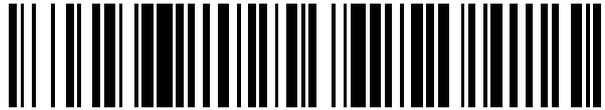


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 620 704**

51 Int. Cl.:

G06F 9/46 (2006.01)

G06F 9/38 (2006.01)

G06F 9/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.11.2012 PCT/IB2012/056625**

87 Fecha y número de publicación internacional: **19.12.2013 WO2013186601**

96 Fecha de presentación y número de la solicitud europea: **22.11.2012 E 12878682 (9)**

97 Fecha y número de publicación de la concesión europea: **22.02.2017 EP 2834736**

54 Título: **Instrucción de almacenamiento no transaccional**

30 Prioridad:

15.06.2012 US 201213524887

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

29.06.2017

73 Titular/es:

**INTERNATIONAL BUSINESS MACHINES CORPORATION (100.0%)
New Orchard Road
Armonk, NY 10504, US**

72 Inventor/es:

**GREINER, DAN;
JACOBI, CHRISTIAN y
SLEGEL, TIMOTHY**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 620 704 T3

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

DESCRIPCIÓN

Instrucción de almacenamiento no transaccional

Campo técnico

5 La presente invención se refiere, en general, a entornos informáticos de multitratamiento, y en particular, al tratamiento transaccional dentro de tales entornos informáticos.

Antecedentes

10 Un desafío duradero en programación de multiprocesador es el de actualizaciones a la misma ubicación de almacenamiento por múltiples unidades de tratamiento central (CPU). Muchas instrucciones que actualizan ubicaciones de almacenamiento, incluyendo operaciones lógicas simples, tales como AND (Y), lo hacen con múltiples accesos a la ubicación. Por ejemplo, en primer lugar, se va a recuperar la ubicación de almacenamiento, y a continuación, el resultado actualizado es almacenado nuevamente.

15 Para que múltiples CPU actualicen de forma segura la misma ubicación de almacenamiento, se "serializa" (se convierte a una serie de bytes) el acceso a la ubicación. Una instrucción, la instrucción PRUEBA Y CONFIGURACIÓN (TEST AND SET), introducida con la arquitectura S/360 ofrecida anteriormente por la International Business Machines Corporation, proporcionaba una actualización entrelazada de una ubicación de almacenamiento. Una actualización entrelazada significa que, cuando es observada por otras CPU y el subsistema de entrada/salida (I/O) (por ejemplo, subsistema de canal), el acceso de almacenamiento completo de la instrucción parece ocurrir atómicamente. Después, la arquitectura S/370 ofrecida por International Business Machines Corporation introdujo las instrucciones de COMPARACIÓN Y CAMBIO ("COMPARE AND SWAP") y de COMPARACIÓN POR DUPLICADO Y CAMBIO ("COMPARE DOUBLE AND SWAP") que proporcionan un medio más sofisticado para realizar la actualización entrelazada, y permiten la implementación de lo que comúnmente se conoce como palabra de bloqueo (o semáforo). Recientemente las instrucciones añadidas han proporcionado capacidades de actualización entrelazada adicionales, incluyendo COMPARACIÓN Y CAMBIO Y PURGADO ("COMPARE AND SWAP AND PURGE"), y COMPARACIÓN Y CAMBIO Y ALMACENAMIENTO ("COMPARE AND SWAP AND STORE"). Sin embargo, todas estas instrucciones proporcionan interconexión solamente para una ubicación de almacenamiento única.

20 Técnicas de programa más complejas pueden requerir la actualización entrelazada de múltiples ubicaciones de almacenamiento, tales como cuando se añade un elemento a una lista doblemente vinculada. En tal operación, tanto un puntero hacia delante como hacia atrás han de parecer actualizados simultáneamente, cuando son observados por otras CPU y el subsistema I/O. Con el fin de efectuar tal actualización de múltiples ubicaciones, se fuerza al programa a utilizar un solo punto de "serialización" separado, tal como una palabra de bloqueo. Sin embargo, las palabras de bloqueo pueden proporcionar un nivel mucho más aproximado de "serialización" que el garantizado; por ejemplo, las palabras de bloqueo puede "serializar" toda una cola de millones de elementos, incluso aunque solamente se actualicen dos elementos. El programa puede estructurar los datos para usar la "serialización" de grano fino (por ejemplo, una jerarquía de puntos de bloqueo), pero que introduce problemas adicionales, tales como situaciones de entrelazado potenciales si se viola la jerarquía, y problemas de recuperación si el programa encuentra un error mientras mantiene uno o más bloqueos o si no se puede adquirir el bloqueo.

25 Además de lo anterior, existen numerosos escenarios donde un programa puede ejecutar una secuencia de instrucciones que pueden o no dar como resultado una condición de excepción. Si no se produce la condición de excepción, entonces el programa continúa; sin embargo, si se reconoce una excepción, entonces el programa puede tomar acciones correctivas para eliminar la condición de excepción. Java, como ejemplo, puede explotar tal ejecución, por ejemplo, en ejecución especulativa, revestimiento interno parcial de una función, y/o en la nueva secuenciación de la comprobación nula de puntero.

30 En entornos de sistema operativo clásico, tal como z/OS y sus predecesores ofrecidos por International Business Machines Corporation, el programa establece un entorno de recuperación para interceptar cualquier condición de excepción de programa que pueda encontrar. Si el programa no intercepta la excepción, el sistema operativo termina típicamente de manera anormal el programa para excepciones que el sistema operativo no está preparado para manejar. Establecer y explotar tal entorno es costoso y complicado.

Compendio

35 Se abordan las deficiencias de la técnica anterior y se proporcionan ventajas mediante la provisión de un producto de programa informático para ejecutar una instrucción dentro de un entorno informático. El producto de programa informático incluye un medio de almacenamiento legible por ordenador, legible mediante un circuito de tratamiento e instrucciones de almacenamiento para la ejecución mediante el circuito de tratamiento para realizar un método. El método incluye, por ejemplo, obtener, mediante el circuito de tratamiento, una instrucción de máquina para la ejecución, siendo definida la instrucción de máquina para ejecución por ordenador de acuerdo con una arquitectura informática, comprendiendo la instrucción de máquina: un código de operación para especificar una operación de almacenamiento no transaccional: un campo para especificar un registro, en donde los contenidos del registro son

un primer operando; y al menos un campo para especificar una dirección del segundo operando, sirviendo la dirección del segundo operando para designar una ubicación en la memoria; y ejecutar, mediante el procesador, la instrucción de máquina, comprendiendo la ejecución: almacenar de manera no transaccional el primer operando en la ubicación especificada por la dirección del segundo operando, en donde el primer operando almacenado en la ubicación es retenido a pesar de una anulación de una transacción asociada con la instrucción de máquina, caracterizado por que el almacenamiento no transaccional es retrasado hasta un final del modo de ejecución transaccional del procesador.

Los métodos y sistemas relativos a una o más realizaciones son descritos y reivindicados también en este documento.

10 Se consiguen características y ventajas adicionales. Otras realizaciones y aspectos son descritos en detalle en este documento y son considerados una parte de la invención reivindicada.

Breve descripción de los dibujos

Las realizaciones de la presente invención serán ahora descritas, a modo de ejemplo solamente, con referencia a los dibujos adjuntos en los que:

- 15 La fig. 1 representa una realización de un entorno informático;
La fig. 2A representa un ejemplo de una instrucción de comienzo de transacción (TBEGIN);
La fig. 2B representa una realización de otros detalles de un campo de la instrucción de TBEGIN de la fig. 2A;
La fig. 3A representa un ejemplo de una instrucción restringida de Comienzo de Transacción (TBEGINC);
La fig. 3B representa una realización de otros detalles de un campo de la instrucción de TBEGINC de la fig. 3A;
- 20 La fig. 4 representa un ejemplo de una instrucción de Final de Transacción (TEND);
La fig. 5 representa un ejemplo de una instrucción de Anulación de Transacción (TABORT);
La fig. 6 representa un ejemplo de transacciones anidadas;
La fig. 7 representa un ejemplo de una instrucción de ALMACENAMIENTO NO TRANSACCIONAL (NTSTG);
La fig. 8 representa un ejemplo de una instrucción de EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN (ETND);
- 25 La fig. 9 representa un ejemplo de un bloque de diagnóstico de transacción;
La fig. 10 representa razones ejemplares para anulación, junto con códigos de anulación y códigos de condición asociados;
- 30 La fig. 11A representa un ejemplo de la lógica asociada con la instrucción de ALMACENAMIENTO NO TRANSACCIONAL;
La fig. 11B representa un ejemplo de lógica adicional asociada con la instrucción de ALMACENAMIENTO NO TRANSACCIONAL;
- La fig. 12 representa una realización de un producto de programa informático;
La fig. 13 representa una realización de un sistema informático anfitrión;
- 35 La fig. 14 representa otro ejemplo de un sistema informático;
La fig. 15 representa otro ejemplo de un sistema informático que comprende una red informática;
La fig. 16 representa una realización de distintos elementos de un sistema informático;
La fig. 17A representa una realización de la unidad de ejecución del sistema informático de la fig. 16;
La fig. 17B representa una realización de la unidad de rama del sistema informático de la fig. 16;
- 40 La fig. 17C representa una realización de la unidad de carga/almacenamiento del sistema informático de la fig. 16; y
La fig. 18 representa una realización de un sistema informático anfitrión emulado.

Descripción detallada

Según una realización, se proporciona una instalación de ejecución transaccional (TX). Esta instalación proporciona el tratamiento transaccional para instrucciones, y en una o más realizaciones, ofrece diferentes modos de ejecución, como se ha descrito a continuación, así como niveles anidados de tratamiento transaccional.

5 La instalación de ejecución transaccional introduce un estado de la CPU llamado el modo de ejecución transaccional (TX). Después del reinicio de la CPU, la CPU no está en el modo TX. La CPU entra en el modo TX mediante una instrucción de COMIENZO DE TRANSACCIÓN. La CPU deja el modo TX bien por (a) una instrucción de FINAL DE TRANSACCIÓN más exterior (a continuación se darán más detalles sobre interior y exterior), o bien por (b) anulando la transacción. Mientras en el modo TX, los accesos al almacenamiento mediante la CPU parecen ser bloques concurrentes cuando es observado por otras CPU y el subsistema I/O. Los accesos al almacenamiento son bien (a) 10 consignados a almacenamiento cuando la transacción más exterior finaliza sin anulación (es decir, por ejemplo, actualizaciones hechas en una memoria caché o memoria temporal local a la CPU son propagadas y almacenadas en memoria real y visibles para otras CPU), o (b) descartados si la transacción es anulada.

15 Las transacciones pueden ser anidadas. Es decir, mientras la CPU está en el modo TX, puede ejecutar otra instrucción COMIENZO DE TRANSACCIÓN. La instrucción que provoca que la CPU entre en modo TX es llamada el COMIENZO DE TRANSACCIÓN más exterior; de manera similar, se dice que el programa se encuentra en la transacción más exterior. Ejecuciones posteriores del COMIENZO DE TRANSACCIÓN son llamadas instrucciones internas; y el programa está ejecutando una transacción interna. El modelo proporciona una profundidad de anidación mínima y una profundidad de anidación máxima dependiente del modelo. Una instrucción de EXTRACCION DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN devuelve el valor de profundidad de 20 anidación actual, y en otra realización, puede devolver un valor de profundidad de anidación máxima. Esta técnica utiliza un modelo llamado "anidación aplanada" en el cual una condición de anulación a cualquier profundidad de anidación provoca que todos los niveles de la transacción sean anulados, y el control es devuelto a la instrucción siguiente al COMIENZO DE TRANSACCIÓN más exterior.

25 Durante el tratamiento de una transacción, un acceso transaccional hecho mediante una CPU se dice que entra en conflicto con bien (a) un acceso transaccional o acceso no transaccional hecho mediante otra CPU, o bien (b) un acceso no transaccional hecho mediante el subsistema I/O, si ambos accesos están en cualquier ubicación dentro de la misma línea de memoria caché, y uno o ambos de los accesos es un almacén. En otras palabras, con el fin de que la ejecución transaccional sea productiva, la CPU no ha de ser observada haciendo accesos transaccionales hasta su consignación. Este modelo de programación puede ser muy efectivo en ciertos entornos; por ejemplo, la actualización de dos puntos en una lista doblemente vinculada de un millón de elementos. Sin embargo, puede ser 30 menos efectiva, si existe mucha competencia para las ubicaciones de almacenamiento que están siendo accedidas de manera transaccional.

35 En un modelo de ejecución transaccional, (referido en este documento como una transacción no restringida), cuando es anulada una transacción, el programa puede bien intentar volver a conducir la transacción con la esperanza de que la condición de anulación ya no esté presente, o bien el programa puede "retroceder" a un trayecto no transaccional equivalente. En otro modelo de ejecución transaccional (referido en este documento como una transacción restringida), una transacción anulada es automáticamente reconducida mediante la CPU; en la ausencia de violaciones de restricción, la transacción restringida es asegurada para una finalización eventual.

40 Cuando se inicia una transacción, el programa puede especificar distintos controles, tales como (a) aquellos cuyos registros generales son restaurados a sus contenidos originales si es anulada la transacción, (b) si se permite que la transacción modifique el contexto de registro de coma flotante, por ejemplo, incluyendo, por ejemplo, los registros de coma flotante y el registro de control de coma flotante, (c) si se permite que la transacción modifique registros de acceso (AR), y (d) si se han de bloquear ciertas condiciones de excepción de programa para que no causen una interrupción. Si se anula una transacción no restringida, pueden ser proporcionadas distintas informaciones de diagnóstico. Por ejemplo, la instrucción TBEGIN más exterior que inicia una transacción no restringida puede 45 designar un bloque de diagnóstico de transacción (TDB) especificado por programa. Además, el TDB en el área de prefijo de la CPU o designado mediante la descripción de estado del anfitrión puede ser utilizado también si la transacción es anulada debido a una interrupción de programa o a una condición que hace que la ejecución interpretativa finalice, respectivamente.

50 Se han indicado anteriormente distintos tipos de registros. Estos son explicados además en detalle en este documento. Los registros generales pueden ser utilizados como acumuladores en operaciones de aritmética general y lógicas. En una realización, cada registro contiene 64 posiciones de bit, y existen 16 registros generales. Los registros generales son identificados por los números 0-15, y son designados por un campo R de cuatro bits en una instrucción. Algunas instrucciones proporcionan medios para direccionar múltiples registros generales que tienen 55 varios campos R. Para algunas instrucciones, el uso de un registro general específico es implicado en vez de ser explícitamente designado mediante un campo R de la instrucción.

60 Además de su uso como acumuladores en operaciones de aritmética general y lógica, 15 de los 16 registros generales son utilizados también como dirección base y registros de índice en generación de dirección. En estos casos, los registros son designados por un campo B o un campo X de cuatro bits en una instrucción. Un valor de cero en el campo B o X especifica que no ha de aplicarse base o índice, y así, el registro general 0 no ha de ser

designado como que contiene una dirección base o índice.

Las instrucciones de coma flotante utilizan un conjunto de registros de coma flotante. La CPU tiene 16 registros de coma flotante, en una realización. Los registros de coma flotante son identificados por los números 0-15, y son designados mediante un campo R de cuatro bits en las instrucciones de coma flotante. Cada registro de coma flotante es de 64 bits de largo y puede contener bien un operando de coma flotante corto (32 bits) o bien uno largo (64 bits).

Un registro de control de coma flotante (FPC) es un registro de 32 bits que contiene bits de máscara, bits de indicador, un código de excepción de datos y bits de modo de redondeo, y es utilizado durante el tratamiento de las operaciones de coma flotante.

Además, en una realización, la CPU tiene 16 registros de control, teniendo cada uno posiciones de 64 bits. Las posiciones de bit en los registros son asignadas a instalaciones particulares en el sistema, tal como Grabación de Eventos de Programa (PER) (descrito posteriormente), y son utilizados bien para especificar que una operación puede tener lugar o para proporcionar información especial requerida por la instalación. En una realización, para la instalación transaccional, son utilizados CR0 (bits 8 y 9) y CR2 (bits 61-63), como se ha descrito posteriormente.

La CPU tiene, por ejemplo, 16 registros de acceso numerados de 0-15. Un registro de acceso consiste de posiciones de 32 bits que contienen una especificación indirecta de un elemento de control de espacio de dirección (ASCE). Un elemento de control de espacio de dirección es un parámetro utilizado por el mecanismo de traducción de dirección dinámica (DAT) para traducir referencias a un espacio de dirección correspondiente. Cuando la CPU está en un modo llamado el modo de registro de acceso (controlado por los bits en la palabra de estado de programa (PSW)), un campo B de instrucción, utilizado para especificar una dirección lógica para una referencia de operando de almacenamiento, designa un registro de acceso, y el elemento de control de espacio de dirección especificado por el registro de acceso es utilizado por la DAT para que se haga la referencia. Para algunas instrucciones, un campo R es utilizado en vez de un campo B. Las instrucciones son proporcionadas para cargar y almacenar los contenidos de los registros de acceso y para mover los contenidos de un registro de acceso a otro.

Cada uno de los registros de acceso 1-15 puede designar cualquier espacio de dirección. El registro de acceso 0 designa el espacio de instrucción primario. Cuando uno de los registros de acceso 1-15 es utilizado para designar un espacio de dirección, la CPU determina qué espacio de dirección es designado traduciendo los contenidos del registro de acceso. Cuando el registro de acceso 0 es utilizado para designar un espacio de dirección, la CPU trata el registro de acceso como designación de un espacio de instrucción primario, y no examina el contenido real del registro de acceso. Por tanto, los 16 registros de acceso pueden designar, en cualquier momento, el espacio de instrucción primario y un máximo de otros 15 espacios.

En una realización, existen múltiples tipos de espacios de dirección. Un espacio de dirección es una secuencia consecutiva de números enteros (direcciones virtuales), junto con los parámetros de transformación específicos que permiten a cada número ser asociado con una ubicación de byte en el almacenamiento. La secuencia comienza en cero y prosigue de izquierda a derecha.

Por ejemplo, en la Arquitectura z, cuando una dirección virtual es utilizada por una CPU para acceder al almacenamiento principal (también conocido como Memoria principal), es convertida en primer lugar, por medio de la traducción de dirección dinámica (DAT), a una dirección real, y a continuación, por medio del prefijo, a una dirección absoluta. La DAT puede utilizar de uno a cinco niveles de tablas (página, segmento, tercera región, segunda región y primera región) como parámetros de transformación. La designación (origen y longitud) de la tabla de nivel más alto para un espacio de dirección específico es llamada un elemento de control de espacio de dirección, y es encontrada para ser utilizada por la DAT en un registro de control o como especificada por un registro de acceso. Alternativamente, el elemento de control de espacio de dirección para un espacio de dirección puede ser una designación de espacio real, que indica que la DAT ha de traducir la dirección virtual simplemente tratándola como una dirección real y sin utilizar ninguna tabla.

La DAT utiliza, en diferentes momentos, los elementos de control de espacio de dirección en diferentes registros de control o especificados mediante los registros de acceso. La elección es determinada por el modo de traducción especificado en la PSW actual. Hay disponibles cuatro modos de traducción: modo de espacio primario, modo de espacio secundario, modo de registro de acceso y modo de espacio doméstico. Los diferentes espacios de dirección se pueden direccionar dependiendo del modo de traducción.

En cualquier momento cuando la CPU está en el modo de espacio primario o modo de espacio secundario, la CPU puede traducir direcciones virtuales que pertenecen a os espacios de dirección - el espacio de dirección primario y el espacio de dirección secundario. En cualquier momento cuando la CPU está en el modo de registro de acceso, puede traducir direcciones virtuales de hasta 16 espacios de dirección - espacio de dirección primario y hasta 15 espacios de dirección especificados AR. En cualquier momento cuando la CPU está en el modo de espacio doméstico, puede traducir direcciones virtuales del espacio de dirección doméstico.

El espacio de dirección primario es identificado como tal porque consiste de direcciones virtuales primarias, que son traducidas por medio del elemento de control de espacio de dirección (ASCE) primario. De manera similar, el

espacio de dirección secundario consiste de direcciones virtuales secundarias traducidas por medio del ASCE secundario; los espacios de dirección especificados por el AR consisten de direcciones virtuales especificadas por el AR traducidas por medio de los ASCE especificados por el AR; y el espacio de dirección doméstico consiste de direcciones virtuales domésticas traducidas por medio del ASCE doméstico. Los ASCE primario y secundario están en los registros de control 1 y 7, respectivamente. Los ASCE especificados por el AR están en las entradas de la segunda tabla ASN que son ubicadas a través de un proceso llamado traducción de registro de acceso (ART) que utiliza el registro de control 2, 5 y 8. El ASCE doméstico está en el registro de control 13.

Una realización de un entorno informático para incorporar y utilizar uno o más aspectos de la instalación transaccional descrita en este documento es descrita con referencia a la fig. 1.

Con referencia a la fig. 1, en un ejemplo, el entorno informático 100 está basado en la Arquitectura z, ofrecida por International Business Machines (IBM®) Corporation, Armonk, New York. La Arquitectura z es descrita en una publicación de IBM titulada "z/Architecture – Principles of Operation", Publicación N° SA22-7932-08, 9ª Edición, Agosto de 2010, que es incorporada aquí por referencia en su totalidad.

La Z/ARCHITECTURE, IBM, y Z/OS y Z/VM (referenciadas posteriormente) son marcas registradas de International Business Machines Corporation, Armonk, New York. Otros nombres utilizados aquí pueden ser marcas registradas, nombres de marcas registradas o productos registrados de International Business Machines Corporation u otras compañías.

Como ejemplo, el entorno informático 100 incluye un complejo de procesador central (CPC) 102 acoplado a uno o más dispositivos 106 de entrada/salida (I/O) a través de una o más unidades de control 108. El complejo de procesador central 102 incluye, por ejemplo, uno o más procesadores centrales 110, una o más particiones 112 (por ejemplo particiones lógicas (LP)), un hipervisor de partición lógica 114, y un subsistema de entrada/salida 115, cada uno de los cuales es descrito posteriormente.

Los procesadores centrales 110 son recursos físicos del procesador asignados a las particiones lógicas. En particular, cada partición lógica 112 tiene uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte compartida de un procesador físico 110 asignado a la partición. Los procesadores lógicos de una partición particular 112 pueden ser bien dedicados a la partición, de manera que el recurso del procesador subyacente 110 está reservado para esa partición; o es compartido con otra partición, de manera que el recurso de procesador subyacente esté potencialmente disponible para otra partición.

Una partición lógica funciona como un sistema separado y tiene una o más aplicaciones, y opcionalmente, un sistema operativo residente en ella que puede diferir para cada partición lógica. En una realización, el sistema operativo es el sistema operativo z/OS, el sistema operativo z/VM, el sistema operativo z/Linux, o el sistema operativo TPF, ofrecido por International Business Machines Corporation, Armonk, New York. Las particiones lógicas 112 son gestionadas por un hipervisor 114 de partición lógica, que es implementado mediante el firmware que se ejecuta en los procesadores 110. Como se ha utilizado en este documento, el firmware incluye, por ejemplo, el micro-código y/o mili-código del procesador. Incluye, por ejemplo, las instrucciones de nivel de hardware y/o estructuras de datos utilizadas en la implementación de código de máquina de nivel más alto. En una realización, incluye, por ejemplo, código propietario que es entregado típicamente como micro-código que incluye software de confianza o micro-código específico para el hardware subyacente y controla el acceso del sistema operativo al hardware del sistema.

Las particiones lógicas y el hipervisor de partición lógica comprenden cada uno de ellos uno o más programas residentes en particiones respectivas de almacenamiento central asociadas con los procesadores centrales. Un ejemplo del hipervisor 114 de partición lógica es el Recurso de Procesador/Gestor de Sistema (PR/SM), ofrecido por International Business Machines Corporation, Armonk, New York.

El subsistema de entrada/salida 115 dirige el flujo de información entre los dispositivos 106 de entrada/salida y el almacenamiento principal (también conocido como Memoria principal). Está acoplado al complejo de tratamiento central, porque puede ser una parte del complejo de tratamiento central o estar separado del mismo. El subsistema I/O libera los procesadores centrales de la tarea de comunicar directamente con los dispositivos de entrada/salida y permite que el tratamiento de datos prosiga de manera concurrente con el tratamiento de entrada/salida. Para proporcionar comunicaciones, el subsistema I/O emplea adaptadores de comunicaciones I/O. Hay distintos tipos de adaptadores de comunicaciones incluyendo por ejemplo, canales, adaptadores I/O, tarjetas PCI, tarjetas Ethernet, tarjetas de Pequeña Interfaz de Almacenamiento de Ordenador (SCSI), etc. En el ejemplo particular descrito en este documento, los adaptadores de comunicaciones I/O son canales, y por lo tanto, el subsistema I/O es referido aquí como un subsistema de canal. Sin embargo, esto es solamente un ejemplo. Pueden ser utilizados otros tipos de subsistemas I/O.

El subsistema I/O utiliza uno o más trayectos de entrada/salida como enlaces de comunicación en la gestión del flujo de información a o desde dispositivos 106 de entrada/salida. En este ejemplo particular, estos trayectos son llamados trayectos de canal, ya que los adaptadores de comunicación son canales.

El entorno informático descrito anteriormente solamente es un ejemplo de un entorno informático que puede ser

utilizado. Pueden ser utilizados otros entornos, que incluyen pero no están limitados a, entornos no particionados, otros entornos particionados, y/o entornos emulados; las realizaciones no están limitadas a ningún entorno.

Según uno o más aspectos, la instalación de ejecución transaccional es una mejora de la CPU que proporciona los medios por los que la CPU puede ejecutar una secuencia de instrucciones - conocida como una transacción -que puede acceder a múltiples ubicaciones de almacenamiento, incluyendo la actualización de dichas ubicaciones. Cuando es observada por otras CPU y el subsistema I/O, la transacción es o bien (a) completada en su totalidad como una operación atómica única, o bien (b) anulada, sin dejar potencialmente evidencia de que alguna vez se ejecutó (excepto para ciertas condiciones descritas en este documento). Así, una transacción completada con éxito puede actualizar numerosas ubicaciones de almacenamiento sin que sea necesario ningún bloqueo especial en el modelo de multi-tratamiento clásico.

La instalación de ejecución transaccional incluye, por ejemplo, uno o más controles; una o más instrucciones; tratamiento transaccional, incluyendo la ejecución restringida y no restringida; y tratamiento de anulación, cada uno de los cuales es además descrito posteriormente.

En una realización, se utilizan tres controles de propósito especial, incluyendo una Palabra de Estado de Programa (PSW) de anulación de transacción, una dirección de bloque de diagnóstico de transacción (TDB), y una profundidad de anidación de transacción; cinco bits de registro de control; y seis instrucciones generales, que incluyen COMIENZO DE TRANSACCIÓN (restringida y no restringida), FINAL DE TRANSACCIÓN, EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN, ANULACIÓN DE TRANSACCIÓN, y ALMACENAMIENTO NO TRANSACCIONAL, son utilizados para controlar la instalación de ejecución transaccional. Cuando la instalación es instalada, es instalada, por ejemplo, en todas las CPU en la configuración. Un bit 73 de indicación de instalación, en una implementación, cuando es uno, indica que la instalación de ejecución transaccional está instalada.

Cuando la instalación de ejecución transaccional está instalada, la configuración proporciona una instalación de ejecución transaccional no restringida, y opcionalmente, una instalación de ejecución transaccional restringida, cada una de las cuales es descrita posteriormente. Cuando las indicaciones de instalación 50 y 73, como ejemplos, son ambas uno, la instalación de ejecución transaccional restringida está instalada. Ambas indicaciones de instalación son almacenadas en la memoria en ubicaciones especificadas.

Como se ha utilizado en este documento, el nombre de instrucción COMIENZO DE TRANSACCIÓN se refiere a instrucciones que tienen la TBEGIN mnemotécnica (Comienzo de Transacción para una transacción no restringida) y TBEGINC (Comienzo de Transacción para una transacción restringida). Descripciones que pertenecen a una instrucción específica son indicadas mediante el nombre de la instrucción seguido por el mnemotécnico entre paréntesis o corchetes, o simplemente por el mnemotécnico.

Una realización de un formato de una instrucción de COMIENZO DE TRANSACCIÓN (TBEGIN) está representada en las figs. 2A-2B. Como un ejemplo, una instrucción TBEGIN 200 incluye un campo código de operación 202 que incluye un código de operación que especifica una operación no restringida de comienzo de transacción; un campo base (B_1) 204; un campo de desplazamiento (D_1) 206; y un campo inmediato (I_2) 208. Cuando el campo B_1 es distinto de cero, los contenidos del registro general especificado por B_1 204 son añadidos a D_1 206 para obtener la dirección del primer operando.

Cuando el campo B_1 es distinto de cero, se aplica lo siguiente:

- cuando la profundidad de anidación de transacción es inicialmente cero, la dirección del primer operando designa la ubicación del bloque de diagnóstico de transacción de 256 bytes, llamada TDB especificada por TBEGIN (descrita más adelante) en que distintas informaciones de diagnóstico puede ser almacenadas si la transacción es anulada. Cuando la CPU está en el modo de espacio primario o modo de registro de acceso, la dirección del primer operando designa una ubicación en el espacio de dirección primario. Cuando la CPU está en el modo de espacio secundario o de espacio doméstico, la dirección del primer operando designa una ubicación en el espacio de dirección secundario o doméstico, respectivamente. Cuando la DAT está desactivada, la dirección (TDBA) del bloque de diagnóstico de transacción (TDB) designa una ubicación en el almacenamiento real.
- Se determina la accesibilidad al almacén para el primer operando. Si es accesible, la dirección lógica del operando es colocada en la dirección de bloque de diagnóstico de transacción (TDBA), y la TDBA es válida.
- Cuando la CPU está ya en el modo de ejecución transaccional no restringido, la TDBA no es modificada, y es impredecible si el primer operando es probado para su accesibilidad.

Cuando el campo B_1 es cero, no se detectan excepciones de acceso para el primer operando y, para la instrucción TBEGIN más exterior, la TDBA es inválida.

Los bits del campo I_2 son definidos como sigue, en un ejemplo:

Máscara para Guardar el Registro General (GRSM) 210 (fig. 2B): los bits 0-7 del campo I_2 contienen la máscara

para guardar el registro general (GRSM). Cada bit de la GRSM representa una pareja par/impar de registros generales, donde el bit 0 representa los registros 0 y 1, el bit 1 representa los registros 2 y 3, y así sucesivamente. Cuando un bit en la GRSM de la instrucción TBEGIN más exterior es cero, la pareja de registro correspondiente no es guardada. Cuando un bit en la GRSM de la instrucción TBEGIN más exterior es uno, la pareja de registro correspondiente es guardada en una ubicación dependiente del modelo que no es accesible directamente mediante el programa.

Si se anula la transacción, las parejas de registro guardadas son restauradas a sus contenidos cuando la instrucción TBEGIN más exterior fue ejecutada. Los contenidos de todos los otros registros generales (no guardados) no son restaurados cuando se anula una transacción.

La máscara para guardar el registro general es ignorada en todas las instrucciones de TBEGIN excepto para la más exterior.

Permitir la Modificación de AR (A) 212: El control A, el bit 12 del campo I_2 , controla si se permite que la transacción modifique un registro de acceso. El control efectivo para permitir la modificación de AR es la AND lógica del control A en la instrucción de TBEGIN para el nivel de anidación actual y para todos los niveles exteriores.

Si el control efectivo A es cero, la transacción será anulada con el código de anulación 11 (instrucción restringida) si se hace un intento de modificar cualquier registro de acceso. Si el control efectivo A es uno, la transacción no será anulada si se modifica un registro de acceso (ausente de cualquier otra condición de anulación).

Permitir la Operación de Coma Flotante (F) 214: El control F, bit 13 del campo I_2 , controla si se permite que la transacción ejecute las instrucciones de coma flotante especificadas. El control efectivo para permitir la operación de coma flotante es la AND lógica del control F en la instrucción de TBEGIN para el nivel de anidación actual y para todos los niveles exteriores.

Si el control efectivo F es cero, entonces (a) la transacción será anulada con el código de anulación 11 (instrucción restringida) si se hace un intento para ejecutar una instrucción de coma flotante, y (b) el código de excepción de datos (DXC) en el byte 2 del registro de control de coma flotante (FPCR) no será configurado por ninguna condición de excepción del programa de excepción de datos. Si el control efectivo F es uno, entonces (a) la transacción no será anulada si se hace un intento de ejecutar una instrucción de coma flotante (ausencia de cualquier otra condición de anulación), y (b) el DXC en el FPCR puede ser configurado mediante una condición de excepción del programa de excepción de datos.

Control de Filtrado de Interrupción del Programa (PIFC) 216: Los bits 14-15 del campo I_2 son el control de filtrado de interrupción del programa (PIFC). El PIFC controla si ciertas clases de condiciones de excepción de programa (por ejemplo, excepción de direccionamiento, excepción de datos, excepción de operación, excepción de protección, etc.) que ocurren mientras la CPU está en el modo de ejecución transaccional dan como resultado una interrupción.

El PIFC efectivo es el valor más elevado del PIFC en la instrucción de TBEGIN para el nivel de anidación actual y para todos los niveles exteriores. Cuando el PIFC efectivo es cero, todas las condiciones de excepción del programa dan como resultado una interrupción. Cuando el PIFC efectivo es uno, las condiciones de excepción del programa que tienen una clase de ejecución transaccional de 1 y 2 da como resultado una interrupción. (Cada condición de excepción de programa es asignada al menos a una clase de ejecución transaccional, dependiendo de la severidad de la excepción. La severidad está basada en la probabilidad de recuperación durante una ejecución repetida de la transacción, y si el sistema operativo necesita ver la interrupción). Cuando el PIFC efectivo es dos, las condiciones de excepción de programa que tiene una clase de ejecución transaccional de 1 dan como resultado una interrupción. Un PIFC de 3 es reservado.

Los bits 8-11 del campo I_2 (bits 40-43 de la instrucción) son reservados y deberían contener ceros; de lo contrario, el programa no puede operar de manera compatible en el futuro.

Una realización de un formato de una instrucción restringida de Comienzo de Transacción (TBEGINC) es descrita con referencia a las FIGS. 3A-3B. En un ejemplo, TBEGINC 300 incluye un campo 302 de código de operación que incluye un código de operación que especifica una operación restringida de comienzo de transacción; un campo base (B_1) 304; un campo de desplazamiento (D_1) 306; y un campo inmediato (I_2) 308. Los contenidos del registro general especificado por B_1 304 son añadidos a D_1 306 para obtener la dirección del primer operando. Sin embargo, con la instrucción restringida de comienzo de transacción, la dirección del primer operando no es utilizada para acceder al almacenamiento. En su lugar, el campo B_1 de la instrucción incluye ceros; de lo contrario, es reconocida una excepción de especificación.

En una realización, el campo I_2 incluye distintos controles, un ejemplo de los cuales está representado en la fig. 3B.

Los bits del campo I_2 son definidos como sigue, en un ejemplo:

Máscara para Guardar el Registro General (GRSM) 310: Los bits 0-7 del campo I_2 contienen la máscara para guardar el registro general (GRSM). Cada bit de la GRSM representa una pareja par-impar de registros generales,

donde el bit 0 representa los registros 0 y 1, el bit 1 representa los registros 2 y 3, y así sucesivamente. Cuando un bit en la GRSM es cero, la pareja de registro correspondiente no es guardada. Cuando un bit en la GRSM es uno, la pareja de registro correspondiente es guardada en una ubicación dependiente del modelo que no es accesible de manera directa mediante el programa.

- 5 Si la transacción se anula, las parejas de registro guardadas son restauradas a sus contenidos cuando la instrucción de COMIENZO DE TRANSACCIÓN más exterior fue ejecutada. Los contenidos de todos los demás registros generales (no guardados) no son restaurados cuando se anula una transacción restringida.

Cuando TBEGINC es utilizada para continuar la ejecución en el modo de ejecución de transacción no restringida, es ignorada la máscara para guardar el registro general.

- 10 **Permitir la modificación de AR (A) 312:** El control A, bit 12 del campo I₂, controla si la transacción es permitida para modificar un registro de acceso. El control efectivo de permiso de modificación de AR es la AND lógica del control A en la instrucción de TBEGINC para el nivel de anidación actual y para cualesquiera instrucciones de TBEGIN o TBEGINC más exteriores.

- 15 Si el control efectivo A es cero, la transacción será anulada con el código de anulación 11 (instrucción restringida) si se hace un intento de modificar cualquier registro de acceso. Si el control efectivo A es uno, la transacción no será anulada si se modifica un registro de acceso (ausencia de cualquier otra condición de anulación). Los bits 8-11 y 13-15 del campo I₂ (bits 40-43 y 45-47 de la instrucción) son reservados y deberían contener ceros.

- 20 El final de una instrucción de Comienzo de Transacción es especificado por una instrucción de FINAL DE TRANSACCIÓN (TEND), un formato de la cual está representado en la fig. 4. Como ejemplo, una instrucción 400 de TEND incluye un campo 402 código de operación que incluye un código de operación que especifica una operación de final de transacción.

Son utilizados un número de términos con respecto a la instalación de ejecución transaccional, y por lo tanto, únicamente por conveniencia, es proporcionada una lista de términos a continuación en orden alfabético. En una realización, estos términos tienen la siguiente definición:

- 25 **Anulación:** Una transacción se anula cuando se termina antes de una instrucción de FINAL DE TRANSACCIÓN que da como resultado una profundidad de anidación de transacción de cero. Cuando se anula una transacción, ocurre lo siguiente, en una realización:

- Se descartan accesos al almacenamiento transaccional hechos por cualquiera y todos los niveles de la transacción (es decir, no consignados).
- Se consignan accesos al almacén no transaccional hechos por cualquiera y todos los niveles de la transacción.
- Se restauran los registros designados por la máscara para guardar el registro general (GRSM) de la instrucción de COMIENZO DE TRANSACCIÓN más exterior a sus contenidos antes de la ejecución transaccional (es decir, a sus contenidos en la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN más exterior). No se restauran los registros generales no designados mediante la máscara para guardar el registro general de la instrucción de COMIENZO DE TRANSACCIÓN más exterior.
- No se restauran los registros de acceso, los registros de coma flotante, y el registro de control de coma flotante. Cualesquiera cambios hechos en estos registros durante la ejecución de la transacción son retenidos cuando se anula la transacción.

- 40 Una transacción puede ser anulada por distintas razones, incluyendo intento de ejecución de una instrucción restringida, intento de modificación de un recurso restringido, conflicto transaccional, exceder de distintos recursos de CPU, cualquier condición de intercepción de ejecución interpretativa, cualquier interrupción, una instrucción de ANULACIÓN DE TRANSACCIÓN, y por otras razones. Un código de anulación de transacción proporciona razones específicas por las cuales se puede anular una transacción.

- 45 Se ha descrito un ejemplo de un formato de una instrucción de ANULACIÓN DE TRANSACCIÓN (TABORT) con referencia a la fig. 5. Como ejemplo, una instrucción TABORT 500 incluye un campo 502 de código de operación que incluye un código de operación que especifica una operación de anulación de transacción; un campo base (B₂) 504; y un campo de desplazamiento (D₂) 506. Cuando el campo B₂ es distinto de cero, los contenidos del registro general especificados por B₂ 504 son añadidos a D₂ 506 para obtener una dirección del segundo operando; de lo contrario, la dirección del segundo operando está formada únicamente a partir del campo D₂, y el campo B₂ es ignorado. La dirección del segundo operando no es utilizada para datos de dirección; en su lugar, la dirección forma el código de anulación de transacción que es colocado en un bloque de diagnóstico de transacción durante el tratamiento de anulación. El cálculo de dirección para la dirección del segundo operando sigue las reglas de la aritmética de direcciones: en el modo de direccionamiento de 24 bits, los bits 0-29 son configurados a ceros; en el modo de direccionamiento de 31 bits, los bits 0-32 son configurados a ceros.

Consignación: A la terminación de una instrucción de FINAL DE TRANSACCIÓN más exterior, la CPU consigna los accesos de almacenamiento hechos por la transacción (es decir, la transacción más exterior y cualesquiera niveles anidados) de tal manera que son visibles a otras CPU y al subsistema I/O. Cuando son observados por otras CPU y por el subsistema I/O, todos los accesos de recuperación y almacenamiento hechos por todos los niveles anidados de la transacción parece que ocurren como una operación concurrente única cuando ocurre la consignación.

Los contenidos de los registros generales, los registros de acceso, los registros de coma flotante, y el registro de control de coma flotante no son modificados por el proceso de consignación. Cualesquiera cambios hechos a estos registros durante la ejecución transaccional son retenidos cuando los almacenamientos de transacción son consignados.

Conflicto: Un acceso transaccional hecho por una CPU está en conflicto bien con (a) un acceso transaccional o acceso no transaccional hecho por otra CPU, o bien con (b) el acceso no transaccional hecho por el subsistema I/O, si ambos accesos están en cualquier ubicación dentro de la misma línea de memoria caché, y uno o más de los accesos es un almacenamiento.

Un conflicto puede ser detectado por una ejecución especulativa de instrucciones de la CPU, incluso aunque el conflicto no pueda ser detectado en la secuencia conceptual.

Transacción restringida: Una transacción restringida es una transacción que se ejecuta en el modo de ejecución transaccional restringido y está sometida a las siguientes limitaciones:

- Hay disponible un subconjunto de las instrucciones generales.
- Puede ejecutarse un número limitado de instrucciones.
- Puede accederse a un número limitado de ubicaciones de operando de almacenamiento.
- La transacción está limitada a un solo nivel de anidación.

En ausencia de interrupciones o conflictos repetidos con otras CPU o el subsistema I/O, una transacción restringida se completa de manera eventual, así no se requiere una rutina de manejo de anulación. Las transacciones restringidas son descritas en detalle posteriormente.

Cuando una instrucción restringida de COMIENZO DE TRANSACCIÓN (TBEGIN) es ejecutada mientras la CPU está ya en el modo de ejecución de transacción no restringida, la ejecución continúa como una transacción no restringida anidada.

Modo de Ejecución Transaccional Restringido: Cuando la profundidad de anidación de transacción es cero, y una transacción es iniciada mediante una instrucción de TBEGIN, la CPU entra en el modo de ejecución transaccional restringido. Mientras la CPU está en el modo de ejecución transaccional restringido, la profundidad de anidación de transacción es uno.

Transacción Anidada: Cuando la instrucción de COMIENZO DE TRANSACCIÓN es emitida mientras la CPU está en el modo de ejecución transaccional no restringido, la transacción es anidada.

La instalación de ejecución transaccional utiliza un modelo llamado anidación aplanada. En el modo de anidación aplanada, los almacenamientos hechos por una transacción interior no pueden ser observados por otras CPU y por el subsistema I/O hasta que la transacción más exterior consigna sus almacenamientos. De manera similar, si se anula una transacción, se anulan todas las transacciones anidadas, y todos los almacenamientos transaccionales de todas las transacciones anidadas son descartados.

Un ejemplo de transacciones anidadas está representado en la fig. 6. Como se ha mostrado, una primera TBEGIN 600 comienza una transacción más exterior 601, la TBEGIN 602 comienza una primera transacción anidada, y la TBEGIN 604 comienza una segunda transacción anidada. En este ejemplo, la TBEGIN 604 y la TEND 606 definen una transacción más interior 608. Cuando se ejecuta la TEND 610, los almacenamientos transaccionales son consignados 612 para la transacción más exterior y todas las transacciones interiores.

Transacción no Restringida: Una transacción no restringida es una transacción que se ejecuta en el modo de ejecución transaccional no restringido. Aunque una transacción no restringida no está limitada en la forma en que lo está una transacción restringida, puede ser anulada aún debido a una variedad de causas.

Modo de Ejecución Transaccional no Restringida: Cuando una transacción es iniciada por la instrucción de TBEGIN, la CPU entra en el modo de ejecución transaccional no restringido. Mientras la CPU está en el modo de ejecución transaccional no restringido, la profundidad de anidación de transacción puede variar desde uno a la profundidad máxima de anidación de transacción.

Acceso No Transaccional: Los accesos no transaccionales son accesos de operando de almacenamiento hechos por la CPU cuando no está en el modo de ejecución transaccional (es decir, accesos de almacenamiento clásico

fuera de una transacción). Además, los accesos hechos por el subsistema I/O son accesos no transaccionales. De manera adicional, la instrucción de ALMACENAMIENTO NO TRANSACCIONAL puede ser utilizada para provocar un acceso al almacenamiento no transaccional mientras la CPU está en el modo de ejecución transaccional no restringido.

5 Una realización de un formato de una instrucción de ALMACENAMIENTO NO TRANSACCIONAL es descrita con referencia a la fig. 7. Como un ejemplo, una instrucción 700 de ALMACENAMIENTO NO TRANSACCIONAL incluye una pluralidad de campos 702a, 702b de código de operación que especifican un código de operación que designa una operación de almacenamiento no transaccional; un campo de registro (R₁) 704 que especifica un registro, cuyos contenidos son llamados el primer operando; un campo índice (X₂) 706; un campo base (B₂) 708; un primer campo de desplazamiento (DL₂) 710; y un segundo campo de desplazamiento (DH₂) 712. Los contenidos de los registros generales designados por los campos X₂ y B₂ son añadidos a los contenidos de una concatenación de contenidos de los campos DH₂ y DL₂ para formar la dirección del segundo operando. Cuando bien uno o ambos campos X₂ o B₂ son cero, el registro correspondiente no toma parte en la suma.

15 El primer operando de 64 bits es colocado de manera no transaccional sin cambios en la ubicación del segundo operando. El desplazamiento, formado por la concatenación de los campos DH₂ y DL₂, es tratado como un número entero binario con signo de 20 bits.

El segundo operando ha de ser alineado sobre un límite de doble palabra; de lo contrario, se reconoce la excepción de especificación y la operación es suprimida.

20 **Transacción Exterior/Más Exterior:** Una transacción con una profundidad de anidación de transacción de número inferior es una transacción exterior. Una transacción con un valor de profundidad de anidación de transacción de uno es la transacción más exterior.

25 Una instrucción de COMIENZO DE TRANSACCIÓN más exterior es la que es ejecutada cuando la profundidad de anidación de transacción es inicialmente cero. Una instrucción de FINAL DE TRANSACCIÓN más exterior es la que hace que la profundidad de anidación de transacción pase de uno a cero. Una transacción restringida es la transacción más exterior, en esta realización.

30 **Filtrado de Interrupción del Programa:** Cuando es anulada una transacción debido a ciertas condiciones de excepción de programa, el programa puede impedir opcionalmente que ocurra la interrupción. Esta técnica es llamada filtrado de interrupción del programa. El filtrado de interrupción del programa es sometida a la clase transaccional de la interrupción, al control efectivo de filtrado de interrupción de programa desde la instrucción de COMIENZO DE TRANSACCIÓN, y a la invalidación del filtrado de interrupción de programa de ejecución transaccional en el registro de control 0.

35 **Transacción:** Una transacción incluye los accesos de operando de almacenamiento hechos, y registros generales seleccionados alterados, mientras la CPU está en el modo de ejecución de transacción. Para una transacción no restringida, los accesos de operando de almacenamiento pueden incluir tanto accesos transaccionales como accesos no transaccionales. Para una transacción restringida, los accesos de operando de almacenamiento están limitados a accesos transaccionales. Cuando son observados por otras CPU y por el subsistema I/O, todos los accesos de operando de almacenamiento hechos por la CPU mientras está en el modo de ejecución de transacción parecen ocurrir como una única operación concurrente. Si se anula una transacción, los accesos de almacenamiento transaccional son descartados, y cualesquiera registros designados por la máscara para guardar el registro general de la instrucción de COMIENZO DE TRANSACCIÓN más exterior son restaurados a sus contenidos anteriores a la ejecución transaccional.

40 **Accesos Transaccionales:** Los accesos transaccionales son accesos de operando de almacenamiento hechos mientras la CPU está en el modo de ejecución transaccional, con la excepción de los accesos hechos por la instrucción de ALMACENAMIENTO NO TRANSACCIONAL.

45 **Modo de Ejecución Transaccional:** El término modo de ejecución transaccional (también conocido como modo de ejecución de transacción) describe la operación común tanto del modo de ejecución transaccional restringida como no restringida. Así, cuando se describe la operación, los términos no restringida y restringida son utilizados para calificar el modo de ejecución transaccional.

50 Cuando la profundidad de anidación de transacción es cero, la CPU no está en el modo de ejecución transaccional (también llamado el modo de ejecución no transaccional).

Cuando son observados por la CPU, las recuperaciones y almacenamientos hechos en el modo de ejecución transaccional no son diferentes de los hechos mientras no está en el modo de ejecución transaccional.

55 En una realización de la arquitectura z, la instalación de ejecución transaccional está bajo el control de los bits 8-9 del registro de control 0, los bits 61-63 del registro de control 2, y la profundidad de anidación de transacción, la dirección de bloque de diagnóstico de transacción, y la palabra de estado del programa (PSW) de anulación de transacción.

Después de un reinicio de la CPU inicial, los contenidos de las posiciones de bit 8-9 del registro de control 0, las posiciones 62-63 de bit del registro de control 2, y la profundidad de anidación de transacción son configurados a cero. Cuando el control de ejecución transaccional, bit 8 del registro de control 0, es cero, la CPU no puede ser colocada en el modo de ejecución transaccional.

5 Son descritos posteriormente otros detalles relacionados con los distintos controles.

Como se ha indicado, la instalación de ejecución transaccional es controlada por dos bits en el registro de control cero y tres bits en el registro de control dos. Por ejemplo:

Bits del Registro de Control 0: Las asignaciones de bits son las siguientes, en una realización:

10 Control de Ejecución Transaccional (TXC): El bit 8 del registro de control cero es el control de ejecución transaccional. Este bit proporciona un mecanismo por el cual el programa de control (por ejemplo, el sistema operativo) puede indicar si la instalación de ejecución transaccional puede ser utilizada por el programa o no. El bit 8 ha de ser uno para entrar con éxito en el modo de ejecución transaccional.

15 Cuando el bit 8 del registro de control 0 es cero, el intento de ejecución de las instrucciones de EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN, de COMIENZO DE TRANSACCIÓN y de FINAL DE TRANSACCIÓN da como resultado una ejecución de operación especial.

20 Una realización de un formato de una instrucción de EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN está descrita con referencia a la fig. 8. Como ejemplo, una instrucción 800 de EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN incluye un campo 802 de código de operación que especifica un código de operación que indica la operación de extracción de profundidad de anidación de transacción; y un campo 804 de registro R₁ que designa un registro general.

La profundidad de anidación de transacción actual es colocada en los bits 48-63 del registro general R₁. Los bits 0-31 del registro permanecen sin cambios, y los bits 32-47 del registro son configurados a cero.

En otra realización, la profundidad de anidación de transacción máxima es colocada también en el registro general R₁, tal como en los bits 16-31.

25 Invalidación del Filtrado de Interrupción de Programa de Ejecución de Transacción (PIFO): El bit 9 del registro de control cero es la anulación del filtrado de interrupción del programa de ejecución transaccional. Este bit proporciona un mecanismo por el cual el programa de control puede asegurar que cualquier condición de excepción de programa que ocurre mientras la CPU está en el modo de ejecución transaccional da como resultado una interrupción, independientemente del control efectivo de filtrado de interrupción de programa especificado o implicado por la instrucción o instrucciones de COMIENZO DE TRANSACCIÓN.

30 **Bits del Control de Registro 2:** Las asignaciones son las siguientes, en una realización:

Alcance de Diagnóstico de Transacción (TDS): El bit 61 del registro de control 2 controla la aplicabilidad del control de diagnóstico de transacción (TDC) en los bits 62-63 del registro, como sigue:

TDS

35	Valor	Significado
	0	El TDC se aplica independientemente de si la CPU está en el estado de problema o supervisor.
	1	El TDC se aplica solamente cuando la CPU está en el estado de problema. Cuando la CPU está en el estado supervisor, el tratamiento es como si el TDC contuviera cero.

40 Control de Diagnóstico de Transacción (TDC): Los bits 62-63 del registro de control 2 son un número entero sin signo de 2 bits que pueden ser utilizados para hacer que las transacciones sean anuladas aleatoriamente con propósitos de diagnóstico. La codificación del TDC es la siguiente, en un ejemplo:

TDC

	Valor	Significado
	0	Operación normal; las transacciones no son anuladas como resultado del TDC.
45	1	Anular cada transacción en una instrucción aleatoria, pero antes de la ejecución de la instrucción de FINAL DE TRANSACCIÓN más exterior.
	2	Anular transacciones aleatorias en una instrucción aleatoria.
	3	Reservado.

Cuando una transacción es anulada debido a un TDC distinto de cero, entonces puede ocurrir cualquiera de las siguientes situaciones:

- 5 • El código de anulación es configurado a cualquiera de los códigos 7-11, 13-16, o 255, con el valor del código elegido aleatoriamente por la CPU; el código de condición es configurado correspondientemente al código de anulación. Los códigos de anulación son descritos posteriormente.
- Para una transacción no restringida, el código de condición es configurado a uno. En este caso, no es aplicable el código de anulación.

Es dependiente del modelo si el valor 1 del TDC es implementado. Si no es implementado, un valor de 1 actúa como si fuese especificado como 2.

10 Para una transacción restringida, un valor de TDC de 1 es tratado como si fuese especificado un valor de TDC de 2.

Si es especificado un valor de TDC de 3, los resultados son impredecibles.

Dirección de Bloque de Diagnóstico de Transacción (TDBA)

15 Una dirección de bloque de diagnóstico de transacción (TDBA) válida es configurada desde la dirección del primer operando de la instrucción de COMIENZO DE TRANSACCIÓN (TBEGIN) más exterior cuando el campo B1 de la instrucción es distinto de cero. Cuando la CPU está en el espacio primario o modo de registro de acceso, la TDBA designa una ubicación en el espacio de dirección primario. Cuando la CPU está en el espacio secundario, o modo de espacio doméstico, la TDBA designa una ubicación en el espacio secundario o espacio de dirección doméstico, respectivamente. Cuando la DAT (Traducción de Dirección Dinámica) está desactivada, la TDBA designa una ubicación en el almacenamiento real.

20 La TDBA es utilizada por la CPU para ubicar el bloque de diagnóstico de transacción – llamado TDB especificado por TBEGIN - si la transacción es anulada después. Los tres bits más a la derecha de la TDBA son cero, lo que significa que el TDB especificado por TBEGIN está sobre un límite de doble palabra.

25 Cuando el campo B₁ de una instrucción de COMIENZO DE TRANSACCIÓN (TBEGIN) más exterior es cero, la dirección del bloque de diagnóstico transaccional es inválida, y el TDB especificado por TBEGIN no es almacenado si la transacción es anulada después.

PSW de Anulación de Transacción (TAPSW)

30 Durante la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN (TBEGIN) cuando la profundidad de anidación es inicialmente cero, la PSW de anulación de transacción es ajustada a los contenidos de la PSW actual; y la dirección de instrucción de la PSW de anulación de transacción designa la siguiente instrucción secuencial (es decir, la instrucción que sigue a la TBEGIN más exterior). Durante la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN restringida (TBEGINC) cuando la profundidad de anidación es inicialmente cero, la PSW de anulación de transacción es configurada a los contenidos de la PSW actual, excepto en que la dirección de instrucción de la PSW de anulación de transacción designa la instrucción de TBEGINC (en vez de la siguiente opción secuencial que sigue a la TBEGINC).

35 Cuando se anula una transacción, el código de condición en la PSW de anulación de transacción es reemplazado con un código que indica la severidad de la condición de anulación. Posteriormente, si la transacción fue anulada debido a causas que no dan como resultado una interrupción, la PSW es cargada a partir de la PSW de anulación de transacción; si la transacción fue anulada debido a causas que dan como resultado una interrupción, la PSW de anulación de transacción es almacenada como la PSW de antigua interrupción.

40 La PSW de anulación de transacción no es alterada durante la ejecución de cualquier instrucción de COMIENZO DE TRANSACCIÓN interna.

Profundidad de Anidación de Transacción (TND)

45 La profundidad de anidación de transacción es, por ejemplo, un valor sin signo de 16 bits que es aumentado cada vez que se completa una instrucción de COMIENZO DE TRANSACCIÓN con el código de condición 0 y disminuido cada vez que se completa una instrucción de FINAL DE TRANSACCIÓN. La profundidad de anidación de transacción es reiniciada a cero cuando la transacción es anulada o por reinicio de la CPU.

En una realización, es implementada una TND máxima de 15.

50 En una implementación, cuando la CPU está en el modo de ejecución transaccional restringido, la profundidad de anidación de transacción es uno. De manera adicional, aunque la TND máxima pueda ser representada como un valor de 4 bits, la TND es definida para ser un valor de 16 bits para facilitar su inspección en el bloque de diagnóstico de transacción.

Bloque de Diagnóstico de Transacción (TDB)

Cuando se anula una transacción, se puede guardar información de distintos estados en un bloque de diagnóstico de transacción (TDB), como sigue:

- 5 1. **TDB especificado por TBEGIN:** Para una transacción no restringida, cuando el campo B₁ de la instrucción de TBEGIN más exterior es distinto de cero, la dirección del primer operando de la instrucción designa el TDB especificado por TBEGIN. Esta es una ubicación especificada de programa de aplicación que puede ser examinada por el gestor de anulación de la aplicación.
- 10 2. **TDB de Interrupción del programa (PI):** Si una transacción no restringida es anulada debido a una condición de excepción de programa no filtrada, o si una transacción restringida es anulada debido a cualquier condición de excepción de programa (es decir, cualquier condición que da como resultado una interrupción de programa que es reconocida), LA PI-TDB es almacenada en las ubicaciones en el área de prefijo. Esto está disponible para que el sistema operativo inspeccione y cierre la sesión en cualquier informe de diagnóstico que pueda proporcionar.
- 15 3. **TDB de Interceptación:** Si la transacción es anulada debido a cualquier condición de excepción del programa que da como resultado una interceptación (es decir, la condición provoca que la ejecución interpretativa finalice y el control vuelva al programa anfitrión), un TDB es almacenado en una ubicación especificada en el bloque de descripción de estado para el sistema operativo invitado.

El TDB especificado por TBEGIN es almacenado solamente, en una realización, cuando la dirección del TDB es válida (es decir, cuando el campo B₁ de la instrucción de TBEGIN más exterior es distinto de cero).

20 Para anulaciones debidas a condiciones de sesión de programa no filtradas, solamente una o bien del PI-TDB o bien del TDB de Interceptación será almacenado. Así, puede haber cero, uno, o dos TDB almacenados para una anulación.

Otros detalles que se refieren a un ejemplo de cada TDB son descritos a continuación:

25 **TDB especificado por TBEGIN:** La ubicación de 256 bytes es especificada por una dirección de bloque de diagnóstico de transacción válida. Cuando la dirección de bloque de diagnóstico de transacción es válida, el TDB especificado por TBEGIN es almacenado sobre una anulación de transacción. El TDB especificado por TBEGIN es sometido a todos los mecanismos de protección de almacenamiento que están en efecto en la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN más exterior. Un evento de alteración de almacenamiento PER (Grabación de Evento de Programa) para cualquier porción del TDB especificado por TBEGIN es detectado durante la ejecución de la instrucción de TBEGIN más exterior, no durante el tratamiento de anulación de transacción.

30 Un propósito de la PER es ayudar en la depuración de programas. Permite que el programa sea alertado por los siguientes tipos de eventos, como ejemplos:

- Ejecución de una acción de derivación con éxito. Se proporciona la opción de que ocurra un evento solamente cuando la ubicación objetivo de derivación está dentro del área de almacenamiento designada.
- Recuperación de una instrucción desde el área de almacenamiento designada.
- 35 • Alteración de los contenidos del área de almacenamiento designada. Se proporciona la opción de que ocurra un evento solamente cuando el área de almacenamiento está dentro de los espacios de dirección designados.
- Ejecución de una instrucción de ALMACENAR UTILIZANDO UNA DIRECCIÓN REAL
- Ejecución de la instrucción de FINAL DE TRANSACCIÓN.

40 El programa puede especificar de manera selectiva que uno o más de los tipos anteriores de eventos sean reconocidos, excepto que el evento para ALMACENAR UTILIZANDO UNA DIRECCIÓN REAL pueda ser especificado solamente junto con el evento de alteración de almacenamiento. La información relativa a un evento de PER es proporcionada al programa por medio de una interrupción de programa, siendo identificada la causa de interrupción en el código de interrupción.

45 Cuando la dirección de bloque de diagnóstico de transacción no es válida, un TDB especificado por TBEGIN no es almacenado.

50 **TDB de Interrupción de Programa:** Ubicaciones reales 6,144-6,399 (1800-18 FF hex). El TDB de interrupción de programa es almacenado cuando es anulada una transacción debido a la interrupción de programa. Cuando es anulada una transacción debido a otras causas, los contenidos del TDB de interrupción de programa son impredecibles.

El TDB de interrupción de programa no está sujeto a ningún mecanismo de protección. Los eventos de alteración de

almacenamiento de PER no son detectados por el TDB de interrupción de programa cuando es almacenado durante una interrupción de programa.

5 **TDB de Interceptación:** La ubicación real de anfitrión de 256 bytes especificada por las ubicaciones 488-495 de la descripción de estado. El TDB de interceptación es almacenado cuando una transacción anulada da como resultado una interceptación de interrupción de programa invitado (es decir, código de interceptación 8). Cuando se anula una transacción debido otras causas, los contenidos del TDB de interceptación son impredecibles. El TDB de interceptación no está sometido a ningún mecanismo de protección.

Como se ha representado en la fig. 9, los campos de un bloque 900 de diagnóstico de transacción son como sigue, en una realización:

10 Formato 902: el Byte 0 contiene una indicación de validez y formato, como sigue:

Valor	Significado
-------	-------------

0	Los campos restantes del TDB son impredecibles.
---	---

1	Un TDB de formato 1, cuyos campos restantes son descritos posteriormente.
---	---

2-255	Reservado.
-------	------------

15 Un TDB en el cual el campo de formato es cero es denominado como un TDB nulo.

Indicadores 904: El byte 1 contiene distintas indicaciones, como sigue:

20 Validez de Identificador de Conflicto (CTV): Cuando es anulada una transacción debido a un conflicto de recuperación o almacenamiento (es decir, códigos de anulación 9 o 10, respectivamente), el bit 0 del byte 1 es la indicación de validez de identificador de conflicto. Cuando la indicación de CTV es uno, el identificador de conflicto 910 en los bytes 16-23 del TDB contiene la dirección lógica en la que fue detectado el conflicto. Cuando la indicación de CTV es cero, los bytes 16-23 del TDB son impredecibles.

Cuando es anulada una transacción debido a cualquier otra razón distinta a un conflicto de recuperación o almacenamiento, el bit 0 del byte 1 es almacenado como cero.

25 Indicación de Transacción Restringida (CTI): Cuando la CPU está en el modo de ejecución transaccional restringido, el bit 1 del byte 1 es configurado a uno. Cuando la CPU está en el modo de ejecución transaccional no restringido, el bit 1 del byte 1 es configurado a cero.

Reservados: Los bits 2-7 del byte 1 son reservados, y almacenados como ceros.

Profundidad de Anidación de Transacción (TND) 906: Los bytes 6-7 contienen la profundidad de anidación de transacción cuando la transacción fue anulada.

30 Código de Anulación de Transacción (TAC) 908: Los bytes 8-15 contienen un código de anulación de transacción sin signo de 64 bits. Cada punto de código indica una razón por la cual es anulada una transacción.

Es dependiente del modelo si el código de anulación de transacción es almacenado en el TDB de interrupción cuando es anulada una transacción debido a condiciones distintas de una interrupción de programa.

35 Identificador de Conflicto 910: Para transacciones que son anuladas debido a conflicto de recuperación o almacenamiento (es decir, códigos de anulación 9 y 10, respectivamente), los bytes 16-23 contienen la dirección lógica de la ubicación de almacenamiento en la cual fue detectado el conflicto. El identificador de conflicto es significativo cuando el bit de CTV, el bit 0 del byte 1, es uno.

Cuando el bit de CTV es cero, los bytes 16-23 son impredecibles.

40 Debido a la ejecución especulativa mediante la CPU, el identificador de conflicto puede designar una ubicación de almacenamiento que no sería accedida necesariamente por la secuencia de ejecución conceptual de la transacción.

45 Dirección de Instrucción de Transacción Anulada (ATIA) 912: Los bytes 24-31 contienen una dirección de instrucción que identifica la instrucción que se estaba ejecutando cuando fue detectada una anulación. Cuando se anula una transacción debido a los códigos de anulación 2, 5, 6, 11, 13, o 256, o mayor, o cuando se anula una transacción debido a los códigos de anulación 4 o 13 y la condición de excepción del programa es nula, la ATIA apunta directamente a la instrucción que estaba siendo ejecutada. Cuando se anula una transacción debido a los códigos de anulación 4 o 12, y la condición de excepción de programa no es nula, la ATIA apunta más allá de la instrucción que se estaba ejecutando.

Cuando se anula una transacción debido a los códigos de anulación 7-10, 14-16, o 255, la ATIA no indica necesariamente la instrucción exacta que provoca la anulación, pero puede apuntar a una instrucción anterior o

posterior dentro de la transacción.

Si se anula una transacción debido una instrucción que es el objetivo de la instrucción de tipo ejecución, la ATIA identifica la instrucción de tipo ejecución, bien apuntando a la instrucción o más allá de ella, dependiendo del código de anulación como se ha descrito anteriormente. La ATIA no indica el objetivo de instrucción de tipo ejecución.

- 5 La ATIA es sometida al modo de direccionamiento cuando se anula la transacción. En el modo de direccionamiento de 24 bits, los bits 0-40 del campo contienen ceros. En el modo de direccionamiento de 31 bits los bits 0-32 del campo contienen ceros.

10 Es dependiente del modelo si la dirección de instrucción de transacción anulada es almacenada en el TDB de interrupción de programa cuando es anulada una transacción debido a condiciones distintas de una interrupción de programa.

15 Cuando es anulada una transacción debido al código de anulación 4 o 12, y la condición de excepción de programa no es nula, la ATIA no apunta a la instrucción que provoca la anulación. Restando el número de medias palabras indicado por el código de longitud de interrupción (ILC) desde la ATIA, la instrucción que provoca la anulación puede ser identificada en condiciones que se están suprimiendo o terminando, o para eventos no PER que se están completando. Cuando se anula una transacción debido a un evento PER, y no está presente otra condición de excepción de programa, la ATIA es impredecible.

Cuando la dirección de bloque de diagnóstico de transacción es válida, el ILC puede ser examinado en la identificación de sección de programa (PIID) en los bytes 36-39 del TDB especificado por TBEGIN. Cuando no se aplica el filtrado, el ILC puede ser examinado en la PIID en la ubicación 140-143 en el almacenamiento real.

- 20 Identificación de Acceso de Excepción (EAID) 914: Para transacciones que son anuladas debido a ciertas condiciones de excepción de programa filtradas, el byte 32 del TDB especificado por TBEGIN contiene la identificación de acceso de excepción. En un ejemplo de la arquitectura z, el formato de la EAID, y los casos para los cuales es almacenada, son los mismos que los descritos en la ubicación real 160 cuando la condición de excepción da como resultado una interrupción, como se ha descrito en los Principios de Funcionamiento incorporados antes por referencia.
- 25

Para transacciones que son anuladas por otras razones, incluyendo cualesquiera condiciones de excepción que dan como resultado una interrupción de programa, el byte 32 es impredecible. El byte 32 es impredecible en el TDB de interrupción de programa.

- 30 Este campo es almacenado solamente en el TDB designado por la dirección de bloque de diagnóstico de transacción; de lo contrario, el campo es reservado. La EAID es almacenada solamente para la lista de acceso controlada o la protección DAT, tipo ASCE, traducción de página, traducción de primera región, traducción de segunda región, traducción de tercera región, y condiciones de sección de programa de traducción de segmento.

35 Código de Excepción de Datos (DXC) 916: Para transacciones que son anuladas debido a condiciones de excepción de programa de excepción de datos filtrados, el byte 33 del TDB especificado por TBEGIN contiene el código de excepción de datos. En un ejemplo de la arquitectura z, el formato del DXC, y los casos para los cuales es almacenado, son los mismos que los descritos en la ubicación real 147 cuando la condición de excepción da como resultado un interrupción, como se ha descrito en los Principios de Funcionamiento incorporados antes por referencia. En un ejemplo, la ubicación 147 incluye el DXC.

- 40 Para transacciones que son anuladas por otras razones, incluyendo cualesquiera condiciones de excepción que dan como resultado una interrupción de programa, el byte 33 es impredecible. El byte 33 es impredecible en el TDB de interrupción de programa.

Este campo es almacenado solamente en el TDB designado por la dirección de bloque de diagnóstico de transacción; de lo contrario, el campo es reservado. El DXC es almacenado solamente para condiciones de excepción de programa de datos.

- 45 Identificación de Interrupción de Programa (PIID) 918. Para transacciones que son anuladas debido a condiciones de excepción de programa filtrado, los bytes 36-39 del TDB especificado por TBEGIN contienen la identificación de interrupción de programa. En un ejemplo de la arquitectura z, el formato de la PIID es el mismo que el descrito en las ubicaciones reales 140-143 cuando la condición da como resultado una interrupción (como se ha descrito en los Principios de Funcionamiento incorporados antes por referencia), excepto que el código de longitud de instrucción en los bits 13-14 de la PIID es respectivo a la instrucción en la cual fue detectada la condición de excepción.
- 50

Para transacciones que son anuladas por otras razones, incluyendo condiciones de excepción que dan como resultado una interrupción del programa, los bytes 36-39 son impredecibles. Los bytes 36-39 son impredecibles en el TDB de interrupción de programa.

Este campo es almacenado solamente en el TDB designado por la dirección de bloque de diagnóstico de

transacción; de lo contrario, el campo es reservado. La identificación de interrupción de programa es almacenada solamente para condiciones de excepción de programa.

Identificación de Excepción de Traducción (TEID) 920: Para transacciones que son anuladas debido a cualquiera de las siguientes condiciones de excepción de programa filtrado, los bytes 40-47 del TDB especificado por TBEGIN contienen la identificación de excepción de traducción.

- 5 • Lista de acceso controlada o protección DAT
- Tipo ASCE
- Traducción de página
- Traducción de primera región
- 10 • Traducción de segunda región
- Traducción de tercera región
- Excepción de traducción de segmento

En un ejemplo de la Arquitectura z, el formato de la TEID es el mismo que el descrito en las ubicaciones reales 168-175 cuando la condición da como resultado una interrupción, como se ha descrito en los Principios de Funcionamiento incorporados antes por referencia.

Para transacciones que son anuladas por otras razones, incluyendo condiciones de excepción que dan como resultado una interrupción de programa, los bytes 40-47 son impredecibles. Los bytes 40-47 son impredecibles en el TDB de interrupción del programa.

Este campo es almacenado solamente en el TDB designado por la dirección de bloque de diagnóstico de transacción; de lo contrario, el campo es reservado.

Dirección del Evento de Ruptura 922: Para transacciones que son anuladas debido a las condiciones de excepción del programa filtrado, los bytes 48-55 del TBEGIN-TDB especificado contienen la dirección de evento de ruptura. En un ejemplo de la Arquitectura z, el formato de la dirección del evento de ruptura es el mismo que el descrito en las ubicaciones reales 272-279 cuando la condición da como resultado una interrupción, como se ha descrito en los Principios de Funcionamiento incorporados antes por referencia.

Para transacciones que son anuladas por otras razones, incluyendo condiciones de excepción que dan como resultado una interrupción de programa, los bytes 48-55 son impredecibles. Los bytes 48-55 son impredecibles en el TDB de interrupción del programa.

Este campo es almacenado solamente en el TDB designado por la dirección de bloque de diagnóstico de transacción; de lo contrario, el campo es reservado.

Otros detalles con relación a los eventos de ruptura son descritos posteriormente.

En una realización de la Arquitectura z, cuando la instalación PER-3 es instalada, proporciona al programa con la dirección de la última instrucción para provocar una ruptura en la ejecución secuencial de la CPU. La grabación de dirección de evento de ruptura puede ser utilizada como una ayuda de depuración para detección de derivación salvaje. Esta instalación proporciona, por ejemplo, un registro de 64 bits en la CPU, llamado el registro de dirección de evento de ruptura. Cada vez que una instrucción distinta de la ANULACIÓN DE TRANSACCIÓN provoca una ruptura en la ejecución de instrucción secuencial (es decir, la dirección de instrucción en la PSW es reemplazada, en vez de incrementada por la longitud de la instrucción), la dirección de esa instrucción es colocada en el registro de dirección de evento de ruptura. Siempre que ocurre una interrupción de programa, sea indicado o no el PER, los contenidos actuales del registro de dirección de evento de ruptura son colocados en ubicaciones de almacenamiento real 272-279.

Si la instrucción que provoca el evento de ruptura es el objetivo de una instrucción de tipo ejecución (EJECUCIÓN o EJECUCIÓN RELATIVA LARGA), a continuación la dirección de instrucción utilizada para recuperar la instrucción de tipo ejecución es colocada en el registro de dirección de evento de ruptura.

En una realización de la Arquitectura z, se considera que un evento de ruptura ocurre si una de las siguientes instrucciones provoca la derivación : DERIVACIÓN Y VÍNCULO (BAL, BALR); DERIVACIÓN Y GUARDADO (BAS, BASR); DERIVACIÓN Y GUARDADO Y CONFIGURACIÓN DE MODO (BASSM); DERIVACIÓN Y CONFIGURACIÓN DE MODO (BSM); DERIVACIÓN Y APILAMIENTO (BAKR); DERIVACIÓN EN CONDICIÓN (BC, BCR); DERIVACIÓN EN CUENTA (BCT, BCTR, BCTG, BCTGR); DERIVACIÓN EN ÍNDICE ALTO (BXH, BXHG); DERIVACIÓN EN ÍNDICE BAJO O IGUAL (BXLE, BXLEG); DERIVACIÓN RELATIVA EN CONDICIÓN (BRC); DERIVACIÓN RELATIVA EN CONDICIÓN LARGA (BRCL); DERIVACIÓN RELATIVA EN CUENTA (BRCT,

BRCTG); DERIVACIÓN RELATIVA EN ÍNDICE ALTO (BRXH, BRXHG); DERIVACIÓN RELATIVA EN ÍNDICE BAJO O IGUAL (BRXLE, BRXLG); COMPARACIÓN Y DERIVACIÓN (CRB, CGRB); COMPARACIÓN Y DERIVACIÓN RELATIVA (CRJ, CGRJ); COMPARACIÓN INMEDIATA Y DERIVACIÓN (CIB, CGIB); COMPARACIÓN INMEDIATA Y DERIVACIÓN RELATIVA (CIJ, CGIJ); COMPARACIÓN LÓGICA Y DERIVACIÓN (CLRB, CLGRB);
 5 COMPARACIÓN LÓGICA Y DERIVACIÓN RELATIVA (CLRJ, CLGRJ); COMPARACIÓN LÓGICA INMEDIATA Y DERIVACIÓN (CLIB, CLGIB); Y COMPARACIÓN LÓGICA INMEDIATA Y DERIVACIÓN RELATIVA (CLIJ, CLGIJ).

Se considera también que ocurre un evento de ruptura si se completa una de las siguientes instrucciones: DERIVACIÓN Y CONFIGURACIÓN DE AUTORIDAD (BSA); DERIVACIÓN EN GRUPO DE SUBESPACIO (BSG);
 10 DERIVACIÓN RELATIVA Y GUARDADO (BRAS); DERIVACIÓN RELATIVA Y GUARDADO LARGO (BRASL); PSW DE CARGA (LPSW); PSW DE CARGA EXTENDIDA (LPSWE); LLAMADA DE PROGRAMA (PC); RETORNO DE PROGRAMA (PR); TRANSFERENCIA DE PROGRAMA (PT); TRANSFERENCIA DE PROGRAMA CON INSTANCIA (PTI); PROGRAMA DE RESUMEN (RP); Y TRAP (TRAP2, TRAP4).

No se considera que ocurre un evento de ruptura como resultado de una transacción que es anulada (bien de manera implícita o bien como resultado de la instrucción de ANULACIÓN DE TRANSACCIÓN).

15 Información de Diagnóstico Dependiente de Modelo 924: Los bytes 112-127 contienen la información del diagnóstico dependiente de modelo.

Para todos los códigos de anulación excepto el 12 (interrupción de programa filtrado), la información de diagnóstico dependiente de modelo es guardada en cada TDB que es almacenado.

En una realización, la información de diagnóstico dependiente de modelo incluye lo siguiente:

- 20 • Los bytes 112-119 contienen un vector de 64 bits llamado las indicaciones de derivación de ejecución transaccional (TXBI). Cada uno de los primeros 63 bits del vector indica el resultado de la ejecución de una instrucción de derivación mientras la CPU estaba en el modo de ejecución transaccional, como sigue:

Valor	Significado
-------	-------------

0	La instrucción completada sin derivación.
---	---

1	la instrucción completada con derivación.
---	---

25

El bit 0 representa el resultado de dicha primera instrucción de derivación, el bit 1 representa el resultado de dicha segunda instrucción, y así sucesivamente.

Si se ejecutaron menos de 63 instrucciones de derivación mientras la CPU estaba en el modo de ejecución transaccional, los bits más a la derecha que no corresponden con las instrucciones de derivación son configurados a ceros (incluyendo el bit 63). Cuando se ejecutaron más de 63 instrucciones de derivación el bit 63 de la TXBI es configurado a uno.
 30

Los bits en la TXBI son configurados por instrucciones que son capaces de provocar un evento de ruptura, como se ha recogido antes, excepto por lo siguiente:

- Cualquier instrucción restringida no hace que un bit sea configurado en la TXBI.
- 35 - Para instrucciones de, por ejemplo, la Arquitectura z, cuando el campo M_1 de la instrucción de DERIVACIÓN EN CONDICIÓN, DERIVACIÓN RELATIVA EN CONDICIÓN, O DERIVACIÓN RELATIVA EN CONDICIÓN LARGA es cero, o cuando el campo R_2 de la siguiente instrucción es cero, es dependiente del modelo si la ejecución de la instrucción hace que un bit sea configurado en la TXBI.

- 40 • DERIVACIÓN Y VÍNCULO (BALR); DERIVACIÓN Y GUARDADO (BASR); DERIVACIÓN Y GUARDADO Y CONFIGURACIÓN DE MODO (BASSM); DERIVACIÓN Y CONFIGURACIÓN DE MODO (BSM); DERIVACIÓN EN CONDICIÓN (BCR); y DERIVACIÓN EN CUENTA (BCTR, BCTGR)

- 45 • Para condiciones de anulación que fueron causadas por una excepción de acceso de anfitrión, la posición de bit 0 del byte 127 es configurada a uno. Para todas las otras condiciones de anulación, la posición del bit 0 del byte 127 es configurada a cero.

- Para condiciones de anulación que fueron detectadas por la unidad de carga/almacenamiento (LSU), los cinco bits más a la derecha del byte 127 contienen una indicación de la causa. Para condiciones de anulación que no fueron detectadas por la LSU, el byte 127 es reservado.

50 Registros Generales 930: Los bytes 128-255 contienen los contenidos de los registros generales 0-15 en el momento en que fue anulada la transacción. Los registros son almacenados en orden ascendente, comenzando con el registro general 0 en los bytes 128-135, el registro general 1 en los bytes 136-143, y así sucesivamente.

Reservado: Todos los demás campos son reservados. A menos que se indique lo contrario, los contenidos de los campos reservados son impredecibles.

5 Cuando es observado por otras CPU y el subsistema I/O, el almacenamiento de los TDB durante una anulación de transacción es una referencia de acceso múltiple que ocurre después de cualesquiera almacenamientos no transaccionales.

Una transacción puede ser anulada debido a causas que están fuera del alcance de la configuración inmediata en la cual se ejecuta. Por ejemplo, eventos transitorios reconocidos por un hipervisor (tal como LPAR o z/VM) puede provocar que se anule una transacción.

10 La información proporcionada en el bloque de diagnóstico de transacción está destinada con propósitos de diagnóstico y es sustancialmente correcta. Sin embargo, ya que una anulación puede haber sido causada por un evento fuera del alcance de la configuración inmediata, la información tal como el código de anulación o identificación de interrupción de programa puede no reflejar con exactitud las condiciones dentro de la configuración, y así, no debería utilizarse en determinar la acción de programa.

15 Además de la información de diagnóstico guardada en el TDB, cuando se anula una transacción debido a cualquier condición de excepción de programa de excepción de datos y tanto el control de registro AFP, bit 45 del registro de control 0, como el control efectivo de operación de coma flotante (F) son uno, el código de excepción de datos (DXC) es colocado en el byte 2 del registro de control de coma flotante (FPCR), independientemente de si el filtrado se aplica a la condición de excepción de programa. Cuando se anula una transacción, y o bien tanto el registro de control AFP o el control efectivo para permitir la operación de coma flotante son cero, el DXC no es colocado en el FPCR.

20 En una realización, como se ha indicado en este documento, cuando la instalación de ejecución transaccional está instalada, se proporcionan las siguientes instrucciones generales.

- EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIÓN
- ALMACENAMIENTO NO TRANSACCIONAL
- 25 • ANULACIÓN DE TRANSACCIÓN
- COMIENZO DE TRANSACCIÓN
- FINAL DE TRANSACCIÓN

Cuando la CPU está en el modo de ejecución transaccional, el intento de ejecución de ciertas instrucciones es restringido y provoca que sea anulada la transacción.

30 Cuando se emite en el modo de ejecución transaccional restringido, el intento de ejecución de instrucciones restringidas puede dar como resultado también una interrupción de programa de restricción de transacción, o puede dar como resultado que la ejecución prosiga como si la transacción no estuviese restringida.

35 En un ejemplo de la Arquitectura z, las instrucciones restringidas incluyen, como ejemplos, las siguientes instrucciones no privilegiadas de: COMPARACIÓN E INTERCAMBIO Y ALMACENAMIENTO; MODIFICACIÓN DE CONTROLES DE INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN; REALIZACIÓN DE OPERACIÓN BLOQUEADA; RECUPERACIÓN PREVIAMENTE DE DATOS (RELATIVOS LARGOS), cuando el código en el campo M₁ es 6 o 7; ALMACENAMIENTO DE CARACTERES DEBAJO DE LA MÁSCARA ALTA, cuando el campo M₃ es cero y el código en el campo R₁ es 6 o 7; ALMACENAMIENTO DE LISTA DE INSTALACIONES EXTENDIDA; ALMACENAMIENTO DE CONTROLES DE INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN; LLAMADA DE SUPERVISOR ; y PRUEBA DE CONTROLES DE INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN.

40 En la lista anterior, COMPARACIÓN E INTERCAMBIO Y ALMACENAMIENTO y REALIZACIÓN DE OPERACIÓN BLOQUEADA son instrucciones complejas que pueden ser implementadas de manera más eficiente por el uso de instrucciones básicas en el modo TX. Los casos para RECUPERACIÓN PREVIAMENTE DE DATOS y RECUPERACIÓN PREVIAMENTE DE DATOS RELATIVOS LARGOS son restringidos cuando los códigos de 6 y 7 liberan una línea de memoria caché, necesitando la consignación de los datos potencialmente antes de la finalización de una transacción. La LLAMADA DE SUPERVISOR está restringida cuando causa una interrupción (que causa que sea anulada una transacción).

Bajo las condiciones listadas a continuación, las siguientes instrucciones están restringidas:

- 50 • DERIVACIÓN Y VINCULO (BALR), DERIVACIÓN Y GUARDADO (BASR), Y DERIVACIÓN Y GUARDADO Y CONFIGURACIÓN DE MODO, cuando el campo R₂ de la instrucción es distinto de cero y el rastreo de derivación es habilitado.

- DERIVACIÓN Y GUARDADO Y CONFIGURACIÓN DE MODO y DERIVACIÓN Y CONFIGURACIÓN DE MODO, cuando el campo R₂ es distinto de cero y es habilitado el modo de rastreo; CONFIGURACIÓN DE MODO DE DIRECCIONAMIENTO, cuando es habilitado el modo de rastreo.
- LLAMADA DE SUPERVISOR, cuando es reconocida una condición de evento de vigilancia.

5 La lista anterior incluye instrucciones que pueden formar entradas de rastreo. Si se permitiera que estas instrucciones se ejecutaran de manera transaccional y formarían las entradas de rastreo, y la transacción posteriormente anulada, el puntero de tabla de rastreo en el registro control 12 sería hecho avanzar, pero los almacenamientos a la tabla de rastreo serían descartados. Esto dejaría un hueco inconsistente en la tabla de rastreo; así, las instrucciones son restringidas en los casos donde formarían entradas de rastreo.

10 Cuando la CPU está en el modo de ejecución transaccional, es dependiente del modelo si son restringidas las siguientes instrucciones: CIFRAR MENSAJE; CIFRAR MENSAJE CON CFB; CIFRAR MENSAJE CON ENCADENAMIENTO; CIFRAR MENSAJE CON CONTADOR; CIFRAR MENSAJE CON OFB; LLAMADA DE COMPRESIÓN; CALCULAR RESUMEN DE MENSAJE INTERMEDIO; CALCULAR RESUMEN DEL ÚLTIMO MENSAJE; CALCULAR CÓDIGO DE AUTENTIFICACIÓN DE MENSAJE; CONVERTIR UNICODE 16 A UNICODE 32; CONVERTIR UNICODE 16 A UNICODE 8; CONVERTIR UNICODE 32 A UNICODE 16; CONVERTIR UNICODE 32 A UNICODE 8; CONVERTIR UNICODE 8 A UNICODE 16; CONVERTIR UNICODE 8 A UNICODE 32; REALIZAR CÁLCULOS CRIPTOGRÁFICOS; DESACTIVAR INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN; y ACTIVAR INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN.

20 Cada una de las instrucciones anteriores es o bien implementada actualmente por el coprocesador de hardware, o bien lo ha sido en máquinas anteriores, y así, es considerada restringida.

Cuando el control efectivo de modificación (A) del AR es cero, las siguientes instrucciones son restringidas: COPIAR DIRECCIÓN, CARGAR MÚLTIPLE ACCESO; CARGAR DIRECCIÓN EXTENDIDA; y CONFIGURAR ACCESO.

25 Cada una de las instrucciones anteriores provoca que se modifiquen los contenidos de un registro de acceso. Si el control A en la instrucción de COMIENZO DE TRANSACCIÓN es cero, entonces el programa ha indicado de manera explícita que no se debe permitir la modificación del registro de acceso.

Cuando el control efectivo para permitir la operación de coma flotante (F) es cero, las instrucciones de coma flotante son restringidas.

30 Bajo ciertas circunstancias, las siguientes instrucciones pueden ser restringidas: EXTRACCIÓN DE TIEMPO DE CPU; EXTRACCIÓN DE PSW; ALMACENAMIENTO DE RELOJ; ALMACENAMIENTO DE RELOJ EXTENDIDO; y ALMACENAMIENTO DE RELOJ RÁPIDO.

Cada una de las instrucciones anteriores es sometida a un control de interceptación en la descripción de Estado de ejecución interpretativa. Si el hipervisor ha configurado el control de interceptación para estas instrucciones, entonces su ejecución puede prolongarse debido a la implementación del hipervisor; así, son consideradas restringidas si ocurre una interceptación.

35 Cuando una transacción no restringida es anulada debido al intento de ejecución de una instrucción restringida, el código de anulación de transacción en el bloque de diagnóstico de transacción es configurado a 11 (instrucción restringida), y el código de condición es configurado a 3, excepto como sigue: cuando una transacción no restringida es anulada debido al intento de ejecución de una instrucción que de lo contrario daría como resultado una excepción de operación privilegiada, es impredecible si el código de anulación es configurado a 11 (instrucción restringida) o a 4 (interrupción de programas sin filtrar que resulta del reconocimiento de la interrupción de programa de operación privilegiada). Cuando se anula una transacción NO restringida debido al intento de ejecución de RECUPERACIÓN PREVIA DE DATOS (RELATIVOS LARGOS) cuando el código en el campo M₁ es 6 o 7 o ALMACENAMIENTO DE CARACTERES BAJO MÁSCARA ALTA cuando el campo M₃ es cero y el código en el campo R₁ es 6 o 7, es impredecible si el código de anulación es configurado a 11 (instrucción restringida) o a 16 (otra memoria caché).

40

45 Cuando se anula una transacción no restringida debido al intento de ejecución de LLAMADA DE VIGILANCIA, y tanto una condición de evento de vigilancia como una condición de excepción de especificación están presentes es impredecible si el código de anulación es configurado a 11 o a 4, o, si en la interrupción de programa es filtrada, 12.

50 Se pueden restringir instrucciones adicionales en una transacción restringida. Aunque estas instrucciones no son actualmente definidas para ser restringidas en una transacción no restringida, pueden ser restringidas bajo ciertas circunstancias en una transacción no restringida en procesos futuros.

Ciertas instrucciones restringidas pueden ser permitidas en el modo de ejecución transaccional en procesos futuros. Por lo tanto, el programa no debería basarse en la transacción que es anulada debido al intento de ejecución de una instrucción restringida. La instrucción de ANULACIÓN DE TRANSACCIÓN debería ser utilizada para provocar que se anule una transacción de manera fiable.

55 En una transacción no restringida, el programa debería proporcionar un trayecto de código no transaccional

alternativo para acomodar una transacción que se anula debido a una instrucción restringida.

En funcionamiento, cuando la profundidad de anidación de transacción es cero, la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN (TBEGIN) que da como resultado la condición de código cero hace que la CPU entre en el modo de ejecución transaccional no restringido. Cuando la profundidad de anidación de transacción es cero, la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN restringida (TBEGINC) que da como resultado la condición de código cero hace que la CPU entre en el modo de ejecución transaccional restringido.

Excepto donde explícitamente se observe de otra manera, todas las reglas que se aplican para la ejecución no transaccional se aplican también para la ejecución transaccional. Más adelante están las características adicionales de tratamiento mientras la CPU está en el modo de ejecución transaccional.

Cuando la CPU está en el modo de ejecución transaccional no restringido, la ejecución de la instrucción de COMIENZO DE TRANSACCIÓN que da como resultado la condición de código cero hace que la CPU permanezca en el modo de ejecución transaccional no restringido.

Cuando son observados por la CPU, recuperaciones y almacenamientos hechos en el modo de ejecución transaccional no son diferentes de los hechos mientras no está en el modo de ejecución transaccional. Cuando es observado por otras CPU y por el subsistema I/O, todos los accesos de operando de almacenamiento hechos mientras una CPU está en el modo de ejecución transaccional parecen ser un acceso concurrente de un solo bloque. Es decir, los accesos a todos los bytes dentro de una media palabra, palabra, doble palabra, o cuádruple palabra son especificados para parecer ser el bloque concurrente cuando es observado por otras CPU y los programas de I/O (por ejemplo, canal). La media palabra, palabra, doble palabra, o cuádruple palabra se refiere en esta sección como un bloque. Cuando se especifica que una referencia de tipo de recuperación parece ser concurrente dentro de un bloque, no es permitido el acceso de almacenamiento al bloque por otras CPU o programa I/O durante el tiempo en el que están siendo recuperados los bytes contenidos en el bloque. Cuando se especifica que una referencia de tipo almacenamiento parece ser concurrente dentro de un bloque, no es permitido el acceso al bloque, a la recuperación o al almacenamiento, por otras CPU o programa I/O durante el tiempo en el que están siendo almacenados los bytes dentro del bloque.

Los accesos al almacenamiento para la instrucción y las recuperaciones de tabla DAT y ART (Tabla de Registro de Acceso) siguen las reglas no transaccionales.

La CPU deja el modo de ejecución transaccional normalmente por medio de una instrucción de FINAL DE TRANSACCIÓN que causa la transición de la profundidad de anidación de transacción a cero, en cuyo caso, finaliza la transacción.

Cuando la CPU deja el modo de ejecución transaccional por medio de la finalización de una instrucción de FINAL DE TRANSACCIÓN, todos los almacenamientos hechos mientras está en el modo de ejecución transaccional son consignados; es decir, los almacenamientos parece que ocurren como una única operación de bloque concurrente cuando es observado por otras CPU y por el subsistema I/O.

Otros detalles con relación al tratamiento asociado con las transacciones restringidas y no restringidas, que incluye el tratamiento asociado con iniciar una transacción mediante TBEGINC y TBEGIN, respectivamente son proporcionados a continuación.

En un ejemplo, el tratamiento asociado con iniciar una transacción restringida mediante, por ejemplo, TBEGINC es, como sigue:

- Si ya está en el modo TX restringido, se reconoce una excepción de transacción restringida.
- Si la TND actual (Profundidad de Anidación de Transacción) > 0 la ejecución prosigue como si fuese una transacción no restringida
 - El control efectivo F se configura a cero
 - El PIFC efectivo no es cambiado
 - permite a la TX no restringida más exterior llamar a la función de servicio que puede o no puede utilizar la TX restringida.
- Si la TND actual = 0:
 - La dirección del bloque de diagnóstico de transacción es inválida
 - No es almacenada en anulación el TDB especificado por la instrucción
 - Configurar PSW de anulación de transacción a la dirección de TBEGINC

- No a la siguiente instrucción secuencial
 - Parejas de registro general designadas por la GRSM guardada en una ubicación dependiente del modelo no accesible por el programa
 - Identificador de transacción opcionalmente formado (a partir del operando D_2). El identificador de transacción es un identificador de la transacción. Puede ser igual a la dirección de operando de almacenamiento u otro valor
- 5
- A efectivo = A TBEGINC y cualquier A exterior
 - TND aumentado
 - Si las transiciones TND desde 0 a 1, la CPU entra en el modo TX restringido
- 10
- De lo contrario, la CPU permanece en el modo TX no restringido
 - La opción finaliza con CC0
 - Excepciones
 - Excepción de especificación (PIC (Código de Interrupción del Programa) 0006) si el campo B_1 es distinto de cero
- 15
- Excepción de operación especial (PIC 0013 hex) si el control de ejecución de transacción (CR0.8) es cero
 - Excepción de restricción de transacción (PIC 0018 hex) si es emitida en el modo TX restringido
 - Excepción de operación (PIC 0001) si la instalación de ejecución transaccional restringida no está instalada
- 20
- Excepción de ejecución (PIC 0003) si la instrucción es el objetivo de una instrucción de tipo ejecución
 - Código de anulación 13 si es excedida la profundidad de anidación
 - Condiciones de anulación en transacción restringida
 - La PSW de anulación apunta a la instrucción de TBEGINC
- 25
- No a la siguiente instrucción
 - La condición de anulación provoca que toda la TX sea accionada de nuevo
 - * Sin trayecto de fallo
 - La CPU toma medidas especiales para asegurar la finalización con éxito sobre el accionamiento de nuevo
- 30
- Asumiendo que no hay conflicto, interrupción, o violación restringida persistente, se garantiza la finalización eventual de la transacción.
 - Violación de restricción:
 - PIC 0018 hex - indica violación de restricción de transacción
 - O, la transacción se ejecuta como si fuese no restringida
- 35
- Además, en una realización, el tratamiento asociado con iniciar una transacción no restringida mediante, por ejemplo, TBEGIN es, como sigue:
- Si TND = 0:
 - Si $B_1 \neq 0$, la dirección del bloque de diagnóstico de transacción se configura desde la dirección del primer operando.
- 40
- La PSW de anulación de transacción se configura a la siguiente dirección de instrucción secuencial

ES 2 620 704 T3

- Las parejas de registro general designados por el campo I_2 son guardados en la ubicación dependiente del modelo.
 - No accesible directamente por el programa
- Los controles PIFC, A, & F efectivos calculados
- 5
 - A efectivo = TBEGIN A y cualquier A exterior
 - F efectivo = TBEGIN F y cualquier F exterior
 - PIFC efectivo = $\max(\text{TBEGIN PIFC}, \text{cualquier PIFC exterior})$
- Profundidad de anidación de transacción (TND) incrementada
- Si la TND cambia desde 0 a 1, la CPU entra en el modo de ejecución transaccional
- 10
 - Código de condición configurado a cero
 - Cuando la instrucción que sigue a TBEGIN recibe el control:
 - TBEGIN indicada con éxito por CC0
 - Transacción anulada indicada por CC distinto de cero
 - Excepciones:
- 15
 - Código de anulación 13 si es excedida la profundidad de anidación
 - Excepción de acceso (uno de distintos PIC) si el campo B_1 es distinto de cero, y el operando de almacenamiento no puede ser accedido durante una operación de almacenamiento
 - Excepción de ejecución (PIC 0003) si la instrucción de TBEGIN es el objetivo de una instrucción de tipo ejecución
- 20
 - Excepción de operación (PIC 0001) si la instalación de ejecución transaccional no está instalada
 - PIC 0006 si bien
 - PIFC es inválido (valor de 3)
 - La dirección del segundo operando no está alineada con la doble palabra
 - PIC 0013 hex si el control de ejecución transaccional (CR0.8) es cero
- 25
 - PIC 0018 hex si es emitido en el modo TX restringido

Una transacción puede ser anulada de manera implícita por distintas causas, o puede ser anulada de manera explícita por la instrucción de ANULACIÓN DE TRANSACCIÓN. Ejemplos de causas posibles de una anulación de transacción, el código de anulación correspondiente, y el código de condición que es colocado en la PSW de anulación de transacción son descritos posteriormente.

- 30 Interrupción Externa: El código de anulación de transacción es configurado a 2, y el código de condición en la PWS de anulación de transacción es configurado a 2. La PSW de anulación de transacción es almacenada como la PSW antigua externa como una parte del tratamiento de interrupción externa.

- 35 Interrupción de Programa (Sin filtrado): Una condición de excepción de programa que da como resultado una interrupción (es decir, una condición sin filtrar) provoca que la transacción sea anulada con el código 4. El código de condición en la PSW de anulación de transacción es configurado específico al código de interrupción de programa. La PSW de anulación de transacción es almacenada como la PSW antigua de programa como una parte de tratamiento de interrupción de programa.

- 40 Una instrucción que de lo contrario daría como resultado una transacción que es anulada debido a una excepción de operación puede dar resultados alternativos: para una transacción no restringida, la transacción puede en vez de ser anulada con el código de anulación 11 (instrucción restringida); para una transacción restringida, puede ser reconocida una interrupción de programa de restricción de transacción en vez de la excepción de operación.

Cuando un evento de PER (Grabación de Evento de Programa) es reconocido en unión con cualquier otra condición de sesión de programa sin filtrar, el código de condición es configurado a 3.

Interrupción de Comprobación de Máquina: El código de anulación de transacción es configurado a 5, y el código de

- condición en la PSW de anulación de transacción es configurado a 2. La PSW de anulación de transacción es almacenada como la PSW antigua de comprobación de máquina como una parte del tratamiento de interrupción de comprobación de máquina.
- 5 Interrupción de I/O: El código de anulación de transacción es configurado a 6, y el código de condición en la PSW de anulación de transacción es configurado a 2. La PSW de anulación de transacción es almacenada como la PSW antigua de I/O como una parte del tratamiento de interrupción de I/O.
- Desbordamiento de Recuperación: Una condición de desbordamiento de recuperación es detectada cuando la transacción intenta recuperar desde más ubicaciones que los soportes de la CPU. El código de anulación de transacción es configurado a 7, y el código de condición es configurado bien a 2 o bien a 3.
- 10 Desbordamiento de almacenamiento: Una condición de desbordamiento de almacenamiento es detectada cuando la transacción intenta almacenar a más ubicaciones que los soportes de la CPU. El código de anulación de transacción es configurado a 8, y el código de condición es configurado bien a 2 o bien a 3.
- 15 Permitir que el código de condición sea bien 2 o bien 3 en respuesta a una anulación de desbordamiento de recuperación o almacenamiento permite que la CPU indique situaciones que potencialmente se vuelven a intentar (por ejemplo, el código de condición 2 indica que la nueva ejecución de la transacción puede ser productiva; mientras el código de condición 3 no recomienda la nueva ejecución).
- Conflicto de Recuperación: Una condición de conflicto de recuperación es detectada cuando otra CPU o el subsistema I/O intenta almacenar en una ubicación que ha sido recuperada de manera transaccional mediante ésta CPU. El código de anulación de transacción es configurado a 9, y el código de condición es configurado a 2.
- 20 Conflicto de Almacenamiento: Una condición de conflicto de almacenamiento es detectada cuando otra CPU o el subsistema I/O intenta acceder a una ubicación que ha sido recuperada de manera transaccional mediante ésta CPU. El código de anulación de transacción es configurado a 10, y el código de condición es configurado a 2.
- Instrucción Restringida: Cuando la CPU está en el modo de ejecución transaccional, el intento de ejecución de una instrucción restringida provoca que la transacción sea anulada. El código de anulación de transacción es configurado a 11, y el código de condición es configurado a 3.
- 25 Cuando la CPU está en el modo de ejecución transaccional restringido, es impredecible si el intento de ejecución de una instrucción restringida da como resultado una interrupción de programa de restricción de transacción o una anulación debido a una instrucción restringida. La transacción es aún anulada pero el código de anulación puede indicar cualquier causa.
- 30 Condición de Excepción de Programa (Filtrado): Una condición de excepción de programa que no da como resultado una interrupción (es decir, una condición filtrada) provoca que la transacción sea anulada con un código de anulación de transacción de 12. El código de condición es configurado a 3.
- 35 Profundidad de Anidación Excedida: La condición de profundidad de anidación excedida es detectada cuando la profundidad de anidación de transacción está en el valor máximo permisible para la configuración, y es ejecutada una instrucción de COMIENZO DE TRANSACCIÓN. La transacción es anulada con un código de anulación de transacción de 13, y el código de condición es configurado a 3.
- 40 Condición Relacionada de Recuperación de Memoria Caché: Una condición relacionada con ubicaciones de almacenamiento recuperadas por la transacción es detectada por la circuitería de memoria caché de la CPU. La transacción es anulada con un código de anulación de transacción de 14, y el código de condición es configurado bien a 2 o bien a 3.
- Condición Relacionada de Almacenamiento de Memoria Caché: Una condición relacionada con las ubicaciones de almacenamiento almacenadas por la transacción es detectada por la circuitería de memoria caché de la CPU. La transacción es anulada con un código de anulación de transacción de 15, y el código de condición es configurado bien a 2 o bien a 3.
- 45 Condición de Otra Memoria Caché: Una condición de otra memoria caché es detectada por la circuitería de memoria caché de la CPU. La transacción es anulada con un código de anulación de transacción de 16, y el código de condición es configurado bien a 2 o bien a 3.
- 50 Durante la ejecución transaccional, si la CPU accede a las instrucciones u operandos de almacenamiento utilizando diferentes direcciones lógicas que son hechas corresponder a la misma dirección absoluta, depende del modelo si la transacción es anulada. Si la transacción es anulada debido a accesos que utilizan diferentes direcciones lógicas hechas corresponder a la misma dirección absoluta, el código de anulación 14, 15, o 16 es configurado, dependiendo de la condición.
- Condición Miscelánea: Una condición miscelánea es cualquier otra condición reconocida por la CPU que causa que se anule la transacción. El código de anulación de transacción es configurado a 255, y el código de condición es

configurado bien a 2 o bien a 3.

Cuando se ejecutan múltiples configuraciones en la misma máquina (por ejemplo, particiones lógicas o máquinas virtuales), una transacción puede ser anulada debido a una comprobación de máquina externa o interrupción I/O que ha ocurrido en una configuración diferente.

- 5 Aunque los ejemplos son proporcionados anteriormente, pueden proporcionarse otras causas de una anulación de transacción con códigos de anulación y códigos de condición correspondientes. Por ejemplo, una causa puede ser una Interrupción de Reinicio, en que el código de anulación de transacción es configurado a 1, y el código de condición en la PSW de anulación de transacción es configurado a 2. La PSW de anulación de transacción es almacenada como la PSW antigua de reinicio como una parte del tratamiento de reinicio. Como otro ejemplo, una causa puede ser una condición de Llamada de Supervisor, en la cual el código de anulación es configurado a 3, y el código de condición en la PSW de anulación de transacción es configurado a 3. Son también posibles otros o diferentes ejemplos.

Observaciones:

1. La condición miscelánea puede resultar a partir de cualquiera de las siguientes
 - 15
 - Instrucciones, tales como, en la Arquitectura z, de COMPARACIÓN Y REEMPLAZAMIENTO DE ENTRADAS DE TABLA DAT, COMPARACIÓN Y CAMBIO Y PURGADO, INVALIDACIÓN DE ENTRADAS DE TABLA DAT, INVALIDACIÓN DE ENTRADAS DE TABLA DE PÁGINAS, REALIZACIÓN DE GESTIÓN DE MARCO, en que el control NQ es cero y el control SK es uno, CONFIGURACIÓN DE CLAVE DE ALMACENAMIENTO EXTENDIDA en la cual el control NQ es cero, realizadas por otra CPU en la configuración; el código de condición es configurado a 2.
 - 20
 - Una función de operador, tal como reinicio, vuelta a poner en marcha o parada, o la orden de PROCESADOR DE SEÑAL equivalente es realizada en la CPU
 - Cualquier otra condición no enumerada anteriormente; el código de condición es configurado a 2 o a 3.
 - 25
 2. La ubicación en la cual los conflictos de recuperación y almacenamiento son detectados puede ser en cualquier sitio dentro de la misma línea de memoria caché.
 3. Bajo ciertas condiciones, la CPU puede no ser capaz de distinguir entre condiciones de anulación similares por ejemplo, un desbordamiento de recuperación o almacenamiento puede ser indistinguible de un conflicto de recuperación o almacenamiento respectivo.
 - 30
 4. La ejecución especulativa de múltiples trayectos de instrucciones por la CPU puede dar como resultado una transacción que es anulada debido al conflicto o condiciones de desbordamiento, incluso si tales condiciones no ocurren en la secuencia conceptual. Mientras en el modo de ejecución transaccional restringido, la CPU puede inhibir temporalmente la ejecución especulativa, permitiendo a la transacción intentar finalizar sin detectar tales conflictos o desbordamientos de manera especulativa.
 - 35 Cuando se aplican múltiples condiciones de anulación, es impredecible qué código de anulación es reportado por la CPU

La fig. 10 resume códigos de anulación ejemplares almacenados en un bloque de diagnóstico de transacción, y el código de condición correspondiente (CC). La descripción en la fig. 10 ilustra una implementación particular. Son posibles otras implementaciones y codificaciones de los valores. En una realización, los códigos de anulación 1, 3, y 17-254 son reservados para mejoras potenciales. Tales mejoras pueden dar como resultado la configuración bien del código de condición 2 o bien el 3. Además, el código de anulación 0 es reservado y no será asignado a una indicación de anulación significativa, en esta realización. Dependiendo del modelo, la CPU no puede ser capaz de distinguir entre ciertas razones de anulación. Por ejemplo, un desbordamiento de recuperación/almacenamiento y un conflicto de recuperación/almacenamiento pueden no ser distinguibles por la CPU en todas las circunstancias.

- 45 Accesos de conflictos por otras CPU o el subsistema I/O para almacenar ubicaciones que son accedidos de manera transaccional por una CPU puede provocar que sea anulada una transacción. Cuando es anulada una transacción, los almacenamientos transaccionales realizados por la transacción son descartados. Esto puede provocar una depuración del código transaccional extremadamente difícil. Así, en una realización, es proporcionada una instrucción de ALMACENAMIENTO NO TRANSACCIONAL (NTSTG) que permite que almacenamientos hechos en el modo transaccional sean retenidos, incluso si es anulada la transacción.

Como se ha descrito anteriormente con referencia a la fig. 7, en un ejemplo, la instrucción 700 de ALMACENAMIENTO NO TRANSACCIONAL incluye una pluralidad de campos 702a, 702b de código de operación que especifican un código de operación que designa una operación de almacenamiento no transaccional; un campo 704 de registro (R₁) que especifica un registro, cuyos contenidos son llamados el primer operando; un campo 706 de

5 índice (X_2) ; un campo 708 de base (B_2) ; un primer campo 710 de desbordamiento (DL_2) ; y un segundo campo 712 de desbordamiento (DH_2). Los contenidos de los registros generales designados por los campos X_2 y B_2 son añadidos a un valor definido por una concatenación de contenidos de los campos DH_2 y DL_2 para formar la dirección del segundo operando. Cuando cualquiera o ambos campos X_2 o B_2 son cero, el registro correspondiente no toma parte en la suma.

El primer operando de 64 bits es colocado de manera no transaccional sin cambios en la ubicación del segundo operando. En una realización, el primer operando es especificado por el usuario.

El desplazamiento, formado por la concatenación de los contenidos de los campos DH_2 y DL_2 , es tratado como un número entero binario con signo de 20 bits.

10 El segundo operando ha de ser alineado en un límite de doble palabra y, de lo contrario una excepción de especificación (Código de Interrupción del Programa (PIC) 0006) es reconocida y la operación es suprimida.

Código de Condición: El código permanece sin cambios.

Excepciones de Programa:

- Acceso (almacenamiento, segundo operando)
- 15 • Operación (instalación de ejecución transaccional no instalada)
- Especificación
- Restricción de transacción (debido a instrucción restringida); PIC 0018 hex

Además, en una realización:

20 El ALMACENAMIENTO NO TRANSACCIONAL proporciona un medio por el cual un programa que se ejecuta en el modo de ejecución transaccional no restringido puede realizar almacenamientos que serán retenidos, incluso si se anula la transacción. Esto puede facilitar la depuración de la transacción si es anulada.

25 Cuando la CPU no está en el modo de ejecución transaccional, la operación del ALMACENAMIENTO NO TRANSACCIONAL es la misma que una instrucción de almacenamiento, tal como ALMACENAMIENTO (STG) de la Arquitectura z, excepto que, en una realización, el segundo operando del ALMACENAMIENTO NO TRANSACCIONAL ha de estar en un límite de doble palabra (independientemente del modo de ejecución transaccional), mientras que el segundo operando del STG no tiene requisitos de alineación. Así, en esta situación, no es proporcionada una excepción si el procesador no está en el modo de ejecución transaccional, pero en su lugar, el ALMACENAMIENTO NO TRANSACCIONAL es ejecutado como si es un ALMACENAMIENTO.

30 La instrucción de ALMACENAMIENTO, cuando ha sido ejecutada por un procesador, coloca el primer operando (por ejemplo, los contenidos de un registro especificado por un campo de registro de la instrucción) en la ubicación del segundo operando (formada por añadir contenidos de los registros generales designados por los campos X_2 y B_2 con un valor definido por una concatenación de contenidos de los campos DH_2 y DL_2).

Dependiendo del modelo del procesador, el rendimiento de ALMACENAMIENTO NO TRANSACCIONAL puede ser más lento que el de ALMACENAMIENTO (STG).

35 Si una CPU hace almacenamientos transaccionales y no transaccionales a la misma ubicación de almacenamiento dentro de una transacción, y la transacción a continuación se anula, los contenidos de todas las ubicaciones de almacenamiento alteradas bien por el almacenamiento transaccional o bien por el almacenamiento no transaccional son impredecibles.

40 Una realización de la lógica asociada con el ALMACENAMIENTO NO TRANSACCIONAL es descrita con referencia a la fig. 11A. En una realización, un procesador (por ejemplo, la CPU) realiza esta lógica.

45 Con referencia a la fig. 11A, basado en la ejecución de la instrucción de ALMACENAMIENTO NO TRANSACCIONAL, es hecha una comprobación para ciertas excepciones, PASO 1100, y si existe una excepción, PETICIÓN 1102, es gestionada, en el PASO 1104. Por ejemplo, si el segundo operando no está alineado sobre un límite de doble palabra, es reconocida una excepción de especificación y es suprimida la ejecución de instrucción. Además, si no es instalada la instalación de ejecución transaccional, es reconocida una excepción de operación y es suprimida la ejecución de instrucción; si la transacción es una transacción restringida, es reconocida una excepción de restricción de transacción y es suprimida la ejecución; y si existe un problema con la ubicación del segundo operando (por ejemplo, excepción de protección, dirección inválida, etc.), es reconocida una excepción de acceso y es suprimida la ejecución de instrucción. De lo contrario si no existen excepciones, el primer operando de la instrucción es colocado de manera no transaccional (es decir, almacenado) sin cambios en una ubicación especificada por el segundo operando, PASO 1106.

En una realización, la instrucción de ALMACENAMIENTO NO TRANSACCIONAL es ejecutada en el modo de ejecución transaccional, y la colocación de manera no transaccional es retrasada hasta un final del modo de ejecución transaccional, tal como hasta la finalización de la transacción más exterior por la TEND o un modo de ejecución transaccional de finalización de anulación. (En otra realización, es retrasado solamente al final de una transacción particular en la cual se está ejecutando la instrucción). Además, puede haber múltiples almacenamientos no transaccionales si estos aparecen como almacenamientos concurrentes a otros procesadores.

En otra realización, el tratamiento del ALMACENAMIENTO NO TRANSACCIONAL depende del modo de ejecución del procesador y/o del tipo de transacción del cual es parte, como se ha descrito con referencia a la fig. 11B.

Con referencia a la fig. 11B, en una realización, basada en la ejecución de la instrucción de ALMACENAMIENTO NO TRANSACCIONAL por un procesador, se hace una determinación en cuanto a si el procesador está en el modo de ejecución transaccional (es decir, profundidad de anidación de transacción mayor que 0), PETICIÓN 1150. Si el procesador está en el modo de ejecución transaccional, entonces se hace otra determinación en cuanto a si la instrucción de ALMACENAMIENTO NO TRANSACCIONAL es parte de una transacción no restringida (por ejemplo, una transacción iniciada por la TBEGIN), PETICIÓN 1152. Si la instrucción es parte de una transacción no restringida, entonces la instrucción es ejecutada, como se ha descrito en este documento, PASO 1154. Sin embargo, si la instrucción es parte de una transacción no restringida (por ejemplo, iniciada por la TBEGINC y no tratada como una transacción no restringida), entonces es proporcionada una excepción de programa y es suprimida otra ejecución de la instrucción, PASO 1156.

Volviendo a la PETICIÓN 1150, si el procesador no está en el modo de ejecución transaccional, la instrucción de ALMACENAMIENTO NO TRANSACCIONAL es tratada como una instrucción de ALMACENAMIENTO y el tratamiento prosigue como tal, paso 1158.

Se ha descrito anteriormente una realización de una capacidad para almacenar información de forma explícita, mientras está en el modo de ejecución transaccional, que persiste incluso después de una anulación de una transacción asociada con la instrucción (es decir, una anulación de la transacción que incluye directamente la instrucción, o una anulación de otra transacción de la cual la transacción que incluye la instrucción está anidada dentro). La información es especificada por el usuario.

De manera adicional, se ha proporcionado anteriormente un medio eficiente de actualizar múltiples objetos no contiguos en la memoria sin serialización clásica (de grano grueso), tal como bloqueo, que proporciona un potencial para la mejora de rendimiento de multiprocesador significativa. Es decir, múltiples objetos, no contiguos son actualizados sin la aplicación de una clasificación de acceso de almacenamiento de grano más grueso que es proporcionada por técnicas clásicas, tales como bloqueos y semáforos. Son ofrecidas transacciones restringidas para actualizaciones simples de pequeña huella.

La ejecución transaccional puede ser utilizada en una variedad de escenarios, incluyendo, pero no limitado a, embebido parcial, tratamiento especulativo, y/u omisión de bloqueo. En el embebido parcial, la región parcial que ha de ser incluida en el trayecto ejecutado está envuelta en la TBEGIN/TEND. La TABORT puede ser incluida en ella al estado retrotraído en una salida lateral. Para especulación, tal como en Java, comprobaciones nulas en los punteros no referenciados pueden ser retrasadas para darles la vuelta utilizando una transacción. Si el puntero es nulo, la transacción puede ser anulada de forma segura utilizando la TABORT, que está incluida dentro de la TBEGIN/TEND.

Como se ha utilizado en este documento, el almacenamiento, almacenamiento central, almacenamiento principal, la memoria y la memoria principal son utilizados de manera que se pueden intercambiar, a menos que se indique lo contrario de manera implícita por el uso o de manera explícita. Además, mientras en una realización, una transacción que se retrasa de forma efectiva incluye retrasar los almacenamientos transaccionales de consignación a la memoria principal hasta la finalización de una transacción seleccionada; en otra realización, una transacción que se retrasa de forma efectiva incluye permitir actualizaciones transaccionales a la memoria, pero manteniendo los antiguos valores y restaurando la memoria a los antiguos valores en la anulación.

Como será apreciado por un experto en la técnica, uno o más aspectos pueden ser realizados como un sistema, método o producto de programa informático. Por consiguiente, uno o más aspectos pueden tomar la forma de una realización totalmente de hardware, una realización totalmente de software (incluyendo firmware, software residente, micro código, etc.) o una realización que combina aspectos de software y hardware que pueden ser todos generalmente referidos en este documento como un "circuito", "módulo" o "sistema". Además, uno o más aspectos pueden tomar la forma de un producto de programa informático realizado en uno o más medios legibles por ordenador que tienen el código del programa legible por ordenador realizado en el mismo.

Cualquier combinación de uno o más medios legibles por ordenador puede ser utilizada. El medio legible por ordenador puede ser un medio de almacenamiento legible por ordenador. Un medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero no está limitado a, un sistema, aparato, o dispositivo, electrónico, magnético, óptico, electromagnético, de infrarrojos o semiconductor o cualquier combinación adecuada de los anteriores. Ejemplos más específicos (una lista no exhaustiva) del medio de almacenamiento legible por ordenador

incluyen lo siguiente: una conexión eléctrica que tiene uno o más cables, un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM), una memoria de sólo lectura que se puede borrar de manera programable (EPROM o memorias Flash), una fibra óptica, una memoria de sólo lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético, o cualquier combinación adecuada de los anteriores. En el contexto de este documento, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que puede contener o almacenar un programa para utilizar por o en conexión con un sistema, aparato, o dispositivo de ejecución de instrucción.

Con referencia ahora a la fig. 12, en un ejemplo, un producto de programa informático 1200 incluye, por ejemplo, uno o más medios 1202 de almacenamiento legible por ordenador no transitorio para almacenar medios o lógica 1204 de código de programa legible por ordenador en ellos para proporcionar y facilitar una o más realizaciones.

El código de programa realizado sobre un medio legible por ordenador puede ser transmitido utilizando un medio apropiado, incluyendo pero no limitado a analógico, fibra óptica, cable, RF, etc., o cualquier combinación adecuada de los anteriores.

El código de programa informático para llevar a cabo las operaciones para una o más realizaciones puede ser escrito en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientada a objetos, tal como Java, Smalltalk, C++ o similar, y lenguajes de programación de procedimiento convencional, tal como lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programa se puede ejecutar completamente en el ordenador de usuario, parcialmente en el ordenador de usuario, como un paquete de software autónomo, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En el último escenario, el ordenador remoto puede estar conectado al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN) o una red de área amplia (WAN), o la conexión puede ser hecha a un ordenador externo (por ejemplo, a través de internet utilizando un Proveedor de Servicios de Internet).

Se han descrito en este documento uno más realizaciones con referencia a las ilustraciones del diagrama de flujo y/o a los diagramas de bloques de los métodos, aparatos (sistemas) y productos de programa informático. Se comprenderá que cada bloque de las ilustraciones de diagrama de flujo y/o de los diagramas de bloques, y combinaciones de los bloques en las ilustraciones del diagrama de flujo y/o los diagramas de bloques, pueden ser implementados por instrucciones de programa informático. Estas instrucciones de programa informático pueden ser proporcionadas a un procesador de un ordenador de propósito general, un ordenador de propósito especial, u otros aparatos de tratamiento de datos programables para producir una máquina, de tal manera que las instrucciones, que se ejecuta mediante el procesador del ordenador u otros aparatos de tratamiento de datos programables, crean medios para implementar las funciones/acciones especificadas en el diagrama de flujo y/o en el bloque o bloques del diagrama de bloques.

Estas instrucciones de programa informático pueden ser almacenadas también en un medio legible por ordenador que puede dirigir un ordenador, otro aparato de tratamiento de datos programables, u otros dispositivos para funcionar de una manera particular, de tal manera que las instrucciones almacenadas en el medio legible por ordenador producen un artículo de fabricación que incluye instrucciones que implementan la función/acción especificada en el diagrama de flujo y/o en el bloque o bloques del diagrama de bloques.

Las instrucciones del programa informático pueden ser cargadas también en un ordenador, en otros aparatos de tratamiento de datos programables, o en otros dispositivos para provocar una serie de pasos operacionales que han de ser realizados en el ordenador, en los otros aparatos programables o en los otros dispositivos para producir un proceso implementado por ordenador de tal manera que las instrucciones que se ejecutan en el ordenador o en otros aparatos programables proporcionan procesos para implementar las funciones/acciones especificadas en el diagrama de flujo y/o en el bloque o bloques del diagrama de bloques.

El diagrama de flujo y los diagramas de bloques en las figuras ilustran la arquitectura, funcionalidad, y funcionamiento de posibles implementaciones de sistemas, métodos y productos de programa informático según distintas realizaciones. A este respecto, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento, o porción de código, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. Debería observarse que, en algunas implementaciones alternativas, las funciones observadas en el bloque pueden ocurrir fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ser ejecutado sustancialmente de forma concurrente, o los bloques pueden algunas veces ser ejecutados en el orden inverso, dependiendo de la funcionalidad involucrada. Se observará también que cada bloque de los diagramas de bloques y/o la ilustración del diagrama de flujo, y combinaciones de los bloques en los diagramas de bloques y/o la ilustración del diagrama de flujo, pueden ser implementados por sistemas basados en hardware de propósito especial que realizan las funciones o acciones especificadas, o combinaciones de instrucciones de hardware y de ordenador de propósito especial.

Además de lo anterior, pueden ser proporcionados, ofrecidos, desplegados, gestionados, mantenidos, etc., uno o más aspectos por un proveedor de servicios que ofrece gestión de entornos de cliente. Por ejemplo, el proveedor de

servicios puede crear, mantener, soportar, etc. código informático y/o infraestructura informática que realiza uno o más aspectos para uno o más clientes. A cambio, el proveedor de servicios puede recibir el pago del cliente bajo una suscripción y/o acuerdo de honorarios como ejemplos. Adicional o alternativamente, el proveedor de servicios puede recibir el pago de la venta de contenido de publicidad de uno o más terceros.

- 5 En un aspecto, se puede desplegar una aplicación para realizar una o más realizaciones. Como ejemplo, el despliegue de una aplicación comprende proporcionar infraestructura informática operable para realizar una o más realizaciones.

10 Como otro aspecto, una infraestructura informática puede ser desplegada comprendiendo integración de código legible por ordenador en un sistema informático, en el que el código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

Como aún otro aspecto, puede proporcionarse un proceso para integrar infraestructura informática que comprende integración de código legible por ordenador en un sistema informático. El sistema informático comprende un medio legible por ordenador, en el que el medio informático comprende una o más realizaciones. El código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

- 15 Aunque se han descrito anteriormente distintas realizaciones, éstas son sólo ejemplos. Por ejemplo, se pueden utilizar los entornos informáticos de otras arquitecturas. Además, se pueden utilizar diferentes instrucciones, formatos de instrucciones, campos de instrucciones y/o valores de instrucciones. Además, pueden ser proporcionadas/utilizadas diferentes limitaciones/restricciones adicionales u otras. Son posibles muchas variaciones.

20 Además, pueden ser utilizados otros tipos de entornos informáticos. Como un ejemplo, un sistema de tratamiento de datos adecuado para almacenar y/o ejecutar código de programa es utilizable que incluye al menos dos procesadores acoplados directa o indirectamente a los elementos de memoria a través de un bus de sistema. Los elementos de memoria incluyen, por ejemplo, memoria local empleada durante la ejecución real del código de programa, almacenamiento a granel, y memoria caché que proporciona almacenamiento temporal de al menos algún código de programa con el fin de reducir el número de veces que el código debe ser recuperado del almacenamiento a granel durante la ejecución.

25 Los dispositivos Entrada/Salida o I/O (que incluyen, pero no están limitados a, teclados, dispositivos de presentación, dispositivos indicadores, DASD, cinta, CD, DVD, memorias USB y otros medios de memoria, etc.) pueden ser acoplados al sistema bien directamente o bien a través de controladores I/O de intervención. Los adaptadores de red pueden ser acoplados también al sistema para permitir que el sistema de tratamiento de datos resulte acoplado a otros sistemas de tratamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas de intervención. Los modem, cable modem, y tarjetas de redes son sólo unos pocos de los dispositivos disponibles de adaptadores de red.

30 Con referencia a la fig. 13, se han descrito componentes representativos del sistema Informático Anfitrión 5000 para implementar una o más realizaciones. El sistema informático anfitrión 5000 representativo comprende una o más CPU 5001 en comunicación con la memoria del ordenador (es decir, almacenamiento central) 5002, así como interfaces I/O para dispositivos de medios de almacenamiento 5011 y redes 5010 para comunicar con otros ordenadores o SAN y similares. La CPU 5001 es conforme a una arquitectura que tiene un conjunto de instrucciones arquitectónicas y funcionalidad arquitectónica. La CPU 5001 puede tener traducción de registro de acceso (ART) 5012, que incluye una memoria temporal que mira hacia un lado ART (ALB) 5013, para seleccionar un espacio de dirección para ser utilizado por traducción de dirección dinámica (DAT) 5003 para transformar las direcciones de programa (direcciones virtuales) en direcciones de memoria real. Una DAT incluye típicamente una memoria temporal que mira hacia un lado de traducción (TLB) 5007 para almacenar traducciones de manera que los últimos accesos al bloque de la memoria de ordenador 5002 no requieren el retraso de traducción de dirección. Típicamente, una memoria caché 5009 es empleada entre la memoria de ordenador 5002 y el procesador 5001. La memoria caché 5009 puede ser jerárquica que tiene una memoria caché grande disponible para más de una CPU y memorias caché más pequeñas, más rápidas (nivel inferior) entre la memoria caché grande y cada CPU. En algunas implementaciones, las memorias caché de nivel inferior son divididas para proporcionar memorias caché de nivel bajo separadas para recuperación de instrucción y accesos de datos. En una realización para la instalación TX, un bloque de diagnóstico de transacción (TDB) 5100 y una o más memorias temporales 5101 pueden ser almacenadas en una o más de las memorias caché 5009 y la memoria 5002. En un ejemplo, en el modo TX, los datos son inicialmente almacenados en una memoria temporal TX, y cuando finaliza el modo TX (por ejemplo, la TEND más exterior), los datos en la memoria temporal son almacenados (consignados) en la memoria, o si hay una anulación, los datos son descartados en la memoria temporal.

55 En una realización, una instrucción es recuperada de la memoria 5002 por una unidad 5004 de recuperación de instrucción mediante una memoria caché 5009. La instrucción es decodificada en una unidad 5006 de decodificación de instrucción y remitida (con otras instrucciones en algunas realizaciones) a la unidad o unidades 5008 de ejecución de instrucción. Típicamente se emplean varias unidades 5008 de ejecución, por ejemplo una unidad de ejecución aritmética, una unidad de ejecución de coma flotante y una unidad de ejecución de instrucción de derivación. Además, en una realización de la instalación TX, pueden ser empleados distintos controles TX 5110. La

instrucción es ejecutada por la unidad de ejecución, accediendo a los operandos desde los registros especificados de instrucción o la memoria si es necesario. Si un operando ha de ser accedido (cargado o almacenado) desde la memoria 5002, una unidad 5005 de carga/almacenamiento maneja típicamente el acceso bajo control de la instrucción que es ejecutada. Las instrucciones pueden ser ejecutadas en circuitos de hardware o en un microcódigo interno (firmware) o por una combinación de ambos.

De acuerdo con un aspecto de la instalación TX, el procesador 5001 incluye también una PSW 5102 (por ejemplo, TX y/o una PSW de anulación), una profundidad de anidación 5104, una TDBA 5106, y uno o más registros de control 5108.

Como se ha observado, un sistema informático incluye información en almacenamiento local (o principal), así como direccionamiento, protección, y referencia y grabación de cambio. Algunos aspectos del direccionamiento incluyen el formato de las direcciones, el concepto de los espacios de dirección, los distintos tipos de direcciones, y la manera en que un tipo de dirección es traducido a otro tipo de dirección. Algunos almacenamientos principales incluyen ubicaciones de almacenamiento permanentemente asignadas. El almacenamiento principal proporciona el sistema con almacenamiento de datos de acceso rápido que se pueden dirigir directamente. Tanto los datos como los programas han de ser cargados en el almacenamiento principal (desde dispositivos de entrada) antes de que puedan ser procesados.

El almacenamiento principal puede incluir uno o más almacenamientos de memoria temporal de acceso rápido, más pequeños, llamados algunas veces memorias caché. Una memoria caché está típicamente asociada de manera física con una CPU o un procesador I/O. Los efectos, excepto en el rendimiento, de la construcción física y uso de los distintos medios de almacenamiento no se pueden observar generalmente por el programa.

Las memorias caché separadas pueden ser mantenidas para instrucciones y para operandos de datos. La información dentro de una memoria caché es mantenida en bytes contiguos sobre un límite integral llamado un bloque caché o línea caché (o línea, para abreviar). Un modelo puede proporcionar una instrucción de EXTRACCIÓN DE ATRIBUTOS DE MEMORIA CACHÉ que devuelve el tamaño de la línea caché en bytes. Un modelo puede proporcionar también instrucciones de DATOS DE RECUPERACIÓN PREVIA y DATOS DE RECUPERACIÓN PREVIA RELATIVOS LARGOS que efectúan la recuperación previa del almacenamiento en los datos o memoria caché de instrucción o la liberación de datos desde la memoria caché.

El almacenamiento es visto como una cadena de bits horizontal larga. Para la mayoría de las operaciones, los accesos al almacenamiento proceden en una secuencia de izquierda a derecha. La cadena de bits es subdividida en unidades de ocho bits. Una unidad de ocho bits es llamada 1 byte, que es el bloque de construcción básico de todos los formatos de información. Cada ubicación de byte en el almacenamiento es identificada por un número entero no negativo único, que es la dirección de dicha ubicación del byte o, simplemente, la dirección del byte. Las ubicaciones del byte adyacentes tienen direcciones consecutivas, comenzando con 0 a la izquierda y procediendo en una secuencia de izquierda a derecha. Las direcciones son números enteros binarios sin signo y son 24, 31, o 64 bits.

La información es transmitida entre el almacenamiento y una CPU o un subsistema de canal de un byte, o grupo de bytes, a la vez. A menos que se especifique lo contrario, por ejemplo, en la Arquitectura z, un grupo de bytes en el almacenamiento es dirigido por el byte más a la izquierda del grupo. El número de bytes en el grupo es o bien implícito o bien está explícitamente especificado por la operación que ha de ser realizada. Cuando se ha utilizado en una operación de CPU, un grupo de bytes es llamado un campo. Dentro de cada grupo de bytes, por ejemplo, en la Arquitectura z, los bits son numerados en una secuencia de izquierda a derecha. En la Arquitectura z, los bits más a la izquierda son algunas veces referidos como bits de "orden alto" y los bits más a la derecha como los bits de "orden bajo". Los números de bits no son direcciones de almacenamiento, sin embargo. Solamente pueden ser direccionados los bytes. Para operar en bits individuales de 1 byte en el almacenamiento, es accedido el byte completo. Los bits en un byte son numerados de 0 a 7, de izquierda a derecha (por ejemplo, en la Arquitectura z). Los bits en una dirección pueden ser numerados de 8-31 o 40-63 para direcciones de 24 bits, o de 1-31 o 33-63 para direcciones de 31 bits; son numerados de 0-63 para direcciones de 64 bits. En un ejemplo, los bits 8-31 y 1-31 se aplican a direcciones que están en una ubicación (por ejemplo, el registro) que es de 32 bits de ancho, mientras que los bits 40-63 y 33-63 se aplican a direcciones que están en una ubicación de 64 bits de ancho. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato son numerados consecutivamente comenzando desde cero. Para propósitos de detección de error, y preferiblemente para corrección, pueden ser transmitidos uno o más bits de comprobación con cada byte o con un grupo de bytes. Tales bits de comprobación son generados automáticamente por la máquina y no pueden ser controlados directamente mediante el programa. Las capacidades de almacenamiento son expresadas en números de bytes. Cuando la longitud de un campo de operando de almacenamiento es implicada por el código de operación de una instrucción, el campo se dice que tiene una longitud fija, que puede ser uno, dos, cuatro, ocho, o dieciséis bits. Pueden implicarse campos más grandes para algunas instrucciones. Cuando la longitud de un campo de operando de almacenamiento no está implícita pero está indicada explícitamente, se dice que el campo tiene una longitud variable. Operandos de longitud variable pueden variar en longitud por incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información es colocada en el almacenamiento, los contenidos de solamente dichas ubicaciones de byte son reemplazados ya que son incluidos en el campo designado, incluso aunque la anchura del trayecto físico para el almacenamiento pueda ser mayor que la longitud del campo que es almacenado.

Ciertas unidades de información deben estar en un límite integral en el almacenamiento. Un límite es llamado integral para una unidad de información cuando su dirección de almacenamiento es un múltiplo de la longitud de la unidad en bytes. Se han dado nombres especiales a los campos de 2, 4, 8, 16 y 32 bytes en un límite integral. Una media palabra es un grupo de dos bytes consecutivos sobre un límite de dos bytes y es el bloque de instrucciones de construcción básica. Una palabra es un grupo de cuatro bytes consecutivos sobre un límite de cuatro bytes. Una doble palabra es un grupo de ocho bytes consecutivos sobre un límite de ocho bytes. Una palabra óctuple es un grupo de 32 bytes consecutivos sobre un límite de 32 bytes. Una palabra cuádruple es un grupo de 16 bytes consecutivos sobre un límite de 16 bytes. Cuando las direcciones de almacenamiento designan medias palabras, palabras, dobles palabras, palabras cuádruples, y palabras óctuples, la representación binaria de la dirección contiene uno, dos, tres, cuatro o cinco bits cero más a la derecha, respectivamente. Las instrucciones deben ser límites integrales de dos bytes. Los operandos de almacenamiento de la mayoría de las instrucciones no tienen requisitos de alineación de límites.

En dispositivos que implementan memorias caché separadas para instrucciones y operandos de datos, pueden experimentarse un retraso significativo si el programa almacena en una línea de memoria caché desde la cual las instrucciones son recuperadas posteriormente, independientemente de si el almacenamiento altera las instrucciones que son recuperadas posteriormente.

En un ejemplo, las realizaciones pueden ser puestas en práctica por software (algunas veces referido como código interno licenciado, firmware, micro-código, mili-código, pico-código, y similares, cualquiera de los cuales sería consistente con una o más realizaciones). Con referencia a la fig. 13, el código de programa de software que realiza uno o más aspectos puede ser accedido por el procesador 5001 del sistema anfitrión 5000 a partir de dispositivos 5011 de medios de almacenamiento a largo plazo, tales como una unidad CD-ROM, una unidad de cinta o un disco duro. El código de programa de software puede ser realizado sobre cualquiera de una variedad de medios conocidos para utilizar con un sistema de tratamiento de datos, tales como un disquete, un disco duro, o un CD-ROM. El código puede ser distribuido sobre tales medios, o puede ser distribuido a usuarios a partir de la memoria de ordenador 5002 o el almacenamiento de un sistema informático sobre una red 5010 a otros sistemas informáticos para utilizar por usuarios de tales otros sistemas.

El código de programa de software incluye un sistema operativo que controla la función e interacción de los distintos componentes informáticos y uno o más programas de aplicación. El código de programa es normalmente paginado a partir del dispositivo 5011 de medios de almacenamiento al almacenamiento 5002 de ordenador de velocidad relativamente más elevada donde está disponible para el tratamiento por el procesador 5001. Las técnicas y métodos para realizar el código de programa de software en la memoria, sobre medio físico, y/o el código de software de distribución mediante redes son conocidos y no serán además descritos en este documento. El código de programa, cuando es creado y almacenado sobre un medio tangible (incluyendo pero no limitado a módulos de memoria electrónica (RAM), memoria flash, Discos Compactos (CD), DVD, Cinta Magnética y similares es a menudo referido como un "producto de programa informático". El medio de producto de programa informático es típicamente legible por un circuito de tratamiento preferiblemente en un sistema informático para la ejecución mediante el circuito de tratamiento.

La fig. 14 ilustra una estación de trabajo representativa o sistema de hardware de servidor en el cual se pueden poner en práctica una o más realizaciones. El sistema 5020 de la fig. 14 comprende un sistema informático 5021 de base representativa, tal como un ordenador personal, una estación de trabajo o un servidor, que incluyen dispositivos periféricos opcionales. El sistema informático 5021 de base incluye uno o más procesadores 5026 y un bus empleado para conectar y habilitar la comunicación entre el procesador o procesadores 5026 y los otros componentes del sistema 5021 según las técnicas conocidas. El bus conecta el procesador 5026 a la memoria 5025 y el almacenamiento 5027 a largo plazo que puede incluir un disco duro (incluyendo cualquier medio magnético, CD, DVD y memoria flash, por ejemplo) o una unidad de cinta por ejemplo. El sistema 5021 podría incluir también un adaptador de interfaz de usuario, que conecta el microprocesador 5026 mediante el bus a uno o más dispositivos de interfaz, tal como un teclado 5024, un ratón 5023, una impresora/escáner 5030 y/u otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal como una pantalla táctil, teclado de entrada digitalizado, etc. El bus conecta también un dispositivo 5022 de presentación, tal como una pantalla o monitor LCD, al microprocesador 5026 mediante un adaptador de presentación.

El sistema 5021 puede comunicar con otros ordenadores o redes de ordenadores por medio de un adaptador de red 5028 capaz de comunicar con una red 5029. Ejemplos de adaptadores de red son canales de comunicaciones, Anillo identificador, Ethernet o módems. Alternativamente, el sistema 5021 puede comunicar utilizando una interfaz inalámbrica, tal como una tarjeta CDPD (datos de paquete digital celular). El sistema 5021 puede estar asociado con tales otros ordenadores en una Red de Área Local (LAN) o una Red de Área Amplia (WAN), o el sistema 5021 puede ser un cliente en una disposición cliente/servidor con otro ordenador, etc. Son conocidas en la técnica todas estas configuraciones, así como el hardware y software de comunicaciones apropiadas.

La fig. 15 ilustra una red 5040 de tratamiento de datos en la cual pueden ser puestas en práctica una o más realizaciones. La red 5040 de tratamiento de datos puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red cableada, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo individuales 5041, 5042, 5043, 5044. Adicionalmente, como los expertos en la técnica apreciarán, una o más

LAN pueden ser incluidas, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador anfitrión.

Con referencia aún a la fig. 15, las redes pueden incluir también ordenadores o servidores centrales, tales como un ordenador de pasarela (servidor cliente 5046) o servidor de aplicación (servidor remoto 5048 que puede acceder a un repositorio de datos y puede ser accedido también directamente desde una estación de trabajo 5045). Un ordenador de pasarela 5046 sirve como un punto de entrada a cada red individual. Una pasarela es necesaria cuando se conecta un protocolo de red a otro. La pasarela 5046 puede ser acoplada preferiblemente a otra red (internet 5047 por ejemplo) por medio de un enlace de comunicaciones. La pasarela 5046 puede ser también acoplada directamente a una o más estaciones de trabajo 5041, 5042, 5043, 5044 utilizando un enlace de comunicaciones. El ordenador de pasarela puede ser implementado utilizando un servidor z del Sistema eServer de IBM disponible desde International Business Machines Corporation.

Con referencia de manera concurrente a la fig. 14 y a la fig. 15, el código 5031 de programación de software que puede constituir una o más realizaciones puede ser accedido por el procesador 5026 del sistema 5020 desde el medio de almacenamiento 5027 a largo plazo, tal como una unidad CD-ROM o un disco duro. El código de programación de software puede ser realizado sobre cualquiera de una variedad de medios conocidos para utilizar con un sistema de tratamiento de datos, tal como un disquete, disco duro, o CD-ROM. El código puede ser distribuido sobre tales medios, o puede ser distribuido a los usuarios 5050, 5051 desde la memoria o almacenamiento de un sistema informático sobre una red a otros sistemas informáticos para utilizar por usuarios de tales otros sistemas.

Alternativamente, el código de programación puede ser realizado en la memoria 5025, y accedido por el procesador 5026 utilizando el bus del procesador. Tal código de programación incluye un sistema operativo que controla la función e interacción de los distintos componentes informáticos y uno o más programas de aplicación 5032. El código de programa es paginado normalmente desde el medio de almacenamiento 5027 a la memoria de alta velocidad 5025 donde está disponible para el tratamiento por el procesador 5026. Las técnicas y métodos para realizar código de programación de software en la memoria, en medios físicos, y/o código de software de distribución mediante redes son bien conocidos y no serán además descritos en este documento. El código de programa, cuando es creado y almacenado sobre un medio tangible (incluyendo pero no limitado a módulos de memoria electrónica (RAM), memoria de flash, Discos Compactos (CD), DVD, Cinta Magnética y similares es a menudo denominado como un "producto de programa informático". El medio de producto de programa informático es legible típicamente por un circuito de tratamiento preferiblemente en un sistema informático para la ejecución mediante el circuito de tratamiento.

La memoria caché que está disponible más fácilmente para el procesador (normalmente más rápida y más pequeña que otras memorias caché del procesador) es la memoria caché más baja (L1 o nivel uno) y el almacenamiento principal (memoria principal) es la memoria caché de nivel más alto (L3 si hay 3 niveles). La memoria caché de nivel más bajo está a menudo dividida en una memoria caché de instrucción (I-Caché) que contiene instrucciones de máquina que han de ser ejecutadas y una memoria caché de datos (D-Caché) que contiene operandos de datos.

Con referencia a la fig. 16, una realización del procesador ejemplar es representada por el procesador 5026. Típicamente uno o más niveles de la memoria caché 5053 son empleados para bloques de memoria temporal con el fin de mejorar el rendimiento del procesador. La memoria caché 5053 es una memoria temporal de alta velocidad que contiene las líneas de memoria caché de los datos de memoria que han de ser utilizados probablemente. Las líneas de memoria caché típicas son 64, 128 o 256 bytes de datos de memoria. Las memorias caché separadas son a menudo empleadas más para almacenar en memoria caché instrucciones que para almacenar datos de memoria caché. La coherencia de la memoria caché (sincronización de copias de líneas en memoria y las memorias caché) es a menudo proporcionada por distintos algoritmos "snoop" bien conocidos en la técnica. El almacenamiento de memoria principal 5025 de un sistema de procesador es a menudo denominado como una memoria caché. En un sistema de procesador que tiene 4 niveles de memoria caché 5053, el almacenamiento principal 5025 es a menudo denominado como la memoria caché de nivel 5 (L5) ya que es típicamente más rápida y solamente contiene una porción del almacenamiento no volátil (DASD, cinta etc.) que está disponible para un sistema informático. Las páginas de datos de las "memorias caché" del almacenamiento principal 5025 son paginadas dentro y fuera del almacenamiento principal 5025 por el sistema operativo.

Un contador de programa 5061 (contador de instrucción) realiza un seguimiento de la dirección de la instrucción actual que ha de ser ejecutada. Un contador de programa en un procesador de Arquitectura z es de 64 bits y puede ser truncado a 31 o 24 bits para soportar los límites de direccionamiento anteriores. Un contador de programa es realizado típicamente en una PSW (palabra de estado de programa) de un ordenador de tal manera que persiste durante el cambio de contexto. Así, un programa en progreso, que tiene un valor de contador de programa, puede ser interrumpido, por ejemplo, por el sistema operativo (cambio de contexto desde el entorno de programa al entorno del sistema operativo). La PSW del programa mantiene el valor del contador de programa mientras el programa no está activo, y el contador de programa (en la PSW) del sistema operativo es utilizado mientras el sistema operativo se está ejecutando. Típicamente, el contador de programa es incrementado por una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Ordenador de Conjunto de Instrucciones Reducidas) son típicamente de longitud fija mientras las instrucciones CISC (Ordenador de Conjunto de Instrucciones Complejas)

son típicamente de longitud variable. Las instrucciones de la Arquitectura z de IBM son instrucciones CISC que tienen una longitud de 2, 4 o 6 bytes. El contador de programa 5061 es modificado bien por una operación de cambio de contexto o bien por una operación tomada de derivación de una instrucción de derivación por ejemplo. En una operación de cambio de contexto, el valor del contador de programa actual es guardado en la palabra de estado de programa junto con otra información de estado sobre el programa que es ejecutado (tales como códigos de condición), y un nuevo valor de contador de programa es cargado apuntando a una instrucción de un nuevo módulo de programa que ha de ser ejecutado. Una operación tomada de derivación es realizada con el fin de permitir al programa tomar decisiones o bucles dentro del programa cargando el resultado de la instrucción de derivación en el contador de programa 5061.

Típicamente una unidad 5055 de recuperación de instrucción es empleada para recuperar instrucciones en nombre del procesador 5026. La unidad de recuperación bien recupera las "siguientes instrucciones secuenciales", las instrucciones objetivo de las instrucciones tomadas de derivación, o bien las primeras instrucciones de un programa que sigue a un cambio de contexto. Las unidades de recuperación de Instrucción de Módem emplean a menudo técnicas de recuperación previa para recuperar previamente de manera especulativa instrucciones basadas en la probabilidad de que las instrucciones recuperadas previamente puedan ser utilizadas. Por ejemplo, una unidad de recuperación puede recuperar 16 bytes de instrucción que incluyen la siguiente instrucción secuencial y bytes adicionales de otras instrucciones secuenciales.

Las instrucciones recuperadas son a continuación ejecutadas por el procesador 5026. En una realización, la instrucción o instrucciones recuperadas son pasadas a una unidad de envío 5056 de la unidad de recuperación. La unidad de envío decodifica la o las instrucciones y reenvía la información acerca de la o las instrucciones decodificadas para unidades apropiadas 5057, 5058, 5060. Una unidad de ejecución 5057 recibirá típicamente información sobre instrucciones aritméticas decodificadas a partir de la unidad 5055 de recuperación de instrucción y realizará operaciones aritméticas sobre operandos de acuerdo con el código de operación de la instrucción. Los operandos son proporcionados a la unidad 5057 de ejecución preferiblemente bien desde la memoria 5025, registros arquitectónicos 5059 o bien desde un campo intermedio de la instrucción que es ejecutada. Los resultados de la ejecución, cuando son almacenados, son almacenados bien en la memoria 5025, en los registros 5059 o bien en otro hardware de máquina (tales como registros de control, registros de PSW y similares).

Las direcciones virtuales son transformadas en direcciones reales utilizando traducción de dirección dinámica 5062 y, opcionalmente, utilizando transacción de registro de acceso 5063.

Un procesador 5026 tiene típicamente una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Con referencia a la fig. 17A, una unidad 5057 de ejecución puede comunicar 5071 con los registros generales 5059 arquitectónicos, una unidad 5056 de codificación/envío, una unidad 5060 de almacenamiento de carga, y otras 5065 unidades de procesador a modo de lógica de interconexión 5071. Una unidad 5057 de ejecución puede emplear distintos circuitos de registro 5067, 5068, 5069 para contener la información sobre la que operará la unidad lógica aritmética (ALU) 5066. La ALU realiza operaciones aritméticas tales como sumar, restar, multiplicar y dividir así como funciones lógicas tales como y, o y o exclusiva (XOR), girar y desplazar. Preferiblemente la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras instalaciones arquitectónicas 5072 que incluyen códigos de condición y lógica de soporte de recuperación por ejemplo. Típicamente el resultado de una operación ALU es contenido en un circuito 5070 de registro de salida que puede reenviar el resultado a una variedad de otras funciones de tratamiento. Existen muchas disposiciones de unidades de procesador, la presente descripción está destinada solamente a proporcionar una comprensión representativa de una realización.

Una instrucción de SUMAR por ejemplo sería ejecutada en una unidad 5057 de ejecución que tiene funcionalidad aritmética y lógica mientras una instrucción de coma flotante por ejemplo sería ejecutada en una ejecución de coma flotante que tiene capacidad de coma flotante especializada. Preferiblemente, una unidad de ejecución opera sobre operandos identificados por una instrucción que realiza una función definida de código de operación sobre los operandos. Por ejemplo, una instrucción de SUMAR puede ser ejecutada por una unidad 5057 de ejecución sobre operandos encontrados en dos registros 5059 identificados por los campos de registro de la instrucción.

La unidad 5057 de ejecución realiza la suma aritmética sobre los operandos y almacena el resultado en el tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros fuente. La unidad de ejecución utiliza preferiblemente una Unidad Lógica Aritmética (ALU) 5066 que es capaz de realizar una variedad de funciones lógicas tales como Desplazar, Girar, Y, O y XO así como una variedad de funciones algebraicas incluyendo cualquiera de entre sumar, restar, multiplicar, dividir. Algunas ALU 5066 están diseñadas para operaciones escalares y algunas para coma flotante. Los datos pueden ser Big Endian (donde el byte menos significativo está en la dirección de byte mayor) o Little Endian (donde el byte menos significativo está en la dirección de byte menor) dependiendo de la arquitectura. La Arquitectura z de IBM es Big Endian. Campos con signo pueden ser signo y magnitud, complemento de 1, o complemento de 2 dependiendo de la arquitectura. Un número complemento de 2 es ventajoso porque la ALU no necesita designar una capacidad de resta ya que o bien un valor negativo o bien un valor positivo en el complemento de 2 requiere solamente una suma dentro de la ALU. Los números son comúnmente descritos en clave, donde un campo de 12 bits define una dirección de un bloque de 4.096 bytes y es comúnmente descrito como un bloque de 4 Kbyte (kilo-byte), por ejemplo.

Con referencia a la fig. 17B, una información de instrucción de derivación para ejecutar una instrucción de derivación es enviada típicamente a una unidad 5058 de derivación que emplea a menudo un algoritmo de predicción de derivación tal como una tabla 5082 de historial de derivación para predecir el resultado de la derivación antes de que se completen otras operaciones condicionales. El objetivo de la instrucción de derivación actual será recuperado y ejecutado especulativamente antes de que se completen las operaciones condicionales. Cuando se completan las operaciones condicionales las instrucciones de derivación especulativamente ejecutadas son bien completadas o bien descartadas basado en las condiciones de la operación condicional y el resultado especulado. Una instrucción de derivación típica puede probar códigos de condición y derivación a una dirección objetivo si los códigos de condición cumplen con el requisito de derivación de la instrucción de derivación, una dirección objetivo puede ser calculada basado en distintos números que incluyen unos encontrados en los campos de registro o un campo inmediato de la instrucción por ejemplo. La unidad 5058 de derivación puede emplear una ALU 5074 que tiene una pluralidad de circuitos de registro de entrada 5075, 5076, 5077 y un circuito de registro de salida 5080. La unidad 5058 de derivación puede comunicar 5081 con registros generales 5059, con la unidad 5056 de envío de decodificación o con otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones puede ser interrumpida por diversas razones que incluyen un cambio de contexto iniciado por un sistema operativo, una excepción de programa o error que provoca un cambio de contexto, una señal interrupción I/O que causa un cambio de contexto o actividad de múltiples hilos de una pluralidad de programas (en un entorno de múltiples hilos), por ejemplo. Preferiblemente una acción de cambio de contexto guarda la información de estado sobre un programa que se está ejecutando actualmente y a continuación carga la información de estado sobre otro programa que es invocado. La información de Estado puede ser guardada en registros hardware o en una memoria por ejemplo. La información de estado comprende preferiblemente un valor de contador de programa que apunta a una siguiente instrucción que ha de ser ejecutada, códigos de condición, información de traducción de memoria y contenido de registro arquitectónico. Una actividad de cambio de contexto puede ser ejercitada por circuitos hardware, programas de aplicación, programas de sistema operativo o código firmware (micro-código, pico-código o código interno licenciado (LIC)) sólo o en combinación.

Un procesador accede a los operandos según los métodos definidos por la instrucción. La instrucción puede proporcionar un operando inmediato que utiliza el valor de una porción de la instrucción, puede proporcionar uno o más campos de registro que apuntan explícitamente bien a los registros de propósito general o bien a los registros de propósito especial (por ejemplo registros de coma flotante). La instrucción puede utilizar registros implicados identificados por un campo de código de operación como operandos. La instrucción puede utilizar ubicaciones de memoria para los operandos. Una ubicación de memoria de un operando puede ser proporcionada por un registro, un campo inmediato, o una combinación de registros y campo inmediato como se ha ejemplificado por la instalación de desplazamiento largo de la Arquitectura z en donde la instrucción define un registro base, un registro índice y campo inmediato (campo de desplazamiento) que son añadidos juntos para proporcionar la dirección del operando en la memoria por ejemplo. La ubicación en este documento implica típicamente una ubicación en la memoria principal (almacenamiento principal) a menos que se indique lo contrario.

Con referencia a la fig. 17C, un procesador accede al almacenamiento utilizando una unidad 5060 de carga/almacenamiento. La unidad 5060 de carga/almacenamiento puede realizar una operación de carga obteniendo la dirección del operando objetivo en la memoria 5053 y cargando el operando en un registro 5059 u otra ubicación de memoria 5053, o puede realizar una operación de almacenamiento obteniendo la dirección del operando objetivo en la memoria 5053 y almacenando datos obtenidos desde un registro 5059 u otra ubicación de memoria 5053 en la ubicación de operando objetivo en la memoria 5053. La unidad 5060 de carga/almacenamiento puede ser especulativa y puede acceder a la memoria en una secuencia que esta fuera de servicio con relación a la secuencia de instrucción, sin embargo la unidad 5060 de carga/almacenamiento está para mantener la apariencia para programas cuyas instrucciones fueron ejecutadas en orden. Una unidad 5060 de carga/almacenamiento puede comunicar 5084 con los registros generales 5059, la unidad 5056 de decodificación/envío, la interfaz de memoria caché/memoria 5053 u otros elementos 5083 y comprende distintos circuitos de registro 5086, 5087, 5088 y 5089, las ALU 5085 y la lógica de control 5090 para calcular las direcciones de almacenamiento y proporcionar secuenciación de estructura para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de servicio pero la unidad de carga/almacenamiento proporciona funcionalidad para hacer que las operaciones fuera de servicio aparezcan en el programa como que han sido realizadas en orden, como es bien conocido en la técnica.

Preferiblemente las direcciones que "ve" un programa de aplicación son a menudo denominadas como direcciones virtuales. Las direcciones virtuales son algunas veces denominadas como "direcciones lógicas" y "direcciones efectivas". Estas direcciones virtuales son virtuales ya que son dirigidas de nuevo a la ubicación de memoria física por una de una variedad de tecnologías de traducción de dirección dinámica (DAT) que incluyen, pero no están limitadas a, prefijar de manera simple una dirección virtual con un valor de desplazamiento, traducir la dirección virtual mediante una o más tablas de traducción, comprendiendo preferiblemente las tablas de traducción al menos una tabla de segmento y una tabla de páginas sola o en combinación, preferiblemente, teniendo la tabla de segmento una entrada que apunta a la tabla de páginas. En la Arquitectura z, es proporcionada una jerarquía de traducción incluyendo una tabla de primera región, una tabla de segunda región, una tabla de tercera región, una tabla de segmento y una tabla de página opcional. El rendimiento de la traducción de dirección es a menudo mejorado utilizando una memoria temporal que mira a un lado de traducción (TLB) que comprende entradas que hacen corresponder una dirección virtual a una ubicación de memoria física asociada. Las entradas son creadas

cuando la DAT traduce una dirección virtual utilizando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar a continuación la entrada de la TLB rápida en vez de los accesos lentos a la tabla de traducción secuencial. El contenido de la TLB puede ser gestionado por diversos algoritmos de reemplazo que incluyen el LRU (usados Menos Recientemente).

- 5 En el caso donde el procesador es un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tales como I/O, memorias caché, TLB y memoria, interconectados por coherencia. Típicamente, las tecnologías "snoop" serán utilizadas para mantener la coherencia de la memoria caché. En un entorno snoop, cada línea de memoria caché puede ser marcada como que está en cualquiera de un estado compartido, un estado exclusivo, un estado cambiado, un estado inválido y similar con el fin de facilitar la compartición.

10 Las unidades 5054 I/O (fig. 16) proporcionan al procesador con medios para unirse a dispositivos periféricos que incluyen cinta, disco, impresoras, dispositivos de presentación, y redes por ejemplo. Las unidades I/O son a menudo presentadas al programa informático por controladores de software. En ordenadores centrales, tales como el Sistema z de IBM®, los adaptadores de canal y adaptadores de sistema abierto son unidades I/O del ordenador central que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

15 Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos. Como ejemplo, un entorno puede incluir un emulador (por ejemplo, software u otro mecanismo de emulación), en el cual una arquitectura particular (incluyendo, por ejemplo, ejecución de instrucción, funciones arquitectónicas, tales como traducción de dirección, y registros arquitectónicos) o un subconjunto del mismo es emulado (por ejemplo, sobre un sistema informático nativo que tiene procesador y memoria). En tal entorno, una o más funciones de emulación del emulador pueden implementar una o más realizaciones, incluso aunque un ordenador que ejecuta el emulador puede tener una arquitectura diferente de las capacidades que son emuladas. Como ejemplo, en el modo emulación, la instrucción u operación específica que es simulada es decodificada, y es construida una función de emulación apropiada para implementar la instrucción u operación individual.

20 En un entorno de emulación, un ordenador anfitrión incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de recuperación de instrucción para recuperar instrucciones desde la memoria y opcionalmente, proporcionar almacenamiento temporal local para la instrucción recuperada; una unidad de decodificación de instrucción para recibir las instrucciones recuperadas y para determinar el tipo de instrucciones que han sido recuperadas; y una unidad de ejecución de instrucción para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde la memoria; almacenar datos de nuevo a la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, como se ha determinado por la unidad de decodificación. En un ejemplo, cada unidad es implementada en software. Por ejemplo, las operaciones que son realizadas por las unidades son implementadas como una o más subrutinas dentro del software de emulador.

25 Más particularmente, en un ordenador central, las instrucciones de máquina arquitectónicas son utilizadas por programadores, normalmente programadores en "C" hoy en día, a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento pueden ser ejecutadas de forma nativa en un servidor de Arquitectura z de IBM®, o alternativamente en máquinas que ejecutan otras arquitecturas. Pueden ser emuladas en servidores centrales existentes y del futuro de IBM® y en otras máquinas de IBM® (por ejemplo, servidores de Sistemas de Energía y Servidores de Sistema x). Pueden ser ejecutadas en máquinas que corren en Linux o en una amplia variedad de máquinas que utilizan hardware fabricado por IBM®, Intel®, AMD, y otros. Además de la ejecución en ese hardware bajo una Arquitectura z, el Linux puede ser utilizado también como unas máquinas que utilizan emulación mediante Hércules, UMX, o FSI (Fundamental Software, Inc.), donde la ejecución generalmente es en un modo de emulación. En el modo de emulación, el software de emulación es ejecutado por un procesador nativo para emular la arquitectura de un procesador emulado.

35 El procesador nativo ejecuta típicamente software de emulación que comprende bien firmware o bien un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de recuperar y ejecutar instrucciones de la arquitectura del procesador emulado. El software de emulación mantiene un contador de programa emulado para seguir el rastro de límites de la instrucción. El software de emulación puede recuperar una o más instrucciones de máquina emuladas a la vez y convertir una o más instrucciones de máquina emuladas a un grupo correspondiente de instrucciones de máquina nativas para la ejecución mediante el procesador nativo. Estas instrucciones convertidas pueden ser almacenadas en memoria caché de tal manera que se puede conseguir una conversión más rápida. No obstante, el software de emulación ha de mantener las reglas de arquitectura de la arquitectura del procesador emulado de modo que asegure sistemas operativos y aplicaciones escritas para que el procesador emulado opere correctamente. Además, el software de emulación ha de proporcionar recursos identificados por la arquitectura del procesador emulado que incluyen, pero no están limitados a, registros de control, registros de propósito general, registros de coma flotante, función de traducción de dirección dinámica que incluye tablas de segmento y tablas de página por ejemplo, mecanismos de interrupción, mecanismos de cambio de contexto, relojes de Hora del Día (TOD) e interfaces arquitectónicas para subsistemas I/O de tal manera que un sistema operativo o un programa de aplicación designado para ejecutarse sobre el procesador emulado, puede ser ejecutado sobre el procesador nativo que tiene el software de emulación.

Una instrucción específica que es simulada es decodificada, y es llamada una subrutina para realizar la función de la instrucción individual. Una función de software de emulación que emula una función de un procesador emulado es implementada, por ejemplo, en una subrutina de "C" o controlador, o algún otro método para proporcionar un controlador para el hardware específico como estará dentro de la experiencia de los expertos en la técnica después de entender la descripción de la realización preferida. Distintas patentes de emulación de software y hardware que incluyen, pero no están limitados a Cartas de patente U.S. N° 5.551.013, titulada "Multiprocessor for Hardware Emulation", por Beausoleil y col.; y las Cartas de patente U.S. N° 6.009.261, titulada "Preprocessing of Stores Target Routines for Emulating Incompatible Instructions on a Target Processor", por Scalzi y col; y las Cartas de patente U.S. N° 5.574.873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", por Davidian y col.; Y las Cartas de patente U.S. N° 6.308.255, titulada "Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to in a System", por Gorishek y col.; y las Cartas de patente U.S. N° 6.463.582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", por Lethin y col.; y las Cartas de patente U.S. N° 5.790.825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions", por Eric Traut, cada una de las cuales es por lo tanto incorporada en este documento por referencia en su totalidad; y muchas otras, ilustran una variedad de modos conocidos para conseguir la emulación de un formato de instrucción diseñado arquitectónicamente para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica.

En la fig. 18, se proporciona un ejemplo de sistema informático 5092 anfitrión emulado que emula un sistema informático 5000' anfitrión de una arquitectura de anfitrión. En el sistema informático 5092 anfitrión emulado, el procesador anfitrión 5091 (CPU) es un procesador anfitrión emulado (o procesador anfitrión virtual) y comprende un procesador de emulación 5093 que tiene una arquitectura de conjunto de instrucciones nativas diferentes que las del procesador 5091 del sistema informático anfitrión 5000'. El sistema informático 5092 anfitrión emulado tiene la memoria 5094 accesible para el procesador de emulación 5093. En la realización ejemplar, la memoria 5094 está particionada en una porción de memoria 5096 de ordenador anfitrión y una porción de rutina de emulación 5097. La memoria 5096 del ordenador anfitrión está disponible para programas del ordenador 5092 anfitrión emulado de acuerdo con la arquitectura informática de anfitrión. El procesador de emulación 5093 ejecuta instrucciones nativas de un conjunto de instrucciones arquitectónicas de una arquitectura distinta del procesador emulado 5091, las instrucciones nativas obtenidas a partir de la memoria 5097 de rutinas de emulación, y puede acceder a una instrucción anfitrión para la ejecución desde un programa en la memoria 5096 de ordenador anfitrión empleando una o más instrucciones obtenidas en una rutina de secuencia y acceso/decodificación que pueden decodificar las instrucciones de anfitrión accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de anfitrión accedida. Otras instalaciones que son definidas para la arquitectura del sistema informático anfitrión 5000' pueden ser emuladas por rutinas de instalaciones arquitectónicas, que incluyen tales instalaciones como registros de propósito general, registros de control, traducción de dirección dinámica y soporte del subsistema I/O y memoria caché del procesador, por ejemplo. Las rutinas de emulación pueden beneficiarse de las funciones disponibles en el procesador de emulación 5093 (tales como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. Se pueden proporcionar también hardware especial y motores de descarga para ayudar al procesador 5093 a emular la función del sistema informático anfitrión 5000'.

La terminología utilizada en este documento tiene el propósito de describir realizaciones particulares solamente y no pretende ser limitativa. Como se ha utilizado en este documento, las formas singulares "un/una", "un/una" y "el/la" están destinadas a incluir las formas plurales también, a menos que el contexto claramente indique lo contrario. Se entenderá además que los términos "comprende" y/o "comprendiendo", cuando son utilizados en esta memoria, especifican la presencia de características, números enteros, pasos, operaciones, elementos, y/o componentes declarados, pero no impiden la presencia o adición de una o más características, números enteros, pasos, operaciones, elementos, componentes y/o grupos de los mismos.

Las estructuras, materiales, acciones, y equivalentes correspondientes de todos los medios o pasos más elementos de función en las reivindicaciones posteriores, si los hay, pretenden incluir cualquier estructura, material, o acción para realizar la función en combinación con otros elementos reivindicados como se ha reivindicado específicamente. La descripción de una o más realizaciones ha sido presentada con propósitos de ilustración y descripción, pero no pretende ser exhaustiva o limitada a la forma descrita. Muchas modificaciones y variaciones serán evidentes para los expertos ordinarios en la técnica. La realización fue elegida y descrita con el fin de explicar mejor distintos aspectos y la aplicación práctica, y permitir a otros expertos ordinarios en la técnica entender las distintas realizaciones con distintas modificaciones que son adecuadas al uso particular contemplado.

REIVINDICACIONES

- 5 1. Un producto de programa informático que comprende un medio de almacenamiento legible por ordenador que almacena instrucciones que se ejecutan sobre un sistema informático que comprende una memoria (5002) y un procesador (5001) para llevar a cabo un método que comprende:
- obtener, mediante el procesador, una instrucción de máquina para la ejecución, siendo definida la instrucción de máquina para ejecución por ordenador de acuerdo con una arquitectura informática, comprendiendo la instrucción de máquina (700):
- un código de operación (702a) para especificar una operación de almacenamiento no transaccional;
- 10 un campo (704) para especificar un registro, en donde los contenidos del registro son un primer operando; y al menos un campo (706, 708, 710, 712) para especificar una dirección del segundo operando, sirviendo la dirección del segundo operando para designar una ubicación en la memoria; y
- ejecutar, mediante el procesador, la instrucción de máquina, comprendiendo la ejecución:
- 15 almacenar de manera no transaccional el primer operando en la ubicación especificada por la dirección del segundo operando, en donde el primer operando almacenado en la ubicación es retenido a pesar de una anulación de una transacción asociada con la instrucción de máquina, y caracterizado por que el almacenamiento no transaccional es retrasado hasta un final del modo de ejecución transaccional del procesador.
- 20 2. El producto de programa informático de la reivindicación 1, en donde el final del modo de ejecución transaccional resulta de un final de una transacción más exterior asociada con la instrucción de máquina o una condición de anulación.
3. El producto de programa informático de la reivindicación 1, en donde el método comprende además:
- determinar si el procesador está en modo de ejecución transaccional;
- 25 basado en que el procesador está en modo de ejecución transaccional, determinar si la transacción es una transacción restringida que está sujeta a un conjunto de limitaciones que comprenden: un número limitado de instrucciones que pueden ser ejecutadas; un número limitado de operandos de almacenamiento que pueden ser accedidos; y la transacción está limitada a un único nivel, o una transacción no restringida que no está sometida al conjunto de limitaciones; y
- 30 basado en que la transacción es una transacción no restringida, continuar la ejecución de la instrucción de máquina.
4. El producto de programa informático de la reivindicación 3, en donde basado en que la transacción es una transacción restringida, proporcionar una excepción de programa y terminar la ejecución de la instrucción de máquina.
- 35 5. El producto de programa informático de la reivindicación 3, en donde basado en que el procesador no está en modo de ejecución transaccional, ejecutar la instrucción de máquina como una instrucción de almacenamiento.
6. El producto de programa informático de la reivindicación 1, en donde la dirección del segundo operando está formada de una combinación de contenidos de un registro especificado por un campo índice de la instrucción de máquina, contenidos de un registro especificado por un campo base de la instrucción de máquina, y contenidos de al menos un campo de desplazamiento.
- 40 7. El producto de programa informático de la reivindicación 1, en donde la instrucción de máquina comprende una instrucción de almacenamiento no transaccional, y en donde el método comprende además:
- iniciar la transacción basada en la ejecución de una instrucción de comienzo de transacción, retrasando de manera efectiva la transacción de almacenamientos transaccionales consignados a la memoria principal hasta completar una transacción seleccionada;
- 45 consignar los almacenamientos transaccionales a la memoria principal basándose en la ejecución de una instrucción final de transacción que completa la ejecución de la transacción seleccionada;
- descartar los almacenamientos transaccionales basándose en la anulación que finaliza la ejecución de la transacción seleccionada; y
- 50 en donde el almacenamiento de manera no transaccional es realizado independientemente de la consignación o del descarte.

8. El producto de programa informático de la reivindicación 1, en donde los contenidos del registro especificado por la instrucción de máquina son específicos de usuario.
9. El producto de programa informático de la reivindicación 1, en donde la transacción es una transacción no restringida en una anidación de transacciones no restringidas.
- 5 10. Un sistema informático para ejecutar una instrucción dentro de un entorno informático, comprendiendo dicho sistema informático:
- una memoria (5002); y
- un procesador (5001) en comunicación con la memoria, en donde el sistema informático está configurado para realizar un método, comprendiendo dicho método:
- 10 obtener, por el circuito de tratamiento, una instrucción de máquina (700) para su ejecución, siendo definida la instrucción de máquina para la ejecución por ordenador según una arquitectura informática, comprendiendo la instrucción de máquina:
- un código de operación (702a) para especificar una operación de almacenamiento no transaccional;
- 15 un campo (704) para especificar un registro, en donde los contenidos del registro son un primer operando; y al menos un campo (706, 708, 710, 712) para especificar una dirección del segundo operando, sirviendo la dirección del segundo operando para
- designar una ubicación en la memoria; y
- ejecutar, mediante el procesador, la instrucción de máquina, comprendiendo la ejecución:
- 20 almacenar de manera no transaccional el primer operando en la ubicación especificada por la dirección del segundo operando, en donde el primer operando almacenado en la ubicación es retenido a pesar de una anulación de una transacción asociada con la instrucción de máquina, y caracterizado por que el almacenamiento no transaccional es retrasado hasta un final del modo de ejecución transaccional del procesador.
11. El sistema informático de la reivindicación 10, en donde el método comprende además:
- 25 determinar si el procesador está en modo de ejecución transaccional;
- basándose en que el procesador está en el modo de ejecución transaccional, determinar si la transacción es una transacción restringida que está sometida a un conjunto de limitaciones que comprenden: que puede ser ejecutado un número limitado de instrucciones; que puede accederse a un número limitado de operandos de almacenamiento; y que la transacción está limitada a un solo nivel, o una transacción no restringida que no está sometida al conjunto de limitaciones; y
- 30 basándose en que la transacción es una transacción no restringida, continuar con la ejecución de la instrucción de máquina.
12. El sistema informático de la reivindicación 11, en donde basándose en que la transacción es una transacción restringida, proporcionar una excepción de programa y finalizar la ejecución de la instrucción de máquina.
- 35 13. El sistema informático de la reivindicación 11, en donde basándose en que el procesador no está en modo de ejecución transaccional, ejecutar la instrucción de máquina como una instrucción de almacén.
14. El sistema informático de la reivindicación 10, en donde los contenidos del registro especificado por la instrucción de máquina son especificados por el usuario, y en donde la dirección del segundo operando está formada de una combinación de contenidos de un registro especificado por un campo índice de la instrucción de máquina, contenidos de un registro especificado por un campo base de la instrucción de máquina, y contenidos de al menos un campo de desplazamiento.
- 40 15. El sistema informático de la reivindicación 10, en donde la transacción es una transacción no restringida en una anidación de transacciones no restringidas.
16. Un método para ejecutar una instrucción dentro de un entorno informático, comprendiendo dicho método:
- 45 obtener, mediante un procesador (5001), una instrucción de máquina (700) para ejecución, siendo definida la instrucción de máquina para ejecución por ordenador según una arquitectura informática, comprendiendo la instrucción de máquina:
- un código de operación (702a) para especificar una operación de almacenamiento no transaccional;

un campo (704) para especificar un registro, en donde los contenidos del registro son un primer operando; y al menos un campo (706, 708, 710, 712) para especificar una dirección del segundo operando, la dirección del segundo operando para

designar una ubicación en la memoria (5002); y

5 ejecutar, mediante el procesador, la instrucción de máquina, comprendiendo la ejecución:

almacenar de manera no transaccional el primer operando en la ubicación especificada por la dirección del segundo operando, en donde el primer operando almacenado en la ubicación es retenido a pesar de una anulación de una transacción asociada con la instrucción de máquina, y caracterizado por que el almacenamiento no transaccional es retrasado hasta un final del modo de ejecución transaccional del procesador.

10 17. El método de la reivindicación 16, que comprende además:

determinar si el procesador está en modo de ejecución transaccional;

basándose en que el procesador está en modo de ejecución transaccional, determinar si la transacción es una transacción restringida que está sometida a un conjunto de limitaciones que comprende: que puede ser ejecutado un número limitado de instrucciones; que puede accederse a un número limitado de operandos de almacenamiento; y que la transacción está limitada a un solo nivel, o que una transacción no restringida no está sometida al conjunto de limitaciones; y

15 basándose en que la transacción es una transacción no restringida, continuar la ejecución de la instrucción de máquina.

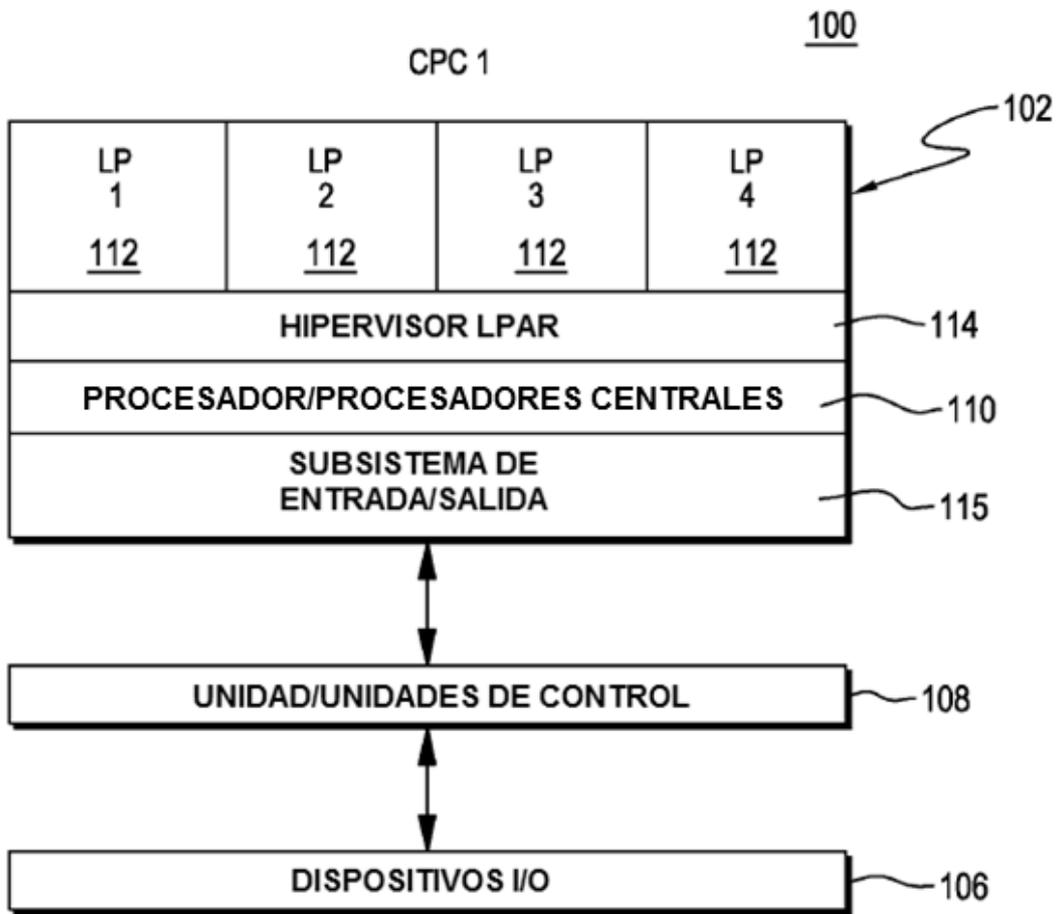


FIG. 1

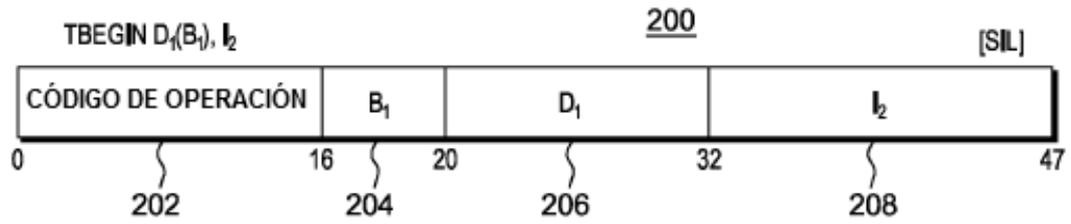


FIG. 2A

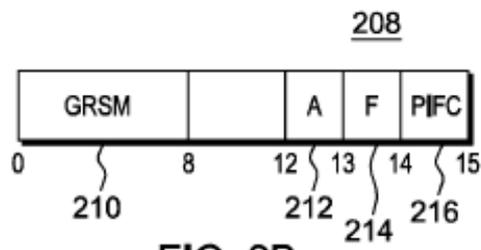


FIG. 2B

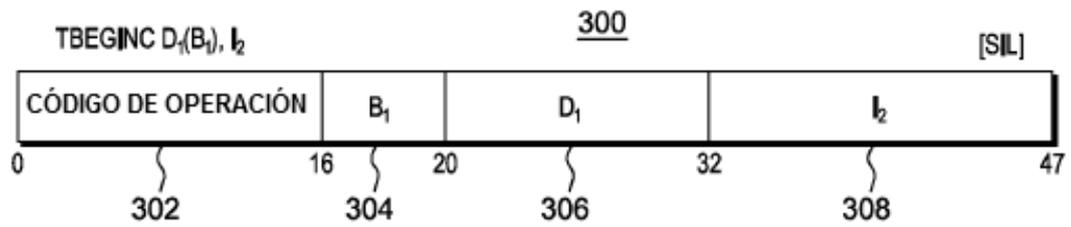


FIG. 3A

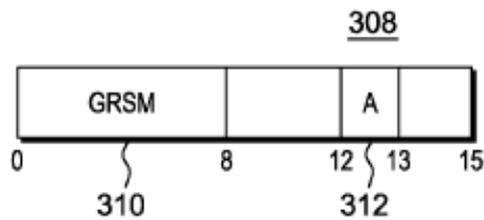


FIG. 3B



FIG. 4

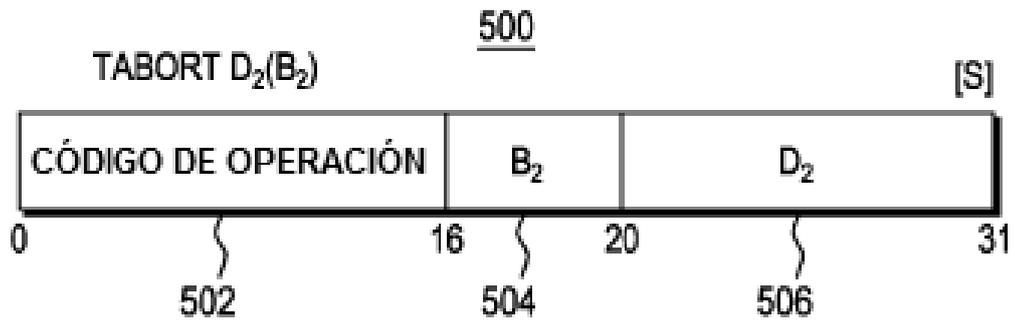


FIG. 5

TRANSACTIONS ANIDADAS

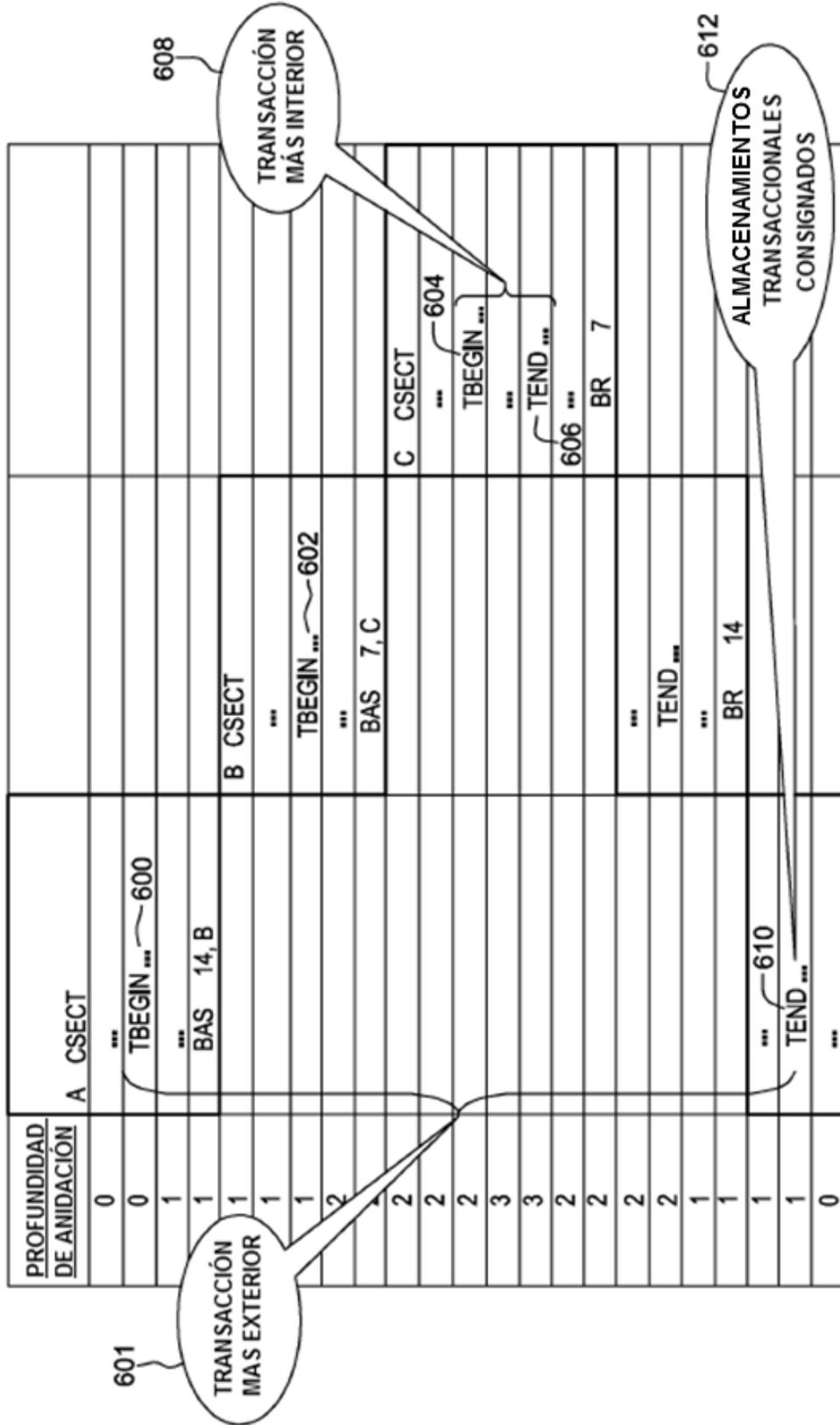


FIG. 6

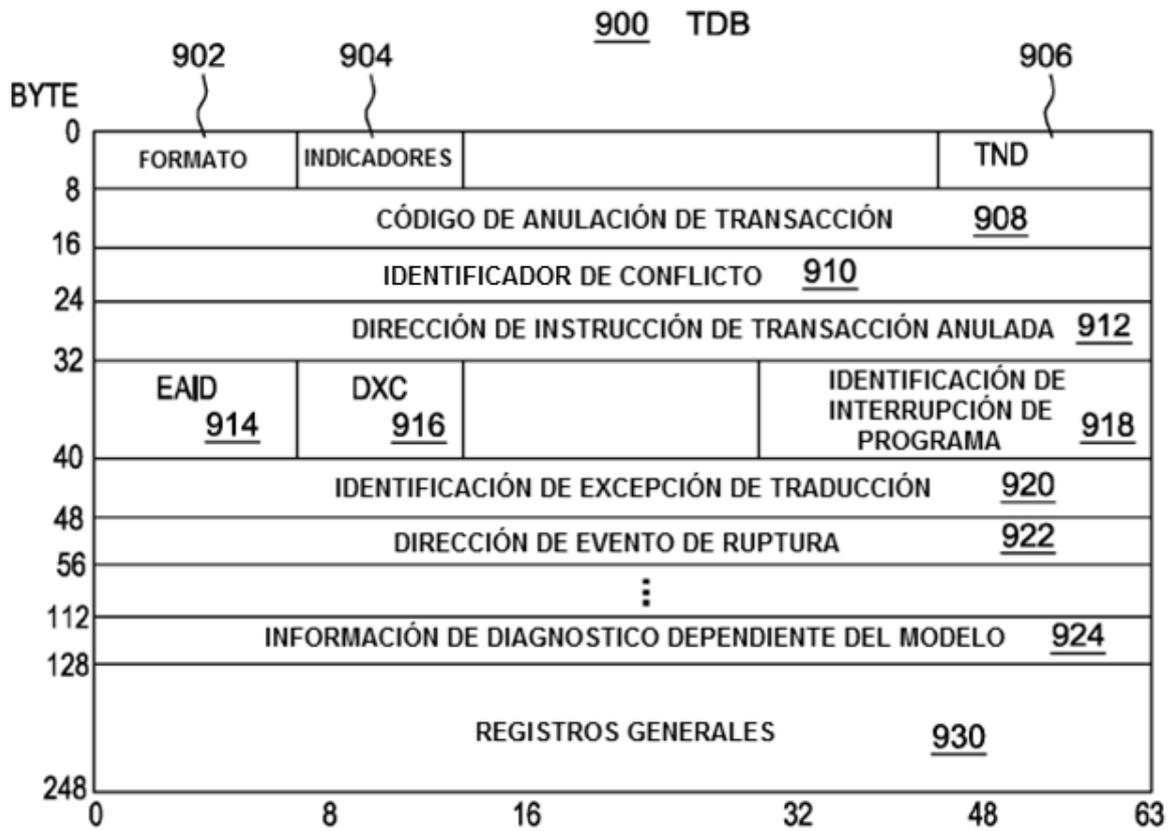


FIG. 9

CÓDIGO	RAZÓN DE ANULACIÓN	CC SET
2	INTERRUPCIÓN EXTERNA	2
4	INTERRUPCIÓN DE PROGRAMA (SIN FILTRAR)	2 OR 3 +
5	INTERRUPCIÓN DE COMPROBACIÓN DE MÁQUINA	2
6	INTERRUPCIÓN I/O	2
7	DESBORDAMIENTO DE RECUPERACIÓN	2 OR 3
8	DESBORDAMIENTO DE ALMACENAMIENTO	2 OR 3
9	CONFLICTO DE RECUPERACIÓN	2
10	CONFLICTO DE ALMACENAMIENTO	2
11	INSTRUCCIÓN RESTRINGIDA	3
12	CONDICIÓN DE EXCEPCIÓN DE PROGRAMA (FILTRADO)	3
13	PROFUNDIDAD DE ANIDACIÓN EXCEDIDA	3
14	RECUPERACIÓN DE MEMORIA CACHÉ RELACIONADA	2 OR 3
15	ALMACENAMIENTO DE MEMORIA CACHÉ RELACIONADA	2 OR 3
16	OTRA MEMORIA CACHÉ	2 OR 3
255	CONDICIÓN MISCELÁNEA	2 OR 3
>255	INSTRUCCIÓN DE TABORT	2 OR 3
‡	NO PUEDE SER DETERMINADO; NO TDB ALMACENADO	1
<p>EXPLICACIÓN:</p> <p>+ EL CÓDIGO DE CONDICIÓN ESTÁ BASADO EN EL CÓDIGO DE INTERRUPCIÓN</p> <p>‡ ESTA SITUACIÓN OCURRE CUANDO SE ANULA UNA TRANSACCIÓN, PERO LA TDB SE HA VUELTO INACCESIBLE DESPUÉS DE LA EJECUCIÓN CON ÉXITO DE LA INSTRUCCIÓN DE TBEGIN MÁS EXTERIOR, NO ES ALMACENADO EL TDB ESPECIFICADO POR TBEGIN, Y EL CÓDIGO DE CONDICIÓN ES CONFIGURADO A 1.</p>		

FIG. 10

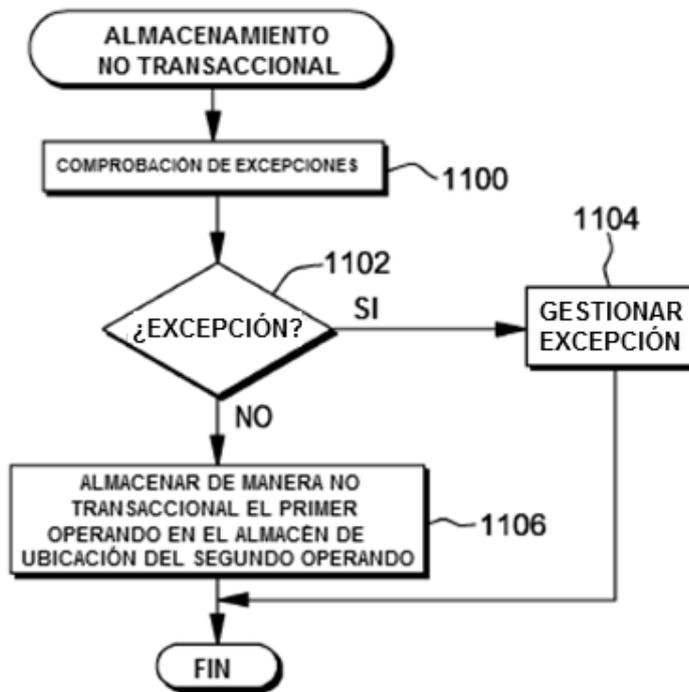


FIG. 11A

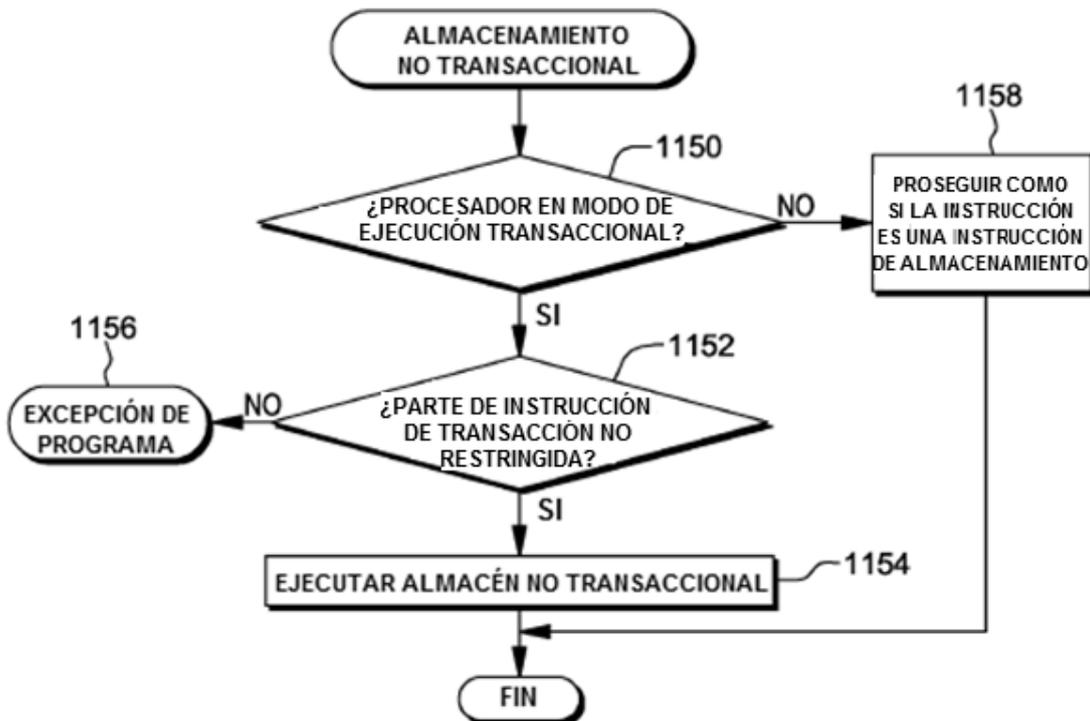


FIG. 11B

PRODUCTO
DE PROGRAMA
INFORMÁTICO
1200



FIG. 12

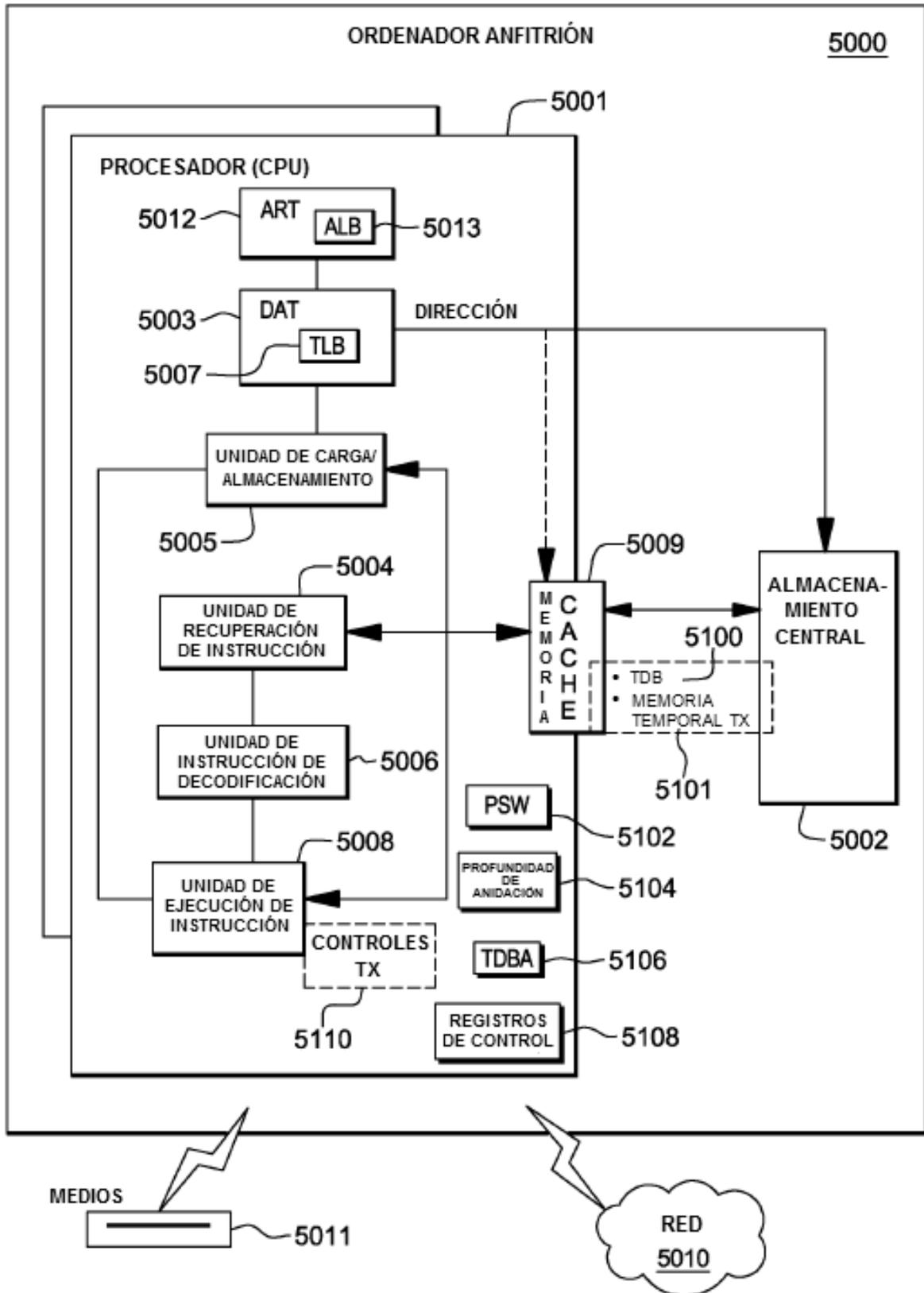


FIG. 13

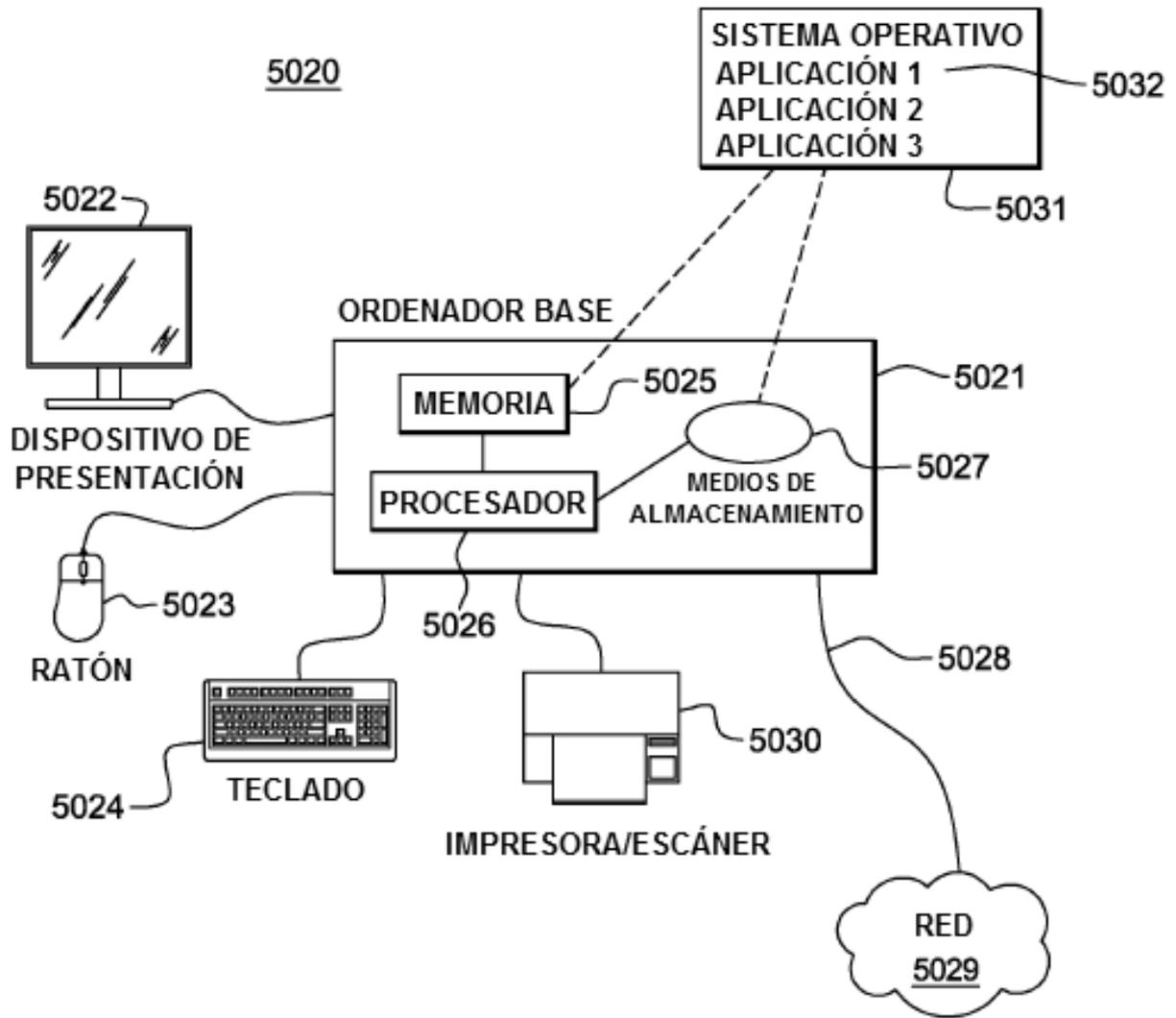


FIG. 14

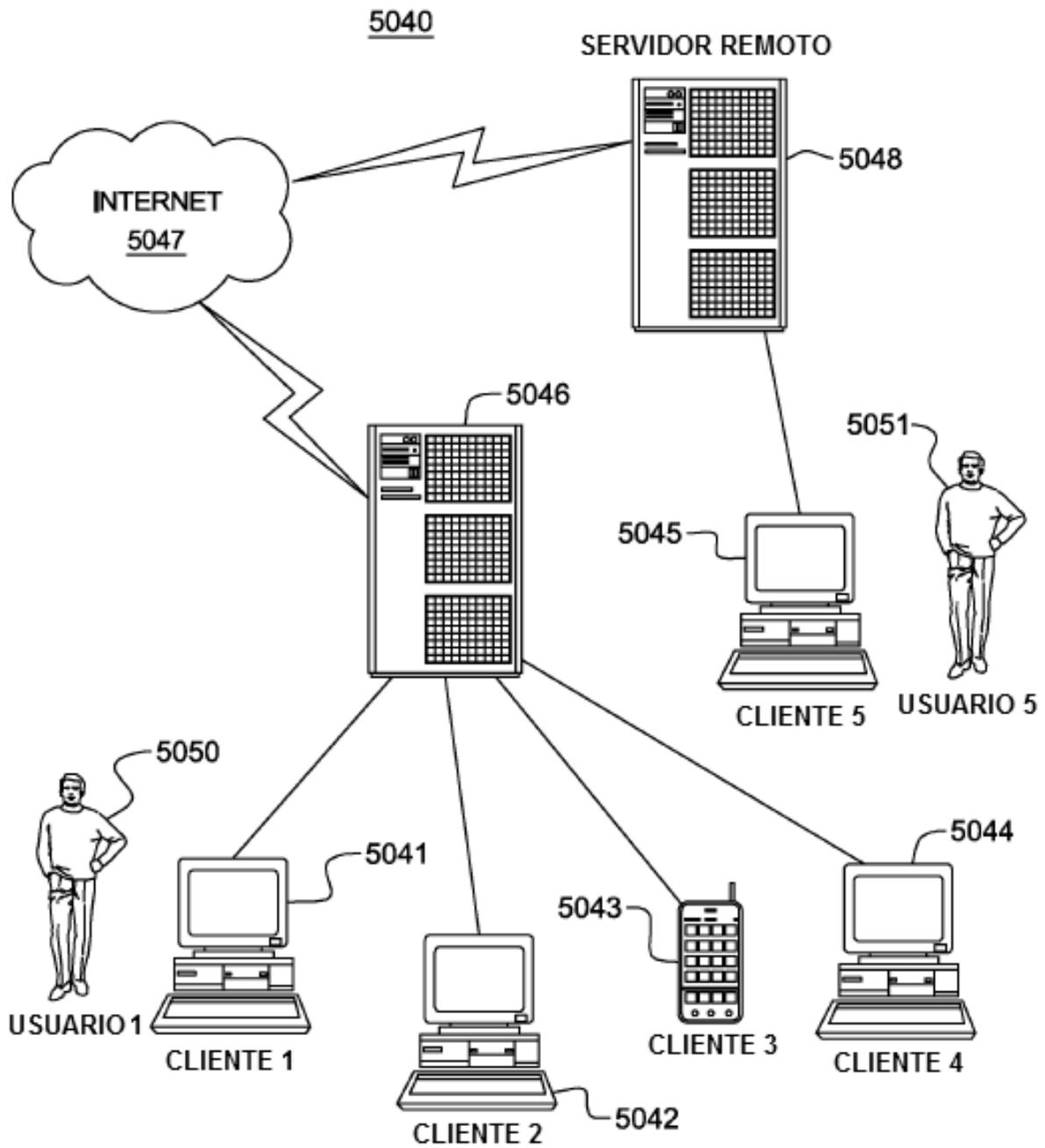


FIG. 15

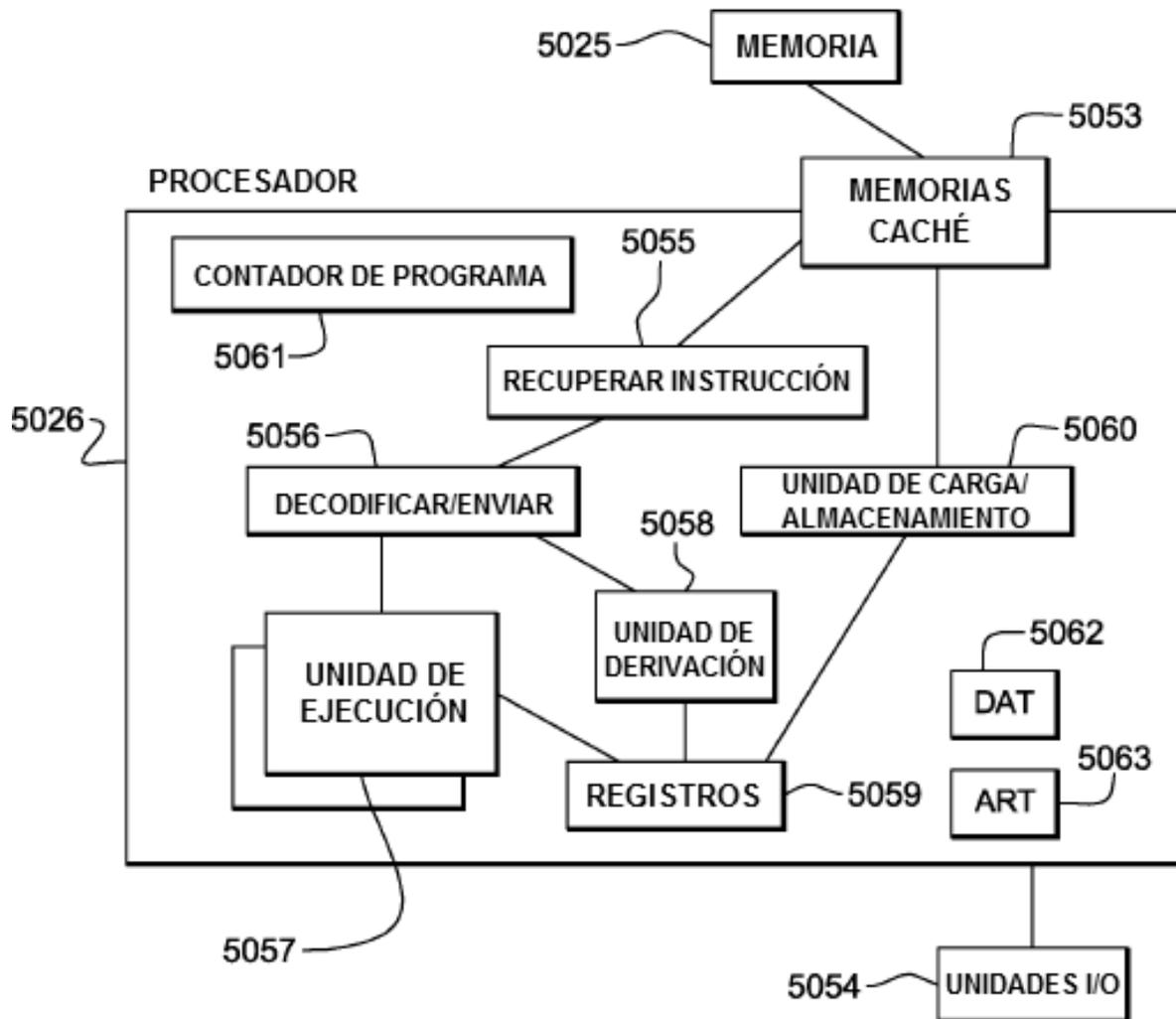


FIG. 16

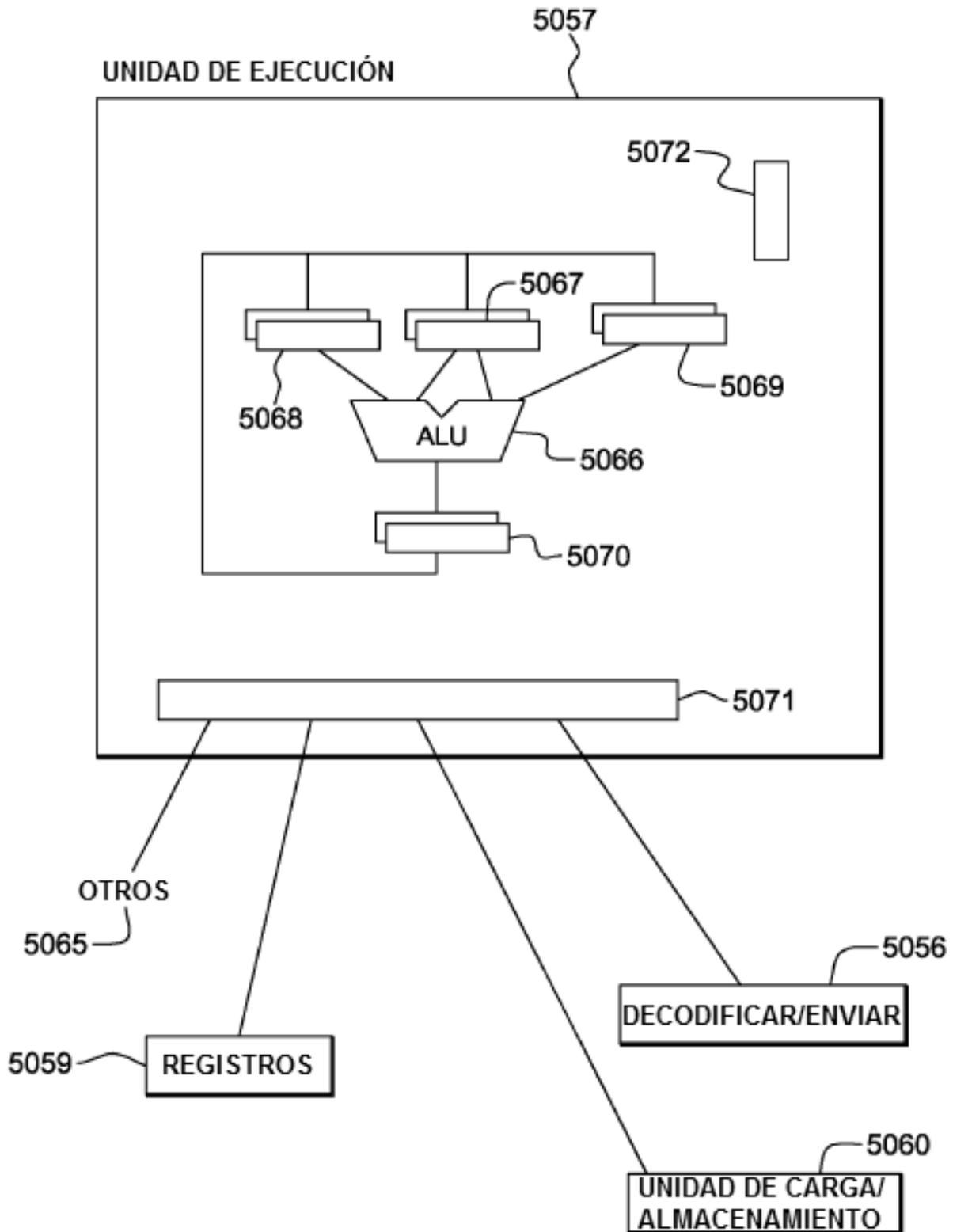


FIG. 17A

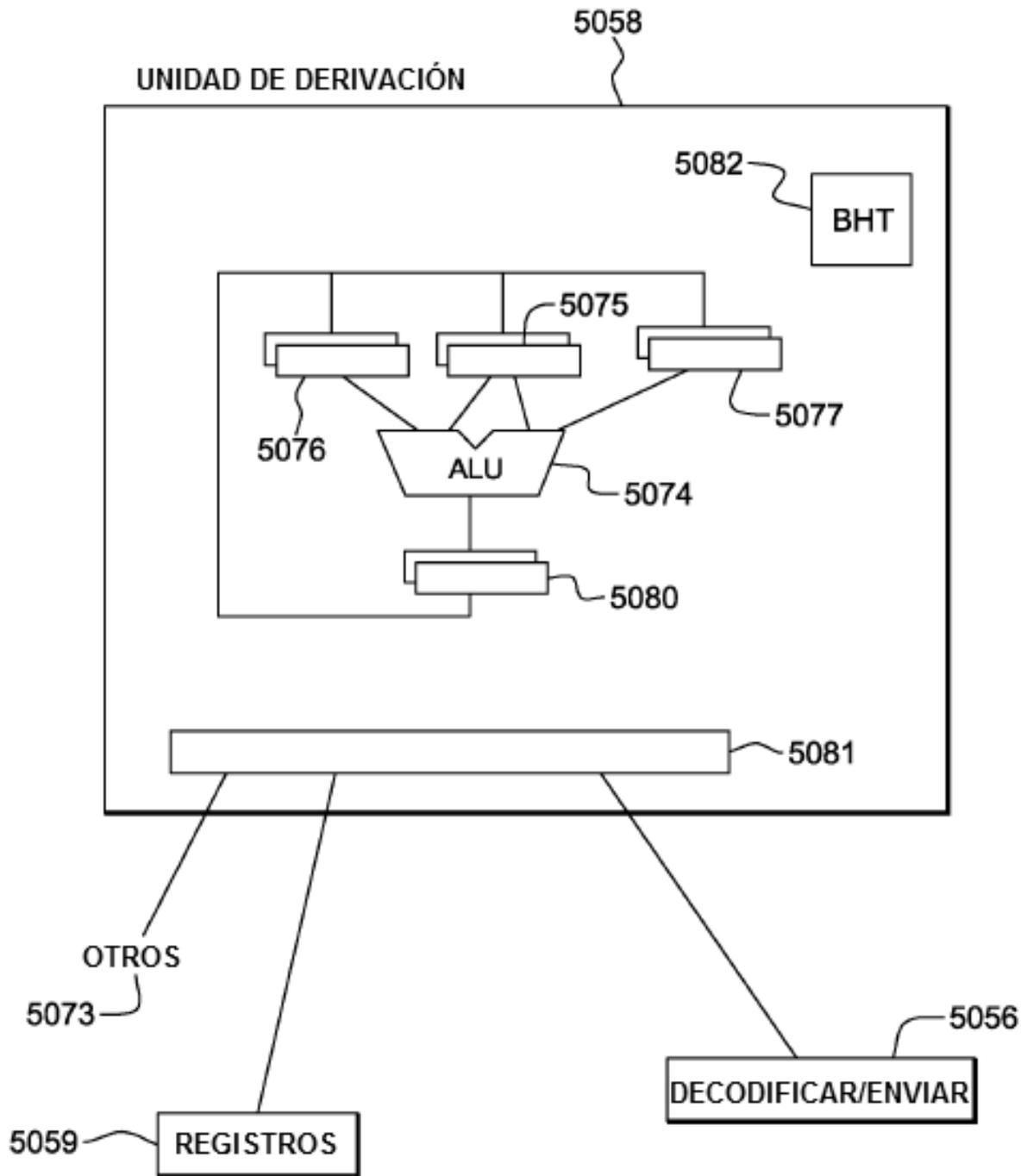


FIG. 17B

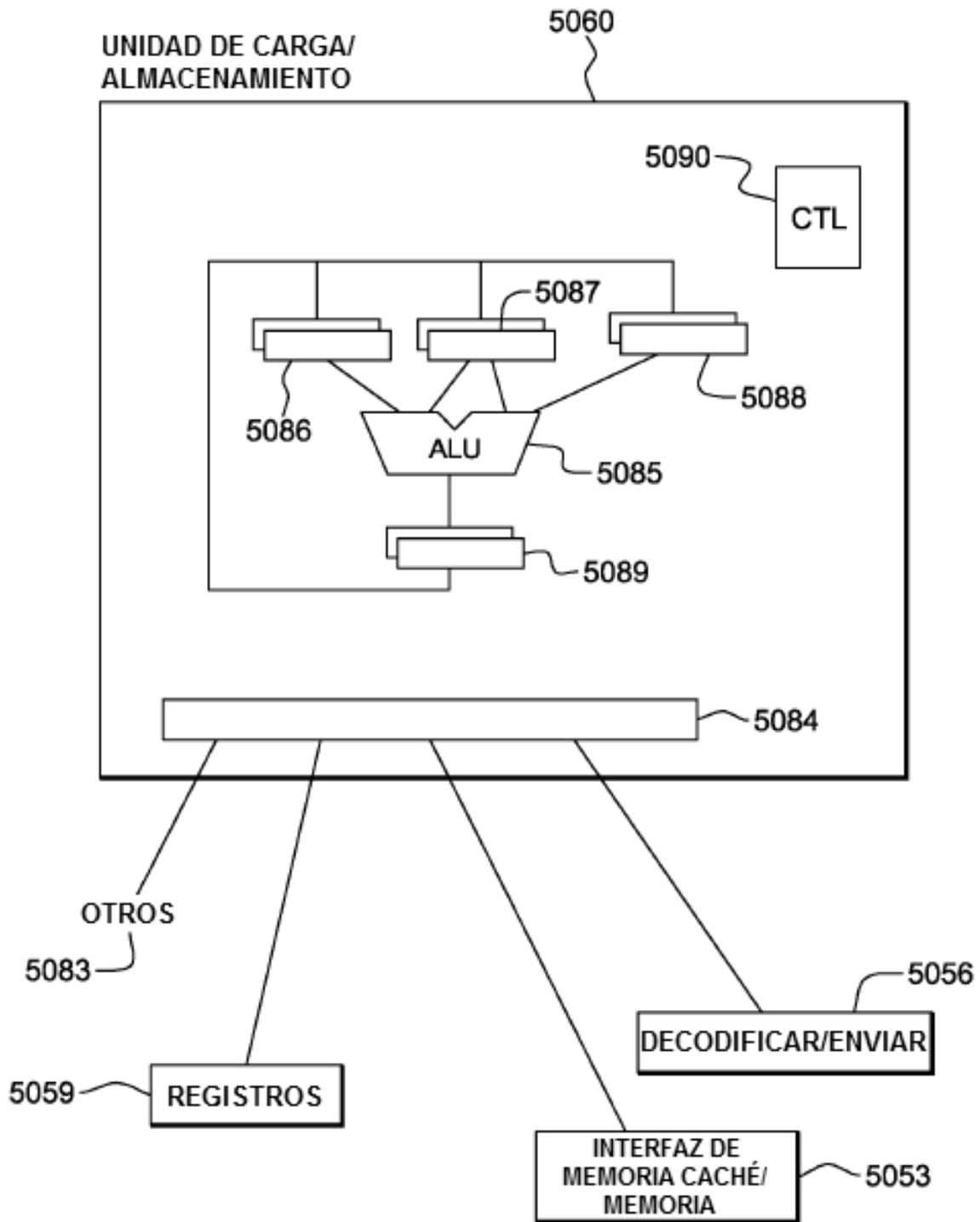


FIG. 17C

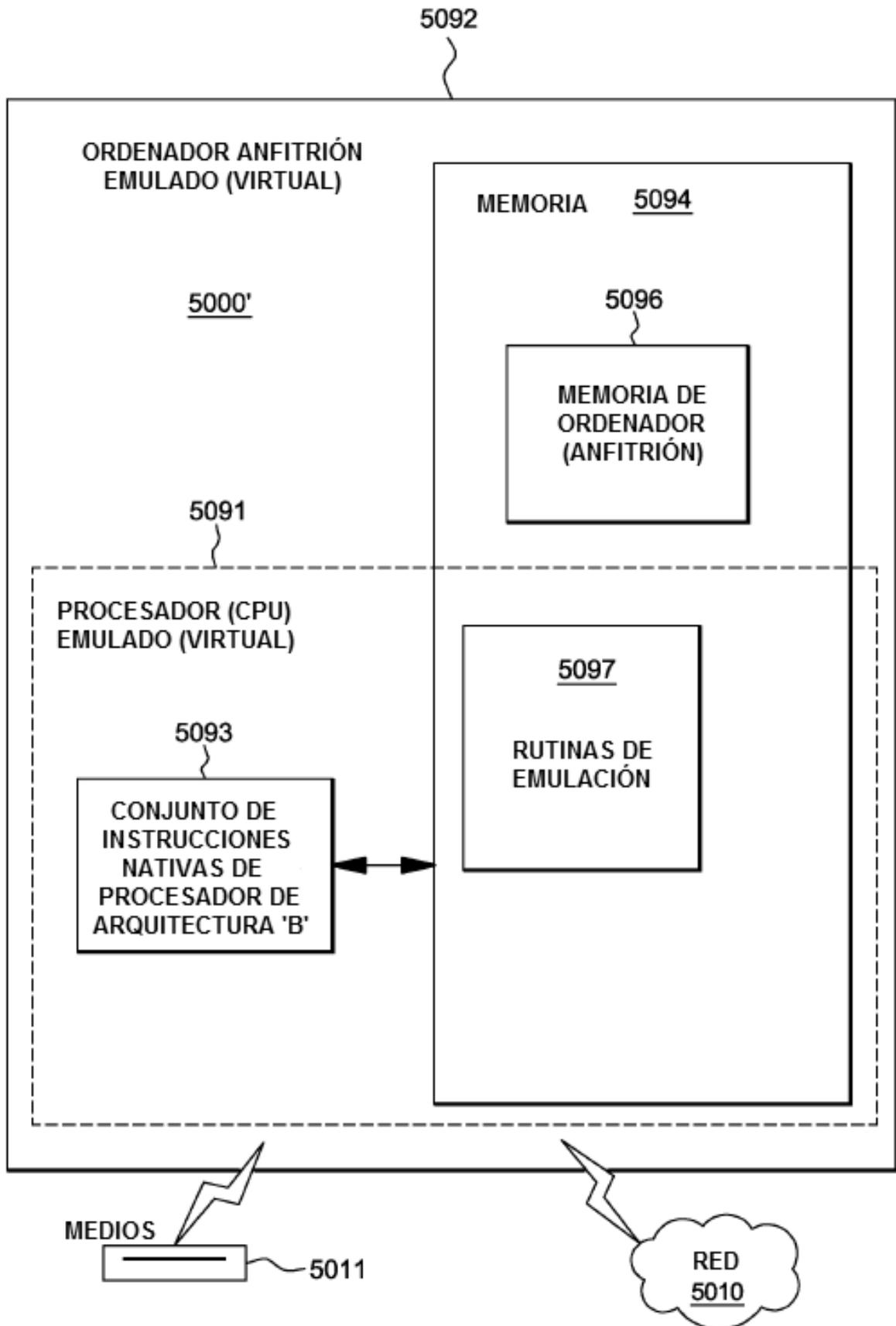


FIG. 18