



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 345 733**

51 Int. Cl.:
G06F 9/38 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07710245 .7**

96 Fecha de presentación : **22.01.2007**

97 Número de publicación de la solicitud: **1974256**

97 Fecha de publicación de la solicitud: **01.10.2008**

54 Título: **Gestión eficiente de la jerarquía de memorias.**

30 Prioridad: **20.01.2006 US 336282**

45 Fecha de publicación de la mención BOPI:
30.09.2010

45 Fecha de la publicación del folleto de la patente:
30.09.2010

73 Titular/es: **QUALCOMM Incorporated**
5775 Morehouse Drive
San Diego, California 92121, US

72 Inventor/es: **Morrow, Michael William y**
Sartorius, Thomas Andrew

74 Agente: **Carpintero López, Mario**

ES 2 345 733 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

Gestión eficiente de la jerarquía de memorias.

5 Campo

La presente revelación se refiere, en general, a técnicas para extraer instrucciones de una memoria que tiene memoria caché para instrucciones y una memoria caché para datos y, más específicamente, a un enfoque mejorado para extraer una instrucción después de un fallo en la memoria caché de instrucciones, extrayendo directamente la instrucción de la memoria caché para datos si la instrucción reside allí.

Antecedentes

Los productos usualmente portátiles, tales como los teléfonos celulares, los ordenadores portátiles, las agendas electrónicas (PDA) o similares, requieren el uso de un procesador que ejecute programas tales como programas de comunicación y multimedia. El sistema de procesamiento para tales productos incluye un complejo de procesador y memoria para almacenar instrucciones y datos. Por ejemplo, las instrucciones y los datos pueden almacenarse en una memoria jerárquica que consiste en niveles múltiples de memorias caché, incluyendo, por ejemplo, una memoria caché para instrucciones, una memoria caché para datos y una memoria del sistema. El uso de una memoria caché separada para instrucciones y una memoria caché separada para datos se conoce como la arquitectura Harvard. Como la arquitectura Harvard aísla la memoria caché de instrucciones de la memoria caché para datos, pueden surgir problemas cuando las instrucciones se almacenan en la memoria caché para datos.

En el procesamiento general de sistemas con una arquitectura Harvard, hay situaciones que surgen, en las cuales las instrucciones pueden almacenarse en la memoria caché para datos. Por ejemplo, si un programa está cifrado o en formato comprimido, debe ser descifrado/descomprimido antes de permitir que el programa se ejecute. El proceso de descifrado/descompresión trata el programa cifrado/comprimido como datos a fin de procesarlo, y almacena las instrucciones descifradas/descomprimidas como datos en una memoria caché para datos, por ejemplo, una memoria caché para datos de nivel 1, en camino hacia la memoria del sistema. La generación de instrucciones a partir de códigos de octetos Java es otra situación en la cual las instrucciones se tratan inicialmente como datos que se almacenan usando la trayectoria de los datos, incluyendo la memoria caché para datos, hacia la memoria del sistema. El estado inicial de un programa en el cual las instrucciones de programa están tratándose como datos crea un problema de consistencia dentro de la jerarquía de memorias, dado que al menos algunas partes de un programa pueden residir en la memoria caché para datos antes de la ejecución del programa.

A fin de resolver el problema de consistencia, se adopta habitualmente un enfoque de software, en el cual el programa, o segmentos de programa, en la memoria caché para datos se llevan a la memoria del sistema bajo control de programa, la memoria caché para datos se invalida habitualmente para limpiar la memoria caché de segmentos de viejos programas, y las instrucciones que comprenden el programa se extraen luego de la memoria del sistema. El movimiento de las instrucciones desde la memoria caché para datos hasta la memoria del sistema, y la extracción de las instrucciones desde la memoria del sistema antes de la ejecución, pueden ocupar varios ciclos, reduciendo el rendimiento del procesador debido al sobregasto de tiempo de procesamiento que debe tener lugar para acceder a instrucciones que residen inicialmente en la memoria caché para datos antes de que el programa se ejecute en el procesador.

El documento US 2002/010837 describe un sistema y procedimiento de memoria de códigos para controlar la memoria caché.

Resumen

Entre sus diversos aspectos, la presente revelación reconoce que el sobregasto de tratar con instrucciones en una memoria caché para datos puede estar limitando el rendimiento del procesador y, posiblemente, limitando la calidad del servicio que puede lograrse. La presente revelación también reconoce que puede ser deseable acceder a instrucciones que están residentes en una memoria caché para datos.

Además, la presente revelación describe un aparato, procedimientos y un medio legible por ordenador para extraer directamente una instrucción desde una memoria caché para datos cuando esa instrucción no ha sido hallada en la memoria caché para instrucciones, un fallo de la memoria caché para instrucciones, y se determina que la instrucción está en la memoria caché para datos. Extrayendo la instrucción directamente de la memoria caché para datos, después de un fallo de la memoria caché para instrucciones, puede mejorarse el rendimiento del procesador.

Según la invención, se proporciona el procedimiento de la reivindicación 1, la disposición de procesador de la reivindicación 11 y el procedimiento de la reivindicación 15.

Una comprensión más completa de los presentes conceptos inventivos revelados en el presente documento, así como otras características, serán evidentes a partir de la siguiente Descripción Detallada y de los dibujos adjuntos.

Breve descripción de los dibujos

La Fig. 1 es un diagrama en bloques de un sistema ejemplar de comunicación inalámbrica.

5 La Fig. 2 es un diagrama en bloques funcionales de un complejo de procesador y memoria, en el cual el funcionamiento de la memoria caché para datos está adaptado para operaciones eficientes, en términos de memoria, de extracción de instrucciones;

10 La Fig. 3 es un diagrama de flujo de un procedimiento ejemplar para extraer una instrucción almacenada en una memoria caché para datos, a fin de reducir el sobregasto de gestión de fallos asociado a la instrucción inicialmente almacenada como datos en la memoria caché para datos;

15 La Fig. 4 es un diagrama en bloques funcionales de un complejo de procesador y memoria que incluye una tabla de páginas de instrucciones, en el cual el funcionamiento de la memoria caché para datos está adaptado para una eficiente extracción de instrucciones;

La Fig. 5 es un diagrama de flujo de un procedimiento ejemplar para extraer una instrucción almacenada en una memoria caché para datos; y

20 La Fig. 6 es un diagrama de flujo de un procedimiento ejemplar para ejecutar código que se genera como datos y que se almacena en una memoria caché para datos.

Descripción detallada

25 Aspectos inventivos de la presente revelación se ilustrarán más completamente con referencia a los dibujos adjuntos, en los cuales se muestran varias realizaciones de la revelación. La realización de esta invención, sin embargo, puede realizarse de diversas formas, y no debería interpretarse como limitada a las realizaciones expuestas en el presente documento. Antes bien, estas realizaciones se proporcionan a fin de que esta revelación sea exhaustiva y completa, y que transmita totalmente el alcance de la revelación a aquellos versados en la tecnología.

30 Se apreciará que la presente revelación puede realizarse como procedimientos, sistemas o productos de programa de ordenador. En consecuencia, los presentes conceptos inventivos revelados en el presente documento pueden tomar la forma de una realización de hardware, una realización de software o una realización que combina aspectos de software y de hardware. Además, los presentes conceptos inventivos revelados en el presente documento pueden tomar la forma de un producto de programa de ordenador en un medio de almacenamiento utilizable por ordenador, con código de programa utilizable por ordenador realizado en el medio. Cualquier medio adecuado legible por ordenador puede utilizarse, incluyendo los discos rígidos, los CD-ROM, los dispositivos ópticos de almacenamiento, las memorias flash o los dispositivos magnéticos de almacenamiento.

40 El código de programa de ordenador que pueda compilarse, ensamblarse y cargarse en un procesador puede escribirse inicialmente en un lenguaje de programación tal como C, C++, Ensamblador nativo, JAVA[®], Smalltalk, JavaScript[®], Visual Basic[®], TSQL, Perl, o en varios otros lenguajes de programación, de acuerdo a las enseñanzas de la presente revelación. El código de programa, o el medio legible por ordenador, se refiere a código de lenguaje de máquina, tal como el código objeto, cuyo formato es comprensible por un procesador. Las realizaciones de software de la revelación no dependen de su implementación con un lenguaje de programación específico. Cuando se ejecuta el código de programa, se crea una nueva tarea que define el entorno operativo para el código de programa.

50 La Fig. 1 muestra un sistema 100 ejemplar de comunicación inalámbrica, en el cual puede emplearse una realización de la revelación. Con fines de ilustración, la Fig. 1 muestra tres unidades remotas 120, 130 y 150, y dos estaciones base 140. Se reconocerá que los sistemas típicos de comunicación inalámbrica pueden tener unidades remotas y estaciones base. Las unidades remotas 120, 130 y 150 incluyen componentes de hardware, componentes de software, o ambos, según lo representado por los componentes 125A, 125C y 125B, respectivamente, que han sido adaptados para realizar la revelación, según se expone adicionalmente más adelante. La Fig. 1 muestra señales 180 del enlace directo desde las estaciones base 140 a las unidades remotas 120, 130 y 150, y señales 190 del enlace inverso desde las unidades remotas 120, 130 y 150 a las estaciones base 140.

60 En la Fig. 1, la unidad remota 102 se muestra como un teléfono móvil, la unidad remota 130 se muestra como un ordenador portátil y la unidad remota 150 se muestra como una unidad remota de ubicación fija en un sistema de bucle local inalámbrico. Por ejemplo, las unidades remotas pueden ser teléfonos celulares, unidades de sistemas de mano de comunicación personal (PCS), unidades de datos portátiles tales como agendas electrónicas, o unidades de datos de ubicación fija tales como equipos de lectura de contadores. Aunque la Fig. 1 ilustra las unidades remotas según las enseñanzas de la revelación, la revelación no se limita a estas unidades ejemplares ilustradas. La revelación puede ser empleada de manera adecuada en cualquier dispositivo con un procesador con una memoria caché para instrucciones, una memoria caché para datos y una memoria de sistema.

65 La Fig. 2 es un diagrama en bloques funcionales de un complejo 200 de procesador y memoria, en el cual el funcionamiento normal de la memoria caché para datos está adaptado para una extracción más eficiente de instrucciones, según se describe adicionalmente en el presente documento. El complejo 200 de procesador y memoria incluye un

ES 2 345 733 T3

procesador 202, una memoria caché 204 de instrucciones de nivel 1 (L1), una unidad 206 de control de memoria caché de instrucciones L1, una memoria caché 208 de datos L1, una unidad 210 de control de memoria caché de datos L1, una sección 211 de control y una memoria 212 de sistema. La unidad 206 de control de memoria caché de instrucciones L1 puede incluir una memoria direccionable por contenido de instrucciones, para la asociación de etiquetas de instrucciones, como la que puede usarse en una memoria caché asociativa fijada. La sección 211 de control incluye los elementos 220, 226 y 234 de multiplexado, los dispositivos 232 y 238 de compuerta, y un invertidor 240. Los dispositivos periféricos que pueden conectarse con el complejo procesador no se muestran para mayor claridad de la exposición de la presente revelación. El complejo 200 de procesador y memoria puede emplearse de manera adecuada en los componentes 125A a 125C para ejecutar código de programa que está almacenado en la memoria 212 del sistema.

A fin de extraer una instrucción en el complejo 200 de procesador y memoria, el procesador 202 genera una dirección 214 de extracción de instrucción (DI) de la instrucción deseada y envía la dirección de extracción de la instrucción a la unidad 206 de control de memoria caché de instrucciones L1. La unidad 206 de control de memoria caché de instrucciones L1 comprueba si la instrucción está presente en la memoria caché 204 de instrucciones L1. Esta comprobación se lleva a cabo, por ejemplo, mediante el uso de una memoria direccionable por contenido interno (CAM) en una búsqueda asociativa de una coincidencia con la dirección de extracción de instrucción proporcionada. Cuando la instrucción está presente, ocurre una coincidencia y la unidad 206 de control de memoria caché de instrucciones L1 indica que la instrucción está presente en la memoria caché 204 de instrucciones. Si la instrucción no está presente, no se hallará ninguna coincidencia en la búsqueda asociativa CAM y la unidad 206 de control de memoria caché de instrucciones L1 indica que la instrucción no está presente en la memoria caché 204 de instrucciones.

Si la instrucción está presente, se selecciona la instrucción en la dirección de extracción de instrucción en la memoria caché 204 de instrucciones. La instrucción se envía luego por el bus 216 de salida de instrucciones, a través del elemento multiplexador 226, al procesador 202.

Si la instrucción no está presente en la memoria caché de instrucciones, se activa una señal 218 de fallo de memoria caché de instrucciones ($ISM=1$), lo que indica que ha ocurrido un fallo. Al detectar un fallo en la memoria caché de instrucciones, el complejo 200 de procesador y memoria intenta extraer la instrucción deseada de la memoria caché 208 de datos L1. Con este fin, el elemento 220 multiplexador es habilitado por la señal 218 de fallo ($ISM=1$) para seleccionar la dirección 214 de extracción de instrucción. La dirección 214 de extracción de instrucción atraviesa luego un elemento 220 multiplexador, sobre un bus 222 de direcciones D y se envía a la unidad 210 de control de memoria caché de datos L1 como una dirección de extracción de datos. Se hace notar que el complejo 200 de procesador y memoria representa una vista lógica del sistema, ya que, por ejemplo, la aplicación de la dirección 214 de extracción de instrucción sobre el bus 222 de direcciones D puede requerir un arbitraje o un periodo de espera antes de que pueda obtenerse el acceso al bus 222 de direcciones D. El enfoque adoptado para multiplexar la dirección 214 de extracción de instrucción con la dirección 223 de datos generada por el procesador puede variar, y depende del enfoque específico adoptado en los diseños de la memoria caché de instrucciones y la memoria caché de datos.

La unidad 210 de control de memoria caché de datos L1 comprueba si hay un acierto en la memoria caché 208 de datos L1 en la dirección de extracción de instrucción proporcionada, mediante una búsqueda asociativa interna, por ejemplo, de la dirección de extracción de instrucción proporcionada. Un acierto indica que hay datos presentes en la dirección de extracción de instrucción proporcionada. Estos datos son, en verdad, una instrucción, y la entrada de la memoria caché de datos se extrae desde la memoria caché 208 de datos L1 y se coloca en el bus 224 de salida de datos. A fin de proporcionar los datos extraídos de la memoria caché 208 de datos L1 como una instrucción para el procesador, puede emplearse de manera adecuada un elemento 226 multiplexador. El bus 224 de salida de datos es seleccionado por el elemento 226 multiplexador colocando los datos extraídos de la memoria caché de datos en el bus 228 de instrucciones del procesador, cuando hay un fallo en la memoria caché de instrucciones seguido por un acierto en la memoria caché de datos en la dirección de extracción de instrucción.

La ocurrencia del fallo en la memoria caché de instrucciones, indicada por estar activada en alto la señal 218 de fallo ($ISM=1$), seguida por el acierto en la memoria caché de datos en la misma dirección de extracción de instrucción, indicado por estar activada en alto la señal 230 de acierto (DH=1$), está lógicamente representada por la compuerta AND 232. La salida de la compuerta AND 232 es la señal 233 de selección para el elemento 226 multiplexador. La instrucción hallada en la memoria caché de datos también es multiplexada, para cargarla en la memoria caché 204 de instrucciones, por el elemento 234 multiplexador, usando la señal 233 de selección proporcionada lógicamente por la compuerta AND 232. Mientras el bus 224 de salida de datos está remitiendo la instrucción al procesador, la entrada 236 de lectura de datos del procesador es desactivada por la compuerta AND 238, usando el invertidor 240 para proporcionar una inversa de la señal 233 de selección.

Si se determinó que hubo un fallo en la memoria caché de datos en la dirección de extracción de instrucción proporcionada, la instrucción no está en la memoria caché de datos, y la instrucción se extrae desde la memoria 212 del sistema. La señal 230 de acierto (DH=1$) también se envía a la unidad 206 de control de memoria caché de instrucciones L1 para indicar, por su estado inactivo, que ocurrió un fallo en el intento de localizar la instrucción en la memoria caché 208 de datos. Obsérvese que pueden usarse otros medios de señalización para indicar que ocurrió un fallo en el intento de localizar la instrucción en la memoria caché 208 de datos. Como la instrucción no está en la memoria caché 204 de instrucciones, y no está en la memoria caché 208 de datos, debe extraerse de la memoria 212 del sistema.

ES 2 345 733 T3

Una vez que se obtiene la instrucción de la memoria 212 del sistema, se envía al procesador 202. Obsérvese que las trayectorias desde la memoria del sistema, para suministrar una instrucción debido a un fallo en la memoria caché de instrucciones o la memoria caché de datos, y para suministrar datos debido a un fallo en la memoria caché de datos, no se muestran, a fin de ilustrar con claridad la presente revelación.

5

La Fig. 3 es un diagrama ejemplar de datos de un procedimiento 300 para extraer directamente una instrucción en una memoria caché de datos después de un fallo en la memoria caché de instrucciones, a fin de minimizar el sobregasto usualmente asociado a la gestión de la instrucción inicialmente almacenada como datos en la memoria caché de datos. Las relaciones ejemplares entre las etapas de la Fig. 3 y los elementos de la Fig. 2 se indican describiendo cómo los elementos del complejo 200 de procesador y memoria pueden cooperar de manera adecuada para llevar a cabo las etapas del procedimiento 300.

10

A fin de extraer una instrucción, se genera una dirección de extracción de instrucción en la etapa 304. Por ejemplo, un procesador, tal como el procesador 202, genera una dirección de extracción de instrucción de la instrucción deseada, y envía la dirección 214 de extracción de instrucción al controlador 206 de memoria caché de instrucciones L1. En la etapa 308, se determina si hay un acierto o un fallo en la memoria caché de instrucciones. Por ejemplo, el controlador 206 de memoria caché de instrucciones L1 comprueba si la instrucción está presente en la memoria caché 204 de instrucciones. Si la instrucción está presente, su presencia se indica como un acierto. Si la instrucción está presente, el procedimiento 300 avanza a la etapa 312 y se selecciona la instrucción en la dirección de extracción de la instrucción. En la etapa 316, la instrucción se envía al procesador. Por ejemplo, la instrucción seleccionada se coloca en el bus 216 de salida de instrucciones y se envía al procesador 202 a través del elemento 226 multiplexador.

15

20

Si la instrucción no está presente en la memoria caché de instrucciones, según lo determinado en la etapa 308, se da una indicación de que ha ocurrido un fallo y se hace un intento de extraer la instrucción de la memoria caché de datos en la etapa 320. Por ejemplo, la dirección 214 de extracción de instrucción se envía, a través del elemento 220 multiplexador, como una dirección 222 de extracción de datos, a la memoria caché 208 de datos. En la etapa 324 se hace una comprobación, por ejemplo, por parte del controlador 210 de memoria caché de datos L1, en cuanto a si hay datos válidos presentes en la dirección de extracción de instrucción proporcionada. Si hay datos válidos presentes en la memoria caché de datos en la dirección de extracción de instrucción proporcionada, los datos son, de hecho, una instrucción, y la entrada de la memoria caché de datos se extrae en la etapa 328. En la etapa 316, los datos extraídos de la memoria caché de datos se envían como una instrucción al procesador. Por ejemplo, los datos extraídos por el bus 224 de salida de datos desde la memoria caché 208 de datos se envían a través del elemento 226 multiplexador y se suministran como una instrucción al procesador 202 por el bus 228 de instrucciones.

25

30

En la etapa 324, si hubo un fallo en la memoria caché de datos en la dirección de extracción de instrucción proporcionada, la instrucción no está en la memoria caché de datos y, en la etapa 332, la instrucción se extrae de la memoria del sistema. Por ejemplo, la señal 230 D\$H=1 de acierto en memoria caché de datos se envía a la unidad 206 de control de memoria caché de instrucciones L1 para indicar, por su estado inactivo, que ocurrió un fallo en el intento de localizar la instrucción en la memoria caché 208 de datos. Como la instrucción no está en la memoria caché 204 de instrucciones y no está en la memoria caché 208 de datos, debe extraerse de la memoria 212 del sistema. Una vez que se ha obtenido la instrucción de la memoria 212 del sistema, la instrucción se envía al procesador 202, según lo indicado en la etapa 316.

35

40

La Fig. 4 es un diagrama en bloques funcionales de un complejo 400 de procesador y memoria que incluye una tabla de páginas de instrucciones, en el cual el funcionamiento normal de la memoria caché de datos está adaptado para un funcionamiento eficiente de la extracción de instrucciones, según la presente revelación. El complejo 400 de procesador y memoria incluye un procesador 402, una memoria caché 404 de instrucciones de nivel 1 (L1), una unidad de gestión de memoria de instrucciones (IMMU) y control 406 de memoria caché (IMMU/\$Control), una memoria caché 408 de datos L1, una unidad de gestión de memoria de datos (DMMU) y control 410 de memoria caché (DMMU/\$Control), una sección 411 de control y una jerarquía 412 de memorias. El IMMU/\$Control 406 puede incluir, por ejemplo, un proceso de traducción de direcciones de instrucciones de virtuales a físicas. La sección 411 de control incluye los elementos 423, 438 y 448 de multiplexado, los dispositivos 428, 444 y 452 de compuerta, y un invertidor 454. Los dispositivos periféricos que pueden conectarse con el complejo procesador no se muestran, para mayor claridad de la exposición de la presente revelación. El complejo 400 de procesador y memoria pueden emplearse de manera adecuada en los componentes 125A a 125C, para ejecutar código de programa que está almacenado en la memoria 412 del sistema.

45

50

55

La memoria caché de instrucciones puede usar un almacén temporal de búsqueda colateral de traducciones (TLB) que contiene una tabla de páginas de instrucciones, a fin de mejorar las prestaciones de la memoria caché de instrucciones. La tabla de páginas de instrucciones tiene, por ejemplo, una lista de números de páginas físicas asociadas a números de páginas virtuales, e información adicional asociada a cada entrada de número de página. Una entrada de la tabla de páginas de instrucciones se crea cuando una página de memoria en la gama de direcciones de instrucciones se carga en la memoria caché de instrucciones o la memoria caché de datos. La carga de una página de memoria puede ocurrir bajo la supervisión de un sistema operativo (OS). En funcionamiento, la tabla de páginas de instrucciones se examina en búsqueda de una coincidencia con un número de página virtual proporcionado al TLB. Si bien se describe en el presente documento un TLB con una tabla de páginas de instrucciones como parte de la MMU de instrucciones y control 406 de memoria caché, se reconocerá que pueden usarse enfoques alternativos.

60

65

ES 2 345 733 T3

A fin de extraer una instrucción en el complejo 400 de procesador y memoria, el procesador 402 genera una dirección 414 de extracción de instrucción (DI) para la instrucción deseada y envía la dirección de extracción de instrucción al IMMU/\$Control 406. Se selecciona una entrada adecuada en una tabla de páginas de instrucciones, tal como la tabla 416 de páginas situada en el IMMU/\$Control 406, sobre la base de un número de página proporcionado que es parte de la DI 414. La dirección de instrucción basada en la entrada seleccionada de la tabla de páginas se combina con una dirección de página, también parte de la DI 414, generando una dirección 418 de instrucción (DG) que se aplica internamente a la memoria caché 404 de instrucciones L1. La entrada seleccionada de la tabla 416 de páginas incluye información adicional almacenada con esa entrada. Uno de los bits adicionales de información que pueden almacenarse con cada entrada de la tabla de páginas es un atributo de comprobación de memoria caché de datos, etiquetado como el bit D 420.

El bit D se fija en "1" cuando se crea la entrada en la tabla de páginas de instrucciones debido a la carga de una página de instrucciones en la memoria caché de datos, o cuando se generan instrucciones que se almacenan en una página en la memoria caché de datos durante el procesamiento. El bit D es habitualmente fijado por el sistema operativo (OS) para indicar que el contenido de una página puede usarse tanto para datos como para instrucciones. En un escenario ejemplar, un programa, que genera datos que serán usados como instrucciones, llama al OS para solicitar que se marquen las páginas adecuadas fijando el bit D en las entradas asociadas de la tabla de páginas. En otro escenario, un programa también puede solicitar páginas al OS que ya estén configuradas con el bit D activado. El bit D no necesariamente necesita ser desactivado explícitamente. Si un programa especifica que la memoria caché de datos puede contener instrucciones, causando que se active(n) el bit D, o los bits D, adecuado(s), entonces esa especificación puede ser válida a lo largo de la vida del programa. El bit D, o los bits D, puede(n) desactivarse luego cuando la tabla de páginas se usa para un proceso distinto.

El IMMU/\$Control 406 comprueba si la instrucción está presente en la memoria caché 404 de instrucciones. Si la instrucción está presente, esta presencia se indica como un acierto. Si la instrucción está presente, se selecciona la instrucción en la dirección de extracción de instrucción de la memoria caché 404 de instrucciones. La instrucción se envía luego por el bus 422 de salida de instrucciones, a través del elemento 438 multiplexador, al procesador 402. Si la instrucción no está presente, se da una indicación, por parte del IMMU/\$Control 406, de que ha ocurrido un fallo, y se activa una señal 424 de fallo de memoria caché de instrucciones (ISM=1), indicando que ha ocurrido un fallo.

Al detectar un fallo en la memoria caché de instrucciones, conjuntamente con el bit D seleccionado activado en "1", el complejo 400 de procesador y memoria intenta extraer la instrucción deseada de la memoria caché 408 de datos L1. Este intento puede lograrse de manera adecuada, por ejemplo, usando el bit D seleccionado en una función de compuerta. El bit D 420 de la entrada seleccionada de la tabla de páginas se envía como la señal 426 del bit D. La señal 426 del bit D, por ejemplo, es sometido a la operación lógica AND por la compuerta AND 428, con la indicación 424 de fallo (ISM=1). La salida 430 de la compuerta AND 428 es usada luego por el elemento 432 multiplexador para seleccionar la dirección 418 de instrucción generada (DG), o una dirección 433 de datos del procesador 402. Cuando se selecciona, la DG 418 atraviesa el elemento 432 multiplexador sobre el bus 434 de direcciones D (DD), y se envía a la MMU de datos y control 410 de memoria caché para determinar si la instrucción reside en la memoria caché 408 de datos en la dirección de extracción de datos. Se observa que el complejo 400 de procesador y memoria representa una vista lógica del sistema, ya que, por ejemplo, la aplicación de la dirección 418 de instrucción generada sobre el bus 434 de direcciones D puede requerir un arbitraje o un periodo de espera antes de que pueda obtenerse el acceso al bus 434 de direcciones D. El enfoque adoptado para multiplexar la dirección 418 de instrucción generada con la dirección 433 de datos generada por el procesador puede variar, y depende del enfoque específico adoptado en los diseños de la memoria caché de instrucciones y la memoria caché de datos.

La memoria caché de datos comprueba luego si hay datos válidos presentes en la dirección de extracción de instrucción proporcionada. Si hay datos válidos presentes en la dirección de extracción de instrucción proporcionada, los datos, de hecho, son una instrucción, y la entrada de la memoria caché de datos se extrae de la memoria caché 408 de datos L1, y se coloca en el bus 436 de salida de datos. A fin de proporcionar la entrada de memoria caché de datos como una instrucción al procesador, se usa un elemento 438 multiplexador, por ejemplo. El elemento 438 multiplexador es habilitado para pasar del bus 436 de salida de datos al bus 440 de instrucciones del procesador cuando hay un fallo en la memoria caché de instrucciones y el bit D seleccionado está fijado en "1", seguido por un acierto en la memoria caché de datos en la dirección de extracción de instrucción. La ocurrencia del fallo en la memoria caché de instrucciones, indicada por estar activada en alto la señal 424 de fallo (ISM=1), y fijada en "1" la señal 426 del bit D, seguida por el acierto en la memoria caché de datos en la dirección de instrucción generada, indicado por estar activada en alto la señal 442 de acierto (DSH=1), está lógicamente representada por la compuerta AND 444. La salida de la compuerta AND 444 es la señal 446 de selección para el elemento 438 multiplexador. La instrucción en el bus de salida de datos también está multiplexada, para la carga en la memoria caché de instrucciones, por el elemento 448 multiplexador, usando la señal 446 de selección. Mientras el bus 436 de salida de datos de la memoria caché de datos L1 está remitiendo la instrucción al procesador 402, el bus 436 de salida de datos está bloqueado, para las transferencias a la entrada 450 de lectura de datos del procesador, por la compuerta AND 452, usando una inversa de la señal 446 de selección proporcionada por el invertidor 454.

Si se determinó que hubo un fallo en la memoria caché de datos en la dirección de extracción de instrucción proporcionada, la instrucción no está en la memoria caché de datos, y la instrucción se extrae de la memoria 412 del sistema. La señal 442 de acierto (DSH=1) también se envía al IMMU/\$Control 406 para indicar, por su estado inactivo, que ocurrió un fallo en el intento de localizar la instrucción en la memoria caché 408 de datos. Una vez

ES 2 345 733 T3

que se ha obtenido la instrucción de la memoria 412 del sistema, se envía al procesador 402. Obsérvese que las trayectorias desde la jerarquía de memorias para proporcionar una instrucción, debido a un fallo en la memoria caché de instrucciones o la memoria caché de datos, y para suministrar datos debido a un fallo en la memoria caché de datos, no se muestran, pero puede emplearse cualquiera entre una amplia variedad de enfoques de conexión, coherentes con la aplicación y el procesador empleados.

La Fig. 5 es un diagrama ejemplar de flujo de un procedimiento 500 para extraer una instrucción en una memoria caché de datos después de un fallo en la memoria caché de instrucciones, y un atributo de comprobación de la memoria caché de datos indica que la memoria caché de datos debería comprobarse para la instrucción. Las relaciones ejemplares entre las etapas de la Fig. 5 y los elementos de la Fig. 4 se indican haciendo referencia a elementos ejemplares del complejo 400 de procesador y memoria, que pueden emplearse de manera adecuada para llevar a cabo las etapas del procedimiento 500 de la Fig. 5.

A fin de extraer una instrucción, una dirección de extracción de instrucción para la instrucción deseada se genera en la etapa 502. Por ejemplo, un procesador, tal como el procesador 402, genera una dirección de extracción de instrucción y envía la dirección 414 de extracción de instrucción al controlador 406 de memoria caché de instrucciones L1. La dirección de extracción de instrucción puede ser una dirección virtual compuesta por un número 504 de página y una dirección 506 de página. En la etapa 508, se selecciona una entrada adecuada en una tabla de páginas de instrucciones, tal como la tabla 416 de páginas de instrucciones, sobre la base del número 504 de página proporcionado. La dirección generada, basada en la entrada seleccionada de la tabla de páginas, se combina en la etapa 509 con la dirección 506 de página para producir una dirección de memoria caché de instrucciones.

La entrada seleccionada de la tabla 416 de páginas de instrucciones incluye la información adicional almacenada con esa entrada. Uno de los bits adicionales de información que puede almacenarse con cada entrada de la tabla de páginas es un atributo de comprobación de memoria caché de datos, tal como el bit etiquetado como el bit D 420. Este atributo se selecciona en la etapa 510.

En la etapa 512, se determina si hay un acierto o un fallo en la memoria caché de instrucciones. Por ejemplo, la memoria caché de instrucciones comprueba si la instrucción está presente. Si la instrucción está presente, su presencia se indica como un acierto. Si la instrucción está presente, el procedimiento 500 avanza a la etapa 514 y se selecciona la instrucción en la dirección de extracción de instrucción. En la etapa 516, la instrucción se envía al procesador. Por ejemplo, la instrucción seleccionada se coloca en el bus 422 de salida de instrucciones y se envía a través del elemento 438 multiplexador al bus 440 de instrucciones del procesador 402.

Si la instrucción no está presente en la memoria caché de instrucciones, según lo determinado en la etapa 512, se da una indicación de que ha ocurrido un fallo, y el procedimiento 500 avanza a la etapa 518. En la etapa 518, se comprueba si el bit D que fue seleccionado en la etapa 510 está fijado en "1", lo que indica que la memoria caché de datos debería comprobarse en busca de la instrucción. Si el bit D fue fijado en "1", el procesador intenta extraer la instrucción de la memoria caché de datos en la etapa 520. Por ejemplo, la dirección generada 418 de extracción de instrucción se envía como una dirección 434 de extracción de datos a la memoria caché de datos.

En la etapa 524, la memoria caché de datos comprueba si hay datos válidos presentes en la dirección de extracción de instrucción proporcionada. Si hay datos válidos presentes en la dirección de extracción de instrucción proporcionada, los datos, de hecho, son una instrucción, y la entrada de la memoria caché de datos se extrae en la etapa 528. En la etapa 516, los datos extraídos de la memoria caché de datos se envían como una instrucción al procesador. Por ejemplo, los datos extraídos por el bus 436 de salida de datos se envían a través del elemento 438 multiplexador, y se suministran como una instrucción al procesador 402 por el bus 440 de instrucciones.

Volviendo a la etapa 518, si se determina en la etapa 518 que el bit D era un "0", se sabe que la instrucción no está presente en la memoria caché de datos, y el procedimiento 500 avanza a la etapa 522. A la etapa 522 también se llega en la situación donde hubo un fallo en la memoria caché de datos en la dirección de extracción de instrucción proporcionada, según lo determinado en la etapa 524. En cualquier caso, se sabe que la instrucción no está presente en la memoria caché de instrucciones ni en la memoria caché de datos, y la instrucción se extrae de la memoria del sistema, según lo indicado en la etapa 522. Por ejemplo, se accederá a la memoria 412 del sistema para la instrucción. Una vez que se obtiene la instrucción de la memoria 412 del sistema, la instrucción se envía al procesador 402, según se indica en la etapa 516.

La Fig. 6 es un diagrama de flujo ejemplar de un procedimiento 600 para ejecutar código de programa que se genera como datos y se almacena en una memoria caché de datos. El código de programa que obedece a este procedimiento puede ejecutarse en un complejo de procesador y memoria con una memoria caché de instrucciones, una memoria caché de datos y una memoria del sistema, tales como las expuestas con respecto a las Figs. 2 y 4, y puede emplearse de manera adecuada en los componentes 125A a 125C de la Fig. 1.

En la etapa 602, un programa genera código. Tal generación puede ocurrir, por ejemplo, cuando un programa genera código ejecutable a partir de un programa comprimido. El código generado se trata inicialmente como datos y se almacena en una memoria caché de datos después de ser generado. Antes de ejecutar el programa, se invalida una memoria caché de instrucciones en la etapa 604. La etapa de invalidación garantiza que no hay ninguna instrucción en la misma dirección que el código generado. En la etapa 606, el código generado es ejecutado por el procesador

ES 2 345 733 T3

extrayendo instrucciones del espacio de direcciones del programa en la memoria caché de instrucciones, y puede incluir instrucciones que se almacenan en la memoria caché de datos. Para aquellas instrucciones almacenadas en la memoria caché de datos, se siguen las técnicas de la presente revelación, lo que permite comprobar la memoria caché de datos en busca de instrucciones en caso de ocurrencia de un fallo en la memoria caché de instrucciones. Al hallar una instrucción en la memoria caché de datos, la instrucción se extrae directamente de la memoria caché de datos para su ejecución en el procesador.

Si bien la presente revelación se ha revelado en un contexto actualmente preferido, se reconocerá que las presentes enseñanzas pueden adaptarse a una gran variedad de contextos coherentes con esta revelación y las reivindicaciones siguientes.

15

20

25

30

35

40

45

50

55

60

65

ES 2 345 733 T3

REIVINDICACIONES

5 1. Un procedimiento (500) para hallar una instrucción en una memoria caché de datos que es distinta a una memoria caché de instrucciones, comprendiendo el procedimiento:

determinar (512) que un intento de extracción de la instrucción falló en la memoria caché de instrucciones, en una dirección de extracción de instrucción; **caracterizado** por:

10 determinar (518) que un atributo de comprobación de memoria caché de datos ha sido fijado en un estado activo en una entrada de una tabla de páginas asociada a la dirección de extracción de la instrucción;

15 seleccionar (520) la dirección de extracción de la instrucción como una dirección de extracción de datos en respuesta a la determinación de que falló un intento de extracción y de que el atributo de comprobación de memoria caché de datos está en el estado activo; y

hacer un intento de extracción en la memoria caché de datos para la instrucción, en la dirección seleccionada de extracción de datos.

20 2. El procedimiento de la reivindicación 1, que comprende adicionalmente:

fijar como activa una señal de acierto si la instrucción se halló en la memoria caché de datos en respuesta al intento de extracción en la memoria caché de datos.

25 3. El procedimiento de la reivindicación 1, que comprende adicionalmente:

30 fijar como activo un atributo de comprobación de memoria caché de datos en la entrada asociada de la tabla de páginas, al generar instrucciones que se almacenan como datos en la memoria caché de datos.

4. El procedimiento de la reivindicación 3, en el cual el atributo de comprobación de memoria caché de datos es un bit almacenado en una tabla de páginas de instrucciones.

35 5. El procedimiento de la reivindicación 1, que comprende adicionalmente:

40 generar, por parte de un programa, datos que se usarán como instrucciones; y solicitar, por parte del programa, que un sistema operativo fije como activo el atributo de comprobación de memoria caché de datos en las entradas de la tabla de páginas asociadas a las instrucciones.

6. El procedimiento de la reivindicación 3, en el cual el atributo de comprobación de memoria caché de datos se desactiva más tarde cuando la entrada de la tabla de páginas se usa para un proceso distinto.

45 7. El procedimiento de la reivindicación 1, en el cual la etapa de seleccionar la dirección de extracción de instrucción comprende adicionalmente:

multiplexar la dirección de extracción de instrucción y una dirección de extracción de datos; y

50 seleccionar la dirección de extracción de instrucción para su aplicación a la memoria caché de datos como dirección transformada de extracción de datos, en el que la dirección de extracción de instrucción se selecciona después de determinar que el intento de extracción de instrucción erró en la memoria caché de instrucciones.

55 8. El procedimiento de la reivindicación 1, en el cual la etapa de hacer un intento de extracción en la memoria caché de datos comprende adicionalmente:

determinar el acierto de la instrucción en la memoria caché de datos; y

60 extraer la instrucción de la memoria caché de datos.

9. El procedimiento de la reivindicación 1, que comprende adicionalmente:

65 determinar que el intento de extracción falló en la memoria caché de datos; y

informar a un control de memoria de instrucciones de que el intento de extracción falló en la memoria caché de datos.

ES 2 345 733 T3

10. El procedimiento de la reivindicación 9, que comprende adicionalmente:

extraer la instrucción desde una memoria del sistema.

5

11. Una disposición de procesador que comprende:

una memoria caché (406) de instrucciones;

10

una memoria caché (408) de datos; **caracterizada** por:

una unidad (4064) de gestión de memoria de instrucciones, con una tabla (416) de páginas con entradas que tienen, cada una, un atributo (420) de comprobación de memoria caché de datos, asociado a una dirección de extracción de instrucción; y

15

un primer selector (432) para seleccionar una dirección de extracción de instrucción o una dirección de extracción de datos, sobre la base de una señal de selección, causando la señal de selección que se aplique la dirección de extracción de instrucción, o la dirección de extracción de datos, a la memoria caché (408) de datos, por lo que las instrucciones o los datos pueden extraerse selectivamente desde la memoria caché (408) de datos, en donde la señal de selección del primer selector (432) selecciona la dirección de extracción de instrucción si una señal de fallo de instrucción indica que ha ocurrido un fallo en una operación de extracción de instrucción en la memoria caché (406) de instrucciones, y está activo un atributo de comprobación de memoria caché de datos en una entrada asociada a la dirección de extracción de la instrucción.

25

12. La disposición procesadora de la reivindicación 11, que comprende adicionalmente:

un segundo selector (438) para seleccionar un bus de salida de instrucciones desde la memoria caché de instrucciones, o un bus de salida de datos desde la memoria caché de datos, a aplicar a una entrada del bus de instrucciones del procesador.

30

13. La disposición procesadora (400) de la reivindicación 12, en la cual el segundo selector (438) selecciona el bus de salida de datos desde la memoria caché de datos si ocurrió un fallo en la memoria caché (404) de instrucciones y ocurrió un acierto en la memoria caché (408) de datos, en la dirección de extracción de instrucción seleccionada a través del primer selector (432).

35

14. La disposición procesadora (400) de la reivindicación 12, en la cual el segundo selector (438) selecciona el bus de salida de instrucciones si ocurre un acierto en la memoria caché (404) de instrucciones.

40

15. Un procedimiento para ejecutar código de programa, que comprende:

generar instrucciones que se almacenan como datos en una memoria caché de datos;

45

invalidar una memoria caché de instrucciones antes de la ejecución del código de programa que usa las instrucciones generadas; **caracterizado** por:

establecer un atributo de comprobación de memoria caché de datos en una tabla de páginas de instrucciones, para indicar que una instrucción puede estar en la memoria caché de datos; y

50

extraer la instrucción de la memoria caché de datos si la instrucción no se halla en la memoria caché de instrucciones y el atributo de comprobación de memoria caché de datos está activado, por lo que el código de programa es ejecutado.

55

60

65

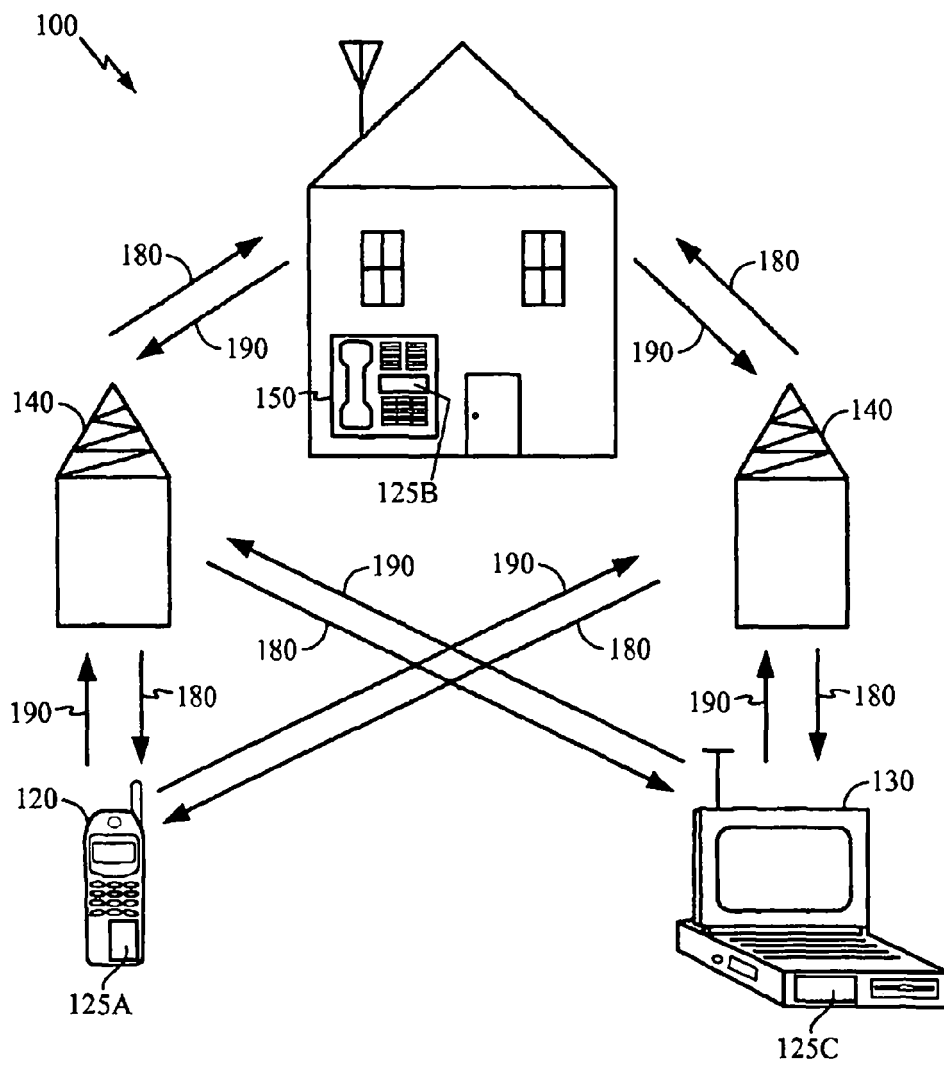


FIG. 1

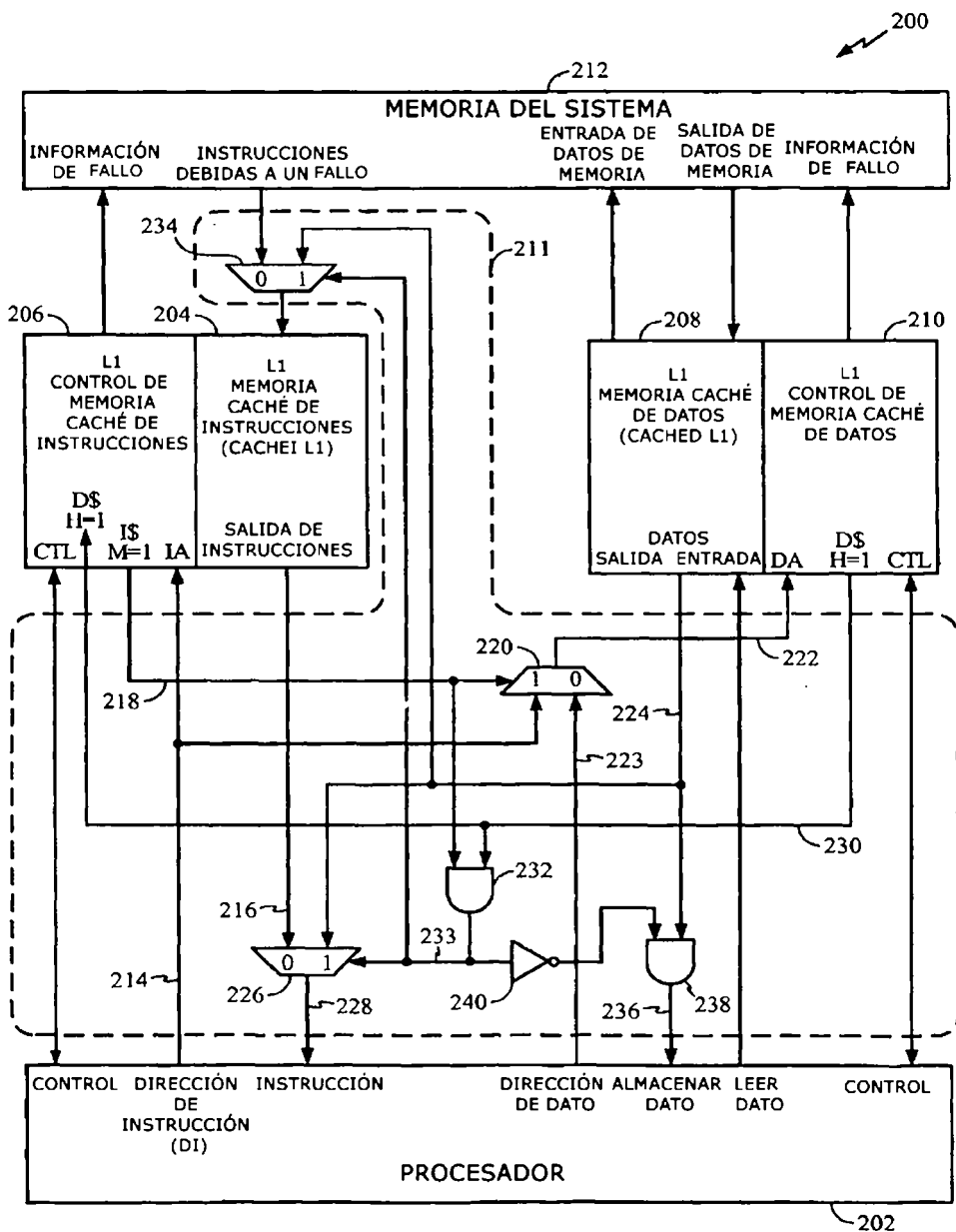


FIG. 2

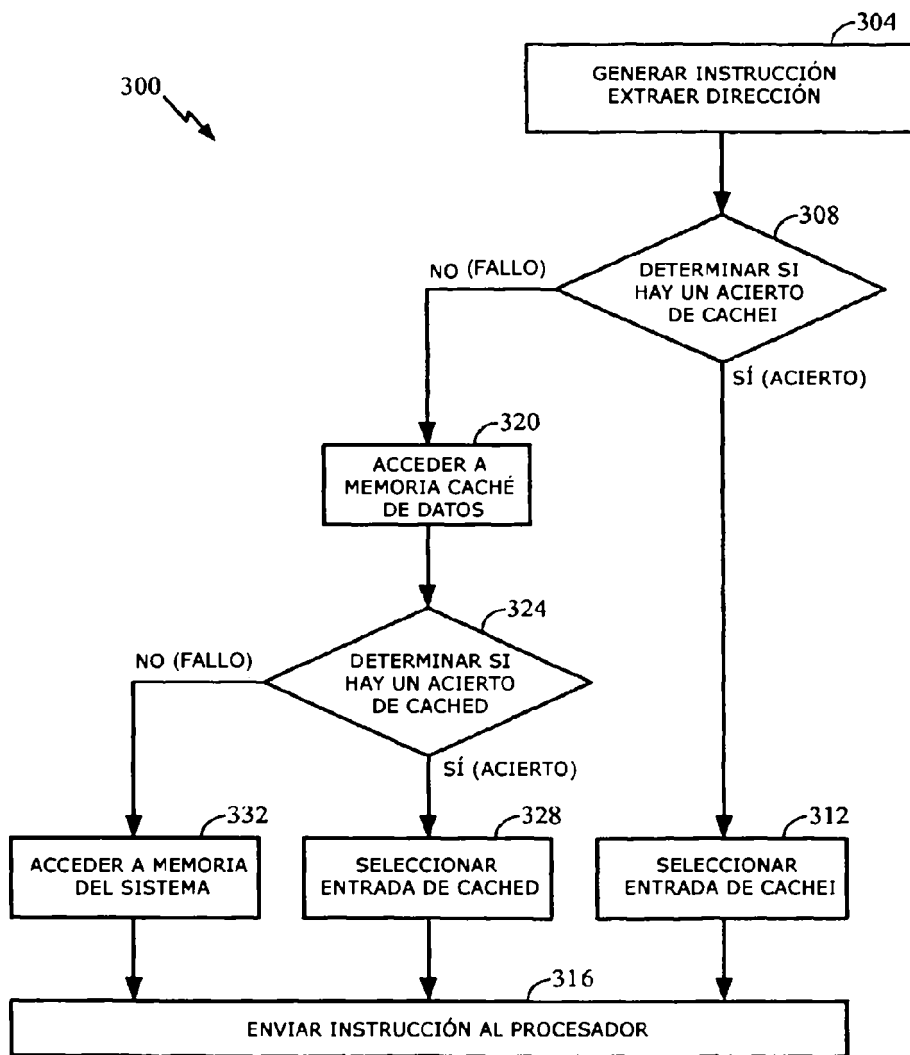


FIG. 3

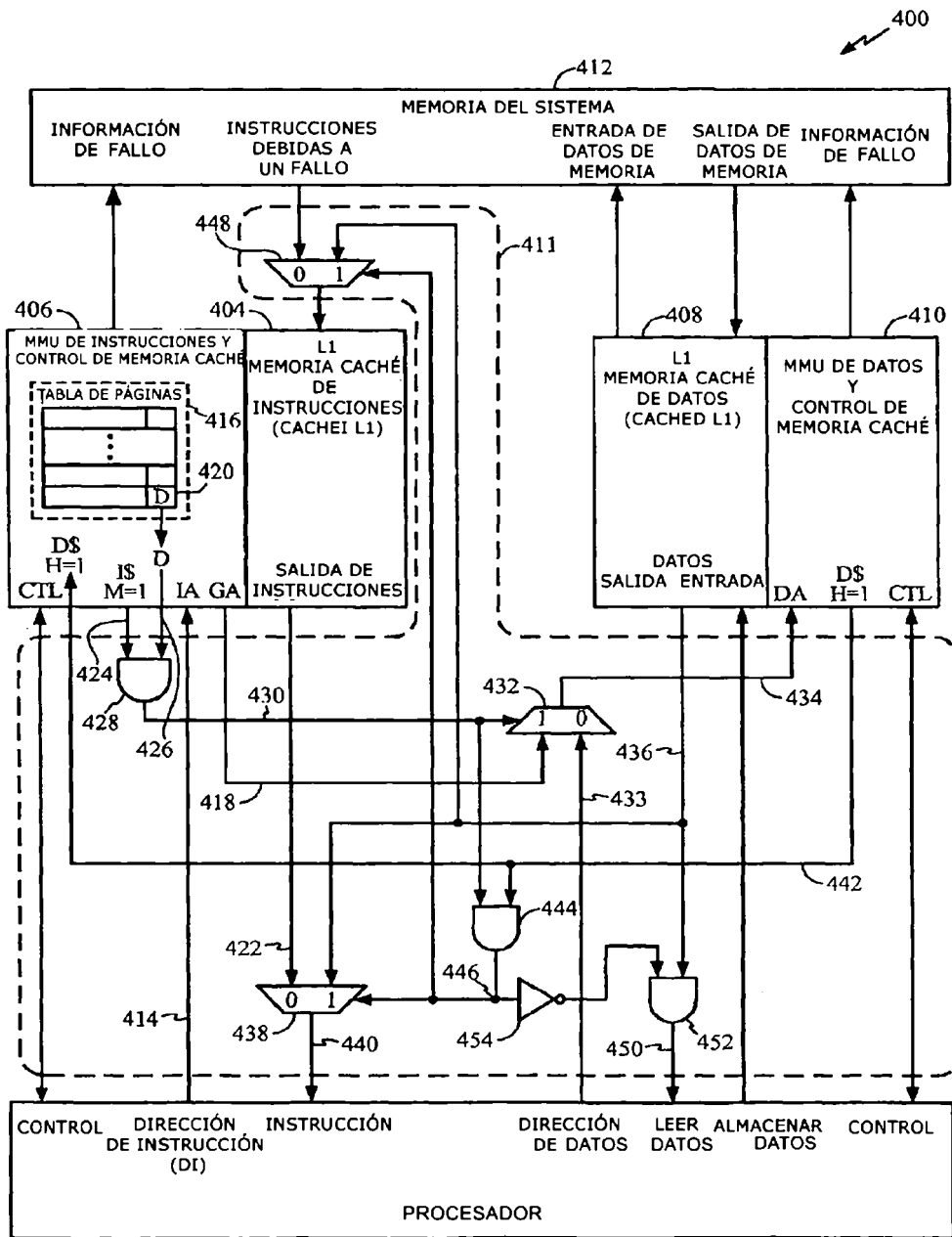


FIG. 4

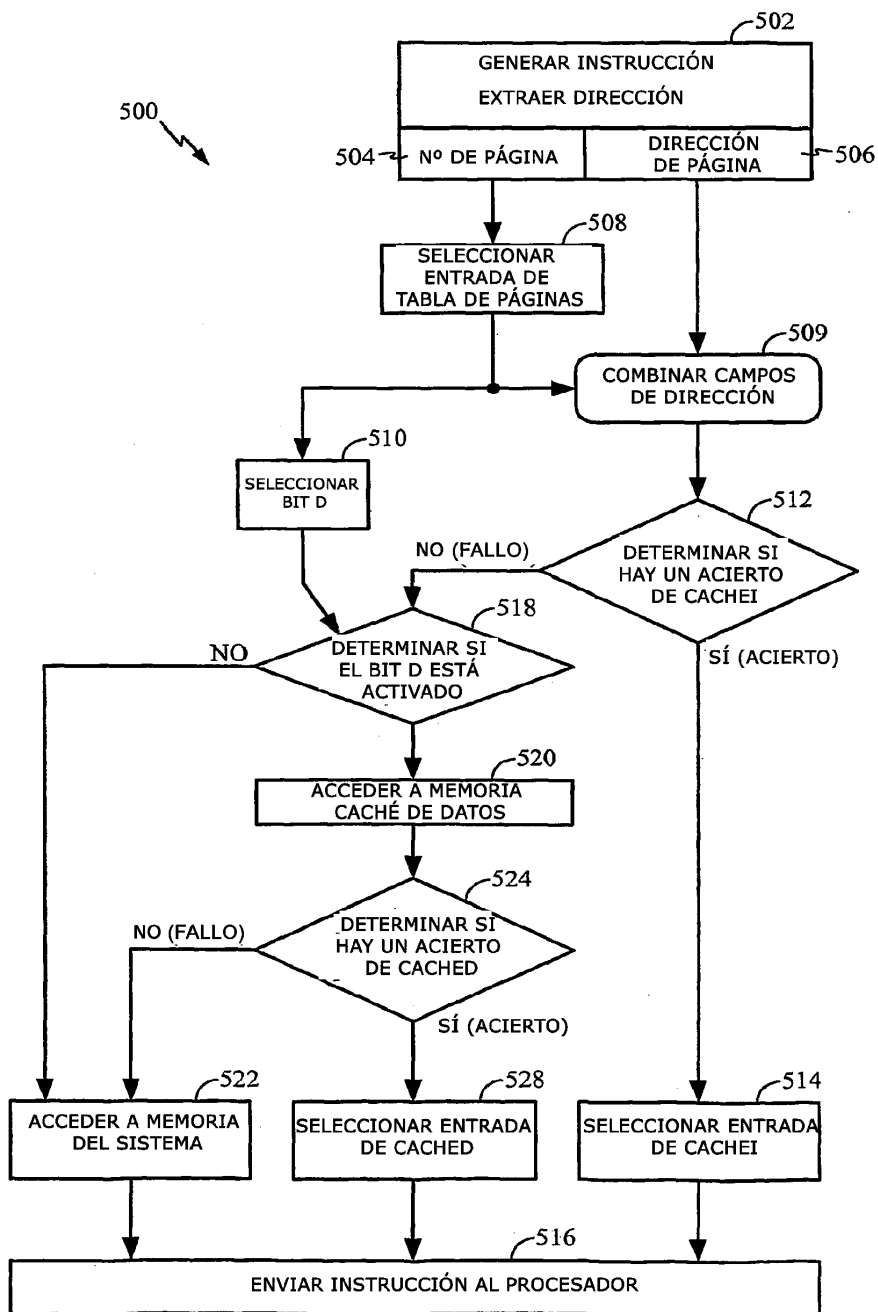


FIG. 5

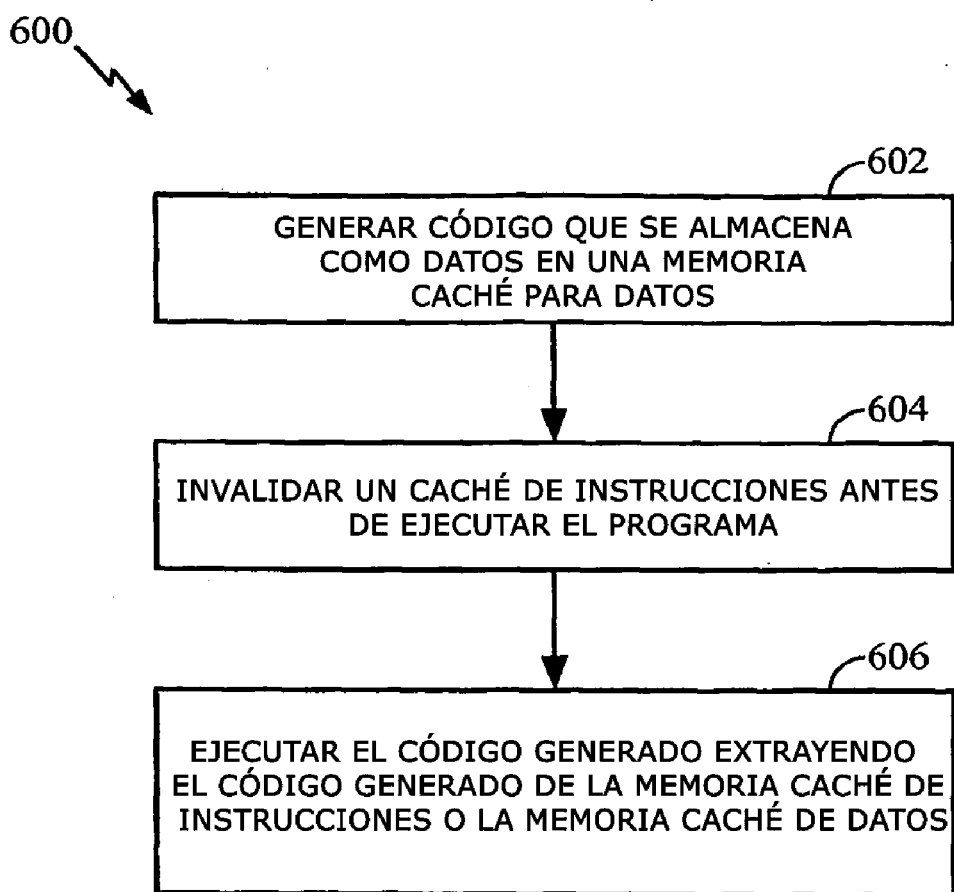


FIG. 6