

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 378 371**

51 Int. Cl.:
G06F 11/10 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **09739378 .9**
- 96 Fecha de presentación: **01.04.2009**
- 97 Número de publicación de la solicitud: **2297642**
- 97 Fecha de publicación de la solicitud: **23.03.2011**

54 Título: **Optimización de recopia para un sistema de memoria**

30 Prioridad:
30.04.2008 US 49330
18.08.2008 US 193638

45 Fecha de publicación de la mención BOPI:
11.04.2012

45 Fecha de la publicación del folleto de la patente:
11.04.2012

73 Titular/es:
APPLE INC.
1 Infinite Loop
Cupertino CA 95014-2084 , US

72 Inventor/es:
WAKRAT, Nir, Jacob y
HELM, Mark, Alan

74 Agente/Representante:
Fàbrega Sabaté, Xavier

ES 2 378 371 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

Optimización de recopia para un sistema de memoria.

Campo

Este objeto se refiere en general a sistemas de memoria.

5 Antecedentes

Una memoria flash es un tipo de memoria programable borrable eléctricamente de solo lectura (EEPROM). Debido a que las memorias flash son no volátiles y relativamente densas, se usan para almacenar archivos y otros objetos persistentes en ordenadores portátiles, teléfonos móviles, cámaras digitales, reproductores de música portátiles, y muchos otros dispositivos en los que otras soluciones de almacenamiento (por ejemplo, discos magnéticos) son inapropiadas. Cuando se procesan datos almacenados en una memoria flash, se lee una página de datos de la memoria flash y se almacena en un registro. El contenido del registro se transfiere a un controlador que es externo a la memoria no volátil para procesamiento. La página de datos procesada se vuelve a introducir en el registro de forma que la página de datos procesada puede volver a escribirse en la memoria flash.

Se conoce por el documento US2006050576 un dispositivo de memoria flash NAND que implementa una función de limpieza basada en una función de recopia, en la que se leen datos del vector de memoria y se revisan/corrigiendo usando EDC, de forma que los datos corregidos se copian de nuevo en el vector de memoria.

Se conoce por el documento US4020466 un sistema de memoria jerárquico que implementa un algoritmo de recopia 'el primero modificado menos recientemente', en el que el volver a copiar los datos en cualquier nivel se hace en base al orden en que se hizo la entrada en el almacén de recopia.

20 RESUMEN

En una operación de recopia o lectura para un subsistema de memoria no volátil, se usan indicadores de cambio en página de datos para gestionar la transferencia de páginas de datos entre un registro en una memoria no volátil y un controlador que es externo a la memoria no volátil. La invención se define en detalle en las reivindicaciones adjuntas independientes 1 y 5.

25 DESCRIPCION DE LOS DIBUJOS

La Figura 1a es un diagrama de bloques de un sistema de memoria de ejemplo para implementar un proceso de recopia optimizado.

La Figura 1B es un diagrama de bloques de otro sistema de memoria de ejemplo para implementar un proceso de recopia optimizado.

30 La Figura 2 es un diagrama de flujo de un proceso de recopia optimizado de ejemplo 200 para computar y almacenar indicadores de cambio de datos utilizados por los sistemas de memoria de las Figuras 1A y 1B.

Las Figuras 3A-3C son diagramas de flujo de un proceso de recopia optimizado de ejemplo implementado por los sistemas de memoria de las Figuras 1A y 1B.

35 La Figura 4 es un diagrama de flujo de una operación de lectura optimizada de ejemplo implementada por los sistemas de memoria de las Figuras 1A y 1B.

La Figura 5 ilustra estructuras de datos de ejemplo utilizadas en los procesos de recopia de las Figuras 3A-3C.

DESCRIPCION DETALLADA

Ejemplos de Subsistemas de Memoria

40 La Figura 1A es un diagrama de bloques de un sistema de memoria 100 de ejemplo para implementar un proceso optimizado de recopia. En algunas implementaciones, el sistema 100 puede incluir un subsistema de memoria 102 acoplado a un dispositivo anfitrión 124 a través de un bus externo 122 (por ejemplo, Interfaz Flash Abierta NAND (ONFI), ATA). El dispositivo anfitrión 124 puede ser cualquier dispositivo electrónico incluyendo pero no limitado a teléfonos inteligentes, reproductores, dispositivos de mensajes de texto, ordenadores portátiles, disquetes de estado sólido, cámaras digitales, etcétera. El subsistema de memoria 102 puede ser cualquier subsistema de memoria no volátil (por ejemplo, NAND gestionada).

45 El dispositivo anfitrión 124 puede incluir un sistema en un chip (SoC) 126 y memoria volátil 128. El SoC 126 puede incluir hardware y software para interactuar con el subsistema de memoria 102, tal como transmitir solicitudes de lectura y escritura hechas por aplicaciones que se están ejecutando en el dispositivo anfitrión 124.

El subsistema de memoria 102 puede incluir memoria no volátil 104 (también denominada “memoria sin editar”) y un controlador externo 116. La memoria 104 puede incluir un vector de celdas de memoria 106, una máquina de estados interna 108, un registro de memoria 110 y un indicador de cambio de datos (DCI) 130. El controlador 116 puede incluir un procesador 118, memoria volátil 114 y una máquina de códigos correctores de errores (ECC) 120.

5 Son posibles otras configuraciones para el subsistema de memoria 102. Por ejemplo, puede incluirse un registro de memoria temporal en la ruta de datos entre el vector de celdas de memoria 106 y el registro de memoria 110 para permitir a la máquina interna de estados leer a siguiente página de datos del vector de celdas de memoria 106 mientras transfiere la pagina actual al controlador 116 a través del bus interno 112.

10 La Figura 1B es un diagrama de bloques de otro sistema de memoria de ejemplo 130 para implementar un proceso optimizado de recopia. El sistema 130 muestra solo el subsistema de memoria 102, cuya operación se describió en referencia a la Figura 1A.

Operaciones de Recopia

La recopia puede ser un comando del subsistema de memoria para trasladar datos desde una página a otra página. La recopia se puede utilizar en nivelación de desgaste y otras operaciones de gestión de memoria no volátil. En una operación típica de recopia, una página de datos es leída del vector de celdas de memoria 106 y es almacenada en el registro de memoria 110 por la máquina de estados interna. El controlador externo 116 lee o saca la página de datos del registro de memoria 110 de forma que el procesador 118 puede llevar a cabo una operación deseada en la página de datos (por ejemplo, una operación ECC). La página de datos procesada puede volver a ser escrita en el registro de memoria 100 por el controlador 116. La máquina de estados interna 108 puede escribir el contenido del registro de memoria 110 en una nueva página de datos en el vector de celdas de memoria 106. Evitando la transferencia de contenidos enteros del registro 110 al controlador externo 116, se pueden reducir los tiempos de procesamiento y el consumo de energía.

Proceso de ejemplo para escribir nuevas páginas de memoria

La Figura 2 es un diagrama de flujo de un proceso optimizado de recopia de ejemplo 200 para computar y almacenar indicadores de cambio de datos utilizados por los sistemas de memoria de las Figuras 1A y 1B. En algunas implementaciones, el proceso 200 puede empezar cuando una nueva página de datos se transfiere a un registro de memoria de un subsistema de memoria (202). Se puede computar un DCI para la página de datos y almacenarlo en el subsistema de memoria (204). Algunos ejemplos de indicadores de cambio de datos pueden incluir códigos de detección de errores (EDC), incluyendo pero no limitados a: suma de comprobación, código Hamming, bit de paridad, comprobación de redundancia cíclica (CRC), inversión de polaridad de símbolo, código Turbo, etcétera. Un EDC también puede ser parte de un ECC, tal como código Reed-Salomon, código Reed-Muller, código binario Golay, y códigos de comprobación de paridad de baja densidad. Una vez se han computado y almacenado los DCI, el contenido del registro puede escribirse en la memoria no volátil (206). Los DCI pueden usarse para el proceso de recopia optimizado descrito en referencia a las Figuras 3A-3C.

Proceso de Recopia Optimizado de Ejemplo

Las Figuras 3A-3C son diagramas de flujo de un proceso de recopia optimizado 300 implementado por los sistemas de memoria de las Figuras 1A y 1B. En algunas implementaciones, el proceso 300 puede empezar leyendo una página de datos de una memoria no volátil (por ejemplo, el vector de celdas de memoria 106) y almacenando la página de datos en un registro de memoria (302) (por ejemplo, el registro 110). Un DCI (por ejemplo, el DCI 130) puede ser verificado para la página de datos en el registro de memoria (304). En algunas implementaciones, el DCI puede ser computado y almacenado previamente en el subsistema de memoria cuando la página de datos se escribe primero en memoria no volátil (por ejemplo utilizando el proceso 200). En otras implementaciones el DCI se puede computar “en el momento” como parte de la operación de lectura o como una operación separada. El subsistema de memoria puede notificar al controlador externo (por ejemplo, controlador externo 116) que se ha completado la operación de lectura (306).

Si el DCI para la página de datos en el registro indica que la página de datos ha cambiado (308), la página de datos puede transferirse al controlador externo (310), se puede computar un ECC para la página de datos (312) y la información ECC puede computarse para la página de datos (312) y la información ECC para la página de datos puede corregirse en el controlador externo (314). La página de datos y el ECC asociado puede transferirse de vuelta al registro de memoria (316). Se puede computar un nuevo DCI para la página de datos en el registro de memoria (318).

Si el DCI para la página de datos en el registro indica que la página de datos no ha cambiado (308), se puede computar un ECC para nuevos metadatos asociados con la página de datos (320). Los nuevos metadatos y el ECC asociado se pueden transferir al registro de memoria (322). Se puede computar un nuevo DCI para los metadatos en el registro de memoria. El contenido del registro de memoria puede escribirse en una nueva página en la memoria no volátil (326).

Operación de Lectura Optimizada

La Figura 4 es un diagrama de flujo de una operación de lectura optimizada de ejemplo implementada por los sistemas de memoria de las Figuras 1A y 1B. En algunas implementaciones, el proceso 400 puede empezar leyendo la página de datos de una memoria no volátil y almacenando la página de datos en un registro de memoria (402). Un DCI pre-computado asociado con la página de datos se puede introducir en el registro de memoria desde un lugar de almacenamiento en el subsistema de memoria y se puede verificar (404). El DCI también se puede computar "en el momento". El estado del DCI puede leerse para determinar si la página de datos ha cambiado (406).

Si la página de datos ha cambiado (408), un ECC para los nuevos metadatos asociados con la página de datos puede computarse y transferirse, junto con los nuevos metadatos al registro de memoria (412). Un nuevo DCI puede computarse para los metadatos en el registro de memoria y el contenido del registro puede escribirse en una nueva página de datos en la memoria no volátil (414).

Si la página de datos no ha cambiado (408), el contenido del registro de memoria puede transferirse a un controlador externo y un ECC puede ser computado por el controlador externo (410). Un ECC para los nuevos metadatos asociados con la página de datos puede ser computado y transferido por el controlador externo, junto con los nuevos metadatos al registro de memoria (412). Un nuevo DCI puede computarse para los nuevos metadatos en el registro de memoria y el contenido del registro puede escribirse en una nueva página de datos en la memoria no volátil (414).

Estructuras de datos de ejemplo

La Figura 5 ilustra estructuras de datos de ejemplo utilizadas por el proceso de recopia de las Figuras 3A-3C. En el ejemplo mostrado, una página de datos 500 en una memoria no volátil incluye metadatos 502, metadatos ECC 504, datos 506 y datos ECC 508. Si los datos no han cambiado entonces solo los metadatos 502 y los metadatos ECC 504 se transfieren desde el registro al controlador externo 510 para procesamiento. Los metadatos 502 y metadatos ECC 504 procesados se transfieren de vuelta al subsistema de memoria y se almacenan en el registro de memoria. La página de datos puede escribirse en una nueva página en memoria no volátil (por ejemplo, un vector de celdas de memoria).

Se han descrito un número de implementaciones. Sin embargo, debe entenderse que ciertas modificaciones pueden ser llevadas a cabo. Por ejemplo, elementos de una o más implementaciones pueden ser combinados, borrados, modificados, o suplementarse para formar implementaciones adicionales. Como otro ejemplo más, los flujos lógicos mostrados en las figuras no requieren el orden particular mostrado, u orden secuencial, para alcanzar los resultados deseados. Además, se pueden proporcionar otros pasos, o se pueden eliminar pasos, de los flujos descritos, y otros componentes pueden ser añadidos a, o eliminados de, los sistemas descritos. De la misma manera, otras implementaciones están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un método que comprende:
 - obtener una página de datos y metadatos asociados de una memoria no volátil (106) de un subsistema de memoria (102) acoplado a un controlador (116),
 - almacenar la página de datos en un registro (110) del subsistema de memoria (102),
 - 5 enviar la página de datos y metadatos asociados desde el registro (110) al controlador (116) para procesamiento;
 - recibir una página de datos procesada y metadatos asociados desde el controlador (116),
 - determinar si la página de datos procesada ha cambiado;
 - si la página de datos procesada ha cambiado,
 - 10 escribir la página de datos procesada y metadatos asociados en el registro (110) en el subsistema de memoria (102);
 - si la página de datos procesada no ha cambiado,
 - escribir los metadatos asociados en el registro (110) en el subsistema de memoria (102); y escribir el contenido del registro en la memoria no volátil (106).
- 15 2. El método según la reivindicación 1 donde determinar si la página de datos procesada ha cambiado, comprende además:
 - leer un indicador de cambio de datos para la página de datos procesada.
3. El método según la reivindicación 2, donde el indicador de cambio de datos está asociado con un código de detección de errores.
- 20 4. El método según la reivindicación 1, en donde determinar si la página de datos procesada ha cambiado, comprende además:
 - computar un indicador de cambio de datos para la página de datos procesada; y
 - determinar si la página de datos procesada ha cambiado en base al indicador de cambio de página.
5. Un sistema (100) que comprende:
 - 25 un interfaz (112) adaptado para acoplarse a un controlador (116);
 - memoria no volátil (106);
 - un procesador (118) acoplado a la memoria no volátil (106) y a la interfaz (112) el procesador operable para obtener una página de datos y metadatos asociados de la memoria no volátil (106) de un subsistema de memoria (106) acoplado al controlador (116);
 - 30 almacenar la página de datos en un registro (110) del subsistema de memoria (102);
 - enviar la página de datos y metadatos asociados desde el registro (110) a través de la interfaz (112) al controlador (116) para procesamiento;
 - recibir una página de datos procesada y metadatos asociados desde el controlador (116) a través de la interfaz (112);
 - 35 determinar si la página de datos procesada ha cambiado;
 - si la página de datos procesada ha cambiado,
 - transferir la página de datos procesada y metadatos asociados en el registro (110) al subsistema de memoria (102);
 - si la página de datos procesada no ha cambiado,
 - 40 transferir los metadatos asociados en el registro (110) en el subsistema de memoria (102); y escribir el contenido del registro en la memoria no volátil (106).

6. El sistema según la reivindicación 5 donde determinar si la página de datos procesada ha cambiado, comprende además:

leer un indicador de cambio de datos para la página de datos procesada.

- 5 7. El sistema según la reivindicación 6, donde el indicador de cambio de datos está asociado con un código de detección de errores.

8. El sistema según la reivindicación 5, donde determinar si la página de datos procesada ha cambiado, comprende además:

computar un indicador de cambio de datos para la página de datos procesada; y

determinar si la página de datos procesada ha cambiado en base al indicador de cambio de página.

10

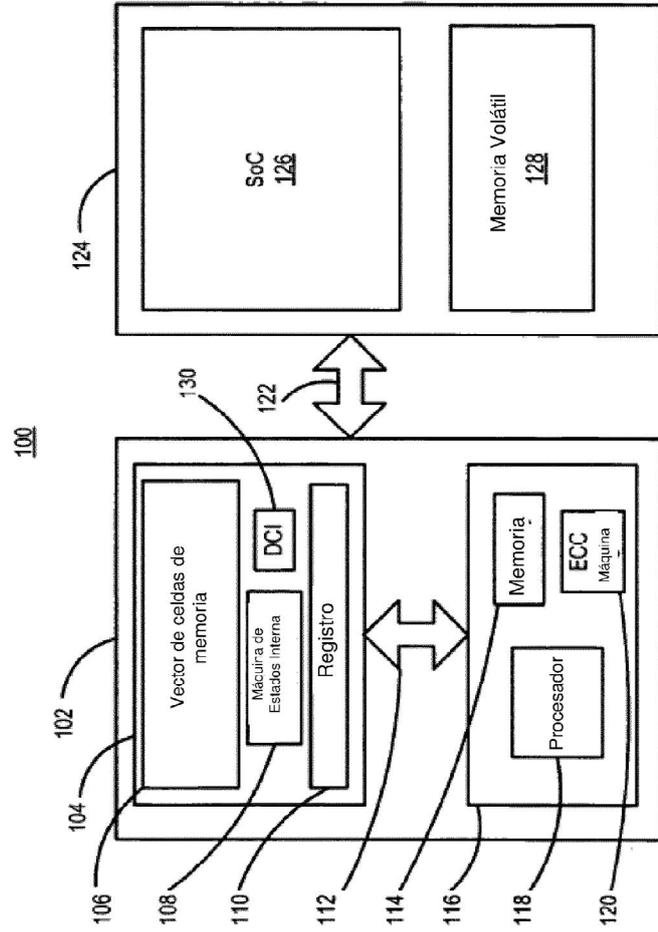


FIG. 1A

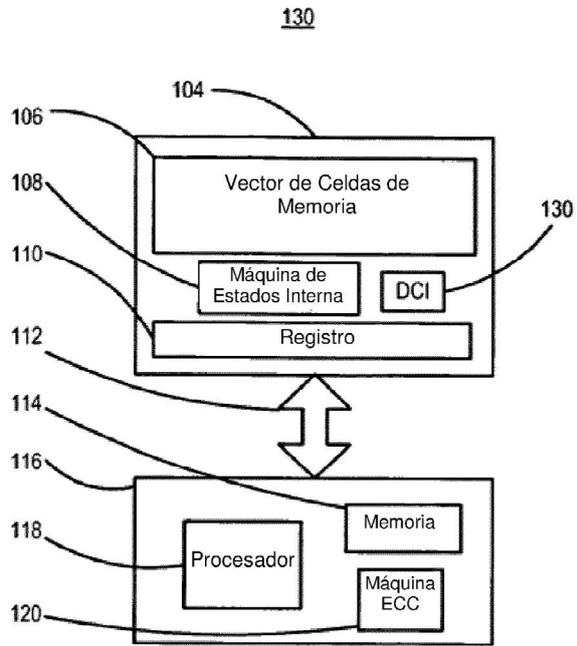


FIG. 1B

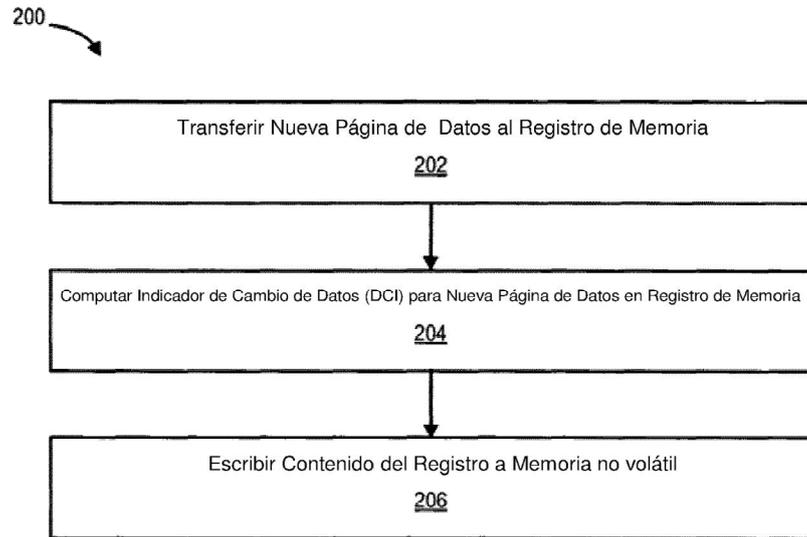


FIG. 2

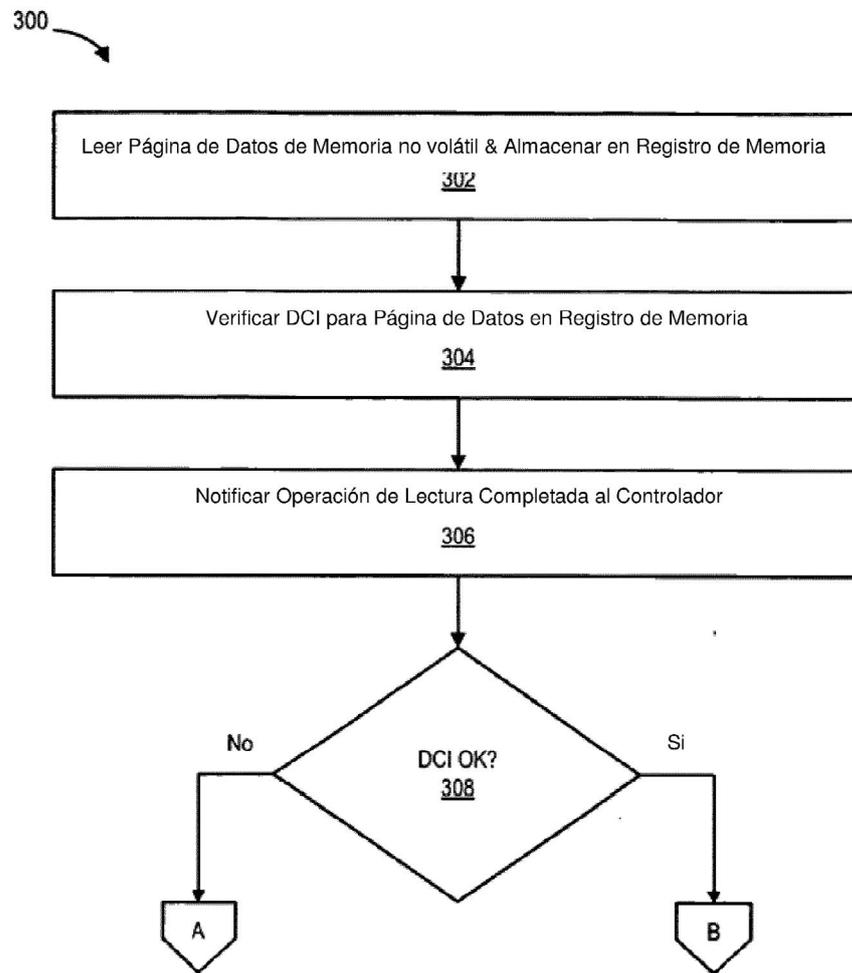


FIG. 3A

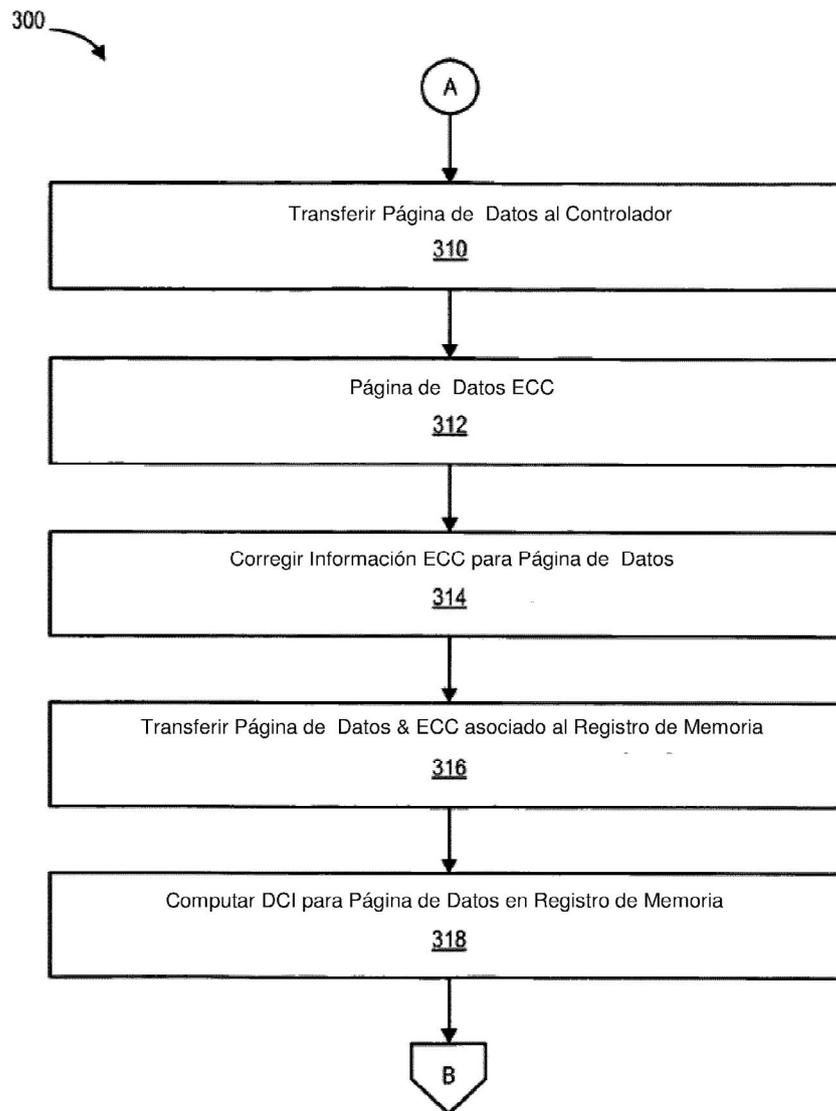


FIG. 3B

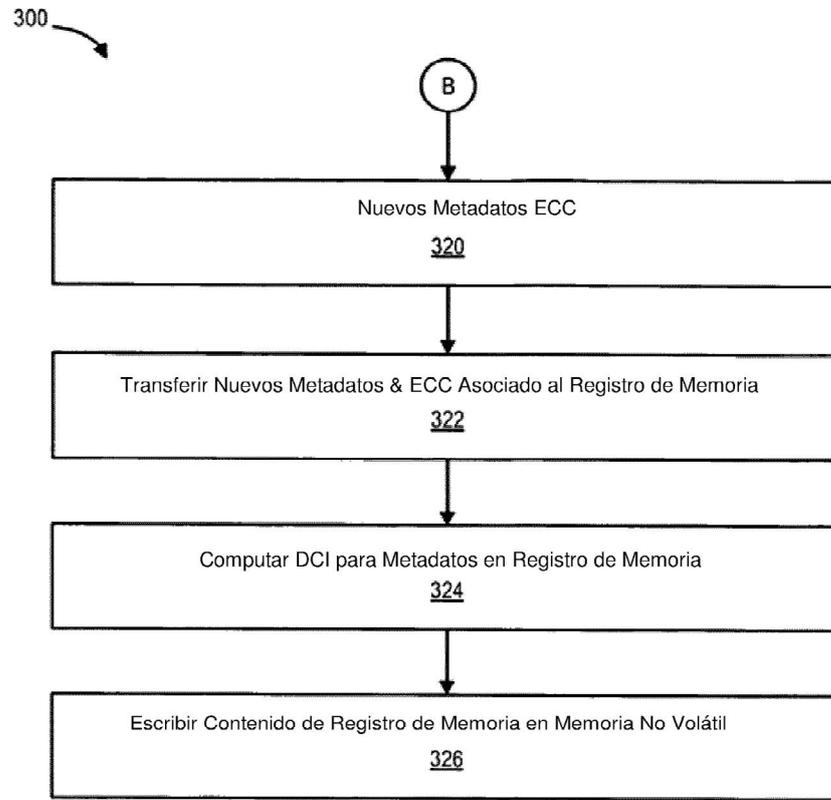
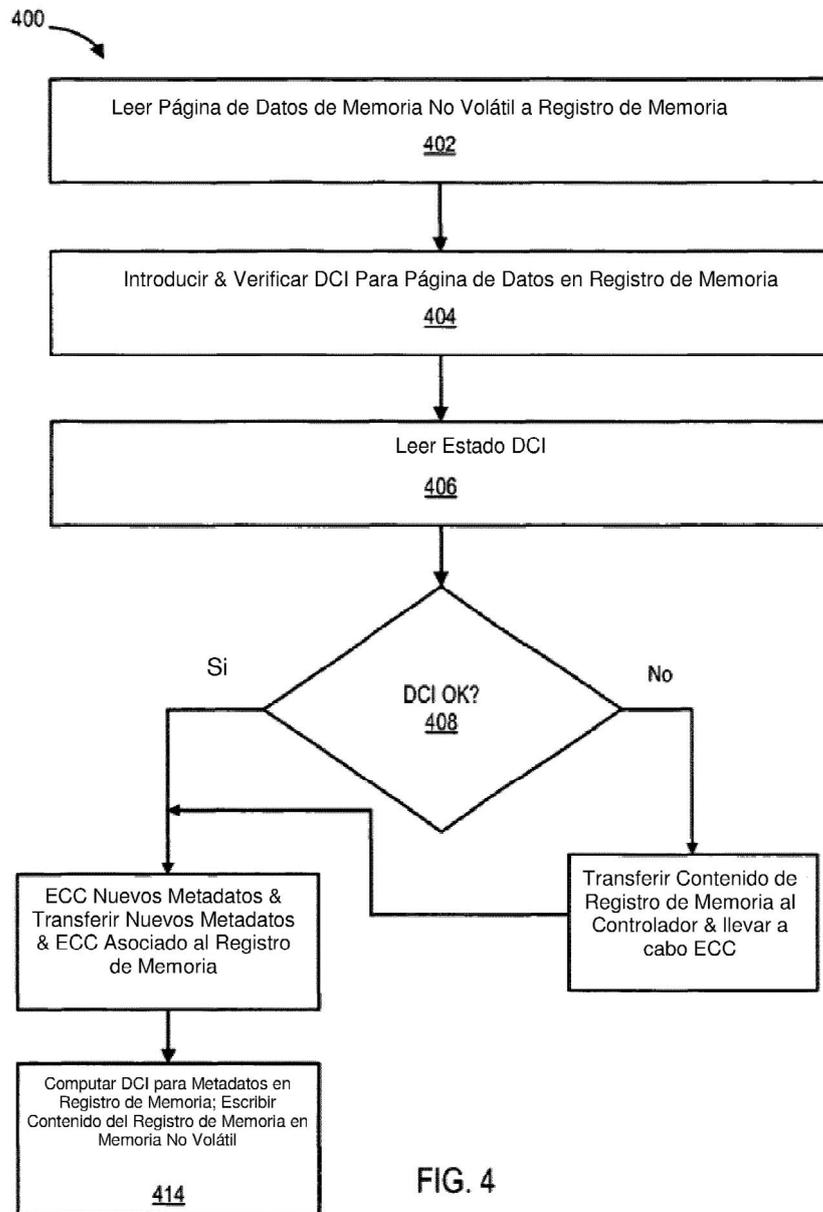


FIG. 3C



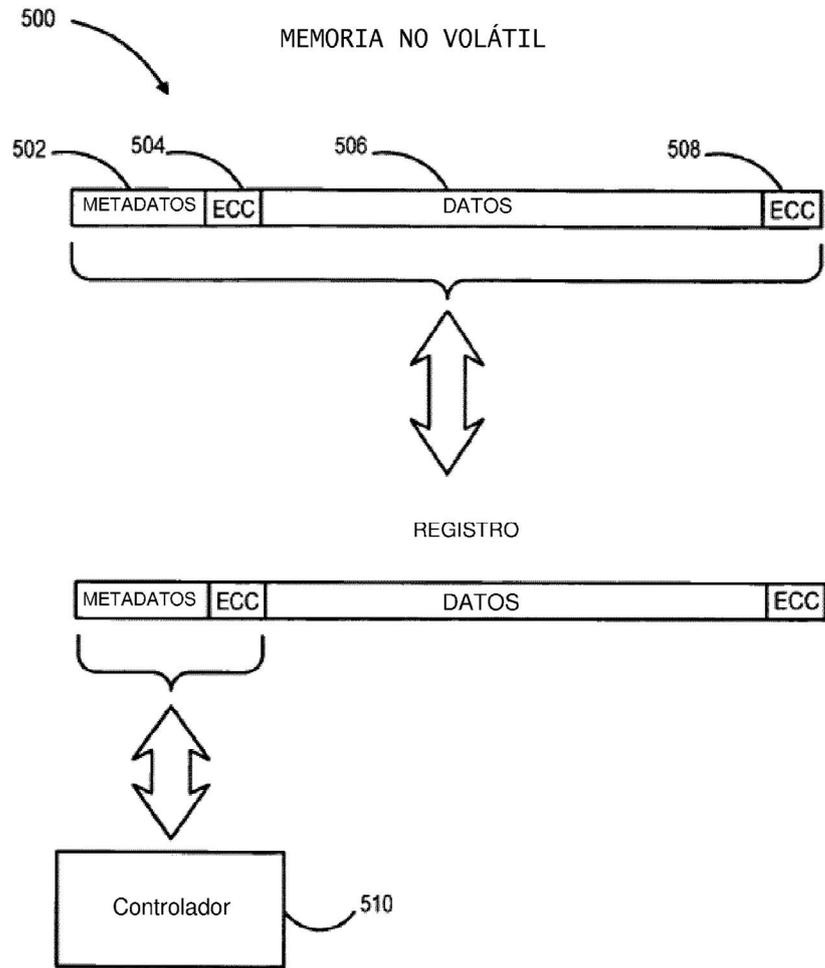


FIG. 5