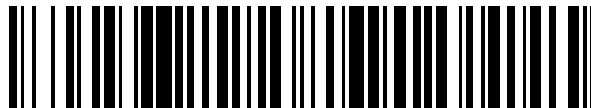


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 752 041**

51 Int. Cl.:

G06F 13/24 (2006.01)

G06F 9/48 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **08.10.2013 PCT/FR2013/052398**

87 Fecha y número de publicación internacional: **24.04.2014 WO14060679**

96 Fecha de presentación y número de la solicitud europea: **08.10.2013 E 13785528 (4)**

97 Fecha y número de publicación de la concesión europea: **14.08.2019 EP 2909729**

54 Título: **Procedimiento y dispositivo de tratamiento de interrupciones en un sistema multiprocesador**

30 Prioridad:

16.10.2012 FR 1259859

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

02.04.2020

73 Titular/es:

**BULL SAS (100.0%)
Rue Jean Jaurès
78340 Les Clayes-sous-Bois, FR**

72 Inventor/es:

**MENYHART, ZOLTAN;
TEMPORELLI, FRÉDÉRIC y
JEAUGEY, SYLVAIN**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 752 041 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

Procedimiento y dispositivo de tratamiento de interrupciones en un sistema multiprocesador

La presente invención se refiere a la gestión de los procesos en los sistemas de tratamiento de información y, más particularmente, a un procedimiento y un dispositivo de tratamiento de interrupciones en un sistema informático multiprocesador.

Los sistemas informáticos modernos están generalmente equipados con controladores de periféricos denominados autónomos, es decir, que los procesadores no tienen que formular peticiones para las operaciones de entrada/salida y los controladores de los periféricos ejecutan estas operaciones sin que los procesadores tengan que intervenir. Cuando se finaliza una operación, el controlador de periféricos en cuestión envía una señal al procesador a través de una interrupción.

Cabe recordar aquí que una interrupción es un suceso que conlleva la parada temporal de la ejecución normal de un proceso por un procesador, con el fin de ejecutar otro programa denominado rutina de interrupción. Una interrupción puede ser de hardware o software.

Para mejorar sus rendimientos, los sistemas informáticos modernos comprenden varios procesadores, a veces centenares de procesadores. Según la arquitectura implementada, una aplicación puede descomponerse en tareas elementales, o procesos, cada una tratada por un procesador o se pueden ejecutar en paralelo varias aplicaciones, cada una en uno o varios procesadores. Se trata de sistemas informáticos multiprocesadores. Una memoria local está generalmente asociada a cada procesador para optimizar el acceso a los datos tratados.

Los procesos implementados en los sistemas informáticos multiprocesadores son a menudo repartidos de tal manera que cada proceso esté localizado en un procesador particular con el fin de que éste sea el mismo procesador que utiliza un subconjunto de datos (tratados por un proceso particular) y que éste sea una misma memoria que alberga estos datos.

Cuando se migra un proceso, es decir, que se desplaza de un procesador a otro, los rendimientos del sistema resultan afectados, ya que es necesario desplazar el contexto de ejecución del proceso. Además, después de la migración, el proceso será, en sí mismo, menos eficiente debido al hecho de que deberá utilizar datos almacenados en una zona de memoria a distancia (memoria local del procesador inicial).

Por otro lado, en los sistemas informáticos modernos, las interrupciones generadas por un controlador de periféricos autónomo son transmitidas en forma de mensajes a los procesadores que están en el origen de las operaciones de entrada/salida. Sin embargo, cuando un controlador de periféricos autónomo es utilizado por varios procesadores, no es siempre el procesador en el origen de una operación de entrada/salida el que recibe la interrupción correspondiente.

Cuando una interrupción no es recibida por el procesador que ha iniciado la operación de entrada/salida correspondiente, es necesario un tratamiento suplementario. Este tratamiento puede, en particular, consistir en la transmisión de una interrupción al procesador que ha iniciado realmente la operación de entrada/salida en cuestión. Con tales fines, un mecanismo de interrupción entre procesadores (denominado IPE o *Inter-Processor Interrupt* en terminología anglosajona). Alternativamente, este tratamiento puede consistir en transferir el proceso que haya llamado a las funciones de entrada/salida al procesador que haya recibido la interrupción.

La solicitud de patente US 2006/0085582 describe un sistema multiprocesador en el cual una unidad de tratamiento, que es la fuente de una notificación de interrupción, transmite la notificación a una unidad de tratamiento destinataria, comprendiendo la notificación un identificador del destinatario. Un controlador analiza la notificación hace referencia a una tabla de correspondencia entre los identificadores de procesos y los procesadores que efectúan los procesos, determina el destinatario y le transmite la notificación de interrupción.

Sea cual sea el tratamiento implementado, hay una influencia no despreciable en los rendimientos del sistema informático.

Existe por tanto una necesidad de mejorar los rendimientos del tratamiento de las interrupciones en los sistemas informáticos multiprocesadores que utilizan controladores de periféricos autónomos compartidos.

La invención permite resolver al menos uno de los problemas descritos anteriormente.

La invención tiene así por objeto un procedimiento de tratamiento de interrupciones en un controlador de periféricos autónomo para un sistema informático multiprocesador, comprendiendo este procedimiento las etapas siguientes,

- recepción de una orden para la ejecución de una operación de entrada/salida y de un identificador del dispositivo del cual se recibe dicha orden;

- ejecución de una operación de entrada/salida vinculada a dicha orden recibida;

- obtención de dicho identificador del dispositivo del cual se recibe dicha orden;

- identificación de un destinatario de una interrupción correspondiente a la ejecución de dicha operación de entrada/salida vinculada a dicha orden recibida, siendo identificado dicho destinatario en función de dicho identificador obtenido del dispositivo del cual se recibe dicha orden; y

- 5 - transmisión a dicho destinatario identificado de dicha interrupción correspondiente a la ejecución de dicha operación de entrada/salida vinculada a dicha orden recibida.

El procedimiento según la invención permite así optimizar los rendimientos de un sistema informático multiprocesador, evitando en particular la implementación de un mecanismo de interrupción entre procesadores y/o una migración de procesos.

- 10 Según una realización particular, el procedimiento comprende además una etapa de memorización de dicho identificador del dispositivo del cual se recibe dicha orden. Tal etapa permite encontrar fácilmente un dispositivo al cual se debe direccionar una interrupción.

- 15 Siempre según una realización particular, el procedimiento comprende además una etapa de transmisión de dicho identificador del dispositivo del cual se recibe dicha orden, comprendiendo dicha etapa de obtención de dicho identificador del dispositivo del cual se recibe dicha orden una etapa de recepción de dicho identificador transmitido. Tal etapa permite conservar un enlace directo entre una orden o su ejecución y un identificador del dispositivo del cual se recibe esta orden, facilitando la identificación del destinatario de una interrupción.

Siempre según una realización particular, dicha etapa de identificación de un destinatario comprende una etapa de acceso a una tabla de mensajes de interrupciones.

- 20 Siempre según una realización particular, dicho destinatario se identifica en función de un número de vector de interrupción.

- 25 Siempre según una realización particular, dicha tabla de mensajes de interrupciones comprende al menos un identificador de destinatario de una interrupción, estando asociado dicho al menos un identificador de destinatario a un par que comprende un número de vector de interrupción y un identificador de un dispositivo del cual se puede recibir una orden para la ejecución de una operación de entrada/salida.

Dicha tabla de mensajes de interrupciones es, por ejemplo, una tabla de mensajes de tipo MSI-X modificada.

- 30 La invención tiene igualmente por objeto un programa de ordenador que comprende instrucciones adaptadas para la implementación de cada una de las etapas del procedimiento descrito anteriormente, cuando dicho programa se ejecuta en un ordenador, así como un controlador de periféricos autónomo que comprende medios adaptados para la implementación de cada una de las etapas del procedimiento descrito anteriormente y un sistema informático que comprende dicho controlador de periféricos autónomo, una pluralidad de procesadores y una pluralidad de periféricos, permitiendo dicho controlador de periféricos autónomo a cada procesador de dicha pluralidad de procesadores acceder a cada periférico de dicha pluralidad de periféricos.

- 35 Las ventajas ocasionadas por este programa de ordenador, este controlador de periféricos autónomo y este sistema informático son similares a las mencionadas anteriormente.

Otras ventajas, objetivos y características de la presente invención destacan de la descripción detallada siguiente, realizada a título de ejemplo no limitativo, con relación a los dibujos anexos, en los cuales:

- 40 - la figura 1 representa esquemáticamente una parte de un sistema informático multiprocesador en el cual se puede implementar la invención;
- la figura 2 ilustra un ejemplo de una tabla de mensajes MSI-X estándar para administrar interrupciones;
- la figura 3 ilustra un ejemplo de una tabla de mensajes MSI-X modificada para administrar interrupciones según tal realización particular;
- la figura 4 ilustra ciertas etapas de un ejemplo de algoritmo implementado en un controlador de periféricos autónomo conforme a la realización descrita con referencia la figura 3; y
- 45 - la figura 5 ilustra esquemáticamente un ejemplo de arquitectura de un controlador de periféricos autónomo adaptado para implementar una realización de la invención, en particular el algoritmo descrito con referencia a la figura 4.

La figura 1 representa esquemáticamente una parte 100 de un sistema informático multiprocesador en el cual se puede implementar la invención.

Como se ilustra, el sistema informático 100 comprende aquí dos procesadores 105-1 y 105-2, por ejemplo procesadores de núcleos múltiples. A título de ilustración, el procesador 105-1 comprende cuatro núcleos referidos como 110-11 a 110-14.

5 Una memoria local, típicamente una memoria de tipo RAM (sigla de *Random Access Memory* en terminología anglosajona) está asociada a cada procesador, que puede leer y escribir en la misma datos a través de un bus de datos. Así, una memoria local, que comprende aquí los módulos 115-11 a 115-1n, está asociada al procesador 105-1.

10 El sistema 100 informático comprende además un controlador 120 de periféricos autónomo conectado a varios procesadores, aquí, a los procesadores 105-1 y 105-2 a través de un bus de comunicación. El controlador 120 de periféricos autónomo está igualmente conectado a periféricos tales como discos 125 duros u otros periféricos 130 tales como tarjetas de red, por ejemplo tarjetas conforme al estándar *Infiniband* (*Infiniband* es una marca).

15 Según una realización particular, las órdenes dirigidas a las operaciones de entrada/salida, emitidas por los procesadores conectados a un mismo controlador de periféricos autónomo, comprenden una identificación del procesador que inició la orden. Tal identificación es memorizada por el controlador de periféricos autónomo. Así, después de la ejecución de una operación de entrada/salida, el controlador de periféricos autónomo puede identificar el procesador que inició esta operación y direccionar al mismo directamente una interrupción.

Esta realización comprende por tanto dos partes:

20 - la adición, a cada orden de entrada/salida, de una identificación apropiada al iniciador de la orden. Con tales fines, los bloques de órdenes de operaciones de entrada/salida son, con preferencia, extendidos para integrar una identificación del procesador en el origen de la petición; y

- la obtención de la identificación de un procesador que inició una operación de entrada/salida después de la ejecución de esta última para transmitir al mismo una interrupción.

25 Tal realización puede implementarse en un sistema informático multiprocesador en el cual los periféricos están conectados a un controlador de periféricos autónomo a través de un bus de comunicación conocido bajo el nombre de PCIe (sigla de *Peripheral Component Interconnect express* en terminología anglosajona). Permite, en particular, el intercambio de mensajes de interrupción de tipo MSI (sigla de *Message Signaled Interrupts* en terminología anglosajona) y MSI-X (sigla de *Message Signaled Interrupts - eXtended* en terminología anglosajona).

30 Se recuerda aquí que según el estándar PCIe, las interrupciones son transmitidas en forma de mensajes a través de un bus de datos, sin utilizar señales dedicadas. Los mensajes MSI-X son mensajes de interrupción de entrada/salida que pueden ser transmitidos a varios procesadores. El principio general consiste en escribir un dato particular, de tamaño limitado, en una dirección predeterminada. Cuando se debe emitir una interrupción, se utiliza su referencia para identificar al destinatario. Por consiguiente, como se describió anteriormente, para una interrupción particular, por ejemplo la recepción de un dato por un periférico, puede haber varios destinatarios, típicamente varios procesadores.

35 En otras palabras, el estándar PCIe utiliza una tabla de interrupciones indexada por números de vector correspondientes a tipos de sucesos detectados a nivel del hardware del sistema informático multiprocesador. A título de ilustración, estos tipos de sucesos de hardware pueden ser los siguientes:

- el final normal de una operación de lectura de datos en un periférico tal como un disco duro; y

- la recepción de un dato por un periférico tal como una tarjeta de red.

40 Cuando se debe señalar una interrupción, su destino es identificado con la ayuda de una tabla de mensajes MSI-X en función de un número de vector.

La figura 2 ilustra un ejemplo de una tabla de mensajes MSI-X estándar para administrar interrupciones.

45 Como se ilustra, la tabla 200 de mensajes MSI-X comprende, de manera estándar y bajo una forma simplificada, dos columnas 205 y 210 que representan un número de vector de interrupción, es decir, un tipo de suceso particular, y un destino de interrupción correspondiente, respectivamente. Así, por ejemplo, la línea 215 corresponde al vector de interrupción que tiene la referencia 1, cuyo destino de interrupción es *Destino_interrupción_1*. A título de ilustración, la interrupción contemplada en la línea 215 puede referirse a la recepción de un dato por una tarjeta de red y la transmisión de la interrupción correspondiente a los procesadores del sistema informático en cuestión.

50 Así, como se ilustra por la flecha 220, la identificación de un destino de una interrupción se determina en función de un número de vector, es decir, de un tipo de suceso de hardware.

Según una realización particular, se añade una segunda dimensión a la tabla de mensajes MSI-X para elegir un destino de una interrupción, no únicamente en función de un número de vector, sino igualmente en función de un

identificador del dispositivo, típicamente un procesador, en el origen de la orden que haya conducido al suceso de hardware correspondiente a la interrupción a señalar.

La figura 3 ilustra un ejemplo de una tabla de mensajes MSI-X modificada para administrar interrupciones según tal realización particular.

5 Como se ilustra, cada línea de la tabla 300 de mensajes MSI-X modificada corresponde a un número de vector, es decir, un tipo de suceso particular, y cada columna corresponde a un identificador de un dispositivo, típicamente un procesador, pudiendo estar en el origen de una orden que conduce a un suceso de hardware correspondiente a una interrupción a señalar.

10 Así, a partir de un número de vector y un identificador de un dispositivo en el origen de la orden que haya conducido al suceso de hardware correspondiente a la interrupción a señalar, es posible determinar un destino de la interrupción, siendo este destino, por naturaleza, más preciso que el obtenido únicamente a partir de un número de vector.

15 A título de ilustración, si el número de vector es el 2 y si el identificador del dispositivo en el origen de la orden que ha conducido al suceso de hardware correspondiente a la interrupción a señalar es 1, el destino de la interrupción es *Dest_inter_1,2*.

Como se ilustra por las flechas 305 y 310, la identificación de un destino de una interrupción se determina en función del número de vector, es decir, de un tipo de suceso, y de un identificador de un dispositivo, típicamente un procesador, pudiendo estar en el origen de una orden que conduce a un suceso de hardware correspondiente a una interrupción a señalar.

20 En otras palabras, un identificador de un dispositivo que puede estar en el origen de una orden que conduce a un suceso de hardware correspondiente a una interrupción a señalar, tratando este dispositivo la interrupción recibida, se utiliza, según esta realización, para añadir una segunda dimensión a la tabla de mensajes MSI-X y refinar el destino de la interrupción.

25 Una interrupción puede por tanto ser enrutada más rápidamente al dispositivo, típicamente al procesador, que trata la operación de entrada/salida en el origen de la interrupción. El tratamiento es así más rápido y perturba menos los tratamientos efectuados por los otros dispositivos del sistema informático.

La figura 4 ilustra ciertas etapas de un ejemplo de algoritmo implementado en un controlador de periféricos autónomo conforme a la realización descrita con referencia la figura 3.

30 Una primera etapa (etapa 400) tiene por objeto la recepción de una orden de una operación de entrada/salida (señalada con I/O para *Input/Output* en terminología anglosajona) y de un identificador del dispositivo, típicamente un procesador señalado con CPU (sigla de *Central Processing Unit* en terminología anglosajona), del cual se recibe la orden. La orden y el identificador pueden ser recibidos de forma distinta o conjunta (el identificador puede, por ejemplo, estar codificado en la orden).

El contexto de la orden recibida es por tanto memorizado (etapa 405), de manera local o a distancia.

35 La memorización del contexto tiene por objeto permitir encontrar el identificador del dispositivo del cual se recibe una orden cuando se ejecuta esta última y que conviene transmitir una interrupción. El contexto comprende, con preferencia, el identificador del dispositivo del cual se recibe la orden, así como la orden o una parte de la orden, un enlace sobre la orden o una parte de la orden o una referencia sobre la orden o una parte de la orden.

40 La memorización local del contexto puede, por ejemplo, comprender el almacenamiento de la orden y el del identificador del dispositivo del cual se recibe la orden en una memoria local. La memorización a distancia puede comprender, por ejemplo, el almacenamiento de la orden y el del identificador del dispositivo del cual se recibe la orden en una memoria a distancia. Alternativamente, o de manera complementaria, la memorización a distancia puede comprender la transmisión del identificador del dispositivo del cual se recibe la orden con uno de los datos y/o parámetros de ejecución de la orden (la transmisión del identificador de elemento en elemento permite conservar este identificador y por tanto memorizarlo).

45 La operación de entrada/salida correspondiente a la orden recibida se ejecuta a continuación (etapa 410).

Se efectúa a continuación una prueba para determinar si se debe transmitir una interrupción (etapa 415). Típicamente, se debe transmitir una interrupción cuando termina la ejecución de la operación de entrada/salida correspondiente a la orden recibida.

50 Si se debe transmitir una interrupción, se obtiene el identificador del dispositivo del cual se recibe la orden correspondiente a la interrupción a transmitir (etapa 420), por ejemplo a partir de la lectura del contexto memorizado con anterioridad que comprende este identificador. Este identificador puede igualmente ser recibido con datos resultantes de la ejecución de la orden.

En el transcurso de una etapa siguiente (etapa 425), se identifica el destinatario de la interrupción. Como se describió anteriormente con referencia a la figura 3, se identifica el destinatario ventajosamente a partir de una tabla de mensajes tal como una tabla de mensajes MSI-X modificada, a partir de un número de vector de la interrupción y de un identificador del dispositivo del cual se recibe la orden correspondiente a la interrupción a transmitir.

5 La interrupción se transmite entonces al dispositivo cuyo identificador se ha obtenido (etapa 430).

Naturalmente, se pueden tratar en paralelo varias operaciones de entrada/salida y varias transmisiones de interrupciones.

10 La figura 5 ilustra esquemáticamente un ejemplo de arquitectura de un controlador de periféricos autónomo adaptado para implementar una realización de la invención, en particular el algoritmo descrito con referencia a la figura 4.

Como se ilustra, el controlador 120 de periféricos autónomo comprende aquí un microcontrolador 500 (o CPU). El mismo aquí se conecta a:

- una primera memoria 505, por ejemplo una memoria de tipo RAM, que comprende registros adaptados para grabar variables y parámetros creados y modificados en el transcurso de la ejecución de programas;

15 - una segunda memoria 510, por ejemplo una memoria de tipo EEPROM (acrónimo de *Electrically-Erasable Programmable Read Only Memory* en terminología anglosajona), adaptada por ejemplo para almacenar programas que implementan el algoritmo descrito con referencia a la figura 4;

- una interfaz 515 de comunicación conectada a varios procesadores, adaptada para recibir órdenes de operaciones de entrada/salida y para transmitir interrupciones; y

20 - una interfaz 520 de comunicación conectada a los periféricos y que permite transmitir o recibir datos de periféricos.

El microcontrolador 500 controla y dirige la ejecución de instrucciones o porciones de código de software del o de los programas.

25 Naturalmente, para satisfacer las necesidades específicas, un experto en el campo de la invención podrá aplicar modificaciones en la descripción anterior. La presente invención no se limita a las realizaciones descritas, son posibles otras variantes y combinaciones de características.

La presente invención ha sido descrita e ilustrada en la presente descripción detallada con referencia a las figuras adjuntas. Sin embargo, la presente invención no se limita a las realizaciones presentadas. Se pueden deducir e implementar otras variantes y realizaciones por el experto en el campo de la invención tras la lectura de la presente descripción y de las figuras anexas.

30 En las reivindicaciones, el término "comprender" no excluye otros elementos u otras etapas. El artículo indefinido "un" no excluye el plural. Se pueden utilizar un solo procesador o varias otras unidades para implementar la invención. Las diferentes características presentadas y/o reivindicadas pueden combinarse ventajosamente. Su presencia en la descripción o en las reivindicaciones dependientes diferentes no excluye, de hecho, la posibilidad de combinarlas. Los signos de referencia no se deberán entender como limitativos del alcance de la invención.

35

REIVINDICACIONES

1. Procedimiento de tratamiento de interrupciones en un controlador de periféricos autónomo para un sistema informático multiprocesador, estando este procedimiento caracterizado porque comprende las etapas siguientes,

- 5 - recepción (400) de una orden para la ejecución de una operación de entrada/salida y de un identificador del dispositivo del cual se recibe dicha orden;
- ejecución (410) de dicha operación de entrada/salida vinculada a dicha orden recibida;
- obtención (420) de dicho identificador del dispositivo del cual se recibe dicha orden después de la ejecución de dicha operación de entrada/salida vinculada a dicha orden recibida;
- 10 - identificación (425) de un destinatario de una interrupción correspondiente a la ejecución de dicha operación de entrada/salida vinculada a dicha orden recibida, siendo identificado dicho destinatario en función de dicho identificador obtenido del dispositivo del cual se recibe dicha orden; y
- transmisión (430) a dicho destinatario identificado de dicha interrupción correspondiente a la ejecución de dicha operación de entrada/salida vinculada a dicha orden recibida,

15 según el cual dicha etapa de identificación de un destinatario comprende una etapa de acceso a una tabla de mensajes de interrupciones,

dicho destinatario es identificado en función de un número de vector de interrupción y dicha tabla de mensajes de interrupciones comprende al menos un identificador de destinatario de una interrupción, estando asociado dicho al menos un identificador de destinatario a un par que comprende un número de vector de interrupción y un identificador de un dispositivo del cual se puede recibir una orden para la ejecución de una operación de entrada/salida.

20 2. Procedimiento según la reivindicación 1, que comprende además una etapa de memorización (405) de dicho identificador del dispositivo del cual se recibe dicha orden.

25 3. Procedimiento según la reivindicación 1 o la reivindicación 2, que comprende además una etapa de transmisión de dicho identificador del dispositivo del cual se recibe dicha orden, comprendiendo dicha etapa de obtención de dicho identificador del dispositivo del cual se recibe dicha orden una etapa de recepción de dicho identificador transmitido.

4. Procedimiento según una cualquiera de las reivindicaciones 1 a 3, según el cual dicha tabla de mensajes de interrupciones es una tabla de tipo MSI-X modificada.

30 5. Programa de ordenador que comprende instrucciones adaptadas para la implementación de cada una de las etapas del procedimiento según una cualquiera de las reivindicaciones anteriores, cuando dicho programa se ejecuta en un controlador de periférico autónomo según la reivindicación 6.

6. Controlador de periféricos autónomo que comprende medios adaptados para la implementación de cada una de las etapas del procedimiento según una cualquiera de las reivindicaciones 1 a 4.

35 7. Sistema informático que comprende un controlador de periféricos autónomo según la reivindicación 6, una pluralidad de procesadores y una pluralidad de periféricos, permitiendo dicho controlador de periféricos autónomo a cada procesador de dicha pluralidad de procesadores acceder a cada periférico de dicha pluralidad de periféricos.

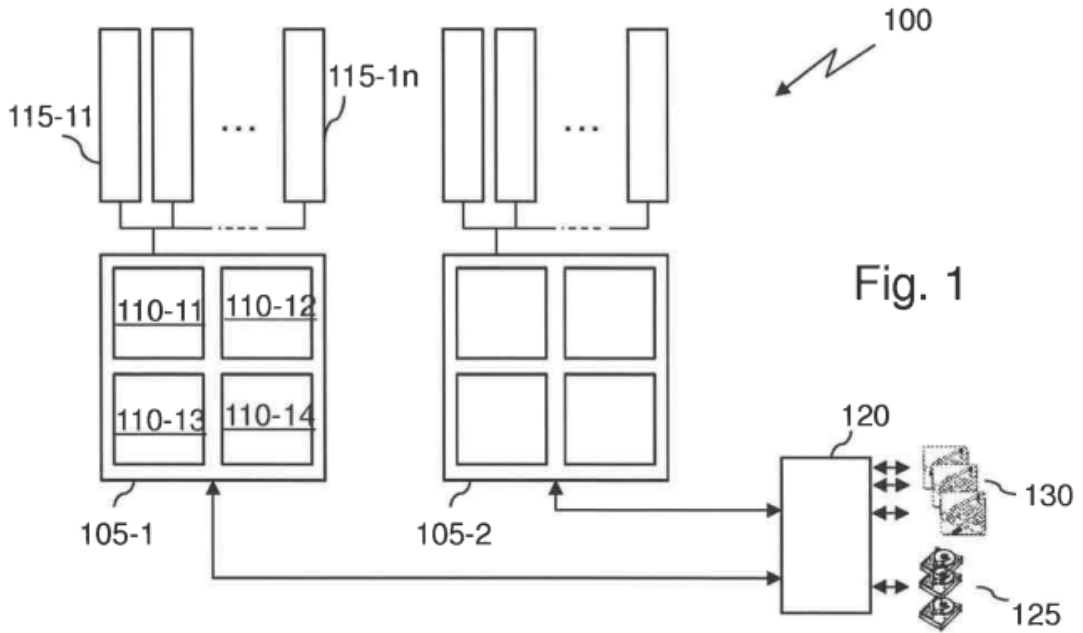


Fig. 1

205	210
0	Destino_interrupción_0
1	Destino_interrupción_1
2	Destino_interrupción_2
...	...
n	Destino_interrupción_n

Fig. 2

305	310			
	0	1	...	n
ID dispositivo / N° vector				
0	Dest_inter_0,0	Dest_inter_1,0	...	Dest_inter_n,0
1	Dest_inter_0,1	Dest_inter_1,1	...	Dest_inter_n,1
2	Dest_inter_0,2	Dest_inter_1,2	...	Dest_inter_n,2
...
n	Dest_inter_0,3	Dest_inter_1,3	...	Dest_inter_n,3

Fig. 3

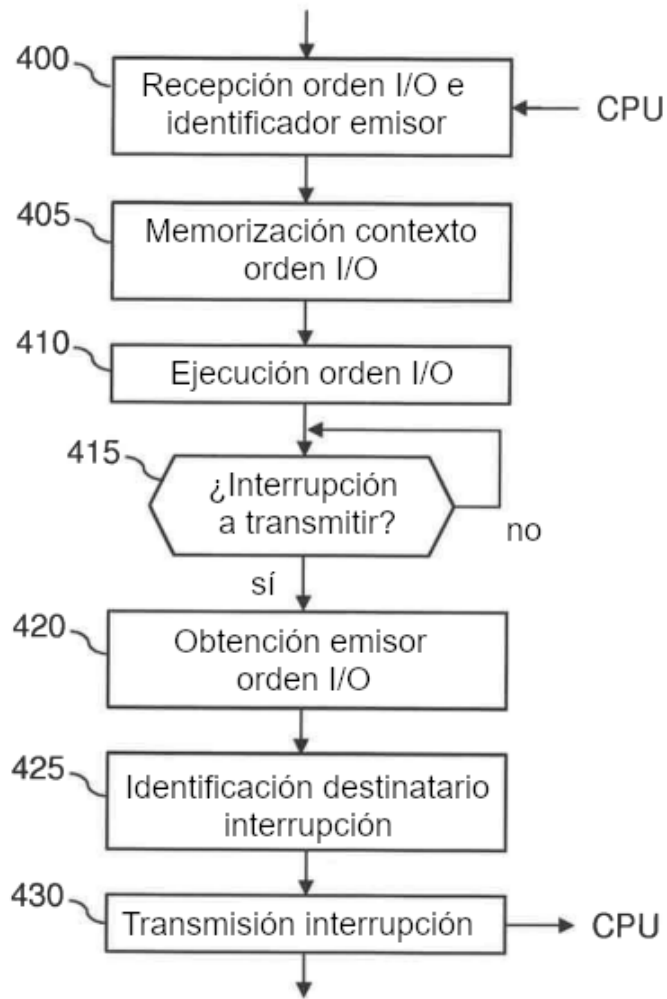


Fig. 4

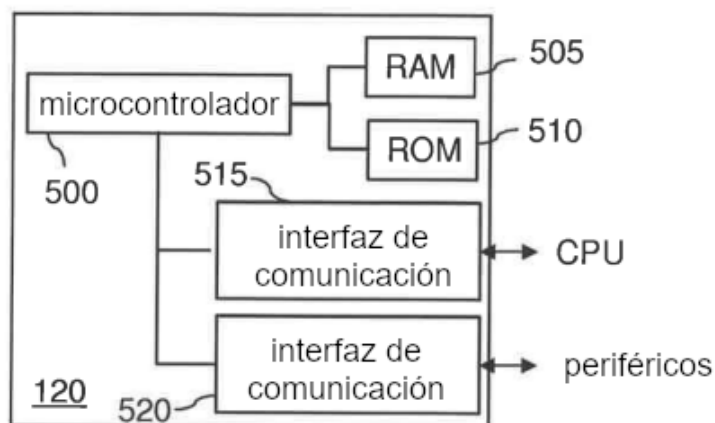


Fig. 5