

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 907 687**

51 Int. Cl.:

G06T 15/00 (2011.01)

G06T 1/20 (2006.01)

G06T 11/40 (2006.01)

G06T 3/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.03.2018** **E 19207151 (2)**

97 Fecha y número de publicación de la concesión europea: **15.12.2021** **EP 3637372**

54 Título: **Representación de plano de imagen multirresolución dentro de una microarquitectura de procesador gráfico mejorada**

30 Prioridad:

01.04.2017 US 201715477028

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

26.04.2022

73 Titular/es:

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

72 Inventor/es:

**APODACA, MICHAEL;
SURTI, PRASOONKUMAR;
VAIDYANATHAN, KARTHIK;
RAMADOSS, MURALI;
VENKATESH, ABHISHEK;
KENNEDY, JONATHAN y
GRAJEWSKI, SLAWOMIR**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 907 687 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

Representación de plano de imagen multirresolución dentro de una microarquitectura de procesador gráfico mejorada

5 CAMPO TÉCNICO

Las realizaciones se refieren en general al procesamiento de datos a través de una unidad de procesamiento gráfico y más particularmente al procesamiento de datos dentro de un sistema informático que tiene una canalización 3D con representación de plano de imagen multirresolución.

10

ANTECEDENTES DE LA DESCRIPCIÓN

El procesamiento de datos gráficos paralelo actual incluye sistemas y métodos desarrollados para realizar operaciones específicas en datos gráficos tales como, por ejemplo, interpolación lineal, teselación, rasterización, correlación de texturas, prueba de profundidad, etc. Tradicionalmente, los procesadores gráficos usan unidades de cálculo de función fija para procesar datos gráficos; sin embargo, más recientemente, porciones de procesadores gráficos se han hecho programables, habilitando que tales procesadores soporten una variedad más amplia de operaciones para el procesamiento de datos de vértices y fragmentos.

15

Como parte del procesamiento de datos gráficos, se generan imágenes que hay que visualizar a un usuario representando objetos gráficos en las imágenes de salida. Los enfoques actuales pueden usar una memoria intermedia de imagen de alta resolución en la que se representan todos los objetos gráficos. Este enfoque puede requerir cantidades significativas de cálculos que hay que realizar en la mayor resolución de imagen, aunque no todos los objetos en la escena 3D tienen la misma importancia para el observador.

20

25

En "Doug Binks: 'Dynamic Resolution Rendering Article', Intel Software - Developer Zone, 13 de julio de 2011, páginas 1-10", se divulga una representación de resolución dinámica. Por consiguiente, la representación de resolución dinámica implica ajustar la resolución a la que representar una escena 3D limitando la representación a una porción de un objetivo de representación que usa una ventanilla y, a continuación, escalando esto a una memoria intermedia posterior de salida. Los componentes de interfaz gráfica de usuario pueden representarse, a continuación, en la resolución de memoria intermedia posterior.

30

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

35

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;

40

La Figura 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 realizaciones;

45

La Figura 4A-4F ilustran una arquitectura ilustrativa en la que se acoplan una pluralidad de GPU comunicativamente a una pluralidad de procesadores de múltiples núcleos;

La Figura 5 ilustra una canalización de procesamiento gráfico, de acuerdo con una realización;

50

La Figura 6 es un diagrama conceptual de un ejemplo de planos de imagen representativos que tienen resolución múltiple que contienen objetos gráficos que hay que procesar por la canalización de procesamiento gráfico, de acuerdo con una realización;

La Figura 7A es un diagrama conceptual de un ejemplo de un plano de imagen representativo que contiene elementos de interfaz de usuario que hay que procesar por la canalización de procesamiento gráfico, de acuerdo con una realización;

55

La Figura 7B es un diagrama conceptual de un ejemplo de otro plano de imagen representativo que contiene elementos gráficos de plano de punto focal que hay que procesar por la canalización de procesamiento gráfico, de acuerdo con una realización;

60

La Figura 8 es un diagrama conceptual de un ejemplo de un plano de imagen combinado que contiene los elementos gráficos de cada plano de imagen con una resolución alta común, de acuerdo con una realización;

La Figura 9 es un diagrama de bloques de un ejemplo de un sistema informático con rendimiento mejorado de acuerdo con una realización;

65

La Figura 10 es un diagrama de flujo de un ejemplo de un método de representación de tareas de objetos gráficos con diversas resoluciones en memorias intermedias de imagen separadas de una canalización de procesamiento gráfico, de acuerdo con una realización;

La Figura 11 es una ilustración de un ejemplo de un sistema de pantalla montada en la cabeza (HMD) de acuerdo con una realización;

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

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 gráfico que puede implementarse dentro de una unidad de procesamiento paralelo, de acuerdo con una realización;

La Figura 15 es un diagrama de bloques de un ejemplo de un multiprocesador de transmisión de acuerdo con una realización;

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

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

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

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

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

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

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

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

DESCRIPCIÓN DETALLADA

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

Visión general del sistema

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

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

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

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

En una realización, el uno o más procesador o procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de gráficos y vídeo, incluyendo, por ejemplo, circuitería de salida de vídeo, y constituyen una unidad de procesamiento gráfico (GPU). En otra realización, el uno o más procesador o procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de fin general, mientras preserva la arquitectura de cálculo subyacente, descrita en mayor detalle en el presente documento. En otra realización más, componentes del sistema informático 100 pueden integrarse con uno o más otros elementos de sistema en un único circuito integrado. Por ejemplo, el uno o más procesador o procesadores paralelos 112, concentrador de memoria 105, procesador o procesadores 102 y concentrador de E/S 107 pueden integrarse en un circuito integrado de sistema en chip (SoC). Como alternativa, los componentes del sistema informático 100 pueden integrarse en un único paquete para formar una configuración de sistema en paquete (SIP). En una realización, al menos una porción de los componentes del sistema informático 100 puede integrarse en un módulo multichip (MCM), que puede interconectarse 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 procesador o procesadores 102 y el número de procesador o procesadores paralelos 112 pueden modificarse según se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 se conecta al procesador o procesadores 102 directamente en lugar de a través de un puente, mientras otros dispositivos se comunican con la memoria de sistema 104 a través del concentrador de memoria 105 y el procesador o procesadores 102. En otras topologías alternativas, el procesador o procesadores paralelos 112 se conectan al concentrador de E/S 107 o directamente a uno del uno o más procesador o procesadores 102, en lugar de al concentrador de memoria 105. En otras realizaciones, el concentrador de E/S 107 y el concentrador de memoria 105 pueden integrarse en un único chip. Algunas realizaciones pueden incluir dos o más conjuntos de procesador o procesadores 102 conectados a través de múltiples zócalos, que pueden acoplarse con dos o más instancias del procesador o procesadores paralelos 112.

Algunos de los componentes particulares mostrados en el presente documento son opcionales y pueden no incluirse en todas las implementaciones del sistema informático 100. Por ejemplo, puede soportarse cualquier número de tarjetas o periféricos de complemento, o algunos componentes pueden eliminarse. 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 como un puente norte en algunas arquitecturas, mientras el concentrador de E/S 107 puede denominarse como un puente sur.

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

procesadores paralelos 112 mostrados en la Figura 1, de acuerdo con una realización.

En una realización, el procesador paralelo 200 incluye una unidad de procesamiento paralelo 202. La unidad de procesamiento paralelo incluye una unidad de E/S 204 que habilita comunicación con otros dispositivos, incluyendo otras instancias de la unidad de procesamiento paralelo 202. La unidad de E/S 204 puede conectarse directamente a otros dispositivos. En una realización, la unidad de E/S 204 se conecta con otros dispositivos a través del uso de un concentrador o interfaz de conmutación, tal como el concentrador de memoria 105. Las conexiones entre el concentrador de memoria 105 y la unidad de E/S 204 forman un enlace de comunicación 113. Dentro de la unidad de procesamiento paralelo 202, la unidad de E/S 204 se conecta con una interfaz de anfitrión 206 y una barra cruzada de memoria 216, en donde la interfaz de anfitrión 206 recibe comandos dirigidos a realizar operaciones de procesamiento y la barra cruzada de memoria 216 recibe comandos dirigidos a realizar operaciones de memoria.

Cuando la interfaz de anfitrión 206 recibe una memoria intermedia de comandos a través de la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para realizar esos comandos a un extremo frontal 208. En una realización, el extremo frontal 208 se acopla con un planificador 210, que está configurado para distribuir comandos u otros artículos de trabajo a una matriz de agrupación de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupación de procesamiento 212 se configura correctamente y en un estado válido antes de que las tareas se distribuyan a las agrupaciones de procesamiento de la matriz de agrupación de procesamiento 212. En una realización, el planificador 210 se implementa a través de lógica de firmware que se ejecuta en un microcontrolador. El planificador implementado en microcontrolador 210 es configurable para realizar operaciones de planificación compleja y distribución de trabajo en granularidad gruesa y fina, habilitando una rápida conmutación de prioridad y contexto de hilos que se ejecutan en la matriz de procesamiento 212. En una realización, el software de anfitrión puede verificar cargas de trabajo para planificación en la matriz de procesamiento 212 a través de una de múltiples timbres de procesamiento gráfico. Las cargas de trabajo pueden distribuirse, a continuación, automáticamente a través de la matriz de procesamiento 212 por el planificador 210 lógica dentro del microcontrolador de planificador.

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

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

En una realización, la matriz de agrupación de procesamiento 212 está configurada para realizar operaciones de procesamiento gráfico paralelo. En realizaciones en las que el procesador paralelo 200 está configurado para realizar operaciones de procesamiento gráfico, la matriz de agrupación de procesamiento 212 puede incluir lógica adicional para soportar la ejecución de tales operaciones de procesamiento gráfico, incluyendo, pero sin limitación a lógica de muestreo de textura para realizar operaciones de textura, así como lógica de teselación y otra lógica de procesamiento de vértices. Adicionalmente, la matriz de agrupación de procesamiento 212 puede configurarse para ejecutar programas de sombreador relacionados con procesamiento gráfico tales como, pero sin limitación a, 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 a través de la unidad de E/S 204 para su procesamiento. Durante el procesamiento, los datos transferidos pueden almacenarse en memoria en chip (por ejemplo, memoria de procesador paralelo 222) durante el procesamiento, a continuación, escribirse de vuelta en la memoria de sistema.

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

214A-214N pueden almacenarse en memorias intermedias para permitir que los datos intermedios se transmitan entre las agrupaciones 214A-214N para su procesamiento adicional.

Durante la operación, la matriz de agrupación de procesamiento 212 puede recibir tareas de procesamiento que hay que ejecutar a través del planificador 210, que recibe comandos que definen tareas de procesamiento desde el extremo frontal 208. Para operaciones de procesamiento gráfico, las tareas de procesamiento pueden incluir índices de datos que hay que procesar, por ejemplo, datos de superficie (parche), datos de primitiva, datos de vértice y/o datos de píxel, así como parámetros de estado y comandos que definen cómo tienen que procesarse los datos (por ejemplo, qué programa tiene que ejecutarse). El planificador 210 puede configurarse para buscar los índices que corresponden a las tareas o puede recibir los índices desde el extremo frontal 208. El extremo frontal 208 puede configurarse para garantizar que la matriz de agrupación de procesamiento 212 está configurada a un estado válido antes de que se inicie la carga de trabajo especificada por memorias intermedias de comandos entrantes (por ejemplo, memorias intermedias por lotes, memorias intermedias de inserción, 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. La memoria de procesador paralelo 222 puede accederse a través de la barra cruzada de memoria 216, que puede recibir peticiones de memoria desde la matriz de agrupación de procesamiento 212 así como la unidad de E/S 204. La barra cruzada de memoria 216 puede acceder a la memoria de procesador paralelo 222 a través de una interfaz de memoria 218. La interfaz de memoria 218 puede incluir múltiples unidades de partición (por ejemplo, unidad de partición 220A, unidad de partición 220B a unidad de partición 220N) cada una de las cuales puede acoplarse a una porción (por ejemplo, unidad de memoria) de memoria de procesador paralelo 222. En una implementación el número de unidades de partición 220A-220N está configurado para ser igual al número de unidades de memoria, de tal forma que una primera unidad de división 220A tiene una correspondiente primera unidad de memoria 224A, una segunda unidad de división 220B tiene una correspondiente unidad de memoria 224B, y una N^{ésima} unidad de partición 220N tiene una correspondiente N^{ésima} unidad de memoria 224N. En otras realizaciones, el número de unidades de partición 220A-220N puede no ser igual al número de dispositivos de memoria.

En diversas realizaciones, las unidades de memoria 224A-224N pueden incluir diversos tipos de dispositivos de memoria, incluyendo memoria de acceso aleatorio dinámica (DRAM) o memoria gráfica de acceso aleatorio, tal como memoria gráfica de acceso aleatorio síncrona (SGRAM), incluyendo memoria de tasa de datos doble gráfica (GDDR). En una realización, las unidades de memoria 224A-224N también pueden incluir memoria apilada 3D, incluyendo, pero sin limitación a, memoria de ancho de banda alto (HBM). Expertos en la materia apreciarán que la implementación específica de las unidades de memoria 224A-224N puede variar, y puede seleccionarse de uno de diversos diseños convencionales. Objetivos de representación, tales como memorias intermedias de fotogramas o mapas de textura, pueden almacenarse a través de las unidades de memoria 224A-224N, permitiendo que las unidades de partición 220A-220N escriban porciones de cada objetivo de representación en paralelo a uso de forma eficiente del ancho de banda disponible de memoria de procesador paralelo 222. En algunas realizaciones, una instancia local de la memoria de procesador paralelo 222 puede excluirse en favor de un diseño de memoria unificada que utiliza memoria de sistema en conjunto con memoria caché local.

En una realización, una cualquiera de las agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212 puede procesar datos que se escribirán en cualquiera de las unidades de memoria 224A-224N dentro de la memoria de procesador paralelo 222. La barra cruzada de memoria 216 puede configurarse para transferir la salida de cada agrupación 214A-214N a cualquier unidad de partición 220A-220N o a otra agrupación 214A-214N, que puede realizar operaciones de procesamiento adicionales en la salida. Cada agrupación 214A-214N puede comunicarse con la interfaz de memoria 218 a través de la barra cruzada de memoria 216 para leer de o escribir en diversos dispositivos de memoria externa. En una realización, la barra cruzada 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, habilitando que las unidades de procesamiento dentro de las diferentes agrupaciones de procesamiento 214A-214N se comuniquen con memoria de sistema u otra memoria que no es local a la unidad de procesamiento paralelo 202. En una realización, la barra cruzada de memoria 216 puede usar canales virtuales para separar flujos de tráfico entre las agrupaciones 214A-214N y las unidades de partición 220A-220N.

Mientras una única instancia de la unidad de procesamiento paralelo 202 se ilustra dentro del procesador paralelo 200, puede incluirse cualquier número de instancias de la unidad de procesamiento paralelo 202. Por ejemplo, múltiples instancias de la unidad de procesamiento paralelo 202 pueden proporcionarse en una única tarjeta de complemento, o pueden interconectarse múltiples tarjetas de complemento. Las diferentes instancias de la unidad de procesamiento paralelo 202 pueden configurarse para interoperar incluso si las diferentes instancias tienen diferentes números de núcleos de procesamiento, diferentes cantidades de memoria local de procesador paralelo, 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 punto flotante de mayor precisión en relación con otras instancias. Sistemas que incorporan una o más instancias de la unidad de procesamiento paralelo 202 o el procesador paralelo 200 pueden implementarse en una diversidad de configuraciones y formar factores, incluyendo, pero sin limitación, ordenadores personales de escritorio, 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 partición 220, de acuerdo con una realización. En una realización, la unidad de partición 220 es una instancia de una de las unidades de partición 220A-220N de la Figura 2A. Como se ilustra, la unidad de partición 220 incluye una memoria caché L2 221, una interfaz de memoria intermedia de fotogramas 225 y una ROP (unidad de operaciones de trama) 226. La memoria caché L2 221 es una memoria caché de lectura/escritura que está configurada para realizar operaciones de carga y almacenamiento recibidas desde la barra cruzada de memoria 216 y la ROP 226. Fallos de lecturas y peticiones de reescritura urgentes se emiten por la memoria caché L2 221 a la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. También pueden enviarse actualizaciones a la memoria intermedia de fotogramas 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 memoria de procesador paralelo, tales como las unidades de memoria 224A-224N de la Figura 2 (por ejemplo, dentro de la memoria de procesador paralelo 222).

En aplicaciones gráficas, la ROP 226 es una unidad de procesamiento que realiza operaciones de trama tales como patrón, prueba z, combinación y similares. La ROP 226 emite, a continuación, datos gráficos procesados que se almacenan en memoria gráfica. En algunas realizaciones, la ROP 226 incluye lógica de compresión para comprimir datos de profundidad o color que se escriben en memoria y descomprime datos de profundidad o color que se leen de memoria. La lógica de compresión puede ser lógica de compresión sin pérdida que hace uso de uno o más de múltiples algoritmos de compresión. El tipo de compresión que se realiza por la ROP 226 puede variar basándose en las características estadísticas de los datos que hay que comprimir. Por ejemplo, en una realización, se realiza compresión de color delta en datos de profundidad y color sobre una base de por losa.

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

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

La operación de la agrupación de procesamiento 214 puede controlarse a través de un gestor de canalización 232 que distribuye tareas de procesamiento a procesadores paralelos de SIMT. El gestor de canalización 232 recibe instrucciones desde el planificador 210 de la Figura 2 y gestiona la ejecución de esas instrucciones a través de un multiprocesador gráfico 234 y/o una unidad de textura 236. El multiprocesador gráfico 234 ilustrado es una instancia ilustrativa de un procesador paralelo de SIMT. Sin embargo, diversos tipos de procesadores paralelos de SIMT de diferentes arquitecturas pueden incluirse dentro de la agrupación de procesamiento 214. Una o más instancias del multiprocesador gráfico 234 pueden incluirse dentro de una agrupación de procesamiento 214. El multiprocesador gráfico 234 puede procesar datos y puede usarse una barra cruzada de datos 240 para distribuir los datos procesados a uno de múltiples posibles destinos, incluyendo otras unidades de sombreador. El gestor de canalización 232 puede facilitar la distribución de datos procesados especificando destinos para datos procesados que hay que distribuir a través de la barra cruzada de datos 240.

Cada multiprocesador gráfico 234 dentro de la agrupación de procesamiento 214 puede incluir un conjunto idéntico de lógica de ejecución funcional (por ejemplo, unidades de lógica aritmética, unidades de carga-almacenamiento, etc.). La lógica de ejecución funcional puede configurarse en una manera de canalización en la que pueden emitirse nuevas instrucciones antes de que se completen las instrucciones anteriores. La lógica de ejecución funcional soporta una diversidad de operaciones que incluyen aritmética de enteros y de punto flotante, operaciones de comparación, operaciones booleanas, desplazamiento de bits y cálculo de diversas funciones de álgebra. En una realización, puede aprovecharse el mismo hardware de unidad funcional para realizar diferentes operaciones y puede estar presente cualquier combinación de unidades funcionales.

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

En una realización, el multiprocesador gráfico 234 incluye una memoria caché interna para realizar operaciones de carga y almacenamiento. En una realización, el multiprocesador gráfico 234 puede prescindir de una memoria caché interna y usar una memoria caché (por ejemplo, memoria caché L1 308) dentro de la agrupación de procesamiento 214. Cada multiprocesador gráfico 234 también tiene acceso a memorias caché L2 dentro de las unidades de partición (por ejemplo, las unidades de partición 220A-220N de la Figura 2) que se comparten entre todas las agrupaciones de procesamiento 214 y pueden usarse para transferir datos entre hilos. El multiprocesador gráfico 234 también puede acceder a memoria global fuera de chip, que puede incluir una o más de memoria local de procesador paralelo y/o memoria de sistema. Cualquier memoria externa a la unidad de procesamiento paralelo 202 puede usarse como memoria global. Realizaciones en las que la agrupación de procesamiento 214 incluye múltiples instancias del multiprocesador gráfico 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 (unidad de gestión de memoria) 245 que está configurada para correlacionar direcciones virtuales con direcciones físicas. En otras realizaciones, una o más instancias de la MMU 245 pueden residir dentro de la interfaz de memoria 218 de la Figura 2. La MMU 245 incluye un conjunto de entradas de tabla de paginación (PTE) usadas para correlacionar una dirección virtual con una dirección física de una losa (hábalese más de mosaico) y opcionalmente un índice de línea de memoria caché. La MMU 245 puede incluir memorias intermedias de traducción anticipada (TLB) de direcciones o memorias caché que pueden residir dentro del multiprocesador gráfico 234 o la memoria caché L1 o la agrupación de procesamiento 214. La dirección física se procesa para distribuir la localización de acceso a datos de superficie para permitir una petición eficiente que se intercala entre unidades de partición. El índice de línea de memoria caché puede usarse para determinar si una petición para una línea de caché es un acierto o fallo.

En aplicaciones gráficas e informáticas, una agrupación de procesamiento 214 puede configurarse de tal forma que cada multiprocesador gráfico 234 se acopla a una unidad de textura 236 para realizar operaciones de correlación de texturas, por ejemplo, determinando posiciones de muestra de texturas, leyendo datos de textura, y filtrando los datos de textura. Los datos de textura se leen de una memoria caché interna de textura L1 (no mostrada) o en algunas realizaciones de la memoria caché L1 dentro del multiprocesador gráfico 234 y se busca en una memoria caché L2, memoria local de procesador paralelo, o memoria de sistema, según se necesite. Cada multiprocesador gráfico 234 emite tareas procesadas a la barra cruzada de datos 240 para proporcionar la tarea procesada a otra agrupación de procesamiento 214 para su procesamiento adicional o para almacenar la tarea procesada en una memoria caché L2, memoria local de procesador paralelo o memoria de sistema a través de la barra cruzada de memoria 216. Una preROP 242 (unidad de operaciones de pretrama) está configurada para recibir datos desde el multiprocesador gráfico 234, dirigir datos a unidades de ROP, que pueden ubicarse con unidades de partición como se describe en el presente documento (por ejemplo, las unidades de partición 220A-220N de la Figura 2). La unidad de preROP 242 puede realizar optimizaciones para combinación de colores, organizar datos de color de píxeles y realizar traducciones de direcciones.

Se apreciará que el núcleo arquitectura descrito en el presente documento es ilustrativo y que son posibles variaciones y modificaciones. Cualquier número de unidades de procesamiento, por ejemplo, el multiprocesador gráfico 234, la unidad de texturas 236, las preROP 242, etc., pueden incluirse dentro de una agrupación de procesamiento 214. Además, mientras se muestra únicamente una agrupación de procesamiento 214, una unidad de procesamiento paralelo como se describe en el presente documento puede incluir cualquier número de instancias de la agrupación de procesamiento 214. En una realización, cada agrupación de procesamiento 214 puede configurarse para operar independientemente de otras agrupaciones de procesamiento 214 usando unidades de procesamiento, memorias caché L1, etc. separadas y distintas.

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

268.

En una realización, la memoria caché de instrucciones 252 recibe un flujo de instrucciones para ejecutar desde el gestor de canalización 232. Las instrucciones se almacenan en memoria caché en la memoria caché de instrucciones 252 y se distribuyen para su ejecución por la unidad de instrucción 254. La unidad de instrucción 254 puede distribuir instrucciones como grupos de hilos (por ejemplo, saltos), con cada hilo del grupo de hilos asignado a una diferente unidad de ejecución dentro del núcleo de GPGPU 262. Una instrucción puede acceder a cualquiera de un espacio de direcciones locales, compartidas o globales especificando una dirección dentro de un espacio de direcciones unificadas. La unidad de correlación de direcciones 256 puede usarse para traducir direcciones en el espacio de direcciones unificadas a una dirección de memoria distinta que puede accederse por las unidades de carga/almacenamiento 266.

El archivo de registro 258 proporciona un conjunto de registros para las unidades funcionales del multiprocesador gráfico 324. El archivo de registro 258 proporciona almacenamiento temporal para operandos conectados a las trayectorias de datos de las unidades funcionales (por ejemplo, núcleos de GPGPU 262, unidades de carga/almacenamiento 266) del multiprocesador gráfico 324. En una realización, el archivo de registro 258 se divide entre cada una de las unidades funcionales de tal forma que a cada unidad funcional se asigna una porción especializada del archivo de registro 258. En una realización, el archivo de registro 258 se divide entre los diferentes saltos que se ejecutan por el multiprocesador gráfico 324.

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

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

La interconexión de memoria y memoria caché 268 es una red de interconexión que conecta cada una de las unidades funcionales del multiprocesador gráfico 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 cruzada que permite que la unidad de carga/almacenamiento 266 implemente operaciones de carga y almacenamiento entre la memoria compartida 270 y el archivo de registro 258. El archivo de registro 258 puede operar en la misma frecuencia que los núcleos de GPGPU 262, por lo tanto, la transferencia de datos entre los núcleos de GPGPU 262 y el archivo de registro 258 es de latencia muy baja. La memoria compartida 270 puede usarse para habilitar comunicación entre hilos que se ejecutan en las unidades funcionales dentro del multiprocesador gráfico 234. La memoria caché 272 puede usarse como una memoria caché de datos, por ejemplo, para almacenar en memoria caché datos de textura comunicados entre las unidades funcionales y la unidad de textura 236. La memoria compartida 270 también puede usarse como una memoria caché gestionada por programa. Los hilos que se ejecutan en los núcleos de GPGPU 262 pueden almacenar programáticamente datos dentro de la memoria compartida además de los datos almacenados en memoria caché automáticamente que se almacenan dentro de la memoria caché 272.

Las Figuras 3A-3B ilustran multiprocesadores gráficos adicionales, de acuerdo con realizaciones. Los multiprocesadores gráficos 325, 350 ilustrados son variantes del multiprocesador gráfico 234 de la Figura 2C. Los multiprocesadores gráficos 325, 350 ilustrados pueden configurarse como un multiprocesador de transmisión (SM) con capacidad de ejecución simultánea de un gran número de hilos de ejecución.

La Figura 3A muestra un multiprocesador gráfico 325 de acuerdo con una realización adicional. El multiprocesador gráfico 325 incluye múltiples instancias adicionales de unidades de recursos de ejecución en relación con el multiprocesador gráfico 234 de la Figura 2D. Por ejemplo, el multiprocesador gráfico 325 puede incluir múltiples instancias de la unidad de instrucción 332A-332B, el archivo de registro 334A-334B y la unidad o unidades de textura 344A-344B. El multiprocesador gráfico 325 también incluye múltiples conjuntos de unidades de ejecución gráficas o de cálculo (por ejemplo, el núcleo de GPGPU 336A-336B, el núcleo de GPGPU 337A-337B, el núcleo de GPGPU

338A-338B) y múltiples conjuntos de unidades de carga/almacenamiento 340A-340B. En una realización, las unidades de recursos de ejecución tienen una memoria caché de instrucciones 330, una memoria caché de textura y/o datos 342 y una memoria compartida 346 común.

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

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

25 Expertos en la materia entenderán que la arquitectura descrita en las Figuras 1, 2A-2D y 3A-3B son descriptivas y no limitantes del alcance de las presentes realizaciones. Por lo tanto, las técnicas descritas en el presente documento pueden implementarse en cualquier unidad de procesamiento configurada apropiadamente, incluyendo, sin limitación, uno o más procesadores de aplicación móvil, una o más unidades de procesamiento central (CPU) de escritorio o servidor que incluyen CPU de múltiples núcleos, una o más unidades de procesamiento paralelo, tal como la unidad de procesamiento paralelo 202 de la Figura 2, así como uno o más procesadores gráficos o unidades de procesamiento de fin especial, sin alejarse del alcance de las realizaciones descritas en el presente documento.

En algunas realizaciones, un procesador paralelo o GPGPU como se describe en el presente documento se acopla comunicativamente a núcleos de anfitrión/procesador para acelerar operaciones gráficas, operaciones de aprendizaje automático, operaciones de análisis de patrón y diversas funciones de GPU de fin general (GPGPU). La GPU puede acoplarse comunicativamente al procesador de anfitrión/núcleos a través de un bus u otra interconexión (por ejemplo, una interconexión de alta velocidad tal como PCIe o NVLink). En otras realizaciones, la GPU puede integrarse en el mismo paquete o chip que los núcleos y acoplarse comunicativamente a los núcleos a través de un bus/interconexión de procesador interno (es decir, interno al paquete o chip). Independientemente de la manera en la que se conecta la GPU, los núcleos de procesador pueden asignar trabajo a la GPU en forma de secuencias de comandos/instrucciones contenidas en un descriptor de trabajo. La GPU usa, a continuación, circuitería/lógica especializada para procesar de forma eficiente estos comandos/instrucciones.

Técnicas para interconexión de GPU a procesador de anfitrión

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

Además, en una realización, dos o más de las GPU 410-413 se interconectan a través de enlaces de alta velocidad 444-445, que pueden implementarse usando los mismos o diferentes protocolos/enlaces que los usados para los enlaces de alta velocidad 440-443. De manera similar, dos o más de los procesadores de múltiples núcleos 405-406 pueden conectarse a través del enlace de alta velocidad 433 que puede ser buses de múltiples procesadores simétricos (SMP) que operan 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 lograrse usando los mismos protocolos/enlaces (por ejemplo, a través de un tejido de interconexión común). Como se ha mencionado, sin embargo, los principios subyacentes de la invención no se limitan a cualquier tipo particular de tecnología de interconexión.

En una realización, cada procesador de múltiples núcleos 405-406 se acopla comunicativamente a una memoria de procesador 401-402, a través de las interconexiones de memoria 430-431, respectivamente, y cada GPU 410-413 se acopla comunicativamente a la memoria de GPU 420-423 a través de las interconexiones de memoria de GPU 450-453, respectivamente. Las interconexiones de memoria 430-431 y 450-453 pueden utilizar las mismas o diferentes

tecnologías de acceso a memoria. A modo de ejemplo, y no como limitación, las memorias de procesador 401-402 y memorias de GPU 420-423 pueden ser memorias volátiles tales como memorias de acceso aleatorio dinámicas (DRAM) (incluyendo DRAM apiladas), SDRAM de DDR Gráfica (GDDR) (por ejemplo, GDDR5, GDDR6), o Memoria de Ancho de Banda Alto (HBM) y/o pueden ser memorias no volátiles tales como 3D XPoint o Nano-Ram. En una realización, alguna porción de las memorias puede ser memoria volátil y otra porción puede ser memoria no volátil (por ejemplo, usando una jerarquía de memorias de dos niveles (2LM)).

Como se describe a continuación, aunque los diversos procesadores 405-406 y las GPU 410-413 pueden acoplarse físicamente a una memoria 401-402, 420-423 particular, respectivamente, puede implementarse una arquitectura de memoria unificada en la que el mismo espacio de direcciones de sistema 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 de sistema y cada una de las memorias de GPU 420-423 puede comprender 32 GB del espacio de direcciones de memoria de sistema (resultando en 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 gráfica 446 de conformidad con una realización. El módulo de aceleración gráfica 446 puede incluir uno o más chips de GPU integrados en una tarjeta de línea que se acopla al procesador 407 a través del enlace de alta velocidad 440. Como alternativa, el módulo de aceleración gráfica 446 puede integrarse 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 traducción anticipada 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 ilustran para evitar obstaculizar los principios subyacentes de la invención (por ejemplo, unidades de búsqueda de instrucción, unidades de predicción de rama, decodificadores, unidades de ejecución, memorias intermedias de reordenación, etc.). Las memorias caché 462A-462D pueden comprender memorias caché de nivel 1 (L1) y nivel 2 (L2). Además, una o más memorias caché compartidas 426 pueden incluirse en la jerarquía de almacenamiento en memoria caché y compartirse por conjuntos de los núcleos 460A-460D. Por ejemplo, una realización del procesador 407 incluye 24 núcleos, cada uno con su propia 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 se comparten por dos núcleos adyacentes. El procesador 407 y el módulo de integración de acelerador gráfico 446 se conectan con la memoria de sistema 441, que puede incluir las memorias de procesador 401-402.

La coherencia se mantiene para datos e instrucciones almacenadas en las diversas memorias caché 462A-462D, 456 y la memoria de sistema 441 a través de comunicación entre núcleos a través de un bus de coherencia 464. Por ejemplo, cada memoria caché puede tener lógica/circuitería de coherencia de memoria caché asociada con 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 inspección de memoria caché a través del bus de coherencia 464 para inspeccionar accesos a memoria caché. Los expertos en la materia entienden bien las técnicas de investigación/coherencia de memoria caché y no se describirán en detalle en este punto para evitar obstaculizar los principios subyacentes de la invención.

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

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

En una realización, el circuito de integración de acelerador 436 incluye una unidad de gestión de memoria (MMU) 439 para realizar diversas funciones de gestión de memoria tales como traducciones de memoria de virtual a física (también denominadas traducciones de memoria efectiva a real) y protocolos de acceso a memoria para acceder a la memoria de sistema 441. La MMU 439 también puede incluir una memoria intermedia de traducción anticipada (TLB) (no mostrada) para almacenar en memoria caché las traducciones de direcciones virtuales/efectivas a físicas/reales. En una implementación, una memoria caché 438 almacena comandos y datos para acceso eficiente por los motores de

procesamiento gráfico 431-432, N. En una realización, los datos almacenados en la memoria caché 438 y las memorias gráficas 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 puede conseguirse a través del circuito intermediario 425 que toma parte en el mecanismo de coherencia de memoria caché en nombre de la memoria caché 438 y las memorias 433-434, N (por ejemplo, enviando actualizaciones a la memoria caché 438 relacionadas con modificaciones/accesos de líneas de memoria caché en las memorias caché de procesador 462A-462D, 456 y recibiendo actualizaciones desde la memoria caché 438).

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

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

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

Debido a que los recursos de hardware de los motores de procesamiento gráfico 431-432, N se correlacionan explícitamente con el espacio de direcciones reales vistas en el procesador de anfitrión 407, cualquier procesador de anfitrión puede direccionar 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 gráfico 431-432, N de modo que aparecen al sistema como unidades independientes.

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

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

La Figura 4C ilustra otra realización en la que el circuito de integración de acelerador 436 se integra dentro del procesador 407. En esta realización, los motores de procesamiento gráfico 431-432, N se comunican directamente a través del enlace de alta velocidad 440 al circuito de integración de acelerador 436 a través de la interfaz 437 y la interfaz 435 (que, de nuevo, puede utilizarse 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 en un caudal mayor dada su proximidad cercana al bus de coherencia 462 y las memorias caché 462A-462D, 426.

Una realización soporta diferentes modelos de programación que incluyen un modelo de programación de proceso especializado (sin virtualización de módulo de aceleración gráfica) y modelos de programación compartidos (con virtualización). Los últimos pueden incluir modelos de programación que se controlan por el circuito de integración de acelerador 436 y modelos de programación que se controlan por el módulo de aceleración gráfica 446.

En una realización del modelo de proceso especializado, los motores de procesamiento gráfico 431-432, N se dedican a una única aplicación o proceso en un único sistema operativo. La única aplicación puede canalizar otras peticiones de aplicación a los motores gráficos 431-432, N, proporcionando virtualización dentro de una VM/partición.

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

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

La Figura 4D ilustra un sector de integración de acelerador 490 ilustrativo. Como se usa en el presente documento, un "sector" comprende una porción 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 las invocaciones de GPU 481 desde las aplicaciones 480 ejecutadas en el procesador 407. Un elemento de proceso 483 contiene el estado de proceso para la correspondiente aplicación 480. Un descriptor de trabajo (WD) 484 contenido en el elemento de proceso 483 puede ser una única petición de trabajo 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 peticiones de trabajo en el espacio de direcciones 482 de la aplicación.

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

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

En la operación, una unidad de búsqueda de WD 491 en el sector de integración de acelerador 490 busca el siguiente WD 484 que incluye una indicación del trabajo que hay que hacer por uno de los motores de procesamiento gráfico del módulo de aceleración gráfica 446. Los datos del WD 484 pueden almacenarse en registros 445 y usarse por la MMU 439, circuito de gestión de interrupción 447 y/o circuito de gestión de contexto 446, como se ilustra. Por ejemplo, una realización de la MMU 439 incluye circuitería de recorrido de segmento/página para acceder a tablas de segmento/paginación 486 dentro del espacio de direcciones virtuales de SO 485. El circuito de gestión de interrupción 447 puede procesar eventos de interrupción 492 recibidos desde el módulo de aceleración gráfica 446. Cuando se realizan operaciones gráficas, una dirección efectiva 493 generada por un motor de procesamiento gráfico 431-432, N se traduce a una dirección real por la MMU 439.

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

Tabla 1 - Registros inicializados por hipervisor

1	Registro de control de sector
2	Puntero de área de procesos planificados de dirección real (RA)
3	Registro de anulación de máscara de autoridad
4	Desplazamiento de entrada de tabla de vector de interrupción
5	Límite de entrada de tabla de vector de interrupción

6	Registro de estados
7	ID de partición lógica
8	Puntero de grabación de utilización de acelerador de hipervisor de dirección real (RA)
9	Registro de descripción de almacenamiento

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

Tabla 2 - Registros inicializados por sistema operativo

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

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

La Figura 4E ilustra detalles adicionales para una realización de un modelo compartido. Esta realización incluye un espacio de direcciones 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 a través de un hipervisor 496 que virtualiza los motores de módulo de aceleración gráfica para el sistema operativo 495.

Los modelos de programación compartidos permiten que todos o un subconjunto de procesos de entre todas o un subconjunto de particiones en el sistema usen un módulo de aceleración gráfica 446. Existen dos modelos de programación en donde el módulo de aceleración gráfica 446 se comparte por múltiples procesos y particiones: uso compartido por sector de tiempo y uso compartido directo de gráficos.

En este modelo, el sistema hipervisor 496 es propietario del módulo de aceleración gráfica 446 y hace su función disponible para todos los sistemas operativos 495. Para que un módulo de aceleración gráfica 446 soporte virtualización por el sistema hipervisor 496, el módulo de aceleración gráfica 446 puede cumplir con los siguientes requisitos: 1) La petición de trabajo de una aplicación debe ser autónoma (es decir, el estado no necesita mantenerse entre trabajos), o el módulo de aceleración gráfica 446 debe proporcionar un mecanismo de guardado y restauración de contexto. 2) La petición de trabajo de una aplicación se garantiza por el módulo de aceleración gráfica 446 para completar en una cantidad especificada de tiempo, incluyendo cualquier fallo de traducción, o el módulo de aceleración gráfica 446 proporciona la capacidad de adelantar el procesamiento del trabajo. 3) El módulo de aceleración gráfica 446 debe garantizarse la equidad entre procesos cuando opera en el modelo de programación de uso compartido directo.

En una realización, para el modelo de uso compartido, se requiere que la aplicación 480 haga una llamada de sistema del sistema operativo 495 con un tipo de módulo de aceleración gráfica 446, un descriptor de trabajo (WD), un valor de registro de máscara de autoridad (AMR) y un puntero de área de guardado/restauración de contexto (CSR). El tipo de módulo de aceleración gráfica 446 describe la función de aceleración dirigida para la llamada de sistema. El tipo de módulo de aceleración gráfica 446 puede ser un valor específico del sistema. El WD se formatea específicamente para el módulo de aceleración gráfica 446 y puede ser en forma de un comando de módulo de aceleración gráfica 446, un puntero de dirección efectiva a una estructura definida por usuario, un puntero de dirección efectiva a una cola de comandos, o cualquier otra estructura de datos para describir el trabajo a hacer por el módulo de aceleración gráfica 446. En una realización, el valor de AMR es el estado de AMR a usar para el proceso actual. El valor pasado al sistema operativo es similar a una aplicación que establece el AMR. Si las implementaciones de circuito de integración de acelerador 436 y de módulo de aceleración gráfica 446 no soportan un Registro de Anulación de Máscara de Autoridad de Usuario (UAMOR), el sistema operativo puede aplicar el valor de UAMOR actual al valor de AMR antes de pasar el AMR en la llamada de hipervisor. El hipervisor 496 puede aplicar opcionalmente el valor de registro de anulación de máscara de autoridad (AMOR) actual antes de situar el AMR en el elemento de proceso 483. En una realización, el CSR 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 gráfica 446 guarde y restaure el estado de contexto. Este puntero es opcional si no se requiere ningún estado a guardar entre trabajos o cuando se adelanta un trabajo. El área de guardado/restauración de contexto puede anclarse en memoria de sistema.

Tras recibir la llamada de sistema, el sistema operativo 495 puede verificar que la aplicación 480 ha registrado y se le

ha dado la autoridad para usar el módulo de aceleración gráfica 446. El sistema operativo 495 llama, a continuación, al hipervisor 496 con la información mostrada en la Tabla 3.

Tabla 3 - Parámetros de llamada de SO a hipervisor

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

Tras recibir la llamada de hipervisor, el hipervisor 496 verifica que el sistema operativo 495 ha registrado y se le ha dado la autoridad de usar el módulo de aceleración gráfica 446. El hipervisor 496 pone, a continuación, el elemento de proceso 483 en la lista enlazada de elementos de proceso para el correspondiente tipo de módulo de aceleración gráfica 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 área de guardado/restauración de contexto (CSRP) de dirección efectiva (EA)
4	Un ID de proceso (PID) e ID de hilo opcional (TID)
5	Un puntero de grabación de utilización de acelerador (AURP) de dirección virtual (VA)
6	La dirección virtual del puntero de tabla de segmentos de almacenamiento (SSTP)
7	Un número de servicio de interrupción lógico (LISN)
8	Tabla de vector de interrupción, derivada a partir de los parámetros de llamada de hipervisor.
9	Un valor de registro de estados (SR)
10	Un ID de partición lógica (LPID)
11	Un puntero de grabación de utilización de acelerador de hipervisor de dirección real (RA)
12	El registro de descriptor de almacenamiento (SDR)

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

Como se ilustra en la Figura 4F, una realización de la invención emplea una memoria unificada direccionable a través de un espacio de direcciones virtuales de memoria común usado para acceder a las memorias de procesador físicas 401-402 y 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 virtuales/efectivas para acceder a las memorias de procesadores 401-402 y viceversa, simplificando de este modo la programabilidad. En una realización, una primera porción del espacio de direcciones virtuales/efectivas se asigna a la memoria de procesador 401, una segunda porción a la segunda memoria de procesador 402, una tercera porción a la memoria de GPU 420, y así sucesivamente. Todo el espacio de memoria virtual/efectiva (en ocasiones denominada como el espacio de direcciones efectivas) se distribuye de este modo a través de cada una de las memorias de procesador 401-402 y memorias de GPU 420-423, permitiendo que cualquier procesador o GPU acceda a cualquier memoria física con una dirección virtual correlacionada con esa memoria.

En una realización, la circuitería de gestión de polarización/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 de anfitrión (por ejemplo, 405) y las GPU 410-413 e implementa técnicas de polarización que indican las memorias físicas en las que deberían almacenarse ciertos tipos de datos. Mientras múltiples instancias de circuitería de gestión de polarización/coherencia 494A-494E se ilustran en la Figura 4F, la circuitería de polarización/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 conectada a GPU 420-423 se correlacione como parte de memoria de sistema, y se acceda usando tecnología de memoria virtual compartida (SVM), pero sin sufrir los inconvenientes de rendimiento típicos asociados con coherencia de memoria caché de sistema completa. La capacidad de que la memoria conectada a GPU 420-423 sea accedida como memoria de sistema sin tara de coherencia de memoria caché onerosa proporciona un entorno de operación beneficioso para descarga de GPU. Esta disposición permite que el software del

procesador de anfitrión 405 configure operandos y acceda a resultados de cálculo, sin la tara de copias de datos de DMA de E/S tradicionales. Tales copias tradicionales implican llamadas de controlador, interrupciones y accesos de E/S con correlación de memoria (MMIO) que son todas ineficientes en relación con accesos de memoria simples. Al mismo tiempo, la capacidad de acceder a la memoria conectada a GPU 420-423 sin taras de coherencia de memoria caché puede ser crítica al tiempo de ejecución de un cálculo descargado. En casos con tráfico de memoria de escritura de transmisión sustancial, por ejemplo, la tara de coherencia de memoria caché puede reducir significativamente el ancho de banda de escritura efectivo visto por una GPU 410-413. La eficiencia de configuración de operando, la eficiencia de acceso a resultados y la eficiencia de cálculo de GPU tienen todas una función en la determinación de la efectividad de la descarga de GPU.

En una implementación, la selección de entre la polarización de GPU y la polarización de procesador de anfitrión se acciona por una estructura de datos de seguimiento de polarización. Puede usarse una tabla de polarización, por ejemplo, que puede ser 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 polarización 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 polarización en la GPU 410-413 (por ejemplo, para almacenar en memoria caché entradas usadas frecuentemente/recientemente de la tabla de polarización). Como alternativa, toda la tabla de polarización puede mantenerse dentro de la GPU.

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

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

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

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

Canalización de procesamiento gráfico

La Figura 5 ilustra una canalización de procesamiento gráfico 500, de acuerdo con una realización. En una realización, un procesador gráfico puede implementar la canalización de procesamiento gráfico 500 ilustrada. El procesador gráfico puede incluirse 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 gráfico 500 a través de una o más instancias de la unidad de procesamiento paralelo (por ejemplo, la unidad de procesamiento paralelo 202 de la Figura 2) como se describe en el presente documento. Por ejemplo, una unidad de sombreador (por ejemplo, el multiprocesador gráfico 234 de la Figura 3) puede configurarse para realizar las funciones de uno o más de una unidad de procesamiento de vértices 504, una unidad de procesamiento de control de teselación 508, una unidad de procesamiento de evaluación de teselación 512, una unidad de procesamiento de geometría 516 y una unidad de procesamiento de fragmento/píxel 524. Las funciones de ensamblador de datos 502, ensambladores de primitivas 506, 514, 518, unidad de teselación 510, rasterizador 522 y unidad de operaciones de trama 526 también pueden realizarse por otros motores de procesamiento dentro de una agrupación de procesamiento (por ejemplo, la agrupación de procesamiento 214 de la Figura 3) y una correspondiente unidad de partición (por ejemplo, la unidad de partición 220A-220N de la Figura 2). La canalización de procesamiento gráfico 500 también puede implementarse usando unidades de procesamiento especializadas para una o más

funciones. En una realización, una o más porciones de la canalización de procesamiento gráfico 500 pueden realizarse mediante lógica de procesamiento paralelo dentro de un procesador de fin general (por ejemplo, CPU). En una realización, una o más porciones de la canalización de procesamiento gráfico 500 pueden acceder a memoria en chip (por ejemplo, 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 vértice para superficies y primitivas. El ensamblador de datos 502 emite, a continuación, los datos de vértice, incluyendo los atributos de vértice, a la unidad de procesamiento de vértices 504. La unidad de procesamiento de vértices 504 es una unidad de ejecución programable que ejecuta programas de sombreador de vértices, iluminando y transformando datos de vértice como se especifica por los programas de sombreador de vértices. La unidad de procesamiento de vértices 504 lee datos que se almacenan en memoria caché, memoria local o de sistema para su uso en el procesamiento de los datos de vértice y puede programarse para transformar los datos de vértice desde una representación de coordenadas basada en objeto a un espacio de coordenadas de espacio global o un espacio de coordenadas de dispositivo normalizado.

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

La unidad de procesamiento de control de teselación 508 trata los vértices introducidos como puntos de control para un parche geométrico. Los puntos de control se transforman desde una representación introducida desde el parche (por ejemplo, las bases del parche) a una representación que es adecuada para su uso en evaluación de superficie por la unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de control de teselación 508 también puede calcular factores de teselación para bordes de parches geométricos. Un factor de teselación se aplica a un único borde y cuantifica un nivel dependiente de vista de detalle asociado con el borde. Una unidad de teselación 510 está configurada para recibir los factores de teselación para bordes de un parche y para teselar el parche en múltiples primitivas geométricas tales como primitivas de línea, de triángulo o cuadriláteros, que se transmiten a una unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de evaluación de teselación 512 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 vértice desde la unidad de procesamiento de evaluación de teselación 512, leyendo atributos de vértice almacenados según se necesite, y construye primitivas geométricas para su procesamiento por la unidad de procesamiento de geometría 516. La unidad de procesamiento de geometría 516 es una unidad de ejecución programable que ejecuta programas de sombreador de geometría para transformar primitivas geométricas recibidas desde el ensamblador de primitivas 514 como se especifica por los programas de sombreador de geometría. En una realización, la unidad de procesamiento de geometría 516 se programa para subdividir las primitivas geométricas en una o más nuevas primitivas geométricas y calcular parámetros usados para rasterizar las nuevas primitivas geométricas.

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

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

La unidad de operaciones de trama 526 es una unidad de procesamiento que realiza operaciones de trama que incluye, pero sin limitación a, patrón, prueba z, combinación y similares, y emite datos de píxel como datos de gráficos procesados que hay que almacenar en memoria gráfica (por ejemplo, memoria de procesador paralelo 222 como en la Figura 2, y/o memoria de sistema 104 como en la Figura 1, que hay que visualizar en el uno o más dispositivo o dispositivos de visualización 110 o para su procesamiento adicional por uno del uno o más procesador o procesadores 102 o procesador o procesadores paralelos 112. En algunas realizaciones, la unidad de operaciones de trama 526 está configurada para comprimir datos z o de color que se escriben en la memoria y descomprimir datos z o de color que se leen de la memoria.

Representación de plano de imagen multirresolución dentro de una microarquitectura de GPU mejorada

Pueden obtenerse ahorros de recursos de procesamiento significativos si se usan planos de imagen de múltiples resoluciones en los que se utilizan imágenes de mayor resolución, y correspondientes recursos de cálculo mayores, para los objetos gráficos más importantes. Los objetos gráficos menos importantes pueden representarse en un proceso menos costoso que utilizan imágenes de menor resolución. Las imágenes resultantes pueden combinarse, a continuación, en una única imagen de alta resolución en un paso final del procesamiento que genera la imagen de salida. La tecnología descrita en el presente documento puede superar las deficiencias de sistemas convencionales implementando una representación de plano de imagen multirresolución como se divulga a continuación.

La Figura 6 es un diagrama conceptual de planos de imagen representativos que tiene múltiples resoluciones que contienen objetos gráficos que hay que procesar por la canalización de procesamiento gráfico, de acuerdo con una realización. Cuando el procesamiento gráfico crea una imagen de salida, los objetos gráficos se representan en una escena cuando se dibujan los objetos, y se aplican sombreado y teselación. Cuando se produce el proceso de representación, los recursos de procesamiento implicados pueden relacionarse con la resolución de una memoria intermedia de imagen en la que se representan los objetos. Para incluir más detalle para cualquier objeto gráfico dado, la memoria intermedia de imagen en la que el objeto se representa puede requerir más píxeles y, posteriormente, más procesamiento.

No todos los objetos pueden requerir el mismo nivel de detalle para obtener un resultado deseado. Habitualmente, se generan imágenes para incluir un plano de imagen de punto focal 602 en el que se representan objetos en el mayor nivel de detalle. El plano de imagen de punto focal 602 representa un plano en la escena que es perpendicular a la dirección de visión para la escena en una profundidad en el eje z en el que los objetos son de máximo interés y en el que los objetos son los más enfocados.

Una profundidad de campo representa un intervalo de distancias a lo largo del eje z tanto en frente como detrás del plano de imagen de punto focal 602 en el que todos los objetos gráficos son también de máximo interés y, por lo tanto, también son los más enfocados. Como tal, los objetos gráficos ubicados en frente y detrás del plano de imagen de punto focal 602 que están dentro de la profundidad de campo también pueden representarse con el mismo nivel alto de resolución.

Los objetos gráficos de primer plano que se ubican en frente del plano de imagen de punto focal 602 pueden representarse en un plano de imagen de resolución media 603. Los objetos en el plano de imagen de resolución media 603 pueden representarse usando una resolución inferior ya que los objetos habitualmente no son tan detallados y/o están enfocados. Como tal, los objetos gráficos representados en el plano de imagen de resolución media 603 utilizarían menos recursos durante la representación que los objetos en el plano de imagen de punto focal 602.

De manera similar, los objetos gráficos ubicados detrás del plano de imagen de punto focal 602 en una profundidad a lo largo del eje z que está más allá de la profundidad de campo se representan en un plano de imagen de fondo 604 que también utiliza una resolución inferior para objetos gráficos se representan con detalle reducido. Una vez más, el uso de la menor resolución puede reducir los recursos computacionales implicados en la generación de la imagen representada.

Un experto en la materia reconocerá que la realización de ejemplo de la Figura 6 puede incluir más de un plano de imagen para el plano de imagen de resolución media 603 y el plano de imagen de fondo 604 que utilizan múltiples valores de resolución para generar una profundidad más detallada de campo sin desviarse de la presente invención. Adicionalmente, el plano de imagen de resolución media 603 y el plano de imagen de fondo 601 pueden utilizar diferentes resoluciones que se reducen desde la mayor resolución usada por el plano de imagen de punto focal 602.

Pasando ahora a la Figura 7A, un plano de imagen de interfaz de usuario (UI) 601 también puede incluirse dentro de un sistema de representación de plano de imagen multirresolución para permitir que objetos gráficos que habitualmente contienen objetos basados en texto se representen con una mayor resolución. En muchas aplicaciones tales como videojuegos, se visualiza una interfaz de usuario a lo largo de uno o más bordes de la escena visualizada como se muestra en la Figura 7A. El plano de imagen de UI 601 puede contener uno o más objetos de UI 701-704 para proporcionar artículos útiles para comunicar información al observador. Estos objetos de UI 701-704 pueden incluir objetos que incluyen imágenes y GIF 701, texto de estado 702, elementos de navegación 703 y texto de mensaje

704 entre otras posibilidades. El plano de imagen de UI 604 requiere una resolución suficiente para visualizar datos de texto al observador donde el texto es legible. Habitualmente, el plano de imagen de UI 601 utiliza la resolución más alta similar al plano de imagen de punto focal 602 para permitir que los objetos de UI 701-704 se detallen tanto como sea posible. Un experto en la materia reconocerá que el plano de imagen de UI 601 puede usar una resolución menor que el plano de imagen de punto focal 602 que permite que se produzca el procesamiento usando menos recursos siempre que los objetos de UI 701-704 representados se detallen de forma suficiente para comunicar información de forma precisa al observador. En algunas realizaciones en las que estos elementos de UI tienen un tamaño fijo y ubicación fija de tal forma que la única región de la escena que contiene estos elementos de UI, el plano de imagen de UI puede necesitar ubicaciones de memoria para las regiones de la imagen de salida que contiene los elementos de UI 701-704. En este ejemplo, los restantes planos de imagen 602-604 también pueden no asignar memoria para las regiones de la imagen de salida en la que se ubican los elementos de UI 701-704 ya que artículos dentro de la escena visualizada no se incluirán dentro de la imagen de salida 901 si se ubican detrás de los elementos de UI 701-704.

La Figura 7B es un diagrama conceptual de otro plano de imagen representativo que contiene elementos gráficos de plano de punto focal que hay que procesar por la canalización de procesamiento gráfico, de acuerdo con una realización. En el ejemplo ilustrado, se representan múltiples objetos gráficos 721-723 en el plano de imagen de punto focal 602 en el que el objeto posee sus propias características con respecto a sombreado y cualquier otro detalle visual. En esta realización de ejemplo, estos objetos gráficos 801-803 se representan usando la resolución más alta. Los objetos gráficos representados en el plano de imagen de fondo 604, el plano de imagen de resolución media 603 y el plano de imagen de UI 601 son similares excepto que se representan en sus respectivos planos de imagen en la correspondiente resolución de imagen. Todos los objetos gráficos 801-803 pueden representarse en el plano de imagen apropiado con la resolución correcta como parte del proceso de representación.

La Figura 8 es un diagrama conceptual de un plano de imagen combinado que contiene los elementos gráficos de cada plano de imagen con una resolución alta común, de acuerdo con una realización. Una vez que todos los objetos gráficos 701-704, 721-723 se representan en el plano de imagen apropiado 601-604, los múltiples planos de imagen 601-604 se combinan en una única imagen de salida 801. Las imágenes de menor resolución tales como el plano de imagen de fondo 604 y el plano de imagen de resolución media 603 se remuestrean para corresponder a una resolución que coincide con la resolución del plano de imagen de punto focal 602. Los cuatro planos de imagen 601-604 se combinan, a continuación, para generar la imagen de salida 801. La combinación de estos cuatro planos de imagen puede usar una selección del valor del plano más frontal de los cuatro planes cuando un objeto existen en ese plano o puede usar un método combinado para combinar los valores para múltiples objetos en dos o más planes para una ubicación de píxel dada para tener en cuenta la opacidad de objetos ubicados en frente de otros objetos. La solución ilustrada puede usarse dentro de otros procesos de apilamiento de planos de imagen conocidos ahora o desarrollados en el futuro.

La Figura 9 ilustra un sistema informático 900 (por ejemplo, servidor, ordenador de sobremesa, ordenador de tableta, tableta convertible, teléfono inteligente, dispositivo de internet móvil/MID, consola de juegos, reproductor de medios, ordenador ponible, etc.) en el que un representador 912 de una canalización 3D 901 obtiene objetos gráficos asociados con una escena 3D desde una o más etapas aguas arriba 911 y/o una interfaz de datos 902 tal como, por ejemplo, un controlador de red, controlador de memoria, bus, etc. El representador 912 ilustrado determina en cuál de una pluralidad de planos de imagen multirresolución tiene que representarse el objeto gráfico recibido usando ubicación de objeto a lo largo del eje z de la escena.

El representador 912 puede representar el objeto gráfico recibido en el plano de imagen determinado en su correspondiente resolución de imagen, en donde todos los objetos gráficos recibidos se procesan en sus respectivos planos de imagen. Una vez que se representan todos los objetos gráficos, un remuestreador 913 puede leer y remuestrear todos los planos de imagen que no tienen la resolución más alta y pasar los valores de píxeles a un rasterizador para combinar en una imagen de salida. El remuestreador 913 y el rasterizador 914 pueden procesar cada píxel en los diversos planos de imagen para el mismo píxel ubican para generar completamente la imagen de salida 801 como se ha analizado anteriormente en referencia con la Figura 8. Puede pasarse una imagen de salida completada a través de cualquier parte aguas abajo 915 de la canalización de procesamiento gráfico 3D 901 para su transmisión a un dispositivo de visualización 903.

La Figura 10 es un diagrama de flujo de operación de un procesador gráfico para representar tareas de objetos gráficos con diversas resoluciones en memorias intermedias de imagen separadas de una canalización de procesamiento gráfico, de acuerdo con una realización. El método ilustrado puede implementarse como uno o más módulos en un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento legible por máquina o legible por ordenador no transitorio tal como memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), ROM programable (PROM), firmware, memoria flash, etc., en lógica configurable tal como, por ejemplo, matrices lógicas programables (PLA), matrices de puertas programables en campo (FPGA), dispositivos lógicos programables complejos (CPLD), en lógica de hardware de funcionalidad fija usando tecnología de circuito tal como, por ejemplo, circuito integrado específico de la aplicación (ASIC), semiconductor complementario de óxido metálico (CMOS) o tecnología de lógica de transistor a transistor (TTL) o cualquier combinación de los mismos.

Por ejemplo, código de programa informático para efectuar operaciones mostradas en el método ilustrado puede escribirse en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado de objeto tal como JAVA, SMALLTALK, C++ o similar y lenguajes de programación procedurales convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares. Adicionalmente, instrucciones lógicas podrían incluir instrucciones de ensamblador, instrucciones de arquitectura de conjunto de instrucciones (ISA), instrucciones de máquina, instrucciones dependientes de máquina, microcódigo, datos de establecimiento de estado, datos de configuración para circuitería integrada, información de estado que personaliza circuitería electrónica y/u otros componentes estructurales que son nativos a hardware (por ejemplo, procesador anfitrión, unidad central de procesamiento/CPU, microcontrolador, etc.).

Continuando haciendo referencia a las Figuras 6, 8, 9 y 10, el procesamiento ilustrado comienza con una instrucción a una canalización 3D que solicita la creación de una representación/rasterización para generar una imagen de salida 801 en el bloque 1001. Esta instrucción puede incluir un número de planos de imagen que hay que usar, una identificación el plano de imagen de punto focal 602 y su ubicación a lo largo del eje z de la escena que hay que representar, la resolución más alta que hay que usar para el plano de imagen de punto focal 602, identidad, ubicación de eje z, y resolución para todos los demás planos de imagen, y una descripción de cualquier elemento en un plano de imagen de UI 601.

Una vez que se inicializan estos planos de imagen 601-604, el objeto gráfico se recibe por el representador 912 en el bloque 1002. El representador 912 usa una ubicación de eje z en el bloque 1003 para determinar en cuál de los planos de imagen multiresolución 601-604 tiene que representarse el objeto gráfico recibido usando ubicación de objeto a lo largo del eje z de la escena. El representador 912 representa el objeto gráfico recibido en el plano de imagen determinado con su correspondiente resolución de imagen en el bloque 1004.

El bloque de decisión 1005 determina si todos los objetos gráficos que hay que incluir en la imagen de salida 801 se han procesado por el representador 912. Si no se han procesado todos los objetos gráficos, el procesamiento vuelve al bloque 1002 para procesar el siguiente objeto. Una vez que se procesan todos los objetos, el bloque de decisión 1005 pasó el procesamiento al bloque 1012.

El remuestreador 912 recupera los datos de lugar de imagen de todos los planos de imagen 601-604 en el bloque 1011 y remuestrea todas las imágenes de menor resolución para reenviar al rasterizador 914 valores de todos los planos de imagen 601-604 en cada ubicación en la imagen de salida 901 para su combinación en el bloque 1013. El bloque de decisión 1014 determina si se han procesado todas las ubicaciones de píxel dentro de la imagen de salida 901, y si no vuelve el procesamiento al bloque 1011 para continuar procesando las ubicaciones de píxel. Cuando se han procesado todos los píxeles, el bloque de decisión 1013 pasa el control al bloque 1020 que transmite la imagen de salida a la siguiente etapa en la canalización de procesamiento 3D 901 para su emisión a un dispositivo de visualización 903.

Visión general del sistema de interfaz integrada montado en la cabeza

La Figura 11 muestra un sistema de pantalla montada en la cabeza (HMD) 1100 que lleva puesto un usuario mientras experimenta un entorno inmersivo tal como, por ejemplo, un entorno de realidad virtual (VR), un entorno de realidad aumentada (AR), un juego tridimensional (3D) multijugador y así sucesivamente. En el ejemplo ilustrado, una o más correas 1120 sujetan un armazón 1102 del sistema de HMD 1100 en frente de los ojos del usuario. Por consiguiente, un visualizador de ojo izquierdo 1104 puede colocarse para visualizarse por el ojo izquierdo del usuario y un visualizador de ojo derecho 1106 puede colocarse para visualizarse por el ojo derecho del usuario. El visualizador de ojo izquierdo 1104 y el visualizador de ojo derecho 1106 pueden integrarse como alternativa en un único visualizador en ciertos ejemplos tales como, por ejemplo, un teléfono inteligente que lleva puesto el usuario. En el caso de AR, los visualizadores 1104, 1106 pueden ser visualizadores de visión a través que permiten que el usuario vea los alrededores físicos, con otro contenido representado (por ejemplo, caracteres virtuales, anotaciones de información, visualizador frontal/HUD) que se presenta encima de una fuente en vivo de los alrededores físicos.

En un ejemplo, el armazón 1102 incluye una cámara de vista hacia abajo izquierda 1108 para capturar imágenes de un área generalmente en frente del usuario y por debajo del ojo izquierdo (por ejemplo, gestos de la mano izquierda). Adicionalmente, una cámara de vista hacia abajo derecha 1110 puede capturar imágenes de un área generalmente en frente del usuario y por debajo del ojo derecho (por ejemplo, gestos de la mano derecha). El armazón 1102 ilustrado también incluye una cámara de vista frontal izquierda 1112 y una cámara de vista frontal derecha 1114 para capturar imágenes en frente de los ojos izquierdo y derecho, respectivamente, del usuario. El armazón 1102 también puede incluir una cámara de vista lateral izquierda 1116 para capturar imágenes de un área a la izquierda del usuario y una cámara de vista lateral derecha 1118 para capturar imágenes de un área a la derecha del usuario.

Las imágenes capturadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118, que pueden tener campos de visión solapantes, pueden usarse para detectar gestos hechos por el usuario, así como para analizar y/o reproducir el entorno externo en los visualizadores 1104, 1106. En un ejemplo, los gestos detectados se usan por una arquitectura de procesamiento gráfico (por ejemplo, interno y/o externo) para representar y/o controlar una representación virtual del usuario en un juego 3D. De hecho, los campos de visión solapantes pueden habilitar la captura de gestos hechos por

otros individuos (por ejemplo, en un juego multijugador), en donde los gestos de otros individuos pueden usarse adicionalmente para representar/controlar la experiencia inmersiva. Los campos de visión solapantes también pueden habilitar que el sistema de HDM 1100 para detecte automáticamente obstrucciones u otros peligros cerca del usuario. Un enfoque de este tipo puede ser particularmente ventajoso en aplicaciones de sistema de asistencia al conductor avanzado (ADAS).

En un ejemplo, proporcionar la cámara de vista hacia abajo izquierda 1108 y la cámara de vista hacia abajo derecha 1110 con campos de visión solapantes proporciona una visión estereoscópica que tiene una resolución incrementada. La resolución incrementada puede habilitar, a su vez, que los movimientos de usuario muy similares se distingan de otro (por ejemplo, con precisión submilimétrica). El resultado puede ser un rendimiento mejorado del sistema de HDM 1100 con respecto a la fiabilidad. De hecho, la solución ilustrada puede ser útil en una amplia variedad de aplicaciones tales como, por ejemplo, coloración de información en ajustes de AR, intercambio de herramientas/dispositivos virtuales entre usuarios en un entorno multiusuario, representación de artículos virtuales (por ejemplo, armas, espadas, bastones) y así sucesivamente. También pueden detectarse y usarse gestos de otros objetos, extremidades y/o partes del cuerpo para representar/controlar el entorno virtual. Por ejemplo, señales mielográficas, señales electroencefalográficas, seguimiento ocular, respiración o jadeo, movimientos de la mano, etc., pueden rastrearse en tiempo real, ya sea desde el portador u otro individuo en un entorno compartido. Las imágenes capturadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118, también pueden servir como una entrada contextual. Por ejemplo, podría determinarse que el usuario está indicando una palabra particular para editar o tecla para presionar en una aplicación de procesamiento de textos, un arma particular a desplegar o una dirección de viaje en un juego y así sucesivamente.

Adicionalmente, las imágenes capturadas por las cámaras 1108, 1110, 1112, 1114, 1116, 1118 pueden usarse para llevar a cabo comunicación compartida o interactividad en red en una operación de equipo, entrenamiento médico y/o aplicaciones de guiado remoto/teleoperación. Bibliotecas de gestos específicos o aprendizaje automático de red neuronal podrían habilitar la identificación de herramienta y realimentación para una tarea. Por ejemplo, puede habilitarse una herramienta virtual que traduce a acciones remotas reales. En otro ejemplo más, el sistema de HDM 1100 traduce la manipulación de un taladro virtual dentro de una escena virtual a la operación remota de un taladro en un dispositivo robótico desplegado para buscar en un edificio derrumbado. Además, el sistema de HDM 1100 puede ser programable en la medida que incluye, por ejemplo, un protocolo que habilita que el usuario añada un nuevo gesto a una lista de gestos identificables asociados con acciones de usuario.

Además, las diversas cámaras en el HMD 1100 pueden ser configurables para detectar frecuencias de espectro además de las longitudes de onda visibles del espectro. Las capacidades de formación de imágenes multiespectral en las cámaras de entrada permiten el seguimiento de la posición del usuario y/u objetos eliminando características de imagen no esenciales (por ejemplo, ruido de fondo). Por ejemplo, en aplicaciones de realidad aumentada (AR) tales como cirugía, instrumentos y equipo pueden seguirse por su reflectividad infrarroja sin la necesidades de ayudas de seguimiento adicionales. Además, el HMD 1100 podría emplearse en situaciones de baja visibilidad en las que una "fuente en vivo" de las diversas cámaras podría mejorarse o aumentarse a través de análisis informático y visualizarse al usuario como indicios visuales o de audio.

El sistema de HDM 1100 también puede renunciar a realizar cualquier tipo de comunicación de datos con un sistema informático remoto o necesitar cables de potencia (por ejemplo, modo independiente de operación). En este sentido, el sistema de HDM 1100 puede ser un dispositivo "sin cables" que tiene una unidad de potencia que habilita que el sistema de HDM 1100 opere independientemente de sistemas de potencia externos. Por consiguiente, el usuario podría jugar un juego con características completas sin estar conectado a otro dispositivo (por ejemplo, consola de juegos) o fuente de alimentación. En un ejemplo de procesamiento de textos, el sistema de HDM 1100 podría presentar un teclado virtual y/o ratón virtual en los visualizadores 1104 y 1106 para proporcionar un escritorio virtual o 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 de usuario en el teclado virtual o movimientos del ratón virtual. Ventajas incluyen, pero sin limitación, facilidad de portabilidad y privacidad del escritorio virtual de individuos cercanos. La arquitectura de procesamiento gráfico subyacente puede soportar la compresión y/o descompresión de señales de vídeo y audio. Además, proporcionar imágenes separadas al ojo izquierdo y al ojo derecho del usuario puede facilitar la representación, generación y/o percepción de escenas en 3D. Las posiciones relativas del visualizador de ojo izquierdo 1104 y el visualizador de ojo derecho 1106 también pueden ser ajustables para coincidir con variaciones en la separación de los ojos entre diferentes usuarios.

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

Componentes funcionales del sistema de HDM

La Figura 12 muestra el sistema de HDM en mayor detalle. En el ejemplo ilustrado, el armazón 1102 incluye una unidad de potencia 1200 (por ejemplo, potencia de batería, adaptador) para proporcionar potencia al sistema de HDM. El armazón 1102 ilustrado también incluye un módulo de seguimiento de movimiento 1220 (por ejemplo, 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 se acopla a un puente de E/S 1206. El puente de E/S 1206 puede habilitar 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 sonido que puede procesarse para reconocer comandos de voz del usuario, así como de 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 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 entregar sonido a los oídos del usuario y/u otros individuos cercanos. Los módulos de salida de audio 1208, que pueden ser en forma de auriculares, altavoces en la oreja, altavoces sobre las orejas, altavoces, etc. o cualquier combinación de los mismos, pueden entregar contenido de audio estéreo y/o 3D al usuario (por ejemplo, localización espacial). El armazón 1102 ilustrado también incluye un módulo inalámbrico 1222, que puede facilitar las comunicaciones entre el sistema de HDM y diversos otros sistemas (por ejemplo, ordenadores, dispositivos ponibles, consolas de juegos). 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 1207 ilustrado 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 puede obtenerse del sistema de procesador 1204 a través del puente de E/S 1206. Las cámaras de entrada 1202 pueden incluir la cámara de vista lateral izquierda 1116, la cámara de vista lateral derecha 1118, la cámara de vista hacia abajo izquierda 1108, la cámara de vista frontal izquierda 1112, la cámara de vista frontal derecha 1114 y la cámara de vista hacia abajo derecha 1110, ya analizadas.

Volviendo ahora a la Figura 13, se muestra una agrupación de procesamiento general (GPC) 1300. La GPC 1300 ilustrada puede incorporarse en un sistema de procesamiento tal como, por ejemplo, el sistema de procesador 1204 (Figura 12), ya analizado. La GPC 1300 puede incluir un gestor de canalización 1302 que se comunica con un planificador. En un ejemplo, el gestor de canalización 1302 recibe tareas del planificador y distribuye las tareas a uno o más multiprocesadores de transmisión (SM) 1304. Cada SM 1304 puede configurarse para procesar grupos de hilos, en donde un grupo de hilos puede considerarse una pluralidad de hilos relacionados que ejecutan las mismas o similares operaciones en diferentes datos de entrada. Por lo tanto, cada hilo en el grupo de hilos puede asignarse 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 canalización 1302 también puede especificar destinos de datos procesados a una barra cruzada de distribución de trabajo 1308, que se comunica con una barra cruzada de memoria.

Por lo tanto, a medida que cada SM 1304 transmite una tarea procesada a la barra cruzada de distribución de trabajo 1308, la tarea procesada puede proporcionarse a otra GPC 1300 para su procesamiento adicional. La salida del SM 1304 también puede enviarse a una unidad de operaciones de pretrama (preROP) 1314, que a su vez dirige datos a una o más unidades de operaciones de trama, o realiza otras operaciones (por ejemplo, realizar traducciones de direcciones, organizar datos de color de instantáneas, combinar 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 correlacionar direcciones virtuales con direcciones físicas. En este sentido, la MMU 1310 puede incluir entradas de tabla de paginación (PTE) que se usan para correlacionar direcciones virtuales con direcciones físicas de una losa, página de memoria y/o índice de línea de memoria caché. La GPC 1300 ilustrada también incluye una unidad de textura 1312.

Arquitectura de canalización de gráficos

Pasando 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 vértice asociados con servicios de orden alto, primitivas de gráficos, triángulos, etc., y transmitir los datos de vértice a una unidad de búsqueda de atributos de vértice (VAF) 1404. La VAF 1404 puede recuperar atributos de vértice asociados con cada uno de los vértices entrantes de la memoria compartida y almacenar los datos de vértice, junto con los atributos de vértice asociados, en la memoria compartida.

La canalización de espacio global 1420 ilustrada también incluye una unidad de procesamiento de vértice, 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 topologías, 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 sombreador de vértices. Los programas ejecutados por la VTG 1406 pueden procesar los datos de vértice y atributos de vértice recibidos desde 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 gráfico 1400.

La unidad de procesamiento de vértices de la VTG 1406 puede ser una unidad de ejecución programable que ejecuta programas de sombreador de vértices, iluminando y transformando datos de vértice como se especifica por los programas de sombreador de vértices. Por ejemplo, la unidad de procesamiento de vértices podría programarse para transformar los datos de vértice desde una representación de coordenadas basada en objeto (por ejemplo, espacio de objeto) a un sistema basado en coordenadas como alternativa tal como espacio global o espacio de coordenadas de dispositivo normalizado (NDC). Adicionalmente, la unidad de procesamiento de vértices puede leer datos de vértice y atributos de vértice que se almacenan en memoria compartida por la VAF 1404 y procesar los datos de vértice y atributos de vértice. En un ejemplo, la unidad de procesamiento de vértices almacena vértices procesados en la memoria compartida.

La unidad de procesamiento de inicialización de teselación (por ejemplo, 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 vértices producidos por la unidad de procesamiento de vértices y genera primitivas de gráficos en ocasiones denominadas "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 la 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 etapas posteriores en la canalización de procesamiento gráfico 1400.

Las tareas producidas por la unidad de generación de tareas pueden redistribuirse 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 gráfico 1400 y otra. Por consiguiente, el distribuidor de tareas puede redistribuir estas tareas de tal forma que cada canalización de procesamiento gráfico 1400 tiene aproximadamente la misma carga de trabajo durante etapas de canalización posteriores.

Como ya se ha indicado, la VTG 1406 también puede incluir una unidad de generación de topologías. En un ejemplo, la unidad de generación de topologías recupera tareas distribuidas por el distribuidor de tareas, indexa los vértices, incluyendo vértices asociados con parches, y calcula coordenadas (UV) para vértices de teselación y los índices que conectan los vértices de teselación para formar primitivas de gráficos. Los vértices indexados pueden almacenarse por la unidad de generación de topologías en la memoria compartida. La unidad de procesamiento de teselación de la VTG 1406 puede configurarse para ejecutar programas de sombreador de teselación (por ejemplo, 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 pueden pasarse desde la memoria compartida a la unidad de procesamiento de geometría (por ejemplo, la siguiente etapa de 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 (por ejemplo, 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 nuevas primitivas de gráficos. La unidad de procesamiento de geometría también puede calcular parámetros tales como, por ejemplo, coeficientes de ecuación planos, que pueden usarse para rasterizar las nuevas primitivas de gráficos.

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

La canalización de procesamiento gráfico 1400 también puede incluir una unidad de mosaico 1410 acoplada a la canalización de espacio global 1420. La unidad de mosaico 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 se transmiten, a continuación, a la unidad de mosaico 1410. En este sentido, la canalización de procesamiento gráfico 1400 también puede incluir una canalización de espacio de pantalla 1422, en donde el espacio de pantalla puede dividirse en losas de memoria caché. Cada losa de memoria caché puede asociarse, por lo tanto, con una porción del espacio de

pantalla. Para cada primitiva de gráficos, la unidad de mosaico 1410 puede identificar el conjunto de losas de memoria caché que se cruzan con la primitiva de gráficos (por ejemplo, "mosaico"). Después de dividir en mosaico un número de primitivas de gráficos, la unidad de mosaico 1410 puede procesar las primitivas de gráficos sobre una base de losa de memoria caché. En un ejemplo, las primitivas de gráficos asociadas con una losa de memoria caché particular se transmiten a una unidad de establecimiento 1412 en la canalización de espacio de pantalla 1422, una losa cada vez. Las primitivas de gráficos que se cruzan con múltiples losas de memoria caché pueden procesarse una vez en la canalización de espacio global 1420, mientras que se transmiten múltiples veces a la canalización de espacio de pantalla 1422.

En un ejemplo, la unidad de establecimiento 1412 recibe datos de vértice desde la VPC 1408 a través de la unidad de mosaico 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 planos de profundidad. La canalización de espacio de pantalla 1422 también puede incluir un rasterizador 1414 acoplado a la unidad de establecimiento 1412. El rasterizador puede convertir con exploración 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 en Z y otras optimizaciones basadas en Z.

La PS 1416, que puede acceder a memoria compartida, puede ejecutar programas de sombreador de fragmentos que transforman fragmentos recibidos desde el rasterizador 1414. Más particularmente, los programas de sombreador de fragmentos pueden sombrear fragmentos con granularidad a nivel de píxel (por ejemplo, funcionando como programas de sombreador de píxeles). En otro ejemplo, los programas de sombreador de fragmentos sombrean fragmentos con granularidad a nivel de muestra, en los que cada píxel incluye múltiples muestras, y cada muestra representa una porción de un píxel. Además, los programas de sombreador de fragmentos pueden sombrear fragmentos con cualquier otra granularidad, dependiendo de las circunstancias (por ejemplo, tasa de muestreo). La PS 1416 puede realizar combinación, sombreado, corrección de perspectiva, correlación de texturas, etc., para generar fragmentos sombreados.

La canalización de espacio de pantalla 1422 ilustrada también incluye una unidad de operaciones de trama (ROP) 1418, que puede realizar operaciones de trama tales como, por ejemplo, patrón, prueba z, combinación y así sucesivamente. La ROP 1418 puede transmitir, a continuación, datos de píxel como datos gráficos procesados a uno o más objetivos representados (por ejemplo, memoria gráfica). La ROP 1418 puede configurarse 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 gráfico 1400 puede implementarse por uno o más elementos de procesamiento. Por ejemplo, la VTG 1406 y/o la PS 1416 pueden implementarse en uno o más SM, el PD 1402, la VAF 1404, la VPC 1408, la unidad de mosaico 1410, la unidad de establecimiento 1412, el rasterizador 1414 y/o la ROP 1418 podrían implementarse en elementos de procesamiento de una GPC particular en conjunto con una correspondiente unidad de partición. La canalización de procesamiento gráfico 1400 también puede implementarse en lógica de hardware de funcionalidad fija. De hecho, la canalización de procesamiento gráfico 1400 puede implementarse en una PPU.

Por lo tanto, la canalización de espacio global 1420 ilustrada procesa objetos gráficos en espacio 3D, en el que la posición de cada objeto gráfico se conoce en relación con objetos gráficos y en relación con un sistema de coordenadas 3D. En contraste, la canalización de espacio de pantalla 1422 puede procesar objetos gráficos que se han proyectado desde el sistema de coordenadas 3D a una superficie plana 2D que representa la superficie del dispositivo de visualización. Adicionalmente, la canalización de espacio global 1420 puede dividirse 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 a la unidad de generación de tareas. La canalización de fase beta podría incluir etapas de canalización desde la unidad de generación de topologías a la VPC 1408. En un caso de este tipo, la canalización de procesamiento gráfico 1400 puede realizar un primer conjunto de operaciones (por ejemplo, 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 (por ejemplo, un único hilo, un grupo de hilos, múltiples grupos de hilos que actúan al unísono) en la canalización de fase beta.

Si están en uso múltiples canalizaciones de procesamiento gráfico 1400, los datos de vértice y atributos de vértice asociados con un conjunto de objetos gráficos pueden dividirse de modo que cada canalización de procesamiento gráfico 1400 tiene una carga de trabajo similar a través de la fase alfa. Por consiguiente, el procesamiento de fase alfa puede expandir sustancialmente la cantidad de datos de vértice y atributos de vértice, de tal forma que la cantidad de datos de vértice y atributos de vértice producidos por la unidad de generación de tareas es significativamente mayor que la cantidad de datos de vértice y atributos de vértice procesados por el PD 1402 y la VAF 1404. Además, las unidades de generación de tareas asociadas con diferentes canalizaciones de procesamiento gráfico 1400 pueden producir datos de vértice y atributos de vértice que tienen diferentes niveles de calidad, incluso cuando se comienza 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 gráfico 1400 tiene aproximadamente la misma carga de trabajo en el comienzo de la canalización de fase beta.

Pasando ahora a la Figura 15, se muestra un multiprocesador de transmisión (SM) 1500. El SM 1500 ilustrado incluye K unidades de planificador 1504 acopladas a una memoria caché de instrucciones 1502, en donde cada unidad de planificador 1504 recibe una matriz de bloque de hilos desde un gestor de canalización (no mostrado) y gestiona la planificación de instrucciones para uno o más bloques de hilo de cada matriz de bloque de hilos activa. La unidad de planificador 1504 puede planificar hilos para su ejecución en grupos de hilos paralelos, donde cada grupo puede denominarse como un "salto". Por lo tanto, cada salto podría incluir, por ejemplo, sesenta y cuatro hilos. Adicionalmente, la unidad de planificador 1504 puede gestionar una pluralidad de diferentes bloques de hilo, asignando los bloques de hilo a saltos para su ejecución. La unidad de planificador puede planificar, a continuación, instrucciones desde la pluralidad de diferentes saltos en diversas unidades funcionales durante cada ciclo de reloj. Cada unidad de planificador 1504 puede incluir una o más unidades de distribución de instrucciones 1522, en donde cada unidad de distribución 1522 transmite instrucciones a una o más de las unidades funcionales. El número de unidades de distribución 1522 puede variar dependiendo de las circunstancias. En el ejemplo ilustrado, la unidad de planificador 1504 incluye dos unidades de distribución 1522 que habilitan que dos instrucciones diferentes del mismo salto se distribuyan 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 forma que a cada unidad funcional se asigna una porción especializada del archivo de registro 1506. El archivo de registro 1506 también puede dividirse entre diferentes saltos que se ejecutan por el SM 1500. En un ejemplo, el archivo de registro 1506 proporciona almacenamiento temporal para operandos conectados a las trayectorias 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 (por ejemplo, 192). Cada núcleo 1508 puede ser una unidad de procesamiento con canalización y precisión única que incluye una unidad de lógica aritmética de punto flotante (por ejemplo, IEEE 754-2008) así como una unidad de lógica aritmética de enteros.

El SM 1500 ilustrado también incluye M unidades de precisión doble (DPU) 1510, N unidades de función especiales (SFU) 1512 y P unidades de carga/almacenamiento (LSU) 1514. Cada DPU 1510 puede implementar aritmética de punto flotante de precisión doble y cada SFU 1512 puede realizar funciones especiales tales como, por ejemplo, combinación de píxeles de copia de rectángulo, etc. Adicionalmente, 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 J memorias caché de unidad de textura/L1 1520 y una red interconectada 1516. En un ejemplo, las J memorias caché de unidad de textura/L1 1520 también se acoplan a una barra cruzada (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 cruzada que conecta cualquiera de las unidades funcionales a cualquiera de los registros en el archivo de registro 1506.

El SM 1500 puede implementarse dentro de un procesador gráfico (por ejemplo, unidad de procesamiento gráfico/GPU), en donde las memorias caché de unidad de textura/L1 1520 pueden acceder a mapas de textura de memoria y muestrear los mapas de textura para producir valores de textura muestreados para su uso en programas de sombreador. Operaciones de textura realizadas por las memorias caché de unidad de textura/L1 1520 incluyen, pero sin limitación, suavizado de contorno basándose en mapas MIP.

Ejemplo de visión general de 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 escritorio de procesador único, un sistema de estación de trabajo de múltiples procesadores o un sistema de servidor que tiene un gran número de procesadores 1602 o núcleos de procesador 1607. En una realización, el sistema 1600 es una plataforma de procesamiento incorporada dentro de un circuito integrado de sistema en un chip (SoC) para su uso en dispositivos móviles, de mano o embebidos.

Una realización del sistema 1600 puede incluir, o incorporarse dentro de una plataforma de juego basada en servidor, una consola de juegos, incluyendo una consola de juegos y medios, una consola de juegos móviles, una consola de juegos de mano o una consola de juegos en línea. En algunas realizaciones, el sistema 1600 es un teléfono móvil, teléfono inteligente, dispositivo informático de tableta o dispositivo de internet móvil. El sistema de procesamiento de datos 1600 también puede incluir, acoplar con o integrarse dentro de un dispositivo ponible, tal como un dispositivo ponible de reloj inteligente, dispositivo de gafas inteligentes, dispositivo de realidad aumentada o dispositivo de realidad virtual. En algunas realizaciones, el sistema de procesamiento de datos 1600 es una televisión o dispositivo de decodificador de salón que tiene uno o más procesadores 1602 y una interfaz gráfica generada por uno o más procesadores gráficos 1608.

En algunas realizaciones, cada uno del uno o más procesadores 1602 incluye uno o más núcleos de procesador 1607 para procesar instrucciones que, cuando se ejecutan, realizan operaciones para sistema y software de usuario. En algunas realizaciones, cada uno de los uno o más núcleos de procesador 1607 está configurado para procesar un conjunto de instrucciones específico 1609. En algunas realizaciones, el conjunto de instrucciones 1609 puede facilitar

Computación con Conjunto de Instrucciones Complejo (CISC), Computación con Conjunto Reducido de Instrucciones (RISC) o computación a través de una Palabra de Instrucción Muy Larga (VLIW). Cada uno de los múltiples núcleos de procesador 1607 puede procesar un conjunto de instrucciones 1609 diferente, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo de procesador 1607 también puede incluir otros dispositivos de procesamiento, tales como un Procesador de Señales Digitales (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 se comparte entre diversos componentes del procesador 1602. En algunas realizaciones, el procesador 1602 también usa una memoria caché externa (por ejemplo, una memoria caché de Nivel 3 (L3) o Memoria Caché de Último Nivel (LLC)) (no mostrada), que puede compartirse entre núcleos de procesador 1607 usando técnicas de coherencia de memoria caché conocidas. En el procesador 1602 se incluye adicionalmente un archivo de registro 1606 que puede incluir diferentes tipos de registros para almacenar diferentes tipos de datos (por ejemplo, registros de enteros, registros de punto flotante, registros de estado y un registro de puntero de instrucción). Algunos registros pueden ser registros de fin general, mientras otros registros pueden ser específicos al diseño del procesador 1602.

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

El dispositivo de memoria 1620 puede ser un dispositivo de memoria de acceso aleatorio dinámica (DRAM), un dispositivo de Memoria de Acceso Aleatorio Estática (SRAM), dispositivo de memoria flash, dispositivo de memoria de cambio de fase o algún otro dispositivo de memoria que tiene un rendimiento adecuado para servir como memoria de proceso. En una realización, el dispositivo de memoria 1620 puede operar como memoria de sistema para el sistema 1600, para almacenar datos 1622 e instrucciones 1621 para usar cuando el uno o más procesadores 1602 ejecutan una aplicación o proceso. El concentrador de controlador de memoria 1616 también se acopla con un procesador 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 medios.

En algunas realizaciones, el ICH 1630 habilita que 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 (por ejemplo, Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1624 (por ejemplo, unidad de disco duro, memoria flash, etc.), y un controlador de E/S heredado 1640 para acoplar dispositivos (por ejemplo, Sistema Personal 2 (PS/2)) heredados al sistema. Uno o más controladores de Bus Serial Universal (USB) 1642 conectan dispositivos de entrada, tales como combinaciones de teclado y ratón 1644. Un controlador de red 1634 también puede acoplarse al ICH 1630. En algunas realizaciones, un controlador de red de alto rendimiento (no mostrado) se acopla al bus de procesador 1610. Se apreciará que el sistema 1600 mostrado es ilustrativo y no limitante, ya que también pueden usarse otros tipos de sistemas de procesamiento de datos que están configurados de forma diferente. Por ejemplo, el concentrador de controlador de E/S 1630 puede integrarse dentro del uno o más procesadores 1602, o el concentrador de controlador de memoria 1616 y concentrador de controlador de E/S 1630 pueden integrarse 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. Esos elementos de la Figura 17 que tienen los mismos números de referencia (o nombres) que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a lo que se describe en cualquier otro sitio en el presente documento, pero sin limitación a tal. El procesador 1700 puede incluir núcleos adicionales hasta e incluyendo el núcleo adicional 1702N representado por las cajas de líneas discontinuas. Cada uno de los núcleos de procesador 1702A-1702N incluye una o más unidades de memoria caché interna 1704A-1704N. En algunas realizaciones, cada núcleo de procesador también tiene acceso a una o más unidades con memoria caché compartida 1706.

Las unidades de memoria caché interna 1704A-1704N y las unidades de memoria caché compartida 1706 representan una jerarquía de memoria caché dentro del procesador 1700. La jerarquía de memoria caché puede incluir al menos un nivel de memoria caché de instrucciones y de datos dentro de cada núcleo de procesador y uno o más niveles de memoria caché de nivel medio compartida, tal como una memoria caché de Nivel 2 (L2), Nivel 3 (L3), Nivel 4 (L4) o de otros niveles, en la que el nivel más alto de memoria caché antes de la memoria externa se clasifica como la 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 (por ejemplo, PCI, PCI Express). El núcleo de agente de sistema 1710 proporciona funcionalidad de gestión para los diversos componentes de procesador. En algunas realizaciones, el núcleo de agente de sistema 1710 incluye uno o más controladores de memoria integrados 1714 para gestionar acceso a diversos dispositivos de memoria externa (no mostrados).

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

En algunas realizaciones, el procesador 1700 incluye adicionalmente el procesador gráfico 1708 para ejecutar operaciones de procesamiento gráfico. En algunas realizaciones, el procesador gráfico 1708 se acopla con el conjunto de unidades de memoria caché compartida 1706, y el núcleo de agente de sistema 1710, incluyendo el uno o más controladores de memoria integrados 1714. En algunas realizaciones, un controlador de visualización 1711 se acopla con el procesador gráfico 1708 para llevar una salida de procesador gráfico a uno o más visualizadores 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 integrarse dentro del procesador gráfico 1708 o núcleo de agente de sistema 1710.

En algunas realizaciones, una se usa unidad interconectada basada en anillo 1712 para acoplar los componentes internos del procesador 1700. Sin embargo, puede usarse una unidad de interconexión alternativa, tal como una interconexión de punto a punto, una interconexión conmutada u otras técnicas, 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 en 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 el procesador gráfico 1708 usan módulos de memoria embebida 1718 como una Memoria Caché de Último Nivel compartida.

En algunas realizaciones, los núcleos de procesador 1702A-1702N son núcleos homogéneos que ejecutan la misma arquitectura de conjunto de instrucciones. En otra realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de arquitectura de conjunto de instrucciones (ISA), en donde uno o más de los núcleos de procesador 1702A-N ejecutan un primer conjunto de instrucciones, mientras 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, en la que uno o más núcleos que tienen un consumo de potencia relativamente mayor se acoplan con uno o más núcleos de potencia que tienen un menor consumo de potencia. Adicionalmente, el procesador 1700 puede implementarse en uno o más chips o como un circuito integrado de SoC 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 gráfico 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 con correlación de memoria con registros en el procesador gráfico y con comandos situados 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 memoria local, una o más memorias caché internas, una o más memorias caché externas compartidas, y/o a memoria de sistema.

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

En algunas realizaciones, el procesador gráfico 1800 incluye un motor de transferencia de imagen de bloque (BLIT) 1804 para realizar operaciones de rasterizador bidimensional (2D) que incluye, por ejemplo, transferencias de bloque

de límite de bit. Sin embargo, en una realización, se realizan operaciones gráficas de 2D usando uno o más componentes de motor de procesamiento gráfico (GPE) 1810. En algunas realizaciones, el motor de procesamiento gráfico 1810 es un motor de cálculo para realizar operaciones gráficas, incluyendo operaciones gráficas tridimensionales (3D) y operaciones de medios.

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

En algunas realizaciones, la canalización de medios 1816 incluye unidades de función fija o de lógica programable para realizar una o más operaciones de medios especializadas, tales como aceleración de decodificación de vídeo, desintercalado 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 de medios/3D 1815. Los hilos generados realizan cálculos para las operaciones de medios en una o más unidades de ejecución de gráficos incluidas en el subsistema de medios/3D 1815.

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

Procesamiento de medios/3D

La Figura 19 es un diagrama de bloques de un motor de procesamiento gráfico 1910 de un procesador gráfico de conformidad 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 de referencia (o nombres) que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otro sitio en el presente documento, pero sin limitación a tal.

En algunas realizaciones, el GPE 1910 se acopla con un transmisor 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 de comandos 1903 se acopla a memoria, que puede ser memoria de sistema, o una o más de memoria caché interna y memoria caché compartida. En algunas realizaciones, el transmisor 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 buscadas en una memoria intermedia de anillo, que almacena comandos para las canalizaciones 3D y de medios 1912, 1916. En una realización, la memoria intermedia de anillo puede incluir adicionalmente memorias intermedias de instrucciones por lotes que almacenan lotes de múltiples comandos. Las canalizaciones 3D y de medios 1912, 1916 procesan los comandos realizando operaciones a través de lógica dentro de las respectivas canalizaciones o distribuyendo uno o más hilos de ejecución a una matriz de unidades de ejecución 1914. En algunas realizaciones, la matriz de unidades de ejecución 1914 es escalable, de tal forma que la matriz incluye un número variable de unidades de ejecución basándose en la potencia objetivo y el nivel de rendimiento del GPE 1910.

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

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

lógica de detección de movimiento especializada (por ejemplo, dentro del motor de estimación de movimiento 1934).

En algunas realizaciones, el motor de estimación de movimiento 1934 proporciona aceleración de hardware para operaciones de vídeo realizando funciones de aceleración de vídeo tales como estimación y predicción de vector de movimiento en datos de vídeo. El motor de estimación de movimiento determina vectores de movimiento que describen la transformación de datos de imagen entre fotogramas de vídeo sucesivos. En algunas realizaciones, un códec de medios de procesador gráfico usa el motor de estimación de movimiento de vídeo 1934 para realizar operaciones en vídeo a nivel de macrobloque que de otra manera puede requerir demasiados cálculos para realizar con un procesador de fin general. En algunas realizaciones, el motor de estimación de movimiento 1934 está generalmente disponible para que componentes de procesador gráfico ayuden con las funciones de decodificación de vídeo y de procesamiento que son sensibles o adaptables a la dirección o magnitud del movimiento dentro de datos de vídeo.

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

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

Unidades de ejecución

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

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 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 fin 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 se interpretan por un transmisor de comandos 2003 en el extremo frontal de canalización 2004. En algunas realizaciones, el procesador gráfico 2000 incluye lógica de ejecución escalable para realizar procesamiento de geometría 3D y procesamiento de medios a través del núcleo gráfico o núcleos gráficos 2080A-2080N. Para los comandos de procesamiento de geometría 3D, el transmisor de comandos 2003 suministra comandos a la canalización de geometría 2036. Para al menos algunos comandos de procesamiento de medios, el transmisor de comandos 2003 suministra los comandos a un extremo frontal de vídeo 2034, que se acopla con un motor de medios 2037. En algunas realizaciones, el motor de medios 2037 incluye un Motor de Calidad de Vídeo (VQE) 2030 para posprocesamiento de vídeo e imágenes y un motor de codificación/decodificación multiformato (MFX) 2033 para proporcionar codificación y decodificación de datos de medios acelerado por hardware. En algunas realizaciones, cada uno de la canalización de geometría 2036 y el motor de medios 2037 genera hilos de ejecución para los recursos de ejecución de hilos proporcionados por al menos un núcleo gráfico 2080A.

En algunas realizaciones, el procesador gráfico 2000 incluye recursos de ejecución de hilos escalables que presentan núcleos modulares 2080A-2080N (en ocasiones denominados sectores de núcleo), teniendo cada uno múltiples subnúcleos 2050A-2050N, 2060A-2060N (en ocasiones denominados subsectores de núcleo). En algunas realizaciones, el procesador gráfico 2000 puede tener cualquier número de núcleos gráficos 2080A a 2080N. En algunas realizaciones, el procesador gráfico 2000 incluye un núcleo gráfico 2080A que tiene al menos un primer subnúcleo 2050A y un segundo subnúcleo de núcleo 2060A. En otras realizaciones, el procesador gráfico es un procesador de baja potencia con un único subnúcleo (por ejemplo, 2050A). En algunas realizaciones, el procesador gráfico 2000 incluye múltiples núcleos gráficos 2080A-2080N, incluyendo cada uno un conjunto de primeros subnúcleos 2050A-2050N y un conjunto de segundos subnúcleos 2060A-2060N. Cada subnúcleo en el conjunto de primeros subnúcleos 2050A-2050N incluye al menos un primer conjunto de unidades de ejecución 2052A-2052N y muestreadores de medios/textura 2054A-2054N. Cada subnúcleo en el conjunto de segundos subnúcleos 2060A-

2060N incluye al menos un segundo conjunto de unidades de ejecución 2062A-2062N y muestreadores 2064A-2064N. En algunas realizaciones, cada subnúcleo 2050A-2050N, 2060A-2060N comparte un conjunto de recursos compartidos 2070A-2070N. En algunas realizaciones, los recursos compartidos incluyen memoria caché compartida y lógica de operación de píxel. En las diversas realizaciones del procesador gráfico también pueden incluirse otros recursos compartidos.

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

En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un sombreador de píxeles 2102, un distribuidor 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 se interconectan a través de un tejido de interconexión que enlaza a cada uno de los componentes. En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye una o más conexiones a memoria, tal como memoria de sistema o memoria caché, a través de uno o más de memoria caché de instrucciones 2106, puerto de datos 2114, muestreador 2110 y matriz de unidades de ejecución 2108A-2108N. En algunas realizaciones, cada unidad de ejecución (por ejemplo, 2108A) es un procesador de vector individual con capacidad de ejecutar múltiples hilos simultáneos y procesar múltiples elementos de datos en paralelo para cada hilo. En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N incluye cualquier número unidades de ejecución individuales.

En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N se usa esencialmente para ejecutar programas de "sombreador". En algunas realizaciones, las unidades de ejecución en la matriz 2108A-2108N ejecutan un conjunto de instrucciones que incluye soporte nativo para muchas instrucciones de sombreador de gráficos 3D estándar, de tal forma que programas de sombreador de bibliotecas gráficas (por ejemplo, Direct 3D y OpenGL) se ejecutan con una traducción mínima. Las unidades de ejecución soportan procesamiento de vértices y geometría (por ejemplo, programas de vértice, programas de geometría, sombreadores de vértices), procesamiento de píxeles (por ejemplo, sombreadores de píxeles, sombreadores de fragmentos) y procesamiento de fin general (por ejemplo, sombreadores de cálculo y de medios).

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

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

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

Durante la ejecución, las canalizaciones de gráficos y medios enviaron peticiones de iniciación de hilos a lógica de ejecución de hilos 2100 a través de lógica de generación y distribución de hilos. En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un distribuidor de hilos local 2104 que arbitra peticiones de iniciación de hilo a partir de las canalizaciones de gráficos y de medios e instancia los hilos requeridos en una o más unidades de ejecución 2108A-2108N. Por ejemplo, la canalización de geometría (por ejemplo, 2036 de la Figura 20) distribuye procesamiento de vértices, teselación o hilos de procesamiento de geometría a lógica de ejecución de hilos 2100 (Figura 21). En algunas realizaciones, el distribuidor de hilos 2104 también puede procesar peticiones de generación de hilos de

tiempo de ejecución a partir de la ejecución de programas de sombreador.

Una vez que un grupo de objetos geométricos se ha procesado y rasterizado en datos de píxel, se invoca un sombreador de píxeles 2102 para calcular adicionalmente información de salida y provocar que los resultados se escriban en superficies de salida (por ejemplo, memorias intermedias de color, memorias intermedias de profundidad, memorias intermedias de patrones, etc.). En algunas realizaciones, el sombreador de píxeles 2102 calcula los valores de los diversos atributos de vértice que tienen que interpolarse a través del objeto rasterizado. En algunas realizaciones, el sombreador de píxeles 2102 ejecuta, a continuación, un programa de sombreador de píxeles suministrado por interfaz de programación de aplicación (API). Para ejecutar el programa de sombreador de píxeles, el sombreador de píxeles 2102 distribuye hilos a una unidad de ejecución (por ejemplo, 2108A) a través del distribuidor de hilos 2104. En algunas realizaciones, el sombreador de píxeles 2102 usa lógica de muestreo de textura en el muestreador 2110 para acceder a datos de textura en mapas de textura almacenados en memoria. Operaciones aritméticas en los datos de textura y los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico, o descartan uno o más píxeles de procesamiento adicional.

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

La Figura 22 es un diagrama de bloques que ilustra unos formatos de instrucción 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. Las cajas de línea continua ilustran los componentes que se incluyen generalmente en una instrucción de unidad de ejecución, mientras las líneas discontinuas incluyen componentes que son opcionales o que se incluyen únicamente en un subconjunto de las instrucciones. En algunas realizaciones, el formato de instrucción 2200 descrito e ilustrado son macroinstrucciones, en que son instrucciones suministradas a la unidad de ejecución, al contrario de micro operaciones que resultan de decodificación de instrucción una vez que se procesa la instrucción.

En algunas realizaciones, la unidad de ejecución de procesadores gráficos soporta de forma nativa instrucciones en un formato de 128 bits 2210. Un formato de instrucción compactado de 64 bits 2230 está disponible para algunas instrucciones basándose en la instrucción seleccionada, opciones de instrucción y número de operandos. El formato nativo de 128 bits 2210 proporciona acceso a todas las opciones de instrucción, mientras algunas opciones y operaciones se restringen 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 unidad de ejecución hace referencia a un conjunto de tablas de compactación basándose en los valores de índice y usa las salidas de tabla de compactación para reconstruir una instrucción nativa en el formato de 128 bits 2210.

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

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

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

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

En algunas realizaciones, las instrucciones se agrupan basándose en campos de bits de opcode 2212 para simplificar la decodificación de opcode 2240. Para un opcode de 8 bits, los bits 4, 5 y 6 permiten que la unidad de ejecución determine el tipo de opcode. El agrupamiento de opcode preciso mostrado es solamente un ejemplo. En algunas realizaciones, un grupo de opcode de movimiento y lógica 2242 incluye instrucciones de movimiento y lógica de datos (por ejemplo, mover (mov), comparar (cmp)). En algunas realizaciones, el grupo de movimiento y lógica 2242 comparte los cinco bits más significativos (MSB), en donde las instrucciones de movimiento (mov) son en forma de 0000xxxxb y las instrucciones de lógica son en forma de 0001xxxxb. Un grupo de instrucciones de control de flujo 2244 (por ejemplo, llamar, saltar (jmp)) incluye instrucciones en forma de 0010xxxxb (por ejemplo, 0x20). Un grupo de instrucciones misceláneas 2246 incluye una mezcla de instrucciones, incluyendo instrucciones de sincronización (por ejemplo, esperar, enviar) en forma de 0011xxxxb (por ejemplo, 0x30). Un grupo de instrucciones matemáticas en paralelo 2248 incluye instrucciones aritméticas de componente (por ejemplo, sumar, multiplicar (mul)) en forma de 0100xxxxb (por ejemplo, 0x40). El grupo de matemáticas en paralelo 2248 realiza las operaciones aritméticas en paralelo a través de canales de datos. El grupo de matemáticas de vectores 2250 incluye instrucciones aritméticas (por ejemplo, dp4) en forma de 0101xxxxb (por ejemplo, 0x50). El grupo de matemáticas de vectores realiza aritmética tal como cálculos de producto escalar 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 de referencia (o nombres) que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otro sitio en el presente documento, pero sin limitación a tal.

En algunas realizaciones, el procesador gráfico 2300 incluye una canalización de gráficos 2320, una canalización de medios 2330, un motor de visualización 2340, lógica de ejecución de hilos 2350 y una canalización de salida de representación 2370. En algunas realizaciones, el procesador gráfico 2300 es un procesador gráfico dentro de un sistema de procesamiento de múltiples núcleos que incluye uno o más núcleos de procesamiento de fin general. El procesador gráfico se controla mediante escrituras de registro en uno o más registros de control (no mostrados) o a través de 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 fin general. Los comandos de la interconexión de anillo 2302 se interpretan por un transmisor de comandos 2303, que suministra instrucciones a componentes individuales de la canalización de gráficos 2320 o la canalización de medios 2330.

En algunas realizaciones, el transmisor de comandos 2303 dirige la operación de un buscador de vértices 2305 que lee datos de vértice de memoria y ejecuta comandos de procesamiento de vértices proporcionados por el transmisor de comandos 2303. En algunas realizaciones, el buscador de vértices 2305 proporciona datos de vértice a un sombreador de vértices 2307, que realiza operaciones de iluminación y transformación de espacio de coordenadas a cada vértice. En algunas realizaciones, el buscador de vértices 2305 y el sombreador de vértices 2307 ejecutan instrucciones de procesamiento de vértices distribuyendo hilos de ejecución a unidades de ejecución 2352A, 2352B a través de un distribuidor de hilos 2331.

En algunas realizaciones, las unidades de ejecución 2352A, 2352B son una matriz de procesadores de vectores que tiene un conjunto de instrucciones para realizar operaciones gráficas y de medios. En algunas realizaciones, las unidades de ejecución 2352A, 2352B tienen una memoria caché L1 2351 conectada que es específica para cada matriz o se comparte entre las matrices. La memoria caché puede configurarse como una memoria caché de datos, una memoria caché de instrucciones, o una única memoria caché que se particiona para contener datos e instrucciones en diferentes particiones.

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

geométrico grueso que se proporciona como una entrada a la canalización de gráficos 2320. En algunas realizaciones, si no se usa teselación, los componentes de teselación 2311, 2313 y 2317 pueden omitirse.

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

Antes de la rasterización, un recortador 2329 procesa datos de vértice. 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 (por ejemplo, componente de prueba de profundidad) en la canalización de salida de representación 2370 distribuye sombreadores de píxeles para convertir los objetos geométricos a sus representaciones por píxel. En algunas realizaciones, se incluye una lógica de sombreador de píxel en la lógica de ejecución de hilos 2350. En algunas realizaciones, una aplicación puede omitir el rasterizador 2373 y acceder a datos de vértice no rasterizados a través de una unidad de flujo de salida 2323.

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

En algunas realizaciones, la canalización de salida de representación 2370 contiene un rasterizador 2373 que convierte objetos basados en vértices en una representación basada en píxel asociada. En algunas realizaciones, la lógica de rasterizador incluye una unidad de ventana/enmascarador para realizar rasterización lineal y triangular de función fija. Una memoria caché de representación 2378 asociada y memoria caché de profundidad 2379 también están disponibles en algunas realizaciones. Un componente de operaciones de píxel 2377 realiza operaciones basadas en píxel en los datos, aunque en algunos casos, operaciones de píxel asociadas con operaciones 2D (por ejemplo, transferencias de imagen de bloque de bit con combinación) se realizan por el motor 2D 2341, o se sustituyen en el momento de visualización por el controlador de visualización 2343 usando planos de visualización de superposición. En algunas realizaciones, una memoria caché compartida L3 2375 está disponible para todos los componentes gráficos, permitiendo el uso compartido de datos sin el uso de memoria de sistema principal.

En algunas realizaciones, la canalización de medios de 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 de comandos 2303. En algunas realizaciones, la canalización de medios 2330 incluye un transmisor 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 distribuir a la lógica de ejecución de hilos 2350 a través del distribuidor 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 tejido de interconexión. En algunas realizaciones, el motor de visualización 2340 incluye un motor 2D 2341 y un controlador de visualización 2343. En algunas realizaciones, el motor de visualización 2340 contiene lógica de fin especial con capacidad 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 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 son configurables para realizar operaciones basándose en múltiples interfaces de programación de gráficos y medios y no son específicos a ninguna interfaz de programación de aplicación (API). En algunas realizaciones, software de controlador para el procesador gráfico traduce llamadas de API, que son específicas para una biblioteca de gráficos o medios particular, a comandos que pueden procesarse por el procesador gráfico. En algunas realizaciones, se proporciona soporte para la Biblioteca de Gráficos Abierta (OpenGL) y Lenguaje de Computación Abierto (OpenCL) de Khronos Grupo, la biblioteca de Direct3D de Microsoft Corporation, o puede proporcionarse soporte tanto a OpenGL como D3D. También puede proporcionarse soporte para la Biblioteca de Visión Artificial de Código Abierto (OpenCV). También se soportaría una API futura con una canalización 3D compatible si puede hacerse una correlación desde la canalización de la futura API a la canalización del procesador gráfico.

Programación de canalización de gráficos

La Figura 24A es un diagrama de bloques que ilustra un formato de comando 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. Las cajas de línea continua en la Figura 24A ilustran los componentes que se incluyen generalmente en un comando de gráficos mientras las líneas discontinuas incluyen componentes que son opcionales o que se incluyen únicamente en un subconjunto de los comandos de gráficos. El formato de comando 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 (opcode) de comando 2404 y los datos pertinentes 2406 para el comando. En algunos comandos también se incluyen un subopcode 2405 y un tamaño de comando 2408.

En algunas realizaciones, el cliente 2402 especifica la unidad de cliente del dispositivo gráfico que procesa los datos de comando. En algunas realizaciones, un analizador de comando de procesador gráfico examina el comando de cliente de cada comando para condicionar el procesamiento adicional del comando y encaminar los datos de comando a la unidad de cliente apropiada. En algunas realizaciones, las unidades de cliente de procesador 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 de cliente tiene una correspondiente canalización de procesamiento que procesa los comandos. Una vez que la unidad de cliente recibe el comando, la unidad de cliente lee el opcode 2404 y, si está presente, el subopcode 2405 para determinar la operación a realizar. La unidad de cliente realiza el comando usando información en el campo de datos 2406. Para algunos comandos se espera que un tamaño de comando explícito 2408 especifique el tamaño del comando. En algunas realizaciones, el analizador de comandos determina automáticamente el tamaño de al menos algunos de los comandos basándose en el opcode de comando. En algunas realizaciones, los comandos se alinean a través de múltiplos de una palabra doble.

El diagrama de flujo en 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 usa una versión de la secuencia de comandos mostrada para configurar, ejecutar y terminar un conjunto de operaciones gráficas. Una secuencia de comandos de muestra se muestra y describe para propósitos de ejemplo únicamente ya que las realizaciones no se limitan a estos comandos específicos o a esta secuencia de comandos. Además, los comandos pueden emitirse como un lote de comandos en una secuencia de comandos, de tal forma que el procesador gráfico procesará la secuencia de comandos en ocurrencia al menos parcialmente.

En algunas realizaciones, la secuencia de comandos de procesador gráfico 2410 puede comenzar con un comando de vaciado de canalización 2412 para provocar que cualquier canalización de gráficos activa complete los comandos pendientes en la actualidad para la canalización. En algunas realizaciones, la canalización 3D 2422 y la canalización de medios 2424 no operan simultáneamente. El vaciado de canalización se realiza para provocar que la canalización de gráficos activa complete cualquier comando pendiente. En respuesta a un vaciado de canalización, el analizador de comandos para el procesador gráfico pausará el procesamiento de comando hasta que los motores de extracción activos completen las operaciones pendientes y se invaliden las memorias caché de lectura pertinentes. Opcionalmente, cualquier dato en la memoria caché de representación que se marca como 'sucio' puede vaciarse a la memoria. En algunas realizaciones, el comando de vaciado de canalización 2412 puede usarse para sincronización de canalización o antes de situar el procesador gráfico en un estado de baja potencia.

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 únicamente una vez está dentro de un contexto de ejecución antes de emitir comandos de canalización a no ser que el contexto sea emitir comandos para ambas canalizaciones. En algunas realizaciones, se requiere un comando de vaciado de canalización es 2412 inmediatamente antes de una conmutación de canalización a través del comando de selección de canalización 2413.

En algunas realizaciones, un comando de control de canalización 2414 configura una canalización de gráficos para 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 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 las respectivas canalizaciones para escribir 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 para realizar comunicación de hilos cruzada. En algunas realizaciones, el estado de memoria intermedia de retorno 2416 incluye seleccionar el tamaño y número de memorias intermedias de retorno a usar para un conjunto de operaciones de canalización.

Los restantes comandos en la secuencia de comandos difieren basándose en la canalización activa para operaciones. Basándose en una determinación de canalización 2420, la secuencia de comandos se adapta a la canalización 3D 2422 que comienza con el estado de canalización 3D 2430, o 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 comandos de establecimiento de estado 3D para estado de memoria intermedia de vértices, estado de elemento de vértices, estado de color constante, estado de memoria intermedia de profundidad y otras variables de estado que tienen que configurarse antes de que se procesen los comandos de primitivas 3D. Los valores de estos comandos se determinan al menos en parte basándose en la API 3D particular en uso. En algunas realizaciones, los comandos del estado de canalización 3D 2430 también son capaces de deshabilitar u omitir selectivamente ciertos elementos de canalización si no se usarán esos elementos.

En algunas realizaciones, el comando de la primitiva 3D 2432 se usa para enviar primitivas 3D a procesar por la canalización 3D. Los comandos y parámetros asociados que se pasan al procesador gráfico a través del comando de la primitiva 3D 2432 se reenvían a la función de búsqueda de vértices en la canalización de gráficos. La función de búsqueda de vértices usa los datos de comando de la primitiva 3D 2432 para generar estructuras de datos de vértice. Las estructuras de datos de vértice se almacenan en una o más memorias intermedias de retorno. En algunas realizaciones, el comando de la primitiva 3D 2432 se usa para realizar operaciones de vértices en primitivas 3D a través de sombreadores de vértices. Para procesar sombreadores de vértices, la canalización 3D 2422 distribuye 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 comando. En algunas realizaciones, la ejecución se desencadena a través de un comando de 'adelante' o 'rechazar' en la secuencia de comandos. En una realización, la ejecución de 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 procesamiento de geometría para las primitivas 3D. Una vez que las operaciones han finalizado, los objetos geométricos resultantes se rasterizan y el motor de píxeles colorea los píxeles resultantes. Para esas operaciones también pueden incluirse comandos adicionales para controlar operaciones de sombreado de píxel y de extremo final de píxel.

En algunas realizaciones, la secuencia de comandos de procesador gráfico 2410 sigue la trayectoria de la canalización de medios 2424 cuando realiza operaciones de medios. En general, el uso específico y manera de programación para la canalización de medios 2424 depende de las operaciones de medios o de cálculo a realizar. Operaciones de decodificación de medios específicas pueden descargarse a la canalización de medios durante la decodificación de medios. En algunas realizaciones, la canalización de medios también puede omitirse y la decodificación de medios puede realizarse en su totalidad o en parte usando recursos proporcionados por uno o más núcleos de procesamiento de fin general. En una realización, la canalización de medios también incluye elementos para operaciones de unidad de procesador gráfico de fin general (GPGPU), en donde el procesador gráfico se usa para realizar operaciones vectoriales de SIMD usando programas de sombreador de cálculo que no están explícitamente relacionados con la representación de primitivas gráficas.

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

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

Arquitectura de software gráfica

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

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

En algunas realizaciones, el sistema operativo 2520 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo de tipo UNIX propietario o un sistema operativo de tipo UNIX de código abierto que usa una variante del núcleo de Linux. Cuando la API Direct3D está en uso, 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 precompilación de sombreador. En algunas realizaciones, los sombreadores de nivel alto se compilan en sombreadores de nivel bajo durante la compilación de la aplicación gráfica 3D 2510.

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

Implementaciones de núcleo de IP

Uno o más aspectos de al menos una realización pueden implementarse mediante código representativo almacenado en un medio legible por máquina que representa y/o define lógica dentro de un circuito integrado tal como un procesador. Por ejemplo, el medio legible por máquina puede incluir instrucciones que representan diversa lógica dentro del procesador. Cuando se leen por una máquina, las instrucciones pueden provocar que la máquina fabrique la lógica para realizar las técnicas descritas en el presente documento. Tales representaciones, conocidas como "núcleos de IP", son unidades reutilizables de lógica para un circuito integrado que puede almacenarse en un medio legible por máquina tangible como un modelo de hardware que describe la estructura del circuito integrado. El modelo de hardware puede suministrarse a diversas instalaciones de cliente o de fabricación, que cargan el modelo de hardware en máquinas de fabricación que fabrican el circuito integrado. El circuito integrado puede fabricarse de tal forma que el circuito realiza operaciones descritas en asociación con cualquiera de las realizaciones descritas en el presente documento.

La Figura 26 es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP 2600 que puede usarse para fabricar un circuito integrado para realizar operaciones de acuerdo con una realización. El sistema de desarrollo de núcleo de IP 2600 puede usarse para generar diseños modulares y reutilizables que pueden incorporarse en un diseño mayor o usarse para construir todo un circuito integrado (por ejemplo, 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 nivel alto (por ejemplo, C/C++). La simulación de software 2610 puede usarse para diseñar, probar y verificar el comportamiento del núcleo de IP. Un diseño de nivel de transferencia de registro (RTL) puede crearse o sintetizarse, a continuación, 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 pueden crearse, diseñarse o sintetizarse diseños de nivel inferior a nivel de lógica o nivel de transistor. Por lo tanto, los detalles particulares del diseño inicial y simulación pueden variar.

El diseño de RTL 2615 o equivalente puede sintetizarse adicionalmente mediante 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 puede simularse o probarse adicionalmente para verificar el diseño de núcleo de IP. El diseño de núcleo de IP puede almacenarse para su entrega a una instalación de fabricación de terceros 2665 usando memoria no volátil 2640 (por ejemplo, disco duro, memoria flash o cualquier medio de almacenamiento no volátil). Como alternativa, el diseño de núcleo de IP puede transmitirse (por ejemplo, a través de la 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 basa al menos en parte en el diseño de núcleo de IP. El circuito integrado fabricado puede configurarse para realizar operaciones de conformidad con al menos una realización descrita en el presente documento.

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

puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El circuito integrado ilustrativo incluye uno o más procesadores de aplicación 2705 (por ejemplo, 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 de la misma o múltiples instalaciones de diseño diferentes. El circuito integrado incluye lógica periférica 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 la industria móvil (MIPI) 2755. Puede proporcionarse almacenamiento por un subsistema de memoria flash 2760 que incluye memoria flash y un controlador de memoria flash. La interfaz de memoria puede proporcionarse a través de un controlador de memoria 2765 para acceder a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad embebido 2770.

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

En un ejemplo, la canalización de procesamiento gráfico 1400 (Figura 14) obtiene objetos gráficos asociados con un extremo frontal de escena 3D 208 y rasteriza los datos de objeto gráfico en una imagen de salida. El rasterizador 1414 procesa los objetos gráficos usando una pluralidad de planos de imagen que tienen múltiple resolución de imagen antes de los píxeles que hay que pasar a la ROP 1418 para su emisión a un dispositivo de visualización como se describe con respecto a las Figuras 6-10 y en los ejemplos a continuación.

Notas y ejemplos adicionales

El Ejemplo 1 puede incluir un sistema informático que tiene una interfaz de datos que incluye uno o más de un controlador de red, un controlador de memoria o un bus, la interfaz de datos para obtener una imagen de salida que comprende una pluralidad de valores de píxel almacenados dentro de una memoria intermedia de imágenes y uno o más objetos gráficos que hay que representar dentro de una imagen de salida asociada con una escena tridimensional (3D); un escáner de imagen de salida para recuperar los contenidos de la memoria intermedia de imágenes y valores de píxel de salida a un dispositivo de visualización basándose en objetos gráficos, y un módulo de representación multiplano para generar la imagen de salida para un dispositivo de visualización de usuario. El módulo de representación multiplano incluye un representador que recibe una pluralidad de objetos gráficos para generar uno o más planos de imagen de datos de objeto, un remuestreador eleva planos de imagen de menor resolución a una mayor resolución usada por la imagen de salida, y un rasterizador combina valores de píxel desde una ubicación común en la pluralidad de planos de imagen después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida que contiene los objetos gráficos. El representador recibe uno de los objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de la escena 3D, determina en cuál de una pluralidad de planos de imagen se ubican los objetos gráficos recibidos usando la ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos posee una correspondiente resolución de imagen, y representa el objeto gráfico recibido en el plano de imagen determinado con la resolución de imagen que corresponde al plano de imagen determinado.

El Ejemplo 2 puede incluir el sistema del Ejemplo 1, un rasterizador combina valores de píxel desde una ubicación común en la pluralidad de planos de imagen después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida que contiene los objetos gráficos, en donde el uno de una pluralidad de planos de imagen comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo.

El Ejemplo 3 puede incluir el sistema del Ejemplo 2, en donde el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

El Ejemplo 4 puede incluir el sistema del Ejemplo 2, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa.

El Ejemplo 5 puede incluir el sistema del Ejemplo 2, en donde el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

El Ejemplo 6 puede incluir el sistema del Ejemplo 2, en donde el plano de imagen de resolución media corresponde a una ubicación a lo largo del eje z entre un observador de la escena 3D y el plano de imagen de punto focal de la escena 3D que se representa.

El Ejemplo 7 puede incluir el sistema del Ejemplo 2, en donde el plano de imagen de fondo corresponde a una ubicación a lo largo del eje z más allá del plano de imagen de punto focal de la escena 3D que se representa como una escena por el observador de la escena 3D.

El Ejemplo 8 puede incluir un método de operación de un rasterizador de exploración de haces para generar una imagen de salida asociada con una escena tridimensional (3D). El método recibe uno de unos objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de la escena 3D, determina en cuál de una pluralidad de planos de imagen se ubican los objetos gráficos recibidos usando una ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos posee una correspondiente resolución de imagen, representa el objeto gráfico recibido en el plano de imagen determinado con la resolución de imagen que corresponde al plano de imagen determinado, eleva planos de imagen de menor resolución a una mayor resolución usada por la imagen de salida, un combina valores de píxel desde una ubicación común en la pluralidad de planos de imagen después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida que contiene los objetos gráficos.

El Ejemplo 9 puede incluir el método del Ejemplo 8, en donde el uno de una pluralidad de planos de imagen comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo.

El Ejemplo 10 puede incluir el método del Ejemplo 9, en donde el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

El Ejemplo 11 puede incluir el método del Ejemplo 9, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa.

El Ejemplo 12 puede incluir el método del Ejemplo 9, en donde el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

El Ejemplo 13 puede incluir el método del Ejemplo 9, en donde el plano de imagen de resolución media corresponde a una ubicación a lo largo del eje z entre un observador de la escena 3D y el plano de imagen de punto focal de la escena 3D que se representa, y el plano de imagen de fondo corresponde a una ubicación a lo largo del eje z más allá del plano de imagen de punto focal de la escena 3D que se representa como una escena por el observador de la escena 3D.

El Ejemplo 14 puede incluir al menos un medio de almacenamiento legible por ordenador que comprende un conjunto de instrucciones que, cuando se ejecuta por una plataforma informática, provoca que la plataforma informática reciba uno de unos objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de una escena 3D, determine en cuál de una pluralidad de planos de imagen se ubican los objetos gráficos recibidos usando una ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos posee una correspondiente resolución de imagen, represente el objeto gráfico recibido en el plano de imagen determinado con la resolución de imagen que corresponde al plano de imagen determinado, eleve planos de imagen de menor resolución a una mayor resolución usada por la imagen de salida, y combine valores de píxel desde una ubicación común en la pluralidad de planos de imagen después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida que contiene los objetos gráficos.

El Ejemplo 15 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 14, en donde el uno de una pluralidad de planos de imagen comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo, y el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

El Ejemplo 16 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 15, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa, y el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

El Ejemplo 17 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 15, en donde el plano de imagen de resolución media corresponde a una ubicación a lo largo del eje z entre un observador de la escena 3D y el plano de imagen de punto focal de la escena 3D que se representa, y el plano de imagen de fondo corresponde a una ubicación a lo largo del eje z más allá del plano de imagen de punto focal de la escena 3D que se representa como una escena por el observador de la escena 3D.

El Ejemplo 18 puede incluir un módulo de representación multiplano para generar una imagen de salida asociada con una escena tridimensional (3D) para un dispositivo de visualización de usuario. El módulo de representación multiplano incluye un representador recibe una pluralidad de objetos gráficos para generar uno o más planos de imagen de datos de objeto, un remuestreador eleva planos de imagen de menor resolución a una mayor resolución usada por la imagen de salida, y un rasterizador combina valores de píxel desde una ubicación común en la pluralidad de planos de imagen después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida que contiene los objetos gráficos. El representador recibe uno de los objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de la escena 3D, determina en cuál de una pluralidad de planos de imagen se ubican los objetos gráficos recibidos

usando la ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos posee una correspondiente resolución de imagen, y representa el objeto gráfico recibido en el plano de imagen determinado con la resolución de imagen que corresponde al plano de imagen determinado.

5 El Ejemplo 19 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 18, en donde el uno de una pluralidad de planos de imagen comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo.

10 El Ejemplo 20 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 19, en donde el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

15 El Ejemplo 21 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 19, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa.

El Ejemplo 22 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 19, en donde el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

20 El Ejemplo 23 puede incluir el medio de almacenamiento legible por ordenador del Ejemplo 19, en donde el plano de imagen de resolución media corresponde a una ubicación a lo largo del eje z entre un observador de la escena 3D y el plano de imagen de punto focal de la escena 3D que se representa.

25 El Ejemplo 19, en donde el plano de imagen de fondo corresponde a una ubicación a lo largo del eje z más allá del plano de imagen de punto focal de la escena 3D que se representa como una escena por el observador de la escena 3D.

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

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

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

REIVINDICACIONES

1. Un sistema informático que comprende:

- 5 una interfaz de datos (902) que incluye uno o más de un controlador de red, un controlador de memoria o un bus, la interfaz de datos (902) para obtener una imagen de salida (801, 901) que comprende una pluralidad de valores de píxel almacenados dentro de una memoria intermedia de imágenes y uno o más objetos gráficos que hay que representar dentro de dicha imagen de salida (801, 901) asociada con una escena tridimensional (3D);
- 10 un escáner de imagen de salida para recuperar los contenidos de la memoria intermedia de imágenes y emitir valores de píxel a un dispositivo de visualización (903); y
- un módulo de representación multiplano para generar la imagen de salida para un dispositivo de visualización de usuario (903), incluyendo el módulo de representación multiplano:
- un representador (912) recibe una pluralidad de objetos gráficos para generar uno o más planos de imagen (601-604) de datos de objeto, el representador (912):
- 15 recibe uno de los objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de la escena 3D; determina en cuál de una pluralidad de planos de imagen (601-604) se ubican los objetos gráficos recibidos usando la ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos (601-604) posee una correspondiente resolución de imagen; y
- 20 representa el objeto gráfico recibido en el plano de imagen determinado (601-604) con la resolución de imagen que corresponde al plano de imagen determinado (601-604);
- un remuestreador (913) eleva planos de imagen de menor resolución (601-604) a una mayor resolución usada por la imagen de salida (801, 901); y
- 25 un rasterizador (914) combina valores de píxel de cada ubicación en la pluralidad de planos de imagen (601-604) después de que cada plano de imagen (601-604) se sobremuestrea a la mayor resolución la imagen de salida (801, 901) que contiene los objetos gráficos.

2. El sistema informático de acuerdo con la reivindicación 1, en donde el uno de una pluralidad de planos de imagen (601-604) comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo.

3. El sistema informático de acuerdo con la reivindicación 2, en donde el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

- 35 4. El sistema informático de acuerdo con la reivindicación 2, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa.

5. El sistema informático de acuerdo con la reivindicación 2, en donde el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

6. Un método implementado por ordenador de operación de un rasterizador de exploración de haces para generar una imagen de salida (801, 901) asociada con una escena tridimensional (3D), que comprende:

- 45 recibir un objeto gráfico que tiene un valor de ubicación a lo largo de un eje z de la escena 3D; determinar en cuál de una pluralidad de planos de imagen (601-604) se ubica el objeto gráfico recibido usando una ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos (601-604) posee una correspondiente resolución de imagen;
- 50 representar el objeto gráfico recibido en el plano de imagen determinado (601-604) con la resolución de imagen que corresponde al plano de imagen determinado (601-604);
- elevar planos de imagen de menor resolución (601-604) a una mayor resolución usada por la imagen de salida; y
- combinar valores de píxel de cada ubicación en la pluralidad de planos de imagen (601-604) después de que cada plano de imagen se sobremuestrea a la mayor resolución la imagen de salida (801, 901) que contiene los objetos gráficos.

- 55 7. El método de operación de un rasterizador de exploración de haces de acuerdo con la reivindicación 6, en donde el uno de una pluralidad de planos de imagen (601-604) comprende: un plano de imagen de interfaz de usuario, un plano de imagen de resolución media, un plano de imagen de punto focal y un plano de imagen de fondo.

- 60 8. El método de operación de un rasterizador de exploración de haces de acuerdo con la reivindicación 7, en donde el plano de imagen de interfaz de usuario y el plano de imagen de punto focal poseen la correspondiente resolución de imagen igual a la mayor resolución.

- 65 9. El método de operación de un rasterizador de exploración de haces de acuerdo con la reivindicación 7, en donde el plano de imagen de interfaz de usuario corresponde a una ubicación a lo largo del eje z frente a la escena 3D que se representa.

10. El método de operación de un rasterizador de exploración de haces de acuerdo con la reivindicación 7, en donde el plano de imagen de punto focal corresponde a una ubicación a lo largo del eje z de la escena 3D que se representa que corresponde a una ubicación dentro de la escena 3D de máximo interés.

5 11. Un representador multiplano para generar una imagen de salida asociada con una escena tridimensional (3D) para un dispositivo de visualización de usuario (903), incluyendo el representador multiplano:

10 un representador (912) para recibir una pluralidad de objetos gráficos para generar uno o más planos de imagen (601-604) de datos de objeto, el representador para:

15 recibir uno de los objetos gráficos que tienen un valor de ubicación a lo largo de un eje z de la escena 3D; determinar en cuál de una pluralidad de planos de imagen (601-604) se ubican los objetos gráficos recibidos usando la ubicación de eje z para el objeto gráfico recibido, cada uno de la pluralidad de planos posee una correspondiente resolución de imagen; y
representar el objeto gráfico recibido en el plano de imagen determinado (601-604) con la resolución de imagen que corresponde al plano de imagen determinado (601-604);

20 un remuestreador (913) para elevar planos de imagen de menor resolución (601-604) a una mayor resolución usada por la imagen de salida (801, 901) en respuesta a que todos de la pluralidad de objetos gráficos se representan a la pluralidad de planos de imagen (601-604); y

25 un rasterizador (914) para combinar valores de píxel de cada ubicación en la pluralidad de planos de imagen (601-604) después de que cada plano de imagen (601-604) se sobremuestrea a la mayor resolución de la imagen de salida (801, 901) que contiene los objetos gráficos.

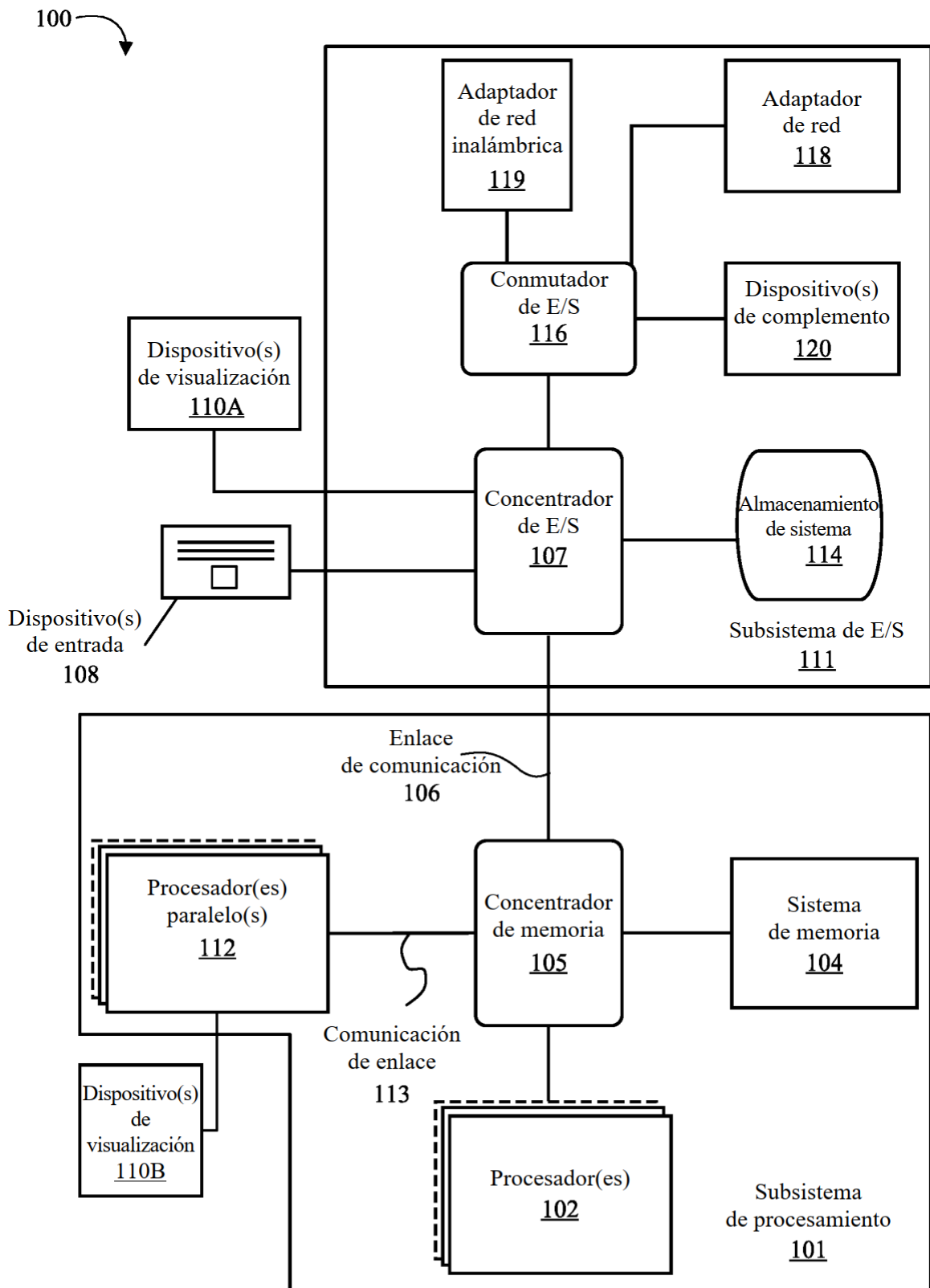


FIG. 1

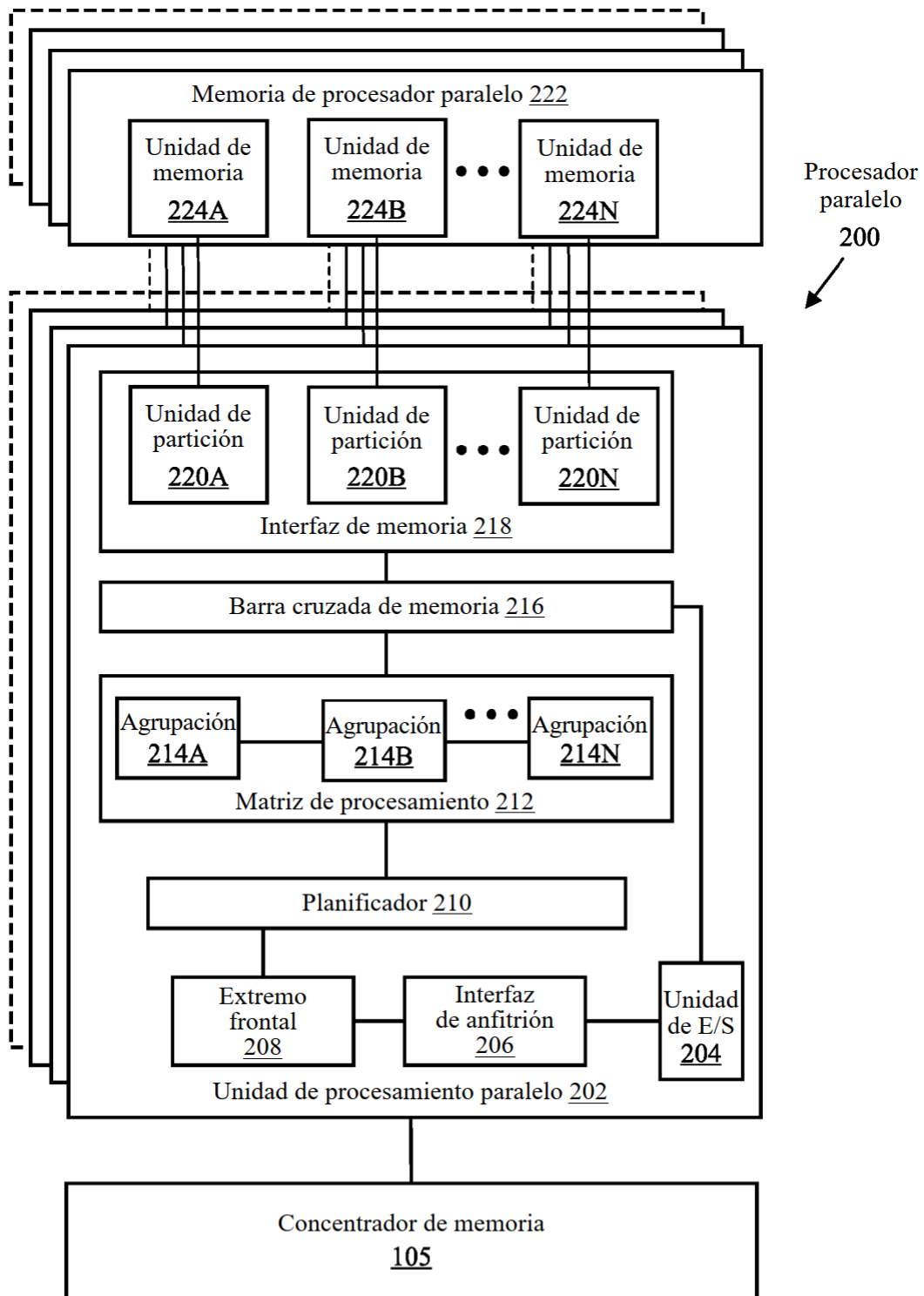


FIG. 2A

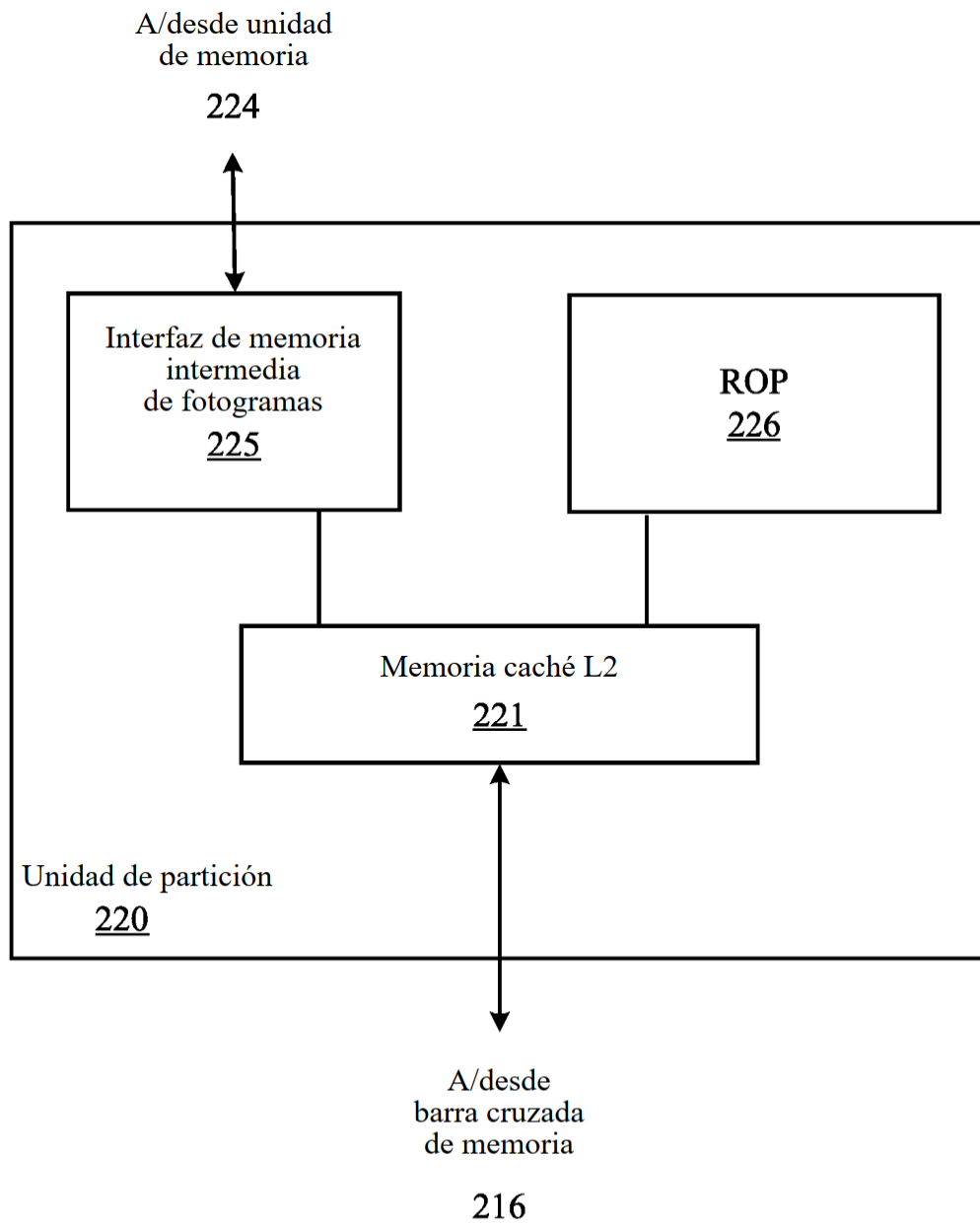


FIG. 2B

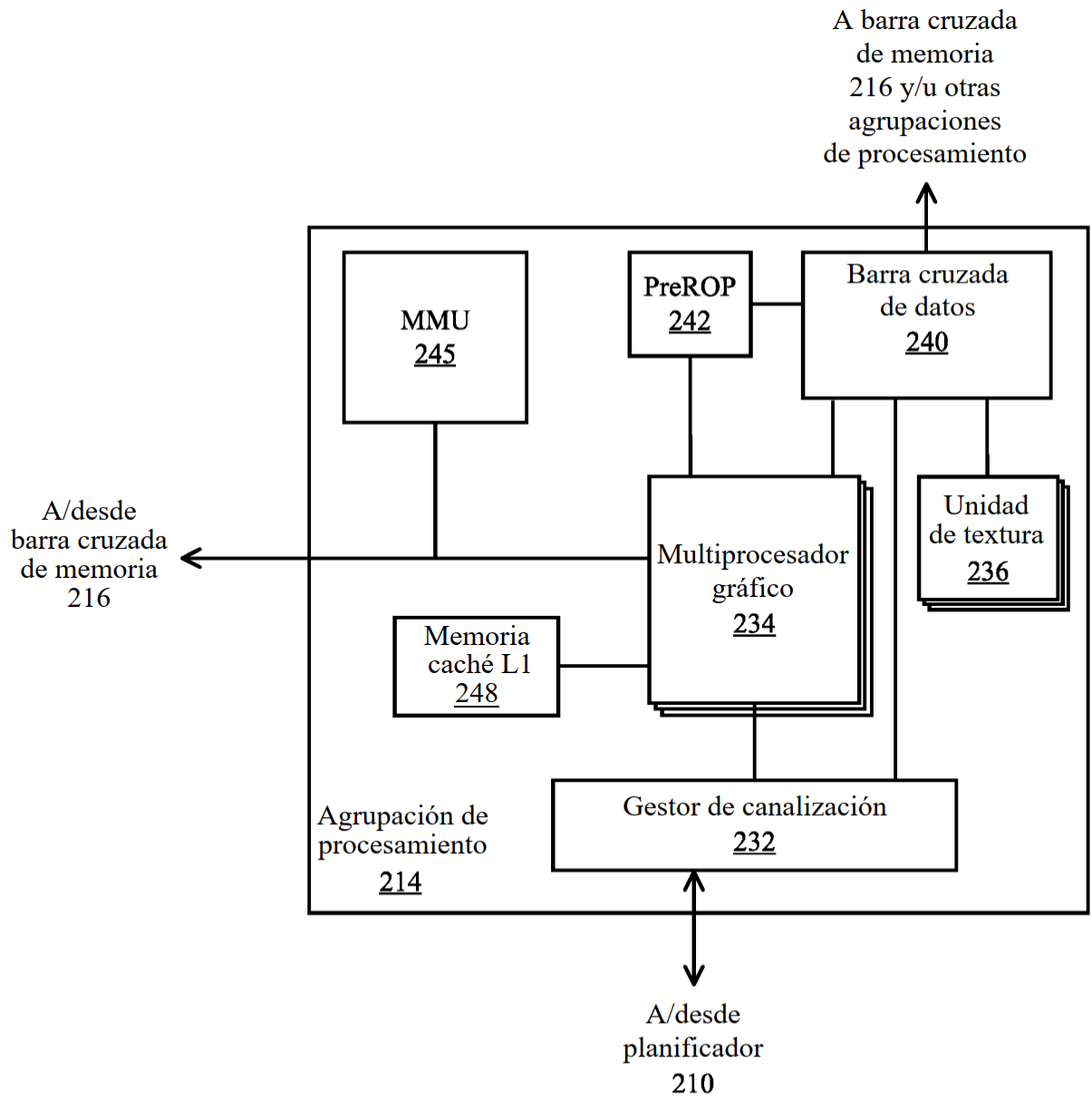


FIG. 2C

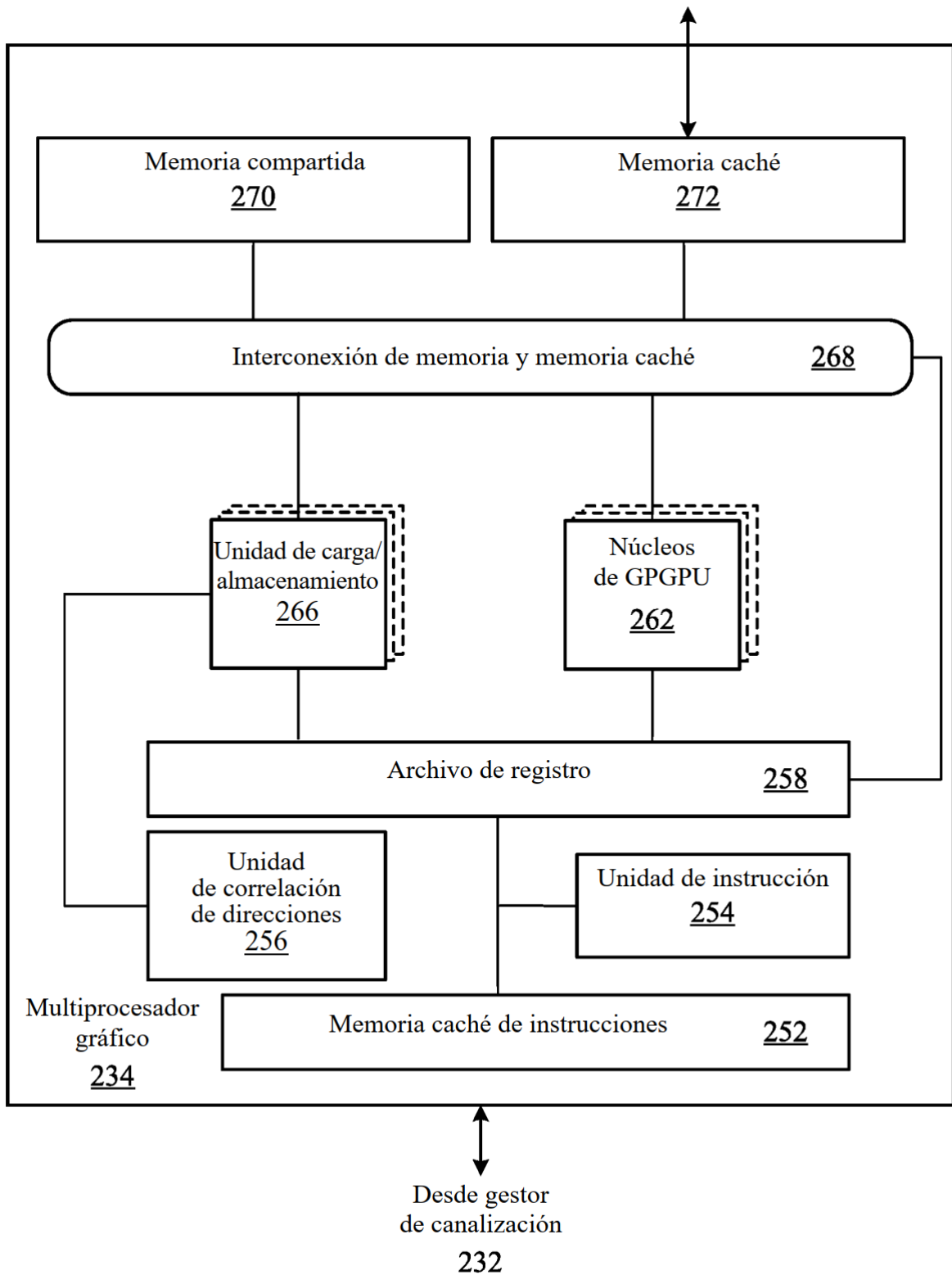


FIG. 2D

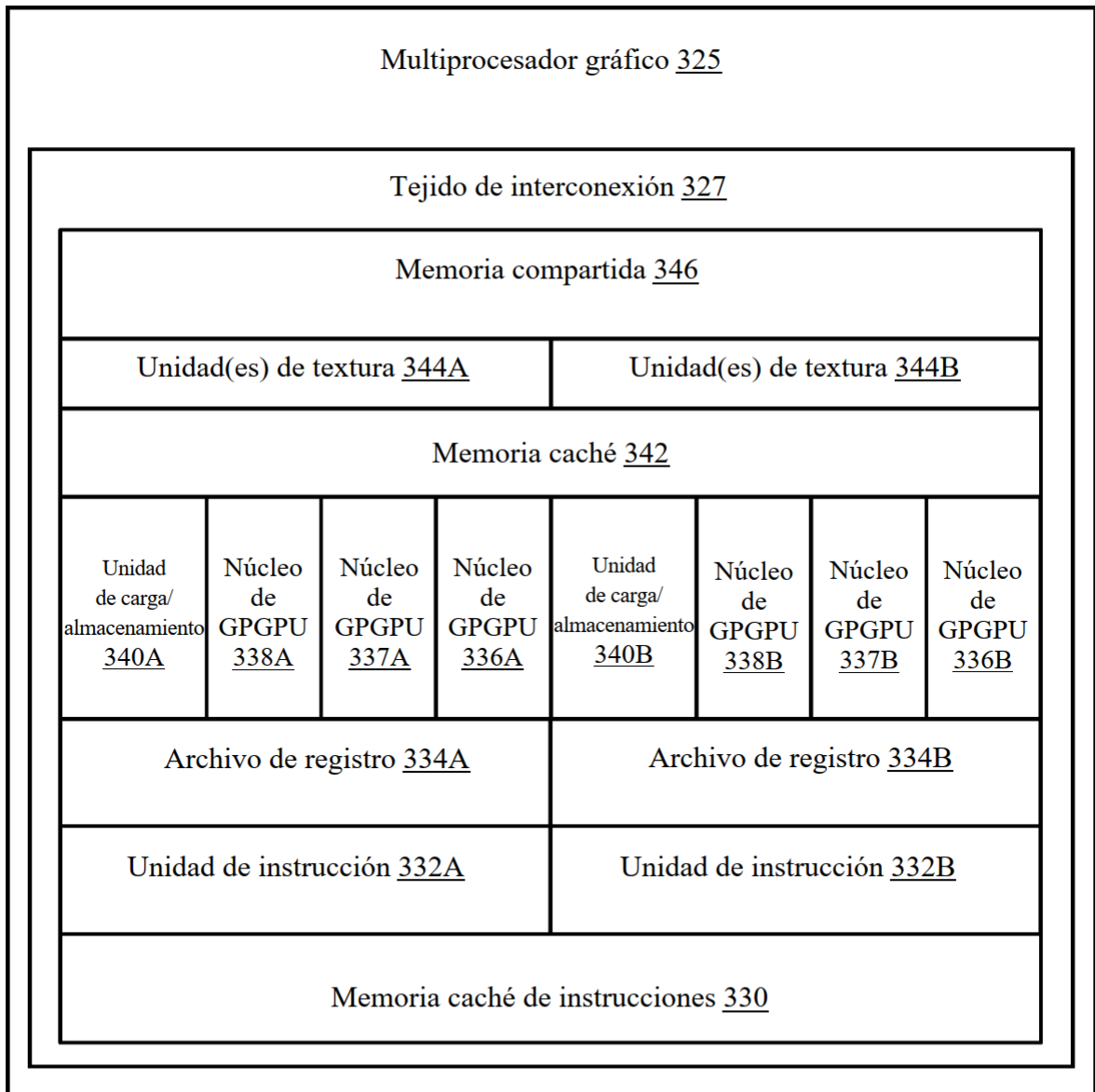


FIG. 3A

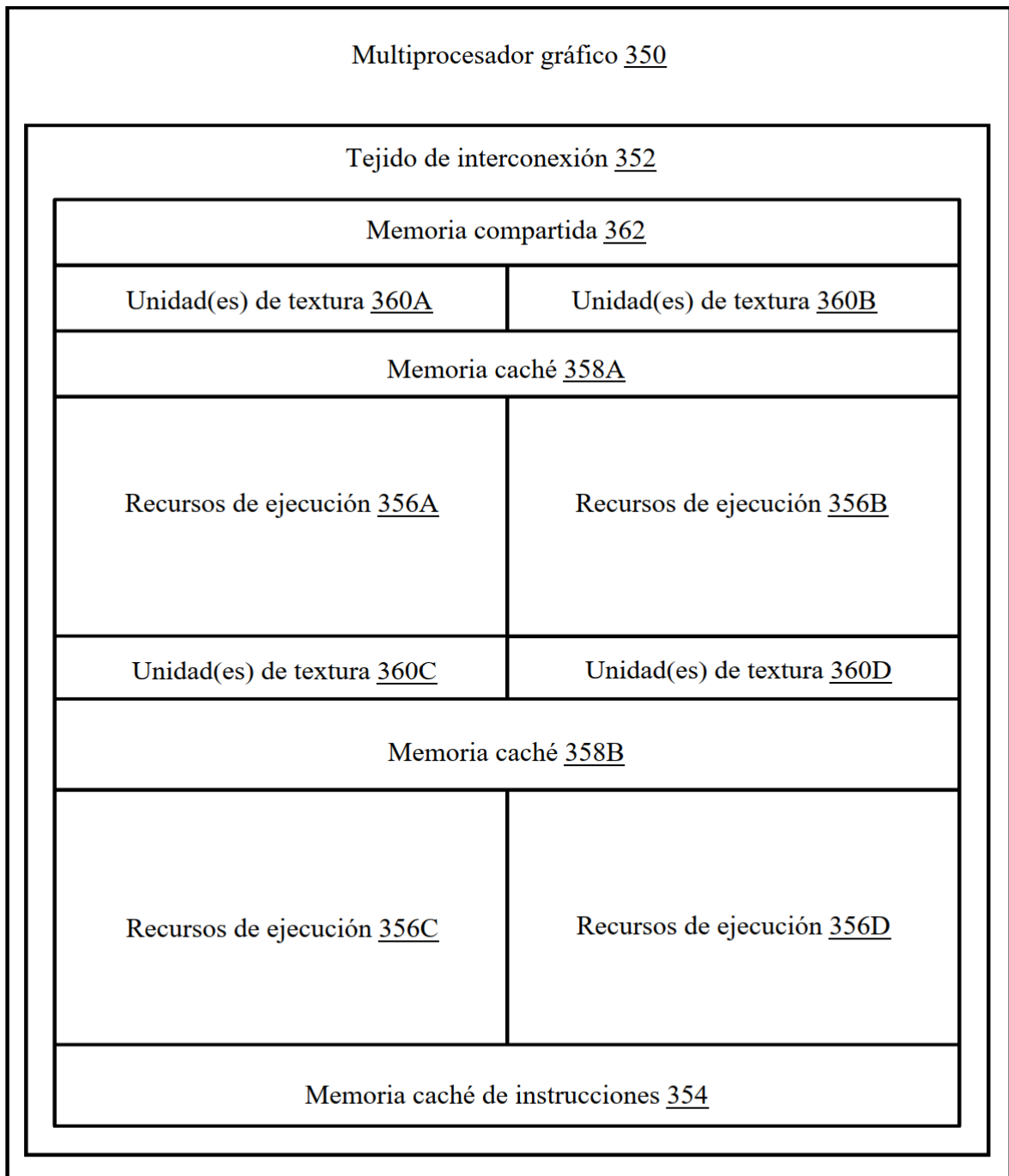


FIG. 3B

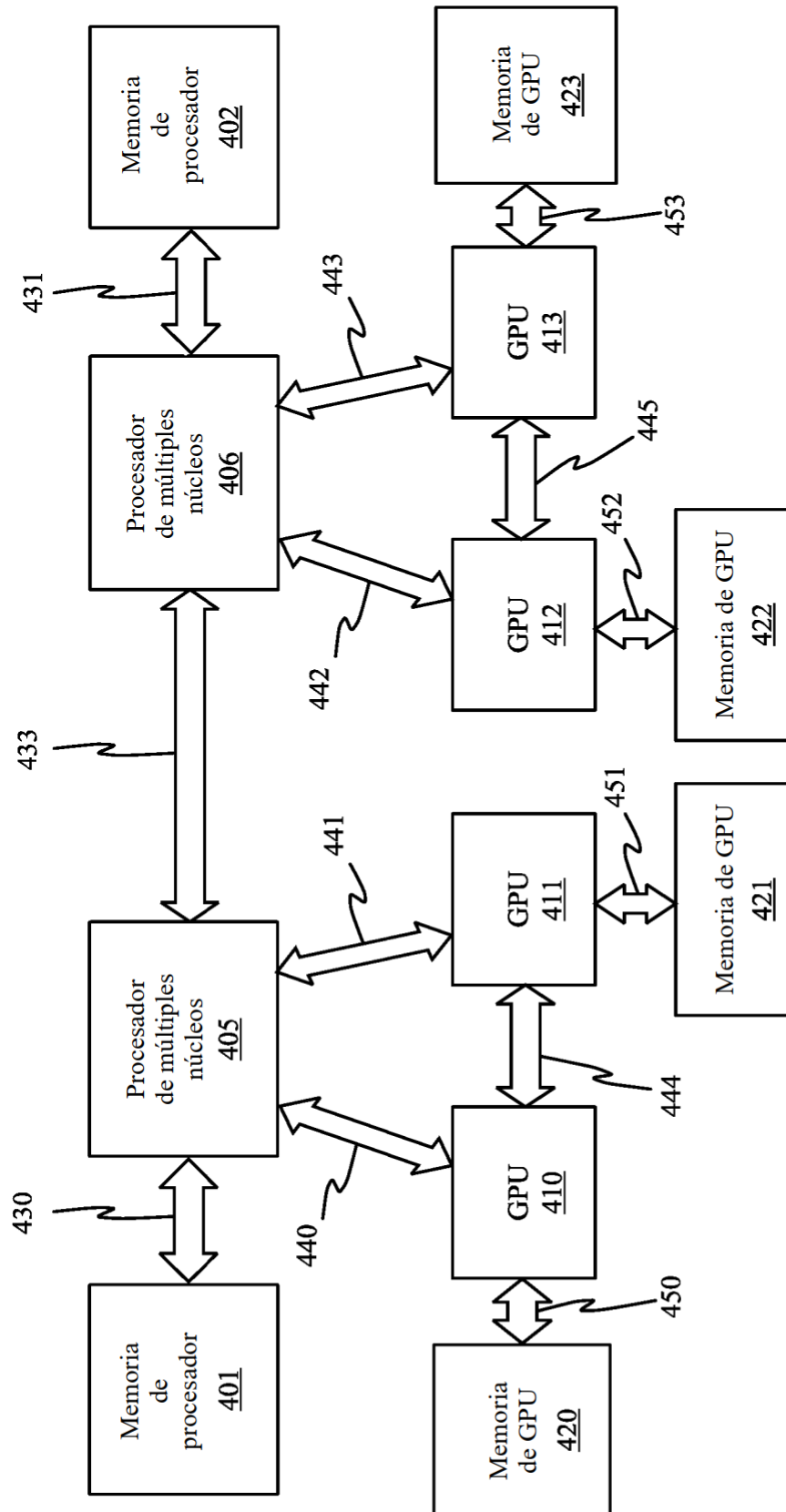


FIG. 4A

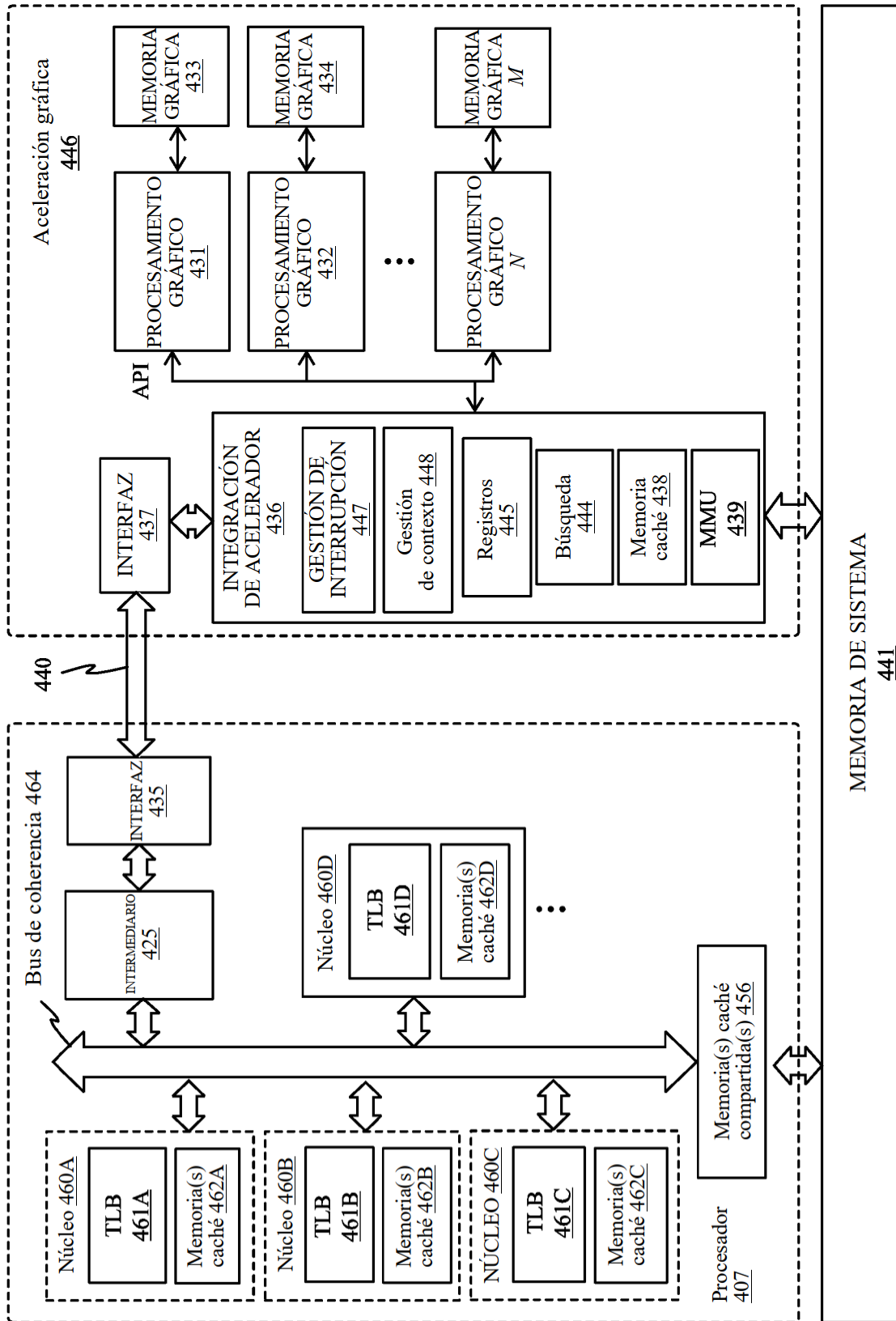


FIG. 4B

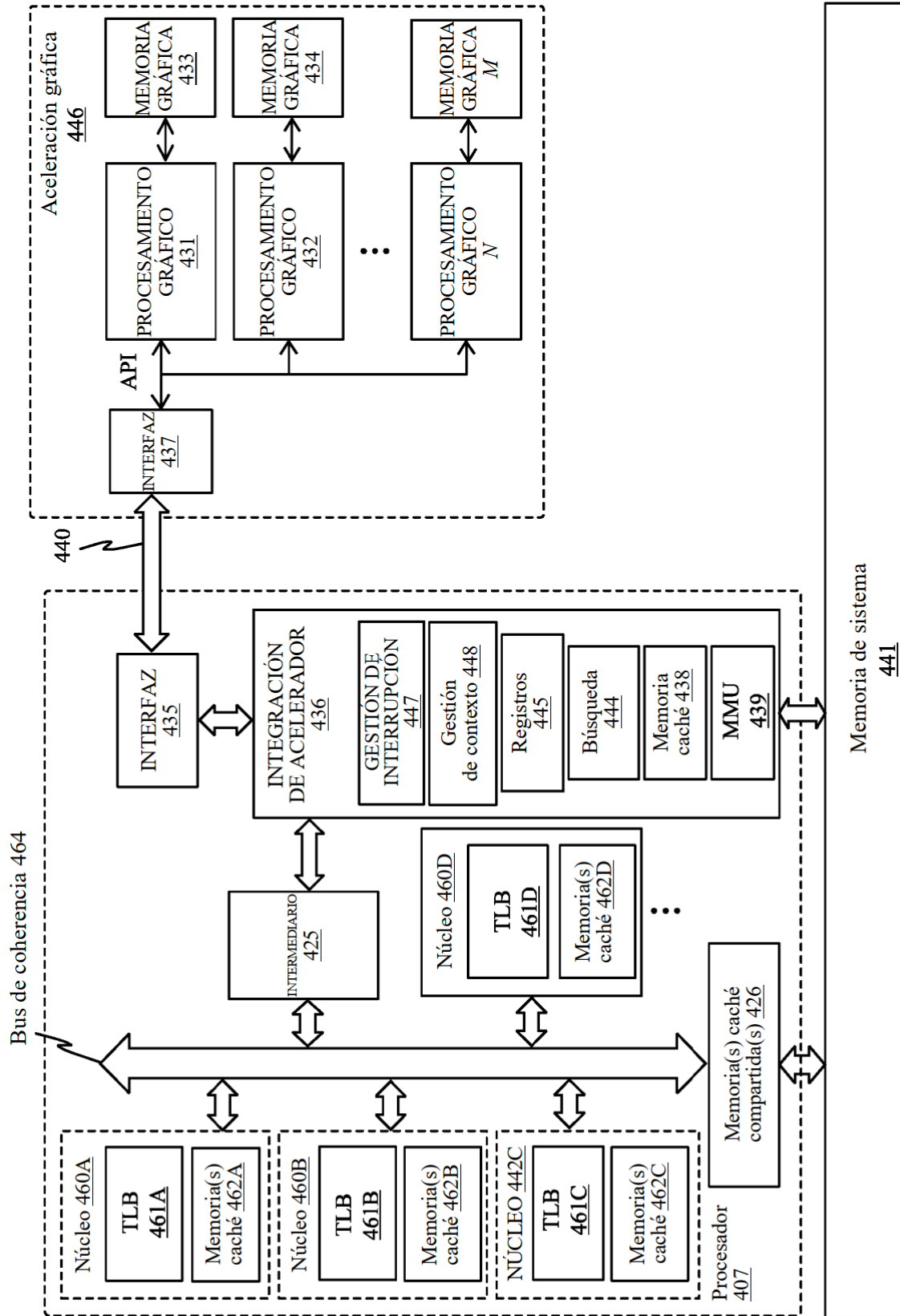


FIG. 4C

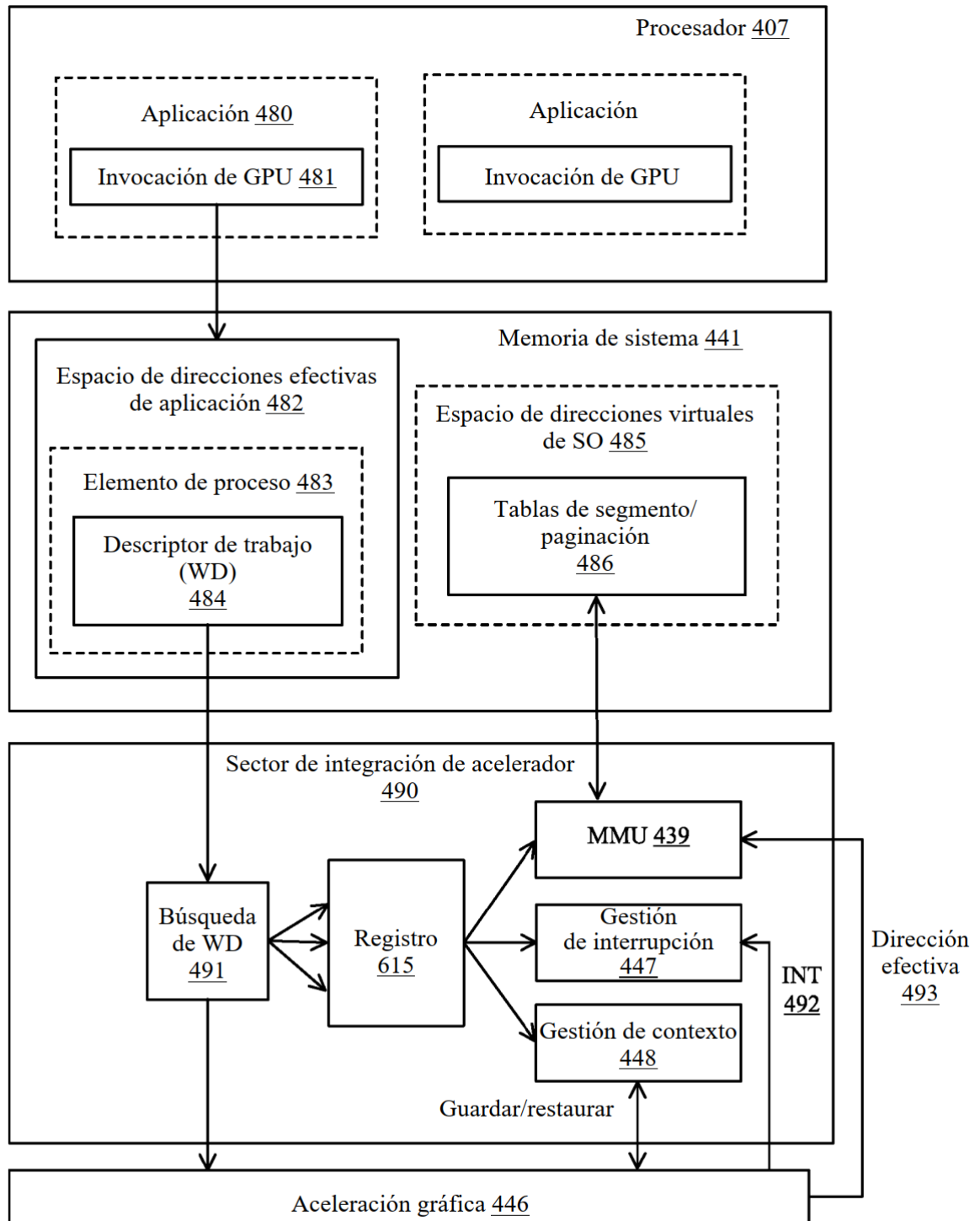


FIG. 4D

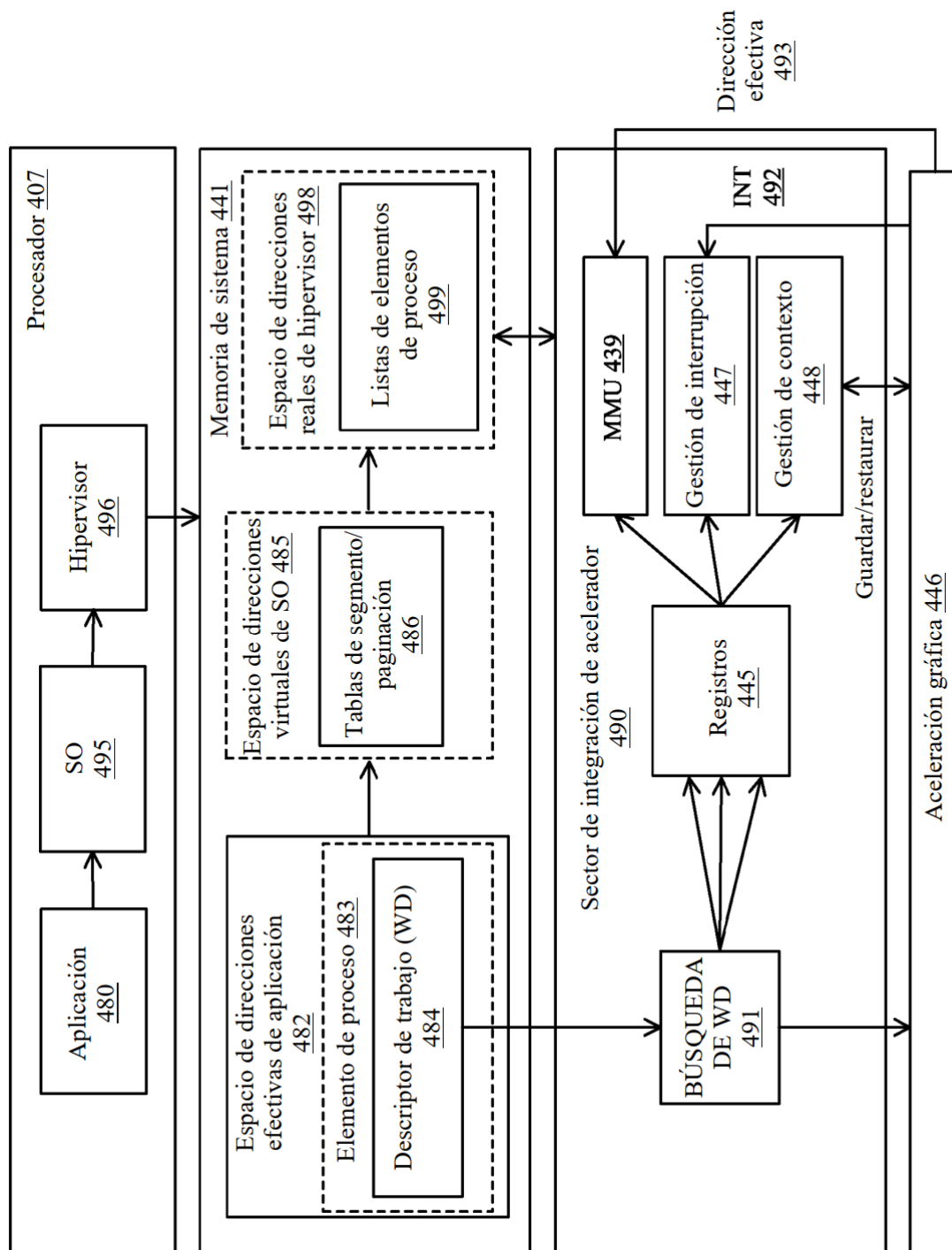


FIG. 4E

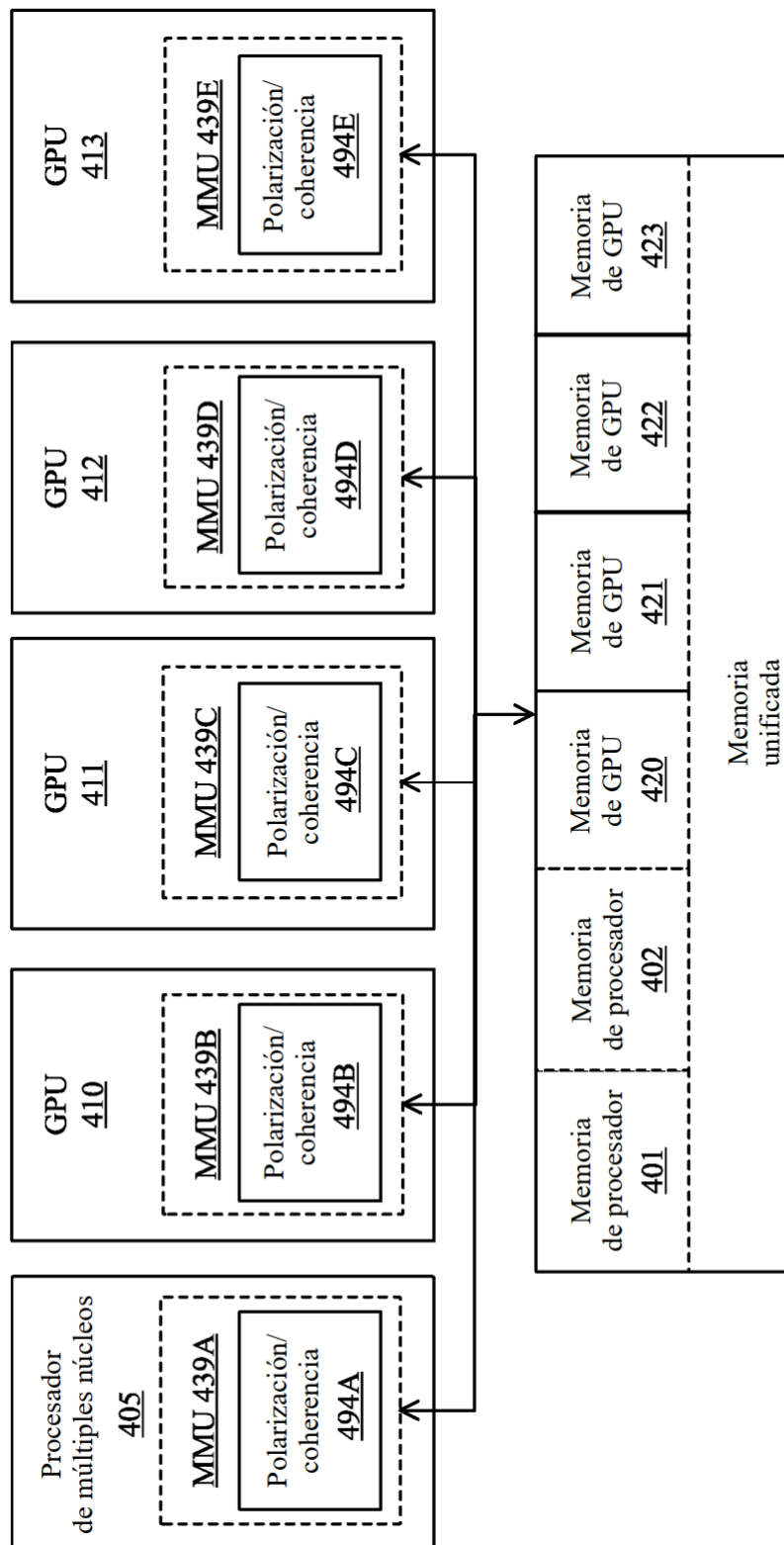


FIG. 4F

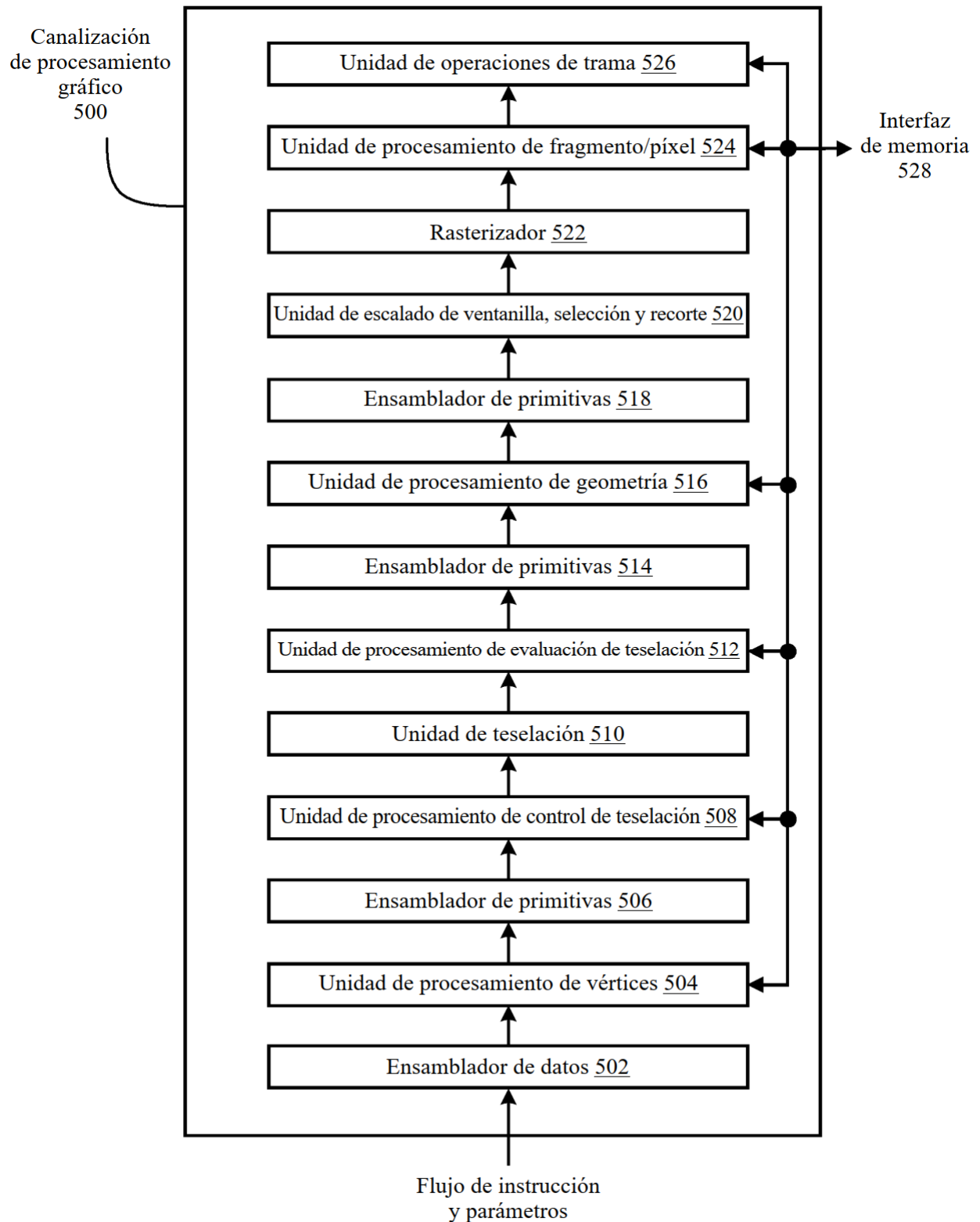


FIG. 5

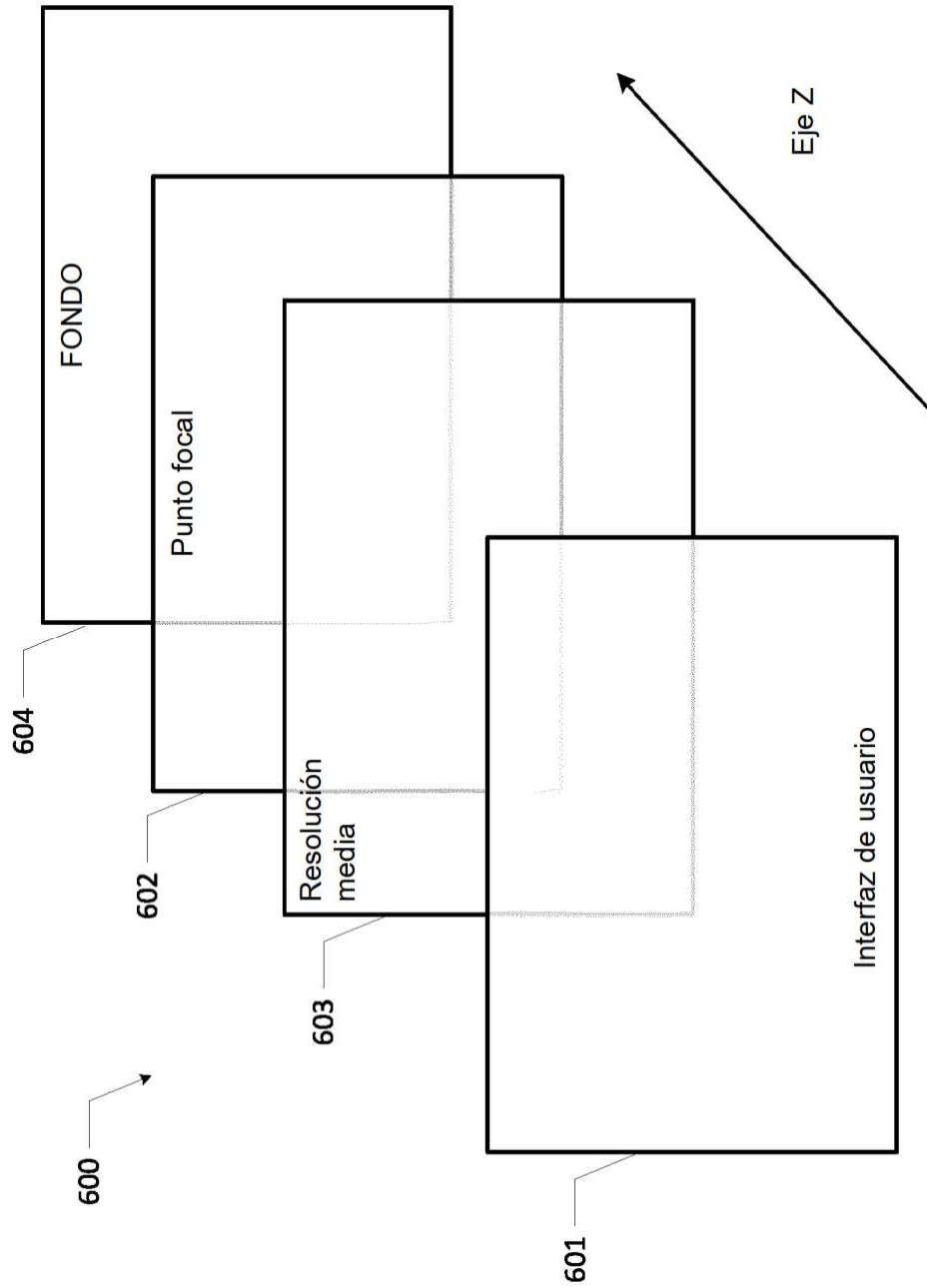


FIG. 6

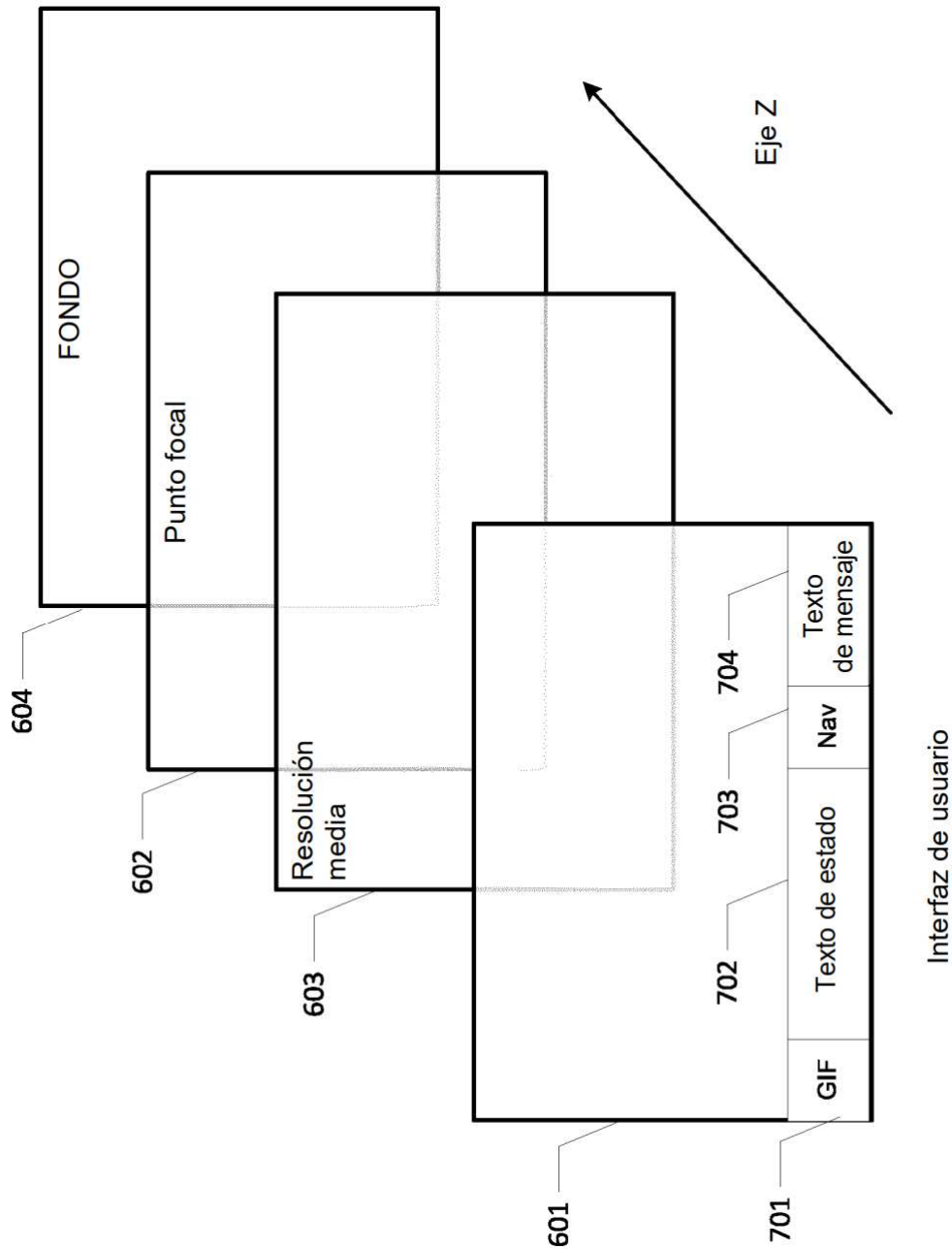


FIG. 7A

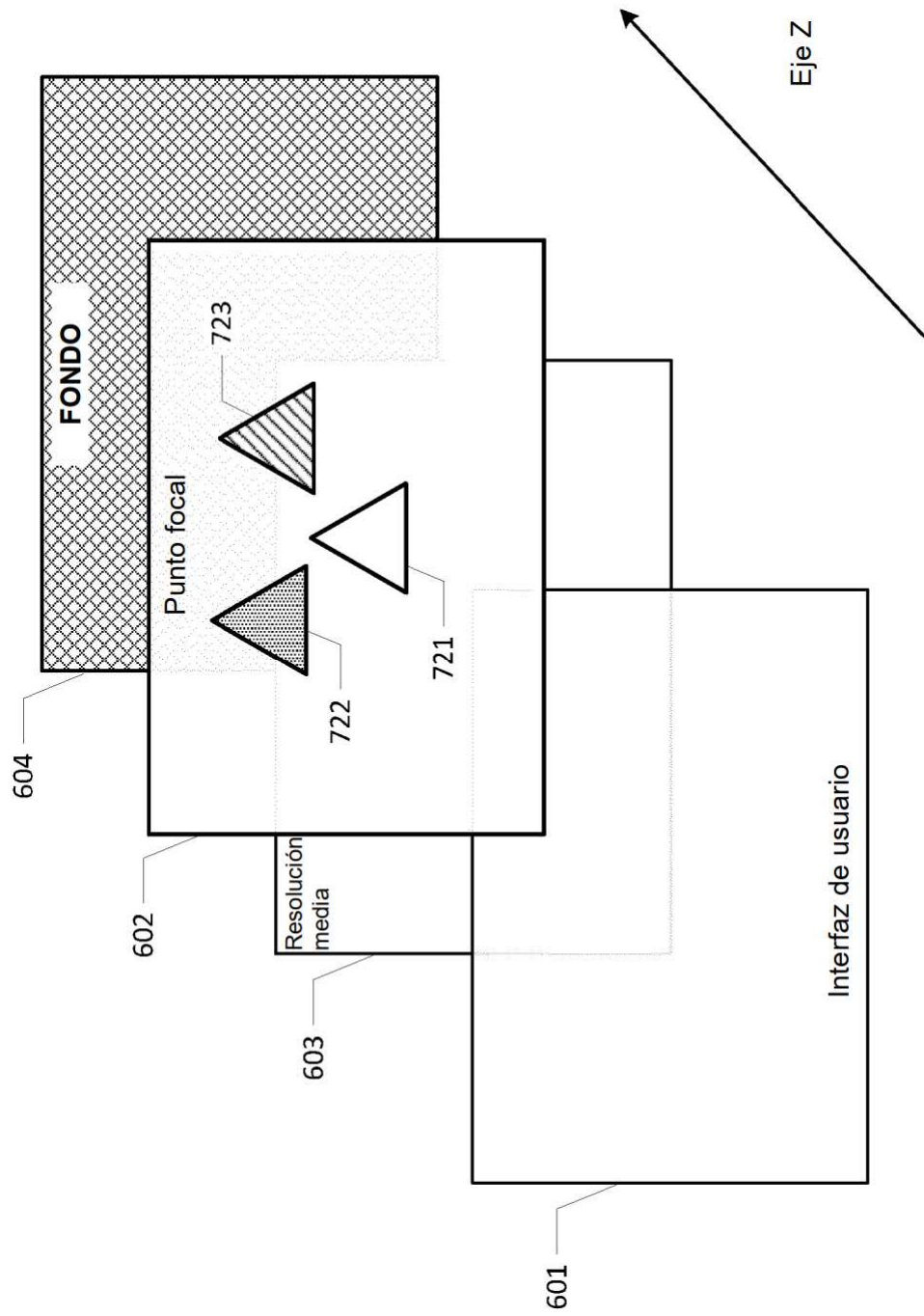


FIG. 7B

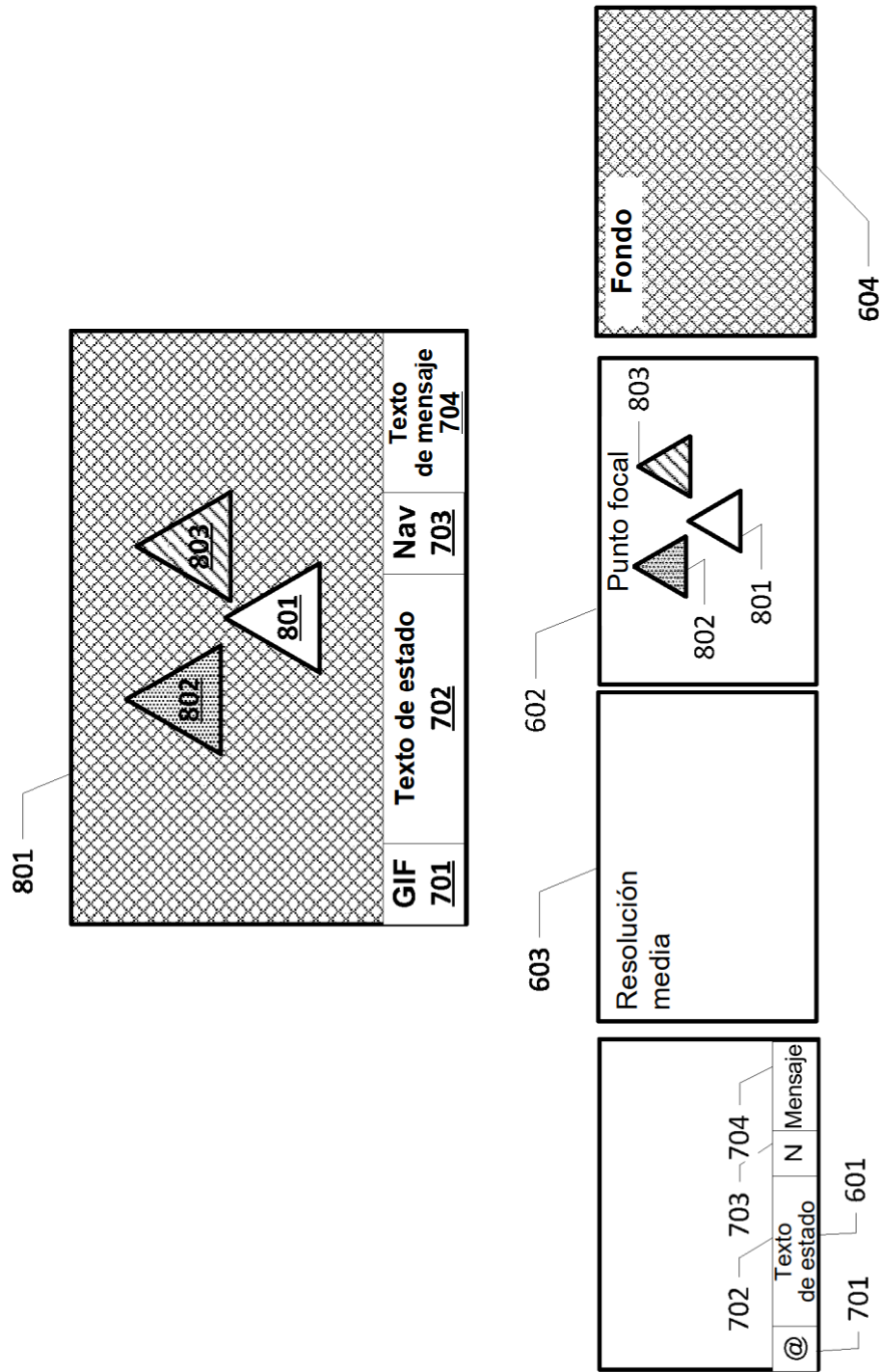


FIG. 8

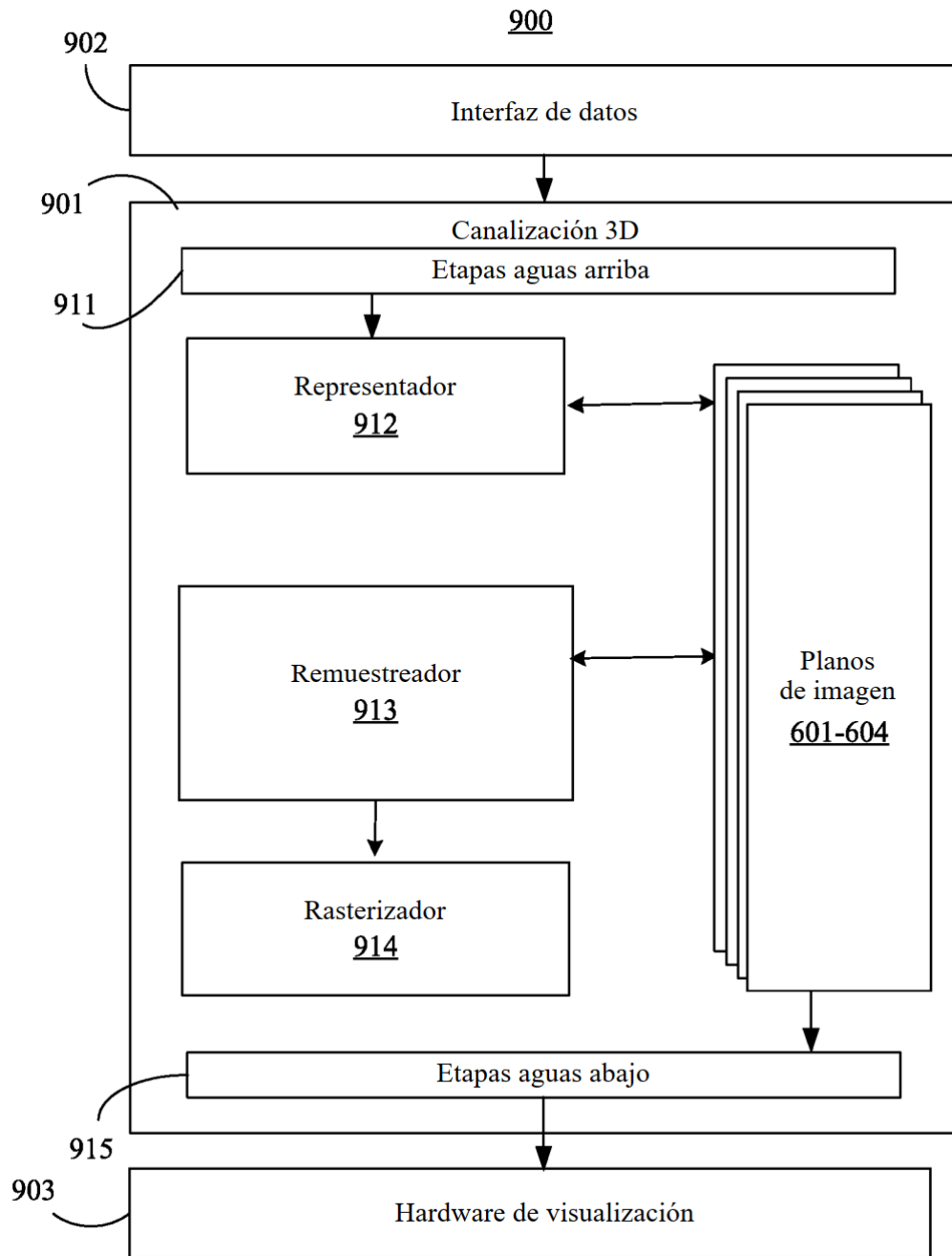


FIG. 9

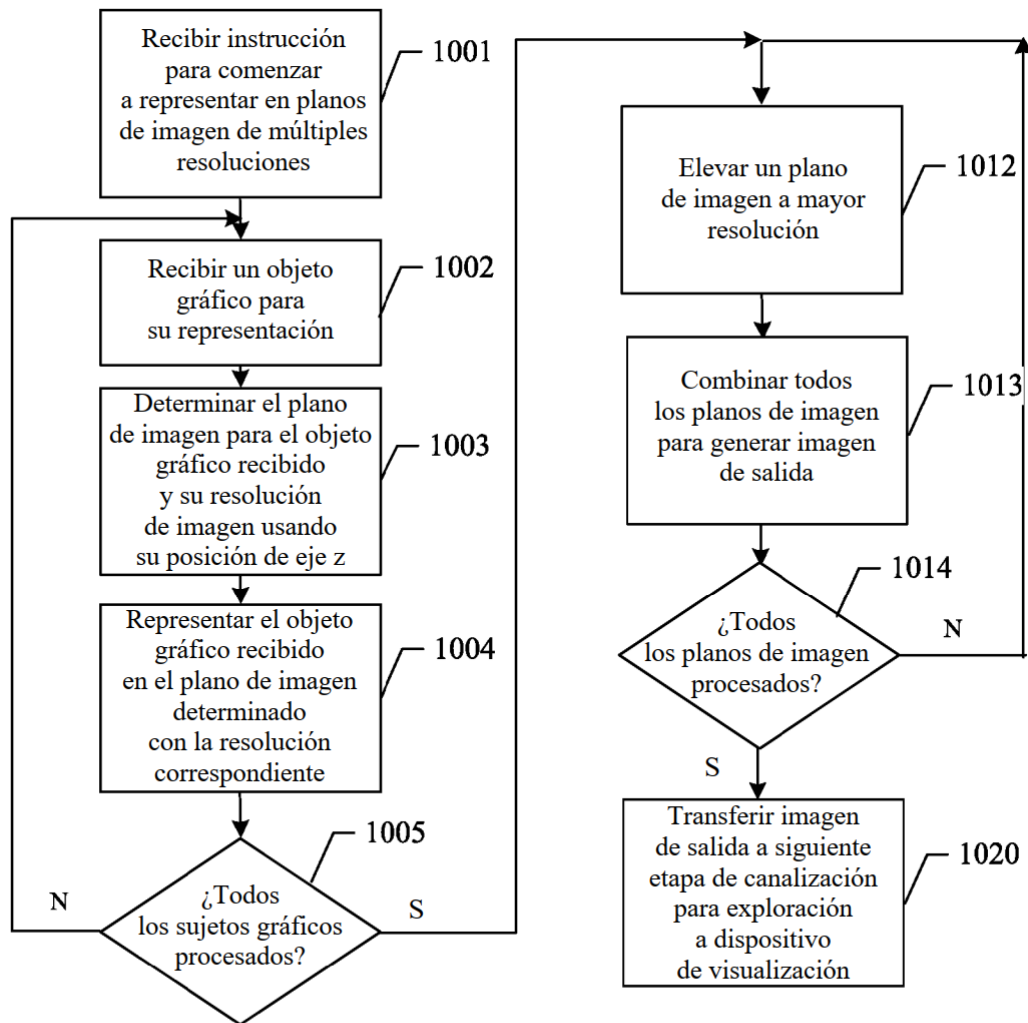


FIG. 10

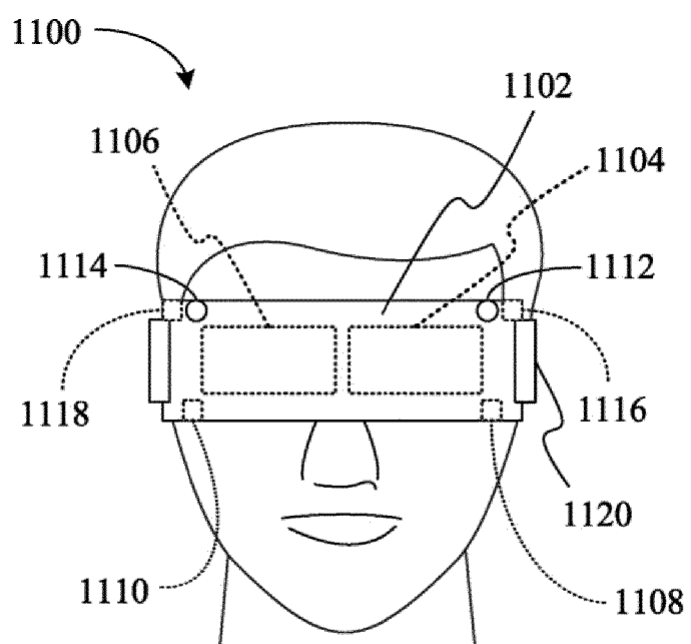


FIG. 11

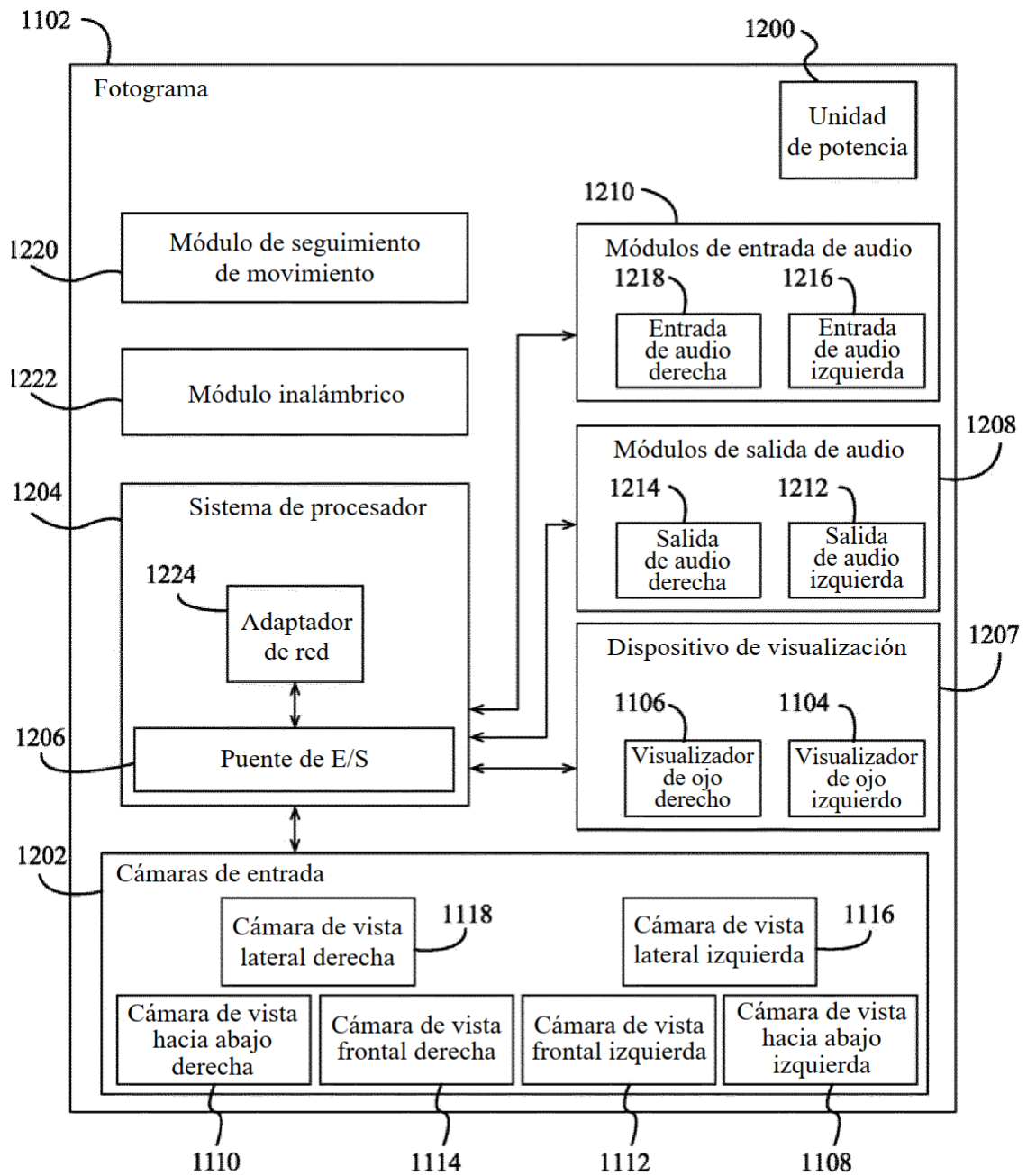


FIG. 12

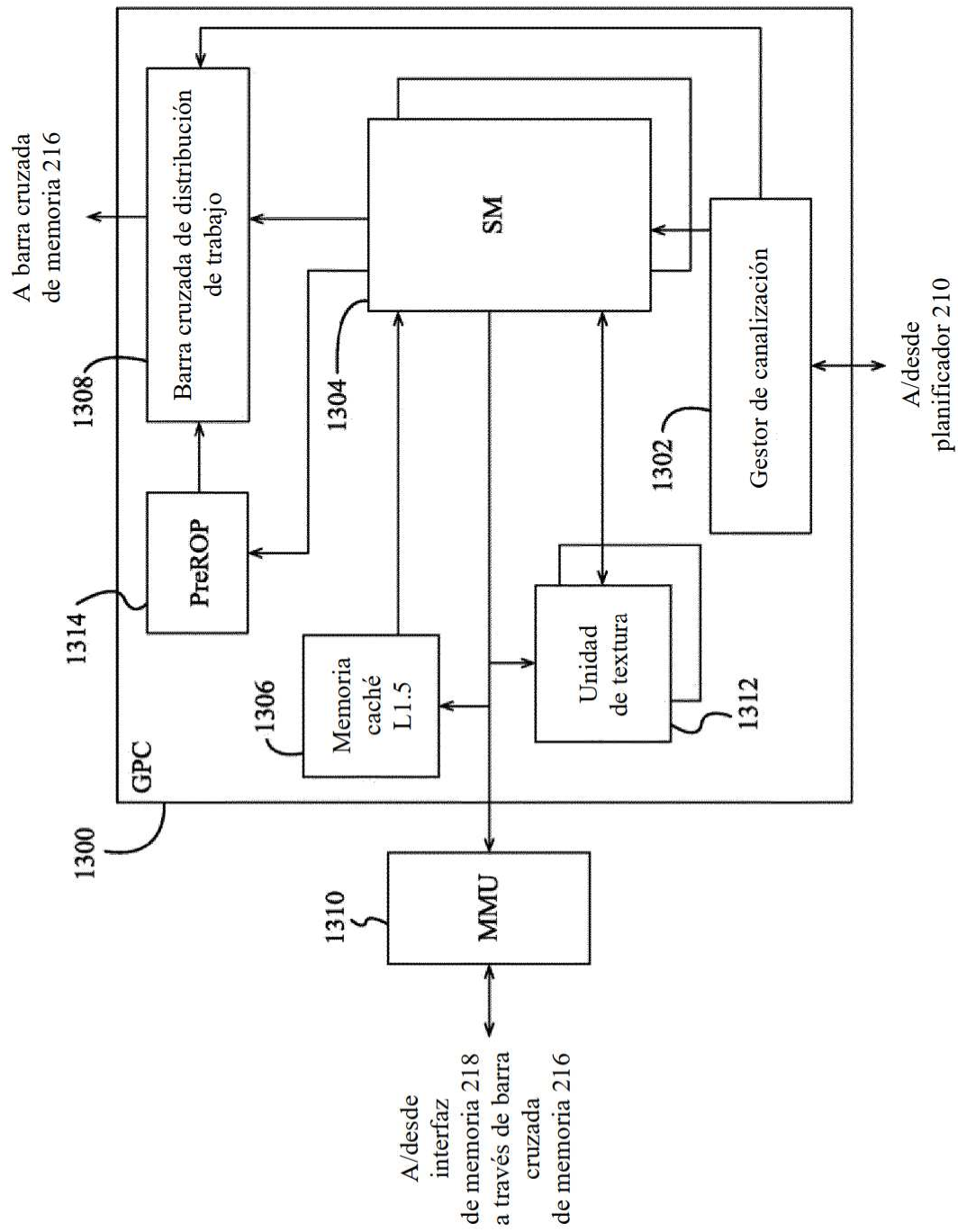


FIG. 13

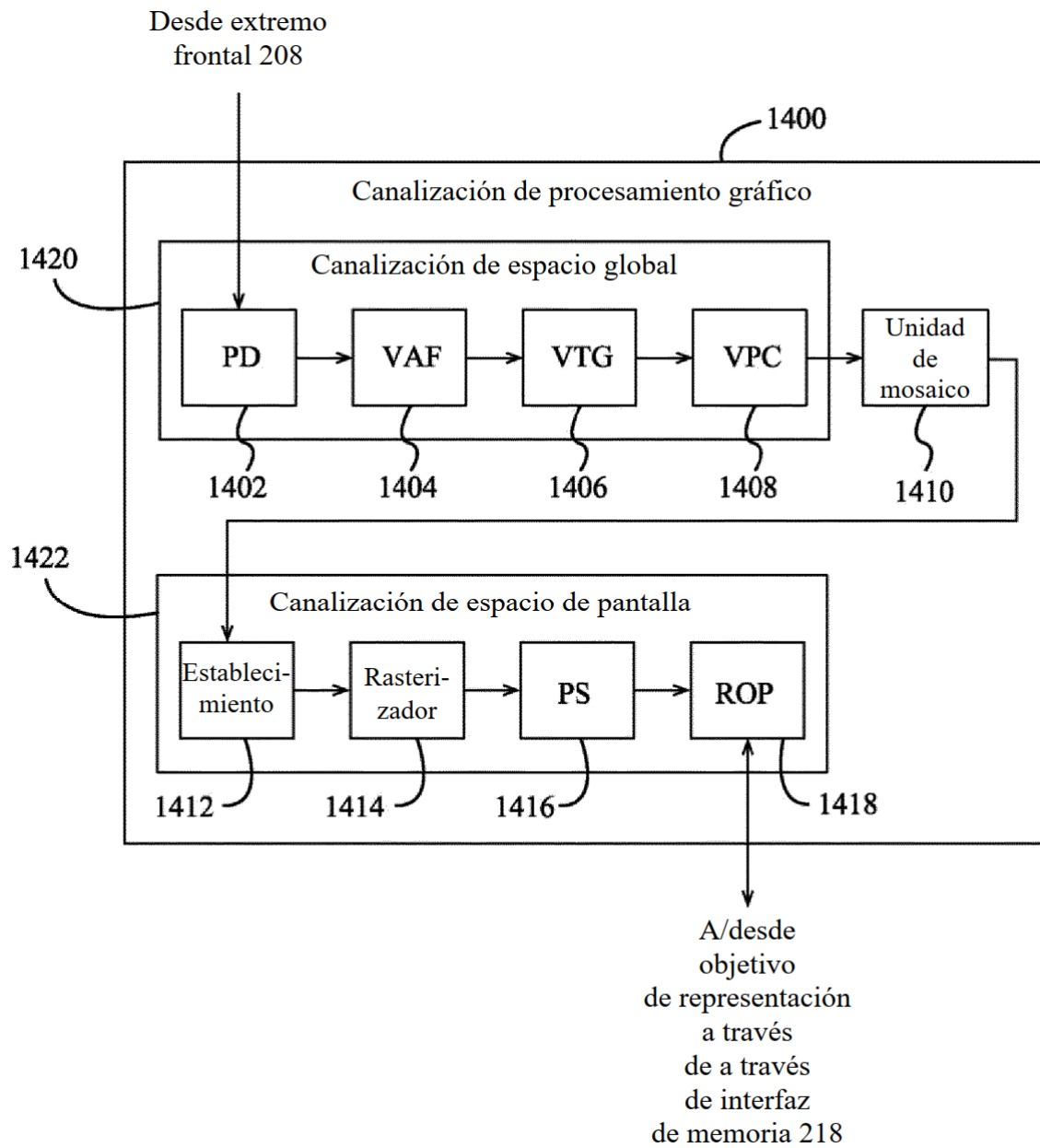


FIG. 14

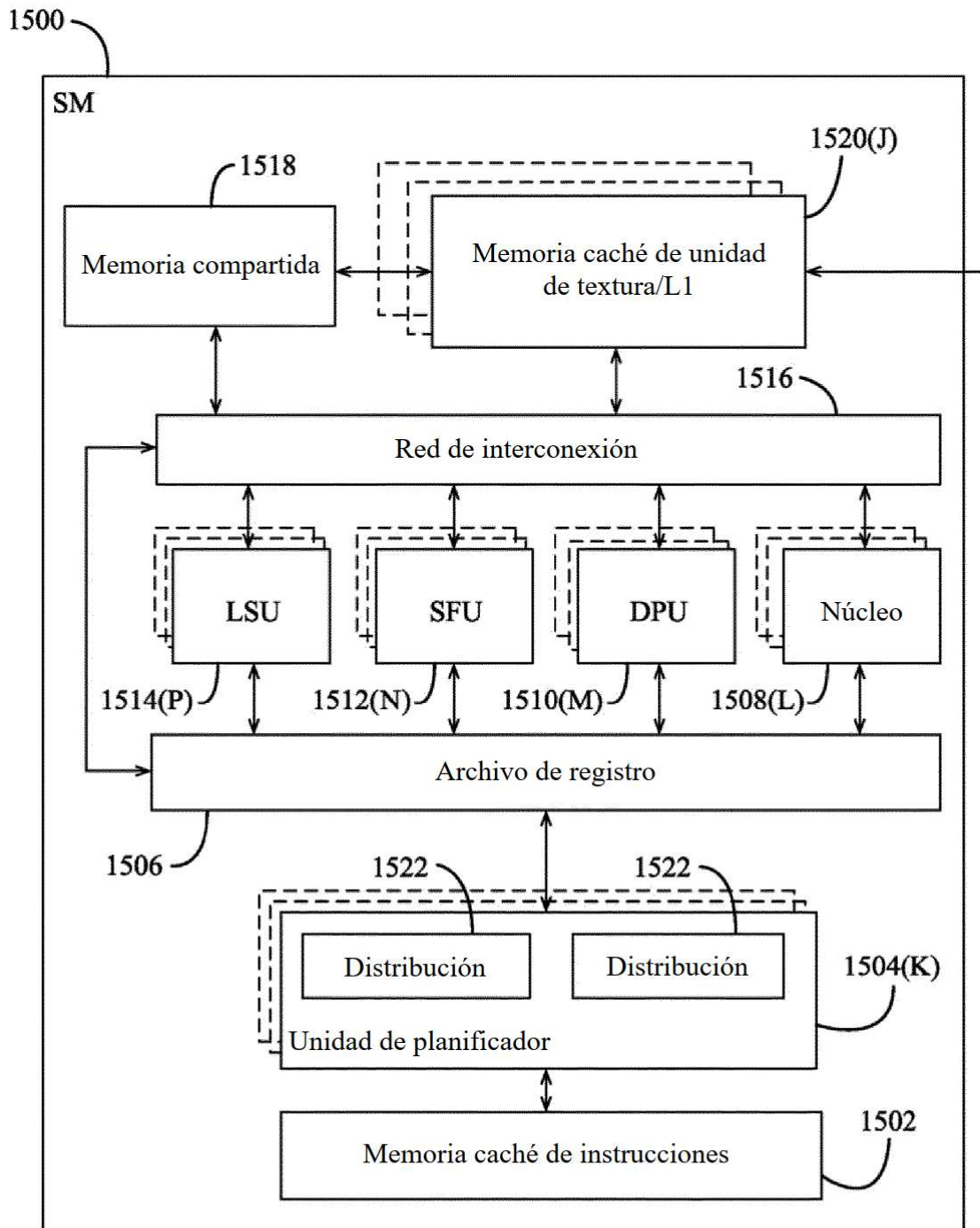


FIG. 15

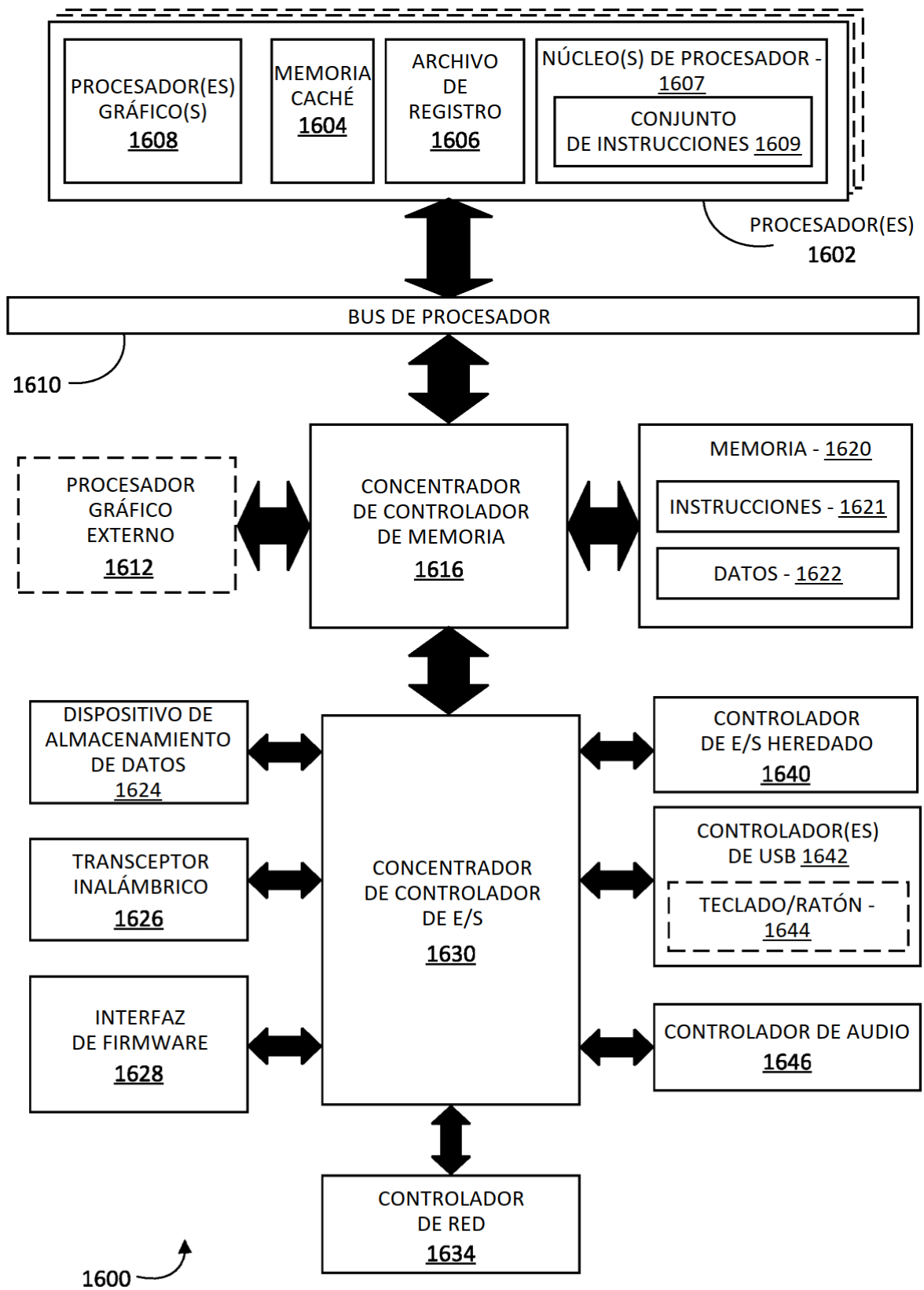


FIG. 16

PROCESADOR 1700 →

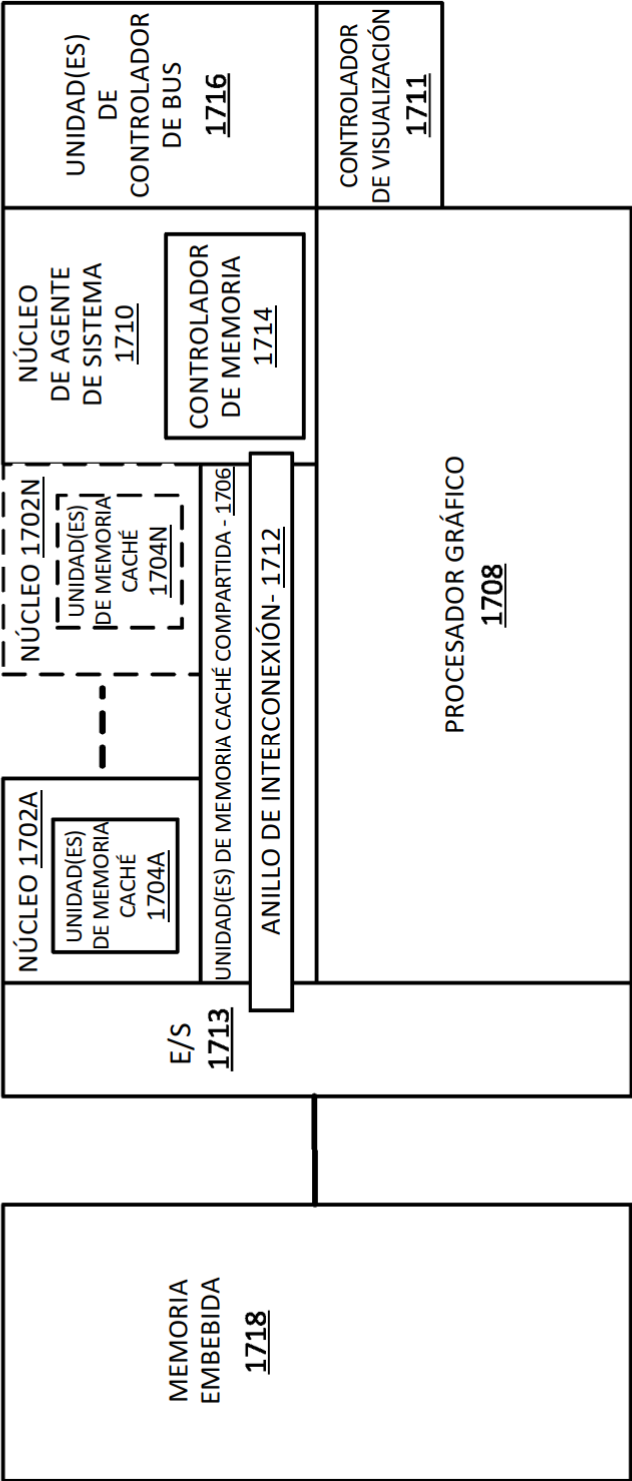


FIG. 17

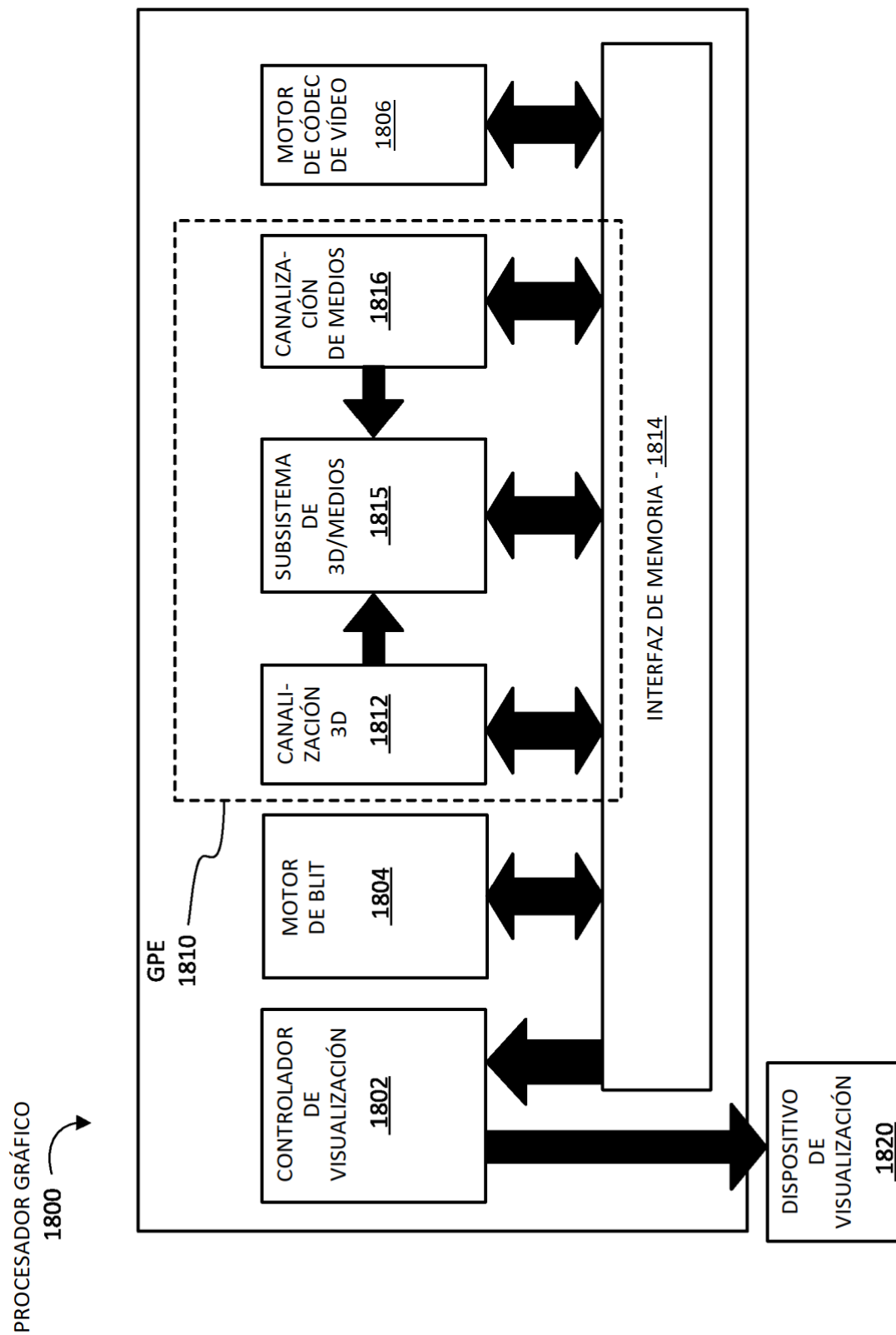


FIG. 18

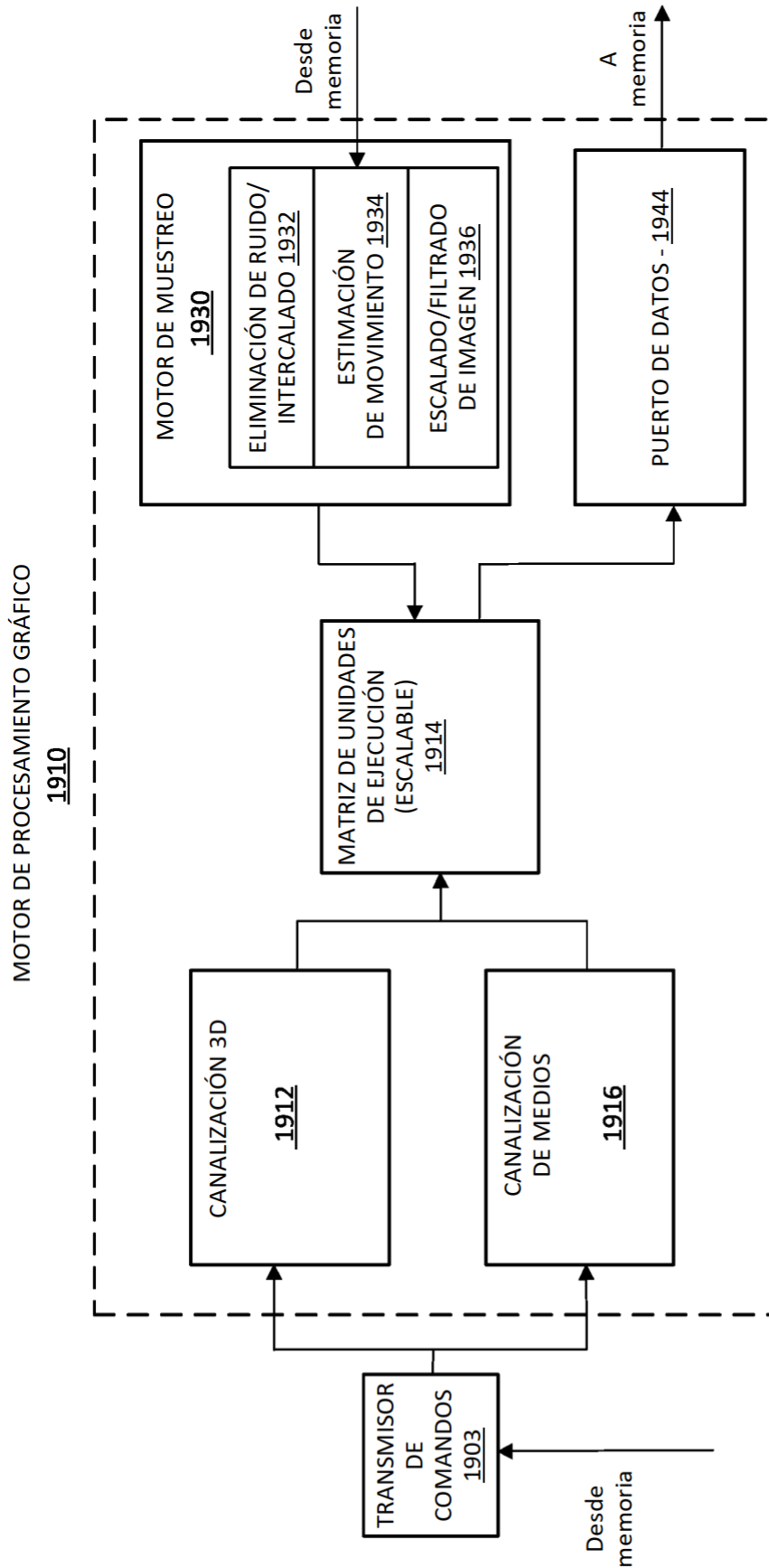


FIG. 19

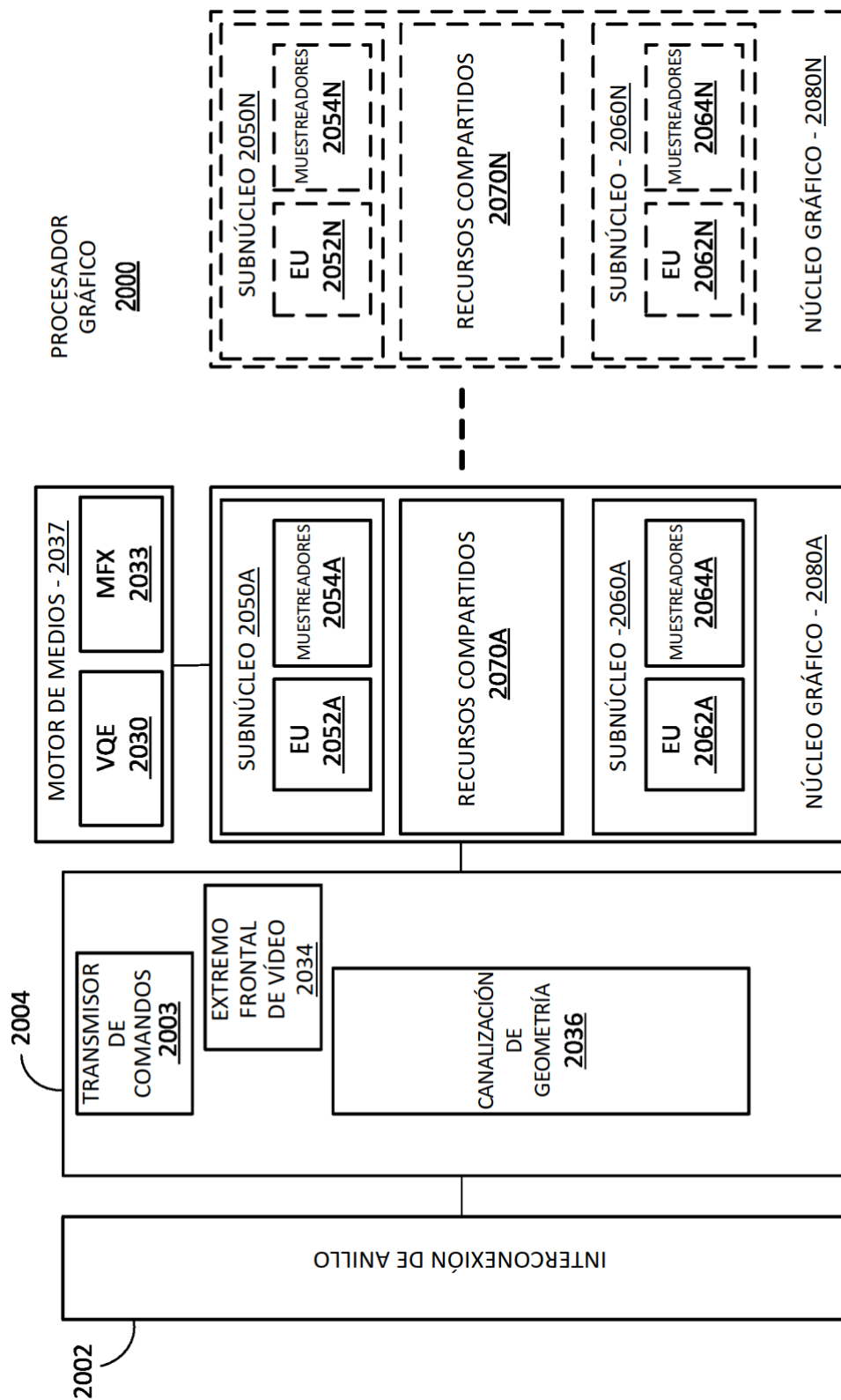


FIG. 20

2100 →

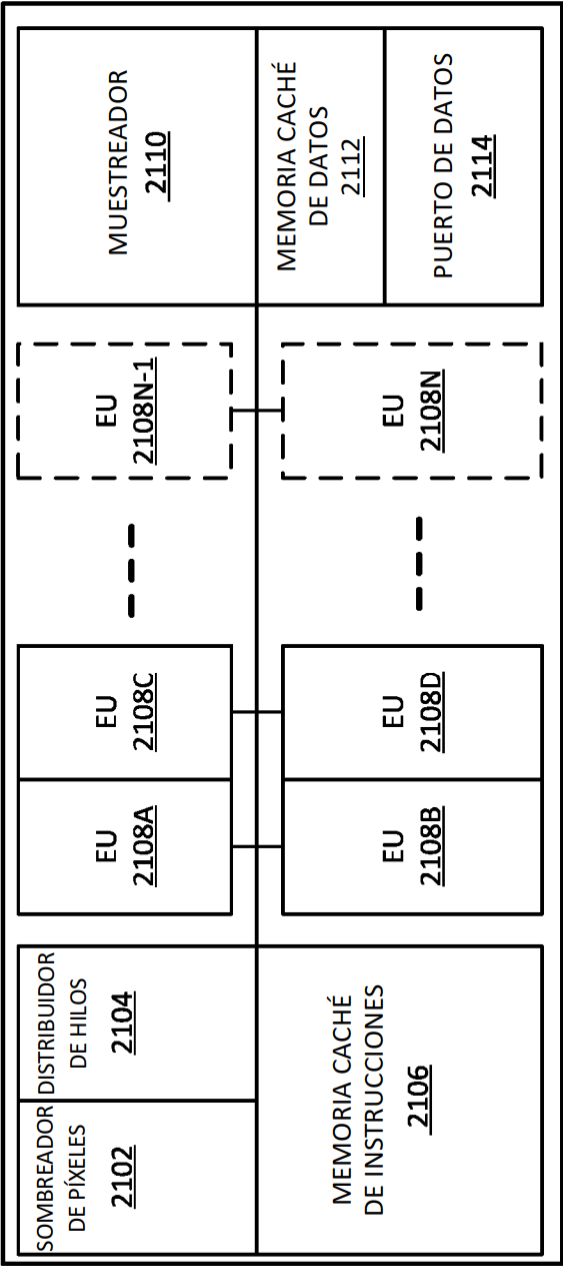
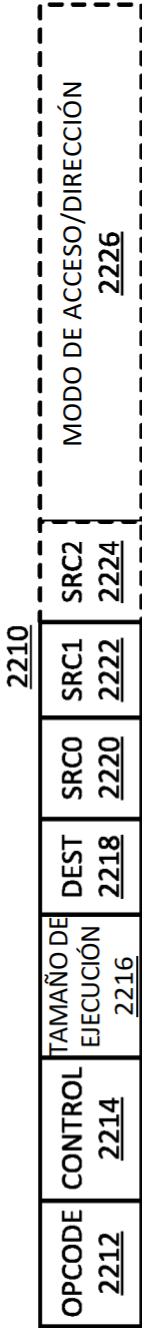


FIG. 21

FORMATOS DE INSTRUCCIÓN
DE PROCESADOR GRÁFICO

2200

INSTRUCCIÓN
DE 128 BITS



INSTRUCCIÓN

COMPACTA DE 64 BITS

2230



DECODIFICACIÓN DE OPCODE

2240

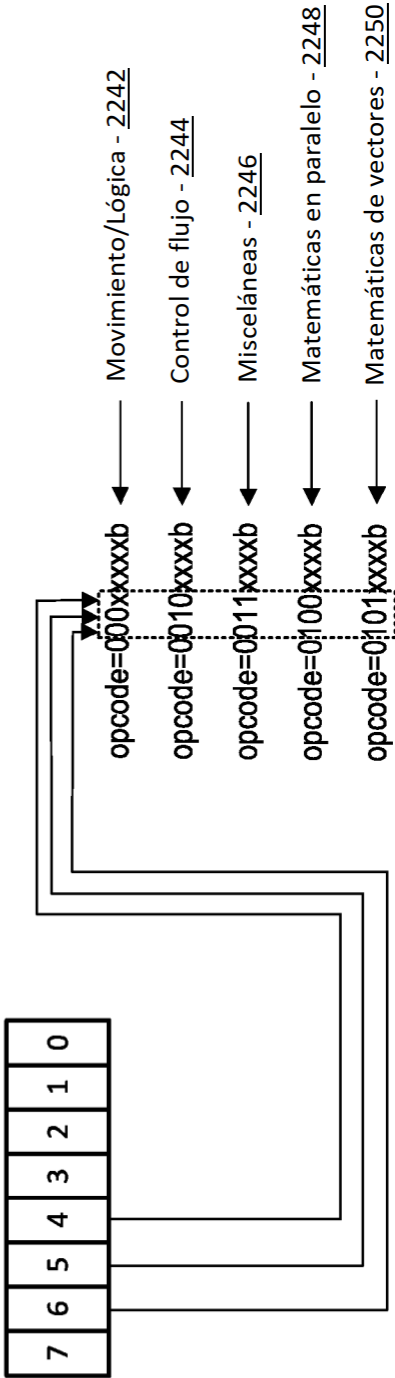


FIG. 22

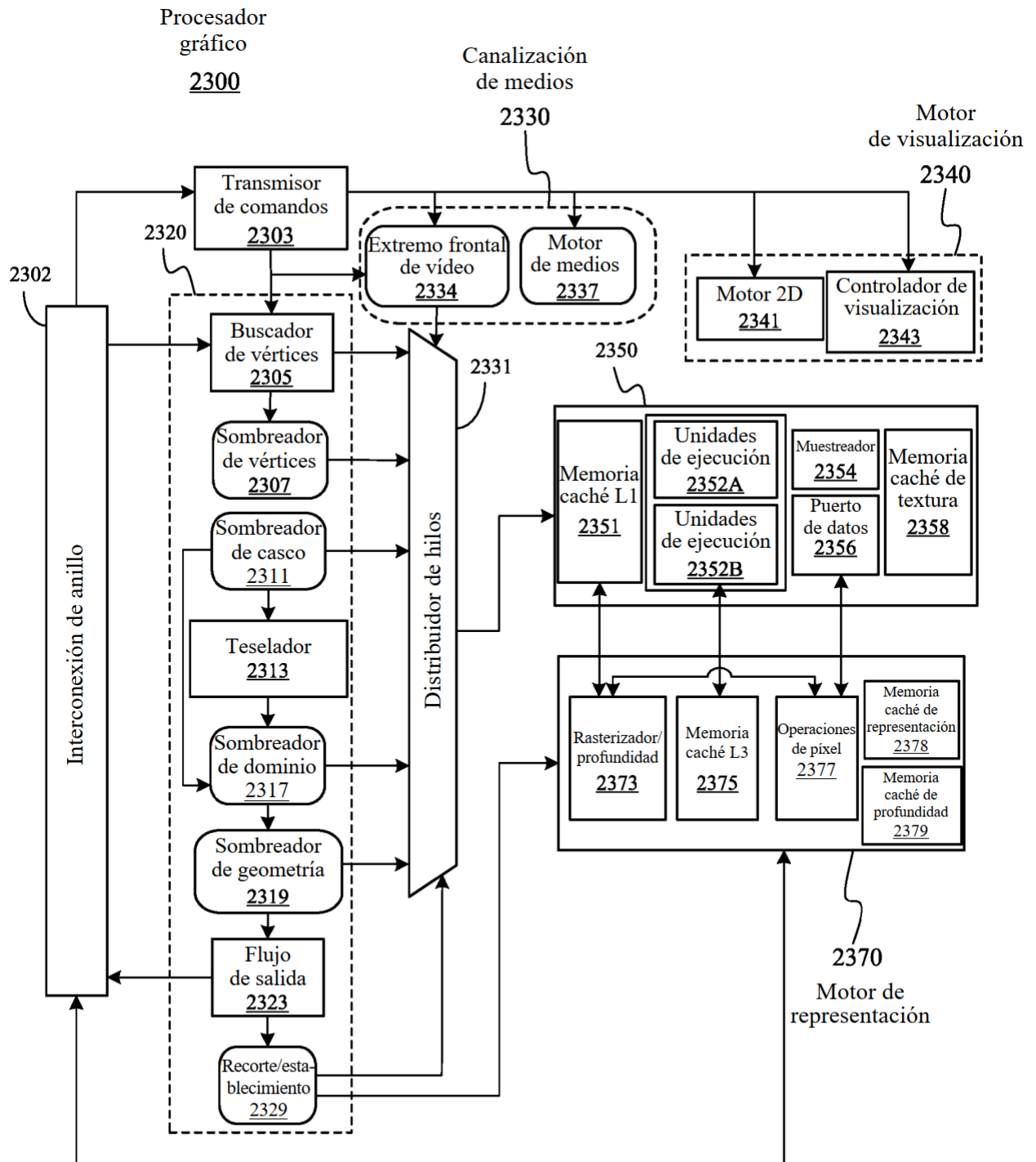


FIG. 23

Formato de comando de procesador gráfico

2400

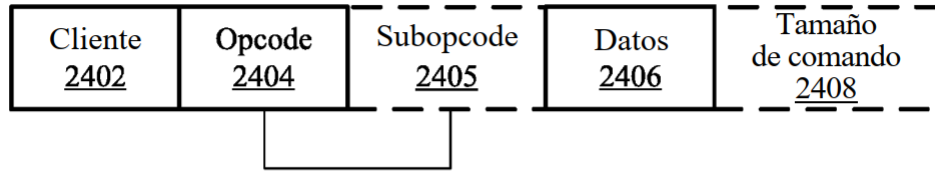


FIG. 24A

Secuencia de comandos de procesador gráfico

2410

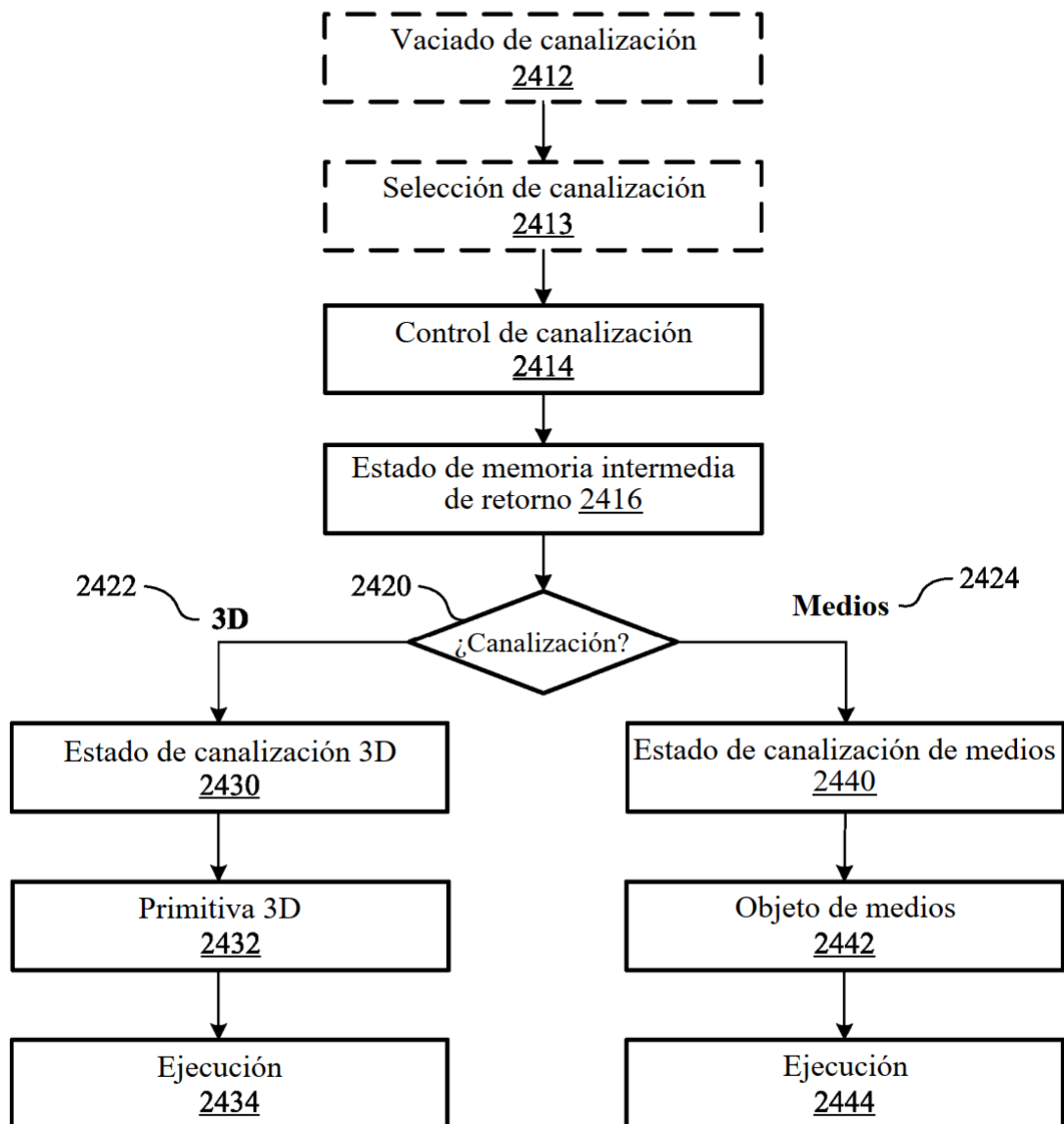


FIG. 24B

Sistema de procesamiento de datos - 2500

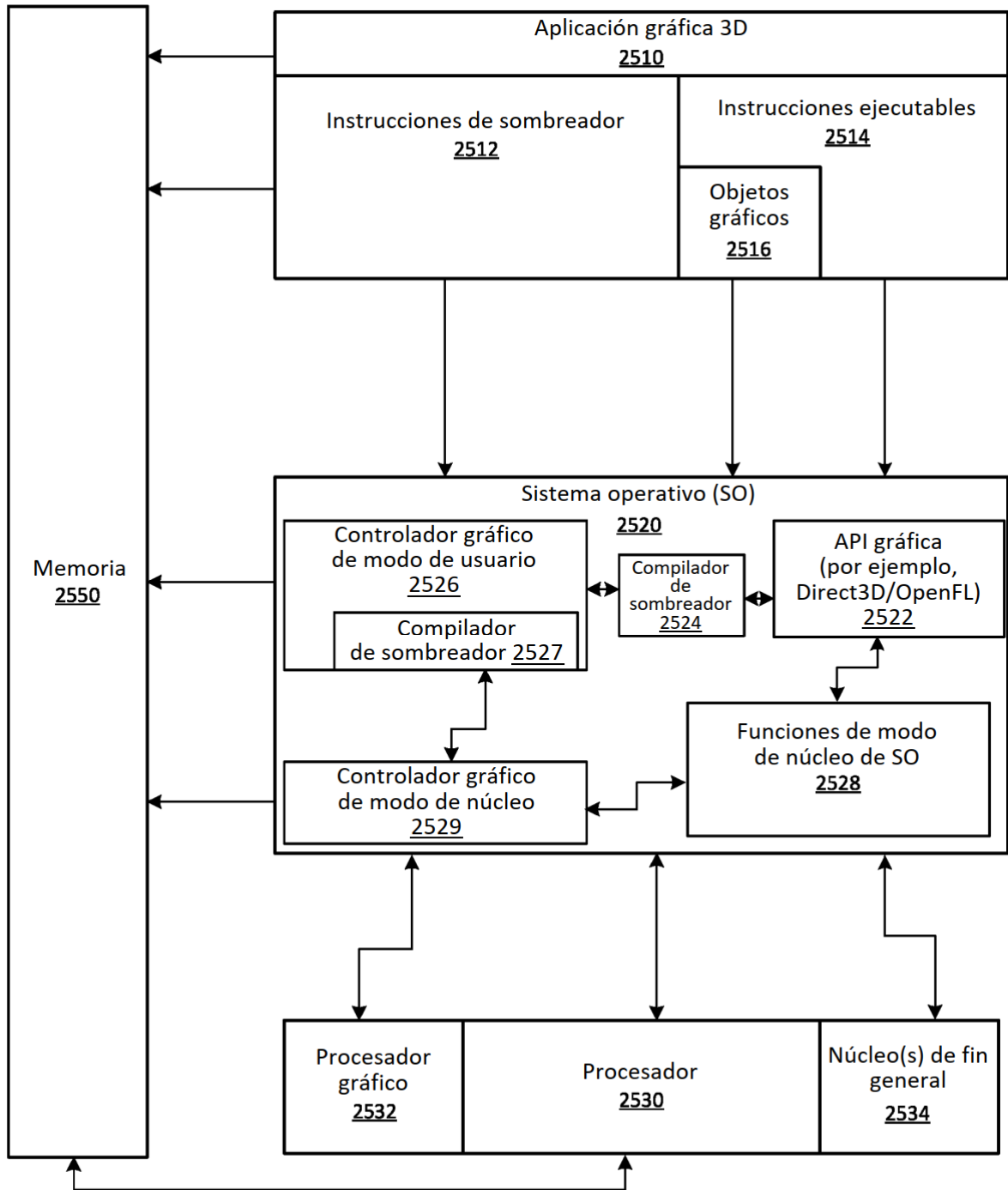


FIG. 25

DESARROLLO DE NÚCLEO DE IP - 2600

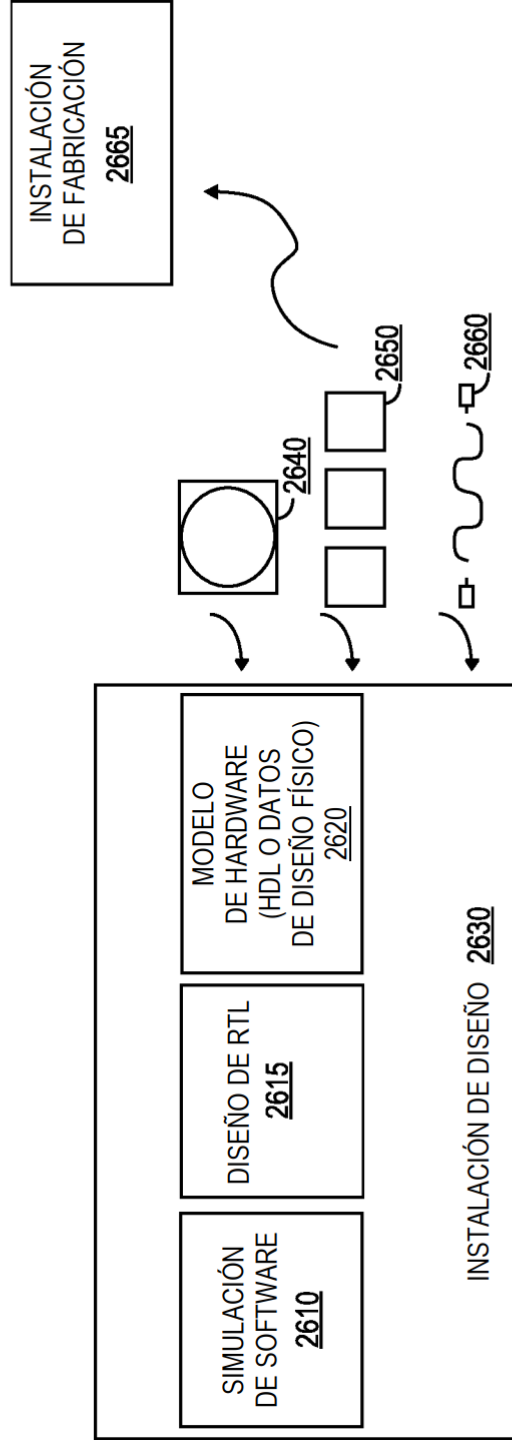


FIG. 26

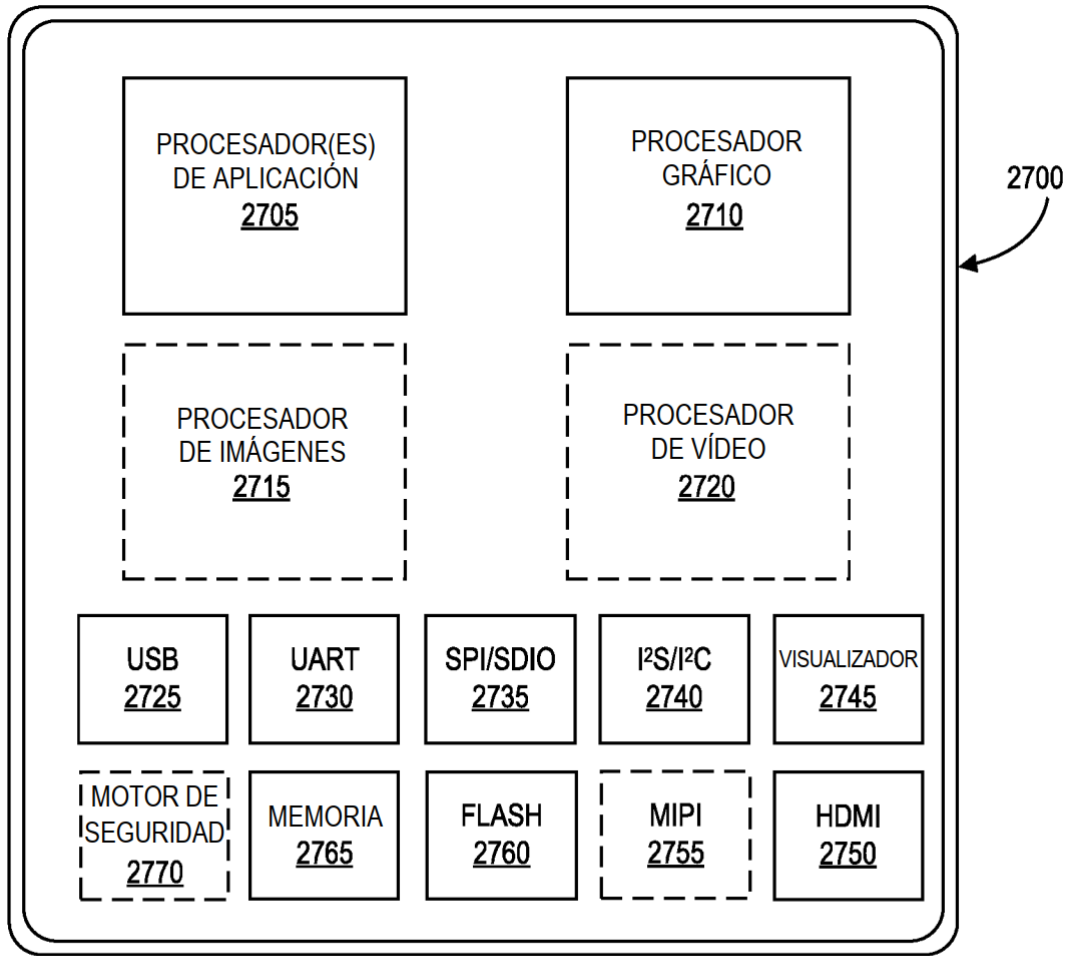


FIG. 27