(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 903 001**

(51) Int. CI.:

G06F 12/02 (2006.01)
G06F 11/07 (2006.01)
G06F 21/60 (2013.01)
G06F 12/109 (2006.01)
G06F 12/00 (2006.01)
G06F 12/14 (2006.01)
G06F 9/38 (2008.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **22.11.2016 PCT/US2016/063211**

(87) Fecha y número de publicación internacional: **29.06.2017 WO17112234**

(96) Fecha de presentación y número de la solicitud europea: **22.11.2016 E 16879720 (7)**

(97) Fecha y número de publicación de la concesión europea: **20.10.2021 EP 3394757**

(54) Título: **Aparatos y métodos de hardware para detección de corrupción de memoria**

(30) Prioridad:

21.12.2015 US 201514977354

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
30.03.2022

(73) Titular/es:

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

(72) Inventor/es:

STARK, TOMER;
GABOR, RON;
NUZMAN, JOSEPH;
SADE, RAANAN y
BIGBEE, BRYANT E.

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Aparatos y métodos de hardware para detección de corrupción de memoria

CAMPO TÉCNICO

La divulgación se refiere, en general, a la electrónica, y, más específicamente, una realización de la divulgación se refiere a un procesador de hardware con hardware de detección de corrupción de memoria.

ANTECEDENTES

Un procesador, o un conjunto de procesadores, ejecuta instrucciones de un conjunto de instrucciones, por ejemplo, la arquitectura de conjunto de instrucciones (ISA). El conjunto de instrucciones es la parte de la arquitectura informática relacionada con la programación, y, en general, incluye los tipos de datos nativos, instrucciones, arquitectura de registros, modos de direccionamiento, arquitectura de memoria, manejo de interrupciones y de excepciones y entradas y salidas externas (E/S). Debería observarse que, el término instrucción, en el presente documento, puede hacer referencia a una macroinstrucción, por ejemplo, una instrucción que se proporciona al procesador para su ejecución, a una microinstrucción, por ejemplo, una instrucción que resulta de un decodificador del procesador que decodifica macroinstrucciones.

El documento US 2013/0013843 A1 pertenece al almacenamiento de datos de versión de memoria.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La presente divulgación se ilustra a modo de ejemplo y no como limitación en las figuras de los dibujos adjuntos, en los que referencias similares indican elementos similares y en los que:

La **Figura 1** ilustra un procesador de hardware de acuerdo con las realizaciones de la divulgación.

La **Figura 2** ilustra una detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 3** ilustra un formato de puntero con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 4** ilustra un formato de puntero con un campo de dirección y un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 5** ilustra un formato de puntero con un campo de dirección, un campo de espacio de detección de corrupción de memoria y un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 6** ilustra formatos de datos de registros para la detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 7** ilustra un sistema de detección de corrupción de memoria (MCD) con una unidad de gestión de memoria de acuerdo con las realizaciones de la divulgación.

La **Figura 8** ilustra una unidad de gestión de memoria de acuerdo con las realizaciones de la divulgación.

La **Figura 9** ilustra un formato de puntero con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 10** ilustra un formato de puntero con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 11** ilustra un formato de puntero con un campo de dirección, un campo de espacio de detección de corrupción de memoria y un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 12A** ilustra un espacio de direcciones lineal de acuerdo con las realizaciones de la divulgación.

La **Figura 12B** ilustra una vista de una porción del espacio de direcciones lineal en la Figura 12A de acuerdo con las realizaciones de la divulgación.

La **Figura 12C** ilustra una vista de la porción del espacio de direcciones lineal de la Figura 12B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 13** ilustra un formato de puntero con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 14** ilustra un formato de puntero con un campo de dirección, un campo de espacio de detección de corrupción de memoria y un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 15A** ilustra un espacio de direcciones lineal de acuerdo con las realizaciones de la divulgación.

La **Figura 15B** ilustra una vista de una porción del espacio de direcciones lineal en la Figura 15A de acuerdo con las realizaciones de la divulgación.

La **Figura 15C** ilustra una vista de la porción del espacio de direcciones lineal de la Figura 12B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 16** ilustra un formato de puntero con un campo de dirección, un campo de espacio de detección de corrupción de memoria y un campo de valor de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 17A** ilustra un espacio de direcciones lineal de acuerdo con las realizaciones de la divulgación.

La **Figura 17B** ilustra una vista de una porción del espacio de direcciones lineal en la Figura 17A de acuerdo con las realizaciones de la divulgación.

La **Figura 17C** ilustra una vista de la porción del espacio de direcciones lineal de la Figura 17B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación.

La **Figura 18** ilustra un diagrama de flujo de acuerdo con las realizaciones de la divulgación.

La **Figura 19A** es un diagrama de bloques que ilustra tanto una canalización en orden ilustrativa como un renombrado de registro ilustrativo, una canalización de emisión/ejecución fuera de orden de acuerdo con las realizaciones de la divulgación.

La **Figura 19B** es un diagrama de bloques que ilustra tanto una realización ilustrativa de un núcleo de arquitectura en orden como un renombrado de registro ilustrativo, un núcleo de arquitectura de emisión/ejecución fuera de orden que ha de incluirse en un procesador de acuerdo con las realizaciones de la divulgación.

La **Figura 20A** es un diagrama de bloques de un único núcleo de procesador, junto con su conexión a la red de interconexión en la matriz y con su subconjunto local de la caché de nivel 2 (L2), de acuerdo con las realizaciones de la divulgación.

La **Figura 20B** es una vista ampliada de parte del núcleo de procesador de la Figura 20A de acuerdo con unas realizaciones de la divulgación.

La **Figura 21** es un diagrama de bloques de un procesador que puede tener más de un núcleo, puede tener un controlador de memoria integrado y puede tener gráficos integrados de acuerdo con las realizaciones de la divulgación.

La **Figura 22** es un diagrama de bloques de un sistema de acuerdo con una realización de la presente divulgación.

La **Figura 23** es un diagrama de bloques de un sistema ilustrativo más específico de acuerdo con una realización de la presente divulgación.

La **Figura 24,** mostrada es un diagrama de bloques de un segundo sistema ilustrativo más específico de acuerdo con una realización de la presente divulgación.

La **Figura 25,** mostrada es un diagrama de bloques de un sistema en un chip (SoC) de acuerdo con una realización de la presente divulgación.

La **Figura 26** es un diagrama de bloques que contrasta el uso de un convertidor de instrucción de software para convertir instrucciones binarias en un conjunto de instrucciones de origen a instrucciones binarias en un conjunto de instrucciones objetivo de acuerdo con las realizaciones de la divulgación.

DESCRIPCIÓN DETALLADA

La invención se define está definida las reivindicaciones independientes. En la siguiente descripción, se exponen

numerosos detalles específicos. Sin embargo, se entiende que las realizaciones de la divulgación pueden ponerse en práctica sin estos detalles específicos. En otras instancias, no se han mostrado en detalle circuitos, estructuras y técnicas bien conocidos para no oscurecer la comprensión de esta descripción.

Las referencias en la memoria descriptiva a "una realización", "una realización", "una realización ilustrativa", etc., indican que la realización descrita puede incluir un rasgo, estructura o característica particular, pero cada realización puede no incluir necesariamente el rasgo, estructura o característica particular. Además, tales expresiones no hacen referencia necesariamente a la misma realización. Además, cuando se describe un rasgo, estructura o característica particular en conexión con una realización, se afirma que está dentro del conocimiento de un experto en la materia determinar tal rasgo, estructura o característica en conexión con otras realizaciones ya se hayan descrito explícitamente o no.

Un (por ejemplo, hardware) procesador (por ejemplo, que tiene uno o más núcleos) puede ejecutar instrucciones para operar en datos, por ejemplo, realizar funciones aritméticas, lógicas u otras. Un procesador de hardware puede acceder a datos en una memoria (por ejemplo, un dispositivo de almacenamiento de datos). En una realización, un procesador de hardware es un cliente que solicita acceso a (por ejemplo, carga o almacena) datos y la memoria es un servidor que contiene los datos. En una realización, un ordenador incluye un procesador de hardware que solicita acceso a (por ejemplo, carga o almacena) datos y la memoria es local al ordenador. La memoria puede dividirse en líneas separadas (por ejemplo, una o más líneas de caché) de datos, por ejemplo, que pueden gestionarse como una unidad a efectos de coherencia. En ciertas realizaciones, un (por ejemplo, dato) puntero (por ejemplo, una dirección) es un valor que hace referencia (por ejemplo, apunta) a la ubicación de los datos, por ejemplo, un puntero puede ser una (por ejemplo, lineal) dirección y los datos pueden almacenarse en esa dirección (por ejemplo, lineal). En ciertas realizaciones, la memoria puede dividirse en múltiples líneas y cada línea puede tener su propia dirección (por ejemplo, única). Por ejemplo, una línea de memoria puede incluir almacenamiento para 512 bits, 256 bits, 128 bits, 64 bits, 32 bits, 16 bits u 8 bits de datos.

En ciertas realizaciones, la corrupción de memoria (por ejemplo, por un atacante) puede producirse por un acceso fuera de límite (por ejemplo, acceso a memoria usando la dirección de base de un bloque de memoria y un desplazamiento que supera el tamaño asignado del bloque) o por un puntero colgante (por ejemplo, un puntero que hace referencia a un bloque de memoria (por ejemplo, memoria intermedia) que se ha desasignado).

Ciertas realizaciones del presente documento pueden utilizar hardware y/o métodos de detección de corrupción de memoria (MCD), por ejemplo, para evitar un acceso fuera de límite o un acceso con un puntero colgante.

Volviendo ahora a las figuras, la **Figura 1** ilustra un procesador de hardware 100 de acuerdo con las realizaciones de la divulgación. El procesador de hardware 100 representado incluye una unidad de decodificación de hardware 102 para decodificar una instrucción, por ejemplo, una instrucción que es para solicitar acceso a un bloque de una memoria 110 a través de un puntero 105 al bloque de la memoria 110. El puntero 105 puede ser un operando de la instrucción. La unidad de ejecución de hardware 104 representada es para ejecutar la instrucción decodificada, por ejemplo, la instrucción decodificada que es para solicitar acceso al bloque de la memoria 110 a través de un puntero 105 (por ejemplo, que tiene un valor de la (por ejemplo, lineal) dirección 114) al bloque de la memoria 110. En una realización, un bloque de datos es una única línea de datos. En una realización, un bloque de datos son múltiples líneas de datos. Por ejemplo, un bloque de memoria puede ser las líneas 1 y 2 de datos de la (por ejemplo, lineal o física) memoria direccionable 112 de la memoria 110 que incluye un puntero 105 (por ejemplo, que tiene un valor de dirección 114) a una (por ejemplo, la primera) línea (por ejemplo, la línea 1). Ciertas realizaciones pueden tener una memoria de un tamaño total de X número de líneas.

El procesador de hardware 100 puede incluir uno o más registros 108, por ejemplo, un registro de control o registros de configuración, tales como, pero sin limitación, un registro específico de modelo (MSR) u otros registros. En una realización, un valor almacenado en un registro de control es para cambiar (por ejemplo, controlar) características seleccionables, por ejemplo, características del procesador de hardware.

El procesador de hardware 100 incluye un acoplamiento (por ejemplo, conexión) a una memoria 110. La memoria 110 puede ser una memoria local al procesador de hardware (por ejemplo, memoria de sistema). La memoria 110 puede ser una memoria separada del procesador de hardware, por ejemplo, una memoria de un servidor. Obsérvese que las figuras del presente documento pueden no representar todas las conexiones de comunicación de datos. Una persona normalmente versada en la materia apreciará que esto no ensombrece ciertos detalles en las figuras. Obsérvese que una flecha de dos puntas en las figuras puede no requerir una comunicación bidireccional, por ejemplo, puede indicar una comunicación unidireccional (por ejemplo, a o desde ese componente o dispositivo). Puede utilizarse cualquiera o todas las combinaciones de rutas de comunicaciones en ciertas realizaciones del presente documento.

El procesador de hardware 100 incluye una unidad de gestión 106 de memoria, por ejemplo, para realizar y/o controlar el acceso (por ejemplo, por la unidad 104 de ejecución) a (por ejemplo, memoria direccionable 112 de) la memoria 110. En una realización, el procesador de hardware incluye una conexión a la memoria. Adicionalmente o como alternativa, la unidad de gestión de memoria 106 puede incluir una conexión a (por ejemplo, la memoria direccionable 112 y/o la tabla de detección de corrupción de memoria 116 de) la memoria 110.

Ciertas realizaciones pueden incluir características de detección de corrupción de memoria (MCD), por ejemplo, en una unidad de gestión de memoria. Ciertas realizaciones pueden utilizar un valor de detección de corrupción de memoria (MCD) en cada puntero y un valor de MCD guardado correspondiente (por ejemplo, coincidente) en la memoria para la memoria a la que se está apuntando, por ejemplo, guardado como metadatos (por ejemplo, datos que describen otros datos) para cada bloque de datos al que se está apuntando por el puntero. Un valor de MCD puede ser una secuencia de bits, por ejemplo, un 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 bits, etc. En una realización, un sistema o procesador de procesamiento de hardware de detección de corrupción de memoria (MCD) (por ejemplo, una unidad de gestión de memoria del procesador o del sistema) es para validar punteros mediante instrucciones de las aplicaciones que están siendo ejecutadas por el sistema o procesador de procesamiento que solicita acceso a la memoria.

Ciertas realizaciones del presente documento (por ejemplo, de ajustes de un circuito de MMU) utilizan uno o más de los siguientes atributos de detección de corrupción de memoria: MCD activada (por ejemplo, para conectar o desconectar la característica de MCD), posición de MCD (por ejemplo, para definir la posición o posiciones de bit de los valores de MCD (metadatos) en punteros), espacio protegido de MCD, por ejemplo, un prefijo en las posiciones del bit más significativo del puntero (por ejemplo, para definir el rango de direcciones lineal que ha de ser protegido por la arquitectura) y base de directorio de MCD (por ejemplo, para apuntar a la tabla (por ejemplo, directorio) del valor de memoria MCD (por ejemplo, metadatos)).

Ciertas realizaciones del presente documento permiten la colocación flexible de valores de MCD (por ejemplo, bits de metadatos) en un puntero, por ejemplo, no limitado a los bits más significativos. Ciertas realizaciones del presente documento permiten llevar a cabo un espacio de direcciones más pequeño (por ejemplo, una reducción en la sobrecarga de espacio de direcciones lineal) y/o de escalamiento (por ejemplo, 64 bits) para modos de paginación. Ciertas realizaciones del presente documento permiten la protección con MCD de únicamente un subconjunto de (por ejemplo, parte de) la memoria a través de una selección de espacio protegido (por ejemplo, seleccionar la dirección o direcciones que hay que proteger con MCD y no proteger las otras direcciones con MCD).

En la Figura 1, la unidad de gestión 106 de memoria (por ejemplo, la unidad de gestión de memoria de hardware) del procesador de hardware 100 puede recibir una solicitud para acceder (por ejemplo, cargar o almacenar) a la memoria 110 (por ejemplo, la memoria direccionable 112). La solicitud puede incluir un puntero 105 (por ejemplo, que tiene un valor de dirección 114), por ejemplo, pasado como un operando (por ejemplo, directo o indirecto) de una instrucción. El puntero puede incluir, como una porción (por ejemplo, campo) del mismo, un valor de detección de corrupción de memoria (MCD). Un bloque de línea múltiple de memoria puede incluir un valor de MCD para ese bloque, por ejemplo, un mismo valor de MCD para todas las líneas en ese bloque, y el valor de MCD para ese bloque es para hacer corresponder (por ejemplo, coincidir) el valor de MCD dentro del puntero a ese bloque. La unidad de gestión 106 de memoria (por ejemplo, un circuito de la misma) puede realizar una comprobación de validación de MCD (por ejemplo, para permitir o denegar el acceso) de acuerdo con esta divulgación.

La **Figura 2** ilustra una detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación. Un sistema de procesamiento o procesador puede mantener una tabla de metadatos (por ejemplo, la tabla de MCD 116 o la tabla de MCD 216) que almacena un valor de MCD (por ejemplo, un identificador de MCD) para cada línea de una pluralidad de líneas de un bloque de memoria, por ejemplo, líneas de un tamaño predefinido (por ejemplo, 64 bytes, aunque pueden utilizarse otros tamaños de línea). En una realización, cuando un bloque de memoria está asignado a un objeto de memoria (por ejemplo, recién creado), se genera un valor de MCD único y se asocia con la una o más líneas de ese bloque. El valor de MCD puede almacenarse en una o más entradas de tabla (por ejemplo, metadatos) que corresponden al bloque de memoria que está asignado para el objeto de memoria (por ejemplo, recién creado). En la Figura 2, las líneas de datos 1 y 2 se representan como asignadas al objeto 1 (por ejemplo, como un bloque de datos) y un valor de MCD (mostrado en este punto como "2") está asociado en la tabla de MCD 216, por ejemplo, de manera que cada línea de datos está asociada con una entrada en la tabla de MCD 216 que indica el valor de MCD (por ejemplo, "2") para ese bloque. En la Figura 2, las líneas de datos 3-5 se representan como asignadas al objeto 2 (por ejemplo, como un bloque de datos) y un valor de MCD (mostrado en este punto como "7") está asociado en la tabla de MCD 216, por ejemplo, de manera que cada línea de datos está asociada con una entrada en la tabla de MCD 216 que indica el valor de MCD (por ejemplo, "7") para ese bloque. En una realización, la tabla de MCD 216 tiene un campo de valor de MCD para cada línea correspondiente de la memoria direccionable 112.

En ciertas realizaciones, el valor de MCD generado, o un valor diferente que corresponde o se mapea al valor de MCD generado para el bloque de datos, se almacena en uno o más bits de un puntero, por ejemplo, un puntero que es devuelto por la rutina de asignación de memoria a la aplicación que solicitó la asignación de memoria. En la Figura 2, el puntero 215 incluye un campo de valor de MCD 215A con el valor de MCD ("2") y un campo de dirección 215B con un valor para la (por ejemplo, lineal) dirección de (por ejemplo, la primera línea de) el bloque de memoria del objeto 1. En la Figura 2, el puntero 217 incluye un campo de valor de MCD 217A con el valor de MCD ("7") y un campo de dirección 217B con un valor para la (por ejemplo, lineal) dirección de (por ejemplo, la primera línea de) el bloque de memoria del objeto 2.

En ciertas realizaciones, en respuesta a recibir una instrucción de acceso a memoria (por ejemplo, según se determina

a partir de un código de operación de la instrucción o un intento de acceso a memoria), el sistema de procesamiento o procesador compara el valor de MCD recuperado de la tabla de MCD (por ejemplo, para el bloque de datos al que se va a acceder) al valor de MCD a partir del (por ejemplo, extraído de) puntero especificado por la instrucción de acceso a memoria. En una realización, cuando coinciden los dos valores de MCD, se concede el acceso al bloque de datos. En una realización, cuando no coinciden los dos valores de MCD, se deniega el acceso al bloque de datos, por ejemplo, puede generarse un error de página. En una realización, la tabla de MCD (por ejemplo, la tabla de MCD 116 o la tabla de MCD 216) se encuentra en el espacio de direcciones lineal de la memoria. En una realización, el circuito y/o lógica para realizar la comprobación de validación de MCD (por ejemplo, en la unidad de gestión de memoria (MMU) 106) es para acceder a la memoria, pero las otras porciones del procesador (por ejemplo, la unidad de ejecución) son para no acceder a la memoria a menos que se pase la comprobación de validación de MCD (por ejemplo, una coincidencia sea verdadera). En una realización, una solicitud de acceso a un bloque de memoria es una instrucción de carga. En una realización, una solicitud de acceso a un bloque de memoria es una instrucción de almacenamiento.

En la Figura 2, una solicitud para acceder al bloque del objeto 1 en la memoria direccionable 212 de la memoria 210 puede iniciar (por ejemplo, mediante una unidad de gestión de memoria) la lectura del puntero 215 para el valor de MCD ("2") en el campo de valor de MCD 215A y la (por ejemplo, lineal) dirección en el campo de dirección 215B. El sistema (por ejemplo, el procesador) puede realizar a continuación una comprobación de validación, por ejemplo, cargando el valor de MCD de la tabla de MCD 216 en la memoria 210 para la línea o líneas a las que se accederá y comparándolas al valor de MCD en el puntero 215 a esa línea o líneas. En ciertas realizaciones, si el sistema determina que los valores de MCD coinciden (por ejemplo, siendo ambos "2" en este ejemplo), entonces el sistema permite el acceso (por ejemplo, de lectura y/o escritura) a la memoria (por ejemplo, únicamente a las líneas de datos 1 o 1 y 2). En ciertas realizaciones, si no hay coincidencia, se deniega la solicitud (por ejemplo, la instrucción solicitante puede fallar). En una realización, la solicitud para acceder al bloque del objeto 1 puede incluir una solicitud para acceder a todas las líneas en el objeto (líneas de datos 1 y 2), y el sistema puede realizar una comprobación de validación en la línea de datos 1 (por ejemplo, como se ha analizado anteriormente) y puede realizar una segunda comprobación de validación en la línea de datos 2. Por ejemplo, el sistema (por ejemplo, el procesador) puede realizar una comprobación de validación en la línea 2 cargando el valor de MCD de la tabla de MCD 216 en la memoria 210 para la línea 2 (por ejemplo, el valor de MCD "2") y comparar ese al valor de MCD en el puntero 215. En ciertas realizaciones, si el sistema determina que coinciden los valores de MCD (por ejemplo, siendo ambos valores de MCD "2" en este ejemplo), entonces el sistema permite el acceso (por ejemplo, de lectura y/o escritura) a la memoria (por ejemplo, la línea de datos 2).

La **Figura 3** ilustra un formato de puntero 300 con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) 301 de acuerdo con las realizaciones de la divulgación. En una realización, un campo de dirección 301 contiene una dirección lineal de la línea de datos que almacena los datos a los que se va a acceder. Las posiciones de bits ilustradas son ejemplos. El tamaño de puntero de 64 bits es un ejemplo.

La **Figura 4** ilustra un formato de puntero 400 con un campo de dirección 401 y un campo de valor de detección de corrupción de memoria (MCD) 403 de acuerdo con las realizaciones de la divulgación. En una realización, el campo de valor de MCD 403 es para almacenar el valor de MCD para el puntero, por ejemplo, cuando el valor de MCD y la dirección para el puntero son devueltos por la rutina de asignación de memoria a la aplicación que solicitó la asignación de memoria. El campo de valor de MCD 403 puede estar ubicado en cualquier posición (por ejemplo, ubicación) en el puntero, por ejemplo, no está fijo en una posición. El campo de valor de MCD 403 puede tener un tamaño de 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 bits, etc. En una realización, el campo de valor de MCD no se encuentra en los 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, etc., bits más significativos o bits menos significativos del puntero. En una realización, la posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre una primera ubicación y una segunda ubicación diferente. En una realización, la posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre una primera ubicación, una segunda ubicación diferente y una tercera ubicación diferente. En una realización, la posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre una primera ubicación, una segunda ubicación diferente, una tercera ubicación diferente y una cuarta ubicación diferente, etc. En una realización, una pluralidad de diferentes ubicaciones incluye una o más posiciones de bits que no se solapan. En una realización, una pluralidad de diferentes ubicaciones incluye una o más posiciones de bits que se solapan.

La **Figura 5** ilustra un formato de puntero 500 con un campo de dirección 501, un campo de espacio protegido de detección de corrupción de memoria (MCD) 505, y un campo de valor de detección de corrupción de memoria (MCD) 503 de acuerdo con las realizaciones de la divulgación. En una realización, el campo de dirección 501 es una dirección lineal de la línea de datos que almacena los datos a los que se va a acceder. En una realización, el campo de valor de MCD 503 es para almacenar el valor de MCD para el puntero. En una realización, el campo de espacio protegido de MCD 505 almacena un valor para indicar si el puntero es una región de la memoria que ha de tener una comprobación de validación de MCD realizada.

En una realización, puede seleccionarse la posición del valor de detección de corrupción de memoria en cada puntero, por ejemplo, en la fabricación, en la configuración o por una aplicación (por ejemplo, software, tal como, pero sin limitación, un sistema operativo), por ejemplo, durante la activación de una característica de MCD. La posición puede

establecerse en el procesador de hardware, por ejemplo, escribiendo en un registro de control (o configuración). En una realización, puede seleccionarse el espacio protegido de MCD (por ejemplo, qué subconjunto o subconjuntos de la memoria está protegido por las características de MCD), por ejemplo, en la fabricación, en la configuración o por una aplicación (por ejemplo, software, tal como, pero sin limitación, un sistema operativo), por ejemplo, durante la activación de una característica de MCD. El espacio protegido (por ejemplo, menos que toda la memoria (direccionable)) puede establecerse en el procesador de hardware, por ejemplo, escribiendo en un registro de control (o configuración). En una realización, el hardware y los métodos de MCD, por ejemplo, mediante una interfaz ISA, permiten la definición de uno o más de lo siguiente, por ejemplo, por software (por ejemplo, el SO): (1) la posición del valor de MCD (por ejemplo, metadatos) en el puntero, por ejemplo, qué bits de la dirección lineal en el puntero se usan para almacenar el valor de MCD, (2) el espacio protegido de MCD (por ejemplo, rango) para definir el subconjunto de memoria (por ejemplo, direcciones) que se someterá a una detección de corrupción de memoria (por ejemplo, y las líneas de dirección en memoria que tendrán un valor de MCD), por ejemplo, el espacio protegido de MCD puede ser el prefijo de bits de dirección lineal que define la región protegida o el rango de memoria que se someterá a una detección de corrupción de memoria (por ejemplo, y contiene el valor de MCD), y (3) un puntero (por ejemplo, puntero de dirección lineal) a la base de la tabla o tablas de MCD de memoria (por ejemplo, metadatos). En una realización, pueden protegerse múltiples subconjuntos (por ejemplo, regiones) de memoria por MCD, por ejemplo, teniendo múltiples conjuntos de atributos que incluyen la información anterior. En una realización, estos atributos pueden implementarse (por ejemplo, establecerse) a través de un registro (por ejemplo, un registro de control y/o de configuración).

En una realización, puede usarse el siguiente pseudocódigo en la siguiente Tabla 1 para comprobar si una dirección lineal en un puntero es parte de un espacio protegido de MCD (por ejemplo, de manera que ha de realizarse una comprobación de validación de MCD).

Tabla 1

```
LA_Prefix = LA[63:(MCD.Position+6)]
Si (MCD.Enabled && MCD.Prefix == LA_Prefix)
   MCD comprobar LA contra MCD.MemoryMetadatosTable
```

En una realización, hay múltiples regiones (por ejemplo, [i] con un índice diferente i para cada región) y cada región que va a protegerse por MCD puede estar definida por uno o más de: MCD[i].Enabled, MCD[i].Position, MCD[i].ProtectedSpace (por ejemplo, MCD[i].Prefix), y MCD[i]. BaseAddressOfMCDTable. En una realización, un orden (por ejemplo, arbitrario) para el espacio protegido de MCD puede ser como en el siguiente pseudocódigo en la Tabla 2 para N regiones protegidas.

Tabla 2

```
Para i=1 a N
LA_Prefix = LA[63:(MCD[i].Position+6)]
Si (MCD[i].Enabled && MCD[i].Prefix == LA_Prefix)
        MCD comprobar contra MCD[i].MemoryMetadataTable
        Break
```

Como se ha indicado anteriormente, que el valor de MCD sea de 6 bits de ancho es simplemente un ejemplo y pueden utilizarse otros tamaños.

La **Figura 6** ilustra formatos de datos de registros 608 para la detección de corrupción de memoria (MCD) de acuerdo con las realizaciones de la divulgación. Aunque se representan dos registros, puede utilizarse uno o más registros. En una realización, un registro de control o de configuración puede ser un registro específico de modelo (MSR). El registro de configuración de MCD (CFG MSR) 620 puede incluir uno o más de lo siguiente: un campo 622 de espacio protegido de detección de corrupción de memoria (MCD) (por ejemplo, para establecer qué subconjunto de memoria ha de ser protegido por el hardware de MCD y/o los métodos divulgados en el presente documento), un campo 626 de tamaño (por ejemplo, para establecer el tamaño (por ejemplo, el número de posiciones de bits) que incluirá un valor de MCD en el puntero y/o en una tabla de MCD), y el campo 628 de posición (por ejemplo, para establecer qué bits en el puntero han de usarse como el valor de MCD, por ejemplo, la primera posición de bit o la última posición de bit del valor de MCD. En una realización, puede no usarse uno o más campos (por ejemplo, el campo 624 reservado) para MCD. El registro de control de MCD (CTRL MSR) 630 puede incluir uno o más de lo siguiente: dirección de base de un campo de tabla de MCD 632 (por ejemplo, donde una dirección de base más un desplazamiento (por ejemplo, un desplazamiento de la dirección de la línea o líneas desde el puntero) indica un valor de MCD para una línea correspondiente en memoria) y un campo de activación 638 (por ejemplo, se activa la comprobación de MCD cuando está establecido (por ejemplo, a 1)). En una realización, uno o más campos (por ejemplo, el campo reservado 634) no se usan para MCD. En una realización, se usa un campo reservado (por ejemplo, el campo reservado 624 y/o el campo reservado 634) para definir diferentes modos para el comportamiento de la validación de MCD. Aunque se enumeran las posiciones de bits (por ejemplo, tamaños), estas son realizaciones ilustrativas y pueden usarse otras posiciones de bits (por ejemplo, tamaños) en ciertas realizaciones, por ejemplo, y también pueden ser fijas (por ejemplo, constantes y no configurables) en algunas realizaciones. En una realización, puede incluirse uno o más de los campos

anteriores en un único registro o cada campo puede estar en su propio registro.

Una escritura (por ejemplo, instrucción de almacenamiento) en un registro puede establecer uno o más de los campos, por ejemplo, una escritura de software para posibilitar y/o establecer la protección de MCD. Puede utilizarse una pluralidad de conjuntos de configuración de MCD y/o registros de control, por ejemplo, MCD CFG MSR [i] y MCD CTRL MSR [i], por ejemplo, donde i puede ser cualquier número entero positivo. En una realización, existe un valor diferente de i para cada subconjunto (por ejemplo, región) de memoria que se protegerá por MCD, por ejemplo, en donde cada subconjunto (por ejemplo, región) puede tener una tabla de MCD diferente (por ejemplo, y, por lo tanto, dirección de base) y/o tamaño diferente, posición, espacio protegido, combinaciones de lo mismo, etc.

La **Figura 7** ilustra un sistema de detección de corrupción de memoria (MCD) 700 con una unidad de gestión de memoria 706 de acuerdo con las realizaciones de la divulgación. En la realización representada, la unidad de gestión de memoria 706 (por ejemplo, el circuito de gestión de memoria) es para recibir características que se activarán (por ejemplo, de un registro de configuración y/o control), por ejemplo, la posición del valor de MCD en un puntero y/o la ubicación de la tabla de MCD para las líneas en memoria. En la realización representada, la unidad de gestión de memoria 706 es para recibir un puntero (por ejemplo, para una solicitud de acceso a memoria). En una realización, la unidad de gestión de memoria 706 puede realizar una traducción de dirección lineal en el valor de dirección a partir del puntero para determinar la dirección lineal de la línea de memoria a la que se señala mediante el puntero. En una realización, la unidad de gestión de memoria 706 retira un valor de MCD en el puntero de la dirección lineal. En una realización, la unidad de gestión de memoria inserta un valor en las posiciones de bits de valor de MCD retiradas. Por ejemplo, todos los bits retirados del valor de MCD retirado pueden sustituirse por todo ceros o todo unos, por ejemplo, haciendo coincidir el valor del bit más significativo (por ejemplo, la posición de bit 63) del puntero. La dirección lineal sin el valor de MCD puede utilizarse para obtener (por ejemplo, de la tabla 716 de MCD) el valor de MCD asociado para la línea de memoria 710. El valor de MCD en el puntero puede compararse a continuación con el valor de MCD en la tabla para esa línea a la que se está apuntando para determinar si hay una coincidencia (por ejemplo, mediante la unidad de gestión de memoria 706). En ciertas realizaciones, si coinciden los valores de MCD, se cumple la solicitud de datos. En ciertas realizaciones, si no coinciden los valores de MCD, se deniega la solicitud de datos.

La **Figura 8** ilustra una unidad 806 de gestión de memoria de acuerdo con las realizaciones de la divulgación. En el circuito representado en la Figura 8, el comparador de hardware 840 es para comparar el valor de espacio protegido de MCD (por ejemplo, en el ejemplo en el que las posiciones de bits son 63:(X+6) del registro de configuración (por ejemplo, CFG MSR 620 en la Figura 6)) con las mismas posiciones de bits (por ejemplo, 63:(X+6)) del puntero (por ejemplo, el valor de prefijo de dirección lineal en el campo de espacio protegido de MCD en el puntero en la Figura 5). En la realización representada, si la salida del comparador es verdadera (por ejemplo, 1 en binario) y el bit de activación de MCD está activado (por ejemplo, el campo de activación 638 en CTRL MSR 630 en la Figura 6 se establece a 1 en binario), la puerta AND lógica 842 pueden emitir una señal (por ejemplo, 1 en binario). El 1 a partir de la misma puede ser la señal de control al multiplexor 844 y, por lo tanto, provocar una salida del puntero (por ejemplo, la dirección lineal) con el valor de MCD del puntero retirado de la misma. En la realización representada, cada uno de los valores de MCD retirados se sustituye por el valor en la posición de bit más significativo (posición de bit 63) del puntero. Un cero como señal de control al multiplexor 844 puede provocar una salida del puntero original (por ejemplo, para una región de MCD no protegida). Un 1 emitido desde la puerta AND lógica 842 puede hacer que la puerta AND lógica 848 emita los resultados de la puerta O lógica exclusiva (XOR) 846 en el valor de MCD a partir del puntero (por ejemplo, (X+5):X) y el número de bits en el valor de MCD en el puntero multiplicado por el valor bit del bit 63. En una realización, esto es para emitir el valor de MCD. En una realización para punteros canónicos (por ejemplo, punteros donde todos los bits canónicos son idénticos), la puerta XOR 846 es para emitir un valor de MCD de 0. En una realización en referencia a la Figura 16, el campo de valor de MCD se almacena en alguno de los bits canónicos (62:57) y sin MCD, todos estos bits han de ser 0 y con MCD, si estos bits son 0 significa que el valor de MCD es 0. En una realización con referencia a la Figura 16, donde el bit 63 es un 1 sin MCD, todos estos bits han de ser canónicos (por ejemplo, bits 63:56 = 1) y con MCD, si los bits 62:57 son 1, entonces aplicando la operación XOR en ellos con el bit 63 dará también como resultado un valor de MCD de 0. En una realización, esto hace que todos los punteros canónicos tengan un valor de MCD de 0, por ejemplo, que puede ser beneficioso en implementaciones de software. Un cero en la puerta AND lógica 848 es para hacer una salida de cero. Un 1 de la puerta AND lógica 842 puede emitirse como una señal en la que el puntero de entrada está apuntando a una línea de memoria que está en un espacio protegido de MCD. Un 0 de la puerta AND lógica 842 puede emitirse como una señal en la que el puntero de entrada está apuntando a una línea de memoria que no está en un espacio protegido de MCD. Obsérvese que 6 es un ejemplo de tamaño de bits del valor de MCD y que pueden usarse otros tamaños.

Lo siguiente analiza ejemplos del número de líneas que puede identificar de manera inequívoca un puntero de un tamaño determinado, por ejemplo, una dirección lineal de 57 bits puede permitir punteros únicos a 128 petabytes (PB).

La **Figura 9** ilustra un formato de puntero 900 con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) 901 de acuerdo con las realizaciones de la divulgación. Por ejemplo, un sistema operativo (SO) de paginación de 5 niveles puede soportar direcciones lineales de 57 bits en el campo de dirección 901 (por ejemplo, de 64 bits de espacio en el puntero 900). Los restantes siete bits lineales superiores (por ejemplo, más significativos) pueden ser canónicos (por ejemplo, de manera que todos los bits 63:57 tienen el mismo valor que el bit 56).

La **Figura 10** ilustra un formato de puntero 1000 con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) 1001 de acuerdo con las realizaciones de la divulgación. Por ejemplo, un SO puede proporcionar a una aplicación de software el espacio de direcciones lineal positivo (por ejemplo, los bits 63:56 iguales a 0) y reservar el espacio de direcciones lineal negativo (por ejemplo, los bits 63:56 iguales a 1) para su propio uso.

La **Figura 11** ilustra un formato de puntero 1100 con un campo de dirección 1101, un campo de espacio protegido de detección de corrupción de memoria (MCD) en los bits 63:56, y un campo de valor de detección de corrupción de memoria (MCD) 1103 de acuerdo con las realizaciones de la divulgación. Por ejemplo, en una realización con protección de MCD para el espacio de direcciones de aplicación linear y permaneciendo aún dentro del rango de direcciones canónico, pueden establecerse los siguientes atributos (por ejemplo, en un registro o registros): MCD.Enabled = True, MCD.Position = 50 y MCD.Prefix = 00000000.

La **Figura 12A** ilustra un espacio de direcciones lineal 1200 de acuerdo con las realizaciones de la divulgación. El espacio de direcciones lineal 1200 representado puede ser el espacio de direcciones lineal completo que es direccionable (por ejemplo, por un SO). El espacio de direcciones lineal 1200 representado incluye el espacio de direcciones lineal canónico negativo 1250, el espacio de direcciones lineal canónico positivo 1258, el espacio de direcciones lineal no canónico positivo 1256 y el espacio de direcciones lineal no canónico negativo 1254. En una realización, el espacio de direcciones lineal no canónico 1252 incluye las direcciones donde los bits 63:57 no equivalen cada uno al bit 56.

La **Figura 12B** ilustra una vista de una porción del espacio de direcciones lineal 1200 en la Figura 12A de acuerdo con las realizaciones de la divulgación. Más particularmente, la Figura 12B es una vista ampliada del espacio de direcciones lineal positivo (1256 y 1258).

La **Figura 12C** ilustra una vista de la porción del espacio de direcciones lineal 1200 de la Figura 12B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) 1260 de acuerdo con las realizaciones de la divulgación. En una realización, el espacio protegido de MCD 1260 es de 63 petabytes de espacio de direcciones lineal canónico positivo de los 64 petabytes de espacio de direcciones lineal canónico positivo 1258, por ejemplo, dejando 1 petabyte de espacio de direcciones lineal no canónico positivo 1262 no protegido por MCD.

La **Figura 13** ilustra un formato de puntero 1300 con un campo de dirección y sin un campo de valor de detección de corrupción de memoria (MCD) 1301 de acuerdo con las realizaciones de la divulgación. Por ejemplo, puede usarse el MCD (por ejemplo, por un SO) para proteger un subconjunto de espacio de direcciones lineal dentro de su espacio de direcciones total. En una realización, un SO puede reservar el rango de direcciones negativo para su propio uso, por ejemplo, como se muestra en la Figura 13 con los bits 63:56 iguales a 1.

La **Figura 14** ilustra un formato de puntero 1400 con un campo de dirección 1401, un campo de espacio protegido de detección de corrupción de memoria (MCD) 1405 (por ejemplo, y los bits 63:56), y un campo de valor de detección de corrupción de memoria (MCD) 1403 de acuerdo con las realizaciones de la divulgación. Por ejemplo, en una realización con protección de MCD para un subconjunto del espacio de direcciones lineal de SO, pueden establecerse los siguientes atributos (por ejemplo, en un registro o registros): MCD.Enabled = True, MCD.Position = 41 y MCD.Prefix = 11111111XXXXXXXXX (por ejemplo, donde XXXXXXXX es un valor de 9 bits específico que define qué área del espacio de direcciones lineal negativo está protegida por MCD).

La **Figura 15A** ilustra un espacio de direcciones lineal 1500 de acuerdo con las realizaciones de la divulgación. El espacio de direcciones lineal 1500 representado puede ser el espacio de direcciones lineal completo que es direccionable (por ejemplo, por un SO). El espacio de direcciones lineal 1500 representado incluye el espacio de direcciones lineal canónico negativo 1550, el espacio de direcciones lineal canónico positivo 1558, el espacio de direcciones lineal no canónico positivo 1556 y el espacio de direcciones lineal no canónico negativo 1554. En una realización, el espacio de direcciones lineal no canónico 1552 incluye las direcciones donde los bits 63:57 no equivalen cada uno al bit 56.

La **Figura 15B** ilustra una vista de una porción del espacio de direcciones lineal 1500 en la Figura 15A de acuerdo con las realizaciones de la divulgación. Más particularmente, la Figura 15B es una vista ampliada del espacio de direcciones lineal canónico negativo 1550.

La **Figura 15C** ilustra una vista de la porción del espacio de direcciones lineal 1500 de la Figura 15B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) 1560 de acuerdo con las realizaciones de la divulgación. En una realización, el espacio de MCD protegido 1560 es de 128 terabytes de espacio de direcciones lineal disponible de los 64 petabytes de espacio de direcciones lineal canónico negativo 1550. En una realización, la sección de espacio protegido de MCD 1560A y la sección de espacio protegido de MCD 1560B combinadas contienen el rango de direcciones completo que coincide con el valor MCD.Prefix (por ejemplo, 111111XXXXXXXXX). En una realización, la sección de espacio protegido de MCD 1560B es las direcciones donde un valor de MCD del puntero no es 0 (por ejemplo, el mismo que el espacio protegido de MCD 1260 en la Figura 12C). En una realización, la sección de espacio protegido de MCD 1560A es las direcciones donde el valor del puntero MCD

es 0 (por ejemplo, el mismo que el espacio 1262 en la Figura 12C). En ciertas realizaciones, se transforman todas las direcciones que residen en la sección de espacio protegido de MCD 1560B (por ejemplo, de acuerdo con el circuito de la Figura 8) y la operación de memoria real debe ir a las direcciones que están en la sección de espacio protegido de MCD 1560A.

La **Figura 16** ilustra un formato de puntero 1600 con un campo de dirección 1601, un campo de espacio de detección de corrupción de memoria, y un campo de valor de detección de corrupción de memoria (MCD) 1603 de acuerdo con las realizaciones de la divulgación. Por ejemplo, pueden establecerse los siguientes atributos (por ejemplo, en un registro o registros): MCD.Enabled = True, MCD.Position = 57 y MCD.Prefix = 0.

La **Figura 17A** ilustra un espacio de direcciones lineal 1700 de acuerdo con las realizaciones de la divulgación. El espacio de direcciones lineal 1700 representado puede ser el espacio de direcciones lineal completo que es direccionable (por ejemplo, por un SO). El espacio de direcciones lineal 1700 representado incluye el espacio de direcciones lineal canónico negativo 1750, el espacio de direcciones lineal canónico positivo 1758, el espacio de direcciones lineal no canónico positivo 1756 y el espacio de direcciones lineal no canónico negativo 1754. En una realización, el espacio de direcciones lineal no canónico 1752 incluye las direcciones donde los bits 63:57 no equivalen cada uno al bit 56.

La **Figura 17B** ilustra una vista de una porción del espacio de direcciones lineal 1700 en la Figura 17A de acuerdo con las realizaciones de la divulgación. Más particularmente, la Figura 17B es una vista ampliada del espacio de direcciones lineal positivo (1756 y 1758).

La **Figura 17C** ilustra una vista de la porción del espacio de direcciones lineal 1700 de la Figura 17B con un subconjunto de espacio protegido de detección de corrupción de memoria (MCD) en el espacio de direcciones lineal no canónico positivo 1756 de acuerdo con las realizaciones de la divulgación. En una realización, el espacio protegido de MCD se encuentra en secciones alternas, por ejemplo, en el espacio de direcciones lineal no canónico positivo 1756. En una realización, el valor de MCD en un puntero se encuentra en los bits canónicos (62:57), pero el bit 63 es (por ejemplo, se requiere que sea) canónico e igual al bit 56. En una realización, esto significa que las direcciones donde el bit 63 es igual al bit 56 son el espacio protegido de MCD y las direcciones donde el bit 63 no es igual al bit 56 son no canónicas. En la realización representada, cada sección de espacio protegido de MCD (por ejemplo, recuadro) es el tamaño del espacio de direcciones 1758, pero está comprimido para ilustrarlo en esta figura.

La **Figura 18** ilustra un diagrama de flujo 1800 de acuerdo con las realizaciones de la divulgación. El diagrama de flujo 1800 incluye recibir una solicitud para acceder a un bloque de una memoria a través de un puntero al bloque de la memoria 1802, y permitir el acceso al bloque de la memoria cuando se valida un valor de detección de corrupción de memoria en el puntero con un valor de detección de corrupción de memoria en la memoria para el bloque, en donde puede seleccionarse una posición del valor de detección de corrupción de memoria en el puntero entre una primera ubicación y una segunda ubicación 1804 diferente.

En una realización, un procesador de hardware incluye una unidad de ejecución para ejecutar una instrucción para solicitar acceso a un bloque de una memoria a través de un puntero al bloque de la memoria, y una unidad de gestión de memoria para permitir el acceso al bloque de la memoria cuando se valida un valor de detección de corrupción de memoria en el puntero con un valor de detección de corrupción de memoria en la memoria para el bloque, en donde puede seleccionarse una posición del valor de detección de corrupción de memoria en el puntero entre una primera ubicación y una segunda ubicación diferente. El procesador de hardware puede incluir un registro de control para establecer la posición en la primera ubicación o en la segunda ubicación diferente. El procesador de hardware puede incluir un registro de control para establecer un espacio protegido de detección de corrupción de memoria para un subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria puede permitir el acceso al bloque de la memoria sin una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria no está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria puede realizar una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El procesador de hardware puede incluir un registro para almacenar una dirección de base de una tabla de detección de corrupción de memoria en la memoria que comprende el valor de detección de corrupción de memoria para el bloque. La posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre la primera ubicación, la segunda ubicación diferente y una tercera ubicación diferente. El puntero puede incluir una dirección lineal del bloque de la memoria.

En otra realización, un método incluye recibir una solicitud para acceder a un bloque de una memoria a través de un puntero al bloque de la memoria, y permitir el acceso al bloque de la memoria cuando se valida un valor de detección de corrupción de memoria en el puntero con un valor de detección de corrupción de memoria en la memoria para el bloque, en donde puede seleccionarse una posición del valor de detección de corrupción de memoria en el puntero

entre una primera ubicación y una segunda ubicación diferente. El método puede incluir establecer la posición en la primera ubicación o la segunda ubicación diferente. El método puede incluir establecer un espacio protegido de detección de corrupción de memoria para un subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y el método puede incluir permitir el acceso al bloque de la memoria sin una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria no está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y el método puede incluir realizar una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El método puede incluir almacenar una dirección de base de una tabla de detección de corrupción de memoria en la memoria que comprende el valor de detección de corrupción de memoria para el bloque. La posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre la primera ubicación, la segunda ubicación diferente y una tercera ubicación diferente. El puntero puede incluir una dirección lineal del bloque de la memoria.

En otra realización más, un sistema incluye una memoria, un procesador de hardware que comprende una unidad de ejecución para ejecutar una instrucción para solicitar acceso a un bloque de la memoria a través de un puntero al bloque de la memoria, y una unidad de gestión de memoria para permitir el acceso al bloque de la memoria cuando se valida un valor de detección de corrupción de memoria en el puntero con un valor de detección de corrupción de memoria en la memoria para el bloque, en donde puede seleccionarse una posición del valor de detección de corrupción de memoria en el puntero entre una primera ubicación y una segunda ubicación diferente. El sistema puede incluir un registro de control para establecer la posición en la primera ubicación o en la segunda ubicación diferente. El sistema puede incluir un registro de control para establecer un espacio protegido de detección de corrupción de memoria para un subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria puede permitir el acceso al bloque de la memoria sin una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria no está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El puntero puede incluir un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria puede realizar una comprobación de validación del valor de detección de corrupción de memoria en el puntero con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria. El sistema puede incluir un registro para almacenar una dirección de base de una tabla de detección de corrupción de memoria en la memoria que comprende el valor de detección de corrupción de memoria para el bloque. La posición del valor de detección de corrupción de memoria en el puntero puede seleccionarse entre la primera ubicación, la segunda ubicación diferente y una tercera ubicación diferente. El puntero puede incluir una dirección lineal del bloque de la memoria.

En otra realización, un procesador de hardware incluye medios para ejecutar una instrucción para solicitar acceso a un bloque de una memoria a través de un puntero al bloque de la memoria, y medios para permitir acceso al bloque de la memoria cuando se valida un valor de detección de corrupción de memoria en el puntero con un valor de detección de corrupción de memoria en la memoria para el bloque, en donde puede seleccionarse una posición del valor de detección de corrupción de memoria en el puntero entre una primera ubicación y una segunda ubicación diferente.

En otra realización más, un aparato comprende un dispositivo de almacenamiento de datos que almacena un código, que cuando es ejecutado por un procesador de hardware, hace que el procesador de hardware lleve a cabo cualquier método divulgado en el presente documento. Un aparato puede ser como el que se describe en la descripción detallada. Un método puede ser como el que se describe en la descripción detallada.

Un conjunto de instrucciones puede incluir uno o más formatos de instrucción. Un formato de instrucción dado puede definir diversos campos (por ejemplo, número de bits, ubicación de bits) para especificar, entre otras cosas, la operación que va a realizarse (por ejemplo, código de operación) y el operando u operandos en los que ha de realizarse la operación y/u otro campo o campos de datos (por ejemplo, máscara). Algunos formatos de instrucción se descomponen aún más a través de la definición de plantillas de instrucción (o subformatos). Por ejemplo, pueden definirse las plantillas de instrucción de un formato de instrucción dado para que tengan diferentes subconjuntos de los campos del formato de instrucción (los campos incluidos habitualmente están en el mismo orden, pero al menos algunos tienen diferentes posiciones de bits puesto que hay menos campos incluidos) y/o están definidos para que tengan un campo dado interpretado de manera diferente. Por lo tanto, cada instrucción de una ISA se expresa usando un formato de instrucción dado (y, si se define, en una dada de las plantillas de instrucción de ese formato de instrucción) e incluye campos para especificar la operación y los operandos. Por ejemplo, una instrucción ADD ilustrativa tiene un código de operación específico y un formato de instrucción que incluye un campo de código de operación para especificar esos campos de código de operación y operando para seleccionar operandos (fuente1/destino y fuente2); y una ocurrencia de esta instrucción ADD es un flujo de instrucción que tendrá contenidos

específicos en los campos de operando que seleccionan operandos específicos. Se ha liberado y/o publicado un conjunto de extensiones de SIMD denominadas Extensiones de Vector Avanzadas (AVX) (AVX1 y AVX2) y que usan el esquema de codificación de Extensiones de Vector (VEX) (por ejemplo, véase el documento Intel® 64 and IA-32 Architectures Software Developer's Manual, de septiembre de 2015; y véase el documento Intel® Architecture Instruction Set Extensions Programming Reference, de agosto de 2015).

Arquitecturas de núcleo ilustrativas, procesadores y arquitecturas informáticas

Los núcleos de procesador pueden implementarse de diferentes maneras, para diferentes propósitos y en diferentes procesadores. Por ejemplo, las implementaciones de tales núcleos pueden incluir: 1) un núcleo en orden de propósito general previsto para una informática de propósito general; 2) un núcleo fuera de orden de propósito general de alto rendimiento previsto para una informática de propósito general; 3) un núcleo de propósito especial previsto principalmente para gráficos y/o computación científica (rendimiento). Las implementaciones de diferentes procesadores pueden incluir: 1) una CPU que incluye uno o más núcleos en orden de propósito general previstos para informática de propósito general y/o uno o más núcleos fuera de orden de propósito más general previstos para una informática de propósito general; y 2) un coprocesador que incluye uno o más núcleos de propósito especial previstos principalmente para gráficos y/o aspectos científicos (rendimiento). Tales procesadores diferentes conllevan diferentes arquitecturas de sistema informático, que pueden incluir: 1) el coprocesador en un chip separado de la CPU; 2) el coprocesador en una matriz separada en el mismo paquete que una CPU; 3) el coprocesador en la misma matriz que una CPU (en cuyo caso, un coprocesador de este tipo en ocasiones se denomina lógica de propósito especial, tal como lógica de gráficos y/o de aspectos científicos (rendimiento) integrada o como núcleos de propósito especial); y 4) un sistema en un chip que puede incluirse en la misma matriz que la CPU descrita (en ocasiones denominado núcleo o núcleos de aplicación o procesador o procesadores de aplicación), el coprocesador anteriormente descrito y funcionalidad adicional. A continuación, se describen arquitecturas de núcleo ilustrativas, seguidas de descripciones de procesadores y arquitecturas informáticas ilustrativas.

Arquitecturas de núcleo ilustrativas

Diagrama de bloques de núcleo en orden y fuera de orden

La **Figura 19A** es un diagrama de bloques que ilustra tanto una canalización en orden ilustrativa como un renombrado de registro ilustrativo, una canalización de emisión/ejecución fuera de orden de acuerdo con las realizaciones de la divulgación. La Figura 19B es un diagrama de bloques que ilustra tanto una realización ilustrativa de un núcleo de arquitectura en orden como un renombrado de registro ilustrativo, un núcleo de arquitectura de emisión/ejecución fuera de orden que ha de incluirse en un procesador de acuerdo con las realizaciones de la divulgación. Los recuadros de línea continua en las **Figuras 19A-B** ilustran la canalización en orden y el núcleo en orden, mientras que la adición opcional de los recuadros de línea discontinua ilustra el renombrado de registro, la canalización de emisión/ejecución fuera de orden y el núcleo. Dado que el aspecto en orden es un subconjunto del aspecto fuera de orden, se describirá el aspecto fuera de orden.

En la **Figura 19A,** una canalización 1900 de procesador incluye una etapa 1902 de búsqueda, una etapa 1904 de decodificación de longitud, una etapa 1906 de decodificación, una etapa 1908 de asignación, una etapa 1910 de renombrado, una etapa 1912 de planificación (también conocida como despacho o emisión), una etapa 1914 de lectura de registro/lectura de memoria, una etapa 1916 de ejecución, una etapa 1918 de escritura diferida/escritura en memoria, una etapa 1922 de manejo de excepciones y una etapa 1924 de confirmación.

La **Figura 19B** muestra el núcleo de procesador 1990 que incluye una unidad de extremo frontal 1930 acoplada a una unidad de motor de ejecución 1950, y ambas están acopladas a una unidad de memoria 1970. El núcleo 1990 puede ser un núcleo informático de conjunto de instrucciones reducido (RISC), un núcleo de conjunto de instrucciones complejo (CISC), un núcleo de palabra de instrucción muy larga (VLIW) o un núcleo de tipo híbrido o alternativo. Como otra opción más, el núcleo 1990 puede ser un núcleo de propósito especial, tal como, por ejemplo, un núcleo de red o de comunicación, motor de compresión, núcleo de coprocesador, núcleo de unidad de procesamiento de gráficos informáticos de propósito general (GPGPU), núcleo de gráficos o similares.

La unidad de extremo delantero 1930 incluye una unidad de predicción de ramal 1932 acoplada a una unidad de caché de instrucción 1934, que está acoplada a una memoria intermedia de traducción adelantada de instrucción (TLB) 1936, que está acoplada a una unidad de búsqueda de instrucción 1938, que está acoplada a una unidad de decodificación 1940. La unidad de decodificación 1940 (o decodificador o unidad decodificadora) puede decodificar instrucciones (por ejemplo, macroinstrucciones), y generar como salida una o más microoperaciones, puntos de entrada de microcódigo, microinstrucciones, otras instrucciones u otras señales de control, que se decodifican o que reflejan de otra manera o se derivan de las instrucciones originales. La unidad de decodificación 1940 puede implementarse usando diversos mecanismos diferentes. Ejemplos de mecanismos adecuados incluyen, pero sin limitación, tablas de búsqueda, implementaciones de hardware, matrices lógicas programables (PLA), memorias de sólo lectura (ROM) de microcódigo, etc. En una realización, el núcleo 1990 incluye una ROM de microcódigo u otro medio que almacena un microcódigo para ciertas macroinstrucciones (por ejemplo, en la unidad de decodificación 1940 o que de otra manera está dentro de la unidad de extremo delantero 1930). La unidad de decodificación 1940 está acoplada a una unidad

renombradora/asignadora 1952 en la unidad de motor de ejecución 1950.

La unidad de motor de ejecución 1950 incluye la unidad renombradora/asignadora 1952 acoplada a una unidad de retiro 1954 y a un conjunto de una o más unidades del planificador 1956. La unidad o unidades del planificador 1956 representan cualquier número de diferentes planificadores, que incluye estaciones de reserva, ventana de instrucción central, etc. La unidad o unidades del planificador 1956 están acopladas a la unidad o unidades de fichero o ficheros de registro físico 1958. Cada una de las unidades de fichero o ficheros de registro físico 1958 representa uno o más ficheros de registro físico, diferentes de los que almacenan uno o más tipos de datos diferentes, tales como un número entero escalar, coma flotante escalar, número entero empaquetado, coma flotante empaquetado, número entero vectorial, coma flotante vectorial, estado (por ejemplo, un puntero de instrucción que es la dirección de la siguiente instrucción que se va a ejecutar), etc. En una realización, la unidad de fichero o ficheros de registro físico 1958 comprende una unidad de registros de vector, una unidad de registros de máscara de escritura y una unidad de registros escalar. Estas unidades de registro pueden proporcionar registros de vector de arquitectura, registros de máscara de vector y registros de propósito general. La unidad o unidades de fichero o ficheros de registro físico 1958 están solapadas por la unidad de retiro 1954 para ilustrar diversas maneras en las que puede implementarse el renombrado de registro y la ejecución fuera de orden (por ejemplo, usando una memoria o memorias de reordenación y un fichero o ficheros de registro de retiro; usando un fichero o ficheros futuros, una memoria o memorias intermedias de historial y un fichero o ficheros de registro de retiro; usando unos mapas de registro y una agrupación de registros; etc.). La unidad de retiro 1954 y la unidad o unidades de fichero o ficheros de registro físico 1958 están acopladas a la agrupación o agrupaciones de ejecución 1960. La agrupación o agrupaciones de ejecución 1960 incluyen un conjunto de una o más unidades de ejecución 1962 y un conjunto de una o más unidades de acceso a memoria 1964. Las unidades de ejecución 1962 pueden realizar diversas operaciones (por ejemplo, desplazamientos, adición, resta, multiplicación) y en diversos tipos de datos (por ejemplo, coma flotante escalar, número entero empaquetado, coma flotante empaquetada, número entero vectorial, coma flotante vectorial. Aunque algunas realizaciones pueden incluir un número de unidades de ejecución dedicadas a funciones o conjuntos de funciones específicas, otras realizaciones pueden incluir únicamente una unidad de ejecución o múltiples unidades de ejecución que realizan todas las funciones. La unidad o unidades del planificador 1956, la unidad o unidades de fichero o ficheros de registro físico 1958 y la agrupación o agrupaciones de ejecución 1960 se muestran siendo posiblemente varias debido a que ciertas realizaciones crean canalizaciones separadas para ciertos tipos de datos/operaciones (por ejemplo, una canalización de números enteros escalar, una canalización de coma flotante escalar/número entero empaquetado/coma flotante empaquetada/número entero vectorial/coma flotante vectorial y/o una canalización de acceso a memoria, que tiene cada una su propia unidad planificadora, unidad de fichero o ficheros de registro físico y/o agrupación de ejecución, y en el caso de una canalización de acceso a memoria separada, se implementan ciertas realizaciones en las que únicamente la agrupación de ejecución de esta canalización tiene la unidad o unidades de acceso a memoria 1964). Debería entenderse también que, cuando se usan canalizaciones separadas, una o más de estas canalizaciones pueden ser de emisión/ejecución fuera de orden y el resto en orden.

El conjunto de unidades de acceso a memoria 1964 está acoplado a la unidad de memoria 1970, que incluye una unidad de TLB de datos 1972 acoplada a una unidad de caché de datos 1974 acoplada a una unidad de caché de nivel 2 (L2) 1976. En una realización ilustrativa, las unidades de acceso a memoria 1964 pueden incluir una unidad de carga, una unidad de dirección de almacén y una unidad de datos de almacén, cada una de las cuales está acoplada a la unidad de TLB de datos 1972 en la unidad de memoria 1970. La unidad de caché de instrucción 1934 está acoplada adicionalmente a una unidad de caché de nivel 2 (L2) 1976 en la unidad de memoria 1970. La unidad de caché L2 1976 está acoplada a uno o más de otros niveles de caché y, eventualmente, a una memoria principal.

A modo de ejemplo, el renombrado de registro ilustrativo, la arquitectura de núcleo de emisión/ejecución fuera de orden, puede implementar la canalización 1900 como sigue: 1) la búsqueda de instrucción 1938 realiza las etapas de decodificación de búsqueda y longitud 1902 y 1904; 2) la unidad de decodificación 1940 realiza la etapa de decodificación 1906; 3) la unidad renombradora/asignadora 1952 realiza la etapa de asignación 1908 y la etapa de renombrado 1910; 4) la unidad o unidades del planificador 1956 realizan la etapa de planificación 1912; 5) la unidad o unidades de fichero o ficheros de registro físico 1958 y la unidad de memoria 1970 realizan la etapa de lectura de registro/lectura de memoria 1914; la agrupación de ejecución 1960 realiza la etapa de ejecución 1916; 6) la unidad de memoria 1970 y la unidad o unidades de fichero o ficheros de registro físico 1958 realizan la etapa de escritura diferida/escritura de memoria 1918; 7) diversas unidades pueden verse implicadas en la etapa de manejo de excepciones 1922; y 8) la unidad de retiro 1954 y la unidad o unidades de fichero o ficheros de registro físico 1958 realizan la etapa de confirmación 1924.

El núcleo 1990 puede soportar uno o más conjuntos de instrucciones (por ejemplo, el conjunto de instrucciones x86 (con algunas extensiones que se han añadido con versiones más modernas); el conjunto de instrucciones MIPS de MIPS Technologies of Sunnyvale, CA; el conjunto de instrucciones ARM (con extensiones adicionales opcionales tales como NEON) de ARM Holdings of Sunnyvale, CA), que incluye la instrucción o instrucciones descritas en el presente documento. En una realización, el núcleo 1990 incluye una lógica para soportar una extensión de conjunto de instrucciones de datos empaquetados (por ejemplo, AVX1, AVX2), permitiendo que se realicen de esta manera las operaciones usadas por muchas aplicaciones multimedia que se realizarán usando datos empaquetados.

Debería entenderse que, el núcleo puede soportar múltiples hilos (ejecutando dos o más conjuntos paralelos de

operaciones o hilos), y puede hacer eso en una diversidad de maneras que incluye múltiples hilos de segmentación en el tiempo, múltiples hilos simultáneos (donde un único núcleo físico proporciona un núcleo lógico para cada uno de los hilos a los que ese núcleo físico está aplicando múltiples hilos simultáneamente), una combinación de los mismos (por ejemplo, búsqueda de segmentación en el tiempo y decodificación y múltiples hilos simultáneos a continuación, tal como en la tecnología Intel® Hyperthreading).

Si bien se describe el renombrado de registro en el contexto de ejecución fuera de orden, debería entenderse que, el renombrado de registro puede usarse en una arquitectura en orden. Aunque la realización ilustrada del procesador también incluye las unidades de instrucciones y caché de datos 1934/1974 independientes y una unidad de caché L2 1976 compartida, en realizaciones alternativas puede haber una única caché interna tanto para instrucciones como para datos, tal como, por ejemplo, una caché interna de nivel 1 (L1), o múltiples niveles de caché interna. En algunas realizaciones, el sistema puede incluir una combinación de una caché interna y una caché externa que es externa al núcleo y/o al procesador. Como alternativa, todas las cachés pueden ser externas al núcleo y/o al procesador.

Arquitectura de núcleo en orden ilustrativa específica

Las **Figuras 20A-B** ilustran un diagrama de bloques de una arquitectura de núcleo en orden ilustrativa más específica, cuyo núcleo sería uno de varios bloques lógicos (que incluyen otros núcleos del mismo tipo y/o de diferentes tipos) en un chip. Los bloques lógicos se comunican a través de una red de interconexión de alto ancho de banda (por ejemplo, una red en anillo) con alguna lógica de función fija, interfaces de E/S de memoria y otra lógica de E/S necesaria, dependiendo de la aplicación.

La **Figura 20A** es un diagrama de bloques de un único núcleo de procesador, junto con su conexión a la red de interconexión en matriz 2002 y con su subconjunto local de la caché de nivel 2 (L2) 2004, de acuerdo con las realizaciones de la divulgación. En una realización, una unidad de decodificación de instrucción 2000 soporta el conjunto de instrucciones x86 con una extensión de conjunto de instrucciones de datos empaquetados. Una caché L1 2006 permite acceso de baja latencia a la memoria caché en unidades escalares y vectoriales. Aunque en una realización (para simplificar el diseño), una unidad escalar 2008 y una unidad vectorial 2010 usan conjuntos de registros separados (respectivamente, los registros escalares 2012 y los registros vectoriales 2014) y los datos transferidos entre ellas se escriben en memoria y, a continuación, se leen desde una caché de nivel 1 (L1) 2006, en realizaciones alternativas de la divulgación se puede usar un enfoque diferente (por ejemplo, usar un único conjunto de registros o incluir una ruta de comunicación que permite que se transfieran datos entre los dos ficheros de registro sin que se escriban y se lean).

El subconjunto local de la caché L2 2004 es parte de una caché global L2 que se divide en subconjuntos locales separados, uno por núcleo de procesador. Cada núcleo de procesador tiene una ruta de acceso directo a su propio subconjunto local de la caché L2 2004. Los datos leídos por un núcleo de procesador se almacenan en su subconjunto de caché L2 2004 y son rápidamente accesibles, en paralelo con otros núcleos de procesador que acceden a sus propios subconjuntos de caché L2 local. Los datos escritos por un núcleo de procesador se almacenan en su propio subconjunto de caché L2 2004 y se purgan de otros subconjuntos, si fuera necesario. La red en anillo asegura coherencia para datos compartidos. La red en anillo es bidireccional para permitir a los agentes, tales como los núcleos de procesador, las cachés L2 y otros bloques lógicos comunicarse entre sí dentro del chip. Cada ruta de datos de anillo es de 1012 bits de ancho por dirección.

La **Figura 20B** es una vista ampliada de parte del núcleo de procesador de la **Figura 20A** de acuerdo con unas realizaciones de la divulgación. La **Figura 20B** incluye una parte de la caché de datos L1 2006A de la caché L1 2004, así como más detalle con respecto a la unidad vectorial 2010 y a los registros vectoriales 2014. Específicamente, la unidad vectorial 2010 es una unidad de procesamiento vectorial de 16 de ancho (VPU) (véase la ALU 2028 de 16 de ancho), que ejecuta una o más de las instrucciones de números enteros, flotante de precisión única y flotante de precisión doble. La VPU soporta el mezclado de las entradas de registro con la unidad de mezclado 2020, conversión numérica con las unidades de conversión numérica y replicación 2022A-B con la unidad de replicación 2024 en la entrada de memoria. Los registros de máscara de escritura 2026 permiten predicar escrituras vectoriales resultantes.

La **Figura 21** es un diagrama de bloques de un procesador 2100 que puede tener más de un núcleo, puede tener un controlador de memoria integrado, y puede tener gráficos integrados de acuerdo con las realizaciones de la divulgación. Los recuadros de línea continua en la **Figura 21** ilustran un procesador 2100 con un único núcleo 2102A, un agente de sistema 2110, un conjunto de una o más unidades de controlador de bus 2116, mientras que la adición opcional de los recuadros de línea discontinua ilustra un procesador 2100 alternativo con múltiples núcleos 2102A-N, un conjunto de una o más unidad o unidades de controlador de memoria 2114 integrado en la unidad de agente de sistema 2110 y lógica de propósito especial 2108.

Por lo tanto, diferentes implementaciones del procesador 2100 pueden incluir: 1) una CPU con la lógica de propósito especial 2108 que es una lógica (que puede incluir uno o más núcleos) de gráficos y/o de aspectos científicos integrada (rendimiento), y los núcleos 2102A-N que son uno o más núcleos de propósito general (por ejemplo, núcleos en orden de propósito general, núcleos fuera de orden de propósito general, una combinación de los dos); 2) un coprocesador con los núcleos 2102A-N que son un gran número de núcleos de propósito especial previstos principalmente para

gráficos y/o aspectos científicos (rendimiento); y 3) un coprocesador con los núcleos 2102A-N que son un gran número de núcleos en orden de propósito general. Por lo tanto, el procesador 2100 puede ser un procesador de propósito general, un coprocesador o procesador de propósito especial, tal como, por ejemplo, una red o procesador de comunicación, un motor de compresión, un procesador de gráficos, una GPGPU (unidad de procesamiento de gráficos de propósito general), un coprocesador de muchos núcleos integrados (MIC) de alto rendimiento (que incluye 30 o más núcleos), un procesador embebido o similares. El procesador puede implementarse en uno o más chips. El procesador 2100 puede ser una parte de y/o puede implementarse en uno o más sustratos usando cualquiera de un número de tecnologías de proceso, tales como, por ejemplo, BiCMOS, CMOS o NMOS.

La jerarquía de memoria incluye uno o más niveles de caché dentro de los núcleos, un conjunto de una o más unidades de caché 2106 compartida, y memoria externa (no mostrada) acoplada al conjunto de unidades de controlador de memoria 2114 integrado. El conjunto de unidades de caché compartida 2106 puede incluir una o más cachés de nivel medio, tales como de nivel 2 (L2), nivel 3 (L3), nivel 4 (L4) u otros niveles de caché, una caché de último nivel (LLC) y/o combinaciones de las mismas. Aunque, en una realización, una unidad de interconexión basada en anillo 2112 interconecta la lógica de gráficos integrada 2108, el conjunto de unidades de caché compartida 2106, y la unidad de agente de sistema/unidad 2110 o las unidades de controlador de memoria integrado 2114, en realizaciones alternativas se puede usar cualquier número de técnicas bien conocidas para interconectar tales unidades. En una realización, se mantiene la coherencia entre una o más unidades de caché 2106 y los núcleos 2102-A-N.

En algunas realizaciones, uno o más de los núcleos 2102A-N son aptos para múltiples hilos. El agente de sistema 2110 incluye aquellos componentes que coordinan y operan los núcleos 2102A-N. La unidad de agente de sistema 2110 puede incluir, por ejemplo, una unidad de control de potencia (PCU) y una unidad de visualización. La PCU puede ser o incluir la lógica y los componentes necesarios para regular el estado de potencia de los núcleos 2102A-N y la lógica de gráficos integrada 2108. La unidad de visualización es para controlar una o más pantallas conectadas de manera externa.

Los núcleos 2102A-N pueden ser homogéneos o heterogéneos en términos de conjunto de instrucciones de arquitectura; es decir, dos o más de los núcleos 2102A-N pueden ser aptos para ejecutar el mismo conjunto de instrucciones, mientras que otros pueden ser aptos para ejecutar únicamente un subconjunto de ese conjunto de instrucciones o un conjunto de instrucciones diferente.

Arquitecturas informáticas ilustrativas

Las **Figuras 22-25** son diagramas de bloques de arquitecturas informáticas ilustrativas. También son adecuados otros diseños y configuraciones de sistema conocidos en las técnicas para portátiles, ordenadores de sobremesa, PC de mano, asistentes digitales personales, estaciones de trabajo de ingeniería, servidores, dispositivos de red, concentradores de red, conmutadores, procesadores embebidos, procesadores de señales digitales (DSP), dispositivos de gráficos, dispositivos de videojuegos, decodificadores, microcontroladores, teléfonos celulares, reproductores de medios portátiles, dispositivos de mano y diversos otros dispositivos electrónicos. En general, una enorme diversidad de sistemas o dispositivos electrónicos aptos para incorporar un procesador y/u otra lógica de ejecución como se divulga en el presente documento son, en general, adecuados.

Haciendo ahora referencia a la **Figura 22,** se muestra un diagrama de bloques de un sistema 2200 de acuerdo con una realización de la presente divulgación. El sistema 2200 puede incluir uno o más procesadores 2210, 2215, que están acoplados a un concentrador de controlador 2220. En una realización, el concentrador de controlador 2220 incluye un concentrador de controlador de memoria de gráficos (GMCH) 2290 y un concentrador de entrada/salida (IOH) 2250 (que pueden estar en chips separados); el GMCH 2290 incluye controladores de memoria y de gráficos que están acoplados a la memoria 2240 y un coprocesador 2245; el IOH 2250 acopla los dispositivos de entrada/salida (E/S) 2260 al GMCH 2290. Como alternativa, uno o ambos de los controladores de memoria y de gráficos están integrados dentro del procesador (como se describe en el presente documento), la memoria 2240 y el coprocesador 2245 están acoplados directamente al procesador 2210, y el concentrador de controlador 2220 en un único chip con el IOH 2250. La memoria 2240 puede incluir un módulo 2240A de detección de corrupción de memoria, por ejemplo, para almacenar un código que, cuando se ejecuta, hace que un procesador lleve a cabo cualquier método de esta divulgación. En otra realización, el módulo de detección de corrupción de memoria 2240A reside dentro de un procesador y se comunica con la memoria 2240.

La naturaleza opcional de los procesadores 2215 adicionales se indica en la **Figura 22** con líneas discontinuas. Cada procesador 2210, 2215 puede incluir uno o más de los núcleos de procesamiento descritos en el presente documento y puede ser alguna versión del procesador 2100.

La memoria 2240 puede ser, por ejemplo, una memoria de acceso aleatorio dinámica (DRAM), una memoria de cambio de fase (PCM) o una combinación de las dos. Para al menos una realización, el concentrador de controlador 2220 se comunica con el procesador o procesadores 2210, 2215 mediante un bus multipunto, tal como un bus frontal (FSB), una interfaz de punto a punto tal como una Interconexión de Ruta Rápida (QPI) o una conexión 2295 similar.

En una realización, el coprocesador 2245 es un procesador de propósito especial, tal como, por ejemplo, un

procesador de MIC de alto rendimiento, un procesador de red o de comunicación, un motor de compresión, un procesador de gráficos, GPGPU, un procesador embebido o similares. En una realización, el concentrador de controlador 2220 puede incluir un acelerador de gráficos integrado.

Puede haber una diversidad de diferencias entre los recursos físicos 2210, 2215 en términos de un espectro de métrica de mérito que incluyen características de arquitectura, de microarquitectura, térmicas, de consumo de potencia y similares.

En una realización, el procesador 2210 ejecuta instrucciones que controlan operaciones de procesamiento de datos de un tipo general. Embebidas en las instrucciones puede haber instrucciones de coprocesador. El procesador 2210 reconoce estas instrucciones de coprocesador como unas de un tipo que debe ejecutarse mediante el coprocesador 2245 adjunto. Por consiguiente, el procesador 2210 emite estas instrucciones de coprocesador (o señales de control que representan instrucciones de coprocesador) en un bus de coprocesador u otra interconexión, al coprocesador 2245. El coprocesador o coprocesadores 2245 aceptan y ejecutan las instrucciones de coprocesador recibidas.

Haciendo ahora referencia a la **Figura 23**, se muestra un diagrama de bloques de un primer sistema 2300 ilustrativo más específico de acuerdo con una realización de la presente divulgación. Como se muestra en la **Figura 23**, el sistema multiprocesador 2300 es un sistema de interconexión de punto a punto, e incluye un primer procesador 2370 y un segundo procesador 2380 acoplados mediante una interconexión de punto a punto 2350. Cada uno de los procesadores 2370 y 2380 puede ser alguna versión del procesador 2100. En una realización de la divulgación, los procesadores 2370 y 2380 son respectivamente los procesadores 2210 y 2215, mientras que el coprocesador 2338 es el coprocesador 2245. En otra realización, los procesadores 2370 y 2380 son respectivamente el procesador 2210 y coprocesador 2245.

Los procesadores 2370 y 2380 se muestran incluyendo las unidades del controlador de memoria integrado (IMC) 2372 y 2382, respectivamente. El procesador 2370 también incluye como parte de sus unidades de controlador de bus, las interfaces de punto a punto (P-P) 2376 y 2378; de manera similar, el segundo procesador 2380 incluye las interfaces P-P 2386 y 2388. Los procesadores 2370, 2380 pueden intercambiar información mediante una interfaz de punto a punto (P-P) 2350 usando los circuitos de interfaz P-P 2378, 2388. Como se muestra en la **Figura 23,** los IMC 2372 y 2382 acoplan los procesadores a las respectivas memorias, en concreto, una memoria 2332 y una memoria 2334, que pueden ser porciones de memoria principal localmente fijadas a los respectivos procesadores.

Cada uno de los procesadores 2370, 2380 puede intercambiar información con un conjunto de chips 2390 mediante las interfaces P-P 2352, 2354 individuales usando los circuitos de interfaz de punto a punto 2376, 2394, 2386, 2398. El conjunto de chips 2390, opcionalmente, puede intercambiar información con el coprocesador 2338 mediante una interfaz de alto rendimiento 2339. En una realización, el coprocesador 2338 es un procesador de propósito especial, tal como, por ejemplo, un procesador de MIC de alto rendimiento, un procesador de red o de comunicación, un motor de compresión, un procesador de gráficos, GPGPU, un procesador embebido o similares.

Puede incluirse una caché compartida (no mostrada) en cualquier procesador o fuera de ambos procesadores, pero conectada con los procesadores mediante interconexión P-P, de manera que la información de caché local de cualquiera o ambos procesadores puede almacenarse en la caché compartida si se pone un procesador en un modo de baja potencia.

El conjunto de chips 2390 puede estar acoplado a un primer bus 2316 mediante una interfaz 2396. En una realización, el primer bus 2316 puede ser un bus de Interconexión de Componentes Periféricos (PCI), o un bus tal como un bus PCI Express u otro bus de interconexión de E/S de tercera generación, aunque el alcance de la presente divulgación no está limitado a ello.

Como se muestra en la **Figura 23,** diversos dispositivos de E/S 2314 pueden acoplarse al primer bus 2316, junto con un puente de bus 2318 que acopla el primer bus 2316 a un segundo bus 2320. En una realización, uno o más procesador o procesadores 2315 adicionales, tales como coprocesadores, procesadores de MIC de alto rendimiento, GPGPU, aceleradores (tales como, por ejemplo, aceleradores gráficos o unidades de procesamiento de señal digital (DSP)), campos de matrices de puertas programables o cualquier otro procesador, están acoplados el primer bus 2316. En una realización, el segundo bus 2320 puede ser un bus de bajo recuento de patillas (LPC). Diversos dispositivos pueden acoplarse a un segundo bus 2320 que incluye, por ejemplo, un teclado y/o ratón 2322, dispositivos de comunicación 2327 y una unidad de almacenamiento 2328, tal como una unidad de disco u otro dispositivo de almacenamiento masivo que puede incluir instrucciones/código y datos 2330, en una realización. Además, una E/S 2324 de audio puede estar acoplada al segundo bus 2320. Obsérvese que son posibles otras arquitecturas. Por ejemplo, en lugar de la arquitectura de punto a punto de la **Figura 23**, un sistema puede implementar un bus multipunto u otra arquitectura de este tipo.

Haciendo ahora referencia a la **Figura 24,** se muestra un diagrama de bloques de un segundo sistema 2400 ilustrativo más específico de acuerdo con una realización de la presente divulgación. Elementos similares de las **Figuras 23 y 24** llevan números de referencia similares, y ciertos aspectos de la **Figura 23** se han omitido en la **Figura 24** para evitar oscurecer otros aspectos de la **Figura 24**.

La **Figura 24** ilustra que los procesadores 2370, 2380 pueden incluir una memoria integrada y una lógica de control de E/S ("CL") 2372 y 2382, respectivamente. Por lo tanto, la CL 2372, 2382 incluye las unidades de controlador de memoria integrado e incluye la lógica de control de E/S. La **Figura 24** ilustra que no sólo las memorias 2332, 2334 están acopladas a la CL 2372, 2382, sino que además los dispositivos 2414 de E/S también están acoplados a la lógica 2372, 2382 de control. Los dispositivos de E/S 2415 heredados también están acoplados al conjunto de chips 2390.

Haciendo ahora referencia a la **Figura 25,** se muestra un diagrama de bloques de un SoC 2500 de acuerdo con una realización de la presente divulgación. Elementos similares a los de la **Figura 21** llevan números de referencia similares. También, los recuadros de línea discontinua son características opcionales en SoC más avanzados. En la **Figura 25,** una unidad o unidades de interconexión 2502 están acopladas a: un procesador de aplicación 2510 que incluye un conjunto de uno o más núcleos 202A-N y unidad o unidades de caché compartida 2106; una unidad de agente de sistema 2110; una unidad o unidades de controlador de bus 2116; una unidad o unidades de controlador de memoria integrado 2114; un conjunto o uno o más coprocesadores 2520 que pueden incluir lógica de gráficos integrada, un procesador de imágenes, un procesador de audio y un procesador de vídeo; una unidad de memoria de acceso aleatorio estática (SRAM) 2530; una unidad de acceso a memoria directo (DMA) 2532; y una unidad de visualización 2540 para acoplarse a una o más pantallas externas. En una realización, el coprocesador o coprocesadores 2520 incluyen un procesador de propósito especial, tal como, por ejemplo, una red o procesador de comunicación, un motor de compresión, una GPGPU, un procesador de MIC de alto rendimiento, un procesador embebido o similares.

Las realizaciones (por ejemplo, de los mecanismos) divulgadas en el presente documento pueden implementarse en hardware, software, firmware o una combinación de tales enfoques de implementación. Las realizaciones de la divulgación pueden implementarse como programas informáticos o un código de programa que se ejecuta en sistemas programables que comprenden al menos un procesador, un sistema de almacenamiento (que incluye memoria volátil y no volátil y/o elementos de almacenamiento), al menos un dispositivo de entrada y al menos un dispositivo de salida.

El código de programa, tal como el código 2330 ilustrado en la **Figura 23**, puede aplicarse a instrucciones de entrada para realizar las funciones descritas en el presente documento y generar información de salida. La información de salida puede aplicarse a uno o más dispositivos de salida, de manera conocida. A efectos de esta solicitud, un sistema de procesamiento incluye cualquier sistema que tenga un procesador, tal como, por ejemplo; un procesador de señales digitales (DSP), un microcontrolador, un circuito integrado para una aplicación específica (ASIC) o un microprocesador.

El código de programa puede implementarse en un lenguaje procedural de alto nivel o de programación orientado a objetos para comunicarse con un sistema de procesamiento. El código de programa también puede implementarse en lenguaje ensamblador o máquina, si se desea. De hecho, los mecanismos descritos en el presente documento no están limitados en alcance a ningún lenguaje de programación particular. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

Uno o más aspectos de al menos una realización pueden implementarse por instrucciones representativas almacenadas en un medio legible por máquina que representa diversas lógicas dentro del procesador, que, cuando son leídas por una máquina, hacen que la máquina fabrique la lógica para realizar las técnicas descritas en el presente documento. Tales representaciones, conocidas como "núcleos de IP" pueden almacenarse en un medio legible por máquina tangible y suministrarse a diversos clientes o instalaciones de fabricación para cargarse en las máquinas de fabricación que realmente constituyen la lógica o el procesador.

Tales medios de almacenamiento legibles por máquina pueden incluir, sin limitación, disposiciones tangibles, no transitorias, de artículos fabricados o formados por una máquina o dispositivo, que incluyen medios de almacenamiento tales como discos duros, cualquier otro tipo de disco, incluyendo discos flexibles, discos ópticos, memorias de sólo lectura en disco compacto (CD-ROM), discos compactos rescribibles (CD-RW), y discos magneto-ópticos, dispositivos de semiconductores tales como memorias de sólo lectura (ROM), memorias de acceso aleatorio (RAM) tales como memorias de acceso aleatorio dinámicas (DRAM), memorias de acceso aleatorio estáticas (SRAM), memorias de sólo lectura programables borrables (EPROM), memorias flash, memorias de sólo lectura programables eléctricamente borrables (EEPROM), memoria de cambio de fase (PCM), tarjetas magnéticas u ópticas o cualquier otro tipo de medios adecuados para almacenar instrucciones electrónicas.

Por consiguiente, las realizaciones de la divulgación también incluyen medios legibles por máquina tangibles no transitorios que contienen instrucciones o que contienen datos de diseño, tales como Lenguaje de Descripción de Hardware (HDL), que define estructuras, circuitos, aparatos, procesadores y/o características de sistema descritas en el presente documento. Tales realizaciones pueden denominarse también productos de programa.

Emulación (incluyendo traducción binaria, transformación de código, etc.)

En algunos casos, puede usarse un convertidor de instrucciones para convertir una instrucción de un conjunto de instrucciones de origen a un conjunto de instrucciones objetivo. Por ejemplo, el convertidor de instrucciones puede

traducir (por ejemplo, usando traducción binaria estática, traducción binaria dinámica que incluye compilación dinámica), transformar, emular o convertir de otra manera una instrucción a una o más instrucciones distintas que van a ser procesadas por el núcleo. El convertidor de instrucciones puede implementarse en software, hardware, firmware o en una combinación de los mismos. El convertidor de instrucciones puede estar en el procesador, fuera del procesador, o estar en parte dentro y en parte fuera del procesador.

La **Figura 26** es un diagrama de bloques que contrasta el uso de un convertidor de instrucción de software para convertir instrucciones binarias en un conjunto de instrucciones de origen a instrucciones binarias en un conjunto de instrucciones objetivo de acuerdo con las realizaciones de la divulgación. En la realización ilustrada, el convertidor de instrucciones es un convertidor de instrucciones de software, aunque, como alternativa, el convertidor de instrucciones puede implementarse en software, firmware, hardware o en diversas combinaciones de los mismos. La **Figura 26** muestra un programa en un lenguaje de alto nivel 2602 que puede compilarse usando un compilador x86 2604 para generar un código binario x86 2606 que puede ser ejecutado de manera nativa por un procesador con al menos un núcleo de conjunto de instrucciones x86 2616. El procesador con al menos un núcleo de conjunto de instrucciones x86 2616 representa cualquier procesador que pueda realizar sustancialmente las mismas funciones que un procesador Intel con al menos un núcleo de conjunto de instrucciones x86 ejecutando de manera compatible o procesando de otra manera (1) una porción sustancial del conjunto de instrucciones del núcleo de conjunto de instrucciones Intel x86 o (2) versiones de código objeto de aplicaciones u otro software dirigido para ejecutarse en un procesador Intel con al menos un núcleo de conjunto de instrucciones x86, para conseguir sustancialmente el mismo resultado que un procesador Intel con al menos un núcleo de conjunto de instrucciones x86. El compilador x86 2604 representa un compilador que puede operarse para generar un código binario x86 2606 (por ejemplo, código objeto) que puede ejecutarse, con o sin procesamiento de vinculación adicional, en el procesador con al menos un núcleo de conjunto de instrucciones x86 2616. De manera similar, la **Figura 26** muestra que el programa en el lenguaje de alto nivel 2602 puede compilarse usando un compilador de conjunto de instrucciones alternativo 2608 para generar un código binario de conjunto de instrucciones alternativo 2610 que puede ser ejecutado de manera nativa por un procesador sin al menos un núcleo de conjunto de instrucciones x86 2614 (por ejemplo, un procesador con núcleos que ejecutan el conjunto de instrucciones MIPS de MIPS Technologies of Sunnyvale, CA y/o que ejecuta el conjunto de instrucciones ARM de ARM Holdings of Sunnyvale, CA). El convertidor de instrucciones 2612 se usa para convertir el código binario x86 2606 en código que puede ser ejecutado de manera nativa por el procesador sin un núcleo de conjunto de instrucciones x86 2614. Este código convertido no es probable que sea el mismo que el código binario de conjunto de instrucciones alternativo 2610 porque es difícil hacer un convertidor de instrucciones capaz de esto; sin embargo, el código convertido conseguirá la operación general y estará compuesto de instrucciones del conjunto de instrucciones alternativo. Por lo tanto, el convertidor de instrucciones 2612 representa software, firmware, hardware o una combinación de los mismos que, a través de emulación, simulación o cualquier otro proceso, permite que un procesador u otro dispositivo electrónico que no tiene un procesador de conjunto de instrucciones x86 o núcleo ejecute el código binario x86 2606.

REIVINDICACIONES

1. Un procesador de hardware (100) que comprende:

una unidad de ejecución (104) para ejecutar una instrucción para solicitar acceso a un bloque de una memoria (100) a través de un puntero (105) al bloque de la memoria (100); y
una unidad de gestión de memoria (106) para permitir el acceso al bloque de la memoria (100) cuando se valida un valor de detección de corrupción de memoria en el puntero (105) con un valor de detección de corrupción de memoria en la memoria para el bloque (100),
caracterizado por que
puede seleccionarse por software una posición del valor de detección de corrupción de memoria en el puntero (105) entre una primera ubicación y una segunda ubicación diferente.

2. El procesador de hardware (100) de la reivindicación 1, que comprende adicionalmente un registro de control para establecer la posición en la primera ubicación o en la segunda ubicación diferente.

3. El procesador de hardware (100) de una cualquiera de las reivindicaciones 1-2, que comprende adicionalmente un registro de control para establecer un espacio protegido de detección de corrupción de memoria para un subconjunto de la memoria.

4. El procesador de hardware (100) de la reivindicación 3, en donde el puntero (105) comprende adicionalmente un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria (106) es para permitir el acceso al bloque de la memoria (100) sin una comprobación de validación del valor de detección de corrupción de memoria en el puntero (105) con el valor de detección de corrupción de memoria en la memoria para el bloque cuando el valor de espacio protegido de detección de corrupción de memoria no está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria.

5. El procesador de hardware (100) de la reivindicación 3, en donde el puntero (105) comprende adicionalmente un valor de espacio protegido de detección de corrupción de memoria, y la unidad de gestión de memoria (106) es para realizar una comprobación de validación del valor de detección de corrupción de memoria en el puntero (105) con el valor de detección de corrupción de memoria en la memoria para el bloque (100) cuando el valor de espacio protegido de detección de corrupción de memoria está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria.

6. El procesador de hardware (100) de una cualquiera de las reivindicaciones 1-5, que comprende adicionalmente un registro para almacenar una dirección de base de una tabla de detección de corrupción de memoria en la memoria que comprende el valor de detección de corrupción de memoria para el bloque (100).

7. El procesador de hardware (100) de una cualquiera de las reivindicaciones 1-6, en donde la posición del valor de detección de corrupción de memoria en el puntero (105) puede seleccionarse entre la primera ubicación, la segunda ubicación diferente y una tercera ubicación diferente.

8. El procesador de hardware (100) de una cualquiera de las reivindicaciones 1-7, en donde el puntero (105) comprende una dirección lineal del bloque de la memoria (100).

9. Un método que comprende:

recibir una solicitud para acceder a un bloque de una memoria (100) a través de un puntero (105) al bloque de la memoria (100); y
permitir el acceso al bloque de la memoria (100) cuando se valida un valor de detección de corrupción de memoria en el puntero (105) con un valor de detección de corrupción de memoria en la memoria para el bloque (100),
caracterizado por que
puede seleccionarse por software una posición del valor de detección de corrupción de memoria en el puntero (105) entre una primera ubicación y una segunda ubicación diferente.

10. El método de la reivindicación 9, que comprende adicionalmente establecer la posición en la primera ubicación o en la segunda ubicación diferente.

11. El método de una cualquiera de las reivindicaciones 9-10, que comprende adicionalmente establecer un espacio protegido de detección de corrupción de memoria para un subconjunto de la memoria.

12. El método de la reivindicación 11, en donde el puntero (105) comprende adicionalmente un valor de espacio protegido de detección de corrupción de memoria, y que comprende adicionalmente permitir el acceso al bloque de la memoria (100) sin una comprobación de validación del valor de detección de corrupción de memoria en el puntero (105) con el valor de detección de corrupción de memoria en la memoria para el bloque (100) cuando el valor de espacio protegido de detección de corrupción de memoria no está dentro del espacio protegido de detección de

corrupción de memoria para el subconjunto de la memoria.

13. El método de la reivindicación 11, en donde el puntero (105) comprende adicionalmente un valor de espacio protegido de detección de corrupción de memoria, y que comprende adicionalmente la realización de una comprobación de validación del valor de detección de corrupción de memoria en el puntero (105) con el valor de detección de corrupción de memoria en la memoria para el bloque (100) cuando el valor de espacio protegido de detección de corrupción de memoria está dentro del espacio protegido de detección de corrupción de memoria para el subconjunto de la memoria.

14. El método de una cualquiera de las reivindicaciones 9-13, que comprende adicionalmente almacenar una dirección de base de una tabla de detección de corrupción de memoria en la memoria que comprende el valor de detección de corrupción de memoria para el bloque (100).

15. El método de una cualquiera de las reivindicaciones 9-14, en donde la posición del valor de detección de corrupción de memoria en el puntero (105) puede seleccionarse entre la primera ubicación, la segunda ubicación diferente y una tercera ubicación diferente.

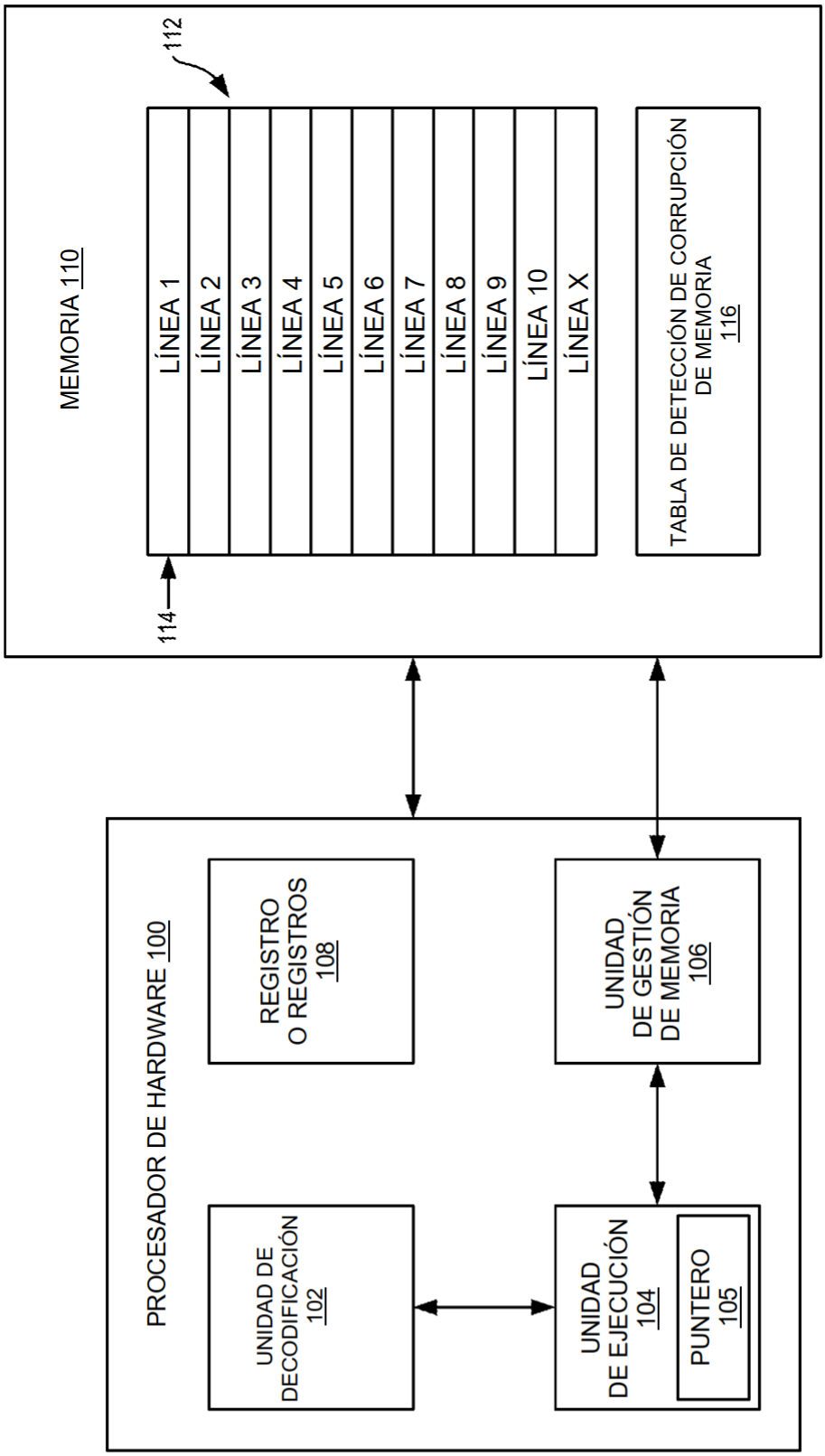
MEMORIA 110
112
LÍNEA 1
LÍNEA 2
LÍNEA 3
LÍNEA 4
LÍNEA 5
LÍNEA 6
LÍNEA 7
LÍNEA 8
LÍNEA 9
LÍNEA 10
LÍNEA X
114
TABLA DE DETECCIÓN DE CORRUPCIÓN DE MEMORIA 116
PROCESADOR DE HARDWARE 100
REGISTRO O REGISTROS 108
UNIDAD DE GESTIÓN DE MEMORIA 106
UNIDAD DE DECODIFICACIÓN 102
UNIDAD DE EJECUCIÓN 104
PUNTERO 105

FIG. 1

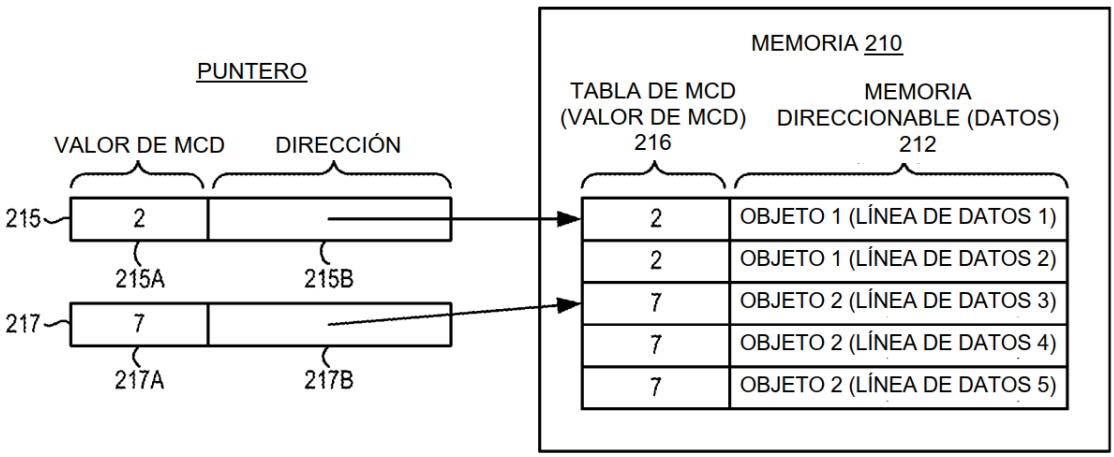

FIG. 2

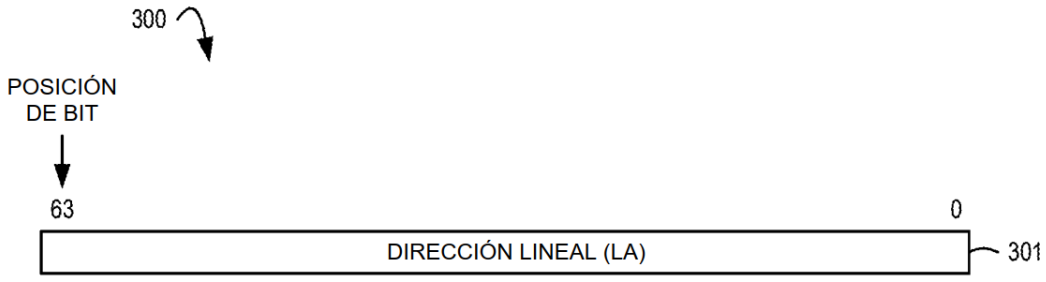

FIG. 3

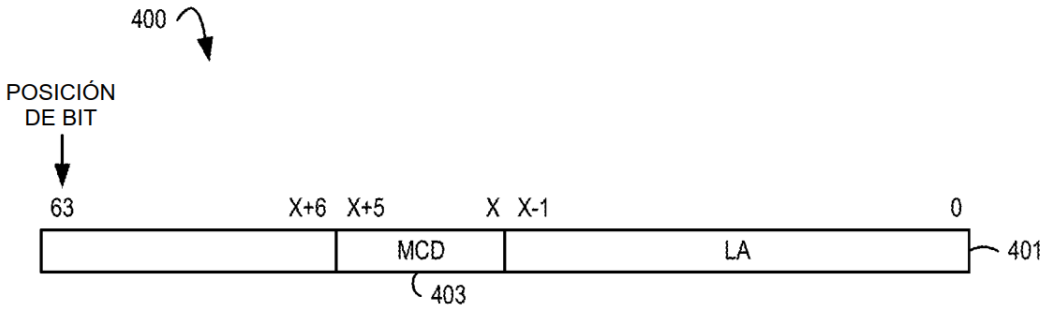

FIG. 4

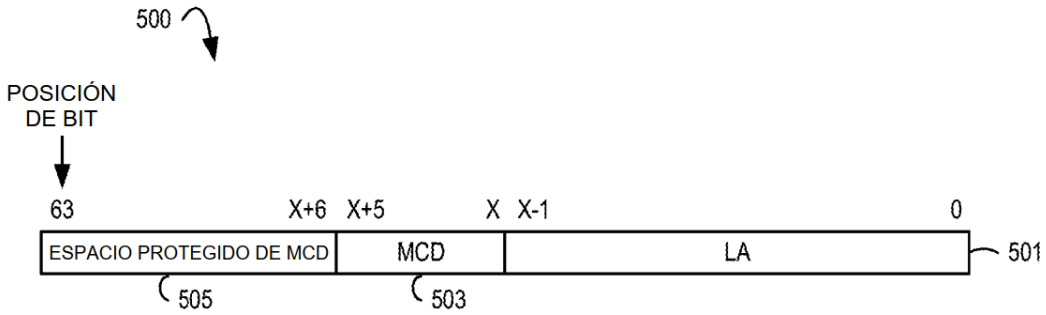

FIG. 5

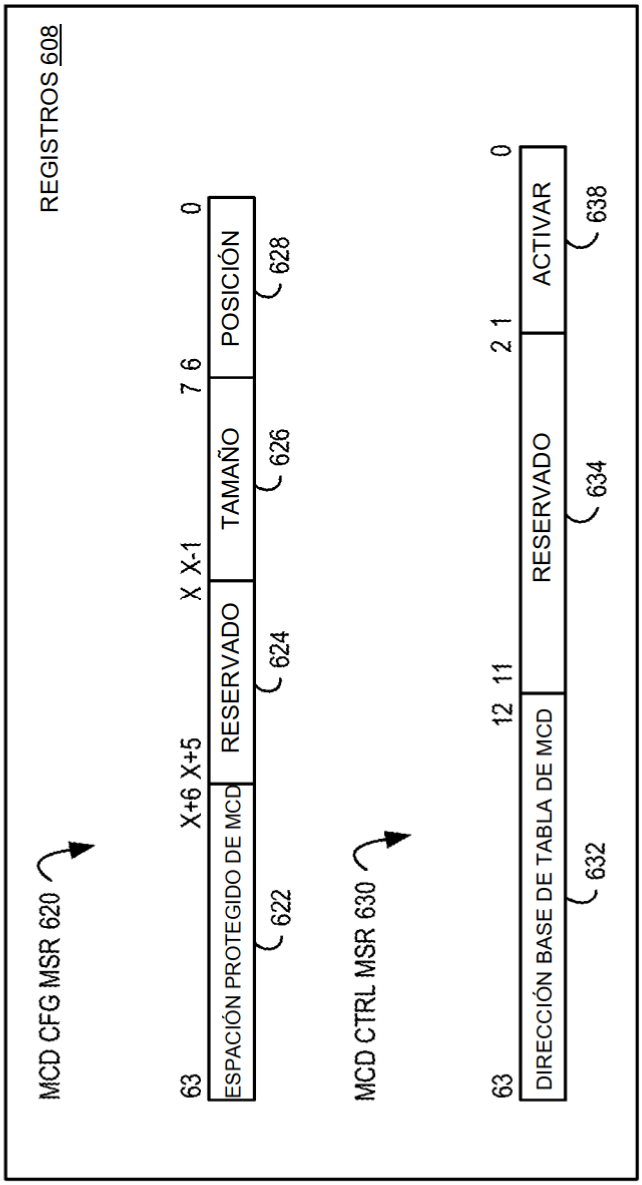
REGISTROS 608
MCD CFG MSR 620
63
X+6 X+5
X X-1
7 6
0
ESPACIÓN PROTEGIDO DE MCD
RESERVADO
TAMAÑO
POSICIÓN
622
624
626
628
MCD CTRL MSR 630
63
12 11
2 1
0
DIRECCIÓN BASE DE TABLA DE MCD
RESERVADO
ACTIVAR
632
634
638

FIG. 6

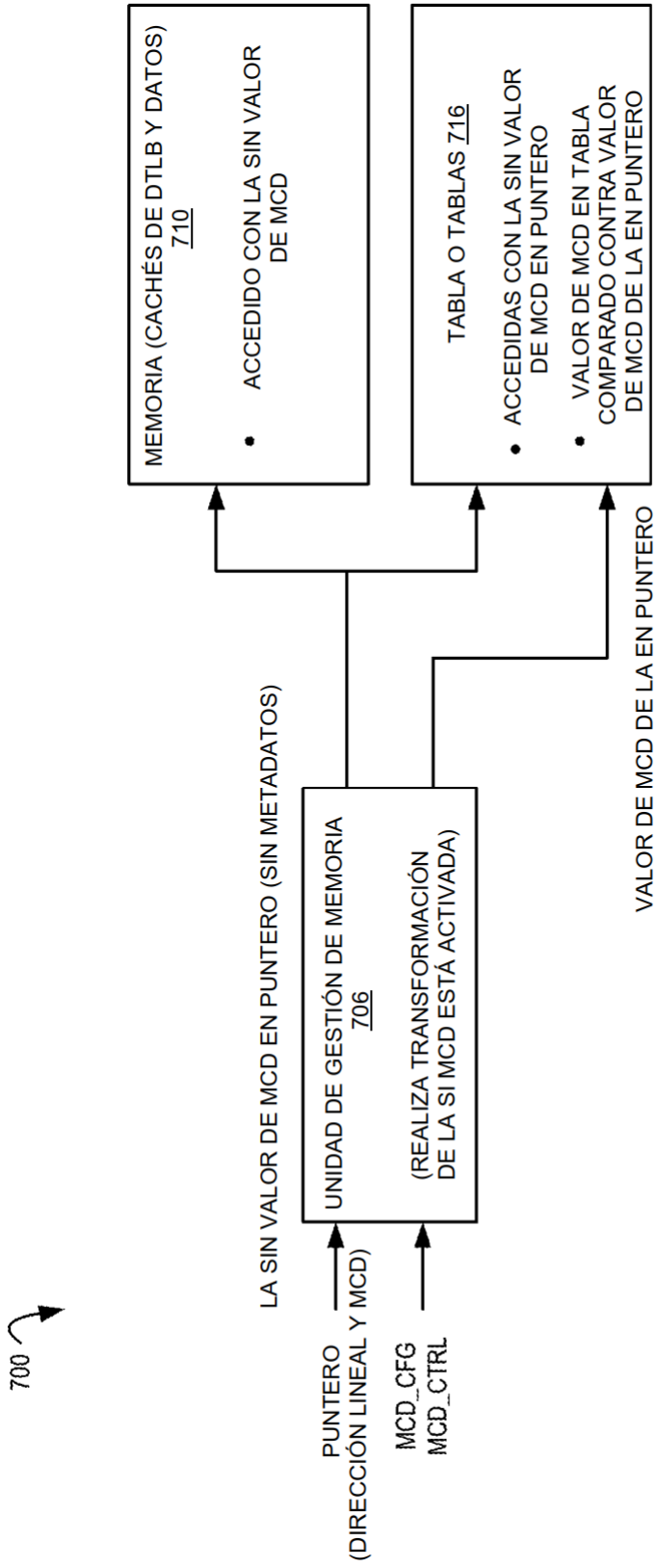
700
PUNTERO
(DIRECCIÓN LINEAL Y MCD)
MCD_CFG
MCD_CTRL
LA SIN VALOR DE MCD EN PUNTERO (SIN METADATOS)
UNIDAD DE GESTIÓN DE MEMORIA
706
(REALIZA TRANSFORMACIÓN
DE LA SI MCD ESTÁ ACTIVADA)
VALOR DE MCD DE LA EN PUNTERO
(METADATO)
MEMORIA (CACHÉS DE DTLB Y DATOS)
710
ACCEDIDO CON LA SIN VALOR
DE MCD
TABLA O TABLAS 716
ACCEDIDAS CON LA SIN VALOR
DE MCD EN PUNTERO
VALOR DE MCD EN TABLA
COMPARADO CONTRA VALOR
DE MCD DE LA EN PUNTERO

FIG. 7

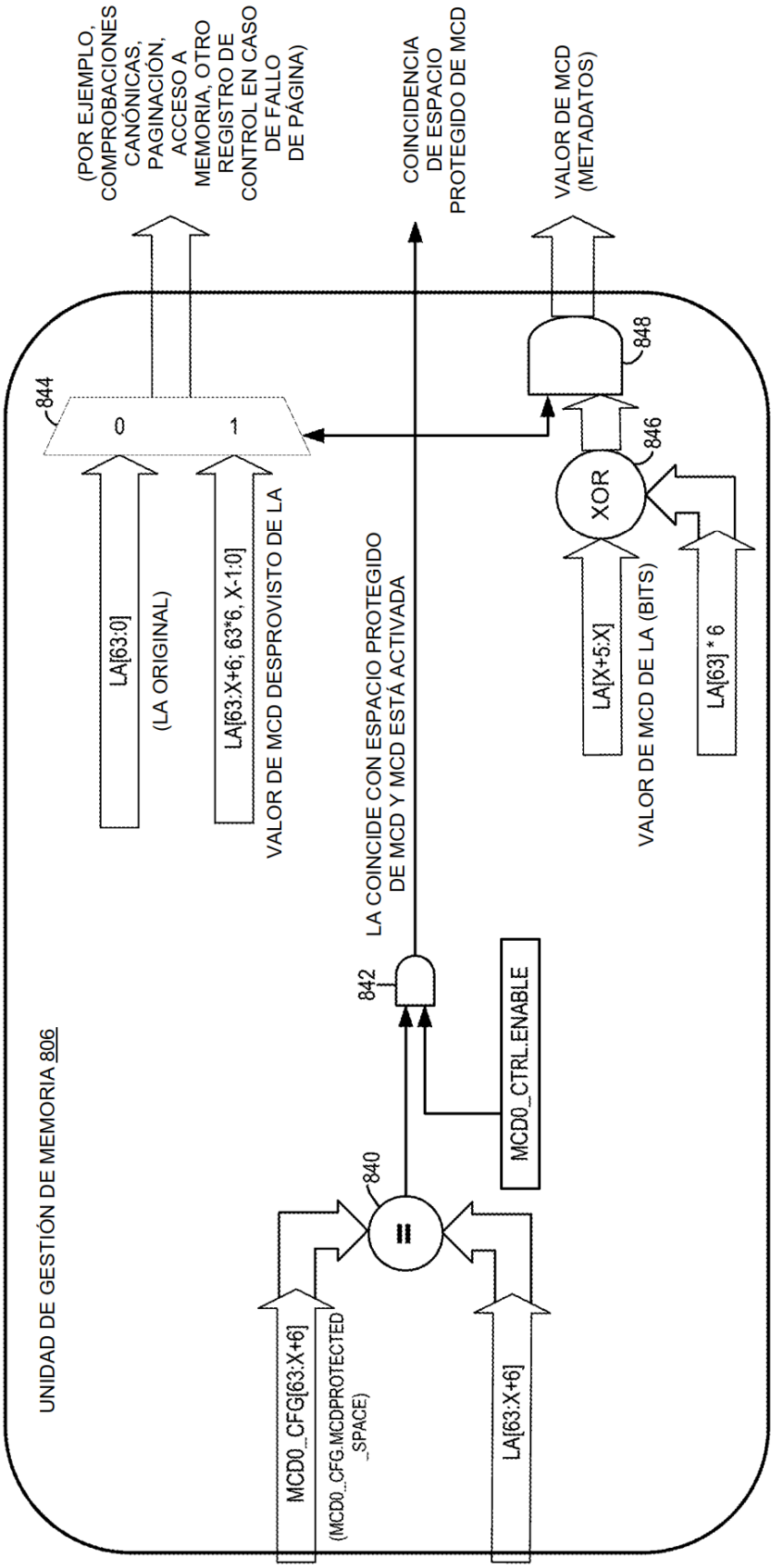
UNIDAD DE GESTIÓN DE MEMORIA 806
MCD0_CFG[63:X+6]
(MCD0_CFG.MCDPROTECTED_SPACE)
LA[63:X+6]
840
=
842
LA COINCIDE CON ESPACIO PROTEGIDO DE MCD Y MCD ESTÁ ACTIVADA
MCD0_CTRL.ENABLE
LA[63:0]
(LA ORIGINAL)
LA[63:X+6; 63*6, X-1:0]
VALOR DE MCD DESPROVISTO DE LA
844
0
1
(POR EJEMPLO, COMPROBACIONES CANÓNICAS, PAGINACIÓN, ACCESO A MEMORIA, OTRO REGISTRO DE CONTROL EN CASO DE FALLO DE PÁGINA)
COINCIDENCIA DE ESPACIO PROTEGIDO DE MCD
VALOR DE MCD (METADATOS)
848
846
XOR
LA[X+5:X]
VALOR DE MCD DE LA (BITS)
LA[63] * 6

FIG. 8

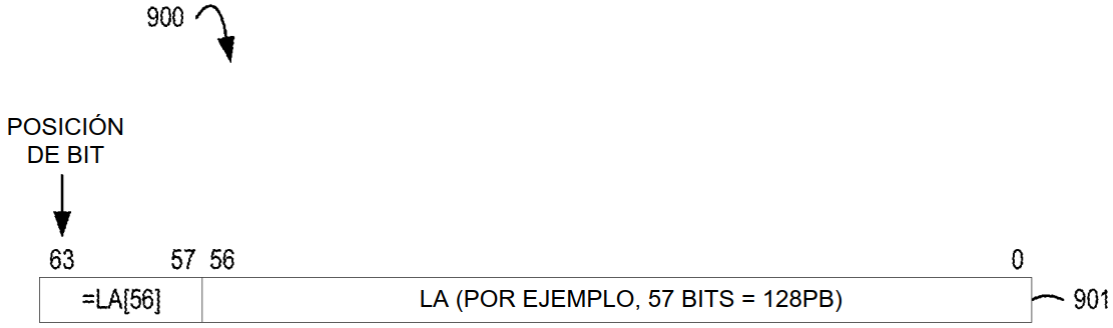

FIG. 9

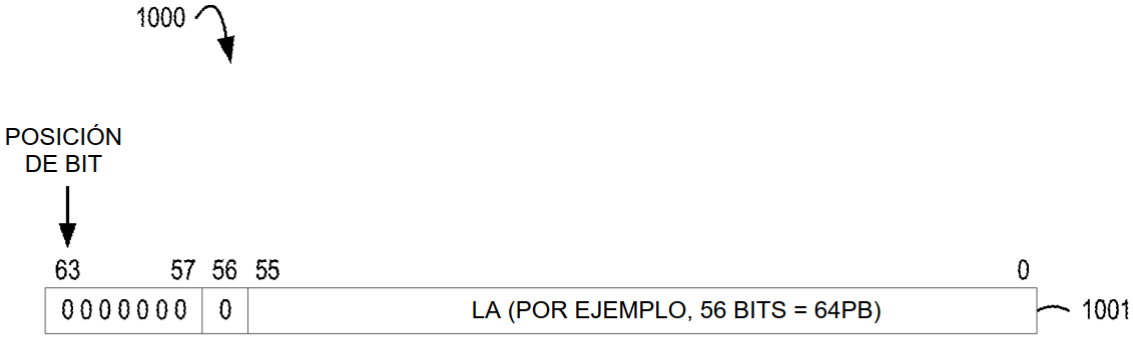

FIG. 10

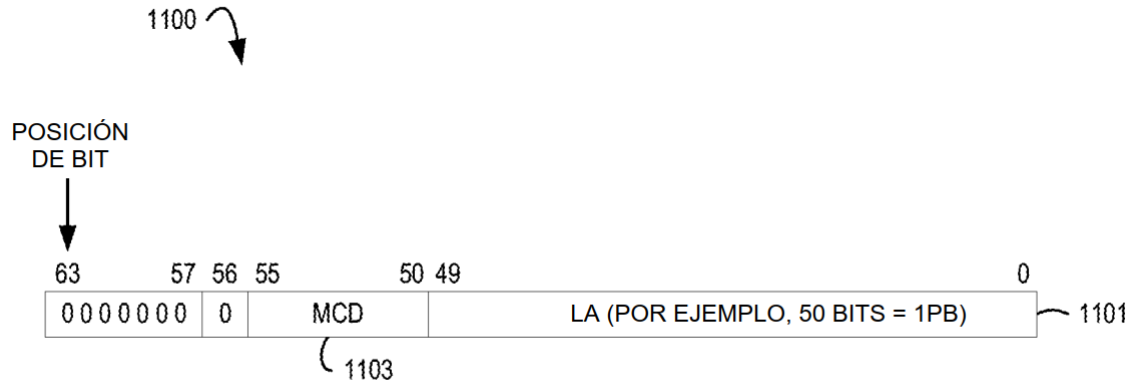

FIG. 11

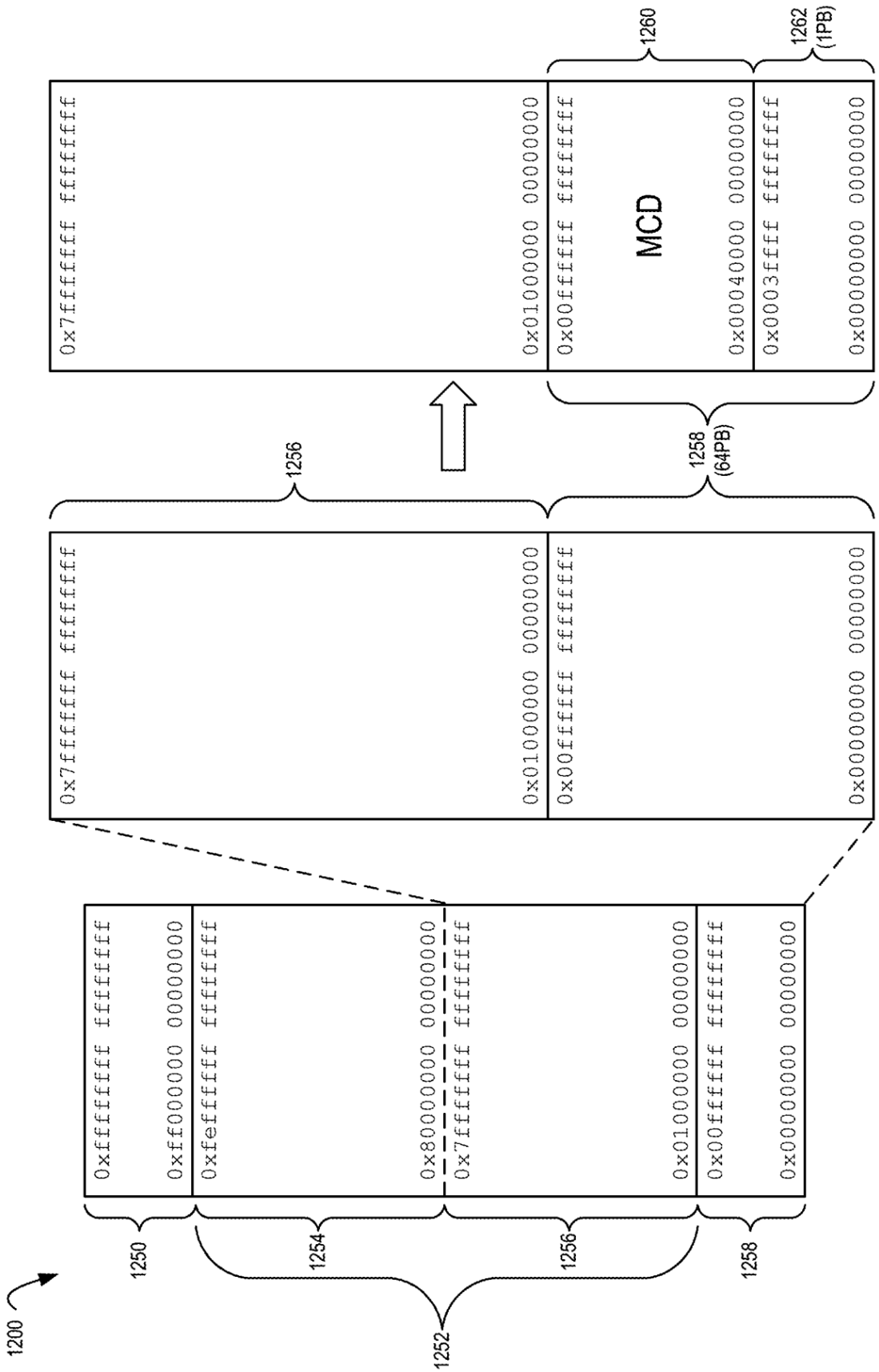

FIG. 12A

FIG. 12B

FIG. 12C

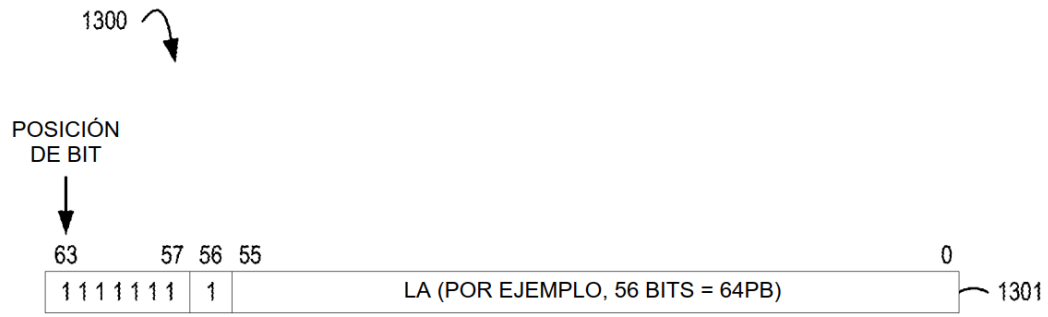
1300
POSICIÓN DE BIT
63
57
56
55
0
1 1 1 1 1 1 1
1
LA (POR EJEMPLO, 56 BITS = 64PB)
1301

FIG. 13

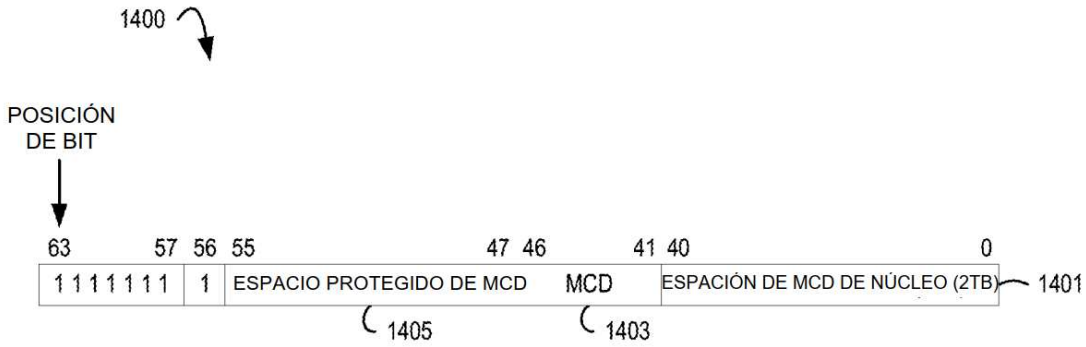
1400
POSICIÓN DE BIT
63
57
56
55
47
46
41
40
0
1 1 1 1 1 1 1
1
ESPACIO PROTEGIDO DE MCD
MCD
ESPACIÓN DE MCD DE NÚCLEO (2TB)
1401
1405
1403

FIG. 14

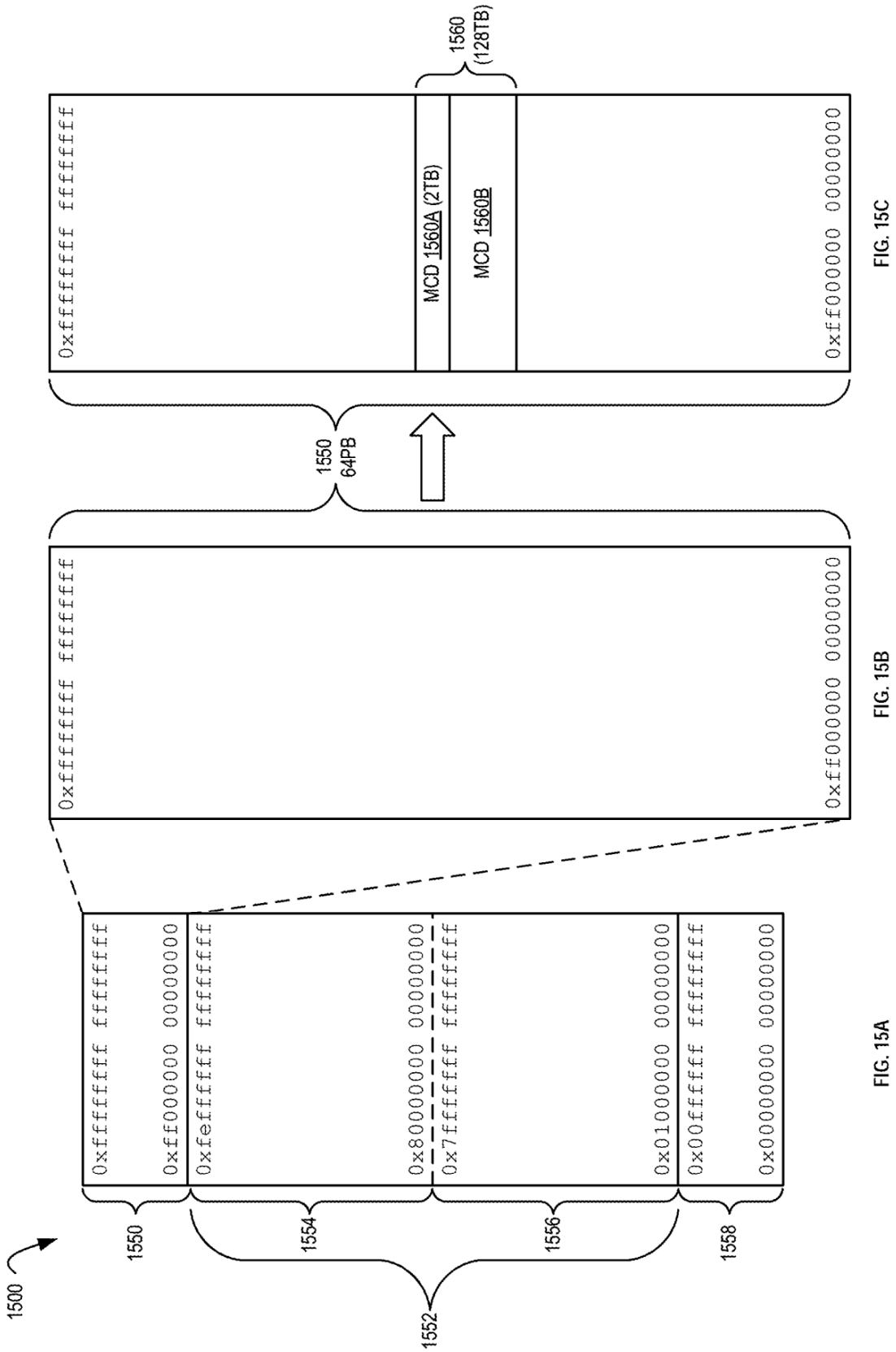

FIG. 15A

FIG. 15B

FIG. 15C

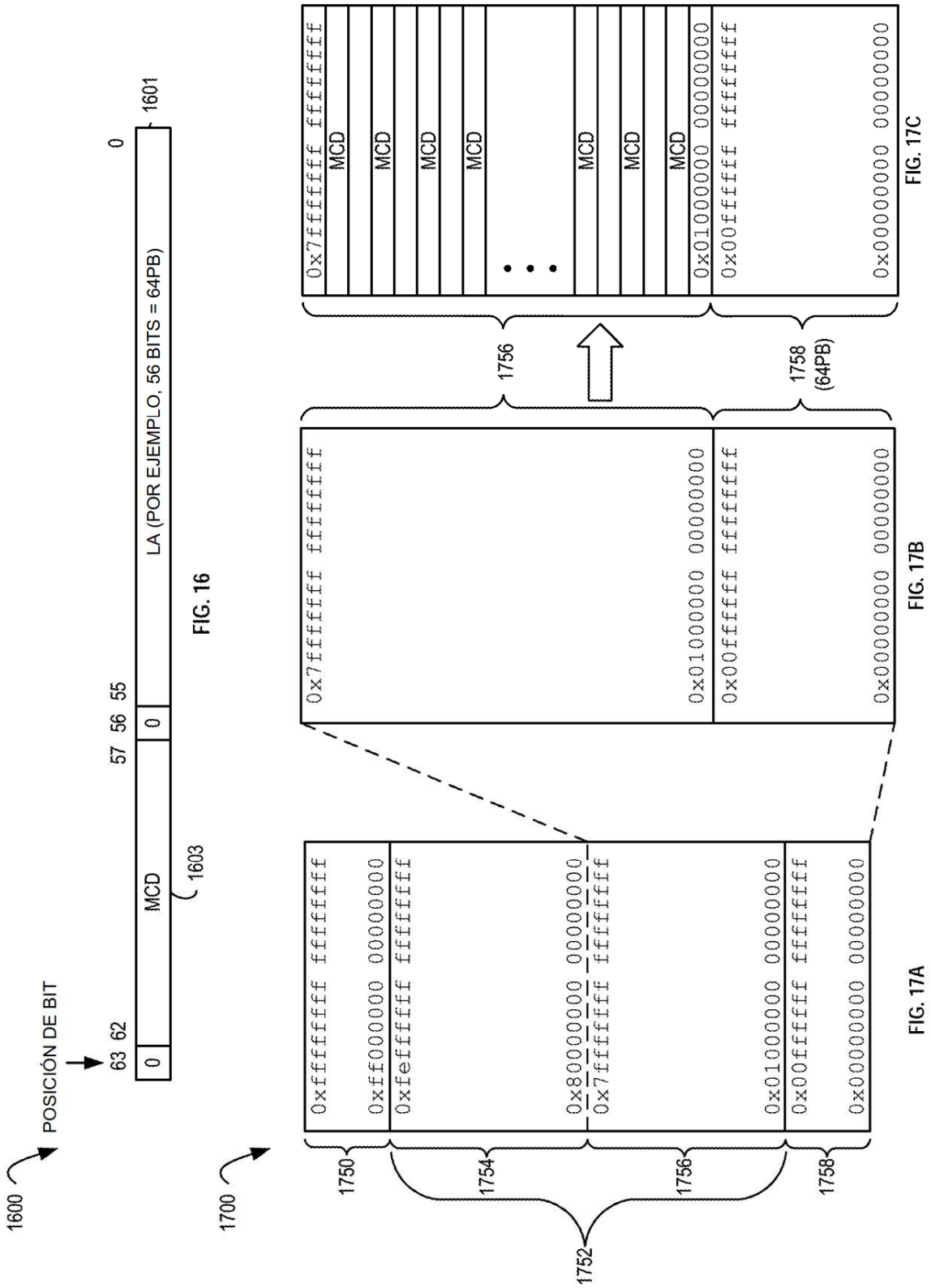

FIG. 16

FIG. 17A

FIG. 17B

FIG. 17C

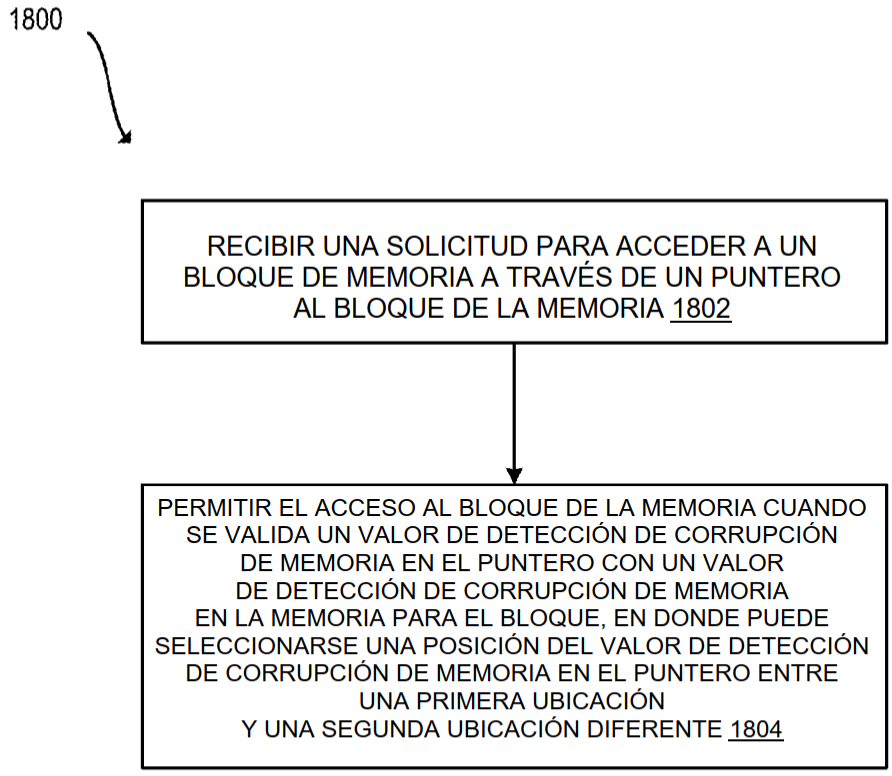
1800
RECIBIR UNA SOLICITUD PARA ACCEDER A UN BLOQUE DE MEMORIA A TRAVÉS DE UN PUNTERO AL BLOQUE DE LA MEMORIA 1802
PERMITIR EL ACCESO AL BLOQUE DE LA MEMORIA CUANDO SE VALIDA UN VALOR DE DETECCIÓN DE CORRUPCIÓN DE MEMORIA EN EL PUNTERO CON UN VALOR DE DETECCIÓN DE CORRUPCIÓN DE MEMORIA EN LA MEMORIA PARA EL BLOQUE, EN DONDE PUEDE SELECCIONARSE UNA POSICIÓN DEL VALOR DE DETECCIÓN DE CORRUPCIÓN DE MEMORIA EN EL PUNTERO ENTRE UNA PRIMERA UBICACIÓN Y UNA SEGUNDA UBICACIÓN DIFERENTE 1804

FIG. 18

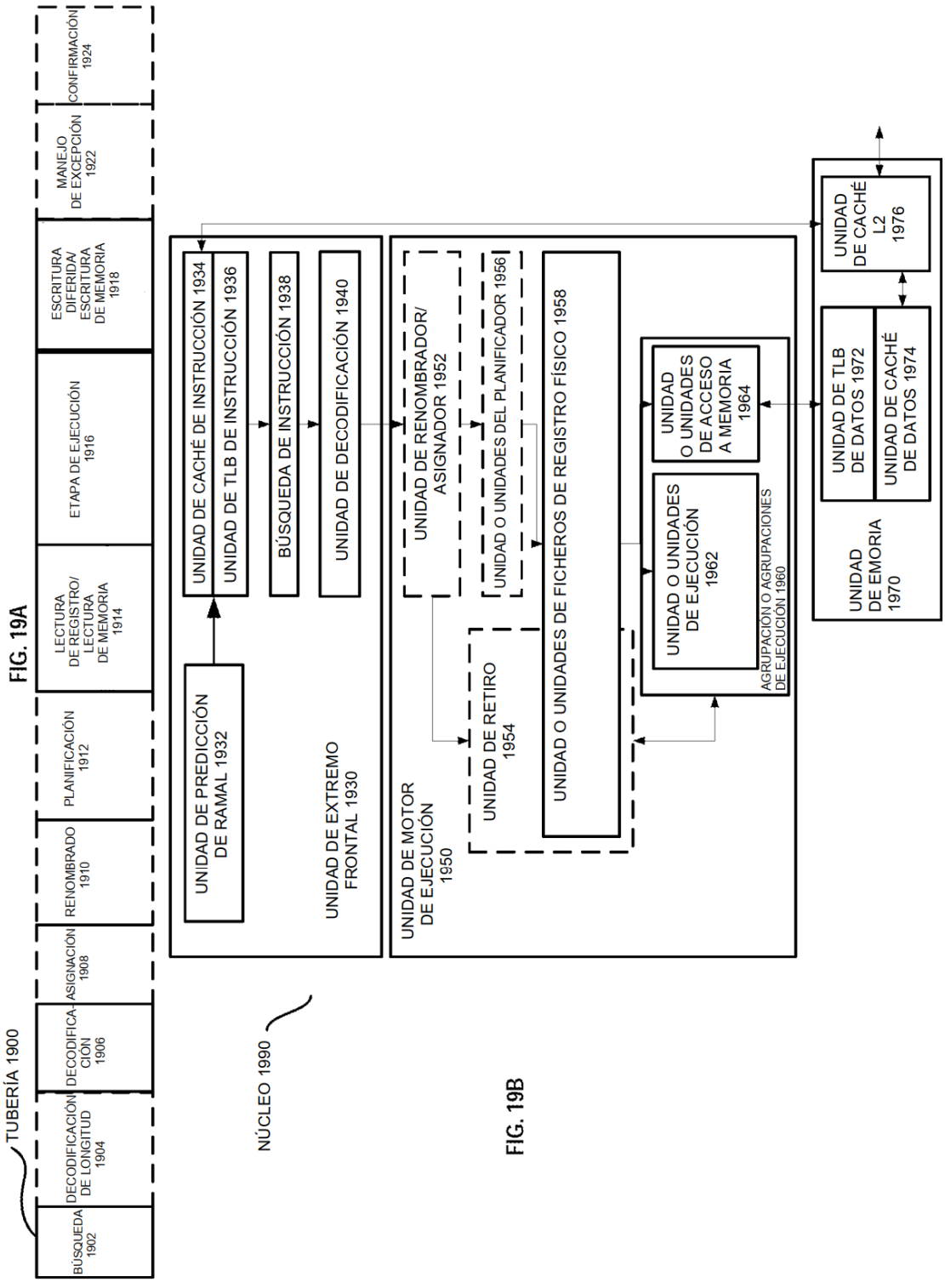

FIG. 19A

FIG. 19B

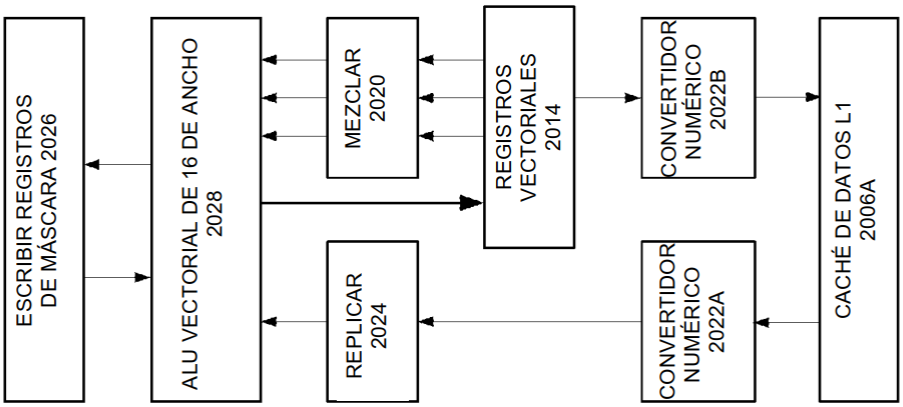

FIG. 20B

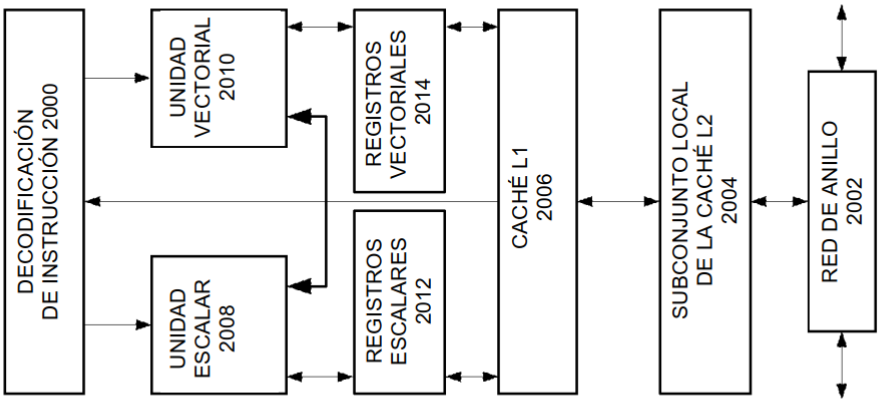

FIG. 20A

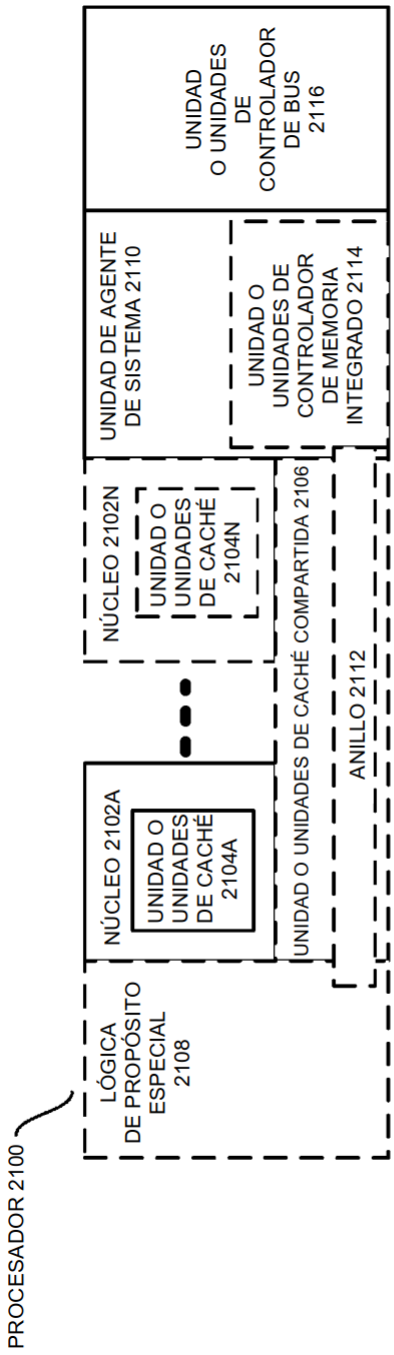
PROCESADOR 2100
LÓGICA DE PROPÓSITO ESPECIAL 2108
NÚCLEO 2102A
UNIDAD O UNIDADES DE CACHÉ 2104A
NÚCLEO 2102N
UNIDAD O UNIDADES DE CACHÉ 2104N
UNIDAD O UNIDADES DE CACHÉ COMPARTIDA 2106
ANILLO 2112
UNIDAD DE AGENTE DE SISTEMA 2110
UNIDAD O UNIDADES DE CONTROLADOR DE MEMORIA INTEGRADO 2114
UNIDAD O UNIDADES DE CONTROLADOR DE BUS 2116

FIG. 21

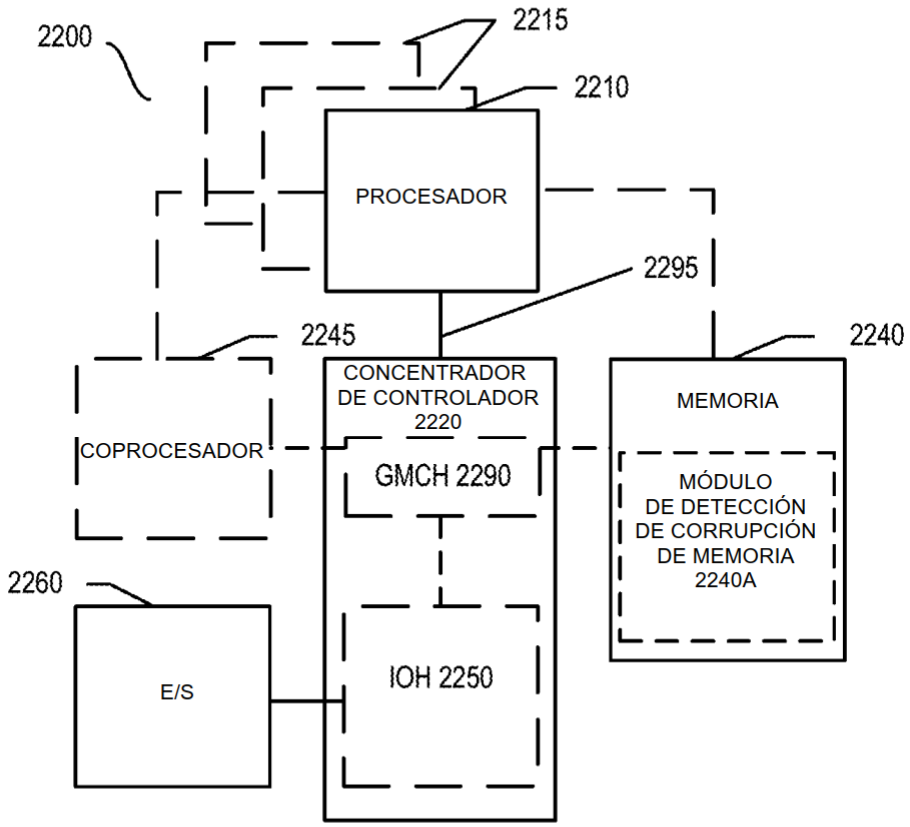
2200
2215
2210
PROCESADOR
2295
2245
2240
CONCENTRADOR
DE CONTROLADOR
2220
MEMORIA
COPROCESADOR
GMCH 2290
MÓDULO
DE DETECCIÓN
DE CORRUPCIÓN
DE MEMORIA
2240A
2260
E/S
IOH 2250

FIG. 22

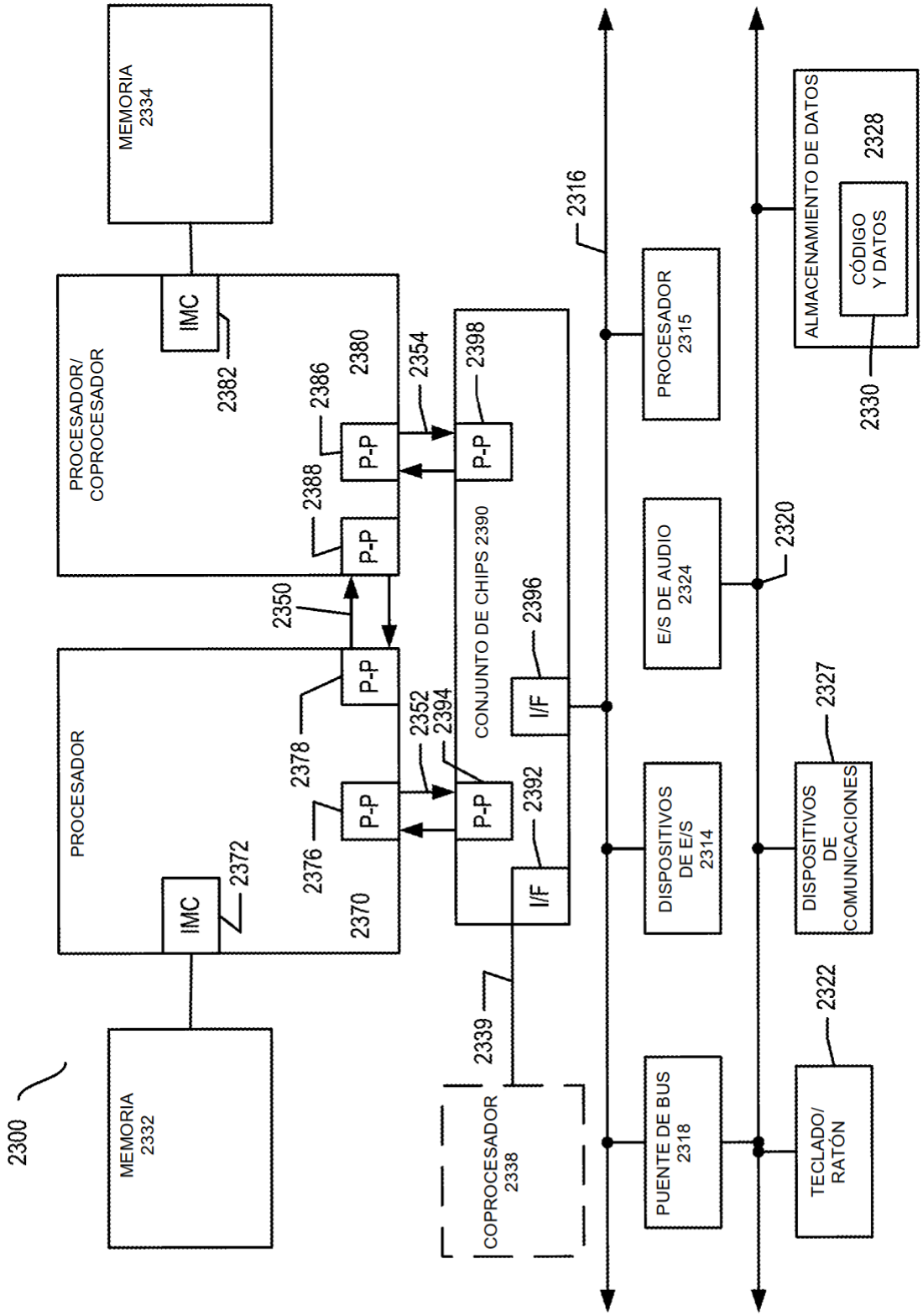
2300
MEMORIA 2332
MEMORIA 2334
PROCESADOR
IMC
2372
2370
PROCESADOR/ COPROCESADOR
IMC
2382
2376
2378
2350
2352
2394
2380
2386
2388
2354
2398
2396
2392
CONJUNTO DE CHIPS 2390
P-P
I/F
2339
COPROCESADOR 2338
2316
PROCESADOR 2315
E/S DE AUDIO 2324
2320
DISPOSITIVOS DE E/S 2314
DISPOSITIVOS DE COMUNICACIONES
2327
PUENTE DE BUS 2318
TECLADO/ RATÓN
2322
ALMACENAMIENTO DE DATOS
CÓDIGO Y DATOS
2328
2330

FIG. 23

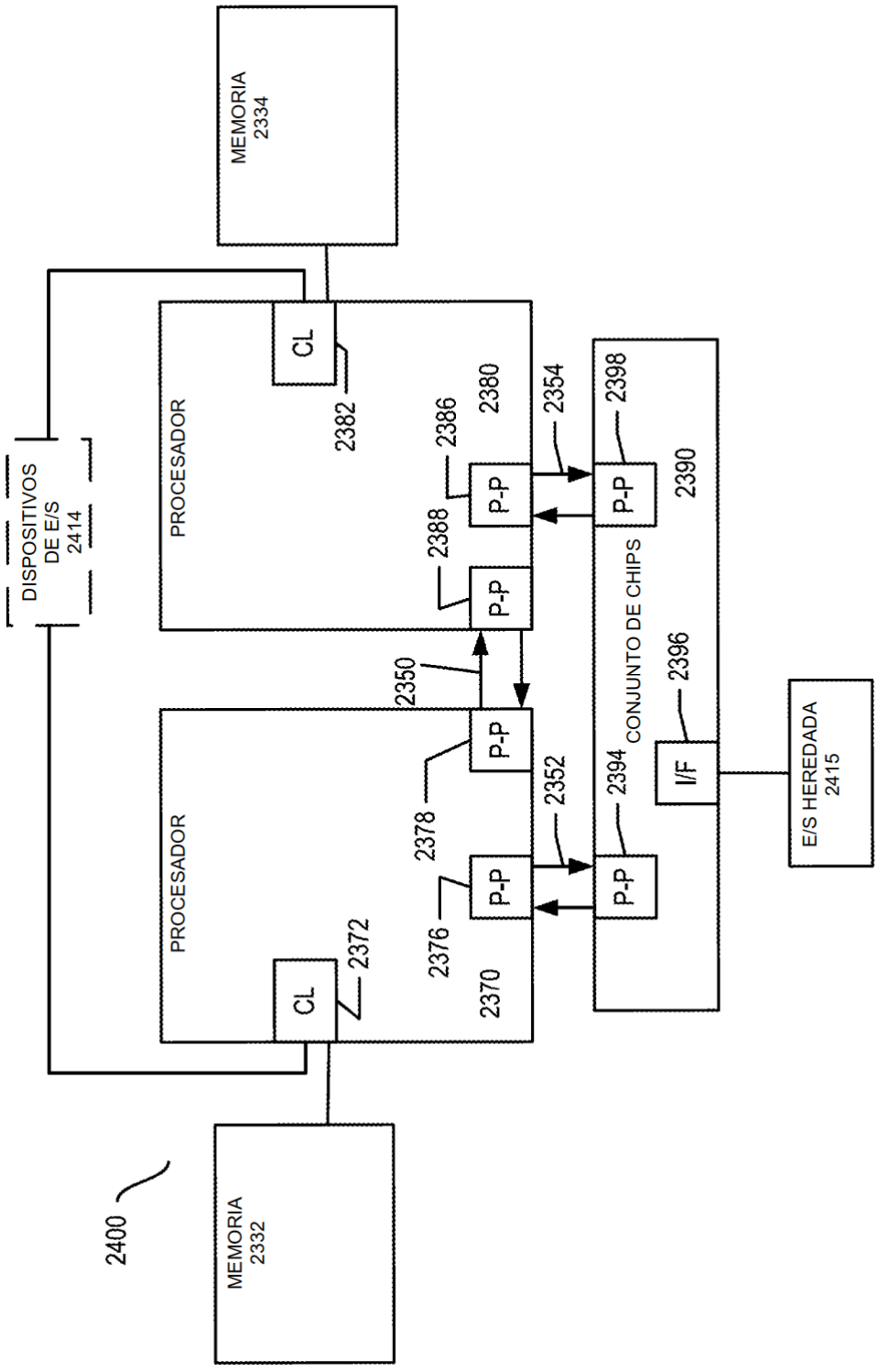
2400
MEMORIA 2332
MEMORIA 2334
DISPOSITIVOS DE E/S 2414
PROCESADOR
CL
2372
2382
2370
2376
2378
2380
2386
2388
2350
2352
2354
P-P
CONJUNTO DE CHIPS
2390
2394
2398
2396
I/F
E/S HEREDADA 2415

FIG. 24

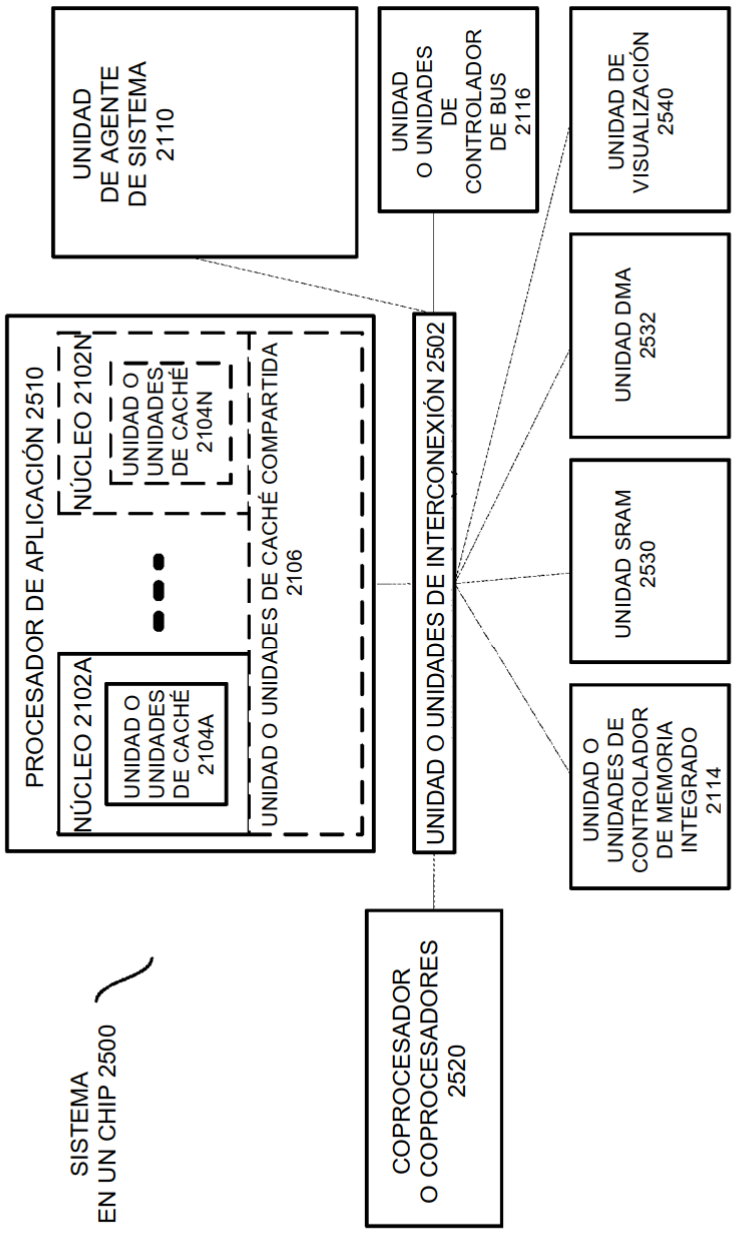
SISTEMA EN UN CHIP 2500
PROCESADOR DE APLICACIÓN 2510
NÚCLEO 2102A
UNIDAD O UNIDADES DE CACHÉ 2104A
NÚCLEO 2102N
UNIDAD O UNIDADES DE CACHÉ 2104N
UNIDAD O UNIDADES DE CACHÉ COMPARTIDA 2106
UNIDAD DE AGENTE DE SISTEMA 2110
UNIDAD O UNIDADES DE CONTROLADOR DE BUS 2116
UNIDAD O UNIDADES DE INTERCONEXIÓN 2502
UNIDAD DE VISUALIZACIÓN 2540
UNIDAD DMA 2532
UNIDAD SRAM 2530
UNIDAD O UNIDADES DE CONTROLADOR DE MEMORIA INTEGRADO 2114
COPROCESADOR O COPROCESADORES 2520

FIG. 25

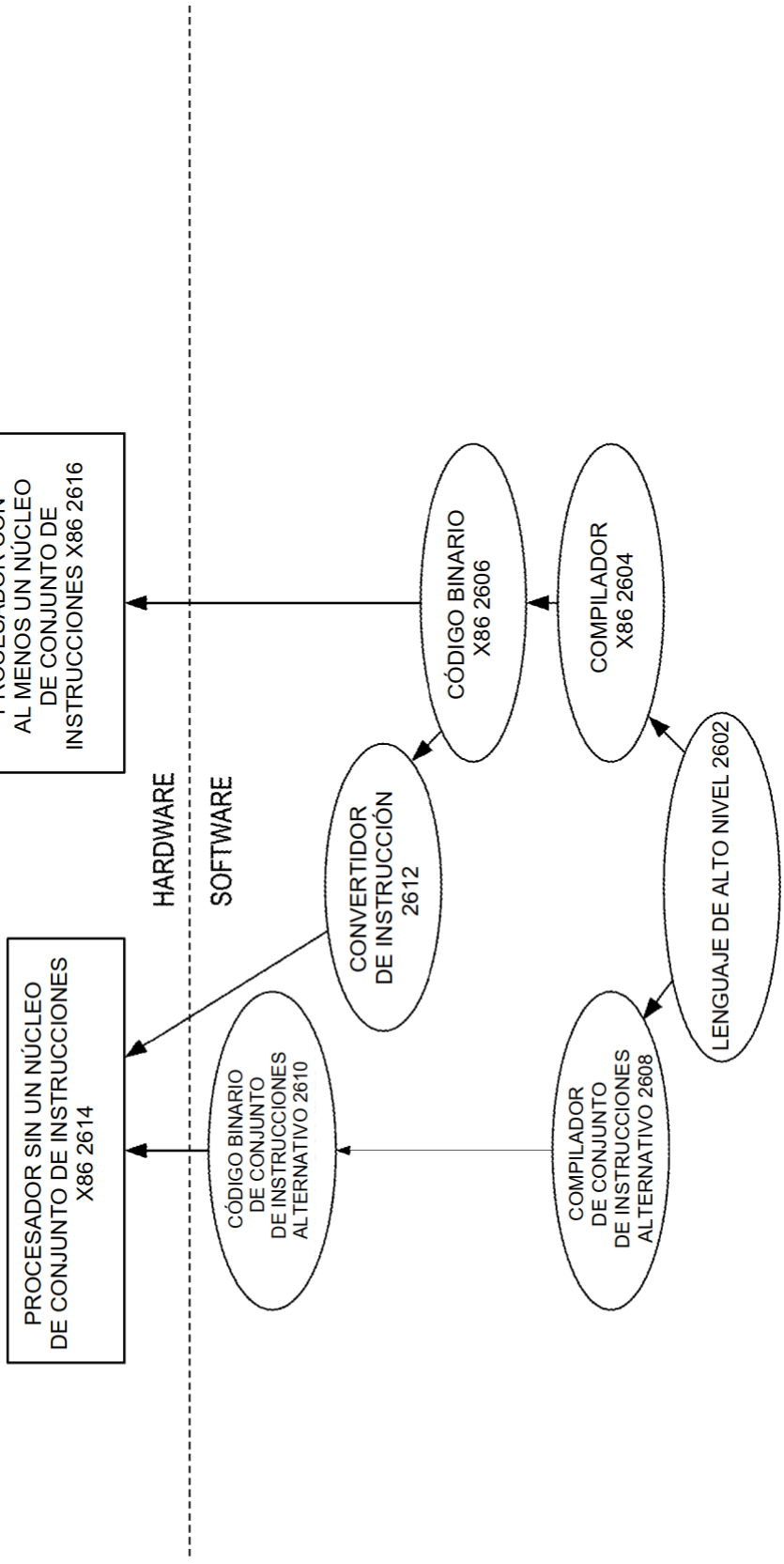
PROCESADOR CON AL MENOS UN NÚCLEO DE CONJUNTO DE INSTRUCCIONES X86 2616
HARDWARE
SOFTWARE
PROCESADOR SIN UN NÚCLEO DE CONJUNTO DE INSTRUCCIONES X86 2614
CÓDIGO BINARIO X86 2606
COMPILADOR X86 2604
CONVERTIDOR DE INSTRUCCIÓN 2612
LENGUAJE DE ALTO NIVEL 2602
CÓDIGO BINARIO DE CONJUNTO DE INSTRUCCIONES ALTERNATIVO 2610
COMPILADOR DE CONJUNTO DE INSTRUCCIONES ALTERNATIVO 2608

FIG. 26