

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 676 303**

51 Int. Cl.:

**G06F 9/38**

(2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **15.08.2007 PCT/US2007/075996**

87 Fecha y número de publicación internacional: **21.02.2008 WO08022190**

96 Fecha de presentación y número de la solicitud europea: **15.08.2007 E 07814118 (1)**

97 Fecha y número de publicación de la concesión europea: **11.04.2018 EP 2057536**

54 Título: **Procedimientos y aparatos para reducir las búsquedas en una memoria caché de dirección de destino de saltos**

30 Prioridad:

**16.08.2006 US 464996**

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

**18.07.2018**

73 Titular/es:

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

72 Inventor/es:

**MORROW, MIKE**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 676 303 T3**

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

## DESCRIPCIÓN

Procedimientos y aparatos para reducir las búsquedas en una memoria caché de dirección de destino de saltos

5 **Campo de la invención**

[1] La presente invención se refiere en general al campo de los procesadores y, en particular, a un procedimiento para mejorar la predicción de saltos reduciendo las búsquedas en una memoria caché de direcciones de destino de saltos.

10

**Antecedentes**

[2] Los microprocesadores realizan tareas de computación en una amplia variedad de aplicaciones. El rendimiento del procesador mejorado casi siempre es deseable, para permitir una operación más rápida y/o una mayor funcionalidad a través de cambios de software. En muchas aplicaciones integradas, tal como dispositivos electrónicos portátiles, la conservación de energía y el rendimiento más rápido también son destinos en el diseño e implementación del procesador.

15

[3] Muchos procesadores modernos emplean una arquitectura canalizada, donde instrucciones secuenciales, cada una con múltiples etapas de ejecución, se superponen en ejecución. Para un mejor rendimiento, las instrucciones deben fluir continuamente a través del canal. Cualquier situación que cause que las instrucciones se estanquen en el canal puede influir negativamente en el rendimiento. Si las instrucciones se eliminan del canal y posteriormente se recuperan, tanto el rendimiento como el consumo de energía se ven afectados.

20

25

[4] La mayoría de los programas incluyen instrucciones indirectas de saltos donde el comportamiento de saltos real no se conoce hasta que la instrucción de saltos indirectos se evalúa en profundidad en el canal. Para evitar el bloqueo que resultaría de esperar la evaluación real de la instrucción de saltos indirecto, los procesadores modernos pueden emplear alguna forma de predicción de saltos, por lo que el comportamiento de los saltos de las instrucciones de saltos indirectos se predice al principio en el canal. En función de la evaluación de saltos predicha, el procesador obtiene de forma especulativa (obtención previa) y procesa las instrucciones desde una dirección predicha, ya sea la dirección de destino de saltos (si se predice que se tomarán los saltos) o la siguiente dirección secuencial después de la instrucción de saltos (si se predice que no se tomará el salto). El que se tome o no una instrucción de saltos indirectos se denomina determinación de la dirección del salto.

30

35

[5] Las técnicas de predicción de saltos convencionales incluyen una memoria caché de direcciones de destino de saltos (BTAC) situada en una etapa de obtención de un canal de procesador y una lógica de predicción de saltos. La BTAC almacena la dirección de destino de una instrucción obtenida previamente y se indexa mediante la dirección de la instrucción. Las I-cachés se rellenan de manera convencional con instrucciones de varios tipos de instrucciones que se recuperaron de una memoria caché o memoria de orden superior. Las BTAC se rellenan de manera convencional después de que una instrucción de saltos indirectos se resuelve más adelante en el canal del procesador.

40

[6] En operación, las técnicas de predicción de saltos convencionales realizan búsquedas de direcciones en las instrucciones de obtención previa tanto en una BTAC como en una I-caché en paralelo. Si hay una falta en la BTAC, estas técnicas de saltos convencionales han consumido energía en la búsqueda de la BTAC sin encontrar una coincidencia. Si hay un acierto en la BTAC, la dirección que se busca puede considerarse una instrucción de saltos indirectos. Después de la búsqueda de la BTAC, las técnicas convencionales invocan la lógica de predicción de saltos para determinar si una dirección de saltos recuperada de la BTAC debería predecirse o no. Si la lógica de predicción de saltos predice que se toma, la lógica de predicción de saltos redirige el flujo de instrucciones recuperando las instrucciones que comienzan en la dirección de destino de saltos.

45

50

[7] Cualquier instrucción secuencial que ingrese al canal del procesador desde la instrucción de saltos normalmente se descarga desde el canal. La trayectoria definida por la búsqueda de la BTAC y la posterior predicción de saltos suele ser una trayectoria de velocidad crítica porque cuanto menor es el tiempo de esta trayectoria, menor es la cantidad de instrucciones que deben descargarse del canal del procesador antes de redirigir el flujo de instrucciones. Consecuentemente, es deseable que esta trayectoria sea lo más corta posible para minimizar la potencia gastada en las instrucciones de descarga.

55

[8] Las técnicas convencionales para reducir el tiempo de la trayectoria crítica incluyen reducir el tamaño de la BTAC y/u organizar la BTAC de forma multidireccional. Sin embargo, al reducir el tamaño de la BTAC, se reduce el número de aciertos potenciales y, por lo tanto, la probabilidad de encontrar una dirección de destino de saltos en la BTAC, lo que reduce la efectividad de la BTAC en conjunto. Además, al organizar la BTAC de forma multidireccional, la indexación en la BTAC puede ser más rápida, pero el tiempo de comparación gastado puede aumentar. En estas situaciones, la BTAC puede ser más lenta que la I-caché, convirtiéndose así en el factor limitante en la porción de búsqueda paralela de la trayectoria crítica. Por lo tanto, se reconoce que son

60

65

necesarios aparatos y procedimientos para reducir el tiempo de redirección del flujo de instrucción cuando se encuentra una instrucción de saltos indirectos en un canal de procesador sin disminuir la efectividad de la predicción de saltos. El documento US 6.651.162 describe un procedimiento de obtención previa de direcciones que incluye la etapa de acceder a una instrucción almacenada usando una dirección actual.

5

**SUMARIO**

[9] De acuerdo con la invención, se proporciona un procedimiento para reducir las búsquedas en una memoria caché de direcciones de destino de saltos (BTAC) de acuerdo con la reivindicación 1; y se proporciona un sistema de acuerdo con la reivindicación 11. La presente divulgación reconoce que las técnicas de predicción de saltos convencionales a menudo consumen energía innecesariamente cuando las búsquedas se realizan en una I-caché y una BTAC en paralelo y las búsquedas fallan en la BTAC. Este reconocimiento es más evidente cuando hay un acierto en la I-caché y una falta en la BTAC, que probablemente se debe a que la I-caché generalmente almacena todos los tipos de instrucciones y la BTAC normalmente almacena las direcciones de las instrucciones de saltos.

[10] De acuerdo con una realización, se divulga un procedimiento para reducir las búsquedas en una memoria caché de direcciones de destino de saltos (BTAC). En este procedimiento, se recupera una dirección de destino de saltos de la BTAC en respuesta a una falta al buscar una dirección de instrucción en una memoria caché de instrucciones (I-caché). La dirección de destino de saltos está asociada con la dirección de la instrucción. La dirección de destino de saltos recuperada de la BTAC se almacena en la I-caché. Con estas técnicas divulgadas, las direcciones de instrucción posteriores se buscan ventajosamente en la I-caché, no en paralelo con la BTAC, ahorrando así potencia al reducir las búsquedas de la BTAC innecesarias.

[11] De acuerdo con otra realización, se divulga un procedimiento para almacenar instrucciones de saltos en una memoria caché de instrucciones. Este procedimiento incluye buscar una dirección de instrucción de saltos en una memoria caché de instrucciones (I-caché), recuperar una dirección de destino de saltos de una memoria caché de direcciones de destino de saltos (BTAC) en respuesta a una falta de la I-caché y almacenar la dirección de destino de saltos en una entrada en la I-caché.

[12] Otra realización se refiere a un sistema para reducir las búsquedas en una memoria caché de direcciones de destino de saltos (BTAC). El sistema incluye una BTAC y una memoria caché de instrucciones (I-caché) configurada para recuperar una dirección de destino de saltos desde la BTAC en respuesta a una falta de memoria caché al buscar una primera dirección de instrucción de saltos. La I-caché está configurada además para almacenar la dirección de destino de saltos.

[13] Debe entenderse que otros modos de realización de la presente invención resultarán fácilmente evidentes a los expertos en la técnica a partir de la siguiente descripción detallada, en la que se muestran y se describen a modo de ilustración diversos modos de realización de la invención. Como se comprobará, la invención es capaz de otros modos de realización diferentes y sus diversos detalles son capaces de modificarse en otros diversos aspectos, todo sin apartarse de la presente invención. Por consiguiente, los dibujos y la descripción detallada han de considerarse como ilustrativos por naturaleza, y no como restrictivos.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[14]

La FIG. 1 es un diagrama de bloques funcional de un procesador de ejemplo.

Las FIGs. 2A-2C (colectivamente FIG. 2) ilustran una lista de un segmento de código a modo de ejemplo y contenidos de ejemplo de una I-caché y una BTAC cuando la I-caché no contiene la instrucción de saltos del segmento de código.

La FIG. 3 ilustra contenido de ejemplo de la I-caché de la Fig. 1 después de que la I-caché se rellena con datos de instrucción desde la BTAC.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento para almacenar una instrucción de saltos indirectos en una I-caché y recuperar una instrucción de la I-caché de la figura 1.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de gestión de una I-caché en la resolución de saltos real.

La FIG. 6 es un diagrama de bloques funcional de dos procesadores que comparten una BTAC común.

**DESCRIPCIÓN DETALLADA**

**[15]** La FIG. 1 representa un diagrama de bloques funcional de un procesador 100 de ejemplo. El procesador 100 ejecuta instrucciones en un canal de ejecución de instrucciones 112 de acuerdo con una lógica de control 114. En algunas realizaciones, el canal 112 puede ser un diseño superescalar, con múltiples canales paralelos, como se describe con más detalle en relación con la FIG. 6. El canal 112 incluye varios registros o cierres 116A-D, organizados en etapas del canal, y una o más unidades de ejecución tales como una unidad lógica aritmética (ALU) 118. Un archivo de registro de propósito general (GPR) 120 proporciona registros que comprenden la parte superior de la jerarquía de memoria.

**[16]** Se accede a los datos desde una memoria caché de datos (D-caché) 140, con la traducción de la dirección de memoria y los permisos administrados por una memoria intermedia principal de traducción adelantada (TLB) 142. En diversas realizaciones, la ITLB 124 puede comprender una copia de parte de la TLB 142. Alternativamente, la ITLB 124 y la TLB 142 pueden estar integradas. De forma similar, en diversas realizaciones del procesador 100, la I-caché 122 y la D-caché 140 pueden estar integradas, o unificadas. Las llamadas faltas en la I-caché 122 y/o en la D-caché 140 provocan un acceso a la memoria principal (fuera de chip) 144, bajo el control de una interfaz de memoria 146. Las faltas en la I-caché 122 se describirán con más detalle en conexión con el análisis de la FIG. 4.

**[17]** El procesador 100 puede incluir una interfaz 148 de Entrada/Salida (E/S) que controla el acceso a diversos dispositivos periféricos 150. Los expertos en la técnica reconocerán que son posibles numerosas variaciones del procesador 100. Por ejemplo, el procesador 100 puede incluir una memoria caché de segundo nivel (L2) para una o ambas memorias caché I y D 122, 140. Además, uno o más de los bloques funcionales representados en el procesador 100 pueden omitirse de una realización particular.

**[18]** El procesador 100 incluye un sistema de predicción de saltos 143. El sistema de predicción de saltos 143 incluye una BTAC 141, un circuito de predicción de saltos (BP) 126, un circuito lógico de actualización 160 y un circuito de decodificación previa 156 opcional. La BTAC 141 está configurada para almacenar una o más entradas, donde cada entrada contiene una dirección de instrucción de saltos y una dirección de destino de saltos correspondiente a una dirección de obtención. La BTAC 141 puede almacenar una combinación de direcciones de instrucción de saltos directos e indirectos.

**[19]** La I-caché 122 y la BTAC 141 se rellenan sobre una trayectoria de llenado 152 cuando falta una dirección de instrucción en la I-caché 122 y se realiza una solicitud para recibir la instrucción correspondiente de la memoria de nivel superior sobre la interfaz de memoria 146. Una vez que se recibe la instrucción desde la interfaz de memoria 146, en una primera realización, el circuito lógico de decodificación previa 156 opcional se configura para determinar si la instrucción recibida es una instrucción de saltos indirectos. Si es así, la dirección de la instrucción recibida se busca en la BTAC 141 sobre la trayectoria 152 para ver si hay una dirección de destino asociada con la dirección de la instrucción recibida. Si hay un acierto en la BTAC 141, la instrucción recibida y la dirección de destino correspondiente se escriben en la I-caché 122. Una falta en la I-caché 122 y un acierto en la BTAC 141 pueden producirse en una situación en la que una instrucción de saltos indirectos se predice incorrectamente, causando así algún tipo de invalidación en la I-caché 122. Los tipos de invalidación de la I-caché se analizarán con más detalle en conexión con el análisis de la FIG. 5. Si la dirección de la instrucción recibida no está en la BTAC 141 (por ejemplo, una falta), la instrucción recibida se escribe en la I-caché 122.

**[20]** Una segunda realización elimina el circuito lógico de decodificación previa 156. En la segunda realización, las instrucciones recibidas a través de la interfaz de memoria 146 se buscarán en la BTAC 141 independientemente de si la instrucción recibida es una instrucción de saltos indirectos. En esta segunda realización, las instrucciones, como la primera realización, se envían a la I-caché 122 para su almacenamiento. Sin embargo, en la primera realización, las búsquedas de la BTAC están limitadas a instrucciones de saltos indirectos. En cualquier realización, si se almacena una dirección de saltos en la BTAC 141, también se almacenará en la I-caché 122.

**[21]** La lógica de control de actualización 160 añade una dirección de destino a la entrada en la BTAC 141 después de que la resolución de saltos real tenga lugar en las últimas etapas del canal 112 a través de la trayectoria 162. La BTAC 141 actualiza la I-caché 122 sobre la trayectoria 154 para incluir tanto la instrucción de saltos indirectos como su dirección de destino correspondiente. El término "predecir no tomar" se refiere al circuito BP 126 que predice que no se redireccionará el canal 112 de acuerdo con la dirección de destino de saltos.

**[22]** Con la traducción de direcciones de memoria y los permisos administrados por una memoria intermedia de traducción adelantada de instrucciones (ITLB) 124, la unidad de obtención previa de instrucción 128 obtiene instrucciones desde una memoria caché de instrucciones (I-caché o I-cache) 122 sobre la trayectoria 125. Si se encuentra una instrucción obtenida (por ejemplo, acierto) en la I-caché 122 y la instrucción obtenida está asociada con una dirección de saltos correspondiente, el circuito BP 126 predice si tomar o no una trayectoria de código que comienza en la dirección de destino. Si el circuito BP 126 predice que debe tomarse la dirección destino, la unidad de obtención previa de instrucciones 128 ajusta su contador de programa para comenzar la obtención de instrucciones desde la dirección destino, redirigiendo así el canal 112 de acuerdo con la dirección

destino de saltos. Si el circuito BP 126 predice que no debe tomarse la dirección de destino, el contador del programa se incrementa para comenzar a obtener instrucciones desde la siguiente dirección secuencial. La operación del sistema de predicción de saltos 143 se describirá con más detalle en relación con el análisis de las Figs. 4 y 5.

5

**[23]** La presente divulgación contempla una realización de combinar la BTAC 141 y la I-caché 122 en un componente integral. Sin embargo, existen varias razones para separar la BTAC y la I-caché en componentes distintos. Una BTAC externa como se divulga solo se consulta o se lee durante los rellenos en la I-caché. Se accede a una BTAC convencional que se coloca en paralelo con la I-caché cada vez que se accede a la I-caché. Al limitar la consulta de la BTAC como se divulga actualmente, la presente divulgación ahorra energía al reducir la actividad en la BTAC. Además, este acceso infrecuente de la BTAC permite que la BTAC 141 sea un dispositivo de puerto único, reduciendo así la complejidad del diseño.

10

**[24]** La FIG. 2A ilustran una lista 210 de un segmento de código ejemplar que contiene una instrucción de saltos indirectos 212 en la dirección 0x000B. Como se representa, el destino de la instrucción de saltos indirectos 212 está definido por el contenido del registro 1 (R1). Dado que el contenido de R1 puede variar cada vez que se ejecuta la instrucción de saltos indirectos 212, se utiliza el sistema de predicción de saltos 143 para determinar si se redirige el canal de proceso 112 comenzando con la dirección especificada por los contenidos de R1.

15

**[25]** La FIG. 2B ilustra los contenidos a modo de ejemplo de la I-caché 230. La I-caché 230 puede ser adecuadamente similar a la I-caché 122. La I-caché 230 incluye una columna de instrucciones 232, una columna de bits válidos 234 y una columna de direcciones de destino de saltos 236. Una entrada en la columna 232 representa una instrucción. Por convención, un valor "0" en la columna 234 indica que el destino del salto, si lo hay, en la columna 236 no es válido, mientras que un valor "1" en la columna 234 indica que el destino del salto en la columna 236 en la misma fila es válido. Como se representa, la I-caché 230 en un punto en el tiempo donde el contador de programa que apunta a la dirección 0x000B no contiene una entrada para la instrucción de saltos indirectos 212 en la dirección 0x000B.

20

25

**[26]** La FIG. 2C ilustra contenidos ejemplares de la BTAC 250. La BTAC 250 puede ser adecuadamente similar a la BTAC 141. La BTAC 250 incluye una columna de dirección de instrucción de saltos 252 y una columna de dirección de destino 254. La columna 252 puede incluir direcciones de instrucción de saltos directos, direcciones de instrucción de saltos indirectos, o cualquier combinación de las mismas. En el momento en que el contador del programa apunta a la dirección 0x000B, la BTAC 250 contiene una dirección de instrucción de saltos indirectos 256 y su dirección de destino correspondiente. En consecuencia, la lógica de control 114 hará que la instrucción de saltos 256 y su dirección de destino correspondiente en la BTAC 250 almacenen la I-caché 230 en una falta de I-caché. Se reconoce que puede almacenarse otra información tanto en la I-caché 230 como en la BTAC 250 y que diversas organizaciones de la I-caché 230 y la BTAC 250 están contempladas por la presente divulgación.

30

35

**[27]** La FIG. 3 ilustra la I-caché 230 después de que la instrucción de saltos indirectos asociada con la dirección de instrucción de saltos indirectos 256 y su dirección de destino correspondiente (0x2000) se almacenen en la I-caché 230 en la dirección 0x000B. En consecuencia, la próxima vez que el contador del programa del procesador apunte a la dirección 0x000B, se realizará una búsqueda de I-caché y se asignará la dirección de destino 0x2000 al contador del programa del procesador para comenzar a redireccionar el canal 112 con instrucciones en la dirección 0x2000. En comparación con las BTAC convencionales, la búsqueda de la I-caché en la presente divulgación se realiza en una instrucción obtenida previamente sin realizar una búsqueda en la BTAC, lo que ahorra consumo de energía al reducir el número de búsquedas de la BTAC. Como se demostró, las búsquedas en la BTAC 250 están limitadas a situaciones en las que no se encuentra una instrucción particular indirecta de saltos en la I-caché 230.

40

45

50

**[28]** La FIG. 4 es un diagrama de flujo que ilustra un procedimiento 400 para almacenar una instrucción de saltos indirectos en una I-caché y recuperar una dirección de destino de saltos de la I-caché de la figura 1. Los bloques 410, 420, 450, 455, 460, 465, 470 y 480 definen el proceso para almacenar una instrucción de saltos indirectos y una dirección de destino de saltos en una I-caché, tal como la I-caché 122 de la figura 1. Los bloques 410, 420, 425, 435, 438, 440 y 445 definen el proceso para recuperar una dirección de destino de saltos desde una I-caché. En el bloque 410, se consulta una I-caché para una instrucción basada en la dirección de la instrucción. Por ejemplo, la dirección de la instrucción almacenada en el contador de programa de un procesador se puede utilizar para consultar una I-caché. En el bloque 420, el procedimiento 400 determina si hay un acierto en la I-caché. Si no hay (una falta de caché), el procedimiento 400 procede al bloque 450 para recuperar los datos de instrucción correspondientes desde la memoria de nivel superior. En el bloque 450, se solicitan datos de instrucción desde la memoria de nivel superior, tal como una memoria caché de capa 2 (L2) o memoria fuera de chip. En el bloque 455, se reciben los datos de instrucción recuperados desde la memoria de nivel superior. En el bloque 460, se busca la dirección de instrucción en la BTAC. En el bloque 465, el procedimiento 400 determina si la dirección de instrucción entra en la BTAC o, en otras palabras, coincide con una dirección de instrucción almacenada en la BTAC. Si la dirección de instrucción acierta en la BTAC, el procedimiento 400 avanza al bloque 470 donde un destino de saltos predicho, si lo hay, almacenado en la BTAC se añade a los datos de

55

60

65

instrucción. En un acierto en la BTAC, el destino de saltos previsto probablemente se almacenará en la BTAC siempre que se haya resuelto previamente la dirección de destino de saltos de una instancia previa de la instrucción de saltos indirectos. El procedimiento 400 avanza al bloque 480, donde los datos de instrucción se almacenan en la I-caché.

5

**[29]** Volviendo al bloque 465, si la dirección de instrucción falla en la BTAC, el procedimiento 400 procede al bloque 480. Esta transición desde el bloque 465 a 480 indica la situación en la que la dirección de instrucción consultada no está en la I-caché o en la BTAC. En esta situación, se almacenará una dirección de destino de saltos en la BTAC después de la resolución del destino de saltos real para los datos de instrucción recuperados que se producen más tarde en el canal del procesador como se describe con más detalle en conexión con la FIG. 6. Cuando se resuelve el destino de saltos real, por ejemplo, el circuito lógico de actualización 160 provocará que la BTAC y la I-caché almacenen el destino de saltos resuelto.

10

**[30]** Ambos bloques 465 y 475 proceden al bloque 480, donde los datos de instrucción se almacenan en la I-caché. De lo contrario, el contador del programa se incrementa secuencialmente. El procedimiento 400 luego avanza al bloque 410 para consultar la siguiente dirección según lo especificado por el contador del programa.

15

**[31]** Volviendo al bloque 420, si hay un acierto en la I-caché, lo que significa que hay una entrada correspondiente en la I-caché, el procedimiento 400 avanza al bloque 425. En el bloque 425, el procedimiento 400 determina si la entrada correspondiente en la I-caché tiene una dirección de destino de saltos. En una realización, el bloque 425 puede realizarse interrogando un bit válido correspondiente, tal como uno almacenado en la columna de bits válida 234 de la FIG. 2. Si no hay una dirección de destino de saltos válida asociada con el acierto de la I-caché, la dirección de instrucción no es una instrucción de saltos indirecta. De este modo, el procedimiento 400 avanza al bloque 445, donde la instrucción sin saltos se recupera desde la memoria I-caché y se procesa de una manera convencional. El procedimiento 400 avanza al bloque 410 para consultar la siguiente dirección desde la I-caché según lo especificado por el contador del programa.

20

25

**[32]** Si hay una dirección de destino de saltos válida correspondiente a la entrada, el procedimiento 400 avanza al bloque 435. En el bloque 435, el procedimiento 400 predice si se debe tomar un almacén de direcciones de destino de saltos en la entrada correspondiente. En una realización, el circuito de predicción de saltos 126 hace esta predicción. Debe observarse que un experto en la técnica apreciará que las técnicas de predicción de saltos pueden estar utilizando estadísticas, heurísticas, parámetros predeterminados y similares. En el bloque 438, el procedimiento 400 prueba si debe tomarse la predicción. Si lo es, el procedimiento 400 avanza al bloque 440 donde la dirección de destino de saltos se envía al contador de programa, tal como la unidad de obtención previa de instrucciones 128 para que las instrucciones empiecen a ser obtenidas desde la dirección de destino de saltos. El procedimiento 400 avanza al bloque 410 para consultar la siguiente dirección según lo especificado por el contador de programa, que en esta trayectoria a través del diagrama de flujo es la dirección de destino de saltos.

30

35

**[33]** Volviendo al bloque 438, si no se toma la trayectoria predicha, el procedimiento 400 avanza al bloque 445 para procesar las instrucciones almacenadas en la memoria caché de una manera convencional. Después del bloque 445, el procedimiento 400 avanza al bloque 410 para consultar la siguiente dirección en el contador de programa que en esta trayectoria a través del diagrama de flujo es la dirección secuencial que sigue a la instrucción de saltos indirectos, tal como la dirección 0x000C en la FIG. 2A, por ejemplo.

40

45

**[34]** La FIG. 5 es un diagrama de flujo que ilustra un procedimiento 500 de gestión de una I-caché en la resolución de saltos real. El procedimiento 500 se realiza para situaciones donde el destino de saltos real de una instrucción de saltos indirectos se calcula por primera vez y donde el mecanismo de predicción, tal como el circuito de predicción de saltos 126 interpreta erróneamente el destino de la instrucción de saltos indirectos. En el bloque 510, se resuelve un destino real de una instrucción de saltos indirectos. Dicha resolución de saltos se puede realizar a medida que la instrucción de saltos indirectos se procesa más adelante en el canal del procesador.

50

**[35]** En el bloque 520, el procedimiento 500 determina si el destino real coincide con el destino predicho o, para el caso en que el destino real se calcula por primera vez, no hay un destino predicho. Si el destino de saltos real coincide con el destino de saltos previsto, ni la BTAC ni la I-caché necesitan actualizarse. El procedimiento 500 avanza al bloque 525 donde espera la siguiente instrucción de saltos antes de proceder de nuevo al bloque 510.

55

**[36]** Si el destino de saltos real no coincide con el destino predicho, el procedimiento 500 detecta un error de lectura y continúa al bloque 530. Al menos dos situaciones pueden dar como resultado una falta de coincidencia. La primera situación puede producirse la primera vez que se ejecuta una instrucción de saltos indirecto después de haber sido almacenada previamente en la BTAC y en la I-caché sin un destino de saltos. En la primera situación, la siguiente instrucción secuencial después de la instrucción de saltos indirecto puede haberse cargado en el canal del procesador. La segunda situación puede producirse si la dirección de destino predicha es

60

65

diferente de la dirección de destino resuelta real. En la segunda situación, la siguiente instrucción tal como se indica en la dirección de destino predicha puede haberse cargado en el canal del procesador.

**[37]** En el bloque 530, el canal del procesador, en cualquier situación, tiene sus instrucciones descargadas del canal para restaurar el canal al punto de saltos que es, en este caso, la dirección de la instrucción de saltos indirecto. En consecuencia, las instrucciones cargadas en el canal del procesador posteriores a la instrucción de saltos indirecto, independientemente de si esas instrucciones se cargaron comenzando desde una dirección de destino predicha o la siguiente dirección secuencial después de la instrucción de saltos indirecto, se eliminarán del canal del procesador. En el bloque 540, la BTAC se actualiza para almacenar la dirección de destino de saltos real en su campo de dirección de destino de saltos, tal como la columna 254.

**[38]** En el bloque 550, se gestiona la I-caché. Por ejemplo, el circuito lógico de control 114 puede incluir lógica para implementar los bloques 550A-550C. Los bloques 550A-550C son realizaciones alternativas del bloque 550. En el bloque 550A, se invalida la línea de caché correspondiente a la dirección de destino predicha. En el bloque 550B, se invalida el destino de saltos de la línea de memoria caché correspondiente a la dirección de destino predicha. En el bloque 550C, la dirección de destino de saltos de la línea de caché correspondiente a la dirección de destino predicha se actualiza para reflejar la dirección de destino real. El bloque 550C es preferible cuando la primera vez que se ejecuta una instrucción de saltos indirecto después de haber sido almacenada previamente en la BTAC y en la I-caché sin un destino de saltos.

**[39]** La FIG. 6 es un diagrama de bloques funcional de dos procesadores 601 y 602 que comparten una BTAC común 611. La BTAC 611 compartida y los procesadores 601 y 602 están preferiblemente integrados en un solo chip de circuito integrado. El procesador 601 incluye su propio canal de procesador de múltiples etapas 620, I-caché 610 y circuito de predicción de saltos 615. En la etapa 625, tiene lugar una resolución de saltos real para actualizar la BTAC 611 y la I-caché 610 con una instrucción de saltos indirecto y una dirección de destino de saltos como se describió previamente en las Figs. 4 y 5, respectivamente. El procesador 602 incluye su propio canal de procesador de múltiples etapas 640, I-caché 630 y circuito de predicción de saltos 635. En la etapa 645, tiene lugar una resolución de saltos real para actualizar la BTAC 611 y la I-caché 630 con una instrucción de saltos indirecto y una dirección de destino de saltos como se describió previamente en las Figs. 4 y 5, respectivamente. Como se muestra, la resolución del salto real se puede realizar en diferentes etapas en el canal de los procesadores. Aunque la FIG. 6 ilustran una BTAC común compartida entre dos procesadores, se observa que la presente divulgación contempla compartir una BTAC común entre tres o más procesadores. Durante la operación, cuando se obtiene una dirección de I-caché 610 o I-caché 630 que contiene una dirección de saltos válida, los respectivos canales 620 y 640 comienzan las instrucciones de obtención previa comenzando desde la dirección de destino de saltos.

**[40]** Los diversos bloques lógicos, módulos, circuitos, elementos y/o componentes ilustrativos descritos en relación con las realizaciones dadas a conocer en el presente documento pueden implementarse o realizarse con un procesador de propósito general, con un procesador de señales digitales (DSP), con un circuito integrado específico de la aplicación (ASIC), con una formación de compuertas programables en el terreno (FPGA) o con otro componente de lógica programable, lógica de transistor o de compuertas discretas, componentes de hardware discretos, o con cualquier combinación de los mismos diseñada para realizar las funciones descritas en el presente documento. Un procesador de propósito general puede ser un microprocesador pero, de forma alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también puede implementarse como una combinación de componentes informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo.

**[41]** Los procedimientos descritos en relación con las realizaciones divulgadas en el presente documento pueden realizarse directamente en hardware, en un módulo de software ejecutado por un procesador o en una combinación de los dos. Un módulo de software puede residir en una memoria RAM, una memoria flash, una memoria ROM, una memoria EPROM, una memoria EEPROM, registros, un disco duro, un disco extraíble, un CD-ROM o en cualquier otra forma de medio de almacenamiento conocida en la técnica. Un medio de almacenamiento puede estar acoplado al procesador de manera que el procesador pueda leer información de, y escribir información en, el medio de almacenamiento. De forma alternativa, el medio de almacenamiento puede estar integrado en el procesador.

**[42]** Aunque la invención se da a conocer en el contexto de realizaciones, se reconocerá que una amplia variedad de implementaciones puede ser empleada por personas de medianamente expertas en la técnica, en consonancia con la exposición anterior y las reivindicaciones a continuación.

**REIVINDICACIONES**

- 5           1.    Un procedimiento para reducir las búsquedas en una memoria caché de direcciones de destino de saltos (BTAC), que comprende:
- recuperar una dirección de destino de saltos desde la BTAC en respuesta a una falta al buscar una primera dirección de instrucción en una memoria caché de instrucciones (I-caché);
- 10           almacenar la dirección de destino de saltos desde la BTAC en la I-caché, estando la dirección de destino de saltos asociada con la primera dirección de instrucción en la I-caché;
- buscar la primera dirección de instrucción en la I-caché; y
- 15           recuperar la dirección de destino de saltos asociada con la primera dirección de instrucción de la I-caché.
2.    El procedimiento de acuerdo con la reivindicación 1, en el que la recuperación de la dirección de destino de saltos desde la BTAC no se produce en respuesta a un acierto en la I-caché.
- 20           3.    El procedimiento, de acuerdo con la reivindicación 1, que comprende además:
- obtener la primera instrucción asociada con la primera dirección de instrucción;
- 25           predecir que una segunda instrucción que se va a obtener está asociada con la dirección de destino de saltos; y
- obtener la segunda instrucción basada en la dirección de destino de saltos.
- 30           4.    El procedimiento de acuerdo con la reivindicación 3, que comprende además:
- detectar un error de lectura para la primera instrucción; y
- 35           administrar la I-caché mediante al menos uno de invalidar una línea de I-caché, invalidar la dirección de destino de saltos en la línea de I-caché y actualizar la dirección de destino de saltos en la I-caché.
- 40           5.    El procedimiento de acuerdo con la reivindicación 4, en el que la detección de una predicción errónea para la primera instrucción comprende:
- calcular una dirección de destino de saltos real para la primera instrucción; y
- 45           comparar la dirección de destino de saltos real con la dirección de destino de saltos en la I-caché, en el que la dirección de destino de saltos real no coincide con la dirección de destino de saltos.
- 50           6.    Un procedimiento para almacenar instrucciones de saltos en una memoria caché de instrucciones (I-caché), que comprende:
- buscar una dirección de instrucción de saltos en una memoria caché de instrucciones (I-caché);
- 55           recuperar una dirección de destino de saltos desde una memoria caché de direcciones de destino de saltos (BTAC) en respuesta a una falta de la I-caché; y
- almacenar la dirección de destino de saltos en una entrada en la I-caché.
- 60           7.    El procedimiento de acuerdo con la reivindicación 6, que comprende además:
- recuperar la dirección de destino de saltos desde la I-caché en respuesta a un acierto de la I-caché.
- 65           8.    El procedimiento de acuerdo con la reivindicación 7, que comprende además:
- predecir que una siguiente instrucción que se va a obtener está asociada con la dirección de destino de saltos.
9.    El procedimiento de acuerdo con la reivindicación 6, que comprende además:
- actualizar la dirección de destino de saltos en la I-caché en una predicción errónea.

10. El procedimiento de acuerdo con la reivindicación 9, en el que la actualización de la dirección de destino de saltos en la entrada en una predicción errónea comprende además:
- 5           determinar una dirección de saltos real de la instrucción de saltos; y
- comparar la dirección de saltos real con la dirección de destino de saltos, en el que la resolución de saltos real no coincide con la dirección de destino predicha.
11. Un sistema para reducir las búsquedas en una memoria caché de direcciones de destino de saltos (BTAC), que comprende:
- 10           una memoria caché de direcciones de destino de saltos (BTAC); y
- una memoria caché de instrucciones (I-caché) configurada para recuperar una dirección de destino de saltos desde la BTAC en respuesta a una falta de memoria caché al buscar una primera dirección de instrucción de saltos, estando la I-caché configurada además para almacenar la dirección de destino de saltos, en el que la I-caché está configurada además para tener la primera dirección de instrucción de saltos en la I-caché buscada y para tener la dirección de destino de saltos asociada con la primera dirección de instrucción de saltos recuperada desde la I-caché.
12. El sistema de acuerdo con la reivindicación 11, en el que la memoria caché de instrucciones (I-caché) no está configurada para recuperar la dirección de destino de saltos de la BTAC en respuesta a un acierto.
13. El sistema de acuerdo con la reivindicación 11, en el que la I-caché está configurada para tener la primera instrucción asociada con la primera dirección de instrucción obtenida, comprendiendo además el sistema: un circuito de predicción de saltos configurado para predecir que una segunda instrucción a obtener está asociada con la dirección de destino de saltos.
14. El sistema de acuerdo con la reivindicación 13, en el que la I-caché está configurada para tener la segunda instrucción basada en la dirección de destino de saltos obtenida desde la I-caché.
15. El sistema de acuerdo con la reivindicación 14, que comprende además:
- 35           un circuito lógico de actualización configurado para predecir de manera errónea la segunda instrucción a obtener; y
- un circuito lógico de control configurado para administrar la I-caché mediante al menos uno de invalidar una línea de I-caché, invalidar la dirección de destino de saltos en la línea de I-caché y actualizar la dirección de destino de saltos en la I-caché.

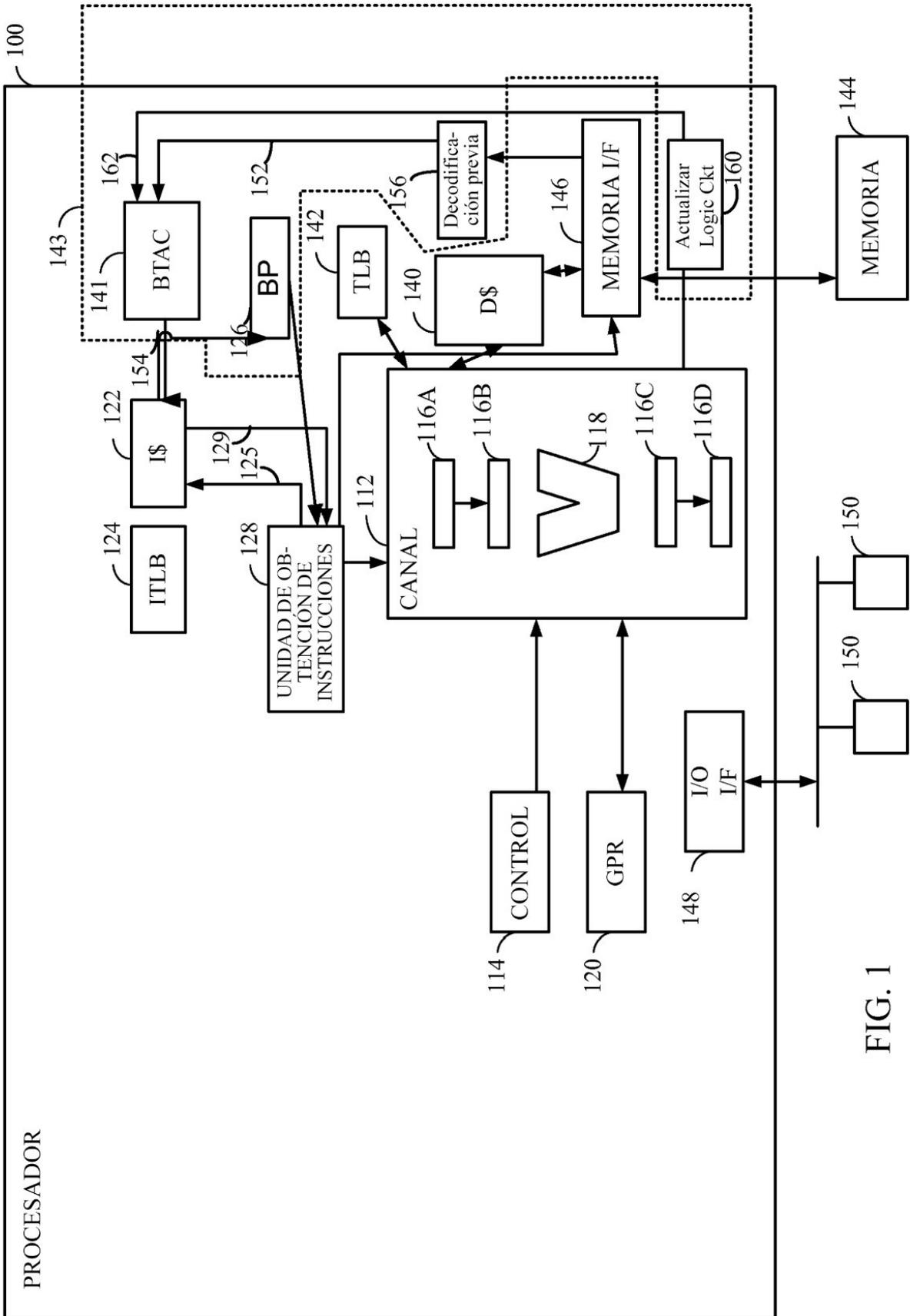


FIG. 1

210  
 212  
 0x000A: Añadir  
 0x000B: Salto [R1]  
 0x000C: Carga X  
 .  
 .  
 0x2000: Mov  
 FIG. 2A

230

232	MULT	0	
	AÑADIR	0	
0x000B		0	
	⋮	⋮	⋮
	CARGAR	0	

234 236

FIG. 2B

252	250	254
256	0x000B	0x2000

FIG. 2C

230

	MULT	0	
	AÑADIR	0	
0x000B	Salto	1	0x2000
	⋮	⋮	⋮
	CARGAR	0	

FIG. 3

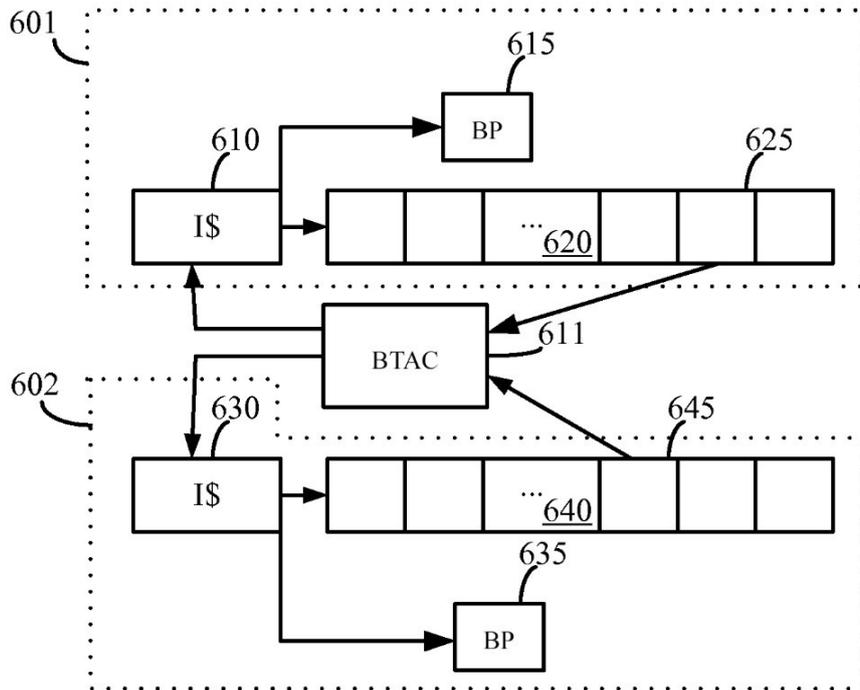


FIG. 6

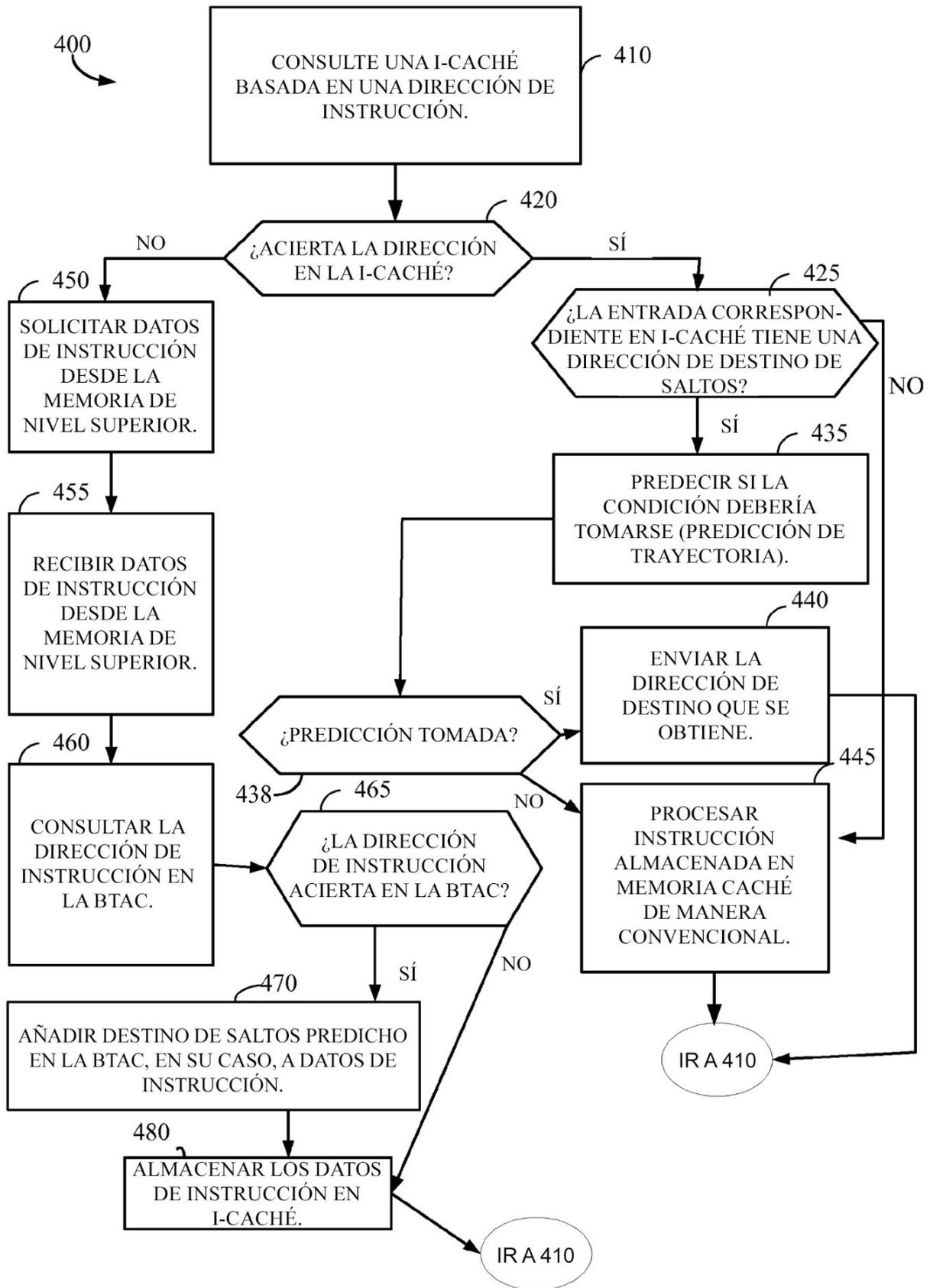


FIG. 4

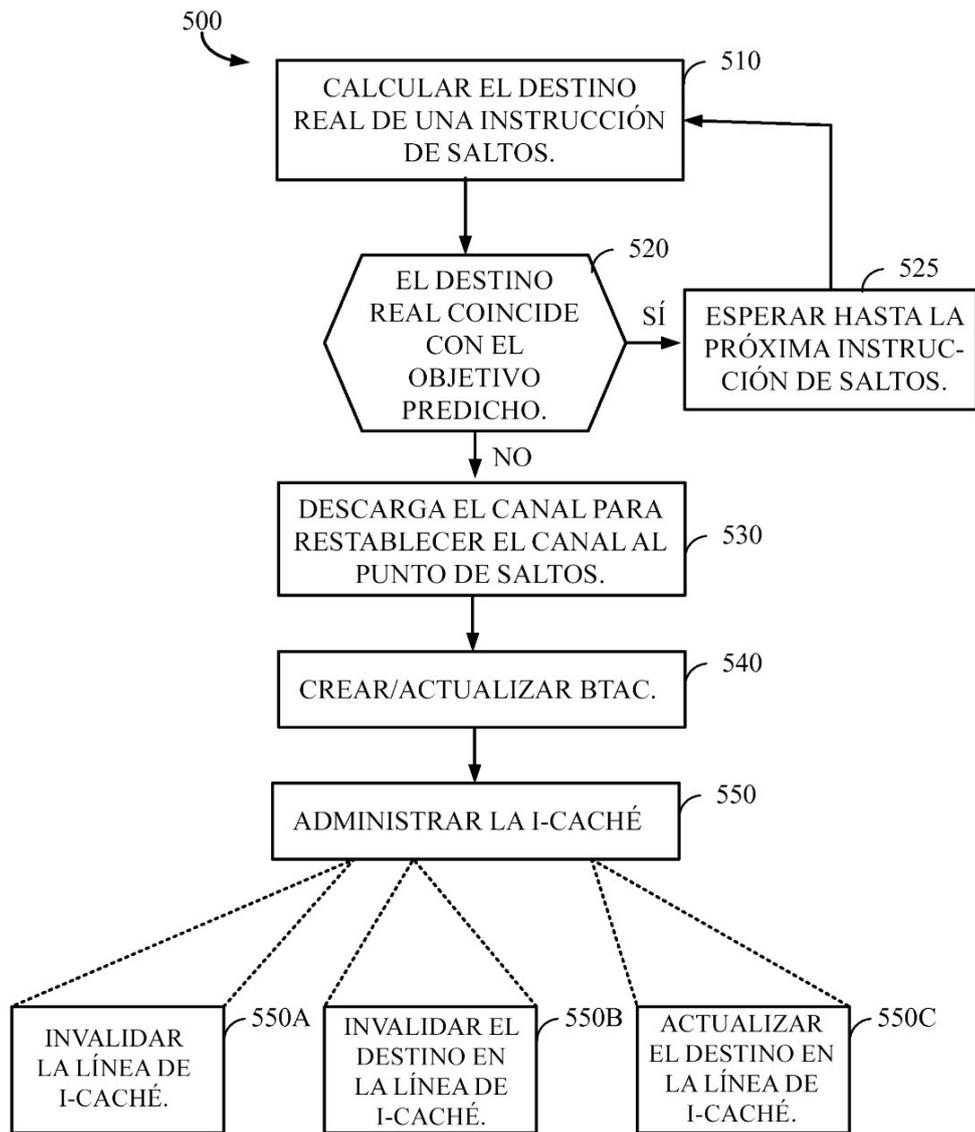


FIG. 5