(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 922 233**

(51) Int. CI.:

G06F 9/30	(2008.01)
G06F 9/38	(2008.01)
G06F 9/46	(2006.01)
G06T 1/20	(2006.01)
G06N 3/063	(2006.01)
G06F 3/14	(2006.01)
G06N 3/04	(2006.01)
G06N 3/08	(2006.01)
G06T 15/00	(2011.01)
G09G 5/36	(2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **02.03.2018 E 19218464 (6)**

(97) Fecha y número de publicación de la concesión europea: **20.04.2022 EP 3657323**

(54) Título: **Mecanismo de optimización de cálculo**

(30) Prioridad:

24.04.2017 US 201715494905

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
12.09.2022

(73) Titular/es:

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

(72) Inventor/es:

APPU, ABHISHEK R.;
KOKER, ALTUG;
HURD, LINDA L.;
KIM, DUKHWAN;
MACPHERSON, MIKE B.;
WEAST, JOHN C.;
CHEN, FENG;
AKHBARI, FARSHAD;
SRINIVASA, NARAYAN;
SATISH, NADATHUR RAJAGOPALAN;
TANG, PING T.;
RAY, JOYDEEP;
STRICKLAND, MICHAEL S.;
CHEN, XIAOMING;
YAO, ANBANG y
SHPEISMAN, TATIANA

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Mecanismo de optimización de cálculo

Campo

Las realizaciones se refieren en general al procesamiento de datos y, más particularmente, a un procesamiento de datos mediante una unidad de procesamiento de gráficos de fin general.

Antecedentes de la descripción

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

Existen algunos enfoques para mejorar el rendimiento de los procesadores de gráficos. El documento US 2006/101244 A1 se refiere a una unidad funcional multipropósito que es configurable para soportar un número de operaciones incluyendo coma flotante y multiplicación-adición de enteros. La unidad funcional multipropósito soporta diferentes formatos de entrada, pero los operandos de diferentes formatos tienen que convertirse para operaciones. El documento US 2015/0378741 A1 describe un método para mejorar potencia, rendimiento, área (PPA) para cálculos de precisión mixtos en un entorno de procesamiento. Las características de arquitectura se definen basándose en un factor de trenzado para reducir el consumo de potencia. El documento US 2016/0026912 A1 pertenece a un mecanismo de desplazamiento de ponderación para unidades de procesamiento reconfigurables que pueden usarse en redes neuronales de convolución. Para mejorar la eficiencia se incluyen circuitos de cálculo modulares que son reconfigurables de acuerdo con las tareas de cálculo.

Para aumentar adicionalmente el rendimiento, los procesadores de gráficos típicamente implementan técnicas de procesamiento, tales como tuberías, que intentan procesar, en paralelo, tantos datos de gráficos como sea posible a través de todas las diferentes partes de la tubería de gráficos. Los procesadores de gráficos paralelos con arquitecturas de única instrucción, múltiples hilos (SIMT) están diseñados para maximizar la cantidad de procesamiento paralelo en la tubería de gráficos. En una arquitectura de SIMT, grupos de hilos paralelos intentan ejecutar instrucciones de programa síncronamente juntas tan a menudo como sea posible para aumentar la eficiencia de procesamiento. Una vista global general de software y hardware para las arquitecturas de SIMT puede encontrarse en Shane Cook, *CUDA Programming*, Capítulo 3, páginas 37-51 (2013).

BREVE DESCRIPCIÓN DE LOS DIBUJOS

De modo que la manera en la que pueden entenderse en detalle las características anteriormente mencionadas de las presentes realizaciones, puede tenerse una descripción más particular, brevemente resumida anteriormente, mediante referencia a las realizaciones, algunas de las cuales se ilustran en los dibujos adjuntos.

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

Las Figuras 2A-2D ilustran componentes de procesador paralelos, de acuerdo con una realización;

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

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

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

La Figura 6 ilustra un dispositivo informático que emplea un mecanismo de optimización, de acuerdo con una realización;

Las Figuras 7A y 7B ilustran realizaciones de un mecanismo de optimización de cálculo;

La Figura 8 ilustra una pila de software de aprendizaje automático, de acuerdo con una realización;

La Figura 9 ilustra una unidad de procesamiento de gráficos de fin general altamente paralelo, de acuerdo con una realización;

La Figura 10 ilustra un sistema informático de múltiples GPU, de acuerdo con una realización;

Las Figuras 11A-11B ilustran capas de redes neuronales profundas ilustrativas;

La Figura 12 ilustra una red neuronal recurrente ilustrativa;

La Figura 13 ilustra el entrenamiento y despliegue de una red neuronal profunda.

La Figura 14 es un diagrama de bloques que ilustra aprendizaje distribuido;

La Figura 15 ilustra un sistema de inferencia en un chip (SOC) ilustrativo adecuado para realizar inferencia usando un modelo entrenado;

La Figura 16 es un diagrama de bloques de un sistema de procesamiento, de acuerdo con una realización;

La Figura 17 es un diagrama de bloques de un procesador de acuerdo con una realización;

La Figura 18 es un diagrama de bloques de un procesador de gráficos, de acuerdo con una realización;

La Figura 19 es un diagrama de bloques de un motor de procesamiento de gráficos de un procesador de gráficos de acuerdo con algunas realizaciones;

La Figura 20 es un diagrama de bloques de un procesador de gráficos proporcionado por una realización adicional;

La Figura 21 ilustra lógica de ejecución de hilos que incluye una matriz de elementos de procesamiento empleada en algunas realizaciones;

La Figura 22 es un diagrama de bloques que ilustra unos formatos de instrucción de procesador de gráficos de acuerdo con algunas realizaciones;

La Figura 23 es un diagrama de bloques de un procesador de gráficos de acuerdo con otra realización;

Las Figuras 24A-24B ilustran un formato de comando de procesador de gráficos y secuencia de comandos, de acuerdo con algunas realizaciones;

La Figura 25 ilustra una arquitectura de software de gráficos ilustrativa para un sistema de procesamiento de datos de acuerdo con algunas realizaciones;

La Figura 26 es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP, de acuerdo con una realización;

La Figura 27 es un diagrama de bloques que ilustra un sistema ilustrativo en un circuito de chip integrado, de acuerdo con una realización;

La Figura 28 es un diagrama de bloques que ilustra un procesador de gráficos ilustrativo adicional; y

La Figura 29 es un diagrama de bloques que ilustra un procesador de gráficos ilustrativo adicional de un sistema en un circuito de chip integrado, de acuerdo con una realización.

DESCRIPCIÓN DETALLADA

La invención se define en las reivindicaciones independientes y se ilustra en las Figuras 6, 7, 9, 15 y 16 y los párrafos asociados en la descripción. Se definen realizaciones preferidas en las reivindicaciones dependientes. En lo que sigue, las realizaciones se refieren únicamente a combinaciones reivindicadas de características. Cuando se usa el término realización para describir combinaciones no reivindicadas de características, el término se entenderá como que hace referencia a ejemplos útiles para entender la presente invención.

En las realizaciones, se divulgan los mecanismos para optimizar el cálculo de un procesador de gráficos. En algunas realizaciones, el mecanismo de cálculo incluye lógica de clasificación para clasificar hilos de procesamiento en grupos de hilos basándose en profundidad de bit de operaciones de hilo de coma flotante. En otras realizaciones, el mecanismo de cálculo incluye lógica flotante para procesar hilos en operaciones de hilo de coma flotante que tienen una mayor profundidad de bit. En realizaciones adicionales, el mecanismo de cálculo incluye lógica para proporcionar soporte de precisión variable en una instrucción de cálculo.

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

documento pueden ponerse en práctica sin uno o más de estos detalles específicos. En otras instancias, no se han descrito características bien conocidas para evitar oscurecer los detalles de las presentes realizaciones.

Vista general del sistema

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

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

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

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

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

Se apreciará que el sistema informático 100 mostrado en el presente documento es ilustrativo y que son posibles variaciones y modificaciones. La topología de conexión, que incluye el número y disposición de los puentes, el número de procesador o procesadores 102 y el número de procesador o procesadores paralelos 112, pueden modificarse según se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 está conectada al procesador o

procesadores 102 directamente en lugar de a través de un puente, mientras que otros dispositivos se comunican con la memoria de sistema 104 mediante el concentrador de memoria 105 y el procesador o procesadores 102. En otras topologías alternativas, el procesador o procesadores paralelos 112 están conectados al concentrador de E/S 107 o directamente a uno del uno o más procesador o procesadores 102, en lugar de al concentrador de memoria 105. En otras realizaciones, el concentrador de E/S 107 y el concentrador de memoria 105 pueden estar integrados en un único chip. Algunas pueden incluir dos o más conjuntos de procesador o procesadores 102 adjuntos mediante múltiples zócalos, que pueden acoplarse con dos o más instancias del procesador o procesadores paralelos 112.

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

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

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

Cuando la interfaz de anfitrión 206 recibe una memoria intermedia de comando mediante la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para realizar estos comandos a un primer extremo frontal 208. En una realización, el extremo frontal 208 se acopla con un planificador 210, que está configurado para distribuir comandos u otros elementos de trabajo a una matriz de agrupación de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupación de procesamiento 212 está configurada apropiadamente y en un estado válido antes de que se distribuyan las tareas a las agrupaciones de procesamiento de la matriz de agrupación de procesamiento 212.

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

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

En una realización, la matriz de agrupación de procesamiento 212 está configurada para realizar operaciones de procesamiento de gráficos paralelas. En las realizaciones en las que el procesador paralelo 200 está configurado para realizar operaciones de procesamiento de gráficos, la matriz de agrupación de procesamiento 212 puede incluir lógica adicional para soportar la ejecución de tales operaciones de procesamiento de gráficos, que incluyen, pero sin limitación, operaciones de lógica de muestreo de textura, así como lógica de teselación y otra lógica de procesamiento de vértice. Adicionalmente, la matriz de agrupación de procesamiento 212 puede estar configurada para ejecutar programas sombreadores relacionados con el procesamiento de gráficos tales como, pero sin limitación, sombreadores de vértices, sombreadores de teselación, sombreadores de geometría y sombreadores de píxeles. La unidad de procesamiento paralelo 202 puede transferir datos de la memoria de sistema mediante la unidad de E/S

204 para su procesamiento. Durante el procesamiento, los datos transferidos pueden almacenarse en una memoria en chip (por ejemplo, la memoria de procesador paralelo 222) durante el procesamiento y, a continuación, volverse a escribir en la memoria de sistema.

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

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

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

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

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

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

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

En aplicaciones de gráficos, la ROP 226 es una unidad de procesamiento que realiza operaciones de rasterización, tales como estarcido, prueba z, mezcla y similares. La ROP 226 emite, a continuación, datos de gráficos procesados que se almacena en memoria de gráficos. En algunas realizaciones, la ROP 226 incluye lógica de compresión 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. 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 una realización de este tipo, se transmiten las solicitudes de escritura y lectura para datos de píxeles a través de la barra transversal de memoria 216 en lugar de datos de fragmento de píxel.

Los datos de gráficos procesados pueden visualizarse en el dispositivo de visualización, tal como uno del uno o más dispositivo o dispositivos de visualización 110 de la Figura 1, encaminarse para su procesamiento adicional por el procesador o procesadores 102, o encaminarse para su procesamiento adicional por una de las entidades de procesamiento dentro del procesador paralelo 200 de la Figura 2A.

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

La operación de la agrupación de procesamiento 214 puede controlarse mediante un gestor de tuberías 232 que distribuye tareas de procesamiento a procesadores paralelos de SIMT. El gestor de tuberías 232 recibe instrucciones del planificador 210 de la Figura 2 y gestiona la ejecución de estas instrucciones mediante un multiprocesador de gráficos 234 y/o una unidad de textura 236. El multiprocesador de gráficos 234 ilustrado es una instancia ilustrativa de un procesador paralelo de SIMT. Sin embargo, pueden incluirse diversos tipos de procesadores paralelos de SIMT de diferentes arquitecturas dentro de la agrupación de procesamiento 214. Puede incluirse una o más instancias del multiprocesador de gráficos 234 dentro de una agrupación de procesamiento 214. El multiprocesador de gráficos 234 puede procesar datos y puede usarse una barra transversal de datos 240 para distribuir los datos procesados a uno de múltiples destinos posibles, que incluyen otras unidades de sombreado. El gestor de tuberías 232 puede facilitar la distribución de datos procesados especificando destinos para datos procesados que van a distribuirse mediante la barra transversal de datos 240.

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

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

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

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

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

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

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

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

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

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

La interconexión de memoria y caché 268 es una red de interconexión que conecta cada una de las unidades funcionales del multiprocesador de gráficos 324 al fichero de registro 258 y a la memoria compartida 270. En una realización, la interconexión de memoria y caché 268 es una interconexión de barra transversal que permite que la unidad de carga/almacén 266 implemente operaciones de carga y almacén entre la memoria compartida 270 y el fichero de registro 258. El fichero de registro 258 puede operar en la misma frecuencia que los núcleos de GPGPU 262, por lo tanto, la transferencia de datos entre los núcleos de GPGPU 262 y el fichero de registro 258 es de latencia muy. La memoria compartida 270 puede usarse para posibilitar la comunicación entre hilos que se ejecutan en las unidades funcionales dentro del multiprocesador de gráficos 234. La memoria caché 272 puede usarse como una caché de datos, por ejemplo, para almacenar en caché datos de textura comunicados entre las unidades funcionales y la unidad de textura 236. La memoria compartida 270 también puede usarse como un programa gestionado en caché. 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 automáticamente en caché que se almacenan dentro de la memoria caché 272.

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

La Figura 3A muestra un multiprocesador de gráficos 325 de acuerdo con una realización adicional. El multiprocesador de gráficos 325 incluye múltiples instancias adicionales de unidades de recursos de ejecución con relación al multiprocesador de gráficos 234 de la Figura 2D. Por ejemplo, el multiprocesador de gráficos 325 puede incluir múltiples instancias de la unidad de instrucciones 332A-332B, del fichero de registro 334A-334B y de la unidad o unidades de textura 344A-344B. El multiprocesador de gráficos 325 también incluye múltiples conjuntos de unidades de ejecución de gráficos o de cálculo (por ejemplo, el núcleo de GPGPU 336A-336B, el núcleo de GPGPU 337A-337B, el núcleo de GPGPU 338A-338B) y múltiples conjuntos de las unidades de carga/almacén 340A-340B. En una realización, las unidades de recurso de ejecución tienen una caché de instrucciones 330 común, memoria caché de textura y/o datos 342 y memoria compartida 346. Los diversos componentes pueden comunicarse mediante un tejido de interconexión 327. En una realización, el tejido de interconexión 327 incluye uno o más conmutadores de barra transversal para posibilitar la comunicación entre los diversos componentes del multiprocesador de gráficos 325.

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

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

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

Técnicas para interconexión de GPU a procesador anfitrión

La Figura 4A ilustra una arquitectura ilustrativa en la que una pluralidad de GPU 410-413 se 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 que incluyen, pero sin limitación, PCIe 4.0 o 5.0 y NVLink 2.0. Sin embargo, los principios subyacentes de la invención no están limitados a ningún protocolo o caudal de comunicación particular.

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

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

Como se describe a continuación, aunque los diversos procesadores 405-406 y las GPU 410-413 pueden estar físicamente acoplados a una memoria particular 401-402, 420-423, respectivamente, puede implementarse una arquitectura de memoria unificada en la que el mismo espacio de direcciones de sistema virtual (también denominado como el espacio de "direcciones efectivo") está distribuido entre todas las diversas memorias físicas. Por ejemplo, las

memorias de procesador 401-402 pueden comprender cada una 64 GB del espacio de direcciones de memoria de sistema y las memorias de GPU 420-423 puede comprender cada una 32 GB del espacio de direcciones de memoria de sistema (dando como resultado un total de 256 GB de memoria direccionable en este ejemplo).

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

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

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

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

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

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

Un conjunto de registros 445 almacena datos de contexto para hilos ejecutados por los motores de procesamiento de gráficos 431-432, N y un circuito de gestión de contexto 448 gestiona los contextos de hilo. Por ejemplo, el circuito de gestión de contexto 448 puede realizar operaciones de grabación y restauración para grabar y restaurar contextos de los diversos hilos durante conmutaciones de contexto (por ejemplo, cuando se graba el primer hilo y se almacena un segundo hilo de modo que el segundo hilo puede ejecutarse por un motor de procesamiento de gráficos). Por ejemplo,

en una conmutación de contexto, el circuito de gestión de contexto 448 puede almacenar valores de registro actuales para una región designada en memoria (por ejemplo, identificada por un puntero de contexto). A continuación, puede restaurar los valores de registro al volver al contexto. En una realización, un circuito de gestión de interrupción 447 recibe y procesa interrupciones recibidas de dispositivos de sistema.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Tabla 1 - Registros de hipervisor inicializados

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

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

Tabla 2 - Registros de sistema operativo inicializados

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

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

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

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

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

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

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

Tabla 3 - parámetros de llamada de SO a Hipervisor

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

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

Tabla 4 - Información de elemento de proceso

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

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

Como se ilustra en la **Figura 4F,** una realización de la invención emplea una memoria unificada direccionable mediante un espacio de direcciones de memoria virtual común usado para acceder a las memorias de procesador físicas 401-402 y a las memorias de GPU 420-423. En esta implementación, las operaciones ejecutadas en las GPU 410-413 utilizan el mismo espacio de direcciones de memoria virtual/efectivo para acceder a las memorias de los procesadores 401-402 y viceversa, simplificando de esta manera la capacidad de la programación. En una realización, una primera porción del espacio de direcciones virtual/efectivo está asignada a la memoria del procesador 401, una segunda porción a la segunda memoria del procesador 402, una tercera porción a la memoria de la GPU 420, y así sucesivamente. El espacio de memoria virtual/efectivo completo (en ocasiones denominado espacio de direcciones efectivo) se distribuye de esta manera a través de cada una de las memorias de procesador 401-402 y las memorias de GPU 420-423, permitiendo que cualquier procesador o GPU acceda a cualquier memoria física con una dirección virtual mapeada a esa memoria.

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

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

de banda de escritura efectivo observado por una GPU 410-413. La eficacia de la configuración del operando, la eficacia de los resultados de acceso, y la eficacia del cálculo de GPU todos desempeñan un papel al determinar la efectividad de la descarga de la GPU.

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

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

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

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

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

Tubería de procesamiento de gráficos

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

En una realización, el ensamblador de datos 502 es una unidad de procesamiento que recopila datos de vértices para superficies y primitivas. El ensamblador de datos 502 emite, a continuación, los datos de vértices, incluyendo los

atributos de vértices, a la unidad de procesamiento de vértice 504. La unidad de procesamiento de vértice 504 es una unidad de ejecución programable que ejecuta programas de sombreador de vértices, iluminando y transformando datos de vértices como se especifica por los programas sombreadores de vértices. La unidad de procesamiento de vértice 504 lee datos que se almacenan en memoria caché, local o de sistema para su uso en el procesamiento de los datos de vértices y puede programarse para transformar los datos de vértices desde una representación de coordenadas basadas en objeto a un espacio de coordenadas de espacio mundial o un espacio de coordenadas de dispositivo normalizado.

Una primera instancia de un ensamblador de primitivas 506 recibe atributos de vértices desde la unidad de procesamiento de vértice 50. Las lecturas del ensamblador de primitivas 506 almacenaron atributos de vértices según sea necesario y construye primitivas de gráficos para su procesamiento por la unidad de procesamiento de control de teselación 508. Las primitivas de gráficos incluyen triángulos, segmentos de línea, puntos, parches y así sucesivamente, según se soportan por los diversas interfaces de programación de aplicación (API) de procesamiento de gráficos.

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

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

En algunas realizaciones, la unidad de procesamiento de geometría 516 puede añadir o borrar elementos en el flujo de geometría. La unidad de procesamiento de geometría 516 emite los parámetros y vértices que especifican nuevas primitivas de gráficos al ensamblador de primitivas 518. El ensamblador de primitivas 518 recibe los parámetros y vértices desde la unidad de procesamiento de geometría 516 y construye primitivas de gráficos para su procesamiento por una unidad de escala, selección y recorte de ventana gráfica 520. La unidad de procesamiento de geometría 516 lee datos que se almacenan en la memoria de procesador paralelo o la memoria de sistema para su uso en el procesamiento de los datos de geometría. La unidad de escala, selección y recorte de ventana gráfica 520 realiza recorte, selección y escalado de ventana gráfica y emite las primitivas de gráficos procesadas a un rasterizador 522. El rasterizador 522 puede realizar selección de profundidad y otras optimizaciones basadas en profundidad. El rasterizador 522 también realiza conversión de exploración en las nuevas primitivas de gráficos para generar fragmentos y emitir esos fragmentos y datos de cobertura asociados a la unidad de procesamiento de fragmento/píxel 524. La exploración del rasterizador 522 convierte las primitivas de gráficos nuevas y emite datos de fragmentos y cobertura a la unidad de procesamiento de fragmentos/píxeles 524.

La unidad de procesamiento de fragmentos/píxeles 524 es una unidad de ejecución programable que está configurada para ejecutar programas sombreadores de fragmentos o programas sombreadores de píxeles. La unidad de procesamiento de fragmentos/píxeles 524 que transforma fragmentos o píxeles recibidos desde el rasterizador 522, según se especifica por los programas sombreadores de fragmentos o de píxeles. Por ejemplo, la unidad de procesamiento de fragmento/píxel 524 puede programarse para realizar operaciones incluidas, pero sin limitación a, mapeo de textura, sombreado, mezcla, corrección de textura y corrección de perspectiva para producir fragmentos o píxeles sombreados que se emiten a una unidad de operaciones de rasterización 526. La unidad de procesamiento de fragmento/píxel 524 puede leer datos que se almacenan en o bien la memoria de procesador paralelo o bien la memoria de sistema para su uso cuando se procesan los datos de fragmento. Los programas de sombreador de fragmento o de píxel pueden estar configurados para sombrear en granularidades de muestra, píxel, mosaico u otra, dependiendo de la tasa de muestreo configurada para las unidades de procesamiento.

La unidad de operaciones de rasterización 526 es una unidad de procesamiento que realiza operaciones de rasterización que incluyen, pero sin limitación a, estarcido, prueba z, mezcla y similares, y emite datos de píxel como datos de gráficos procesados que hay que almacenar en memoria de gráficos (por ejemplo, la memoria de procesador paralelo 222 como en la Figura 1, que hay que visualizar en el uno o más dispositivo o dispositivos de visualización

110 o para su procesamiento adicional por uno del uno o más procesador o procesadores 102 o procesador o procesadores paralelos 112. En algunas realizaciones, la unidad de operaciones de rasterización 526 está configurada para comprimir z o datos de color que se escriben en memoria y descomprimir z o datos de color que se leen desde la memoria.

La Figura 6 ilustra una realización de un dispositivo informático 600 que emplea un mecanismo de optimización de cálculo. El dispositivo informático 600 (por ejemplo, dispositivos ponibles inteligentes, dispositivos de realidad virtual (RV), dispositivo montado en la cabeza (HMD), ordenadores móviles, dispositivos de Internet de las Cosas (IoT), ordenadores portátiles, ordenadores de sobremesa, ordenadores de servidor, etc.) puede ser el mismo que el sistema de procesamiento de datos 100 de la Figura 1 y por consiguiente, por brevedad, claridad y facilidad de entendimiento, muchos de los detalles indicados anteriormente con referencia a las Figuras 1-5 no se analizan o repiten adicionalmente en lo sucesivo. Como se ilustra, en una realización, el dispositivo informático 600 se muestra como que aloja un mecanismo (de cálculo) de optimización de cálculo 610.

Como se ilustra, en una realización, el mecanismo de cálculo 610 puede alojarse por la unidad de procesamiento de gráficos (GPU) 614. Sin embargo, en otras realizaciones, el mecanismo de cálculo 610 puede alojarse por o ser parte de firmware del controlador de gráficos 616. En otras realizaciones más, el mecanismo de cálculo 610 puede alojarse por o ser parte de firmware de la unidad central de procesamiento ("CPU" o "procesador de aplicación") 612. Por brevedad, claridad y facilidad de entendimiento, a lo largo de todo el resto de este documento, el mecanismo de cálculo 610 puede analizarse como parte de la GPU 614; sin embargo, las realizaciones no se limitan como tales.

En otra realización más, el mecanismo de cálculo 610 puede alojarse como lógica de software o firmware por el sistema operativo 606. En una realización adicional más, el mecanismo de cálculo 610 puede alojarse parcial y simultáneamente por múltiples componentes del dispositivo informático 600, tal como uno o más del controlador de gráficos 616, la GPU 614, el firmware de GPU, la CPU 612, el firmware de CPU, el sistema operativo 606 y/o similares. Se contempla que el mecanismo de cálculo 610 o uno o más de sus componentes pueden implementarse como hardware, software y/o firmware.

A lo largo de todo el documento, el término "usuario" puede denominarse indistintamente "espectador", "observador", "persona", "individuo", "usuario final" y/o similares. Se ha de observar que a lo largo de todo este documento, puede hacerse referencia a términos como "dominio de gráficos" indistintamente con "unidad de procesamiento de gráficos", "procesador de gráficos", o simplemente "GPU" y, de manera similar, puede hacerse referencia a "dominio de CPU" o "dominio de anfitrión" indistintamente con "unidad de procesamiento de ordenador", "procesador de aplicación" o simplemente "CPU".

El dispositivo informático 600 puede incluir cualquier número y tipo de dispositivos de comunicación, tales como grandes sistemas informáticos, tales como ordenadores de servidor, ordenadores de sobremesa, etc., y puede incluir adicionalmente decodificadores de salón (por ejemplo, decodificadores de salón de televisión por cable basados en internet, etc.), dispositivos basados en sistema de posicionamiento global (GPS), etc. El dispositivo informático 600 puede incluir dispositivos informáticos móviles que sirven como dispositivos de comunicación, tales como teléfonos celulares que incluyen teléfonos inteligentes, asistentes digitales personales (PDA), ordenadores de tableta, ordenadores portátiles, lectores electrónicos, televisiones inteligentes, plataformas de televisión, dispositivos ponibles (por ejemplo, gafas, relojes, pulseras, tarjetas inteligentes, joyería, artículos de rota, etc.), reproductores de medios, etc. Por ejemplo, en una realización, el dispositivo informático 600 puede incluir un dispositivo informático móvil que emplea una plataforma informática que aloja un circuito integrado ("CI"), tal como un sistema en un chip ("SoC" o "SOC"), que integra diversos componentes de hardware y/o software del dispositivo informático 600 en un único chip.

Como se ilustra, en una realización, el dispositivo informático 600 puede incluir cualquier número y tipo de componentes de hardware y/o software, tales como (sin limitación) la GPU 614, el controlador de gráficos (también denominado "controlador de GPU", "lógica de controlador de gráficos", "lógica de controlador", controlador de modo de usuario (UMD), UMD, marco de controlador de modo de usuario (UMDF), UMDF o simplemente "controlador") 616, la CPU 612, la memoria 608, dispositivos de red, controladores o similares, así como fuentes de entrada/salida (E/S) 604, tales como pantallas táctiles, paneles táctiles, placas táctiles, teclados virtuales o regulares, ratón virtual o regular, puertos, conectores, etc.

El dispositivo informático 600 puede incluir el sistema operativo (SO) 606 que sirve como una interfaz entre hardware y/o recursos físicos del dispositivo informático 600 y un usuario. Se contempla que la CPU 612 puede incluir uno o más procesadores, tales como el procesador o procesadores 102 de la Figura 1, mientras que la GPU 614 puede incluir uno o más procesadores de gráficos (o multiprocesadores).

Se ha de observar que términos como "nodo", "nodo informático", "servidor", "dispositivo de servidor", "ordenador en la nube", "servidor en la nube", "ordenador de servidor en la nube", "máquina", "máquina de anfitrión", "dispositivo", "dispositivo informático", "ordenador", "sistema informático" y similares pueden usarse indistintamente a lo largo de todo este documento. Se ha de observar adicionalmente que términos como "aplicación", "aplicación de software", "programa", "programa de software", "paquete", "paquete de software" y similares pueden usarse indistintamente a lo

largo de todo este documento. También, términos como "trabajo", "entrada", "solicitud", "mensaje" y similares pueden usarse indistintamente a lo largo de todo este documento.

Se contempla y como se describe adicionalmente con referencia a las Figuras 1-5, algunos procesos de la tubería de gráficos como se ha descrito anteriormente se implementan en software, mientras que el resto se implementan en hardware. Una tubería de gráficos puede implementarse en un diseño de coprocesador de gráficos, en el que la CPU 612 se diseña para trabajar con la GPU 614 que puede incluirse en o coubicarse con la CPU 612. En una realización, la GPU 614 puede emplear cualquier número y tipo de lógica de software y hardware convencional para realizar las funciones convencionales relacionadas con la renderización de gráficos así como la lógica de software y hardware novedosa para ejecutar cualquier número y tipo de instrucciones.

Como se ha mencionado anteriormente, la memoria 608 puede incluir una memoria de acceso aleatorio (RAM) que comprende una base de datos de aplicación que tienen información de objeto. Un concentrador de controlador de memoria, tal como el concentrador de memoria 105 de la Figura 1, puede acceder a datos en la RAM y reenviar los mismos a la GPU 614 para el procesamiento de tubería de gráficos. La RAM puede incluir RAM de doble tasa de datos (DDR RAM), RAM de salida de datos extendida (EDO RAM), etc. La CPU 612 interactúa con una tubería de gráficos de hardware para compartir una funcionalidad de tuberías de gráficos.

Los datos procesados se almacenan en una memoria intermedia en la tubería de gráficos de hardware, e información de estado se almacena en la memoria 608. La imagen resultante se transfiere, a continuación, a las fuentes de E/S 604, tales como un componente de visualización para la visualización de la imagen. Se contempla que el dispositivo de visualización puede ser de diversos tipos, tales como Tubo de Rayos Catódicos (CRT), Transistor de Película Delgada (TFT), Pantalla de Cristal Líquido (LCD), matriz de Diodo Orgánico Emisor de Luz (OLED), etc., para visualizar información a un usuario.

La memoria 608 puede comprender una región preasignada de una memoria intermedia (por ejemplo, memoria intermedia de trama); sin embargo, debería entenderse por un experto en la materia que las realizaciones no están así limitadas, y que puede usarse cualquier memoria accesible para la tubería de gráficos inferior. El dispositivo informático 600 puede incluir adicionalmente el concentrador de control de entrada/salida (E/S) (ICH) 107 como se hace referencia en la Figura 1, como una o más fuentes de E/S 604, etc.

La CPU 612 puede incluir uno o más procesadores para ejecutar instrucciones para realizar cualesquiera rutinas de software que implementa el sistema informático. Las instrucciones implican frecuentemente alguna clase de operación realizada en datos. Tanto datos como instrucciones pueden almacenarse en la memoria de sistema 608 y cualquier caché asociada. La caché se diseña habitualmente para tener tiempos de latencia más cortos que la memoria de sistema 608; por ejemplo, la caché podría integrarse en el mismo chip o chips de silicio que el procesador o procesadores y/o construirse con células de RAM estática (SRAM) más rápidas, mientras que la memoria de sistema 608 podría construirse con células de RAM dinámica (DRAM) más lentas. Tendiendo a almacenar las instrucciones y datos usados más frecuentemente en la caché al contrario que la memoria de sistema 608 mejora la eficiencia de rendimiento global del dispositivo informático 600. Se contempla que en algunas realizaciones, la GPU 614 puede existir como parte de la CPU 612 (tal como parte de un paquete de CPU físico) en cuyo caso, la memoria 608 puede compartirse por la CPU 612 y la GPU 614 o mantenerse separada.

La memoria de sistema 608 puede hacerse disponible a otros componentes dentro del dispositivo informático 600. Por ejemplo, cualquier dato (por ejemplo, datos de gráficos de entrada) recibido desde diversas interfaces al dispositivo informático 600 (por ejemplo, teclado y ratón, puerto de impresión, puerto de Red de Área Local (LAN), puerto de módem, etc.) o recuperado desde un elemento de almacenamiento interno del dispositivo informático 600 (por ejemplo, unidad de disco duro) se pone en cola a menudo temporalmente en la memoria de sistema 608 antes de operarse por el uno o más procesador o procesadores en la implementación de un programa de software. De manera similar, los datos que un programa de software determina que deberían enviarse desde el dispositivo informático 600 a una entidad externa a través de una de las interfaces de sistema informático, o almacenarse en un elemento de almacenamiento interno, se ponen el cola a menudo temporalmente en la memoria de sistema 608 antes de transmitirse o almacenarse.

Además, por ejemplo, puede usarse un ICH para asegurar que tales datos se pasan apropiadamente entre la memoria de sistema 608 y su correspondiente interfaz de sistema informático apropiado (y dispositivo de almacenamiento interno si el sistema informático se diseña así) y puede tener enlaces de punto a punto bidireccionales entre sí mismo y las fuentes/dispositivos de E/S 604 observados. De manera similar, puede usarse un MCH para gestionar las diversas solicitudes en disputa para accesos de la memoria de sistema 608 entre la CPU 612 y la GPU 614, interfaces y elementos de almacenamiento internos que pueden surgir aproximadamente en tiempo el uno con respecto del otro.

Las fuentes de E/S 604 pueden incluir uno o más dispositivos de E/S que se implementan para transferir datos a y/o desde el dispositivo informático 600 (por ejemplo, un adaptador de red); o para un almacenamiento no volátil a gran escala dentro del dispositivo informático 600 (por ejemplo, unidad de disco duro). El dispositivo de entrada de usuario, incluyendo teclas alfanuméricas y otras, puede usarse para comunicar información y selecciones de comandos a la GPU 614. Otro tipo de dispositivo de entrada de usuario es un control de cursor, tal como un ratón, una bola de mando,

una pantalla táctil, un panel táctil o teclas de dirección de cursor para comunicar información de dirección y selecciones de comandos a la GPU 614 y para controlar el movimiento de cursor en el dispositivo de visualización. Pueden emplearse matrices de cámara y micrófono del dispositivo informático 600 para observar gestos, grabar audio y vídeo y para recibir y transmitir comandos visuales y audio.

El dispositivo informático 600 puede incluir adicionalmente interfaz o interfaces de red para proporcionar acceso a una red, tal como una LAN, una red de área extensa (WAN), una red de área metropolitana (MAN), una red de área personal (PAN), Bluetooth, una red en la nube, una red móvil (por ejemplo, 3ª Generación (3G), 4ª Generación (4G), etc.), una intranet, la Internet, etc. La interfaz o interfaces de red pueden incluir, por ejemplo, una interfaz de red inalámbrica que tiene una antena, que puede representar una o más antena o antenas. La interfaz o interfaces de red también pueden incluir, por ejemplo, una interfaz de red por cable para comunicarse con dispositivos remotos mediante cable de red, que puede ser, por ejemplo, un cable de Ethernet, un cable coaxial, un cable de fibra óptica, un cable en serie o un cable paralelo.

La interfaz o interfaces de red pueden proporcionar acceso a una LAN, por ejemplo, cumpliendo con las normas IEEE 802.11b y/o IEEE 802.11g, y/o la interfaz de red inalámbrica puede proporcionar acceso a una red de área personal, por ejemplo, cumpliendo con las normas Bluetooth. También pueden soportarse otras interfaces de red inalámbrica y/o protocolos, incluyendo versiones anteriores y posteriores de las normas. Además de, o en lugar de, comunicación mediante las normas de LAN inalámbricas, la interfaz o interfaces de red pueden proporcionar comunicación inalámbrica usando, por ejemplo, protocolos de Acceso Múltiple por División de Tiempo (TDMA), protocolos de Sistemas Globales para Comunicaciones Móviles (GSM), protocolos de Acceso Múltiple por División de Código (CDMA) y/o cualquier otro tipo de protocolos de comunicaciones inalámbricas.

La interfaz o interfaces de red pueden incluir una o más interfaces de comunicación, tales como un módem, una tarjeta de interfaz de red u otros dispositivos de interfaz bien conocidos, tales como los usados para acoplar a la Ethernet, anillo de testigo u otros tipos de conexiones por cable o inalámbricas para propósitos de proporcionar un enlace de comunicación para soportar una LAN o una WAN, por ejemplo. De esta manera, el sistema informático puede acoplarse también a un número de dispositivos periféricos, clientes, superficies de control, consolas o servidores mediante una infraestructura de red convencional, incluyendo una Intranet o la Internet, por ejemplo.

Se ha de apreciar que para ciertas implementaciones puede preferirse un sistema más o menos equipado que el ejemplo descrito anteriormente. Por lo tanto, la configuración de dispositivo informático 600 puede variar de implementación a implementación dependiendo de numerosos factores, tales como restricciones de precio, requisitos de rendimiento, mejoras tecnológicas u otras circunstancias. Ejemplos del dispositivo electrónico o del sistema informático 600 pueden incluir (sin limitación) un dispositivo móvil, un asistente digital personal, un dispositivo informático móvil, un teléfono inteligente, un teléfono celular, un microteléfono, un buscador bidireccional, un dispositivo de radiobúsqueda bidireccional, un dispositivo de mensajería, un ordenador, un ordenador personal (PC), un ordenador de sobremesa, un ordenador portátil, un ordenador portátil ligero, un ordenador de bolsillo, un ordenador de tableta, un servidor, una matriz de servidores o parque de servidores, un servidor web, un servidor de red, un servidor de Internet, una estación de trabajo, un miniordenador, un ordenador de marco principal, un superordenador, una aplicación de red, una aplicación web, un sistema informático distribuido, sistemas multiprocesador, sistemas basados en procesador, electrónica de consumo, electrónica de consumo programable, televisión, televisión digital, decodificador de salón, punto de acceso inalámbrico, estación base, estación de abonado, centro de abonado móvil, controlador de red de radio, encaminador, concentrador, pasarela, puente, conmutador, máquina o combinaciones de los mismos.

Las realizaciones pueden implementarse como cualquiera o una combinación de: uno o más microchips o circuitos integrados interconectados usando una placa madre, lógica por cable, software almacenado por un dispositivo de memoria y ejecutado por un microprocesador, firmware, un circuito integrado específico de la aplicación (ASIC) y/o una matriz de puertas programable en campo (FPGA). El término "lógica" puede incluir, a modo de ejemplo, software o hardware y/o combinaciones de software y hardware.

Las realizaciones pueden proporcionarse, por ejemplo, como un producto de programa informático que puede incluir uno o más medios legibles por máquina que tienen almacenadas en los mismos instrucciones ejecutables en máquinas que, cuando se ejecutan por una o más máquinas, tales como un ordenador, red de ordenadores u otros dispositivos electrónicos, pueden resultar en la una o más máquinas que efectúan operaciones de acuerdo con realizaciones descritas en el presente documento. Un medio legible por máquina puede incluir, pero sin limitación, discos flexibles, discos ópticos, CD-ROM (Memorias de Solo Lectura de Disco Compacto) y discos magneto-ópticos, ROM, RAM, EPROM (Memorias de Solo Lectura Borrables y Programables), EEPROM (Memoria de Solo Lectura Eléctricamente Borrable y Programable), tarjetas ópticas o magnéticas, memoria flash u otro tipo de medios/medio legible por máquina adecuado para almacenar instrucciones ejecutables en máquinas.

Además, las realizaciones pueden descargarse como un producto de programa informático, en donde el programa puede transferirse desde un ordenador remoto (por ejemplo, un servidor) a un ordenador solicitante (por ejemplo, un cliente) por medio de una o más señales de datos incorporadas en y/o moduladas por una onda portadora u otro medio de propagación mediante un enlace de comunicación (por ejemplo, un módem y/o conexión de red).

Los procesadores de gráficos de extremo superior (por ejemplo, chip de gráficos de paquete discreto o en paquete) implementan habitualmente memoria de ancho de banda alto (HBM), que es una interfaz de RAM de alto rendimiento. HBM se incluye en el mismo paquete que una GPU y se conecta mediante un puente de silicio. El puente de silicio incluye una densidad alta de alambres que conectan las patillas de chip de GPU con las patillas de HBM. A menudo, las operaciones de gráficos se realizan en una ubicación de memoria de una manera de flujo continuo con una localidad de caché muy baja. Para tales operaciones, no es beneficioso llevar los datos de memoria a la caché de GPU, realizar la operación y, finalmente, expulsar los datos de vuelta a la memoria, ya que esto es un despilfarro en potencia y rendimiento (memoria ancho de banda alto y un despilfarro en entradas de caché). Un ejemplo común son las operaciones atómicas en sombreadores de cálculo que resultan del cálculo de histograma de imágenes.

De acuerdo con diversas realizaciones, el mecanismo de cálculo 610 presenta diversas operaciones que optimizan el cálculo en la GPU 614. En una realización, el mecanismo de cálculo 610 puede clasificar hilos en grupos de hilos basándose en una operación de 8 bits o 16 bits. En la actualidad, una GPU opera en un modo mixto en el que se procesan operaciones de 8 bits o 16 bits aleatorias en el hardware de la GPU 614. En una realización, el mecanismo de cálculo 610 incluye un clasificador para clasificar hilos en grupos basándose en profundidad de bit (8 bits o 16 bits).

La Figura 7A ilustra una realización del mecanismo de cálculo 610 que incluye un clasificador 710, contenedores de coma flotante 712 y 714 y unidades de coma flotante 716. En una realización, el clasificador 710 recibe hilos que tienen operaciones de coma flotante tanto de 8 bits como de 16 bits, y clasifica estas operaciones en respectivos contenedores 712 y 714. Una vez en los contenedores, las operaciones se reenvían a las unidades de coma flotante 716. Por consiguiente, las operaciones de FP8 recibidas desde el contenedor 712 se procesan como una única unidad de FP8 716, mientras las operaciones de FP16 recibidas desde el contenedor 714 se procesan en dos unidades de FP8 716.

En una realización adicional, el mecanismo de cálculo 610 también puede recibir hilos que tienen operaciones de 32 bits. En esta realización, el mecanismo de cálculo 610 incluye lógica de coma flotante para procesar operaciones de hilo de coma flotante que tienen una profundidad de bit más alta. Por ejemplo, la lógica de coma flotante procesa operaciones de 32 bits usando lógica de 16 bits, en lugar de lógica de 32 bits. **La Figura 7B** ilustra una realización del mecanismo de cálculo 610 que tiene lógica de coma flotante 730 para realizar tales operaciones.

Como se muestra en la **Figura 7B**, la lógica 730 incluye un componente de procesamiento de FP16 y un componente de procesamiento de delta. Siempre que se recibe una operación de 32 bits, se implementa FP16 para procesar los 16 bits inferiores, mientras el componente de delta se usa para procesar los 16 bits superiores. En una realización adicional, las operaciones de 16 bits también se procesan en la lógica 730. Sin embargo, en esta realización, las operaciones de 16 bits se procesan por únicamente las operaciones de FP16 recibidas en la lógica 730. En una realización adicional más, el mecanismo de cálculo 610 puede apagar la potencia al componente de delta durante el procesamiento de operaciones de 16 bits, ahorrando por lo tanto potencia. En otras realizaciones, el mecanismo de cálculo 610 puede procesar operaciones que tienen profundidades de bit más altas (por ejemplo, 64 bits).

En sistemas convencionales, se proporciona soporte para operandos que tienen la misma precisión. Si algunos operandos están en un formato diferente, se necesitan instrucciones separadas para convertir primero los operandos a un formato común. El mecanismo de cálculo 610 también proporciona soporte de precisión variable en una instrucción de cálculo. En una realización de este tipo, el mecanismo de cálculo 610 soporta operaciones de multiplicar-acumular fusionadas de precisión mixta (FMAC), de tal forma que en una operación de ALU de $D = A*B+C$; A y B y/o C incluyen precisión y formato diferentes. Por lo tanto, A, B, C incluyen INT32 e INT8.

De acuerdo con una realización, una instrucción de ALU es con la condición de que incluye un atributo de 16 bits que especifica el formato para cada operando y un formato requerido para el resultado (destino). En respuesta a la instrucción, una ALU de GPU (por ejemplo, en un núcleo de sombreador) ejecuta la instrucción evaluando el atributo para determinar el formato para cada operando. Posteriormente, cada operando se convierte al formato de destino y se ejecuta la operación de FMAC.

Visión general de aprendizaje automático

Un algoritmo de aprendizaje automático es un algoritmo que puede aprender basándose en un conjunto de datos. Las realizaciones de algoritmos de aprendizaje automático pueden diseñarse para modelar abstracciones de nivel alto dentro de un conjunto de datos. Por ejemplo, pueden usarse algoritmos de reconocimiento de imágenes para determinar cuál de varias categorías a la que pertenece una entrada dada; los algoritmos de regresión pueden emitir un valor numérico dada una entrada; y pueden usarse algoritmos de reconocimiento de patrones para generar texto traducido o realizar reconocimiento de texto a voz y/o voz.

Un tipo ilustrativo de algoritmo de aprendizaje automático es una red neuronal. Existen muchos tipos de redes neuronales; un tipo simple de red neuronal es una red predictiva. Una red predictiva puede implementarse como un gráfico acíclico en el que los nodos se disponen en capas. Habitualmente, una topología de red predictiva incluye una capa de entrada y una capa de salida que están separadas por al menos una capa oculta. La capa oculta transforma

una entrada recibida por la capa de entrada en una representación que es útil para generar una salida en la capa de salida. Los nodos de red se conectan totalmente mediante bordes a los nodos en capas adyacentes, pero no existen bordes entre nodos dentro de cada capa. Los datos recibidos en los nodos de una capa de entrada de una red predictiva se propagan (es decir, "se alimentan hacia delante") a los nodos de la capa de salida mediante una función de activación que calcula los estados de los nodos de cada capa sucesiva en la red basándose en coeficientes ("ponderaciones") asociados respectivamente con cada uno de los bordes que conectan las capas. Dependiendo del modelo específico que se representa por el algoritmo que se ejecuta, la salida desde el algoritmo de red neuronal puede tomar diversas formas.

Antes de que pueda usarse un algoritmo de aprendizaje automático para modelar un problema particular, se entrena el algoritmo usando un conjunto de datos de entrenamiento. Entrenar una red neuronal implica seleccionar una topología de red, usando un conjunto de datos de entrenamiento que representan un problema que se modela por la red, y ajustando las ponderaciones hasta que el modelo de red funciona con un error mínimo para todas las instancias del conjunto de datos de entrenamiento. Por ejemplo, durante un proceso de entrenamiento de aprendizaje supervisado para una red neuronal, la salida producida por la red en respuesta a la entrada que representa una instancia en un conjunto de datos de entrenamiento se compara con la salida etiquetada "correcta" para esa instancia, se calcula una señal de error que representa la diferencia entre la salida y la salida etiquetada, y las ponderaciones asociadas con las conexiones se ajustan para minimizar ese error a medida que la señal de error se propaga hacia atrás a través de las capas de la red. La red se considera "entrenada" cuando se minimizan los errores para cada una de las salidas generadas desde las instancias del conjunto de datos de entrenamiento.

La precisión de un algoritmo de aprendizaje automático puede verse afectada significativamente por la calidad del conjunto de datos usados para entrenar el algoritmo. El proceso de entrenamiento puede requerir cálculos intensivos y puede requerir una cantidad significativa de tiempo en un procesador de fin general convencional. Por consiguiente, se usa un hardware de procesamiento paralelo para entrenar muchos tipos de algoritmos de aprendizaje automático. Esto es particularmente útil para optimizar el entrenamiento de redes neurales, ya que los cálculos realizados en el ajuste de los coeficientes en redes neuronales se prestan naturalmente a implementaciones paralelas. Específicamente, se han adaptado muchos algoritmos de aprendizaje automático y aplicaciones de software para hacer uso del hardware de procesamiento paralelo dentro de dispositivos de procesamiento de gráficos de fin general.

La Figura 8 es un diagrama generalizado de una pila de software de aprendizaje automático 800. Una aplicación de aprendizaje automático 802 puede configurarse para entrenar una red neuronal usando un conjunto de datos de entrenamiento o para usar una red neuronal profunda entrenada para implementar inteligencia artificial. La aplicación de aprendizaje automático 802 puede incluir una funcionalidad de entrenamiento e inferencia para una red neuronal y/o software especializado que puede usarse para entrenar una red neuronal antes de su despliegue. La aplicación de aprendizaje automático 802 puede implementar cualquier tipo de inteligencia artificial que incluye, pero sin limitación a, reconocimiento de imágenes, mapeo y localización, navegación autónoma, síntesis de voz, formación de imágenes médicas o traducción del lenguaje.

Puede posibilitarse una aceleración de hardware para la aplicación de aprendizaje automático 802 mediante un marco de aprendizaje automático 804. El marco de aprendizaje automático 804 puede proporcionar una librería de primitivas de aprendizaje automático. Las primitivas de aprendizaje automático son operaciones básicas que se realizan comúnmente por algoritmos de aprendizaje automático. Sin el marco de aprendizaje automático 804, se requeriría que los desarrolladores de algoritmos de aprendizaje automático creasen y optimizasen la lógica de cálculo principal asociada con el algoritmo de aprendizaje automático, a continuación, optimizasen de nuevo la lógica de cálculo a medida que se desarrollasen nuevos procesadores paralelos. En su lugar, la aplicación de aprendizaje automático puede configurarse para realizar los cálculos necesarios usando las primitivas proporcionadas por el marco de aprendizaje automático 804. Las primitivas ilustrativas incluyen convoluciones de tensor, funciones de activación y agrupamiento, que son operaciones de cálculo que se realizan mientras se entrena una red neuronal convolucional (CNN). El marco de aprendizaje automático 804 también puede proporcionar primitivas para implementar subprogramas de álgebra lineal básicos por muchos algoritmos de aprendizaje automático, tales como operaciones de matriz y vectoriales.

El marco de aprendizaje automático 804 puede procesar datos de entrada recibidos desde la aplicación de aprendizaje automático 802 y generar la entrada apropiada a un marco de cálculo 806. El marco de cálculo 806 puede extraer las instrucciones subyacentes proporcionadas al controlador de GPGPU 808 para posibilitar que el marco de aprendizaje automático 804 se aproveche de aceleración de hardware a través del hardware de GPGPU 810 sin requerir que el marco de aprendizaje automático 804 tenga un conocimiento profundo de la arquitectura del hardware de GPGPU 810. Adicionalmente, el marco de cálculo 806 puede posibilitar una aceleración de hardware para el marco de aprendizaje automático 804 a través de una diversidad de tipos y generaciones del hardware de GPGPU 810.

Aceleración de aprendizaje automático de GPGPU

La Figura 9 ilustra una unidad de procesamiento de gráficos de fin general altamente paralelo 900, de acuerdo con una realización. En una realización, la unidad de procesamiento de fin general (GPGPU) 900 puede configurarse para ser particularmente eficiente en el procesamiento del tipo de cargas de trabajo de cálculo asociadas con el

entrenamiento de redes neuronales profundas. Adicionalmente, la GPGPU 900 puede enlazarse directamente a otras instancias de la GPGPU para crear una agrupación de múltiples GPU para mejorar la velocidad de entrenamiento para redes neuronales particularmente profundas.

La GPGPU 900 incluye una interfaz de anfitrión 902 para posibilitar una conexión con un procesador de anfitrión. En una realización, la interfaz de anfitrión 902 es una interfaz de PCI Express. Sin embargo, la interfaz de anfitrión también puede ser una interfaz de comunicaciones específica de proveedor o tejido de comunicaciones. La GPGPU 900 recibe comandos desde el procesador de anfitrión y usa un planificador global 904 para distribuir hilos de ejecución asociados con esos comandos a un conjunto de agrupaciones de cálculo 906A-H. Las agrupaciones de cálculo 906A-H comparten una memoria caché 908. La memoria caché 908 puede servir como una caché de nivel más alto para memorias caché dentro de las agrupaciones de cálculo 906A-H.

La GPGPU 900 incluye la memoria 914A-B acoplada con las agrupaciones de cálculo 906A-H mediante un conjunto de controladores de memoria 912A-B. En diversas realizaciones, la memoria 914A-B puede incluir diversos tipos de dispositivos de memoria que incluyen memoria de acceso aleatorio dinámica (DRAM) o memoria de acceso aleatorio de gráficos, tal como memoria de acceso aleatorio de gráficos síncrona (SGRAM), incluyendo memoria de doble tasa de datos de gráficos (GDDR). En una realización, las unidades de memoria 224A-N pueden incluir también memoria en 3D apilada, que incluye, pero sin limitación, memoria de alto ancho de banda (HBM).

En una realización, cada agrupación de cálculo GPLAB06A-H incluye un conjunto de multiprocesadores de gráficos, tales como el multiprocesador de gráficos 400 de la Figura 4A. Los multiprocesadores de gráficos de la agrupación de cálculo múltiples tipos de unidades de lógica de enteros y coma flotante que pueden realizar operaciones de cálculo en un intervalo de precisiones que incluyen adecuadas para cálculos de aprendizaje automático. Por ejemplo, y en una realización, al menos un subconjunto de las unidades de coma flotante en cada una de las agrupaciones de cálculo 906A-H puede configurarse para realizar operaciones de coma flotante de 16 bits o 32 bits, mientras puede configurarse un subconjunto diferente de la unidades de coma flotante para realizar operaciones de coma flotante de 64 bits.

Pueden configurarse múltiples instancias de la GPGPU 900 para operar como una agrupación de cálculo. El mecanismo de comunicación usado por la agrupación de cálculo para sincronización e intercambio de datos varía a través de realizaciones. En una realización, las múltiples instancias de la GPGPU 900 se comunican a través de la interfaz de anfitrión 902. En una realización, la GPGPU 900 incluye un concentrador de E/S 908 que acopla la GPGPU 900 con un enlace de GPU 910 que posibilita una conexión directa a otras instancias de la GPGPU. En una realización, el enlace de GPU 910 se acopla a un puente de GPU a GPU especializado que posibilita una comunicación y sincronización entre múltiples instancias de la GPGPU 900. En una realización, el enlace de GPU 910 se acopla con una interconexión de alta velocidad para transmitir y recibir datos a otros GPGPU o procesadores paralelos. En una realización, las múltiples instancias de la GPGPU 900 se ubican en sistemas de procesamiento de datos separados y se comunican mediante un dispositivo de red que es accesible mediante la interfaz de anfitrión 902. En una realización, el enlace de GPU 910 puede configurarse para posibilitar una conexión a un procesador de anfitrión además de o como una alternativa a la interfaz de anfitrión 902.

Mientras la configuración ilustrada de la GPGPU 900 puede configurarse para entrenar redes neurales, una realización proporciona una configuración alternativa de la GPGPU 900 que puede configurarse para su despliegue dentro de una plataforma de inferencia de alto rendimiento o baja potencia. En una configuración de inferencia la GPGPU 900 incluye menos de las agrupaciones de cálculo 906A-H en relación con la configuración de entrenamiento. Adicionalmente, la tecnología de memoria asociada con la memoria 914A-B puede diferir entre configuraciones de inferencia y de entrenamiento. En una realización, la configuración de inferencia de la GPGPU 900 puede soportar instrucciones específicas de inferencia. Por ejemplo, una configuración de inferencia puede proporcionar soporte para una o más instrucciones de producto vectorial de entero de 8 bits, que se usan comúnmente durante operaciones de inferencia para redes neuronales desplegadas.

La Figura 10 ilustra un sistema informático de múltiples GPU 1000, de acuerdo con una realización. El sistema informático de múltiples GPU 1000 puede incluir un procesador 1002 acoplado a múltiples GPGPU 1006A-D mediante un conmutador de interfaz de anfitrión 1004. El conmutador de interfaz de anfitrión 1004, en una realización, es un dispositivo de conmutador de PCI Express que acopla el procesador 1002 a un bus de PCI Express a través del cual el procesador 1002 puede comunicarse con el conjunto de las GPGPU 1006A-D. Cada una de las múltiples GPGPU 1006A-D puede ser una instancia de la GPGPU 900 de la Figura 9. Las GPGPU 1006A-D pueden interconectarse mediante un conjunto de enlaces de GPU a GPU de punto a punto de alta velocidad 1016. Los enlaces de GPU a GPU de alta velocidad pueden conectarse a cada una de las GPGPU 1006A-D mediante un enlace de GPU especializado, tal como el enlace de GPU 910 como en la Figura 9. Los enlaces de GPU de P2P 1016 posibilitan comunicación directa entre cada una de las GPGPU 1006A-D sin requerir comunicación a través del bus de interfaz de anfitrión al que se conecta el procesador 1002. Con tráfico de GPU a GPU dirigido a los enlaces de GPU de P2P, el bus de interfaz de anfitrión permanece disponible para acceso a memoria de sistema o para comunicarse con otras instancias del sistema informático de múltiples GPU 1000, por ejemplo, mediante uno o más dispositivos de red. Mientras en la realización ilustrada las GPGPU 1006A-D se conectan al procesador 1002 a través del conmutador de

interfaz de anfitrión 1004, en una realización el procesador 1002 incluye soporte directo para los enlaces de GPU de P2P 1016 y puede conectarse directamente a las GPGPU 1006A-D.

Implementaciones de red neuronal de aprendizaje automático

La arquitectura informática proporcionada por las realizaciones descritas en el presente documento puede configurarse para realizar los tipos de procesamiento paralelo que es particularmente adecuado para el entrenamiento y despliegue de redes neuronales para aprendizaje automático. Una red neuronal puede generalizarse como una red de funciones que tienen una relación gráfica. Como se conoce bien en la técnica, existen una diversidad de tipos de implementaciones de red neuronal usadas en el aprendizaje automático. Un tipo ilustrativo de red neuronal es la red predictiva, como se ha descrito anteriormente.

Un segundo tipo ilustrativo de red neuronal es la Red Neuronal Convolucional (CNN). Una CNN es una red neuronal predictiva especializada para procesar datos que tiene una topología de tipo cuadrícula conocida, tal como datos de imagen. Por consiguiente, las CNN se usan comúnmente para aplicaciones de visión por ordenador y reconocimiento de imágenes, pero también pueden usarse para otros tipos de reconocimiento de patrones, tales como procesamiento de voz y del lenguaje. Los nodos en la capa de entrada de CNN se organizan en un conjunto de "filtros" (detectores de características inspirados por los campos receptivos encontrados en la retina), y la salida de cada conjunto de filtros se propaga a nodos en capas sucesivas de la red. Los cálculos para una CNN incluyen aplicar la operación matemática de convolución a cada filtro para producir la salida de ese filtro. La convolución es una clase especializada de operación matemática realizada por dos funciones para producir una tercera función que es una versión modificada de una de las dos funciones originales. En terminología de red convolucional, la primera función a la convolución puede denominarse como la entrada, mientras que la segunda función puede denominarse como el núcleo de convolución. La salida puede denominarse como el mapa de características. Por ejemplo, la entrada a una capa de convolución puede ser una matriz multidimensional de datos que define los diversos componentes de color de una imagen de entrada. El núcleo de convolución puede ser una matriz multidimensional de parámetros, en la que los parámetros se adaptan por el proceso de entrenamiento para la red neuronal.

Las redes neuronales recurrentes (RNN) son una familia de redes neuronales predictivas que incluyen conexiones de realimentación entre capas. Las RNN posibilitan el modelado de datos secuenciales compartiendo datos de parámetros a través de diferentes partes de la red neuronal. La arquitectura para una RNN incluye ciclos. Los ciclos representan la influencia de un valor presente de una variable en su propio valor en un momento futuro, ya que al menos una porción de los datos de salida desde la RNN se usa como realimentación para procesar una entrada posterior en una secuencia. Esta característica hace las RNN particularmente útiles para el procesamiento del lenguaje debido a la naturaleza variable en la que pueden componerse los datos del lenguaje.

Las figuras descritas a continuación presentan redes predictivas, CNN y RNN, así como describen un proceso general para entrenar y desplegar respectivamente cada uno de esos tipos de redes. Se entenderá que estas descripciones son ilustrativas y no limitantes en cuanto a cualquier realización específica descrita en el presente documento y los conceptos ilustrados pueden aplicarse generalmente a redes neuronales profundas y técnicas de aprendizaje automático en general.

Las redes neuronales ilustrativas descritas anteriormente pueden usarse para realizar aprendizaje profundo. El aprendizaje profundo es aprendizaje automático que usa redes neuronales profundas. La redes neuronales profundas usadas en aprendizaje profundo son redes neuronales artificiales compuestas de múltiples capas ocultas, al contrario que redes neuronales poco profundas que incluyen únicamente una única capa oculta. Las redes neuronales más profundas requieren generalmente más cálculos intensivos para entrenar. Sin embargo, las capas ocultas adicionales de la red posibilitan un reconocimiento de patrones de múltiples etapas que resulta en error de salida reducido en relación con técnicas de aprendizaje automático poco profundas.

Las redes neuronales profundas usadas en aprendizaje profundo habitualmente incluyen una red de extremo frontal para realizar reconocimiento de características acoplado a red de extremo final que representa un modelo matemático que puede realizar operaciones (por ejemplo, clasificación de objetos, reconocimiento de voz, etc.) basándose en la representación de características proporcionada al modelo. El aprendizaje profundo posibilita que se realice el aprendizaje automático sin requerir que se realice ingeniería de características hecha a mano para el modelo. En su lugar, las redes neuronales profundas pueden aprender características basándose en una estructura o correlación estadística dentro de los datos de entrada. Las características aprendidas pueden proporcionarse a un modelo matemático que puede mapear características detectadas a una salida. El modelo matemático usado por la red se especializa generalmente para la tarea específica que hay que realizar, y se usarán diferentes modelos para realizar una tarea diferente.

Una vez que se estructura la red neuronal, puede aplicarse un modelo de aprendizaje a la red para entrenar la red para realizar tareas específicas. El modelo de aprendizaje describe cómo ajustar las ponderaciones dentro del modelo para reducir el error de salida de la red. La propagación hacia atrás de errores es un método común usado para entrenar redes neuronales. Se presenta un vector de entrada a la red para su procesamiento. La salida de la red se compara con la salida deseada usando una función de pérdida y se calcula un valor de error para cada una de las

neuronas en la capa de salida. Los valores de error se propagan, a continuación, hacia atrás hasta que cada neurona tiene un valor de error asociado que representa aproximadamente su contribución a la salida original. La red puede aprender, a continuación, de esos errores usando un algoritmo, tal como el algoritmo descendente de gradiente estocástico, para actualizar las ponderaciones de la red neuronal.

Las Figuras 11A y 11B ilustran una red neuronal convolucional ilustrativa. **La Figura 11A** ilustra diversas capas dentro de una CNN. Como se muestra en la **Figura 11A**, una CNN ilustrativa usada para modelar procesamiento de imágenes puede recibir la entrada 1102 que describe los componentes rojo, verde y azul (RGB) de una imagen de entrada. La entrada 1102 puede procesarse por múltiples capas convolucionales (por ejemplo, la capa convolucional 1104, la capa convolucional 1106). La salida de las múltiples capas convolucionales puede procesarse opcionalmente por un conjunto de capas totalmente completadas 1108. Las neuronas de una capa totalmente conectada tienen conexiones completas a todas las activaciones en la capa anterior, como se ha descrito anteriormente para una red predictiva. La salida de las capas totalmente completadas 1108 puede usarse para generar un resultado de salida de la red. Las activaciones dentro de las capas totalmente completadas 908 pueden calcularse usando multiplicación de matriz en lugar de convolución. No todas las implementaciones de CNN hacen uso de las capas totalmente conectadas 1108. Por ejemplo, en algunas implementaciones la capa convolucional 1106 puede generar una salida para la CNN.

Las capas convolucionales se conectan de forma dispersa, que difiere de una configuración de red neuronal tradicional encontrada en las capas totalmente completadas 1108. Las capas de red neuronal tradicionales se conectan totalmente, de tal forma que cada unidad de salida interactúa con cada unidad de entrada. Sin embargo, las capas convolucionales se conectan de forma dispersa porque la salida de la convolución de un campo se introduce (en lugar del respectivo valor de estado de cada uno de los nodos en el campo) a los nodos de la capa posterior, como se ilustra. Los núcleos asociados con las capas convolucionales realizan operaciones de convolución, cuya salida se envía a la siguiente capa. La reducción de dimensionalidad realizada dentro de las capas convolucionales es un aspecto que posibilita que la CNN escale para procesar imágenes grandes.

La Figura 11B ilustra etapas de cálculo ilustrativas dentro de una capa convolucional de una CNN. La entrada en una capa convolucional 1112 de una CNN puede procesarse en tres etapas de una capa convolucional 1114. Las tres etapas pueden incluir una etapa de convolución 1116, una etapa de detección 1118 y una etapa de agrupamiento 1120. La capa de convolución 1114 puede emitir, a continuación, datos a una capa convolucional sucesiva. La capa convolucional final de la red puede generar datos de mapa de características de salida o proporcionar una entrada a una capa totalmente conectada, por ejemplo, para generar un valor de clasificación para la entrada a la CNN.

En la etapa de convolución 1116 realiza varias convoluciones en paralelo para producir un conjunto de activaciones lineales. La etapa de convolución 1116 puede incluir una transformación afín, que es cualquier transformación que puede especificarse como una transformación lineal más una traslación. Las transformaciones afines incluyen rotaciones, traslaciones, escalado y combinaciones de estas transformaciones. La etapa de convolución calcula la salida de funciones (por ejemplo, neuronas) que se conectan a regiones específicas en la salida, que puede determinarse como la región local asociada con la neurona. Las neuronas calculan un producto vectorial entre las ponderaciones de las neuronas y la región en la entrada local a la que se conectan las neuronas. La salida desde la etapa de convolución 1116 define un conjunto de activaciones lineales que se procesan por etapas sucesivas de la capa convolucional 1114.

Las activaciones lineales pueden procesarse mediante una etapa de detección 1118. En la etapa de detección 1118, cada activación lineal se procesa por una función de activación no lineal. La función de activación no lineal aumenta las propiedades no lineales de la red general sin afectar a los campos receptivos de la capa de convolución. Pueden usarse varios tipos de funciones de activación no lineales. Un tipo particular es la unidad lineal rectificada (ReLU), que usa una función de activación definida como $f(x) = \max(0, x)$, de tal forma que la activación tiene un umbral de cero.

La etapa de agrupamiento 1120 usa una función de agrupamiento que sustituye la salida de la capa convolucional 1106 con una estadística de resumen de salidas cercanas. La función de agrupamiento puede usarse para introducir una invarianza de traslación en la red neuronal, de tal forma que pequeñas traslaciones a la entrada no cambian los resultados agrupados. La invarianza a la traslación local puede ser útil en escenarios en los que la presencia de una característica en los datos de entrada es más importante que la ubicación precisa de la característica. Pueden usarse diversos tipos de funciones de agrupamiento durante la etapa de agrupamiento 1120, incluyendo agrupamiento máximo, agrupamiento promedio y agrupamiento de norma l2. Adicionalmente, algunas implementaciones de CNN no incluyen una etapa de agrupamiento. En su lugar, tales implementaciones sustituyen y etapa de convolución adicional que tiene un intervalo aumentado en relación con etapas convolucionales anteriores.

La salida desde la capa convolucional 1114 puede procesarse, a continuación, por la capa siguiente 1122. La capa siguiente 1122 puede ser una capa convolucional adicional o una de las capas totalmente completadas 1108. Por ejemplo, la primera capa convolucional 1104 de la **Figura 11A** puede emitir a la segunda capa convolucional 1106, mientras la segunda capa convolucional puede emitir a una primera capa de las capas totalmente completadas 1108.

La Figura 12 ilustra una red neuronal recurrente 1200 ilustrativa. En una red neuronal recurrente (RNN), el estado anterior de la red influencia la salida del estado actual de la red. Las RNN pueden crearse de una diversidad de formas

usando una diversidad de funciones. El uso de las RNN generalmente gira en torno del uso de modelos matemáticos para predecir el futuro basándose en una secuencia anterior de entradas. Por ejemplo, puede usarse una RNN para realizar modelado de lenguaje estadístico para predecir una próxima palabra dada una secuencia anterior de palabras. La RNN 1200 ilustrada puede describirse como que tiene una capa de entrada 1202 que recibe un vector de entrada, capas ocultas 1204 para implementar una función recurrente, un mecanismo de realimentación 1205 para posibilitar una 'memoria' de estados anteriores y una capa de salida 1206 para emitir un resultado. La RNN 1200 opera basándose en pasos de tiempo. El estado de la RNN en un paso de tiempo dado se ve influenciado basándose en el paso de tiempo anterior a través del mecanismo de realimentación 1205. Para un paso de tiempo dado, el estado de las capas ocultas 1204 se define por el estado anterior y la entrada en el paso de tiempo actual. La capa oculta 1204 puede procesar una entrada inicial (x_1) en un primer paso de tiempo. La capa oculta 1204 puede procesar una segunda entrada (x_2) usando información de estado que se determina durante el procesamiento de la entrada inicial (x_1). Un estado dado puede calcularse como $s_t = f(Ux_t + Ws_{t-1})$, donde U y W son matrices de parámetros. La función f es generalmente una no linealidad, tal como la función tangente hiperbólica (Tanh) o una variante de la función de rectificador $f(x) = \text{máx}(0, x)$. Sin embargo, la función matemática específica usada en las capas ocultas 1004 puede variar dependiendo de los detalles de implementación específicos de la RNN 1200.

Además de las redes de CNN y RNN básicas descritas, pueden posibilitarse variaciones en esas redes. Una variante de RNN de ejemplo es la RNN de memoria a largo/corto plazo (LSTM). Las RNN de LSTM son capaces de aprender dependencias a largo plazo que pueden ser necesarias para procesar secuencias de lenguaje más largas. Una variante en la CNN es una red de creencia profunda convolucional, que tiene una estructura similar a una CNN y se entrena de una manera similar una red de creencia profunda. Una red de creencia profunda (DBN) es una red neuronal generativa que se compone de múltiples capas de variables (aleatorias) estocásticas. Las DBN pueden entrenarse capa a capa usando aprendizaje sin supervisión voraz. Las ponderaciones aprendidas de la DBN pueden usarse, a continuación, para proporcionar redes neuronales preentrenadas determinando un conjunto inicial óptimo de ponderaciones para la red neuronal.

La Figura 13 ilustra el entrenamiento y despliegue de una red neuronal profunda. Una vez que se ha estructurado una red dada para una tarea, la red neuronal se entrena usando un conjunto de datos de entrenamiento 1302. Se han desarrollado diversos marcos de entrenamiento 1304 para posibilitar una aceleración de hardware del proceso de entrenamiento. Por ejemplo, el marco de aprendizaje automático 804 de la Figura 8 puede configurarse como un marco de entrenamiento 1304. El marco de entrenamiento 1304 puede conectarse a una red neuronal no entrenada 1306 y posibilitar que la red neuronal no entrenada se entrene usando los recursos de procesamiento paralelo descritos en el presente documento para generar una red neuronal entrenada 1308.

Para iniciar el proceso de entrenamiento, las ponderaciones iniciales pueden elegirse aleatoriamente o mediante preentrenamiento usando una red de creencia profunda. El ciclo de entrenamiento puede realizarse, a continuación, de una manera o bien supervisada o bien no supervisada.

El aprendizaje supervisado en un método de aprendizaje en el que se realiza el entrenamiento como una operación mediada, tal como cuando el conjunto de datos de entrenamiento 1302 incluye una entrada emparejada con la salida deseada para la salida, o en el que el conjunto de datos de entrenamiento incluye una entrada que tiene una salida conocida y la salida de la red neuronal se califica manualmente. La red procesa las entradas y compara las salidas resultantes contra un conjunto de salidas esperadas o deseadas. Los errores se propagan, a continuación, de vuelta a través del sistema. El marco de entrenamiento 1304 puede ajustarse para ajustar las ponderaciones que controlan la red neuronal no entrenada 1306. El marco de entrenamiento 1304 puede proporcionar herramientas para monitorizar cómo de bien está convergiendo la red neuronal no entrenada 1306 hacia un modelo adecuado para generar respuestas correctas basándose en datos de entrada conocidos. El proceso de entrenamiento se produce repetidamente a medida que se ajustan las ponderaciones de la red para refinar la salida generada por la red neuronal. El proceso de entrenamiento puede continuar hasta que la red neuronal alcanza una precisión estadísticamente deseada asociada con una red neuronal entrenada 1308. La red neuronal entrenada 1308 puede desplegarse, a continuación, para implementar cualquier número de operaciones de aprendizaje automático.

El entrenamiento no supervisado es un método de aprendizaje en el que la red intenta entrenarse a sí misma usando datos no etiquetados. Por lo tanto, para un entrenamiento no supervisado, el conjunto de datos de entrenamiento 1302 incluirá datos de entrada sin ningún dato de salida asociado. La red neuronal no entrenada 1306 puede aprender agrupaciones dentro de la entrada no etiquetada y puede determinar cómo las entradas individuales se relacionan con el conjunto de datos general. El entrenamiento no supervisado puede usarse para generar un mapa de auto organización, que es un tipo de red neuronal entrenada 1307 con capacidad de realizar operaciones útiles en la reducción de la dimensionalidad de datos. El entrenamiento no supervisado también puede usarse para realizar detección de anomalías, que permite la identificación de puntos de datos en un conjunto de datos de entrada que se desvían de los patrones normales de los datos.

También pueden emplearse variaciones en entrenamiento supervisado y no supervisado. El aprendizaje semi supervisado es una técnica en la que en el conjunto de datos de entrenamiento 1302 incluye una mezcla de datos etiquetados o no etiquetados de la misma distribución. El aprendizaje incremental es una variante del aprendizaje supervisado en el que los datos de entrada se usan continuamente para entrenar adicionalmente el modelo. El

aprendizaje incremental posibilita que la red neuronal entrenada 1308 se adapte a los nuevos datos 1312 sin olvidar el conocimiento inculcado dentro de la red durante el entrenamiento inicial.

Ya sea supervisado o no supervisado, el proceso de entrenamiento para redes neuronales particularmente profundas puede requerir cálculos demasiado intensivos para un único nodo de cálculo. En lugar de usar un único nodo de cálculo, puede usarse una red distribuida de nodos de cálculo para acelerar el proceso de entrenamiento.

La Figura 14 es un diagrama de bloques que ilustra el aprendizaje distribuido. El aprendizaje distribuido es un modelo de entrenamiento que usa múltiples nodos informáticos distribuidos para realizar entrenamiento supervisado o no supervisado de una red neuronal. Cada uno de los nodos de cálculo distribuidos puede incluir uno o más procesadores de anfitrión y uno o más de los nodos de procesamiento de fin general, tales como la unidad de procesamiento de gráficos de fin general altamente paralelo 900 como en la Figura 9. Como se ilustra, el aprendizaje distribuido puede realizarse paralelismo de modelo 1402, paralelismo de datos 1404 o una combinación de paralelismo de modelo y de datos 1204.

En el paralelismo de modelo 1402, diferentes nodos de cálculo en un sistema distribuido pueden realizar cálculos de entrenamiento para diferentes partes de una única red. Por ejemplo, cada capa de una red neuronal puede entrenarse por un nodo de procesamiento diferente del sistema distribuido. Los beneficios del paralelismo de modelo incluyen la capacidad de escalar a modelos particularmente grandes. Dividir los cálculos asociados con diferentes capas de la red neuronal posibilita el entrenamiento de redes neuronales grandes en las que las ponderaciones de todas las capas no cabrían en la memoria de un único nodo de cálculo. En algunos casos, el paralelismo de modelo puede ser particularmente útil en la realización de entrenamiento no supervisado de redes neuronales grandes.

En el paralelismo de datos 1404, los diferentes nodos de la red distribuida tienen una instancia completa del modelo y cada nodo recibe una porción diferente de los datos. Los resultados de los diferentes nodos se combinan a continuación. Mientras son posibles diferentes enfoques para el paralelismo de datos, todos los enfoques de entrenamiento paralelo de datos requieren una técnica de combinación de resultados y sincronización de los parámetros de modelo en cada nodo. Los enfoques ilustrativos para combinar datos incluyen promedio de parámetros y paralelismo de datos basado en actualización. El promedio de parámetros entrena cada nodo en un subconjunto de los datos de entrenamiento y establece los parámetros globales (por ejemplo, ponderaciones, desviaciones) al promedio de los parámetros de cada nodo. El promedio de parámetros usa un servidor de parámetros central que mantiene los datos de parámetro. El paralelismo de datos basado en actualización es similar al promedio de parámetros excepto que en lugar de transferir parámetros desde los nodos al servidor de parámetros, se transfieren las actualizaciones al modelo.
Adicionalmente, el paralelismo de datos basado en actualización puede realizarse de una manera descentralizada, en la que las actualizaciones se comprimen y transfieren entre nodos.

El paralelismo de modelo y de datos combinado 1406 puede implementarse, por ejemplo, en un sistema distribuido en el que cada nodo de cálculo incluye múltiples GPU. Cada nodo puede tener una instancia completa del modelo con GPU separadas dentro de cada nodo se usan para entrenar diferentes porciones del modelo.

El entrenamiento distribuido ha aumentado la sobrecarga en relación con el entrenamiento de una única máquina. Sin embargo, cada uno de los procesadores paralelos y de las GPGPU descritos en el presente documento puede implementar diversas técnicas para reducir la sobrecarga de entrenamiento distribuido, incluyendo técnicas para posibilitar transferencia de datos de GPU a GPU de ancho de banda alto y sincronización de datos remota acelerada.

Aplicaciones de aprendizaje automático ilustrativas

El aprendizaje automático puede aplicarse para resolver una diversidad de problemas tecnológicos, incluyendo pero sin limitación a, visión por ordenador, conducción y navegación autónoma, reconocimiento de voz y procesamiento del lenguaje. La visión por ordenador ha sido tradicionalmente una de las áreas de investigación más activas para aplicaciones de aprendizaje automático. Aplicaciones de visión por ordenador varían desde la reproducción de capacidades visuales humanas, tales como el reconocimiento facial, hasta la creación de nuevas categorías de capacidades visuales. Por ejemplo, las aplicaciones de visión por ordenador pueden configurarse para reconocer ondas de sonido a partir de vibraciones inducidas en objetos visibles en un vídeo. El aprendizaje automático acelerado de procesador paralelo posibilita que las aplicaciones de visión por ordenador se entrenen usando un conjunto de datos de entrenamiento significativamente mayor que el anteriormente factible y posibilita que se desplieguen sistemas de inferencia usando procesadores paralelos de baja potencia.

El aprendizaje automático acelerado de procesador paralelo tiene aplicaciones de conducción autónoma que incluyen reconocimiento de línea y señales de tráfico, evitación de obstáculos, navegación y control de la conducción. Las técnicas de aprendizaje automático acelerado pueden usarse para entrenar modelos de conducción basándose en conjuntos de datos que definen las respuestas apropiadas a entrada de entrenamiento específica. Los procesadores paralelos descritos en el presente documento pueden posibilitar un entrenamiento rápido de las redes neuronales crecientemente complejas usadas para soluciones de aprendizaje autónomo y posibilita el despliegue de procesadores de inferencia de baja potencia en una plataforma móvil adecuada para su integración en vehículos autónomos.

Las redes neuronales profundas aceleradas de procesador paralelo han posibilitado los enfoques de aprendizaje automático para reconocimiento automático de la voz (ASR). ASR incluye la creación de una función que calcula la secuencia lingüística más probable dada una secuencia acústica de entrada. El aprendizaje automático acelerado que usa redes neuronales profundas han posibilitado la sustitución de los modelos ocultos de Markov (HMM) y modelos de mezcla gaussiana (GMM) usados anteriormente para ASR.

El aprendizaje automático acelerado de procesador paralelo también puede usarse para acelerar el procesamiento de lenguaje natural. Los procedimientos de aprendizaje automático pueden hacer uso de algoritmos de inferencia estadísticos para producir modelos que son robustos a entrada errónea o desconocida. Las aplicaciones de procesador del lenguaje natural ilustrativas incluyen la traducción de máquina automática entre idiomas humanos.

Las plataformas de procesamiento paralelo usadas para aprendizaje automático pueden dividirse en plataformas de entrenamiento y plataformas de despliegue. Las plataformas de entrenamiento generalmente son altamente paralelas e incluyen optimizaciones para acelerar entrenamiento de único nodo y múltiples GPU y entrenamiento de múltiples nodos y múltiples GPU. Los procesadores paralelos ilustrativos adecuados para el entrenamiento incluyen la unidad de procesamiento de gráficos de fin general altamente paralelo y el sistema informático de múltiples GPU. Por el contrario, las plataformas de aprendizaje automático desplegadas incluyen procesadores paralelos de baja potencia adecuados para su uso en productos, tales como cámaras, robots autónomos y vehículos autónomos.

La Figura 15 ilustra un sistema de inferencia en un chip (SOC) 1500 ilustrativo adecuado para realizar inferencia usando un modelo entrenado. El SOC 1500 puede integrar componentes de procesamiento que incluyen un procesador de medios 1502, un procesador de visión 1504, una GPGPU 1506 y un procesador de múltiples núcleos 1508. El SOC 1500 puede incluir adicionalmente una memoria en chip 1505 que puede posibilitar un agrupamiento de datos en chip compartidos que es accesible por cada uno de los componentes de procesamiento. Los componentes de procesamiento pueden optimizarse para operación de baja potencia para posibilitar el despliegue a una diversidad de plataformas de aprendizaje automático, incluyendo vehículos autónomos y robots autónomos. Por ejemplo, una implementación del SOC 1500 puede usarse como una porción del sistema de control principal para un vehículo autónomo. Donde el SOC 1500 está configurado para su uso en vehículos autónomos, el SOC se diseña y configura para cumplir con las normas de seguridad funcional pertinentes de la jurisdicción de despliegue.

Durante la operación, el procesador de medios 1502 y el procesador de visión 1504 puede trabajar en concierto para acelerar las operaciones de visión por ordenador. El procesador de medios 1502 puede posibilitar decodificación de baja latencia de múltiples flujos de vídeo de alta resolución (por ejemplo, 4K, 8K). Los flujos de vídeo decodificados pueden escribirse en una memoria intermedia en la memoria en chip 1505. El procesador de visión 1304 puede analizar, a continuación, el vídeo decodificado y realizar operaciones de procesamiento preliminares en los fotogramas del vídeo decodificado en preparación del procesamiento de fotogramas usando un modelo de reconocimiento de imágenes entrenado. Por ejemplo, el procesador de visión 1504 puede acelerar las operaciones de convolución para una CNN que se usa para realizar reconocimiento de imágenes en los datos de vídeo de alta resolución, mientras la GPGPU 1506 realiza los cálculos de modo de extremo final.

El procesador de múltiples núcleos 1508 puede incluir lógica de control para ayudar con la secuenciación y sincronización de transferencias de datos y operaciones de memoria compartida realizadas por el procesador de medios 1502 y el procesador de visión 1504. El procesador de múltiples núcleos 1308 también puede funcionar como un procesador de aplicación para ejecutar aplicaciones de software que pueden hacer uso de la capacidad de cálculo de inferencia de la GPGPU 1506. Por ejemplo, al menos una porción de la lógica de navegación y conducción puede implementarse en software que se ejecuta en el procesador de múltiples núcleos 1508. Tal software puede emitir directamente cargas de trabajo de cálculo a la GPGPU 1506 o las cargas de trabajo de cálculo pueden emitirse al procesador de múltiples núcleos 1508, que puede descargar al menos una porción de esas operaciones en la GPGPU 1506.

La GPGPU 1506 puede incluir agrupaciones de cálculo tales como una configuración de baja potencia de las agrupaciones de cálculo 906A-906H dentro de la unidad de procesamiento de gráficos de fin general altamente paralelo 900. Las agrupaciones de cálculo dentro de la GPGPU 1506 soportan instrucciones que se optimizan específicamente para realizar cálculos de inferencias en una red neuronal entrenada. La GPGPU 1506 soporta instrucciones para realizar cálculos de baja precisión, tales como operaciones vectoriales de enteros de 8 bits y 4 bits.

Sistema de procesamiento de gráficos ilustrativo adicional

Detalles de las realizaciones descritas anteriormente pueden incorporarse dentro de sistemas y dispositivos de procesamiento de gráficos descritos a continuación. El sistema y dispositivos de procesamiento de gráficos de las **Figuras 16-29** ilustran sistemas y hardware de procesamiento de gráficos alternativos que pueden implementar cualquiera y todas las técnicas anteriormente descritas.

Visión general de sistema de procesamiento de gráficos ilustrativo adicional

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

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

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

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

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

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

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

usarse también otros tipos de sistemas de procesamiento de datos que están configurados de manera diferente. Por ejemplo, el concentrador del controlador de E/S 1630 puede estar integrado dentro del uno o más procesadores 1602, o el concentrador de controlador de memoria 1616 y el concentrador de controlador de E/S 1630 pueden estar integrados en un procesador de gráficos externo discreto, tal como el procesador de gráficos externo 1612.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Motor de procesamiento de gráficos

La Figura 19 es un diagrama de bloques de un motor de procesamiento de gráficos 1910 de un procesador de gráficos de acuerdo con algunas realizaciones. En una realización, el motor de procesamiento de gráficos (GPE) 1910 es una versión del GPE 1810 mostrado en la **Figura 18.** Los elementos de la **Figura 19** que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en otra parte en el presente documento, pero no están limitados a este tipo. Por ejemplo, se ilustran la tubería 3D 1812 y la tubería de medios 1816 de la **Figura 18**. La tubería de medios 1816 es opcional en algunas realizaciones del GPE 1910 y puede no incluirse explícitamente dentro del GPE 1910. Por ejemplo, y en al menos una realización, un procesador de medios y/o de imagen separado está acoplado al GPE 1910.

En algunas realizaciones, el GPE 1910 se acopla con o incluye un emisor de envío por flujo continuo de comando 1903, que proporciona un flujo de comandos a la tubería 3D 1812 y/o a las tuberías de medios 1816. En algunas realizaciones, el emisor de flujo continuo de comando 1903 está acoplado con memoria, que puede ser memoria de sistema, o una o más de memoria caché interna y memoria caché compartida. En algunas realizaciones, el emisor de

flujo continuo de comando 1903 recibe comandos desde la memoria y envía los comandos a la tubería 3D 1812 y/o a la tubería de medios 1816. Los comandos son directivas extraídas desde una memoria intermedia en anillo, que almacena comandos para la tubería 3D 1812 y la tubería de medios 1816. En una realización, la memoria intermedia en anillo puede incluir adicionalmente memorias intermedias de comando por lotes que almacenan lotes de múltiples comandos. Los comandos para la tubería 3D 1812 pueden incluir también referencias a datos almacenados en memoria, tales como, pero sin limitación, datos de vértices y geometría para la tubería 3D 1812 y/o datos de imagen y objetos de memoria para la tubería de medios 1816. La tubería 3D 1812 y la tubería de medios 1816 procesan los comandos y los datos realizando operaciones mediante lógica dentro de las respectivas tuberías o despachando uno o más hilos de ejecución a una matriz de núcleo de gráficos 1914.

En diversas realizaciones, la tubería 3D 1812 puede ejecutar uno o más programas sombreadores, tales como sombreadores de vértices, sombreadores de geometría, sombreadores de píxel, sombreadores de fragmento, sombreadores de cálculo u otros programas sombreadores, procesando las instrucciones y despachando los hilos de ejecución a la matriz de núcleo de gráficos 1914. La matriz de núcleo de gráficos 1914 proporciona un bloque unificado de recursos de ejecución. La lógica de ejecución de múltiples fines (por ejemplo, las unidades de ejecución) dentro de la matriz de núcleo de gráficos 1914 incluye el soporte de diversos leguajes de sombreador de API de 3D y puede ejecutar múltiples hilos de ejecución simultánea asociados con múltiples sombreadores.

En algunas realizaciones, la matriz de núcleo de gráficos 1914 también incluye la lógica de ejecución para realizar funciones de medios, tales como el procesamiento de vídeo y/o de imagen. En una realización, las unidades de ejecución incluyen adicionalmente lógica de fin general que es programable para realizar operaciones computacionales de fin general paralelas, además de operaciones de procesamiento de gráficos. La lógica de fin general puede realizar operaciones de procesamiento en paralelo o en conjunto con la lógica de fin general dentro del núcleo o núcleos de procesador 1607 de la **Figura 16** o del núcleo 1702A-1702N como en la **Figura 17.**

Los datos de salida generados por los hilos que se ejecutan en la matriz de núcleo de gráficos 1914 pueden emitir datos a memoria en una memoria intermedia de retorno unificada (URB) 1918. La URB 1918 puede almacenar datos para múltiples hilos. En algunas realizaciones, puede usarse la URB 1918 para enviar datos entre diferentes hilos que se ejecutan en la matriz de núcleo de gráficos 1914. En algunas realizaciones, la URB 1918 puede usarse adicionalmente para la sincronización entre hilos en la matriz de núcleo de gráficos y en la lógica de función fija dentro de la lógica de función compartida 1920.

En algunas realizaciones, la matriz de núcleo de gráficos 1914 es escalable, de manera que la matriz incluye un número variable de núcleos de gráficos, teniendo cada uno un número variable de unidades de ejecución basándose en la potencia objetivo y en el nivel de rendimiento del GPE 1910. En una realización, los recursos de ejecución son dinámicamente escalables, de manera que los recursos de ejecución pueden activarse o desactivarse según sea necesario.

La matriz de núcleo de gráficos 1914 se acopla con la lógica de función compartida 1920 que incluye múltiples recursos que se comparten entre los núcleos de gráficos en la matriz de núcleo de gráficos. Las funciones compartidas dentro de la lógica de función compartida 1920 son unidades de lógica de hardware que proporcionan funcionalidad complementaria especializada a la matriz de núcleo de gráficos 1914. En diversas realizaciones, la lógica de función compartida 1920 incluye, pero sin limitación, el muestreador 1921, el cálculo 1922 y la lógica de comunicación inter hilo (ITC) 1923. Adicionalmente, algunas realizaciones implementan una o más cachés 1925 dentro de la lógica de función compartida 1920. Se implementa una función compartida donde la demanda para una función especializada es insuficiente para la inclusión dentro de la matriz de núcleo de gráficos 1914. En su lugar, se implementa una única instanciación de esa función especializada como una entidad independiente en la lógica de función compartida 1920 y se comparte entre los recursos de ejecución dentro de la matriz de núcleo de gráficos 1914. El conjunto preciso de funciones que se comparten entre la matriz de núcleo de gráficos 1914 y que están incluidas dentro de la matriz de núcleo de gráficos 1914 varía entre las realizaciones.

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

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

En algunas realizaciones, el procesador de gráficos 2000 recibe lotes de comandos mediante la interconexión en anillo 2002. Los comandos entrantes se interpretan por un emisor por flujo continuo de comando 2003 en el extremo frontal de la tubería 2004. En algunas realizaciones, el procesador de gráficos 2000 incluye lógica de ejecución escalable para realizar el procesamiento de geometría en 3D y el procesamiento de medios mediante el núcleo o núcleos de

gráficos 2080A-2080N. Para los comandos de procesamiento de geometría en 3D, el emisor de envío por flujo continuo 2003 suministra comandos a la tubería de geometría 2036. Para al menos algunos comandos de procesamiento de medios, el emisor por flujo continuo de comando 2003 suministra los comandos a un extremo frontal de vídeo 2034, que se acopla con un motor de medios 2037. En algunas realizaciones, el motor de medios 2037 incluye un motor de calidad de vídeo (VQE) 2030 para posprocesamiento de vídeo y de imagen y un motor de codificación/decodificación multiformato (MFX) 2033 para proporcionar la codificación y decodificación de datos de medios acelerados por hardware. En algunas realizaciones, cada uno de la tubería de geometría 2036 y del motor de medios 2037 genera hilos de ejecución para los recursos de ejecución de hilo proporcionados por al menos un núcleo de gráficos 2080A.

En algunas realizaciones, el procesador de gráficos 2000 incluye recursos de ejecución de hilo escalables que presentan núcleos modulares 2080A-2080N (en ocasiones denominados cortes de núcleo), teniendo cada uno múltiples subnúcleos 2050A-550N, 2060A-2060N (en ocasiones denominados subcortes de núcleo). En algunas realizaciones, el procesador de gráficos 2000 puede tener cualquier número de núcleos de gráficos 2080A a 2080N. En algunas realizaciones, el procesador de gráficos 2000 incluye un núcleo de gráficos 2080A que tiene al menos un primer subnúcleo 2050A y un segundo subnúcleo 2060A. En otras realizaciones, el procesador de gráficos es un procesador de baja potencia con un único subnúcleo (por ejemplo, 2050A). En algunas realizaciones, el procesador de gráficos 2000 incluye múltiples núcleos de gráficos 2080A-2080N, incluyendo cada uno un conjunto de primeros subnúcleos 2050A-2050N y un conjunto de segundos subnúcleos 2060A-2060N. Cada subnúcleo en el conjunto de primeros subnúcleos 2050A-2050N incluye al menos un primer conjunto de unidades de ejecución 2052A-2052N y muestreadores de medios/textura 2054A-2054N. Cada subnúcleo en el conjunto de segundos subnúcleos 2060A-2060N incluye al menos un segundo conjunto de unidades de ejecución 2062A-2062N y los muestreadores 2064A-2064N. En algunas realizaciones, cada subnúcleo 2050A-2050N, 2060A-2060N comparte un conjunto de recursos compartidos 2070A-2070N. En algunas realizaciones, los recursos compartidos incluyen memoria caché compartida y lógica de operación de píxel. Pueden incluirse también otros recursos compartidos en las diversas realizaciones del procesador de gráficos.

Unidades de ejecución

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

En algunas realizaciones, la lógica de ejecución de hilo 2100 incluye un procesador de sombreador 2102, un despachador de hilo 2104, caché de instrucciones 2106, una matriz de unidad de ejecución escalable que incluye una pluralidad de unidades de ejecución 2108A-2108N, un muestreador 2110, una caché de datos 2112 y un puerto de datos 2114. En una realización, la matriz de unidad de ejecución escalable puede escalar dinámicamente activando o desactivando una o más unidades de ejecución (por ejemplo, cualquiera de la unidad de ejecución 2108A, 2108B, 2108C, 2108D, a 2108N-1 y 2108N) basándose en los requisitos computacionales de una carga de trabajo. En una realización, los componentes incluidos están interconectados mediante un tejido de interconexión que enlaza a cada uno de los componentes. En algunas realizaciones, la lógica de ejecución de hilo 2100 incluye una o más conexiones a memoria, tal como la memoria de sistema o memoria caché, a través de uno o más de la caché de instrucciones 2106, el puerto de datos 2114, el muestreador 2110 y las unidades de ejecución 2108A-2108N. En algunas realizaciones, cada unidad de ejecución (por ejemplo, 2108A) es una unidad computacional de fin general programable independiente que puede ejecutar múltiples hilos de hardware simultáneos mientras procesa múltiples elementos de datos en paralelo para cada hilo. En diversas realizaciones, la matriz de unidades de ejecución 2108A-2108N es escalable para incluir cualquier número de unidades de ejecución individuales.

En algunas realizaciones, las unidades de ejecución 2108A-2108N se usan principalmente para ejecutar programas sombreadores. Un procesador de sombreador 2102 puede procesar los diversos programas sombreadores y despachar hilos de ejecución asociados con los programas sombreadores mediante un despachador de hilo 2104. En una realización, el despachador de hilo incluye lógica para arbitrar las solicitudes de iniciación de hilo de las tuberías de gráficos y de medios e instanciar los hilos solicitados en una o más unidades de ejecución en las unidades de ejecución 2108A-2108N. Por ejemplo, la tubería de geometría (por ejemplo, 2036 de la Figura 20) puede despachar sombreadores de vértice, de teselación o de geometría a la lógica de ejecución de hilo 2100 **(Figura 21)** para su procesamiento. En algunas realizaciones, el despachador de hilo 2104 puede procesar también hilos en tiempo de ejecución que generan solicitudes de los programas sombreadores en ejecución.

En algunas realizaciones, las unidades de ejecución 2108A-2108N soportan un conjunto de instrucciones que incluye soporte nativo para muchas instrucciones de sombreador de gráficos en 3D convencionales, de manera que se ejecutan los programas sombreadores de las librerías gráficas (por ejemplo, Direct 3D y OpenGL) con una traducción mínima. Las unidades de ejecución soportan procesamiento de vértices y de geometría (por ejemplo, programas de vértices, programas de geometría, sombreadores de vértices), procesamiento de píxeles (por ejemplo, sombreadores de píxeles, sombreadores de fragmentos), y procesamiento de fin general (por ejemplo, sombreadores de cálculo y de medios). Cada una de las unidades de ejecución 2108A-2108N puede emitir de manera múltiple la ejecución de datos de múltiples instrucciones sencillas (SIMD) y la operación de múltiples hilos posibilita un entorno de ejecución

eficiente frente a accesos a memoria de latencia superior. Cada hilo de hardware dentro de cada unidad de ejecución tiene un fichero de registro de alto ancho de banda especializado y un estado de hilo independiente asociado. La ejecución se emite múltiples veces por reloj a las tuberías aptas para operaciones de números enteros, de coma flotante de precisión sencilla y doble, capacidad de ramal SIMD, operaciones lógicas, operaciones transcendentales y otras operaciones de miscelánea. Mientras se esperan los datos de la memoria o una de las funciones compartidas, la lógica de dependencia dentro de las unidades de ejecución 2108A-2108N hace que un hilo en espera pase a inactividad hasta que se hayan devuelto los datos solicitados. Mientras el hilo en espera está inactivo, los recursos de hardware pueden dedicarse a procesar otros hilos. Por ejemplo, durante un retardo asociado con una operación de sombreador de vértice, una unidad de ejecución puede realizar operaciones para un sombreador de píxel, sombreador de fragmento u otro tipo de programa sombreador, que incluye un sombreador de vértice diferente.

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

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

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

Durante la ejecución, las tuberías de gráficos y de medios envían solicitudes de iniciación de hilo a la lógica de ejecución de hilo 2100 mediante lógica de generación y despacho de hilo. Una vez que se ha procesado y rasterizado un grupo de objetos geométricos en datos de píxeles, se invoca la lógica de procesador de píxel (por ejemplo, lógica de sombreador de píxel, lógica de sombreador de fragmento, etc.) dentro del procesador de sombreador 2102 para que calcule adicionalmente información de salida y haga que se escriban los resultados en las superficies de salida (por ejemplo, memorias intermedias de color, memorias intermedias de profundidad, memorias intermedias de estarcido, etc.). En algunas realizaciones, un sombreador de píxel o sombreador de fragmento calcula los valores de los diversos atributos de vértice que han de interpolarse a través del objeto rasterizado. En algunas realizaciones, la lógica de procesador de píxel dentro del procesador de sombreador 2102 ejecuta, a continuación, un programa sombreador de píxel o de fragmento suministrado por la interfaz de programación de aplicación (API). Para ejecutar el programa sombreador, el procesador de sombreador 2102 despacha hilos a una unidad de ejecución (por ejemplo, 2108A) mediante el despachador de hilo 2104. En algunas realizaciones, el sombreador de píxel 2102 usa lógica de muestreo de textura en el sombreador 2110 para acceder a datos de textura en mapas de textura almacenados en memoria. Las operaciones aritméticas en los datos de textura y en los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico o descartan uno o más píxeles de su procesamiento adicional.

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

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

En algunas realizaciones, las unidades de ejecución de procesador de gráficos soportan de manera nativa instrucciones en un formato de instrucciones de 128 bits 2210. Está disponible un formato de instrucciones de 64 bits compacto 2230 para algunas instrucciones basándose en la instrucción seleccionada, en las opciones de la instrucción y en el número de operandos. El formato de instrucciones de 128 bits nativo 710 proporciona acceso a todas las opciones de instrucción, mientras que algunas opciones y operaciones están restringidas en el formato de 64 bits 2230. Las instrucciones nativas disponibles en el formato de 64 bits 2230 varían por realización. En algunas realizaciones, la instrucción se compacta en parte usando un conjunto de valores de índice en un campo de índice 2213. El hardware de la unidad de ejecución hace referencia a un conjunto de tablas de compactación basándose en los valores de índice y usa las salidas de la tabla de compactación para reconstruir una instrucción nativa en el formato de instrucción de 128 bits 2210.

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

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

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

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, se usa el modo de acceso para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones soportan modos de acceso que incluyen un modo de acceso alineado de 16 bytes y un modo de acceso alineado de 1 byte, donde la alineación de byte del modo de acceso determina la alineación de acceso de los operandos de instrucción. Por ejemplo, cuando está en un primer modo, la instrucción puede usar direccionamiento alineado en byte para operandos de origen y destino y cuando está en un segundo modo, la instrucción puede usar direccionamiento alineado en 16 bytes para todos los operandos de origen y destino.

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

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

Tubería de gráficos

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

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

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

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

En algunas realizaciones, la tubería de gráficos 2320 incluye componentes de teselación para realizar teselación acelerada por hardware de objetos 3D. En algunas realizaciones, un sombreador de casco programable 811 configura las operaciones de teselación. Un sombreador de dominio programable 817 proporciona una evaluación de extremo trasero de la salida de la teselación. Un teselador 2313 opera en la dirección del sombreador de casco 2311 y contiene la lógica de fin especial para generar un conjunto de objetos geométricos detallados basándose en un modelo geométrico aproximado que se proporciona como entrada en la tubería de gráficos 2320. En algunas realizaciones, si no se usa teselación, pueden omitirse los componentes de teselación (por ejemplo, el sombreador de casco 2311, el teselador 2313 y el sombreador de dominio 2317).

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

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

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

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

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

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

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

Programación de tubería de gráficos

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

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

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

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

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

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

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

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

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

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

En algunas realizaciones, se activa la tubería 3D 2422 mediante un comando o evento de ejecución 2434. En algunas realizaciones, una escritura de registro activa la ejecución del comando. En algunas realizaciones, se activa la ejecución mediante un comando 'ir' o 'disparar' en la secuencia de comandos. En una realización, se activa la ejecución

del comando usando un comando de sincronización de tubería para vaciar la secuencia de comandos a través de la tubería de gráficos. La tubería 3D realizará un procesamiento de geometría para las primitivas 3D. Una vez que las operaciones están completadas, se rasterizan los objetos geométricos resultantes y el motor de píxeles colorea los píxeles resultantes. Pueden incluirse también comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo trasero para estas operaciones.

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

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

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

Arquitectura de software de gráficos

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

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

En algunas realizaciones, el sistema operativo 2520 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo similar a UNIX propietario o un sistema operativo similar a UNIX de código abierto que usa una variante del núcleo de Linux. El sistema operativo 2520 puede soportar una API de gráficos 2522 tal como la API Direct3D, la API OpenGL o la API Vulkan. Cuando está en uso la API Direct3D, el sistema operativo 2520 usa un compilador de sombreador de extremo frontal 2524 para compilar cualquier instrucción del sombreador 2512 en HLSL en un lenguaje de sombreador de nivel inferior. La compilación puede ser una compilación justo a tiempo (JIT) o la aplicación puede realizar compilación previa de sombreador. En algunas realizaciones, se compilan sombreadores de alto nivel en sombreadores de bajo nivel durante la compilación de la aplicación de gráficos en 3D 2510. En algunas realizaciones, se proporcionan las instrucciones de sombreador 2512 en una forma intermedia, tal como una versión de la Representación Intermedia Portátil Convencional (SPIR) usada por la API Vulkan.

En algunas realizaciones, el controlador de gráficos de modo de usuario 2526 contiene un compilador de sombreador de extremo trasero 2527 para convertir las instrucciones de sombreador 2512 en una representación específica de hardware. Cuando está en uso la API OpenGL, las instrucciones de sombreador 2512 en el lenguaje de alto nivel

GLSL se pasan a un controlador de gráficos de modo de usuario 2526 para su compilación. En algunas realizaciones, el controlador de gráficos de modo de usuario 2526 usa las funciones de modo de núcleo de sistema operativo 2528 para comunicarse con un controlador de gráficos de modo de núcleo 2529. En algunas realizaciones, el controlador de gráficos de modo de núcleo 2529 se comunica con el procesador de gráficos 2532 para despachar comandos e instrucciones.

Implementaciones de núcleo de IP

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

La Figura 26 es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP 2600 que puede usarse para fabricar un circuito integrado para realizar operaciones de acuerdo con una realización. Puede usarse el sistema de desarrollo de núcleo de IP 2600 para generar diseños modulares, reutilizables que pueden incorporarse en un diseño mayor o usarse para construir un circuito integrado entero (por ejemplo, un circuito integrado SOC). Una instalación de diseño 2630 puede generar una simulación por software 2610 de un diseño de núcleo de IP en un lenguaje de programación de alto nivel (por ejemplo, C/C++). La simulación de software 2610 puede usarse para diseñar, probar y verificar el comportamiento del núcleo de IP usando un modelo de simulación 2612. El modelo de simulación 2612 puede incluir simulaciones funcionales, de comportamiento y/o de temporización. A continuación, puede crearse un diseño de nivel de transferencia de registro (RTL) 2615 o sintetizarse a partir del modelo de simulación 2612. El diseño RTL 2615 es una abstracción del comportamiento del circuito integrado que modela el flujo de señales digitales entre registro de hardware, que incluye la lógica asociada realizada usando las señales digitales modeladas. Además de un diseño RTL 2615, pueden crearse, diseñarse o sintetizarse también diseños de nivel inferior al nivel de lógica o al nivel de transistores. Por lo tanto, los detalles particulares del diseño y simulación iniciales pueden variar.

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

Sistema ilustrativo en un circuito de chip integrado

Las Figuras 27-29 ilustraron circuitos integrados ilustrativos y procesadores de gráficos asociados que pueden fabricarse usando uno o más núcleos de IP, de acuerdo con diversas realizaciones descritas en el presente documento. Además de lo que se ilustra, puede incluirse otra lógica y circuitos, incluyendo procesadores/núcleos de gráficos adicionales, controladores de interfaz de periférico o núcleos de procesador de fin general.

La Figura 27 es un diagrama de bloques que ilustra un sistema ilustrativo en un circuito de chip integrado 2700 que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El circuito integrado 2700 ilustrativo incluye uno o más procesador o procesadores 2705 de aplicación (por ejemplo, la o las CPU), al menos un procesador de gráficos 2710, y puede incluir adicionalmente un procesador de imagen 2715 y/o un procesador de vídeo 2720, cualquiera de los cuales puede ser un núcleo de IP modular de las mismas o múltiples instalaciones de diseño diferentes. El circuito integrado 2700 incluye lógica de periféricos o de bus que incluye un controlador USB 2725, un controlador UART 2730, un controlador SPI/SDIO 2735 y un controlador I^2S/I^2C 2740. Adicionalmente, el circuito integrado puede incluir un dispositivo de visualización 2745 acoplado a uno o más de un controlador de interfaz multimedia de alta definición (HDMI) 2750 y una interfaz de visualización de la interfaz de procesador industrial móvil (MIPI) 2755. El almacenamiento puede proporcionarse por un subsistema de memoria flash 2760 que incluye memoria flash y un controlador de memoria flash. La interfaz de memoria puede proporcionarse mediante un controlador de memoria 2765 para acceso a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad integrado 2770.

La Figura 28 es un diagrama de bloques que ilustra un procesador de gráficos 2810 ilustrativo de un sistema en un circuito de chip integrado que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El procesador de gráficos 2810 puede ser una variante del procesador de gráficos 2710 de la **Figura 27**. El procesador de gráficos 2810 incluye un procesador de vértices 2805 y uno o más procesador o procesadores de fragmentos 2815A-2815N (por ejemplo, 2815A, 2815B, 2815C, 2815D, a 2815N-1 y 2815N). El procesador de gráficos 2810 puede ejecutar diferentes programas sombreadores mediante lógica separada, de manera que el procesador de vértices 2805 está optimizado para ejecutar operaciones para programas sombreadores de vértices, mientras que el uno o más procesador o procesadores de fragmentos 2815A-2815N ejecutan operaciones de sombreado de fragmentos (por ejemplo, píxeles) para programas sombreadores de fragmentos o píxeles. El procesador de vértices 2805 realiza la etapa de procesamiento de vértices de la tubería de gráficos 3D y genera primitivas y datos de vértices. El procesador o procesadores de fragmentos 2815A-2815N usan los datos de primitivas y de vértices generados por el procesador de vértices 2805 para producir una memoria intermedia de tramas que se visualiza en un dispositivo de visualización. En una realización, el procesador o procesadores de fragmentos 2815A-2815N están optimizados para ejecutar programas sombreadores de fragmentos según se proporcionan en la API OpenGL, que pueden usarse para realizar operaciones similares como un programa sombreador de píxeles según se proporcionan para la API Direct 3D.

El procesador de gráficos 2810 incluye adicionalmente una o más unidades de gestión de memoria (MMU) 2820A-2820B, caché o cachés 2825A-2825B e interconexión o interconexiones de circuito 2830A-2830B. La una o más MMU 2820A-2820B proporcionan un mapeo de direcciones virtual a físico para el circuito integrado 2810, que incluye el procesador de vértices 2805 y/o el procesador o procesadores de fragmentos 2815A-2815N, que pueden hacer referencia a datos de vértices o de imagen/textura almacenados en memoria, además de datos de vértices o de imagen/textura almacenados en la una o más caché o cachés 2825A-2825B. En una realización, la una o más MMU 2825A-2825B pueden estar sincronizadas con otras MMU dentro del sistema, que incluye una o más MMU asociadas con el uno o más procesador o procesadores de aplicación 2705, el procesador de imagen 2715 y/o el procesador de vídeo 2720 de la **Figura 27,** de manera que cada procesador 2705-2720 puede participar en un sistema de memoria virtual compartido o unificado. La una o más interconexiones de circuito 2830A-2830B posibilitan que el procesador de gráficos 2810 se interconecte con otros núcleos de IP dentro del SoC, mediante un bus interno del SoC o mediante una conexión directa, de acuerdo con las realizaciones.

La Figura 29 es un diagrama de bloques que ilustra un procesador de gráficos 2910 ilustrativo adicional de un sistema en un circuito de chip integrado que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El procesador de gráficos 2910 puede ser una variante del procesador de gráficos 2710 de la **Figura 27.** El procesador de gráficos 2910 incluye la una o más MMU 2820A-2820B, cachés 2825A-2825B e interconexiones de circuito 2830A-2830B del circuito integrado 2800 de la **Figura 28.**

El procesador de gráficos 2910 incluye uno o más núcleo o núcleos de sombreador 2915A-2915N (por ejemplo, 2915A, 2915B, 2915C, 2915D, 2915E, 2915F, a 2915N-1 y 2915N), que proporcionan una arquitectura de núcleo de sombreador unificada en la que un único núcleo o tipo o núcleo pueden ejecutar todos los tipos de código sombreador programables, que incluyen código de programa sombreador para implementar sombreadores de vértices, sombreadores de fragmentos y/o sombreadores de cálculo. El número exacto de núcleos de sombreadores presente puede variar entre realizaciones e implementaciones. Adicionalmente, el procesador de gráficos 2910 incluye un gestor de tareas inter-núcleo 2905, que actúa como un despachador de hilo para despachar hilos de ejecución a uno o más núcleos de sombreador 2915A-2915N y una unidad de mosaico 2918 para acelerar operaciones de mosaico para representación basada en mosaicos, en la que las operaciones de representación para una escena se subdividen en el espacio de las imágenes, por ejemplo, para aprovechar la coherencia espacial local dentro de una escena o para optimizar el uso de cachés internas.

REIVINDICACIONES

1. Una unidad de procesamiento de gráficos (200; 614; 900; 1506), que comprende:

un conjunto de controladores de memoria (912A-B; 1714);
una memoria caché (908; 1604); y
al menos una agrupación de cálculo (214; 906A-H) que comprende al menos un multiprocesador de gráficos (234; 325; 407; 1608) acoplado al conjunto de controladores de memoria (912A, 912B; 1714), incluyendo el al menos un multiprocesador de gráficos (234; 325; 407; 1608) una unidad de instrucciones (254; 332A-B), una pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N) y una memoria compartida (270) acoplada a la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N), en donde la unidad de instrucciones (254; 332A-B) está configurada para despachar instrucciones para su ejecución por al menos uno de la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N), en donde un mecanismo de cálculo (610) soporta una instrucción de multiplicación-acumulación fusionada de precisión mixta, FMAC, de D = A*B + C con operandos que tienen diferentes precisiones;
caracterizada por que
la unidad de procesamiento de gráficos (200; 614; 900; 1506) comprende una configuración de inferencia de baja potencia de la al menos una agrupación de cálculo (214; 906A-H) para realizar cálculos de inferencias en una red neuronal entrenada; y
la instrucción de FMAC se soporta por la configuración de inferencia, en donde A y B son elementos de datos enteros de 8 bits y C es un elemento de datos entero de 32 bits.

2. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de la reivindicación 1, en donde el al menos un multiprocesador de gráficos tiene una arquitectura de única instrucción, múltiples hilos, SIMT.

3. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de la reivindicación 2, en donde la arquitectura de SIMT incluye multihilos de hardware.

4. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de una cualquiera de las reivindicaciones 1 a 3, en donde la unidad de procesamiento de gráficos comprende, además, al menos una unidad de operaciones de rasterización, ROP (226).

5. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de una cualquiera de las reivindicaciones 1 a 4, en donde el al menos un multiprocesador de gráficos (234; 325; 407; 1608) comprende, además, un fichero de registro (258; 334A-334B; 1606) para almacenar operandos.

6. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de la reivindicación 5, en donde el al menos un multiprocesador de gráficos (234; 325; 407; 1608) está configurado para cargar datos asociados con operandos de la instrucción de FMAC D = A * B + C en el fichero de registro (258; 334A-334B; 1606) desde la memoria caché (908; 1604).

7. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de la reivindicación 5, en donde el al menos un multiprocesador de gráficos (234; 325; 407; 1608) está configurado para cargar datos asociados con operandos de la instrucción de FMAC D = A * B + C en el fichero de registro (258; 334A-334B; 1606) desde la memoria compartida (270).

8. La unidad de procesamiento de gráficos (200; 614; 900; 1506) de una cualquiera de las reivindicaciones 1 a 7, en donde el al menos un multiprocesador de gráficos (234; 325; 407; 1608) comprende, además, una caché de instrucciones para almacenar al menos una instrucción para su ejecución por la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N) del al menos un multiprocesador de gráficos (234; 325; 407; 1608), en donde la al menos una instrucción se ejecuta como una envoltura.

9. Un sistema de procesamiento de gráficos (1600; 1700), que comprende:

al menos un dispositivo de memoria de gráficos (433-434; 1620);
la unidad de procesamiento de gráficos (200; 614; 900; 1506) como se cita en una cualquiera de las reivindicaciones 1 a 8,
en donde el conjunto de controladores de memoria (912A-B; 1714) se acopla adicionalmente al al menos un dispositivo de memoria de gráficos (433-434; 1620).

10. El sistema de procesamiento de gráficos (1600; 1700) de la reivindicación 9, en donde el al menos un dispositivo de memoria de gráficos (433-434) comprende una memoria de doble tasa de datos de gráficos, GDDR.

11. El sistema de procesamiento de gráficos (1600; 1700) de la reivindicación 10, en donde la memoria de GDDR comprende una DDR-SDRAM de gráficos, en particular una memoria de GDDR6.

12. El sistema de procesamiento de gráficos (1600; 1700) de una cualquiera de las reivindicaciones 9 a 11, en donde parámetros asociados con una capa de la red neuronal se procesan por el al menos un multiprocesador de gráficos.

13. El sistema de procesamiento de gráficos (1600; 1700) de la reivindicación 12, en donde los parámetros se asocian con una ponderación de la capa de la red neuronal o una entrada a la capa de la red neuronal.

14. Un método implementado en un multiprocesador de gráficos (234; 325; 407; 1608) comprendido por al menos una agrupación de cálculo (214; 906A-H) de una unidad de procesamiento de gráficos (200; 614; 900; 1506), comprendiendo el multiprocesador de gráficos una unidad de instrucciones (254; 332A-B), una pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N) y una memoria compartida (270) acoplada a la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N), en donde el multiprocesador de gráficos (234; 325; 407; 1608) se acopla adicionalmente a un conjunto de controladores de memoria (912A-B; 1714); y una memoria caché (908; 1604), en donde la unidad de procesamiento de gráficos (200; 614; 900; 1506) comprende una configuración de inferencia de baja potencia de la al menos una agrupación de cálculo (214; 906A-H) para realizar cálculos de inferencias en una red neuronal entrenada, comprendiendo el método:
despachar, por la unidad de instrucciones (254; 332A-B), instrucciones para su ejecución por al menos uno de la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N);
ejecutar, por al menos un mecanismo de cálculo (610) asociado con al menos uno de la pluralidad de núcleos de procesamiento (262; 406A-D; 1607; 1702A-1702N), una instrucción de multiplicación-acumulación fusionada de precisión mixta, FMAC, de $D = A * B + C$ con operandos que tienen diferentes precisiones, en donde la instrucción de FMAC se soporta por la configuración de inferencia, en donde A y B son elementos de datos enteros de 8 bits y C es un elemento de datos entero de 32 bits.

15. El método de la reivindicación 14, en donde el método se ejecuta por una unidad de procesamiento de gráficos (200; 614; 900; 1506) de acuerdo con una cualquiera de las reivindicaciones 1 a 8.

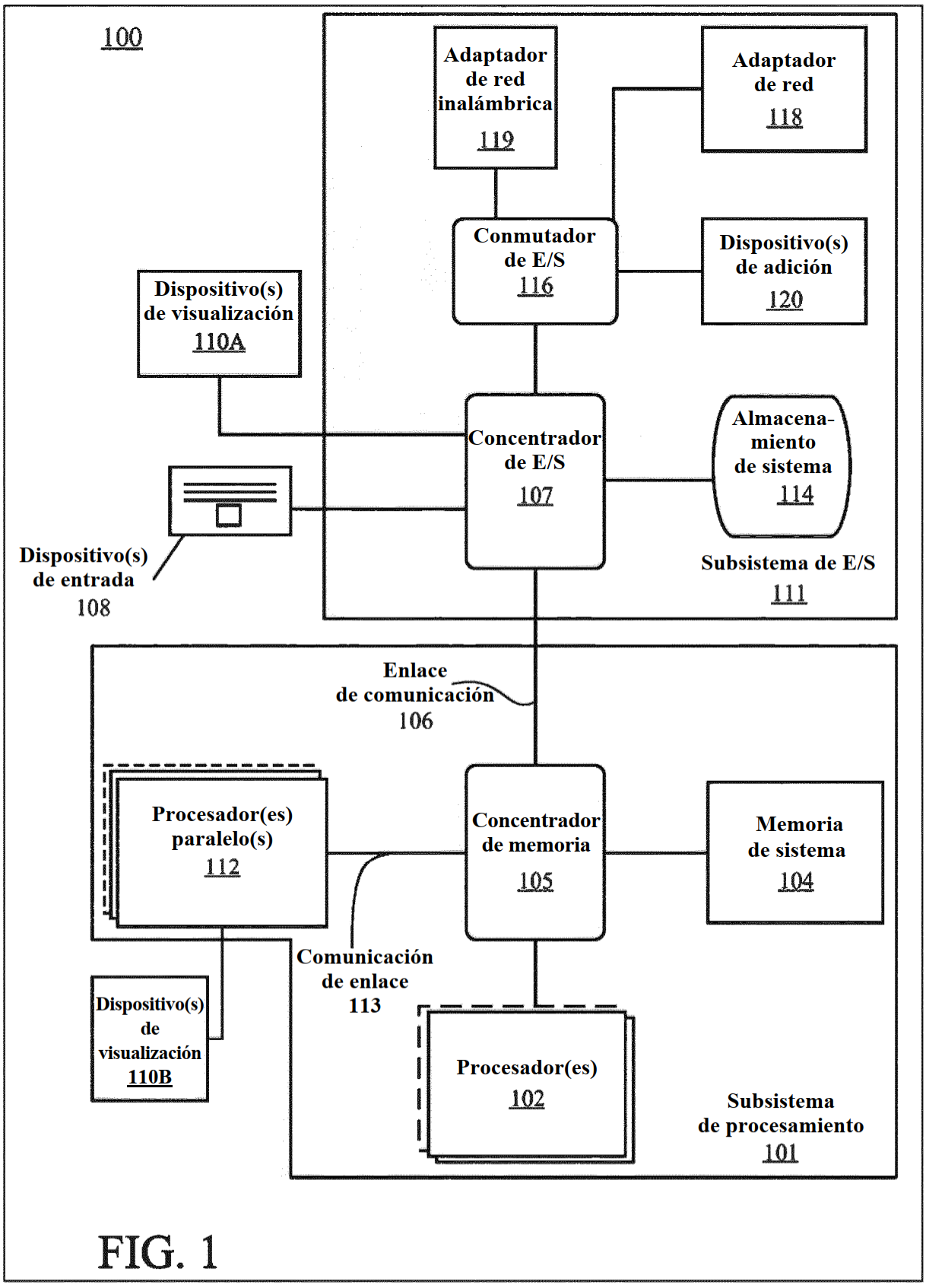

100
Adaptador de red inalámbrica 119
Adaptador de red 118
Conmutador de E/S 116
Dispositivo(s) de adición 120
Dispositivo(s) de visualización 110A
Concentrador de E/S 107
Almacena-miento de sistema 114
Dispositivo(s) de entrada 108
Subsistema de E/S 111
Enlace de comunicación 106
Procesador(es) paralelo(s) 112
Concentrador de memoria 105
Memoria de sistema 104
Comunicación de enlace 113
Dispositivo(s) de visualización 110B
Procesador(es) 102
Subsistema de procesamiento 101

FIG. 1

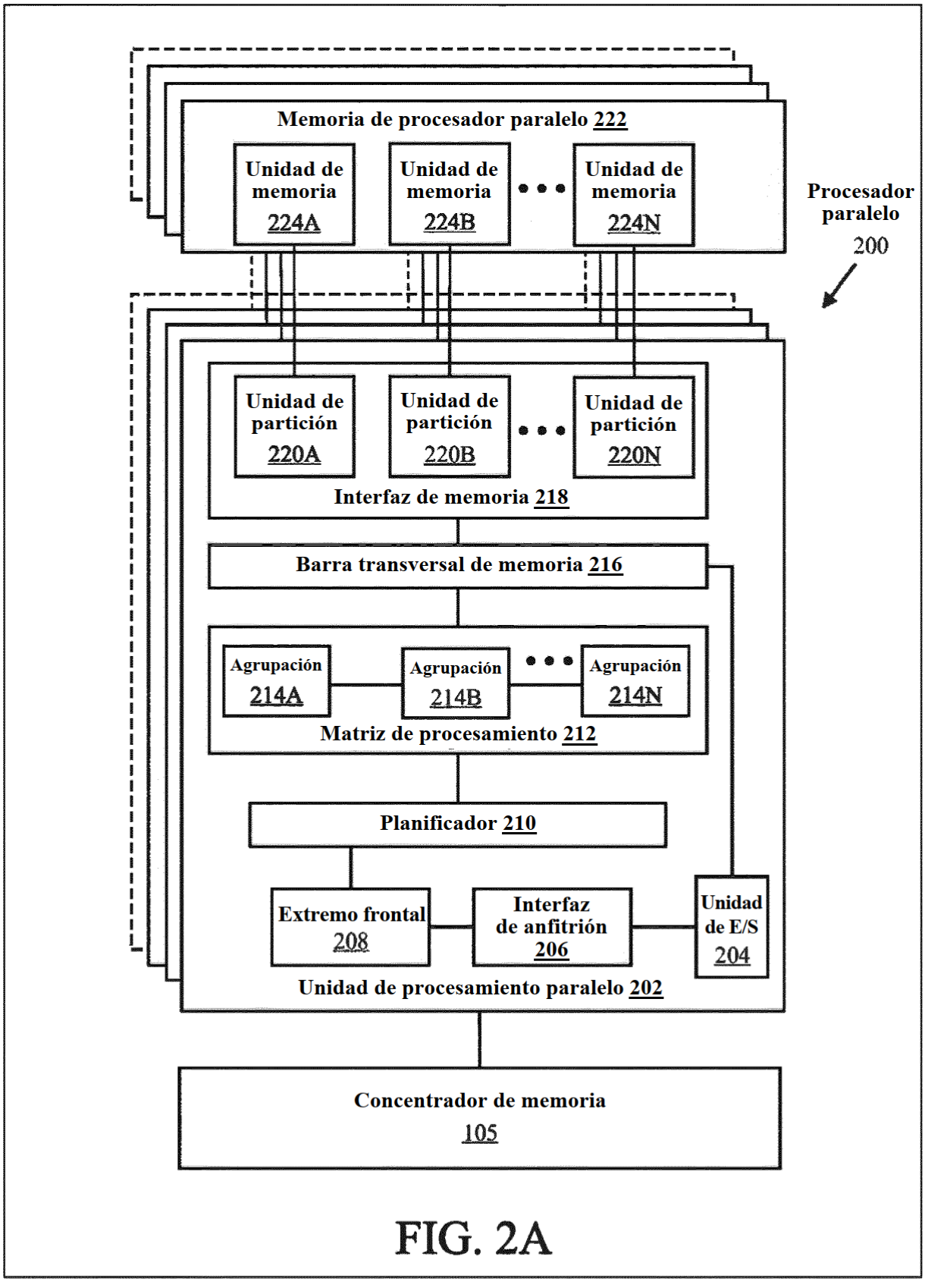

Memoria de procesador paralelo 222
Unidad de memoria 224A
Unidad de memoria 224B
Unidad de memoria 224N
Procesador paralelo 200
Unidad de partición 220A
Unidad de partición 220B
Unidad de partición 220N
Interfaz de memoria 218
Barra transversal de memoria 216
Agrupación 214A
Agrupación 214B
Agrupación 214N
Matriz de procesamiento 212
Planificador 210
Extremo frontal 208
Interfaz de anfitrión 206
Unidad de E/S 204
Unidad de procesamiento paralelo 202
Concentrador de memoria 105

FIG. 2A

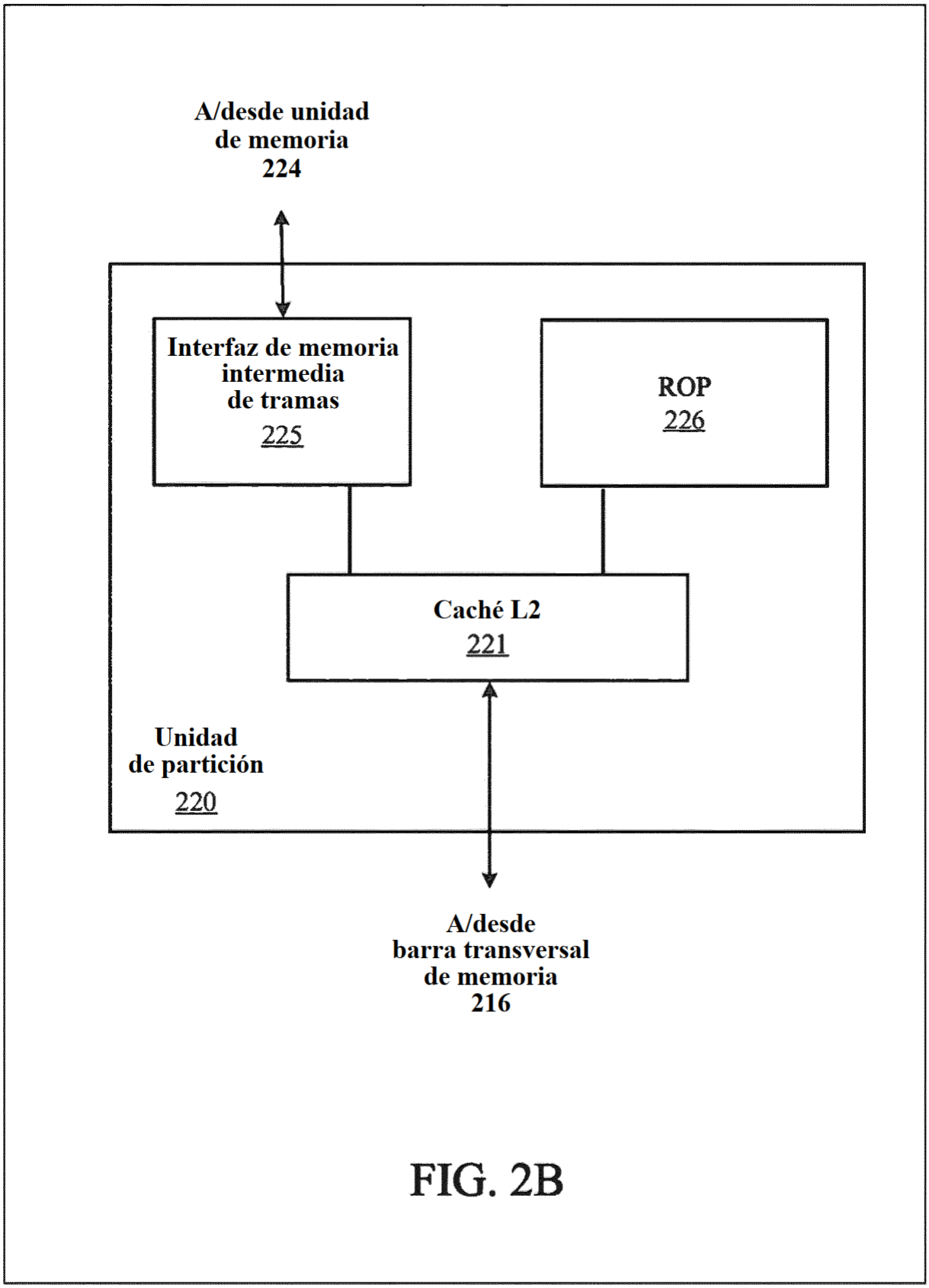

A/desde unidad
de memoria
224
Interfaz de memoria
intermedia
de tramas
225
ROP
226
Caché L2
221
Unidad
de partición
220
A/desde
barra transversal
de memoria
216

FIG. 2B

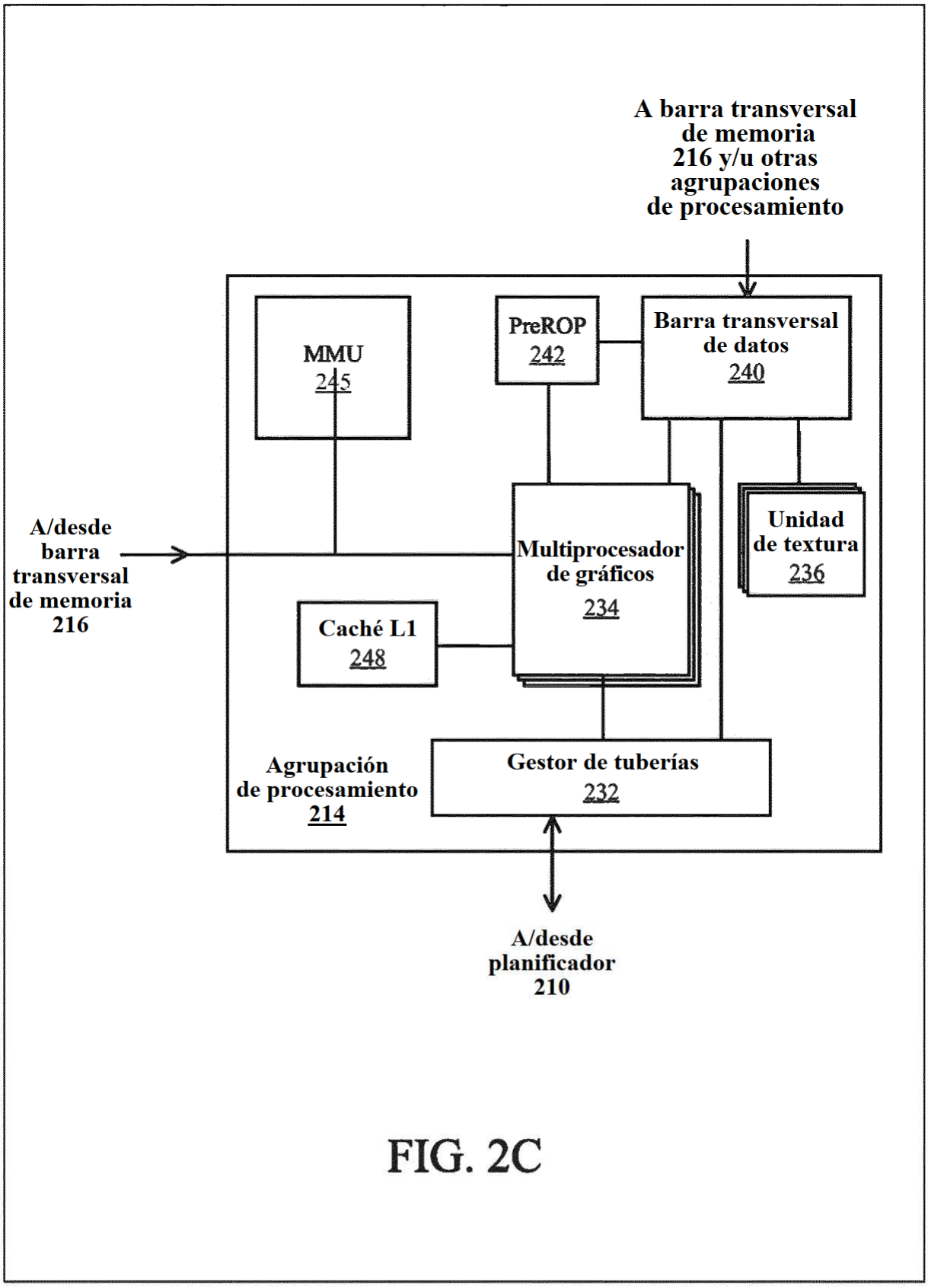

A barra transversal de memoria 216 y/u otras agrupaciones de procesamiento
MMU 245
PreROP 242
Barra transversal de datos 240
A/desde barra transversal de memoria 216
Multiprocesador de gráficos 234
Unidad de textura 236
Caché L1 248
Agrupación de procesamiento 214
Gestor de tuberías 232
A/desde planificador 210

FIG. 2C

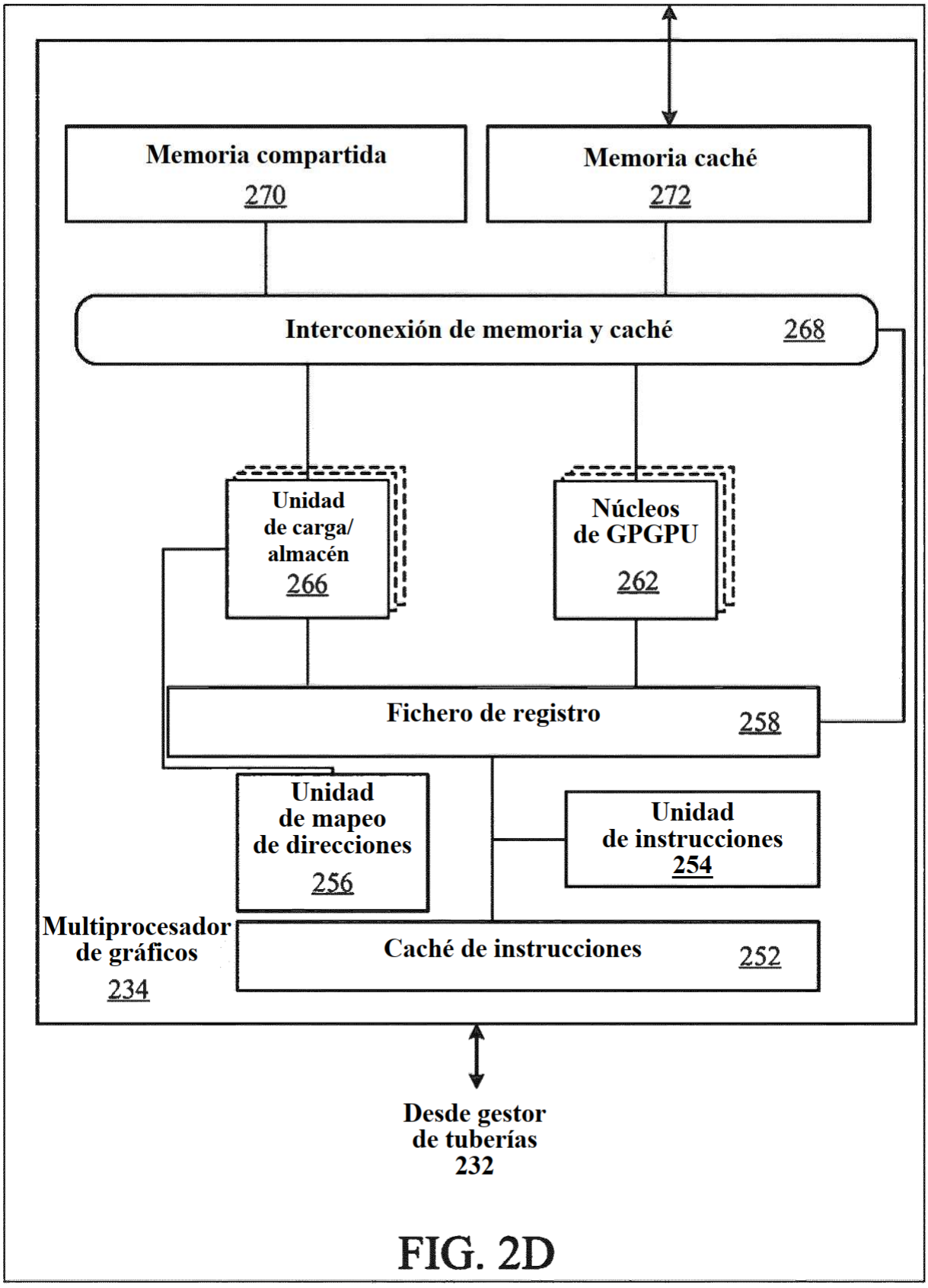

Memoria compartida
270
Memoria caché
272
Interconexión de memoria y caché
268
Unidad
de carga/
almacén
266
Núcleos
de GPGPU
262
Fichero de registro
258
Unidad
de mapeo
de direcciones
256
Unidad
de instrucciones
254
Multiprocesador
de gráficos
234
Caché de instrucciones
252
Desde gestor
de tuberías
232

FIG. 2D

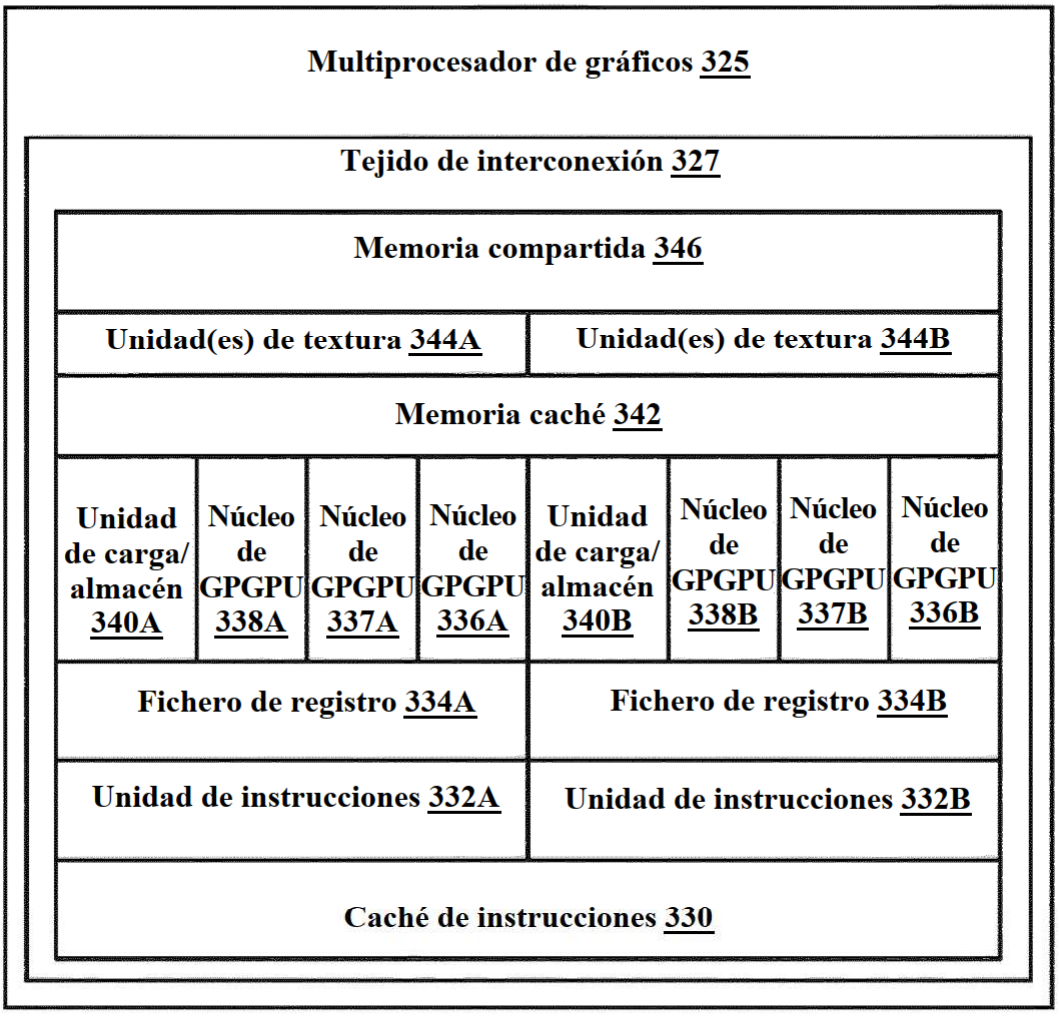

Multiprocesador de gráficos 325
Tejido de interconexión 327
Memoria compartida 346
Unidad(es) de textura 344A
Unidad(es) de textura 344B
Memoria caché 342
Unidad de carga/ almacén 340A
Núcleo de GPGPU 338A
Núcleo de GPGPU 337A
Núcleo de GPGPU 336A
Unidad de carga/ almacén 340B
Núcleo de GPGPU 338B
Núcleo de GPGPU 337B
Núcleo de GPGPU 336B
Fichero de registro 334A
Fichero de registro 334B
Unidad de instrucciones 332A
Unidad de instrucciones 332B
Caché de instrucciones 330

FIG. 3A

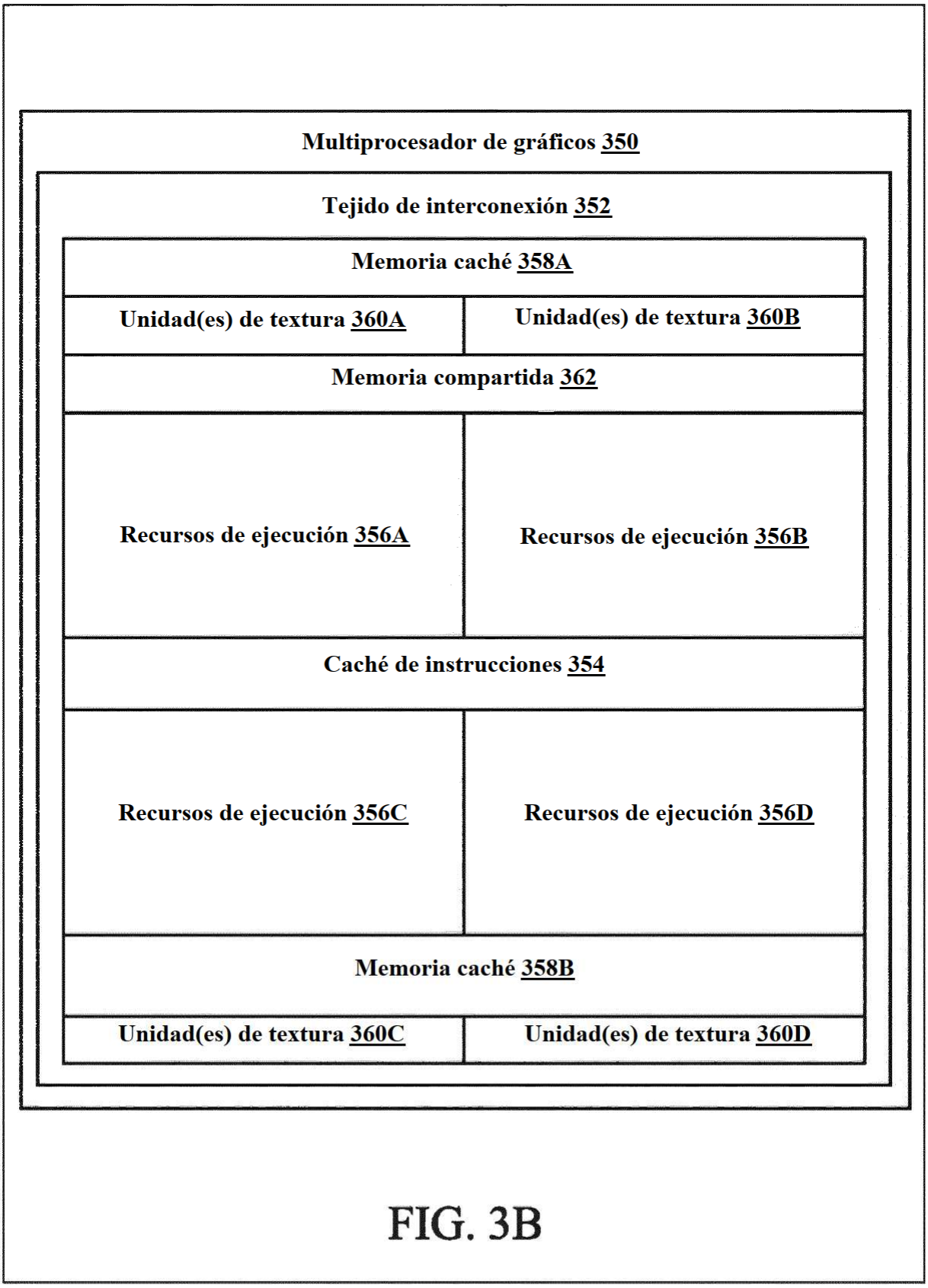

Multiprocesador de gráficos 350
Tejido de interconexión 352
Memoria caché 358A
Unidad(es) de textura 360A
Unidad(es) de textura 360B
Memoria compartida 362
Recursos de ejecución 356A
Recursos de ejecución 356B
Caché de instrucciones 354
Recursos de ejecución 356C
Recursos de ejecución 356D
Memoria caché 358B
Unidad(es) de textura 360C
Unidad(es) de textura 360D

FIG. 3B

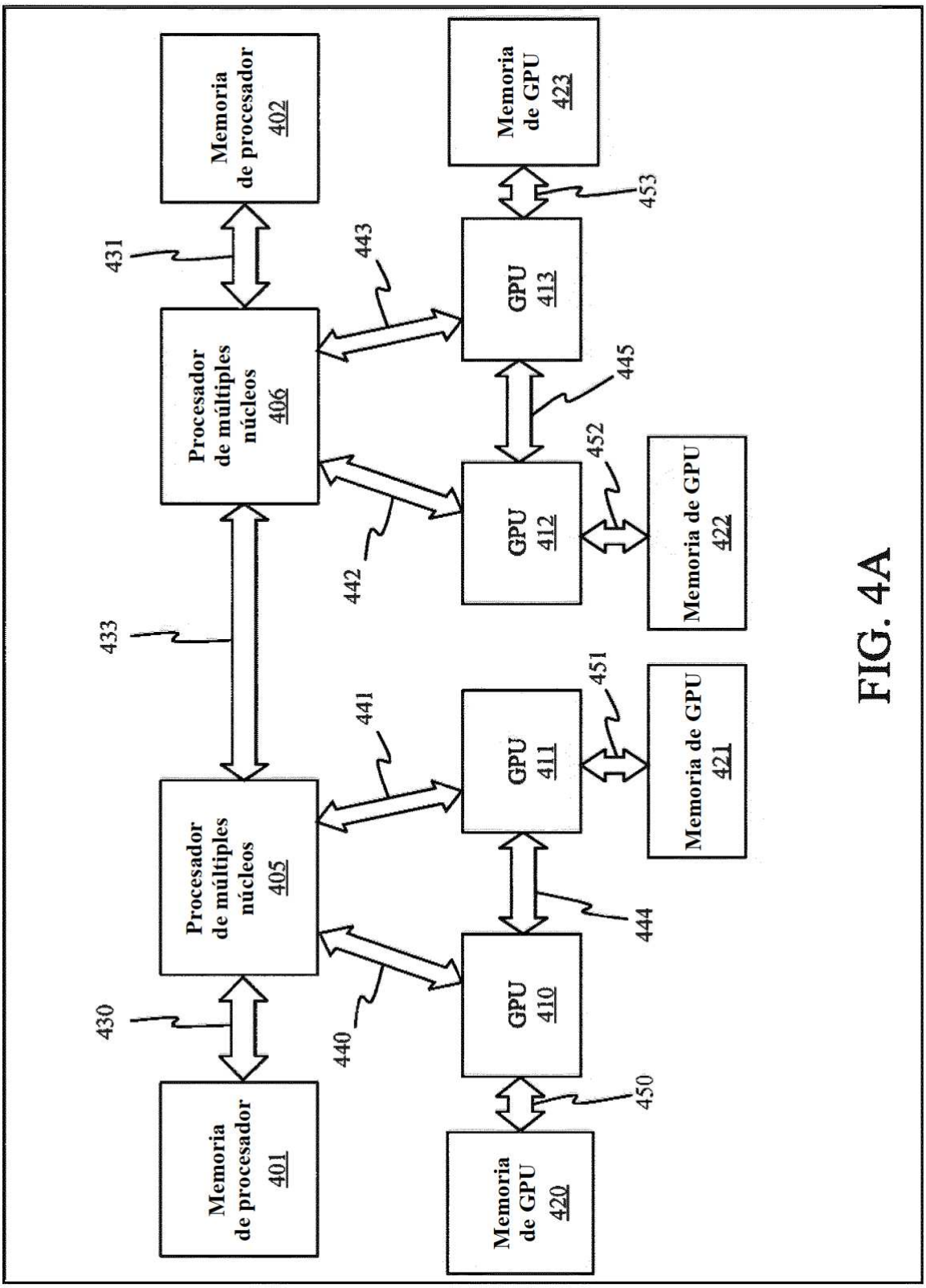

Memoria de procesador 401
Memoria de procesador 402
Procesador de múltiples núcleos 405
Procesador de múltiples núcleos 406
GPU 410
GPU 411
GPU 412
GPU 413
Memoria de GPU 420
Memoria de GPU 421
Memoria de GPU 422
Memoria de GPU 423
430
431
433
440
441
442
443
444
445
450
451
452
453

FIG. 4A

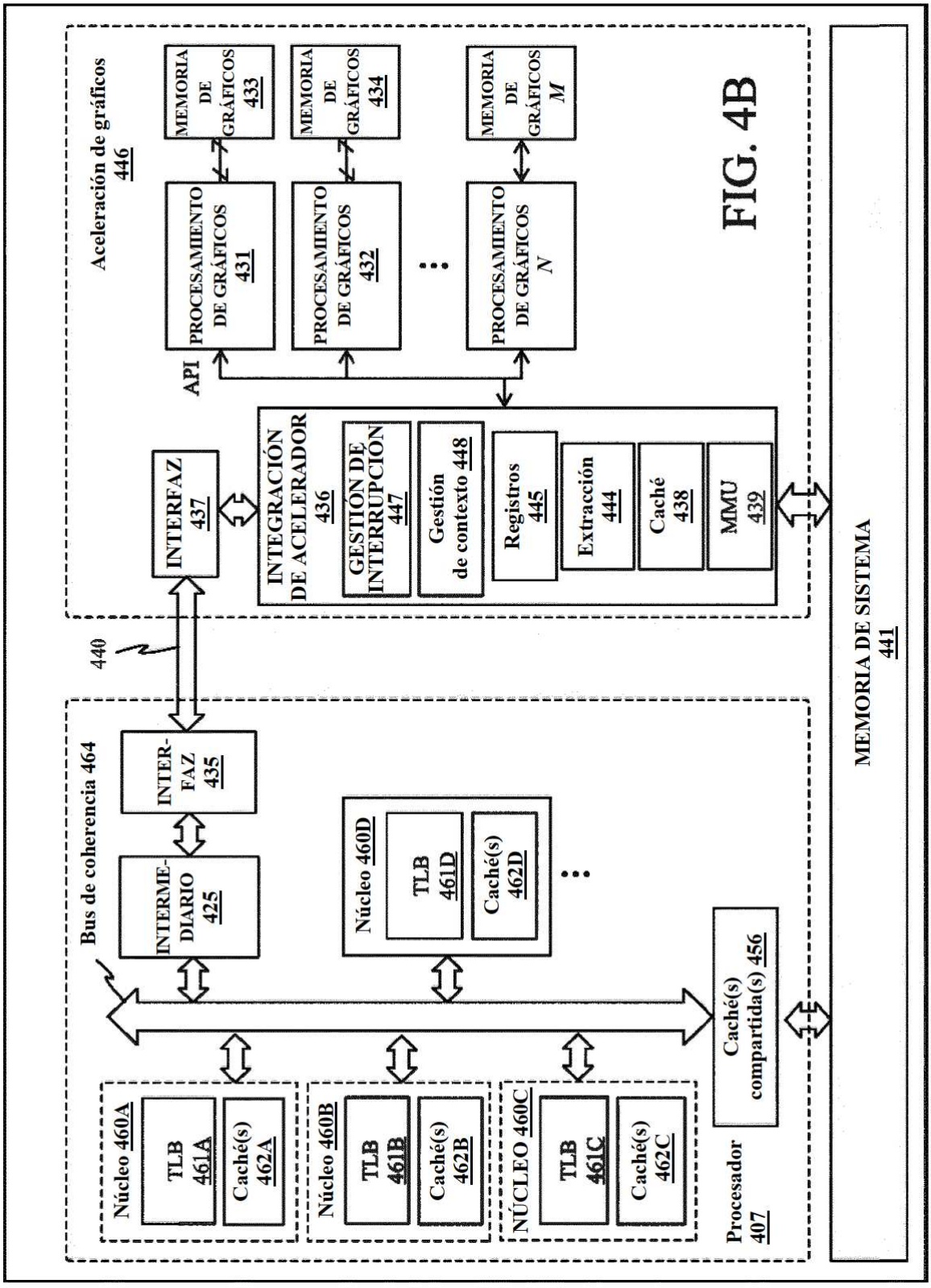

Aceleración de gráficos 446
MEMORIA DE GRÁFICOS 433
MEMORIA DE GRÁFICOS 434
MEMORIA DE GRÁFICOS M
PROCESAMIENTO DE GRÁFICOS 431
PROCESAMIENTO DE GRÁFICOS 432
PROCESAMIENTO DE GRÁFICOS N
API
INTERFAZ 437
INTEGRACIÓN DE ACELERADOR 436
GESTIÓN DE INTERRUPCIÓN 447
Gestión de contexto 448
Registros 445
Extracción 444
Caché 438
MMU 439
440
Bus de coherencia 464
INTER-FAZ 435
INTERME-DIARIO 425
Núcleo 460D
TLB 461D
Caché(s) 462D
Caché(s) compartida(s) 456
Núcleo 460A
TLB 461A
Caché(s) 462A
Núcleo 460B
TLB 461B
Caché(s) 462B
NÚCLEO 460C
TLB 461C
Caché(s) 462C
Procesador 407
MEMORIA DE SISTEMA 441

FIG. 4B

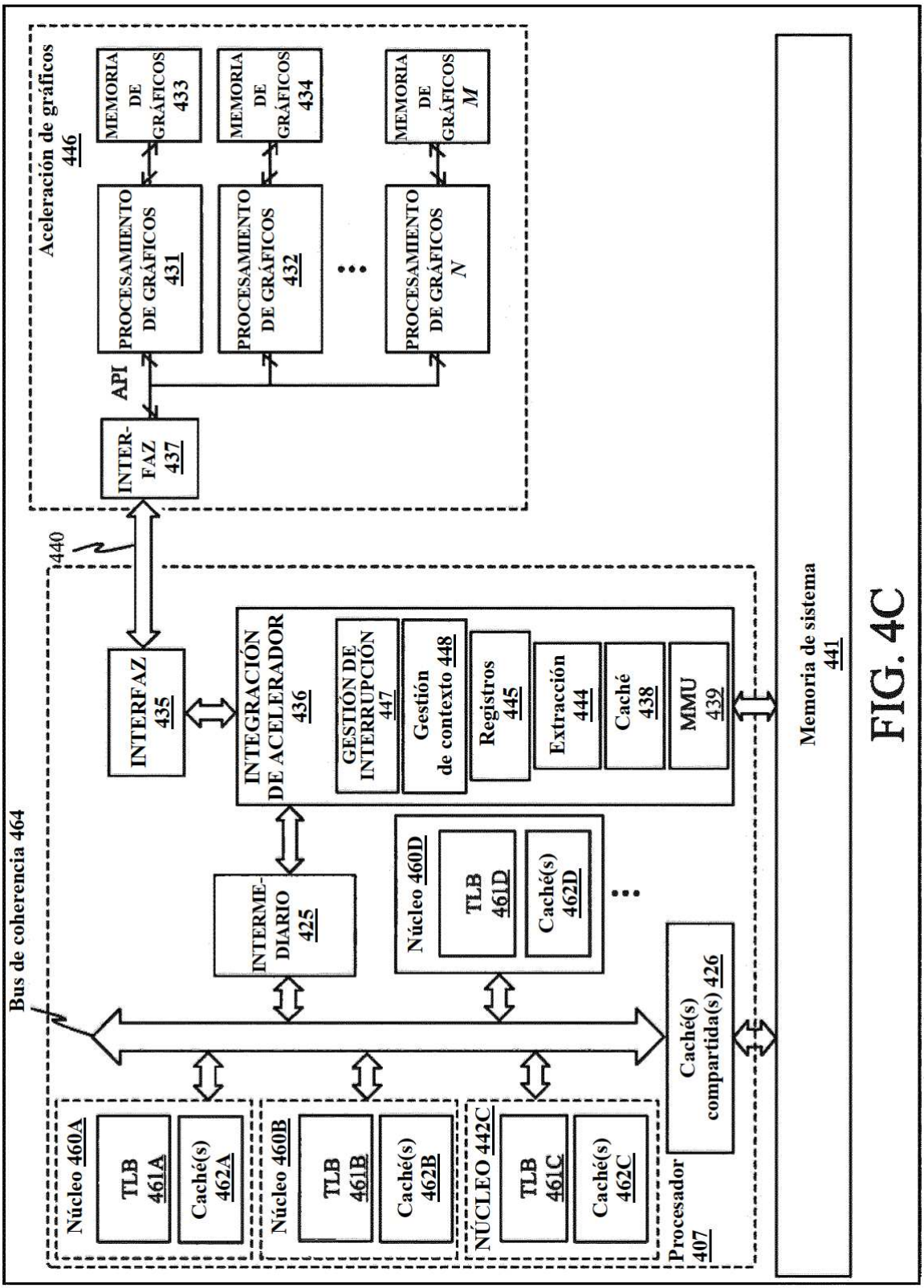

Bus de coherencia 464
Aceleración de gráficos 446
MEMORIA DE GRÁFICOS 433
MEMORIA DE GRÁFICOS 434
MEMORIA DE GRÁFICOS M
PROCESAMIENTO DE GRÁFICOS 431
PROCESAMIENTO DE GRÁFICOS 432
PROCESAMIENTO DE GRÁFICOS N
API
INTER-FAZ 437
440
INTERFAZ 435
INTEGRACIÓN DE ACELERADOR 436
GESTIÓN DE INTERRUPCIÓN 447
Gestión de contexto 448
Registros 445
Extracción 444
Caché 438
MMU 439
INTERME-DIARIO 425
Núcleo 460D
TLB 461D
Caché(s) 462D
Memoria de sistema 441
Núcleo 460A
TLB 461A
Caché(s) 462A
Núcleo 460B
TLB 461B
Caché(s) 462B
NÚCLEO 442C
TLB 461C
Caché(s) 462C
Caché(s) compartida(s) 426
Procesador 407

FIG. 4C

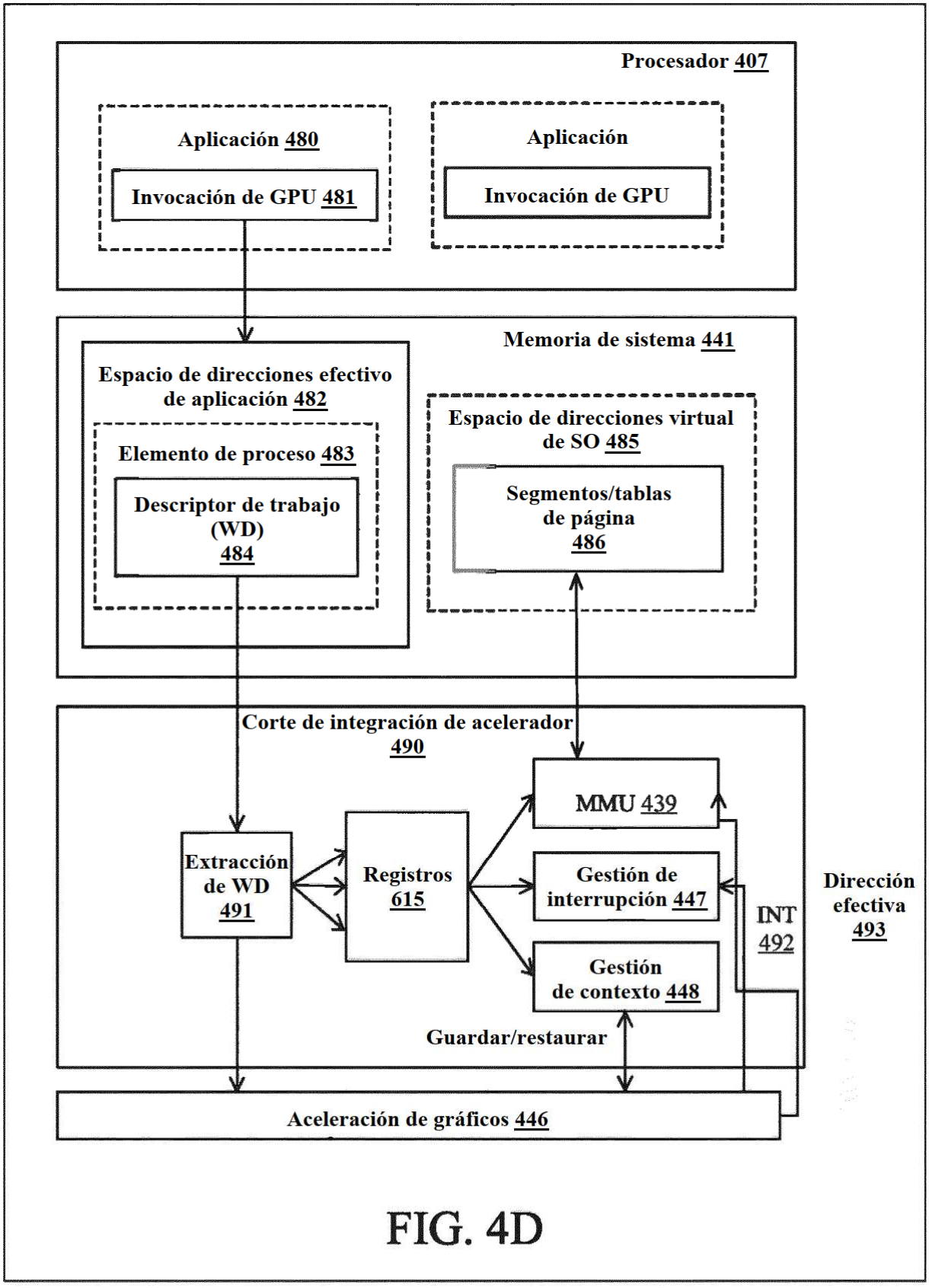

Procesador 407
Aplicación 480
Invocación de GPU 481
Aplicación
Invocación de GPU
Memoria de sistema 441
Espacio de direcciones efectivo de aplicación 482
Elemento de proceso 483
Descriptor de trabajo (WD) 484
Espacio de direcciones virtual de SO 485
Segmentos/tablas de página 486
Corte de integración de acelerador 490
MMU 439
Extracción de WD 491
Registros 615
Gestión de interrupción 447
INT 492
Dirección efectiva 493
Gestión de contexto 448
Guardar/restaurar
Aceleración de gráficos 446

FIG. 4D

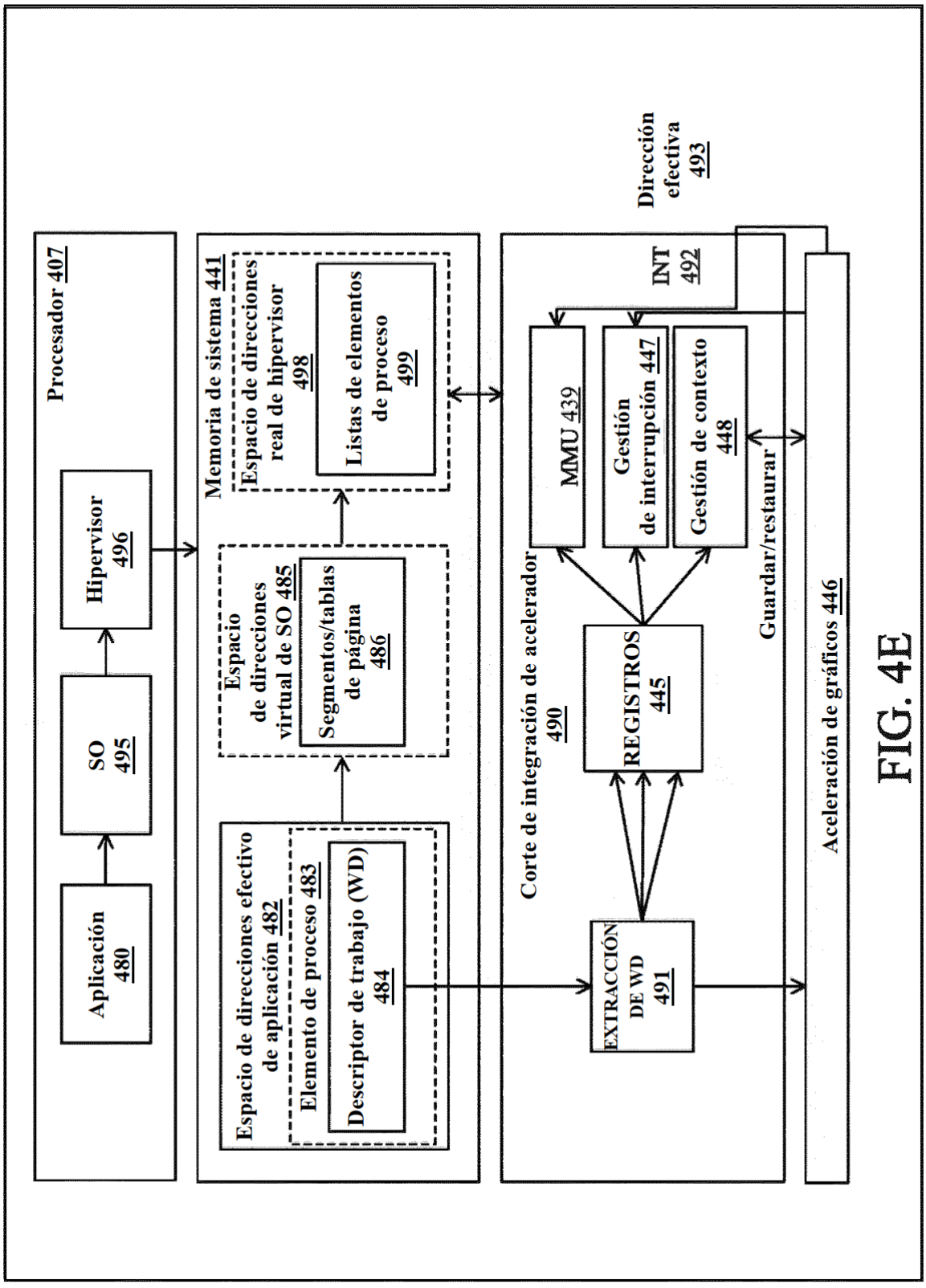

Procesador 407
Aplicación 480
SO 495
Hipervisor 496
Memoria de sistema 441
Espacio de direcciones efectivo de aplicación 482
Elemento de proceso 483
Descriptor de trabajo (WD) 484
Espacio de direcciones virtual de SO 485
Segmentos/tablas de página 486
Espacio de direcciones real de hipervisor 498
Listas de elementos de proceso 499
Corte de integración de acelerador 490
EXTRACCIÓN DE WD 491
REGISTROS 445
MMU 439
Gestión de interrupción 447
Gestión de contexto 448
INT 492
Dirección efectiva 493
Guardar/restaurar
Aceleración de gráficos 446

FIG. 4E

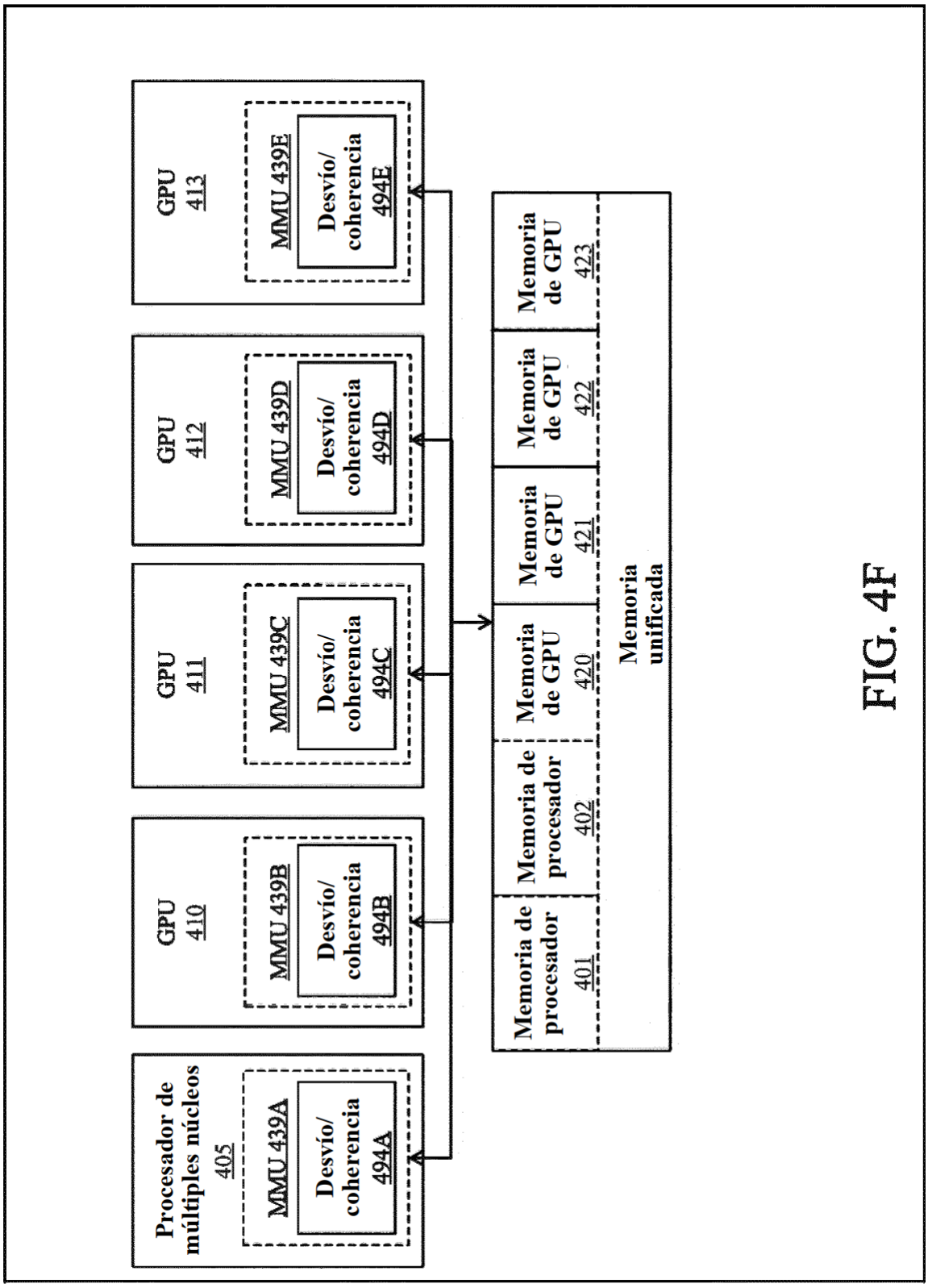

Procesador de múltiples núcleos 405
MMU 439A
Desvío/ coherencia 494A
GPU 410
MMU 439B
Desvío/ coherencia 494B
GPU 411
MMU 439C
Desvío/ coherencia 494C
GPU 412
MMU 439D
Desvío/ coherencia 494D
GPU 413
MMU 439E
Desvío/ coherencia 494E
Memoria de procesador 401
Memoria de procesador 402
Memoria de GPU 420
Memoria de GPU 421
Memoria de GPU 422
Memoria de GPU 423
Memoria unificada

FIG. 4F

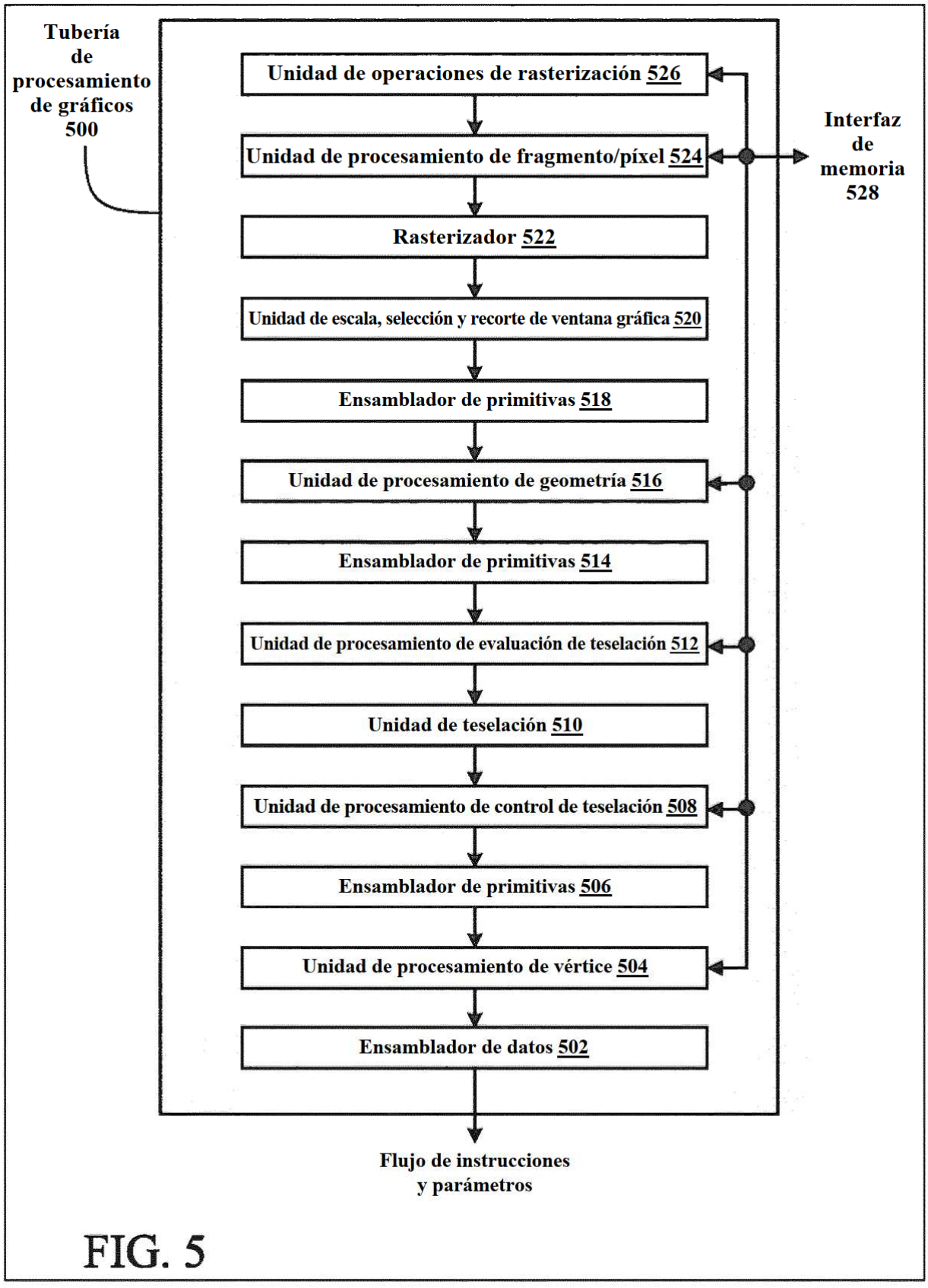

Tubería de procesamiento de gráficos 500
Unidad de operaciones de rasterización 526
Unidad de procesamiento de fragmento/píxel 524
Interfaz de memoria 528
Rasterizador 522
Unidad de escala, selección y recorte de ventana gráfica 520
Ensamblador de primitivas 518
Unidad de procesamiento de geometría 516
Ensamblador de primitivas 514
Unidad de procesamiento de evaluación de teselación 512
Unidad de teselación 510
Unidad de procesamiento de control de teselación 508
Ensamblador de primitivas 506
Unidad de procesamiento de vértice 504
Ensamblador de datos 502
Flujo de instrucciones y parámetros

FIG. 5

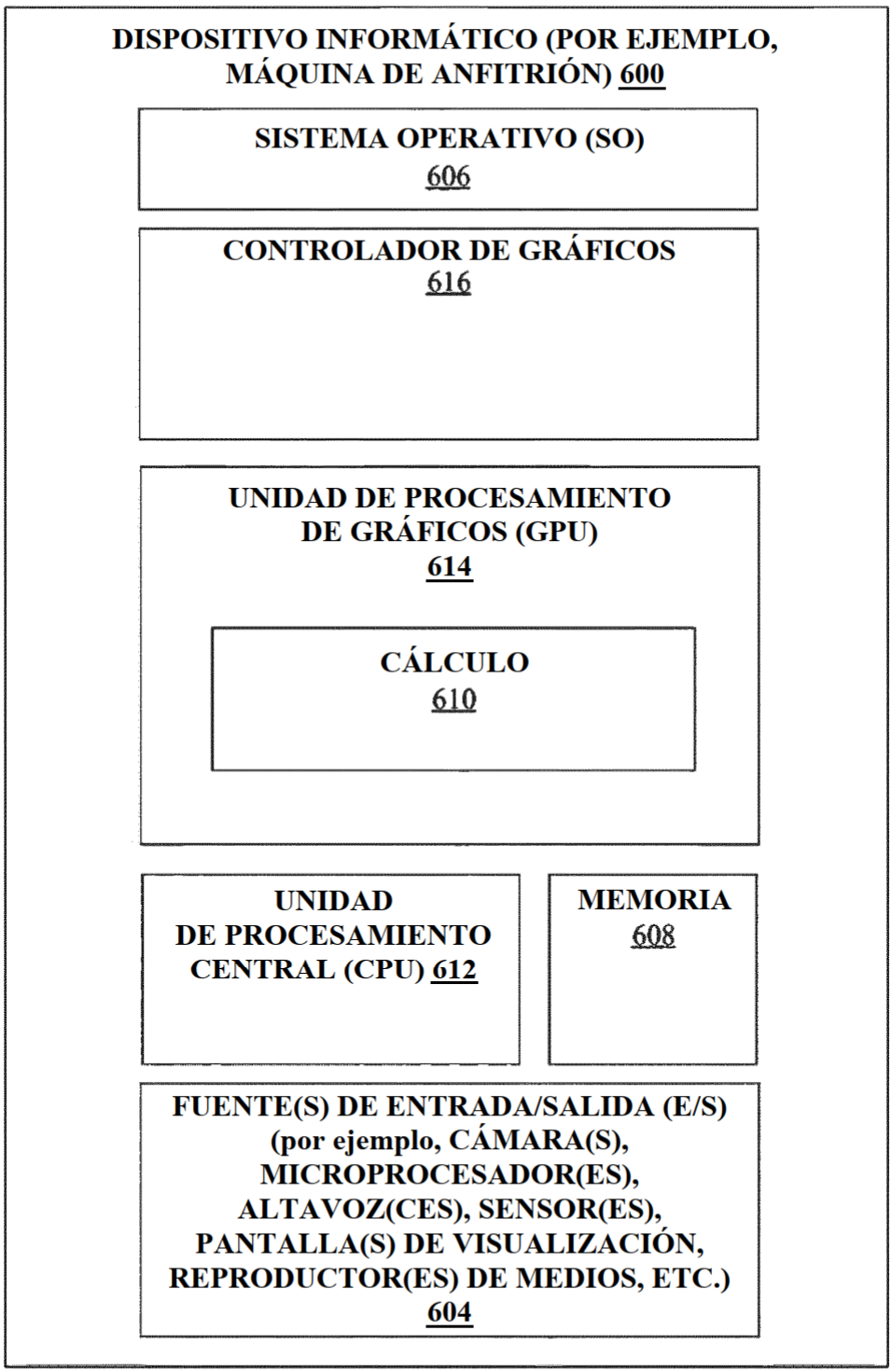

DISPOSITIVO INFORMÁTICO (POR EJEMPLO, MÁQUINA DE ANFITRIÓN) 600
SISTEMA OPERATIVO (SO) 606
CONTROLADOR DE GRÁFICOS 616
UNIDAD DE PROCESAMIENTO DE GRÁFICOS (GPU) 614
CÁLCULO 610
UNIDAD DE PROCESAMIENTO CENTRAL (CPU) 612
MEMORIA 608
FUENTE(S) DE ENTRADA/SALIDA (E/S) (por ejemplo, CÁMARA(S), MICROPROCESADOR(ES), ALTAVOZ(CES), SENSOR(ES), PANTALLA(S) DE VISUALIZACIÓN, REPRODUCTOR(ES) DE MEDIOS, ETC.) 604

FIG. 6

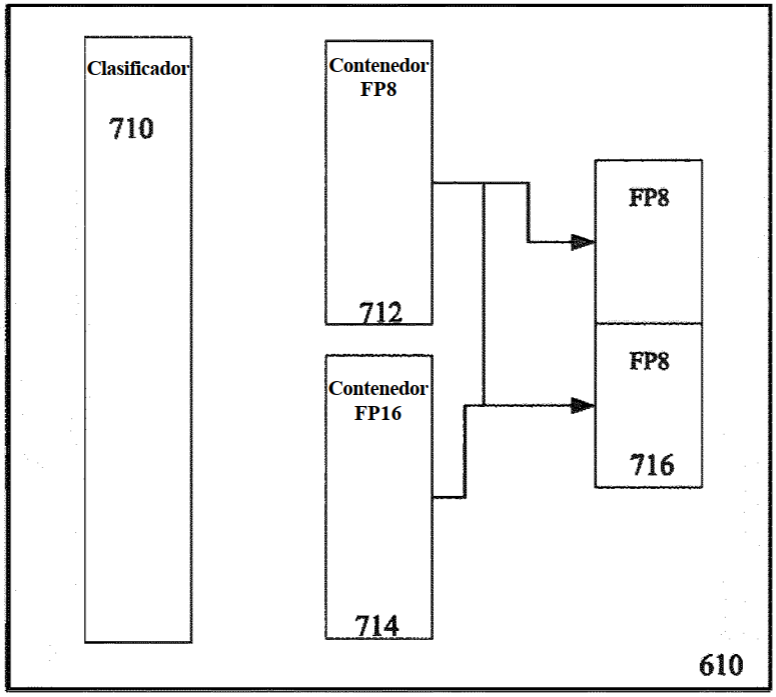

Clasificador
710
Contenedor FP8
712
Contenedor FP16
714
FP8
FP8
716
610

FIG. 7A

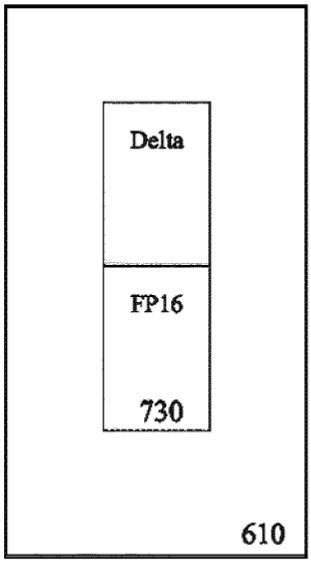

Delta
FP16
730
610

FIG. 7B

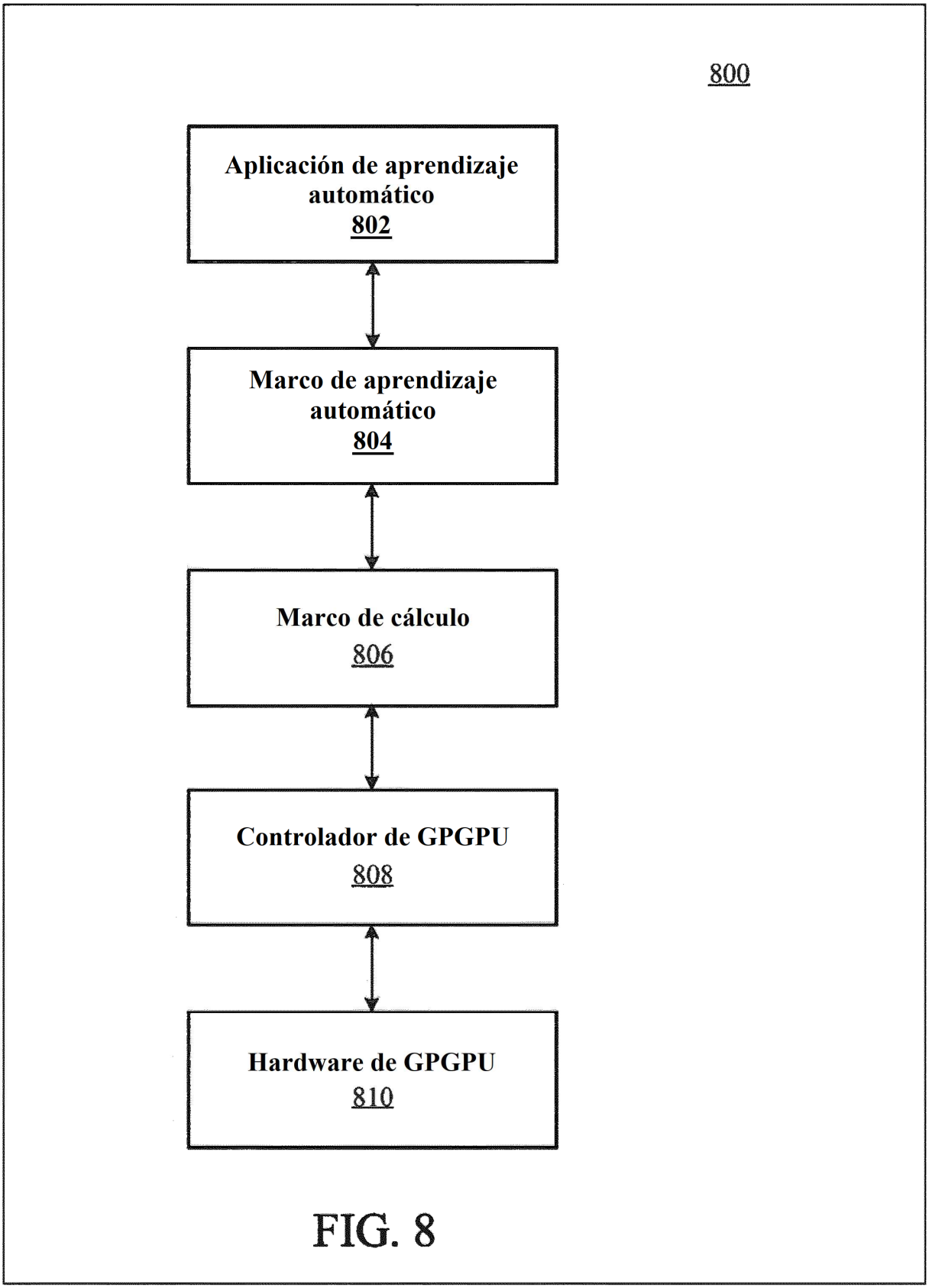

800
Aplicación de aprendizaje automático
802
Marco de aprendizaje automático
804
Marco de cálculo
806
Controlador de GPGPU
808
Hardware de GPGPU
810

FIG. 8

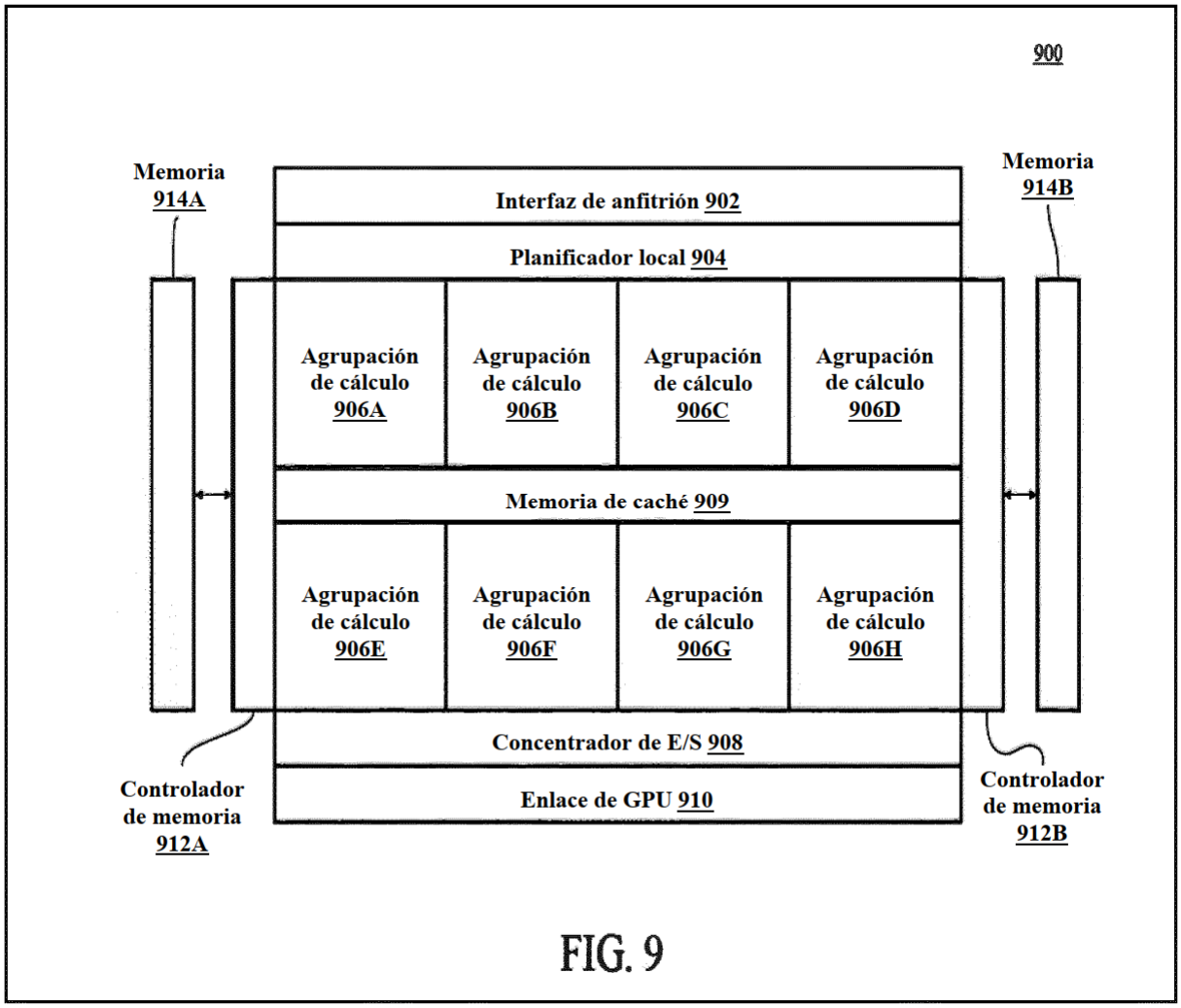

900
Memoria
914A
Memoria
914B
Interfaz de anfitrión 902
Planificador local 904
Agrupación de cálculo 906A
Agrupación de cálculo 906B
Agrupación de cálculo 906C
Agrupación de cálculo 906D
Memoria de caché 909
Agrupación de cálculo 906E
Agrupación de cálculo 906F
Agrupación de cálculo 906G
Agrupación de cálculo 906H
Concentrador de E/S 908
Enlace de GPU 910
Controlador de memoria 912A
Controlador de memoria 912B

FIG. 9

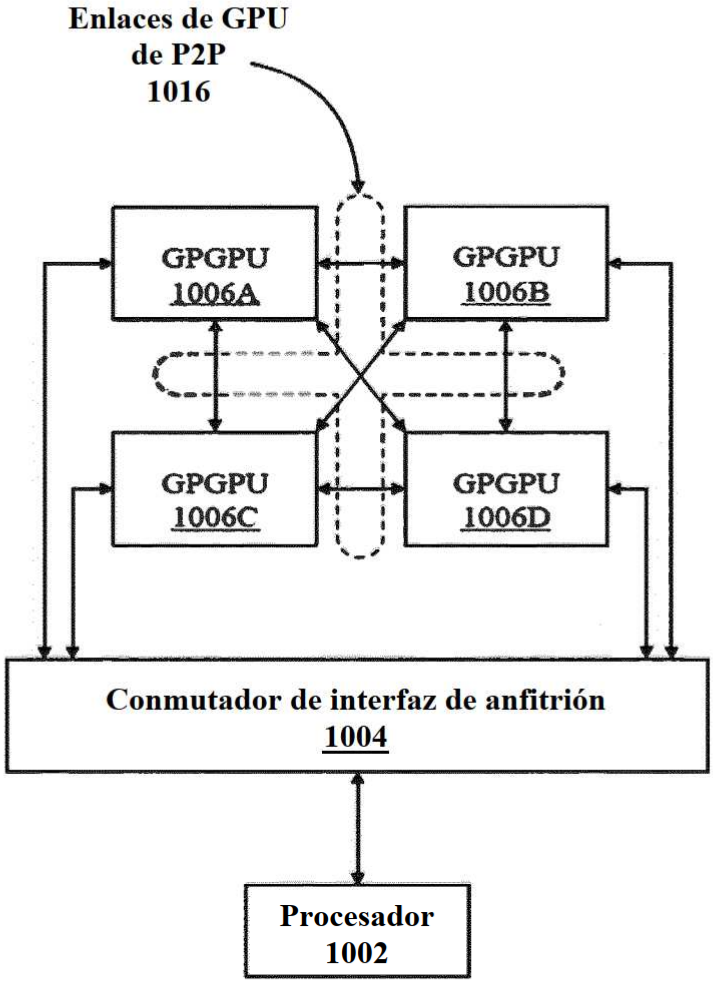

Enlaces de GPU
de P2P
1016
GPGPU
1006A
GPGPU
1006B
GPGPU
1006C
GPGPU
1006D
Conmutador de interfaz de anfitrión
1004
Procesador
1002

FIG. 10

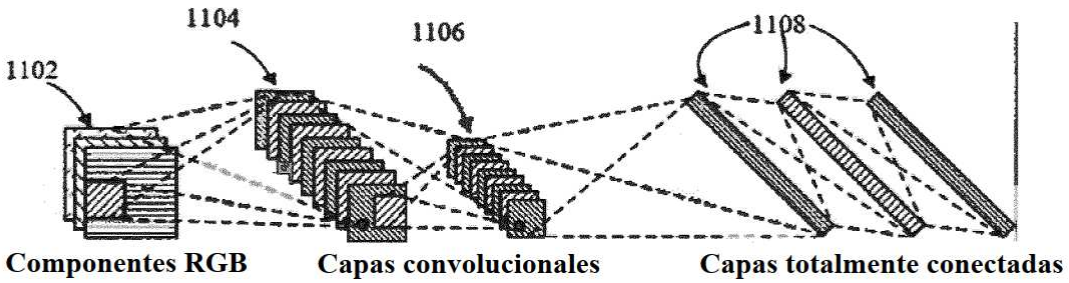


FIG. 11A

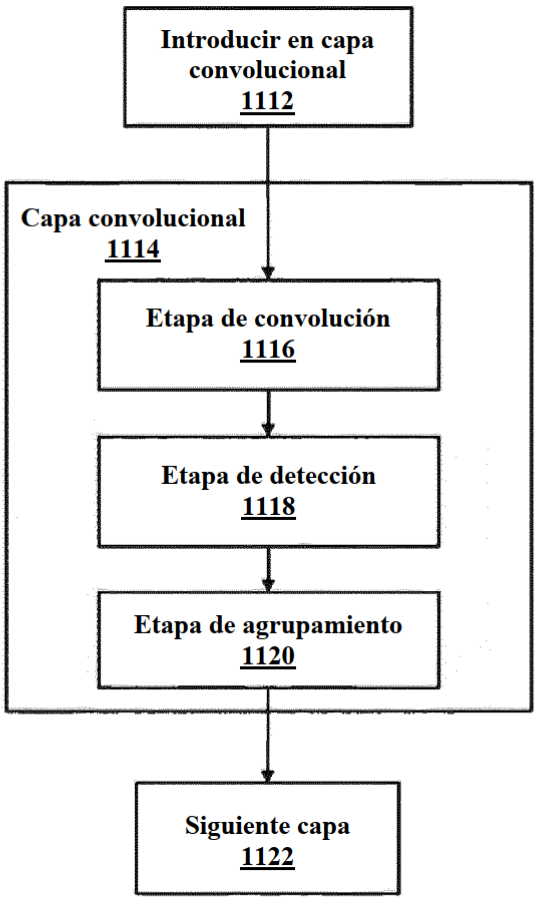


FIG. 11B

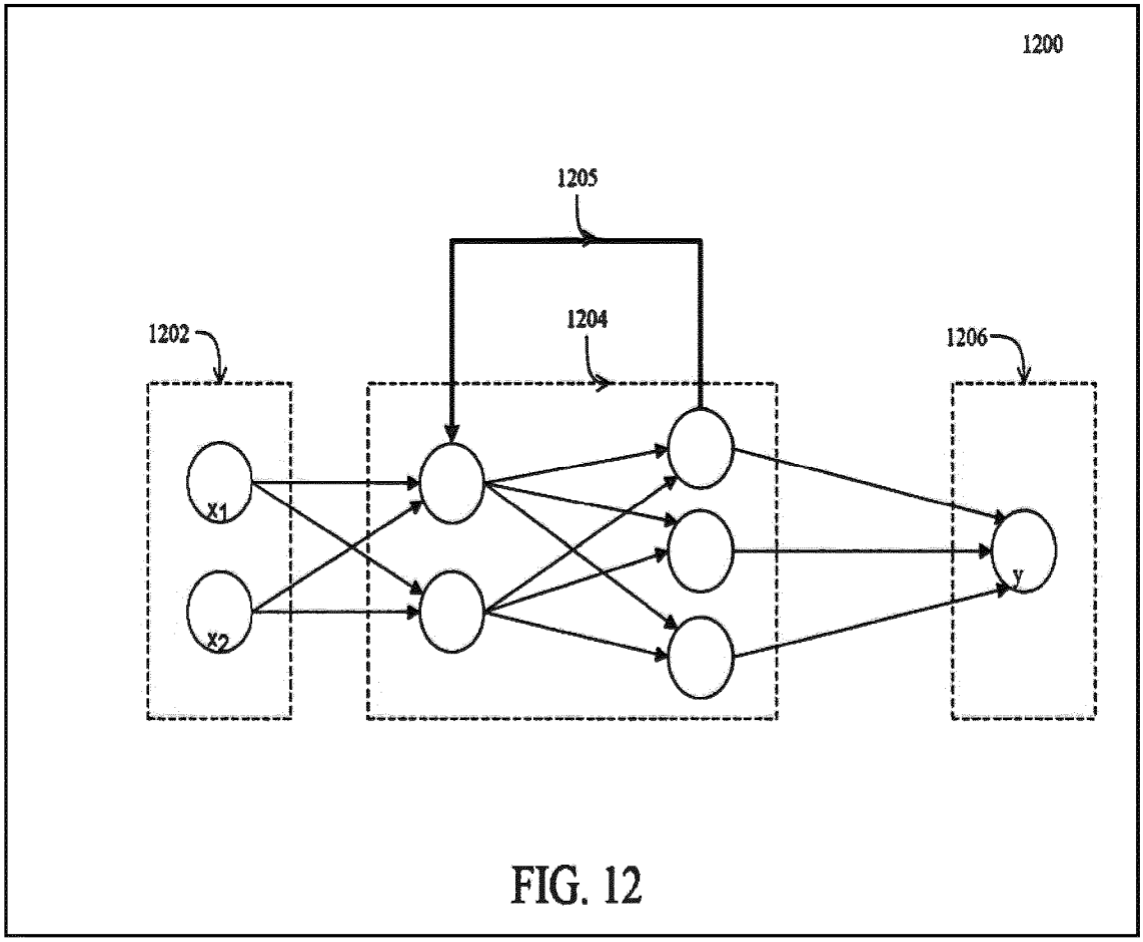


FIG. 12

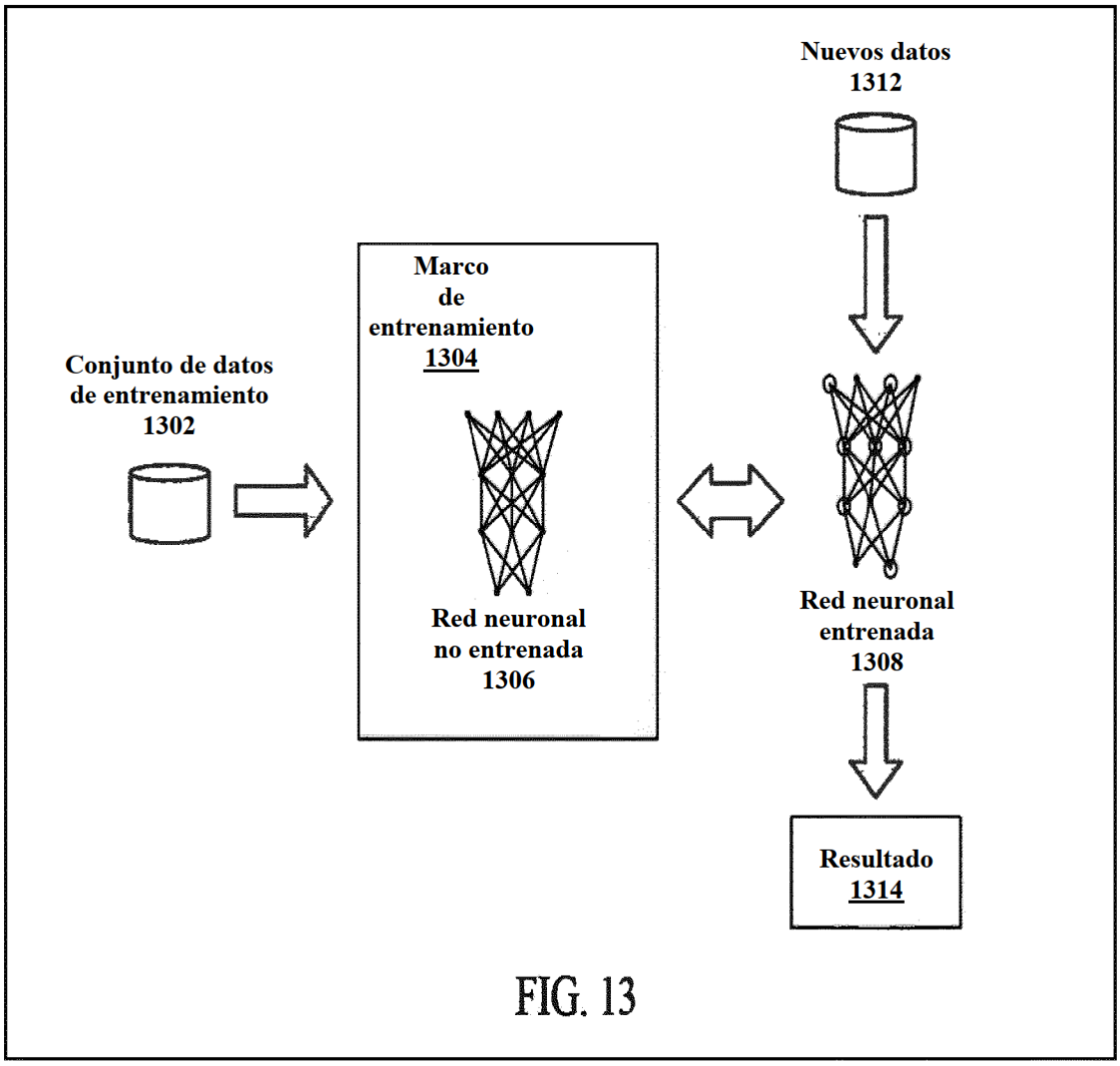

Nuevos datos
1312
Marco
de
entrenamiento
1304
Conjunto de datos
de entrenamiento
1302
Red neuronal
no entrenada
1306
Red neuronal
entrenada
1308
Resultado
1314

FIG. 13

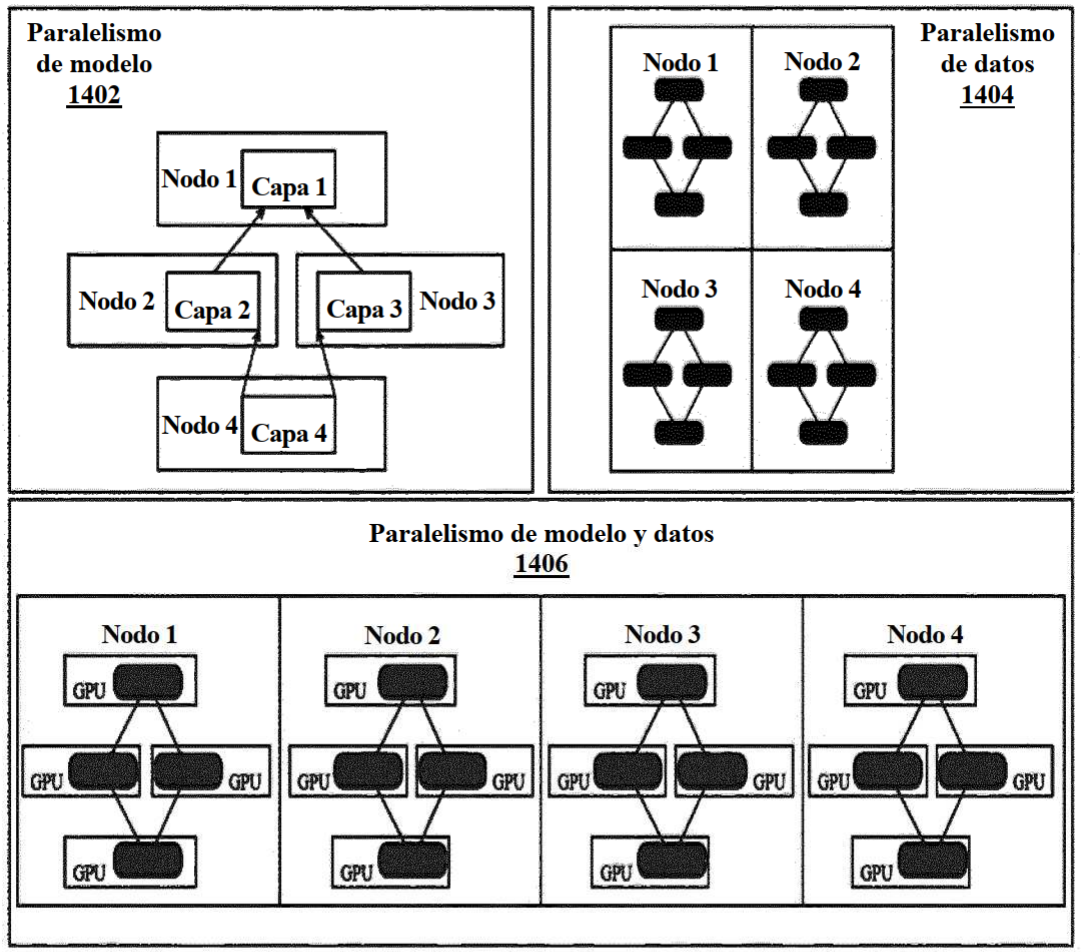

Paralelismo de modelo 1402
Nodo 1
Capa 1
Nodo 2
Capa 2
Capa 3
Nodo 3
Nodo 4
Capa 4
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Paralelismo de datos 1404
Paralelismo de modelo y datos 1406
Nodo 1
Nodo 2
Nodo 3
Nodo 4
GPU

FIG. 14

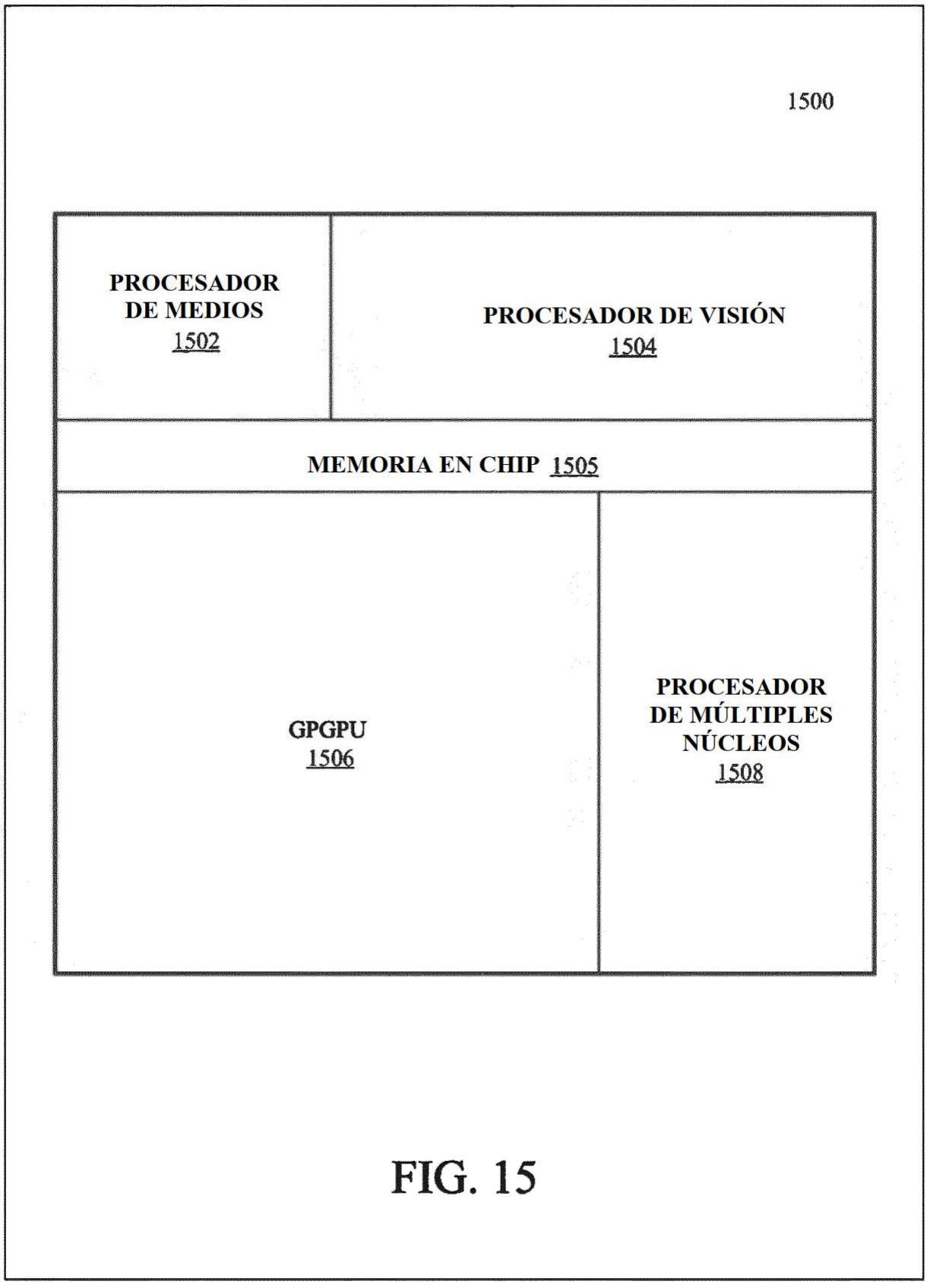

1500
PROCESADOR DE MEDIOS 1502
PROCESADOR DE VISIÓN 1504
MEMORIA EN CHIP 1505
GPGPU 1506
PROCESADOR DE MÚLTIPLES NÚCLEOS 1508

FIG. 15

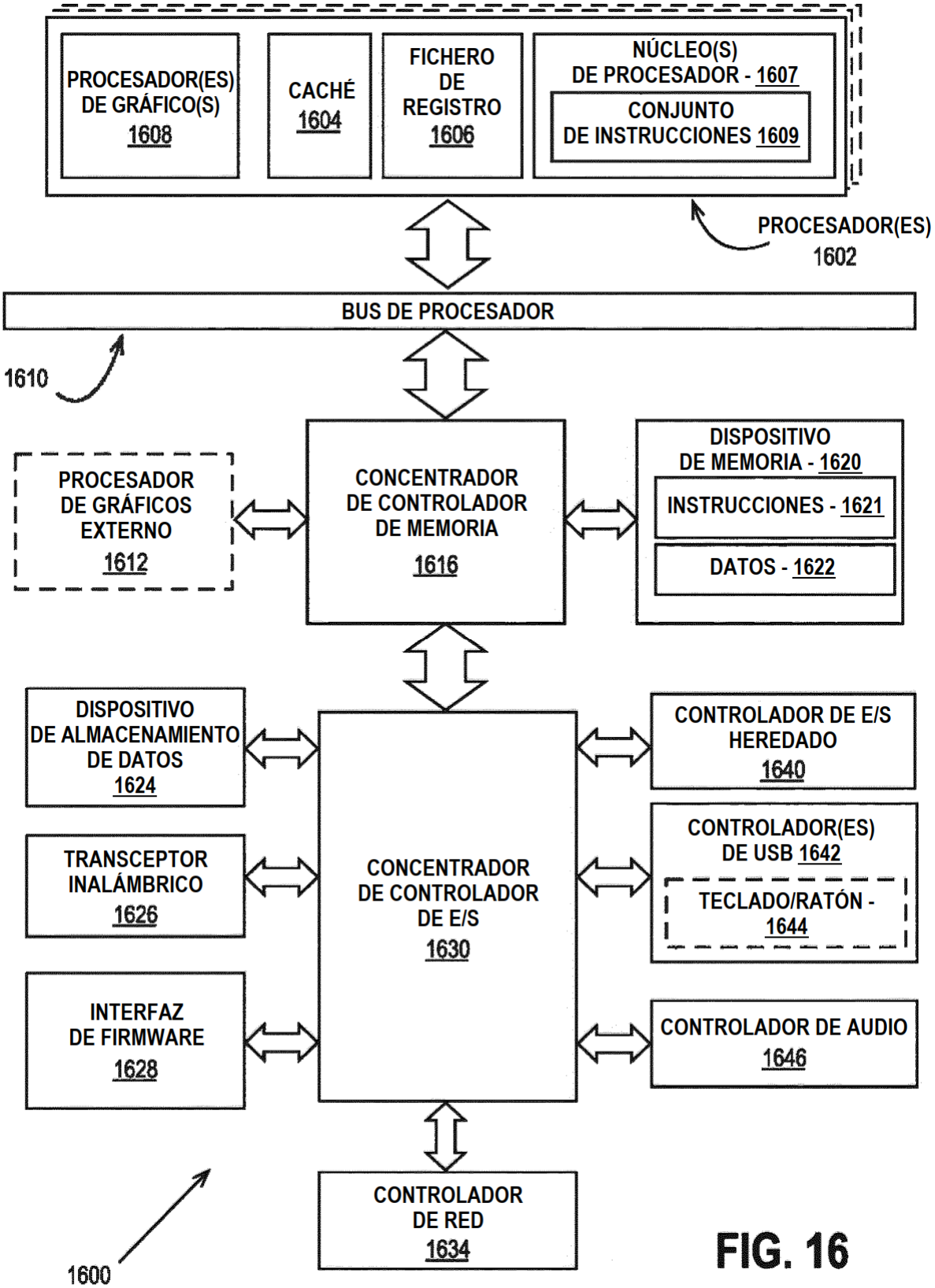

PROCESADOR(ES) DE GRÁFICO(S) 1608
CACHÉ 1604
FICHERO DE REGISTRO 1606
NÚCLEO(S) DE PROCESADOR - 1607
CONJUNTO DE INSTRUCCIONES 1609
PROCESADOR(ES) 1602
BUS DE PROCESADOR
1610
PROCESADOR DE GRÁFICOS EXTERNO 1612
CONCENTRADOR DE CONTROLADOR DE MEMORIA 1616
DISPOSITIVO DE MEMORIA - 1620
INSTRUCCIONES - 1621
DATOS - 1622
DISPOSITIVO DE ALMACENAMIENTO DE DATOS 1624
TRANSCEPTOR INALÁMBRICO 1626
INTERFAZ DE FIRMWARE 1628
CONCENTRADOR DE CONTROLADOR DE E/S 1630
CONTROLADOR DE E/S HEREDADO 1640
CONTROLADOR(ES) DE USB 1642
TECLADO/RATÓN - 1644
CONTROLADOR DE AUDIO 1646
CONTROLADOR DE RED 1634
1600

FIG. 16

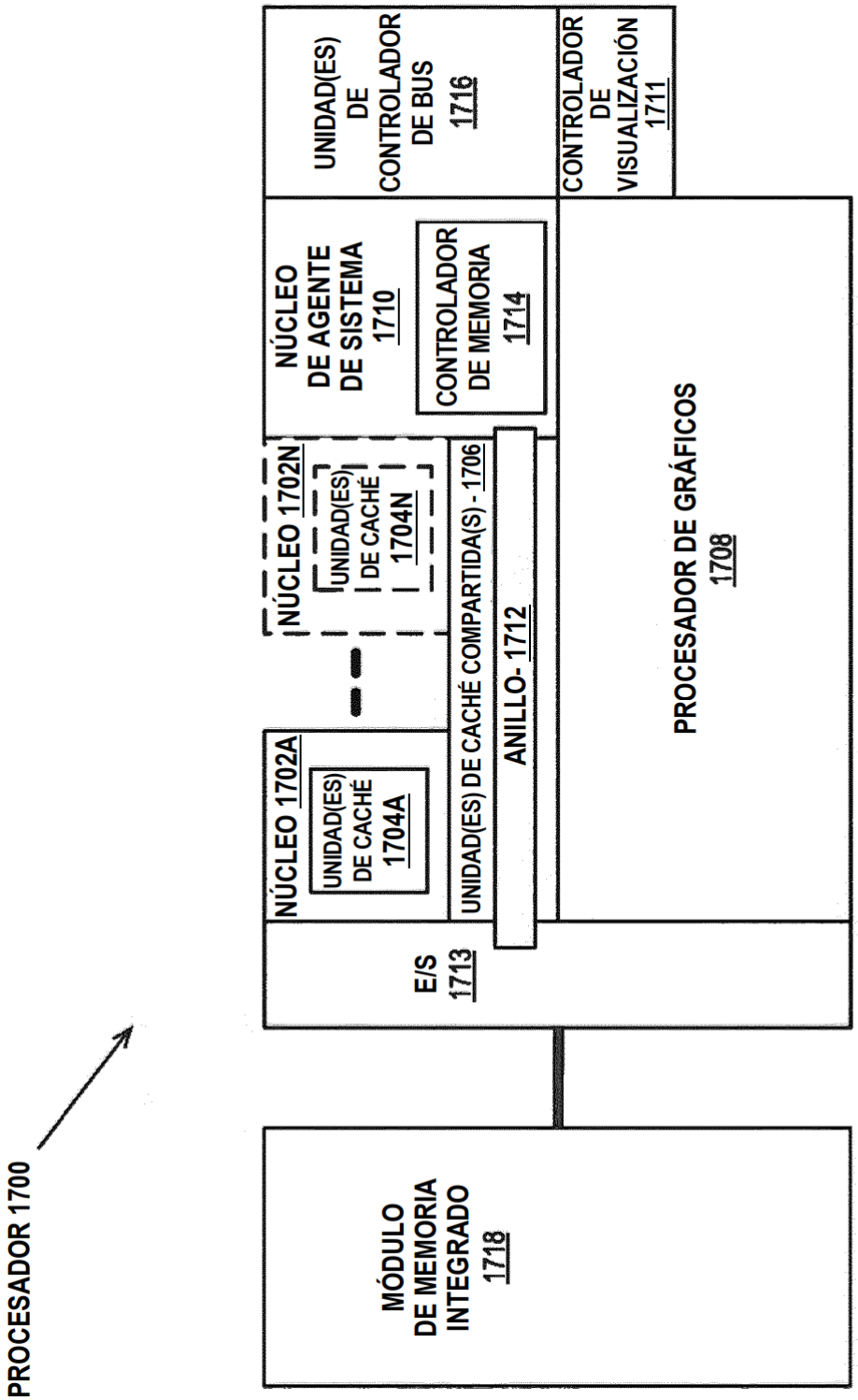

PROCESADOR 1700
MÓDULO DE MEMORIA INTEGRADO 1718
E/S 1713
NÚCLEO 1702A
UNIDAD(ES) DE CACHÉ 1704A
NÚCLEO 1702N
UNIDAD(ES) DE CACHÉ 1704N
UNIDAD(ES) DE CACHÉ COMPARTIDA(S) - 1706
ANILLO- 1712
NÚCLEO DE AGENTE DE SISTEMA 1710
CONTROLADOR DE MEMORIA 1714
UNIDAD(ES) DE CONTROLADOR DE BUS 1716
CONTROLADOR DE VISUALIZACIÓN 1711
PROCESADOR DE GRÁFICOS 1708

FIG. 17

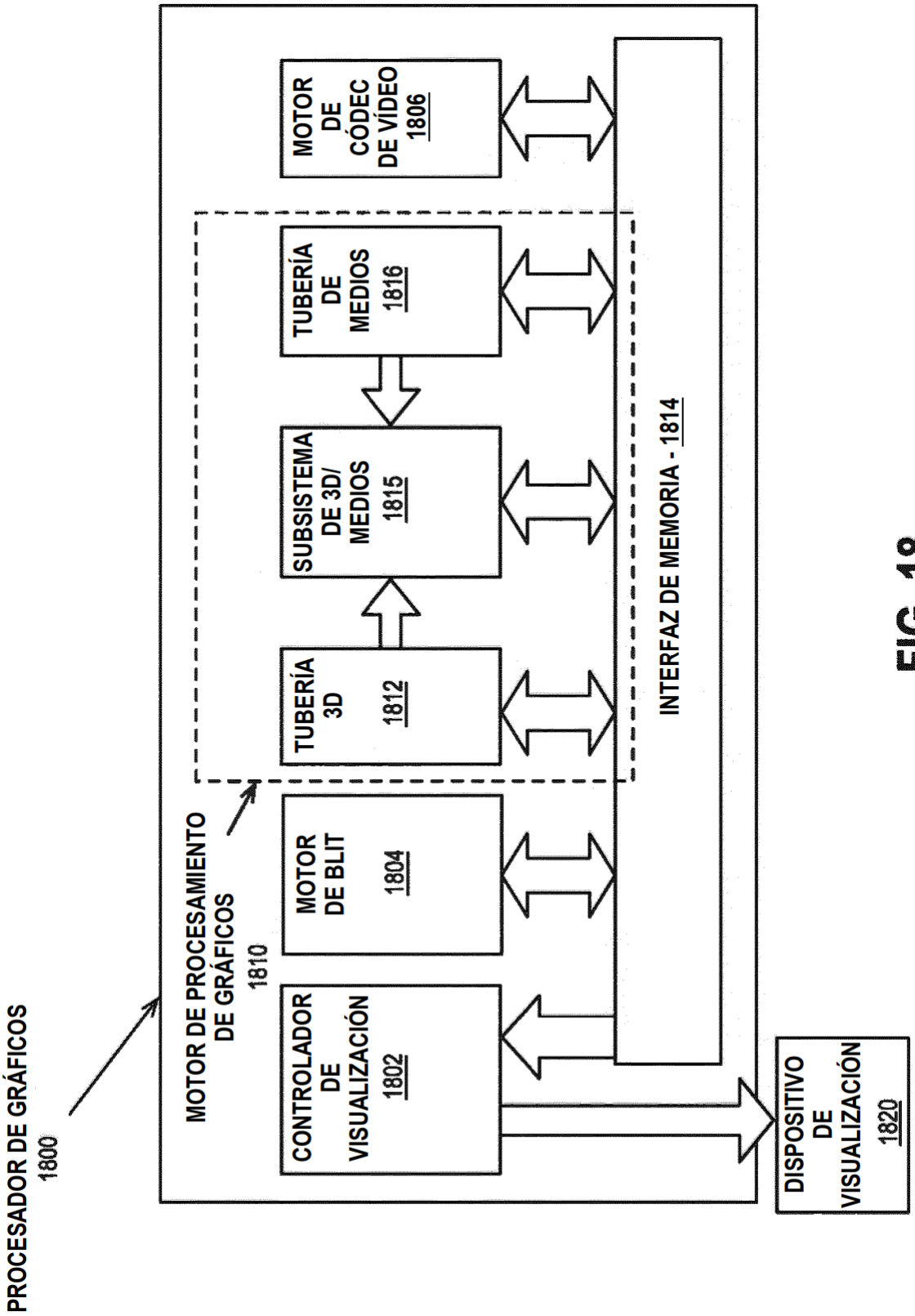

PROCESADOR DE GRÁFICOS
1800
MOTOR DE PROCESAMIENTO DE GRÁFICOS
1810
CONTROLADOR DE VISUALIZACIÓN
1802
MOTOR DE BLIT
1804
TUBERÍA 3D
1812
SUBSISTEMA DE 3D/ MEDIOS
1815
TUBERÍA DE MEDIOS
1816
MOTOR DE CÓDEC DE VÍDEO
1806
INTERFAZ DE MEMORIA - 1814
DISPOSITIVO DE VISUALIZACIÓN
1820

FIG. 18

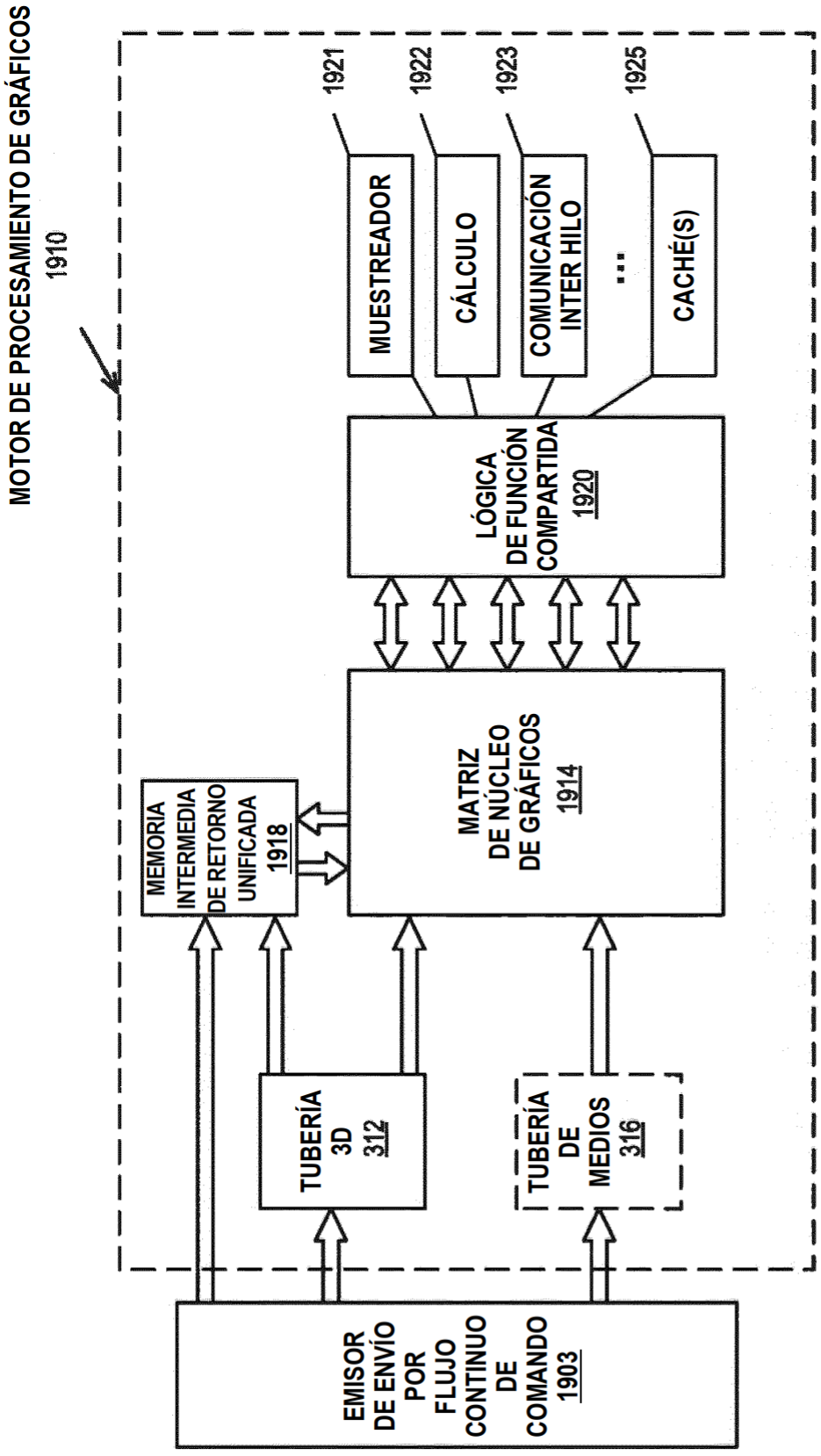

MOTOR DE PROCESAMIENTO DE GRÁFICOS
1910
1921
1922
1923
1925
MUESTREADOR
CÁLCULO
COMUNICACIÓN INTER HILO
...
CACHÉ(S)
LÓGICA DE FUNCIÓN COMPARTIDA 1920
MATRIZ DE NÚCLEO DE GRÁFICOS 1914
MEMORIA INTERMEDIA DE RETORNO UNIFICADA 1918
TUBERÍA 3D 312
TUBERÍA DE MEDIOS 316
EMISOR DE ENVÍO POR FLUJO CONTINUO DE COMANDO 1903

FIG. 19

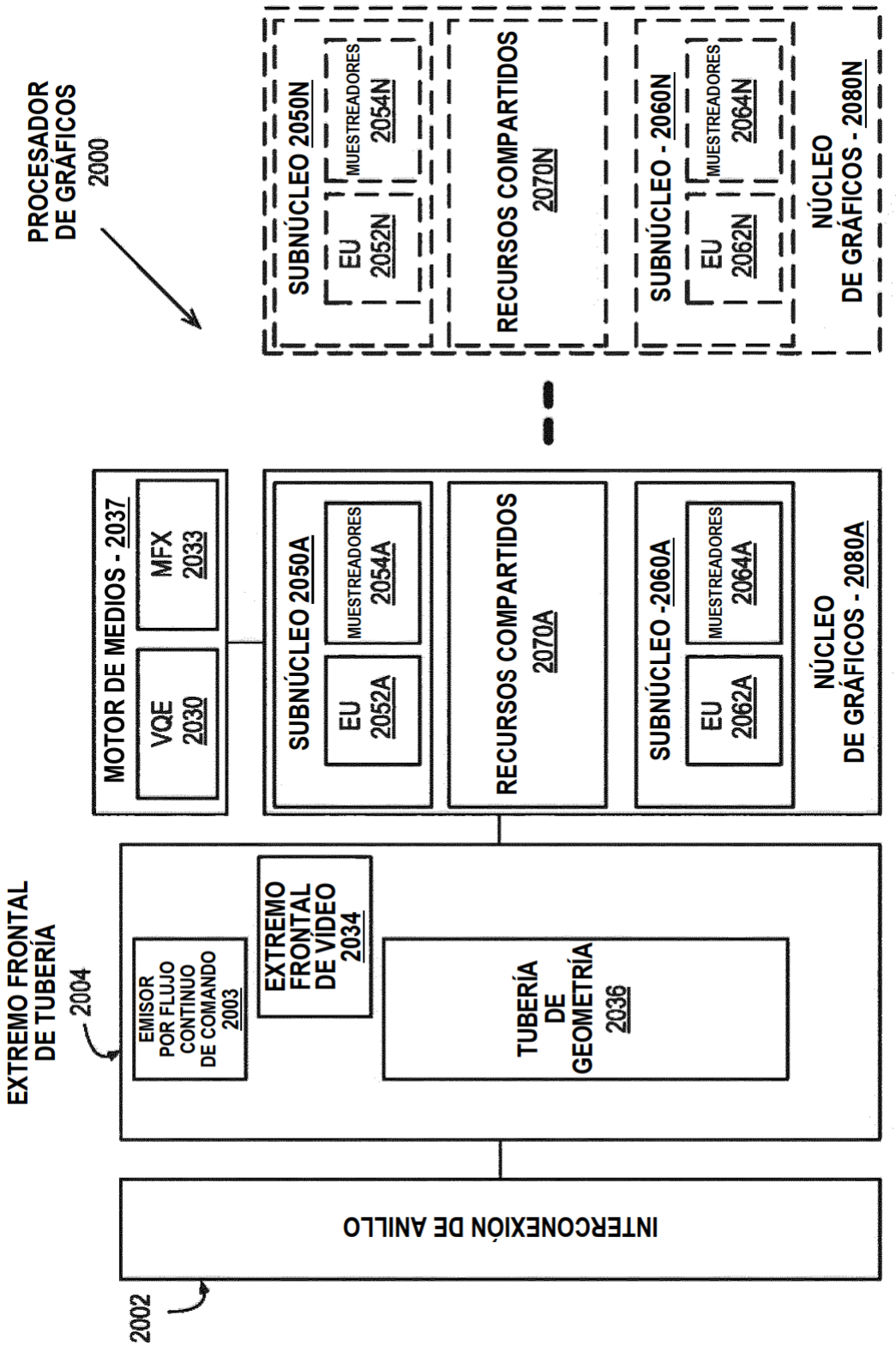

PROCESADOR DE GRÁFICOS 2000
SUBNÚCLEO 2050N
EU 2052N
MUESTREADORES 2054N
RECURSOS COMPARTIDOS 2070N
SUBNÚCLEO - 2060N
EU 2062N
MUESTREADORES 2064N
NÚCLEO DE GRÁFICOS - 2080N
MOTOR DE MEDIOS - 2037
VQE 2030
MFX 2033
SUBNÚCLEO 2050A
EU 2052A
MUESTREADORES 2054A
RECURSOS COMPARTIDOS 2070A
SUBNÚCLEO -2060A
EU 2062A
MUESTREADORES 2064A
NÚCLEO DE GRÁFICOS - 2080A
EXTREMO FRONTAL DE TUBERÍA
2004
EMISOR POR FLUJO CONTINUO DE COMANDO 2003
EXTREMO FRONTAL DE VIDEO 2034
TUBERÍA DE GEOMETRÍA 2036
INTERCONEXIÓN DE ANILLO
2002

FIG. 20

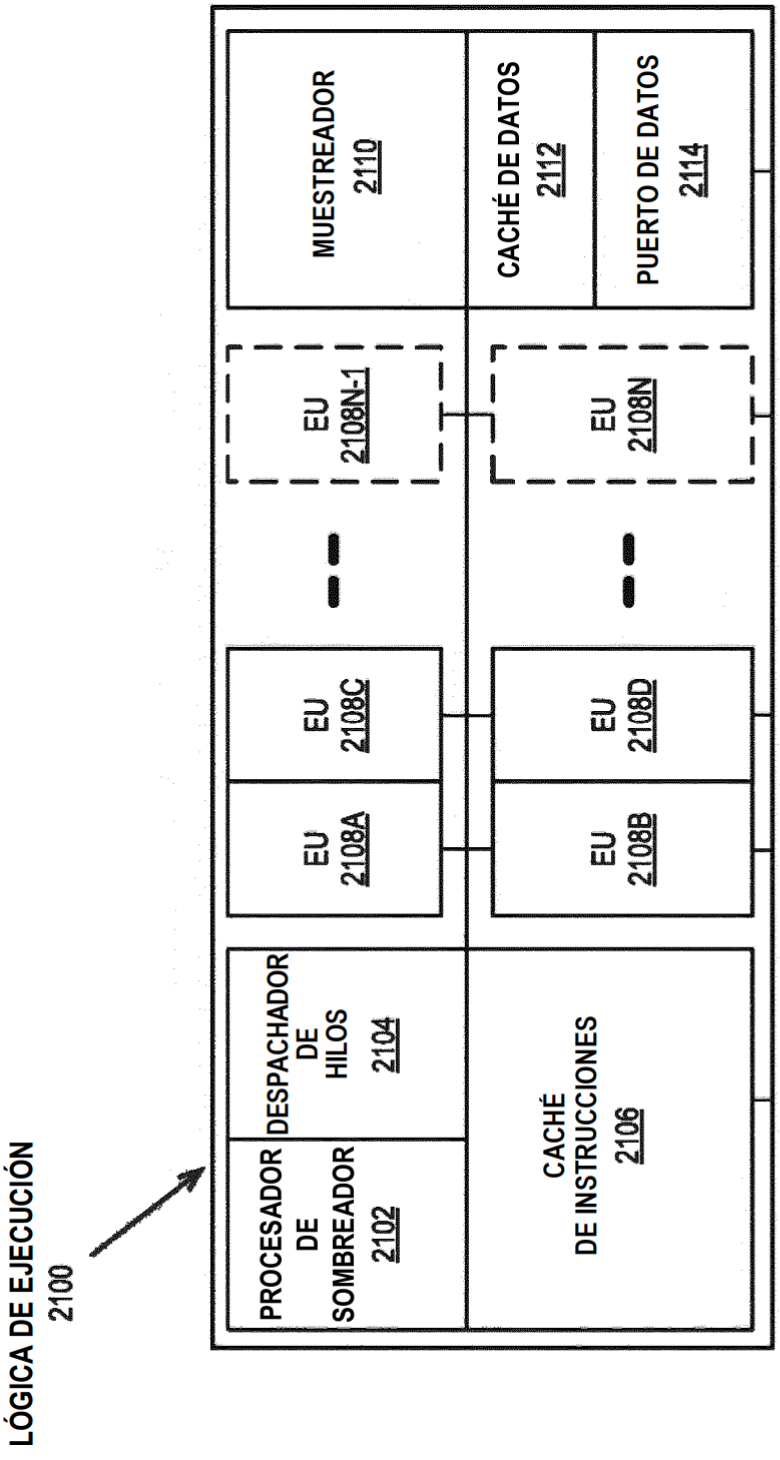

LÓGICA DE EJECUCIÓN
2100
PROCESADOR DE SOMBREADOR
2102
DESPACHADOR DE HILOS
2104
CACHÉ DE INSTRUCCIONES
2106
EU 2108A
EU 2108C
EU 2108B
EU 2108D
EU 2108N-1
EU 2108N
MUESTREADOR
2110
CACHÉ DE DATOS
2112
PUERTO DE DATOS
2114

FIG. 21

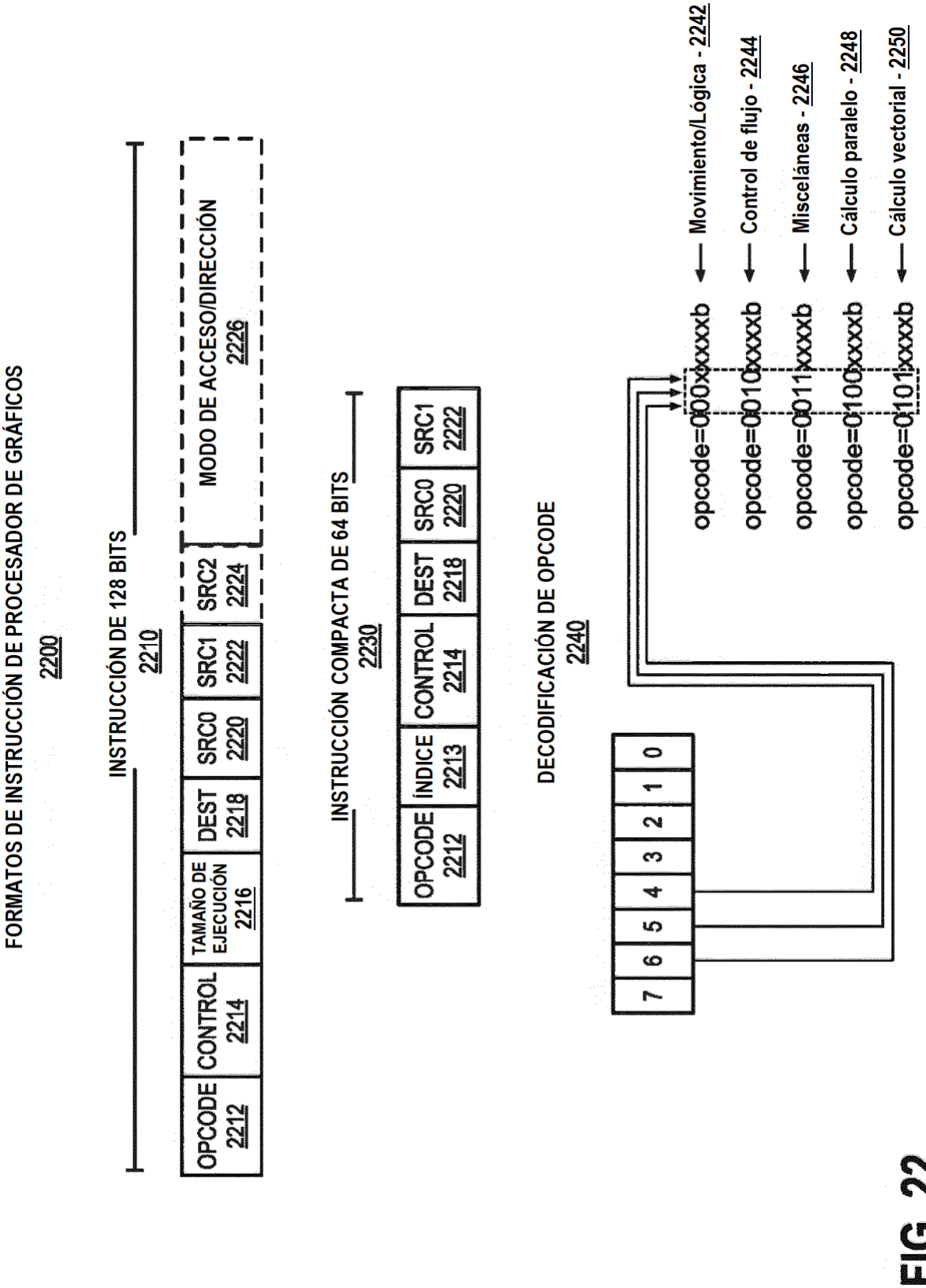

FORMATOS DE INSTRUCCIÓN DE PROCESADOR DE GRÁFICOS
2200
INSTRUCCIÓN DE 128 BITS
2210
OPCODE 2212
CONTROL 2214
TAMAÑO DE EJECUCIÓN 2216
DEST 2218
SRC0 2220
SRC1 2222
SRC2 2224
MODO DE ACCESO/DIRECCIÓN 2226
INSTRUCCIÓN COMPACTA DE 64 BITS
2230
OPCODE 2212
ÍNDICE 2213
CONTROL 2214
DEST 2218
SRC0 2220
SRC1 2222
DECODIFICACIÓN DE OPCODE
2240
7 6 5 4 3 2 1 0
opcode=000xxxxb
opcode=001xxxxb
opcode=0011xxxb
opcode=0100xxxb
opcode=0101xxxb
Movimiento/Lógica - 2242
Control de flujo - 2244
Misceláneas - 2246
Cálculo paralelo - 2248
Cálculo vectorial - 2250

FIG. 22

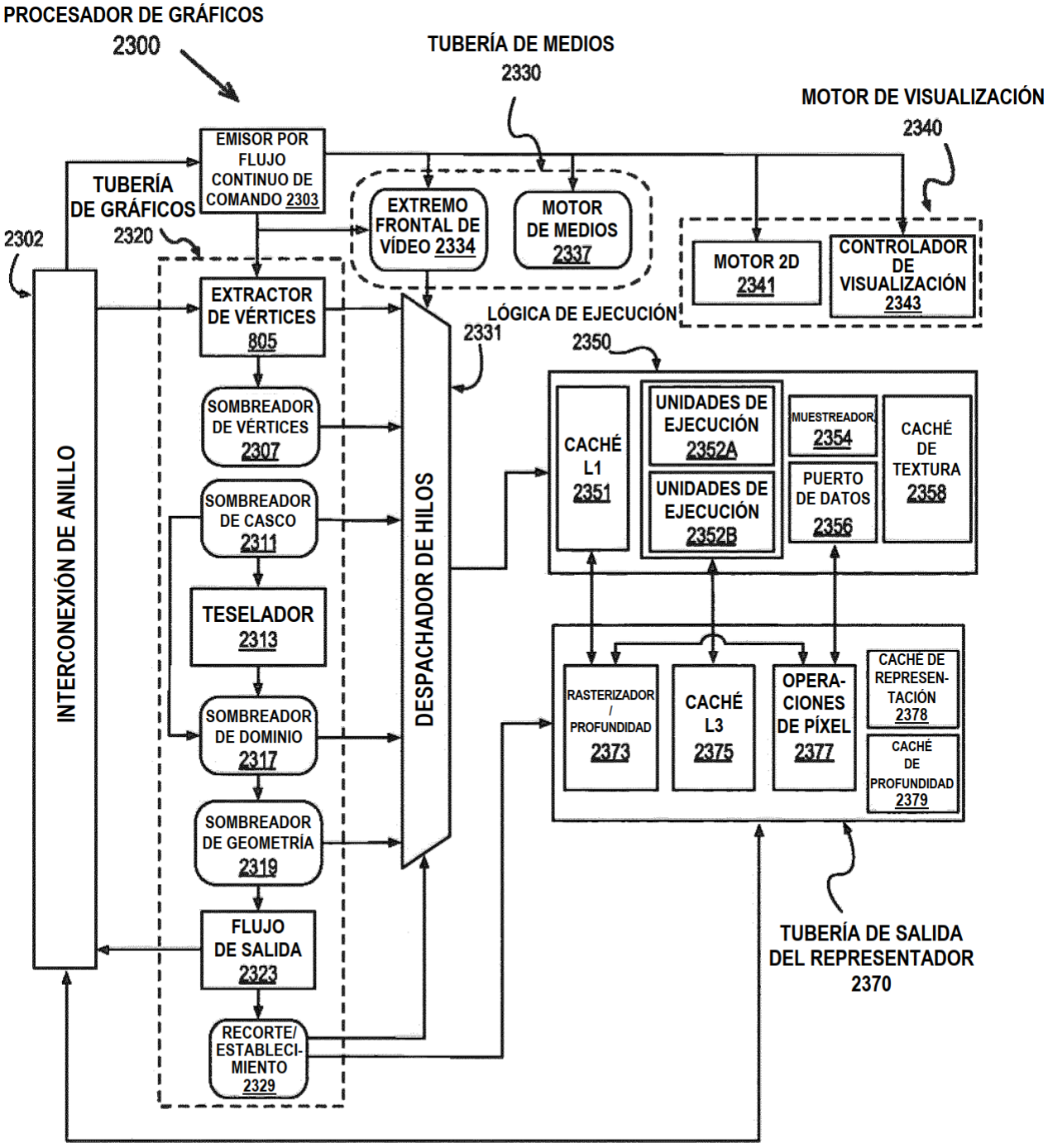

PROCESADOR DE GRÁFICOS
2300
TUBERÍA DE MEDIOS
2330
MOTOR DE VISUALIZACIÓN
2340
EMISOR POR FLUJO CONTINUO DE COMANDO 2303
TUBERÍA DE GRÁFICOS
2320
2302
EXTREMO FRONTAL DE VÍDEO 2334
MOTOR DE MEDIOS 2337
MOTOR 2D 2341
CONTROLADOR DE VISUALIZACIÓN 2343
LÓGICA DE EJECUCIÓN
2331
2350
EXTRACTOR DE VÉRTICES 805
SOMBREADOR DE VÉRTICES 2307
SOMBREADOR DE CASCO 2311
TESELADOR 2313
SOMBREADOR DE DOMINIO 2317
SOMBREADOR DE GEOMETRÍA 2319
FLUJO DE SALIDA 2323
RECORTE/ ESTABLECI-MIENTO 2329
INTERCONEXIÓN DE ANILLO
DESPACHADOR DE HILOS
CACHÉ L1 2351
UNIDADES DE EJECUCIÓN 2352A
UNIDADES DE EJECUCIÓN 2352B
MUESTREADOR 2354
PUERTO DE DATOS 2356
CACHÉ DE TEXTURA 2358
RASTERIZADOR / PROFUNDIDAD 2373
CACHÉ L3 2375
OPERA-CIONES DE PÍXEL 2377
CACHÉ DE REPRESEN-TACIÓN 2378
CACHÉ DE PROFUNDIDAD 2379
TUBERÍA DE SALIDA DEL REPRESENTADOR 2370

FIG. 23

FIG. 24A FORMATO DE COMANDO DE PROCESADOR DE GRÁFICOS

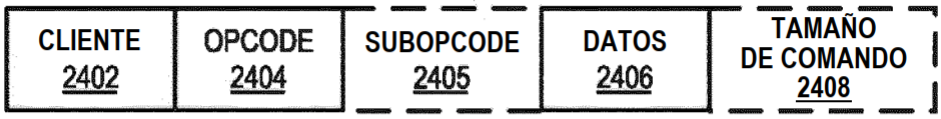
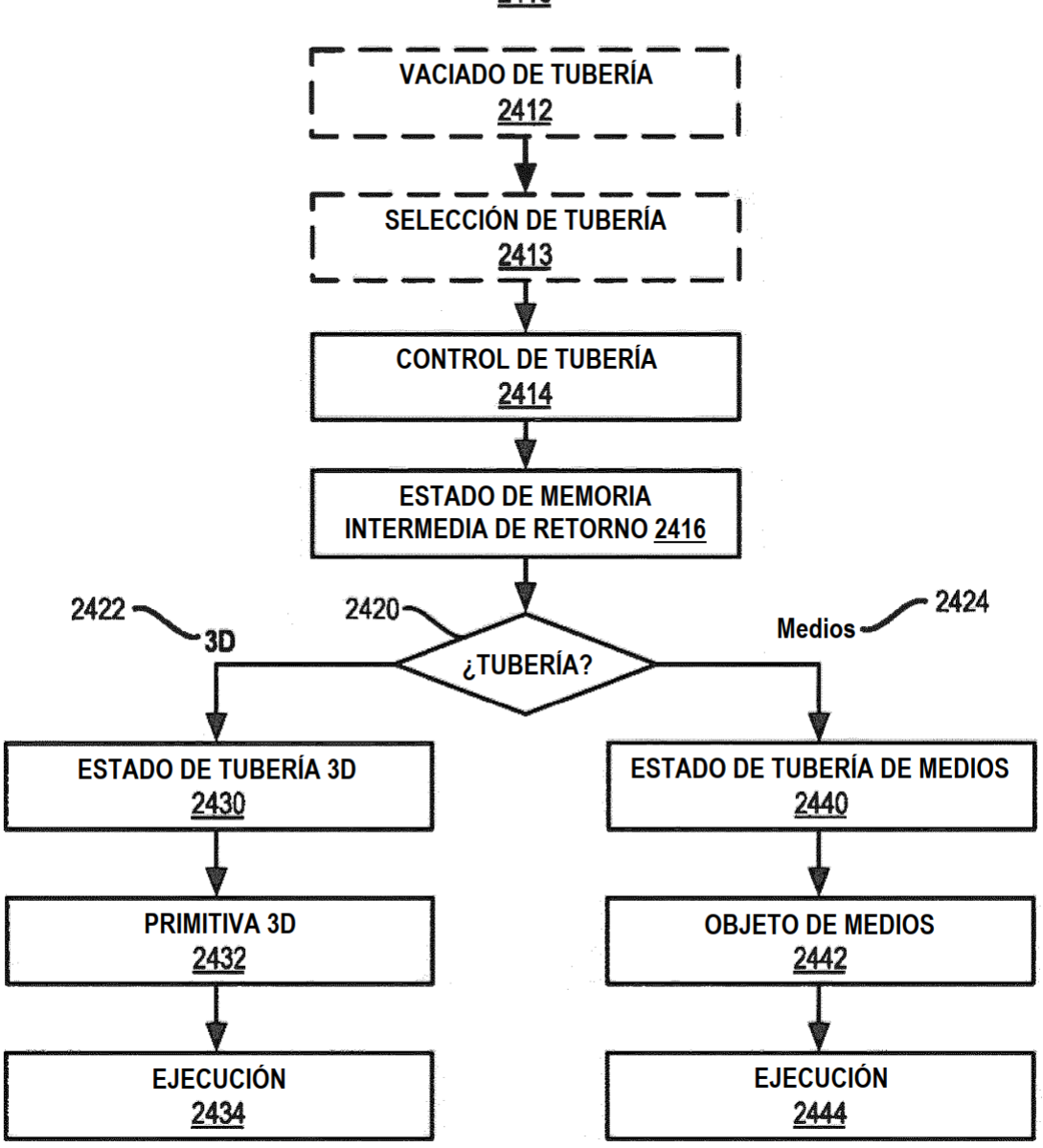


FIG. 24B SECUENCIA DE COMANDOS DE PROCESADOR DE GRÁFICOS



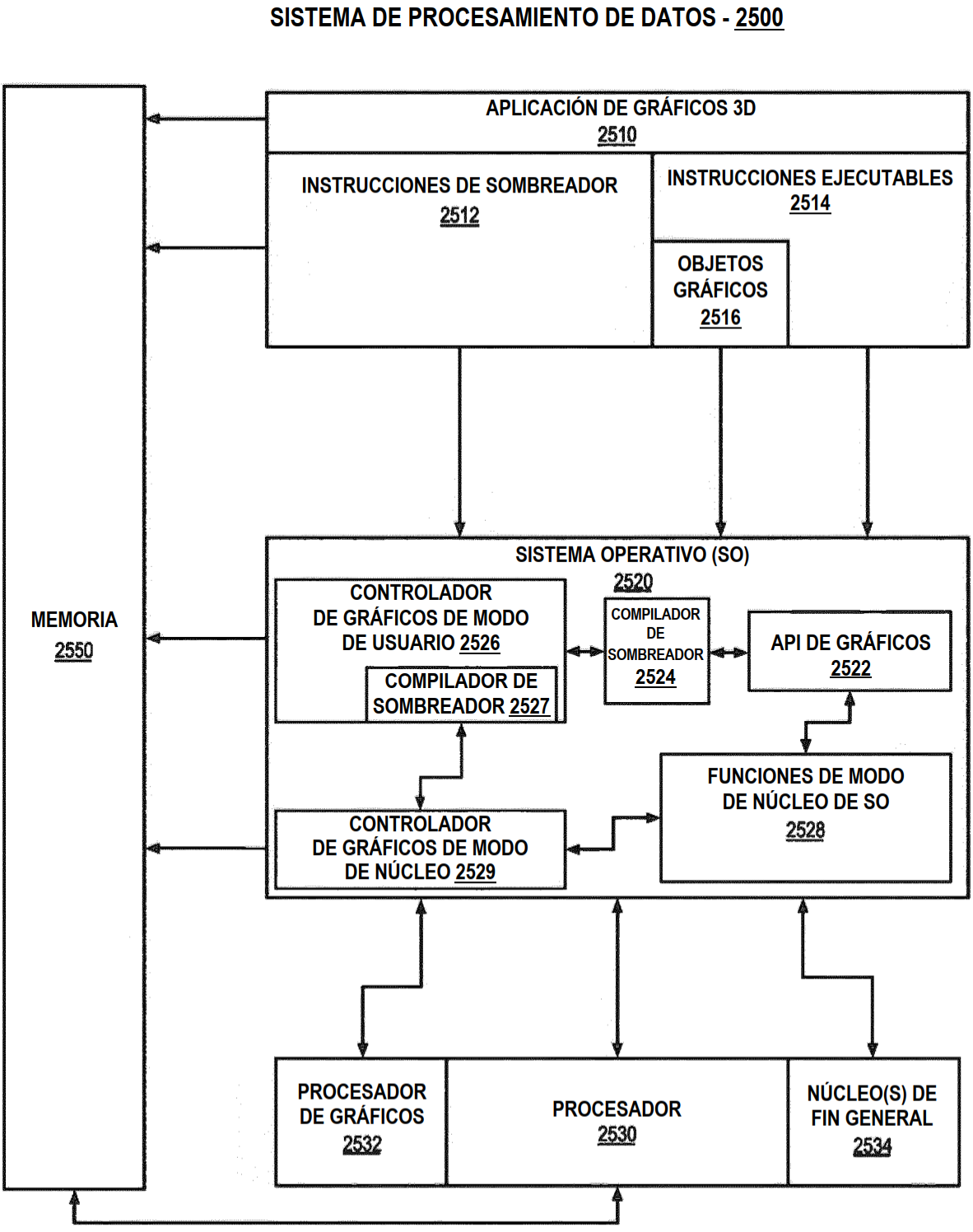

SISTEMA DE PROCESAMIENTO DE DATOS - 2500
APLICACIÓN DE GRÁFICOS 3D
2510
INSTRUCCIONES DE SOMBREADOR
2512
INSTRUCCIONES EJECUTABLES
2514
OBJETOS GRÁFICOS
2516
MEMORIA
2550
SISTEMA OPERATIVO (SO)
2520
CONTROLADOR DE GRÁFICOS DE MODO DE USUARIO 2526
COMPILADOR DE SOMBREADOR 2527
COMPILADOR DE SOMBREADOR
2524
API DE GRÁFICOS
2522
FUNCIONES DE MODO DE NÚCLEO DE SO
2528
CONTROLADOR DE GRÁFICOS DE MODO DE NÚCLEO 2529
PROCESADOR DE GRÁFICOS
2532
PROCESADOR
2530
NÚCLEO(S) DE FIN GENERAL
2534

FIG. 25

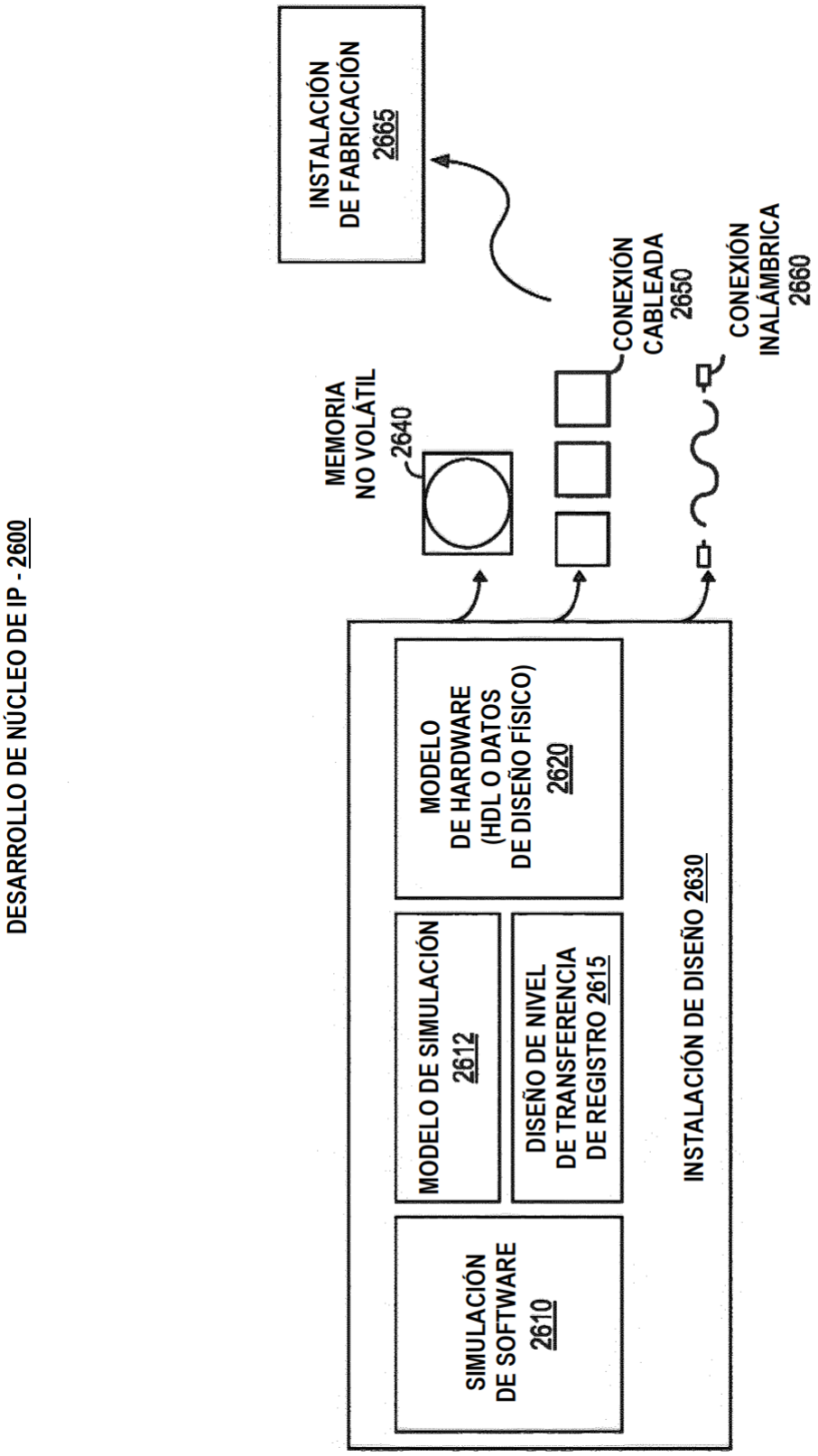

DESARROLLO DE NÚCLEO DE IP - 2600
INSTALACIÓN DE FABRICACIÓN 2665
MEMORIA NO VOLÁTIL 2640
CONEXIÓN CABLEADA 2650
CONEXIÓN INALÁMBRICA 2660
SIMULACIÓN DE SOFTWARE 2610
MODELO DE SIMULACIÓN 2612
DISEÑO DE NIVEL DE TRANSFERENCIA DE REGISTRO 2615
MODELO DE HARDWARE (HDL O DATOS DE DISEÑO FÍSICO) 2620
INSTALACIÓN DE DISEÑO 2630

FIG. 26

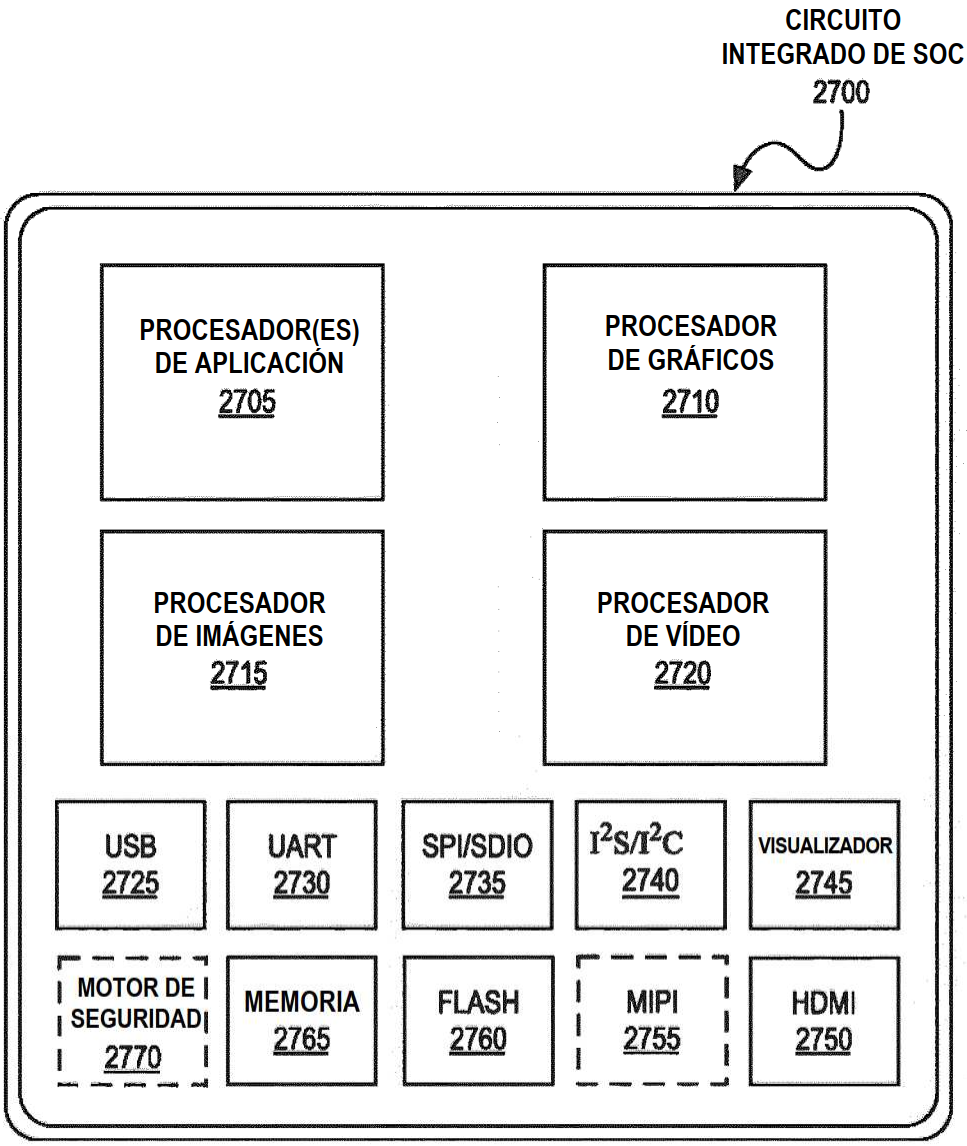

CIRCUITO
INTEGRADO DE SOC
2700
PROCESADOR(ES)
DE APLICACIÓN
2705
PROCESADOR
DE GRÁFICOS
2710
PROCESADOR
DE IMÁGENES
2715
PROCESADOR
DE VÍDEO
2720
USB
2725
UART
2730
SPI/SDIO
2735
I2S/I2C
2740
VISUALIZADOR
2745
MOTOR DE
SEGURIDAD
2770
MEMORIA
2765
FLASH
2760
MIPI
2755
HDMI
2750

FIG. 27

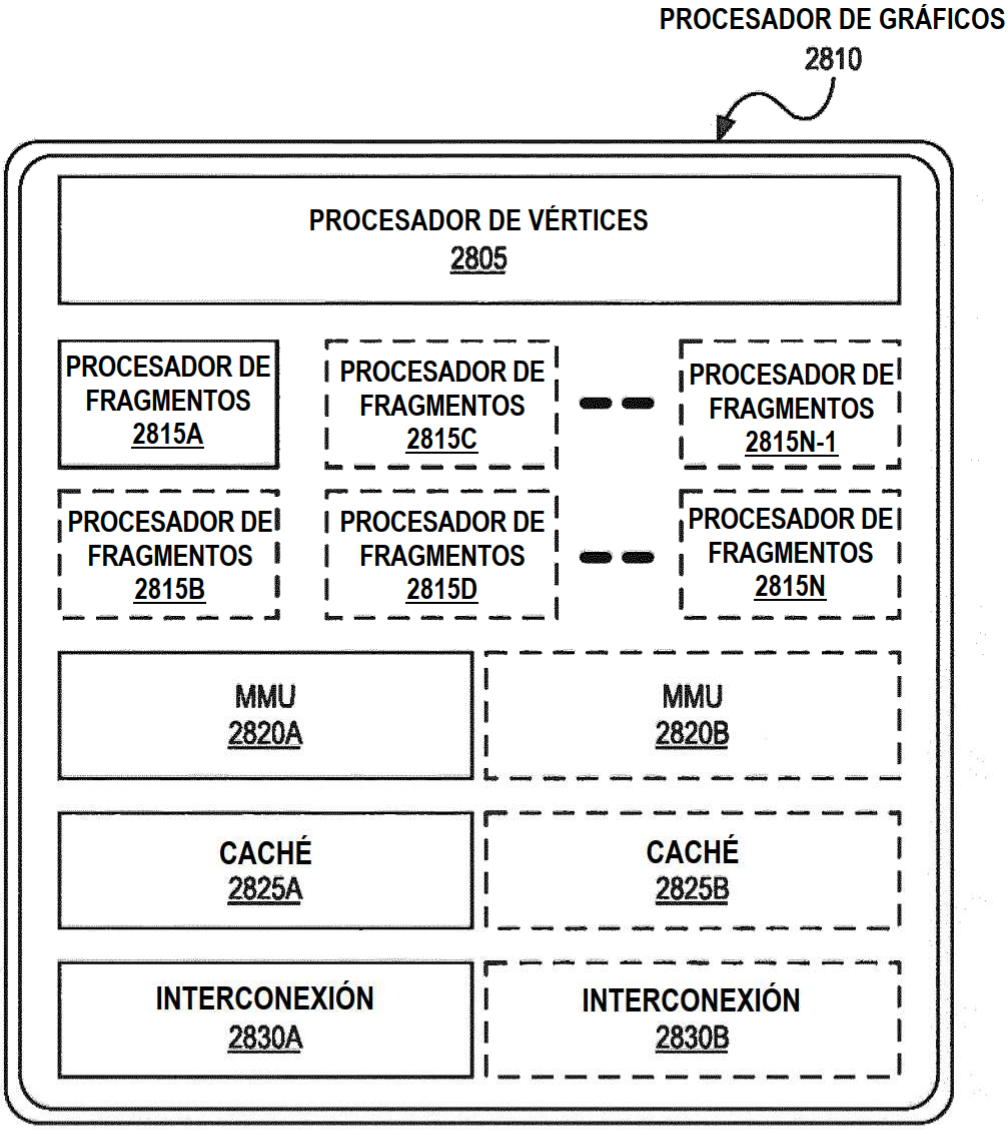

PROCESADOR DE GRÁFICOS
2810
PROCESADOR DE VÉRTICES
2805
PROCESADOR DE FRAGMENTOS
2815A
PROCESADOR DE FRAGMENTOS
2815C
PROCESADOR DE FRAGMENTOS
2815N-1
PROCESADOR DE FRAGMENTOS
2815B
PROCESADOR DE FRAGMENTOS
2815D
PROCESADOR DE FRAGMENTOS
2815N
MMU
2820A
MMU
2820B
CACHÉ
2825A
CACHÉ
2825B
INTERCONEXIÓN
2830A
INTERCONEXIÓN
2830B

FIG. 28

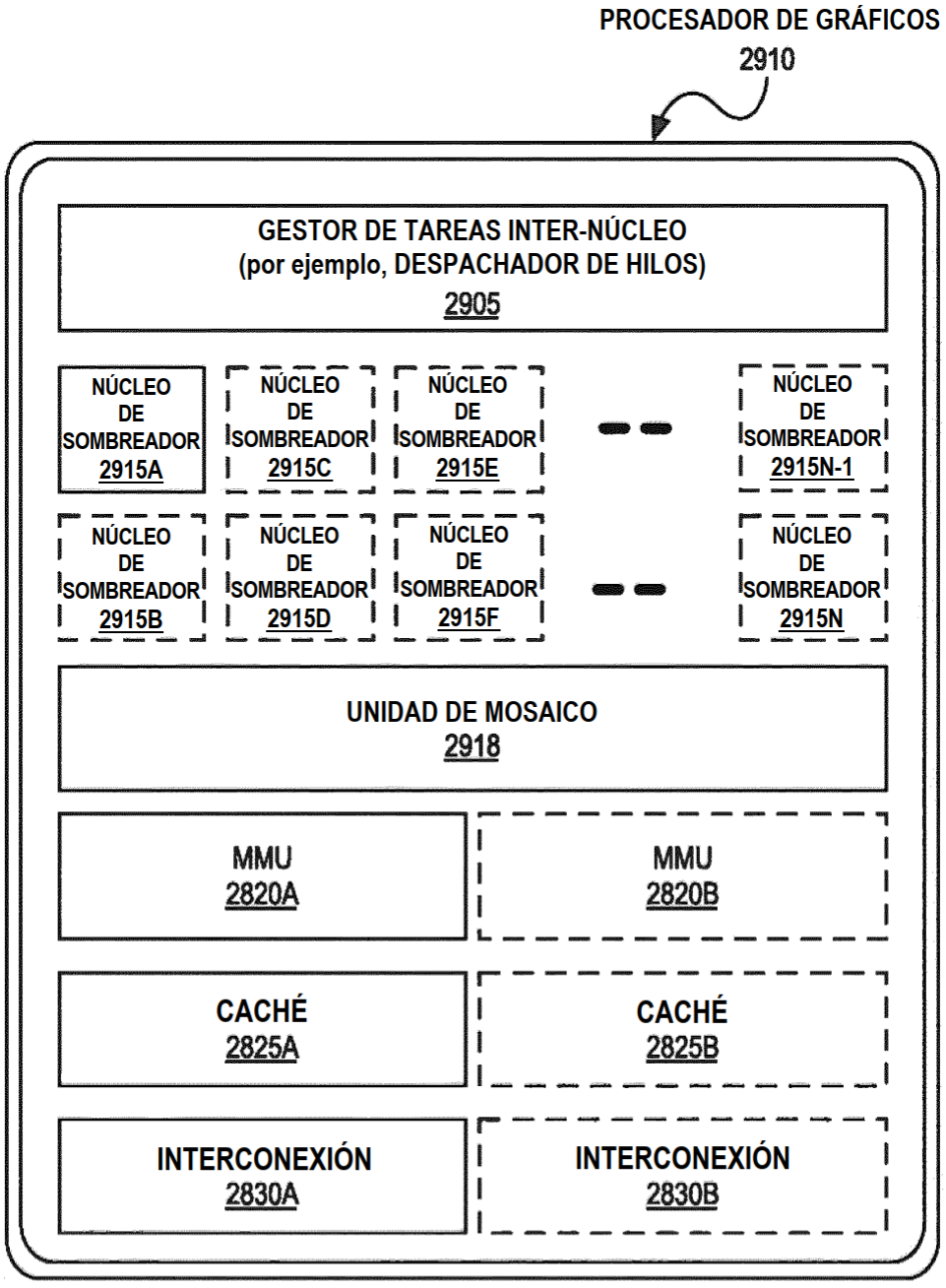

PROCESADOR DE GRÁFICOS
2910
GESTOR DE TAREAS INTER-NÚCLEO
(por ejemplo, DESPACHADOR DE HILOS)
2905
NÚCLEO DE SOMBREADOR 2915A
NÚCLEO DE SOMBREADOR 2915C
NÚCLEO DE SOMBREADOR 2915E
NÚCLEO DE SOMBREADOR 2915N-1
NÚCLEO DE SOMBREADOR 2915B
NÚCLEO DE SOMBREADOR 2915D
NÚCLEO DE SOMBREADOR 2915F
NÚCLEO DE SOMBREADOR 2915N
UNIDAD DE MOSAICO
2918
MMU
2820A
MMU
2820B
CACHÉ
2825A
CACHÉ
2825B
INTERCONEXIÓN
2830A
INTERCONEXIÓN
2830B

FIG. 29