

DESCRIPCIÓN
UN MÉTODO PARA EL TRAZADO SELECTIVO DE LA EJECUCIÓN DE
INSTRUCCIONES, DISPOSITIVO DE PROCESAMIENTO RELACIONADO Y
PROCESADOR

5

SECTOR DE LA TÉCNICA

10 La invención se encuadra, de forma general, en el sector Electrónica, Informática y Telecomunicaciones (TIC), si bien tiene aplicación específica en sistemas críticos propios de los sectores Aeroespacial, Automoción y de alta fiabilidad.

ANTECEDENTES DE LA INVENCION

15

Se han identificado diferentes invenciones que proponen soluciones para facilitar el trazado de instrucciones.

20

La solicitud de patente US 5996092, "System and method for tracing program execution within a processor before and after a triggering event" permite iniciar e interrumpir la traza de instrucciones, empleando un procesador de traza que trabaja en paralelo al procesador que ejecuta las propias instrucciones. El procesador de traza, tras detectar el momento de inicio de la traza mediante una instrucción específica, almacena en una memoria compartida información relativa a toda la secuencia de ejecución de instrucciones hasta el momento en el que detecta la instrucción de parada de la traza. Así que, para activar y desactivar la traza inserta una instrucción específica en el código ejecutable.

25

Herramientas comerciales como RapiTime (G. Bernat et al., "Identifying Opportunities for Worst-case Execution Time Reduction in an Avionics System," Ada User Journal, Volume 30 28, Number 3, 2007, pp. 189–194), usan ciertas técnicas con el fin de caracterizar sistemas críticos en el ámbito de la aviónica, la automoción, o los sistemas espaciales. Para ello

emplean la instrumentación del código, insertando instrucciones que disparan la traza en los puntos seleccionados previamente analizando los archivos fuente. La instrumentación, sin embargo, tiene como desventajas la sobrecarga introducida en el tiempo de ejecución debido a las instrucciones insertadas, y el consiguiente incremento del espacio ocupado por el código ejecutable.

5

Con respecto a la solicitud de patente US 2017147472, "Systems and methods for a real time embedded trace", la diferencia principal es que el sistema traza las instrucciones de salto de forma autónoma.

10

Como ya se ha descrito, la identificación del momento de ejecución de instrucciones previamente seleccionadas está siendo utilizada en sistemas críticos y permite trazar bloques de código para evaluar su tiempo de ejecución en el peor caso, teniendo en cuenta que la transición entre algunos de estos bloques, como el correspondiente a un bloque *e/se* y el bloque posterior, no implican un salto en la ejecución por lo que no serían detectados por la solución que presenta US2017147472.

15

En cuanto a la patente US6513134, "System and method for tracing program execution within a superscalar processor", presenta una mejora respecto a la US 5996092, permitiendo trabajar con procesadores superescalares que trabajan a altas frecuencias, por encima de los 400MHz. Para ello utiliza una codificación de la información que se desea trazar que permite reducir el espacio que es necesario utilizar para almacenarla en el *buffer* de traza que se proporciona como salida. Al igual que en esta patente, se trazan bloques de instrucciones para analizar su ejecución, pero definiendo una forma más flexible para el disparo de la traza y usando una codificación de la traza que permite un ahorro en cuanto a la información almacenada y al número de *pins* utilizados. Esta patente, por tanto, no evita la sobrecarga del uso de técnicas de instrumentación de código que afectan a todo el sistema, como ocurre con la invención reivindicada, sino que pretende optimizar un mecanismo de trazado, no basado en la instrumentación, sino en la detección de eventos que se ajustan a unas condiciones predefinidas.

20

25

30

Finalmente, existe una solicitud de patente denominada "UN MÉTODO Y UN DISPOSITIVO DE PROCESAMIENTO EN PARALELO DE INSTRUCCIONES DE PROGRAMA E INSTRUCCIONES DE TRAZA", con número de solicitud P201830266, pendiente de aprobación, que se diferencia notablemente de la nueva invención presentada. Si bien el problema que pretenden resolver es el mismo, en la solicitud P201830266 se facilita la instrumentación del código añadiendo una instrucción específica de traza al conjunto de instrucciones del procesador, y un diseño de procesador capaz de decodificar y ejecutar simultáneamente una instrucción efectiva y una instrucción de traza, evitando la sobrecarga de tiempo de ejecución, pero siendo intrusiva con respecto a la ubicación del código y al espacio total que ocupa, ya que requiere su modificación para la inserción de las instrucciones de traza.

Por lo tanto, se concluye que muchos de los sistemas existentes para la traza de instrucciones permiten programar eventos específicos de disparo de traza para recopilar información de traza limitada a un determinado intervalo anterior y/o posterior a dicho evento. Estos métodos, sin embargo, adolecen de cierta rigidez en cuanto a que el número de bloques que se puede trazar en cada ejecución es siempre limitado, y no se adaptan bien a las técnicas de instrumentación de código que se utilizan en la caracterización del tiempo de ejecución en el peor caso en sistemas críticos, como la que emplea la citada herramienta RapiTime.

DESCRIPCIÓN DE LA INVENCIÓN

En un primer aspecto de la invención se divulga un método para generar el contenido de una memoria auxiliar que, estando alineada con la memoria que almacena las instrucciones de un programa, se destina a almacenar la información que habilita la traza de un conjunto de instrucciones seleccionadas de dicho programa. Este método consta de un procedimiento en tres pasos: 1) A partir de los archivos fuente pre-procesados (esto es, con todas las macros expandidas) de un programa, se seleccionan las sentencias del código fuente a trazar y el tipo de traza para cada sentencia. 2) Tras la selección, se utilizan los archivos objeto con información de depuración de ese programa para localizar

automáticamente la dirección relativa de memoria donde se almacena la instrucción correspondiente a la ejecución de cada sentencia seleccionada. 3) Por último, empleando el archivo de localización de símbolos, y la información obtenida en los dos pasos anteriores, se genera el contenido de la memoria auxiliar alineada con la memoria de programa. Este contenido ubica en las posiciones absolutas, que han sido calculadas a partir del archivo de localización de símbolos, y de las direcciones relativas de memoria obtenidas en el paso 2), el código del tipo de traza elegido para cada sentencia en el paso 1), quedando así la memoria configurada con la información que habilita la traza de las sentencias seleccionadas.

5

10

El segundo aspecto de la invención es un dispositivo de procesamiento configurable capaz de trabajar en dos modos denominados "TRAZA" y NOMINAL.

15

En modo "NOMINAL" el dispositivo lee e interpreta el contenido de la memoria auxiliar como códigos de detección y corrección de errores asociado a las instrucciones almacenadas en la memoria de programa, corrigiendo las instrucciones en el caso de que el fallo sea recuperable, o generando una excepción en el caso de que no lo sea.

20

En modo "TRAZA" el dispositivo es capaz de leer e interpretar los códigos de traza almacenados en la memoria auxiliar, ejecutarlos en paralelo con las instrucciones alineadas a cada código, dando como resultado final la generación de información de traza en un registro de salida, siendo además esa información suficiente para identificar unívocamente la instrucción trazada.

25

La invención es aplicable directamente a aquellas arquitecturas que utilizan módulos de detección y corrección de errores como los módulos EDAC, si bien cualquier arquitectura puede ser objeto de la invención habilitando el acceso a una memoria auxiliar alineada con la memoria de programa, de forma que con cada palabra de la memoria de programa también se capture la palabra de la memoria auxiliar que tiene el mismo desplazamiento.

30

El dispositivo de procesamiento configurable en modo "TRAZA" y en "modo "NOMINAL",

comprende:

- 5

 - una etapa de búsqueda de instrucciones que a partir de la dirección de la siguiente instrucción a ejecutar lee, simultáneamente, una instrucción de la memoria de programa, y una palabra de la memoria auxiliar alineada con la memoria de programa. En modo “TRAZA” esta etapa genera una instrucción de traza mediante la composición del código de traza, y la dirección de la memoria de programa, lo que permite identificar unívocamente la instrucción trazada. En modo “NOMINAL” la etapa genera una instrucción de programa válida, una vez corregidos los posibles errores de almacenamiento en memoria, o genera una excepción, en el caso de que los errores detectados no pudieran corregirse;
- 10

 - una etapa de decodificación que comprende:
 - un módulo de decodificación de una instrucción de programa;
 - un módulo de decodificación de una instrucción de traza (que sólo opera en modo “TRAZA”).
- 15

 - un *pipeline* de instrucciones de programa para el procesado únicamente de instrucciones del programa;
 - un *pipeline* de traza para el procesado únicamente de instrucciones de traza (que sólo opera en modo “TRAZA”);
 - un registro de salida para la traza (que sólo opera en modo “TRAZA”);
- 20

 - una ruta de datos que a su vez comprende un conjunto de multiplexores;
 - un controlador de la ruta de datos, que tiene una interfaz compuesta de un conjunto de entradas, provenientes de elementos de las distintas etapas, y un conjunto de salidas que controlan los multiplexores, así como la carga en los registros asociados las distintas etapas y en el registro de salida para la traza;
- 25

donde el controlador de la ruta de datos está configurado para determinar, en función de su propio estado y del valor de las entradas en dicho controlador, el valor de las salidas que son enviadas a los multiplexores de la ruta de datos de tal forma que, en modo “TRAZA”, una instrucción de traza se ejecute de forma sincronizada con la instrucción de programa que ha sido leída simultáneamente, haciéndose efectiva dicha ejecución durante
- 30

la última etapa del *pipeline* de traza.

En una forma de realización de la invención, el dispositivo de procesamiento detecta la inserción de una burbuja en la etapa de decodificación de instrucciones de programa, lo que conduce a que el controlador de la ruta de datos fije una ruta que en el siguiente ciclo cargue un “0” como instrucción de traza inoperante en la etapa de decodificación de instrucciones de traza, quedando esta instrucción de traza inoperante en paralelo con la burbuja insertada en la etapa de decodificación de instrucciones de programa.

En una forma de realización de la invención, el dispositivo de procesamiento detecta la inserción en el siguiente ciclo de una burbuja en la etapa 3 del pipeline de instrucciones de programa, lo que conduce a que el controlador de la ruta de datos fije una ruta que en el siguiente ciclo cargue un “0” como instrucción de traza inoperante en la etapa 3 de instrucciones de traza, quedando esta instrucción de traza inoperante en paralelo con la burbuja insertada en la etapa 3 de instrucciones de programa.

En una forma de realización de la invención, el dispositivo de procesamiento detecta la inserción en el siguiente ciclo de una burbuja en la etapa 4 del pipeline de instrucciones de programa, lo que conduce a que el controlador de la ruta de datos fije una ruta que en el siguiente ciclo cargue un “0” como instrucción de traza inoperante en la etapa 4 de instrucciones de traza, quedando esta instrucción de traza inoperante en paralelo con la burbuja insertada en la etapa 4 de instrucciones de programa.

En una forma de realización de la invención, el dispositivo de procesamiento detecta la inserción en el siguiente ciclo de una burbuja en la etapa 5 del pipeline de instrucciones de programa, lo que conduce a que el controlador de la ruta de datos fije una ruta que en el siguiente ciclo cargue un “0” como instrucción de traza inoperante en la etapa 5 de instrucciones de traza, quedando esta instrucción de traza inoperante en paralelo con la burbuja insertada en la etapa 5 de instrucciones de programa.

En una forma de realización de la invención, el dispositivo de procesamiento utiliza un campo de un registro interno de control para determinar la configuración de su modo de trabajo “TRAZA” o “NOMINAL”. En otra forma de realización de la invención, el dispositivo

de procesamiento utiliza el valor asignado a una señal externa del dispositivo para configurar el modo de trabajo "TRAZA" o "NOMINAL".

5 En un tercer aspecto de la invención se divulga un procesador que comprende un dispositivo de búsqueda, decodificación y procesamiento en paralelo, y de forma sincronizada, de instrucciones de programa e instrucciones de traza según una cualquiera de las formas de realización anteriores para el primer aspecto de la invención.

10 En el presente dispositivo, no se hace uso de una memoria compartida y el mecanismo de traza está basado en la ejecución de códigos de trazado almacenados en palabras de una memoria auxiliar, y no en la ejecución de una instrucción que activa y desactiva la traza por el propio procesador. El procedimiento de la invención propuesta permite controlar de forma independiente el trazado de cada instrucción del código ejecutable ya que los
15 códigos de traza se insertan en palabras que están alineadas con las instrucciones del código ejecutable de las que se desea obtener una traza puntual. El método de la invención propuesta define un procedimiento en tres pasos, que emplea el código fuente, el código objeto, y el archivo de localización de símbolos de un programa, para localizar las posiciones de las palabras de la memoria auxiliar en las que insertar la información de traza, sin alterar el código ejecutable. El dispositivo propuesto en esta invención procesa
20 la información de traza en paralelo con la instrucción alineada que se pretende trazar, lo que da como resultado la escritura en un registro de salida de una palabra de identificación unívoca de la instrucción que se ha ejecutado. Un hardware de análisis captura la palabra de identificación, de forma que queda registrado el instante específico en el que se ejecutó la instrucción trazada. De esta forma, la presente invención, no establece intervalos de
25 instrucciones a trazar, sino que, mediante el control selectivo de la traza de instrucciones ubicadas en cualquier parte del código, permite obtener el tiempo de ejecución en el peor caso de cada una de las funciones de sistema.

30 En la invención que aquí se presenta, las instrucciones que se trazan son definidas mediante un método que permite, de forma selectiva, insertar en una memoria auxiliar, y en una posición alineada a la instrucción que deseamos trazar, la información de traza. El

dispositivo presentado en esta invención procesa la información de traza en paralelo y de forma sincronizada con la instrucción que se pretende trazar.

5 La nueva invención define un método que, a partir del código fuente, el código objeto, y el archivo de localización de símbolos de un programa, y una vez seleccionados los puntos del código fuente a trazar, se determina el contenido de una memoria auxiliar, alineada con la memoria que almacena el código, de forma que quedan marcadas las instrucciones a trazar. La nueva invención, por tanto, no modifica el conjunto de instrucciones del procesador y no requiere insertar instrucciones de traza al código, de forma que incluir
10 códigos de traza no altera el código ejecutable del programa a trazar. Además, la nueva invención propone la configuración del dispositivo de procesamiento en dos modos, uno denominado TRAZA, donde la información de la memoria auxiliar se interpreta como códigos de traza que son ejecutados en paralelo, y otra NOMINAL, que utiliza la memoria auxiliar para contener códigos de detección y corrección de errores.

15

Dado que la aplicación de la instrumentación de código empleando los procesadores actuales introduce sobrecarga en el tiempo de ejecución, la invención presentada, destinada a eliminar estas sobrecargas, aporta una mejora con un objetivo concreto enmarcado en este ámbito. La nueva invención evita el incremento del espacio ocupado
20 por el código ejecutable, ya que no se instrumenta el código para marcar las instrucciones a trazar, y la necesidad de añadir nuevos códigos de operación al procesador.

BREVE DESCRIPCIÓN DE LAS FIGURAS

25 Para complementar la descripción de la invención y con objeto de ayudar a una mejor comprensión de sus características, se acompaña, como parte integrante de dicha descripción, un juego de dibujos en donde con carácter ilustrativo y no limitativo, se ha representado lo siguiente:

30 Figura 1. Método de generación automática del contenido de una memoria auxiliar, alineada con la memoria de programa, que contenga los códigos de traza de aquellas

instrucciones seleccionadas para ser trazadas.

Figura 2. Estructura del dispositivo para la búsqueda, decodificación y ejecución paralela de instrucciones de programa e instrucciones de traza propuesta en la invención. En la
 5 figura 2 se referencian los elementos de este dispositivo, que son los siguientes:

	100	Etapa 1 de búsqueda de instrucciones
	101	Memoria de programa que contiene las instrucciones a leer por la etapa de búsqueda
10	102	Memoria auxiliar que contiene los códigos a interpretar por el dispositivo
	103	Módulo principal de la etapa de búsqueda
	104	Instrucción leída por el módulo principal de la etapa de búsqueda
	105	Código leído por el módulo principal de la etapa de búsqueda
15	106	Dirección de la instrucción leída por el módulo principal de la etapa de búsqueda
	107	ESPERA: Entrada al controlador de la ruta de datos que monitoriza la espera en la búsqueda de instrucciones del módulo principal de la etapa de búsqueda
20	108	NR_ERROR: Entrada al controlador de la ruta de datos que monitoriza la detección de un error no recuperable en la instrucción leída por el módulo principal de la etapa de búsqueda
	109	Módulo de cálculo de la dirección de la instrucción a leer en la etapa de búsqueda
25	110	Multiplexor de entrada al registro que almacena la dirección de la siguiente instrucción a buscar por la etapa de búsqueda.
	111	Registro que contiene la dirección de la instrucción a buscar por la etapa de búsqueda
30	112	LD_DIR: Señal que controla el almacenamiento en el registro que contiene la dirección de la siguiente instrucción a buscar por la etapa de búsqueda.

	113	SEL_DIR: Salida del controlador de la ruta de datos que controla el multiplexor de entrada al registro que contiene la dirección de la siguiente instrucción a buscar por la etapa de búsqueda.
5	114	Etapa de decodificación simultánea de una instrucción de programa y una instrucción de traza
	115	Módulo de decodificación de instrucciones de programa de la etapa 2
	116	Módulo de decodificación de instrucciones de traza de la etapa 2
10	117	Instrucción de programa cargada en el módulo de decodificación de instrucciones de programa de la etapa 2
	118	Instrucción de traza cargada en el módulo de decodificación de instrucciones de traza de la etapa 2
	119	Controlador de la ruta de datos del dispositivo
	120	Entradas al controlador de la ruta de datos
15	121	Salidas del controlador de la ruta de datos
	122	LD_INS: Salida del controlador de la ruta de datos que controla la carga del módulo de decodificación de instrucciones de programa de la etapa 2
20	123	LD_TRZ: Salida del controlador de la ruta de datos que controla la carga del módulo de decodificación de instrucciones de traza la etapa 2
	124	Multiplexor de selección de la entrada del módulo de decodificación de instrucciones de traza la etapa 2
25	125	SEL_TE2: Salida del controlador de la ruta de datos que controla el multiplexor de entrada al módulo de decodificación de instrucciones de traza la etapa 2
	126	Etapa 3 del <i>pipeline</i> de las instrucciones del programa
	127	Etapa 3 del <i>pipeline</i> de las instrucciones de traza
	128	Etapa 4 del <i>pipeline</i> de las instrucciones del programa
30	129	Etapa 4 del <i>pipeline</i> de las instrucciones de traza
	130	Etapa 5 del <i>pipeline</i> de las instrucciones del programa

	131	Etapa 5 del <i>pipeline</i> de las instrucciones de traza
	132	<i>Pipeline</i> de las instrucciones del programa
	133	<i>Pipeline</i> de las instrucciones de traza
	134	Registro de salida de la información de traza
5	135	TE5_DIRI_ES_0: Señal que monitoriza si la etapa 5 de <i>pipeline</i> de instrucciones de traza almacena un valor cero en el campo de dirección
	136	LD_TR_OUT: Señal que controla el almacenamiento en el registro de salida de la información de traza. Toma el valor complementario a la señal TE5_DIRI_ES_0, por lo que el registro sólo se carga cuando el campo de dirección de la información de traza es distinto de cero.
10		
	137	Multiplexor de selección de la entrada de la etapa 3 del <i>pipeline</i> de instrucciones de traza
15	138	Multiplexor de selección de la entrada de la etapa 4 del <i>pipeline</i> de instrucciones de traza
	139	Multiplexor de selección de la entrada de la etapa 5 del <i>pipeline</i> de instrucciones de traza
	140	SEL_TE3: Salida del controlador de la ruta de datos que controla el multiplexor de entrada a la etapa 3 del <i>pipeline</i> de instrucciones de traza
20		
	141	SEL_TE4: Salida del controlador de la ruta de datos que controla el multiplexor de entrada a la etapa 4 del <i>pipeline</i> de instrucciones de traza
25	142	SEL_TE5: Salida del controlador de la ruta de datos que controla el multiplexor de entrada a la etapa 4 del <i>pipeline</i> de instrucciones de traza
	143	BURBUJA_E2: Señal de entrada al controlador que monitoriza la inserción de una burbuja en la etapa 2 de decodificación de instrucciones de programa
30		
	144	BURBUJA_E3: Señal de entrada al controlador que monitoriza la

- 5
- 145 inserción de una burbuja en la etapa 3 de *pipeline* de instrucciones de programa
- BURBUJA_E4: Señal de entrada al controlador que monitoriza la inserción de una burbuja en la etapa 4 de *pipeline* de instrucciones de programa
- 146 BURBUJA_E5: Señal de entrada al controlador que monitoriza la inserción de una burbuja en la etapa 5 de *pipeline* de instrucciones de programa
- 10 147 Información de traza generada por el pipeline de instrucciones de traza tras la ejecución de la instrucción de traza
- 148 Entrada al módulo de cálculo de la dirección de la instrucción a leer en la etapa de búsqueda corresponde a la dirección de salto
- 149 Campo dirección de la información obtenida como resultado de la ejecución de una instrucción de traza.

15

DESCRIPCIÓN DE UNA FORMA DE REALIZACIÓN DE LA INVENCION

20 La realización de la invención estará basada en la implementación del método, descrito en la figura 1, que permite obtener el contenido de una memoria auxiliar, formado por códigos de traza y códigos de no operación, que habilita la traza selectiva de un programa almacenado en una memoria de programa alineada con la memoria auxiliar, así como la construcción de un dispositivo, descrito en la figura 2, con dos modos de operación, denominados "TRAZA" y "NOMINAL", y que en su modo "TRAZA" es capaz de buscar, decodificar y ejecutar los códigos de traza en paralelo con las instrucciones del programa,

25 sin introducir ninguna sobrecarga en el tiempo de ejecución del programa, mientras que en modo "NOMINAL" el dispositivo es capaz de interpretar cada palabra de la memoria auxiliar alineada con una instrucción de la memoria de programa, como un código de detección y corrección de errores en el almacenamiento de esa instrucción.

30 El método y el dispositivo en su modo "TRAZA" se pueden utilizar para realizar un análisis híbrido de programas. Este análisis combina el análisis estático del código del programa,

con información obtenida durante la ejecución de este programa sobre la plataforma de despliegue, con el objetivo de inferir propiedades del programa, tales como el tiempo de “ejecución en caso peor” de cada una de las funciones del programa. El análisis estático determina qué sentencias es necesario trazar y, mediante el método de trazado selectivo de instrucciones de programa descrito en la invención, se obtiene el contenido de la memoria auxiliar que contiene los códigos de traza correspondientes a las instrucciones seleccionadas para la traza.

El método de trazado selectivo de instrucciones de programa consta de un procedimiento en tres pasos (1, 2, 3). Para ejecutar este procedimiento se requieren los archivos fuente pre-procesados (4) del programa, sus archivos objeto con información de depuración (5), y el archivo de localización de símbolos (6) obtenido durante el enlazado (7) destinado a generar el archivo ejecutable (13). El proceso permite obtener el contenido de una memoria auxiliar (12), consistente en códigos de traza (17) y en códigos de no operación (18), alineados con el contenido de la memoria de programa (13) donde se encuentran las instrucciones a trazar.

El primer paso del procedimiento (1) consiste en la selección de las sentencias que se van a trazar sobre los archivos fuente pre-procesados del programa (4), siendo los elementos que caracterizan a cada sentencia seleccionada (10): el nombre del archivo fuente (15), el número de línea (8) de la sentencia dentro de ese archivo, y el tipo de traza (9) que se pretende realizar. El procedimiento permite distintos tipos de traza, siendo el tipo básico aquél que sólo traza el instante de ejecución de la sentencia, mientras que el resto de tipos no básicos permiten trazar diferentes elementos del estado interno del procesador, como, por ejemplo, el contenido concreto de un registro.

El segundo paso del procedimiento (2) consiste en utilizar los archivos objeto con información de depuración del programa (5) para identificar de forma automática, y para cada sentencia seleccionada (10) en el primer paso (1), la dirección relativa (14) de la instrucción a trazar (11) que completa la ejecución de dicha sentencia.

El tercer y último paso del procedimiento (3) utiliza, para cada sentencia seleccionada (10) en el primero paso (1), los elementos que caracterizan dicha selección, que son: nombre del archivo (15), número de línea (8) y tipo de traza (9). La información aportada por estos elementos, junto con la dirección relativa (14), dentro del archivo objeto, de la instrucción a trazar (11) identificada en el segundo paso (2), se emplea para realizar una búsqueda sobre el archivo de localización de símbolos (6), y de esa forma, poder obtener de forma automática el contenido de una memoria auxiliar (12) que ubica los códigos de traza (17) de cada sentencia seleccionada (10) en el primer paso (1), siendo el desplazamiento (16) de cada código de traza (17) dentro del contenido de la memoria auxiliar (12) el mismo que el de la instrucción a trazar (11), identificada en el segundo paso del procedimiento (2), dentro del contenido de la memoria de programa (13). Aquellas posiciones de la memoria auxiliar que tienen el mismo desplazamiento que las instrucciones que no han sido seleccionadas para la traza son completadas con códigos de no operación NOP (18), obteniendo de esta forma un contenido de la memoria auxiliar (12) que mantiene alineados los códigos de traza (17) de las sentencias seleccionadas para la traza (10), con las instrucciones a trazar (11) que completan esas sentencias, que forman parte del contenido de la memoria de programa (13).

El dispositivo propuesto en la invención consta de una estructura interna que permite buscar, decodificar y ejecutar en paralelo instrucciones de programa e instrucciones de traza, de forma que el trazado de las instrucciones de un programa no sea intrusivo ni en el tiempo de ejecución ni en la ubicación del programa en la memoria de programa, ya que es una memoria auxiliar la que se utiliza para ubicar los códigos de traza. De esta forma la dirección de almacenamiento, la secuencia y el instante de ejecución de las instrucciones del programa bajo análisis no se ven modificados por la introducción de las trazas.

Los elementos principales de la estructura del dispositivo propuesto son los siguientes: una etapa de búsqueda (100), que está formada por un módulo de cálculo de la dirección de la instrucción (109) y un módulo principal de la etapa de búsqueda (103); una etapa de decodificación (114), que tiene un módulo de decodificación de instrucciones de programa

(115) y un módulo de decodificación de instrucciones de traza (116); un *pipeline* para las instrucciones de programa (132); un *pipeline* específico para las instrucciones de traza (133); un registro de salida (134), que recibe la información de traza obtenida tras la ejecución de las instrucciones de traza; una ruta de datos, formada por un conjunto de multiplexores (110,124,137,138,139); un controlador de la ruta de datos del dispositivo (119), que determina, en función de su estado y del valor de sus entradas (120), el valor de las salidas (121) que controlan tanto los multiplexores (110,124,137,138,139) de las diferentes etapas, como las señales de carga en registros asociados las distintas etapas (111, 115, 116, 126, 127, 128, 129, 130, 131), así como el registro de salida (134). Tanto las entradas (120) como las salidas (121) están representadas gráficamente en la figura 2, junto a la etiqueta asignada para cada señal.

El dispositivo de procesamiento utiliza la etapa de búsqueda (100) para cargar, simultáneamente, una instrucción de programa (104) de la memoria de programa (101) y un código (105) de la memoria auxiliar (102), estando el código alineado a la instrucción; lo que significa que tiene el mismo desplazamiento dentro de la memoria auxiliar (102) que la instrucción de programa (104) dentro de la memoria de programa (101).

El dispositivo de procesamiento, en su modo NOMINAL, interpreta el código (105) dentro del módulo principal de la etapa de búsqueda (103) como un código de detección y corrección de errores, de forma que la instrucción de programa (104) leída de la memoria de programa (101) pueda ser corregida en caso de error recuperable, y se suministra sin errores en el ciclo siguiente como una instrucción válida de programa (117) al decodificador de instrucciones de programa (115) de la etapa de decodificación (114). En caso de que el error no sea recuperable, el módulo principal (103) de la etapa de búsqueda (100) activará una señal (108) que será interpretada por el procesador como una excepción.

El dispositivo de procesamiento, en su modo "TRAZA", combina el código (105) leído de la memoria auxiliar (102) con la dirección (106) correspondiente a la instrucción de programa (104) leída de la memoria de programa (101), con el fin de construir una

instrucción de traza (118) que en los ciclos siguientes es decodificada y ejecutada en paralelo con la instrucción de programa (117).

5 En el modo "TRAZA" del dispositivo de procesamiento, el módulo de decodificación de instrucciones de programa (115), correspondiente a la etapa de decodificación (114), decodifica la instrucción de programa (117) y, en paralelo, el módulo de decodificación de instrucciones de traza (116) decodifica la instrucción de traza (118), cuyo código de traza estaba almacenado en la memoria auxiliar (102) en una posición alineada con la que mantenía la instrucción de programa (117) en la memoria de programa (101).

10

En el modo "TRAZA" del dispositivo de procesamiento, el controlador de la ruta (119) utiliza los valores de las señales "ESPERA" (107), "BURBUJA_E2" (143), "BURBUJA_E3" (144), "BURBUJA_E4" (145), "BURBUJA_E5" (146) para determinar la ruta que seguirán las instrucciones de traza hacia las siguientes etapas. El controlador configura los multiplexores (124, 137, 138 y 139) de la ruta para asegurar que una instrucción de traza se ejecuta de forma sincronizada con la instrucción de programa alineada, haciéndose efectiva dicha ejecución durante la última etapa (131) del *pipeline* de traza (133), en el que se comprueba que la señal "TE5_DIR_ES_0" (135) está desactivada, en cuyo caso la señal "LD_TR_OUT" se activa (136) y la información de traza es dirigida hacia el registro de salida (134).

15

20

La información de traza almacenada en el registro de salida (134) del dispositivo puede ser analizada mediante un hardware de soporte y un analizador lógico, de forma que sea posible capturar el instante de ejecución de las instrucciones de programa trazadas, así como el resultado de la ejecución de la instrucción de traza que corresponde con el campo "TR" (146) de la información de traza. La información de traza identifica unívocamente la instrucción de programa trazada gracias al campo DIRI (149) que fue asignado en la etapa de búsqueda al valor de la dirección (106) de la instrucción de programa (104) leída.

25

30 En el modo "TRAZA" del dispositivo de procesamiento, el controlador de la ruta de datos (119) monitorizará la señal "BURBUJA_E2" (143) para detectar la inserción de una burbuja

en el módulo de decodificación de instrucciones (115) de la etapa de decodificación (114), y fijará la señal "SEL_TE2"(125) al mismo valor que la señal "BURBUJA_E2" (143), de forma que en caso de insertarse la burbuja, el multiplexor de entrada (124) al módulo de decodificación de instrucciones de traza (116) de la etapa de decodificación (114) enrute un 0 como entrada al módulo de decodificación de instrucciones de traza (116); y en caso contrario lo que se enrute sea la instrucción de traza formada por el código (105) y la dirección de la instrucción a trazar (106), proveniente del módulo principal (103) de la etapa anterior de búsqueda (100).

10 En el modo "TRAZA" del dispositivo de procesamiento, el controlador de la ruta de datos (119) monitorizará la señal "BURBUJA_E3" (144) para detectar la inserción de una burbuja en la etapa 3 (126) del pipeline de instrucciones (132), y fijará la señal "SEL_TE3"(140) al mismo valor que la señal "BURBUJA_E3" (144) de forma que, en caso de insertarse la burbuja, el multiplexor de entrada (137) a la etapa 3 (127) del pipeline de traza (133) enrute un 0 como entrada a esa etapa (127); y en caso contrario, lo que se enrute sea la instrucción de traza (118) proveniente de la etapa anterior de decodificación (116).

20 En el modo "TRAZA" del dispositivo de procesamiento, el controlador de la ruta de datos (119) monitorizará la señal "BURBUJA_E4" (145) para detectar la inserción de una burbuja en la etapa 4 (128) del pipeline de instrucciones (132) y fijará la señal "SEL_TE4"(141) al mismo valor que la señal "BURBUJA_E4" (145), de forma que, en caso de insertarse la burbuja, el multiplexor de entrada (138) a la etapa 4 (129) del pipeline de traza (133), enrute un 0 como entrada a esa etapa (129); y en caso contrario, lo que se enrute hacia la etapa 4 (129) sea la salida proveniente de la etapa anterior (127).

25 En el modo "TRAZA" del dispositivo de procesamiento, el controlador de la ruta de datos (119) monitorizará la señal "BURBUJA_E5" (146) para detectar la inserción de una burbuja en la etapa 5 (130) del pipeline de instrucciones (132) y fijará la señal "SEL_TE5"(142) al mismo valor que la señal "BURBUJA_E5" (146), de forma que, en caso de insertarse la burbuja, el multiplexor de entrada (139) a la etapa 5 (131) del pipeline de traza (133), enrute un 0 hacia el campo DIRI (149) de esa etapa (131); y en caso contrario, lo que se

enrute hacia la etapa 5 (131) sea la salida proveniente de la etapa anterior (129).

5 El dispositivo de procesamiento, independientemente de que esté configurado en modo “TRAZA”, o en modo “NOMINAL”, en el caso de hacerse efectiva una instrucción de salto en una etapa del pipeline de instrucciones (132), insertará en el ciclo siguiente una burbuja tanto en esa etapa como en las etapas anteriores del pipeline de instrucciones (132), y fijará la señal “SEL_DIR” (113) a 0, para que el módulo de cálculo de dirección (109) tome el valor de la dirección efectiva de salto localizada en NEXT_PC (148).

10 El dispositivo de procesamiento, independientemente de que esté configurado en modo “TRAZA” o en modo “NOMINAL”, en el caso de que su etapa de búsqueda (100) detecte que la instrucción direccionada en la memoria de programa (101), cuya dirección es proporcionada por el registro “DIR” (111), no está disponible en un único ciclo y se requieran uno o más ciclos de espera para completarse la lectura, activará la señal
15 “ESPERA” tantos ciclos como ciclos de espera requiera la lectura; requiriendo cada ciclo de espera la inserción de una burbuja en la etapa de decodificación (114).

La realización física preferida tendrá por un lado una parte “Software”, correspondiente al programa que realiza el método de obtención del contenido de la memoria auxiliar descrito
20 en la figura 1, y otra parte de implementación “Hardware/Firmware” de la funcionalidad descrita del dispositivo de procesamiento representado en la figura 2, partiendo de un modelo de descripción de una arquitectura estándar de procesador que utilice un módulo auxiliar de memoria para la detección y corrección de errores sobre el que se realizarán las mencionadas modificaciones, que afectan, básicamente, a la posibilidad de trabajar en
25 dos modos; uno denominado “NOMINAL”, que interpreta el contenido de la memoria auxiliar como códigos de detección y corrección de errores, y otro, denominado “TRAZA”, que mediante la modificación del diseño de las etapas de búsqueda y decodificación, y la incorporación del *pipeline de traza*, habilita la búsqueda, decodificación y ejecución en paralelo de las instrucciones de programa con los códigos de traza almacenados en la
30 memoria auxiliar, calculados mediante el “Software” que implementa el método de obtención del contenido de la memoria auxiliar descrito en la figura 1. Dichos modelos de

descripción de arquitecturas van a permitir generar los detalles de fabricación del dispositivo, que podrá ser materializado sobre un dispositivo programable como una FPGA (Matriz de Puertas Programables, *Field Programmable Gate Array*) o bien sobre un Circuito Integrado de Aplicación Específica (ASIC, *Application Specific Integrated Circuit*).

5

Hay diferentes opciones de realización. Todas ellas parten del modelo VHDL de un "IP Core" de un procesador segmentado, como ARM, LEON o RISC V, sobre el que se modificará la implementación de la estructura del *pipeline* del dispositivo para incluir la funcionalidad descrita en esta patente. El objetivo es generar un nuevo "IP Core", que podrá ser fabricado sobre FPGA o ASIC.

10

En el modo de trabajo tipo "TRAZA", la etapa de búsqueda (100) lee simultáneamente una instrucción de programa (104) de la memoria de programa (101) y un código (105) de la memoria auxiliar (102), y realiza en el módulo principal (103) de la etapa de búsqueda (100) la combinación del código (105) leído de la memoria auxiliar (102) con la dirección (106) correspondiente a la instrucción de programa (104) leída de la memoria de programa (101), para construir una instrucción de traza (118) que en los ciclos siguientes es decodificada y ejecutada en paralelo y de forma sincronizada con la instrucción de programa (117), empleando para ello un *pipeline* de instrucciones de traza (133) que genera como resultado de la ejecución un valor resultado TR(147) que será almacenado en la última etapa "TE5" del *pipeline* de instrucciones de traza (133);

15

20

donde el dispositivo de procesamiento en su modo "TRAZA" emplea un decodificador de instrucciones de traza (116) presente en la etapa de decodificación (114), que alimenta al *pipeline* de instrucciones de traza (133);

25

donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA, y en el caso de que el decodificador de instrucciones de traza (116) detecte un código de no operación "NOP" en la instrucción de traza (118), fija la ruta de datos del pipeline de instrucciones de traza (133) de forma que su etapa "TE3" (127) reciba un 0 y no cargue una instrucción válida de traza, poniendo para ello un valor 1 en la señal "SEL_TE3" (140) que controla el multiplexor (137);

30

donde el *pipeline* de instrucciones de traza (133), estando el dispositivo de procesamiento

en modo TRAZA, almacena en su última etapa "TE5" (131) la información de traza obtenida tras la ejecución de una instrucción de traza (116), estando formada esta información por la dirección de la instrucción de programa "DIRI" (149) que se ejecuta en paralelo y el dato obtenido tras la ejecución de la instrucción de traza "TR" (147). En el caso de que no se hubiera ejecutado una instrucción de traza válida en paralelo con la instrucción de programa, el valor del campo "DIRI" (149) almacenado en "TE5" (131) será igual a 0;

donde el *pipeline* de instrucciones de traza (133), estando el dispositivo de procesamiento en modo TRAZA, carga en el registro de salida (134) esa información siempre que el campo el campo "DIRI" (149) almacenado en "TE5" (131) sea distinto de 0, de acuerdo al valor de la señal "TE_DIRI_ES_0" (135);

donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA, y en el caso de detectar la activación de la señal "BURBUJA_E2" (143), que corresponde a la inserción de una burbuja en el módulo de decodificación de instrucciones de programa (115), fija a 1 la señal de selección "SEL_TE2" del multiplexor de entrada (124) al módulo de decodificación de la instrucción de traza (116), y de esa forma, y en paralelo a la burbuja insertada en el módulo de decodificación de instrucciones de programa (115), se cargue una instrucción de traza no operante en el módulo de decodificación de la instrucción de traza (116);

donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E3" (144), que corresponde a la inserción de una burbuja en la etapa 3, "E3" (126) del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE3" (140) del multiplexor de entrada (137) a la etapa 3, "TE3" (127), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 3, "E3" (126), del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en la etapa 3, "TE3" (127), del *pipeline* de instrucciones de traza (133);

donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E4" (145) que corresponde a la inserción de una burbuja en la etapa 4, "E4" (128), del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE4" (141) del

multiplexor de entrada (138) a la etapa 4, "TE4" (129), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 4, "E4" (128), del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en la etapa 4, "TE4" (129), del *pipeline* de instrucciones de traza (133);

- 5 donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E5" (146) que corresponde a la inserción de una burbuja en la etapa 5, "E5" (130), del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE5" (142) del multiplexor de entrada (139) a la etapa 5, "TE5" (131), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 5, "E5" (130),
- 10 del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en la etapa 5, "TE5" (131), del *pipeline* de instrucciones de traza (133).

REIVINDICACIONES

1.- Un método de trazado selectivo de instrucciones de programa que, dado un programa del que se proveen unos archivos fuente pre-procesados (4), unos archivos objeto con información de depuración (5), y un archivo de localización de símbolos (6) generado durante el enlazado (7), comprende los siguientes pasos (1, 2, 3):

1. seleccionar las sentencias que se van a trazar sobre los archivos fuente pre-procesados del programa (4), siendo los elementos que caracterizan a cada sentencia: el nombre del archivo fuente (15), el número de línea (8) de la sentencia dentro de ese archivo, y el tipo de traza (9), básico si sólo traza el instante de ejecución de la sentencia o no básico si traza además el valor de algún elemento del estado interno del procesador, que se pretende realizar;

2. identificar de forma automática, para cada sentencia seleccionada (10), la dirección relativa (14) de la instrucción a trazar (11) que completa la ejecución de dicha sentencia, utilizando los archivos objeto con información de depuración del programa (5);

3. obtener de forma automática el contenido de una memoria auxiliar (12), que comprende códigos de traza (17) y códigos de no operación (18), utilizando los elementos que caracterizan a cada sentencia seleccionada (10), la dirección relativa (14) identificada dentro del archivo objeto de la instrucción a trazar que se corresponde con la sentencia seleccionada (10), y la información del archivo de localización de símbolos (6), obtenido durante el enlazado (7) de los archivos objeto del programa (5) que genera el archivo ejecutable (13); con el fin de obtener, para cada sentencia seleccionada (10), el código de traza (17) y su desplazamiento dentro de la memoria auxiliar que lo almacena quedando alineados los códigos de traza (17) con las instrucciones a trazar (11) del contenido de la memoria de programa(13); donde alineados significa que el desplazamiento (16) de cada código de traza (17) dentro del contenido de la memoria auxiliar (12) es el mismo que el de la instrucción a trazar (11) dentro del contenido de la memoria de programa (13), y aquellas posiciones de la memoria auxiliar que tienen el mismo desplazamiento que las instrucciones de programa que no han sido seleccionadas

para la traza son completadas con códigos de no operación NOP (18).

2.- Un dispositivo de procesamiento capaz de realizar la búsqueda, decodificación y ejecución en paralelo de instrucciones de programa y códigos de traza obtenidos a partir del método de la reivindicación 1, que comprende dos modos de trabajo "TRAZA" y NOMINAL, caracterizado porque comprende:

- una etapa de búsqueda (100), que a su vez comprende:
 - un módulo de cálculo de la dirección de la instrucción (109),
 - un módulo principal de búsqueda (103), capaz de realizar una lectura simultánea de una instrucción (104) de la memoria de programa (101) y de un código (105) de una memoria auxiliar (102),
- una etapa de decodificación (114) que a su vez comprende:
 - un módulo de decodificación de instrucciones de programa (115),
 - un módulo de decodificación de instrucciones de traza (116),
- un *pipeline* de instrucciones de programa (132),
- un *pipeline* de instrucciones de traza (133),
- un registro de salida (134) que recibe la información de traza obtenida tras la ejecución de una instrucción de traza,
- una ruta de datos que comprende un conjunto de multiplexores (110,124,137,138,139),
- un controlador de la ruta de datos, que tiene un conjunto de entradas (120), formadas por las señales (107, 108, 135, 143, 144, 145, 146) calculadas en las diferentes etapas, y genera un conjunto de salidas (121), que controlan los multiplexores (110,124,137,138,139) y la carga en los registros asociados a las distintas etapas (111, 115, 126, 127, 128, 129, 130, 131), así como la carga en el registro de salida (134) de la información de traza obtenida tras la ejecución de una instrucción de traza,

3.- El dispositivo de procesamiento de acuerdo con la reivindicación 2, donde en el modo de trabajo tipo "NOMINAL", la etapa de búsqueda (100) lee simultáneamente una instrucción de programa (104) de la memoria de programa (101) y un código (105) de

la memoria auxiliar (102), e interpreta dicho código (105) dentro del módulo principal de la etapa de búsqueda (103) como un código de detección y corrección de errores; de forma que la instrucción (104) leída de la memoria de programa (101) pueda ser corregida, en caso de error recuperable, y se suministra sin errores en un ciclo subsiguiente como una instrucción válida de programa (117) al decodificador de instrucciones de programa (115) de la etapa de decodificación (114) y, en caso de que el error no sea recuperable, el módulo principal (103) de la etapa de búsqueda (100) activa una señal (108) que es interpretada por el procesador como una excepción.

5

10

4.- El dispositivo de procesamiento de acuerdo con la reivindicación 2, donde, en el modo de trabajo tipo "TRAZA", la etapa de búsqueda (100) lee simultáneamente una instrucción de programa (104) de la memoria de programa (101) y un código (105) de la memoria auxiliar (102), y realizar en el módulo principal (103) de la etapa de búsqueda (100) la combinación del código (105) leído de la memoria auxiliar (102) con la dirección (106) correspondiente a la instrucción de programa (104) leída de la memoria de programa (101), para construir una instrucción de traza (118) que en los ciclos siguientes es decodificada y ejecutada en paralelo y de forma sincronizada con la instrucción de programa (117), empleando para ello un *pipeline* de instrucciones de traza (133) que genera como resultado de la ejecución un valor resultado (TR, 147) que será almacenado en la última etapa (TE5) del *pipeline* de instrucciones de traza (133); y emplea el módulo de decodificación de instrucciones de traza (116), presente en la etapa de decodificación (114), que alimenta al *pipeline* de instrucciones de traza (133).

15

20

25

5.- El dispositivo de procesamiento de acuerdo con la reivindicación 4, donde el controlador de la ruta de datos (119), en el caso de que el decodificador de instrucciones de traza (116) detecte un código de no operación "NOP" en la instrucción de traza (118), fija la ruta de datos del pipeline de instrucciones de traza (133) de forma que su etapa "TE3" (127) reciba un 0 y no cargue una instrucción válida de traza, poniendo para ello un valor 1 en las señal "SEL_TE3" (140) que controla el multiplexor (137).

30

6.- El dispositivo de procesamiento de acuerdo con la reivindicación 5, donde el *pipeline*

de instrucciones de traza (133), almacena en su última etapa "TE5" (131) la información de traza obtenida tras la ejecución de una instrucción de traza (116), estando formada esta información por la dirección de la instrucción de programa "DIRI" (149) que se ejecuta en paralelo y el dato obtenido tras la ejecución de la instrucción de traza "TR" (147), y en el caso de que no se hubiera ejecutado una instrucción de traza válida en paralelo con la instrucción de programa, el valor del campo "DIRI" (149) almacenado en "TE5" (131) es igual a 0.

5

10

7.- El dispositivo de procesamiento de acuerdo con la reivindicación 6, donde el *pipeline* de instrucciones de traza (133), carga en el registro de salida (134) esa información siempre que el campo "DIRI" (149) almacenado en "TE5" (131) sea distinto de 0, de acuerdo al valor de la señal "TE_DIRI_ES_0" (135).

15

20

8.- El dispositivo de procesamiento de acuerdo con la reivindicación 7, donde el controlador de la ruta de datos (119), en el caso de detectar la activación de la señal "BURBUJA_E2" (143), que corresponde a la inserción de una burbuja en el módulo de decodificación de instrucciones de programa (115), fija a 1 la señal de selección "SEL_TE2" del multiplexor de entrada (124) al módulo de decodificación de la instrucción de traza (116), y de esa forma, y en paralelo a la burbuja insertada en el módulo de decodificación de instrucciones de programa (115), se cargue una instrucción de traza no operante en el módulo de decodificación de la instrucción de traza (116).

25

30

9.- El dispositivo de procesamiento de acuerdo con la reivindicación 8, donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E3" (144), que corresponde a la inserción de una burbuja en la etapa 3, "E3" (126) del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE3" (140) del multiplexor de entrada (137) a la etapa 3, "TE3" (127), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 3, "E3" (126), del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en

la etapa 3, "TE3" (127), del *pipeline* de instrucciones de traza (133);

- 5 10.- El dispositivo de procesamiento de acuerdo con la reivindicación 9, donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E4" (145) que corresponde a la inserción de una burbuja en la etapa 4, "E4" (128), del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE4" (141) del multiplexor de entrada (138) a la etapa 4, "TE4" (129), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 4, "E4" (128), del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en la etapa 4, "TE4" (129), del *pipeline* de instrucciones de traza (133);
- 10
- 15 11.- El dispositivo de procesamiento de acuerdo con la reivindicación 10, donde el controlador de la ruta de datos (119), estando el dispositivo de procesamiento en modo TRAZA y en el caso de detectar la activación de la señal "BURBUJA_E5" (146) que corresponde a la inserción de una burbuja en la etapa 5, "E5" (130), del *pipeline* de instrucciones de programa (132), fija a 1 la señal de selección "SEL_TE5" (142) del multiplexor de entrada (139) a la etapa 5, "TE5" (131), del *pipeline* de instrucciones de traza (133); y de esa forma, y en paralelo a la burbuja insertada en la etapa 5, "E5" (130), del *pipeline* de instrucciones de programa (132), se cargue una instrucción de traza no operante en la etapa 5, "TE5" (131), del *pipeline* de instrucciones de traza (133).
- 20
- 25 12.- Un procesador caracterizado porque comprende un dispositivo de procesamiento capaz de trabajar en dos modos, que denominamos modo "TRAZA" y modo "NOMINAL" según cualquiera de las reivindicaciones anteriores, donde en modo "NOMINAL" utiliza una memoria auxiliar, cuyas palabras se leen en paralelo con las instrucciones del programa, y son empleadas como códigos de
- 30 detección y corrección de errores de las instrucciones de programa. donde en modo "TRAZA" se utiliza esa misma memoria auxiliar para almacenar códigos

que habilitan la traza de las instrucciones del programa, y que son leídos y ejecutados en paralelo con las instrucciones de programa según cualquiera de las reivindicaciones anteriores.

5

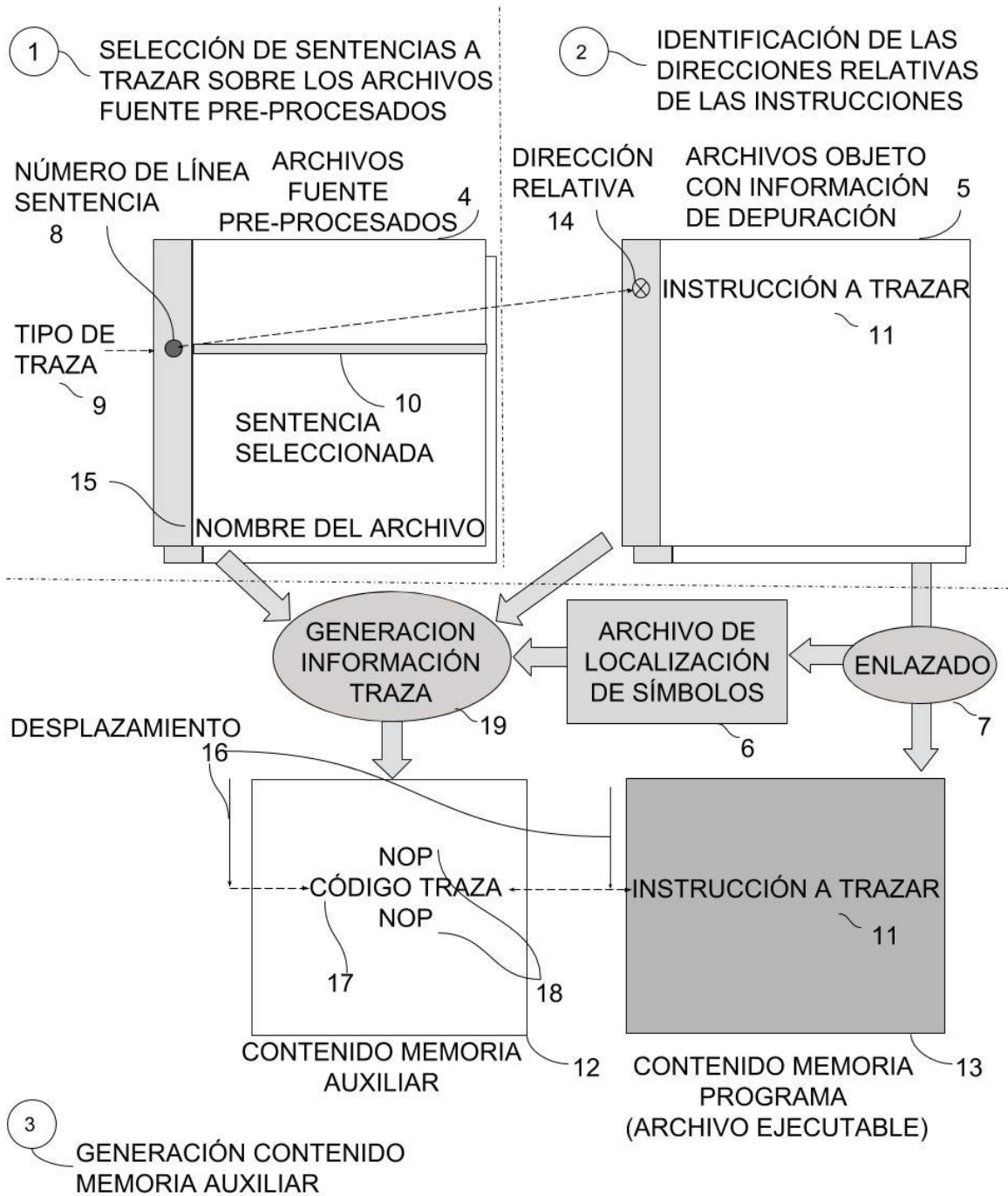


FIG. 1

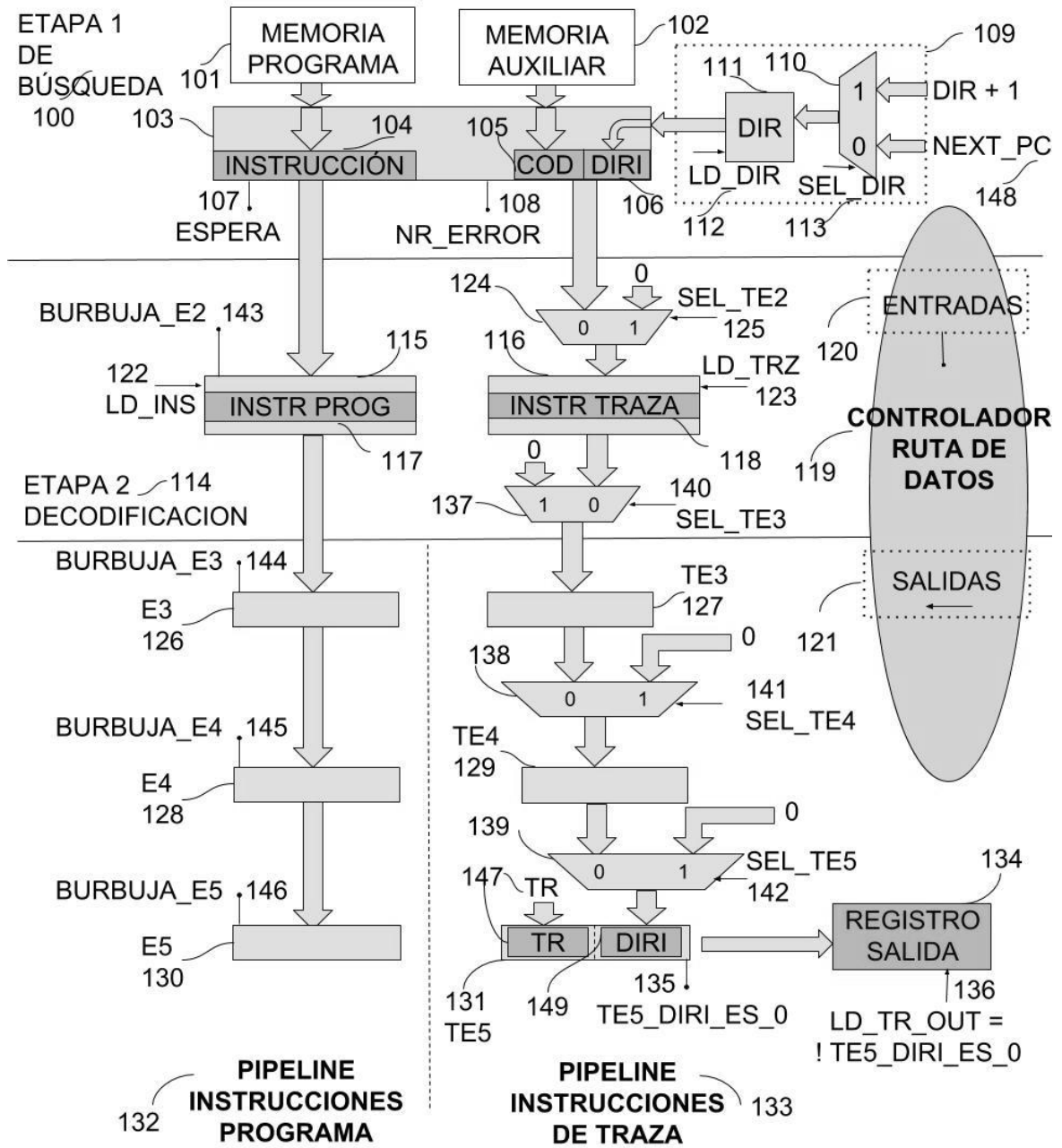


FIG. 2