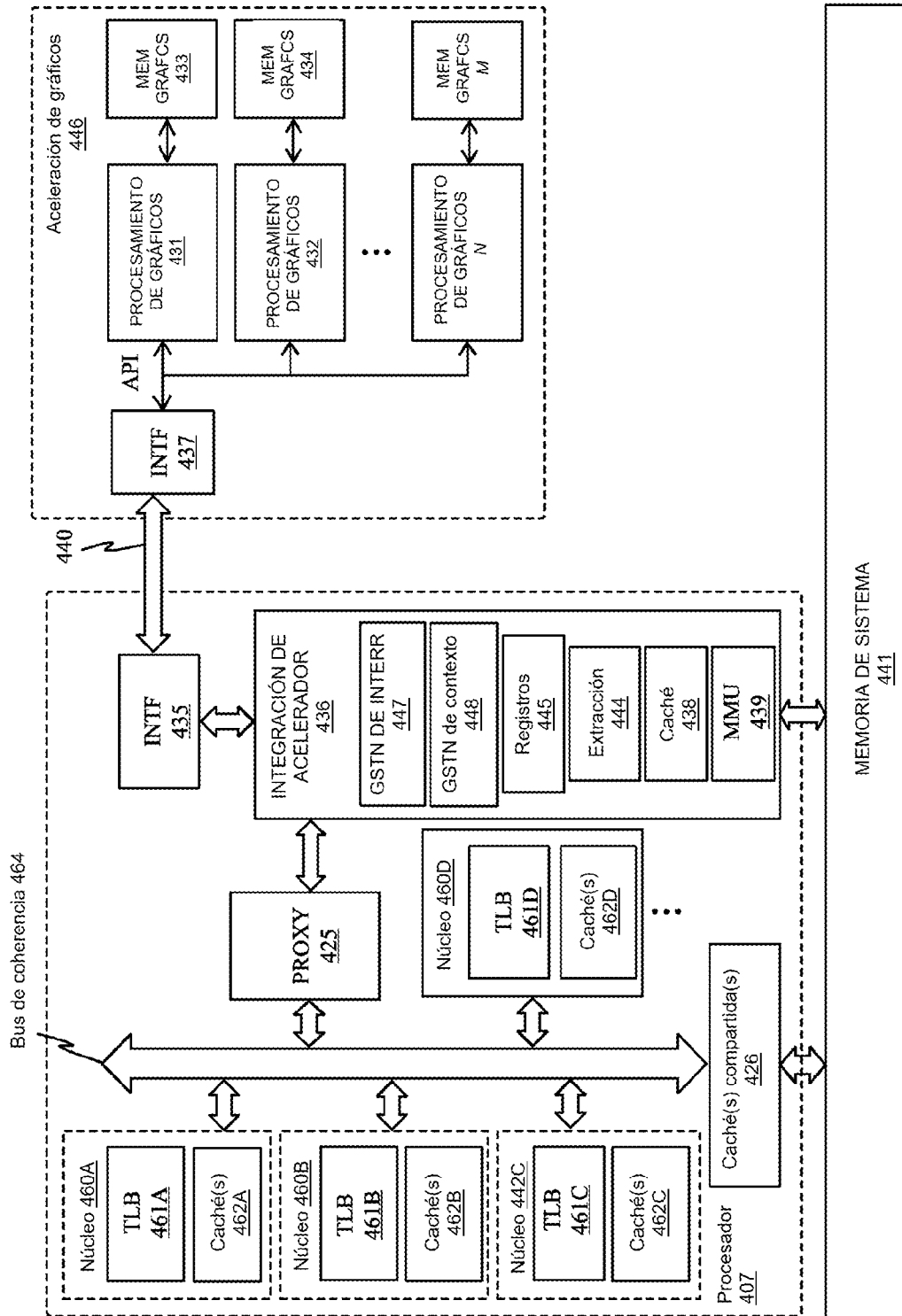


FIG. 4C

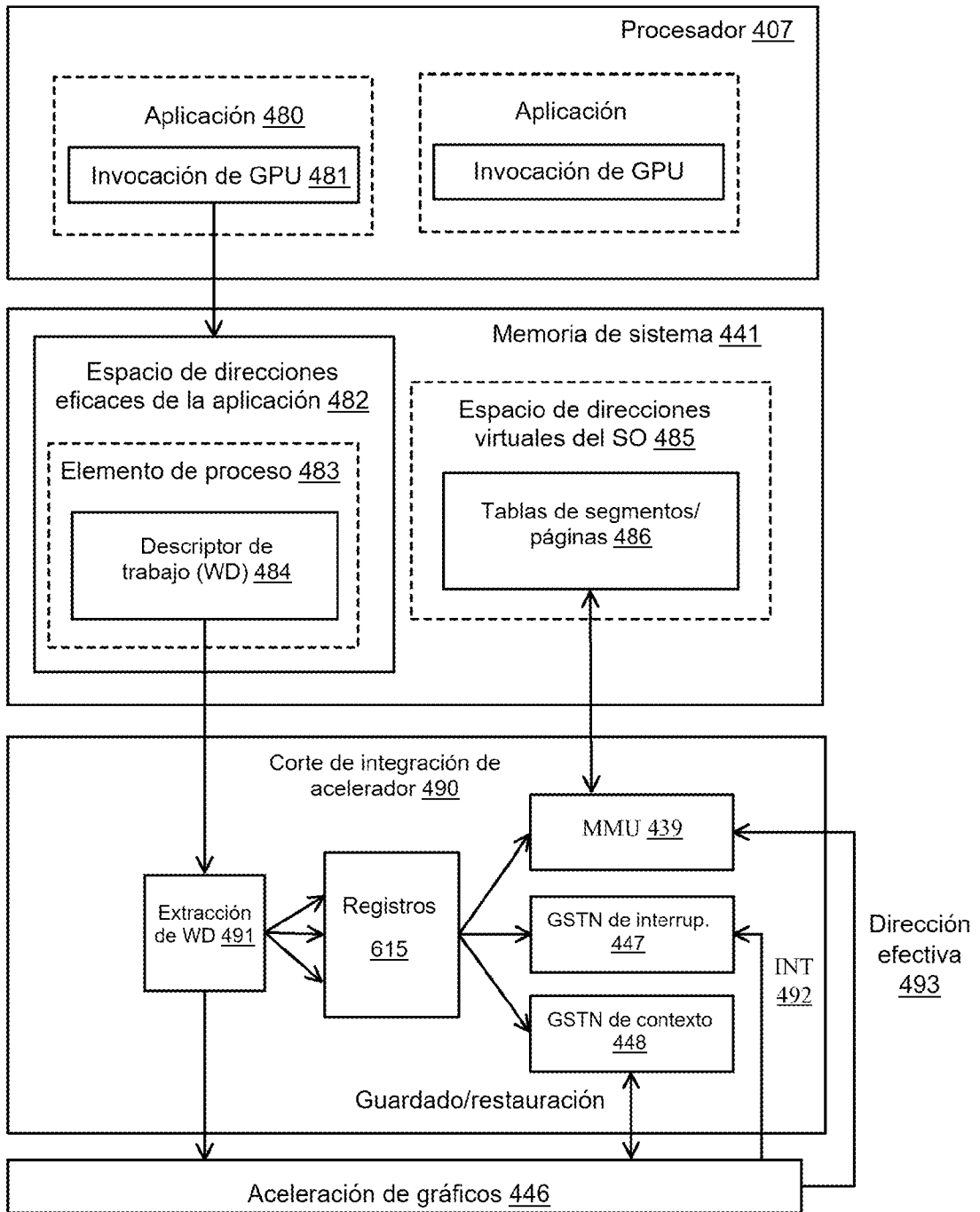


FIG. 4D

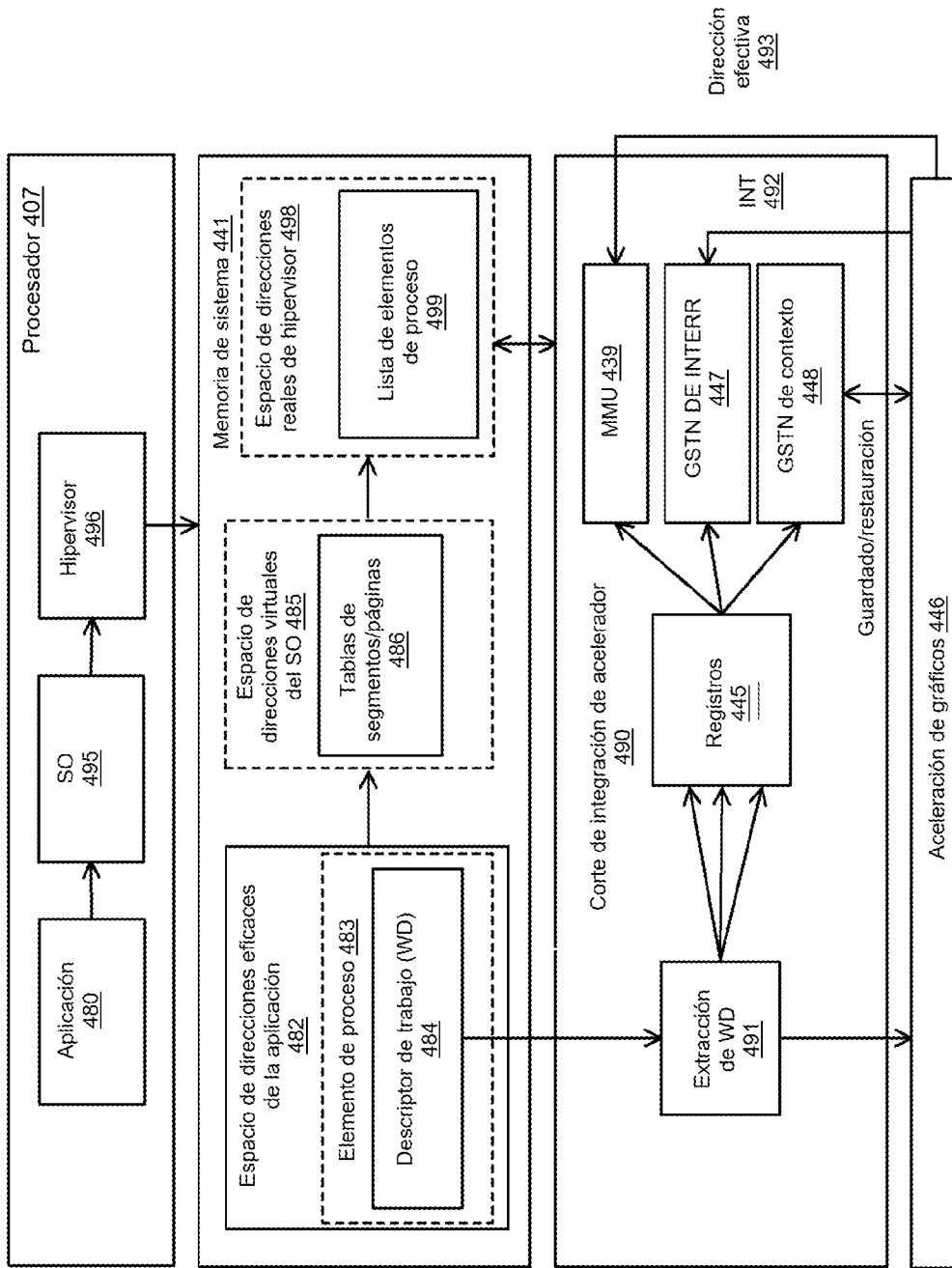


FIG. 4E

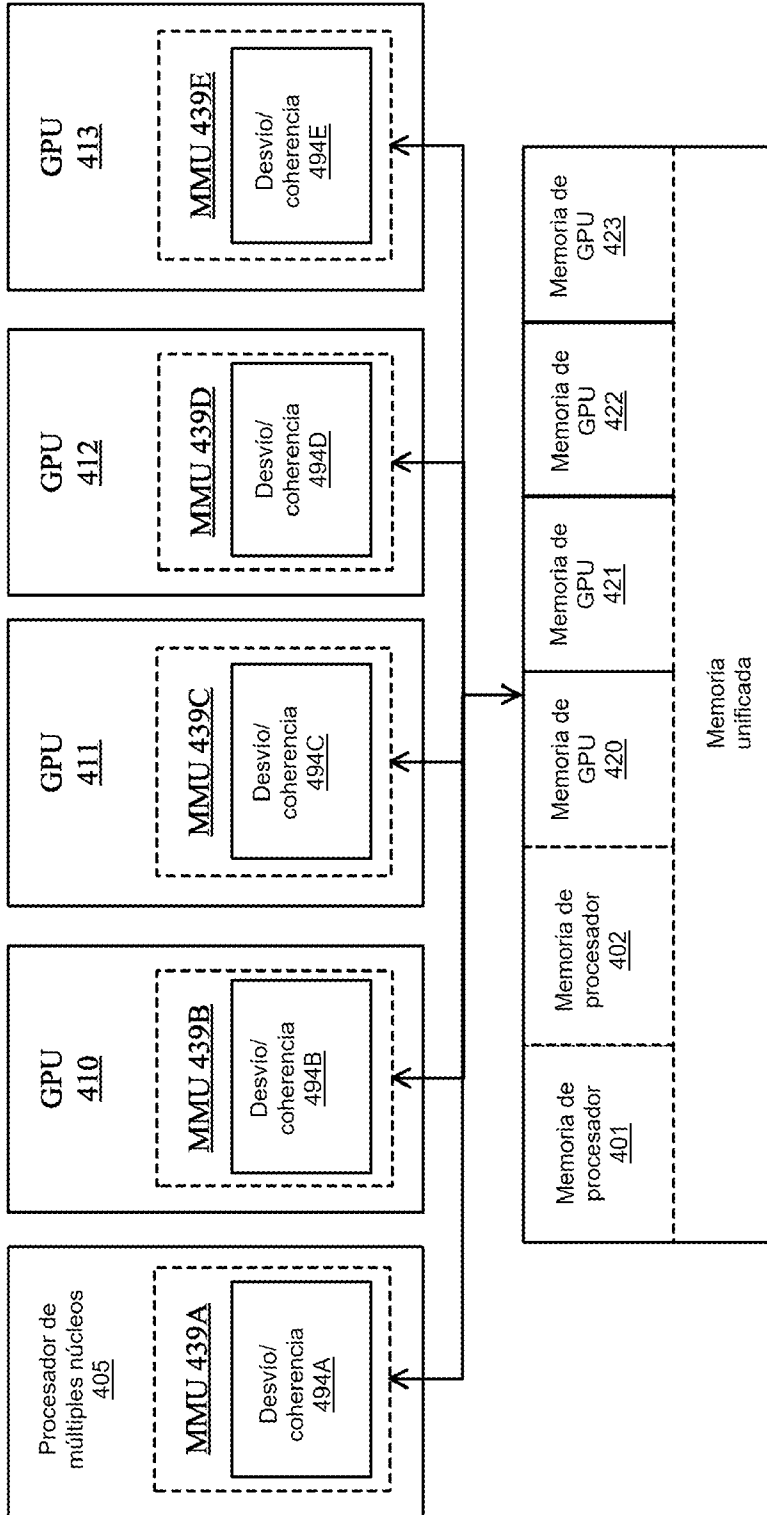


FIG. 4F

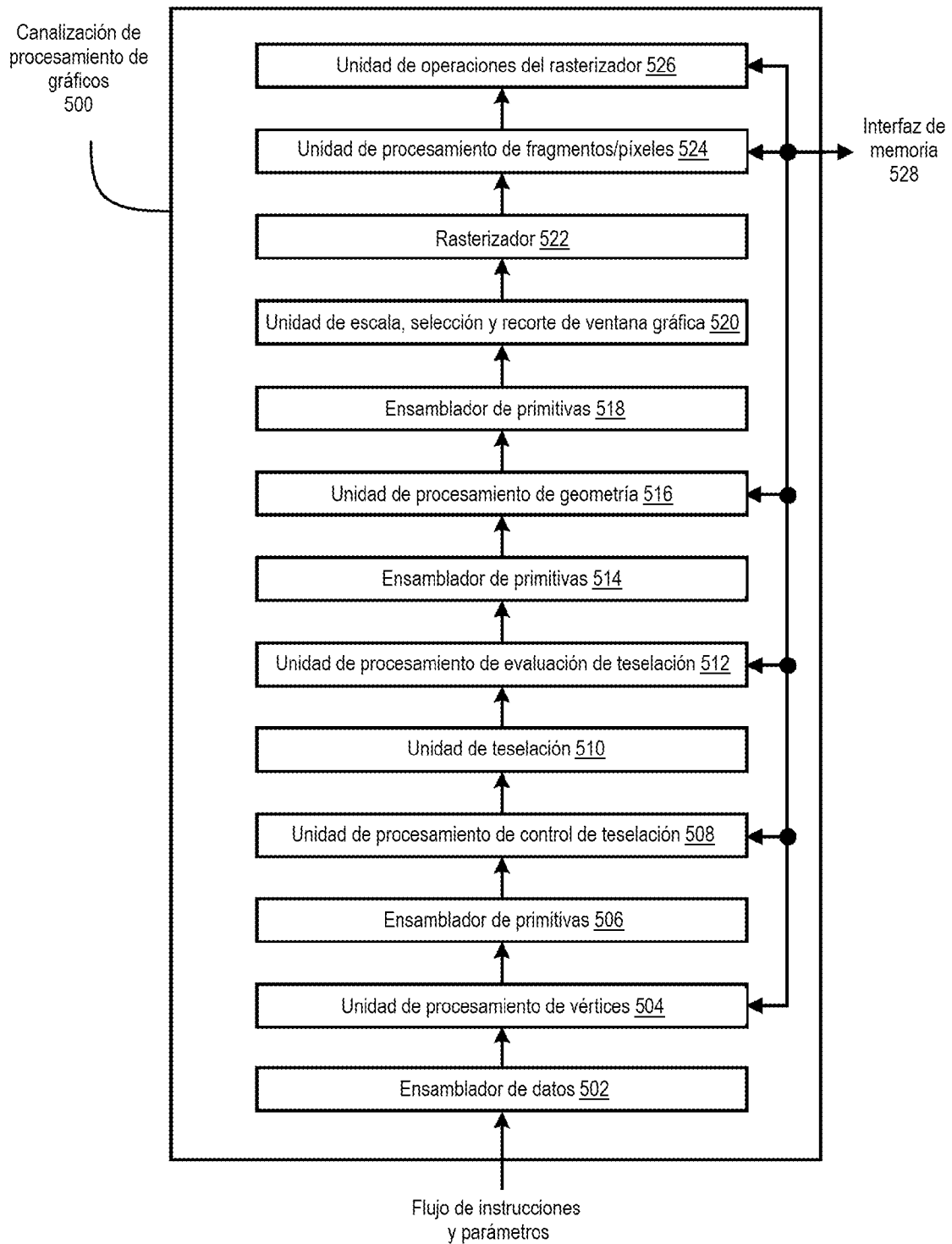


FIG. 5

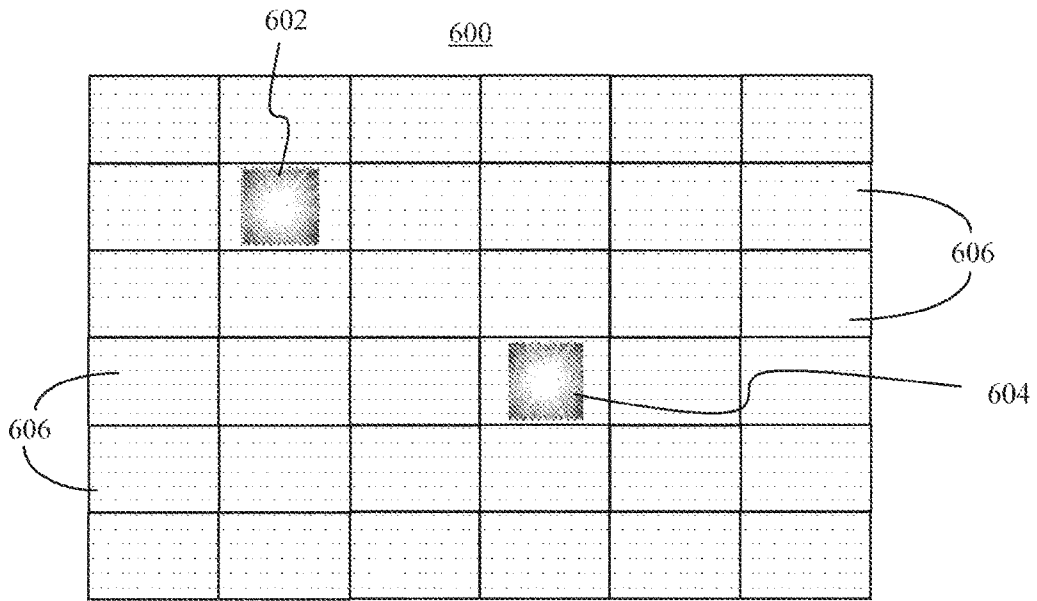


FIG. 6A

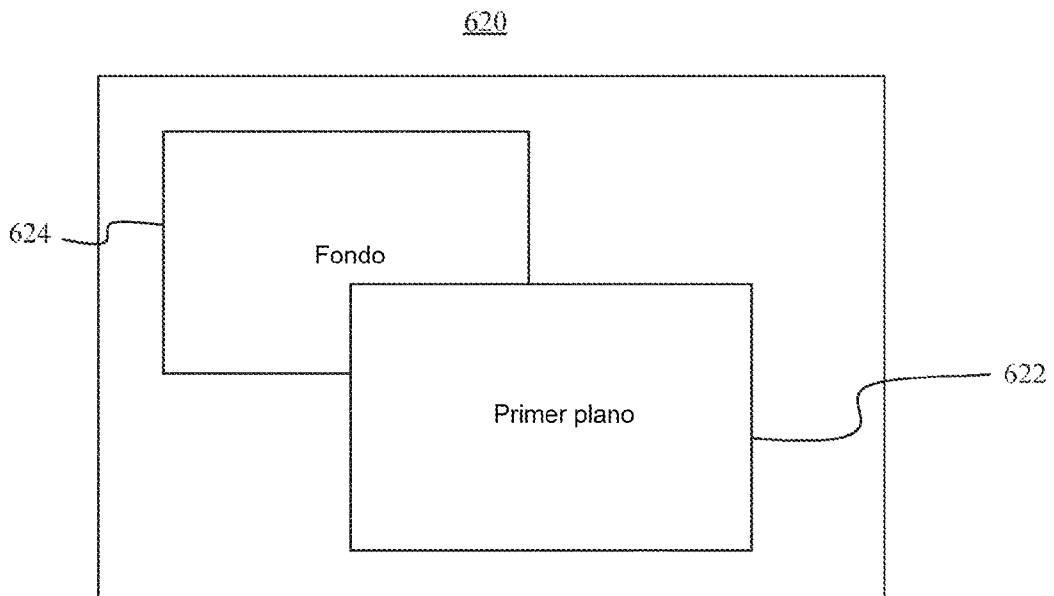


FIG. 6B

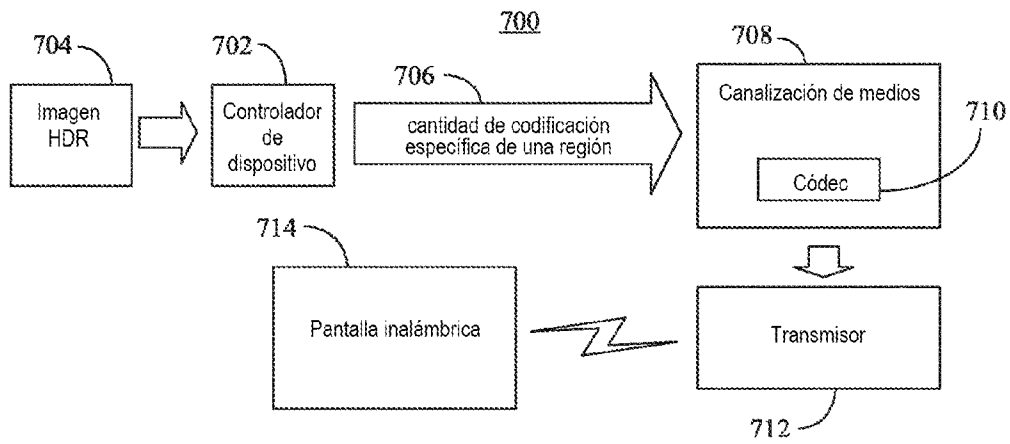


FIG. 7

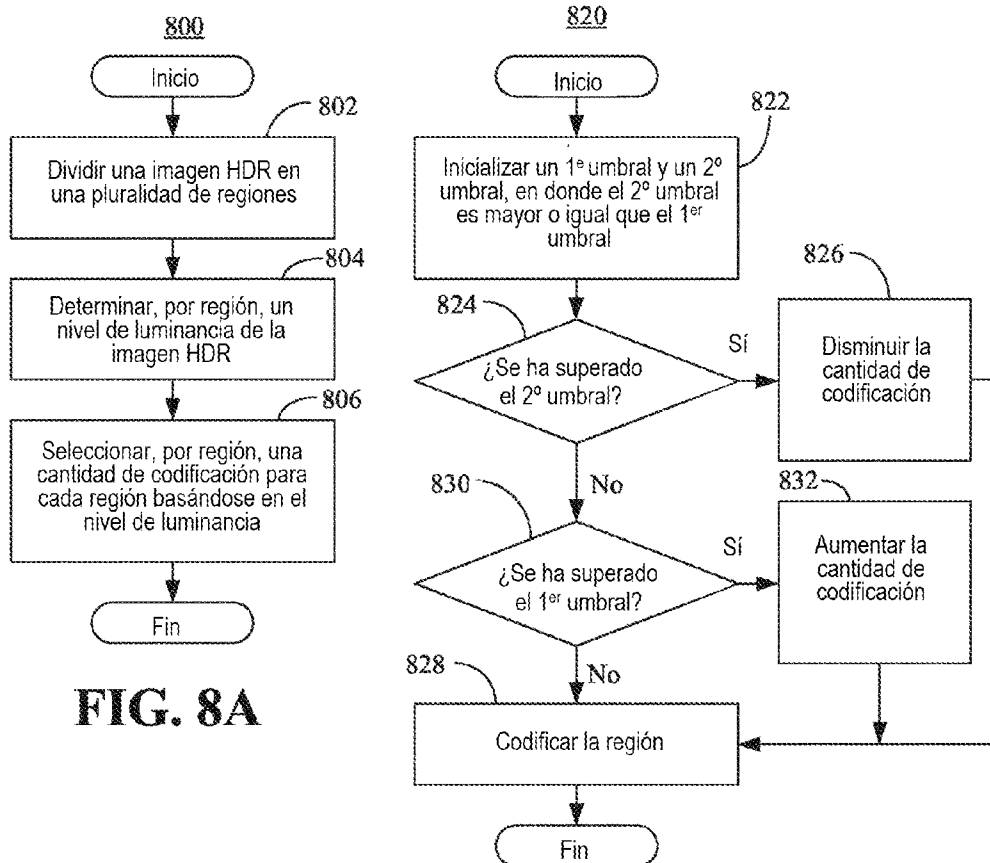


FIG. 8A

FIG. 8B

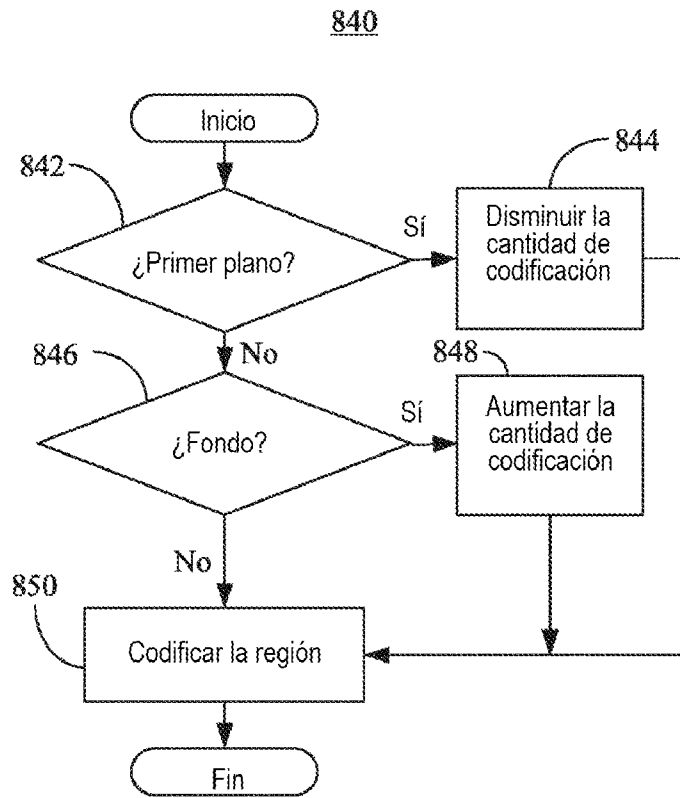


FIG. 8C

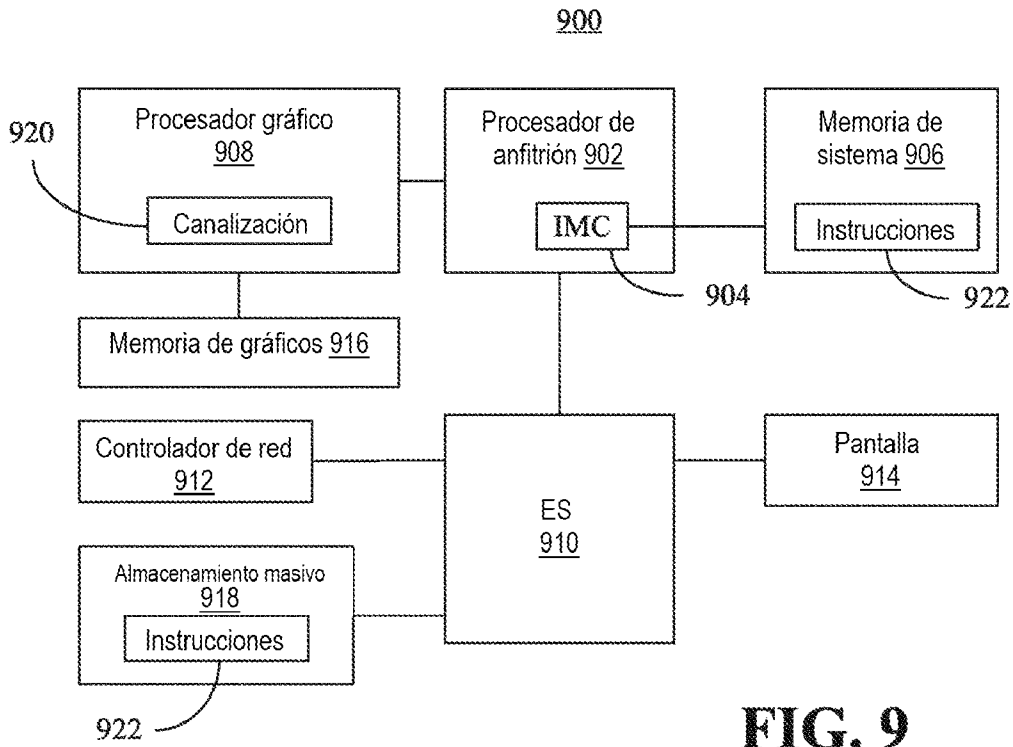


FIG. 9

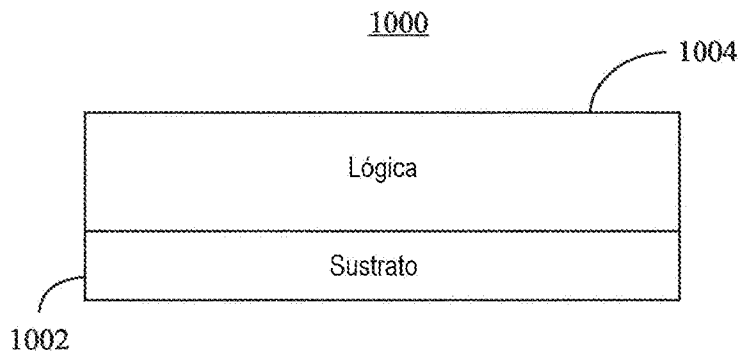


FIG. 10

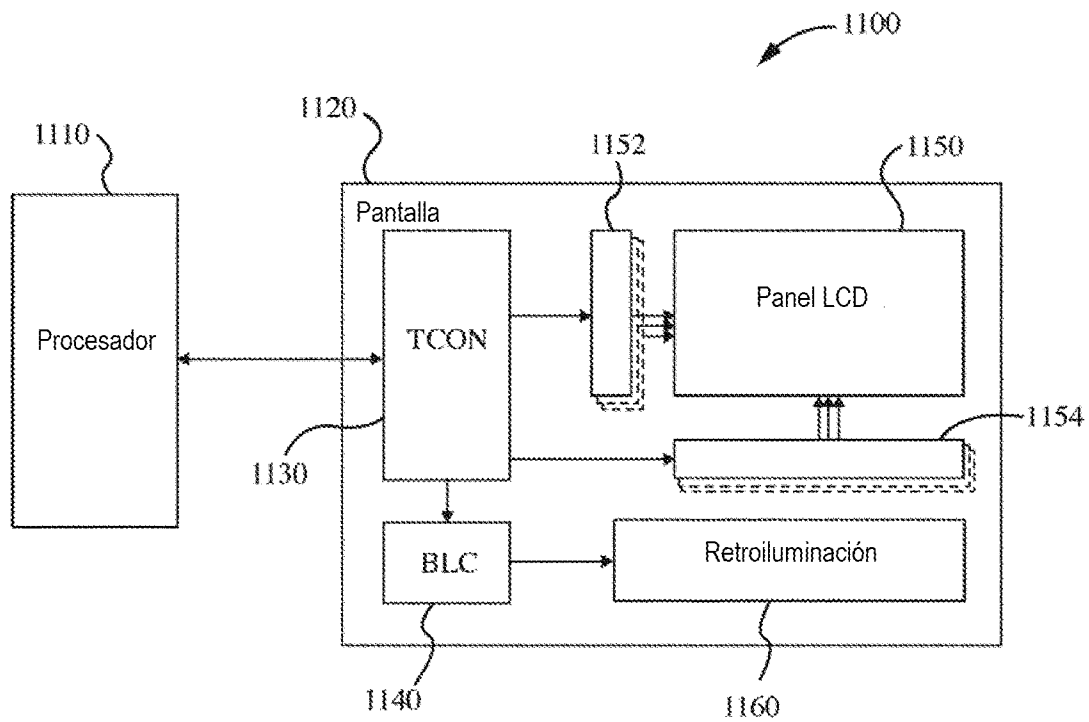


FIG. 11

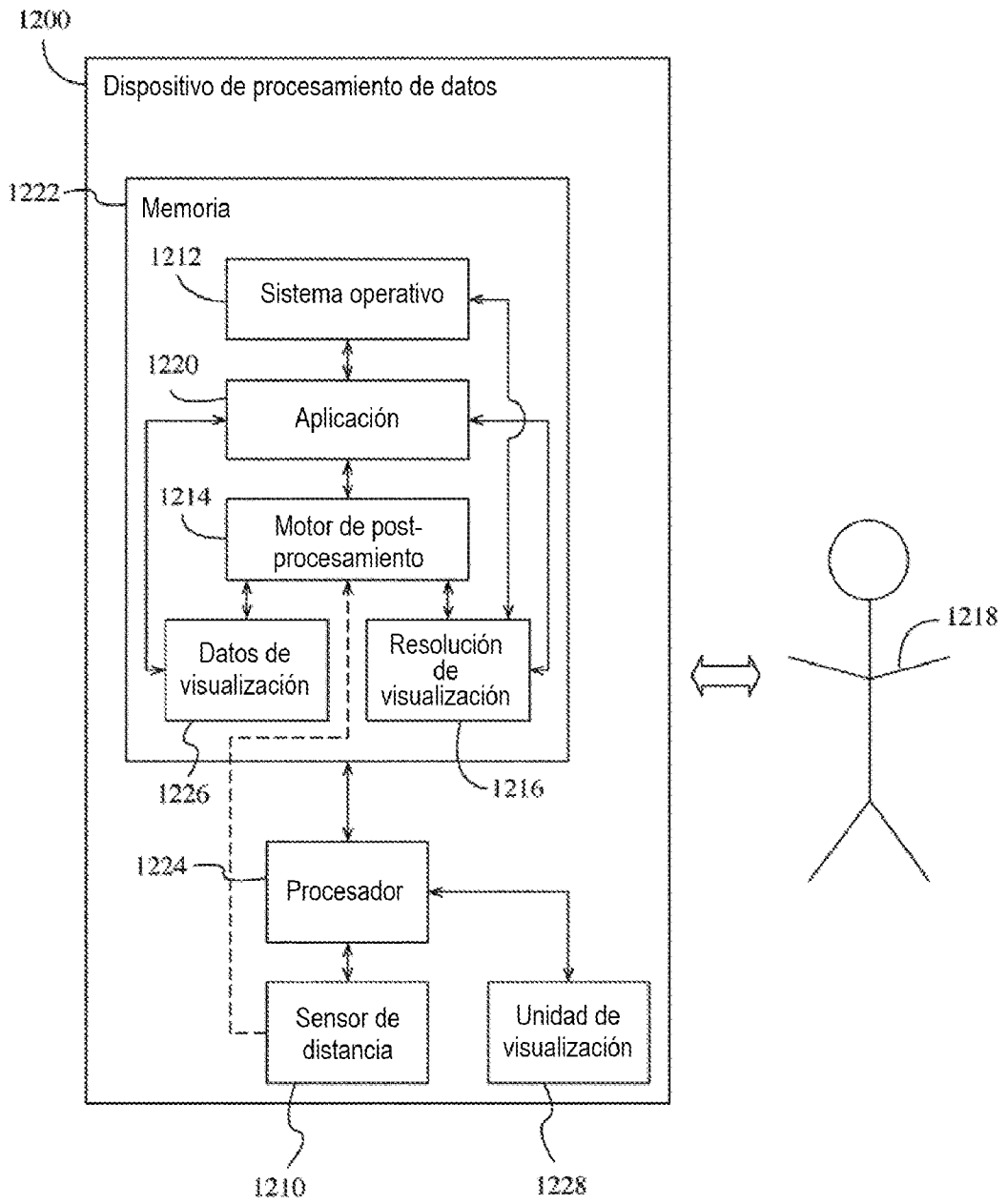


FIG. 12A

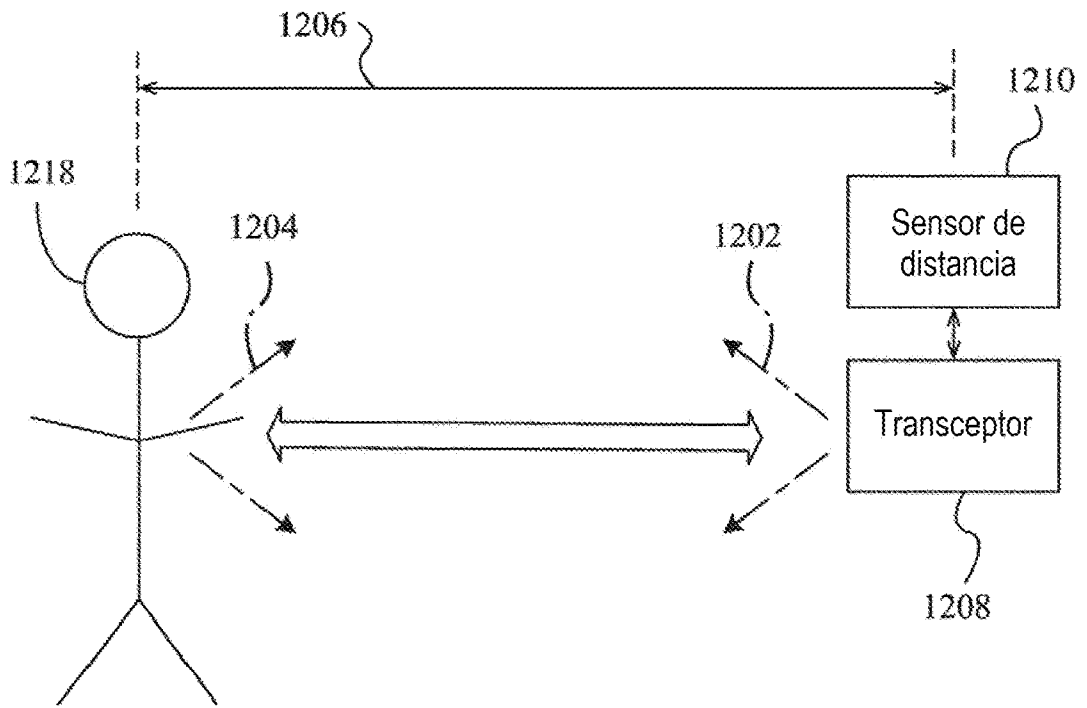


FIG. 12B

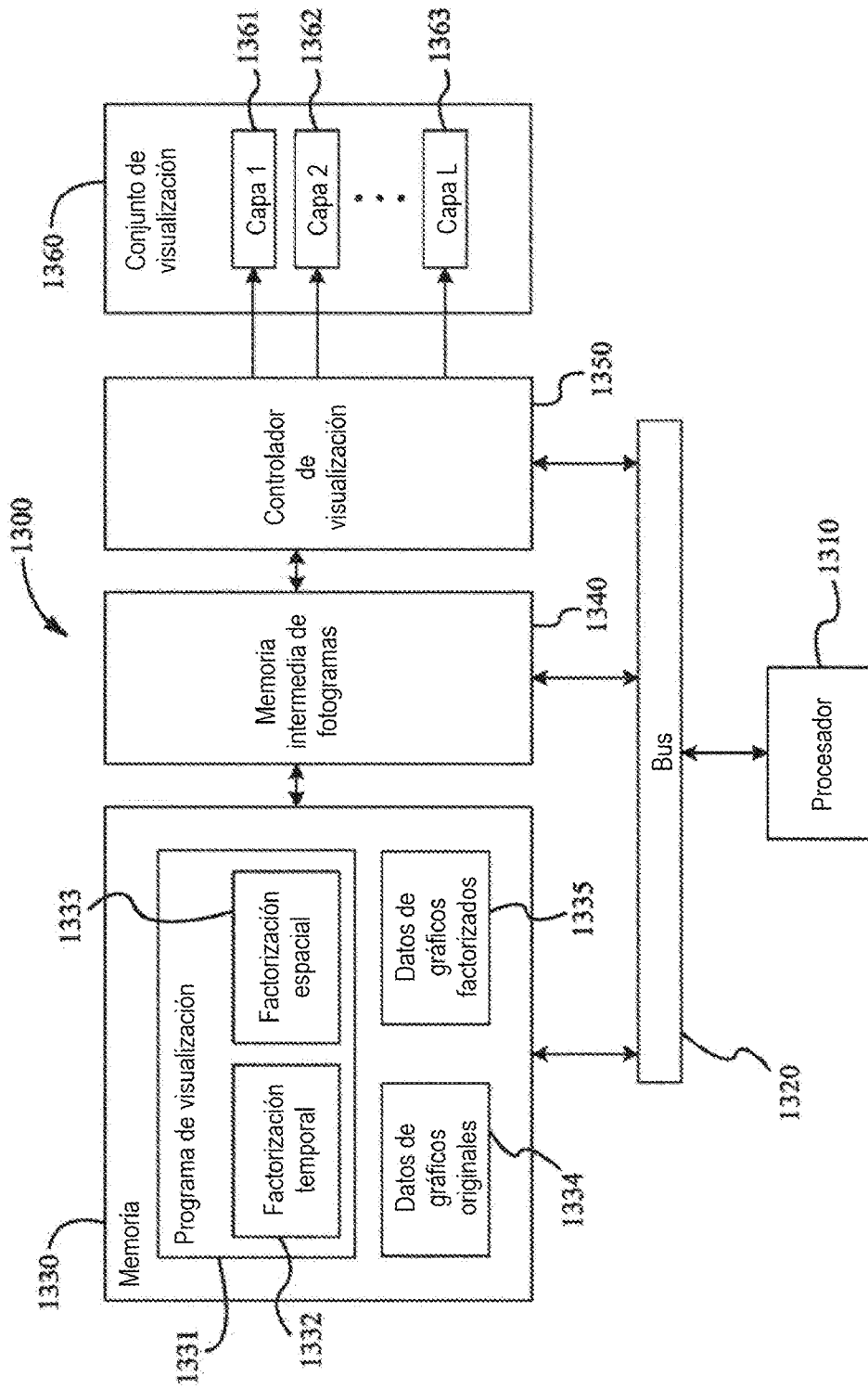


FIG. 13

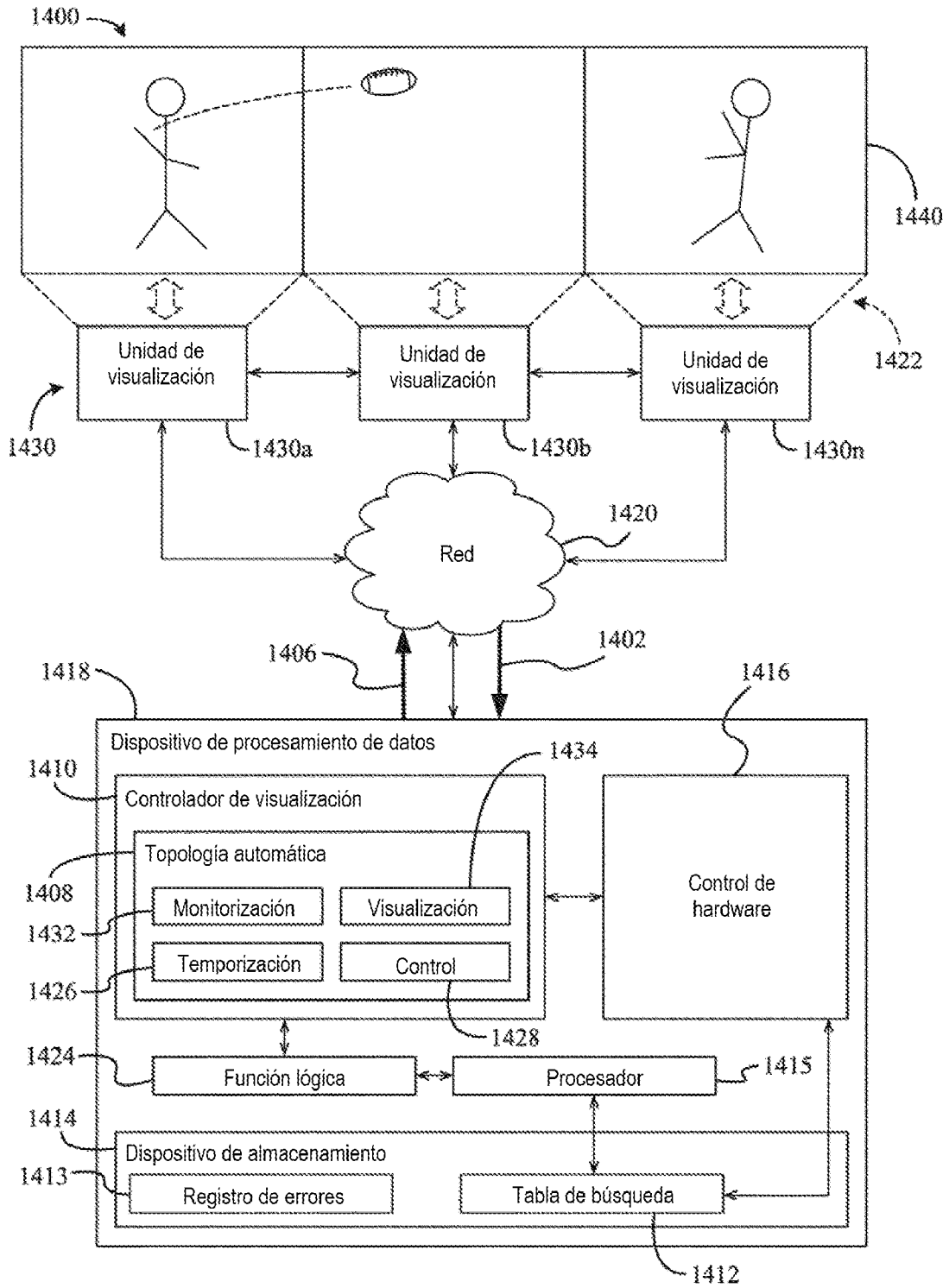


FIG. 14

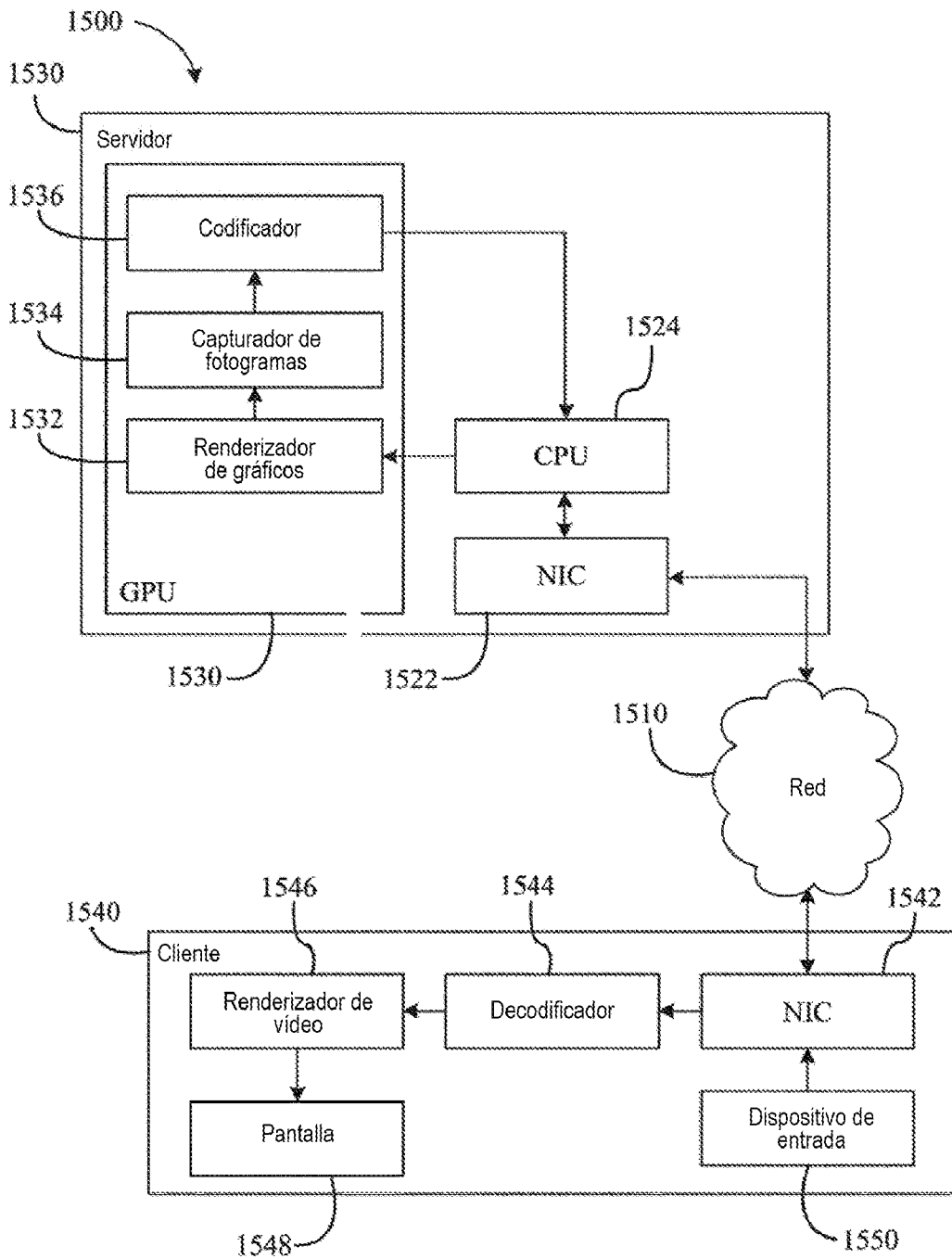


FIG. 15

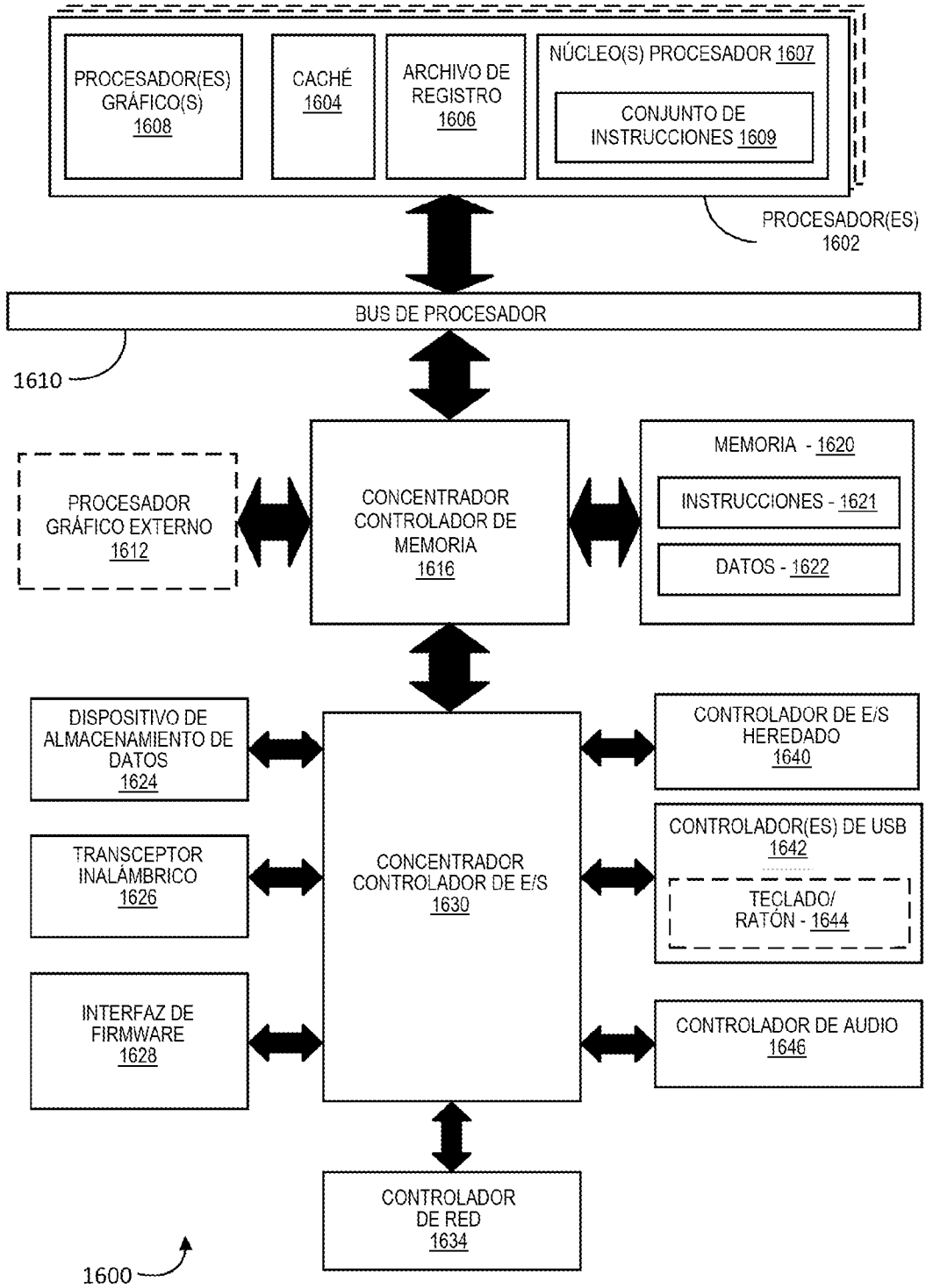


FIG. 16

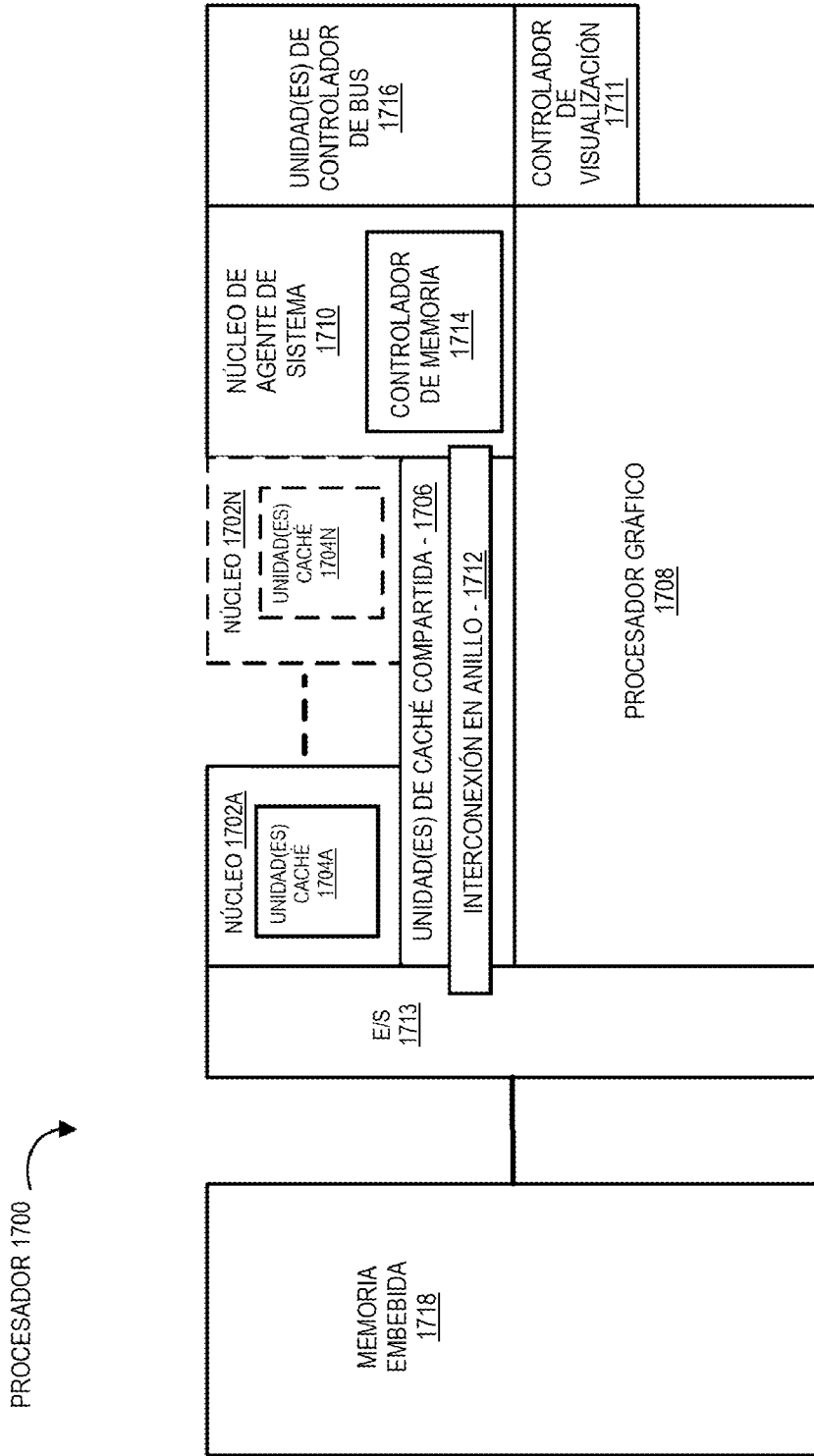


FIG. 17

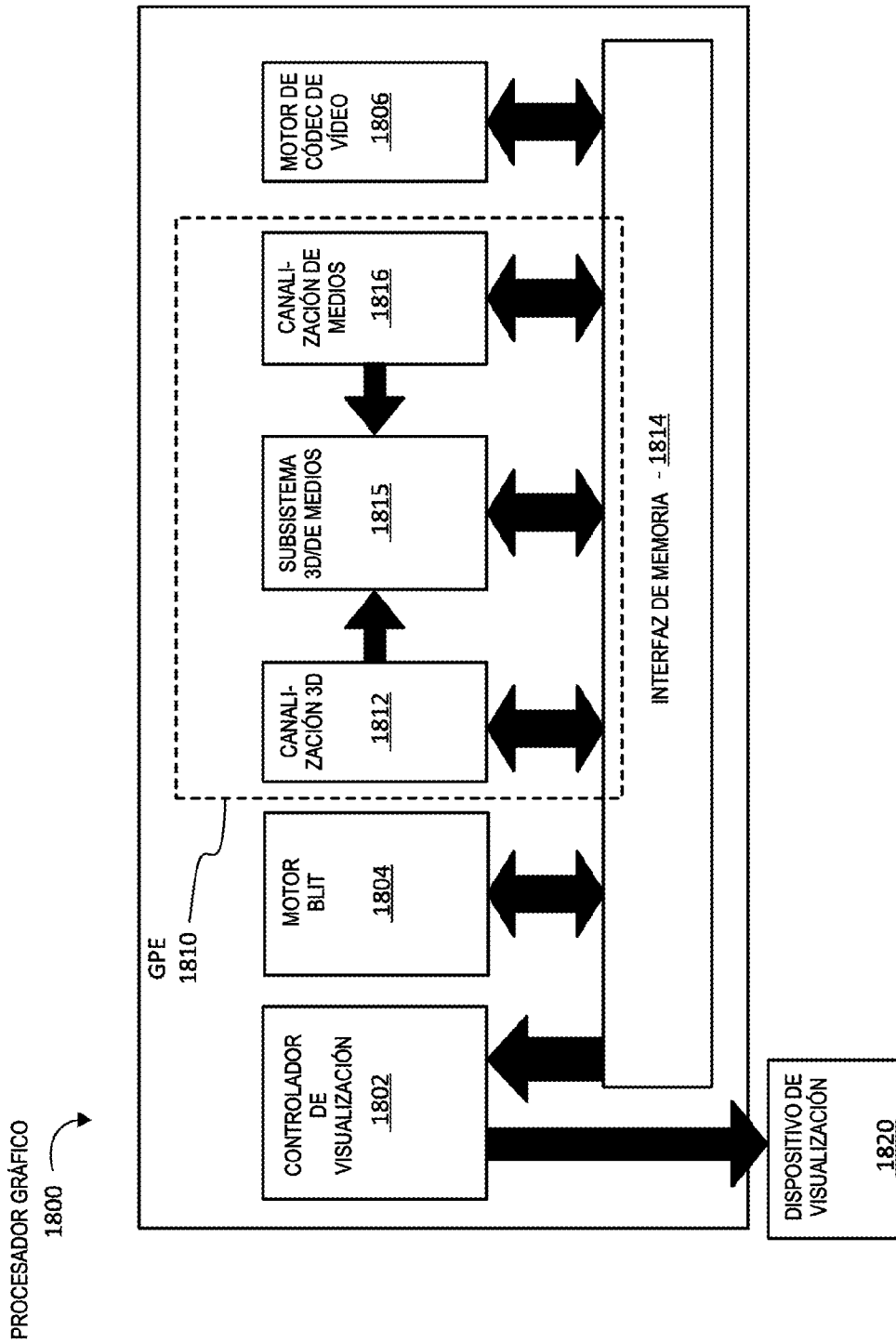


FIG. 18

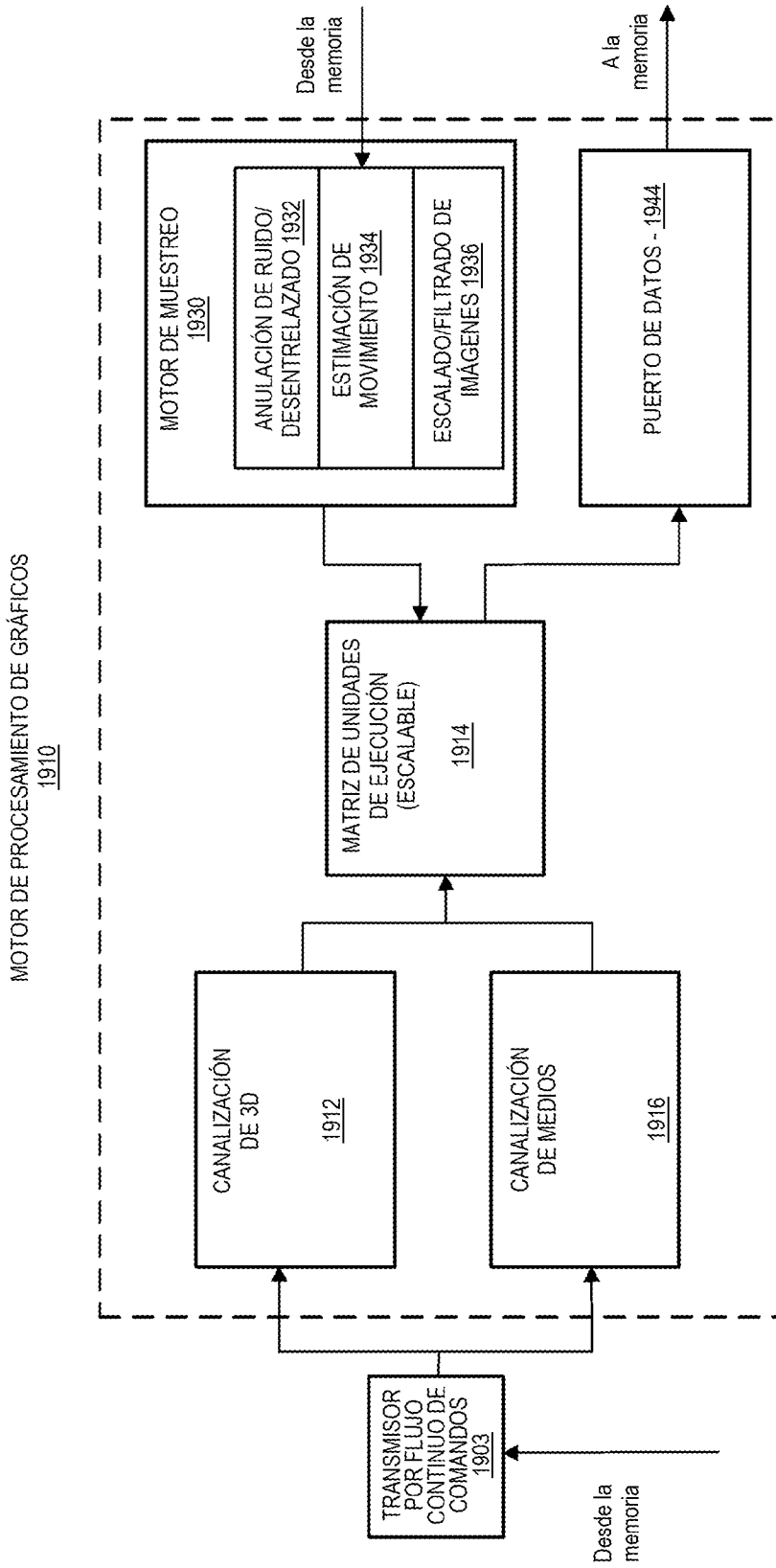


FIG. 19

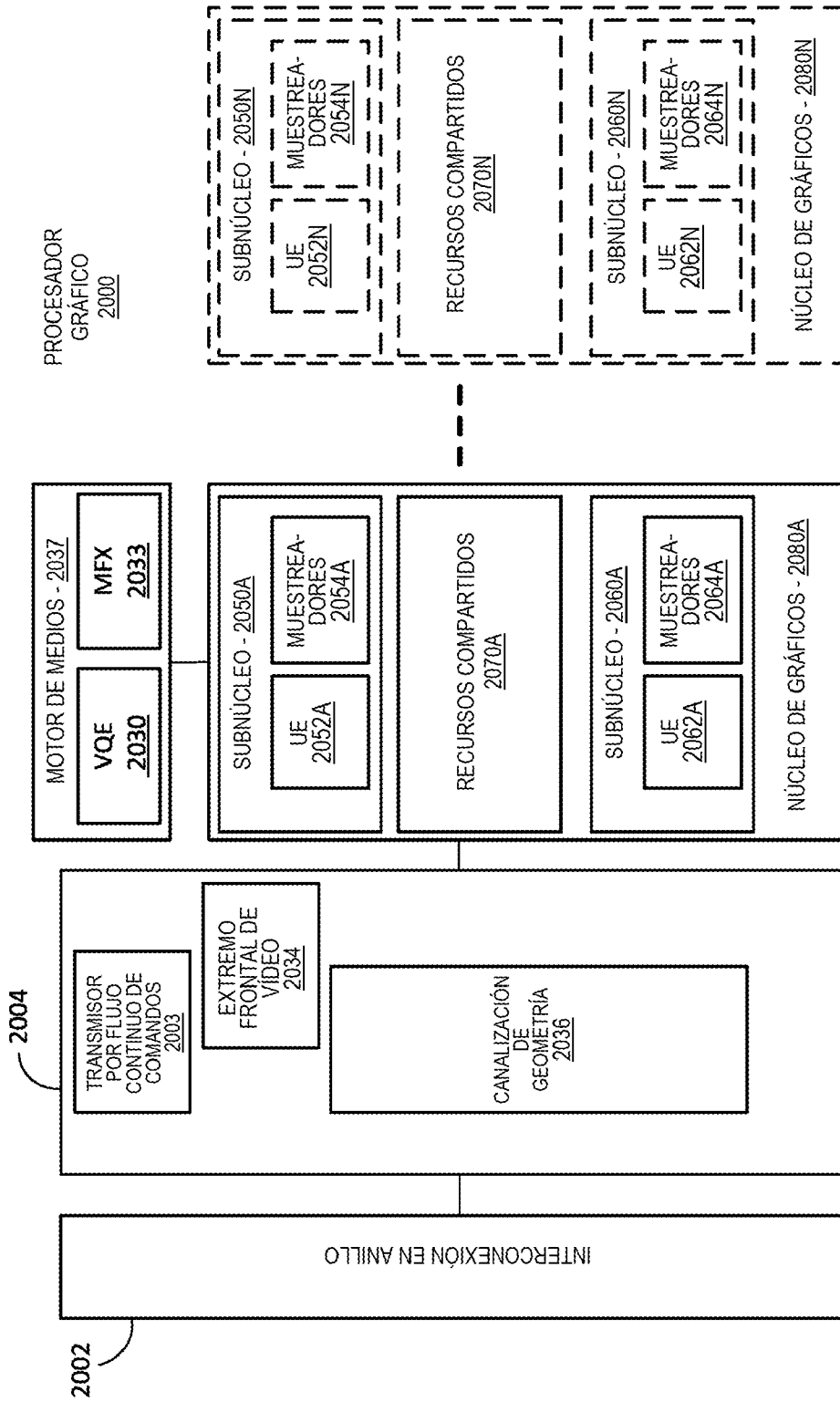


FIG. 20

2100 →

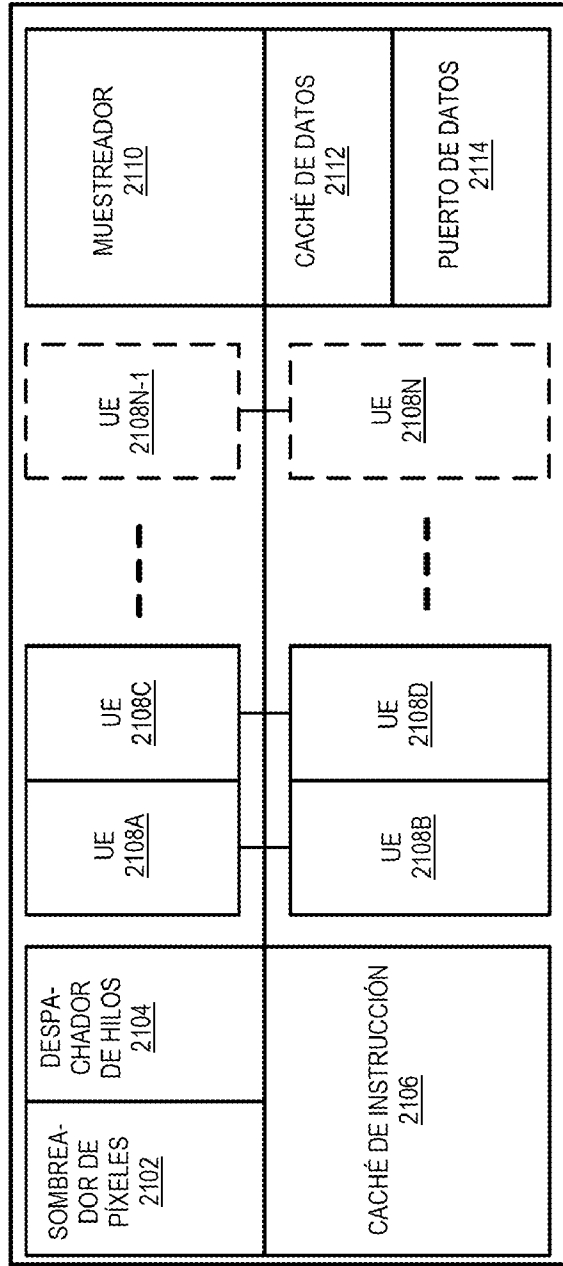
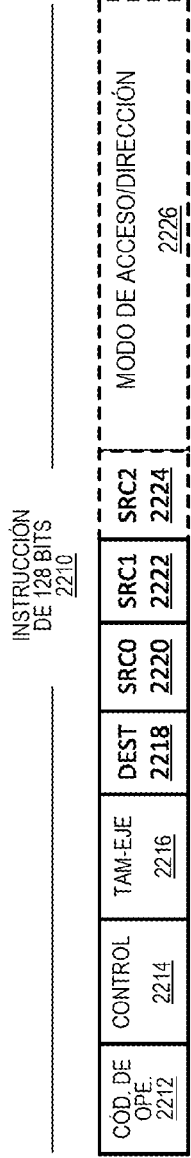
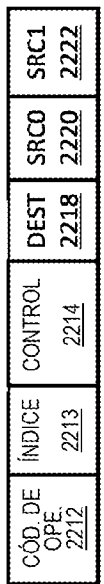


FIG. 21

FORMATOS DE INSTRUCCIONES DE PROCESADOR GRÁFICO
2200



INSTRUCCIÓN COMPACTA DE 64 BITS
2230



DESCODIFICACIÓN DE CÓDIGO DE OPERACIÓN
2240

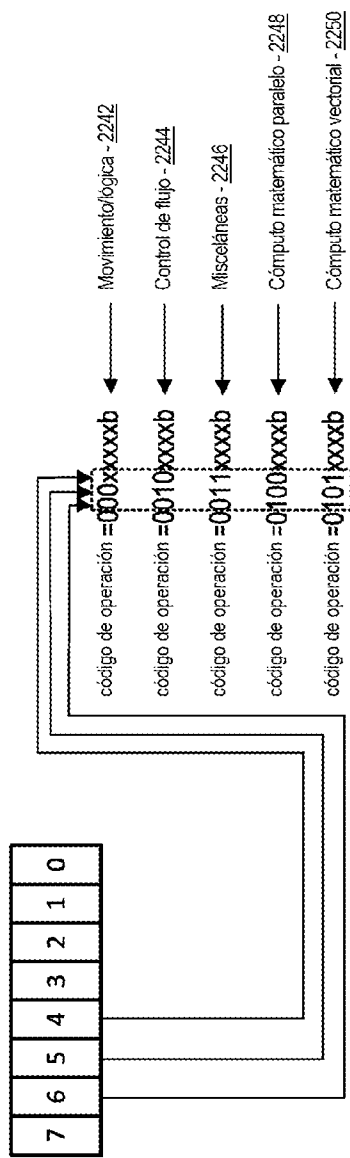


FIG. 22

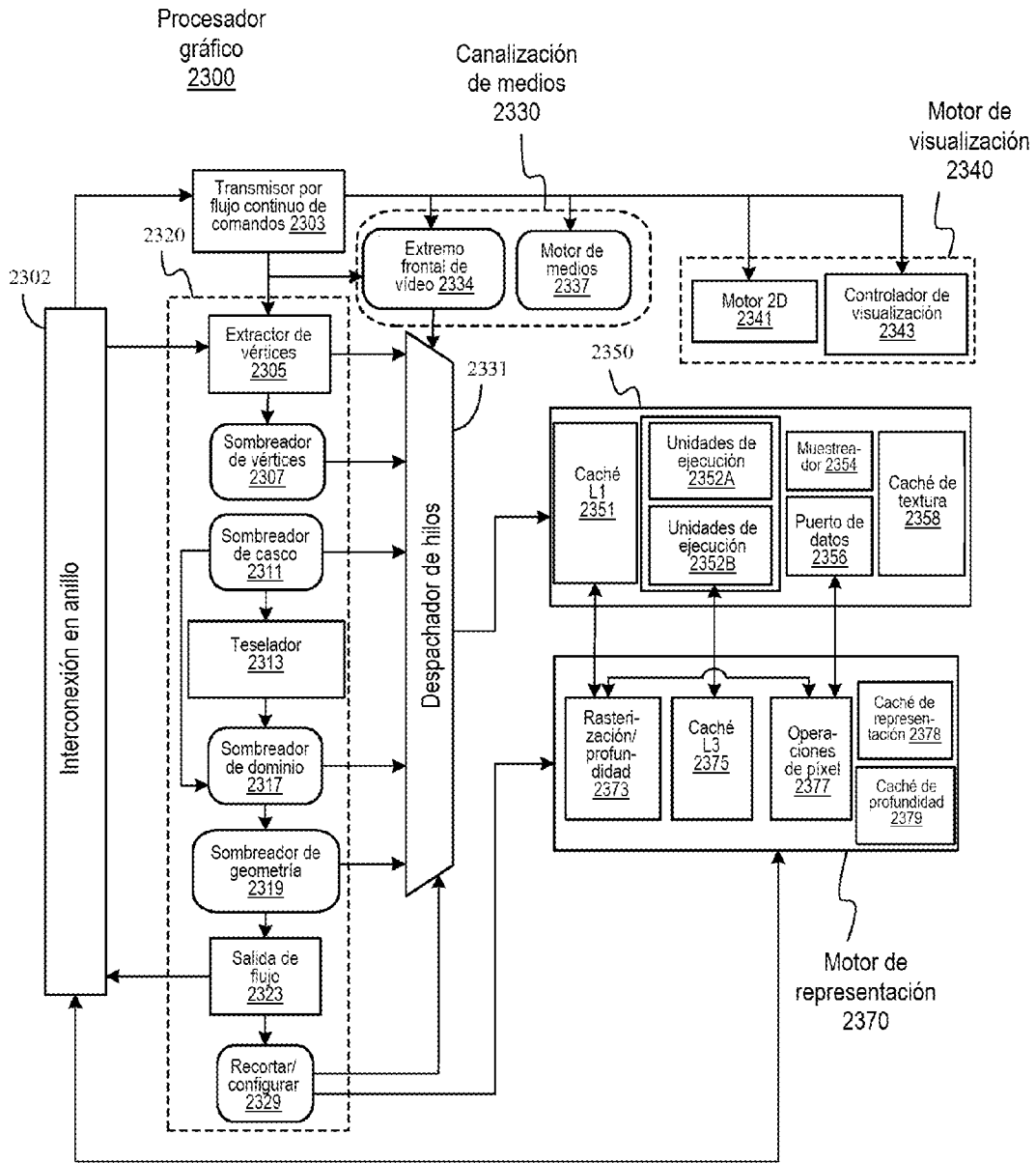


FIG. 23

Formato de comandos de procesador gráfico
2400

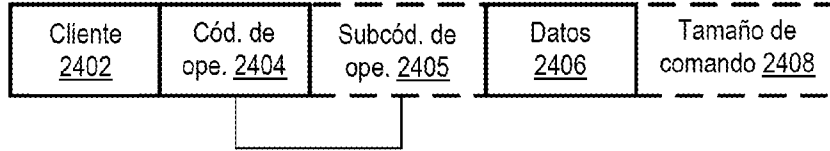


FIG. 24A

Secuencia de comandos de procesador gráfico
2410

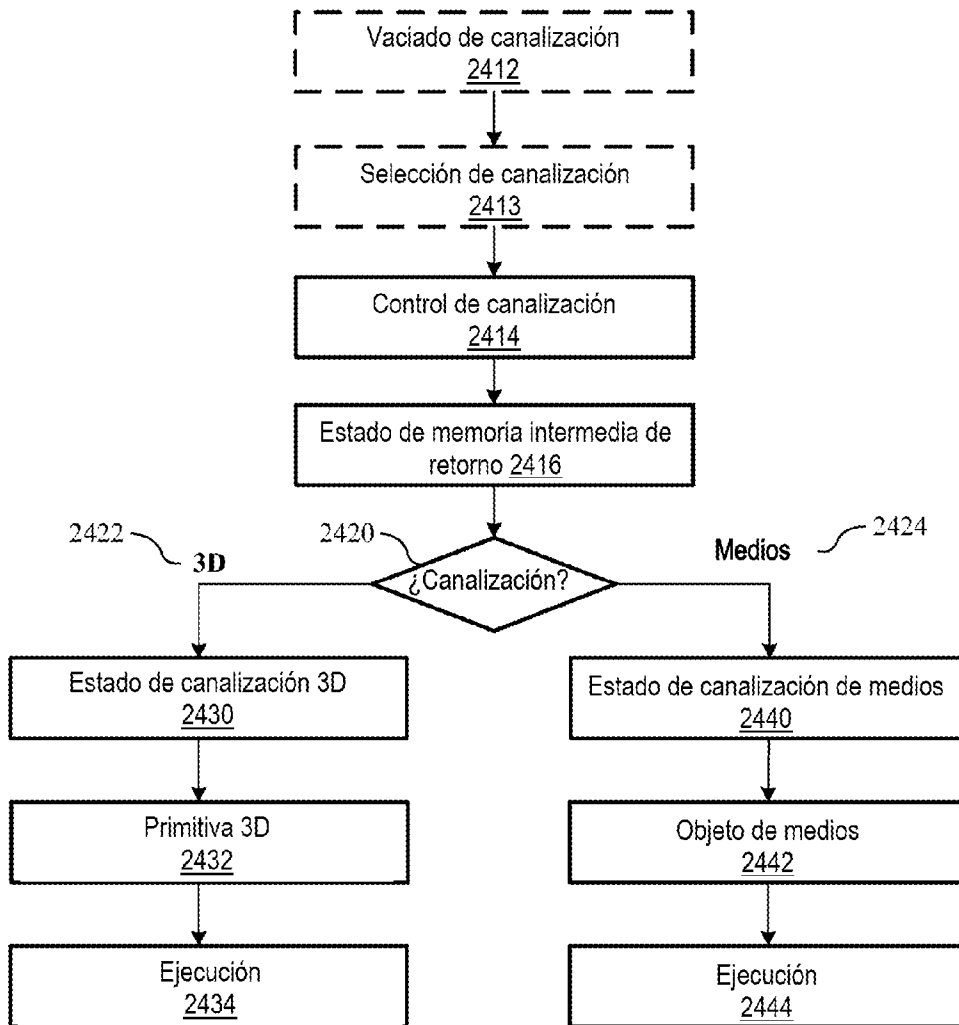


FIG. 24B

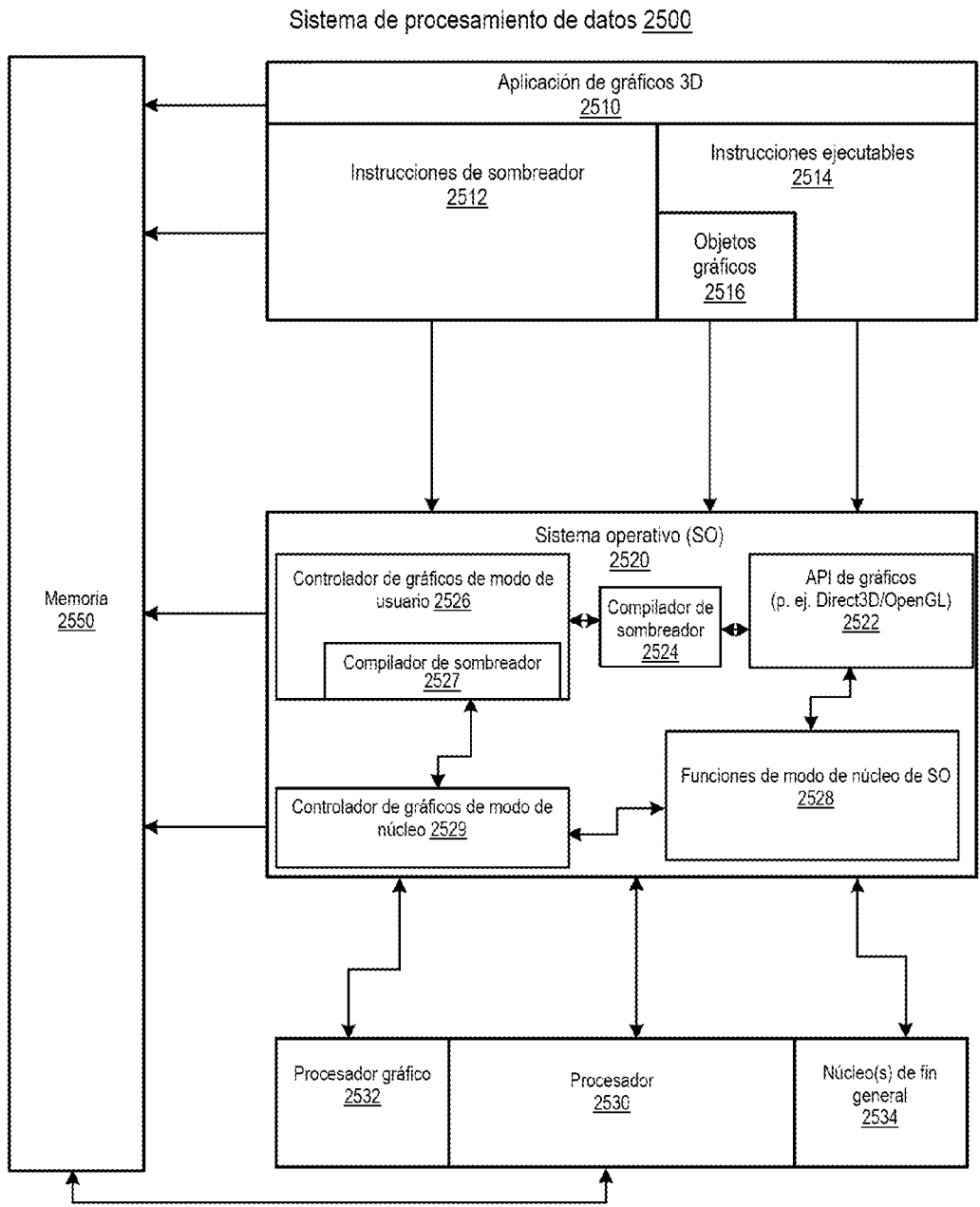


FIG. 25

DESARROLLO DE NÚCLEO DE IP - 2600

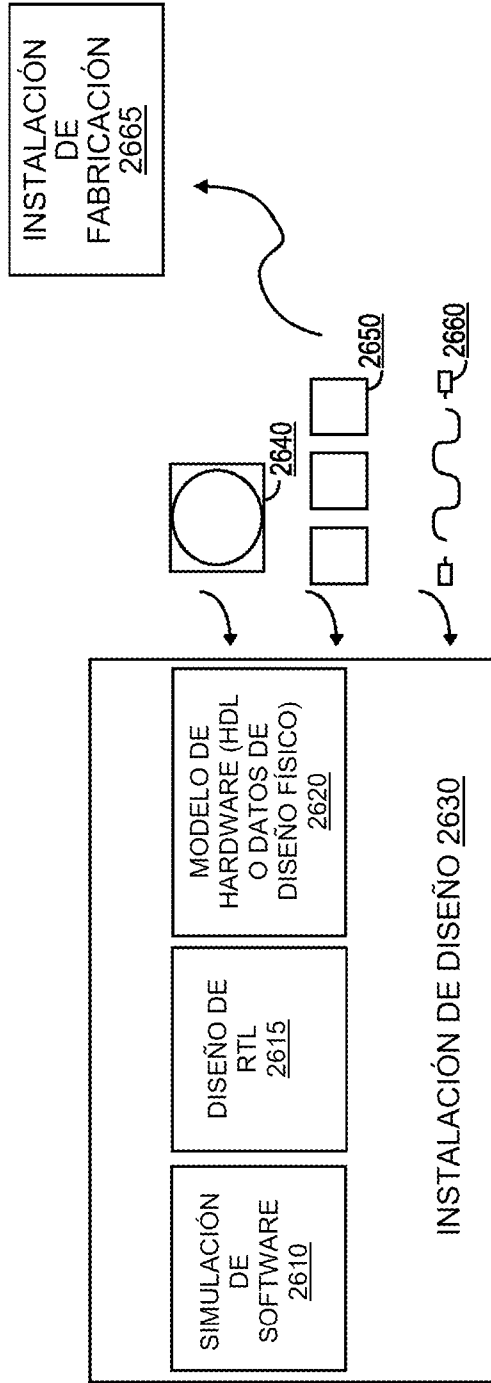


FIG. 26

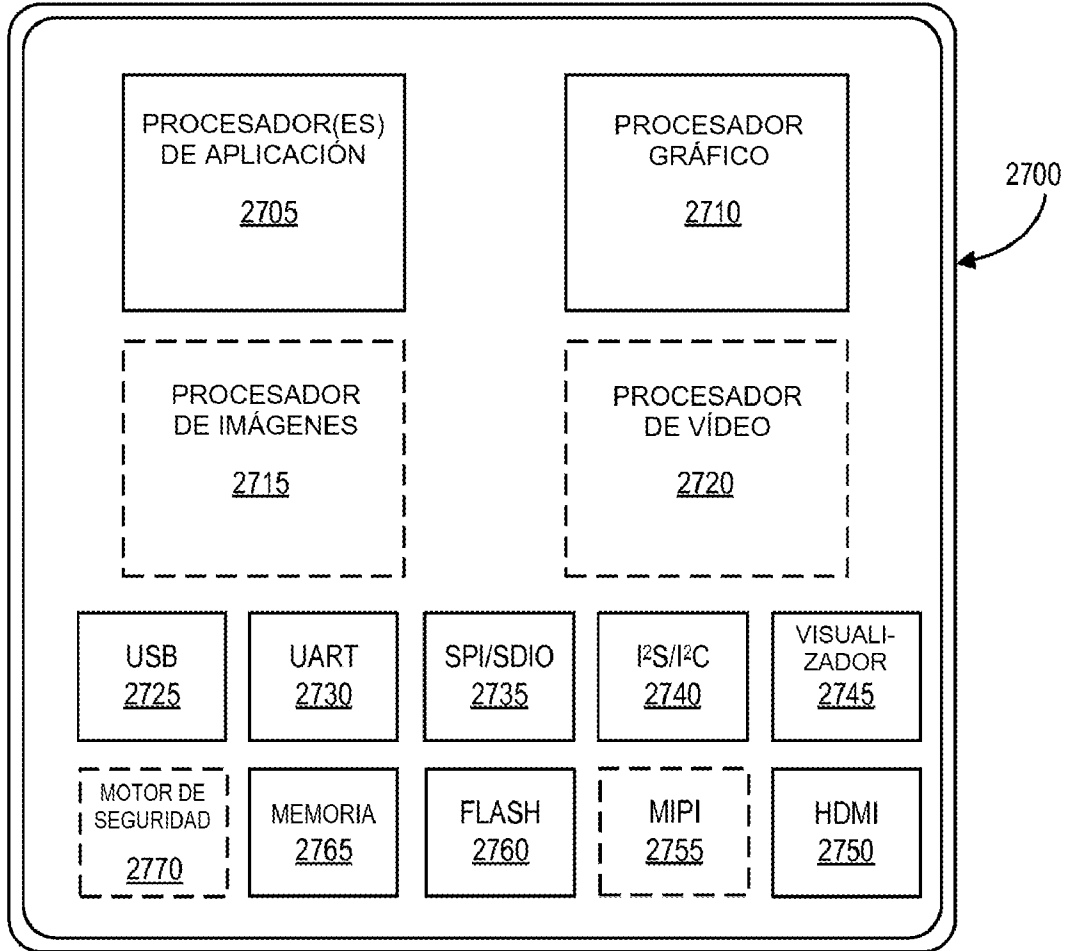


FIG. 27