

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 678 413**

51 Int. Cl.:

G06F 11/36 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **19.03.2008 E 08005116 (2)**

97 Fecha y número de publicación de la concesión europea: **18.04.2018 EP 2037367**

54 Título: **Procedimiento y sistema de alineación de trazas entre subprocesos para un procesador de subprocesos múltiples**

30 Prioridad:

11.04.2007 US 734199

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

10.08.2018

73 Titular/es:

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

72 Inventor/es:

**GIANNINI, LOUIS ACHILLE;
ANDERSON, WILLIAM C. y
CHEN, XUFENG**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 678 413 T3

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

DESCRIPCIÓN

Procedimiento y sistema de alineación de trazas entre subprocesos para un procesador de subprocesos múltiples

5 **CAMPO**

10 [0001] La materia objeto divulgada se refiere a sistemas y procesos de procesamiento de datos tales como el que puede encontrar uso en comunicaciones de datos y en aplicaciones similares. Más particularmente, esta divulgación se refiere a un procedimiento y a un sistema novedosos y mejorados para operaciones de depuración de procesamiento de señales digitales, que incluyen proporcionar un procedimiento y un sistema de alineación de trazas entre subprocesos para un procesador de subprocesos múltiples.

DESCRIPCIÓN DE LA TÉCNICA RELACIONADA

15 [0002] Cada vez más, las telecomunicaciones y otros tipos de equipo electrónico y de video de apoyo, de audio complejo, de videoconferencia y de otras aplicaciones de software enriquecidas implican procesamiento de señales. El procesamiento de señales requiere cálculos matemáticos rápidos y generación de datos en algoritmos complejos pero repetitivos. Muchas aplicaciones requieren cálculos en tiempo real, es decir, la señal es una función continua de tiempo, que debe muestrearse y convertirse en señales digitales para el procesamiento numérico. El procesador debe ejecutar algoritmos que realicen cálculos discretos en las muestras a medida que lleguen.

20 [0003] La arquitectura de un procesador de señales digitales (DSP) está optimizado para manejar dichos algoritmos. Las características de un buen motor de procesamiento de señales incluyen unidades de cálculo aritméticas rápidas y flexibles, flujo de datos no restringido hacia y desde las unidades de cálculo, precisión extendida y rango dinámico en las unidades de cálculo, generadores de direcciones duales, secuenciación eficiente de programas y facilidad de programación.

25 [0004] Una prometedora aplicación de la tecnología DSP incluye sistemas de comunicaciones tales como un sistema de acceso múltiple por división de código (CDMA) que da soporte a comunicaciones de voz y datos, así como a mensajes de texto y otras aplicaciones, entre usuarios a través de un satélite o enlace terrestre. El uso de técnicas CDMA en un sistema de comunicación de acceso múltiple se describe en la patente de Estados Unidos N.º 4.901.307, titulada "SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEM USING SATELLITE OR TERRESTRIAL REPEATERS" [SISTEMA DE COMUNICACIÓN DE ACCESO MÚLTIPLE DE ESPECTRO EXTENDIDO USANDO REPETIDORES SATELITALES O TERRESTRES], y en la patente de Estados Unidos N.º 5.103.459 titulada "SYSTEM AND METHOD FOR GENERATING WAVEFORMS IN A CDMA CELLULAR TELEHANDSET SYSTEM" [SISTEMA Y PROCEDIMIENTO PARA GENERAR FORMAS DE ONDA EN UN SISTEMA TELEFÓNICO CELULAR CDMA], ambas asignadas al cesionario de la materia objeto reivindicada.

30 [0005] Un sistema CDMA está típicamente diseñado para adaptarse a una o más normas. Una de dichas normas de primera generación es el "TIA/EIA/IS-95 Terminal-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System", en lo sucesivo denominado norma IS-95. Los sistemas IS-95 CDMA pueden transmitir datos de voz y paquetes de datos. Un consorcio denominado "3rd Generation Partnership Project" [Proyecto de Asociación de 3ª Generación] (3GPP) ofrece una norma de generación más reciente que puede transmitir de manera más eficiente los paquetes de datos y está incorporado en un conjunto de documentos que incluyen los documentos Nos. 3G TS 25.211, 3G TS 25.212, 3G TS 25.213 y 3G TS 25.214, que están disponibles para el público. La norma 3GPP se denomina en lo sucesivo Norma W-CDMA.

35 [0006] El software operativo DSP complejo que emplea la norma W-DCMA, por ejemplo, requiere herramientas de desarrollo robustas. Dichas herramientas de desarrollo pueden incluir aquellas para la generación de código, la integración, la prueba, la depuración y la evaluación del rendimiento de la aplicación. Al desarrollar y hacer funcionar el software o las aplicaciones DSP complejas, tales como las aplicaciones de telecomunicaciones avanzadas, existe la necesidad de un software de depuración sofisticado pero no intrusivo. Es decir, las aplicaciones de software de depuración no solo deben ser suficientemente robustas para supervisar, probar y dar soporte a la corrección de defectos de software y de problemas de funcionamiento. Al mismo tiempo, es posible que el software de depuración deba funcionar para no interferir con las operaciones del software del procesador de núcleo durante las operaciones de depuración concurrentes. De lo contrario, cualquier problema en el software de procesamiento de núcleo puede no detectarse o detectarse correctamente durante las operaciones de depuración del software.

40 [0007] Durante las operaciones de depuración, existe una necesidad de asociar con las funciones de rastreo de las operaciones de depuración no intrusiva del software para rastrear las operaciones del procesamiento de los subprocesos de funcionamiento en el DSP. Dicho sistema puede proporcionar información sobre los parámetros de estado del DSP para capturar dicha información tanto antes como después de que se produzca un evento específico. Al mismo tiempo, las funciones de rastreo deseadas no pueden agregar ninguna carga significativa al rendimiento del procesador, incluso mientras el DSP funcione a toda velocidad. En combinación con una operación de depuración no intrusiva, dicho proceso de rastreo puede capturar tipos específicos de información. Las funciones

de rastreo, por lo tanto, proporcionan la supervisión y el registro junto con operaciones de depuración no intrusiva en un procesador de subprocesos múltiples.

5 [0008] Un conjunto particularmente útil de información que puede proporcionar una función de rastreo incluye un comportamiento de ejecución entre subprocesos. Es decir, existe la necesidad de un conjunto de funciones de rastreo capaces de interrelacionar datos de rastreo entre diferentes subprocesos de un DSP de subprocesos múltiples. Ningún sistema conocido proporciona dicha información. En las operaciones de depuración del software, el usuario puede desear elegir un punto arbitrario en el tiempo para saber qué instrucciones puede estar ejecutando un procesador de subprocesos múltiples en un punto particular en el tiempo. Esta información puede ser particularmente valiosa en el caso de que diferentes subprocesos activen diferentes operaciones de depuración por subproceso en diferentes momentos.

15 [0009] En un DSP de subprocesos múltiples entrelazados, la ejecución de múltiples secuencias de instrucciones se puede producir al mismo tiempo. Como tal, el procesador se puede ver como varios procesadores de un único subproceso que funcionan de forma independiente. Una vez que dicho procesador puede incluir una unidad de rastreo de ejecución que registra la secuencia de ejecución del tiempo de ejecución de cada uno de los subprocesos de funcionamiento. Estas trazas facilitan la operación de depuración del programa al dividir el flujo del programa en una secuencia de paquetes. En dicho sistema, se pueden agregar campos de números de subprocesos a ciertos paquetes para identificar qué paquete pertenece a cada subproceso. Con dicho enfoque, cada secuencia de paquetes para un subproceso en particular puede recrear la secuencia de ejecución completa incluyendo todos los cambios de flujo del programa y todos los tiempos de instrucción.

25 [0010] Mientras que la identificación de los paquetes pertenece al subproceso que es ventajoso, una función de rastreo no conocida proporciona la capacidad de identificar relaciones de temporización entre subprocesos durante el rastreo de ejecución. Por ejemplo, las diferencias de tiempo entre la activación de trazas pueden ser muy grandes de un subproceso a otro. Cuando el rastreo de subprocesos comienza en diferentes subprocesos en diferentes momentos, es posible que no se pueda alinear la temporización de un subproceso con otros subprocesos que también se estén rastreando.

30 [0011] Por consiguiente, existe la necesidad de un conjunto de funciones de rastreo capaces de funcionar dentro de un proceso de depuración no intrusiva que establezca y mantenga la relación de temporización entre diferentes subprocesos de DSP.

35 [0012] Sin embargo, existe una necesidad adicional de un procedimiento y de un sistema que permita la alineación de los diferentes subprocesos de un procesador de subprocesos múltiples para su uso en asociación con un proceso de macrocelda de trazas incrustado del procesador de subprocesos múltiples.

40 [0013] Aún más, existe una necesidad de un procedimiento de alineación de trazas entre subprocesos y de un sistema para un procesador de subprocesos múltiples entrelazados capaz de funcionar en asociación con una amplia gama de procesos de depuración en silicio no intrusiva que se produzca durante las operaciones de software del procesador de núcleo.

45 * El documento WO 2006/030195 está dirigido a un procedimiento para supervisar la ejecución de subprocesos dentro de una arquitectura de procesador de núcleos múltiples. Un controlador de depuración de subprocesos tiene una interfaz de Tiempo. Esta interfaz proporciona una indicación de 32 bits, que se usa por el controlador de depuración de subprocesos para marcar la fecha de todos los eventos registrados, ya que el controlador de depuración de subprocesos los recibe desde cada subbloque individual de un controlador a través de interfaces de entrada de subbloques. Cada uno de los subbloques del controlador tiene una interfaz de depuración para transportar señales al controlador de depuración de subprocesos. Estas señales de entrada notifican al controlador de depuración de subprocesos los eventos que suceden en cada uno de los subbloques correspondientes del controlador, ya que los subbloques interactúan para administrar y asignar subprocesos individuales entre los recursos de procesamiento individuales. El controlador de depuración de subprocesos también puede filtrar eventos de subbloque para seguir y disparar información.

55 **SUMARIO**

[0014] La necesidad mencionada anteriormente se satisface mediante la materia objeto de las reivindicaciones independientes. Los modos de realización ventajosos están contenidos en las reivindicaciones dependientes.

60 [0015] Se divulgan las técnicas para proporcionar alineación de trazas entre subprocesos en un procesador de subprocesos múltiples, cuyas técnicas cooperan con una macrocelda de trazas incrustada para identificar relaciones de temporización entre subprocesos y, por lo tanto, para establecer y mantener diversas relaciones de temporización entre diferentes subprocesos. El procedimiento y el sistema divulgados aquí mejoran tanto el funcionamiento de un procesador de señales digitales como el uso eficiente de las instrucciones del procesador de señales digitales para aplicaciones de software cada vez más potentes, incluyendo aplicaciones que funcionan en ordenadores personales,

asistentes digitales personales, teléfonos inalámbricos y dispositivos electrónicos similares, así como aumentan la velocidad del procesador digital y la calidad del servicio asociados.

De acuerdo con un aspecto de la materia objeto divulgada, se proporcionan un procedimiento y un sistema para la alineación de trazas entre subprocesos con el procesamiento de trazas de ejecución que incluye datos de temporización de registro relacionados con un evento predeterminado común. Dicho evento predeterminado común puede ser el número de ciclos desde el último rastreo de ejecución iniciado por un subproceso o el número de ciclos desde que todos los subprocesos terminaron el rastreo de la ejecución. El número de ciclos en los que un subproceso inicia el rastreo de ejecución hace referencia al evento común predeterminado para mantener el tiempo del rastreo de ejecución. Los datos relacionados con el evento predeterminado común se actualizan para asociarlos con el momento en que el subproceso inició el rastreo de ejecución. El resultado es permitir alinear los datos de temporización asociados con todos los subprocesos. Los registros interrelacionados permiten reconstruir la información de rastreo de ejecución interdependiente para subprocesos que funcionen en el procesador de subprocesos múltiples, así como sincronizar datos de temporización para todos los subprocesos de funcionamiento.

Estas y otras ventajas de la materia objeto divulgada, así como las características novedosas adicionales, resultarán evidentes a partir de la descripción proporcionada en el presente documento. La intención de este sumario no es ser una descripción exhaustiva de la materia objeto reivindicada, sino proporcionar una breve visión general de algunas de las funciones de la materia objeto. Otros sistemas, procedimientos, características y ventajas proporcionadas aquí resultarán evidentes para el experto en la técnica tras el examen de las siguientes **FIGURAS** y de la descripción detallada. Se pretende que todos estos sistemas, procedimientos, características y ventajas adicionales estén incluidos dentro de esta descripción, dentro del alcance de las reivindicaciones adjuntas.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0016] Las características, la naturaleza y las ventajas de la materia objeto divulgada resultarán más evidentes a partir de la descripción detallada expuesta a continuación cuando se tome en consideración junto con los dibujos en los que unos caracteres de referencia similares identifican de manera correspondiente componentes similares, y en los que:

la **FIGURA 1** es un diagrama de bloques simplificado de un sistema de comunicaciones que puede implementar uno de los diversos ejemplos aquí divulgados;

la **FIGURA 2** ilustra una arquitectura de DSP para llevar a cabo las enseñanzas de la presente divulgación;

la **FIGURA 3** proporciona un diagrama de bloques de arquitectura de un ejemplo de un procesador de señales digitales de subprocesos múltiples que incorpora las ventajas técnicas de la materia objeto divulgada actualmente;

la **FIGURA 4** divulga ciertos aspectos de un núcleo de procesador de señales digitales que aplica las características de la interfaz ISDB/JTAG de la presente divulgación;

la **FIGURA 5** presenta un diagrama de flujo de proceso aplicable a los modos de funcionamiento del procesador de señales digitales, incluyendo el modo de depuración del procedimiento al que pertenece la presente divulgación;

la **FIGURA 6** proporciona un diagrama de bloques que representa una vista funcional general de la macrocélula de trazas incrustada de la presente divulgación;

la **FIGURA 7** muestra un circuito de bloque de disparo del proceso y del sistema de macrocélula de trazas incrustado divulgado;

la **FIGURA 8** ilustra conceptos importantes del proceso de alineación de trazas entre subprocesos divulgado en la actualidad;

la **FIGURA 9** muestra una vista general funcional de la unidad de generación de paquetes según sea aplicable en la materia objeto divulgada;

la **FIGURA 10** muestra una tabla de definiciones para los diversos átomos aplicables a las instrucciones de la materia objeto divulgada en la actualidad;

la **FIGURA 11** proporciona un paquete de dirección de bifurcación para la presente divulgación que da soporte a un campo TID de 32 bits como útil en la presente divulgación;

la **FIGURA 12** presenta los contenidos a modo de ejemplo de un paquete de reinicio isincrónico para su uso en el proceso divulgado en la actualidad;

la **FIGURA 13** representa un ejemplo de un paquete periódico isincrónico para dar soporte a la materia objeto divulgada en la actualidad;

5 la **FIGURA 14** presenta un paquete de recuento de ciclos para su uso en el proceso divulgado en la actualidad;

la **FIGURA 15** muestra un modo de realización de un paquete del formato de encabezado 4 como relevante para la presente divulgación; y

10 la **FIGURA 16** representa un paquete asincrónico para su uso con un modo de realización de la materia objeto divulgada.

DESCRIPCIÓN DETALLADA

15 **[0017]** La materia objeto divulgada para un procedimiento de alineación de trazas entre subprocesos y para un sistema para su uso en asociación con procesos de trazas de ejecución de un procesador de señales digitales de subprocesos múltiples tiene aplicación para el procesamiento de subprocesos múltiples de cualquier tipo para el que los beneficios aquí presentados pueden ser ventajosos. Una de dichas aplicaciones aparece en las telecomunicaciones y, en particular, en los teléfonos inalámbricos que emplean uno o más circuitos de procesamiento de señales digitales. Para explicar cómo se puede usar dicho dispositivo inalámbrico, la **FIGURA 1** proporciona un diagrama de bloques simplificado de un sistema de comunicaciones **10** que puede implementar los ejemplos presentados del procedimiento y del sistema de procesamiento de interrupciones divulgado. En una unidad transmisora **12**, los datos se envían, típicamente en bloques, desde una fuente de datos **14** a un procesador de datos de transmisión (TX) **16** que formatea, codifica y procesa los datos para generar una o más señales analógicas. Las señales analógicas se proporcionan entonces a un transmisor (TMTR) **18** que modula, filtra, amplifica y convierte ascendentemente las señales de banda base para generar una señal modulada. Entonces, la señal modulada se transmite a través de una antena **20** a una o más unidades receptoras.

30 En el sistema receptor **22**, la señal modulada transmitida se recibe por una antena **24** y se proporciona a un receptor (RCVR) **26**. Dentro del receptor **26**, la señal recibida se amplifica, se filtra, se convierte descendentemente, se demodula y se digitaliza para generar en las muestras de fase (I) y (Q). Las muestras se decodifican y se procesan por un procesador de datos (RX) de recepción **28** para recuperar los datos transmitidos. La decodificación y el procesamiento en la unidad receptora **22** se realizan de una manera complementaria a la codificación y al procesamiento realizados en la unidad transmisora **12**. Los datos decodificados se proporcionan a un colector de datos **30**.

40 **[0018]** El procesamiento de señales descrito anteriormente da soporte a transmisiones de voz, a vídeo, a datos por paquetes, a mensajería y a otros tipos de comunicación en una dirección. Un sistema de comunicaciones bidireccional da soporte a la transmisión de datos bidireccional. Sin embargo, el procesamiento de señales para la otra dirección no se muestra en la **FIGURA 1** por simplicidad.

45 El sistema de comunicaciones **10** puede ser un sistema de acceso múltiple por división de código (CDMA), un sistema de comunicaciones de acceso múltiple por división de tiempo (TDMA) (por ejemplo, un sistema GSM), un sistema de comunicaciones de acceso múltiple por división de frecuencia (FDMA) u otro sistema de comunicaciones de acceso múltiple que dé soporte a la comunicación de voz y de datos entre usuarios a través de un enlace terrestre. En un ejemplo específico, el sistema de comunicaciones **10** es un sistema CDMA que se ajusta a la norma W-CDMA.

50 La **FIGURA 2** ilustra la arquitectura del DSP **40** que puede servir como el procesador de datos de transmisión **16** y el procesador de datos de recepción **28** de la **FIGURA 1**. Hacemos hincapié en que el DSP **40** solo representa un ejemplo entre una gran cantidad de posibles ejemplos de procesadores de señales digitales que pueden usar de manera efectiva las enseñanzas y conceptos aquí presentados. En el DSP **40**, por lo tanto, los subprocesos **T0:T5** (números de referencia **42** a **52**) contienen conjuntos de instrucciones de diferentes subprocesos. El circuito **54** representa el mecanismo de acceso a las instrucciones y se usa para extraer instrucciones para los subprocesos **T0:T5**. Las instrucciones para el circuito **54** están en cola en la cola de instrucciones **56**. Las instrucciones en la cola de instrucciones **56** están listas para emitirse en la tubería de procesador **66** (véase a continuación). Desde la cola de instrucciones **56**, se puede seleccionar un único subproceso, por ejemplo, el subproceso **T0**, mediante el circuito lógico de problema **58**. El archivo de registro **60** de un subproceso seleccionado se lee y los datos leídos se envían a las trayectorias de datos de ejecución **62** para **SLOT0: SLOT3**. **SLOT0: SLOT3**, en este ejemplo, proporcionan la combinación de agrupamiento de paquetes empleada en el presente aspecto de la invención. La salida de las trayectorias de datos de ejecución **62** va al circuito de escritura de archivos de registro **64**, también configurado para alojar subprocesos individuales **T0: T5**, para devolver los resultados de las operaciones del DSP **40**. Por tanto, la ruta de datos desde el circuito **54** y antes del circuito de escritura de archivos de registro **64** forma una tubería de procesamiento **66**. El presente modo de realización puede emplear un sistema híbrido de un procesador de elementos heterogéneos (HEP) que use un único procesador con hasta seis subprocesos, **T0: T5**. La tubería de procesador **66** tiene seis etapas, que coinciden con la cantidad mínima de ciclos de procesador necesarios para

extraer un elemento de datos del circuito **54** a los registros **60** y **64**. El DSP **40** ejecuta simultáneamente instrucciones de diferentes subprocesos **T0: T5** dentro de una tubería de procesador **66**. Es decir, el DSP **40** proporciona seis contadores de programa independientes, un mecanismo de etiquetado interno para distinguir las instrucciones de los subprocesos **T0: T5** dentro de la tubería de procesador **66** y un mecanismo que dispara un conmutador de subprocesos. La sobrecarga del conmutador del subproceso varía de cero a solo unos pocos ciclos.

[0019] El DSP **40**, por lo tanto, proporciona un procesador de señales digitales de uso general diseñado para alto rendimiento y baja potencia a lo largo de una amplia variedad de aplicaciones de procesamiento de señales, imágenes y vídeo. La **FIGURA 3** proporciona una breve visión general de la arquitectura del DSP **40**, que incluye algunos aspectos de la arquitectura del conjunto de instrucciones asociado para una manifestación de la materia objeto divulgada. Las implementaciones de la arquitectura del DSP **40** dan soporte al procesamiento de subprocesos múltiples intercalados (IMT). En este modelo de ejecución, el hardware da soporte a la ejecución simultánea de múltiples subprocesos de hardware **T0: T5** entrelazando instrucciones de diferentes subprocesos en la tubería. Esta característica permite al DSP **40** incluir una frecuencia de reloj agresiva a la vez que mantiene un alto uso del núcleo y de la memoria. El procesamiento de IMT proporciona un alto rendimiento sin la necesidad de costosos mecanismos de compensación tales como ejecución fuera de orden, redes de envío extensas, etc. Además, el DSP **40** puede incluir variaciones del procesamiento de IMT, tales como las variaciones y enfoques novedosos divulgados en las Solicitudes de Patente de Estados Unidos cedidas comúnmente por M. Ahmed, et al, y titulados "Variable Interleaved Multi-threaded Processor Method and System" y "Method and System for Variable Thread Allocation and Switching in a Multi-threaded Processor."

[0020] La **FIGURA 3**, en particular, proporciona un diagrama de bloques de arquitectura de procesamiento de núcleo **70** para el DSP **40** como se aplica a un solo subproceso que puede emplear las enseñanzas de la materia objeto divulgada. El diagrama de bloques **70** representa la memoria caché de instrucciones compartidas **72** que recibe instrucciones a través de la interfaz de bus (I/F) **73** del bus AXI **74**, cuyas instrucciones incluyen instrucciones mixtas de 16 bits y 32 bits. Estas instrucciones llegan al secuenciador **76**, al registro de control de usuario **78** y al registro de control de supervisor **80** de los subprocesos **T0: T5**. La arquitectura del sistema a nivel del núcleo de la materia objeto divulgada también incluye el sistema de depuración en silicio (ISDB) **82**, que interconecta el procesador de núcleo **70** a través de la interfaz JTAG **84**, que se describen con más detalle a continuación.

[0021] El secuenciador **76** proporciona instrucciones superescalares bidireccionales híbridas e instrucciones VLIW de cuatro vías para la unidad de tubería S **86**, para la unidad de tubería M **88**, para la unidad de tubería [carga] LD **90** y para la unidad de tubería LD-ST [Almacenamiento] **92**, todas las cuales se comunican con registros generales **94**. AXI Bus **74** también se comunica a través del Bus I/F **73** con instrucciones LD/ST de la memoria caché de datos compartidos **96** a los subprocesos **T0: T5**. Las señales opcionales de la memoria caché **L2/TCM 98** incluyen instrucciones LD/ST con datos compartidos TCM **100**, cuyas instrucciones LD/ST fluyen además a los Registros Generales de subprocesos **94**. Desde el bus periférico AHB **102**, el controlador específico de MSM **104** se comunica con las interrupciones con **T0: T5**, incluyendo las instrucciones del controlador de interrupción, las instrucciones de depuración y las instrucciones de temporización. Los registros de control globales **106** se comunican con las instrucciones de registro de control con los subprocesos **T0: T5**.

[0022] El DSP **40**, por lo tanto, incluye seis núcleos DSP virtuales, conteniendo cada uno registros de control globales **106** y registros de control de supervisor privados **80**. Los registros de control globales **106** se comparten entre todos los subprocesos. Cada subproceso comparte una memoria caché de datos común y una memoria caché de instrucciones común. Las operaciones de carga, almacenamiento y extracción se sirven por una interfaz de bus común. El bus AXI de alto rendimiento **74** y el bus AHB de menor rendimiento **102** se usan para conectar el tráfico de datos y de instrucciones a la memoria y a los periféricos fuera del núcleo. Una entrada integrada de memoria de nivel dos (memoria caché y/o TCM) **98** es opcional. El acceso periférico puede ser a través de cargas y almacenes mapeados en memoria. La partición de la dirección física entre AHB y AXI se puede configurar al nivel de MSM.

[0023] Claramente, la arquitectura presentada para el DSP **40** puede evolucionar y cambiar con el tiempo. Por ejemplo, el número de memorias caché de instrucciones que puede usar el DSP **40** podría cambiar de seis a uno u otros números de caché. El envío superescalar, los datos L1 en TCM **98** y otros aspectos arquitectónicos pueden cambiar. Sin embargo, la presente materia objeto puede tener relevancia continua en una amplia variedad de configuraciones y para una gran familia de modificaciones de DSP **40**.

[0024] El ISDB **82**, a través de la interfaz JTAG **84**, proporciona un depurador de hardware para DSP **40**. El ISDB **82** proporciona funciones de depuración de software a través de la interfaz JTAG **84** compartiendo registros de un solo sistema o supervisor. Estos registros están divididos en registros de control de supervisor **80** por subproceso, así como en registros de control globales **106** entre todos los subprocesos. Los registros de control del sistema se usan para las actividades de gestión de la interrupción por subproceso y del control de excepción y por la memoria de subprocesos. Los registros globales permiten interactuar con el ISDB **82** para operaciones de depuración.

[0025] El ISDB **82** permite a los desarrolladores de software depurar su software, mientras el DSP **40** funciona. El hardware ISDB **82**, en combinación con un programa de depuración de software que funciona en el ISDB **82**, se puede usar para depurar el software del sistema operativo del DSP **40**. El ISDB **82** da soporte a la depuración de

subprocesos de hardware individualmente. Los usuarios pueden suspender la ejecución de subprocesos, ver y alterar registros de subprocesos, ver y alterar las instrucciones y la memoria de datos, los subprocesos de una sola etapa, las instrucciones de cosas para los subprocesos y reanudar la ejecución de subprocesos. Los usuarios de confianza tienen acceso a todas las funciones del ISDB **82**, mientras que los usuarios que no son de confianza tienen acceso a un subconjunto de características.

[0026] El ISDB **82** puede interactuar con una tarjeta de interfaz de depurador para comunicarse con el software de depuración del ISDB **82** que reside en un contador de programa (PC), aunque todo a través de la interfaz JTAG **84**. El software del depurador del servidor puede interactuar con el ISDB **82** leyendo y escribiendo los registros de control de ISDB. La comunicación, por ejemplo, puede ser a través de un paquete de **40** bits que identifique el registro de ISDB al que se deba realizar la lectura/escritura, así como una carga de datos de **32** bits. Un formato de paquete que da soporte a esta operación puede ser de hasta **64** registros de control que pueden tener 32 bits de ancho cada uno.

[0027] La **FIGURA 4** muestra aspectos importantes de la interfaz ISDB/JTAG **110** entre el mecanismo de depuración y el procesador de núcleo de la materia objeto divulgada. En asociación con la arquitectura de núcleo **70** del DSP **40**, el ISDB **82** se comunica con la JTAG **84** a través de la ruta de interfaz JTAG **112**, desde el circuito de ISDB de JTAG **114**. El circuito de ISDB de JTAG **114** procesa flujos de datos entre la JTAG **84** y el ISDB **82**. El circuito de ISDB de JTAG **114** interconecta además el circuito de ISDB JTAGSync **116**. El circuito ISDB JTAGSync **116** se comunica además con el controlador de ISDB **118**, con la unidad de instrucción (IU) **120** y con la unidad de control (CU) **122**. Particularmente, el circuito de ISDB JTAGSync **116** interconecta el circuito lógico de IU de ISDB **120** y el controlador de CU de ISDB **126** de la CU **122**. El controlador de CU de ISDB **126** se comunica con el circuito lógico de CU de ISDB **128**, así como con el controlador de ISDB **118**. Las salidas de control del controlador de ISDB **118** incluyen la salida de datos de ISDB **130**, la señal de reinicio de ISDB **132** y la interrupción de ISDB **134**. Otras interfaces para el controlador de ISDB **118** incluyen la interfaz MCD **136** y el disparador de punto de interrupción ETM **138**.

[0028] La **FIGURA 5** presenta un diagrama de modo de procesamiento **140** para los diversos aspectos de control de modo del DSP **40**, incluyendo las operaciones del ISDB **82** durante los procesos de depuración. En la **FIGURA 5**, el DSP **40** da soporte a modos de procesamiento que son tanto globales para todos los subprocesos como locales para subprocesos individuales. Cada subproceso de hardware del DSP **40** da soporte individualmente a dos modos de ejecución, el modo USUARIO **142** y el modo SUPERVISOR **144**, y a tres modos de no procesamiento, el modo ESPERA **146**, el modo DESACTIVADO **148** y el modo DEPURAR **150**, todos los cuales pueden aparecer en la **FIGURA 5**. El modo de un subproceso es independiente de otros subprocesos, por ejemplo, un subproceso puede estar en el modo ESPERA **146** mientras que otro está en el modo USUARIO **142**, y así sucesivamente.

[0029] El diagrama de estado de modo por subproceso de la **FIGURA 5** recibe soporte de diversas instrucciones o eventos. Estos incluyen "Excepto" o un evento de excepción interno, un evento de interrupción "Int" o externo, una "RTE" o instrucción de devolución de software del modo de excepción y una "SSR" o actualización a la instrucción de registro SSR, una instrucción "Detener" o de instrucción de detener el software que se puede ingresar desde cualquier modo, un "Inicio" o una Instrucción de inicio de software que también se puede introducir desde cualquier modo, una "trampa" o instrucción de trampa de software, una instrucción de "espera" o de espera de software, una "Reanudar" o Instrucción de Reanudación del software, un "DE" o un evento de depuración y una instrucción "DR" o de Depurar. Si bien las funciones en diferentes implementaciones de la materia objeto reivindicada pueden variar ligeramente de las presentadas aquí, los significados de "Inicio", "Esperar", "Reanudar", "DE" y/o "DR" pueden recibir sus interpretaciones más amplias, consistente con el alcance de la materia reclamada.

[0030] Los registros están disponibles en el DSP **40** tanto en el modo USUARIO **142** como en el modo SUPERVISOR **144**. Los registros de modo de usuario están divididos en un conjunto de registros generales y en un conjunto de registros de control. Los registros generales se usan para todos los cálculos generales, incluyendo la generación de direcciones, la aritmética escalar y vectorial. Los registros de control admiten funcionalidades de uso especial, tales como bucles de hardware, predicados, etc. Los registros de uso general tienen 32 bits de ancho y se puede acceder a ellos como registros únicos o como pares alineados de dos registros. El archivo de registro general proporciona todos los operandos para las instrucciones, incluyendo las direcciones para cargar/almacenar, los operandos de datos para las instrucciones numéricas y los operandos vectoriales para las instrucciones vectoriales.

[0031] El modo DEPURAR **150** proporciona un estado especial donde el subproceso está esperando órdenes del ISDB **82**. Cada vez que se produce un evento de depuración de ISDB, tal como por ejemplo la ejecución de una instrucción de punto de interrupción de software, un comando de punto de interrupción de ISDB **82** o la aparición de un punto de interrupción de hardware, los subprocesos indicados pueden entrar en el modo DEPURAR **150**. Mientras está en el modo DEPURAR **150**, el núcleo se controla por el ISDB **82** a través de comandos desde la interfaz JTAG **84**. Cuando el ISDB **82** libera el subproceso debido a la ejecución de un comando de reanudación, el subproceso puede reanudar su funcionamiento de acuerdo con su configuración de modo actual. Cuando un subproceso está en el modo DEPURAR **150**, se controla por el ISDB **82** y no puede controlarse por otros subprocesos. Las instrucciones Esperar, Reanudar o Detener de un subproceso en ejecución, que se enrutan hacia

un subproceso en el modo DEPURAR **150**, pueden ignorarse. De manera similar, los subprocesos del modo DEPURAR **150** pueden ignorar una Interrupción No Enmascarable (NMI).

[0032] Un modo REINICIO DE HARDWARE (no mostrado en la **FIGURA 5**) y el modo DEPURAR **150** son globales a todos los subprocesos. Cada vez que se activa el pin de reinicio de hardware, independientemente del estado de procesamiento de cualquier subproceso, el DSP **40** puede entrar en un modo REINICIO DE HARDWARE. En el modo REINICIO DE HARDWARE, todos los registros se configuran a sus valores de reinicio. No se puede procesar hasta que el pin de reinicio de hardware se desactive. Cuando se active el pin de reinicio, el procesador puede pasar al modo de reinicio y todos los registros pueden reiniciarse a sus valores de REINICIO DEL HARDWARE. Después de desactivar el pin de reinicio, el subproceso **T0** puede recibir una interrupción de reinicio suave. Esto puede causar que el subproceso **T0** entre en el modo SUPERVISOR **144** y comience a ejecutarse en la ubicación del vector de reinicio. Todos los otros subprocesos pueden permanecer apagados. En este punto, el software es libre de controlar las transiciones de modo para cada subproceso individualmente.

[0033] Las **FIGURAS 6 a 17** se refieren a las características novedosas y ventajosas divulgadas en la actualidad de una unidad de la macrocélula de trazas incrustada (ETM) del DSP **40**, que mejora la depuración del código por parte del usuario mediante la captura en tiempo real de información detallada sobre el flujo de ejecución del software. La ETM supervisa y registra de manera no intrusiva la ejecución del DSP **40** seleccionado, forma la información de ejecución en paquetes y envía el flujo de paquetes fuera de chip o a una memoria en chip conocida como ETB. La ETM también contiene una serie de mecanismos para limitar o enfocar la generación de información de trazas a la región de interés. Usando el flujo de paquetes, se puede crear una reconstrucción de la ejecución, dando al usuario una visibilidad directa del comportamiento del tiempo de ejecución del código.

[0034] La ETM, por lo tanto, proporciona funciones integrales de depuración y trazas para el DSP **40** y para otros procesadores de señales digitales similares. Estas funciones permiten capturar información sobre el estado del procesador tanto antes como después de un evento específico, sin agregar carga al rendimiento del procesador, incluso cuando el DSP **40** funcione a toda velocidad. La ETM puede configurarse en software para capturar solo información de trazas y solo después de una secuencia específica de condiciones. Un puerto de trazas dedicado y configurable y un FIFO permiten entonces que los datos de trazas comprimidos se lean desde el chip por un analizador de puerto de trazas externo sin interrumpir ni afectar al procesador.

[0035] El puerto de trazas puede configurarse a partir de un bus de datos de 1 a 32 bits, con el reloj de trazas independiente para el reloj de núcleo. Por ejemplo, la velocidad de transferencia de datos de la ETM puede ser la mitad del reloj de núcleo y del número de pines aumentados para mantener el ancho de banda de datos. De manera similar, el número de pines se puede reducir a la mitad y la velocidad de transferencia de datos puede aumentarse. La ETM se puede usar tanto en modo autónomo como dentro de un entorno multinúcleo para permitir que el desarrollador vea trazas correlacionadas simultáneas de núcleos asincrónicos múltiples.

[0036] La **FIGURA 6** proporciona el diagrama de bloques **160** que representa una variedad de funciones de ETM globales que dan soporte a la presente divulgación. El procesador de núcleo de DSP **70** interconecta la ETM **162**, que incluye el circuito de disparo y filtrado **164** y el circuito de compresión y paquetización **166**. Después del procesamiento por el circuito de disparo y filtrado **164** y por el circuito de compresión y paquetización **166**, la salida de ETM **168** fluye al depósito de trazas **170**, que puede ser, por ejemplo, un circuito de la memoria intermedia de trazas incrustada (ETB) o un circuito fuera de chip. Desde el depósito de trazas **170**, los registros de ejecución del software fluyen como la salida **172** para depurar el servidor **173**. El servidor de depuración **173** incluye el componente decompresor **174** para recibir la salida de depósito de trazas **172** y generar desde allí el flujo de ejecución reconstruido **176**. La ETM **162** recibe la entrada de control **178** desde la JTAG **84**, que la JTAG de entrada **84** genera en respuesta a los datos e instrucciones a partir del Servidor Depurado **173**.

[0037] Como se muestra en la **FIGURA 6**, las ETM **162** supervisan la tubería del DSP **40**. Con esta información, la ETM **162** realiza dos funciones principales: filtrado/disparo y compresión/paquetización. Las operaciones de filtrado y disparo se programan por el usuario a través de la interfaz JTAG **84** y se usan para definir cuándo activar y desactivar el rastreo. La unidad de compresión/paquetización toma la información de ejecución del DSP **40** y la forma eficientemente en paquetes que se envían desde la ETM **162** a través del puerto de trazas. El flujo de trazas que sale de la ETM **162** se alimenta al depósito de trazas **170**. El depósito de trazas **170** proporciona una gran capacidad de memoria para registrar registros de trazas y puede ser en chip o fuera de chip. El depósito en chip se conoce como una memoria intermedia de trazas incrustada (ETB). El componente decompresor **174** es un componente de software que se ejecuta en el Servidor Depurado **173** que toma el flujo de paquetes del depósito de trazas **170** y, junto con la imagen del programa, reconstruye el flujo de ejecución del DSP **40**, proporcionando al usuario una visibilidad detallada de la tubería de DSP **66**.

[0038] La ETM **162** proporciona secuenciación y temporización de instrucciones de trazas para los seis subprocesos, así como también la capacidad de registrar y enviar recuentos de perfiles (fallos de la memoria caché, conflictos bancarios y errores de microtlb). La ETM **162** puede activarse en direcciones de PC y de LDST, así como en datos de LDST. La ETM **162** da soporte a detecciones de eventos en serie y externos. Además, la ETM **162** también puede generar eventos de disparo de punto de interrupción de ISDB, evento de disparo externo e

interrupciones del DSP **40**. La ETM **162** es programable a través de la JTAG **84** y puede dar soporte a un depósito de trazas de ETB dedicado **170** de, en un ejemplo, 512x32 bits, en un ejemplo. La ETM **162** puede contener bloques de 4 disparos (cada uno con **2** comparadores de direcciones y **1** de datos) y puede contener un secuenciador de 3 estados. El rastreo de la ETM **162** puede funcionar bajo el control de un registro habilitado seguro del DSP **40** y puede programarse para su funcionamiento durante el colapso de energía del DSP **40**.

La ETM **162** genera una traza de instrucciones como registro de la progresión completa del contador de programa para un subproceso sobre una ventana dada en el tiempo. Opcionalmente, la temporización de la progresión del contador del programa (*es decir*, la identificación de los ciclos de bloqueo) también se puede incluir en una traza de instrucciones. Los mecanismos de recursos de eventos se usan para definir cuándo generar estas trazas de instrucciones. Las funciones de disparo y filtrado se controlan a través de la programación de los recursos de evento. Más detalladamente, los recursos de evento controlan el filtrado, el disparo y la generación de punto de interrupción de ISDB. El filtrado incluye las funciones de decidir cuándo habilitar y deshabilitar una traza de instrucciones. El disparo implica decidir cuándo insertar un marcador de disparo en el flujo de paquetes. La determinación del punto de interrupción de ISDB implica especificar las condiciones bajo las cuales el ISDB **82** genera y responde a un punto de interrupción para las operaciones de depuración.

[0039] La ETM **162** contiene varios recursos de eventos primarios (*por ejemplo*, comparadores de direcciones y datos) para detectar cuándo se producen condiciones específicas dentro del DSP **40** (*por ejemplo*, si se ejecuta un determinado PC, o si se lee cierta ubicación de memoria). Además, hay recursos de eventos secundarios (bloques de disparo y el secuenciador) que permiten la detección de disposiciones de eventos más complejos.

El depósito de trazas de ETB **170** proporciona un área de memoria en chip donde la información de trazas se almacena durante la captura en lugar de exportarse inmediatamente a través de un puerto de trazas en los pines del dispositivo. La información almacenada puede leerse entonces a una velocidad de reloj reducida desde el depósito de trazas de ETB **170** una vez que se haya completado la captura. Esto se hace a través de la interfaz JTAG **84**. Este proceso de dos etapas elimina la necesidad de un puerto de trazas ancho que use muchos pines del dispositivo de alta velocidad. Efectivamente, se crea un puerto de trazas "de cero pines" donde el dispositivo ya tiene un puerto JTAG en los pines. El depósito de trazas de ETB **170** puede aceptar datos a una frecuencia más alta y con el puerto de datos de 32 bits completo, excediendo las limitaciones de ancho de banda del puerto de trazas y puede integrarse con un bloque de RAM suministrado por el integrador de sistema.

[0040] En un ejemplo, el depósito de trazas de ETB **170** tiene un tamaño de 2KB dispuesto como 512 entradas, cada una de 32 bits de ancho. Sin embargo, otros tamaños para el depósito de trazas de ETB están claramente dentro del alcance de la materia objeto divulgada. El depósito de trazas de ETB **170** se interconecta con el usuario a través de un conjunto de registros accesibles JTAG. Cada registro puede leerse o escribirse a través de la interfaz JTAG **84**. Estos registros se usan para configurar el depósito de trazas de ETB **170** para una sesión de captura de trazas y para leer los contenidos del depósito de trazas de ETB **170** una vez que se complete la captura de trazas. El depósito de trazas de ETB **170** proporciona un puntero de lectura como un índice en la matriz de memoria del depósito de trazas de ETB **170**. Al leer los contenidos del depósito de trazas de ETB **170** a través de la interfaz JTAG **84**, el puntero de lectura indica la ubicación para leer. El depósito de trazas de ETB **170** también proporciona un puntero de escritura como un índice en la matriz de memoria del depósito de trazas de ETB **170**. Cuando los datos de trazas se escriben en el depósito de trazas de ETB **170**, se escriben en la entrada indicada por el puntero de escritura. Cada una de las operaciones de escritura aumenta automáticamente el puntero de escritura a la siguiente ubicación después de que se produce la escritura. El depósito de trazas de ETB **170** solo puede capturar una pequeña ventana del flujo de trazas de ETM. La ETB busca un paquete de activación de la ETM para determinar cuándo capturar los datos, y el contador de disparo se usa para especificar la división entre los datos previos al disparo y los datos posteriores al disparo capturados por el depósito de trazas de ETB **240**.

[0041] La ETM **162**, por lo tanto, ayuda a un programador a depurar el código para el DSP **40**. La ETM **162** genera trazas de instrucciones que son un registro del flujo de ejecución de un subproceso en una ventana dada en el tiempo. Usando una traza de instrucciones registradas, el programador puede ver una vista detallada del comportamiento del tiempo de ejecución de su código. Por ejemplo, si el programa del usuario genera una excepción inexplicada, la ETM **162** ayuda a determinar el flujo de instrucciones previas a la excepción, permitiendo de este modo al usuario evaluar exactamente lo que sucedió. La ETM **162** usa un protocolo particular basado en paquetes para representar de manera eficiente el flujo del programa y para minimizar la generación de datos de trazas.

[0042] Un aspecto de la ETM **162** incluye un proceso de secuenciador para el encadenamiento de eventos y para escenarios de detección de eventos más complejos. Para ilustrar el funcionamiento del proceso de secuenciador y de los circuitos de bloque de disparo asociados **180**, la **FIGURA 7** presenta el diagrama de flujo de secuenciador **190**. En respuesta a las entradas de circuito de bloque de disparo **182: 188**, el proceso de secuenciador **180** funciona en el ejemplo mostrado en tres estados, **S0: S2**. Para el funcionamiento del proceso de secuenciador **310**, desde el estado **S0**, el flujo del proceso puede ir a **S1** o a **S2**. Desde **S1**, la secuencia avanza hacia adelante a **S2** o hacia atrás a **S0**. Desde **S2**, la secuencia avanza a **S1** o a **S0**.

[0043] El proceso de secuenciador **180**, por lo tanto, incluye tres estados **S0: S2**, con las transiciones entre estados que son programables y basándose en coincidencias a partir de circuitos de bloque de disparo **182:188**. El proceso de secuenciador **180** tiene uso en el filtrado de trazas para permitir que cada circuito de bloque de disparo **182: 188** esté condicionado en habilitaciones de estado. Esto permite que el rastreo esté restringido en ciertos estados. Al entrar a un nuevo estado, el contador en cada uno de los circuitos de bloque de disparo **182: 188** puede volver a cargar a un valor inicial. Tras entrar en un estado dado, se puede activar un punto de interrupción de ISDB. Al entrar en un estado dado, se puede insertar un marcador de disparo en el flujo de trazas. Al entrar en un estado dado, también se puede activar un control de disparo externo. El disparo externo puede permanecer activo siempre que el secuenciador esté en el estado dado. Al entrar en un estado dado, puede activarse una interrupción al DSP **40**. Después del reinicio, el contador se inicializa en **S0**. Si se activan múltiples transiciones al mismo tiempo, el secuenciador permanece en el estado actual.

[0044] En un ejemplo, la ETM **162** contiene seis contadores que pueden registrar diversos eventos relacionados con el rendimiento del DSP **40**. El funcionamiento básico hace uso de cada contador como fuente programable. Un contador de región programable por el usuario divide la ejecución en ventanas de un número fijo de ciclos. Durante la ventana, los eventos se acumulan en contadores. Al final de la ventana, los valores del contador se forman en paquetes y se envían a través del puerto de trazas. Los contadores se reinician entonces y el proceso comienza de nuevo. Cuando la unidad de creación de perfiles se maneja al mismo tiempo que el rastreo del flujo del programa, esto da como resultado una superposición del rastreo del flujo del programa con información detallada sobre los eventos de rendimiento. Además, la unidad de creación de perfiles contiene una máscara de habilitación de estado para limitar cuándo la unidad está activa.

[0045] Un contador de región se usa para dividir la ejecución en ventanas de un número fijo de ciclos. El tamaño de la región está determinado por un registro programable por el usuario. El contador de región se inicializa en el valor especificado por el usuario y todos los contadores de eventos de creación de perfiles se reinician. El contador de región comienza a contar hacia abajo. Cuando el contador de región llega a cero, el valor para cada uno de los recuentos de eventos de creación de perfiles se emite en el flujo de trazas. El proceso entonces comienza de nuevo. El contador de región solo cuenta cuando la habilitación de estado coincide. Cuando la creación de perfiles está inactiva, el contador regional mantiene su valor y se reanuda cuando se vuelve a entrar en un estado habilitado.

[0046] Diferentes eventos para los que un contador de creación de perfiles puede acumular información incluyen (a) fallos de memoria caché d; (b) ciclos de bloqueo de memoria caché d; (c) fallos de memoria caché i; (d) ciclos de bloqueo de memoria caché i; (e) fallos de ITLB y DTLB; y ciclos de pérdida total (f). Además, cada contador de creación de perfiles contiene una máscara de seis subprocesos para restringir el contador a los eventos que se produzcan en ciertos subprocesos de hardware.

[0047] Al igual que con el contador de región, los contadores de creación de perfiles solamente están activos cuando la máscara de habilitación de estado coincide con el estado actual. Durante el resto del tiempo, los recuentos mantienen sus valores y el recuento se reanuda cuando se vuelve a entrar en un estado habilitado.

[0048] La presente divulgación proporciona un conjunto de protocolo de paquete novedoso de funciones para dar soporte a la alineación de trazas entre subprocesos. Es decir, para cualquier subproceso que se rastree usando la ETM **162**, en cualquier punto arbitrario en el tiempo, la presente divulgación permite identificar para todos los demás subprocesos que estén rastreando el valor de contador de programa y el estado de instrucción (por ejemplo, bloqueado, ejecutado, etc.) de esos subprocesos en ese punto. Por tanto, cuando la ETM **162** funciona en un modo de exactitud de ciclo, la alineación de trazas usa un protocolo de puerto de trazas que da soporte a la alineación de trazas entre subprocesos. La presente divulgación, por lo tanto, hace los campos de recuento de ciclos para los paquetes de reinicio isincrónico relacionados con el subproceso, así como el campo de recuento de ciclos totalmente precisos para la alineación entre subprocesos. La disposición de la divulgación genera un paquete periódico isincrónico en el mismo ciclo de subprocesos para todos los subprocesos. Esto además permite la realineación, en caso de que la alineación del subproceso se pierda por cualquier motivo.

[0049] En un modo de realización de la presente divulgación se proporciona la ejecución de múltiples secuencias de instrucciones que se producen al mismo tiempo y durante la cual el DSP **40** puede verse como varios procesadores de un solo subproceso que funcionan independientemente. En la ETM **162**, el flujo del programa está dividido en una secuencia de paquetes y que incluyen campos de número (tnum) de subprocesos a ciertos paquetes para identificar qué paquete pertenece a cada subproceso. Para identificar las relaciones de temporización entre los subprocesos, la presente divulgación establece y mantiene relaciones de temporización entre subprocesos diferentes. Como cada subproceso puede habilitar y deshabilitar el rastreo de instrucciones de manera independiente, cuando un subproceso activa el rastreo de instrucciones, es posible que otros subprocesos ya hayan habilitado su rastreo durante un tiempo. El procedimiento y el sistema divulgados en la actualidad establecen la temporización relativa de las secuencias de ejecución de subprocesos al marcar el desplazamiento entre cuando un subproceso activa el rastreo y cuando otros subprocesos se han activado y entonces activan el rastreo.

[0050] La presente divulgación, por lo tanto, incluye un campo de desplazamiento de ciclo de subproceso para indicar el número de ciclos desde el último rastreo activado de subproceso. Asimismo, si no hay otros subprocesos

activos, el campo de desplazamiento de ciclo contiene el número de ciclos desde el apagado del rastreo más reciente entre todos los subprocesos. Después de una sesión de trazas, los paquetes permiten reconstruir la ejecución del subproceso. Entonces, usando los campos de desplazamiento, las secuencias de ejecución se pueden alinear correctamente entre los subprocesos. Además, un mecanismo de alineación de instrucciones permite restablecer el rastreo de la ejecución en caso de pérdida de datos.

[0051] La materia objeto divulgada permite además el restablecimiento de las relaciones de temporización entre subprocesos. Al mantener un contador global que permite generar periódicamente paquetes de sincronización, cuando el contador llega a cero, se puede generar un paquete para cada subproceso. Dicho paquete contiene el valor del contador del programa actual para ese subproceso. Ahora, debido a las diversas limitaciones que pueden existir con el DSP **40** en relación con la generación simultánea de paquetes para todos los subprocesos, la presente divulgación incluye un campo de desplazamiento de ciclo para un paquete de sincronización. El campo de desplazamiento de ciclo indica el número de ciclos desde que el último subproceso (entre los otros subprocesos) generó un paquete de sincronización. El campo de desplazamiento de ciclo tiene un tamaño limitado y, si el recuento se satura, la sincronización no podrá lograrse y puede esperar al contador. Al usar estos mecanismos, que se describen con más detalle a continuación, la presente divulgación permite mantener relaciones de temporización de ejecución entre subprocesos en cualquier punto dado para un subproceso. El resultado es la capacidad de observar operaciones y estados de todos los subprocesos para la depuración y para otros propósitos importantes.

Con este entendimiento, la **FIGURA 8** ilustra conceptos importantes del proceso de alineamiento de trazas entre subprocesos divulgado en la actualidad. Con referencia a la **FIGURA 8**, el proceso de alineación entre subprocesos **200** se basa en el tiempo $t=0$, como indica la línea de tiempo **202**. Para los diversos subprocesos **204, 206, 208, 210** y **212** que pueden estar funcionando en el DSP **40**, la iniciación del subproceso puede producirse en diferentes puntos en el tiempo. Por tanto, en el momento $t=t_0$, el subproceso **204** puede iniciar operaciones. Entonces, en un momento posterior, $t=t_1$, el subproceso **206** puede iniciar operaciones. En el momento, $t=t_2$, comienza el subproceso **208**, en $t=t_3$, el subproceso **210** comienza a funcionar, y, en $t=t_4$, el subproceso **212** inicia las operaciones. Como se describirá con más detalle a continuación, la duración **214** representa el número de ciclos hasta que el rastreo se activa para el subproceso **212**. La duración **214** representa el número de ciclos para el contador de programa de inicio de trazas (PC) del paquete de reinicio isincrónico. Para todos los subprocesos interconectados **204:212**, se genera un paquete periódico isincrónico para todos los subprocesos al mismo tiempo. Además, la duración **220** se refiere a la situación de un bloqueo prolongado que se produce, por ejemplo, en el subproceso **212**.

Para describir con más detalle las operaciones entre subprocesos de la presente divulgación, la **FIGURA 9** muestra una vista general funcional de un ejemplo de una unidad de generación de paquetes **230**. La unidad de generación de paquetes **230** incluye una circuitería de control de generación de paquetes **232**, que recibió entradas de atomW, atomN y atomE para generar las salidas de gensyncP, genAsync, genBranch, genPheader, gensyncR, genProf y genCCCount al multiplexador de entrada FIFO **232**. El circuito de generación de encabezado **236** recibe las entradas de recuento w, de recuento e y de recuento n desde el circuito de control de generación de paquetes **232** y la entrada de atomE, todo para generar la salida del encabezado PK al multiplexador de entrada FIFO **234**. La entrada de PC/tld/asid/tnum fluye a funciones de generación de reinicio isincrónico/generación de ciclos **238**, a funciones de generación periódica isincrónica **240** y a funciones de generación de dirección de bifurcación **242**. Las funciones de generación de reinicio isincrónica/recuento de ciclos **238** proporcionan salidas isyncRPK, isyncRLen, CCPK y CCLen al multiplexador de entrada FIFO **234**. Las funciones de generación periódica isincrónica **240** proporcionan las entradas isyncPPK e isyncPLen en el multiplexador de entrada FIFO **234**. Las funciones de generación de dirección de bifurcación **242**, que también reciben entradas de tipo bifurcación, proporcionan salidas branchPK y branchLen al multiplexador de entrada FIFO **234**. Además, las funciones de generaciones de perfiles **244** reciben entradas de identificador de perfil para generar salidas profPK y profLen al multiplexador de entrada FIFO **234**.

[0052] El multiplexador de entrada FIFO **234** toma todos los paquetes que necesitan generarse en un ciclo dado y los forma en un fragmento contiguo. El multiplexador de entrada FIFO **234** debe tener en cuenta los paquetes de tamaño variable al hacer su concatenación. La salida del multiplexador de entrada FIFO **234** se registra antes de enviarse al FIFO.

[0053] Para la generación de paquetes de la presente divulgación y el multiplexador de entrada FIFO **234**, las operaciones se producen a través de tres etapas. En la primera etapa de la generación de paquetes, funcionan las funciones de control de generación de paquetes **232** y los motores de generación de paquetes individuales **236** a **244**, así como el incremento del recuento w, del recuento e y del recuento n de los contadores de átomos. Las operaciones del multiplexador de entrada de FIFO **234** incluyen la fusión de todos los paquetes generados en el ciclo en un bloque contiguo en esta primera etapa. En la segunda etapa, la escritura FIFO se produce para incluir datos rotativos para alinearse con el puntero de escritura FIFO, calcular y rotar las habilitaciones de escritura y escribir datos en registros del multiplexador FIFO **234**. En la tercera etapa, la etapa de lectura FIFO, los datos pueden leerse desde los registros FIFO. En esta tercera etapa, también se produce el caso especial de insertar el paquete de disparo y enviar datos a la ETB.

[0054] Como se ve en la **FIGURA 9**, cada una de las unidades de generación de paquetes individuales empaqueta sus respectivos datos y entonces envía el paquete y la longitud resultantes al multiplexador de entrada FIFO **234**

para la concatenación. Además, el circuito de control de generación de paquetes **232** mantiene los contadores de átomos y decide qué paquetes generar en un ciclo particular. La unidad de generación de paquetes **230** mantiene tres contadores para registrar el número de átomos pendientes que pueden enviarse en el futuro encabezado o paquete de recuento de ciclos. El contador de átomo E aumenta cada vez que se encuentra un átomo E. El contador de átomo N aumenta cada vez que se encuentra un átomo N y el recuento incluye el átomo actual. Es decir, si el átomo actual es un N, se incluirá en el recuento de átomos N. El contador de átomos W aumenta cada vez que se encuentra un átomo W. Estos contadores son contadores por subproceso y, como tal, hay seis copias de cada uno en el presente ejemplo. Los contadores se reinician siempre que los recuentos se envían a través de un encabezado o paquete de recuento de ciclos.

A partir de las funciones de generación de dirección de bifurcación **242**, se puede generar un paquete de direcciones de bifurcación para un destino de un bucle de retorno de bifurcación indirecto, si el destino es diferente del bucle de retorno previo. Si se establece el modo de todo bucles de retorno, el paquete se envía para cada bucle de retorno. También un nuevo paquete de bucle de retorno se expulsa después de que se genera un paquete de reinicio isincrónico y periódico isincrónico. El objetivo de un modo de bifurcaciones directas de IF de bifurcación relacionado con el PC se establece entonces. El objetivo de un evento (interrupción, excepción, SWI, TRAP, etc.) devuelve la instrucción de RTE. El paquete de dirección de bifurcación se puede generar cuando se active la habilitación de trazas y no sea el primer ciclo de rastreo (en su lugar, se puede generar un paquete de reinicio isincrónico). La unidad de control de generación de paquetes mantiene un registro de objetivo de bucle de retorno previo para que cada subproceso determine si se necesita un paquete de dirección de bifurcación de bucle de retorno.

La **FIGURA 10** muestra la tabla **250** de definiciones para los diversos átomos aplicables a las instrucciones de la materia objeto divulgada en la actualidad. Un objetivo principal del protocolo de paquetes es admitir la alineación de trazas entre subprocesos. Es decir, para cualquier subproceso que se rastree, en cualquier punto arbitrario en el tiempo, el usuario debería ser capaz de identificar para todos los demás subprocesos el valor de contador de programa y el estado de instrucción (bloqueado, ejecutado) de esos subprocesos en ese punto. La alineación de trazas funciona cuando la ETM **162** funciona en modo de exactitud de ciclo. El protocolo de puerto de trazas se ha ampliado para admitir la alineación de trazas entre subprocesos. Un átomo de instrucción de tipo 'S' proporciona un mecanismo de bifurcación conocido como salto doble. Aquí, la ETM **162** incluye un tipo S, que se define usando un tipo de paquete de formato 4 de encabezado para sostener el átomo.

La **FIGURA 11** proporciona un paquete de dirección de bifurcación **280** para la presente divulgación que da soporte a un campo TID de **32** bits. El paquete de dirección de bifurcación divulgado en la actualidad da soporte a un campo TID de **32** bits, cuyos contenidos aparecen en la **FIGURA 11**. El paquete de dirección de bifurcación es de longitud variable, incluyendo entre 1 y 11 bytes. Los bits de continuación en los MSB de los bytes 0 a 4 indican si el paquete continúa más allá de ese punto. Por separado, el bit T indica si se agregará un campo de tipo al final del paquete. Un 1 en el MSB de byte-4 indica que seguirán 5 bytes (4 bytes del TID y 1 byte de ASID).

[0055] El paquete de dirección de bifurcación es variable con el PC, el TID/ASID y el tipo de compresión. El TID/ASID puede enviarse si alguno ha cambiado desde el último paquete de dirección de bifurcación o isincrónico (reinicio o periódico) para el tnum. El tipo puede enviarse si ha cambiado desde que se envió el último tipo. Se puede generar en el primer paquete de dirección de bifurcación después de un reinicio isincrónico o periódico isincrónico. El PC se comprime en relación con el PC previo enviado por una dirección de bifurcación, un reinicio isincrónico o un periódico isincrónico. La parte inferior del PC que haya cambiado puede enviarse. Para cada subproceso, el PC previo, el TID/ASID anterior y registros de tipo previos se usan para determinar la compresión adecuada.

[0056] La **FIGURA 12** presenta los contenidos a modo de ejemplo de un paquete de reinicio isincrónico **260** para su uso en el proceso divulgado en la actualidad. El paquete de reinicio isincrónico puede generarse cuando se inicie el rastreo (la traza habilitada era previamente baja). Además, al salir de una condición de desbordamiento, si el rastreo habilitado sigue siendo alto, se genera un paquete de reinicio isincrónico con el campo de razón que indica que hubo un desbordamiento. La presente divulgación proporciona alineación usando paquetes de reinicio isincrónico. La alineación de instrucciones se puede ver como la asignación de un ciclo de subprocesos global a cada ciclo de cada subproceso que se rastree. Un procedimiento de decompresor logra esto para el caso en el que los paquetes de reinicio isincrónico están disponibles en el flujo de trazas. El decompresor mantendrá un valor de contador del último recuento de reinicio isincrónico.

[0057] En cualquier momento se detecta un paquete de reinicio isincrónico, la presente divulgación puede inspeccionar el campo de tipo de ciclo y el campo de tipo de recuento de ciclos. Si el campo de tipo es 'global', el proceso asigna un último recuento de reinicio isincrónico al valor de campo de recuento de ciclos y anota ese paquete con el último recuento de reinicio isincrónico. Si el campo de tipo se 'desplaza', el proceso puede incrementar el último recuento de reinicio isincrónico por el valor en el campo de recuento de ciclos y anotar además el paquete con el último recuento de reinicio isincrónico. En ese momento, cada paquete de reinicio isincrónico se anotará con un valor de ciclo de subprocesos global.

[0058] La siguiente etapa es anotar los átomos. El proceso divide el flujo de trazas en los flujos de paquetes locales de subprocesos en base a los valores de tnum en el flujo de paquetes. Para cada flujo de paquete local de subprocesos, el proceso mantiene un ciclo de subprocesos global variable. Este valor se mantiene de manera que, cada vez que se encuentre un paquete de reinicio isincrónico, se asigna un ciclo de subproceso global al valor anotado. Cada vez que se encuentre un átomo de instrucción (E, N, S o W), el proceso puede anotar ese átomo con ciclo de subprocesos global e incrementar el ciclo de subprocesos global en 1. Usando el procedimiento de descompresión normal, el proceso determina los valores del contador del programa para cada átomo. Después de aplicar este procedimiento, cada ciclo de cada subproceso que se rastree puede anotarse con un valor de recuento global de ciclos de subprocesos. Como resultado, por ejemplo, para un ciclo de subproceso dado, se puede determinar el valor de contador de programa actual para todos los seis subprocesos, así como el estado de la tubería para esos subprocesos (bloqueado, ejecutado, etc.)

Para la versión no precisa del ciclo, el campo TID se expande a 32 bits. Para la versión de exactitud de ciclo, el TID se amplía a 32 bits y el nuevo campo de recuento de ciclos se define para dar soporte a la alineación de trazas. Un recuento global de ciclos indica el número de ciclos de subprocesos desde el último reinicio del hardware. Un recuento de ciclos de desplazamiento indica el número de ciclos desde que se generó el último paquete de reinicio isincrónico.

[0059] La FIGURA 13 representa un ejemplo de un paquete periódico isincrónico 290 para dar soporte a la materia objeto divulgada en la actualidad. Los paquetes periódicos isincrónicos incluyen un campo de 'desplazamiento' que indica el número de ciclos de subprocesos desde que se generó el último paquete periódico isincrónico (de entre todos los subprocesos). El proceso divulgado también admite la alineación usando paquetes periódicos isincrónicos. En algunos casos, solo hay disponible una ventana limitada de datos de trazas. Por lo tanto, un paquete de reinicio isincrónico puede no estar disponible. En estos casos, el paquete periódico isincrónico se puede usar para determinar la alineación entre subprocesos. Si se desconocen los límites del paquete, el proceso puede buscar un paquete asincrónico para recuperar la alineación del paquete. El decompresor puede mantener un último recuento periódico isincrónico de contador. Al pasar por el flujo de paquetes de subprocesos múltiples, el proceso puede incluir la búsqueda del paquete periódico isincrónico más antiguo que no tenga un campo de desplazamiento. Entonces, el proceso puede asignar el último recuento periódico isincrónico a 0 y anotar ese paquete con el valor. Para cada uno de los siguientes paquetes periódicos isincrónicos de los otros tnums, el proceso puede incrementar el recuento periódico isincrónico por el valor del campo de desplazamiento y cada uno de los paquetes periódicos isincrónicos con el recuento periódico isincrónico. Entonces, el proceso puede dividir los subprocesos en flujos de paquetes individuales y descomprimirlos por separado.

[0060] Los contadores periódicos isincrónicos de ETM pueden disminuir cada vez que se envíen bytes paquetes al FIFO. Cuando el contador llegue a cero para un subproceso, un paquete periódico isincrónico se puede marcar como pendiente. En la siguiente oportunidad, se genera el paquete periódico isincrónico. Un periódico isincrónico puede retenerse si se genera alguno de los siguientes paquetes (reinicio isincrónico, disparo, dirección de bifurcación, asincrónico, perfil). Cada subproceso puede mantener su propio contador periódico isincrónico. Los contadores isincrónicos pueden reiniciarse siempre que se genere un paquete de reinicio isincrónico para ese subproceso.

[0061] La FIGURA 14 presenta un conjunto de paquetes de recuento de ciclos para su uso en el proceso divulgado en la actualidad. Se puede usar un paquete de recuento de ciclos para acumular átomos W. Además, el paquete de recuento de ciclos solo puede generarse en modo de exactitud de ciclo. El paquete de recuento de ciclos puede generarse en los siguientes casos: conmutar de W a E o de W a N y el recuento de átomos de W no cabe en la capacidad de la pseudocelda de formato 3, la rotura de átomos con átomos W pendientes y demasiados átomos W para encajar en un encabezado de formato 3. Los átomos W pueden colocarse en el paquete de recuento de ciclos y el átomo E o N puede salir en un encabezado de formato 1, y el paquete de recuento de ciclos puede alcanzar su máxima exactitud completa. En lugar de permitir que el paquete de recuento de ciclos se incremente hasta llegar a su valor máximo de 32 bits, la ETM envía múltiples valores de recuento de ciclos más pequeños en el límite entre la precisión total y la precisión limitada. En este caso, el átomo actual puede ser un átomo W. Debido a que este átomo no está incluido en el paquete, el recuento se reinicia en 1 en lugar de 0.

El recuento de ciclos en el paquete de reinicio isincrónico se ha redefinido como dos tipos posibles: un recuento 'global' de 64 bits al que se hace referencia desde el último reinicio de hardware y un recuento de desplazamiento de 16 bits que indica el número de ciclos de subprocesos desde el último paquete de reinicio isincrónico (de entre todos los subprocesos).

Además de los paquetes de recuento de ciclos de dos bytes 300 y de tres bytes 310, la presente divulgación proporciona un paquete de tres bytes 320 para ajustar un valor de 32 bits en la carga útil de 16 bits. En la presente materia objeto, se define un paquete de cinco bytes, que incluye un encabezado de un byte y una carga útil de cuatro bytes, como se muestra en la FIGURA 14. Para cada subproceso, el proceso divulgado mantiene un recuento global de ciclos de subprocesos. Para el primer paquete periódico isincrónico que se encuentra, el proceso establece un ciclo de subproceso global en el valor anotado de ese paquete y anota el átomo anterior con ese valor. El ciclo de subproceso global se incrementa en 1. Cada vez que se encuentra un átomo (E, N, S, W), el proceso

anota ese átomo con el ciclo de subproceso global e incrementa el recuento global de subprocesos en 1. El proceso puede descomprimir el flujo como sea normal para asignar valores de contador de programa a cada átomo. En este punto, como antes, cada ciclo de cada subproceso que se rastree tendrá un ciclo de subprocesos global asociado.

5 **[0062]** La **FIGURA 15** muestra un ejemplo de un paquete de formato 4 de encabezado **330** como sea relevante para la presente divulgación. Los paquetes de encabezado se pueden generar en diferentes casos. Por ejemplo, cuando no se pueden retener más átomos en el paquete y cuando se genera un paquete de dirección de bifurcación, el perfil o el paquete periódico isincrónico, se genera un encabezado para marcar la ubicación del paquete respectivo dentro de la transmisión atómica. El paquete de encabezado de formato 1 puede tener tres campos de carga útil atómica:
 10 de 0 a 31 átomos E (campo-0), seguidos de 0 a 3 átomos N (campo-1), seguido de 0 o 1 átomo E (campo-2). Las reglas para generar paquetes de encabezado de formato 1 pueden ser: recuento de átomo E al máximo, y el átomo actual es un E. Los E contados irán al campo-0 y el átomo de E actual puede colocarse en el campo 2. El recuento de átomos N al máximo (el recuento incluye el átomo N actual). Los E contados se pueden colocar en el campo 0 y el recuento de átomos N se puede colocar en el campo 1. El átomo actual es E y el número de átomos N no es cero.
 15 Los recuentos de átomos E y N contados pueden colocarse en el campo 0 y en el campo 1 y los átomos E actuales pueden colocarse en el campo 2. Cuando el átomo actual tiene átomos W y E o N pendientes, los recuentos actuales E y N pueden enviarse en el campo 0 y en el campo 1 de carga útil. Cuando se interrumpe el átomo y no hay átomos W pendientes, los recuentos actuales de E y N pueden colocarse en el campo 0 y en el campo 1 y, si el átomo actual es E, puede colocarse en el campo 2.

20 **[0063]** La **FIGURA 16** representa un paquete asincrónico **340** para su uso con un ejemplo de la materia objeto divulgada. La longitud del paquete asincrónico aloja unos campos de recuento de ciclos más largos y puede, por ejemplo, incluir diez bytes. El contador periódico asincrónico también se puede usar para definir cuándo se generan paquetes asincrónicos. Un paquete asincrónico se marca como pendiente cada vez que llega el contador periódico de ETM. El paquete asincrónico debe salir solo y permanecerá pendiente cada vez que se generen otros paquetes. El contador asincrónico es global y puede disminuirse cada vez que los paquetes de bytes se envíen al FIFO.

En resumen, la materia objeto divulgada proporciona un procedimiento y un sistema para la alineación de trazas entre subprocesos con un procesador de subprocesos múltiples para su uso con el procesamiento de trazas de ejecución. La materia objeto divulgada, por lo tanto, incluye registrar datos de temporización relacionados con un evento predeterminado común. Dicho evento puede ser el número de ciclos desde que un último subproceso inició el rastreo de ejecución o el número de ciclos desde que todos los subprocesos terminaron el rastreo de ejecución. El número de ciclos en los que un subproceso inicia el rastreo de ejecución se referencia al evento común predeterminado para mantener el tiempo del rastreo de ejecución. Los datos relacionados con el evento predeterminado común se pueden actualizar entonces para asociarlos con el momento en que el subproceso inició el rastreo de ejecución. El resultado es permitir alinear los datos de temporización asociados con todos los subprocesos. Los registros interrelacionados permiten reconstruir la información de trazas de ejecución interdependientes para subprocesos que funcionan en el procesador de subprocesos múltiples, así como sincronizar datos de temporización para todos los subprocesos de funcionamiento.

40 Como se ha visto anteriormente, las características y funciones de procesamiento descritas en el presente documento para la alineación de trazas entre subprocesos con un procesador de subprocesos múltiples en un procesador de señales digitales de subprocesos múltiples, que incluye un procesador de subprocesos múltiples entrelazados, se pueden implementar de diversas maneras. Por ejemplo, no solo el DSP **40** puede realizar las operaciones descritas anteriormente, sino que también los ejemplos presentes pueden implementarse en un circuito integrado específico de la aplicación (ASIC), en un microcontrolador, en un procesador de señales digitales o en otros circuitos electrónicos diseñados para realizar las funciones descritas en el presente documento. Además, el proceso y las características aquí descritas se pueden almacenar en medios de registro magnéticos, ópticos o en otros medios de lectura y ejecución mediante dichos diversos sistemas de procesamiento de señales e instrucciones. La descripción anterior de los modos de realización preferentes se proporciona para permitir que cualquier experto en la técnica realice o use la materia objeto reivindicada.

REIVINDICACIONES

1. Un procedimiento para la alineación de temporización de trazas entre subprocesos entre los subprocesos de un procesador de subprocesos múltiples (40) durante un proceso de rastreo de ejecución, que comprende:
- 5 registrar los datos de temporización relacionados con un evento predeterminado común, referenciándose dichos eventos predeterminados comunes por todos los subprocesos de funcionamiento de dicho procesador de subprocesos múltiples (40) durante el rastreo de ejecución del procesador de núcleo (70);
- 10 referenciar un momento en el que un subproceso inicia el rastreo de ejecución a dicho evento predeterminado común para mantener el tiempo del rastreo de ejecución para dicho subproceso con relación a dicho evento predeterminado común; y
- 15 actualizar dichos datos de temporización relacionados con dicho evento predeterminado común para asociarlos con dicho tiempo en el que dicho subproceso inició el rastreo de ejecución, alineando de este modo dicho tiempo en el que dicho subproceso inició el rastreo de ejecución a datos de temporización asociados con todos los otros subprocesos de dicho procesador de subprocesos múltiples (40) para el cual se puede estar produciendo el rastreo de ejecución.
- 20 2. El procedimiento según la reivindicación 1, que comprende además relacionar dicho evento predeterminado común con un momento en el que un último subproceso activó el rastreo de ejecución.
3. El procedimiento según la reivindicación 1, que comprende además relacionar dicho evento predeterminado común con el número de ciclos del procesador de subprocesos múltiples (40) ya que todos los subprocesos desactivaron el rastreo de ejecución.
- 25 4. El procedimiento según la reivindicación 1, que comprende además registrar dicho evento predeterminado común en una pluralidad de paquetes de datos.
- 30 5. El procedimiento según la reivindicación 1, que comprende además reconstruir información de rastreo de ejecución interrelacionadas para subprocesos que funcionen en dicho procesador de subprocesos múltiples (40) usando una pluralidad de paquetes de datos asociados con dicho evento predeterminado común.
- 35 6. El procedimiento según la reivindicación 1, que comprende además sincronizar dichos datos de temporización relacionados con dicho evento predeterminado común.
7. El procedimiento según la reivindicación 1, que comprende además generar un recuento global de ciclos desde la aparición de dicho evento predeterminado común.
- 40 8. El procedimiento según la reivindicación 1, que comprende además restablecer los datos de temporización entre subprocesos entre todos los subprocesos que realizan el rastreo de ejecución usando un recuento global de ciclos desde dicho evento predeterminado común.
- 45 9. El procedimiento según la reivindicación 1, que comprende además determinar una pérdida de datos y, en respuesta a dicha pérdida de datos, restablecer los datos de temporización entre subprocesos entre todos los subprocesos que realizan el rastreo de ejecución usando un recuento global de ciclos desde dicho evento predeterminado común, en el caso de una pérdida de datos.
- 50 10. El procedimiento según la reivindicación 1, que comprende además generar un paquete de sincronización para todos los subprocesos en el caso de un contador global que circule a través de un valor cero.
- 55 11. Un procesador de señales digitales de subprocesos múltiples (40) para su funcionamiento en soporte de un dispositivo electrónico personal, comprendiendo el procesador de señales digitales de subprocesos múltiples medios de depuración para realizar un proceso de rastreo de ejecución y, en relación con esto, alinear el tiempo de trazas entre subprocesos entre los subprocesos de dicho procesador de subprocesos múltiples, que comprende:
- 60 medios para registrar datos de temporización relacionados con un evento predeterminado común, referenciándose dichos eventos predeterminados comunes por todos los subprocesos de funcionamiento de dicho procesador de subprocesos múltiples (40) durante el rastreo de ejecución del procesador de núcleo (70);
- 65 medios para referenciar un momento en el que un subproceso inicia el rastreo de ejecución a dicho evento predeterminado común para mantener la temporización del rastreo de ejecución para dicho subproceso relativo a dicho evento predeterminado común; y

- 5 medios para actualizar dichos datos de temporización relacionados con dicho evento predeterminado común para asociar con dicho tiempo en el que dicho subproceso inició el rastreo de ejecución, alineando de este modo dicho tiempo en el que dicho subproceso inició el rastreo de ejecución a los datos de temporización asociados con todos los otros subprocesos de dicho procesador de subprocesos múltiples (40) para el cual se puede estar produciendo el rastreo de ejecución.
- 10 12. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para relacionar dicho evento predeterminado común con el tiempo en el que un último subproceso activó el rastreo de ejecución.
- 15 13. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para relacionar dicho evento predeterminado común con el número de ciclos del procesador de subprocesos múltiples (40) ya que todos los subprocesos desactivaron el rastreo de la ejecución.
- 20 14. El procesador de señales digitales (40) de la reivindicación 11, que comprende además medios para registrar dicho evento predeterminado común en una pluralidad de paquetes de datos.
- 25 15. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para reconstruir información de rastreo de ejecución interconectada para subprocesos que funcionen en dicho procesador de subprocesos múltiples (40) usando una pluralidad de paquetes de datos asociados con dicho evento predeterminado común.
- 30 16. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para sincronizar dichos datos de temporización relacionados con dicho evento predeterminado común.
- 35 17. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para generar un recuento global de ciclos desde la aparición de dicho evento predeterminado común.
- 40 18. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para restablecer datos de temporización entre subprocesos entre todos los subprocesos que realizan el rastreo de ejecución usando un recuento global de ciclos desde dicho evento predeterminado común.
- 45 19. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para determinar una pérdida de datos y, en respuesta a dicha pérdida de datos, restablecer datos de temporización entre subprocesos entre todos los subprocesos que realicen el rastreo de ejecución usando un recuento global de ciclos desde dicho evento predeterminado común.
- 50 20. El procesador de señales digitales (40) según la reivindicación 11, que comprende además medios para generar un paquete de sincronización para todos los subprocesos en el caso de un contador global que circula a través de un valor cero.
- 55 21. Un medio utilizable por ordenador que tiene un código de programa legible por ordenador incorporado en el mismo para depurar un procesador de señales digitales de subprocesos múltiples (40) que incluye realizar un proceso de rastreo de ejecución y, en relación con esto, alinear el tiempo de trazas entre subprocesos entre los subprocesos de dicho procesador de subprocesos múltiples (40), comprendiendo el medio utilizable por ordenador:
- 60 un código de programa legible por ordenador para registrar datos de temporización relacionados con un evento predeterminado común, siendo referenciable de manera común dicho evento predeterminado común por todos los subprocesos de funcionamiento de dicho procesador de subprocesos múltiples durante el rastreo de ejecución del procesador de núcleo (70);
- 65 medios de código de programa legible por ordenador para referenciar un momento en el que un subproceso inicia el rastreo de ejecución a dicho evento predeterminado común para mantener el tiempo de trazas de ejecución para dicho subproceso con relación a dicho evento predeterminado común; y
- un código de programa legible por ordenador para actualizar dichos datos de temporización relativos a dicho evento predeterminado común para asociar con dicho tiempo en el que dicho subproceso inició el rastreo de ejecución, alineando dicho tiempo en el que dicho subproceso inició el rastreo de ejecución a los datos de temporización asociados con todos los otros subprocesos del procesador de subprocesos múltiples (40) para el cual se puede estar produciendo el rastreo de ejecución.
22. El medio utilizable por ordenador según la reivindicación 21, que comprende además un código de programa legible por ordenador para relacionar dicho evento predeterminado común con el momento en el que un último subproceso activó el rastreo de ejecución.

23. El medio utilizable por ordenador según la reivindicación 21, que comprende además un código de programa legible por ordenador para relacionar dicho evento predeterminado común con el número de ciclos del procesador de subprocesos múltiples (40) ya que todos los subprocesos desactivaron el rastreo de ejecución.
- 5 24. El medio utilizable por ordenador según la reivindicación 21, que comprende además un medio de código de programa legible por ordenador para registrar dicho evento predeterminado común en una pluralidad de paquetes de datos.
- 10 25. El medio utilizable por ordenador según la reivindicación 21, que comprende además un código de programa legible por ordenador para reconstruir información de rastreo de ejecución interconectada para subprocesos que funcionen en dicho procesador de subprocesos múltiples (40) usando una pluralidad de paquetes de datos asociados con dicho evento predeterminado común.

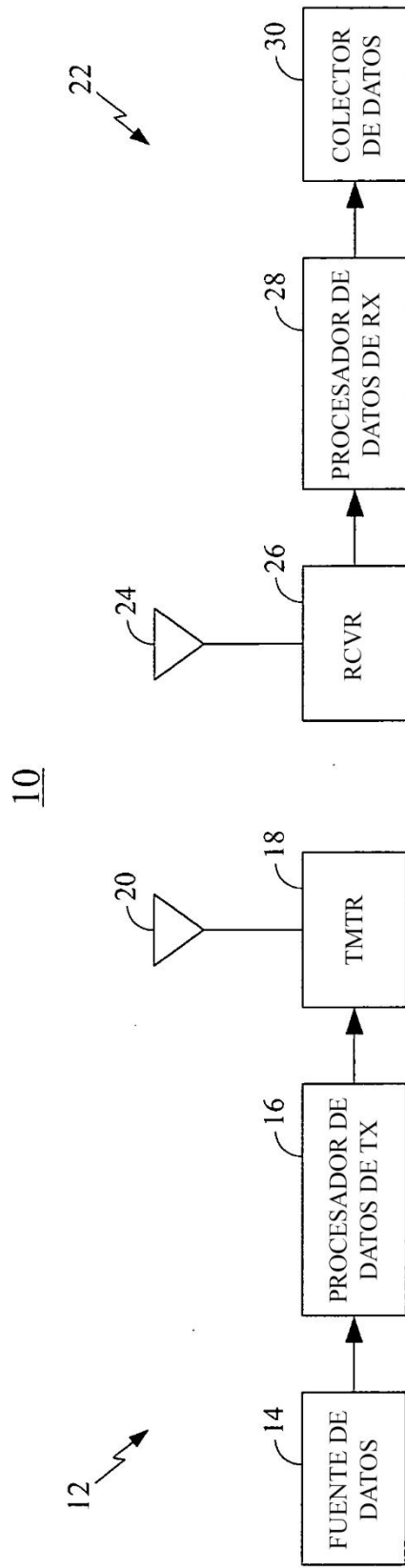


FIG. 1

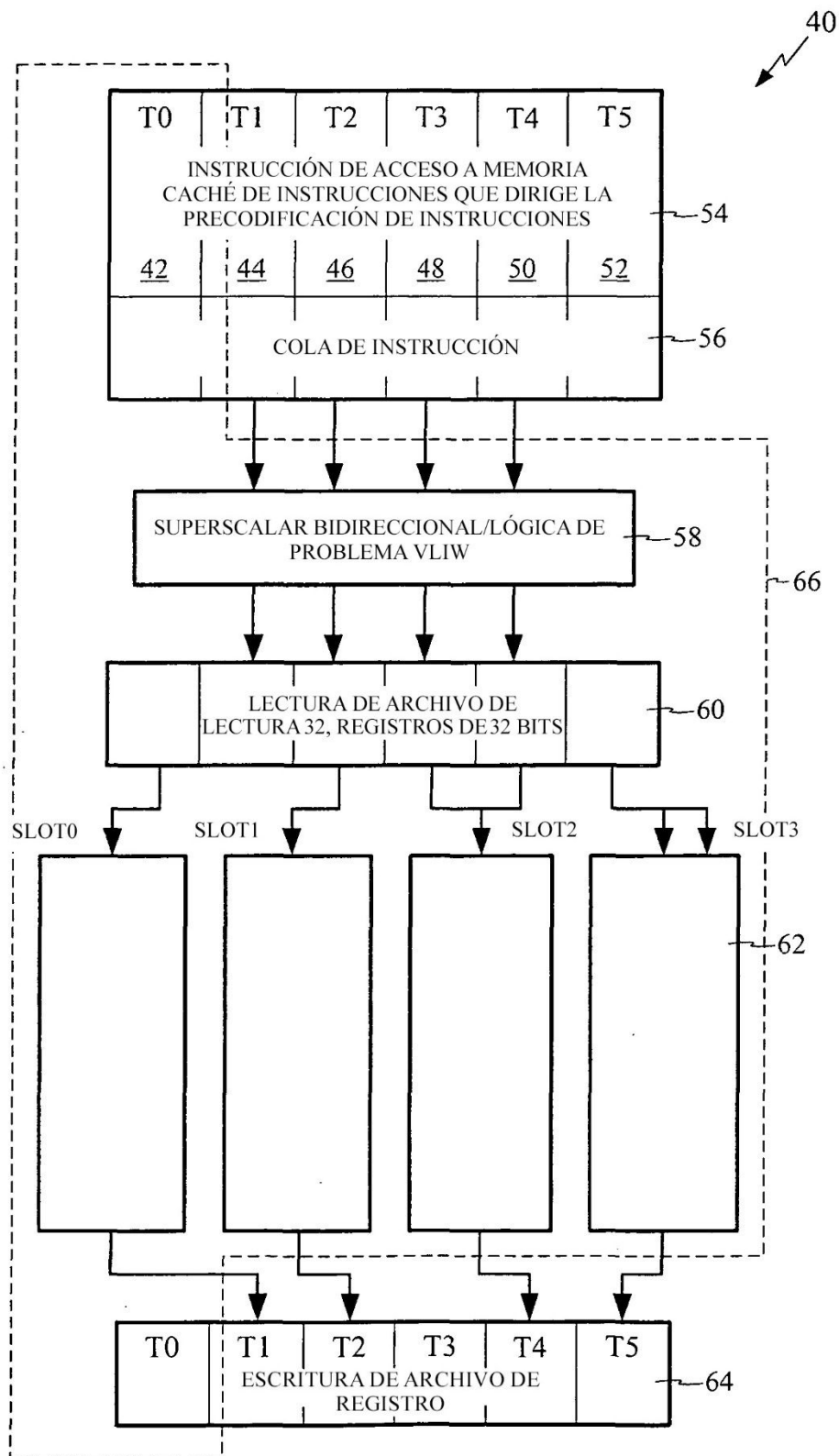


FIG. 2

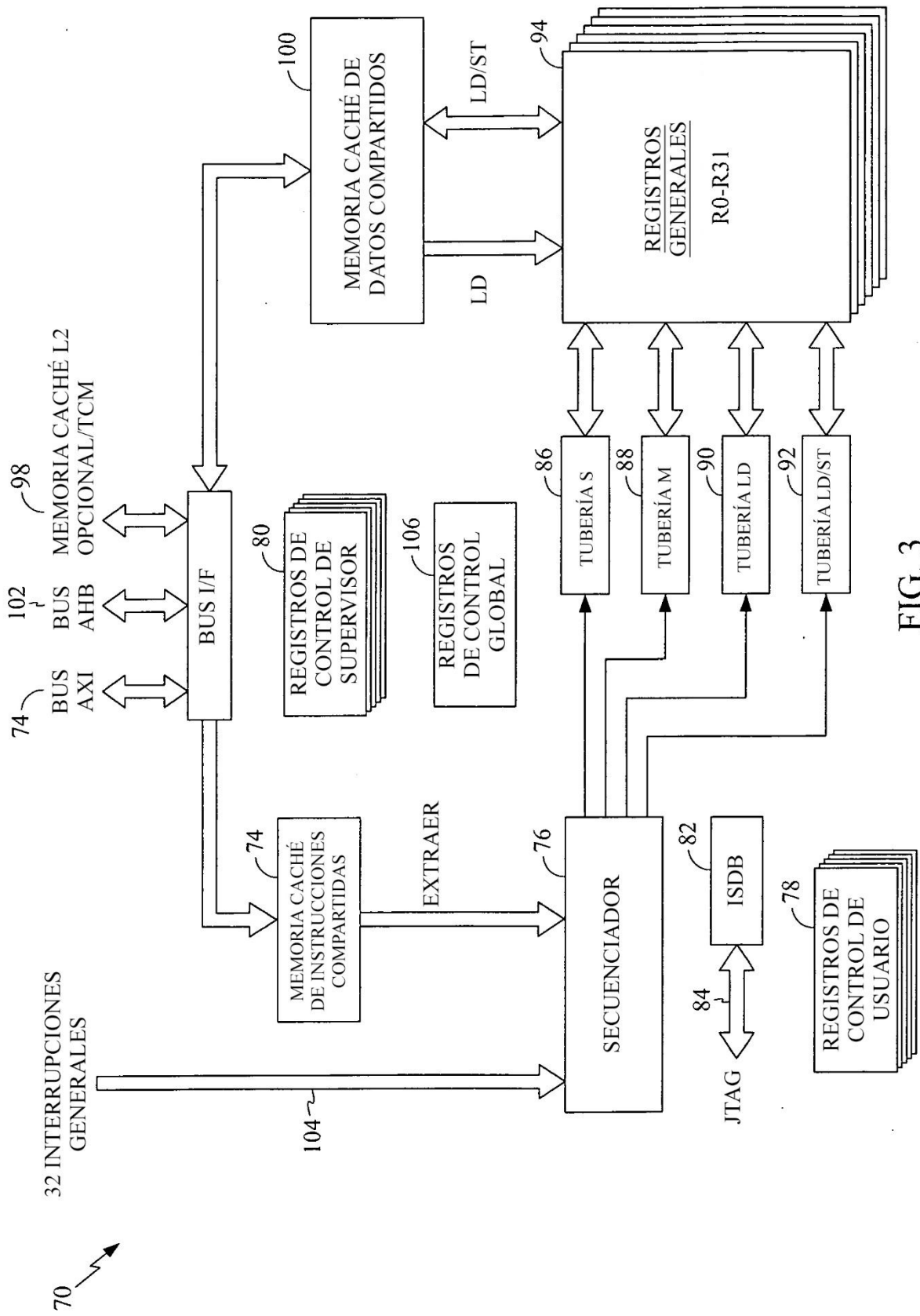


FIG. 3

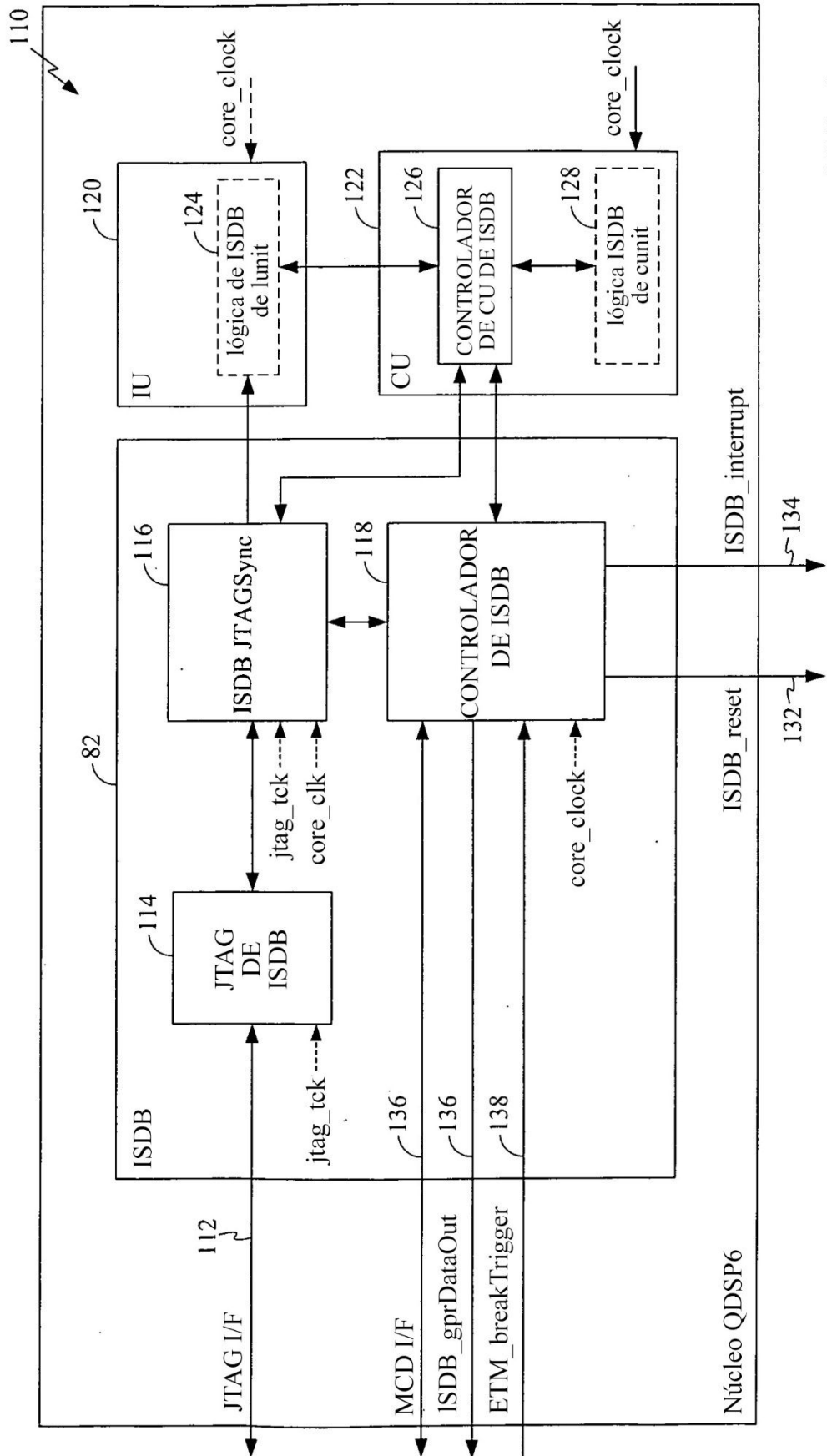


FIG. 4

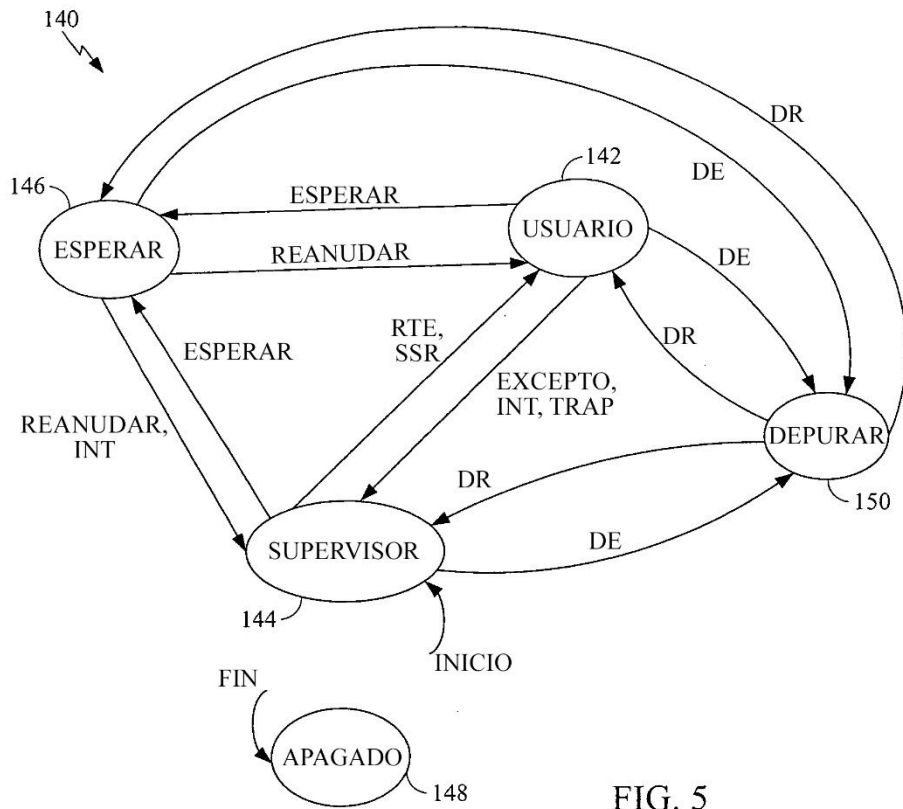


FIG. 5

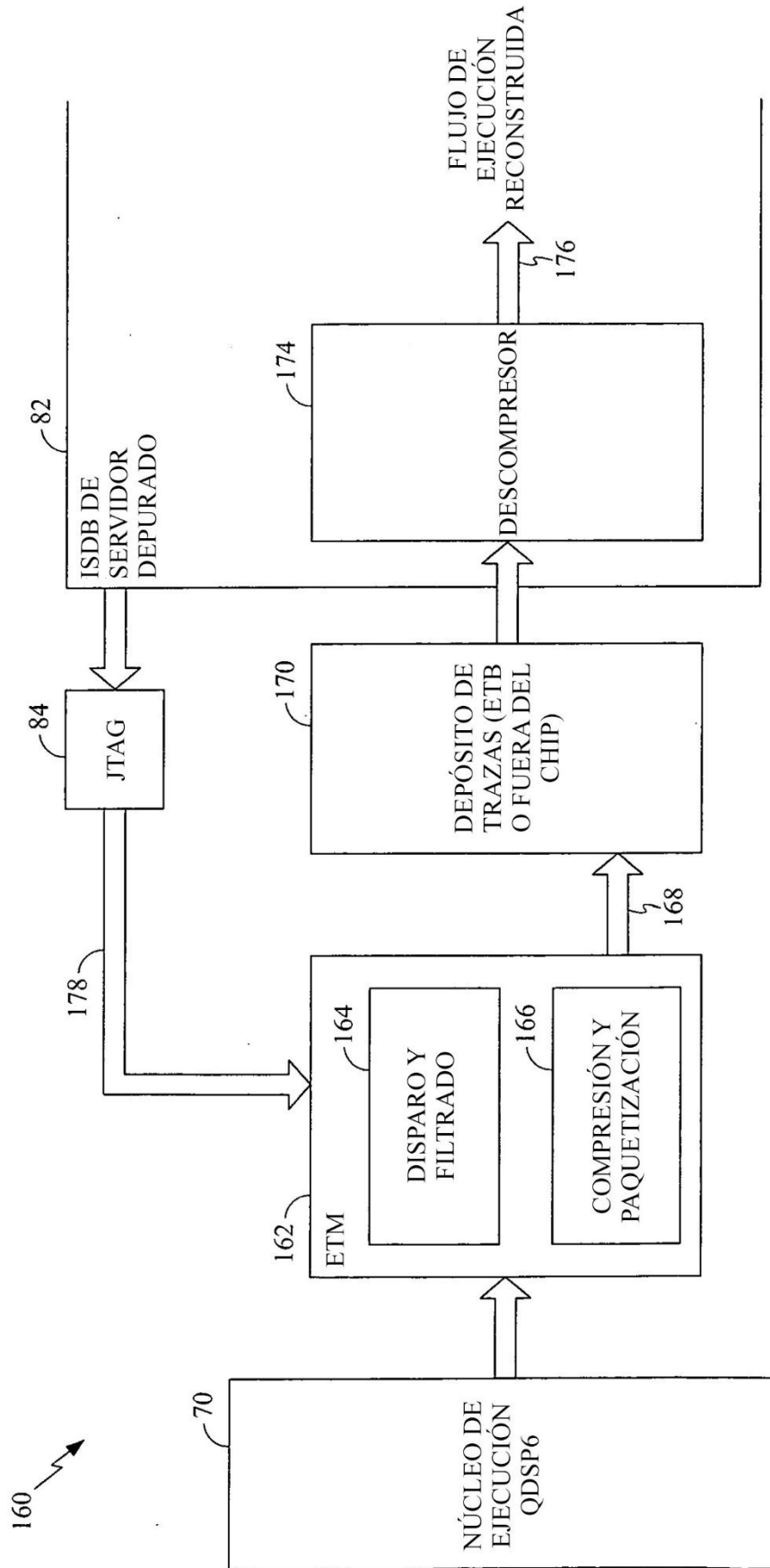


FIG. 6

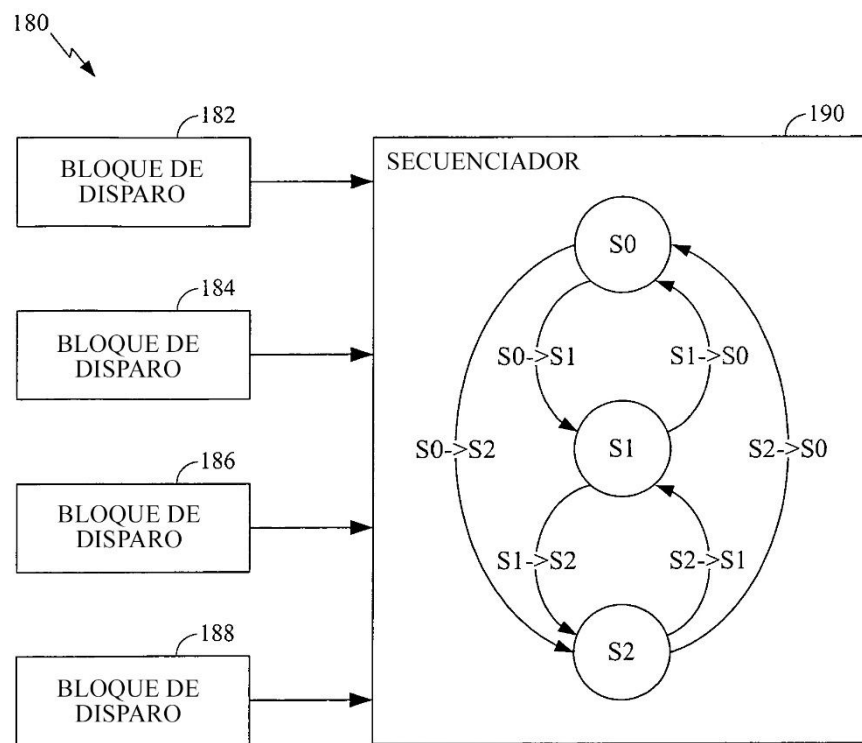


FIG. 7

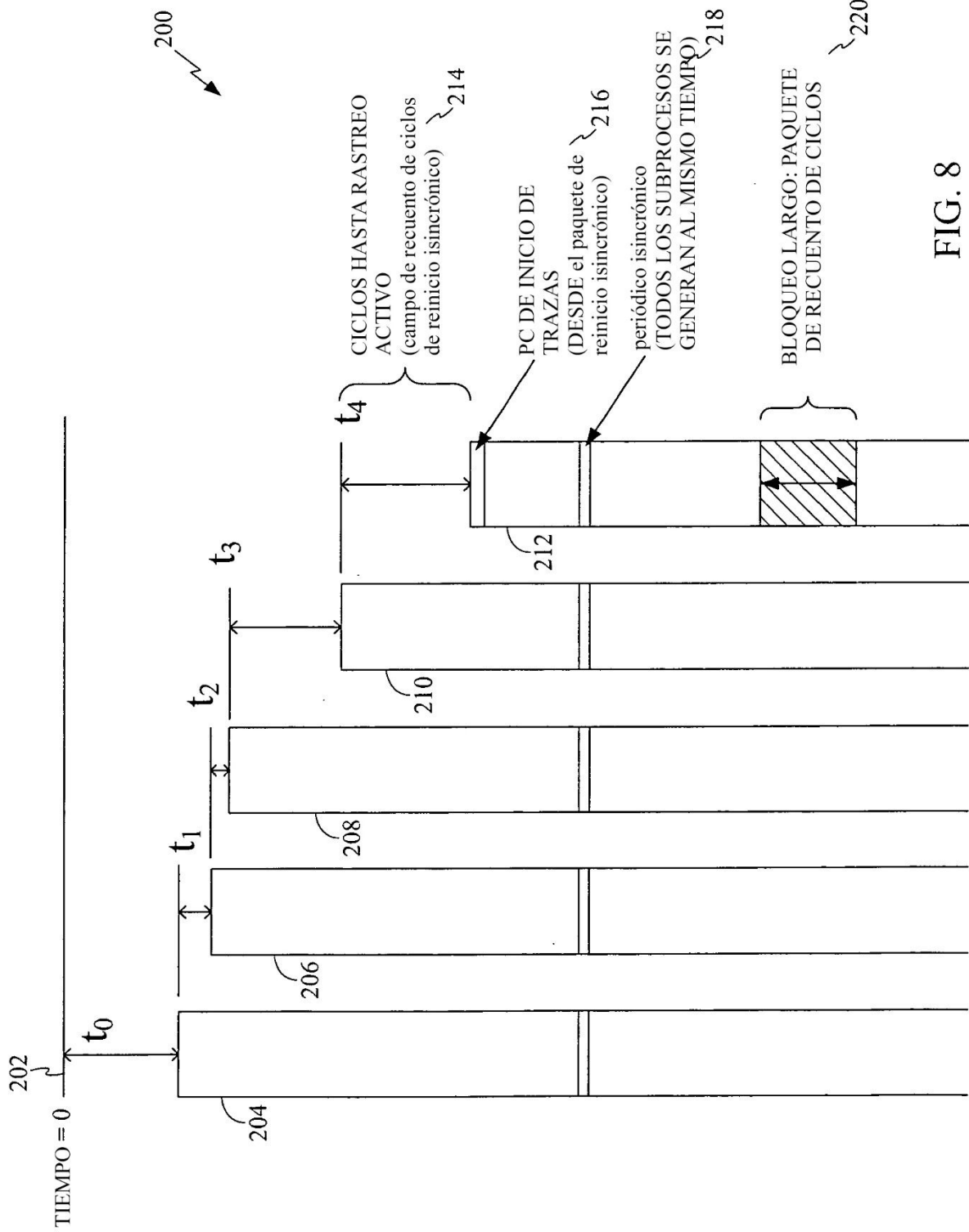


FIG. 8

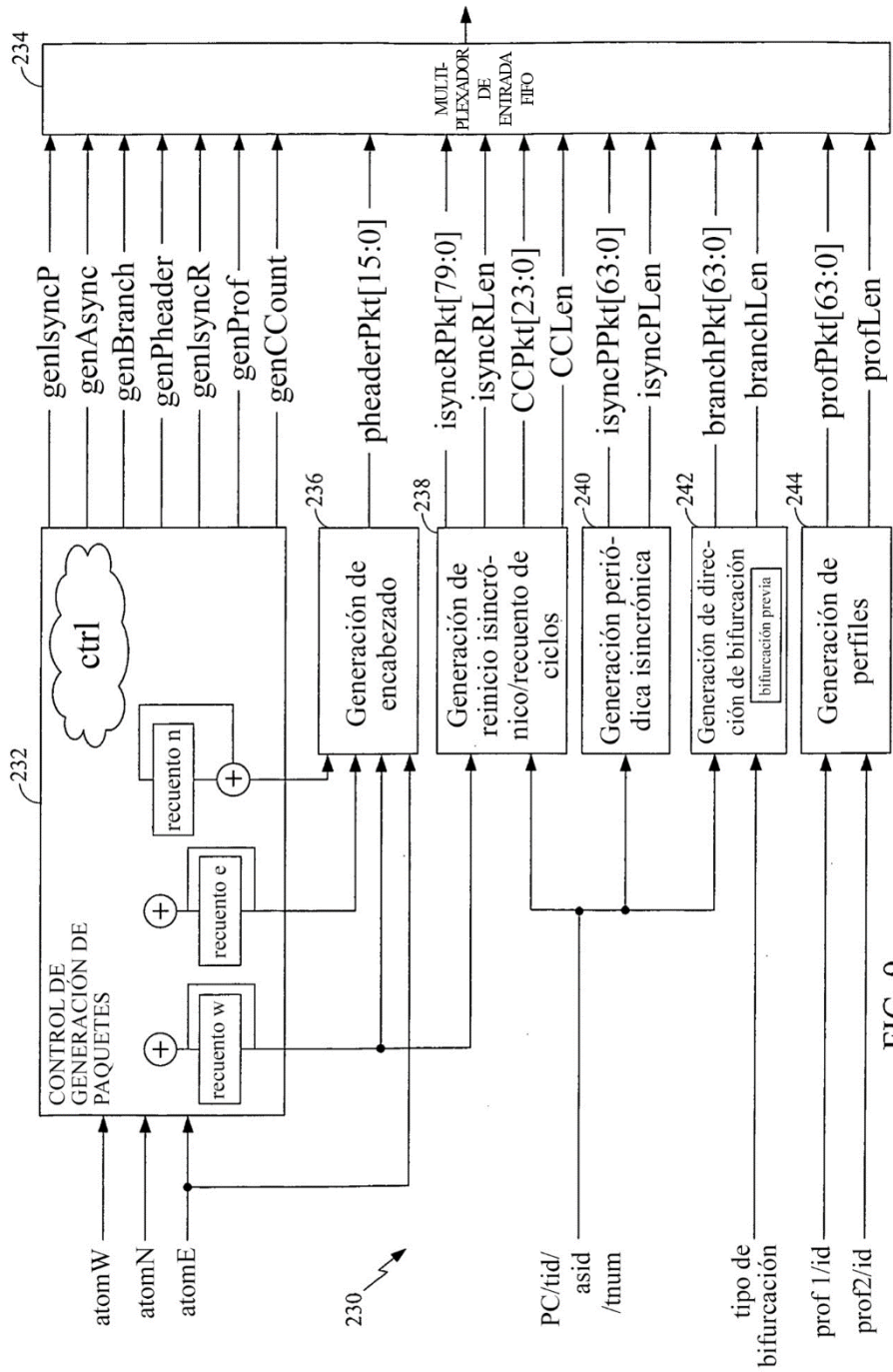


FIG. 9

250 ↗

ÁTOMO	DEFINICIÓN (SI NO ES SALTO DUAL)	DEFINICIÓN (SI ES SALTO DUAL)
W	CICLO DE BLOQUEO	CICLO DE BLOQUEO
E	INSTRUCCIÓN EJECUTADA. SALTO CONDICIONAL TOMADO	PRIMER SALTO TOMADO
N	SALTO CONDICIONAL FALSO. EXTREMUM DE BUCLE CAIDO A TRAVES	AMBOS SALTOS NO TOMADOS
S		SEGUNDO SALTO TOMADO.

FIG. 10

byte 0	1	PC[5:1]	T	1
byte 1	1	PC[12:6]		
byte 2	1	PC[19:13]		
byte 3	1	PC[26:20]		
byte 4	1	0	0	PC[31:27]
byte 5		TID[7:0]		
byte 6		TID[15:8]		
byte 7		TID[23:16]		
byte 8		TID[31:24]		
byte 9	1	0	ASID[5:0]	
byte 10		TIPO[7:0]		

FIG. 11

T : INDICADOR PRESENTE DE CAMPO DE TIPO

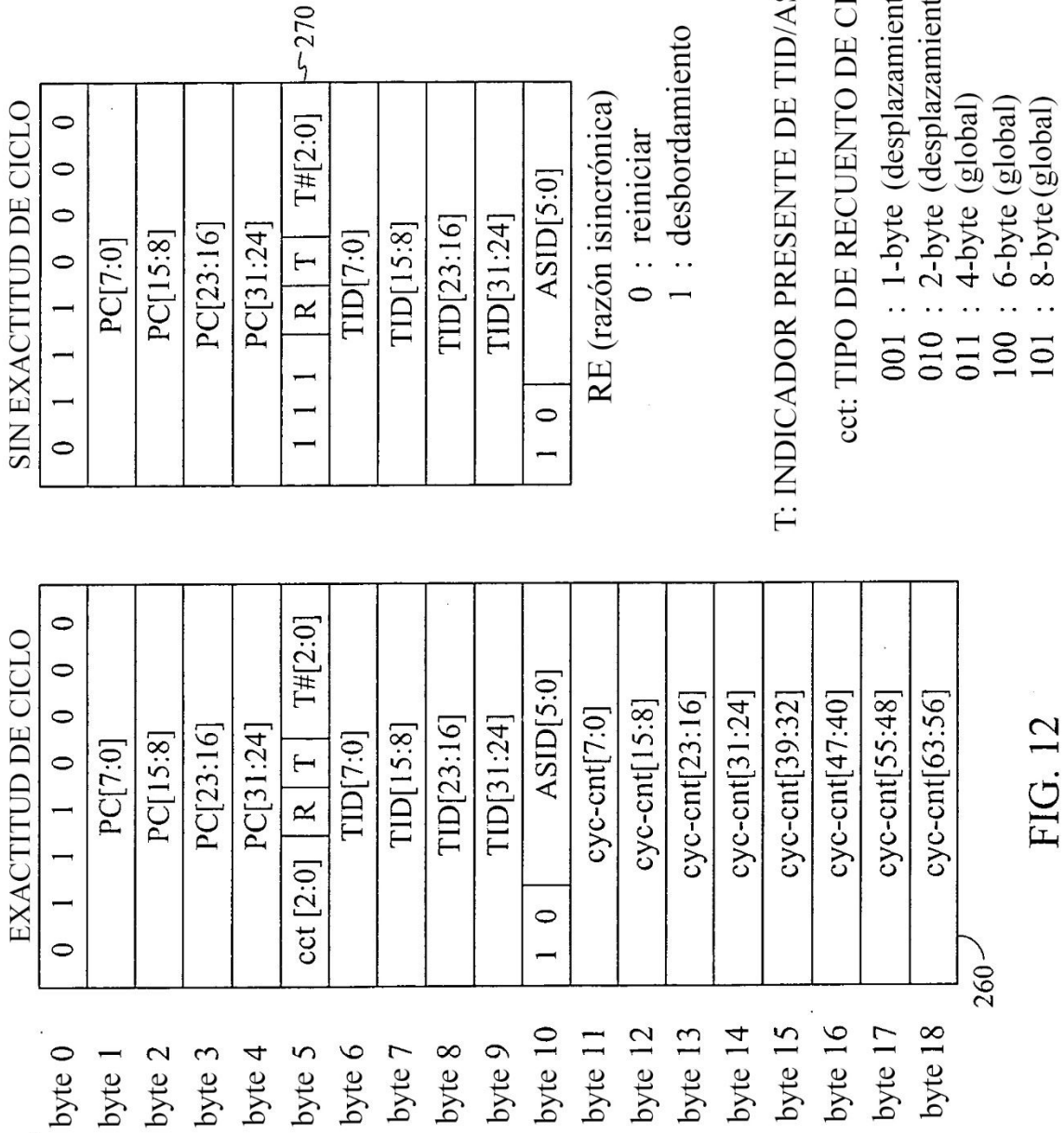
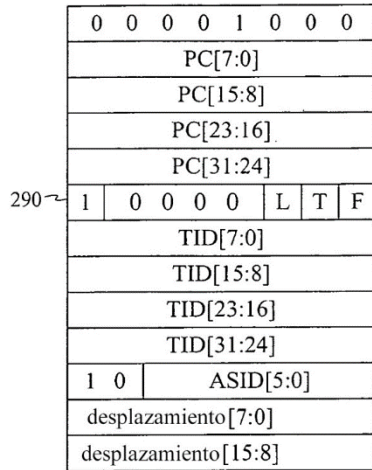


FIG. 12



290

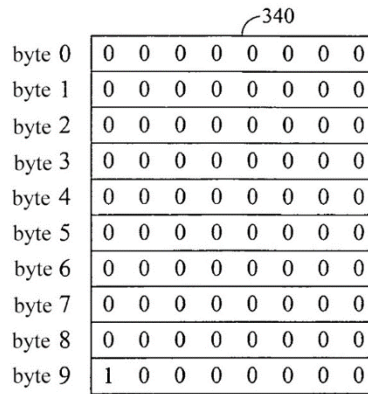


FIG. 16

T: INDICADOR PRESENTE DE TID/ASID
 F: CAMPO DE DESPLAZAMIENTO PRESENTE
 V: LONGITUD DEL CAMPO DE DESPLAZAMIENTO
 (1 O 2 bytes)

FIG. 13

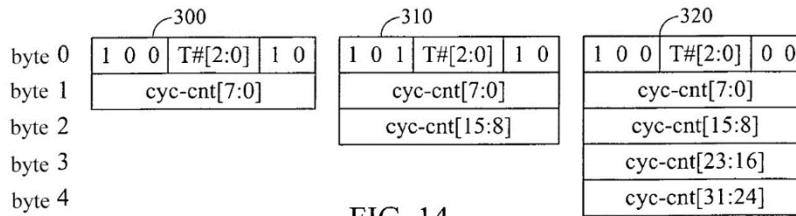


FIG. 14

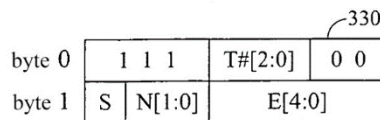


FIG. 15

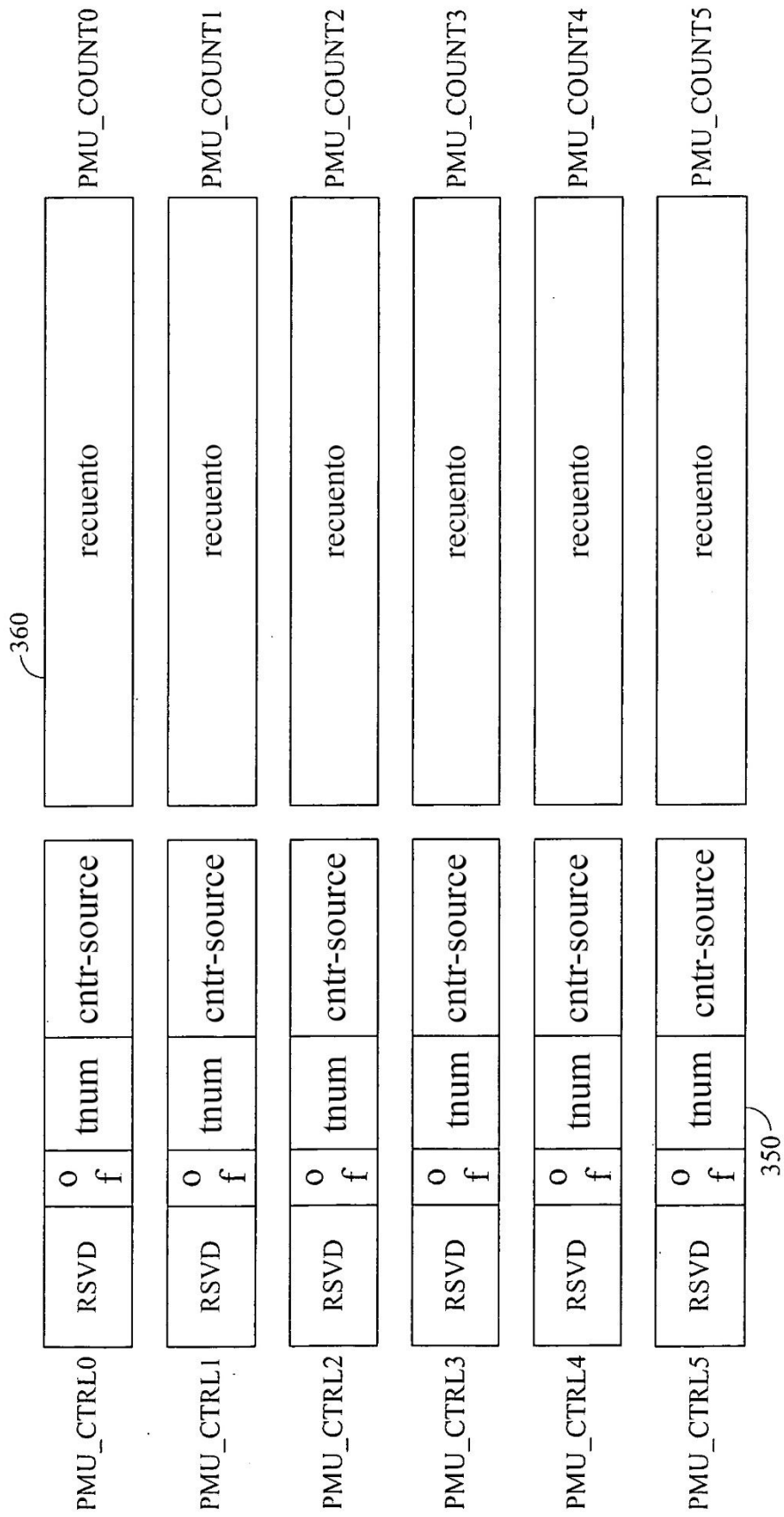


FIG. 17