

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 915 592**

51 Int. Cl.:

B60R 1/00

(2012.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **01.03.2018** **E 19205415 (3)**

97 Fecha y número de publicación de la concesión europea: **30.03.2022** **EP 3663136**

54 Título: **Uso óptimo de espejos virtuales de automóvil**

30 Prioridad:

01.04.2017 US 201715477004

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

23.06.2022

73 Titular/es:

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

72 Inventor/es:

**BOYCE, JILL M.;
BARAN, STANLEY J.;
MOHAN, SUMIT;
TANNER, JASON;
CHIU, YI-JEN y
MOHAMMED, ATTHAR H.**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 915 592 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

Uso óptimo de espejos virtuales de automóvil

5 CAMPO TÉCNICO

Las realizaciones se refieren en general a automóviles y, más particularmente, al uso óptimo de espejos virtuales en automóviles.

10 ANTECEDENTES DE LA DESCRIPCIÓN

Los espejos encontrados en el exterior e interior de automóviles se usan para ayudar al conductor a ver áreas detrás y a los lados del vehículo. Muchas veces los espejos permiten que el conductor vea objetos que están fuera de la visión periférica del conductor. Hoy, los espejos de vista trasera y laterales pueden sustituirse con espejos virtuales.

15 El documento US 2015/0232030 A1 divulga un sistema de visión de un vehículo que incluye una cámara, una unidad de visualización y un controlador. Una pantalla de visualización se activa y desactiva en respuesta a la determinación de que el conductor del vehículo está mirando al área de visualización.

20 El documento US 7375728 B2 divulga un espejo virtual, representado por un visualizador o pantalla gráfica informática, en donde el visualizador replica sustancialmente las propiedades ópticas de un espejo físico. En particular, muestra visualizar artículos generados por ordenador en un espejo virtual y cambiar la visualización cambiando un radio de curvatura generado por ordenador del espejo virtual.

25 El documento DE 102012020170 A1 divulga un sistema de visualización para visualizar electrónicamente un área trasera de un vehículo. Un brillo y/o contraste del sistema de visualización se reduce en caso de que el dispositivo de visualización no esté en línea directa de visión del conductor del vehículo.

30 El documento EP 2743133 A2 divulga un sistema de cámara para un vehículo que comprende una pantalla de visualización para visualizar una imagen en movimiento, una cámara de vista trasera, un procesador de imágenes y un transmisor de datos. La cámara, así como la pantalla de visualización, tiene una resolución fija que puede no modificarse.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

35 Las diversas ventajas de las realizaciones serán evidentes para un experto en la materia leyendo las siguiente memoria descriptiva y reivindicaciones adjuntas, y haciendo referencia a los siguientes dibujos, en los que:

40 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 componentes de procesador paralelo, de acuerdo con una realización;

45 Las Figuras 3A-3B son diagramas de bloque de multiprocesadores de gráficos, de acuerdo con realizaciones;

Las Figuras 4A-4F ilustran una arquitectura ilustrativa en la que una pluralidad de GPU se acopla comunicativamente a una pluralidad de procesadores de múltiples núcleos;

50 La Figura 5 ilustra una tubería de procesamiento de gráficos, de acuerdo con una realización;

Las Figuras 6A, 6B y 6C ilustran un automóvil convencional con espejos de vista lateral exteriores;

Las Figuras 7A y 7B ilustran un automóvil convencional con un espejo de vista trasera interior;

55 La Figura 8 es un diagrama que ilustra puntos ciegos de conductor y cobertura de visión normal usando espejos de vista lateral y vista trasera convencionales para un automóvil;

La Figura 9A es un diagrama de bloques que ilustra un sistema vehicular ilustrativo para uso de potencia óptimo de espejos virtuales en un automóvil de acuerdo con una realización;

60 La Figura 9B es un diagrama de bloques que ilustra un espejo virtual ilustrativo de acuerdo con una realización;

La Figura 10 es un diagrama de flujo que ilustra un método ilustrativo para uso de potencia óptimo de espejos virtuales en un automóvil de acuerdo con una realización;

65 La Figura 11 es un diagrama de bloques de un ejemplo de un visualizador con una capacidad de luz de fondo localizada

de acuerdo con una realización;

La Figura 12A es un diagrama de bloques de un ejemplo de un dispositivo de procesamiento de datos de acuerdo con una realización;

La Figura 12B es una ilustración de un ejemplo de una determinación de distancia de acuerdo con una realización;

La Figura 13 es un diagrama de bloques de un ejemplo de una arquitectura de visualización en capas de acuerdo con una realización;

La Figura 14 es un diagrama de bloques de un ejemplo de una arquitectura de visualizador que incluye múltiples unidades de visualización de acuerdo con una realización; y

La Figura 15 es un diagrama de bloques de un ejemplo de una arquitectura de entrega de medios asistida en la nube de acuerdo con una realización;

Las Figuras 16-18 son diagramas de bloque de un ejemplo de una vista general de un sistema de procesamiento de datos de acuerdo con una realización;

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

Las Figuras 20-22 son diagramas de bloque de ejemplos de unidades de ejecución de acuerdo con una realización;

La Figura 23 es un diagrama de bloques de un ejemplo de una tubería de gráficos de acuerdo con una realización;

Las Figuras 24A-24B son diagramas de bloque de ejemplos de programación de tubería de gráficos de acuerdo con una realización;

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

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

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

DESCRIPCIÓN DETALLADA

En la siguiente descripción, se exponen numerosos detalles específicos para proporcionar un entendimiento más completo de la presente invención. Sin embargo, será evidente para un experto en la materia que la presente invención puede ponerse en práctica sin uno o más de estos detalles específicos. En otros casos, no se han descrito características bien conocidas para evitar obstaculizar la presente invención.

Vista general del sistema

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

En una realización, el subsistema de procesamiento 101 incluye uno o más procesador o procesadores paralelos 112 acoplados al concentrador de memoria 105 a través de un bus u otro enlace de comunicación 113. El enlace de comunicación 113 puede ser uno de cualquier número de tecnologías o protocolos de enlace de comunicación basados en normas, tales como, pero sin limitación, PCI Express, o puede ser una interfaz de comunicaciones o tejido de comunicaciones específicos de proveedor. En una realización, el uno o más procesador o procesadores paralelos

112 forman un sistema de procesamiento paralelo o vectorial computacionalmente enfocado que incluye un gran número de núcleos de procesamiento y/o agrupaciones de procesamiento, tales como un procesador de muchos núcleos integrados (MIC). En una realización, el uno o más procesador o procesadores paralelos 112 forman un subsistema de procesamiento de gráficos que puede emitir píxeles a uno del uno o más dispositivo o dispositivos de visualización 110A acoplados mediante el concentrador de E/S 107. El uno o más procesador o procesadores paralelos 112 pueden incluir también un controlador de visualización e interfaz de visualización (no mostrados) para posibilitar una conexión directa a uno o más dispositivo o dispositivos de visualización 110B.

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

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

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

Se apreciará que el sistema informático 100 mostrado en el presente documento es ilustrativo y que son posibles variaciones y modificaciones. La topología de conexión, que incluye el número y disposición de los puentes, el número de procesador o procesadores 102 y el número de procesador o procesadores paralelos 112, pueden modificarse según se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 está conectada al procesador o procesadores 102 directamente en lugar de a través de un puente, mientras que otros dispositivos se comunican con la memoria de sistema 104 mediante el concentrador de memoria 105 y el procesador o procesadores 102. En otras topologías alternativas, el procesador o procesadores paralelos 112 están conectados al concentrador de E/S 107 o directamente a uno del uno o más procesador o procesadores 102, en lugar de al concentrador de memoria 105. En otras realizaciones, el concentrador de E/S 107 y el concentrador de memoria 105 pueden estar integrados en un único chip. Algunas realizaciones pueden incluir dos o más conjuntos de procesador o procesadores 102 adjuntos mediante múltiples zócalos, que pueden acoplarse con dos o más instancias del procesador o procesadores paralelos 112.

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

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

En una realización, el procesador paralelo 200 incluye una unidad de procesamiento paralelo 202. La unidad de procesamiento paralelo incluye una unidad de E/S 204 que posibilita la comunicación con otros dispositivos, que

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

Cuando la interfaz de anfitrión 206 recibe una memoria intermedia de comando mediante la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para realizar estos comandos a un primer 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 agrupación de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupación de procesamiento 212 está configurada apropiadamente y en un estado válido antes de que se distribuyan las tareas a las agrupaciones de procesamiento de la matriz de agrupación de procesamiento 212. En una realización, el planificador 210 se implementa a través de lógica de firmware que se ejecuta en un microcontrolador. El planificador implementado en el microcontrolador 210 es configurable para realizar operaciones complejas de planificación y distribución de trabajo en granularidad gruesa y fina, posibilitando conmutación rápida de prioridad y contexto de hilos que se ejecutan en la matriz de procesamiento 212. En una realización, el software anfitrión puede probar cargas de trabajo para planificar en la matriz de procesamiento 212 a través de uno de múltiples timbres de procesamiento de gráfico. Las cargas de trabajo pueden distribuirse, a continuación, automáticamente a través de la matriz de procesamiento 212 por la lógica del planificador 210 dentro del microcontrolador de planificador.

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

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

En una realización, la matriz de agrupación de procesamiento 212 está configurada para realizar operaciones de procesamiento de gráficos paralelas. En las realizaciones en las que el procesador paralelo 200 está configurado para realizar operaciones de procesamiento de gráficos, la matriz de agrupación de procesamiento 212 puede incluir lógica adicional para soportar la ejecución de tales operaciones de procesamiento de gráficos, que incluyen, pero sin limitación, operaciones de lógica de muestreo de textura, así como lógica de teselación y otra lógica de procesamiento de vértice. Adicionalmente, la matriz de agrupación de procesamiento 212 puede estar configurada para ejecutar programas sombreadores relacionados con el procesamiento de gráficos tales como, pero sin limitación, sombreadores de vértices, sombreadores de teselación, sombreadores de geometría y sombreadores de píxeles. La unidad de procesamiento paralelo 202 puede transferir datos de la memoria de sistema mediante la unidad de E/S 204 para su procesamiento. Durante el procesamiento, los datos transferidos pueden almacenarse en una memoria en chip (por ejemplo, la memoria de procesador paralelo 222) durante el procesamiento y, a continuación, volverse a escribir en la memoria de sistema.

En una realización, cuando se usa la unidad de procesamiento paralelo 202 para realizar procesamiento de gráficos, el planificador 210 puede estar configurado para dividir la carga de trabajo de procesamiento en tareas con tamaño aproximadamente igual, para posibilitar mejor la distribución de las operaciones de procesamiento de gráficos a múltiples agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212. En algunas realizaciones, las porciones de la matriz de agrupación de procesamiento 212 pueden estar configuradas para realizar diferentes tipos de procesamiento. Por ejemplo, una primera porción puede estar configurada para realizar el sombreado de vértices y la generación de la topología, una segunda porción puede estar configurada para realizar la teselación y el sombreado de geometría, y una tercera porción puede estar configurada para realizar el sombreado de píxel 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 los datos intermedios se transmitan entre las agrupaciones 214A-214N para su procesamiento adicional.

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

Cada una de la una o más instancias de la unidad de procesamiento paralelo 202 puede acoplarse con la memoria de procesador paralelo 222. La memoria de procesador paralelo 222 puede accederse mediante la barra transversal de memoria 216, que puede recibir solicitudes de memoria de la matriz de agrupación de procesamiento 212, así como de la unidad de E/S 204. La barra transversal de memoria 216 puede acceder a la memoria de procesador paralelo 222 mediante una interfaz de memoria 218. La interfaz de memoria 218 puede incluir múltiples unidades de partición (por ejemplo, la unidad de partición 220A, la unidad de partición 220B, hasta la unidad de partición 220N) cada una de las cuales puede acoplarse a una porción (por ejemplo, unidad de memoria) de la memoria de procesador paralelo 222. En una implementación, el número de unidades de partición 220A-220N está configurado para ser igual al número de unidades de memoria, de tal forma que una primera unidad de división 220A tiene una correspondiente primera unidad de memoria 224A, una segunda unidad de división 220B tiene una correspondiente unidad de memoria 224B, y una ^Nésima unidad de partición 220N tiene una correspondiente ^Nésima unidad de memoria 224N. En otras realizaciones, el número de unidades de partición 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 dinámico (DRAM) o memoria de acceso aleatorio de gráficos, tal como memoria de acceso aleatorio de gráficos síncrona (SGRAM), que incluye memoria de doble tasa de datos de gráficos (GDDR). En una realización, las unidades de memoria 224A-224N pueden incluir también memoria en 3D apilada, que incluye, pero sin limitación, memoria de alto ancho de banda (HBM). Los expertos en la materia apreciarán que la implementación específica de las unidades de memoria 224A-224N puede variar, y que puede seleccionarse de uno de diversos diseños convencionales. Los objetivos de la representación, tales como memorias intermedias de tramas o mapas de textura, pueden almacenarse a través de las unidades de memoria 224A-224N, lo que permite a las unidades de partición 220A-220N escribir porciones de cada objetivo de representación en paralelo para usar eficientemente 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 unificada que utiliza memoria de sistema en conjunto con memoria de caché local.

En una realización, una cualquiera de las agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212 puede procesar datos para que se escriban en una cualquiera de las unidades de memoria 224A-224N dentro de la memoria de procesador paralelo 222. La barra transversal de memoria 216 puede configurarse para transferir la salida de cada agrupación 214A-214N a cualquier unidad de partición 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 diversos dispositivos de memoria externa. En una realización, la barra transversal de memoria 216 tiene una conexión a la interfaz de memoria 218 para comunicarse con la unidad de E/S 204, así como una conexión a una instancia local de la memoria de procesador paralelo 222, posibilitando que las unidades de procesamiento dentro de las diferentes agrupaciones de procesamiento 214A-214N se comuniquen con la memoria de sistema u otra memoria que no sea local a la unidad de procesamiento paralelo 202. En una realización la barra transversal de memoria 216 puede usar canales virtuales para separar flujos de tráfico entre las agrupaciones 214A-214N y las unidades de partición 220A-220N.

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

La **Figura 2B** es un diagrama de bloques de una unidad de partición 220, de acuerdo con una realización. En una realización, la unidad de partición 220 es una instancia de una de las unidades de partición 220A-220N de la Figura

2A. Como se ilustra, la unidad de partición 220 incluye una caché L2 221, una interfaz de memoria intermedia de tramas 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 almacén recibidas de la barra transversal de memoria 216 y la ROP 226. Se emiten pérdidas de lectura y solicitudes de escritura de vuelta urgentes por la caché L2 221 a la interfaz de memoria intermedia de tramas 225 para su procesamiento. También pueden enviarse actualizaciones a la memoria intermedia de tramas mediante la interfaz de memoria intermedia de tramas 225 para su procesamiento. En una realización, la interfaz de memoria intermedia de tramas 225 interconecta con una de las unidades de memoria en la 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 aplicaciones de gráficos, la ROP 226 es una unidad de procesamiento que realiza operaciones de rasterización tales como estarcido, prueba z, mezcla y similares. La ROP 226 emite, a continuación, datos gráficos procesados que se almacenan en la memoria de gráficos. En algunas realizaciones, la ROP 226 incluye lógica de compresión para comprimir datos de profundidad o de color que se escriben en la memoria y descomprimir datos de profundidad o de color que se leen de la memoria. La lógica de compresión puede ser lógica de compresión sin pérdidas que hace uso de uno o más de múltiples algoritmos de compresión. El tipo de compresión que se realiza por la ROP 226 puede variar basándose en las características estadísticas de los datos a comprimir. Por ejemplo, en una realización, se realiza compresión de color delta en datos de profundidad y de color sobre una base de 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 partición 220. En una realización de este tipo, se transmiten las solicitudes de escritura y lectura para datos de píxeles a través de la barra transversal de memoria 216 en lugar de datos de fragmento de píxel. Los datos de gráficos procesados pueden visualizarse en un dispositivo de visualización, tal como uno del uno o más dispositivo o dispositivos de visualización 110 de la Figura 1, encaminarse para su procesamiento adicional por el procesador o procesadores 102, o encaminarse para su procesamiento adicional por una de las entidades de procesamiento dentro del procesador paralelo 200 de la Figura 2A.

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

La operación de la agrupación de procesamiento 214 puede controlarse mediante un gestor de tuberías 232 que distribuye tareas de procesamiento a procesadores paralelos de SIMT. El gestor de tuberías 232 recibe instrucciones del planificador 210 de la Figura 2 y gestiona la ejecución de estas 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 diferentes arquitecturas dentro de la agrupación de procesamiento 214. Puede incluirse una o más instancias del multiprocesador de gráficos 234 dentro de una agrupación de procesamiento 214. El multiprocesador de gráficos 234 puede procesar datos y puede usarse una barra transversal de datos 240 para distribuir los datos procesados a uno de múltiples destinos posibles, que incluyen otras unidades de sombreado. El gestor de tuberías 232 puede facilitar la distribución de datos procesados especificando destinos para datos procesados que van a distribuirse mediante la barra transversal de datos 240.

Cada multiprocesador de gráficos 234 dentro de la agrupación de procesamiento 214 puede incluir un conjunto idéntico de lógica de ejecución funcional (por ejemplo, unidades de lógica aritmética, unidades de carga-almacén, etc.). La lógica de ejecución funcional puede configurarse de una manera en tubería en la que pueden emitirse nuevas instrucciones antes de que se completen instrucciones anteriores. La lógica de ejecución funcional soporta una diversidad de operaciones que incluyen aritmética de enteros y de coma flotante, operaciones de comparación, operaciones booleanas, desplazamiento de bits y cálculo de diversas funciones algebraicas. En una realización, puede aprovecharse 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 ejecuta a través del conjunto de motores de procesamiento paralelo es un grupo de hilos. Un grupo de hilos ejecuta el

mismo programa en diferentes datos de entrada. Cada hilo dentro de un grupo de hilos puede asignarse a un motor de procesamiento diferente dentro de un multiprocesador de gráficos 234. Un grupo de hilos puede incluir menos hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234. Cuando un grupo de hilos incluye menos hilos que el número de motores de procesamiento, uno o más de los motores de procesamiento pueden estar en reposo durante ciclos en los que ese grupo de hilos se está procesando. 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, el procesamiento puede realizarse en ciclos de reloj consecutivos. En una realización, múltiples grupos de hilos pueden ejecutarse concurrentemente en un multiprocesador de gráficos 234.

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

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

En aplicaciones de gráficos e informáticas, una agrupación de procesamiento 214 puede estar configurada de manera que cada multiprocesador de gráficos 234 está 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, de la caché L1 dentro del multiprocesador de gráficos 234 y se extraen de 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 rasterización previa) está configurada para recibir datos del multiprocesador de gráficos 234, dirigir datos a las unidades de ROP, que pueden estar ubicadas con unidades de partición como se describe en el presente documento (por ejemplo, las unidades de partición 220A-220N de la Figura 2). La unidad preROP 242 puede realizar optimizaciones para mezcla de color, organizar datos de color de píxel y realizar traducciones de direcciones.

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

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

En una realización, la caché de instrucciones 252 recibe un flujo de instrucciones para que se ejecuten desde el gestor de tuberías 232. Las instrucciones se almacenan en caché en la caché de instrucciones 252 y se despachan para su ejecución por la unidad de instrucciones 254. La unidad de instrucciones 254 puede despachar instrucciones como

grupos de hilos (por ejemplo, envolturas), 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. La unidad de mapeo de direcciones 256 puede usarse para traducir direcciones en el espacio de direcciones unificado en una dirección de memoria distinta que puede accederse por las unidades de carga/almacén 266.

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

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

En una realización, los núcleos de GPGPU 262 incluyen lógica de SIMD con capacidad de realizar una única instrucción en múltiples conjuntos de datos. En una realización, los núcleos de GPGPU 262 pueden ejecutar físicamente instrucciones de SIMD4, SIMD8 y SIMD16 y ejecutar lógicamente instrucciones de SIMD1, SIMD2 y SIMD32. Las instrucciones de SIMD para los núcleos de GPGPU pueden generarse en tiempo de compilación por un compilador de sombreador o generarse automáticamente cuando se ejecutan programas escritos y compilados para arquitecturas de único programa, múltiples datos (SPMD) o SIMT. Múltiples hilos de un programa configurado para el modelo de ejecución de SIMT pueden ejecutarse a través de una única instrucción de SIMD. Por ejemplo y en una realización, ocho hilos de SIMT que realizan las mismas o similares operaciones pueden ejecutarse en paralelo a través de una única unidad lógica de 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 fichero de registro 258 y a la memoria compartida 270. En una realización, la interconexión de memoria y caché 268 es una interconexión de barra transversal que permite que la unidad de carga/almacén 266 implemente operaciones de carga y almacén entre la memoria compartida 270 y el fichero de registro 258. El fichero de registro 258 puede operar en la misma frecuencia que los núcleos de GPGPU 262, por lo tanto la transferencia de datos entre los núcleos de GPGPU 262 y el fichero de registro 258 es de latencia muy baja. La memoria compartida 270 puede usarse para posibilitar comunicación entre hilos que se ejecutan en las unidades funcionales dentro del multiprocesador de gráficos 324. La memoria caché 272 puede usarse como una caché de datos, por ejemplo, para almacenar en caché datos de textura comunicados entre las unidades funcionales y la unidad de textura 236. La memoria compartida 270 también puede usarse como una caché gestionada por programa. Los hilos que se ejecutan en los núcleos de GPGPU 262 pueden almacenar programáticamente datos dentro de la memoria compartida además de los datos almacenados en caché automáticamente que se almacenan dentro de la memoria caché 272.

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

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

Los diversos componentes pueden comunicarse mediante un tejido de interconexión 327. En una realización, el tejido de interconexión 327 incluye uno o más conmutadores de barra transversal para posibilitar la comunicación entre los

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

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

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

En algunas realizaciones, un procesador paralelo o GPGPU como se describe en el presente documento se acopla comunicativamente a núcleos de anfitrión/procesador para acelerar operaciones gráficas, operaciones de aprendizaje automático, operaciones de análisis de patrón y diversas funciones de GPU de fin general (GPGPU). La GPU puede acoplarse comunicativamente al procesador de anfitrión/núcleos 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 estar integrada en el mismo paquete o chip que los núcleos y estar acoplada comunicativamente a los núcleos a través de un bus/interconexión de procesador interna (es decir, interna al paquete o chip). Independientemente de la manera en la que esté conectada la GPU, los núcleos de procesador pueden asignar trabajo a la GPU en forma de secuencias de comandos/instrucciones contenidas en un descriptor de trabajo. La GPU a continuación usa circuitería/lógica especializada para procesar eficientemente estos comandos/instrucciones.

Técnicas para interconexión de GPU a procesador anfitrión

La Figura 4A ilustra una arquitectura ilustrativa en la que una pluralidad de GPU 410-413 se acopla comunicativamente 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 soportan un caudal de comunicación de 4 GB/s, 30 GB/s, 80 GB/s o mayor, dependiendo de la implementación. Pueden usarse diversos protocolos de interconexión que incluyen, pero sin limitación, PCIe 4.0 o 5.0 y NVLink 2.0. Sin embargo, los principios subyacentes de la invención no están limitados a ningún protocolo o caudal de comunicación particular.

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

En una realización, cada procesador de múltiples núcleos 405-406 está acoplado comunicativamente a una memoria de procesador 401-402, mediante interconexiones de memoria 430-431, respectivamente, y cada GPU 410-413 está acoplada comunicativamente a memoria de GPU 420-423 a través de 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 de memoria. A modo de ejemplo, y no como limitación, las memorias de procesador 401-402 y las memorias de GPU 420-423 pueden ser memorias volátiles, tales como memorias de acceso aleatorio dinámicas (DRAM) (que incluyen DRAM apiladas), SDRAM DDR de gráficos (GDDR) (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, alguna porción de las memorias puede ser memoria volátil y otra porción puede ser memoria no volátil (por ejemplo, usando una jerarquía de memoria de dos niveles (2LM)).

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

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

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

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

En una realización, un circuito de intermediario 425 acopla comunicativamente el módulo de aceleración de gráficos 446 al bus de coherencia 464, lo que permite que el módulo de aceleración de gráficos 446 participe en el protocolo de coherencia de caché como un par de los núcleos. En particular, una interfaz 435 proporciona conectividad al circuito de intermediario 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 440.

En una implementación, un circuito de integración del acelerador 436 proporciona la gestión de caché, el acceso a memoria, la gestión de contexto y los servicios de gestión de interrupción en nombre 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 puede cada uno comprender una unidad de procesamiento de gráficos (GPU) separada. Como alternativa, los motores de procesamiento de gráficos 431, 432, N pueden comprender diferentes tipos de motores de procesamiento de gráficos dentro de una GPU, tal como las unidades de ejecución de gráficos, los motores de procesamiento de medios (por ejemplo, los 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.

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

procesador 462A-462D, 456 y recibiendo actualizaciones de la caché 438).

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 grabación y restauración para grabar y restaurar contextos de los diversos hilos durante conmutaciones de contexto (por ejemplo, cuando se graba el primer hilo y se almacena un segundo hilo de modo que el segundo hilo puede ejecutarse por un motor de procesamiento de gráficos). Por ejemplo, en una conmutación de contexto, el circuito de gestión de contexto 448 puede almacenar valores de registro actuales para una región designada en memoria (por ejemplo, identificada por un puntero de contexto). A continuación, puede restaurar los valores de registro al volver al contexto. En una realización, un circuito de gestión de interrupción 447 recibe y procesa interrupciones recibidas de dispositivos de sistema.

En una implementación, las direcciones virtuales/efectivas de un motor de procesamiento de gráficos 431 se traducen a direcciones reales/físicas en la memoria de sistema 411 por la MMU 439. Una realización del circuito de integración del acelerador 436 soporta múltiples módulos del acelerador de gráficos 446 (por ejemplo, 4, 8, 16) y/u otros dispositivos aceleradores. El módulo del acelerador de gráficos 446 puede estar especializado a una única aplicación ejecutada en el procesador 407 o puede compartirse entre múltiples aplicaciones. En una realización, se presenta un entorno de ejecución de gráficos virtualizado en el que se comparten los recursos de los motores de procesamiento de gráficos 431-432, N 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 prioridades asociadas con las VM y/o las aplicaciones.

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

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

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

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

La **Figura 4C** ilustra otra realización en la que el circuito de integración del 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 del 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 del acelerador 436 puede realizar las mismas operaciones que aquellas descritas con respecto a la **Figura 4B**, pero potencialmente a un caudal superior dado su proximidad cercana al bus de coherencia 462 y a las cachés 462A-462D, 426.

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

En una realización del modelo de proceso especializado, los motores de procesamiento de gráficos 431-432, N están especializados 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, que proporcionan virtualización dentro de una VM/partición.

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

Para el modelo de programación compartida, el módulo de aceleración de gráficos 446 o un motor de procesamiento de gráficos individual 431-432, N selecciona un elemento de proceso usando un manejador de proceso. En una realización, los elementos de proceso se almacenan en memoria de sistema 411 y son direccionables usando técnicas de traducción 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, el software del sistema solicitante para añadir el elemento de proceso a la lista de elementos de proceso vinculados). Los 16 bits más bajos del manejador de proceso pueden ser el desplazamiento del elemento de proceso dentro de la lista de elementos de proceso vinculados.

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

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

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

En la operación, una unidad de extracción de WD 491 en el corte de integración del acelerador 490 extrae el siguiente WD 484 que incluye una indicación del trabajo que va a hacerse por uno de los motores de procesamiento de gráficos del módulo de aceleración de gráficos 446. Los datos del WD 484 pueden almacenarse en registros 445 y usarse por la MMU 439, el circuito de gestión de interrupción 447 y/o el circuito de gestión de contexto 446 como se ilustra. Por ejemplo, una realización de la MMU 439 incluye circuitería de paso de segmentos/página para acceder a segmentos/tablas de página 486 dentro del espacio de direcciones virtual del SO 485. El circuito de gestión de interrupciones 447 puede procesar eventos de interrupción 492 recibidos del módulo de aceleración de gráficos 446. Cuando se realizan operaciones de gráficos, una dirección efectiva 493 generada por un motor de procesamiento de gráficos 431-432, N se traduce a una dirección real por la MMU 439.

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

Tabla 1 - Registros de hipervisor inicializados

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

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

Tabla 2 - Registros de sistema operativo inicializados

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

En una realización, cada WD 484 es específico para un módulo de aceleración de gráficos particular 446 y/o motor de procesamiento de gráficos 431-432, N. 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 donde la aplicación ha configurado una cola de comandos de trabajo para que se complete.

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

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

En este modelo, el hipervisor de sistema 496 posee el módulo de aceleración de gráficos 446 y hace disponibles sus funciones a todos los sistemas operativos 495. Para que un módulo de aceleración de gráficos 446 soporte la virtualización por el sistema hipervisor 496, el módulo de aceleración de gráficos 446 puede adherirse a los siguientes requisitos: 1) Una solicitud de trabajo de la aplicación debe ser autónoma (es decir, el estado no necesita que se mantenga entre trabajos), o el módulo de aceleración de gráficos 446 debe proporcionar un mecanismo de grabación y restauración de contexto. 2) Se garantiza que se completa una solicitud de trabajo de la aplicación por el módulo de aceleración de gráficos 446 en una cantidad especificada de tiempo, que incluye cualquier fallo de traducción, o el módulo de aceleración de gráficos 446 proporciona la capacidad de anticiparse al procesamiento del trabajo. 3) El módulo de aceleración de gráficos 446 debe garantizar equidad entre procesos cuando opera en el modelo de programación compartido dirigido.

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

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

Tabla 3 - parámetros de llamada de SO a Hipervisor

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

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

Tabla 4 - Información de elemento de proceso

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

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

Como se ilustra en la **Figura 4F**, una realización de la invención emplea una memoria unificada direccionable mediante un espacio de direcciones de memoria virtual común usado para acceder a las memorias de procesador físicas 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 los procesadores 401-402 y viceversa, simplificando de esta manera la capacidad de la programación. En una realización, una primera porción del espacio de direcciones virtual/efectivo está asignada a la memoria del procesador 401, una segunda porción a la segunda memoria del procesador 402, una tercera porción a la memoria de la GPU 420, y así sucesivamente. El espacio de memoria virtual/efectivo completo (en ocasiones denominado espacio de direcciones efectivo) se distribuye de esta manera a través de cada una de las memorias de procesador 401-402 y las memorias de GPU 420-423, permitiendo que cualquier procesador o GPU acceda a cualquier memoria física con una dirección virtual mapeada a esa memoria.

En una realización, la circuitería de gestión de desvío/coherencia 494A-494E dentro de una o más de las MMU 439A-439E garantiza la coherencia de caché entre las cachés de los procesadores de anfitrión (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 para desvío/coherencia puede implementarse dentro de la MMU de uno o más procesadores de anfitrión 405 y/o dentro del circuito de integración del acelerador 436.

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

efectividad de la descarga de la GPU.

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

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

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

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

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

Tubería de procesamiento de gráficos

La Figura 5 ilustra una tubería de procesamiento de gráficos 500, de acuerdo con una realización. En una realización, un procesador de gráficos puede implementar la tubería de procesamiento de gráficos 500 ilustrada. El procesador de gráficos puede estar incluido dentro de los subsistemas de procesamiento paralelos como se describe en el presente documento, tal como el procesador paralelo 200 de la Figura 2, que, en una realización, es una variante del procesador o procesadores paralelos 112 de la Figura 1. Los diversos sistemas de procesamiento paralelos pueden implementar la tubería 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 de sombreado (por ejemplo, el multiprocesador de gráficos 234 de la Figura 3) puede estar configurada para realizar las funciones de una o más de una unidad de procesamiento de vértice 504, una unidad de proceso 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 fragmento/píxel 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 pueden realizarse también por otros motores de procesamiento dentro de una agrupación de procesamiento (por ejemplo, la agrupación de procesamiento 214 de la Figura 3) y una correspondiente unidad de partición (por ejemplo, la unidad de partición 220A-220N de la Figura 2). La tubería de procesamiento de gráficos 500 puede implementarse usando unidades de procesamiento especializadas para una o más funciones. En una realización, puede realizarse una o más porciones de la tubería de procesamiento de gráficos 500 por una lógica de procesamiento paralela dentro de un procesador de fin general (por ejemplo, la CPU). En una realización, una o más porciones de la tubería de procesamiento de gráficos 500 pueden acceder a 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 vértice para superficies y primitivas. El ensamblador de datos 502 emite, a continuación, los datos de vértice, incluyendo los atributos de vértice, a la unidad de procesamiento de vértice 504. La unidad de procesamiento de vértice 504 es una unidad de ejecución programable que ejecuta programas de sombreador de vértices, iluminando y transformando

datos de vértice como se especifica por los programas de sombreador de vértices. La unidad de procesamiento de vértice 504 lee datos que se almacenan en caché, memoria local o de sistema para su uso en el procesamiento de los datos de vértice y puede programarse para transformar los datos de vértice desde una representación de coordenadas basada en objeto a un espacio de coordenadas de espacio global o un espacio de coordenadas de dispositivo normalizado.

Una primera instancia de un ensamblador de primitivas 506 recibe atributos de vértice desde la unidad de procesamiento de vértice 504. Las lecturas del ensamblador de primitivas 506 almacenaron atributos de vértice según se necesite y construye primitivas gráficas para su procesamiento por la unidad de procesamiento de control de teselación 508. Las primitivas gráficas incluyen triángulos, segmentos de línea, puntos, parches y así sucesivamente, según se soportan por diversas interfaces de programación de aplicación (API) de procesamiento gráfico.

La unidad de procesamiento de control de teselación 508 trata los vértices introducidos como puntos de control para un parche geométrico. Los puntos de control se transforman desde una representación introducida desde el parche (por ejemplo, las bases del parche) a una representación que es adecuada para su uso en evaluación de 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 cuantifica un nivel de vista dependiente de detalle asociado con el borde. Una unidad de teselación 510 está configurada para recibir los factores de teselación para bordes de un parche y para teselar el parche en múltiples primitivas geométricas, tales como primitivas de línea, triángulo o cuadrilátero, que se transmiten a una unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de evaluación de teselación 512 opera en coordenadas parametrizadas del parche subdividido para generar una representación superficial y atributos de vértice para cada vértice asociado con las primitivas geométricas.

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

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

El rasterizador 522 puede realizar selección de profundidad y otras optimizaciones basadas en profundidad. El rasterizador 522 también realiza conversión de exploración en las nuevas primitivas geométricas para generar fragmentos y emitir esos fragmentos y datos de cobertura asociados a la unidad de procesamiento de fragmento/píxel 524. La unidad de procesamiento de fragmento/píxel 524 es una unidad de ejecución programable que está configurada para ejecutar programas sombreadores de fragmentos o programas sombreadores de píxeles. La unidad de procesamiento de fragmento/píxel 524 que transforma fragmentos o píxeles recibidos desde el rasterizador 522, según se especifica por los programas sombreadores de fragmentos o de píxeles. Por ejemplo, la unidad de procesamiento de fragmento/píxel 524 puede programarse para realizar operaciones incluidas, pero sin limitación a, mapeo de texturas, sombreado, combinación, corrección de textura y corrección de perspectiva para producir fragmentos o píxeles sombreados que se emiten a una unidad de operaciones de rasterización 526. La unidad de procesamiento de fragmento/píxel 524 puede leer datos que se almacenan en o bien la memoria de procesador paralelo o bien la memoria de sistema para su uso cuando procesa los datos de fragmento. Los programas de sombreador de fragmento o píxeles pueden configurarse para sombrear en muestra, píxel, mosaico u otras granularidades dependiendo de la tasa de muestreo configurada para las unidades de procesamiento.

La unidad de operaciones de rasterización 526 es una unidad de procesamiento que realiza operaciones de ráster que incluye, pero sin limitación a, estarcido, prueba z, mezcla y similares, y emite datos de píxel como datos de gráficos procesados que hay que almacenar en memoria gráfica (por ejemplo, memoria de procesador paralelo 222 como en la Figura 2, y/o memoria de sistema 104 como en la Figura 1, que hay que visualizar en el uno o más dispositivo o dispositivos de visualización 110 o para su procesamiento adicional por uno del uno o más procesador o procesadores 102 o procesador 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 la memoria y descomprimir datos z o de color que se leen de la memoria.

Uso óptimo de espejos virtuales de automóvil

En los automóviles, los espejos se usan para ayudar al conductor a ver áreas por detrás del automóvil y a los lados derechos e izquierdos del automóvil. Los automóviles habitualmente tienen dos espejos de vista lateral en el exterior del coche, uno en la izquierda y uno en la derecha, y un espejo de vista trasera en el interior del coche. La Figura 6A ilustra un automóvil 600 con espejos de vista lateral exteriores 602 y 604. El espejo de vista lateral 602 se ubica en el lado izquierdo del coche. El espejo de vista lateral 604 se ubica en el lado derecho del coche. Las Figuras 6B y 6C son ejemplos de lo que un conductor puede ver 606 cuando él o ella mira a través de los espejos de vista lateral. Como se muestra en las Figuras 6B y 6C, el conductor es capaz de ver áreas por detrás y al lado 606 del automóvil cuando él o ella mira a través de los espejos de vista lateral 602 y 604.

La Figura 7A ilustra una vista interior 700 de un automóvil. La vista interior 700 incluye un espejo de vista trasera 702 para permitir que un conductor vea áreas por detrás del automóvil. La Figura 7B ilustra un ejemplo de lo que un conductor puede ver 704 cuando él o ella mira a través del espejo de vista trasera. Como se muestra, el conductor es capaz de ver áreas por detrás del automóvil.

La Figura 8 ilustra una situación de conducción 800 de ejemplo en la que se muestran la cobertura de visión normal de un conductor desde los espejos de vista lateral y de vista trasera y los puntos ciegos de un conductor durante la operación de un automóvil. Un automóvil 802 operado por un conductor está viajando por una autopista 804 rodeado por otros automóviles 806, 808, 810, 812, 814, 816 y 818. Como se muestra en la Figura 8, hay cuatro automóviles 812, 814, 816 y 818 dentro de la vista del conductor de automóvil 802, y tres de los cuatro automóviles 814, 816 y 818 están en frente del automóvil 802. El único automóvil en la parte trasera del automóvil 802 que es visible para el conductor de automóvil 802 es el automóvil 812. El conductor de automóvil 802 tiene una vista clara del automóvil 812 usando el espejo de vista trasera 702. Como se muestra en la Figura 8, el automóvil 812 reside dentro la de cobertura de visualización 820 del automóvil 802 como se ve a través de los espejos de vista trasera y de vista lateral del automóvil 802. Los automóviles 806, 808 y 810 no son visibles para el conductor del automóvil 802 y, por lo tanto, se consideran que están en un área referida como un punto ciego 822 del automóvil 802. Si el automóvil 802 fuera a intentar cambiar de carril moviéndose al carril de la derecha o al carril de la izquierda, podría producirse un potencial accidente porque los automóviles 806 y 808 no son visibles al conductor del automóvil 802.

Existe una tendencia de sustituir los espejos de vista lateral y de vista trasera en un automóvil por espejos virtuales. Los espejos virtuales son cámaras acopladas comunicativamente a visualizadores. La sustitución de espejos por cámaras y visualizadores proporciona a un conductor de un automóvil una cobertura de visión más amplia, disminuyendo, si no eliminando, de este modo las áreas de punto ciego. Los espejos virtuales utilizan una gran cantidad de potencia para codificar y decodificar la imagen que hay que visualizar. También, cuanto mayor sea la resolución y mayor sea la tasa de fotogramas usada para el visualizador, mayor será la potencia consumida. A continuación, se describen realizaciones en las que se utilizan espejos virtuales de forma óptima para ahorrar potencia.

La Figura 9A es un diagrama de bloques que ilustra un sistema vehicular 900 para un uso de potencia óptimo de espejos virtuales 904 en un automóvil de acuerdo con una realización. El sistema 900 incluye un detector de la mirada 902, y una pluralidad de espejos virtuales 904 colocados estratégicamente en el interior y el exterior de un automóvil en lugar de los espejos de vista trasera y vista lateral. Los espejos virtuales 904 también pueden colocarse en el automóvil para eliminar puntos ciegos.

El detector de la mirada 902 es un dispositivo de sensor que mide la posición del ojo y el movimiento del ojo para determinar el punto de mirada o dónde se está mirando. El detector de la mirada 902 también detecta el movimiento de la cabeza para determinar dónde se está mirando. En una realización, el detector de la mirada 902 puede tener un procesador integrado y memoria integrada para rastrear dónde está mirando un conductor. En otra realización, el detector de la mirada 902 puede acoplarse a uno o más procesadores en un automóvil, el uno o más procesadores acoplados a uno o más dispositivos de memoria para hacer que el detector de la mirada 902 rastree dónde se está mirando. En las realizaciones, el detector de la mirada 902 se usa para determinar si un conductor de un automóvil está viendo uno de los espejos virtuales 904 en o que rodean a un automóvil.

Los espejos virtuales 904 son espejos de sustitución para un espejo de vista trasera y espejos de vista lateral en un automóvil. Los espejos virtuales 904 se usan para ayudar al conductor de un automóvil a ver áreas por detrás del automóvil y áreas a los lados izquierdo y derecho del automóvil. Como se ha indicado anteriormente, cada espejo virtual 904 comprende una cámara 906 acoplada comunicativamente a un visualizador 908. Un codificador 910 y un decodificador 912 se acoplan entre la cámara y el visualizador para codificar y decodificar la imagen de la cámara para su visualización en el visualizador. El detector de la mirada 902 se acopla a cada espejo virtual 904.

En otra realización, cada espejo virtual 904 puede tener su propio detector de la mirada 902. La Figura 9B es un diagrama de bloques que ilustra un espejo virtual ilustrativo que tiene un detector de la mirada de acuerdo con una realización. El detector de la mirada 902 se acopla a la cámara 906 y el visualizador 908. La Figura 9B también muestra que el detector de la mirada 902 puede acoplarse opcionalmente al codificador 910. El detector de la mirada 902 puede controlar la cámara 906, el visualizador 908 y el codificador 910 enviando una señal para situar los dispositivos en un modo de baja potencia o situar los dispositivos en un modo de operación completo como se describe a

continuación.

Teniendo en cuenta que un conductor puede ver únicamente un espejo virtual 904 a la vez, el detector de la mirada 902 detecta si el conductor está mirando a un espejo virtual 904 particular en un momento dado. Únicamente cuando el conductor está mirando a un espejo virtual 904 el espejo virtual 904 capturará y visualizará en tiempo real la imagen a resolución completa y tasa de fotogramas completa o modo de operación completo. Si un espejo virtual 904 particular no se está viendo de acuerdo con el detector de la mirada 902, la calidad de la imagen en tiempo real capturada por el espejo virtual 904 puede reducirse para ahorrar potencia. Cuando el conductor no está mirando a un espejo virtual 904, puede reducirse la resolución y/o tasa de fotogramas del visualizador 908 para el espejo virtual 904. Por ejemplo, si un conductor está mirando hacia la ubicación de un espejo de vista trasera, el espejo virtual 904 que sustituyó al espejo de vista trasera se operará a resolución completa y tasa de fotogramas completa, mientras todos los otros espejos virtuales 904 se operarán en un modo de baja potencia. Por ejemplo, 60 fotogramas por segundo es una tasa de fotogramas normal usada en un visualizador y una resolución de alta definición puede ser, por ejemplo, 1920 x 1080 píxeles. Cuando un conductor no está viendo un espejo virtual 904, puede reducirse la tasa de fotogramas y/o resolución del visualizador 908. En una realización, la tasa de fotogramas puede reducirse por un factor de cuatro, por lo tanto, reduciendo la tasa de fotogramas a 15 fotogramas por segundo. En una realización, la resolución puede reducirse por un factor de 4, llevando la resolución a 960 x 540 píxeles. Por lo tanto, reducir la tasa de fotogramas y/o resolución del visualizador por al menos un factor de cuatro ahorra una gran cantidad de potencia. En aún otra realización, la resolución puede reducirse hasta un factor de dieciséis, llevando la resolución a 480 x 270 píxeles.

El modo de operación bajo no significa únicamente una tasa de fotogramas y/o resolución inferiores para el visualizador 908. En una realización, puede reducirse el brillo de los píxeles en un visualizador 908. En una realización, la resolución y/o tasa de fotogramas de la cámara 906 en el espejo virtual 904 también pueden reducirse para ahorrar potencia. En una realización, la resolución y/o tasa de fotogramas del codificador 910 pueden reducirse para ahorrar potencia. Los espejos virtuales no vistos 904 deberían permanecer en un estado de potencia bajo y no apagarse (a no ser que el automóvil se apague). Los conductores están viendo los espejos virtuales 904 constantemente mientras conducen y a menudo cambian su mirada desde un espejo virtual 904 a otro muy rápidamente. Apagar cualquier espejo virtual 904 mientras se conduce podría hacer un daño irreparable si el espejo virtual 904 tardase demasiado en encenderse.

La Figura 10 es un diagrama de flujo que ilustra un método 1000 para uso de potencia óptimo de espejos virtuales 904 en un automóvil de acuerdo con una realización. El proceso comienza en el bloque 1002 en el que inmediatamente continúa al bloque 1004.

En el bloque 1004, tras el arranque del automóvil, todos los espejos virtuales 904 se sitúan en modo de operación completo. En una realización, el modo de operación completo puede referirse a una tasa de fotogramas de 60 fotogramas por segundo y una resolución de 1920 x 1080 píxeles para el visualizador 908. En otra realización, el modo de operación completo puede referirse a una tasa de fotogramas de 30 fotogramas por segundo y una resolución de 1920 x 1080 píxeles para el visualizador 908. También pueden usarse valores similares para la tasa de fotogramas y la resolución de la cámara 906 y el codificador 910. El proceso continúa, a continuación, al bloque 1006.

En el bloque 1006, usando el detector de la mirada 902, se supervisa al conductor para determinar si él o ella está mirando hacia un espejo virtual 904. El proceso continúa al bloque de decisión 1008.

En el bloque de decisión 1008, se determina si el conductor está mirando en la dirección de un espejo virtual 904. Si el conductor no está mirando en la dirección de un espejo virtual 904, el proceso continúa al bloque 1012 en el que todos los espejos virtuales 904 se sitúan en un modo de operación bajo. Como se ha indicado anteriormente, un modo de operación bajo puede ser una reducción de cuatro veces del modo de operación completo tanto en tasa de fotogramas como resolución del visualizador 908 así como una reducción en el brillo del visualizador 908. En una realización, también pueden reducirse la resolución de codificación y la tasa de fotogramas. También pueden reducirse la resolución y la tasa de fotogramas de la cámara 906. El proceso continúa, a continuación, de vuelta al bloque 1006 para continuar supervisando al conductor para determinar si el conductor está viendo un espejo virtual 904.

Volviendo al bloque de decisión 1008, si se determina que el conductor está mirando en la dirección de un espejo virtual 904, el espejo virtual 904 que se está viendo se sitúa en modo de operación completo mientras todos los otros espejos virtuales 904 se sitúan en un modo de operación bajo. En una realización, el modo de operación completo requiere que todos los dispositivos en el espejo virtual 904 se sitúen de vuelta a resolución completa y tasa de fotogramas. Esto puede incluir la cámara 906, el codificador 910 y/o el visualizador 908. El proceso continúa, a continuación, de vuelta al bloque 1006 para continuar supervisando al conductor para determinar si el conductor está mirando en la dirección de un espejo virtual 904. Este proceso de supervisar al conductor y ajustar los espejos virtuales 904 por consiguiente continúa hasta que se apaga el automóvil.

Tecnología de visualización

Pasando ahora a la Figura 11, se muestra un sistema informático con rendimiento mejorado 1100. En el ejemplo ilustrado, un procesador 1110 se acopla a un visualizador 1120. El procesador 1110 puede generar generalmente imágenes que hay que visualizarse en un panel de LCD 1150 del visualizador 1120. En un ejemplo, el procesador

1110 incluye una interfaz de comunicación tal como, por ejemplo, una matriz de gráficos de vídeo, una interfaz de DisplayPort (DP), una interfaz de DisplayPort integrado (eDP), una interfaz multimedia de alta definición (HDMI), una interfaz visual digital (DVI) y así sucesivamente. El procesador 1110 puede ser un procesador de gráficos (por ejemplo, una unidad de procesamiento de gráficos/GPU) que procesa datos de gráficos y genera las imágenes (por ejemplo, fotogramas de vídeo, imágenes fijas) visualizadas en el panel de LCD 1150. Además, el procesador 1110 puede incluir una o más tuberías de procesamiento de imágenes que generan datos de píxel. Las tuberías de procesamiento de imágenes pueden cumplir con la arquitectura de OPENGL, u otra arquitectura adecuada. Adicionalmente, el procesador 1110 puede conectarse a un procesador anfitrión (por ejemplo, central unidad de procesamiento/CPU), en donde el procesador anfitrión ejecuta uno o más controladores de dispositivo que controlan y/o interactúan con el procesador 1110.

El visualizador ilustrado 1120 incluye un controlador de temporización (TCON) 1130, que puede abordar individualmente diferentes píxeles en el panel de LCD 1150 y actualizar cada píxel individual en el panel de LCD 1150 por ciclo de refresco. En este sentido, el panel de LCD 1150 puede incluir una pluralidad de elementos de cristal líquido tales como, por ejemplo, un filtro de cristal líquido y color integrado. Cada píxel del panel de LCD 1150 puede incluir un trío de elementos de cristal líquido con filtros de color rojo, verde y azul, respectivamente. El panel de LCD 1150 puede disponer los píxeles en una matriz bidimensional (2D) que se controla a través de controladores de fila 1152 y controladores de columna 1154 para actualizar la imagen que se está visualizando por el panel de LCD 1150. Por lo tanto, el TCON 1130 puede controlar los controladores de fila 1152 y los controladores de columna 1154 para abordar píxeles específicos del panel de 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 visualizado en la superficie del panel de LCD 1150.

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

Los elementos emisores de luz también pueden incluir una o más láminas de material electroluminiscente situada detrás del panel de LCD 1150. En un caso de este tipo, la luz desde la superficie de la lámina puede dispersarse a través de los píxeles del panel de 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 únicamente una porción del panel de LCD 1150. También pueden usarse otras soluciones de retroiluminación.

El visualizador 1120 ilustrado también incluye un controlador de luz de fondo (BLC) 1140 que proporciona una tensión a los elementos emisores de luz de la luz de fondo 1160. Por ejemplo, el BLC 1140 puede incluir un controlador de modulación de anchura de pulso (PWM) (no mostrado) para generar una señal de PWM que activa al menos una porción de los elementos emisores de luz de la luz de fondo 1160. El ciclo de trabajo y la frecuencia de la señal de PWM puede hacer que la luz generada por los elementos emisores de luz se atenúe. Por ejemplo, un ciclo de trabajo del 100 % puede corresponder a los elementos emisores de luz estando totalmente encendidos y un ciclo de trabajo del 0 % puede corresponder a los elementos emisores de luz estando totalmente apagados. Por lo tanto, los ciclos de trabajo intermedios (por ejemplo, 25 %, 50 %) hacen habitualmente que los elementos emisores de luz se enciendan para una porción de un periodo de ciclo que es proporcional al porcentaje del ciclo de trabajo. El periodo de ciclo puede ser lo suficientemente rápido para que la intermitencia de los elementos emisores de luz no sea perceptible al ojo humano. Además, el efecto al usuario puede ser que el nivel de la luz emitida por la luz de fondo 1160 es menor que si la luz de fondo 1160 estuviera totalmente activada. El BLC 1140 puede estar separado de o incorporado en el TCON 1130.

Como alternativa, puede usarse un sistema de visualización de emisión en el que el panel de LCD 1150 se sustituiría por un panel de visualización de emisión (por ejemplo, diodo de emisión de luz orgánico/OLED), la luz de fondo 1160 se omitiría, y los controladores de fila y columna 1152 y 1154, respectivamente, pueden usarse para modular directamente color y brillo de píxel.

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, un ordenador de tableta, una tableta convertible, un dispositivo de internet móvil (MID), un asistente digital personal (PDA), un dispositivo ponible (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 anfitrión, procesador de gráficos) acoplado a una memoria 1222, que puede incluir ubicaciones de almacenamiento que son direccionables a través del procesador 1224. Como se analizará en mayor detalle, un sensor de distancia 1210 puede posibilitar resolución de visualización basada en distancia con respecto a las unidades de visualización 1228.

La memoria 1222 ilustrada incluye datos de visualización 1226 que tienen que representarse en la unidad de visualización 1228. En un ejemplo, el procesador 1224 lleva a cabo 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 posprocesamiento 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 posprocesamiento 1214 puede modificar los datos de visualización 1226 para mejorar la legibilidad de contenido de pantalla en la unidad de visualización 1228, reducir consumo de potencia en el dispositivo de procesamiento de datos 1200, etc., o cualquier combinación de los mismos.

La memoria ilustrada 1222 almacena un ajuste de resolución de visualización 1216, además de un sistema operativo 1212 y una aplicación 1220. El ajuste de resolución de visualización 1216 puede especificar un número de píxeles de los datos de visualización 1226 que hay que presentar en la unidad de visualización 1228 a lo largo de una dimensión de longitud y una dimensión de anchura. Si los datos de visualización 1226 como se generan 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 coincidir con el formato de las unidades de visualización 1228. En este sentido, el ajuste de resolución de visualización 1216 puede asociarse con y/o incorporarse en los datos de configuración que definen otros ajustes para la unidad de visualización 1228. Además, el ajuste de resolución de visualización 1216 puede definirse en términos de distancia o área de unidad (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 el ajuste de resolución de visualización 1216 de entre una o más opciones proporcionadas a través de la interfaz de usuario, introducir el ajuste de resolución de visualización 1216 como un valor solicitado y así sucesivamente. Por lo tanto, los datos de visualización 1226 pueden redimensionarse para ajustarse al ajuste de resolución de visualización 1216 antes de representarse 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 desencadenarse 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 encendido del dispositivo de procesamiento de datos 1200, el sistema operativo 1212 puede llevar a cabo un proceso automático para desencadenar la detección de distancia en el primer o segundo plano. La detección de distancia puede llevarse a cabo de forma periódica o continua.

La Figura 12B muestra un ejemplo de un escenario de detección de distancia. En el ejemplo ilustrado, el sensor de distancia 1210 usa un transceptor 1208 para emitir un haz electromagnético 1202 en la dirección del usuario 1218. Por lo tanto, el transceptor 1202 podría colocarse en una superficie orientada hacia delante del dispositivo de procesamiento de datos 1200 (Figura 12A). El haz electromagnético 1202 puede impactar al usuario 1218 y reflejarse/dispersarse desde el usuario 1218 como un haz electromagnético de retorno 1204. El haz electromagnético de retorno 1204 puede analizarse por, por ejemplo, el procesador 1224 (Figura 12A) y/o el motor de posprocesamiento 1214 (Figura 12A) para determinar la distancia 1206 entre el usuario 1218 y la unidad de visualización 1228 (Figura 12A). La distancia 1206 puede usarse para ajustar el ajuste de resolución de visualización 1216.

Capas de visualización

Pasando 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 conseguir superresolución espacial/temporal en un conjunto de visualización 1360. En el ejemplo ilustrado, un procesador 1310 proporciona datos de gráficos originales 1334 (por ejemplo, fotogramas de vídeo, imágenes fijas), al sistema 1300 a través de un bus 1320. Un programa de visualización en cascada 1331 puede almacenarse en una memoria 1330, en donde el programa de visualización en cascada 1331 puede ser parte de un controlador de visualización asociado con el conjunto de visualización 1360. La memoria ilustrada 1330 también incluye los datos de gráficos originales 1334 y 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 cálculo de factorización temporal y el componente de factorización espacial puede realizar cálculo de factorización espacial. El programa de visualización en cascada 1331 puede derivar los datos de gráficos factorizados 1335 para presentación en cada capa de visualización 1361, 1362 y 1363 basándose en configuraciones de usuarios y los datos de gráficos originales 1334.

El conjunto de visualización 1360 puede implementarse como una LCD (pantalla de cristal líquido) usado en, por ejemplo, una aplicación de pantalla montada en la cabeza (HMD). Más particularmente, el conjunto de visualización 1360 puede incluir una pila de paneles de LCD, placas de interfaz, una instalación de lente, y así sucesivamente. Cada

panel puede operarse a una resolución nativa de, por ejemplo, 1280x800 píxeles y con una tasa de refresco de 60 Hz. Pueden usarse otras resoluciones nativas, tasas de refresco, tecnología de panel de visualización y/o configuraciones de capa.

5 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 pueden usarse generalmente para emitir 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 datos de perfil de hardware 1402 recibidos desde el conjunto de unidades de visualización 1430 a través de una red 1420. La aplicación de la función lógica 1424 a los datos de perfil de hardware 1402 puede crear un conjunto de ajustes de topología automáticos 1406 cuando no se encuentra una coincidencia de los datos de perfil de hardware con un conjunto de ajustes en una tabla de consulta de perfiles de hardware 1412. El conjunto ilustrado de ajustes de topología automáticos 1406 se transmiten 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 tras la recepción de la función lógica 1424 desde un controlador de visualización 1410. En este sentido, el controlador de visualización 1410 puede incluir un módulo de topología automático 1408 que automáticamente configura y estructura 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 se ejecutan por el procesador 1415, hacen que el dispositivo de procesamiento de datos 1418 se comunique con las unidades de visualización 1430, tarjetas de vídeo, etc., y llevar a cabo operaciones de generación de topología automática.

El dispositivo de procesamiento de datos 1418 puede incluir, por ejemplo, un servidor, sobremesa, ordenador portátil, ordenador de tableta, tableta convertible, MID, PDA, dispositivo ponible, reproductor de medios y así sucesivamente. Por lo 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 de controlador que incluyen una o más tarjetas de controlador de vídeo y así sucesivamente. En un ejemplo, las unidades de visualización 1430 son visualizadores de panel plano (por ejemplo, cristal líquido, matriz activa, plasma, etc.), HMD, dispositivos de proyección de vídeo y así sucesivamente, que coordinan entre sí para producir la presentación 1440. Además, la presentación 1440 puede generarse basándose en un archivo de medios almacenado en el dispositivo de almacenamiento 1414, en donde el archivo de medios podría incluir, por ejemplo, una película, clip de vídeo, animación, publicidad, etc., o cualquier combinación de los mismos.

El término "topología" puede considerarse el número, escalado, 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 y así sucesivamente. Por consiguiente, la topología de las unidades de visualización 1430 pueden posibilitar que la presentación 1440 se presente visualmente en concierto de tal forma que las secciones individuales de la presentación 1440 son proporcionales y compatibles con las dimensiones originales y alcance de los medios que se reproducen a través de las unidades de visualización 1430. Por lo tanto, la topología puede constituir relaciones espaciales y/o propiedades geométricas que no se vean impactadas por el cambio continuo de la forma o tamaño del contenido representado en la presentación 1440. En un ejemplo, el módulo de topología automático 1408 incluye un módulo de temporización 1426, un módulo de control 1428, un módulo de supervisión de señales 1432 y un módulo de visualización de señales 1434. El módulo de temporización 1426 puede designar una unidad de visualización particular en el conjunto de unidades de visualización 1430 como una unidad de visualización de muestra. En un caso de este tipo, el módulo de temporización 1426 puede designar las restantes unidades de visualización 1430 como unidades de visualización adicionales. En un ejemplo, el módulo de temporización 1426 establece automáticamente un factor de forma para ser compatible con los datos de perfil de hardware 1402, en donde la presentación 1440 se inicia automáticamente por una secuencia de señales gráficas 1422.

En un ejemplo, el módulo de control 1428 modifica el conjunto de ajustes de topología automáticos 1406. Adicionalmente, el módulo de supervisión de señales 1432 puede supervisar automáticamente la secuencia de señales gráficas 1422 y desencadenar el dispositivo de almacenamiento 1414 para asociar el conjunto de ajustes de topología automáticos 1406 con la tabla de consulta de perfiles de hardware 1412. Además, el módulo de supervisión de señales 1432 puede detectar automáticamente cambios en el conjunto de unidades de visualización 1430 de acuerdo con un conjunto de criterios de cambio y generar automáticamente un nuevo perfil de topología que corresponde al cambio en el conjunto de unidades de visualización 1430. Por lo tanto, el nuevo perfil de topología puede aplicarse al conjunto de unidades de visualización 1430. El módulo de supervisión de señales 1432 también pueden desencadenar el módulo de visualización de señales 1434 para aplicar de nuevo el conjunto de ajustes de topología automáticos 1406 si la secuencia de señales gráficas 1422 no cumple con un conjunto de criterios. Si los datos de perfil de hardware 1402 no soportan visualización de topología automática de la secuencia de señales gráficas 1422, el dispositivo de procesamiento de datos 1418 puede notificar un error y registrar el error en un registro de errores 1413.

Entrega de medios asistidos en la nube

Pasando ahora a la Figura 15, un sistema de juegos en la nube 1500 incluye un cliente 1540 que se acopla a un servidor 1520 a través de una red 1510. El cliente 1540 puede ser generalmente un consumidor de contenido de gráficos (por ejemplo, juegos, realidad virtual/RV, realidad aumentada/RA) que se aloja, procesa y representa en el servidor 1520. El servidor 1520 ilustrado, que puede ser escalable, tiene la capacidad para proporcionar el contenido de gráficos a múltiples clientes simultáneamente (por ejemplo, aprovechando procesamiento paralelo y distribuido y recursos de representación). En un ejemplo, la escalabilidad del servidor 1520 está limitada por la capacidad de la red 1510. Por consiguiente, puede haber algún número de clientes umbral por encima del cual puede degradarse el servicio a todos los clientes.

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

Más particularmente, el procesador de gráficos 1530 ilustrado incluye un representador de gráficos 1532 que ejecuta procedimientos de representación de acuerdo con los comandos de representación generados por el procesador anfitrión 1524. La salida del representador de gráficos 1532 puede ser un flujo de fotogramas de vídeo sin procesar que se proporcionan a un capturador de fotogramas 1534. El capturador de fotogramas 1534 ilustrado se acopla a un codificador 1536, que puede comprimir/formatear el flujo de vídeo sin procesar para su transmisión a través de la red 1510. El codificador 1536 puede usar una amplia variedad de algoritmos de compresión de vídeo tales como, por ejemplo, la norma H.264 de Sector de Normalización de Telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU), la norma de Codificación de Vídeo Avanzada (AVC) de MPEG4 de la Organización Internacional de Normalización/Comisión Electrotécnica Internacional (ISO/IEC), y así sucesivamente.

El cliente 1540 ilustrado, que puede ser un ordenador de sobremesa, ordenador portátil, ordenador de tableta, tableta convertible, dispositivo ponible, MID, PDA, reproductor de medios, 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 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 lo tanto, el flujo de vídeo descomprimido puede proporcionarse desde el decodificador 1544 a un representador de vídeo 1546. El representador de vídeo 1546 ilustrado se acopla a un visualizador 1548 que presenta visualmente el contenido de gráficos.

Como ya se ha indicado, el contenido de gráficos puede incluir contenido de juegos. En este sentido, el cliente 1540 puede llevar a cabo difusión en continuo interactiva en tiempo real que implica la recopilación de entrada de usuario desde un dispositivo de entrada 1550 y entrega de la entrada de usuario al servidor 1520 a través de la red 1510. Este componente interactivo en tiempo real de juego en la nube puede plantear desafíos con respecto a la latencia.

Ejemplo de visión general del sistema adicional

La Figura 16 es un diagrama de bloques de un sistema de procesamiento 1600, de acuerdo con una realización. En diversas realizaciones, el sistema 1600 incluye uno o más procesadores 1602 y uno o más procesadores de gráficos 1608, y puede ser un único sistema de escritorio de procesador, 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 uso en dispositivos móviles, portátiles o integrados.

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

En algunas realizaciones, cada uno del uno o más procesadores 1602 incluye uno o más núcleos de procesador 1607 para procesar instrucciones que, cuando se ejecutan, realizan operaciones para el sistema y el software del usuario.

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 el cálculo de conjunto de instrucciones complejo (CISC), el cálculo de conjunto de instrucciones reducido (RISC), o cálculo mediante una palabra de instrucción muy larga (VLIW). Cada uno de los múltiples núcleos de procesador 1607 puede procesar un conjunto de instrucciones 1609 diferente, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo de procesador 1607 puede incluir también otros dispositivos de procesamiento, tal como un procesador de señales digitales (DSP).

En algunas realizaciones, el procesador 1602 incluye la memoria caché 1604. Dependiendo de la arquitectura, el procesador 1602 puede tener una única caché interna o múltiples niveles de caché interna. En algunas realizaciones, la memoria caché se comparte entre diversos componentes del procesador 1602. En algunas realizaciones, el procesador 1602 también usa una caché externa (por ejemplo, una caché de nivel 3 (L3) o caché de último nivel (LLC)) (no mostrada), que puede compartirse entre núcleos de procesador 1607 usando técnicas de coherencia de caché conocidas. Un fichero de registro 1606 está incluido adicionalmente 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 de instrucción). Algunos registros pueden ser registros de fin general, mientras que otros registros pueden ser específicos al diseño del procesador 1602.

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

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

En algunas realizaciones, el ICH 1630 posibilita que los periféricos se conecten al dispositivo de memoria 1620 y al procesador 1602 mediante a un bus de E/S de alta velocidad. Los periféricos de E/S incluyen, pero sin limitación, un controlador de audio 1646, una interfaz de firmware 1628, un transceptor inalámbrico 1626 (por ejemplo, Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1624 (por ejemplo, una unidad de disco duro, memoria flash, etc.), y un controlador de E/S heredado 1640 para acoplar dispositivos heredados (por ejemplo, de tipo sistema personal 2 (PS/2)) al sistema. Uno o más controladores de Bus Serie Universal (USB) 1642 conectan los dispositivos de entrada, tales como las combinaciones de teclado y ratón 1644. Un controlador de red 1634 puede acoplarse también al ICH 1630. En algunas realizaciones, un controlador de red de alto rendimiento (no mostrado) se acopla al bus del procesador 1610. Se apreciará que el sistema 1600 mostrado es ilustrativo y no limitante, ya que pueden usarse también 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 de controlador de memoria 1616 y el concentrador de controlador de E/S 1630 pueden estar integrados en un procesador de gráficos externo discreto, tal como el procesador de gráficos externo 1612.

La Figura 17 es un diagrama de bloques de una realización de un procesador 1700 que tiene uno o más núcleos de procesador 1702A-1702N, un controlador de memoria integrado 1714 y un procesador de gráficos integrado 1708. Esos elementos de la Figura 17 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a este tipo. El procesador 1700 puede incluir núcleos adicionales hasta e incluyendo el núcleo adicional 1702N representado por los recuadros con línea discontinua. 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 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 instrucción y caché de datos dentro de cada núcleo de procesador y uno o más niveles de 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 caché antes de memoria externa se clasifica como el LLC. En algunas realizaciones, la lógica de coherencia de caché mantiene la coherencia entre las diversas unidades de caché 1706 y 1704A-1704N.

- En algunas realizaciones, el procesador 1700 puede incluir también un conjunto de una o más unidades de controlador de bus 1716 y un núcleo de agente de sistema 1710. La una o más unidades de controlador de bus 1716 gestionan un conjunto de buses periféricos, tal como uno o más buses de Interconexión de Componentes Periféricos (por ejemplo, PCI, PCI Express). El núcleo de agente de sistema 1710 proporciona la funcionalidad de gestión para los diversos componentes de procesador. En algunas realizaciones, el núcleo de agente de sistema 1710 incluye uno o más controladores de memoria integrados 1714 para gestionar el acceso a diversos dispositivos de memoria externos (no mostrados).
- En algunas realizaciones, uno o más de los núcleos de procesador 1702A-1702N incluyen soporte para múltiples hilos simultáneos. En tal realización, el núcleo de agente de sistema 1710 incluye componentes para coordinar y operar los núcleos 1702A-1702N durante el procesamiento de múltiples hilos. El núcleo de agente de sistema 1710 puede incluir adicionalmente una unidad de control de potencia (PCU), que incluye lógica y componentes para regular el estado de potencia de los núcleos de procesador 1702A-1702N y del procesador de gráficos 1708.
- En algunas realizaciones, el procesador 1700 incluye adicionalmente el procesador de gráficos 1708 para ejecutar operaciones de procesamiento de gráficos. En algunas realizaciones, el procesador de gráficos 1708 se acopla con el conjunto de unidades de caché compartidas 1706, y el núcleo de agente de sistema 1710, que incluye el uno o más controladores de memoria integrados 1714. En algunas realizaciones, un controlador de visualización 1711 está acoplado con el procesador de gráficos 1708 para controlar la salida del procesador de gráficos a una o más pantallas acopladas. En algunas realizaciones, el controlador de visualización 1711 puede ser un módulo separado acoplado con el procesador de gráficos mediante al menos una interconexión, o puede estar integrado dentro del procesador de gráficos 1708 o el núcleo de agente de sistema 1710.
- En algunas realizaciones, se usa una unidad de interconexión basada en anillo 1712 para acoplar los componentes internos del procesador 1700. Sin embargo, puede usarse una unidad de interconexión alternativa, tal como una interconexión de punto a punto, una interconexión conmutada, u otras técnicas, que incluyen técnicas bien conocidas en la técnica. En algunas realizaciones, el procesador de gráficos 1708 se acopla con la interconexión en anillo 1712 mediante un enlace de E/S 1713.
- El enlace de E/S 1713 ilustrativo representa al menos una de múltiples variedades de interconexiones de E/S, que incluyen una interconexión de E/S en paquete que facilita la comunicación entre diversos componentes de procesador y un módulo de memoria integrado de alto rendimiento 1718, tal como un módulo eDRAM. En algunas realizaciones, cada uno de los núcleos de procesador 1702-1702N y del procesador de gráficos 1708 usan módulos de memoria integrados 1718 como una caché de último nivel compartida.
- En algunas realizaciones, los núcleos de procesador 1702A-1702N son núcleos homogéneos que ejecutan la misma arquitectura de conjunto de instrucciones. En otra realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de arquitectura de conjunto de instrucciones (ISA), donde uno o más de los núcleos de procesador 1702A-N ejecutan un primer conjunto de instrucciones, mientras que al menos uno de los otros núcleos ejecuta un subconjunto del primer conjunto de instrucciones o un conjunto de instrucciones diferente. En una realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de microarquitectura, donde uno o más núcleos que tienen un consumo de potencia relativamente mayor se acoplan con uno o más núcleos de potencia que tienen un consumo de potencia menor. Adicionalmente, el procesador 1700 puede implementarse en uno o más chips como un circuito integrado SoC que tiene los componentes ilustrados, además de otros componentes.
- La Figura 18 es un diagrama de bloques de un procesador de gráficos 1800, que puede ser una unidad de procesamiento de gráficos discreta, o puede ser un procesador de gráficos integrado con una pluralidad de núcleos de procesamiento. En algunas realizaciones, el procesador de gráficos se comunica mediante una interfaz de E/S de memoria mapeada a registros en el procesador de gráficos y con comandos colocados en la memoria del procesador. En algunas realizaciones, el procesador de gráficos 1800 incluye una interfaz de memoria 1814 para acceder a la memoria. La interfaz de memoria 1814 puede ser una interfaz a memoria local, a una o más cachés internas, a una o más cachés externas compartidas y/o a memoria de sistema.
- En algunas realizaciones, el procesador de gráficos 1800 también incluye un controlador de visualización 1802 para controlar la 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 vídeo o elementos de interfaz de usuario. En algunas realizaciones, el procesador de gráficos 1800 incluye un motor de códec de vídeo 1806 para codificar, decodificar o transcodificar medios a, desde o entre uno o más formatos de codificación de medios, que incluyen, pero sin limitación, formatos del Grupo de Expertos de Imágenes en Movimiento (MPEG) tales como MPEG-2, formatos de Codificación de Vídeo Avanzada (AVC) tales como H.264/MPEG-4 AVC, así como de la Sociedad de Ingenieros de Imagen en Movimiento y Televisión (SMPTE) 421M/VC-1, y formatos del Grupo Mixto de Expertos en Fotografía (JPEG) tales como los formatos JPEG y Motion JPEG (MJPEG).
- En algunas realizaciones, el procesador de gráficos 1800 incluye un motor de transferencia de imagen de bloque (BLIT) 1804 para realizar operaciones del rasterizador bidimensionales (2D) que incluyen, por ejemplo, transferencias de

bloque de límite de bits. Sin embargo, en una realización, se realizan operaciones en gráficos 2D usando uno o más componentes del motor de procesamiento de gráficos (GPE) 1810. En algunas realizaciones, el motor de procesamiento de gráficos 1810 es un motor de cálculo para realizar operaciones de gráficos, que incluyen operaciones de gráficos tridimensionales (3D) y operaciones de medios.

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

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

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

Procesamiento 3D/de medios

La Figura 19 es un diagrama de bloques de un motor de procesamiento de gráficos 1910 de un procesador de gráficos de acuerdo con algunas realizaciones. En una realización, el GPE 1910 es una versión del GPE 1810 mostrado en la Figura 18. Los elementos de la Figura 19 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a tal.

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

En algunas realizaciones, un motor de muestreo 1930 se acopla con la memoria (por ejemplo, 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 de memoria para la matriz de unidades de ejecución 1914 que permite que la matriz de ejecución 1914 lea datos de gráficos y de medios desde la memoria. En algunas realizaciones, el motor de muestreo 1930 incluye lógica para realizar operaciones de muestreo de imagen especializadas para medios.

En algunas realizaciones, la lógica de muestreo de medios especializada en el motor de muestreo 1930 incluye un módulo de eliminación de ruido/entrelazado 1932, un módulo de estimación de movimiento 1934 y un módulo de escalado y filtrado de imágenes 1936. En algunas realizaciones, el módulo de eliminación de ruido/intercalado 1932 incluye lógica para realizar uno o más de un algoritmo de eliminación de ruido o eliminación de entrelazado en datos de vídeo decodificados. La lógica de eliminación de entrelazado combina campos alternos de contenido de vídeo entrelazado en un único fotograma de vídeo. La lógica de eliminación de ruido reduce o elimina ruido de datos de vídeo y de imagen. En algunas realizaciones, la lógica de eliminación de ruido y lógica de eliminación de entrelazado son adaptables al movimiento y usan filtrado temporal o espacial basándose en la cantidad de movimiento

detectado en los datos de vídeo. En algunas realizaciones, el módulo de eliminación de ruido/entrelazado 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 vectores de movimiento que describen la transformación de datos de imagen entre fotogramas de vídeo sucesivos. En algunas realizaciones, un códec de medios de procesador de gráficos usa el motor de estimación de movimiento de vídeo 1934 para realizar operaciones en vídeo a nivel de macrobloque que de otra manera puede requerir demasiados cálculos para realizar con un procesador de fin general. En algunas realizaciones, el motor de estimación de movimiento 1934 está generalmente disponible para que componentes de procesador de gráficos ayuden con las funciones de decodificación de vídeo y de procesamiento que son sensibles o adaptables a la dirección o magnitud del movimiento dentro de 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 imágenes y 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 gráficos accedan a memoria. En algunas realizaciones, el puerto de datos 1944 facilita acceso a memoria para operaciones que incluyen escrituras objetivo de representación, lecturas de memoria intermedia constantes, lecturas/escrituras de espacio de memoria reutilizable y accesos a superficie de medios. En algunas realizaciones, el puerto de datos 1944 incluye espacio de memoria caché para almacenar en caché accesos a memoria. La memoria caché puede ser una única caché de datos o separada en múltiples cachés para los múltiples subsistemas que acceden a 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, hilos que se ejecutan en una unidad de ejecución en la matriz de unidades de ejecución 1914 se comunican con el puerto de datos intercambiando mensajes a través de una interconexión de distribución de datos que acopla cada uno de los subsistemas del GPE 1910.

Unidades de ejecución

La Figura 20 es un diagrama de bloques de otra realización de un procesador de gráficos 2000. Los elementos de la Figura 20 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a tal.

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

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

En algunas realizaciones, el procesador de gráficos 2000 incluye recursos de ejecución de hilo escalables que presentan núcleos modulares 2080A-2080N (en ocasiones denominados cortes de núcleo), teniendo cada uno múltiples subnúcleos 2050A-2050N, 2060A-2060N (en ocasiones denominados subcortes de núcleo). En algunas realizaciones, el procesador de gráficos 2000 puede tener cualquier número de núcleos de gráficos 2080A a 2080N. En algunas realizaciones, el procesador de gráficos 2000 incluye un núcleo de gráficos 2080A que tiene al menos un primer subnúcleo 2050A y un segundo subnúcleo de núcleo 2060A. En otras realizaciones, el procesador de gráficos es un procesador de baja potencia con un único subnúcleo (por ejemplo, 2050A). En algunas realizaciones, el procesador de gráficos 2000 incluye múltiples núcleos de gráficos 2080A-2080N, incluyendo cada uno un conjunto de primeros subnúcleos 2050A-2050N y un conjunto de segundos subnúcleos 2060A-2060N. Cada subnúcleo en el 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 en el conjunto de segundos subnúcleos 2060A-2060N incluye al menos un segundo conjunto de unidades de ejecución 2062A-2062N y los 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 píxel. Pueden incluirse también otros recursos compartidos en las diversas realizaciones del procesador de gráficos.

La Figura 21 ilustra lógica de ejecución de hilo 2100 que incluye una matriz de elementos de procesamiento empleados en algunas realizaciones de un GPE. Los elementos de la Figura 21 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a tal.

En algunas realizaciones, la lógica de ejecución de hilo 2100 incluye un sombreador de píxeles 2102, un despachador de hilo 2104, caché de instrucciones 2106, una matriz de unidades de ejecución escalable que incluye una pluralidad de unidades de ejecución 2108A-2108N, un muestreador 2110, una caché de datos 2112 y un puerto de datos 2114. En una realización, los componentes incluidos están interconectados mediante un tejido de interconexión que enlaza a cada uno de los componentes. En algunas realizaciones, la lógica de ejecución de hilo 2100 incluye una o más conexiones a memoria, tal como la memoria de sistema o memoria caché, a través de uno 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 con capacidad de 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 unidad de ejecución individual de número.

En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N se usa esencialmente para ejecutar programas "sombreadores". 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 en 3D convencionales, de manera que se ejecutan los programas sombreadores de las librerías gráficas (por ejemplo, Direct 3D y OpenGL) con una traducción mínima. Las unidades de ejecución soportan procesamiento de vértices y de 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 fin general (por ejemplo, sombreadores de cálculo y de medios).

Cada unidad de ejecución en la matriz de unidades de ejecución 2108A-2108N opera en matrices de elementos de datos. El número de elementos de datos es el "tamaño de ejecución", o el número de canales para la instrucción. Un canal de ejecución es una unidad lógica de ejecución para acceso de elemento de datos, enmascaramiento y control de flujo dentro de las instrucciones. El número de canales puede ser independiente del número de Unidades Aritmético-Lógicas (ALU) físicas o Unidades de Coma Flotante (FPU) para un procesador de gráficos particular. En algunas realizaciones, las unidades de ejecución 2108A-2108N soportan tipos de datos de números enteros y de coma flotante.

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

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

Durante la ejecución, los gráficos y las tuberías de medios envían solicitudes de iniciación de hilo a la lógica de ejecución de hilo 2100 mediante lógica de generación y despacho de hilo. En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un despachador de hilo local 2104 que arbitra peticiones de iniciación de hilo a partir de las tuberías de gráficos y de medios e instancia los hilos requeridos en una o más unidades de ejecución 2108A-2108N. Por ejemplo, la tubería de geometría (por ejemplo, 2036 de la Figura 20) despacha procesamiento de vértices, teselación o hilos de procesamiento de geometría a lógica de ejecución de hilos 2100 (Figura 21). En algunas realizaciones, el despachador de hilo 2104 puede procesar también hilos en tiempo de ejecución que generan

solicitudes de los programas sombreadores en ejecución.

Una vez que un grupo de objetos geométricos se ha procesado y rasterizado en datos de píxel, se invoca un sombreador de píxeles 2102 para calcular adicionalmente información de salida y hacer que los resultados se escriban en 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 han de interpolarse través del objeto rasterizado. En algunas realizaciones, el sombreador de píxeles 2102 ejecuta, a continuación, un programa de sombreador de píxeles suministrado por la interfaz de programación de aplicación (API). Para ejecutar el programa de sombreador de píxeles, el sombreador de píxeles 2102 despacha hilos a una unidad de ejecución (por ejemplo, 2108A) a través del despachador de hilo 2104. En algunas realizaciones, el sombreador de píxeles 2102 usa 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 en los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico o descartan uno o más píxeles de su procesamiento adicional.

En algunas realizaciones, el puerto de datos 2114 proporciona un mecanismo de acceso a memoria para los datos procesados de salida de la lógica de ejecución de hilo 2100 para su procesamiento en una tubería de salida del procesador de gráficos. En algunas realizaciones, el puerto de datos 2114 incluye o se acopla a una o más memorias de caché (por ejemplo, la caché de datos 2112) para almacenar en caché datos para acceso de memoria mediante el puerto de datos.

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

En algunas realizaciones, las unidades de ejecución de procesador de gráficos soportan de forma nativa instrucciones en un formato de 128 bits 2210. Está disponible un formato de instrucciones de 64 bits compacto 2230 para algunas instrucciones basándose en la instrucción seleccionada, en las opciones de la instrucción y en el número de operandos. El formato de 128 bits nativo 2210 proporciona acceso a todas las opciones de instrucción, mientras que algunas opciones y operaciones están restringidas en el formato de 64 bits 2230. Las instrucciones nativas disponibles en el formato de 64 bits 2230 varían por realización. En algunas realizaciones, la instrucción se compacta en parte usando un conjunto de valores de índice en un campo de índice 2213. El hardware de la unidad de ejecución hace referencia a un conjunto de tablas de compactación basándose en los valores de índice y usa las salidas de la tabla de compactación para reconstruir una instrucción nativa en el formato de 128 bits 2210.

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

Algunas instrucciones de unidad de ejecución tienen hasta tres operandos que incluyen dos operandos de origen, src0 2220, src1 2222 y uno de destino 2218. En algunas realizaciones, las unidades de ejecución soportan instrucciones de destino duales, donde está implicado uno de los destinos. Las instrucciones de manipulación de datos pueden tener un tercer operando de origen (por ejemplo, SRC2 2224), donde el código de operación de instrucción 2212 determina el número de operandos de origen. Un último operando de origen de la instrucción puede ser un valor inmediato (por ejemplo, precodificado) pasado con la instrucción.

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

En algunas realizaciones, el formato de instrucción de 128 bits 2210 incluye un campo de modo de acceso/dirección 2226, que especifica un modo de dirección y/o un modo de acceso para la instrucción. En una realización, el modo de acceso para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones soportan modos de

acceso que incluyen un modo de acceso alineado de 16 bytes y un modo de acceso alineado de 1 byte, donde la alineación de byte del modo de acceso determina la alineación de acceso de los operandos de instrucción. Por ejemplo, cuando está en un primer modo, la instrucción 2210 puede usar direccionamiento alineado en byte para operandos de origen y destino y cuando está en un segundo modo, la instrucción 2210 puede usar direccionamiento alineado en 16 bytes para todos los operandos de origen y destino.

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

En algunas realizaciones, las instrucciones se agrupan basándose en campos de bits de código de operación 2212 para simplificar la decodificación del código de operación 2240. Para un código de operación de 8 bits, los bits 4, 5 y 6 permiten que la unidad de ejecución determine el tipo de código de operación. La agrupación de código de operación precisa mostrada es simplemente un ejemplo. En algunas realizaciones, un grupo de código de operación de movimiento y lógica 2242 incluye instrucciones de movimiento y lógica 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 mover (mov) son en forma de 0000xxxxb y las instrucciones lógicas son en forma de 0001xxxxb. Un grupo de instrucciones de control de flujo 2244 (por ejemplo, llamada, salto (jmp)) incluye instrucciones en forma de 0010xxxxb (por ejemplo, 0x20). Un grupo de instrucciones de miscelánea 2246 incluye una mezcla de instrucciones, que incluye instrucciones de sincronización (por ejemplo, espera, envío) en forma de 0011xxxxb (por ejemplo, 0x30). Un grupo de instrucciones de cálculo paralelo 2248 incluye instrucciones aritméticas a nivel de componente (por ejemplo, añadir, multiplicar (mul)) en forma de 0100xxxxb (por ejemplo, 0x40). El grupo de cálculos paralelo 2248 realiza las operaciones aritméticas en paralelo a través de canales de datos. El grupo de cálculos vectoriales 2250 incluye instrucciones aritméticas (por ejemplo, dp4) en forma de 0101xxxxb (por ejemplo, 0x50). El grupo de cálculos vectoriales realiza cálculos aritméticos tales como cálculos de producto vectorial en operandos vectoriales.

Tubería de gráficos

La Figura 23 es un diagrama de bloques de otra realización de un procesador de gráficos 2300. Los elementos de la Figura 23 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a tal.

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

En algunas realizaciones, el emisor por flujo continuo de comando 2303 dirige la operación de un extractor de vértices 2305 que lee los datos de vértices desde la memoria y ejecuta comandos de procesamiento de vértices proporcionados por el emisor de envío por flujo continuo 2303. En algunas realizaciones, el extractor de vértices 2305 proporciona datos de vértices al sombreador de vértices 2307, que realiza operaciones de transformación e iluminación de espacio de coordenadas a 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 hilo 2331.

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

En algunas realizaciones, la tubería de gráficos 2320 incluye componentes de teselación para realizar teselación acelerada por hardware de objetos 3D. En algunas realizaciones, un sombreador de casco programable 2311 configura las operaciones de teselación. Un sombreador de dominio programable 2317 proporciona una evaluación de extremo trasero de la salida de la teselación. Un teselador 2313 opera en la dirección del sombreador de casco 2311 y contiene la lógica de fin especial para generar un conjunto de objetos geométricos detallados basándose en

un modelo geométrico aproximado que se proporciona como entrada en la tubería de gráficos 2320. En algunas realizaciones, si no se usa teselación, pueden omitirse los componentes de teselación 2311, 2313, 2317.

En algunas realizaciones, pueden procesarse los objetos geométricos completos por un sombreador de geometría 2319 a través de uno o más hilos despachados a las unidades de ejecución 2352A, 2352B o pueden proceder directamente al recortador 2329. En algunas realizaciones, el sombreador de geometría opera en objetos geométricos completos, en lugar de en vértices o en parches de vértices como en etapas anteriores de la tubería de gráficos. Si se deshabilita la teselación, el sombreador de geometría 2319 recibe la entrada desde el sombreador de vértices 2307. En algunas realizaciones, el sombreador de geometría 2319 es programable por un programa sombreador de geometría para realizar teselación de geometría si se desactivan las unidades de teselación.

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

El procesador de gráficos 2300 tiene un bus de interconexión, un tejido 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 caché o cachés asociadas 2351, el muestreador de texturas y medios 2354, y la caché de texturas/muestreador 2358 se interconectan mediante un puerto de datos 2356 para realizar acceso de memoria y se comunican con los componentes de tubería de salida del representador del procesador. En algunas realizaciones, cada uno del muestreador 2354, las cachés 2351, 2358 y las unidades de ejecución 2352A, 2352B tienen rutas de acceso a memoria separadas.

En algunas realizaciones, la tubería de salida del representador 2370 contiene un rasterizador 2373 que convierte objetos basados en vértices en una representación basada en píxeles asociada. En algunas realizaciones, la lógica de rasterización incluye una unidad generadora de ventanas/enmascaradora para realizar triangulación de función fija y rasterización de línea. También están disponibles una caché del representador asociada 2378 y caché de profundidad 2379, en algunas realizaciones. Un componente de operaciones de píxel 2377 realiza operaciones basadas en píxel en los datos, aunque, en algunos casos, las operaciones de píxel asociadas con las operaciones 2D (por ejemplo, transferencias de imagen de bloque de bits con mezcla) se realizan por el motor 2D 2341, o se sustituyen en tiempo de visualización por el controlador de visualización 2343 usando planos de visualización superpuestos. En algunas realizaciones, está disponible una caché L3 compartida 2375 para todos los componentes de gráficos, lo que permite la compartición de datos sin el uso de memoria de sistema principal.

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

En algunas realizaciones, el procesador de gráficos 2300 incluye un motor de visualización 2340. En algunas realizaciones, el motor de visualización 2340 es externo al procesador 2300 y se acopla con el procesador de gráficos mediante la interconexión en anillo 2302, o algún otro bus o tejido de interconexión. En algunas realizaciones, el motor de visualización 2340 incluye un motor 2D 2341 y un controlador de visualización 2343. En algunas realizaciones, el motor de visualización 2340 contiene lógica de fin especial que puede operar independientemente de la tubería 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 de sistema integrado, como en un ordenador portátil, o un dispositivo de visualización externo adjunto mediante un conector de dispositivo de visualización.

En algunas realizaciones, la tubería de gráficos 2320 y la tubería de medios 2330 son configurables para realizar operaciones basándose en múltiples gráficos e interfaces de programación de medios y no son específicas a ninguna interfaz de programación de aplicación (API). En algunas realizaciones, el software del controlador para el procesador de gráficos traduce llamadas de API que son específicas para unos gráficos o librería de medios particular en comandos que pueden procesarse por el procesador de gráficos. En algunas realizaciones, se proporciona soporte para la Biblioteca de Gráficos Abierta (OpenGL) y Lenguaje de Computación Abierto (OpenCL) de Khronos Group, la biblioteca de Direct3D de Microsoft Corporation, o puede proporcionarse soporte tanto a OpenGL como D3D. Puede proporcionarse también soporte para la Biblioteca de Visión Informática de Código Abierto (OpenCV). También se soportaría una API futura con una tubería 3D compatible si pudiera realizarse un mapeo desde la tubería de la API futura a la tubería del procesador de gráficos.

Programación de tubería de gráficos

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

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

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

En algunas realizaciones, la secuencia de comandos del procesador de gráficos 2410 puede comenzar con un comando de vaciado de tubería 2412 para hacer que cualquier tubería de gráficos activa complete los comandos actualmente pendientes para la tubería. En algunas realizaciones, la tubería 3D 2422 y la tubería de medios 2424 no operan concurrentemente. Se realiza el vaciado de tubería para hacer que la tubería de gráficos activa complete cualquier comando pendiente. En respuesta a un vaciado de tubería, el analizador de comando para el procesador de gráficos pausará el procesamiento de comandos hasta que los motores de dibujo activos completen las operaciones pendientes y se invaliden las cachés de lectura relevantes. Opcionalmente, cualquier dato en la caché del representador que se marque 'sucio' puede vaciarse en memoria. En algunas realizaciones, el comando de vaciado de tubería 2412 puede usarse para la sincronización de tubería o antes de colocar el procesador de gráficos en un estado de baja potencia.

En algunas realizaciones, se usa un comando de selección de tubería 2413 cuando una secuencia de comandos requiere que el procesador de gráficos conmute explícitamente entre tuberías. En algunas realizaciones, se requiere únicamente un comando de selección de tubería 2413 una vez dentro de un contexto de ejecución antes de emitir comandos de tubería a menos que el contexto sea emitir comandos para ambas tuberías. En algunas realizaciones, se requiere un comando de vaciado de tubería 2412 inmediatamente antes de una conmutación de tubería mediante el comando de selección de tubería 2413.

En algunas realizaciones, un comando de control de tubería 2414 configura una tubería de gráficos para su operación y se usa para programar la tubería 3D 2422 y la tubería de medios 2424. En algunas realizaciones, el comando de control de tubería 2414 configura el estado de la tubería para la tubería activa. En una realización, se usa el comando de control de tubería 2414 para sincronización de tubería y para limpiar datos de una o más memorias de caché dentro de la tubería activa antes del procesamiento de un lote de comandos.

En algunas realizaciones, se usan comandos de estado de memoria intermedia de retorno 2416 para configurar un conjunto de memorias intermedias de retorno para que las respectivas tuberías escriban datos. Algunas operaciones de tubería requieren la asignación, selección o configuración de una o más memorias intermedias de retorno en las que las operaciones escriben datos intermedios durante el procesamiento. En algunas realizaciones, el procesador de gráficos también usa una o más memorias intermedias de retorno para almacenar datos de salida y para realizar comunicación de hilo transversal. En algunas realizaciones, el estado de la memoria intermedia de retorno 2416 incluye seleccionar el tamaño y número de memorias intermedias de retorno para usar para un conjunto de operaciones de tubería.

Los comandos restantes en la secuencia de comandos difieren basándose en la tubería activa para las operaciones. Basándose en una determinación de tubería 2420, la secuencia de comandos está adaptada a la tubería 3D 2422 que comienza con el estado de tubería 3D 2430 o la tubería de medios 2424 que comienza en el estado de la tubería de medios 2440.

Los comandos para el estado de la tubería 3D 2430 incluyen comandos de ajuste de estado 3D para el estado de la memoria intermedia de vértice, el estado de color constante, el estado de la memoria intermedia de profundidad y otras variables de estado que han de configurarse antes de que se procesen los comandos de las primitivas 3D. Los valores de estos comandos se determinan, al menos en parte, basándose en la API 3D particular en uso. En algunas realizaciones, los comandos de estado de tubería 3D 2430 también pueden desactivar u omitir de manera selectiva ciertos elementos de tubería si no se usarán estos elementos.

En algunas realizaciones, se emite el comando de la primitiva 3D 2432 para enviar primitivas 3D para que se procesen por la tubería 3D. Los comandos y parámetros asociados que se pasan al procesador de gráficos mediante el comando de primitiva 3D 2432 se reenvían a la función de extracción de vértice en la tubería de gráficos. La función de extracción de vértice usa los datos de comando de primitiva 3D 2432 para generar estructuras de datos de vértice. Las estructuras de datos de vértice se almacenan en una o más memorias intermedias de retorno. En algunas realizaciones, se usa el comando de primitiva 3D 2432 para realizar operaciones de vértice en primitivas 3D mediante sombreadores de vértice. Para procesar sombreadores de vértice, la tubería 3D 2422 despacha hilos de ejecución de sombreador a las unidades de ejecución de procesador de gráficos.

En algunas realizaciones, se activa la tubería 3D 2422 mediante un comando o evento de ejecución 2434. En algunas realizaciones, una escritura de registro activa la ejecución del comando. En algunas realizaciones, se activa la ejecución mediante un comando 'ir' o 'disparar' en la secuencia de comandos. En una realización, se activa la ejecución del comando usando un comando de sincronización de tubería para vaciar la secuencia de comandos a través de la tubería de gráficos. La tubería 3D realizará un procesamiento de geometría para las primitivas 3D. Una vez que las operaciones están completadas, se rasterizan los objetos geométricos resultantes y el motor de píxeles colorea los píxeles resultantes. Pueden incluirse también comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo trasero para estas operaciones.

En algunas realizaciones, la secuencia de comandos del procesador de gráficos 2410 sigue la ruta de tubería de medios 2424 cuando se realizan operaciones de medios. En general, el uso y la manera específica de la programación para la tubería de medios 2424 depende de los medios o de las operaciones de cálculo que han de realizarse. Las operaciones de decodificación de medios específicas pueden descargarse en la tubería de medios durante la decodificación de medios. En algunas realizaciones, puede omitirse también la tubería de medios y puede realizarse la decodificación de medios en su totalidad o en parte usando recursos proporcionados por uno o más núcleos de procesamiento de fin general. En una realización, la tubería de medios también incluye elementos para las operaciones de unidad de procesador de gráficos de fin general (GPGPU), donde se usa el procesador de gráficos para realizar operaciones vectoriales SIMD usando programas sombreadores computacionales que no están relacionados explícitamente con la representación de las primitivas gráficas.

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

En algunas realizaciones, los comandos de objeto de medios 2442 suministran punteros a objetos de medios para su procesamiento por la tubería de medios. Los objetos de medios incluyen memorias intermedias de memoria que contiene datos de vídeo que van a procesarse. En algunas realizaciones, todos los estados de la tubería de medios deben ser válidos antes de emitir un comando de objeto de medios 2442. Una vez que está configurado el estado de la tubería y se ponen en cola los comandos de objeto de medios 2442, se activa la tubería de medios 2424 mediante un comando de ejecución 2444 o un evento de ejecución equivalente (por ejemplo, una escritura de registro). La salida desde la tubería de medios 2424 puede a continuación procesarse posteriormente por operaciones proporcionadas por la tubería 3D 2422 o la tubería de medios 2424. En algunas realizaciones, las operaciones de la GPGPU se configuran y ejecutan de una manera similar que las operaciones de medios.

Arquitectura de software de gráficos

La Figura 25 ilustra una arquitectura de software de gráficos ilustrativa para un sistema de procesamiento de datos 2500 de acuerdo con algunas realizaciones. En algunas realizaciones, la arquitectura de software incluye una aplicación de gráficos en 3D 2510, un sistema operativo 2520 y al menos un procesador 2530. En algunas

realizaciones, el procesador 2530 incluye un procesador de gráficos 2532 y uno o más núcleo o núcleos de procesador de fin general 2534. La aplicación de gráficos 2510 y el sistema operativo 2520 cada uno puede ejecutarse en la memoria de sistema 2550 del sistema de procesamiento de datos.

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

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

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

Implementaciones de núcleo de IP

Pueden implementarse uno o más aspectos de al menos una realización por código representativo almacenado en un medio legible por máquina que representa y/o define lógica dentro de un circuito integrado tal como un procesador. Por ejemplo, el medio legible por máquina puede incluir representaciones que representan diversa lógica dentro del procesador. Cuando se leen por una máquina, las instrucciones pueden hacer que la fábrica fabrique la lógica para realizar las técnicas descritas en el presente documento. Tales representaciones, conocidas como "núcleos de IP", son unidades reutilizables de lógica para un circuito integrado que puede almacenarse en un medio legible tangible por máquina 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 puede fabricarse de manera que el circuito realiza las operaciones descritas en asociación con cualquiera de las realizaciones descritas en el presente documento.

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

El diseño de RTL 2615 o equivalente puede sintetizarse adicionalmente por la instalación de diseño en un modelo de hardware 2620, que puede ser en un lenguaje de descripción de hardware (HDL) o alguna otra representación de datos de diseño físico. El HDL puede simularse o probarse adicionalmente para verificar el diseño de núcleo de IP. El diseño de núcleo de IP puede almacenarse para su entrega a una instalación de fabricación de terceros 2665 usando memoria no volátil 2640 (por ejemplo, disco duro, memoria flash o cualquier medio de almacenamiento no volátil). Como alternativa, el diseño de núcleo de IP puede transmitirse (por ejemplo, mediante Internet) a través de una conexión cableada 2650 o conexión inalámbrica 2660. La instalación de fabricación 2665 puede fabricar a continuación un circuito integrado que está basado al menos en parte en el diseño de núcleo de IP. El circuito integrado fabricado puede estar configurado para realizar operaciones de acuerdo con al menos una realización descrita en el presente documento.

La Figura 27 es un diagrama de bloques que ilustra un sistema ilustrativo en un circuito de chip integrado 2700 que

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

Adicionalmente, pueden incluirse otra lógica y circuitos en el procesador del circuito integrado 2700, incluyendo procesadores/núcleos de gráficos adicionales, controladores de interfaz periféricos o núcleos de procesador de fin general.

El Ejemplo 3 puede incluir el sistema de automóvil del Ejemplo 1, en donde el al menos un espejo virtual comprende una pluralidad de espejos virtuales, la pluralidad de espejos virtuales acoplada al detector de la mirada, en donde si el conductor no está mirando en la dirección de uno de la pluralidad de espejos virtuales, situar cada uno de la pluralidad de espejos virtuales en un modo de operación bajo.

El Ejemplo 4 puede incluir el sistema de automóvil del Ejemplo 3, en donde si el conductor está mirando en la dirección de uno de la pluralidad de espejos virtuales, situar el espejo virtual que se está viendo en un modo de operación alto y situar cada uno de los restantes espejos virtuales en un modo de operación bajo.

El Ejemplo 5 puede incluir el sistema de automóvil del Ejemplo 3, en donde cada espejo virtual incluye una cámara acoplada comunicativamente a un visualizador, la cámara para capturar imágenes en tiempo real de los alrededores del automóvil y el visualizador para visualizar las imágenes capturadas.

El Ejemplo 6 puede incluir el sistema de automóvil del Ejemplo 5, en donde acoplado entre la cámara y el visualizador de cada espejo virtual están un codificador y un decodificador.

El Ejemplo 7 puede incluir el sistema de automóvil del Ejemplo 6, en donde un modo de operación bajo comprende una resolución inferior de al menos uno del visualizador, cámara y codificador.

El Ejemplo 8 puede incluir el sistema de automóvil del Ejemplo 6, en donde un modo de operación bajo comprende una tasa de fotogramas inferior de al menos uno del visualizador, cámara y codificador.

El Ejemplo 9 puede incluir el sistema de automóvil del Ejemplo 6, en donde un modo de operación alto comprende una resolución y tasa de fotogramas del visualizador, cámara y codificador que opera a plena capacidad.

El Ejemplo 10 puede incluir el sistema de automóvil del Ejemplo 9, en donde plena capacidad comprende una tasa de fotogramas de 60 fotogramas por segundo y una resolución de alta definición de 1920 × 1080 píxeles.

El Ejemplo 11 puede incluir el sistema de automóvil de los Ejemplos 1, 3 y 4, en donde situar los espejos virtuales en un modo de operación bajo comprende situar los espejos virtuales en un modo de ahorro de potencia.

El Ejemplo 12 puede incluir un método para rendimiento óptimo de espejos virtuales de automóvil, que comprende supervisar, a través de un detector de la mirada, los ojos de un conductor para determinar si el conductor está mirando en la dirección de un espejo virtual, y si el conductor no está mirando en la dirección de un espejo virtual, situar todos los espejos virtuales en un modo de operación bajo.

El Ejemplo 13 puede incluir el método del Ejemplo 12, en donde si el conductor está mirando en la dirección de un espejo virtual, situar el espejo virtual que se está viendo en un modo de operación alto y situar todos los otros espejos virtuales en el modo de operación bajo.

El Ejemplo 14 puede incluir el método del Ejemplo 12-13, en donde el modo de operación bajo comprende una resolución inferior de al menos uno de un visualizador, una cámara y un codificador del espejo virtual.

El Ejemplo 15 puede incluir el método del Ejemplo 12-13, en donde el modo de operación bajo comprende una tasa de fotogramas inferior de al menos uno del visualizador, la cámara y el codificador del espejo virtual.

El Ejemplo 16 puede incluir el método del Ejemplo 13, en donde el modo de operación alto comprende una resolución y tasa de fotogramas del espejo virtual que opera a plena capacidad.

El Ejemplo 17 puede incluir el método del Ejemplo 16, en donde plena capacidad comprende una tasa de fotogramas de 60 fotogramas por segundo y una resolución de alta definición de 1920×1080 píxeles.

5 El Ejemplo 18 puede incluir el método del Ejemplo 17, en donde modo de operación bajo es al menos un factor de 4 menor en tasa de fotogramas y resolución que plena capacidad.

10 El Ejemplo 19 puede incluir el método del Ejemplo 16, en donde el espejo virtual comprende una cámara, un codificador, un decodificador y un visualizador, y operar a plena capacidad comprende operar la cámara, el codificador y el visualizador a tasa de fotogramas completa y resolución completa.

15 El Ejemplo 20 puede incluir un sistema para rendimiento de potencia óptimo de espejos virtuales en automóviles, que comprende un detector de la mirada para supervisar los ojos de un conductor y movimientos de la cabeza para determinar si el conductor está mirando en la dirección de un espejo virtual, y si el conductor no está mirando en la dirección de un espejo virtual, el detector de la mirada para enviar una señal a todos los espejos virtuales para situar los mismos en un modo de operación bajo, y si el conductor está mirando en la dirección de un espejo virtual, el detector de la mirada para enviar una señal al espejo virtual que se está viendo para situar el mismo en un modo de operación alto y el detector de la mirada para enviar una señal a todos los otros espejos virtuales para situar los mismos en el modo de operación bajo.

20 El Ejemplo 21 puede incluir el sistema del Ejemplo 20, en donde los espejos virtuales incluyen un visualizador, una cámara y un codificador, y modo de operación bajo comprende una resolución inferior de al menos uno del visualizador, la cámara y el codificador del espejo virtual.

25 El Ejemplo 22 puede incluir el sistema del Ejemplo 21, en donde el modo de operación bajo comprende una tasa de fotogramas inferior de al menos uno del visualizador, la cámara y el codificador del espejo virtual.

El Ejemplo 23 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la resolución del visualizador.

30 El Ejemplo 24 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la tasa de fotogramas del visualizador.

35 El Ejemplo 25 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la resolución de la cámara.

El Ejemplo 26 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la tasa de fotogramas de la cámara.

40 El Ejemplo 27 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la resolución del codificador.

El Ejemplo 28 puede incluir el sistema del Ejemplo 21, en donde situar todos los espejos virtuales en un modo de operación bajo comprende disminuir la tasa de fotogramas del codificador.

45 El Ejemplo 29 puede incluir el sistema del Ejemplo 20, en donde cada uno de los espejos virtuales incluye un visualizador, una cámara y un codificador, y el modo de operación alto comprende una resolución y tasa de fotogramas del visualizador, operando la cámara y el codificador de cada uno de los espejos virtuales a plena capacidad.

50 El Ejemplo 30 puede incluir el sistema del Ejemplo 29, en donde plena capacidad comprende una tasa de fotogramas de al menos 60 fotogramas por segundo y una resolución de alta definición de al menos 1920×1080 píxeles.

El Ejemplo 31 puede incluir el sistema del Ejemplo 29, en donde modo de operación bajo es al menos un factor de 4 menor que plena capacidad.

55 El Ejemplo 32 puede incluir el sistema del Ejemplo 20, en donde el espejo virtual comprende una cámara, un codificador, un decodificador y un visualizador, y operar a plena capacidad comprende operar la cámara, el codificador y el visualizador a tasa de fotogramas completa y resolución completa.

60 El Ejemplo 33 puede incluir al menos un medio legible por ordenador, que comprende un conjunto de instrucciones, que cuando se ejecutan por un dispositivo informático, hacen que el dispositivo informático supervise, a través de un detector de la mirada, los ojos de un conductor para determinar si el conductor está mirando en la dirección de un espejo virtual, y si el conductor no está mirando en la dirección de un espejo virtual, situar todos los espejos virtuales en un modo de operación bajo.

65 El Ejemplo 34 puede incluir al menos un medio legible por ordenador del Ejemplo 33, en donde si el conductor está mirando en la dirección de un espejo virtual, el conjunto de instrucciones, que cuando se ejecutan por el dispositivo

informático, hacen que el dispositivo informático sitúe el espejo virtual que se está viendo en un modo de operación alto y sitúe todos los otros espejos virtuales en el modo de operación bajo.

5 El Ejemplo 35 puede incluir al menos un medio legible por ordenador del Ejemplo 33-34, en donde el modo de operación bajo comprende una resolución inferior de al menos uno de un visualizador, una cámara y un codificador del espejo virtual.

10 El Ejemplo 36 puede incluir al menos un medio legible por ordenador del Ejemplo 33-34, en donde el modo de operación bajo comprende una tasa de fotogramas inferior de al menos uno del visualizador, la cámara y el codificador del espejo virtual.

El Ejemplo 37 puede incluir al menos un medio legible por ordenador del Ejemplo 34, en donde el modo de operación alto comprende una resolución y tasa de fotogramas del espejo virtual que opera a plena capacidad.

15 El Ejemplo 38 puede incluir al menos un medio legible por ordenador del Ejemplo 37, en donde plena capacidad comprende una tasa de fotogramas de 60 fotogramas por segundo y una resolución de alta definición de 1920 × 1080 píxeles.

20 El Ejemplo 39 puede incluir al menos un medio legible por ordenador del Ejemplo 38, en donde modo de operación bajo es al menos un factor de 4 menor en tasa de fotogramas y resolución que plena capacidad.

25 El Ejemplo 40 puede incluir al menos un medio legible por ordenador del Ejemplo 37, en donde el espejo virtual comprende una cámara, un codificador, un decodificador y un visualizador, y operar a plena capacidad comprende operar la cámara, el codificador y el visualizador a tasa de fotogramas completa y resolución completa.

30 El Ejemplo 41 puede incluir un sistema para rendimiento de potencia óptimo de espejos virtuales en automóviles, que comprende medios para supervisar el ojo de un conductor y movimientos de la cabeza para determinar si el conductor está mirando en la dirección de un espejo virtual, si el conductor no está mirando en la dirección de un espejo virtual, medios para situar todos los espejos virtuales en un modo de operación bajo, y si el conductor está mirando en la dirección de un espejo virtual, medios para situar el espejo virtual que se está viendo en un modo de operación alto y medios para situar todos los otros espejos virtuales en el modo de operación bajo.

35 El Ejemplo 42 puede incluir el sistema del Ejemplo 41, en donde el modo de operación bajo comprende una resolución inferior de al menos uno de un visualizador, una cámara y un codificador del espejo virtual.

El Ejemplo 43 puede incluir el sistema del Ejemplo 41, en donde el modo de operación bajo comprende una tasa de fotogramas inferior de al menos uno de un visualizador, una cámara y un codificador del espejo virtual.

40 El Ejemplo 44 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la resolución de un visualizador para cada espejo virtual.

45 El Ejemplo 45 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la tasa de fotogramas de un visualizador para cada espejo virtual.

El Ejemplo 46 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la resolución de una cámara para cada espejo virtual.

50 El Ejemplo 47 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la tasa de fotogramas de una cámara para cada espejo virtual.

55 El Ejemplo 48 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la resolución de un codificador para cada espejo virtual.

El Ejemplo 49 puede incluir el sistema del Ejemplo 41, en donde medios para situar todos los espejos virtuales en un modo de operación bajo comprende medios para disminuir la tasa de fotogramas de un codificador para cada espejo virtual.

60 El Ejemplo 50 puede incluir el sistema del Ejemplo 41, en donde el modo de operación alto comprende una resolución y tasa de fotogramas de un visualizador, una cámara y un codificador del espejo virtual que opera a plena capacidad.

El Ejemplo 51 puede incluir el sistema del Ejemplo 50, en donde plena capacidad comprende una tasa de fotogramas de al menos 60 fotogramas por segundo y una resolución de alta definición de al menos 1920 × 1080 píxeles.

65 El Ejemplo 52 puede incluir el sistema del Ejemplo 50, en donde modo de operación bajo es al menos un factor de 4

menor que plena capacidad.

5 El Ejemplo 53 puede incluir el sistema del Ejemplo 41, en donde el espejo virtual comprende una cámara, un codificador, un decodificador y un visualizador, y operar a plena capacidad comprende medios para operar la cámara, el codificador y el visualizador a tasa de fotogramas completa y resolución completa.

10 El Ejemplo 54 puede incluir un sistema de automóvil que comprende un detector de la mirada, incluyendo el detector de la mirada un procesador integrado y un dispositivo de memoria, y una pluralidad de espejos virtuales, cada uno de los espejos virtuales acoplados al detector de la mirada, en donde el detector de la mirada para supervisar de forma continua un conductor de un automóvil para determinar si el conductor está mirando en la dirección de uno de la pluralidad de espejos virtuales, y si el conductor no está mirando en la dirección de uno de la pluralidad de espejos virtuales, situar cada uno de la pluralidad de espejos virtuales en un modo de operación bajo.

15 El término "acoplado" puede usarse en el presente documento para hacer referencia a cualquier tipo de relación, directa o indirecta, entre los componentes en cuestión, y puede aplicarse a conexiones eléctricas, mecánicas, fluidas, ópticas, electromagnéticas, electromecánicas u otras conexiones. Además, los términos "primero", "segundo", etc. pueden usarse en el presente documento únicamente para facilitar la descripción, y no conllevan ninguna significancia temporal o cronológica particular a no ser 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".

20 Como se usa en esta solicitud y en las reivindicaciones, una lista de artículos unidos por la expresión "uno o más de" puede significar cualquier combinación de los términos listados. 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.

25 Las realizaciones se han descrito anteriormente con referencia a realizaciones específicas. Expertos en la materia, sin embargo, entenderán que pueden hacerse diversas modificaciones y cambios a las mismas sin alejarse del alcance de las realizaciones como se expone en las reivindicaciones adjuntas. La descripción y dibujos anteriores deben considerarse, por consiguiente, en un sentido ilustrativo en lugar de restrictivo.

REIVINDICACIONES

- 1.Un sistema de automóvil que comprende:
un detector de la mirada (902); y
5 una pluralidad de espejos virtuales (904), la pluralidad de espejos virtuales (904) acoplada al detector de la mirada (902), en donde cada espejo virtual (904) incluye una cámara (906) acoplada comunicativamente a un visualizador (908), la cámara (906) para capturar imágenes en tiempo real de los alrededores del automóvil (802) y el visualizador (908) para visualizar las imágenes capturadas y en donde acoplados entre la cámara (906) y el visualizador (908) de cada espejo virtual (904) hay un codificador (910) y un decodificador (912);
10 en donde el detector de la mirada (902) para supervisar de forma continua un conductor de un automóvil (802) para determinar si el conductor está mirando en la dirección de uno de la pluralidad de espejos virtuales (904); y
si el conductor no está mirando en la dirección del espejo virtual (904), situar cada uno de la pluralidad de espejos virtuales (904) en un modo de operación bajo, en donde, si el conductor está mirando en la dirección de uno de la pluralidad de espejos virtuales (904), situar el espejo virtual (904) que se está viendo en un modo de operación alto y
15 situar cada uno de los restantes espejos virtuales (904) en un modo de operación bajo,
en donde un modo de operación bajo comprende una resolución inferior y/o una tasa de fotogramas inferior del codificador del espejo virtual (904) sin apagar el espejo virtual (904).
- 2.El sistema de automóvil de la reivindicación 1, en donde un modo de operación alto comprende una resolución y
20 tasa de fotogramas del visualizador (908), cámara (906) y codificador (910) que opera a plena capacidad.
- 3.El sistema de automóvil de la reivindicación 2, en donde plena capacidad comprende una tasa de fotogramas de 60 fotogramas por segundo y una resolución de alta definición de 1920 × 1080 píxeles.
- 25 4.El sistema de automóvil de la reivindicación 1, en donde situar los espejos virtuales (904) en un modo de operación bajo comprende situar los espejos virtuales (904) en un modo de ahorro de potencia.
- 5.Un método para rendimiento óptimo de espejos virtuales de automóvil (904), que comprende:
supervisar, a través de un detector de la mirada (902), los ojos de un conductor para determinar si el conductor está
30 mirando en la dirección de un espejo virtual (904); y
si el conductor no está mirando en la dirección de un espejo virtual (904), situar todos los espejos virtuales (904) en un modo de operación bajo, en donde, si el conductor está mirando en la dirección del espejo virtual (904), situar el espejo virtual (904) que se está viendo en un modo de operación alto y situar todos los otros espejos virtuales en el modo de operación bajo, en donde un modo de operación bajo comprende una resolución inferior y/o una tasa de
35 fotogramas inferior de un codificador del espejo virtual (904) sin apagar el espejo virtual (904).

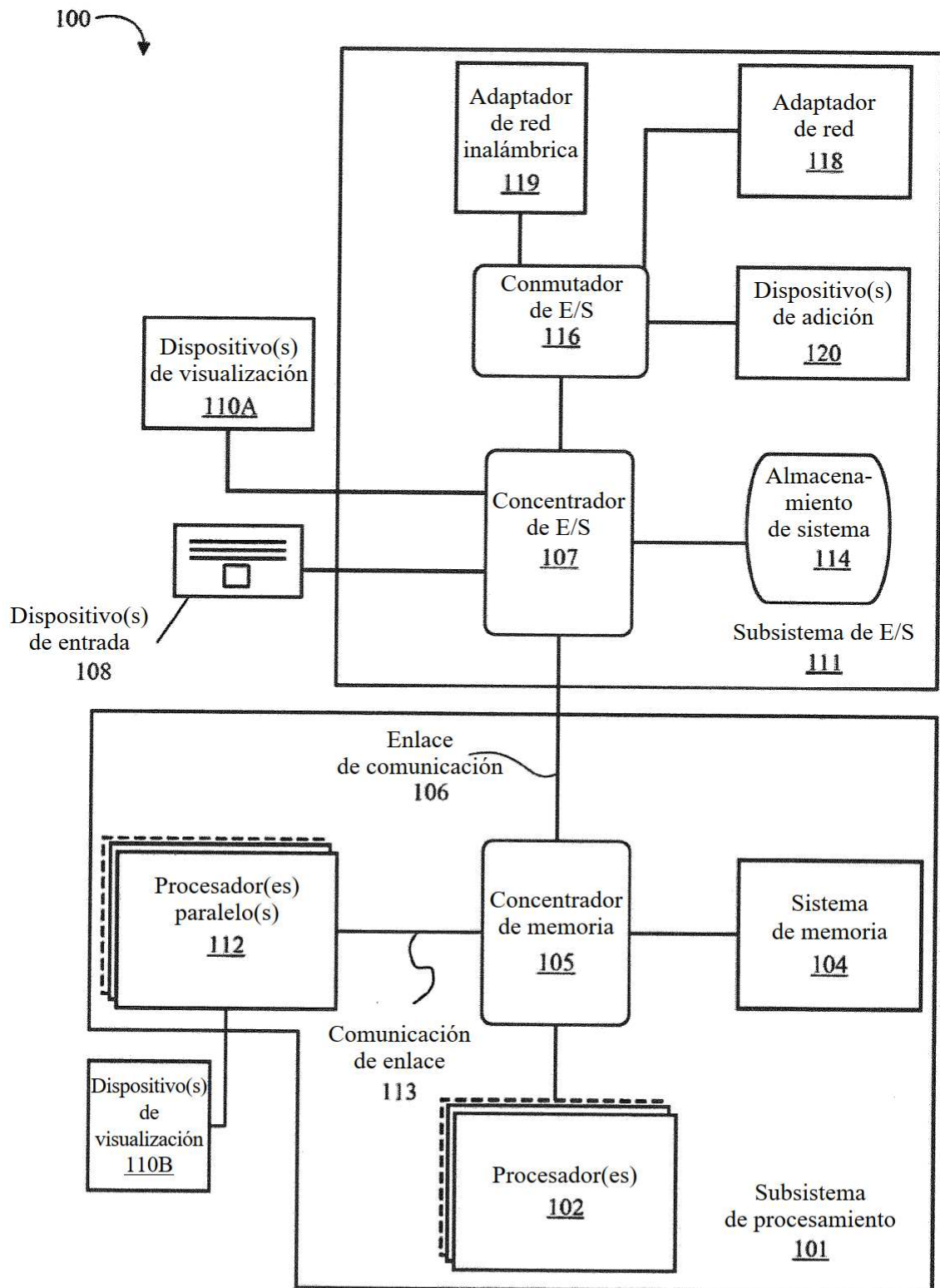


FIG. 1

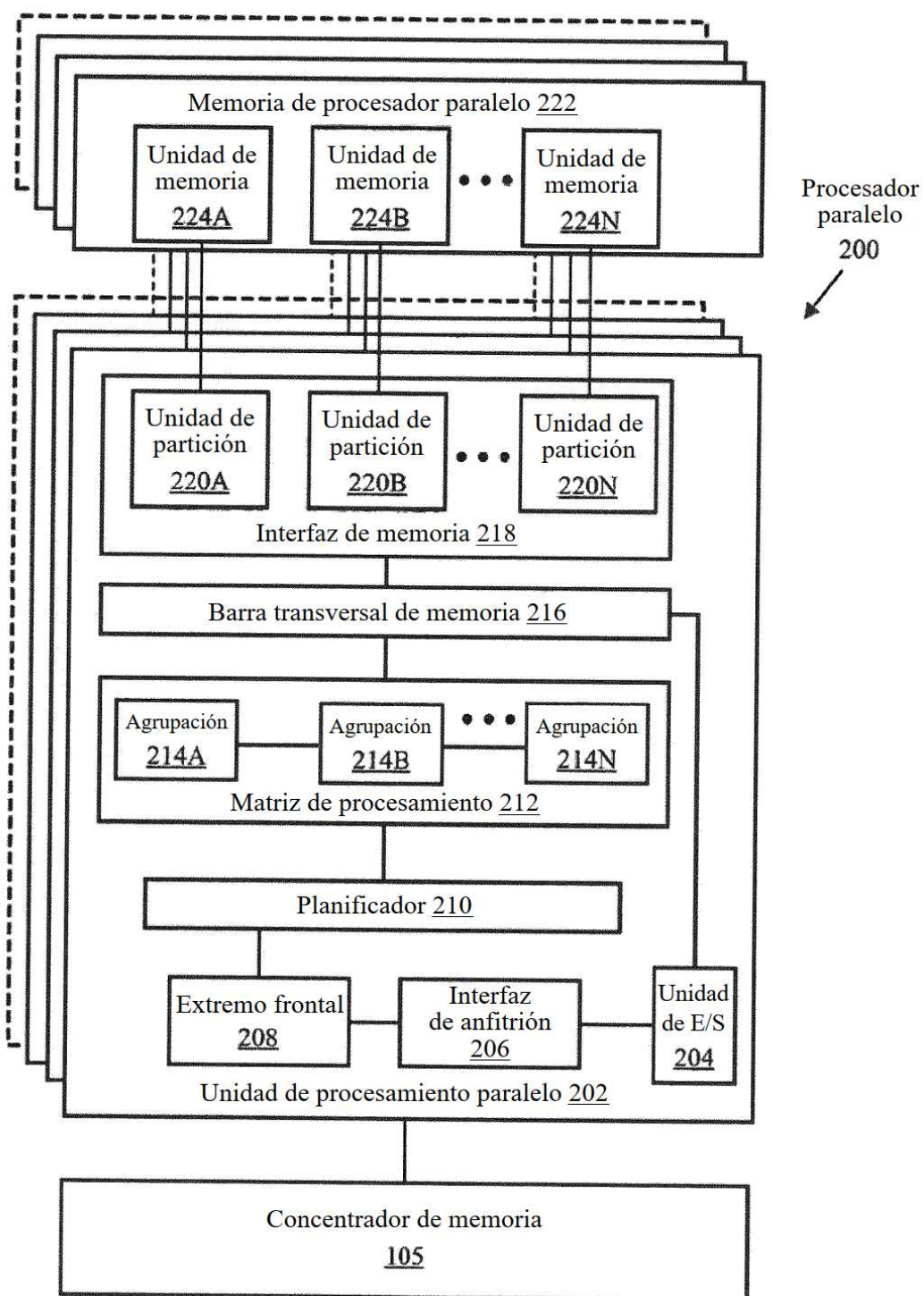


FIG. 2A

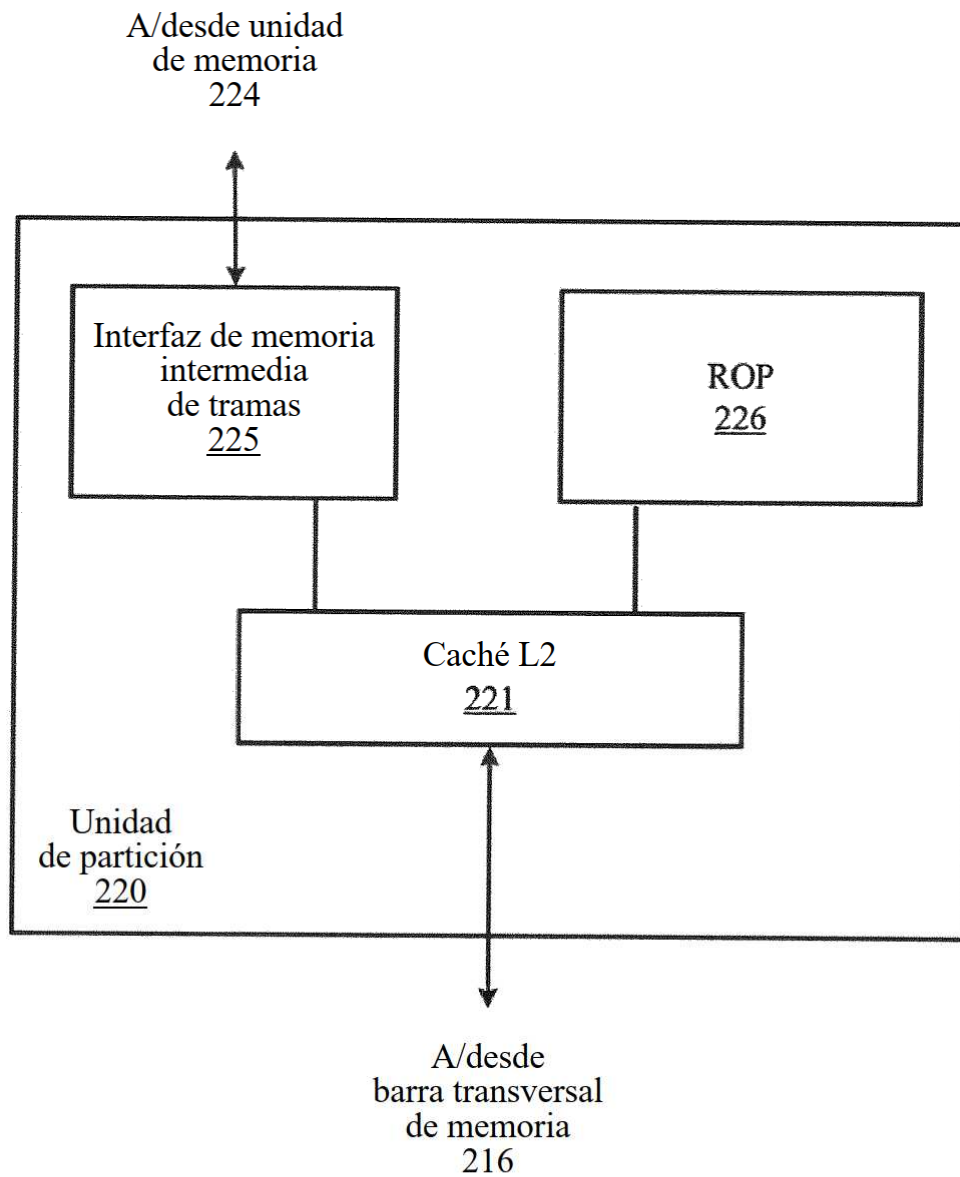


FIG. 2B

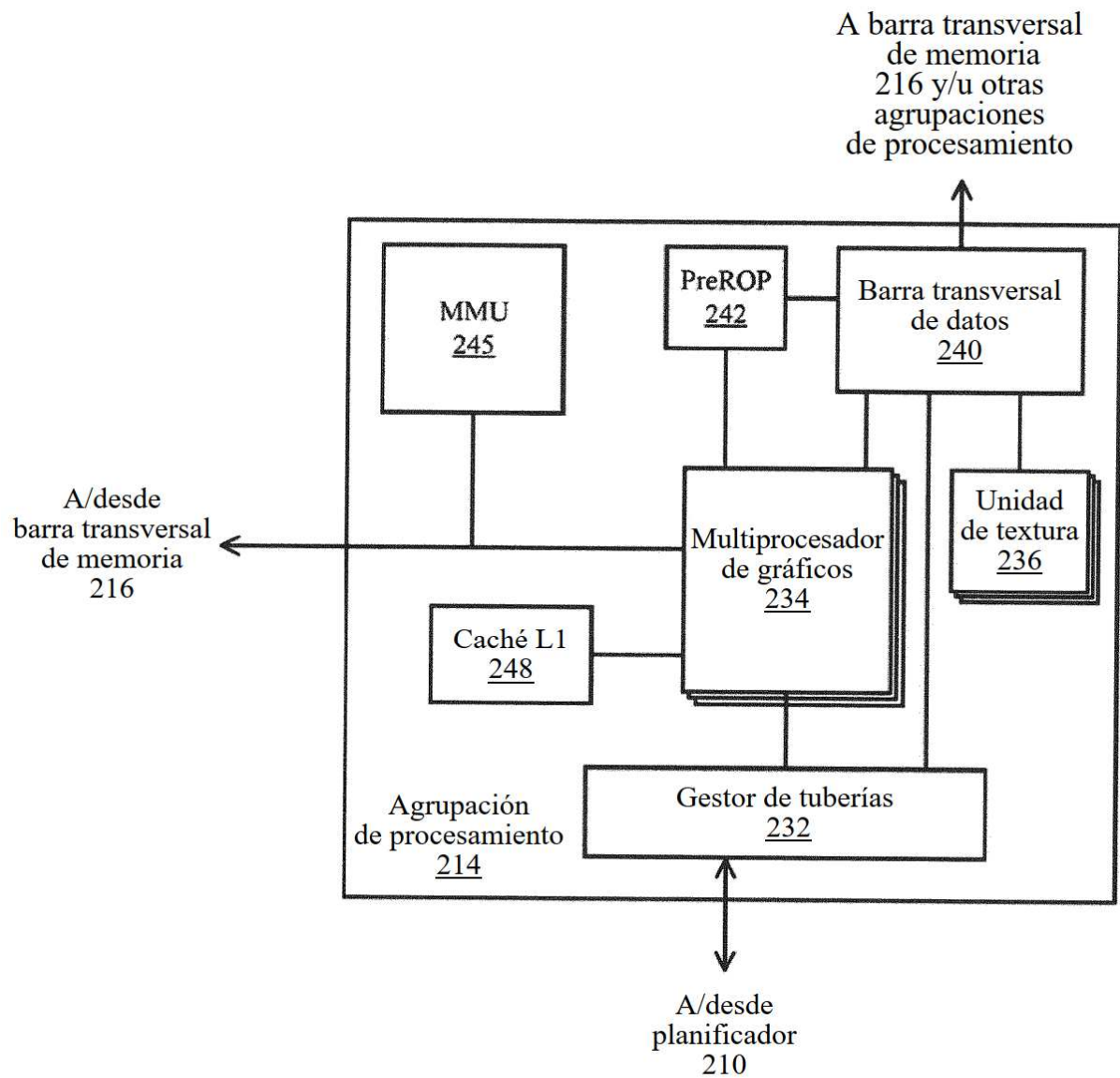


FIG. 2C

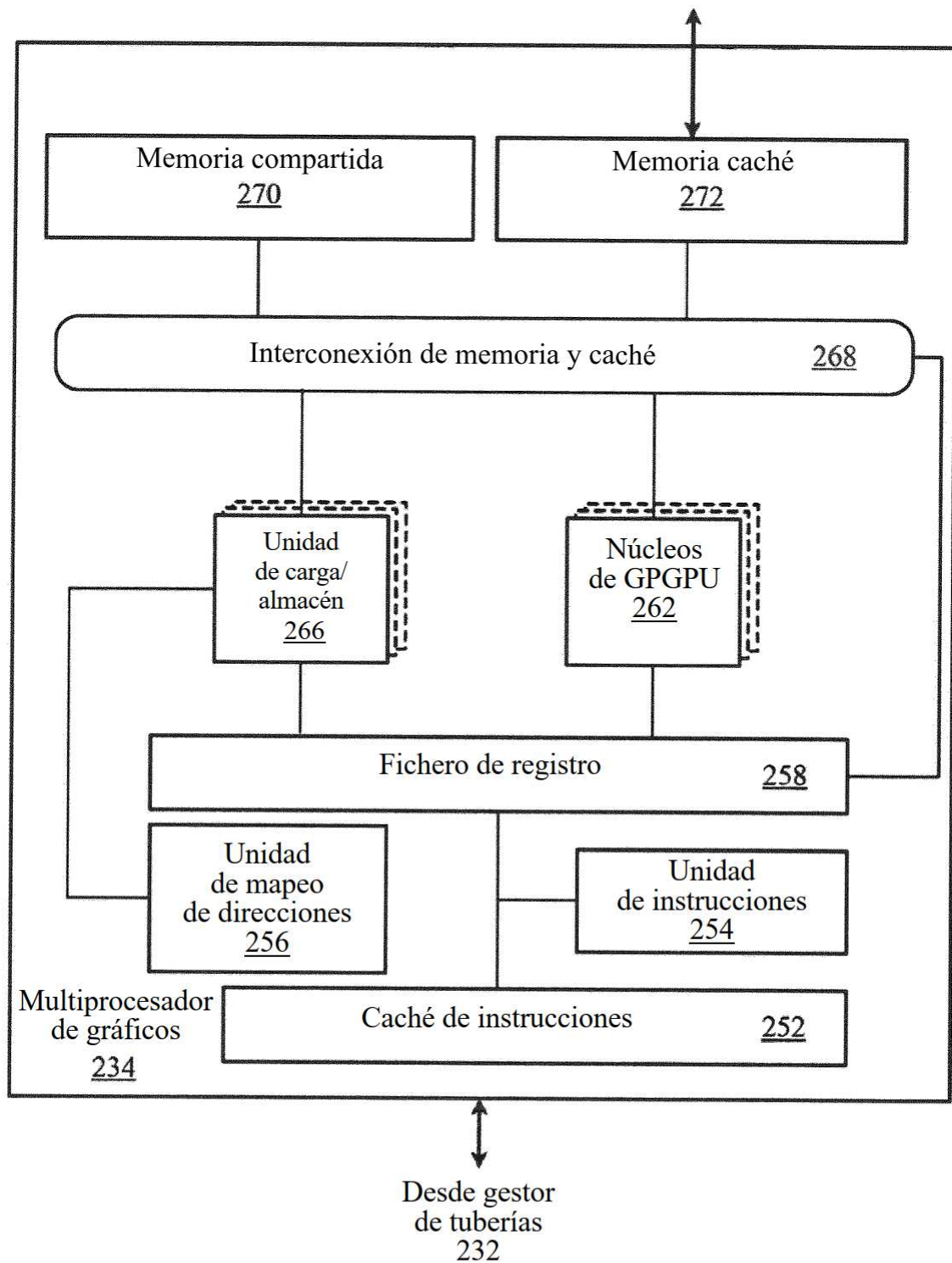


FIG. 2D

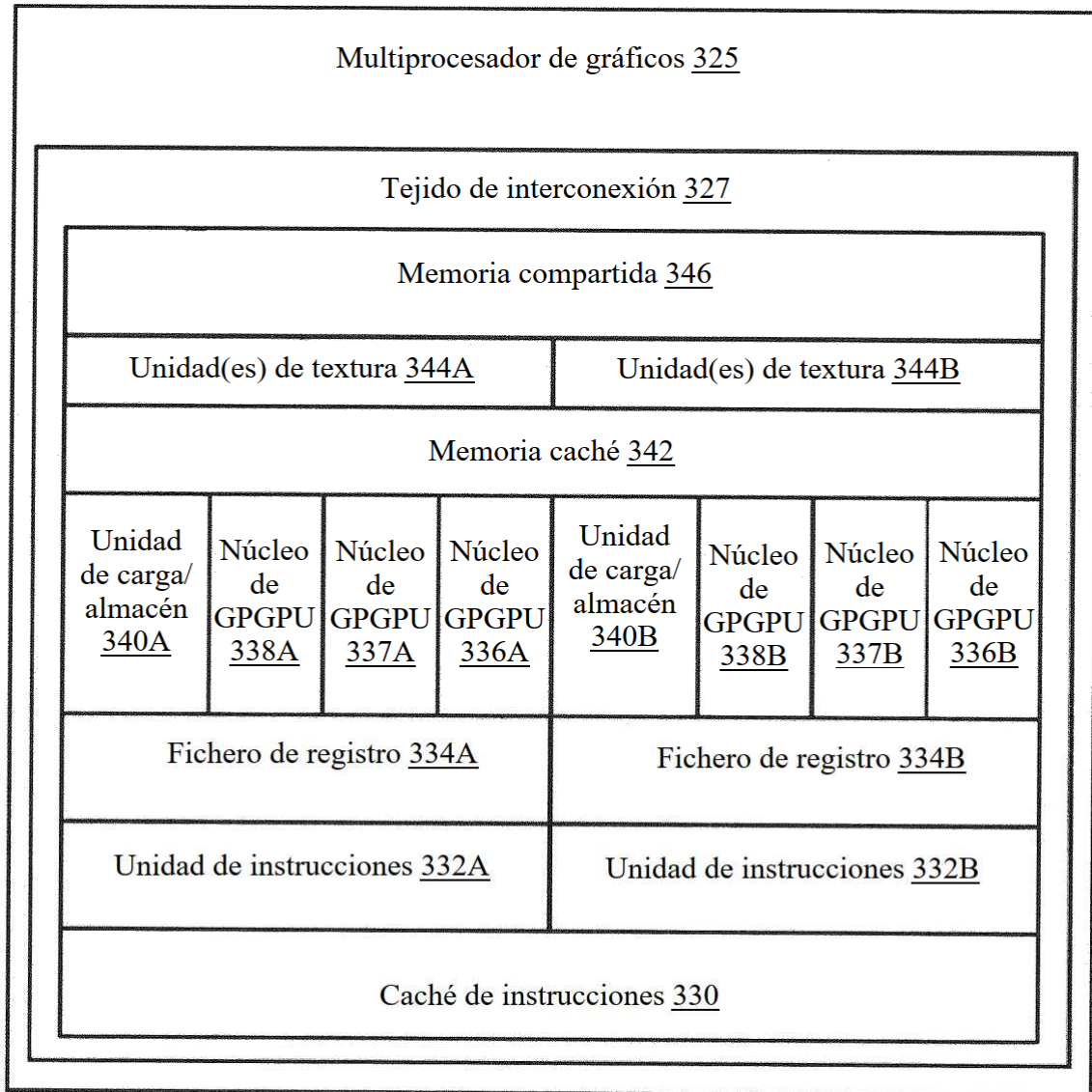


FIG. 3A

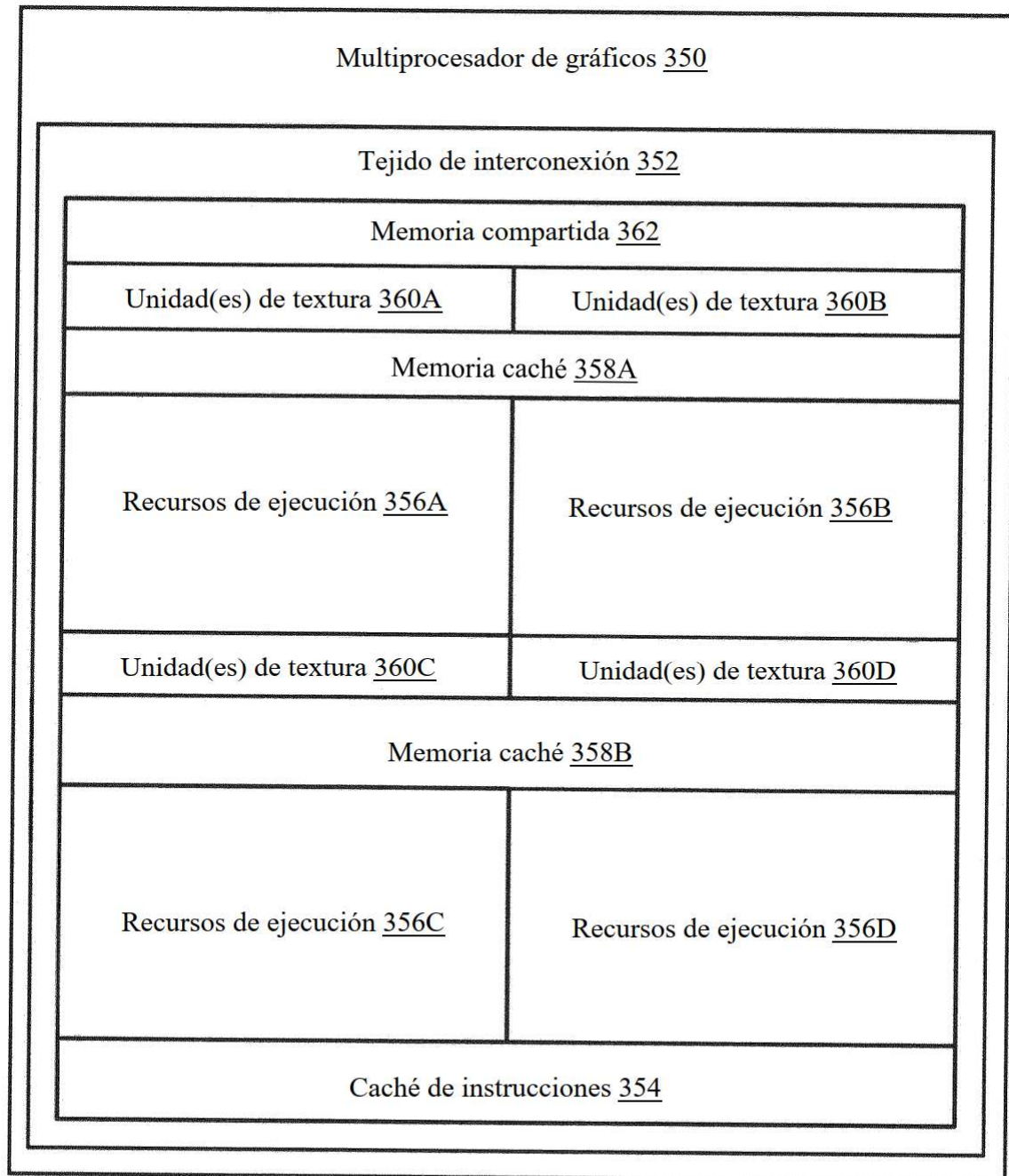


FIG. 3B

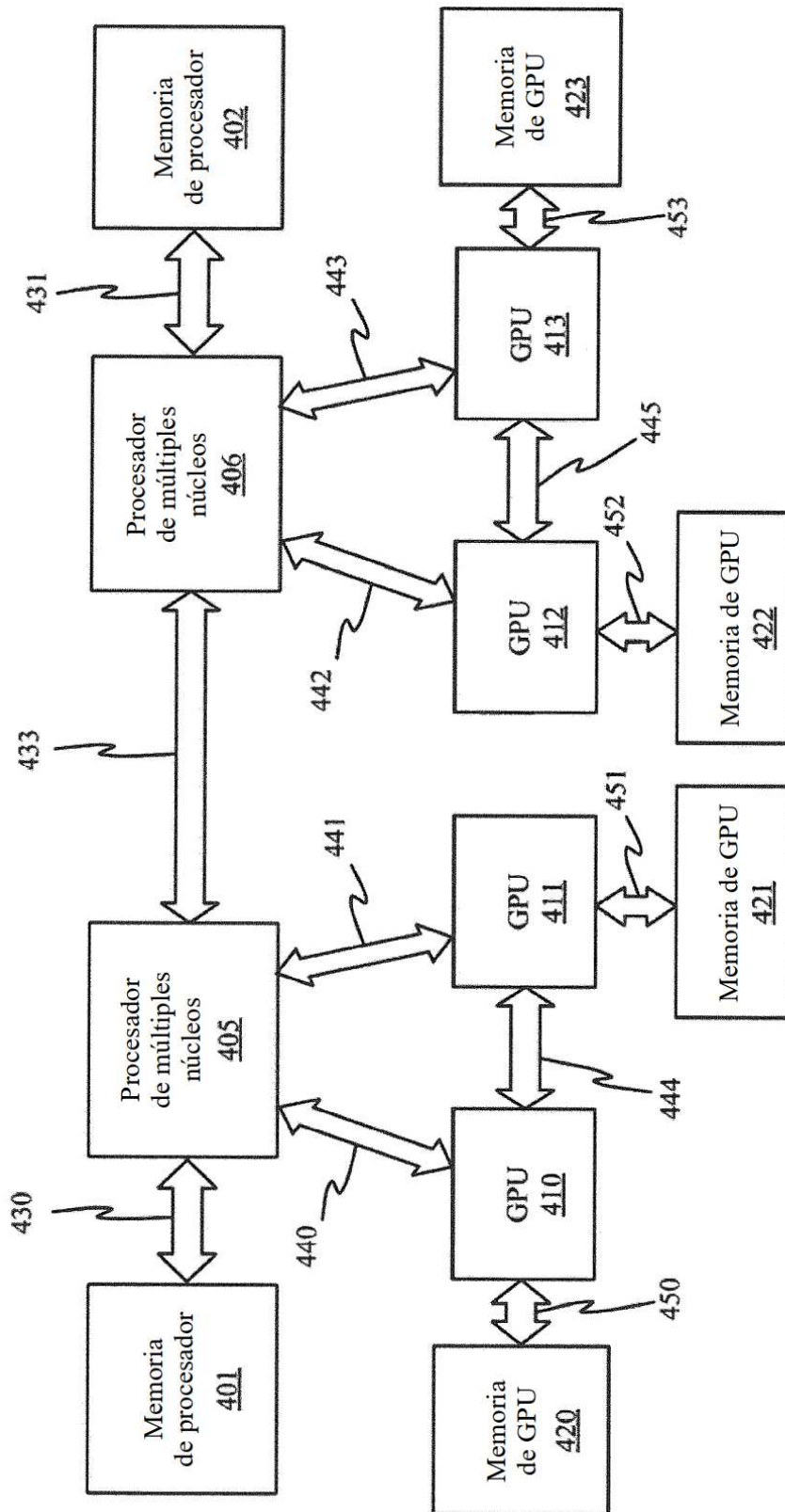


FIG. 4A

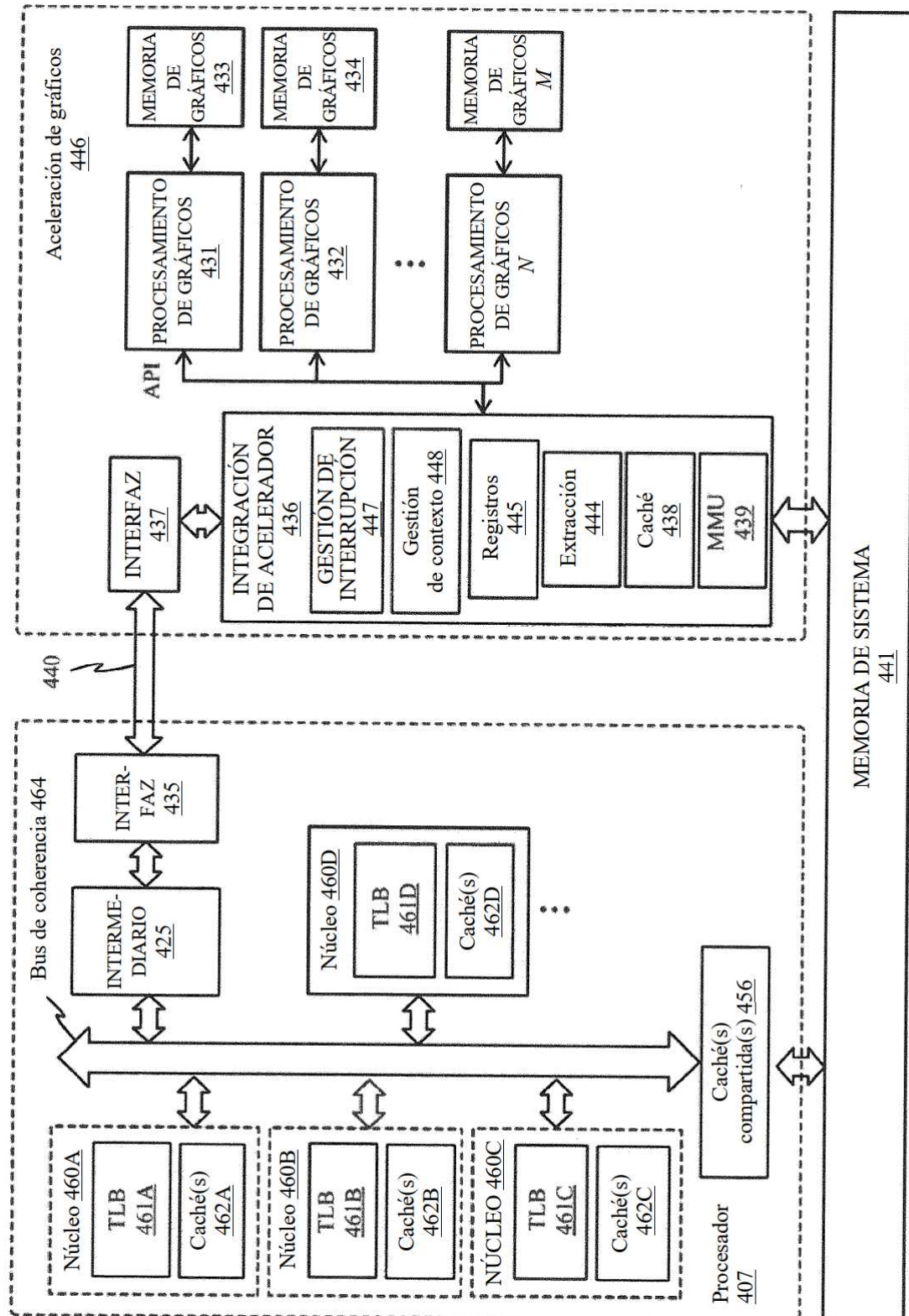


FIG. 4B

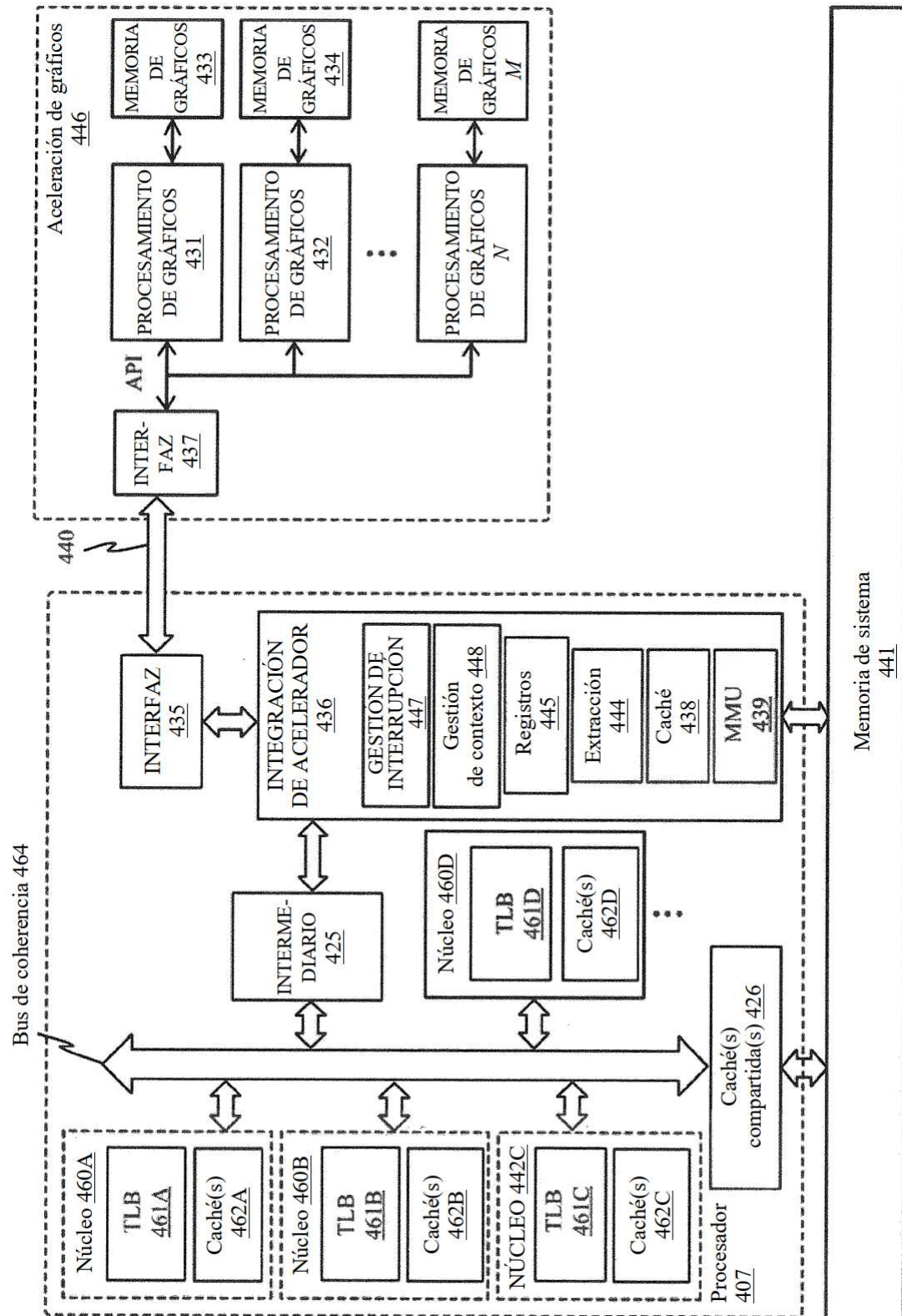


FIG. 4C

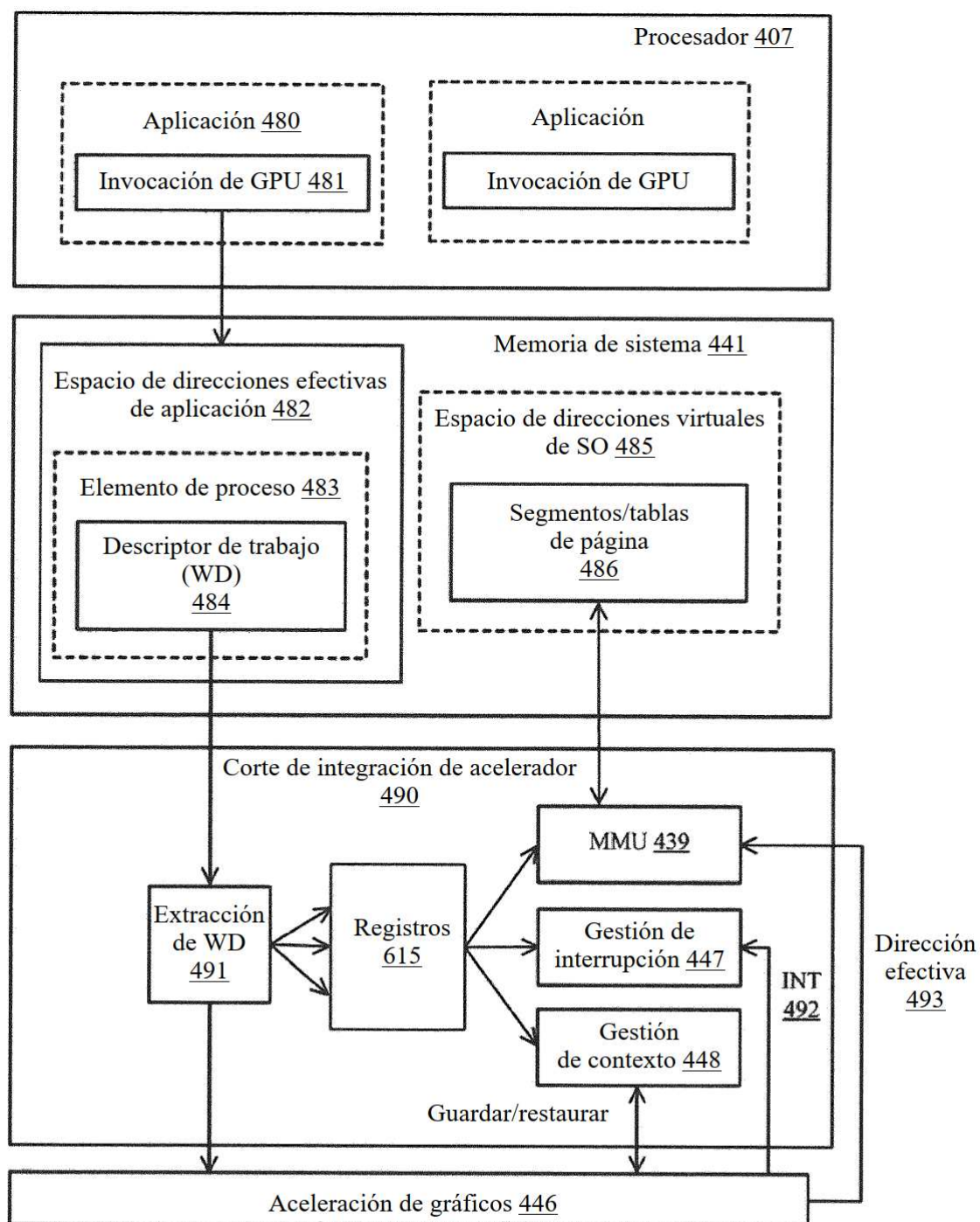


FIG. 4D

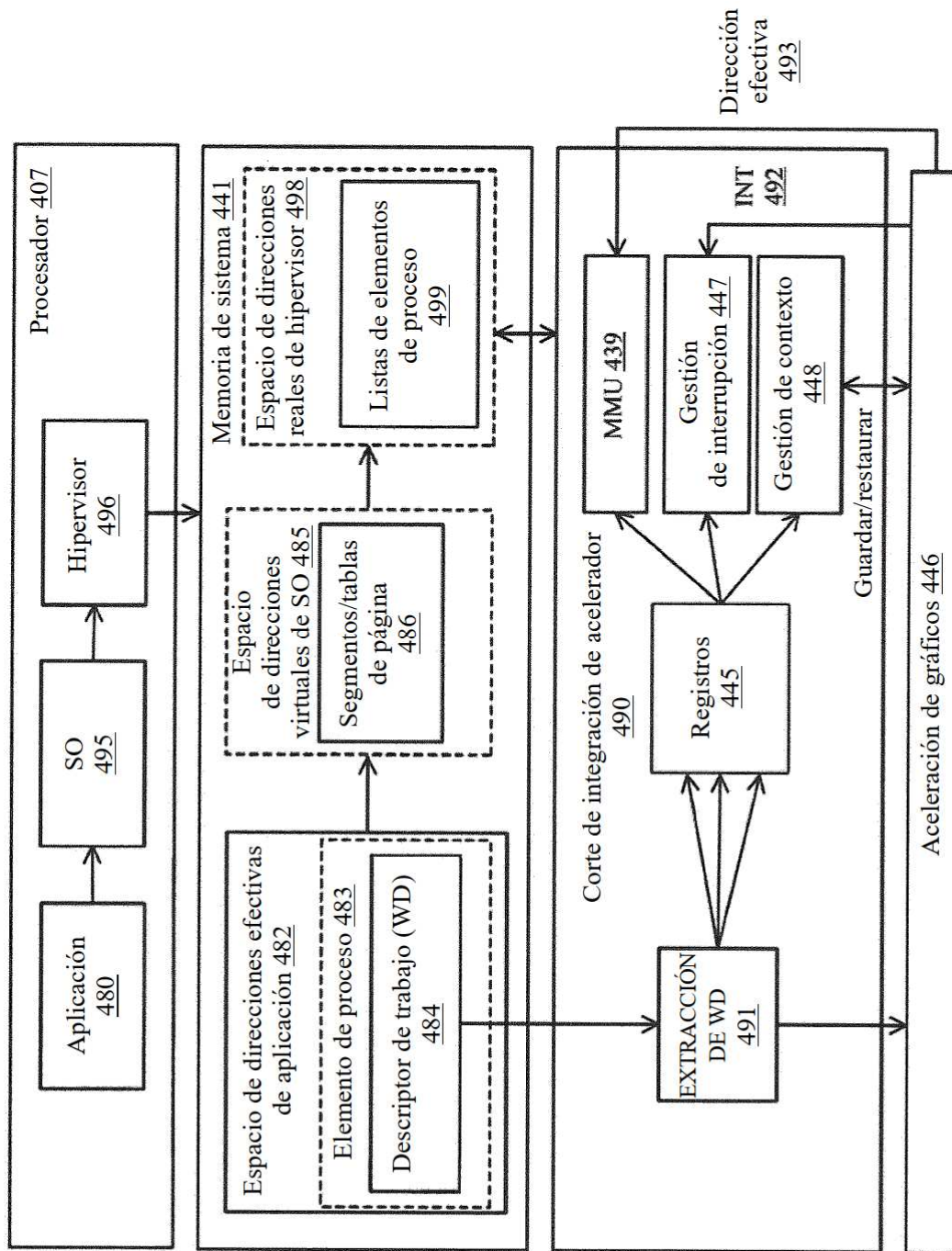


FIG. 4E

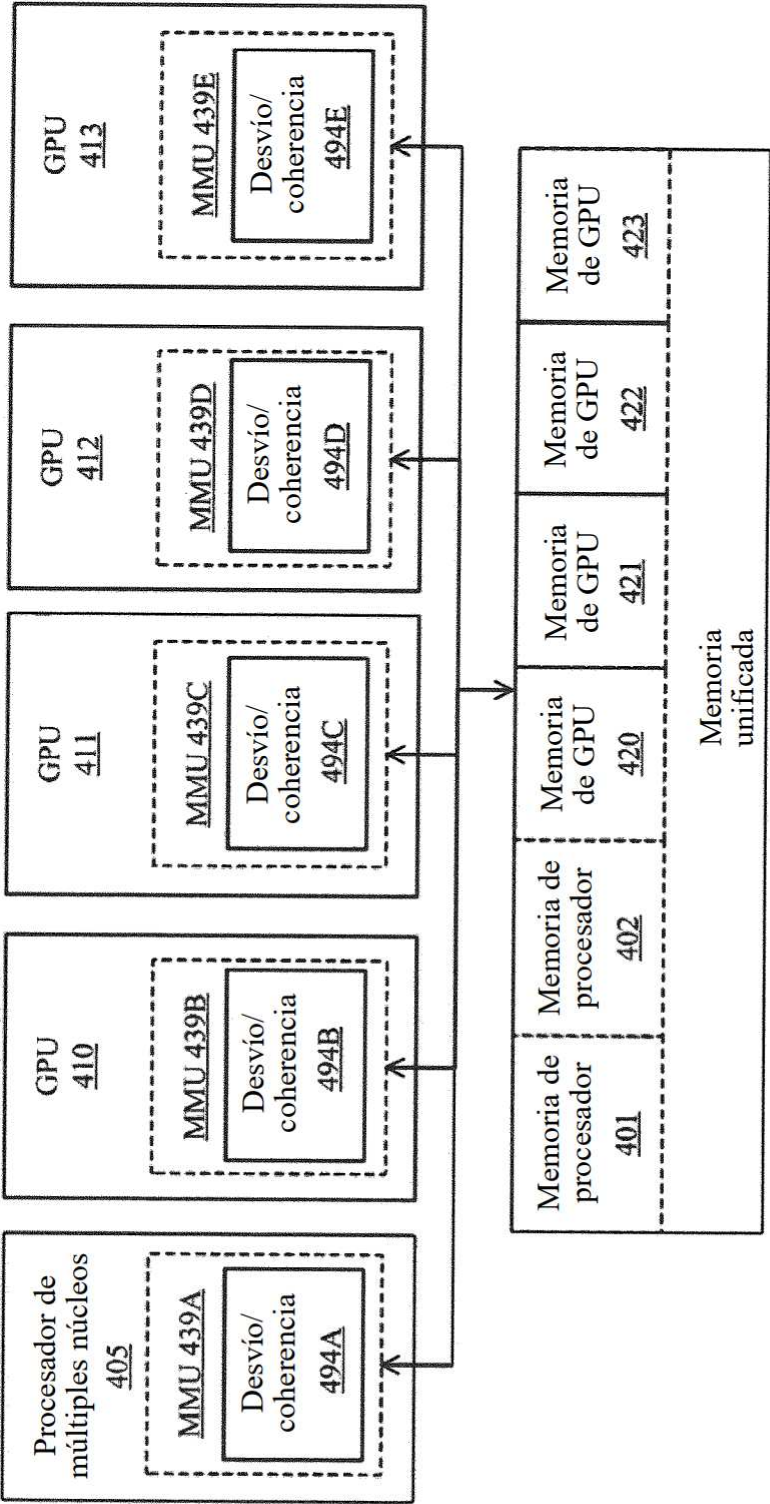


FIG. 4F

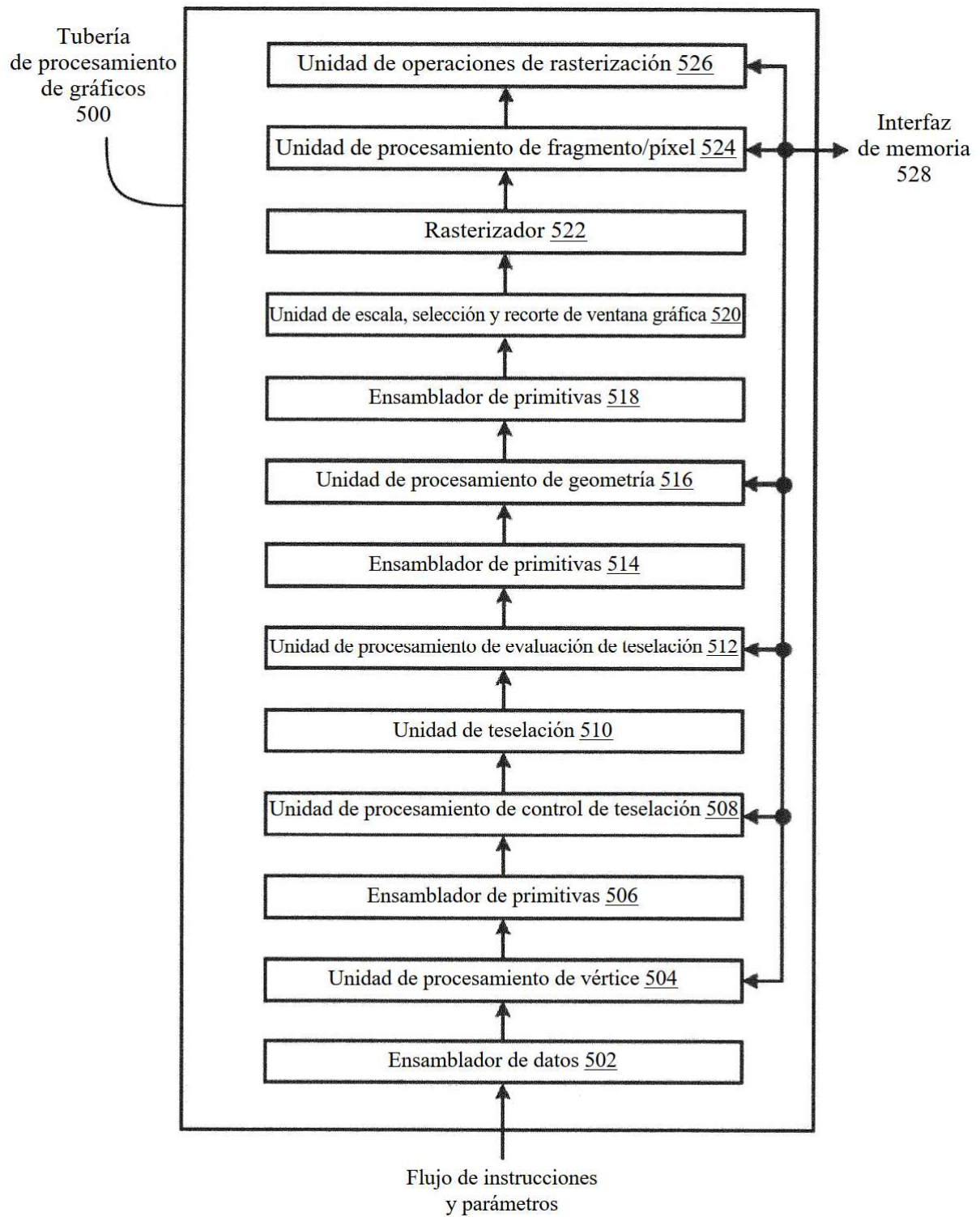


FIG. 5

600

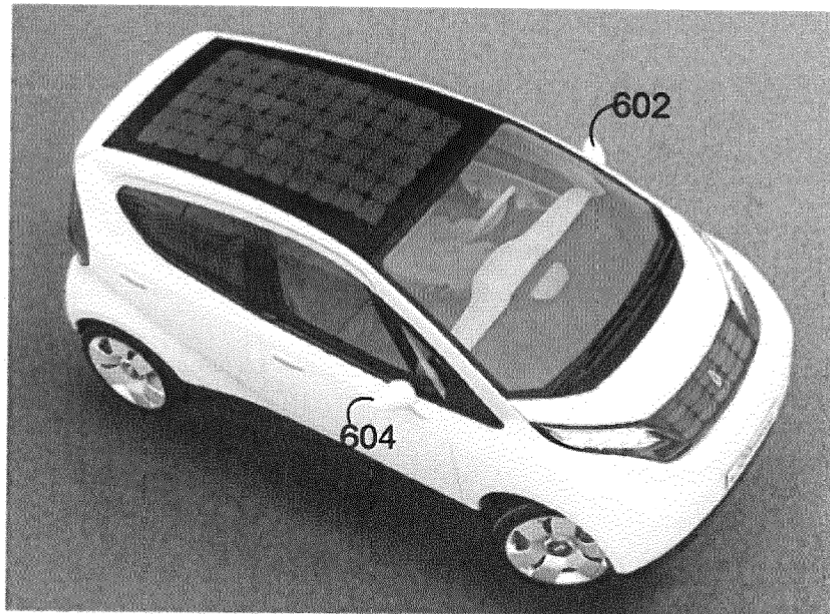


Fig. 6a

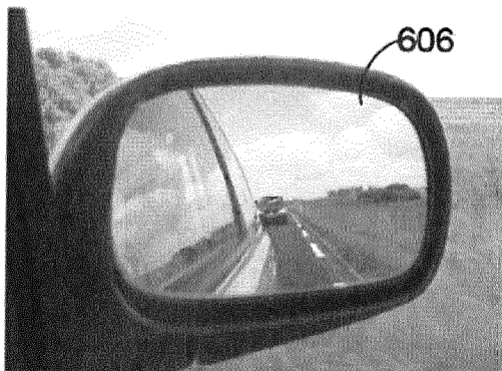


FIG. 6B



FIG. 6C

700



FIG. 7A

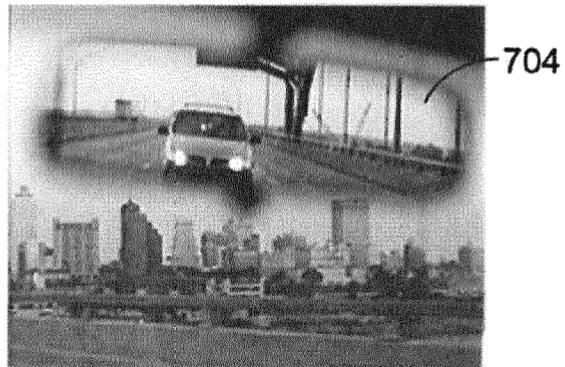


FIG. 7B

800

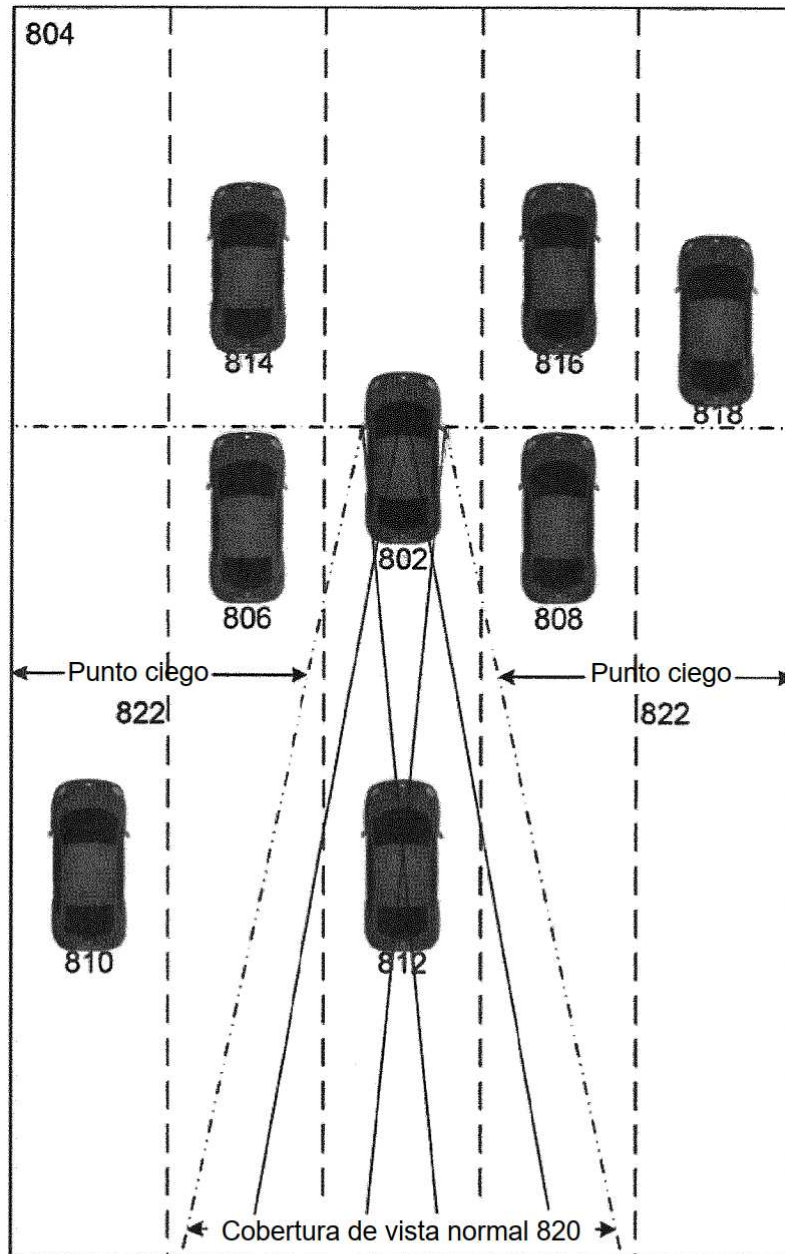


FIG. 8

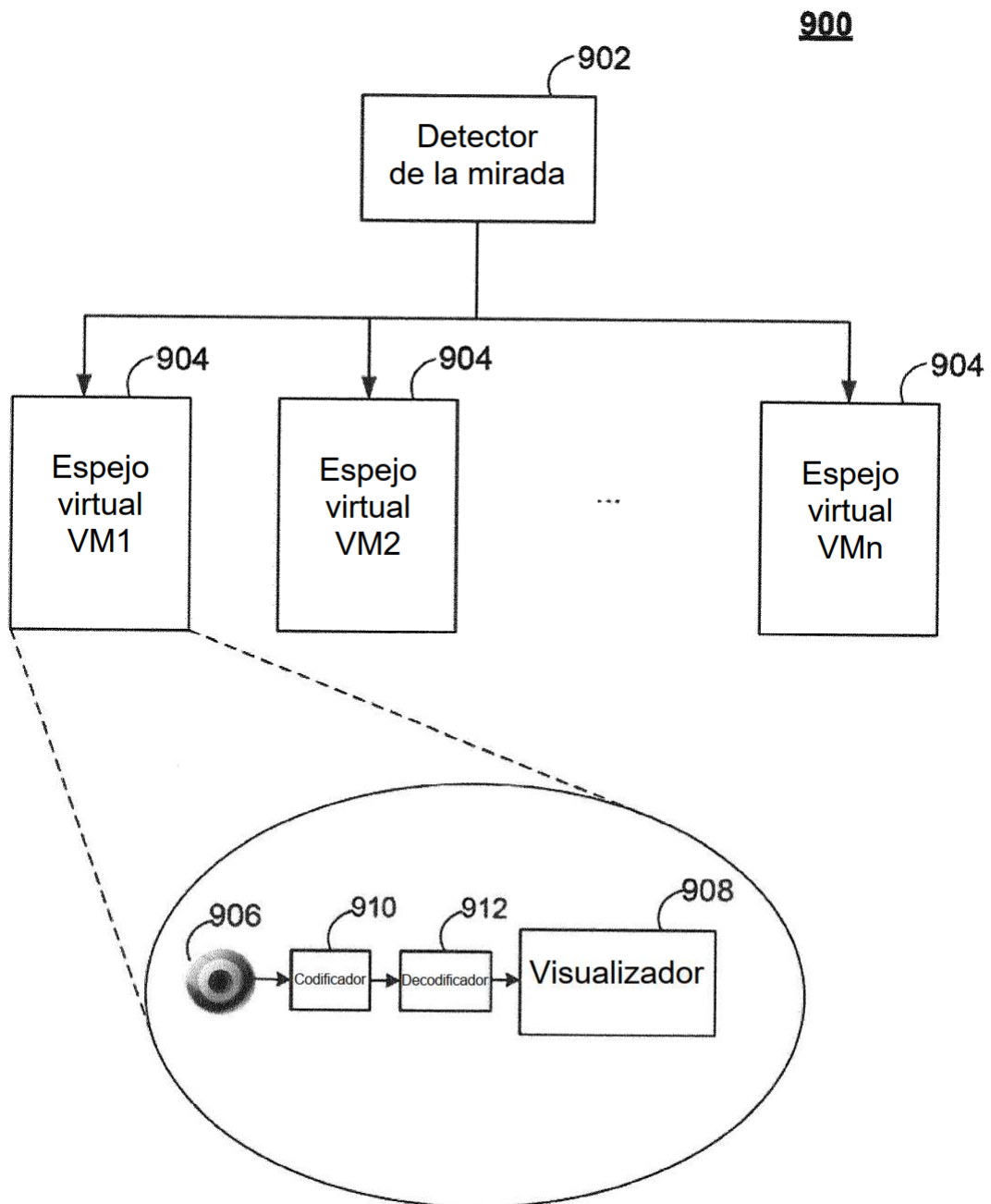


FIG. 9A

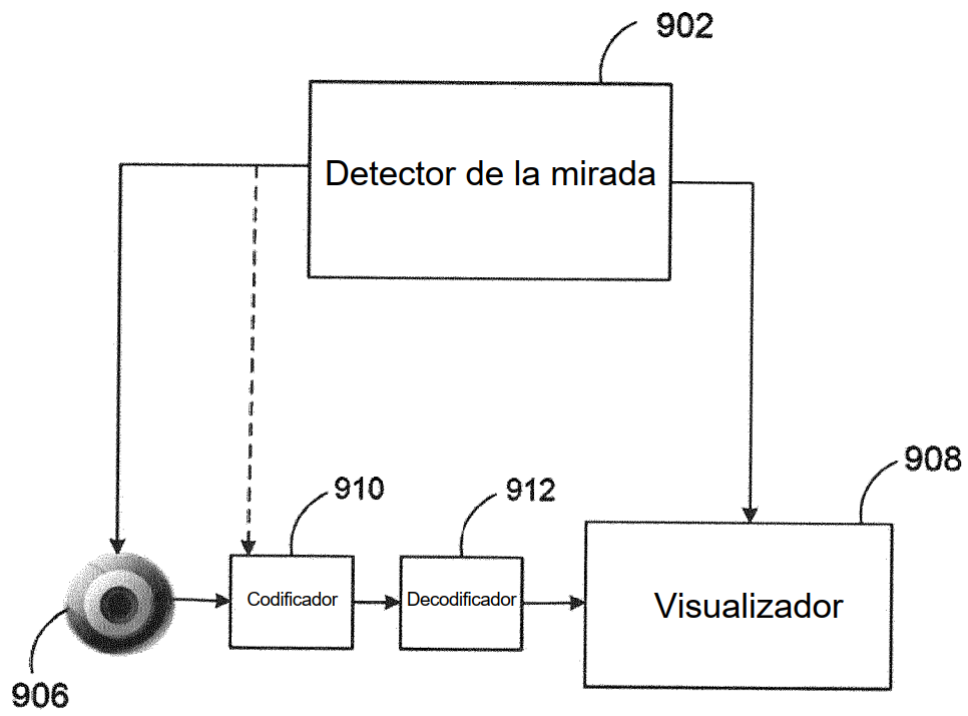


FIG. 9B

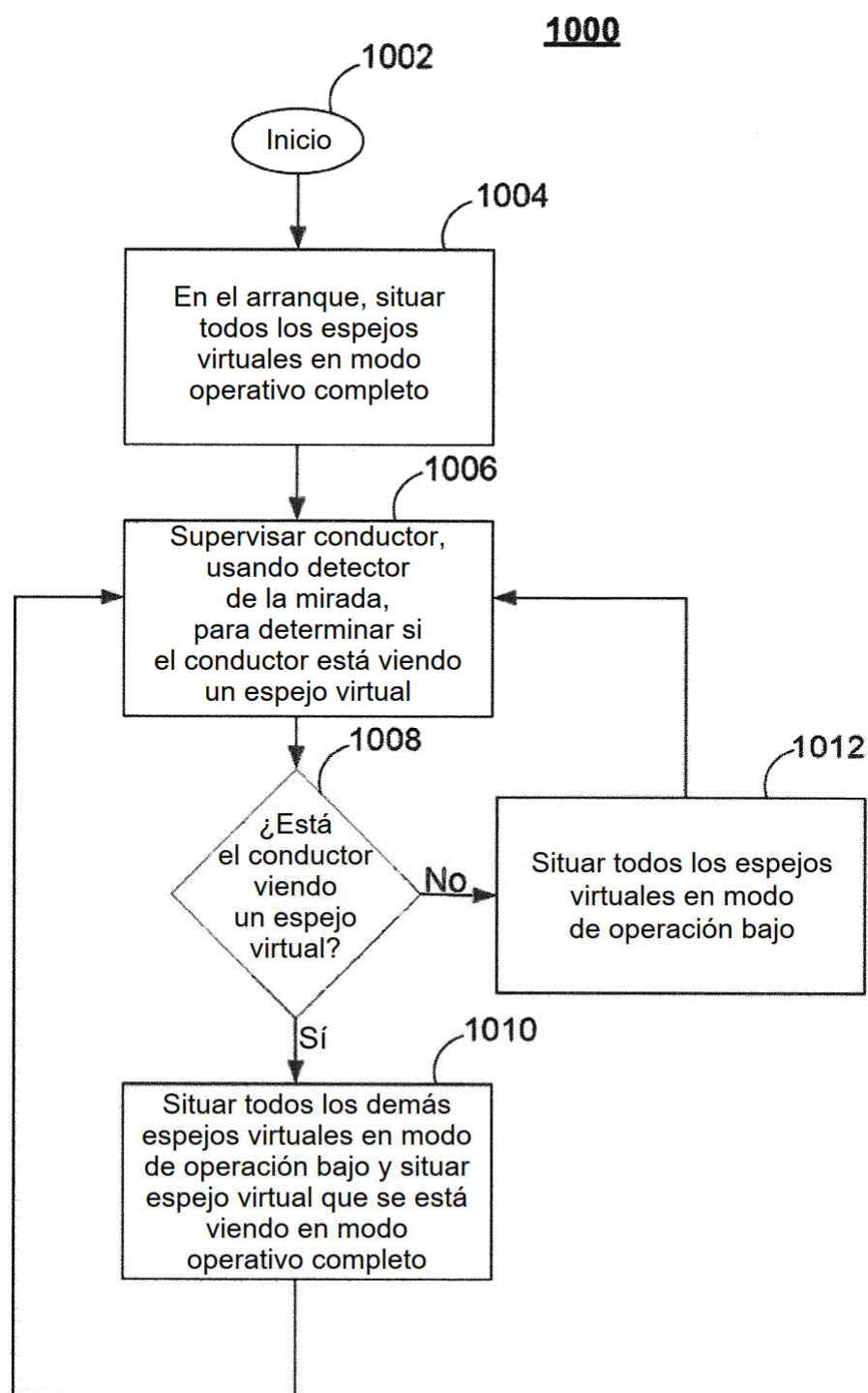


Fig. 10

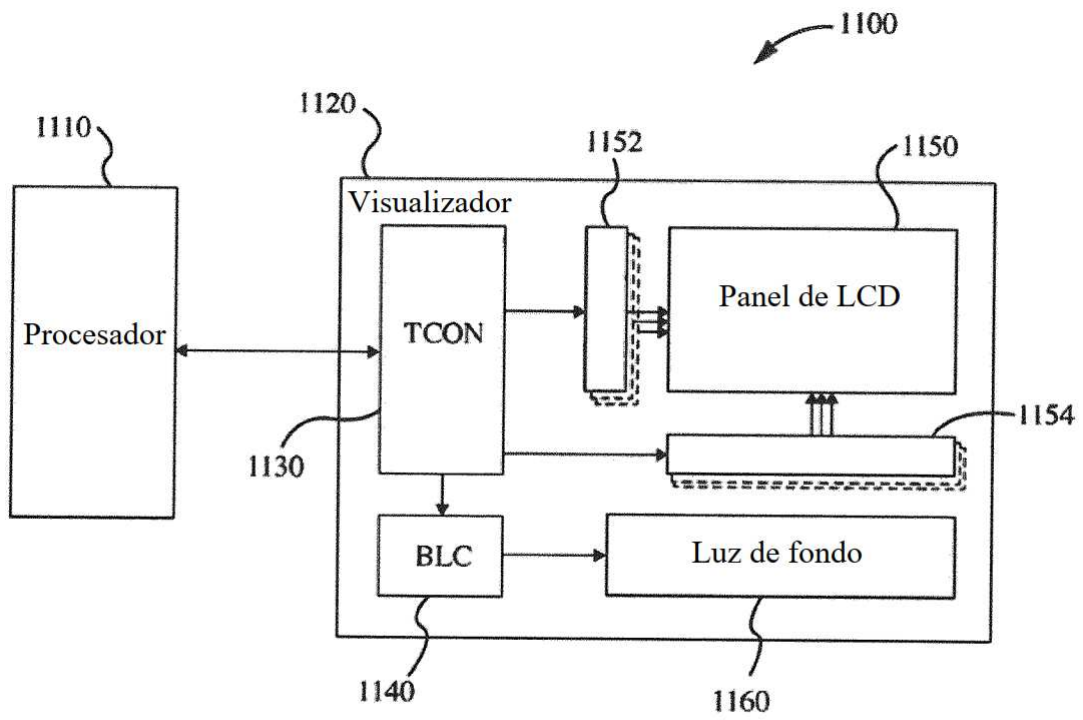


FIG. 11

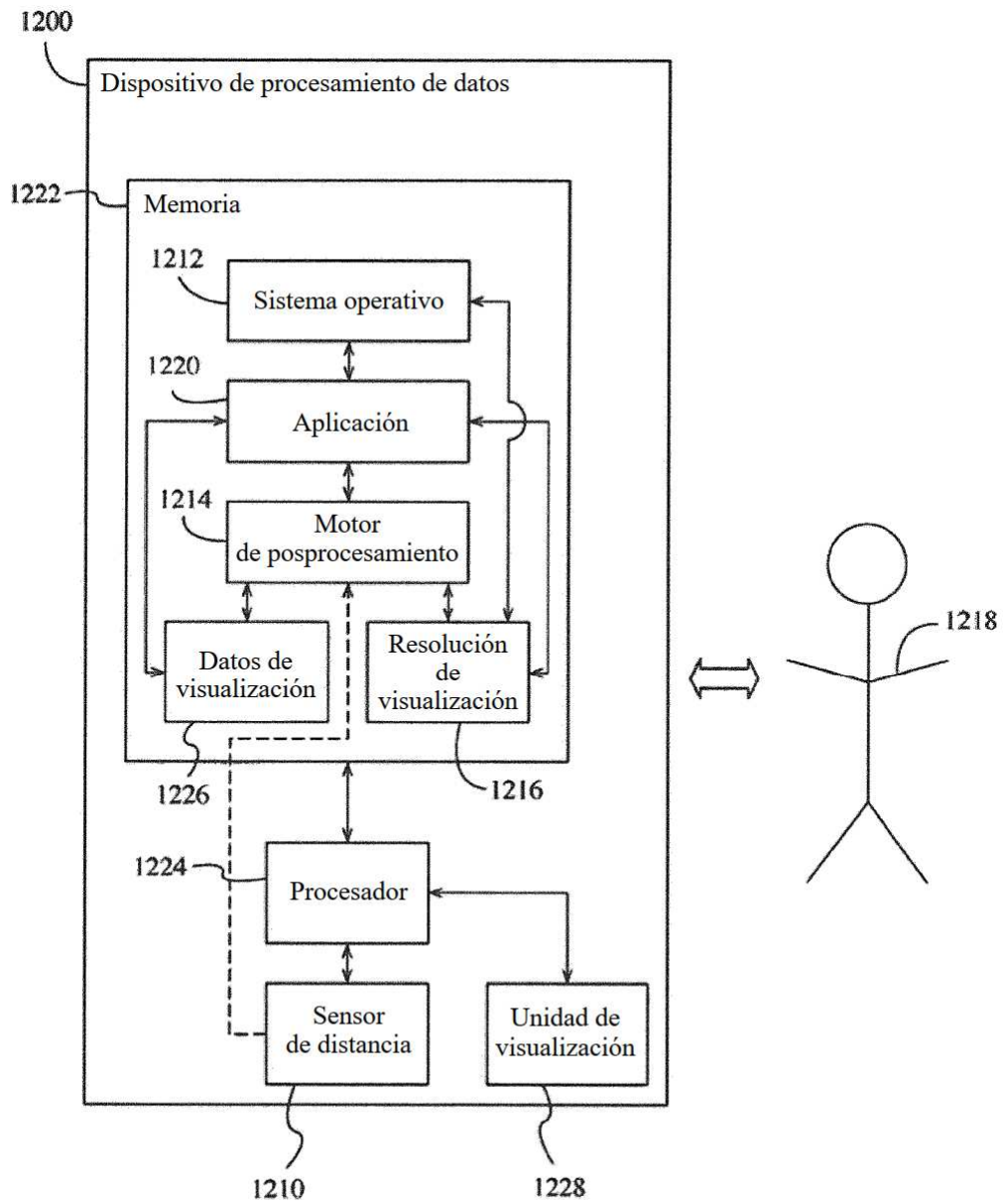


FIG. 12A

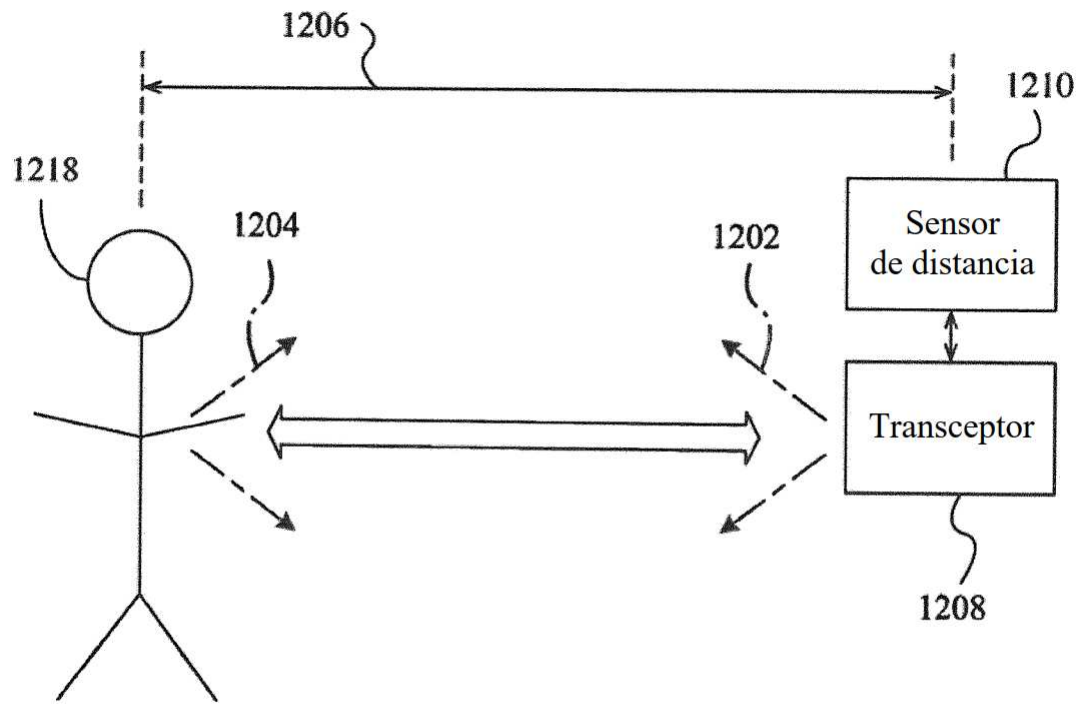


FIG. 12B

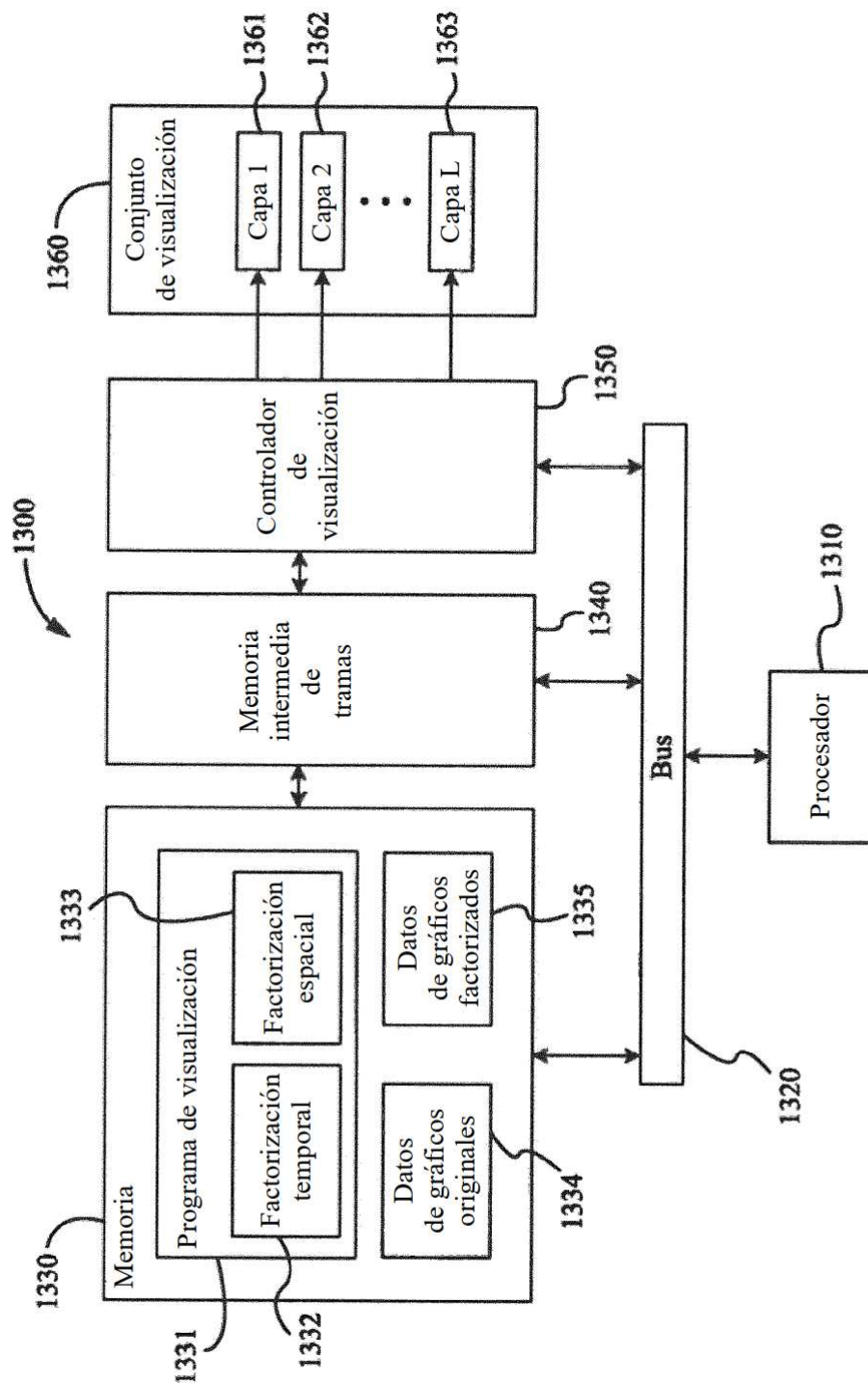


FIG. 13

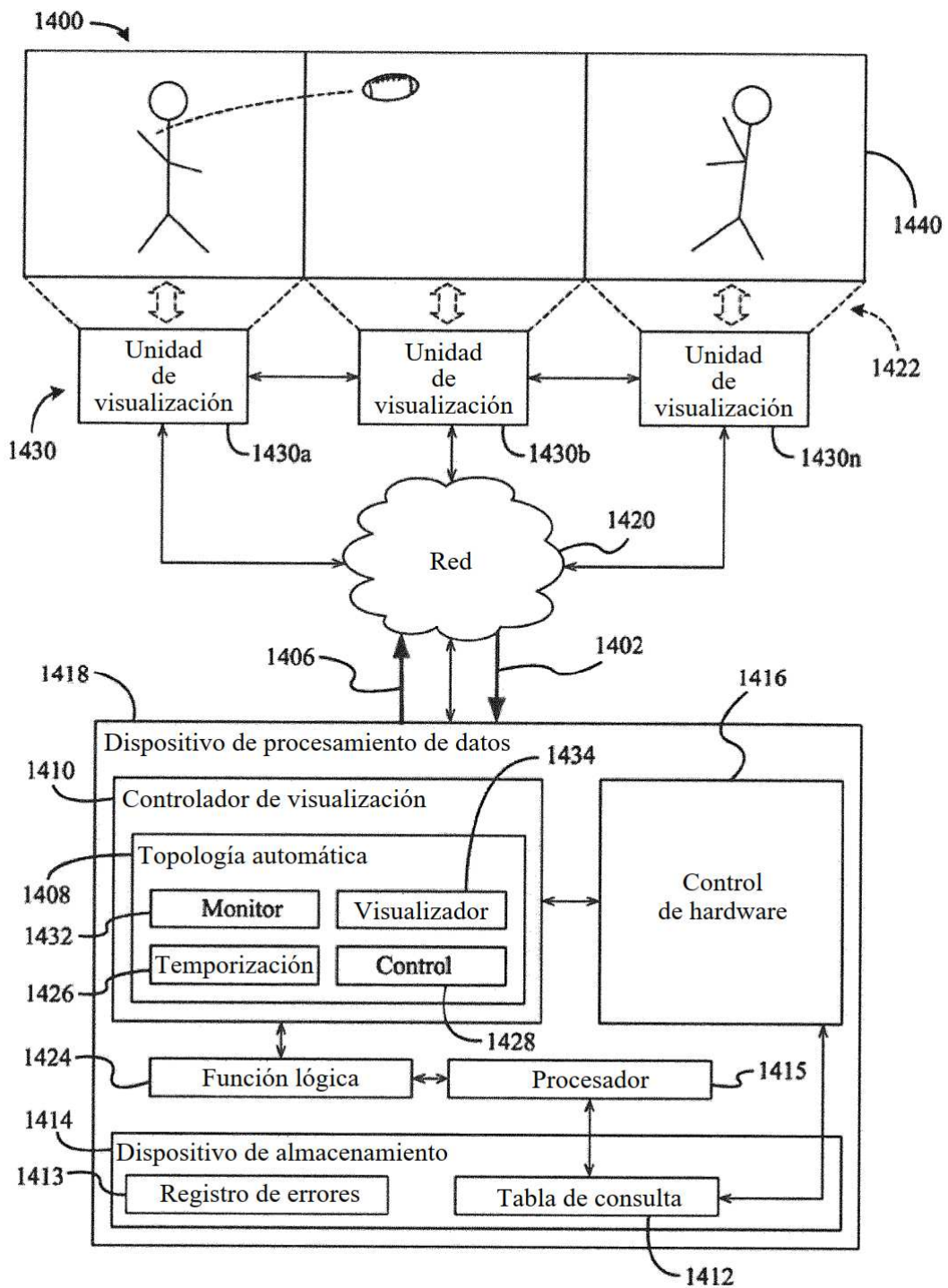


FIG. 14

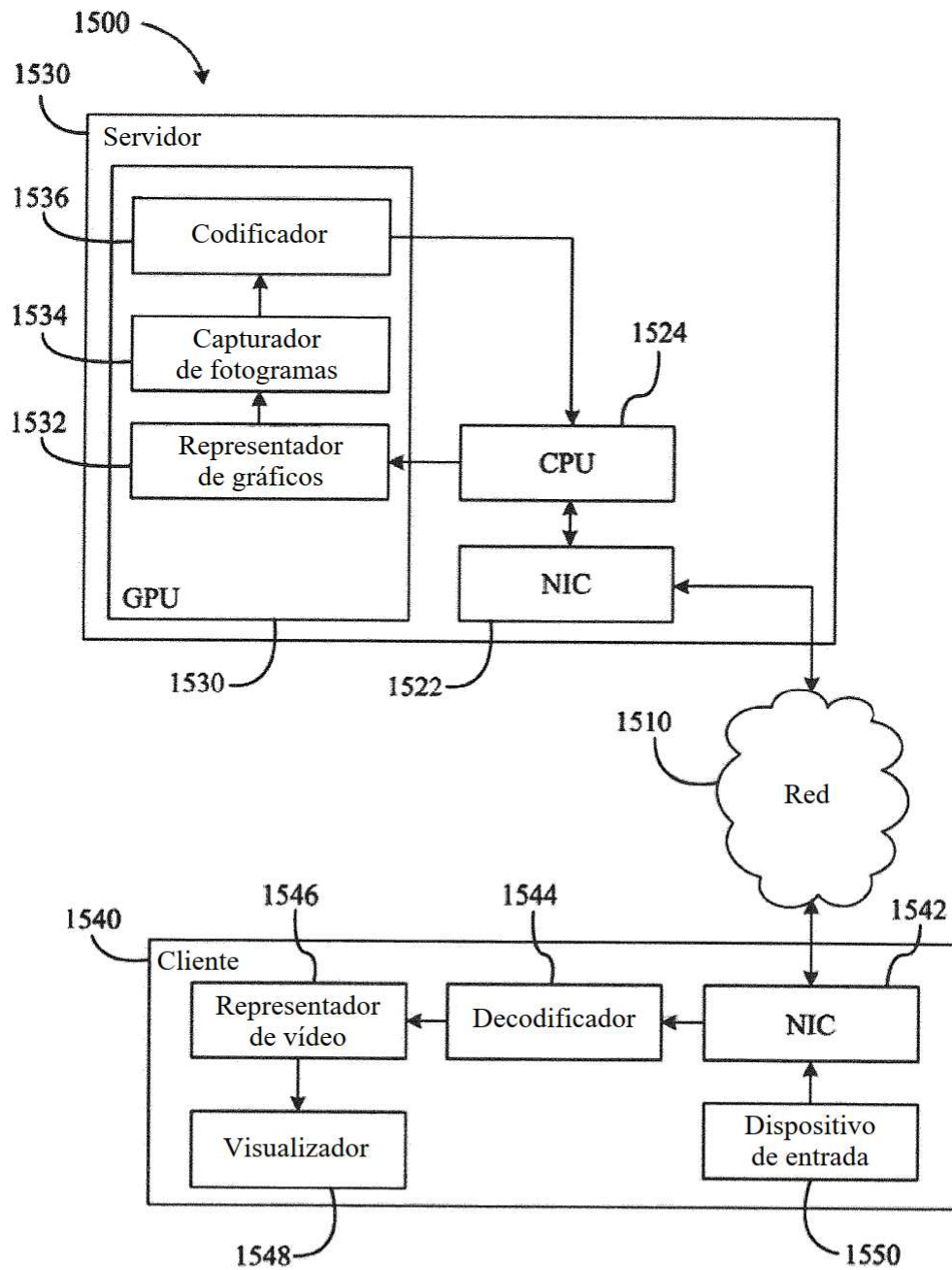


FIG. 15

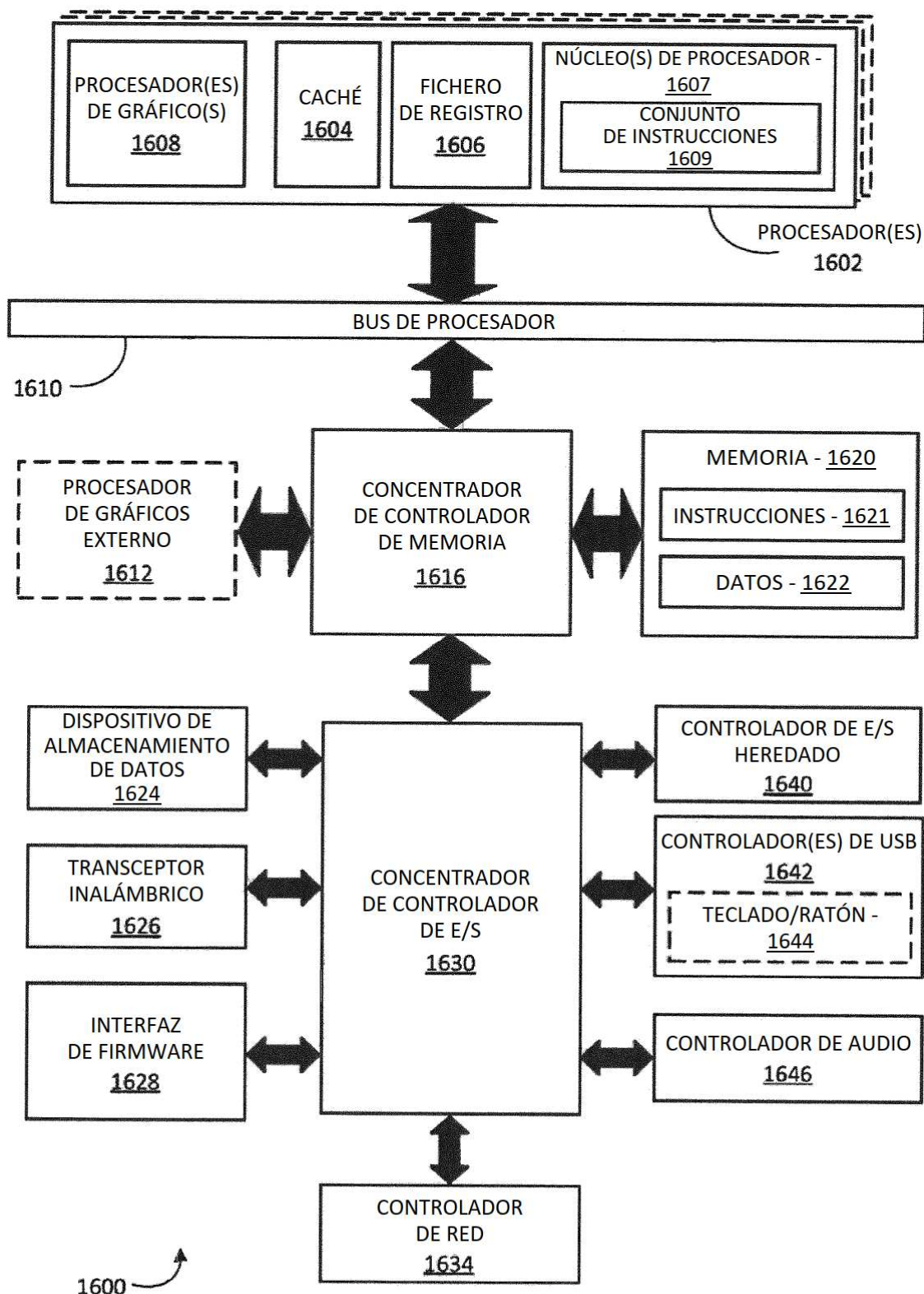


FIG. 16

PROCESADOR 1700 →

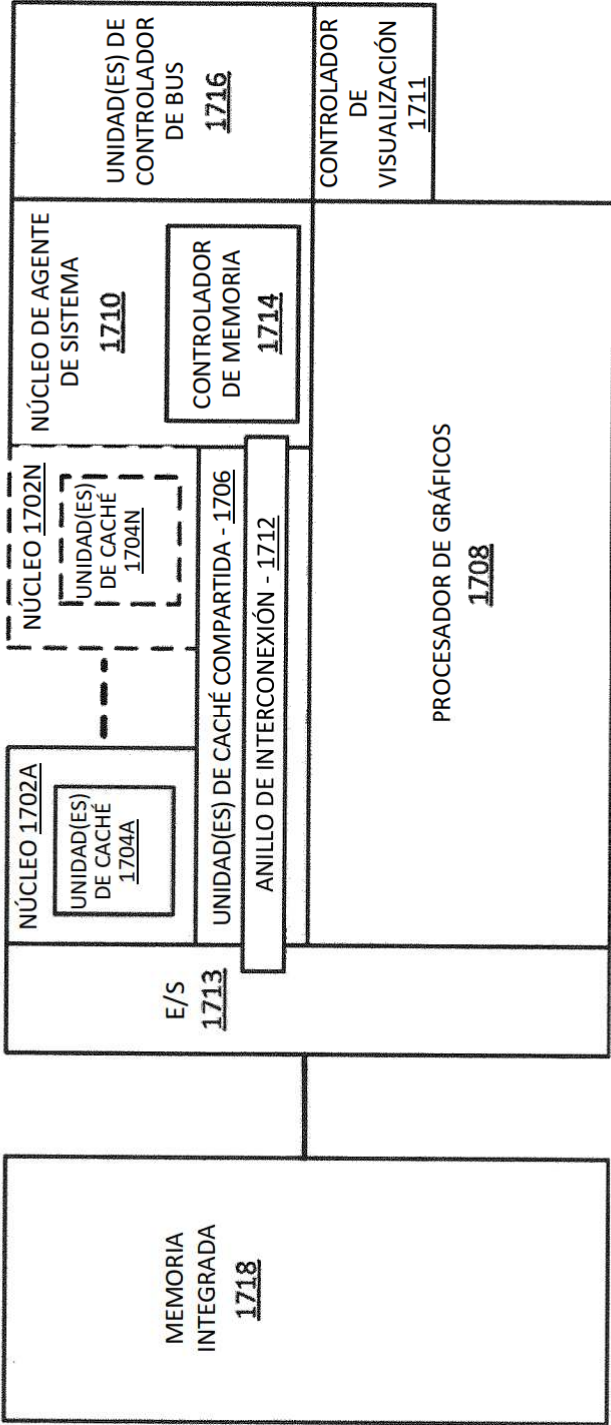


FIG. 17

PROCESADOR DE GRÁFICOS
1800

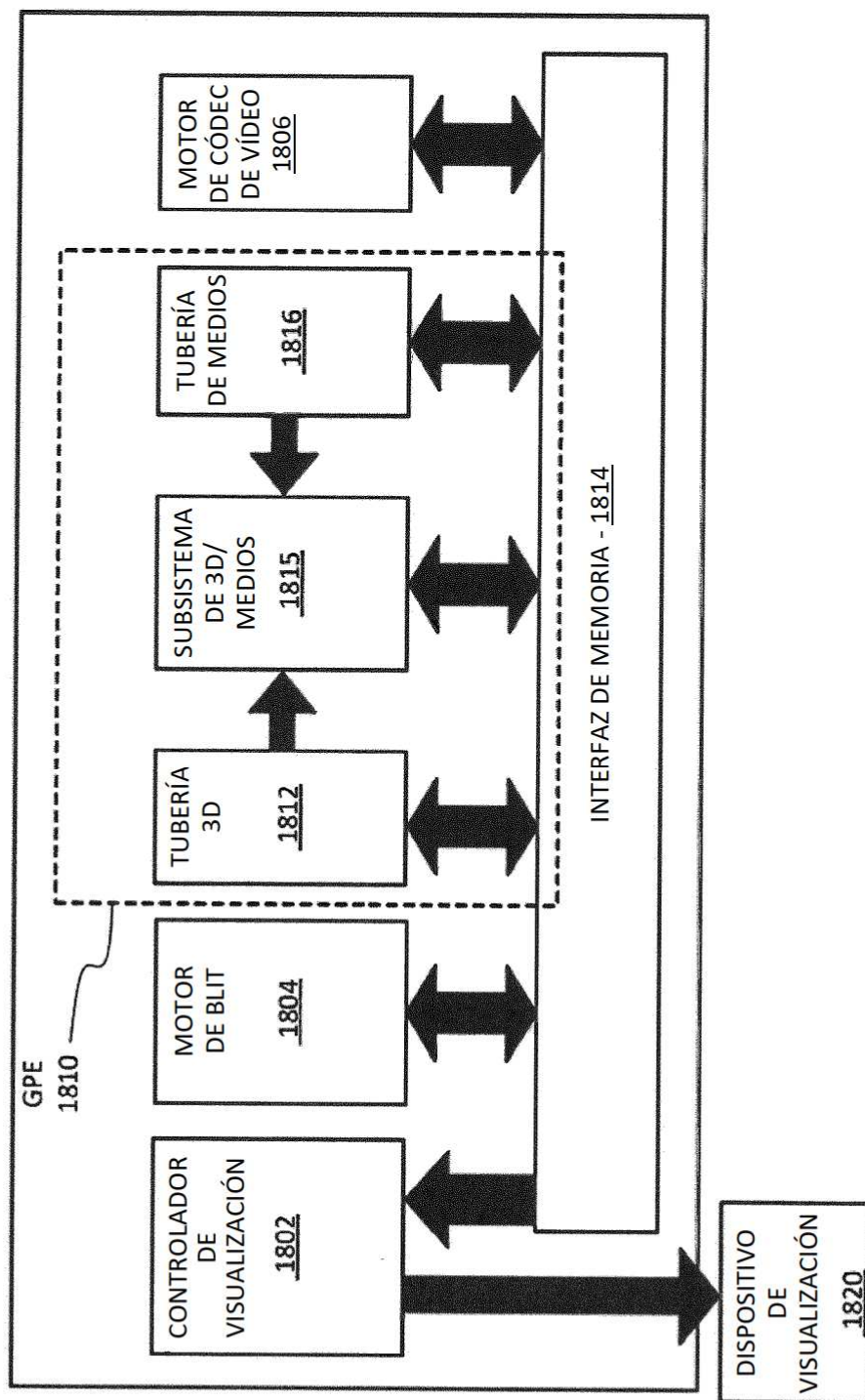
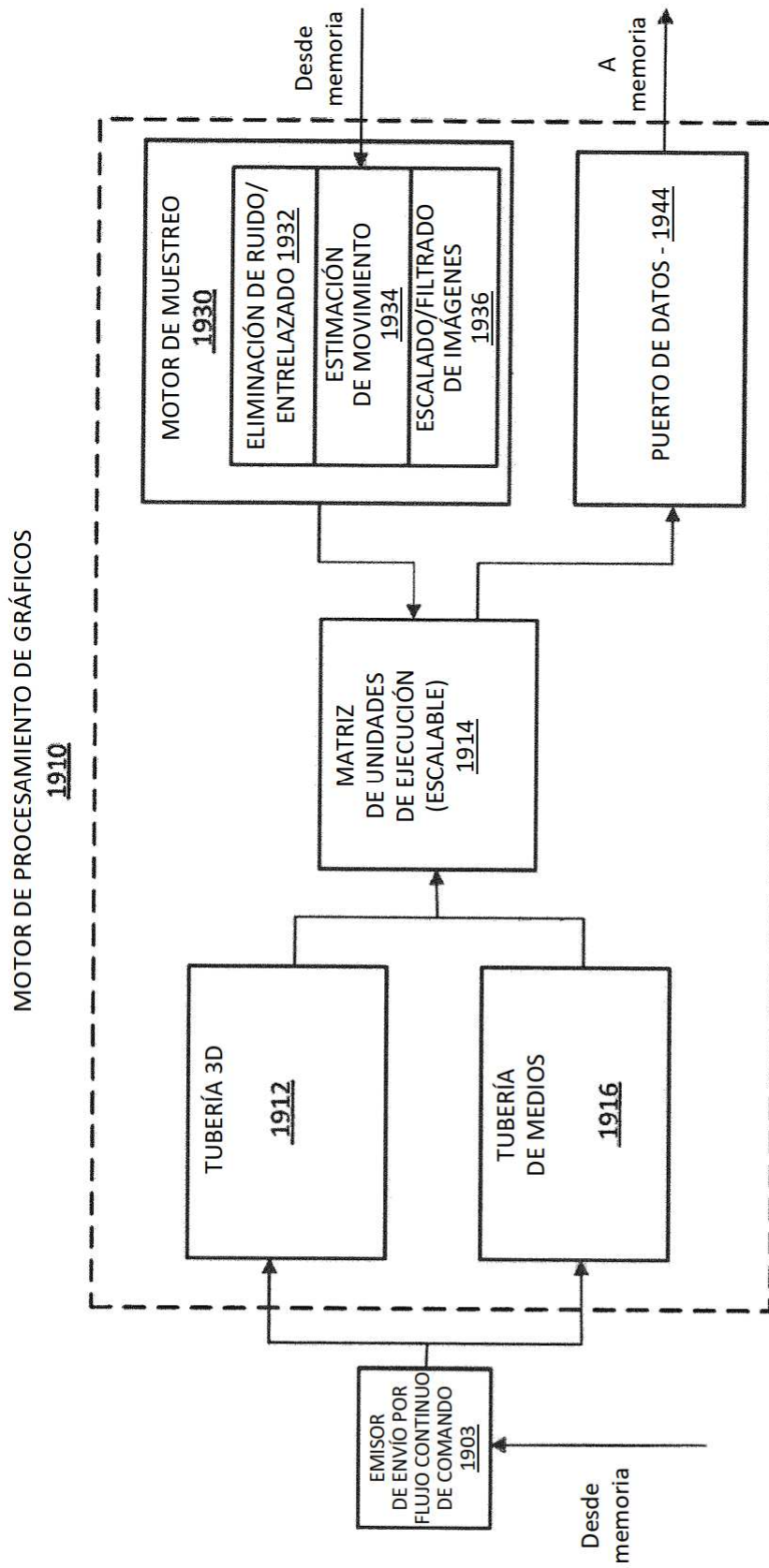


FIG. 18



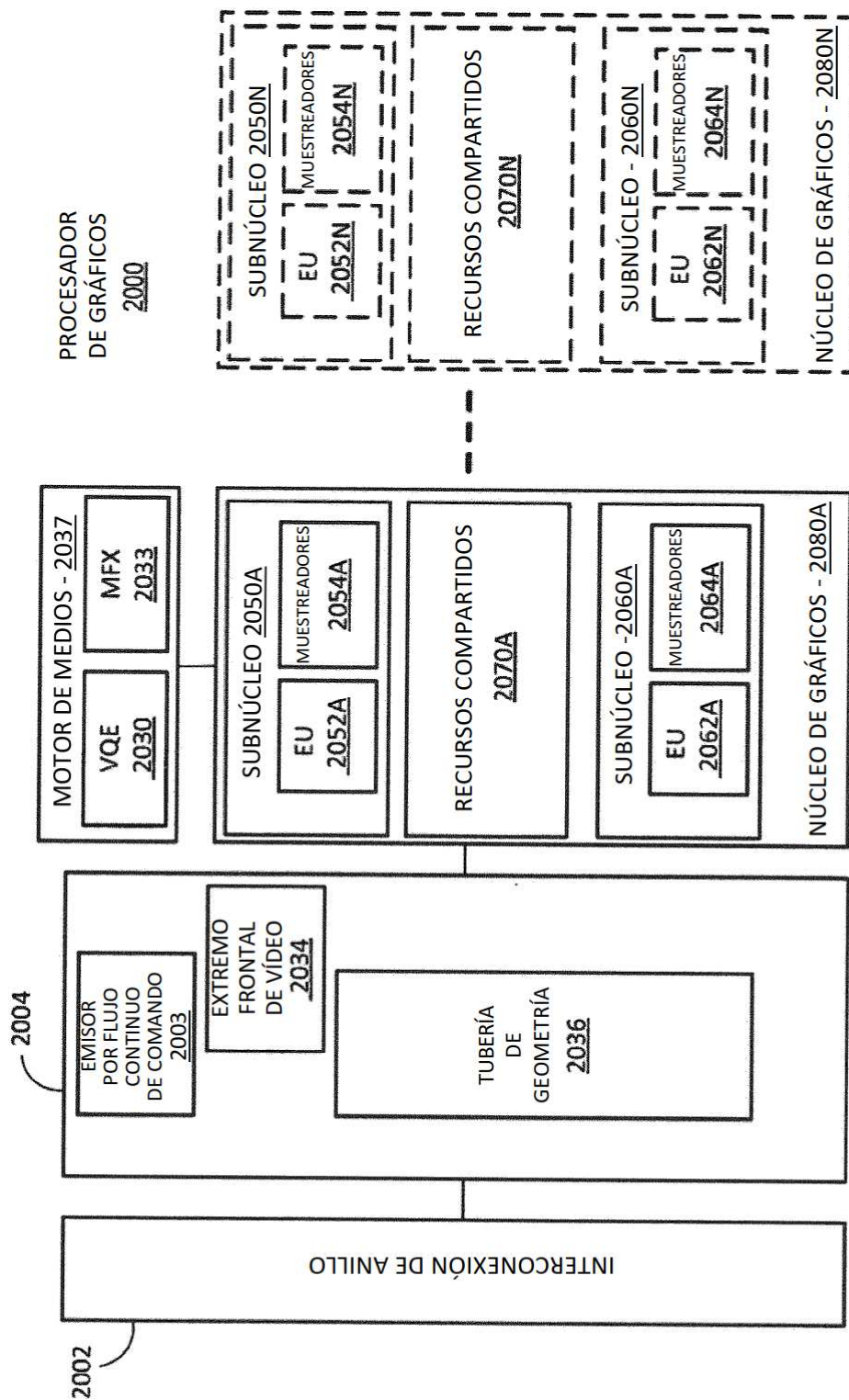


FIG. 20

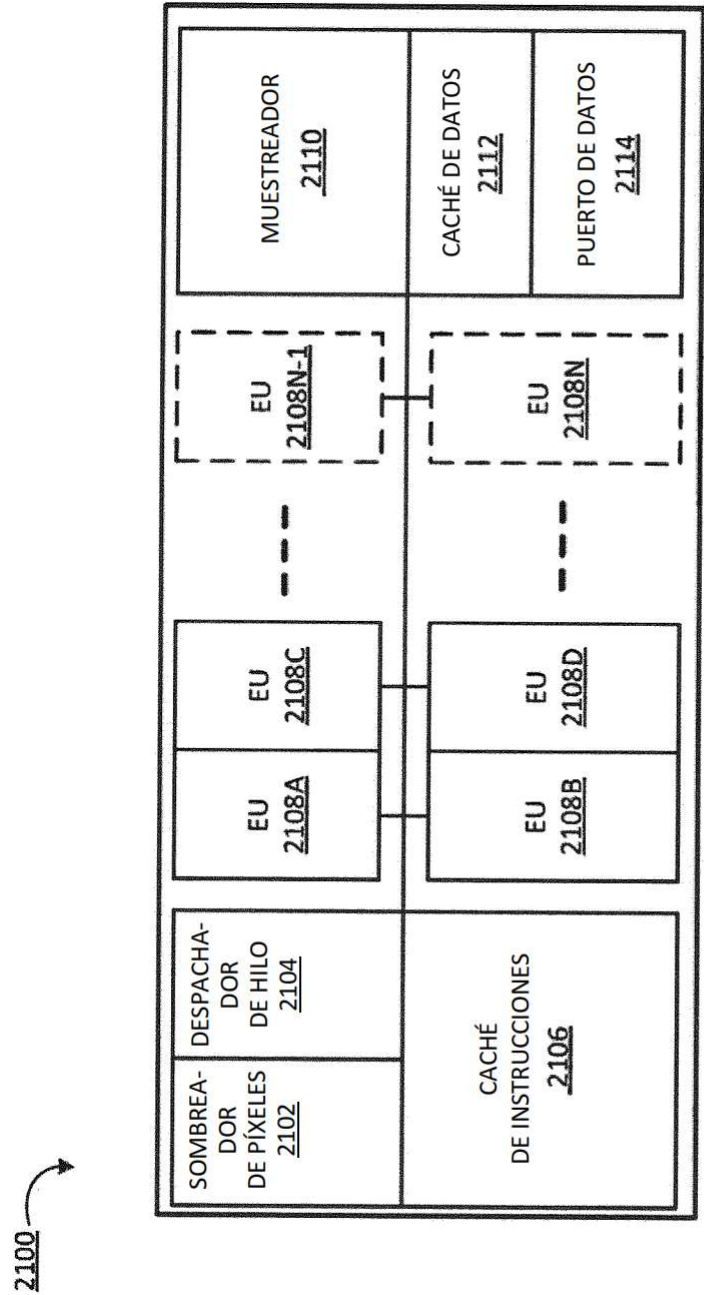
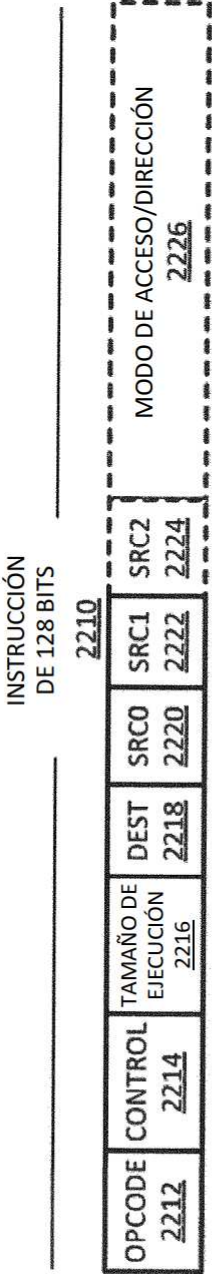


FIG. 21

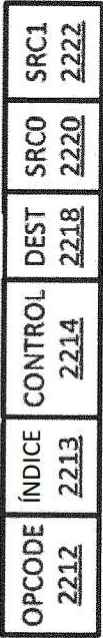
FORMATOS DE INSTRUCCIÓN DE PROCESADOR DE GRÁFICOS

2200



INSTRUCCIÓN COMPACTA DE 64 BITS

2230



DECODIFICACIÓN DE OPCODE

2240

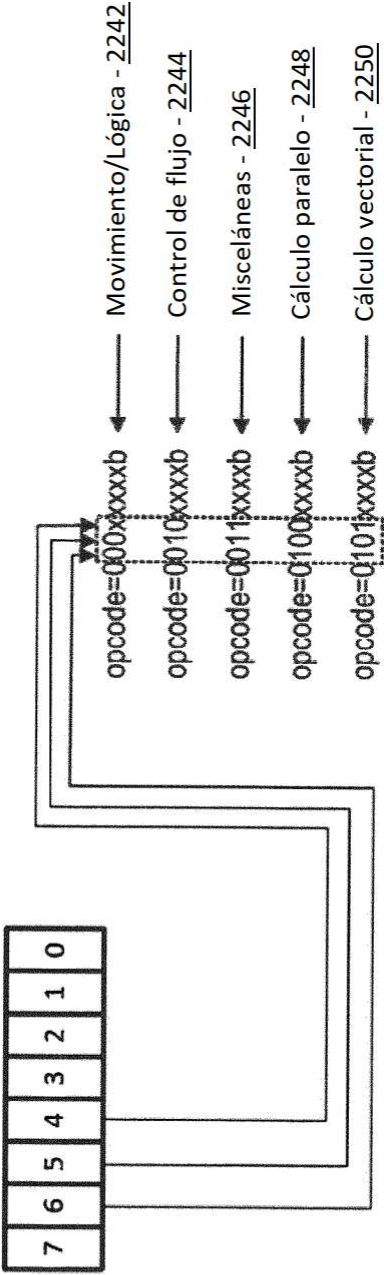


FIG. 22

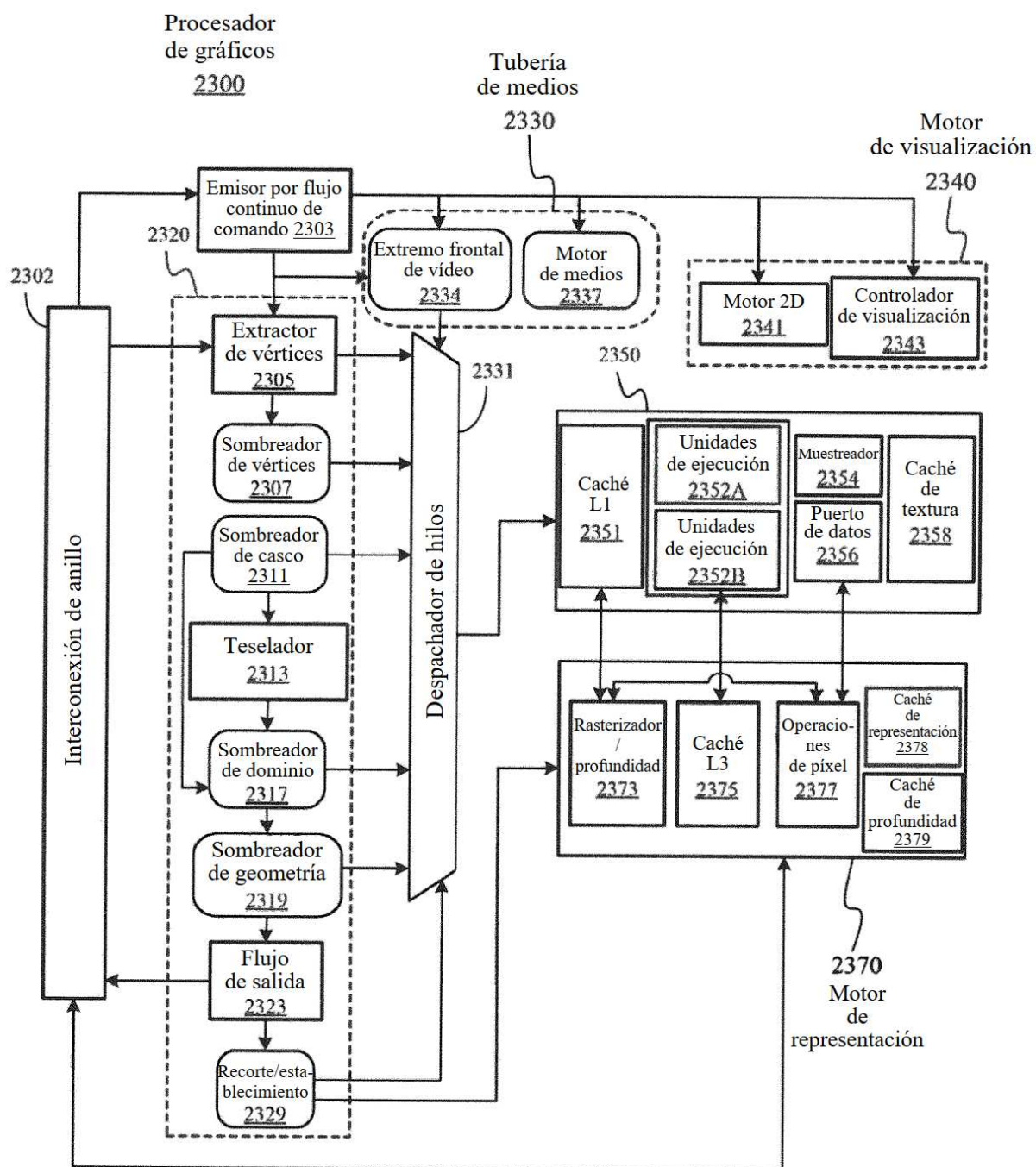


FIG. 23

Formato de comando de procesador de gráficos

2400

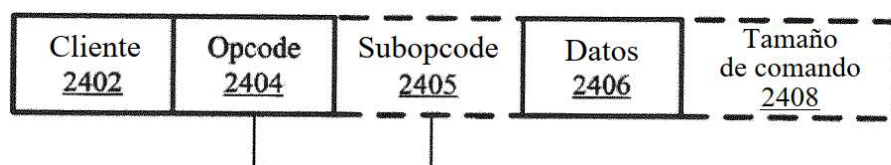


FIG. 24A

Secuencia de comandos de procesador de gráficos

2410

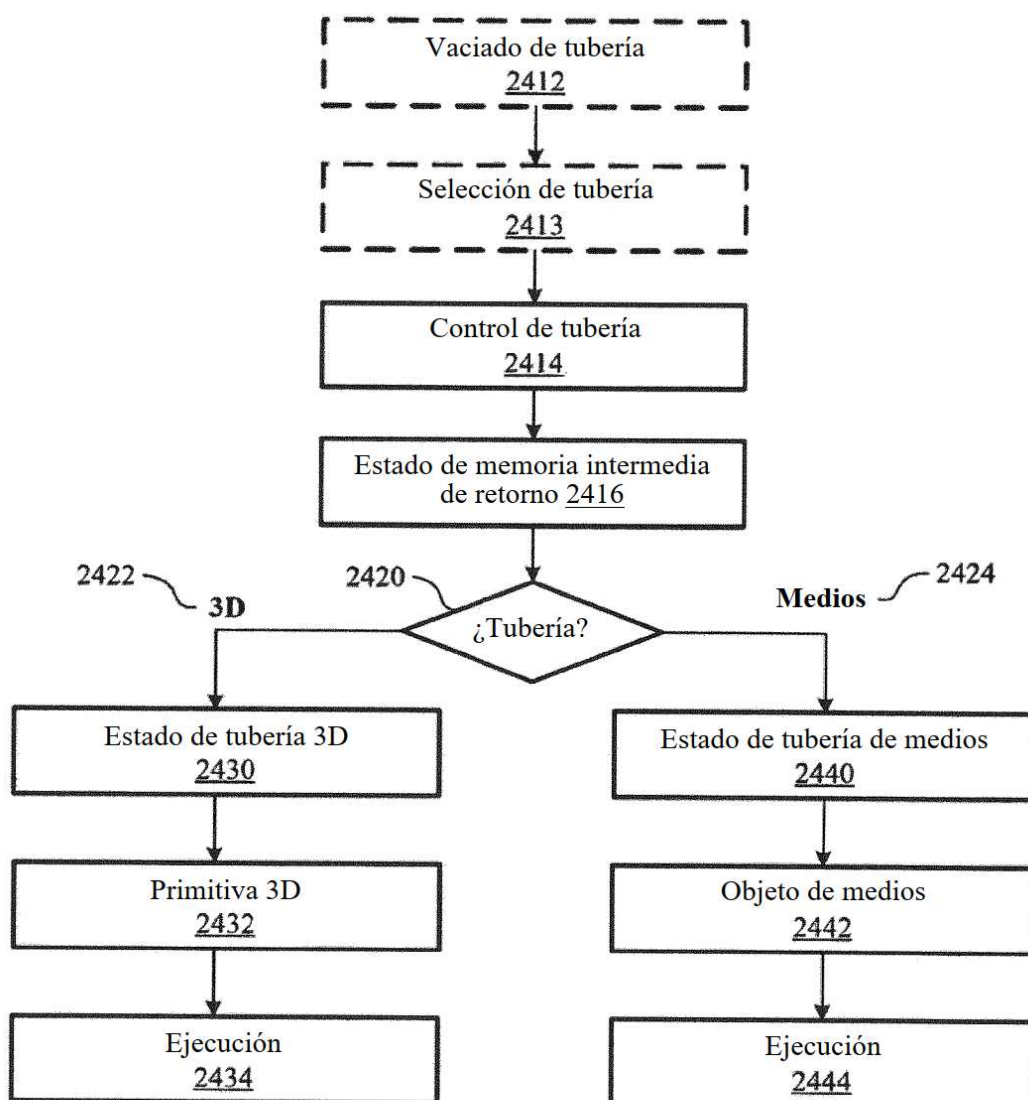


FIG. 24B

Sistema de procesamiento de datos - 2500

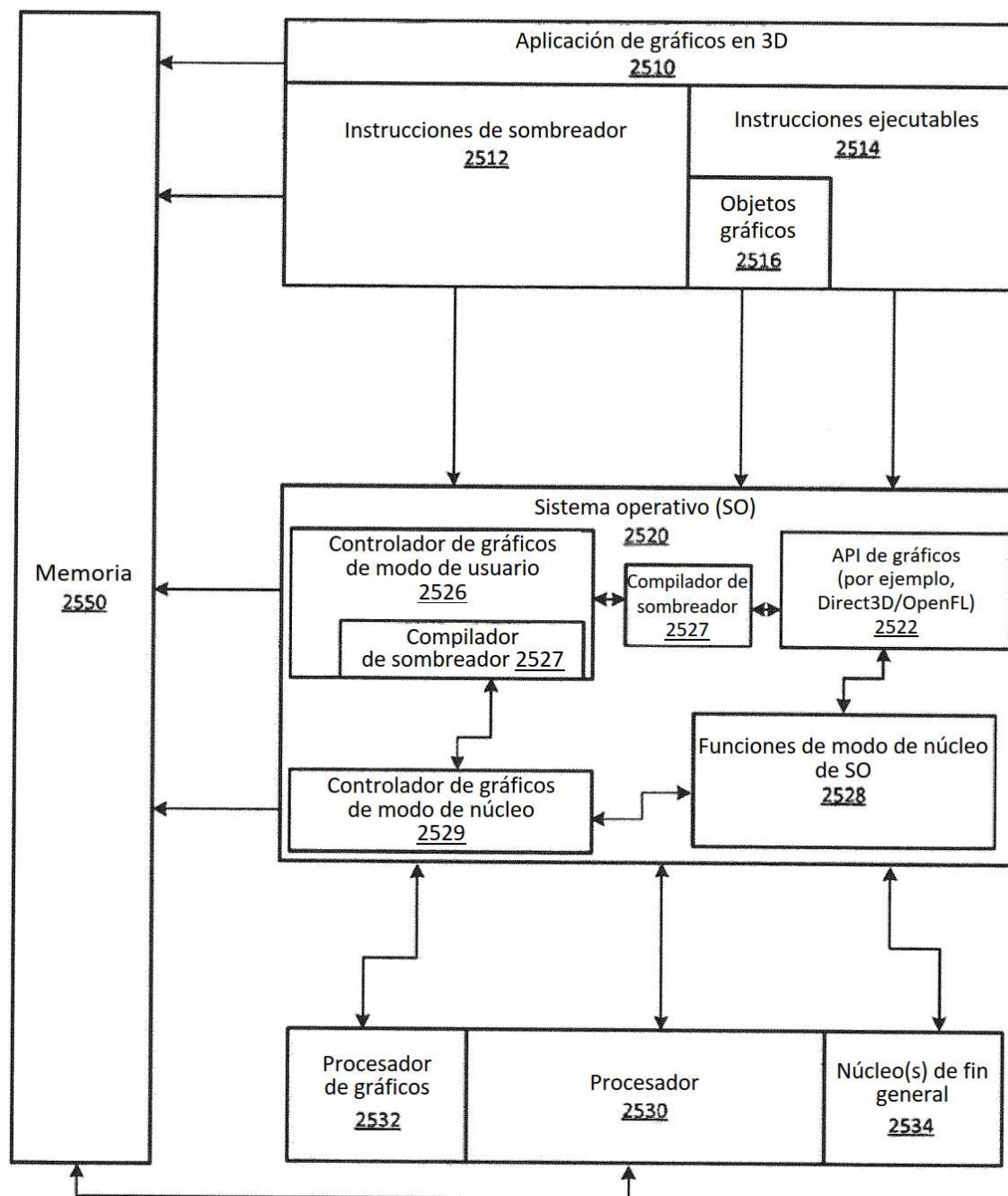


FIG. 25

DESARROLLO DE NÚCLEO DE IP - 2600

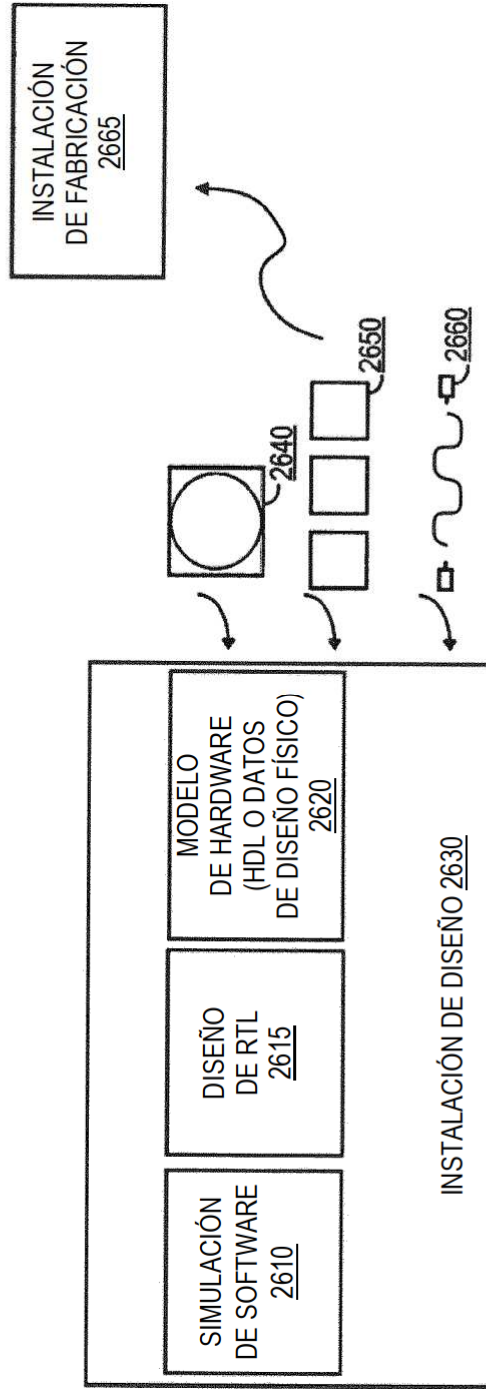


FIG. 26

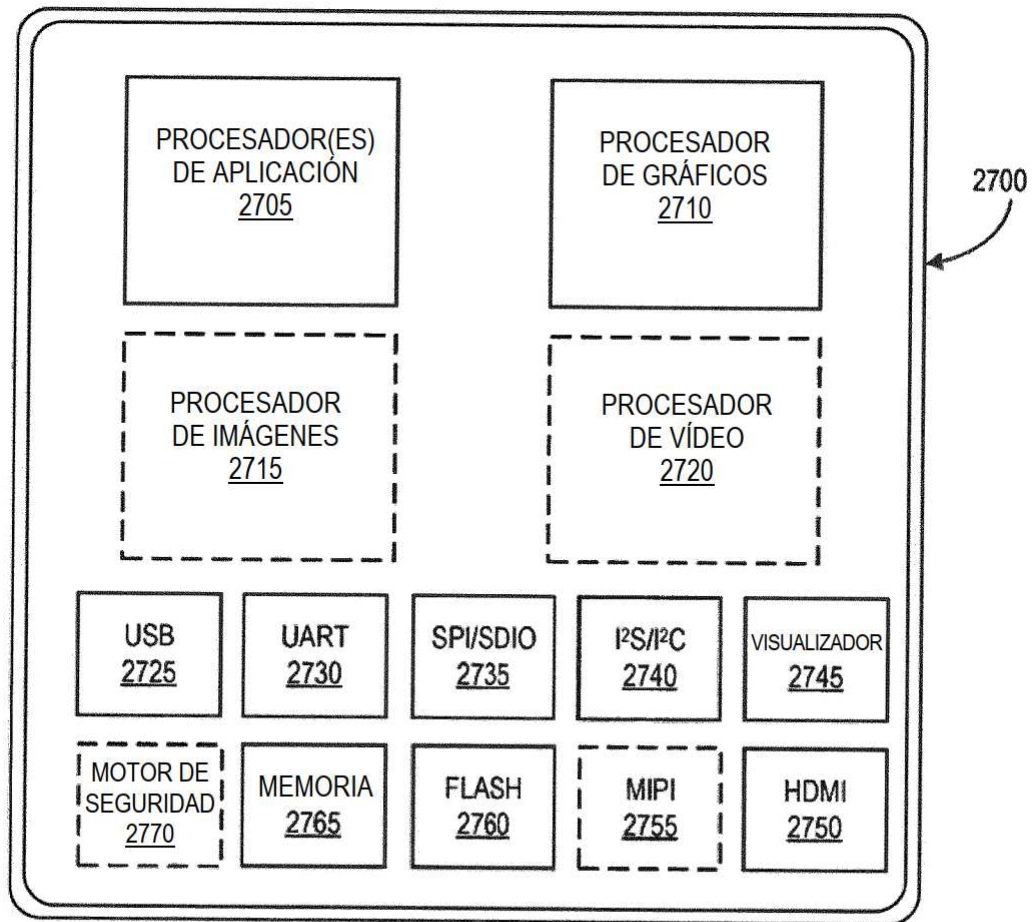


FIG. 27