

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 787 002**

51 Int. Cl.:

G06F 12/0895 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **08.04.2016 PCT/US2016/026664**

87 Fecha y número de publicación internacional: **24.11.2016 WO16186747**

96 Fecha de presentación y número de la solicitud europea: **08.04.2016 E 16717073 (7)**

97 Fecha y número de publicación de la concesión europea: **12.02.2020 EP 3298493**

54 Título: **Procedimiento y aparato para compresión de etiquetas de memoria caché**

30 Prioridad:

20.05.2015 US 201514716947

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

14.10.2020

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US**

72 Inventor/es:

**PELLERIN, III, HENRY ARTHUR;
SPEIER, THOMAS PHILIP;
SARTORIUS, THOMAS ANDREW;
MORROW, MICHAEL WILLIAM;
DIEFFENDERFER, JAMES NORRIS;
DOCKSER, KENNETH ALAN y
MCILVAINE, MICHAEL SCOTT**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 787 002 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

Procedimiento y aparato para compresión de etiquetas de memoria caché

5 **ANTECEDENTES**

[0001] Los aspectos descritos en el presente documento se refieren al campo de los microprocesadores informáticos (también denominados procesadores en el presente documento). Más específicamente, los aspectos divulgados en el presente documento se refieren a la compresión de etiquetas de memoria caché.

10

[0002] Los procesadores de módem convencionalmente se basan en memorias caché para mejorar el rendimiento de procesamiento. Las memorias caché funcionan aprovechando la localidad temporal y espacial en las corrientes de instrucciones y las corrientes de datos de la carga de trabajo. Una parte de la memoria caché está dedicada a almacenar matrices de etiquetas de memoria caché. Las etiquetas de memoria caché almacenan la dirección de los datos reales obtenidos de la memoria principal. La velocidad de determinación de un acierto o un fallo en la memoria caché está limitada por la anchura de la matriz de etiquetas, donde las matrices de etiquetas más grandes crean convencionalmente más presión de temporización. Una etiqueta más ancha (o más grande) tiene más bits. Para determinar si hay un acierto en la memoria caché, todos los bits de la etiqueta se deben comparar con la dirección de sondeo. Por lo tanto, las comparaciones con más bits pueden requerir más tiempo para presentar un resultado de la comparación. El documento US 2013/007358 divulga un tipo de memoria caché que comprende una estructura de compresión de etiquetas.

15

20

BREVE EXPLICACIÓN

25

[0003] La invención se expone en el conjunto adjunto de reivindicaciones. Las reivindicaciones dependientes exponen modos de realización particulares. Se considera que los modos de realización o los ejemplos de la siguiente descripción que no están cubiertos por las reivindicaciones adjuntas no forman parte de la invención de acuerdo con esta descripción.

30

BREVE DESCRIPCIÓN DE LAS DIVERSAS VISTAS DE LOS DIBUJOS

[0004] Para que la manera en que se alcanzan los aspectos mencionados anteriormente se pueda comprender en detalle, se puede obtener una descripción más particular de los aspectos de la divulgación, resumidos brevemente anteriormente, haciendo referencia a los dibujos adjuntos.

35

[0005] Sin embargo, cabe señalar que los dibujos adjuntos ilustran solo unos aspectos de esta divulgación y, por lo tanto, no se han de considerar limitantes de su alcance, ya que la divulgación puede admitir otros aspectos.

40

La figura 1 ilustra un procesador que proporciona compresión de etiquetas de memoria caché, de acuerdo con un aspecto.

La figura 2 es una vista lógica de los componentes de un procesador configurados para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto.

45

La figura 3 es un esquema que ilustra valores de ejemplo almacenados en una memoria caché de compresión de etiquetas, una matriz de etiquetas y una matriz de datos, de acuerdo con un aspecto.

50

La figura 4 es un diagrama de flujo que ilustra un procedimiento para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto.

La figura 5 es un diagrama de flujo que ilustra un procedimiento para establecer una nueva entrada de memoria caché de compresión de etiquetas, de acuerdo con un aspecto.

55

La figura 6 es un diagrama de bloques que ilustra un dispositivo informático que integra un procesador configurado para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto.

DESCRIPCIÓN DETALLADA

60

[0006] Los aspectos divulgados en el presente documento proporcionan técnicas para reducir el área para el almacenamiento de etiquetas de memoria. Por ejemplo, las memorias caché se pueden etiquetar con algún conjunto de bits de dirección superior. En general, las etiquetas de memoria caché requieren almacenamiento para un gran número de bits, y las arquitecturas de memoria caché modernas convencionalmente requieren etiquetas sustancialmente más grandes que en las generaciones anteriores. Por ejemplo, se va a considerar una memoria caché de instrucciones de 4 vías de 32 kilobytes (KB) hipotética que tiene líneas de 64 bits. El número de líneas de dicha memoria caché es de $2^{15}/2^9$, o 512 líneas. Cada línea está asociada con una etiqueta. El número de conjuntos de la memoria caché es igual a $2^{15}/2^6/2^2$, o 128 conjuntos (con una decodificación de conjunto de 7

65

bits). En al menos un aspecto, los bits de etiqueta no incluyen la descodificación de conjunto de 7 bits o el desplazamiento de bytes de 6 bits, lo que significa que los bits de etiqueta empiezan en A[7+6] o A[13], donde "A" indica una entrada de dirección para la búsqueda de etiquetas. Por lo tanto, para una dirección de ejemplo de 64 bits, la etiqueta se compara con A [63:13], y el almacenamiento de etiqueta es de 51 bits por línea. El almacenamiento de etiquetas total para la memoria caché es, por lo tanto, de 51 bits/línea * 512 líneas, o 26 112 bits, lo que da como resultado una sobrecarga de almacenamiento del 10 % para almacenar las etiquetas de la memoria caché.

[0007] Aunque los valores de etiqueta son arbitrarios y dependientes del código, un subconjunto de los bits de cada uno de los valores de etiqueta puede tener poca variabilidad, de modo que el subconjunto de bits tiene los mismos valores en múltiples etiquetas de memoria caché. Los aspectos divulgados en el presente documento aprovechan esta tendencia eliminando el almacenamiento redundante de múltiples copias del mismo valor en múltiples etiquetas de memoria caché. Específicamente, los aspectos divulgados en el presente documento introducen una memoria caché de compresión de etiquetas que está configurada para almacenar valores de etiquetas únicos. Con ello, se elimina un número de bits de la etiqueta de memoria caché. Los bits que se eliminan de la etiqueta de memoria caché se pueden reemplazar por un número menor de bits que apuntan a una entrada de memoria caché de compresión de etiquetas. La memoria caché de compresión de etiquetas puede contener cualquier conjunto de bits de la etiqueta de memoria caché, que incluyen, sin limitación, bits de dirección superior (o más alta), identificadores de espacio de direcciones (ASID), bits de seguridad, bits de nivel de excepción y similares.

[0008] Dado que la memoria caché de compresión de etiquetas contiene una parte de la etiqueta de memoria caché original, un acceso debe "acertar" con la memoria caché de compresión de etiquetas para "acertar" con la memoria caché. Del mismo modo, un acceso que "falla" con la memoria caché de compresión de etiquetas también debe "fallar" con la propia memoria caché. El desalojo de una entrada de memoria caché de compresión de etiquetas requiere la invalidación de todas las líneas de memoria caché correspondientes (en concreto, las líneas que tienen un valor de índice de memoria caché de compresión de etiquetas que coincide con el índice de la memoria caché de compresión de etiquetas que se está desalojando). Sin embargo, invalidar una línea de memoria caché no requiere ninguna actualización de la memoria caché de compresión de etiquetas.

[0009] Los aspectos divulgados en el presente documento acceden a la memoria caché de compresión de etiquetas y a la memoria caché en paralelo para determinar si una dirección de memoria de entrada da como resultado un "acierto" o un "fallo" de memoria caché. Específicamente, los aspectos divulgados en el presente documento acceden a la memoria caché de compresión de etiquetas usando un conjunto de bits de orden superior de la dirección de memoria de entrada. Si hay un acierto en la memoria caché de compresión de etiquetas (es decir, una entrada en la memoria caché de compresión de etiquetas coincide con los bits de orden superior de la dirección de memoria de entrada), la memoria caché de compresión de etiquetas está configurada para presentar el valor de índice de la entrada coincidente. En paralelo, los aspectos divulgados en el presente documento buscan en la memoria caché usando un conjunto de bits de orden inferior de la dirección de memoria de entrada. Si hay una coincidencia en la memoria caché (es decir, una entrada de etiqueta de la matriz de etiquetas de memoria caché coincide con los bits de orden inferior de la dirección de memoria de entrada), la memoria caché está configurada para presentar los bits de orden inferior y el valor de índice de la entrada coincidente. Los aspectos divulgados en el presente documento pueden comparar a continuación los valores de índice presentados desde la memoria caché y la memoria caché de compresión de etiquetas. Si los valores de índice no coinciden, los aspectos divulgados en el presente documento presentan una indicación de un fallo de memoria caché. Los aspectos divulgados en el presente documento pueden comparar además los bits inferiores presentados desde la entrada de la etiqueta de memoria caché con los bits inferiores de la dirección de memoria de entrada. Si los bits inferiores no coinciden, los aspectos divulgados en el presente documento presentan una indicación de fallo de memoria caché. Sin embargo, si los valores de índice coinciden y los bits inferiores coinciden, los aspectos divulgados en el presente documento presentan una indicación de un acierto de memoria caché (es decir, los datos solicitados están presentes en la memoria caché).

[0010] En el presente documento se usa una memoria caché como ejemplo de referencia para facilitar la descripción de aspectos de la divulgación. Sin embargo, el uso de una memoria caché no se debe considerar limitante de la divulgación, ya que las técnicas descritas en el presente documento se aplican a otras estructuras de hardware que tienen campos de etiquetas, tales como memorias caché de procesador, memorias caché de datos, memorias caché de instrucciones, memorias intermedias de relleno de línea, memorias intermedias de traducción adelantada (TLB), y similares. Cualquier referencia a una estructura de hardware en particular no se debe considerar limitante de la divulgación.

[0011] La figura 1 ilustra un procesador 101 que proporciona compresión de etiquetas de memoria caché, de acuerdo con un aspecto. Como se muestra, el procesador 101 incluye una segmentación de ejecución de instrucciones 112 que ejecuta instrucciones. La segmentación 112 puede ser un diseño superescalar, con múltiples segmentaciones paralelas, cada una de las cuales incluye diversos registros no arquitectónicos (no representados) y una o más unidades de lógica aritmética (tampoco representadas). Como se muestra, el procesador 101 también incluye una memoria caché 102 que almacena líneas de datos de uno o más niveles superiores de memoria 108.

Los niveles más altos de memoria 108 pueden incluir, sin limitación, memorias caché de nivel superior y/o una memoria (de sistema) principal. En al menos un aspecto, la memoria caché 102 es una memoria caché de datos de nivel 1 (L1). En general, la CPU 101 puede incluir numerosas variantes, y la CPU 101 mostrada en la figura 2 es ilustrativa y no se debe considerar limitante de la divulgación. Por ejemplo, la CPU 101 puede ser una unidad de procesamiento de gráficos (GPU).

[0012] Como se muestra, la memoria caché 102 incluye la lógica de memoria caché 103, una memoria caché de compresión de etiquetas 104, una matriz de etiquetas 105 y una matriz de datos 106. La lógica de memoria caché 103 en general controla el funcionamiento de la memoria caché 102, tal como cuando se determina si se producen aciertos o errores de memoria caché en una operación en particular. La memoria caché de compresión de etiquetas (TCC) 104 es una estructura de hardware configurada para contener un conjunto de bits de orden superior (o más alto) de etiquetas de memoria. Por ejemplo, la TCC 104 puede contener uno o más de los bits de dirección superior de la etiqueta de memoria, bits ASID, bits de seguridad, bits de nivel de excepción y similares. La TCC 104 en general está configurada para almacenar valores redundantes que de otro modo se almacenarían de forma redundante en la matriz de etiquetas 105. La matriz de etiquetas 105 está configurada en general para almacenar las direcciones de datos almacenados en la memoria caché 102. Más específicamente, cada entrada de la matriz de etiquetas 105 está configurada para almacenar un conjunto de bits de orden inferior de etiquetas de memoria y un valor de índice que corresponde al valor de índice de una entrada de la TCC 104. La matriz de datos 106 almacena los datos de las líneas de memoria caché. En al menos un aspecto, la TCC 104 se implementa como una estructura de memoria de contenido direccionable (CAM). De manera similar, la matriz de etiquetas 105 y la matriz de datos 106 se pueden implementar como una CAM. Además, en al menos un aspecto, la TCC 104 es una estructura de reemplazo menos usada recientemente (LRU) totalmente asociativa. La TCC 104 puede tener cualquier número de entradas, tal como dos, cuatro u ocho. Los medios para almacenar datos en la TCC 104, la matriz de etiquetas 105 y la matriz de datos 106 pueden incluir una o más celdas de memoria.

[0013] En funcionamiento, el procesador 101 puede tratar de determinar si los datos localizados en uno de los niveles más altos de memoria 108 están presentes dentro de la memoria caché 102. Cuando el procesador 101 proporciona una dirección de memoria de entrada (que puede ser una dirección virtual) a la memoria caché 102, la lógica de memoria caché 103 puede determinar si los datos en la dirección de memoria de entrada están en la memoria caché 102. Para ello, la lógica de memoria caché 103 puede iniciar una búsqueda CAM de la TCC 104 usando los bits de orden superior de la dirección de memoria de entrada. Si una entrada de la TCC 104 coincide con los bits de orden superior de la dirección de memoria de entrada, la TCC 104 está configurada para presentar el valor de índice de la entrada coincidente en la TCC 104. La lógica de memoria caché 103 puede iniciar una búsqueda CAM del conjunto de etiquetas 105 en paralelo con la búsqueda CAM de la TCC 104. La lógica de memoria caché 103 puede buscar en la matriz de etiquetas 105 usando los bits de orden inferior de la dirección de memoria de entrada. Si los bits de orden inferior de la dirección de memoria de entrada, la matriz de etiquetas 105 está configurada para presentar los bits de orden inferior y el valor de índice de la entrada coincidente. La lógica de memoria caché 103 puede comparar a continuación el valor de índice presentado por la TCC 104 y la matriz de etiquetas 105. Si los valores de índice no coinciden, la lógica de memoria caché 103 puede presentar una indicación de fallo de memoria caché. La lógica de memoria caché 103 también puede comparar los bits de orden inferior presentados por la matriz de etiquetas 105 con los bits de orden inferior de la dirección de memoria de entrada. Si los bits de orden inferior coinciden y los valores de índice coinciden, la lógica de memoria caché 103 está configurada para presentar una indicación de acierto de memoria caché.

[0014] Si la búsqueda de la TCC 104 da como resultado un fallo, la lógica de memoria caché 103 puede estar configurada para crear una entrada para la dirección de memoria de entrada en la TCC 104. Si existe una entrada libre (o disponible de otro modo) en la TCC 104, la lógica de memoria caché 103 puede almacenar un conjunto de bits de orden superior de la dirección de memoria de entrada en la entrada libre. Si no hay entradas disponibles en la TCC 104, la lógica de memoria caché 103 puede desalojar una entrada existente en la TCC 104 e invalidar cualquier entrada correspondiente en la matriz de etiquetas 105 (y la matriz de datos 106) que especifique el valor de índice de la entrada desalojada. La lógica de memoria caché 103 puede aplicar cualquier política de reemplazo adecuada para la TCC 104, tal como una LRU, o reemplazar la entrada de TCC 104 que está asociada con la menor cantidad de líneas de memoria caché (que se puede determinar mediante un contador para cada entrada de la TCC 104). La lógica de memoria caché 103 puede invalidar las entradas de cualquier manera factible, tal como mediante una invalidación *flash* o una invalidación línea por línea. Además de añadir una entrada que especifica los bits de orden superior de la dirección de memoria de entrada a la TCC 104, la lógica de memoria caché 103 también puede crear una entrada en la matriz de etiquetas 105 que especifica los bits de orden inferior de la dirección de memoria de entrada y un valor de índice de la entrada de TCC 104 que almacena los bits de orden superior de la dirección de memoria de entrada.

[0015] En al menos un aspecto, la lógica de memoria caché 103 puede sondear la TCC 104 en cada ciclo de reloj. Sin embargo, esto puede consumir grandes cantidades de energía y puede estar en la vía de la temporización crítica. Cuantas más entradas estén integradas en la TCC 104, más potencia y tiempo se podrían necesitar para sondear la TCC 104. Por lo tanto, en algunos aspectos, la lógica de memoria caché 103 puede incluir una lógica que evita la necesidad de buscar en la TCC 104 en una mayoría de los ciclos. Por ejemplo, un conjunto de registros

puede estar configurado para almacenar los bits presentados por un acceso previo a la TCC 104 (y si la entrada de TCC correspondiente 104 sigue siendo válida). En general, la lógica de memoria caché 103 puede determinar si los bits de dirección de memoria de entrada correspondientes a los que se mantienen en la TCC 104 son los mismos que en el acceso previo a la TCC 104. La lógica de memoria caché 103 puede determinar a continuación si la entrada de TCC 104 asociada no se ha invalidado desde el último acceso. Si ambas determinaciones son ciertas, la lógica de memoria caché 103 puede determinar que la etiqueta de memoria caché de compresión debe estar en la TCC 104, y que es seguro reusar el índice del ciclo anterior, haciendo innecesario buscar en la TCC 104 en el ciclo actual. Si cualquiera de las determinaciones no es cierta, la lógica de memoria caché 103 no puede reusar el índice de TCC 104 del ciclo previo y la lógica de control 103 debe buscar en la TCC 104. Los medios para buscar en la TCC 104, la matriz de etiquetas 105 y la matriz de datos 106 incluyen lógica implementada como hardware y/o software. De forma similar, la lógica implementada como hardware y/o software puede servir como medio para leer y/o escribir valores, presentar indicaciones de aciertos y/o fallos, y presentar valores de la TCC 104, la matriz de etiquetas 105 y la matriz de datos 106. Los ejemplos de dichos medios lógicos incluyen controladores de memoria, controladores de memoria caché y controladores de datos.

[0016] Como se ha indicado previamente, el sistema de compresión representado en la figura 1 se puede aplicar a otras estructuras de microarquitectura con etiquetas de PC y/o ASID. Por ejemplo, una dirección virtual se puede transformar en la forma {valor de índice de compresión de etiqueta, dirección virtual parcial}, que se puede usar en todo el bloque. Con ello se permitiría que todos los elementos de la trayectoria de datos funcionaran en esta forma acortada, incluyendo una memoria intermedia de traducción (TLB), unos predictores de ramificación, unos puntos de interrupción, unas memorias caché y similares. Si es necesario, la dirección transformada se podría convertir en la dirección original siempre que fuera necesaria, tal como en límites de bloque, para servicios de depuración de contenido de memoria caché y similares. De forma similar, la técnica de compresión de etiquetas se puede aplicar independientemente a direcciones de instrucciones, direcciones de carga/almacenamiento y estructuras etiquetadas físicamente.

[0017] En un aspecto, el procesador 101 está dispuesto en un circuito integrado que incluye la segmentación de ejecución de instrucciones 112 y la memoria caché 102. En otro aspecto, la memoria caché 102 puede estar localizada en un circuito integrado separado de un circuito integrado que incluye el procesador 101.

[0018] La figura 2 es una vista lógica de una parte de una lógica de memoria caché 103 de un procesador 101 configurado para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto. Como se muestra, el procesador 101 incluye la memoria caché de compresión de etiquetas (TCC) 104 y la matriz de etiquetas de memoria caché 105. Como se describe previamente, la TCC 104 es una estructura de hardware indexada que almacena una pluralidad de entradas de TCC 202. Las entradas de TCC 202 pueden incluir un conjunto de bits de dirección de orden superior de una etiqueta de memoria. En al menos algunos aspectos, las entradas de TCC 202 pueden incluir además bits relacionados con ASID, niveles de seguridad, niveles de excepción y similares. Como se muestra, la matriz de etiquetas de memoria caché 105 incluye una pluralidad de entradas, teniendo cada una de las entradas un valor de índice 204 de memoria caché de compresión de etiquetas (TCC) y una etiqueta de memoria caché 205. El valor de índice de TCC 204 puede corresponder al valor del índice de una entrada 202 de la TCC 104. Las etiquetas de memoria caché 205 pueden incluir los bits de orden inferior de una etiqueta de memoria. Como se muestra, las etiquetas de memoria caché 205 almacenan los bits de orden inferior de la etiqueta de memoria, en concreto, los bits 0 a M de una etiqueta de memoria de N bits. De forma similar, la TCC 104 almacena los bits de orden superior de la etiqueta de memoria de N bits, en concreto, los bits M+1 a N de la etiqueta de memoria. Por lo tanto, se puede generar una etiqueta de memoria completa (bits 0 a N) combinando los bits de orden superior almacenados en la TCC 104 (bits M+1 a N) con los bits de orden inferior de la etiqueta de memoria caché 205 (bits 0 a M).

[0019] Como se muestra, el procesador 101 puede solicitar datos almacenados en una dirección de memoria 201. La lógica de memoria caché 103 puede estar configurada para determinar si los datos almacenados en la dirección de memoria 201 están localizados en la memoria caché 102. En al menos un aspecto, la dirección de memoria 201 puede ser una dirección de memoria virtual que tiene una longitud de N bits. Para determinar si los datos almacenados en la dirección de memoria 201 están en la memoria caché 102, la lógica de memoria caché 103 puede sondear la TCC 104 y la matriz de etiquetas de memoria caché 105 en paralelo. Más específicamente, como se muestra, la lógica de memoria caché 103 puede realizar una búsqueda CAM de la matriz de etiquetas de memoria caché 105. Si una de las etiquetas de memoria caché 205 incluye una entrada que coincide con los bits 0 a M de la dirección de memoria 201, la matriz de etiquetas de memoria caché 105 puede presentar un valor de índice de TCC 208 y una etiqueta de memoria caché 209. Además, como se muestra, un divisor 213 puede dividir la dirección de memoria 201, lo cual proporciona los bits M+1 a N de la dirección de memoria 201 a la TCC 104. Se puede realizar una búsqueda CAM de la TCC 104 usando los bits M+1 a N de la dirección de memoria 201. Si una entrada 202 de la TCC 104 incluye un valor que coincide con el valor de los bits M+1 a N de la dirección de memoria 201, la TCC 104 está configurada para presentar una entrada de TCC 206 y un valor de índice de TCC 207, donde el valor de índice 207 identifica la entrada de TCC coincidente 206. El valor de índice de TCC 207 puede tener una longitud de cualquier número de bits, adecuada para permitir que cada entrada 202 de la TCC 104 se identifique de forma exclusiva mediante un valor de índice de TCC 207 respectivo. Por lo tanto, por ejemplo, si la TCC 104 tuviera cuatro entradas de hardware (por ejemplo, cuatro entradas 202 de TCC), el valor de índice

de TCC 207 correspondiente a cada entrada podría tener dos bits de longitud. En dicho ejemplo, cada índice de TCC 204 almacenado en la matriz de etiquetas de memoria caché 105 también tendría dos bits de longitud. En al menos un aspecto, los valores de índice de TCC 207 se identifican implícitamente en base a la ubicación (o dirección) de una entrada de TCC 202 dada.

5

[0020] Como se muestra, un comparador 210 compara el valor de índice de TCC 207 presentado desde la TCC 104 con el valor de índice de TCC 208 presentado por la matriz de etiquetas de memoria caché 105. Además, un comparador 211 compara los bits de orden inferior 0:M de la dirección de memoria 201 con los bits de orden inferior 0:M almacenados en una entrada de etiqueta de memoria caché 209 presentada desde la matriz de etiquetas de memoria caché 105. Como se muestra, la lógica AND 212 está configurada para realizar una operación lógica AND con las salidas de los comparadores 210, 211. Si ambos comparadores 210, 211 indican una coincidencia, entonces la salida de la lógica AND 212 está configurada para presentar una indicación de un acierto de memoria caché. Dicho de otra manera, si los valores de índice comparados por el comparador 210 coinciden, y los bits de orden inferior 0:M comparados por el comparador 211 coinciden, entonces el contenido de la dirección de memoria solicitada 201 se almacena en la memoria caché 102, y la lógica 103 de la memoria caché puede presentar una indicación de acierto de memoria caché. Sin embargo, si el comparador 210 indica que los valores de índice comparados no coinciden, o si el comparador 211 indica que los bits de orden inferior 0:M no coinciden, los datos solicitados no están en la memoria caché 102, y la lógica de memoria caché 103 puede presentar una indicación de fallo de memoria caché. Por lo tanto, como se muestra, la lógica de memoria caché 103 provoca una comparación de bits 0:M y valores de índice. Con ello se permite que la lógica de memoria caché 103 compare menos bits que en una comparación de etiquetas tradicional (de bits 0:N), lo que da como resultado una comparación más rápida que la comparación tradicional.

10

15

20

25

[0021] En caso de fallo de memoria caché, la lógica de memoria caché 103 puede estar configurada para recuperar los datos solicitados de una memoria de orden superior y llenar la TCC 104 y la matriz de etiquetas de memoria caché 105 como se describe anteriormente. De forma similar, si falla cualquiera de las búsquedas CAM iniciales de la TCC 104 y la matriz de etiquetas de memoria caché 105, la lógica de memoria caché 103 puede estar configurada para obtener los datos solicitados desde una memoria de orden superior, y llenar la TCC 104 y la matriz de etiquetas de memoria caché 105 como se describe anteriormente.

30

[0022] La figura 3 es un esquema 300 que ilustra valores de ejemplo almacenados en la memoria caché de compresión de etiquetas (TCC) 104, la matriz de etiquetas de memoria caché 105 y la matriz de datos 106, de acuerdo con un aspecto. Como se muestra, cada entrada de TCC 202 de la TCC 104 especifica una parte de una etiqueta de memoria y se identifica mediante un valor de índice de TCC 207 respectivo. Los valores de índice 207 respectivos identifican de forma exclusiva cada entrada de TCC 202. En al menos un aspecto, el valor de índice 207 se proporciona implícitamente mediante la ubicación (o dirección) de cada entrada de la TCC 104. En dichos aspectos, por lo tanto, el valor de índice 207 no se define expresamente en la TCC 104, y los valores de índice 207 se representan en la figura 3 para facilitar la explicación de dichos aspectos. La parte de etiqueta de cada entrada de TCC 202 puede ser los bits de dirección de orden superior de una etiqueta de memoria, y puede incluir información adicional, tal como bits ASID, bits de seguridad, bits de nivel de excepción y similares.

35

40

[0023] Como se muestra, la tabla 301 incluye valores de ejemplo de la matriz de etiquetas de memoria caché 105 y la matriz de datos 106. La matriz de datos 106 almacena una línea de datos 302 para cada entrada de la memoria caché 102. La matriz de etiquetas de memoria caché 105, como se describe previamente, incluye valores de índice 204 y etiquetas de memoria caché 205. Los valores de índice 204 de la matriz de etiquetas de memoria caché 105 corresponden al valor de índice 207 correspondiente a una entrada de TCC 202 dada de la TCC 104, mientras que cada etiqueta de memoria caché 205 correspondiente almacena los bits de orden inferior de una etiqueta de memoria. Por lo tanto, como se muestra, las entradas múltiples de la tabla 301 pueden almacenar los mismos valores de índice 204. De forma ventajosa, la TCC 104 almacena una copia única del patrón de bits completo de bits de orden superior de la etiqueta de memoria, reduciendo la cantidad de información redundante almacenada en la matriz de etiquetas de memoria caché 105 (en concreto, los diferentes valores de índice 204, y no los bits de orden superior más largos de la etiqueta de memoria). Por lo tanto, una etiqueta de memoria se almacena en la TCC 104 y la matriz de etiquetas de memoria caché 105, estando los bits de orden superior almacenados en la TCC 104, y los bits de orden inferior almacenados en la matriz de etiquetas de memoria caché 105. Desde un punto de vista conceptual, la parte de etiqueta de la entrada de TCC 202 de la TCC 104 puede reemplazar el valor de índice 204 de la matriz de etiquetas de memoria caché 105 para generar una etiqueta de memoria completa. Por ejemplo, en la primera fila de la tabla 301, se reemplazarían los bits "00" del índice 204 por los bits de etiqueta "01... 0100" de la entrada de TCC 202 correspondiente al valor de índice 207 de "00" de la TCC 104.

50

55

60

[0024] En un aspecto de ejemplo, la configuración representada en la figura 3 puede reemplazar una memoria caché de instrucciones hipotética por una organización de ejemplo representada en la tabla 1 :

Tabla 1:

| | |
|--|--|
| Tamaño de línea | 64 bytes |
| Asociatividad | 4 vías |
| Número de conjuntos | 128 |
| Número total de líneas | 512 |
| Capacidad nominal | 32 KB |
| Anchura de dirección | 49 bits |
| Índice establecido | Dirección virtual (7 bits) |
| Número de bits de etiqueta (por línea) | {Dirección virtual, ASID} (36 + 16 = 52 bits) |
| Número de bits de etiqueta (total) | (52 bits por línea * 512 líneas) = 26 624 bits |

5 **[0025]** Por lo tanto, como se muestra en la tabla 1, la memoria caché de instrucciones hipotética requiere 26 624 bits de sobrecarga de almacenamiento para almacenamiento de etiquetas, del almacenamiento de memoria caché total de 32 KB (o 32 768 bytes, o 262 144 bits). La etiqueta de memoria caché de instrucciones requerida en esta configuración de ejemplo sería {PC [48:13], ASID}.

10 **[0026]** Sin embargo, un subconjunto de estos bits (tal como {PC [48:28], ASID}) tiende a mostrar poca variabilidad en las cargas de trabajo reales. Al crear una TCC 104 de cuatro entradas para esta memoria caché de instrucciones de ejemplo, cada entrada de TCC 202 almacenaría los 21 bits PC[48:28] y los 16 bits de ASID, para un total de 37 bits. Por lo tanto, la cantidad total de almacenamiento requerida para la TCC 104 sería de 4 entradas multiplicadas por 37 bits, o 148 bits. Las entradas de TCC 202 se pueden indexar implícitamente mediante la ubicación (o dirección) de cada entrada de TCC 202 respectiva de la TCC 104 (por ejemplo, en la TCC de cuatro entradas 104, las ubicaciones 0, 1, 2 y 3 corresponden a valores de índice binario de 00, 01, 10 y 11, respectivamente). Mientras tanto, los índices 204 de la matriz de etiquetas de memoria caché 105 almacenarían los valores de índice de dos bits, mientras que las etiquetas de memoria caché 205 almacenarían los bits de etiqueta PC[27:13], o 15 bits, para un total de 17 bits. A través de las 512 líneas de la memoria caché de instrucciones, el almacenamiento total en la matriz de etiquetas de memoria caché 105 sería de 512 entradas, multiplicadas por 17 bits, para un total de 8704 bits. El almacenamiento total requerido a través de la TCC 104 y la matriz de etiquetas de memoria caché 105 sería de 8852 bits (148 + 8704), que es significativamente menor que los 26 624 bits requeridos por la memoria caché de instrucciones descrita en la tabla 1.

25 **[0027]** La figura 4 es un diagrama de flujo que ilustra un procedimiento 400 para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto. El procedimiento 400 comienza en la etapa 410, donde la lógica de memoria caché 103 puede recibir una dirección de memoria de N bits de longitud. La dirección de memoria se puede recibir como parte de una petición para determinar si los datos almacenados en la dirección de memoria están localizados en la memoria caché 102. La lógica de memoria caché 103 puede a continuación sondear la TCC 104 y la matriz de etiquetas de memoria caché 105 en paralelo. Más específicamente, en la etapa 30 415, la lógica de memoria caché 103 puede realizar una búsqueda CAM en la matriz de etiquetas de memoria caché 105 usando M bits más bajos de la dirección de memoria recibida en la etapa 410. Si una entrada de la matriz de etiquetas de memoria caché 105 almacena un valor que coincide con los M bits inferiores de la dirección de memoria, hay un acierto en la matriz de etiquetas de memoria caché 105, y el procedimiento continúa con la etapa 460. Además, la matriz de etiquetas de memoria caché 105 puede presentar los M bits inferiores de la dirección de memoria y el valor de índice almacenado en la entrada coincidente. Si no hay ninguna entrada en la matriz de etiquetas de memoria caché 105 que almacene un valor que coincida con los M bits inferiores de la dirección de memoria, hay un fallo en la matriz de etiquetas de memoria caché 105, y el procedimiento continúa con la etapa 480.

40 **[0028]** En la etapa 420, la lógica de memoria caché 103 puede sondear la TCC 104 usando los bits de orden superior (M+1 a N) de la dirección de memoria solicitada. Si una entrada de la TCC 104 almacena un valor que coincide con los bits de orden superior de la dirección de memoria solicitada, el procedimiento continúa con la etapa 460. En caso de acierto en la TCC 104, la TCC 104 puede presentar el valor de índice de la entrada en la TCC 104 que coincide con los bits de orden superior de la dirección de memoria. Si hay un fallo en la TCC 104, el procedimiento continúa con la etapa 430, donde la lógica de memoria caché 103 puede determinar si existen 45 entradas disponibles en la TCC 104. Si una entrada no está disponible, el procedimiento continúa con la etapa 440, donde la lógica de memoria caché 103 puede desalojar una entrada de la TCC 104 e invalidar cualquier entrada correspondiente en la matriz de etiquetas de memoria caché 105. La lógica de memoria caché 103 puede usar cualquier política de desalojo adecuada, tal como desalojar la entrada menos usada recientemente, o desalojar la entrada de la TCC 104 que está asociada con la menor cantidad de entradas de la matriz de etiquetas de memoria caché 105. La lógica de memoria caché 103 puede invalidar las entradas en la matriz de etiquetas de memoria caché 105 de cualquier manera factible, tal como mediante una invalidación *flash* o una invalidación línea por línea. En la etapa 450, descrita en mayor detalle con referencia a la figura 5, la lógica de memoria caché 103 puede establecer una nueva entrada en la TCC 104. En general, la lógica de memoria caché 103 puede dividir la 50

dirección de memoria en un conjunto inferior de bits 0:M y un conjunto superior de bits M+1:N. La lógica de memoria caché 103 puede almacenar a continuación el conjunto inferior de bits en la matriz de etiquetas de memoria caché 105, y almacenar el conjunto superior de bits en la TCC 104.

5 **[0029]** En la etapa 460, la lógica de memoria caché 103 puede comparar los valores de índice presentados por la TCC 104 y la matriz de etiquetas de memoria caché 105. La lógica de memoria caché 103 puede comparar además los bits de orden inferior presentados por la matriz de etiquetas de memoria caché 105 con los bits de orden inferior de la dirección de memoria solicitada. Si ambas comparaciones dan como resultado una coincidencia, el procedimiento continúa con la etapa 470, donde la lógica de memoria caché puede presentar una
10 indicación de acierto de memoria caché. Si una (o ambas) de las comparaciones realizadas en la etapa 460 no dan como resultado una coincidencia, el procedimiento continúa con la etapa 480, donde la lógica de memoria caché 103 presenta una indicación de fallo de memoria caché.

15 **[0030]** La figura 5 es un diagrama de flujo que ilustra un procedimiento 500 correspondiente a la etapa 450 para establecer una nueva entrada en la memoria caché de compresión de etiquetas 104, de acuerdo con un aspecto. Como se muestra, el procedimiento 500 comienza en la etapa 510, donde el procesador 101 puede recuperar los datos asociados con la dirección de memoria proporcionada en la etapa 410 desde una memoria de nivel superior (tal como una memoria caché o una memoria principal de nivel superior). En la etapa 520, la lógica de memoria caché 103 puede almacenar los bits de orden superior (tales como los bits M+1:N) de la dirección de memoria en una primera entrada de la TCC 104. En la etapa 530, la lógica de memoria caché 103 puede almacenar el valor de índice de la primera entrada de la TCC 104 y los bits de orden inferior (tales como los bits 0:M) de la dirección de memoria en una primera entrada de la matriz de etiquetas 105. En la etapa 540, la lógica de memoria caché 103 puede almacenar los datos recuperados en la etapa 510 en una entrada de la matriz de datos 106 correspondiente a la primera entrada de la matriz de etiquetas 105.

25 **[0031]** La figura 6 es un diagrama de bloques que ilustra un dispositivo informático 601 que integra el procesador 101 configurado para proporcionar compresión de etiquetas de memoria caché, de acuerdo con un aspecto. Todos los aparatos y procedimientos representados en las figuras 1-5 pueden estar incluidos en, o realizarse mediante, el dispositivo informático 601. El dispositivo informático 601 también puede estar conectado a otros dispositivos informáticos por medio de una red 630. En general, la red 630 puede ser una red de telecomunicaciones y/o una red de área amplia (WAN). En un aspecto en particular, la red 630 es Internet. En general, el dispositivo informático 601 puede ser cualquier dispositivo que incluye un procesador configurado para implementar la compresión de etiquetas de memoria caché, incluyendo, sin limitación, un ordenador de escritorio, un ordenador portátil, una tableta y un teléfono inteligente.

35 **[0032]** El dispositivo informático 601 en general incluye el procesador 101 conectado por medio de un bus 620 a una memoria 608, un dispositivo de interfaz de red 618, un almacenamiento 609, un dispositivo de entrada 622 y un dispositivo de salida 624. El dispositivo informático 601 en general está bajo el control de un sistema operativo (no mostrado). Se puede usar cualquier sistema operativo que admite las funciones divulgadas en el presente documento. El procesador 101 incluido es representativo de una sola CPU, múltiples CPU, una sola CPU que tiene múltiples núcleos de procesamiento, y similares. El dispositivo de interfaz de red 618 puede ser cualquier tipo de dispositivo de comunicaciones de red que permite que el dispositivo informático 601 se comunique con otros dispositivos informáticos por medio de la red 630.

45 **[0033]** El almacenamiento 609 puede ser un dispositivo de almacenamiento persistente. Aunque el almacenamiento 609 se muestra como una sola unidad, el almacenamiento 609 puede ser una combinación de dispositivos de almacenamiento fijos y/o extraíbles, tales como unidades de disco fijo, unidades de estado sólido, almacenamiento SAN, almacenamiento NAS, tarjetas de memoria extraíbles o almacenamiento óptico. La memoria 608 y el almacenamiento 609 pueden formar parte de un espacio de direcciones virtuales que abarca múltiples dispositivos de almacenamiento primario y secundario.

50 **[0034]** El dispositivo de entrada 622 puede ser cualquier dispositivo para proporcionar entradas al dispositivo informático 601. Por ejemplo, se puede usar un teclado y/o un ratón. El dispositivo de salida 624 puede ser cualquier dispositivo para proporcionar salidas a un usuario del dispositivo informático 601. Por ejemplo, el dispositivo de salida 624 puede ser cualquier pantalla de visualización convencional o conjunto de altavoces. Aunque se muestra por separado del dispositivo de entrada 622, el dispositivo de salida 624 y el dispositivo de entrada 622 se pueden combinar. Por ejemplo, se puede usar una pantalla de visualización con una pantalla táctil integrada.

60 **[0035]** De forma ventajosa, los aspectos divulgados en el presente documento reducen la cantidad de espacio requerido para almacenar valores de etiquetas de memoria. Almacenar valores redundantes de bits de orden superior de una etiqueta de memoria en una memoria caché de compresión de etiquetas más pequeña permite usar una estructura de matriz de etiquetas más pequeña para almacenar los bits de orden inferior de la etiqueta de memoria. Con ello, se puede permitir que dichas estructuras se creen usando una pastilla de silicio global más pequeña, lo que reduce los costes de fabricación, reduce la corriente de fuga y mejora la temporización del circuito

65

al permitir que otras estructuras de hardware se coloquen más cercanas entre sí. De forma adicional o alternativa, el espacio guardado se puede emplear en otras estructuras microarquitectónicas.

5 **[0036]** Se ha descrito un número de aspectos. Sin embargo, son posibles diversas modificaciones a estos aspectos, pudiéndose aplicar los principios presentados en el presente documento también a otros aspectos. Las diversas tareas de dichos procedimientos se pueden implementar como conjuntos de instrucciones ejecutables por una o más matrices de elementos lógicos, tales como microprocesadores, controladores integrados o núcleos IP.

10 **[0037]** Cualquier medio adecuado capaz de realizar las operaciones, tal como un procesador, firmware, circuito integrado específico de la aplicación (ASIC), lógica/registros de puertas, controlador de memoria o un controlador de memoria caché puede realizar las diversas operaciones de los procedimientos descritos anteriormente. En general, unos medios funcionales correspondientes capaces de realizar las operaciones pueden realizar cualquier operación ilustrada en las figuras.

15 **[0038]** Los dispositivos y las funcionalidades divulgados anteriormente se pueden diseñar y configurar como archivos de ordenador (por ejemplo, RTL, GDSII, GERBER, etc.) almacenados en medios legibles por ordenador. Algunos, o la totalidad, de dichos archivos se pueden proporcionar a los responsables de fabricación que fabrican dispositivos en base a dichos archivos. Los productos resultantes incluyen obleas semiconductoras que, a continuación, se cortan como una pastilla semiconductor y se ofrecen como un chip semiconductor. Algunos, o la totalidad, de dichos archivos se pueden proporcionar a los responsables de fabricación que configuran los equipos de fabricación usando los datos de diseño para fabricar dispositivos descritos en el presente documento. Los productos resultantes formados a partir de los archivos de ordenador incluyen obleas semiconductoras que a continuación se cortan en pastillas semiconductoras (por ejemplo, el procesador 101) y se empaquetan, y se pueden integrar aún más en productos que incluyen, pero sin limitarse a, teléfonos móviles, teléfonos inteligentes, ordenadores portátiles, ordenadores ultraportátiles, tabletas, *ultrabooks*, ordenadores de escritorio, grabadoras de vídeo digital, descodificadores y cualquier otro dispositivo donde se usen circuitos integrados.

30 **[0039]** En un aspecto, los archivos de ordenador forman una estructura de diseño que incluye los circuitos descritos anteriormente y mostrados en las figuras en forma de trazados de diseño físico, esquemas, lenguaje de descripción de hardware (por ejemplo, Verilog, VHDL, etc.). Por ejemplo, una estructura de diseño puede ser un archivo de texto o una representación gráfica de un circuito como se describe anteriormente y se muestra en las figuras. El proceso de diseño sintetiza (o traduce) preferentemente los circuitos descritos a continuación en una lista de conectividad, donde la lista de conectividad es, por ejemplo, una lista de cables, transistores, puertas lógicas, circuitos de control, E/S, modelos, etc. que describe las conexiones a otros elementos y circuitos en un diseño de circuito integrado y registradas en al menos uno de los medios legibles por máquina. Por ejemplo, el medio puede ser un medio de almacenamiento tal como un CD, una memoria *flash* compacta, otra memoria *flash* o una unidad de disco duro. En otro aspecto, el hardware, los circuitos y el procedimiento descritos en el presente documento pueden estar configurados como archivos de ordenador que simulan la función de los circuitos descritos anteriormente y mostrados en las figuras cuando son ejecutados por un procesador. Estos archivos de ordenador se pueden usar en herramientas de simulación de circuitos, editores de esquemas u otras aplicaciones de software.

45 **[0040]** Las implementaciones de los aspectos divulgados en el presente documento también se pueden realizar de forma tangible (por ejemplo, en características legibles por ordenador tangibles de uno o más medios de almacenamiento legibles por ordenador como se menciona en el presente documento) como uno o más conjuntos de instrucciones ejecutables por una máquina que incluye una matriz de elementos lógicos (por ejemplo, un procesador, un microprocesador, un microcontrolador u otra máquina de estados finitos). El término "medio legible por ordenador" puede incluir cualquier medio que pueda almacenar o transferir información, incluyendo medios de almacenamiento volátiles, no volátiles, extraíbles y no extraíbles. Los ejemplos de medio legible por ordenador incluyen un circuito electrónico, un dispositivo de memoria semiconductor, una ROM, una memoria *flash*, una ROM borrable (EROM), un disquete u otro almacenamiento magnético, un CD-ROM/DVD u otro almacenamiento óptico, un disco duro o cualquier otro medio que se pueda usar para almacenar la información deseada, un medio de fibra óptica, un enlace de radiofrecuencia (RF) o cualquier otro medio que se pueda usar para transportar la información deseada y al que se pueda acceder. La señal de datos de ordenador puede incluir cualquier señal que se pueda propagar por un medio de transmisión tal como unos canales de red electrónica, fibras ópticas, aire, medios electromagnéticos, enlaces de RF, etc. Los segmentos de código se pueden descargar por medio de redes informáticas tales como Internet o una intranet. En cualquier caso, no se debe interpretar que el alcance de la presente divulgación está limitado por dichos aspectos.

60 **[0041]** La descripción previa de los aspectos divulgados se proporciona para permitir que un experto en la técnica haga o use los aspectos divulgados. Diversas modificaciones de estos aspectos resultarán fácilmente evidentes a los expertos en la técnica, y los principios definidos en el presente documento se pueden aplicar a otros aspectos sin apartarse del alcance de la divulgación. Por tanto, no se pretende limitar la presente divulgación a los aspectos mostrados en el presente documento, sino que se le ha de conceder el alcance más amplio posible consecuente con los principios y características novedosas, como se define en las reivindicaciones siguientes.

65

REIVINDICACIONES

1. Una memoria caché, que comprende:

5 una estructura de compresión de etiquetas que almacena una pluralidad de entradas de compresión de etiquetas (202), en la que al menos una de la pluralidad de entradas de compresión de etiquetas contiene un conjunto único de bits de orden superior compartidos por al menos dos etiquetas de memoria y se identifica mediante un valor de índice respectivo; y

10 una matriz de etiquetas (105) que almacena una pluralidad de entradas de matriz de etiquetas, en la que cada una de la pluralidad de entradas de matriz de etiquetas contiene un conjunto respectivo de bits de orden inferior de uno de las etiquetas de memoria y un valor de índice (204), en la que el valor de índice de cada entrada de matriz de etiquetas corresponde al valor de índice de la entrada de estructura de compresión de etiquetas que contiene los bits de orden superior de la etiqueta de memoria de la entrada de matriz de etiquetas respectiva, en la que cada una de la pluralidad de etiquetas de memoria comprende al menos una parte de una dirección de memoria respectiva de un elemento de datos almacenado en una matriz de datos.

2. La memoria caché de la reivindicación 1, que comprende además lógica configurada para presentar una indicación de un acierto en la matriz de etiquetas para una dirección de memoria de entrada:

25 presentando el valor de índice correspondiente a una primera entrada de compresión de etiquetas desde la estructura de compresión de etiquetas tras determinar que el conjunto de bits de orden superior de la etiqueta de memoria en la primera entrada de compresión de etiquetas coincide con un conjunto de bits de orden superior de la dirección de memoria de entrada;

30 presentando el valor de índice y el conjunto de bits de orden inferior de la etiqueta de memoria almacenada en una primera entrada de matriz de etiquetas tras determinar que el conjunto de bits de orden inferior de la dirección de memoria de entrada coincide con el conjunto de bits de orden inferior de la etiqueta de memoria en la primera entrada de matriz de etiquetas; y

determinando que el valor de índice presentado de la primera entrada de compresión de etiquetas coincide con el valor de índice presentado de la primera entrada de matriz de etiquetas.

3. La memoria caché de la reivindicación 1, que comprende además lógica configurada para presentar una indicación de fallo en la matriz de etiquetas para una dirección de memoria de entrada tras determinar al menos uno de:

40 que un conjunto de bits de orden superior de la dirección de memoria de entrada no coincide con ninguno de los conjuntos de bits de orden superior en la pluralidad de entradas de compresión de etiquetas;

45 que un conjunto de bits de orden inferior de la dirección de memoria de entrada no coincide con ninguno de los conjuntos de bits de orden inferior almacenados en la pluralidad de entradas de matriz de etiquetas; y

50 que un valor de índice presentado desde una entrada de matriz de etiquetas que almacena un conjunto de bits de orden inferior que coinciden con los bits de orden inferior de la dirección de memoria de entrada no coincide con un valor de índice de una entrada de compresión de etiquetas que almacena un conjunto de bits de orden superior que coinciden con los bits de orden superior de la dirección de memoria de entrada.

4. La memoria caché de la reivindicación 3, que comprende además lógica configurada para, como respuesta a presentar la indicación del fallo en la matriz de etiquetas:

55 recuperar unos datos almacenados en un siguiente nivel de una jerarquía de memoria;

almacenar los datos en una primera entrada de la matriz de datos;

60 almacenar un conjunto de bits de orden superior de la dirección de memoria de entrada en una segunda entrada de compresión de etiquetas en la estructura de compresión de etiquetas; y

65 almacenar: (i) un conjunto de bits de orden inferior de la dirección de memoria de entrada, y (ii) un valor de índice correspondiente a la segunda entrada de compresión de etiquetas en una segunda entrada de matriz de etiquetas de la matriz de etiquetas, en la que la segunda entrada de matriz de etiquetas corresponde a la primera entrada de la matriz de datos.

5. La memoria caché de la reivindicación 1, en la que las entradas en la estructura de compresión de etiquetas están configuradas además para almacenar uno o más de: (i) un identificador de espacio de direcciones (ASID) de un proceso, (ii) un conjunto de bits de seguridad, y (iii) un conjunto de bits de nivel de excepción.
- 5 6. La memoria caché de la reivindicación 1, en la que una primera entrada de matriz de etiquetas y una segunda entrada de matriz de etiquetas especifican cada una un valor de índice correspondiente a una primera entrada de compresión de etiquetas, en la que un conjunto de bits de orden inferior de la segunda entrada de matriz de etiquetas no coincide con un conjunto de bits de orden inferior de la primera entrada de matriz de etiquetas.
- 10 7. La memoria caché de la reivindicación 1, en la que la estructura de compresión de etiquetas comprende una memoria de contenido direccionable completamente asociativa, en la que la matriz de etiquetas comprende una parte de al menos una de: (i) una memoria caché de datos, (ii) una memoria intermedia de traducción adelantada, (iii) una memoria intermedia de relleno de línea, y (iv) una memoria caché de instrucciones.
- 15 8. La memoria caché de la reivindicación 1, en la que la memoria caché está dispuesta en un circuito integrado.
9. Un procedimiento, que comprende:
- 20 almacenar, en una primera entrada de compresión de etiquetas de una pluralidad de entradas de compresión de etiquetas (202) en una estructura de compresión de etiquetas, un conjunto de bits de orden superior compartidos por una primera etiqueta de memoria y una segunda etiqueta de memoria, en el que cada una de la pluralidad de entradas de compresión de etiquetas se identifica mediante un valor de índice respectivo; y
- 25 almacenar, en una primera entrada de matriz de etiquetas (105) de una pluralidad de entradas de matriz de etiquetas en una matriz de etiquetas, un conjunto de bits de orden inferior de la primera etiqueta de memoria y un valor de índice (204) para la primera entrada de matriz de etiquetas, en el que el valor de índice de la primera entrada de matriz de etiquetas corresponde al valor de índice de la primera entrada de compresión de etiquetas, en el que la primera etiqueta de memoria comprende al menos una parte de
- 30 una dirección de memoria de un elemento de datos almacenado en una matriz de datos, en el que una segunda entrada de matriz de etiquetas de la pluralidad de entradas de matriz de etiquetas especifica un valor de índice correspondiente a la primera entrada de compresión de etiquetas y almacena un conjunto de bits de orden inferior de la segunda etiqueta de memoria, en el que el conjunto de bits de orden inferior de la segunda etiqueta de memoria no coincide con el conjunto de bits de orden inferior de la primera etiqueta de memoria.
- 35 10. El procedimiento de la reivindicación 9, que comprende además presentar una indicación de un acierto en la matriz de etiquetas para una dirección de memoria de entrada:
- 40 presentando el valor de índice correspondiente a la primera entrada de compresión de etiquetas desde la estructura de compresión de etiquetas tras determinar que el conjunto de bits de orden superior de la etiqueta de memoria en la primera entrada de compresión de etiquetas coincide con un conjunto de bits de orden superior de la dirección de memoria de entrada;
- 45 presentando el valor de índice y el conjunto de bits de orden inferior de la etiqueta de memoria almacenada en la primera entrada de matriz de etiquetas tras determinar que el conjunto de bits de orden inferior de la dirección de memoria de entrada coincide con el conjunto de bits de orden inferior de la etiqueta de memoria en la primera entrada de matriz de etiquetas; y
- 50 determinando que el valor de índice presentado de la primera entrada de compresión de etiquetas coincide con el valor de índice presentado de la primera entrada de matriz de etiquetas.
11. El procedimiento de la reivindicación 9, en el que cada una de la pluralidad de entradas de compresión de etiquetas en la estructura de compresión de etiquetas está configurada para almacenar un conjunto único de bits de orden superior compartidos por una pluralidad respectiva de etiquetas de memoria, en el que cada una de la pluralidad de entradas de matriz de etiquetas está configurada para almacenar un conjunto de bits de orden inferior de una etiqueta de memoria respectiva y un valor de índice, en el que el valor de índice de cada entrada de matriz de etiquetas corresponde al valor de índice de la entrada de compresión de etiquetas que contiene los bits de orden superior de la etiqueta de memoria de la entrada de matriz de etiquetas respectiva, comprendiendo además
- 55 el procedimiento presentar una indicación de fallo en la matriz de etiquetas para una dirección de memoria de entrada tras determinar al menos uno de:
- 60 que un conjunto de bits de orden superior de la dirección de memoria de entrada no coincide con ninguno de los conjuntos de bits de orden superior en la pluralidad de entradas de compresión de etiquetas;
- 65

que un conjunto de bits de orden inferior de la dirección de memoria de entrada no coincide con ninguno de los conjuntos de bits de orden inferior almacenados en la pluralidad de entradas de matriz de etiquetas;
y

5 que un valor de índice presentado desde una entrada de matriz de etiquetas que almacena un conjunto de bits de orden inferior que coinciden con los bits de orden inferior de la dirección de memoria de entrada no coincide con un valor de índice de una entrada de compresión de etiquetas que almacena un conjunto de bits de orden superior que coinciden con los bits de orden superior de la dirección de memoria de entrada.

10

12. El procedimiento de la reivindicación 11, que comprende además:

como repuesta a presentar la indicación del fallo en la matriz de etiquetas:

15

recuperar unos datos almacenados en un siguiente nivel de una jerarquía de memoria;

almacenar los datos en una primera entrada de la matriz de datos;

20

almacenar un conjunto de bits de orden superior de la dirección de memoria de entrada en una segunda entrada de compresión de etiquetas en la estructura de compresión de etiquetas; y

25

almacenar: (i) un conjunto de bits de orden inferior de la dirección de memoria de entrada, y (ii) un valor de índice correspondiente a la segunda entrada de compresión de etiquetas en una segunda entrada de matriz de etiquetas de la matriz de etiquetas, en la que la segunda entrada de matriz de etiquetas corresponde a la primera entrada de la matriz de datos.

30

13. El procedimiento de la reivindicación 9, en el que las entradas en la estructura de compresión de etiquetas están configuradas además para almacenar uno o más de: (i) un identificador de espacio de direcciones (ASID) de un proceso, (ii) un conjunto de bits de seguridad, y (iii) un conjunto de bits de nivel de excepción.

35

14. El procedimiento de la reivindicación 9, en el que la estructura de compresión de etiquetas comprende una memoria de contenido direccionable completamente asociativa, en el que la matriz de etiquetas comprende una parte de al menos una de: (i) una memoria caché de datos, (ii) una memoria intermedia de traducción adelantada, (iii) una memoria intermedia de relleno de línea, y (iv) una memoria caché de instrucciones.

15. Un medio no transitorio legible por ordenador que almacena instrucciones que, cuando se ejecutan mediante un procesador, hacen que el procesador realice las etapas de procedimiento de las reivindicaciones 9-14.

100 ↘

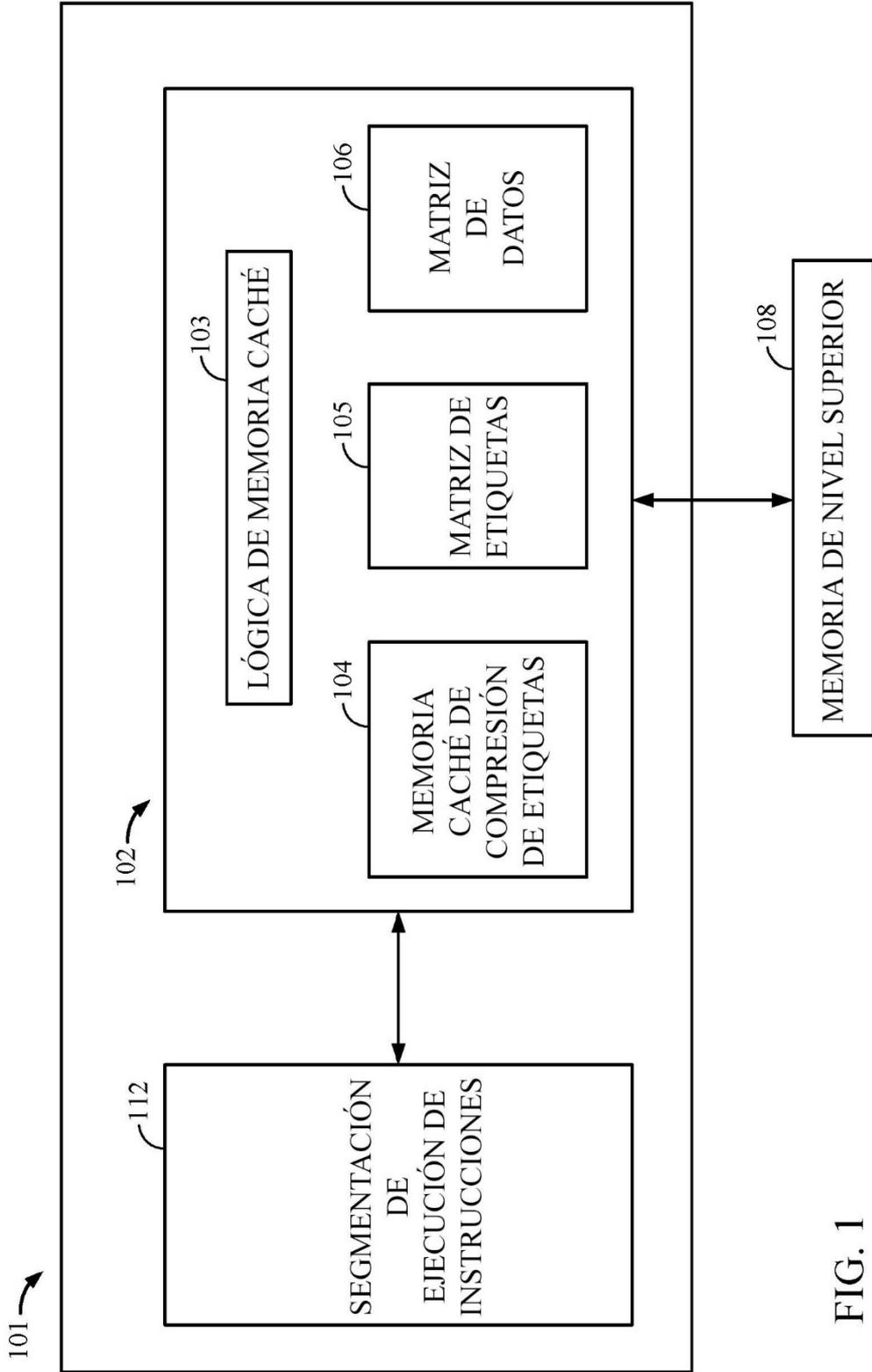


FIG. 1

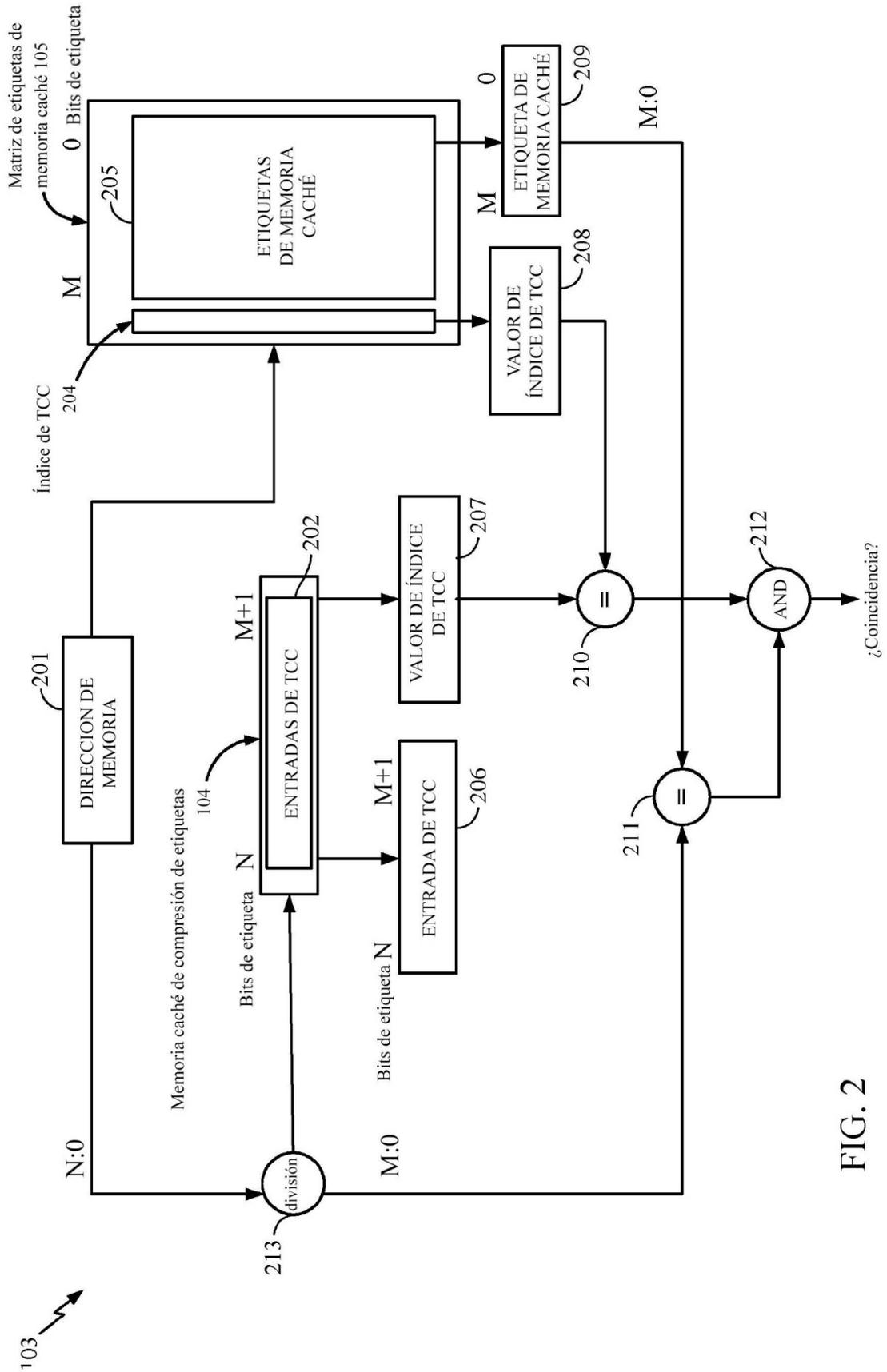


FIG. 2

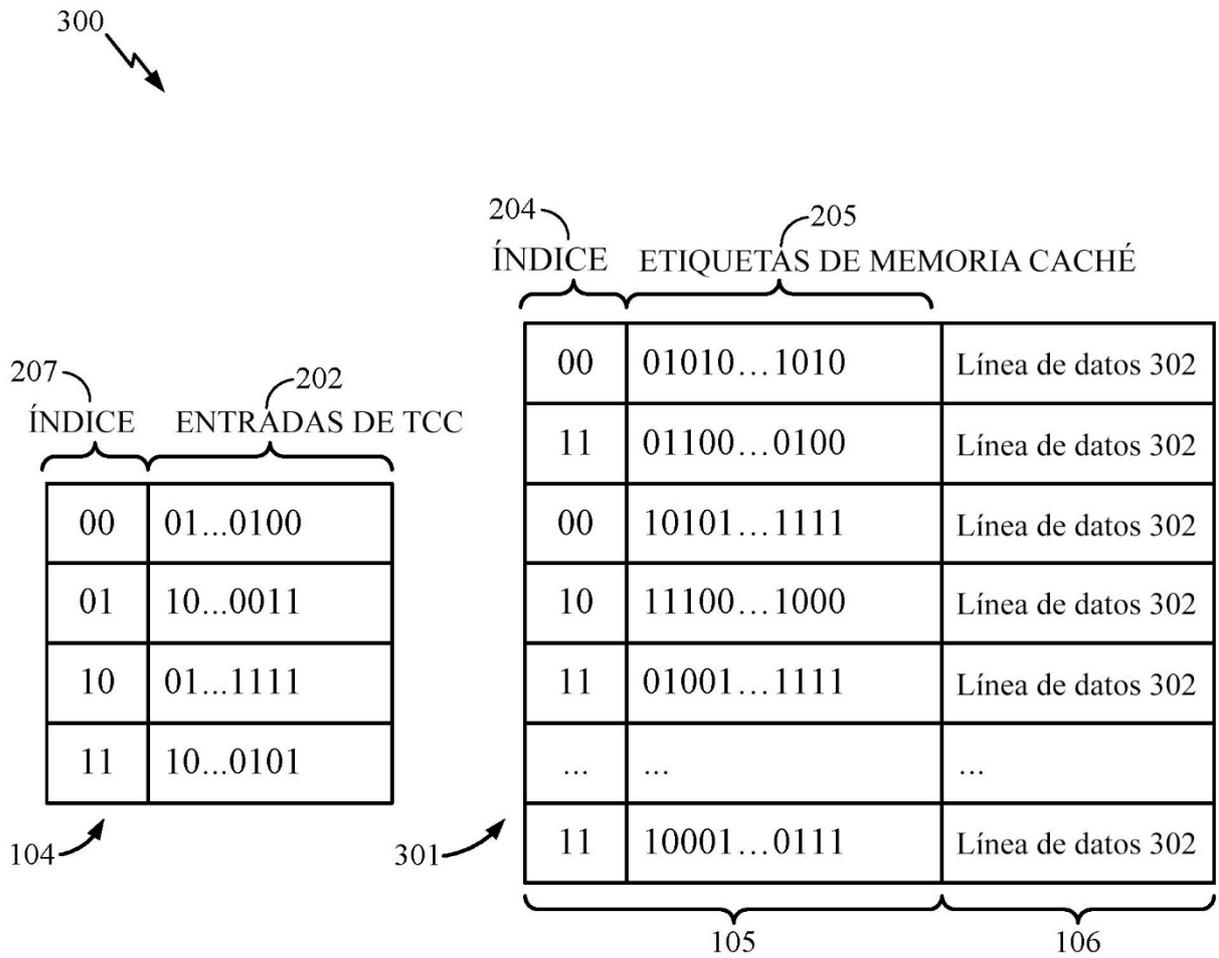


FIG. 3

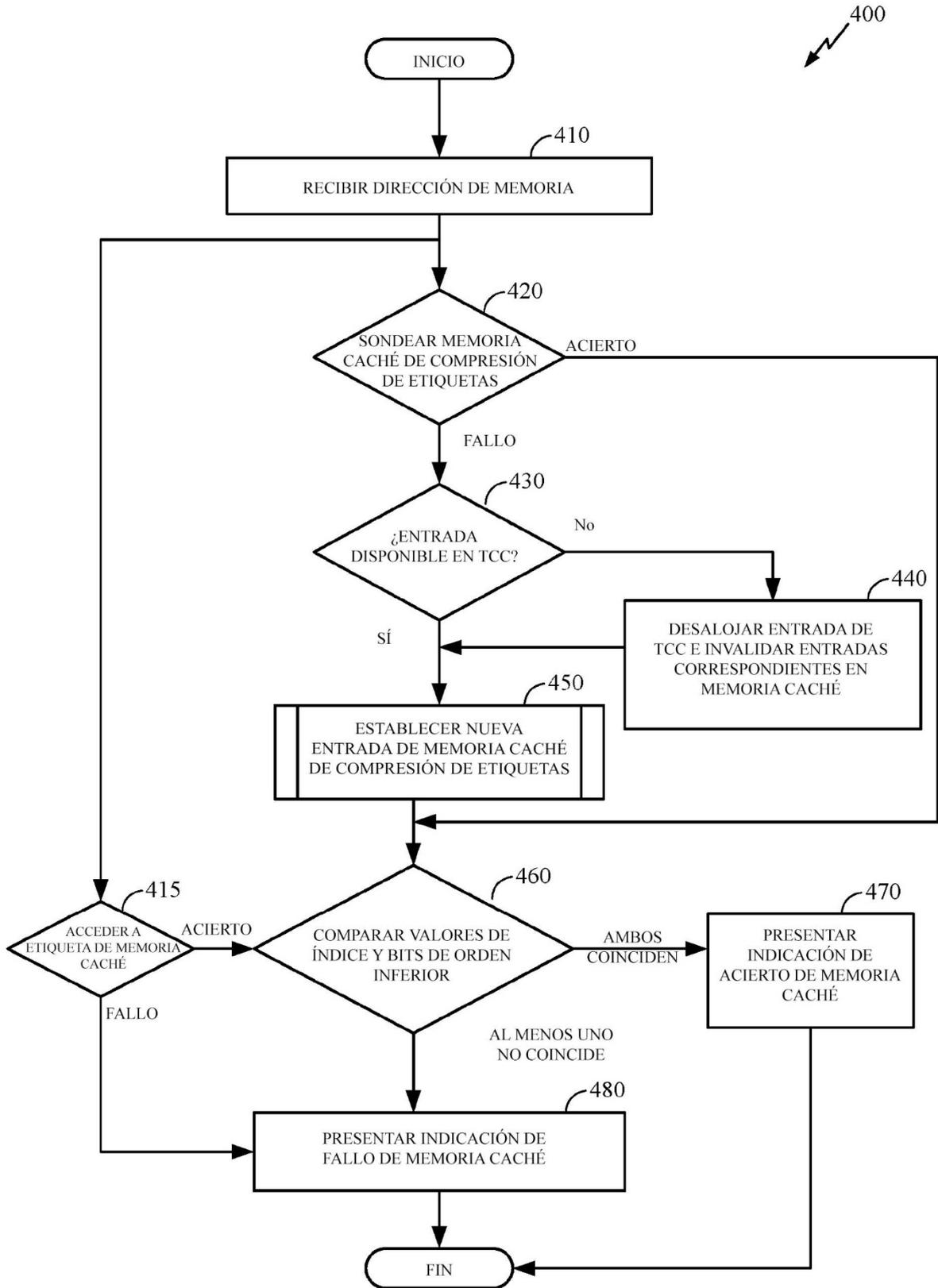


FIG. 4

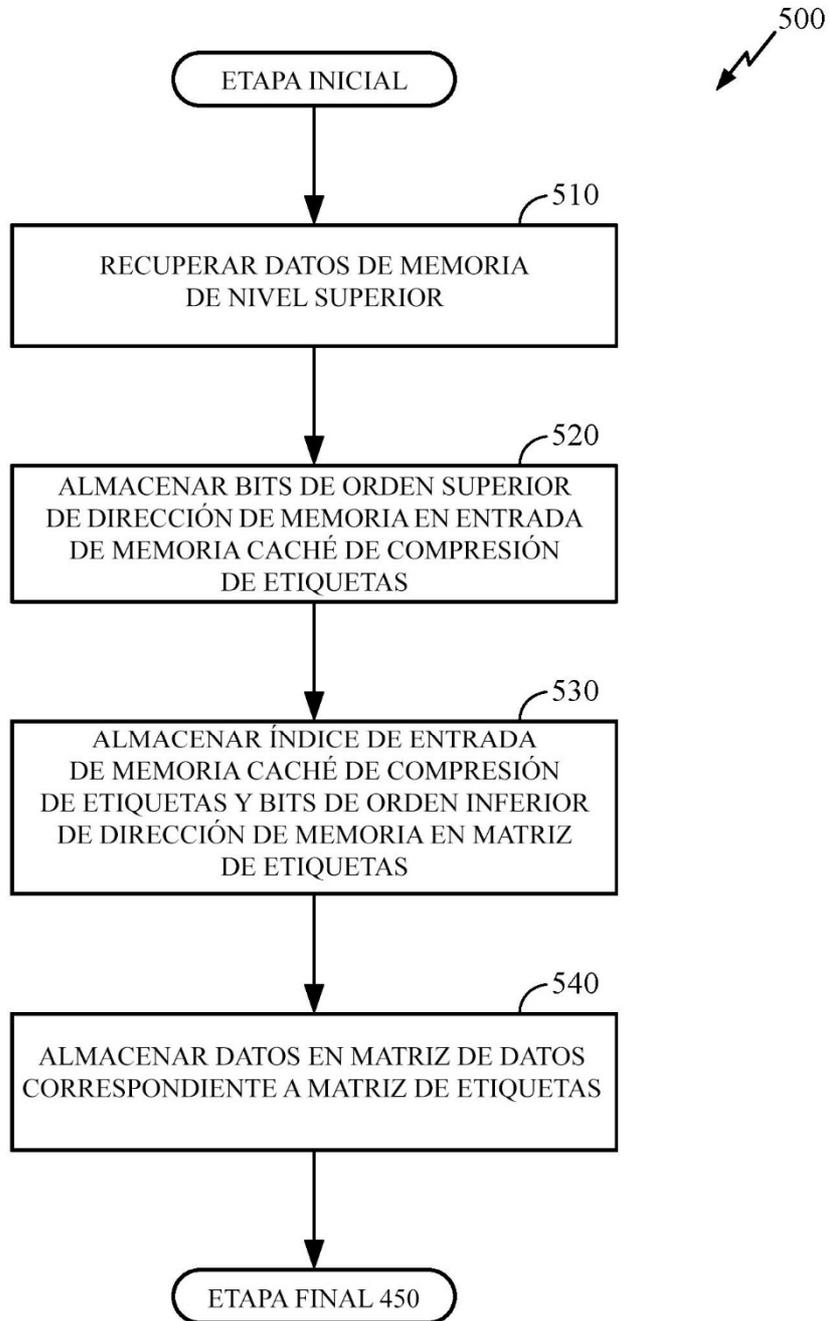


FIG. 5

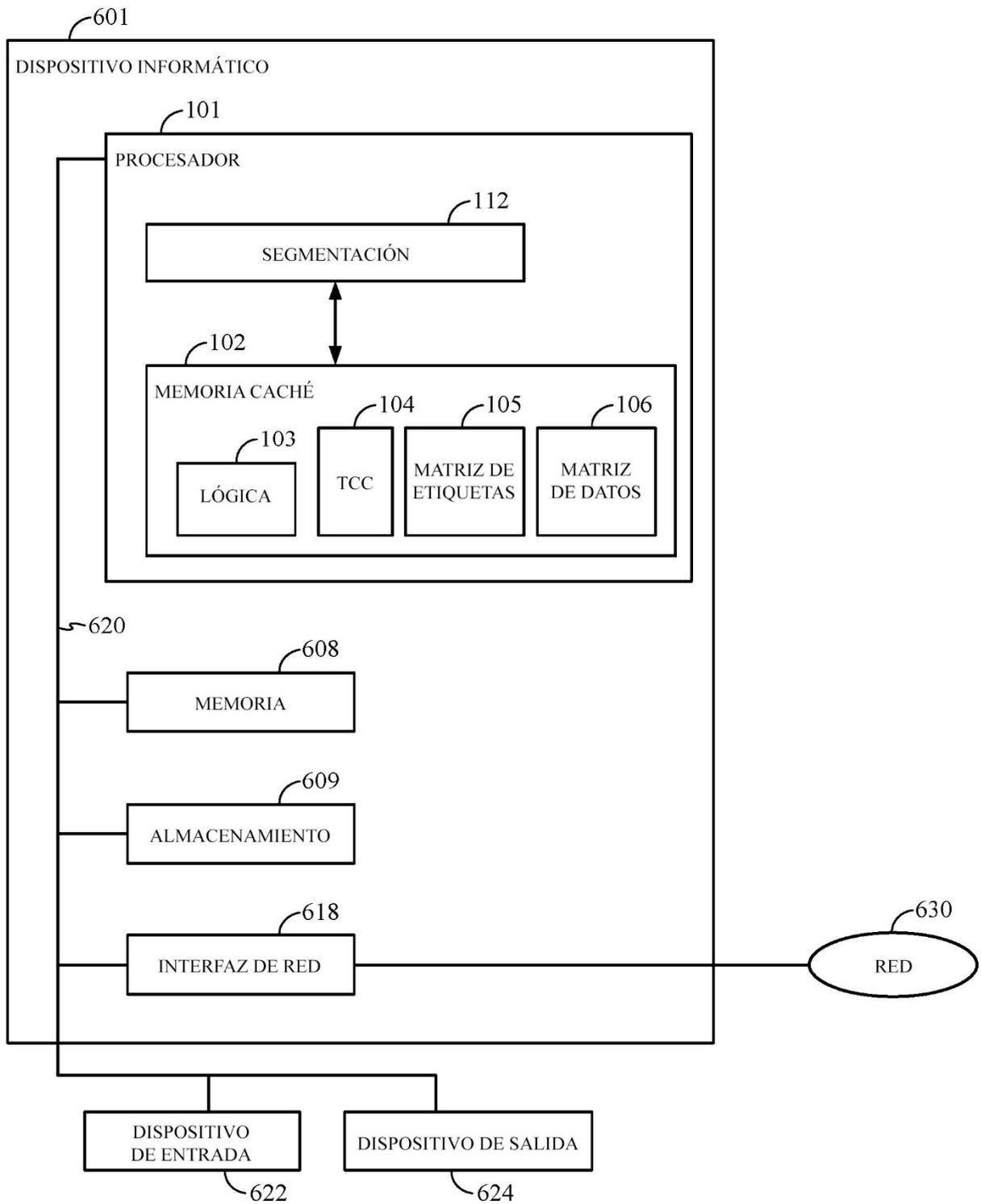


FIG. 6