(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 988 624**

(51) Int. CI.:

G06V 40/10 (2012.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.03.2018 E 18163805 (7)**

(97) Fecha y número de publicación de la concesión europea: **24.07.2024 EP 3396591**

(54) Título: **Mejoras de reconocimiento, reidentificación y seguridad usando máquinas autónomas**

(30) Prioridad:

24.04.2017 US 201715495327

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.11.2024

(73) Titular/es:

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

(72) Inventor/es:

DAS, BARNAN;
VARERKAR, MAYURESH M.;
BISWAL, NARAYAN;
BARAN, STANLEY J.;
CILINGIR, GOKCEN;
SHAH, NILESH V.;
SHARMA, ARCHIE;
ABDELHAK, SHERINE;
KOTHA, PRANEETHA;
PANDIT, NEELAY;
WEAST, JOHN C.;
MACPHERSON, MIKE B.;
KIM, DUKHWAN;
HURD, LINDA L.;
APPU, ABHISHEK R.;
KOKER, ALTUG y
RAY, JOYDEEP

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Mejoras de reconocimiento, reidentificación y seguridad usando máquinas autónomas

CAMPO

Las realizaciones descritas en el presente documento se refieren en general al procesamiento de datos y, más particularmente, a facilitar el reconocimiento, la reidentificación y la mejora de seguridad usando máquinas autónomas.

ANTECEDENTES

El procesamiento de datos de gráficos paralelo actual incluye sistemas y métodos desarrollados para realizar operaciones específicas sobre 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, partes de procesadores de gráficos se han hecho programables, lo que posibilita que tales procesadores soporten una gama más amplia de operaciones para procesar datos de vértice y de fragmento.

Para aumentar adicionalmente el rendimiento, los procesadores de gráficos habitualmente implementan técnicas de procesamiento, tales como canalización, que intenta procesar, en paralelo, tantos datos de gráficos como sea posible a lo largo de todas las diferentes partes de la canalización de gráficos. Los procesadores de gráficos paralelos con arquitecturas de múltiples hilos y única instrucción (SIMT) se diseñan para maximizar la cantidad de procesamiento paralelo en la canalización de gráficos. En una arquitectura de SIMT, grupos de hilos paralelos intentan ejecutar conjuntamente instrucciones de programa de manera sincrónica tan a menudo como sea posible para aumentar la eficiencia de procesamiento. Puede encontrarse una vista global general del software y hardware para arquitecturas SIMT en Shane Cook, *CUDA Programming*, capítulo 3, páginas 37-51 (2013) y/o Nicholas Wilt, *CUDA Handbook, A Comprehensive Guide to GPU Programming*, secciones 2.6.2 a 3.1.2 (junio de 2013).

El aprendizaje automático ha tenido éxito en la resolución de muchos tipos de tareas. Los cálculos que surgen cuando se entrenan y se usan algoritmos de aprendizaje automático (por ejemplo, redes neuronales) se prestan naturalmente a implementaciones paralelas eficientes. En consecuencia, los procesadores paralelos, tales como las unidades de procesamiento gráfico de propósito general (GPGPU), han desempeñado un papel importante en la implementación práctica de las redes neuronales profundas. Los procesadores de gráficos paralelos con arquitecturas de múltiples hilos y única instrucción (SIMT) se diseñan para maximizar la cantidad de procesamiento paralelo en la canalización de gráficos. En una arquitectura de SIMT, grupos de hilos paralelos intentan ejecutar conjuntamente instrucciones de programa de manera sincrónica tan a menudo como sea posible para aumentar la eficiencia de procesamiento. La eficiencia proporcionada por las implementaciones de algoritmos de aprendizaje automático paralelo posibilita el uso de redes de alta capacidad y posibilita que esas redes se entrenen en conjuntos de datos más grandes.

Las técnicas convencionales están severamente limitadas con respecto al reconocimiento de seres humanos, tal como los sistemas basados en visión convencionales se basan principalmente en características faciales, conocidas popularmente como reconocimiento facial. En muchas aplicaciones convencionales del mundo real, los perfiles faciales de individuos a menudo no están disponibles.

Dong Seon Cheng y col., "*Custom Pictorial Structures for Re-identification*" se refiere a una metodología para la reidentificación basándose en estructuras pictóricas.

Dat Nguyen y col., "*Person Recognition System Based on a Combination of Body Images from Visible Light and Thermal Cameras*" se refiere a un método de reconocimiento de personas que usa información extraída de imágenes corporales.

Meltem Demirkus y col., "*Automated person categorization for video surveillance using soft biometrics*" se refiere a un sistema de rastreo de vídeo y categorización de personas que usa características biométricas suaves de cara y persona para etiquetar personas mientras las rastrea en múltiples vistas de cámara.

Vaquero y col., "*Attribute-Based People Search in Surveillance Environments*" se refiere a una estructura para buscar personas en entornos de vigilancia.

La invención se define por un aparato, un método y al menos un medio legible por máquina de acuerdo con las reivindicaciones independientes. Se definen realizaciones preferidas en las reivindicaciones dependientes.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Las realizaciones se ilustran a modo de ejemplo, y no a modo de limitación, en las figuras de los dibujos adjuntos en los que números de referencia similares se refieren a elementos similares. Para que puedan entenderse en detalle las características antes citadas, se proporciona una descripción más particular, resumida anteriormente de manera breve,

haciendo referencia a las realizaciones, algunas de las cuales se ilustran en los dibujos adjuntos. Sin embargo, cabe señalar que los dibujos adjuntos ilustran únicamente realizaciones típicas y, por lo tanto, no deben considerarse limitativos de su alcance, ya que los dibujos pueden ilustrar otras realizaciones igualmente eficaces.

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

Las **Figuras 2A-2D** ilustran componentes de procesador paralelo, de acuerdo con una realización.

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 unidades de procesamiento de gráficos están acopladas de manera comunicativa a una pluralidad de procesadores de múltiples núcleos.

La **Figura 5** es un diagrama conceptual de una canalización de procesamiento de gráficos, de acuerdo con una realización.

La **Figura 6** ilustra un dispositivo informático que aloja un mecanismo de reconocimiento y seguridad de acuerdo con una realización.

La **Figura 7** ilustra un mecanismo de reconocimiento y seguridad de acuerdo con una realización.

La **Figura 8A** ilustra una secuencia de transacciones para el reconocimiento de personas de acuerdo con una realización.

La **Figura 8B** ilustra un método para la reidentificación de personas de acuerdo con una realización de acuerdo con una realización.

La **Figura 8C** ilustra una secuencia de transacciones para la puesta en correspondencia de personas de acuerdo con una realización de acuerdo con una realización.

La **Figura 9A** ilustra un modelo para comprobaciones de autenticación y verificación para redes neuronales en el aprendizaje automático de acuerdo con una realización.

La **Figura 9B** ilustra una estructura para la ejecución paralela de redes neuronales en el aprendizaje automático de acuerdo con una realización.

La **Figura 9C** ilustra una secuencia de transacciones para usar una salida de red neuronal para alterar o confirmar una decisión pendiente en máquinas autónomas de acuerdo con una realización.

La **Figura 10** ilustra una pila de software de aprendizaje automático, de acuerdo con una realización.

La **Figura 11** ilustra una unidad de procesamiento de gráficos de propósito general altamente paralela, de acuerdo con una realización.

La **Figura 12** ilustra un sistema informático de múltiples GPU, de acuerdo con una realización.

Las **Figuras 13A-13B** ilustran capas de redes neuronales profundas ilustrativas.

La **Figura 14** ilustra el entrenamiento y despliegue de una red neuronal profunda.

La **Figura 15** ilustra el entrenamiento y despliegue de una red neuronal profunda

La **Figura 16 es** un diagrama de bloques que ilustra un aprendizaje distribuido.

La **Figura 17** ilustra un sistema de inferencia ilustrativo en un chip (SOC) adecuado para realizar inferencias utilizando un modelo entrenado.

La **Figura 18** es un diagrama de bloques de una realización de un sistema informático con un procesador que tiene uno o más núcleos de procesador y procesadores de gráficos.

La **Figura 19** es un diagrama de bloques de una realización de un procesador que tiene uno o más núcleos de procesador, un controlador de memoria integrado y un procesador de gráficos integrado.

La **Figura 20** es un diagrama de bloques de una realización de un procesador de gráficos, 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.

La **Figura 21** es un diagrama de bloques de una realización de un motor de procesamiento de gráficos de un procesador de gráficos.

La **Figura 22** es un diagrama de bloques de otra realización de un procesador de gráficos.

La **Figura 23** es un diagrama de bloques de la lógica de ejecución de hilos que incluye una matriz de elementos de procesamiento.

La **Figura 24** ilustra un formato de instrucción de unidad de ejecución de procesador de gráficos de acuerdo con una realización.

La **Figura 25** es un diagrama de bloques de otra realización de un procesador de gráficos que incluye una canalización de gráficos, una canalización de medios, un motor de visualización, una lógica de ejecución de hilos y una canalización de salida de representación.

La **Figura 26A** es un diagrama de bloques que ilustra un formato de comando de procesador de gráficos de acuerdo con una forma de realización.

La **Figura 26B** es un diagrama de bloques que ilustra una secuencia de comandos de procesador de gráficos de acuerdo con una realización.

La **Figura 27** ilustra una arquitectura de software de gráficos ilustrativa para un sistema de procesamiento de datos de acuerdo con una realización.

La **Figura 28** es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP que puede usarse para fabricar un circuito integrado para realizar las operaciones de acuerdo con una realización.

La **Figura 29** es un diagrama de bloques que ilustra un circuito integrado de sistema en un chip ilustrativo que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización.

La **Figura 30** es un diagrama de bloques que ilustra un procesador de gráficos ilustrativo de un sistema en circuito integrado de chip.

La **Figura 31** es un diagrama de bloques que ilustra un procesador de gráficos ilustrativo adicional de un sistema en un circuito integrado de chip.

DESCRIPCIÓN DETALLADA

Las realizaciones prevén una técnica novedosa para reconocer individuos (también denominados "gente", "personas", "seres humanos", "usuarios") a partir de características visuales de sus cuerpos usando un mecanismo de visión informática y aprendizaje profundo. Por ejemplo, en una realización, las realizaciones prevén: 1) el reconocimiento de personas más allá de la cara, tal como usando solo su cuerpo; 2) el reconocimiento en presencia de amplias variantes de postura humana y ocultación corporal; y 3) el uso de hardware de cámara especializado, tal como cámaras infrarrojas (IR) o térmicas, etc.

Las realizaciones prevén adicionalmente la autenticación basada en capas y la comprobación de integridad, la ejecución paralela de redes neuronales con aislamiento de ejecución y la detección de anomalías de sistema usando una comparación de salida de red neuronal.

Cabe señalar que, términos o acrónimos como "red neuronal convolucional", "CNN", "red neuronal", "NN", "red neuronal profunda", "DNN", "red neuronal recurrente", "RNN" y /o similares pueden ser referenciados de manera intercambiable a lo largo de todo este documento. Además, expresiones como "máquina autónoma" o simplemente "máquina", "vehículo autónomo" o simplemente "vehículo", "agente autónomo" o simplemente "agente", "dispositivo autónomo" o "dispositivo informático", "robot", y/o similares, se pueden hacer referencia de manera intercambiable a lo largo de todo este documento.

En algunas realizaciones, una unidad de procesamiento de gráficos (GPU) está acoplada de manera comunicativa a núcleos de anfitrión/de procesador para acelerar las operaciones de gráficos, las operaciones de aprendizaje automático, las operaciones de análisis de patrones y diversas funciones de GPU de propósito general (GPGPU). La GPU puede acoplarse de manera comunicativa al procesador/núcleos de anfitrión a través de un bus u otra interconexión (por ejemplo, una interconexión de alta velocidad tal como PCIe o NVLink). En otras realizaciones, la GPU puede integrarse en el mismo paquete o chip que los núcleos y estar acoplada de manera comunicativa a los

núcleos a través de un bus/interconexión de procesador interno (es decir, internamente 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 usa entonces circuitería/lógica dedicada para procesar de manera eficiente estos comandos/instrucciones.

En la siguiente descripción, se exponen numerosos detalles específicos. Sin embargo, las realizaciones, como se describe en el presente documento, pueden ponerse en práctica sin estos detalles específicos. En otros casos, no se han mostrado en detalle circuitos, estructuras y técnicas bien conocidos para no complicar la comprensión de esta descripción.

Vista general del sistema I

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

En una realización, el subsistema de procesamiento 101 incluye uno o más procesadores paralelos 112 acoplados al concentrador de memoria 105 mediante un bus u otro enlace de comunicación 113. El enlace de comunicación 113 puede ser uno de cualquier número de tecnologías o protocolos de enlace de comunicación basados en normas, tales como, pero sin limitación, PCI Express, o puede ser una interfaz de comunicaciones o tejido de comunicaciones específica del proveedor. En una realización, los uno o más procesadores paralelos 112 forman un sistema de procesamiento paralelo o vectorial de enfoque computacional 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, los uno o más procesadores paralelos 112 forman un subsistema de procesamiento de gráficos que puede proporcionar píxeles a uno de los uno o más dispositivos de visualización 110A acoplados por medio del concentrador de E/S 107. Los uno o más procesadores paralelos 112 pueden incluir también un controlador de visualización y una interfaz de visualización (no mostrados) para posibilitar una conexión directa a uno o más dispositivos de visualización 110B.

Dentro del subsistema de E/S 111, una unidad de almacenamiento de sistema 114 puede conectarse al concentrador de E/S 107 para proporcionar un mecanismo de almacenamiento para el sistema informático 100. 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, tales como un adaptador de red 118 y/o un adaptador de red inalámbrica 119 que pueden estar integrados en la plataforma, y otros diversos dispositivos que puedan añadirse por medio de uno o más dispositivo(s) de adición 120. El adaptador de red 118 puede ser un adaptador de Ethernet u otro adaptador de red cableado. El adaptador de red inalámbrico 119 puede incluir uno o más de un dispositivo de red de Wi-Fi, de Bluetooth, de comunicación de campo cercano (NFC) o de otro tipo que incluya 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 también puede conectarse 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 cualesquiera otras interfaces y/o protocolo(s) de comunicación de bus o de punto a punto, tales como la interconexión de alta velocidad NV-Link, o protocolos de interconexión conocidos en la técnica.

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

módulo de múltiples chips (MCM), que puede interconectarse con otros módulos de múltiples chips para dar un sistema informático modular.

Se apreciará que el sistema informático 100 mostrado en el presente documento es ilustrativo y que son posibles variaciones y modificaciones. La topología de conexión, incluyendo el número y disposición de puentes, el número de procesador(es) 102 y el número de procesador(es) paralelo(s) 112 puede modificarse como se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 está conectada a los procesador(es) 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/los procesador(es) 102. En otras topologías alternativas, el/los procesador(es) paralelo(s) 112 se conecta(n) al concentrador de E/S 107 o directamente a uno del/de los uno o más procesador(es) 102, en lugar de al concentrador de memoria 105. En otras realizaciones, el concentrador de E/S 107 y el concentrador de memoria 105 pueden integrarse en un único chip. Algunas realizaciones pueden incluir dos o más conjuntos de procesadores 102 conectados mediante múltiples zócalos, que pueden acoplarse con dos o más instancias del/de los procesador(es) paralelo(s) 112.

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

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

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

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

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

En una realización, pueden asignarse diferentes agrupaciones 214A-214N de la matriz de agrupaciones de procesamiento 212 para procesar diferentes tipos de programas o para realizar diferentes tipos de cálculos.

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

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

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

Durante la operación, la matriz de agrupaciones de procesamiento 212 puede recibir tareas de procesamiento que se van a ejecutar a través del planificador 210, que recibe comandos que definen tareas de procesamiento desde el extremo frontal 208. Para operaciones de procesamiento de gráficos, las tareas de procesamiento pueden incluir índices de datos que hay que procesar, por ejemplo, datos de superficie (parche), datos de primitiva, datos de vértice y/o datos de píxel, así como parámetros de estado y comandos que definen cómo han de procesarse los datos (por ejemplo, qué programa ha de ejecutarse). El planificador 210 puede configurarse para extraer los índices que corresponden a las tareas o puede recibir los índices desde el extremo frontal 208. El extremo frontal 208 puede configurarse para garantizar que la matriz de agrupaciones de procesamiento 212 está configurada en un estado válido antes de que se inicie la carga de trabajo especificada por memorias intermedias de comando de entrada (por ejemplo, memorias intermedias de lotes, memorias intermedias de inserción, etc.).

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

En diversas realizaciones, las unidades de memoria 224A-224N pueden incluir diversos tipos de dispositivos de memoria, que incluyen memoria de acceso aleatorio dinámica (DRAM) o memoria de acceso aleatorio de gráficos, tal como la memoria de acceso aleatorio de gráficos síncrona (SGRAM), que incluye la memoria de tasa de datos doble de gráficos (GDDR). En una realización, las unidades de memoria 224A-224N pueden incluir también memoria 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 puede seleccionarse a partir de uno de diversos diseños convencionales. Los objetivos de representación, tales como las memorias intermedias de fotograma o los mapas de textura pueden almacenarse a lo largo de las unidades de memoria 224A-224N, permitiendo que las unidades de subdivisión 220A-220N escriban partes de cada objetivo de representación en paralelo para usar de manera eficiente el ancho de banda disponible de la memoria de procesador paralelo 222. En algunas realizaciones, puede excluirse una instancia local de la memoria de procesador paralelo 222 en favor de un diseño de memoria unificado que utiliza memoria de sistema junto con memoria caché local.

En una realización, una cualquiera de las agrupaciones 214A-214N de la matriz de agrupaciones de procesamiento 212 puede procesar datos que se escribirán en cualquiera de las unidades de memoria 224A-224N dentro de la memoria de procesador paralelo 222. La barra transversal de memoria 216 puede configurarse para transferir la salida

de cada agrupación 214A-214N a cualquier unidad de subdivisión 220A-220N o a otra agrupación 214A-214N, que puede realizar operaciones de procesamiento adicionales sobre la salida. Cada agrupación 214A-214N puede comunicarse con la interfaz de memoria 218 a través de la barra transversal de memoria 216 para leer desde o escribir en diversos dispositivos de memoria externos. En una realización, la barra transversal de memoria 216 tiene una conexión a la interfaz de memoria 218 para comunicarse con la unidad de E/S 204, así como una conexión a una instancia local de la memoria de procesador paralelo 222, lo que 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 subdivisión 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 configurarse para interfuncionar 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 en relación con otras instancias. Los sistemas que incorporan una o más instancias de la unidad de procesamiento paralelo 202 o el procesador paralelo 200 pueden implementarse en una diversidad de configuraciones y factores de forma, incluyendo, pero sin limitación, ordenadores personales de sobremesa, portátiles o de mano, servidores, estaciones de trabajo, consolas de juegos y/o sistemas embebidos.

La **Figura 2B** es un diagrama de bloques de una unidad de subdivisión 220, de acuerdo con una realización. En una realización, la unidad de subdivisión 220 es una instancia de una de las unidades de subdivisión 220A-220N de la **Figura 2A**. Como se ilustra, la unidad de subdivisión 220 incluye una caché L2 221, una interfaz de memoria intermedia de fotogramas 225 y una ROP 226 (unidad de operaciones de rasterización). La caché L2 221 es una caché de lectura/escritura que está configurada para realizar operaciones de carga y de almacenamiento recibidas desde la barra transversal de memoria 216 y la ROP 226. Los desaciertos de lectura y las solicitudes de escritura diferida urgente son emitidas por la caché L2 221 a la interfaz de memoria intermedia de fotogramas 225 para su procesamiento. También pueden enviarse actualizaciones sucias a la memoria intermedia de fotogramas mediante la interfaz de memoria intermedia de fotogramas 225 para un procesamiento oportunista. En una realización, la interfaz de memoria intermedia de fotogramas 225 interacciona con una de las unidades de memoria en memoria de procesador paralelo, tales como las unidades de memoria 224A-224N de la **Figura 2A** (por ejemplo, dentro de la memoria de procesador paralelo 222).

En las aplicaciones de gráficos, la ROP 226 es una unidad de procesamiento que realiza operaciones de rasterización tales como estarcido, prueba z, mezcla y similares. La ROP 226 emite entonces datos de gráficos procesados que se almacenan en una memoria de gráficos. En algunas realizaciones, la ROP 226 incluye lógica de compresión para comprimir datos z o de color que se escriben en memoria y descomprimir datos z o de color que se leen desde memoria. En algunas realizaciones, la ROP 226 se incluye dentro de cada agrupación de procesamiento (por ejemplo, la agrupación 214A-214N de la **Figura 2A**) en lugar de dentro de la unidad de subdivisión 220. En tal realización, se transmiten solicitudes de lectura y de escritura de datos de píxel a través de la barra transversal de memoria 216 en lugar de datos de fragmento de píxel.

Los datos de gráficos procesados pueden visualizarse en un dispositivo de visualización, tal como uno de los uno o más dispositivo(s) de visualización 110 de la **Figura 1**, encaminarse para su procesamiento adicional por medio del/de los procesador(es) 102, o encaminarse para su procesamiento adicional por medio de 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 2A**. La agrupación de procesamiento 214 puede configurarse para ejecutar muchos hilos en paralelo, donde el término "hilo" se refiere a una instancia de un programa particular que se ejecuta en un conjunto particular de datos de entrada. En algunas realizaciones, se usan técnicas de emisión de instrucciones de única instrucción de múltiples datos (SIMD) para soportar la ejecución paralela de un gran número de hilos sin proporcionar múltiples unidades de instrucción independientes. En otras realizaciones, se usan técnicas de única instrucción de múltiples hilos (SIMT) para soportar la ejecución paralela de un gran número de hilos generalmente sincronizados, usando una unidad de instrucciones común configurada para emitir instrucciones en un conjunto de motores de procesamiento dentro de cada una de las agrupaciones de procesamiento. A diferencia del régimen de ejecución de SIMD, donde todos los motores de procesamiento ejecutan habitualmente instrucciones idénticas, la ejecución de SIMT permite que diferentes hilos sigan más fácilmente rutas de ejecución divergentes a través de un programa de hilos dado. Los expertos en la materia entenderán que un régimen de procesamiento de SIMD representa un subconjunto funcional de un régimen de procesamiento de SIMT.

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

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

Las instrucciones transmitidas a la agrupación de procesamiento 214 constituyen un hilo. Un conjunto de hilos que se ejecutan a lo largo del conjunto de motores de procesamiento paralelo es un grupo de hilos. Un grupo de hilos ejecuta el mismo programa sobre 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 inactivos durante los ciclos en los que se está procesando ese grupo de hilos. Un grupo de hilos puede incluir también más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234. Cuando el grupo de hilos incluye más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234, puede realizarse un procesamiento a lo largo de ciclos de reloj consecutivos. En una realización, pueden ejecutarse múltiples grupos de hilos concurrentemente en un multiprocesador de gráficos 234.

En una realización, el multiprocesador de gráficos 234 incluye una memoria caché interna para realizar operaciones de carga y de almacenamiento. En una realización, el multiprocesador de gráficos 234 puede prescindir de una caché interna y usar una memoria caché (por ejemplo, la caché L1 308) dentro de la agrupación de procesamiento 214. Cada multiprocesador de gráficos 234 también tiene acceso a cachés de L2 dentro de las unidades de subdivisión (por ejemplo, las unidades de subdivisión 220A-220N de la Figura 2A) que se comparten entre todas las agrupaciones de procesamiento 214 y pueden usarse para transferir datos entre hilos. El multiprocesador de gráficos 234 puede acceder también a memoria global fuera de chip, que puede incluir una o más de memoria de procesador paralelo local y/o memoria de sistema. Cualquier memoria externa a la unidad de procesamiento paralelo 202 puede usarse 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 2A**. La MMU 245 incluye un conjunto de entradas de tabla de página (PTE) usadas para mapear una dirección virtual a una dirección física de una tesela (más información sobre el teselado) y, opcionalmente, un índice de línea de caché. La MMU 245 puede incluir memorias intermedias de traducción adelantada (TLB) de direcciones 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 de datos de superficie para permitir una intercalación de solicitud eficiente entre unidades de subdivisión. El índice de líneas de caché puede usarse para determinar si una solicitud de una línea de caché es un acierto o un desacierto.

En aplicaciones de gráficos e informáticas, una agrupación de procesamiento 214 puede configurarse de manera que cada multiprocesador de gráficos 234 se acopla a una unidad de textura 236 para realizar operaciones de mapeo de textura, por ejemplo, determinar posiciones de muestra de textura, leer datos de textura y filtrar los datos de textura. Los datos de textura se leen desde una caché L1 de textura interna (no mostrada) o, en algunas realizaciones, desde la caché L1 dentro del multiprocesador de gráficos 234 y se extraen desde una caché L2, memoria de procesador paralelo local o memoria de sistema, según sea necesario. Cada multiprocesador de gráficos 234 emite tareas procesadas a la barra transversal de datos 240 para proporcionar la tarea procesada a otra agrupación de procesamiento 214 para su procesamiento adicional o para almacenar la tarea procesada en una caché L2, memoria de procesador paralelo local o memoria de sistema mediante la barra transversal de memoria 216. Una preROP 242 (unidad de operaciones prerrasterización) está configurada para recibir datos desde el multiprocesador de gráficos

234, dirigir datos a unidades de ROP, que pueden ubicarse con unidades de subdivisión como se describe en el presente documento (por ejemplo, las unidades de subdivisión 220A-220N de la **Figura 2A**). La unidad preROP 242 puede realizar optimizaciones para la mezcla de color, organizar datos de color de píxel y realizar traducciones de dirección.

Se apreciará que la arquitectura de núcleo descrita en el presente documento es ilustrativa y que son posibles variaciones y modificaciones. 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 solo 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 configurarse para funcionar independientemente de otras agrupaciones de procesamiento 214 usando unidades de procesamiento separadas y distintas, cachés L1, etc.

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

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

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

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

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

Las **Figuras 3A-3B** ilustran multiprocesadores de gráficos adicionales, 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 configurarse como un multiprocesador de transmisión por flujo continuo (SM) que puede realizar la ejecución simultánea de un gran número de hilos de ejecución.

La **Figura 3A** muestra un multiprocesador de gráficos 325 de acuerdo con una realización adicional. El multiprocesador de gráficos 325 incluye múltiples instancias adicionales de unidades de recurso de ejecución relativas al multiprocesador de gráficos 234 de la **Figura 2D**. Por ejemplo, el multiprocesador de gráficos 325 puede incluir múltiples instancias de la unidad de instrucciones 332A-332B, el archivo de registro 334A-334B y la(s) unidad(es) de textura 344A-344B. El multiprocesador de gráficos 325 también incluye múltiples conjuntos de unidades de ejecución de cálculo o de gráficos (por ejemplo, el núcleo de GPGPU 336A-336B, el núcleo de GPGPU 337A-337B, el núcleo de GPGPU 338A-338B) y múltiples conjuntos de unidades de carga/almacenamiento 340A-340B. En una realización, las unidades de recurso de ejecución tienen una caché de instrucciones común 330, una memoria caché de textura y/o de datos 342 y una memoria compartida 346. Los diversos componentes pueden comunicarse mediante 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 recursos de ejecución 356A-356D, donde cada conjunto de recursos de ejecución incluye múltiples unidades de instrucciones, archivos de registro, núcleos de GPGPU y unidades de carga-almacenamiento, como se ilustra en la **Figura 2D** y en la **Figura 3A**. Los recursos de ejecución 356A-356D pueden funcionar en conjunto con la(s) unidad(es) de textura 360A-360D para operaciones de textura, mientras comparten una caché de instrucciones 354 y una memoria compartida 362. En una realización, los recursos de ejecución 356A-356D pueden compartir una caché de instrucciones 354 y una memoria compartida 362, así como múltiples instancias de una memoria caché de textura y/o de datos 358A-358B. Los diversos componentes pueden comunicarse a través de un tejido de interconexión 352 similar al tejido de interconexión 327 de la **Figura 3A**.

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, incluyendo, sin limitación, uno o más procesadores de aplicación móvil, una o más unidades centrales de procesamiento (CPU) de sobremesa o de servidor, incluyendo CPU de múltiples núcleos, una o más unidades de procesamiento paralelo, tales como la unidad de procesamiento paralelo 202 de la **Figura 2A**, así como uno o más procesadores de gráficos o unidades de procesamiento de propósito especial, sin apartarse del alcance de las realizaciones descritas en el presente documento.

En algunas realizaciones, un procesador paralelo o GPGPU como se describe en el presente documento está acoplado de manera comunicativa a núcleos de anfitrión/procesador para acelerar operaciones de gráficos, operaciones de aprendizaje automático, operaciones de análisis de patrones y diversas funciones de GPU de propósito general (GPGPU). La GPU puede acoplarse de manera comunicativa al procesador/núcleos de anfitrión a través de un bus u otra interconexión (por ejemplo, una interconexión de alta velocidad tal como PCIe o NVLink). En otras realizaciones, la GPU puede integrarse en el mismo paquete o chip que los núcleos y estar acoplada de manera comunicativa a los núcleos a través de un bus/interconexión de procesador interno (es decir, internamente 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 usa entonces circuitería/lógica dedicada para procesar de manera eficiente estos comandos/instrucciones.

Técnicas para interconexión de GPU a procesador de anfitrión

La **Figura 4A** ilustra una arquitectura ilustrativa en la que una pluralidad de GPU 410-413 están acopladas de manera comunicativa a una pluralidad de procesadores de múltiples núcleos 405-406 a través de enlaces de alta velocidad 440-443 (por ejemplo, buses, interconexiones de punto a punto, etc.). En una realización, los enlaces de alta velocidad 440-443 soportan un caudal de comunicación de 4 GB/s, 30 GB/s, 80 GB/s o superior, dependiendo de la implementación. Pueden usarse diversos protocolos de interconexión incluyendo, 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 protocolos/enlaces iguales o diferentes a los usados para los enlaces de alta velocidad 440-443. De manera similar, dos o más de los procesadores de múltiples núcleos 405-406 pueden conectarse a través del enlace de alta velocidad 433, que puede ser buses de múltiples procesadores simétricos (SMP) que operan a 20 GB/s, 30 GB/s, 120 GB/s o más. Como alternativa, toda la comunicación entre los diversos componentes de sistema mostrados en la **Figura 4A** puede conseguirse usando los mismos protocolos/enlaces (por ejemplo, a través de un tejido de interconexión común). Sin embargo, como se menciona, los principios subyacentes de la invención no están limitados a ningún tipo particular de tecnología de interconexión.

En una realización, cada procesador de múltiples núcleos 405-406 está acoplado de manera comunicativa a una memoria de procesador 401-402, mediante las interconexiones de memoria 430-431, respectivamente, y cada GPU 410-413 está acoplada de manera comunicativa a la memoria de GPU 420-423 a través de las interconexiones de memoria de GPU 450-453, respectivamente. Las interconexiones de memoria 430-431 y 450-453 pueden utilizar las mismas tecnologías de acceso de memoria u otras diferentes. A modo de ejemplo, y no de limitación, las memorias de procesador 401-402 y las memorias de GPU 420-423 pueden ser memorias volátiles, tal como memorias de acceso aleatorio dinámicas (DRAM) (incluyendo DRAM apiladas), SDRAM DDR de gráficos (GDDR) (por ejemplo, GDDR5, GDDR6), o memoria de alto ancho de banda (HBM) y/o pueden ser memorias no volátiles, tales como 3D XPoint o Nano-Ram. En una realización, una parte de las memorias puede ser memoria volátil y otra parte puede ser memoria no volátil (por ejemplo, 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 espacio "de direcciones eficaces") está distribuido entre todas las diversas memorias físicas. Por ejemplo, cada una de las memorias de procesador 401-402 puede comprender 64 GB del espacio de direcciones de memoria de sistema y cada una de las memorias de GPU 420-423 puede comprender 32 GB del espacio de direcciones de memoria de sistema (dando como resultado un total de memoria direccionable de 256 GB 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 se acopla al procesador 407 mediante el enlace de alta velocidad 440. Como alternativa, el módulo de aceleración de gráficos 446 puede integrarse en el mismo paquete o chip que el procesador 407.

El procesador 407 ilustrado 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 han ilustrado para evitar oscurecer los principios subyacentes de la invención, (por ejemplo, unidades de extracción de instrucciones, unidades de predicción de ramificación, 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, una o más cachés compartidas 426 pueden incluirse en la jerarquía de almacenamiento en caché y ser compartidas por conjuntos de núcleos 460A-460D. Por ejemplo, una realización del procesador 407 incluye 24 núcleos, cada uno con su propia caché L1, doce cachés L2 compartidas y doce cachés L3 compartidas. En esta realización, una de las cachés L2 y L3 es compartida por dos núcleos adyacentes. El procesador 407 y el módulo de integración de acelerador de gráficos 446 se conectan con la memoria de sistema 441, que puede incluir las memorias de procesador 401-402.

La coherencia se mantiene para los datos e instrucciones almacenados en las diversas cachés 462A-462D, 456 y la memoria de sistema 441 a través de comunicación entre núcleos a través de un bus de coherencia 464. Por ejemplo, cada caché puede tener una lógica/circuitería de coherencia de caché asociada con la misma para comunicarse a través del bus de coherencia 464 en respuesta a lecturas o escrituras detectadas en líneas de caché particulares. En una implementación, se implementa un protocolo de monitorización de caché a través del bus de coherencia 464 para monitorizar los accesos de caché. Las técnicas de coherencia/monitorización de caché son bien entendidas por los expertos en la técnica y no se describirán en el presente caso en detalle para evitar complicar los principios subyacentes de la invención.

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

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

En una realización, el circuito de integración de acelerador 436 incluye una unidad de gestión de memoria (MMU) 439 para realizar diversas funciones de gestión de memoria tales como traducciones de memoria virtual a física (también denominadas traducciones de memoria eficaz 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 dirección virtual/eficaz a física/real. En una implementación, una caché 438 almacena comandos y datos para un 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 menciona, esto puede conseguirse mediante el circuito intermediario 425 que toma parte en el mecanismo de coherencia de caché en nombre de la caché 438 y las memorias 433-434, N (por ejemplo, enviando actualizaciones a la caché 438 relacionadas con modificaciones/accesos de líneas de caché en las cachés de procesador 462A-462D, 456 y recibiendo actualizaciones desde la caché 438).

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

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

Por tanto, el circuito de integración de acelerador actúa como un puente al sistema para el módulo de aceleración de gráficos 446 y proporciona servicios de traducción de direcciones y de caché de sistema. Además, el circuito de integración de acelerador 436 puede proporcionar instalaciones de virtualización para que el procesador de anfitrión gestione la virtualización de los motores de procesamiento 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 visto por el procesador de anfitrión 407, cualquier procesador de anfitrión puede direccionar estos recursos directamente usando un valor de dirección eficaz. Una función del circuito de integración de acelerador 436, en una realización, es la separación física de los motores de procesamiento de gráficos 431-432, N de modo que aparecen al sistema como unidades independientes.

Como se menciona, 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 son procesados 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 (incluyendo 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 sean datos que serán usados con mayor frecuencia por los motores de procesamiento de gráficos 431-432, N y, preferentemente, no usados por los núcleos 460A-460D (al menos no con frecuencia). De manera similar, el mecanismo de desvío intenta mantener datos que necesitan los núcleos (y, preferentemente, no los motores de procesamiento de gráficos 431-432, N) dentro de las cachés 462A-462D, 456 de los núcleos y la memoria de sistema 411.

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

virtualización). Este último puede incluir modelos de programación que son controlados por el circuito de integración de acelerador 436 y modelos de programación que son controlados por el módulo de aceleración de gráficos 446.

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

En los modelos de programación de proceso dedicado, los motores de procesamiento de gráficos 431-432, N, pueden ser compartidos por múltiples subdivisiones de aplicación/VM. Los modelos compartidos requieren que un hipervisor de sistema virtualice los motores de procesamiento de gráficos 431-432, N para permitir el acceso por cada sistema operativo. Para sistemas de subdivisión única 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 431-432, N individual selecciona un elemento de proceso usando un manejador de proceso. En una realización, se almacenan elementos de proceso en la memoria de sistema 411, y estos son direccionables usando las técnicas de traducción de dirección eficaz a dirección real descritas en el presente documento. El manejador de proceso puede ser un valor específico de la implementación proporcionado al proceso de anfitrión cuando se registra su contexto con el motor de procesamiento de gráficos 431-432, N (es decir, llamando a software de sistema para añadir el elemento de proceso a la lista vinculada de elementos de proceso). Los 16 bits inferiores del manejador de proceso pueden ser el desplazamiento del elemento de proceso dentro de la lista vinculada de elementos de proceso.

La **Figura 4D** ilustra un segmento de integración de acelerador 490 ilustrativo. Como se usa en el presente documento, un "segmento" comprende una parte especificada de los recursos de procesamiento del circuito de integración de acelerador 436. El espacio de direcciones eficaces de aplicación 482 dentro de la memoria de sistema 411 almacena elementos de proceso 483. En una realización, los elementos de proceso 483 se almacenan en respuesta a las invocaciones de GPU 481 desde las aplicaciones 480 ejecutadas en el procesador 407. Un elemento de proceso 483 contiene el estado de proceso para la aplicación 480 correspondiente. 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 este último caso, el WD 484 es un puntero a la cola de solicitudes de trabajos en el espacio de direcciones 482 de la aplicación.

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

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

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

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

Tabla 1 - Registros inicializados por hipervisor

1	Registro de control de segmento
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 subdivisión lógica
8	Puntero de registro de utilización del acelerador del hipervisor de dirección real (RA)
9	Registro de descripción de almacenamiento

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

Tabla 2 - Registros inicializados por sistema operativo

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

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

La **Figura 4E** ilustra detalles adicionales para 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 de 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 compartida permiten que todos o un subconjunto de procesos de todas o un subconjunto de subdivisiones en el sistema usen un módulo de aceleración de gráficos 446. Hay dos modelos de programación donde el módulo de aceleración de gráficos 446 se comparte por múltiples procesos y subdivisiones: compartido en segmentos de tiempo y compartido dirigido a gráficos.

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

En una realización, para el modelo compartido, se requiere que la aplicación 480 realice una llamada al sistema operativo 495 con un tipo de módulo de aceleración de gráficos 446, un descriptor de trabajo (WD), un valor de registro de máscara de autoridad (AMR) y un puntero de área de guardado/restauración de contexto (CSRP). El tipo del módulo de aceleración de gráficos 446 describe la función de aceleración dirigida como objetivo para la llamada de sistema. El tipo del módulo de aceleración de gráficos 446 puede ser un valor específico de sistema. El WD se formatea específicamente para el módulo de aceleración de gráficos 446 y puede estar en forma de un comando de módulo de aceleración de gráficos 446, un puntero de dirección eficaz a una estructura definida por el usuario, un puntero de dirección eficaz a una cola de comandos o cualquier otra estructura de datos para describir el trabajo que va a hacerse por el módulo de aceleración de gráficos 446. En una realización, el valor de AMR es el estado de AMR que usar para el proceso actual. El valor pasado al sistema operativo es similar a que una aplicación establezca el AMR. Si las implementaciones del circuito de integración de 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 de hipervisor. Opcionalmente, el hipervisor 496 puede aplicar 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 contiene la dirección eficaz de un área en el espacio de direcciones de la aplicación 482 para que el módulo de aceleración de gráficos 446 guarde y restaure el estado de contexto. Este puntero es opcional si no se requiere que se guarde estado alguno entre trabajos o cuando se da prioridad a un trabajo. El área de guardado/restauración de contexto puede ser memoria de sistema anclada.

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

Tabla 3 - Parámetros de llamada de SO a hipervisor

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

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

Tabla 4 - Información de elemento de proceso

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

En una realización, el hipervisor inicializa una pluralidad de registros 445 del segmento de integración de 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 físico 401-402 y a las memorias de GPU 420-423. En esta implementación, las operaciones ejecutadas en las GPU 410-413 utilizan el mismo espacio de direcciones de memoria virtual/eficaz para acceder a las memorias de procesador 401-402 y viceversa, simplificando de esta manera la programabilidad. En una realización, una primera parte del espacio de direcciones virtual/eficaz está asignada a la memoria de procesador 401, una segunda parte a la segunda memoria de procesador 402, una tercera parte a la memoria de GPU 420, y así sucesivamente. El espacio de memoria virtual/eficaz total (en ocasiones denominado el espacio de direcciones eficaz) está distribuido, de esta manera, a lo largo de cada una de las memorias de procesador 401-402 y de las memorias de GPU 420-423, permitiendo que cualquier procesador o GPU acceda a cualquier memoria física con una dirección virtual mapeada a esa memoria.

En una realización, la circuitería de gestión de desvío/coherencia 494A-494E dentro de una o más de las MMU 439A-439E garantiza la coherencia de caché entre las cachés de los procesadores de anfitrión (por ejemplo, 405) y las GPU 410-413 e implementa técnicas de desvío que indican las memorias físicas en las que deben almacenarse ciertos

tipos de datos. Aunque se ilustran múltiples instancias de la circuitería de gestión de desvío/coherencia 494A-494E en la **Figura 4F**, la circuitería de desvío/coherencia puede implementarse dentro de la MMU de uno o más procesadores de anfitrión 405 y/o dentro del circuito de integración de acelerador 436.

Una realización permite que la memoria anexada a GPU 420-423 se mapee como parte de la memoria de sistema, y que se acceda a la misma usando tecnología de memoria virtual compartida (SVM), pero sin sufrir de 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 la descarga de la GPU. Esta disposición permite que el software del procesador de anfitrión 405 establezca operandos y acceda a resultados de cálculo, sin la sobrecarga de copias de datos de DMA de E/S tradicionales. Tales copias tradicionales implican llamadas de controlador, interrupciones y accesos de E/S mapeada en memoria (MMIO) que son, todos ellos, ineficientes en relación con los accesos de memoria sencillos. Al mismo tiempo, la capacidad de acceder a la memoria adjunta a la GPU 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 escritura de transmisión por flujo continuo sustancial, por ejemplo, la sobrecarga de coherencia de caché puede reducir significativamente el ancho de banda de escritura eficaz observado por una GPU 410-413. La eficiencia del establecimiento de operandos, la eficiencia del acceso a resultados y la eficiencia del cálculo de GPU desempeñan, todas ellas, un papel en la determinación de la eficacia de la descarga de GPU.

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

En una implementación, se accede a la entrada de tabla de desvíos asociada a cada acceso a la memoria adjunta a la GPU 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 GPU se reenvían directamente a una memoria de GPU 420-423 correspondiente. Las solicitudes locales desde 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 del procesador 405 que encuentran la página solicitada en el desvío de procesador de anfitrión completan la solicitud como una lectura de memoria normal. Como alternativa, solicitudes dirigidas a una página con desvío de GPU pueden redirigirse a la GPU 410-413. A continuación, la GPU puede hacer que la página pase a un desvío de procesador anfitrión si no está usando actualmente la página.

El estado de desvío de una página puede cambiarse mediante o bien un mecanismo basado en software, o bien un mecanismo basado en software asistido por hardware, o bien, para un conjunto limitado de casos, 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 de dispositivos de la GPU que, a su vez, envía un mensaje a (o pone en cola un descriptor de comandos para) la GPU que le indica que cambie el estado de desvío y, para algunas transiciones, que realice una operación de vaciado de caché en el anfitrión. Se requiere la operación de vaciado de caché para una transición desde un desvío del procesador de anfitrión 405 a un desvío de GPU, pero no se requiere para la transacción opuesta.

En una realización, la coherencia de caché se mantiene haciendo temporalmente que las páginas con desvío de GPU no puedan ser almacenadas en caché por el procesador de anfitrión 405. Para acceder a estas páginas, el procesador 405 puede solicitar acceso desde la GPU 410 que puede conceder, o no, acceso de manera inmediata, 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 desvío de GPU son aquellas que son requeridas por la GPU, pero no por el procesador de anfitrión 405, y viceversa.

Canalización de procesamiento de gráficos

La **Figura 5** ilustra una canalización de procesamiento de gráficos 500, de acuerdo con una realización. En una realización, un procesador de gráficos puede implementar la canalización de procesamiento de gráficos 500 ilustrada. El procesador de gráficos puede incluirse dentro de los subsistemas de procesamiento paralelo como se describe en el presente documento, tal como el procesador paralelo 200 de la **Figura 2A**, que, en una realización, es una variante del/de los procesador(es) paralelo(s) 112 de la **Figura 1**. Los diversos sistemas de procesamiento paralelo pueden implementar la canalización de procesamiento de gráficos 500 mediante una o más instancias de la unidad de procesamiento paralelo (por ejemplo, la unidad de procesamiento paralelo 202 de la **Figura 2A**) como se describe en el presente documento. Por ejemplo, una unidad sombreadora (por ejemplo, el multiprocesador de gráficos 234 de la **Figura 2D**) puede configurarse para realizar las funciones de una o más de una unidad de procesamiento de vértices 504, una unidad de procesamiento de control de teselación 508, una unidad de procesamiento de evaluación de

teselación 512, una unidad de procesamiento de geometría 516 y una unidad de procesamiento de fragmentos/píxeles 524. Las funciones del ensamblador de datos 502, los ensambladores de primitivas 506, 514, 518, la unidad de teselación 510, el rasterizador 522 y la unidad de operaciones de rasterización 526 también pueden ser realizadas por otros motores de procesamiento dentro de una agrupación de procesamiento (por ejemplo, la agrupación de procesamiento 214 de la **Figura 3A**) y una unidad de subdivisión correspondiente (por ejemplo, la unidad de subdivisión 220A-220N de la **Figura 2C**). La canalización de procesamiento de gráficos 500 puede implementarse también usando unidades de procesamiento especializadas para una o más funciones. En una realización, una o más partes de la canalización de procesamiento de gráficos 500 pueden realizarse mediante lógica de procesamiento paralelo dentro de un procesador de propósito general (por ejemplo, CPU). En una realización, una o más partes de la canalización de procesamiento de gráficos 500 pueden acceder a una memoria en chip (por ejemplo, la memoria de procesador paralelo 222 como en la **Figura 2A**) mediante una interfaz de memoria 528, que puede ser una instancia de la interfaz de memoria 218 de la **Figura 2A**.

En una realización, el ensamblador de datos 502 es una unidad de procesamiento que recopila datos de vértice para superficies y primitivas. El ensamblador de datos 502 emite entonces los datos de vértice, incluyendo los atributos de vértice, a la unidad de procesamiento de vértices 504. La unidad de procesamiento de vértices 504 es una unidad de ejecución programable que ejecuta programas sombreadores de vértices, iluminando y transformando datos de vértice según lo especificado por los programas sombreadores de vértices. La unidad de procesamiento de vértices 504 lee datos que se almacenan en caché, local o de sistema para su uso en el procesamiento de los datos de vértice y puede programarse para transformar los datos de vértice desde una representación de coordenadas basada en objetos hasta 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 vértice desde la unidad de procesamiento de vértices 504. El ensamblador de primitivas 506 lee atributos de vértice almacenados según sea necesario y construye primitivas de gráficos para su procesamiento por la unidad de procesamiento de control de teselación 508. Las primitivas de gráficos incluyen triángulos, segmentos de línea, puntos, parches y así sucesivamente, según sea soportado por diversas interfaces de programación de aplicaciones (API) de procesamiento de gráficos.

La unidad de procesamiento de control de teselación 508 trata los vértices de entrada como puntos de control para un parche geométrico. Los puntos de control se transforman de una representación de entrada a partir del parche (por ejemplo, las bases del parche) a una representación que es adecuada para su uso en una evaluación superficial por la unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de control de teselación 508 puede calcular también factores de teselación para bordes de parches geométricos. Un factor de teselación es aplicable a un único borde y cuantifica un nivel de detalle, dependiente de la vista, asociado con el borde. Una unidad de teselación 510 está configurada para recibir los factores de teselación para bordes de un parche y para teselar el parche en múltiples primitivas geométricas, tales como primitivas de línea, de triángulo o cuadrilaterales, que se transmiten a una unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de evaluación de teselación 512 opera sobre 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 vértices desde la unidad de procesamiento de evaluación de teselación 512, leyendo 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 según se especifica por los programas sombreadores de geometría. En una realización, la unidad de procesamiento de geometría 516 está programada para subdividir las primitivas de gráficos en una o más primitivas de gráficos nuevas y calcular parámetros usados para rasterizar las nuevas primitivas de gráficos.

En algunas realizaciones, la unidad de procesamiento de geometría 516 puede añadir o borrar elementos en el flujo de geometría. La unidad de procesamiento de geometría 516 emite los parámetros y vértices que especifican primitivas de gráficos nuevas 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 memoria de procesador paralelo o 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 el recorte, la selección y el ajuste a escala de ventana gráfica y emite primitivas de gráficos procesadas a un rasterizador 522.

El rasterizador 522 puede realizar optimizaciones de selección de profundidad y otras basadas en la profundidad. El rasterizador 522 también realiza una conversión de exploración sobre las nuevas primitivas de gráficos para generar fragmentos y emitir esos fragmentos y datos de cobertura asociados a la unidad de procesamiento de fragmentos/píxeles 524.

La unidad de procesamiento de fragmentos/píxeles 524 es una unidad de ejecución programable que está configurada para ejecutar programas sombreadores de fragmentos o programas sombreadores de píxeles. Transformando, la

unidad de procesamiento de fragmentos/píxeles 524, fragmentos o píxeles recibidos desde el rasterizador 522, según sea especificado por los programas sombreadores de fragmentos o de píxeles. Por ejemplo, la unidad de procesamiento de fragmentos/píxeles 524 puede programarse para realizar operaciones que incluyen, pero sin limitación, 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 fragmentos/píxeles 524 puede leer datos que se almacenan o bien en la memoria de procesador paralelo o bien en la memoria de sistema para su uso cuando se procesan los datos de fragmento. Los programas sombreadores de fragmentos o de píxeles pueden estar configurados para sombrear a granularidad de muestra, de píxel, de tesela u otras dependiendo de la tasa de muestreo configurada para las unidades de procesamiento.

La unidad de operaciones de rasterización 526 es una unidad de procesamiento que realiza operaciones de rasterización que incluyen, pero sin limitación estarcido, prueba z, mezcla y similares, y emite datos de píxel como datos de gráficos procesados para almacenarse en memoria de gráficos (por ejemplo, la me**moria de procesador paralelo 222 como en** la **Figura 2A**, y/o la me**moria de sistema 104 como en** la **Figura 1**), para visualizarse en el/los uno o más dispositivo(s) de visualización 110 o para su procesamiento adicional por uno del/los uno o más procesador(es) 102 o procesador(es) paralelo(s) 112. En algunas realizaciones, la unidad de operaciones de rasterización 526 está configurada para comprimir datos z o de color que se escriben en memoria y descomprimir datos z o de color que se leen desde la memoria.

La **Figura 6** ilustra un dispositivo informático 600 que aloja un mecanismo de reconocimiento y seguridad 610 de acuerdo con una realización. El dispositivo informático 600 representa un dispositivo de comunicación y procesamiento de datos que incluye (pero sin limitación) dispositivos portátiles inteligentes, teléfonos inteligentes, dispositivos de realidad virtual (VR), pantallas montadas en la cabeza (HMD), ordenadores móviles, dispositivos del Internet de las cosas (IoT), ordenadores portátiles, ordenadores de sobremesa, ordenadores de servidor, etc., y ser similar o igual que el dispositivo informático 100 de la **Figura 1;** en consecuencia, por brevedad, claridad y facilidad de comprensión, muchos de los detalles establecidos anteriormente con referencia a las **Figuras 1-5** no se analizan ni se repiten adicionalmente más adelante.

El dispositivo informático 600 puede incluir además (sin limitaciones) una máquina autónoma o un agente artificialmente inteligente, tal como un agente o máquina mecánica, un agente o máquina electrónica, un agente o máquina virtual, un agente o máquina electromecánica, etc. Ejemplos de máquinas o agentes artificialmente inteligentes pueden incluir (sin limitación) robots, vehículos autónomos (por ejemplo, automóviles autónomos, aviones autónomos, barcos autónomos, etc.), equipos autónomos (vehículos de construcción autónomos, equipos médicos autónomos, etc.), y/o similares. A través de todo este documento, "dispositivo informático" puede denominarse de manera intercambiable "máquina autónoma" o "agente artificialmente inteligente" o simplemente "robot".

Se contempla que, aunque a lo largo de este documento se hace referencia a "vehículo autónomo" y "conducción autónoma", las realizaciones no están limitadas en este sentido. Por ejemplo, "vehículo autónomo" no se limita a un automóvil, sino que puede incluir cualquier número y tipo de máquinas autónomas, tales como robots, equipos autónomos, dispositivos domésticos autónomos y/o similares, y una o más tareas u operaciones relacionadas con tales máquinas autónomas pueden denominarse de manera intercambiable con la conducción autónoma.

El dispositivo informático 600 puede incluir además (sin limitaciones) grandes sistemas informáticos, tales como ordenadores de servidor, ordenadores de sobremesa, etc., y puede incluir además decodificadores de salón (por ejemplo, decodificadores de salón de televisión por cable basados en Internet, etc.), dispositivos basados en el sistema de posicionamiento global (GPS), etc. El dispositivo informático 600 puede incluir dispositivos informáticos móviles que dan servicio como dispositivos de comunicación, tales como teléfonos celulares, que incluyen teléfonos inteligentes, asistentes digitales personales (PDA), ordenadores de tabletas, ordenadores portátiles, lectores electrónicos, televisores inteligentes, plataformas de televisión, dispositivos llevables (por ejemplo, gafas, relojes, pulseras, tarjetas inteligentes, joyas, prendas de vestir, etc.), reproductores multimedia, 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 unidad de procesamiento gráfico ("GPU" o simplemente "procesador de gráficos") 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, estructura de controlador de modo de usuario (UMDF), UMDF, o simplemente "controlador") 616, unidad central de procesamiento ("CPU" o simplemente "procesador de aplicaciones") 612, memoria 608, dispositivos de red, controladores, o similares, así como fuentes de entrada/salida (E/S) 604, tales como pantallas táctiles, paneles táctiles, almohadillas táctiles, teclados virtuales o normales, ratones virtuales o normales, puertos, conectores, etc. El dispositivo informático 600 puede incluir un sistema operativo (SO) 606 que da servicio como interfaz entre el hardware y/o los recursos físicos del dispositivo informático 600 y un usuario. Se contempla que el procesador de gráficos 614 y el procesador de aplicaciones 612 pueden ser uno o más procesadores 102 de la **Figura 1**.

Debe apreciarse que para determinadas implementaciones puede preferirse un sistema menos o más equipado que el ejemplo descrito anteriormente. Por lo tanto, la configuración del dispositivo informático 600 puede variar de una implementación a otra dependiendo de numerosos factores, tales como limitaciones de precio, requisitos de rendimiento, mejoras tecnológicas u otras circunstancias.

Las realizaciones pueden implementarse como cualquiera o una combinación de: uno o más microchips o circuitos integrados interconectados usando una placa base, lógica cableada, software almacenado por un dispositivo de memoria y ejecutado por un microprocesador, firmware, un circuito integrado de aplicación específica (ASIC), y /o una matriz de puertas programables en campo (FPGA). Los términos "lógica", "módulo", "componente", "motor" y "mecanismo" pueden incluir, a modo de ejemplo, software o hardware y/o combinaciones de software y hardware.

En una realización, el mecanismo de reconocimiento y seguridad 610 puede alojarse o facilitarse por el sistema operativo 606 del dispositivo informático 600. En otra realización, el mecanismo de reconocimiento y seguridad 610 puede estar alojado en o ser parte de la unidad de procesamiento de gráficos ("GPU" o simplemente "procesador de gráficos") 614 o firmware del procesador de gráficos 614. Por ejemplo, el mecanismo de reconocimiento y seguridad 610 puede integrarse o implementarse como parte del hardware de procesamiento del procesador de gráficos 614. De manera similar, en otro otra realización más, el mecanismo de reconocimiento y seguridad 610 puede estar alojado en o ser parte de la unidad central de procesamiento ("CPU" o simplemente "procesador de aplicaciones") 612. Por ejemplo, el mecanismo de reconocimiento y seguridad 610 puede embeberse en o implementarse como parte del hardware de procesamiento del procesador de aplicaciones 612. En otra realización más, el mecanismo de reconocimiento y seguridad 610 puede estar alojado en o ser parte de cualquier número y tipo de componentes del dispositivo informático 600, tal como una parte del mecanismo de reconocimiento y seguridad 610 puede estar alojado en o ser parte del sistema operativo 606, otra parte puede estar alojada en o ser parte del procesador de gráficos 614, otra parte puede estar alojada en o ser parte del procesador de aplicaciones 612, mientras que una o más partes del mecanismo de reconocimiento y seguridad 610 pueden estar alojadas en o ser parte del sistema operativo 606 y/o cualquier número y tipo de dispositivos del dispositivo informático 600. Se contempla que una o más partes o componentes del mecanismo de reconocimiento y seguridad 610 puedan emplearse como hardware, software y/o firmware.

Se contempla que las realizaciones no están limitados a ninguna implementación o alojamiento particular del mecanismo de reconocimiento y seguridad 610 y que el mecanismo de reconocimiento y seguridad 610 y uno o más de sus componentes pueden implementarse como hardware, software, firmware o cualquier combinación de los mismos.

El dispositivo informático 600 puede alojar una(s) interfaz/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, Internet, etc. La(s) interfaz/interfaces de red pueden incluir, por ejemplo, una interfaz de red inalámbrica que tiene una antena, que puede representar una o más antenas. La(s) interfaz/interfaces de red también pueden incluir, por ejemplo, una interfaz de red alámbrica para comunicarse con dispositivos remotos por medio de un cable de red, que puede ser, por ejemplo, un cable Ethernet, un cable coaxial, un cable de fibra óptica, un cable serie o un cable paralelo.

Se pueden proporcionar realizaciones, por ejemplo, como un producto de programa informático que puede incluir uno o más medios legibles por máquina que tienen almacenados en los mismos instrucciones ejecutables por máquina que, cuando se ejecutan por una o más máquinas tales como un ordenador, una red de ordenadores u otros dispositivos electrónicos, pueden dar como resultado que las una o más máquinas lleven a cabo operaciones de acuerdo con las realizaciones descritas en el presente documento. Un medio legible por máquina puede incluir, pero sin limitación, disquetes, discos ópticos, CD-ROM (memorias de sólo lectura en disco compacto) y discos magnetoópticos, ROM, RAM, EPROM (memorias de sólo lectura programables y borrables), EEPROM (memorias de sólo lectura programables y borrables eléctricamente), tarjetas magnéticas u ópticas, memoria flash u otro tipo de soporte/medio legible por máquina adecuado para almacenar instrucciones ejecutables por máquina.

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 a través de un enlace de comunicación (por ejemplo, un módem y/o conexión de red).

A través de todo del documento, el término "usuario" puede denominarse de manera intercambiable "espectador", "observador", "persona", "individuo", "usuario final" y/o similares. Cabe señalar que, a lo largo de todo este documento, se puede hacer referencia a términos como "dominio de gráficos" de manera intercambiable con "unidad de procesamiento de gráficos", "procesador de gráficos" o simplemente "GPU" y, de manera similar, "dominio de CPU" o "dominio de anfitrión" se pueden hacer referencia de manera intercambiable a "unidad de procesamiento de ordenador", "procesador de aplicaciones" o simplemente "CPU".

Cabe señalar que, términos y expresiones 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 de manera intercambiable en este documento. Cabe señalar además que, términos como "aplicación", "aplicación de software", "programa", "programa de software", "paquete", "paquete de software" y similares, pueden usarse de manera intercambiable a lo largo de todo este documento. Además, términos como "trabajo", "entrada", "solicitud", "mensaje" y similares se pueden usar de manera intercambiable en este documento.

La **Figura 7** ilustra el mecanismo de reconocimiento y seguridad 610 de la **Figura 6** de acuerdo con una realización. Para abreviar, muchos de los detalles ya analizados con referencia a las **Figuras 1-6** no se repiten ni se analizan a continuación. En una realización, el mecanismo de reconocimiento y seguridad 610 puede incluir cualquier número y tipo de componentes, tales como (sin limitaciones): lógica de detección/observación 701; motor de reconocimiento y reidentificación 703 que incluye lógica de reconocimiento y registro 711, lógica de extracción y comparación 713, lógica de reidentificación y modelo 715 y lógica de almacenamiento y entrenamiento 717; lógica de autenticación 705; lógica de comunicación/compatibilidad 707; lógica de ejecución paralela 709; y lógica de comparación de salida 711.

Como se ha mencionado anteriormente, las técnicas de reconocimiento de personas convencionales se basan simplemente en sistemas basados en visión que se basan principalmente en rasgos faciales, conocidos como reconocimiento facial. Naturalmente, depender solo en la cara no proporciona un perfil total de la persona que se está reconociendo o identificando. Por ejemplo, considerando una familia en casa, independientemente del número y tipo de cámaras instaladas en la casa, siempre se espera que haya algunos obstáculos u obstrucciones, que pueden hacer que las personas sean irreconocibles o inidentificables si la confianza se basa completamente en el reconocimiento facial.

Las realizaciones prevén una técnica novedosa que permite el reconocimiento de personas basándose en características visuales de sus cuerpos usando, por ejemplo, algoritmos de visión informática y aprendizaje profundo. Por ejemplo, esta técnica novedosa prevé 1) reconocimiento de personas usando el cuerpo y más allá dependiendo simplemente de la cara; 2) el reconocimiento en presencia de amplias variantes de posturas humanas y ocultaciones corporales; y 3) el uso de hardware de cámara especializado, tal como cámaras IR o térmicas de las fuentes de E/S 604 de la **Figura 6**.

Se contempla que las realizaciones no se limiten a ningún dispositivo particular, tal como un tipo de cámara, sensores, lentes, etc. De manera similar, las realizaciones no se limitan a ninguna implementación o aplicación particular y pueden aplicarse a cualquier número y tipo de escenarios, tal como hogares inteligentes, oficinas inteligentes, asistentes domésticos inteligentes, asistentes personales, aplicaciones de vigilancia y seguridad personal o comercial, etc., en escenarios tanto de interiores como de exteriores.

Las realizaciones prevén una técnica novedosa para ofrecer un enfoque de autoentrenamiento para construir modelos corporales singulares y distinguibles de una persona a lo largo de un período de tiempo según es facilitado por el motor de reconocimiento 703. Por ejemplo, en el despliegue inicial, tal como en un entorno de tipo hogar familiar, la lógica de detección/observación 701 puede activarse en primer lugar para detectar u observar a una persona o cualquier movimiento, gesto, etc., asociado con la persona capturada por uno o más dispositivos de captura de imagen/vídeo, tales como una o más cámaras (por ejemplo, cámaras de IR, cámaras térmicas, cámaras de detección de profundidad, etc.) de las fuentes de E/S 604 de la **Figura 6**.

Tras la detección inicial, en una realización, la lógica de reconocimiento y registro 711 puede activarse para registrar la cara de la persona detectada por la lógica de detección/observación 701 y continuar registrando o inscribiendo la cara cada vez que se detecta o se reconoce. Esto permite que la cara se registre y se almacene en una o más bases de datos 730 para usarse a continuación para un análisis y un procesamiento adicionales.

En una realización, cada vez o como sea que la persona es visible para una cámara según es facilitado por la lógica de detección/observación 701, la cara es reconocida y registrada por la lógica de reconocimiento y registro 711 y, posteriormente, la lógica de extracción y comparación 713 se activa para extraer cualquier información relevante, tal como el recuadro delimitador, de la cara y el cuerpo de la persona. Por ejemplo, se contempla que no siempre el cuerpo completo de la persona pueda ser capturado por la cámara y, por lo tanto, la lógica de extracción y comparación 713 puede usarse para extraer la información relevante acerca del perfilado de la persona a partir de una cualquiera o más partes del cuerpo de la persona (por ejemplo, hombros, espalda, costado, piernas, etc.) que son visibles para la cámara y que son registradas por la lógica de reconocimiento y registro 711.

Por ejemplo, si solo los hombros son capturados por la cámara, la lógica de extracción y comparación 713 puede usarse para extraer características visuales de los hombros de la persona usando una red neural convolucional (CNN) preentrenada y activar la lógica de almacenamiento y entrenamiento 717 para almacenar el vector de características relacionado con la persona como galería de características en una o más bases de datos 730 a través de uno o más medios de comunicación 725, tales como una red en la nube, una red de proximidad, Internet, etc.

Se contempla adicionalmente que, en algunas realizaciones, el cuerpo de la persona puede ser visible para la cámara, pero la cara no es visible por cualquier número de razones, tal como debido a cierta ocultación, de forma deliberada o no deliberada. En tales realizaciones, la lógica de extracción y comparación 713 aún puede activarse para usar el recuadro delimitador de la persona para extraer características corporales de la parte del cuerpo o de todo el cuerpo que es visible para la cámara y estas características se comparan a continuación con el vector de características anteriormente mencionado almacenado en la galería de características. En una realización, esta comparación entre las características corporales extraídas y el vector de características almacenado previamente puede realizarse usando una métrica de similitud, donde, por ejemplo, cualquier correspondencia que sea la más cercana de incluso uno de los vectores de características sondeados o extraídos con al menos una en la galería de características puede contribuir a la identidad resultante de la persona. Esto se ilustra y se analiza adicionalmente con referencia a las **figuras 8A**, **8B** y **8C**.

En una realización, estos procesos de detección, reconocimiento, registro, extracción, almacenamiento, etc., pueden continuar con cualquier cantidad y tipo de partes del cuerpo de la persona (por ejemplo, ojos, nariz, pies, brazos, torso, pecho, etc.) a lo largo de un período de tiempo para que la lógica de reidentificación y modelo 715 genere, actualice y/o use continuamente un modelo, tal como un modelo de clasificación, de la persona que se basa cada vez menos en la cara de la persona para la identificación/reidentificación. Por ejemplo, tras capturar una cierta cantidad de datos a lo largo de un cierto período de tiempo, el modelo creado por la lógica de reidentificación y modelo 715 puede alcanzar un punto con un perfilado suficiente de la persona con el que puede reconocerse o identificarse a la persona sin necesidad del reconocimiento facial de la persona y de forma completamente independiente del mismo. Este modelo de clasificación puede usarse a continuación para la clasificación de la persona y, además, mediante la lógica de almacenamiento y entrenamiento 717, para fines de entrenamiento, tales como vectores de características corporales de la persona como se establece en el modelo, puede usarse para entrenar un clasificador para el reconocimiento de personas que es completamente independiente del reconocimiento facial o las características del usuario.

Las realizaciones tienen conocimiento del escenario del mundo real donde los perfiles faciales no siempre son visibles para las cámaras, tal como en casas pequeñas o cuando se trabaja con aplicaciones de seguridad y vigilancia. Esta técnica novedosa prevé una dependencia baja del reconocimiento facial y una dependencia alta de características basadas en el cuerpo más prácticas. Además, en contraposición a necesitar cámaras de profundidad para unas estimaciones de ubicaciones de articulaciones esqueléticas tridimensionales (3D) precisas, las realizaciones prevén basarse en datos de color, tales como rojo verde azul (RGB), del cuerpo humano sin tener que requerir ningún hardware de cámara específico, tal como cámaras de detección de profundidad, etc.

Las realizaciones prevén adicionalmente una autenticación basada en capas y una comprobación de integridad para redes neuronales (NN) en máquinas autónomas según es facilitado por la lógica de autenticación 705. Las redes neuronales profundas (DNN) convencionales no emplean autenticación o verificación de integridad. Las realizaciones prevén la adición de esta capacidad novedosa para la autenticación y verificación de integridad a redes neuronales (tales como DNN) de modo que puede detectarse cualquier atacante malicioso que intente ofuscar y cambiar una salida de red neuronal, y evitarse que haga ningún daño.

A medida que las redes neuronales se van usando cada vez más en sistemas críticos para la seguridad, el atractivo para que agentes maliciosos ataquen estas redes también va en aumento. Por ejemplo, ha habido una tendencia a recomendar el uso de redes neuronales para la toma de decisiones y la estrategia de conducción para vehículos autónomos, tal como la máquina/vehículo autónomo 600. Por lo tanto, se contempla el nivel de daño que un atacante podría hacer si inyectara parámetros o datos falsos en una red o sistema que provocaría que un vehículo autónomo o bien identifique erróneamente a un peatón (y golpee a esa persona) o bien haga que la decisión errónea se emita desde una red neuronal que está tomando decisiones de planificación de ruta a través de un entorno urbano complejo.

En una realización, la lógica de autenticación 705 se proporciona para mejorar la construcción de red neuronal básica añadiendo verificación de integridad como parte del propio modelo de modo que, en cada capa de la red, una etapa de verificación garantiza que no se han inyectado datos maliciosos en los cálculos de inferencia. En una realización, esta verificación puede adoptar una diversidad de formas, incluyendo comprobaciones de redundancia cíclica (CRC), o puede tomar prestados conceptos de memoria de código de corrección de errores (ECC) que es capaz de corregir automáticamente errores pequeños u optar por algo más robusto, tal como con una propagación independiente hacia delante y hacia atrás con una comparación de los resultados para garantizar la coherencia. Esto se ilustra con referencia a la **Figura 9A**.

Las realizaciones prevén adicionalmente redes neuronales de ejecución paralela (por ejemplo, DNN) con aislamiento de ejecución según es facilitado por la lógica de ejecución paralela 709. Por ejemplo, pueden planificarse múltiples DNN en una GPU, tal como el procesador de gráficos 614, donde todas las DNN de este tipo deben asegurarse contra atacantes maliciosos que intentan cambiar pesos que podrían dar como resultado diferentes salidas de DNN. Por ejemplo, en una realización, la lógica de ejecución paralela 709 se usa para habilitar que múltiples DNN se ejecuten en una única GPU, tal como el procesador de gráficos 614, mientras se evita que una DNN cualquiera interfiera con una nueva DNN o con otra DNN.

Se contempla que los ataques de canal lateral son una forma singular de ataque que puede intentar recabar información inherente en la implementación de un algoritmo o sistema particular. Por ejemplo, considérese una red neuronal que se ejecuta en una GPU de propósito general (GPGPU) que tiene cientos, si no miles, de unidades de ejecución individuales, por lo que, dentro de una única GPGPU, pueden emplearse y estar ejecutándose múltiples aplicaciones diferentes. Una aplicación puede ser una red neuronal, mientras que otra es una aplicación de gráficos aparentemente simple, que típicamente es la aplicación objetivo para atacantes potenciales. Así es frecuentemente como los atacantes obtienen información privada y segura de plataformas informáticas, tal como aprovechando vulnerabilidades en el sistema operativo o en las estructuras de aplicación que permitirían a un atacante acceder a memoria y datos desde otras aplicaciones.

En una realización, la lógica de ejecución paralela 709 permite barreras de protección flexibles y programables (o fijas) en torno a diferentes unidades de ejecución con una GPGPU, tal como el procesador de gráficos 614, por lo que puede no haber ningún desbordamiento o posibilidad de que los atacantes vean información residual o ninguna otra acerca de operaciones de inferencia de red neuronal segura. Esto se ilustra con referencia a la **Figura 9B**.

Las realizaciones prevén adicionalmente la detección de anomalías de sistema usando una comparación de salida de red neuronal, tal como DNN, según es facilitado por la lógica de comparación de salida 711.

Se contempla que, si una salida indica una activación inesperada, tal como identificar a una persona delante de un vehículo autónomo sin tomar una decisión de frenar, entonces, en tales circunstancias, el sistema puede verse comprometido. Un objetivo de ataque probable para las entidades maliciosas es intentar afectar a las decisiones tomadas en algoritmos de toma de decisiones que se encuentran comúnmente en tipos de bucle de control en tiempo real de aplicaciones tales como robots industriales, dispositivos médicos o vehículos autónomos.

Además, los "fallos sistemáticos", que son esencialmente errores lógicos (tales como "*bugs*" (errores de programación)) introducidos por la complejidad de estos sistemas que pueden extenderse a millones de líneas de código donde es probable que se introduzcan errores con el potencial de tener consecuencias catastróficas. Irónicamente, en los sistemas de seguridad, las redes neuronales no se ven a menudo como siempre "fiables" debido a la naturaleza probabilística de su diseño. Sin embargo, si una red neuronal debidamente entrenada está notificando un escenario, tal como con un 99,99 % de confianza, de que el objeto delante de un coche es una persona, entonces esa información puede usarse para fines de toma de decisiones inteligente. En las técnicas convencionales, esta salida de red neuronal se descarta por completo bastante temprano en la canalización y no es susceptible de usarse como una contracomprobación de errores sistemáticos o ataques maliciosos potenciales en algoritmos de toma de decisiones.

En una realización, la lógica de comparación de salida 711 se usa para crear un nuevo enlace entre cualquier salida de una red neuronal (tal como una detección y un clasificador de objetos) y un algoritmo de toma de decisiones. Por ejemplo, en la máquina/vehículo autónomo 600, una salida de red neuronal sugiere que un objeto delante del vehículo 600 es un ser humano con un 99,9 % de certeza, entonces la lógica de comparación de salida 711 puede activarse para tener en cuenta esta salida y, en lugar de descartarla, usar la misma para comprobar cualquier decisión o tarea potencial en la que esté trabajando el algoritmo de toma de decisiones del vehículo 600.

Por ejemplo, si una decisión pendiente es acelerar el vehículo 600, esta decisión puede alterarse o cancelarse por completo dado que la información obtenida de la salida es que el objeto es un ser humano, que puede estar en peligro si el vehículo 600 acelera. En otras palabras, usando el conocimiento de salida, se puede dar instrucciones al algoritmo de toma de decisiones para o bien que no acelere o bien que acelere, pero que se mantenga apartado del objeto, de modo que no se cause dañe al objeto, muy probablemente una persona. Esto se ilustra con referencia a la **Figura 9C**.

Además, la lógica de comunicación/compatibilidad 707 se puede usar para facilitar la comunicación y compatibilidad necesarias entre cualquier número de dispositivos del dispositivo informático 600 y diversos componentes del mecanismo de reconocimiento y seguridad 610.

La lógica de comunicación/compatibilidad 707 puede usarse para facilitar la comunicación dinámica y la compatibilidad entre el dispositivo informático 600 y cualquier número y tipo de otros dispositivos informáticos (tales como dispositivo informático móvil, ordenador de escritorio, dispositivo informático de servidor, etc.); dispositivos o componentes de procesamiento (tales como CPU, GPU, etc.); dispositivos de captura/localización/detección (tales como componentes de captura/detección que incluyen cámaras, cámaras de detección de profundidad, sensores de cámara, sensores de rojo, verde, azul ("RGB" o "rgb"), micrófonos, etc.); dispositivos de visualización (tales como componentes de salida, que incluyen pantallas de visualización, áreas de visualización, proyectores de visualización, etc.); componentes de reconocimiento de usuario/contexto y/o sensores/dispositivos de identificación/verificación (tales como sensores/detectores biométricos, escáneres, etc.); base(s) de datos 730, tales como memoria o dispositivos de almacenamiento, bases de datos y/o fuentes de datos (tales como dispositivos de almacenamiento de datos, discos duros, unidades de estado sólido, discos duros, tarjetas o dispositivos de memoria, circuitos de memoria, etc.); medio(s) de comunicación 725, tales como uno o más canales o redes de comunicación (por ejemplo, redes en la nube, Internet, intranets, redes celulares, redes de proximidad, tales como Bluetooth, Bluetooth de baja energía (BLE), Bluetooth inteligente, Wi-Fi de proximidad, identificación por radiofrecuencia (RFID), comunicación de campo cercano

(NFC), red de área corporal (BAN), etc.); comunicaciones inalámbricas o alámbricas y protocolos pertinentes (por ejemplo, Wi-Fi®, WiMAX, Ethernet, etc.); técnicas de conectividad y gestión de ubicación; aplicaciones de software/sitios web (por ejemplo, sitios web de redes sociales y/o comerciales, etc., aplicaciones comerciales, juegos y otras aplicaciones de entretenimiento, etc.); y lenguajes de programación, etc., garantizando al mismo tiempo la compatibilidad con tecnologías, parámetros, protocolos, normas, etc., cambiantes.

Además, cualquier uso de una marca, palabra, término, expresión, nombre y/o acrónimo particular, tal como "detectar", "observar", "reconocer", "registrar", "reidentificar", "reconocimiento facial", "perfil corporal", "modelo de clasificación", "red neuronal", "en cuarentena", "protegido", "conjunto de entrenamiento", "máquina autónoma", "agente", "máquina", "vehículo", "robot", "conducción", "CNN", "DNN", "NN", "unidad de ejecución", "EU", "memoria local compartida", "SLM", "flujos de gráficos", "caché", "caché de gráficos", "GPU", "procesador de gráficos", "dominio de GPU", "GPGPU", "CPU", "procesador de aplicaciones", "dominio de CPU", "controlador de gráficos", "carga de trabajo", "aplicación", "canalización de gráficos", "procesos de canalización", "API", "API 3D", "OpenGL®", "DirectX®", "hardware", "software", "agente", "controlador de gráficos", "controlador de gráficos en modo de núcleo", "controlador en modo de usuario", "estructura de controlador en modo de usuario", "memoria intermedia", "memoria intermedia de gráficos", "tarea", "proceso", "operación", "aplicación de software", "juego", etc., no deben interpretarse como limitantes de las realizaciones al software o dispositivos que llevan esa etiqueta en productos o en bibliografía externa a este documento.

Se contempla que se puede añadir y/o eliminar cualquier número y tipo de componentes del mecanismo de reconocimiento y seguridad 610 para facilitar diversas realizaciones que incluyen añadir, eliminar y/o mejorar ciertas características. Por brevedad, claridad y facilidad de comprensión del mecanismo de reconocimiento y seguridad 610, muchos de los componentes convencionales y/o conocidos, tales como los de un dispositivo informático, no se muestran ni analizan en el presente caso. Se contempla que las realizaciones, como se describen en el presente documento, no se limiten a ninguna tecnología, topología, sistema, arquitectura y/o norma particular y sean lo suficientemente dinámicas para adoptar y adaptarse a cualquier cambio futuro.

La **Figura 8A** ilustra una secuencia de transacciones 800 para el reconocimiento de personas de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-7** pueden no analizarse o repetirse posteriormente en el presente caso. Cualquier proceso relacionado con la secuencia de transacciones 800 puede realizarse mediante lógica de procesamiento que puede comprender hardware (por ejemplo, circuitería, lógica especializada, lógica programable, etc.), software (tal como instrucciones ejecutadas en un dispositivo de procesamiento) o una combinación de los mismos, según es facilitado por el mecanismo de reconocimiento y seguridad 610 de la **Figura 6**. Los procesos asociados con la secuencia de transacciones 800 pueden ilustrarse o indicarse en secuencias lineales para mayor brevedad y claridad en la presentación; sin embargo, se contempla que cualquier número de ellos pueda realizarse en paralelo, de forma asincrónica o en diferentes órdenes.

Como una cuestión inicial, para el modelo de reconocimiento de cuerpos de persona de múltiples clases, se puede usar una diversidad de modelos de CNN para recopilar y usar un gran número de imágenes corporales de una persona para construir un modelo de clasificación para ayudar a clasificar entre miembros del hogar (tal como para casos de uso de hogares inteligentes). Si las imágenes tienen una buena variabilidad de postura, ocultación y condición de iluminación, es probable que el modelo de clasificación sea mucho más robusto.

La secuencia de transacciones 800 comienza con la recepción de los fotogramas de vídeo 801 en el motor de reconocimiento de personas 803, donde los fotogramas de vídeo 801 pueden ser capturados por una cámara y detectados por la lógica de detección/observación 701 de la **Figura 7**, mientras que el motor de reconocimiento 803 puede ser el mismo que el de, o facilitarse por, la lógica de reconocimiento y registro 711 de la **Figura 7**. Como se describe adicionalmente con referencia a **Figura 7**, una persona en los fotogramas de vídeo 801 se detecta a través del detector de personas 805 según es facilitado por la lógica de reconocimiento y registro 711. En el bloque 807, se realiza una determinación en cuanto a si está presente un reconocedor de cuerpos de modo que puede reconocerse el cuerpo, o si hay suficiente información relacionada con el cuerpo para reconocer el cuerpo.

Si no, en el bloque 811, se realiza otra determinación en cuanto a si una cara se detecta suficientemente. Si no, la secuencia de transacciones 800 continúa con el reidentificador de personas 813, donde se envía cualquier información a la galería de reidentificación 815, que puede ser parte de una o más bases de datos 730 de la **Figura 7**, y cualquier información relevante, tal como el nombre de la persona, se comunica al motor de reconocimiento de personas 803. Si la cara de la persona se detecta en el bloque 811, entonces la información puede enviarse al reconocedor de caras 827 en comunicación con el almacén de datos de plantillas corporales 819 y el modelo de extractor de características de imagen corporal 817. Como se ha analizado anteriormente, la lógica de extracción y comparación 713 de la **Figura 7** puede usarse para extraer la imagen corporal usando el detector de personas 805 y la etiqueta de persona usando el reconocedor de caras 827 que puede almacenarse a continuación como parte del modelo de extractor de características de imagen corporal 817 en una o más bases de datos 730 de la **Figura 7**. De manera similar, en una realización, la imagen corporal se extrae usando el detector de personas 805 y la etiqueta de persona usando el reconocedor de caras 827 y se comunica y se almacena en el almacén de datos de plantillas corporales 819 en una o más bases de datos 730 de la **Figura 7**.

Haciendo referencia de nuevo al bloque 807, en una realización, si el reconocedor de cuerpos 821 está presente, puede usarse entonces para reconocer el cuerpo o una o más partes del cuerpo visibles para la cámara. En una realización, cualquier información reconocida a través del reconocedor de cuerpos 821 puede comunicarse para crear y ser parte del modelo de reconocimiento de cuerpos de persona de múltiples clases 823, que también puede usarse, en comunicación con el almacén de datos de plantillas corporales 819, para entrenar el modelo cuando se recopila una muestra de datos suficiente para el usuario. En una realización, los modelos 817, 823 pueden generarse y rellenarse usando la lógica de reidentificación y modelo 715 de la **Figura 7**. Cualquier información relevante, tal como el nombre de la persona, se comunica de vuelta al motor de reconocimiento de personas 803.

La **Figura 8B** ilustra un método 830 para la reidentificación de personas de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-8A** pueden no analizarse o repetirse posteriormente en el presente caso. Cualquier proceso relacionado con el método 830 puede realizarse mediante lógica de procesamiento que puede comprender hardware (por ejemplo, circuitería, lógica especializada, lógica programable, etc.), software (tal como instrucciones ejecutadas en un dispositivo de procesamiento) o una combinación de los mismos, según es facilitado por el mecanismo de reconocimiento y seguridad 610 de la **Figura 6**. Los procesos asociados con el método 830 pueden ilustrarse o indicarse en secuencias lineales para mayor brevedad y claridad en la presentación; sin embargo, se contempla que cualquier número de ellos pueda realizarse en paralelo, de forma asincrónica o en diferentes órdenes.

El método 830 comienza con información relacionada con múltiples personas, tal como una imagen de persona reconocida de la persona 1 831, la persona 2 833 y la persona N 835, que se reciben y se registran en el reidentificador de personas 837 según es facilitado por la lógica de reconocimiento y registro 711 y la lógica de reidentificación y modelo 715 de la **Figura 7**. Cualquier información de recuadro delimitador puede comunicarse desde el reidentificador de personas 837 a la NN preentrenada 841 en el caffe 839 que puede usarse a continuación para generar los descriptores 843 que se envían a continuación a la galería 845 que es una o más de la base de datos 730 de la **Figura 7**.

Al igual que con las imágenes relacionadas con las personas reconocidas 831, 833, 835, en una realización, la imagen de persona de la persona detectada, tal como la persona x 849, puede recibirse y consultarse en el reidentificador de personas 837, que a continuación proporciona cualquier información de recuadro delimitador para la CNN preentrenada 853 en el caffe 851, generando los descriptores 855. En una realización, los descriptores 855 y la información obtenida de la galería 845 se reciben en la unidad de puesta en correspondencia 847 donde se realiza una comparación entre los dos conjuntos de información según es facilitado por la lógica de extracción y comparación 713 de la Figura 7 para determinar si hay una correspondencia entre la imagen de persona detectada de la persona x 849 y una o más de las imágenes de persona reconocida de las personas 1 831, 2 833 y N 835.

La **Figura 8C** ilustra una secuencia de transacciones 860 para la puesta en correspondencia de personas de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-8B** pueden no analizarse o repetirse posteriormente en el presente caso. Cualquier proceso relacionado con la secuencia de transacciones 860 puede realizarse mediante lógica de procesamiento que puede comprender hardware (por ejemplo, circuitería, lógica especializada, lógica programable, etc.), software (tal como instrucciones ejecutadas en un dispositivo de procesamiento) o una combinación de los mismos, según es facilitado por el mecanismo de reconocimiento y seguridad 610 de la **Figura 6**. Los procesos asociados con la secuencia de transacciones 860 pueden ilustrarse o indicarse en secuencias lineales para mayor brevedad y claridad en la presentación; sin embargo, se contempla que cualquier número de ellos pueda realizarse en paralelo, de forma asincrónica o en diferentes órdenes.

Como se ha analizado con referencia a la **Figura 8B**, los descriptores de características 861 (por ejemplo, la gente en un reconocimiento de personas de cuerpo completo en un álbum de fotos (conjunto de datos PIPA), etc.) pueden recibirse y almacenarse en el modelo de descriptor de características 863 según es facilitado por la lógica de reidentificación y modelo 715 de la **Figura 7**. De manera similar, en una realización, se accede a información relevante a partir de los conjuntos de datos de reidentificación de personas 873 (tales como el conjunto de datos 1, el conjunto de datos 2, el conjunto de datos N, etc.) para determinar si cualquiera de las imágenes es una correspondencia, tal como pares iguales o diferentes de imágenes, tal como que unas imágenes iguales 875 se consideren una correspondencia o que unas imágenes diferentes 877 se consideren una falta de correspondencia. Esta información se comunica al modelo de descriptor de características 863.

En una realización, el vector de características 865 se extrae del modelo de descriptor de características 863 y se usa para calcular una diferencia absoluta normalizada y añadir unas etiquetas de iguales o diferentes a los hallazgos en 867 que a continuación se convierten en la máquina de vectores de soporte (SVM) lineal 869. La SVM lineal 869, que es un modelo de aprendizaje de supervisión con algoritmos de aprendizaje correspondientes para analizar datos para la clasificación, el análisis de regresión, etc., puede usarse para generar el modelo de clasificación 871, que tiene información que identifica las mismas y diferentes imágenes según es facilitado por la lógica de reidentificación y modelo 715 de la **Figura 7**.

La **Figura 9A** ilustra un modelo 900 para comprobaciones de autenticación y verificación para redes neuronales en el aprendizaje automático de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-8C** pueden no analizarse o repetirse posteriormente en el presente caso. Además, las realizaciones no se limitan a ninguna ubicación, estructura, configuración o estructura arquitectónica particular de procesos y/o componentes, tal como el modelo 900.

Como se ilustra en el presente caso y se ha analizado anteriormente con referencia **a** la **Figura 7**, la lógica de autenticación 705 puede usarse para mejorar y garantizar una construcción de red neuronal básica añadiendo verificación de integridad como parte del propio modelo 900 de modo que, en cada capa de la red, hay una etapa de verificación para garantizar que no puede inyectarse ningún dato malicioso en los cálculos de inferencia. Como se ilustra, en una realización, esta verificación puede adoptar una diversidad de formas, tales como doble vuelta a lo largo de múltiples capas del modelo 900, y puede usarse para calcular la CRC 901, 903, 905, 907, 909 o una evidencia de verificación de integridad similar en cada capa correspondiente para una robustez aumentada. En otra realización, pueden usarse testigos de seguridad para pasar a través de cada capa que a continuación puede verificarse al final.

La **Figura 9B** ilustra una estructura 920 para la ejecución paralela de redes neuronales en el aprendizaje automático de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-9A** pueden no analizarse o repetirse posteriormente en el presente caso. Además, las realizaciones no se limitan a ninguna ubicación, estructura, configuración o estructura arquitectónica particular de procesos y/o componentes, tal como la estructura 920.

Como se ilustra en la estructura 920, en una realización, la lógica de ejecución paralela 709 de la **Figura 7** permite barreras de protección flexibles y programables (o fijas) en torno a diferentes unidades de ejecución con una GPGPU, tal como el procesador de gráficos 614, por lo que puede no haber ningún desbordamiento o posibilidad de que los atacantes vean información residual o ninguna otra acerca de operaciones de inferencia de red neuronal segura.

Por ejemplo, como se ilustra, esta protección según se introduce usando la lógica de ejecución paralela 709 de la **Figura 7** también puede extenderse a la memoria de GPU a bordo 921, donde, en una realización, los chips de memoria y el controlador de memoria inteligente pueden estar físicamente separados. Como se ilustra adicionalmente, una máquina autónoma, tal como la máquina autónoma 600 de la **Figura 6**, puede estar alojando y ejecutando una o más aplicaciones de gráficos, tales como la aplicación de gráficos 925 que está en cuarentena, junto con una o más redes neuronales, tales como las redes neuronales 923, 929 que se protegen de cualquier ataque potencial en la aplicación de gráficos 925, mientras que algunas unidades de ejecución 927 permanecen sin usar.

En una realización, pueden soportarse diferentes niveles o tipos de protección a través de esta técnica novedosa y también puede aplicarse una producción inversa, tal como la cuarentena de aplicaciones potencialmente no fiables.

La **Figura 9C** ilustra una secuencia de transacciones 950 para usar una salida de red neuronal para alterar o confirmar una decisión pendiente en máquinas autónomas de acuerdo con una realización. Para abreviar, muchos de los detalles previamente analizados con referencia a las **Figuras 1-9B** pueden no analizarse o repetirse posteriormente en el presente caso. Además, las realizaciones no se limitan a ninguna ubicación, estructura, configuración o estructura arquitectónica particular de procesos y/o componentes, tal como la secuencia de transacciones 950.

En una realización, la cámara 951, tal como una cámara de las fuentes de E/S 604 de la **Figura 6**, se usa para capturar una o más imágenes (por ejemplo, imágenes fijas, vídeos, etc.) de objetos dentro de las proximidades de un vehículo autónomo, tal como la máquina autónoma 600 de la **Figura 6**. Como se ilustra adicionalmente, el modelo de cámara 953 puede usarse para proporcionar una salida de red neuronal de que lo más probable es que el objeto capturado por la cámara 951 sea una persona, tal como con una precisión del 99,9 %. En este caso, usando la lógica de comparación de salida 711 de la **Figura 7**, esta salida puede usarse a continuación para comparar con cualquier decisión pendiente o venidera mediante un algoritmo de toma de decisiones, tal como una planificación de ruta potencial u otra toma de decisiones, etc., en 957.

Ahora, por ejemplo, algún atacante malicioso 961 puede intentar acceder al sistema y permitir que este tome una mala decisión, tal como acelerar y dirigirse directamente 963 hacia la persona. Sin embargo, en una realización, esto se evita usando la salida 955 en conjunto con el algoritmo de toma de decisiones o bien para alterar o bien para suspender la decisión venidera en 959, tal como detenerse por completo o dirigirse lejos de la persona o simplemente evitar la aceleración (tal como si la persona está lo suficientemente lejos) y, a su vez, evitar malas influencias de los bloques 961 y 963.

Vista general de aprendizaje automático

Un algoritmo de aprendizaje automático es un algoritmo que puede aprender basándose en un conjunto de datos. Pueden diseñarse realizaciones de algoritmos de aprendizaje automático para modelar abstracciones de alto nivel dentro de un conjunto de datos. Por ejemplo, pueden usarse algoritmos de reconocimiento de imágenes para determinar a cuál de varias categorías pertenece una entrada dada; los algoritmos de regresión pueden emitir un valor

numérico dada una entrada; y pueden usarse los algoritmos de reconocimiento de patrones para generar texto traducido o para realizar texto a habla y/o reconocimiento de habla.

Un tipo ilustrativo de algoritmo de aprendizaje automático es una red neuronal. Hay muchos tipos de redes neuronales; un tipo sencillo de red neuronal es una red de realimentación prospectiva. Una red de realimentación prospectiva puede implementarse como un grafo acíclico en el que los nodos se disponen en capas. Habitualmente, una topología de red de realimentación prospectiva incluye una capa de entrada y una capa de salida que están separadas por al menos una capa oculta. La capa oculta transforma la entrada recibida por la capa de entrada en una representación que es útil para generar la salida en la capa de salida. Los nodos de red están completamente conectados mediante bordes a los nodos en capas adyacentes, pero no hay bordes entre nodos dentro de cada capa. Los datos recibidos en los nodos de una capa de entrada de una red de realimentación prospectiva se propagan (es decir, "se realimentan prospectivamente") 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 ("pesos") asociados, respectivamente, con cada uno de los bordes que conectan las capas. Dependiendo del modelo específico que está siendo representado por el algoritmo que se está ejecutando, la salida desde el algoritmo de la red neuronal puede adoptar 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, usar un conjunto de datos de entrenamiento que representa un problema que es modelado por la red, y ajustar los pesos hasta que el modelo de red rinde 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 se ajustan los pesos asociados con las conexiones para minimizar ese error a medida que la señal de error se retropropaga 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 a partir de 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 usado para entrenar el algoritmo. El proceso de entrenamiento puede ser computacionalmente intensivo y puede requerir una cantidad de tiempo significativa en un procesador de propósito general convencional. En consecuencia, se usa hardware de procesamiento paralelo para entrenar muchos tipos de algoritmos de aprendizaje automático. Esto es particularmente útil para optimizar el entrenamiento de redes neuronales, debido a que los cálculos realizados en el ajuste de los coeficientes en redes neuronales se prestan de manera natural a implementaciones paralelas. Específicamente, muchos algoritmos de aprendizaje automático y aplicaciones de software se han adaptado para hacer uso del hardware de procesamiento paralelo dentro de dispositivos de procesamiento de gráficos de propósito general.

La **Figura 10** es un diagrama generalizado de una pila de software de aprendizaje automático 1000. Una aplicación de aprendizaje automático 1002 puede configurarse para entrenar una red neuronal usando un conjunto de datos de entrenamiento o para usar una red neuronal profunda entrenada para implementar una inteligencia automática. La aplicación de aprendizaje automático 1002 puede incluir una funcionalidad de entrenamiento y de inferencia para una red neuronal y/o software especializado que puede usarse para entrenar una red neuronal antes del despliegue. La aplicación de aprendizaje automático 1002 puede implementar cualquier tipo de inteligencia automática incluyendo, pero sin limitación, reconocimiento de imágenes, mapeo y localización, navegación autónoma, síntesis de habla, formación de imágenes médicas o traducción de idioma.

Puede posibilitarse una aceleración de hardware para la aplicación de aprendizaje automático 1002 mediante una estructura de aprendizaje automático 1004. La estructura de aprendizaje automático 1004 puede proporcionar una biblioteca de primitivas de aprendizaje automático. Las primitivas de aprendizaje automático son operaciones básicas que se realizan comúnmente por algoritmos de aprendizaje automático. Sin la estructura de aprendizaje automático 1004, se requeriría que los desarrolladores de algoritmos de aprendizaje automático crearan y optimizaran la lógica computacional principal asociada con el algoritmo de aprendizaje automático, y que reoptimizaran, a continuación, la lógica computacional a medida que se desarrollan nuevos procesadores paralelos. En su lugar, la aplicación de aprendizaje automático puede configurarse para realizar los cálculos necesarios usando las primitivas proporcionadas por la estructura de aprendizaje automático 1004. Las primitivas ilustrativas incluyen convoluciones tensoriales, funciones de activación y agrupamiento, que son operaciones computacionales que se realizan mientras se entrena una red neuronal convolucional (CNN). La estructura de aprendizaje automático 1004 puede proporcionar también primitivas para implementar subprogramas de álgebra lineal básicos realizados por muchos algoritmos de aprendizaje automático, tales como operaciones matriciales y vectoriales.

La estructura de aprendizaje automático 1004 puede procesar datos de entrada recibidos desde la aplicación de aprendizaje automático 1002 y generar la entrada apropiada a una estructura de cálculo 1006. La estructura de cálculo 1006 puede abstraer las instrucciones subyacentes proporcionadas al controlador de GPGPU 1008 para posibilitar que la estructura de aprendizaje automático 1004 se aproveche de la aceleración de hardware mediante el hardware de GPGPU 1010 sin requerir que la estructura de aprendizaje automático 1004 tenga un conocimiento íntimo de la

arquitectura del hardware de GPGPU 1010. Adicionalmente, la estructura de cálculo 1006 puede posibilitar la aceleración de hardware para la estructura de aprendizaje automático 1004 a lo largo de una diversidad de tipos y generaciones del hardware de GPGPU 1010.

Aceleración de aprendizaje automático de GPGPU

La **Figura 11** ilustra una unidad de procesamiento de gráficos de propósito general altamente paralela 1100, de acuerdo con una realización. En una realización, la unidad de procesamiento de propósito general (GPGPU) 1100 puede estar configurada para ser particularmente eficiente al procesar el tipo de cargas de trabajo computacionales asociadas con el entrenamiento de las redes neuronales profundas. Adicionalmente, la GPGPU 1100 puede vincularse 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 1100 incluye una interfaz de anfitrión 1102 para posibilitar una conexión con un procesador de anfitrión. En una realización, la interfaz de anfitrión 1102 es una interfaz PCI Express. Sin embargo, la interfaz de anfitrión también puede ser una interfaz de comunicaciones o tejido de comunicaciones específico de proveedor. La GPGPU 1100 recibe comandos desde el procesador de anfitrión y usa un planificador global 1104 para distribuir hilos de ejecución asociados con estos comandos a un conjunto de agrupaciones de cálculo 1106A-H. Las agrupaciones de cálculo 1106A-H comparten una memoria caché 1108. La memoria caché 1108 puede servir como una caché de nivel superior para memorias caché dentro de las agrupaciones de cálculo 1106A-H.

La GPGPU 1100 incluye la memoria 1114A-B acoplada con las agrupaciones de cálculo 1106A-H mediante un conjunto de controladores de memoria 1112A-B. En diversas realizaciones, la memoria 1114A-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 la memoria de acceso aleatorio de gráficos síncrona (SGRAM), que incluye la memoria de tasa de datos doble de gráficos (GDDR). En una realización, las unidades de memoria 224A-N pueden incluir también memoria 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, tal como el multiprocesador de gráficos 400 de la Figura 4A. Los multiprocesadores de gráficos de la agrupación de cálculo tienen múltiples tipos de unidades de lógica de enteros y de coma flotante que pueden realizar operaciones computacionales con un intervalo de precisiones que incluyen unas 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 1106A-H puede estar configurado para realizar operaciones de coma flotante de 16 bits o de 32 bits, mientras que un subconjunto diferente de las unidades de coma flotante puede estar configurado para realizar operaciones de coma flotante de 64 bits.

Múltiples instancias de la GPGPU 1100 pueden configurarse para operar como una agrupación de cálculo. El mecanismo de comunicación usado por la agrupación de cálculo para la sincronización y el intercambio de datos varía a lo largo de las realizaciones. En una realización, las múltiples instancias de la GPGPU 1100 se comunican a través de la interfaz de anfitrión 1102. En una realización, la GPGPU 1100 incluye un concentrador de E/S 1108 que acopla la GPGPU 1100 con un enlace de GPU 1110 que posibilita una conexión directa a otras instancias de la GPGPU. En una realización, el enlace de GPU 1110 está acoplado a un puente de GPU a GPU dedicado que posibilita la comunicación y sincronización entre múltiples instancias de la GPGPU 1100. En una realización, el enlace de GPU 1110 se acopla con una interconexión de alta velocidad para transmitir y recibir datos a otras GPGPU o procesadores paralelos. En una realización, las múltiples instancias de la GPGPU 1100 están ubicadas en sistemas de procesamiento de datos separados y se comunican a través de un dispositivo de red al que puede accederse a través de la interfaz de anfitrión 1102. En una realización, el enlace de GPU 1110 puede estar configurado para posibilitar una conexión a un procesador de anfitrión además de o como una alternativa a la interfaz de anfitrión 1102.

Aunque la configuración ilustrada de la GPGPU 1100 puede configurarse para entrenar redes neuronales, una realización proporciona una configuración alternativa de la GPGPU 1100 que puede configurarse para el despliegue dentro de una plataforma de inferencia de alto rendimiento o de baja potencia. En una configuración de inferencia, la GPGPU 1100 incluye menos de las agrupaciones de cálculo 1106A-H en relación con la configuración de entrenamiento. Adicionalmente, una tecnología de memoria asociada con la memoria 1114A-B puede diferir entre configuraciones de inferencia y de entrenamiento. En una realización, la configuración de inferencia de la GPGPU 1100 puede soportar instrucciones específicas de inferencia. Por ejemplo, una configuración de inferencia puede proporcionar soporte para una o más instrucciones de producto escalar de números enteros de 8 bits, que se usan comúnmente durante las operaciones de inferencia para redes neuronales desplegadas.

La **Figura 12** ilustra un sistema informático de múltiples GPU 1200, de acuerdo con una realización. El sistema informático de múltiples GPU 1200 puede incluir un procesador 1202 acoplado a múltiples GPGPU 1206A-D mediante un conmutador de interfaz de anfitrión 1204. El conmutador de interfaz de anfitrión 1204, en una realización, es un dispositivo de conmutador de PCI Express que acopla el procesador 1202 a un bus de PCI Express a través del que el procesador 1202 puede comunicarse con el conjunto de GPGPU 1206A-D. Cada una de las múltiples GPGPU 1206A-D puede ser una instancia de la GPGPU 1100 de la Figura 11. Las GPGPU 1206A-D pueden interconectarse

mediante un conjunto de enlaces de GPU a GPU de punto a punto de alta velocidad 1216. Los enlaces de GPU a GPU de alta velocidad pueden conectarse a cada una de las GPGPU 1206A-D mediante un enlace de GPU dedicado, tal como el enlace de GPU 1110 como en la Figura 11. Los enlaces de GPU de P2P 1216 posibilitan una comunicación directa entre cada una de las GPGPU 1206A-D sin requerir una comunicación a través del bus de interfaz de anfitrión al que se conecta el procesador 1202. Con el tráfico de GPU a GPU dirigido a los enlaces de GPU de P2P, el bus de interfaz de anfitrión permanece disponible para el acceso de memoria de sistema o para comunicarse con otras instancias del sistema informático de múltiples GPU 1200, por ejemplo, mediante uno o más dispositivos de red. Aunque en la realización ilustrada las GPGPU 1206A-D se conectan al procesador 1202 mediante el conmutador de interfaz de anfitrión 1204, en una realización, el procesador 1202 incluye el soporte directo para los enlaces de GPU de P2P 1216 y puede conectarse directamente a las GPGPU 1206A-D.

Implementaciones de red neuronal de aprendizaje automático

La arquitectura informática proporcionada por realizaciones descritas en el presente documento puede configurarse para realizar los tipos de procesamiento paralelo que son particularmente adecuados para entrenar y desplegar redes neuronales para un aprendizaje automático. Una red neuronal puede generalizarse como una red de funciones que tienen una relación de grafo. Como es bien conocido en la técnica, hay una diversidad de tipos de implementaciones de red neuronal usadas en el aprendizaje automático. Un tipo ilustrativo de red neuronal es la red de realimentación prospectiva, como se ha descrito previamente.

Un segundo tipo ilustrativo de red neuronal es la red neuronal convolucional (CNN). Una CNN es una red neuronal de realimentación prospectiva especializada para procesar datos que tienen una topología de tipo cuadrícula conocida, tales como datos de imagen. En consecuencia, las CNN se usan comúnmente para aplicaciones de reconocimiento de imágenes y de visión de cálculo, pero también pueden usarse para otros tipos de reconocimiento de patrones, tales como procesamiento de habla y de idioma. Los nodos en la capa de entrada de CNN están organizados 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 un tipo especializado 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 la terminología de redes convolucionales, la primera función para la convolución puede denominarse entrada, mientras que la segunda función puede denominarse núcleo de convolución. La salida puede denominarse mapa de características. Por ejemplo, la entrada a una capa de convolución puede ser una matriz multidimensional de datos que definen las diversas componentes de color de una imagen de entrada. El núcleo de convolución puede ser una matriz multidimensional de parámetros, donde los parámetros están adaptados por el proceso de entrenamiento para la red neuronal.

Las redes neuronales recurrentes (RNN) son una familia de redes neuronales de realimentación prospectiva que incluyen conexiones de realimentación entre capas. Las RNN posibilitan el modelado de datos secuenciales compartiendo datos de parámetro a lo largo 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 sobre su propio valor en un tiempo futuro, debido a que al menos una parte de los datos de salida desde la RNN se usa como realimentación para procesar una entrada subsiguiente en una secuencia. Esta característica hace que las RNN sean particularmente útiles para el procesamiento de idioma debido a la naturaleza variable en la que pueden componerse los datos de idioma.

Las figuras descritas a continuación presentan redes de realimentación prospectiva, CNN y RNN ilustrativas, así como describen un proceso general para entrenar y desplegar respectivamente cada uno de estos 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, en general, a redes neuronales profundas y técnicas de aprendizaje automático en general.

Las redes neuronales ilustrativas descritas anteriormente pueden usarse para realizar un aprendizaje profundo. El aprendizaje profundo es un aprendizaje automático que usa redes neuronales profundas. Las redes neuronales profundas usadas en el aprendizaje profundo son redes neuronales artificiales compuestas por múltiples capas ocultas, en contraposición a redes neuronales poco profundas que solo incluyen una única capa oculta. El entrenamiento de redes neuronales más profundas es, en general, más intensivo desde el punto de vista computacional. Sin embargo, las capas ocultas adicionales de la red posibilitan un reconocimiento de patrones de múltiples etapas que da como resultado un error de salida reducido en relación con técnicas de aprendizaje automático poco profundo.

Las redes neuronales profundas usadas en el aprendizaje automático habitualmente incluyen una red de extremo frontal para realizar un reconocimiento de características, acoplada a una red de extremo posterior que representa un modelo matemático que puede realizar operaciones (por ejemplo, clasificación de objetos, reconocimiento de habla, etc.) basándose en la representación de características proporcionada al modelo. Un aprendizaje profundo posibilita que se realice un aprendizaje automático sin requerir que se realice una ingeniería de características artesanal para el modelo. En su lugar, las redes neuronales profundas pueden aprender características basándose en una correlación

o estructura 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 está especializado, en general, para la tarea específica que va a realizarse, y se usarán diferentes modelos para realizar diferentes tareas.

Una vez que se ha estructurado 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 los pesos dentro del modelo para reducir el error de salida de la red. La retropropagación 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 retropropagan, a continuación, 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 de descenso de gradiente estocástico, para actualizar los pesos de la red neuronal.

Las **Figuras 13A-B** ilustran una red neuronal convolucional ilustrativa. La Figura 13A ilustra diversas capas dentro de una CNN. Como se muestra en la Figura 13A, una CNN ilustrativa usada para modelar el procesamiento de imagen puede recibir la entrada 1302 que describe las componentes de rojo, verde y azul (RGB) de una imagen de entrada. La entrada 1302 puede ser procesada por múltiples capas convolucionales (por ejemplo, la capa convolucional 1304, la capa convolucional 1306). La salida desde las múltiples capas convolucionales puede ser procesada opcionalmente por un conjunto de capas completamente conectadas 1308. Las neuronas en una capa completamente conectada tienen conexiones completas a todas las activaciones en la capa previa, como se ha descrito previamente para una red de realimentación prospectiva. La salida desde las capas completamente conectadas 1308 puede usarse para generar un resultado de salida a partir de la red. Las activaciones dentro de las capas completamente conectadas 1308 pueden calcularse usando una multiplicación matricial en lugar de una convolución. No todas las implementaciones de CNN hacen uso de las capas completamente conectadas DPLA08. Por ejemplo, en algunas implementaciones, la capa convolucional 1306 puede generar una salida para la CNN.

Las capas convolucionales se conectan de manera dispersa, lo que difiere de la configuración de red neuronal tradicional encontrada en las capas completamente conectadas 1308. Las capas de red neuronal tradicionales están completamente conectadas, de manera que cada unidad de salida interacciona con cada unidad de entrada. Sin embargo, las capas convolucionales se conectan de manera dispersa debido a que se introduce la salida de la convolución de un campo (en lugar del valor de estado respectivo de cada uno de los nodos en el campo) en los nodos de la capa subsiguiente, 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 realice un ajuste a escala para procesar imágenes grandes.

La **Figura 13B** ilustra fases de cálculo ilustrativas dentro de una capa convolucional de una CNN. La entrada a una capa convolucional 1312 de una CNN puede procesarse en tres fases de una capa convolucional 1314. Las tres fases pueden incluir una fase de convolución 1316, una fase de detector 1318 y una fase de agrupamiento 1320. La capa de convolución 1314 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 completamente conectada, por ejemplo, para generar un valor de clasificación para la entrada a la CNN.

En la fase de convolución 1316 se realizan varias convoluciones en paralelo para producir un conjunto de activaciones lineales. La fase de convolución 1316 puede incluir una transformación afín, que es cualquier transformación que pueda especificarse como una transformación lineal más una traslación. Las transformaciones afines incluyen rotaciones, traslaciones, ajuste a escala y combinaciones de estas transformaciones. La fase de convolución calcula la salida de funciones (por ejemplo, neuronas) que se conectan a regiones específicas en la entrada, lo que puede determinarse como la región local asociada con la neurona. Las neuronas calculan un producto escalar entre los pesos de las neuronas y la región en la entrada local a la que se conectan las neuronas. La salida desde la fase de convolución 1316 define un conjunto de activaciones lineales que son procesadas por fases sucesivas de la capa convolucional 1314.

Las activaciones lineales pueden ser procesadas por una fase de detector 1318. En la fase de detector 1318, cada activación lineal es procesada por una función de activación no lineal. La función de activación no lineal aumenta las propiedades no lineales de la red global sin afectar a los campos receptivos de la capa de convolución. Pueden usarse varios tipos de funciones de activación no lineal. Un tipo particular es la unidad lineal rectificada (ReLU), que usa una función de activación definida como $f(x) = \max(0, x)$, de manera que se fija un umbral de cero para la activación.

La fase de agrupamiento 1320 usa una función de agrupamiento que sustituye la salida de la capa convolucional 1306 con una estadística de resumen de las salidas cercanas. La función de agrupamiento puede usarse para introducir la invarianza de traslación en la red neuronal, de manera que traslaciones pequeñas a la entrada no cambian las salidas agrupadas. La invarianza a la traslación local puede ser útil en escenarios donde 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 fase de agrupamiento 1320, incluyendo agrupamiento máximo, agrupamiento promedio y agrupamiento de norma l2. Adicionalmente, algunas implementaciones de CNN no incluyen una fase de agrupamiento. En su lugar, tales implementaciones sustituyen una fase de convolución adicional que tiene un paso aumentado en relación con fases de convolución previas.

La salida desde la capa convolucional 1314 puede ser procesada, a continuación, por la siguiente capa 1322. La siguiente capa 1322 puede ser una capa convolucional adicional o una de las capas completamente conectadas 1308. Por ejemplo, la primera capa convolucional 1304 de la **Figura 13A** puede emitir a la segunda capa convolucional 1306, mientras que la segunda capa convolucional puede emitir a una primera capa de las capas completamente conectadas 1308.

La **Figura 14** ilustra una red neuronal recurrente 1400 ilustrativa. En una red neuronal recurrente (RNN), el estado previo de la red influye sobre la salida del estado actual de la red. Las RNN pueden construirse de una diversidad de maneras usando una diversidad de funciones. El uso de las RNN pivota, en general, alrededor del uso de modelos matemáticos para predecir el futuro basándose en una secuencia anterior de entradas. Por ejemplo, una RNN puede usarse para realizar un modelado de idioma estadístico para predecir una palabra venidera, dada una secuencia previa de palabras. La RNN 1400 ilustrada puede describirse como que tiene una capa de entrada 1402 que recibe un vector de entrada, las capas ocultas 1404 para implementar una función recurrente, un mecanismo de realimentación 1405 para posibilitar una 'memoria' de estados previos y una capa de salida 1406 para emitir un resultado. La RNN 1400 opera basándose en escalones de tiempo. El estado de la RNN en un escalón de tiempo dado se ve influenciado basándose en el escalón de tiempo previo mediante el mecanismo de realimentación 1405. Para un escalón de tiempo dado, el estado de las capas ocultas 1404 se define por el estado previo y la entrada en el escalón de tiempo actual. Una entrada inicial (x_1) en un primer escalón de tiempo puede ser procesada por la capa oculta 1404. Una segunda entrada (x_2) puede ser procesada por la capa oculta 1404 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, en general, una no linealidad, tal como la función tangente hiperbólica (Tanh) o una variante de la función rectificadora $f(x) = \text{máx}(0, x)$. Sin embargo, la función matemática específica usada en las capas ocultas 1404 puede variar dependiendo de los detalles de implementación específicos de la RNN 1400.

Además de las redes CNN y RNN básicas descritas, pueden posibilitarse variaciones en estas redes. Una variante de RNN ilustrativa es la RNN de memoria a corto plazo larga (LSTM). Las RNN de LSTM pueden aprender dependencias a largo plazo que pueden ser necesarias para procesar secuencias de idioma 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 a una red de creencia profunda. Una red de creencia profunda (DBN) es una red neuronal generativa que está compuesta por múltiples capas de variables estocásticas (aleatorias). Las DBN pueden entrenarse capa a capa usando aprendizaje no supervisado voraz. Los pesos aprendidos de la DBN pueden usarse a continuación para proporcionar redes neuronales de preentrenamiento determinando un conjunto inicial óptimo de pesos para la red neuronal.

La **Figura 15** 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 1502. Se han desarrollado diversas estructuras de entrenamiento 1504 para posibilitar la aceleración de hardware del proceso de entrenamiento. Por ejemplo, la estructura de aprendizaje automático 1004 de la Figura 10 puede configurarse como una estructura de entrenamiento 1004. La estructura de entrenamiento 1004 puede engancharse a una red neuronal no entrenada 1506 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 1508.

Para iniciar el proceso de entrenamiento, los pesos 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 es un método de aprendizaje en el que se realiza un entrenamiento como una operación mediada, tal como cuando el conjunto de datos de entrenamiento 1502 incluye una entrada emparejada con la salida deseada para la entrada, o donde 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 retropropagan, a continuación, a través del sistema. La estructura de entrenamiento 1504 puede ajustarse para ajustar los pesos que controlan la red neuronal no entrenada 1506. La estructura de entrenamiento 1504 puede proporcionar herramientas para monitorizar cómo está convergiendo de bien la red neuronal no entrenada 1506 hacia un modelo adecuado para generar respuestas correctas basándose en datos de entrada conocidos. El proceso de entrenamiento tiene lugar repetidamente a medida que se ajustan los pesos de la red para perfeccionar 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 1508. La red neuronal entrenada 1508 puede desplegarse a continuación para implementar cualquier número de operaciones de aprendizaje automático.

El aprendizaje 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 aprendizaje no supervisado, el conjunto de datos de entrenamiento 1502 incluirá datos de entrada sin ningún dato de salida asociado. La red neuronal no entrenada 1506 puede aprender agrupamientos dentro de la entrada no etiquetada y puede determinar cómo las entradas individuales están relacionadas con el conjunto de datos global. El entrenamiento no supervisado puede usarse para generar un mapa de autoorganización, que es un tipo de red neuronal entrenada 1507 que puede realizar operaciones útiles en cuanto a la reducción de la dimensionalidad de los datos. El entrenamiento no supervisado puede usarse también para realizar una detección de anomalías, lo 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 al entrenamiento supervisado y no supervisado. El aprendizaje semisupervisado es una técnica en la que el conjunto de datos de entrenamiento 1502 incluye una mezcla de datos etiquetados y no etiquetados de la misma distribución. El aprendizaje incremental es una variante del aprendizaje supervisado en el que se usan continuamente datos de entrada para entrenar adicionalmente el modelo. El aprendizaje incremental posibilita que la red neuronal entrenada 1508 se adapte a los nuevos datos 1512 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 ser demasiado intensivo desde el punto de vista computacional para un único nodo de cálculo. En lugar de usar un único nodo de cálculo, puede usarse una red distribuida de nodos computacionales para acelerar el proceso de entrenamiento.

La **Figura 16** es un diagrama de bloques que ilustra un aprendizaje distribuido. El aprendizaje distribuido es un modelo de entrenamiento que usa múltiples nodos informáticos distribuidos para realizar un entrenamiento supervisado o no supervisado de una red neuronal. Cada uno de los nodos computacionales distribuidos puede incluir uno o más procesadores de anfitrión y uno o más de los nodos de procesamiento de propósito general, tales como la unidad de procesamiento de gráficos de propósito general altamente paralela 1100 como en la **Figura 1100**. Como se ilustra, un aprendizaje distribuido puede realizarse con el paralelismo de modelo 1602, el paralelismo de datos 1604 o una combinación del paralelismo de modelo y de datos 1604.

En el paralelismo de modelo 1602, diferentes nodos computacionales 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 ajustar a escala a modelos particularmente grandes. La división de los cálculos asociados con diferentes capas de la red neuronal posibilita el entrenamiento de redes neuronales muy grandes en las que los pesos de todas las capas no cabrían en la memoria de un único nodo computacional. En algunas instancias, el paralelismo de modelo puede ser particularmente útil en la ejecución de un entrenamiento no supervisado de redes neuronales grandes.

En el paralelismo de datos 1604, los diferentes nodos de la red distribuida tienen una instancia completa del modelo y cada nodo recibe una parte diferente de los datos. Los resultados desde los diferentes nodos se combinan a continuación. Aunque son posibles diferentes enfoques para el paralelismo de datos, los enfoques de entrenamiento de datos paralelos requieren, todos ellos, una técnica de combinación de resultados y de sincronización de los parámetros de modelo entre cada nodo. Los enfoques ilustrativos para la combinación de datos incluyen promediado de parámetros y paralelismo de datos basado en actualizaciones. El promediado de parámetros entrena cada nodo en un subconjunto de los datos de entrenamiento y establece los parámetros globales (por ejemplo, pesos, desvíos) al promedio de los parámetros desde cada nodo. El promediado de parámetros usa un servidor de parámetros central que mantiene los datos de parámetro. El paralelismo de datos basado en actualizaciones es similar al promediado de parámetros excepto en 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 actualizaciones puede realizarse de una manera descentralizada, donde las actualizaciones se comprimen y se transfieren entre nodos.

El paralelismo de modelo y de datos 1606 combinado puede implementarse, por ejemplo, en un sistema distribuido en el que cada nodo computacional incluye múltiples GPU. Cada nodo puede tener una instancia completa del modelo con GPU separadas dentro de cada nodo que se usan para entrenar diferentes partes del modelo.

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

Aplicaciones de aprendizaje automático ilustrativas

El aprendizaje automático puede aplicarse a resolver una diversidad de problemas tecnológicos, incluyendo, pero sin limitación, visión informática, conducción y navegación autónoma, reconocimiento de habla y procesamiento de

idioma. La visión informática ha sido tradicionalmente una de las áreas de investigación más activas para aplicaciones de aprendizaje automático. Las aplicaciones de visión informática varían desde la reproducción de capacidades visuales humanas, tales como el reconocimiento de caras, hasta la creación de nuevas categorías de capacidades visuales. Por ejemplo, las aplicaciones de visión informática pueden configurarse para reconocer ondas de sonido de las vibraciones inducidas en los objetos visibles en un vídeo. El aprendizaje automático acelerado por procesador paralelo posibilita que se entrenen aplicaciones de visión informática usando un conjunto de datos de entrenamiento significativamente mayor que el previamente factible y posibilita que se desarrollen sistemas de inferencia usando procesadores paralelos de baja potencia.

El aprendizaje automático acelerado por procesador paralelo tiene aplicaciones de conducción autónoma que incluyen reconocimiento de señales de carretera y de carril, evitación de obstáculos, navegación y control de conducción. Las técnicas de aprendizaje automático aceleradas pueden usarse para entrenar modelos de conducción basándose en conjuntos de datos que definen las respuestas apropiadas a una entrada de entrenamiento específica. Los procesadores paralelos descritos en el presente documento pueden posibilitar el entrenamiento rápido de las redes neuronales cada vez más complejas usadas para soluciones de conducción autónoma 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 por procesador paralelo han posibilitado enfoques de aprendizaje automático para un reconocimiento de habla automático (ASR). El ASR incluye la creación de una función que, dada una secuencia acústica de entrada, calcula la secuencia lingüística más probable. El aprendizaje automático acelerado usando redes neuronales profundas ha posibilitado la sustitución de los modelos ocultos de Markov (HMM) y los modelos de mezcla gaussiana (GMM) previamente usados para el ASR.

El aprendizaje automático acelerado por procesador paralelo puede usarse también para acelerar el procesamiento de lenguaje natural. Los procedimientos de aprendizaje automático pueden hacer uso de algoritmos de inferencia estadística para producir modelos que son robustos ante una entrada errónea o no familiar. Las aplicaciones de procesador de lenguaje natural ilustrativas incluyen la traducción mecánica automática entre idiomas humanos.

Las plataformas de procesamiento paralelo usadas para el aprendizaje automático pueden dividirse en plataformas de entrenamiento y plataformas de despliegue. Las plataformas de entrenamiento son, en general, altamente paralelas e incluyen optimizaciones para acelerar el entrenamiento de múltiples GPU y un único nodo y el entrenamiento de múltiples nodos y múltiples GPU. Los procesadores paralelos ilustrativos adecuados para el entrenamiento incluyen la unidad de procesamiento de gráficos de propósito general altamente paralela 1100 de la **Figura 1100** y el sistema informático de múltiples GPU 1200 de la **Figura 1200**. Por el contrario, las plataformas de aprendizaje automático desplegadas incluyen, en general, procesadores paralelos de potencia inferior adecuados para su uso en productos tales como cámaras, robots autónomos y vehículos autónomos.

La **Figura 17** ilustra un sistema en un chip (SOC) de inferencia 1700 ilustrativo adecuado para realizar la inferencia usando un modelo entrenado. El SOC 1700 puede integrar componentes de procesamiento que incluyen un procesador de medios 1702, un procesador de visión 1704, una GPGPU 1706 y un procesador de múltiples núcleos 1708. El SOC 1700 puede incluir adicionalmente la memoria en chip 1705 que puede posibilitar un agrupamiento de datos en chip compartida a la que puede acceder cada uno de los componentes de procesamiento. Los componentes de procesamiento pueden optimizarse para una operación de baja potencia para posibilitar el despliegue en una diversidad de plataformas de aprendizaje automático, incluyendo vehículos autónomos y robots autónomos. Por ejemplo, una implementación del SOC 1700 puede usarse como una parte del sistema de control principal para un vehículo autónomo. Donde el SOC 1700 está configurado para su uso en vehículos autónomos, el SOC se diseña y está configurado para cumplir con las normas de seguridad funcional relevantes de la jurisdicción de despliegue.

Durante el funcionamiento, el procesador de medios 1702 y el procesador de visión 1704 pueden trabajar conjuntamente para acelerar las operaciones de visión informática. El procesador de medios 1702 puede posibilitar la decodificación de latencia baja de múltiples flujos de vídeo de alta resolución (por ejemplo, 4K, 8K). Los flujos de vídeo descodificados pueden escribirse en una memoria intermedia en la memoria en chip 1705. El procesador de visión 1704 puede analizar, a continuación, el vídeo decodificado y realizar operaciones de procesamiento preliminares sobre los fotogramas del vídeo decodificado como preparación al procesamiento de los fotogramas usando un modelo de reconocimiento de imágenes entrenado. Por ejemplo, el procesador de visión 1704 puede acelerar las operaciones de convolución para una CNN que se usa para realizar un reconocimiento de imágenes sobre los datos de vídeo de alta resolución, mientras que los cálculos de modelo de extremo posterior son realizados por la GPGPU 1706.

El procesador de múltiples núcleos 1708 puede incluir una lógica de control de asistencia en la secuenciación y la sincronización de transferencias de datos y operaciones de memoria compartida realizadas por el procesador de medios 1702 y el procesador de visión 1704. El procesador de múltiples núcleos 1708 también puede funcionar como un procesador de aplicaciones para ejecutar aplicaciones de software que pueden hacer uso de la capacidad de cálculo de inferencia de la GPGPU 1706. Por ejemplo, al menos una parte de la lógica de navegación y de conducción puede implementarse en software que se ejecuta en el procesador de múltiples núcleos 1708. Tal software puede emitir directamente cargas de trabajo computacionales a la GPGPU 1706 o las cargas de trabajo computacionales

pueden emitirse al procesador de múltiples núcleos 1708, que puede descargar al menos una parte de esas operaciones a la GPGPU 1706.

La GPGPU 1706 puede incluir agrupaciones de cálculo, tal como una configuración de baja potencia de las agrupaciones de cálculo 1106A-1106H dentro de la unidad de procesamiento de gráficos de propósito general altamente paralela 1100. Las agrupaciones de cálculo dentro de la GPGPU 1706 pueden soportar instrucciones que se optimizan específicamente para realizar cálculos de inferencia sobre una red neuronal entrenada. Por ejemplo, la GPGPU 1706 puede soportar instrucciones para realizar cálculos de precisión baja, tales como operaciones vectoriales de números enteros de 8 bits y de 4 bits.

Vista general del sistema II

La **Figura 18** es un diagrama de bloques de un sistema de procesamiento 1800, de acuerdo con una realización. En diversas realizaciones, el sistema 1800 incluye uno o más procesadores 1802 y uno o más procesadores de gráficos 1808, y puede ser un sistema de sobremesa de procesador único, un sistema de estación de trabajo de multiprocesador o un sistema de servidor que tiene un gran número de procesadores 1802 o núcleos de procesador 1807. En una realización, el sistema 1800 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 1800 puede incluir o incorporarse dentro de una plataforma de juegos basada en servidor, una consola de juegos, incluyendo una consola de juegos y medios, una consola de juegos móvil, una consola de juegos portátil o una consola de juegos en línea. En algunas realizaciones, el sistema 1800 es un teléfono móvil, un teléfono inteligente, un dispositivo informático de tipo tableta o un dispositivo de Internet móvil. El sistema de procesamiento de datos 1800 también puede incluir, acoplarse o estar integrado dentro de un dispositivo portátil, tal como un dispositivo ponible tipo reloj inteligente, un dispositivo de gafas inteligentes, un dispositivo de realidad aumentada o un dispositivo de realidad virtual. En algunas realizaciones, el sistema de procesamiento de datos 1800 es un televisor o dispositivo decodificador que tiene uno o más procesadores 1802 y una interfaz gráfica generada por uno o más procesadores de gráficos 1808.

En algunas realizaciones, cada uno de los uno o más procesadores 1802 incluye uno o más núcleos de procesador 1807 para procesar instrucciones que, cuando se ejecutan, realizan operaciones para software de usuario y sistema. En algunas realizaciones, cada uno de los uno o más núcleos de procesador 1807 está configurado para procesar un conjunto de instrucciones 1809 específico. En algunas realizaciones, el conjunto de instrucciones 1809 puede facilitar el cálculo de conjunto de instrucciones complejo (CISC), el cálculo de conjunto de instrucciones reducido (RISC) o el cálculo mediante una palabra de instrucción muy larga (VLIW). Múltiples núcleos de procesador 1807 pueden procesar, cada uno, un conjunto de instrucciones 1809 diferente, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo de procesador 1807 también puede incluir otros dispositivos de procesamiento, tales como un procesador de señales digitales (DSP).

En algunas realizaciones, el procesador 1802 incluye la memoria caché 1804. Dependiendo de la arquitectura, el procesador 1802 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 1802. En algunas realizaciones, el procesador 1802 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 1807 usando técnicas de coherencia de caché conocidas. Se incluye adicionalmente un archivo de registro 1806 en el procesador 1802 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 propósito general, mientras que otros registros pueden ser específicos del diseño del procesador 1802.

En algunas realizaciones, el procesador 1802 está acoplado a un bus de procesador 1810 para transmitir señales de comunicación tales como señales de dirección, de datos o de control entre el procesador 1802 y otros componentes en el sistema 1800. En una realización, el sistema 1800 usa una arquitectura de sistema de 'concentrador' ilustrativa, incluyendo un concentrador de controlador de memoria 1816 y un concentrador de controlador de entrada-salida (E/S) 1830. Un concentrador de controlador de memoria 1816 facilita la comunicación entre un dispositivo de memoria y otros componentes del sistema 1800, mientras que un concentrador de controlador de E/S (ICH) 1830 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 1816 está integrada dentro del procesador.

El dispositivo de memoria 1820 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, un dispositivo de memoria de cambio de fase o algún otro dispositivo de memoria que tiene un rendimiento adecuado para servir como memoria de proceso. En una realización, el dispositivo de memoria 1820 puede operar como memoria de sistema para el sistema 1800, para almacenar datos 1822 e instrucciones 1821 para su uso cuando los uno o más procesadores 1802 ejecutan una aplicación o proceso. El concentrador de controlador de memoria 1816 también se acopla con un procesador de gráficos externo 1812 opcional, que puede comunicarse con los uno o más procesadores de gráficos 1808 en los procesadores 1802 para realizar operaciones de gráficos y de medios.

En algunas realizaciones, el ICH 1830 posibilita que los periféricos se conecten al dispositivo de memoria 1820 y al procesador 1802 mediante un bus de E/S de alta velocidad. Los periféricos de E/S incluyen, pero sin limitación, un controlador de audio 1846, una interfaz de firmware 1828, un transceptor inalámbrico 1826 (por ejemplo, Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1824 (por ejemplo, unidad de disco duro, memoria flash, etc.), y un controlador de E/S heredado 1840 para acoplar dispositivos heredados (por ejemplo, dispositivos de sistema personal 2 (PS/2)) al sistema. Uno o más controladores de bus serie universal (USB) 1842 conectan dispositivos de entrada, tales como combinaciones de teclado y ratón 1844. Un controlador de red 1834 también puede acoplarse al ICH 1830. En algunas realizaciones, un controlador de red de alto rendimiento (no mostrado) se acopla al bus de procesador 1810. Se apreciará que el sistema 1800 mostrado es ilustrativo y no limitante, debido a que también pueden usarse otros tipos de sistemas de procesamiento de datos que están configurados de manera diferente. Por ejemplo, el concentrador de controlador de E/S 1830 puede integrarse dentro de los uno o más procesadores 1802, o el concentrador de controlador de memoria 1816 y el concentrador de controlador de E/S 1830 pueden integrarse en un procesador de gráficos externo discreto, tal como el procesador de gráficos externo 1812.

La **Figura 19** es un diagrama de bloques de una realización de un procesador 1900 que tiene uno o más núcleos de procesador 1902A-1902N, un controlador de memoria integrado 1914 y un procesador de gráficos integrado 1908. Aquellos 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 cualquier otra parte en el presente documento, pero sin limitación a esto. El procesador 1900 puede incluir núcleos adicionales hasta e incluyendo el núcleo adicional 1902N representado por los recuadros de línea discontinua. Cada uno de los núcleos de procesador 1902A-1902N incluye una o más unidades de caché internas 1904A-1904N. En algunas realizaciones, cada núcleo de procesador también tiene acceso a una o más unidades almacenadas en caché compartidas 1906.

Las unidades de caché internas 1904A-1904N y las unidades de caché compartidas 1906 representan una jerarquía de memoria caché dentro del procesador 1900. La jerarquía de memoria caché puede incluir al menos un nivel de caché de instrucciones y de datos dentro de cada núcleo de procesador y uno o más niveles de caché de nivel medio compartida, tal como una caché de Nivel 2 (L2), de Nivel 3 (L3), de Nivel 4 (L4) o de otros niveles, donde el nivel más alto de caché antes de la memoria externa se clasifica como LLC. En algunas realizaciones, la lógica de coherencia de caché mantiene la coherencia entre las diversas unidades de caché 1906 y 1904A-1904N.

En algunas realizaciones, el procesador 1900 también puede incluir un conjunto de una o más unidades de controlador de bus 1916 y un núcleo de agente de sistema 1910. Las una o más unidades de controlador de bus 1916 gestionan un conjunto de buses de periféricos, tales como uno o más buses de interconexión de componentes periféricos (por ejemplo, PCI, PCI Express). El núcleo de agente de sistema 1910 proporciona funcionalidad de gestión para los diversos componentes de procesador. En algunas realizaciones, el núcleo de agente de sistema 1910 incluye uno o más controladores de memoria integrados 1914 para gestionar el acceso a diversos dispositivos de memoria externos (no mostrados).

En algunas realizaciones, uno o más de los núcleos de procesador 1902A-1902N incluyen el soporte para múltiples hilos simultáneos. En una realización de este tipo, el núcleo de agente de sistema 1910 incluye componentes para coordinar y operar los núcleos 1902A-1902N durante el procesamiento de múltiples hilos. El núcleo de agente de sistema 1910 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 1902A-1902N y el procesador de gráficos 1908.

En algunas realizaciones, el procesador 1900 incluye adicionalmente el procesador de gráficos 1908 para ejecutar las operaciones de procesamiento de gráficos. En algunas realizaciones, el procesador de gráficos 1908 se acopla con el conjunto de unidades de caché compartidas 1906 y el núcleo de agente de sistema 1910, incluyendo los uno o más controladores de memoria integrados 1914. En algunas realizaciones, un controlador de visualización 1911 está acoplado con el procesador de gráficos 1908 para controlar la salida de procesador de gráficos a una o más pantallas acopladas. En algunas realizaciones, el controlador de visualización 1911 puede ser un módulo separado acoplado con el procesador de gráficos a través de al menos una interconexión, o puede estar integrado dentro del procesador de gráficos 1908 o el núcleo de agente de sistema 1910.

En algunas realizaciones, se usa una unidad de interconexión basada en anillo 1912 para acoplar los componentes internos del procesador 1900. Sin embargo, puede usarse una unidad de interconexión alternativa, tal como una interconexión de punto a punto, una interconexión conmutada u otras técnicas, incluyendo técnicas bien conocidas en la técnica. En algunas realizaciones, el procesador de gráficos 1908 se acopla con la interconexión en anillo 1912 a través de un enlace de E/S 1913.

El enlace de E/S 1913 ilustrativo representa al menos una de múltiples diversidades de interconexiones de E/S, incluyendo una interconexión de E/S en paquete que facilita la comunicación entre diversos componentes de procesador y un módulo de memoria integrado de alto rendimiento 1918, tal como un módulo de eDRAM. En algunas realizaciones, cada uno de los núcleos de procesador 1902-1902N y el procesador de gráficos 1908 usan módulos de memoria integrados 1918 como una caché de último nivel compartida.

En algunas realizaciones, los núcleos de procesador 1902A-1902N son núcleos homogéneos que ejecutan la misma arquitectura de conjunto de instrucciones. En otra realización, los núcleos de procesador 1902A-1902N son heterogéneos en términos de arquitectura de conjunto de instrucciones (ISA), donde uno o más de los núcleos de procesador 1902A-N ejecutan un primer conjunto de instrucciones, mientras que al menos uno de los otros núcleos ejecuta un subconjunto del primer conjunto de instrucciones o un conjunto de instrucciones diferente. En una realización, los núcleos de procesador 1902A-1902N son heterogéneos en términos de microarquitectura, donde uno o más núcleos que tienen un consumo de energía relativamente superior se acoplan con uno o más núcleos de potencia que tienen un consumo de energía inferior. Adicionalmente, el procesador 1900 puede implementarse en uno o más chips o como un circuito integrado de SoC que tiene los componentes ilustrados, además de otros componentes.

La **Figura 20** es un diagrama de bloques de un procesador de gráficos 2000, 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 por medio de una interfaz de E/S mapeada en memoria con registros del procesador de gráficos y con los comandos almacenados en la memoria de procesador. En algunas realizaciones, el procesador de gráficos 2000 incluye una interfaz de memoria 2014 para acceder a memoria. La interfaz de memoria 2014 puede ser una interfaz a una memoria local, una o más cachés internas, una o más cachés externas compartidas y/o a una memoria de sistema.

En algunas realizaciones, el procesador de gráficos 2000 también incluye un controlador de visualización 2002 para enviar datos de salida de visualización a un dispositivo de visualización 2020. El controlador de visualización 2002 incluye hardware para uno o más planos de superposición para la visualización y composición de múltiples capas de elementos de interfaz de usuario o de vídeo. En algunas realizaciones, el procesador de gráficos 2000 incluye un motor de códec de vídeo 2006 para codificar, decodificar o transcodificar medios a, desde o entre uno o más formatos de codificación de medios, incluyendo, pero sin limitación, formatos del Grupo de Expertos en Imágenes en Movimiento (MPEG) tales como MPEG-2, formatos de Codificación de Vídeo Avanzada (AVC) tales como H.264/MPEG-4 AVC, así como de la Sociedad de Ingenieros de Imágenes en Movimiento y de Televisión (SMPTE) 421M/VC-1 y formatos del Grupo Conjunto de Expertos en Fotografía (JPEG) tales como los formatos JPEG y Motion JPEG (MJPEG).

En algunas realizaciones, el procesador de gráficos 2000 incluye un motor de transferencia de imágenes en bloque (BLIT) 2004 para realizar operaciones de rasterizador bidimensionales (2D), incluyendo, por ejemplo, transferencias de bloque de frontera de bits. Sin embargo, en una realización, se realizan operaciones de gráficos 2D usando uno o más componentes del motor de procesamiento de gráficos (GPE) 2010. En algunas realizaciones, el motor de procesamiento de gráficos 2010 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 2010 incluye una canalización 3D 2012 para realizar operaciones 3D, tales como representar imágenes y escenas tridimensionales usando funciones de procesamiento que actúan sobre formas de primitivas 3D (por ejemplo, rectángulo, triángulo, etc.). La canalización 3D 2012 incluye elementos de función programable y fija que realizan diversas tareas dentro del elemento y/o generan hilos de ejecución en un subsistema 3D/de medios 2015. Aunque la canalización 3D 2012 se puede usar para realizar operaciones de medios, una realización del GPE 2010 también incluye una canalización de medios 2016 que se usa específicamente para realizar operaciones de medios, tales como post-procesamiento de vídeo y mejora de imagen.

En algunas realizaciones, la canalización de medios 2016 incluye unidades de lógica programable o de función fija para realizar una o más operaciones de medios especializadas, tales como aceleración de descodificación de vídeo, desentrelazado de vídeo y aceleración de codificación de vídeo en lugar o en nombre del motor de códec de vídeo 2006. En algunas realizaciones, la canalización de medios 2016 incluye adicionalmente una unidad de generación de hilos para generar hilos para su ejecución en el subsistema 3D/de medios 2015. Los hilos generados realizan cálculos para las operaciones de medios en una o más unidades de ejecución de gráficos incluidas en el subsistema 3D/de medios 2015.

En algunas realizaciones, el subsistema 3D/de medios 2015 incluye una lógica para ejecutar hilos generados por la canalización 3D 2012 y la canalización de medios 2016. En una realización, las canalizaciones envían solicitudes de ejecución de hilos al subsistema 3D/de medios 2015, que incluye lógica de despacho de hilo para arbitrar y despachar las diversas solicitudes a recursos de ejecución de hilos disponibles. Los recursos de ejecución incluyen una matriz de unidades de ejecución de gráficos para procesar los hilos 3D y de medios. En algunas realizaciones, el subsistema 3D/de medios 2015 incluye una o más cachés internas para datos e instrucciones de hilo. En algunas realizaciones, el subsistema también incluye memoria compartida, que incluye registros y memoria direccionable, para compartir datos entre hilos y para almacenar datos de salida.

Procesamiento 3D/de medios

La **Figura 21** es un diagrama de bloques de un motor de procesamiento de gráficos 2110 de un procesador de gráficos de acuerdo con algunas realizaciones. En una realización, el motor de procesamiento de gráficos (GPE) 2110 es una versión del GPE 2010 mostrado en la **Figura 20**. Aquellos 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 cualquier otra parte en el presente documento, pero sin limitación a esto. Por ejemplo, se ilustra la canalización 3D 2012 y la canalización de medios 2016 de la **Figura 20**. La canalización de medios 2016 es opcional en algunas realizaciones del GPE 2110 y puede no incluirse explícitamente dentro del GPE 2110. Por ejemplo, y en al menos una realización, un procesador de medios y/o de imágenes separado se acopla al GPE 2110.

En algunas realizaciones, el GPE 2110 se acopla con o incluye un transmisor por flujo continuo de comandos 2103, que proporciona un flujo de comandos a la canalización 3D 2012 y/o a las canalizaciones de medios 2016. En algunas realizaciones, el transmisor por flujo continuo de comandos 2103 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 transmisor por flujo continuo de comandos 2103 recibe comandos desde la memoria y envía los comandos a la canalización 3D 2012 y/o a la canalización de medios 2016. Los comandos son directivas extraídas de una memoria intermedia en anillo, que almacena comandos para la canalización 3D 2012 y la canalización de medios 2016. En una realización, la memoria intermedia en anillo puede incluir adicionalmente memorias intermedias de comandos por lotes que almacenan lotes de múltiples comandos. Los comandos para la canalización 3D 2012 también pueden incluir referencias a datos almacenados en memoria, tales como, pero sin limitación, datos de vértice y de geometría para la canalización 3D 2012 y/o datos de imagen y objetos de memoria para la canalización de medios 2016. La canalización 3D 2012 y la canalización de medios 2016 procesan los comandos y datos realizando operaciones mediante lógica dentro de las canalizaciones respectivas o despachando uno o más hilos de ejecución a la matriz de núcleos de gráficos 2114.

En diversos ejemplos, la canalización 3D 2012 puede ejecutar uno o más programas sombreadores, tales como sombreadores de vértices, sombreadores de geometría, sombreadores de píxeles, sombreadores de fragmentos, sombreadores de cálculos u otros programas sombreadores, procesando las instrucciones y despachando hilos de ejecución a la matriz de núcleos de gráficos 2114. La matriz de núcleos de gráficos 2114 proporciona un bloque unificado de recursos de ejecución. La lógica de ejecución de múltiples propósitos (por ejemplo, unidades de ejecución) dentro de la matriz de núcleos de gráficos 2114 incluye un soporte para diversos lenguajes de sombreador de API 3D y puede ejecutar múltiples hilos de ejecución simultáneos asociados con múltiples sombreadores.

En algunas realizaciones, la matriz de núcleos de gráficos 2114 también incluye lógica de ejecución para realizar funciones de medios, tales como procesamiento de vídeo y/o de imagen. En una realización, las unidades de ejecución incluyen adicionalmente una lógica de propósito general que es programable para realizar operaciones computacionales de propósito general paralelas, además de operaciones de procesamiento de gráficos. La lógica de propósito general puede realizar operaciones de procesamiento en paralelo o en conjunto con la lógica de propósito general dentro del/de los núcleo(s) de procesador 1807 de la **Figura 18** o el núcleo 1902A-1902N como en la **Figura 19**.

Los datos de salida generados por hilos que se ejecutan en la matriz de núcleos de gráficos 2114 pueden proporcionar datos a memoria en una memoria intermedia de retorno unificado (URB) 2118. La URB 2118 puede almacenar datos para múltiples hilos. En algunas realizaciones, la URB 2118 puede usarse para enviar datos entre diferentes hilos que se ejecutan en la matriz de núcleos de gráficos 2114. En algunas realizaciones, la URB 2118 puede usarse adicionalmente para la sincronización entre hilos en la matriz de núcleos de gráficos y la lógica de función fija dentro de la lógica de funciones compartidas 2120.

En algunas realizaciones, la matriz de núcleos de gráficos 2114 es ajustable a escala, 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 2110. En una realización, los recursos de ejecución son dinámicamente ajustables a escala, de manera que los recursos de ejecución pueden habilitarse o deshabilitarse según sea necesario.

La matriz de núcleos de gráficos 2114 se acopla con la lógica de funciones compartidas 2120 que incluye múltiples recursos que se comparten entre los núcleos de gráficos en la matriz de núcleos de gráficos. Las funciones compartidas dentro de la lógica de funciones compartidas 2120 son unidades de lógica de hardware que proporcionan una funcionalidad complementaria especializada a la matriz de núcleos de gráficos 2114. En diversas realizaciones, la lógica de funciones compartidas 2120 incluye, pero sin limitación, la lógica del muestreador 2121, del cálculo matemático 2122 y de la comunicación entre hilos (ITC) 2123. Adicionalmente, algunas realizaciones implementan una o más cachés 2125 dentro de la lógica de funciones compartidas 2120. Se implementa una función compartida donde la demanda de una función especializada dada es insuficiente para su inclusión dentro de la matriz de núcleos de gráficos 2114. En su lugar, una única instanciación de esa función especializada se implementa como una entidad autónoma en la lógica de funciones compartidas 2120 y se comparte entre los recursos de ejecución dentro de la matriz de núcleos de gráficos 2114. El conjunto preciso de funciones que se comparten entre la matriz de núcleos de gráficos 2114 y se incluyen dentro de la matriz de núcleos de gráficos 2114 varía entre realizaciones.

La **Figura 22** es un diagrama de bloques de otra realización de un procesador de gráficos 2200. Aquellos elementos de la **Figura 22** 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 cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, el procesador de gráficos 2200 incluye una interconexión en anillo 2202, un extremo frontal de canalización 2204, un motor de medios 2237 y unos núcleos de gráficos 2280A-2280N. En algunas realizaciones, la interconexión en anillo 2202 acopla el procesador de gráficos a otras unidades de procesamiento, que incluyen otros procesadores de gráficos o uno o más núcleos de procesadores de propósito 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 2200 recibe lotes de comandos mediante la interconexión en anillo 2202. Los comandos entrantes son interpretados por un transmisor por flujo continuo de comandos 2203 en el extremo frontal de canalización 2204. En algunas realizaciones, el procesador de gráficos 2200 incluye una lógica de ejecución ajustable a escala para realizar procesamiento de geometría 3D y procesamiento de medios a través del/de los núcleo(s) de gráficos 2280A-2280N. Para los comandos de procesamiento de geometría 3D, el transmisor por flujo continuo de comandos 2203 suministra comandos a la canalización de geometría 2236. Para al menos algunos comandos de procesamiento de medios, el transmisor por flujo continuo de comandos 2203 suministra los comandos a un extremo frontal de vídeo 2234, que se acopla con un motor de medios 2237. En algunas realizaciones, el motor de medios 2237 incluye un motor de calidad de vídeo (VQE) 2230 para post-procesamiento de vídeo y de imagen y un motor de codificación/decodificación de múltiples formatos (MFX) 2233 para proporcionar codificación y decodificación de datos de medios acelerados por hardware. En algunas realizaciones, cada uno de la canalización de geometría 2236 y el motor de medios 2237 generan hilos de ejecución para los recursos de ejecución de hilos proporcionados por al menos un núcleo de gráficos 2280A.

En algunas realizaciones, el procesador de gráficos 2200 incluye recursos de ejecución de hilos ajustables a escala que cuentan con los núcleos modulares 2280A-2280N (denominados, en ocasiones, segmentos de núcleo), teniendo cada uno múltiples subnúcleos 2250A-2250N, 2260A-2260N (denominados, en ocasiones, subsegmentos de núcleo). En algunas realizaciones, el procesador de gráficos 2200 puede tener cualquier número de núcleos de gráficos 2280A a 2280N. En algunas realizaciones, el procesador de gráficos 2200 incluye un núcleo de gráficos 2280A que tiene al menos un primer subnúcleo 2250A y un segundo subnúcleo de núcleo 2260A. En otras realizaciones, el procesador de gráficos es un procesador de baja potencia con un único subnúcleo (por ejemplo, 2250A). En algunas realizaciones, el procesador de gráficos 2200 incluye múltiples núcleos de gráficos 2280A-2280N, incluyendo cada uno un conjunto de primeros subnúcleos 2250A-2250N y un conjunto de segundos subnúcleos 2260A-2260N. Cada subnúcleo en el conjunto de primeros subnúcleos 2250A-2250N incluye al menos un primer conjunto de unidades de ejecución 2252A-2252N y muestreadores de medios/texturas 2254A-2254N. Cada subnúcleo en el conjunto de segundos subnúcleos 2260A-2260N incluye al menos un segundo conjunto de unidades de ejecución 2262A-2262N y muestreadores 2264A-2264N. En algunas realizaciones, cada subnúcleo 2250A-2250N, 2260A-2260N comparte un conjunto de recursos compartidos 2270A-2270N. En algunas realizaciones, los recursos compartidos incluyen memoria caché compartida y lógica de operación de píxeles. También pueden incluirse otros recursos compartidos en las diversas realizaciones del procesador de gráficos.

Lógica de ejecución

La **Figura 23** ilustra lógica de ejecución de hilos 2300, que incluye una matriz de elementos de procesamiento empleados en algunas realizaciones de un GPE. Aquellos 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 cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, la lógica de ejecución de hilos 2300 incluye un sombreador de píxeles 2302, un despachador de hilos 2304, una caché de instrucciones 2306, una matriz de unidades de ejecución escalable que incluye una pluralidad de unidades de ejecución 2308A-2308N, un muestreador 2310, una caché de datos 2312 y un puerto de datos 2314. En una realización, los componentes incluidos están interconectados mediante un tejido de interconexión que se enlaza a cada uno de los componentes. En algunas realizaciones, la lógica de ejecución de hilos 2300 incluye una o más conexiones a memoria, tal como la memoria de sistema o memoria caché, a través de una o más de la caché de instrucciones 2306, el puerto de datos 2314, el muestreador 2310 y la matriz de unidades de ejecución 2308A-2308N. En algunas realizaciones, cada unidad de ejecución (por ejemplo, 2308A) es un procesador de vector individual que puede ejecutar múltiples hilos simultáneos y procesar múltiples elementos de datos en paralelo para cada hilo. En algunas realizaciones, la matriz de unidades de ejecución 2308A-2308N incluye cualquier número de unidades de ejecución individuales.

En algunas realizaciones, la matriz de unidades de ejecución 2308A-2308N se usa principalmente para ejecutar programas de "sombreador". En algunas realizaciones, las unidades de ejecución en la matriz 2308A-2308N ejecutan un conjunto de instrucciones que incluye el soporte nativo para muchas instrucciones de sombreador de gráficos 3D convencional, de manera que se ejecutan los programas de sombreador de las bibliotecas de gráficos (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 propósito general (por ejemplo, sombreadores de cálculo y de medios).

Cada unidad de ejecución en la matriz de unidades de ejecución 2308A-2308N opera en matrices de elementos de datos. El número de elementos de datos es el "tamaño de ejecución" o el número de canales para la instrucción. Un canal de ejecución es una unidad lógica de ejecución para el acceso, enmascaramiento y control de flujo de elementos de datos dentro de las instrucciones. El número de canales puede ser independiente del número de unidades aritmético-lógicas (ALU) o unidades de coma flotante (FPU) físicas de un procesador de gráficos en particular. En algunas realizaciones, las unidades de ejecución 2308A-2308N soportan tipos de datos de números enteros y de coma flotante.

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

Una o más cachés de instrucciones internas (por ejemplo, 2306) se incluyen en la lógica de ejecución de hilos 2300 para almacenar en caché instrucciones de hilo para las unidades de ejecución. En algunas realizaciones, una o más cachés de datos (por ejemplo, 2312) se incluyen para almacenar en caché datos de hilo durante la ejecución de hilos. En algunas realizaciones, se incluye un muestreador 2310 para proporcionar un muestreo de textura para operaciones 3D y muestreo de medios para operaciones de medios. En algunas realizaciones, el muestreador 2310 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 canalizaciones de gráficos y de medios envían solicitudes de iniciación de hilo a la lógica de ejecución de hilos 2300 mediante lógica de generación y de despacho de hilos. En algunas realizaciones, la lógica de ejecución de hilos 2300 incluye un despachador de hilos local 2304 que arbitra las solicitudes de inicio de hilos de las canalizaciones de gráficos y medios y genera instancias a los hilos solicitados en una o más unidades de ejecución 2308A-2308N. Por ejemplo, la canalización de geometría (por ejemplo, 2236 de la **Figura 22**) despacha hilos de procesamiento de vértices, teselación o procesamiento de geometría a la lógica de ejecución de hilos 2300 (**Figura 23**). En algunas realizaciones, el despachador de hilos 2304 también puede procesar solicitudes de generación de hilos en tiempo de ejecución desde los programas sombreadores en ejecución.

Una vez que un grupo de objetos geométricos ha sido procesado y rasterizado en datos de píxeles, se invoca el sombreador de píxeles 2302 para calcular además la información de salida y hacer que los resultados se escriban en las superficies de salida (por ejemplo, memorias intermedias de color, memorias intermedias de profundidad, memorias intermedias de estarcido, etc.). En algunas realizaciones, el sombreador de píxeles 2302 calcula los valores de los diversos atributos de vértice que se van a interpolar a lo largo del objeto rasterizado. En algunas realizaciones, el sombreador de píxeles 2302 a continuación ejecuta un programa de sombreador de píxeles suministrado por la interfaz de programación de aplicaciones (API). Para ejecutar el programa de sombreador de píxeles, el sombreador de píxeles 2302 despacha hilos a una unidad de ejecución (por ejemplo, 2308A) mediante el despachador de hilos 2304. En algunas realizaciones, el sombreador de píxeles 2302 usa la lógica de muestreo de textura en el muestreador 2310 para acceder a datos de textura en mapas de textura almacenados en memoria. Operaciones aritméticas sobre los datos de textura y los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico, o descartan el procesamiento adicional de uno o más píxeles.

En algunas realizaciones, el puerto de datos 2314 proporciona un mecanismo de acceso de memoria para que la lógica de ejecución de hilos 2300 emita datos procesados a la memoria para su procesamiento en una canalización de salida de procesador de gráficos. En algunas realizaciones, el puerto de datos 2314 incluye o se acopla a una o más memorias caché (por ejemplo, la caché de datos 2312) para almacenar en caché datos para un acceso de memoria por medio del puerto de datos.

La **Figura 24** es un diagrama de bloques que ilustra los formatos de instrucción de procesador de gráficos 2400 de acuerdo con algunas realizaciones. En una o más realizaciones, las unidades de ejecución de procesador de gráficos soportan un conjunto de instrucciones que tiene instrucciones en múltiples formatos. Los recuadros con línea continua ilustran los componentes que se incluyen, en general, en una instrucción de unidad de ejecución, mientras que las líneas discontinuas incluyen componentes que son opcionales o que solo se incluyen en un subconjunto de las instrucciones. En algunas realizaciones, el formato de instrucción 2400 descrito e ilustrado son macro-instrucciones,

en el sentido de que las mismas son instrucciones suministradas a la unidad de ejecución, en contraposición a micro-operaciones resultantes de la decodificación de instrucciones 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 instrucción de 128 bits 2410. Un formato de instrucción compactado de 64 bits 2430 está disponible para algunas instrucciones basándose en la instrucción, las opciones de instrucción y el número de operandos seleccionados. El formato de instrucción de 128 bits nativo 2410 proporciona acceso a todas las opciones de instrucción, mientras que algunas opciones y operaciones están restringidas en el formato de instrucción de 64 bits 2430. Las instrucciones nativas disponibles en el formato de instrucción de 64 bits 2430 varían según la realización. En algunas realizaciones, la instrucción se compacta en parte usando un conjunto de valores de índice en un campo de índice 2413. El hardware de unidad de ejecución consulta un conjunto de tablas de compactación basándose en los valores de índice y usa las salidas de tabla de compactación para reconstruir una instrucción nativa en el formato de instrucción de 128 bits 2410.

Para cada formato, el código de operación de instrucción 2412 define la operación que ha de realizar la unidad de ejecución. Las unidades de ejecución ejecutan cada instrucción en paralelo a lo largo de los múltiples elementos de datos de cada operando. Por ejemplo, en respuesta a una instrucción de suma, la unidad de ejecución realiza una operación de suma simultánea a lo largo de cada canal de color que representa un elemento de textura o un elemento de imagen. Por defecto, la unidad de ejecución ejecuta cada instrucción a lo largo de todos los canales de datos de los operandos. En algunas realizaciones, el campo de control de instrucción 2414 posibilita el control sobre ciertas opciones de ejecución, tales como la selección de canales (por ejemplo, predicación) y el orden de canal de datos (por ejemplo, mezcla). Para las instrucciones de 128 bits 2410, un campo de tamaño de ejecución 2416 limita el número de canales de datos que se ejecutarán en paralelo. En algunas realizaciones, el campo de tamaño de ejecución 2416 no está disponible para su uso en el formato de instrucción compacto de 64 bits 2430.

Algunas instrucciones de unidad de ejecución tienen hasta tres operandos, incluyendo dos operandos de origen, src0 2420, src1 2422 y un destino 2418. En algunas realizaciones, las unidades de ejecución soportan instrucciones de destino dual, donde uno de los destinos está implícito. Las instrucciones de manipulación de datos pueden tener un tercer operando de origen (por ejemplo, SRC2 2424), donde el código de operación de instrucción 2412 determina el número de operandos de origen. El último operando de origen de una instrucción puede ser un valor inmediato (por ejemplo, codificado de manera fija) pasado con la instrucción.

En algunas realizaciones, el formato de instrucción de 128 bits 2410 incluye una información de modo de acceso/dirección 2426 que especifica, por ejemplo, si se usa el modo de direccionamiento de registro directo o el modo de direccionamiento de registro indirecto. Cuando se usa el modo de direccionamiento de registro directo, la dirección de registro de uno o más operandos es proporcionada directamente por bits en la instrucción 2410.

En algunas realizaciones, el formato de instrucción de 128 bits 2410 incluye un campo de modo de acceso/dirección 2426, que especifica un modo de dirección y/o un modo de acceso para la instrucción. En una realización, el modo de acceso para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones soportan modos de acceso que incluyen un modo de acceso alineado de 16 bytes y un modo de acceso alineado de 1 byte, donde la alineación de bytes del modo de acceso determina la alineación de acceso de los operandos de instrucción. Por ejemplo, cuando está en un primer modo, la instrucción 2410 puede usar un direccionamiento alineado en bytes para operandos de origen y destino y, cuando está en un segundo modo, la instrucción 2410 puede usar direccionamiento alineado de 16 bytes para todos los operandos de origen y destino.

En una realización, la parte de modo de dirección del campo de modo de acceso/dirección 2426 determina si la instrucción va a usar direccionamiento directo o indirecto. Cuando se usa el modo de direccionamiento de registro directo, unos bits 2410 en la instrucción proporcionan directamente la dirección de registro de uno o más operandos. Cuando se usa un modo de direccionamiento de registro indirecto, la dirección de registro de uno o más operandos puede calcularse basándose en un valor de registro de dirección y un campo inmediato de dirección en la instrucción.

En algunas realizaciones, las instrucciones se agrupan basándose en los campos de bits del código de operación 2412 para simplificar la decodificación de código de operación 2440. 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. El agrupamiento del código de operación preciso mostrado es simplemente un ejemplo. En algunas realizaciones, un grupo de código de operación de movimiento y lógica 2442 incluye instrucciones de movimiento y lógica de datos (por ejemplo, mover (mov), comparar (cmp)). En algunas realizaciones, el grupo de movimiento y lógica 2442 comparte los cinco bits más significativos (MSB), donde las instrucciones de movimiento (mov) están tienen la forma 0000xxxxb y las instrucciones lógicas tienen la forma 0001xxxxb. Un grupo de instrucciones de control de flujo 2444 (por ejemplo, llamada, salto (jmp)) incluye instrucciones en forma de 0010xxxxb (por ejemplo, 0x20). Un grupo de instrucciones misceláneas 2446 incluye una mezcla de instrucciones, incluyendo instrucciones de sincronización (por ejemplo, espera, envío) en forma de 0011xxxxb (por ejemplo, 0x30). Un grupo de instrucciones de cálculo matemático paralelo 2448 incluye instrucciones aritméticas a nivel de componente (por ejemplo, añadir, multiplicar (mult)) en forma de 0100xxxxb (por ejemplo, 0x40). El grupo de cálculo matemático paralelo 2448 realiza las operaciones aritméticas en paralelo a lo largo de canales de datos. El grupo de cálculo matemático vectorial 2450 incluye instrucciones aritméticas (por ejemplo,

dp4) en forma de 0101xxxxb (por ejemplo, 0x50). El grupo de cálculo matemático vectorial realiza una aritmética tal como cálculos de producto escalar sobre operandos de vectores.

Canalización de gráficos

La **Figura 25** es un diagrama de bloques de otra realización de un procesador de gráficos 2500. Aquellos elementos de la **Figura 25** 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 cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, el procesador de gráficos 2500 incluye una canalización de gráficos 2520, una canalización de medios 2530, un motor de visualización 2540, una lógica de ejecución de hilos 2550 y una canalización de salida de representación 2570. En algunas realizaciones, el procesador de gráficos 2500 es un procesador de gráficos dentro de un sistema de procesamiento de múltiples núcleos que incluye uno o más núcleos de procesamiento de propósito general. El procesador de gráficos es controlado por escrituras de registro en uno o más registros de control (no mostrados) o mediante comandos emitidos al procesador de gráficos 2500 mediante una interconexión en anillo 2502. En algunas realizaciones, la interconexión en anillo 2502 acopla el procesador de gráficos 2500 a otros componentes de procesamiento, tales como otros procesadores de gráficos o procesadores de propósito general. Los comandos desde la interconexión en anillo 2502 se interpretan por un transmisor de envío por flujo continuo de comandos 2503, que suministra instrucciones a componentes individuales de la canalización de gráficos 2520 o la canalización de medios 2530.

En algunas realizaciones, el transmisor por flujo continuo de comandos 2503 dirige la operación de un extractor de vértices 2505 que lee datos de vértice desde memoria y ejecuta comandos de procesamiento de vértices proporcionados por el transmisor por flujo continuo de comandos 2503. En algunas realizaciones, el extractor de vértices 2505 proporciona datos de vértice a un sombreador de vértices 2507, que realiza operaciones de transformación y de iluminación de espacio de coordenadas en cada vértice. En algunas realizaciones, el extractor de vértices 2505 y el sombreador de vértices 2507 ejecutan instrucciones de procesamiento de vértices despachando hilos de ejecución a unidades de ejecución 2552A, 2552B mediante un despachador de hilos 2531.

En algunas realizaciones, las unidades de ejecución 2552A, 2552B son una matriz de procesadores de vectores que tienen un conjunto de instrucciones para realizar operaciones de gráficos y de medios. En algunas realizaciones, las unidades de ejecución 2552A, 2552B tienen una caché L1 2551 adjunta que es específica para cada matriz o que se comparte entre las matrices. La caché puede configurarse como una caché de datos, una caché de instrucciones o una única caché que se subdivide para contener datos e instrucciones en diferentes subdivisiones.

En algunas realizaciones, la canalización de gráficos 2520 incluye componentes de teselación para realizar una teselación acelerada por hardware de objetos 3D. En algunas realizaciones, un sombreador de casco programable 2511 configura las operaciones de teselación. Un sombreador de dominio programable 2517 proporciona una evaluación de extremo posterior del resultado de la teselación. Un teselador 2513 opera en la dirección del sombreador de casco 2511 y contiene una lógica de propósito especial para generar un conjunto de objetos geométricos detallados basándose en un modelo geométrico grueso que se proporciona como entrada a la canalización de gráficos 2520. En algunas realizaciones, si no se usa la teselación, pueden omitirse los componentes de teselación 2511, 2513, 2517.

En algunas realizaciones, unos objetos geométricos completos pueden ser procesados por un sombreador de geometría 2519 mediante uno o más hilos despachados a las unidades de ejecución 2552A, 2552B, o puede avanzar directamente al recortador 2529. En algunas realizaciones, el sombreador de geometría opera sobre objetos geométricos enteros, en lugar de vértices o parches de vértices como en fases previas de la canalización de gráficos. Si la teselación está deshabilitada, el sombreador de geometría 2519 recibe una entrada desde el sombreador de vértices 2507. En algunas realizaciones, el sombreador de geometría 2519 puede programarse mediante un programa sombreador de geometría para realizar una teselación de geometría si las unidades de teselación están deshabilitadas.

Antes de la rasterización, un recortador 2529 procesa datos de vértice. El recortador 2529 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 prueba de rasterizador y de profundidad 2573 en la canalización de salida de representación 2570 despacha sombreadores de píxeles para convertir los objetos geométricos en sus representaciones por píxel. En algunas realizaciones, la lógica de sombreador de píxeles está incluida en la lógica de ejecución de hilos 2550. En algunas realizaciones, una aplicación puede omitir la rasterización y acceder a datos de vértices no rasterizados a través de una unidad de salida de flujo 2523.

El procesador de gráficos 2500 tiene un bus de interconexión, un tejido de interconexión o algún otro mecanismo de interconexión que permite que datos y mensajes pasen entre los componentes principales del procesador. En algunas realizaciones, las unidades de ejecución 2552A, 2552B y la(s) caché(s) 2551 asociada(s), el muestreador de textura y de medios 2554 y la caché de textura/muestreador 2558 se interconectan mediante un puerto de datos 2556 para realizar accesos a memoria y comunicarse con componentes de canalización de salida de representación del

procesador. En algunas realizaciones, el muestreador 2554, las memorias caché 2551, 2558 y las unidades de ejecución 2552A, 2552B tienen, cada uno, rutas de acceso de memoria separadas.

En algunas realizaciones, la canalización de salida de representación 2570 contiene un componente de rasterización y prueba de profundidad 2573 que convierte objetos basados en vértices en una representación asociada basada en píxeles. En algunas realizaciones, la canalización de salida de representador 2570 incluye una unidad generadora de ventanas/enmascaradora para realizar una rasterización de líneas y de triángulos de función fija. En algunas realizaciones también están disponibles una caché de representación 2578 y una caché de profundidad 2579 asociadas. Un componente de operaciones de píxel 2577 realiza operaciones basadas en píxeles sobre los datos, aunque, en algunas instancias, las operaciones de píxel asociadas con operaciones 2D (por ejemplo, transferencias de imagen de bloque de bits con mezcla) son realizadas por el motor 2D 2541, o son sustituidas en el momento de la visualización por el controlador de visualización 2543 usando planos de visualización de superposición. En algunas realizaciones, está disponible una caché L3 compartida 2575 para todos los componentes de gráficos, permitiendo compartir datos sin el uso de memoria de sistema principal.

En algunas realizaciones, la canalización de medios de procesador de gráficos 2530 incluye un motor de medios 2537 y un extremo frontal de vídeo 2534. En algunas realizaciones, el extremo frontal de vídeo 2534 recibe comandos de canalización desde el transmisor por flujo continuo de comandos 2503. En algunas realizaciones, la canalización de medios 2530 incluye un transmisor por flujo continuo de comandos separado. En algunas realizaciones, el extremo frontal de vídeo 2534 procesa comandos de medios antes de enviar el comando al motor de medios 2537. En algunas realizaciones, el motor de medios 2537 incluye una funcionalidad de generación de hilos para generar hilos para despacharlos a la lógica de ejecución de hilos 2550 mediante el despachador de hilos 2531.

En algunas realizaciones, el procesador de gráficos 2500 incluye un motor de visualización 2540. En algunas realizaciones, el motor de visualización 2540 es externo al procesador 2500 y se acopla con el procesador de gráficos mediante la interconexión en anillo 2502, o algún otro bus o tejido de interconexión. En algunas realizaciones, el motor de visualización 2540 incluye un motor 2D 2541 y un controlador de visualización 2543. En algunas realizaciones, el motor de visualización 2540 contiene una lógica de propósito especial capaz de operar independientemente de la canalización 3D. En algunas realizaciones, el controlador de visualización 2543 se acopla con un dispositivo de visualización (no mostrado), que puede ser un dispositivo de visualización integrado en sistema, como en un ordenador portátil, o un dispositivo de visualización externo adjunto mediante un conector de dispositivo de visualización.

En algunas realizaciones, la canalización de gráficos 2520 y la canalización de medios 2530 pueden configurarse para realizar operaciones basándose en múltiples interfaces de programación de gráficos y de medios y no son específicas de ninguna interfaz de programación de aplicaciones (API). En algunas realizaciones, el software de controlador para el procesador de gráficos traduce llamadas de API que son específicas de una biblioteca de medios o de gráficos particular a comandos que pueden ser procesados por el procesador de gráficos. En algunas realizaciones, se proporciona soporte para la biblioteca Open Graphics (OpenGL) y Open Computing Language (OpenCL) de Khronos Group, la biblioteca Direct3D de Microsoft Corporation, o se puede proporcionar soporte tanto para OpenGL como para D3D. También puede proporcionarse soporte para la Biblioteca de Visión Informática de Código Abierto (OpenCV). También se soportaría una API futura con una canalización 3D compatible si puede hacerse un mapeo desde la canalización de la API futura a la canalización del procesador de gráficos.

Programación de canalización de gráficos

La **Figura 26A** es un diagrama de bloques que ilustra un formato de comando de procesador de gráficos 2600 de acuerdo con algunas realizaciones. La **Figura 26B** es un diagrama de bloques que ilustra una secuencia de comandos de procesador de gráficos 2610 de acuerdo con una realización. Los recuadros con línea continua en la **Figura 26A** ilustran los componentes que se incluyen, en general, en un comando de gráficos, mientras que las líneas discontinuas incluyen componentes que son opcionales o que solo se incluyen en un subconjunto de los comandos de gráficos. El formato de comando de procesador gráfico 2600 ilustrativo de la **Figura 26A** incluye campos de datos para identificar un cliente objetivo 2602 del comando, un código de operación (cód. de ope.) de comando 2604 y los datos 2606 relevantes para el comando. También se incluyen un subcódigo de operación 2605 y un tamaño de comando 2608 en algunos comandos.

En algunas realizaciones, el cliente 2602 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 de 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 del procesador de gráficos incluyen una unidad de interfaz de memoria, una unidad de representación, una unidad 2D, una unidad 3D y una unidad multimedia. Cada unidad de cliente tiene una canalización de procesamiento correspondiente que procesa los comandos. Una vez que el comando es recibido por la unidad de cliente, la unidad de cliente lee el código de operación 2604 y, si está presente, el sub-código de operación 2605 para determinar la operación que hay que realizar. La unidad de cliente realiza el comando usando información en el campo de datos 2606. Para algunos comandos, se espera que un tamaño de comando explícito 2608 especifique el tamaño del comando. En algunas realizaciones, el analizador de

comandos determina automáticamente el tamaño de al menos algunos de los comandos basándose en el código de operación de comando. En algunas realizaciones, los comandos se alinean mediante múltiplos de una palabra doble.

El diagrama de flujo de la **Figura 26B** muestra una secuencia de comandos de procesador de gráficos 2610 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 establecer, ejecutar y terminar un conjunto de operaciones de gráficos. Se muestra y se describe una secuencia de comandos de muestra solo a fines de ejemplo, debido a que las realizaciones no se limitan a estos comandos específicos o a esta secuencia de comandos. Además, los comandos pueden emitirse como un lote de comandos en una secuencia de comandos, de manera que el procesador de gráficos procesará la secuencia de comandos de manera al menos parcialmente concurrente.

En algunas realizaciones, la secuencia de comandos de procesador de gráficos 2610 puede comenzar con un comando de vaciado de canalización 2612 para hacer que alguna canalización de gráficos activa complete los comandos actualmente pendientes para la canalización. En algunas realizaciones, la canalización 3D 2622 y la canalización de medios 2624 no operan concurrentemente. El vaciado de canalización se realiza para hacer que la canalización de gráficos activa complete cualquier comando pendiente. En respuesta a un vaciado de canalización, el analizador de comandos del procesador gráfico pausará el procesamiento de comandos hasta que los motores de dibujo activos completen las operaciones pendientes y se invaliden las memorias caché de lectura relevantes. Opcionalmente, cualquier dato en la caché de representación que se marque como 'sucio' puede vaciarse a memoria. En algunas realizaciones, el comando de vaciado de canalización 2612 puede usarse para la sincronización de canalización o antes de colocar el procesador gráfico en un estado de bajo consumo de energía.

En algunas realizaciones, se usa un comando de selección de canalización 2613 cuando una secuencia de comandos requiere que el procesador de gráficos conmute explícitamente entre canalizaciones. En algunas realizaciones, un comando de selección de canalización 2613 solo se requiere una vez dentro de un contexto de ejecución antes de emitir comandos de canalización a menos que el contexto sea emitir comandos para ambas canalizaciones. En algunas realizaciones, un comando de vaciado de canalización 2612 se requiere inmediatamente antes de un conmutador de canalización por medio del comando de selección de canalización 2613.

En algunas realizaciones, un comando de control de canalización 2614 configura una canalización de gráficos para la operación y se usa para programar la canalización 3D 2622 y la canalización de medios 2624. En algunas realizaciones, el comando de control de canalización 2614 configura el estado de canalización para la canalización activa. En una realización, el comando de control de canalización 2614 se usa para la sincronización de canalización y para limpiar datos de una o más memorias caché dentro de la canalización activa antes de procesar un lote de comandos.

En algunas realizaciones, los comandos para el estado de memoria intermedia de retorno 2616 se usan para configurar un conjunto de memorias intermedias de retorno para que las canalizaciones respectivas escriban datos. Algunas operaciones de canalización requieren la asignación, selección o configuración de una o más memorias intermedias de retorno en las que las operaciones escriben datos intermedios durante el procesamiento. En algunas realizaciones, el procesador de gráficos también usa una o más memorias intermedias de retorno para almacenar datos de salida y realizar comunicación de hilos cruzada. En algunas realizaciones, configurar el estado de memoria intermedia de retorno 2616 incluye seleccionar el tamaño y número de memorias intermedias de retorno que usar para un conjunto de operaciones de canalización.

Los comandos restantes en la secuencia de comandos difieren basándose en la canalización activa para las operaciones. Basándose en una determinación de canalización 2620, la secuencia de comandos se adapta a la canalización 3D 2622 comenzando con el estado de canalización 3D 2630, o a la canalización de medios 2624 comenzando en el estado de canalización de medios 2640.

Los comandos para el estado de canalización 3D 2630 incluyen comandos de ajuste de estado 3D para estado de memoria intermedia de vértice, estado de elemento de vértice, estado de color constante, estado de memoria intermedia de profundidad y otras variables de estado que han de configurarse antes de que se procesen los comandos de primitiva 3D. Los valores de estos comandos se determinan, al menos en parte, basándose en la API 3D particular en uso. En algunas realizaciones, los comandos de estado de canalización 3D 2630 también pueden deshabilitar u omitir selectivamente ciertos elementos de la canalización si no se van a usar estos elementos.

En algunas realizaciones, el comando de primitivas 3D 2632 se usa para enviar primitivas 3D que serán procesadas mediante la canalización 3D. Los comandos y parámetros asociados que se pasan al procesador de gráficos a través del comando de primitiva 3D 2632 se reenvían a la función de extracción de vértices en la canalización de gráficos. La función de extracción de vértices usa los datos de comando de la primitiva 3D 2632 para generar estructuras de datos de vértice. Las estructuras de datos de vértice se almacenan en una o más memorias intermedias de retorno. En algunas realizaciones, el comando de la primitiva 3D 2632 se usa para realizar operaciones de vértice sobre primitivas 3D mediante sombreadores de vértices. Para procesar sombreadores de vértices, la canalización 3D 2622 despacha hilos de ejecución de sombreador a unidades de ejecución de procesador de gráficos.

En algunas realizaciones, la canalización 3D 2622 se desencadena a través de un comando o evento de ejecución 2634. En algunas realizaciones, una escritura de registro desencadena una ejecución de comando. En algunas realizaciones, la ejecución se desencadena por medio de un comando 'ir' o 'poner en marcha' en la secuencia de comandos. En una realización, la ejecución del comando se desencadena usando un comando de sincronización de canalización para vaciar la secuencia de comandos a través de la canalización de gráficos. La canalización 3D realizará un procesamiento de geometría para las primitivas 3D. Una vez que se han completado las operaciones, los objetos geométricos resultantes se rasterizan y el motor de píxeles da color a los píxeles resultantes. También pueden incluirse comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo posterior de píxeles para esas operaciones.

En algunas realizaciones, la secuencia de comandos de procesador de gráficos 2610 sigue la ruta de canalización de medios 2624 cuando se llevan a cabo operaciones de medios. En general, el uso específico y la forma de programación para la canalización de medios 2624 depende de los medios o de las operaciones de cálculo que se van a realizar. Operaciones de decodificación de medios específicas pueden descargarse a la canalización de medios durante la decodificación de medios. En algunas realizaciones, puede omitirse también la canalización 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 propósito general. En una realización, la canalización de medios también incluye elementos para operaciones de unidad de procesador de gráficos de propósito general (GPGPU), donde el procesador de gráficos se usa para realizar operaciones de vector SIMD usando programas sombreadores computacionales que no están relacionados explícitamente con la representación de primitivas de gráficos.

En algunas realizaciones, se configura la canalización de medios 2624 de una manera similar que la canalización 3D 2622. Un conjunto de comandos para configurar el estado de canalización de medios 2640 se despacha o coloca en una cola de comandos antes de los comandos de objeto de medios 2642. En algunas realizaciones, los comandos para el estado de canalización de medios 2640 incluyen datos para configurar los elementos de canalización de medios que se usarán para procesar los objetos de medios. Esto incluye datos para configurar la lógica de decodificación y codificación de vídeo dentro de la canalización de medios, tal como el formato de codificación o decodificación. En algunas realizaciones, los comandos para el estado de canalización de medios 2640 también soportan el uso de uno o más punteros a elementos de estado "indirectos" que contienen un lote de configuraciones de estado.

En algunas realizaciones, los comandos de objeto de medios 2642 suministran punteros a objetos de medios para su procesamiento por la canalización de medios. Los objetos de medios incluyen memorias intermedias de memoria que contienen datos de vídeo que hay que procesar. En algunas realizaciones, todos los estados de canalización de medios deben ser válidos antes de emitir un comando de objeto de medios 2642. Una vez que se ha configurado el estado de canalización y los comandos de objeto de medios 2642 se han puesto en cola, la canalización de medios 2624 se desencadena mediante un comando de ejecución 2644 o un evento de ejecución equivalente (por ejemplo, una escritura de registro). La salida desde la canalización de medios 2624 puede post-procesarse, a continuación, mediante operaciones proporcionadas por la canalización 3D 2622 o la canalización de medios 2624. En algunas realizaciones, las operaciones de GPGPU se configuran y se ejecutan de una manera similar a la de las operaciones de medios.

Arquitectura de software de gráficos

La **Figura 27** ilustra una arquitectura de software de gráficos ilustrativa para un sistema de procesamiento de datos 2700 de acuerdo con algunas realizaciones. En algunas realizaciones, la arquitectura de software incluye una aplicación de gráficos 3D 2710, un sistema operativo 2720 y al menos un procesador 2730. En algunas realizaciones, el procesador 2730 incluye un procesador de gráficos 2732 y uno o más núcleos de procesador de propósito general 2734. Cada uno de la aplicación de gráficos 2710 y el sistema operativo 2720 se ejecutan en la memoria de sistema 2750 del sistema de procesamiento de datos.

En algunas realizaciones, la aplicación de gráficos 3D 2710 contiene uno o más programas sombreadores que incluyen instrucciones de sombreador 2712. Las instrucciones de lenguaje de sombreador pueden estar en un lenguaje de sombreador de alto nivel, tal como el Lenguaje de Sombreador de Alto Nivel (HLSL) o el Lenguaje de Sombreador OpenGL (GLSL). La aplicación también incluye las instrucciones ejecutables 2714 en un lenguaje máquina adecuado para su ejecución por el/los núcleo(s) de procesador de propósito general 2734. La aplicación también incluye los objetos de gráficos 2716 definidos por datos de vértice.

En algunas realizaciones, el sistema operativo 2720 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo similar a UNIX patentado o un sistema operativo similar a UNIX de código abierto que usa una variante del núcleo Linux. El sistema operativo 2720 puede soportar una API de gráficos 2722 tal como la API Direct3D o la API OpenGL. Cuando está en uso la API de Direct3D, el sistema operativo 2720 usa un compilador de sombreador de extremo frontal 2724 para compilar cualquier instrucción de sombreador 2712 en HLSL a un lenguaje de sombreador de nivel inferior. La compilación puede ser una compilación justo a tiempo (JIT) o la aplicación puede llevar a cabo una precompilación de sombreador. En algunas realizaciones, los sombreadores de alto nivel se compilan a sombreadores de bajo nivel durante la compilación de la aplicación de gráficos 3D 2710.

En algunas realizaciones, el controlador de gráficos en modo de usuario 2726 contiene un compilador de sombreador de extremo posterior 2727 para convertir las instrucciones de sombreador 2712 en una representación específica de hardware. Cuando está en uso la API de OpenGL, las instrucciones de sombreador 2712 en el lenguaje de alto nivel GLSL se pasan a un controlador de gráficos de modo de usuario 2726 para su compilación. En algunas realizaciones, el controlador de gráficos de modo de usuario 2726 usa las funciones de modo de núcleo de sistema operativo 2728 para comunicarse con un controlador de gráficos de modo de núcleo 2729. En algunas realizaciones, el controlador de gráficos de modo de núcleo 2729 se comunica con el procesador de gráficos 2732 para despachar comandos e instrucciones.

Implementaciones de núcleo de IP

Uno o más aspectos de al menos una realización pueden implementarse mediante un código representativo almacenado en un medio legible por máquina que representa y/o define la lógica dentro de un circuito integrado, tal como un procesador. Por ejemplo, el medio legible por máquina puede incluir instrucciones que representan una lógica diversa dentro del procesador. Cuando son leídas por una máquina, las instrucciones pueden hacer que la máquina 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 pueden almacenarse en un medio legible por máquina tangible como un modelo de hardware que describe la estructura del circuito integrado. El modelo de hardware puede suministrarse a 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 operaciones descritas en asociación con cualquiera de las realizaciones descritas en el presente documento.

La **Figura 28** es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP 2800 que puede usarse para fabricar un circuito integrado para realizar las operaciones de acuerdo con una realización. El sistema de desarrollo de núcleo de IP 2800 puede usarse para generar diseños reutilizables modulares que pueden incorporarse en un diseño más grande o usarse para construir un circuito integrado entero (por ejemplo, un circuito de SOC integrado). Una instalación de diseño 2830 puede generar una simulación de software 2810 de un diseño de núcleo de IP en un lenguaje de programación de alto nivel (por ejemplo, C/C++). La simulación de software 2810 puede usarse para diseñar, someter a prueba y verificar el comportamiento del núcleo de IP usando un modelo de simulación 2812. El modelo de simulación 2812 puede incluir simulaciones funcionales, de comportamiento y/o de temporización. Puede crearse o sintetizarse, a continuación, un diseño de nivel de transferencia de registro (RTL) 2815 a partir del modelo de simulación 2812. El diseño de RTL 2815 es una abstracción del comportamiento del circuito integrado que modela el flujo de señales digitales entre registros de hardware, incluyendo la lógica asociada realizada usando las señales digitales modeladas. Además de un diseño de RTL 2815, también pueden crearse, diseñarse o sintetizarse diseños de nivel inferior al nivel de lógica o al nivel de transistores. Por tanto, los detalles particulares del diseño y simulación inicial pueden variar.

El diseño de RTL 2815, o un equivalente, puede ser sintetizado adicionalmente por la instalación de diseño para dar un modelo de hardware 2820, que puede estar en un lenguaje de descripción de hardware (HDL) o alguna otra representación de datos de diseño físico. El HDL puede simularse o someterse a prueba 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 2865 usando la memoria no volátil 2840 (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, por Internet) a través de una conexión cableada 2850 o una conexión inalámbrica 2860. La instalación de fabricación 2865 puede fabricar, a continuación, un circuito integrado que se basa, al menos en parte, en el diseño de núcleo de IP. El circuito integrado fabricado puede configurarse para realizar operaciones de acuerdo con al menos una realización descrita en el presente documento.

Circuito integrado de sistema en un chip ilustrativo

Las **Figuras 29-31** ilustran 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, pueden incluirse otros circuitos y lógica, incluyendo procesadores/núcleos de gráficos adicionales, controladores de interfaz de periféricos o núcleos de procesador de propósito general.

La **Figura 29** es un diagrama de bloques que ilustra un circuito integrado de sistema en un chip 2900 ilustrativo que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El circuito integrado 2900 ilustrativo incluye uno o más procesadores de aplicaciones 2905 (por ejemplo, unas CPU), al menos un procesador de gráficos 2910, y puede incluir adicionalmente un procesador de imágenes 2915 y/o un procesador de vídeo 2920, cualquiera de los cuales puede ser un núcleo de IP modular desde las mismas o múltiples instalaciones de diseño diferentes. El circuito integrado 2900 incluye una lógica de bus o de periféricos que incluye un controlador de USB 2925, un controlador de UART 2930, un controlador de SPI/SDIO 2935 y un controlador de I^2S/I^2C 2940. Adicionalmente, el circuito integrado puede incluir un dispositivo de visualización 2945 acoplado a uno o más de un controlador de interfaz multimedia de alta definición (HDMI) 2950 y una interfaz de visualización de interfaz de procesador de industria móvil

(MIPI) 2955. El almacenamiento puede ser proporcionado por un subsistema de memoria flash 2960 que incluye memoria flash y un controlador de memoria flash. La interfaz de memoria puede proporcionarse mediante un controlador de memoria 2965 para el acceso a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad integrado 2970.

La **Figura 30** es un diagrama de bloques que ilustra un procesador de gráficos 3010 ilustrativo de un circuito integrado de sistema en un chip que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El procesador de gráficos 3010 puede ser una variante del procesador de gráficos 2910 de la **Figura 29.** El procesador de gráficos 3010 incluye un procesador de vértices 3005 y uno o más procesadores de fragmentos 3015A-3015N (por ejemplo, 3015A, 3015B, 3015C, 3015D a 3015N-1 y 3015N). El procesador de gráficos 3010 puede ejecutar diferentes programas sombreadores por medio de una lógica diferente, de tal forma que el procesador de vértices 3005 se optimiza para ejecutar operaciones para programas sombreadores de vértices, mientras que los uno o más procesadores de fragmentos 3015A-3015N ejecutan operaciones de sombreado de fragmentos (por ejemplo, píxeles) para programas sombreadores de fragmentos o de píxeles. El procesador de vértices 3005 realiza la fase de procesamiento de vértices de la canalización de gráficos 3D y genera primitivas y datos de vértice. El/los procesador(es) de fragmentos 3015A-3015N usa(n) los datos de primitiva y de vértice generados por el procesador de vértices 3005 para producir una memoria intermedia de fotogramas que se visualiza en un dispositivo de visualización. En una realización, el/los procesador(es) de fragmentos 3015A-3015N se optimiza(n) para ejecutar programas sombreadores de fragmentos según lo previsto en la API de OpenGL, que pueden usarse para llevar a cabo operaciones similares como un programa sombreador de píxeles según lo previsto en la API de Direct 3D.

El procesador de gráficos 3010 incluye adicionalmente una o más unidades de gestión de memoria (MMU) 3020A-3020B, caché(s) 3025A-3025B e interconexión(es) de circuito 3030A-3030B. La(s) una o más MMU 3020A-3020B prevé(n) un mapeo de dirección virtual a física para el procesador de gráficos 3010, incluyendo para el procesador de vértices 3005 y/o el/los procesador(es) de fragmentos 3015A-3015N, que pueden hacer referencia a datos de vértice o de imagen/textura almacenados en memoria, además de datos de vértice o de imagen/textura almacenados en la(s) una o más caché(s) 3025A-3025B. En una realización, las una o más MMU 3020A-3020B pueden sincronizarse con otras MMU dentro del sistema, incluyendo una o más MMU asociadas a los uno o más procesadores de aplicación 2905, el procesador de imagen 2915 y/o el procesador de vídeo 2920 de la **Figura 29**, de manera que cada procesador 2905-2920 puede participar en un sistema de memoria virtual compartido o unificado. Las una o más interconexiones de circuito 3030A-3030B posibilitan que el procesador de gráficos 3010 interaccione con otros núcleos de IP dentro del SoC, o bien mediante un bus interno del SoC o bien mediante una conexión directa, de acuerdo con unas realizaciones.

La **Figura 31** es un diagrama de bloques que ilustra un procesador de gráficos 3110 ilustrativo adicional de un circuito integrado de sistema en un chip que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El procesador de gráficos 3110 puede ser una variante del procesador de gráficos 2910 de la **Figura 29.** El procesador de gráficos 3110 incluye las una o más MMU 3020A-3020B, caché(s) 3025A-3025B e interconexión(es) de circuito 3030A-3030B del circuito integrado 3000 de la **Figura 30**.

El procesador de gráficos 3110 incluye uno o más núcleos de sombreador 3115A-3115N (por ejemplo, 3115A, 3115B, 3115C, 3115D, 3115E, 3115F a 3015N-1 y 3015N), lo que prevé una arquitectura de núcleo de sombreador unificada en la que un único núcleo o tipo o núcleo puede ejecutar todos los tipos de código sombreador programable, 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 sombreador presentes puede variar entre realizaciones e implementaciones. Además, el procesador de gráficos 3110 incluye un gestor de tareas inter-núcleo 3105, que actúa como un despachador de hilos para enviar hilos de ejecución a uno o más núcleos de sombreado 3115A-3115N. El procesador de gráficos 3110 incluye adicionalmente una unidad de teselado 3118 para acelerar las operaciones de teselado para la representación basada en teselas, en la que las operaciones de representación para una escena se subdividen en el espacio de la imagen. La representación basada en teselas se puede usar para aprovechar la coherencia espacial local dentro de una escena o para optimizar el uso de cachés internas.

Las referencias a "una realización", "realización de ejemplo", "diversas realizaciones", etc., indican que la(s) realización(es) así descritas pueden incluir rasgos, estructuras o características particulares, pero no todas las realizaciones incluyen necesariamente los rasgos, estructuras o características particulares. Además, algunas realizaciones pueden tener algunas, todas o ninguna de las características descritas para otras realizaciones.

En la siguiente descripción y las reivindicaciones, puede usarse el término "acoplado" junto con sus derivadas. "Acoplado" se usa para indicar que dos o más elementos cooperan o interactúan entre sí, pero pueden tener o no componentes físicos o eléctricos intermedios entre ellos.

Como se usa en las reivindicaciones, a menos que se especifique lo contrario, el uso de los adjetivos ordinales "primero", "segundo", "tercero", etc., para describir un elemento común, simplemente indica que se hace referencia a diferentes instancias de elementos similares, y no pretenden implicar que los elementos así descritos deban estar en una secuencia determinada, ya sea temporal, espacial, en clasificación o de cualquier otra manera.

Las siguientes cláusulas y/o ejemplos se refieren a realizaciones o ejemplos adicionales. Los detalles específicos en los ejemplos pueden usarse en cualquier parte en una o más realizaciones. Las diversas características de las diferentes realizaciones o ejemplos pueden combinarse de manera diversa con algunas características incluidas y otras excluidas para adecuarse a una diversidad de aplicaciones diferentes. Los ejemplos pueden incluir materia objeto tal como un método, medios para realizar actos del método, al menos un medio legible por máquina que incluye instrucciones que, cuando son realizadas por una máquina, hacen que la máquina realice actos del método, o de un aparato o sistema para facilitar una comunicación híbrida de acuerdo con realizaciones y ejemplos descritos en el presente documento.

Algunas realizaciones se refieren al Ejemplo 1 que incluye un aparato para facilitar el reconocimiento, la reidentificación y la seguridad en el aprendizaje automático en máquinas autónomas, comprendiendo el aparato: lógica de detección/observación, según es facilitado por o incorporada al menos parcialmente en un procesador, para facilitar que una cámara detecte uno o más objetos dentro de una proximidad física, incluyendo los uno o más objetos un persona, e incluyendo la proximidad física una casa, en donde la detección incluye capturar una o más imágenes de una o más partes del cuerpo de la persona; lógica de extracción y comparación, según es facilitado por o incorporada al menos parcialmente en el procesador, para extraer características corporales para crear un perfil de la persona basándose en las una o más partes del cuerpo, en donde la lógica de extracción y comparación sirve además para comparar las características corporales extraídas con vectores de características almacenados en una base de datos; y lógica de reidentificación y modelo, según es facilitado por o incorporada al menos parcialmente en el procesador, para construir un modelo de clasificación basándose en las características corporales extraídas a lo largo de un período de tiempo para facilitar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial de la persona.

El ejemplo 2 incluye la materia objeto del ejemplo 1, que comprende además lógica de reconocimiento y registro, según es facilitado por o incorporada al menos parcialmente en el procesador, para realizar un registro inicial de la persona basándose en el reconocimiento facial, en donde la lógica de extracción y comparación sirve además para poner en correspondencia las características corporales extraídas con el reconocimiento facial después del registro inicial.

El ejemplo 3 incluye la materia objeto de los ejemplos 1-2, que comprende además lógica de almacenamiento y entrenamiento, según es facilitado por o incorporada al menos parcialmente en el procesador, para entrenar un clasificador asociado con una red neuronal para garantizar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial.

El ejemplo 4 incluye la materia objeto de los ejemplos 1-3, que comprende además lógica de autenticación, según es facilitado por o incorporada al menos parcialmente en el procesador, para insertar una o más comprobaciones de verificación en una o más capas de la red neuronal para comprobar la integridad de la red neuronal en cada una de las una o más capas para evitar ataques maliciosos.

El ejemplo 5 incluye la materia objeto de los ejemplos 1-4, que comprende además lógica de ejecución paralela, según es facilitado por o incorporada al menos parcialmente en el procesador, para facilitar ejecuciones paralelas separadas de la red neuronal y una aplicación de software asociada con el procesador que incluye un procesador de gráficos si la aplicación de software está sujeta a ciberataques, en donde la red neuronal se protege en una primera unidad de ejecución, mientras que la aplicación de software está en cuarentena en una segunda unidad de ejecución.

El ejemplo 6 incluye la materia objeto de los ejemplos 1-5, que comprende además lógica de comparación de salida, según es facilitado por o incorporada al menos parcialmente en el procesador, para comparar una salida de la red neuronal con una decisión pendiente de una entidad de toma de decisiones, en donde, basándose en la salida de la red neuronal, la decisión pendiente está al menos una de alterada, suspendida o mantenida.

El ejemplo 7 incluye la materia objeto de los ejemplos 1-6, en donde el procesador de gráficos se ubica conjuntamente con un procesador de aplicaciones en un paquete de semiconductores común.

Algunas realizaciones se refieren al Ejemplo 8 que incluye un método para facilitar el reconocimiento, la reidentificación y la seguridad en el aprendizaje automático en máquinas autónomas, comprendiendo el método: facilitar que una cámara detecte uno o más objetos dentro de una proximidad física, incluyendo los uno o más objetos un persona, e incluyendo la proximidad física una casa, en donde la detección incluye capturar una o más imágenes de una o más partes del cuerpo de la persona; extraer características corporales basándose en las una o más partes del cuerpo; comparar las características corporales extraídas con vectores de características almacenados en una base de datos; y construir un modelo de clasificación basándose en las características corporales extraídas a lo largo de un período de tiempo para facilitar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial de la persona.

El ejemplo 9 incluye la materia objeto del ejemplo 8, que comprende además realizar un registro inicial de la persona basándose en el reconocimiento facial; y poner en correspondencia las características corporales extraídas con el reconocimiento facial después del registro inicial.

El ejemplo 10 incluye la materia objeto de los ejemplos 8-9, que comprende además entrenar un clasificador asociado con una red neuronal para garantizar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial.

El ejemplo 11 incluye la materia objeto de los ejemplos 8-10, que comprende además insertar una o más comprobaciones de verificación en una o más capas de la red neuronal para comprobar la integridad de la red neuronal en cada una de las una o más capas para evitar ataques maliciosos.

El ejemplo 12 incluye la materia objeto de los ejemplos 8-11, que comprende además facilitar ejecuciones paralelas separadas de la red neuronal y una aplicación de software asociada con un procesador de gráficos si la aplicación de software está sujeta a ciberataques, en donde la red neuronal se protege en una primera unidad de ejecución, mientras que la aplicación de software está en cuarentena en una segunda unidad de ejecución.

El ejemplo 13 incluye la materia objeto de los ejemplos 8-12, que comprende además comparar una salida de la red neuronal con una decisión pendiente de una entidad de toma de decisiones, en donde, basándose en la salida de la red neuronal, la decisión pendiente está al menos una de alterada, suspendida o mantenida.

El ejemplo 14 incluye la materia objeto de los ejemplos 8-13, en donde el procesador de gráficos se ubica conjuntamente con un procesador de aplicaciones en un paquete de semiconductores común.

Algunas realizaciones se refieren al Ejemplo 15 que incluye un sistema de procesamiento de gráficos que comprende un dispositivo informático que tiene memoria acoplada a un procesador, el procesador para: facilitar que una cámara detecte uno o más objetos dentro de una proximidad física, incluyendo los uno o más objetos un persona, e incluyendo la proximidad física una casa, en donde la detección incluye capturar una o más imágenes de una o más partes del cuerpo de la persona; extraer características corporales basándose en las una o más partes del cuerpo; comparar las características corporales extraídas con vectores de características almacenados en una base de datos; y construir un modelo de clasificación basándose en las características corporales extraídas a lo largo de un período de tiempo para facilitar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial de la persona.

El ejemplo 16 incluye la materia objeto del ejemplo 15, en donde el procesador sirve además para realizar un registro inicial de la persona basándose en el reconocimiento facial; y poner en correspondencia las características corporales extraídas con el reconocimiento facial después del registro inicial.

El ejemplo 17 incluye la materia objeto de los ejemplos 15-16, en donde el procesador sirve además para entrenar un clasificador asociado con una red neuronal para garantizar el reconocimiento o la reidentificación de la persona independientemente del reconocimiento facial.

El ejemplo 18 incluye la materia objeto de los ejemplos 15-17, en donde el procesador sirve además para insertar una o más comprobaciones de verificación en una o más capas de la red neuronal para comprobar la integridad de la red neuronal en cada una de las una o más capas para evitar ataques maliciosos.

El ejemplo 19 incluye la materia objeto de los ejemplos 15-18, en donde el procesador sirve además para facilitar ejecuciones paralelas separadas de la red neuronal y una aplicación de software asociada con un procesador de gráficos si la aplicación de software está sujeta a ciberataques, en donde la red neuronal se protege en una primera unidad de ejecución, mientras que la aplicación de software está en cuarentena en una segunda unidad de ejecución, en donde el procesador de gráficos se ubica conjuntamente con un procesador de aplicaciones en un paquete de semiconductores común.

El ejemplo 20 incluye la materia objeto de los ejemplos 15-19, en donde el procesador sirve además para comparar una salida de la red neuronal con una decisión pendiente de una entidad de toma de decisiones, en donde, basándose en la salida de la red neuronal, la decisión pendiente está al menos una de alterada, suspendida o mantenida.

El ejemplo 21 incluye la materia objeto de los ejemplos 15-20, en donde el procesador de gráficos se ubica conjuntamente con un procesador de aplicaciones en un paquete de semiconductores común.

El ejemplo 22 incluye al menos un medio legible por máquina no transitorio o tangible que comprende una pluralidad de instrucciones, cuando se ejecutan en un dispositivo informático, para implementar o realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 23 incluye al menos un medio legible por máquina que comprende una pluralidad de instrucciones, cuando se ejecutan en un dispositivo informático, para implementar o realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 24 incluye un sistema que comprende un mecanismo para implementar o realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 25 incluye un aparato que comprende medios para realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 26 incluye un dispositivo informático dispuesto para implementar o realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 27 incluye un dispositivo de comunicaciones dispuesto para implementar o realizar un método según cualquiera de las reivindicaciones o ejemplos 8-14.

El ejemplo 28 incluye al menos un medio legible por máquina que comprende una pluralidad de instrucciones, cuando se ejecutan en un dispositivo informático, para implementar o realizar un método o materializar un aparato según cualquiera de las reivindicaciones anteriores.

El ejemplo 29 incluye al menos un medio legible por máquina no transitorio o tangible que comprende una pluralidad de instrucciones, cuando se ejecutan en un dispositivo informático, para implementar o realizar un método o materializar un aparato según cualquiera de las reivindicaciones anteriores.

El ejemplo 30 incluye un sistema que comprende un mecanismo para implementar o realizar un método o materializar un aparato según cualquiera de las reivindicaciones anteriores.

El ejemplo 31 incluye un aparato que comprende medios para realizar un método según cualquiera de las reivindicaciones anteriores.

El ejemplo 32 incluye un dispositivo informático dispuesto para implementar o realizar un método o materializar un aparato según cualquiera de las reivindicaciones anteriores.

El ejemplo 33 incluye un dispositivo de comunicaciones dispuesto para implementar o realizar un método o materializar un aparato según cualquiera de las reivindicaciones anteriores.

REIVINDICACIONES

1. Un aparato (600) para facilitar el reconocimiento, la reidentificación y la seguridad en el aprendizaje automático en máquinas autónomas, comprendiendo el aparato:
lógica de detección/observación (701) para hacer que una cámara detecte uno o más objetos dentro de una casa, incluyendo los uno o más objetos una persona, en donde la detección incluye capturar una o más imágenes de una o más partes del cuerpo de la persona;
lógica de extracción y comparación (713) para extraer características corporales para crear un perfil de la persona basándose en las una o más partes del cuerpo, en donde la lógica de extracción y comparación (713) sirve además para comparar las características corporales extraídas con vectores de características almacenados en una base de datos;
lógica de reidentificación y modelado (715) para construir un modelo de clasificación basándose en las características corporales extraídas a lo largo de un período de tiempo para facilitar el reconocimiento y la reidentificación de la persona independientemente del reconocimiento facial de la persona, en donde, tras la captura de una cierta cantidad de datos que son las imágenes de las una o más partes del cuerpo de la persona a lo largo del período de tiempo, el modelo de clasificación construido por la lógica de reidentificación y modelado (715) alcanza un punto con un perfilado suficiente de la persona, en donde, cuando se alcanza dicho punto, se reconoce y se reidentifica a la persona basándose en el módulo de clasificación construido sin necesidad del reconocimiento facial de la persona y de forma completamente independiente del mismo; y
lógica de reconocimiento y registro (711) para realizar un registro inicial de la persona basándose en el reconocimiento facial y para continuar registrando o inscribiendo una cara de la persona cada vez que se detecta o se reconoce la cara en la base de datos,
en donde la lógica de extracción y comparación (713) sirve además para poner en correspondencia las características corporales extraídas con la persona que está siendo reconocida por reconocimiento facial después del registro inicial.

2. El aparato de la reivindicación 1, que comprende además:
lógica de almacenamiento y entrenamiento (717) para entrenar un clasificador asociado con una red neuronal para garantizar el reconocimiento y la reidentificación de la persona independientemente del reconocimiento facial.

3. Un método para facilitar el reconocimiento, la reidentificación y la seguridad en el aprendizaje automático en máquinas autónomas, comprendiendo el método:
hacer que una cámara detecte uno o más objetos dentro de una casa, incluyendo los uno o más objetos una persona, en donde la detección incluye capturar una o más imágenes de una o más partes del cuerpo de la persona;
extraer características corporales para crear un perfil de la persona basándose en las una o más partes del cuerpo;
comparar las características corporales extraídas con vectores de características almacenados en una base de datos;
construir un modelo de clasificación basándose en las características corporales extraídas a lo largo de un período de tiempo para facilitar el reconocimiento y la reidentificación de la persona independientemente del reconocimiento facial de la persona, en donde, tras la captura de una cierta cantidad de datos que son las imágenes de las una o más partes del cuerpo de la persona a lo largo del período de tiempo, alcanzar un punto con un perfilado suficiente de la persona, en donde, cuando se alcanza dicho punto, se reconoce y se reidentifica a la persona basándose en el modelo de clasificación construido sin necesidad del reconocimiento facial de la persona y de forma completamente independiente del mismo;
realizar un registro inicial de la persona basándose en el reconocimiento facial y continuar registrando o inscribiendo una cara de la persona cada vez que se detecta o se reconoce la cara en la base de datos; y
poner en correspondencia las características corporales extraídas con la persona que está siendo reconocida por reconocimiento facial después del registro inicial.

4. El método de la reivindicación 3, que comprende además:
entrenar un clasificador asociado con una red neuronal para garantizar el reconocimiento y la reidentificación de la persona independientemente del reconocimiento facial.

5. Al menos un medio legible por máquina que comprende una pluralidad de instrucciones que, cuando se ejecutan en un dispositivo informático, configuran el dispositivo informático para realizar un método según la reivindicación 3 o 4.

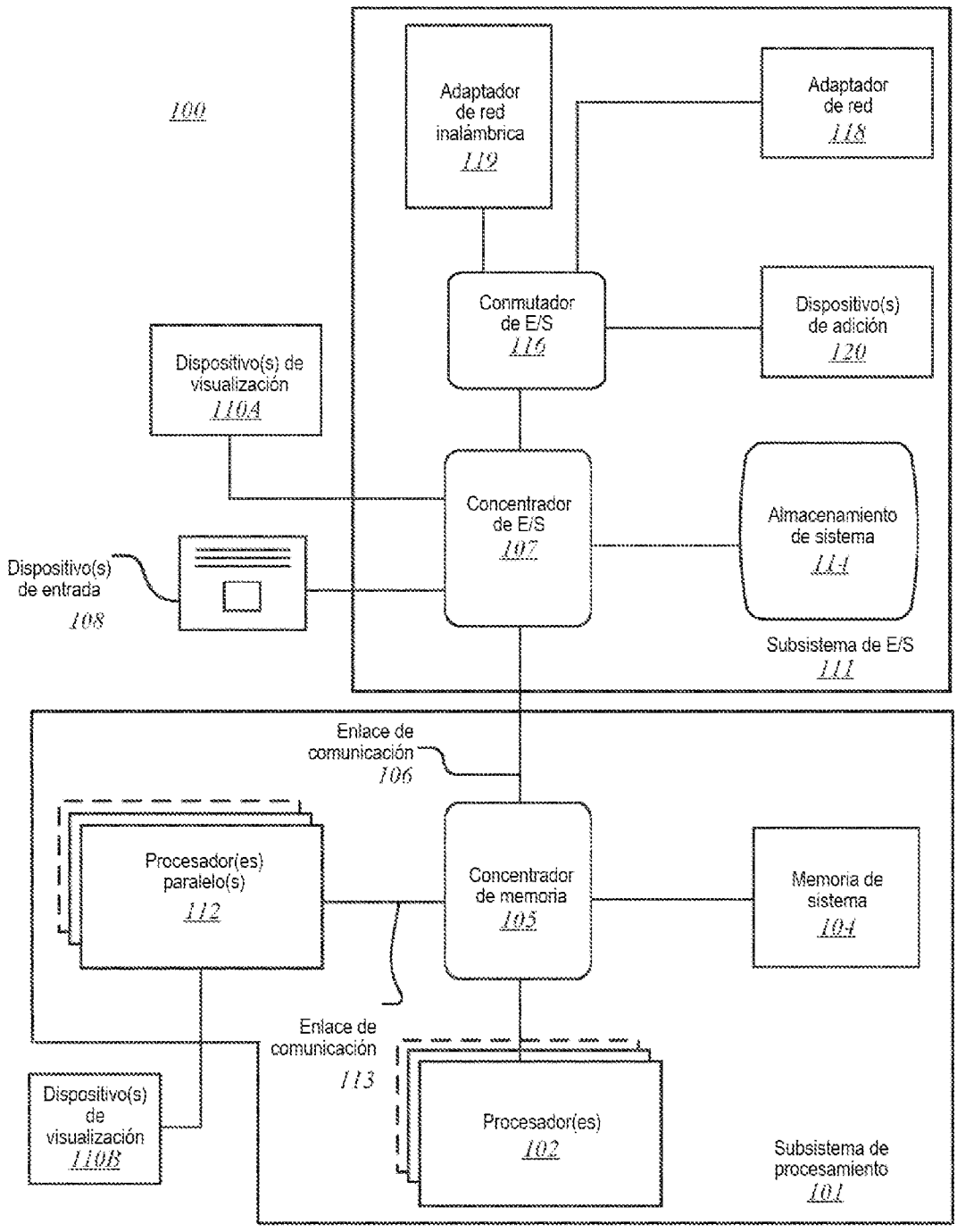

100
Adaptador de red inalámbrica 119
Adaptador de red 118
Conmutador de E/S 116
Dispositivo(s) de adición 120
Dispositivo(s) de visualización 110A
Concentrador de E/S 107
Almacenamiento de sistema 114
Dispositivo(s) de entrada 108
Subsistema de E/S 111
Enlace de comunicación 106
Procesador(es) paralelo(s)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

FIG. 1

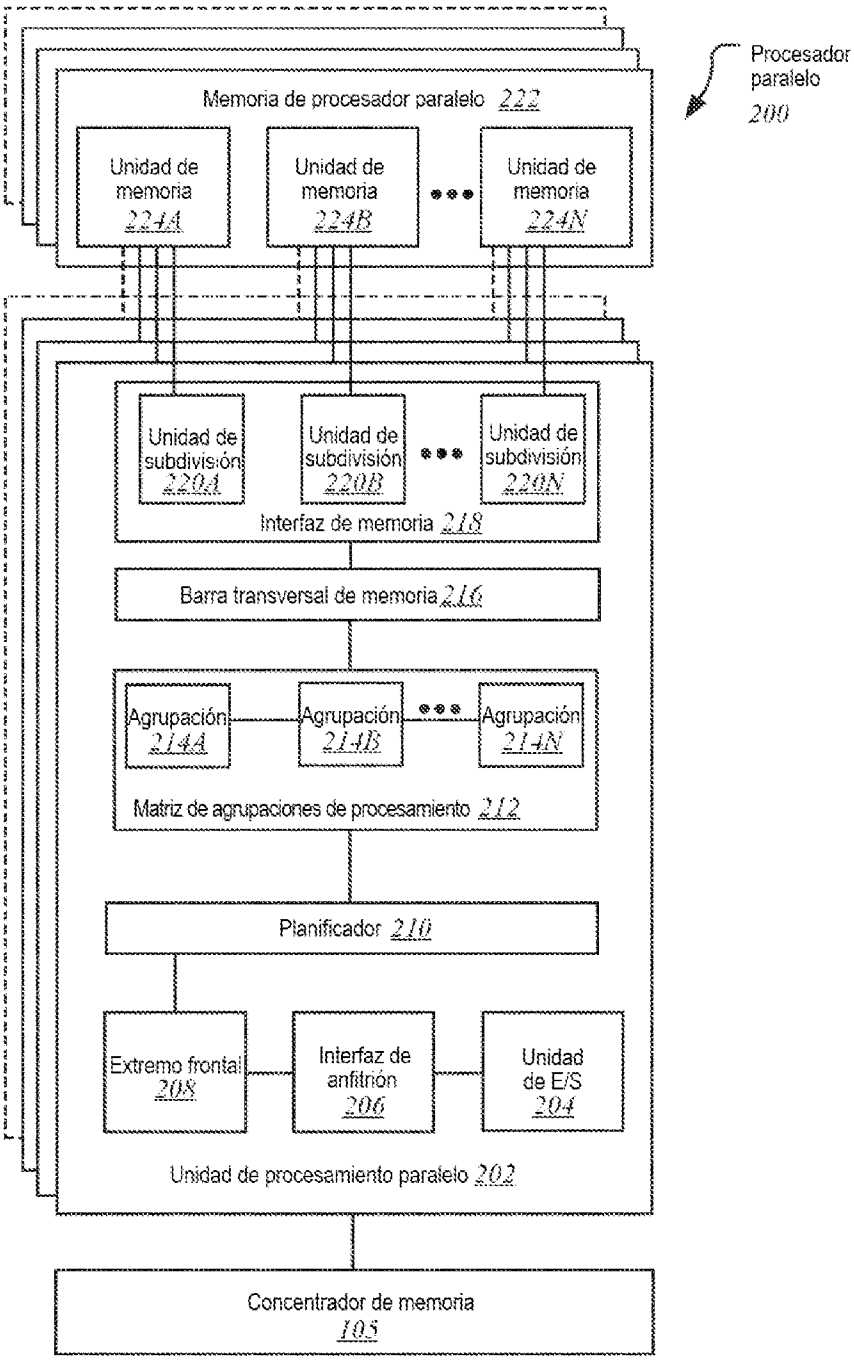

Procesador paralelo 200
Memoria de procesador paralelo 222
Unidad de memoria 224A
Unidad de memoria 224B
Unidad de memoria 224N
Unidad de subdivisión 220A
Unidad de subdivisión 220B
Unidad de subdivisión 220N
Interfaz de memoria 218
Barra transversal de memoria 216
Agrupación 214A
Agrupación 214B
Agrupación 214N
Matriz de agrupaciones 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

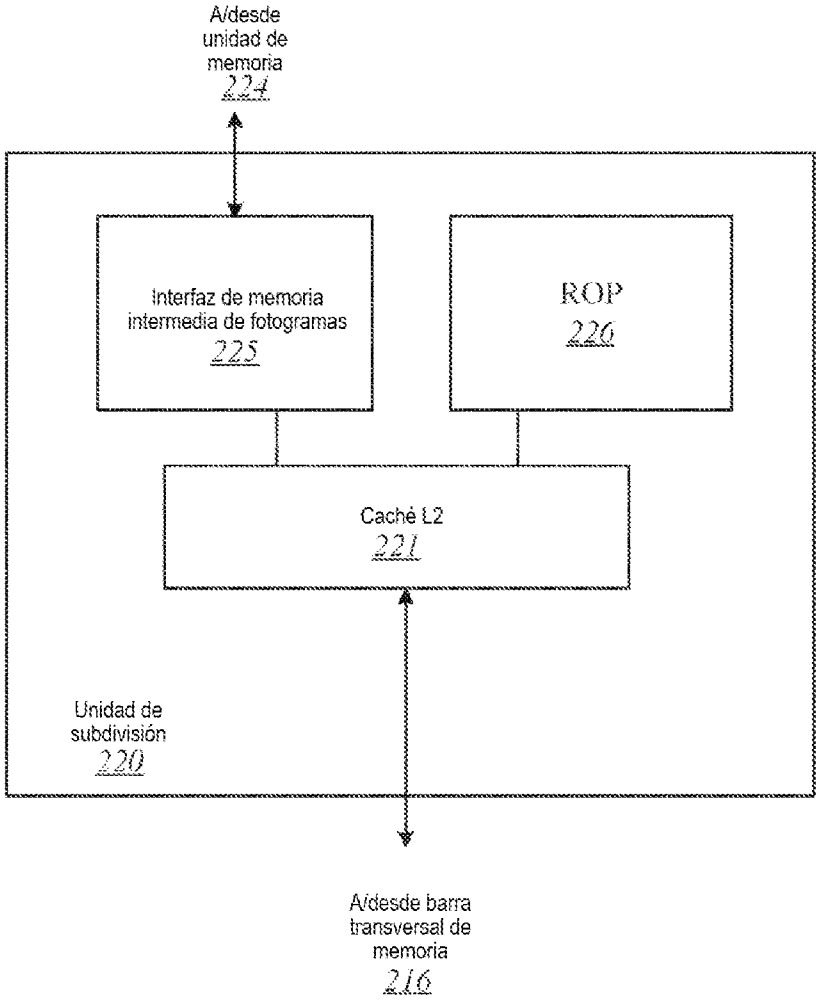

A/desde
unidad de
memoria
224
Interfaz de memoria
intermedia de fotogramas
225
ROP
226
Caché L2
221
Unidad de
subdivisión
220
A/desde barra
transversal de
memoria
216

FIG. 2B

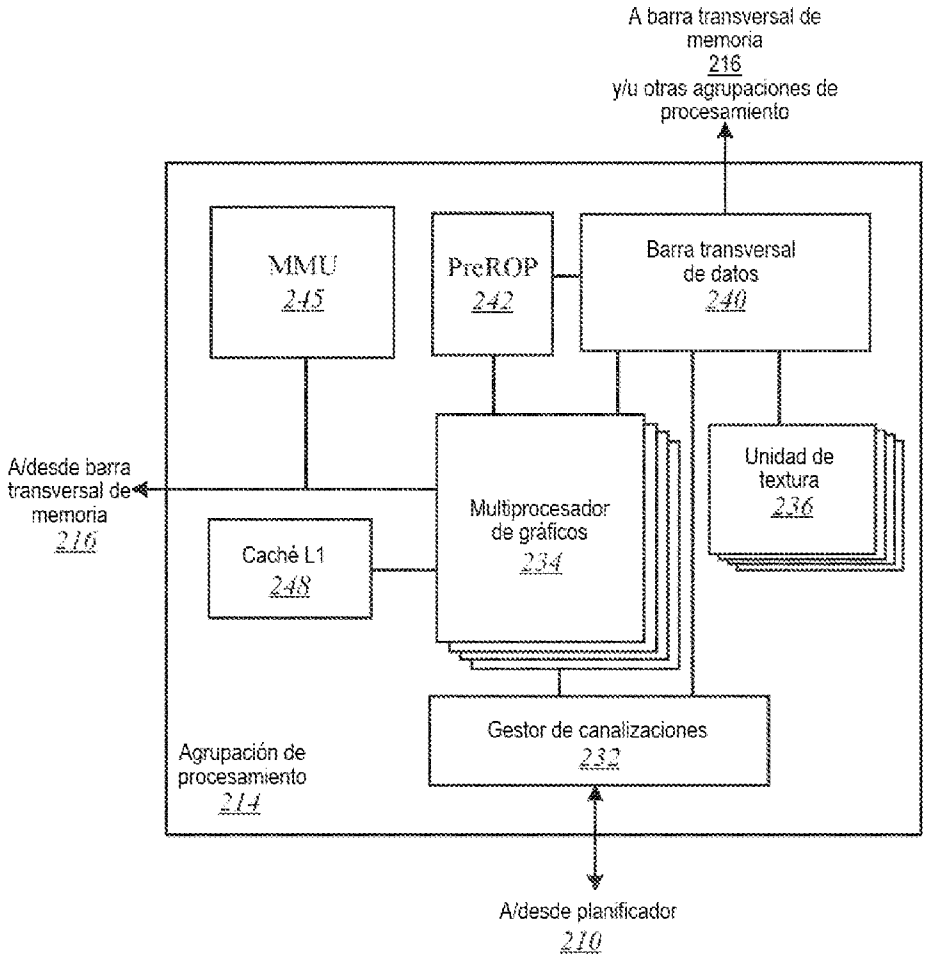

A barra transversal de memoria
216
y/u otras agrupaciones de procesamiento
MMU
245
PreROP
242
Barra transversal de datos
240
A/desde barra transversal de memoria
216
Multiprocesador de gráficos
234
Unidad de textura
236
Caché L1
248
Gestor de canalizaciones
232
Agrupación de procesamiento
214
A/desde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

FIG. 2C

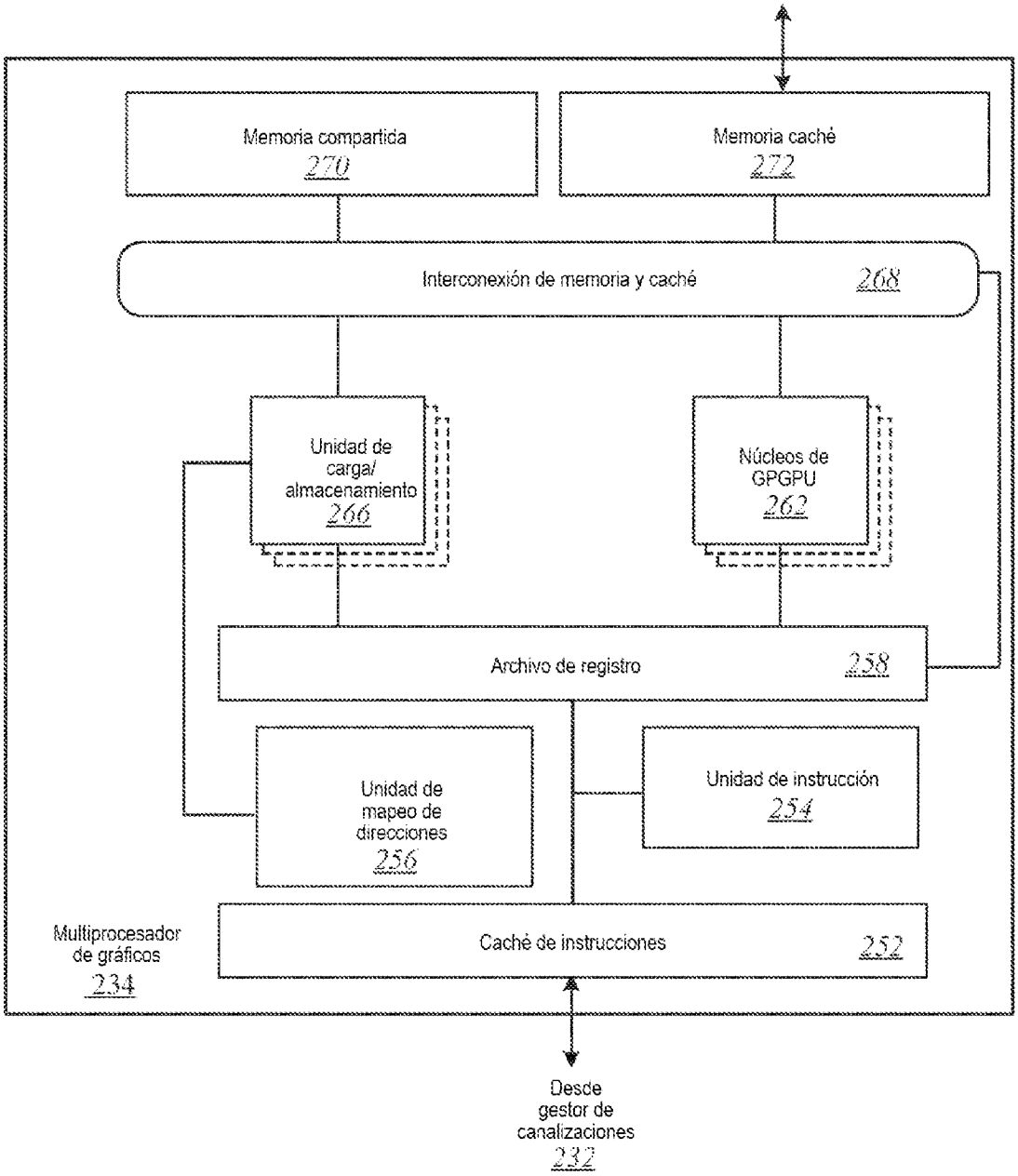

Memoria compartida
270
Memoria caché
272
Interconexión de memoria y caché
268
Unidad de carga/ almacenamiento
266
Núcleos de GPGPU
262
Archivo de registro
258
Unidad de mapeo de direcciones
256
Unidad de instrucción
254
Multiprocesador de gráficos
234
Caché de instrucciones
252
Desde gestor de canalizaciones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

FIG. 2D

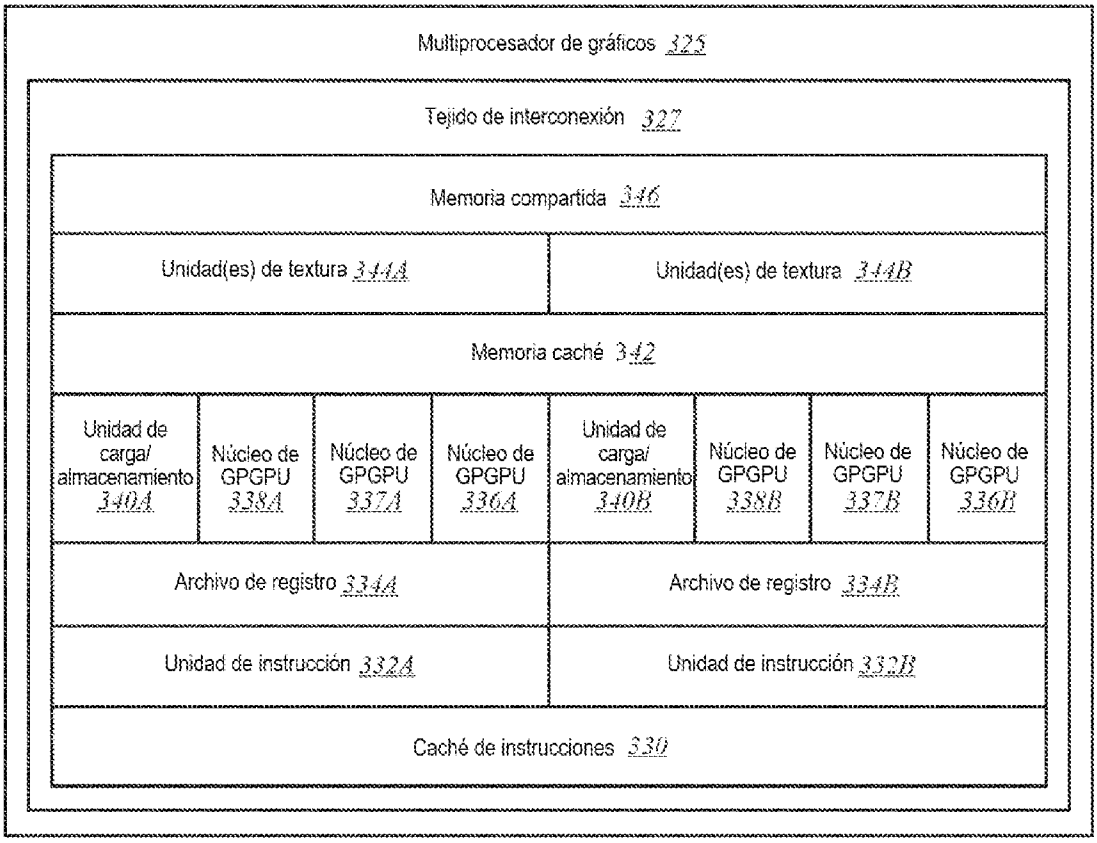

Multiprocesador de gráficos 325
Tejido de interconexión 327
Memoria compartida 346
Unidad(es) de textura 344A
Unidad(es) de textura 344B
Memoria caché 342
Unidad de carga/ almacenamiento 340A
Núcleo de GPGPU 338A
Núcleo de GPGPU 337A
Núcleo de GPGPU 336A
Unidad de carga/ almacenamiento 340B
Núcleo de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

FIG. 3A

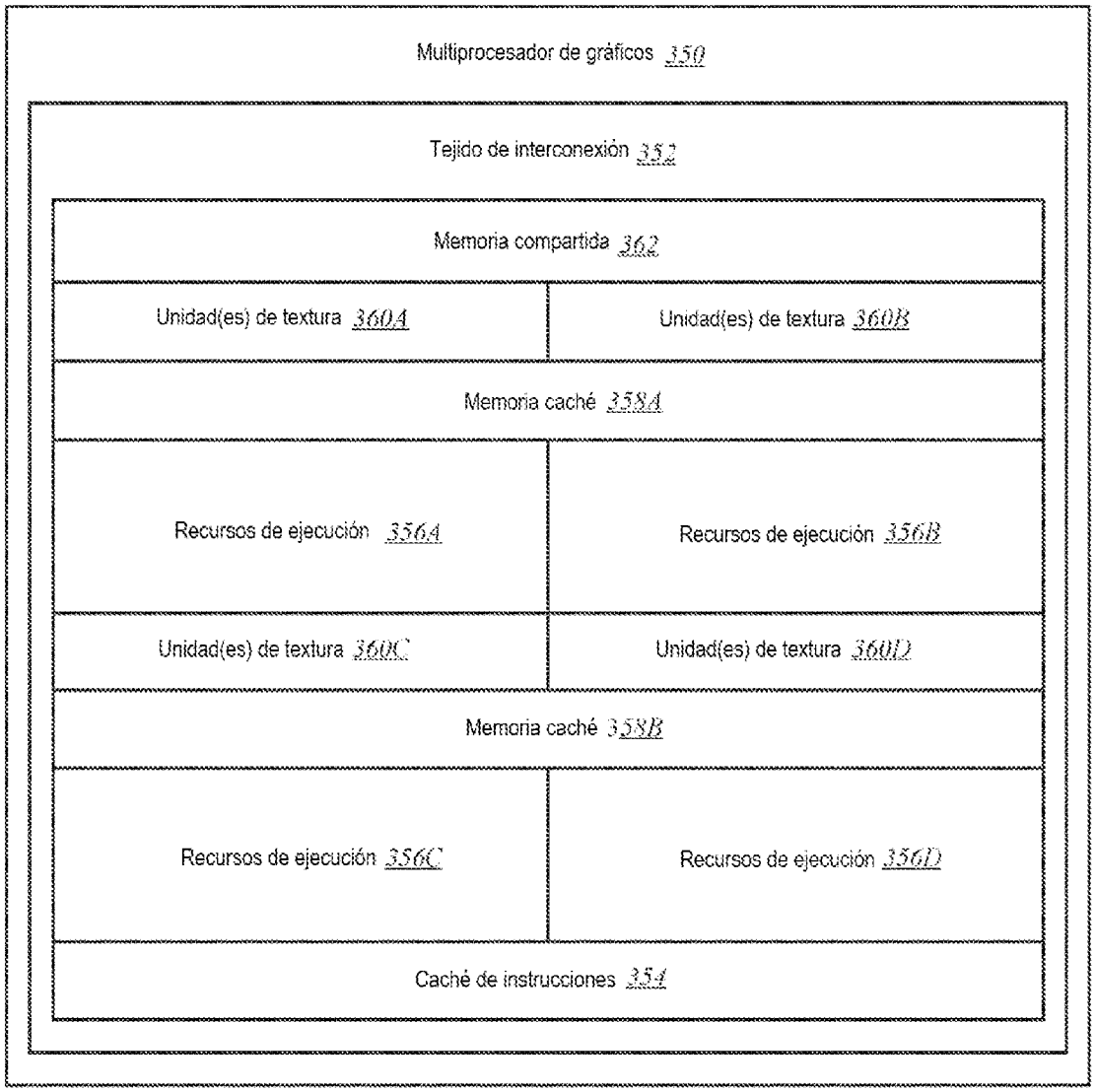

Multiprocesador de gráficos 350
Tejido de interconexión 352
Memoria compartida 362
Unidad(es) de textura 360A
Unidad(es) de textura 360B
Memoria caché 358A
Recursos de ejecución 356A
Recursos de ejecución 356B
Unidad(es) de textura 360C
Unidad(es) de textura 360D
Memoria caché 358B
Recursos de ejecución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

FIG. 3B

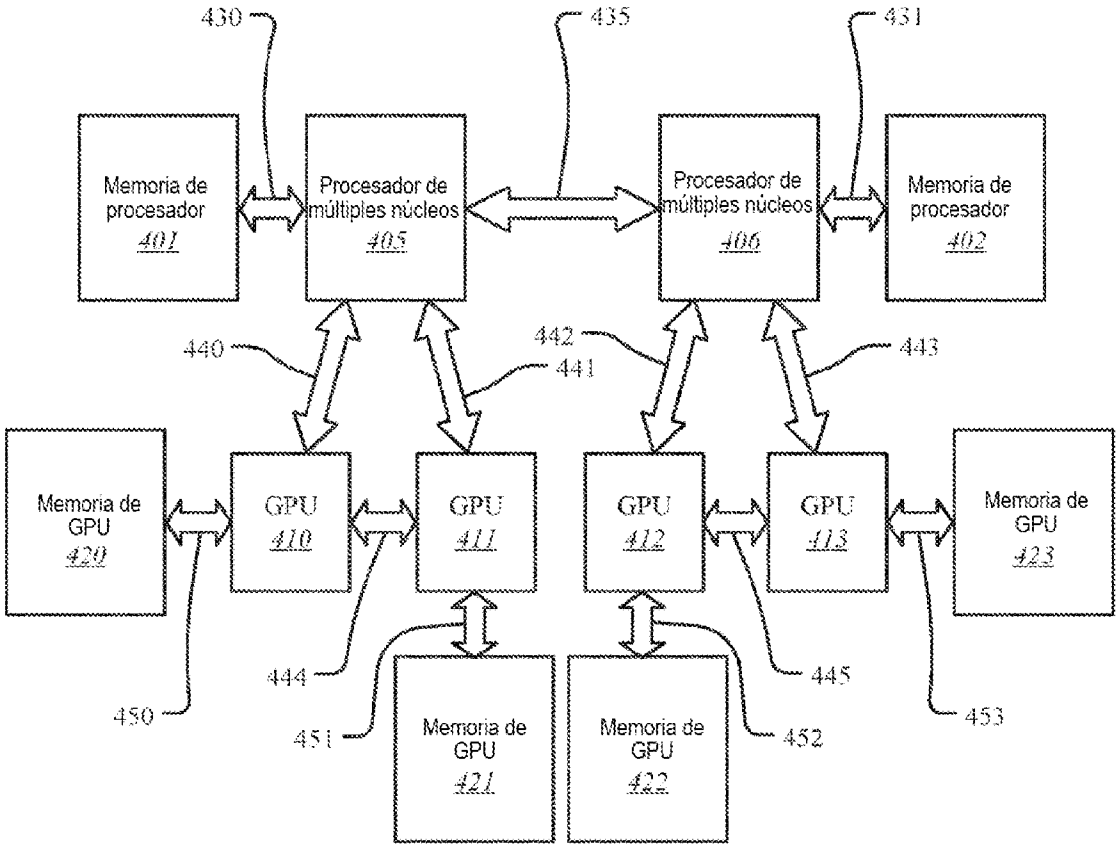

430
435
431
Memoria de procesador 401
Procesador de múltiples núcleos 405
Procesador de múltiples núcleos 406
Memoria de procesador 402
440
442
441
443
Memoria de GPU 420
GPU 410
GPU 411
GPU 412
GPU 413
Memoria de GPU 423
444
445
450
451
452
453
Memoria de GPU 421
Memoria de GPU 422

FIG. 4A

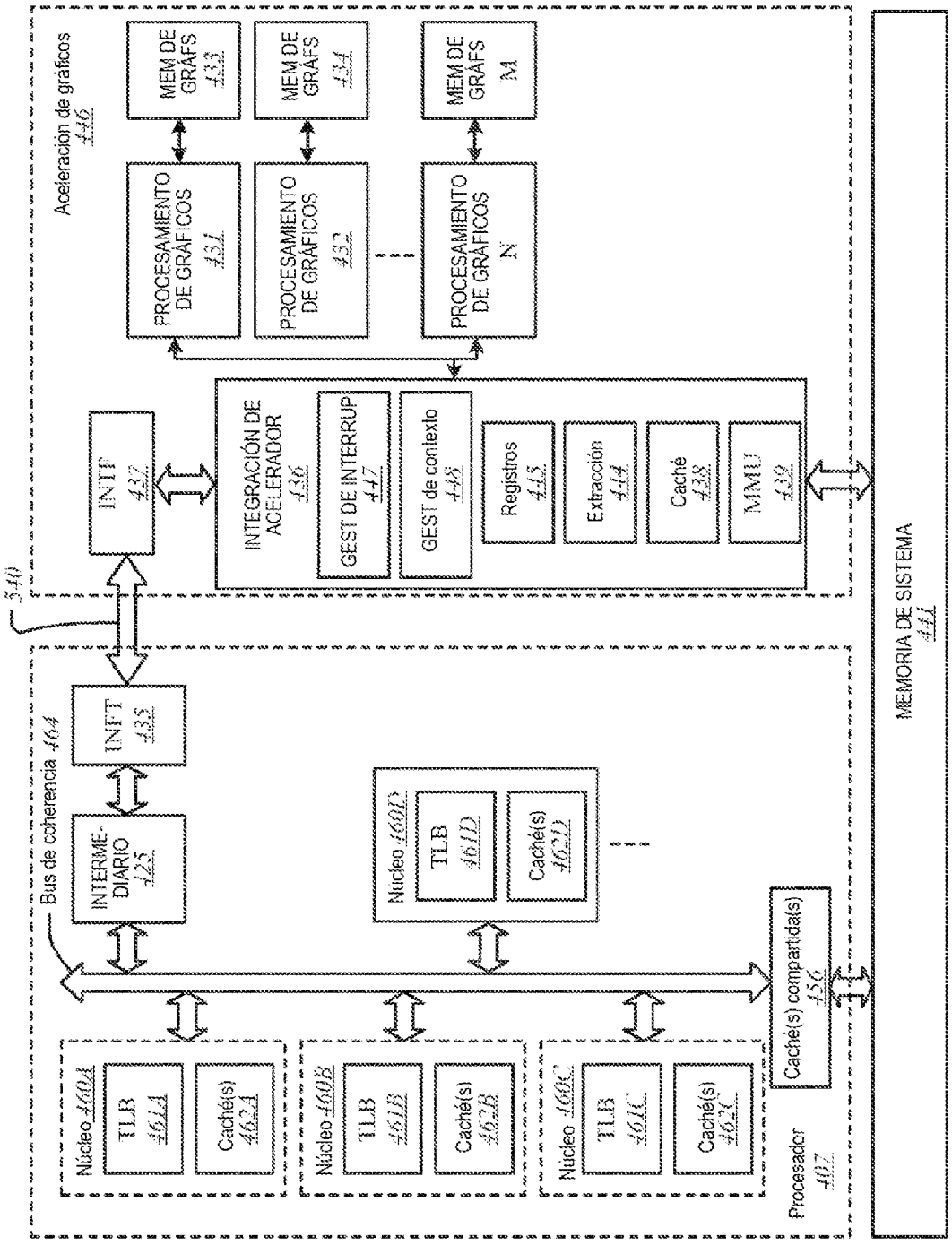

Aceleración de gráficos 446
MEM DE GRÁFS 433
MEM DE GRÁFS 434
MEM DE GRÁFS M
PROCESAMIENTO DE GRÁFICOS 431
PROCESAMIENTO DE GRÁFICOS 432
PROCESAMIENTO DE GRÁFICOS N
INTF 437
INTEGRACIÓN DE ACELERADOR 436
GEST DE INTERRUP 447
GEST de contexto 448
Registros 445
Extracción 444
Caché 438
MMU 439
340
INTF 435
INTERME-DIARIO 425
Bus de coherencia 464
Núcleo 460D
TLB 461D
Caché(s) 462D
Caché(s) compartida(s) 456
Procesador 407
Núcleo 460A
TLB 461A
Caché(s) 462A
Núcleo 460B
TLB 461B
Caché(s) 462B
Núcleo 460C
TLB 461C
Caché(s) 462C
MEMORIA DE SISTEMA 441

FIG. 4B

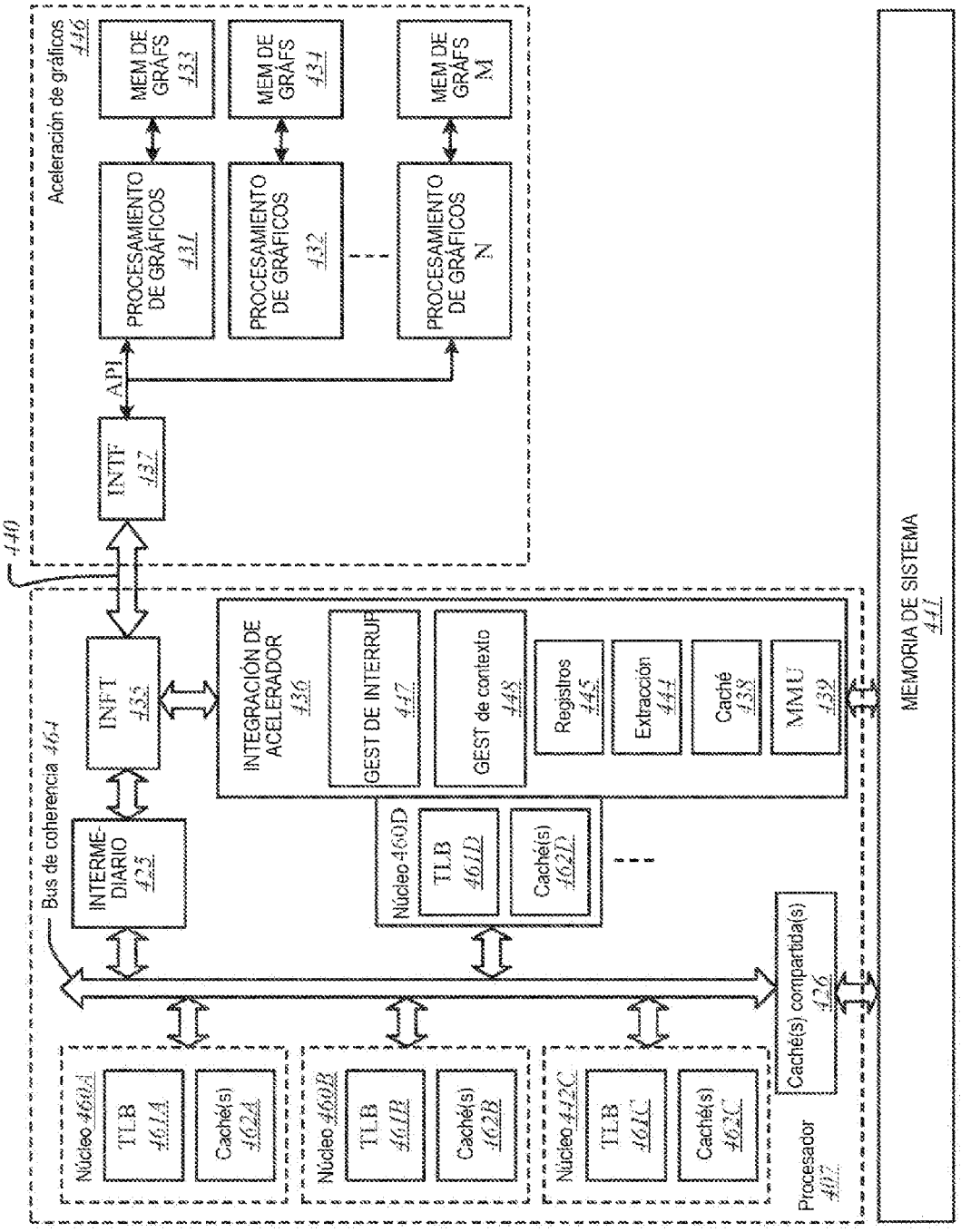

440
Aceleración de gráficos 446
MEM DE GRÁFS 433
MEM DE GRÁFS 434
MEM DE GRÁFS M
PROCESAMIENTO DE GRÁFICOS 431
PROCESAMIENTO DE GRÁFICOS 432
PROCESAMIENTO DE GRÁFICOS N
API
INTF 437
Bus de coherencia 464
INTERME-DIARIO 425
INFT 435
INTEGRACIÓN DE ACELERADOR 436
GEST DE INTERRUP 447
GEST de contexto 448
Registros 445
Extracción 444
Caché 438
MMU 432
Núcleo 460D
TLB 461D
Caché(s) 462D
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
MEMORIA DE SISTEMA 441

FIG. 4C

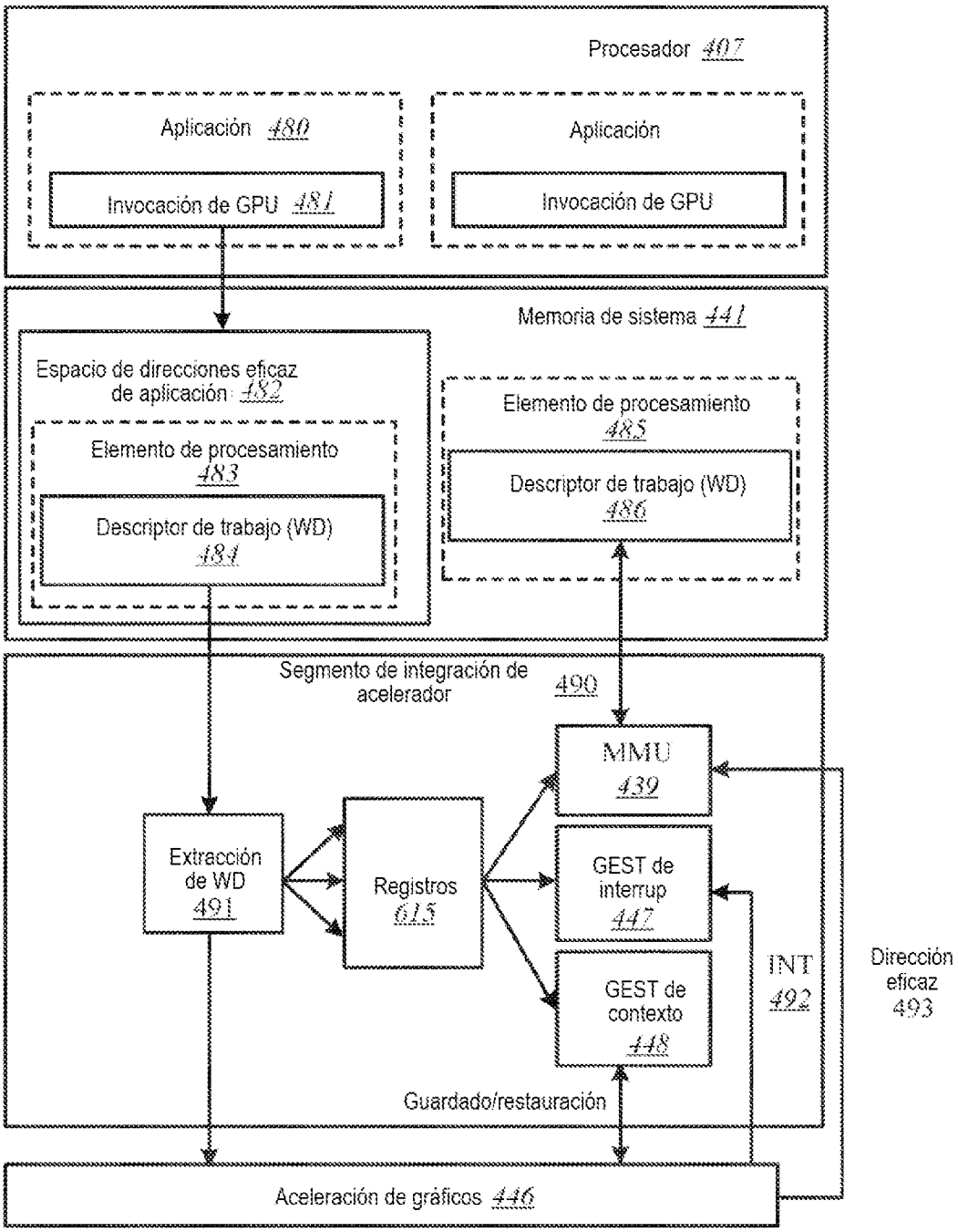

Procesador 407
Aplicación 480
Invocación de GPU 481
Aplicación
Invocación de GPU
Memoria de sistema 441
Espacio de direcciones eficaz de aplicación 482
Elemento de procesamiento 483
Descriptor de trabajo (WD) 484
Elemento de procesamiento 485
Descriptor de trabajo (WD) 486
Segmento de integración de acelerador 490
MMU 439
Extracción de WD 491
Registros 615
GEST de interrup 447
GEST de contexto 448
INT 492
Dirección eficaz 493
Guardado/restauración
Aceleración de gráficos 446

FIG. 4D

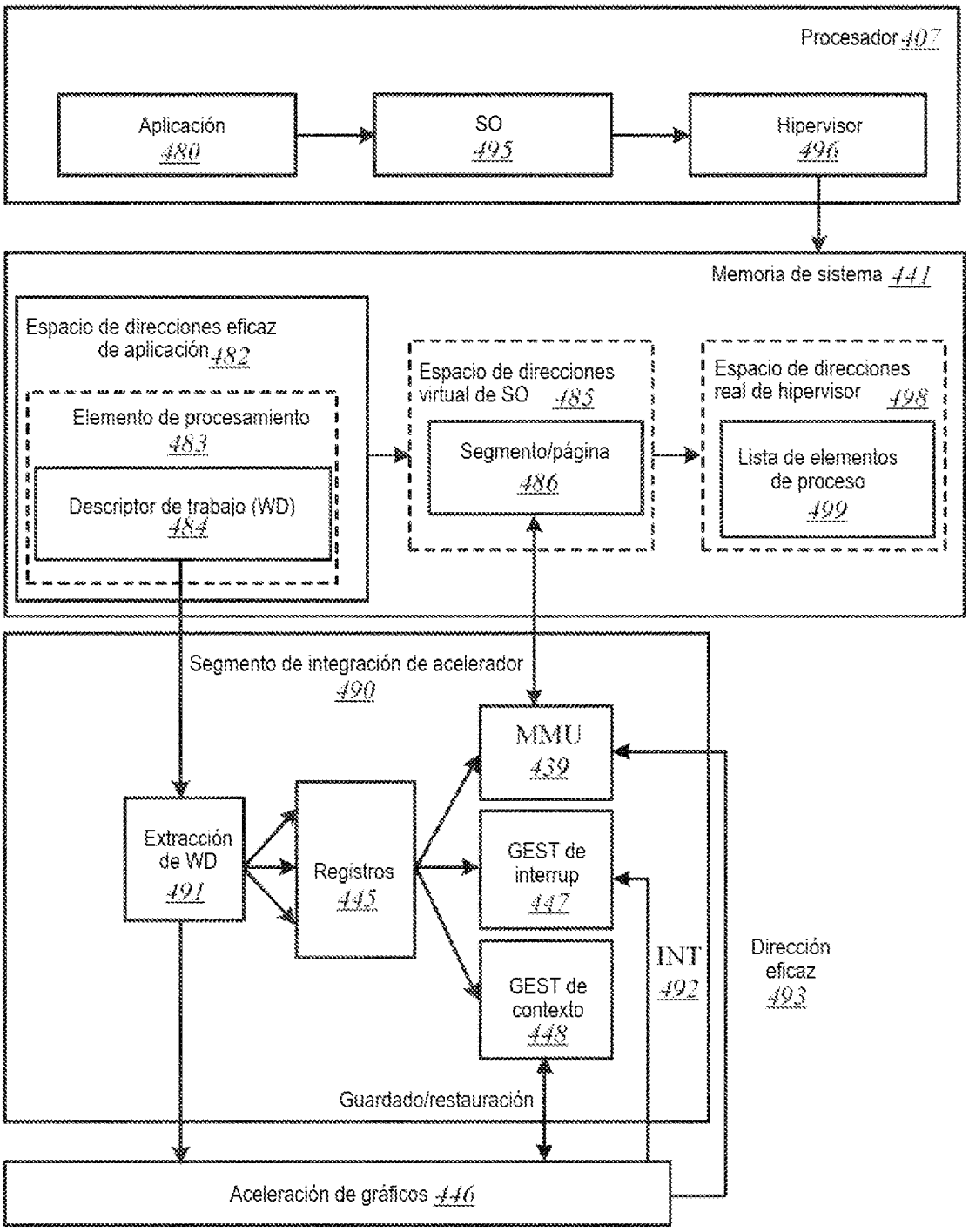

Procesador 407
Aplicación 480
SO 495
Hipervisor 496
Memoria de sistema 441
Espacio de direcciones eficaz de aplicación 482
Elemento de procesamiento 483
Descriptor de trabajo (WD) 484
Espacio de direcciones virtual de SO 485
Segmento/página 486
Espacio de direcciones real de hipervisor 498
Lista de elementos de proceso 499
Segmento de integración de acelerador 490
MMU 439
Extracción de WD 491
Registros 445
GEST de interrup 447
GEST de contexto 448
INT 492
Dirección eficaz 493
Guardado/restauración
Aceleración de gráficos 446

FIG. 4E

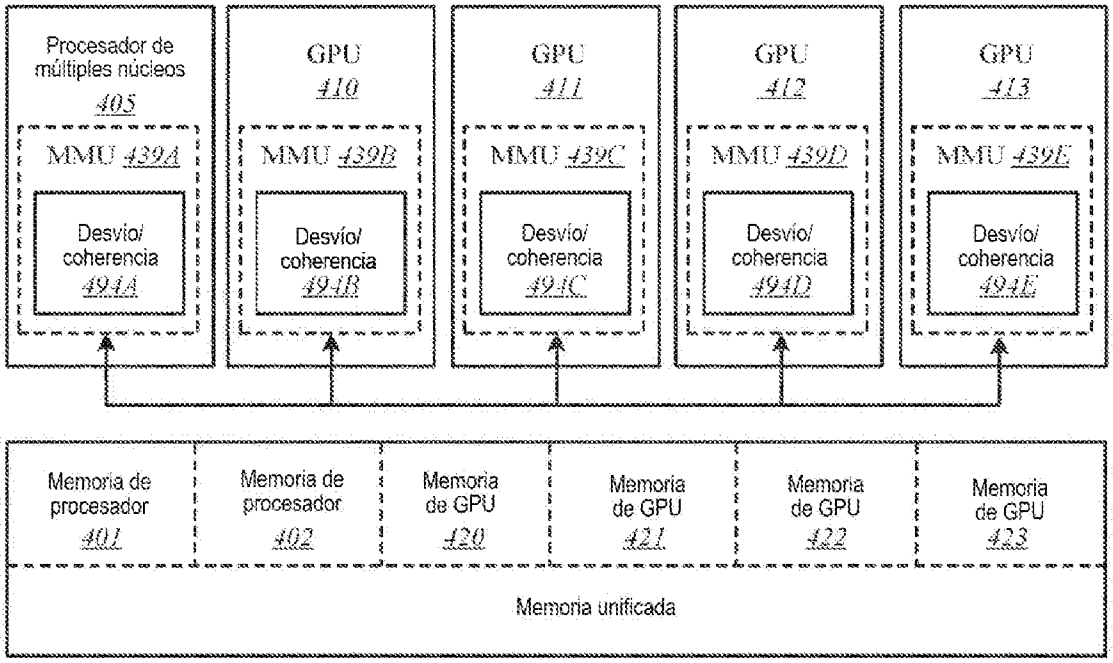

Procesador de múltiples núcleos
405
MMU 439A
Desvío/ coherencia
494A
GPU
410
MMU 439B
Desvío/ coherencia
494B
GPU
411
MMU 439C
Desvío/ coherencia
494C
GPU
412
MMU 439D
Desvío/ coherencia
494D
GPU
413
MMU 439E
Desvío/ coherencia
494E
Memoria de procesador
401
Memoria de procesador
402
Memoria de GPU
420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

FIG. 4F

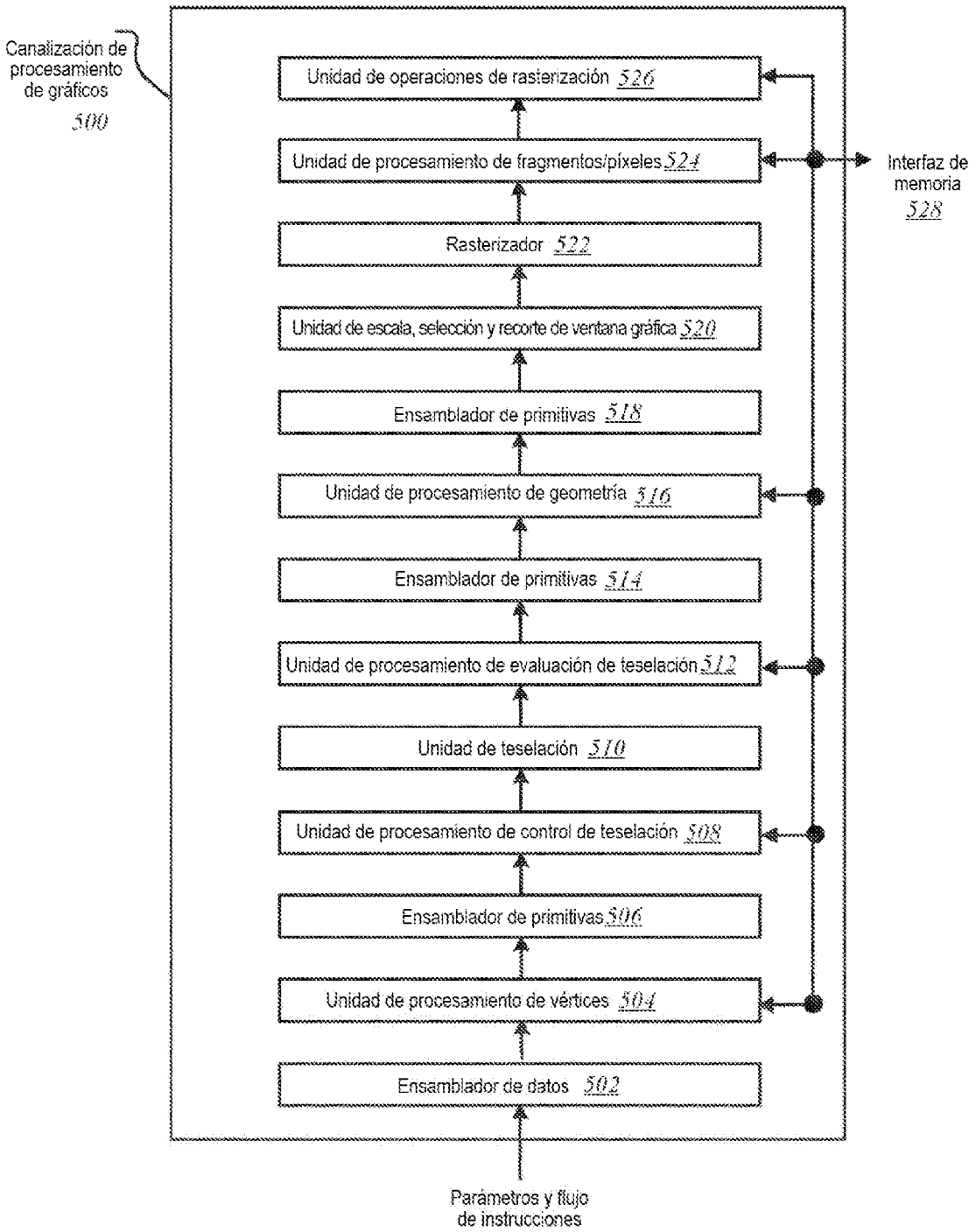

Canalización de procesamiento de gráficos 500
Unidad de operaciones de rasterización 526
Unidad de procesamiento de fragmentos/píxeles 524
Rasterizador 522
Unidad de escala, selección y recorte de ventana gráfica 520
Ensamblador de primitivas 518
Unidad de procesamiento de geometría 516
Ensamblador de 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 vértices 504
Ensamblador de datos 502
Interfaz de memoria 528
Parámetros y flujo de instrucciones

FIG. 5

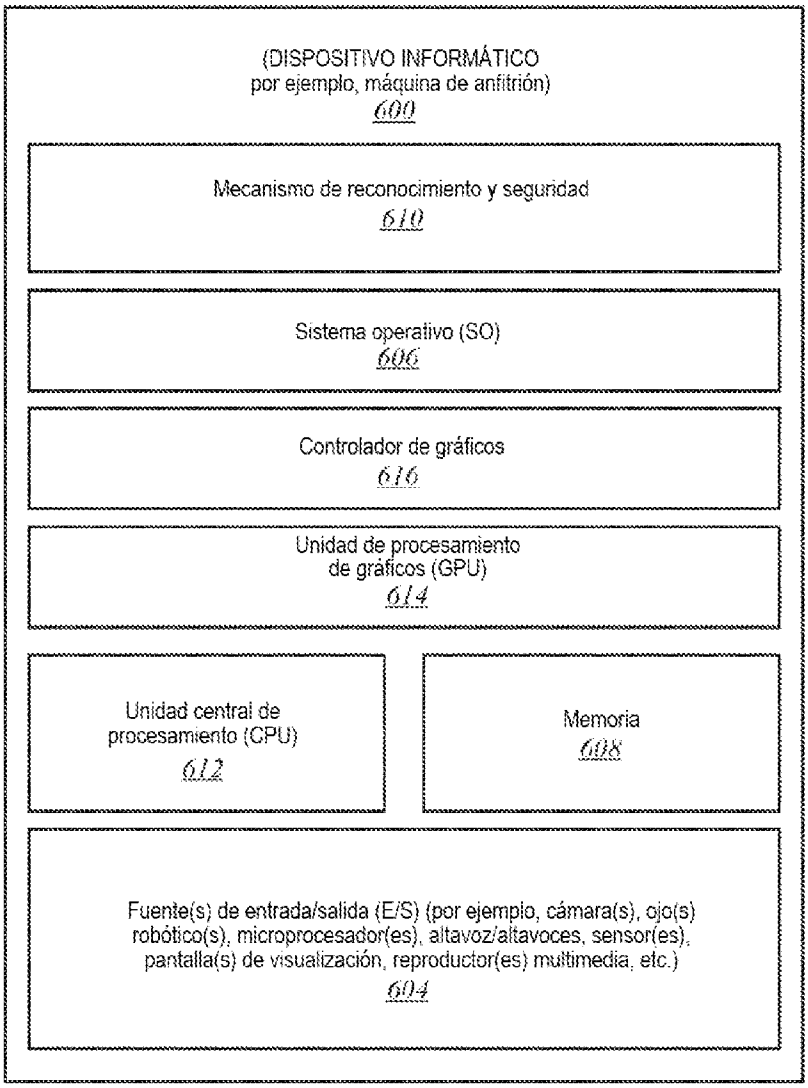

(DISPOSITIVO INFORMÁTICO
por ejemplo, máquina de anfitrión)
600
Mecanismo de reconocimiento y seguridad
610
Sistema operativo (SO)
606
Controlador de gráficos
616
Unidad de procesamiento
de gráficos (GPU)
614
Unidad central de
procesamiento (CPU)
612
Memoria
608
Fuente(s) de entrada/salida (E/S) (por ejemplo, cámara(s), ojo(s)
robótico(s), microprocesador(es), altavoz/altavoces, sensor(es),
pantalla(s) de visualización, reproductor(es) multimedia, etc.)
604

FIG. 6

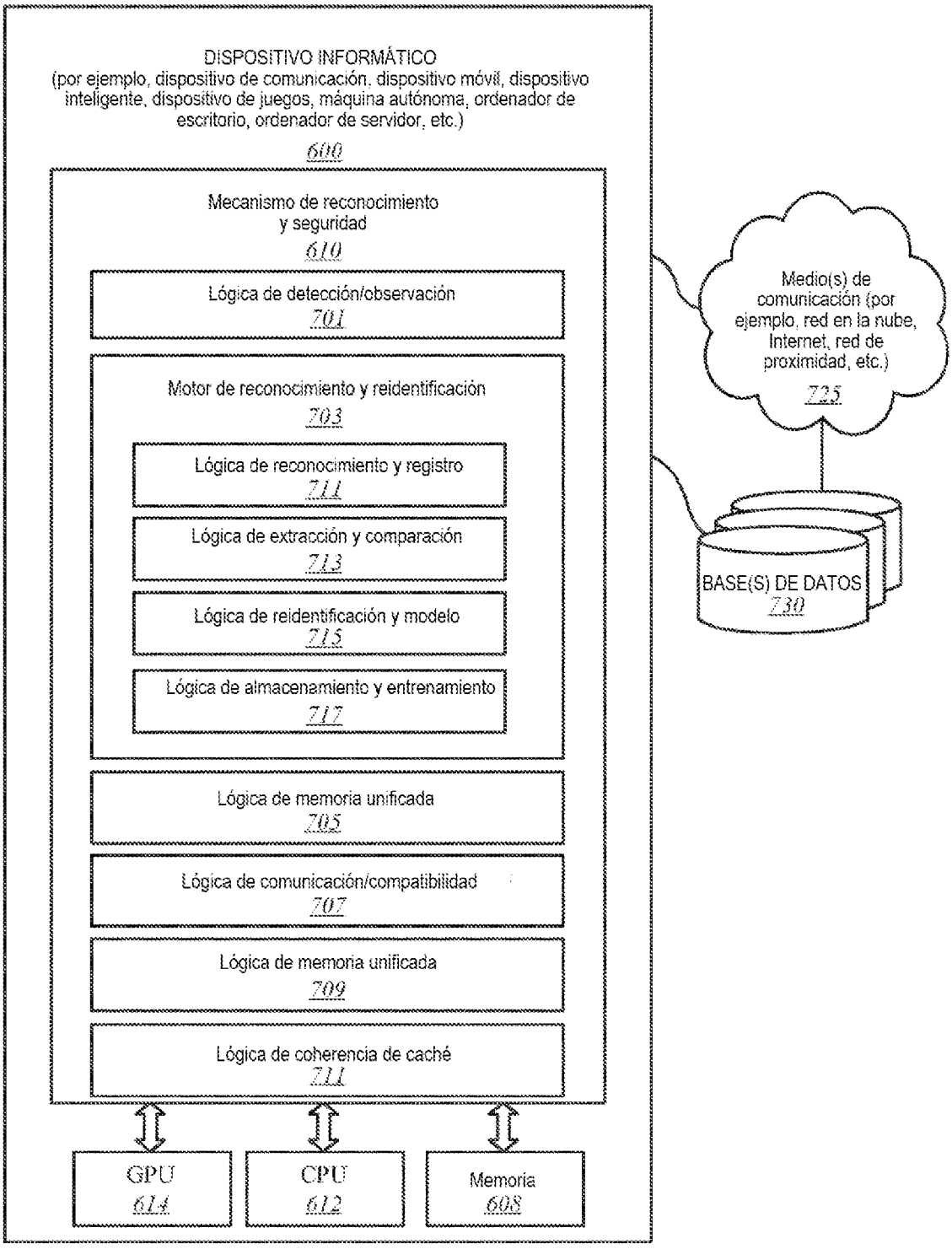

DISPOSITIVO INFORMÁTICO
(por ejemplo, dispositivo de comunicación, dispositivo móvil, dispositivo inteligente, dispositivo de juegos, máquina autónoma, ordenador de escritorio, ordenador de servidor, etc.)
600
Mecanismo de reconocimiento y seguridad
610
Lógica de detección/observación
701
Motor de reconocimiento y reidentificación
703
Lógica de reconocimiento y registro
711
Lógica de extracción y comparación
713
Lógica de reidentificación y modelo
715
Lógica de almacenamiento y entrenamiento
717
Lógica de memoria unificada
705
Lógica de comunicación/compatibilidad
707
Lógica de memoria unificada
709
Lógica de coherencia de caché
711
GPU
614
CPU
612
Memoria
608
Medio(s) de comunicación (por ejemplo, red en la nube, Internet, red de proximidad, etc.)
725
BASE(S) DE DATOS
730

FIG. 7

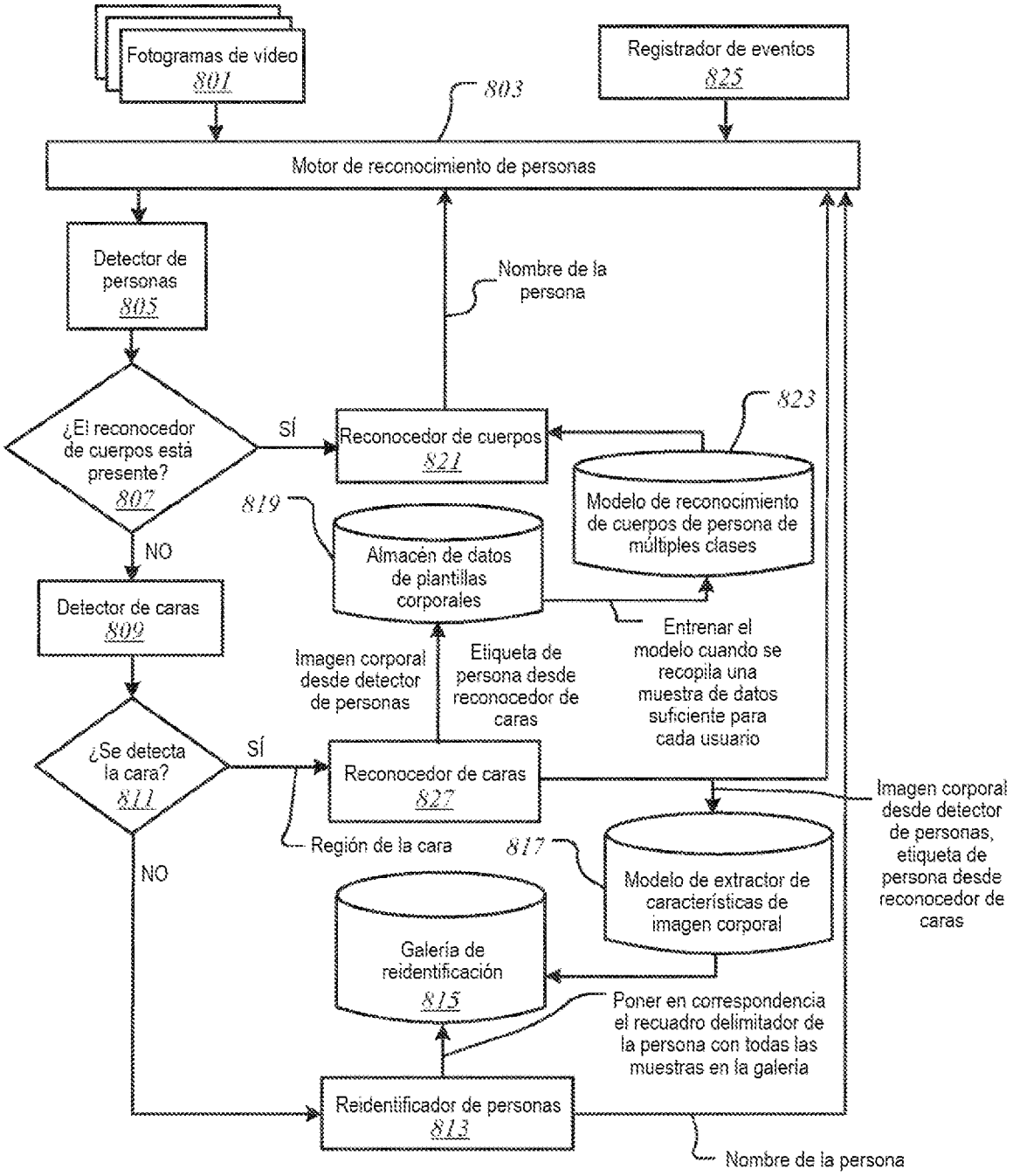

Fotogramas de vídeo
801
803
Registrador de eventos
825
Motor de reconocimiento de personas
Detector de personas
805
Nombre de la persona
¿El reconocedor de cuerpos está presente?
807
SÍ
Reconocedor de cuerpos
821
823
Modelo de reconocimiento de cuerpos de persona de múltiples clases
819
Almacén de datos de plantillas corporales
NO
Detector de caras
809
Entrenar el modelo cuando se recopila una muestra de datos suficiente para cada usuario
Imagen corporal desde detector de personas
Etiqueta de persona desde reconocedor de caras
¿Se detecta la cara?
811
SÍ
Reconocedor de caras
827
Imagen corporal desde detector de personas, etiqueta de persona desde reconocedor de caras
Región de la cara
817
Modelo de extractor de características de imagen corporal
NO
Galería de reidentificación
815
Poner en correspondencia el recuadro delimitador de la persona con todas las muestras en la galería
Reidentificador de personas
813
Nombre de la persona

FIG. 8A

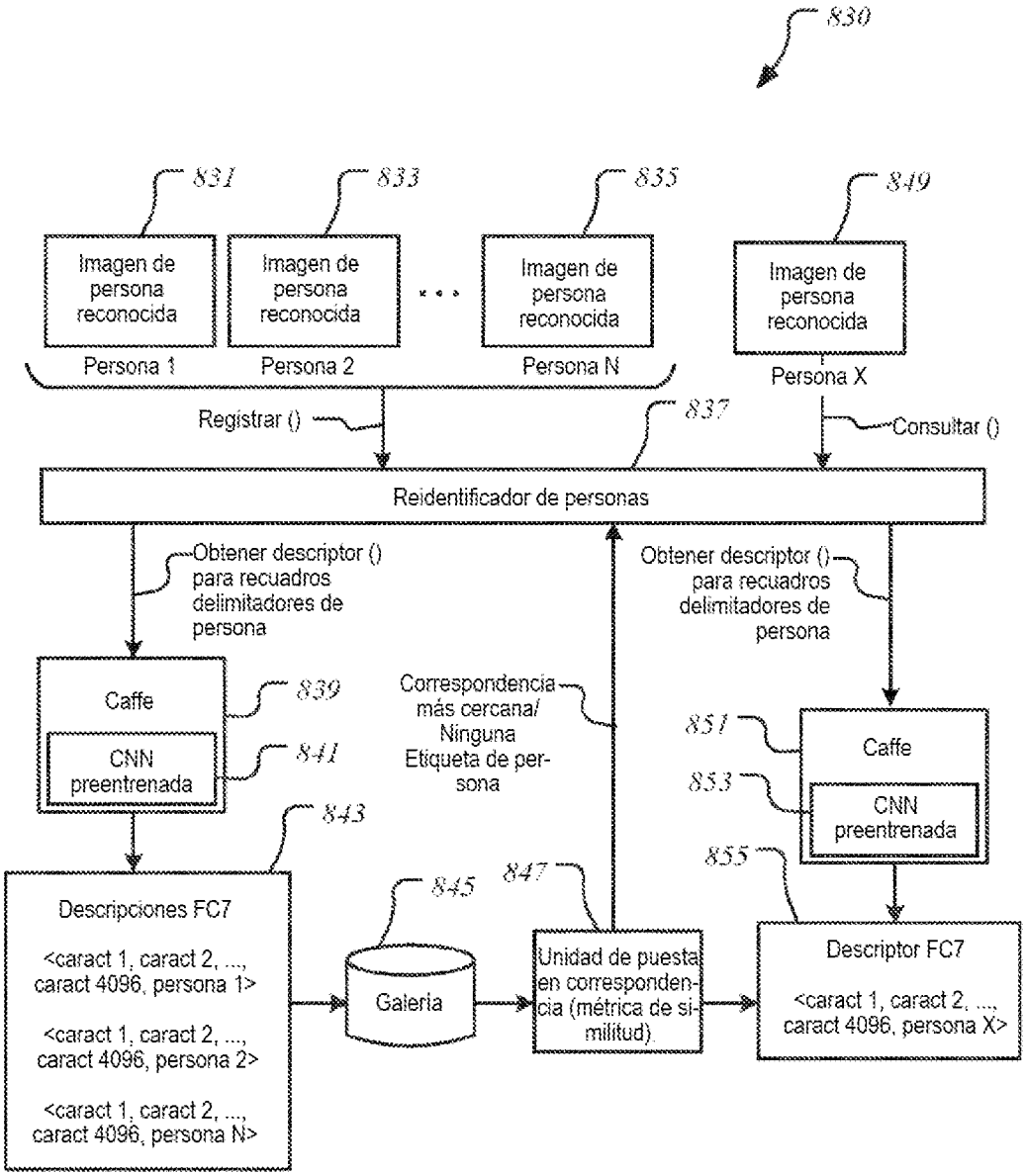

830
831
833
835
849
Imagen de persona reconocida
Imagen de persona reconocida
Imagen de persona reconocida
Imagen de persona reconocida
Persona 1
Persona 2
Persona N
Persona X
Registrar ()
837
Consultar ()
Reidentificador de personas
Obtener descriptor () para recuadros delimitadores de persona
Obtener descriptor () para recuadros delimitadores de persona
Caffe
CNN preentrenada
839
841
Correspondencia más cercana/ Ninguna Etiqueta de persona
851
Caffe
853
CNN preentrenada
855
843
845
847
Descripciones FC7
<caract 1, caract 2, ..., caract 4096, persona 1>
<caract 1, caract 2, ..., caract 4096, persona 2>
<caract 1, caract 2, ..., caract 4096, persona N>
Galeria
Unidad de puesta en correspondencia (métrica de similitud)
Descriptor FC7
<caract 1, caract 2, ..., caract 4096, persona X>

FIG. 8B

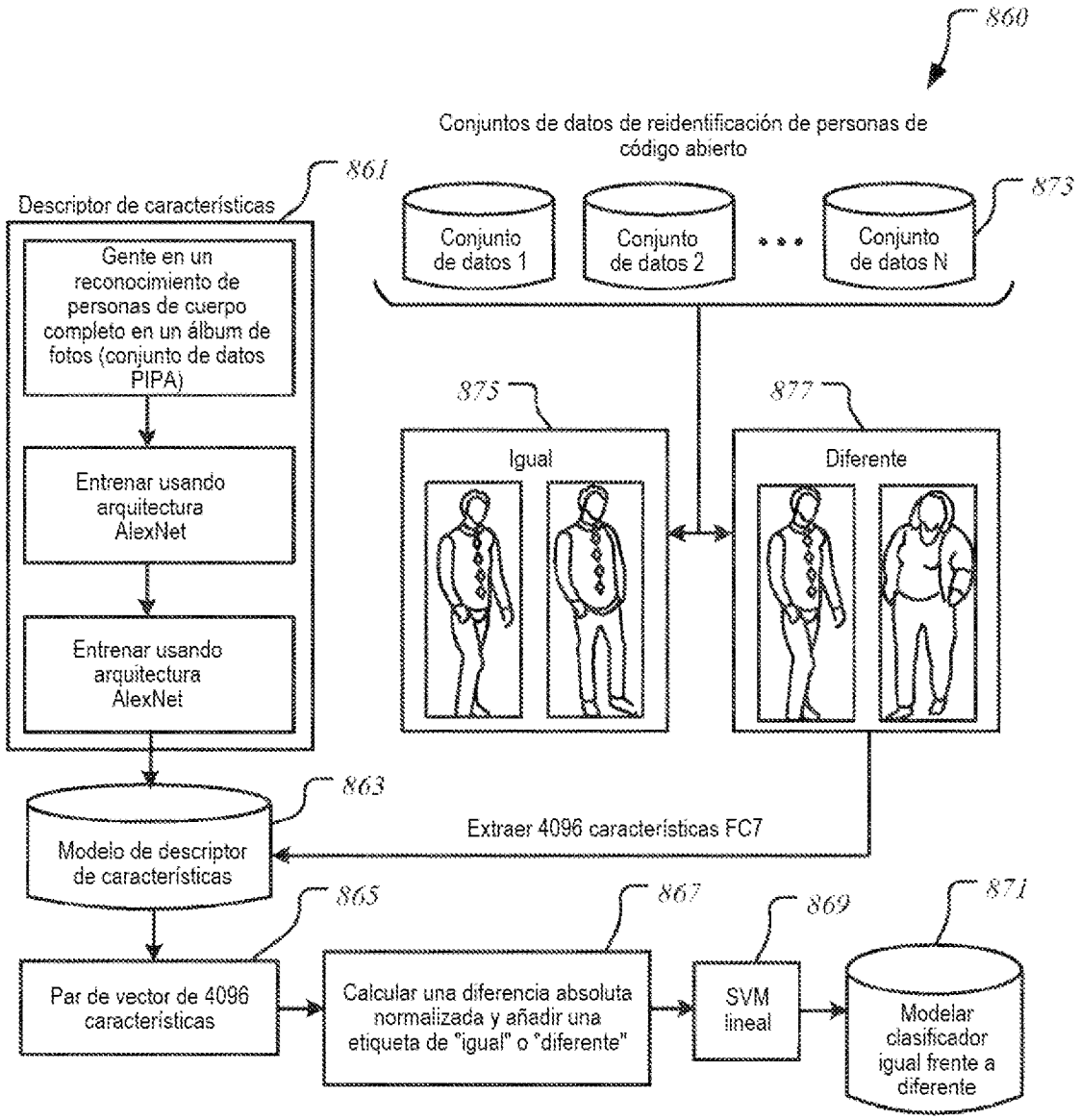

860
Conjuntos de datos de reidentificación de personas de código abierto
861
Descriptor de características
Gente en un reconocimiento de personas de cuerpo completo en un álbum de fotos (conjunto de datos PIPA)
Entrenar usando arquitectura AlexNet
Entrenar usando arquitectura AlexNet
Conjunto de datos 1
Conjunto de datos 2
Conjunto de datos N
873
875
Igual
877
Diferente
863
Modelo de descriptor de características
Extraer 4096 características FC7
865
Par de vector de 4096 características
867
Calcular una diferencia absoluta normalizada y añadir una etiqueta de "igual" o "diferente"
869
SVM lineal
871
Modelar clasificador igual frente a diferente

FIG. 8C

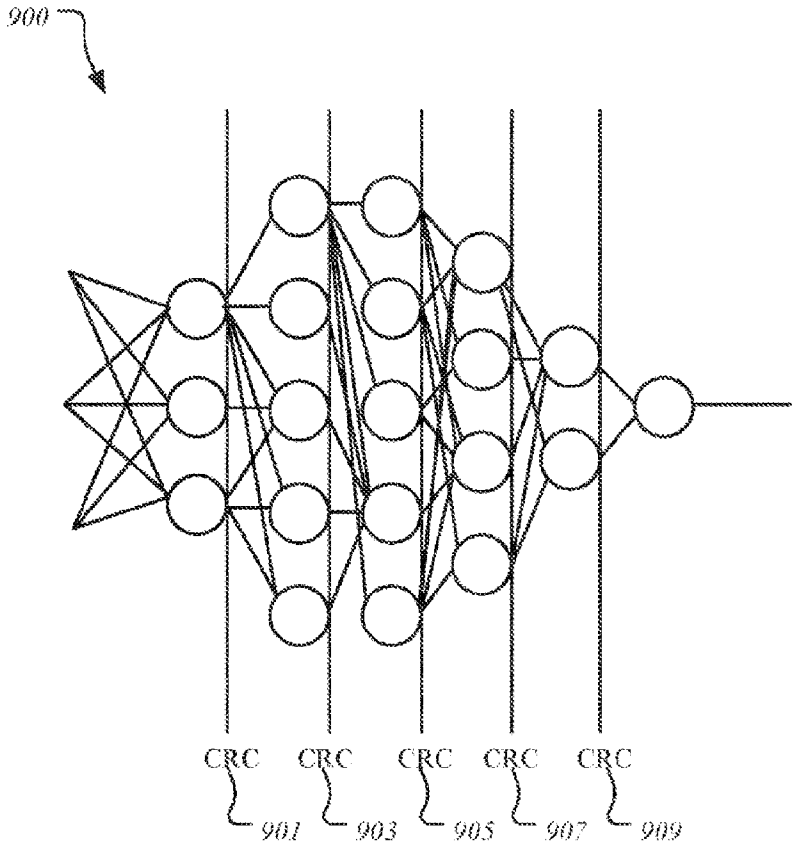


FIG. 9A

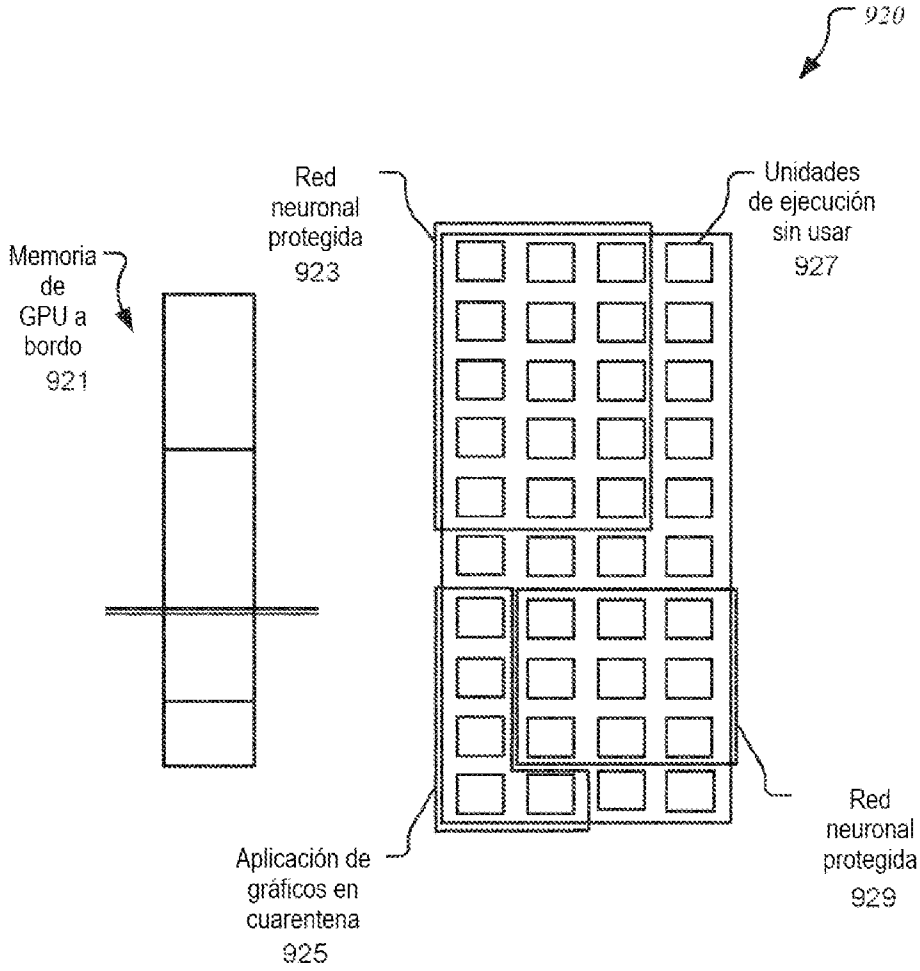

920
Red neuronal protegida 923
Unidades de ejecución sin usar 927
Memoria de GPU a bordo 921
Aplicación de gráficos en cuarentena 925
Red neuronal protegida 929

FIG. 9B

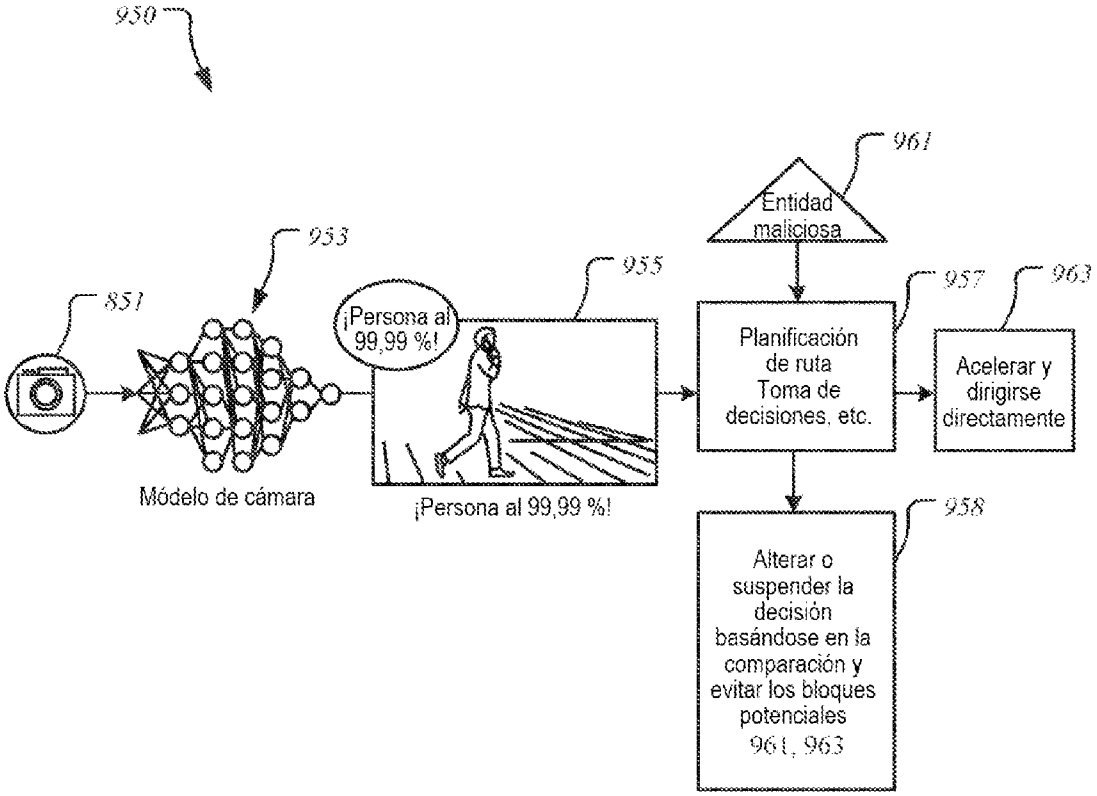

950
961
Entidad maliciosa
953
955
957
963
851
¡Persona al 99,99 %!
Planificación de ruta Toma de decisiones, etc.
Acelerar y dirigirse directamente
Módelo de cámara
¡Persona al 99,99 %!
958
Alterar o suspender la decisión basándose en la comparación y evitar los bloques potenciales 961, 963

FIG. 9C

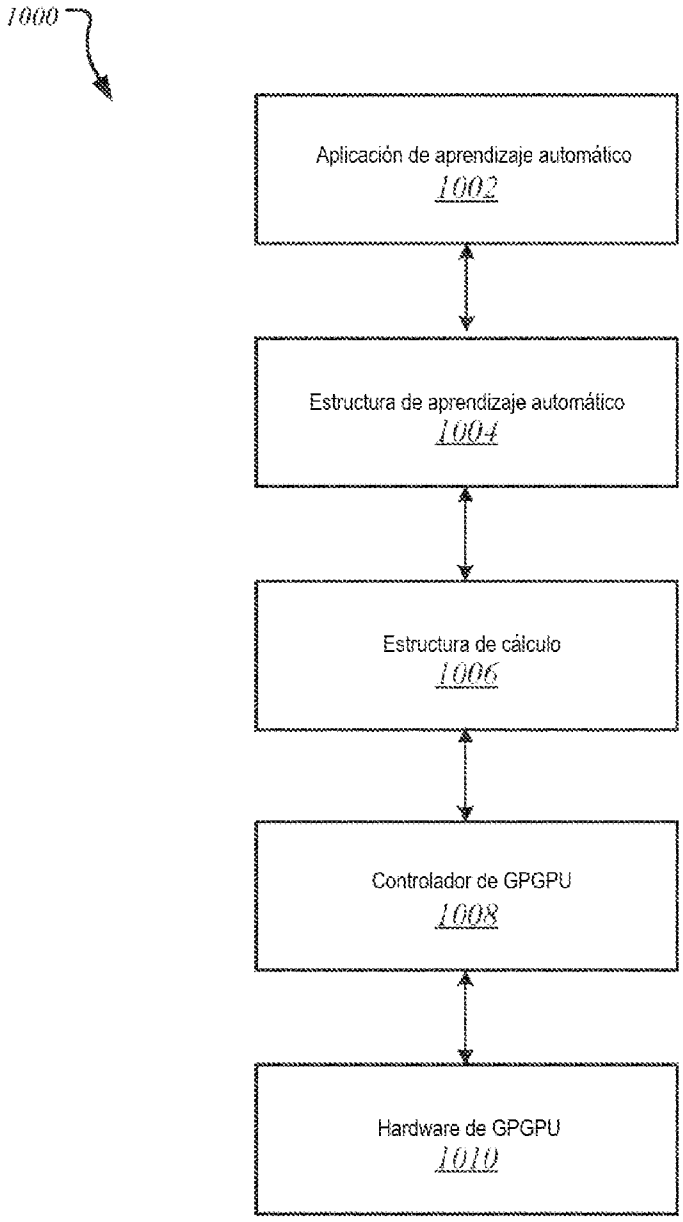

1000
Aplicación de aprendizaje automático
1002
Estructura de aprendizaje automático
1004
Estructura de cálculo
1006
Controlador de GPGPU
1008
Hardware de GPGPU
1010

FIG. 10

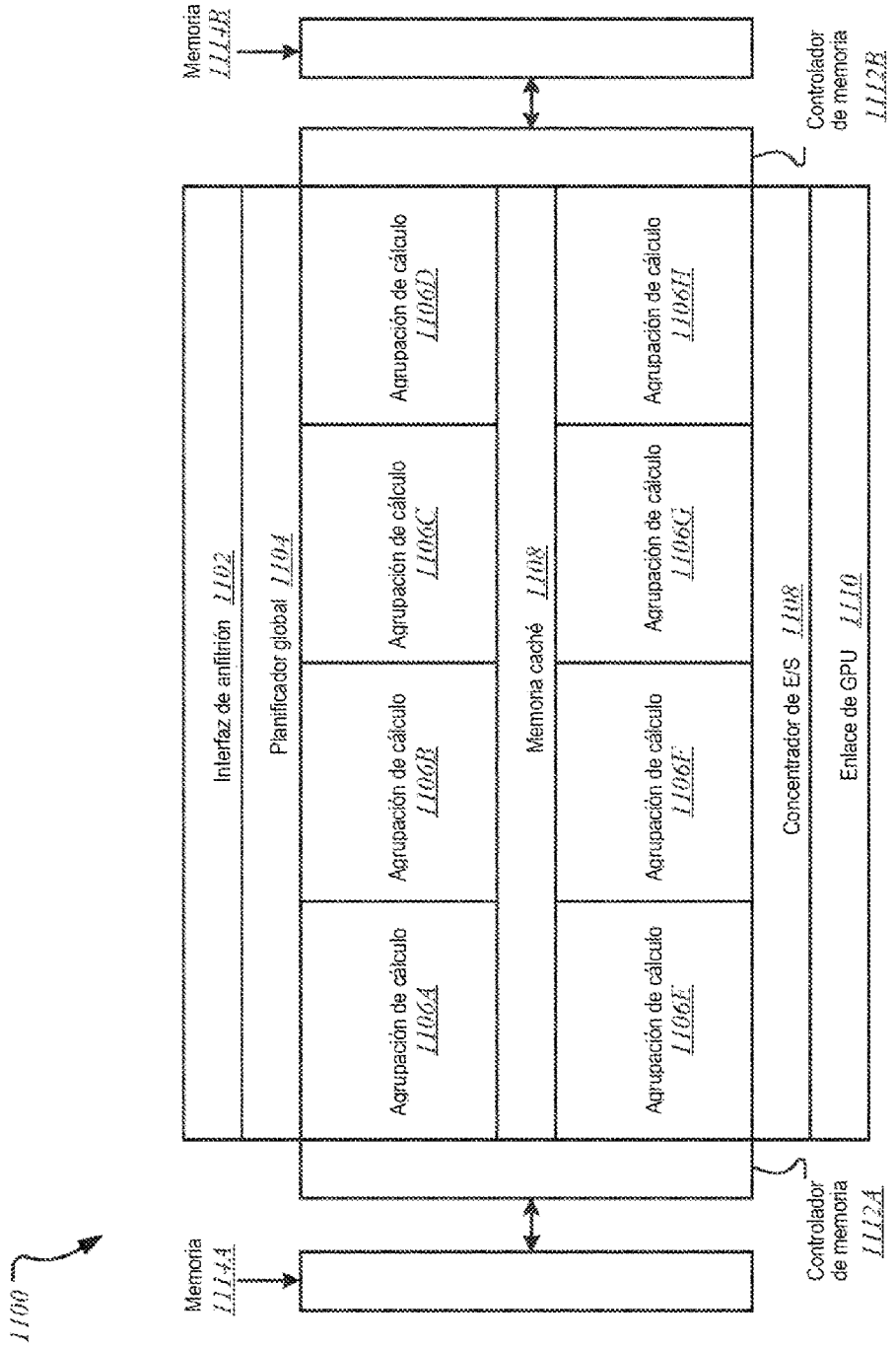

1100
Memoria 1114A
Memoria 1114B
Controlador de memoria 1112A
Controlador de memoria 1112B
Interfaz de anfitrión 1102
Planificador global 1104
Agrupación de cálculo 1106A
Agrupación de cálculo 1106B
Agrupación de cálculo 1106C
Agrupación de cálculo 1106D
Agrupación de cálculo 1106E
Agrupación de cálculo 1106F
Agrupación de cálculo 1106G
Agrupación de cálculo 1106H
Memoria caché 1108
Concentrador de E/S 1118
Enlace de GPU 1110

FIG. 11

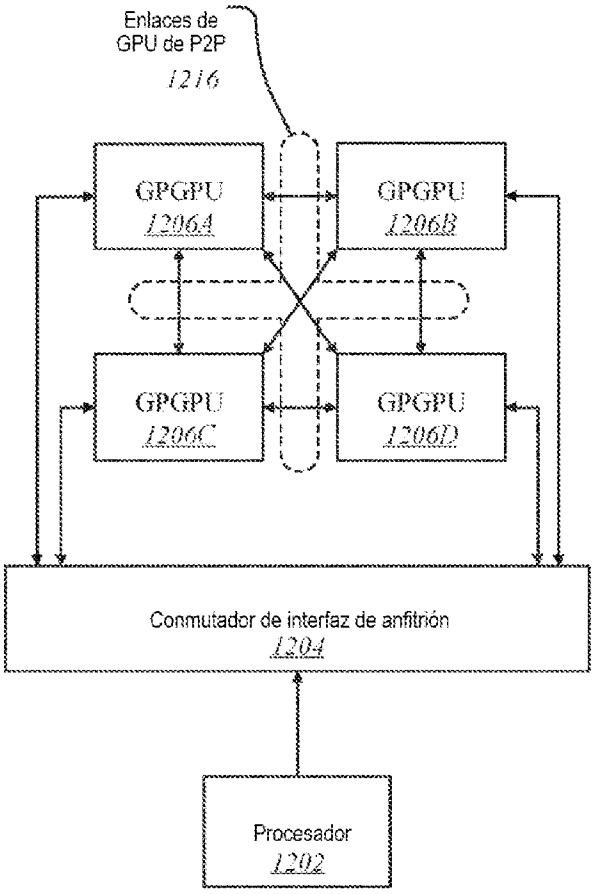

Enlaces de
GPU de P2P
1216
GPGPU
1206A
GPGPU
1206B
GPGPU
1206C
GPGPU
1206D
Conmutador de interfaz de anfitrión
1204
Procesador
1202

FIG. 12

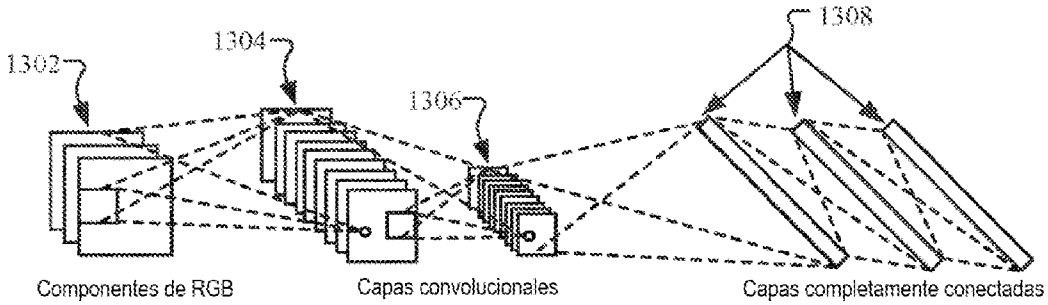


FIG. 13A

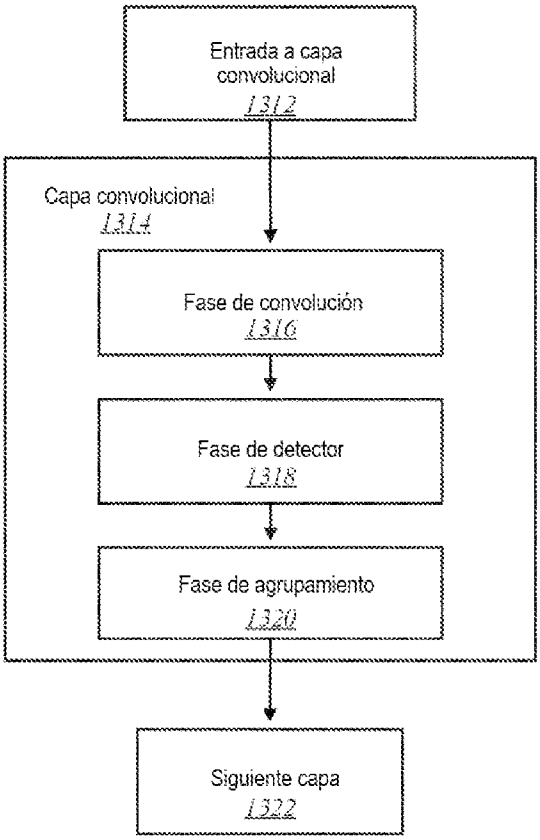


FIG. 13B

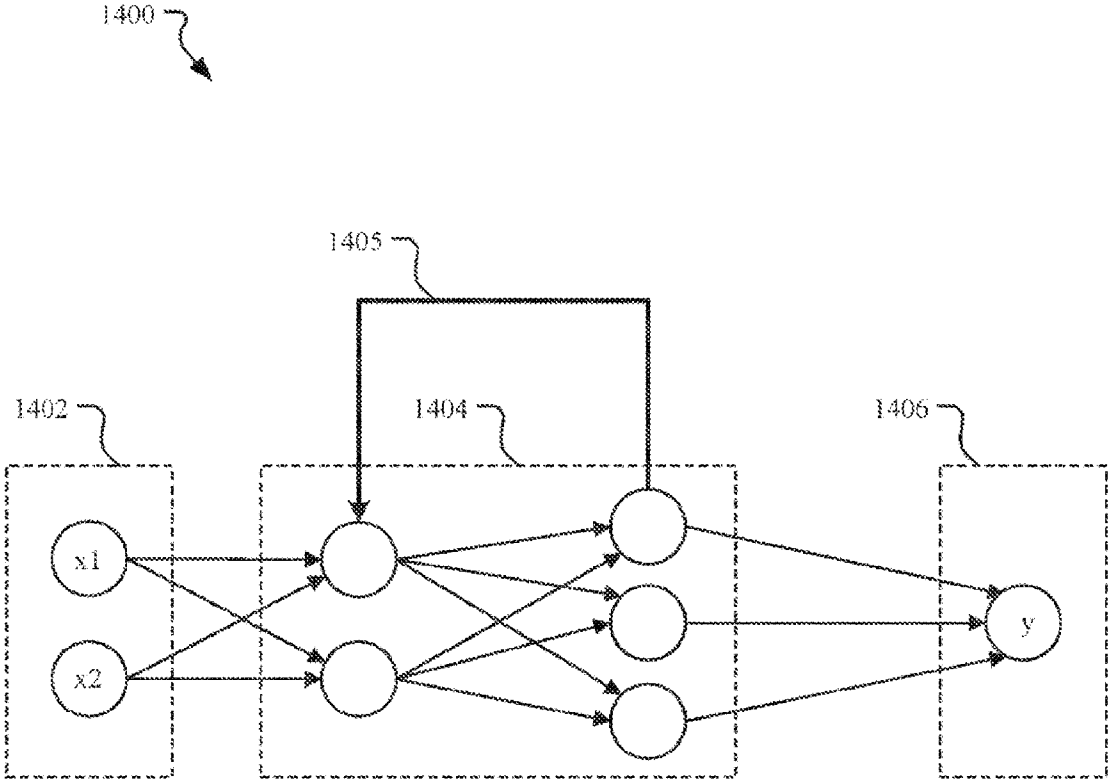


FIG. 14

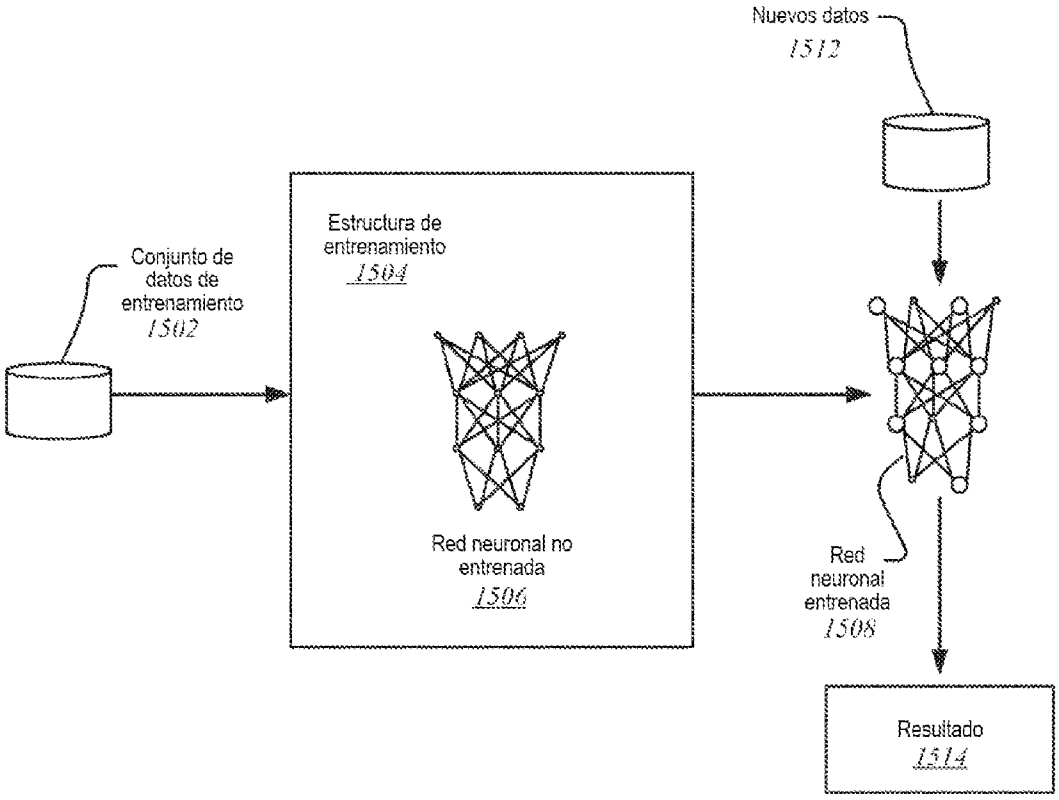

Nuevos datos
1512
Conjunto de datos de entrenamiento
1502
Estructura de entrenamiento
1504
Red neuronal no entrenada
1506
Red neuronal entrenada
1508
Resultado
1514

FIG. 15

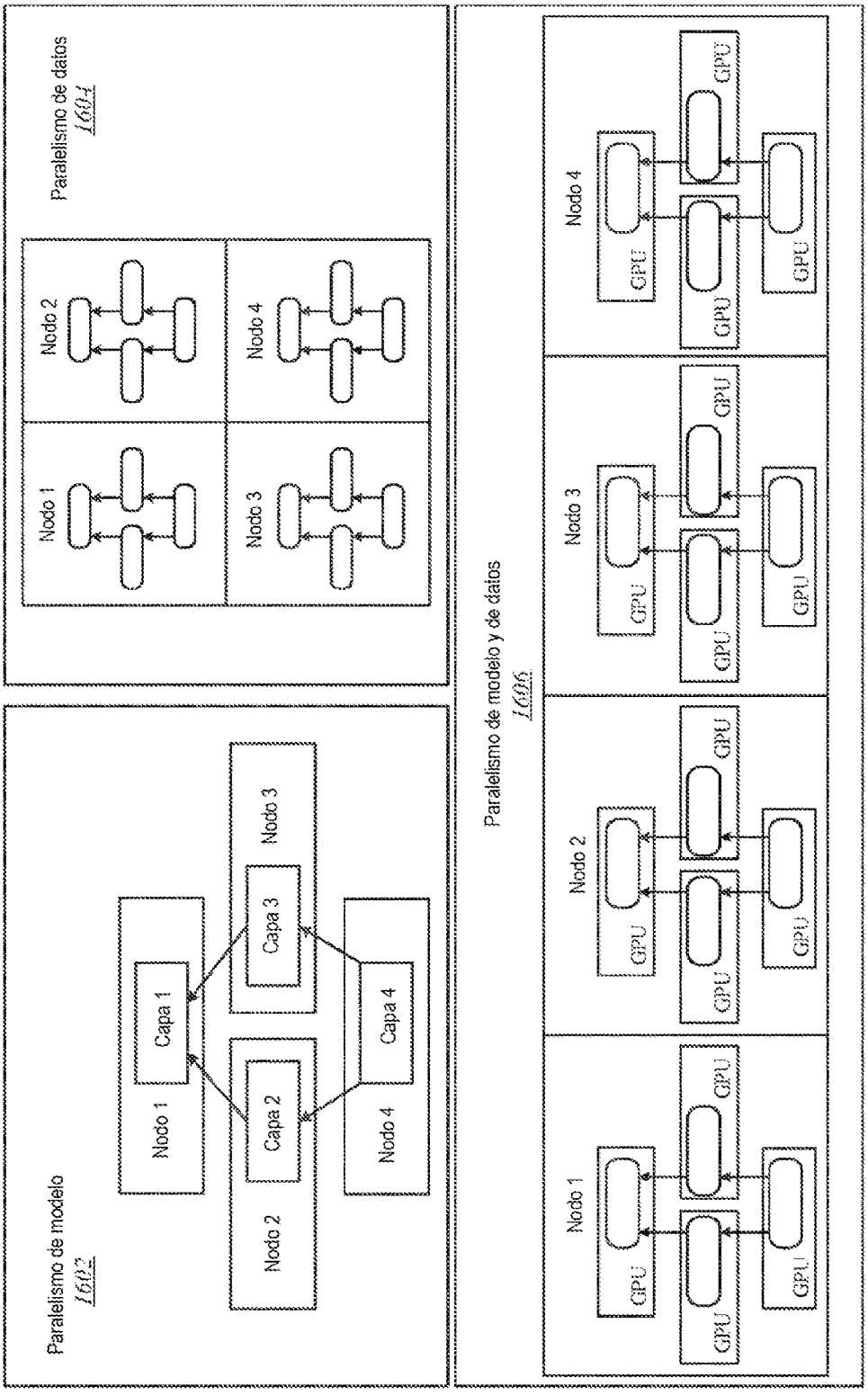

Paralelismo de datos
1604
Nodo 1
Nodo 2
Nodo 3
Nodo 4
Paralelismo de modelo
1602
Nodo 1
Capa 1
Nodo 2
Capa 2
Nodo 3
Capa 3
Nodo 4
Capa 4
Paralelismo de modelo y de datos
1606
Nodo 1
Nodo 2
Nodo 3
Nodo 4
GPU

FIG. 16

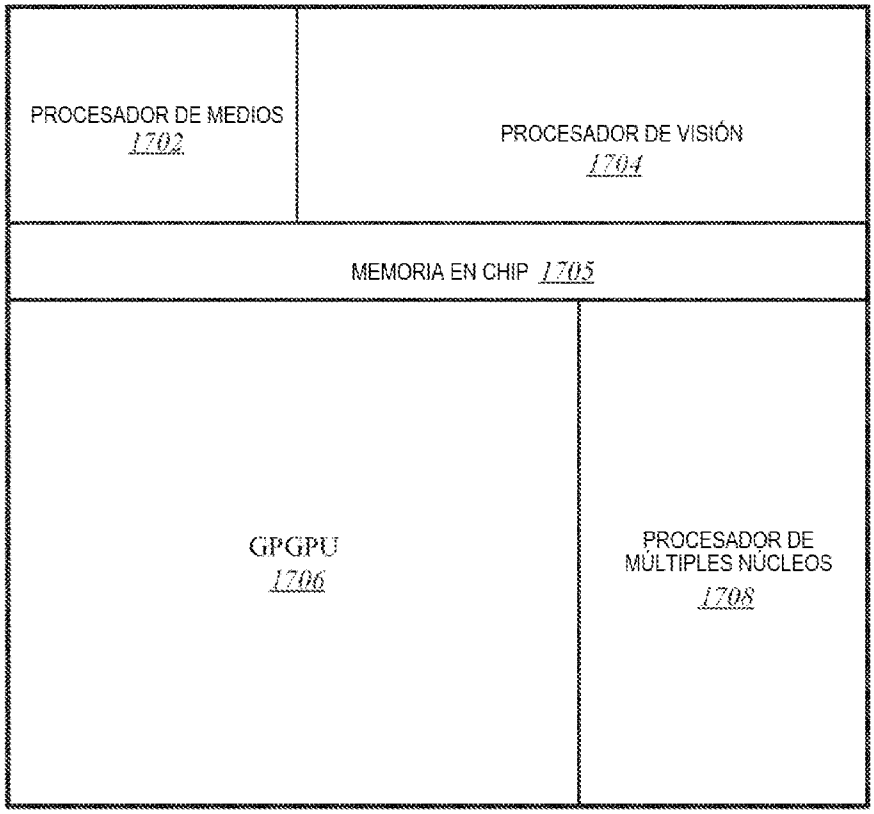

PROCESADOR DE MEDIOS
1702
PROCESADOR DE VISIÓN
1704
MEMORIA EN CHIP 1705
GPGPU
1706
PROCESADOR DE
MÚLTIPLES NÚCLEOS
1708

FIG. 17

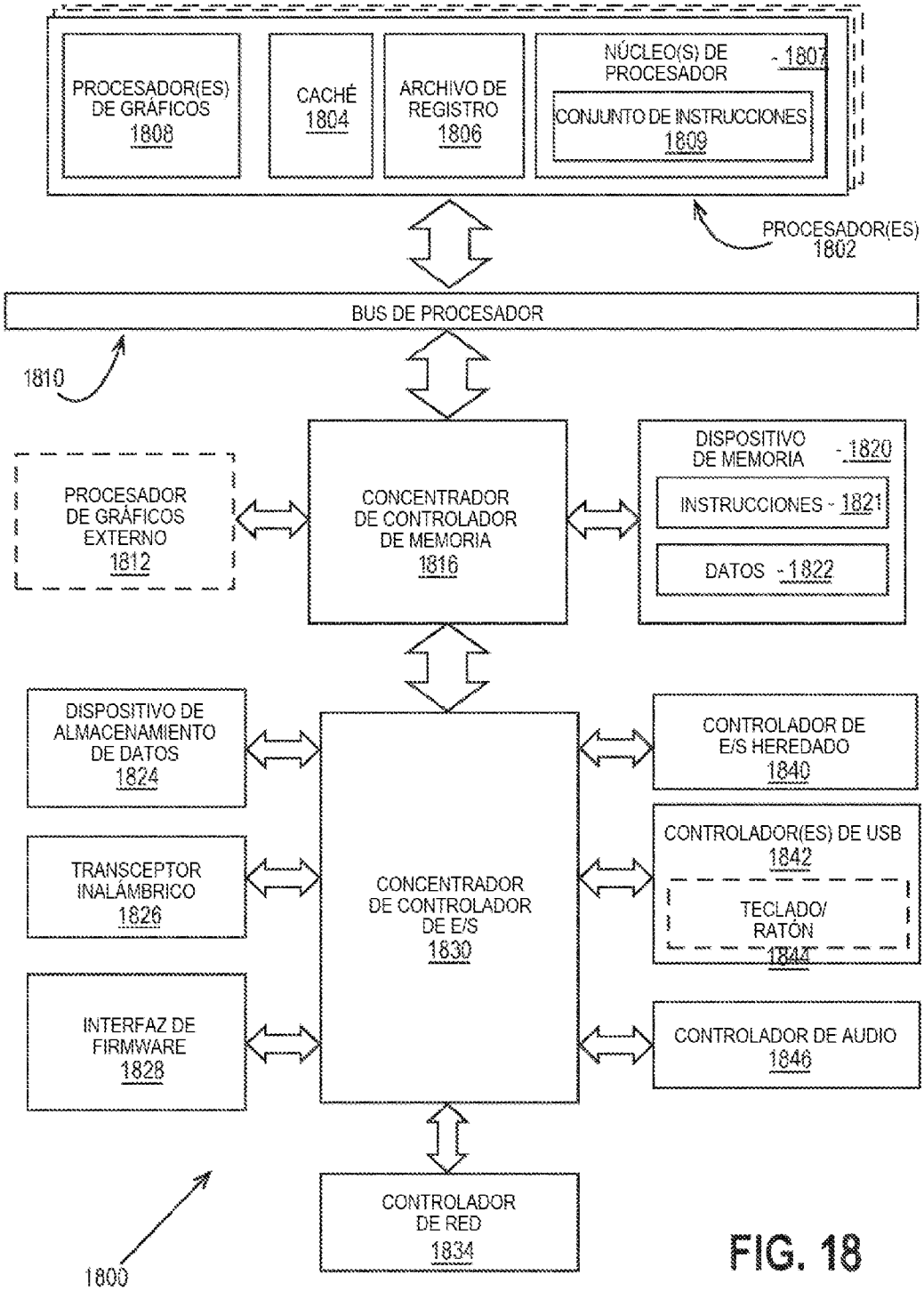

PROCESADOR(ES) DE GRÁFICOS 1808
CACHÉ 1804
ARCHIVO DE REGISTRO 1806
NÚCLEO(S) DE PROCESADOR 1807
CONJUNTO DE INSTRUCCIONES 1809
PROCESADOR(ES) 1802
BUS DE PROCESADOR
1810
PROCESADOR DE GRÁFICOS EXTERNO 1812
CONCENTRADOR DE CONTROLADOR DE MEMORIA 1816
DISPOSITIVO DE MEMORIA 1820
INSTRUCCIONES 1821
DATOS 1822
DISPOSITIVO DE ALMACENAMIENTO DE DATOS 1824
TRANSCEPTOR INALÁMBRICO 1826
INTERFAZ DE FIRMWARE 1828
CONCENTRADOR DE CONTROLADOR DE E/S 1830
CONTROLADOR DE E/S HEREDADO 1840
CONTROLADOR(ES) DE USB 1842
TECLADO/ RATÓN 1844
CONTROLADOR DE AUDIO 1846
CONTROLADOR DE RED 1834
1800

FIG. 18

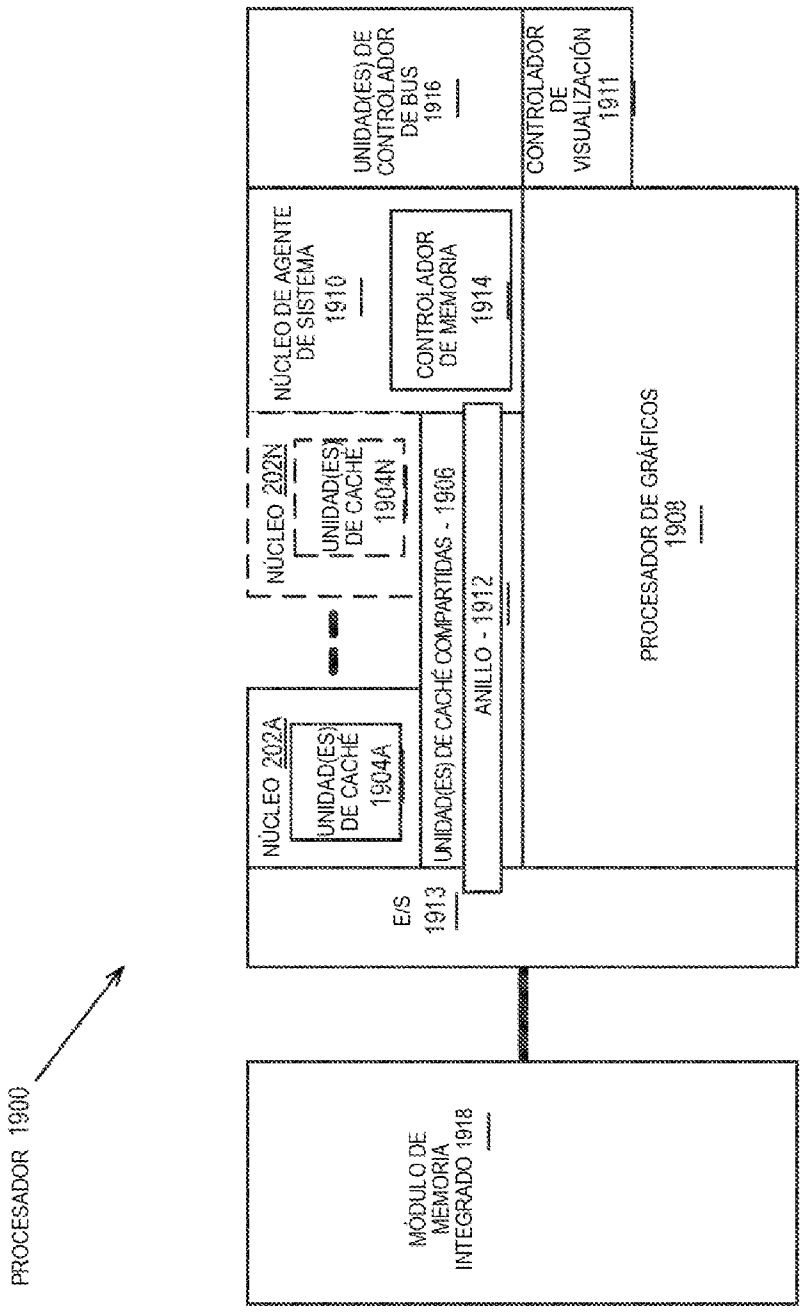

PROCESADOR 1900
MÓDULO DE MEMORIA INTEGRADO 1918
E/S 1913
NÚCLEO 202A
UNIDAD(ES) DE CACHÉ 1904A
NÚCLEO 202N
UNIDAD(ES) DE CACHÉ 1904N
UNIDAD(ES) DE CACHÉ COMPARTIDAS - 1906
ANILLO - 1912
NÚCLEO DE AGENTE DE SISTEMA 1910
CONTROLADOR DE MEMORIA 1914
PROCESADOR DE GRÁFICOS 1908
UNIDAD(ES) DE CONTROLADOR DE BUS 1916
CONTROLADOR DE VISUALIZACIÓN 1911

FIG. 19

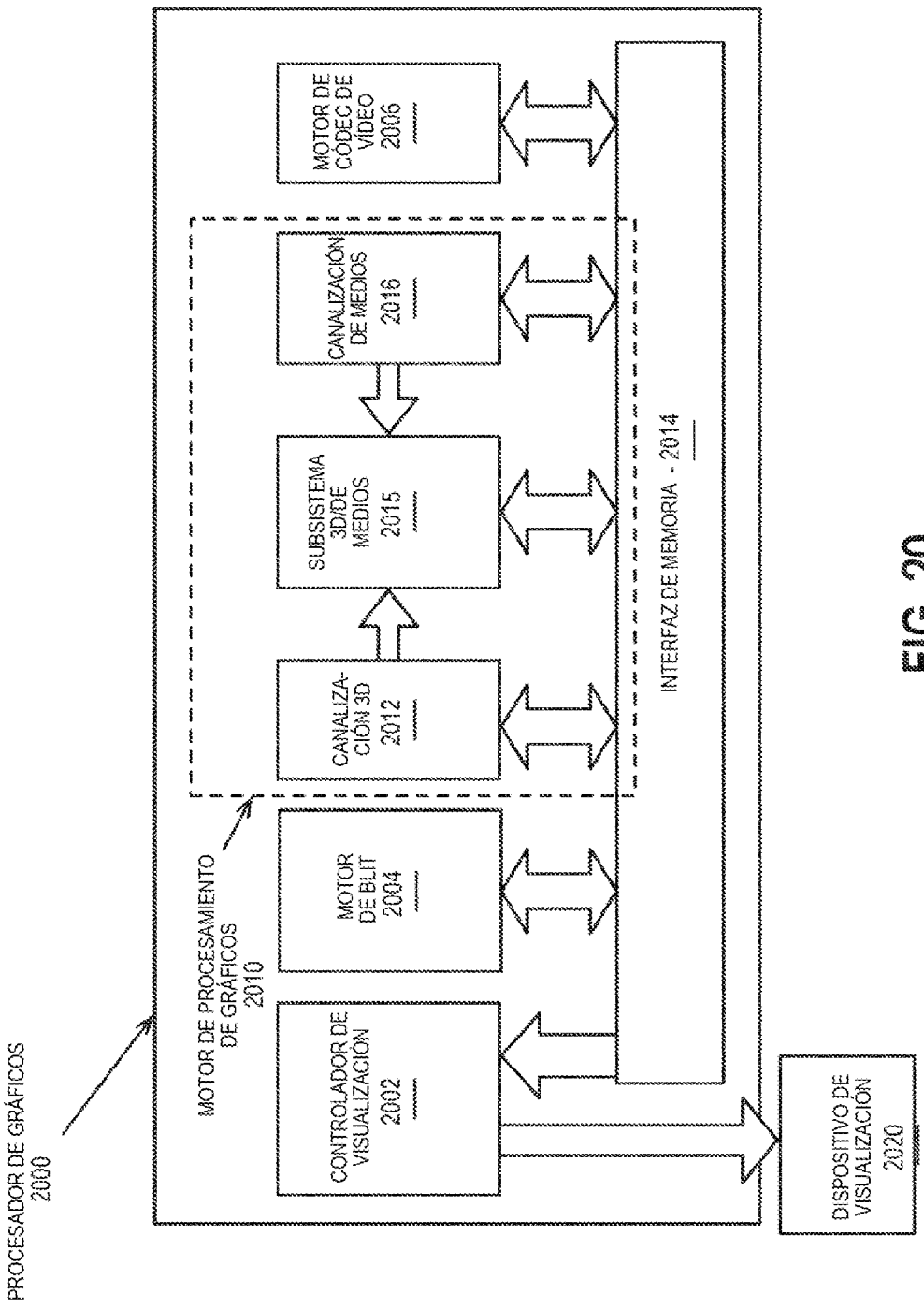

PROCESADOR DE GRÁFICOS
2000
MOTOR DE PROCESAMIENTO DE GRÁFICOS
2010
CONTROLADOR DE VISUALIZACIÓN
2002
MOTOR DE BLIT
2004
CANALIZA-CIÓN 3D
2012
SUBSISTEMA 3D/DE MEDIOS
2015
CANALIZACIÓN DE MEDIOS
2016
MOTOR DE CÓDEC DE VÍDEO
2006
INTERFAZ DE MEMORIA - 2014
DISPOSITIVO DE VISUALIZACIÓN
2020

FIG. 20

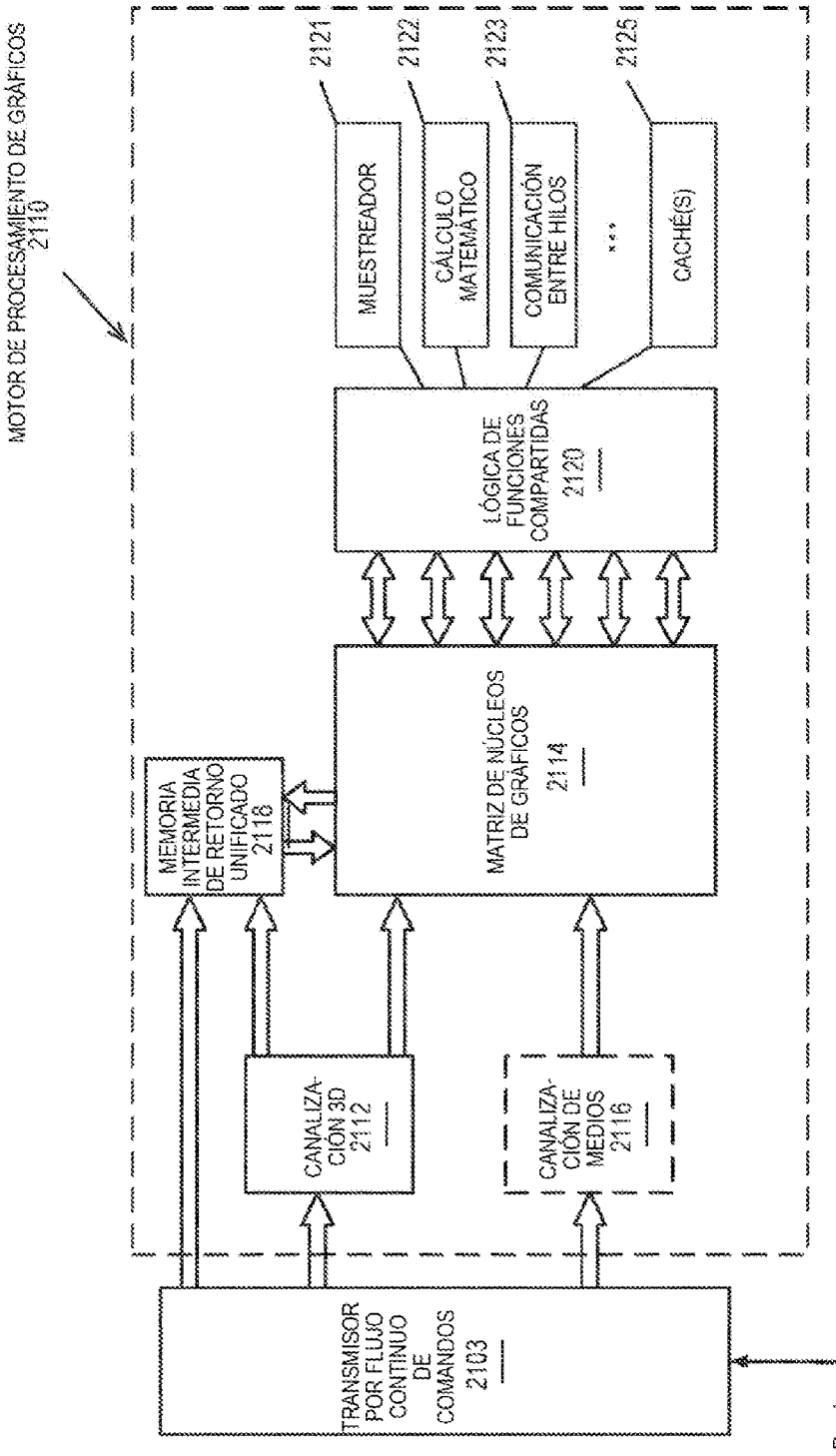

MOTOR DE PROCESAMIENTO DE GRÁFICOS
2110
2121
2122
2123
2125
MUESTREADOR
CÁLCULO MATEMÁTICO
COMUNICACIÓN ENTRE HILOS
...
CACHÉ(S)
LÓGICA DE FUNCIONES COMPARTIDAS
2120
MATRIZ DE NÚCLEOS DE GRÁFICOS
2114
MEMORIA INTERMEDIA DE RETORNO UNIFICADO
2116
CANALIZACIÓN 3D
2112
CANALIZACIÓN DE MEDIOS
2116
TRANSMISOR POR FLUJO CONTINUO DE COMANDOS
2103
Desde memoria

FIG. 21

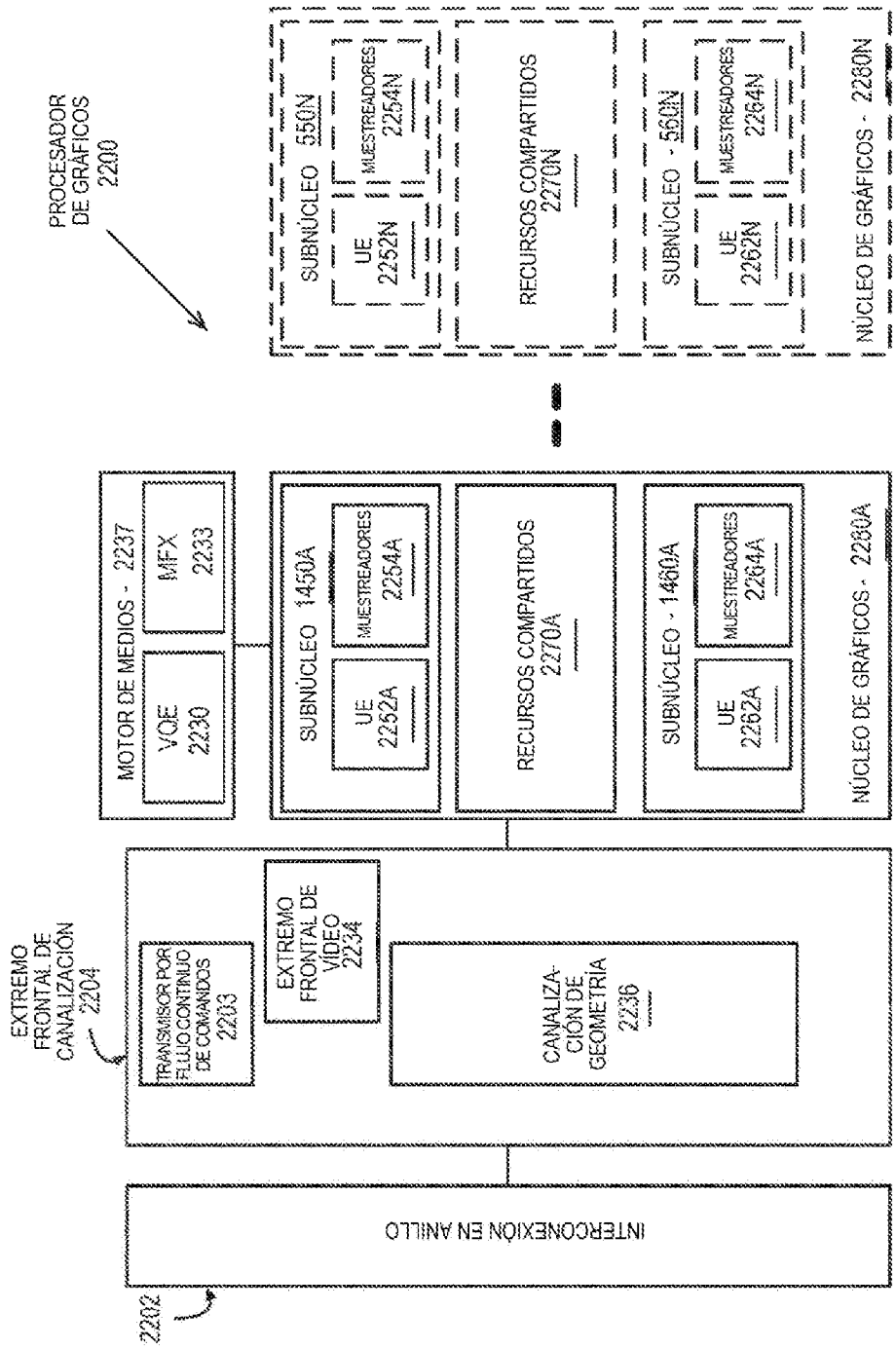

PROCESADOR DE GRÁFICOS 2200
2202
INTERCONEXIÓN EN ANILLO
EXTREMO FRONTAL DE CANALIZACIÓN 2204
TRANSMISOR POR FLUJO CONTINUO DE COMANDOS 2203
EXTREMO FRONTAL DE VÍDEO 2234
CANALIZACIÓN DE GEOMETRÍA 2236
MOTOR DE MEDIOS - 2237
VQE 2230
MFX 2233
SUBNÚCLEO 1450A
UE 2252A
MUESTREADORES 2254A
RECURSOS COMPARTIDOS 2270A
SUBNÚCLEO - 1460A
UE 2262A
MUESTREADORES 2264A
NÚCLEO DE GRÁFICOS - 2260A
SUBNÚCLEO 550N
UE 2252N
MUESTREADORES 2254N
RECURSOS COMPARTIDOS 2270N
SUBNÚCLEO - 560N
UE 2262N
MUESTREADORES 2264N
NÚCLEO DE GRÁFICOS - 2260N

FIG. 22

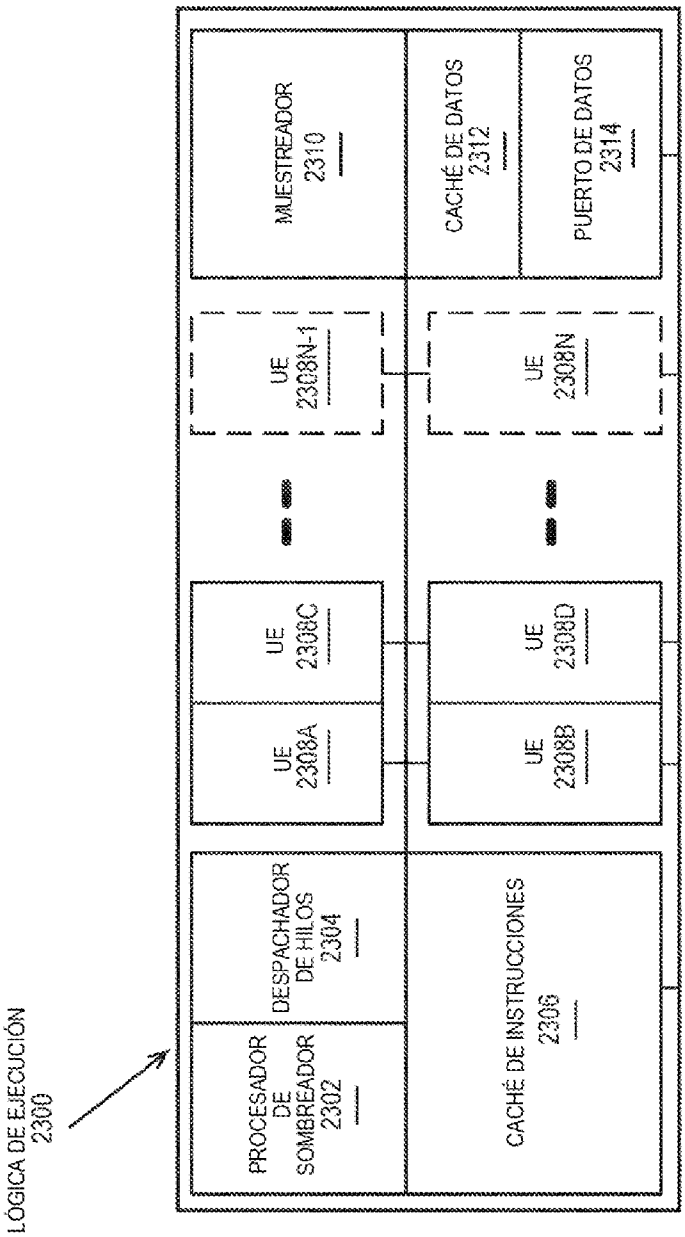

LÓGICA DE EJECUCIÓN
2300
PROCESADOR DE SOMBREADOR
2302
DESPACHADOR DE HILOS
2304
CACHÉ DE INSTRUCCIONES
2306
UE
2308A
UE
2308B
UE
2308C
UE
2308D
UE
2308N-1
UE
2308N
MUESTREADOR
2310
CACHÉ DE DATOS
2312
PUERTO DE DATOS
2314

FIG. 23

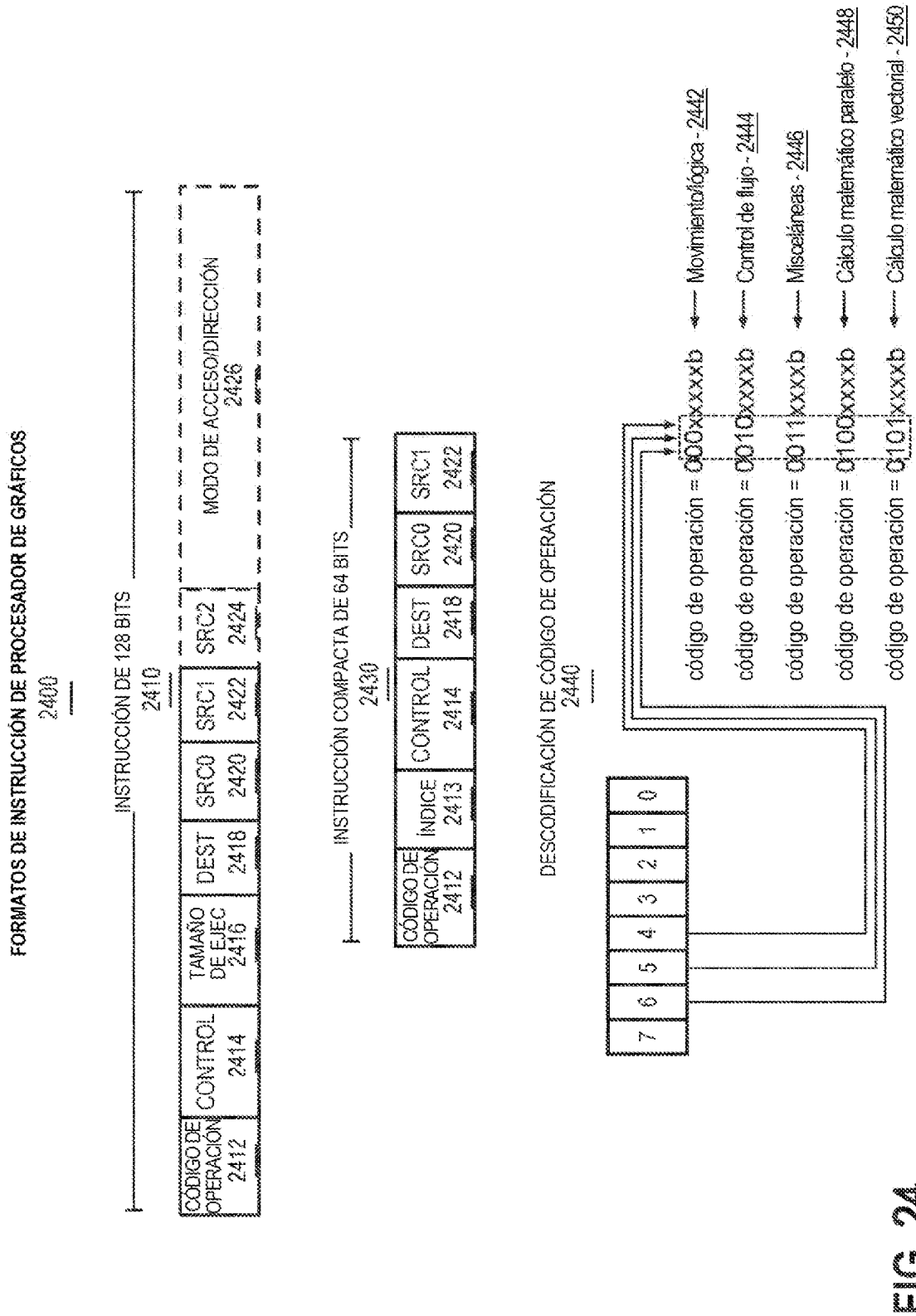

FORMATOS DE INSTRUCCIÓN DE PROCESADOR DE GRÁFICOS
2400
INSTRUCCIÓN DE 128 BITS
2410
CÓDIGO DE OPERACIÓN 2412
CONTROL 2414
TAMAÑO DE EJEC 2416
DEST 2418
SRC0 2420
SRC1 2422
SRC2 2424
MODO DE ACCESO/DIRECCIÓN 2426
INSTRUCCIÓN COMPACTA DE 64 BITS
2430
CÓDIGO DE OPERACIÓN 2412
ÍNDICE 2413
CONTROL 2414
DEST 2418
SRC0 2420
SRC1 2422
DESCODIFICACIÓN DE CÓDIGO DE OPERACIÓN
2440
7 6 5 4 3 2 1 0
código de operación = 000xxxxxb
código de operación = 001Oxxxxb
código de operación = 0011xxxxb
código de operación = 0100xxxxb
código de operación = 0101xxxxb
Movimiento/lógica - 2442
Control de flujo - 2444
Miscelánea - 2446
Cálculo matemático paralelo - 2448
Cálculo matemático vectorial - 2450

FIG. 24

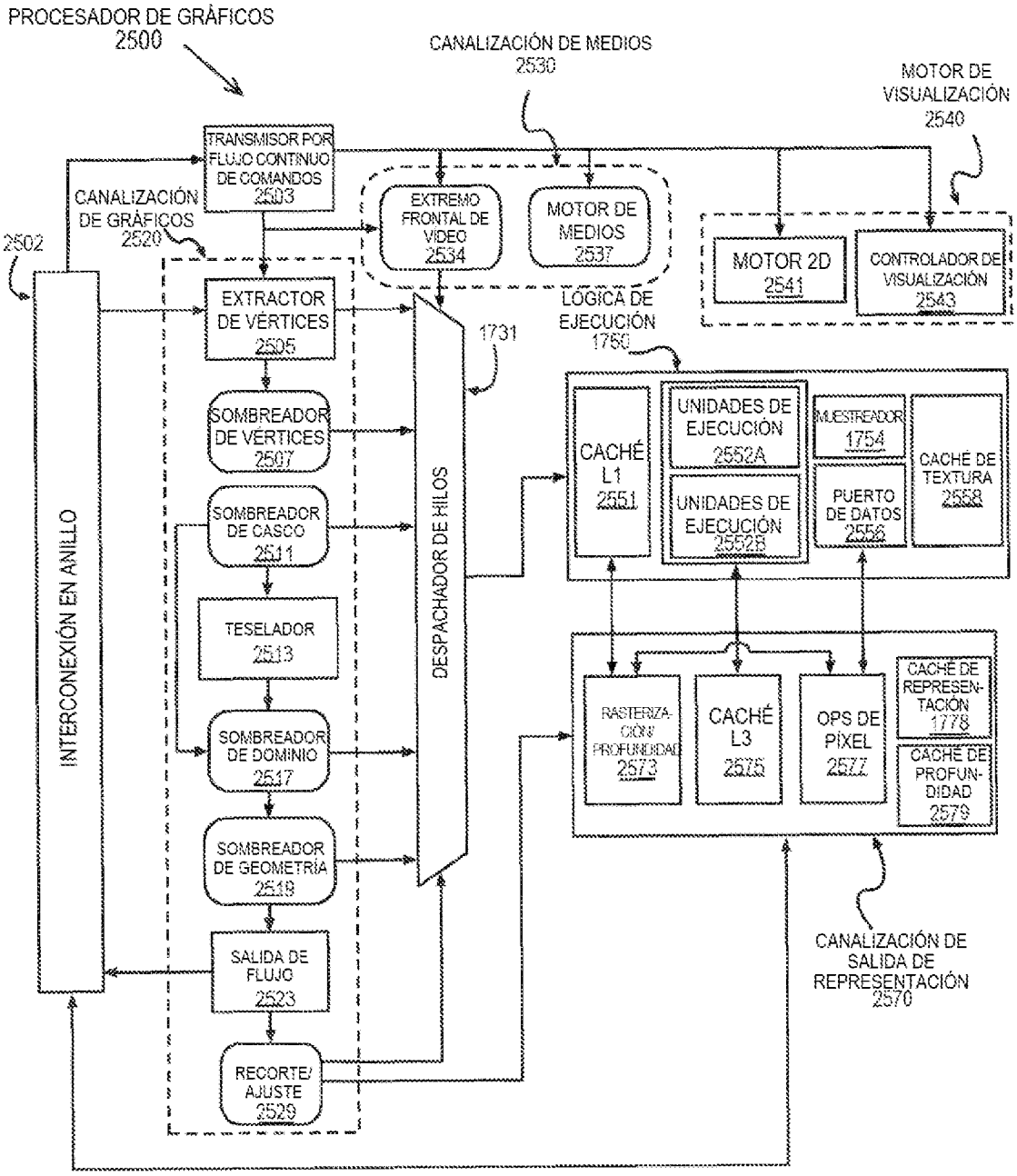

PROCESADOR DE GRÁFICOS
2500
CANALIZACIÓN DE MEDIOS
2530
MOTOR DE VISUALIZACIÓN
2540
TRANSMISOR POR FLUJO CONTINUO DE COMANDOS
2503
CANALIZACIÓN DE GRÁFICOS
2520
2502
EXTREMO FRONTAL DE VÍDEO
2534
MOTOR DE MEDIOS
2537
MOTOR 2D
2541
CONTROLADOR DE VISUALIZACIÓN
2543
EXTRACTOR DE VÉRTICES
2505
LÓGICA DE EJECUCIÓN
1750
1731
SOMBREADOR DE VÉRTICES
2507
SOMBREADOR DE CASCO
2511
TESELADOR
2513
SOMBREADOR DE DOMINIO
2517
SOMBREADOR DE GEOMETRÍA
2519
SALIDA DE FLUJO
2523
RECORTE/ AJUSTE
2529
INTERCONEXIÓN EN ANILLO
DESPACHADOR DE HILOS
CACHÉ L1
2551
UNIDADES DE EJECUCIÓN
2552A
UNIDADES DE EJECUCIÓN
2552B
MUESTREADOR
1754
PUERTO DE DATOS
2556
CACHÉ DE TEXTURA
2558
RASTERIZACIÓN/ PROFUNDIDAD
2573
CACHÉ L3
2575
OPS DE PÍXEL
2577
CACHÉ DE REPRESENTACIÓN
1778
CACHÉ DE PROFUNDIDAD
2579
CANALIZACIÓN DE SALIDA DE REPRESENTACIÓN
2570

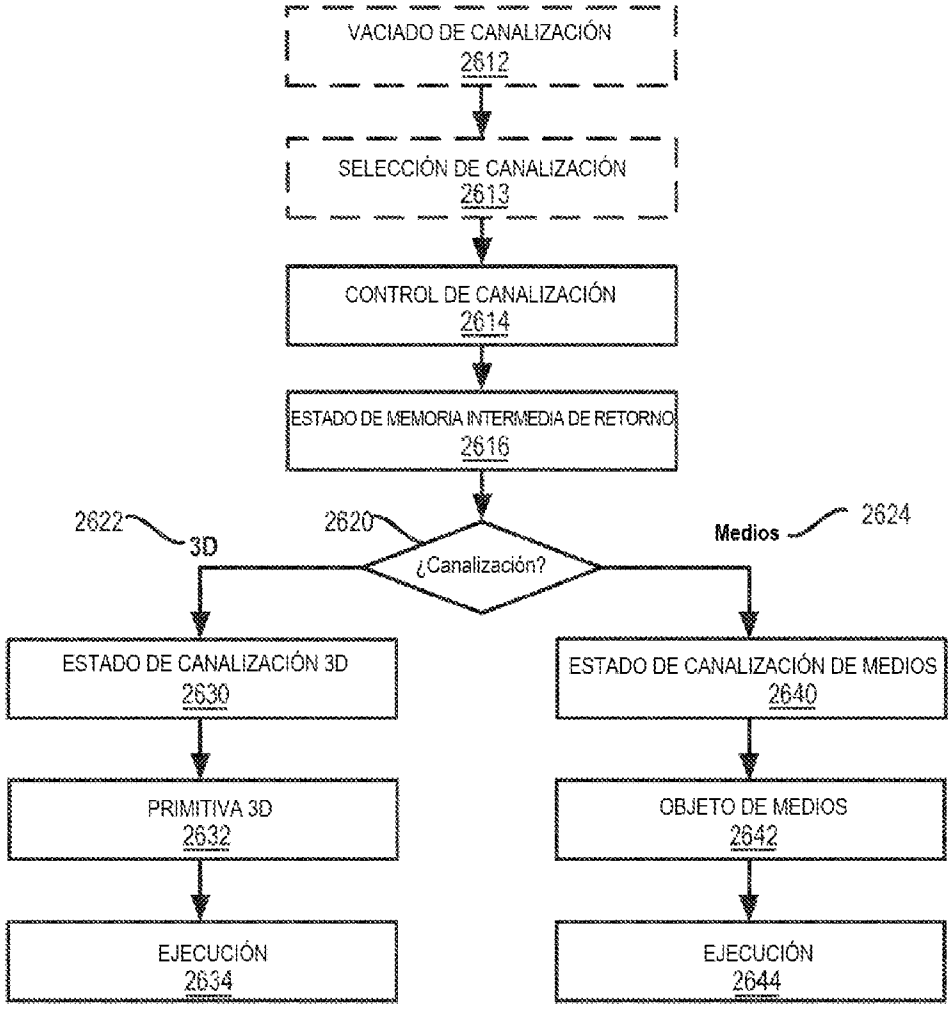
FIG. 25

FIG. 26A FORMATO DE COMANDO DE PROCESADOR DE GRÁFICOS

2600

FIG. 26B SECUENCIA DE COMANDOS DE PROCESADOR DE GRÁFICOS

2610



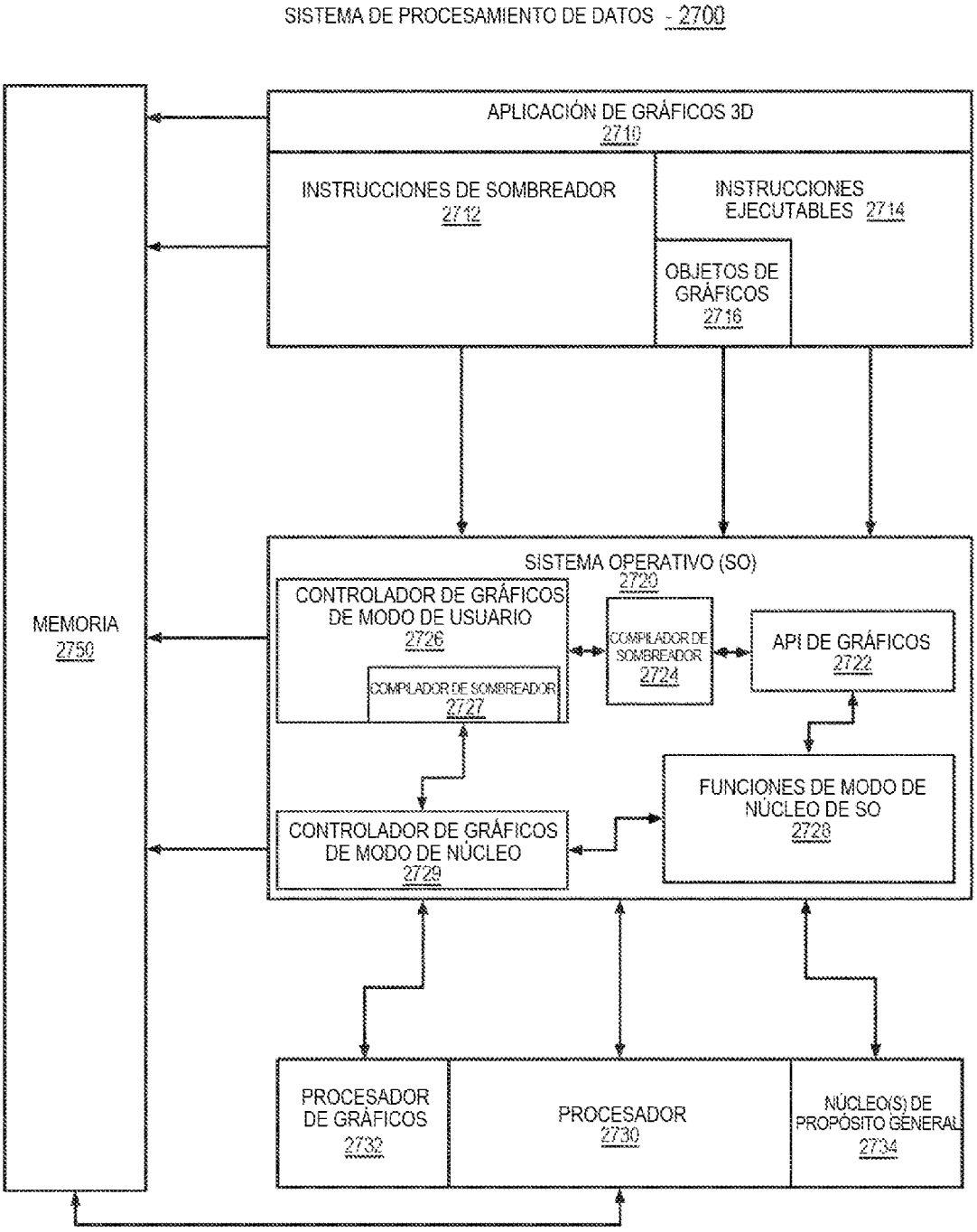

SISTEMA DE PROCESAMIENTO DE DATOS - 2700
APLICACIÓN DE GRÁFICOS 3D
2710
INSTRUCCIONES DE SOMBREADOR
2712
INSTRUCCIONES EJECUTABLES 2714
OBJETOS DE GRÁFICOS
2716
MEMORIA
2750
SISTEMA OPERATIVO (SO)
2720
CONTROLADOR DE GRÁFICOS DE MODO DE USUARIO
2726
COMPILADOR DE SOMBREADOR
2727
COMPILADOR DE SOMBREADOR
2724
API DE GRÁFICOS
2722
FUNCIONES DE MODO DE NÚCLEO DE SO
2728
CONTROLADOR DE GRÁFICOS DE MODO DE NÚCLEO
2729
PROCESADOR DE GRÁFICOS
2732
PROCESADOR
2730
NÚCLEO(S) DE PROPÓSITO GENERAL
2734

FIG. 27

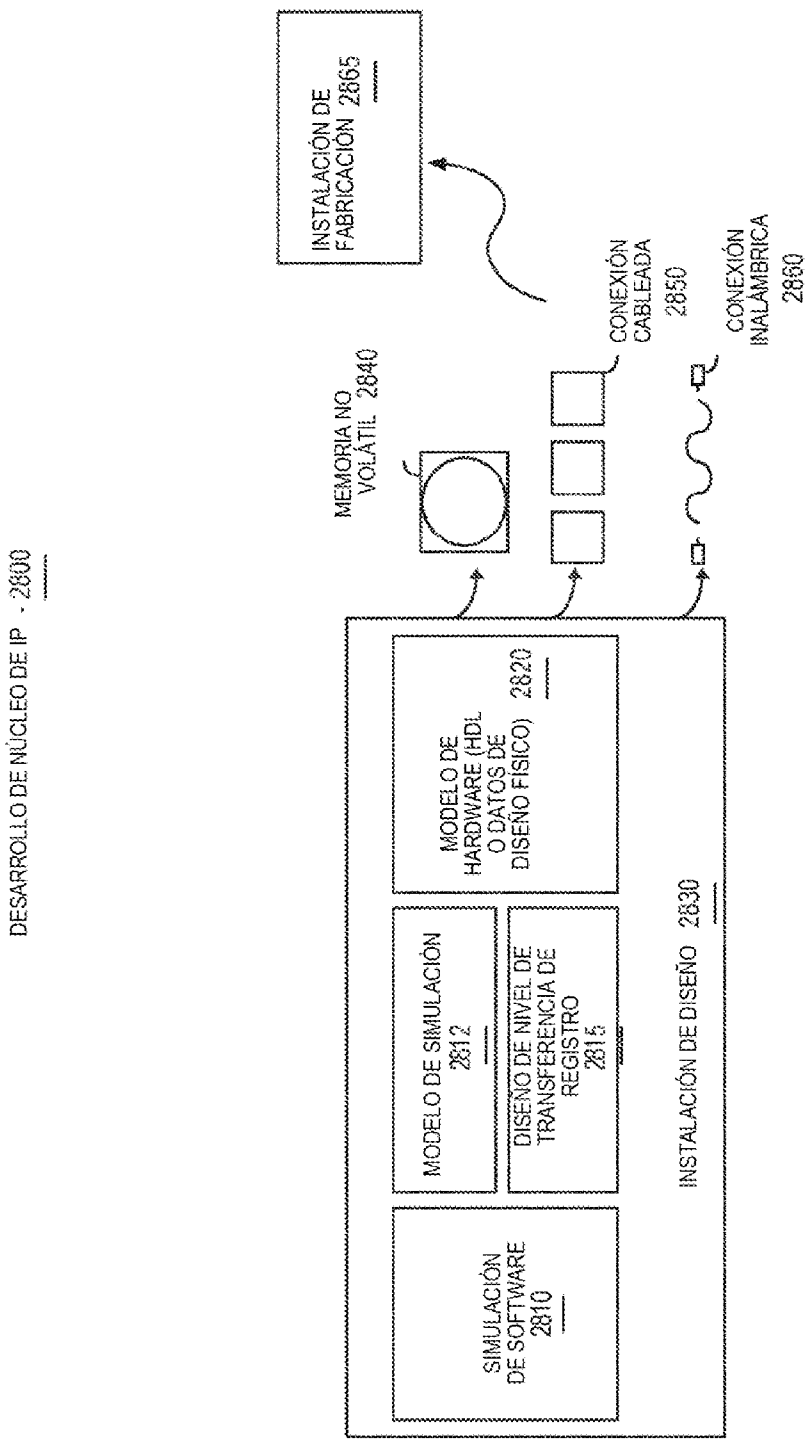

DESARROLLO DE NÚCLEO DE IP - 2600
INSTALACIÓN DE FABRICACIÓN 2665
MEMORIA NO VOLÁTIL 2640
CONEXIÓN CABLEADA 2650
CONEXIÓN INALÁMBRICA 2660
SIMULACIÓN DE SOFTWARE 2610
MODELO DE SIMULACIÓN 2612
DISEÑO DE NIVEL DE TRANSFERENCIA DE REGISTRO 2615
MODELO DE HARDWARE (HDL O DATOS DE DISEÑO FÍSICO) 2620
INSTALACIÓN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

FIG. 28

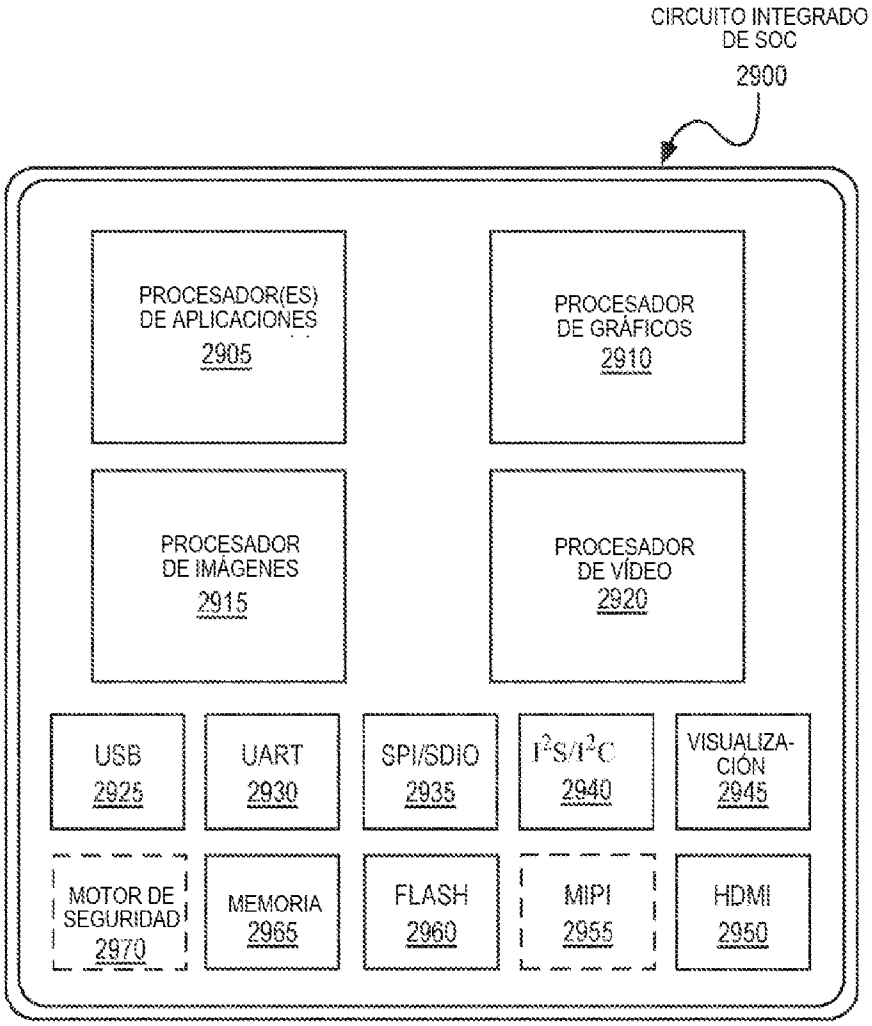

CIRCUITO INTEGRADO DE SOC
2900
PROCESADOR(ES) DE APLICACIONES
2905
PROCESADOR DE GRÁFICOS
2910
PROCESADOR DE IMÁGENES
2915
PROCESADOR DE VÍDEO
2920
USB
2925
UART
2930
SPI/SDIO
2935
I²S/I²C
2940
VISUALIZA-CIÓN
2945
MOTOR DE SEGURIDAD
2970
MEMORIA
2965
FLASH
2960
MIPI
2955
HDMI
2950

FIG. 29

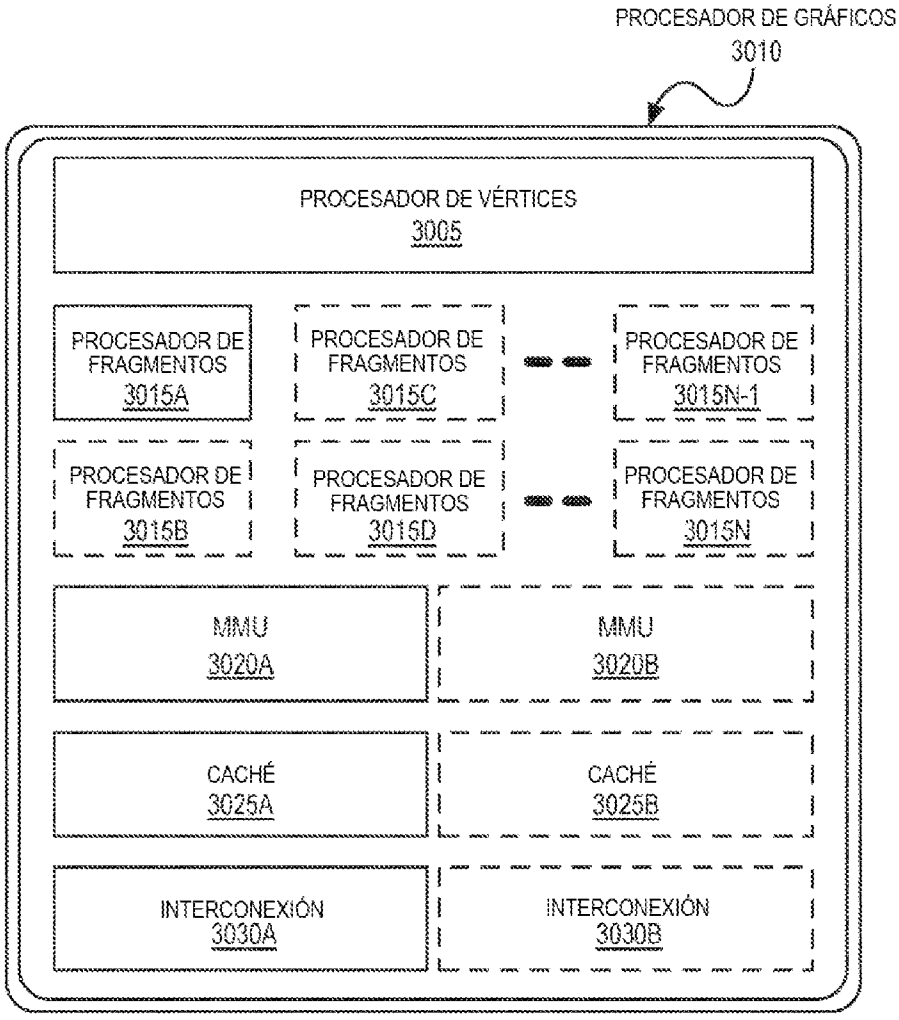

PROCESADOR DE GRÁFICOS
3010
PROCESADOR DE VÉRTICES
3005
PROCESADOR DE FRAGMENTOS
3015A
PROCESADOR DE FRAGMENTOS
3015C
PROCESADOR DE FRAGMENTOS
3015N-1
PROCESADOR DE FRAGMENTOS
3015B
PROCESADOR DE FRAGMENTOS
3015D
PROCESADOR DE FRAGMENTOS
3015N
MMU
3020A
MMU
3020B
CACHÉ
3025A
CACHÉ
3025B
INTERCONEXIÓN
3030A
INTERCONEXIÓN
3030B

FIG. 30

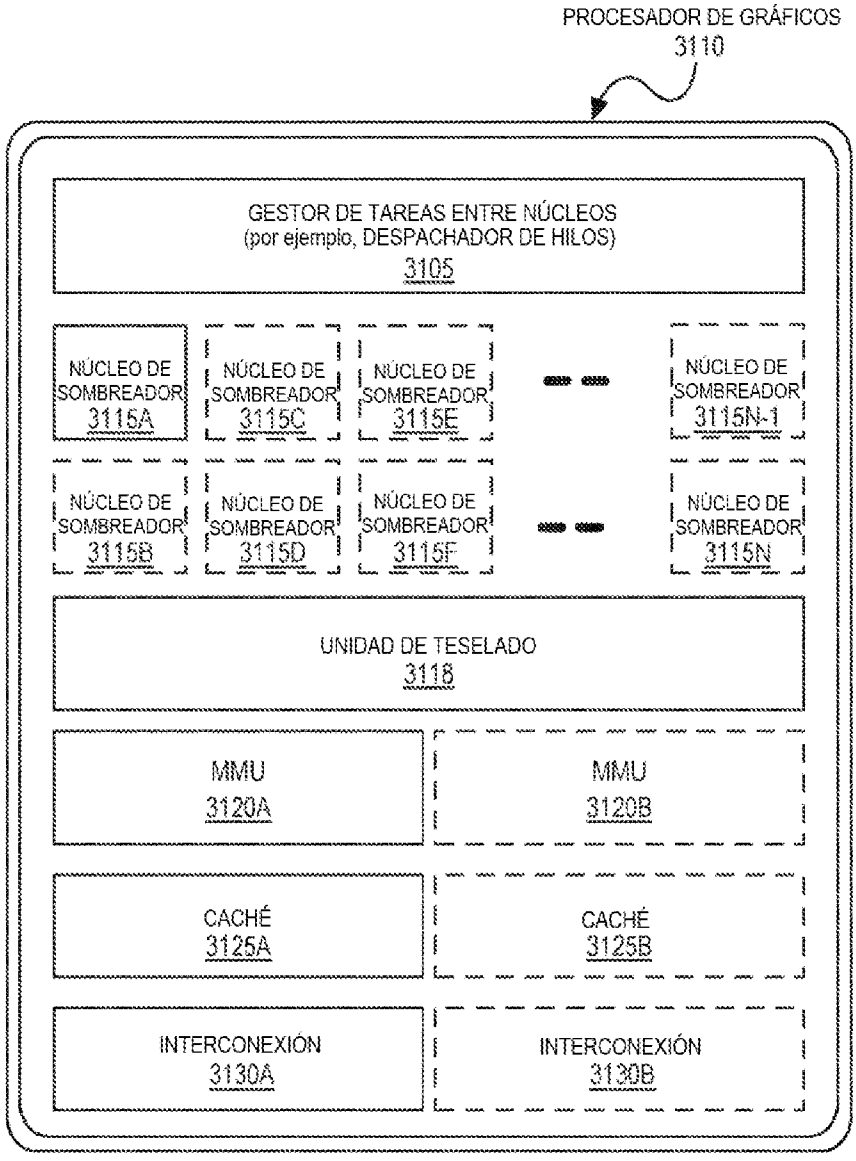

PROCESADOR DE GRÁFICOS
3110
GESTOR DE TAREAS ENTRE NÚCLEOS
(por ejemplo, DESPACHADOR DE HILOS)
3105
NÚCLEO DE SOMBREADOR
3115A
NÚCLEO DE SOMBREADOR
3115C
NÚCLEO DE SOMBREADOR
3115E
NÚCLEO DE SOMBREADOR
3115N-1
NÚCLEO DE SOMBREADOR
3115B
NÚCLEO DE SOMBREADOR
3115D
NÚCLEO DE SOMBREADOR
3115F
NÚCLEO DE SOMBREADOR
3115N
UNIDAD DE TESELADO
3118
MMU
3120A
MMU
3120B
CACHÉ
3125A
CACHÉ
3125B
INTERCONEXIÓN
3130A
INTERCONEXIÓN
3130B

FIG. 31