

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 727 124**

51 Int. Cl.:

**G06F 9/46** (2006.01)

**G06F 9/30** (2008.01)

**G06F 9/38** (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.06.2013 PCT/IB2013/054812**

87 Fecha y número de publicación internacional: **19.12.2013 WO13186721**

96 Fecha de presentación y número de la solicitud europea: **12.06.2013 E 13804476 (3)**

97 Fecha y número de publicación de la concesión europea: **17.04.2019 EP 2862070**

54 Título: **Procesamiento transaccional**

30 Prioridad:

**15.06.2012 US 201213524921**

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

**14.10.2019**

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

**ELZABURU, S.L.P**

**ES 2 727 124 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

Procesamiento transaccional

**Antecedentes**

5 Uno o más aspectos relacionados, en general, con sistemas informáticos de multiprocesamiento, y en concreto, con el procesamiento transaccional dentro de dichos sistemas informáticos.

10 Un desafío permanente en la programación multiprocesador es el de las actualizaciones por las múltiples unidades centrales de procesamiento (CPU) en la misma ubicación de almacenamiento. Muchas instrucciones que actualizan ubicaciones de almacenamiento, incluyendo incluso operaciones lógicas simples, tales como la Y, hacen esto con los múltiples accesos a la ubicación. Por ejemplo, primero, se extrae la ubicación de almacenamiento, y después se actualiza de nuevo el resultado actualizado.

15 Para que las múltiples CPU actualicen de manera segura la misma ubicación de almacenamiento, se serializa el acceso a la ubicación. Una instrucción, la instrucción de PRUEBA y CONFIGURACIÓN introducida con la arquitectura S/360 anteriormente ofrecida por la Corporación de Máquinas de Negocios Internacionales, proporciona una actualización interconectada de una ubicación de almacenamiento. Una actualización interconectada significa que, según es observado por otras CPU y el subsistema de entrada/salida (I/O) (por ejemplo, el subsistema de canal), todo el acceso de almacenamiento de la instrucción parece producirse de manera atómica. Después, la arquitectura S/370 ofrecida por la Corporación de Máquinas de Negocios Internacionales introdujo las instrucciones COMPARACIÓN E INTERCAMBIO y COMPARACIÓN DOBLE E INTERCAMBIO que proporcionan unos medios más sofisticados de realizar la actualización interconectada, y permiten la implementación de lo que comúnmente se conoce como una palabra de bloqueo (o semáforo). Las instrucciones añadidas recientemente han proporcionado capacidades adicionales de actualización interconectada, incluyendo la COMPARACIÓN E INTERCAMBIO, y la COMPARACIÓN Y PURGA, y la COMPARACIÓN E INTERCAMBIO Y ALMACENAMIENTO. Sin embargo, todas estas funciones proporcionan interconexión para sólo una ubicación de almacenamiento única.

25 Técnicas de programa más complejas pueden requerir la actualización interconectada de múltiples ubicaciones de almacenamiento, tales como añadir un elemento a una lista doblemente enlazada. En dicha operación, ha de aparecer tanto un puntero hacia delante como hacia atrás para estar actualizada de manera simultánea, según es observado por otras CPU y el subsistema I/O. Para efectuar dicha actualización múltiple de ubicaciones, se fuerza al programa a usar un punto de serialización único, separado, tal como una palabra de bloqueo. Sin embargo, las palabras de bloqueo pueden proporcionar un nivel mucho más alto de serialización del que se garantiza, por ejemplo, las palabras de bloqueo pueden serializar una cola entera de millones de elementos, incluso aunque sólo se estén actualizando dos elementos. El programa puede estructurar los datos para usar una serialización más fina (por ejemplo, una jerarquía de puntos de bloqueo), pero esto introduce problemas adicionales, tales como las posibles situaciones de bloqueo 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.

35 Además de lo anterior, existen numerosos escenarios en los que un programa puede ejecutar una secuencia de instrucciones que pueden o no resultar en una condición de excepción. Si no se produce ninguna 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 dicha ejecución en, por ejemplo, una ejecución especulativa, parcial de una función, y/o en la re-secuenciación de la comprobación del puntero nulo.

40 En entornos de sistemas operativos clásicos, tales como el z/OS y sus predecesores ofrecidos por la Corporación de Máquinas de Negocios Internacionales, 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 normalmente termina de manera anormal el programa para las excepciones que el sistema operativo no está preparado a manejar. Establecer y explotar dicho entorno es costoso y complicado.

45 El documento US 2010/023703 A1 describe un método para controlar una transacción en un entorno informático y describe una instrucción ESPECULACIÓN y una instrucción INICIO. La instrucción ESPECULACIÓN puede iniciar una transacción en modo de retroceso y la instrucción INICIO puede iniciar una transacción en modo almacenamiento condicional. En el modo de retroceso, en caso de una cancelación, el control se transfiere a un punto de control registrado, que puede ser una instrucción que sigue de manera inmediata a la instrucción ESPECULACIÓN. En el modo de almacenamiento condicional, la transacción puede continuar, pero sin realizar las operaciones de almacenamiento transaccional.

**Breve compendio**

55 Las deficiencias de la técnica anterior se superan y se proporcionan ventajas a través de la provisión de un método como se reivindica en la reivindicación 1, y un sistema correspondiente como se reivindica en la reivindicación 12 y un programa informático como se reivindica en la reivindicación 13.

Se realizan características y ventajas adicionales. Otras realizaciones y aspectos se describen en detalle.

**Breve descripción de las diversas vistas de los dibujos**

Se señalan y se reivindican de manera distintiva uno o más aspectos como ejemplos en las reivindicaciones a la conclusión de la especificación. El anterior y otros objetivos, características, y ventajas serán evidentes a partir de la siguiente descripción detallada tomada en conjunción con los dibujos adjuntos en los que:

- 5 La FIG. 1 representa una realización de un entorno informático;
- La FIG. 2A representa un ejemplo de una instrucción de Inicio de Transacción (TINICIO);
- La FIG. 2B representa una realización de detalles adicionales de un campo de la instrucción TINICIO de la FIG. 2A;
- La FIG. 3A representa un ejemplo de un Inicio de Transacción restringido (TINICIOOC)
- 10 La FIG. 3B representa una realización de detalles adicionales de un campo de la instrucción TINICIOOC de la FIG. 3A;
- La FIG. 4 representa un ejemplo de una instrucción de Fin de Transacción (TFIN).
- La FIG. 5 representa un ejemplo de una instrucción de Cancelación de Transacción (TCANCELACIÓN);
- La FIG. 6 representa un ejemplo de transacciones anidadas;
- 15 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 LA PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES (ETND);
- La FIG. 9 representa un ejemplo de un bloque de diagnóstico de transacciones;
- La FIG. 10 representa razones de ejemplo para cancelar, junto con los códigos de cancelación asociados y los códigos de condición;
- 20 La FIG. 11 representa una realización de la lógica asociada con la ejecución de una instrucción TINICIOOC;
- La FIG. 12 representa una realización de la lógica asociada con la ejecución de una instrucción TINICIO;
- La FIG. 13 representa una realización de la lógica asociada con la ejecución de una instrucción TFIN;
- La FIG. 14 representa una realización de la lógica asociada con el procesamiento de la cancelación de la transacción;
- 25 La FIG. 15A representa una realización de la lógica asociada con el procesamiento transaccional;
- La FIG. 15B representa una realización de la lógica asociada con el procesamiento de reanudación después de una cancelación de una transacción no restringida;
- Las FIG. 16A-16B representan un ejemplo de inserción de un elemento de cola en una lista doblemente enlazada de elementos de cola;
- 30 La FIG. 17 representa una realización de un producto de programa informático;
- La FIG. 18 representa una realización de un sistema informático servidor;
- La FIG. 19 representa un ejemplo adicional de un sistema informático;
- La FIG. 20 representa otro ejemplo de un sistema informático que comprende una red informática;
- 35 La FIG. 21 representa una realización de diversos elementos de un sistema informático;
- La FIG. 22A representa una realización de la unidad de ejecución del sistema informático de la FIG. 21;
- La FIG. 22B representa una realización de la unidad de salto del sistema informático de la FIG. 21;
- La FIG. 22C representa una realización de la unidad de carga/almacenamiento del sistema informático de la FIG. 21;
- y
- 40 La FIG. 23 representa una realización de un sistema informático de servidor emulado.

### Descripción detallada

De acuerdo con un aspecto, se proporciona una función de ejecución transaccional (TX). Esta función proporciona un procesamiento transaccional para instrucciones, y en una o más realizaciones, ofrece diferentes modos de ejecución, como se describe más adelante, así como niveles anidados de procesamiento transaccional.

5 La función de ejecución transaccional introduce un estado de CPU llamado el modo de ejecución transaccional (TX). A continuación de un reinicio de la CPU, la CPU no está en el modo TX. La CPU entra al modo TX mediante una instrucción de INICIO DE TRANSACCIÓN. La CPU abandona el modo TX mediante ya sea (a) una instrucción de FIN DE TRANSACCIÓN más externa (más detalles sobre interna y externa más adelante), o (b) siendo cancelada la transacción. Mientras que en el modo TX, los accesos al almacenamiento de la CPU parecen ser bloques  
10 concurrentes según son observados por otra CPU y el subsistema de I/O. Los accesos al almacenamiento son bien (a) comprometidos a almacenar cuando finaliza la transacción más externa sin cancelar (esto es, por ejemplo, las actualizaciones hechas en la caché o la memoria intermedia local a la CPU se propagan y almacenan en memoria real y son visibles a otras CPU), o (b) descartados si se cancela la transacción.

15 Las transacciones se pueden anidar, Esto es, mientras que la CPU está en el modo TX, puede ejecutar otra instrucción de INICIO DE TRANSACCIÓN. La instrucción que provoca que la CPU entre en el modo TX es llamada la instrucción de INICIO DE TRANSACCIÓN más externa; de manera similar, se dice que el programa está en la transacción más externa. Las ejecuciones posteriores del INICIO 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 dependiendo del modelo. Una instrucción de  
20 EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES devuelve el valor de profundidad de anidación actual, y en una realización adicional, puede devolver un valor de profundidad de anidación máximo. Esta técnica usa un modelo llamado "anidación plana" en la que una condición de cancelación en cualquier profundidad de anidación provoca que todos los niveles de la transacción sean cancelados, y el control se devuelva a la instrucción que sigue a la instrucción de INICIO DE TRANSACCIÓN más externa.

25 Durante el procesamiento de una transacción, el acceso transaccional hecho por una CPU se dice que está en conflicto con ya sea (a) un acceso transaccional o un acceso no transaccional hecho por otra CPU, o (b) un acceso no transaccional hecho por el subsistema de I/O, si ambos accesos son a cualquier ubicación dentro de la misma línea de caché, y uno o ambos de los accesos es un almacenamiento. En otras palabras, para que la ejecución transaccional sea productiva, no se debe observar la CPU haciendo accesos transaccionales hasta que se  
30 confirmen. Este modelo de programación puede ser altamente efectivo en ciertos entornos, por ejemplo, la actualización de dos puntos en una lista doblemente enlazada de un millón de elementos. Sin embargo, puede ser menos efectivo, si existe una gran disputa de las ubicaciones de almacenamiento que están siendo accedidas de manera transaccional.

35 En un modelo de ejecución transaccional (referido en la presente memoria como una transacción no restringida), cuando se cancela una transacción, el programa puede bien intentar reconducir la transacción con la esperanza de que la condición de cancelación ya no esté presente, o que el programa pueda "retroceder" a una ruta no transaccional equivalente. En otro modelo de ejecución transaccional (referido en la presente memoria como una transacción restringida), una transacción cancelada es automáticamente reconducida por la CPU; en ausencia de violaciones de restricción, se asegura la finalización eventual de la transacción restringida.

40 Al iniciar una transacción, el programa puede especificar diversos controles, tales como (a) qué registros generales son restaurados a sus contenidos originales si se cancela la transacción, (b) si se permite que la transacción modifique el contexto del registro de punto flotante, incluyendo, por ejemplo, los registros de punto flotante y el registro de control de punto flotante, (c) si se permite que la transacción modifique los registros de acceso (AR), y (d) si se han de bloquear ciertas condiciones de excepción de programa de provocar una interrupción. Si se cancela una  
45 transacción no restringida, se puede proporcionar diversa información de diagnóstico. Por ejemplo, la instrucción TINICIO más externa que inicia una transacción no restringida puede designar un bloque de diagnóstico de transacciones (TDB) especificado por el programa. Además, se puede usar también el TDB en el área de prefijo de la CPU o designado por la descripción de estado del servidor si se cancela la transacción debido a una interrupción de programa o una condición que provoca que la ejecución interpretativa finalice, respectivamente.

50 Anteriormente se indican diversos tipos de registros. Estos se explican con más detalle en la presente memoria. Los registros generales se pueden usar como acumuladores en las operaciones de aritmética y lógica general. 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 para abordar múltiples registros generales tener varios registros R. Para algunas  
55 instrucciones, se implica el uso de un registro general específico en lugar de ser designado por un campo R de la instrucción.

Además de su uso como acumuladores en operaciones de aritmética y lógica general, 15 de los 16 registros generales se usan también como direcciones de base y registro de índices en la generación de direcciones. En estos casos, los registros son designados mediante un campo B o un campo X de cuatro bits en la instrucción. Un

valor de cero en el campo B o X especifica que no se ha de aplicar una base o índice, y por tanto, no se ha de designar el registro general 0 como que contiene una dirección de base o índice.

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

Un registro de control de punto flotante (FPC) es un registro de 32 bits que contiene bits de máscara, bits de bandera, un código de excepción de datos, y bits de modo de redondeo, y se usa durante el procesamiento de las operaciones de punto flotante.

Además, en una realización, la CPU tiene 16 registros de control, teniendo cada uno 64 posiciones de bit. Las posiciones de bit en los registros se asignan a funciones concretas en el sistema, tales como de Registro de Eventos de Programa (PER) (discutidas más adelante), y se usan ya sea para especificar que una operación puede tener lugar o para proporcionar información especial requerida por la función. En una realización, para la función transaccional, se usa el CR0 (bits 8 y 9) y el CR2 (bits 61-63), como se describe más adelante.

La CPU tiene, por ejemplo, 16 registros de acceso numerados de 0-15. Un registro de acceso consiste de 32 posiciones de bit que contienen una especificación indirecta de un elemento de control de espacio de direcciones (ASCE). Un elemento de control del espacio de direcciones es un parámetro usado por el mecanismo de traducción de direcciones dinámicas (DAT) para traducir las referencias a un espacio de direcciones 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), una instrucción del campo B, usada para especificar una dirección lógica para el almacenamiento del operando de referencia, designa un registro de acceso, y el elemento de control del espacio de direcciones especificado por el registro de acceso es usado por el DAT para que se haga la referencia. Para algunas instrucciones, se usa un campo R en lugar de un campo B. Las instrucciones se proporcionan 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 1-15 de acceso pueden designar cualquier espacio de direcciones. El registro 0 de acceso designa el espacio de instrucciones principal. Cuando uno de los registros 1-15 de acceso se usa para designar un espacio de direcciones, la CPU determina qué espacio de direcciones se designa traduciendo los contenidos del registro de acceso. Cuando se usa el registro 0 de acceso para designar un espacio de direcciones, la CPU trata el registro de acceso como que designa el espacio de instrucciones principal, y no examina los contenidos reales del registro de acceso. Por lo tanto, los 16 registros de acceso pueden designar, en cualquier momento, el espacio de instrucciones principal y un máximo de otros 15 espacios.

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

En, por ejemplo, la z/Arquitectura, cuando una dirección virtual es usada por una CPU para acceder al almacenamiento principal (también conocido como, la memoria principal), primero se convierte, por medio de la traducción de direcciones dinámicas (DAT), a una dirección real, y después, por medio del prefijo, a una dirección absoluta. El DAT puede usar desde uno o 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 mayor nivel para un espacio de direcciones específico es llamado un elemento de control del espacio, y es encontrado para su uso por el DAT en un registro de control o según sea especificado por un registro de acceso. De manera alternativa, el elemento de control del espacio de direcciones para un espacio de direcciones puede ser una designación de espacio real, que indique que el DAT a de traducir la dirección virtual simplemente tratándola como una dirección real y sin usar ninguna tabla.

El DAT usa, en diferentes momentos, los elementos de control del espacio de direcciones en diferentes registros de control o especificados por los registros de acceso. La elección es determinada por el modo de traducción especificado en la PSW. Hay disponibles cuatro modos de traducción: modo de espacio principal, modo de espacio secundario, modo de registro de acceso y modo de espacio local. Los diferentes espacios de direcciones son abordables dependiendo del modo de traducción.

En cualquier instante cuando la CPU está en el modo de espacio principal o el modo de espacio secundario, la CPU puede traducir las direcciones virtuales que pertenecen a dos espacios de direcciones – el espacio de direcciones principal y el segundo espacio de direcciones. En cualquier instante cuando la CPU está en el modo de registro de acceso, ésta puede traducir las direcciones virtuales de hasta 16 espacios de direcciones – el espacio principal y hasta 15 espacios de direcciones especificados por AR. En cualquier instante en que la CPU esté en el modo de espacio local, puede traducir las direcciones virtuales del espacio de direcciones local.

- El espacio de direcciones principal se identifica como tal ya que está compuesto de direcciones virtuales principales, que son traducidas por medio del elemento de control del espacio de direcciones principal (ASCE). De manera similar, el espacio de direcciones secundario está compuesto de direcciones virtuales secundarias traducidas por medio del ASCE secundario; los espacios de direcciones especificados por AR consisten de direcciones virtuales especificadas por AR traducidas por medio del ASCE local. Los ASCE principal y secundario están en los registros de control 1 y 7, respectivamente. Los ASCE especificados por AR están en entradas de segunda-tabla-ASN que se ubican a través de un proceso llamado traducción del registro de acceso (ART) que usa los registros 2, 5 y 8 de control. El ASCE local está en el registro 13 de control.
- Una realización de un entorno informático para incorporar y usar uno o más aspectos de la función transaccional descrita en la presente memoria se describe con referencia a la FIG. 1.
- Referente a la FIG. 1, en un ejemplo, el entorno 100 informático está basado en la z/Arquitectura, ofrecida por la Corporación de Máquinas de Negocios Internacionales (IBM®), Armonk, Nueva York. La z/Arquitectura se describe en una Publicación de IBM titulada "z/Arquitectura – Principios de Funcionamiento ", Publicación N°. SA22-7932-08, 9ª Edición, agosto de 2010.
- Z/ARQUITECTURA, IBM, y Z/OS y Z/VM (referenciados más adelante) son marcas registradas de la Corporación de Máquinas de Negocios Internacionales, Armonk, Nueva York. Otros nombres usados en la presente memoria pueden ser marcas registradas o nombres de productos de la Corporación de Máquinas de Negocios Internacionales u otras compañías.
- Como un ejemplo, el entorno 100 informático incluye un complejo 102 de procesador central (CPC) acoplado a uno o más dispositivos 106 de entrada/salida (I/O) a través de una o más unidades 108 de control. El complejo 102 de procesador central incluye, por ejemplo, uno o más procesadores 110 centrales, una o más particiones 112 (por ejemplo, particiones lógicas (LP)), un hipervisor 114 de partición lógica, y un subsistema 115 de entrada/salida, cada uno de los cuales se describe más adelante.
- Los procesadores 110 centrales son recursos de procesador físicos asignados a las particiones lógicas. En concreto, cada partición 112 lógica tiene uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte del procesador 110 físico asignado a la partición. Los procesadores lógicos de una partición 112 concreta pueden ser bien dedicados a la partición, para que el recurso 110 de procesador subyacente se reserve para esa partición; o compartidos con otra partición, para 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 de manera opcional, un sistema operativo residente en ésta, 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 la Corporación de Máquinas de Negocios Internacionales, Armonk, Nueva York. Las particiones 112 lógicas son gestionadas por un hipervisor 114 de partición lógica, que es implementado por un firmware que corre sobre los procesadores 110. Tal como se usa en la presente memoria, el firmware incluye, por ejemplo, el micro código y/o el mili código del procesador. Esto incluye, por ejemplo, las instrucciones de nivel de hardware y/o las estructuras de datos usadas en la implementación de código de máquina de nivel superior. En una realización, incluye, por ejemplo, el código propietario que se entrega normalmente como el micro código que incluye un software o micro código de confianza específico del hardware subyacente y controla el acceso del sistema operativo al hardware del sistema.
- Las particiones lógicas y el hipervisor de la partición lógica comprenden cada una uno o más programas que residen en las respectivas particiones del almacenamiento central asociado con los procesadores centrales. Un ejemplo de hipervisor 114 de partición lógica es el Gestor de Recursos/Sistema de Procesador (PR/SM), ofrecido por la Corporación de Máquinas de Negocios Internacionales, Armonk, Nueva York.
- El subsistema 115 de entrada/salida dirige el flujo de información entre los dispositivos 106 de entrada/salida y el almacenamiento principal (también conocido como, memoria principal). Se acopla al complejo de procesamiento central, pudiendo ser parte del complejo de procesamiento central o separado del mismo. El subsistema I/O descarga a los procesadores centrales de la tarea de comunicarse directamente con los dispositivos de entrada/salida y permite que el procesamiento de los datos se desarrolle de manera concurrente con el procesamiento de entrada/salida: Para proporcionar las comunicaciones, el subsistema de I/O emplea adaptadores de comunicaciones de I/O. Existen diversos tipos de adaptadores de comunicaciones que incluyen, por ejemplo, canales, adaptadores de I/O, tarjetas PCI, tarjetas Ethernet, tarjetas de Pequeña Interfaz de Almacenamiento Informático (SCSI), etc. En el ejemplo concreto descrito en la presente memoria, los adaptadores de comunicaciones de I/O son canales, y por lo tanto, el subsistema de I/O es referido en la presente memoria como un subsistema de canal. Sin embargo, esto es sólo un ejemplo. Se pueden usar otros tipos de subsistemas de I/O.
- Los subsistemas de I/O usan una o más rutas de entrada/salida como enlaces de comunicación en la gestión del flujo de información hacia o desde los dispositivos 106 de entrada/salida. En este ejemplo concreto, estas rutas son llamadas rutas de canal, ya que los adaptadores de comunicación son canales.

El entorno informático descrito anteriormente es sólo un ejemplo de un entorno informático que se puede usar. Se pueden usar otros entornos, que incluyen, pero no se limitan a, entornos no particionados, otros entornos particionados, y/o entornos emulados; las realizaciones no se limitan a ningún entorno.

5 De acuerdo con uno o más aspectos, la función de ejecución transaccional es una mejora de la CPU que proporciona los medios mediante los cuales la CPU puede ejecutar una secuencia de instrucciones – conocido como transacción – que pueden acceder a múltiples ubicaciones de almacenamiento, incluyendo la actualización de estas ubicaciones. Según es observado por otras CPU y el subsistema de I/O, la transacción es bien (a) completada en su totalidad como una única operación atómica, o (b) cancelada, potencialmente sin dejar evidencia alguna de que alguna vez se haya ejecutado (excepto para ciertas condiciones descritas en la presente memoria). Por tanto, una  
10 transacción completada de manera exitosa puede actualizar numerosas ubicaciones de almacenamiento sin ningún bloqueo especial lo que es necesario en el modelo clásico de multiprocesamiento.

La función de ejecución transaccional incluye, por ejemplo, uno o más controles; una o más instrucciones; procesamiento transaccional, incluir la ejecución restringida y no restringida; y cancelar el procesamiento, cada uno de los cuales se describe de manera adicional a continuación.

15 En una realización, tres controles de propósito especial, incluyendo una Palabra de Estado de Programa (PSW), una dirección de bloque de diagnóstico de transacciones (TDB), y una profundidad de anidación de transacciones; cinco bits de registro de control; y seis instrucciones generales, incluyendo el INICIO DE TRANSACCIÓN (restringido y no restringido), el FIN DE TRANSACCIÓN, la EXTRACCIÓN DE LA PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES, la CANCELACIÓN DE TRANSACCIÓN, y el ALMACENAMIENTO NO TRANSACCIONAL, se  
20 usan para controlar la función de ejecución transaccional. Cuando la función se instala, se instala, por ejemplo, en todas las CPU en la configuración. Una indicación de función, bit 73 en una implementación, cuando es un uno, indica que se instala la función de ejecución transaccional.

25 Cuando la función de ejecución transaccional se instala, la configuración proporciona una función de ejecución transaccional no restringida, y de manera opcional, una función de ejecución transaccional restringida, cada una de las cuales se describe más adelante. Cuando las indicaciones 50 y 73 de función, como ejemplo, son ambas un uno, se instala la función de ejecución transaccional restringida. Ambas indicaciones de función se almacenan en la memoria en las ubicaciones especificadas.

30 Tal como se usa en la presente memoria, la instrucción llamada INICIO DE TRANSACCIÓN se refiere a las instrucciones que tienen el mnemónico TINICIO (Inicio de Transacción para una transacción no restringida) y TINICIOC (Inicio de Transacción para una transacción restringida). Las discusiones pertenecientes a una instrucción específica son indicadas mediante el nombre de instrucción seguido por el mnemónico entre paréntesis o corchetes, o simplemente por el mnemónico.

35 Una realización de un formato de una instrucción de INICIO DE TRANSACCIÓN (TINICIO) se representa en las FIG. 2A-2B. Como un ejemplo, una instrucción 200 TINICIO incluye un campo 202 de código de operación que incluye un código de operación que especifica una operación de inicio de transacción no restringida; un campo 204 de base ( $B_1$ ); un campo 206 de desplazamiento ( $D_1$ ); y un campo 208 inmediato ( $I_2$ ). Cuando el campo  $B_1$  es distinto de cero, los contenidos del registro general especificados por el  $B_1$  204 se añaden al  $D_1$  206 para obtener la primera dirección de operando.

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

40 • Cuando la profundidad de anidación de transacción es inicialmente cero, la primera dirección de operando designa la ubicación del bloque de diagnóstico de transacciones de 256 byte, llamado el TDB de TINICIO especificado (descrito de manera adicional más adelante) dentro del cual se puede almacenar diversa información de diagnóstico si se cancela la transacción. Cuando la CPU está en modo espacio principal o modo registro de acceso, la primera dirección de operando designa una ubicación en el espacio de  
45 direcciones principal. Cuando la CPU está en el modo de espacio secundario o de espacio local, la primera dirección de operando designa una ubicación en el espacio de direcciones secundario o local, respectivamente. Cuando el DAT está apagado, la dirección (TDBA) del bloque de diagnóstico de transacciones (TDB) designa una ubicación en el almacenamiento real.

50 Se determina la accesibilidad de almacenamiento del primer operando. Si es accesible, la dirección lógica del operando se ubica dentro de la dirección del bloque de diagnóstico de transacciones (TDBA), y se valida la TDBA.

- Cuando la CPU está ya en el modo de ejecución transaccional no restringido, la TDBA no se modifica y no se puede predecir si se comprueba la accesibilidad del primer operando.

55 Cuando el campo  $B_1$  es cero, no se detectan excepciones para el primer operando y, para la instrucción TINICIO externa, la TDBA es inválida.

Los bits del campo  $I_2$  se definen como sigue, en un ejemplo:

- 5 Máscara 210 de Guardado de Registro General (GRSM) (FIG. 2B): Los bits 0-7 del campo  $I_2$  contienen la máscara de guardado de registro general (GRSM). Cada bit de la GRSM representa un pareja par-impar de registros generales, donde el bit 0 representa los registros 0 y 1, el bit 1 represente los registros 2 y 3, y así sucesivamente. Cuando un bit en el GRSM de la instrucción TINICIO más externa es cero, no se guarda la correspondiente pareja de registros. Cuando el bit en la GRSM de la instrucción TINICIO más externa es uno, se guarda la pareja de registros correspondientes en un modelo dependiente de la ubicación que no es directamente accesible por el programa.
- 10 Si la transacción se cancela, las parejas de registros guardados se restauran a sus contenidos de cuando se ejecutó la instrucción TINICIO más externa. Los contenidos de todos los demás registros (no guardados) generales no se restauran cuando se cancela la transacción.
- La máscara de guardado de registro general se ignora en todos los TINICIO excepto para los más externos.
- 15 Modificación 212 AR permitida (A): El control A, bit 12 del campo  $I_2$ , controla si la transacción está habilitada para modificar un registro de acceso. El control efectivo de la modificación AR permitida es la Y lógica del control A en la instrucción TINICIO para el nivel de anidación actual y para todos los niveles exteriores.
- Si el control A efectivo es cero, la transacción será cancelada con código de cancelación 11 (instrucción restringida) si se hace un intento de modificar cualquier registro de acceso. Si el control A efectivo es uno, la transacción no será cancelada si se modifica un registro de acceso (ausencia de cualquier otra condición de cancelación).
- 20 Operación 214 de Punto Flotante Permitida (F): El control F, bit 13 del campo  $I_2$ , controla si se permite la transacción para ejecutar instrucciones específicas de punto flotante. El control efectivo de la operación de punto flotante permitida es la Y lógica del control en la instrucción TINICIO para el nivel de anidación actual y para todos los niveles exteriores.
- Si el control F efectivo es cero, entonces (a) la transacción será cancelada con código de cancelación 11 (instrucción restringida) si se hace un intento de ejecutar una instrucción en punto flotante, y (b) el código de excepción de datos (DXC) en el byte 2 del registro de control de punto flotante (FPCR) no será fijado por ninguna condición de excepción del programa de excepción de datos. Si el control F efectivo es uno, entonces (a) la transacción no será cancelada si se hace un intento de ejecutar una instrucción en punto flotante (ausencia de cualquier otra condición de cancelación), y (b) el DXC en el FPCR se puede fijar mediante una condición de excepción del programa de excepción de datos.
- 25 Control 216 de Filtrado de Interrupción de Programa (PIFC): Los bits 14-15 del campo  $I_2$  son el control de filtrado de interrupción de programa (PIFC). El PIFC controla si ciertas clases de condiciones de excepción de programa (por ejemplo, la excepción de direccionamiento, la excepción de datos, la excepción de operación, la excepción de protección, etc.) que se producen mientras que la CPU está en el modo de ejecución transaccional resultan en una interrupción.
- 30 El PIFC efectivo es el mayor valor del PIFC en la instrucción TINICIO para el nivel de anidación actual y para todos los niveles exteriores. Cuando el PIFC es cero, todas las condiciones de excepción de programa resultan en una interrupción. Cuando el PIFC es uno, las condiciones de excepción de programa que tienen una clase de ejecución transaccional de 1 y 2 resultan en una interrupción. (A cada condición de excepción de programa se le asigna al menos 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 ejecución transaccional, y si el sistema operativo necesita ver la interrupción). Cuando el PIFC efectivo es dos, las condiciones de excepción de programa que tienen una clase de ejecución transaccional de 1 resultan en una interrupción. Se reserva un PIFC de 3.
- 35 Los bits 8-11 del campo  $I_2$  (bits 40-43) de la instrucción) se reservan y deberían contener ceros; en cualquier otro caso, el programa puede no funcionar de manera compatible en el futuro.
- Se describe una realización de un formato de una instrucción de Inicio de Transacción restringido (TINICIOOC) con referencia a las FIG. 3A-3B. En un ejemplo, TINICIO 300 incluye un campo 302 de código de operación que incluye un código de operación que especifica una operación de inicio de transacción restringida; un campo 304 de base ( $B_1$ ); un campo 306 de desplazamiento ( $D_1$ ); y un campo 308 inmediato ( $I_2$ ). Los contenidos del registro general especificado por el  $B_1$  304 se añaden al  $D_1$  306 para obtener la primera dirección de operando. Sin embargo, con la instrucción de inicio de transacción restringida, la primera dirección de operando no se usa para acceder al almacenamiento. En su lugar, el campo  $B_1$  de la instrucción incluye ceros; en otro caso, se reconoce una excepción de especificación.
- 40 En una realización, el campo  $I_2$  incluye diversos controles, un ejemplo de los cuales se representa en la FIG. 3B.
- 55 Los bits del campo  $I_2$  se definen como sigue, en un ejemplo:



5 Máscara 310 de Guardado de Registro General (GRSM): Los bits 0-7 del campo  $I_2$  contienen la máscara de guardado de 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, no se guarda la correspondiente pareja de registros. Cuando un bit en la GRSM es uno, se guarda la correspondiente pareja de registros en una ubicación dependiente del modelo que no es directamente accesible por el programa.

Si se cancela la transacción, las parejas de registros guardados se restauran a sus contenidos de cuando se ejecutó la instrucción de INICIO DE TRANSACCIÓN externa. Los contenidos de todos los otros registros generales (sin guardar) no se restauran cuando se cancela una transacción restringida.

10 Cuando se usa el TINICIO C para continuar la ejecución en el modo de ejecución de transacción no restringido, se ignora la máscara de guardado de registro general.

15 Modificación 312 AR Permitida (A): El control A, bit 12 del campo  $I_2$ , controla si se permite la transacción para modificar un registro de acceso. El control efectivo de la modificación AR permitida es la Y lógica del control A en la instrucción TINICIO para el nivel de anidación actual para cualquier instrucción TINICIO o TINICIO externa.

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

20 Los bits 8-11 y 13-15 del campo  $I_2$  (bits 40-43 y 45-47 de la instrucción) se reservan y deberían contener ceros.

25 El final de una instrucción de Inicio de Transacción es especificado por una instrucción de FIN DE TRANSACCIÓN (TFIN), un formato de la cual se representa en la FIG. 4. Como un ejemplo, una instrucción TFIN 400 incluye un campo 402 de código de operación que incluye un código de operación que especifica una operación de final de transacción.

Se usan un número de términos con respecto a la función de ejecución transaccional, y por lo tanto, únicamente por conveniencia, se proporciona una lista de términos más adelante en orden alfabético. En una realización, estos términos tienen la siguiente definición:

30 Cancelación: Una transición se cancela cuando se finaliza antes de la instrucción de FIN DE TRANSACCIÓN que resulta en una profundidad de anidación de transacción de cero. Cuando se cancela la transacción, ocurre lo siguiente, en una realización:

- Los accesos transaccionales al almacenamiento hechos por cualquiera y todos los niveles de la transacción se descartan (esto es, no se confirman).
- Los accesos no transaccionales al almacenamiento hechos por cualquiera y todos los niveles de la transacción se confirman.
- Los registros designados por la máscara de guardado de registro general (GRSM) de la instrucción de INICIO DE TRANSACCIÓN más externa se restauran a sus contenidos antes de la ejecución transaccional (esto es, a sus contenidos en la ejecución de la instrucción de INICIO DE TRANSACCIÓN más externa). Los registros generales no designados por la máscara de guardado de registro general de la instrucción de INICIO de TRANSACCIÓN más externa no se restauran.
- Los registros de acceso, los registros de punto flotante, y el registro de control de punto flotante no se restauran. Cualquier cambio hecho a estos registros durante la ejecución de la transacción se retienen cuando se cancela la transacción.

45 Una transacción puede ser cancelada debido a una variedad de razones, incluyendo el intento de ejecución de una instrucción restringida, el intento de modificación de un recurso restringido, el conflicto transaccional, el exceso de diversos recursos CPU, cualquier condición de intercepción de ejecución interpretativa, cualquier interrupción, una instrucción de CANCELACIÓN DE TRANSACCIÓN, y otras razones. Un código de cancelación de transacción proporciona razones específicas por las que se puede cancelar una transacción.

50 Un ejemplo de un formato de una instrucción de CANCELACIÓN DE TRANSACCIÓN (TCANCELACIÓN) se describe con referencia a la FIG. 5. Como un ejemplo, una instrucción 500 TCANCELACIÓN incluye un campo 502 de código de operación que incluye un código de operación que especifica una operación de cancelación de transacción; un campo 504 de base ( $B_2$ ); y un campo 506 de desplazamiento ( $D_2$ ). Cuando el campo  $B_2$  es distinto de cero, los contenidos del registro general especificados por  $B_2$  504 se añaden a  $D_2$  506 para obtener una segunda dirección de operando; en otro caso, la segunda dirección de operando se forma únicamente a partir del campo  $D_2$  y

5 se ignora el campo B<sub>2</sub>. La segunda dirección e operando no se usa para direccionar datos; en su lugar, la dirección forma el código de cancelación de transacción que se ubica en el bloque de diagnóstico de transacciones durante el procesamiento de la cancelación. El cálculo de la dirección para la segunda dirección de operando sigue las reglas de la aritmética de direcciones: en el modo de direccionamiento de 24 bits, los bits 0-29 se fijan a cero, en el modo de direccionamiento de 31 bits, los bits 0-32 se fijan a cero.

10 Confirmación: Al finalizar una instrucción de FIN DE TRANSACCIÓN la CPU confirma los accesos de almacenamiento hechos por la transacción (esto es, la transacción más externa y cualesquiera niveles anidados) de manera tal que son visibles a otras CPU y al subsistema de I/O. Según es observado por otras CPU y por el subsistema I/O, todos los accesos recuperados y almacenados hechos por todos los niveles anidados de la transacción parecen que ocurren como una única operación concurrente cuando se produce la confirmación.

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

15 Conflicto: Un acceso transaccional hecho por una CPU entra en conflicto con ya sea (a) un acceso transaccional o un acceso no transaccional hecho por otra CPU, o (b) el acceso no transaccional hecho por el subsistema de I/O, si ambos accesos son a alguna ubicación dentro de la misma línea de caché, y uno o más de los accesos es un almacenamiento.

20 Un conflicto puede ser detectado por una ejecución especulativa de la CPU de instrucciones, incluso aunque el conflicto pueda no 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 sometido a las siguientes limitaciones:

- Está disponible un subconjunto de instrucciones generales.
- Se pueden ejecutar un número limitado de instrucciones.
- 25 • Se pueden acceder a un número limitado de ubicaciones de almacenamiento del operando.
- Se limita la transacción a un nivel de anidación único.

En ausencia de interrupciones o conflictos repetidos con otras CPU o el subsistema de I/O, finalmente se completa una transacción restringida, por tanto no se requiere una rutina controladora de cancelaciones. Las transacciones restringidas se describen en detalle más adelante.

30 Cuando una instrucción restringida de INICIO DE TRANSACCIÓN (TINICIO) se ejecuta mientras la CPU está ya en el modo de ejecución no restringido, la ejecución continua como una transacción no restringida anidada.

35 Modo de Ejecución Transaccional Restringido: Cuando la profundidad de anidación de transacciones es cero, y una transacción es iniciada por una instrucción TINICIO, 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 transacciones es uno.

Transacción Anidada: Cuando se emite la instrucción de INICIO DE TRANSACCIÓN mientras la CPU está en el modo de ejecución transaccional no restringido, se anida la transacción.

40 La función de ejecución transaccional usa un modelo llamado anidación plana. En el modo de anidación plana, los almacenamientos hechos por una transacción interna no se pueden observar por otras CPU y por el subsistema de I/O hasta que la transacción más externa confirme sus almacenamientos. De manera similar, si se cancela una transacción, se cancelan todas las transacciones anidadas, y se cancelan todos los almacenamientos transaccionales de todas las transacciones anidadas.

45 Un ejemplo de transacciones anidadas se representa en la FIG. 6. Como se muestra, una primera TINICIO 600 inicia la transacción 601 más externa, TINICIO 602 inicia una primera transacción anidada, y TINICIO 604 inicia una segunda transacción anidada. En este ejemplo, TINICIO 604 y TFIN 606 definen la transacción 608 más interna. Cuando TFIN 610 se ejecuta, se confirman los almacenamientos transacciones 612 para la transacción más externa y todas las transacciones internas.

50 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 se limita en la manera de una transacción restringida, aún se puede cancelar debido a una variedad de causas.

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

5 Acceso No Transaccional: Los accesos no transaccionales son accesos de operandos de almacenamiento hechos por la CPU cuando no está en el modo de ejecución transaccional (esto es, los accesos de almacenamiento clásicos fuera de una transacción). Además, los accesos hechos por el subsistema de I/O son accesos no transaccionales. De manera adicional, la instrucción de ALMACENAMIENTO NO TRANSACCIONAL se puede usar para provocar un  
10 acceso al almacenamiento no transaccional mientras que la CPU está en el modo de ejecución transaccional no restringido.

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 códigos de operación que especifican un código de operación que designa una operación de almacenamiento no transaccional; un campo 704 de registro ( $R_1$ ) que especifica un  
15 registro, los contenidos del cual son llamados el primer operando; un campo 706 de índice ( $X_2$ ), un campo 708 de base ( $B_2$ ); un primer campo 710 de desplazamiento ( $DL_2$ ); y un segundo campo 712 de desplazamiento ( $DH_2$ ). Los contenidos de los registros generales designados por los campos  $X_2$  y  $B_2$  se añaden a los contenidos de la concatenación de los contenidos de los campos  $DH_2$  y  $DL_2$  para formar la segunda dirección de operando. Cuando ya sea uno o ambos de los campos  $X_2$  o  $B_2$  son cero, los registros correspondientes no toman parte en la adición.

20 El primer operando de 64 bits no se ubica no transaccionalmente sin cambios en la ubicación del segundo operando. El desplazamiento, formado por la concatenación de los campos  $DH_2$  y el  $DL_2$ , se trata como un número entero binario con signo de 20 bits.

El segundo operando se ha de alinear en un límite de doble palabra; en otro caso, se reconoce la excepción de especificación y se suprime la operación.

25 Transacción Externa/Más Externa: Una transacción con una profundidad de anidación de transacciones de baja numeración es una transacción externa. Una transacción con un valor de profundidad de anidación de transacciones de uno es la transacción más externa.

Una instrucción de INICIO DE TRANSACCIÓN más externa es una que se ejecuta cuando la profundidad de anidación de transacciones es inicialmente cero. Una instrucción de FIN DE TRANSACCIÓN más externa es una  
30 que provoca que la profundidad de anidación de transacciones cambie de uno a cero. Una transacción restringida es la transacción más externa, en esta realización.

Filtrado de Interrupción de Programa: Cuando se cancela una transacción debido a ciertas condiciones de excepción de programa, el programa puede de manera opcional evitar que ocurra la interrupción. Esta técnica es llamada filtrado de interrupción de programa. El filtrado de interrupción de programa se somete a la clase transaccional de la  
35 interrupción, el control efectivo del filtrado de interrupción de programa de la instrucción de INICIO DE TRANSACCIÓN, y el filtrado de interrupción de programa de ejecución transaccional se anulan en el registro 0 de control.

Transacción: Una transacción incluye los accesos de operando de almacenamiento hechos, y los registros generales seleccionados alterados, mientras la CPU está en el modo de ejecución de transacción. Para una transacción no  
40 restringida, los accesos de operando de almacenamiento pueden incluir tanto los accesos transaccionales y los accesos no transaccionales. Para una transacción restringida, los accesos de operando de almacenamiento se limitan a los accesos transaccionales. Como se observa por otras CPU y por el subsistema de I/O, todos los accesos de operando de almacenamiento hechos por la CPU mientras está en el modo de ejecución de transacción parece que se producen como una operación concurrente única. Si se cancela una transacción, los accesos de  
45 almacenamiento transaccional se descartan, y cualesquiera registros designados por la máscara de guardado registro general de la instrucción de INICIO DE TRANSACCIÓN más externa se restauran a sus contenidos anteriores a la ejecución transaccional.

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  
50 instrucción de ALMACENAMIENTO NO TRANSACCIONAL.

Modo de Ejecución Transaccional: El término modo de ejecución transaccional (también conocido como modo de ejecución de transacción) describe el funcionamiento común de tanto los modos de ejecución transaccional restringido y no restringido. Por tanto, cuando se describe el funcionamiento, se usan los términos restringido y no restringido para calificar el modo de ejecución transaccional.

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

Como se observa por la CPU, las extracciones y almacenamiento hechos en el modo de ejecución transaccional no son diferentes de aquellos hechos cuando no se está en el modo de ejecución transaccional.

5 En una realización de la z/Arquitectura, la función de ejecución transaccional está bajo el control de los bits 8-9 del registro 0 de control, los bits 61-63 del registro 2 de control, la profundidad de anidación de transacciones, la dirección del bloque de diagnóstico de transacciones, y la palabra de estado de programa de cancelación de transacción (PSW).

10 Siguiendo a un reinicio inicial de la CPU, los contenidos de las posiciones 8-9 de bit del registro 0 de control, las posiciones 62-63 del registro 2 de control, y la profundidad de anidación de transacción se fijan a cero. Cuando el control de ejecución transaccional, el bit 8 del registro 0 de control, es cero, la CPU no se puede ubicar dentro del modo de ejecución transaccional.

Detalles adicionales respecto a los diversos controles se describen a continuación.

15 Como se indica, la función de ejecución transaccional está controlada por dos bits en el registro cero de control y tres bits en el registro dos de control. Por ejemplo:

Bits del Registro 0 de Control: La asignación de bit es como sigue, en una realización:

20 Control de Ejecución Transaccional (TXC): El bit 8 del registro cero de control es el control de ejecución transaccional. Este bit proporciona un mecanismo a través del cual el programa de control (por ejemplo, el sistema operativo) puede indicar si la función de ejecución de transacciones es usable o no por el programa. El bit 8 ha de ser uno para entrar de manera exitosa al modo de ejecución transaccional.

Cuando el bit 8 del registro 0 de control es cero, los intentos de ejecución de las instrucciones EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES, INICIO DE TRANSACCIÓN y FIN DE TRANSACCIÓN resulta en una ejecución de operación especial.

25 Una realización de un formato de una instrucción de EXTRACCIÓN DE LA PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES (ETND) se describe con referencia a la FIG. 8. Como ejemplo, una instrucción 800 de EXTRACCIÓN DE LA PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES incluye un campo 802 de código de operación que especifica un código de operación que indica la operación de extracción de la profundidad de anidación de transacciones; y un campo 804 del registro R<sub>1</sub> que designa un registro general.

30 La profundidad de anidación de transacciones actual se coloca en los bits 48-63 del registro general R<sub>1</sub>. Los bits 0-31 del registro permanecen sin cambio, y los bits 32-47 del registro se fijan a cero.

En una realización adicional, la máxima profundidad de anidación de transacción se coloca también en el registro R<sub>1</sub> general, tal como en los bits 16-31.

35 En una realización, la instrucción de EXTRACCIÓN DE LA PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES permite una transacción no restringida para inspeccionar el valor actual de la profundidad de anidación de transacciones y almacenar este valor en un registro. El valor actual de la profundidad de anidación actual se mantiene en un contador interno de la CPU. Durante el modo de transacción no restringida, cuando se ejecuta un TINICIO sin que intervenga un TFIN, el contador se incrementa. Este contador puede ser de interés para el programa, tanto al determinar la secuenciación de código apropiada, como para propósitos de diagnóstico.

40 En una realización, se reconoce una excepción especial de operación y se suprime la operación si el control de ejecución transaccional, bit 8 del registro 0 de control, es cero.

Código de Condición: El código permanece sin cambio.

Excepciones de Programa:

- 45
- Operación (función de ejecución transaccional no instalada)
  - Operación especial
  - Restricción de transacción (debido a una instrucción restringida)

Nota: una profundidad de anidación de transacciones de cero indica que la CPU no está en el modo de ejecución transaccional.

Anulación de Filtrado de Interrupción de Programa de Ejecución de Transacciones (PIFO): El bit 9 del registro cero de control es la anulación de filtrado de interrupción de programa de ejecución de transacciones. Este bit proporciona un mecanismo mediante el cual el programa de control puede asegurar que cualquier condición de excepción de programa que ocurra mientras la CPU está en el modo de ejecución transaccional resulta en una interrupción, independientemente del control efectivo de filtrado de interrupción de programa o implicado por la instrucción o instrucciones de INICIO DE TRANSACCIÓN.

Bits del Registro 2 de Control: Las asignaciones son como sigue, en una realización:

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

TDS

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

0	El TDC se aplica independientemente de si la CPU está en el estado de problema o supervisor.
---	--

1	El TDC aplica sólo cuando la CPU está en el estado de problema. Cuando la CPU está en el estado supervisor, el procesamiento es como si el TDC contuviera un cero.
---	--

Control de Diagnóstico de Transacción (TDC): Los bits 62-63 del registro 2 de control son un número entero sin signo de 2 bits que se puede usar para provocar que las transacciones sean canceladas de manera aleatoria con propósitos de diagnóstico. La codificación del TDC es como sigue, en un ejemplo:

TDC

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

0	Operación normal; las transacciones no se cancelan como resultado del TDC.
---	--

1	Se cancela cada transacción en una instrucción aleatoria, pero antes de la ejecución de la instrucción de FIN DE TRANSACCIÓN más externa.
---	---

2	Se cancelan transacciones aleatorias en una instrucción aleatoria.
---	--

3	Reservado
---	-----------

Cuando se cancela una transacción debido a un TDC distinto de cero, entonces puede ocurrir cualquiera de lo siguiente:

- El código de cancelación se fija a cualquiera de los códigos 7-11, 13-16, o 255, con el valor del código elegido de manera aleatoria por la CPU; el código de condición se fija correspondiente al código de cancelación. Los códigos de cancelación se describen de manera adicional más adelante.

- Para una transacción no restringida, el código de condición se fija a uno. En este caso, el código de cancelación no es aplicable.

Es dependiente del modelo si se implementa el valor 1 del TDC. Si no ese implementa un valor de 1 actúa como si se hubiera especificado un 2.

Para una transacción restringida, un valor del TDC de 1 es tratado como si se hubiera especificado un valor del TDC de 2.

Si se especifica un valor del TDC de 3, los resultados son impredecibles.

Dirección del Bloque de Diagnóstico de Transacciones (TDBA)

Una dirección válida de la dirección del bloque de diagnóstico de transacciones (TDBA) se fija desde la primera dirección de operando de la instrucción de INICIO DE TRANSACCIÓN (TINICIO) más externa cuando el campo B<sub>1</sub> de la instrucción no es cero. Cuando la CPU está en el espacio principal o en el modo de registro de acceso, la TDBA designa una ubicación en el espacio de direcciones principal. Cuando la CPU está en el espacio secundario, o en el modo de espacio local, la TDBA designa una ubicación en el espacio secundario o de direcciones locales, respectivamente. Cuando la DAT (Traducción de Direcciones Dinámica) está apagada, la TDBA designa una ubicación en el almacenamiento real.

La TDBA es usada por la CPU para ubicar el bloque de diagnóstico de transacciones – llamado el TDB de TINICIO especificado – si la transacción se cancela posteriormente. Los tres bits más a la derecha de la TDBA son cero, lo que significa que el TDB de TINICIO especificado está en un límite de doble palabra.

Cuando el campo B<sub>1</sub> de una instrucción INICIO DE TRANSACCIÓN (TINICIO) más externa es cero, la dirección del bloque de diagnóstico transaccional es inválida, y no se almacena un TDB de TINICIO especificado si la transacción se cancela posteriormente.

PSW de Cancelación de Transacciones (TAPSW)

- 5 Durante la ejecución de la instrucción de INICIO DE TRANSACCIÓN (TINICIO) cuando la profundidad de anidación es inicialmente cero, la PSW de cancelación de transacciones designa la siguiente instrucción secuencial (esto es, la instrucción que sigue el TINICIO más externo). Durante la ejecución de la instrucción de INICIO DE TRANSACCIÓN restringido (TINICIOC) cuando la profundidad de anidación es inicialmente cero, la PSW de cancelación de transacciones se fija a los contenidos de la PSW actual, excepto en que la dirección de instrucción de la PSW de cancelación de transacciones designa la instrucción TINICIOC (en lugar de la siguiente instrucción secuencial que sigue al TINICIOC).
- 10

- 15 Cuando se cancela la transacción, el código de condición en la PSW de cancelación de transacciones se reemplaza con un código que indica la gravedad de la condición de cancelación. Posteriormente, si se canceló la transacción debido a causas que no resultan en una interrupción, la PSW se carga desde la PSW de cancelación de transacciones; si la transacción fue cancelada debido a causas que resultan en una interrupción, la PSW de cancelación de transacciones se almacena como la antigua PSW de interrupción.

La PSW de cancelación de transacciones no se altera durante la ejecución de cualquier instrucción INICIO DE TRANSACCIÓN interna.

Profundidad de Anidación de Transacciones (TND)

- 20 La profundidad de anidación de transacciones es, por ejemplo, un valor sin signo de 16 bits que es aumentado cada vez que se completa una instrucción de INICIO DE TRANSACCIÓN con el código de condición 0 y disminuye cada vez que se completa la instrucción FIN DE TRANSACCIÓN. La profundidad de anidación de transacciones se reinicia a cero cuando una transacción es cancelada o por un reinicio de la CPU.

En una realización, se implementa una TND máxima de 15.

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

Bloque de Diagnóstico de Transacciones (TDB)

- 30 Cuando se cancela una transacción, se puede guardar diversa información de estado en un bloque de diagnóstico de transacciones (TDB), como sigue:

1. TDB de TINICIO especificado: Para una transacción no restringida, cuando el campo B<sub>1</sub> de la instrucción TINICIO más externa no es cero, la primera dirección de operando de la instrucción designa el TDB de TINICIO especificado. Esto es una ubicación especificada del programa de aplicación que puede ser examinada por el controlador de cancelaciones de la aplicación.
  2. TDB de Interrupción de Programa (PI): Si se cancela una transacción no restringida debido a una condición de excepción de programa no filtrada, o si se cancela una transacción no restringida debido a alguna condición de excepción de programa (esto es, cualquier condición que resulte en una interrupción de programa que es reconocida), el PI-TDB se almacena en ubicaciones en el área prefijada. Esto está disponible para que el sistema operativo inspeccione y finalice sesión en cualquier reporte de diagnóstico que éste pueda proporcionar.
  3. TDB de Intercepción: Si se cancela la transacción debido a cualquier condición de excepción de programa que resulta en una intercepción (esto es, la condición provoca la ejecución interpretativa para finalizar y controlar para volver al programa servidor), se almacena un TDB dentro de la ubicación especificada en el bloque de descripción de estado para el sistema operativo invitado.
- 35
- 40
- 45

El TDB de TINICIO especificado sólo se almacena, en una realización, cuando la dirección de TDB es válida (esto es, cuando el campo B<sub>1</sub> de la instrucción TINICIO más externa no es cero).

- 50 Para las cancelaciones debidas a condiciones de excepción de programa no filtradas, sólo uno se almacenará de bien el TDB de PI o el TDB de Intercepción. Por tanto, pueden existir cero, uno, o dos TDB almacenados para una cancelación.

A continuación, se describen detalles adicionales respecto un ejemplo de cada uno de los TDB:

5 TDB de TINICIO especificado: La ubicación de 256 byte especificada por una dirección del bloque de diagnóstico de transacciones válida. Cuando la dirección del bloque de diagnóstico de transacciones es válida, se almacena el TDB de TINICIO especificado en una cancelación de transacción. El TDB de TINICIO especificado se somete a todos los mecanismos de protección de almacenamiento que están en efecto en la ejecución de la instrucción de INICIO DE TRANSACCIÓN más externa. Se detecta una alteración de almacenamiento PER (Registro de Evento de Programa) para cada parte del TDB de TINICIO especificado durante la ejecución del TINICIO más externo, no durante el procesamiento de cancelación de transacción.

Un propósito del PER es ayudar en la depuración de los programas. Permite al programa estar alerta a los siguientes tipos de eventos, como ejemplo:

- 10 • La ejecución de una instrucción de ramificación exitosa. Se proporciona la opción de tener un evento que ocurre sólo cuando la ubicación objetivo de ramificación está dentro del área de almacenamiento designado.
- La extracción de una instrucción desde el área de almacenamiento designado.
- 15 • La alteración de los contenidos del área de almacenamiento designada. Se proporciona la opción de tener un evento que ocurre sólo cuando el área de almacenamiento está dentro de los espacios de direcciones designados.
- La ejecución de una instrucción de ALMACENAMIENTO USANDO UNA DIRECCIÓN REAL.
- Ejecución de la instrucción de FIN DE TRANSACCIÓN.

20 El programa puede especificar de manera selectiva que uno o más de los anteriores tipos de eventos sea reconocido, excepto que el evento para ALMACENAR USANDO LA DIRECCIÓN REAL sólo se puede especificar junto con el evento de alteración de almacenamiento. La información respecto al evento PER es proporcionada al programa por medio de una interrupción de programa, con la causa de la interrupción siendo identificada en el código de interrupción.

25 Cuando la dirección de bloque de diagnóstico de transacciones no es válida, no se almacena el TDB de TINICIO especificado.

TDB de Interrupción de Programa: Ubicaciones reales 6, 144-6, 399 (1800-18FF hex).

El TDB de interrupción de programa se almacena cuando se cancela una transacción debido a una interrupción de programa. Cuando se cancela una transacción debido a otras causas, los contenidos del TDB de interrupción de programa son impredecibles.

30 El TDB de interrupción de programa no está sujeto a ningún mecanismo de protección. Los eventos de alteración de almacenamiento PER no se detectan para el TDB de interrupción de programa cuando se almacenan durante una interrupción de programa.

35 TDB de Intercepción: La ubicación real de servidor de 256 bytes especificada por las ubicaciones 488-495 de la descripción de estado. El TDB de intercepción se almacena cuando una transacción cancelada resulta en una intercepción de interrupción de programa invitado (esto es, código 8 de intercepción). Cuando se cancela una transacción debido a otras causas, los contenidos del TDB de intercepción son impredecibles. El TDB de intercepción no está sujeto a ningún mecanismo de protección.

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

40 Formato 902: El byte 0 contiene una indicación de validez y de formato, como sigue:

Valor	Significado
0	Los campos restantes del TDB son impredecibles.
1	Un TDB de formato-1, los campos restantes del cual se describen más adelante.
2-255	Reservado

45 Un TDB cuyo campo de formato es cero es referido como un TDN nulo.

Banderas 904: El byte 1 contiene diversas indicaciones, como sigue:

Validez del Identificador de Conflicto (CTV): Cuando se cancela una transacción debido a un conflicto de extracción o almacenamiento (esto es, códigos 9 o 10 de cancelación,

respectivamente), el bit 0 del byte 1 es la indicación de validez del identificador de conflicto. Cuando la indicación CTV es uno, el identificador 910 de conflicto en los bytes 16-23 del TDB contiene la dirección lógica en la que se detectó el conflicto. Cuando la indicación CTV es cero, los bytes 16-23 del TDB son impredecibles.

5 Cuando se cancela una transacción debido a cualquier razón distinta del conflicto de extracción o almacenamiento, el bit 0 del byte 1 se almacena como cero.

Indicación de Transacción Restringida (CTI): Cuando la CPU está en el modo de ejecución transaccional restringida, el bit 1 del byte 1 se fija a uno. Cuando la CPU está en el modo de ejecución transaccional no restringida, el bit 1 del byte 1 se fija a cero.

10 Reservados: Los bits 2-7 del byte 1 se reservan, y se almacenan como ceros.

Profundidad 906 de Anidación de Transacciones (TND): Los bytes 6-7 contienen la profundidad de anidación de transacciones cuando se canceló la transacción.

15 Código 908 de Cancelación de Transacción (TAC): Los bytes 8-15 contienen un código de cancelación de transacción sin signo de 64 bits. Cada punto de código indica una razón para que la transacción sea cancelada.

Este modelo depende de si el código de cancelación de transacción se almacena en el TDB de interrupción de programa cuando se cancela la transacción debido a condiciones distintas de la interrupción de programa.

20 Identificador 910 de Conflicto: Para las transacciones que se cancelan debido a un conflicto de extracción o almacenamiento (esto es, códigos 9 y 10 de cancelación, respectivamente), los bytes 16-23 contienen la dirección lógica de la ubicación de almacenamiento en la que fue detectado el conflicto. El identificador de conflicto es significativo cuando el bit CTV, bit 0 del byte 1, es uno.

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

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

30 Dirección 912 de Instrucción de Transacción Cancelada (ATIA): Los bytes 24-31 contienen una dirección de instrucción que identifica la instrucción que se estaba ejecutando cuando se detectó la cancelación. Cuando se canceló la transacción debido a los códigos de cancelación 2, 5, 6, 11, 13, o 256 o mayores, o cuando se canceló la transacción debido a los códigos de cancelación 4 o 13 y la condición de excepción de programa se anula, la ATIA apunta directamente a la instrucción que estaba siendo ejecutada. Cuando se cancela la transacción debido a los códigos de cancelación 4 o 12, y la condición de excepción de programa no se anula, La ATIA apunta pasada la instrucción que estaba siendo ejecutada.

35 Cuando una transacción se cancela debido a los códigos de cancelación 7-10, 14-16, o 255, la ATIA no indica necesariamente la instrucción exacta que provoca la cancelación, pero puede apuntar a una instrucción anterior o posterior dentro de la transacción.

40 Si se cancela la transacción debido a una instrucción que es el objetivo de una instrucción de tipo ejecutar la ATIA identifica la instrucción de tipo ejecución, ya sea apuntando a la instrucción o pasándola, dependiendo del código de cancelación como se describió anteriormente. La ATIA no indica el objetivo de la instrucción de tipo ejecutar.

La ATIA se somete al modo de direccionamiento cuando se cancela la transacción. En el modo de direccionamiento de 24 bit, los bits 0-40 del campo contienen ceros. En el modo de direccionamiento de 31 bits, los bits 0-32 del campo contienen ceros.

45 Es dependiente de modelo si la dirección de instrucción de transacción cancelada se almacena en el TDB de interrupción de programa cuando se cancela una transacción debido a condiciones distintas a una interrupción del programa.

50 Cuando se cancela una transacción debido al código de cancelación 4 o 12, y la condición de excepción de programa no se anula, la ATIA no apunta a la instrucción que provoca la cancelación. Restando el número de medias palabras indicado por el código de longitud de interrupción (ILC) de la ATIA, la instrucción que provoca la cancelación se puede identificar en condiciones que se están eliminando o terminando, o para eventos no PER que se están completando. Cuando se cancela una transacción debido a un evento PER, y no hay presente otra condición de excepción de programa, la ATIA es impredecible.



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

5 Identificación 914 de Acceso de Excepción (EAID): Para las transacciones que se cancelan debido a ciertas condiciones de excepción de programa filtradas, el byte 32 del TDB de TINICIO especificado contiene la identificación de acceso de excepción. En un ejemplo de z/Arquitectura, el formato de la EAID, y los casos para los que se almacena, son los mismos que aquellos descritos en la ubicación 160 real cuando los resultados de la condición de excepción en una interrupción, como se describe en la referencia anterior de Principios de Operación.

10 Para las transacciones que se cancelan por otras razones, incluyendo cualesquiera condiciones de excepción que resultan en una interrupción de programa, el byte 32 es impredecible. El byte 32 es impredecible en el TDB de interrupción de programa.

15 Este campo se almacena sólo en el TDB designado por la dirección de bloque de diagnóstico de transacciones; en otro caso, se reserva el campo. La EAID se almacena sólo para condiciones de excepción de programa de listas de acceso controladas o protección DAT, tipo ASCE, traducción de páginas, traducción de primera región, traducción de segunda región, traducción de tercera región, y traducción de segmentos.

20 Código 916 de Excepción de Datos (DXC): Para traducciones que se cancelan debido a condiciones de excepción de programa de excepción de datos filtrados, el byte 33 del TDB de TINICIO especificado contiene el código de excepción de datos. En un ejemplo de la z/Arquitectura, el formato del DXC, y los casos para lo que se almacena, son los mismos que aquellos descritos en la ubicación 147 real cuando los resultados de condición de excepción en una interrupción, como se describen en los Principios de Operación de referencia. En un ejemplo, la ubicación 147 incluye el DXC.

25 Para transacciones que se cancelan por otras razones, incluyendo cualesquiera condiciones de excepción que resulten en una interrupción de programa, el byte 33 es impredecible. El byte 33 es impredecible en el TDB de interrupción de programa.

30 Este campo se almacena sólo en el TDB designado por la dirección de bloque de diagnóstico de transacciones; en otro caso, el campo se reserva. El DXC se almacena sólo para condiciones de excepción de programa de datos.

35 Identificación 918 de Interrupción de Programa (PIID): Para transacciones que se cancelan debido a condiciones de excepción de programa filtradas, los bytes 36-39 del TDB de TINICIO especificado contiene la identificación de interrupción de programa. En un ejemplo de la z/Arquitectura, el formato de la PIID es el mismo que el descrito en las ubicaciones 140-143 reales cuando la condición resulta en una interrupción (como se describe en los anteriores Principios de Operación de referencia), excepto que el código de longitud de instrucción en los bits 13-14 de la PIID es respectivo a la instrucción en la que se detectó la condición de excepción.

40 Para transacciones que se cancelan por otras razones, incluyendo las condiciones de excepción que resultan en una interrupción de programa, los bytes 36-39 son impredecibles. Los bytes 36-39 son impredecibles en el TDB de interrupción de programa.

Este campo se almacena sólo en el TDB designado por la dirección de bloque de diagnóstico de transacciones; en otro caso, el campo se reserva. La identificación de interrupción de programa sólo se almacena para condiciones de excepción de programa.

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

- Lista de acceso controlada o protección DAT
- Tipo-ASCE
- Traducción de páginas
- Traducción de primera región
- Traducción de segunda región
- Traducción de tercera región

50

- Excepción de traducción de segmentos

En un ejemplo de la z/Arquitectura, el formato de la TEID es el mismo que el descrito en las ubicaciones 168-175 reales cuando la condición resulta en una interrupción, como se describe en los Principios de Operación anteriormente incorporados como referencia.

- 5 Para transacciones que se cancelan por otras razones, incluyendo las condiciones de excepción que resultan en una interrupción de programa, los bytes 40-47 son impredecibles. Los bytes 40-47 son impredecibles en el TDB de interrupción de programa.

Este campo se almacena sólo en el TDB designado por la dirección de bloque de diagnóstico de transacciones; en otro caso, el campo se reserva.

- 10 Dirección 922 de Evento de Rotura: Para transacciones que se cancelan debido a las condiciones de excepción de programa filtradas, los bytes 48-55 del TDB de TINICIO especificado contienen la dirección de evento de rotura. En un ejemplo de la z/Arquitectura, el formato de la dirección de evento de rotura es el mismo que el descrito en las ubicaciones 272-279 reales cuando la condición resulta en una interrupción, como se describe en los Principios de Operación de referencia anteriores.

- 15 Para transacciones que se cancelan por otras razones, incluyendo las condiciones de excepción que resultan en una interrupción de programa, los bytes 48-55 son impredecibles. Los bytes 48-55 son impredecibles en el TDB de interrupción de programa.

Este campo se almacena sólo en el TDB designado por la dirección de bloque de diagnóstico de transacciones; en otro caso, el campo se reserva.

- 20 Los detalles adicionales que se relacionan con los eventos de rotura se describen más adelante.

En una realización de la z/Arquitectura, cuando la función PER-3 se instala, proporciona al programa la dirección de la última instrucción para provocar la rotura en la ejecución secuencial de la CPU. El registro de la dirección de evento de rotura se puede usar como una ayuda de depuración para la detección de una ramificación descabellada. Esta función proporciona, por ejemplo, un registro de 64 bits en la CPU, llamado registro de direcciones de eventos de rotura. Cada vez que una instrucción distinta de la CANCELACIÓN DE TRANSACCIÓN provoca una rotura en la ejecución secuencial de la instrucción (esto es, la dirección de instrucción en la PSW se reemplaza en lugar de ser aumentada por la longitud de la instrucción), la dirección de esa instrucción se coloca en el registro de direcciones de eventos de rotura. Cuando se produce una interrupción de programa, si se indica o no el PER, los contenidos actuales del registro de direcciones de eventos de rotura se colocan en las ubicaciones 272-279 de almacenamiento reales.

- 25 Si la instrucción que provoca el evento de rotura es el objetivo de una instrucción de tipo ejecución (EJECUCIÓN O EJECUCIÓN RELATIVA LARGA), entonces la dirección de la instrucción usada para extraer la instrucción de tipo ejecución se coloca en el registro de direcciones de eventos de rotura.

- 35 En una realización de la z/Arquitectura, un evento de rotura se considera que ocurre cuando una de las siguientes instrucciones provoca una ramificación: RAMIFICACIÓN Y ENLAZADO (BAL, BALR), RAMIFICACIÓN Y GUARDADO (BAS, BASR); RAMIFICACIÓN Y GUARDADO Y FIJACIÓN DE MODO (BASSM); RAMIFICACIÓN Y FIJACIÓN DE MODO (BSM); RAMIFICACIÓN Y APILADO (BAKR); RAMIFICACIÓN SOBRE CONDICIÓN (BC, BCR); RAMIFICACIÓN SOBRE CONTADOR (BCT, BCTR, BCTG, BCTGR); RAMIFICACIÓN SOBRE ÍNDICE SUPERIOR (BXH, BXHG); RAMIFICACIÓN SOBRE ÍNDICE INFERIOR O IGUAL (BXLE, BXLEG); RAMIFICACIÓN EN RELACIÓN A CONDICIÓN (BRC); RAMIFICACIÓN EN RELACIÓN A CONDICIÓN LARGA (BRCL); RAMIFICACIÓN EN RELACIÓN A CONTADOR (BRCT, BRCTG); RAMIFICACIÓN EN RELACIÓN A ÍNDICE SUPERIOR (BRXH, BRXHG); RAMIFICACIÓN EN RELACIÓN A ÍNDICE INFERIOR O IGUAL (BRXLE, BRXLG); COMPARACIÓN Y RAMIFICACIÓN (CRB, CGRB); COMPARACIÓN Y RAMIFICACIÓN RELATIVA (CRJ, CGRJ); RAMIFICACIÓN INMEDIATA Y RAMIFICACIÓN (CIB, CGIB); COMPARACIÓN INMEDIATA Y RAMIFICACIÓN RELATIVA (CIJ, CGIJ); COMPARACIÓN LÓGICA Y RAMIFICACIÓN (CLRB, CLGRB); COMPARACIÓN LÓGICA Y RAMIFICACIÓN RELATIVA (CLRJ, CLGRJ); COMPARACIÓN LÓGICA INMEDIATA Y RAMIFICACIÓN (CLIB, CLGIB); y COMPARACIÓN LÓGICA INMEDIATA Y RAMIFICACIÓN RELATIVA (CLIJ, CLGIJ)

- 40 Un evento de rotura se considera también que ocurra cuando se completa una de las siguientes instrucciones: RAMIFICACIÓN Y FIJACIÓN DE AUTORIDAD (BSA); RAMIFICACIÓN EN GRUPO DE SUBESPACIO (BSG); RAMIFICACIÓN RELATIVA Y GUARDADO (BRAS); RAMIFICACIÓN RELATIVA Y GUARDADO LARGO (BRASL); CARGA DE PSW (LPSW); CARGA DE PSW EXTENDIDA (LPSWE); LLAMADA A PROGRAMA (PC); VUELTA A PROGRAMA (PR); TRANSFERENCIA DE PROGRAMA (PT); TRANSFERENCIA DE PROGRAMA CON CONSULTA (PTI); REANUDACIÓN DE PROGRAMA (RP); y TRAP (TRAP2, TRAP4).

Un evento de rotura no se considera que ocurra como resultado de una transacción que se cancela (ya sea de manera implícita o como resultado de la instrucción de CANCELACIÓN DE TRANSACCIÓN).

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

5 Para todos los códigos de cancelación excepto el 12 (interrupción de programa filtrada), la información de diagnóstico dependiente de modelo se guarda en cada TDB que se almacena.

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

- Los bytes 112-119 contienen un vector de 64 bits llamado de indicaciones de ramificación de ejecución transaccional (TXBI). Cada uno de los primeros 63 bits del vector indican los resultados de la ejecución de una instrucción de ramificación mientras que la CPU estaba en el modo de ejecución transaccional, como sigue:

10

Valor	Significado
0	Instrucción completada sin ramificación.
1	Instrucción completada con ramificación

15 El bit 0 representa el resultado de la primera instrucción de ramificación, el bit 1 representa el resultado de la segunda instrucción, y así sucesivamente.

20 Si se ejecutaron menos de 63 instrucciones de ramificación mientras la CPU estaba en el modo de ejecución transaccional, los bits más a la derecha que no corresponden a las instrucciones de ramificación se fijan a cero (incluyendo el bit 63). Cuando se ejecutaron más de 63 instrucciones de ramificación, el bit 63 del TXBI se fija a uno.

Los bits en el TXBI son fijados por las instrucciones que son capaces de provocar un evento de rotura, como se listan más adelante, excepto por las siguientes:

- Cualquier instrucción restringida que no provoca que se fije un bit en el TXBI.
- Para instrucciones de, por ejemplo, la z/Arquitectura, cuando el campo M<sub>1</sub> de la instrucción de RAMIFICACIÓN SOBRE LA CONDICIÓN, RAMIFICACIÓN EN RELACIÓN A CONDICIÓN, o RAMIFICACIÓN EN RELACIÓN A CONDICIÓN LARGA es cero, o cuando el campo R<sub>2</sub> de las siguientes instrucciones es cero, es dependiente de modelo si la ejecución de la instrucción provoca que se fije un bit en el TXBI.

25

- RAMIFICACIÓN Y ENLACE (BALR); RAMIFICACIÓN Y GUARDADO (BASR); RAMIFICACIÓN Y GUARDADO Y FIJACIÓN DE MODO (BASSM); RAMIFICACIÓN Y FIJACIÓN DE MODO (BSM); RAMIFICACIÓN SOBRE CONDICIÓN (BCR); Y RAMIFICACIÓN SOBRE CONTADOR (BCTR, BCTGR).

30

- Para condiciones de cancelación que fueron provocadas por una excepción de acceso a servidor, la posición 0 de bit del byte 127 se fija a uno. Para todas las demás condiciones de cancelación, la posición 0 de bit del byte 127 se fija a cero.

35

- Para condiciones de cancelación 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 cancelación que fueron detectadas por la LSU, se reserva el byte 127.

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

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

Según es observado por otras CPU y el subsistema de I/O, el almacenamiento del TDB o los TDB durante una cancelación de transacción es una referencia de acceso múltiples que ocurre después de cualquier almacenamiento no transaccional.

Una transacción se puede cancelar debido a causas que están fuera del alcance de la configuración inmediata en la que se ejecuta. Por ejemplo, los eventos transitorios reconocidos por un hipervisor (tal como el LPAR o el z/VM) pueden provocar que una transacción sea cancelada.

5 La información proporcionada en el bloque de diagnóstico de transacciones está destinada a propósitos de diagnóstico y es sustancialmente correcta. Sin embargo, ya que una cancelación puede haber sido provocada por un evento fuera del alcance de la configuración inmediata, la información tal como el código de cancelación o la identificación de interrupción de programa puede no reflejar de manera precisa las condiciones dentro de la configuración, y por tanto, no se debería usar para determinar la acción de programa.

10 Además de la información de diagnóstico guardada en el TDB, cuando se cancela 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 0 de control, como el control efectivo de operación de punto flotante (F) son uno, el código de excepción de datos (DXC) se coloca dentro del byte 2 del registro de control de punto flotante (FPCR), independientemente de si se aplica filtrado a la condición de excepción de programa. Cuando se cancela una transacción, y ya sea uno o  
15 ambos de entre el control de registro AFP o el control efectivo de operación de punto flotante son cero, el DXC no se ubica dentro del FPCR.

En una realización, tal como se indica en la presente memoria, cuando se instala la función de ejecución transaccional, se proporcionan las siguientes instrucciones generales.

- EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES
- ALMACENAMIENTO NO TRANSACCIONAL
- 20 • CANCELACIÓN DE TRANSACCIÓN
- INICIO DE TRANSACCIÓN
- FIN DE TRANSACCIÓN

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

25 Cuando se emite en el modo de ejecución transaccional restringido, el intento de ejecución de instrucciones restringidas puede resultar también en una interrupción de programa de transacción restringida, o puede resultar en la ejecución que procede como si la transacción no fuera restringida.

30 En un ejemplo de la z/Arquitectura, las instrucciones restringidas incluyen, como ejemplos, las siguientes instrucciones no privilegiadas: COMPARACIÓN E INTERCAMBIO Y ALMACENAMIENTO; MODIFICACIÓN DE LOS CONTROLES DE INSTRUMENTACIÓN DEL TIEMPO DE EJECUCIÓN; REALIZACIÓN DE OPERACIÓN DE BLOQUEO, PREEXTRACCIÓN DE DATOS (RELATIVOS LARGOS); cuando el código en el campo M<sub>1</sub> es 6 o 7; ALMACENAMIENTO DE CARACTERES BAJO MÁSCARA SUPERIOR, cuando el campo M<sub>3</sub> es cero y el código en el campo R<sub>1</sub> es 6 o 7; LISTADO DE FUNCIONES DE ALMACENAMIENTO EXTENDIDO; ALMACENAMIENTO DE  
35 LOS CONTROLES DE INSTRUMENTACIÓN DEL TIEMPO DE EJECUCIÓN; LLAMADA A SUPERVISOR; y CONTROLES DE INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN DE PRUEBA.

40 En la lista anterior, COMPARACIÓN E INTERCAMBIO Y ALMACENAMIENTO y REALIZACIÓN DE OPERACIÓN DE BLOQUEO son instrucciones complejas que pueden ser implementadas de manera más eficiente mediante el uso de instrucciones básicas en el modo TX. Los casos para la PREEXTRACCIÓN DE DATOS y PREEXTRACCIÓN DE DATOS RELATIVOS LARGOS se restringen según los códigos de 6 o 7 liberen una línea de caché, necesitando la confirmación de los datos potencialmente antes de la finalización de la transacción. La LLAMADA A SUPERVISOR provoca una interrupción (que provoca que una transacción sea cancelada).

Bajo las condiciones listadas anteriormente, se restringen las siguientes instrucciones:

- 45 • RAMIFICACIÓN Y ENLACE (BALR), RAMIFICACIÓN Y GUARDADO (BASR), y RAMIFICACIÓN Y GUARDADO Y MODO DE CONFIGURACIÓN, cuando el campo R<sub>2</sub> de la instrucción no es cero y está habilitado el rastreo de ramificación.
- RAMIFICACIÓN Y GUARDADO Y MODO DE CONFIGURACIÓN y RAMIFICACIÓN Y MODO DE CONFIGURACIÓN, cuando el campo R<sub>2</sub> de la instrucción no es cero y está habilitado el rastreo de modo; MODO DE DIRECCIONAMIENTO Y CONFIGURACIÓN, cuando está habilitado el rastreo de modo.
- LLAMADA A MONITOR, cuando está habilitado el rastreo de modo.

50 La lista anterior incluye instrucciones que pueden formar entradas de trazas. Si estas instrucciones estuvieron habilitadas para ejecutar de manera transaccional y crear entradas de trazas, y la transacción posteriormente se

cancela, el puntero de tabla de trazas en el registro 12 de control se avanzaría, pero los almacenamientos a la tabla de trazas se descartarían. Esto dejaría un hueco inconsistente en la tabla de trazas. Por tanto, se restringen las instrucciones en los casos en los que formarían entradas de trazas.

5 Cuando la CPU está en el modo de ejecución transaccional, es dependiente de modelo si se restringen las siguientes instrucciones: CIFRADO DE MENSAJE; CIFRADO DE MENSAJE CON CFB; CIFRADO DE MENSAJE CON CONCATENACIÓN; CIFRADO DE MENSAJE CON CONTADOR, CIFRADO DE MENSAJE CON OFB; LLAMADA DE COMPRESIÓN; CÁLCULO DE MENSAJE DE RESUMEN INTERMEDIO; CÁLCULO DE RESUMEN DE ÚLTIMO MENSAJE; CÁLCULO DE CÓDIGO DE AUTENTICACIÓN DE MENSAJE; CONVERSIÓN DE UNICÓDIGO-16 A UNICÓDIGO-32; CONVERSIÓN DE UNICÓDIGO-16 A UNICÓDIGO-8; CONVERSIÓN DE UNICÓDIGO-32 A UNICÓDIGO-16; CONVERSIÓN DE UNICÓDIGO-32 A UNICÓDIGO-8; CONVERSIÓN DE UNICÓDIGO-8 A UNICÓDIGO-16; CONVERSIÓN DE UNICÓDIGO-8 A UNICÓDIGO-32; REALIZACIÓN DE CÁLCULO CRIPTOGRÁFICO; INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN APAGADA; e INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN ENCENDIDA.

15 Cada una de las instrucciones anteriores actualmente está implementada por el co-procesador de hardware, o ha estado en máquinas anteriores, y por tanto, se considera restringido.

Cuando el control (A) efectivo de modificación AR permitida es cero, las siguientes instrucciones están restringidas: ACCESO DE COPIA; MÚLTIPLE ACCESO DE CARGA; DIRECCIÓN DE CARGA EXTENDIDA; y ACCESO FIJADO.

20 Cada una de las instrucciones anteriores provoca que los contenidos de un registro de acceso se modifiquen. Si el control A en la instrucción INICIO DE TRANSACCIÓN es cero, entonces el programa indica de manera explícita que la modificación del registro de acceso no está permitida.

Cuando el control (F) efectivo de operación de punto flotante permitida es cero, las instrucciones de punto flotante están restringidas.

Bajo ciertas circunstancias, las siguientes circunstancias pueden estar restringidas:

25 EXTRACCIÓN DE TIEMPO DE CPU; EXTRACCIÓN DE PSW; ALMACENAMIENTO DE RELOJ; ALMACENAMIENTO DE RELOJ EXTENDIDO; y ALMACENAMIENTO DE RELOJ RÁPIDO.

30 Cada una de las instrucciones anteriores está sometida a un control de intercepción en la descripción de estado de ejecución interpretativa. Si el hipervisor ha fijado el control de intercepción para estas instrucciones, entonces su ejecución se puede prolongar debido a la implementación del hipervisor; por tanto, se consideran restringidas si se produce una intercepción.

35 Cuando se cancela una transacción no restringida debido al intento de ejecución de una instrucción restringida, el código de cancelación de transacción en el bloque de diagnóstico de transacciones se fija a 11 (instrucción restringida), y el código de condición se fija a 3, excepto lo siguiente: cuando una transacción no restringida se cancela debido al intento de ejecución de una instrucción que en otro caso resultaría en una excepción de operación privilegiada, no se puede predecir si el código de cancelación está fijado a 11 (instrucción restringida) o a 4 (interrupción de programa no filtrada que resulta del reconocimiento de la interrupción de programa de operación privilegiada). Cuando se cancela una transacción no restringida debido al intento de ejecución de la DATOS DE PREEXTRACCIÓN (RELATIVOS LARGOS) cuando el código en el campo M<sub>1</sub> es 6 o 7 o CARACTERES DE ALMACENAMIENTO BAJO MÁSCARA SUPERIOR cuando el campo M<sub>3</sub> es cero y el código en el campo R<sub>1</sub> es 6 o 40 7, es impredecible si el código de cancelación se fija a 11 (instrucción restringida) o 16 (otra caché). Cuando se cancela una transacción no restringida debido al intento de ejecución de LLAMADA A MONITOR, y se presentan tanto una condición de evento de monitor como una condición de excepción de especificación es impredecible si el código de cancelación se fija a 11 o 4, o, si se filtra la interrupción de programa, 12.

45 Se pueden restringir instrucciones adicionales en una transacción restringida. Aunque estas instrucciones actualmente no están definidas para ser restringidas en una transacción no restringida, se pueden restringir bajo ciertas circunstancias en una transacción no restringida en procesadores futuros.

50 Ciertas instrucciones restringidas pueden estar permitidas en el modo de ejecución transaccional en procesadores futuros. Por lo tanto, el programa debería no confiar en la transacción que se cancela debido al intento de ejecución de una instrucción restringida. La instrucción de CANCELACIÓN DE TRANSACCIÓN debería ser usada para provocar que se cancele una transacción de manera segura.

En una transacción no restringida, el programa debería proporcionar una ruta de código no transaccional alternativa para acomodar una transacción que se cancela debido a una instrucción restringida.

55 En funcionamiento, cuando la profundidad de anidación de transacciones es cero, la ejecución de la instrucción INICIO DE TRANSACCIÓN (TINICIO) que resulta en un código de condición cero provoca que la CPU entre en el modo de ejecución transaccional no restringido. Cuando la profundidad de anidación de transacciones es cero, la

ejecución de la instrucción de INICIO DE TRANSACCIÓN restringido (TINICIO) que resulta en un código de condición cero provoca que la CPU entre en el modo de ejecución transaccional restringido.

5 Excepto donde explícitamente se indique lo contrario, todas las reglas que aplican para la ejecución no transacción aplican también a la ejecución transaccional. Más adelante hay características adicionales de procesamiento 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 INICIO DE TRANSACCIÓN en el código de condición cero provoca que la CPU mantenga el modo de ejecución transaccional no restringido.

10 Según es observado por la CPU, las extracciones y almacenamientos hechos en el modo de ejecución de transacción no son diferentes de aquellos hechos mientras no está en el modo de ejecución transaccional. Según es observado por otras CPU y por el subsistema de I/O, todos los accesos de operando hechos mientras una CPU está en el modo de ejecución transaccional parecen ser un único bloque de acceso concurrente. Esto es, los accesos a todos los bytes dentro de una media palabra, una palabra, una doble palabra, o una cuádruple palabra son especificados para parecer ser un bloque concurrente según es observado por otras CPU y programas (por ejemplo, canales) de I/O. La media palabra, la palabra, doble palabra, o la cuádruple palabra son referidos en esta sección como un bloque. Cuando se especifica una referencia de tipo extracción para parecer ser concurrente dentro de un bloque, no se permite el acceso de almacenamiento al bloque por otra CPU o programa de I/O durante el tiempo en que los bytes contenidos en el bloque están siendo extraídos. Cuando se especifica una referencia de tipo almacenamiento para parecer ser concurrente dentro de un bloque, no se permite ningún acceso al bloque, ya sea extracción o almacenamiento, por otra CPU o programa de I/O durante el tiempo en que los bytes dentro del bloque están siendo almacenados.

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

25 La CPU abandona el modo de ejecución transaccional de manera normal por medio de una instrucción de FIN DE TRANSACCIÓN que provoca que la profundidad de anidación de transacciones cambie a cero, en cuyo caso, la transacción se completa.

30 Cuando la CPU abandona el modo de ejecución transaccional por medio de la finalización de una instrucción de FIN DE TRANSACCIÓN, todos los almacenamientos hechos mientras esté en el modo de ejecución transaccional se confirman; esto es, los almacenamientos parecen que ocurren como una única operación concurrente de bloque según es observado por las otras CPU y el subsistema de I/O.

Una transacción se puede cancelar de manera implícita por una variedad de motivos, o puede ser cancelada de manera explícita mediante la instrucción de CANCELACIÓN DE TRANSACCIÓN. Las posibles causas de ejemplo de la cancelación de transacción, el correspondiente código de cancelación, y el código de condición que se ubica dentro de la PSW de cancelación de transacción se describe a continuación.

35 Interrupción Externa: El código de cancelación de transacción se fija a 2, y el código de condición en la PSW de cancelación de transacción se fija a 2. La PSW de cancelación de transacción se almacena como la PSW antigua externa como parte del procesamiento de interrupción externo.

40 Interrupción de Programa (No Filtrada): Una condición de excepción de programa que resulta en una interrupción (esto es, una condición no filtrada) provoca que se cancele la transacción con un código 4. El código de condición en la PSW de cancelación de transacción se fija específico al código de interrupción de programa. La PSW de cancelación de transacción se almacena como la PSW antigua de programa como parte del procesamiento de interrupción de programa.

45 Una instrucción que en otro caso resultaría en una transacción que se cancela debido a una excepción de operación puede dar resultados alternos: para una transacción no restringida, la transacción puede cancelar en su lugar con código de cancelación 11 (instrucción restringida); para una transacción restringida, se puede reconocer una interrupción de programa restringido de transacción en lugar de la excepción de operación.

Cuando se reconoce un evento PER (Registro de Evento de Programa) en conjunción con cualquier condición de excepción de programa no filtrada, el código de condición se fija a 3.

50 Interrupción de Comprobación de Máquina: El código de cancelación de transacción se fija a 5, y el código de condición en la PSW de cancelación de transacción se fija a 2. La PSW de cancelación de transacción se almacena como la antigua PSW de comprobación de máquina como una parte del procesamiento de interrupción de comprobación de máquina.

Interrupción I/O: El código de cancelación de transacción se fija a 6, y el código de condición en la PSW de cancelación de transacción se fija a 2. La PSW de cancelación de transacción se almacena como la antigua PSW de I/O como parte del procesamiento de interrupción de I/O.

5 Desbordamiento de Extracción: Se detecta una condición de desbordamiento de extracción cuando la transacción intenta extraer de más ubicaciones de las que soporta la CPU. El código de cancelación de transacción se fija a 7, y el código de condición se fija a bien 2 o 3.

Desbordamiento de Almacenamiento: Se detecta una condición de desbordamiento de almacenamiento cuando la transacción intenta almacenar en más ubicaciones de las que soporta la CPU. El código de cancelación de transacción se fija a 8, y el código de condición se fija a bien 2 o 3.

10 Permitir que el código de condición sea bien 2 o 3 en respuesta a un desbordamiento de extracción o almacenamiento permite a la CPU indicar situaciones potencialmente reutilizables (por ejemplo, un código de condición 2 indica que volver a ejecutar de la transacción puede ser productiva, mientras que un código de condición de 3 no recomienda volver a ejecutar la transacción).

15 Conflicto de Extracción: Se detecta una condición de conflicto de extracción cuando otra CPU o el subsistema de I/O intentan almacenar en una ubicación que ha sido extraída de manera transaccional por esta CPU. El código de cancelación de transacción se fija a 9, y el código de condición se fija a 2.

20 Conflicto de Almacenamiento: Se detecta una condición de conflicto de almacenamiento cuando otra CPU o el subsistema de I/O intentan acceder a una ubicación que se ha almacenado durante la ejecución transaccional de esta CPU. El código de cancelación de transacción se fija a 10, y el código de condición se fija 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 cancelada. El código de cancelación de la transacción se fija a 11, y el código de condición se fija 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 resulta en una interrupción de programa de transacción restringida o una cancelación debida a una instrucción restringida. La transacción se cancela, pero el código de cancelación puede indicar cualquiera de las causas.

30 Condición de Excepción de Programa (Filtrada): Una condición de excepción de programa que no resulta en una interrupción (esto es, una condición filtrada) provoca que la transacción sea cancelada con un código de cancelación de transacción de 12. El código de condición se fija a 3.

Profundidad de Anidación Excedida: La condición de profundidad de anidación excedida se detecta cuando la profundidad de anidación de transacciones está en el valor máximo permitido para la configuración, y se ejecuta una instrucción de INICIO DE TRANSACCIÓN. La transacción se cancela con un código de cancelación de transacción de 13, y el código de condición se fija a 3.

35 Condición Relacionada a la Extracción de Caché: Una condición relacionada con las ubicaciones de almacenamiento extraídas por la transacción es detectada por la circuitería de caché de la CPU. La transacción se cancela con un código de cancelación de 14, y el código de condición se fija a bien 2 o 3.

40 Condición Relacionada al Almacenamiento de Caché: Una condición relacionada con las ubicaciones de almacenamiento almacenadas por la transacción es detectada por la circuitería de caché de la CPU. La transacción se cancela con un código de cancelación de transacción de 15, y el código de condición se fija a bien 2 o 3.

Otras condiciones de Caché: Otra condición de caché es detectada por la circuitería de caché de la CPU. La transacción se cancela con un código de cancelación de transacción de 16, y el código de condición se fija a bien 2 o 3.

45 Durante la ejecución transaccional, si la CPU accede a las instrucciones o los operandos de almacenamiento usando direcciones lógicas diferentes que se hacen corresponder a la misma dirección absoluta, es dependiente de modelo si se cancela la transacción. Si se cancela la transacción debido a accesos que usan diferentes direcciones lógicas hechas corresponder a la misma dirección absoluta, se fija el código de cancelación 14, 15, o 16, dependiendo de la condición.

50 Condición Miscelánea: Una condición miscelánea es cualquier otra condición reconocida por la CPU que provoca que se cancele la transacción. El código de cancelación de la transacción se fija a 255 y el código de condición se fija a bien 2 o 3.

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

5 Aunque anteriormente se proporcionan ejemplos, se pueden proporcionar otras causas de cancelación de transacción con los correspondientes códigos de cancelación y códigos de condición. Por ejemplo, una causa puede ser una Interrupción de Reinicio, en la cual el código de cancelación de transacción se fija a 1, y el código de condición en la PSW de cancelación de transacción se fija a 2. La PSW de cancelación de transacción se almacena como la PSW antigua de reinicio como parte del procesamiento de reinicio. Como un ejemplo adicional, una causa puede ser una condición de Llamada de Supervisor, en la que el código de cancelación se fija a 3, y el código de condición en la PSW de cancelación de transacción se fija a 3. Otros o diferentes ejemplos son posibles.

Notas:

1. La condición miscelánea puede resultar a partir de cualquiera de lo siguiente:
  - 15 • Instrucciones, tales como, en la z/Arquitectura, COMPARACIÓN Y REEMPLAZO, ENTRADA DE TABLA DE DATOS, COMPARACIÓN E INTERCAMBIO Y PURGA, ENTRADA DE TABLA DE DATOS INVALIDADA, ENTRADA DE TABLA DE PÁGINAS INVALIDADA, REALIZACIÓN DE FUNCIÓN DE GESTIÓN DE TRAMA en la cual el control NQ es cero y el control SK es uno, FIJACIÓN DE CLAVE DE ALMACENAMIENTO EXTENDIDA en la que el control NQ es cero, realizada por otra CPU en la configuración; el código de condición se fija a 2.
  - 20 • Una función operador, tal como un restablecimiento, un reinicio o una parada, o la orden de PROCESADOR DE SEÑAL equivalente es realizada en la CPU.
  - Cualquier otra condición no enumerada anteriormente; el código de condición se fija a 2 o 3.
- 25 2. La ubicación en la que los conflictos de extracción y almacenamiento son detectados puede ser cualquiera dentro de la misma línea de caché.
3. Bajo ciertas condiciones, la CPU puede no ser capaz de distinguir entre condiciones de cancelación similares. Por ejemplo, un desbordamiento de extracción o de almacenamiento pueden ser indistinguible del respectivo conflicto de extracción o de almacenamiento.
- 30 4. La ejecución especulativa de múltiples rutas de instrucción por la CPU puede resultar en una transacción que se cancela debido a condiciones de conflicto o de desbordamiento, incluso si dichas condiciones no se producen en la secuencia conceptual. Mientras que en el modo de ejecución transaccional restringido, la CPU puede inhibir de manera temporal la ejecución especulativa, permitiendo intentar completar la transacción sin detectar dichos conflictos o desbordamientos de manera especulativa.
- 35

La ejecución de una instrucción de CANCELACIÓN DE TRANSACCIÓN provoca que se cancele la transacción. El código de cancelación de transacción se fija a partir de la segunda dirección de operando. El código de condición se fija a bien 2 o 3, dependiendo de si el bit 63 de la segunda dirección de operando es cero o uno, respectivamente.

40 La FIG. 10 resume los códigos de cancelación de ejemplo almacenados en un bloque de diagnóstico de transacciones, y el correspondiente código de condición (CC). La descripción en la FIG. 10 ilustra una implementación concreta. Son posibles otros valores de implementaciones y codificaciones.

En una realización, y como se mencionó anteriormente, la función transaccional proporciona para tanto las transacciones restringidas como las transacciones no restringidas, así como el procesamiento asociado con estas. De manera inicial, se discuten las transacciones restringidas y después las transacciones no restringidas.

45 Una transacción restringida se ejecuta en modo transaccional sin una ruta de retroceso. Es un modo de procesamiento útil para funciones compactas. En ausencia de interrupciones o conflictos repetidos con otras CPU o con el subsistema de I/O (esto es, provocadas por condiciones que no permiten que la transacción se complete con éxito), una transacción restringida finalmente se completará; por tanto, no se requiere la rutina controladora de cancelaciones y no se especifica. Por ejemplo, en ausencia de una violación de una condición que no se puede abordar (por ejemplo, dividir entre 0); una condición que no permita que se complete la transacción (por ejemplo, una interrupción de temporizador que no permite que se ejecuta la instrucción; un I/O caliente; etc.); o una violación de una restricción o limitación asociada con una transacción restringida, la transacción finalmente se completará.

50



Una transacción restringida es iniciada por una instrucción restringida de INICIO DE TRANSACCIÓN (TINICIOOC) cuando la profundidad de anidación de transacciones inicialmente es cero. Una transacción restringida se somete a las siguientes restricciones, en una realización.

- 5 1. La transacción no ejecuta más de 32 instrucciones, no incluyendo las instrucciones de INICIO DE TRANSACCIÓN restringida (TINICIOOC) y de FIN DE TRANSACCIÓN.
2. Todas las instrucciones en la transacción han de estar dentro de los 256 bytes contiguos de almacenamiento, incluyendo las instrucciones de INICIO DE TRANSACCIÓN restringida (TINICIOOC) y de FIN DE TRANSACCIÓN.
- 10 3. Además de las instrucciones restringidas, las siguientes restricciones aplican a una transacción restringida.
  - a. Las instrucciones se limitan a aquellas referidas como Instrucciones Generales, incluyendo por ejemplo, la suma, resta, multiplicación, división, desplazamiento, rotación, etc.
  - 15 b. Las instrucciones de ramificación se limitan a las siguientes (las instrucciones listadas son de la z/Arquitectura en un ejemplo):
    - RAMIFICACIÓN EN RELACIÓN A CONDICIÓN en la que el campo  $M_1$  es distinto de cero y el campo  $Rl_2$  contiene un valor positivo.
    - RAMIFICACIÓN EN RELACIÓN A CONDICIÓN LARGA en la que el campo  $M_1$  es distinto de cero y el campo  $Rl_2$  contiene un valor positivo que no provoca direccionamiento envolvente.
    - 20 • COMPARACIÓN Y RAMIFICACIÓN RELATIVA, COMPARACIÓN INMEDIATA Y RAMIFICACIÓN RELATIVA, COMPARACIÓN LÓGICA Y RAMIFICACIÓN RELATIVA, y COMPARACIÓN LÓGICA INMEDIATA Y RAMIFICACIÓN RELATIVA en las que el campo  $M_3$  no es cero y el campo  $Rl_4$  contiene un valor positivo. (Esto es, sólo realiza ramificación con máscaras de ramificación distintas de cero).
    - 25 c. Excepto para el FIN DE TRANSACCIÓN y las instrucciones que provocan una serialización de un operando especificado, se restringen las instrucciones que provocan una función de serialización.
    - 30 d. Las operaciones de almacenamiento y almacenamiento (SS-), y las operaciones de almacenamiento y almacenamiento con un código de operación extendido (SSE-) están restringidas.
    - 35 e. Todas las instrucciones generales siguientes (que son de la z/Arquitectura en este ejemplo) están restringidas: SUMA DE CONTROL, MENSAJE CIFRADO; MENSAJE CIFRADO CON CFB; MENSAJE CIFRADO CON ENCADENAMIENTO; MENSAJE CIFRADO CON CONTADOR; MENSAJE CIFRADO CON OFB; COMPARACIÓN Y CREACIÓN DE PALABRAS DE CÓDIGO; COMPARACIÓN LÓGICA LARGA; COMPARACIÓN LÓGICA LARGA EXTENDIDA; COMPARACIÓN LÓGICA LARGA UNICODE; COMPARACIÓN DE CADENA LÓGICA; COMPARACIÓN HASTA SUBCADENA IGUAL; LLAMADA DE COMPRESIÓN; CÁLCULO DE RESUMEN DE MENSAJE INTERMEDIO; CÁLCULO DE ÚLTIMO MENSAJE DE RESUMEN; CÁLCULO DE CÓDIGO DE AUTENTICACIÓN DE MENSAJE; CONVERSIÓN A BINARIO; CONVERSIÓN A DECIMAL; CONVERSIÓN DE UNICODE-16 A UNICODE-32; CONVERSIÓN DE UNICODE-16 A UNICODE-8; CONVERSIÓN DE UNICODE-32 A UNICODE-16; CONVERSIÓN DE UNICODE-32 A UNICODE-8; CONVERSIÓN DE UNICODE 8 A UNICODE-16; CONVERSIÓN DE UNICODE-8 A UNICODE-32; DIVISIÓN; DIVISIÓN LÓGICA; DIVISIÓN ÚNICA; EJECUCIÓN, EJECUCIÓN EN RELACIÓN LARGA; EXTRACCIÓN DE ATRIBUTO DE CACHE; EXTRACCIÓN DE TIEMPO DE CPU; EXTRACCIÓN DE PSW; EXTRACCIÓN DE PROFUNDIDAD DE ANIDACIÓN DE TRANSACCIONES; CARGA Y SUMA; CARGA Y SUMA LÓGICA; CARGA E Y LÓGICA; CARGA Y O EXCLUSIVA; CARGA Y O LÓGICA; DIVISIÓN DE PAREJA DE CARGA; PAREJA DE CARGA DE PALABRA CUÁDRUPLE; LLAMADA A MONITOR; MOVIMIENTO LARGO; MOVIMIENTO LARGO EXTENDIDO; MOVIMIENTO LARGO UNICODE; MOVIMIENTO DE CADENA; ALMACENAMIENTO NO TRANSACCIONAL; REALIZACIÓN DE CÁLCULO CRIPTOGRÁFICO, PREESTRACCIÓN DE DATOS; PREESTRACCIÓN DE DATOS RELATIVOS LARGOS; EMISIÓN DE
    - 40
    - 45
    - 50
    - 55

5 INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN , SIGUIENTE INSTRUMENTACIÓN  
 DE TIEMPO DE EJECUCIÓN; APAGADO DE INSTRUMENTACIÓN DE TIEMPO DE  
 EJECUCIÓN; ENCENDIDO DE INSTRUMENTACIÓN DE TIEMPO DE EJECUCIÓN;  
 CADENA DE BÚSQUEDA; BÚSQUEDA; CADENA UNICÓDIGO; MODO FIJACIÓN DE  
 DIRECCIÓN; ALMACENAMIENTO DE CARACTERES BAJO MÁSCARA SUPERIOR,  
 cuando el campo M<sub>3</sub> es cero, y el código en el campo R<sub>1</sub> es 6 o 7; ALMACENAMIENTO  
 10 DE RELOJ; ALMACENAMIENTO DE RELOJ EXTENDIDO; y ALMACENAMIENTO DE  
 RELOJ RÁPIDO; LISTA DE FUNCIONES DE ALMACENAMIENTO EXTENDIDA;  
 ALMACENAMIENTO DE PAREJA A PALABRA CUÁDRUPLE, PRUEBA DE MODO DE  
 DIRECCIONAMIENTO; CANCELACIÓN DE TRANSACCIÓN; INICIO DE  
 TRANSACCIÓN (tanto TINICIO como TINICIOC); TRADUCCIÓN Y PRUEBA  
 EXTENDIDA; TRADUCCIÓN Y PRUEBA INVERSA EXTENDIDA; TRADUCCIÓN  
 EXTENDIDA; TRADUCCIÓN UNO A UNO; TRADUCCIÓN UNO A DOS; TRADUCCIÓN  
 DOS A UNO; y TRADUCCIÓN DOS A DOS.

- 15 4. Los operandos de almacenamiento de la transacción acceden a no más de cuatro palabras  
 óptuples. Nota: CARGA SOBRE CONDICIÓN y ALMACENAMIENTO SOBRE CONDICIÓN se  
 consideran para referenciar al almacenamiento independientemente del código de condición.  
 Una palabra óptuple es, por ejemplo, un grupo de 32 bytes consecutivos en una delimitación  
 de 32 bytes.
- 20 5. La transacción que se ejecuta en esta CPU, o se almacena por otras CPU o el subsistema de  
 I/O, no accede a los operandos de almacenamiento en ninguno de los bloques de 4K-bytes  
 que contienen los 256 bytes del almacenamiento comenzando don la instrucción de INICIO  
 DE TRANSACCIÓN (TINICIOC) restringida.
- 25 6. La transacción no accede a las instrucciones o los operandos de almacenamiento usando  
 diferentes direcciones lógicas que son hechas corresponder a la misma dirección absoluta.
- 30 7. Las referencias de operando hechas por la transacción han de estar dentro de una doble  
 palabra única, excepto para la MÚLTIPLE ACCESO DE CARGA, MÚLTIPLE CARGA,  
 MÚLTIPLE CARGA SUPERIOR, MÚLTIPLE ACCESO DE ALMACENAMIENTO, MÚLTIPLE  
 ALMACENAMIENTO, y MÚLTIPLE ALMACENAMIENTO SUPERIOR, las referencias al  
 operando se han de hacer dentro de una palabra óptuple única.

35 Si una transacción restringida viola cualquiera de las restricciones 1-7, listadas anteriormente, entonces bien (a) se  
 reconoce una interrupción de programa restringido de transacción, o (b) se procede a la ejecución como si la  
 transacción no estuviera restringida, excepto que violaciones de restricción adicionales pueden aún resultar en una  
 interrupción de programa restringido de transacción. Es impredecible qué acción se toma, y la acción tomada puede  
 diferir en base a qué restricción se viola.

En ausencia de violaciones de restricción, interrupciones repetidas, o conflictos con otras CPU o el subsistema de  
 I/O, una transacción restringida finalmente se completará, como se describe anteriormente.

- 40 1. La oportunidad de completar con éxito una transacción restringida mejora si la transacción  
 alcanza los siguientes criterios:
- a. Las instrucciones emitidas son menos que el máximo de 32
- b. Las referencias al operando de almacenamiento son menores que el máximo de 4  
 palabras óptuples.
- c. Las referencias al operando de almacenamiento están en la misma línea de caché.
- 45 d. Las referencias al operando de almacenamiento a las mismas ubicaciones ocurren  
 en el mismo orden para todas las transacciones.
2. Una transacción restringida no está necesariamente asegurada de que se complete con éxito  
 en su primera ejecución. Sin embargo, si una transacción restringida que no viola ninguna de  
 las restricciones listadas se cancela, la CPU emplea circuitería para asegurar que una  
 ejecución repetida de la transacción posteriormente sea exitosa.
- 50 3. Dentro de una transacción restringida, INICIO DE TRANSACCIÓN es una instrucción  
 restringida, por tanto no se puede anidar una transacción restringida.
4. La violación de cualquiera de las restricciones 1-7 anteriores por una transacción restringida  
 puede resultar en un bucle de programa.

5. Las limitaciones de una transacción restringida son similares a aquellas de un bucle de comparación e intercambio. Debido a la interferencia potencial de otras CPU y del subsistema de I/O, no existe garantía arquitectónica de que una instrucción de COMPARACIÓN E INTERCAMBIO se complete jamás con el código de condición 0. Una transacción restringida puede sufrir de una interferencia similar en la forma de cancelaciones por conflictos o interrupciones calientes de extracciones o almacenamientos.

La CPU emplea algoritmos justos para asegurar que, en ausencia de cualquier violación de restricción, finalmente se complete una transacción restringida.

6. Para determinar el número de iteraciones repetidas requeridas para completar una transacción restringida, el programa puede emplear un contador en un registro general que no esté sometido a la máscara de guardado de registro general. Un ejemplo se muestra a continuación

Bucle	LH1	15,0	Contador de reintentos a cero
	TINICIOC	0(0), X 'FE00'	Preservar GR 0-13
	AH1	15,1	Incrementar contador
	...		
	...		Código de ejecución transaccional restringido
	...		
	TFIN		Fin de transacción

\* R15 contiene ahora un contador de los intentos transaccionales repetidos.

Observe que tanto los registros 14 como 15 no se restauran en este ejemplo. Observe también que en algunos modelos, el contador en el registro 15 puede ser inferior si la CPU detecta la condición de cancelación que sigue a la finalización de la instrucción TINICIOC, pero antes de la finalización de la instrucción AH1.

Según es observado por la CPU, las extracciones y almacenamientos hechos en el modo de ejecución transaccional no son diferentes de aquellos hechos mientras no se está en el modo de ejecución de transacción.

En una realización, el usuario (esto es, el que crea la transacción) selecciona si la transacción ha de ser restringida o no. Una realización de la lógica asociada con el procesamiento de las transacciones restringidas, y, en concreto, el procesamiento asociado con una instrucción TINICIOC, se describe con referencia a la FIG. 11. La ejecución de la instrucción TINICIOC provoca que la CPU entre en el modo de ejecución transaccional restringido o se mantenga en el modo de ejecución no restringido. La CPU (esto es, el procesador) que ejecuta la TINICIOC realiza la lógica de la FIG. 11.

Referente a la FIG. 11, en base a la ejecución de una instrucción TINICIOC, se realiza una función de serialización, PASO 1100. Una función u operación de serialización incluye completar todos los accesos de almacenamiento conceptualmente anteriores (y, para la z/Arquitectura, como ejemplo las configuraciones del bit de referencia y del bit de cambio) por la CPU, según es observado por otras CPU y por el subsistema de I/O, antes de que se produzcan los accesos de almacenamiento conceptualmente posteriores (y, las configuraciones del bit de referencia y del bit de cambio relacionadas). La serialización afecta a la secuencia de accesos de la CPU para almacenar y para almacenar las claves de almacenamiento, excepto para aquellos asociados con las extracciones de la entrada de la tabla ART y de la entrada de la tabla DAT.

Según es observado por una CPU en el modo de ejecución transaccional, la serialización opera de manera normal (como se describe anteriormente). Según es observado por otras CPU y por el subsistema de I/O, se produce una operación de serialización mientras la CPU está en el modo de ejecución transacción cuando la CPU abandona el modo de ejecución transaccional, ya sea como resultado de una instrucción de FIN DE TRANSACCIÓN que disminuye la profundidad de anidación de transacción a cero (finalización normal), o como resultado de la transacción que se cancela.

Después de realizar la serialización, se toma la determinación de si se reconoce una excepción, CONSULTA 1102. Si es así, la excepción se maneja, PASO 1104. Por ejemplo, se reconoce una excepción de operación especial y la operación se suprime si el control de ejecución transaccional, el bit 8 del registro 0 de control, es 0. Como ejemplos adicionales, se reconoce una excepción de especificación y la operación se suprime, si el campo B<sub>1</sub>, bits 16-19 de la instrucción, no es cero; se reconoce una excepción de ejecución y la operación se suprime, si el TINICIOC es el objetivo de una instrucción de tipo ejecución; y se reconoce una excepción de operación y la operación se suprime, si la función de ejecución transaccional no está instalada en la configuración. Si la CPU está ya en el modo de

ejecución de transacción restringido, entonces se reconoce la excepción de programa de excepción restringida de transacción y la operación se suprime. Además, si la profundidad de anidación de transacciones, cuando es aumentada en uno, excede la máxima profundidad de transacciones dependiente de modelo, la transacción es cancelada con código de cancelación 13. Se pueden reconocer y manejar otras o diferentes excepciones.

5 Sin embargo, si no hay una excepción, entonces se toma la determinación de si la profundidad de anidación de transacciones es cero, CONSULTA 1106. Si la profundidad de anidación de transacciones es cero, entonces la dirección del bloque de diagnóstico de transacciones se considera como inválida, PASO 1108; la PSW de cancelación de transacción se fija a partir de los contenidos de la PSW actual, excepto en que la dirección de instrucción de la PSW de cancelación de transacción designa la instrucción TINICIO, en lugar de la siguiente  
 10 instrucción secuencial, PASO 1110; y los contenidos de las parejas de registros generales según son designados por la máscara de guardado de registro general se guardan en una ubicación dependiente de modelo que no es directamente accesible por el programa, PASO 1112. Además, la profundidad de anidación se fija a 1, PASO 1114. De manera adicional, el valor efectivo de la operación (F) en punto flotante permitida y los controles de filtrado de interrupción de programa (PIFC) se fijan a cero, PASO 1316. Además, el valor efectivo del control de modificación  
 15 AR permitida (A), bit 12 del campo I<sub>2</sub> de la instrucción, se determina, PASO 1118. Por ejemplo, el control A efectivo es la Y lógica del control A en la instrucción TINICIO para el nivel actual y para cualquier instrucción TINICIO externa.

Volviendo a CONSULTA 1106, si la profundidad de anidación de transacción es mayor que cero, entonces la profundidad de anidación es aumentada en 1, PASO 1120. Además, el valor efectivo de la operación (F) en punto flotante permitida se fija a cero, y el valor efectivo del control de filtrado de interrupción de programa (PIFC) no se cambia, PASO 1122. El procesamiento entonces continúa con el PASO 1118. En una realización, un inicio satisfactorio de la transacción resulta en un código de condición 0. Esto concluye una realización de la lógica asociada con la ejecución de una instrucción TINICIO.

En una realización, la comprobación de la excepción proporcionada anteriormente puede ocurrir en un orden variante. Un orden particular para la comprobación de la excepción es como sigue:

Excepciones con la misma prioridad que la prioridad de las condiciones de interrupción de programa para el caso general.

Excepción de especificación debido al campo B<sub>1</sub> que contiene un valor distinto de cero.

Cancelación debida a una profundidad de anidación de transacciones excesiva

30 Código de condición 0 debido a la finalización normal.

Además, lo siguiente aplica en una o más realizaciones:

1. Los registros designados para ser guardados por la máscara de guardado de registro general sólo se restauran si se cancela la transacción, no cuando la transacción finaliza de manera normal por medio del FIN DE TRANSACCIÓN. Sólo los registros designados por el GRSM de la instrucción de INICIO DE TRANSACCIÓN más externa se restauran sobre la cancelación.

El campo I<sub>2</sub> debería designar todas las parejas de registros que proporcionan valores de entrada que son cambiados por una transacción restringida. Por tanto, si se cancela la transacción, los valores de registro de entrada serán restaurados a sus contenidos originales cuando se vuelva a ejecutar la transacción restringida.

2. En muchos modelos, se puede realizar un rendimiento mejorado, tanto en el INICIO DE TRANSACCIÓN como cuando se cancela la transacción, especificando el mínimo número de registros que es necesario guardar y restaurar en la máscara de guardado de registro general.

3. Lo siguiente ilustra los resultados de la instrucción de INICIO DE TRANSACCIÓN (tanto de TINICIO como de TINICIO) en base a la profundidad de anidación de transacciones actual (TND) y, cuando la TND no es cero, si la CPU está en el modo de ejecución transaccional restringido o no restringido:

Instrucción TND = 0

TINICIO Entra el modo de ejecución transaccional no restringido

50 TINICIO Entra el modo de ejecución transaccional restringido

Instrucción TND > 0

TINICIO	Modo NTX	Modo CTX
	Continúa en el modo de ejecución transaccional no restringido	Excepción de transacción restringida
TINICIOC	Continúa en el modo de ejecución transacción no restringido	Excepción de transacción restringida

5

Explicación:

CTX	La CPU está en el modo de ejecución transaccional restringido
NTX	La CPU está en el modo de ejecución transaccional no restringido
TND	Profundidad de anidación de transacciones en el comienzo de la instrucción.

10 Como se describe en la presente memoria, en un aspecto, se asegura la finalización de una transacción restringida, asumiendo que no contiene una condición que la haga incapaz de ser completada. Para asegurar que se completa, el procesador (por ejemplo, la CPU) que ejecuta la transacción puede tomar ciertas acciones. Por ejemplo, si la transacción restringida tiene una condición de cancelación, la CPU puede de manera temporal:

- (a) Inhibir la ejecución fuera de orden;
- 15 (b) Inhibir que otras CPU accedan a las ubicaciones de almacenamiento conflictivas;
- (c) Inducir retardos aleatorios en el procesamiento de cancelación; y/o
- (d) Invocar otras mediciones para facilitar la finalización con éxito.

Para resumir, el procesamiento de una transacción restringida es, como sigue:

- Si ya está en el modo de TX restringido, se reconoce una excepción de transacción restringida.
- 20 • Si la TND (Profundidad de Anidación de Transacciones) > 0, la ejecución procede como si fuera una transacción no restringida
  - El Control F efectivo se fija a cero
  - El PIFC efectivo no se cambia
  - Permite a la TX no restringida externa llamar una función de servicio que pueda usar o no TX restringida
- 25 • Si la TND actual = 0:
  - La dirección del bloque de diagnóstico de transacciones es inválida
    - Ningún TDB específico de instrucción almacenado sobre la cancelación
  - La PSW de cancelación de transacción se fija a la dirección de TINICIOC
    - 30 - No la siguiente instrucción secuencial
  - Parejas de registros generales designadas por el GRSM guardadas en una ubicación dependiente de modelo no accesible por el programa.
  - Identificador de transacción creado de manera opcional (a partir del operando D<sub>2</sub>). El identificador de transacción es un identificador de la transacción. Puede ser igual a la dirección del operando de almacenamiento u otro valor.
- 35 • A Efectiva = A de TINICIOC & cualquier A externa
- TND incrementada
  - Si TND varía de 0 a 1, la CPU entra en el modo de TX restringido
  - En otro caso, la CPU se mantiene en el modo de TX no restringido
- 40 • La instrucción se completa con CC0

- Excepciones:
  - La excepción de especificación (PIC (Código de Interrupción de Programa) 0006) si el campo B<sub>1</sub> no es cero
  - 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 se emite en el modo TX restringido.
  - Excepción de operación (PIC 0001) si la función de ejecución transaccional restringida no está instalada.
  - Excepción de ejecución (PIC 0003) si la instrucción es el objetivo de una instrucción de tipo ejecución
  - Código de cancelación 13 si se excede la profundidad de anidación
- Condiciones de cancelación en la transacción restringida:
  - La PSW de cancelación apunta a la instrucción TINICIO
    - No la instrucción que la sigue
    - La condición de cancelación provoca que toda la TX sea reconducida
      - \* No fallo de ruta
  - La CPU toma mediciones especiales para asegurar la finalización con éxito de la reconducción
  - Asumiendo un conflicto, interrupción, o violación restringida, no persistente se asegura la finalización eventual de la transacción.
- Violación de restricción:
  - PIC 0018 hex – indica la violación de la restricción de transacción
  - O, la transacción se ejecuta como si fuera no restringida

Como se describió anteriormente, además del procesamiento de transacción restringida, que es opcional, en una realización, la función transaccional proporciona también procesamiento de transacción no restringida. Detalles adicionales respecto al procesamiento de transacciones no restringidas, y, en concreto, el procesamiento asociado con una instrucción TINICIO se describen con referencia a la FIG. 12. La ejecución de la instrucción TINICIO provoca que bien la CPU entre o se mantengan en el modo de ejecución transaccional no restringido. La CPU (esto es, el procesador) que ejecuta el TINICIO realiza la lógica de la FIG. 12.

Referente a la FIG. 12, en base a la ejecución de la instrucción TINICIO, se realiza una función de serialización (descrita anteriormente), PASO 1200. Después de realizar la serialización, se toma la determinación de si se reconoce la excepción, CONSULTA 1202. Si es así, entonces se maneja la excepción, PASO 1204. Por ejemplo, se reconoce una excepción de operación especial y la operación se suprime si el control de ejecución transacción, bit 8 del registro 0 de control, es cero. Además, se reconoce una excepción de especificación y se suprime la operación si el control de filtrado de interrupción de programa, bits 14-15 del campo I<sub>2</sub> de la instrucción, contiene el valor 3; o la primera dirección de operando no designa un límite de doble palabra. Se reconoce una excepción de operación y se suprime la operación, si la función de ejecución transaccional no está instalada en la configuración; y se reconoce una excepción de ejecución y se suprime la operación si el TINICIO es el objetivo de una instrucción de tipo ejecución. De manera adicional, si la CPU está en el modo de ejecución transaccional restringido, entonces se reconoce una excepción de programa de excepción restringida de transacción. Además, si la profundidad de anidación de transacción, al ser aumentada en 1, excede la máxima profundidad de anidación de transacciones dependiente de modelo, la transacción se cancela con código de cancelación 13.

Aún más, cuando el campo B<sub>1</sub> de la instrucción no es cero y la CPU no está en el modo de ejecución transaccional, esto es, la profundidad de anidación de transacciones es cero, entonces se determina la accesibilidad de almacenamiento al primer operando. Si el primer operando no puede ser accedido para los almacenamientos, entonces se reconoce una excepción de acceso y la operación bien se anula, se suprime, o se termina, dependiendo de la condición de excepción de acceso específica. De manera adicional, se reconoce cualquier evento de alteración de almacenamiento PER para el primer operando. Cuando el campo B<sub>1</sub> no es cero y la CPU está ya en el modo de ejecución transaccional, es impredecible si se determina la accesibilidad de almacenamiento al primer operando, y si se detectan los eventos de alteración de almacenamiento PER para el primer operando. Si el campo B<sub>1</sub> es cero, entonces el primer operando no se accede.

Además de la comprobación de excepción, se toma la determinación de si la CPU está en el modo de ejecución transaccional (esto es, la profundidad de anidación de transacciones es cero), CONSULTA 1206. Si la CPU no está en el modo de ejecución transaccional, entonces se guardan los contenidos de las parejas seleccionadas de registros generales, PASO 1208. En particular, los contenidos de las parejas de registros generales designadas por la máscara de guardado de registro general se guardan en una ubicación dependiente de modelo que no es directamente accesible por el programa.

Además, se toma la determinación de si el campo  $B_1$  de la instrucción es cero, CONSULTA 1210. Si el campo  $B_1$  no es igual a cero, la dirección del primer operando se coloca en la dirección del bloque de diagnóstico de transacciones, PASO 1214, y la dirección del bloque de diagnóstico de transacciones es válido. Además, la PSW de cancelación de transacción se fija a partir de los contenidos de la PSW actual, PASO 1216. La dirección de instrucción de la PSW de cancelación de transacción designa la siguiente instrucción secuencial (esto es, la instrucción que sigue a la TINICIO más externa).

Además, se toma la determinación del valor efectivo del control (A) de modificación AR permitida, bit 12 del campo  $I_2$  de la instrucción, PASO 1218. El control A efectivo es la Y lógica del control A en la instrucción TINICIO para el nivel actual y para todos los niveles externos. De manera adicional, se determina, un valor efectivo del control (F) de operación de punto flotante permitida, bit 13 del campo  $I_2$  de la instrucción, PASO 1220. El control F efectivo es la Y lógica del control F en la instrucción TINICIO para el nivel actual y para todos los niveles externos. Además, se determina el valor efectivo del control de filtrado de interrupción de programa (PIFC), bits 14-15 del campo  $I_2$  de la instrucción, PASO 1222. El valor PIFC efectivo es el máximo valor en la instrucción TINICIO para el nivel actual y para todos los niveles externos.

De manera adicional, se añade un valor de uno a la profundidad de anidación de transacciones, PASO 1224, y la instrucción se completa con código de condición de fijación 0, PASO 1226. Si la profundidad de anidación de transacciones varía de cero a uno, la CPU entra en el modo de ejecución transaccional; en otro caso, la CPU se mantiene en el modo de ejecución transaccional.

Volviendo a la CONSULTA 1210, si  $B_1$  es igual a cero, entonces la dirección del bloque de diagnóstico de transacciones es inválida, PASO 1211, y el procesamiento continúa con el PASO 1218. De manera similar, si la CPU está en el modo de ejecución transaccional, CONSULTA 1206, el procesamiento continúa con el PASO 1218.

El Código de Condición Resultante de la ejecución de TINICIO incluye, por ejemplo:

	0	Transacción inicial con éxito
30	1	–
	2	–
	3	–

Las Excepciones de Programa incluyen, por ejemplo:

- Acceso (almacenamiento, primer operando)
- Operación (función de ejecución transaccional no instalada)
- Operación especial
- Especificación
- Restricción de transacción (debido a una instrucción restringida)

En una realización, la comprobación de excepción proporcionada anteriormente puede producirse en un orden variante. Un orden particular a la comprobación de excepción es como sigue:

- Excepciones con la misma prioridad que la prioridad de las condiciones de interrupción de programa para el caso general.
- Excepción de especificación debido a un valor PIFC reservado.
- Excepción de especificación debido a una dirección de primer operando no en el límite de doble palabra.
- Excepción de acceso (cuando el campo  $B_1$  no es cero).
- Cancelación debida a exceder la máxima profundidad de anidación de transacciones.

- Código de condición 0 debido a la finalización normal.

Notas:

1. Cuando el campo B<sub>1</sub> no es cero, aplica lo siguiente:

- 5
- Se ha de proporcionar un bloque de diagnóstico de transacciones (TDB) accesible cuando se inicie una transacción más externa - incluso si la transacción nunca se cancela.
  - Ya que es impredecible si la accesibilidad del TDB se comprueba para transacciones anidadas, se debería proporcionar un TDB accesible para cualquier instrucción TINICIO anidada.
  - El rendimiento de cualquier TINICIO en el que el campo B<sub>1</sub> sea distinto de cero, y el rendimiento de cualquier procesamiento de cancelación que ocurra para una transacción que fue iniciada por la TINICIO más externa en la que el campo B<sub>1</sub> no es cero, puede ser inferior que cuando el campo B<sub>1</sub> es cero.
- 10

2. Los registros designados para ser guardados por la máscara de guardado de registro general sólo se restauran, en una realización, si la transacción se cancela, no cuando la transacción finaliza de manera normal por medio del FIN DE TRANSACCIÓN. Sólo los registros designados por el GRSM de la instrucción de INICIO DE TRANSACCIÓN más externa se restauran sobre la cancelación.

15

El campo I<sub>2</sub> debería designar todas las parejas de registros que proporcionan valores de entrada que son cambiados por la transacción. Por tanto, si la transacción se cancela, los valores de registro de entrada serán restaurados a sus contenidos originales cuando se introduzca el controlador de cancelaciones.

3. Se espera que la instrucción de INICIO DE TRANSACCIÓN (TINICIO) sea seguida por una instrucción de ramificación condicional que determinará si la transacción fue iniciada de manera exitosa.

20

4. Si una transacción se cancela debido a condiciones que no resultan en una interrupción, la instrucción designada por la PSW de cancelación de transacción recibe el control (esto es, la instrucción que sigue la de INICIO DE TRANSACCIÓN más externa (TINICIO)). Además del código de condición fijado por la instrucción de INICIO DE TRANSACCIÓN (TINICIO), se fijan también los códigos de condición 1-3 cuando se cancela una transacción.

25

Por lo tanto, la secuencia de instrucciones que sigue la instrucción de INICIO DE TRANSACCIÓN (TINICIO) más externa debería ser capaz de acomodar todos los cuatro códigos de condición, incluso aunque la instrucción TINICIO sólo fije el código 0, en este ejemplo.

5. En muchos modelos, se puede conseguir un rendimiento mejorado, tanto en el INICIO DE TRANSACCIÓN como cuando se cancela una transacción, especificando el número mínimo de registros necesarios para ser guardados y restaurados en la máscara de guardado de registros generales.

30

6. Mientras que en el modo de ejecución transaccional no restringido, un programa puede llamar a una función de servicio que pueda alterar los registros de acceso o los registros de punto flotante (incluyendo el registro de control de punto flotante). Aunque dicha rutina de servicio pueda guardar los registros alterados en la entrada y restaurarlos en la salida, la transacción se puede cancelar antes de la salida normal de la rutina. Si el programa de llamada no hace provisión para preservar estos registros mientras la CPU está en el modo de ejecución transaccional no restringido, puede no ser capaz de tolerar la alteración de la función de servicio de los registros.

35

Para evitar la alteración inadvertida de los registros de acceso mientras está en el modo de ejecución transaccional no restringido, el programa puede fijar el control de modificación AR permitido, bit 12 del campo I<sub>2</sub> de la instrucción INICIO DE TRANSACCIÓN, a cero. De manera similar, para evitar la alteración inadvertida de los registros de punto flotante, el programa puede fijar el control de operación de punto flotante permitida, bit 13 del campo I<sub>2</sub> de la instrucción TINICIO, a cero.

40

7. Las condiciones de excepción de programa reconocidas durante la ejecución de la instrucción de INICIO DE TRANSACCIÓN (TINICIO) están sometidas al control efectivo de filtrado de interrupción de programa fijado por cualesquiera instrucciones TINICIO externas. Las condiciones de excepción de programa reconocidas durante la ejecución de la instrucción TINICIO más externa no están sometidas al filtrado.

45

8. Para actualizar múltiples ubicaciones de almacenamiento de una manera serializada, las secuencias de código convencionales pueden emplear una palabra de bloqueo (semáforo). Si (a) se usa la ejecución transaccional para implementar actualizaciones de múltiples ubicaciones de almacenamiento, (b) el programa proporciona también una ruta de "retroceso" para ser invocada si se cancela la transacción, y (c) la ruta de retroceso emplea una palabra de bloqueo, entonces la ruta de ejecución transaccional debería probar también la disponibilidad del bloqueo, y, si el bloqueo no está disponible, finalizar la transacción por medio de la instrucción de FIN DE

50



TRANSACCIÓN y ramificar a la ruta de retroceso. Esto asegura el acceso consistente a los recursos serializados, independientemente de si se actualizan de manera transaccional.

De manera alternativa, el programa podría cancelarse si el bloqueo no está disponible, sin embargo el procesamiento de cancelación puede ser significativamente más lento que simplemente finalizando la transacción a través de un TFIN.

9. Si el control efectivo de control de filtrado de interrupción de programa (PIFC) es mayor que cero, la CPU filtra más interrupciones de programa de excepción de datos. Si el control (F) efectivo de operación de punto flotante permitida es cero, el código (DXC) de excepción de datos no será fijado en el registro de control de punto flotante como resultado de una cancelación debido a una condición de excepción de programa de excepción de datos. En este escenario (el filtrado se aplica y el control F efectivo es cero), la única ubicación en la que el DXC es inspeccionado es en el TDB de TINICIO especificado. Si el controlador de cancelaciones de programa ha de inspeccionar el DXC en dicha situación, el registro general B<sub>1</sub> debería ser distinto de cero, de manera tal que se fija una dirección de bloque de diagnóstico de transacciones (TDBA) válida.
10. Si existe una alteración de almacenamiento PER o cero condiciones de detección de dirección para el TDB de TINICIO especificado de la instrucción TINICIO más externa, y la supresión del evento PER no aplica, el evento PER es reconocido durante la ejecución de la instrucción, provocando por tanto que se cancele la transacción de manera inmediata, independientemente de si existe cualquier otra condición de cancelación.

En una realización, la instrucción TINICIO fija de manera implícita la dirección de cancelación de transacción para ser la siguiente instrucción secuencial que sigue el TINICIO. Esta dirección está destinada a ser una instrucción de ramificación condicional que determina si ramificar o no dependiendo del código de condición (CC). Un TINICIO con éxito fija un CC0, mientras que una transacción cancelada fija un CC1, CC2, o CC3.

En una realización, la instrucción TINICIO proporciona un operando de almacenamiento opcional que designa la dirección de un bloque de diagnóstico de transacciones (TDB) dentro del cual se almacena la información si se cancela la transacción.

Además, proporciona un operando inmediato que incluye lo siguiente:

Una máscara de guardado de registro general (GRSM) que indica que parejas de registros generales se han de guardar al comienzo de la ejecución transaccional y restaurar si se cancela la transacción;

Un bit (A) para permitir la cancelación de la transacción si la transacción modifica los registros de acceso;

Un bit (F) para permitir la cancelación de la transacción si la transacción intenta ejecutar instrucciones en punto flotante; y

Un control de filtrado de interrupción de programa (PIFC) que permita niveles de transacción individual para puentear la presentación real de una interrupción de programa si se cancela la transacción.

Los controles A, F, y PIFC pueden ser diferentes a varios niveles de anidación y restaurados al nivel anterior cuando se finalicen los niveles de transacción internos.

Además, el TINICIO (o en otra realización, el TINICIOOC) se usa para formar un identificador de transacción. De manera opcional, el identificador se puede hacer corresponder con un identificador creado por la instrucción TFIN. Para cada instrucción TINICIO (o TINICIOOC), como ejemplo, se crea un identificador a partir de la primera dirección de operando. Este identificador puede crearse de manera independiente o si el registro de base es cero (a diferencia de la configuración de dirección TDB que sólo se produce cuando el registro base no es cero). Para cada instrucción de FIN DE TRANSACCIÓN ejecutada con un registro base distinto de cero, se crea un identificador similar a partir de su operando de almacenamiento. Si los identificadores no coinciden, se puede reconocer una excepción de programa para alertar al programa de una instrucción no asociada.

La correspondencia de identificadores proporciona un mecanismo destinado a mejorar la fiabilidad del software asegurando que la instancia TFIN se asocia de manera apropiada con un TINICIO (o TINICIOOC). Cuando una instrucción TINICIO se ejecuta con un nivel de anidación concreto, se crea un identificador a partir de la dirección del operando de almacenamiento que identifica esta instancia de una transacción. Cuando se ejecuta una instrucción TFIN correspondiente, se crea un identificador a partir de la dirección del operando de almacenamiento de la instrucción, y la CPU compara el identificador de inicio para el nivel de anidación con el identificador de fin. Si los identificadores no coinciden, se reconoce una condición de excepción. Un modelo puede implementar la correspondencia de identificadores para sólo un cierto número de niveles de anidación (o para ningún nivel de anidación). El identificador puede no involucrar todos los bits de la dirección del operando de almacenamiento, o los bits se pueden combinar a través de hashing u otros métodos. Un identificador puede estar creado mediante la instrucción TINICIO incluso si su operando de almacenamiento no es accedido.

Para resumir, el procesamiento de una transacción no restringida es, como sigue:

- Si TND = 0:
  - Si  $B_1 \neq 0$ , la dirección de bloque de diagnóstico de transacciones fijada a partir de la primera dirección de operando.
  - La PSW de cancelación de transacción fijada a la siguiente dirección de instrucción secuencial.
  - Las parejas de registros generales designadas mediante el campo  $I_2$  se guardan en la ubicación dependiente de modelo.
    - No directamente accesibles por el programa
- Calculados los controles efectivos PIFC, A, & F
  - A Efectivo = TINICIO A & cualquier A externo
  - F Efectivo = TINICIO F & cualquier F externo
  - PIFC Efectivo = max (TINICIO PIFC, cualquier PIFC externo)
- Aumentada la profundidad de anidación de transacción (TND)
- Si la TND varía de 0 a 1, la CPU entra en el modo de ejecución transaccional
- El código de condición se fija a cero
  - Cuando la instrucción que sigue a TINICIO recibe el control:
    - Éxito de TINICIO indicado por CC0
    - Transacción de cancelación indicado por un CC distinto de cero
  - Código de cancelación 13 si se excede la profundidad de anidación
  - Excepción de acceso (una de las diversas PIC) si el campo  $B_1$  no es cero, y el operando de almacenamiento no se puede acceder para una operación de almacenamiento.
  - Excepción de ejecución (PIC 0003) si la instrucción TINICIO es el objetivo de una instrucción de tipo ejecución
  - Excepción de operación (PIC 0001) si la función de ejecución transaccional no está instalada
  - PIC 0006 si bien
    - PIFC es inválido (valor de 3)
    - Dirección del segundo operando no alineada con doble palabra
  - PIC 0013 hex si el control de ejecución transaccional (CR0.8) es cero
  - PIC 0018 hex si emitido en el modo de TX restringido

Como se indica anteriormente, una transacción, ya sea restringida o no restringida, puede ser finalizada por una instrucción de FIN DE TRANSACCIÓN (TFIN). Se describen detalles adicionales respecto al procesamiento de una instrucción de fin de transacción (TFIN) con referencia a la FIG. 13. La CPU (esto es, el procesador) que ejecuta el TINICIO realiza la lógica de la FIG. 13.

- 35 Referente a la FIG. 13, inicialmente, en base al procesador que obtiene (por ejemplo, extrayendo, recibiendo, etc.) la instrucción TFIN, se hacen diversas comprobaciones de excepción y si existe una excepción, CONSULTA 1300, entonces la excepción se maneja, PASO 1302. Por ejemplo, si el FIN DE TRANSACCIÓN es el objetivo de una instrucción de tipo ejecución, se suprime la operación y se reconoce una excepción de ejecución, y se reconoce una excepción de operación especial y se suprime la operación si el control de ejecución transaccional, bit 8 de CR0, es cero. Aún más, se reconoce una excepción de operación y se suprime la operación, si la función de ejecución transaccional no está instalada en la configuración.
- 40

Volviendo a la CONSULTA 1300, si no se reconoce una excepción, entonces se disminuye la profundidad de anidación de transacción (por ejemplo, por uno), PASO 1304. Se toma la determinación de si la profundidad de anidación transaccional es cero que sigue a la disminución, CONSULTA 1306. Si la profundidad de anidación de

transacción es cero, entonces todos los accesos de almacenamiento hechos por la transacción (y otras transacciones dentro de la anidación de transacciones, si hay alguna, de las cuales esta transacción es parte) se confirman, PASO 1308. Además, la CPU abandona el modo de ejecución transaccional, PASO 1310, y la instrucción se completa, PASO 1312.

- 5 Volviendo a la CONSULTA 1306, si la profundidad de anidación de transacción no es igual a cero, entonces la instrucción FIN DE TRANSACCIÓN sólo se completa.

Si la CPU está en el modo de ejecución de transacción al comienzo de la operación, el código de condición se fija a 0; en otro caso el código de condición se fija a 2.

- 10 Se observa que el control efectivo (F) de operación de punto flotante permitida, el control (A) de modificación AR permitida, y el control de filtrado de interrupción de programa (PIFC) se reinician a sus respectivos valores antes de la instrucción de INICIO DE TRANSACCIÓN que inició el nivel que es finalizado. Además, se realiza una función de serialización a la finalización de la operación.

La instrucción PER que extrae y finaliza la transacción de los eventos que se reconocen a la finalización de la instrucción de FIN DE TRANSACCIÓN más externa no resulta en que se cancela la transacción.

- 15 En un ejemplo, la instrucción TFIN incluye también un campo  $B_2$  de base y un campo  $D_2$  de desplazamiento, que se combinan (por ejemplo, se suman) para crear una segunda dirección de operando. En este ejemplo, se puede realizar la correspondencia de identificadores. Por ejemplo, cuando  $B_2$  no es cero, los bits seleccionados de la segunda dirección de operando se hacen corresponder contra un identificador de transacción creado mediante el correspondiente TINICIO. Si hay un desajuste, hay una excepción (por ejemplo, la PIC 0006).

- 20 Además de lo anterior, se puede cancelar una transacción de manera implícita o explícita mediante una instrucción de CANCELACIÓN de TRANSACCIÓN. Cancelar una transacción mediante TCANCELACIÓN o de otra forma incluye realizar un número de pasos. Un ejemplo de los pasos para cancelar el procesamiento, en general, se describe con referencia a la FIG. 14. Si existe una diferencia en el procesamiento en base a si la cancelación es iniciada mediante TCANCELACIÓN o de otra forma, se indica en la descripción de más adelante. En un ejemplo, un procesador (por ejemplo, una CPU) realiza la lógica de la FIG. 14.

- 25 Referente a la FIG. 14, inicialmente, en base a la ejecución de la instrucción de TCANCELACIÓN o a una cancelación implícita, no se confirman accesos de almacenamiento no transaccionales hechos mientras la CPU estaba en el modo de ejecución transaccional, PASO 1400. Otros almacenamientos (por ejemplo, los almacenamientos transaccionales) hechos mientras que la CPU estaba en el modo de ejecución transaccional se descartan, PASO 1402.

- 30 La CPU abandona el modo de ejecución transaccional, PASO 1404, y los almacenamientos posteriores se producen de manera no transaccional. La PSW actual es reemplazada con los contenidos de la PSW de cancelación de transacción, excepto por que el código de condición se fija como se describe anteriormente (distinta a la situación de más adelante, en la que, si la TDBA es válida, pero el bloque no es accesible, entonces  $CC = 1$ ), PASO 1406. Como parte de o después de cancelar el procesamiento, el procesamiento ramifica a la ubicación especificada de la PSW de cancelación de transacción para realizar una acción. En un ejemplo en el que la transacción es una transacción restringida, la ubicación es la instrucción TINICIO y la acción es la re-ejecución de esa instrucción; y en un ejemplo adicional en el que la transacción es una transacción no restringida, la ubicación es la instrucción después de TINICIO, y la acción es la ejecución de esa instrucción, que puede ser, por ejemplo, una ramificación a un controlador de cancelación.

- 35 A continuación, se toma la determinación de si la dirección del bloque de diagnóstico de transacciones es válida, CONSULTA 1408. Cuando la dirección del bloque de diagnóstico de transacciones es válida, la información de diagnóstico que identifica la razón para la cancelación y los contenidos de los registros generales se almacena en el bloque de diagnóstico de transacciones de TINICIO especificado, PASO 1410. Los campos del TDB almacenados y las condiciones bajo las cuales se almacenan se describen anteriormente con referencia al bloque de diagnóstico de transacciones.

Si la dirección del bloque de diagnóstico de transacciones es válida, pero el bloque ha resultado inaccesible, después de la ejecución de la instrucción de TINICIO más externa, el bloque no es accedido, y aplica el código de condición 1.

- 40 50 Para transacciones que se cancelan debido a condiciones de excepción de programa que resultan en una interrupción, se almacena el TDB de interrupción de programa.

Volviendo a CONSULTA 1408, si la dirección del bloque de diagnóstico de transacciones no es válida, no se almacena el TDB de TINICIO específico y aplica el código de condición 2 o 3, dependiendo de la razón de la cancelación.

Además de lo anterior, la profundidad de anidación de transacciones se fija igual a cero, PASO 1412. Además, cualesquiera parejas de registros generales designados para ser guardados mediante la instrucción de TINICIO más externa son restaurados, PASO 1414. Las parejas de registros generales que no fueron designados para ser guardados mediante la instrucción TINICIO más externa no se restauran cuando se cancela una transacción.

5 Además, se realiza una función de serialización, PASO 1416. Una función u operación de serialización incluye completar de manera conceptual todos los accesos de almacenamiento anteriores (y, para la z/Arquitectura, como ejemplo, las configuraciones del bit de referencia y del bit de cambio relacionados) por la CPU, según es observado por otras CPU y por el subsistema de I/O, antes de que se produzcan de manera conceptual los accesos de almacenamiento posteriores (las configuraciones del bit de referencia y del bit de cambio relacionados). La serialización lleva a cabo todos los accesos de la CPU al almacenamiento y a las claves de almacenamiento, excepto para aquellos asociados con la extracción de la entrada de la tabla ART y de la tabla DAT.

Según es observado por una CPU en el modo de ejecución transaccional, la serialización opera de manera normal (como se describió anteriormente). Según es observado por otras CPU y por el subsistema de I/O, una operación de serialización realizada mientras una CPU está en el modo de ejecución transaccional ocurre cuando la CPU abandona el modo de ejecución transaccional, ya sea como resultado de una instrucción de FIN DE TRANSACCIÓN que disminuye la profundidad de anidación de transacciones a cero (finalización normal) o como resultado de que la transacción se cancele.

Para el procesamiento de cancelación iniciado de manera distinta que mediante TCANCELACIÓN, si la transacción se cancela debido a una condición de excepción que resulta en una interrupción, CONSULTA 1418, los códigos o parámetros de interrupción asociados con la interrupción se almacenan en las ubicaciones de almacenamiento asignados correspondientes al tipo de interrupción, PASO 1420. Además, la PSW actual, como se indica anteriormente, se almacena en la antigua PSW de interrupción, PASO 1422. Después de esto, o si la transacción no fue cancelada debido a una condición de excepción que resultó en una interrupción, la instrucción finaliza con el código de condición cero.

Además de lo anterior, en una realización para la ejecución interpretativa de la z/Arquitectura, cuando la CPU está en el modo de ejecución transaccional, y se produce una condición invitada que resultaría normalmente en los códigos de intercepción 4, 12, 44, 56, 64, 68 o 72, no se produce una intercepción. En su lugar, la CPU se mantiene en el modo de ejecución interpretativo, y las condiciones de cancelación se indican a la invitada como siguen:

- Para una transacción no restringida, la transacción se cancela debido a una instrucción restringida (código de cancelación 11). Si un evento PER concurrente fue detectado y la CPU está preparada para el PER, se produce una interrupción de programa con un código de interrupción 0280 hex.
- Para una transacción restringida, se reconoce una excepción restringida de transacción. Si un evento PER concurrente fue detectado y la CPU está habilitada para el PER, se produce una interrupción de programa con código de interrupción 0298 hex.

35 Cuando se cancela una transacción debido a una condición de excepción de programa, el filtrado de interrupción de programa puede inhibir la presentación de una interrupción. Para las interrupciones de programa que pueden resultar en intercepción, el filtrado también inhibe la intercepción.

Detalles adicionales respecto al procesamiento transaccional se describen con referencia a las FIG. 15A-15B. En una realización, esta lógica es realizada por un procesador.

40 Referente a la FIG. 15A, inicialmente, se inicia una transacción a través de una instrucción de INICIO DE TRANSACCIÓN, PASO 1500. Por ejemplo, si la transacción es una transacción restringida, entonces la instrucción TINICIO se usa para iniciar la transacción. Sin embargo, si la transacción es una transacción no restringida, entonces se emplea la instrucción TINICIO.

45 Durante la ejecución de la transacción, si se cancela la transacción, CONSULTA 1502, se toma la determinación según el tipo de transacción, PASO 1504. Esto es porque el procesamiento asociado con la cancelación depende del tipo de transacción, por ejemplo, una transacción restringida o una transacción no restringida.

Se toma la determinación de si el tipo de transacción es un primer tipo, tal como una transacción restringida, CONSULTA 1506. Si la transacción es una transacción restringida, entonces en base a la cancelación, uno o más registros especificados en la instrucción TINICIO son restaurados y se descartan los almacenamientos transaccionales, PASO 1508. Además, se toma la determinación de si la cancelación fue debida a una interrupción/intercepción, CONSULTA 1510. Si fue debido a una interrupción/intercepción, entonces se gestiona la interrupción/intercepción, PASO 512. Por ejemplo, si la cancelación fue debida a una interrupción, la PSW de cancelación de transacción (que apunta al INICIO DE TRANSACCIÓN y contiene el código de condición que indica la probabilidad de un reintento exitoso) se almacena como la PSW antigua de interrupción. Existen PSW antiguas y nuevas de interrupción separadas para los diversos tipos de interrupción, por ejemplo, interrupciones externas, de programa, de I/O, de comprobación de máquina. La PSW antigua de interrupción indica que estuvo en control en el

momento de la interrupción. Entonces, se transfiere el control a una nueva PSW de interrupción (por ejemplo, la dirección del controlador de interrupción). Además, si la cancelación fue debida a una intercepción, entonces la PSW de cancelación de transacción se guarda en la descripción de estado, y ocurre una interrupción que provoca que la CPU abandone el modo de ejecución interpretativo. Sin embargo, si la cancelación no está debida a una interrupción/intercepción, la ejecución se reanuda en la instrucción de INICIO DE TRANSACCIÓN, PASO 1516.

Volviendo a CONSULTA 1506, si el tipo de transacción es un segundo tipo, tal como una transacción no restringida, entonces la información se almacena de manera opcional en uno o más bloques de diagnóstico de transacciones, PASO 1520. Por ejemplo, si se proporciona una dirección válida de un bloque de diagnóstico de transacciones en la instrucción TINICIO, entonces la información de diagnóstico se almacena en un TDB específico de programa (también conocido como TDB-TINICIO). Además, si la transacción se cancela debido a ciertas condiciones de excepción de programa no filtradas, entonces la información se almacena en un TDB de interrupción de programa; y si se cancela una transacción debido a ciertas condiciones de intercepción, entonces se almacena la información en un TDB de intercepción.

De manera adicional, el procesamiento de cancelación para una transacción no restringida incluye restaurar uno o más registros específicos en la instrucción TINICIO y descartar los almacenamientos transaccionales, CONSULTA 1522. Además, se toma la determinación de si la cancelación fue debida a una interrupción/intercepción, CONSULTA 1524. Si fue debida a una interrupción/intercepción, entonces se gestiona la interrupción/intercepción, como anteriormente, PASO 1526. Sin embargo si la cancelación no es debida a una interrupción/intercepción, se reanuda la ejecución en la instrucción siguiente a la instrucción de INICIO DE TRANSACCIÓN (por ejemplo, la primera instrucción que sigue al TINICIO; esto es, la siguiente instrucción secuencia en la transacción directamente a continuación del TINICIO en el orden de programa), PASO 1530.

Detalles adicionales respecto a la reanudación de la ejecución después de la instrucción de INICIO DE TRANSACCIÓN se describen con referencia a la FIG. 15B.

Inicialmente, el control de recepción de instrucción comprueba el código de condición de la transacción (por ejemplo, el código de condición de TINICIO) para determinar por qué está recibiendo el control, PASO 1550. Si el código de condición indica que la instrucción siguiente a la instrucción de INICIO DE TRANSACCIÓN está recibiendo el control debido a una cancelación (por ejemplo, un código de condición distinto de cero), entonces se toma la determinación de si el código de condición indica la re-ejecución, CONSULTA 1556. Por ejemplo, si el código de condición es de un valor concreto (por ejemplo, 2), entonces la re-ejecución de la transacción se indica y se realiza la acción, PASO 1558. Sin embargo, si la condición indica que la re-ejecución es poco probable que sea exitosa (por ejemplo, código de condición 3), CONSULTA 1556, entonces se toma una ruta de retroceso, PASO 1560.

Volviendo a CONSULTA 1552, si la instrucción que sigue a la instrucción de TINICIO recibe un control distinto de una cancelación (por ejemplo, un código de condición cero), entonces la transacción continua para ejecutarse, PASO 1554.

Descrito anteriormente hay un ejemplo de procesamiento transaccional en el que en caso de una cancelación, el tipo de transacción se usa para determinar cómo el procesamiento transaccional se ha de realizar. Además, el tipo de transacción define también el procesamiento de cancelación. Esto permite un entorno transaccional para tener diferentes tipos de transacciones.

Además de lo anterior, se proporciona además un medio eficiente de actualización de objetos no contiguos, múltiples en memoria sin la serialización clásica (de grano gordo), tal como el bloqueo, que proporciona un potencial para una mejora significativa del rendimiento multiprocesador. Esto es, se actualizan múltiples objetos no contiguos sin la aplicación de más órdenes de acceso a almacenamiento de grano grueso que son proporcionadas por las técnicas clásicas, tales como los bloqueos y semáforos. La ejecución especulativa se proporciona sin que se ofrezcan configuraciones de recuperación onerosa, y transacciones restringidas para las actualizaciones simples, de pequeña huella.

La ejecución transaccional se puede usar en una variedad de escenarios, que incluyen, pero no se limitan a, el alineamiento parcial, el procesamiento especulativo, y la elisión del bloqueo. En el alineamiento parcial, la región parcial a ser incluida en la ruta ejecutada se empaqueta en TINICIO/TFIN. TCANCELACIÓN puede ser incluida para hacer el retroceso del estado en una salida lateral. Para la especulación, tal como en Java, las comprobaciones a nulo de los punteros referenciados se pueden retrasar para hacer bucle usando una transacción. Si el puntero es nulo, la transacción puede cancelarse de manera segura usando TCANCELACIÓN, que se incluye dentro de TINICIO/TFIN.

Como para la elisión de bloqueo, se describe un ejemplo de su uso con referencia a las FIG. 16A-16B y el fragmento de código proporcionado más adelante.

La FIG. 16A representa una lista 1600 doblemente enlazada de una pluralidad de elementos 1602a-1602d de cola. Un nuevo elemento 1602e de cola se ha de insertar dentro de la lista doblemente enlazada de elementos 1600 de cola. Cada elemento 1602a-1602e de cola incluye un puntero 1604a-1604e hacia delante y un puntero 1606a-1606e

hacia atrás. Como se muestra en la FIG. 16B, para añadir el elemento 1602e de cola entre los elementos 1602b y 1602c de cola, (1) el puntero 1606e hacia atrás se fija para apuntar el elemento 1602b de cola, (2) el puntero 1604e hacia delante se fija para apuntar al elemento 1602c de cola, (3) el puntero 1606c hacia atrás se fija para apuntar al elemento 1602e de cola, y (4) el puntero 1604b hacia delante se fija para apuntar al elemento 1602e de cola.

5 Un fragmento de código de ejemplo correspondiente a las FIG. 16A-16B es el siguiente:

\* R1 – dirección del nuevo elemento de cola a ser insertado

\* R2 – dirección del nuevo punto de inserción; se inserta un nuevo elemento antes del elemento apuntado por R2.

	NUEVO	USANDO	QEL, R1	
10	ACTUAL	USANDO	QEL, R2	
		LH1	R15, 10	Carga contador de reintentos
	BUCLE	TINICIO	TDB, X'C000'	Inicia la transacción (guarda GR 0-3)
	cancelados	JNZ	CANCELADO	Medios CC distintos de cero
15		LG	R3, CURR.BWD	Apuntar a elemento anterior
	PREV	USAR	QEL, R3	Lo hace direccionable
	prev.	STG	R1, PREV, FWD	Actualizar puntero hacia delante
		STG	R1, CURR, BWD	Actualizar puntero hacia atrás prev.
20		STG	R2, NUEVO, FWD	Actualizar puntero hacia delante
	prev.			
		STG	R3, NUEVO, BWD	Actualizar puntero hacia atrás
		TFIN		Finaliza la transacción
25		...		
	CANCELADO	JO	NO REINTENTO	CC3: Cancelación no reintentable
	veces	JCT	R15, BUCLE	Reintentar transacción unas pocas
30	hacerlo a la manera difícil.	J	NO REINTENTO	No satisfactorio después 10x;

En un ejemplo, si la transacción se usa para la elisión de bloqueo, pero la ruta de retroceso usa un bloqueo, la transacción ha de extraer al menos la palabra de bloqueo para ver que está disponible. El procesador asegura que la transacción se cancela, si otra CPU accede el bloqueo no transaccional.

35 Tal como se usa en la presente memoria, el almacenamiento, el almacenamiento central, el almacenamiento principal, la memoria y la memoria principal se usan de manera intercambiable, a menos que se diga lo contrario, de manera implícita mediante el uso o de manera explícita. Además, mientras en una realización, una transacción que se retrasa de manera efectiva incluye retrasar las confirmaciones de los almacenamientos transaccionales a la memoria principal hasta que se finalice la transacción seleccionada; en otra realización, una transacción que se retrasa de manera efectiva incluye permitir actualizaciones transaccionales a la memoria, pero mantener los valores antiguos y restaurar la memoria a los valores antiguos en una cancelación

40 Como será apreciado por alguien experto en la técnica, uno o más aspectos se pueden realizar como un sistema, método o producto de programa informático. Por consiguiente, uno o más aspectos pueden tomar la forma de una realización completamente de hardware, una realización completamente de software (incluyendo firmware, software residente, micro código, etc.) o una realización que combine aspectos de software y hardware que pueden ser referidos de manera general en la presente memoria 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 código de programa legible por ordenador realizado en éstos.

Se puede utilizar cualquier combinación de uno o más medios legibles por ordenador. 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 limitado a, un sistema, aparato, o dispositivo electrónico, magnético, óptico, electromagnético, de infrarrojos o semiconductor o cualquier combinación adecuada de lo anterior. 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 disco informático portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM), una memoria de sólo lectura programable borrable (EPROM o memoria 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 lo anterior. En el contexto de este documento, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que pueda contener o almacenar un programa para su uso por o en conexión con un sistema, aparato, o dispositivo de ejecución de instrucción.

Referente ahora a la FIG. 17, en un ejemplo, un producto 1700 de programa informático incluye, por ejemplo, uno o más medios 1702 de almacenamiento legibles por ordenador no transitorios para almacenar medios o lógica 1704 de código de programa legibles por ordenador para proporcionar y facilitar uno o más aspectos.

El código de programa realizado en un medio legible por ordenador se puede transmitir usando un medio apropiado, que incluye, pero no se limita al inalámbrico, por cable, de cable de fibra óptica, de RF, etc., o cualquier combinación adecuada de lo anterior.

El código de programa informático para llevar a cabo las operaciones para uno o más aspectos se puede escribir en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos, tal como el Java, el Smalltalk, C++ o similar, y lenguajes de programación procedurales convencionales, tales como el lenguaje de programación "C", el ensamblador o lenguajes de programación similares. El código de programa puede ejecutarse por completo en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o por completo en el ordenador o servidor remoto. En un último escenario, el ordenador remoto se puede conectar 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 se puede hacer a un ordenador externo (por ejemplo, a través de Internet usando un Proveedor de Servicios de Internet).

Uno o más aspectos se describen en la presente memoria con referencia a las ilustraciones de diagramas de flujo y/o diagramas de bloques de los métodos, aparatos (sistemas) y productos de programa informático. Se entenderá que cada bloque de las ilustraciones de los diagramas de flujo y/o de los diagramas de bloques, y las combinaciones de los bloques en las ilustraciones de los diagramas de flujo y/o los diagramas de bloques, pueden ser implementados por instrucciones de programas informáticos. Estas instrucciones de programas informáticos se pueden proporcionar a un procesador de un ordenador de propósito general, un ordenador de propósito específico, u otro aparato de procesamiento de datos programable para producir una máquina, tal como las instrucciones, que ejecuta a través del procesador del ordenador u otro aparato de procesamiento de datos programable, crean los medios para implementar las funciones/acciones específicas en el diagrama de flujo y/o el bloque o los bloques en el diagrama de bloques.

Estas instrucciones de programa informático se pueden almacenar también en un medio legible por ordenador que pueda dirigir a un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para funcionar de una manera concreta, de manera tal que las instrucciones almacenadas en el medio legible por ordenador produzcan un artículo de fabricación que incluye instrucciones que implementan la función/acción especificada en el diagrama de flujo y/o el bloque o bloques del diagrama de bloques.

Las instrucciones del programa informático se pueden cargar también en un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para provocar que una serie de pasos operacionales sean realizados en el ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para producir un proceso implementado por ordenador de manera tal que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionen los procesos para implementar las funciones/acciones especificadas en el diagrama de flujo y/o el bloque o bloques del diagrama de bloques.

El diagrama de flujo y los diagramas de bloques de las figuras ilustran la arquitectura, la funcionalidad, y la operación de las posibles implementaciones de los sistemas, métodos y productos de programa informáticos según diversas realizaciones. A este respecto, cada bloque en el diagrama de flujo o los diagramas de bloques pueden representar un módulo, segmento, o parte de código, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. Se debería observar también que, en algunas implementaciones alternativas, las funciones observadas en el bloque pueden darse fuera del orden observado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ser ejecutados de manera sustancialmente concurrente, o los bloques pueden a veces ser ejecutados en el orden inverso, dependiendo de la funcionalidad

involucrada. Se observa también que cada bloque de los diagramas de bloques y/o la ilustración del diagrama de flujo, y las combinaciones de los bloques en los diagramas de bloques y/o la ilustración del diagrama de flujo, pueden ser implementadas por sistemas basados en hardware de propósito especial que realizan las funciones o acciones especificadas, o las combinaciones de hardware de propósito especial y las instrucciones informáticas.

- 5 Además de lo anterior, se pueden proporcionar, ofertar, desplegar, gestionar, servir, etc. uno o más aspectos por un proveedor de servicio que ofrece gestión de entornos de cliente. Por ejemplo, el proveedor de servicio puede crear, mantener, soportar, etc. código de programa y/o una infraestructura informática que realiza uno o más aspectos para uno o más clientes. En cambio, el proveedor de servicio puede recibir un pago desde el cliente bajo una suscripción y/o un acuerdo de pago, como ejemplos. De manera adicional o alternativa, el proveedor de servicio puede recibir el pago de la venta de contenidos publicitarios a una o más terceras partes.

En una realización, se puede desplegar una aplicación para realizar una o más realizaciones. Como un ejemplo, el despliegue de la aplicación comprende proporcionar unas infraestructuras informáticas operables para realizar una o más realizaciones.

- 15 Como un aspecto adicional, se puede desplegar una infraestructura informática que comprende un código legible por ordenador integrado en un sistema informático, código el cual en combinación con el sistema informático es capaz de realizar una o más realizaciones.

- 20 Aún como un aspecto adicional, se puede proporcionar un proceso para integrar la infraestructura informática que comprende integrar el 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.

- 25 Aunque a continuación se describen diversas realizaciones, estas son sólo ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas se pueden usar incorporadas y usar una o más realizaciones. Además, se pueden usar diferentes instrucciones, formatos de instrucción, campos de instrucción y/o valores de instrucción. Además, se pueden proporcionar/usar tipos de transacciones distintas y/o adicionales. Son posibles muchas variaciones.

- 30 Además, pueden beneficiar y usarse otros tipos de entornos informáticos. Como ejemplo, se puede usar un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar código de programa que incluye al menos dos procesadores acoplados de manera directa o indirecta 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 masivo, y memoria de caché que proporciona almacenamiento temporal de al menos algún código de programa para reducir el número de veces que el código debe ser recuperado desde el almacenamiento masivo durante la ejecución.

- 35 Los dispositivos de Entrada/Salida o I/O (incluyendo, pero no limitados a, teclados, elementos de presentación, dispositivos de apuntamiento, DASD, cintas, CD, DVD, dispositivos de memoria y otros medios de memoria, etc.) se pueden acoplar al sistema bien de manera directa o a través de controladores de I/O intervinientes. Los adaptadores de red se pueden acoplar también al sistema para permitir al sistema de procesamiento de datos que se acople a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes intermedias privadas o públicas. Los módems, los cable-módems, y las tarjetas Ethernet son sólo unos pocos de los tipos disponibles de adaptadores de red.

- 40 Referente a la FIG. 18, se representan los componentes representativos de un sistema 5000 Informático de Servidor para implementar una o más realizaciones. El sistema 5000 informático de servidor comprende una o más CPU 5001 en comunicación con la memoria 5002 informática (por ejemplo, de almacenamiento central), así como interfaces de I/O para almacenar los dispositivos 5011 de medios y las redes 5010 para comunicarse con otros ordenadores o SAN y similares. La CPU 5001 es compatible con una arquitectura que tiene un conjunto de instrucciones de arquitectura y una funcionalidad de arquitectura. La CPU 5001 puede tener traducción 5012 de registro de acceso (ART), que incluye una memoria intermedia 5013 de traducción adelantada (ALB), para seleccionar un espacio de direcciones a ser usado por la traducción 5003 de direcciones dinámica (DAT) para transformar las direcciones de programa (direcciones virtuales) en direcciones reales de memoria. Una DAT normalmente incluye una memoria intermedia 5007 de traducción adelantada (TLB) para almacenar en caché las traducciones para que posteriores accesos al bloque de la memoria 5002 informática no requieran el retraso de la traducción de dirección. Normalmente, se emplea una caché 5009 entre la memoria 5002 informática y el procesador 5001. La caché 5009 puede ser jerárquica teniendo una gran caché disponible para más de una CPU y unas cachés menores, más rápidas (nivel inferior) entre la caché grande y cada CPU. En algunas implementaciones, las cachés de nivel inferior se dividen para proporcionar cachés de nivel inferior separadas para la extracción de las instrucciones y de los accesos de datos. En una realización, para la función TX, el bloque 5100 de diagnóstico de transacción (TDB) y una o más memorias 5101 intermedias se pueden almacenar en una o más cachés 5009 y la memoria 5002. En un ejemplo, en el modo TX, los datos se almacenan de manera inicial en una memoria intermedia TX, y cuando el modo TX finaliza (por ejemplo, el TINICIO más externo), los datos en la memoria intermedia se almacenan (se confirman) en memoria, o si hay una cancelación, los datos en la memoria intermedia se descartan.



En una realización, se extrae una instrucción de la memoria 5002 mediante una unidad 5004 de extracción de instrucciones a través de la caché 5009. La instrucción se decodifica en una unidad 5006 de decodificación de instrucciones y se despacha (con otras instrucciones en algunas realizaciones) a la unidad o unidades 5008 de ejecución de instrucciones. Normalmente se emplean varias unidades 5008 de ejecución, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución de punto flotante y una unidad de ejecución de instrucción de ramificación. Además, en una realización de la función TX, se pueden emplear diversos controles 5110 TX. La instrucción es ejecutada mediante la unidad de ejecución, accediendo a los operandos desde los registros o la memoria especificados por la instrucción según sea necesario. Si se ha de acceder a un operando (cargado o almacenado) desde la memoria 5002, una unidad 5005 de carga/almacenamiento normalmente maneja el acceso bajo el control de la instrucción que se ejecuta. Las instrucciones se pueden ejecutar en circuitos de hardware o en micro código interno (firmware) o mediante una combinación de ambos.

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

Como se observa, un sistema informático incluye información en un almacenamiento local (o principal), así como un registro de direccionamiento, protección y referencia y cambio. Algunos aspectos de la dirección incluyen el formato de las direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones, y la manera en la que un tipo de dirección se traduce a otro tipo de dirección. Algo del almacenamiento principal incluye ubicaciones de almacenamiento asignadas de manera permanente. El almacenamiento principal proporciona al sistema con un almacenamiento de datos de rápido acceso que se pueda direccionar de manera directa. Tanto los datos como los programas se han de cargar en el almacenamiento principal (desde los dispositivos de entrada) antes de que se puedan procesar.

El almacenamiento principal puede incluir uno o más almacenamientos de memoria intermedia, a veces llamados cachés. Una caché normalmente se asocia con una CPU o un procesador de I/O. Los efectos, excepto sobre el rendimiento, de la construcción física y uso de los distintos medios de almacenamiento generalmente no son observables por el programa.

Las cachés separadas se pueden mantener para instrucciones y para los operandos de datos. La información dentro de la caché se mantiene en los bytes contiguos a un límite integral llamado bloque de caché o línea de caché (o línea, para abreviar). Un modelo puede proporcionar una instrucción de EXTRACCIÓN DE ATRIBUTO DE CACHÉ que devuelve el tamaño de una línea de caché en bytes. Un modelo puede proporcionar también las instrucciones de PREEXTRACCIÓN DE DATOS y de PREEXTRACCIÓN DE DATOS RELATIVOS LARGOS que efectúan la pre extracción del almacenamiento en la caché de datos o instrucción o la liberación de datos de la caché.

El almacenamiento es visto como una larga cadena horizontal de bits. Para muchas operaciones, los accesos al almacenamiento proceden en una secuencia de izquierda a derecha. La cadena de bits se subdivide en unidades de ocho bits. Una unidad de ocho bits es llamada un byte, que es el bloque básico de construcción de todos los formatos de información. Cada ubicación de byte en el almacenamiento es identificada por un número entero único no negativo, que es la dirección de la ubicación del byte o simplemente, la dirección del byte. Las ubicaciones de 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 de 24, 31, o 64 bits.

La información se transmite entre el almacenamiento y la CPU o un subsistema de canal en un byte, o un grupo de bytes, cada vez. A menos que se especifique lo contrario, por ejemplo, la z/Arquitectura, un grupo de bytes en el almacenamiento es accedido mediante el byte más a la izquierda del grupo. El número de bytes en el grupo es bien implicado o explícitamente especificado por la operación a ser realizada. Cuando se usa en una operación de la CPU, un grupo de bytes es llamado un campo. Dentro de cada grupo de bytes, en, por ejemplo, la z/Arquitectura, los bits se numeran en una secuencia de izquierda a derecha. En la z/Arquitectura, los bits más a la izquierda son referidos a veces como los bits de "mayor orden" y los bits más a la derecha como los bits de "menor orden". Los números de bit no son direcciones de almacenamiento, sin embargo. Sólo se pueden direccionar bytes. Para operar sobre bits individuales de un byte en el almacenamiento, se accede al byte completo. Los bits en un byte se numeran de 0 a 7, de izquierda a derecha (en, por ejemplo, la z/Arquitectura). Los bits en una dirección se pueden numerar de 8-31 o 40-63 para direcciones de 24 bits, o 1-31 o 33-63 para direcciones de 31 bits; se numeran 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, un registro) que es de 32 bits de ancho, donde los bits 40-63 y 33-63 aplican a las 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 constituyen el formato se numeran de manera consecutiva comenzando desde 0. Para propósitos de detección de errores, y preferiblemente de corrección, se pueden transmitir uno o más bits de comprobación con cada tipo o con un grupo de bytes. Dichos bits de comprobación se generan de manera automática por la máquina y no pueden ser directamente controlados por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de operando de almacenamiento es implicada por el código de operación de una instrucción, se dice que el campo tiene una longitud fija, que puede ser de uno, dos, cuatro, ocho, o dieciséis bytes. Se pueden implicar campos más largos para algunas instrucciones. Cuando la longitud del campo

de un operando de almacenamiento no está implicada pero se establece de manera explícita, se dice que el campo tiene una longitud variable. Los operandos de longitud variable pueden variar en longitud con incrementos de un byte (o en algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando se pone la información en el almacenamiento, sólo se reemplazan los contenidos de aquellas ubicaciones de byte que se incluyen en el campo designado, incluso aunque la anchura de la ruta física para el almacenamiento pueda ser mayor que la longitud del campo que se almacena.

Ciertas unidades de información han de estar en un límite integral en el almacenamiento. Un límite se llama integral para una unidad de información cuando su dirección de almacenamiento es un múltiplo de la longitud de la unidad en bytes. Se dan 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 en un límite de dos bytes y es el bloque de construcción básico de instrucciones. Una palabra es un grupo de cuatro bytes consecutivos en un límite de cuatro bytes. Una doble palabra es un grupo de ocho bytes consecutivos en un límite de ocho bytes. Una palabra cuádruple es un grupo de 16 bytes consecutivos en un límite de 16 bytes. Una palabra óctuple es un grupo de 32 bytes consecutivos en un límite de 32 bytes. Cuando las direcciones de almacenamiento designan medias palabras, palabras, dobles palabras, cuádruples palabras y óctuples palabras, la representación binaria de la dirección contiene uno, dos, tres, cuatro, o cinco bits cero más a la derecha, respectivamente. Las instrucciones han de ser en límites integrales de dos bytes. Los operandos de almacenamiento de la mayoría de las instrucciones no tiene requisitos de alineamiento de límites.

En los dispositivos que implementan cachés separadas para las instrucciones y los operandos de datos, se puede experimentar un retardo significativo si el programa almacena en una línea de caché desde la cual las instrucciones son posteriormente extraídas, independientemente de si el almacenamiento altera las instrucciones que son posteriormente extraídas.

En un ejemplo, las realizaciones se pueden poner en práctica mediante software (a veces referido como código interno con licencia, firmware, micro código, mili código, pico código y similares, cualquiera de los cuales sería consistente con una o más realizaciones). Referente a la FIG. 18, el código de programa de software que realiza uno o más aspectos puede ser accedido por el procesador 5001 del sistema 5000 de servidor desde los dispositivos 5011 de medios de almacenamiento de largo plazo, tales como la unidad de CD-ROM, la unidad de cinta o la unidad de disco duro. El código de programa de software se puede realizar en cualquiera de una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disco, una unidad de disco duro, o un CD-ROM. El código se puede distribuir sobre dichos medios, o se puede distribuir a los usuarios desde la memoria 5002 informática o el almacenamiento de un sistema informático sobre una red 5010 a otros sistemas informáticos para su uso por los usuarios de dichos otros sistemas.

El código de programa de software incluye un sistema operativo que controla la función e interacción de diversos componentes informáticos y uno o más programas de aplicación. El código de programa normalmente está paginado desde el dispositivo 5011 de medio de almacenamiento al almacenamiento 5002 informático de relativamente mayor velocidad donde está disponible para su procesamiento por el procesador 5001. Las técnicas y métodos para realizar el código de programa de software en memoria, en el medio físico, y/o distribuir el código de software a través de las redes son bien conocidos y no se discutirán de manera adicional en la presente memoria. El código de programa, cuando es creado y almacenado en un medio tangible (incluyendo, pero no limitado a módulos de memoria electrónica (RAM), memoria flash, Discos Compactos (CD), DVD, Cintas Magnéticas y similares es a menudo referido como "producto de programa informático". El medio del producto de programa informático normalmente es legible por un circuito de procesamiento preferiblemente en un sistema informático para su ejecución por el circuito de procesamiento.

La FIG. 19 ilustra un sistema de hardware de estación de trabajo o servidor representativo en el que se pueden poner en práctica una o más realizaciones. El sistema 5020 de la FIG. 19 comprende un sistema 5021 informático base representativo, tal como un ordenador personal, una estación de trabajo o un servidor, incluyendo dispositivos periféricos opcionales. El sistema 5021 informático base incluye uno o más procesadores 5026 y un bus empleado para conectar y permitir la comunicación entre el procesador o los procesadores 5026 y los otros componentes del sistema 5021 de acuerdo con las técnicas conocidas. El bus conecta el procesador 5026 a la memoria 5025 y al almacenamiento 5027 a largo plazo que puede incluir una unidad de disco duro (que incluye cualquier medio magnético, CD, DVD y Memoria Flash por ejemplo) o una unidad de cintas por ejemplo. El sistema 5021 puede incluir también un adaptador de interfaz de usuario, que conecta el microprocesador 5026 a través del 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 usuario, tal como una pantalla táctil, un teclado de entrada digitalizada, etc. El bus también conecta un dispositivo 5022 de presentación, tal como una pantalla o monitor LCD, al microprocesador 5026 a través de un adaptador de presentación.

El sistema 5021 puede comunicarse con otros ordenadores o redes de ordenadores por medio de un adaptador de red capaz de comunicarse en 5028 con una red 5029. Adaptadores de red de ejemplo son los canales de comunicación, el token ring, Ethernet o los módems. De manera alternativa, el sistema 5021 puede comunicarse usando una interfaz inalámbrica, tal como una tarjeta CDPD (datos de paquetes digitales móviles). El sistema 5021 se puede asociar con dichos 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. Todas estas configuraciones, así como el apropiado hardware y software de comunicación, son conocidas en la técnica.

La FIG. 20 ilustra la red 5040 de procesamiento de datos en la que se pueden poner en práctica una o más realizaciones. La red 5040 de procesamiento de datos puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red por cable, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo 5041, 5042, 5043, 5044 individuales. De manera adicional, como apreciarán aquellos expertos en la técnica, se pueden incluir una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador servidor.

Aún en referencia a la FIG. 20, las redes pueden incluir también ordenadores o servidores centrales, tales como un ordenador de puerta de enlace (servidor 5046 de cliente) o un servidor de aplicaciones (servidor 5048 remoto que puede acceder a un repositorio de datos y puede ser accedido de manera directa desde una estación de trabajo 5045). Un ordenador 5046 de puerta de enlace sirve como punto de entrada en cada red individual. Una puerta de enlace es necesaria al conectar un protocolo de red a otro. La puerta de enlace 5046 se puede acoplar de manera preferible a otra red (Internet 5047 por ejemplo) por medio de un enlace de comunicación. La puerta de enlace 5046 puede acoplarse también de manera directa a una o más estaciones de trabajo 5041, 5042, 5043, 5044 usando un enlace de comunicaciones. El ordenador de puerta de enlace puede ser implementado utilizando un servidor z de Sistema eServidor de IBM disponible de la Corporación de Máquinas de Negocios Internacionales.

Referente de nuevo a la FIG. 19 y a la FIG. 20, el código 5031 de programación de código que puede realizar uno o más aspectos puede ser accedido por el procesador 5026 del sistema 5020 desde los medios 5027 de almacenamiento a largo plazo, tales como una unidad de CD-ROM o una unidad de disco duro. El código de programación de software puede ser realizado en cualquiera de entre una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disco, una unidad de disco duro, o un CD-ROM. El código puede ser distribuido en dichos medios, o puede ser distribuido a los usuarios 5050, 5051 desde la memoria o el almacenamiento de un sistema informático sobre una red a otros sistemas informáticos para su uso por usuarios de dichos otros sistemas.

De manera alternativa, el código de programación puede ser realizado en la memoria 5025, y accedido por el procesador 5026 usando el bus de procesador. Dicho código de programación incluye un sistema operativo que controla la función e interacción de los diversos componentes informáticos y uno o más programas 5032 de aplicación. El código de programa está normalmente paginado desde el medio 5027 de almacenamiento a una memoria 5025 de alta velocidad donde está disponible para ser procesado por el procesador 5026. Las técnicas y métodos para realizar el código de programación de software en memoria, en medios físicos, y/o distribuir el código de software a través de redes son bien conocidos y no se discutirán más en la presente memoria. El código de programa, cuando se crea y almacena en un medio tangible (que incluye pero no se limita a módulos de memoria electrónica (RAM), memoria flash, Discos Compactos (CD), DVD, Cintas Magnéticas y similares es referido a menudo como "producto de programa informático". El medio de producto de programa informático es normalmente legible por un circuito de procesamiento preferiblemente en un sistema informático para su ejecución por el circuito de procesamiento.

La caché que está más fácilmente disponible al procesador (normalmente es más rápida y más pequeña que otras cachés del procesador) es la caché de nivel más bajo (L1 o nivel uno) y el almacenamiento principal (memoria principal) es la caché de nivel más alto (L3 si existiesen 3 niveles). La caché de nivel más bajo a menudo se divide en la caché de instrucciones (I-Cache) que almacena instrucciones de máquina a ser ejecutadas y una caché de datos (D-Cache) que almacena los operandos de datos.

Referente a la FIG. 21, se representa una realización ejemplar de procesador para el procesador 5026. Normalmente se emplean uno o más niveles de caché 5053 para almacenar bloques de memoria para mejorar el rendimiento del procesador. La caché 5053 es una memoria intermedia de alta velocidad que almacena líneas de caché de datos de memoria que probablemente se han de usar. Las líneas de caché típicas son de 64, 128, 256 bytes de datos de memoria. A menudo se emplean cachés separadas para almacenar en caché instrucciones y para almacenar en caché datos. La coherencia de caché (la sincronización de las copias de líneas en la memoria y en las cachés) es proporcionada a menudo por diversos algoritmos de "espionaje" bien conocidos en la técnica. El almacenamiento 5025 de memoria principal de un sistema procesador es referido a menudo como caché. En un sistema de procesador que tiene 4 niveles de caché 5053, el almacenamiento 5025 principal es referido a veces como la caché de nivel 5 (L5) ya que normalmente es más rápida y sólo almacena una parte del almacenamiento no volátil (DASD, cinta etc.) que está disponible a un sistema informático. El almacenamiento 5025 principal "almacena en caché" páginas de datos paginados dentro y fuera del almacenamiento 5025 principal por el sistema operativo.

Un contador 5061 de programa (contador de instrucciones) mantiene un seguimiento de la dirección de la instrucción actual a ser ejecutada. Un contador de programa en un procesador de z/Arquitectura 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 normalmente en una PSW (palabra de estado de programa) de un ordenador de manera tal que persiste durante el cambio de contexto. Por tanto, un programa en progreso, que tiene un valor de contador de programa, puede ser interrumpido por, por ejemplo, el sistema operativo (cambio de contexto desde el entorno de programa al

entorno de sistema operativo). La PSW de programa mantiene el valor de contador de programa mientras el programa no está activo, y el contador de programa (en la PSW) del sistema operativo se usa mientras el sistema operativo esté en ejecución. Normalmente, el contador de programa es aumentado en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Computación con Conjunto de Instrucciones Reducidas) son normalmente de longitud fija mientras que las instrucciones CISC (Computación con Conjunto de Instrucciones Complejas) son normalmente de longitud variable. Las instrucciones de la z/Arquitectura de IBM son instrucciones CISC que tienen una longitud de 2, 4, o 6 bytes. El contador 5061 de Programa es modificado por bien una operación de cambio de contexto o una operación de ramificación de una instrucción de ramificación, por ejemplo. En una operación de cambio de contexto, el valor de contador de programa actual es guardado en la palabra de estado del programa junto con otra información de estado sobre el programa que se ejecuta (tal como los códigos de condición), y se carga un nuevo valor de contador de programa que apunta a una instrucción de un nuevo módulo de programa a ser ejecutado. Se realiza una operación de ramificación para permitir que el programa tome las decisiones o entre en bucle dentro del programa cargando el resultado de la instrucción de ramificación en el contador 5061 de programa.

Normalmente se emplea una unidad 5055 de extracción de instrucciones para extraer instrucciones en nombre del procesador 5026. La unidad de extracción bien extrae "las siguientes instrucciones secuenciales", las instrucciones objetivo de las instrucciones de ramificación, o las primeras instrucciones de un programa que sigue a un cambio de contexto. Las unidades de extracción de Instrucciones modernas emplean técnicas de pre extracción para pre extraer instrucciones de manera especulativa en base a la probabilidad de que se puedan usar las instrucciones pre extraídas. Por ejemplo, una unidad de extracción puede extraer 16 bytes de instrucción que incluyan la siguiente instrucción secuencial y bytes adicionales a las instrucciones secuenciales siguientes.

Las instrucciones extraídas son ejecutadas después por el procesador 5026. En una realización, la instrucción o instrucciones extraídas se pasan a una unidad 5056 de despacho de la unidad de extracción. La unidad de despacho decodifica la instrucción o instrucciones y envía la información acerca de la instrucción o instrucciones decodificadas a las unidades 5057, 5058, 5060 apropiadas. Una unidad 5057 de ejecución normalmente recibe información sobre las instrucciones aritméticas decodificadas desde la unidad 5055 de extracción de instrucciones y realizará las operaciones aritméticas sobre los operandos según el código de operación de la instrucción. Se proporcionan operandos a la unidad 5057 de ejecución preferiblemente bien desde la memoria 5025, los registros 5059 de arquitectura o desde un campo inmediato de la instrucción que se ejecuta. Los resultados de la ejecución, cuando se almacenan, se almacenan bien en la memoria 5025, los registros 5059 o en otro hardware de máquina (tal como los registros de control, los registros PSW y similares).

Las direcciones virtuales se transforman en direcciones reales usando la traducción 5062 de direcciones dinámicas, y de manera opcional, usando la traducción 5063 del registro de acceso.

Un procesador 5026 normalmente tiene una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Referente a la FIG. 22A, una unidad 5057 de ejecución puede comunicarse con los registros 5059 generales de la arquitectura, una unidad 5056 de decodificación/despacho, y una unidad 5060 de almacenamiento, y otras unidades 5065 de procesador por medio de lógica 5071 de interfaz. Una unidad 5057 de ejecución puede emplear varios circuitos 5067, 5068, 5069 de registro para almacenar información sobre la que la unidad 5066 lógica aritmética (ALU) operará. La ALU realiza operaciones aritméticas tales como la suma, resta, multiplicación y división, así como funciones lógicas tales como la y, la o y la o exclusiva (XOR), la rotación y el desplazamiento. Preferiblemente la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras funciones 5072 de arquitectura que incluyen los códigos de condición y la lógica de soporte de recuperación por ejemplo. Normalmente el resultado de una operación ALU es almacenado en un circuito 5070 de registro de salida que puede enviar el resultado a una variedad de otras funciones de procesamiento. Existen muchas disposiciones de unidades de procesador, la presente descripción está destinada sólo a proporcionar un entendimiento representativo de una realización.

Una instrucción SUMA por ejemplo sería ejecutada en una unidad 5057 de ejecución que tiene una funcionalidad aritmética y lógica mientras que una instrucción en punto flotante por ejemplo sería ejecutada en una unidad de ejecución de punto flotante especializada en funciones de punto flotante. Preferiblemente, una unidad de ejecución opera sobre operandos identificados por una instrucción realizando una operación definida por un código de operación sobre los operandos. Por ejemplo, una instrucción de SUMA puede ser ejecutada por una unidad 5057 de ejecución sobre los 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 dos operandos y almacena el resultado en un tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros de origen. La unidad de ejecución preferiblemente utiliza una Unidad Lógica Aritmética (ALU) 5066 que es capaz de realizar una gran variedad de funciones lógicas tales como un Desplazamiento, Rotación, Y, O y XOR así como una gran variedad de funciones algebraicas que incluyen cualquiera de entre la suma, resta, multiplicación, división. Algunas ALU 5066 se diseñan para operaciones escalares y otras para operaciones en punto flotante. Los datos pueden ser Big Endian (donde el byte menos significativo está en la dirección del byte superior) o Little Endian (donde el byte

menos significativo está en la dirección del byte inferior) dependiendo de la arquitectura. La z/arquitectura de IBM es Big Endian. Los campos con signo pueden ser de signo y magnitud, complemento a 1 o complemento a 2 dependiendo de la arquitectura. Un número complemento a 2 es ventajoso ya que la ALU no necesita diseñar una capacidad de resta ya que bien un valor negativo o un valor positivo en complemento a 2 requiere sólo una suma dentro de la ALU. Los números se describen normalmente de manera abreviada, 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 (Kilobyte), por ejemplo.

Referente a la FIG. 22B, la información de la instrucción de ramificación para ejecutar una instrucción de ramificación se envía normalmente a una unidad 5058 de ramificación que a menudo emplea un algoritmo de predicción de ramificación tal como una tabla 5082 de historial de ramificación para predecir la salida de la ramificación antes de que se completen otras operaciones condicionales. El objetivo de la instrucción de ramificación actual se extraerá y ejecutará de manera especulativa antes de que se completen las operaciones condicionales. Cuando las operaciones condicionales se completan las instrucciones de ramificación ejecutadas de manera especulativa son bien completadas o descartadas en base a las condiciones de la operación condicional y la salida especulada. Una instrucción de ramificación típica puede probar los códigos de condición y ramificar a una dirección objetivo si los códigos de condición alcanzan el requisito de ramificación de la instrucción de ramificación, una dirección objetivo puede ser calculada en base a varios números que incluyen los unos encontrados en los campos de registro o un campo inmediato de la instrucción por ejemplo. La unidad 5058 de ramificación puede emplear una ALU 5074 que tenga una pluralidad de circuitos 5075, 5076, 5077 de registro de entrada y un circuito 5080 de registro de salida. La unidad 5058 de ramificación puede comunicarse en 5081 con los registros 5059 generales, la unidad 5056 de despacho de decodificación u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones puede ser interrumpida por una gran variedad de razones incluyendo un cambio de contexto iniciado por un sistema operativo, una excepción de programa o un error que provoca un cambio de contexto, una señal de interrupción de I/O que provoca un cambio de contexto o una actividad de multihilo de una pluralidad de programas (en un entorno multihilo), por ejemplo. Preferiblemente una acción de cambio de contexto guarda la información de estado sobre un programa actualmente en ejecución y después carga la información de estado sobre otro programa que está siendo invocado. La información de estado puede ser guardada en registros de hardware o en la memoria por ejemplo. La información de estado comprende preferiblemente un valor de contador de programa que apunta a la siguiente instrucción a ejecutar, los códigos de condición, la información de traducción de memoria y el contenido del registro de arquitectura. Una actividad de cambio de contexto puede ser ejercida por circuitos de hardware, programas de aplicación, programas de sistema operativo o código de firmware (micro código, pico código o código interno con licencia (LIC)) solos o en combinación.

Un procesador accede a los operandos según los métodos definidos por las instrucciones. La instrucción puede proporcionar un operando inmediato que usa el valor de una parte de la instrucción, puede proporcionar uno o más campos de registro que apuntan explícitamente a bien los registros de propósito general o bien los registros de propósito específico (registros de punto flotante por ejemplo). 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 campos inmediatos como se ejemplifica por la función de desplazamiento largo de la z/Arquitectura en donde la instrucción define un registro base, un registro de índice y un campo inmediato (campo de desplazamiento) que se añade para proporcionar la dirección del operando en memoria por ejemplo. La ubicación en la presente memoria implica normalmente una ubicación en la memoria principal (almacenamiento principal) a menos que se indique lo contrario.

Referente a la FIG. 22C, un procesador accede al almacenamiento usando 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 5053 de memoria, o puede realizar una operación de almacenamiento obteniendo la dirección del operando objetivo en la memoria 5053 y almacenando los datos obtenidos a partir de un registro 5059 u otra ubicación 5053 de memoria en la ubicación del 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 esté fuera de orden en relación a la secuencia de instrucciones, sin embargo, la unidad 5060 de carga/almacenamiento ha de mantener la apariencia hacia los programas de que las instrucciones se ejecutaron en orden. Una unidad 5060 de carga/almacenamiento puede comunicarse en 5084 con los registros 5059 generales, la unidad 5056 de decodificación/despacho, la interfaz 5053 de caché/memoria u otros elementos 5083 y comprende diversos circuitos 5086, 5087, 5088 y 5089 de registro, las ALU 5085 y la lógica de control 5090 para calcular las direcciones de almacenamiento y proporcionar una secuencia de canales para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de orden pero la unidad de carga/almacenamiento proporciona la funcionalidad para hacer que las operaciones fuera de orden aparezcan al programa como habiendo sido realizadas en orden, tal como es bien conocido en la técnica.

Preferiblemente las direcciones que un programa de aplicación "ve" son referidas a menudo como direcciones virtuales. Las direcciones virtuales son referidas a veces como "direcciones lógicas" y "direcciones efectivas". Estas

direcciones virtuales son virtuales ya que son redirigidas a una ubicación de memoria física mediante una de entre una variedad de tecnologías de traducción de direcciones dinámica (DAT) que incluyen, pero no se limitan a, simplemente prefijar una dirección virtual con un valor de desplazamiento, traducir la dirección virtual a través de una o más tablas de traducción, comprendiendo preferiblemente las tablas de traducción al menos una tabla de segmentos y una tabla de páginas solas o en combinación, preferiblemente, teniendo la tabla de segmentos una entrada que apunta a la tabla de páginas. En la z/Arquitectura, se proporciona una jerarquía de traducción que incluye una primera tabla de región, una segunda tabla de región, una tercera tabla de región, una tabla de segmentos y una tabla de páginas opcional. El rendimiento de la traducción de direcciones es a menudo mejorado utilizando una memoria intermedia de traducción adelantada (TLB) que comprende entradas que hacen corresponder una dirección virtual a una ubicación de memoria física asociada. Las entradas se crean cuando la DAT traduce una dirección virtual usando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar la entrada de la TLB rápida en lugar de los accesos a la tabla de traducción secuencial lenta. El contenido de la TLB puede ser gestionado mediante una variedad de algoritmos de reemplazo que incluyen el LRU /Menos Recientemente Usado).

En el caso en que el procesador sea un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener recursos compartidos, tales como la I/O, las cachés, las TLB y la memoria, entrelazados para su coherencia. Normalmente, las tecnologías de "espionaje" se utilizarán en el mantenimiento de la coherencia de la caché. En un entorno de espionaje, cada línea de caché se puede marcar como si estuviera en un estado compartido, un estado exclusivo, un estado cambiado, un estado inválido y similares para facilitar la compartición.

Las unidades 5054 de I/O (FIG. 21) proporcionan al procesador los medios para unirse a los dispositivos periféricos incluyendo, cintas, discos, impresoras, pantallas, y redes por ejemplo. Las unidades de I/O se presentan a menudo al programa informático mediante los controladores de software. En sistemas centrales, tales como el Sistema z de IBM®, los adaptadores de canal y los adaptadores de sistema abierto son unidades de I/O del sistema central que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos. Como ejemplo un entorno puede incluir un emulador (por ejemplo, un software u otros mecanismos de emulación), en los que se emula una arquitectura concreta (incluyendo, por ejemplo, la ejecución de instrucciones, las funciones de arquitectura, tales como la traducción de direcciones, y los registros de arquitectura) o un subconjunto de la misma (por ejemplo, en un sistema informático nativo que tiene un procesador y una memoria). En dicho entorno, una o más funciones de emulación del emulador pueden implementar una o más realizaciones, incluso aunque el ordenador que ejecuta el emulador pueda tener una arquitectura diferente de las capacidades que se emulan. Como un ejemplo, en el modo de emulación, la instrucción específica que está siendo emulada se decodifica, y se construye una función de emulación apropiada para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador servidor incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de extracción de instrucciones para extraer instrucciones de la memoria y para de manera opcional, proporcionar almacenamiento intermedio local para la instrucción extraída; una unidad de decodificación de instrucciones para recibir las instrucciones extraídas y para determinar el tipo de instrucciones que se han extraído; y una unidad de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir datos de carga dentro de un registro desde la memoria; el almacenamiento de datos de vuelta a la memoria desde un registro; o la realización de algún tipo de operación lógica o aritmética, como es determinado por la unidad de decodificación. En un ejemplo, cada unidad se implementa en software. Por ejemplo, las operaciones que son realizadas por las unidades son implementadas como una o más subrutinas dentro del software emulador.

Más concretamente, en un sistema principal, las instrucciones máquina de la arquitectura son usadas por los programadores, normalmente programadores de "C" hoy en día, a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento se pueden ejecutar de manera nativa en una Servidor IBM® de z/Arquitectura, o de manera alternativa en máquinas que ejecutan otras arquitecturas. Se pueden emular en los servidores centrales de IBM® existentes y futuros y en otras máquinas de IBM® (por ejemplo servidores de Sistemas de Alimentación y Servidores x Sistema). Se pueden ejecutar en máquinas que ejecutan Linux en una amplia variedad de máquinas que usan hardware fabricado por IBM®, Intel®, AMD, y otras. Junto a la ejecución en ese hardware bajo una z/arquitectura, se puede usar Linux así como máquinas que usan emulación por Hércules, UMX, o FSI (Software fundamental), donde generalmente la ejecución se realiza en un modo de emulación. En el modo de emulación, el software de emulación es ejecutado mediante un procesador nativo para emular la arquitectura de un procesador emulado.

El procesador emulado normalmente ejecuta software de emulación que comprende bien firmware o un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de la extracción y ejecución de instrucciones de la arquitectura del procesador emulado. El software de emulación mantiene un contador de programa emulado para mantener un seguimiento de los límites de instrucción. El software de emulación puede extraer una o más instrucciones máquina emuladas en el mismo momento y convertir la una o más instrucciones máquina emuladas a un grupo correspondiente de instrucciones máquina nativas para su

ejecución por el procesador nativo. Estas instrucciones convertidas se pueden almacenar en caché de manera tal que se pueda conseguir una conversión más rápida. A pesar de esto, el software de emulación ha de mantener las reglas de arquitectura de la arquitectura del procesador emulado para asegurar que los sistemas operativos y las aplicaciones escritas para el procesador emulado operen de manera correcta. Además, el software de emulación ha de proporcionar los recursos identificados por la arquitectura del procesador emulada que incluyen, pero no se limitan a, los registros de control, los registros de propósito general, los registros de punto flotante, la función de traducción de direcciones dinámicas que incluye las tablas de segmentos y las tablas de páginas por ejemplo, los mecanismos de interrupción, los mecanismos de cambio de contexto, los relojes con la Hora del Día (TOD) y las interfaces de arquitectura a los subsistemas de I/O de manera tal que un sistema operativo o un programa de aplicación diseñado para ejecutarse en el procesador emulado, se pueda ejecutar en el procesador nativo que tiene el software emulado.

Una instrucción específica que se emula se decodifica, y se llama a una subrutina para realizar la función de la instrucción individual. Se implementa una función de software de emulación que emula una función de un procesador emulado, por ejemplo, en una subrutina o controlador en "C", o algún otro método para proporcionar un controlador para el hardware específico según esté dentro de las capacidades de aquellos expertos en la técnica después de la comprensión de la descripción de la realización preferida. Existen diversas patentes de emulación de software y hardware que incluyen, pero no se limitan a, los Documentos de Patente de los EE.UU. N.º. 5.551.013, titulados "Multiprocesador para Emulación de Hardware", por Beausoleil et al; y los Documentos de Patente de los EE.UU. N.º. 6.009.261, titulados "Pre procesamiento de Rutinas Objetivas Almacenadas para Emular Instrucciones Incompatibles en un Procesador Objetivo", por Scalzi et al; y los Documentos de Patente de los EE.UU. N.º. 5.574.873, titulados "Decodificación de Instrucción de Invitado para Acceder Directamente a las Rutinas de Emulación que Emulan las Instrucciones de Invitado", por Davidian et al; y los Documentos de Patente de los EE.UU. N.º. 6.308.255, titulados "Bus y Chipset de Multiprocesamiento Simétrico Usado para el Soporte de Coprocesadores que Permite que Código No Nativo se Ejecute en un Sistema", por Gorishek et al; y los Documentos de Patente de los EE.UU. N.º. 6.463.582, titulados "Traductor de Código de Objetos de Optimización Dinámica para Emulación de Arquitectura y Método de Traducción de Código de Objetos de Optimización Dinámica", por Lethin et al; y los Documentos de Patente de los EE.UU. N.º. 5.790.825, titulados "Método para Emular Instrucciones de Invitado en un Ordenador Servidor A través de una Recopilación Dinámica de Instrucciones de Servidor", por Eric Traut; y muchos otros, ilustran una variedad de maneras conocidas para conseguir la emulación de una arquitectura de formato de instrucción para una máquina diferente para una máquina objetivo disponible a todos aquellos expertos en la técnica.

En la FIG. 23, se proporciona un ejemplo de un sistema 5092 informático de servidor emulado que emula un sistema 5000' informático de servidor de una arquitectura de servidor. En el sistema 5092 informático de servidor emulado, el procesador 5091 (CPU) de servidor es un procesador de servidor emulado (o un procesador de servidor virtual) y comprende un procesador 5093 de emulación que tiene una arquitectura del conjunto de instrucciones nativas diferente a la del procesador 5091 del ordenador 5000' informático servidor. El sistema 5092 informático servidor emulado tiene una memoria 5094 accesible al procesador 5093 de emulación. En la realización de ejemplo, la memoria 5094 se parte en una memoria 5096 informática de servidor y una parte 5097 de rutinas de emulación. La memoria 5096 informática de servidor está disponible a los programas del ordenador 5092 servidor emulado según la arquitectura informática de servidor. El procesador 5093 de emulación ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta a la del procesador 5091 emulado, las instrucciones nativas obtenidas a partir de la memoria 5097 de rutinas de emulación, y puede acceder a una instrucción de servidor para su ejecución desde un programa en una memoria 5096 informática de servidor empleando una o más instrucciones obtenidas en una rutina de secuencia & acceso/decodificación que puede decodificar la instrucción o las instrucciones de servidor accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de servidor accedida. Otras funciones que se definen para la arquitectura del sistema 5000' informático de servidor pueden ser emuladas por rutinas de funciones de la arquitectura, incluyendo dichas funciones registros de propósito general, registros de control, soporte de traducción de direcciones dinámicas y del subsistema de I/O y caché de procesador, por ejemplo. Las rutinas de emulación pueden tomar ventaja de las funciones disponibles en el procesador 5093 de emulación (así como de registros generales y la traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. El hardware especial y los motores de descarga se pueden proporcionar también para ayudar al procesador 5093 a emular la función del ordenador 5000' de servidor.

La terminología usada en la presente memoria tiene el propósito de describir realizaciones concretas sólo y no está destinada a ser limitante. Tal como se usa en la presente memoria, las formas singulares "un", "una" y "el" están destinadas a incluir las formas plurales también, a menos que se indique lo contrario. Se entenderá además que los términos "comprende" y/o "comprendiendo", cuando se usan en esta especificación, especifican la presencia de características establecidas, números enteros, pasos, operaciones, elementos, y/o componentes, pero no imposibilita la presencia o adición de una o más características adicionales, números enteros, pasos, operaciones, elementos, componentes y/o grupos de los mismos.

Las estructuras, materiales, acciones correspondientes de todos los medios o elementos de paso más función en las reivindicaciones de más adelante, si los hubiera, están destinadas a incluir cualquier estructura, material, o acción

5 para realizar la función en combinación con otros elementos reivindicados como se reivindica de manera específica. La descripción de una o más realizaciones ha sido presentada con propósitos de ilustración y descripción, pero no está destinada a ser exhaustiva o limitada a la forma descrita. Muchas modificaciones y variaciones serán evidentes para aquellos de habilidad ordinaria en la técnica. La realización fue elegida y descrita para ser la mejor explicación de diversos aspectos y de la aplicación práctica, y para permitir a otros de habilidad ordinaria en la técnica entender como las diversas realizaciones con las diversas modificaciones son adecuadas para el uso concreto contemplado. El alcance de la invención se define en las reivindicaciones adjuntas.



**REIVINDICACIONES**

1. Un método de controlar la ejecución de una transacción en un entorno informático, comprendiendo el método los pasos de:

5 Iniciar, mediante un procesador, la ejecución de una transacción en un modo de transacción a través de una instrucción de inicio de transacción, la instrucción de inicio de transacción para iniciar cualquiera de entre un primer tipo de transacción o un segundo tipo de transacción, comprendiendo la transacción una pluralidad de instrucciones a ser ejecutadas, e indicando de manera explícita la instrucción de inicio de transacción mediante un campo en la instrucción de inicio de transacción que se ha iniciado el primer tipo de transacción o el segundo tipo de transacción;

10 en base a que la transacción sea el segundo tipo de transacción, la ejecución fija un código de condición a un primer valor,

en base a estar en el modo de transacción, retrasar de manera efectiva las confirmaciones de almacenamiento transaccionales a la memoria principal hasta la finalización de la transacción seleccionada;

15 en base a una cancelación de transacción, determinar el tipo de transacción iniciada, siendo el tipo de transacción el primer tipo de transacción o el segundo tipo de transacción, y estando la determinación basada en el campo de la instrucción de inicio de transacción;

en base a una cancelación de transacción y a que la transacción sea del primer tipo de transacción, reanudar la ejecución en la instrucción de inicio de transacción; y

20 en base a una cancelación de transacción y a que la transacción sea del segundo tipo de transacción, fijar el código de condición a un segundo valor, en donde el segundo valor indica la siguiente instrucción secuencial · que sigue a la instrucción de inicio de transacción · que está recibiendo el control desde la cancelación, y reanudar la ejecución en la siguiente instrucción secuencial que sigue directamente a la instrucción de inicio de transacción en el orden de programa;

25 en donde el primer tipo de transacción es una transacción restringida que tiene una pluralidad de restricciones asociadas y se define para completarse de manera exitosa si se siguen la pluralidad de restricciones, y la instrucción de inicio de transacción es una instrucción restringida de inicio de transacción, y en donde la pluralidad de restricciones asociadas con la transacción restringida incluyen: sólo un subconjunto de instrucciones está disponible para su inclusión en la transacción restringida; la transacción restringida se limita a un número predeterminado de instrucciones a ser ejecutadas en la transacción restringida, siendo el número predeterminado de instrucciones no más de 32 instrucciones, no incluyendo instrucciones de inicio de transacción y de fin de transacción; las instrucciones en la transacción restringida han de estar dentro de los 256 bytes contiguos de almacenamiento; los operandos de almacenamiento del acceso de transacción restringida ser no más de 4 palabras óctuples; y la transacción restringida se limita a un único nivel de anidación, y en donde el segundo tipo de transacción es una transacción no restringida, la transacción no restringida no limitada por la pluralidad de restricciones asociadas con la transacción restringida.

35 2. El método de la reivindicación 1, en donde la transacción es el segundo tipo de transacción, siendo el segundo tipo una transacción no restringida, y en donde la ejecución de la reanudación comprende:

comprobar mediante la siguiente instrucción secuencial que sigue a la instrucción de inicio de transacción un código de condición de la transacción para determinar si la instrucción está recibiendo el control desde una cancelación; y

40 en base a la instrucción que recibe el control desde la cancelación, realizar el procesamiento para manejar la cancelación.

3. El método de la reivindicación 2, en donde la realización del procesamiento comprende tomar acciones en base al código de condición, en donde en base al código de condición de un valor, la acción incluye volver a ejecutar la transacción, y en base al código de condición de otro valor, la acción incluye la ramificación a una ruta de retroceso.

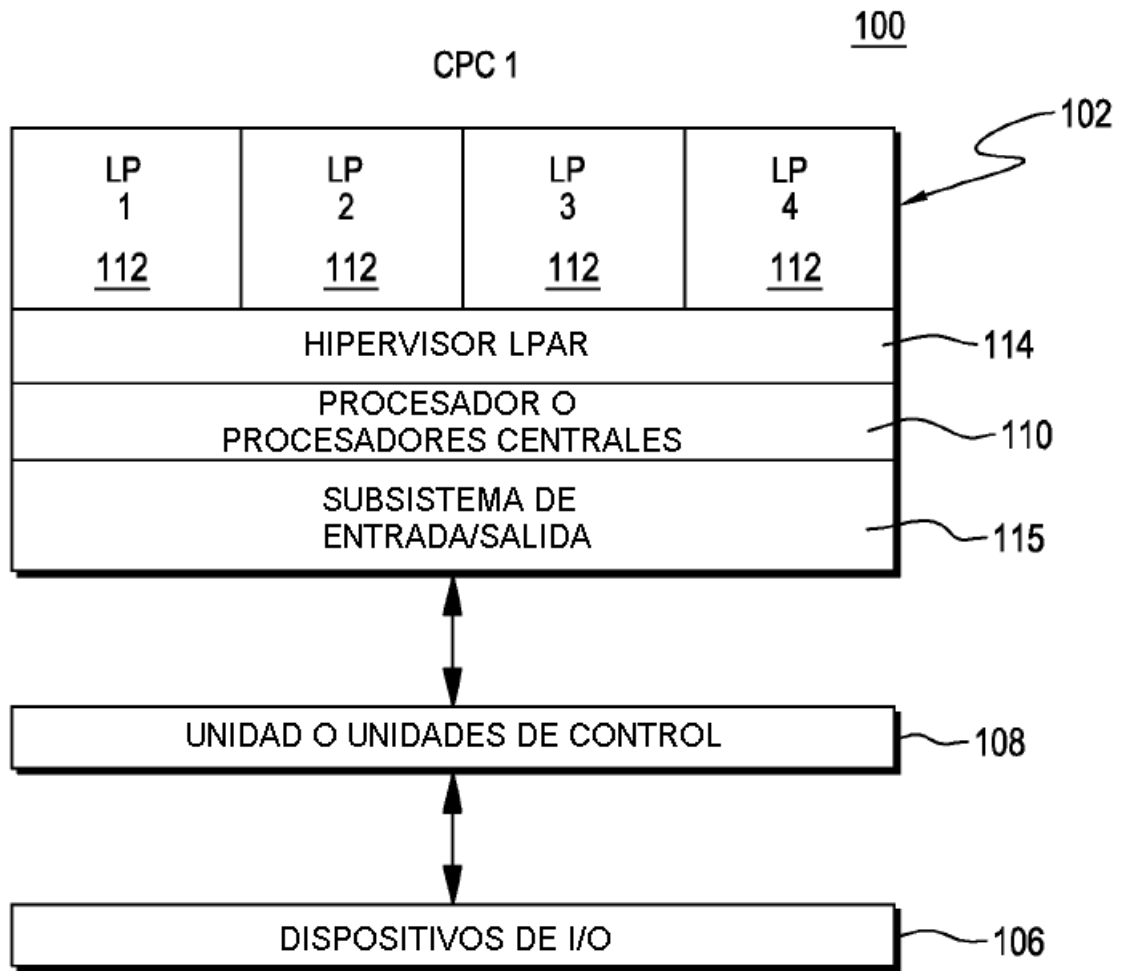
4. El método de la reivindicación 1, en donde el método comprende además:

45 ejecutar la instrucción de inicio de transacción, la ejecución fija el código de condición a un primer valor; y

realizar una cancelación de la transacción, la realización fija el código de condición a un segundo valor, en donde el segundo valor indica que la siguiente instrucción secuencial que sigue a la instrucción de inicio de transacción está recibiendo el control desde la cancelación.

50 5. El método de la reivindicación 1, en donde la transacción es el segundo tipo de transacción, siendo el segundo tipo de transacción una transacción no restringida, y en donde el método comprende además en base a la cancelación de la transacción, almacenar la información de diagnóstico en un bloque de diagnóstico de transacciones especificado por la instrucción de inicio de transacción.

6. El método de la reivindicación 1, en donde el método además comprende en base a la cancelación, restaurar uno o más registros designados en la instrucción de inicio de transacción y descartar los almacenamientos transaccionales.
7. El método de la reivindicación 1, en donde el método además comprende:
- 5 determinar que la transacción se ha cancelado;
- en base a determinar que la transacción se ha cancelado, determinar si la cancelación fue debida a una intercepción o una interrupción; y
- en base a que la cancelación sea debida a algo distinto a una intercepción o una interrupción, reanudar la ejecución en la instrucción de inicio de transacción o reanudar la ejecución en la siguiente instrucción secuencial dependiendo de que la transacción sea del primer tipo o del segundo tipo, respectivamente.
- 10
8. El método de la reivindicación 1, en donde la transacción es parte de un nido de transacciones, y en donde el método comprende además la ejecución por parte del procesador de una instrucción de extracción de la profundidad de anidación para obtener una indicación de la profundidad de anidación del nido de transacciones.
9. El método de la reivindicación 8, en donde la transacción es la transacción más externa del nido de transacciones.
- 15
10. El método de la reivindicación 1, en donde la ejecución de la reanudación en la instrucción de inicio de transacción en base al primer tipo de cancelación de transacción comprende además reconducir automáticamente la transacción restringida, en donde en ausencia de violaciones de restricción, se asegura la posible finalización de la transacción restringida.
- 20
11. El método de la reivindicación 10, en donde para asegurar la posible finalización, el procesador realiza ciertas acciones, incluyendo al menos una de entre: inhibir la ejecución fuera de orden; inhibir que otros procesadores accedan a las ubicaciones de almacenamiento conflictivas; o incluir retardos aleatorios en el procesamiento de la cancelación.
12. Un sistema que comprende los medios adaptados para llevar a cabo todos los pasos del método según cualquier reivindicación anterior del método.
- 25
13. Un programa informático que comprende instrucciones para llevar a cabo todos los pasos del método según cualquier reivindicación anterior del método, cuando dicho programa informático se ejecuta en un sistema informático.



**FIG. 1**

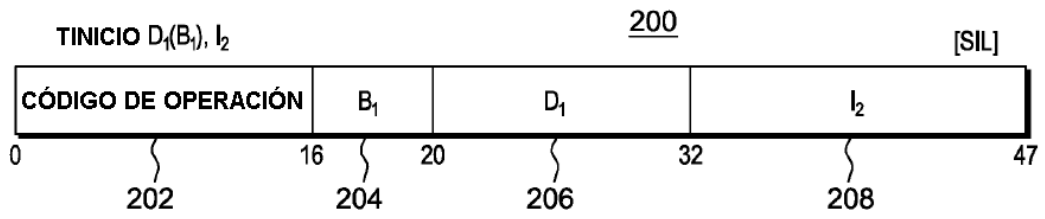


FIG. 2A

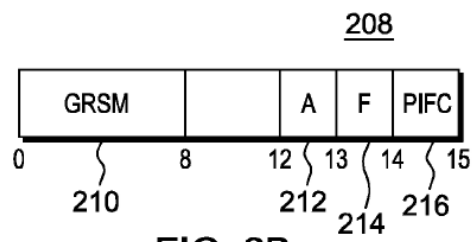


FIG. 2B

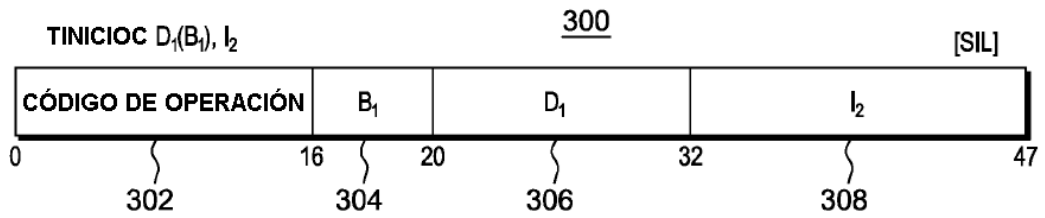


FIG. 3A

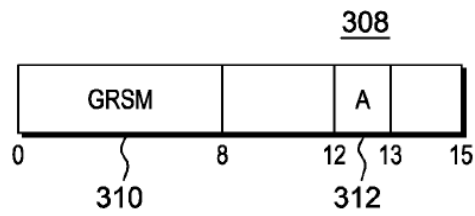


FIG. 3B

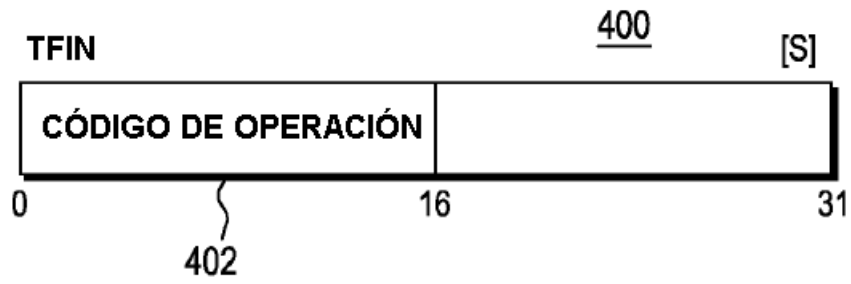


FIG. 4

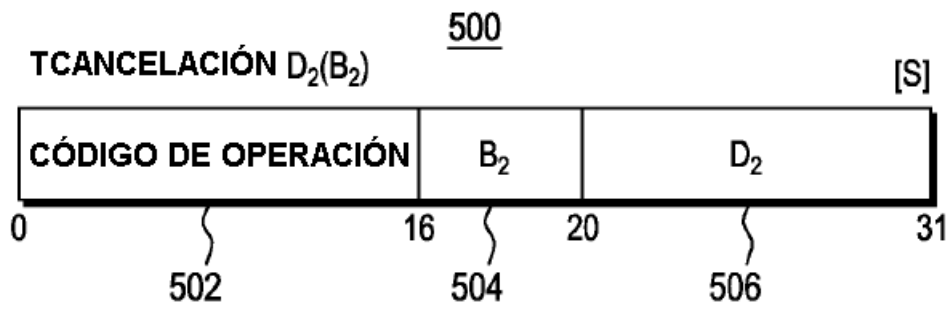


FIG. 5

TRANSACCIONES ANIDADAS

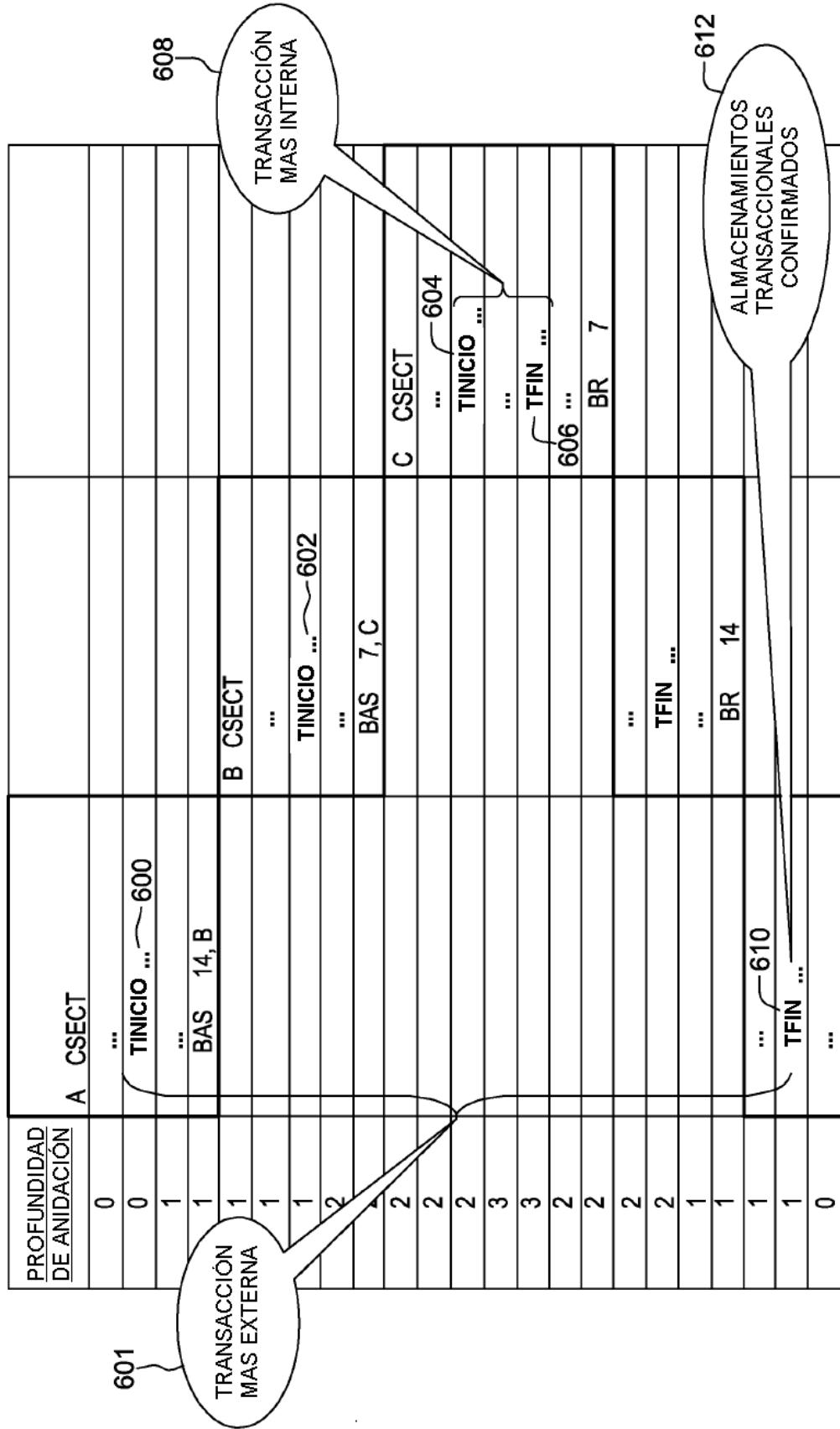


FIG. 6

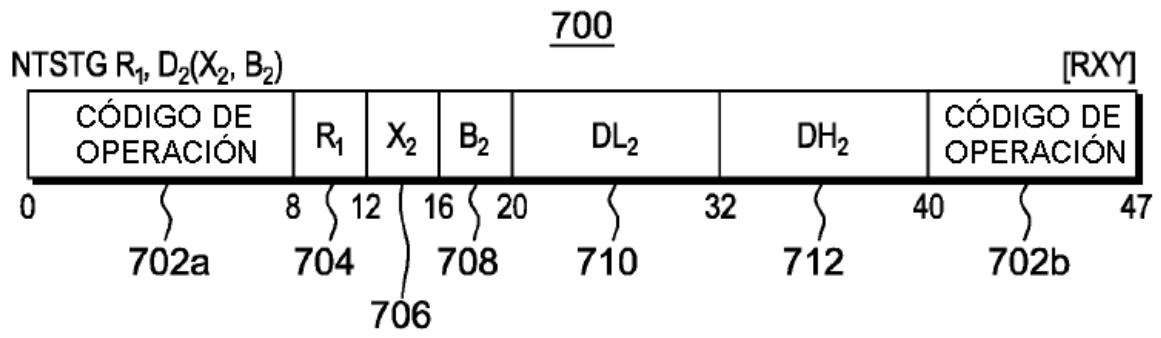


FIG. 7

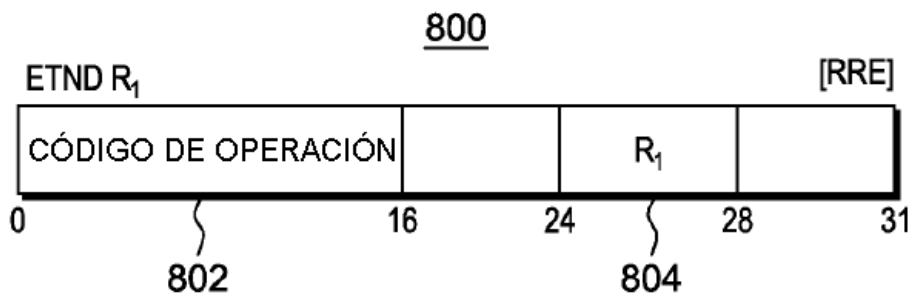


FIG. 8

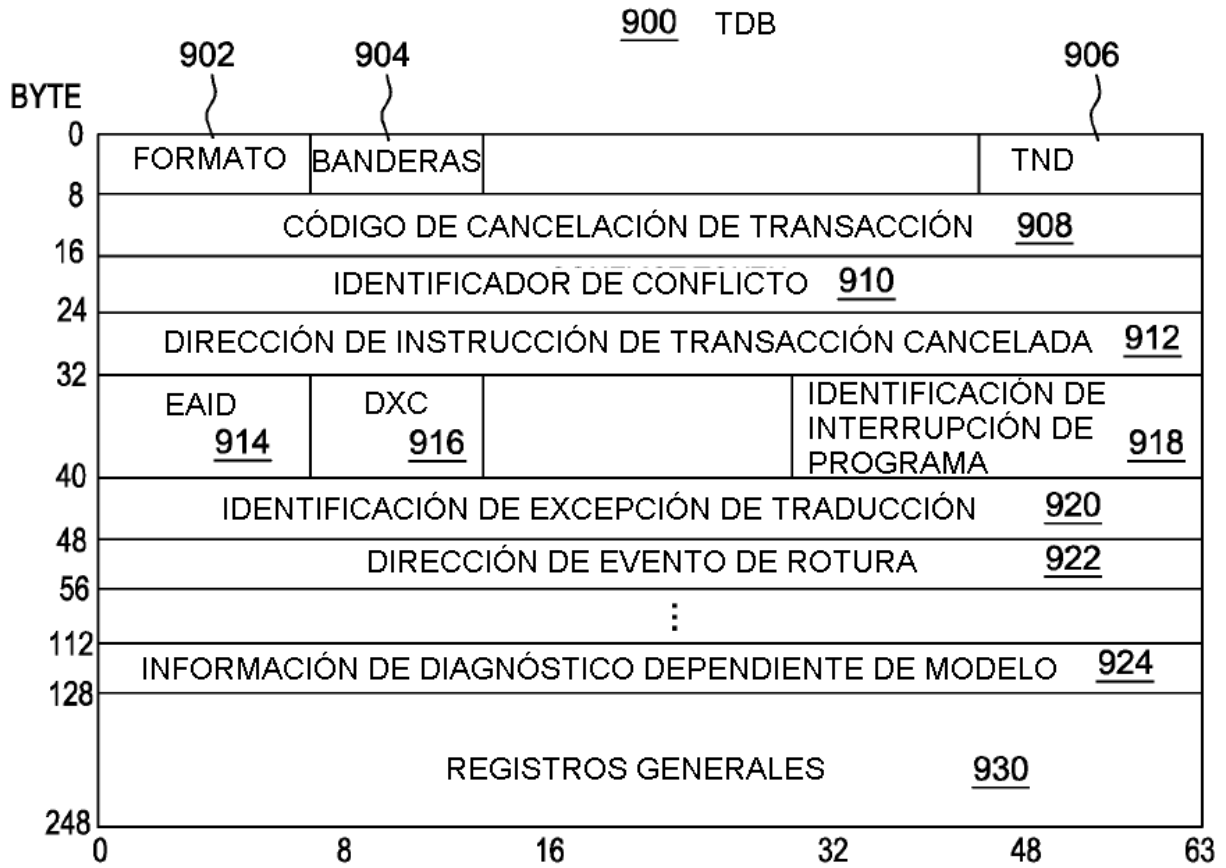


FIG. 9



CÓDIGO	RAZÓN PARA LA CANCELACIÓN	CC FIJADO
2	INTERRUPCIÓN EXTERNA	2
4	INTERRUPCIÓN DE PROGRAMA (SIN FILTRAR)	2 o 3 +
5	INTERRUPCIÓN DE COMPROBACIÓN DE MÁQUINA	2
6	INTERRUPCIÓN DE I/O	2
7	DESBORDAMIENTO DE EXTRACCIÓN	2 o 3
8	DESBORDAMIENTO DE ALMACENAMIENTO	2 o 3
9	CONFLICTO DE EXTRACCIÓN	2
10	CONFLICTO DE ALMACENAMIENTO	2
11	INSTRUCCIÓN RESTRINGIDA	3
12	CONDICIÓN DE EXCEPCIÓN DE PROGRAMA (FILTRADA)	3
13	PROFUNDIDAD DE ANIDACIÓN EXCEDIDA	3
14	EXTRACCIÓN DE CACHÉ RELACIONADA	2 o 3
15	ALMACENAMIENTO EN CACHÉ RELACIONADO	2 o 3
16	OTRAS CACHÉS	2 o 3
255	CONDICIÓN MISCELANEA	2 o 3
>255	INSTRUCCIÓN DE TCANCELACIÓN	2 o 3
‡	NO SE PUEDE DETERMINAR; TDB NO ALMACENADO	1
<p><b>EXPLICACIÓN:</b></p> <p>+ EL CÓDIGO DE CONDICIÓN SE BASA EN EL CÓDIGO DE INTERRUPCIÓN</p> <p>‡ ESTA SITUACIÓN OCURRE CUANDO SE CANCELA UNA TRANSACCIÓN, PERO EL TDB HA RESULTADO INACCESIBLE DESPUÉS DE LA EJECUCIÓN CON ÉXITO DE LA INSTRUCCIÓN TINICIO MÁS EXTERNA. NO SE ALMACENATBD DE TINICIO ESPECIFICADO, Y EL CÓDIGO DE CONDICIÓN SE FIJA A 1.</p>		

FIG. 10

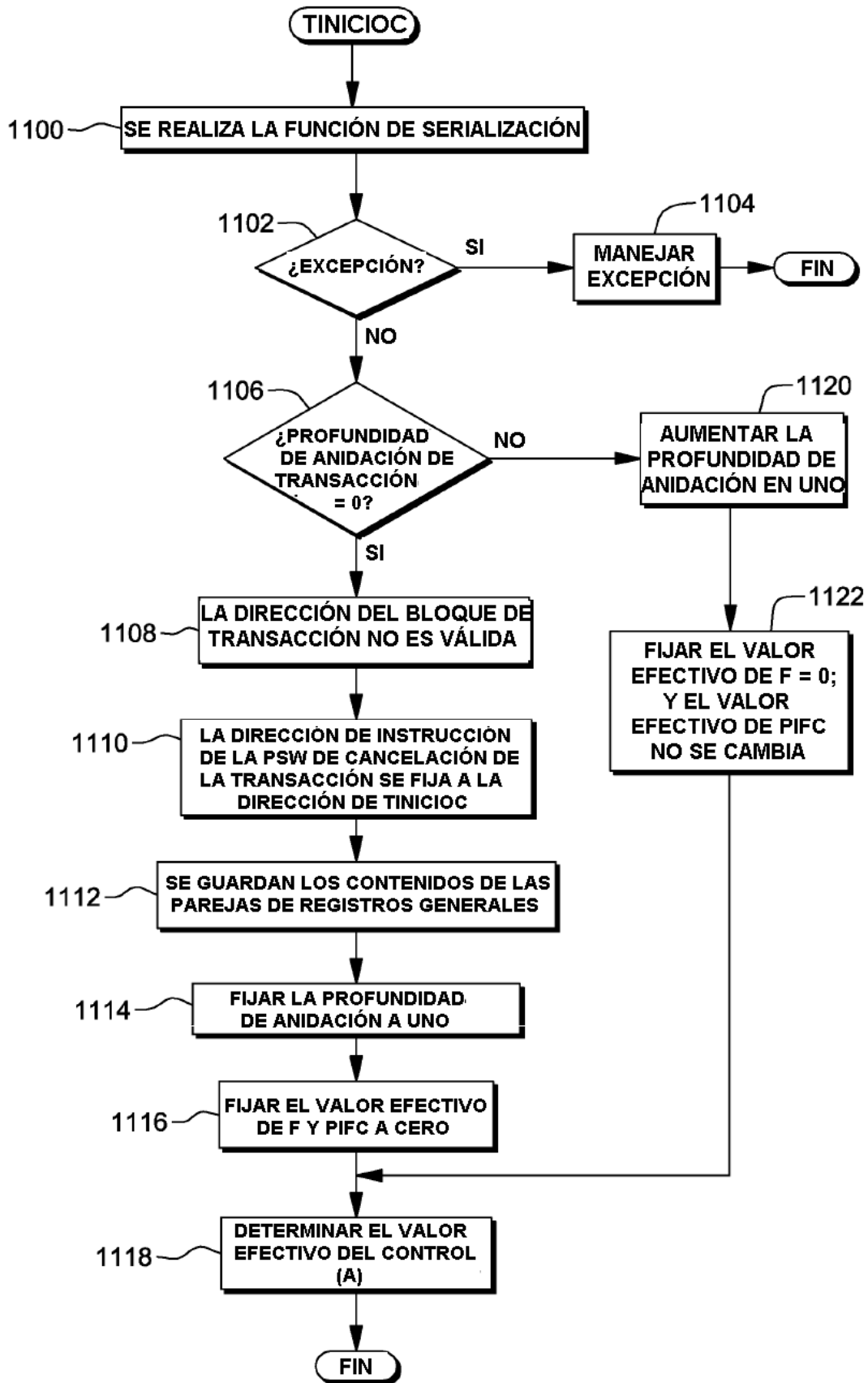


FIG. 11

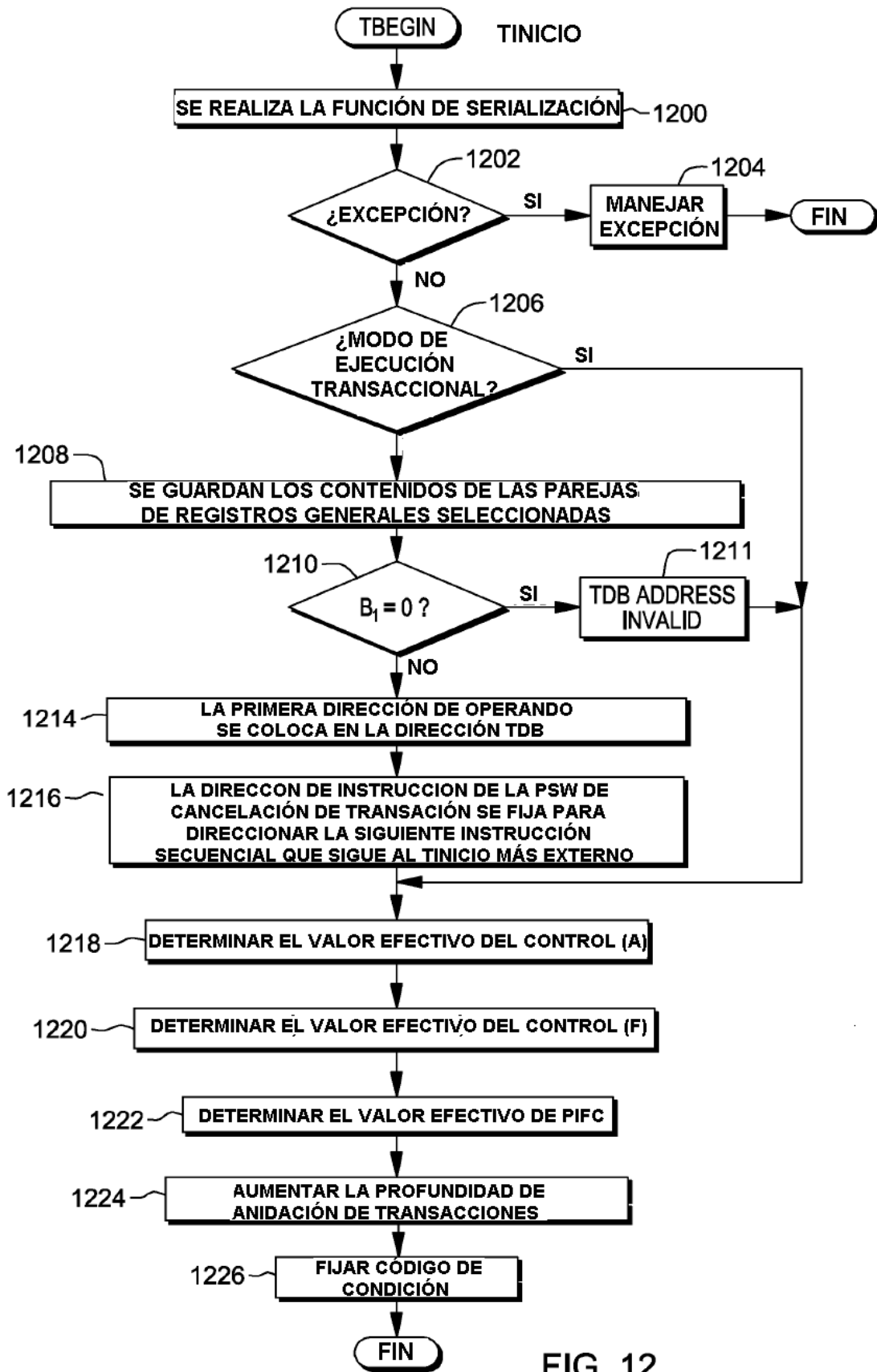


FIG. 12

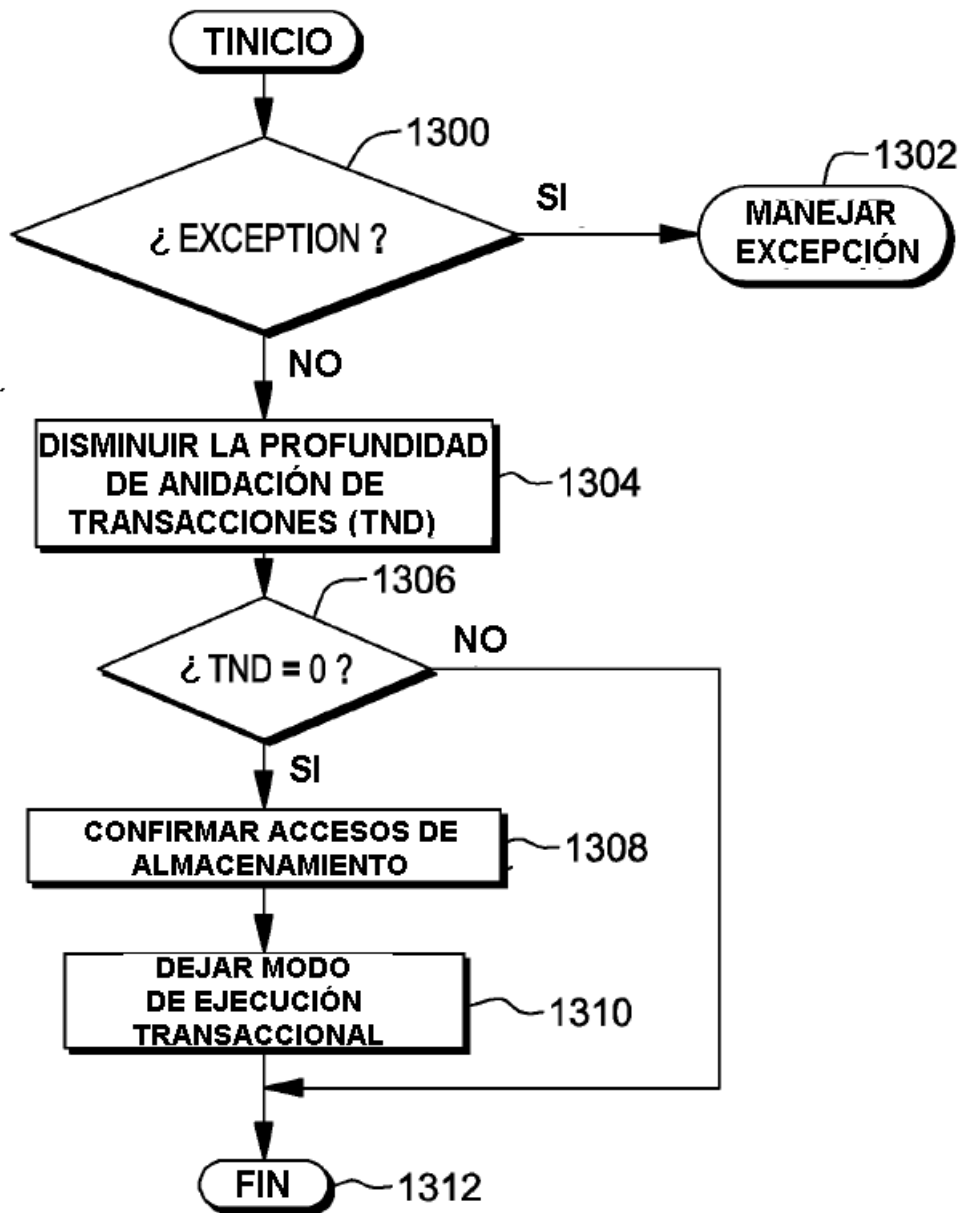


FIG. 13

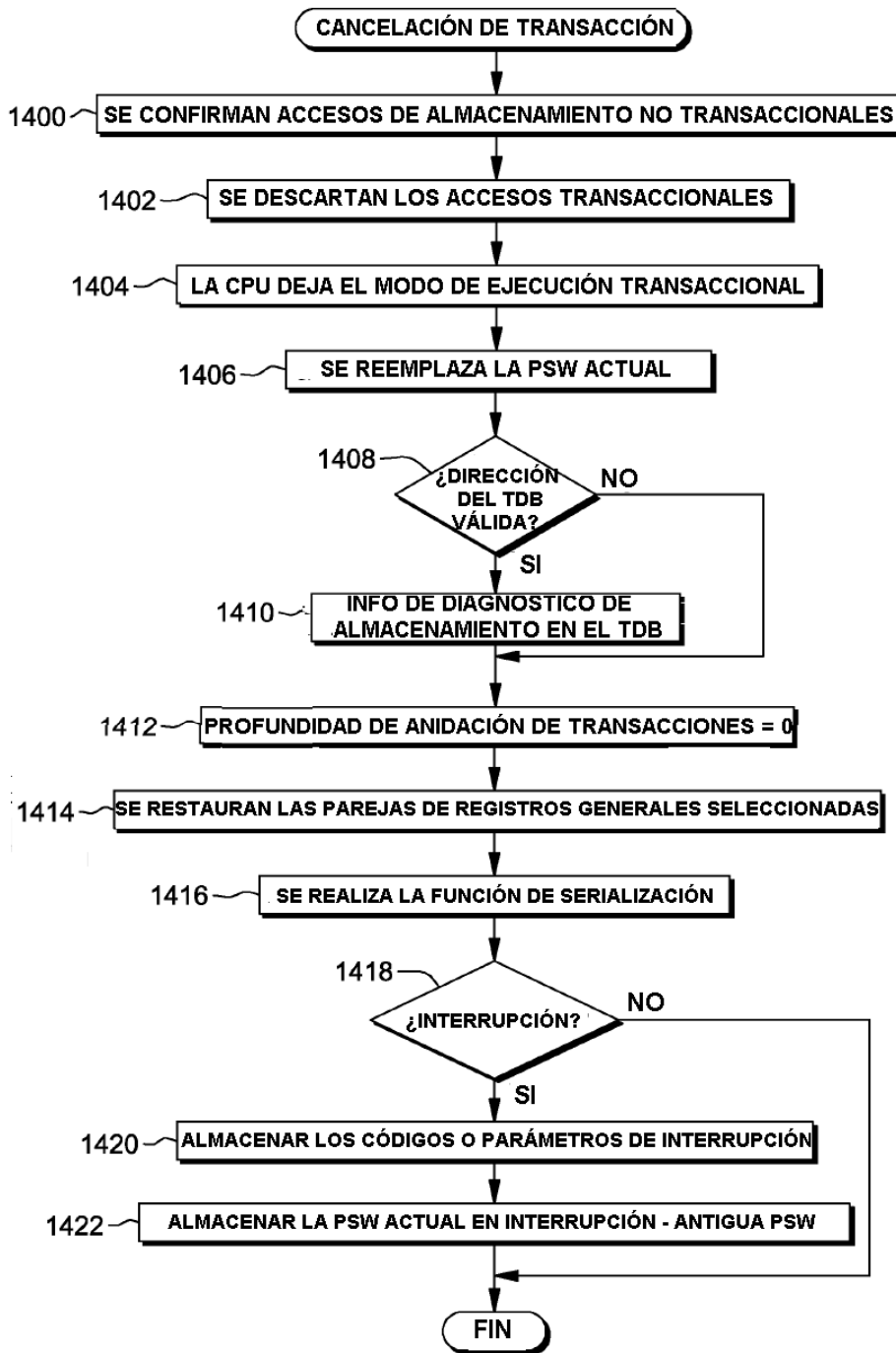


FIG. 14

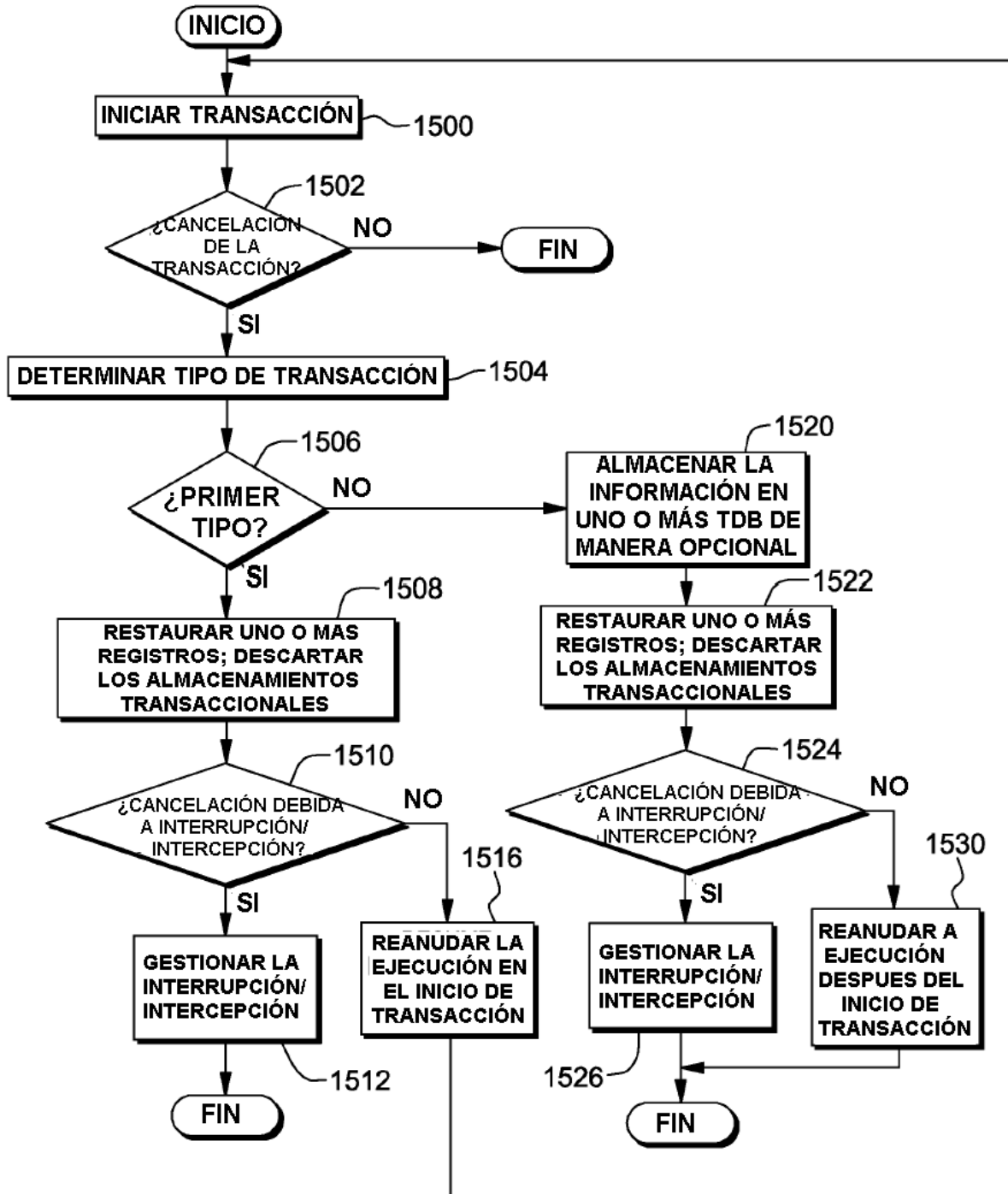


FIG. 15A

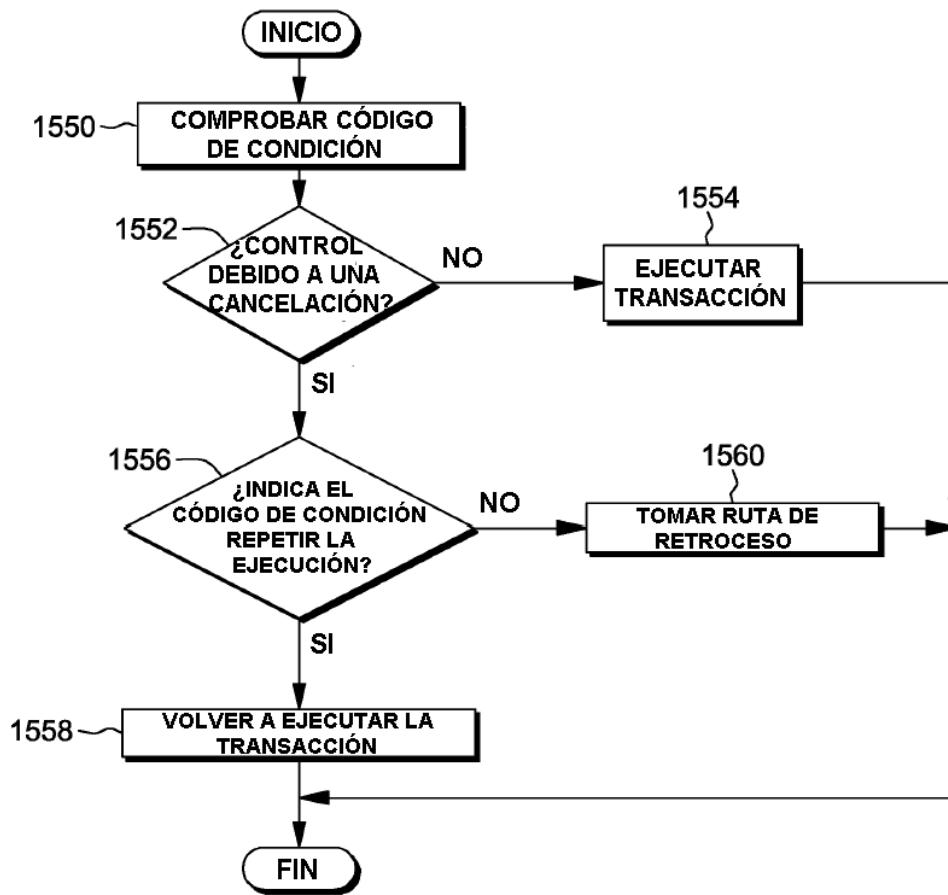


FIG. 15B

INSERTAR UN ELEMENTO EN UNA LISTA DOBLEMENTE ENLAZADA (ANTES)

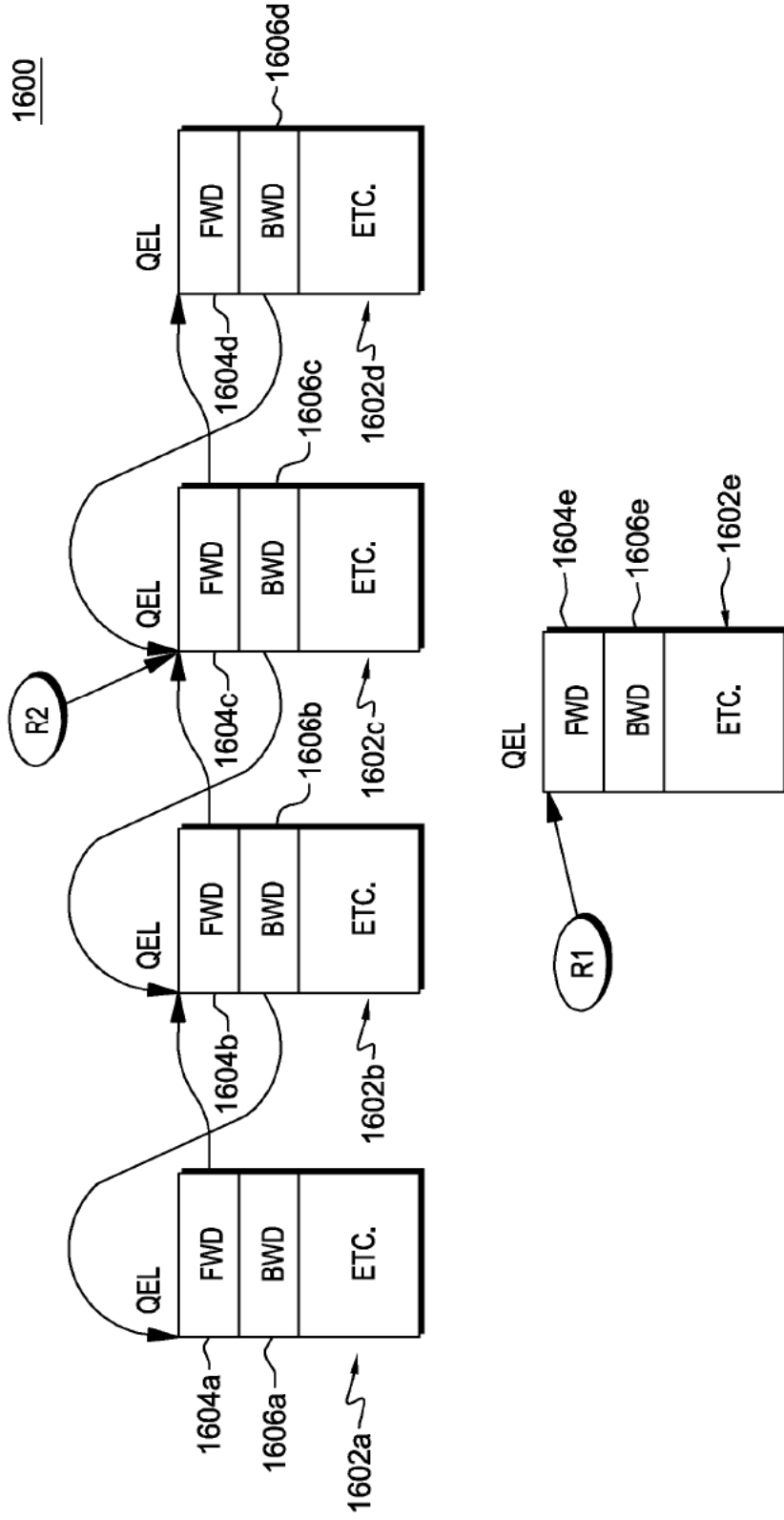


FIG. 16A



INSERTAR UN ELEMENTO EN UNA LISTA DOBLEMENTE ENLAZADA (DESPUÉS)

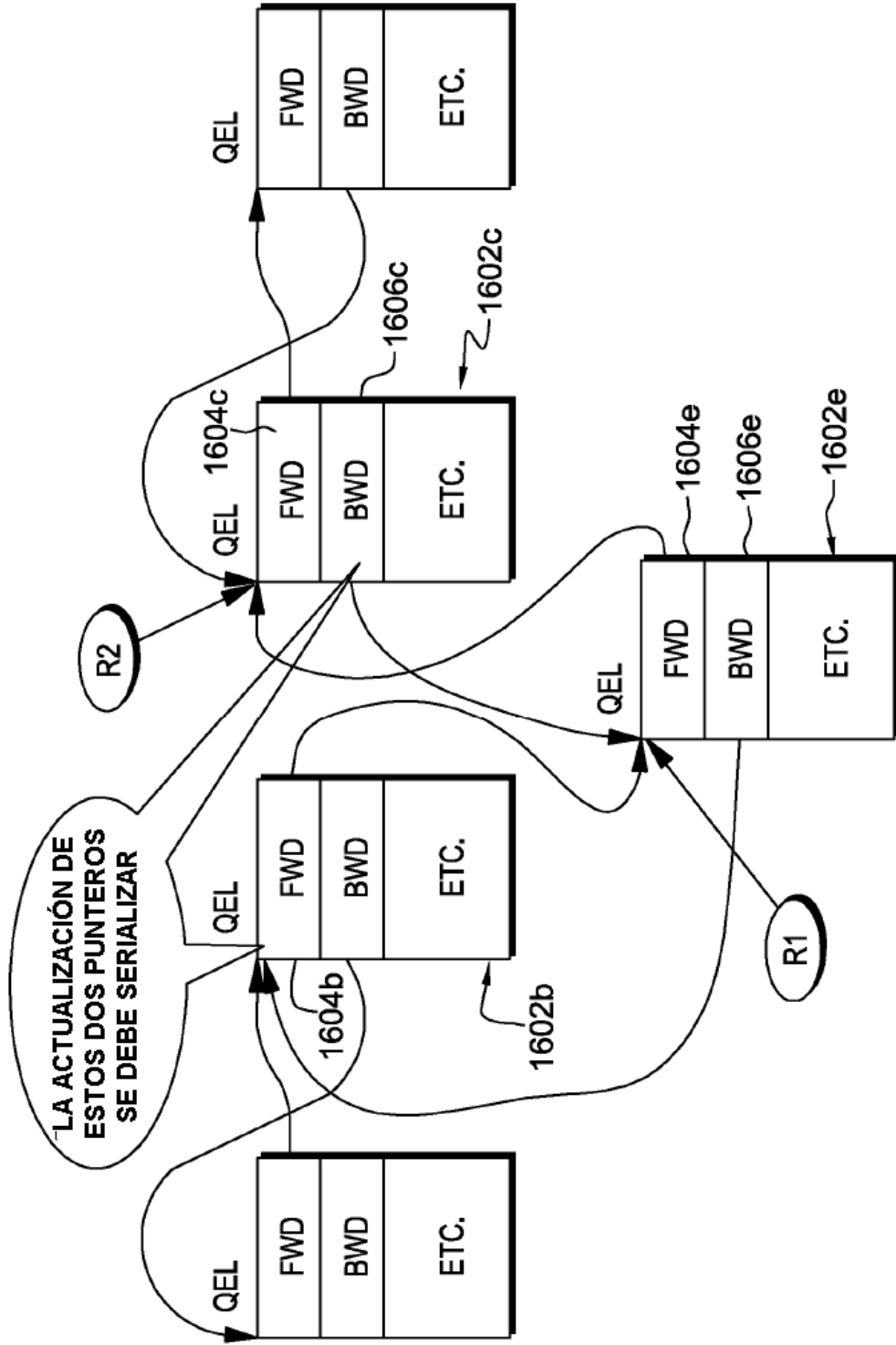


FIG. 16B

**PRODUCTO 1700  
DE PROGRAMA  
INFORMÁTICO**



**FIG. 17**

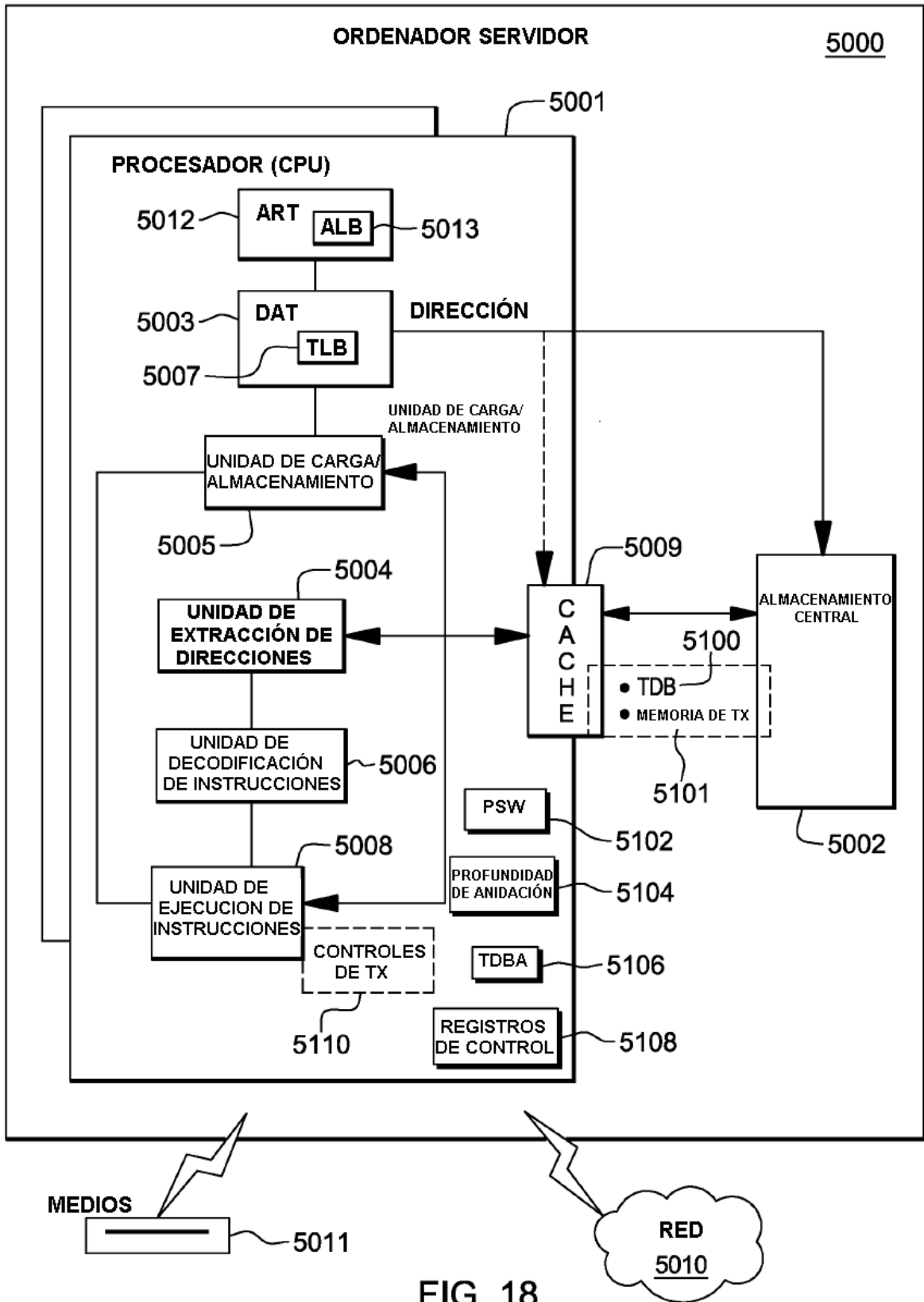


FIG. 18

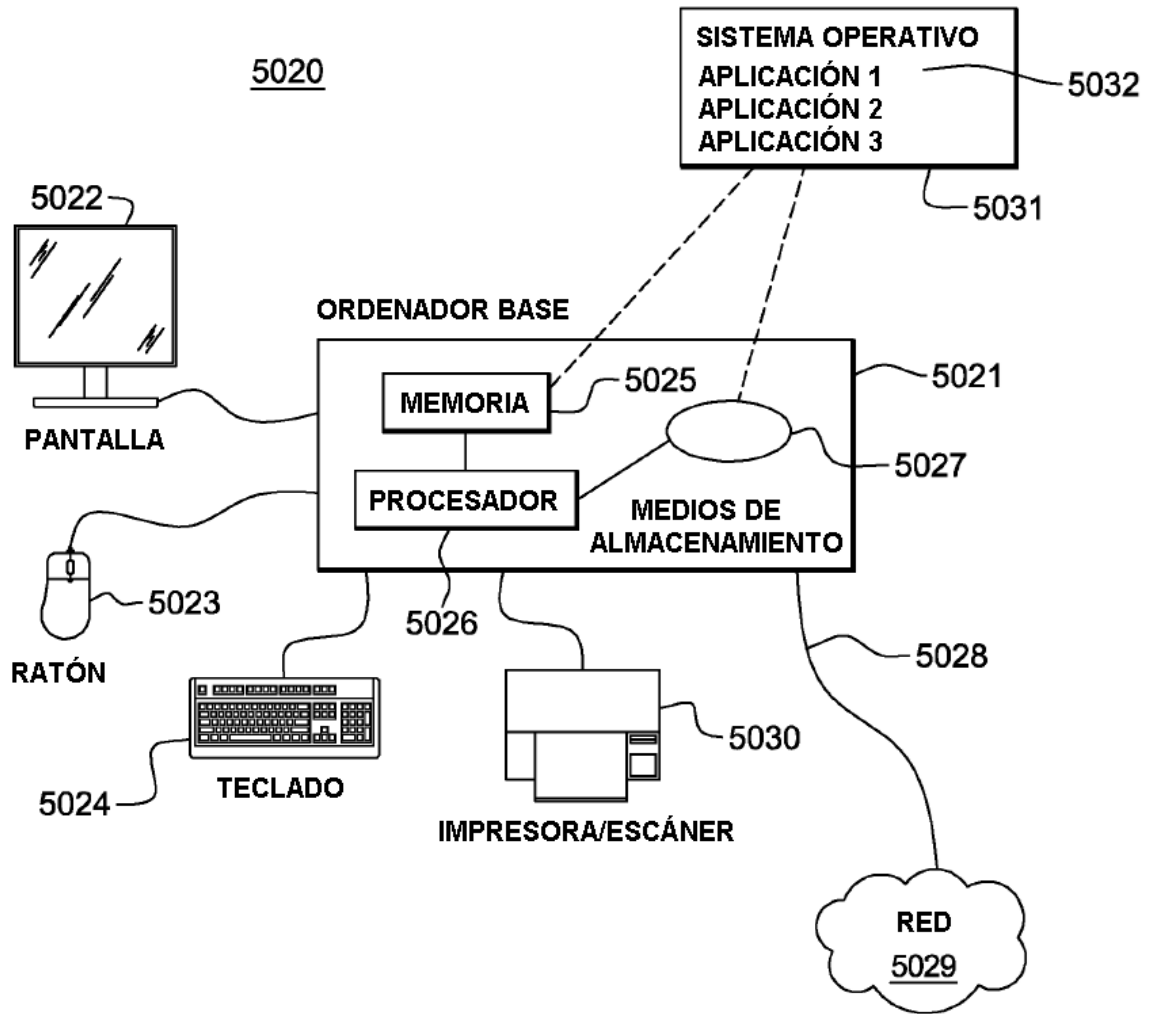
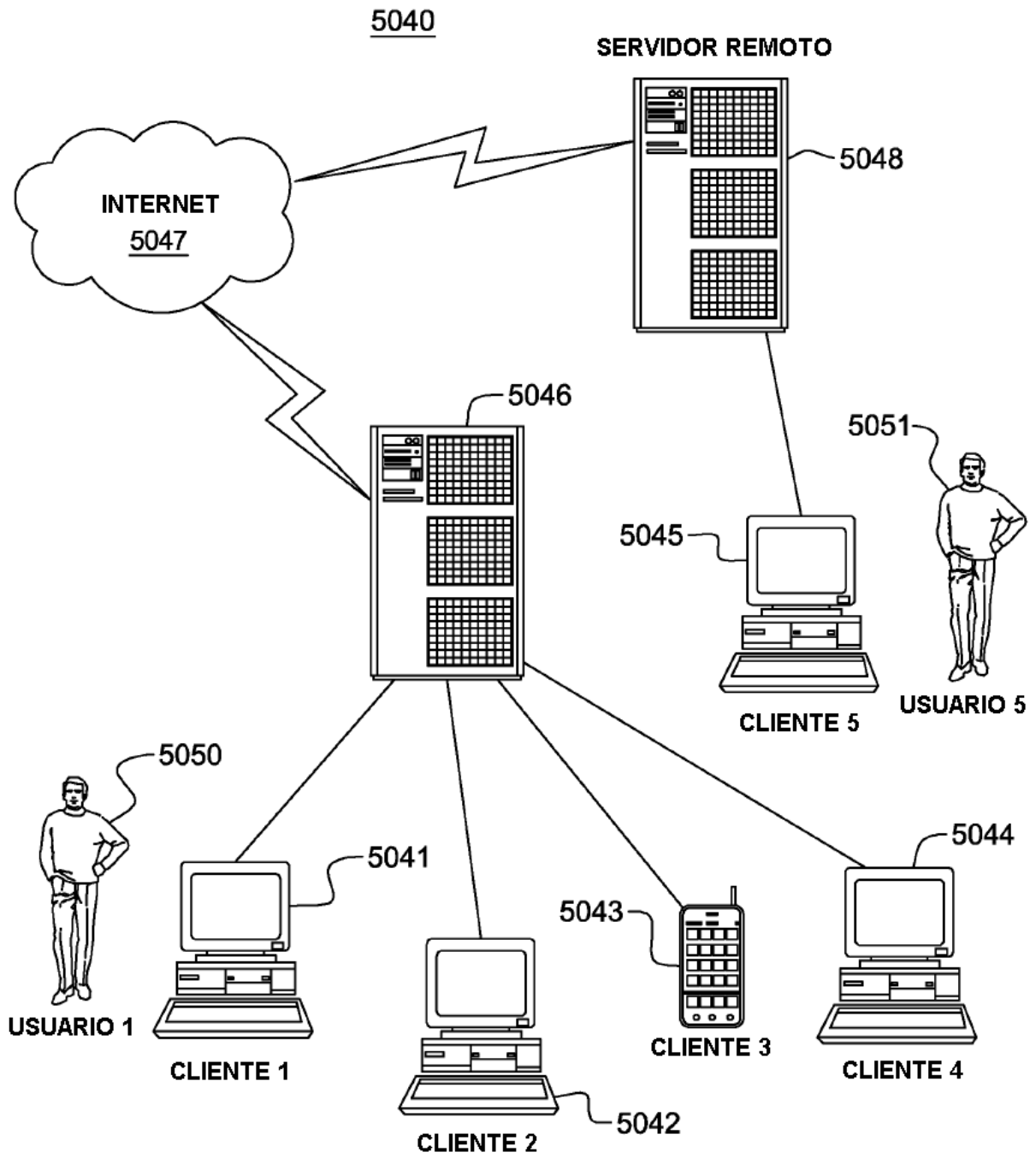


FIG. 19



**FIG. 20**

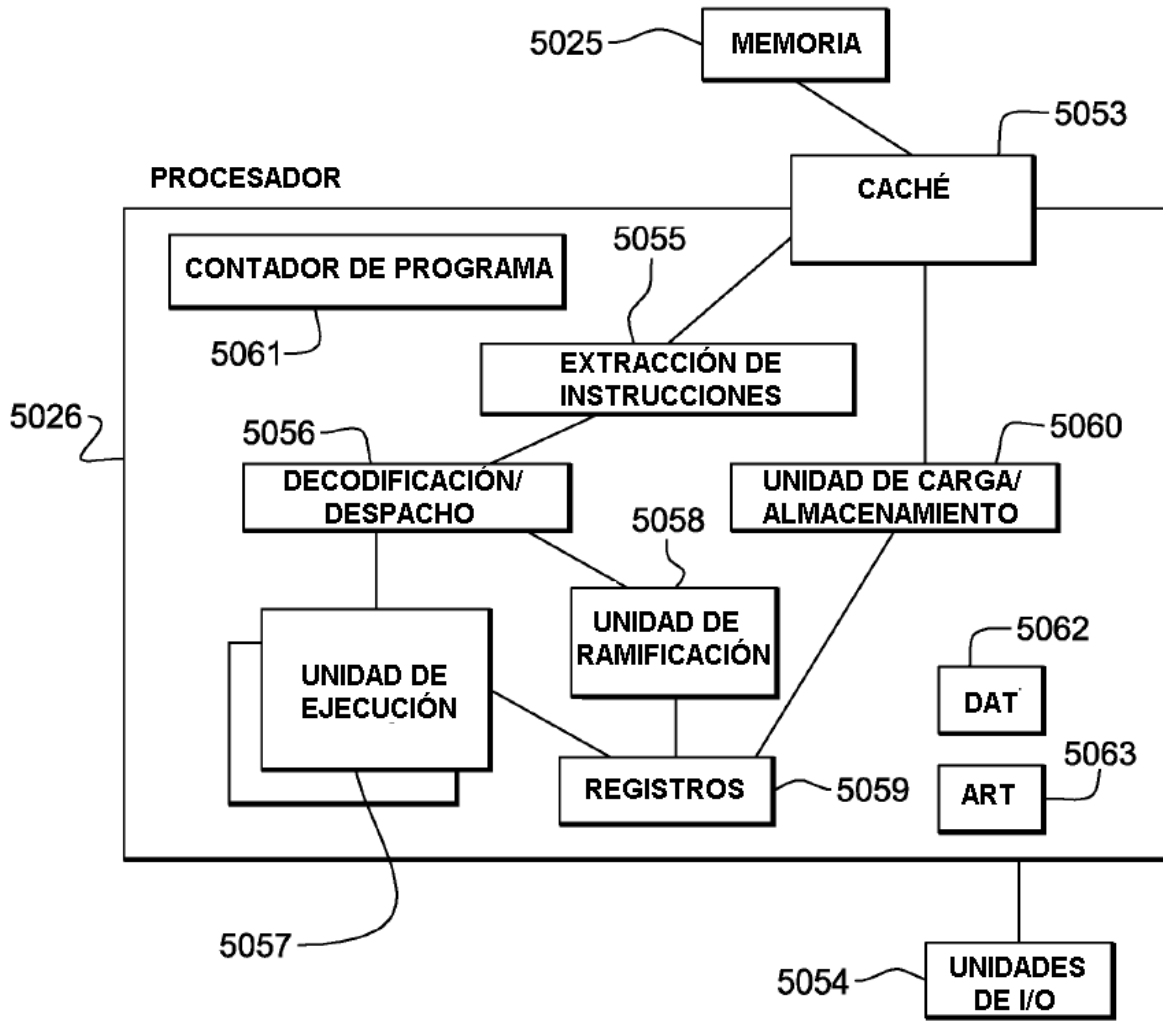


FIG. 21

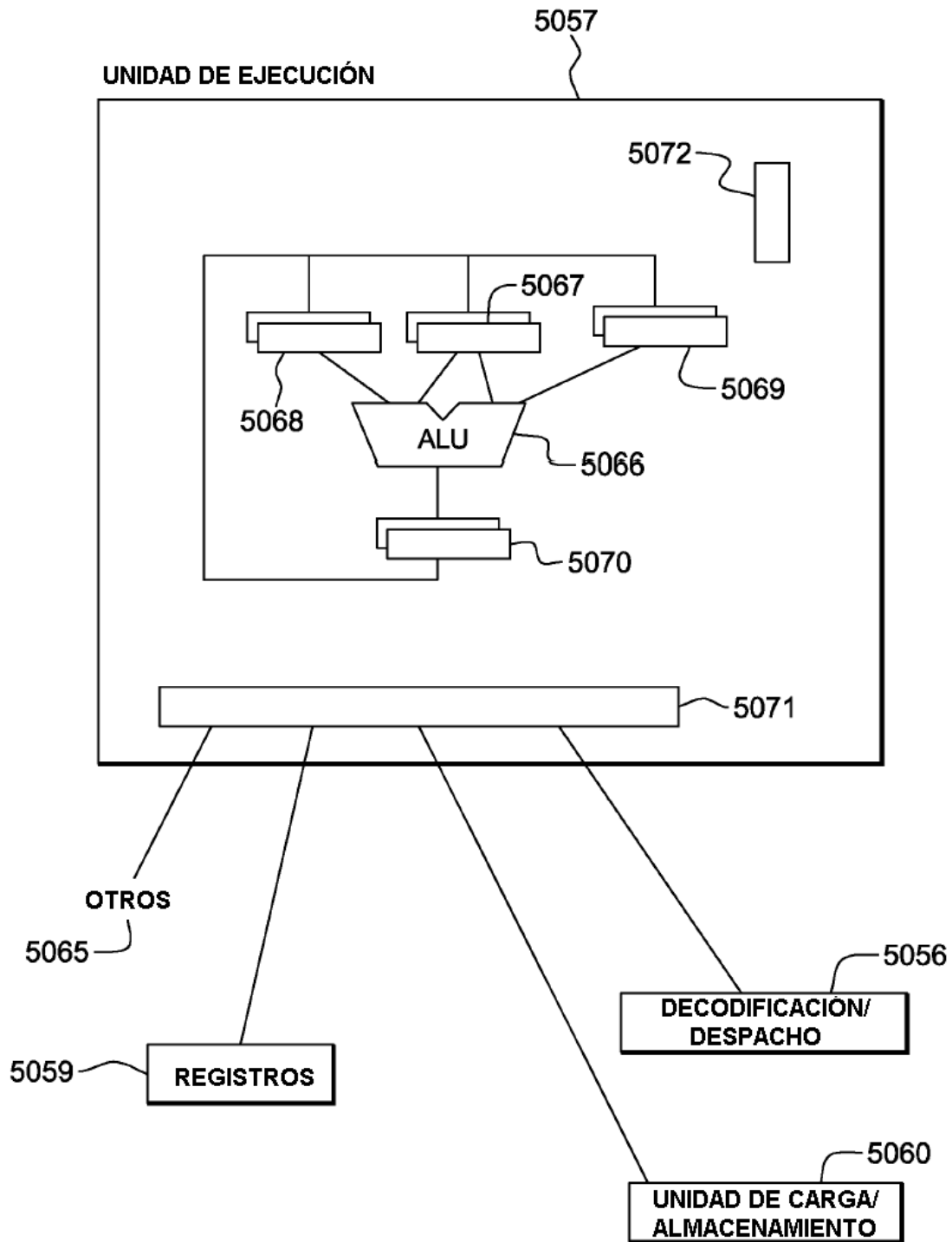


FIG. 22A

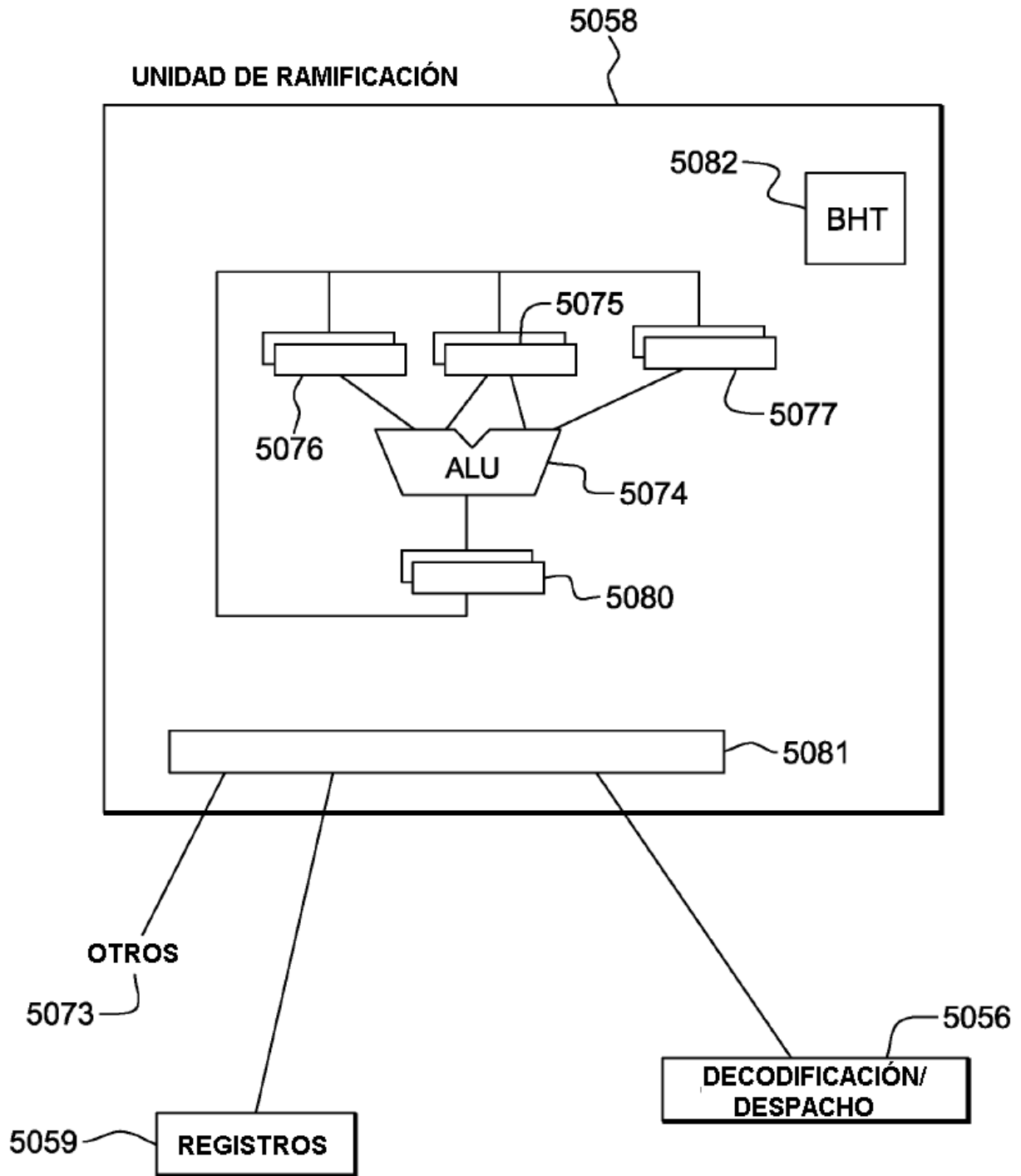


FIG. 22B



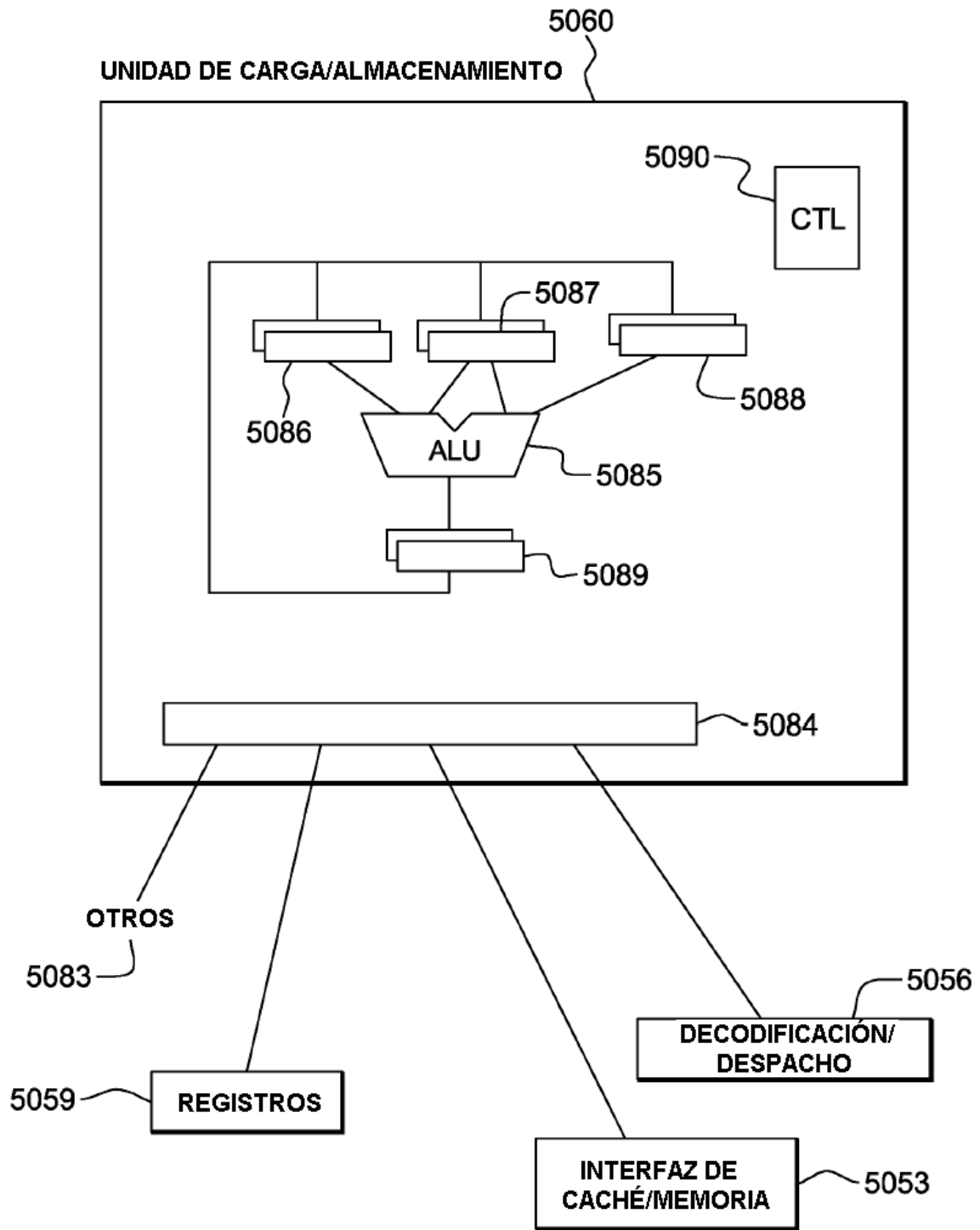


FIG. 22C

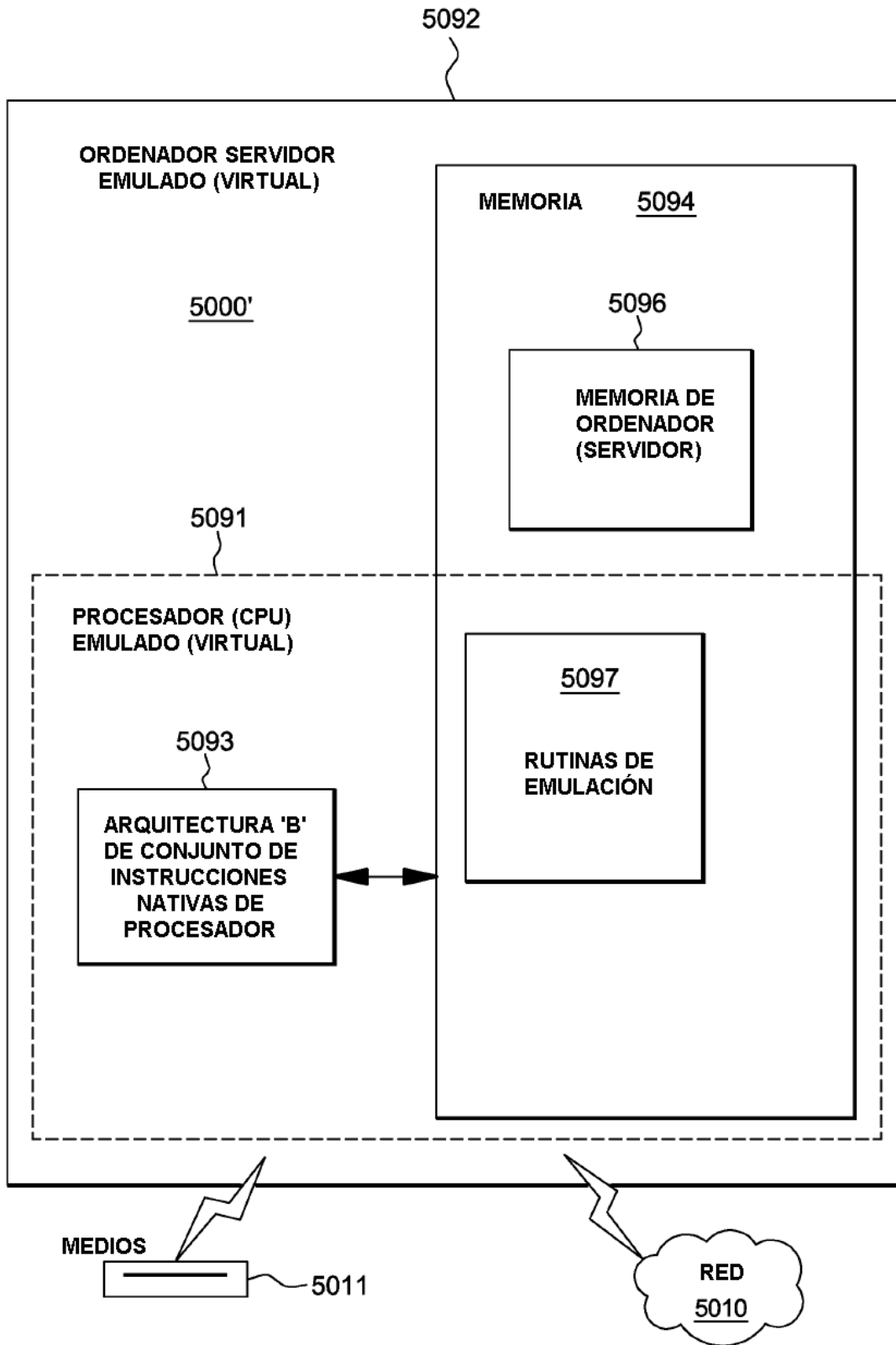


FIG. 23