

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 866 156**

51 Int. Cl.:

G06F 13/40 (2006.01)

G06F 13/42 (2006.01)

G06F 13/28 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **02.05.2013 E 18155911 (3)**

97 Fecha y número de publicación de la concesión europea: **27.01.2021 EP 3385854**

54 Título: **Sistema informático, método para acceder a un terminal de interconexión de componentes periféricos exprés, y equipo**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
19.10.2021

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

DU, GE

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 866 156 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

Sistema informático, método para acceder a un terminal de interconexión de componentes periféricos exprés, y equipo

Campo técnico

5 Los modos de realización de la presente invención están relacionados con el campo de las tecnologías de computación y, en particular, con un método de acceso a un terminal de interconexión de componentes periféricos exprés, un sistema informático y un equipo.

Antecedentes

10 Una Interconexión de Componentes Periféricos Exprés (Peripheral Component Interconnect Express, PCIe) es un bus (canal compartido de comunicación) de alto rendimiento de un sistema que se utiliza en una plataforma de cálculo y comunicación. El bus PCIe es ampliamente utilizado en el sistema de interconexión de una unidad central de procesamiento (central processing unit, CPU) y un dispositivo periférico, y funciona como un canal de servicio principal en un dispositivo de cálculo y almacenamiento. Puede haber múltiples tipos de dispositivos periféricos que se interconectan con una CPU a través de un bus PCIe, tales como un dispositivo tarjeta de interfaz de red o un disco de estado sólido (Solid State Disk, SSD). En este documento, tales dispositivos se denominan terminales PCIe.

20 El bus PCIe es ampliamente utilizado como interfaz de bus de un servidor o un sistema de almacenamiento. A consecuencia de una necesidad en línea de ampliación y mantenimiento de capacidad, esto es, una necesidad de conexión/desconexión en caliente, durante la operación normal de un sistema, es necesario que un terminal PCIe se instale o retire sin apagarlo. La conexión/desconexión actual en caliente de un PCIe sigue el siguiente proceso de operación: un operador inicia una solicitud de conexión/desconexión en caliente pulsando un botón; después de detectar un evento de conexión/desconexión en caliente, un controlador de conexión/desconexión en caliente notifica a todas las unidades que puedan acceder al terminal PCIe que dejen de acceder al terminal PCIe, y desinstala los recursos del terminal PCIe en el que es necesario realizar la conexión/desconexión en caliente; y, a continuación, el terminal PCIe se apaga y el operador extrae el terminal PCIe.

30 Con el fin de garantizar el funcionamiento normal del sistema, es necesario notificar previamente la conexión/desconexión en caliente actual de un terminal PCIe. Sin embargo, en los últimos años, el bus PCIe ha evolucionado progresivamente desde la interconexión interna del sistema hasta la interconexión entre sistemas, y la utilización de, por ejemplo, cables externos, ha aumentado. Los cables se desconectan fácilmente de forma anómala y, en consecuencia, se produce una situación en la que un terminal PCIe se desconecta de forma anómala sin notificación previa. Además, cada vez más se da la situación en la que un usuario conecta directamente un disco de estado sólido SSD a un sistema. El usuario puede insertar o extraer el disco SSD sin previo aviso debido a un hábito del usuario. En la situación mencionada en la que el terminal PCIe se desconecta de forma anómala y repentina, si la CPU ha iniciado las instrucciones de lectura y escritura para el terminal PCIe, las instrucciones en curso permanecerán en estado de espera para ser ejecutadas; y cuando las instrucciones para acceder al terminal PCIe desde la CPU se acumulan de forma continua hasta un cierto límite, la CPU considera que todo el sistema se encuentra en un estado anómalo y notifica un error de excepción de comprobación de equipo (Machine check exception, MCE). Como resultado, se produce un reinicio de todo el sistema.

Resumen

40 La invención se define mediante las reivindicaciones.

Breve descripción de los dibujos

45 Con el fin de ilustrar de forma más clara las soluciones técnicas contenidas en los modos de realización de la presente invención, a continuación se introducen brevemente los dibujos adjuntos requeridos para describir los modos de realización o la técnica anterior. Obviamente, los dibujos adjuntos en la siguiente descripción muestran solamente algunas realizaciones de la presente invención, y aún así una persona con un conocimiento normal de la técnica puede obtener otros dibujos a partir de estos dibujos adjuntos sin esfuerzos creativos.

La FIG. 1 es un diagrama de la composición de un sistema informático de acuerdo con un modo de realización de la presente invención;

50 la FIG. 2 es un diagrama de la composición de los módulos de programa incluidos en una memoria de acuerdo con un modo de realización de la presente invención;

la FIG. 3 es un diagrama de la composición de otro sistema informático de acuerdo con un modo de realización de la presente invención;

la FIG. 4 es un diagrama de la composición de otro sistema informático de acuerdo con un modo de realización de la presente invención;

la FIG. 5 es un diagrama de flujo de un método de acuerdo con un modo de realización de la presente invención;

la FIG. 6 es un diagrama de flujo de otro método de acuerdo con un modo de realización de la presente invención;

la FIG. 7 es un diagrama de flujo de otro método de acuerdo con un modo de realización de la presente invención;

la FIG. 8 es un diagrama de flujo de otro método de acuerdo con un modo de realización de la presente invención;

5 la FIG. 9 es un diagrama de flujo de otro método de acuerdo con un modo de realización de la presente invención;

la FIG. 10 es un diagrama de flujo de otro método de acuerdo con un modo de realización de la presente invención;

la FIG. 11 es un diagrama de la composición de un equipo para acceder a un terminal PCIe de acuerdo con un modo de realización de la presente invención; y

10 la FIG. 12 es un diagrama de la composición de un ordenador de acuerdo con un modo de realización de la presente invención.

Descripción de los modos de realización

15 Los modos de realización de la presente invención proporcionan un método para acceder a un terminal de interconexión de componentes periféricos exprés, un sistema informático y un equipo. Cuando es necesario realizar una operación de conexión/desconexión en caliente en un terminal PCIe, es posible que no se le notifique al sistema con antelación el procesamiento y la conexión entre el terminal PCIe y el procesador se desconecte directamente. En ese instante no existe en el procesador un riesgo de reinicio por MCE. En los modos de realización de la presente invención, una situación en la que el terminal PCIe se extrae directamente del sistema o se encuentra desconectado debido a un error se denomina globalmente desconexión anómala de la central del terminal PCIe.

Arquitectura del sistema en un modo de realización de la presente invención

20 La FIG. 1 es un diagrama de la composición de un sistema informático de acuerdo con un modo de realización de la presente invención. El sistema informático que se ilustra en la FIG. 1 incluye una CPU 110, una memoria 120 y un terminal PCIe 130. El terminal PCIe 130 está conectado a la CPU 110 a través de un bus PCIe 140 y se puede insertar en o extraer del sistema informático. El terminal PCIe 130 incluye múltiples tipos, tales como una unidad 131 de procesamiento de gráficos, una tarjeta 132 de interfaz de red, un disco 133 de estado sólido y un componente 25 134 de aceleración de vídeo. La memoria 120 está configurada para almacenar datos, en donde los datos almacenados pueden ser datos obtenidos por la CPU desde un dispositivo externo o también pueden ser datos del programa que permiten que la CPU opere. En particular, en la memoria se pueden almacenar uno o más módulos de programa, y la CPU 110 ejecuta una operación relacionada basada en una instrucción de ejecución para el ordenador de un módulo de programa. En el sistema informático que se ilustra en la FIG. 1, el terminal PCIe 130 y la CPU 110 constituyen un campo PCIe, y todos los dispositivos en el campo PCIe se conectan a la CPU 110 a través del bus PCIe 140 y son controlados por la CPU 110.

30 En la arquitectura del sistema que se ilustra en la FIG. 1, tal como se ilustra en la FIG. 2, los módulos de programa almacenados en la memoria 120 pueden incluir específicamente un módulo 121 de aplicación, un módulo 122 de control y un sistema operativo anfitrión (Host Operating System, HOS) 123. El módulo 121 de aplicación genera una solicitud de acceso a un terminal PCIe. El módulo 122 de control puede pasarle el control a una interfaz correspondiente del HOS 123 (si el HOS proporciona una interfaz de acceso) de acuerdo con la solicitud del módulo de aplicación para acceder al terminal PCIe. El HOS 123 genera una instrucción de acceso de acuerdo con, en respuesta a, la invocación por parte del controlador, de modo que la CPU accede o controla el terminal PCIe correspondiente de acuerdo con la instrucción de acceso. En general, un terminal PCIe se corresponde con un controlador (después de eso, es posible que un controlador se corresponda con múltiples terminales PCIe, siempre que se asegure que cada terminal PCIe esté equipado con un controlador correspondiente). Por ejemplo, de acuerdo con la arquitectura del sistema que se ilustra en la FIG. 1, los controladores de los terminales PCIe en la memoria 120 pueden incluir un controlador 122-1 de la unidad de procesamiento de gráficos, un controlador 122-2 de la tarjeta de interfaz de red NIC, un controlador 122-3 del disco de estado sólido SSD, y un controlador 122-4 del componente de 45 aceleración de vídeo.

Por ejemplo, si el módulo de aplicación genera una solicitud de acceso a un disco de estado sólido SSD, de acuerdo con el método actual para acceder a un terminal PCIe por parte de la CPU, el controlador 122-3 del SSD invoca al HOS 123 después de recibir la respuesta a la invocación por el módulo 121 de aplicación; el HOS 123 genera una instrucción de acceso para la CPU 110 de acuerdo con una interfaz de acceso configurada por defecto, en donde la instrucción de acceso incluye una instrucción del dispositivo SSD 133 al que se desea acceder y el requisito de 50 operación relacionado; la CPU 110 le envía una solicitud de acceso al SSD 133 de acuerdo con la instrucción de acceso del controlador 122-3 del SSD, con el fin de solicitar acceso a un registro del SSD 133; si el SSD 133 se ha desconectado de forma anómala, la CPU 110 no recibirá desde el SSD 133 un mensaje de respuesta a la solicitud de acceso de la CPU 110, y en ese instante la CPU considera que la tarea de acceso no se ha completado; y si dichas tareas incompletas se acumulan en la CPU hasta un cierto límite, la CPU considera que todo el sistema se 55

encuentra en un estado anómalo y notifica un error de MCE para su reinicio.

En este modo de realización de la presente invención cambia el método para acceder a un terminal PCIe por parte de la CPU. La CPU ya no accede directamente al terminal PCIe, sino que accede al terminal PCIe a través de un tercero. Tal como se ilustra en la FIG. 1, en este modo de realización de la presente invención se incorpora al sistema un proxy 160 de acceso, en donde el proxy 160 de acceso se utiliza en lugar de la CPU 110 para acceder al terminal PCIe y confinar el impacto en la CPU 110 de una desconexión anómala del terminal PCIe. Tal como se ilustra en la FIG. 1, la CPU 110 ya no utiliza la línea 1 para acceder al SSD 133, sino que utiliza la línea 2 y la línea 3 (la línea 1 es la Línea 1 que se ilustra en la figura, la línea 2 es la Línea 2 que se ilustra en la figura, la línea 3 es la Línea 3 que se ilustra en la figura, y las líneas de puntos de Línea 1-3 que se ilustran en la figura no son conexiones reales, sino que se utilizan únicamente para ilustrar visualmente las líneas de flujo de señal entre cada uno de los módulos de la composición). La CPU 110 recibe en primer lugar la instrucción de acceso, en donde la instrucción de acceso le indica a la CPU que acceda al SSD 133 a través del proxy 160 de acceso. A continuación, la CPU 110 le envía la solicitud de acceso al proxy 160 de acceso a través de la línea 2. El proxy 160 de acceso le devuelve a la CPU 110 a través de la línea 2 un mensaje de respuesta a la solicitud de acceso. A continuación, el proxy de acceso realiza el acceso al terminal PCIe de acuerdo con la solicitud de acceso, esto es, realiza operaciones de lectura y escritura en el registro del SSD 133 a través de la línea 3. De este modo, en un aspecto, como la CPU 110 no genera una relación de señal directa con el terminal PCIe 130, para la CPU 110 resulta invisible si el terminal PCIe 130 está desconectado, esto es, el terminal PCIe no afecta al procesamiento del servicio de la CPU 110; en otro aspecto, el proxy 160 de acceso proporcionado por este modo de realización de la presente invención es capaz de devolverle a la CPU 110 un mensaje de respuesta después de recibir desde la CPU 110 la instrucción de acceso, de modo que siempre se puede recibir el correspondiente mensaje de respuesta a la solicitud de acceso enviada por la CPU 110. Por consiguiente, las tareas de acceso de la CPU 110 no se acumulan como consecuencia de no haberse completado dichas tareas, no se genera un error de MCE y se evita el reinicio del sistema activado por la CPU.

En este modo de realización de la presente invención se puede modificar el método para acceder a un terminal PCIe por parte de la CPU actualizando o mejorando un controlador correspondiente al terminal PCIe. Cuando esto se implementa reconstruyendo el controlador correspondiente al terminal PCIe, en primer lugar se configura una interfaz de acceso en el controlador correspondiente al terminal PCIe, en donde la interfaz de acceso predeterminada apunta al proxy de acceso; y cuando el controlador correspondiente al terminal PCIe determina que es necesario acceder al terminal PCIe, el controlador del terminal PCIe genera una instrucción de acceso de la CPU en relación con la interfaz de acceso predeterminada, en donde la instrucción de acceso le indica a la CPU que acceda al terminal PCIe a través del proxy de acceso.

Adicionalmente, el cambio del método para acceder a un terminal PCIe por parte de la CPU se puede implementar de acuerdo con otro método. Por ejemplo, el cambio se implementa modificando el HOS. En primer lugar se configura una interfaz de acceso en el HOS, en donde la interfaz de acceso predeterminada apunta al proxy de acceso. Cuando se determina que es necesario acceder al terminal PCIe, el controlador del terminal PCIe aún invoca al HOS para acceder al terminal PCIe. Después de que el HOS reciba la instrucción de invocación enviada por el terminal PCIe, como se ha predeterminado que la interfaz para acceder al terminal PCIe configurado en el HOS apunte al proxy de acceso, el HOS genera la instrucción de acceso, en donde la instrucción de acceso le indica a la CPU que acceda al terminal PCIe a través del proxy de acceso.

A continuación se describen las funciones y las formas de implementación detalladas del proxy de acceso de acuerdo con este modo de realización de la presente invención. El proxy de acceso de acuerdo con este modo de realización de la presente invención incluye una función de confinamiento y una función de proxy de acceso. Funcionando como módulo de confinamiento, el proxy de acceso tiene que asegurar su independencia del terminal PCIe, así como asegurar su independencia de la CPU. Para mantener la independencia del terminal PCIe es necesario garantizar que el proxy de acceso no se pueda extraer directamente con el terminal PCIe y, por consiguiente, el proxy de acceso y el terminal PCIe deben pertenecer a dispositivos diferentes en la configuración física; para mantener la independencia de la CPU, se asegura esencialmente que el proxy de acceso tenga un procesador independiente. Cuando el procesador del proxy de acceso es independiente de la CPU del sistema, incluso si se extrae directamente el terminal PCIe, un impacto en el módulo proxy de acceso no contaminará a la CPU. Funcionando como módulo proxy, el proxy de acceso necesita realizar el acceso al terminal PCIe y devolver un mensaje de respuesta a una solicitud de acceso recibida desde la CPU, en donde el mensaje de respuesta a la solicitud de acceso puede ser una respuesta de confirmación, una respuesta de rechazo, o una respuesta de fallo. No obstante, cualquiera que sea el tipo de mensaje de respuesta, el mensaje le indica a la CPU que se ha recibido la solicitud de acceso enviada por la CPU. Después de recibir el mensaje de respuesta la CPU determina que la tarea actual se ha completado, y puede poner a cero el temporizador iniciado para la tarea. De este modo, el mecanismo de la CPU de puesta a cero del temporizador de tareas se comporta de forma normal, y los otros mensajes no se acumulan debido a la expiración del tiempo de espera, evitándose de este modo un reinicio por MCE generado por la CPU.

A partir de la consideración de las funciones del proxy de acceso, el proxy de acceso también se puede configurar en el sistema de múltiples formas. En la arquitectura del sistema que se ilustra en la FIG. 1, el proxy 160 de acceso se configura en el sistema informático como un dispositivo independiente y de nueva incorporación, y el proxy de acceso se conecta a la CPU y al terminal PCIe a través del bus PCIe. Por otro lado, el proxy 160 de acceso también

se puede empaquetar con un dispositivo existente en el campo PCIe para su implementación. Por ejemplo, el proxy 160 de acceso se empaqueta con la CPU como firmware.

En este modo de realización de la presente invención el proxy de acceso se puede implementar mediante un hardware ya existente. Por ejemplo, el proxy de acceso se implementa utilizando un controlador de acceso directo a memoria (Direct Memory Access, DMA). El proxy de acceso también se puede implementar mediante un nuevo hardware. Por ejemplo, en un dispositivo de hardware que dispone de un procesador independiente se instala un módulo de software que tiene una función de proxy de acceso.

Como en este modo de realización de la presente invención el proxy de acceso tiene que devolverle a la CPU un mensaje de respuesta a la solicitud de acceso, en la forma concreta de implementación existen diferentes maneras de implementar la función del proxy de acceso. Una de las formas de implementación consiste en asegurar que la conexión entre el proxy de acceso y la CPU está asegurada, esto es, la relación de conexión entre los dos no se desconectará, o que el proxy de acceso no se puede conectar/desconectar en caliente de la CPU. Por ejemplo, un dispositivo de hardware en el que se ha cargado el proxy de acceso o un dispositivo de hardware que se utiliza para implementar el proxy de acceso se suelda a una placa de circuito impreso PCB conectada a la CPU, o una interfaz para conectar el dispositivo de hardware en el que se ha cargado el proxy de acceso o el dispositivo de hardware que se utiliza para implementar el proxy de acceso y el procesador se han fijado mediante un conector.

La FIG. 3 ilustra un sistema informático de acuerdo con otro modo de realización de la presente invención.

En el sistema informático que se ilustra en la FIG. 3, excepto la CPU, el bus PCIe y el terminal PCIe que se ilustran en la FIG. 1, se incluye además un intercambiador PCIe 150. Un puerto de flujo ascendente del intercambiador PCIe 150 está conectado a la CPU 110 a través de un bus PCIe 140, y un puerto de flujo descendente proporciona un puerto PCIe para cada uno de los terminales PCIe, en donde cada uno de los puertos PCIe se conecta a un terminal PCIe a través del bus PCIe 140. El intercambiador PCIe 150 está configurado para encaminar en el flujo descendente los datos hacia el puerto PCIe correspondiente y encaminar en el flujo ascendente hacia la CPU 110 los datos desde cada uno de los puertos PCIe independientes. En el modo de realización que se ilustra en la FIG. 3, se configura un nuevo proxy 160 de acceso dentro del intercambiador PCIe 150, y el proxy 160 de acceso de este modo de realización se implementa utilizando un controlador de DMA. El terminal PCIe 130 se conecta al intercambiador PCIe 150 a través del bus PCIe 140. Como el intercambiador PCIe 150 y el terminal PCIe 130 pertenecen a dispositivos diferentes, la extracción directa de cualquier terminal PCIe no provoca la extracción del intercambiador PCIe 150 del sistema, esto es, se asegura que con la extracción del terminal PCIe no se elimina el proxy 160 de acceso, implementándose de este modo la independencia del proxy 160 de acceso y el del terminal PCIe 130. Adicionalmente, en este modo de realización, como el controlador de DMA tiene un procesador independiente, si se extrae directamente cualquier terminal PCIe e incluso se afecta el acceso al terminal PCIe desde el DMA, el DMA confina el impacto. Independientemente de si el acceso al terminal PCIe se realiza con éxito, se garantiza que el DMA le devuelve a la CPU 110 un mensaje de respuesta a la solicitud de acceso enviada por la CPU 110, evitándose de este modo el problema de reinicio por MCE activado por la CPU.

Se continúa utilizando a modo de ejemplo la generación por parte de un módulo de aplicación de una solicitud de acceso a un disco 133 de estado sólido SSD. La CPU 110 recibe una instrucción de acceso generada por un controlador 122-3 del disco de estado sólido SSD, en donde la instrucción de acceso le indica a la CPU 110 que acceda al disco 133 de estado sólido SSD utilizando el DMA. La CPU 110 le envía una petición de migración de datos al DMA de acuerdo con la instrucción de acceso del controlador 122-3 del disco de estado sólido SSD, en donde la petición de migración de datos le indica al controlador de DMA que transfiera unos datos determinados de la memoria del terminal PCIe a la memoria del sistema informático, o que transfiera unos datos determinados de la memoria del sistema informático a la memoria del terminal PCIe. Después de recibir de la CPU 110 la petición de migración de datos, el DMA le devuelve a la CPU 110 un mensaje de respuesta a la petición de migración de datos, realiza la transferencia de datos en el disco 133 de estado sólido SSD, y después de completar la transferencia de datos le devuelve a la CPU 110 un mensaje de notificación de haber completado el acceso con el fin de indicarle a la CPU 110 que puede recibir el resultado del acceso.

Además, como en este modo de realización de la presente invención el DMA está integrado en el intercambiador PCIe 150, el intercambiador PCIe 150 se puede soldar, además, en una placa de circuito impreso PCB conectada a la CPU 110, o la interfaz para conectar el intercambiador PCIe 150 y la CPU 110 se fija mediante el uso de un terminal. De este modo se asegura que el DMA integrado en el intercambiador PCIe 150 no se pueda extraer del sistema y, por consiguiente, asegura que el DMA siempre sea capaz de devolver a la CPU un mensaje de respuesta a una solicitud de acceso.

La FIG. 4 ilustra un sistema informático de acuerdo con otro modo de realización de la presente invención.

El modo de realización que se ilustra en la FIG. 4 es diferente del modo de realización que se ilustra en la FIG. 3 en que un proxy 160 de acceso se ha incorporado recientemente en una CPU 110, en donde el proxy 160 de acceso puede implementarse utilizando un controlador de DMA. El proxy 160 de acceso se integra dentro de la CPU 110, esto es, se asegura que el proxy 160 de acceso no se elimine con la eliminación de un terminal PCIe, implementando así la independencia del servidor proxy 160 de acceso y el terminal PCIe 130. Además, en este

modo de realización, como el controlador de DMA tiene un procesador independiente, si se elimina directamente cualquier terminal PCIe e incluso se afecta el acceso al terminal PCIe del DMA, el DMA confina el impacto para que la CPU 110 no se vea afectada. Independientemente de si el acceso al terminal PCIe se ha realizado con éxito, se garantiza que el DMA devuelve a la CPU 110 un mensaje de respuesta a la solicitud de acceso enviada por la CPU 110, evitando así un problema de reinicio por MCE iniciado por la CPU. Un método de acceso detallado en este modo de realización es consistente con los descritos en los modos de realización que se ilustran en la FIG. 1 y la FIG. 3, y por lo tanto, no se proporcionan más detalles en la presente solicitud.

En los modos de realización de la presente invención, en el sistema informático que se ilustra en las FIG. 1, 3 ó 4 se puede implementar un método para acceder a un terminal PCIe. Sin embargo, lo que se ilustra en las FIG. 1, 3 ó 4 es solo un ejemplo aplicable a los modos de realización de la presente invención pero no constituye una limitación específica a la aplicación de la presente invención. En el documento de esta solicitud no se describen otros modos de realización o escenarios de aplicación del sistema. Por otro lado, las configuraciones del proxy de acceso en el sistema que se ilustra en las FIG. 1, 3 ó 4 son solo ejemplos. Aquellos experimentados en la técnica pueden ubicar, además, el proxy de acceso recién incorporado en los modos de realización de la presente invención en otra posición del sistema o utilizar otros medios técnicos para la implementación de acuerdo con el principio técnico de los modos de realización de la presente invención.

La CPU 110 que se ilustra en las FIG. 1, 3 ó 4 también es solo un ejemplo. Por ejemplo, puede ser un circuito integrado específico. No importa en qué forma, implementa una función de procesador en un sistema informático. El sistema informático en los modos de realización de la presente invención puede ser un servidor de cálculo, o puede ser un servidor que gestiona rutas, como un conmutador. Una forma de implementación detallada del sistema informático no está limitada en este modo de realización de la presente invención.

Proceso para acceder a un terminal PCIe

A continuación, se describe un proceso para acceder a un terminal PCIe de acuerdo con un modo de realización de la presente invención, en donde el proceso se implementa mediante un proxy de acceso recién incorporado a un sistema informático. Tal como se ilustra en la FIG. 5, en este modo de realización de la presente invención un proceso para acceder a un terminal PCIe incluye:

S501: Una CPU recibe una instrucción de acceso, en donde la instrucción de acceso le indica a la CPU que acceda al terminal PCIe mediante el proxy de acceso del sistema informático.

En particular, la instrucción de acceso puede ser generada por un controlador del terminal PCIe. Como el controlador del terminal PCIe ha configurado previamente el proxy de acceso como interfaz para acceder al terminal PCIe, cuando un módulo de aplicación de una capa superior genera una solicitud de acceso a un terminal PCIe, el controlador del terminal PCIe genera una instrucción de acceso para acceder al terminal PCIe, en donde la instrucción de acceso le indica a la CPU que acceda al terminal PCIe al que se desea acceder a través del proxy de acceso del sistema informático. Alternativamente, la instrucción de acceso también puede ser generada por el HOS del sistema informático. El HOS ha configurado previamente el proxy de acceso como interfaz para acceder al terminal PCIe. Cuando el módulo de aplicación de la capa superior genera una solicitud de acceso a un terminal PCIe, el controlador del terminal PCIe invoca al HOS y el HOS genera la instrucción de acceso de acuerdo con la interfaz de acceso predeterminada, en donde la instrucción de acceso le indica a la CPU que acceda al terminal PCIe al que se desea acceder a través del proxy de acceso del sistema informático.

S502: La CPU le envía una solicitud de acceso al proxy de acceso de acuerdo con la instrucción de acceso, en donde la solicitud de acceso le indica al proxy de acceso que acceda al terminal PCIe.

S503: Después de recibir la solicitud de acceso enviada por la CPU el proxy de acceso le devuelve a la CPU un mensaje de respuesta a la solicitud de acceso.

El mensaje de respuesta a la solicitud de acceso puede ser una respuesta de confirmación, una respuesta de rechazo o una respuesta de fallo. Sin embargo, cualquiera que sea el tipo de mensaje de respuesta, el mensaje le indica a la CPU que se ha recibido la solicitud de acceso enviada por la CPU. Después de recibir el mensaje de respuesta la CPU determina que la tarea actual se ha completado y puede poner a cero el temporizador iniciado para la tarea. De este modo, el mecanismo de la CPU de puesta a cero del temporizador de tareas se comporta de forma normal.

En el proceso, la CPU no accede directamente a un terminal PCIe al que se desea acceder, sino que completa el acceso mediante un proxy de acceso. El proxy de acceso es capaz de confinar el impacto ocasionado por una desconexión anómala del terminal PCIe y el proxy de acceso le devuelve a la CPU un mensaje de respuesta a la solicitud de acceso, de modo que las tareas almacenadas en la CPU no se acumulan debido a la expiración del tiempo de espera, evitándose de este modo un reinicio por MCE para la CPU.

Además, tal como se ilustra en la FIG. 6, en otro modo de realización del proceso de la presente invención, el proceso para que un proxy de acceso acceda a un terminal PCIe incluye:

S601-S603: Estos pasos son los mismos que los pasos S501-S503 y, por consiguiente, no se proporcionan más detalles en la presente solicitud.

S604: El proxy de acceso inicia una operación de acceso al terminal PCIe de acuerdo con la solicitud de acceso de la CPU.

5 S605: El proxy de acceso determina si la operación de acceso iniciada para el terminal PCIe se ha ejecutado con éxito; si la operación de acceso se ha realizado con éxito, continúa en el paso 606; y si la operación de acceso ha fallado, continúa en el paso 608.

S606: El proxy de acceso le envía a la CPU un primer mensaje de notificación de que el acceso se ha completado.

S607: Después de recibir el primer mensaje de notificación la CPU recibe el resultado del acceso.

10 Posteriormente, en función del resultado del acceso la CPU puede notificarle al módulo de la capa superior que el acceso se ha completado.

S608: El proxy de acceso le envía a la CPU un segundo mensaje de notificación de un fallo de acceso.

S609: Después de recibir el segundo mensaje de notificación la CPU ejecuta el procesamiento subsiguiente al fallo de acceso.

15 En particular, el procesamiento subsiguiente a un fallo de acceso incluye: determinar el motivo por el que el proxy de acceso no ha podido acceder al terminal PCIe; si el motivo del fallo de acceso es que el terminal PCIe al que se desea acceder se ha desconectado de forma anómala, interrumpir, por parte de la CPU, el acceso al terminal PCIe; y si el motivo del fallo de acceso es que ha fallado el proxy de acceso, reiniciar, por parte de la CPU, el proxy de acceso, o enviar una notificación indicando que el proxy de acceso ha fallado, con el fin de rectificar el fallo del proxy de acceso.

20 Después de interrumpir el acceso al terminal PCIe, la CPU puede, además, notificarle al módulo de capa superior que interrumpa el acceso al terminal PCIe.

25 El proceso describe un método para acceder a un terminal PCIe en este modo de realización de la presente invención. En dicho método un proxy de acceso, en lugar de la CPU, accede al terminal PCIe y le devuelve a la CPU un mensaje de respuesta a la solicitud de acceso, evitando de este modo un reinicio de todo el sistema debido a un error de MCE generado por la CPU. Además, cuando el proxy de acceso no consigue acceder al terminal PCIe, el proxy de acceso le notifica a la CPU un mensaje de fallo de acceso. La CPU realiza un diagnóstico del fallo y cuando se determina que el fallo de acceso se debe a una desconexión anómala del terminal PCIe al que se desea acceder, se interrumpe el acceso al terminal PCIe al que se desea acceder, evitándose de este modo el consumo inútil de recursos que se produce cuando el sistema continúa intentando repetidamente accesos que no se pueden completar con éxito.

30 En combinación con el modo de realización del sistema informático que se ilustra en las FIG. 3 ó 4, cuando se implementa un proxy de acceso utilizando un controlador de DMA, un módulo de aplicación de una capa superior genera una solicitud de operación de lectura para un SSD, en la FIG. 7 se ilustra un proceso específico de acceso, que incluye:

35 S701: La CPU de un sistema informático recibe una instrucción de acceso, en donde la instrucción de acceso incluye una interfaz de acceso y un contenido del acceso, la interfaz de acceso apunta al controlador de DMA, el contenido del acceso indica que el objeto de acceso es el SSD y el acceso es una operación de lectura, e indica la dirección de origen de la operación de lectura; y el contenido del acceso puede indicar además la longitud de la operación de lectura, aunque, generalmente, la longitud de la operación de lectura puede estar sujeta a una longitud por defecto en el sistema.

40 Cuando un terminal del flujo ascendente genera una solicitud de operación de lectura para el dispositivo SSD, un controlador del dispositivo SSD recibe en respuesta a la invocación por parte del terminal del flujo ascendente, y genera, una instrucción de acceso para acceder a un terminal PCIe de acuerdo con la interfaz de acceso predeterminada.

45 El método de implementación detallado de la instrucción de acceso enviada por el controlador a la CPU puede ser además de otra forma. Por ejemplo, la instrucción de acceso incluye informaciones que indican que el objeto de acceso es el SSD y el acceso es una operación de lectura e indica una dirección de inicio de la operación de lectura; además, a la instrucción de acceso se le añade una información para indicar el acceso al SSD que se deberá implementar mediante el controlador de DMA.

50 S702: La CPU le envía una petición de migración de datos al controlador de DMA de acuerdo con la instrucción de acceso, en donde la petición de migración de datos se utiliza para indicarle al controlador de DMA que transfiera unos datos determinados de la memoria del terminal PCIe a la memoria del sistema informático.

En particular, después de recibir la instrucción de acceso del controlador del SSD, la CPU solicita de la memoria del sistema informático una dirección de destino de la operación de lectura y, después de recibir la dirección de destino de la operación de lectura, le envía la petición de migración de datos al controlador de DMA, en donde la petición de migración de datos indica la dirección de origen, la dirección de destino y la longitud de la operación de lectura, con el fin de indicarle al controlador de DMA que transfiera datos de la longitud indicado para la operación de lectura desde la dirección de origen de la operación de lectura a la dirección de destino de la operación de lectura.

5

S703: Después de recibir la petición de migración de datos de la CPU el controlador de DMA le devuelve a la CPU un mensaje de respuesta a la petición de migración de datos, y después de recibir el mensaje de respuesta a la petición de migración de datos la CPU no cuenta el tiempo de espera para la petición de migración de datos, con el fin de asegurar que otros mensajes almacenados en la CPU no darán lugar a que la CPU genere un reinicio por MCE debido a la acumulación.

10

S704: El controlador de DMA inicia una solicitud de lectura para el dispositivo SSD, en donde la solicitud de lectura incluye la dirección de origen de la operación de lectura, y la operación de lectura se utiliza para solicitar que se lea en la memoria intermedia del controlador de DMA el valor de un registro correspondiente a la dirección de origen de la operación de lectura.

15

S705: El controlador de DMA determina si la solicitud de lectura se ha ejecutado con éxito; si la solicitud de lectura se ha ejecutado con éxito, continúa en el paso 706; y si la solicitud de lectura no se ha ejecutado con éxito, continúa en el paso 709.

S706: El controlador de DMA graba los datos en la dirección de destino de la operación de lectura dentro de su propia memoria intermedia de acuerdo con la solicitud de escritura.

20

S707: El controlador de DMA le envía a la CPU un primer mensaje de notificación, en donde el primer mensaje de notificación puede ser específicamente una primera interrupción MSI (Message Signaled Interrupts, MSI) para notificarle a la CPU que el acceso se ha completado.

S708: Después de recibir el primer mensaje de interrupción MSI la CPU lee los datos de la dirección de destino de la operación de lectura y puede notificarle al controlador del dispositivo de SSD que el acceso se ha completado.

25

S709: El controlador de DMA le envía a la CPU un segundo mensaje de notificación, en donde el segundo mensaje de notificación puede ser específicamente una segunda interrupción MSI para notificarle a la CPU que el acceso ha fallado.

S710: Después de recibir el segundo mensaje de interrupción MSI la CPU ejecuta el procesamiento subsiguiente a un fallo de acceso.

30

En particular, el procesamiento subsiguiente a un fallo de acceso puede incluir: iniciar un diagnóstico del controlador de DMA para determinar si ha fallado el controlador de DMA;

si el controlador de DMA ha fallado, reiniciar, por parte de la CPU, el controlador de DMA o enviar una notificación indicando que el controlador de DMA ha fallado, con el fin de recuperar el fallo del controlador de DMA; y

35 si el controlador de DMA no ha fallado, determinar que la causa del fallo de acceso es que el dispositivo SSD se ha desconectado de forma anómala, e interrumpir, por parte de la CPU, el acceso al dispositivo SSD.

Por otro lado, la CPU puede además indicarle al controlador del dispositivo SSD que interrumpa el acceso al dispositivo SSD.

En otro aspecto, haciendo referencia al modo de realización del sistema informático que se ilustra en las FIG. 3 ó 4, cuando se implementa un proxy de acceso utilizando un controlador de DMA, un módulo de aplicación de una capa superior genera una solicitud de operación de escritura para el SSD, en la FIG. 8 se ilustra un proceso específico de acceso, que incluye:

40

S801: La CPU recibe una instrucción de acceso generada por el controlador del SSD, en donde la instrucción de acceso incluye la interfaz de acceso y el contenido del acceso, la interfaz de acceso apunta al controlador de DMA y el contenido del acceso indica que el objeto del acceso es el dispositivo SSD y el acceso es una operación de escritura, e indica la dirección de origen y la dirección de destino de la operación de escritura.

45

El método de implementación detallado de la instrucción de acceso enviada por el controlador a la CPU puede ser además de otra forma. Por ejemplo, la instrucción de acceso incluye informaciones que indican que el objeto de acceso es el SSD y el acceso es una operación de escritura, e indica la dirección de origen y la dirección de destino de la operación de escritura; adicionalmente, en la instrucción de acceso se agrega una nueva indicación para indicar el acceso al SSD que se deberá implementar mediante el controlador de DMA.

50

S802: La CPU le envía al controlador de DMA una solicitud de acceso de acuerdo con la instrucción de acceso del controlador del SSD, en donde la petición de migración de datos se utiliza para indicarle al controlador de DMA que

transfiera unos datos determinados de la memoria del sistema informático a la memoria del terminal PCIe.

5 En particular, después de recibir la instrucción de acceso del controlador del SSD la CPU le envía al controlador de DMA una petición de migración de datos, en donde la petición de migración de datos indica la dirección de origen, la dirección de destino y la longitud de la operación de escritura, con el fin de indicarle al controlador de DMA que transfiera datos de la longitud indicada para la operación de escritura desde la dirección de origen de la operación de escritura a la dirección de destino de la operación de escritura.

S803: Después de recibir la petición de migración de datos de la CPU el controlador de DMA le devuelve a la CPU un mensaje de respuesta a la petición de migración de datos.

10 S804: El controlador de DMA inicia una solicitud de lectura de la dirección de origen de la operación de escritura con el fin de leer los datos de la dirección de origen en un memoria intermedia del controlador de DMA.

S805: Después de leer los datos de la dirección de origen en su propia memoria intermedia el controlador de DMA inicia una solicitud de escritura en el dispositivo SSD, en donde la solicitud de escritura incluye la dirección de destino de la operación de escritura y la solicitud de escritura se utiliza para grabar los datos de la memoria intermedia del controlador de DMA en un registro que corresponde a la dirección de destino.

15 S806: El controlador de DMA determina si la solicitud de escritura se ha ejecutado con éxito; si la solicitud de escritura se ha ejecutado con éxito, continúa en el paso 807; y si la solicitud de escritura no se ha ejecutado con éxito, continúa en el paso 809.

S807: El controlador de DMA inicia una primera interrupción MSI (Message Signaled Interrupts, MSI) para la CPU con el fin de notificarle a la CPU que el acceso se ha completado.

20 S808: Después de recibir el primer mensaje de interrupción MSI la CPU detecta que la operación de escritura se ha completado y puede además notificarle al controlador del dispositivo SSD que el acceso se ha completado.

S809: El controlador de DMA inicia una segunda interrupción MSI para la CPU con el fin de notificarle a la CPU que el acceso ha fallado.

25 S810: Después de recibir el segundo mensaje de interrupción MSI la CPU ejecuta el procesamiento subsiguiente a un fallo de acceso.

En particular, el procesamiento subsiguiente a un fallo de acceso puede incluir: iniciar un diagnóstico en el controlador de DMA para determinar si ha fallado el controlador de DMA;

si el controlador de DMA ha fallado, reiniciar, por parte de la CPU, el controlador de DMA o enviar una notificación indicando que el controlador de DMA ha fallado, con el fin de recuperar el fallo del controlador de DMA; y

30 si el controlador de DMA no ha fallado, determinar que la causa del fallo de acceso es que el dispositivo SSD se ha desconectado de forma anómala, e interrumpir, por parte de la CPU, el acceso al dispositivo SSD.

Además, la CPU también puede indicarle al controlador del dispositivo SSD que interrumpa el acceso al dispositivo SSD.

35 En los modos de realización de la presente invención, los procesos que se ilustran en las FIG. 7 y 8 describen procesos del método para que un controlador de DMA complete la lectura o escritura en un dispositivo SSD. En el método de lectura o escritura, el controlador de DMA, en lugar de la CPU, accede al terminal PCIe y le devuelve a la CPU un mensaje de respuesta a la solicitud de acceso, con el fin de que la CPU no genere un error de MCE y así evitar un reinicio de todo el sistema. Además, cuando el controlador de DMA no consigue transferir los datos desde el dispositivo SSD, el controlador de DMA le notifica a la CPU un mensaje de fallo de acceso, la CPU realiza un diagnóstico del fallo y, cuando determina que el fallo de acceso se debe a que el dispositivo SSD se ha extraído directamente del sistema o ha fallado, se interrumpe el acceso al dispositivo SSD, evitando de este modo el consumo inútil de recursos que se produce cuando el sistema continúa intentando repetidamente accesos que no se pueden completar con éxito.

45 Adicionalmente, en este modo de realización de la presente invención se puede modificar el método para que la CPU acceda a un terminal PCIe actualizando o mejorando el controlador que corresponde al terminal PCIe o el sistema operativo anfitrión. Si el controlador correspondiente al terminal PCIe se utiliza para modificar el método para que la CPU acceda a un terminal PCIe, se puede incluir el siguiente proceso:

S901: El controlador del terminal PCIe recibe una instrucción de invocación de un módulo de aplicación de una capa superior, en donde la instrucción de invocación indica que es necesario acceder al terminal PCIe.

50 S902: El controlador correspondiente al terminal PCIe genera una instrucción de acceso de acuerdo con una interfaz predeterminada de acceso al terminal PCIe, en donde la interfaz predeterminada de acceso al terminal PCIe apunta a un proxy de acceso y la instrucción de acceso se utiliza para indicarle a la CPU que acceda al terminal PCIe a

través del proxy de acceso.

Si para modificar el método para que una CPU acceda a un terminal PCIe se ha configurado el sistema operativo anfitrión, se puede incluir el siguiente proceso:

5 S1001: El controlador correspondiente al terminal PCIe recibe una instrucción de invocación de un módulo de aplicación de una capa superior, en donde la instrucción de invocación indica que es necesario acceder al terminal PCIe.

S1002: El controlador correspondiente al terminal PCIe invoca al sistema operativo anfitrión, en donde la instrucción de invocación indica que es necesario acceder al terminal PCIe.

10 S1003: El sistema operativo anfitrión genera una instrucción de acceso de acuerdo con una interfaz predeterminada para acceder al terminal PCIe, en donde la interfaz predeterminada de acceso al terminal PCIe apunta a un proxy de acceso, y la instrucción de acceso se utiliza para indicarle a la CPU que acceda al terminal PCIe a través del proxy de acceso.

Equipo en un modo de realización de la presente invención

15 Tal como se ilustra en la FIG. 11, un equipo para acceder a un terminal de Interconexión de Componentes Periféricos Exprés PCIe proporcionado en un modo de realización de la presente invención incluye:

un módulo 1101 de recepción, configurado para recibir una instrucción de invocación, en donde la instrucción de invocación indica que es necesario acceder al terminal PCIe; y

20 un módulo 1102 de generación, configurado para generar, de acuerdo con una interfaz predeterminada de acceso al terminal PCIe, una instrucción de acceso para acceder al terminal PCIe, en donde la interfaz predeterminada de acceso al terminal PCIe apunta a un proxy de acceso y la instrucción de acceso se utiliza para indicarle a la CPU que acceda al terminal PCIe a través del proxy de acceso.

En particular, el equipo de acceso puede ser el controlador del terminal PCIe o el sistema operativo anfitrión del sistema informático.

25 La FIG. 12 es un diagrama de composición de la estructura de un ordenador de acuerdo con un modo de realización de la presente invención. En este modo de realización de la presente invención el ordenador puede incluir:

un procesador 1201, una memoria 1202, un bus 1204 del sistema y una interfaz 1205 de comunicaciones. La CPU 1201, la memoria 1202 y la interfaz 1205 de comunicaciones conectan y completan las comunicaciones entre las mismas mediante el bus 1204 del sistema.

30 En este modo de realización de la presente invención el procesador 1201 puede ser una unidad de procesamiento central de un solo núcleo o múltiples núcleos, o un circuito integrado específico, o está configurado como uno o más circuitos integrados.

La memoria 1202 puede ser una memoria RAM de alta velocidad o puede ser una memoria no volátil (non-volatile memory) como, por ejemplo, al menos una memoria de disco magnético.

35 La memoria 1202 está configurada para almacenar una instrucción 1203 de ejecución para el ordenador. En particular, la instrucción 1203 de ejecución para el ordenador puede incluir un código de programa.

Cuando el ordenador se encuentra en ejecución, el procesador 1201 ejecuta la instrucción 1203 de ejecución para el ordenador y se puede ejecutar el proceso del método que se ilustra en una cualquiera de las FIG. 5 a 10.

Sistema informático para acceder a un terminal PCIe

40 Cuando se extrae un terminal PCIe de un sistema informático, es posible que más tarde se vuelva a insertar en el sistema informático. Por otro lado, también se da el caso de que se necesite conectar un nuevo terminal PCIe a un sistema informático en funcionamiento. Por ejemplo, como resultado de la popularidad de los dispositivos SSD, cada vez sucede con más frecuencia que un usuario inserta o extrae directamente un dispositivo SSD. En la técnica anterior, cuando se enciende cualquier terminal PCIe y se conecta al sistema, la CPU inicia un proceso de exploración y asignación de recursos al terminal PCIe; en el proceso de exploración del terminal PCIe recién encendido por parte de la CPU, si el terminal PCIe se extrae directamente del sistema, es posible que la CPU notifique un error de MCE, lo cual provoca un reinicio del sistema. Para evitar este problema, este modo de realización de la presente invención propone una nueva solución para asignar recursos a un terminal PCIe en la que, cuando el terminal PCIe se acaba de encender y está conectado al sistema, la CPU no necesita explorar o asignar recursos al terminal PCIe recién activado.

50 Durante el arranque del sistema informático, el sistema básico de entrada-salida (Basic Input-Output System, BIOS) tiene que reservar recursos para cada dispositivo del sistema. Para el terminal PCIe, el BIOS explora cada uno de

los puertos para acceder al terminal PCIe. Cuando se localiza un terminal PCIe mediante exploración, el BIOS lee el registro correspondiente del terminal PCIe y reserva los recursos correspondientes en función de las necesidades del terminal PCIe; por ejemplo, reserva los recursos del bus y los recursos de dirección de memoria. En términos específicos, el puerto de acceso a un terminal PCIe que se describe en este modo de realización de la presente invención puede ser un puerto de enlace descendente de un intercambiador PCIe o un puerto de enlace descendente de un northbridge (puente norte) del sistema.

En la solución que se proporciona en este modo de realización de la presente invención para asignar recursos a un terminal PCIe, el método de reserva de recursos del BIOS del sistema informático es diferente al de la técnica anterior. Durante el arranque del sistema informático, el BIOS no reserva recursos en función de los requisitos reales del terminal PCIe localizado mediante la exploración, sino que reserva un subconjunto específico de recursos para cada uno de los puertos de acceso al terminal PCIe, en donde el subconjunto específico es mayor o igual que la cantidad de recursos que necesita el terminal PCIe y, preferentemente, el subconjunto específico puede ser la cantidad de recursos requerida por el tipo de terminal PCIe que requiere la cantidad máxima de recursos. Por ejemplo, el BIOS explora cada puerto de acceso a un terminal PCIe en el sistema informático, e independientemente de si mediante la exploración se encuentra un terminal PCIe, e independientemente de qué tipo de terminal PCIe se encuentra mediante la exploración, especifica que el tipo de terminal PCIe que requiere la cantidad máxima de recursos puede acceder posteriormente a cada uno de los puertos de acceso a un terminal PCIe. Si en el sistema actual se pueden usar 10 tipos de terminales PCIe y el que requiere la cantidad máxima de recursos es un dispositivo SSD que requiere 10M de recursos de memoria que no se pueden asignar previamente, y 3 buses PCIe, el BIOS reserva para cada puerto de acceso a un terminal PCIe unos recursos de 10M que no se pueden asignar previamente y 3 recursos de bus PCIe.

En segundo lugar, después de que el BIOS haya reservado los recursos, un módulo de administración PCIe del sistema informático permite que todos los terminales PCIe gestionados por una CPU en el sistema informático y el intercambiador PCIe constituyan un campo PCIe y configura un árbol PCIe correspondiente para el campo PCIe, en donde el árbol PCIe se utiliza para describir cada una de las capas de relaciones de conexión desde cada uno de los terminales PCIe del campo PCIe a la CPU, y el estado de la configuración de recursos de cada uno de los terminales PCIe. Como el BIOS ha reservado un subconjunto específico de recursos para cada puerto de acceso a un terminal PCIe, al cargar cada puerto de acceso a un terminal PCIe el módulo de administración PCIe no analiza los requisitos reales de cantidad de recursos del terminal PCIe del puerto, sino que asigna los recursos de acuerdo con una condición de reserva de recursos del BIOS, esto es, asigna un subconjunto específico de recursos reservados por el BIOS a cada puerto de acceso a un terminal PCIe, y registra el estado de asignación del subconjunto específico de recursos en el árbol PCIe.

Además, cuando un terminal PCIe falla o se encuentra desconectado del sistema informático, cuando se determina que el terminal PCIe está desconectado el módulo de administración PCIe no libera el subconjunto específico de recursos asignados al terminal PCIe que se encuentra desconectado. Por otro lado, la arquitectura del árbol PCIe permanece inalterada, esto es, la relación de conexión y el estado de la configuración de recursos del terminal PCIe desconectado se mantienen en el árbol PCIe. De este modo, como los recursos y la relación de conexión del terminal PCIe se han configurado en el campo PCIe, cuando el terminal PCIe se enciende y accede al campo PCIe, el módulo de administración PCIe le notifica al controlador correspondiente que el terminal PCIe está encendido y el terminal PCIe completa el acceso al campo PCIe del sistema informático. En esta solución, cuando el terminal PCIe se encuentra encendido, la CPU no necesita explorar el terminal PCIe, evitándose de este modo un reinicio de todo el sistema debido a un error de MCE que se pudiera producir cuando el terminal PCIe accede al sistema informático.

Las personas con un conocimiento normal de la técnica pueden entender que cada uno de los aspectos de la presente invención o una forma posible de implementación de cada uno de dichos aspectos se pueden implementar específicamente como un sistema, un método o un producto de programa para ordenador. Por lo tanto, cada uno de los aspectos de la presente invención o formas posibles de implementación de cada uno de dichos aspectos puede adoptar la forma de un modo de realización totalmente hardware, un modo de realización totalmente software (incluyendo firmware, software residente y similares) o un modo de realización que combine software y hardware, el cual, en la presente solicitud, se designa colectivamente como "circuito", "módulo" o "sistema". Además, cada uno de los aspectos de la presente invención o formas posibles de implementación de cada uno de los aspectos pueden adoptar la forma de producto de programa para ordenador, en donde el producto de programa para ordenador se refiere a un código de programa legible por ordenador en un medio legible por ordenador.

El medio legible por ordenador puede ser un medio de señales legibles por ordenador o un medio de memoria legible por ordenador. El medio de memoria legible por ordenador incluye, pero no se limita a, un sistema, un dispositivo o equipo electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor, o cualquier combinación apropiada de los mismos, tal como una memoria de acceso aleatorio (RAM), una memoria de solo lectura, una memoria de solo lectura programable y borrable (EPROM o memoria Flash), una fibra óptica, y una memoria portátil de solo lectura (CD-ROM).

El procesador de un ordenador lee el código de programa legible por ordenador almacenado en el medio legible por ordenador, de modo que el procesador es capaz de ejecutar las acciones funcionales estipuladas en cada paso o combinaciones de cada uno de los pasos de un diagrama de flujo, y se constituye un equipo que implementa las

acciones funcionales estipuladas en cada bloque o una combinación de bloques del diagrama de bloques.

El código del programa legible por ordenador puede ser ejecutado completamente en el ordenador de un usuario, ejecutado parcialmente en el ordenador del usuario, ejecutado parcialmente en el ordenador del usuario y ejecutado completamente en un ordenador remoto cuando funciona como un paquete de software independiente, o ejecutado completamente en el ordenador remoto o en un servidor. También se debe observar que, en algunas soluciones de implementación alternativas, las funciones indicadas en cada paso en el diagrama de flujo o en cada bloque de un diagrama de bloques pueden no tener lugar en la secuencia indicada en el diagrama de flujo o en el diagrama. Por ejemplo, dependiendo de las funciones involucradas, dos pasos o dos bloques que se muestran uno tras otro, en la práctica se pueden ejecutar al mismo tiempo, o en ocasiones los bloques se ejecutan en una secuencia inversa.

- 5
 - 10
 - 15
- Las personas con un conocimiento normal de la técnica pueden ser conscientes de que, en combinación con los ejemplos descritos en los modos de realización que se divulgan en esta memoria descriptiva, las unidades y los pasos de algoritmo se pueden implementar mediante un hardware electrónico, o una combinación de software para ordenador y hardware electrónico. El que las funciones se lleven a cabo mediante hardware o software depende de las aplicaciones particulares y las condiciones de restricción de diseño de las soluciones técnicas. Los expertos en la técnica pueden utilizar distintos métodos para implementar las funciones descritas para cada aplicación particular, aunque no se debería considerar que la implementación se encuentra fuera del alcance de la presente invención.

REIVINDICACIONES

1. Un método para acceder a un terminal de interconexión de componentes periféricos exprés, PCIe, (130), en donde un procesador (110) de un sistema informático se conecta al terminal PCIe (130) a través de un bus PCIe (140), comprendiendo el método:

5 recibir, por parte del procesador (110), una instrucción de acceso, en donde la instrucción de acceso le indica al procesador (110) que acceda al terminal PCIe (130) utilizando un proxy de acceso (160);

enviar, por parte del procesador (110), una solicitud de acceso al proxy de acceso (160) de acuerdo con la instrucción de acceso, en donde la solicitud de acceso le indica al proxy de acceso (160) que acceda al terminal PCIe (130); y

10 determinar, por parte del procesador (110), una tarea correspondiente a que se complete la solicitud de acceso después de recibir, desde el proxy de acceso (160), un mensaje de respuesta que indique que se ha recibido la solicitud de acceso enviada por el procesador;

15 recibir, por parte del procesador (110), desde el proxy de acceso, un primer mensaje de notificación que indique que el acceso al terminal PCIe se ha ejecutado con éxito y recibir un resultado del acceso cuando se recibe el primer mensaje de notificación.

2. El método de acuerdo con la reivindicación 1, en donde recibir la instrucción de acceso comprende recibir,

por parte del procesador, la instrucción de acceso generada por un módulo de control del terminal PCIe de acuerdo con una interfaz predeterminada para acceder al terminal PCIe, en donde la interfaz predeterminada para acceder al terminal PCIe apunta al proxy de acceso; o

20 recibir la instrucción de acceso comprende recibir, por parte del procesador, la instrucción de acceso generada por el sistema operativo anfitrión de acuerdo con una interfaz predeterminada de acceso al terminal PCIe, en donde la interfaz predeterminada de acceso al terminal PCIe apunta al proxy de acceso.

3. El método de acuerdo con una cualquiera de las reivindicaciones 1-2, en donde el proxy de acceso (160) es un controlador de acceso directo a memoria, DMA,

25 en donde enviar, por parte del procesador (110), la solicitud de acceso al proxy de acceso (160) de acuerdo con la instrucción de acceso comprende:

enviar, por parte del procesador (130), una petición de migración de datos al controlador de DMA de acuerdo con la instrucción de acceso; y

30 en donde la petición de migración de datos indica al controlador de DMA que migre los datos especificados desde una memoria del terminal PCIe (130) hasta una memoria (120) del sistema informático o que migre los datos desde la memoria (120) del sistema informático hasta la memoria del terminal PCIe.

35 4. El método de acuerdo con la reivindicación 3, en donde la instrucción de acceso indica, además, que la instrucción de acceso es una operación de lectura, en donde la instrucción de acceso indica, además, una dirección de origen de la operación de lectura y la longitud de la operación de lectura, y en donde enviar una petición de migración de datos al controlador de DMA de acuerdo con la instrucción de acceso comprende:

recibir, por parte del procesador (110), la dirección de destino de la operación de lectura reservada por la memoria (120) del sistema informático; y

40 enviar, por parte del procesador (110), la petición de migración de datos al controlador de DMA, en donde la petición de migración de datos incluye la dirección de origen de la operación de lectura, la dirección de destino de la operación de lectura y la longitud de la operación de lectura, con el fin de indicarle al controlador de DMA que transfiera datos de la longitud indicada para la operación de lectura desde la dirección de origen de la operación de lectura a la dirección de destino de la operación de lectura.

45 5. El método de acuerdo con la reivindicación 4, en donde la instrucción de acceso indica, además, que la instrucción de acceso es una operación de escritura, en donde la instrucción de acceso indica, además, una dirección de origen de la operación de escritura, una dirección de destino de la operación de escritura, y una longitud de la operación de escritura, en donde enviar la petición de migración de datos al controlador de DMA de acuerdo con la instrucción de acceso comprende enviar, por parte del procesador (110), la petición de migración de datos al controlador de DMA, y en donde la petición de migración de datos incluye la dirección de origen de la operación de escritura, la dirección de destino de la operación de escritura y la longitud de la operación de escritura, con el fin de indicarle al controlador de DMA que transfiera datos de la longitud indicada para la operación de escritura desde la dirección de origen de la operación de escritura a la dirección de destino de la operación de escritura.

6. El método de acuerdo con una cualquiera de las reivindicaciones 1-5, que comprende, además:

recibir, por parte del procesador (110), un segundo mensaje de notificación enviado por el proxy de acceso (160), en donde el segundo mensaje de notificación indica que el proxy de acceso (160) no ha conseguido acceder al terminal PCIe (130); y

5 determinar, por parte del procesador (110), el motivo por el que el proxy de acceso (160) no ha conseguido acceder al terminal PCIe (130); y

interrumpir, por parte del procesador (110), el acceso al terminal PCIe (130) cuando el motivo del fallo de acceso es que el terminal PCIe se ha desconectado de forma anómala.

7. Un ordenador, que comprende:

un bus PCIe; y

10 un procesador (110) acoplado al terminal PCIe (130) a través del bus PCIe (140);

en donde el procesador (110) se configura y diseña para realizar un método cualquiera de acuerdo con las reivindicaciones 1-6.

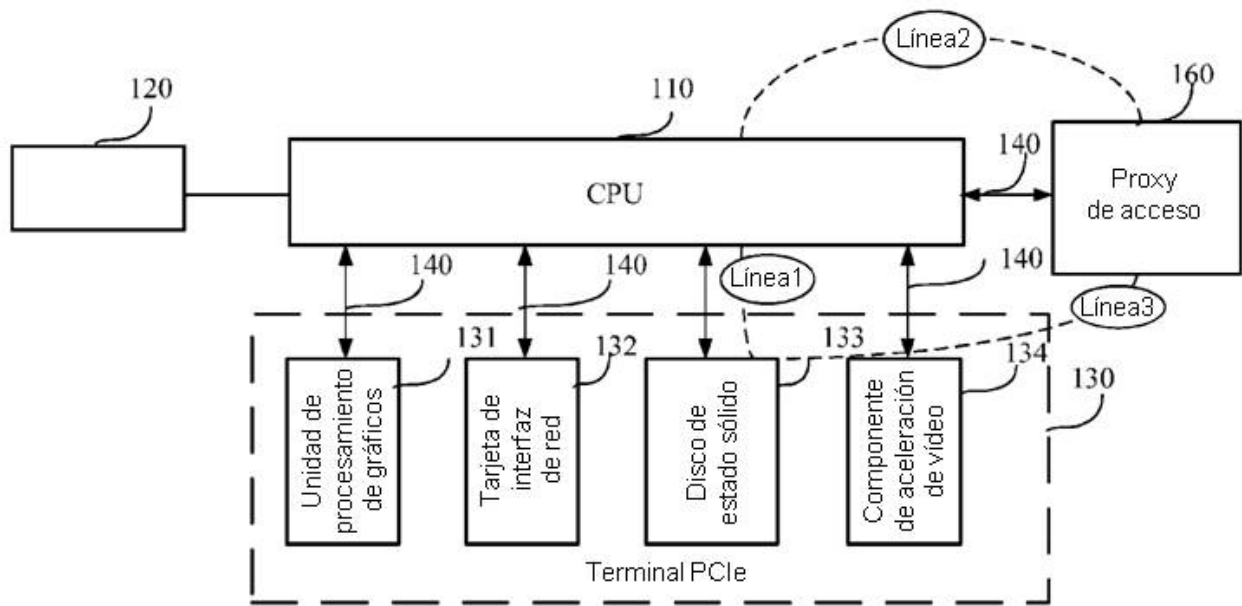


FIG. 1

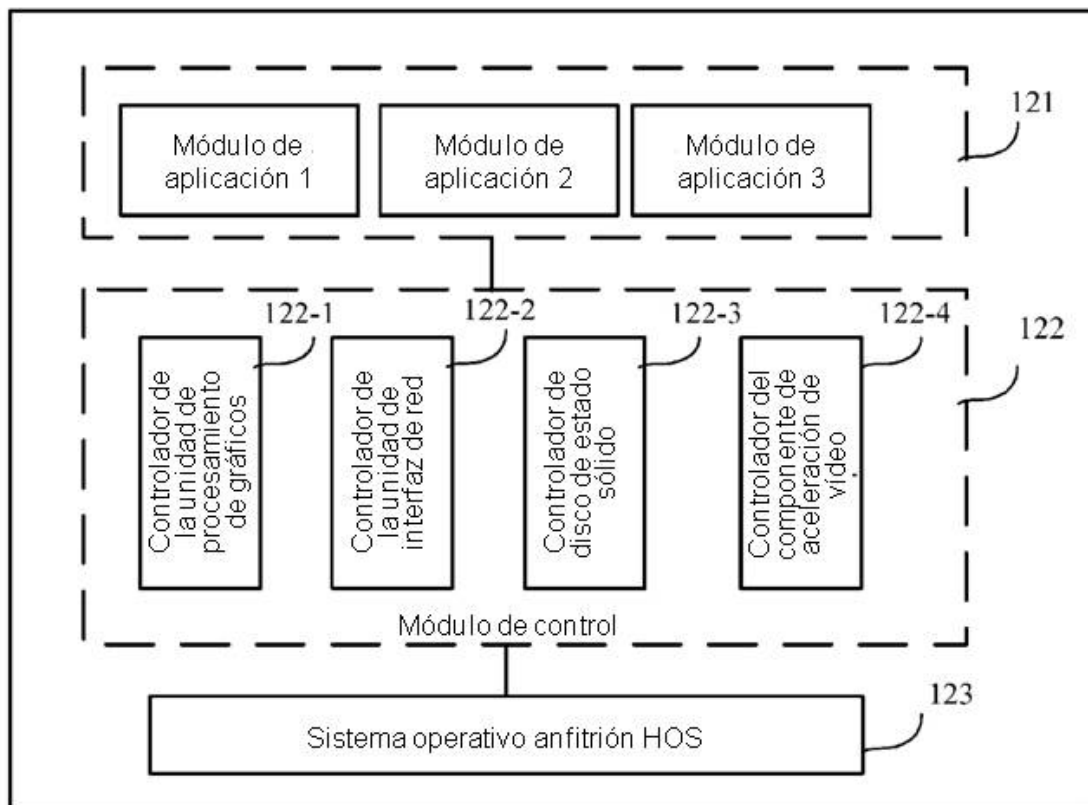


FIG. 2

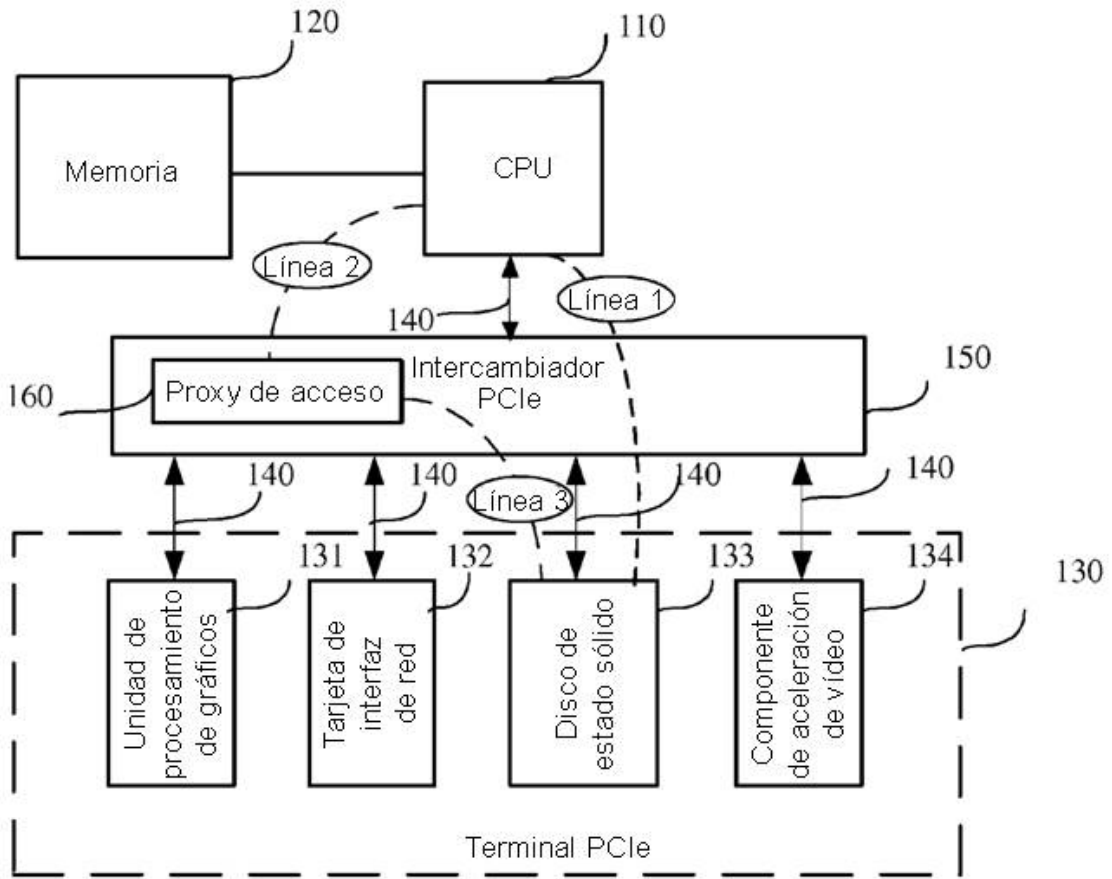


FIG. 3

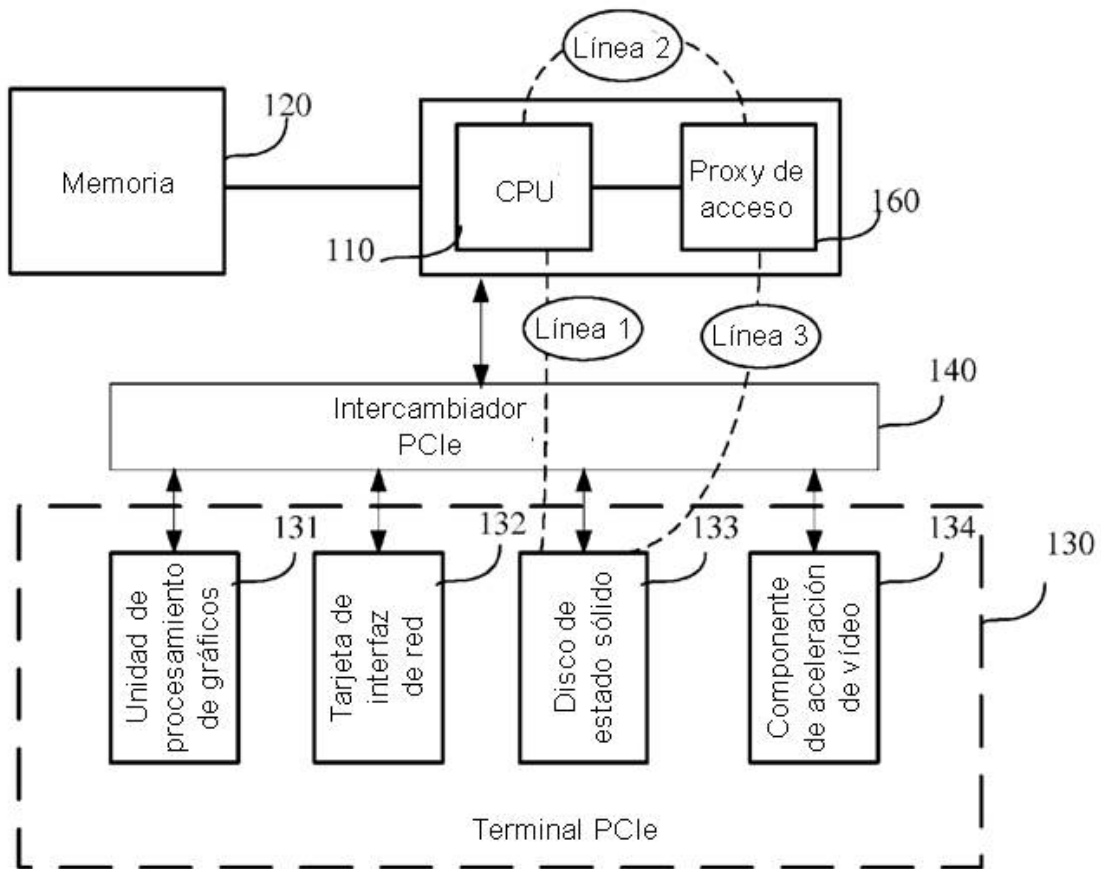


FIG. 4

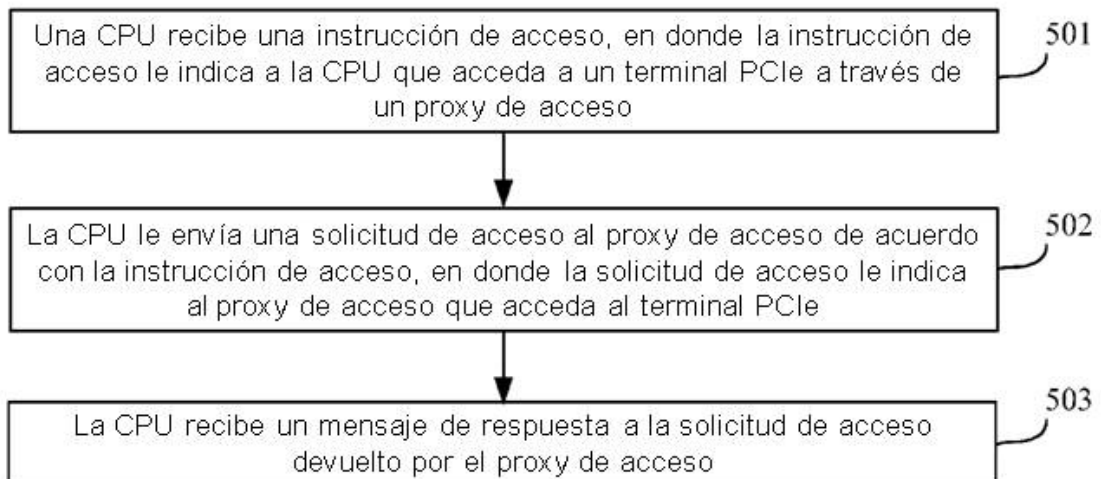


FIG. 5

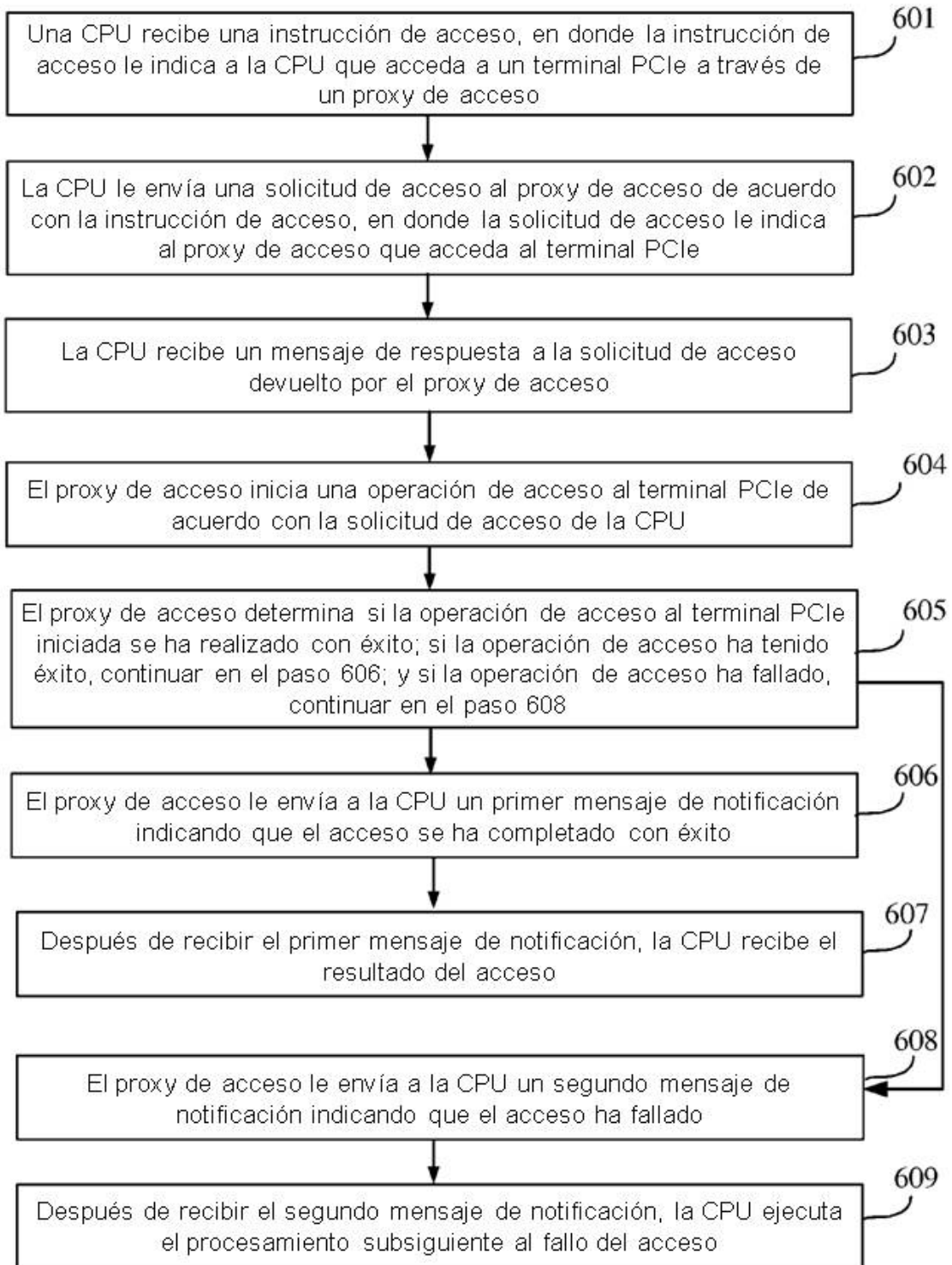


FIG. 6

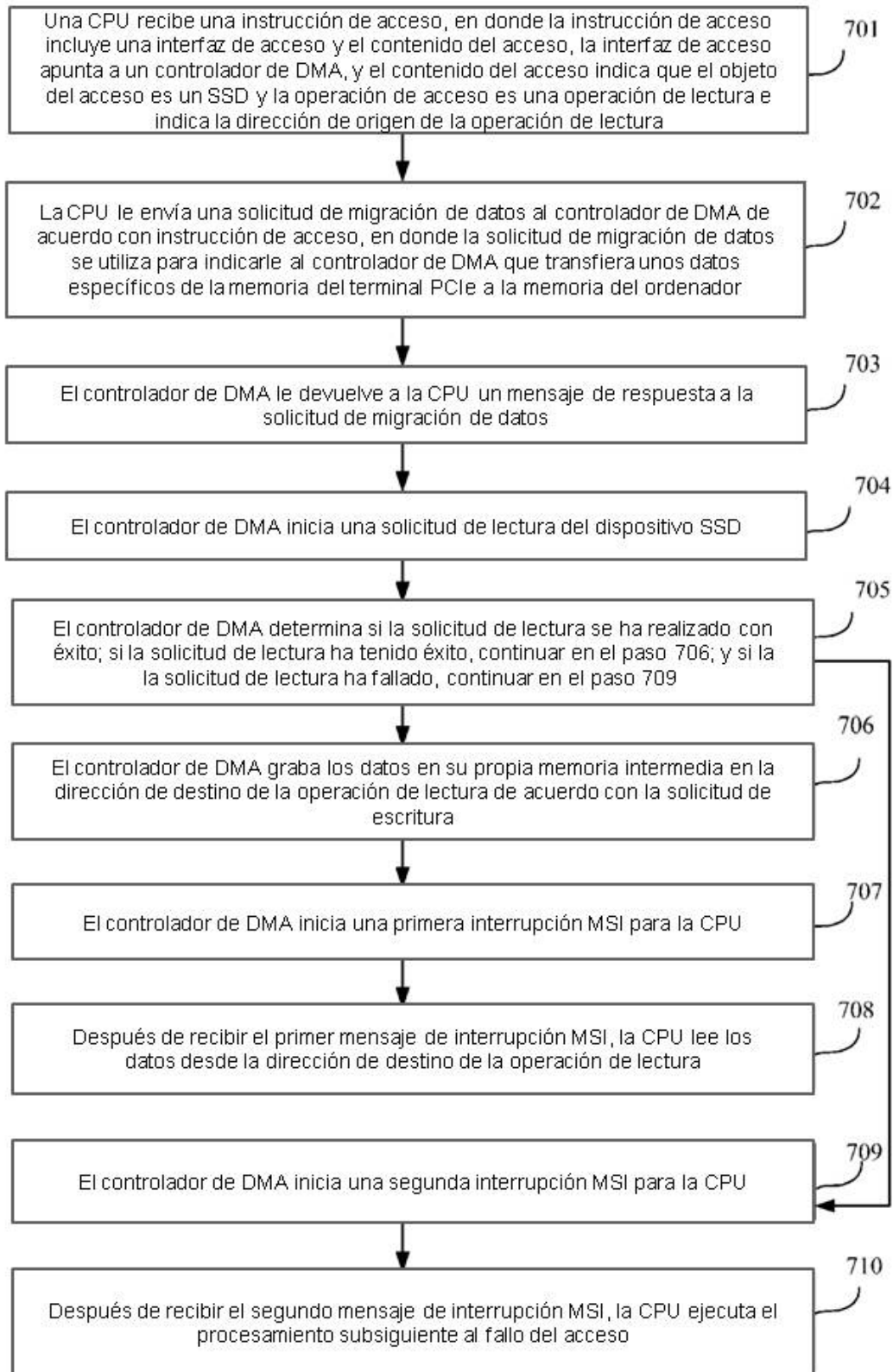


FIG. 7

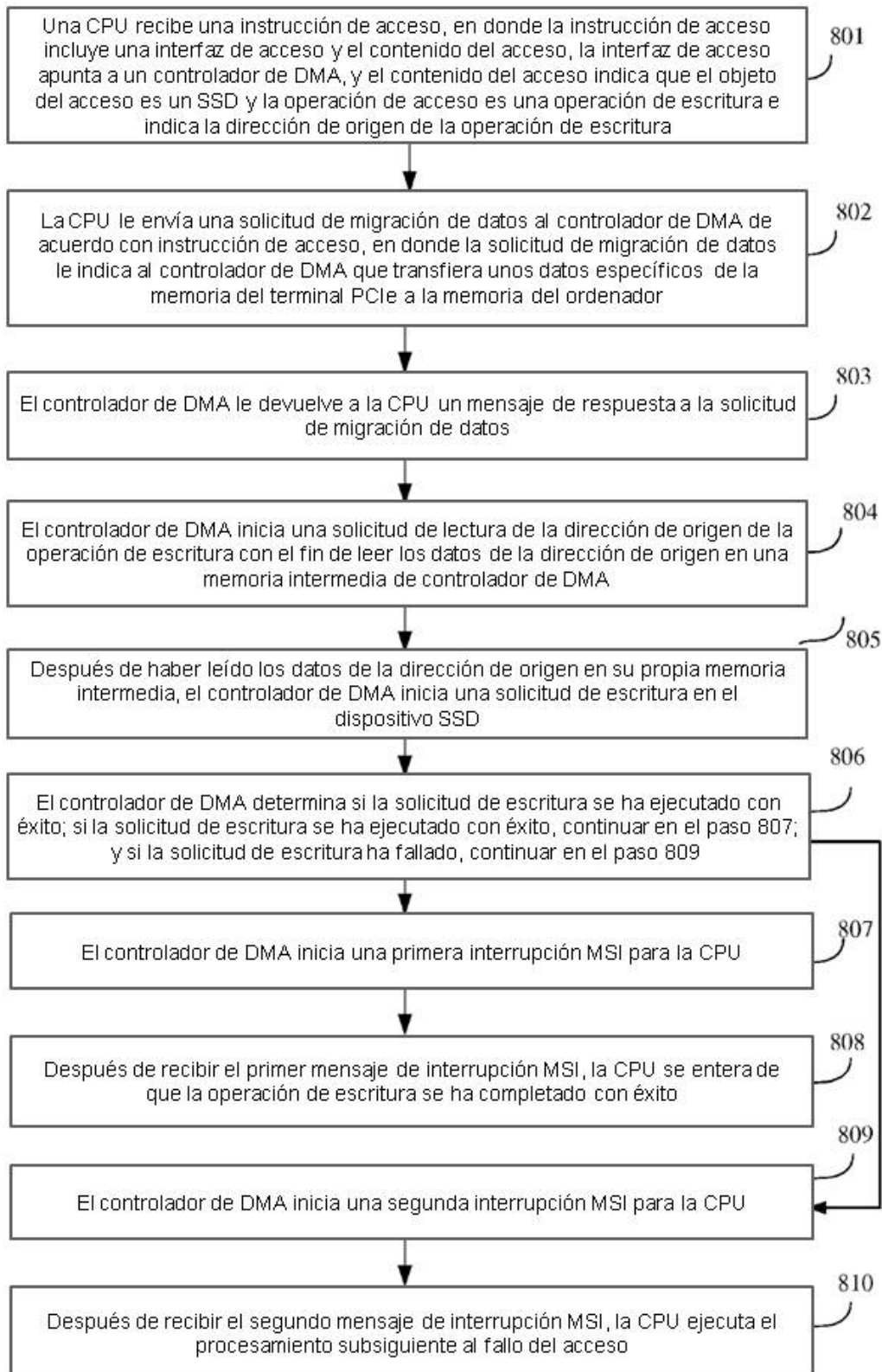


FIG. 8

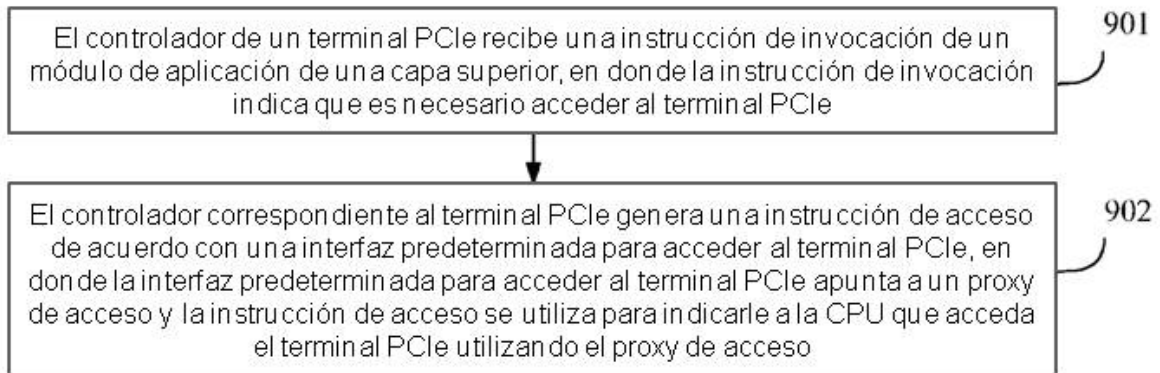


FIG. 9

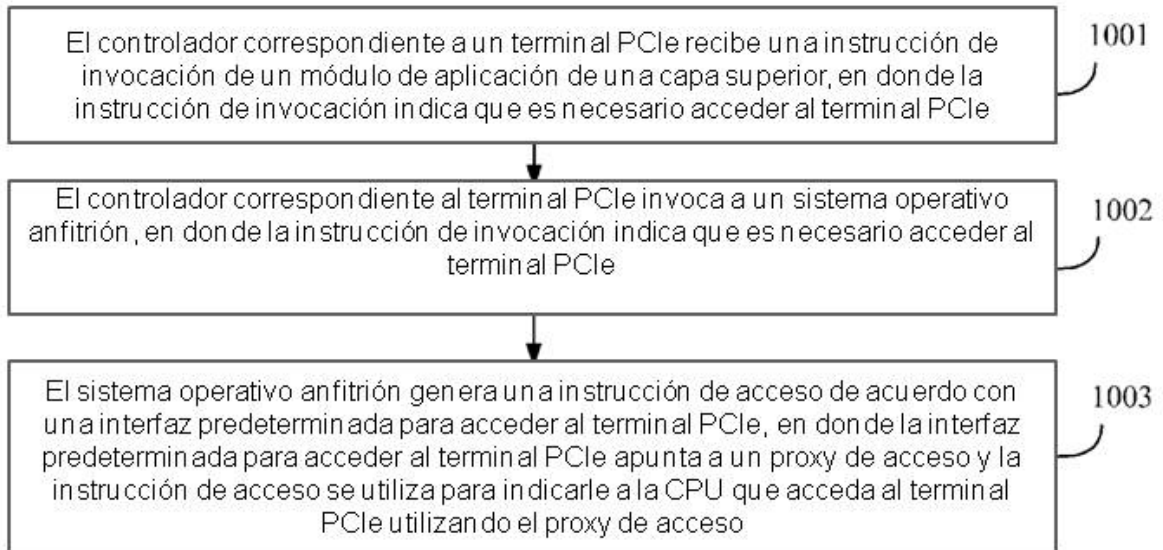


FIG. 10

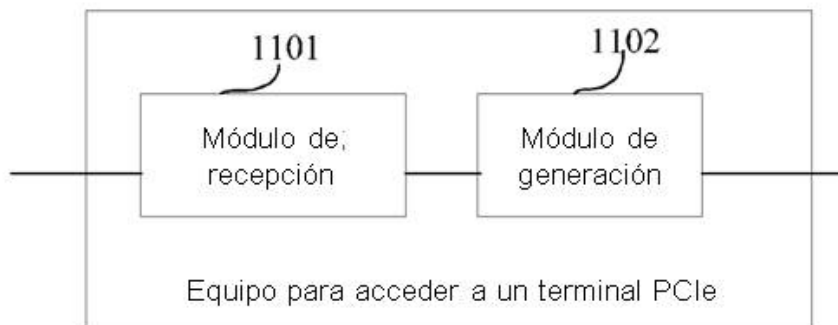


FIG. 11

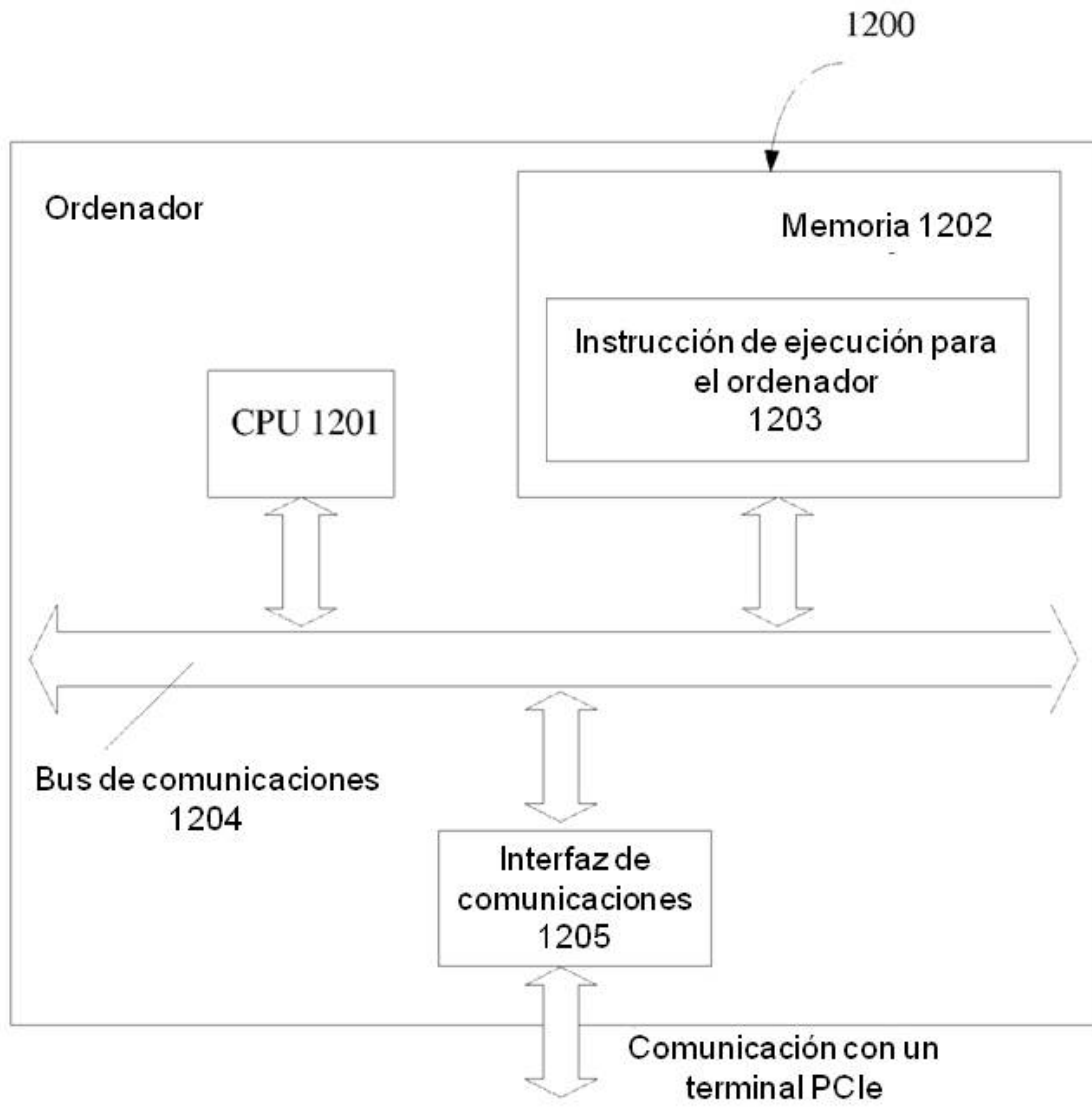


FIG. 12