

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 635 540**

51 Int. Cl.:

H04N 19/127	(2014.01)	G06T 15/04	(2011.01)
H04N 19/14	(2014.01)		
H04N 19/156	(2014.01)		
H04N 19/176	(2014.01)		
H04N 19/182	(2014.01)		
H04N 19/436	(2014.01)		
H04N 19/44	(2014.01)		
H04N 19/80	(2014.01)		
G06T 9/00	(2006.01)		
G06T 11/00	(2006.01)		

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **24.02.2010 PCT/SE2010/050217**
- 87 Fecha y número de publicación internacional: **03.06.2011 WO11065886**
- 96 Fecha de presentación y número de la solicitud europea: **24.02.2010 E 10833652 (0)**
- 97 Fecha y número de publicación de la concesión europea: **03.05.2017 EP 2504814**

54 Título: **Sistema y método de decodificación utilizable en bloques de elementos de textura codificada**

30 Prioridad:

24.11.2009 US 263922 P
03.12.2009 US 266316 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
04.10.2017

73 Titular/es:

TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)
(100.0%)
164 83 Stockholm, SE

72 Inventor/es:

STRÖM, JACOB

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 635 540 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sistema y método de descodificación utilizable en bloques de elementos de textura codificada

5 CAMPO TÉCNICO

La presente invención se refiere, en general, a la descodificación y, en particular, a un sistema y método de descodificación utilizable en bloques de elementos de textura codificada.

10 ANTECEDENTES

Cuando se intenta aumentar el rendimiento de las unidades de procesamiento gráfico (GPU, por sus siglas en inglés), una solución es aplicar varias técnicas para reducir el consumo de ancho de banda de memoria. La reducción de ancho de banda se está convirtiendo también en algo cada vez más importante dado que la tasa de crecimiento de rendimiento para la potencia de procesamiento es mucho mayor que la tasa de crecimiento de rendimiento para el ancho de banda y latencia para la memoria de acceso aleatorio (RAM, por sus siglas en inglés).

15 La compresión de textura es un modo popular de reducir los requisitos de ancho de banda. Mediante el almacenamiento de texturas en forma comprimida en la memoria y la transferencia de bloques de datos comprimidos en el bus, el ancho de banda de textura se reduce de manera sustancial.

20 Actualmente, los sistemas de compresión de textura más usados son DXTC [1] para sistemas basados en Windows y Xbox, y ETC [2] para microteléfonos móviles. Ambos sistemas dividen una imagen, textura denotada, en bloques de elementos de textura (télxel) de 4 x 4 téxeles y los datos rojos, verdes y azules (RGB, por sus siglas en inglés) de los téxeles se comprimen luego $(8+8+8) \times 16 = 384$ bits a 64 bits. Por consiguiente, a cada bloque télxel se le da el mismo número de bits. Esto es importante dado que el rasterizador usado en la descompresión puede necesitar acceder a una parte de la textura y necesita poder calcular, de forma fácil, la dirección de memoria del bloque télxel deseado. En otras palabras, un códec de tasa fija, a saber, un códec donde cada bloque télxel ocupa la misma cantidad de espacio de almacenamiento, es muy deseable y es la norma entre los algoritmos de compresión de textura de hoy en día.

30 El desarrollo del descodificador ETC ha resultado en el así llamado descodificador ETC2 [3] que tiene una funcionalidad adicional para manejar el así llamado modo T, modo H y un modo PLANAR. Los modos T y H se diseñan para manejar bloques télxel con distribución irregular de color y colores dispuestos en dos grupos, respectivamente. Los circuitos de hardware del descodificador ETC se pueden reusar en gran medida para incluir los modos T y H adicionales. El modo PLANAR es, sin embargo, bastante diferente del ETC y de los modos T y H y se diseña para manejar bloques télxel con suaves transiciones de colores. Los circuitos de hardware complejos para el modo PLANAR suponen que la complejidad del descodificador ETC2 aumenta de manera significativa en comparación con el descodificador ETC.

40 Existe, por consiguiente, una necesidad de proveer una implementación eficiente del modo PLANAR y, en particular, de dicha implementación que se puede proveer como implementación de hardware usando un circuito rentable. Dicho circuito rentable es, en particular, deseable si el modo PLANAR se usa junto con el descodificador ETC y los modos T y H para formar el descodificador ETC2.

45 Un problema similar se presenta, en general, en la técnica, a saber, existe una necesidad general de una técnica que se pueda aplicar a un sistema de descodificación con el propósito de reducir la complejidad de implementación.

50 Se-Jeong Park y otros: "A Reconfigurable Multilevel Parallel Texture Cache Memory With 75-GB/s Parallel Cache Replacement Bandwidth", *IEEE Journal of Solid-State Circuits*, IEEE Service Center, Piscataway, NJ, Estados Unidos, vol. 37, No. 5, 1 mayo 2002 (01-05-2002), XP011065764, ISSN: 0018-9200 describe una memoria caché de textura paralela multinivel reconfigurable con múltiples descodificadores idénticos que trabajan en paralelo, lo cual apunta a reducir el ancho de banda requerido para cargar datos de imágenes de textura y admitir tuberías de gráficos paralelas.

55 COMPENDIO

Es un objetivo general proveer una descodificación eficiente de bloques télxel codificados.

Es un objetivo particular proveer una estructura de descodificación que se pueda implementar, de manera eficiente, en hardware.

60 Estos y otros objetivos se cumplen a través de las realizaciones de la invención según se establece en el conjunto anexo de reivindicaciones; los ejemplos adicionales llamados realizaciones en la descripción son ejemplos ilustrativos, no realizaciones reivindicadas en la presente solicitud.

65 De manera breve, una realización se refiere a un sistema de descodificación para generar un píxel descodificado basado en al menos un bloque télxel codificado. El sistema de descodificación comprende N descodificadores paralelos que se configuran, cada uno, para descodificar un bloque télxel codificado para generar uno o más valores

de t xel. Seg n la realizaci n, los circuitos de los N descodificadores son diferentes en comparaci n unos con otros de modo que cada descodificador de los N descodificadores tiene un circuito  nico que es diferente de los circuitos de los otros $N-1$ descodificadores. Ello permite a los circuitos de descodificador tener diferentes niveles de complejidad dado que los descodificadores tienen capacidades diferentes en t rminos de las posiciones del t xel dentro de un bloque t xel que pueden manejar.

El sistema de descodificaci n comprende tambi n un selector de valor conectado a los N descodificadores y configurado para producir, de manera selectiva, N valores de t xel. Dichos N valores de t xel se producen adem s a partir de uno o m s de los N descodificadores. Los valores de t xel particulares para producir y desde los cuales el descodificador depende de la posici n de un parche de N t xeles, tienen los N valores de t xel, respecto a un bloque t xel que comprende al menos uno de los N t xeles. En otras palabras, la posici n de los N t xeles respecto a los l mites del bloque t xel de una textura dicta qu  resultados del descodificador se seleccionan por el selector de valor y se reenv an adem s a un calculador de p xeles. El calculador de p xeles se configura para calcular un valor de p xel descodificado seg n los N valores de t xel producidos selectivamente por el selector de valor.

La elecci n cuidadosa del selector de valor permite el dise o de los descodificadores de modo que al menos algunos de los N descodificadores pueden tener un circuito menos complejo que los otros descodificadores. Ello es posible limitando algunos de los descodificadores para que solamente puedan descodificar y producir valores de t xel para ciertas posiciones de t xel dentro de un bloque t xel. Como consecuencia, los N descodificadores no necesitan tener circuitos id nticos sino, en claro contraste, diferentes circuitos con diferentes niveles de complejidad en t rminos de las estructuras de hardware por circuito.

Una realizaci n define un m todo para generar un p xel descodificando al menos un bloque t xel codificado. El m todo comprende producir, selectivamente, N valores de t xel a partir de al menos uno de los N descodificadores diferentes, cada uno de los cuales tiene un circuito que es  nico y diferente de los circuitos de los otros $N - 1$ descodificadores. Cada descodificador produce, adem s, al menos un valor de t xel seg n un bloque t xel codificado de entrada. La selecci n de dichos valores de t xel de los N descodificadores para usar para el c lculo del valor de p xel del p xel a descodificar se lleva a cabo seg n la posici n de los N t xeles asociados a los N valores de t xel respecto al l mite de un bloque t xel. Los N valores de t xel seleccionados se usan entonces para calcular el valor del p xel.

Las realizaciones permiten una implementaci n eficiente de una estructura de descodificaci n de p xeles reduciendo la complejidad que se necesita para al menos algunos de los descodificadores paralelos. La complejidad total de la estructura de descodificaci n se reduce, por consiguiente, en comparaci n con los N descodificadores paralelos id nticos, como en la t cnica anterior.

BREVE DESCRIPCI N DE LOS DIBUJOS

La invenci n, junto con objetos y ventajas adicionales de la presente, se podr  comprender mejor mediante referencia a la siguiente descripci n tomada junto con los dibujos anexos, en los cuales:

- Las Figuras 1A a 1D ilustran diferentes posiciones de un parche bilineal respecto a los bordes del bloque t xel;
- la Figura 2 es una visi n general esquem tica de una disposici n de descodificaci n que comprende un sistema de descodificaci n seg n una realizaci n;
- la Figura 3 es una ilustraci n de un sistema de descodificaci n seg n una realizaci n;
- la Figura 4 ilustra, de forma esquem tica, un bloque t xel codificado seg n una realizaci n;
- la Figura 5 ilustra el concepto de emplear una aproximaci n planar de valores de propiedad de t xel en un bloque t xel;
- la Figura 6 es un ejemplo de implementaci n de hardware del cuarto descodificador en la Figura 3;
- la Figura 7 es un ejemplo de implementaci n de hardware del tercer descodificador en la Figura 3;
- la Figura 8 es un ejemplo de implementaci n de hardware del segundo descodificador en la Figura 3;
- la Figura 9 es un ejemplo de implementaci n de hardware del primer descodificador en la Figura 3;
- la Figura 10 es un ejemplo de implementaci n de hardware de un extensor de bits de siete a ocho;
- la Figura 11 es una implementaci n de hardware de un fijador;
- la Figura 12 es otra implementaci n de hardware de un fijador;
- la Figura 13 es un ejemplo de implementaci n de hardware del selector de valor de la Figura 3; y
- la Figura 14 es un diagrama de flujo que ilustra un m todo de descodificaci n seg n una realizaci n.

DESCRIPCI N DETALLADA

A lo largo de los dibujos, los mismos n meros de referencia se usan para elementos similares o correspondientes.

Las realizaciones se refieren, en general, a la descodificaci n o descompresi n de bloques de elementos de textura (t xel) codificada o comprimida con el fin de generar valores de p xeles descodificados.

La descodificaci n se adapta bien para su uso con gr ficos tridimensionales (3D) como, por ejemplo, juegos, mapas y escenas 3D, mensajes 3D, p.ej., mensajes animados, protectores de pantalla, interfaces hombre-m quina (MMI,

por sus siglas en inglés), etc., pero no se limita a ello. Por consiguiente, las realizaciones se pueden emplear también para descodificar otros tipos de imágenes o gráficos, p.ej., imágenes unidimensionales (1D), bidimensionales (2D) o 3D.

5 Un proceso de renderización de gráficos comprende, normalmente, tres subetapas. De manera breve, una primera etapa, la etapa de aplicación, crea varios triángulos. Las esquinas de dichos triángulos se transforman, proyectan e iluminan en una segunda etapa, la etapa de geometría. En una tercera etapa, la etapa de rasterización, las imágenes, con frecuencia texturas denotadas, se pueden "pegar" en los triángulos y aumentar así el realismo de la imagen renderizada. La tercera etapa también lleva a cabo la clasificación usando un búfer de profundidad.

10 Sin embargo, la renderización de texturas y, en particular, de gráficos e imágenes 3D, es una tarea computacionalmente costosa en términos de ancho de banda de memoria y potencia de procesamiento requeridos para los sistemas gráficos. Por ejemplo, las texturas son costosas tanto en términos de memoria, las texturas se pueden ubicar en memoria rápida en chip, como en términos de ancho de banda de memoria, se puede acceder a una textura varias veces para dibujar un solo píxel. Asimismo, múltiples texturas se pueden necesitar, a veces, para dibujar un píxel.

15 Con el fin de reducir los requisitos de ancho de banda y potencia de procesamiento, normalmente se emplea un método o sistema de codificación de textura. Dicho sistema de codificación resulta en un uso más eficiente de la memoria en chip costosa y en un ancho de banda de memoria más bajo durante la renderización y, por consiguiente, en un consumo de potencia inferior y/o en una renderización más rápida. Dicha reducción de los requisitos de ancho de banda y potencia de procesamiento es particularmente importante para clientes livianos como, por ejemplo, unidades y teléfonos móviles, con una pequeña cantidad de memoria, poco ancho de banda de memoria y potencia limitada, a saber, alimentados por baterías.

20 En la técnica, las texturas se dividen en los así llamados bloques téxel, cada uno de los cuales comprende un número de téxeles. Dicho bloque téxel es, normalmente, rectangular o cuadrático como, por ejemplo, un tamaño de $M_R \times M_C$ téxeles, donde M_R , M_C son número íntegros positivos con la condición de que tanto M_R como M_C no sean simultáneamente uno. Preferiblemente, $M_R=2m_R$ y $M_C=2m_C$, donde m_R , m_C son íntegros positivos. En una implementación típica, $M_R = M_C$ y las realizaciones preferidas de dicho bloque podrían ser de 4 x 4 téxeles.

25 En la técnica, el elemento de píxel o imagen se emplea, a veces, para denotar un elemento de textura de una textura. En la siguiente descripción, el téxel se emplea, sin embargo, para denotar un elemento de textura de una textura y el píxel se usa para denotar un elemento de imagen generado a partir de dichos téxeles en conexión con la rasterización.

30 Un téxel se asocia a un valor de téxel y un píxel se asocia a un valor de píxel. Los valores de téxel y píxel pueden ser, de manera ventajosa, valores de color. Un valor de color se puede representar en varios formatos de color como, por ejemplo, de color rojo, verde, azul (RGB) o como luminancia y crominancia. Además, un valor de téxel puede representar otras propiedades de un téxel como, por ejemplo, un valor normal o valor Z. En la siguiente descripción, un valor de téxel se ejemplifica, sin limitación, como un valor de color.

35 Un rasterizador típico no accede a la textura un solo téxel por vez. En su lugar, se emplea un pequeño grupo de téxeles como, por ejemplo, téxeles 2x2. Por ejemplo, durante el filtrado bilineal, cuatro de dichos téxeles vecinos se leen para producir un valor filtrado de manera bilineal, a saber, un valor de píxel. En la técnica, dicho grupo de téxeles vecinos coprocesados para generar un valor de píxel es, normalmente, un parche denotado. Por lo tanto, un grupo de téxeles de 2x2 empleado en el filtrado bilineal se denota como un parche bilineal. Dicho concepto se puede ampliar además, por ejemplo, para el filtrado trilineal. El filtrado trilineal usa dos parches bilineales a partir de dos mapas MIP diferentes.

40 Un problema con el manejo de parches de téxeles es que los téxeles en un parche pueden abarcar más de un bloque téxel. Ello se ilustra de forma esquemática en las Figuras 1A a 1D. La Figura 1A ilustra el caso donde un parche bilineal 20 de cuatro téxeles 30-36 se encuentra en un solo bloque téxel 10. Las Figuras 1B y 1C ilustran otra situación donde el parche bilineal 20 abarca dos bloques de píxeles 10, 12, 14 ya sea presentes lado a lado (Figura 1B) o unos sobre otros (Figura 1C). El caso extremo se ilustra en la Figura 1D con un téxel 30-36 del parche bilineal 20 en un bloque téxel 10-16 respectivo.

45 El modo tradicional de resolver este problema es usar cuatro descodificadores idénticos, cada uno de los cuales descodifica un téxel predeterminado en el parche bilineal. Por ejemplo, en el caso de la Figura 1A, el bloque téxel codificado que representa el bloque téxel 10 se ingresa en los cuatro descodificadores. El primer téxel de salida de descodificador (1, 2) 30, téxel (2, 2) se obtiene del segundo descodificador con el tercer y cuarto descodificadores que generan téxel (1, 3) y (2, 3) 36 del parche bilineal, respectivamente. Se asume un sistema de coordenadas en el cual el origen se encuentra en el téxel izquierdo superior del bloque téxel 10. En la Figura 1D, el bloque téxel codificado del bloque téxel 10 se ingresa en el primer descodificador para generar un téxel (3, 3) 10, el bloque téxel codificado del bloque téxel 12 se procesa por el segundo descodificador para obtener un téxel (0, 3) 32. El téxel (3,

0) 34 se obtiene del tercer descodificador según el bloque téxel codificado para el bloque téxel 14 y el cuarto descodificador produce el téxel (0, 0) 36 a partir del bloque téxel codificado del bloque téxel 16.

De esta manera, es, por lo tanto, posible producir un parche bilineal 2x2 en cada ciclo de reloj independientemente del tipo de configuración (Figuras 1A-1D) que se encuentre. El hardware de rasterización de alto rendimiento produce dos de dichos parches bilineales por ciclo de reloj para permitir el filtrado trilineal. Sin embargo, en dicho caso, ocho descodificadores idénticos se necesitan según las técnicas de la técnica anterior.

El concepto de coprocesamiento de téxeles en descodificadores paralelos, sin embargo, aumenta radicalmente la complejidad del sistema de descodificación, lo cual es un problema para los clientes livianos. Por ejemplo, añadir el modo PLANAR previamente mencionado al sistema de descodificación ETC llevaría a un aumento de la complejidad, definida como el número de sumadores, de 6 sumadores a 21 sumadores [3] por descodificador. Por consiguiente, PLANAR usa 3,5 más sumadores que ETC y la complejidad del sistema de descodificación que tiene cuatro descodificadores ETC y cuatro descodificadores PLANAR aumentaría con 3,5.

Las realizaciones, según se describen en la presente memoria, reducen la complejidad de un sistema de descodificación mediante el no uso de descodificadores paralelos idénticos. En claro contraste, diferentes circuitos de descodificador se pueden usar para los descodificadores si los descodificadores se diseñan para manejar diferentes números de téxeles y diferentes posiciones de téxel dentro de un parche.

En una realización general, el sistema de descodificación comprende N descodificadores diferentes. Dicho sistema de descodificación 1 se ilustra en la Figura 3, N ejemplificado como cuatro. En una realización general, N es un íntegro igual a o mayor que dos, preferiblemente igual a cuatro. Cada descodificador 100, 200, 300, 400 tiene un circuito que es diferente del circuito de los otros $N - 1$ descodificadores. Por consiguiente, el sistema de descodificación 1 no comprende descodificadores idénticos como en la técnica anterior sino, en claro contraste, diferentes descodificadores 100, 200, 300, 400 con diferentes circuitos únicos. La diferencia en los circuitos de los N descodificadores 100, 200, 300, 400 supone que la combinación de los elementos de circuito de uno de los descodificadores es diferente de la combinación de los elementos de circuito en los otros descodificadores. Los descodificadores 100, 200, 300, 400 tienen, por lo tanto, diferente costo y complejidad de implementación.

Cada descodificador 100, 200, 300, 400 se configura además para descodificar un bloque téxel codificado para generar al menos un valor de téxel. El sistema de descodificación 1 comprende también un selector de valor 500 conectado a los N descodificadores 100, 200, 300, 400 y configurado para producir, de manera selectiva, al menos N valores de téxel, preferiblemente N valores de téxel. Dichos al menos N valores de téxel se producen a partir de al menos uno de los N descodificadores 100, 200, 300, 400. Además, el selector de valor 500 selecciona cuáles de los al menos N valores de téxel y desde qué descodificador los valores de téxel deben producirse según las posiciones de los al menos N téxeles asociados a los al menos N valores de téxel respecto a un límite de un bloque téxel que comprende al menos uno de los N téxeles. Por consiguiente, las posiciones de los N téxeles, cuyo valor de téxel se debe generar por el sistema de descodificación 1, respecto a un bloque téxel, determinan desde qué descodificador o desde qué descodificadores de los N descodificadores 100, 200, 300, 400, se deben traer los valores de téxel. Un calculador de píxeles 600 del sistema de descodificación 1 se conecta al selector de valor 500 y se configura para calcular un valor de píxel descodificado de un píxel descodificado según los al menos N valores de téxel producidos, de manera selectiva, por el selector de valor 500.

La realización general logra una reducción de la complejidad de los descodificadores 100, 200, 300, 400 limitando algunos de los descodificadores para que solamente puedan manejar téxeles en ciertas posiciones de téxel dentro de un bloque téxel. Como consecuencia, los circuitos de dichos descodificadores 100, 200, 300, 400 se pueden reducir, de manera significativa, en comparación con el caso de la técnica anterior sin dichas restricciones sobre los descodificadores.

En una realización preferida, el primer descodificador 100 de los N descodificadores 100, 200, 300, 400 se configura para descodificar un bloque téxel codificado de entrada para generar N valores de téxel. El selector de valor 500 se configura entonces para producir, de manera selectiva, dichos N valores de téxel a partir del primer descodificador 100 y calculador de píxeles 500 si los N téxeles se posicionan dentro de los límites de un bloque téxel. La Figura 1A ilustra el presente caso con $N = 4$. Por consiguiente, si los cuatro téxeles 30-36 de un parche bilineal 20 se posicionan todos dentro de un solo bloque téxel 10, los cuatro valores de téxel se generan todos por el primer descodificador 100. Ello significa que cualquier salida de los otros descodificadores 200, 300, 400 no se selecciona por el selector de valor 500 ni se usa por el calculador de píxeles 600 para calcular un valor de píxel descodificado.

Si la mitad de los N téxeles 30, 34 (30, 32) se posiciona dentro de los límites de un bloque téxel 10 y la mitad restante de los N téxeles 32, 36 (34, 36) se posiciona dentro de los límites de un bloque de píxeles vecino 12 (14), es preciso ver la Figura 1B (Figura 1C), el selector de valor 500 se configura, preferiblemente, para producir, de manera selectiva, la mitad de los N valores de téxel a partir del primer descodificador 100 y la mitad restante de los N valores de téxel a partir del segundo descodificador 200.

5 En una realización preferida con $N = 4$, el primer descodificador 100 tiene un primer circuito configurado para descodificar un bloque téxel codificado de entrada para generar cuatro valores de téxel. El segundo descodificador 200 tiene un segundo circuito y produce dos valores de téxel según un bloque téxel codificado de entrada. El tercer 300 y cuarto 400 descodificadores comprenden un tercer circuito y un cuarto circuito, respectivamente. El tercer 300 y cuarto 400 descodificadores producen, cada uno, un solo valor de téxel dado un bloque téxel codificado de entrada.

10 En la situación ilustrada en la Figura 1D, preferiblemente los cuatro descodificadores 100, 200, 300, 400 producen valores de téxel que se seleccionan por el selector de valor 500. Por consiguiente, el selector de valor 500 se configura entonces para producir, de manera selectiva, un primer valor de téxel a partir del primer descodificador 100, un segundo valor de téxel a partir del segundo descodificador 200, un tercer valor de téxel a partir del tercer descodificador 300 y, finalmente, un cuarto valor de téxel a partir del cuarto descodificador 400. En el presente caso, cada uno de los cuatro téxeles 30-36 se posiciona dentro de un bloque téxel 10-16 respectivo.

15 La Figura 2 ilustra, de manera esquemática, una estructura de implementación con un sistema de descodificación 1 según una realización. La estructura comprende una memoria 5, normalmente una RAM, que tiene ubicaciones de memoria que comprenden bloques téxel codificados 40. Una unidad de procesamiento gráfico (GPU) 4 se conecta a la memoria 5 a través de un bus de memoria. La GPU 4 incluye un identificador de bloque téxel 2 para identificar un bloque téxel, cuyo bloque codificado 40 se debe traer desde la memoria 5. El identificador de bloque téxel 2 calcula básicamente la dirección del bloque téxel codificado 40 en la memoria 5 y envía una solicitud de acceso a la memoria en el bus a la memoria 5. El bloque téxel codificado 40 presente en la ubicación de memoria asignada o dedicada al bloque téxel se lee desde la memoria 5 y se transfiere en el bus al sistema de descodificación 1 o a un caché 3 de la GPU 4.

25 El sistema de descodificación 1 comprende o se conecta a al menos un caché 3 que tiene ubicaciones caché o áreas para almacenar bloques téxel codificados o valores de téxel descodificados. En una realización alternativa, la GPU 4 puede comprender múltiples cachés 3 como, por ejemplo, un caché de textura y un caché de búfer de color. La GPU 4 se conecta, normalmente, a una pantalla de visualización 6, en la cual se visualiza una imagen descodificada.

30 La GPU 4 y sus unidades incluidas 1, 2, 3 se pueden proveer en uno o más chips, por ejemplo de una tarjeta gráfica, tarjeta de vídeo o una placa madre. La estructura de implementación de la Figura 2 se puede implementar en cualquier dispositivo o unidad de procesamiento de textura o imagen incluidos, por ejemplo, un ordenador, una consola de juegos, un dispositivo portátil como, por ejemplo, un teléfono móvil o un dispositivo de procesamiento de medios.

35 A continuación, las realizaciones del sistema de descodificación se describirán en más detalle en relación con diferentes ejemplos de implementación. Dichos ejemplos se diseñan además para que sean apropiados para manejar bloques téxel codificados según el PLANAR descrito en [3, 4]. De manera breve, PLANAR se diseña para manejar bloques téxel con crominancias suavemente variables. PLANAR aproxima los componentes de color (rojo, verde, azul) en el bloque téxel con una aproximación planar. Con el fin de especificar un plano, es suficiente especificar los colores en tres ubicaciones en el bloque téxel. La Figura 5 ilustra dicho concepto con tres componentes rojos R_0 , R_H y R_V en ciertas posiciones respecto a un bloque téxel 10. Con dicho posicionamiento de los tres componentes rojos, el componente rojo se puede calcular en cualquier lugar en el bloque téxel 10 usando la siguiente ecuación:

$$R(x, y) = \text{redondo} \left(R_0 + \frac{1}{4}(R_H - R_0)x + \frac{1}{4}(R_V - R_0)y \right)$$

50 Los componentes azul y verde se calculan de igual manera que el componente rojo. Un bloque téxel codificado 40 se ilustra en la Figura 4 y comprende, por lo tanto, tres palabras de código de color 41-49 por componente de color, a saber, X_0 , X_H y X_V , donde X es R por el componente rojo, G por el componente verde y B por el componente azul. En una implementación preferida, seis bits se gastan preferiblemente en cada una de las palabras de código rojas 41-43 y las palabras de código azules 47-49 y siete bits para cada una de las palabras de código verdes 44-46 dado que el ojo humano es más sensible al componente verde en comparación con los componentes rojo y azul. El tamaño total del bloque téxel codificado será de 57 bits, el cual encaja exactamente en el modo de 57 bits de ETC2 [3]. Si PLANAR se usa de manera autónoma y no como un modo especial para ETC, el bloque téxel codificado 40 gasta, de manera ventajosa, siete bits para todas, excepto una de, las palabras de código de color 41-47 como, por ejemplo, G_0 que, en cambio, es de ocho bits. El tamaño total será entonces de 64 bits.

60 En una implementación PLANAR, el primer descodificador 100 de la Figura 3 puede descodificar cualquier parche bilineal 2x2 que tenga los cuatro téxeles dentro del bloque téxel, es preciso ver la Figura 1A. El segundo descodificador 200 es menos avanzado y produce valores de téxel para los téxeles en la primera fila o la primera

columna de un bloque téxel, es preciso ver el téxel 32, 36 en la Figura 1B y los téxeles 34, 36 en la Figura 1C. El segundo descodificador 200 produce, por lo tanto, preferiblemente, dos valores de téxel a la vez, ya sea un parche de 2x1 o un parche de 1x2. El tercer descodificador 300 descodifica un bloque téxel codificado para generar un solo valor de téxel, preferiblemente el derecho superior en la coordenada (3, 0), es preciso ver el téxel 34 en la Figura 1D. El cuarto descodificador 400 produce, también preferiblemente, un solo valor de téxel del téxel de origen (0, 0), es preciso ver el téxel 36 en la Figura 1D.

Ello significa que si todos los téxeles 30-36 de un parche bilineal 20 se encuentran dentro del límite de un bloque téxel 10 como en la Figura 1A, la salida del primer descodificador 100 se selecciona por el selector de valor 500. Si dos téxeles 30, 34 (30, 32) se encuentran en un bloque téxel 10 y dos 32, 36 (34, 36) se encuentran en un bloque téxel vecino 12 (14) como se muestra en las Figuras 1B y 1C, los valores de téxel de los dos téxeles izquierdos 30, 34 o superiores 30, 32 se obtienen del primer descodificador 100 y los valores de téxel de los dos téxeles derechos 32, 36 o inferiores 34 provienen del segundo descodificador 200. Finalmente, si los cuatro téxeles 30-36 se encuentran en diferentes bloques téxel 10-16 como en la Figura 1D, el selector de valor 500 selecciona el valor de téxel para el téxel izquierdo superior 30 del primer descodificador 100, el valor de téxel para el téxel derecho superior 32 del segundo descodificador 200, el valor de téxel para el téxel derecho inferior 34 del tercer descodificador 300 y el valor de téxel para el téxel izquierdo inferior 36 del cuarto descodificador 400. De esta manera, los cuatro téxeles 30-36 de un parche bilineal 20 se pueden obtener en un solo ciclo de reloj, con la ventaja de que el segundo al cuarto descodificadores 200, 300, 400 se pueden implementar usando muy poco hardware.

$$X(x, y) = \text{redondo} \left(X_0 + \frac{1}{4}(X_H - X_0)x + \frac{1}{4}(X_V - X_0)y \right)$$

De la ecuación, sigue que el téxel en el origen (0, 0) es

$$X(0,0) = \text{redondo} \left(X_0 + \frac{1}{4}(X_H - X_0)0 + \frac{1}{4}(X_V - X_0)0 \right) = X_0$$

especialmente simple para calcular. Como se describe más arriba, el cuarto descodificador se configura, preferiblemente, para solamente descodificar téxeles en el origen. Ello significa que el circuito del cuarto descodificador se puede volver extremadamente simple. La Figura 6 ilustra el circuito del cuarto descodificador 400 para uno de los componentes de color. Como se ve en la figura, la única estructura es un extensor 480 opcional pero preferido que se configura para extender la primera palabra de código de color de entrada X_0 a una longitud de bit definida, preferiblemente de ocho bits. Como se describe también en la presente memoria en relación con la Figura 10, dicho extensor 480 se puede implementar sin estructuras de hardware y, en su lugar, solo con cables desde donde los respectivos componentes de color se almacenan para la salida del cuarto descodificador 400.

Por lo tanto, el cuarto descodificador 400 se puede implementar sin puertas.

El tercer descodificador 300 debe, según la descripción de más arriba, preferiblemente, manejar téxeles en la

$$X(3,0) = \text{redondo} \left(X_0 + \frac{1}{4}(X_H - X_0) \times 3 + \frac{1}{4}(X_V - X_0) \times 0 \right)$$

posición. El último término no contribuye y se puede eliminar. Con el fin de eliminar el operador redondo(), el resultado se puede calcular en unidades de (1/4), añadir dos y desplazar

$$\gg 2: X(3,0) = ((X_0 \ll 2) + (X_H - X_0) \times 3 + 2) \gg 2$$

a la derecha dos veces, indicado por

La multiplicación por tres se puede intercambiar añadiendo el número a una versión desplazada de sí misma:

$$3t = 2t + t = (t \ll 1) + t \quad \text{con} \quad t = X_H - X_0$$

La ecuación se puede, por lo tanto, escribir como

$$X(3,0) = ((X_0 \ll 2) + (t \ll 1) + t + 2) \gg 2$$

En cada sumador, existe la posibilidad de añadir 1 estableciendo el arrastre en = 1 en lugar de 0. Por lo tanto, la adición de 2 se puede reemplazar cambiando los dos sumadores de esta manera. En la ecuación de más abajo, +!

$$X(3,0) = ((X_0 \ll 2) + !(t \ll 1) + !t) \gg 2$$

se usa para indicar un sumador que tiene el arrastre establecido en uno:

$$X(3,0) = \text{redondo} \left(X_0 + \frac{1}{4}(X_H - X_0) \times 3 + \frac{1}{4}(X_V - X_0) \times 0 \right) = \text{redondo} \left(\frac{1}{4}X_0 + \frac{3}{4}X_H \right)$$

Expresado de otra forma a lo establecido más arriba, la ecuación se puede escribir como

$$X(3,0) = (X_0 + !((X_H \ll 1) + !X_H)) \gg 2$$

Dicha expresión se implementa más fácilmente en hardware en comparación con la presentada más arriba.

Los operadores de desplazamiento se pueden implementar conectando cables de manera diferente. Las únicas operaciones restantes que se necesitan para calcular $X(3,0)$ son tres sumas.

5 La Figura 7 ilustra una realización de los circuitos del tercer descodificador 300. El tercer descodificador 300 comprende un operador de desplazamiento izquierdo 310 para desplazar la segunda palabra de código de color X_H una posición de bit a la izquierda para obtener una secuencia de bits desplazada a la izquierda. El operador de desplazamiento izquierdo 310 se conecta a un primer sumador 320 que tiene el arrastre establecido en uno. El primer sumador 320 adiciona la secuencia de bits desplazada a la izquierda a la segunda palabra de código de color X_H . La salida resultante se añade a la primera palabra de código de color X_0 en un segundo sumador 322 que tiene el arrastre establecido en uno y que se conecta al primer sumador 320. Finalmente, el segundo sumador 322 se conecta a un operador de desplazamiento derecho 330 configurado para desplazar la secuencia de bits correspondiente a la salida del segundo sumador 322 dos posiciones de bit a la derecha para formar la salida final del tercer descodificador 300, a saber el valor de t́xel $X(3,0)$.

15 En una realización opcional pero preferida, la primera palabra de código de color X_0 y la segunda palabra de código de color X_H se extienden o expanden primero preferiblemente de seis o siete bits a ocho bits. Como consecuencia, un primer extensor 380 se dispone preferiblemente conectado al segundo sumador 322 y se configura para extender la primera palabra de código de color X_0 a un tamaño de bit predefinido, preferiblemente de ocho bits. Un segundo extensor 382 se dispone preferiblemente conectado al operador de desplazamiento izquierdo 310 y al primer sumador 320 y se configura para extender o expandir la segunda palabra de código de color X_H al tamaño de bit predefinido, a saber, preferiblemente de ocho bits.

20 Un extensor se puede implementar muy f́cilmente mediante un simple cableado de modo que los k bits ḿs significativos de una secuencia de bits l ($k < l$) se copian a los K bits menos significativos para obtener una secuencia de bits $l+k$. La Figura 10 es una ilustraci3n esquemática de una implementaci3n del primer y segundo extensores 380, 382 apropiados para el componente de color verde. Como se puede ver en la figura, el bit ḿs significativo de la secuencia de 7 bits de entrada se copia a la posici3n del bit menos significativo para obtener una secuencia de 8 bits de salida. Para los componentes rojos y azules, los dos bits ḿs significativos se copian a las dos posiciones de bit menos significativo para ir de una secuencia de 6 bits a una secuencia de 8 bits.

25 Con el tamaño de bit previo mencionado para el bloque t́xel codificado, a saber, seis bits por palabra de código de color para los componentes rojos y azules y siete bits por palabra de código de color para el componente verde y el uso de los dos extensores, los circuitos de hardware del tercer descodificador 300 se pueden implementar con un total de 250,8 puertas, si cada sumador cuesta 4,4 puertas por bit:

R/G/B:	1 sumador de 9 bits $\rightarrow 9 \times 4,4 = 39,6$ puertas
	1 sumador de 10 bits $\rightarrow 10 \times 4,4 = 44$ puertas
Total:	$3 \times 83,6 = 250,8$ puertas

40 No se necesita ning3n fijador para el tercer descodificador 300 como en una implementaci3n preferida del primer y segundo descodificadores, lo cual se describe en la presente memoria.

45 La Figura 8 ilustra una realizaci3n de una implementaci3n de hardware del segundo descodificador 200. El segundo descodificador 200 comprende un primer multiplexor 250 configurado para producir una de la segunda palabra de código de color X_H y la tercera palabra de código de color X_V . La selecci3n de palabra de código de color depende de si la mitad de los cuatro t́xeles se posiciona en la primera fila o primera columna dentro del ĺmite del bloque t́xel, cuyo bloque t́xel codificado se ingresa en el segundo descodificador 200. Si los dos ṕxeles se posicionan en la primera fila, la segunda palabra de código de color X_H se produce desde el primer multiplexor 250, de lo contrario, se produce la tercera palabra de código de color X_V . Un operador de negaci3n 240 se configura para negar la primera palabra de código de color X_0 . Un primer sumador 220 se conecta al primer multiplexor 250 y al operador de negaci3n 240 y adiciona la salida de dichas unidades 240, 250 para formar la diferencia $X_H - X_0$ o $X_V - X_0$. En una implementaci3n alternativa, el operador de negaci3n 240 se reemplaza por un operador NOT por el primer sumador 220 que es un sumador con arrastre establecido en uno. El uso de un operador NOT junto con un sumador con arrastre establecido en uno se prefiere, en general, a un operador de negaci3n con un sumador normal en t́rminos de complejidad de hardware.

55 Un operador de desplazamiento izquierdo 210 se conecta al primer sumador 220 y se configura para desplazar la secuencia de bits correspondiente a la salida de sumador una posici3n de bit a la izquierda.

60 Un segundo multiplexor 252 se configura para producir una de una coordenada de fila y una coordenada de columna de un t́xel. Dicho t́xel es el primer t́xel de los dos t́xeles presentes en el bloque t́xel, cuyo bloque t́xel codificado se ingresa en el segundo descodificador 200. En la figura, S_X denota la coordenada de columna y S_Y denota la coordenada de fila de dicho t́xel. Si los dos t́xeles se posicionan en la primera fila, la coordenada de columna S_X se produce desde el segundo multiplexor 252, de lo contrario, la coordenada de fila S_Y .

Un tercer multiplexor 254 se conecta al operador de desplazamiento izquierdo 210 y al primer sumador 220. El tercer multiplexor 254 selecciona entre $2x(X_{HN}-X_0)=2d$, a saber, la salida del operador de desplazamiento izquierdo 210 y $(X_{HN}-X_0)=d$, a saber, la salida del primer sumador 220. La selección se lleva a cabo según el valor del bit más significativo de la coordenada de fila o columna producido desde el segundo multiplexor 252.

5 Un operador OR 360 se conecta al segundo multiplexor 352 y se configura para llevar a cabo una operación OR en el bit más significativo y el bit menos significativo de la coordenada de fila o columna desde el segundo multiplexor 352. El operador OR 360 y el tercer multiplexor 254 se conectan además a un operador AND 270. Dicho operador AND 270 lleva a cabo una operación AND del bit a bit de salida del multiplexor con el único bit del operador OR 260.
 10 El resultado aquí es que la salida del multiplexor se preserva si la salida OR es 1_{bin} y la salida del operador AND 270 es cero si la salida OR es 0_{bin} . Por lo tanto, la salida del operador AND 270 es $2d, d$ o 0 .

15 Un segundo sumador 222 se conecta al operador AND 270 y se configura para añadir la salida del operador AND 270 y una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por la primera palabra de código de color X_0 . Dicha secuencia de bits corresponde a $f=4xX_0+2$, que se puede implementar desplazando la secuencia de bits de la primera palabra de código de color X_0 dos posiciones de bit a la izquierda e insertando $2=10_{bin}$ como los últimos dos bits, los cuales de otra forma son cero después del desplazamiento. La salida del segundo sumador 222 es, por lo tanto, $2d+f, d+f$ o f .

20 Un tercer sumador 224 se conecta al primer sumador 220 y al segundo sumador 222 y se configura para añadir las salidas de dichos sumadores 220, 222. Dicho tercer sumador 224, por lo tanto, da el resultado $3d+f, 2d+f$ o $d+f$. Un primer operador de desplazamiento derecho 230 se conecta al tercer sumador 224 y se configura para desplazar la secuencia de bits del tercer sumador 224 dos posiciones de bit a la derecha para obtener el valor de téxel del téxel inferior o derecho de los dos téxeles en la primera columna o primera fila en el parche bilineal. Un segundo operador de desplazamiento derecho 232 se conecta al segundo sumador 222 y desplaza las dos posiciones de bit de salida de sumador a la derecha para formar el valor de téxel del superior del téxel izquierdo de los dos téxeles en la primera columna o primera fila.
 25

30 Las palabras de código de color de entrada X_0, X_H, X_V se expanden preferiblemente de seis o siete bits a ocho bits en similitud con el tercer descodificador en la Figura 7. Como consecuencia, los extensores 280, 282, 284 opcionales pero preferidos se disponen preferiblemente en los terminales de entrada para tener así un primer extensor 280 configurado para extender la primera palabra de código de color X_0 antes de la suma de 10_{bin} como los dos bits menos significativos para formar, en el presente caso, una secuencia de 10 bits. El primer extensor 280 se conecta también al operador de negación 240. Un segundo 282 y tercer 284 extensor se conectan al primer multiplexor 250 y extiende la segunda X_H y la tercera X_V palabras de código de color, respectivamente.
 35

40 En una implementación opcional, los operadores de desplazamiento derecho 230, 232 se conectan preferiblemente a un fijador u operador de fijación 290, 292 respectivo. Dichos fijadores 290, 292 se configuran para fijar la salida de los operadores de desplazamiento derecho 230, 232 entre un valor de téxel mínimo y un valor de téxel máximo. Por consiguiente, los fijadores 290, 292 aseguran que la secuencia de bits de salida del segundo descodificador 200 se encuentra dentro de un intervalo de valor permitido. Un ejemplo de dicho intervalo de valor permitido va de 0 a 255 si las secuencias de salida del segundo descodificador 200 son de 8 bits.

45 La Figura 11 ilustra un ejemplo de una implementación de hardware de un fijador 190-196, 290, 292 como, por ejemplo, empleado en el segundo descodificador de la Figura 8 o primer descodificador de la Figura 9. La presente realización de fijador se diseña para fijar una secuencia de 11 bits que potencialmente representa un valor en el intervalo de $[-1024, 1023]$ a una secuencia de 8 bits que representa un valor en el intervalo de $[0, 255]$. El fijador 190-196, 290, 292 comprende ocho operadores OR 701-708, cada uno de los cuales recibe uno de los ocho bits menos significativos de la secuencia de bits de entrada. Las otras entradas respectivas a los operadores OR 701-708 se conectan a la salida de otro operador OR 700. Dicho operador OR 700 recibe los dos bits más significativos de la secuencia de bits de entrada. Las salidas de los ocho operadores OR 701-708 se conectan, cada una, a un operador AND 711-718 respectivo. La otra entrada respectiva de los operadores AND 711-718 recibe el bit de signo de la secuencia de bits de entrada.
 50

55 Una realización de implementación alternativa para el fijador 190-196, 290, 292 se ilustra en la Figura 12. En la presente realización, los ocho operadores OR se reemplazan por ocho operadores NOR 721-724. En consecuencia, los ocho operadores AND de la Figura 11 se reemplazan también por ocho operadores NOR 731-738. Además, un operador NOT 740 se conecta entre los ocho operadores NOR 731-738 y la entrada para el bit de signo de la secuencia de bits de entrada.
 60

El costo de implementación de la realización de hardware ilustrada en la Figura 12 es:

65 $16 \text{ NOR} \rightarrow 16 \times 1 = 16$ puertas
 $1 \text{ OR} \rightarrow 1 \times 1,5 = 1,5$ puertas
 $1 \text{ NOT} \rightarrow 1 \times 0,5 = 0,5$ puertas

ES 2 635 540 T3

Total: $16+1,5+0,5=18$ puertas

El número de puertas para el ejemplo de implementación del segundo descodificador 200 en la Figura 8, excluidos los fijadores, es:

5

R/G/B: 1 sumador de 8 bits $\rightarrow 8 \times 4,4=35,2$ puertas
1 sumador de 11 bits $\rightarrow 11 \times 4,4=48,4$ puertas
1 sumador de 12 bits $\rightarrow 12 \times 4,4=52,8$ puertas

10

1 mux de 8 bits $\rightarrow 8 \times 2,2=17,6$ puertas
1 mux de 10 bits $\rightarrow 10 \times 2,2=22$ puertas
1 mux de 2 bits $\rightarrow 2 \times 2,2=4,4$ puertas

15

1 OR de 1 bit $\rightarrow 1 \times 1,5=1,5$ puertas
1 AND de 10 bits $\rightarrow 10 \times 1,5=15$ puertas
1 NOT de 8 bits $\rightarrow 8 \times 0,5=4$ puertas

Total: $3 \times 200,9=602,7$ puertas

20

Finalmente, la Figura 9 es un ejemplo de una implementación de hardware del primer descodificador 100. El primer descodificador 100 comprende un operador de negación 140 configurado para negar la primera palabra de código de color X_0 . Un primer sumador 120 se conecta al operador de negación 140 y suma la primera palabra de código de color negada a la segunda palabra de código de color X_H para obtener $g=X_H-X_0$. Un segundo sumador 122 se conecta también al operador de negación 140 y suma la primera palabra de código de color negada a la tercera palabra de código de color X_V para obtener $h=X_V-X_0$. En una implementación alternativa, el operador de negación 140 se reemplaza por un operador NOT. El primer sumador 120 y el segundo sumador 122 tienen, ambos, arrastre establecido en uno.

25

30

Un primer operador de desplazamiento izquierdo 110 se conecta al primer sumador 120 y desplaza una posición de bit de salida de sumador a la izquierda para obtener una secuencia de bits desplazada a la izquierda correspondiente a $2g$. Un segundo operador de desplazamiento izquierdo 112 se conecta al segundo sumador 122 y desplaza una posición de bit de salida de sumador a la izquierda para obtener la secuencia de bits de $2h$. Un primer multiplexor 150 se conecta al primer operador de desplazamiento izquierdo 110 y al primer sumador 120 y se configura para seleccionar entre g y $2g$ según el bit más significativo de una coordenada de columna S_x . Un segundo multiplexor 152 se conecta, por consiguiente, al segundo operador de desplazamiento izquierdo 112 y al segundo sumador 122 y selecciona entre h y $2h$ según el bit más significativo de una coordenada de fila S_y . Las coordenadas de columna y fila son las coordenadas del téxel izquierdo superior en el parche bilineal 2×2 , a saber, téxel 30 en las Figuras 1A a 1D.

35

40

Un primer operador OR 160 lleva a cabo una operación OR en el bit más significativo y el bit menos significativo de la coordenada de columna S_x . Un segundo operador OR 162 lleva a cabo una operación OR en el bit más significativo y el bit menos significativo de la coordenada de fila S_y . El primer operador OR 160 y el primer multiplexor 150 se conectan a un primer operador AND 170 que lleva a cabo una operación AND bit a bit en la salida de multiplexor con el único bit del primer operador OR 160. La salida del primer operador AND 170 es, por lo tanto, $2g, g$ o 0 . Un segundo operador AND 172 se conecta al segundo multiplexor 152 y al segundo operador OR 162. El segundo operador AND 172 lleva a cabo una operación AND bit a bit en la salida de multiplexor con el bit del segundo operador OR 162 para obtener una salida de $2h, h$ o 0 .

45

50

Un tercer sumador 124 se conecta al primer operador AND 170 y suma la salida de sumador a una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por la primera palabra de código de color. Dicha secuencia de bits corresponde, como se ha descrito previamente, a $f=4X_0+2$. La salida de sumador es, por lo tanto, $2g+f, g+f$ o f . Un cuarto sumador 126 se conecta al primer sumador 120 y al tercer sumador 124 y se configura para añadir las salidas de dichos sumadores 120, 124 para obtener $3g+f, 2g+f$ o $g+f$. Un quinto sumador 128 se conecta al segundo sumador 122 y al segundo operador AND 172 y suma las salidas de dichas dos unidades 122, 172 para obtener $3h, 2h$ o h .

55

60

Un sexto sumador 121 se conecta al tercer sumador 124 y al segundo operador AND 172 y se configura para añadir las salidas de dichas unidades 124, 172. Un primer operador de desplazamiento derecho 130 se conecta al sexto sumador 121 y se configura para desplazar la secuencia de bits de la salida de sumador dos posiciones de bit a la derecha para obtener un valor de téxel del téxel 30 en la Figura 1A.

65

Un séptimo sumador 123 se conecta al cuarto sumador 126 y al segundo operador AND 172 y se configura para añadir las salidas de dichas unidades 126, 172. Un segundo operador de desplazamiento derecho 132 se conecta al séptimo sumador 123 y se configura para desplazar la secuencia de bits de la salida de sumador dos posiciones de bit de a la derecha para obtener el valor de téxel del téxel 32 en la Figura 1A.

Un octavo sumador 125 se conecta al tercer sumador 124 y el quinto sumador 128 y se configura para añadir las salidas de dichos sumadores 124, 128. Un tercer operador de desplazamiento derecho 134 se conecta al octavo sumador 125 y se configura para desplazar la secuencia de bits de la salida de sumador dos posiciones de bit de a la derecha para obtener el valor de t xel del t xel 34 en la Figura 1A.

5 Un noveno sumador 127 se conecta al cuarto sumador 126 y al quinto sumador 128 y se configura para a adir las salidas de dichos sumadores 126, 128. Un cuarto operador de desplazamiento derecho 136 se conecta al cuarto sumador 127 y se configura para desplazar la secuencia de bits de la salida de sumador dos posiciones de bit a la derecha para obtener el valor de t xel del t xel 36 en la Figura 1A.

10 En una implementaci n opcional pero preferida, las palabras de c digo de color de entrada X_0 , X_H , X_V se extienden o expanden primero de seis o siete bits a ocho bits. Como consecuencia, un primer extensor de bits 180 se dispone preferiblemente conectado al operador de negaci n 140 y se configura adem s para extender o expandir la primera palabra de c digo de color X_0 a la secuencia de bits predefinida, preferiblemente de ocho bits, antes de a adir 10_{bin} como los bits menos significativos para obtener una secuencia de 10 bits. Un segundo extensor 182 se dispone, por consiguiente, conectado al primer sumador 120 y se configura para extender la segunda palabra de c digo de color X_H . Finalmente, un tercer extensor 184 se conecta al segundo sumador 122 y se configura para extender la tercera palabra de c digo de color X_V .

15 20 En una implementaci n opcional, cada operador de desplazamiento derecho 130-136 se conecta preferiblemente a un fijador u operador de fijaci n 190-196 respectivo. Dichos fijadores 190-196 se configuran para fijar la salida de los operadores de desplazamiento derecho 130-136 entre un valor de t xel m nimo y un valor de t xel m ximo. Por consiguiente, los fijadores 190-196 aseguran que la secuencia de bits de salida del primer descodificador 100 se encuentra dentro de un intervalo de valor permitido. Un ejemplo de dicho intervalo de valor permitido va de 0 a 255 si las secuencias de salida del primer descodificador 100 son de 8 bits.

25 El n mero de puertas para el ejemplo de implementaci n de la Figura 8, excluidos los fijadores opcionales 190-196, es:

30 R/G/B: 2 sumadores de 8 bits $\rightarrow 16 \times 4,4 = 70,4$ puertas
 1 sumador de 10 bits $\rightarrow 10 \times 4,4 = 44$ puertas
 1 sumador de 11 bits $\rightarrow 11 \times 4,4 = 48,4$ puertas
 5 sumadores de 12 bits $\rightarrow 60 \times 4,4 = 264$ puertas

35 2 mux de 10 bits $\rightarrow 20 \times 2,2 = 44$ puertas

2 OR de 1 bit $\rightarrow 2 \times 1,5 = 3$ puertas
 2 AND de 10 bits $\rightarrow 20 \times 1,5 = 30$ puertas
 1 NOT de 8 bits $\rightarrow 8 \times 0,5 = 4$ puertas

40 Total: $3 \times 507,8 = 1523,4$ puertas

45 Los cuatro descodificadores del sistema de descodificaci n pueden, por lo tanto, implementarse con solo $0 + 250,8 + 602,7 + 1523,4 = 2376,9$ puertas, excluidos los fijadores.

50 Una implementaci n tradicional con cuatro descodificadores id nticos, como se sugiere en la t cnica anterior [3], ser a muy similar al primer descodificador en la Figura 9. La  nica diferencia ser a que se necesitar a solo un t xel en la salida en lugar de cuatro como en la figura. Por lo tanto, los cuatro sumadores 121, 123, 125, 127 se reemplazar an entonces por dos multiplexores de cuatro direcciones seguidos por un sumador. Ello costar a 6 sumadores, 4 multiplexores y dos operadores AND por canal de color. Ello significa que con cuatro descodificadores y tres componentes de color diferentes, el costo de hardware o la complejidad seg n la t cnica anterior hubieran sido:

55 $6 \times 3 \times 4 = 72$ sumadores
 $4 \times 3 \times 4 = 48$ multiplexores
 $2 \times 3 \times 4 = 24$ operadores AND

Ello se compara con las realizaciones de implementaci n descritas de las Figuras 6-8 que necesitan

60 $(9 + 3 + 2 + 0) \times 3 = 42$ sumadores
 $(2 + 3 + 0 + 0) \times 3 = 15$ multiplexores
 $(2 + 1 + 0 + 0) \times 3 = 9$ operadores AND

65 El n mero de sumadores se ha reducido, por lo tanto, de 72 a 42, o en un 42%. Los multiplexores se han reducido incluso m s en t rminos relativos, de 48 a 15, o en un 69%.

La Figura 13 ilustra una realización de una implementación de hardware del selector de valor 500. En la presente realización, el selector de valor 500 comprende cuatro multiplexores 510, cada uno de los cuales produce un valor de t́xel de un t́xel 30-36 del parche bilineal. El primer multiplexor 510 se conecta a las cuatro salidas del primer descodificador 100 y selecciona una de las cuatro secuencias de bits como el valor de t́xel del primer t́xel 30 en el parche bilineal. El segundo multiplexor 520 se conecta a la segunda y cuarta salidas del primer descodificador 100 y a las dos salidas del segundo descodificador 200. Dicho segundo multiplexor 520 produce el valor de t́xel del segundo t́xel 32 en el parche bilineal. Un tercer multiplexor 530 se conecta a la tercera y cuarta salidas del primer descodificador 100, a la primera salida del segundo descodificador 200 y a la única salida del tercer descodificador 300. El tercer multiplexor 530 produce el valor de t́xel para el tercer t́xel 34 en el parche bilineal. Finalmente, el cuarto multiplexor 540 se conecta a la cuarta salida del primer descodificador 100, a la segunda salida del segundo descodificador 200 y a la única salida del cuarto descodificador 400. El cuarto multiplexor 540 selecciona y produce el valor de t́xel del cuarto t́xel 36 en el parche bilineal.

La posición del parche bilineal respecto de un borde del bloque t́xel dicta qú secuencias de bits se producen desde los multiplexores 510-540. Por consiguiente, en la situación ilustrada en la Figura 1A, las secuencias de bits de las cuatro salidas del primer descodificador 100 se seleccionan por los multiplexores 510-540. En mayor detalle, el primer multiplexor 510 produce la secuencia de bits desde la primera salida del primer descodificador 100, el segundo multiplexor 520 produce la secuencia de bits desde la segunda salida con el tercer 530 y cuarto 540 multiplexores que producen las secuencias de bits desde la tercera y cuarta salidas de descodificador, respectivamente. Un parche bilineal ubicado como se ilustra en la Figura 1B supone que el primer multiplexor 510 selecciona la secuencia de bits de la segunda salida del primer descodificador 100. El segundo multiplexor 520 selecciona la secuencia de bits de la primera salida del segundo descodificador 200. El tercer descodificador 300 selecciona la secuencia de bits de la cuarta salida del primer descodificador 100 y el cuarto descodificador 400 selecciona la secuencia de bits de la segunda salida del segundo descodificador 200. El concepto puede entonces aplicarse de manera similar a las otras dos posiciones de parche ilustradas en las Figuras 1C y 1D.

El costo en t́rminos de ńmero de puertas de implementar la realización del selector de valor 500 como se ilustra en la Figura 13 es:

$$3 \text{ mux de 4 direcciones de 8 bits} \rightarrow 24 \times 6,6 = 158,4 \text{ puertas}$$

$$1 \text{ mux de 3 direcciones de 8 bits} \rightarrow 8 \times 4,4 = 35,2 \text{ puertas}$$

Total: $158,4 + 35,2 = 193,6$ puertas

El costo total para los descodificadores y el selector de valor, incluidos los fijadores, es de $2376,9 + 6 \times 18 + 193,6 = 2678,5$ puertas.

Si los fijadores del primer y segundo descodificadores como se ilustran en las Figuras 8 y 9 se hubieran omitido y, en su lugar, se hubieran reemplazado por cuatro fijadores, cada uno conectado a uno de los multiplexores 510-540 del selector de valor 500, el costo de implementación del selector de valor 500 aumentaría dado que entonces se necesitarían los tres multiplexores de 4 direcciones de 11 bits y un multiplexor de 3 direcciones de 11 bits y ello resultaría en un costo de 266,2 puertas en lugar de 193,6 puertas.

El calculador de píxeles del sistema de descodificación se puede implementar en hardware según las técnicas conocidas en la técnica. En mayor detalle, supongamos que los cuatro valores de t́xel, en el caso del filtrado bilineal, del selector de valor se denotan A, B, C y D. El calculador de píxeles entonces calcula el valor de píxel como:

$$\text{valor de píxel} = A(1 - dx)(1 - dy) + Bdx(1 - dy) + C(1 - dx)dy + Ddxdy$$

donde dx es la distancia a lo largo del eje x entre el punto medio del t́xel A y el punto de muestra para el píxel y dy es la distancia correspondiente a lo largo del eje y entre el punto medio del t́xel A y el punto de muestra para el píxel.

La ecuación descrita más arriba se lleva a cabo en paralelo para los componentes de color rojo, verde y azul. Una implementación de hardware se puede entonces definir según la ecuación. Por ejemplo, supongamos que la posición del punto de muestra se puede expresar con tres decimales binarios, p.ej. la posición dx puede asumir los siguientes valores [0,000, 0,125, 0,250, 0,375, 0,500, 0,625, 0,750, 0,875]. En dicho caso, la posición se puede definir en ochos de modo que los valores de más arriba se pueden representar por [0, 1, 2, 3, 4, 5, 6, 7]. La ecuación de más arriba puede entonces reescribirse como:

$$\text{valor de píxel} = A(8 - DX)(8 - DY) + BDX(8 - DY) + C(8 - DX)DY + DDXDY \gg 6$$

donde $DX = 8 \times dx$ y $DY = 8 \times dy$ y DX, DY son íntegros entre 0 y 7.

Al investigar el segundo término $8DX(8 - DY)$ nos damos cuenta de que tanto DX como $(8-DY)$ son íntegros entre 0 y 8. Con el fin de representar ocho, se requieren cuatro bits. Se puede usar un operador de multiplicación que maneja cuatro bits en cada entrada y produce una secuencia de 8 bits. Sin embargo, el mayor número que el operador de multiplicación puede producir es $7 \times 8 = 56$ de modo que solamente los seis bits menos significativos se necesitan de la salida.

La salida del operador de multiplicación se debe luego multiplicar por el valor de t́xel B, preferiblemente un número entre 0 y 255. Dicha multiplicación se puede manejar por un operador de multiplicación que toma dos entradas de 8 bits y produce una secuencia de bits de 16 bits. Sin embargo, se necesitan solo los catorce bits menos significativos del operador de multiplicación dado que el mayor número posible es $56 \times 255 < 2^{14}$.

Una implementación correspondiente se lleva a cabo luego para los otros términos en la ecuación y las salidas de los respectivos operadores de multiplicación final se añaden y desplazan seis pasos a la derecha con el fin de obtener el valor de píxel final de 8 bits.

Los ejemplos de implementación descritos más arriba se deben ver simplemente como realizaciones preferidas pero no restrictivas de cómo las unidades del sistema de descodificación se pueden implementar en hardware. Diferentes variantes de aquel son posibles y dentro del alcance de la invención. La característica importante es, sin embargo, que los circuitos de los descodificadores incluidos son diferentes con la creciente complejidad y que cada descodificador se configura para manejar y producir el valor de t́xel de uno o más t́xeles en posiciones predefinidas en el parche.

La Figura 14 es un diagrama de flujo que ilustra un método para generar un píxel descodificado descodificando al menos un bloque t́xel codificado. El método comienza en la etapa E1 donde se determina la posición de al menos N valores de t́xel, a saber, un parche, respecto a un borde de bloque t́xel. Un bloque t́xel codificado respectivo, ya sea el mismo bloque t́xel codificado o bloques t́xel codificados diferentes que dependen de la posición de parche relativa determinada, se ingresa en N descodificadores para obtener valores de t́xel descodificados. Una siguiente etapa E2 produce, de manera selectiva, al menos N valores de t́xel desde al menos uno de los N descodificadores. Los al menos N valores de t́xel particulares para seleccionar en la etapa E2 y desde cuyos descodificadores se traen se determinan según la posición de parche relativa obtenida en la etapa E1. Además, cada uno de los N descodificadores tiene un circuito respectivo que es diferente de los circuitos de los otros $N - 1$ descodificadores. La etapa final E3 calcula un valor de píxel descodificado según al menos N valores de t́xel seleccionados en la etapa E2, preferiblemente como una combinación lineal de dichos al menos N valores de t́xel. El método entonces finaliza o comienza de nuevo en la etapa E1 con el fin de generar un nuevo píxel descodificado.

La etapa E2 incluye preferiblemente producir, de manera selectiva, todos los N valores de t́xel desde un primer descodificador de los N descodificadores si los N t́xeles se posicionan dentro del límite de un bloque t́xel. Sin embargo, si la mitad de los N t́xeles se posiciona dentro de un bloque t́xel y la mitad restante de los t́xeles se encuentra, en cambio, dentro de un bloque t́xel vecino, la mitad de los N valores de t́xel se selecciona preferiblemente de la salida del primer descodificador con la mitad restante de un segundo descodificador.

En una realización preferida, $N = 4$. Si los cuatro t́xeles se posicionan entonces de modo que cada t́xel se encuentra en un bloque t́xel separado, la etapa E2 incluye, preferiblemente, seleccionar un valor de t́xel respectivo de cada uno de los cuatro descodificadores.

Las realizaciones descritas más arriba se deben comprender como unos pocos ejemplos ilustrativos de la presente invención.

El alcance de la presente invención se define por las reivindicaciones anexas.

REFERENCIAS

[1] Patente de Estados Unidos No. 5,956,431

[2] Ström y Möller, "*iPACKMAN: High-Quality, Low-Complexity Texture Compression for Mobile Phones*", *Graphics Hardware*, 2005, ACM Press, pp. 63-70

[3] Ström y Pettersson, "*ETC2: Texture Compression using Invalid Combinations*", *Graphics Hardware*, 2007, ACM Press, pp. 49-54

[4] Solicitud internacional No. WO 2006/126949

REIVINDICACIONES

- 5 1. Un sistema de descodificación (1) para generar un píxel descodificado basado en al menos N téxeles (30-36) de al menos un bloque téxel codificado (40), N siendo un número más grande que 1, dicho sistema de descodificación (1) comprende:
- 10 N descodificadores diferentes (100, 200, 300, 400), cada uno de los cuales tiene un circuito que tiene un nivel de complejidad que es diferente del nivel de complejidad de los circuitos de los otros $N - 1$ descodificadores y cada uno de los cuales se configura para descodificar un bloque téxel codificado (40) para generar al menos un valor de téxel respectivo;
- 15 un selector de valor (500) conectado a dichos N descodificadores (100, 200, 300, 400) y configurado para producir, de manera selectiva, los al menos N valores de téxel asociados a dichos al menos N téxeles (30-36) de al menos uno de dichos N descodificadores (100, 200, 300, 400) según las posiciones de los al menos N valores de téxeles (30-36) respecto a un límite de un bloque téxel (10) que comprende al menos uno de dichos al menos N téxeles (30-36); y
- 20 un calculador de píxeles (600) configurado para calcular un valor de píxel descodificado de dicho píxel descodificado según dichos al menos N valores de téxel producidos, de forma selectiva, por dicho selector de valor (500).
- 25 2. El sistema de descodificación según la reivindicación 1, en donde dicho selector de valor (500) se configura para producir, de forma selectiva, N valores de téxel de un primer descodificador (100) de dichos N descodificadores (100, 200, 300, 400) si los N téxeles (30-36) asociados a dichos N valores de téxel se posicionan dentro de dicho límite de un bloque téxel (10).
- 30 3. El sistema de descodificación según la reivindicación 1 o 2, en donde dicho selector de valor (500) se configura para producir, de forma selectiva, una mitad de N valores de téxel de un primer descodificador (100) de dichos N descodificadores (100, 200, 300, 400) y una mitad restante de dichos N valores de téxel de un segundo descodificador (200) de dichos N descodificadores (100, 200, 300, 400) si la mitad de los N téxeles (30-36) asociados a dichos N valores de téxel se posicionan dentro de dicho límite de dicho bloque téxel (10) y la mitad de dichos N téxeles (30-36) se posiciona dentro del límite de un bloque téxel vecino (12, 14).
- 35 4. El sistema de descodificación según cualquiera de las reivindicaciones 1 a 3, en donde N es igual a cuatro y dicho selector de valor (500) se configura para producir, de forma selectiva, un primer valor de téxel de dichos cuatro valores de téxel de un primer descodificador (100) de dichos cuatro descodificadores (100, 200, 300, 400), un segundo valor de téxel de dichos cuatro valores de téxel de un segundo descodificador (200) de dichos cuatro descodificadores (100, 200, 300, 400), un tercer valor de téxel de dichos cuatro valores de téxel de un tercer descodificador (300) de dichos cuatro descodificadores (100, 200, 300, 400) y un cuarto valor de téxel de dichos cuatro valores de téxel de un cuarto descodificador (400) de dichos cuatro descodificadores (100, 200, 300, 400) si uno de dichos cuatro téxeles (30-36) se posiciona dentro de dicho límite de dicho bloque téxel (10) y cada uno de los tres téxeles restantes se posiciona dentro del límite de un bloque téxel vecino (12, 14, 16) respectivo.
- 40 5. El sistema de descodificación según cualquiera de las reivindicaciones 1 a 4, en donde N es igual a cuatro y un primer descodificador (100) de dichos cuatro descodificadores (100, 200, 300, 400) tiene un primer circuito y se configura para descodificar un bloque téxel codificado (40) para generar cuatro valores de téxel;
- 45 un segundo descodificador (200) de dichos cuatro descodificadores (100, 200, 300, 400) tiene un segundo circuito y se configura para descodificar un bloque téxel codificado (40) para generar dos valores de téxel;
- 50 un tercer descodificador (300) de dichos cuatro descodificadores (100, 200, 300, 400) tiene un tercer circuito y se configura para descodificar un bloque téxel codificado (40) para generar un valor de téxel; y
- un cuarto descodificador (400) de dichos cuatro descodificadores (100, 200, 300, 400) tiene un cuarto circuito y se configura para descodificar un bloque téxel codificado (40) para generar un valor de téxel.
- 55 6. El sistema de descodificación según cualquiera de las reivindicaciones 1 a 5, en donde dicho bloque téxel codificado (40) comprende una primera palabra de código de color (41, 44, 47), una segunda palabra de código de color (42, 45, 48) y una tercera palabra de código de color (43, 46, 49).
- 60 7. El sistema de descodificación según las reivindicaciones 5 y 6, en donde dicho cuarto descodificador (400) comprende al menos un cable para producir dicha primera palabra de código de color (41, 44, 47).
8. El sistema de descodificación según la reivindicación 6 o 7, en donde dicho tercer descodificador (300) comprende:
- un operador de desplazamiento izquierdo (310) configurado para desplazar una secuencia de bits correspondiente a dicha segunda palabra de código de color (42, 45, 48) una posición de bit a la izquierda para formar una secuencia de bits desplazada a la izquierda;

un primer sumador (320) que tiene arrastre establecido en uno y que se conecta a dicho operador de desplazamiento izquierdo (310) y se configura para sumar dicha segunda palabra de código de color (42, 45, 48) y dicha secuencia de bits desplazada a la izquierda;

5 un segundo sumador (322) que tiene arrastre establecido en uno y que se conecta a dicho primer sumador (320) y se configura para sumar la salida de dicho primer sumador (320) y dicha primera palabra de código de color (41, 44, 47); y

un operador de desplazamiento derecho (330) conectado a dicho segundo sumador (322) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho segundo sumador (322) dos posiciones de bit a la derecha para formar una secuencia desplazada a la derecha.

10 9. El sistema de descodificación según cualquiera de las reivindicaciones 6 a 8, en donde dicho segundo descodificador (200) comprende:

15 un primer multiplexor (250) configurado para producir una de dicha segunda palabra de código de color (42, 45, 48) y dicha tercera palabra de código de color (43, 46, 49) según si la mitad de los cuatro téxeles (30-36) se posiciona en una fila o columna dentro de dicho límite de dicho bloque téxel (10);

un operador de negación (240) configurado para negar dicha primera palabra de código de color (41, 44, 47);

20 un primer sumador (220) conectado a dicho primer multiplexor (250) y a dicho operador de negación (240) y configurado para sumar la salida de dicho primer multiplexor (250) y dicha primera palabra de código de color negada;

un operador de desplazamiento izquierdo (210) conectado a dicho primer sumador (220) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho primer sumador (220) una posición de bit a la izquierda para formar una secuencia de bits desplazada a la izquierda;

25 un segundo multiplexor (252) configurado para producir una de una coordenada de fila y una coordenada de columna de un téxel (32, 34) según si dicha mitad de cuatro téxeles (30-36) se posiciona en una fila o columna dentro de dicho límite de dicho bloque téxel (10);

un tercer multiplexor (254) conectado a dicho operador de desplazamiento izquierdo (210) y a dicho primer sumador (220) y configurado para producir una de dicha salida de dicho primer sumador (220) y dicha secuencia de bits desplazada a la izquierda según el bit más significativo de la salida de dicho segundo multiplexor (252);

30 un operador OR (260) conectado a dicho segundo multiplexor (252) y configurado para recibir dicho bit más significativo de dicha salida de dicho segundo multiplexor (252) y el bit menos significativo de dicha salida de dicho segundo multiplexor;

un operador AND (270) conectado a dicho tercer multiplexor (254) y a dicho operador OR (260) y configurado para recibir la salida de dicho tercer multiplexor (254) y la salida de dicho operador OR (260);

35 un segundo sumador (222) conectado a dicho operador AND (270) y configurado para sumar la salida de dicho operador AND (270) y una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por dicha primera palabra de código de color (41, 44, 47);

40 un tercer sumador (224) conectado a dicho primer sumador (220) y a dicho segundo sumador (222) y configurado para sumar dicha salida de dicho primer sumador (220) y la salida de dicho segundo sumador (222);

un primer operador de desplazamiento derecho (230) conectado a dicho tercer sumador (224) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho tercer sumador (224) dos posiciones de bit a la derecha para formar una primera secuencia desplazada a la derecha;

45 un segundo operador de desplazamiento derecho (232) conectado a dicho segundo sumador (222) y configurado para desplazar la secuencia de bits correspondiente a dicha salida de dicho segundo sumador (222) dos posiciones de bit a la derecha para formar una segunda secuencia desplazada a la derecha.

50 10. El sistema de descodificación según cualquiera de las reivindicaciones 6 a 8, en donde dicho segundo descodificador (200) comprende:

un primer multiplexor (250) configurado para producir una de dicha segunda palabra de código de color (42, 45, 48) y dicha tercera palabra de código de color (43, 46, 49) según si la mitad de cuatro téxeles (30-36) se posiciona en una fila o columna dentro de dicho límite de dicho bloque téxel (10);

55 un operador not configurado para negar dicha primera palabra de código de color (41, 44, 47);

un primer sumador (220) que tiene arrastre establecido en uno conectado a dicho primer multiplexor (250) y a dicho operador not y configurado para sumar la salida de dicho primer multiplexor (250) y dicha primera palabra de código de color negada;

60 un operador de desplazamiento izquierdo (210) conectado a dicho primer sumador (220) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho primer sumador (220) una posición de bit a la izquierda para formar una secuencia de bits desplazada a la izquierda;

un segundo multiplexor (252) configurado para producir una de una coordenada de fila y una coordenada de columna de un téxel (32, 34) según si dicha mitad de cuatro téxeles (30-36) se posiciona en una fila o columna dentro de dicho límite de dicho bloque téxel (10);

65 un tercer multiplexor (254) conectado a dicho operador de desplazamiento izquierdo (210) y a dicho primer sumador (220) y configurado para producir una de dicha salida de dicho primer sumador (220) y dicha

secuencia de bits desplazada a la izquierda según el bit más significativo de la salida de dicho segundo multiplexor (252);

un operador OR (260) conectado a dicho segundo multiplexor (252) y configurado para recibir dicho bit más significativo de dicha salida de dicho segundo multiplexor (252) y el bit menos significativo de dicha salida de dicho segundo multiplexor;

un operador AND (270) conectado a dicho tercer multiplexor (254) y a dicho operador OR (260) y configurado para recibir la salida de dicho tercer multiplexor (254) y la salida de dicho operador OR (260);

un segundo sumador (222) conectado a dicho operador AND (270) y configurado para sumar la salida de dicho operador AND (270) y una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por dicha primera palabra de código de color (41, 44, 47);

un tercer sumador (224) conectado a dicho primer sumador (220) y a dicho segundo sumador (222) y configurado para sumar dicha salida de dicho primer sumador (220) y la salida de dicho segundo sumador (222);

un primer operador de desplazamiento derecho (230) conectado a dicho tercer sumador (224) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho tercer sumador (224) dos posiciones de bit a la derecha para formar una primera secuencia desplazada a la derecha;

un segundo operador de desplazamiento derecho (232) conectado a dicho segundo sumador (222) y configurado para desplazar la secuencia de bits correspondiente a dicha salida de dicho segundo sumador (222) dos posiciones de bit a la derecha para formar una segunda secuencia desplazada a la derecha.

11. El sistema de descodificación según cualquiera de las reivindicaciones 6 a 10, en donde dicho primer descodificador (100) comprende:

un operador de negación (140) configurado para negar dicha primera palabra de código de color (41, 44, 47); un primer sumador (120) conectado a dicho operador de negación (140) y configurado para sumar dicha segunda palabra de código de color (42, 45, 48) y dicha primera palabra de código de color negada;

un segundo sumador (122) conectado a dicho operador de negación (140) y configurado para sumar dicha tercera palabra de código de color (43, 46, 49) y dicha primera palabra de código de color negada;

un primer operador de desplazamiento izquierdo (110) conectado a dicho primer sumador (120) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho primer sumador (120) una posición de bit a la izquierda para formar una primera secuencia de bits desplazada a la izquierda;

un segundo operador de desplazamiento izquierdo (112) conectado a dicho segundo sumador (122) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho segundo sumador (122) una posición de bit a la izquierda para formar una segunda secuencia de bits desplazada a la izquierda;

un primer multiplexor (150) conectado a dicho primer sumador (120) y a dicho primer operador de desplazamiento izquierdo (110) y configurado para producir una de dicha salida de dicho primer sumador (120) y dicha primera secuencia de bits desplazada a la izquierda según el bit más significativo de una coordenada de columna de un téxel (30);

un segundo multiplexor (152) conectado a dicho segundo sumador (122) y a dicho segundo operador de desplazamiento izquierdo (112) y configurado para producir una de dicha salida de dicho segundo sumador (122) y dicha segunda secuencia de bits desplazada a la izquierda según el bit más significativo de una coordenada de fila de dicho téxel (30);

un primer operador OR (160) configurado para recibir dicho bit más significativo de dicha coordenada de columna y dicho bit menos significativo de dicha coordenada de columna;

un segundo operador OR (162) configurado para recibir dicho bit más significativo de dicha coordenada de fila y dicho bit menos significativo de dicha coordenada de fila;

un primer operador AND (170) conectado a dicho primer multiplexor (150) y a dicho primer operador OR (160) y configurado para recibir la salida de dicho primer multiplexor (150) y la salida de dicho primer operador OR (160);

un segundo operador AND (172) conectado a dicho segundo multiplexor (152) y a dicho segundo operador OR (162) y configurado para recibir la salida de dicho segundo multiplexor (152) y la salida de dicho segundo operador OR (162);

un tercer sumador (124) conectado a dicho primer operador AND (170) y configurado para sumar la salida de dicho primer operador AND (170) y una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por dicha primera palabra de código de color (41, 44, 47);

un cuarto sumador (126) conectado a dicho primer sumador (120) y a dicho tercer sumador (124) y configurado para sumar dicha salida de dicho primer sumador (120) y la salida de dicho tercer sumador (124);

un quinto sumador (128) conectado a dicho segundo sumador (122) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho segundo sumador (122) y la salida de dicho segundo operador AND (172);

un sexto sumador (121) conectado a dicho tercer sumador (124) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho tercer sumador (124) y dicha salida de dicho segundo operador AND (172);

un séptimo sumador (123) conectado a dicho cuarto sumador (126) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho cuarto sumador (126) y dicha salida de dicho segundo operador AND (172);

5 un octavo sumador (125) conectado a dicho tercer sumador (124) y a dicho quinto sumador (128) y configurado para sumar dicha salida de dicho tercer sumador (124) y la salida de dicho quinto sumador (128); un noveno sumador (127) conectado a dicho cuarto sumador (126) y a dicho quinto sumador (128) y configurado para sumar dicha salida de dicho cuarto sumador (126) y dicha salida de dicho quinto sumador (128);

10 un primer operador de desplazamiento derecho (130) conectado a dicho sexto sumador (121) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho sexto sumador (121) dos posiciones de bit a la derecha para formar una primera secuencia desplazada a la derecha;

un segundo operador de desplazamiento derecho (132) conectado a dicho séptimo sumador (123) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho séptimo sumador (123) dos posiciones de bit a la derecha para formar una segunda secuencia desplazada a la derecha;

15 un tercer operador de desplazamiento derecho (134) conectado a dicho octavo sumador (125) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho octavo sumador (125) dos posiciones de bit a la derecha para formar una tercera secuencia desplazada a la derecha; y

un cuarto operador de desplazamiento derecho (136) conectado a dicho noveno sumador (127) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho noveno sumador (127) dos posiciones de bit a la derecha para formar una cuarta secuencia desplazada a la derecha.

20 12. El sistema de descodificación según cualquiera de las reivindicaciones 6 a 10, en donde dicho primer descodificador (100) comprende:

un operador not configurado para negar dicha primera palabra de código de color (41, 44, 47);

25 un primer sumador (120) que tiene arrastre establecido en uno conectado a dicho operador not y configurado para sumar dicha segunda palabra de código de color (42, 45, 48) y dicha primera palabra de código de color negada;

un segundo sumador (122) que tiene arrastre establecido en uno conectado a dicho operador not y configurado para sumar dicha tercera palabra de código de color (43, 46, 49) y dicha primera palabra de código de color negada;

30 un primer operador de desplazamiento izquierdo (110) conectado a dicho primer sumador (120) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho primer sumador (120) una posición de bit a la izquierda para formar una primera secuencia de bits desplazada a la izquierda;

un segundo operador de desplazamiento izquierdo (112) conectado a dicho segundo sumador (122) y configurado para desplazar una secuencia de bits correspondiente a la salida de dicho segundo sumador (122) una posición de bit a la izquierda para formar una segunda secuencia de bits desplazada a la izquierda;

35 un primer multiplexor (150) conectado a dicho primer sumador (120) y a dicho primer operador de desplazamiento izquierdo (110) y configurado para producir una de dicha salida de dicho primer sumador (120) y dicha primera secuencia de bits desplazada a la izquierda según el bit más significativo de una coordenada de columna de un téxel (30);

40 un segundo multiplexor (152) conectado a dicho segundo sumador (122) y a dicho segundo operador de desplazamiento izquierdo (112) y configurado para producir una de dicha salida de dicho segundo sumador (122) y dicha segunda secuencia de bits desplazada a la izquierda según el bit más significativo de una coordenada de fila de dicho téxel (30);

un primer operador OR (160) configurado para recibir dicho bit más significativo de dicha coordenada de columna y dicho bit menos significativo de dicha coordenada de columna;

45 un segundo operador OR (162) configurado para recibir dicho bit más significativo de dicha coordenada de fila y dicho bit menos significativo de dicha coordenada de fila;

un primer operador AND (170) conectado a dicho primer multiplexor (150) y a dicho primer operador OR (160) y configurado para recibir la salida de dicho primer multiplexor (150) y la salida de dicho primer operador OR (160);

50 un segundo operador AND (172) conectado a dicho segundo multiplexor (152) y a dicho segundo operador OR (162) y configurado para recibir la salida de dicho segundo multiplexor (152) y la salida de dicho segundo operador OR (162);

un tercer sumador (124) conectado a dicho primer operador AND (170) y configurado para sumar la salida de dicho primer operador AND (170) y una secuencia de bits que tiene 10_{bin} como los dos bits menos significativos precedidos por dicha primera palabra de código de color (41, 44, 47);

55 un cuarto sumador (126) conectado a dicho primer sumador (120) y a dicho tercer sumador (124) y configurado para sumar dicha salida de dicho primer sumador (120) y la salida de dicho tercer sumador (124);

un quinto sumador (128) conectado a dicho segundo sumador (122) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho segundo sumador (122) y la salida de dicho segundo operador AND (172);

60 un sexto sumador (121) conectado a dicho tercer sumador (124) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho tercer sumador (124) y dicha salida de dicho segundo operador AND (172);

un séptimo sumador (123) conectado a dicho cuarto sumador (126) y a dicho segundo operador AND (172) y configurado para sumar dicha salida de dicho cuarto sumador (126) y dicha salida de dicho segundo operador AND (172);

65

- 5 un octavo sumador (125) conectado a dicho tercer sumador (124) y a dicho quinto sumador (128) y configurado para sumar dicha salida de dicho tercer sumador (124) y la salida de dicho quinto sumador (128);
un noveno sumador (127) conectado a dicho cuarto sumador (126) y a dicho quinto sumador (128) y configurado para sumar dicha salida de dicho cuarto sumador (126) y dicha salida de dicho quinto sumador (128);
un primer operador de desplazamiento derecho (130) conectado a dicho sexto sumador (121) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho sexto sumador (121) dos posiciones de bit a la derecha para formar una primera secuencia desplazada a la derecha;
- 10 un segundo operador de desplazamiento derecho (132) conectado a dicho séptimo sumador (123) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho séptimo sumador (123) dos posiciones de bit a la derecha para formar una segunda secuencia desplazada a la derecha;
- un tercer operador de desplazamiento derecho (134) conectado a dicho octavo sumador (125) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho octavo sumador (125) dos posiciones de bit a la derecha para formar una tercera secuencia desplazada a la derecha; y
- 15 un cuarto operador de desplazamiento derecho (136) conectado a dicho noveno sumador (127) y configurado para desplazar la secuencia de bits correspondiente a la salida de dicho noveno sumador (127) dos posiciones de bit a la derecha para formar una cuarta secuencia desplazada a la derecha.
- 20 13. El sistema de descodificación según cualquiera de las reivindicaciones 6 a 12, en donde dicho selector de valor (500) comprende:
- un primer multiplexor (510) conectado a dicho primer descodificador (100) y configurado para producir un primer valor de téxel;
- 25 un segundo multiplexor (520) conectado a dicho primer descodificador (100) y a dicho segundo descodificador (200) y configurado para producir un segundo valor de téxel;
- un tercer multiplexor (530) conectado a dicho primer descodificador (100), a dicho segundo descodificador (200) y a dicho tercer descodificador (300) y configurado para producir un tercer valor de téxel; y
- un cuarto multiplexor (540) conectado a dicho primer descodificador (100), a dicho segundo descodificador (200) y a dicho cuarto descodificador (400) y configurado para producir un cuarto valor de téxel.
- 30 14. Un método para generar un píxel descodificado descodificando al menos N téxeles (30-36) de al menos un bloque téxel codificado (40), siendo N un número más grande que 1, dicho método comprende:
- 35 producir, de manera selectiva, al menos N valores de téxel asociados a dichos al menos N téxeles (30-36) de al menos uno de N descodificadores diferentes (100, 200, 300, 400), cada uno de los cuales tiene un circuito que tiene un nivel de complejidad que es diferente del nivel de complejidad de los circuitos de los otros $N - 1$ descodificadores y cada uno de los cuales se configura para descodificar un bloque téxel codificado (40) para generar al menos un valor téxel respectivo, según las posiciones de los al menos N valores de téxeles (30-36) respecto a un límite de un bloque téxel (10) que comprende al menos uno de dichos al menos N téxeles (30-36); y
- 40 calcular un valor de píxel descodificado de dicho píxel descodificado según dichos al menos N valores de téxel.
- 45 15. El método según la reivindicación 14, en donde dicha producción selectiva comprende producir, de forma selectiva, N valores de téxel a partir de un primer descodificador (100) de dichos N descodificadores (100, 200, 300, 400) si los N téxeles (30-36) asociados a dichos N valores de téxel se posicionan dentro de dicho límite de un bloque téxel (10).
- 50 16. El método según la reivindicación 14 o 15, en donde dicha producción selectiva comprende producir, de forma selectiva, una mitad de N valores de téxel a partir de un primer descodificador (100) de dichos N descodificadores (100, 200, 300, 400) y una mitad restante de dichos N valores de téxel a partir de un segundo descodificador (200) de dichos N descodificadores (100, 200, 300, 400) si la mitad de los N téxeles (30-36) asociados a dichos N valores de téxel se posiciona dentro de dicho límite de dicho bloque téxel (10) y la mitad de dichos N téxeles (30-36) se posiciona dentro del límite de un bloque téxel vecino (12, 14).
- 55 17. El método según cualquiera de las reivindicaciones 14 a 16, en donde N es igual a cuatro y dicha producción selectiva comprende producir, de forma selectiva, un primer valor de téxel de dichos cuatro valores de téxel a partir de un primer descodificador (100) de dichos cuatro descodificadores (100, 200, 300, 400), un segundo valor de téxel de dichos cuatro valores de téxel a partir de un segundo descodificador (200) de dichos cuatro descodificadores (100, 200, 300, 400), un tercer valor de téxel de dichos cuatro valores de téxel a partir de un tercer descodificador (300) de dichos cuatro descodificadores (100, 200, 300, 400) y un cuarto valor de téxel de dichos cuatro valores de téxel a partir de un cuarto descodificador (400) de dichos cuatro descodificadores (100, 200, 300, 400) si uno de dichos cuatro téxeles (30-36) se posiciona dentro de dicho límite de dicho bloque téxel (10) y cada uno de los tres téxeles restantes se posiciona dentro del límite de un bloque téxel vecino (12, 14, 16) respectivo.
- 60
- 65

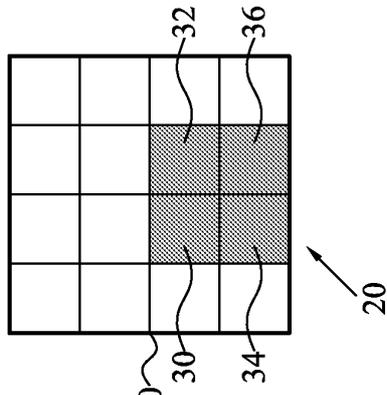


Fig. 1A

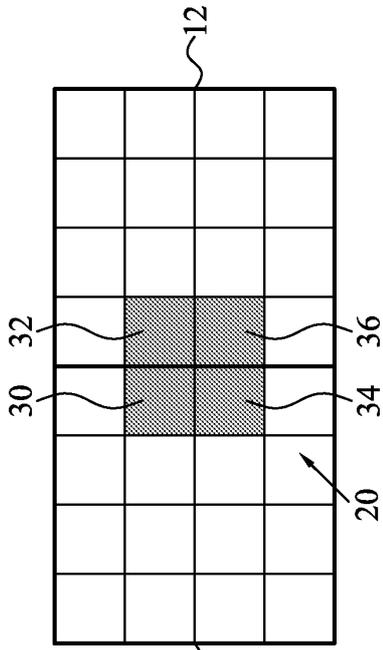


Fig. 1B

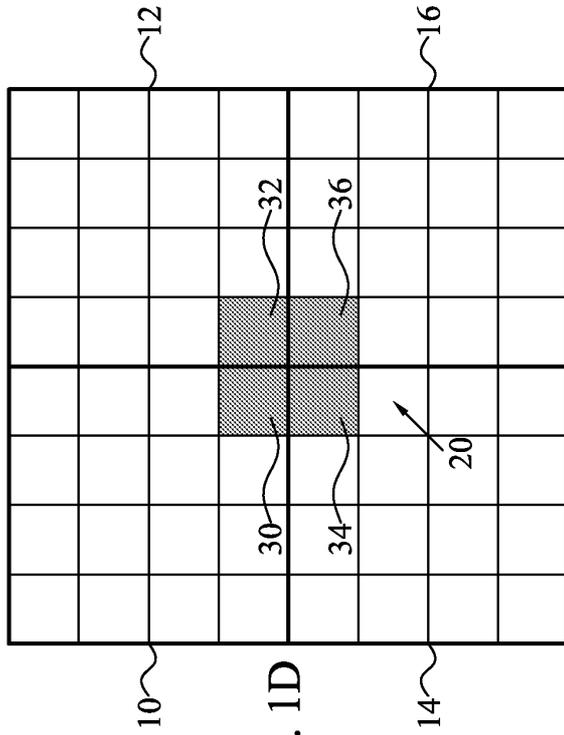


Fig. 1D

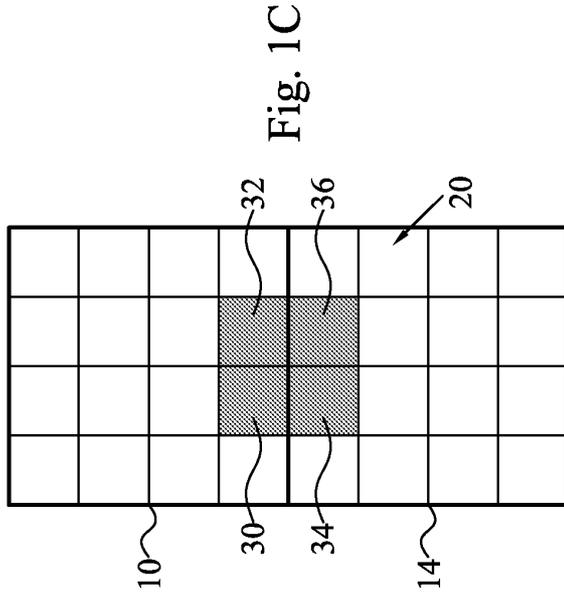


Fig. 1C

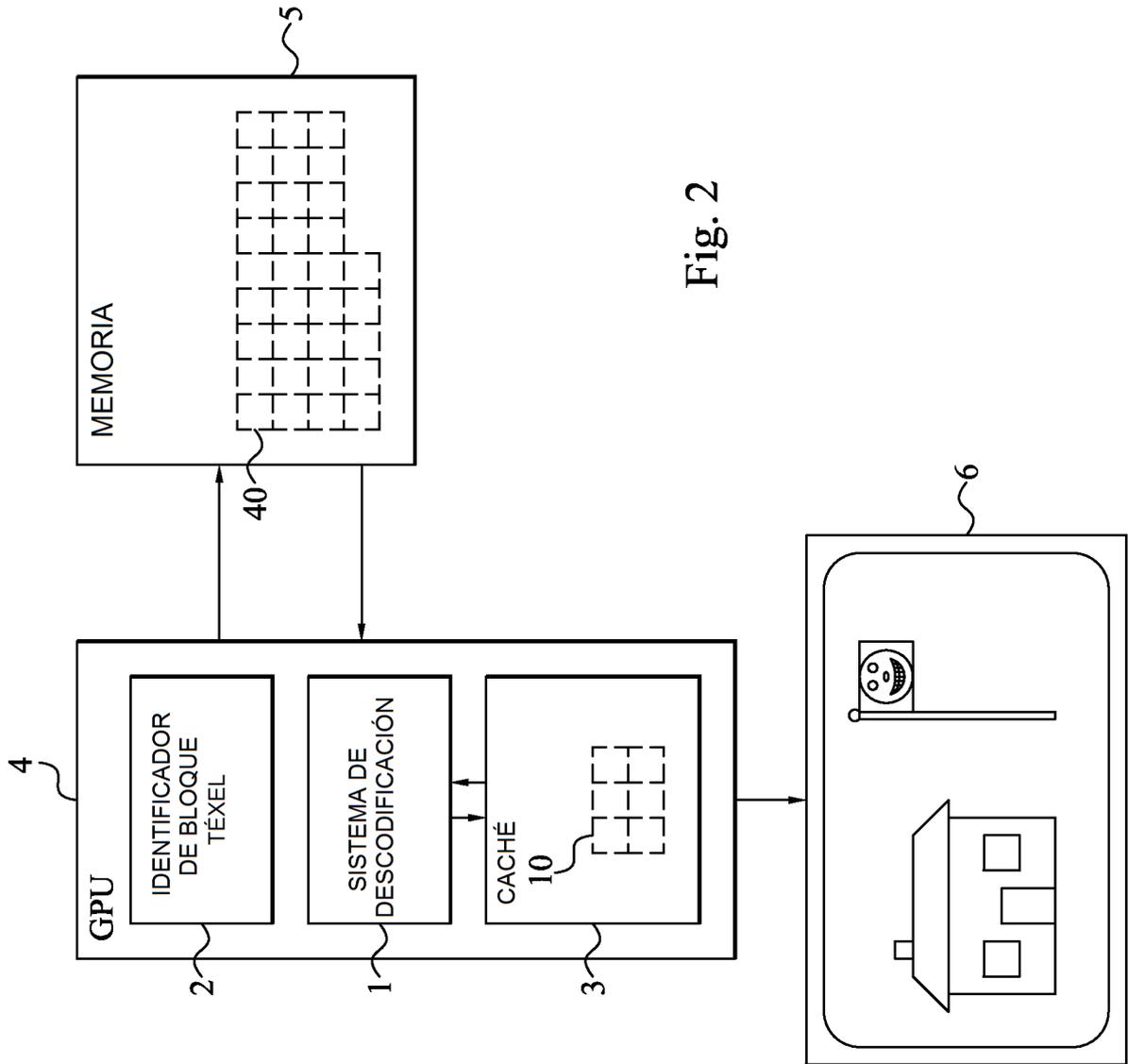


Fig. 2

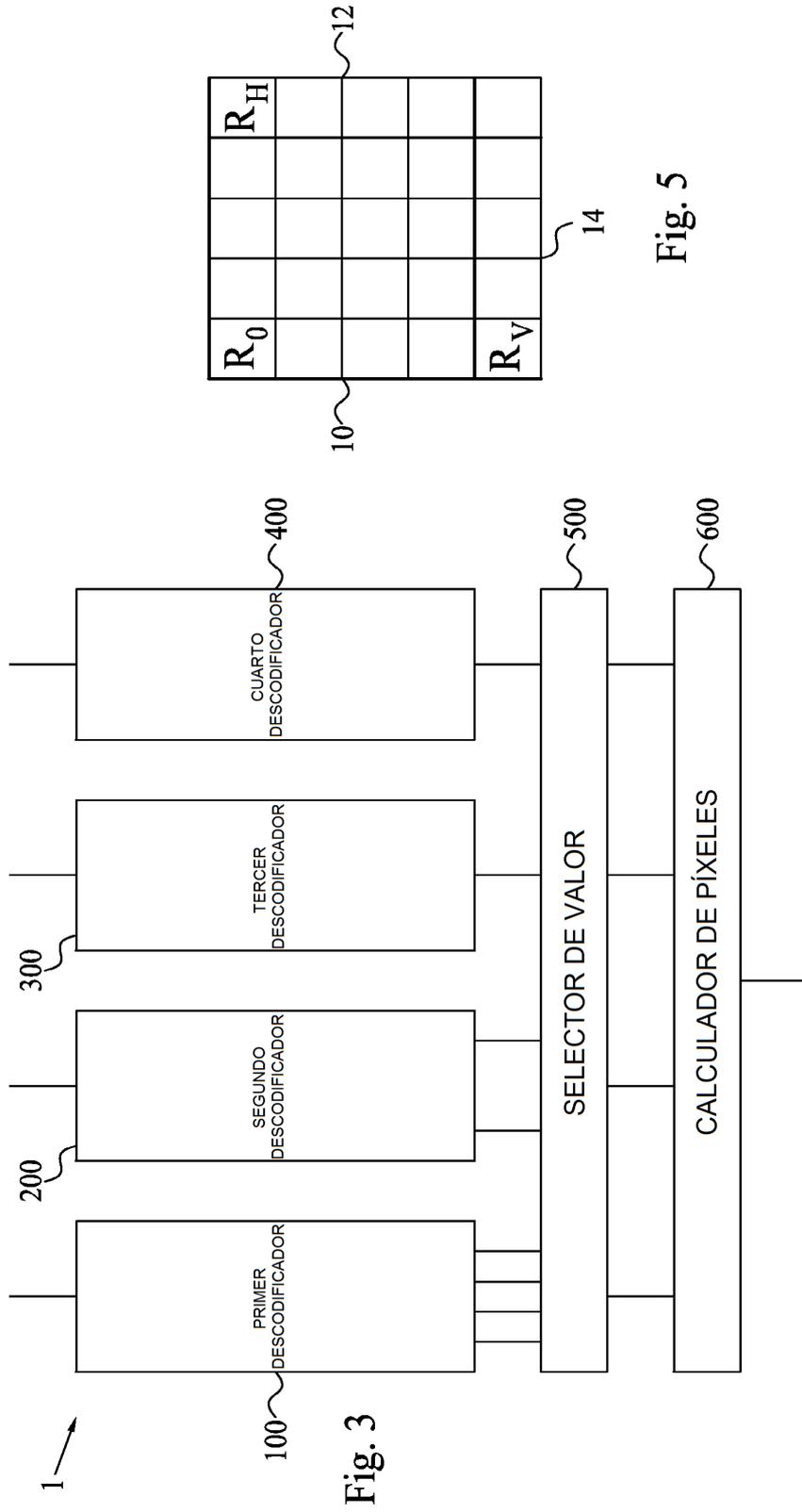


Fig. 3

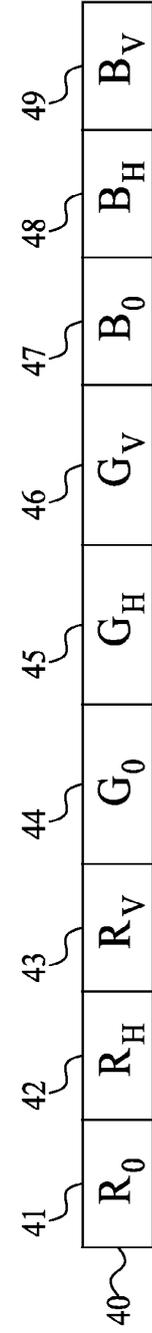


Fig. 4

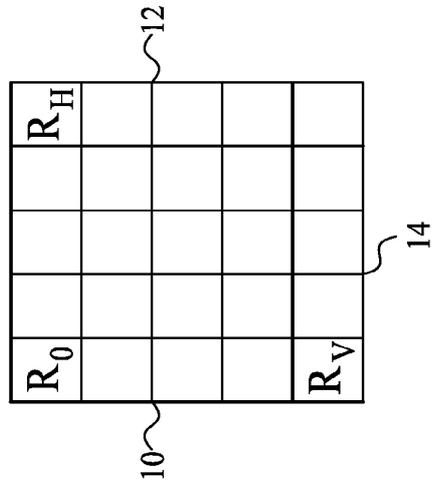
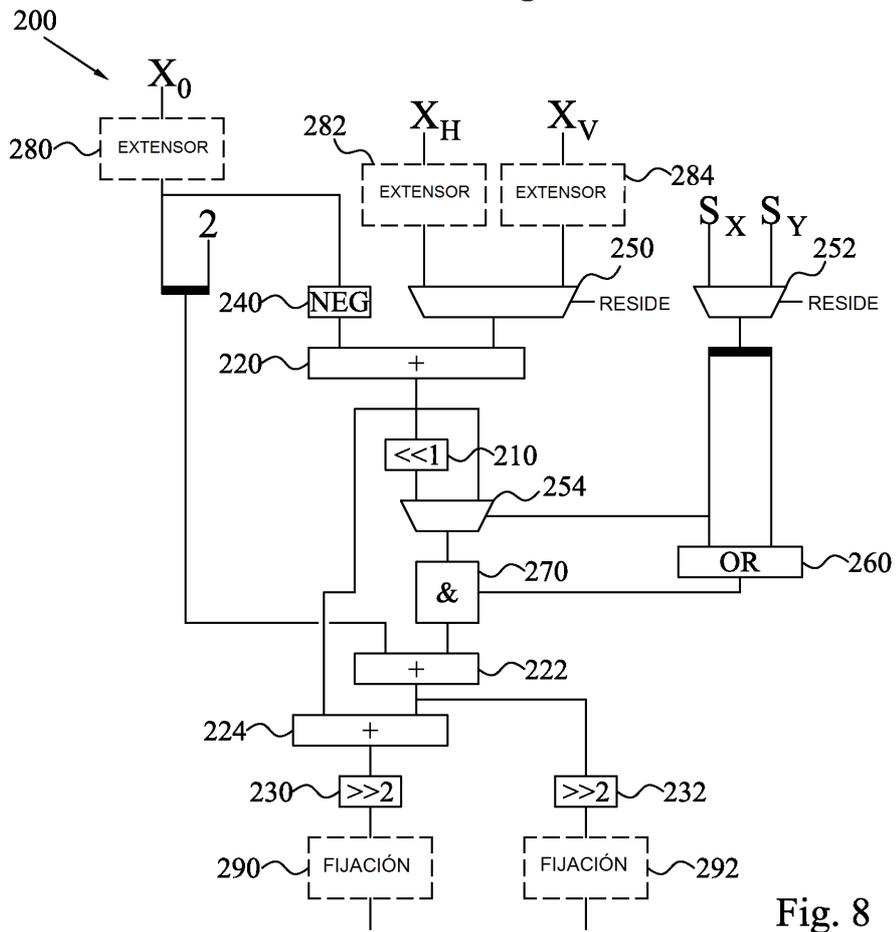
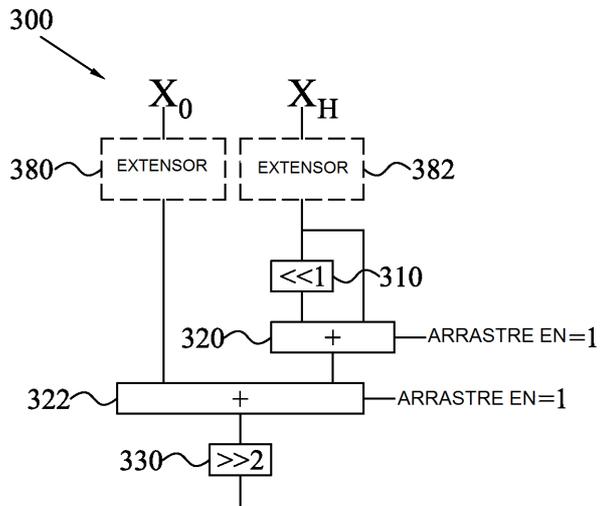
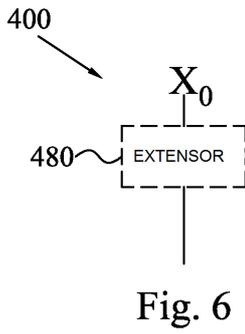


Fig. 5



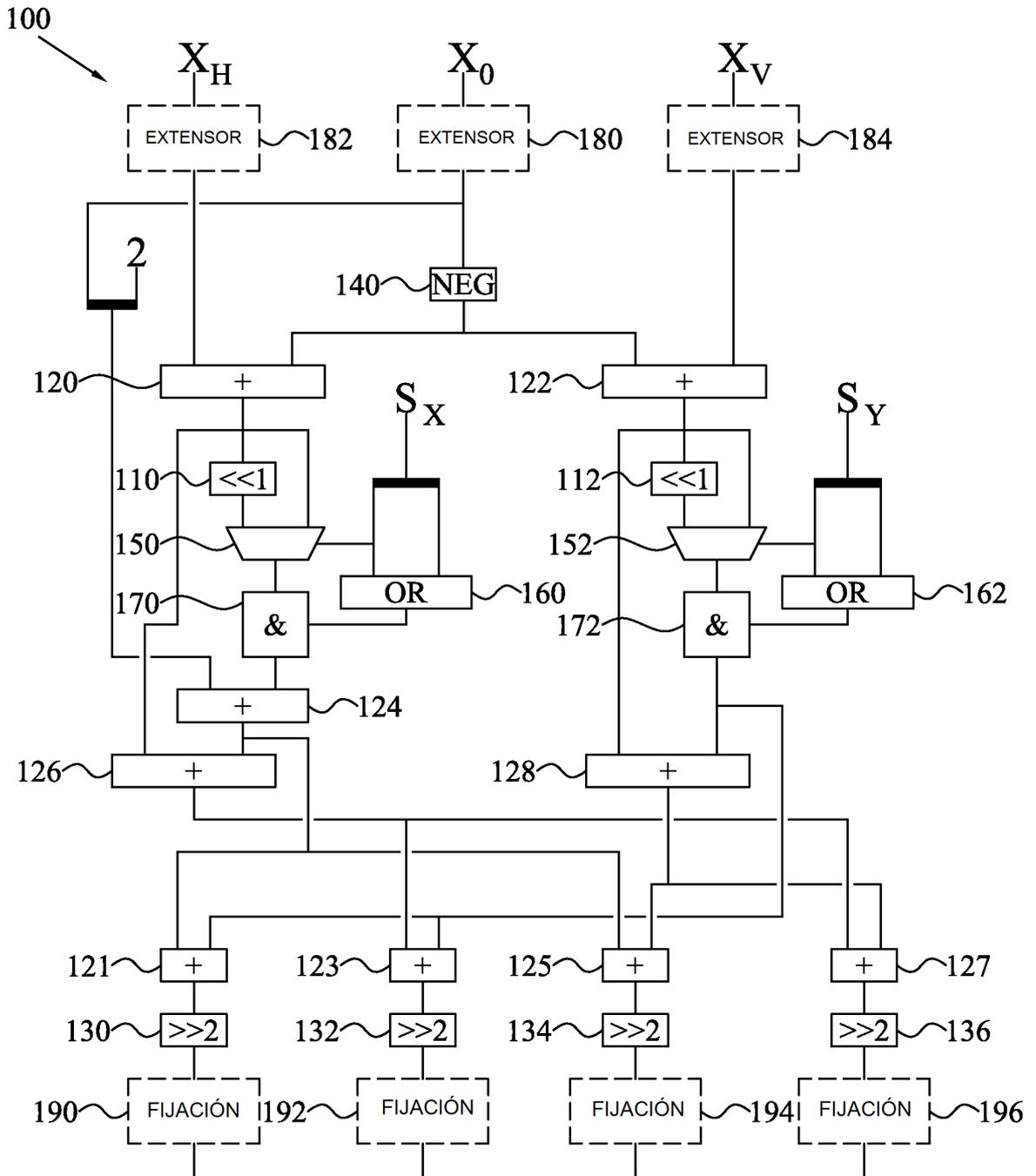
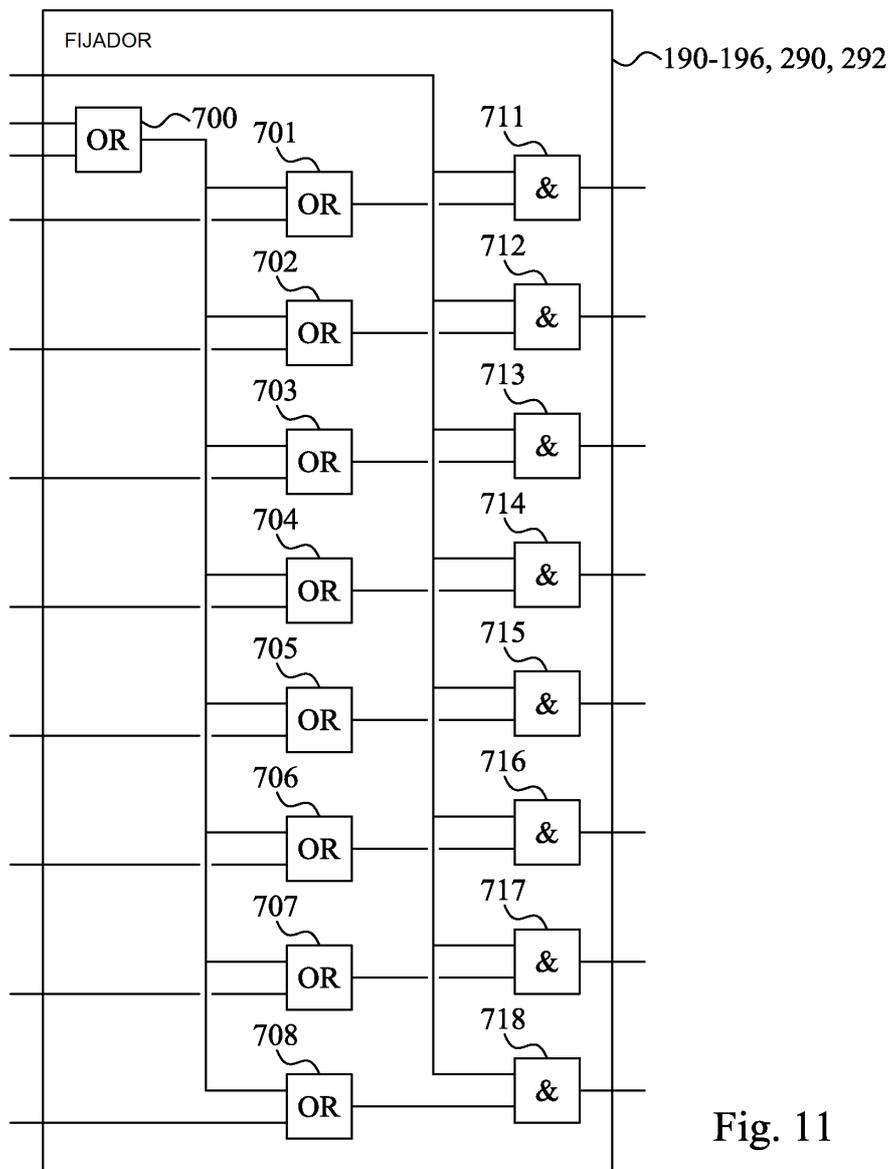
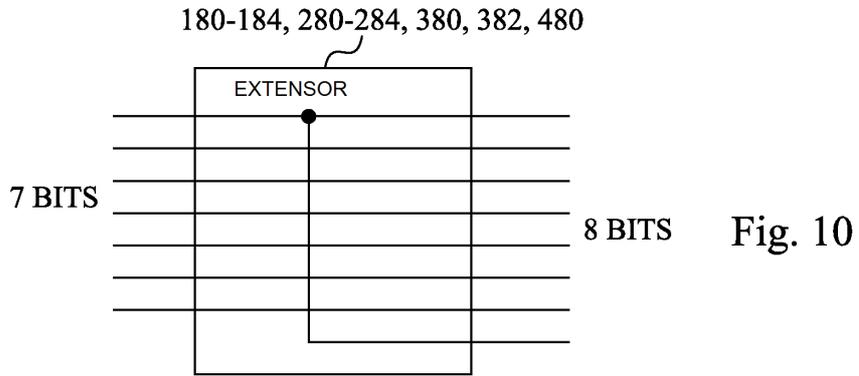


Fig. 9



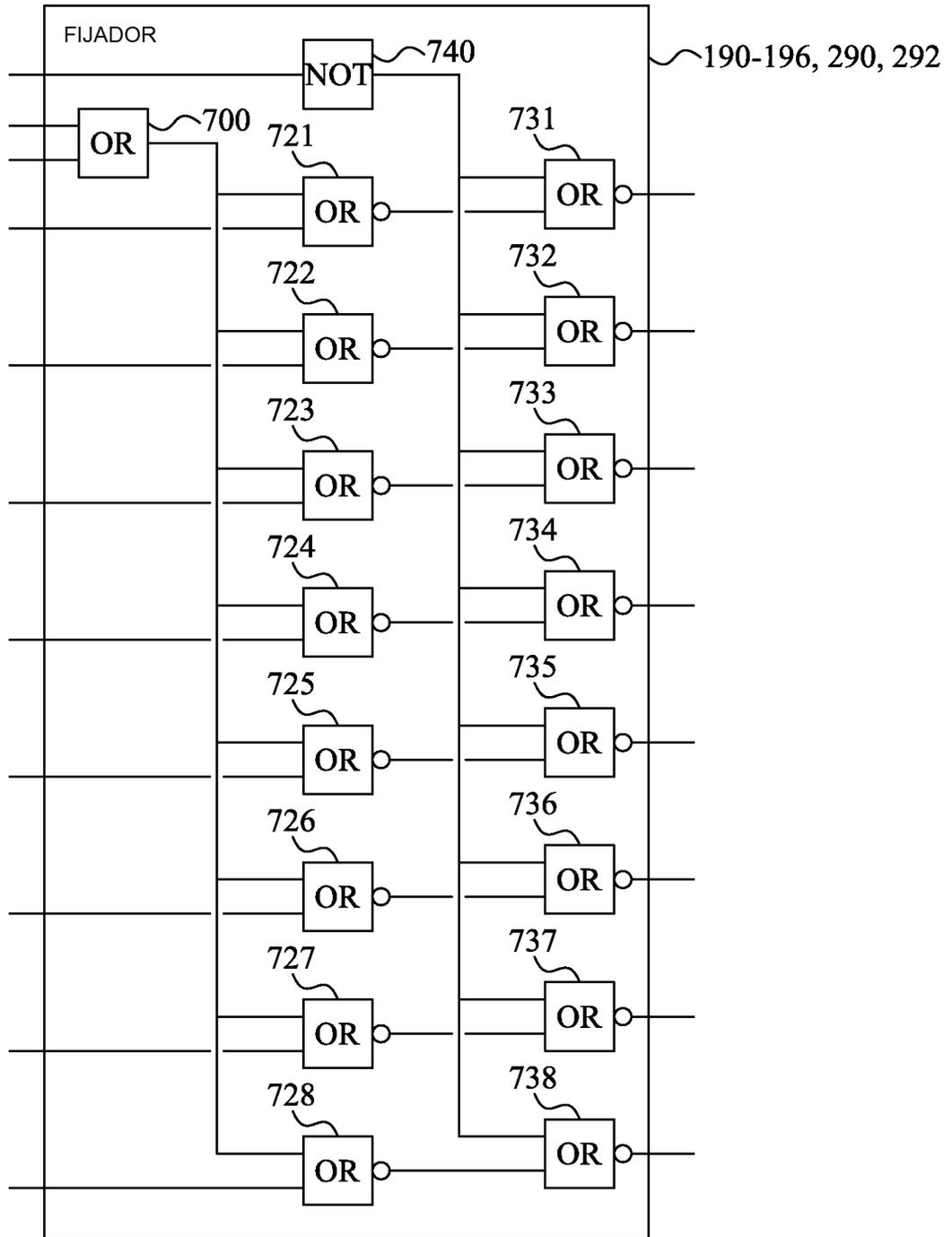


Fig. 12

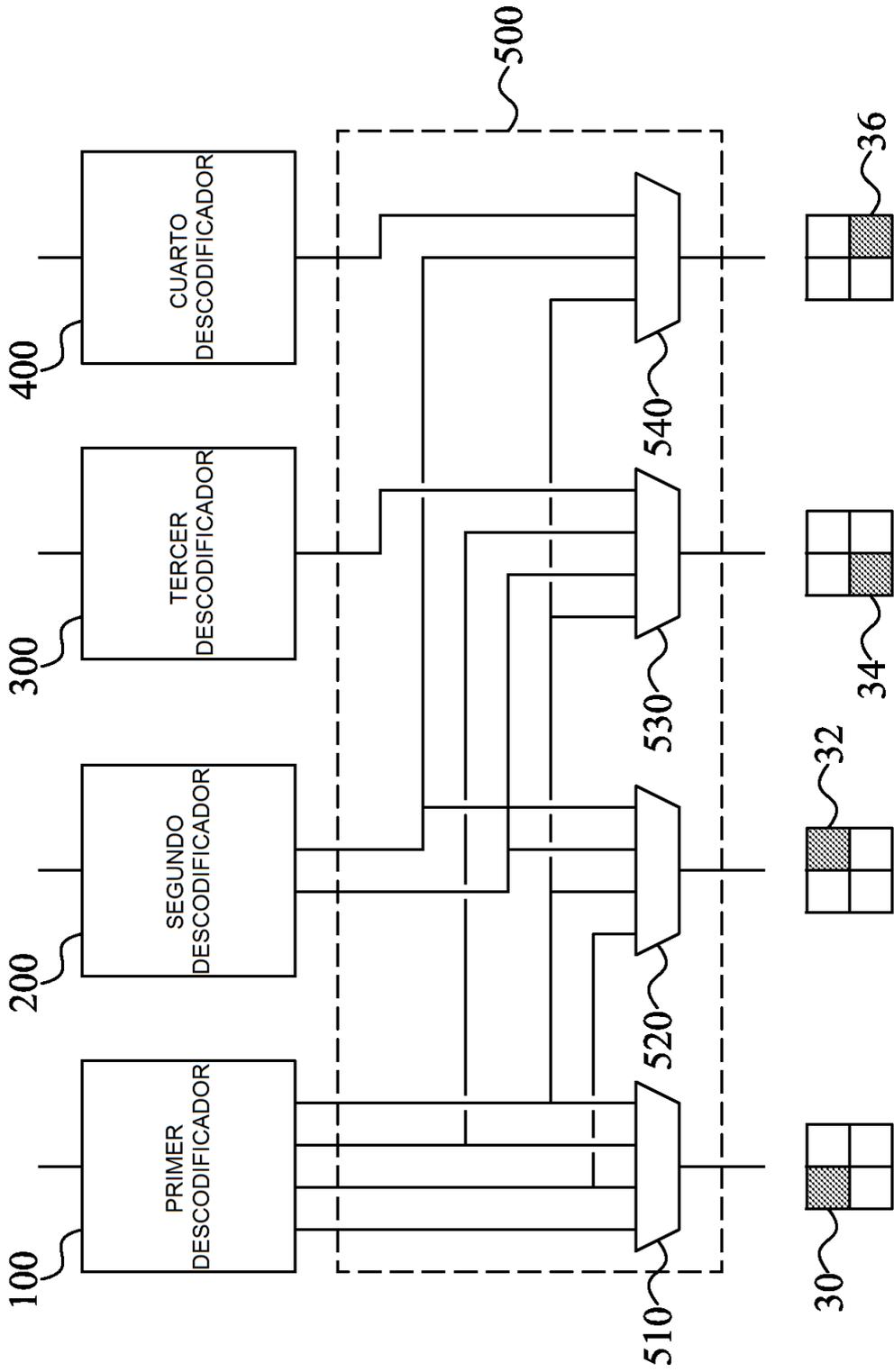


Fig. 13

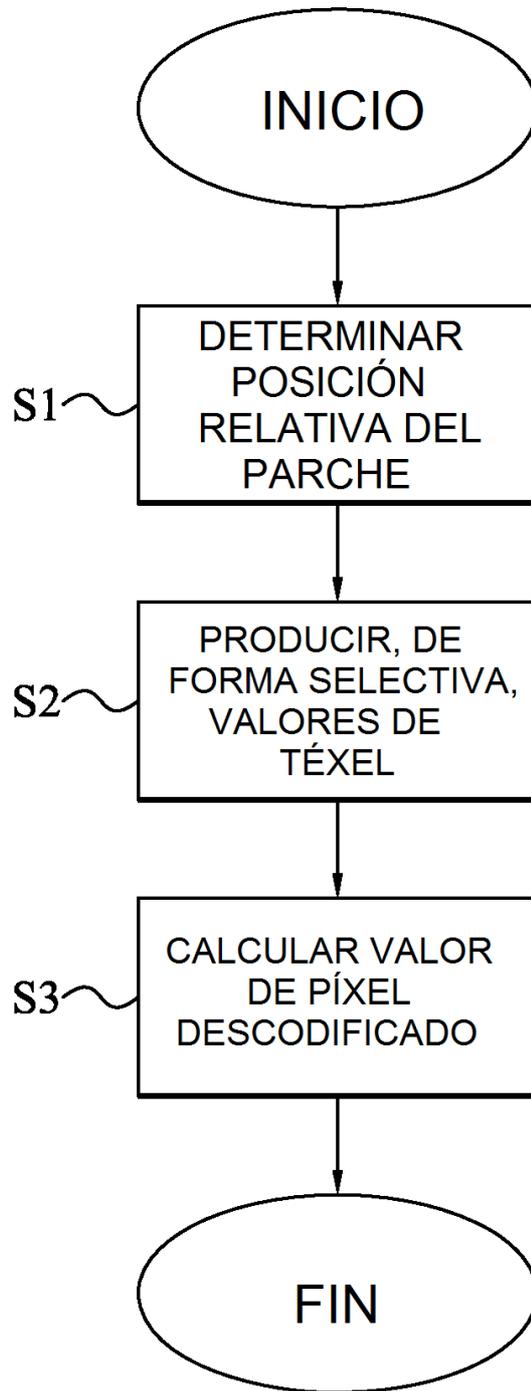


Fig. 14