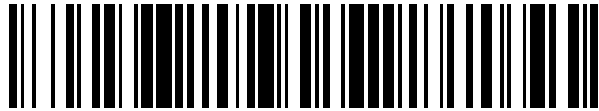


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 697 548**

21 Número de solicitud: 201830266

51 Int. Cl.:

G06F 11/00 (2006.01)

12

PATENTE DE INVENCION CON EXAMEN

B2

22 Fecha de presentación:

20.03.2018

43 Fecha de publicación de la solicitud:

24.01.2019

Fecha de modificación de las reivindicaciones:

09.04.2019

Fecha de concesión:

15.07.2020

45 Fecha de publicación de la concesión:

22.07.2020

73 Titular/es:

**UNIVERSIDAD POLITÉCNICA DE MADRID (15.0%)
Avda. Ramiro de Maeztu, 7
28040 MADRID (Madrid) ES y
UNIVERSIDAD DE ALCALÁ (85.0%)**

72 Inventor/es:

**DA SILVA, Antonio;
RODRIGUEZ POLO, Oscar;
MARTINEZ HELLIN, Agustín;
PARRA ESPADA, Pablo y
SANCHEZ PRIETO, Sebastian**

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

54 Título: **UN MÉTODO Y UN DISPOSITIVO DE PROCESAMIENTO EN PARALELO DE INSTRUCCIONES DE PROGRAMA E INSTRUCCIONES DE TRAZA**

57 Resumen:

Método y dispositivo de sincronización y ejecución paralela de instrucciones de traza sobre un procesador RISC segmentado. La invención consiste en un dispositivo cuya estructura interna, basada en un procesador segmentado, permite eliminar la sobrecarga de tiempo de ejecución que introduce la instrumentación de código empleada para medir el tiempo de ejecución en el peor caso. Para ello el dispositivo utiliza un código de instrucción específico para la instrumentación, que es interpretado como la habilitación de traza de la instrucción que le precede, y que permite identificar unívocamente el momento de ejecución de dicha instrucción. El dispositivo propuesto ejecuta en paralelo, y de forma sincronizada, cada instrucción de traza con la instrucción a trazar que le precede, y condiciona dicha ejecución a que se complete, sin que se vea afectada por burbujas, la ejecución de la instrucción a trazar.

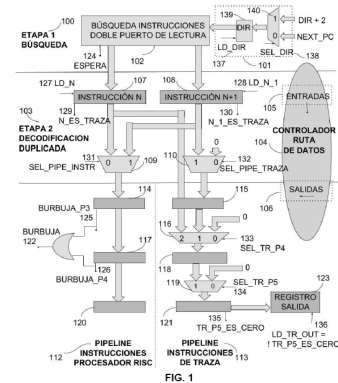


FIG. 1

Aviso: Se puede realizar consulta prevista por el art. 41 LP 24/2015. Dentro de los seis meses siguientes a la publicación de la concesión en el Boletín Oficial de la Propiedad Industrial cualquier persona podrá oponerse a la concesión. La oposición deberá dirigirse a la OEPM en escrito motivado y previo pago de la tasa correspondiente (art. 43 LP 24/2015).

ES 2 697 548 B2

DESCRIPCIÓN

UN MÉTODO Y UN DISPOSITIVO DE PROCESAMIENTO EN PARALELO DE INSTRUCCIONES DE PROGRAMA E INSTRUCCIONES DE TRAZA

5

SECTOR DE LA TÉCNICA

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, Defensa y de alta fiabilidad.

10

ANTECEDENTES DE LA INVENCIÓN

Se han identificado diferentes invenciones que proponen soluciones para facilitar el trazado de instrucciones pero que difieren de la presente invención.

15

La solicitud de patente US 5996092 A, "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. En el presente dispositivo, no hay memoria compartida ni un procesador de traza en paralelo, y la traza está basada en una instrumentación del código que añade, en puntos concretos del código en los que se desea obtener una traza puntual, instrucciones de traza que identifican unívocamente el punto que deseamos trazar, sin recurrir al contador de programa. La instrucción de traza es ejecutada en paralelo con la instrucción previa que se pretende trazar, introduciendo redundancia de las partes necesarias del *pipeline* del procesador, y el resultado de la ejecución es la escritura en un registro de salida donde un hardware de análisis lo captura, de forma que queda registrado el instante específico en el que se ejecutó la instrucción trazada. La presente invención presenta un

20

25

30

enfoque diferente, donde es posible, mediante la instrumentación selectiva de instrucciones ubicadas en cualquier parte del código, obtener el peor tiempo de ejecución de cada una de las funciones de sistema. Este tipo de instrumentación está siendo utilizada por herramientas comerciales como RapiTime en sistemas críticos de aviónica (G. Bernat et al., "Identifying Opportunities for Worst-case Execution Time Reduction in an Avionics System," Ada User Journal, Volume 28, Number 3, 2007, pp. 189–194). Sin embargo, su aplicación utilizando los procesadores disponibles en el mercado tiene como principal desventaja la sobrecarga introducida en el tiempo de ejecución, que se elimina con la invención presentada.

10

Con respecto a la solicitud de patente US 2017147472 (A1), "Systems and methods for a real time embedded trace", la diferencia principal es que el sistema traza las instrucciones de salto, de forma autónoma. En la invención que aquí se presenta, las instrucciones que se trazan son definidas mediante técnicas de instrumentación selectivas, que insertan, a continuación de cada instrucción que deseamos trazar, una instrucción de traza. Lo que hace esta invención es ejecutarlas en paralelo de forma sincronizada con la instrucción que se pretende trazar. Como ya se ha descrito, esta técnica de instrumentación 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 *else*, y el bloque posterior, no implican un salto en la ejecución, por lo que no serían detectados por la solución que presenta la patente US 2017147472 (A1).

15

20

25

30

En cuanto a la patente US 6513134 (B1), "System and method for tracing program execution within a superscalar processor", presenta una mejora respecto a la US 5996092 A , 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

5 *pipes* 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.

10 Por lo tanto, se concluye que 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 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. Dado que la aplicación de la instrumentación de código empleando los
15 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.

DESCRIPCIÓN DE LA INVENCIÓN

20 En un primer aspecto de la invención, se divulga un dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza. El dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza comprende:

- 25
- una etapa de búsqueda de instrucciones, que a su vez comprende:
 - un módulo de cálculo de la dirección de la instrucción; y,
 - módulo de búsqueda de las instrucciones con doble puerto de lectura;
 - una etapa de decodificación duplicada;
 - un *pipeline-traza* (pipeline de traza) para el procesado únicamente de
30 instrucciones de traza;
 - un registro de salida para la traza;
 - una ruta de datos que a su vez comprende un conjunto de multiplexores;

- un controlador de la ruta de datos, que a su vez comprende unas entradas y unas salidas que controlan los multiplexores, la carga en unos registros asociados las distintas etapas y el registro de salida para la traza;

5 donde el controlador de la ruta de datos está configurado para determinar, en función del estado de dicho controlador 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 una instrucción de traza se ejecuta de forma sincronizada con la instrucción que la precede, haciéndose efectiva dicha ejecución durante la última etapa del *pipeline* de traza.

10 En una realización de la invención, el controlador comprende las siguientes secuencias de instrucciones S1, S2 y S3:

S1: corresponde a pares Instrucción-Traza en los que las instrucciones a ser trazadas siempre se cargan en el elemento "INSTRUCCIÓN N", mientras las instrucciones correspondientes de traza se cargan en el elemento "INSTRUCCIÓN N+1";

15 S2: corresponde a una secuencia de instrucciones que no son trazadas, de forma en los elementos "INSTRUCCIÓN N" e "INSTRUCCIÓN N+1" siempre se cargan instrucciones (y no trazas);

20 S3: corresponde a dos pares Instrucción-Traza en los que las instrucciones de traza se cargan en ciclos sucesivos en el elemento "INSTRUCCIÓN N +1", mientras las instrucciones a trazar se cargan en esos mismos ciclos en el elemento "INSTRUCCIÓN N+1".

25 El dispositivo de procesamiento ejecuta la secuencia S1 durante dos ciclos de reloj, T y T+1; de tal forma que las instrucciones almacenadas en unas direcciones X+1, X+3 y X+5, son las instrucciones de traza de las instrucciones que las preceden, ubicadas, respectivamente, en unas direcciones X, X+2 y X+4.

30 El dispositivo de procesamiento ejecuta la secuencia S2 en la que durante dos ciclos no se cargan instrucciones de traza; de tal forma que durante el primer ciclo T se detecta que las dos instrucciones que están cargadas en la etapa de decodificación no son de traza, y por tanto las señales "N_ES_TRAZA" "N_1_ES_TRAZA" valen ambas "0"; y, en el ciclo T +1 el controlador se encuentra en el estado denominado "INSTR PENDIENTE",

en el que la instrucción pendiente ubicada en la segunda unidad de decodificación se dirige hacia la "etapa 3" del *pipeline* de instrucciones del procesador.

5 El dispositivo de procesamiento ejecuta la secuencia S3: en el ciclo T el valor de la señal "N_ES_TRAZA" es "1", mientras que "N_1_ES_TRAZA" vale "0", y el valor de la señal de multiplexación "SEL_TR_P4" es "2", de tal forma que se habilita una ruta donde la instrucción de traza se sincroniza con la ejecución de la instrucción a trazar; en el ciclo T+1, se ubica la instrucción de traza en la etapa 4 del *pipeline* de traza. En el ciclo T, además, la señal de multiplexación "SEL_PIPE_TRAZA" toma el valor 0, con el fin de
10 que en el ciclo T+1 se encuentre un cero en la etapa 3 del *pipeline* de traza.

En una forma de realización de la invención, el dispositivo de procesamiento durante un ciclo "T", detecta una burbuja en la etapa 3 del *pipeline* de instrucciones, de tal forma que el controlador fija la ruta que carga un "0" en las etapas 3 y 4 del *pipeline*- traza y una
15 dirección de salto "Z" se enruta hacia el registro de entrada de la etapa de búsqueda, de tal forma que la detección de la burbuja en la etapa 3 se corresponde con la puesta a "1" de la señal "BURBUJA_P3" y de la señal "BURBUJA". El controlador controla: la ruta hacia la etapa 3 asignando "0" a la señal "SEL_PIPE_TRAZA"; la ruta hacia la etapa 4 asignando un "0" a la señal "SEL_TR_P4"; y la carga del registro de entrada de la etapa
20 de búsqueda activando la señal "LD_DIR" y enrutando la dirección "Z" hacia dicho registro asignando un "0" a la señal "SEL_DIR".

En una forma de realización de la invención, el dispositivo el dispositivo de procesamiento durante un ciclo "T", detecta una burbuja en la etapa 4 del *pipeline* de
25 instrucciones, de tal forma el controlador fija la ruta que carga un "0" en las etapas 3, 4 y 5 del *pipeline*-traza y la dirección de salto "Z" se enruta hacia el registro de entrada de la etapa de búsqueda, de tal forma que la detección de la burbuja en la etapa 4 se corresponde con la puesta a "1" de la señal "BURBUJA_P4" y de la señal "BURBUJA". El controlador controla: la ruta hacia la etapa 3 del *pipeline*-traza asignando "0" a la señal
30 "SEL_PIPE_TRAZA" ; la ruta hacia la etapa 4 asignando un "0" a la señal "SEL_TR_P4"; la ruta hacia la etapa 5 se controla asignando un "0" a la señal "SEL_TR_P5"; y, la carga del registro de entrada de la etapa de búsqueda activando la señal "LD_DIR" y enrutando

la dirección "Z" hacía dicho registro asignando un 0 a la señal "SEL_DIR".

En un segundo aspecto de la invención se divulga un procesador RISC, "Computador con Conjunto de Instrucciones Reducidas", que comprende un dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza según una cualquiera de las formas de realización anteriores para el primer aspecto de la invención.

En un tercer aspecto de la invención se divulga un método de procesamiento en paralelo de instrucciones de programa e instrucciones de traza que, ejecutado sobre un dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza definido en cualquiera de las realizaciones del primer aspecto de la invención, procesa de forma paralela una instrucción y una instrucción de traza.

15 BREVE DESCRIPCIÓN DE LAS FIGURAS

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:

Figura 1. Estructura del dispositivo para la sincronización y ejecución paralela propuesta en la invención.

Figura 2. Máquina de Mealy del controlador de la ruta de datos.

25 Figura 3. Evolución de la Ruta de datos ante la secuencia de instrucciones S1.

Figura 4. Evolución de la Ruta de datos ante la secuencia de instrucciones S2.

Figura 5. Evolución de la Ruta de datos ante la secuencia de instrucciones S3.

Figura 6. Evolución de la Ruta de datos ante una burbuja en la etapa 3 del *pipeline*.

Figura 7. Evolución de la Ruta de datos ante una burbuja en la etapa 4 del *pipeline*.

30 Figura 8. Evolución de la Ruta de datos tras una burbuja en el ciclo anterior.

Figura 9 Tabla de transición de estados del controlador de la ruta de datos.

Figura 10 Tabla de las salidas del controlador de la ruta de datos relativas a la etapa de

búsqueda.

Figura 11 Tabla de las salidas del controlador de la ruta de datos relativas a la etapa de decodificación.

Figura 12 Tabla de las salidas del controlador de la ruta de datos relativas a la etapas 3, 4 y 5 del *pipeline* de las instrucciones de traza.

En la figura 1 se referencian los elementos del dispositivo de sincronización y ejecución paralela propuesta en la invención. Estos elementos son los siguientes:

	100	Etapa 1 de búsqueda de instrucciones
10	101	Módulo de selección de la dirección de la siguiente instrucción
	102	Módulo de búsqueda de instrucciones con doble puerto de lectura
	103	Etapa 2 de decodificación duplicada
	104	Controlador de la ruta de datos del dispositivo
	105	Entradas al controlador de la ruta de datos
15	106	Salidas del controlador de la ruta de datos
	107	Módulo de decodificación "INSTRUCCIÓN N" de la etapa 2
	108	Módulo de decodificación " INSTRUCCIÓN N + 1" de la etapa 2
	109	Multiplexor de selección de la entrada de la etapa 3 del <i>pipeline</i> de instrucciones del procesador RISC
20	110	Multiplexor de selección de la entrada de la etapa 3 del <i>pipeline</i> de las instrucciones de traza
	112	<i>Pipeline</i> de las instrucciones del procesador RISC
	113	<i>Pipeline</i> de las instrucciones de traza
	114	Etapa 3 del <i>pipeline</i> de las instrucciones del procesador RISC
25	115	Etapa 3 del <i>pipeline</i> de las instrucciones de traza
	116	Multiplexor de selección de la entrada de la etapa 4 del <i>pipeline</i> de instrucciones de traza
	117	Etapa 4 del <i>pipeline</i> de las instrucciones del procesador RISC
	118	Etapa 4 del <i>pipeline</i> de las instrucciones de traza
30	119	Multiplexor de selección de la entrada de la etapa 5 del <i>pipeline</i> de instrucciones de traza
	120	Etapa 5 del <i>pipeline</i> de las instrucciones del procesador RISC

	121	Etapa 5 del <i>pipeline</i> de las instrucciones de traza
	122	Entrada al controlador de la ruta de datos que monitoriza la detección de burbujas
	123	Registro de salida de la información de traza
5	124	ESPERA: Entrada al controlador de la ruta de datos que monitoriza la espera en la búsqueda de instrucciones
	125	BURBUJA_P3: Señal de entrada al controlador que monitoriza la detección de una burbuja en la etapa 3 de <i>pipeline</i> de instrucciones del procesador RISC
10	126	BURBUJA_P4: Señal de entrada al controlador que monitoriza la detección de una burbuja en la etapa 4 de <i>pipeline</i> de instrucciones del procesador RISC
	127	LD_N: Salida del controlador de la ruta de datos que controla la carga del módulo de decodificación "INSTRUCCIÓN N" de la etapa 2
15	128	LD_N_1: Salida del controlador de la ruta de datos que controla la carga del módulo de decodificación "INSTRUCCIÓN N + 1" de la etapa 2
	129	N_ES_TRAZA: Señal de entrada al controlador que monitoriza si la instrucción que se ha decodificado en el elemento "Instrucción N" es de tipo traza
20	130	N_1_ES_TRAZA: Señal de entrada al controlador que monitoriza si la instrucción que se ha decodificado en el elemento "Instrucción N+1" es de tipo traza
	131	SEL_PIPE_INSTR: Salida del controlador de la ruta de datos que controla el multiplexor de entrada a la etapa 3 del <i>pipeline</i> de instrucciones del procesador RISC
	132	SEL_PIPE_TRAZA: 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
30	133	SEL_TR_P4: 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
	134	SEL_TR_P5: Salida del controlador de la ruta de datos que controla el multiplexor de entrada a la etapa 5 del <i>pipeline</i> de instrucciones de traza
5	135	TR_P5_ES_CERO: Señal que monitoriza si la etapa 5 de <i>pipeline</i> de instrucciones de traza almacena un valor cero
	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 TR_P5_ES_CERO, por lo que el registro sólo se carga cuando la información de traza es distinta de cero.
10	137	LD_DIR: Señal que controla el almacenamiento en el registro de entrada a la etapa de búsqueda de la dirección de la siguiente instrucción a buscar.
	138	SEL_DIR: Salida del controlador de la ruta de datos que controla el multiplexor de entrada al registro de entrada a la etapa de búsqueda que almacena la dirección de la siguiente instrucción a buscar.
15	139	Registro de entrada a la etapa de búsqueda que almacena la dirección de la siguiente instrucción a buscar.
20	140	Multiplexor de entrada al registro de entrada a la etapa de búsqueda que almacena la dirección de la siguiente instrucción a buscar.

DESCRIPCIÓN DE UNA FORMA DE REALIZACIÓN DE LA INVENCION

25

La invención consiste en un dispositivo dotado de una estructura interna de procesamiento que permite eliminar la sobrecarga de tiempo de ejecución que introduce la instrumentación de código utilizada para medir el tiempo de “ejecución en caso peor” empleando análisis híbrido. Este análisis combina el análisis estático del código con

30 medidas del tiempo de ejecución sobre la plataforma de despliegue. El análisis estático determina qué instrucciones es necesario trazar y, mediante técnicas de instrumentación, añade código de trazado tras cada instrucción que se desea trazar, de

forma que el instante de ejecución de dicha instrucción puede capturarse mediante un hardware de soporte y un analizador lógico. El código añadido tras la instrucción que deseamos trazar permite identificar unívocamente el momento de ejecución de dicha instrucción, pero introduce una sobrecarga que con esta invención se puede eliminar. El dispositivo es capaz de detectar las instrucciones de traza y ejecutarlas en paralelo, de forma sincronizada, y condicionada a la ejecución completa de la instrucción que le precede. De esa forma permite que el proceso de trazado sea no intrusivo en lo que respecta al tiempo de ejecución, ya que la secuencia y el instante de ejecución del programa bajo análisis, no se ven modificados por la introducción de las trazas, al ser éstas ejecutadas en paralelo.

El dispositivo propuesto en la invención utiliza un código de instrucción específico, que será empleado para la instrumentación, y cuya estructura interna interpreta como la habilitación de traza de la instrucción que le precede. Los elementos principales de este dispositivo, que se muestran en la figura 1, son: 1) una etapa de búsqueda de instrucciones (100), que cuenta con un módulo de cálculo de la dirección de la instrucción (101) y módulo de búsqueda de las instrucciones con doble puerto de lectura (102); 2) una etapa de decodificación duplicada (103); 3) un *pipeline* específico para las instrucciones de traza (113); 4) un registro de salida para la traza (123); 5) la ruta de datos, formada por un conjunto de multiplexores (109, 110, 116 y 119); 6) el controlador de la ruta de datos del dispositivo (104), que determina, en función de su estado, y del valor de sus entradas (105), el valor de las salidas (106) que controlan tanto los multiplexores (109, 110, 116 y 119), como la carga en registros asociados las distintas etapas (107, 108, 114, 115, 117, 118, 120 y 121), así como el registro de salida (123). Tanto las entradas (105) como las salidas (106) están representadas gráficamente en la figura 1 junto a la etiqueta asignada para cada señal.

El dispositivo, gracias a la etapa de búsqueda con doble puerto (100), permite cargar dos instrucciones simultáneamente a la etapa de decodificación (103) para que sean decodificadas en paralelo. La señal "ESPERA" (124) de esta etapa se utiliza para modelar posibles estados de espera en dicha búsqueda, y puede activarse tras un *reset* del procesador, o como consecuencia de hacerse efectivo un salto, lo que provoca la

inyección de burbujas en el *pipeline* de instrucciones del procesador (112), monitorizadas por las señales “BURBUJA_P3” (inyección de burbuja en la etapa 3, 125), “BURBUJA_P4” (inyección de burbuja en la etapa 4, 126), y la función OR de éstas, etiquetada como “BURBUJA” (122). La señal “ESPERA” (124) se desactivará para
5 notificar a la etapa de decodificación (103) que las instrucciones están disponibles para ser cargadas.

Las dos instrucciones que se encuentran en la etapa de decodificación corresponden siempre a instrucciones almacenadas en palabras consecutivas de memoria. En la figura
10 1, el elemento etiquetado como “INSTRUCCIÓN N” (107) será el que recibirá la primera de las dos instrucciones, mientras que el etiquetado como “INSTRUCCIÓN N+1” (108) recibirá la siguiente. Los valores N y N+1 no corresponden a direcciones físicas de memoria que sean consecutivas, sino que representan dos instrucciones almacenadas en palabras de memoria consecutivas, sin considerar el tamaño de palabra en bytes del
15 procesador, que en el caso más general de un procesador RISC de 32 bits, sería de 4.

En la etapa de decodificación (103), y como consecuencia de decodificar cada una de las instrucciones, se determina si las instrucciones son de tipo traza o pertenecen al resto del conjunto de instrucciones, calculando para ello las señales etiquetadas en la
20 figura 1 como “N_ES_TRAZA” (127) y “N_1_ES_TRAZA” (128).

El controlador de la ruta (104) utiliza los valores de esas señales, y el de las señales “ESPERA” (124) y “BURBUJA” (122), junto con el estado del propio controlador, para determinar la ruta que seguirán las instrucciones hacia las siguientes etapas. El
25 controlador configura los multiplexores (109, 110, 116 y 119) de la ruta para asegurar que una instrucción de traza se ejecuta de forma sincronizada con la instrucción que la precede, haciéndose efectiva dicha ejecución durante la última etapa (121) del *pipeline* de traza (113), en el que se comprueba que la señal “TR_P5_ES_CERO” (135) está desactivada, en cuyo caso la señal “LD_TR_OUT” se activa (136) y el valor de traza es
30 dirigido hacia el registro de salida (123).

La figura 2 representa la máquina de *Mealy* del controlador de la ruta de datos de este

dispositivo (200), que es especificada formalmente en las tablas de las figuras 9, 10, 11 y 12.

Las figuras 3, 4 y 5 son, respectivamente, ejemplos de cómo el controlador, para hacer efectiva la sincronización, fija la ruta en las siguientes secuencias de instrucciones posibles S1, S2, y S3:

- La secuencia S1 corresponde a pares Instrucción-Traza (301- 302, 303-304 y 305-306) en los que las instrucciones que se desean trazar (301, 303 y 305) siempre se cargan en el elemento "INSTRUCCIÓN N" (107), mientras las instrucciones correspondientes de traza (302, 304 y 306) se cargan en el elemento "INSTRUCCIÓN N+1" (115).
- S2 corresponde a una secuencia de instrucciones (401, 402, 403 y 404) que no son trazadas, de forma en los elementos "INSTRUCCIÓN N" (114) e "INSTRUCCIÓN N+1" (115) siempre se cargan instrucciones y no trazas.
- La secuencia S3 corresponde a dos pares Instrucción-Traza (502- 503 y 504-505) en los que las instrucciones de traza (503 y 505) se cargan en ciclos sucesivos (500 y 510) en el elemento "INSTRUCCIÓN N +1" (107), mientras las instrucciones a trazar (502 y 504) se cargan en esos mismos ciclos en el elemento "INSTRUCCIÓN N+1" (108).

La figura 3 muestra el funcionamiento del dispositivo durante dos ciclos de reloj, T (300) y T+1 (307), en el que el procesador ejecuta la secuencia de instrucciones S1. En la secuencia S1 las instrucciones almacenadas en las direcciones X+1 (302), X+3 (304) y X+5 (306), son las instrucciones de traza de las instrucciones que las preceden, ubicadas, respectivamente, en las direcciones X (301), X+2 (303) y X+4 (305). El esquema muestra en los dos ciclos, T (300) y T+1 (307), cómo las instrucciones de traza (302, 304, 306), añadidas como fruto de la instrumentación, se dirigen hacia las etapas (115 y 118) que pertenecen al *pipeline* de las instrucciones de tipo traza, mientras que las instrucciones a trazar (301, 303, y 305), se dirigen hacia las etapas (114 y 117) que pertenecen al *pipeline* de las instrucciones del procesador. De esta forma se produce una ejecución sincronizada de la instrucción trazada y su instrucción de traza, y se evita la sobrecarga en tiempo de ejecución de insertar instrucciones de traza en un programa, ya que estas se ejecutan en paralelo.

La figura 4 muestra el funcionamiento del dispositivo en la secuencia S2, en la que durante dos ciclos no se cargan instrucciones de traza. En ese caso durante el primer ciclo T (400) se detecta que las dos instrucciones que están cargadas en la etapa de decodificación (403 y 404) no son de traza, y por tanto las señales "N_ES_TRAZA" (129) y "N_1_ES_TRAZA" (130) valen ambas 0. Esta situación conduce a que la etapa de decodificación (103) no cargue dos nuevas instrucciones al inicio del ciclo T + 1 (408), ya que en el ciclo T (400) los valores de "LD_N" (127) y "LD_N_1" (128) que controlan dicha carga son ambos 0. En el ciclo T + 1 (408) el controlador se encuentra en el estado denominado "INSTR PENDIENTE" (202), en el que la instrucción pendiente (404) ubicada en la segunda unidad de decodificación (108) se dirige hacia la etapa 3 del *pipeline* de instrucciones del procesador (114). En los dos ciclos, T (400) y T + 1 (408), el controlador carga la etapa 3 del *pipeline* de las instrucciones de traza (115) con valores 0, por lo que las instrucciones no serán trazadas.

15

En la figura 5 se muestra el funcionamiento del dispositivo para la secuencia S3, que cubre el caso en el que en el ciclo T (500) la instrucción a trazar (502) se encuentra en la etapa 3 del *pipeline* de instrucciones (114), mientras que la instrucción de traza (503) está cargada en el elemento "INSTRUCCIÓN N" (107) de la etapa de decodificación (103). La secuencia incluye, además, que en ese mismo ciclo el elemento "INSTRUCCIÓN N + 1" (108) de la etapa de decodificación (103) contenga la siguiente instrucción (504) a ejecutar. De acuerdo a esta secuencia, en el ciclo T (500) el valor de la señal "N_ES_TRAZA" (129) es 1, mientras que "N_1_ES_TRAZA" (130) vale 0, y el valor de la señal de multiplexación "SEL_TR_P4" (133) es 2, lo que habilita una ruta donde la instrucción de traza (503) se sincroniza con la ejecución de la instrucción a trazar (502). La sincronización se hace efectiva en el ciclo T+1 (510), ubicándose la instrucción de traza (503) en la etapa 4 del *pipeline* de traza (118). En el ciclo T (500), además, la señal de multiplexación "SEL_PIPE_TRAZA" (132) toma el valor 0, con el fin de que en el ciclo T+1 (510) se encuentre un cero (507) en la etapa 3 del *pipeline* de traza (115).

30

La secuencia S3 provoca, además, que durante el ciclo T (500), la siguiente instrucción (504), que se encuentra almacenada en el elemento "INSTRUCCIÓN N + 1" (108), tenga

habilitada la ruta hacia la etapa 3 del *pipeline* de instrucciones (114). Para habilitar esta ruta la señal "SEL_PIPE_INSTR" (131) toma el valor 1 durante el ciclo T (500).

En el ciclo T+1 (510), el dispositivo repite la misma configuración de la ruta de datos que había en el ciclo T (500), ya que la secuencia ubica de nuevo una instrucción de tipo traza (505) en el elemento "INSTRUCCIÓN N" (107) de la etapa de decodificación (103) y la siguiente instrucción a ejecutar (506) en el elemento "INSTRUCCIÓN N + 1" (108).

Finalmente, las figuras 6, 7 y 8 describen el funcionamiento del dispositivo ante la detección de burbujas. Las burbujas se insertan en el *pipeline* de instrucciones de un procesador RISC en todas aquellas situaciones en las que se interrumpe la ejecución secuencial de instrucciones, como ocurre con las instrucciones de salto, tanto condicional como incondicional, o en las llamadas y retornos de funciones. Cuando una instrucción provoca que se interrumpa el orden secuencial de ejecución, el procesador debe descartar la ejecución de las instrucciones posteriores a dicha instrucción, e iniciar la búsqueda de la instrucción cuya dirección ha sido determinada tras la ejecución de la instrucción que ha provocado la ruptura de secuencia. En las figuras 6 y 7 esta dirección está etiquetada como dirección "Z" (600). La figura 6 explica la ruta de datos ante una burbuja en la etapa 3 del *pipeline* de instrucciones (114), mientras que la figura 7 corresponde a una burbuja detectada en la etapa 4 (117), y que provoca además una burbuja en la etapa 3 (114). La figura 8 explica la evolución de la ruta en los ciclos siguientes a la detección de una burbuja hasta que la instrucción de la dirección de salto (600) es suministrada por la etapa de búsqueda (100) .

En la figura 6, se muestra cómo durante el ciclo T se detecta una burbuja sólo en la etapa 3 del *pipeline* de instrucciones (114), y cómo el controlador fija la ruta que carga un 0 en las etapas 3 (115) y 4 (118) del *pipeline* de traza (113), mientras la dirección de salto "Z" (600) se enruta hacia el registro de entrada de la etapa de búsqueda (139). La detección de la burbuja en la etapa 3 se corresponde con la puesta a 1 de la señal "BURBUJA_P3" (125) y consecuentemente de la señal "BURBUJA" (122). La ruta hacia la etapa 3 (115) se controla asignando 0 a la señal "SEL_PIPE_TRAZA" (132), y la ruta hacia la etapa 4 (118) se controla asignando un 0 a la señal "SEL_TR_P4" (133). La carga del registro de entrada de la etapa de búsqueda (139) se controla activando la

señal "LD_DIR" (137) y enrutando la dirección "Z" (600) hacía dicho registro (139) asignando un 0 a la señal "SEL_DIR" (138).

En la figura 7, se muestra cómo durante el ciclo T se detecta una burbuja en la etapa 4 del *pipeline* de instrucciones (117), y cómo el controlador fija la ruta que carga un 0 en las etapas 3 (115), 4 (118) y 5 (121) del *pipeline* de traza (113), mientras la dirección de salto "Z" (600) se enruta hacia el registro de entrada de la etapa de búsqueda (139). La detección de la burbuja en la etapa 4 corresponde con la puesta a 1 de la señal "BURBUJA_P4" (126) y consecuentemente de la señal "BURBUJA" (122). La ruta hacia la etapa 3 del *pipeline* de traza (115) se controla asignando 0 a la señal "SEL_PIPE_TRAZA" (132), la ruta hacia la etapa 4 (118) se controla asignando un 0 a la señal "SEL_TR_P4" (133), y la ruta hacia la etapa 5 (121) se controla asignando un 0 a la señal "SEL_TR_P5" (134). La carga del registro de entrada de la etapa de búsqueda (139) se controla activando la señal "LD_DIR" (137) y enrutando la dirección "Z" (600) hacía dicho registro (139) asignando un 0 a la señal "SEL_DIR" (138).

La figura 8 representa la espera de dos ciclos (801 y 802) tras cualquiera de las dos burbujas descritas en las figuras 6 y 7, de forma que en el ciclo T+2 (802) la etapa de búsqueda (100) desactiva la señal "ESPERA" (124) indicando que las instrucciones están disponibles para cargarse en la etapa de decodificación (103) en el siguiente ciclo, activándose las señales "LD_N" (127) y "LD_N_1" (128).

En el conjunto de casos presentados en las figuras 3, 4, 5, 6, 7 y 8 se describe cómo se comporta el dispositivo propuesto en la invención ante las distintas secuencias de instrucciones, y la aparición de posibles burbujas. En todos los casos se comprueba que el dispositivo hace efectiva la sincronización de la ejecución de las instrucciones de traza con las instrucciones trazadas, evitándose además la sobrecarga en tiempo de ejecución, ya que las instrucciones de traza siempre se ejecutan en paralelo con las instrucciones a trazar.

La realización de la invención estará basada en la especificación estructural de la figura 1 y de funcionamiento según el diagrama de transición de estados de la figura 2, y las

tablas definidas en las figuras 9, 10, 11 y 12. Las figuras 3, 4, 5 y 6 completan los detalles que facilitan la implementación.

5 La realización física preferida consistirá en la implementación “Hardware/Firmware” de la funcionalidad descrita, partiendo de un modelo de descripción de una arquitectura estándar de procesador sobre el que se realizarán las mencionadas modificaciones y que afectan, básicamente, al diseño del *pipeline*. 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
10 Programables, *Field Programmable Gate Array*) o bien sobre un Circuito Integrado de Aplicación Específica (ASIC, *Application Specific Integrated Circuit*).

Hay diferentes opciones de realización. Todas ellas parten del modelo VHDL de un “IP Core” de un procesador RISC segmentado, como ARM o LEON, sobre el que se
15 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.

20

REIVINDICACIONES

1. Un dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza, caracterizado porque comprende:

- 5
- una etapa de búsqueda de instrucciones (100), que a su vez comprende:
 - un módulo de cálculo de la dirección de la instrucción (101); y,
 - módulo de búsqueda de las instrucciones con doble puerto de lectura (102);

- una etapa de decodificación duplicada (103);

- 10
- un *pipeline-traza* (113) para el procesado únicamente de instrucciones de traza;
 - un registro de salida (123) para la traza;
 - una ruta de datos que a su vez comprende un conjunto de multiplexores (109, 110, 116, 119);

- 15
- un controlador de la ruta de datos (104), que a su vez comprende unas entradas (105) y unas salidas (106) que controlan los multiplexores (109, 110, 116, 119), la carga en unos registros asociados las distintas etapas (107, 108, 114, 115, 117, 118, 120, 121) y el registro de salida (123) para la traza;

donde el controlador de la ruta de datos (104) está configurado para determinar, en función del estado de dicho controlador (104) y del valor de las entradas (105) en dicho controlador, el valor de las salidas (106) que son enviadas a los multiplexores (109, 110, 116 y 119) de la ruta de datos de tal forma que una instrucción de traza se ejecuta de forma sincronizada con la instrucción que la precede, haciéndose efectiva dicha ejecución durante la última etapa (121) del *pipeline* de traza (113);

donde el controlador comprende las siguientes secuencias de instrucciones:

- 25
- S1: corresponde a pares Instrucción-Traza (301- 302, 303-304 y 305-306) en los que las instrucciones a ser trazadas (301, 303 y 305) siempre se cargan en el elemento "INSTRUCCIÓN N" (107), mientras las instrucciones correspondientes de traza (302, 304 y 306) se cargan en el elemento "INSTRUCCIÓN N+1" (115); donde el dispositivo de procesamiento ejecuta la secuencia S1 durante dos ciclos de reloj, T (300) y T+1 (307); de tal forma que las instrucciones almacenadas en unas direcciones X+1 (302), X+3 (304) y X+5 (306), son las instrucciones de traza de las instrucciones que las preceden, ubicadas, respectivamente, en unas
- 30

direcciones X (301), X+2 (303) y X+4 (305)

- S2: corresponde a una secuencia de instrucciones (401, 402, 403 y 404) que no son trazadas, de forma en los elementos "INSTRUCCIÓN N" (114) e "INSTRUCCIÓN N+1" (115) siempre se cargan instrucciones; donde el dispositivo de procesamiento ejecuta la secuencia S2 en la que durante dos ciclos no se cargan instrucciones de traza; de tal forma que durante el primer ciclo T (400) se detecta que las dos instrucciones que están cargadas en la etapa de decodificación (403 y 404) no son de traza, y por tanto las señales "N_ES_TRAZA" (129) y "N_1_ES_TRAZA" (130) valen ambas "0"; y, en el ciclo T +1 (408) el controlador se encuentra en el estado denominado "INSTR PENDIENTE" (202), en el que la instrucción pendiente (404) ubicada en la segunda unidad de decodificación (108) se dirige hacia la "etapa 3" del *pipeline* de instrucciones del procesador (114);

- S3: corresponde a dos pares Instrucción-Traza (502- 503 y 504-505) en los que las instrucciones de traza (503 y 505) se cargan en ciclos sucesivos (500 y 510) en el elemento "INSTRUCCIÓN N +1" (107), mientras las instrucciones a trazar (502 y 504) se cargan en esos mismos ciclos en el elemento "INSTRUCCIÓN N+1" (108); el dispositivo de procesamiento ejecuta la secuencia S3: en el ciclo T (500) el valor de la señal "N_ES_TRAZA" (129) es "1", mientras que "N_1_ES_TRAZA" (130) vale "0", y el valor de la señal de multiplexación "SEL_TR_P4" (133) es "2", de tal forma que se habilita una ruta donde la instrucción de traza (503) se sincroniza con la ejecución de la instrucción a trazar (502); en el ciclo T+1 (510), se ubica la instrucción de traza (503) en la etapa 4 del *pipeline* de traza (118). En el ciclo T(500), además, la señal de multiplexación "SEL_PIPE_TRAZA" (132) toma el valor 0, con el fin de que en el ciclo T+1 (510) se encuentre un cero (507) en la etapa 3 del *pipeline* de traza (115);

donde el dispositivo de procesamiento durante un ciclo "T", detecta una burbuja en la etapa 3 del *pipeline* de instrucciones (114), de tal forma que el controlador fija la ruta que carga un "0" en las etapas 3 (115) y 4 (118) del *pipeline*- traza (113) y una dirección de salto "Z" (600) se enruta hacia el registro de entrada de la etapa de búsqueda (139), de tal forma que la detección de la burbuja en la etapa 3 se corresponde con la puesta a "1" de la señal "BURBUJA_P3" (125) y de la señal "BURBUJA" (122);

donde el controlador controla: la ruta hacia la etapa 3 (115) asignando "0" a la señal "SEL_PIPE_TRAZA" (132); la ruta hacia la etapa 4 (118) asignando un "0" a la señal "SEL_TR_P4" (133); y la carga del registro de entrada de la etapa de búsqueda (139) activando la señal "LD_DIR" (137) y enrutando la dirección "Z" (600) hacía dicho registro (139) asignando un "0" a la señal "SEL_DIR" (138);

donde el dispositivo de procesamiento durante un ciclo "T", detecta una burbuja en la etapa 4 del *pipeline* de instrucciones (117), de tal forma el controlador fija la ruta que carga un "0" en las etapas 3 (115), 4 (118) y 5 (121) del *pipeline*-traza (113) y la dirección de salto "Z" (600) se enruta hacia el registro de entrada de la etapa de búsqueda (139), de tal forma que la detección de la burbuja en la etapa 4 se corresponde con la puesta a "1" de la señal "BURBUJA_P4" (126) y de la señal "BURBUJA" (122);

donde el controlador controla: la ruta hacia la etapa 3 del *pipeline*-traza (115) asignando "0" a la señal "SEL_PIPE_TRAZA" (132); la ruta hacia la etapa 4 (118) asignando un "0" a la señal "SEL_TR_P4" (133); la ruta hacia la etapa 5 (121) se controla asignando un "0" a la señal "SEL_TR_P5" (134); y, la carga del registro de entrada de la etapa de búsqueda (139) activando la señal "LD_DIR" (137) y enrutando la dirección "Z" (600) hacía dicho registro (139) asignando un 0 a la señal "SEL_DIR" (138).

2. Un procesador RISC, "Computador con Conjunto de Instrucciones Reducidas", caracterizado porque comprende un dispositivo de procesamiento en paralelo de instrucciones de programa e instrucciones de traza según la reivindicación 1.

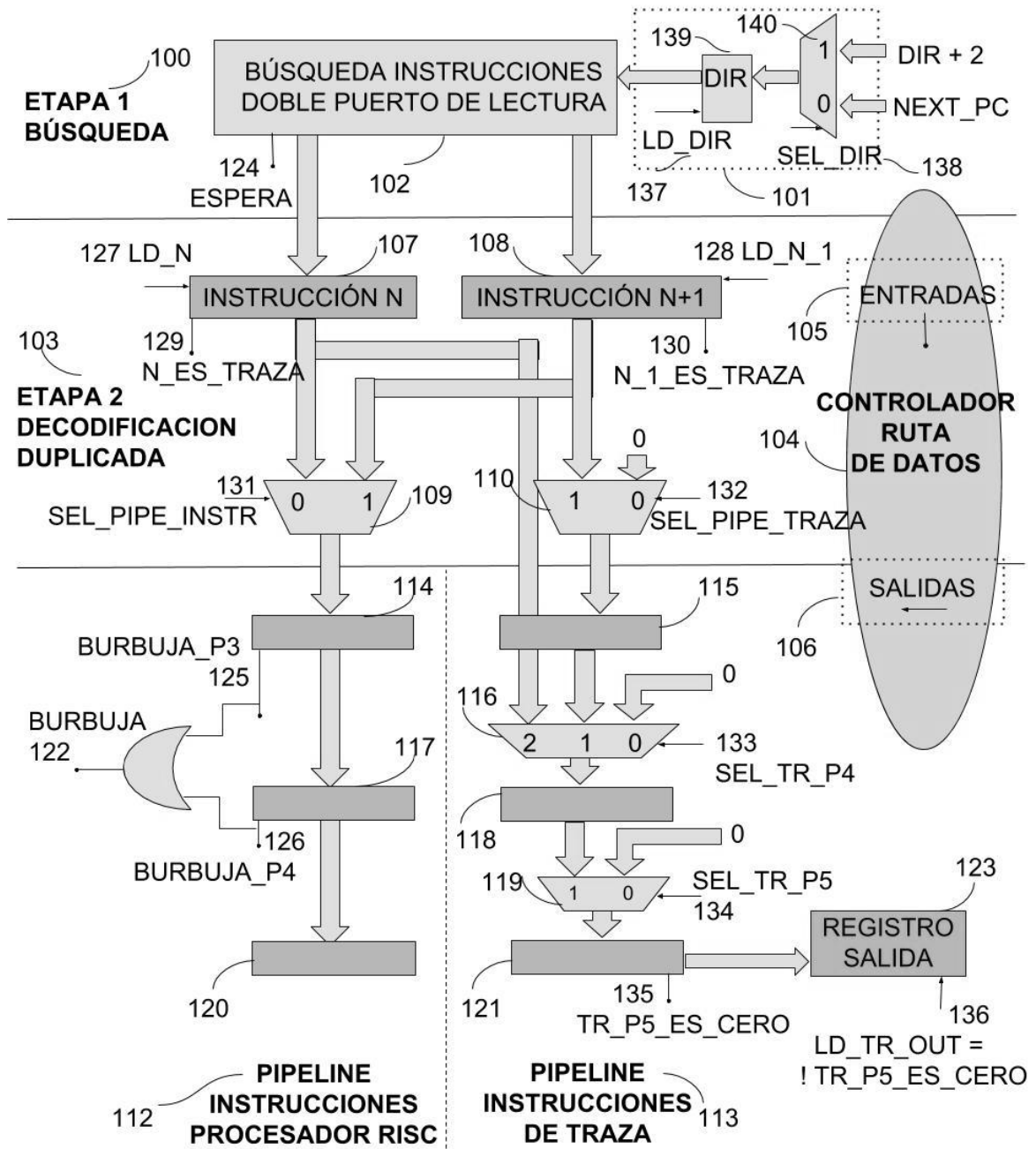


FIG. 1

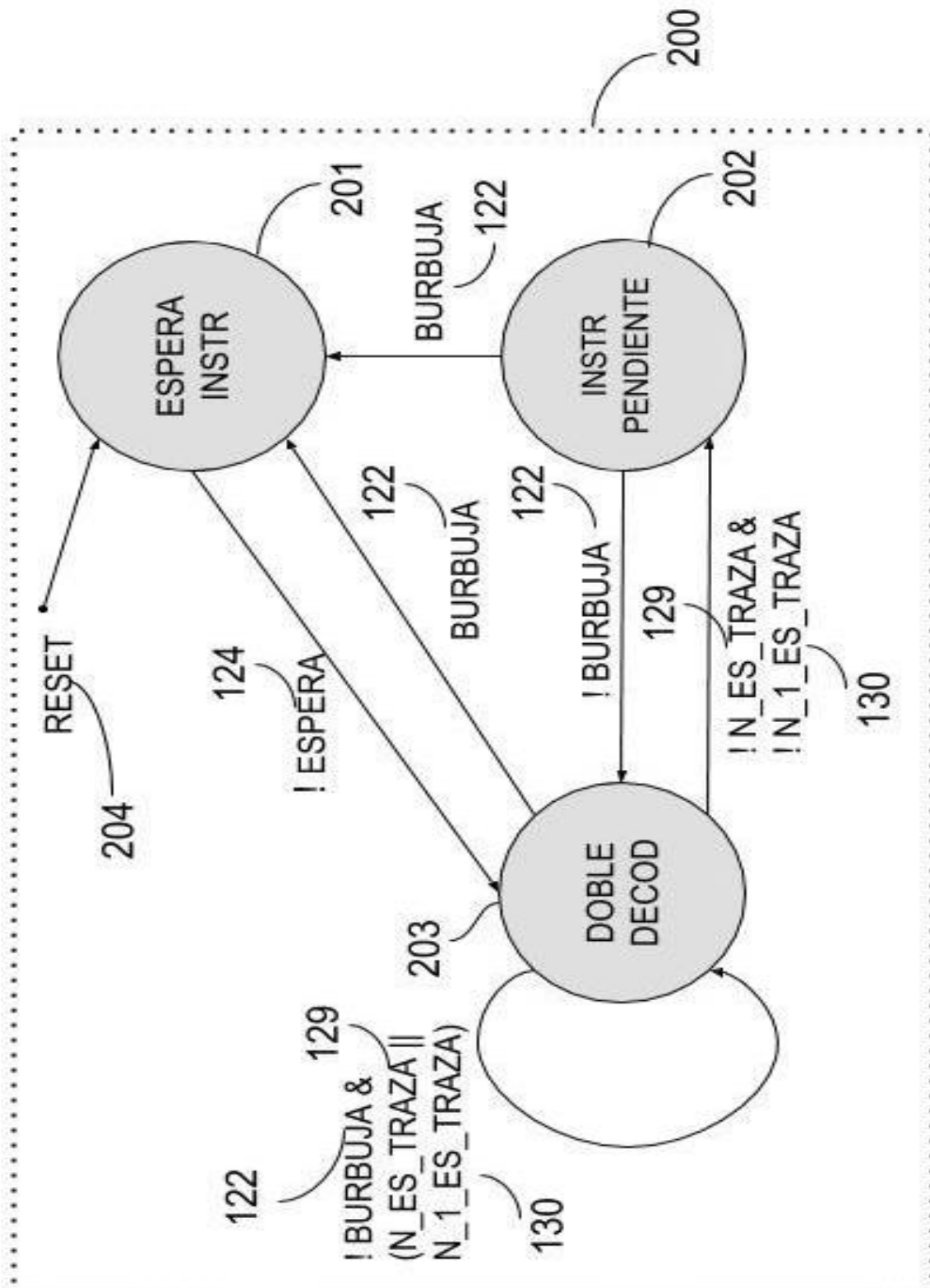


FIG. 2

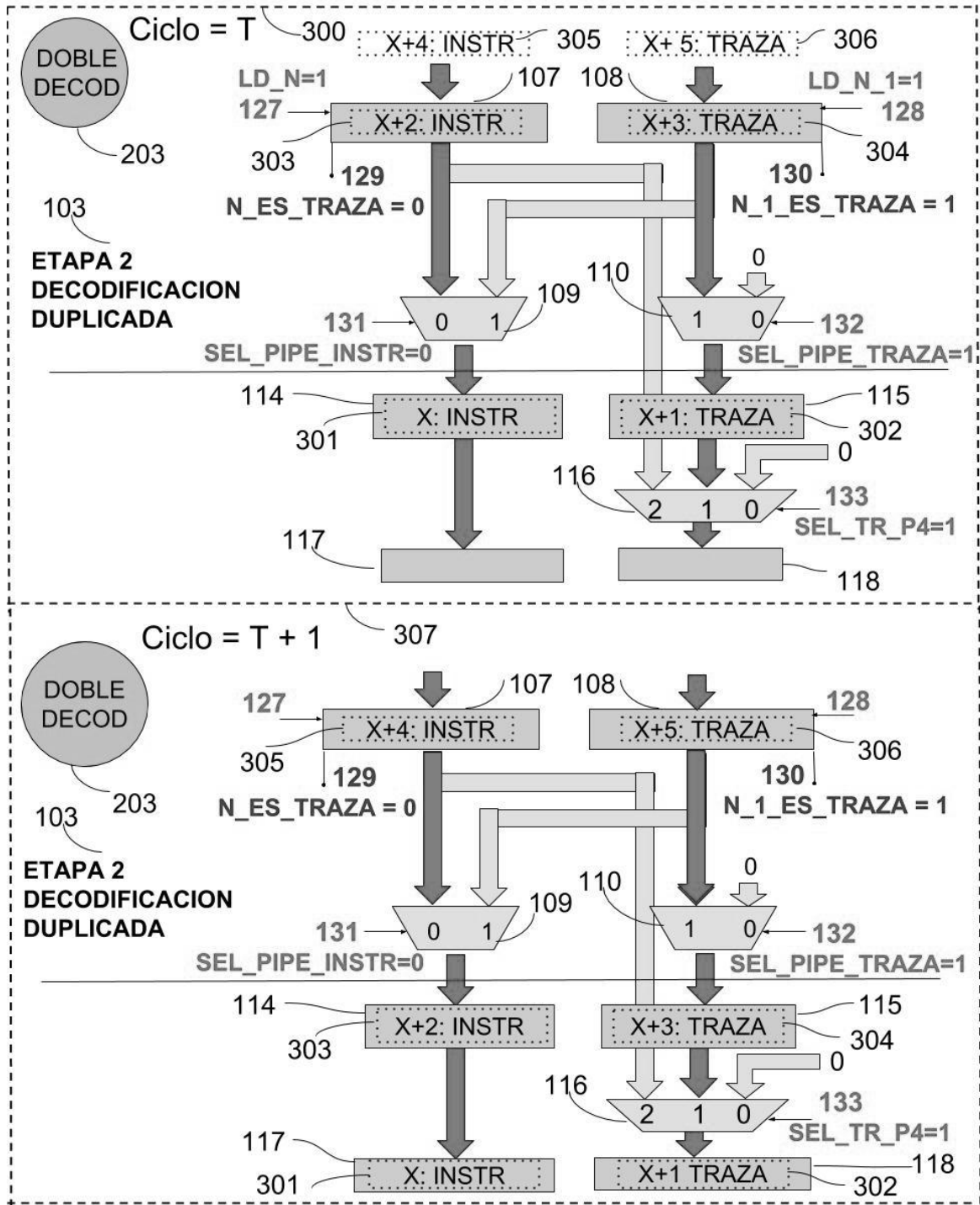


FIG. 3

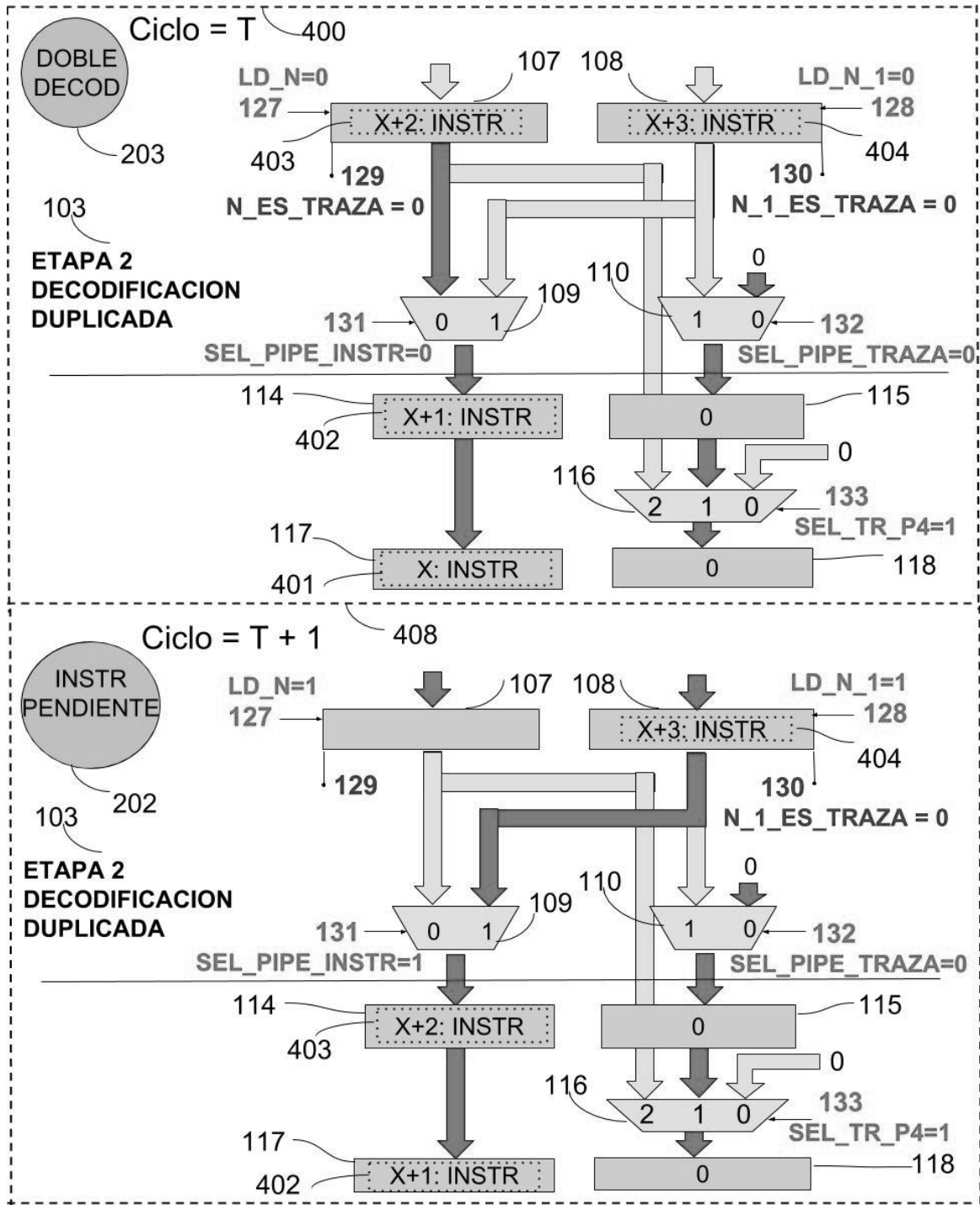


FIG. 4

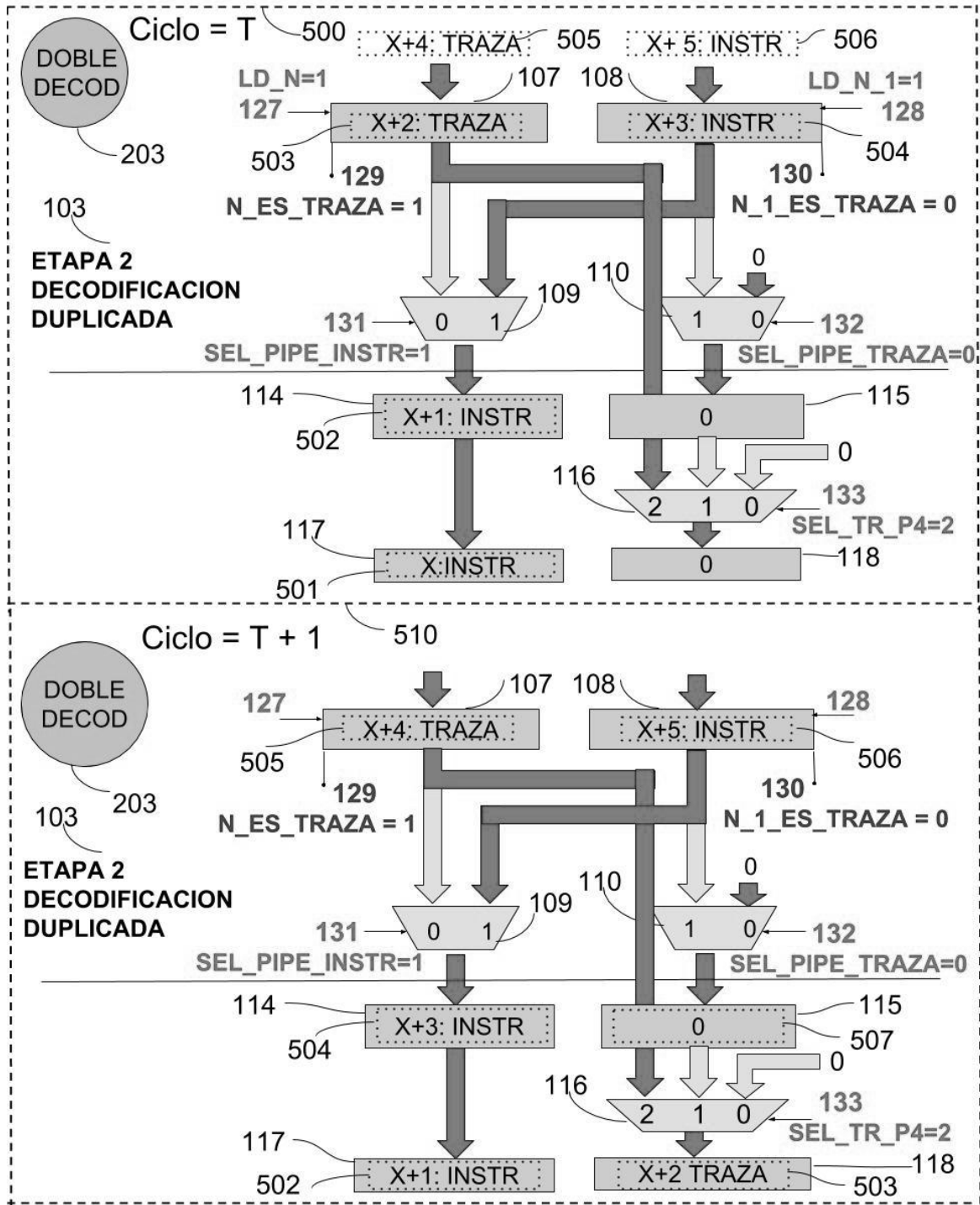


FIG. 5

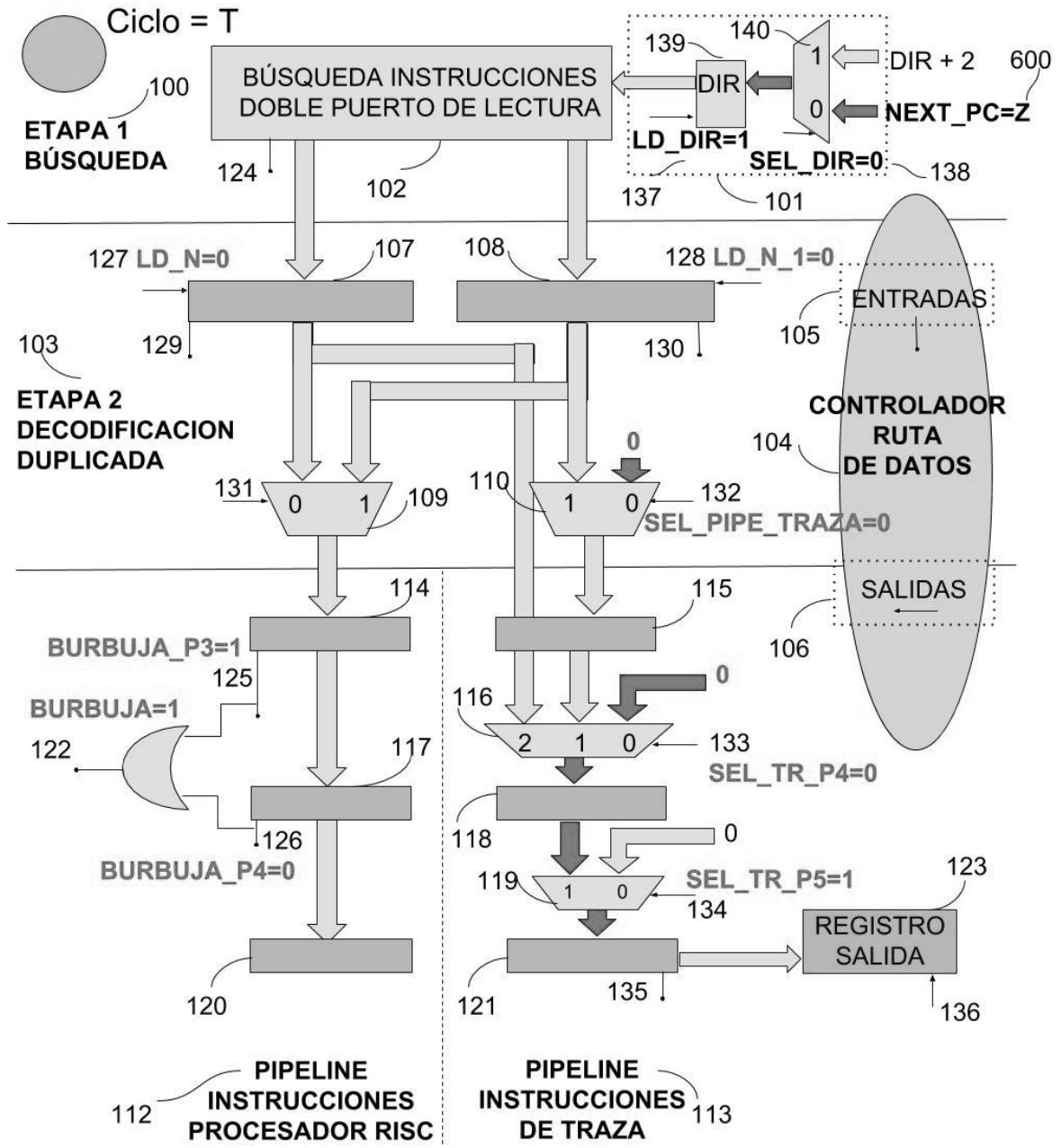


FIG. 6

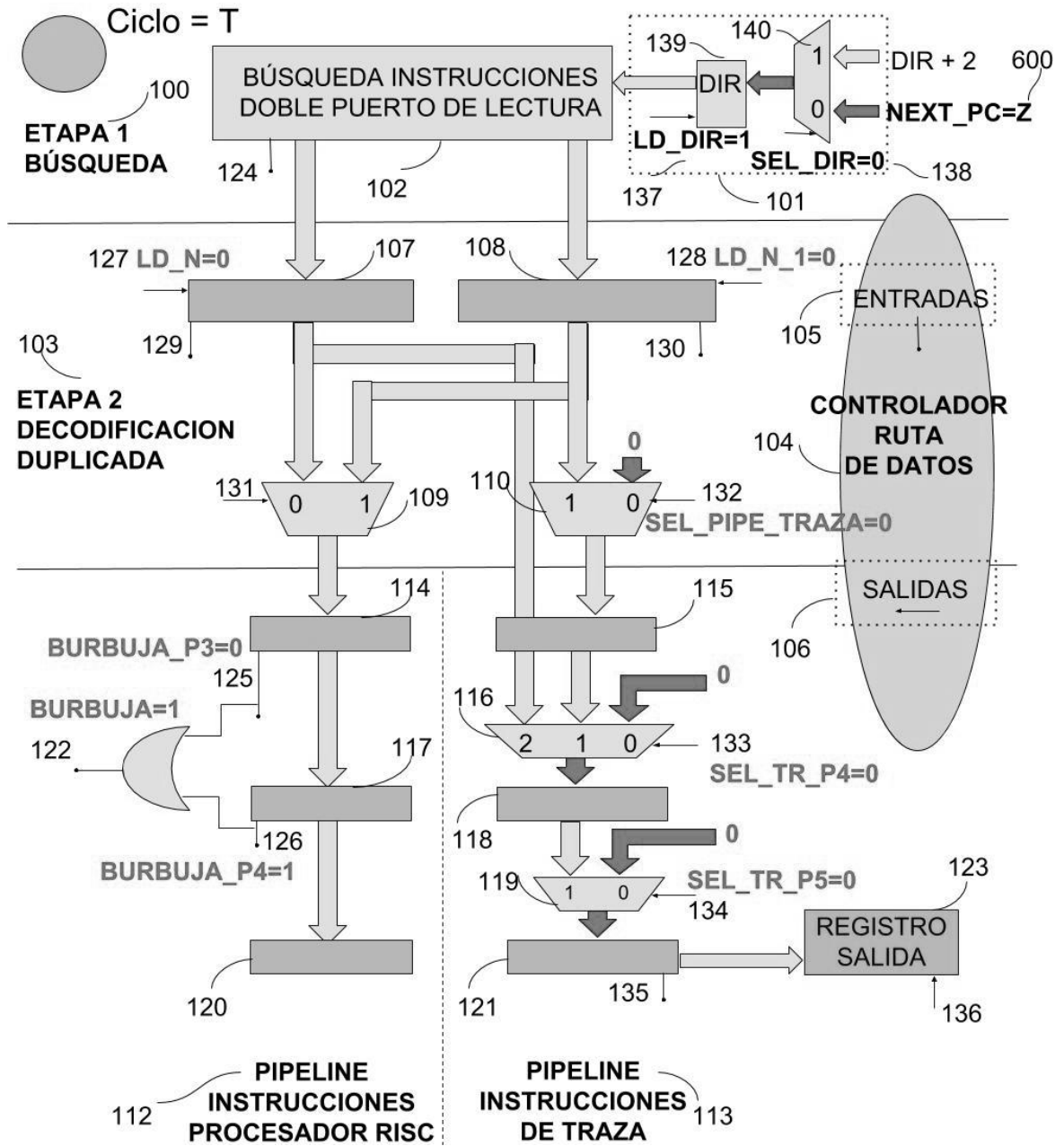


FIG. 7

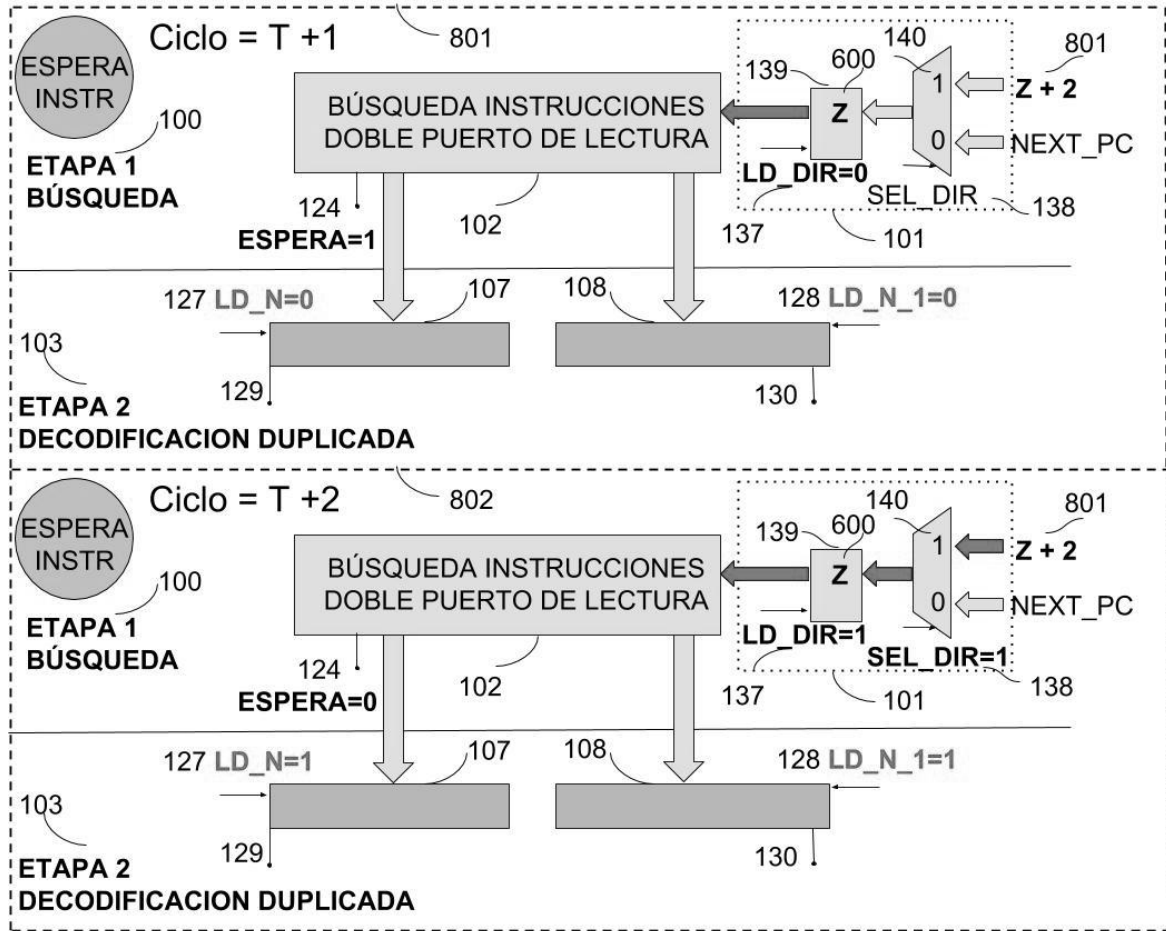


FIG. 8

ESTADO ACTUAL	INPUTS				ESTADO SIGUIENTE
	BURBUJA	N_ES_TRAZA	N_1_ES_TRAZA	ESPERA	
DOBLE_DECOD	0	0	0	X	INSTR PENDIENTE
DOBLE_DECOD	0	0	1	X	DOBLE DECOD
DOBLE_DECOD	0	1	0	X	DOBLE DECOD
DOBLE_DECOD	0	1	1	X	DOBLE DECOD
INSTR PENDIENTE	0	X	X	X	DOBLE DECOD
ESPERA INSTR	X	X	X	0	DOBLE DECOD
ESPERA INSTR	X	X	X	1	ESPERA INSTR
X	1	X	X	X	ESPERA INSTR

FIG. 9

ESTADO ACTUAL	INPUTS					OUTPUTS	
	BURBUJA	N_ES_TRAZA	N_1_ES_TRAZA	ESPERA	LD_DIR	SEL_DIR	
DOBLE_DECOD	0	0	0	X	0	1	
DOBLE_DECOD	0	0	1	X	1	1	
DOBLE_DECOD	0	1	0	X	1	1	
DOBLE_DECOD	0	1	1	X	1	1	
INSTR PENDIENTE	0	X	X	X	1	1	
ESPERA INSTR	X	X	X	0	1	1	
ESPERA INSTR	X	X	X	1	RESET	0	
X	1	X	X	X	1	0	

FIG. 10

ESTADO ACTUAL	INPUTS					OUTPUTS	
	BURBUJA	N_ES_TRAZA	N_1_ES_TRAZA	ESPERA	LD_N	LD_N_1	
DOBLE_DECOD	0	0	0	X	0	0	
DOBLE_DECOD	0	0	1	X	1	1	
DOBLE_DECOD	0	1	0	X	1	1	
DOBLE_DECOD	0	1	1	X	1	1	
INSTR PENDIENTE	0	X	X	X	1	1	
ESPERA INSTR	X	X	X	0	1	1	
ESPERA INSTR	X	X	X	1	0	0	
X	1	X	X	X	0	0	

FIG. 11

ESTADO ACTUAL	INPUTS				OUTPUTS			
	BURBUJA	N_ES_TRAZA	N_1_ES_TRAZA	ESPERA	SEL_PIPE_TRAZA	SEL_TR_P4	SEL_TR_P5	BURBUJA_P4
DOBLE_DECOD	0	0	0	X	0	1	1	1
DOBLE_DECOD	0	0	1	X	1	1	1	1
DOBLE_DECOD	0	1	0	X	0	2	1	1
DOBLE_DECOD	0	1	1	X	1	1	1	1
INSTR PENDIENTE	0	X	X	X	0	1	1	1
ESPERA INSTR	X	X	X	0	X	X	X	X
ESPERA INSTR	X	X	X	1	X	X	X	X
X	1	X	X	X	0	0	0	BURBUJA_P4

FIG. 12