(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 357 802**

(51) Int. Cl.:
***G06F 9/312*** (2006.01)
***G06F 12/10*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **05108510 .8**
(96) Fecha de presentación : **06.05.2004**
(97) Número de publicación de la solicitud: **1653343**
(97) Fecha de publicación de la solicitud: **03.05.2006**

(54) Título: **Invalidación de almacenamiento, borrado de elementos de la memoria intermedia.**

(30) Prioridad: **12.05.2003 US 435919**

(45) Fecha de publicación de la mención BOPI:
**29.04.2011**

(45) Fecha de la publicación del folleto de la patente:
**29.04.2011**

(73) Titular/es: **INTERNATIONAL BUSINESS MACHINES CORPORATION**
**New Orchard Road**
**Armonk, New York 10504, US**

(72) Inventor/es: **Slegel, Timothy;**
**Heller, Lisa;**
**Pfeffer, Erwin y**
**Plambeck, Kenneth**

(74) Agente: **Elzaburu Márquez, Alberto**

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



Venta de fascículos: Oficina Española de Patentes y Marcas. Pº de la Castellana, 75 – 28071 Madrid

**DESCRIPCIÓN**

Esta invención está relacionada, en general, con el proceso dentro de un entorno informático y, en particular, con uno o más de invalidación de almacenamiento, borrado de elementos de la memoria intermedia y una instrucción para ellos.

Los entornos informáticos existentes utilizan el almacenamiento virtual que es mantenido normalmente en almacenamiento auxiliar para aumentar el tamaño del almacenamiento disponible. Cuando se hace referencia a una página de almacenamiento virtual, la dirección virtual utilizada cuando se hace referencia a esta página es traducida, por medio de una traducción dinámica de direcciones, en una dirección real de almacenamiento. Si la traducción ha tenido éxito, el almacenamiento virtual es válido; en otro caso, es inválido. Una página de almacenamiento virtual está indicada como válida o inválida por medio de un bit de invalidación en un elemento de la tabla de páginas, incluyendo este elemento una dirección real de almacenamiento si la página es válida.

Las páginas de almacenamiento pueden ser invalidadas de una en una, tal como se describe en el documento US 6119204. Por ejemplo, en un entorno informático basado en la arquitectura/z, ofrecida por International Business Machines Corporation, se utiliza una instrucción, denominada como Instrucción de Elemento de Invalidación de La Tabla de Páginas (IPTE), para invalidar una página de almacenamiento. La invalidación incluye la colocación de un indicador de invalidación en una estructura de control, asociado con el almacenamiento virtual y situado en el almacenamiento físico principal. En el almacenamiento físico principal, para indicar que no puede traducirse una dirección virtual de un lugar del almacenamiento virtual, por medio de una traducción dinámica de direcciones, en una dirección de almacenamiento físico principal, denominada también dirección real. Además, se utiliza la instrucción para borrar los elementos de la memoria intermedia interna a la máquina, asociados con la página, pudiendo utilizarse esos elementos durante la traducción dinámica de direcciones para evitar el acceso a estructuras de control en el almacenamiento físico principal. Por ejemplo, en la arquitectura/z, existen Memorias Intermedias de Traducción Adelantada, que se utilizan durante la traducción dinámica de direcciones de las direcciones virtuales a direcciones reales. Cuando se invalida una página de almacenamiento, se borran también las correspondientes memorias intermedias de elementos asociados con esa página.

También hay instrucciones que permiten la purga de todas las Memorias Intermedias de Traducción Adelantada. Por ejemplo, en la arquitectura/z, una instrucción, denominada instrucción de Comparar e Intercambiar y Purgar (CSP), purga todos los elementos de las Memorias Intermedias de Traducción Adelantada.

Así, aunque hay instrucciones para invalidar una página de almacenamiento y limpiar las memorias intermedias de elementos asociados con esa página, e instrucciones para limpiar todas las memorias intermedias, sigue existiendo la necesidad de instrucciones que puedan invalidar selectivamente diversos tamaños de unidades de almacenamiento y/o limpiar las memorias intermedias de elementos asociados con ellas. Por ejemplo, existe la necesidad de una instrucción que permita la especificación de una unidad seleccionada de almacenamiento mayor que una página para que sea invalidada y/o borre los elementos de la memoria intermedia asociados con ella. Existe una necesidad adicional de una instrucción que permita borrar los elementos de la memoria intermedia para unidades de almacenamiento seleccionadas, sin afectar elementos de memoria intermedia de otras unidades de almacenamiento.

La presente invención proporciona un método para invalidar el almacenamiento de un entorno informático, como se reivindica en la reivindicación 1.

Preferiblemente, la presente invención proporciona un método para invalidar el almacenamiento de un entorno informático. El método incluye, por ejemplo, especificar una unidad de almacenamiento a invalidar, incluyendo la unidad de almacenamiento una pluralidad de páginas de almacenamiento, y donde la especificación incluye proporcionar un descriptor de la unidad de almacenamiento; e invalidar la unidad de almacenamiento.

Preferiblemente, la presente invención proporciona un método para facilitar el borrado de las memorias intermedias de un entorno informático. El método incluye, por ejemplo, identificar una unidad de almacenamiento para la cual se ha de efectuar el borrado de uno o más elementos de una o más memorias intermedias, donde la unidad de almacenamiento incluye una pluralidad de páginas de almacenamiento; y borrar uno o más elementos de una o más memorias intermedias asociadas con la unidad de almacenamiento identificada, donde uno o más elementos asociados con otra unidad de almacenamiento no son borrados.

Preferiblemente, se proporciona una instrucción que ha de ejecutarse dentro de un entorno informático. La instrucción incluye, por ejemplo, un código de operación para identificar una instrucción a ejecutar; una designación de invalidación para especificar la información asociada con una unidad de almacenamiento a invalidar, incluyendo la unidad de almacenamiento una pluralidad de páginas de almacenamiento; otra designación para especificar información adicional para utilizar por la instrucción; y una designación de borrado para especificar información asociada con una de las unidades de almacenamiento y otra unidad de almacenamiento para la cual han de borrarse uno o más elementos de una o más memorias intermedias, donde uno o más componentes de una o más de las designaciones de invalidación, de la otra designación y de la designación de borrado, son configurables como no-operaciones.

Preferiblemente, se proporciona una instrucción a ejecutar dentro de un entorno informático. La instrucción incluye, por ejemplo, un código de operación para identificar una instrucción a ejecutar; y al menos una de las siguientes:

una primera designación de invalidación para especificar la información asociada con una unidad de almacenamiento a invalidar, incluyendo la unidad de almacenamiento una pluralidad de páginas de almacenamiento; una segunda designación de invalidación para especificar información adicional asociada con la unidad de almacenamiento a invalidar; y una designación de borrado para especificar información asociada con una de las unidades de almacenamiento y otra unidad de almacenamiento para la cual han de borrarse uno o más elementos de una o más memorias intermedias, donde la información especificada por la designación de borrado no está asociada con al menos otra unidad de almacenamiento, para la cual no han de borrarse uno o más elementos.

En esta memoria se describen y reivindican también productos de sistemas y programas de ordenador correspondientes a los métodos resumidos anteriormente.

Se realizan características y ventajas adicionales por medio de las técnicas de la presente invención. Otros modos de realización y aspectos de la invención se describen también en detalle en esta memoria y se consideran parte de la invención reivindicada.

La materia objeto que se considera como la invención, está particularmente indicada y reivindicada separadamente en las reivindicaciones del final de la memoria. Los anteriores y otros objetos, características y ventajas de la invención son evidentes a partir de la siguiente descripción detallada, tomada conjuntamente con los dibujos que se acompañan, en los cuales:

La figura 1 representa un modo de realización de un entorno informático que incorpora y utiliza uno o más aspectos de la presente invención;

La figura 2 representa un modo de realización de detalles adicionales asociados con un controlador de la figura 1, de acuerdo con un aspecto de la presente invención;

La figura 3 representa un modo de realización de un ordenador central que puede emular a otro ordenador, de acuerdo con un aspecto de la presente invención;

La figura 4 representa un modo de realización de la lógica asociada con el almacenamiento de invalidación, de acuerdo con un aspecto de la presente invención;

La figura 5 representa un modo de realización de la lógica asociada con el borrado de elementos de la memoria intermedia, de acuerdo con un aspecto de la presente invención;

La figura 6a representa un modo de realización de un formato de una instrucción de Invalidación de Elementos de la Tabla de Traducción Dinámica de Direcciones (IDTE), de acuerdo con un aspecto de la presente invención;

La figura 6b representa un modo de realización de los campos asociados con el registrador general R1 de la instrucción IDTE de la figura 6a, de acuerdo con un aspecto de la presente invención;

La figura 6c representa un modo de realización de los campos asociados con el registrador general R2 de la instrucción IDTE de la figura 6a, de acuerdo con un aspecto de la presente invención;

La figura 6d representa un modo de realización de los campos asociados con el registrador general R3 de la instrucción IDTE de la figura 6a, de acuerdo con un aspecto de la presente invención;

Las figuras 7a-7b representan un modo de realización de la lógica asociada con la instrucción IDTE, de acuerdo con un aspecto de la presente invención;

La figura 8a representa un modo de realización de los campos del registrador general R1 que se utiliza para una operación alternativa de la instrucción IDTE, de acuerdo con un aspecto de la presente invención;

La figura 8b representa un modo de realización de los campos del registrador general R2 que se utiliza para una operación alternativa de la instrucción IDTE, de acuerdo con un aspecto de la presente invención;

La figura 8c representa un modo de realización de los campos del registrador general R3 que se utiliza para una operación alternativa de la instrucción IDTE, de acuerdo con un aspecto de la presente invención; y

La figura 9 representa un modo de realización de la lógica asociada con la operación alternativa de la instrucción IDTE, de acuerdo con un aspecto de la presente invención.

De acuerdo con un aspecto de la presente invención, se invalida una parte de almacenamiento especificada (por ejemplo, un almacenamiento virtual). Como ejemplos, se selecciona un segmento de almacenamiento, que incluye una pluralidad de páginas de almacenamiento o una región de almacenamiento, que incluye una pluralidad de segmentos de almacenamiento, para ser invalidado. La invalidación se facilita colocando indicadores de invalidación situados en elementos de la estructura de datos, correspondientes a las unidades de almacenamiento a invalidar. En un aspecto adicional de la presente invención, se borran los elementos de la memoria intermedia (por ejemplo, los elementos de la

Memoria Intermedia de Tablas de Paginación de Traducciones) asociados con el almacenamiento a invalidar, u otro almacenamiento. Más aún, en otro aspecto de la presente invención, los elementos de la memoria intermedia asociados con espacios de direcciones seleccionadas, son borrados de las memorias intermedias sin afectar a los elementos de la memoria intermedia de espacios de direcciones no seleccionadas, y sin realizar la invalidación. En un ejemplo, una instrucción, denominada en esta memoria Instrucción de Elementos de Invalidación de la Tabla (IDTE) de Traducción Dinámica de Direcciones (DAT), de la arquitectura/z de IBM, se utiliza para realizar una o más de las operaciones anteriores.

Se describe un modo de realización del entorno informático 100, que incorpora y utiliza uno o más aspectos de la presente invención, con referencia a la figura 1. El entorno informático 100 está basado, por ejemplo, en la arquitectura/z ofrecida por International Business Machines Corporation, Armonk, Nueva York. La arquitectura/z está descrita en una publicación de IBM® titulada "z/Arquitecture Principles of Operation (Principios de funcionamiento de la arquitectura/z)", publicación de IBM núm. SA22-7832-00, Diciembre de 2000, (IBM® es una marca registrada de International Business Machines Corporation, Armonk, Nueva York, Estados Unidos. Otros nombres utilizados en esta memoria pueden ser marcas comerciales registradas, marcas o nombre de productos registrados de International Business Machines Corporation o de otras compañías). En un ejemplo, un entorno informático basado en la arquitectura/z incluye un eServer zSeries, ofrecidos por International Business Machines Corporation, Armonk, Nueva York. Como ejemplo, el entorno informático 100 incluye un complejo 102 de procesador central (CPC) acoplado a un controlador 120. El complejo 102 de procesador central incluye, por ejemplo, una o más particiones 104 (por ejemplo, las particiones lógicas LP1 – LPn), uno o más procesadores centrales 106 (por ejemplo, CP1 – CPm), y un hipervisor 108 (por ejemplo un gestor de particiones lógicas), cada uno de los cuales se describe a continuación.

Cada partición lógica 104 es capaz de funcionar como un sistema independiente. Es decir, puede hacerse una reposición de cada partición lógica independientemente, inicialmente cargada con un sistema operativo, si se desea, y funcionar con programas diferentes. Un sistema operativo o un programa de aplicación ejecutándose en una partición lógica parece tener acceso a un sistema total y completo, pero en realidad, solamente está disponible una parte de él. Una combinación de hardware y Código Interno Licenciado (comúnmente denominado microcódigo) mantiene un programa en una partición lógica sin interferir con un programa en una partición lógica diferente. Esto permite a varias particiones lógicas diferentes funcionar sobre uno solo o múltiples procesadores físicos, de una manera que realiza intervalos de tiempo. Es este ejemplo particular, cada partición lógica tiene un sistema operativo residente 110, que puede diferir de una o más particiones lógicas. En un modo de realización, el sistema operativo 110 es el sistema operativo z/OS, ofrecido por International Business Machines Corporation, Armonk, Nueva York.

Los procesadores centrales 106 son recursos de procesadores físicos que están asignados a las particiones lógicas. Por ejemplo, una partición lógica 104 incluye uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte de un recurso físico 106 de un procesador asignado a la partición. Los procesadores lógicos de una partición particular 104, pueden estar dedicados a la partición, de manera que el recurso de procesador subyacente está reservado para esa partición; o bien compartidos con otra partición, de manera que el recurso de procesador subyacente está potencialmente disponible para otra partición. Las particiones lógicas 104 están gestionadas por el hipervisor 108, implementadas por microcódigo que se ejecuta en los procesadores 106. Las particiones lógicas 104 y el hipervisor 108 comprenden, cada una de ellas, uno o más programas residentes en respectivas partes del almacenamiento central asociado con los procesadores centrales. Un ejemplo de hipervisor 108 es el Gestor de Recursos/Sistemas de Procesador (PR/SM), ofrecido por International Business Machines Corporation, Armonk, Nueva York.

El controlador 120, que está acoplado al complejo de procesador central, incluye una lógica centralizada responsable del arbitraje entre distintos procesadores que emiten peticiones. Por ejemplo, cuando el controlador 120 recibe una petición, determina si el solicitante es el procesador maestro para esa petición, y que los demás procesadores son procesadores esclavos; transmite los mensajes; y por otra parte gestiona las peticiones. Un ejemplo de controlador está descrito en la patente de Estados Unidos núm. 6.199.219. Otros detalles se describen también con referencia a la figura 2. La figura 2 representa un ejemplo de controlador 200 acoplado a una pluralidad de procesadores centrales (CPU) 201. En este ejemplo, se representan dos procesadores centrales. Sin embargo, se comprenderá que pueden acoplarse más de dos procesadores al controlador 200.

El controlador 200 incluye diversos controles que incluyen, por ejemplo, controles 202 para la puesta en serie del sistema. Los controles de la puesta en serie del sistema se utilizan para asegurar que las operaciones que han de ser puestas en serie, tal como las instrucciones de invalidación, son puestas en serie, en cuanto que solamente una de tales instrucciones se está ejecutando en un momento dado en el entorno informático. También supervisa la secuencia de eventos para esa operación.

El controlador 200 está acoplado a cada procesador central a través de diversos interfaces. Por ejemplo, se utiliza un interfaz 204 por el Código Licenciado Interno de un procesador central para enviar órdenes de "control" al controlador, que especifican una acción a tomar, y a enviar órdenes de "detección", que devuelven información desde el controlador. Otro interfaz es un bus 206 de respuesta, que se utiliza para devolver información desde el controlador para las órdenes de "detección". El bus de respuesta se utiliza también para comunicar el estatus de las órdenes de "control", y puede ser activado desde una pluralidad de fuentes dentro del controlador, incluyendo los controles de puesta en serie del sistema. Un procesador central puede utilizar este interfaz para detectar el estado de los controles de puesta en serie

del sistema en el controlador 200.

Un interfaz adicional es el interfaz 208, que lo utiliza el controlador para enviar órdenes a cada CPU. Éste puede ser controlado también desde una pluralidad de fuentes, dentro del controlador, incluyendo los controles 202 de puesta en serie del sistema. Un interfaz adicional más es el interfaz 210, que proporciona señales a los controles 212 de la caché del procesador central 202. Los controles 212 de la caché procesan las órdenes, como respuesta a las señales. En un ejemplo, los controles 212 de la caché procesan las órdenes que afectan a una o más memorias intermedias, tal como las Memorias Intermedias de Traducción Adelantada (TLB) 213, como se describe con más detalles más adelante.

Además de los controles 212 de la caché, el procesador central 201 incluye otros controles diversos, que incluyen, por ejemplo, controles 220 de interrupción y controles 222 de ejecución. Como respuesta a eventos particulares, los controles 220 de interrupción originan que quede pendiente una interrupción interna en la CPU, lo cual a su vez hace que los controles 222 de ejecución suspendan el proceso de instrucciones de programa, en el siguiente punto que pueda interrumpirse. Como respuesta a la interrupción, los controles 222 de ejecución invocan una rutina de un Código Licenciado Interno para activar un circuito bloqueador 224 que tiene permiso para una operación de transmisión, para permitir que los controles 212 de la caché procesen las órdenes pendientes.

El procesador central 201 incluye también un circuito bloqueador 226 de desactivación de la CPU, que indica si el procesador está o no desactivado.

El entorno informático anteriormente descrito es solamente un ejemplo. Por ejemplo, puede haber una o más particiones corriendo en modos de arquitecturas diferentes. Además, como un ejemplo más, el entorno no necesita estar basado en la arquitectura/z, sino que en lugar de eso, puede estar basado en otras arquitecturas ofrecidas por Intel, Sun Microsystems, así como en otras. Más aún, un entorno puede incluir un emulador (por ejemplo, software u otros mecanismos de emulación), en los cuales se emula una arquitectura particular o subconjunto de la misma. En tal entorno, una o más funciones de emulación del emulador pueden implementar uno o más aspectos de la presente invención, aún cuando un ordenador que esté ejecutando el emulador pueda tener una arquitectura diferente a las capacidades que son emuladas. Como ejemplo, en modo de emulación, se descodifica la instrucción u operación específicas que están siendo emuladas, y se construye una función de emulación apropiada para implementar la instrucción u operación individuales.

Con referencia a la figura 3, se describen detalles adicionales de un entorno de emulación. Como ejemplo, un ordenador central 300 es capaz de emular otra arquitectura, ordenador y/o capacidades de proceso de otro ordenador. Por ejemplo, el ordenador central 300 está basado en una arquitectura Intel; una arquitectura RISC tal como el PowerPC (o PC de potencia); una arquitectura SPARC, ofrecida por Sun Microsystems; u otra arquitectura, y es capaz de emular la Arquitectura/z de IBM® u otra arquitectura de IBM® o de otra entidad. El ordenador central 300 incluye, por ejemplo, una memoria 302 para almacenar instrucciones y datos; una unidad 304 de extracción de instrucciones para extraer instrucciones desde la memoria 302, y para proporcionar, opcionalmente, un almacenamiento intermedio local para las instrucciones extraídas; una unidad 306 de descodificación de instrucciones, para recibir instrucciones desde la unidad 304 de extracción de instrucciones y para determinar el tipo de instrucciones que han sido extraídas; y una unidad 308 de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir la carga de datos en un registrador de la memoria 302; devolver los datos almacenados en la memoria para almacenarlos en el registrador; o realizar algún tipo de operación lógica o aritmética, como determine la unidad de descodificación.

En un ejemplo, cada unidad descrita anteriormente se implementa por software. Por ejemplo, las operaciones que son realizadas por las unidades son implementadas como una o más subrutinas dentro del software del emulador. En otro ejemplo, una o más de las operaciones se implementan en firmware (programación fija), hardware, software o alguna combinación de los mismos.

Además, aunque la figura 3 se describe con referencia a la emulación, el entorno de la figura 3 no necesita ser un entorno de emulación. En otro ejemplo, las instrucciones se ejecutan en un entorno nativo y las operaciones se implementan en hardware, firmware, software o alguna combinación de los mismos.

Un entorno informático puede incluir almacenamiento virtual, así como un almacenamiento principal. El almacenamiento virtual puede exceder por mucho el tamaño del almacenamiento principal disponible en la configuración y se mantiene normalmente en un almacenamiento auxiliar. El almacenamiento virtual se considera estar compuesto por bloques de direcciones, denominados páginas. Las páginas referenciadas más recientemente de almacenamiento virtual están asignadas a la ocupación de bloques de almacenamiento físico principal. Cuando un usuario hace referencia a páginas de almacenamiento virtual que no aparecen en el almacenamiento principal, se traen las páginas virtuales para sustituir las páginas del almacenamiento principal, que tienen menos probabilidad de ser utilizadas. El intercambio de páginas del almacenamiento puede ser realizado por el sistema operativo sin conocimiento del usuario.

Las direcciones utilizadas para designar lugares del almacenamiento virtual son denominadas direcciones virtuales. Un bloque de direcciones virtuales secuenciales que abarque, por ejemplo, hasta 4k bytes, es denominado una página. De forma similar, un bloque de páginas virtuales secuenciales que abarque, por ejemplo, hasta 1 Megabyte, es denominado segmento; y un bloque de segmentos virtuales secuenciales que abarque, por ejemplo, hasta 2 Gigabytes, es denominado una región. Además, una secuencia de direcciones virtuales asociada con almacenamiento virtual al que

apunta un elemento de control de espacio de direcciones (ASCE) es denominado un espacio de direcciones. Los espacios de direcciones pueden ser utilizados para proporcionar grados de aislamiento entre usuarios. Un espacio de direcciones puede incluir una o más regiones, uno o más segmentos, una o más páginas, o alguna combinación de los mismos.

Asociadas con los diferentes tipos de unidades de almacenamiento, (por ejemplo regiones, segmentos, páginas) hay estructuras de datos a utilizar en el proceso asociado con las unidades de almacenamiento. Por ejemplo, asociadas con las regiones hay tablas de regiones; asociadas con los segmentos hay tablas de segmentos; y asociadas con las páginas hay tablas de páginas. Estas tablas se utilizan, por ejemplo, durante la traducción (por ejemplo, la Traducción Dinámica de Direcciones) de una dirección virtual a una dirección real, que se utiliza para acceder al almacenamiento principal. Las tablas que han de utilizarse en la traducción, denominadas en esta memoria tablas de traducción, están designadas por un elemento de control de espacios de direcciones (ASCE). Esto está descrito con más detalles en la publicación de IBM® titulada “z/Arquitecture Principles of Operation (Principios de funcionamiento de la arquitectura/z)”, publicación de IBM núm. SA22-7832-00, Diciembre de 2000. Una unidad de almacenamiento virtual que no esté actualmente asignada al almacenamiento principal, se denomina inválida. El estado de invalidez de una unidad de almacenamiento virtual está indicado por un indicador inválido en la estructura de datos asociada con la unidad.

El mecanismo de traducción dinámica de direcciones se implementa, en un modo de realización, de forma tal que la información obtenida a partir de las tablas de traducción (por ejemplo, las tablas de regiones, las tablas de segmentos y/o las tablas de páginas), a través de procesos DAT, se mantiene en una o más memorias intermedias situadas dentro de los procesadores, denominadas en esta memoria como Memorias Intermedias de Traducción Adelantada, con el fin de mejorar el rendimiento de la traducción de direcciones. Es decir, durante la traducción, se comprueba si las memorias intermedias tienen la información necesaria, y después, si la información no está en las memorias intermedias, se accede a una o más tablas de traducción.

En un modo de realización, un elemento de memoria intermedia se considera como uno de los tres tipos: un elemento de tabla TLB combinada de regiones y segmentos; un elemento de tabla TLB de páginas; o un elemento de espacio real de TLB, cada uno de los cuales se describe a continuación.

Un elemento de tabla TLB combinada de regiones y segmentos (CRSTE) incluye la información obtenida a partir del elemento o elementos de la tabla, en un almacenamiento real o absoluto, y los atributos utilizados para extraer la información del almacenamiento. En un ejemplo, un elemento de tabla TLB combinada de regiones y segmentos (CRSTE) incluye los siguientes campos:

TO El origen de la tabla en el elemento de control del espacio de direcciones, utilizado para formar el elemento;

RX El índice de la región de la dirección virtual utilizada para formar el elemento;

SX El índice del segmento de la dirección virtual utilizada para formar el elemento;

PTO El origen de la tabla de páginas extraídas de la tabla de segmentos, en el almacenamiento real o absoluto;

C El bit de segmento común extraído del elemento de tablas de segmentos, en almacenamiento real o absoluto; y

P El bit de protección de páginas extraído del elemento de tablas de segmentos, en almacenamiento real o absoluto.

V El bit de validez indica si el elemento de la TLB es válido y, por tanto, puede ser utilizado para consultar elementos subsiguientes.

Un elemento de la tabla de páginas TLB incluye la información obtenida desde el elemento o elementos de la tabla, en almacenamiento real o absoluto, y los atributos utilizados para extraer esta información desde el almacenamiento. En un ejemplo, un elemento de la tabla de páginas TLB incluye:

TO El origen de la tabla en el elemento de control del espacio de direcciones o el elemento de tabla TLB combinada de regiones y segmentos utilizado para formar el elemento, dependiendo de cómo se formó el elemento;

PTO El origen de la tabla de páginas utilizado para formar el elemento;

PX El índice de páginas de la dirección virtual utilizado para formar el elemento;

PFRA La dirección real del marco de página extraída del elemento de tablas de páginas en almacenamiento real o absoluto; y

P El bit de protección de páginas extraído del elemento de tabla de páginas en almacenamiento real o absoluto.

V El bit de validez indica si el elemento de la TLB es válido y, por tanto, puede ser utilizado para consultar

elementos subsiguientes.

A veces, ha de invalidarse una unidad de almacenamiento. Para facilitar esta invalidación, se emplea la tabla de traducciones asociada con esa unidad de almacenamiento, como se ha descrito en esta memoria. Además, como respuesta a la invalidación del almacenamiento, pueden borrarse los correspondientes elementos de la memoria intermedia. Por ejemplo, con el fin de invalidar una página de almacenamiento, se fija un bit inválido de un elemento de la tabla de páginas. Además, en un ejemplo, se borran atómicamente los correspondientes elementos de una o más TLB. En un ejemplo particular, la invalidación y borrado de elementos asociados con las páginas de almacenamiento se realizan por medio de una instrucción de Elementos de Invalidación de la Tabla de Páginas (IPTE).

Además de invalidar páginas de almacenamiento, de acuerdo con un aspecto de la presente invención, pueden invalidarse segmentos de almacenamiento y/o regiones de almacenamiento. Como ejemplo, las tablas de segmentos y/o regiones se utilizan en esta invalidación. Se describe un modo de realización de la lógica asociada con la invalidación de una unidad particular de almacenamiento, con referencia a la figura 4.

Inicialmente, se identifica una tabla de regiones o segmentos en la cual ha de invalidarse uno o más elementos, PASO 400. En un ejemplo, esto incluye proporcionar un origen de la tabla de regiones o un origen de la tabla de segmentos. Después, se identifica para la invalidación un elemento dentro de la tabla de regiones o de segmentos designada, PASO 402, y se invalida el elemento, PASO 404. En un ejemplo, la identificación se hace proporcionando un índice de la región o un índice del segmento (bits de una dirección virtual) que selecciona un elemento dentro de la tabla identificada, y la invalidación incluye colocar un indicador inválido dentro del elemento. Como respuesta a la colocación de un indicador inválido, se invalida la correspondiente unidad de almacenamiento (por ejemplo, una región o un segmento). Después, se hace una determinación sobre si hay más elementos a invalidar, PREGUNTA 406. Si hay más elementos a invalidar, el proceso continúa con el PASO 402. En otro caso, el proceso termina.

Además de invalidar el uno o más elementos de una tabla de regiones o de segmentos, y por tanto de invalidar esas unidades de almacenamiento, los elementos correspondientes (u otros elementos) pueden ser borrados o purgados de una o más memorias intermedias en uno o más procesadores, dentro del entorno. Un modo de realización de la lógica asociada con este borrado está descrito con más detalle con referencia a la figura 5.

Inicialmente, se hace una determinación sobre si se ha especificado un origen de la tabla de regiones o segmentos, PREGUNTA 500. Si se ha especificado un origen de la tabla de regiones o segmentos, se borra una o más memorias intermedias de un elemento especificado por el índice de la región o segmento, PASO 502. Subsiguientemente, se hace una determinación sobre si hay más elementos a borrar, PREGUNTA 504. Si es así, el proceso continúa en el PASO 502. En otro caso, el proceso termina.

Volviendo a la PREGUNTA 500, si el origen de la tabla de regiones o segmentos no está especificado, se borra una o más memorias intermedias de un elemento especificado por el índice de la región o segmento, PASO 508. De ahí en adelante, se hace una determinación sobre si hay más elementos a borrar, PREGUNTA 510. Si hay más elementos a borrar, el proceso continúa en el PASO 508. En otro caso, el proceso termina.

En un ejemplo, la invalidación y borrado se realizan con una instrucción. La instrucción invalida los elementos seleccionados de una tabla de regiones o segmentos y purga atómicamente al menos los elementos correspondientes (u otros elementos) de las TLB en los procesadores de la configuración. En un ejemplo, los elementos se purgan de todas las TLB en todos los procesadores, ya sean físicos, virtuales o emulados por software. Si existe una estructura de TLB multinivel, todos los niveles se purgan en un ejemplo. Si existen múltiples particiones lógicas en el sistema, los elementos de la TLB de este y otros procesadores que se formaron para la partición lógica en la que se ejecuta la instrucción IDTE, serán purgados.

Esta instrucción puede ser implementada en muchas arquitecturas y puede ser emulada. Como ejemplos, la instrucción puede ser ejecutada en hardware por un procesador, o por emulación del conjunto de instrucciones que incluye esta instrucción, por software que se ejecuta en un ordenador que tiene un conjunto de instrucciones diferente. En un ejemplo particular, la instrucción se implementa en la Arquitectura/z, y es denominada en esta memoria como instrucción del elemento (IDTE) de la Tabla de Traducción Dinámica de Direcciones (DAT).

La IDTE proporciona, por ejemplo, una opción de invalidación y de borrado, en la cual una o más regiones o segmentos del almacenamiento son invalidados a través de uno o más elementos de la tabla de regiones o de la tabla de segmentos, y se purgan al menos los correspondientes elementos (u otros elementos) de las TLB (por ejemplo, todas las TLB) de la configuración. La instrucción especifica un conjunto mínimo de elementos de TLB a purgar. En una implementación adicional, pueden purgarse más o incluso todos los elementos de una TLB al ejecutar la instrucción IDTE.

Un modo de realización del formato de una instrucción de invalidación de un elemento (IDTE) de la tabla DAT, se describe con referencia a la figura 6a. En un ejemplo, una instrucción IDTE 600 incluye, por ejemplo, un código de operación 602 que designa la instrucción de invalidación de un elemento de la tabla DAT (por ejemplo, ‘B98E’x), y una pluralidad de registradores 604-608. Cada uno de los registradores se describe con más detalles a continuación, con referencia a las figuras 6b-6d.

Haciendo referencia a la figura 6b, el registrador 606, que es el registrador general R1, tiene por ejemplo un formato de un elemento de control del espacio de direcciones con un origen de la tabla (por ejemplo, los bits 0-51), y el control del tipo de designación utilizado (bits 60 y 61). Los bits 52-59, 62 y 63 son ignorados. En particular, R1 incluye, por ejemplo, un origen de la tabla de regiones o de la tabla de segmentos para la invalidación 610, que designa la tabla de traducción de la cual han de invalidarse uno o más elementos; y un control del tipo de designación (DT) 612, que especifica el tipo de esa tabla (por ejemplo, la tabla de regiones o la tabla de segmentos).

El registrador 608 (figura 6c), que es el registrador general R2, incluye un primer índice 614 de región, que especifica una tabla de regiones de primer nivel; un segundo índice 616 de región, que especifica una tabla de regiones de segundo nivel; y un tercer índice 618 de región, que especifica una tabla de regiones de tercer nivel. Es decir, para acomodar direcciones virtuales grandes, se utilizan tres niveles de tablas de regiones. El registrador R2 incluye también un índice 620 de segmentos, que es un índice dentro de una tabla de segmentos; un indicador 621 de modo, que determina la función a realizar; y un indicador 622 de elementos adicionales, que indica cuántos elementos han de invalidarse. Por ejemplo, los bits 53-63 del registrador general R2 incluyen un entero binario que especifica el número de elementos adicionales de la tabla que han de invalidarse. En un ejemplo, el número de elementos a invalidar es 1-2048, correspondiente a un valor en los bits 53-63 de 0-2047. Los bits 44-51 del registrador general son ceros; en otro caso, se reconoce una excepción de la especificación. Así, los bits 0-43 del registrador general R2 tienen el formato del índice de la región y el índice del segmento de una dirección virtual. La parte de los bits 0-43 utilizada por la DAT para seleccionar un elemento en el tipo de tabla, designado por el registrador general R1, se denomina Índice de Invalidación Efectiva, y se describe con más detalle a continuación.

El registrador 604 (figura 6d), que es el registrador general R3, tiene, por ejemplo, un formato de elemento de control del espacio de direcciones, siendo utilizado el origen de la tabla, bits 0-51, si es distinto de cero. Este origen de la tabla se utiliza para seleccionar los elementos de la TLB a borrar, y el tipo de tabla que designa es independiente de los bits 60 y 61 del registrador general R1. Los bits 52-63 del registrador general R3 son ignorados. Si R3 es cero, todo el contenido del registrador general 0 es ignorado. En un ejemplo, los elementos a borrar pueden ser los mismos o diferentes de los elementos a invalidar.

Un modo de realización de la lógica asociada con una operación de invalidación y borrado, se describe con referencia a las figuras 7a-7b. En un ejemplo, se especifica una operación de invalidación y borrado cuando el bit 52 del registrador general R2 es cero. Con esta operación, se invalida el elemento designado de la tabla de regiones o de la tabla de segmentos del almacenamiento, o una gama de elementos que comienza con el elemento designado, y se borran las memorias intermedias de traducción adelantada (TLB) de las CPU de la configuración de los elementos asociados (o de otros elementos). En un ejemplo, se borran todos los niveles de las memorias intermedias, si son de niveles múltiples, de todas las CPU.

Haciendo referencia a la figura 7a, inicialmente, se especifica un origen de la tabla de regiones o de la tabla de segmentos, que designa una tabla de regiones o una tabla de segmentos de las cuales han de invalidarse uno o más elementos, PASO 700. En un ejemplo, esto se especifica en el registrador general R1. Además, se selecciona un elemento de la tabla del almacenamiento a invalidar, PASO 702. En un ejemplo, se designa un elemento de la tabla añadiendo el origen de la tabla en el registrador general R1 y el índice de invalidación efectiva en el registrador general R2, y siguiendo diversas reglas, como se describe en esta memoria.

Los bits 60 y 61 del registrador general R1 especifican un tipo de tabla y un índice de invalidación efectiva, de la manera siguiente:

| **Bits 60 y 61 del Reg. R1** | **Tipo de Tabla** | **Índice de Invalidación efectiva del Reg. R2** |
|---|---|---|
| **11** | **Tabla de la primera región** | **Primer índice de región (bits 0-10)** |
| **10** | **Tabla de la segunda región** | **Segundo índice de región (bits 11-21)** |
| **01** | **Tabla de la tercera región** | **Tercer índice de región (bits 22-32)** |
| **00** | **Tabla de segmentos** | **Índice de segmento (bits 33-43)** |

**(La parte de los bits 0-43 de la derecha del índice de invalidación efectiva se ignora).**

Con referencia a la tabla anterior, cuando los bits 60 y 61 tienen el valor binario 11, la parte del primer índice de la región de R2, junto con el origen de la tabla de la primera región de R1, se utiliza para seleccionar el elemento en la tabla de la primera región. El origen de la tabla se trata como una dirección de 64 bits. La dirección de 64 bits del elemento de la tabla de la primera región, en almacenamiento real o absoluto, se obtiene añadiendo doce ceros a la derecha de los bits 0-51 de la designación de la tabla de la primera región, y añadiendo el primer índice de la región con tres ceros añadidos a la parte más a la derecha y cincuenta ceros más a la izquierda. Esta adición se realiza de acuerdo con las reglas de la aritmética de direcciones de 64 bits, independientemente del modo de direccionamiento actual, especificado por los bits 31 y 32 de la actual Palabra de Estatus del Programa (PSW). Cuando durante la suma se lleva una unidad desde la posición cero de bits, se ignora. La dirección formada a partir de estos dos componentes es una dirección real o absoluta. Se utilizan todos los 64 bits de la dirección, independientemente de si la PSW actual especifica

el modo de dirección de 24 bits, 31 bits o 64 bits.

En un ejemplo, se extrae el elemento completo de la tabla de la primera región concurrentemente, desde el almacenamiento, según se observa por otras CPU. El acceso de extracción al elemento está sujeto a la protección controlada por clave, y el acceso al almacenamiento está sujeto a la protección controlada por clave y la protección de direcciones bajas. Cuando la dirección de almacenamiento generada por la extracción del elemento de la tabla de la primera región designa un lugar que no está disponible en la configuración, se reconoce una excepción de direccionamiento y se suprime la unidad de operación.

Cuando los bits 60 y 61 tienen el valor binario 10, el segundo índice de región de R2, junto con el origen de la tabla de la segunda región de R1, se utilizan para seleccionar un elemento de la tabla de la segunda región. El proceso de consulta de la tabla de la segunda región es similar al proceso de consulta de la tabla de la primera región. Cuando no se reconocen excepciones, el elemento extraído de la tabla de la segunda región designa el comienzo y especifica la compensación y longitud de la correspondiente tabla de la tercera región.

Cuando los bits 60 y 61 tienen el valor binario 01, la parte del tercer índice de la región de R2, junto con el origen de la tabla de la tercera región de R1, se utilizan para seleccionar un elemento de la tabla de la tercera región. El proceso de consulta de la tabla de la tercera región es similar al proceso de consulta de la tabla de la primera región. Cuando no se reconocen excepciones, el elemento extraído de la tabla de la tercera región designa el comienzo y especifica la compensación y longitud de la correspondiente tabla de segmentos.

Cuando los bits 60 y 61 tienen el valor binario 00, la parte del índice de segmento de R2, junto con el origen de la tabla de segmentos de R1, se utilizan para seleccionar un elemento de la tabla de segmentos. El proceso de consulta de la tabla de segmentos es similar al proceso de consulta de la tabla de la primera región. Cuando no se reconocen excepciones, el elemento extraído de la tabla de segmentos designa el comienzo de la correspondiente tabla de páginas.

En un ejemplo, se extrae concurrentemente todo el elemento de la tabla desde el almacenamiento. Subsiguientemente, se invalida el elemento, PASO 704. En un ejemplo, el elemento se invalida dando al bit 58 del elemento el valor 1 binario. Al invalidar este elemento, se invalida la correspondiente unidad de almacenamiento.

De ahí en adelante, se hace una determinación sobre si han de invalidarse más elementos, PREGUNTA 706. En un ejemplo, esta determinación se hace comprobando los bits 53-63 del registrador general R2. Si los bits 53-63 del registrador general R2 no son todos ceros, entonces hay más elementos a invalidar. Así, se añade 1 al valor anterior utilizado del índice de invalidación efectiva y el proceso continúa en el PASO 700. Esto se repite de manera que se invalida un número de elementos de uno más que el número especificado por los bits 53-63. Se ignora la unidad que se lleva en la posición más a la izquierda del índice de invalidación efectiva, y en este caso tiene lugar un ciclo en la tabla. El contenido del registrador general R2 permanece inalterado.

Después de la invalidación de uno o más elementos, de acuerdo con un aspecto adicional de la presente invención, esta CPU borra los elementos seleccionados de sus TLB e indica a las demás CPU de la configuración que borren los elementos seleccionados de sus TLB, PASO 708. En un ejemplo, cada TLB se borra en al menos esos elementos de la Tabla Combinada de Regiones y Segmentos (CRSTE) que reúnen las siguientes condiciones:

El índice de invalidación efectiva y los bits de su izquierda en el registrador general R2, y la misma parte del campo del índice de la región y segmento del elemento de la TLB coinciden. (En un modo de realización, si estos campos incluyen campos de índice de segmento, pueden borrarse los CRSTE independientemente de sus campos de índices de región, o bien pueden borrarse todos los CRSTE. Si los campos no incluyen campos de índice de segmento, se pueden borrar todos los CRSTE). Obsérvese que cuando se invalidan múltiples elementos de la tabla, debido a los bits 53-63 del registrador general R2, se incrementa el índice de invalidación efectiva y se pierde la unidad que se lleva desde la posición del bit más a la izquierda del índice.

O R3 es cero, o bien el campo del origen de la tabla del registrador general R3 coincide con el campo de origen de la tabla del elemento de la TLB.

Si el elemento invalidado es un elemento de la tabla de segmentos, el campo de origen de la tabla de páginas de elemento invalidado coincide con el campo de origen de la tabla de páginas del elemento de la TLB. (En un modo de realización, puede borrarse un elemento de TLB independientemente del origen de la tabla de páginas del elemento).

Cada TLB se borra también en al menos aquellos elementos de la tabla de páginas en los cuales el campo de origen de la tabla de páginas coincide con el campo del origen de la tabla de página en (1) el elemento de la tabla de segmentos invalidado, si el elemento de la tabla de segmentos fue invalidado, o (2) cualquier elemento de tabla combinada de regiones y segmentos que fue borrado (está en el conjunto que ha de borrarse) en la misma TLB. (En otro modo de realización, la implementación puede borrar un elemento de la tabla de páginas independientemente del origen de la tabla de páginas en el elemento. Esto es, puede borrar todos los elementos de la tabla de páginas).

Con referencia a la figura 7b, se describe un modo de realización de la lógica asociada con el borrado de elementos de una TLB. Como ejemplo, esto es una continuación del proceso realizado por un IDTE, y se realiza atómicamente para la

invalidación, como se observa por esta CPU y otras CPU del sistema.

Haciendo referencia a la figura 7b, se hace inicialmente una determinación sobre si R3 es cero, PREGUNTA 720. Si R3 es distinto de cero, se indica en el registrador R3 un origen de la tabla de regiones o de la tabla de segmentos para el borrado y, para borrar un elemento de la TLB, el origen de la tabla de regiones/segmentos designado en el registrador R3 debe coincidir con el origen de la tabla de regiones/segmentos utilizado para formar el elemento, PASO 722. Si R3 es cero, el contenido del registrador R3 (registrador 0) es ignorado, y se borra un elemento independientemente del origen de la tabla de regiones/segmentos utilizado para formar el elemento, PASO 724.

La TLB se borra en todos los procesadores del sistema de aquellos elementos especificados en el índice de la región/segmento especificado en el registrador R2, PASO 726. En un ejemplo, la TLB se borra en al menos aquellos elementos de la tabla combinada de regiones y segmentos en los cuales, por ejemplo, coincide el índice de validación efectiva y los bits de su izquierda en el registrador general R2, y la misma parte del elemento TLB de la tabla de índices de regiones y segmentos. Para borrar un elemento, el bit V se desactiva, como ejemplo.

Además, si se invalida un elemento de la tabla de segmentos, los TLB de todos los procesadores del sistema son borrados de los elementos de la tabla de páginas en los cuales el origen de la tabla de páginas del elemento es igual al origen de la tabla de páginas de la TLB, PASO 728. Además, si se invalida un elemento de la tabla de segmentos, se borran las TLB en al menos aquellos elementos de la tabla de páginas en las cuales el campo del origen de la tabla de páginas coincide con el campo del origen de la tabla de páginas de cualquier CRSTE que fuera borrado (es decir, necesariamente) en la misma TLB, PASO 730. Este proceso de borrado se repite para varios elementos una vez más que el número especificado en el campo 622 de elementos adicionales de la figura 6c, PASO 732.

De ahí en adelante, se hace una determinación sobre si la operación de invalidación y borrado se ha completado, PREGUNTA 734. En un ejemplo, la instrucción IDTE no se ha completado en el procesador emisor hasta que los elementos TLB designados han sido borrados en esa CPU y en todas las demás CPU del sistema que hayan terminado cualquier acceso al almacenamiento utilizando el elemento de la TLB especificado por esta instrucción. En un ejemplo, es el controlador inactivo el que indica a este procesador emisor si los demás procesadores han completado sus accesos. Es decir, el procesador emisor es considerado como procesador maestro y cualquier procesador esclavo que esté afectado por la instrucción IDTE indica su estado al controlador. Como respuesta al controlador que recibe el estado apropiado de los procesadores esclavos, indica su terminación al maestro. Si el proceso ha terminado, la instrucción termina, PASO 736. En otro caso, el procesador espera un periodo de tiempo predeterminado y comprueba el estado, una vez más, o es indicado como terminado, PREGUNTA 734.

En un modo de realización, puede ocurrir o no el almacenamiento en el elemento de la tabla de regiones o segmentos y el borrado de los elementos TLB, si el bit inválido era ya uno del elemento de la tabla de regiones o segmentos.

Aunque en el modo de realización descrito anteriormente, el borrado de los elementos de la TL se realiza posteriormente a la invalidación de los elementos deseados, en otros modos de realización, el borrado de elementos de la TLB puede tener lugar tras cada invalidación, tras un número seleccionado de invalidaciones, etc.

En un aspecto adicional de la presente invención, la instrucción IDTE puede ser utilizada alternativamente para purgar elementos de la TLB asociados con un espacio de direcciones (por ejemplo, designados por un Elemento de Control del Espacio de Direcciones (ASCE)). En un ejemplo, esta opción se indica poniendo en 1 el bit 52 del registrador general R2. Con esta opción, el contenido del registrador general R1 es ignorado (véase la figura 8a), así como los bits 0-43 y 53-63 del registrador general R2 (véase la figura 8b). El contenido del registrador general R3 tiene el formato de un elemento de control del espacio de direcciones utilizando el origen de la tabla, bits 0-51 (figura 8c). Los bits 52-63 del registrador general R3 son ignorados. R3 puede ser cero o distinto de cero; es decir, puede designarse cualquier registrador general que incluya el registrador 0. Esta operación no realiza ninguna invalidación del almacenamiento, sino que borra de las TLB de las CPU de esta configuración al menos aquellos elementos de la tabla combinada de regiones y segmentos y los elementos de la tabla de páginas en las cuales el campo de origen de la tabla coincide con el campo del origen de la tabla del registrador R3. (En un modo de realización adicional, la implementación puede borrar un elemento de la tabla de página, independientemente del origen de la tabla en el elemento. Es decir, puede borrar todos los elementos de la tabla de páginas).

Con referencia a la figura 9, se describe un modo de realización de la lógica asociada con el borrado de una operación ASCE. Inicialmente, se especifica un origen de la tabla de regiones o de la tabla de segmentos que designa una tabla de regiones o una tabla de segmentos para las cuales se especifica que han de borrarse los correspondientes elementos TLB, PASO 900. En un ejemplo, esto se especifica en los bits 0-51 del registrador general R3. De ahí en adelante, se borran al menos los elementos de la tabla combinada de regiones y segmentos y los elementos de la tabla de páginas correspondientes al origen de la tabla especificado en R3, en todos los procesadores del sistema, PASO 902.

Subsiguientemente, se hace una determinación sobre si la ejecución de la instrucción IDTE ha terminado, PREGUNTA 904. En un modo de realización, la instrucción IDTE no se completa en el procesador emisor hasta que los elementos de la TLB correspondientes a los parámetros especificados han sido borrados en la CPU emisora y en las demás CPU de la configuración, han completado cualquier acceso al almacenamiento, incluyendo la actualización de los bits de

cambio y de referencia, utilizando los elementos de la TLB correspondientes a los parámetros especificados. Si se han borrado los elementos y los accesos al almacenamiento se han completado, la ejecución de la instrucción IDTE está completa, PASO 906. En otro caso, el procesador espera un periodo de tiempo predeterminado y comprueba el estatus una vez más o bien recibe una indicación de terminación, PREGUNTA 904.

El proceso asociado con una instrucción IDTE se ha descrito con detalle anteriormente. En un modo de realización, las operaciones no tienen necesariamente ningún efecto sobre los elementos del espacio real de la TLB. Lo que sigue proporciona información adicional, mejoras y/o modos de realización asociados con el proceso de un modo de realización de una instrucción IDTE.

1. El borrado selectivo de elementos de TLB puede ser implementado de distintas maneras, dependiendo del modelo o modo de realización seleccionado. Por ejemplo, en general, pueden borrarse más elementos que el número mínimo especificado. Cuando se realiza la operación de invalidación y borrado, pueden borrarse todos los elementos de la tabla combinada de regiones y segmentos (CRSTE) que contienen un índice de segmento igual a un índice de invalidación efectiva que es un índice de segmento, o bien pueden borrarse todos los CRSTE cuando el índice de invalidación efectiva no es un índice de segmento. Además, puede borrarse un CRSTE o un elemento de la tabla de páginas, independientemente del origen de la tabla en el elemento. Cuando se realiza la operación de borrado por ASCE, puede borrarse un elemento de la tabla de páginas independientemente del origen de la tabla en el elemento. Cuando se realiza cualquiera de esas dos operaciones, en un modo de realización, puede borrarse el número mínimo preciso de elementos requeridos.

2. Un elemento del espacio de direcciones puede contener un origen del testigo del espacio real, en lugar de un origen de la tabla de regiones o de la tabla de segmentos. Un origen de la tabla en un registrador general R3 es indistinguible de un origen de testigo en el registrador.

3. Cuando se especifica una operación de invalidación y borrado, el borrado de elementos de TLB puede hacer uso del origen de la tabla de páginas en un elemento de la tabla de segmentos. Por tanto, si el elemento de la tabla de segmentos, cuando está en estado anexado, contuviera un origen de la tabla de páginas que fuera diferente del valor actual, podrían permanecer en la TLB copias de elementos que contuvieran valores previos.

4. Cuando se especifica la operación de invalidación y borrado, la dirección del elemento de la tabla DAT para la IDTE es una dirección de 64 bits, y se realiza la aritmética de direcciones siguiendo las reglas normales de la aritmética de direcciones de 64 bits, efectuando un ciclo al llegar a $2^{64} – 1$. Además, no se utilizan los campos de compensación y de longitud. Consecuentemente, las tablas DAT no han de especificarse para que efectúen un ciclo desde los lugares de máximo almacenamiento hasta el lugar 0, y el primer elemento designado y todos los elementos adicionales especificados por los bits 63-63 del registrador general R3 deben caer dentro de la tabla designada.

5. La instrucción IDTE está sometida a interceptación bajo la SIE. En un modo de realización, el sistema es puesto en serie de forma que el procesador que ejecuta la instrucción IDTE no puede comenzar la operación hasta que otros procesadores han dejado de utilizar los elementos de la TLB a especificar. Además, el procesador que ejecuta la instrucción IDTE no continúa con una ejecución adicional hasta que hayan sido borrados los elementos apropiados de las TLB de todos los procesadores del sistema.

6. Se realiza una operación de puesta en serie antes de que comience la operación y nuevamente después de haberse completado la operación. La puesta en serie es aplicable a esta CPU; otras CPU no son puestas en serie necesariamente.

Anteriormente se ha descrito en detalle un ejemplo de una instrucción que puede invalidar elementos de una tabla de segmentos o de regiones y borrar atómicamente al menos los elementos correspondientes (u otros elementos) de una o más memorias intermedias de traducción adelantada. Aunque una instrucción particular o concreta es descrita aquí, pueden realizarse muchas variaciones sin alejarse del espíritu de la presente invención. Por ejemplo, pueden utilizarse diferentes códigos de operaciones, diferentes campos, diferentes registradores o incluso ningún registrador, diferentes bits, etc. Por ejemplo, otras arquitecturas de conjuntos de instrucciones pueden definir una instrucción equivalente en formas que son diferentes de alguna manera (por ejemplo, diferentes códigos de operaciones, diferentes campos en la instrucción, diferentes registradores utilizados, etc.) pero se siguen aplicando uno o más aspectos de la presente invención. Además, como se ha descrito anteriormente, se aplican igualmente bien uno o más aspectos de la presente invención a una emulación por software de una arquitectura objetivo, tal como la arquitectura/z de IBM. En este caso, las TLB estudiadas anteriormente pueden ser una serie de otras estructuras de datos implementadas por el emulador por software para mantener las correspondencias recientes de direcciones virtuales con las reales. También puede ser que el emulador por software no implemente ninguna estructura análoga a una TLB. En este caso, solamente es aplicable la invalidación de elementos en el almacenamiento.

Por tanto, se proporciona en esta memoria una instrucción que es capaz de invalidar el almacenamiento, invalidar el almacenamiento y borrar atómicamente elementos de la memoria intermedia, o borrar elementos de la memoria intermedia. Por tanto, se dice que uno o más registradores de la instrucción son configurables como no-operaciones (no-ops). Es decir, la información en esos registradores, si la hay, asociada con una operación particular (por ejemplo, la

invalidación, el borrado, etc.) que no sea proporcionada, es ignorada o indica que la operación no se ha realizado.

Aunque se han descrito anteriormente una instrucción particular en la cual se proporciona la invalidación, la invalidación y el borrado, o el borrado, la instrucción puede ser modificada o pueden implementarse otras instrucciones que realicen justamente uno o más aspectos de las operaciones implementadas por IDTE. Por ejemplo, una instrucción puede implementar simplemente la invalidación del almacenamiento. Como ejemplo adicional, la instrucción puede simplemente purgar elementos de las memorias intermedias que están asociadas con tablas particulares de segmentos o regiones. Como ejemplo adicional más, puede implementarse una instrucción que purgue simplemente los elementos basados en el espacio de direcciones.

En una implementación comercial del formato de la instrucción con arquitectura informática con posibilidad de invalidación y/o borrado, las instrucciones son utilizadas por programadores, tales como los programadores del sistema operativo que escriben en lenguaje ensamblador. Estos formatos de instrucciones almacenados en un medio de almacenamiento pueden ser ejecutados en modo nativo en un Servidor IBM de arquitectura/z, o alternativamente, en máquinas que ejecuten otras arquitecturas. Pueden ser emulados en servidores de ordenadores centrales de IBM existentes y futuros, y en otras máquinas de IBM (por ejemplo, en servidores pSeries y servidores xSeries). Pueden ejecutarse en máquinas que ejecuten Linux en una amplia diversidad de máquinas que utilizan hardware fabricado por IBM, Intel, AMD, Sun Microsystems y otros. Además de la ejecución en hardware bajo la arquitectura/z, puede utilizarse Linux, así como máquinas que utilizan la emulación por Hércules, UMX, FXI o Soluciones de Plataformas, donde generalmente la ejecución es en un modo de emulación.

En el modo de emulación, se descodifica la instrucción específica que se está emulando, y se construye una subrutina para implementar la instrucción individual, como en una subrutina o controlador de C, o se utiliza alguna otra técnica para proporcionar un controlador para el hardware específico, como se comprende dentro del conocimiento de los expertos en la técnica, tras entender la descripción de un modo de realización de la invención. Hay diversas patentes de emulación por software y hardware, incluyendo, aunque sin limitarse a ellas, las patentes de Estados Unidos núms. US 5.551.013, US 6.009.261, US 5.574.873, US 6.308.255, US 6.463.582 y US 5.790.825, ilustran una diversidad de maneras para conseguir la emulación de un formato de instrucción con una arquitectura hecha para una máquina diferente de una máquina objetivo disponible para los expertos en la técnica, así como aquellas técnicas de software comercial utilizadas por las arriba referenciadas.

Ventajosamente, uno o más aspectos de la presente invención aumentan el rendimiento del sistema. Por ejemplo, purgando selectivamente los elementos de las memorias intermedias, en lugar de purgar completamente las memorias intermedias, el rendimiento del sistema se mejora al no requerir volver a crear innecesariamente los elementos. Es decir, se requiere desde 20+ hasta 100s de ciclos para crear un elemento de una TLB, y la purga selectiva evita volver a crear innecesariamente los elementos.

Como ventaja adicional, si se desea una purga de TLB completa, la implementación permite la sobre-purga. Además, la parte de TLB puede ser tratada como una no-operación si, por ejemplo, no se utiliza una TLB. Ventajosamente, una instrucción IDTE puede ser implementada en hardware, firmware, software o una combinación de los mismos, o puede ser utilizada en un emulador de un conjunto de instrucciones de un ordenador objetivo.

Son posibles muchas variaciones en el entorno anterior sin apartarse del espíritu de la presente invención. Por ejemplo, uno o más aspectos de la presente invención son igualmente aplicables, por ejemplo, a la emulación de máquinas virtuales, en la cual se ejecutan una o más entidades paginables (es decir, invitadas) en uno o más procesadores. Como ejemplo, los invitados paginables son definidos por la arquitectura de Ejecución Interpretativa de Inicio (SIE), un ejemplo de lo cual está descrito en una publicación de IBM titulada “IBM System/370 Extended Arquitecture” (Arquitectura extendida del Sistema 370 de IBM), publicación de IBM núm. SA22-7095 (1985).

Aunque se han mencionado anteriormente la SIE y la arquitectura/z, uno o más aspectos de la presente invención son igualmente aplicables a otras arquitecturas y/o entornos que emplean entidades paginables o construcciones similares.

Más aún, los diversos modos de realización o realizaciones descritas anteriormente son meros ejemplos. Puede haber muchas variaciones a esas realizaciones sin apartarse del espíritu de la presente invención. Por ejemplo, aunque se describe en esta memoria un entorno lógicamente particionado, esto es sólo un ejemplo. Aspectos de la invención son beneficiosos para muchos tipos de entornos, incluyendo otros entornos que tengan una pluralidad de zonas, y entornos no particionados. Además, puede no haber complejos de ordenadores centrales, pero aún así, puede haber múltiples procesadores acoplados conjuntamente. Más aún, uno o más aspectos de la invención son aplicables a entornos de un solo procesador.

Aunque en esta memoria se han descrito entornos particulares, de nuevo, pueden implementarse muchas variaciones de estos entornos sin apartarse del espíritu de la presente invención. Por ejemplo, si el entorno está lógicamente particionado, pueden incluirse más o menos particiones lógicas en el entorno. Además, puede haber múltiples complejos de proceso central acoplados conjuntamente. Estas son solamente algunas de las variaciones que pueden hacerse. Adicionalmente, son posibles otras variaciones. Por ejemplo, aunque el controlador descrito en esta memoria pone en serie la instrucción, de manera que se ejecuta una instrucción IDTE de una vez, en otro modo de realización, se pueden ejecutar múltiples instrucciones de una vez. Además, el entorno puede incluir múltiples controladores. Más aún,

en el sistema puede haber en vigor múltiples peticiones de inactividad (desde uno o más controladores). Son posibles también variaciones adicionales.

Según se ha utilizado en esta memoria, el término “unidad de proceso” incluye entidades paginables, tales como invitados; procesadores; emuladores y/u otros componentes similares. Más aún, el término “por una unidad de proceso” incluye también “en nombre de una unidad de proceso”. El término “memoria intermedia” incluye una zona de almacenamiento, así como diferentes tipos de estructuras de datos, incluyendo, aunque no limitándose a ellas, las series ordenadas; y el término “tabla” puede incluir otras además de las estructuras de datos de tipo de tablas. Además, la instrucción puede incluir otros dispositivos además de los registradores para designar la información. Más aún, una página, un segmento y/o una región pueden ser de tamaños diferentes a los descritos en esta memoria.

Una o más de las capacidades de la presente invención pueden ser implementadas en software, firmware, hardware, o alguna combinación de los mismos. Además, pueden emularse una o más de las capacidades.

Uno o más aspectos de la presente invención pueden ser incluidos en un artículo de fabricación (por ejemplo, en uno o más productos de programas informáticos) que tengan, por ejemplo, medios utilizables por ordenador. Los medios han incorporado en ellos, por ejemplo, medios de código de programas legibles por ordenador o lógica (por ejemplo, instrucciones, código, órdenes, etc.) para proporcionar y facilitar las capacidades de la presente invención. El artículo de fabricación puede ser incluido como parte de un sistema informático o vendido separadamente.

Adicionalmente, puede proporcionarse al menos un dispositivo de almacenamiento de programas legible por una máquina que materializa al menos un programa de instrucciones ejecutable por la máquina para realizar las capacidades de la presente invención.

Los diagramas de flujo representados en esta memoria son meros ejemplos. Puede haber muchas variaciones a estos diagramas o a los pasos (u operaciones) descritos en ellos. Por ejemplo, los pasos pueden ser realizados en un orden diferente, o los pasos pueden ser añadidos, eliminados o modificados.

**REIVINDICACIONES**

1. Un método para invalidar una gama de dos o más elementos de una tabla de traducción de direcciones en un sistema informático que tiene tablas de traducción de direcciones para traducir dinámicamente direcciones virtuales a direcciones de almacenamiento principal, el método comprende los pasos de:

determinar una instrucción ejecutable para máquina desde un código de operación para ser ejecutada, que la instrucción esté configurada para iniciar la ejecución de una operación de invalidación y borrado; y

ejecutar la instrucción, comprendiendo el paso de ejecución los pasos de:

determinar, desde la información proporcionada por la instrucción, una primera dirección de elemento de tabla de traducción de un primer elemento de una tabla de traducción, de la gama de dos o más elementos de una tabla de traducción de direcciones para ser invalidadas;

determinar, desde la información de gama proporcionada por la instrucción, un número de elementos de una tabla de traducción de direcciones para ser invalidadas; y

basándose en la primera dirección del elemento de tabla de traducción determinada, invalidar la gama de dos o más elementos de una tabla de traducción de direcciones indicadas por la información de la gama.

2. El método de acuerdo con la reivindicación 1, en el que la gama de dos o más elementos de una tabla de traducción de direcciones para ser invalidadas comprende cada una un bit inválido para la invalidación, en el que cuando el bit inválido es 0, se permite el uso o utilización del elemento de la tabla de traducción de direcciones para la traducción dinámica, en el que cuando el bit inválido es1, no se permite el uso o utilización del elemento de la tabla de traducción de direcciones para la traducción dinámica.

3. El método de acuerdo con la reivindicación 1, en el que la gama de dos o más elementos de la tabla de traducción consta de uno cualquiera de entre los elementos de la tabla de segmentos o de los elementos de la tabla de regiones.

4. El método de acuerdo con la reivindicación 1, en el que el sistema informático comprende procesadores centrales (CPU), registradores con fines generales de arquitectura, memorias intermedias de traducción de direcciones, las memorias intermedias de traducción de direcciones teniendo elementos de memorias intermedias de traducción de direcciones que mantienen información de traducción de direcciones, las tablas de traducción de dirección constando de una cualquiera de entre:

una o más tablas de segmentos que contienen elementos que apuntan a las tablas de páginas.,

una o más primeras tablas de regiones que contienen elementos que apuntan a segundas tablas de regiones, conteniendo las segundas tablas de regiones elementos que apuntan a terceras tablas de regiones, conteniendo las terceras tablas de regiones elementos que apuntan a las tablas de segmentos, conteniendo las tablas de segmentos elementos que apuntan a las tablas de páginas,

una o más segundas tablas de regiones que contienen elementos que apuntan a terceras tablas de regiones, conteniendo las terceras tablas de regiones elementos que apuntan a las tablas de segmentos, conteniendo las tablas de segmentos elementos que apuntan a las tablas de páginas,

una o más terceras tablas de regiones que contienen elementos que apuntan a tablas de segmentos, conteniendo las tablas de segmentos elementos que apuntan a las tablas de páginas; y la determinación desde el paso del código de operación que comprende los pasos adicionales de:

recuperar la segunda información desde una posición especificada por la instrucción, comprendiendo la segunda información una indicación de la función de invalidar y borrar.

5. El método de acuerdo con la reivindicación 4, en el que la segunda información indica la función de invalidar y borrar cuando el bit 52 de la segunda información es 0.

6. El método de acuerdo con la reivindicación 1, en el que el sistema informático comprende una o más memorias intermedias de traducción de direcciones para poner en la memoria caché la información de traducción de direcciones, comprendiendo el método el paso adicional de:

borrar las memorias intermedias de traducción de direcciones de los elementos de memorias intermedias de traducción de direcciones asociadas con los elementos de la tabla de traducción de direcciones de la gama de dos o más elementos de la tabla de traducción de direcciones.

7. El método de acuerdo con la reivindicación 6, en el que las memorias intermedias de traducción de direcciones constan de una o más Memorias Intermedias de Traducción Adelantada (TLB).

8. El método de acuerdo con la reivindicación 6, en el que las memorias intermedias de traducción de direcciones comprenden una cualquiera de entre:

una memoria intermedia de traducción de direcciones asociada con el procesador central,

una memoria intermedia de traducción de direcciones asociada con otro procesador central del sistema informático, o

todas las memorias intermedias de traducción de direcciones asociadas con el sistema informático.

9. El método de acuerdo con la reivindicación 1, en el que el paso de la determinación desde un código de operación comprende los pasos adicionales de:

extraer la instrucción desde el almacenamiento principal, comprendiendo la instrucción:

a) un primer campo para ubicar tercera información, comprendiendo la tercera información una localización del origen de la tabla de traducción de direcciones para invalidar; y

b) un segundo campo para ubicar cuarta información, comprendiendo la cuarta información una porción del índice de la tabla de traducción de direcciones; y para formar la primera dirección del elemento de la tabla de traducción a partir de la tercera información en conjunción con la cuarta información.

10. El método de acuerdo con la reivindicación 9, donde en el primer campo se coloca o ubica la tercera información en un primer registrador con fines generales de arquitectura, en el que la ubicación del origen de la tabla de traducción de direcciones consta de una cualquiera de entre una dirección de una tabla de región o una dirección de una tabla de segmentos.

11. El método de acuerdo con la reivindicación 9, en el que la tercera información comprende además un campo de control del tipo de designación para designar una cualquiera de entre una región para la invalidación o una tabla de segmentos para la invalidación.

12. El método de acuerdo con la reivindicación 9, donde en el segundo campo se ubica la cuarta información, comprendiendo la cuarta información un primer campo del índice de región, un segundo campo del índice de región, un tercer campo del índice de región, un campo del índice de segmento y la gama.

13. El método de acuerdo con la reivindicación 1, en el que la instrucción consta de 32 bits contiguos, que incluye un campo de código de operación que consta de 16 bits contiguos, un primer campo de operando que consta de 4 bits contiguos, un segundo campo de operando que consta de 4 bits contiguos, y un tercer campo de operando que consta de 4 bits contiguos, en el que los bits contiguos del primer campo son contiguos a los bits contiguos del segundo campo y en el que los campos de operandos especifican los registradores con fines generales respectivos de un espacio del registrador de arquitectura informática.

14. El método de acuerdo con la reivindicación1, en el que el paso de la invalidación de la gama de dos o más elementos de la tabla de traducción de direcciones comprende el paso adicional de marcar inválidos, un bit inválido de cada uno de los elementos de la tabla de traducción de direcciones de la gama de dos o más elementos de la tabla de traducción de direcciones.

15. El método de acuerdo con la reivindicación 1, en el que el sistema informático comprende además tablas de páginas, teniendo las tablas de páginas direcciones de almacenamiento principal reales, en el que las direcciones de almacenamiento principal reales comprenden una cualquiera de entre una dirección absoluta, una dirección real o una dirección física.

16. Un programa de ordenador o programa informático que comprende instrucciones para llevar a cabo todos los pasos del método de acuerdo con cualquier reivindicación precedente del método, cuando dicho programa de ordenador se ejecute en un sistema informático.

17. Un sistema que comprende medios adaptados para llevar a cabo todos los pasos del método de acuerdo con cualquier reivindicación precedente del método.

18. El sistema de la reivindicación 17, que comprende:

un almacenamiento principal;

uno o más procesadores centrales en comunicación con el almacenamiento principal;

registradores con fines generales de arquitectura accesibles para el/los procesador/es central/es;

tablas de traducción de direcciones para traducir dinámicamente direcciones virtuales para las direcciones del almacenamiento principal, las tablas de traducción de direcciones siendo accesibles para el/los procesador/es central/es; y

una o más unidades de ejecución de instrucciones accesibles para el/los procesador/es central/es, las unidades de ejecución de instrucciones estando en comunicación con los registradores con fines generales de arquitectura, las unidades de ejecución de instrucciones ejecutando instrucciones para máquina extraídas del almacenamiento principal.

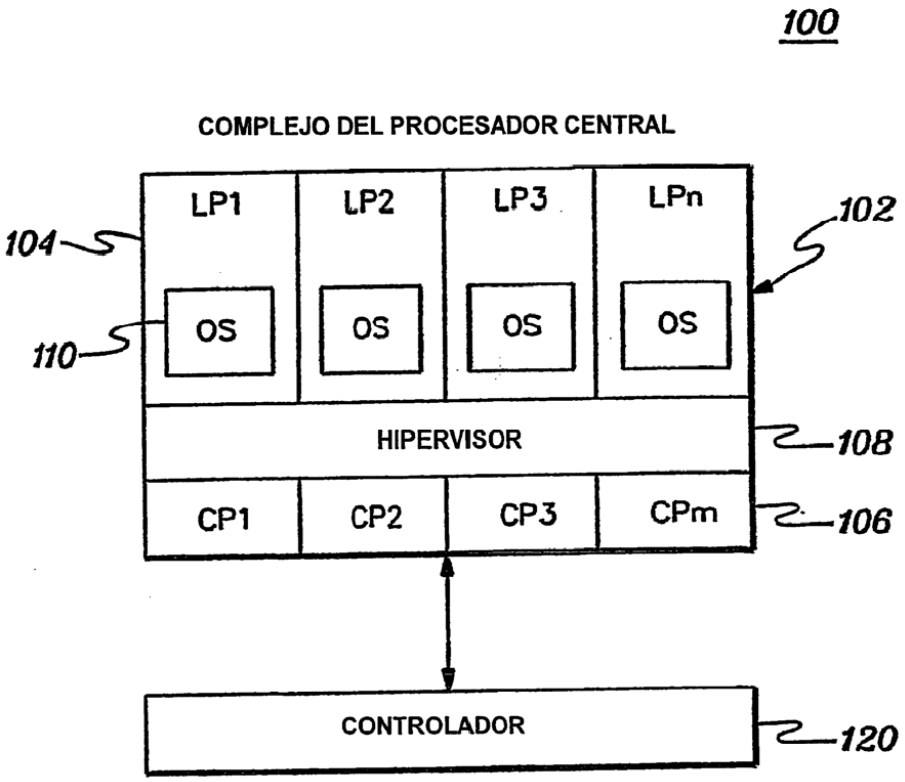
100
COMPLEJO DEL PROCESADOR CENTRAL
LP1
LP2
LP3
LPn
104
102
110
OS
OS
OS
OS
HIPERVISOR
108
CP1
CP2
CP3
CPm
106
CONTROLADOR
120

*fig. 1*

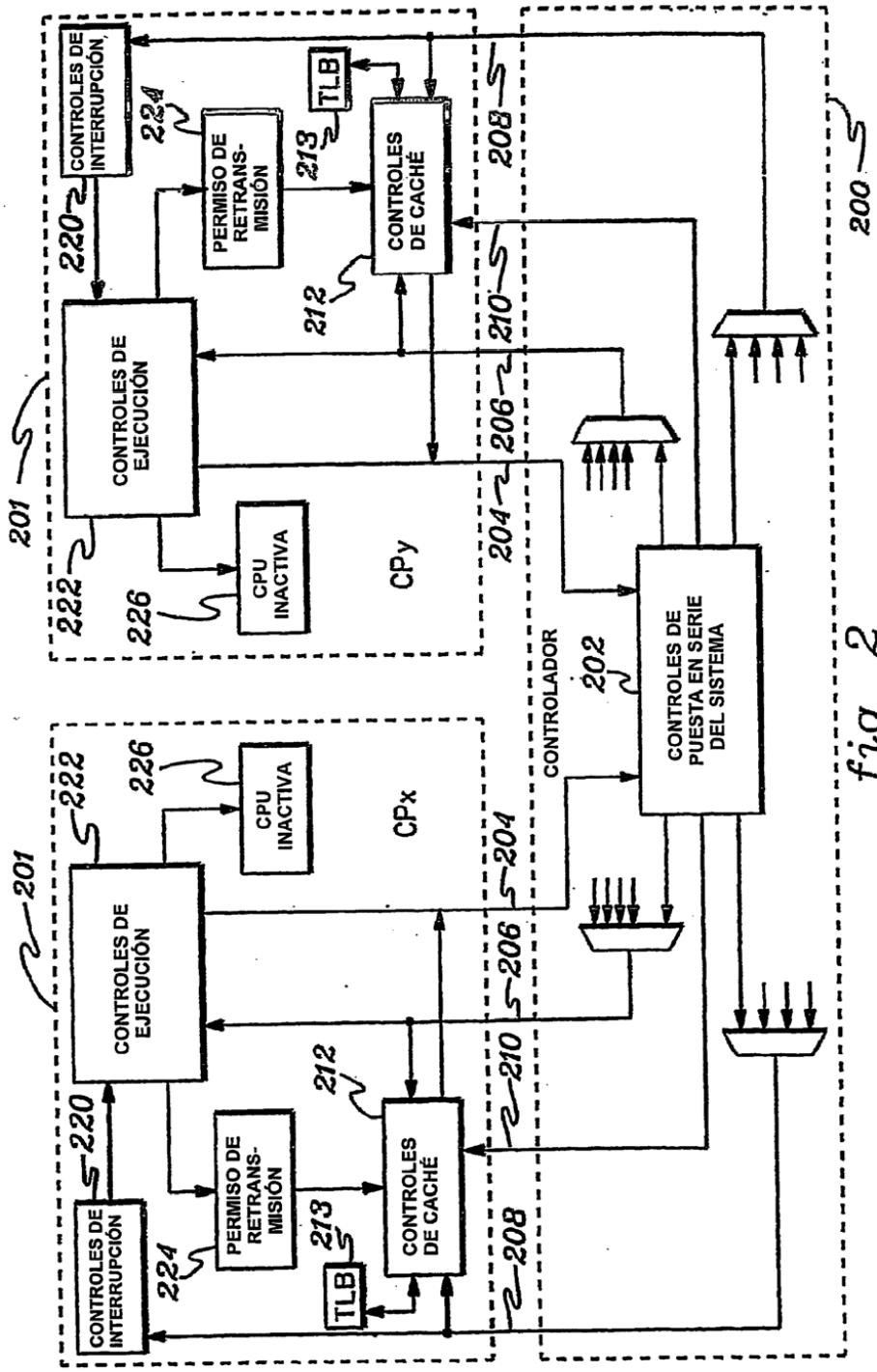
CONTROLES DE INTERRUPCIÓN
PERMISO DE RETRANS-MISIÓN
TLB
CONTROLES DE CACHÉ
CONTROLES DE EJECUCIÓN
CPU INACTIVA
CPy
CPx
CONTROLADOR
CONTROLES DE PUESTA EN SERIE DEL SISTEMA
200
201
202
204
206
208
210
212
213
220
222
224
226

fig. 2

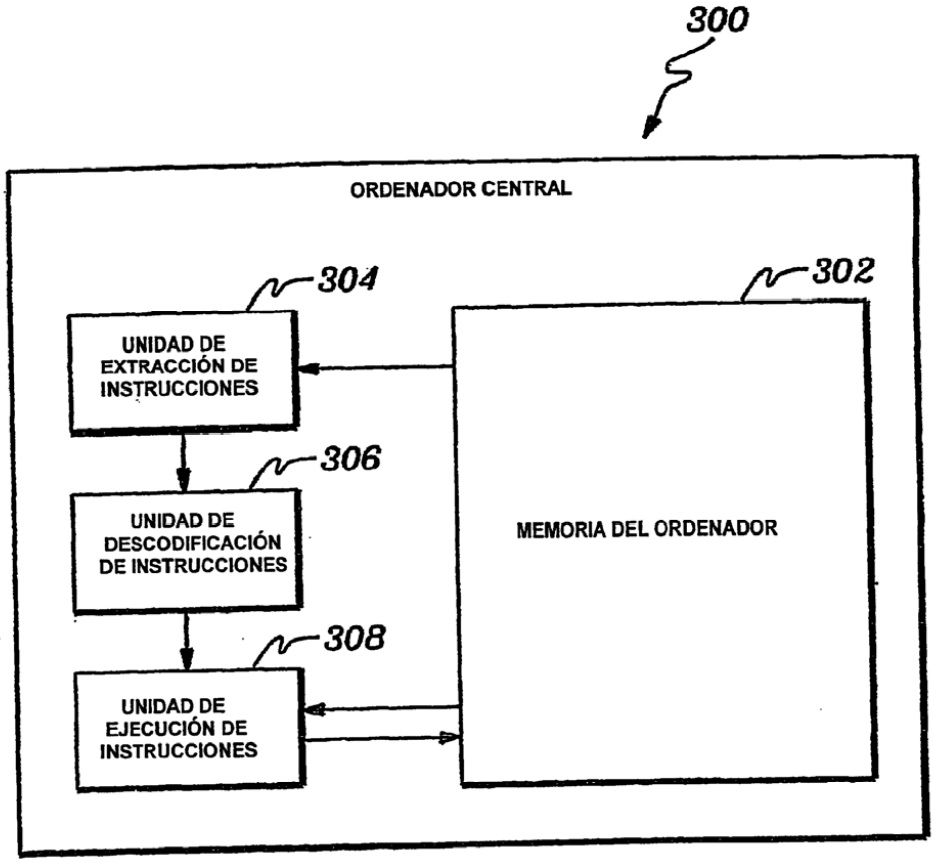
300
ORDENADOR CENTRAL
304
UNIDAD DE EXTRACCIÓN DE INSTRUCCIONES
306
UNIDAD DE DESCODIFICACIÓN DE INSTRUCCIONES
308
UNIDAD DE EJECUCIÓN DE INSTRUCCIONES
302
MEMORIA DEL ORDENADOR

*fig. 3*

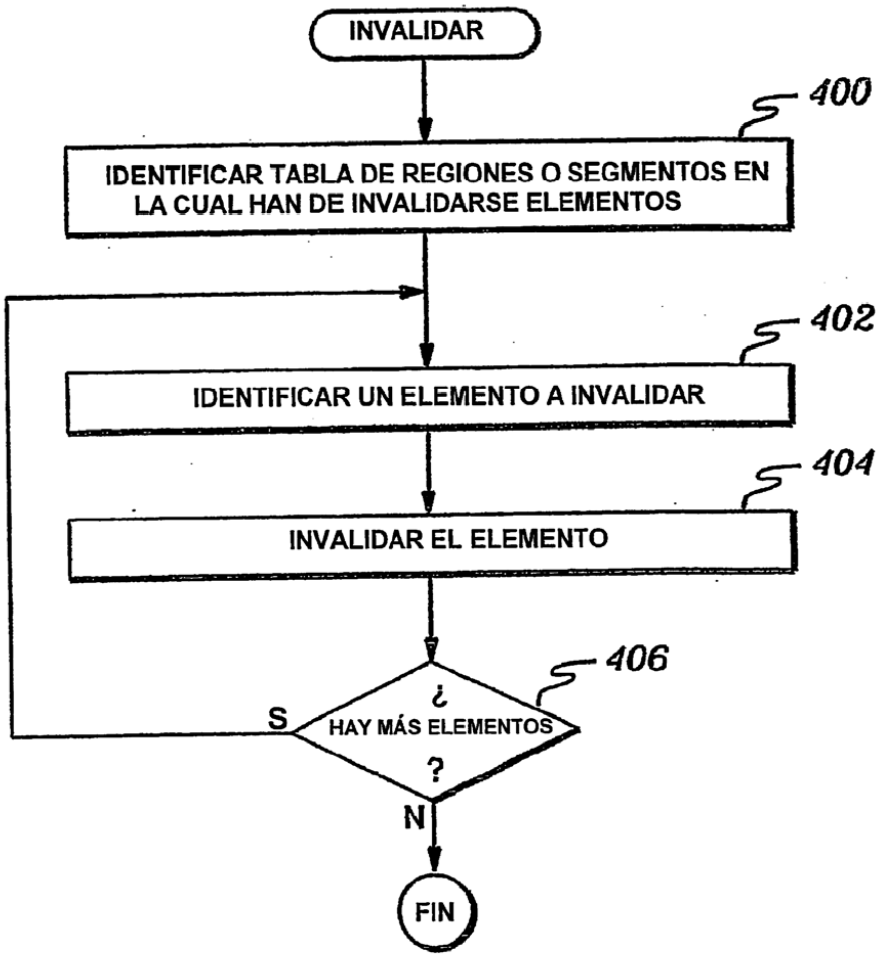
INVALIDAR
400
IDENTIFICAR TABLA DE REGIONES O SEGMENTOS EN LA CUAL HAN DE INVALIDARSE ELEMENTOS
402
IDENTIFICAR UN ELEMENTO A INVALIDAR
404
INVALIDAR EL ELEMENTO
406
¿ HAY MÁS ELEMENTOS ?
S
N
FIN

*fig. 4*

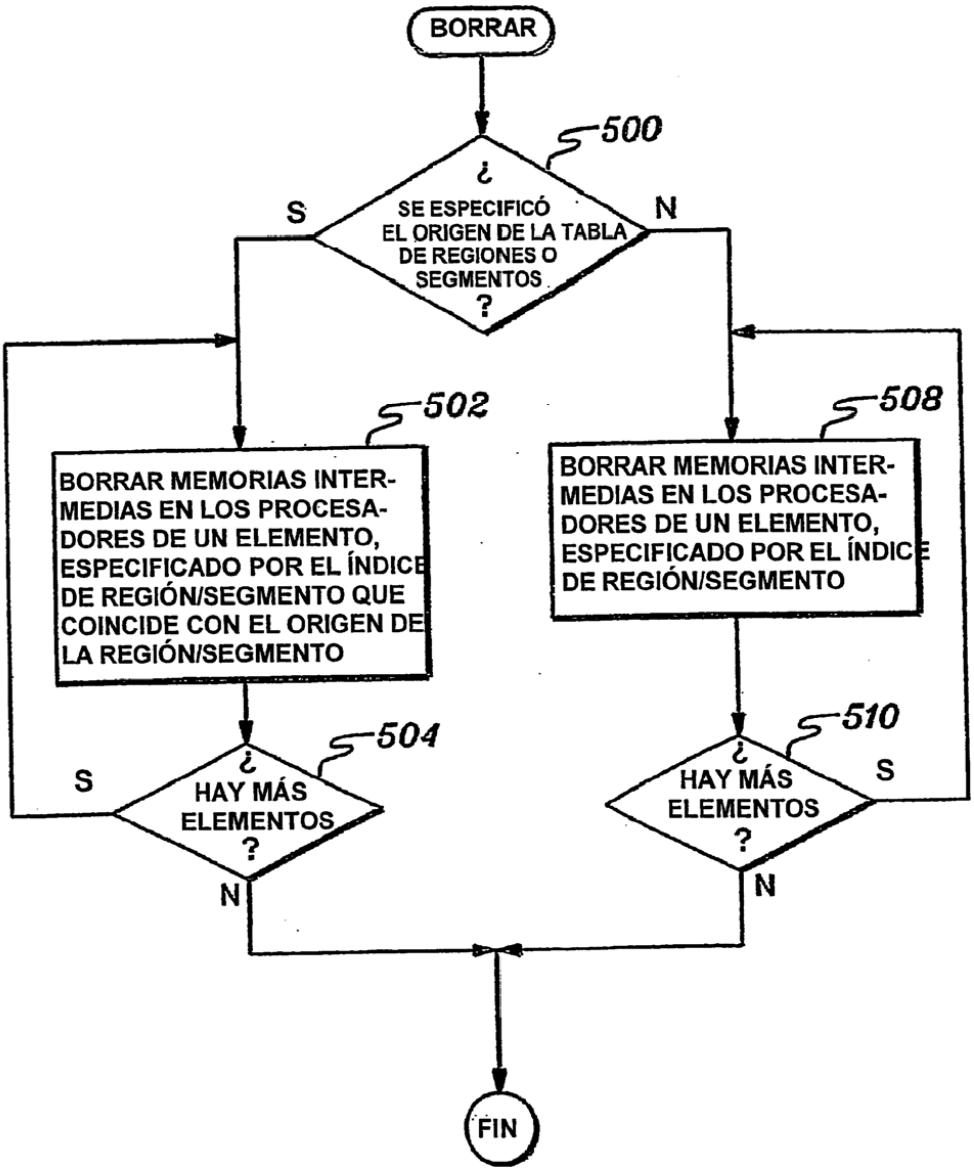
BORRAR
500
¿ SE ESPECIFICÓ EL ORIGEN DE LA TABLA DE REGIONES O SEGMENTOS ?
S
N
502
BORRAR MEMORIAS INTERMEDIAS EN LOS PROCESADORES DE UN ELEMENTO, ESPECIFICADO POR EL ÍNDICE DE REGIÓN/SEGMENTO QUE COINCIDE CON EL ORIGEN DE LA REGIÓN/SEGMENTO
504
¿ HAY MÁS ELEMENTOS ?
S
N
508
BORRAR MEMORIAS INTERMEDIAS EN LOS PROCESADORES DE UN ELEMENTO, ESPECIFICADO POR EL ÍNDICE DE REGIÓN/SEGMENTO
510
¿ HAY MÁS ELEMENTOS ?
S
N
FIN

*fig. 5*

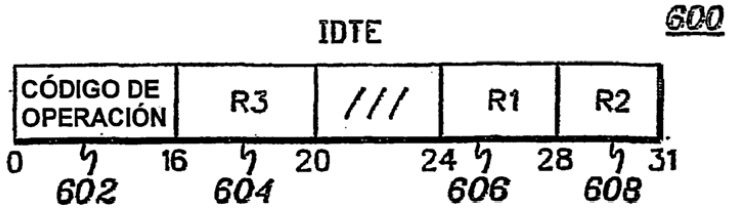


fig. 6A

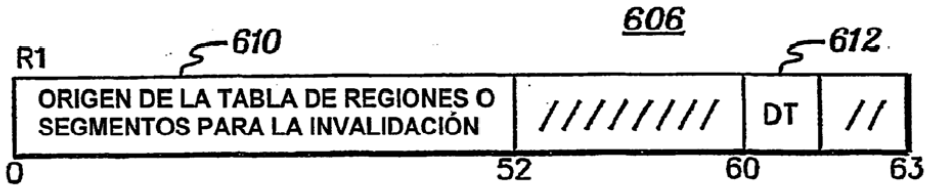


fig. 6B

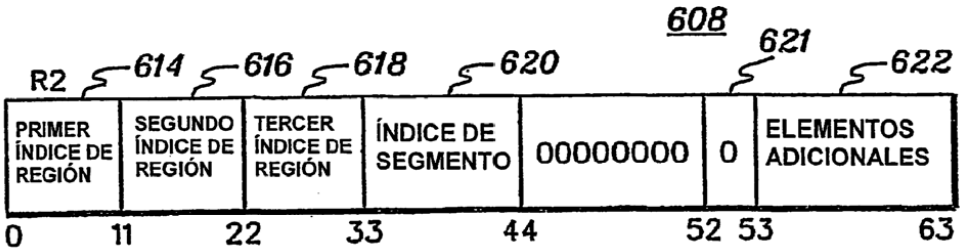


fig. 6C

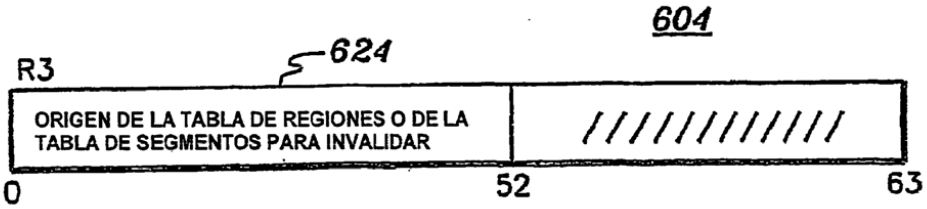


fig. 6D

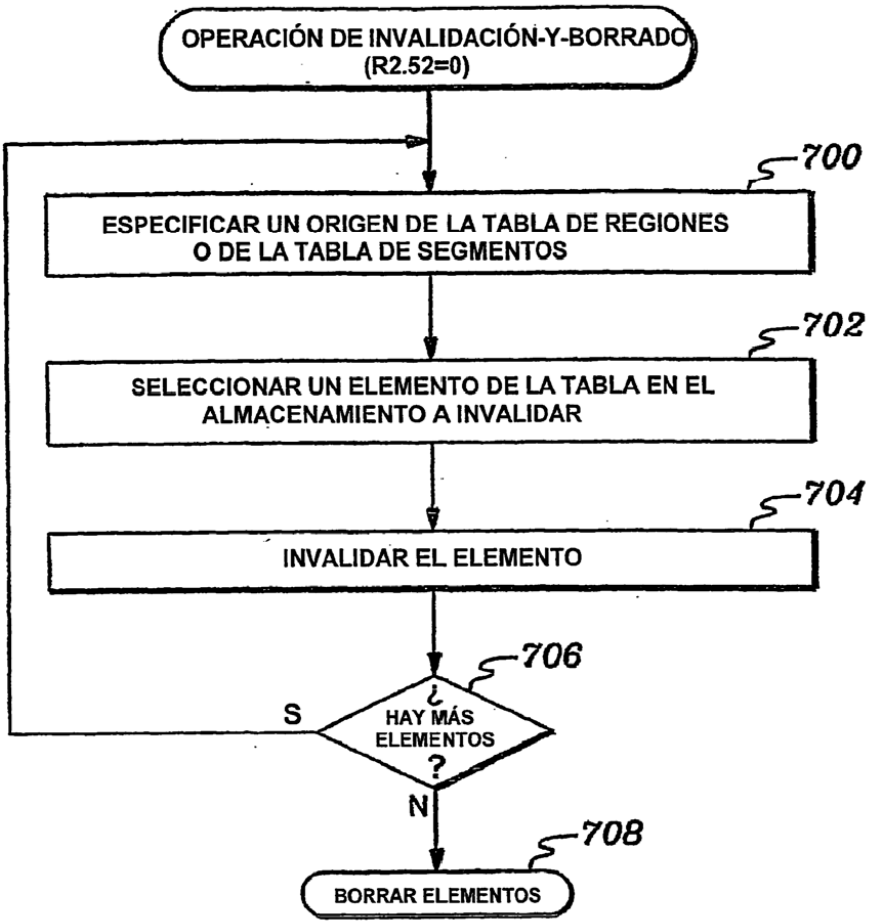
OPERACIÓN DE INVALIDACIÓN-Y-BORRADO
(R2.52=0)
700
ESPECIFICAR UN ORIGEN DE LA TABLA DE REGIONES O DE LA TABLA DE SEGMENTOS
702
SELECCIONAR UN ELEMENTO DE LA TABLA EN EL ALMACENAMIENTO A INVALIDAR
704
INVALIDAR EL ELEMENTO
706
¿HAY MÁS ELEMENTOS?
S
N
708
BORRAR ELEMENTOS

*fig. 7A*

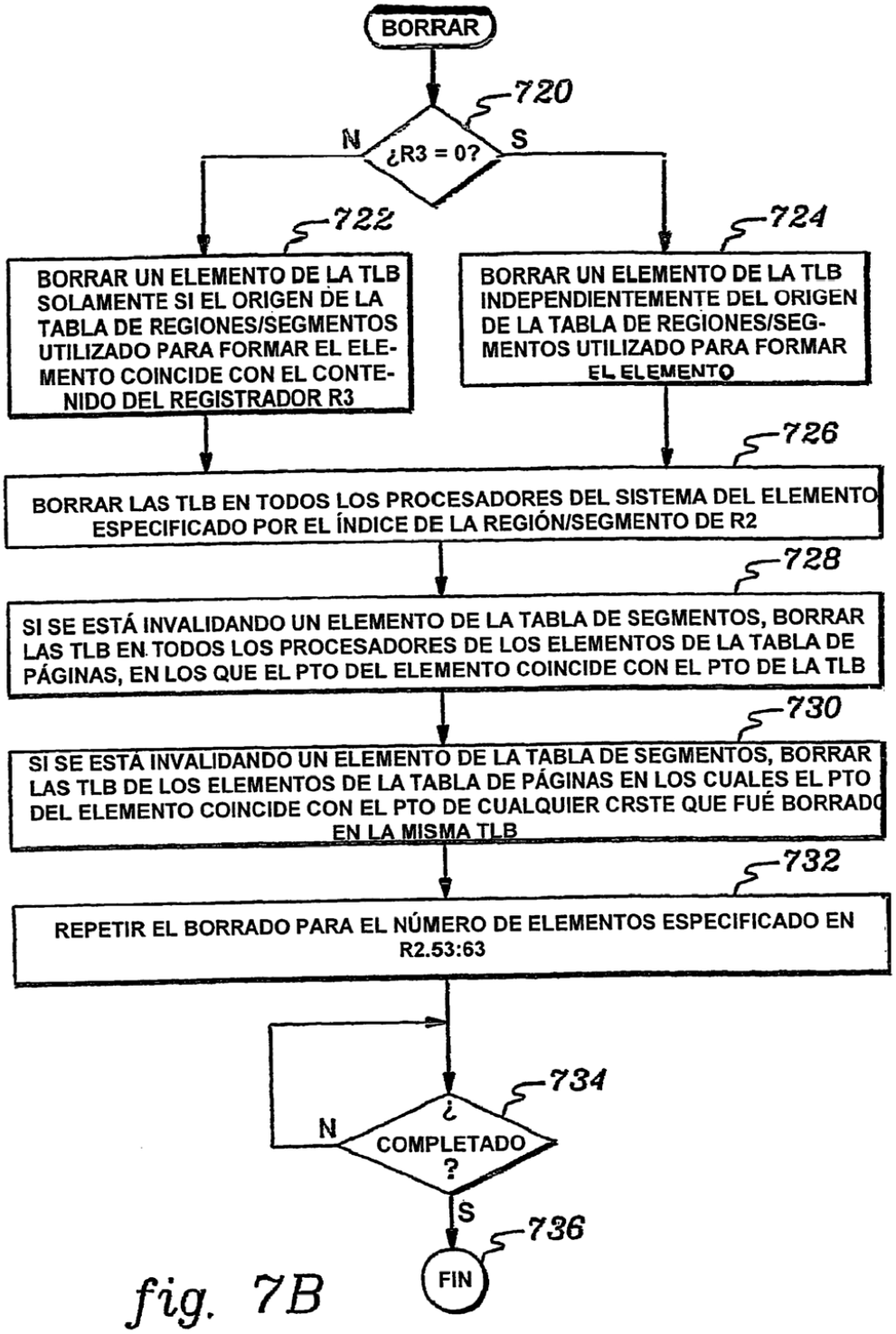
BORRAR
720
¿R3 = 0?
N
S
722
BORRAR UN ELEMENTO DE LA TLB SOLAMENTE SI EL ORIGEN DE LA TABLA DE REGIONES/SEGMENTOS UTILIZADO PARA FORMAR EL ELEMENTO COINCIDE CON EL CONTENIDO DEL REGISTRADOR R3
724
BORRAR UN ELEMENTO DE LA TLB INDEPENDIENTEMENTE DEL ORIGEN DE LA TABLA DE REGIONES/SEGMENTOS UTILIZADO PARA FORMAR EL ELEMENTO
726
BORRAR LAS TLB EN TODOS LOS PROCESADORES DEL SISTEMA DEL ELEMENTO ESPECIFICADO POR EL ÍNDICE DE LA REGIÓN/SEGMENTO DE R2
728
SI SE ESTÁ INVALIDANDO UN ELEMENTO DE LA TABLA DE SEGMENTOS, BORRAR LAS TLB EN TODOS LOS PROCESADORES DE LOS ELEMENTOS DE LA TABLA DE PÁGINAS, EN LOS QUE EL PTO DEL ELEMENTO COINCIDE CON EL PTO DE LA TLB
730
SI SE ESTÁ INVALIDANDO UN ELEMENTO DE LA TABLA DE SEGMENTOS, BORRAR LAS TLB DE LOS ELEMENTOS DE LA TABLA DE PÁGINAS EN LOS CUALES EL PTO DEL ELEMENTO COINCIDE CON EL PTO DE CUALQUIER CRSTE QUE FUÉ BORRADO EN LA MISMA TLB
732
REPETIR EL BORRADO PARA EL NÚMERO DE ELEMENTOS ESPECIFICADO EN R2.53:63
734
¿ COMPLETADO ?
N
S
736
FIN

fig. 7B

*fig. 8A*

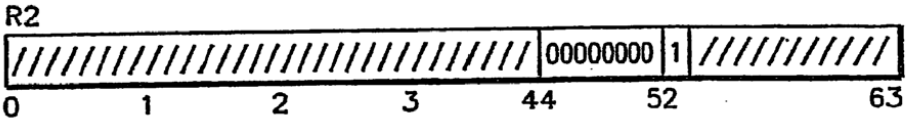


*fig. 8B*

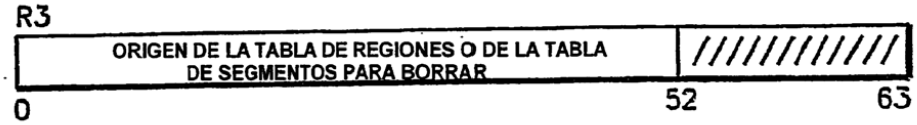


*fig. 8C*

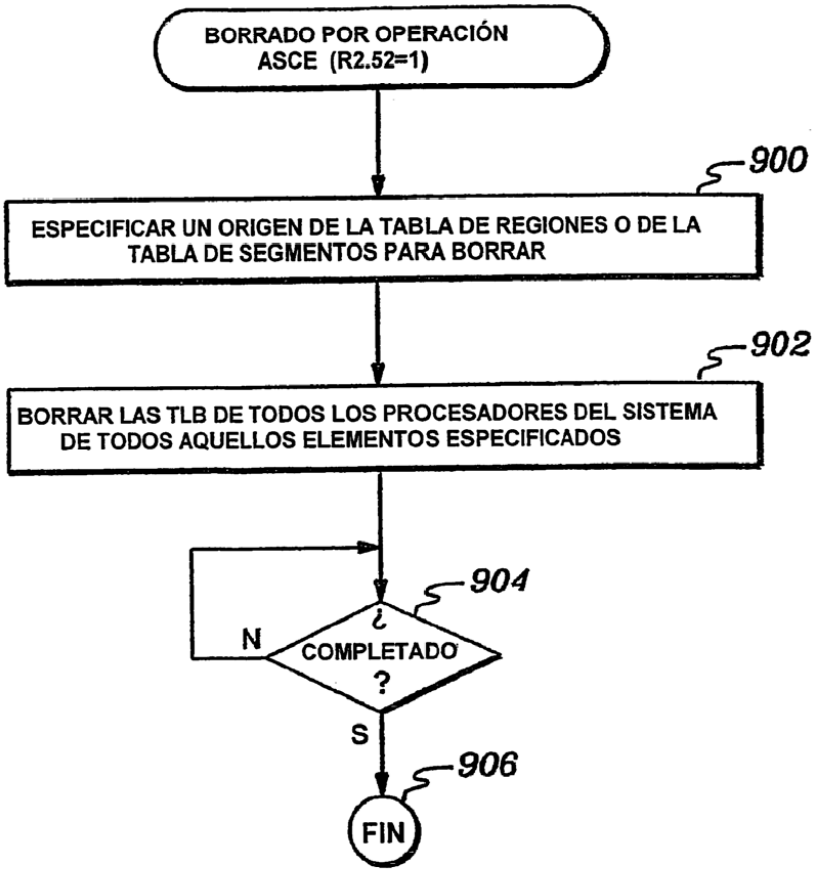
BORRADO POR OPERACIÓN ASCE (R2.52=1)
900
ESPECIFICAR UN ORIGEN DE LA TABLA DE REGIONES O DE LA TABLA DE SEGMENTOS PARA BORRAR
902
BORRAR LAS TLB DE TODOS LOS PROCESADORES DEL SISTEMA DE TODOS AQUELLOS ELEMENTOS ESPECIFICADOS
904
i
COMPLETADO ?
N
S
906
FIN

*fig. 9*