



19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 304 641**

51 Int. Cl.:  
**G11C 16/10** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **04819996 .2**

86 Fecha de presentación : **06.12.2004**

87 Número de publicación de la solicitud: **1702339**

87 Fecha de publicación de la solicitud: **20.09.2006**

54 Título: **Método de software para emular una memoria EEPROM.**

30 Prioridad: **08.12.2003 PL 363945**

45 Fecha de publicación de la mención BOPI:  
**16.10.2008**

45 Fecha de la publicación del folleto de la patente:  
**16.10.2008**

73 Titular/es:  
**Advanced Digital Broadcast Polska Sp. Zo.o.  
ul. Trasa Polnocna 16  
65-119 Zielona Gora, PL  
Advanced Digital Broadcast Ltd.**

72 Inventor/es: **Przybylek, Piotr**

74 Agente: **Zea Checa, Bernabé**

**ES 2 304 641 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).

# ES 2 304 641 T3

## DESCRIPCIÓN

Método de software para emular una memoria EEPROM.

5 Método de emulación de la memoria EEPROM por vía de programa.

10 El objeto de la invención es el método de emulación de la memoria EEPROM por vía de programa en otra memoria permanente, p.ej. del tipo Flash. Esta manera se aplica a los sistemas en que, con el objetivo de reducir los costes de los equipos que utilizan la memoria permanente EEPROM, se puede utilizar la memoria existente, p.ej. del tipo Flash para la emulación de memoria EEPROM. La solución comporta la integración de la manera con los sistemas de equipos y los métodos de gestión de la memoria, p. ej. el seguimiento de cambios y la gestión del registro de datos. Además, representa el formato ejemplar del registro de datos en la memoria que se emula.

15 Por la solicitud europea de patente de nº EP0991081 es conocido el método de hardware de de la memoria en el mismo circuito integrado. Si el punto flaco es la limitación a un solo sistema y aun tipo de memoria así como la relación del tamaño de la memoria EEPROM que se emula respecto al tamaño de la memoria Flash (1:8) utilizada para este fin y el pequeño tamaño de la memoria emulada que es de hasta do 8 KB. La solución que se muestra funciona en cada uno de los sistemas de la memoria Flash que se utilizan actualmente y también se puede emular los grandes sistemas EEPROM de cualquier tamaño de hasta 64 KB que actualmente se utilizan.

20 Otra solución de hardware es el sistema descrito en el documento estadounidense de patente número US 5,651,128 en que se presenta la memoria compuesta de matrices de células y de los sistemas que permiten la programación del borrado del contenido de la memoria emulada. En el ejemplo de la ejecución de la emulación se utiliza la memoria Flash.

25 En el caso de las soluciones descritas, se realiza la emulación de la memoria EEPROM aplicando la solución de hardware en que el sistema utiliza la memoria Flash para almacenar los datos tanto previstos a ser registrados en la memoria Flash como en la EEPROM.

30 La diferencia entre los métodos presentados y la solución conforme a la invención consiste en que la emulación según la invención, al contrario a las soluciones conocidas, se realiza por vía de software. Gracias a eso, este método puede ser aplicado en cualquier equipo y en cualquier tipo de memoria Flash. Lo esencial de la solución consiste en la capa lógica del programa que controla el seguimiento y la utilización de la memoria emulada. Además, seleccionado este tipo de emulación se reduce tanto el coste de adquisición de los sistemas de memoria como la utilización de la superficie de las cartas imprimidas eliminando el más caro que la memoria Flash (computado respecto a 1 KB de memoria) sistema de memoria EEPROM el cual se suele montar como un módulo aparte. El beneficio adicional consiste en la reducción del coste, en la simplificación y la miniaturización del sistema electrónico que hasta ahora utilizaba la memoria EEPROM.

40 En la solución descrita según la invención, conforme a la reivindicación 1, también se ha presentado el formato que sirve de ejemplo de registro de los datos en la memoria que se emula.

45 El ejemplo descrito a continuación del descodificador de la televisión digital en que se ha utilizado la solución conforme a la invención debe ser considerado como una de las posibles aplicaciones del método de emulación de la memoria EEPROM según la invención. Cada equipo que requiere la memoria permanente del primer tipo, por ej. la memoria del tipo Flash y las memorias EEPROM, puede ser diseñado de tal forma que se pueda utilizar la emulación de la memoria EEPROM en la memoria del primer tipo según la invención y reduciendo al mismo tiempo tanto los costes de los sistemas como la cantidad de espacio requerido para los sistemas digitales montados en las cartas imprimadas. Eso permite diseñar los equipos universales en que el elemento clave es la independencia de la configuración del bloque de memoria donde puede producirse la combinación de la memoria Flash y de la EEPROM o de la Flash con la emulada EEPROM.

50 Suele ser posible la eliminación de memoria EEPROM sin necesidad de aumentar el tamaño de la memoria FLASH. El aumento de la memoria Flash resulta más ventajoso que el coste del sistema adicional EEPROM por ser el precio de la memoria EEPROM 32 veces más elevado por 1 KB de la capacidad respecto al precio de 1 KB de la memoria FLASH. La memoria adicional Flash puede ser utilizada no solo para la emulación de la memoria EEPROM.

55 Uno de los problemas con que se tropieza en la emulación de la memoria EEPROM en la memoria del tipo Flash es que la memoria Flash funciona de otra manera. Los datos deben sustituirse por sectores, no se puede cambiar los datos p. ej.: con bits lo que es el caso de la memoria EEPROM, y requiere la utilización del controlador (driver) el cual dando el funcionamiento conforme con las memorias típicas del tipo EEPROM operará en los sectores de la memoria Flash. Para reducir la cantidad de registros que se requieren en la memoria Flash, se ha utilizado la solución siguiente. Por eso, durante el funcionamiento típico de la memoria EEPROM, los datos se actualizan a menudo y en pequeña cantidad, p. ej. bit tras bit, en la emulación de la memoria EEPROM los datos se recogen y se registran al cabo de un cierto rato de tiempo como paquete de datos actualizados. Tal tiempo se puede definir por ej. en segundos o como la cantidad de los cambios realizados en los datos almacenados en la memoria interna. Adicionalmente, si resultase interesante, los datos registrados podrían ser comprimidos. Uno de los requisitos del funcionamiento de la emulación de la memoria permanente EEPROM consiste en garantizar que se tiene la copia adecuada de los datos

incluso si éstos no son los más actuales. Por ej. tal situación se produce en el caso de falta durante la operación de registro de los datos. Para garantizar la seguridad que se requiere de los datos de la memoria emulada EEPROM, se ha utilizado el servicio de la copia de los datos. Por eso, el sistema requiere la doble cantidad de espacio en la memoria Flash para emular la cantidad dada de la memoria EEPROM. Una vez programados los datos en los dos sectores de la memoria Flash, el segundo sector tiene que ser borrado. En el caso de la emulación de la memoria EEPROM cuyo tamaño es de 32 KB, se utilizan 2 sectores de la memoria Flash, que típicamente tienen el tamaño de 64 KB cada uno. Adicionalmente, existen tres tipos de las memorias intermedias en la memoria interna RAM. El primer de los sectores almacena siempre la imagen más actual de la memoria emulada EEPROM. El segundo sector almacena el último paquete de datos actualizados, y el último se utiliza opcionalmente para el almacenamiento del paquete de datos de actualización con vistas a la compresión.

La invención se presenta en el ejemplo de la ejecución en el dibujo en que la fig. 1 es el ejemplo de un equipo en forma del descodificador de la televisión digital en que se utiliza la manera conforme a la invención, la fig. 2 presenta el procedimiento de la puesta en marcha del sistema de emulación de la memoria, la fig. 3 - el procedimiento de registro de los datos a la memoria, la fig. 4 es un ejemplo del formato del paquete de los datos actualizados, la fig. 5 presenta el sector de la memoria Flash que comporta los datos emulados de la memoria EEPROM así como los paquetes de los datos actualizados, la fig. 6 presenta el formato del encabezamiento del paquete de datos actualizados, la fig. 7 - el procedimiento de registro de los datos actualizados, y la fig. 8 presenta el procedimiento de preparación de los datos para actualización con la posibilidad de la anulación del paquete anteriormente registrado, la fig. 9 presenta el procedimiento de selección del sector actual de la memoria permanente.

El receptor de la señal en la fig. 1 del dibujo el cual es el descodificador de la televisión digital se presenta para las necesidades de la invención en la versión simplificada con la sola reproducción de los elementos que se requieren para la presentación de la concepción de la invención. El descodificador de la televisión digital (101) comporta varios módulos. El más importante de ellos es el procesador (120) que rige el funcionamiento del equipo. Adicionalmente, según la invención, el procesador da servicio al bloque interno (121) que rige la emulación de la memoria EEPROM y el bloque de tratamiento de la señal (122). Al procesador está conectada la señal del bloque de recepción de señal (110). Además, el procesador tiene la posibilidad de intercambio bidireccional de datos a través de los interfaces externos (140). El descodificador de la televisión digital comporta también la memoria de varios tamaños que están conectadas bidireccionalmente con el procesador. Son por ejemplo la memoria permanente (es mejor que sea del tipo Flash) (150) y la interna RAM (160). En estas memorias se almacenan los programas que rigen el funcionamiento del descodificador de la televisión digital. Los bloques (130) y (131) permiten respectivamente emitir la señal de salida A/V y hacer posible la comunicación con los equipos externos de control, por ej. con el mando a distancia.

En la fig. 2 se ha presentado el proceso de iniciación de la emulación de la memoria EEPROM según la invención. Este procedimiento se realiza al arrancar el equipo que utiliza la emulación. El procedimiento empieza en el punto (201). Luego, en el paso (202), se selecciona el sector en que se leerán los datos con el sector actual. Uno de los dos sectores se selecciona en función de varios criterios. Estos criterios se presentan detalladamente en la fig. 9. Luego, en el paso (203) del procedimiento, se produce el borrado del contenido del segundo sector. Se ejecuta el punto (203) del procedimiento Si el sector adicional de la memoria contiene datos a pesar de que no debiera contenerlos. Tal situación puede producirse en el primer arranque de la emulación de la memoria EEPROM, si en este sector de la memoria antes se han encontrado otros datos o en el caso de que se haya producido una interrupción en el registro del primer paquete de los datos actualizados al segundo sector, por ej.: por la caída de la alimentación o también después de su finalización, pero siempre antes de borrar el contenido del primer sector. En tal caso, con el arranque siguiente de la emulación, existen algunos datos en dos sectores y por eso uno de ellos se borra. Luego, se inicia la lectura del sector seleccionado (204) de la memoria Flash, tomando en la memoria interna RAM la imagen primaria de la memoria emulada EEPROM. Por ej.: puede ser 32 KB de un sector de la memoria Flash cuyo tamaño suele ser de 64 KB. Si durante la lectura se produce un error, por ej. los datos no son correctos, en tal caso se borra el contenido del sector actual. En el punto (205) del procedimiento se toma el primer paquete de los datos actualizados en la memoria Flash y se comprueba si está válido, es decir si contiene los datos que son actuales (206). Si el paquete no ha sido anulado (punto 205), en el punto (207) del procedimiento se comprueba si los datos del paquete están comprimidos. De ser así, en el punto (208) éste se descomprime. En el caso contrario, el procedimiento pasa directamente al punto (209) en que los datos del paquete se registran en la memoria interna RAM, almacenando la imagen actual de la memoria EEPROM que se emula. El último punto del procedimiento es la comprobación (210) si en la memoria Flash aun están los datos por leer. Si es así, el procedimiento trata los paquetes siguientes de los datos actualizados conforme al algoritmo descrito. Si se ha registrado el último paquete, el procedimiento termina su funcionamiento. Después de haber realizado el procedimiento presentado en la fig. 2 en la memoria interna están disponibles todos los datos de la memoria emulada EEPROM.

En la fig. 3 se ha presentado el procedimiento del registro de los datos de la memoria emulada EEPROM en la memoria Flash. El procedimiento empieza en el punto (301) donde se preparan los datos para la actualización del contenido de la memoria Flash. Estos datos se registran también en la memoria intermedia de la memoria interna RAM la cual siempre almacena el contenido actual de la memoria emulada EEPROM, lo que se presentará detalladamente en la fig. 8 del dibujo. Luego, en el paso (302) del procedimiento, se comprueba si el tamaño del paquete de los datos actualizados es más grande que el valor fijado. Hay 64 bits en el ejemplo de la ejecución que se presenta. De ser así, el procedimiento pasa al punto (303) donde se comprime el paquete de los datos actualizados. En el ejemplo de la implantación se supone que el paquete de los datos actualizados cuyo tamaño es menor a 64 bits no

## ES 2 304 641 T3

se comprime por ser poco probable que se logre la reducción del tamaño del paquete. Luego, en el punto (304) del procedimiento, se comprueba el resultado de la compresión del punto (303). Esta comprobación determina si el beneficio resultante de la compresión de los datos es lo suficiente grande para sufragar el coste adicional de tiempo necesario para la decompresión del paquete en el momento de la iniciación del sistema de emulación de la memoria EEPROM conforme a la invención. Al ser así, se sigue tratando el paquete comprimido. En el caso contrario, no se trata del paquete no comprimido. Luego, el procedimiento de registro pasa al punto (305) en que se produce la comprobación si el sector actual de la memoria Flash tiene la cantidad apropiada de espacio para registrar de nuevo el paquete de los datos actualizados. Sin es así, en el punto (306), se registra el paquete de los datos actualizados, es decir los datos últimamente modificados en la memoria Flash, lo que se presenta con detalle en la fig. 7 en relación con la fig. 6. En el caso contrario, en el punto (307) del procedimiento, se cambia el sector actual por el segundo. Luego, se registra como nueva la imagen primaria de la memoria emulada EEPROM, y la imagen completa almacenada en la memoria intermedia de la memoria interna RAM, es decir la imagen actual de la memoria emulada EEPROM, en el punto (308) del procedimiento. Luego, en el punto (309) se borra el contenido del sector de la memoria Flash porque entonces es seguro que no se pierden los datos de la memoria emulada EEPROM. Ahora, en el sector de la memoria Flash nuevamente seleccionado, ya hay espacio adicional para los nuevos paquetes de los datos actualizados. El procedimiento de registro termina el funcionamiento en el punto (310).

En la fig. 4 se ha presentado el formato del paquete de los datos actualizados según el cual se registran las informaciones en la memoria Flash. El formato está compuesto de cuatro campos de que los primeros son el encabezamiento del paquete (401), el segundo es el tamaño del grupo de datos (402) (campo parece en el caso de los paquetes de datos actualizados en que aparecen los numerosos grupos de datos actualizados (inglés: Multi block patch), el tercero es el desplazamiento de los datos respecto a la dirección primaria (403). Este campo aparece solo en el caso de los paquetes no comprimidos de datos actualizados. El último campo son solo los datos (404). El paquete de los datos de actualización puede contener numerosos grupos de datos de los cuales cada grupo se registra bajo diferente dirección de la memoria.

En el caso de que el paquete de datos de actualización esté comprimido, no comporta el campo de desplazamiento, y el valor mismo del desplazamiento se lee tan solo después de la compresión del paquete. En el caso de que el paquete contenga numerosos grupos de datos, los valores de los campos (403) comprenden el desplazamiento respecto a la dirección final del grupo anterior de datos. En tal manera solo el primer desplazamiento determina la dirección absoluta, y los valores siguientes son las direcciones absolutas respecto al anterior de datos. Eso permite la reducción de la cantidad de bits en que se registran las direcciones de la memoria.

En la fig. 5 se ha presentado el sector de la memoria Flash en dos partes. La primera de ellas (501) es la imagen primaria de la memoria emulada EEPROM, y la segunda, es decir (502), es el juego de los paquetes de datos actualizados de esta memoria. El sector en la memoria Flash tiene el tamaño de 64 KB, por eso contiene siempre la imagen primaria completa de la memoria emulada EEPROM, incluso no comprimida, así como hasta varios miles de paquetes de datos actualizados (inglés. Patch). La imagen completa que se registra al principio de cada sector como la imagen primaria de la memoria emulada. Cada vez se puede registrar la imagen completa y modificar el sector con cada registro, pero la solución con la utilización de los paquetes de datos actualizados es más eficiente, porque, por la regla general, estos paquetes están significativamente más pequeños que 32 KB.

En la fig. 6 se ha presentado el formato del paquete de datos de actualización de la fig. 4. Los elementos (601) - (604) corresponden a los (401) - (404). El encabezamiento (601) se compone de 7 partes (601a) - (601g). El (601a) es el bit de arranque, que se cambia al iniciar la preparación al registro del paquete de los datos de actualización. El (601b) es el bit "de correcto registro del tamaño y del formato" que se modifica cuando el tamaño y el formato de datos queden correctamente registrados. El (601c) es el bit "de registro correcto" que se modifica después del registro correcto del paquete total de los datos actualizados en la memoria Flash. Con estos tres bits durante la lectura de datos (206) se puede determinar cuantos bits consecutivos en la memoria Flash pudieran ser modificados y conforme a eso definir donde se puede encontrar el paquete siguiente de datos. Si, en cualquier momento, se produce la caída de alimentación, los paquetes siguientes de datos pueden ser registrados sin que sea necesario el borrado de todo el sector directamente detrás del paquete cuyo registro no ha finalizado. El (601d) es el bit que se modifica después la anulación del paquete de los datos de actualización. Si el paquete siguiente de datos reproducirá el estado de la memoria EEPROM de antes de la actualización anterior, se puede, en vez del registro del paquete siguiente, anular el anterior. El (601e) son dos bits cuyo valor define el formato del paquete de datos de actualización. Con ayuda de dos bits se puede definir cuatro formatos, pero en la solución cuyo ejemplo se presenta aquí se puede definir 3 formatos, y el cuarto valor posible queda reservado para la futura ampliación del sistema conforme a la concepción de la invención. Los ejemplos de los formatos son los siguientes:

- 0x00 bloque simple no comprimido
- 0x01 bloque simple no comprimido
- 0x10 numerosos grupos de datos no comprimidos

El último campo del encabezamiento del paquete de los datos de actualización es el (601f) en que se determina la cantidad total de datos en el paquete de datos actualizados, sin tener en cuenta el tamaño del encabezamiento. En el caso de los campos (601f), (602) y (603) los dos primeros bits definen el formato bajo el cual se registran los valores

## ES 2 304 641 T3

de la cantidad de datos y el desplazamiento de la dirección. Los valores de estos bits para el campo (602) pueden determinar por ejemplo:

- 0x00 - el campo está descrito con la utilización de 4 bits.
- 0x01 - el campo está descrito con la utilización de 8 bits.
- 0x10 - el campo está descrito con la utilización de 12 bits.

Los valores de estos bits para los campos (601f) y (603) pueden definir por ejemplo:

- 0x00 - el campo está descrito con la utilización de los bits del bit actual.
- 0x01 - el campo está descrito con la utilización de los bits del actual bit y de él siguiente.
- 0x10 - el campo está descrito con la utilización de los bits del bit actual y de los dos bits siguientes.

En la fig. 7 se ha presentado detalladamente el punto (306) del procedimiento de la fig. 3 del dibujo. El registro del paquete de los datos de actualización empieza en el punto (701) en que cambia el valor del bit de arranque (601a). Luego, en el punto (702) del procedimiento se registran el tamaño y el tipo del paquete de datos de actualización - campos (601e), (601f). Si se produce un error, el procedimiento se termina. En la mayoría de los casos el error lo es la falta de alimentación. Si el registro de los valores de los campos es correcto, el procedimiento pasa al punto (703). En este paso cambia el valor del bit en el campo (601b). Luego, en el punto (704) del procedimiento, se registran los grupos sucesivos de los datos. Si se produce un error, el procedimiento termina el funcionamiento. Si el registro es correcto, el procedimiento pasa al punto (705). En este paso cambia el valor del bit en el campo (601c). En este momento el procedimiento termina el funcionamiento, y el paquete de los datos de actualización está correctamente registrado en la memoria Flash. Después de haber registrado hay un posibilidad adicional de su anulación. Si el paquete de los datos de actualización debe ser anulado, en el procedimiento presentado en la fig. 8 cambia el valor del campo (601d).

En la fig. 8 se ha presentado el procedimiento de preparación de los datos para actualización con la posibilidad de anulación del paquete previamente registrado. El diagrama es la presentación con más detalle del punto (301) de la fig. 3 del dibujo. El procedimiento empieza en el punto (801) donde el paquete de datos de actualización está preparado para el registro en la memoria. Entre otras se genera el encabezamiento del paquete. Luego, en el punto (802) se registran los datos en la memoria RAM donde se almacenan la imagen más actual de la memoria EEPROM, y al mismo tiempo en la memoria RAM se almacena el paquete preparando los datos de actualización. Por su parte, en el paso (803) del procedimiento se comprueba si el últimamente registrado paquete de datos de actualización en la memoria Flash está valido - el valor del campo (601d). Si el paquete está anulado, seguimos con el proceso de registro (807) del nuevo paquete de datos de actualización. En el caso contrario, es decir cuando está válido el paquete anteriormente registrado, el procedimiento para al punto (804) donde se comprueba si el paquete actualmente tratado revierte los cambios introducidos por el registro del paquete anterior de los datos de actualización. De no ser así, continua el proceso de registro (807) del nuevo paquete de los datos de actualización. En el caso contrario, cuando el paquete revierte los cambios introducidos mediante el registro del paquete anterior, el procedimiento pasa al punto (805). Aquí cambia el valor del bit (601d) que anula el paquete anteriormente registrado. Después, el procedimiento pasa al punto (806), donde se analiza (interrumpe) el registro del nuevo paquete de los datos de actualización para la memoria Flash.

En la fig. 9 se ha representado con detalle el procedimiento de selección del sector actual que se genera en el punto (202) en la fig. 2 del dibujo. El sector actual y adicional se seleccionan en base al análisis de varios criterios. El primer de ellos es si se ha terminado correctamente el registro anterior de los datos. El segundo es si el sector comprende los datos conformes con el formato requerido. La tercera de las posibilidades se produce en el caso de que la emulación de la memoria EEPROM se inicie por la primera vez en los equipos dados.

El procedimiento arranca en el (901) donde se regula el primer sector como actual. Después, en el punto (902) del procedimiento, se comprueba si los datos registrados tienen el formato correcto y si los datos han sido correctamente registrados. Eso se puede determinar a través del análisis del bit correspondiente del encabezamiento del paquete de datos. Si la comprobación del paso (903) determina que en el primer sector se encuentran los datos incorrectos, el segundo sector se ajusta como el correcto, y el primero como auxiliar (909). Si la comprobación del paso (903) determina que en el primer sector se encuentran los datos correctos, en el punto (904) se ajusta el segundo sector como el actual. Luego, en el punto (905) del procedimiento, se comprueba si los datos registrados tienen el formato correcto y si estos han sido correctamente registrados. Si la comprobación del paso (906) determina que en el segundo sector se encuentran los datos incorrectos, el primer sector se ajusta como el actual, y el segundo como el auxiliar (908). Si la comprobación del paso (906) determina que en el primer sector se encuentran los datos correctos, en el punto (907) del procedimiento como el sector actual se ajusta el sector en que hay más espacio libre. El sector en que hay más espacio libre comporta los datos más actuales. El procedimiento, la selección del sector actual y del auxiliar terminan el funcionamiento en el punto (910).

## REIVINDICACIONES

- 5 1. Método de emulación de una memoria EEPROM mediante un software en otra memoria no volátil (150) de un dispositivo con una memoria operativa (160), comprendiendo el método las etapas de:
- 10 crear, en la memoria operativa (160), una primera memoria de tránsito o buffer para el almacenamiento de una imagen actual de la memoria EEPROM emulada;
- 15 fijar dos sectores de la memoria no volátil (150) como un sector actual y un sector auxiliar;
- 20 almacenar una imagen de la memoria EEPROM emulada en una parte (501) del sector actual;
- 25 actualizar la memoria EEPROM emulada añadiendo, a la parte restante (502) del sector actual, uno o más paquetes, describiendo cambios en el contenido de la imagen actual de la memoria EEPROM emulada, y almacenando la imagen actual de la memoria EEPROM emulada, incluyendo los cambios introducidos por un o más paquetes, en la primera memoria de tránsito o buffer;
- 30 repetir la etapa de actualización de la memoria EEPROM emulada hasta que no se pueda añadir un nuevo paquete al sector actual;
- 35 fijar el sector auxiliar como el sector actual y el sector actual como el sector auxiliar;
- 40 almacenar la imagen actual de la memoria emulada desde la primera memoria de tránsito o buffer de la memoria operativa (160) hasta el sector actual; y
- 45 borrar el sector auxiliar.
- 50 2. Método según la reivindicación 1, que comprende además la etapa de volver a la etapa de actualización de la memoria emulada EEPROM después de la etapa de borrado del sector auxiliar.
- 55 3. Método según la reivindicación 1, en el que se crea una segunda memoria de tránsito o buffer, en la memoria operativa (160), para el almacenamiento del último paquete que describen cambios en el contenido de la imagen actual de la memoria emulada EEPROM.
- 60 4. Método de emulación de la memoria EEPROM según la reivindicación 1, que comprende además la etapa de crear una la memoria de tránsito o buffer en la memoria operativa (160), la cual se utiliza para el almacenamiento del paquete después de la compresión.
- 65 5. Método de emulación de la memoria EEPROM según la reivindicación 1, en el que la etapa de fijar los dos sectores de la memoria no volátil (150) como el sector actual y el sector auxiliar, comprende las etapas de:
- 70 seleccionar el sector actual a partir de los dos sectores de la memoria no volátil (150), conteniendo el sector actual la imagen de la memoria emulada EEPROM y los paquetes que describen cambios en el contenido de la imagen de la memoria emulada EEPROM;
- 75 borrar contenido del sector auxiliar;
- 80 tomar la imagen de la memoria emulada EEPROM desde el sector actual en la memoria no volátil (150) a la memoria operativa;
- 85 tomar un primer paquete de los paquetes del sector actual de la memoria no volátil (150);
- 90 verificar la validez del primer paquete;
- 95 salvar datos del primer paquete en la primera memoria de tránsito o buffer de la memoria operativa (160) cuando el primer paquete es válido;
- 100 verificar si existen otros paquetes en el sector actual de la memoria no volátil (150); y
- 105 procesar los paquetes restantes en el sector actual de la memoria no volátil (150) como el primer paquete.
- 110 6. Método de emulación de la memoria EEPROM según la reivindicación 5, en el que la etapa de seleccionar el sector actual de entre dos sectores de la memoria no volátil (150) comprende las etapas de:
- 115 ajustar inicialmente un primer sector de los dos sectores de la memoria no volátil (150) como el sector actual;
- 120 verificar si los datos salvados en el primer sector tienen un formato correcto y si han sido correctamente salvados;
- 125 fijar inicialmente un segundo sector de los dos sectores de la memoria no volátil (150) como el sector actual cuando los datos salvados en el primer sector tienen un formato incorrecto o han sido incorrectamente salvados;

## ES 2 304 641 T3

verificar si los datos salvados en el segundo sector tienen un formato correcto y han sido correctamente salvados;

fijar el primer sector de los dos sectores de la memoria no volátil (150) como el sector actual y fijar el segundo sector de los dos sectores de la memoria no volátil (150) como el sector auxiliar cuando los datos salvados en el segundo sector tienen un formato incorrecto o han sido incorrectamente salvados; y

fijar un sector de los dos sectores que tiene más memoria libre como el sector actual cuando los datos salvados en el primer sector y el segundo sector tienen un formato incorrecto o han sido incorrectamente salvados.

7. Método de emulación de la memoria EEPROM según la reivindicación 1, en el que un proceso de escritura del nuevo paquete comprende las etapas de:

preparar datos para actualizar el contenido de la memoria no volátil (150) y salvar los datos en la primera memoria de tránsito o buffer de la memoria operativa (160);

verificar el tamaño de los datos;

verificar el espacio libre para salvar los datos en el sector actual de la memoria no volátil (150);

salvar los datos como el paquete nuevo cuando el espacio libre en el sector actual de la memoria no volátil (150) tiene un tamaño suficiente para salvar el nuevo paquete;

cambiar el sector auxiliar en el sector actual y un sector, que previamente es el sector actual, en el sector auxiliar cuando en el sector actual de la memoria no volátil (150) no hay suficiente espacio libre para salvar el nuevo paquete; y

salvar el nuevo paquete en un sector que es previamente el sector auxiliar y borrar el contenido del sector que previamente es el sector actual.

8. Método de emulación de la memoria EEPROM según la reivindicación 1, en el que un formato de los paquetes se compone de un primer campo que es un encabezamiento del paquete, un segundo campo que define el tamaño del grupo de datos, estando presente el segundo campo en el caso de que los paquetes contengan muchos grupos de datos, un tercer campo que contiene un campo offset (desplazamiento) de datos respecto a una dirección inicial y un cuarto campo que contiene datos.

9. Método de emulación de la memoria EEPROM según la reivindicación 8, en el que cuando el paquete contiene grupos de datos salvados bajo diferentes direcciones de memoria, los valores de offset de los grupos de datos contienen un offset respecto a la última dirección de un grupo previo de datos, de los cuales sólo el primer offset define una dirección absoluta, mientras que los siguientes valores son direcciones relativas respecto al grupo anterior de datos.

10. Método de emulación de la memoria EEPROM según la reivindicación 8, en el que el formato del encabezamiento del paquete consta de un bit de arranque que cambia cuando se arranca la preparación del paquete para salvar, un bit de escritura correcta del tamaño y del formato cambiado después de que sean grabados correctamente un valor de tamaño y formato de datos, un bit de escritura correcta cambiada después de que todo el paquete sea grabado correctamente en la memoria no volátil, un bit de invalidación cambiado después de que el paquete sea invalidado y el campo que define el formato del paquete y el campo que define la cantidad total de datos en el paquete, no considerando el tamaño del encabezamiento.

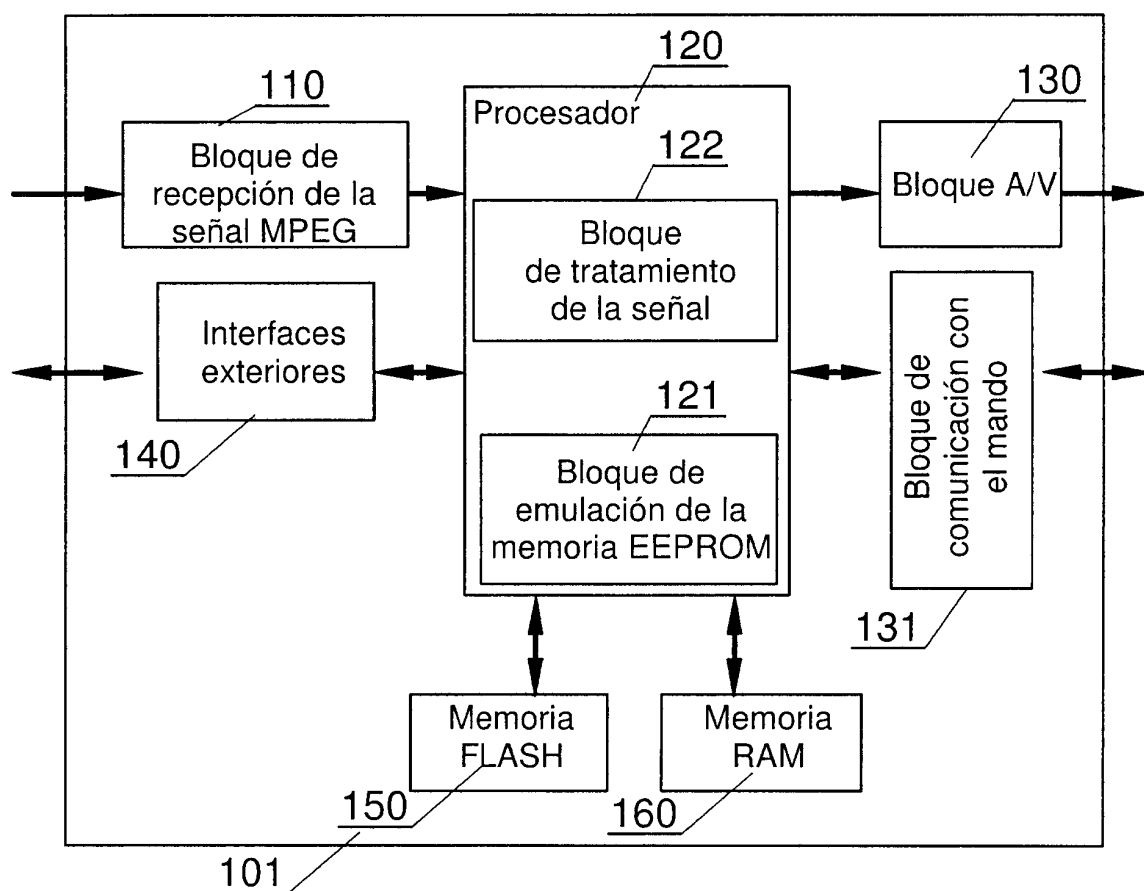


Fig. 1

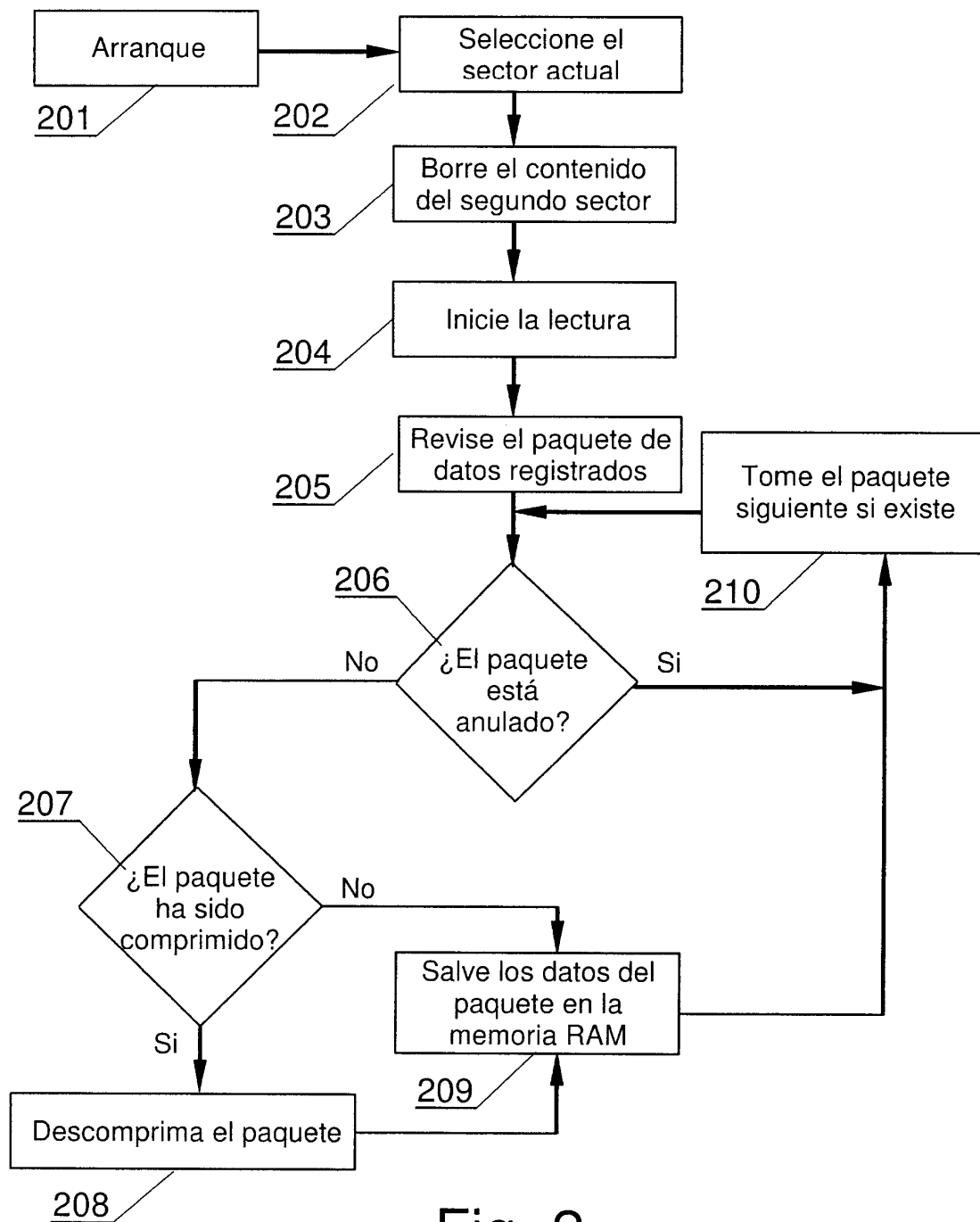


Fig. 2

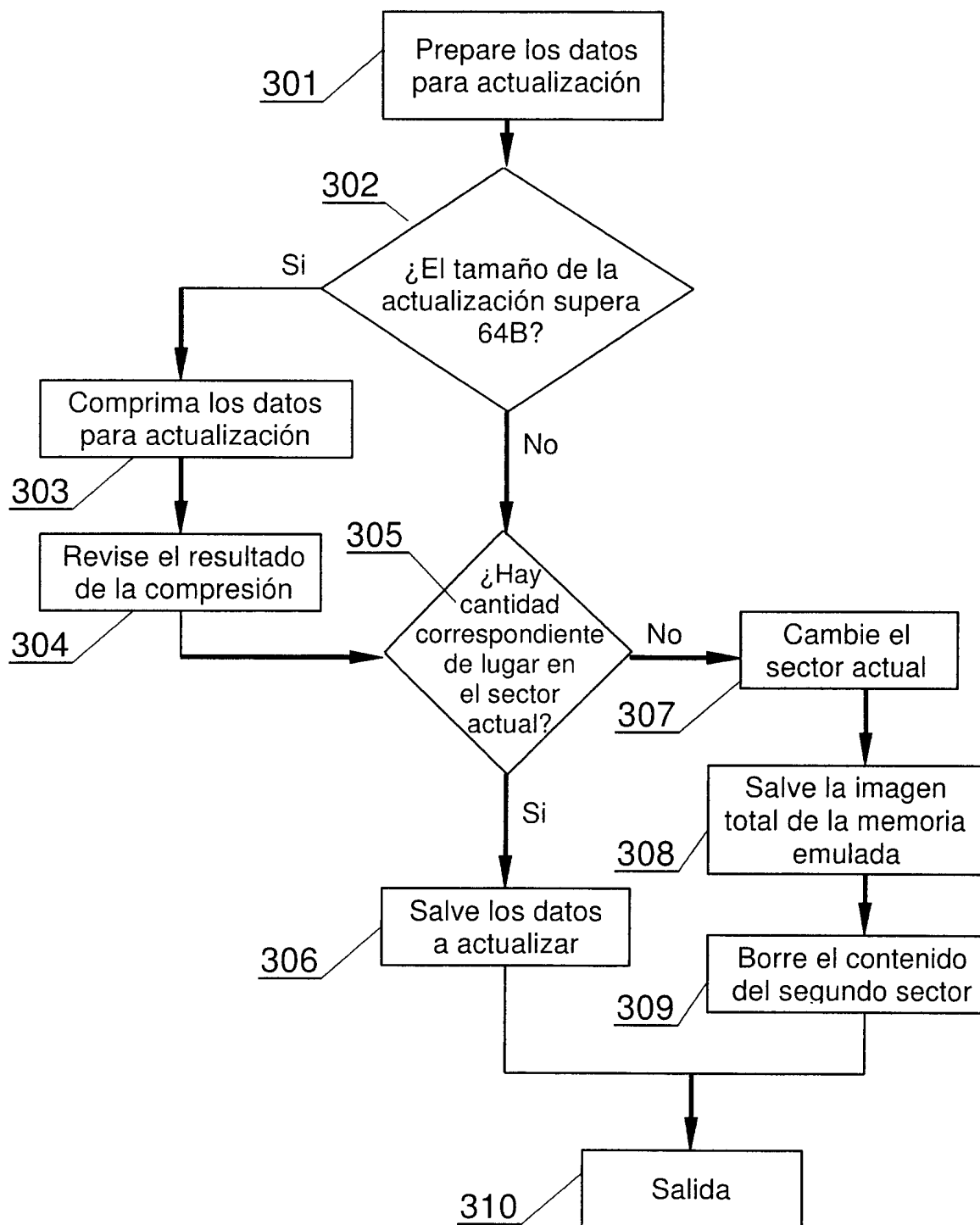


Fig. 3

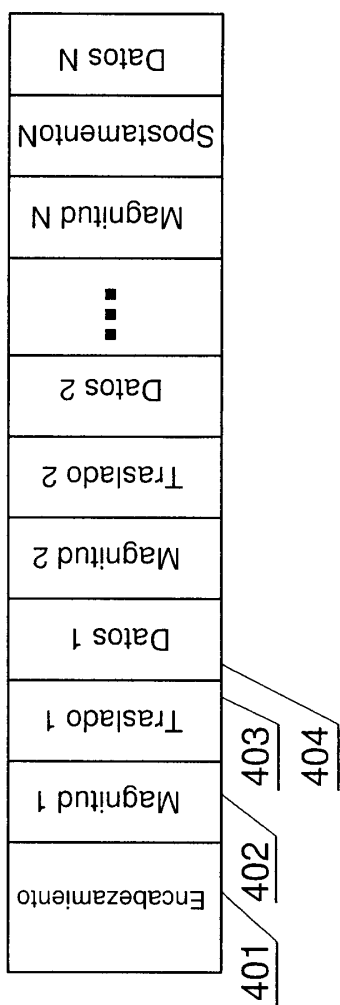


Fig. 4

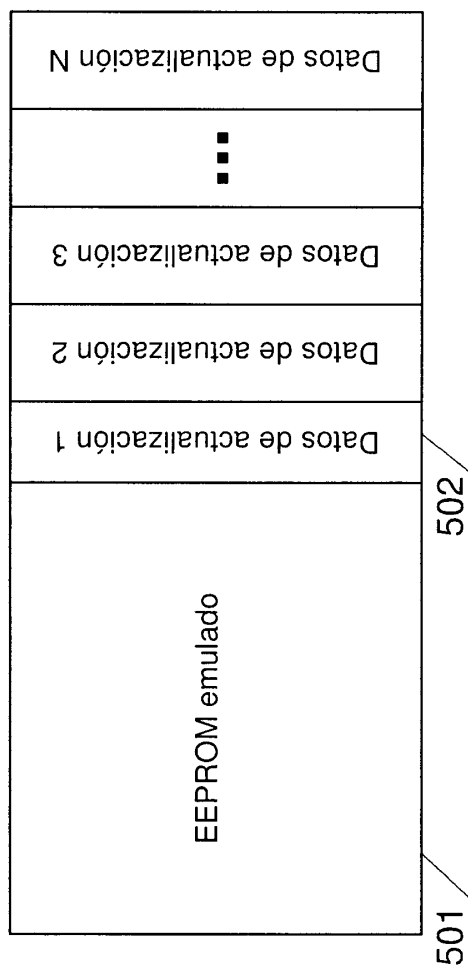


Fig. 5

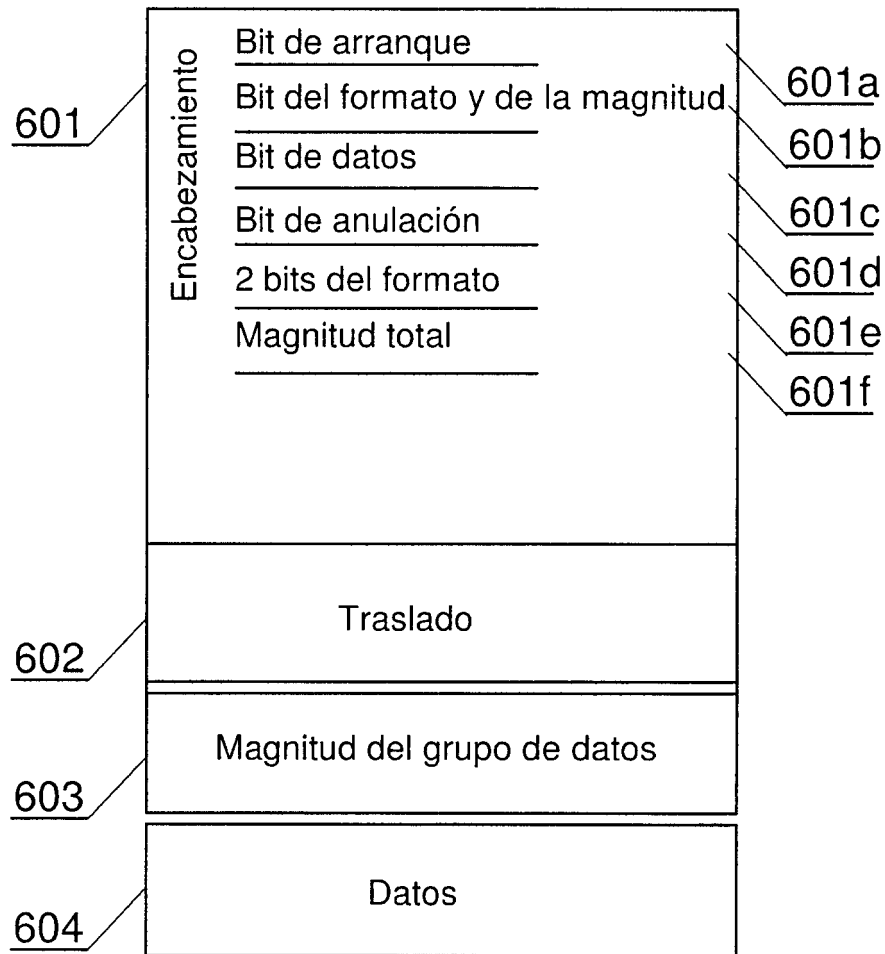


Fig. 6

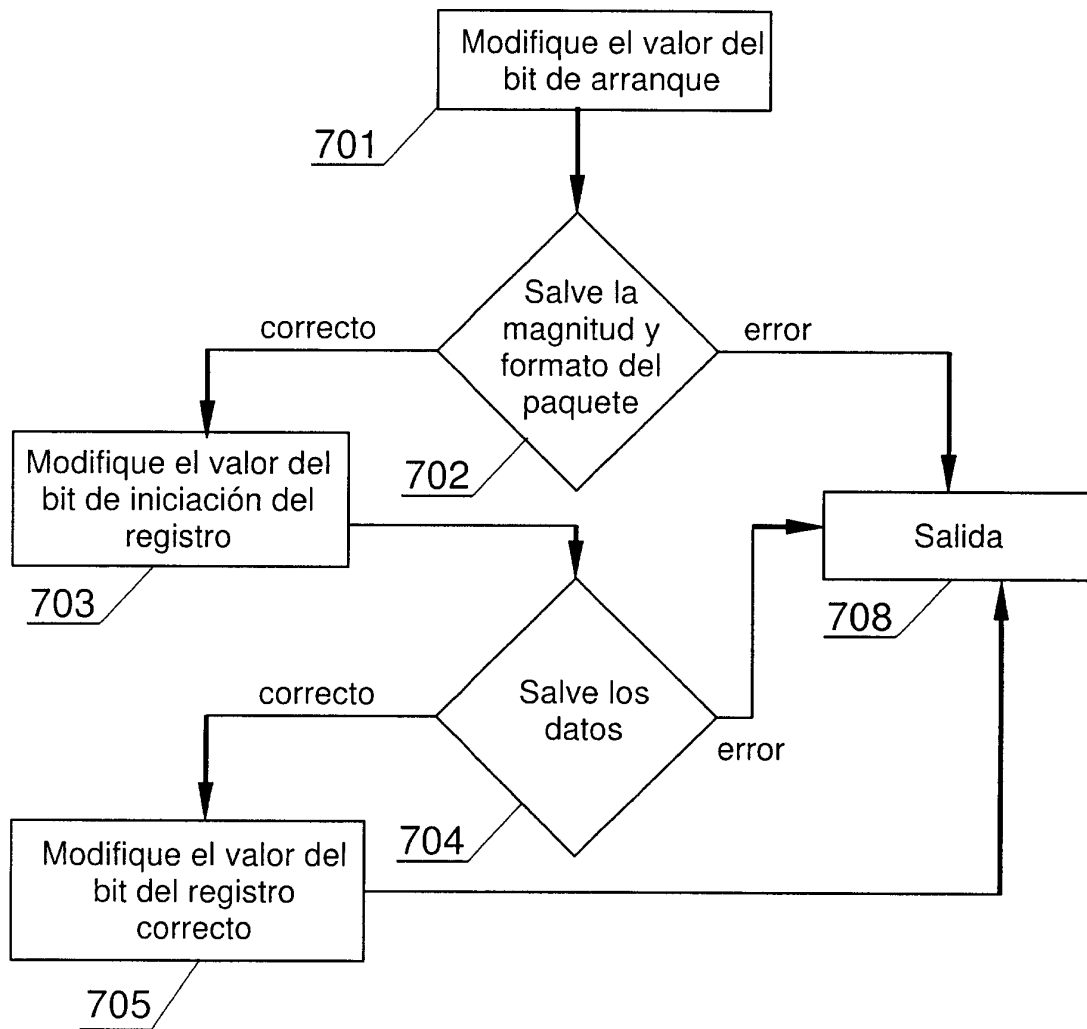


Fig. 7

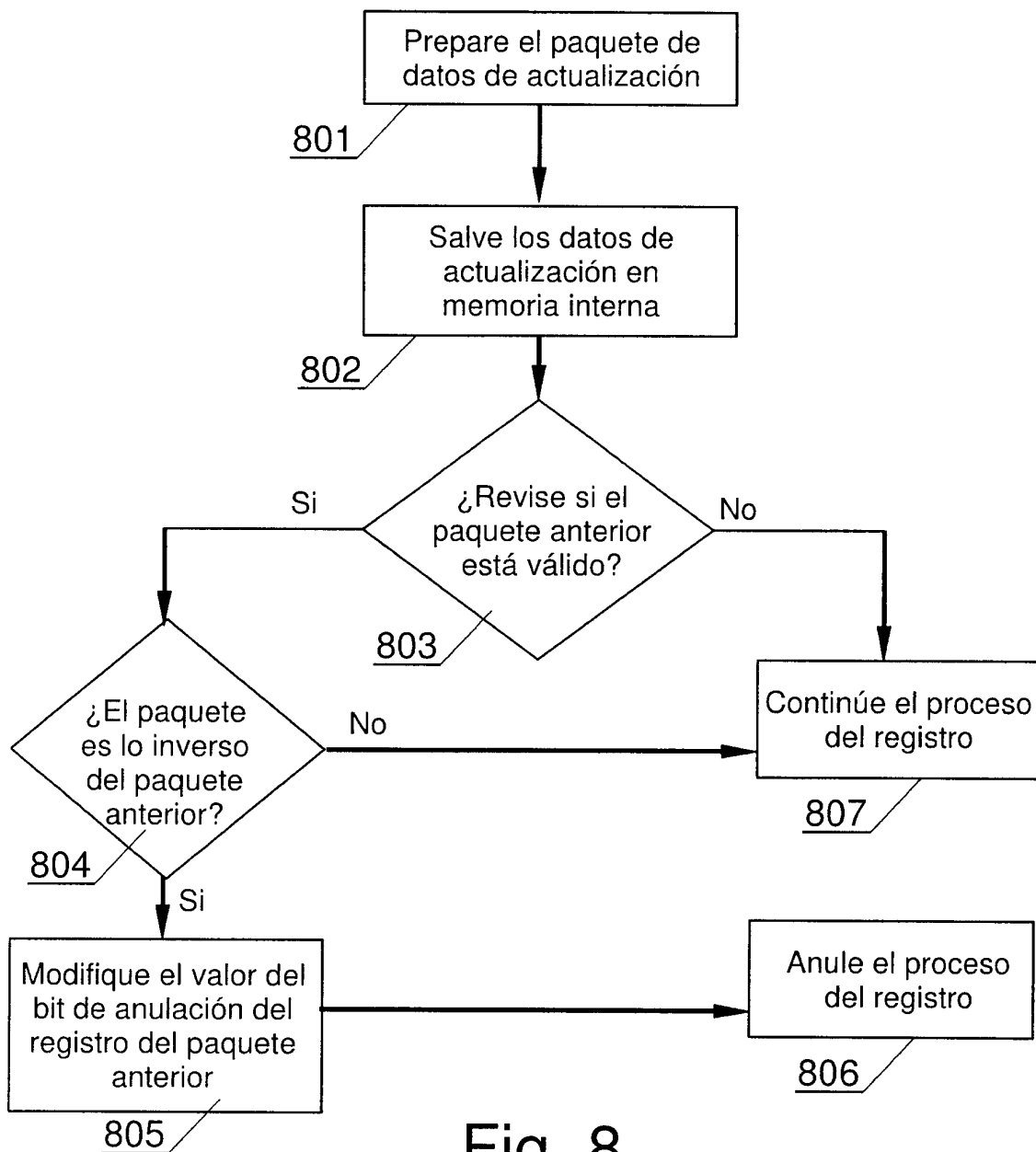


Fig. 8

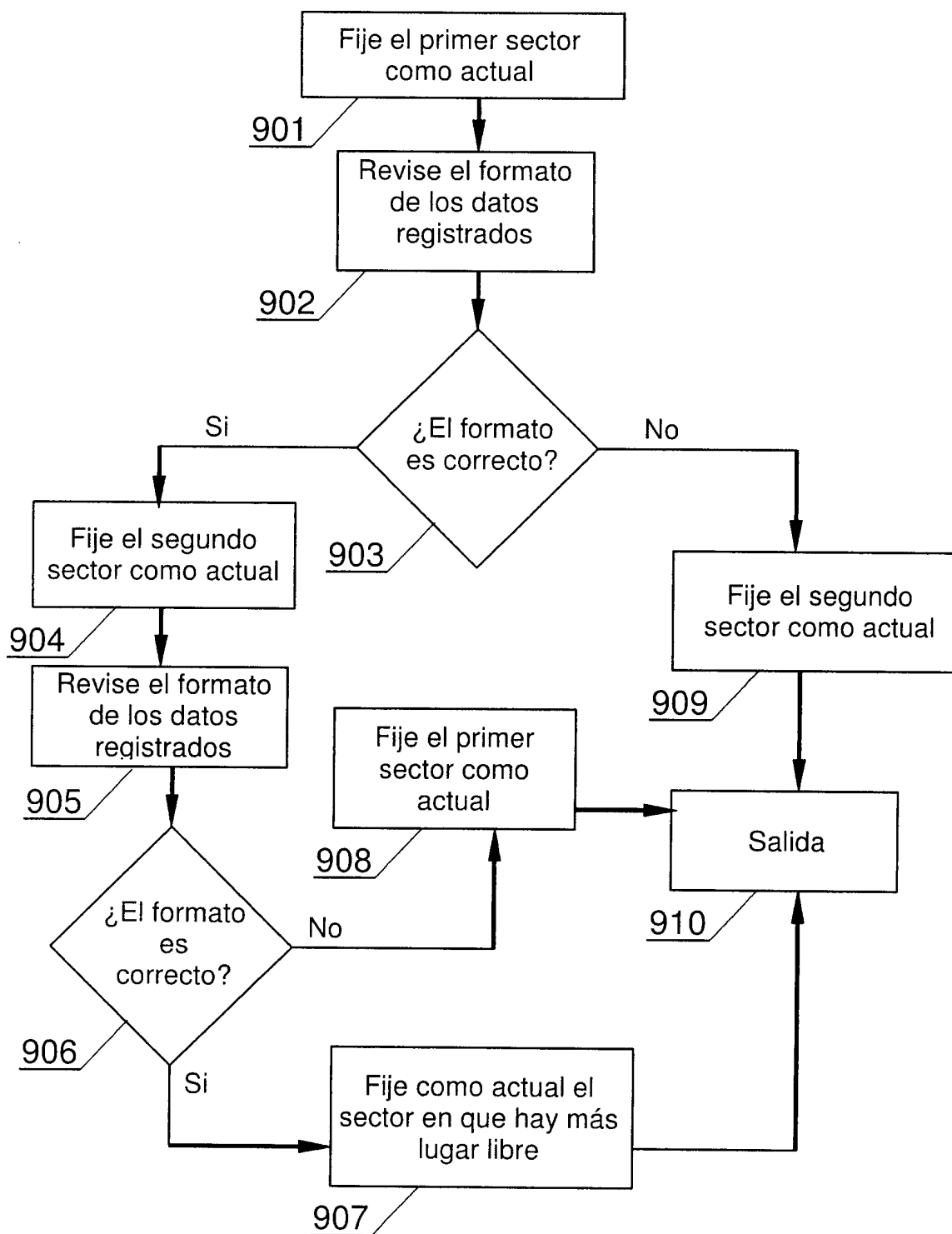


Fig. 9