

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 974 452**

51 Int. Cl.:

G06F 3/01 (2006.01)

G06F 3/048 (2013.01)

G06T 1/20 (2006.01)

G06V 40/16 (2012.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **09.04.2018** **E 18166421 (0)**

97 Fecha y número de publicación de la concesión europea: **10.01.2024** **EP 3388923**

54 Título: **Ajuste de la representación gráfica en función de la expresión facial**

30 Prioridad:

10.04.2017 US 201715483442

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

27.06.2024

73 Titular/es:

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

72 Inventor/es:

**SCHLUESSLER, TRAVIS T.;
RAY, JOYDEEP;
FEIT, JOHN H.;
KABURLASOS, NIKOS;
KWIATKOWSKI, JACEK;
AMSTUTZ, JEFFERSON;
BROWNLIE, CARSON;
TIWARI, VIVEK;
LAHIRI, SAYAN;
XIAO, KAI;
APPU, ABHISHEK R.;
OULD-AHMED-VALL, ELMOUSTAPHA;
VEMBAR, DEEPAK S.;
SHAH, ANKUR N.;
VEMBU, BALAJI y
MASTRONARDE, JOSH B.**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 974 452 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

Ajuste de la representación gráfica en función de la expresión facial

5 **CAMPO TÉCNICO**

Las realizaciones se refieren en general al procesamiento de datos y al procesamiento de gráficos a través de una unidad de procesamiento de gráficos. Más particularmente, las realizaciones se refieren al ajuste de la representación gráfica en función de la expresión facial.

10 **ANTECEDENTES DE LA DESCRIPCIÓN**

El procesamiento paralelo actual de datos gráficos incluye sistemas y métodos desarrollados para realizar operaciones específicas sobre datos gráficos, tales como, por ejemplo, interpolación lineal, teselación, rasterización, mapeo de textura, prueba de profundidad, etc. Tradicionalmente, los procesadores gráficos usan unidades computacionales de función fija para procesar datos gráficos; sin embargo, más recientemente, partes de los procesadores gráficos se han hecho programables, lo que permite que tales procesadores soporten una gama más amplia de operaciones para procesar datos de vértice y de fragmento. Se pueden aplicar varios ajustes, parámetros y configuraciones a las operaciones con datos gráficos.

La fovea se puede referir a una pequeña depresión en la retina del ojo donde la agudeza visual puede ser mayor. El centro del campo de visión puede centrarse en esta región, donde los conos de la retina pueden estar particularmente concentrados. En el contexto de algunas aplicaciones gráficas, una fovea o un área foveada puede corresponder a un área de enfoque en una imagen o visualización.

El documento US2015235084 A1 divulga el ajuste de parámetros gráficos de una escena gráfica mostrada en función de una expresión facial detectada del usuario. Se determina si el usuario está experimentando dificultades para resolver visualmente el contenido, en función de un parámetro facial obtenido. La visualización se reconfigura para mejorar las condiciones de visualización mediante el uso de nuevas configuraciones de visualización para cambiar la forma en que se muestra el contenido.

El documento US2014049563 A1 divulga un visualizador y un módulo de determinación de expresión facial. El tamaño del contenido mostrado y el brillo de visualización se pueden ajustar en función de la expresión facial.

El documento US2007159470 A1 divulga el ajuste automático de parámetros de visualización, tales como brillo, contraste, fuente, tamaño de fuente, en función del rendimiento visual. El sistema de detección del rendimiento visual puede ser un sistema de detección de distancia de visión, un sistema de detección de velocidad de parpadeo o un sistema de seguimiento ocular.

El documento US2016091877 A1 divulga el control de visualización y de las condiciones ambientales a través de un sistema informático montado en la cabeza en función del nivel de fatiga visual detectado.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

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

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

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

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

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

la Figura 6A es un diagrama de bloques de un ejemplo de un sistema de procesamiento electrónico de acuerdo con una realización;

la Figura 6B es un diagrama de bloques de un ejemplo de un motor sensor de acuerdo con una realización;

la Figura 6C es un diagrama de bloques de un ejemplo de un motor de enfoque de acuerdo con una realización;

5 la Figura 6D es un diagrama de bloques de un ejemplo de un motor de movimiento de acuerdo con una realización;

la Figura 7 es un diagrama de bloques de un ejemplo de un aparato gráfico de acuerdo con una realización;

10 las Figuras 8A a 8F son diagramas de flujo de un ejemplo de un método de ajuste de gráficos de acuerdo con una realización;

la Figura 9 es un diagrama de bloques de un ejemplo de un sistema gráfico de acuerdo con una realización;

15 las Figuras 10A a 10B son diagramas ilustrativos de un ejemplo de ajuste de gráficos en función de una expresión facial de acuerdo con una realización;

la Figura 11 es una ilustración de un ejemplo de un sistema de visualización montado en la cabeza (HMD) de acuerdo con una realización;

20 la Figura 12 es un diagrama de bloques de un ejemplo de los componentes funcionales incluidos en el sistema de HMD de la Figura 11 de acuerdo con una realización;

25 la Figura 13 es un diagrama de bloques de un ejemplo de una agrupación de procesamiento general incluida en una unidad de procesamiento paralelo de acuerdo con una realización;

la Figura 14 es una ilustración conceptual de un ejemplo de una canalización de procesamiento de gráficos que puede implementarse dentro de una unidad de procesamiento paralelo, de acuerdo con una realización;

30 la Figura 15 es un diagrama de bloques de un ejemplo de un multiprocesador de envío por flujo continuo de acuerdo con una realización;

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

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

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

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

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

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

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

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

55 **DESCRIPCIÓN DETALLADA**

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

Descripción general del sistema

65 La Figura 1 es un diagrama de bloques que ilustra un sistema informático 100 configurado para implementar uno o más aspectos de las realizaciones descritas en el presente documento. El sistema informático 100 incluye un

subsistema de procesamiento 101 que tiene uno o más procesadores 102 y una memoria de sistema 104 que se comunica por medio de una ruta de interconexión que puede incluir un concentrador de memoria 105. El concentrador de memoria 105 puede ser un componente separado dentro de un componente de conjunto de chips o puede integrarse dentro de los uno o más procesadores 102. El concentrador de memoria 105 se acopla con un subsistema de E/S 111 mediante un enlace de comunicación 106. El subsistema de E/S 111 incluye un concentrador de E/S 107 que puede permitir que el sistema informático 100 reciba una entrada desde uno o más dispositivos de entrada 108. Adicionalmente, el concentrador de E/S 107 puede permitir que un controlador de visualización, que puede incluirse en los uno o más procesadores 102, proporcione salidas para uno o más dispositivos de visualización 110A. En una realización, el uno o más dispositivos de visualización 110A acoplados al concentrador de E/S 107 pueden incluir un dispositivo de visualización local, interno o embebido.

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

Dentro del subsistema de E/S 111, una unidad de almacenamiento de sistema 114 puede conectarse al concentrador de E/S 107 para proporcionar un mecanismo de almacenamiento para el sistema informático 100. Se puede usar un conmutador de E/S 116 para proporcionar un mecanismo de interfaz para permitir conexiones entre el concentrador de E/S 107 y otros componentes, tales como un adaptador de red 118 y/o un adaptador de red inalámbrica 119 que pueden estar integrados en la plataforma, y otros diversos dispositivos que puedan añadirse por medio de uno o más dispositivos suplementarios 120. El adaptador de red 118 puede ser un adaptador de Ethernet u otro adaptador de red cableada. El adaptador de red inalámbrica 119 puede incluir uno o más dispositivos de red Wi-Fi, Bluetooth, comunicación de campo cercano (NFC) u otro que incluya una o más radios inalámbricas.

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

En una realización, el uno o más procesadores paralelos 112 incorporan circuitos optimizados para procesamiento de gráficos y vídeo, incluyendo, por ejemplo, circuitería de salida de vídeo, y constituyen una unidad de procesamiento de gráficos (GPU). En otra realización, los uno o más procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de propósito general, mientras conservan la arquitectura computacional subyacente, descrita en mayor detalle en el presente documento. En otra realización más, los componentes del sistema informático 100 pueden integrarse con uno o más elementos del sistema en un único circuito integrado. Por ejemplo, el uno o más procesadores paralelos 112, el concentrador de memoria 105, el o los procesadores 102 y el concentrador de E/S 107 se pueden integrar en un circuito integrado de sistema en chip (SoC). Como alternativa, los componentes del sistema informático 100 pueden integrarse en un único paquete para formar una configuración de sistema en paquete (SIP). En una realización, al menos una parte de los componentes del sistema informático 100 se puede integrar en un módulo multichip (MCM), que se puede interconectar con otros módulos multichip 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, incluyendo el número y disposición de puentes, el número de procesadores 102 y el número de procesadores paralelos 112 se puede modificar como se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 está conectada al o a los 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 o los procesadores 102. En otras topologías alternativas, el o los procesadores paralelos 112 están conectados al concentrador de E/S 107 o directamente a uno del uno o más 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 se pueden integrar en un único microprocesador. Algunas realizaciones pueden incluir dos o más conjuntos de procesadores 102 adjuntos mediante múltiples zócalos, que pueden acoplarse con dos o más instancias de uno o más procesadores paralelos 112.

Algunos de los componentes particulares que se muestran en el presente documento son opcionales y es posible que no se incluyan en todas las implementaciones del sistema informático 100. Por ejemplo, puede ser compatible con

cualquier número de tarjetas o periféricos suplementarios o se pueden eliminar algunos componentes. Adicionalmente, algunas arquitecturas pueden usar diferente terminología para componentes similares a los ilustrados en la Figura 1. Por ejemplo, el concentrador de memoria 105 puede denominarse Puente norte en algunas arquitecturas, mientras que el concentrador de E/S 107 puede denominarse Puente sur.

La Figura 2A ilustra un procesador paralelo 200, de acuerdo con una realización. Los diversos componentes del procesador paralelo 200 se pueden implementar 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 200 ilustrado es una variante del uno o más 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 permite una comunicación con otros dispositivos, incluyendo otras instancias de la unidad de procesamiento paralelo 202. La unidad de E/S 204 se puede conectar directamente a otros dispositivos. En una realización, la unidad de E/S 204 se conecta con otros dispositivos mediante el uso de una interfaz de concentrador o de conmutador, tal como un concentrador de memoria 105. Las conexiones entre el concentrador de memoria 105 y la unidad de E/S 204 forman un enlace de comunicación 113. Dentro la unidad de procesamiento paralelo 202, la unidad de E/S 204 se conecta con una interfaz de anfitrión 206 y una barra transversal de memoria 216, donde la interfaz de anfitrión 206 recibe comandos dirigidos a realizar operaciones de procesamiento y la barra transversal de memoria 216 recibe comandos dirigidos a realizar operaciones de memoria.

Cuando la interfaz de anfitrión 206 recibe una memoria intermedia de comandos mediante la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para realizar estos comandos a un extremo frontal 208. En una realización, el extremo frontal 208 se acopla con un planificador 210, que está configurado para distribuir comandos u otros elementos de trabajo a una matriz de agrupación de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupaciones de procesamiento 212 está configurada correctamente y en un estado válido antes de que las tareas se distribuyan a las agrupaciones de procesamiento de la matriz de agrupaciones de procesamiento 212. En una realización, el planificador 210 se implementa mediante una lógica de firmware que se ejecuta en un microcontrolador. El planificador implementado por microcontrolador 210 se puede configurar para realizar operaciones de planificación compleja y distribución de trabajo con granularidad gruesa y fina, lo que permite dar prioridad de manera rápida y la conmutación de contexto de hilos que se ejecutan en la matriz de procesamiento 212. En una realización, el software de anfitrión puede probar cargas de trabajo para la planificación en la matriz de procesamiento 212 mediante uno de múltiples timbres de procesamiento de gráficos. Las cargas de trabajo pueden entonces distribuirse automáticamente a lo largo de la matriz de procesamiento 212 mediante la lógica del planificador 210 dentro del microcontrolador planificador.

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

La matriz de agrupaciones de procesamiento 212 se puede configurar para realizar diversos tipos de operaciones de procesamiento paralelo. En una realización, la matriz de agrupaciones de procesamiento 212 está configurada para realizar operaciones informáticas paralelas de propósito general. Por ejemplo, la matriz de agrupaciones de procesamiento 212 puede incluir una lógica para ejecutar tareas de procesamiento que incluye filtración de datos de vídeo y/o de audio, realización de operaciones de modelado, incluyendo operaciones físicas y la realización de transformaciones de datos.

En una realización, la matriz de agrupaciones de procesamiento 212 está configurada para realizar operaciones de procesamiento de gráficos en paralelo. En realizaciones en las que el procesador paralelo 200 está configurado para realizar operaciones de procesamiento de gráficos, la matriz de agrupaciones de procesamiento 212 puede incluir una lógica adicional para soportar la ejecución de tales operaciones de procesamiento de gráficos, incluyendo, pero sin limitación, una lógica de muestreo de textura para realizar operaciones de textura, así como una lógica de teselación y otra lógica de procesamiento de vértices. Además, la matriz de clúster de procesamiento 212 se puede configurar para ejecutar programas de sombreado relacionados con el procesamiento de gráficos tales como, pero sin limitación, sombreadores de vértices, sombreadores de teselación, sombreadores de geometría y sombreadores de píxeles. La unidad de procesamiento paralelo 202 puede transferir datos desde la memoria de sistema por medio de la unidad de E/S 204 para su procesamiento. Durante el procesamiento, los datos transferidos pueden almacenarse en una memoria en chip (p. ej., la memoria de procesador paralelo 222) durante el procesamiento y, a continuación, escribirse en diferido en la memoria del sistema.

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

Durante la operación, la matriz de agrupaciones de procesamiento 212 puede recibir tareas de procesamiento que se van a ejecutar a través del 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 se van a procesar, p. ej., datos de superficie (parche), datos de primitivas, datos de vértices y/o datos de píxeles, así como parámetros de estado y comandos que definen cómo se deben procesar los datos (p. ej., qué programa se va a ejecutar). El planificador 210 puede estar configurado para extraer los índices correspondientes 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 agrupaciones de procesamiento 212 esté configurada en un estado válido antes de que se inicie la carga de trabajo especificada en memorias intermedias de comandos entrantes (p. ej., memorias intermedias de lotes, memorias intermedias de carga, etc.).

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

En diversas realizaciones, las unidades de memoria 224A-224N pueden incluir diversos tipos de dispositivos de memoria, que incluyen memoria de acceso aleatorio dinámica (DRAM) o memoria gráfica de acceso aleatorio, tal como una memoria gráfica de acceso aleatorio síncrona (SGRAM), incluyendo una memoria gráfica de doble velocidad de datos (GDDR). En una realización, las unidades de memoria 224A-224N también pueden incluir memoria 3D apilada, que incluye, pero sin limitación, memoria de ancho de banda alto (HBM). Los expertos en la técnica apreciarán que la implementación específica de las unidades de memoria 224A-224N puede variar y que puede seleccionarse de entre uno de varios diseños convencionales. Se pueden almacenar objetivos de representación gráfica, tales como memorias intermedias de fotogramas o correlaciones de textura, en las unidades de memoria 224A-224N, lo que permite que las unidades de subdivisión 220A-220N escriban partes de cada objetivo de representación gráfica en paralelo para usar de manera eficaz el ancho de banda disponible de la memoria de procesador paralelo 222. En algunas realizaciones, se puede excluir una instancia local de la memoria de procesador paralelo 222 en favor de un diseño de memoria unificado que utiliza memoria de sistema junto con memoria caché local.

En una realización, una cualquiera de las agrupaciones 214A-214N de la matriz de agrupaciones de procesamiento 212 puede procesar datos que se escribirán en cualquiera de las unidades de memoria 224A-224N dentro de la memoria de procesador paralelo 222. La barra transversal de memoria 216 puede estar configurada para transferir la salida de cada agrupación 214A-214N a cualquier unidad de subdivisión 220A-220N o a otra agrupación 214A-214N, que puede realizar operaciones de procesamiento adicionales en la salida. Cada agrupación 214A-214N puede comunicarse con la interfaz de memoria 218 a través de la barra transversal de memoria 216 para leer desde o escribir en diversos dispositivos de memoria externos. En una realización, la barra transversal de memoria 216 tiene una conexión a la interfaz de memoria 218 para comunicarse con la unidad de E/S 204, así como una conexión a una instancia local de la memoria de procesador paralelo 222, lo que permite que las unidades de procesamiento dentro de las diferentes agrupaciones de procesamiento 214A-214N se comuniquen con la memoria de sistema u otra memoria que no sea local a la unidad de procesamiento paralelo 202. En una realización, la barra transversal de memoria 216 puede usar canales virtuales para separar flujos de tráfico entre las agrupaciones 214A-214N y las unidades de subdivisión 220A-220N.

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

La Figura 2B es un diagrama de bloques de una unidad de subdivisión 220, de acuerdo con una realización. En una realización, la unidad de subdivisión 220 es una instancia de una de las unidades de subdivisión 220A-220N de la Figura 2A. Como se ilustra, la unidad de subdivisión 220 incluye una memoria caché L2 221, una interfaz de memoria intermedia de fotogramas 225 y una ROP 226 (unidad de operaciones de rasterización). La memoria caché L2 221 es una caché de lectura/escritura que está configurada para realizar operaciones de carga y almacenamiento recibidas desde la barra transversal de memoria 216 y la ROP 226. La memoria caché L2 221 emite los fallos de lectura y las solicitudes de escritura urgentes a la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. Las actualizaciones también se pueden enviar a la memoria intermedia de fotogramas a través de la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. En una realización, la interfaz de memoria intermedia de fotogramas 225 interactúa con una de las unidades de memoria en la memoria de procesador paralelo, tal como las unidades de memoria 224A-224N de la Figura 2 (p. ej., dentro de la memoria de procesador paralelo 222).

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

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

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

El funcionamiento de la agrupación de procesamiento 214 se puede controlar por medio de un gestor de canalizaciones 232 que distribuye tareas de procesamiento a procesadores paralelos de SIMT. El administrador de canalización 232 recibe instrucciones del planificador 210 de la Figura 2 y gestiona la ejecución de esas instrucciones a través de 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, se pueden incluir diversos tipos de procesadores paralelos de SIMT de arquitecturas diferentes dentro de la agrupación de procesamiento 214. Una o más instancias del multiprocesador de gráficos 234 se pueden incluir dentro de una agrupación de procesamiento 214. El multiprocesador de gráficos 234 puede procesar datos y se puede usar una barra transversal de datos 240 para distribuir los datos procesados a uno de múltiples destinos posibles, incluyendo otras unidades sombreadoras. El

gestor de canalizaciones 232 puede facilitar la distribución de datos procesados especificando destinos para que se distribuyan datos procesados mediante la barra transversal de datos 240.

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

Las instrucciones transmitidas a la agrupación de procesamiento 214 constituyen un hilo. Un conjunto de hilos que se ejecutan a través del conjunto de motores de procesamiento paralelo es un grupo de hilos. Un grupo de hilos ejecuta el mismo programa en diferentes datos de entrada. Cada hilo dentro de un grupo de hilos se puede asignar 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 inactivos durante los ciclos en los que se está procesando ese grupo de hilos. Un grupo de hilos puede incluir también más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234. Cuando el grupo de hilos incluye más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234, se puede realizar el procesamiento a través de ciclos de reloj consecutivos. En una realización, se pueden ejecutar múltiples grupos de hilos simultáneamente en un multiprocesador de gráficos 234.

En una realización, el multiprocesador de gráficos 234 incluye una memoria caché interna para realizar operaciones de carga y almacenamiento. En una realización, el multiprocesador de gráficos 234 puede prescindir de una memoria caché interna y usar una memoria caché (p. ej., memoria caché L1 308) dentro de la agrupación de procesamiento 214. Cada multiprocesador de gráficos 234 también tiene acceso a memorias caché L2 dentro de las unidades de subdivisión (p. ej., unidades de subdivisión 220A-220N de la Figura 2) que se comparten entre todos los grupos de procesamiento 214 y se pueden usar para transferir datos entre hilos. El multiprocesador de gráficos 234 puede acceder también a memoria global fuera de chip, que puede incluir uno o más de memoria de procesador paralelo local y/o memoria de sistema. Cualquier memoria externa a la unidad de procesamiento paralelo 202 se puede usar 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 memoria 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 con una dirección física de una tesela (más información sobre la generación de teselas) y, opcionalmente, un índice de línea de memoria caché. La MMU 245 puede incluir memorias intermedias de conversión adelantada (TLB) de dirección o memorias caché que pueden residir dentro del multiprocesador de gráficos 234 o la memoria caché L1 o la agrupación de procesamiento 214. La dirección física se procesa para distribuir la localidad de acceso a los datos de superficie para permitir un entrelazado eficiente de solicitudes entre unidades de subdivisión. El índice de línea de memoria caché se puede usar para determinar si una solicitud para una línea de memoria caché es un acierto o un fallo.

En aplicaciones de gráficos e informática, se puede configurar una agrupación de procesamiento 214 de tal manera que cada multiprocesador de gráficos 234 esté acoplado a una unidad de textura 236 para realizar operaciones de mapeo de textura, p. ej., determinar posiciones de muestras de textura, leer datos de textura y filtrar los datos de textura. Los datos de textura se leen desde una memoria caché L1 de textura interna (no mostrada) o, en algunas realizaciones, desde la memoria caché L1 dentro del multiprocesador de gráficos 234 y se extraen desde una memoria 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 un procesamiento adicional o para almacenar la tarea procesada en una memoria caché L2, una memoria de procesador paralelo local o una memoria de sistema a través de la barra transversal de memoria 216. Una preROP 242 (unidad de operaciones previas a la rasterización) está configurada para recibir datos desde el multiprocesador de gráficos 234, dirigir datos a las unidades de ROP, que pueden estar ubicadas con unidades de subdivisión como se describe en el presente documento (p. ej., las unidades de subdivisión 220A-220N de la Figura 2). La unidad preROP 242 puede realizar optimizaciones para la mezcla de color, organizar datos de color de píxel y realizar conversiones de dirección.

Se apreciará que la arquitectura de núcleo descrita en el presente documento es ilustrativa y que son posibles variaciones y modificaciones. Cualquier número de unidades de procesamiento, p. ej., multiprocesador de gráficos 234, unidades de textura 236, preROP 242, etc., puede incluirse dentro de una agrupación de procesamiento 214. Además, si bien solo se muestra una agrupación de procesamiento 214, una unidad de procesamiento paralelo como

la descrita en el presente documento puede incluir cualquier número de instancias de la agrupación de procesamiento 214. En una realización, cada agrupación de procesamiento 214 se puede configurar para funcionar independientemente de otros grupos de procesamiento 214 usando unidades de procesamiento separadas y distintas, memorias caché L1, etc.

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

En una realización, la memoria caché de instrucciones 252 recibe un flujo de instrucciones para ejecutarse desde el gestor de canalizaciones 232. Las instrucciones se almacenan en memoria caché en la caché de instrucciones 252 y se envían para su ejecución por la unidad de instrucciones 254. La unidad de instrucciones 254 puede despachar instrucciones como grupos de hilos (p. ej., envoltorios), donde cada hilo del grupo de hilos está 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 correlación de direcciones 256 se puede usar para convertir direcciones del espacio de direcciones unificado en una dirección de memoria distinta a la que se puede acceder mediante las unidades de carga/almacenamiento 266.

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

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

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

La interconexión de memoria y memoria caché 268 es en una red de interconexión que conecta cada una de las unidades funcionales del multiprocesador de gráficos 324 al archivo de registro 258 y a la memoria compartida 270. En una realización, la interconexión de memoria y memoria caché 268 es una interconexión de barra transversal que permite que la unidad de carga/almacenamiento 266 implemente operaciones de carga y almacenamiento entre la memoria compartida 270 y el archivo de registro 258. El archivo de registros 258 puede funcionar a la misma frecuencia que los núcleos de GPGPU 262; por tanto, la transferencia de datos entre los núcleos de GPGPU 262 y el archivo de registros 258 tiene una latencia muy baja. La memoria compartida 270 se puede usar para permitir la comunicación entre hilos que se ejecutan en las unidades funcionales dentro del multiprocesador de gráficos 234. La memoria caché 272 se puede usar como 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 se puede usar como un programa gestionado en memoria caché. Los hilos que se ejecutan en los núcleos de GPGPU 262 pueden almacenar datos mediante programación dentro de la memoria compartida además de los datos almacenados en memoria caché automáticamente que se almacenan dentro de la memoria caché 272.

Las Figuras 3A-3B ilustran multiprocesadores de gráficos adicionales, de acuerdo con las realizaciones. Los multiprocesadores de gráficos 325, 350 ilustrados son variantes del multiprocesador de gráficos 234 de la Figura 2C. Los multiprocesadores de gráficos ilustrados 325, 350 se pueden configurar como un multiprocesador de transmisión (SM) capaz de ejecutar simultáneamente una gran cantidad de hilos de ejecución.

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

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

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

Los expertos en la materia entenderán que la arquitectura descrita en las Figuras 1, 2A-2D y 3A-3B es descriptiva y no limitativa en cuanto al alcance de las presentes realizaciones. Por lo tanto, las técnicas descritas en el presente documento se pueden implementar en cualquier unidad de procesamiento configurada correctamente, incluyendo, sin limitación, uno o más procesadores de aplicaciones móviles, una o más unidades centrales de procesamiento (CPU) de escritorio o de servidor, incluyendo CPU de múltiples núcleos, una o más unidades de procesamiento paralelo, tales como la unidad de procesamiento paralelo 202 de la Figura 2, así como uno o más procesadores gráficos o unidades de procesamiento para propósitos especiales, sin apartarse del alcance de las realizaciones descritas en el presente documento.

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

Técnicas para la interconexión de GPU a un procesador anfitrión

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

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

En una realización, cada procesador de múltiples núcleos 405-406 está acoplado de manera comunicativa a una memoria de procesador 401-402, a través de las interconexiones de memoria 430-431, respectivamente, y cada GPU 410-413 está acoplada de manera comunicativa a la memoria de GPU 420-423 a través de las interconexiones de memoria de GPU 450-453, respectivamente. Las interconexiones de memoria 430-431 y 450-453 pueden usar las mismas tecnologías de acceso de memoria u otras diferentes. A modo de ejemplo y no como limitación, las memorias de procesador 401-402 y las memorias de GPU 420-423 pueden ser memorias volátiles, tal como memorias de acceso aleatorio dinámicas (DRAM) (incluyendo DRAM apiladas), SDRAM DDR de gráficos (GDDR) (p. ej., GDDR5, GDDR6) o memoria de ancho de banda alto (HBM) y/o pueden ser memorias no volátiles, tales como 3D XPoint o Nano-Ram. En una realización, una parte de las memorias puede ser memoria volátil y otra parte puede ser memoria no volátil (p. ej., usando una jerarquía de memoria de dos niveles (2LM)).

Como se describe a continuación, aunque los diversos procesadores 405-406 y las GPU 410-413 pueden estar acoplados físicamente a una memoria particular 401-402, 420-423, respectivamente, se puede implementar una arquitectura de memoria unificada en la que el mismo espacio de direcciones de sistema virtuales (también denominado espacio de "direcciones efectivas") se distribuye entre todas las diversas memorias físicas. Por ejemplo, cada una de las memorias de procesador 401-402 puede comprender 64 GB del espacio de direcciones de memoria del sistema y cada una de las memorias de GPU 420-423 puede comprender 32 GB del espacio de direcciones de memoria del 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 407 ilustrado incluye una pluralidad de núcleos 460A-460D, cada uno con una memoria intermedia de conversión anticipada de direcciones 461A-461D y una o más memorias caché 462A-462D. Los núcleos pueden incluir diversos otros componentes para ejecutar instrucciones y procesar datos, que no se han ilustrado para evitar oscurecer los principios subyacentes de la invención, (p. ej., unidades de extracción de instrucciones, unidades de predicción de ramificaciones, decodificadores, unidades de ejecución, memorias intermedias de reordenación, etc.). Las memorias caché 462A-462D pueden comprender memorias caché de nivel 1 (L1) y de nivel 2 (L2). Además, una o más memorias caché compartidas 426 pueden incluirse en la jerarquía de almacenamiento en memoria caché y ser compartidas por conjuntos de núcleos 460A-460D. Por ejemplo, una realización del procesador 407 incluye 24 núcleos, cada uno con su propia memoria caché L1, doce memorias caché L2 compartidas y doce memorias caché L3 compartidas. En esta realización, una de las memorias caché L2 y L3 están compartidas por dos núcleos adyacentes. El procesador 407 y el módulo de integración de acelerador de gráficos 446 se conectan con la memoria de sistema 441, que puede incluir memorias de procesador 401-402.

Se mantiene la coherencia para los datos e instrucciones almacenados en las diversas memorias caché 462A-462D, 456 y la memoria de sistema 441 por medio de una comunicación entre núcleos a través de un bus de coherencia 464. Por ejemplo, cada memoria caché puede tener una lógica/circuitaría de coherencia de memoria caché asociada a la misma para comunicarse a través del bus de coherencia 464 en respuesta a lecturas o escrituras detectadas en líneas de memoria caché particulares. En una implementación, se implementa un protocolo de monitorización de memoria caché a través del bus de coherencia 464 para monitorizar los accesos a memoria caché. Los expertos en la materia entienden bien las técnicas de coherencia/monitorización de caché y no se describirán en el presente documento en detalle para evitar oscurecer los principios subyacentes de la invención.

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

En una implementación, un circuito de integración de acelerador 436 proporciona servicios de gestión de memoria caché, de acceso a memoria, de gestión de contexto y de gestión de interrupciones en beneficio de una pluralidad de motores de procesamiento de gráficos 431, 432, N del módulo de aceleración de gráficos 446. Cada motor de procesamiento de gráficos 431, 432, N puede comprender una unidad de procesamiento de gráficos (GPU) distinta.

Como alternativa, los motores de procesamiento de gráficos 431, 432, N pueden comprender diferentes tipos de motores de procesamiento de gráficos dentro de una GPU, tales como unidades de ejecución de gráficos, motores de procesamiento de medios (p. ej., codificadores/descodificadores de vídeo), muestreadores y motores de transferencia de imágenes en bloques (*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 de acelerador 436 incluye una unidad de gestión de memoria (MMU) 439 para realizar diversas funciones de gestión de memoria, tales como conversiones de memoria virtual a física (también denominadas conversiones de memoria efectiva a real) y protocolos de acceso a memoria para acceder a memoria de sistema 441. La MMU 439 también puede incluir una memoria intermedia de conversión anticipada de direcciones (TLB) (no mostrada) para almacenar en memoria caché las conversiones de dirección virtual/efectiva a física/real. En una implementación, una memoria caché 438 almacena comandos y datos para un acceso eficaz por los motores de procesamiento de gráficos 431-432, N. En una realización, los datos almacenados en la memoria caché 438 y en las memorias de gráficos 433-434, N se mantienen coherentes con las memorias caché de núcleo 462A-462D, 456 y la memoria de sistema 411. Como se ha mencionado, esto se puede conseguir mediante el circuito proxy 425 que toma parte en el mecanismo de coherencia de caché en beneficio de la caché 438 y las memorias 433-434, N (p. ej., enviando actualizaciones a la caché 438 relacionadas con modificaciones/accesos de líneas de caché en las memorias caché de procesador 462A-462D, 456 y recibiendo actualizaciones desde la caché 438).

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

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

Por tanto, el circuito de integración de acelerador actúa como un puente al sistema para el módulo de aceleración de gráficos 446 y proporciona servicios de conversión de direcciones y de memoria caché de sistema. Además, el circuito de integración de acelerador 436 puede proporcionar funciones de virtualización para que el procesador 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 con el espacio de direcciones reales observado por el procesador anfitrión 407, cualquier procesador anfitrión puede dirigir estos recursos directamente usando un valor de dirección efectiva. Una función del circuito de integración de acelerador 436, en una realización, es la separación física de los motores de procesamiento de gráficos 431-432, N para que aparezcan ante el sistema como unidades independientes.

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

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

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

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

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

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

Para el modelo de programación compartida, el módulo de aceleración de gráficos 446 o un motor de procesamiento de gráficos 431-432, N individual selecciona un elemento de proceso usando un gestor de procesos. En una realización, los elementos de proceso se almacenan en la memoria de sistema 411 y son direccionables usando las técnicas de conversión de direcciones efectivas a direcciones reales descritas en el presente documento. El gestor de procesos puede ser un valor específico de la implementación proporcionado al proceso de anfitrión al registrar su contexto con el motor de procesamiento de gráficos 431-432, N (es decir, solicitando que el software de sistema añada el elemento de proceso a la lista enlazada de elementos de proceso). Los 16 bits inferiores del gestor de procesos pueden ser el desplazamiento del elemento de proceso dentro de la lista enlazada de elementos de proceso.

La Figura 4D ilustra un corte de integración de acelerador 490 ilustrativo. Tal y como se usa en el presente documento, un "corte" comprende una parte especificada de los recursos de procesamiento del circuito de integración de acelerador 436. El espacio de direcciones efectivas de aplicación 482 dentro de la memoria de sistema 411 almacena los elementos de proceso 483. En una realización, los elementos de proceso 483 se almacenan en respuesta a invocaciones de GPU 481 desde las aplicaciones 480 ejecutadas en el procesador 407. Un elemento de proceso 483 contiene el estado de proceso para la aplicación correspondiente 480. Un descriptor de trabajo (WD) 484 contenido en el elemento de proceso 483 puede ser un único trabajo solicitado por una aplicación o puede contener un puntero 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 ser compartidos por todos o un subconjunto de los procesos del 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 dedicado es específico para la implementación. En este modelo, un único proceso acapara el módulo de aceleración de gráficos 446 o un motor de procesamiento de gráficos 431 individual. Debido a que un único proceso acapara el módulo de aceleración de gráficos 446, el hipervisor inicializa el circuito de integración de acelerador 436 para la subdivisión acaparadora y el sistema operativo inicializa el circuito de integración de acelerador 436 para el proceso acaparador en el momento en que se asigna el módulo de aceleración de gráficos 446.

En funcionamiento, una unidad de obtención de WD 491 en la sección de integración de acelerador 490 extrae el siguiente WD 484, que incluye una indicación del trabajo que va a hacer uno de los motores de procesamiento de gráficos del módulo de aceleración de gráficos 446. Los datos del WD 484 se pueden almacenar en registros 445 y ser usados por la MMU 439, el circuito de gestión de interrupciones 447 y/o el circuito de gestión de contexto 446, como se ilustra. Por ejemplo, una realización de la MMU 439 incluye circuitos de recorrido de páginas/segmentos para acceder a tablas de segmentos/páginas 486 dentro del espacio de direcciones virtuales 485 del sistema operativo. El circuito de gestión de interrupciones 447 puede procesar eventos de interrupción 492 recibidos desde el módulo de aceleración de gráficos 446. Cuando se realizan operaciones gráficas, una dirección efectiva 493 generada por un motor de procesamiento de gráficos 431-432, N se convierte en una dirección real por medio de la MMU 439.

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

5

Tabla 1 - Registros inicializados por el hipervisor

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

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

10

Tabla 2 - Registros inicializados por el sistema operativo

1	Identificación de procesos e hilos
2	Puntero de guardado/restauración de contexto de dirección efectiva (EA)
3	Puntero de registro de utilización de acelerador de dirección virtual (VA)
4	Puntero de tabla de segmentos de almacenamiento de dirección virtual (VA)
5	Máscara de autoridad
6	Descriptor de trabajo

15

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

20

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

25

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

30

En este modelo, el hipervisor de sistema 496 acapara el módulo de aceleración de gráficos 446 y pone su función a disposición de todos los sistemas operativos 495. Para que un módulo de aceleración de gráficos 446 soporte la virtualización por el hipervisor de sistema 496, el módulo de aceleración de gráficos 446 puede adherirse a los siguientes requisitos: 1) La solicitud de trabajo de una aplicación debe ser autónoma (es decir, no es necesario mantener el estado entre trabajos) o el módulo de aceleración de gráficos 446 debe proporcionar un mecanismo de almacenamiento y restauración de contexto. 2) El módulo de aceleración de gráficos 446 garantiza que la solicitud de trabajo de una aplicación se completa en una cantidad especificada de tiempo, incluyendo cualquier fallo de conversión, o el módulo de aceleración de gráficos 446 proporciona la capacidad de dar prioridad al procesamiento del trabajo. 3) Se ha de garantizar al módulo de aceleración de gráficos 446 la equidad entre procesos cuando se opera en el modelo de programación compartido dirigido.

40

En una realización, para el modelo compartido, se requiere que la aplicación 480 realice una llamada al sistema operativo 495 con un tipo de módulo de aceleración de gráficos 446, un descriptor de trabajo (WD), un valor de registro de máscara de autoridad (AMR) y un puntero a área de guardado/restauración de contexto (CSRP). El tipo del módulo

de aceleración de gráficos 446 describe la función de aceleración objetivo para la llamada al sistema. El tipo del módulo de aceleración de gráficos 446 puede ser un valor específico del sistema. El WD tiene un formato específico para el módulo de aceleración de gráficos 446 y puede estar en forma de comando de módulo de aceleración de gráficos 446, puntero de dirección efectiva a una estructura definida por el usuario, puntero de dirección efectiva a una cola de comandos o cualquier otra estructura de datos para describir el trabajo que debe realizar el módulo de aceleración de gráficos 446. En una realización, el valor de AMR es el estado de AMR que se debe usar para el proceso actual. El valor pasado al sistema operativo es similar a que una aplicación establezca el AMR. Si el circuito de integración de acelerador 436 y las implementaciones del módulo de aceleración de gráficos 446 no admiten 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 al hipervisor. El hipervisor 496 puede aplicar opcionalmente el valor actual del registro de anulación de máscara de autoridad (AMOR) antes de colocar el AMR en el elemento de proceso 483. En una realización, el CSRP es uno de los registros 445 que contienen la dirección efectiva de un área en el espacio de direcciones 482 de la aplicación para que el módulo de aceleración de gráficos 446 guarde y restaure el estado de contexto. Este puntero es opcional si no se requiere que se guarde ningún estado entre trabajos o cuando se anula un trabajo. El área de guardado/restauración de contexto puede estar fijada en la memoria de sistema.

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

Tabla 3 - Parámetros de llamada del SO al hipervisor

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

Al recibir la llamada al hipervisor, el hipervisor 496 verifica que el sistema operativo 495 se ha registrado y se le ha otorgado la autoridad para usar el módulo de aceleración de gráficos 446. A continuación, el hipervisor 496, pone el elemento de proceso 483 en la lista enlazada de elementos de proceso 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 de registro de máscara de autoridad (AMR) (potencialmente enmascarado).
3	Un puntero a área de guardado/restauración de contexto (CSRP) de dirección efectiva (EA)
4	Un ID de proceso (PID) y un ID de hilo (TID) opcional
5	Un puntero de registro de utilización de acelerador (AURP) de dirección virtual (VA)
6	La dirección virtual del puntero a tabla de segmentos de almacenamiento (SSTP)
7	Un número de servicio de interrupción lógica (LISN)
8	Tabla de vectores de interrupción, derivada de los parámetros de llamada de hipervisor.
9	Un valor de registro de estado (SR)
10	Un ID de subdivisión lógica (LPID)
11	Un puntero de registro de utilización de acelerador de hipervisor de dirección real (RA)
12	El registro de descriptor de almacenamiento (SDR)

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

Como se ha ilustrado en la Figura 4F, una realización de la invención emplea una memoria unificada direccionable mediante un espacio de direcciones de memoria virtual común usado para acceder a las memorias de procesador

físico 401-402 y a las memorias de GPU 420-423. En esta implementación, las operaciones ejecutadas en las GPU 410-413 utilizan el mismo espacio de direcciones de memoria virtual/efectiva para acceder a las memorias de procesadores 401-402, y viceversa, simplificando de esta manera la programabilidad. En una realización, una primera parte del espacio de direcciones virtuales/efectivas se asigna a la memoria de procesador 401, una segunda parte a la segunda memoria de procesador 402, una tercera parte a la memoria de GPU 420, y así sucesivamente. Todo el espacio de memoria virtual/efectiva (a veces denominado espacio de direcciones efectivas) se distribuye así entre 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, el circuito de gestión de desvío/coherencia 494A-494E dentro de una o más de las MMU 439A-439E garantiza la coherencia de memoria caché entre las memorias caché de los procesadores anfitriones (p. ej., 405) y las GPU 410-413 e implementa técnicas de desvío que indican las memorias físicas en las que deben almacenarse determinados tipos de datos. Aunque se ilustran múltiples instancias de la circuitería de gestión de desvío/coherencia 494A-494E en la Figura 4F, la circuitería de desvío/coherencia puede implementarse dentro de la MMU de uno o más procesadores de anfitrión 405 y/o dentro del circuito de integración de acelerador 436.

Una realización permite que la memoria adjunta a la GPU 420-423 se mapee como parte de la memoria de sistema, y que se acceda a ella usando tecnología de memoria virtual compartida (SVM), pero sin sufrir las desventajas de rendimiento típicas asociadas con la coherencia de caché de sistema completa. La capacidad de que se acceda a la memoria conectada a GPU 420-423 como memoria de sistema sin sobrecarga de coherencia de memoria caché onerosa proporciona un entorno operativo beneficioso para la descarga de la GPU. Esta disposición permite que el software del procesador anfitrión 405 establezca operandos y acceda a resultados de cálculo, sin la sobrecarga de copias de datos de DMA de E/S tradicionales. Tales copias tradicionales implican llamadas al controlador, interrupciones y accesos de E/S mapeada en memoria (MMIO), todos ellos ineficaces en relación con los simples accesos a memoria. Al mismo tiempo, la capacidad de acceder a memoria conectada a GPU 420-423 sin sobrecargas de coherencia de memoria caché puede ser crucial para el tiempo de ejecución de un cálculo descargado. En caso de tráfico de memoria sustancial de escritura de transmisión continua, por ejemplo, la sobrecarga de coherencia de memoria caché puede reducir significativamente el ancho de banda de escritura efectivo visto por una GPU 410-413. La eficacia de la configuración de los operandos, la eficacia del acceso a los resultados y la eficiencia del cálculo de la GPU juegan un papel en la determinación de la efectividad de la descarga de la GPU.

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

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

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

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

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

Canalización de procesamiento de gráficos

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

En una realización, el ensamblador de datos 502 es una unidad de procesamiento que recopila datos de vértices para superficies y primitivas. El ensamblador de datos 502 emite entonces los datos de vértice, incluyendo los atributos de vértice, a la unidad de procesamiento de vértices 504. La unidad de procesamiento de vértices 504 es una unidad de ejecución programable que ejecuta programas de sombreado de vértices, iluminando y transformando datos de vértices según lo especificado por los programas de sombreado de vértices. La unidad de procesamiento de vértices 504 lee datos que están almacenados en memoria caché, local o de sistema para su uso en el procesamiento de los datos de vértices y puede programarse para transformar los datos de vértices desde una representación de coordenadas basada en objetos hasta un espacio de coordenadas de espacio global o un espacio de coordenadas de dispositivo normalizado.

Una primera instancia de un ensamblador de primitivas 506 recibe atributos de vértices desde la unidad de procesamiento de vértices 504. El ensamblador de primitivas 506 lee atributos de vértices almacenados según sea necesario y construye primitivas de gráficos para su procesamiento por la unidad de procesamiento de control de teselación 508. Las primitivas de gráficos incluyen triángulos, segmentos de línea, puntos, parches y así sucesivamente según lo admitido por varias interfaces de programación de aplicaciones (API) de procesamiento de gráficos.

La unidad de procesamiento de control de teselación 508 trata los vértices de entrada como puntos de control para un parche geométrico. Los puntos de control se transforman desde una representación de entrada del parche (p. ej., las bases del parche) a una representación que es adecuada para su uso en la evaluación de superficies por parte de 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 detalle dependiente de la vista asociado con el borde. Una unidad de teselación 510 está configurada para recibir los factores de teselación para los 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 de superficie y atributos de vértice para cada vértice asociado con las primitivas geométricas.

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

En algunas realizaciones, la unidad de procesamiento de geometría 516 puede añadir o borrar elementos en el flujo de geometría. La unidad de procesamiento de geometría 516 envía los parámetros y vértices que especifican nuevas primitivas gráficas al ensamblador de primitivas 518. El ensamblador de primitivas 518 recibe los parámetros y vértices de la unidad de procesamiento de geometría 516 y construye primitivas de gráficos para su procesamiento mediante

una unidad de escala, selección y recorte de ventana gráfica 520. La unidad de procesamiento de geometría 516 lee datos que están almacenados en la memoria de procesador paralelo o en la memoria de sistema para su uso en el procesamiento de los datos de geometría. La unidad de escala, selección y recorte de ventana gráfica 520 realiza un recorte, una selección y un escalado de la ventana gráfica y genera primitivas gráficas procesadas a un rasterizador 522.

El rasterizador 522 puede realizar optimizaciones de selección de profundidad y otras basadas en profundidad. El rasterizador 522 también realiza una conversión de escaneo en las nuevas primitivas de gráficos para generar fragmentos y enviar esos fragmentos y los datos de cobertura asociados a la unidad de procesamiento de fragmentos/píxeles 524. La unidad de procesamiento de fragmentos/píxeles 524 es una unidad de ejecución programable que está configurada para ejecutar programas de sombreado de fragmentos o programas de sombreado de píxeles. La unidad de procesamiento de fragmentos/píxeles 524 transforma fragmentos o píxeles recibidos del rasterizador 522, según lo especificado por los programas de sombreado de fragmentos o píxeles. Por ejemplo, la unidad de procesamiento de fragmentos/píxeles 524 puede programarse para realizar operaciones incluidas, entre otras, mapeo de textura, sombreado, combinación, corrección de textura y corrección de perspectiva para producir fragmentos o píxeles sombreados que se envían a una unidad de operaciones de trama 526. La unidad de procesamiento de fragmentos/píxeles 524 puede leer datos almacenados en la memoria de procesador paralelo o bien en la memoria de sistema para su uso cuando se procesan los datos de fragmentos. Los programas de sombreado de fragmentos o de píxeles pueden estar configurados para sombrear con granularidades muestras, píxeles, teselas u otros 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 rasterización que incluyen, pero sin limitación, estarcido, prueba z, mezcla y similares, y emite datos de píxeles como datos de gráficos procesados para que se almacenen en la memoria de gráficos (p. ej., la memoria de procesador paralelo 222 como en la Figura 2 y/o la memoria de sistema 104 como en la Figura 1, para que se visualicen en el uno o más dispositivos de visualización 110 o para su procesamiento adicional por uno del uno o más procesadores 102 o procesadores paralelos 112. En algunas realizaciones, la unidad de operaciones de rasterización 526 está configurada para comprimir datos z o de color que se escriben en memoria y descomprimir datos z o de color que se leen desde la memoria.

Ejemplos de ajuste de la representación gráfica en función de la expresión facial

Volviendo ahora a la Figura 6A, una realización de un sistema de procesamiento electrónico 600 puede incluir un procesador de aplicaciones 601, medios de almacenamiento persistentes 602 acoplados comunicativamente al procesador de aplicaciones 601, un subsistema de gráficos 603 acoplado comunicativamente al procesador de aplicaciones 601, y un ajustador de gráficos de expresiones faciales 604 acoplado comunicativamente al subsistema de gráficos 603 para ajustar un parámetro del subsistema de gráficos 603 en función de una expresión facial detectada de un usuario. El sistema 600 además puede incluir uno o más de un motor sensor 605, un motor de enfoque 606 y/o un motor de movimiento 607, cada uno de los cuales puede estar acoplado comunicativamente al subsistema de gráficos 603.

En algunas realizaciones, el motor sensor 605 puede incluir un detector de luz 608 para detectar una cantidad de luz ambiental. De acuerdo con la invención, se proporciona un ajuste de la visualización en función de las condiciones de iluminación ambiental. El subsistema de gráficos se puede configurar para cambiar un parámetro de pantalla en función de la cantidad de luz ambiental detectada. Por ejemplo, el brillo de visualización se puede ajustar en función de la salida de uno o más sensores de luz que detectan las condiciones de luz ambiental. En algunos entornos, la calidad de la visualización se puede ajustar dependiendo de las condiciones de luz ambiental. Por ejemplo, en un entorno oscuro, podría ser más fácil ver una visualización más tenue, mientras que en un entorno brillante el usuario puede preferir un mayor brillo de visualización (consume más energía). El sistema 600 puede usar el detector de luz 608 para regular el brillo, el contraste, etc., de visualización en función de la luz detectada en el entorno. La información del detector de luz 608 se puede combinar con otras realizaciones descritas en el presente documento (p. ej., detección de expresiones faciales) para una experiencia de usuario mejorada.

En algunas realizaciones, el ajustador de gráficos de expresiones faciales 604 puede incluir un detector de expresiones faciales para detectar la expresión facial del usuario y un ajustador de parámetros acoplado comunicativamente al detector de expresiones faciales para ajustar el parámetro del subsistema de gráficos en función de la expresión facial detectada del usuario (p. ej., como se describe con más detalle a más adelante). Por ejemplo, la expresión facial detectada puede incluir uno o más de un entrecerrado los ojos, parpadeo, guiño de un ojo y tensión de los músculos faciales del usuario. El parámetro del subsistema de gráficos 603 puede incluir uno o más de un zum de escena, una posición de escena y un parámetro de pantalla.

Las realizaciones de cada uno del procesador de aplicaciones 601, medios de almacenamiento persistentes 602, subsistema de gráficos 603, ajustador de gráficos de expresiones faciales 604, motor sensor 605, motor de enfoque 606, motor de movimiento 607, detector de luz 608 anteriores y de otros componentes del sistema se pueden implementar en hardware, software o cualquier combinación adecuada de los mismos. Por ejemplo, las implementaciones de hardware pueden incluir lógica configurable tal como, por ejemplo, matrices lógicas

programables (PLA), FPGA, dispositivos lógicos programables complejos (CPLD), o en hardware lógico de funcionalidad fija usando tecnología de circuitos tal como, por ejemplo, ASIC, semiconductor de óxido metálico complementario (CMOS) o tecnología lógica de transistor-transistor (TTL), o cualquier combinación de las mismas. Como alternativa, o adicionalmente, estos componentes se pueden implementar en uno o más módulos como un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento legible por máquina u ordenador, tal como memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), ROM programable (PROM), firmware, memoria flash, etc., para ser ejecutadas por un procesador o dispositivo informático. Por ejemplo, el código de programa informático para llevar a cabo las operaciones de los componentes se puede escribir en cualquier combinación de uno o más lenguajes de programación apropiados/aplicables al sistema operativo, incluyendo un lenguaje de programación orientado a objetos tal como PYTHON, PERL, JAVA, SMALLTALK, C++, C# o similares y lenguajes de programación de procedurales convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares.

Por ejemplo, el sistema 600 puede incluir componentes y/o características similares a las del sistema 100, configurado además para ajustar un parámetro gráfico en función de una expresión facial. Por ejemplo, el subsistema de gráficos 603 puede incluir componentes y/o características similares a las del procesador paralelo 200, configurado además con un ajustador de gráficos de expresiones faciales como se describe en el presente documento. El sistema 600 también se puede adaptar para trabajar con un sistema estéreo montado en la cabeza tal como, por ejemplo, el sistema descrito en relación con las Figuras 11-15, más adelante.

Ejemplos de motor sensor

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

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

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

Ejemplos de motor de enfoque

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

A un alto nivel, algunas realizaciones pueden usar información de enfoque para ajustar los parámetros de rendimiento del sistema gráfico en función de 1) dónde se supone que está mirando el usuario, 2) dónde se determina que está mirando el usuario, 3) dónde una aplicación desea que mire el usuario, y/o 4) dónde se predice que el usuario mirará

en el futuro. Algunas señales de enfoque pueden ser más intensas en la región focal adonde mira el usuario. Si el usuario está mirando al frente, es posible que vea las cosas con un enfoque nítido. Con escenas u objetos hacia la periferia, el usuario puede notar movimiento, pero no detalles con un enfoque nítido.

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

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

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

Ejemplos de motor de movimiento

Volviendo ahora a la Figura 6D, un motor de movimiento 616 puede obtener información del motor sensor y/o del motor de enfoque y de otras fuentes para proporcionar información de movimiento. La información de movimiento puede incluir, por ejemplo, la posición de la cabeza, la velocidad de la cabeza, la aceleración de la cabeza, la dirección del movimiento de la cabeza, la velocidad del ojo, la aceleración del ojo, la dirección de movimiento del ojo, la posición del objeto, la velocidad del objeto, la aceleración del objeto, la dirección de movimiento del objeto, etc. La información de movimiento también puede incluir información de movimiento anterior, información de movimiento futuro determinado y/o información de movimiento previsto (p. ej., una velocidad de la cabeza prevista, una aceleración de la cabeza prevista, una posición de la cabeza prevista, una dirección de movimiento de la cabeza prevista, una velocidad del ojo prevista, una aceleración del ojo prevista, una dirección de movimiento del ojo prevista, posición de contenido futuro determinada, velocidad de objeto de contenido futuro determinada, aceleración de objeto de contenido futuro determinada, posición de objeto prevista, velocidad de objeto prevista, aceleración de objeto prevista, etc.).

A un nivel alto, algunas realizaciones pueden usar información de movimiento para ajustar los parámetros de rendimiento del sistema gráfico basado en 1) el usuario moviendo su cabeza, 2) el usuario moviendo sus ojos, 3) el usuario moviendo su cuerpo, 4) adonde desea una aplicación que el usuario gire la cabeza, los ojos y/o el cuerpo, y/o 4) adonde se predice que el usuario girará la cabeza, los ojos y/o el cuerpo en el futuro. Se puede determinar fácilmente parte de la información de movimiento a partir de la información detectada. Por ejemplo, la posición de la cabeza, la velocidad, la aceleración, la dirección del movimiento, etc., pueden determinarse a partir de un acelerómetro. La información de movimiento de los ojos se puede determinarse rastreando la información de la posición de los ojos a lo largo del tiempo (p. ej., si el seguidor de ojos únicamente proporciona información de la posición de los ojos).

Parte de la información de movimiento puede estar basada en el contenido. En un juego o contenido 3D sobre la marcha, por ejemplo, la aplicación puede saber cómo de rápido y hacia dónde se mueven los objetos. La aplicación

le puede proporcionar la información al motor de movimiento (p. ej., a través de una llamada de API). Para la toma de decisiones, también se puede alimentar al motor de movimiento, con información de un futuro movimiento del objeto basada en el contenido para un fotograma/escena siguiente. Algo de la información de movimiento basada en el contenido se puede determinar mediante el procesamiento de imágenes o el procesamiento de visión artificial del contenido.

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

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

Ejemplos de superposición de motores

Los expertos en la materia apreciarán que los aspectos de diversos motores descritos en el presente documento se pueden superponer con otros motores y que partes de cada motor se pueden implementar o distribuir a través de diversas partes de un sistema de procesamiento electrónico. Por ejemplo, el motor de enfoque puede usar información de movimiento para proporcionar una futura área de enfoque prevista y el motor de movimiento puede usar información de enfoque para predecir un futuro movimiento. La información de movimiento del ojo puede provenir directamente del motor sensor, se puede determinar/predecir mediante el motor de enfoque y/o se puede determinar/predecir mediante el motor de movimiento. Los ejemplos del presente documento deben considerarse ilustrativos y no limitativos en términos de implementaciones específicas.

Volviendo ahora a la Figura 7, una realización de un aparato gráfico 700 de acuerdo con la invención incluye un detector de expresiones faciales 721 para detectar una expresión facial de un usuario, y un ajustador de parámetros 722 acoplado comunicativamente al detector de expresiones faciales 721 para ajustar un parámetro gráfico en función de la expresión facial detectada del usuario. Por ejemplo, la expresión facial detectada puede incluir uno o más de un entrecerrado los ojos, parpadeo, guiño de un ojo y tensión de los músculos faciales del usuario. El parámetro gráfico puede incluir uno o más de una resolución de fotograma, un contraste de pantalla, un brillo de pantalla y una tasa de sombreado. Según la invención, el ajustador de parámetros 722 está configurado para ajustar el parámetro gráfico en un área de enfoque en función de la expresión facial detectada del usuario para mejorar la percepción del usuario en el área de enfoque. En algunas realizaciones, el detector de expresiones faciales 721 puede incluir una cámara que captura continuamente imágenes del rostro del usuario. Las imágenes pueden analizarse automáticamente (p. ej., utilizando un sistema de aprendizaje automático, ya sea localmente o en la nube) para determinar la expresión facial del usuario.

En algunas realizaciones del aparato 700, el detector de expresiones faciales 721 se puede configurar para detectar una cantidad de la expresión facial del usuario. Por ejemplo, una cantidad de ajuste del parámetro gráfico puede basarse en la cantidad de expresión facial detectada del usuario. En algunas realizaciones, el parámetro gráfico puede incluir un zum de una escena gráfica. Por ejemplo, el detector de expresiones faciales 721 se puede configurar para detectar una cantidad de un entrecerrado de ojos del usuario en función de una distancia detectada entre los párpados del usuario, y el ajustador de parámetros 722 se puede configurar para ajustar una cantidad del zum de la escena

gráfica en función de la cantidad de entrecerrado detectado del usuario. El ajustador de parámetros 722 también se puede configurar para ajustar el parámetro gráfico en función de al menos otra entrada detectada. Por ejemplo, al menos otra entrada detectada puede incluir una o más de una frecuencia de pulso, una frecuencia cardíaca y un gesto del usuario.

En algunas realizaciones del aparato 700, el detector de expresiones faciales 721 se puede configurar adicionalmente o, como alternativa, para detectar un número de parpadeos del usuario. Por ejemplo, el ajustador de parámetros 722 se puede configurar para ajustar el parámetro gráfico en función del número detectado de parpadeos del usuario para mejorar la percepción del usuario. En algunas realizaciones, el detector de expresiones faciales 721 también se puede configurar para detectar un impedimento visual del usuario. Por ejemplo, el impedimento visual detectado puede incluir uno o más de la presencia de gafas y la presencia de un parche en el ojo. En algunas realizaciones, el aparato 700 además puede incluir un detector de luz 723 acoplado comunicativamente al ajustador de parámetros 722 para detectar una cantidad de luz ambiental. El ajustador de parámetros 722 se puede configurar para ajustar el parámetro gráfico en función de la cantidad de luz ambiental detectada.

En algunas realizaciones del aparato 700, el detector de expresiones faciales 721 se puede configurar adicionalmente o, como alternativa, para ajustar de manera predictiva el parámetro gráfico en función de una comparación de la expresión facial detectada y las expresiones faciales detectadas previamente. Por ejemplo, si el usuario parpadea cada 3 segundos y el sistema detecta que el usuario empieza a parpadear tres segundos después del último, entonces el sistema puede predecir que es un parpadeo normal y puede suspender la representación mientras se produce el parpadeo completo. Ventajosamente, el ajuste predictivo puede reducir cierta latencia asociada con el sistema de detección facial.

Las realizaciones de cada uno de los detectores de expresión facial 721, ajustador de parámetros 722, detector de luz 723 y otros componentes del aparato 700 anteriores se pueden implementar en hardware, software o cualquier combinación de los mismos. Por ejemplo, partes o la totalidad del aparato 700 se pueden implementar como parte del procesador paralelo 200, configurado además con un detector de expresiones faciales y un ajustador de parámetros, como se describe en el presente documento. El aparato 700 también se puede adaptar para trabajar con un sistema estéreo montado en la cabeza tal como, por ejemplo, el sistema descrito en relación con las Figuras 11-15, más adelante. Por ejemplo, las implementaciones de hardware pueden incluir lógica configurable tal como, por ejemplo, PLA, FPGA, CPLD, o en hardware de lógica de funcionalidad fija usando tecnología de circuitos tal como, por ejemplo, tecnología ASIC, CMOS o TTL, o cualquier combinación de las mismas. Como alternativa, o adicionalmente, estos componentes se pueden implementar en uno o más módulos como un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento legible por máquina u ordenador, tal como RAM, ROM, PROM, firmware, memoria flash, etc., para ser ejecutadas por un procesador o dispositivo informático. Por ejemplo, el código de programa informático para llevar a cabo las operaciones de los componentes se puede escribir en cualquier combinación de uno o más lenguajes de programación apropiados/aplicables al sistema operativo, incluyendo un lenguaje de programación orientado a objetos tal como PYTHON, PERL, JAVA, SMALLTALK, C++, C# o similares y lenguajes de programación de procedurales convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares.

Volviendo ahora a las Figuras 8A a 8F, una realización de un método 800 para ajustar gráficos puede incluir, en el bloque 831, detectar una expresión facial de un usuario y, en el bloque 832, ajustar un parámetro gráfico en función de la expresión facial detectada del usuario. Por ejemplo, en el bloque 833, la expresión facial detectada puede incluir uno o más de un entrecerrado de ojos, parpadeo, guiño de un ojo y tensión de los músculos faciales del usuario. En el bloque 834, el parámetro gráfico puede incluir uno o más de una resolución de fotograma, un contraste de pantalla, un brillo de pantalla y una tasa de sombreado. Según la invención, en el bloque 835, el método 800 incluye ajustar el parámetro gráfico en un área de enfoque en función de la expresión facial detectada del usuario para mejorar la percepción del usuario en el área de enfoque.

En algunas realizaciones, el método 800 además puede incluir, en el bloque 836, detectar una cantidad de la expresión facial del usuario. Por ejemplo, en el bloque 837, una cantidad de ajuste del parámetro gráfico se basa en la cantidad de expresión facial detectada del usuario. En el bloque 838, el parámetro gráfico puede incluir un zum de una escena gráfica. Por ejemplo, el método 800 puede incluir, en el bloque 839, detectar una cantidad de entrecerrado los ojos del usuario en función de una distancia detectada entre los párpados del usuario, y ajustar, en el bloque 840, una cantidad del zum de la escena gráfica en función de la cantidad del entrecerrado de ojos detectada del usuario. Algunas realizaciones además pueden incluir, en el bloque 841, ajustar el parámetro gráfico en función de al menos otra entrada detectada. Por ejemplo, en el bloque 842, al menos otra entrada detectada puede incluir una o más de una frecuencia de pulso, una frecuencia cardíaca y un gesto del usuario.

En algunas realizaciones, el método 800 además puede incluir, en el bloque 843, detectar un número de parpadeos del usuario y, en el bloque 844, ajustar el parámetro gráfico en función del número detectado de parpadeos del usuario para mejorar la percepción del usuario. Algunas realizaciones además pueden incluir, en el bloque 845, detectar un impedimento visual del usuario. Por ejemplo, en el bloque 846, el impedimento visual detectado puede incluir una o más de la presencia de gafas y la presencia de un parche en el ojo. Algunas realizaciones también pueden incluir, en el bloque 847, detectar una cantidad de luz ambiental y, en el bloque 848, ajustar el parámetro gráfico en función de la cantidad de luz ambiental detectada. Por ejemplo, el método 800 además puede incluir, en el bloque 849, ajustar

predictivamente el parámetro gráfico en función de una comparación de la expresión facial detectada y las expresiones faciales detectadas previamente.

Las realizaciones del método 800 se pueden implementar en un sistema, aparato, GPU, PPU o en un aparato de canalización de procesador gráfico tal como, por ejemplo, los descritos en el presente documento. Más particularmente, las implementaciones de hardware del método 800 pueden incluir lógica configurable tal como, por ejemplo, PLA, FPGA, CPLD, o en hardware de lógica de funcionalidad fija usando tecnología de circuitos tal como, por ejemplo, tecnología ASIC, CMOS o TTL, o cualquier combinación de las mismas. Como alternativa, o adicionalmente, el método 800 se puede implementar en uno o más módulos como un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento legible por máquina u ordenador, tal como RAM, ROM, PROM, firmware, memoria flash, etc., para ser ejecutadas por un procesador o dispositivo informático. Por ejemplo, el código de programa informático para llevar a cabo las operaciones de los componentes se puede escribir en cualquier combinación de uno o más lenguajes de programación apropiados/aplicables al sistema operativo, incluyendo un lenguaje de programación orientado a objetos tal como PYTHON, PERL, JAVA, SMALLTALK, C++, C# o similares y lenguajes de programación de procedurales convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares. Por ejemplo, el método 800 se puede implementar en un medio legible por ordenador como se describe en relación con los ejemplos 36 a 51, más adelante.

Por ejemplo, las realizaciones o partes del método 800 se pueden implementar en aplicaciones (p. ej., a través de una API) o software de controlador. Otras realizaciones o partes del método 800 se pueden implementar en código especializado (p. ej., sombreadores) para ejecutarse en una GPU. Otras realizaciones o partes del método 800 se pueden implementar en lógica de función fija o hardware especializado (p. ej., en la GPU).

Volviendo ahora a la Figura 9, un sistema gráfico 900 puede incluir un visualizador 902 y una cámara 904 para capturar información ocular para un usuario 906. La cámara 904 puede capturar una imagen de uno o ambos ojos del usuario 906. La imagen puede incluir una parte de la cara del usuario 906, incluyendo los párpados y/o el área de la cara cerca de los ojos del usuario 906. El sistema 900 puede procesar la imagen de la cámara 904 para detectar si el usuario 906 entrecierra los ojos, parpadea y/o guiña un ojo. El sistema 900 puede procesar adicionalmente o, como alternativa, la imagen de la cámara para detectar la tensión de los músculos faciales (p. ej., si la piel está arrugada o fruncida cerca de los ojos). Por ejemplo, si ambos ojos se cierran y se vuelven a abrir en aproximadamente 100-200 ms, el sistema 900 puede determinar que el usuario ha parpadeado. Si ambos párpados se cierran parcialmente, el sistema 900 puede determinar que el usuario está entrecerrando los ojos. Si uno ojo se cierra mientras el otro permanece abierto, el sistema 900 puede determinar que el usuario está guiñando un ojo. Tanto para el entrecerrado de ojos como para el guiño de un ojo, el sistema 900 puede analizar la información de la tensión de los músculos faciales junto con la información del párpado para determinar si el usuario está entrecerrando los ojos o guiñando el ojo.

Las acciones asociadas y/o los parámetros de rendimiento ajustados pueden ser específicos del usuario (p. ej., basados en un proceso de configuración y/o basados en la calibración del usuario). En algunas realizaciones, se puede seleccionar un promedio, o el usuario puede seleccionar de entre un conjunto de perfiles predeterminados para diversas expresiones faciales. En algunas realizaciones, puede haber una calibración por usuario. Por ejemplo, algunas personas pueden guiñar un ojo con un ojo abierto, mientras que otras solo pueden hacerlo con ambos ojos cerrados. Diferentes personas pueden tener diferentes distancias de descanso entre sus párpados. Por ejemplo, una calibración puede proporcionar un reconocimiento más preciso de las expresiones faciales del usuario. La cámara 904 puede ser parte de un seguidor ocular. Generalmente, los seguidores oculares implican cierta calibración por parte del usuario. La calibración de expresiones faciales se puede realizar al mismo tiempo que la calibración del seguidor ocular. Durante una fase de la calibración, se le puede pedir al usuario que se focalice en un área y entrecierre los ojos, parpadee y/o guiñe un ojo. En algunas realizaciones, la calibración u otras configuraciones (p. ej., tipo o cantidad de ajuste asociado con una expresión facial pueden ser ajustables por parte del usuario (p. ej., configuración/barra deslizante para más ajustes/menos ajustes) o incluirse como parte de varias configuraciones de potencia/rendimiento.

Algunas realizaciones, ventajosamente, pueden proporcionar una función de "entrecerrado de ojos para hacer zum" basada en el seguimiento del enfoque de los auriculares. Por ejemplo, algunas realizaciones pueden usar seguidores oculares para seguir las distancias de los párpados para detectar un "entrecerrado de ojos" por parte del usuario y mapear tal condición detectada a la tendencia natural del usuario con el entrecerrado de ojos para ver objetos distantes. Si el sistema detecta que el usuario está entrecerrando los ojos, puede proporcionar una señal a la aplicación actual para ampliar la escena actual (p. ej., o simplemente ampliar el área de enfoque actual del usuario). Además, o como alternativa, para una aplicación de realidad aumentada (RA), la señal de "entrecerrado de ojos" puede hacer que la aplicación ajuste el enfoque de una cámara de RA. En algunas realizaciones, el sistema puede combinar el entrecerrado de ojos detectado con otra entrada (p. ej., pulso, frecuencia cardíaca, reconocimiento de gestos tal como señalar) para desencadenar una acción asociada. Del mismo modo, algún otro reconocimiento facial (p. ej., tal como aplastar la cabeza o alguna otra tensión muscular facial detectada) puede desencadenar una acción asociada (p. ej., hacer zum o una panorámica). Por ejemplo, un guiño con el ojo derecho puede hacer que la aplicación haga una panorámica/barrido hacia la derecha, mientras que un guiño con el ojo izquierdo puede hacer que la aplicación haga una panorámica/barrido hacia la izquierda. Arrugar la nariz puede hacer que la aplicación haga una panorámica/barrido hacia arriba, mientras que dejar caer la mandíbula puede hacer que la aplicación haga una panorámica/barrido hacia abajo.

Algunas realizaciones pueden controlar adicionalmente o, como alternativa, un sistema de realidad virtual en función del parpadeo. Por ejemplo, algunas realizaciones pueden proporcionar un control de manos libres para ajustar la resolución de una imagen de realidad virtual utilizando un parpadeo voluntario detectado. En algunas realizaciones, un número de parpadeos (p. ej., parpadear dos veces) pueden hacer que el sistema mejore la resolución de la imagen de realidad virtual (p. ej., o cambie el contraste o algún otro parámetro o acción).

Algunas realizaciones, ventajosamente, pueden proporcionar un ajuste de la calidad de la visualización para adaptarse a las necesidades visuales del usuario. Por ejemplo, el visualizador puede estar configurado con una configuración de calidad inicial de forma predeterminada. Una cámara puede seguir al usuario para determinar si el usuario lleva gafas y/o muestra ciertos rasgos faciales (p. ej., entrecerrado de ojos, tensión de los músculos faciales, etc.). El sistema puede ajustar las características de visualización (p. ej., brillo, contraste, etc.) en función de las características detectadas del usuario. Por ejemplo, el sistema puede ajustar la calidad de visualización aumentando el brillo y el contraste de los colores (p. ej., configuraciones no usadas por defecto debido a una mayor disipación de energía), lo que facilita al usuario ver el contenido. Por ejemplo, si el usuario entrecierra los ojos en las condiciones de iluminación actuales, el sistema puede aumentar el brillo de visualización hasta que el usuario deje de entrecerrar los ojos y luego guardar la configuración como referencia futura en condiciones de iluminación similares.

Volviendo ahora a las Figuras 10A a 10B, un sistema de representación foveada 1000 de acuerdo con la invención proporciona una región de resolución relativamente más alta 1002 y una región de resolución relativamente más baja 1004 de un visualizador 1006 basada en un área de enfoque de un usuario 1008. De acuerdo con la invención, se proporciona un seguimiento de la expresión facial para la representación foveada. Por ejemplo, algunas realizaciones pueden detectar expresiones faciales tales como un entrecerrado de ojos para controlar un área de enfoque. El tamaño de la región de resolución relativamente más alta 1002 se puede ajustar en función del entrecerrado detectado del usuario 1008 (p. ej., véase la Figura 10B). Además o, como alternativa, algunas realizaciones pueden alterar la representación foveada para la región de resolución relativamente más alta 1002 para cambiar atributos tales como la tasa de sombreado, el contraste, etc. para hacer que al usuario 1008 le sea más fácil percibir la escena.

Descripción general del sistema de interfaz integrada montado en la cabeza

La Figura 11 muestra un sistema de visualización montado en la cabeza (HMD) 1100 que lleva puesto un usuario mientras experimenta un entorno inmersivo tal como, por ejemplo, un entorno de realidad virtual (RV), un entorno de realidad aumentada (RA), un juego tridimensional (3D) multijugador y así sucesivamente. En el ejemplo ilustrado, una o más correas 1120 sujetan una montura 1102 del sistema de HMD 1100 enfrente de los ojos del usuario. En consecuencia, se puede situar un visualizador de ojo izquierdo 1104 para que lo vea el ojo izquierdo del usuario y se puede situar un visualizador de ojo derecho 1106 para que lo vea el ojo derecho del usuario. Como alternativa, el visualizador de ojo izquierdo 1104 y el visualizador de ojo derecho 1106 se pueden integrar en una única pantalla en determinados ejemplos tales como, por ejemplo, un teléfono inteligente que lleva el usuario. En el caso de RA, los visualizadores 1104, 1106 pueden ser visualizadores transparentes que permiten al usuario ver el entorno físico, con otro contenido representado (p. ej., personajes virtuales, anotaciones informativas, visualizador frontal/HUD) que se presenta por encima de una transmisión en vivo del entorno físico.

En un ejemplo, la montura 1102 incluye una cámara orientada hacia abajo a la izquierda 1108 para capturar imágenes de un área generalmente enfrente del usuario y debajo del ojo izquierdo (p. ej., gestos con la mano izquierda). Además, una cámara orientada hacia abajo a la derecha 1110 puede capturar imágenes de un área generalmente enfrente del usuario y debajo del ojo derecho (p. ej., gestos con la mano derecha). La montura 1102 ilustrada también incluye una cámara orientada hacia delante a la izquierda 1112 y una cámara orientada hacia delante a la derecha 1114 para capturar imágenes enfrente de los ojos izquierdo y derecho, respectivamente, del usuario. La montura 1102 también puede incluir una cámara orientada hacia el lado izquierdo 1116 para capturar imágenes desde un área a la izquierda del usuario y una cámara orientada hacia el lado derecho 1118 para capturar imágenes desde un área a la derecha del usuario.

Las imágenes captadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118, que pueden tener campos de visión superpuestos, se pueden usar para detectar gestos realizados por el usuario, así como para analizar y/o reproducir el entorno externo en los visualizadores 1104, 1106. En un ejemplo, una arquitectura de procesamiento de gráficos (p. ej., interna y/o externa) usa los gestos detectados para representar y/o controlar una representación virtual del usuario en un juego 3D. De hecho, los campos de visión superpuestos pueden permitir la captura de gestos realizados por otros individuos (p. ej., en un juego multijugador), donde los gestos de otros individuos además se pueden usar para representar/controlar la experiencia inmersiva. Los campos de visión superpuestos también pueden permitir que el sistema de HMD 1100 detecte automáticamente obstáculos u otros peligros cerca del usuario. Tal enfoque puede ser particularmente ventajoso en las aplicaciones de sistema avanzado de asistencia al conductor (ADAS).

En un ejemplo, proporcionar unos campos de visión superpuestos a la cámara orientada hacia abajo a la izquierda 1108 y a la cámara orientada hacia abajo a la derecha 1110 proporciona una vista estereoscópica que tiene una resolución aumentada. La resolución aumentada puede permitir, a su vez, que movimientos de usuario muy similares se distingan entre sí (p. ej., con una precisión submilimétrica). El resultado puede ser un rendimiento mejorado del

sistema de HMD 1100 con respecto a la fiabilidad. De hecho, la solución ilustrada puede ser útil en una amplia gama de aplicaciones tales como, por ejemplo, colorear información en configuraciones de RA, intercambiar herramientas/dispositivos virtuales entre usuarios en un entorno multiusuario, representar elementos virtuales (p. ej., armas, espadas, personal) y así sucesivamente. Los gestos de otros objetos, extremidades y/o partes del cuerpo también se pueden detectar y usar para representar/controlar el entorno virtual. Por ejemplo, las señales mielográficas, las señales electroencefalográficas, el seguimiento ocular, la respiración o el resoplido, los movimientos de las manos, etc., se pueden seguir en tiempo real, ya sea del portador o de otro individuo en un entorno compartido. Las imágenes capturadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118 también pueden servir como entrada contextual. Por ejemplo, se podría determinar que el usuario está indicando una palabra en particular para editar o una tecla para presionar en una aplicación de procesamiento de texto, un arma en particular para desplegar o una dirección de desplazamiento en un juego y así sucesivamente.

Adicionalmente, las imágenes capturadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118 se pueden usar para llevar a cabo una comunicación compartida o una interactividad en red en aplicaciones de funcionamiento de equipos, entrenamiento médico y/o guiado remoto/teleoperativo. Las bibliotecas de gestos específicos de tareas o el aprendizaje automático de redes neuronales podrían permitir la identificación de herramientas y la realimentación para una tarea. Por ejemplo, se puede habilitar una herramienta virtual que los convierta en acciones reales remotas. En otro ejemplo más, el sistema de HMD 1100 convierte la manipulación de un taladro virtual dentro de una escena virtual en la operación remota de un taladro en un dispositivo robótico desplegado para buscar en un edificio derrumbado. Además, el sistema de HMD 1100 puede programarse en la medida en que incluye, por ejemplo, un protocolo que permite al usuario añadir un nuevo gesto a una lista de gestos identificables asociados con las acciones del usuario.

Además, las diversas cámaras del HMD 1100 se pueden configurar para detectar frecuencias de espectro además de las longitudes de onda visibles del espectro. Las capacidades de formación de imágenes multispectrales en las cámaras de entrada permiten el seguimiento de la posición del usuario y/o de los objetos eliminando características de imagen no esenciales (p. ej., ruido de fondo). Por ejemplo, en aplicaciones de realidad aumentada (RA), tales como cirugía, los instrumentos y equipos se pueden seguir por su reflectividad infrarroja sin necesidad de ayudas de seguimiento adicionales. Además, el HMD 1100 se podría emplear en situaciones de baja visibilidad donde una "transmisión en vivo" de las diversas cámaras se podría mejorar o aumentar a través de un análisis por ordenador y mostrarse al usuario como señales visuales o de audio.

El sistema de HMD 1100 también puede renunciar a realizar cualquier tipo de comunicación de datos con un sistema informático remoto o necesitar cables de alimentación (p. ej., modo de operación independiente). A este respecto, el sistema de HMD 1100 puede ser un dispositivo "sin cables" que tenga una unidad de alimentación que permite que el sistema de HMD 1100 opere independientemente de los sistemas de alimentación externos. En consecuencia, el usuario puede desempeñar un papel completo sin estar conectado a otro dispositivo (p. ej., una consola de juegos) o fuente de alimentación. En un ejemplo de procesamiento de textos, el sistema de HMD 1100 podría presentar un teclado virtual y/o un ratón virtual en las pantallas 1104 y 1106 para proporcionar un escritorio virtual o una escena de procesamiento de textos. Por lo tanto, los datos de reconocimiento de gestos capturados por una o más de las cámaras pueden representar actividades de escritura del usuario en el teclado virtual o movimientos del ratón virtual. Las ventajas incluyen, pero sin limitación, la facilidad de portabilidad y la privacidad del escritorio virtual de los individuos cercanos. La arquitectura de procesamiento de gráficos subyacente puede soportar la compresión y/o descompresión de señales de vídeo y audio. Además, proporcionar imágenes separadas para el ojo izquierdo y el ojo derecho del usuario puede facilitar la representación, generación y/o percepción de escenas 3D. Las posiciones relativas del visualizador de ojo izquierdo 1104 y del visualizador de ojo derecho 1106 también pueden ajustarse para adaptarse a las variaciones en la separación de ojos entre los diferentes usuarios.

El número de cámaras ilustradas en la Figura 11 es para facilitar el análisis únicamente. De hecho, el sistema de HMD 1100 puede incluir menos de seis o más de seis cámaras, dependiendo de las circunstancias.

Componentes funcionales del sistema de HMD

La Figura 12 muestra el sistema de HMD con mayor detalle. En el ejemplo ilustrado, la montura 1102 incluye una unidad de alimentación 1200 (p. ej., alimentación con batería, adaptador) para proporcionar alimentación al sistema de HMD. La montura 1102 ilustrada también incluye un módulo de seguimiento de movimiento 1220 (p. ej., acelerómetros, giroscopios), en donde el módulo de seguimiento de movimiento 1220 proporciona datos de seguimiento de movimiento, datos de orientación y/o datos de posición a un sistema de procesador 1204. El sistema de procesador 1204 puede incluir un adaptador de red 1224 que está acoplado a un puente de E/S 1206. El puente de E/S 1206 puede permitir comunicaciones entre el adaptador de red 1224 y diversos componentes tales como, por ejemplo, módulos de entrada de audio 1210, módulos de salida de audio 1208, un dispositivo de visualización 1207, cámaras de entrada 1202, y así sucesivamente.

En el ejemplo ilustrado, los módulos de entrada de audio 1210 incluyen una entrada de audio derecha 1218 y una entrada de audio izquierda 1216, que detectan el sonido que se puede procesar para reconocer los comandos de voz del usuario y de los individuos cercanos. Los comandos de voz reconocidos en las señales de audio capturadas pueden aumentar el reconocimiento de gestos durante la conmutación de modalidad y otras aplicaciones. Además,

las señales de audio capturadas pueden proporcionar información en 3D que se usa para mejorar la experiencia inmersiva.

Los módulos de salida de audio 1208 pueden incluir una salida de audio derecha 1214 y una salida de audio izquierda 1212. Los módulos de salida de audio 1208 pueden suministrar sonido a los oídos del usuario y/u otros individuos cercanos. Los módulos de salida de audio 1208, que pueden tener forma de auriculares, altavoces de oído, altavoces sobre la oreja, altavoces, etc., o cualquier combinación de los mismos, pueden entregar contenido de audio estéreo y/o 3D al usuario (p. ej., localización espacial). La montura 1102 ilustrada también incluye un módulo inalámbrico 1222, que puede facilitar las comunicaciones entre el sistema de HMD y diversos otros sistemas (p. ej., ordenadores, dispositivos ponibles, consolas de juego). En un ejemplo, el módulo inalámbrico 1222 se comunica con el sistema de procesador 1204 a través del adaptador de red 1224.

El dispositivo de visualización ilustrado 1207 incluye el visualizador de ojo izquierdo 1104 y el visualizador de ojo derecho 1106, en donde el contenido visual presentado en los visualizadores 1104, 1106 se puede obtener del sistema de procesador 1204 a través del puente de E/S 1206. Las cámaras de entrada 1202 pueden incluir la cámara orientada hacia el lado izquierdo 1116, la cámara orientada hacia el lado derecho 1118, la cámara orientada hacia abajo a la izquierda 1108, la cámara orientada hacia delante a la izquierda 1112, la cámara orientada hacia delante a la derecha 1114 y la cámara orientada hacia abajo a la derecha 1110, ya analizadas.

Volviendo ahora a la Figura 13, se muestra una agrupación de procesamiento general (GPC) 1300. La GPC 1300 ilustrada se puede incorporar a un sistema de procesamiento tal como, por ejemplo, el sistema de procesamiento 1204 (Figura 12), ya analizado. La GPC 1300 puede incluir un gestor canalizaciones 1302 que se comunica con un planificador. En un ejemplo, el gestor canalizaciones 1302 recibe tareas del programador y distribuye las tareas a uno o más multiprocesadores (SM) de envío por flujo continuo 1304. Cada SM 1304 se puede configurar para procesar grupos de hilos, donde un grupo de hilos se puede considerar una pluralidad de hilos relacionados que ejecutan operaciones iguales o similares en diferentes datos de entrada. Por lo tanto, cada hilo del grupo de hilos se puede asignar a un SM 1304 particular. En otro ejemplo, el número de hilos puede ser mayor que el número de unidades de ejecución en el SM 1304. En este sentido, los hilos de un grupo de hilos pueden operar en paralelo. El gestor de canalizaciones 1302 también puede especificar destinos de datos procesados a una barra transversal de distribución de trabajo 1308, que se comunica con una barra transversal de memoria.

Por lo tanto, como cada SM 1304 transmite una tarea procesada a la barra transversal de distribución de trabajo 1308, se puede proporcionar la tarea procesada a otra GPC 1300 para su procesamiento adicional. La salida del SM 1304 también se puede enviar a una unidad de operaciones de prerrasterización (preROP) 1314, que a su vez dirige los datos a una o más unidades de operaciones de rasterización o realiza otras operaciones (p. ej., realizar conversiones de direcciones, organizar datos de color de imagen, mezcla de colores y así sucesivamente). El SM 1304 puede incluir una memoria caché de nivel uno (L1) interna (no mostrada) en la que el SM 1304 puede almacenar datos. El SM 1304 también puede tener acceso a una memoria caché de nivel dos (L2) (no mostrada) a través de una unidad de gestión de memoria (MMU) 1310 y una memoria caché de nivel uno punto cinco (L1.5) 1306. La MMU 1310 puede mapear direcciones virtuales a direcciones físicas. A este respecto, la MMU 1310 puede incluir entradas de tabla de páginas (PTE) que se usan para mapear direcciones virtuales a direcciones físicas de una tesela, página de memoria y/o índice de línea de caché. La GPC 1300 ilustrada también incluye una unidad de texturas 1312.

Arquitectura de canalización de gráficos

Volviendo ahora a la Figura 14, se muestra una canalización de gráficos 1400. En el ejemplo ilustrado, una canalización de espacio global 1420 incluye un distribuidor de primitivas (PD) 1402. El PD 1402 puede recopilar datos de vértices asociados con servicios de alto orden, primitivas de gráficos, triángulos, etc., y transmitir los datos de vértices a una unidad de extracción de atributos de vértices (VAF) 1404. La VAF 1404 puede recuperar atributos de vértices asociados con cada uno de los vértices entrantes de la memoria compartida y almacenar los datos de vértices, junto con los atributos de vértices asociados, en la memoria compartida.

La canalización de espacio global 1420 ilustrada también incluye una unidad de procesamiento de vértices, teselación y geometría (VTG) 1406. La VTG 1406 puede incluir, por ejemplo, una unidad de procesamiento de vértices, una unidad de procesamiento de inicialización de teselación, un distribuidor de tareas, una unidad de generación de tareas, una unidad de generación de topología, una unidad de procesamiento de geometría, una unidad de procesamiento de teselación, etc., o cualquier combinación de los mismos. En un ejemplo, la VTG 1406 es una unidad de ejecución programable que está configurada para ejecutar programas de geometría, programas de teselación y programas de sombreado de vértices. Los programas ejecutados por la VTG 1406 pueden procesar los datos de vértices y los atributos de vértices recibidos de la VAF 1404. Además, los programas ejecutados por la VTG 1406 pueden producir primitivas de gráficos, valores de color, factores normales de superficie y valores de transparencia en cada vértice para las primitivas de gráficos para su procesamiento adicional dentro de la canalización de procesamiento de gráficos 1400.

La unidad de procesamiento de vértices 1406 de la VTG es una unidad de ejecución programable que ejecuta programas de sombreado de vértices, datos de vértices de iluminación y transformación como especifican los

programas de sombreador de vértices. Por ejemplo, la unidad de procesamiento de vértices podría programarse para transformar los datos de vértices de una representación de coordenadas basada en objetos (p. ej., espacio de objetos) a un sistema de coordenadas basado de manera alternativa, tal como el espacio global o el espacio de coordenadas de dispositivo normalizado (NDC). Adicionalmente, la unidad de procesamiento de vértices puede leer datos de vértices y atributos de vértices que están almacenados en la memoria compartida por la VAF 1404 y procesar los datos de vértices y los atributos de vértices. En un ejemplo, la unidad de procesamiento de vértices almacena los vértices procesados en memoria compartida.

La unidad de procesamiento de inicialización de teselación (p. ej., sombreador de casco, sombreador de control de teselación) puede ejecutar programas de sombreador de inicialización de teselación. En un ejemplo, la unidad de procesamiento de inicialización de teselación procesa los vértices producidos por la unidad de procesamiento de vértices y genera primitivas de gráficos a las que en ocasiones se hace referencia como "parches". La unidad de procesamiento de inicialización de teselación también puede generar diversos atributos de parche, en donde los datos de parche y los atributos de parche se almacenan en memoria compartida. La unidad de generación de tareas de la VTG 1406 puede recuperar datos y atributos para vértices y parches de la memoria compartida. En un ejemplo, la unidad de generación de tareas genera tareas para procesar los vértices y parches para su procesamiento por las últimas etapas en la canalización de procesamiento de gráficos 1400.

Las tareas producidas por la unidad de generación de tareas pueden ser redistribuidas por el distribuidor de tareas de la VTG 1406. Por ejemplo, las tareas producidas por las diversas instancias del programa de sombreador de vértices y el programa de inicialización de teselación pueden variar significativamente entre una canalización de procesamiento de gráficos 1400 y otra. En consecuencia, el distribuidor de tareas puede redistribuir estas tareas de tal manera que cada canalización de procesamiento de gráficos 1400 tenga aproximadamente la misma carga de trabajo durante las etapas de canalización posteriores.

Como ya se indicó, la VTG 1406 también puede incluir una unidad de generación de topología. En un ejemplo, la unidad de generación de topología recupera tareas distribuidas por el distribuidor de tareas, indexa los vértices, incluyendo los vértices asociados con parches y calcula las coordenadas (UV) para los vértices de teselación y los índices que conectan los vértices de teselación para formar primitivas de gráficos. Los vértices indexados pueden ser almacenados por la unidad de generación de topología en memoria compartida. La unidad de procesamiento de teselación de la VTG 1406 se puede configurar para ejecutar programas de sombreador de teselación (p. ej., sombreadores de dominio, sombreadores de evaluación de teselación). La unidad de procesamiento de teselación puede leer datos de entrada de la memoria compartida y escribir datos de salida en la memoria compartida. Los datos de salida se pueden pasar de la memoria compartida a la unidad de procesamiento de geometría (p. ej., la siguiente etapa del sombreador) como datos de entrada.

La unidad de procesamiento de geometría de la VTG 1406 puede ejecutar programas de sombreador de geometría para transformar primitivas de gráficos (p. ej., triángulos, segmentos de línea, puntos, etc.). En un ejemplo, los vértices se agrupan para construir primitivas de gráficos, en donde la unidad de procesamiento de geometría subdivide las primitivas de gráficos en una o más primitivas de gráficos nuevas. La unidad de procesamiento de geometría también puede calcular parámetros tales como, por ejemplo, coeficientes de ecuaciones simples, que pueden usarse para rasterizar las nuevas primitivas de gráficos.

La canalización de espacio global 1420 ilustrada también incluye una unidad de escala, selección y recorte de ventana gráfica (VPC) 1408 que recibe los parámetros y vértices que especifican nuevas primitivas de gráficos de la VTG 1406. En un ejemplo, la VPC 1408 realiza el recorte, la selección, la corrección de la perspectiva y la transformación de la ventana gráfica para identificar las primitivas de gráficos que son potencialmente visibles en la imagen final representada. La VPC 1408 también puede identificar las primitivas de gráficos que pueden no ser visibles.

La canalización de procesamiento de gráficos 1400 también puede incluir una unidad de generación de teselas 1410 acoplada a la canalización de espacio global 1420. La unidad de generación de teselas 1410 puede ser un motor de clasificación de primitivas de gráficos, en donde las primitivas de gráficos se procesan en la canalización de espacio global 1420 y a continuación se transmiten a la unidad de generación de teselas 1410. A este respecto, la canalización de procesamiento de gráficos 1400 también puede incluir una canalización de espacio de pantalla 1422, en donde el espacio de pantalla puede dividirse en teselas de caché. Por lo tanto, cada tesela de caché puede estar asociada con una parte del espacio de pantalla. Para cada primitiva de gráficos, la unidad de generación de teselas 1410 puede identificar el conjunto de teselas de caché que se cruzan con la primitiva de gráficos (p. ej., "generación de teselas"). Después de generar teselas de un número de primitivas de gráficos, la unidad de generación de teselas 1410 puede procesar las primitivas de gráficos basándose en una tesela de caché. En un ejemplo, las primitivas de gráficos asociadas con una tesela de caché particular se transmiten a una unidad de configuración 1412 en la canalización de espacio de pantalla 1422, una tesela cada vez. Las primitivas de gráficos que se cruzan con múltiples teselas de caché se pueden procesar una vez en la canalización de espacio global 1420, mientras se transmiten múltiples veces a la canalización de espacio de pantalla 1422.

En un ejemplo, la unidad de configuración 1412 recibe datos de vértices desde la VPC 1408 a través de la unidad de generación de teselas 1410 y calcula los parámetros asociados con las primitivas de gráficos. Los parámetros pueden

incluir, por ejemplo, ecuaciones de borde, ecuaciones de plano parcial y ecuaciones de plano de profundidad. La canalización de espacio de pantalla 1422 también puede incluir un rasterizador 1414 acoplado a la unidad de configuración 1412. El rasterizador puede explorar y convertir las nuevas primitivas de gráficos y transmitir fragmentos y datos de cobertura a una unidad de sombreado de píxeles (PS) 1416. El rasterizador 1414 también puede realizar selección Z y otras optimizaciones basadas en Z.

El PS 1416, que puede acceder a la memoria compartida, puede ejecutar programas de sombreador de fragmentos que transforman los fragmentos recibidos del rasterizador 1414. Más particularmente, los programas de sombreador de fragmentos pueden sombrear fragmentos con granularidad a nivel de píxel (p. ej., funcionando como programas de sombreador de píxeles). En otro ejemplo, el sombreador de fragmentos programa fragmentos de sombreado con una granularidad a nivel de muestra, donde cada píxel incluye múltiples muestras y cada muestra representa una parte de un píxel. Además, los programas de sombreador de fragmentos pueden sombrear fragmentos con cualquier otra granularidad, dependiendo de las circunstancias (p. ej., la tasa de muestreo). La PS 1416 puede realizar mezcla, sombreado, corrección de perspectiva, asignación de texturas, etc., para generar fragmentos sombreados.

La canalización de espacio de pantalla 1422 ilustrada también incluye una unidad de operaciones de rasterización (ROP) 1418, que puede realizar operaciones de rasterización tales como, por ejemplo, estarcido, prueba Z, mezcla y así sucesivamente. A continuación, la ROP 1418 puede transmitir datos de píxeles como datos de gráficos procesados a uno o más objetivos representados (p. ej., memoria de gráficos). La ROP 1418 se puede configurar 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. La ubicación de la ROP 1418 puede variar dependiendo de las circunstancias.

La canalización de procesamiento de gráficos 1400 puede ser implementada por uno o más elementos de procesamiento. Por ejemplo, la VTG 1406 y/o la PS 1416 se pueden implementar en uno o más SM, el PD 1402, la VAF 1404, la VPC 1408, la unidad de generación de teselas 1410, la unidad de configuración 1412, el rasterizador 1414 y/o la ROP 1418 se pueden implementar en elementos de procesamiento de una GPC particular junto con una unidad de subdivisión correspondiente. La canalización de procesamiento de gráficos 1400 también se puede implementar en la lógica de hardware de funcionalidad fija. De hecho, la canalización de procesamiento de gráficos 1400 se puede implementar en una PPU.

Por lo tanto, la canalización de espacio global 1420 ilustrada procesa objetos gráficos en el espacio 3D, donde la posición de cada objeto gráfico es conocida en relación con otros objetos gráficos y en relación con un sistema de coordenadas 3D. Por el contrario, la canalización de espacio de pantalla 1422 puede procesar objetos gráficos que han sido proyectados desde el sistema de coordenadas 3D sobre una superficie plana 2D que representa la superficie del dispositivo de visualización. Además, la canalización de espacio global 1420 se puede dividir en una canalización de fase alfa y una canalización de fase beta, en donde la canalización de fase alfa incluye etapas de canalización desde el PD 1402 hasta la unidad de generación de tareas. La canalización de fase beta puede incluir etapas de canalización desde la unidad de generación de topología hasta la VPC 1408. En un caso de este tipo, la canalización de procesamiento de gráficos 1400 puede realizar un primer conjunto de operaciones (p. ej., un único hilo, un grupo de hilos, múltiples grupos de hilos que actúan al unísono) en la canalización de fase alfa y un segundo conjunto de operaciones (p. ej., un único hilo, un grupo de hilos, múltiples grupos de hilos que actúan al unísono) en la canalización de fase beta.

Si se usan múltiples canalizaciones de procesamiento de gráficos 1400, los datos de vértices y los atributos de vértices asociados con un conjunto de objetos gráficos pueden dividirse para que cada canalización de procesamiento de gráficos 1400 tenga una carga de trabajo similar a través de la fase alfa. En consecuencia, el procesamiento de fase alfa puede expandir sustancialmente la cantidad de datos de vértices y atributos de vértices, de tal manera que la cantidad de datos de vértices y atributos de vértices producidos por la unidad de generación de tareas es significativamente mayor que la cantidad de datos de vértices y atributos de vértices procesados por el PD 1402 y la VAF 1404. Además, las unidades de generación de tareas asociadas con diferentes canalizaciones de procesamiento de gráficos 1400 pueden producir datos de vértices y atributos de vértices que tienen diferentes niveles de calidad, incluso cuando comienzan la fase alfa con la misma cantidad de atributos. En tales casos, el distribuidor de tareas puede redistribuir los atributos producidos por la canalización de fase alfa de modo que cada canalización de procesamiento de gráficos 1400 tenga aproximadamente la misma carga de trabajo al comienzo de la canalización de fase beta.

Volviendo ahora a la Figura 15, se muestra un multiprocesador de envío por flujo continuo (SM) 1500. El SM 1500 ilustrado incluye K unidades de planificación 1504 acopladas a una memoria caché de instrucciones 1502, en donde cada unidad de planificación 1504 recibe una matriz de bloques de hilos de un gestor de canalizaciones (no mostrado) y gestiona la planificación de instrucciones para uno o más bloques de hilos de cada matriz de bloques de hilos activa. La unidad de planificación 1504 puede planificar hilos para su ejecución en grupos de hilos paralelos, donde cada grupo puede denominarse "urdimbre". Por tanto, cada urdimbre podría incluir, por ejemplo, sesenta y cuatro hilos. Adicionalmente, la unidad de planificación 1504 puede gestionar una pluralidad de bloques de hilos diferentes, asignando los bloques de hilos a urdimbres para su ejecución. A continuación, la unidad de planificación puede planificar instrucciones de la pluralidad de urdimbres diferentes en diversas unidades funcionales durante cada ciclo de reloj. Cada unidad de planificación 1504 puede incluir una o más unidades de despacho de instrucciones 1522, en

donde cada unidad de despacho 1522 transmite instrucciones a una o más de las unidades funcionales. El número de unidades de despacho 1522 puede variar dependiendo de las circunstancias. En el ejemplo ilustrado, la unidad de planificación 1504 incluye dos unidades de despacho 1522 que permiten despachar dos instrucciones diferentes de la misma urdimbre durante cada ciclo de reloj.

El SM 1500 también puede incluir un archivo de registro 1506. El archivo de registro 1506 puede incluir un conjunto de registros que se dividen entre las unidades funcionales de tal manera que a cada unidad funcional se le asigna una parte especializada del archivo de registro 1506. El archivo de registro 1506 también se puede dividir entre diferentes urdimbres que ejecuta el SM 1500. En un ejemplo, el archivo de registro 1506 proporciona un almacenamiento temporal para operandos conectados a las rutas de datos de las unidades funcionales. El SM 1500 ilustrado también incluye L núcleos de procesamiento 1508, en donde L puede ser un número relativamente grande (p. ej., 192). Cada núcleo 1508 puede ser una unidad de procesamiento de precisión simple canalizada que incluye una unidad lógica aritmética de coma flotante (p. ej., IEEE 754-2008) así como una unidad aritmética de números enteros.

El SM 1500 ilustrado también incluye M unidades de doble precisión (DPU) 1510, N unidades de funciones especiales (SFU) 1512 y P unidades de carga/almacenamiento (LSU) 1514. Cada DPU 1510 puede implementar aritmética de coma flotante de doble precisión y cada SFU 1512 puede realizar funciones especiales tales como, por ejemplo, la mezcla de píxeles de copia de rectángulo, etc. Además, cada LSU 1514 puede llevar a cabo operaciones de carga y almacenamiento entre una memoria compartida 1518 y el archivo de registro 1506. En un ejemplo, las operaciones de carga y almacenamiento se llevan a cabo a través de unidades de textura J/memorias caché LI 1520 y una red de interconexión 1516. En un ejemplo, la unidad de textura J/memorias caché LI 1520 también están acopladas a una barra transversal (no mostrada). Por lo tanto, la red de interconexión 1516 puede conectar cada una de las unidades funcionales al archivo de registro 1506 y a la memoria compartida 1518. En un ejemplo, la red de interconexión 1516 funciona como una barra transversal que conecta cualquiera de las unidades funcionales a cualquiera de los registros del archivo de registro 1506.

El SM 1500 puede implementarse dentro de un procesador gráfico (p. ej., unidad de procesamiento de gráficos/GPU), en donde la unidad de textura/memorias caché LI 1520 puede acceder a mapas de textura desde la memoria y muestrear los mapas de textura para producir valores de textura muestreados para su uso en programas de sombreador. Las operaciones de textura realizadas por la unidad de textura/memorias caché LI 1520 incluyen, pero sin limitación, suavizado basado en mipmaps.

Ejemplo de descripción 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 gráficos 1608, y puede ser un sistema de sobremesa de procesador único, un sistema de estación de trabajo de multiprocesador o un sistema de servidor que tenga 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 embebidos.

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

En algunas realizaciones, cada uno de los uno o más procesadores 1602 incluye uno o más núcleos de procesador 1607 para procesar instrucciones que, cuando se ejecutan, realizan operaciones para software de usuario y sistema. En algunas realizaciones, cada uno de los uno o más núcleos de procesador 1607 está configurado para procesar un conjunto de instrucciones 1609 específico. En algunas realizaciones, el conjunto de instrucciones 1609 puede facilitar la Computación de Conjunto de Instrucciones Complejas (CISC), la Computación de Conjunto de Instrucciones Reducidas (RISC) o la computación a través de una Palabra de Instrucción Muy Larga (VLIW). Múltiples núcleos de procesador 1607 pueden procesar, cada uno, un conjunto de instrucciones 1609 diferente, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo de procesador 1607 también puede incluir otros dispositivos de procesamiento, tales como un procesador de señal digital (DSP).

En algunas realizaciones, el procesador 1602 incluye una memoria caché 1604. Dependiendo de la arquitectura, el procesador 1602 puede tener una única memoria caché interna o múltiples niveles de memoria caché interna. En algunas realizaciones, la memoria caché está compartida entre varios componentes del procesador 1602. En algunas realizaciones, el procesador 1602 también usa una memoria caché externa (p. ej., una memoria caché de nivel-3 (L3) o una memoria caché de último nivel (LLC)) (no mostrada), que puede estar compartida entre los núcleos de

procesador 1607 usando técnicas de coherencia de memoria caché conocidas. Adicionalmente, se incluye un archivo de registros 1606 en el procesador 1602, que puede incluir diferentes tipos de registros para almacenar diferentes tipos de datos (p. ej., registros de números enteros, registros de coma flotante, registros de estado y un registro de puntero a instrucción). Algunos registros pueden ser registros de propósito general, mientras que otros registros pueden ser específicos para el diseño del procesador 1602.

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

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

En algunas realizaciones, el ICH 1630 permite que los periféricos se conecten al dispositivo de memoria 1620 y al procesador 1602 a través de un bus de E/S de alta velocidad. Los periféricos de E/S incluyen, pero sin limitación, un controlador de audio 1646, una interfaz de firmware 1628, un transceptor inalámbrico 1626 (p. ej., Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1624 (p. ej., unidad de disco duro, unidad de memoria flash, etc.) y un controlador de E/S heredado 1640 para acoplar dispositivos heredados (p. ej., Sistema Personal 2 (PS/2)) al sistema. Uno o más controladores de bus serie universal (USB) 1642 conectan dispositivos de entrada, tales como combinaciones de teclado y ratón 1644. También se puede acoplar un controlador de red 1634 al ICH 1630. En algunas realizaciones, un controlador de red de alto rendimiento (no mostrado) se acopla al bus de procesador 1610. Se apreciará que el sistema 1600 mostrado es ilustrativo y no limitativo, ya que se pueden usar otros tipos de sistemas de procesamiento de datos que están configurados de manera diferente. Por ejemplo, el concentrador de controladores de E/S 1630 se puede integrar dentro de los uno o más procesadores 1602, o el concentrador de controladores de memoria 1616 y el concentrador de controladores de E/S 1630 se pueden integrar en un procesador gráfico externo discreto, tal como el procesador gráfico externo 1612.

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

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

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

En algunas realizaciones, uno o más de los núcleos de procesador 1702A-1702N incluyen soporte para múltiples hilos simultáneos. En una realización de este tipo, el núcleo de agente de sistema 1710 incluye componentes para coordinar

y hacer funcionar los núcleos 1702A-1702N durante el procesamiento de múltiples hilos. El núcleo de agente de sistema 1710 puede incluir adicionalmente una unidad de control de energía (PCU), que incluye una lógica y componentes para regular el estado de energía de los núcleos de procesador 1702A-1702N y el procesador gráfico 1708.

En algunas realizaciones, el procesador 1700 incluye adicionalmente el procesador gráfico 1708 para ejecutar las operaciones de procesamiento de gráficos. En algunas realizaciones, el procesador gráfico 1708 se acopla con el conjunto de unidades de caché compartidas 1706 y el núcleo de agente del sistema 1710, incluyendo uno o más controladores de memoria integrados 1714. En algunas realizaciones, un controlador de visualización 1711 está acoplado al procesador gráfico 1708 para controlar la salida del procesador gráfico hacia uno o más dispositivos de visualización acoplados. En algunas realizaciones, el controlador de visualización 1711 puede ser un módulo separado acoplado con el procesador gráfico a través de al menos una interconexión, o puede estar integrado dentro del procesador gráfico 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, se puede usar una unidad de interconexión alternativa, tal como una interconexión punto a punto, una interconexión conmutada u otras técnicas, incluyendo técnicas bien conocidas en la técnica. En algunas realizaciones, el procesador gráfico 1708 se acopla con la interconexión de anillo 1712 a través de un enlace de E/S 1713.

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

En algunas realizaciones, los núcleos de procesador 1702A-1702N son núcleos homogéneos que ejecutan la misma arquitectura de conjunto de instrucciones. En otra realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de arquitectura de conjunto de instrucciones (ISA), donde uno o más de los núcleos de procesador 1702A-N ejecutan un primer conjunto de instrucciones, mientras que al menos uno de los otros núcleos ejecuta un subconjunto del primer conjunto de instrucciones o un conjunto de instrucciones diferente. En una realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de microarquitectura, donde uno o más núcleos que tienen un consumo de potencia relativamente más alto se acoplan con uno o más núcleos de potencia que tienen un consumo de potencia más bajo. Adicionalmente, el procesador 1700 puede implementarse en uno o más chips o como un circuito de SoC integrado que tiene los componentes ilustrados, además de otros componentes.

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

En algunas realizaciones, el procesador gráfico 1800 también incluye un controlador de visualización 1802 para enviar datos de salida de visualización a un dispositivo de visualización 1820. El controlador de visualización 1802 incluye hardware para uno o más planos de superposición para la visualización y composición de múltiples capas de elementos de interfaz de usuario o de vídeo. En algunas realizaciones, el procesador gráfico 1800 incluye un motor de códec de vídeo 1806 para codificar, decodificar o transcodificar medios hacia, desde o entre uno o más formatos de codificación de medios, incluyendo, entre otros, formatos del Grupo de Expertos en Imágenes en Movimiento (MPEG), tales como MPEG-2, formatos de codificación de vídeo avanzada (AVC) como H.264/MPEG-4 AVC, así como el Society of Motion Picture & Television Engineers (SMPTE) 421M/VC-1 y el Joint Photographic Experts Group (JPEG) formatos como JPEG y Motion JPEG (MJPEG).

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

En algunas realizaciones, el GPE 1810 incluye una canalización 3D 1812 para realizar operaciones 3D, tal como representar imágenes y escenas tridimensionales usando funciones de procesamiento que actúan sobre formas de primitivas 3D (p. ej., rectángulo, triángulo, etc.). La canalización 3D 1812 incluye elementos de función fija y programables que realizan diversas tareas dentro del elemento y/o generan hilos de ejecución en un subsistema 3D/de

medios 1815. Aunque se puede usar la canalización 3D 1812 para realizar operaciones de medios, una realización del GPE 1810 también incluye una canalización de medios 1816 que se usa específicamente para realizar operaciones de medios, tales como postprocesamiento de vídeo y mejora de imagen.

En algunas realizaciones, la canalización de medios 1816 incluye funciones fijas o unidades lógicas programables 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 canalización de medios 1816 incluye adicionalmente una unidad de generación de hilos para generar hilos para su ejecución en el subsistema 3D/de medios 1815. Los hilos generados realizan cálculos para las operaciones de medios en una o más unidades de ejecución de gráficos incluidas en el subsistema 3D/Medios 1815.

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

Procesamiento 3D/de medios

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

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

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

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

En algunas realizaciones, el motor de estimación de movimiento 1934 proporciona aceleración de hardware para operaciones de vídeo realizando funciones de aceleración de vídeo tales como estimación y predicción de vector de movimiento en datos de vídeo. El motor de estimación de movimiento determina los vectores de movimiento que describen la transformación de los datos de imagen entre fotogramas de vídeo sucesivos. En algunas realizaciones, un códec de medios de procesador gráfico usa el motor de estimación de movimiento de vídeo 1934 para realizar operaciones en vídeo a nivel de macrobloque que, de lo contrario, podrían ser demasiado intensivas a nivel de

computación para realizarlas con un procesador de propósito general. En algunas realizaciones, el motor de estimación de movimiento 1934 está generalmente disponible para los componentes de procesador gráfico para ayudar con las funciones de procesamiento y decodificación de vídeo que son sensibles o adaptables a la dirección o magnitud del movimiento dentro de los datos de vídeo.

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

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

Unidades de ejecución

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

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

En algunas realizaciones, el procesador gráfico 2000 recibe lotes de comandos a través de la interconexión de anillo 2002. Los comandos entrantes son interpretados por un transmisor por flujo continuo de comandos 2003 en el extremo frontal de canalización 2004. En algunas realizaciones, el procesador gráfico 2000 incluye una lógica de ejecución escalable para realizar procesamiento de geometría 3D y procesamiento de medios a través de los núcleos de gráficos 2080A-2080N. Para los comandos de procesamiento de geometría 3D, el transmisor por flujo continuo de comandos 2003 suministra comandos a la canalización de geometría 2036. Para al menos algunos comandos de procesamiento de medios, el transmisor por flujo continuo de comandos 2003 suministra los comandos al extremo frontal de vídeo 2034, que se acopla a un motor de medios 2037. En algunas realizaciones, el motor de medios 2037 incluye un motor de calidad de vídeo (VQE) 2030 para posprocesamiento de vídeo e imágenes y un motor de codificación/decodificación multiformato (MFX) 2033 para proporcionar codificación y decodificación de datos de medios acelerados por hardware. En algunas realizaciones, tanto la canalización de geometría 2036 como el motor de medios 2037 generan hilos de ejecución para los recursos de ejecución de hilos proporcionados por al menos un núcleo de gráficos 2080A.

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

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

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

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

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

El conjunto de instrucciones de la unidad de ejecución incluye instrucciones de datos múltiples de una sola instrucción (SIMD). Los diversos elementos de datos se pueden almacenar como un tipo de datos empaquetado en un registro y la unidad de ejecución procesará los diversos elementos en función del 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 sobre el vector como cuatro elementos de datos empaquetados separados de 64 bits (elementos de datos de tamaño de palabra cuádruple (QW)), ocho elementos de datos empaquetados separados de 32 bits (elementos de datos de tamaño de palabra doble (DW)), dieciséis elementos de datos empaquetados separados de 16 bits (elementos de datos de tamaño de palabra (W)), o treinta y dos elementos de datos separados de 8 bits (elementos de datos de tamaño byte (B)). Sin embargo, se permiten diferentes anchos de vector y tamaños de registros.

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

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

Una vez que un grupo de objetos geométricos ha sido procesado y rasterizado en datos de píxeles, se invoca el sombreador de píxeles 2102 para calcular además la información de salida y hacer que los resultados se escriban en las superficies de salida (p. ej., memorias intermedias de color, memorias intermedias de profundidad, memorias intermedias de estarcido, etc.). En algunas realizaciones, el sombreador de píxeles 2102 calcula los valores de los

diversos atributos de vértice que han de interpolarse a través del objeto rasterizado. En algunas realizaciones, el sombreador de píxeles 2102 a continuación ejecuta un programa de sombreador de píxeles suministrado por la interfaz de programación de aplicaciones (API). Para ejecutar el programa de sombreador de píxeles, el sombreador de píxeles 2102 despacha hilos a una unidad de ejecución (p. ej., 2108A) mediante el despachador de hilos 2104. En algunas realizaciones, el sombreador de píxeles 2102 usa la lógica de muestreo de textura en el muestreador 2110 para acceder a datos de textura en mapas de textura almacenados en memoria. Las operaciones aritméticas en los datos de textura y los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico, o descartan uno o más píxeles del procesamiento posterior.

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

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

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

Para cada formato, el código de operación de instrucción 2212 define la operación que debe 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 ejecuta cada instrucción a través de todos los canales de datos de los operandos. En algunas realizaciones, el campo de control de instrucciones 2214 permite el control sobre ciertas opciones de ejecución, tales como la selección de canales (p. ej., predicación) y el orden de los canales de datos (p. ej., mezcla). Para las instrucciones de 128 bits 2210, un campo de tamaño de ejecución 2216 limita el número de canales de datos que se ejecutarán en paralelo. En algunas realizaciones, el campo de tamaño de ejecución 2216 no está disponible para su uso en el formato de instrucción compacto de 64 bits 2230.

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

En algunas realizaciones, el formato de instrucción de 128 bits 2210 incluye una información de modo de acceso/dirección 2226 que especifica, por ejemplo, si se usa el modo de direccionamiento de registro directo o el modo de direccionamiento de registro indirecto. Cuando se usa el modo de direccionamiento de registro directo, la dirección de registro de uno o más operandos viene proporcionada 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 es para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones soportan modos de acceso que incluyen un modo de acceso alineado de 16 bytes y un modo de acceso alineado de 1 byte, donde la alineación de bytes del modo de acceso determina la alineación de acceso de los operandos de instrucción. Por ejemplo, cuando está en un primer modo, la instrucción 2210 puede usar un direccionamiento alineado en bytes para operandos de origen y destino y, cuando está en un segundo modo, la instrucción 2210 puede usar direccionamiento alineado de 16 bytes para todos los operandos de origen y destino.

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

En algunas realizaciones, las instrucciones se agrupan en función de los campos de bits del código de operación 2212 para simplificar la decodificación del código de operación 2240. Para un código de operación de 8 bits, los bits 4, 5 y 6 permiten que la unidad de ejecución determine el tipo de código de operación. La agrupación precisa de códigos de operación que se muestra es un mero ejemplo. En algunas realizaciones, un grupo de códigos de operaciones lógicas y de movimiento 2242 incluye instrucciones lógicas y de movimiento de datos (p. ej., mover (mov), comparar (cmp)). En algunas realizaciones, el grupo lógico y de movimiento 2242 comparte los cinco bits más significativos (MSB), donde las instrucciones de movimiento (mov) tienen la forma 0000xxxxb y las instrucciones lógicas tienen la forma 0001xxxxb. Un grupo de instrucciones de control de flujo 2244 (p. ej., llamada, salto (jmp)) incluye instrucciones en forma de 0010xxxxb (p. ej., 0x20). Un grupo de instrucciones misceláneas 2246 incluye una mezcla de instrucciones, incluyendo instrucciones de sincronización (p. ej., esperar, enviar) en forma de 0011xxxxb (p. ej., 0x30). Un grupo de instrucciones matemáticas paralelas 2248 incluye instrucciones aritméticas por componentes (p. ej., sumar, multiplicar (mul)) en forma de 0100xxxxb (p. ej., 0x40). El grupo matemático paralelo 2248 realiza operaciones aritméticas en paralelo a través de canales de datos. El grupo matemático vectorial 2250 incluye instrucciones aritméticas (p. ej., dp4) en forma de 0101xxxxb (p. ej., 0x50). El grupo de matemáticas vectoriales realiza la aritmética, tal como cálculos de productos escalares en operandos vectoriales.

Canalización de gráficos

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

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

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

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

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

En algunas realizaciones, objetos geométricos completos pueden ser procesados por un sombreador de geometría 2319 a través de uno o más hilos despachados a unidades de ejecución 2352A, 2352B o pueden proseguir directamente al recortador 2329. En algunas realizaciones, el sombreador de geometría opera en objetos geométricos

enteros, en lugar de en vértices o parches de vértices como en etapas anteriores de la canalización de gráficos. Si la teselación está inhabilitada, el sombreador de geometría 2319 recibe entrada del sombreador de vértices 2307. En algunas realizaciones, el sombreador de geometría 2319 se puede programar mediante un programa de sombreado de geometría para realizar la teselación de geometría si las unidades de teselación están inhabilitadas.

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 sombreador de geometría. En algunas realizaciones, un rasterizador 2373 (p. ej., el componente de prueba de profundidad) en la canalización de salida de representación 2370 despacha sombreadores de píxeles para convertir los objetos geométricos en sus representaciones por píxeles. En algunas realizaciones, la lógica de sombreador de píxeles está incluida en la lógica de ejecución de hilos 2350. En algunas realizaciones, una aplicación puede omitir el rasterizador 2373 y acceder a datos de vértices no rasterizados mediante una unidad de salida de flujo 2323.

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

En algunas realizaciones, la canalización de salida de representación 2370 contiene un rasterizador 2373 que convierte objetos basados en vértices en una representación asociada basada en píxeles. En algunas realizaciones, la lógica del rasterizador incluye una unidad generadora de ventanas/enmascaradora para realizar una rasterización de triángulo y de línea de función fija. En algunas realizaciones también están disponibles una memoria caché de representación 2378 y una memoria caché de profundidad 2379 asociadas. Un componente de operaciones de píxeles 2377 realiza operaciones basadas en píxeles en los datos, aunque en algunos casos, las operaciones de píxeles asociadas a operaciones 2D (p. ej., transferencias de imágenes en bloques de bits con mezcla) son realizadas por el motor 2D 2341 o sustituidas en tiempo de visualización por el controlador de visualización 2343 usando planos de visualización superpuestos. En algunas realizaciones, una memoria caché L3 compartida 2375 está disponible para todos los componentes gráficos, lo que permite compartir datos sin usar la memoria del sistema principal.

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

En algunas realizaciones, el procesador gráfico 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 gráfico a través de la interconexión de anillo 2302, o algún otro bus o estructura de interconexión. En algunas realizaciones, el motor de visualización 2340 incluye un motor 2D 2341 y un controlador de visualización 2343. En algunas realizaciones, el motor de visualización 2340 contiene una lógica de propósito especial capaz de operar independientemente de la canalización 3D. En algunas realizaciones, el controlador de visualización 2343 se acopla con un dispositivo de visualización (no mostrado), que puede ser un dispositivo de visualización integrado en el sistema, como en un ordenador portátil, o un dispositivo de visualización externo conectado a través de un conector de dispositivo de visualización.

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

Programación de canalización de gráficos

La Figura 24A es un diagrama de bloques que ilustra un formato de comandos de procesador gráfico 2400 de acuerdo con algunas realizaciones. La Figura 24B es un diagrama de bloques que ilustra una secuencia de comandos de procesador gráfico 2410 de acuerdo con una realización. Los recuadros en líneas continuas de la Figura 24A ilustran

los componentes que están incluidos por lo general en un comando de gráficos, mientras que las líneas discontinuas incluyen componentes que son opcionales o que están incluidos únicamente en un subconjunto de los comandos de gráficos. El formato de comandos de procesador gráfico 2400 ilustrativo de la Figura 24A incluye campos de datos para identificar un cliente objetivo 2402 del comando, un código de operación del comando (cód. de ope.) 2404 y los datos relevantes 2406 para el comando. También se incluye un subcódigo de operación 2405 y un tamaño de comando 2408 en algunos comandos.

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

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

En algunas realizaciones, la secuencia de comandos de procesador gráfico 2410 puede comenzar con un comando de vaciado de canalización 2412 para hacer que cualquier canalización de gráficos activa complete los comandos actualmente pendientes para la canalización. En algunas realizaciones, la canalización 3D 2422 y la canalización de medios 2424 no funcionan simultáneamente. Se realiza el vaciado de la canalización para hacer que la canalización de gráficos activa complete cualquier comando pendiente. En respuesta a un vaciado de canalización, el analizador de comandos del procesador gráfico pausará el procesamiento de comandos hasta que los motores de dibujo activos completen las operaciones pendientes y se invaliden las memorias caché de lectura relevantes. Opcionalmente, cualquier dato en la memoria caché de representación gráfica que esté marcado como "sucio" se puede volcar a la memoria. En algunas realizaciones, el comando de vaciado de canalización 2412 se puede usar para la sincronización de canalización o antes de colocar el procesador gráfico en un estado de bajo consumo de energía.

En algunas realizaciones, se usa un comando de selección de canalización 2413 cuando una secuencia de comandos requiere que el procesador gráfico conmute explícitamente entre canalizaciones. En algunas realizaciones, se requiere un comando de selección de canalización 2413 solo una vez dentro de un contexto de ejecución antes de emitir comandos de canalización, a menos que el contexto sea para emitir comandos para ambas canalizaciones. En algunas realizaciones, se requiere un comando de vaciado de canalización 2412 inmediatamente antes de un conmutador de canalización mediante el comando de selección de canalización 2413.

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

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

Los comandos restantes de la secuencia de comandos difieren en función de la canalización activa para las operaciones. Basándose en una determinación de la canalización 2420, la secuencia de comandos se adapta a la

canalización 3D 2422 que comienza con el estado de canalización 3D 2430, o a la canalización de medios 2424 que comienza en el estado de canalización de medios 2440.

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

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

En algunas realizaciones, la canalización 3D 2422 se desencadena a través de un comando o evento de ejecución 2434. En algunas realizaciones, una escritura de registro desencadena la ejecución de un comando. En algunas realizaciones, la ejecución se desencadena mediante un comando "ir" o "patear" en la secuencia de comandos. En una realización la ejecución del comando se desencadena usando un comando de sincronización de canalización para vaciar la secuencia de comandos a través de la canalización de gráficos. La canalización 3D realizará un procesamiento de geometría para las primitivas 3D. Una vez completadas las operaciones, los objetos geométricos resultantes se rasterizan y el motor de píxeles coloreará los píxeles resultantes. También se pueden incluir comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo trasero de píxeles para esas operaciones.

En algunas realizaciones, la secuencia de órdenes del procesador gráfico 2410 sigue la ruta de la canalización de medios 2424 cuando se llevan a cabo operaciones de medios. En general, el uso específico y la forma de programación de la canalización de medios 2424 depende de las operaciones de medios o de cálculo a realizar. Es posible que se descarguen operaciones de decodificación de medios específicas a la canalización de medios durante la decodificación de medios. En algunas realizaciones, la canalización de medios también se puede omitir y la decodificación de medios se puede realizar en su totalidad o en parte usando recursos proporcionados por uno o más núcleos de procesamiento de propósito general. En una realización, la canalización de medios también incluye elementos para operaciones de unidad de procesador gráfico de propósito general (GPGPU), donde el procesador gráfico se usa para realizar operaciones vectoriales SIMD usando programas de sombreado computacional que no están explícitamente relacionados con la representación de primitivas de gráficos.

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

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

Arquitectura de software de gráficos

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

aplicación de gráficos 2510 y el sistema operativo 2520 se ejecutan cada uno en la memoria del sistema 2550 del sistema de procesamiento de datos.

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

En algunas realizaciones, el sistema operativo 2520 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo tipo UNIX propietario o un sistema operativo tipo UNIX de código abierto que 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 de sombreador 2512 en HLSL en un lenguaje de sombreador de nivel inferior. La compilación puede ser una compilación justo a tiempo (JIT) o la aplicación puede realizar una precompilación de sombreadores. En algunas realizaciones, los sombreadores de alto nivel se compilan en sombreadores de bajo nivel durante la compilación de la aplicación de gráficos 3D 2510.

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

Implementaciones de núcleos IP

Uno o más aspectos de al menos una realización se pueden implementar mediante un código representativo almacenado en un medio legible por máquina que representa y/o define la lógica dentro de un circuito integrado tal como un procesador. Por ejemplo, el medio legible por máquina puede incluir instrucciones que representan una lógica diversa dentro del procesador. Cuando las lee una máquina, las instrucciones pueden hacer que la máquina fabrique la lógica para realizar las técnicas descritas en este documento. Estas representaciones, conocidas como "núcleos IP", son unidades lógicas reutilizables para un circuito integrado que se pueden almacenar en un medio tangible y legible por máquina como un modelo de hardware que describe la estructura del circuito integrado. El modelo de hardware se puede suministrar a diversos clientes o instalaciones de fabricación, que cargan el modelo de hardware en máquinas de fabricación que fabrican el circuito integrado. El circuito integrado se puede fabricar de tal manera que el circuito realice 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 se puede usar para fabricar un circuito integrado para realizar las operaciones de acuerdo con una realización. El sistema de desarrollo de núcleo de IP 2600 se puede usar para generar diseños reutilizables modulares que se pueden incorporar en un diseño más grande o usar para construir un circuito integrado entero (p. ej., un circuito integrado de SOC). Una instalación de diseño 2630 puede generar una simulación de software 2610 de un diseño de núcleo de IP en un lenguaje de programación de alto nivel (p. ej., C/C++). La simulación de software 2610 puede usarse para diseñar, probar y verificar el comportamiento del núcleo de IP. A continuación, se puede crear o sintetizar un diseño de nivel de transferencia de registro (RTL) a partir del modelo de simulación 2600. El diseño de RTL 2615 es una abstracción del comportamiento del circuito integrado que modela el flujo de señales digitales entre registros de hardware, incluyendo la lógica asociada realizada usando las señales digitales modeladas. Además de un diseño de RTL 2615, también se pueden crear, diseñar o sintetizar diseños de nivel inferior al nivel de lógica o al nivel de transistores. Por tanto, los detalles particulares del diseño y simulación inicial pueden variar.

El diseño RTL 2615 o equivalente puede ser sintetizado adicionalmente por la instalación de diseño en un modelo de hardware 2620, que puede estar en un lenguaje de descripción de hardware (HDL) o alguna otra representación de datos de diseño físico. El HDL se puede simular o probar más para verificar el diseño del núcleo IP. El diseño de núcleo de IP se puede almacenar para suministrarse a una instalación de fabricación de terceros 2665 usando memoria no volátil 2640 (p. ej., disco duro, memoria flash o cualquier medio de almacenamiento no volátil). Como alternativa, el diseño del núcleo IP se puede transmitir (p. ej., a través de Internet) a través de una conexión por cable 2650 o una conexión inalámbrica 2660. La instalación de fabricación 2665 puede fabricar a continuación un circuito integrado que se base 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 circuito integrado de sistema en un chip 2700 ilustrativo que se puede fabricar usando uno o más núcleos de IP, de acuerdo con una realización. El circuito integrado ilustrativo incluye

uno o más procesadores de aplicación 2705 (p. ej., las CPU), al menos un procesador gráfico 2710, y puede incluir adicionalmente un procesador de imágenes 2715 y/o un procesador de vídeo 2720, cualquiera de los cuales puede ser un núcleo de IP modular desde las mismas o múltiples diferentes instalaciones de diseño. El circuito integrado incluye una lógica de periférico o de bus que incluye un controlador de USB 2725, un controlador de UART 2730, un controlador de SPI/SDIO 2735 y un controlador de I²S/I²C 2740. Adicionalmente, el circuito integrado puede incluir un dispositivo de visualización 2745 acoplado a uno o más de un controlador de interfaz multimedia de alta definición (HDMI) 2750 y una interfaz de visualización de interfaz de procesador de industria móvil (MIPI) 2755. El almacenamiento lo puede proporcionar un subsistema de memoria flash 2760 que incluye memoria flash y un controlador de memoria flash. La interfaz de memoria se puede proporcionar a través de un controlador de memoria 2765 para el acceso a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad embebido 2770.

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

Ventajosamente, cualquiera de los sistemas, procesadores, procesadores gráficos, aparatos y/o métodos anteriores se puede integrar o configurar con cualquiera de las diversas realizaciones descritas en el presente documento (p. ej., o partes de las mismas).

Las realizaciones son aplicables para su uso con todos los tipos de chips de circuitos integrados ("CI") de semiconductores. Ejemplos de estos chips CI incluyen, pero sin limitación, procesadores, controladores, componentes de conjuntos de chips, conjuntos lógicos programables (PLA), chips de memoria, chips de red, sistemas en chip (SoC), ASIC de controlador de SSD/NAND y similares. Además, en algunos de los dibujos, las líneas conductoras de señales se han representado con líneas. Algunas pueden ser diferentes, para indicar más rutas de señal constituyentes, tener una etiqueta numérica, para indicar un número de rutas de señal constituyentes y/o tener flechas en uno o más extremos, para indicar la dirección de flujo de información principal. Esto, sin embargo, no debe interpretarse de manera limitativa. Más bien, tal detalle añadido se puede usar en relación con una o más realizaciones ilustrativas para facilitar una comprensión más fácil de un circuito. Cualquier línea de señal representada, tenga o no información adicional, en realidad puede comprender una o más señales que pueden viajar en múltiples direcciones y puede implementarse con cualquier tipo adecuado de esquema de señal, p. ej., líneas digitales o analógicas implementadas con pares diferenciales, líneas de fibra óptica y/o líneas de un solo extremo.

Se pueden haber dado ejemplos de tamaños/modelos/valores/intervalos, aunque las realizaciones no se limitan a lo mismo. A medida que las técnicas de fabricación (p. ej., la fotolitografía) maduran con el tiempo, se espera que se puedan fabricar dispositivos de menor tamaño. Además, las bien conocidas conexiones de alimentación/masa a los chips de CI y otros componentes pueden mostrarse o no dentro de las figuras, para simplificar la ilustración y el análisis, y para no oscurecer ciertos aspectos de las realizaciones. Además, las disposiciones se pueden mostrar en forma de diagrama de bloques para evitar oscurecer las realizaciones, y también en vista del hecho de que los detalles específicos con respecto a la implementación de tales disposiciones de diagrama de bloques dependen en gran medida de la plataforma en la que se va a implementar la realización, es decir, tales detalles específicos deberían estar dentro del alcance de un experto en la materia. Cuando se establecen detalles específicos (p. ej., circuitos) para describir realizaciones ilustrativas, debería ser evidente para un experto en la materia que las realizaciones se pueden poner en práctica sin o con una variación de estos detalles específicos. Por lo tanto, la descripción debe considerarse ilustrativa en lugar de limitativa.

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

Tal y como se usa en la presente solicitud y en las reivindicaciones, una lista de elementos unidos por el término "uno o más de" puede referirse a cualquier combinación de los términos enumerados. Por ejemplo, las expresiones "uno o más de A, B o C" pueden significar A, B, C; A y B; A y C; B y C; o A, B y C.

Las realizaciones se han descrito anteriormente con referencia a realizaciones específicas. La invención está definida por las reivindicaciones adjuntas. Por consiguiente, la descripción y los dibujos anteriores deben considerarse en un sentido ilustrativo más que restrictivo.

REIVINDICACIONES

1. Un aparato de gráficos (700) adaptado para trabajar con un sistema gráfico que comprende un visualizador (1006) y una cámara, mostrando el visualizador una escena gráfica y capturando la cámara información ocular de un usuario que está viendo la escena gráfica, que comprende:
5 un detector de expresiones faciales (721) para detectar una expresión facial de un usuario (1008);
un motor de enfoque (606) para detectar un área de enfoque del usuario (1008);
un detector de luz (723) para detectar una cantidad de luz ambiental;
10 un sistema de representación foveada (1000) para proporcionar una región de resolución relativamente más alta (1002) y una región de resolución relativamente más baja (1004) del visualizador (1006) en función del área de enfoque del usuario (1008); y
un ajustador de parámetros (722) acoplado comunicativamente al detector de expresiones faciales (721), al motor de enfoque (606) y al detector de luz (723) para
ajustar un parámetro gráfico de la región de resolución relativamente más alta en función de la expresión facial
15 detectada del usuario (1008) y de la cantidad de luz ambiental detectada, en donde el parámetro gráfico incluye uno o más de un tamaño, un contraste y una tasa de sombreado de la región de resolución relativamente más alta.
2. El aparato (700) de la reivindicación 1, en donde la expresión facial detectada incluye uno o más de entrecerrado de ojos, parpadeo, guiño de un ojo y tensión de los músculos faciales del usuario (1008).
20
3. El aparato (700) de la reivindicación 2, en donde el parámetro gráfico además incluye un zum de la escena gráfica.
4. Un método para ajustar gráficos, que comprende:
25 detectar una expresión facial de un usuario (1008) que está viendo una escena gráfica mostrada por un visualizador, mientras se detecta luz ambiental;
detectar un área de enfoque del usuario (1008);
proporcionar una región de resolución relativamente más alta (1002) y una región de resolución relativamente más baja (1004) del visualizador (1006) en función del área de enfoque detectada del usuario (1008); y
ajustar un parámetro gráfico de la región de resolución relativamente más alta en función de la expresión facial
30 detectada del usuario (1008) y de la cantidad de luz ambiental detectada, en donde el parámetro gráfico incluye uno o más de un tamaño, un contraste y una tasa de sombreado de la región de resolución relativamente más alta.
5. El método de la reivindicación 4, en donde la expresión facial detectada incluye uno o más de entrecerrado de ojos, parpadeo, guiño de un ojo y tensión de los músculos faciales del usuario (1008).
35
6. El método de la reivindicación 5, en donde el parámetro gráfico además incluye un zum de la escena gráfica.

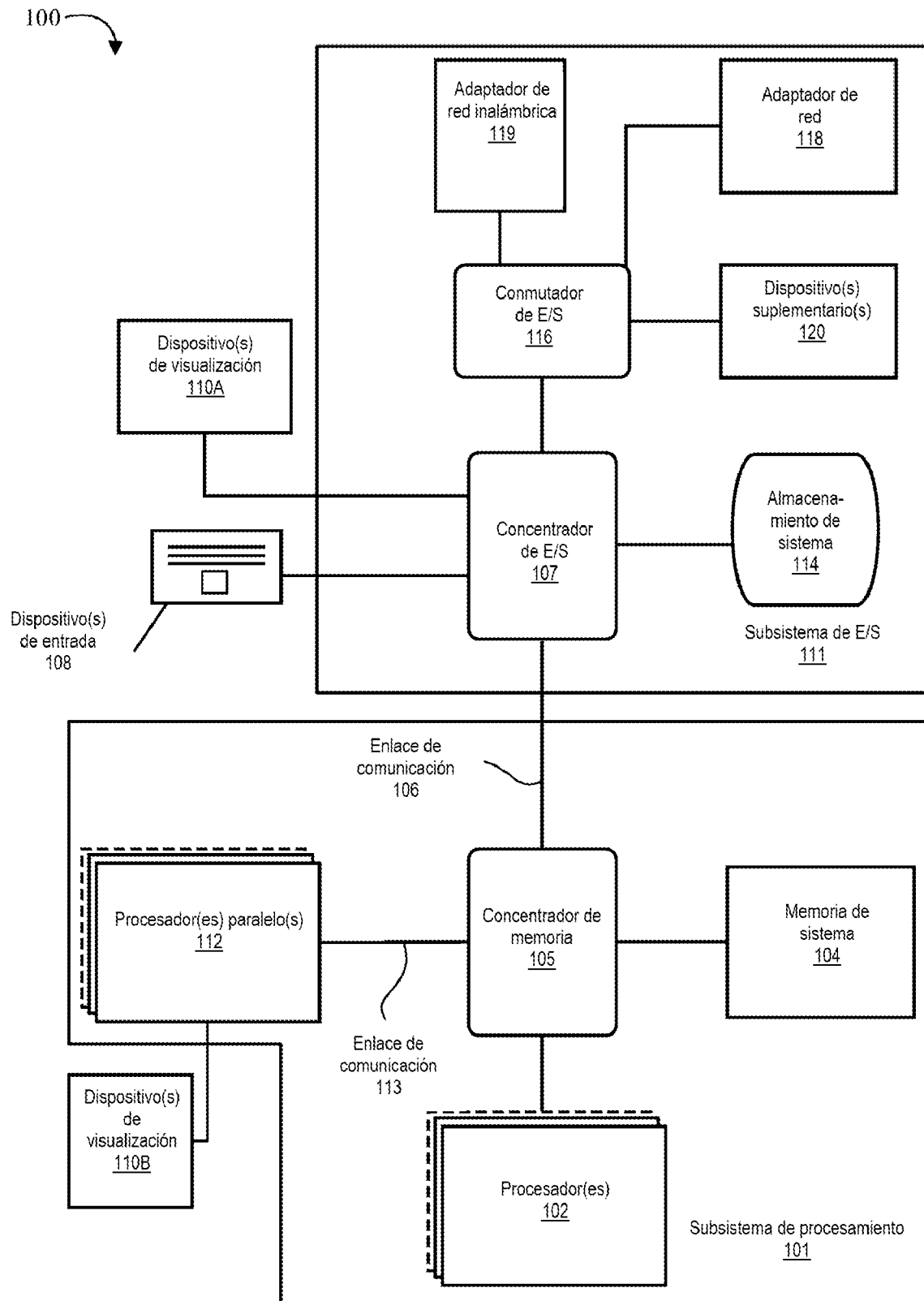


FIG. 1

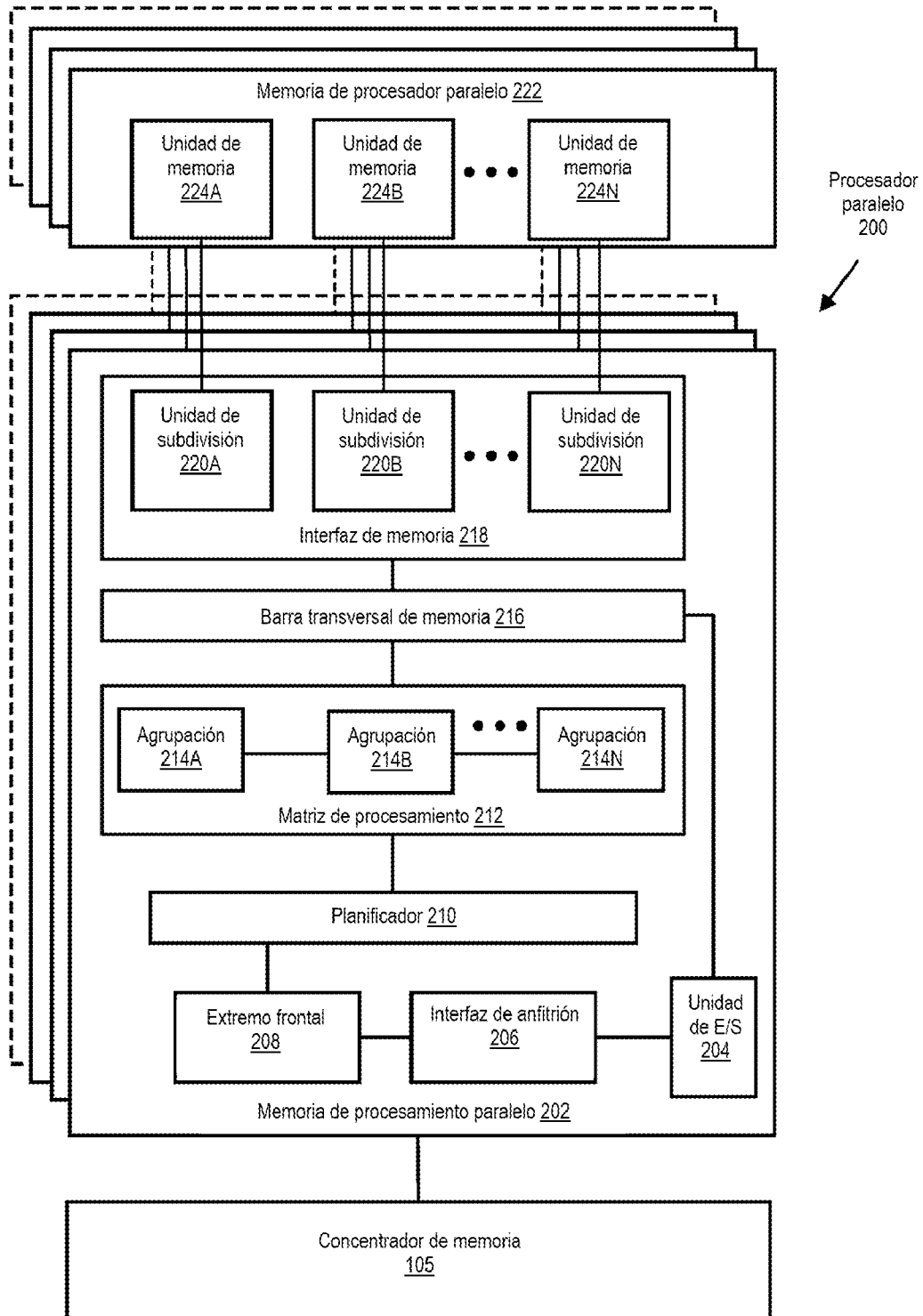


FIG. 2A

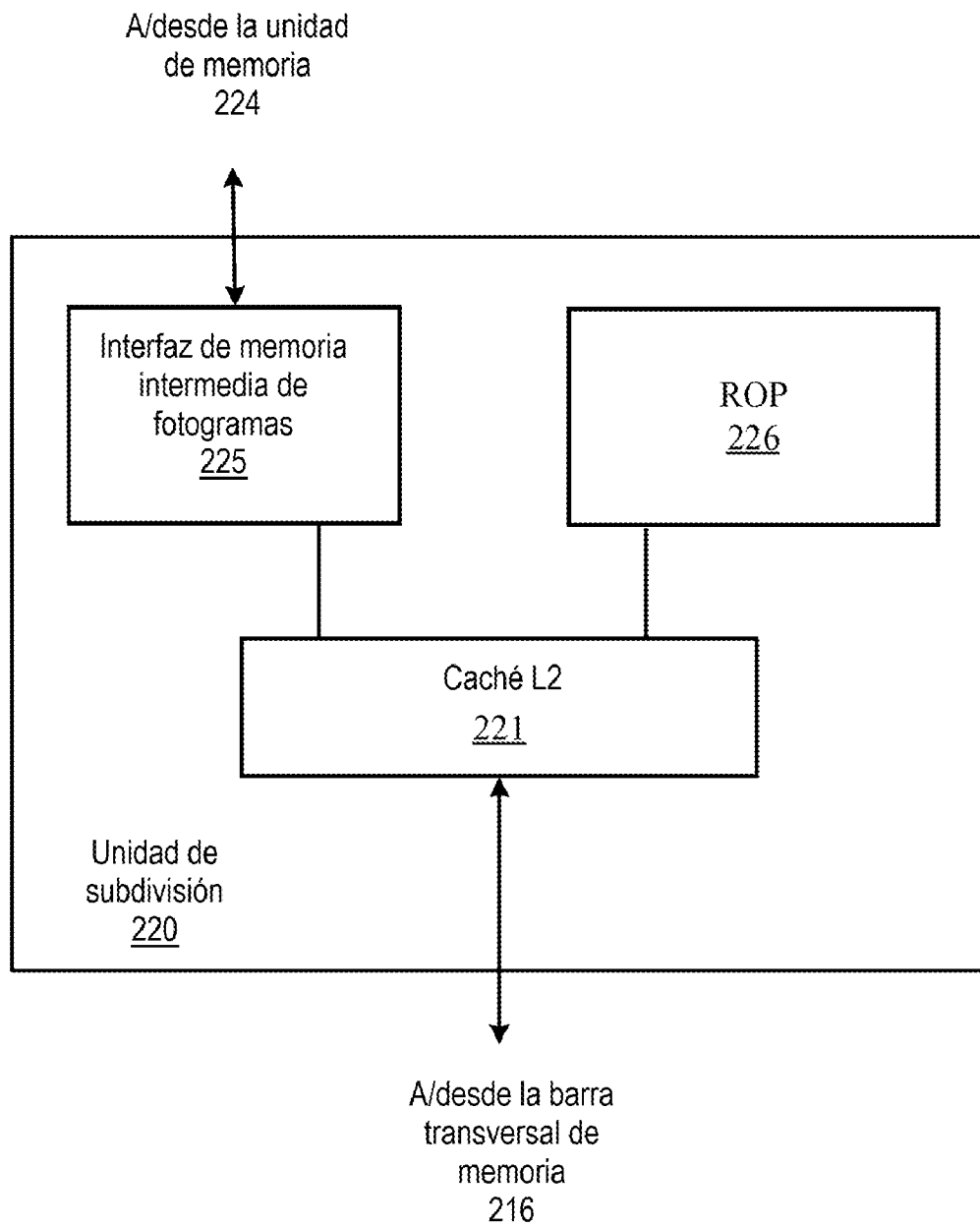


FIG. 2B

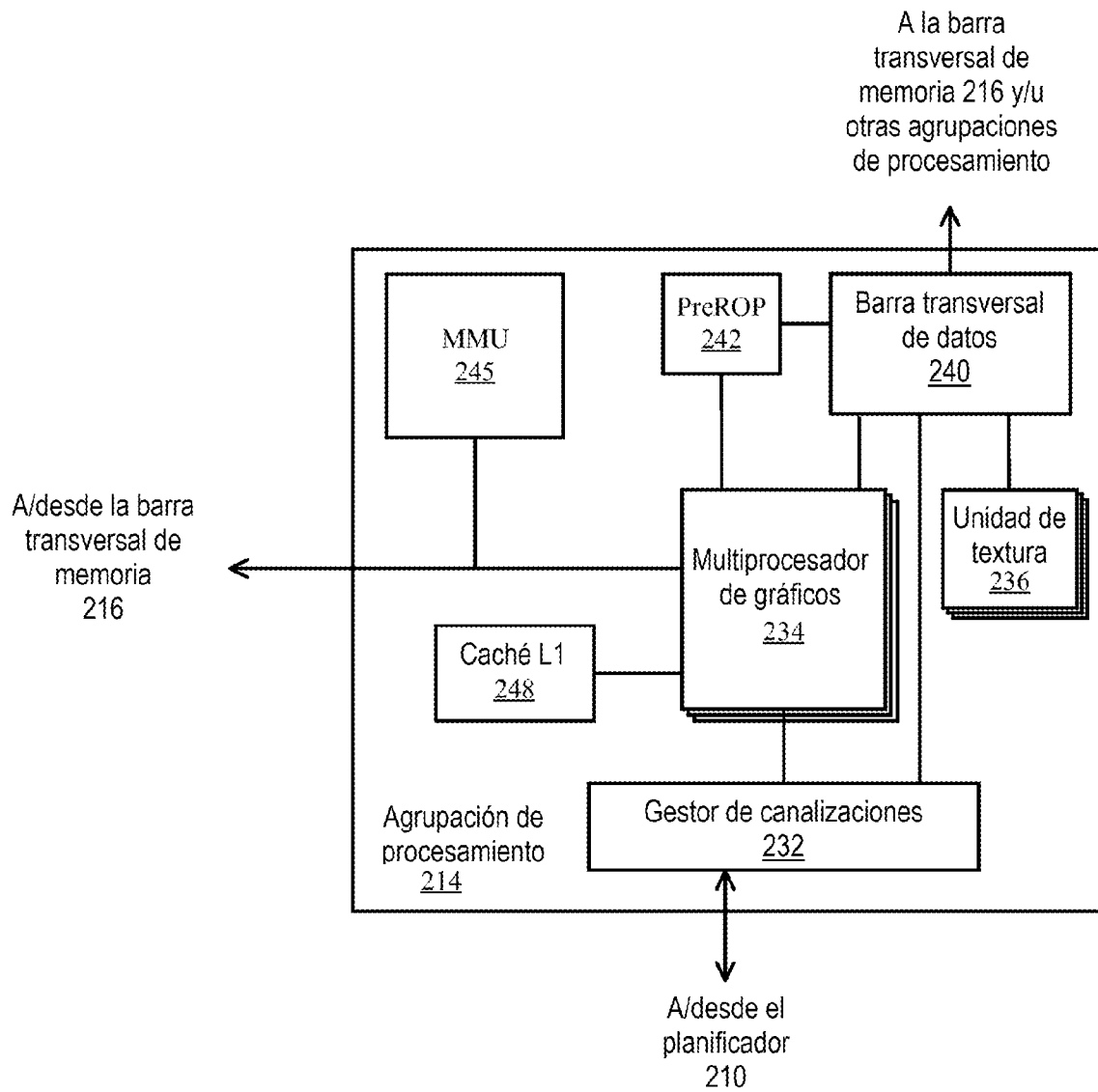
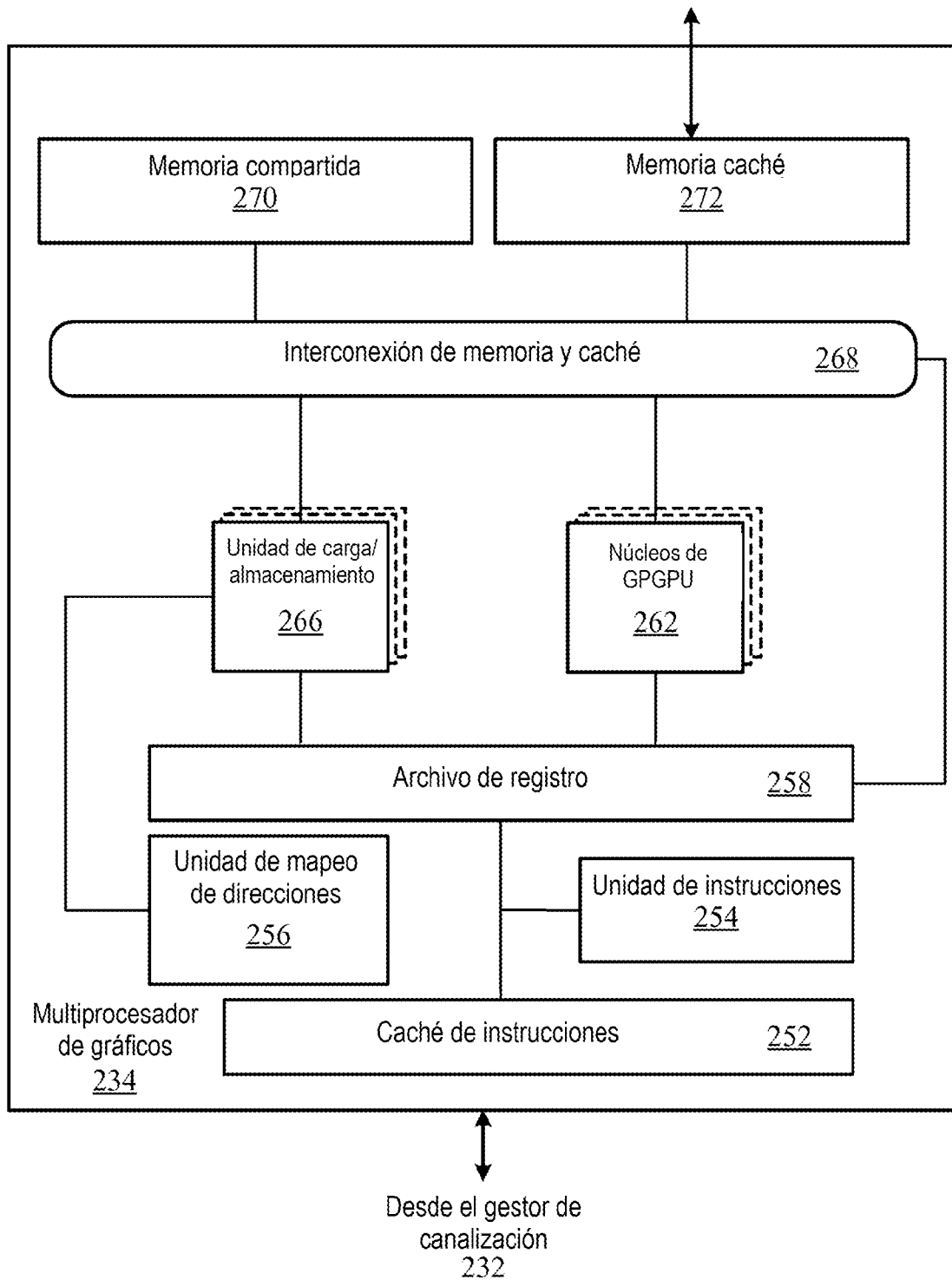


FIG. 2C

**FIG. 2D**

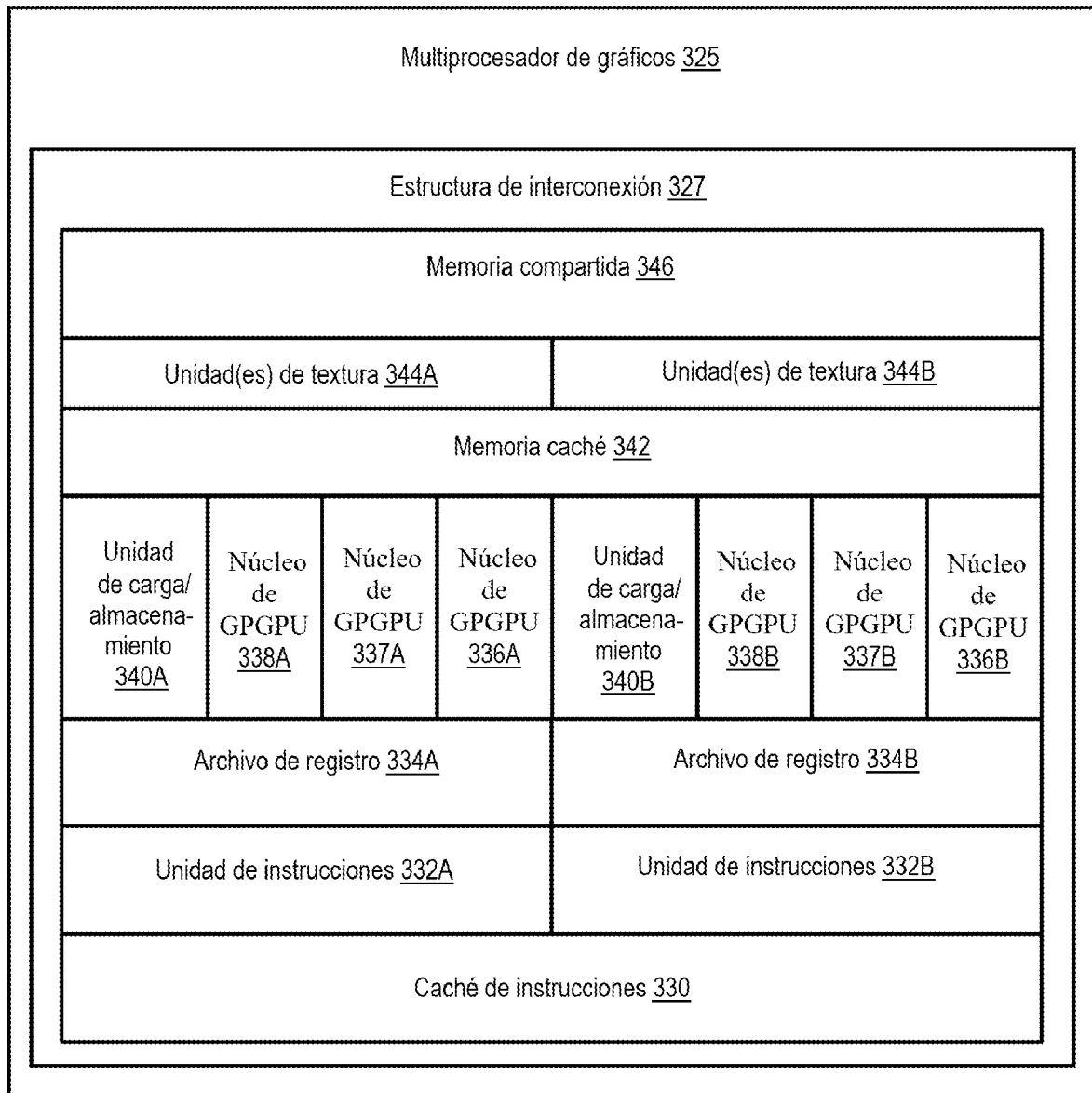


FIG. 3A

I

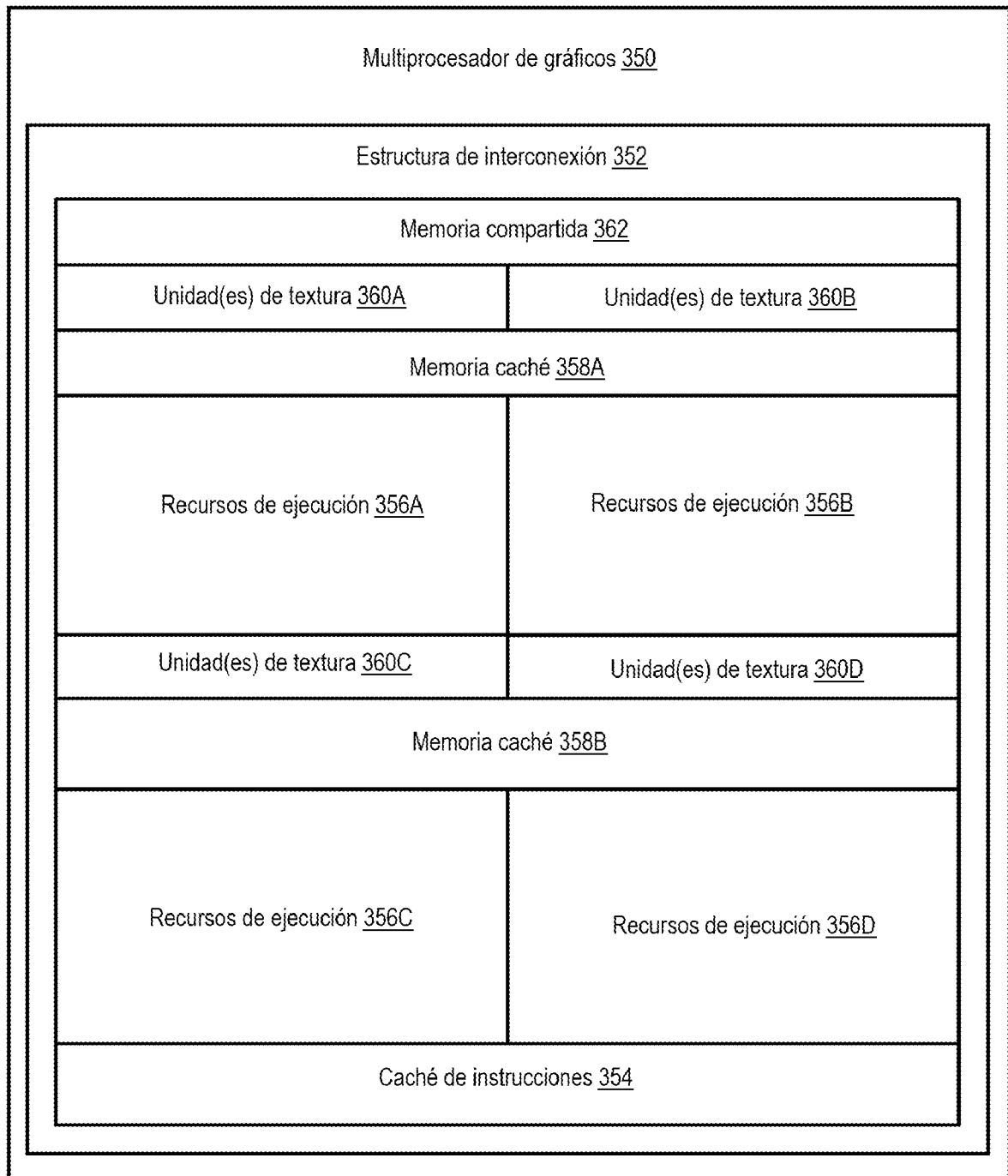


FIG. 3B

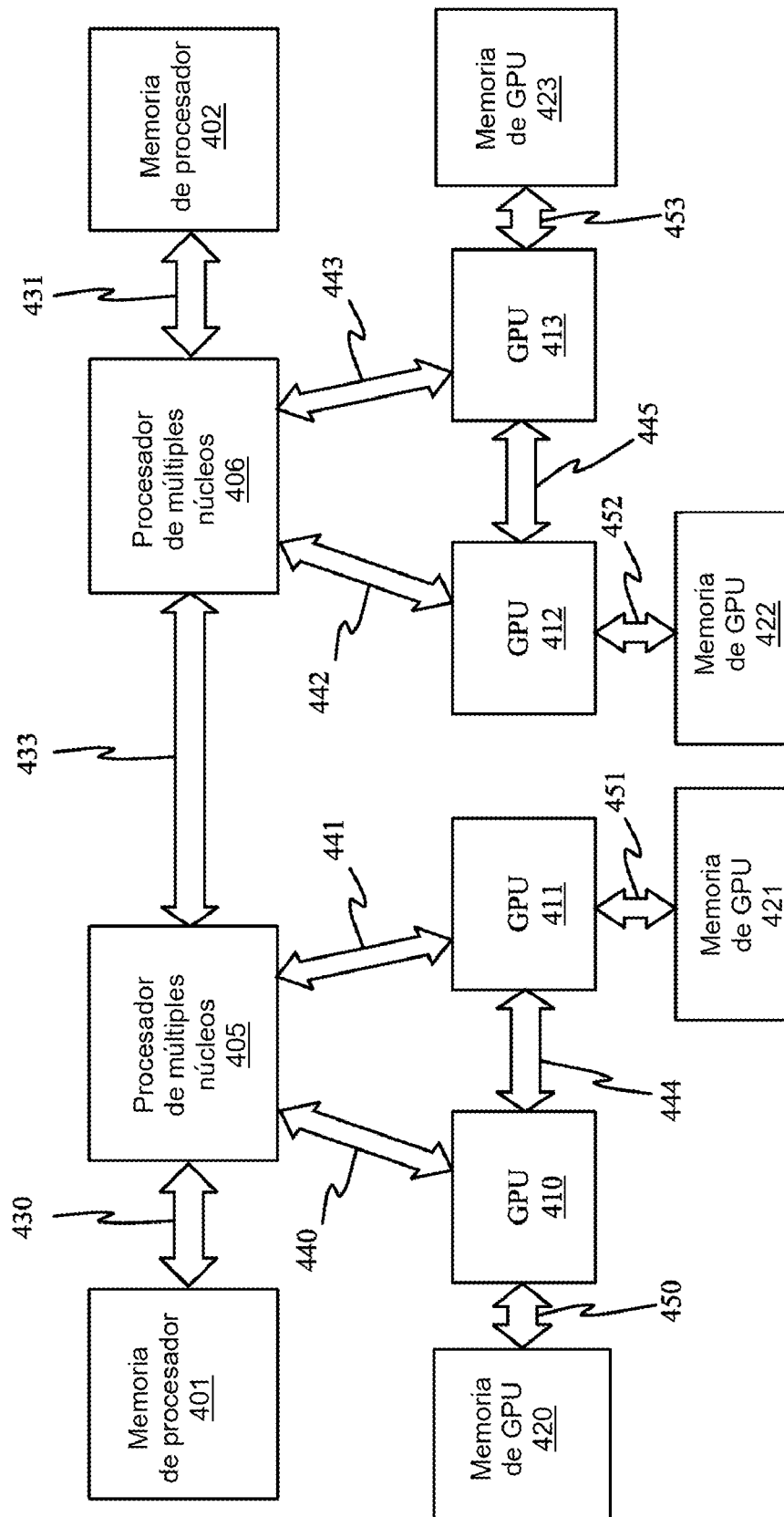


FIG. 4A

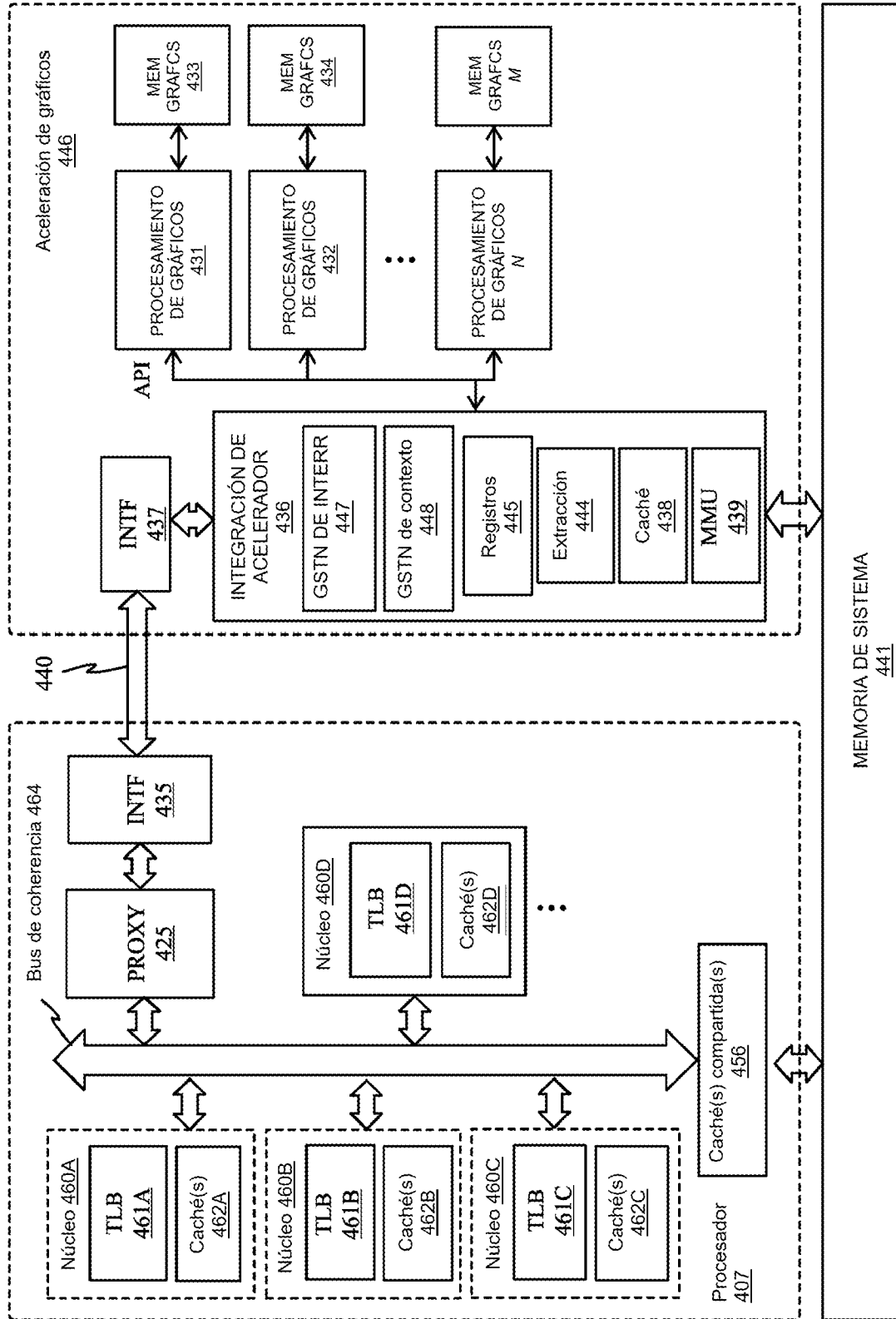
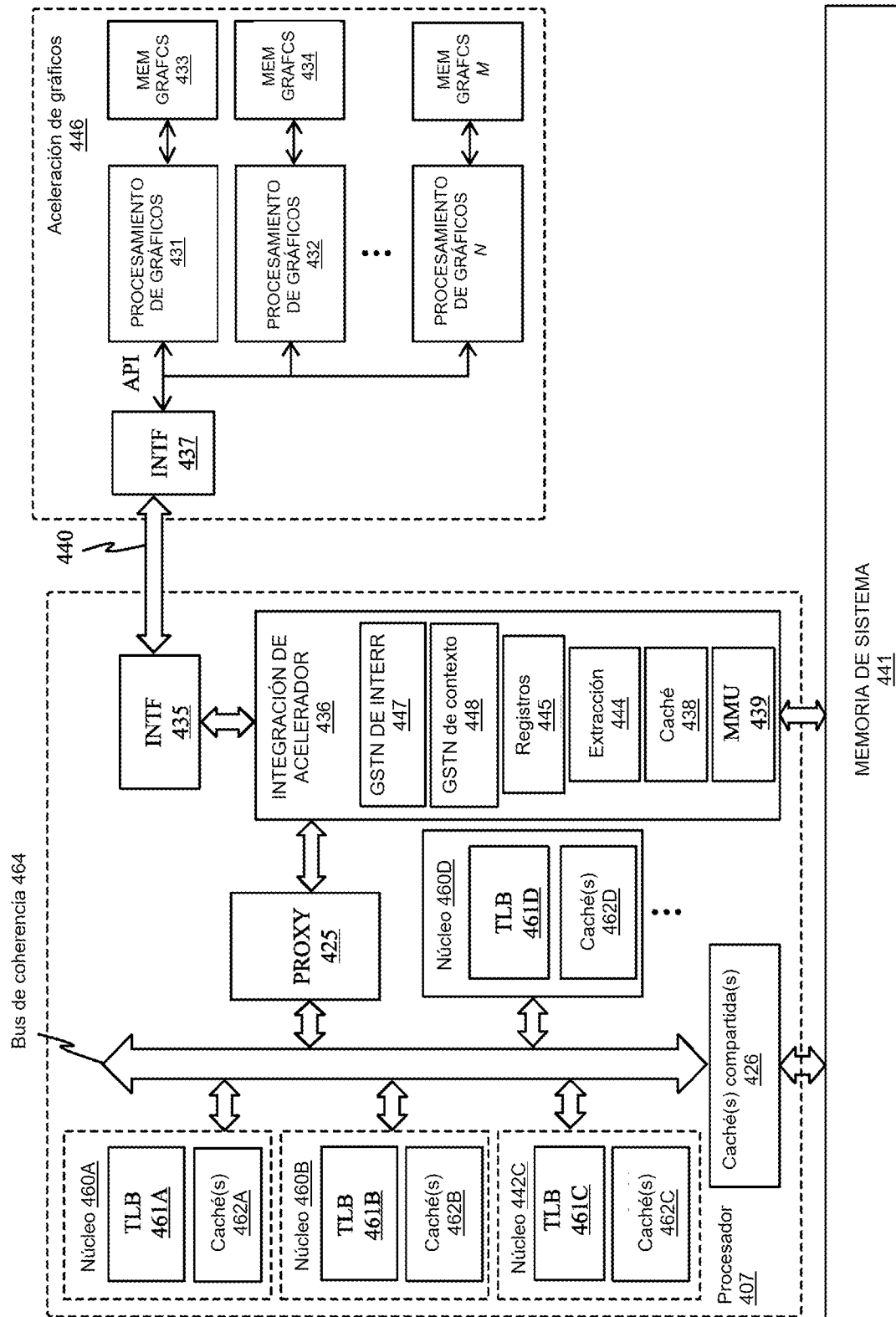


FIG. 4B

**FIG. 4C**

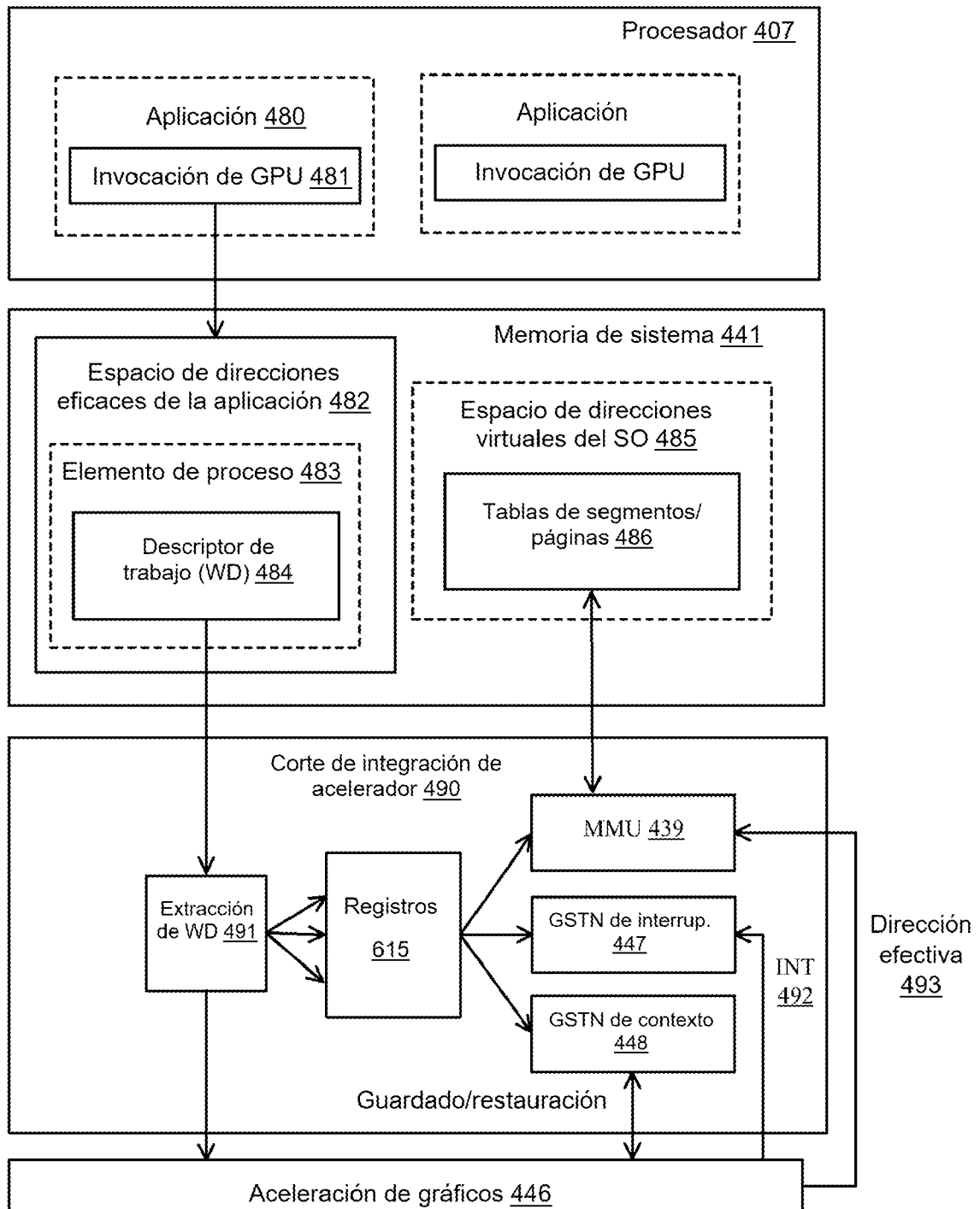


FIG. 4D

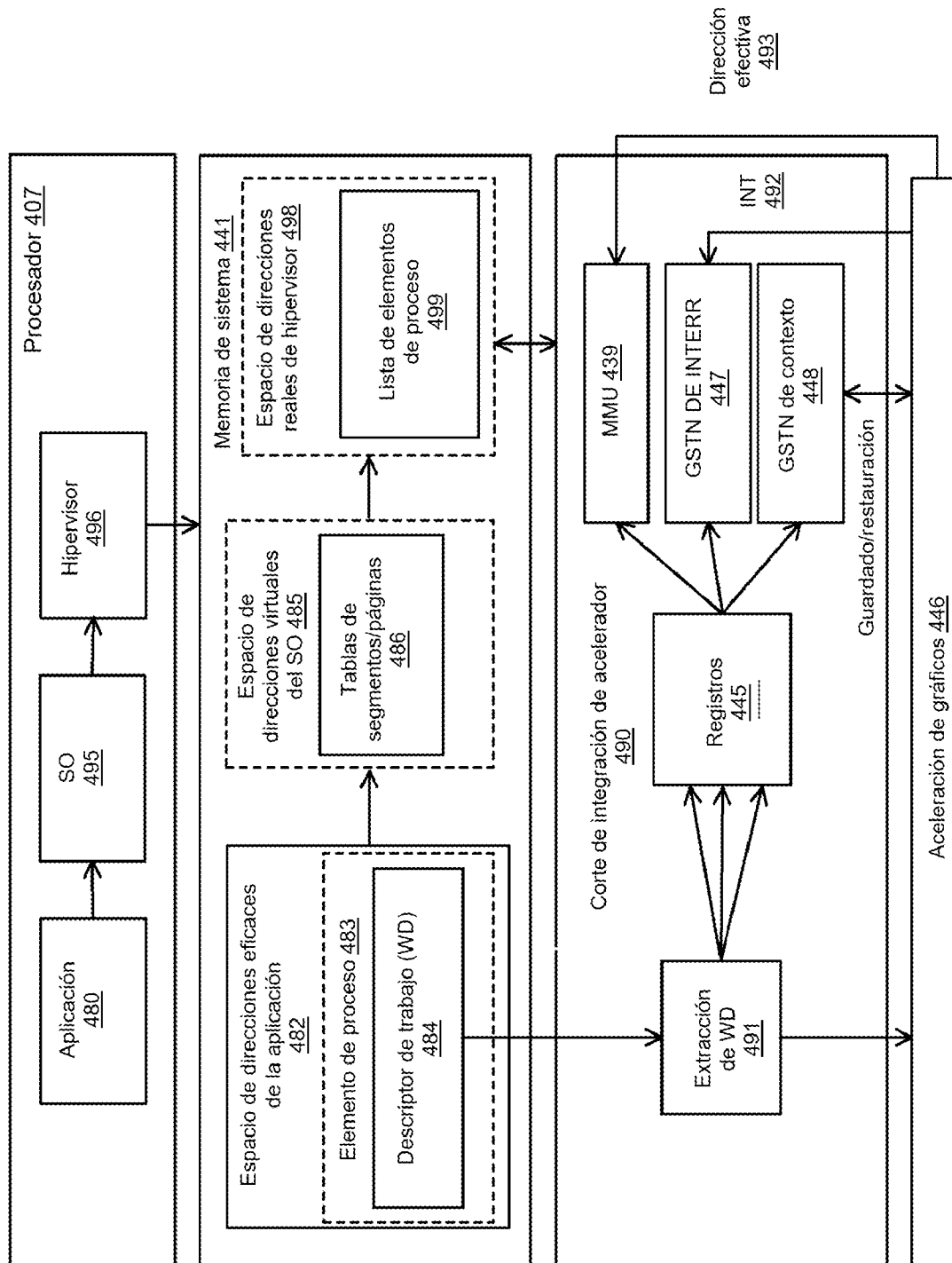


FIG. 4E

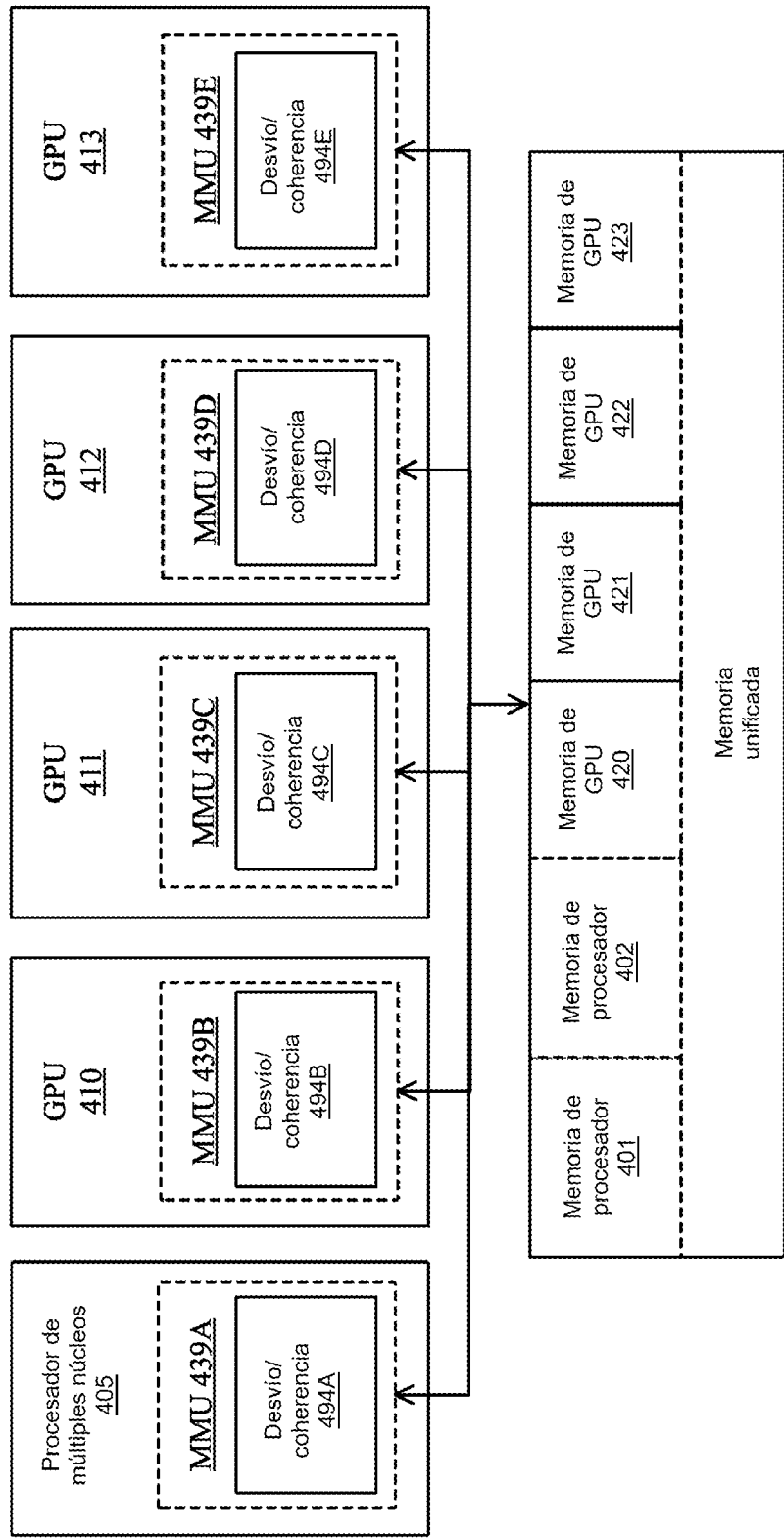


FIG. 4F

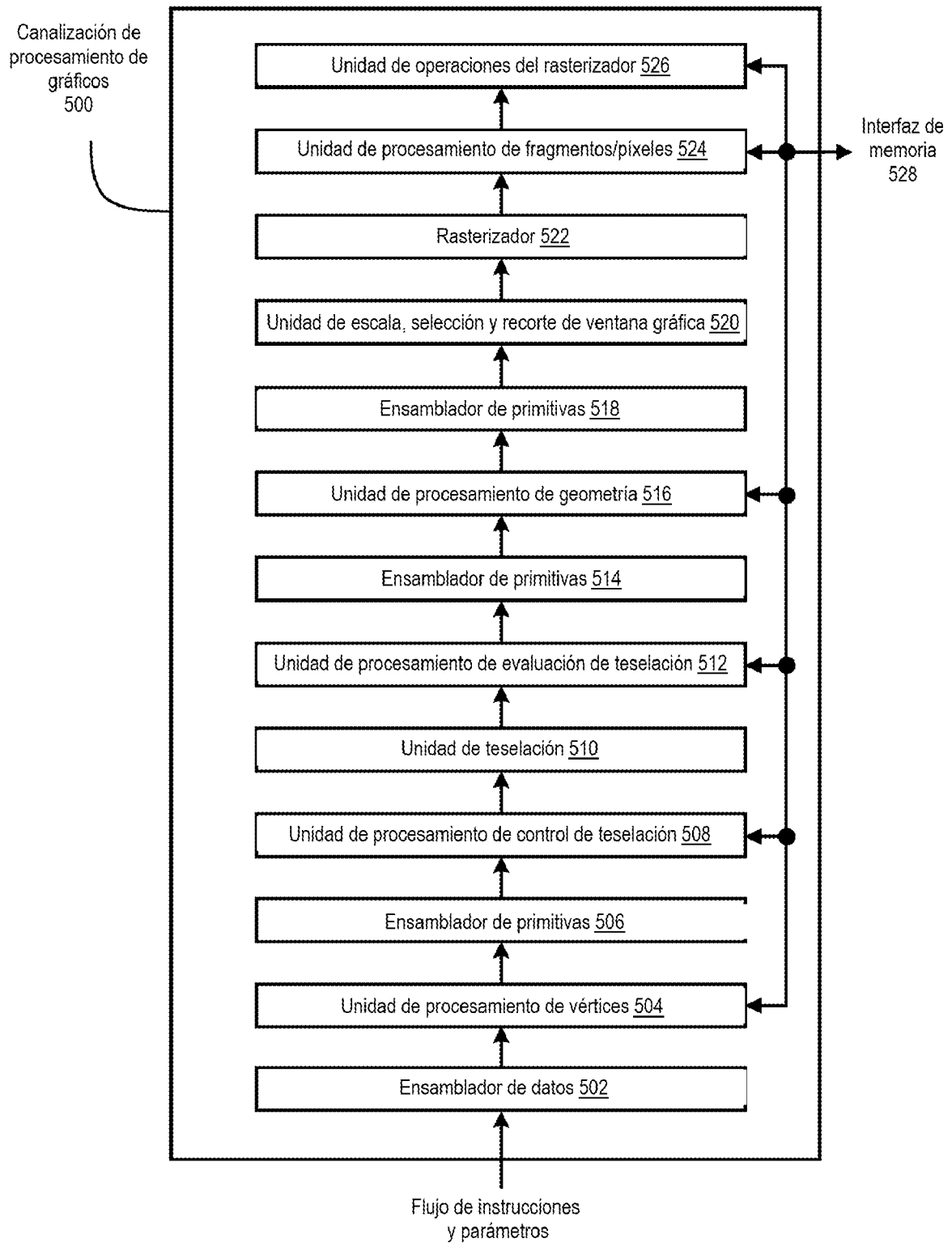


FIG. 5

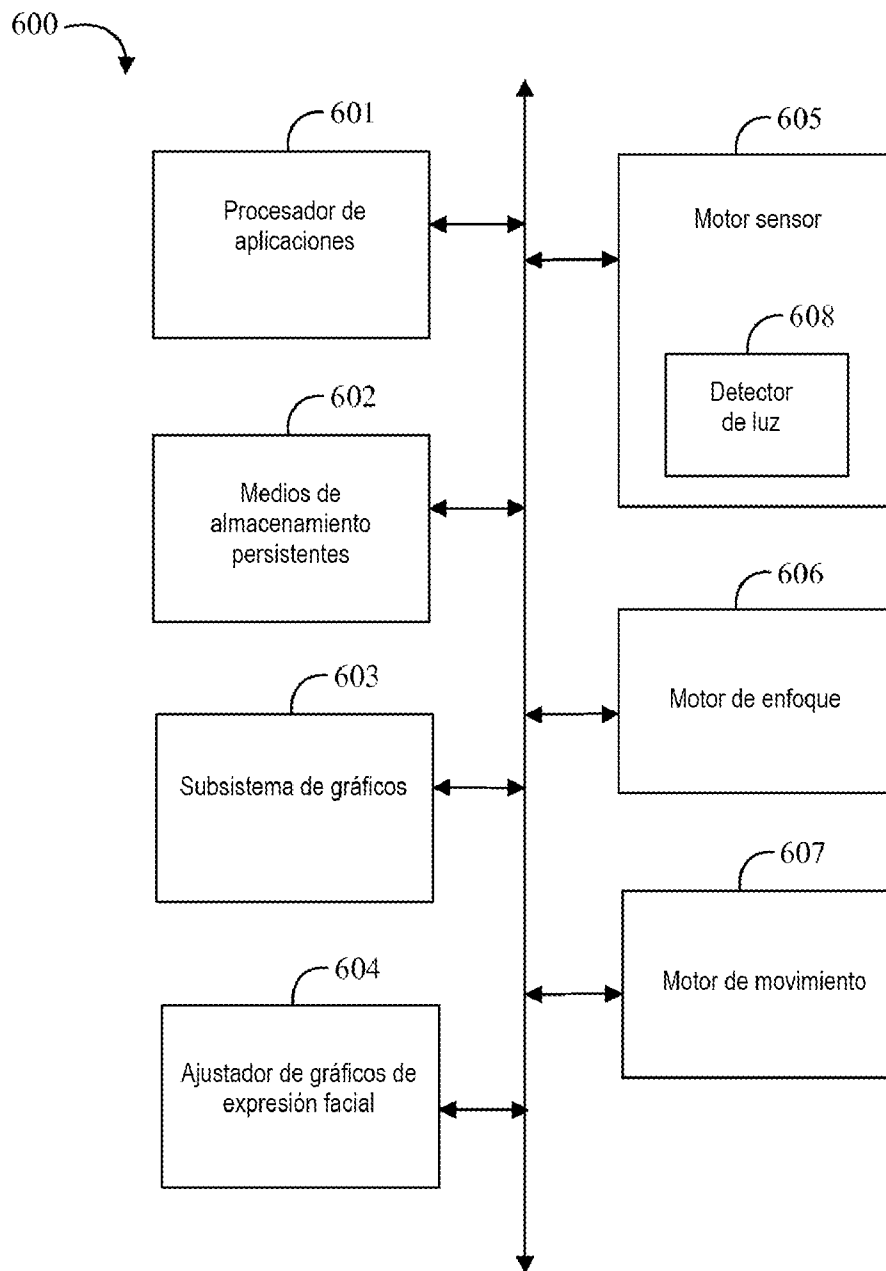


FIG. 6A

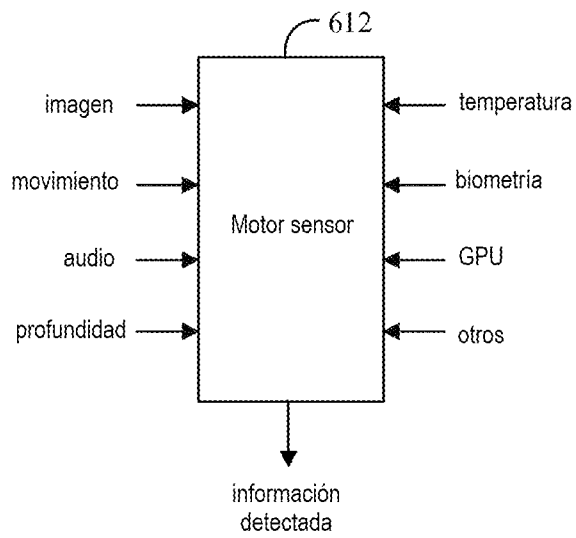


FIG. 6B

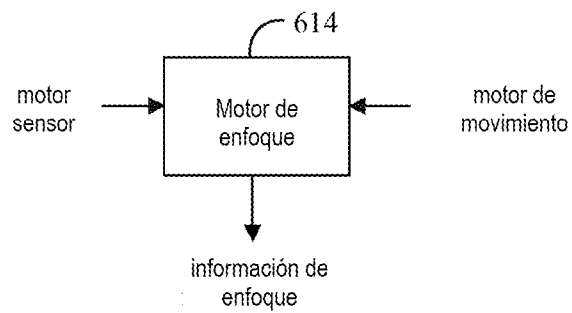


FIG. 6C

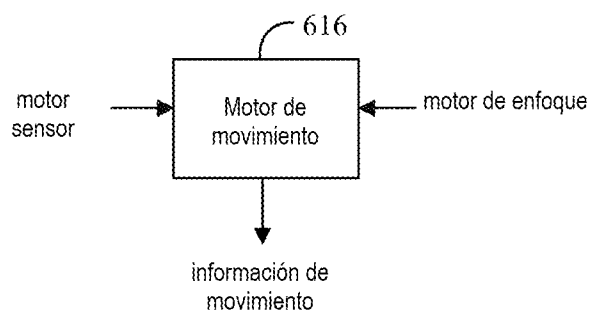


FIG. 6D

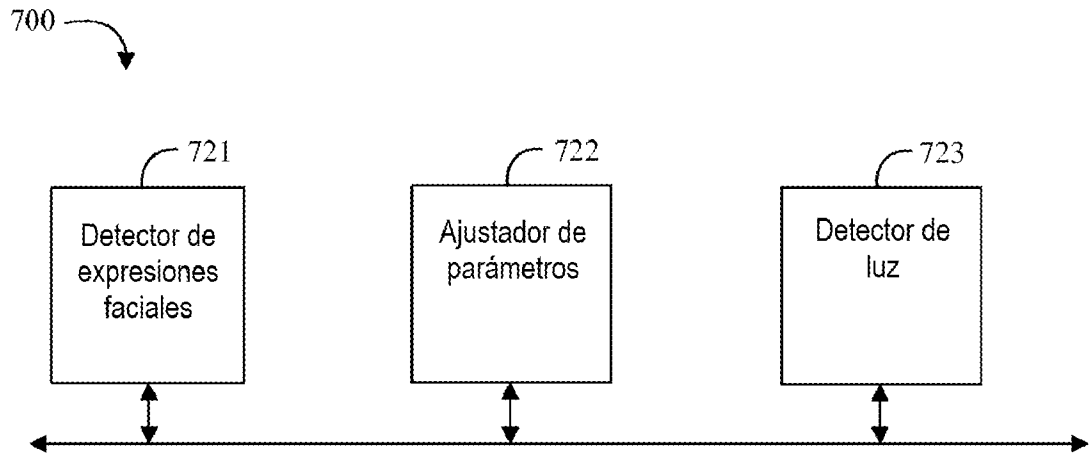


FIG. 7

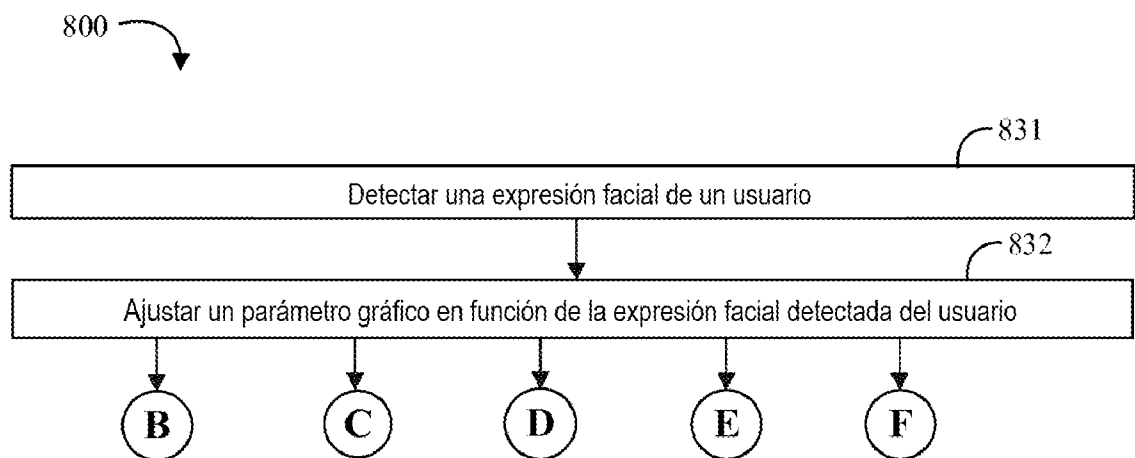


FIG. 8A

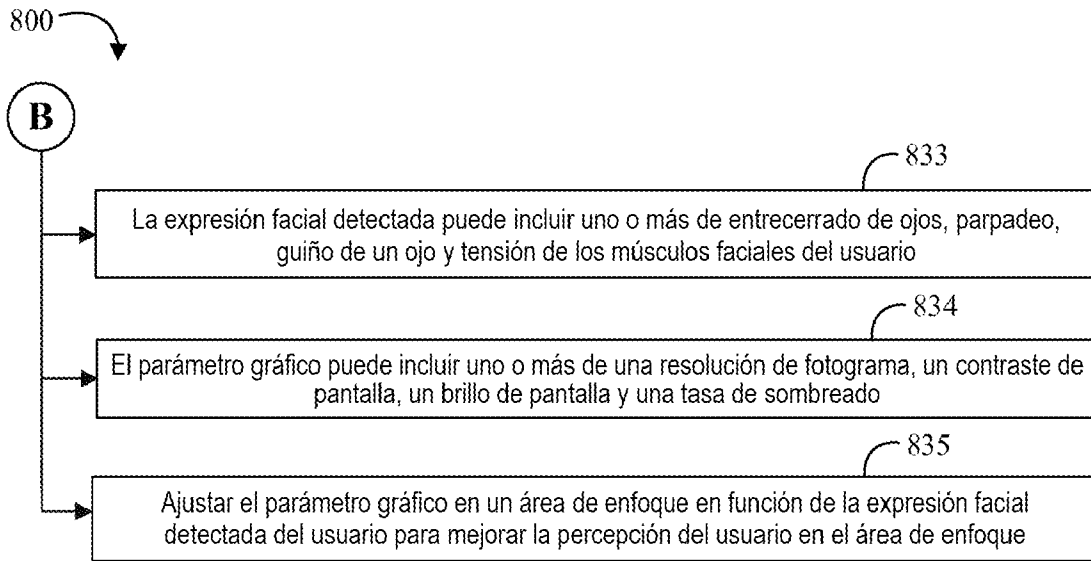


FIG. 8B

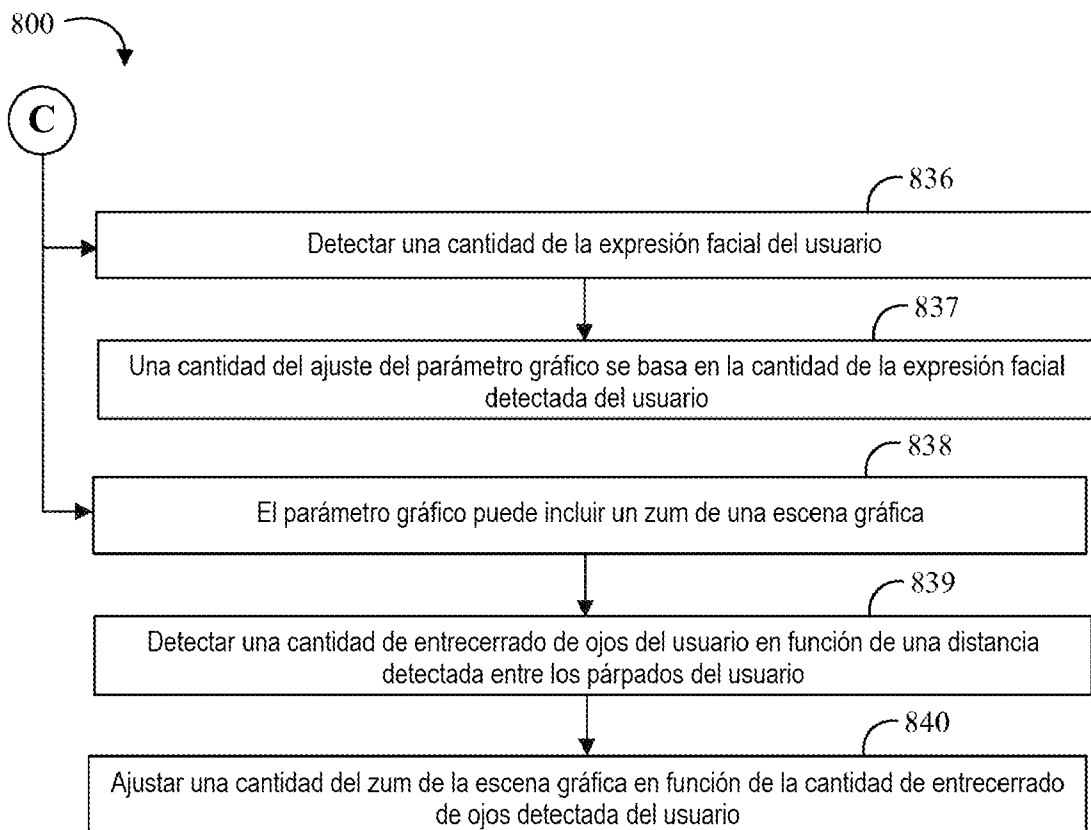


FIG. 8C

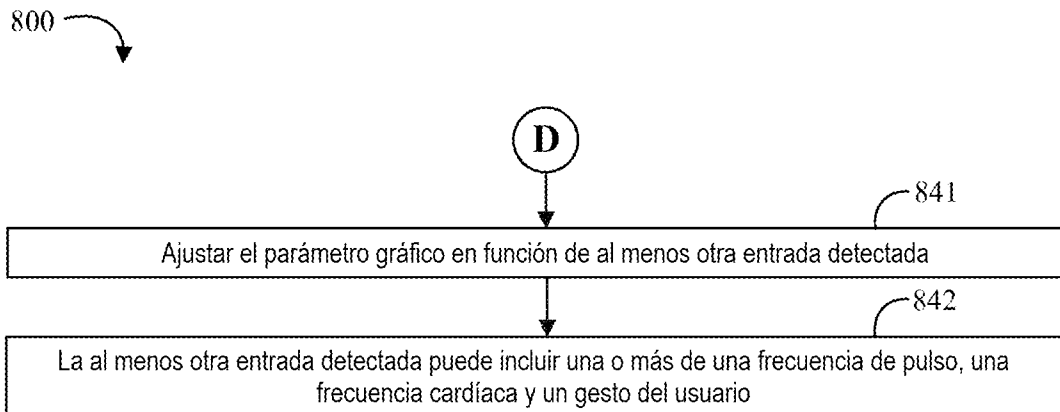


FIG. 8D

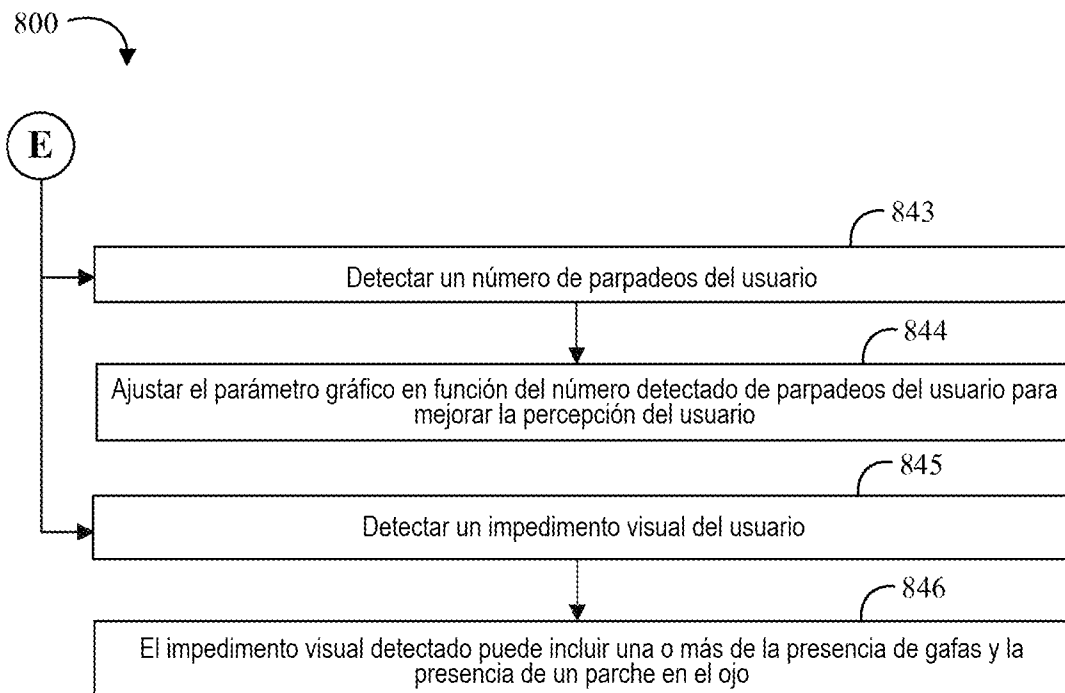


FIG. 8E

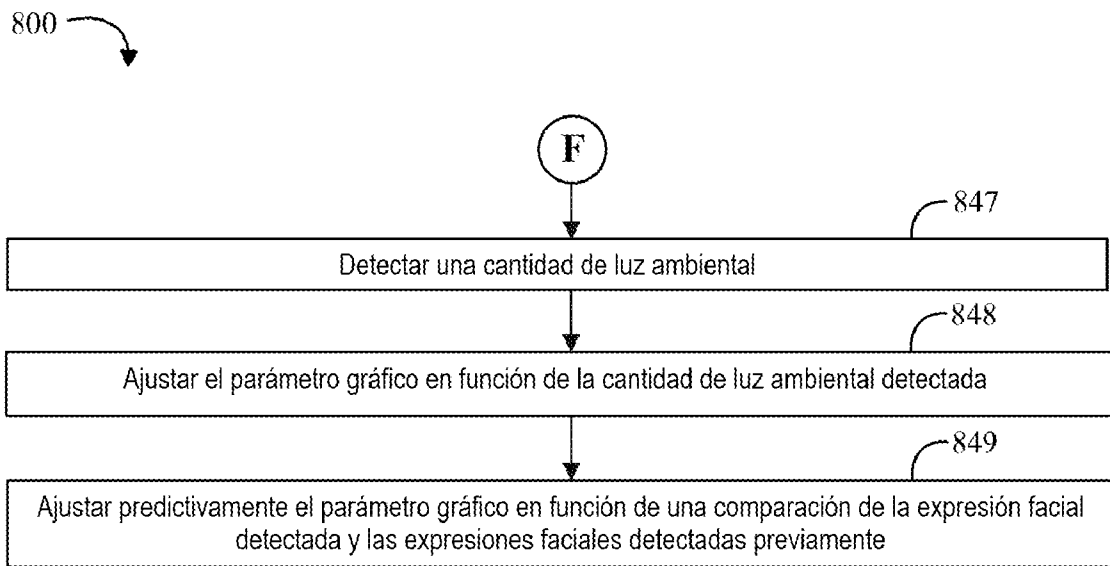


FIG. 8F

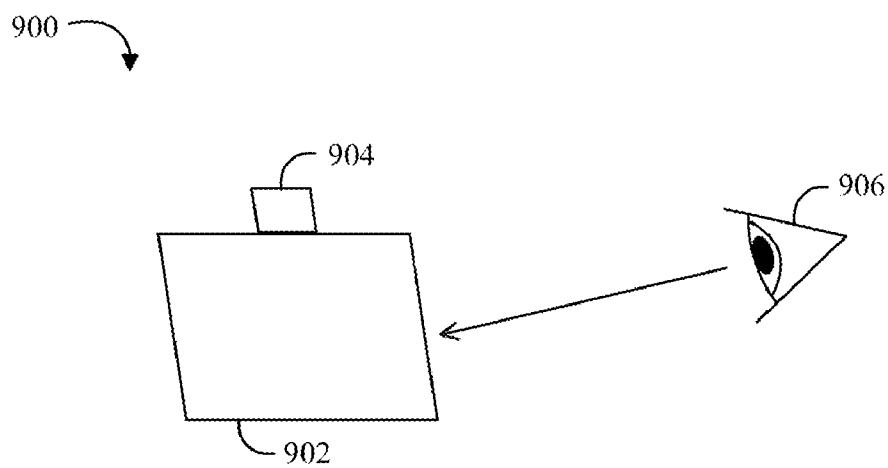


FIG. 9

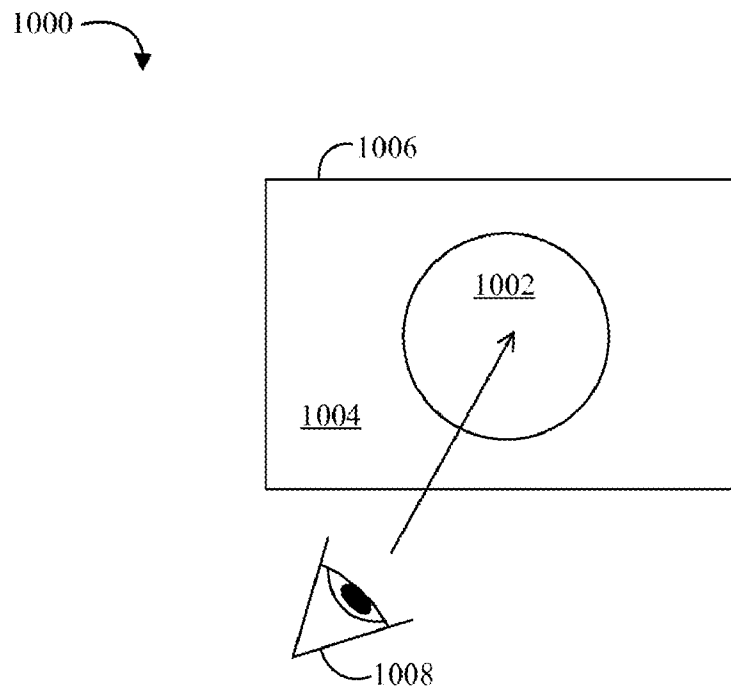


FIG. 10A

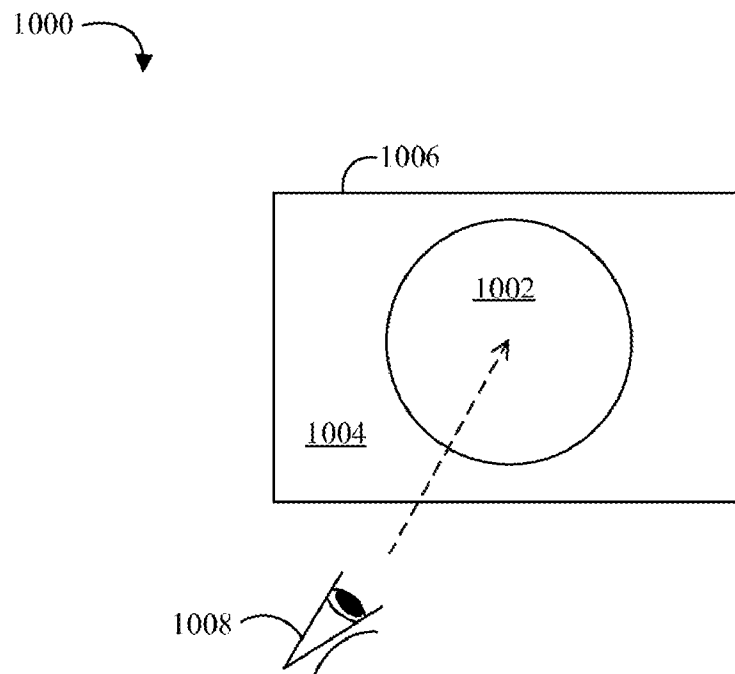


FIG. 10B

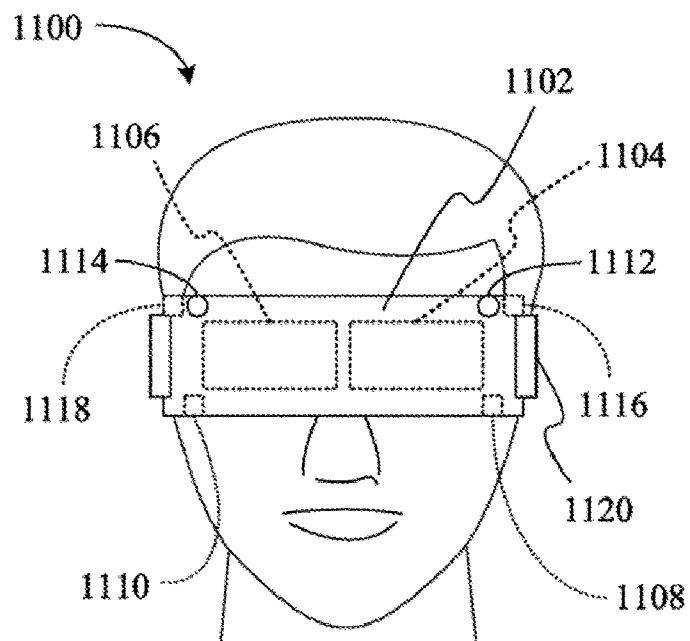


FIG. 11

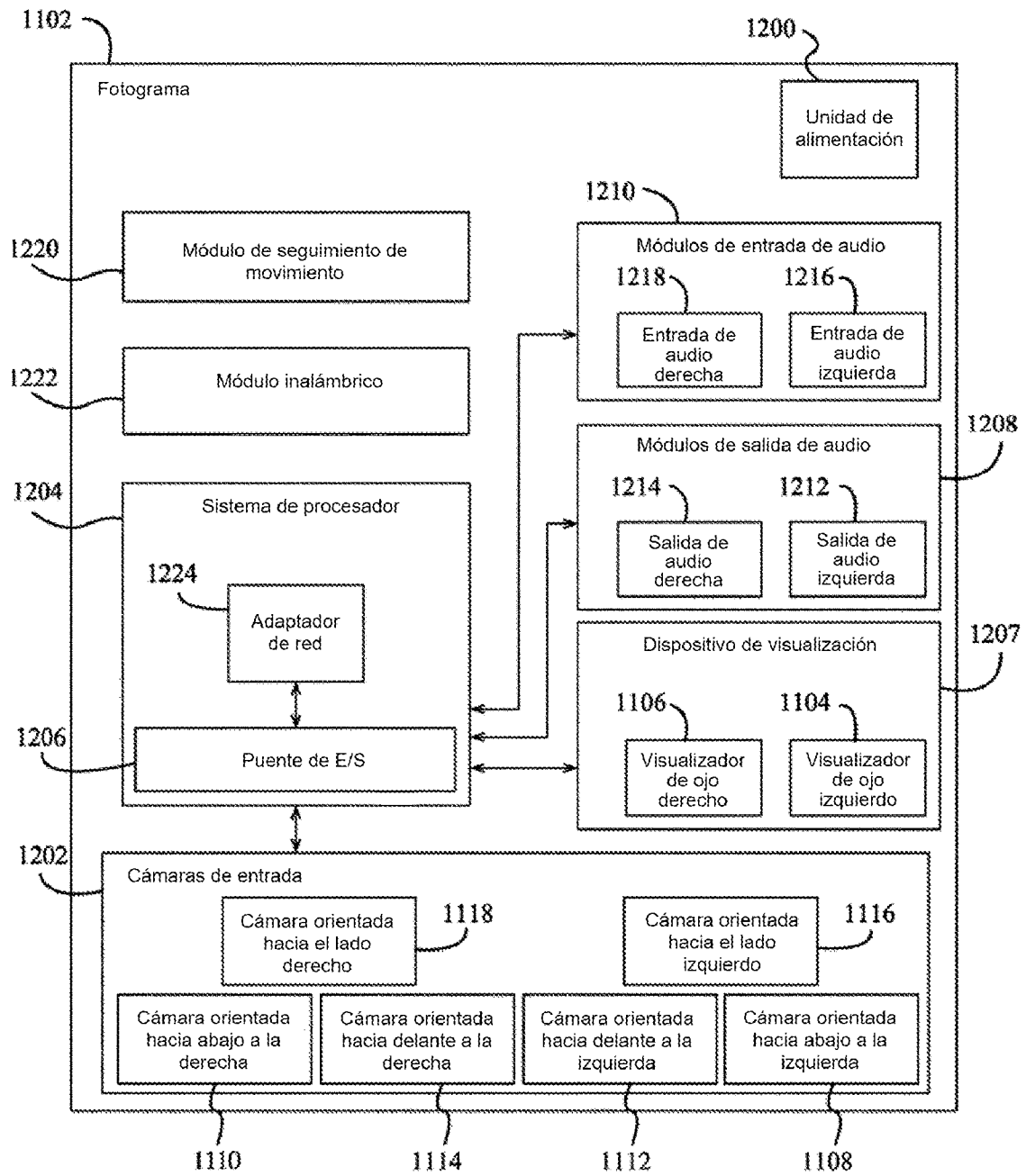


FIG. 12

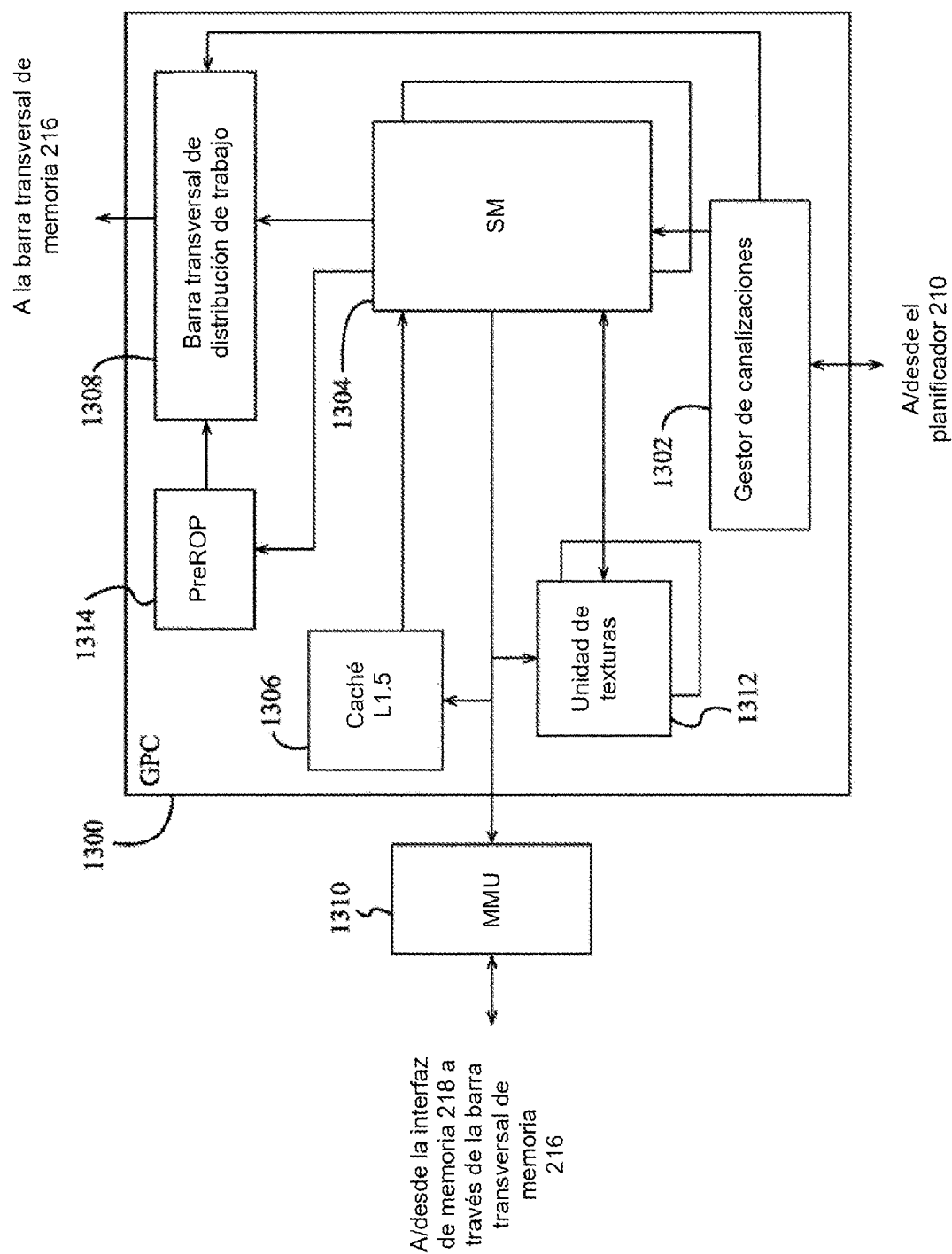


FIG. 13

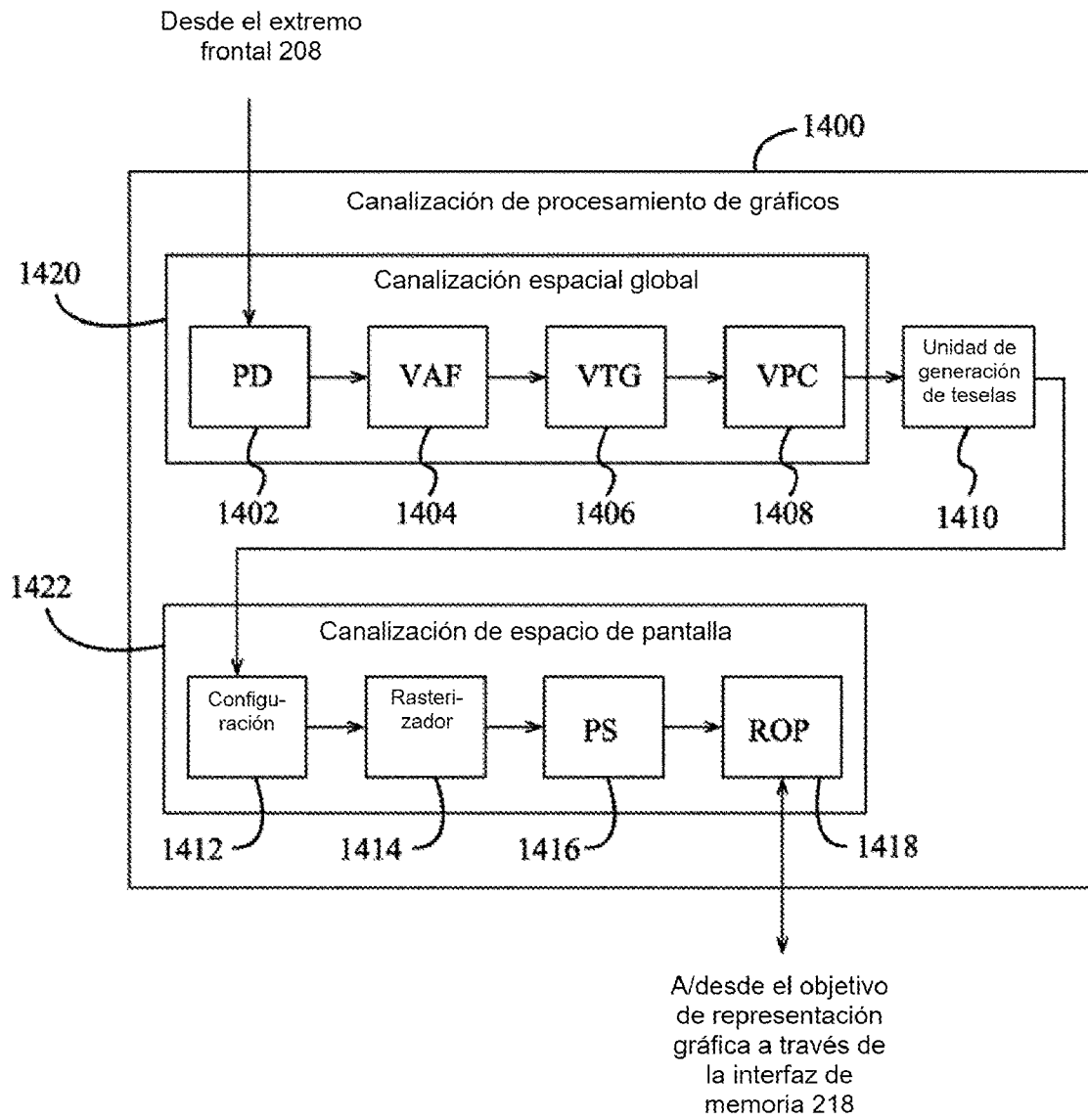


FIG. 14

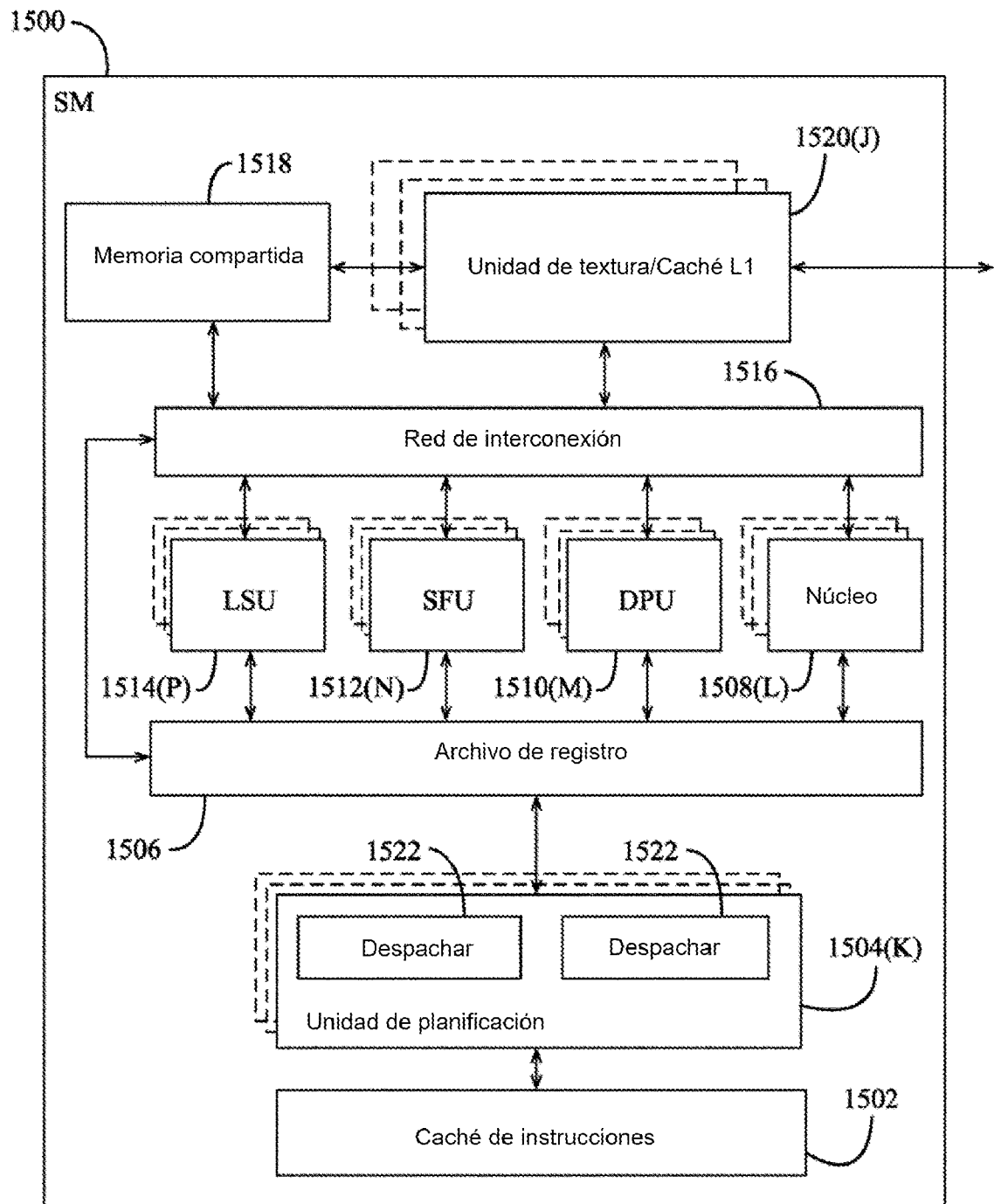


FIG. 15

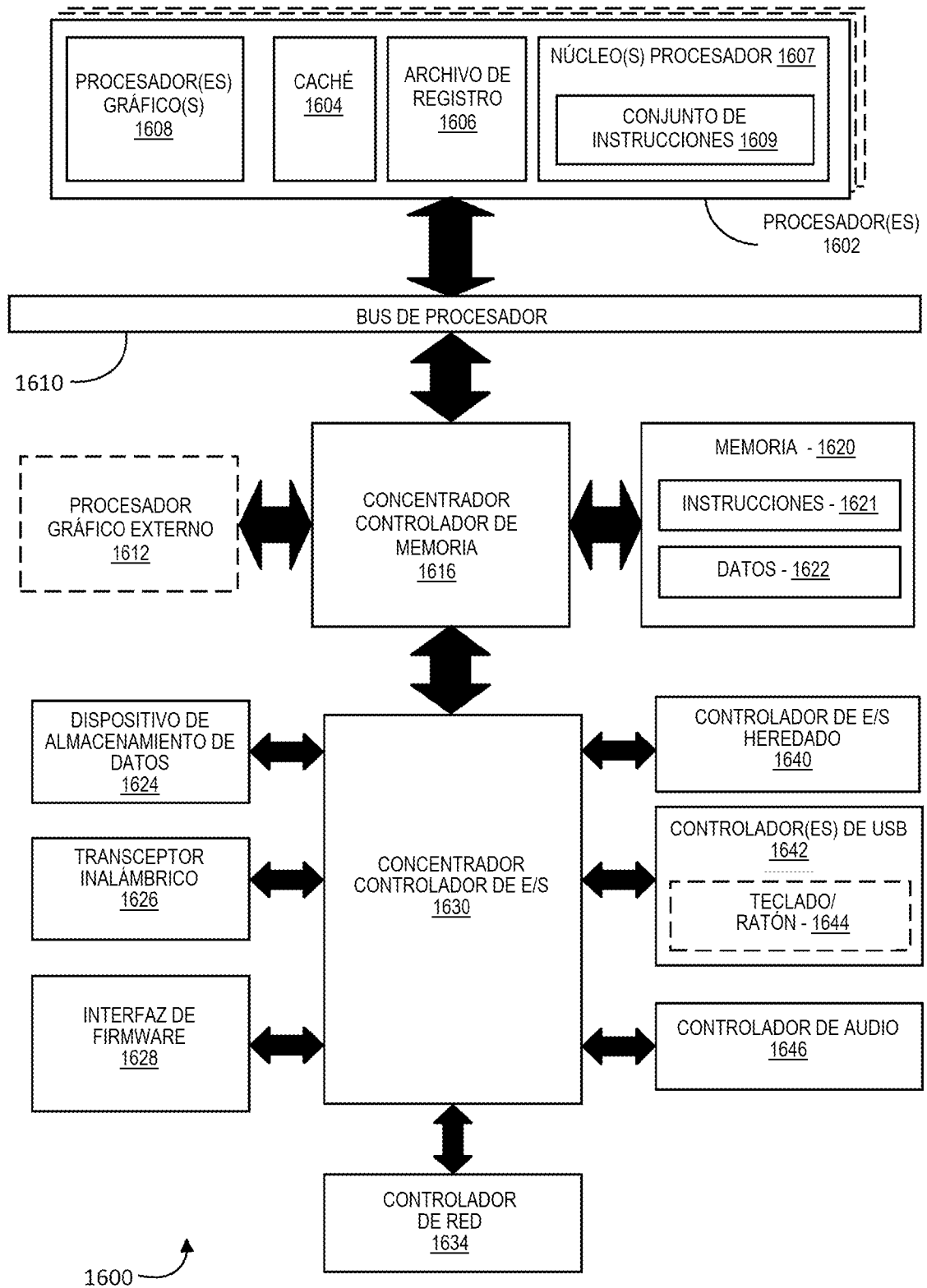


FIG. 16



PROCESADOR 1700 →

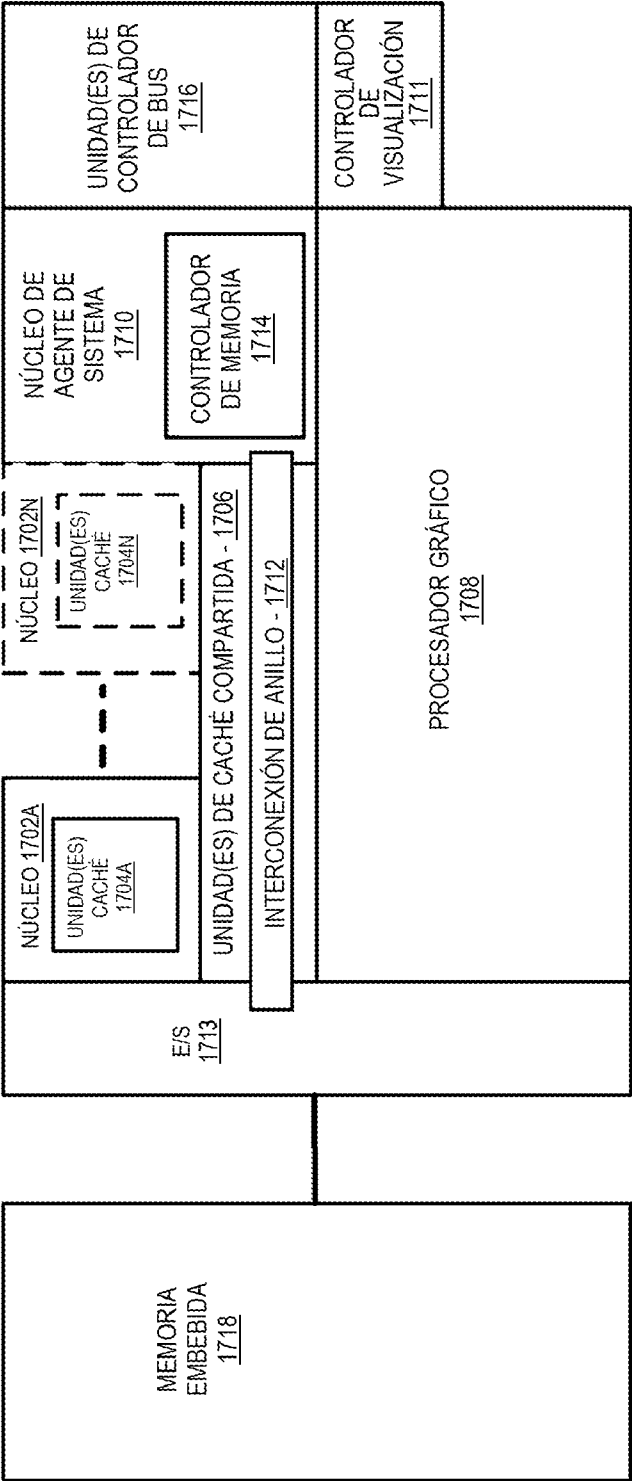


FIG. 17

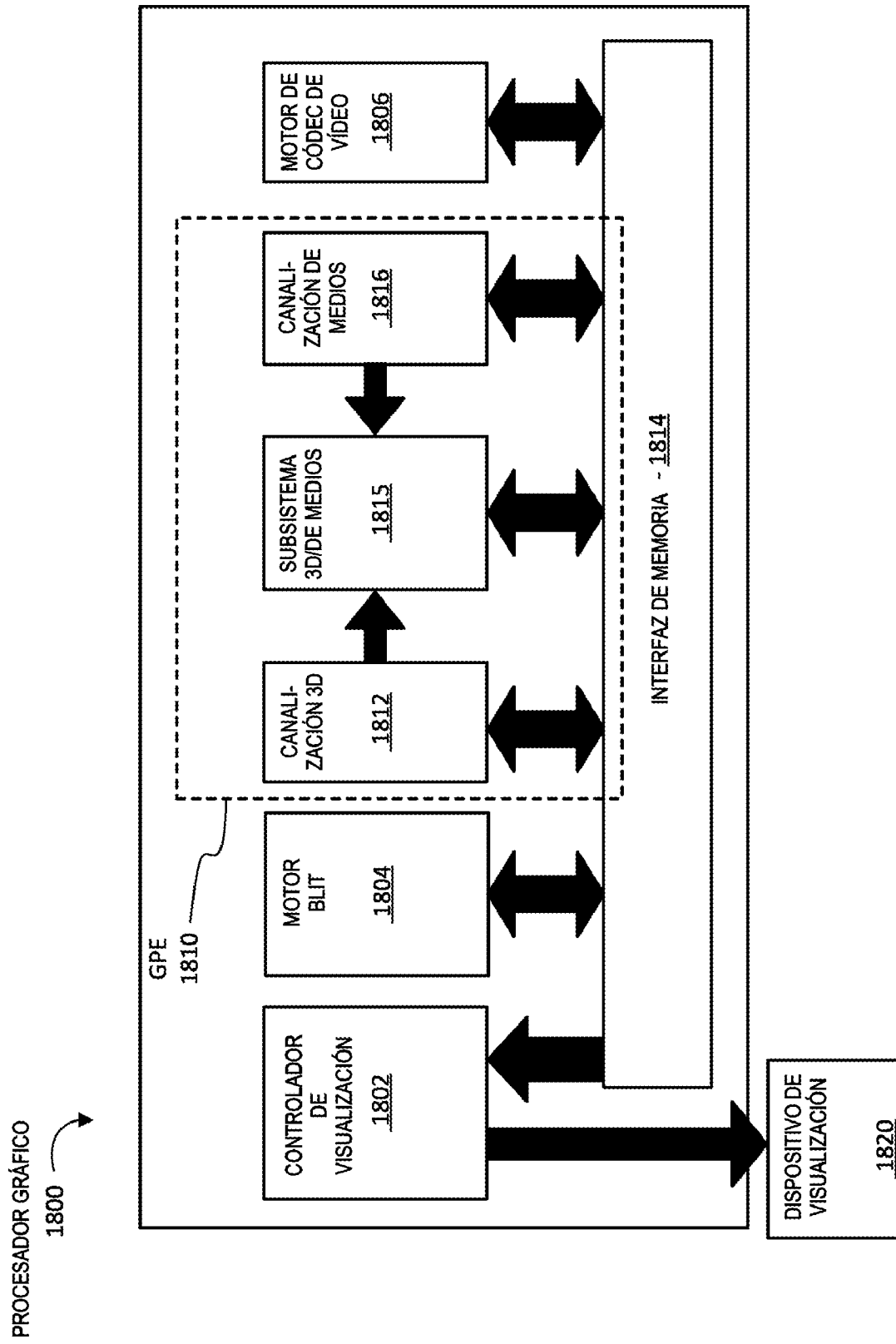


FIG. 18

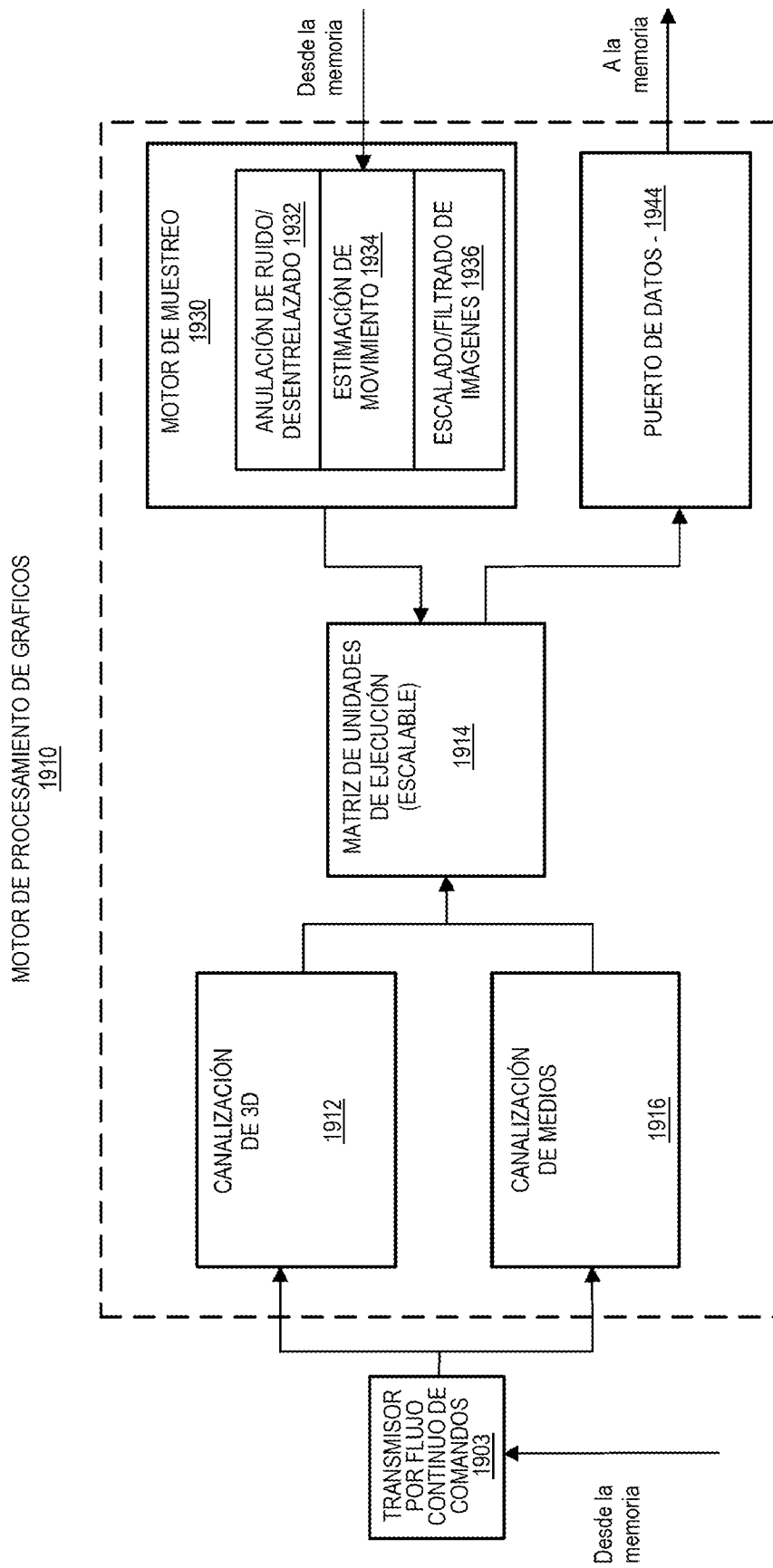


FIG. 19

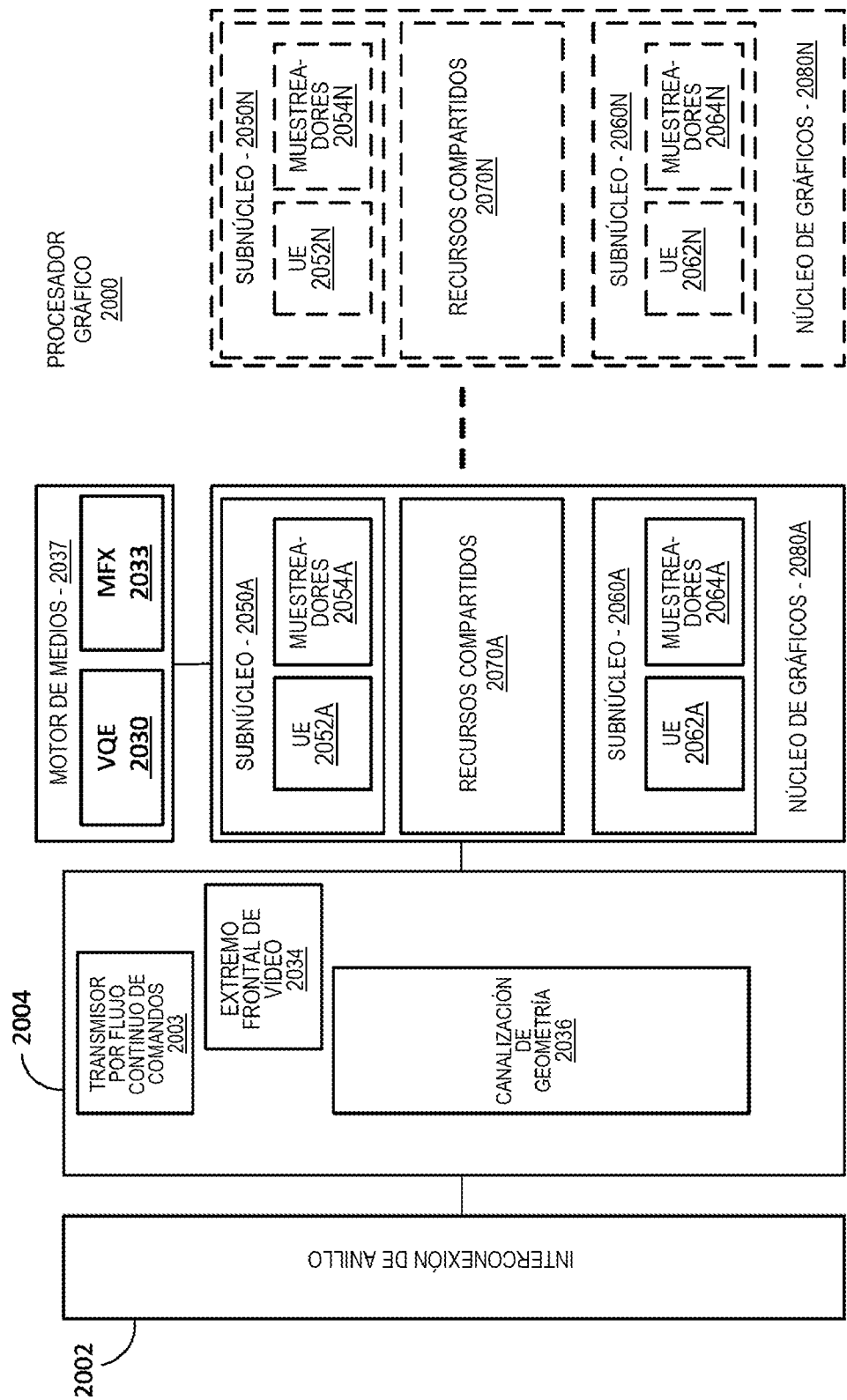


FIG. 20

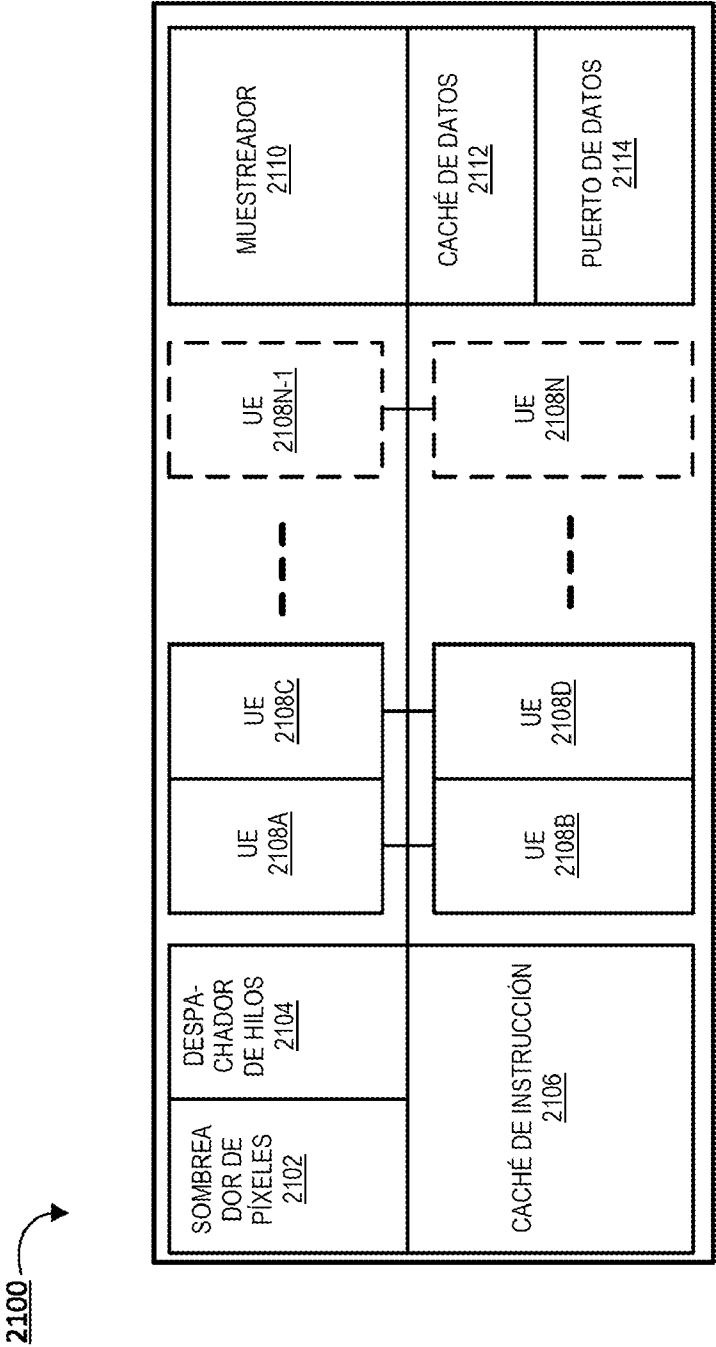
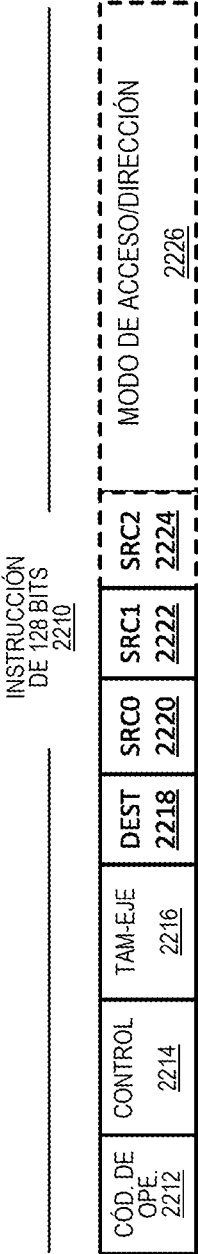


FIG. 21

FORMATOS DE INSTRUCCIONES DE PROCESADOR GRÁFICO
2200



INSTRUCCIÓN COMPACTA DE 64 BITS
2230



DESCODIFICACIÓN DE CÓDIGO DE OPERACIÓN
2240

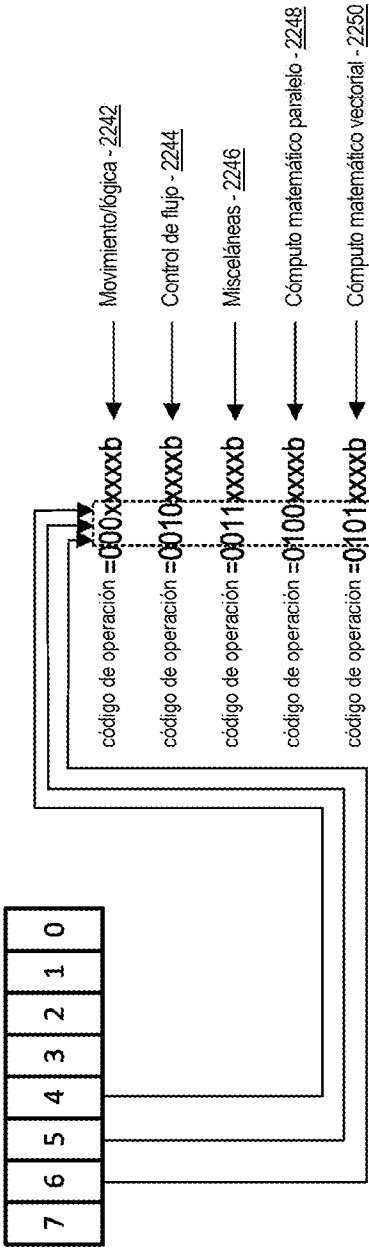


FIG. 22

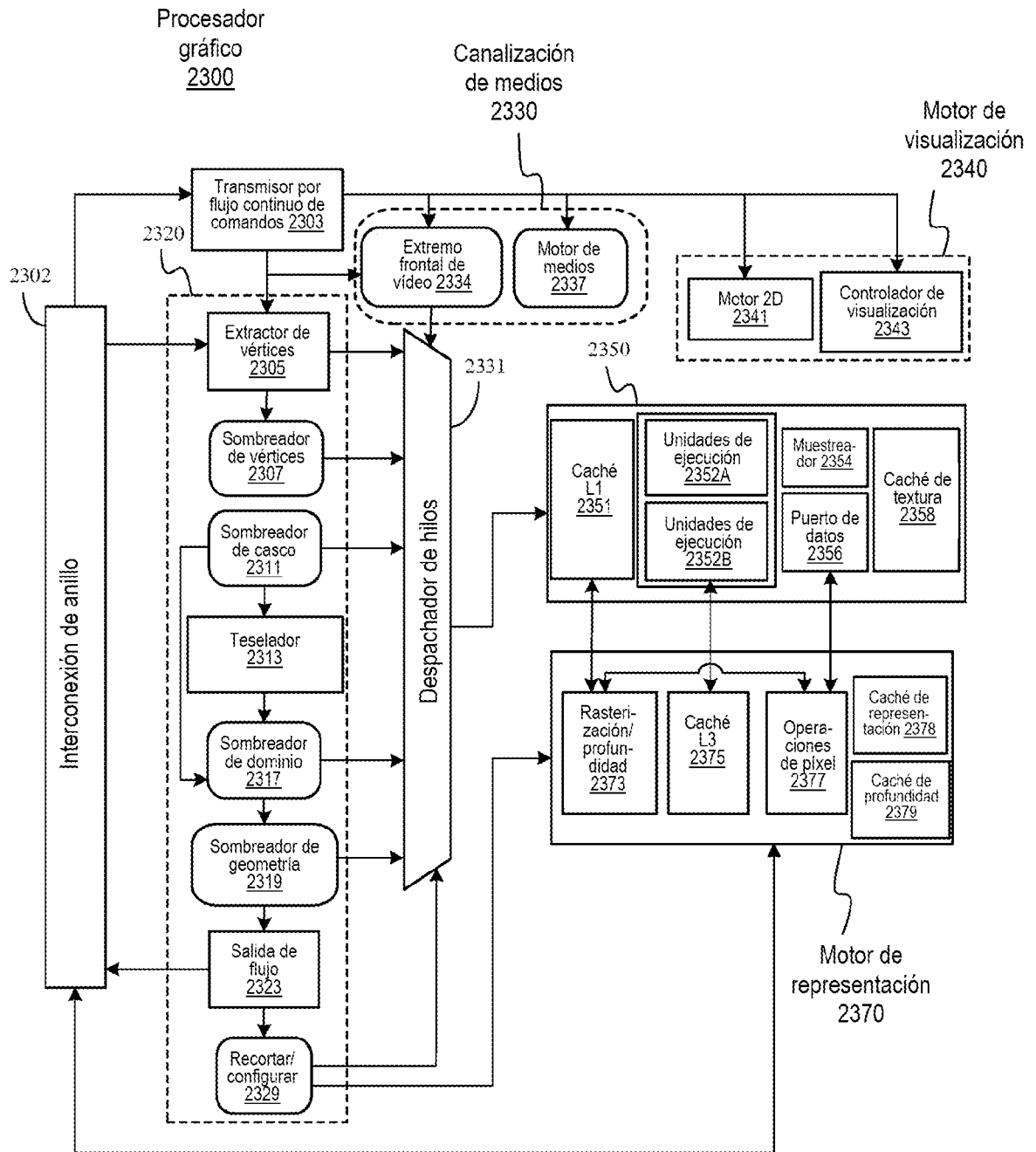


FIG. 23

Formato de comandos de procesador gráfico
2400

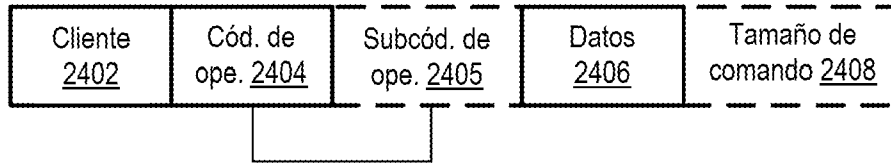


FIG. 24A

Secuencia de comandos de procesador gráfico
2410

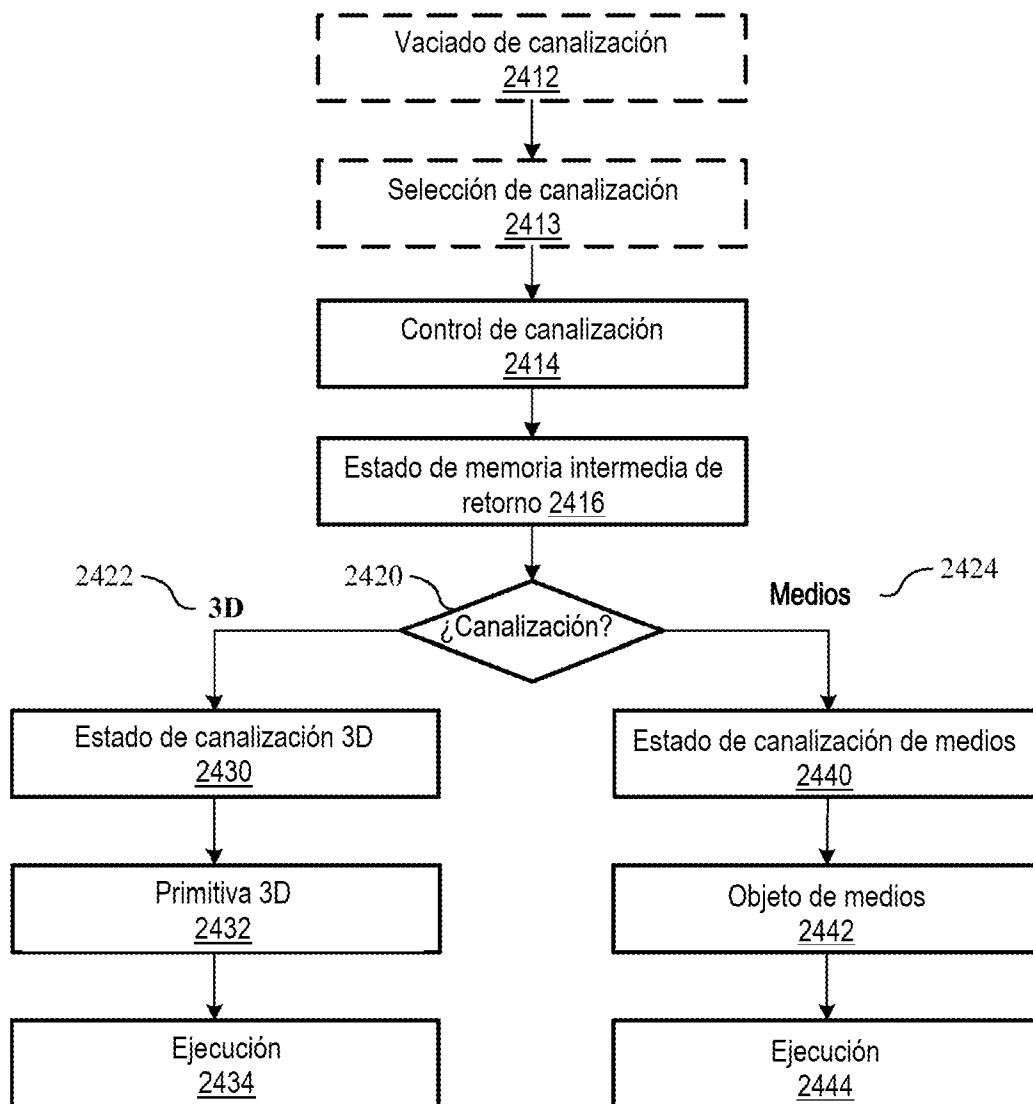


FIG. 24B

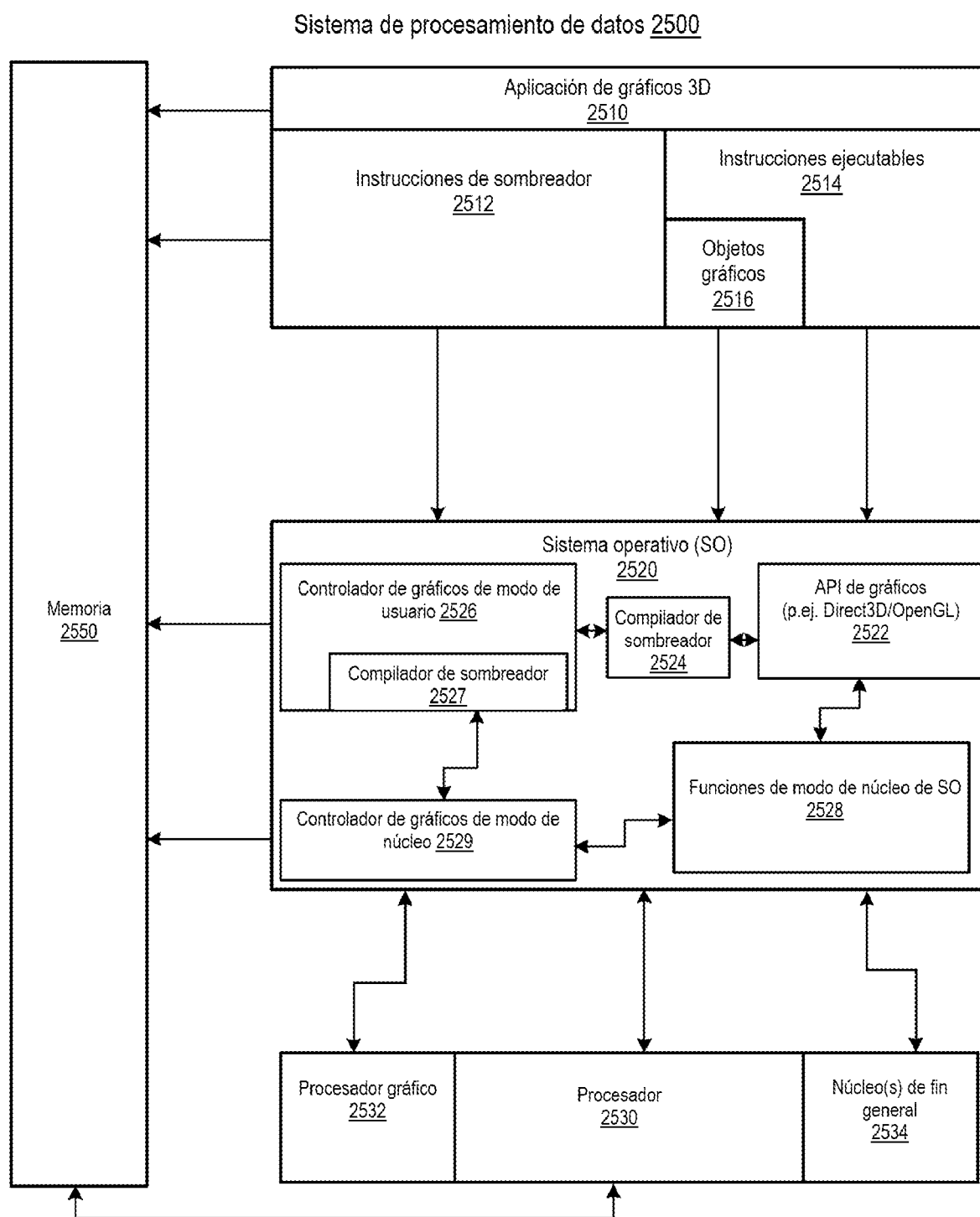


FIG. 25

DESARROLLO DE NÚCLEO DE IP - 2600

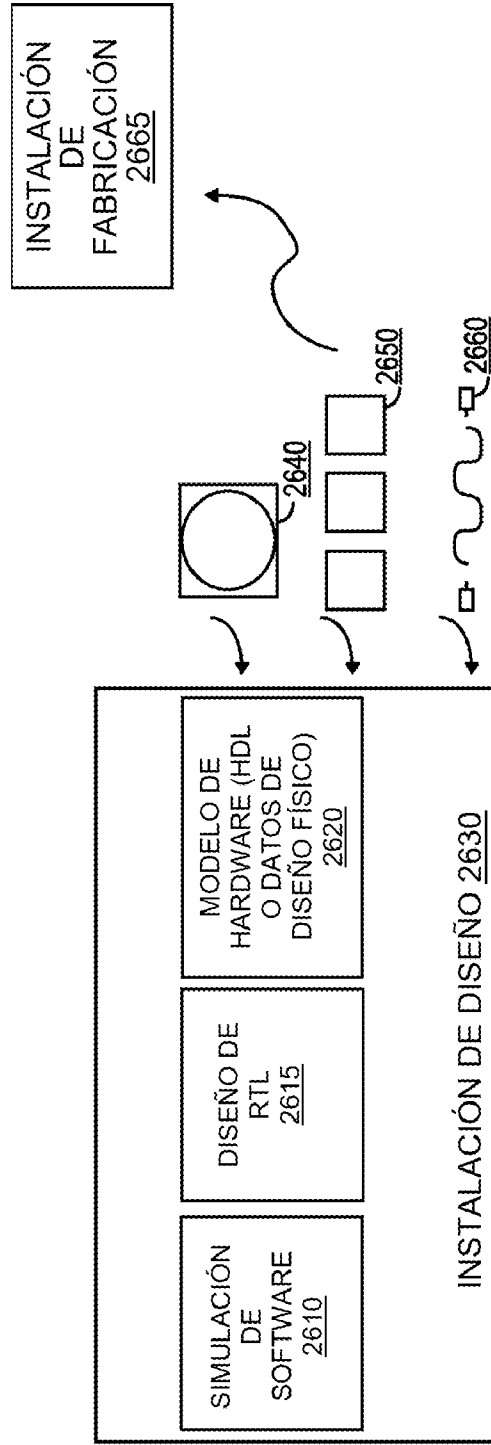


FIG. 26

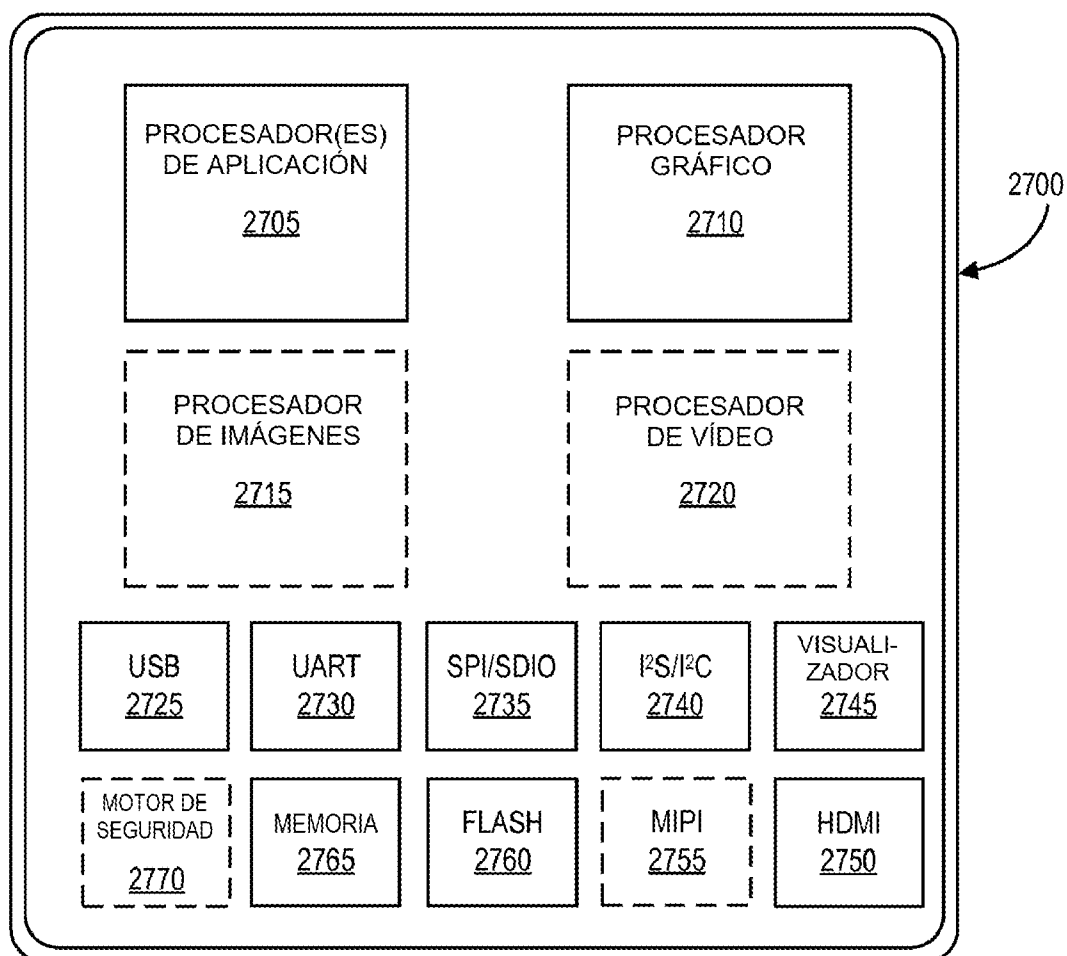


FIG. 27