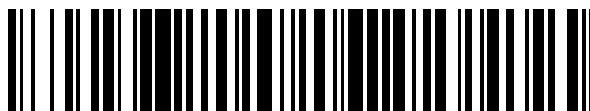


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 588 185**

51 Int. Cl.:

G06F 11/36

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **03.08.2007 PCT/US2007/075194**

87 Fecha y número de publicación internacional: **21.02.2008 WO08021763**

96 Fecha de presentación y número de la solicitud europea: **03.08.2007 E 07813770 (0)**

97 Fecha y número de publicación de la concesión europea: **25.05.2016 EP 2054808**

54 Título: **Circuito de depuración de comparación de modo operativo de un conjunto de instrucciones de procesador**

30 Prioridad:

09.08.2006 US 463379

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

31.10.2016

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)
ATTN: INTERNATIONAL IP ADMINISTRATION,
5775 MOREHOUSE DRIVE
SAN DIEGO, CA 92121, US**

72 Inventor/es:

**BURKE, KEVIN, CHARLES;
STEMPEL, BRIAN, MICHAEL;
STREETT, DAREN, EUGENE;
SAPP, KEVIN, ALLEN;
DEBRUYNE, LESLIE, MARK;
RIZK, NABIL, AMIR;
SARTORIUS, THOMAS, ANDREW y
SMITH, RODNEY, WAYNE**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 588 185 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

Circuito de depuración de comparación de modo operativo de un conjunto de instrucciones de procesador

5 CAMPO

La presente invención se refiere en general al campo de los procesadores y, en particular, a un sistema y procedimiento de depuración de código en un procesador a través de un circuito de depuración que incluye la consideración del modo operativo de un conjunto de instrucciones de procesador.

10

ANTECEDENTES

Los procesadores modernos son sistemas muy complicados. La mayoría de los procesadores modernos emplean una arquitectura segmentada, donde las instrucciones secuenciales, cada una con múltiples etapas de ejecución, se solapan en ejecución. Muchos procesadores, conocidos como "procesadores superescalares", incluyen dos o más segmentos separados para la ejecución de instrucciones en paralelo. Para evitar paradas en el segmento debido a las instrucciones de bifurcación, la mayoría de los procesadores emplean diversas formas de predicción de ramificación, con la carga y la ejecución especulativa de las instrucciones cuando se toman las ramificaciones predichas. Para mejorar el rendimiento, muchos procesadores modernos separan las direcciones de registros lógicos de los registros de almacenamiento de memoria física correspondientes, conocido como cambio de nombre de registro. Para proporcionar programas con la ilusión de una memoria ilimitada y rápida, muchos procesadores ejecutan código en un espacio de direcciones virtuales, traduciendo las direcciones a uno o más espacios de direcciones físicas como datos que atraviesan una jerarquía de memoria (por ejemplo, registro, memoria caché, memoria principal). Con esta complejidad, la ejecución de código dentro de un procesador moderno es muy difícil de seguir y validar con precisión. En particular, las anomalías pueden ser muy difíciles de depurar.

Además, el software que se ejecuta en los procesadores modernos es por sí mismo extremadamente complejo. Con el advenimiento de las arquitecturas de procesador de cálculo de conjunto de instrucciones reducidas (RISC), gran parte de la complejidad computacional y lógica de la computación emigraron de las instrucciones del procesador a los compiladores de optimización. Es decir, los compiladores crean operaciones complejas a partir de un conjunto relativamente pequeño de instrucciones del procesador, cada una de las cuales está optimizada para una función particular y específica. Esto se traduce en una secuencia más larga y compleja de instrucciones, incluyendo, por ejemplo, lógica, aritmética, carga/almacenamiento y operación de ramificación, para una tarea computacional dada. Dicho código complejo puede ser difícil de depurar cuando los errores provocan un comportamiento anómalo del programa.

Para ayudar en la difícil tarea de depuración del código complejo que se ejecuta en un procesador complejo, las herramientas de depuración están integradas en muchos procesadores. Estas pueden comprender comparadores de direcciones y de datos para la identificación de instrucciones específicas y/o patrones de datos. Las herramientas de depuración pueden incluir adicionalmente comparadores de rango de direcciones, de modo que la depuración o el rastreo puede estar limitado a segmentos de código predeterminados. Otras herramientas de depuración pueden incluir contadores, secuenciadores, y similares, para proporcionar flexibilidad en la especificación de las condiciones que especifican los puntos de interrupción y/o trazan los factores desencadenantes. La información de rastreo puede proporcionarse fuera del chip, tal como a través de un bus dedicado, o se puede almacenar en una memoria intermedia en el chip dedicado. Los puntos de interrupción pueden desencadenar una o más señales externas, tales como armar y/o desencadenar un analizador lógico o iluminar un LED; pueden causar una excepción, ramificando la ejecución de código a una rutina de depuración; o pueden simplemente detener la ejecución, permitiendo inspeccionar el contenido de varios registros y de la memoria caché. Estas herramientas de depuración, que no se utilizan durante la ejecución normal del procesador, se han diseñado en los circuitos de procesador, de tal manera que se minimice el impacto sobre el rendimiento del procesador y el consumo de energía.

Los programadores pueden establecer explícitamente el modo operativo del conjunto de instrucciones adecuado en software, antes de la ejecución de instrucciones de acuerdo con la codificación del conjunto de instrucciones correspondiente. Sin embargo, el software errante puede ramificarse a veces a una dirección de instrucción particular que estaba destinada a ser ejecutada mientras estaba en un modo operativo del conjunto de instrucciones, pero mientras el procesador está en realidad en algún otro modo operativo del conjunto de instrucciones. En tal caso, el procesador puede intentar ejecutar la instrucción en esa dirección mediante la decodificación inadecuada del conjunto de instrucciones, dando lugar a resultados incorrectos.

Por ejemplo, algunas versiones de la arquitectura de procesador ARM incluyen al menos dos modos operativos del conjunto de instrucciones: un modo ARM de 32 bits y un modo Thumb de 16 bits. La Tabla 1 enumera un fragmento de código de instrucciones ARM:

Tabla 1: Codificación de instrucciones de modo ARM

000096F0	cpy	r4, r0
000096F4	b1	0xA8A0
000096FC	cpy	r1, r4
00009700	ldr	r0,[r0,#0x8]

5 La Tabla 2 enumera el mismo código traducido en modo Thumb:

Tabla 2: Codificación de instrucciones de modo Thumb

000096F0	y	r0, r0
000096F2	b	0x9A36
000096F4	lsl	r1, r5, #0x11
000096F6	add.w	r0, r0, r4, lsl #0x4
000096FA	b	0x9A3E
000096FC	lsl	r0, r1, #0x0
000096FE	b	0x9222
00009700	y	r0, r2

10 Deben tenerse en cuenta, en particular, las instrucciones de ramificación en 96F2, 96FA y 96FE. Debido a la ramificación irregular, puede ser difícil determinar el punto en el cual se provocó un error en el código ARM de la Tabla 1 para ser interpretado en modo Thumb como en la Tabla 2 y, por lo tanto, para depurar el error. Esta dificultad surge porque los circuitos de depuración de la técnica anterior no incluyen el modo operativo del conjunto de instrucciones del procesador como una entrada a la lógica que desencadena los puntos de interrupción, inicia los rastros, y similares. El documento GB 2374694 A describe un codificador que codifica un conjunto de instrucciones con la dirección de instrucción.

La mayoría de los procesadores que soportan dos o más "modos" operativos, tales como modos de supervisor y de usuario, o modos reales y protegidos, cambian entre los modos provocando una excepción y ramificando a una rutina de cambio de modo. Esta operación se detecta fácilmente utilizando herramientas de depuración de la técnica anterior y software mediante el establecimiento de un punto de interrupción o de inicio de un rastro en la dirección de la instrucción de la rutina de cambio de modo. Un procesador que conmuta los modos operativos del conjunto de instrucciones sin provocar una excepción es un ejemplo de comportamiento de código que es difícil de diagnosticar con herramientas de depuración de la técnica anterior, que no incluyen el modo operativo del conjunto de instrucciones del procesador como una consideración en el desencadenamiento de los puntos de interrupción, el inicio de los rastros, y similares.

El documento GB2374694 describe un procedimiento para almacenar información del conjunto de instrucciones que comprende un circuito de procesamiento para ejecutar instrucciones de procesamiento de cualquiera de una pluralidad de conjuntos de instrucciones de las instrucciones de procesamiento, especificándose cada instrucción de procesamiento mediante una dirección de instrucción que identifica esa ubicación de la instrucción de procesamiento en la memoria.

El documento US2006/149927 describe un procesador capaz de recibir una pluralidad de conjuntos de instrucciones desde al menos una memoria, y capaz de la ejecución de múltiples subprocesos de la pluralidad de conjuntos de instrucciones.

El documento US6308323 describe un procedimiento para la compilación de un programa fuente para un procesador que tiene una pluralidad de diferentes conjuntos de instrucciones a alta velocidad mediante la selección de un conjunto de instrucciones óptimas.

RESUMEN

La invención se refiere a un procedimiento de depuración de código de software como se establece en las reivindicaciones 1 a 7, y se refiere además a un procesador tal como se establece en las reivindicaciones 8 a 14.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La figura 1 es un diagrama de bloques funcional de un procesador.

La figura 2 es un diagrama de bloques funcional de un segmento de procesador, incluyendo un circuito de depuración.

La figura 3 es un diagrama de flujo de un proceso de diagnóstico/depuración.

DESCRIPCIÓN DETALLADA

La figura 1 representa un diagrama de bloques funcional de un procesador 10. El procesador 10 ejecuta instrucciones en un segmento de ejecución de instrucciones 12 de acuerdo con una lógica de control 14. La lógica de control 14 mantiene un contador de programa (PC) 15, y establece y borra bits en uno o más registros de estado 16 para indicar, por ejemplo, el modo operativo del conjunto de instrucciones actual, información sobre los resultados de las operaciones aritméticas y las comparaciones lógicas (cero, llevar, igual, no igual), y similares. En algunos modos de realización, el segmento 12 puede ser un diseño superescalar, con múltiples segmentos paralelos. El segmento 12 también puede denominarse como una unidad de ejecución. Un registro de propósito general (GPR) 20 proporciona registros accesibles por el segmento 12, y que comprende la parte superior de la jerarquía de memoria.

El procesador 10, que ejecuta las instrucciones de al menos dos conjuntos de instrucciones en diferentes modos operativos del conjunto de instrucciones, además, incluye un circuito de depuración 18, operativo para comparar, tras la ejecución de cada instrucción, al menos un modo operativo de un conjunto de instrucciones objetivo predeterminadas con el actual modo operativo del conjunto de instrucciones, y para proporcionar una indicación de una coincidencia entre los dos. El circuito de depuración 18 se describe con mayor detalle a continuación.

El segmento 12 extrae instrucciones desde una memoria caché de instrucciones (caché I) 26, con la traducción de las direcciones de memoria y los permisos administrados por una memoria intermedia de consulta de traducción secundaria de instrucciones (ITLB) 28. A los datos se accede desde una memoria caché de datos (caché D) 30, con traducción de las direcciones de memoria y permisos gestionados por una memoria intermedia de consulta de direcciones de traducción principal (TLB) 32. En diversos modos de realización, la ITLB 28 puede comprender una copia de parte de la TLB 32. Como alternativa, la ITLB 28 y la TLB 32 pueden estar integradas. Del mismo modo, en diversos modos de realización del procesador 10, la memoria caché I 26 y la memoria caché D 30 pueden estar integradas, o unificadas. La falta en la memoria caché I 26 y/o la memoria caché D 30 provoca un acceso a la memoria principal (fuera de chip) 38, 40 mediante una interfaz de memoria 34. La interfaz de memoria 34 puede ser una entrada principal a una interconexión de bus 42 que implementa un bus compartido a uno o más dispositivos de memoria 38, 40. Pueden conectarse adicionalmente dispositivos principales adicionales (no mostrados) a la interconexión de bus 42.

El procesador 10 puede incluir una interfaz de entrada/salida (E/S) 44, que puede ser un dispositivo principal en un bus periférico, a través de la que la interfaz E/S 44 puede acceder a diversos dispositivos periféricos 48, 50. Los expertos en la técnica reconocerán que son posibles numerosas variaciones del procesador 10. Por ejemplo, el procesador 10 puede incluir una memoria caché de segundo nivel (L2) para una o ambas de las memorias caché I y D 26, 30. Además, uno o más de los bloques funcionales representados en el procesador 10 pueden omitirse de un modo de realización particular. Otros bloques funcionales que pueden residir en el procesador 10, tales como un controlador JTAG, un predecodificador de instrucciones, una memoria caché de dirección de destino de ramificación, y similares, no son pertinentes para una descripción de la presente invención, y se omiten para mayor claridad.

La figura 2 representa un diagrama de bloques funcional de un modo de realización del circuito de depuración 18. El circuito de depuración 18 incluye un registro del modo operativo del conjunto de instrucciones objetivo 52, un registro de direcciones del inicio del intervalo de direcciones 54, y un registro de direcciones del final del intervalo de direcciones 56. Los registros 52, 54, 56 se cargan mediante programadores a través de software de diagnóstico antes de una operación de diagnóstico/depuración. El valor escrito en el registro del modo operativo del conjunto de instrucciones objetivo 52 se compara con el modo operativo del conjunto de instrucciones del procesador actual durante la ejecución de cada instrucción, para desencadenar un punto de interrupción, una función de rastreo, u otro aviso. Los valores de los registros de direcciones de inicio y final 52, 56, son el principio y el final, respectivamente, del intervalo de direcciones objetivo sobre el que el circuito de depuración 18 monitoriza una coincidencia entre los modos operativos del conjunto de instrucciones actual y objetivo.

Como los expertos en la técnica reconocerán fácilmente, el circuito de depuración 18 puede incluir parámetros adicionales personalizables, y bloques funcionales adicionales, permitiendo puntos de interrupción, rastros, y similares que desencadenen en una amplia diversidad de condiciones. Estos se han omitido de la figura 2 para mayor claridad, pero en general pueden incluir todos los parámetros del circuito de depuración y la funcionalidad conocidos en la técnica.

Antes de una ejecución de diagnóstico/depuración, un programador carga los registros de dirección inicial y final 54, 56 para definir una región de dirección de destino, que puede ir desde una sola dirección a la totalidad del intervalo de direcciones del código que está siendo probado. En otros modos de realización, pueden proporcionarse múltiples registros de dirección inicial y final para definir una pluralidad de intervalos de direcciones objetivo. El programador, además, carga el registro del modo operativo del conjunto de instrucciones objetivo 52. Por ejemplo, en el caso de un procesador ARM que ejecuta el código en su totalidad en el modo ARM, un programador puede cargar el registro 52 con un valor que indica el modo Thumb, para establecer un punto de interrupción que hace que el circuito de depuración 18 detenga la ejecución, por ejemplo, si el procesador 10 ejecuta la instrucción dentro del intervalo de

direcciones definido por los registros 54, 56 en el modo Thumb.

Durante la operación de diagnóstico/depuración, el valor almacenado en el registro del modo operativo del conjunto de instrucciones objetivo 52 se compara con un bit de registro de estado que indica el conjunto de instrucciones del modo operativo actual del procesador 10 en la lógica de comparación, tal como la puerta AND 58. Por ejemplo, en un procesador ARM 10, el bit 5 del registro del estado del programa actual (CPSR) 16 puede monitorizarse. La salida 60 de la lógica de comparación 58 se determinará solamente cuando el modo operativo del conjunto de instrucciones actual del procesador 10 coincida con el modo operativo del conjunto de instrucciones indicado por el registro del modo operativo del conjunto de instrucciones objetivo 52.

En modos de realización en los que el procesador 10 ejecuta instrucciones de más de dos conjuntos de instrucciones en diferentes modos operativos del conjunto de instrucciones, el registro del modo operativo del conjunto de instrucciones objetivo 52 y los bits indicadores de modo del CPSR (u otro registro de estado) 16 comprenderán los valores de múltiples bits, y la lógica de comparación 58 puede incluir circuitos adicionales, tales como decodificadores. En diversos modos de realización, se pueden utilizar otras indicaciones del modo operativo actual del conjunto de instrucciones en lugar del CPSR 16, tal como la salida de la lógica de decodificación de instrucciones.

Al mismo tiempo, el valor actual del PC 15 se compara con los registros del intervalo de direcciones 54, 56 en la lógica de comparación de direcciones 68. La salida 70 de la lógica de comparación de direcciones 68 se determina cada vez que el valor actual del PC 15 está dentro del intervalo de direcciones objetivo. En diversos modos de realización, las indicaciones de la dirección de la instrucción "actual" pueden variar del PC 15, según se requiera o se desee. Por ejemplo, el circuito de comparación de direcciones 68 puede comparar la dirección de instrucciones en una etapa de decodificación del segmento 12, una etapa de ejecución, o cualquier otra etapa del segmento. Además, se puede utilizar una dirección más dinámica, tal como la dirección de la última instrucción en el segmento 12 para comprometerse con la ejecución.

La salida 70 de la lógica de comparación de direcciones 68 está en AND con la salida 60 de la lógica de comparación del indicador de modo operativo del conjunto de instrucciones 58 en la puerta AND 72, para generar una salida 20 que indica una coincidencia diaria entre los modos operativos del conjunto de instrucciones actual y objetivo que se produce dentro del intervalo de direcciones objetivo. En un modo de realización, la salida 20 se proporciona como una salida para el procesador (véase la figura 1). Esta salida 20 puede armar y/o desencadenar un analizador lógico, iluminar un LED, o desencadenar alguna otra acción o alarma. En un modo de realización, como se representa en la figura 1, la salida del indicador de coincidencias del modo operativo del conjunto de instrucciones 20 puede comprender una entrada al controlador 14, para detener la ejecución del procesador 10.

En un modo de realización, la salida del indicador de coincidencias del modo operativo del conjunto de instrucciones 20 puede inicializar o terminar una operación de trazado de instrucciones, estando la información de trazado dirigida a las clavijas de salida del procesador 10 o almacenada en una memoria intermedia de seguimiento en el chip (no mostrada). En un modo de realización, como se representa en la figura 2, la salida del indicador de coincidencias del modo operativo del conjunto de instrucciones 20 puede desencadenar un registro de direcciones de correspondencia con el modo operativo del conjunto de instrucciones 74, capturando el valor del PC 15 y, por lo tanto, la dirección de la instrucción en la que se puede detectar la coincidencia entre los modos operativos del conjunto de instrucciones actual y objetivo. Esto puede ser de particular valor en la detección de un cambio de un modo operativo del conjunto de instrucciones a otro. La salida 76 de este registro 74 puede encaminarse a unas clavijas del procesador 10 para una inspección externa.

En un modo de realización, la salida del indicador de correspondencia del modo operativo del conjunto de instrucciones 20 puede provocar una excepción, ramificando la ejecución del código a una rutina predefinida. Esta rutina de correspondencia del modo operativo del conjunto de instrucciones puede, por ejemplo, leer el registro de direcciones de correspondencia del modo operativo del conjunto de instrucciones 74. En general, la salida del indicador de correspondencia del modo operativo del conjunto de instrucciones 20 puede desencadenar cualquier operación de depuración conocida, según se requiera o se desee.

La figura 3 representa un procedimiento de depuración de código de acuerdo con uno o más modos de realización de la presente invención. Comenzando en el bloque 78, un programador inicia los registros del circuito de depuración (bloque 80). Esto puede comprender, por ejemplo, cargar el registro del modo operativo del conjunto de instrucciones objetivo 52, y ajustar el intervalo de direcciones objetivo mediante la carga del registro de la dirección de inicio 54 y el registro de la dirección final 56. Pueden iniciarse los registros del circuito de depuración adicionales (no mostrados), según sea necesario o se desee.

Después, el programador comienza la ejecución del código que se va a depurar (bloque 82), cargando y ejecutando sucesivamente la siguiente instrucción (bloque 84) en la secuencia de código. Independientemente de la dirección, si el modo operativo del conjunto de instrucciones actual del procesador 10 no coincide con el establecido por el programador en el registro del modo operativo del conjunto de instrucciones objetivo 52 (bloque 86), entonces se carga la siguiente la instrucción y se ejecuta (bloque 84) hasta el final del código a depurar (bloque 88), en cuyo

caso el procedimiento se detiene en el bloque 90.

Si, tras la ejecución de una instrucción (bloque 84), la dirección de la instrucción se encuentra dentro del intervalo de direcciones predeterminado, y el modo operativo del conjunto de instrucciones actual coincide con el establecido por el programador en el registro del modo operativo del conjunto de instrucciones objetivo 52 (bloque 86), entonces el circuito de depuración determina la salida 20 y puede tomar cualquier número de acciones, según lo especificado por el programador.

Por ejemplo, la salida del circuito de depuración 20 puede provocar que el procesador 10 tome una excepción, ramificando la ejecución de código a una rutina de coincidencia del modo operativo del conjunto de instrucciones (bloque 92). La salida del circuito de depuración 20 puede hacer que el controlador del segmento 14 detenga la ejecución (bloque 94), lo que permite al programador inspeccionar el contenido de varios registros, líneas de caché, y similares. El circuito de depuración 18 puede determinar la salida 20 externamente (bloque 96), que puede utilizarse para armar o desencadenar un analizador lógico, iluminar un LED, o similares. El circuito de depuración 18 puede, además, enviar a la dirección de correspondencia del modo operativo del conjunto de instrucciones para la inspección externa (bloque 98). El circuito de depuración 18 puede iniciar o detener una operación de trazado (bloque 100). Aunque se representa como alternativas, el circuito de depuración 18 puede, en un modo de realización dado, ejecutar dos o más de los bloques 92-96. En general, el circuito de depuración 18 puede tomar cualquier acción conocida en la técnica que sea beneficiosa para la depuración del código en el procesador 10, después de haber detectado una coincidencia entre el modo operativo del conjunto de instrucciones actual y el modo operativo del conjunto de instrucciones objetivo.

Como se utiliza en el presente documento, la expresión "conjunto de instrucciones" se refiere a un conjunto de codificaciones mediante las cuales los datos de instrucciones (por ejemplo, el contenido de una línea de memoria caché I) se interpretan por un procesador 10 como instrucciones ejecutables. La expresión "modo operativo del conjunto de instrucciones" se refiere a un modo operativo que se puede identificar del procesador 10 en el que los datos de instrucción se interpretan de acuerdo con una codificación del conjunto de instrucciones particular. En particular, "modo operativo del conjunto de instrucciones" se distingue de los "modos" operativos del procesador conocidos respecto a los permisos (por ejemplo, modo de supervisor frente a modo de usuario), direccionamiento de memoria (por ejemplo, modo real frente a modo protegido), y similares.

Aunque la presente invención se ha descrito en el presente documento con respecto a características, aspectos y modos de realización particulares de la misma, será evidente que numerosas variaciones, modificaciones y otros modos de realización son posibles dentro del alcance de las reivindicaciones adjuntas. Por lo tanto, los presentes modos de realización deben interpretarse en todos los aspectos como ilustrativos y no restrictivos, y todos los cambios que entran dentro del significado y el rango de equivalencia de las reivindicaciones adjuntas están destinados a incluirse en los mismos.

REIVINDICACIONES

- 5 1. Un procedimiento para la depuración de código de software que comprende una pluralidad de instrucciones en un procesador (10) que tiene al menos dos modos operativos de conjuntos de instrucciones diferentes, **caracterizado por que** comprende:

10 recibir (80) un modo operativo de conjunto de instrucciones objetivo antes de ejecutar cualquier instrucción de código; ejecutar (82, 84) instrucciones de código y, para cada instrucción ejecutada, comparar (86) el modo operativo del conjunto de instrucciones del procesador actual (10) con el modo operativo del conjunto de instrucciones objetivo; y

15 desencadenar (92, 94, 96, 98, 100) una alerta si el modo operativo del conjunto de instrucciones del procesador actual (10) coincide con el modo operativo del conjunto de instrucciones objetivo.
2. El procedimiento de la reivindicación 1, en el que la alerta comprende detener la ejecución de instrucciones.
3. El procedimiento de la reivindicación 1, en el que la alerta comprende controlar una operación de trazado.
- 20 4. El procedimiento de la reivindicación 1, en el que la alerta comprende causar una excepción.
5. El procedimiento de la reivindicación 1, en el que la alerta comprende emitir una señal indicativa de una coincidencia entre el modo operativo del conjunto de instrucciones del procesador actual y el modo operativo del conjunto de instrucciones objetivo.
- 25 6. El procedimiento de la reivindicación 1, en el que la alerta comprende emitir la dirección de la instrucción para la que el modo operativo del conjunto de instrucciones del procesador actual coincida con el modo operativo del conjunto de instrucciones objetivo.
- 30 7. El procedimiento de la reivindicación 1, que comprende además:

 recibir (80) un intervalo de direcciones objetivo antes de ejecutar (82, 84) cualquier instrucción de código;

35 para cada instrucción ejecutada, comparar (86) la dirección de la instrucción actual en el intervalo de direcciones objetivo; y

40 desencadenar (92, 94, 98, 100) la alerta sólo si el modo operativo del conjunto de instrucciones del procesador actual (10) coincide con el modo operativo del conjunto de instrucciones objetivo y la dirección de la instrucción actual está dentro del intervalo de direcciones objetivo.
8. Un procesador (10) operativo para ejecutar instrucciones de acuerdo con dos o más codificaciones de conjuntos de instrucciones, cada una en un modo operativo del conjunto de instrucciones diferente, **caracterizado por que** comprende:

45 un indicador del modo operativo del conjunto de instrucciones actual;

 una ubicación del almacenamiento de datos (52) operativa para almacenar un indicador del modo operativo del conjunto de instrucciones objetivo; una unidad de ejecución operativa para ejecutar instrucciones de acuerdo con un modo operativo del conjunto de instrucciones actual; y

50 un circuito de comparación (58, 68, 72) operativo para comparar el modo operativo del conjunto de instrucciones actual con el modo operativo del conjunto de instrucciones objetivo al ejecutar cada instrucción, y para emitir una indicación (20) si el modo operativo del conjunto de instrucciones actual coincide con el modo operativo del conjunto de instrucciones objetivo para activar una alerta.
- 55 9. El procedimiento de la reivindicación 8, en el que la indicación comprende una señal operativa para detener la ejecución de instrucciones.
- 60 10. El procedimiento de la reivindicación 8, en el que la indicación comprende una señal operativa para controlar una operación de trazado.
11. El procedimiento de la reivindicación 8, en el que la indicación comprende una señal operativa para causar una excepción.
- 65 12. El procesador (10) de la reivindicación 8, en el que la indicación (20) comprende una señal que indica que el modo operativo del conjunto de instrucciones actual coincide con el modo operativo del conjunto de

instrucciones objetivo.

- 5 **13.** El procesador (10) de la reivindicación 8, en el que la indicación (20) comprende la dirección de la instrucción que se ejecuta cuando se detecta una coincidencia entre el modo operativo del conjunto de instrucciones actual y el modo operativo del conjunto de instrucciones objetivo.
- 10 **14.** El procesador (10) de la reivindicación 8, que comprende además una ubicación de almacenamiento de datos (52, 54) operativa para almacenar un intervalo de direcciones de destino, y en el que el circuito de comparación (58, 68, 72) es operativo para emitir una indicación sólo si el modo operativo del conjunto de instrucciones actual coincide con el modo operativo del conjunto de instrucciones objetivo y la dirección de la instrucción que se está ejecutando actualmente está dentro del intervalo de direcciones objetivo.

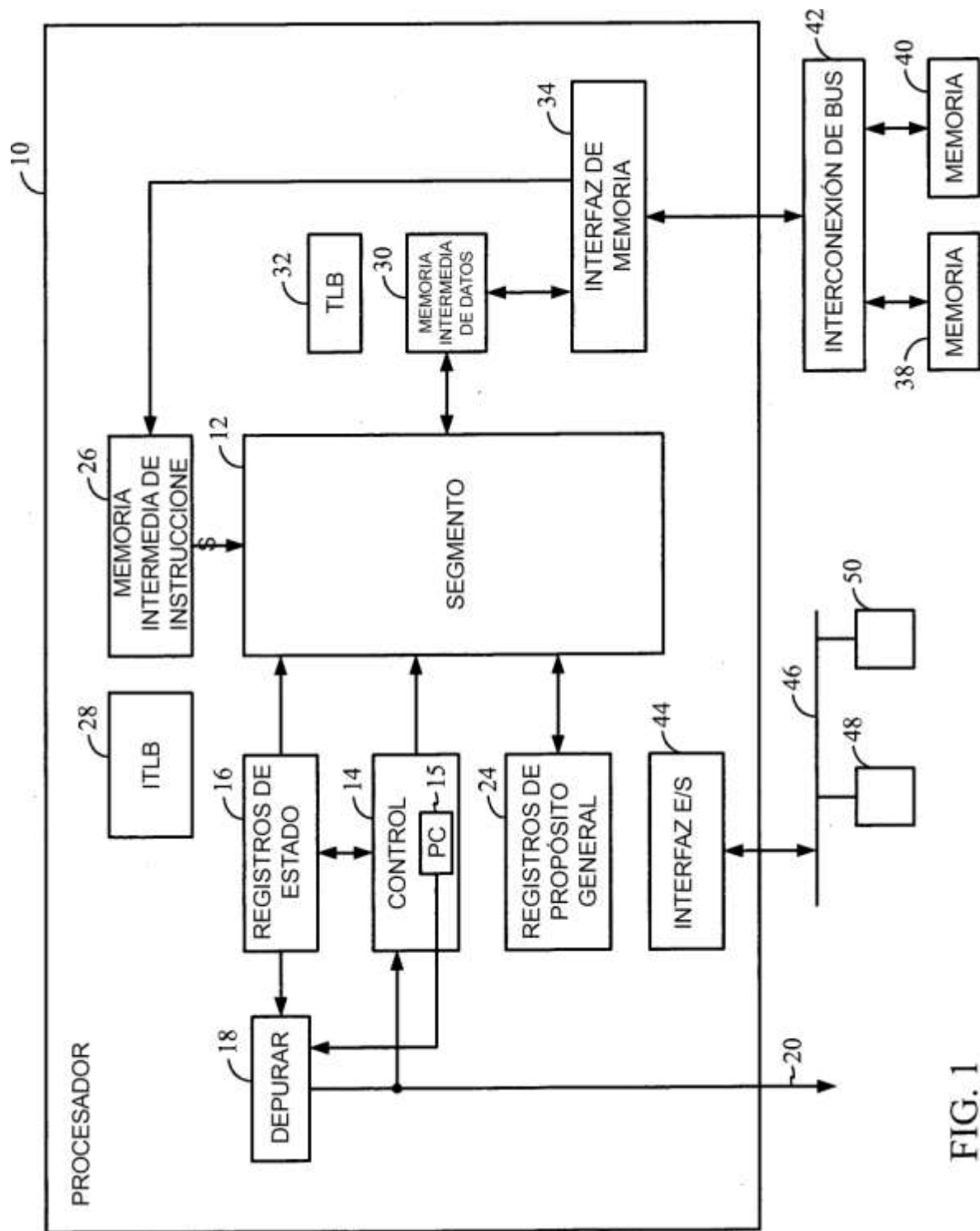


FIG. 1

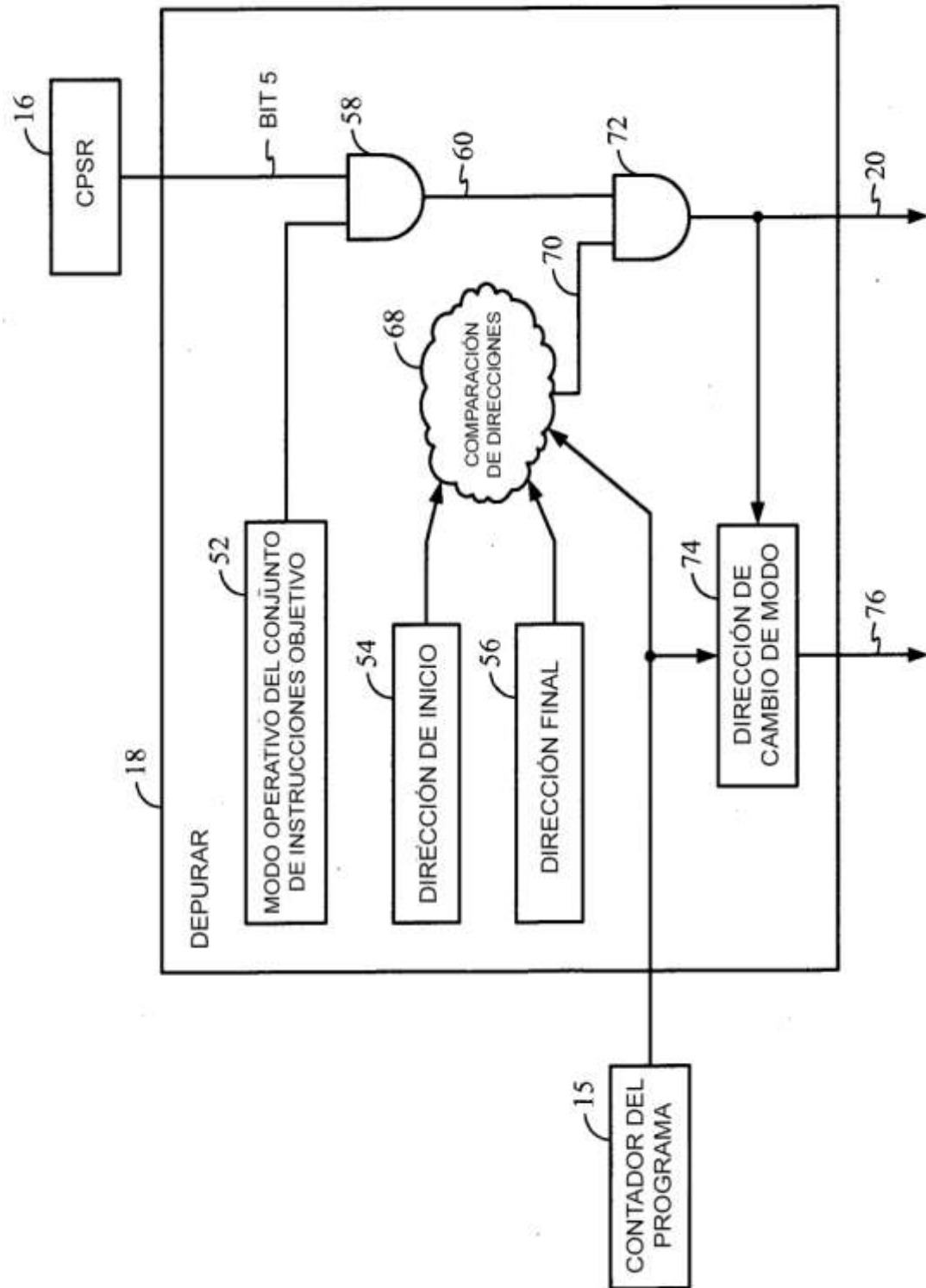


FIG. 2

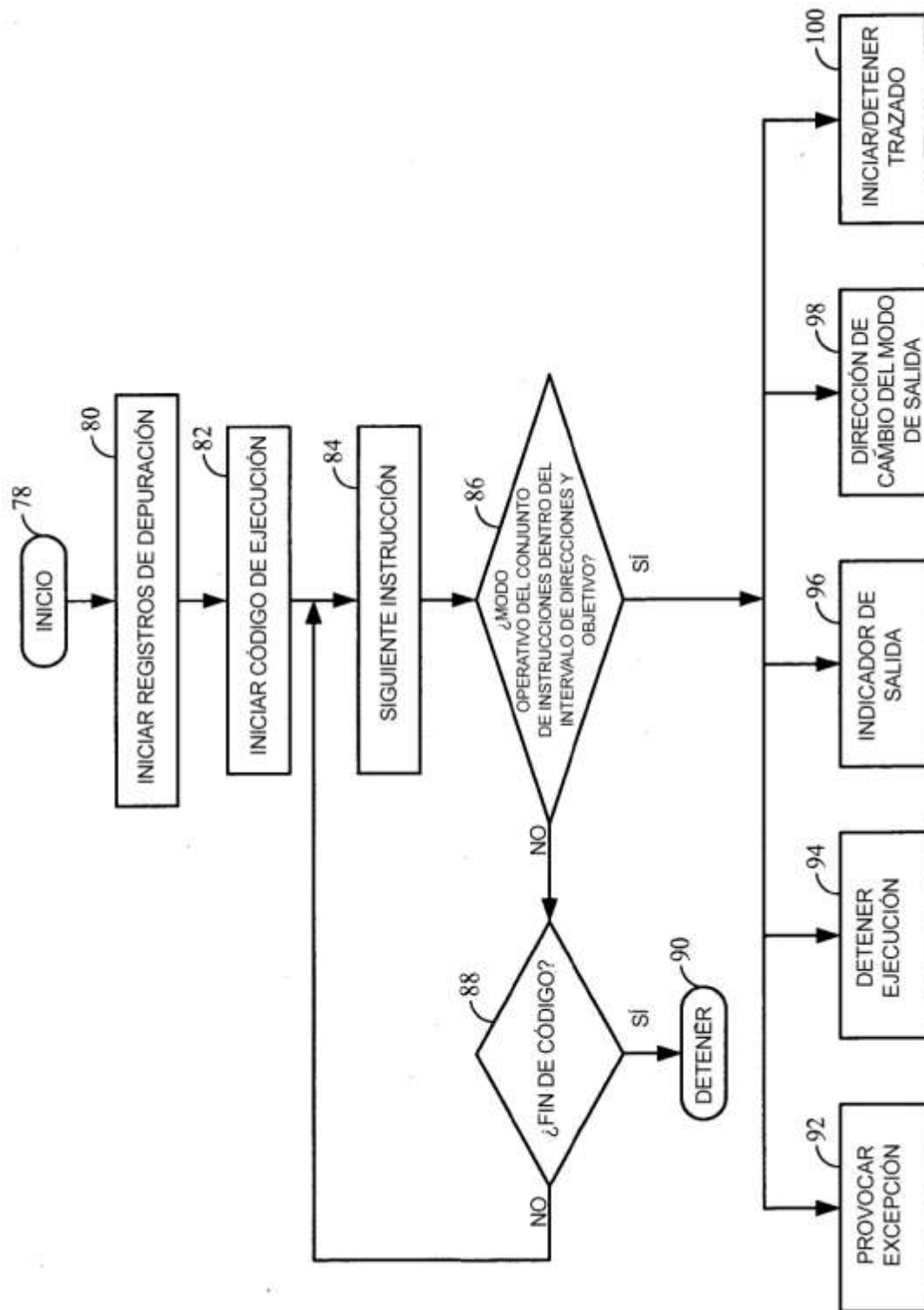


FIG. 3