

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 388 309**

51 Int. Cl.:
G06F 9/38

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07758843 .2**

96 Fecha de presentación: **19.03.2007**

97 Número de publicación de la solicitud: **1999578**

97 Fecha de publicación de la solicitud: **10.12.2008**

54 Título: **Representar saltos de bucle en un registro de historia de saltos con múltiples bits**

30 Prioridad:
17.03.2006 US 378712

45 Fecha de publicación de la mención BOPI:
11.10.2012

45 Fecha de la publicación del folleto de la patente:
11.10.2012

73 Titular/es:
**QUALCOMM INCORPORATED
ATTN: INTERNATIONAL IP ADMINISTRATION
5775 MOREHOUSE DRIVE
SAN DIEGO, CA 92121, US**

72 Inventor/es:
**DIEFFENDERFER, James Norris y
RYCHLIK, Bohuslav**

74 Agente/Representante:
Fàbrega Sabaté, Xavier

ES 2 388 309 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

Representar saltos de bucle en un registro de historia de saltos con múltiples bits

CAMPO

5 La presente invención se refiere en general al campo de los procesadores y, en particular a un procedimiento de representación de los saltos de bucle en un registro de historia de saltos con múltiples bits.

ANTECEDENTES

10 Los microprocesadores realizan tareas de cálculo en una amplia variedad de aplicaciones. Un rendimiento mejorado del procesador es casi siempre deseable, para permitir un funcionamiento más rápido y/o mayor funcionalidad a través de cambios en el software. En aplicaciones empotradas comunes, tales como dispositivos electrónicos portátiles es también deseable ahorrar energía.

15 Los procesadores modernos comunes utilizan una arquitectura con línea de ejecución, en la que instrucciones secuenciales, cada una con varias etapas de ejecución, se solapan durante la ejecución. Para un máximo rendimiento, las instrucciones deben fluir continuamente a través de la línea de ejecución. Cualquier situación que provoque que las instrucciones se paralicen en la línea de ejecución afecta negativamente al rendimiento. Si las instrucciones deben ser eliminadas de la línea de ejecución y, a continuación, volver a cargarlas, el rendimiento y el consumo de energía pueden verse afectados.

20 Normalmente todos los programas del mundo real incluyen instrucciones de salto condicional, el comportamiento de salto real de las cuales no se conoce, normalmente, hasta que la instrucción se evalúa dentro de la línea de ejecución. Para evitar una paralización que pueda resultar de esperar por la evaluación real de la instrucción de salto, los procesadores modernos comunes utilizan alguna forma de predicción de saltos, por la cual el comportamiento de salto de las instrucciones de salto condicional se prevé al principio de la línea de ejecución. Sobre la base de la evaluación prevista de salto, el procesador carga de forma especulativa (precarga) y ejecuta las instrucciones de una dirección prevista, ya sea la dirección de destino de salto (si se prevé que se saltará) o la siguiente dirección secuencial después de la instrucción de salto (si se prevé que no se saltará). Una vez que se determina el comportamiento real del salto, si el salto se predijo incorrectamente, las instrucciones cargadas de forma especulativa deben ser eliminadas de la línea de ejecución, y deben cargarse nuevas instrucciones a partir de la siguiente dirección correcta. Precargar instrucciones en respuesta a una predicción incorrecta de salto impacta de manera negativa el rendimiento del procesador y el consumo de energía. En consecuencia, es deseable mejorar la precisión de predicción de saltos.

30 Las técnicas conocidas de predicción de salto incluyen tanto predicciones estáticas como dinámicas. El comportamiento probable de algunas instrucciones de salto puede ser previsto de forma estática por un programador y/o compilador. Un ejemplo es una rutina de comprobación de errores. El código común se ejecuta correctamente, y los errores son escasos. Por lo tanto, la instrucción de salto que implementa una función de "salto en caso de error" evaluará "no se saltará" un porcentaje muy alto de las veces. Tal instrucción puede incluir un bit de predicción estática de predicción estática de saltos en el código de operación, fijada por un programador o compilador con conocimiento del resultado común probable de la condición de salto.

40 La predicción dinámica se basa generalmente en la historia de evaluación de saltos (y en algunos casos, la precisión de la historia de evaluación de saltos) de la instrucción que se está prediciendo y/u otras instrucciones de salto en el mismo código. Un análisis exhaustivo de código real indica que los patrones de evaluación de salto del pasado reciente pueden ser un buen indicador de la evaluación de las futuras instrucciones de salto.

45 Una forma conocida de predicción de salto dinámica, representada en la Figura 1, utiliza un Registro de Historia de Saltos (BHR) 100 para almacenar las n evaluaciones de salto anteriores. En una implementación simple, el BHR 30 comprende un registro de desplazamiento. El resultado común de evaluación de salto reciente se introduce (por ejemplo, un 1 indicando se hizo el salto y un 0 indicando que no se hizo se hizo el salto), siendo desplazada la evaluación pasada más antigua en el registro. Un procesador puede mantener un BHR local 100 por cada instrucción de salto. Alternativamente (o adicionalmente), un BHR 100 puede contener las evaluaciones anteriores recientes de todas las instrucciones de salto condicional, a veces conocidos en la técnica como BHR global, o GHR. Tal y como se usa en este documento, BHR denomina tanto a Registros de Historia de Saltos locales como globales.

50 Tal y como se muestra en la Figura 1, el BHR 100 puede indexar una Tabla de Predicción de Saltos (BPT) 102, que puede ser también local o global. El BHR 100 puede indexar la BPT 102 directamente, o puede combinarse con otra información, como por ejemplo el contador de programa (PC) de la instrucción de salto en la lógica de indexación de

BPT 104. Además, se pueden utilizar otras entradas a la lógica de indexación de BPT 104. La lógica de indexación de BPT 104 puede concatenar las entradas (conocido en la técnica como gselect), realizar una operación OR-Exclusiva de las entradas (gshare), lleve a cabo una función hash, o combinar o transformar los datos de una variedad de maneras.

- 5 Como un ejemplo, la BPT 102 puede comprender una pluralidad de contadores de saturación, los MSB de los cuales sirven como predictores de salto bimodales. Por ejemplo, cada entrada de la tabla puede comprender un contador de 2 bits que asume uno de cuatro estados, cada uno asignado un valor ponderado de predicción, tales como:

11 - Muy esperado se saltará

10 - Débilmente esperado se saltará

- 10 01 - Débilmente esperado no se saltará

00 - Muy esperado no se saltará

- 15 El contador se incrementa cada vez que una instrucción de salto correspondiente evalúa "se saltará" y disminuye cada vez que la instrucción se evalúa "no se saltará." El MSB del contador es un predictor de saltos bimodal, predecirá si se saltará o no se saltará, independientemente de la fuerza o el peso de la predicción subyacente. Un contador de saturación reduce el error de predicción de una dirección infrecuente de evaluación de salto. Un salto que se evalúa de forma consistente saturará el contador. Una evaluación infrecuente, por otro lado, alterará el valor del contador (y la fuerza de la predicción), pero no el valor de predicción bimodal. Por lo tanto, una evaluación infrecuente sólo predecirá incorrectamente una vez, no dos veces. La tabla de contadores de saturación es un ejemplo únicamente ilustrativo, en general, una BHT puede indexar una tabla que contiene una variedad de
- 20 mecanismos de predicción de salto.

- Independientemente del mecanismo de predicción de saltos empleado en la BPT 102, el BHR 100 - ya sea solo o en combinación con otra información tal como el PC de la instrucción de salto - indexa la BPT 102 para obtener predicciones de salto. Almacenando las evaluaciones de salto anteriores en el BHR 100 y utilizando las evaluaciones en la predicción de saltos, la instrucción de salto que está prediciendo se correla con el
- 25 comportamiento del salto en el pasado - su propio comportamiento en el pasado en el caso de un BHR local 100 y con el comportamiento de otras instrucciones de salto en el caso de un BHR global 100. Esta correlación es la clave para predicciones exactas de saltos, al menos en el caso de código altamente repetitivo.

- Nótese que la Figura 1 muestra las evaluaciones de salto almacenándose en el BHR 100 - es decir, la evaluación real de una instrucción de salto condicional, que sólo puede ser conocida una vez la línea de ejecución ha avanzado bastante, tal como en una etapa de ejecución de la línea de ejecución. Mientras que esto es el resultado final, en la práctica, los procesadores comunes de alto rendimiento almacenan la evaluación de salto prevista a partir de la BPT
- 30 102 en el BHR 100, y corrigen el BHR 100 más adelante como parte de una operación de recuperación de predicción errónea si la predicción resulta ser errónea. Las figuras de los dibujos no reflejan esta característica de la aplicación, para una mayor claridad.

- 35 Una estructura de código común que puede reducir la eficacia de un predictor de saltos que emplea un BHR 100 es el bucle. Un bucle termina con una instrucción de salto condicional que evalúa una condición de final de bucle, por ejemplo, si una variable de índice que se incrementa cada vez durante el bucle ha alcanzado un valor de final de bucle. Si no, la ejecución salta hacia atrás al comienzo del bucle para otra iteración, y otra evaluación de salto condicional de final de bucle.

- 40 Si el bucle se ejecuta durante un gran número de iteraciones, los saltos ejecutados hacia atrás de la instrucción de salto de final de bucle saturan parcial o totalmente el BHR 100. Cuando el número de iteraciones del bucle es igual o superior a la anchura del BHR 100, al final del bucle una BHR de n bits contendrá precisamente $n-1$ unos (se saltó), seguido de un solo cero (no se saltó), que corresponde a una larga serie de evaluaciones de saltos tomados como resultado de las iteraciones del bucle, y termina con una única evaluación de salto "no se saltó" cuando el bucle
- 45 termina. Esto destruye a nivel práctico la eficacia del BHR 100, ya que todas las correlaciones con las evaluaciones anteriores (tanto para un BHR 100 local como global) se pierden. En este caso, el BHR 100 es probable que asigne a la misma entrada de la BPT 102 una instrucción de salto dada (dependiendo de las otras entradas a la lógica de indexación de BPT 104), en lugar de a una entrada que contiene una predicción de salto que refleja la correlación de la instrucción de salto con evaluaciones de salto anteriores.

- 50 Además, el BHR 100 saturado puede aumentar los solapamientos en la BPT 102. Es decir, todas las instrucciones de salto que siguen a bucles con iteraciones comunes se asignarán a la misma entrada de la BPT 102, si el BHR 100 indexa directamente la BPT 102. Aun cuando el BHR 100 se combina con otra información, la posibilidad de

solapamiento es mayor. Esto afecta negativamente a la precisión de predicción no sólo para la instrucción después del bucle, sino también a todas las instrucciones de salto que se solapan con su entrada en la BPT 102.

5 Si el bucle se ejecuta durante un menor número de iteraciones que la anchura del BHR 100, el BHR 100 no está saturado y se mantiene parte de la historia de evaluación de saltos previa. Sin embargo, los bits que representan la historia de evaluación de saltos anterior se encuentran desplazados en el BHR 100 por numerosos resultados "se saltó" de la instrucción de final de bucle. En particular, cuando el número de iteraciones del bucle varía, esto tiene dos efectos perjudiciales sobre la predicción de saltos. En primer lugar, la instrucción de salto corresponderá con un número mucho mayor de entradas en la BPT 102 para capturar la misma correlación con las evaluaciones anteriores de salto, lo que requiere una BPT 102 mayor para alcanzar la misma precisión para el mismo número de instrucciones de salto que sería necesario sin el salto de final de bucle que afecta al BHR 30. En segundo lugar, los predictores de saltos en la BPT 102 necesitarán más tiempo para "ser entrenados" incrementando la cantidad de código que debe ejecutar antes de que la BPT 102 comience a proporcionar predicciones exactas de saltos.

15 La solicitud de patente 11/066,508, asignada al titular de la presente solicitud e incorporada aquí por referencia, se propone suprimir la actualización del BHR 100 para instrucciones de salto de final de bucle. Esto resuelve normalmente los efectos perjudiciales de la saturación total o parcial del BHR 100 en la precisión de predicción. Sin embargo, no logra captar y explotar las correlaciones que puedan existir entre el comportamiento del bucle y la evaluación de saltos posterior.

20 En aplicaciones comunes, la evaluación de una instrucción de salto puede estar correlada con el número de iteraciones de un bucle anterior. Por ejemplo, un programa científico puede capturar puntos de datos en un bucle y, siguiendo el bucle, saltar a una subrutina de análisis estadístico en la que se analizan los datos capturados. Sin embargo, si el bucle itera relativamente pocas veces, capturar pocos puntos de datos alcanzando una pequeña muestra, el análisis estadístico puede ser poco fiable, y puede omitirse. En este caso, la evaluación de la instrucción condicional saltando a la subrutina de análisis estadístico está fuertemente correlada con el número de iteraciones del bucle de adquisición de datos.

25 En otro ejemplo, una aplicación puede utilizar una estructura de bucle para buscar a lo largo de una lista, registro de transacciones, archivo de historia, o estructura de datos similar. Si un ítem que coincide con los parámetros de búsqueda aparece con frecuencia en la lista, se requerirán relativamente pocas iteraciones del bucle para localizar el ítem. En consecuencia, un ítem que aparece con frecuencia puede requerir un gran número de iteraciones del bucle de búsqueda. La evaluación de una instrucción de salto posterior puede correlarse con la frecuencia con la que un ítem en particular aparece en la lista y, por tanto, correlada con el número de iteraciones del bucle requeridas para localizar el ítem.

30 Suprimir la actualización del BHR 100 en respuesta a las evaluaciones de instrucción de salto de final de bucle no permite capturar ninguna correlación entre el número de iteraciones del bucle y el comportamiento del salto de una instrucción de salto posterior. Por otro lado, mantener una historia completa de las evaluaciones de instrucciones de salto de final de bucle satura total o parcialmente el BHR 100, perdiendo la correlación con evaluaciones de instrucciones de salto anteriores al bucle.

RESUMEN

40 En respuesta a una propiedad de una instrucción de salto condicional asociada con un bucle, tal como una propiedad que indica que el salto es un salto de final de bucle, se mantiene un recuento del número de iteraciones del bucle, y se almacena en el BHR un valor de múltiples bits indicativo del recuento de iteraciones del bucle. En una realización, el valor de múltiples bits puede comprender el recuento real de bucle, en cuyo caso el número de bits es variable. En otra realización, el número de bits es fijo (por ejemplo, dos) y el recuento de iteraciones de bucle se asigna a un valor de múltiples bits mediante comparación con umbrales. Pueden mantenerse recuentos separados de iteraciones para bucles anidados, y un valor de múltiples bits almacenado en el BHR puede indicar un número de iteraciones de bucle únicamente de un bucle interior, sólo del bucle exterior, o ambas cosas.

Una realización se refiere a un procedimiento de predicción de saltos. Se identifica una característica de una instrucción de salto asociada con un bucle. En respuesta a la propiedad, se almacena un valor de múltiples bits en un BHR, el valor de múltiples bits indicativo del número de iteraciones de un bucle asociado con la instrucción de salto.

50 Otra realización se refiere a un procesador. El procesador incluye un predictor de saltos para predecir la evaluación de las instrucciones de salto condicional, y una línea de ejecución de instrucciones operable para cargar y ejecutar instrucciones de forma especulativa en base a una predicción del predictor de saltos. El procesador también incluye un BHR operable para almacenar la evaluación de las instrucciones de salto condicional, y un contador de bucle operable para contar el número de iteraciones de un bucle de código. El procesador incluye además un circuito de

control operable para almacenar en el BHR un valor de múltiples bits indicativo del número de iteraciones de un bucle asociado con una instrucción de salto condicional.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es un diagrama de bloques funcional de un circuito de predicción de saltos de la técnica anterior.

5 La Figura 2 es un diagrama de bloques funcional de un procesador.

La Figura 3 es un diagrama de flujo de un procedimiento de ejecución de una instrucción de salto.

La Figura 4 es un diagrama de bloques funcional de un circuito de predicción de saltos que incluye uno o más registros de PC de Último Salto y Contadores de Bucle.

DESCRIPCIÓN DETALLADA

10 La Figura 2 representa un diagrama de bloques funcional de un procesador 10. El procesador 10 ejecuta las instrucciones en una línea de ejecución de instrucciones 12 de acuerdo con la lógica de control 14. En algunas realizaciones, la línea de ejecución 12 puede ser un diseño superescalar, con múltiples líneas de ejecución. La línea de ejecución 12 incluye diversos registros o biestables de tipo cerrojo 16, organizados en etapas de línea de ejecución, y una o más unidades de aritmético-lógicas (ALU) 18. Un archivo de Registro de Propósito General (GPR) 20 proporciona registros que comprenden la parte superior de la jerarquía de memoria.

La línea de ejecución 12 carga las instrucciones en una caché de instrucciones (I-cache) 22, siendo la traducción de direcciones y los permisos de memoria administrados por una memoria temporal de traducción de instrucciones adyacentes (ITLB) 24. Cuando las instrucciones condicionales de salto se decodifican temprano en la línea de ejecución 12, un predictor de saltos 26 predice el comportamiento del salto, y proporciona la predicción a una unidad de precarga de instrucciones 28. La unidad de precarga de instrucciones 28 carga de forma especulativa instrucciones de la caché-1 22, en una dirección de destino de salto de calculada en la línea de ejecución 12 para predicciones de salto (se saltará), o en la siguiente dirección secuencial para saltos previstos "no se saltará." En cualquier caso, las instrucciones precargadas se cargan en la línea de ejecución 12 para ejecución especulativa.

20 El predictor de saltos 26 incluye un Registro de Historia de Saltos (BHR) 30, una Tabla de Predicción de saltos (BPT) 32, lógica de indexación de BPT 34, y lógica de actualización 36 de BHR. El predictor de saltos 26, además, pueden incluir uno o más registros PC de Último Salto 38 y uno o más contadores de bucle 37 que proporcionan entradas a la lógica de actualización 36 de BHR.

30 A los datos se accede desde una caché de datos (D-cache) 40, con traducción y permisos de direcciones gestionados por una memoria temporal de traducción de búsqueda adyacente (TLB) 42. La TLB 42 puede ser una TLB de datos dedicada, o puede comprender una TLB integrada que gestiona traducciones y permisos de direcciones para instrucciones y datos. Además, en diversas realizaciones del procesador 10, la I-caché 22 y la D-caché 40 pueden estar integradas, o unificada. Fallos en la I-caché 22 y/o la D-caché 40 causan un acceso a la memoria principal (fuera del chip) 44, bajo el control de una interfaz de memoria 46.

35 El procesador 10 puede incluir un interfaz de entrada/salida (I/O) 48, para controlar el acceso a los diversos dispositivos periféricos 50, 52. Los expertos en la técnica reconocerán que son posibles numerosas variaciones del procesador 10. Por ejemplo, el procesador 10 puede incluir una caché de segundo nivel (L2) para cada una o ambas cachés I y D cachés 22, 40. Además, uno o más de los bloques funcionales representados en el procesador 10 puede omitirse de una realización particular.

Indicaciones de múltiples bits de iteraciones del bucle

40 De acuerdo con una o más realizaciones, la precisión de predicción se mejora correlando la evaluación de saltos tanto con las evaluaciones anteriores de saltos y el recuento de las iteraciones de los bucles anteriores. Esto se logra almacenando de un valor de múltiples bits indicativo del recuento de iteraciones del bucle en el BHR 30. Asignando recuentos de iteraciones de bucle potencialmente grandes a relativamente pocos bits, la información característica sobre el bucle se conserva, al tiempo que se evita que los saltos de final de bucle saturan total o parcialmente uno o más BHR 30 en el predictor de saltos 26.

45 Este proceso se representa como un diagrama de flujo en la Figura 3. Se carga una instrucción y se decodifica (bloque 52). Si la instrucción es una instrucción de salto no condicional (bloque 53), se prosigue la línea de ejecución y el proceso vuelve a cargar la siguiente instrucción (bloque 52). Si la instrucción es una instrucción de salto condicional (bloque 53), se hace una determinación de si el salto es un salto de final de bucle (bloque 54). Si no,

cuando se evalúa el salto (tal como en una etapa de ejecución de la línea de ejecución) el BHR 30 se actualiza para registrar la evaluación de salto utilizando un único bit (bloque 56), es decir, si la instrucción de salto se evaluó como "se saltará" o "no se saltará". La ejecución continúa entonces (bloque 64) en la dirección de destino del salto o en la siguiente dirección secuencial, respectivamente. Si el salto es un salto de final de bucle (bloque 54), se incrementa un recuento de iteraciones de bucle (bloque 58). Cuando finaliza el bucle (bloque 60), un valor de múltiples bits indicativo del número de iteraciones del bucle se escribe en el BHR 30 (bloque 62). El contador de iteraciones del bucle se borra (bloque 63) y la ejecución continúa (bloque 64) en la siguiente dirección secuencial.

Tanto la consulta (bloque 54) - es decir, la identificación de una instrucción de salto como una instrucción de salto de final de bucle - y mantener el recuento de iteraciones del bucle (bloque 58) puede llevarse a cabo de una variedad de formas. En una realización, uno o más contadores de bucle (LC) 37 se incrementan cuando una instrucción de salto condicional que se determina que es un salto de final de bucle se evalúa "se saltará". El valor del LC 37, o un valor derivado del mismo (como se discute con más detalle en este documento), se escribe en el BHR 30 cuando el salto de final de bucle evalúa "no se saltará", indicando una salida del bucle. El LC 37 también se borra en ese momento.

15 Identificar instrucciones de salto de final de bucle

Los bucles iteran saltando hacia atrás desde el final del bucle al principio del bucle. De acuerdo con una forma según la realización, cada instrucción de salto condicional con una dirección de destino de salto inferior a la dirección de la instrucción de salto, o PC - es decir, un salto hacia atrás - se supone que es una instrucción de fin de bucle. Este procedimiento de identificación de saltos de fin de bucle ofrece simplicidad. Como se muestra en la Figura 4, el PC de la instrucción de salto se compara con la dirección de destino en la lógica de Detección de Salto de Final de Bucle (BTA) 35 cuando la instrucción de salto se evaluó finalmente en la línea de ejecución. Si BTA < PC, se incrementa un contador del bucle (LC) 37. Esta realización requiere una comparación de direcciones cuando se determina la dirección de destino del salto, y pueden sufrir alguna inexactitud, ya que no todos los saltos hacia atrás son saltos de final de bucle.

Otra forma de detectar un salto de final de bucle es reconocer la ejecución repetida de la misma instrucción de salto. En una realización, como se muestra en la Figura 4, un registro PC de Último Salto (LBPC) 38 almacena el PC de la última instrucción de salto ejecutada. En el caso de un bucle simple, si el PC de una instrucción de salto coincide con el LBPC 38 en la lógica de Detección de Salto de Final de Bucle 35 - es decir, la instrucción de salto fue la última instrucción de salto evaluada - se supone que la instrucción de salto es una instrucción de salto de final de bucle, y se incrementa un LC 37.

El código que incluye instrucciones de salto condicional dentro de un bucle puede complicar la detección de saltos de final de bucle. Por ejemplo, los algoritmos comunes para buscar en un vector o en una lista enlazada crean un bucle. Un salto dentro del bucle comprueba si hay una coincidencia y sale del bucle antes de tiempo si la encuentra, mientras que un salto de final de bucle comprueba el final del vector o lista enlazada. En este caso, un único LBPC 38 no puede detectar el salto de final de bucle, ya que será sobrescrito de forma alternativa por los dos PCs de salto. Por consiguiente, en una forma según la realización, pueden proporcionarse dos o más registros LBPC 38 (como se representa en la Figura 4), con los PCs de instrucciones de salto evaluadas sucesivamente almacenados en los registros LBPC correspondientes (LBPC₀, LBPC₁, ... LBPC_M) 38. Esto permite la detección de saltos de final de bucle, incluso en presencia de otros saltos condicionales dentro del bucle.

Las instrucciones de salto de final de bucle también pueden ser marcadas de forma estática por un compilador o ensamblador. En una realización, un compilador genera un tipo particular de instrucción de salto que sólo se utiliza para saltos de final de bucle, por ejemplo, "BRLP". La instrucción BRLP es reconocida, y un se incrementa LC 37 cada vez que una instrucción BRLP es evaluada como "se saltará" en una etapa de la línea de ejecución. En otra realización, un compilador o ensamblador puede incrustar una indicación salto de final de bucle en una instrucción de salto, tal como poniendo a '1' uno o más bits predefinidos en el código de operación de la instrucción. Los bits de salto de final de bucle se detectan, y se incrementa un LC 37 cuando la instrucción de salto se evalúa como "se saltará" en una etapa de la línea de ejecución. En general, se puede proporcionar un Indicador de Final de Bucle a la lógica de Detección de Salto de Final de Bucle 35. La identificación estática de saltos de final de bucle reduce el hardware y la complejidad computacional, trasladando la función de identificación de final de bucle al compilador o al ensamblador.

Generar y almacenar un indicador del número de iteraciones del bucle

Independientemente de la técnica utilizada para identificar una instrucción de salto de final de bucle (ya sea para un bucle interior o exterior), de acuerdo con una o más realizaciones, una indicación del número de veces que itera el bucle se almacena en el BHR 30, para explotar cualquier correlación entre el número de iteraciones del bucle y la evaluación de la instrucción de salto condicional posterior.

En una realización, el recuento real de iteraciones del bucle se almacena en el BHR 30. Esto es, el recuento de múltiples bits de, por ejemplo, un LC 37 se introduce en el BHR 30, desplazando las evaluaciones anteriores de saltos en el MSB del BHR 30. Para bucles de gran tamaño, esta cuenta necesitará $\log_2(N)$ bits para n iteraciones del bucle, desplazando un número correspondiente de evaluaciones anteriores de salto. Sin embargo, incluso para recuentos de iteraciones de bucle grandes, la representación binaria es una compresión significativa del comportamiento del salto de final de bucle almacenado en los registros de historia de salto de la técnica anterior (es decir, $n-1$ unos seguidos por un solo cero). En una realización, el número de iteraciones se acumula en un LC 37 y se introduce en el BHR 30. Por lo menos en otra realización (no mostrada), al menos los k bits de menor peso del BHR 30 pueden implementarse como un registro/contador de modo dual, y el contador de iteraciones del bucle puede ser incrementado directamente en el BHR 30 (con los contenidos anteriores BHR 30 siendo desplazados a la izquierda a medida que crece el recuento).

En otra realización, una indicación fija y una de múltiples bits del número de iteraciones del bucle se almacena en el BHR 30. Por ejemplo, pueden asignarse dos bits para grabar una indicación del número de iteraciones del bucle, con los siguientes significados:

11 - Número de iteraciones del bucle muy grande

10 - Número de iteraciones del bucle grande

01 - Número de iteraciones del bucle moderado

00 - Número de iteraciones del bucle pequeño

En este ejemplo, se utilizan tres valores de umbral - entre pequeñas y moderadas (00 -> 01), entre moderada y grande (01 -> 10), y entre grande y muy grandes (10 -> 11) - para asignar el recuento real del número de iteraciones a una representación de dos bits, como indica una Lógica de Umbralización 39. En una realización, los valores de umbral se pueden fijar para una implementación de procesador dada, tal como para ciertas aplicaciones empotradas en las que los tamaños comunes de bucle son generalmente conocidos. En otra realización, la asignación puede ser escalable, con los valores de umbral leídos de GPR 20, una tabla en la memoria 44, o similar. En las aplicaciones más comunes, una indicación fija de múltiples bits del número de iteraciones del bucle puede proporcionar una indicación suficiente del recuento de iteraciones del bucle para permitir correlarla coherentemente con el comportamiento de la evaluación del salto, mientras que al mismo tiempo limita el número de evaluaciones de salto anteriores desplazadas del BHR 30 por saltos de final de bucle.

Por supuesto, los recuentos reales de iteraciones del bucle se pueden asignar a cualquier número fijo de bits, con el número correspondiente de valores de umbral, por ejemplo, tres bits y siete umbrales, cuatro bits y quince umbrales, etc., para lograr el equilibrio deseado o requerido entre la precisión del recuento de iteraciones del bucle y los desplazamientos de la evaluación de saltos en el BHR 30.

Bucles anidados

Un bucle puede contener uno o más bucles anidados, o interiores. Por consiguiente, en una realización, se proporciona una pluralidad de contadores LC (LC_0, LC_1, \dots, LC_M) 37, que pueden corresponder al número de registros LBPC 38. Un LC 37 se incrementa en una evaluación "se saltará" y una correspondencia con la $LBPC_M$ 38 relevante, para mantener los recuentos de iteraciones de los bucles anidados.

Al almacenar una indicación de múltiples bits del número de iteraciones de un bucle interior cada vez que el bucle interior sale, el BHR 30 puede llegar a saturarse parcial o completamente una vez que el bucle exterior itera relativamente pocas veces. Por consiguiente, en una realización, una vez que se determina que un bucle que es un bucle interior, recuentos de iteraciones posteriores después de una ejecución inicial del bucle interior pueden no actualizar el BHR 30. En esta realización, en el momento en el que el bucle externo sale, el BHR 30 incluirá una indicación del recuento de iteraciones del bucle interno sólo para la primera iteración del bucle externo, y una indicación del recuento de iteraciones del bucle externo.

En otra realización, el recuento de iteraciones del bucle interior puede ser desechado sobreescribiendo su valor en el BHR 30 con una indicación del recuento de iteraciones del bucle exterior. Esto conserva el mayor número de evaluaciones de saltos anteriores en el BHR 30, al tiempo que tiene en cuenta la correlación de las evaluaciones de saltos posteriores con el recuento de iteraciones del bucle externo.

En otra realización, el recuento de iteraciones del bucle exterior puede ser desechado suprimiendo una actualización del BHR 30 cuando el bucle exterior sale. Esto preserva de nuevo el mayor número de evaluaciones anteriores de salto en el BHR 30, permitiendo al mismo tiempo correlar las evaluaciones de salto posteriores con el recuento de

iteraciones del bucle interior, que puede ser predecir mejor la evaluación de saltos posteriores en algunas aplicaciones.

5 Una instrucción de salto condicional tiene propiedades comunes, como por ejemplo la dirección de la instrucción de salto o PC, el tipo de instrucción, y la presencia, o no, de bits indicadores en el código de operación. Como se usa aquí, las propiedades de la operación de salto, y/o las propiedades del programa relativas a los saltos, se consideran propiedades de la instrucción de salto. Por ejemplo, si el PC de instrucción de salto coincide con el contenido de uno o más registros LBPC 38, y si la dirección de destino es de salto hacia adelante o hacia atrás en relación al PC de instrucción de salto, son propiedades de la instrucción de salto.

10 Aunque la presente descripción se ha descrito en este documento con respecto a características, aspectos y realizaciones particulares de la misma, será evidente que numerosas variaciones, modificaciones y otras realizaciones son posibles dentro del amplio alcance de la presente descripción, y en consecuencia, todas las modificaciones, variaciones y realizaciones deben ser consideradas como dentro del alcance de la descripción. Las realizaciones presentes deben por tanto interpretarse en todos los aspectos como cambios ilustrativos y no restrictivos y todos los cambios comprendidos en el rango de significado y equivalencia de las reivindicaciones
15 adjuntas están destinados a ser abarcados por este documento.

Otros ejemplos ilustrativos que no se reivindican:

1. Un procedimiento de predicción de salto, que comprende:

identificar una característica de una instrucción de salto asociada con un bucle, y

20 en respuesta a la propiedad, almacenar un valor de múltiples bits en un Registro de Historia de Saltos (BHR), el valor indicativo del número de iteraciones de un bucle asociado con la instrucción de salto.

2. El procedimiento según la realización 1 en el que la propiedad es la dirección del salto.

3. El procedimiento según la realización 2 en el que el salto es hacia atrás.

4. El procedimiento según la realización 1 en el que la instrucción de salto es una instrucción de salto de final de bucle.

25 5. El procedimiento según la realización 4, en el que el PC de la instrucción de salto coincide con el contenido de un registro de PC de Último Salto (LBPC) que almacena el PC de la última instrucción de salto para actualizar el BHR.

6. El procedimiento según la realización 4 en el que el PC de la instrucción de salto coincide con el contenido de cualquiera de una pluralidad de registros LBPC que almacenan PCs de la última pluralidad de instrucciones de salto para actualizar el BHR.

30 7. El procedimiento según la realización 4 en el que la instrucción de salto es una instrucción de salto único generado por un compilador para finalizar bucles.

8. El procedimiento según la realización 4 en el que el código de operación de la instrucción de salto incluye uno o más bits que indican que es una instrucción de salto de final de bucle.

35 9. El procedimiento de la realización 1 en el que almacenar de un valor múltiples bits en el BHR comprende almacenar un número predeterminado de bits en el BHR.

10. El procedimiento según la realización 9 que comprende además determinar el valor del número predeterminado de bits de acuerdo con una asignación fija del número de iteraciones del bucle con el valor de múltiples bits.

11. El procedimiento según la realización 9 que comprende además determinar el valor del número predeterminado de bits de acuerdo con una asignación escalable del número de iteraciones de bucle con el valor de múltiples bits.

40 12. El procedimiento según la realización 11 que comprende además leer una pluralidad de umbrales para determinar la asignación escalable del número de iteraciones de bucle con el valor de múltiples bits.

13. El procedimiento de la realización 1 en el que el almacenamiento de un valor de múltiples bits en el BHR comprende almacenar un número variable de bits en el BHR, variando el número de bits en respuesta al número de iteraciones del bucle.

14. El procedimiento según la realización 13 que comprende además contar las iteraciones del bucle en la pluralidad de bits menos significativa del BHR.
15. El procedimiento según la realización 13 que comprende además contar las iteraciones del bucle en un contador de bucle, y transferir el valor del contador del bucle al BHR cuando el bucle termina.
- 5 16. El procedimiento de la realización 1 en el que identificar una propiedad de una instrucción de salto asociada con un bucle comprende detectar una primera instrucción de salto de final de bucle asociada con un primer bucle y una segunda instrucción de salto de final de bucle asociada con un segundo bucle, el primer bucle anidado dentro el segundo bucle.
- 10 17. El procedimiento según la realización 16 en el que almacenar un valor de múltiples bits en el BHR comprende almacenar un valor de múltiples bits indicativo del número de iteraciones del primer bucle en el BHR, y que comprende además almacenar un valor de múltiples bits indicativo del número de iteraciones del segundo bucle en el BHR.
18. El procedimiento según la realización 16 en el que almacenar un valor múltiples bits en el BHR comprende almacenar sólo un valor de múltiples bits indicativo del número de iteraciones del primer bucle en el BHR.
- 15 19. El procedimiento según la realización 16 en el que el almacenamiento de un valor de múltiples bits en el BHR comprende almacenar sólo un valor de múltiples bits indicativo del número de iteraciones del segundo bucle en el BHR.
20. Un procesador, que comprende:
- un predictor de saltos operable para predecir la evaluación de instrucciones de salto condicional;
- 20 una línea de ejecución de instrucciones operable para cargar y ejecutar de forma especulativa las instrucciones en base a una predicción del predictor de saltos;
- una Registro de Historia de Saltos (BHR) operable para almacenar la evaluación de las instrucciones de salto condicional;
- un contador de bucle (LC) operable para contar el número de iteraciones de un bucle de código, y
- 25 un circuito de control operable para almacenar en el BHR un valor da múltiples bits indicativo del número de iteraciones de un bucle asociado con una instrucción de salto condicional.
21. El procesador según la realización 20 que comprende además un registro de PC de Último Salto (LBPC) operable para almacenar el PC de una instrucción de salto condicional, y en el que el circuito de control determina qze una instrucción de salto condicional está asociada con un bucle si el PC de la instrucción de salto coincide con el contenido del registro LBPC.
- 30
22. El procesador según la realización 21 que comprende además dos o más registros LBPC y dos o más LCs correspondientes, un primer LBPC operable para almacenar el PC de una instrucción de salto condicional asociada con un primer bucle y un primer LC operable para mantener un recuento de iteraciones del primer bucle, y un segundo LBPC operable para almacenar el PC de una instrucción de salto condicional asociada con un segundo bucle y un segundo LC operable para mantener un recuento de iteraciones del segundo bucle, en el que el primer bucle está anidado dentro del segundo bucle.
- 35
23. El procesador según la realización 22, en el que el circuito de control es operable para almacenar en el BHR un valor de múltiples bits indicativo del número de iteraciones de una ejecución del primer bucle y un valor de múltiples bits indicativo del número de iteraciones del segundo bucle.
- 40 24. El procesador según la realización 22, en el que el circuito de control es operable para almacenar en el BHR un valor de múltiples bits indicativo del número de iteraciones de una ejecución del primer bucle y para no almacenar en el BHR una indicación del número de iteraciones del segundo bucle.
25. El procesador según la realización 22, en el que el circuito de control es operable para almacenar en el BHR un valor de múltiples bits indicativo del número de iteraciones del segundo bucle y para no almacenar en el BHR una indicación del número de iteraciones de cualquier ejecución del primer bucle.
- 45

26. El procesador según la realización 20 en el que el BHR es operable para incrementar una pluralidad de bits en respuesta a cada evaluación "se saltará" de la instrucción de salto condicional asociada con el bucle a fin de mantener un recuento de iteraciones del bucle directamente en el BHR.

5 27. El procesador según la realización 20 que comprende además lógica de umbral operable para asignar un recuento de iteraciones de bucle a un valor fijo, de múltiples bits en respuesta a una pluralidad de valores de umbral.

REIVINDICACIONES

1. Un procedimiento de predicción de saltos, que comprende:
 - determinar que una instrucción de salto condicional es una instrucción de final de bucle;
 - 5 contar un número total de iteraciones de un bucle finalizado por la instrucción de final de bucle;
 - almacenar un valor de múltiples bits en un Registro de Historia de Saltos, BHR, tras la finalización del bucle, el valor indicativo del número total de iteraciones del bucle, y
 - indexar una Tabla de Predicción de Saltos, BPT, con el BHR, después de la finalización del bucle, para obtener una predicción de saltos.
- 10 2. El procedimiento según la reivindicación 1, en el que determinar que una instrucción de salto condicional es una instrucción de salto de final de bucle comprende detectar que el salto condicional es hacia atrás.
3. El procedimiento según la reivindicación 1, en el que determinar que una instrucción de salto condicional es una instrucción de salto de final de bucle comprende detectar que un contador de programa, PC de la instrucción de salto coincide con el contenido de un PC de Último Salto, LBPC, registro que almacena PCs de la última pluralidad de instrucciones de salto para actualizar el BHR.
- 15 4. El procedimiento según la reivindicación 3 en el que la instrucción de salto es un tipo particular de instrucción de salto utilizada únicamente para saltos de final de bucle y generada por un compilador para finalizar bucles.
5. El procedimiento según la reivindicación 1, en el que almacenar un valor de múltiples bits en el BHR comprende almacenar un número predeterminado de bits en el BHR.
- 20 6. El procedimiento según la reivindicación 5 que comprende además determinar el valor del número predeterminado de bits según una asignación entre el número de iteraciones de bucle y el valor de múltiples bits.
7. El procedimiento según la reivindicación 1, en el que almacenar un valor de múltiples bits en el BHR comprende almacenar un número variable de bits en el BHR, el número de bits variando en respuesta al número total de iteraciones del bucle.
- 25 8. El procedimiento según la reivindicación 7 que comprende además almacenar iteraciones de bucle en la pluralidad de bits menos significativa del BHR.
9. El procedimiento según la reivindicación 7 que comprende además contar iteraciones de bucle en un contador de bucle, y transferir el valor del contador de bucle al BHR cuando el bucle finaliza.
- 30 10. El procedimiento según la reivindicación 1, en el que la etapa de determinar comprende detectar una primera instrucción de salto de final de bucle asociada con un primer bucle y una segunda instrucción de salto de final de bucle asociada con un segundo bucle, el primer bucle anidado dentro del segundo bucle.
11. Un procesador, que comprende:
 - un predictor de saltos operable para predecir la evaluación de instrucciones de salto condicional;
 - 35 una línea de ejecución de instrucciones operable para cargar y ejecutar de forma especulativa instrucciones en base a una predicción del predictor de saltos;
 - un Registro de Historia de Saltos, BHR, en el predictor de saltos operable para almacenar la evaluación de las instrucciones de salto condicional;
 - un Contador de Bucle, LC, en el predictor de saltos operable para contar un número total de iteraciones de un bucle de código;
 - 40 un circuito de control operable para almacenar en el BHR un valor de múltiples bits indicativo del número total de iteraciones de un bucle asociado con una instrucción de salto condicional tras la terminación del bucle, y

una tabla de predicción de saltos, BPT, en el predictor de saltos, indexada por el BHR, tras la terminación del bucle, operable para conseguir una predicción de saltos.

- 5 12. El procesador según la reivindicación 11 que comprende además un registro PC de Último Salto, LBPC, operable para almacenar un contador de programa, PC, de una instrucción de salto condicional, y en el que el circuito de control determina que una instrucción de salto condicional está asociada con un bucle si el PC de la instrucción de salto coincide con el contenido del registro LBPC.
- 10 13. El procesador según la reivindicación 12 que comprende además dos o más registros LBPC y dos o más LC correspondientes, un primer LBPC operable para almacenar el PC de una instrucción de salto condicional asociada con un primer bucle y un primer LC operable para mantener una cuenta de iteraciones del primer bucle, y un segundo LBPC operable para almacenar el PC de una instrucción de salto condicional asociada con un segundo bucle y un segundo LC operable para mantener una cuenta de iteraciones del segundo bucle, en el que el primer bucle está anidado dentro del segundo bucle.
- 15 14. El procesador según la reivindicación 11 en el que el BHR es operable para incrementar una pluralidad de bits en respuesta a cada evaluación tomada de la instrucción de salto condicional asociada con el bucle de forma que mantiene una cuenta de iteraciones del bucle directamente en el BHR.
15. El procesador según la reivindicación 11 comprende además lógica de umbralización operable para asignar una cuenta de iteraciones de bucle a un valor de fijo, de múltiples bits en respuesta a la pluralidad de valores de umbral.

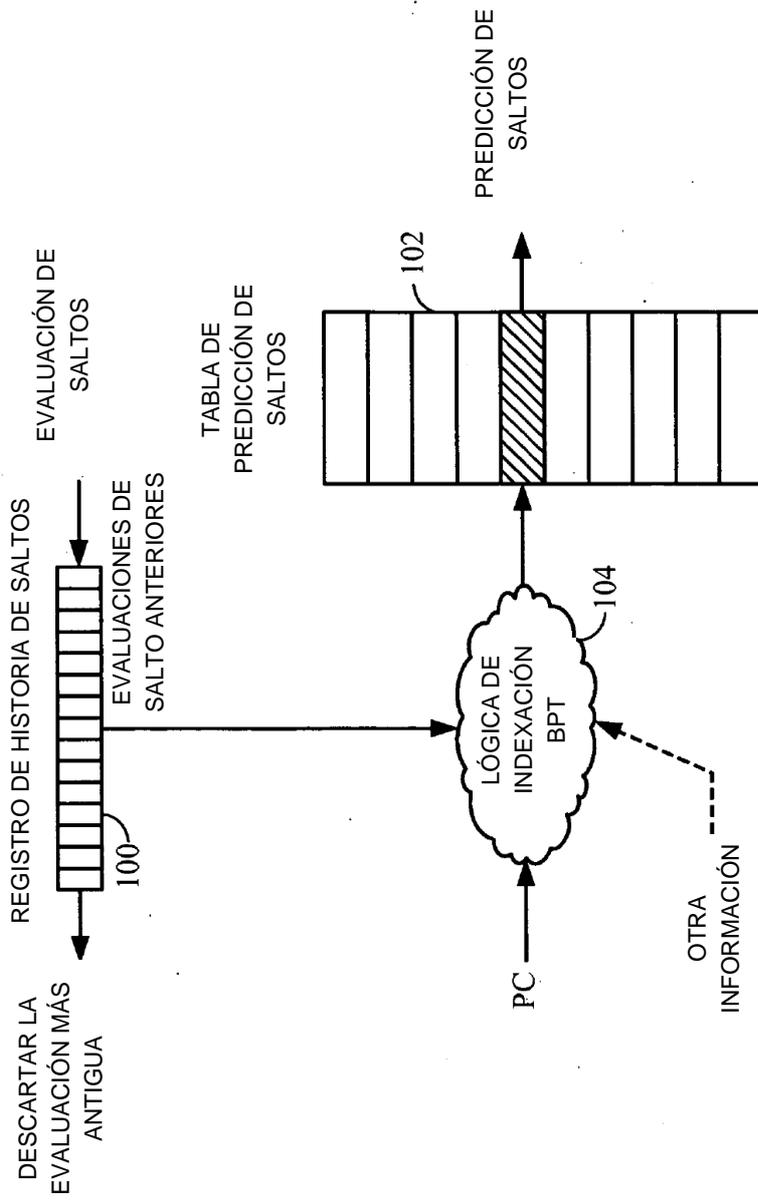


FIG. 1

(ESTADO DE LA TÉCNICA)

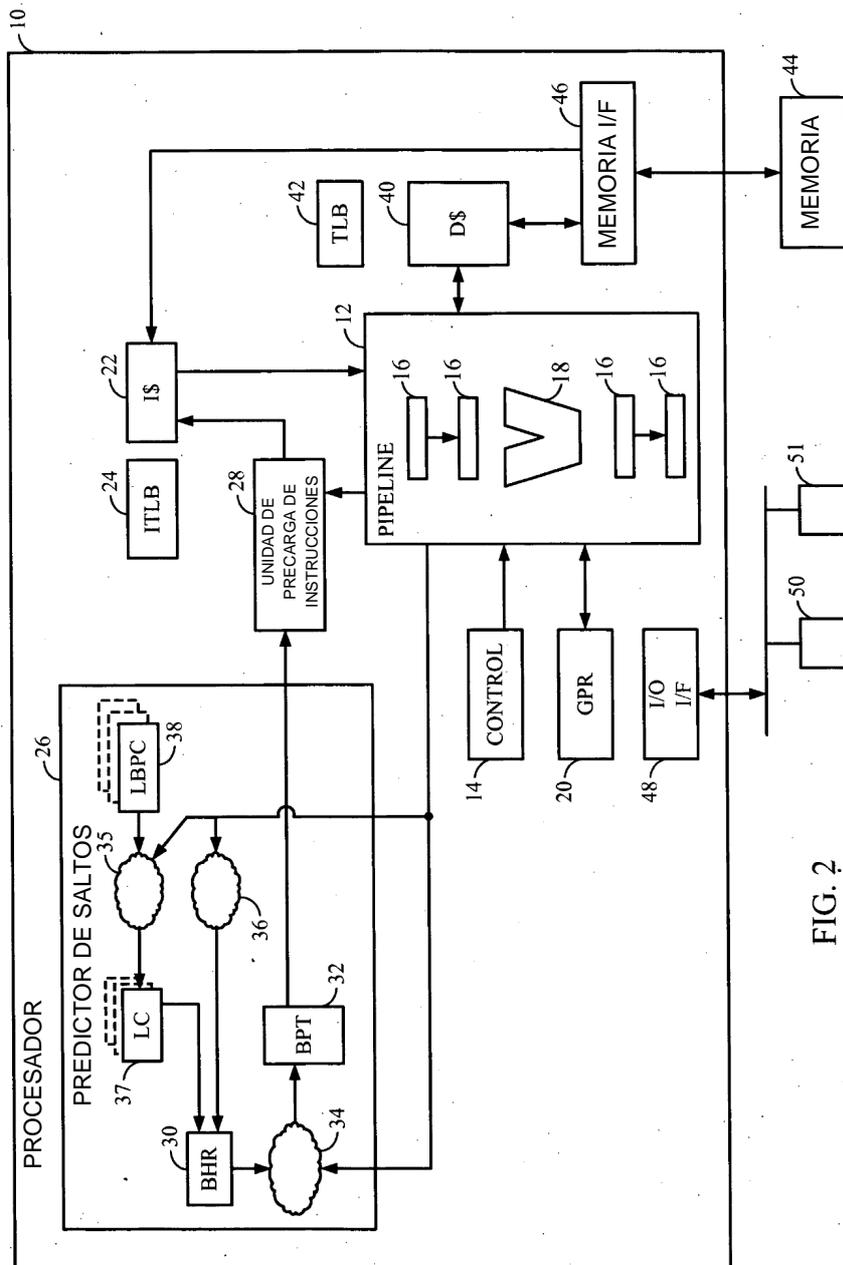


FIG. 2

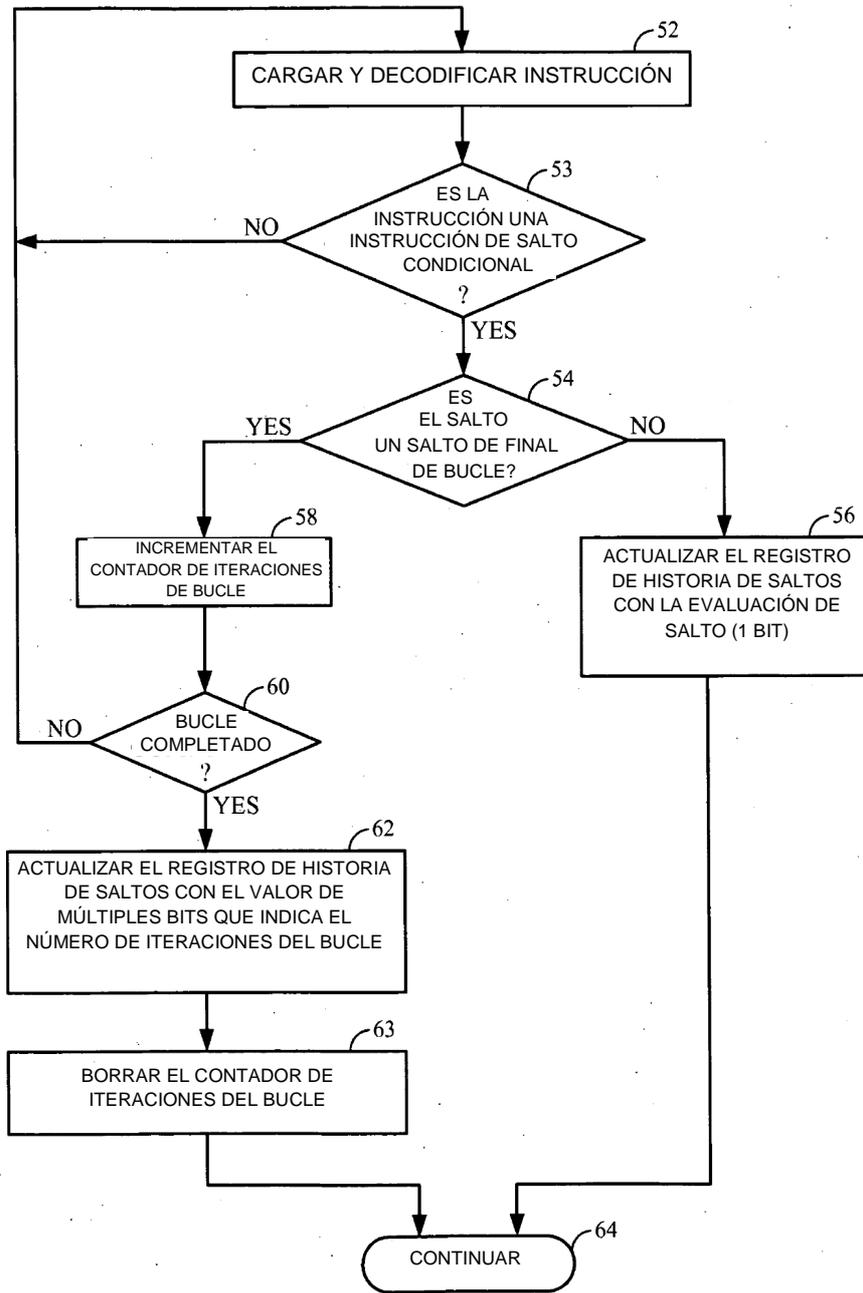


FIG. 3

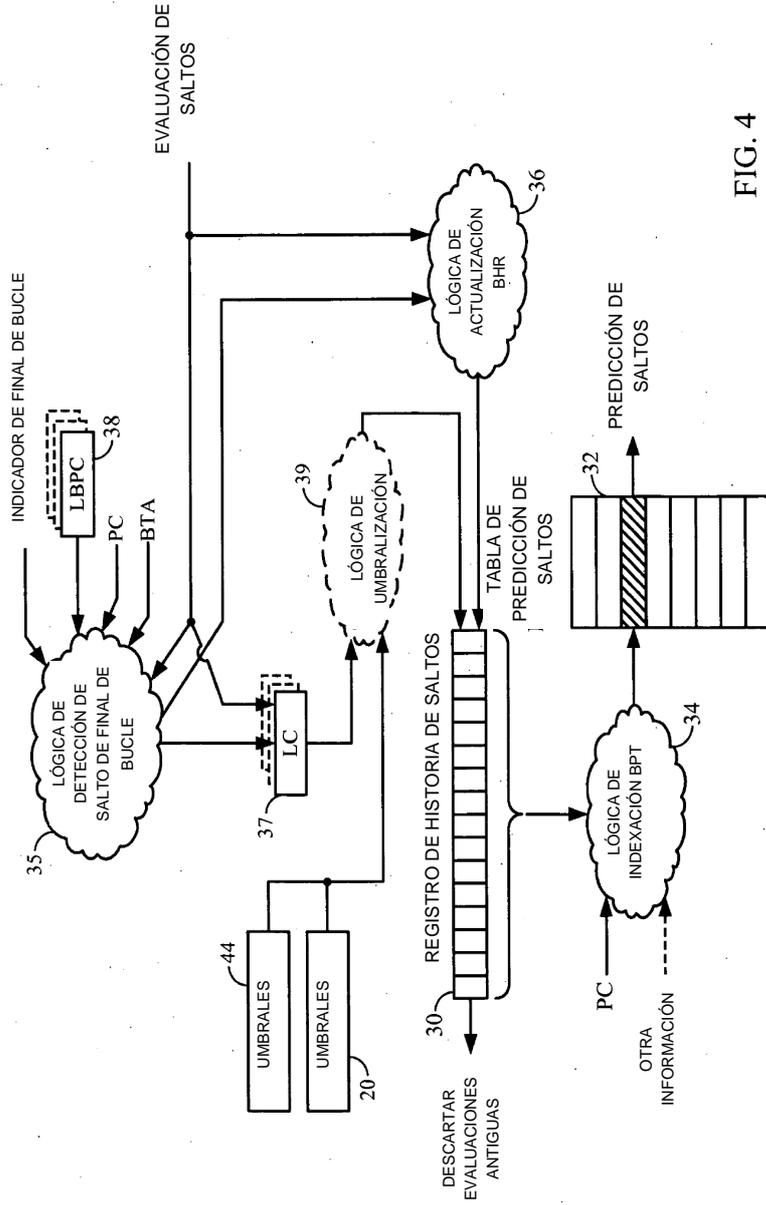


FIG. 4