(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 655 852**

(51) Int. CI.:

G06F 9/38 (2006.01)
G06F 9/345 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.01.2014 PCT/US2014/012152**

(87) Fecha y número de publicación internacional: **24.07.2014 WO14113741**

(96) Fecha de presentación y número de la solicitud europea: **18.01.2014 E 14704714 (6)**

(97) Fecha y número de publicación de la concesión europea: **25.10.2017 EP 2946286**

(54) Título: **Procedimientos y aparatos para cancelar solicitudes de captura previa de datos para un bucle**

(30) Prioridad:

21.01.2013 US 201313746000

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.02.2018

(73) Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

(72) Inventor/es:

GILBERT, MATTHEW, M.

(74) Agente/Representante:

FORTEA LAGUNA, Juan José

DESCRIPCIÓN

Procedimientos y aparatos para cancelar solicitudes de captura previa de datos para un bucle

Campo de la divulgación

[0001] La presente descripción se refiere en general a los aspectos de los sistemas de procesamiento y, en particular, a procedimientos y aparatos para reducir la contaminación de memoria caché causada por la captura previa de datos.

Antecedentes

[0002] Muchos productos portátiles, tales como teléfonos móviles, ordenadores portátiles, asistentes de datos personales (PDA) y similares, utilizan un sistema de procesamiento que ejecuta programas, tales como programas de comunicación y multimedios. Un sistema de procesamiento para tales productos puede incluir múltiples procesadores, sistemas de memoria complejos que incluyen niveles múltiples de memorias caché para almacenar instrucciones y datos, controladores, dispositivos periféricos tales como interfaces de comunicación y bloques lógicos de función fija configurados, por ejemplo, en un único chip. Al mismo tiempo, los productos portátiles tienen una fuente de energía limitada en forma de baterías a las que a menudo se requiere que presten soporte a operaciones de alto rendimiento por parte del sistema de procesamiento. Para aumentar la duración de la batería, es deseable realizar estas operaciones de la manera más eficaz posible. Muchos ordenadores personales también se están desarrollando con diseños eficaces para funcionar con un consumo de energía total reducido.

[0003] Con el fin de proporcionar un alto rendimiento en la ejecución de los programas, puede utilizarse la captura previa de datos que se basa en el concepto de localidad espacial de referencias a memoria y que se usa generalmente para mejorar el rendimiento del procesador. Al capturar previamente múltiples elementos de datos desde una memoria caché en direcciones que están cerca de un elemento de datos capturados o que están relacionados por un delta de dirección de paso o un puntero indirecto, y que probablemente se utilizarán en futuros accesos, las tasas de error de caché pueden reducirse. Los diseños de memoria caché generalmente implementan una forma de captura previa capturando una línea de datos de memoria caché para una captura de elemento individual de datos. Los capturadores previos de datos de hardware pueden ampliar esto capturando de forma especulativa una o más líneas adicionales de datos de memoria caché, donde las direcciones de captura previa puede formarse basándose en información secuencial, de paso o de puntero. Tal operación de captura previa de hardware para cargas de trabajo intensivas en memoria, tales como el procesamiento de una gran formación de datos, puede reducir significativamente la latencia de la memoria. Sin embargo, la captura previa de datos no está exenta de inconvenientes. Por ejemplo, en un bucle de software utilizado para procesar una formación de datos, un circuito de captura previa de datos captura previamente los datos que se utilizarán en futuras iteraciones del bucle, incluida la última iteración del bucle. Sin embargo, los datos previamente capturados para la última iteración del bucle no se utilizarán y la contaminación de la memoria caché se produce al almacenar estos datos que no se usarán en la memoria caché. El problema de la contaminación de memoria caché se agrava cuando se desenrollan bucles.

[0004] La patente estadounidense Nº 6.775.765 se refiere a un sistema de procesamiento de datos que tiene plegado de instrucciones y un procedimiento del mismo. La patente estadounidense Nº US 6,260,116 se refiere a un sistema y procedimiento para la captura previa de datos. El artículo "Una arquitectura eficaz para la precarga de datos basada en bucles", de Chen et al., Propone un almacén temporal de precarga como soporte arquitectónico para la precarga.

RESUMEN

[0005] Entre sus diversos aspectos, la presente divulgación reconoce que la provisión de procedimientos y aparatos más eficaces para la captura previa puede mejorar el rendimiento y reducir los requisitos de potencia en un sistema procesador. Para tales fines, una realización de la invención aborda un procedimiento para cancelar solicitudes de captura previa. Se identifica una situación de salida de bucle basándose en una evaluación de información de flujo de programa. Las solicitudes pendientes de captura previa de memoria caché se cancelan en respuesta a la situación de salida de bucle identificada.

[0006] Otra realización aborda un procedimiento para cancelar solicitudes de captura previa. Los datos se capturan previamente de manera especulativa de acuerdo a una función llamada. Las solicitudes pendientes de captura previa de datos se cancelan en respuesta a una salida de función desde la función llamada.

[0007] Otra forma de realización aborda un aparato para la cancelación de las solicitudes de captura previa. Un monitor de direcciones de datos de bucle está configurado para determinar un paso de acceso a datos basado en la ejecución repetida de una instrucción de acceso a memoria en un bucle de programa. La lógica de captura previa de datos está configurada para emitir especulativamente solicitudes de captura previa de acuerdo al paso de acceso a los datos. Un circuito de detención de captura previa está configurado para cancelar solicitudes pendientes de

captura previa en respuesta a una salida de bucle identificada.

[0008] Otra forma de realización aborda un medio no transitorio legible por ordenador, codificado con datos y código de programa legibles por ordenador. Se identifica una situación de salida de bucle basándose en una evaluación de información de flujo de programa. Las solicitudes pendientes de captura previa de memoria caché se cancelan en respuesta a la situación de salida de bucle identificada.

[0009] Una realización adicional aborda un aparato para la cancelación de las solicitudes de captura previa. El medio se utiliza para determinar un paso de acceso a datos basado en la ejecución repetida de una instrucción de acceso a memoria en un bucle de programa. El medio se utiliza para emitir especulativamente solicitudes de captura previa de acuerdo al paso de acceso a los datos. El medio también se utiliza para cancelar solicitudes pendientes de captura previa en respuesta a una salida de bucle identificada.

[0010] Debe entenderse que otros modos de realización de la presente divulgación resultarán inmediatamente evidentes para 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 del alcance de la presente invención. Por consiguiente, debe considerarse que los dibujos y la descripción detallada tienen naturaleza ilustrativa y no restrictiva.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011] Diversos aspectos de la presente invención se ilustran a modo de ejemplo, y no de manera limitativa, en los dibujos adjuntos, en los que:

la FIG. 1 ilustra un sistema procesador ejemplar en el que se puede emplear ventajosamente una realización de la invención;

la FIG. 2A ilustra un proceso para cancelar solicitudes pendientes de captura previa de datos no de demanda al detectar una bifurcación de finalización de bucle; y

la FIG. 2B ilustra un proceso para cancelar solicitudes pendientes de captura previa de datos no de demanda al detectar un retorno de función; y

la FIG. 3 ilustra una realización particular de un dispositivo portátil que tiene un complejo de procesadores que está configurado para cancelar solicitudes pendientes de captura previa de datos, seleccionadas para reducir la contaminación de la memoria caché.

DESCRIPCIÓN DETALLADA

[0012] La descripción detallada expuesta a continuación en relación con los dibujos adjuntos está concebida como una descripción de varios modos de realización a modo de ejemplo de la presente invención y no está concebida para representar los únicos modos de realización en los cuales la presente invención pueda practicarse. La descripción detallada incluye detalles específicos con el objeto de proporcionar una comprensión exhaustiva de la presente invención. Sin embargo, a los expertos en la técnica les resultará evidente que la presente invención puede llevarse a la práctica sin estos detalles específicos. En algunos casos, estructuras y dispositivos ampliamente conocidos se muestran en forma de diagrama de bloques para no oscurecer los conceptos de la presente invención.

[0013] La FIG. 1 ilustra un sistema procesador ejemplar 100 en el que se emplea ventajosamente una realización de la invención. El sistema procesador 100 incluye un procesador 110, un sistema de memoria caché 112, una memoria de sistema 114 y un sistema de entrada y salida (I/O) 116. El sistema de memoria caché 112, por ejemplo, comprende una memoria caché de instrucciones de nivel 1 (Icache) 124, un controlador de memoria 126 y una memoria caché de datos de nivel 1 (Dcache) 128. El sistema de memoria caché 112 también puede incluir una memoria caché unificada de nivel 2 (no mostrada) u otros componentes de memoria caché según se desee para un entorno de implementación particular. La memoria del sistema 114 proporciona acceso para las instrucciones y datos que no se encuentran en Icache 124 o Dcache 128. Se observa que el sistema de memoria caché 112 se puede integrar con el procesador 110 y también puede incluir múltiples niveles de memoria caché en una organización jerárquica. El sistema de I/O 116 comprende una pluralidad de dispositivos de I/O, tales como dispositivos de I/O 140 y 142, que interactúan con el procesador 110.

[0014] Las realizaciones de la invención pueden emplearse adecuadamente en un procesador que tiene instrucciones de bifurcación condicional. El procesador 110 comprende, por ejemplo, un conducto de instrucciones 120, lógica de captura previa de datos 121, lógica de predicción 122 y un circuito lógico de pila 123. El conducto de instrucciones 120 está compuesto de una serie de etapas, tales como una etapa de captura y captura previa 130, una etapa de decodificación 131, una etapa de emisión de instrucciones 132, una etapa de captura de operandos 133, una etapa de ejecución 134, tal como para la ejecución de instrucciones de carga (Ld) y de almacenamiento

(St), y la etapa de finalización 135. Los expertos en la materia reconocerán que cada etapa 130 a 135 en el conducto de instrucciones 120 puede comprender una serie de etapas adicionales de conducto en función de la frecuencia operativa del procesador y la complejidad de las operaciones requeridas en cada etapa. Por ejemplo, la etapa de ejecución 134 puede incluir una o más etapas de conducto correspondientes a uno o más circuitos de etapas de ejecución de instrucciones, tales como un sumador, un multiplicador, operaciones lógicas, operaciones de carga y almacenamiento, operaciones de desplazamiento y rotación, y otros circuitos funcionales de mayor o menor complejidad. Por ejemplo, cuando se ejecuta una instrucción de carga, solicita datos de Dcache 128 y, si los datos solicitados no están presentes en Dcache, se emite una solicitud de captura al siguiente nivel de la memoria caché o del sistema. Dicha solicitud de captura se considera una solicitud de demanda ya que es una respuesta directa a la ejecución de una instrucción, en este caso, una instrucción de carga.

[0015] Una solicitud de captura previa es una solicitud que se hace en respuesta a la información del flujo del programa, tal como la detección de un bucle de programa que tiene una o más instrucciones de carga en el bucle con direcciones de carga basadas en un paso, por ejemplo. La lógica de captura previa de datos 121 utiliza dicha información de flujo de programa, que puede basarse en un número de iteraciones del bucle detectado, para identificar con mayor precisión un patrón de uso de demanda de las direcciones de operandos de las instrucciones de carga antes de emitir una solicitud de captura previa. Las solicitudes de relleno se insertan cuando se detecta un patrón. El procesador 110 puede operar para diferenciar una solicitud de demanda de una solicitud de captura previa mediante el uso de un indicador adicional asociado a la solicitud que se rastrea en el conducto del procesador. Este indicador también podría propagarse con la solicitud a la memoria caché, donde cada relleno pendiente de línea de memoria caché podría identificarse como una captura previa o un relleno de demanda. Cada una de las etapas del conducto puede tener implementaciones variadas sin apartarse de los procedimientos y aparatos de cancelación de solicitud de captura previa, descritos en este documento.

[0016] Con el fin de minimizar los retrasos que podrían ocurrir si los datos requeridos por un programa no estuvieran en la memoria Dcache de nivel 1 asociada 128, la etapa de captura y captura previa 130 registra la información de flujo de programa asociada a una o más instrucciones de acceso a memoria que se ejecutan en un bucle de programa detectado. La información de programa puede incluir una indicación desde la etapa de decodificación 131 en cuanto a que se ha recibido una instrucción de carga y la información de dirección de operandos para la instrucción de carga puede estar disponible en una etapa del conducto antes de la ejecución, tal como la etapa de captura de operandos 133 o en la etapa de ejecución 134. La lógica de captura previa de datos 121 supervisa las direcciones de carga a medida que quedan disponibles para detectar un patrón. Después de que el patrón se determina con un nivel de confianza aceptable, tal como al monitorizar instrucciones de carga mediante tres o más iteraciones de un bucle, se emite una solicitud de captura previa de datos esperados antes de que la instrucción de carga se encuentre nuevamente en el bucle. Esta solicitud de captura previa especulativa garantiza que los datos requeridos estén disponibles en la Dcache de nivel 1 cuando los necesite la etapa de ejecución 134. Entonces es más probable que la etapa de ejecución de carga y almacenamiento 134 acceda a los datos requeridos directamente desde la Dcache de nivel 1 sin tener que esperar para acceder a los datos de niveles superiores en la jerarquía de memoria.

[0017] La lógica de captura previa de datos 121 también puede incluir un monitor de direcciones de datos de bucle de memoria caché de datos para determinar un paso de acceso a datos. La lógica de captura previa de datos 121 luego emite especulativamente solicitudes de captura previa con direcciones de operandos establecidas de acuerdo al paso de acceso a los datos. Por ejemplo, la lógica de captura previa de datos 121 puede incluir un circuito de paso 119 que está configurado para supervisar ejecuciones repetidas de una instrucción de carga para determinar una diferencia entre la dirección de operandos de cada ejecución de la instrucción de carga que representa un valor de paso. El circuito de paso 119 también puede incluir una función de adición que está configurada para agregar el valor del paso determinado a la dirección de operando de la instrucción de carga ejecutada más recientemente, para generar la siguiente dirección de operando. A diferencia del valor del paso como una dirección predicha, una instrucción de bifurcación condicional capturada usa una lógica de predicción de bifurcación, tal como la contenida en el circuito lógico de predicción 122, para predecir si se tomará la bifurcación condicional y la dirección de bifurcación. Una instrucción capturada no de bifurcación avanza a la etapa de decodificación 131 para ser decodificada, emitida para su ejecución en la etapa de emisión de instrucciones 132, ejecutada en la etapa de ejecución 134 y retirada en la etapa de finalización 135.

[0018] El circuito de lógica de predicción 122 comprende un circuito de lógica de detección 146 para supervisar sucesos, un filtro 150 y una tabla de historia condicional 152. En una realización, se supone que una mayoría de las instrucciones de bifurcación condicional generalmente tienen sus condiciones resueltas en el mismo valor para la mayoría de las iteraciones de un bucle de software.

[0019] El circuito de lógica de detección 146, en una realización, actúa como un detector de bucle de software que funciona basándose en las características dinámicas de instrucciones de bifurcación condicional utilizadas en los bucles de software, tal como se describe con respecto a la FIG. 2A. El circuito lógico de detección 146 también puede detectar salidas de funciones de software llamadas, tal como se describe con respecto a la FIG. 2B.

[0020] En bucles de software con una sola entrada y una sola salida, una bifurcación de final de bucle es

generalmente una instrucción de bifurcación condicional que bifurca de vuelta al inicio del bucle de software para todas las iteraciones del bucle, a excepción de la última iteración, que sale del bucle de software. El circuito lógico de detección 146 puede tener múltiples realizaciones para la detección de bucles de software, tal como se describe con más detalle a continuación y en la solicitud de patente estadounidense 11 / 066.508, cedida al cesionario de la presente solicitud, titulada "Supresión de actualización de un registro de historia de bifurcación por bifurcaciones de fin de bucle".

[0021] Según una realización, el circuito de lógica de detección 146 identifica las instrucciones de bifurcación condicional con una dirección de destino de bifurcación inferior a la dirección de instrucción de bifurcación condicional y, por lo tanto, consideradas como una bifurcación inversa, y se supone que marcan el final de un bucle de software. Dado que no todas las bifurcaciones inversas son bifurcaciones de fin de bucle, existe un cierto nivel de imprecisión que puede necesitar ser tenido en cuenta por mecanismos de supervisión adicionales, por ejemplo.

[0022] Además, como se describe con respecto a la FIG. 2B, se puede detectar una instrucción de retorno de función (comúnmente denominada RET). De acuerdo a una realización, la detección de un retorno de función está adaptada para desencadenar cancelaciones de captura previa de solicitudes cualesquiera de captura previa no de demanda. La cancelación de una solicitud de captura previa también se realiza en respuesta a información de flujo de programa, tal como la detección de una salida de bucle.

[0023] En otra realización, una bifurcación de fin de bucle puede ser detectada en bucles simples mediante el reconocimiento de la ejecución repetida de la misma instrucción de bifurcación. Al almacenar el valor del contador de programa para la última instrucción de bifurcación inversa en un registro de propósito especial, y comparar este valor almacenado con la dirección de instrucción de la próxima instrucción de bifurcación inversa, se puede reconocer una bifurcación de fin de bucle cuando las dos direcciones de instrucción coinciden. Como el código puede incluir instrucciones de bifurcación condicional dentro de un bucle de software, la determinación de la instrucción de bifurcación de fin de bucle puede volverse más complicada. En tal situación, múltiples registros de propósito especial pueden ser instanciados en hardware para almacenar las direcciones de instrucciones de cada instrucción de bifurcación condicional. Al comparar con todos los valores almacenados, se puede determinar una coincidencia para la bifurcación que termina el bucle. Habitualmente, las bifurcaciones de bucle son bifurcaciones directas inversas condicionales con un desplazamiento fijo a partir del contador de programa (PC). Este tipo de bifurcaciones no necesitaría comparaciones de direcciones para la detección de una salida de bucle. En cambio, una vez que se detecta un bucle de programa basándose en una bifurcación directa inversa condicional, la salida del bucle se determina a partir de la resolución del predicado de la bifurcación. Por ejemplo, si el predicado se resuelve en una condición verdadera para regresar al bucle, se indicaría la salida del bucle cuando el predicado se resuelva en una condición falsa. Para que haya capturas previas pendientes, un ciclo de programa ya se habría ejecutado unas pocas veces para activar el hardware de captura previa. La lógica de captura previa de datos 121 requiere unas pocas cargas de demanda de calentamiento para reconocer un patrón antes de que comience la captura previa.

[0024] Además, una bifurcación de final de bucle puede estar marcada estáticamente por un compilador o ensamblador. Por ejemplo, en una realización, un compilador genera un tipo particular de instrucción de bifurcación, mediante el uso de un código de operación único, o mediante la configuración de un campo de bits de formato especial, que solo se usa para bifurcaciones de fin de bucle. La bifurcación por fin de bucle puede detectarse luego fácilmente durante la ejecución del conducto, tal como durante una etapa de descodificación en el conducto.

[0025] El circuito de lógica de predicción 122 comprende un filtro 150, una tabla de historia condicional (CHT) 152 y la lógica de monitorización asociada. En una realización, un proceso de supervisión guarda información de estado de sucesos de condición pre-especificados que se han producido en una o más ejecuciones previas de un bucle de software que tiene una instrucción de bifurcación condicional que es elegible para predicción. En apoyo del circuito lógico de predicción 122, el filtro 150 determina si se ha recibido una instrucción de bifurcación condicional capturada y si está habilitada la CHT 152. Se selecciona una entrada en la CHT 152 para proporcionar información de predicción que es rastreada, por ejemplo, por las etapas del conducto 132 a 135 a medida que las instrucciones se desplazan por el conducto.

[0026] La entrada de la CHT 152 registra la historia de la ejecución de la instrucción capturada, elegible para su ejecución predicha. Por ejemplo, cada entrada de la CHT puede comprender adecuadamente una combinación de valores de recuento de contadores de estado de ejecución y bits de estado que son entradas para la lógica de predicción. La CHT 152 también puede comprender una lógica de índice para permitir que una instrucción de bifurcación condicional capturada sirva como índice para una entrada en la CHT 152 asociada a la instrucción capturada, ya que pueden existir múltiples instrucciones de bifurcación condicional en un bucle de software. Por ejemplo, al contar el número de instrucciones de bifurcación condicional desde el extremo superior de un bucle de software, el recuento puede usarse como un índice para la CHT 152. El circuito lógico de predicción 122 incluye contadores de bucle para contar iteraciones de bucles de software y garantizar que los contadores de estado de ejecución hayan tenido la oportunidad de saturarse en un valor de recuento especificado que represente, por ejemplo, un estado sumamente no ejecutado. Si un contador de estado de ejecución se ha saturado, la lógica de predicción está habilitada para realizar una predicción de la dirección de bifurcación de la instrucción asociada de bifurcación condicional capturada en la siguiente iteración del bucle.

[0027] El circuito de lógica de predicción 122 genera información de predicción que se rastrea en la etapa de emisión de instrucciones 132, la etapa de captura de operandos 133, la etapa de ejecución 134 y la etapa de terminación 135 en la emisión de registro de rastreo (TRI) 162, la captura de operandos de registro de rastreo 163, la ejecución de registro de rastreo (TrE) 164 y la completitud del registro de rastreo (TrC) 165, respectivamente. Cuando se detecta una bifurcación condicional inversa con un predicado fallido que indica el final del bucle, o un retorno de función, tal como durante la etapa de ejecución 134 en el conducto procesador, se genera una señal de cancelación de solicitudes pendientes de captura previa 155. En otra realización, las solicitudes pendientes de captura previa se cancelan basándose en una predicción de bifurcación condicional generada por la lógica de predicción de bifurcación. Cada bifurcación condicional generalmente es predicha por la lógica de predicción de bifurcación. para tomar o no la bifurcación condicional. Por ejemplo, cuando la información de predicción indica que se toma la bifurcación condicional que, en este ejemplo, continúa un bucle de programa, el capturador de instrucciones captura especulativamente instrucciones en el bucle de programa indicado por la predicción. La información de predicción también está acoplada a un circuito lógico de cancelación de solicitud pendiente de captura previa 141 que puede residir en el circuito de captura y captura previa 130. El circuito lógico de cancelación de solicitud pendiente de captura previa 141 puede cancelar entonces especulativamente las solicitudes pendientes de captura previa, basándose en la información de flujo de programa que indica que las solicitudes pendientes de captura previa no son necesarias. Por ejemplo, el procesador puede configurarse para que no cancele las solicitudes pendientes de captura previa basándose en una salida de bucle predicha débilmente. Al cancelar una o más solicitudes pendientes de captura previa de datos, se reduce la contaminación de la memoria caché de datos y se reduce la potencia utilizada para abordar dicha contaminación en el procesador 110. La señal de cancelación de solicitud pendiente de captura previa 155 está acoplada al conducto de instrucciones del procesador 120, como se muestra en la FIG. 1, y es aceptada por el circuito lógico de cancelación de solicitud pendiente de captura previa 141, que provoca que se cancelen las solicitudes de captura previa que estén pendientes, excepto las solicitudes de captura previa de demanda. Además, el rendimiento del procesador se mejora al no almacenar datos innecesarios en la memoria caché de datos, que puede haber desalojado datos que hubieran sido capturados y ahora, en cambio, se genera un error.

[0028] Al llegar a la etapa de ejecución 134, si la condición de ejecución especificada para la instrucción de bifurcación condicional de final de bucle ha sido evaluada como opuesta a su predicción, se corrige cualquier ejecución especulativa del conducto de instrucciones en el trayecto erróneo de instrucciones, por ejemplo, mediante el vaciado del conducto, y tal corrección puede incluir la cancelación de capturas previas pendientes que están asociadas al trayecto erróneo de instrucciones. Por ejemplo, en una realización, una corrección para el conducto incluye vaciar las instrucciones en el conducto que comiencen en la etapa en la que se realizó la predicción. En una realización alternativa, el conducto se vacía desde la etapa de captura inicial, donde la instrucción de bifurcación condicional del fin del bucle fue capturada inicialmente. Además, la entrada adecuada de la CHT también puede corregirse después de una predicción incorrecta.

[0029] El circuito de detección 146, que actúa como un detector de bucle, funciona para detectar una bifurcación de fin de bucle. Por ejemplo, una bifurcación de fin de bucle es generalmente una instrucción de bifurcación condicional que bifurca inversamente al inicio del bucle para todas las iteraciones del bucle, excepto para la última iteración que sale del bucle. La información relativa a cada bucle identificado se pasa al circuito de filtro 150 y, en una situación de salida de bucle, un circuito lógico de cancelación de solicitud pendiente de captura previa 141 cancela solicitudes pendientes de captura previa no de demanda, en respuesta a cada salida de bucle identificada.

[0030] En una realización, el circuito de filtro 150, por ejemplo, es un contador de bucle que proporciona una indicación de que ha ocurrido un número determinado de iteraciones de un bucle de software, tal como tres iteraciones de un bucle particular. Para cada iteración del bucle, el filtro determina si una instrucción de bifurcación condicional es elegible para la predicción. Si una instrucción de bifurcación condicional (CB) elegible está en el bucle, el estado de ejecución de la instrucción de CB se registra en el circuito de la tabla de historia condicional (CHT) 152. Por ejemplo, un contador de estado de ejecución se puede usar para registrar una historia de ejecución de ejecuciones anteriores intentadas de una instrucción de CB elegible. Un contador de estado de ejecución se actualiza en una dirección para indicar la instrucción de CB ejecutada condicionalmente y, en una dirección opuesta, para indicar que la instrucción de CB no se ejecutó condicionalmente. Por ejemplo, se puede usar un contador de estado de ejecución de dos bits donde un estado no ejecutado causa una disminución del contador y un estado ejecutado causa un incremento del contador. A los estados de salida del contador de estado de ejecución se asignan, por ejemplo, una salida de "11" para indicar que se indica firmemente que las instrucciones de CB anteriores han sido ejecutadas, una salida de "10" para indicar que se indica inciertamente que las instrucciones de CB anteriores han sido ejecutadas, una salida de "01" para indicar que se indica inciertamente que las instrucciones de CB anteriores no hayan sido ejecutadas, y una salida de "00" para indicar que se indica firmemente que las instrucciones de CB anteriores no hayan sido ejecutadas. La salida del contador "11" y la salida "00" del estado de ejecución serían valores de salida saturados. Un contador de estado de ejecución se asociaría a, o proporcionaría estado para, cada instrucción de CB en un bucle de software detectado. Sin embargo, una implementación particular puede limitar el número de contadores de estado de ejecución que se utilizan en la implementación y, por lo tanto, limitar el número de instrucciones de CB que se predicen. El circuito de detección 146 generalmente restablece los contadores de estado de ejecución en la primera entrada a un bucle de software.

[0031] Alternativamente, un indicador de inhabilitación de predicción puede estar asociado a cada instrucción de CB a predecir, en lugar de un contador de estado de ejecución. El indicador de inhabilitación de predicción se activa para inhabilitar la predicción si se ha determinado previamente que se ha ejecutado una instrucción de CB asociada. La identificación de una instrucción de CB previa que se ha ejecutado implica que el nivel de confianza para predecir una situación de no ejecución para la instrucción de CB sería inferior a un nivel aceptable.

[0032] Un contador de índice también se puede usar con la CHT 152 para determinar qué instrucción de CB se está contando o evaluando en el bucle de software. Por ejemplo, en un bucle que tenga cinco o más instrucciones de CB, la primera instrucción de CB podría tener un índice de "000" y la cuarta instrucción de bifurcación condicional elegible podría tener un índice de "011". El índice representa una dirección dentro de la CHT 152, para acceder a los valores almacenados del contador de estado de ejecución, para la instrucción de CB correspondiente.

[0033] El circuito de predicción 122 recibe la información de predicción para una instrucción de CB en particular, tal como los valores de salida del contador de estado de ejecución, y predice, durante la etapa de decodificación 131 de la FIG. 1, por ejemplo, que la instrucción de CB generalmente bifurcará de nuevo al inicio del bucle de software, y no predirá que se ha alcanzado una situación de salida del bucle. En una realización, el circuito de predicción 122 puede predecir que la condición especificada por la instrucción de CB se evalúa como un estado de no bifurcación, sale del código o atraviesa el bucle. El circuito de predicción 122 rastrea la instrucción de CB. Si se predice que una instrucción de CB se bifurca de nuevo al inicio del bucle, la información de predicción indica tal estado. Si se determinó que una instrucción de CB no se bifurca inversamente, entonces un circuito de rastreo genera una señal de cancelación de solicitud pendiente de captura previa y se realiza una evaluación de condición para determinar si se realizó una predicción incorrecta. Si se realizó una predicción incorrecta, el conducto también se puede vaciar, los contadores adecuados de estado de ejecución en la CHT 152 se actualizan y, en una realización, la entrada asociada de la CHT se marca para indicar que esta instrucción de CB particular no se ha de predecir a partir de este momento. En otra realización, el circuito de lógica de predicción 122 también puede cambiar el criterio de evaluación pre-especificado al determinar que la instrucción de CB fue mal predicha, por ejemplo, para hacer que el criterio de predicción sea más conservador a partir de este momento.

[0034] Además, se reconoce que no todos los bucles tienen características similares. Si un bucle particular proporciona malos resultados de predicción, ese bucle se marca en el circuito de lógica de predicción 122 para desactivar la predicción. De manera similar, un bucle particular puede funcionar con buena predicción en un conjunto de escenarios operativos y puede funcionar con mala predicción en un conjunto diferente de escenarios operativos. En tal caso, el reconocimiento de los escenarios operativos permite que la predicción se habilite, deshabilite o habilite, pero con un diferente criterio de evaluación, adecuado para el escenario operativo.

[0035] La FIG. 2A ilustra un proceso 200 para cancelar solicitudes pendientes de captura previa de datos no de demanda, al detectar una bifurcación de finalización de bucle. En el bloque 202, la ejecución del código del procesador se supervisa para un bucle de software. En el bloque de decisión 204, se toma una determinación en cuanto a si se ha detectado un bucle de software. Se puede determinar un bucle de software, por ejemplo, identificando una bifurcación inversa a una ubicación que representa el inicio del bucle de software en un primer paso a través del bucle de software, como se ha descrito anteriormente. Si no se ha identificado ningún bucle de software, el proceso 200 vuelve al bloque 202. Si se ha identificado un bucle de software, entonces el proceso 200 avanza hasta el bloque 206. En este punto en el código, ya se ha ejecutado un primer ciclo del bucle de software y el siguiente ciclo del bucle de software está listo para comenzar.

[0036] En el siguiente ciclo del bucle de software en el bloque 206, el código del procesador se monitoriza para una instrucción de CB. En la etapa de decisión 208 se toma una determinación en cuanto a si se ha detectado una instrucción de CB, por ejemplo, durante una etapa de decodificación del conducto, tal como la etapa de decodificación 131 de la FIG. 1. Si no se ha detectado ninguna instrucción de CB, el proceso 200 regresa al bloque 206. Si se ha detectado una instrucción de CB, el proceso 200 avanza al bloque de decisión 210. En el bloque de decisión 210, se toma una determinación en cuanto a si la instrucción de bifurcación condicional (CB) resolvió finalizar el bucle, basándose en una evaluación del predicado condicional, por ejemplo. Hay varios tipos de evaluaciones de instrucciones de CB que pueden haberse detectado. Por ejemplo, una primera evaluación de la instrucción de CB detectada podría resolverse en cuanto a que la instrucción de CB está al final del bucle de software, pero se evalúa para continuar con el procesamiento del bucle. La instrucción de CB de bifurcación inversa que identificó el bucle de software en el primer paso por el bucle de software está etiquetada por su ubicación de dirección en el código del procesador, por ejemplo. Además, en el caso de que no se hayan completado una serie de iteraciones especificadas del bucle de software, la instrucción de CB resuelve volver a bifurcar el procesador al comienzo del bucle de software. Una segunda evaluación de la instrucción de CB detectada podría resolverse en cuanto a que la instrucción de CB está al final del bucle de software y se evalúa para finalizar el bucle de software. Una tercera evaluación de la instrucción de CB detectada podría resolverse en cuanto a que la instrucción de CB está dentro del bucle de software, pero cuando se evalúa como adoptada o no adoptada, el código del procesador permanece en el bucle de software. Además, una cuarta evaluación de la instrucción CB podría resolverse en cuanto a que la instrucción de CB está dentro del bucle de software, pero, cuando se evalúa como adoptada o no adoptada, el código del procesador sale del bucle de software. En la cuarta evaluación, se considera que una

instrucción de CB que está dentro del bucle de software, pero que se resuelve como una bifurcación directa más allá de la ubicación de dirección de la instrucción de CB de bifurcación inversa, ha salido del bucle de software.

[0037] Volviendo al bloque de decisión 210, si la instrucción de CB detectada no resolvió salir del bucle de software, como en las evaluaciones primera y tercera de la instrucción de CB, el proceso 200 avanza al bloque 212. En el bloque 212, el proceso 200 continúa con el procesamiento de bifurcación normal y luego regresa al bloque 206. Si la instrucción de CB detectada resolvió salir del bucle de software, como en las evaluaciones segunda y cuarta de la instrucción de CB, el proceso 200 avanza al bloque 214. En el bloque 214, el proceso 200 cancela las solicitudes pendientes de captura previa de datos, excepto las solicitudes de captura previa de datos por demanda, procesa la instrucción de CB y vuelve al bloque 202 para comenzar a buscar el siguiente bucle de software.

[0038] La FIG. 2B ilustra un proceso 250 para cancelar solicitudes pendientes de captura previa de datos no de demanda al detectar un retorno de función. En el bloque 252, se supervisa la ejecución del código del procesador para una salida de función de software. Se observa que la función de software puede ejecutarse especulativamente. Por ejemplo, la ejecución especulativa puede ocurrir para una llamada de función en un bucle de software. En el caso de ejecución especulativa de la función de software, la salida de la función de software, tal como la ejecución de una instrucción RET, también puede ejecutarse especulativamente. En el bloque de decisión 254, se toma una determinación en cuanto a si se ha detectado una salida de función de software, tal como al detectar una instrucción de retorno en el conducto de ejecución de un procesador. Si no se ha detectado ninguna salida de función de software, el proceso 250 vuelve al bloque 252.

[0039] Si se ha detectado una salida de función de software, el proceso 250 prosigue hasta el bloque de decisión 256. En el bloque de decisión 256, se toma una determinación en cuanto a si esta situación de salida detectada es un retorno desde una rutina de interrupción. Si la salida detectada es un retorno desde una rutina de interrupción, entonces el proceso 250 regresa al bloque 252. Si la salida detectada no es un retorno desde una rutina de interrupción, el proceso 250 continúa al bloque 258. En el bloque 258, el proceso 250 cancela las solicitudes pendientes de captura previa de datos, a excepción de solicitudes de captura previa de datos demandados, procesa la instrucción de retorno y luego regresa al bloque 252 para continuar monitorizando el código del procesador con respecto a una salida de función de software.

[0040] Con frecuencia, ya sea a mano o por medio de optimizaciones del compilador, un bucle de software se desenrollará de manera que múltiples iteraciones del bucle se ejecuten secuencialmente. Esta ejecución secuencial de cada iteración desenrollada se convierte en un candidato de captura previa adicional. En la última iteración del bucle, cada candidato desenrollado puede generar entonces solicitudes de captura previa innecesarias que agravan el problema de la contaminación de la memoria caché de datos previamente capturados. Una realización de la invención también se aplica al desenrollado del bucle detectando la salida del bucle, o el retorno desde una función, y cancelando todas las solicitudes de captura previa innecesarias procedentes de cada bucle desenrollado.

[0041] La FIG. 3 ilustra una realización particular de un dispositivo portátil 300 que tiene un complejo procesador que está configurado para cancelar solicitudes pendientes de captura previa de datos, seleccionadas para reducir la contaminación de la memoria caché. El dispositivo 300 puede ser un dispositivo electrónico inalámbrico e incluir el complejo procesador 310 acoplado a una memoria de sistema 312 que tiene instrucciones de software 318. La memoria del sistema 312 puede incluir la memoria del sistema 114 de la FIG. 1. El complejo procesador 310 puede incluir un procesador 311, un subsistema de memoria integrado 314 que tiene una memoria caché de datos de nivel 1 (Dcache de L1) 222, una memoria caché de instrucciones de nivel 1 (Icache de L1) 326, un circuito controlador de memoria caché 328 y lógica de predicción 316. El procesador 311 puede incluir el procesador 110 de la FIG. 1. El subsistema de memoria integrado 314 también puede incluir una memoria caché unificada de nivel 2 (no mostrada). La Icache de L1 326 puede incluir la Icache de L1 124 de la FIG. 1 y la Dcache de L1 322 puede incluir la memoria Dcache de L1 128 de la FIG. 1.

[0042] El subsistema de memoria integrado 314 puede estar incluido en el complejo procesador 310 o puede ser implementado como uno o más dispositivos o circuitos individuales (no mostrados) externos al complejo procesador 310. En un ejemplo ilustrativo, el complejo procesador 310 funciona de acuerdo a cualquiera de las realizaciones ilustradas en, o asociadas a, las FIGs. 1 y 2. Por ejemplo, como se muestra en la FIG. 3, la Icache de L1 326, la Dcache de L1 322 y el circuito controlador de memoria caché 328 son accesibles dentro del complejo procesador 310, y el procesador 311 está configurado para acceder a datos o instrucciones de programa almacenadas en las memorias del subsistema de memoria integrado 314 o en la memoria del sistema 312.

[0043] Una interfaz de cámara 334 está acoplado al complejo procesador 310 y también acoplada a una cámara, tal como una cámara de vídeo 336. Un controlador de visualización 340 está acoplado al complejo procesador 310 y a un dispositivo de visualización 342. También se puede acoplar un codificador / decodificador (CÓDEC) 344 al complejo procesador 310. Un altavoz 346 y un micrófono 348 se pueden acoplar al CÓDEC 344. Una interfaz inalámbrica 350 puede estar acoplada al complejo procesador 310 y a una antena inalámbrica 352 de manera que los datos inalámbricos recibidos a través de la antena 352 y la interfaz inalámbrica 350 puedan proporcionarse al procesador 311.

[0044] El procesador 311 puede estar configurado para ejecutar instrucciones de software 318 almacenadas en un medio legible por ordenador no transitorio, tal como la memoria del sistema 312, que son ejecutables para hacer que un ordenador, tal como el procesador 311, ejecute un programa, tal como el proceso de programa 200 de la FIG. 2. Las instrucciones de software 318 son además ejecutables para hacer que el procesador 311 procese instrucciones que accedan a las memorias del subsistema de memoria integrado 314 y a la memoria del sistema 312.

[0045] En un modo de realización particular, el complejo procesador 310, el controlador de visualización 340, la memoria del sistema 312, el CÓDEC 344, la interfaz inalámbrica 350 y la interfaz de cámara 334 pueden estar incluidos en un dispositivo de sistema en un paquete o de sistema en un chip 304. En un modo de realización particular, un dispositivo de entrada 356 y una fuente de alimentación 358 están acoplados al dispositivo de sistema en un chip 304. Además, en un modo de realización particular, como se ilustra en la FIG. 3, el dispositivo de visualización 342, el dispositivo de entrada 356, el altavoz 346, el micrófono 348, la antena inalámbrica 352, la cámara de vídeo 336 y la fuente de alimentación 358 son externos al dispositivo de sistema en un chip 304. Sin embargo, cada uno entre el dispositivo de visualización 342, el dispositivo de entrada 356, el altavoz 346, el micrófono 348, la antena inalámbrica 336 y la fuente de alimentación 358 se puede acoplar a un componente del dispositivo de sistema en un chip 304, tal como una interfaz o un controlador.

[0046] El dispositivo 300, de acuerdo a formas de realización descritas en este documento, puede ser incorporado en varios dispositivos, tales como un equipo de sobremesa, una unidad de entretenimiento, un dispositivo de navegación, un dispositivo de comunicaciones, un asistente digital personal (PDA), una unidad de datos de ubicación fija, una unidad de datos de ubicación móvil, un teléfono móvil, un teléfono celular, un ordenador, un ordenador portátil, tabletas, un monitor, un monitor de ordenador, un televisor, un sintonizador, una radio, una radio satelital, un reproductor de música, un reproductor de música digital, un reproductor de música portátil, un reproductor de vídeo, un reproductor de vídeo digital, un reproductor de discos de vídeo digitales (DVD), un reproductor de vídeo digital portátil, cualquier otro dispositivo que almacene o recupere datos o instrucciones de ordenador, o cualquier combinación de los mismos.

[0047] Los diversos bloques lógicos, módulos, circuitos, elementos y/o componentes ilustrativos descritos en relación con los ejemplos divulgados en el presente documento pueden implementarse o realizarse con un procesador de uso 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 otros componentes 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, micro-controlador 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, adecuada para una aplicación deseada.

[0048] Los procedimientos o algoritmos descritos en relación con los modos de realización divulgados 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 memoria RAM, memoria flash, memoria ROM, memoria EPROM, memoria EEPROM, registros, un disco duro, un disco extraíble, un CD-ROM o en cualquier otra forma de medio de almacenamiento no transitorio conocida en la técnica. Un medio de almacenamiento no transitorio 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 no transitorio puede estar integrado en el procesador.

[0049] El procesador 110 de la FIG. 1 o el procesador 311 de la FIG. 3, por ejemplo, pueden configurarse para ejecutar instrucciones que incluyen instrucciones condicionales no de bifurcación, bajo el control de un programa almacenado en un medio de almacenamiento no transitorio legible por ordenador, directamente asociado localmente al procesador, tal como puede estar disponible mediante una memoria caché de instrucciones, o bien accesible a través de un dispositivo de I/O, tal como uno de los dispositivos de I/O 140 o 142 de la FIG. 1, por ejemplo. El dispositivo de I/O también puede acceder a datos que residen en un dispositivo de memoria, ya sea directamente asociados localmente a los procesadores, tales como la Dcache 128, o bien accesibles desde la memoria de otro procesador. El medio de almacenamiento no transitorio legible por computadora puede incluir memoria de acceso aleatorio (RAM), memoria dinámica de acceso aleatorio (DRAM), memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria flash, memoria de solo lectura (ROM), memoria programable de solo lectura (PROM), memoria de solo lectura programable borrable (EPROM), memoria de solo lectura programable y borrable eléctricamente (EEPROM), disco compacto (CD), disco de vídeo digital (DVD), otros tipos de discos extraíbles o cualquier otro medio adecuado de almacenamiento no transitorio.

[0050] Si bien la invención se da a conocer en el contexto de realizaciones ilustrativas para su uso en sistemas procesadores, 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. Por ejemplo, una implementación de función fija también puede utilizar varias realizaciones de la

presente invención.

REIVINDICACIONES

1. Un procedimiento (200) para cancelar solicitudes de captura previa de memoria caché de datos no de demanda, en un sistema procesador (100) que comprende un procesador (110) que tiene un sistema de memoria caché (112) que comprende una memoria caché de datos (124) y que tiene un conducto de instrucciones (120), comprendiendo el procedimiento:

determinar un paso de acceso a datos basado en la ejecución repetida de una instrucción de acceso a memoria en un bucle de programa;

emitir especulativamente solicitudes de captura previa de la memoria caché de datos de acuerdo al paso de acceso a los datos;

identificar (210) una salida de bucle basándose en una evaluación de información de flujo de programa; y **caracterizado por**:

cancelar (214) las solicitudes de captura previa de memoria caché de datos que son solicitudes pendientes de captura previa de memoria caché de datos no de demanda en respuesta a la salida de bucle identificada.

2. El procedimiento de la reivindicación 1, en el que la salida de bucle se basa en la identificación de una bifurcación de fin de bucle que evalúa salir del bucle de programa.

3. El procedimiento de la reivindicación 1, en el que la salida del bucle se basa en una predicción de bifurcación incorrecta que provocó que se cancelara la captura y ejecución especulativa de instrucciones.

4. El procedimiento de la reivindicación 1, en el que la identificación de la salida de bucle comprende detectar que una instrucción de bifurcación condicional ha resuelto finalizar el bucle de programa.

5. El procedimiento de la reivindicación 1 que comprende además:

detectar que una instrucción de bifurcación condicional no ha resuelto finalizar el bucle del programa; y

monitorizar (202) con respecto a una salida de bucle.

6. Un aparato (110) para cancelar solicitudes de captura previa de memoria caché de datos no de demanda, en un sistema procesador (100) que comprende un procesador (110) que tiene un sistema de memoria caché (112) que comprende una memoria caché de datos (124) y que tiene un conducto de instrucciones (120), comprendiendo el aparato:

un monitor de direcciones de datos de bucle configurado para determinar un paso de acceso a datos basado en la ejecución repetida de una instrucción de acceso a memoria en un bucle de programa;

lógica de captura previa de datos (121) configurada para emitir especulativamente solicitudes de captura previa de memoria caché de datos de acuerdo al paso de acceso a los datos;

medios para identificar (210) una salida de bucle basándose en una evaluación de información de flujo de programa; y **caracterizado por**:

un circuito de detención de captura previa configurado para cancelar las solicitudes de captura previa de memoria caché de datos que son solicitudes pendientes de captura previa de memoria caché de datos no de demanda en respuesta a la salida del bucle identificada.

7. El aparato de la reivindicación 6, en el que el monitor de direcciones de datos de bucle comprende:

un circuito de paso (119) configurado para supervisar la ejecución repetida de la instrucción de acceso a memoria para determinar una diferencia en una dirección de operando para cada ejecución de la instrucción de acceso a memoria, en donde la diferencia en la dirección de operando es un valor de dirección de paso; y

un circuito de función de suma configurado para agregar el valor de dirección de paso a la dirección de operando de la instrucción de acceso a memoria ejecutada más recientemente para determinar la siguiente dirección de operando.

8. El aparato de la reivindicación 6, en el que la salida de bucle identificada se basa en la identificación de una bifurcación de final de bucle que evalúa salir del bucle de programa.

9. El aparato de la reivindicación 6, en el que la salida de bucle identificada se basa en una predicción de bifurcación incorrecta que cancela la captura y ejecución especulativa de instrucciones.

10. El aparato de la reivindicación 6, en el que la salida de bucle identificada se basa en la detección de una instrucción de bifurcación condicional que ha resuelto finalizar el bucle de programa.

11. El aparato de la reivindicación 6, en el que el circuito de detención de captura previa está configurado además para detectar que una instrucción de bifurcación condicional no ha resuelto finalizar el bucle de programa y en el que el bucle de programa continúa hasta que se identifica la salida de bucle.

12. El aparato de la reivindicación 6, en el que el circuito de detención de captura previa está configurado además para no cancelar solicitudes pendientes de captura previa basadas en una salida de bucle inciertamente predicha.

13. Un medio no transitorio legible por ordenador, codificado con datos y código de programa legibles por ordenador; los datos y código de programa, cuando son ejecutados por un procesador operable para realizar un procedimiento de acuerdo a cualquiera de las reivindicaciones 1 a 5.

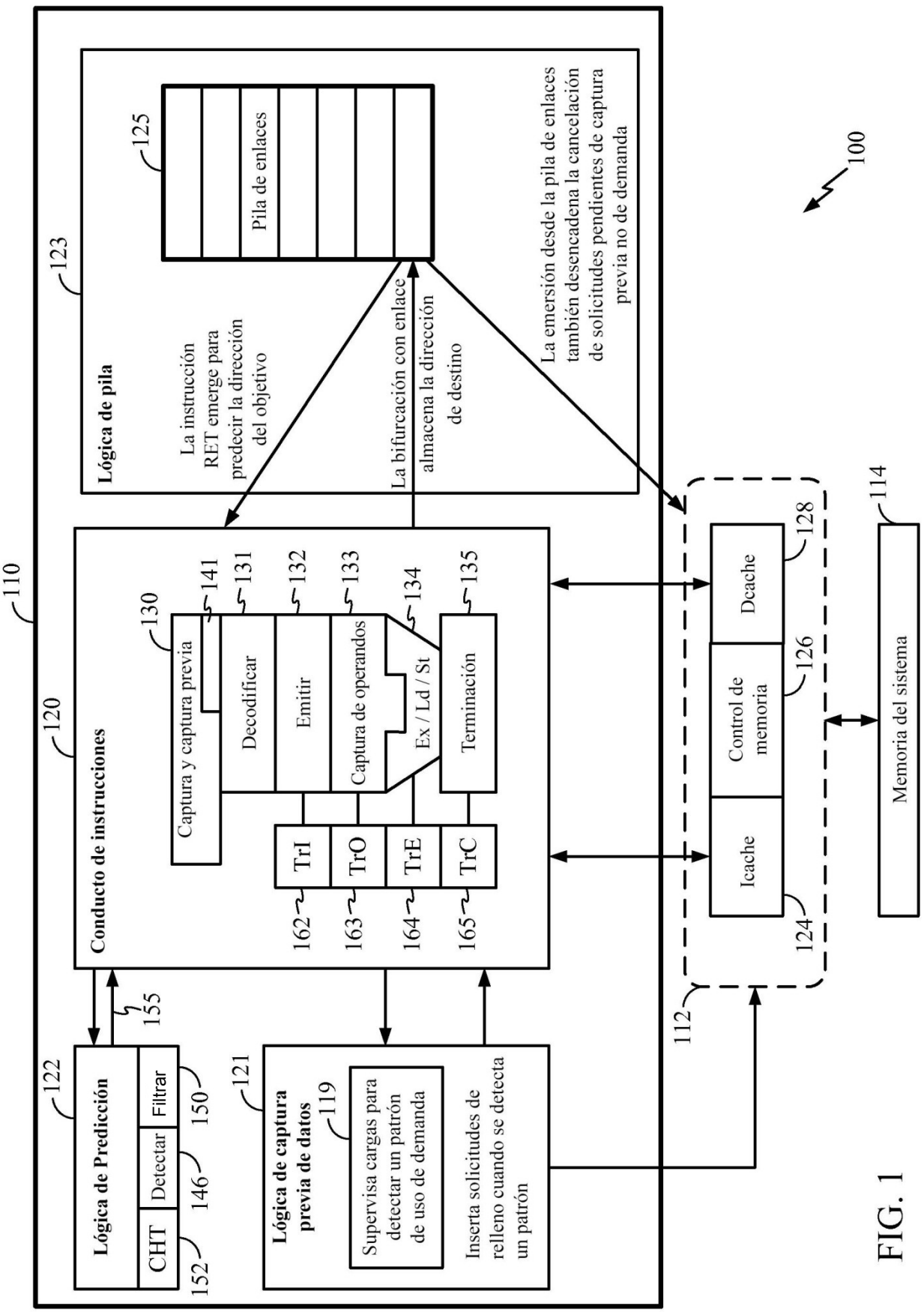
100
110
120
Conducto de instrucciones
130
Captura y captura previa
141
131
Decodificar
132
Emitir
133
Captura de operandos
134
Ex / Ld / St
135
Terminación
TrI
TrO
TrE
TrC
162
163
164
165
123
Lógica de pila
125
Pila de enlaces
La instrucción RET emerge para predecir la dirección del objetivo
La bifurcación con enlace almacena la dirección de destino
La emersión desde la pila de enlaces también desencadena la cancelación de solicitudes pendientes de captura previa no de demanda
122
Lógica de Predicción
CHT
Detectar
Filtrar
152
146
150
155
121
Lógica de captura previa de datos
119
Supervisa cargas para detectar un patrón de uso de demanda
Inserta solicitudes de relleno cuando se detecta un patrón
112
Icache
Control de memoria
Dcache
124
126
128
114
Memoria del sistema

FIG. 1

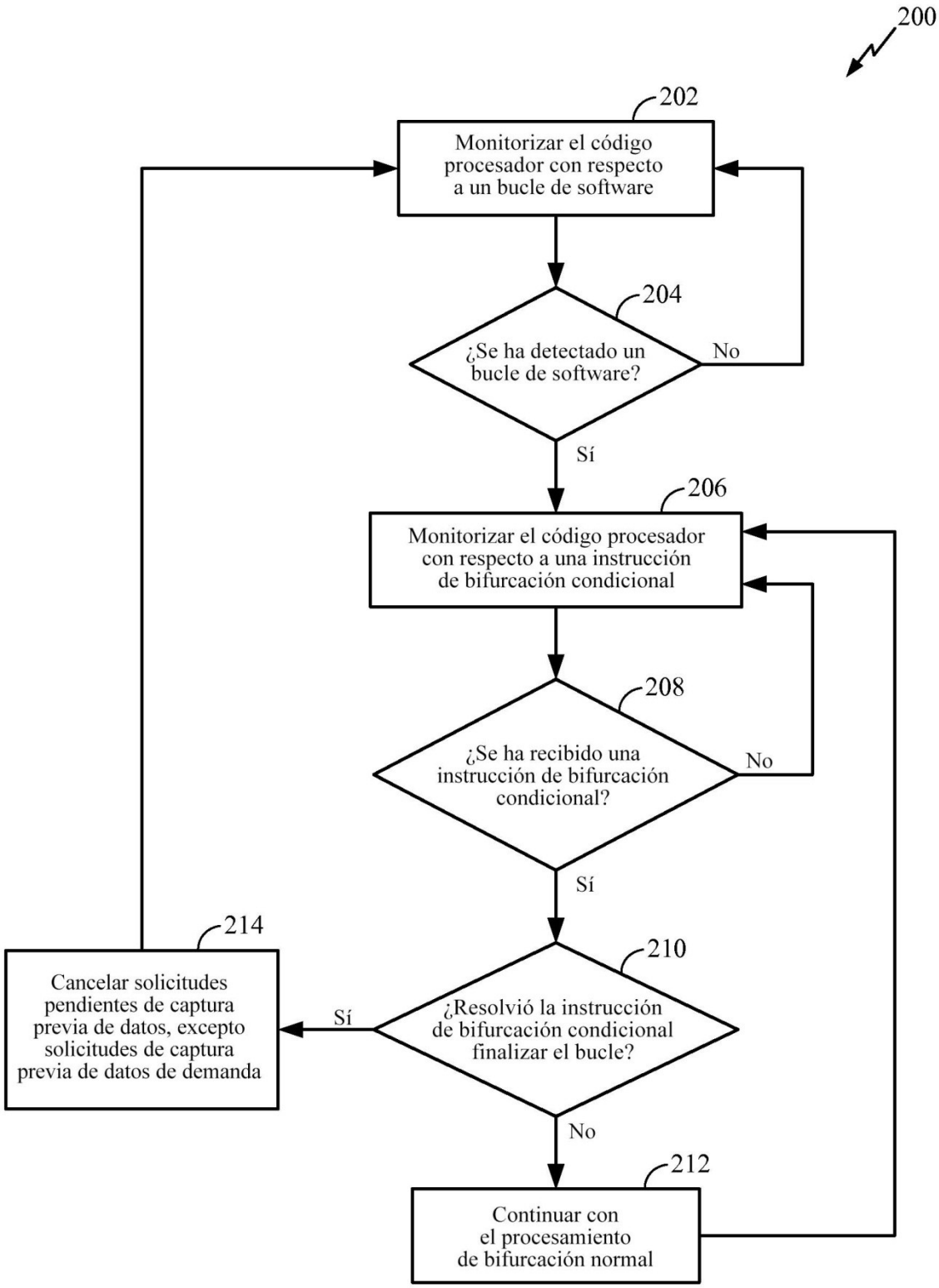
200
202
Monitorizar el código procesador con respecto a un bucle de software
204
¿Se ha detectado un bucle de software?
No
Sí
206
Monitorizar el código procesador con respecto a una instrucción de bifurcación condicional
208
¿Se ha recibido una instrucción de bifurcación condicional?
No
Sí
210
¿Resolvió la instrucción de bifurcación condicional finalizar el bucle?
Sí
214
Cancelar solicitudes pendientes de captura previa de datos, excepto solicitudes de captura previa de datos de demanda
No
212
Continuar con el procesamiento de bifurcación normal

FIG. 2A

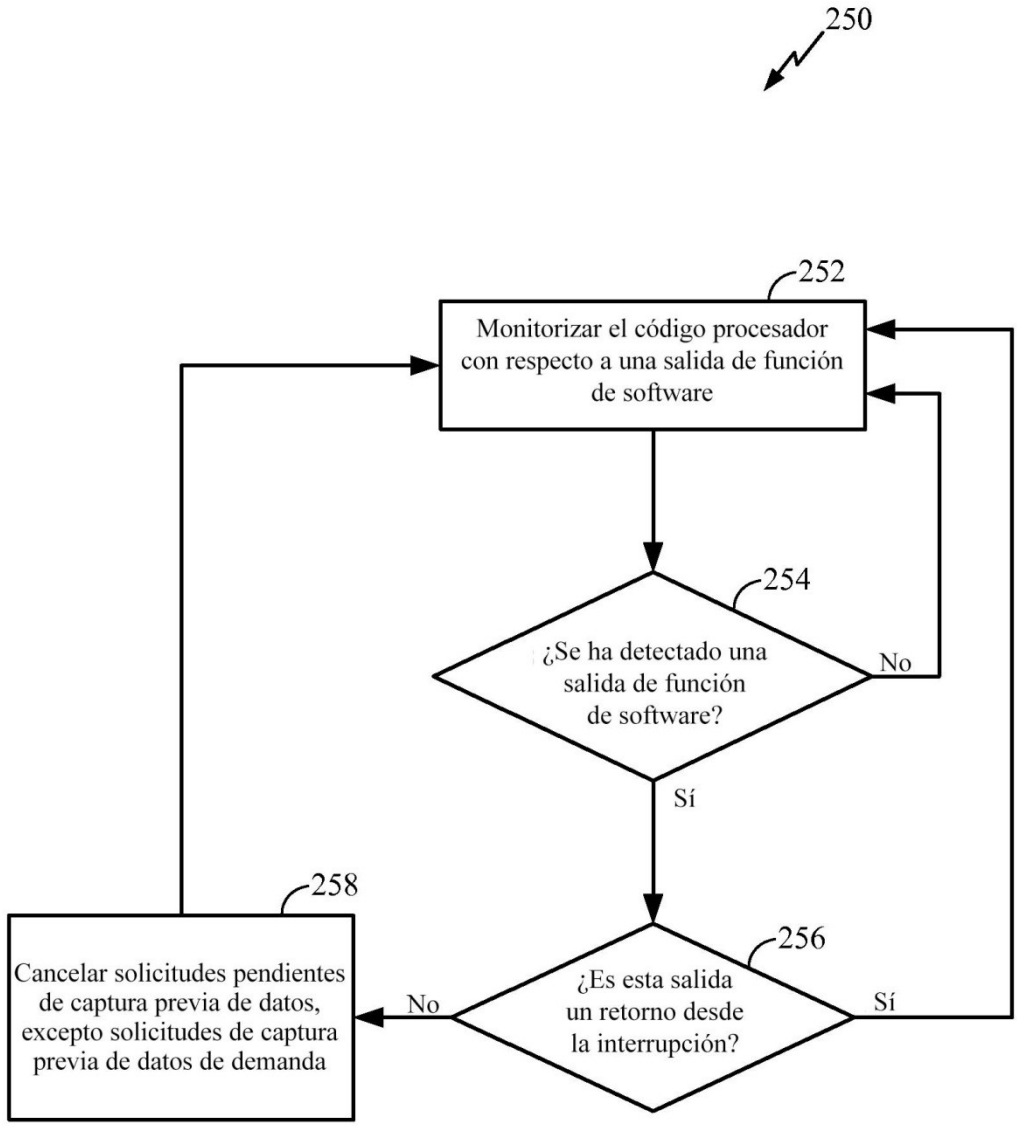
250
252
Monitorizar el código procesador con respecto a una salida de función de software
254
¿Se ha detectado una salida de función de software?
No
Sí
256
¿Es esta salida un retorno desde la interrupción?
No
Sí
258
Cancelar solicitudes pendientes de captura previa de datos, excepto solicitudes de captura previa de datos de demanda

FIG. 2B

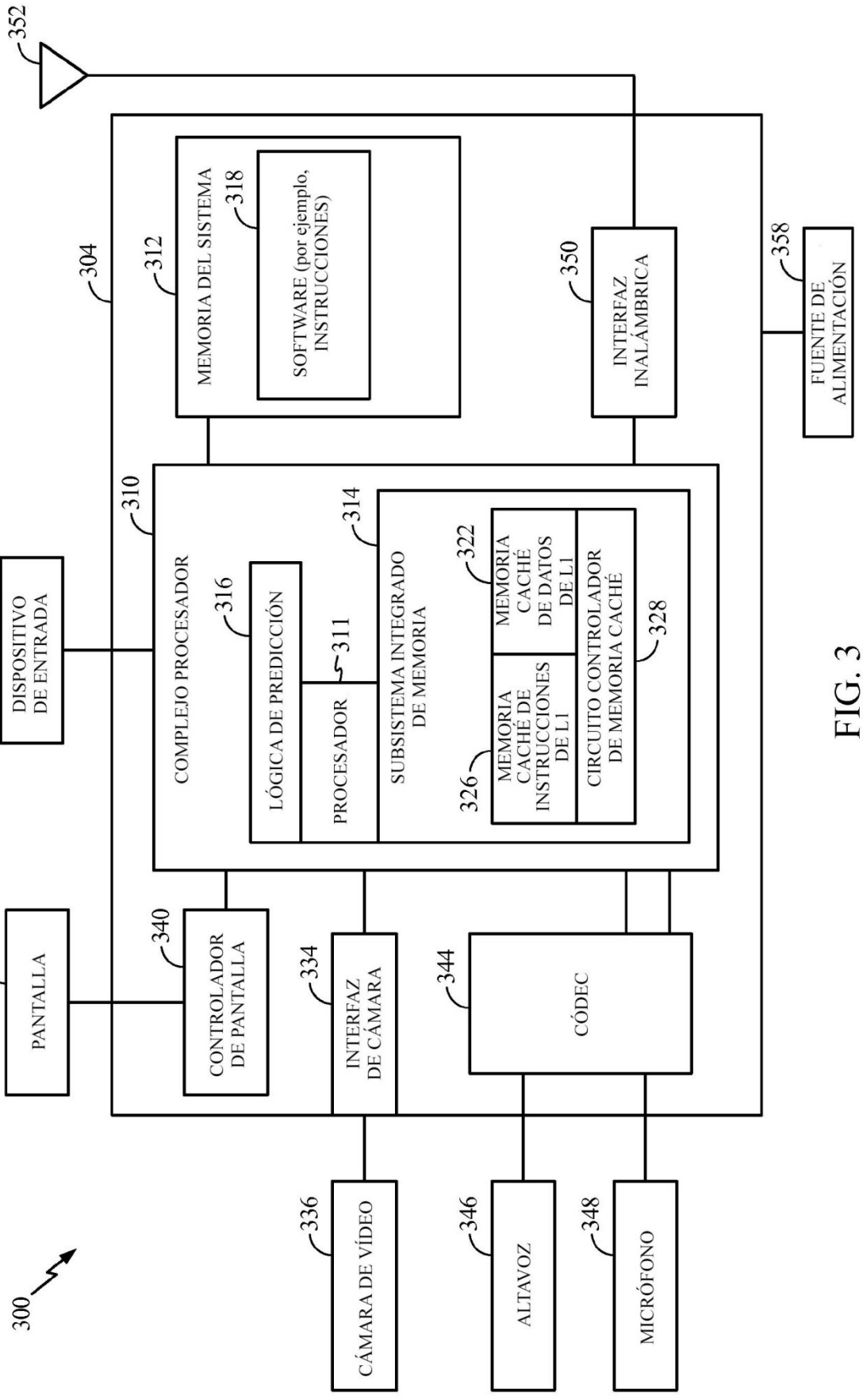
352
304
312
MEMORIA DEL SISTEMA
318
SOFTWARE (por ejemplo, INSTRUCCIONES)
350
INTERFAZ INALÁMBRICA
358
FUENTE DE ALIMENTACIÓN
310
COMPLEJO PROCESADOR
316
LÓGICA DE PREDICCIÓN
311
PROCESADOR
314
SUBSISTEMA INTEGRADO DE MEMORIA
322
MEMORIA CACHÉ DE DATOS DE L1
326
MEMORIA CACHÉ DE INSTRUCCIONES DE L1
328
CIRCUITO CONTROLADOR DE MEMORIA CACHÉ
356
DISPOSITIVO DE ENTRADA
342
PANTALLA
340
CONTROLADOR DE PANTALLA
334
INTERFAZ DE CÁMARA
344
CÓDEC
336
CÁMARA DE VÍDEO
346
ALTAVOZ
348
MICRÓFONO
300

FIG. 3