

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 721 751**

51 Int. Cl.:

G11C 13/00 (2006.01)

G11C 11/00 (2006.01)

G11C 29/24 (2006.01)

G11C 29/42 (2006.01)

G11C 29/44 (2006.01)

G11C 29/50 (2006.01)

G06F 11/10 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **16.03.2016** E 16160734 (6)

97 Fecha y número de publicación de la concesión europea: **30.01.2019** EP 3098815

54 Título: **Métodos y sistemas para detectar y corregir errores en una memoria no volátil**

30 Prioridad:

29.05.2015 US 201514724899

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

05.08.2019

73 Titular/es:

**WINBOND ELECTRONICS CORP. (100.0%)
No. 8 Keya 1st Rd., Daya District, Central Taiwan
Science Park
Taichung City, Taiwan, TW**

72 Inventor/es:

**LIEN, CHUEN-DER;
SHIEH, MING-HUEI y
LIN, CHI-SHUN (ELBERT)**

74 Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

ES 2 721 751 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

Métodos y sistemas para detectar y corregir errores en una memoria no volátil

Campo técnico

5 La presente divulgación se refiere, en general, a una memoria no volátil. Más específicamente, y sin limitación, las realizaciones de ejemplo descritas en el presente documento se refieren a métodos y sistemas para detectar y corregir errores en una memoria no volátil.

Antecedentes

10 Memorias de semiconductores no volátiles retienen, en general, los datos almacenados incluso cuando no se alimentan. Las memorias no volátiles basadas en transistores, tal como la memoria flash y la memoria de solo lectura programable y borrable eléctricamente (EEPROM), ofrecen tiempos de acceso de lectura rápidos y resistencia a los golpes, lo que los hace deseables en diversas aplicaciones. Algunas aplicaciones de memorias no volátiles incluyen el almacenamiento de datos en dispositivos informáticos, teléfonos móviles, reproductores de audio portátiles y otros productos electrónicos de consumo.

15 La **figura 1** ilustra un diagrama de circuito de un ejemplo de una matriz de memoria no volátil basada en transistores 100. La matriz de memoria no volátil basada en transistores 100 incluye una pluralidad de celdas de memoria no volátiles basadas en transistores 110. Cada celda de memoria no volátil basada en transistores 110 está asociada con una línea de palabras (por ejemplo, WL0 a WLn) y una línea de bits (por ejemplo, BL0 a BLm). La matriz de memoria no volátil basada en transistores 100 puede estar formada por memorias no volátiles que almacenan datos que usan uno o más transistores como un elemento de almacenamiento. Las memorias no volátiles basadas en transistores incluyen una memoria de solo lectura programable (PROM), una memoria de solo lectura programable eléctricamente (EPROM), una memoria EEPROM, una memoria flash o una memoria eFUSE.

20 Como se muestra en la **figura 1**, cuando la matriz de memoria no volátil basada en transistores 100 se implementa usando EPROM, EEPROM o una memoria flash, cada celda de memoria no volátil basada en transistores 110 incluye un transistor de efecto de campo de semiconductores de óxido metálico de puerta flotante (MOSFET). El MOSFET de puerta flotante (FGMOS) puede almacenar la carga en una puerta flotante aislada eléctricamente. El aislamiento eléctrico permite que la puerta flotante retenga una carga durante largos períodos de tiempo sin alimentación. Una puerta flotante completamente cargada puede representar un estado lógico "0" y una puerta flotante sin carga puede representar un estado lógico "1", o viceversa.

25 Las tecnologías de memoria no volátil emergentes están desarrollándose para hacer frente a diversas limitaciones asociadas con las memorias no volátiles basadas en transistores, tales como la memoria flash. Por ejemplo, la mayoría de las memorias flash disponibles en el mercado sufren una resistencia a la escritura relativamente baja. Una memoria flash típica puede resistir hasta 1×10^5 ciclos de escritura (también denominados ciclos de programa/borrado); mientras que algunas memorias no volátiles emergentes, tales como la memoria de acceso aleatorio magnético (MRAM), pueden resistir hasta 1×10^{12} ciclos de escritura. Como otro ejemplo, las matrices de memoria flash pueden sufrir problemas de escala, tal como la alteración de lectura (ciclos de lectura secuenciales que hacen que las celdas cercanas cambien con el tiempo) y reducciones en la resistencia a la escritura.

30 Las memorias no volátiles emergentes, sin embargo, tienen defectos. Por ejemplo, las altas temperaturas de operación pueden provocar errores de datos, tal como los bits cambiados en algunas memorias no volátiles emergentes, tal como las memorias no volátiles resistivas. Dichos errores de datos pueden conducir a fallos en el sistema, corrupción de datos y/o vulnerabilidades de seguridad. Además, someter las memorias no volátiles emergentes a altas temperaturas de almacenamiento y fabricación puede provocar problemas de retención de datos tales como un desfase de datos sistemático, una pérdida de datos, una corrupción de datos significativa y una disminución del tiempo de retención de datos. El documento US2006/039196 A1 desvela la aplicación de una actualización completa o parcial de los bloques de datos de memoria flash, que se decide por el número de errores de bits de datos y los recursos de sistema disponibles.

Sumario

35 De acuerdo con una realización de ejemplo, un sistema de memoria comprende una matriz de memoria no volátil resistiva configurada para almacenar datos, bits de anticipación, y bits de código de corrección de errores (ECC) que corresponden a los datos almacenados y a los bits de anticipación almacenados, en el que los bits de anticipación indican la calidad de los bits de datos almacenados en la matriz de memoria no volátil, y un número total de los bits de anticipación es menor que un número total de los bits de datos y un controlador de memoria. El controlador de memoria está configurado para realizar una operación de lectura en los bits de anticipación y en los bits de ECC para detectar errores de bit de datos en unidades de los datos almacenados y errores de bit de anticipación entre los

bits de anticipación almacenados, detectar un número de errores de bit de anticipación, comparar el número de errores de bit de anticipación con un número umbral para los errores de bit de anticipación, realizar una actualización fuerte en todos los bits de datos y los bits de anticipación almacenados en la matriz de memoria no volátil resistiva cuando el número de errores de bit de anticipación es igual o superior al número umbral para los errores de bit de anticipación, y realizar una actualización débil actualizando solo las unidades de los datos almacenados que tienen los errores de bit de datos y los bits de anticipación almacenados que tienen los errores de bit de anticipación cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación.

De acuerdo con otro ejemplo de realización, un método para corregir errores en un sistema de memoria que incluye una matriz de memoria no volátil resistiva, comprende realizar una operación de lectura en los bits de anticipación y en los bits de código de corrección de errores (ECC) para detectar errores de bit de datos en unidades de datos almacenados en la matriz de memoria no volátil resistiva y errores de bit de anticipación entre los bits de anticipación almacenados en la matriz de memoria no volátil resistiva, en el que los bits de ECC se almacenan en la matriz de memoria resistiva no volátil y corresponde a las unidades de los datos almacenados y los bits de anticipación almacenados, y los bits de anticipación indican la calidad de los bits de datos almacenados en la matriz de memoria no volátil, y un número total de los bits de anticipación es menor que un número total de los bits de datos; y además comprende detectar un número de los errores de bit de anticipación, comparar el número de los errores de bit de anticipación con un número umbral para los errores de bit de anticipación, realizar una actualización fuerte de todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva cuando el número de los errores de bit de anticipación es igual o superior al número umbral para los errores de bit de anticipación y realizar una actualización débil actualizando solo las unidades de los datos almacenados que tienen los errores de bit de datos y los bits de anticipación que tienen los errores de bit de anticipación cuando el número de errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación.

Breve descripción de los dibujos

Los dibujos adjuntos, que se incorporan en y constituyen parte de esta memoria descriptiva, y junto con la descripción, ilustran y sirven para explicar los principios de diversas realizaciones a modo de ejemplo.

la figura 1 es un diagrama de circuito de una matriz de memoria no volátil basada en transistores de ejemplo;

la figura 2 es un diagrama de circuito de una matriz de memoria no volátil resistiva de ejemplo para implementar las realizaciones consecuentes con la presente divulgación;

la figura 3 es un diagrama de bloques de una matriz de memoria no volátil resistiva de ejemplo para implementar las realizaciones consecuentes con la presente divulgación;

la figura 4 es un diagrama de bloques de un sistema de memoria de ejemplo para implementar las realizaciones consecuentes con la presente divulgación;

la figura 5 es un diagrama de flujo que representa un método de ejemplo para corregir errores en un sistema de memoria consecuente con las realizaciones de la presente divulgación;

la figura 6 es un diagrama de flujo que representa otro método de ejemplo para corregir errores en un sistema de memoria consecuente con las realizaciones de la presente divulgación;

la figura 7 es un diagrama de flujo que representa un método de ejemplo para realizar una operación de actualización fuerte consecuente con las realizaciones de la presente divulgación;

la figura 8 es un diagrama de flujo que representa un método de ejemplo para realizar una operación de actualización débil consecuente con las realizaciones de la presente divulgación;

la figura 9 ilustra una comparación entre una operación de lectura normal y una operación de lectura de margen consecuente con las realizaciones de la presente divulgación.

Descripción detallada de las realizaciones de ejemplo

Las realizaciones de la presente divulgación proporcionan métodos y sistemas mejorados para la detección y corrección de errores en las memorias no volátiles emergentes. Muchas memorias no volátiles emergentes tienen ventajas significativas sobre las memorias no volátiles basadas en transistores en términos de velocidad de escritura y resistencia, consumo de energía, retención de datos, seguridad de datos y acceso aleatorio a nivel de bytes. Sin embargo, algunas memorias no volátiles emergentes, tales como las memorias no volátiles resistivas, pueden ser susceptibles a problemas de retención y corrupción de datos, debido a las altas temperaturas y/o a la interferencia

5 electromagnética (EMI). Las memorias no volátiles resistivas pueden incluir, por ejemplo, cualquier memoria no volátil que use diferentes estados de resistencia para almacenar datos. Los ejemplos de memorias no volátiles resistivas incluyen una memoria de acceso aleatorio de cambio de fase (PCRAM), una memoria de acceso aleatorio magnético (MRAM), una memoria de celda de metalización programable (PMC) tal como una memoria de acceso aleatorio de puente conductor (CBRAM) y una memoria de acceso aleatorio resistiva (RRAM).

10 La PCRAM almacena datos usando las diferencias en la resistencia entre fases amorfas y cristalinas de un material basado en calcogenuro. La fase amorfa corresponde a una fase de alta resistencia y la fase cristalina corresponde a una fase de baja resistencia. La MRAM almacena datos usando una capa de barrera de túnel entre dos capas ferromagnéticas para conmutar entre diferentes estados de resistencia. Se logra un estado de baja resistencia cuando las direcciones de magnetización de las dos capas ferromagnéticas son paralelas. Se logra un estado de alta resistencia cuando las direcciones de magnetización de las dos capas ferromagnéticas son antiparalelas. Las celdas CBRAM incluyen una película delgada de electrolito entre dos electrodos de metal sólido. Una celda CBRAM almacena datos en diferentes estados de resistencia formando un nanohilo entre los electrodos de metal en la película de electrolito. La ausencia del nanohilo corresponde a un estado de alta resistencia y la presencia del nanohilo corresponde a un estado de baja resistencia. La RRAM almacena los datos como dos o más estados de resistencia basados en la conmutación resistiva en óxidos de metal de transición. Similar a las celdas CBRAM, una celda RRAM incluye una estructura metálica aislante de metal. Se representan diferentes estados lógicos produciendo un rastro de defectos conductores (denominado como "filamento") en la capa aislante. La ausencia del filamento corresponde a un estado de alta resistencia y la presencia del filamento corresponde a un estado de baja resistencia. Los respectivos estados de resistencia alta y baja de las diversas memorias resistivas sirven para almacenar datos que representan los valores lógicos primero y segundo, por ejemplo, "0" y "1".

25 Las memorias no volátiles pueden clasificarse para operar dentro de diversos intervalos de temperatura. Por ejemplo, una memoria no volátil puede clasificarse para operar a temperaturas ambiente de 0 a 85 °C (clasificación comercial), -40 a 125 °C (clasificación industrial/automotriz), o -55 a 125 °C (clasificación militar). Las clasificaciones de temperatura de almacenamiento normalmente varían hasta 150 °C. Las memorias no volátiles también pueden experimentar altas temperaturas durante el montaje en la placa de CI. Por ejemplo, un proceso típico de soldadura por onda puede alcanzar temperaturas de hasta 250 °C. Las memorias no volátiles resistivas pueden experimentar un rendimiento degradado cerca de las partes superiores de estos intervalos de operación. Por ejemplo, los errores de datos, tales como el cambio de bits, pueden ocurrir en celdas de memoria no volátiles resistivas a altas temperaturas de operación ambiente. Además, los datos escritos en una memoria no volátil resistiva antes de la exposición a una temperatura alta, tal como la provocada por un proceso de soldadura por onda, pueden experimentar errores de datos significativos, tal como el desfase de datos sistemático y la disminución del tiempo de retención de datos.

35 Las realizaciones de la presente divulgación proporcionan mejoras en la operación de memoria a alta temperatura y la retención de datos en las memorias no volátiles resistivas, permitiendo de este modo mejoras de rendimiento en los sistemas y/o en los circuitos integrados (CI) que incorporan tales memorias. Las realizaciones de la presente divulgación proporcionan una memoria no volátil resistiva que está configurada para renovar sus datos almacenados usando códigos de corrección de errores (ECC). La renovación de datos se realiza en diversos puntos de la operación de sistema de memoria, tal como durante una inicialización de encendido o en modo inactivo, o en respuesta a una orden del sistema. La renovación de datos puede realizarse leyendo, corrigiendo y escribiendo los datos almacenados. Algunas realizaciones proporcionan una renovación de datos mejorada con una reescritura más fuerte realizada en el caso de bits de ECC insuficientes o un desfase de datos sistemático. La renovación de datos mejorada puede actualizar todos los datos almacenados en la memoria resistiva de tal manera que el margen de memoria se recupere después del montaje en la placa y/o el almacenamiento a alta temperatura. Otras realizaciones pueden proporcionar combinaciones de técnicas de renovación de datos.

A continuación, se hará referencia en detalle a las realizaciones a modo de ejemplo implementadas de acuerdo con la divulgación, cuyos ejemplos se ilustran en los dibujos adjuntos. Siempre que sea posible, se usarán los mismos números de referencia en todos los dibujos para referirse a partes iguales o similares.

50 La **figura 2** ilustra un diagrama de circuito de una matriz de memoria no volátil resistiva de ejemplo 200 para implementar las realizaciones consecuentes con la presente divulgación. La matriz de memoria no volátil resistiva 200 puede formarse usando cualquiera de las memorias no volátiles resistivas descritas anteriormente, tales como PCRAM, RRAM, MRAM o CBRAM. Además, la matriz de memoria no volátil resistiva 200 puede comprender una matriz de memoria bidimensional o una matriz de memoria tridimensional.

55 Como se muestra en la **figura 2**, la matriz de memoria no volátil resistiva 200 incluye una pluralidad de líneas de palabras (WL0 a WLn), líneas de bits (BL0 a BLm) y celdas de memoria resistiva 210. Cada celda de memoria resistiva 210 está asociada con una línea de palabras y una línea de bits. Cada celda de memoria resistiva 210 incluye un elemento de selección 212 y un elemento de almacenamiento resistivo 214. Las celdas de memoria resistiva 210 pueden implementarse mediante cualquiera de las memorias no volátiles resistivas descritas anteriormente, tales como una celda PCRAM, una celda RRAM, una celda MRAM o una celda CBRAM. El elemento

de selección 212 se comporta como un conmutador de selección para controlar el acceso a la celda de memoria resistiva 210. El elemento de selección 212 puede implementarse, por ejemplo, mediante un diodo, un transistor de efecto de campo de semiconductor de óxido de metal (MOSFET) o un transistor de unión bipolar (BJT). En algunas realizaciones, el elemento de selección 212 puede ser opcional, por ejemplo, cuando la celda de memoria resistiva 210 se implementa mediante una celda de RRAM.

La **figura 3** ilustra un diagrama de bloques de la matriz de memoria no volátil resistiva 200 para implementar las realizaciones consecuentes con la presente divulgación. Como se muestra en la **figura 3**, la matriz de memoria no volátil resistiva 200 incluye una pluralidad de tipos de celdas. Por ejemplo, las celdas de datos 310 están configuradas para almacenar bits de datos. Cada celda de datos 310 almacena uno o más bits de datos. Los bits de datos pueden escribirse y accederse a los mismos en diversas unidades de almacenamiento, tales como páginas, bloques, sectores, palabras o cualquier otra configuración de almacenamiento conocida en la técnica. Cada unidad de almacenamiento puede comprender cualquier número de bits de datos.

En algunas realizaciones, la matriz de memoria no volátil resistiva 200 incluye celdas de ECC 320. Cada celda de ECC 320 está configurada para almacenar un bit de ECC (o bits) para detectar y corregir errores o bits de datos defectuosos almacenados en las celdas de datos 310 y en las celdas de anticipación 330. Los errores pueden ser errores de software provocados, por ejemplo, por una EMI, una operación a alta temperatura, un almacenamiento a alta temperatura, unas temperaturas de soldadura por onda de montaje en placa, etc. Los errores de software suelen ser transitorios y no siempre se repiten. Los errores pueden ser errores de hardware provocados por defectos de hardware en la celda de memoria. Los errores de hardware suelen ser repetibles y siempre devuelven resultados incorrectos. Los bits de ECC almacenados pueden incluir códigos de Hamming, códigos Bose, Chaudhuri y Hocquenghem (BCH), bits de paridad, códigos Reed-Solomon, códigos Turbo, códigos de verificación de paridad de baja densidad (LDPC), códigos convolucionales o cualquier otro código de corrección de errores conocido en la técnica.

El número de bits de ECC almacenados en las celdas de ECC 320 depende de un esquema de codificación de ECC. El esquema de codificación de ECC puede asignar un número de bits de ECC por unidad de almacenamiento (por ejemplo, por página, por bloque, por sector, por palabra, por byte, por bit, etc.) de datos almacenados. Por ejemplo, cuando 16 palabras de datos se almacenan en las celdas de datos 310 y el esquema de codificación de ECC asigna cuatro bits de ECC por palabra de datos, el número total de bits de ECC almacenados en las celdas de ECC 320 es 64, uno o más bits de ECC por celda de ECC 320. El número de bits de ECC asignados a cada unidad de almacenamiento puede ser flexible y puede depender de los requisitos de diseño, las capacidades del sistema, los requisitos de tiempo y rendimiento, etc. El número de errores de bit de datos que pueden corregirse por los bits de ECC almacenados se determina como $i - 1$ donde i es el número de bits de ECC asignados a cada unidad de almacenamiento. Por ejemplo, cuando las celdas de ECC 320 almacenan tres bits de ECC por palabra de datos, los tres bits de ECC pueden corregir dos errores de bit de datos por palabra de datos.

En algunas realizaciones, la matriz de memoria no volátil resistiva 200 incluye unas celdas de anticipación 330. Las celdas de anticipación 330 están configuradas para almacenar bits de anticipación para detectar la corrupción de datos sustancial o el desfase de datos sistemático de los datos almacenados en las celdas de datos 310. Cada celda de anticipación 330 está configurada para almacenar uno o más bits de anticipación. Los bits de anticipación almacenados en las celdas de anticipación 330 sirven como indicadores de la corrupción de datos o del desfase sistemático debido a, por ejemplo, una EMI, unas temperaturas de soldadura por onda de montaje en placa o un almacenamiento a alta temperatura de la matriz de memoria no volátil resistiva 200. En otras palabras, la presencia de errores significativos en los bits de anticipación almacenados indica probablemente que los datos almacenados en las celdas de datos 310 también contienen errores significativos.

El número de bits de anticipación almacenados en las celdas de anticipación 330 depende de un número de factores tales como el número total de celdas de memoria incluidas en la matriz de memoria no volátil resistiva 200, el número total de bits de datos almacenados en las celdas de datos 310, la importancia de la confiabilidad del sistema de memoria, la cantidad de rendimiento del sistema de memoria asignable a la detección de desfase sistemático, etc. Los ejemplos del número total de bits de anticipación almacenados en las celdas de anticipación 330 incluyen 16 bits o 32 bits. El número total de bits de anticipación almacenados en las celdas de anticipación 330 puede ser pequeño en comparación con el número de bits de datos almacenados en las celdas de datos 310. Por consiguiente, usar un número pequeño de bits de anticipación como un proxy para la calidad de un gran número de bits de datos mejora la eficacia del sistema de memoria y reduce la cantidad de recursos del sistema de memoria necesarios para la detección de errores.

En algunas realizaciones, las celdas de datos 310, las celdas de ECC 320, y/o las celdas de anticipación 330 están configuradas para almacenar múltiples bits por celda. Por ejemplo, pueden usarse celdas multinivel en una matriz de memoria resistiva 200. Las celdas multinivel usan tres o más estados de resistencia diferentes para almacenar dos o más bits de datos. Por ejemplo, una celda multinivel con cuatro estados de resistencia diferentes puede almacenar dos bits, usando cada bit dos estados de resistencia para representar un "0" y "1" lógicos. Sin embargo, aumentar el número de bits almacenados por celda puede resultar en una mayor tasa de error de bits y, por lo tanto, requiere un

mayor número de bits de ECC por unidad de almacenamiento para corregir esos errores.

Las celdas de datos 310, las celdas de ECC 320, y las celdas de anticipación 330 pueden estar incluidas en la misma matriz de memoria no volátil resistiva 200, almacenadas en diferentes matrices de memoria no volátil resistivas 200 formadas en el mismo chip semiconductor, almacenadas en diferentes matrices de memoria no volátil resistivas 200 formadas en chips semiconductores separados que están incluidos en el mismo paquete de circuito integrado (CI), almacenadas en diferentes matrices de memoria no volátil resistivas 200 formadas en chips semiconductores separados incluidos en paquetes de CI separados, etc.

La **figura 4** ilustra un diagrama de bloques de un sistema de memoria de ejemplo 400 para implementar las realizaciones consecuentes con la presente divulgación. Como se muestra en la **figura 4**, el sistema de memoria 400 incluye una matriz de memoria no volátil resistiva 200 y un controlador de memoria 410. Se apreciará a partir de la presente divulgación que el número y la disposición de estos componentes son solo a modo de ejemplo y se proporciona con fines ilustrativos. Pueden usarse otras disposiciones y números de componentes sin alejarse de las enseñanzas y realizaciones de la presente divulgación. Como ejemplo, el controlador de memoria 410 y la matriz de memoria no volátil basada en resistividad 200 pueden formarse en el mismo chip semiconductor, en chips semiconductores separados que se incluyen en el mismo paquete de CI, en chips semiconductores separados incluidos en paquetes de CI independientes o en cualquier configuración conocida en la técnica. Como otro ejemplo, el controlador de memoria 410 puede controlar múltiples matrices de memoria no volátil resistivas 200 o una combinación de matrices de memoria no volátil basadas en transistores y matrices de memoria no volátil resistivas 200.

En algunas realizaciones, el controlador de memoria 410 incluye una interfaz de entrada/salida (E/S) 411, un decodificador de fila (también denominado como un decodificador de línea de palabras) 412, un decodificador de columna 413, un amplificador de detección y controlador de escritura 414, un controlador de código de corrección de errores (ECC) 415 y un controlador de anticipación 416. Los componentes 411-416, incluidos en el controlador 410 de memoria, pueden implementarse mediante combinaciones de hardware y software. Por ejemplo, los componentes 411-416 pueden implementarse mediante un microprocesador, un circuito integrado de aplicación específica (ASCI), una matriz de puertas programables (PGA) o una matriz de puertas programables en campo (FPGA), circuitos eléctricos, o cualquier combinación de los mismos. Además, algunas o todas las funciones de los componentes 411-416 pueden realizarse mediante la misma combinación de hardware/software. Por ejemplo, las funciones del controlador de ECC 415 y el controlador de anticipación 416 pueden realizarse por el mismo microprocesador, ASCI, PGA, FPGA, CPLD o circuito lógico personalizado, y todos los controladores pueden o no compartir el mismo algoritmo de escritura.

La E/S 411 recibe señales de entrada desde fuentes exteriores y convierte las señales de entrada para que puedan comprenderse y aceptarse por los componentes incluidos en el controlador de memoria 410. Por ejemplo, la E/S 411 puede recibir órdenes tales como órdenes de lectura, órdenes de escritura, órdenes de renovación de datos, etc.

En una operación de lectura normal, la E/S 411 transmite las coordenadas de fila y columna al decodificador de fila 412 y al decodificador de columna 413, respectivamente. El decodificador de fila 412 decodifica las coordenadas de la fila y proporciona una tensión de detección a una línea de palabras correspondiente. La tensión de detección activa el elemento de selección 212 de cada celda de memoria resistiva 210 conectada a la línea de palabras. El decodificador de columna 413 decodifica las coordenadas de la columna y conecta una línea de bits correspondiente al amplificador de detección y controlador de escritura 414. El amplificador de detección y controlador de escritura 414 incluye un circuito de lectura y un circuito de escritura. El circuito de lectura mide la corriente a través de la celda de memoria resistiva 210 en la intersección de línea de palabras/línea de bits apropiada, y determina el valor lógico correspondiente a la corriente medida. Durante un ciclo de lectura, la corriente de celda de memoria se distribuye en cualquiera de los lados de un valor de corriente umbral (generalmente denominado como corriente de detección). Los valores de corriente mayores que el valor umbral corresponden a un "0" lógico y los valores de corriente menores que el valor umbral corresponden a un "1" lógico o viceversa. En consecuencia, el amplificador de detección y controlador de escritura 414 mide la corriente a través de la celda de memoria resistiva 210, compara la corriente medida con uno o más valores de corriente umbral e identifica el valor lógico basándose en si la corriente medida es mayor o menor que el valor de corriente umbral. El amplificador de detección y controlador de escritura 414 proporciona el valor lógico identificado a la E/S 411 como una salida de la operación de lectura. La E/S 411 transmite la salida a un destino exterior o retiene la salida hasta que lo solicite una fuente exterior. En algunas realizaciones, puede leerse una pluralidad de celdas de memoria resistiva 210 asociadas con la misma línea de palabras activada en una sola operación de lectura. Cuando el decodificador de fila 412 proporciona la tensión de detección a la línea de palabras correspondiente, se activa cada celda de memoria resistiva 210 en la línea de palabras. Por lo tanto, el circuito de lectura mide la corriente a través de cada celda de memoria resistiva activada 210 en las líneas de bits asociadas con esas celdas.

En algunas realizaciones, el controlador de memoria 410 realiza una operación de lectura de margen en las celdas de memoria 210 cuando se desea una mayor fiabilidad de memoria y precisión de lectura. Por ejemplo, el controlador de memoria 410 realiza una operación de lectura de margen cuando lee de las celdas de anticipación

330. Ya que los bits de anticipación sirven como un proxy para la calidad de un gran número de bits de datos, se usa una lectura de margen para mejorar la detección de la degradación en la calidad del bit de anticipación (y, por lo tanto, esa corrupción de datos es inminente). Como otro ejemplo, el controlador de memoria 410 realiza una operación de lectura de margen en las celdas de datos 310 después de una operación de actualización fuerte o débil (que se describe a continuación haciendo referencia a las **figuras 7 y 8**) para verificar que las celdas de datos 310 se han actualizado correctamente.

En lugar de usar un único valor de corriente umbral, el amplificador de detección y controlador de escritura 414 determina el valor lógico correspondiente a la corriente medida comparando la corriente medida para una pluralidad de valores de corriente umbral durante una operación de lectura de margen. Los valores de corriente umbral incluyen un valor de corriente umbral superior y un valor de corriente umbral inferior. Los valores de corriente umbral superior e inferior se distribuyen a ambos lados del valor de corriente umbral normal. Por ejemplo, cuando un valor de corriente umbral de un ciclo de escritura normal se selecciona como 25 μA , los valores de corriente umbral superior e inferior pueden seleccionarse como 40 μA y 10 μA , respectivamente. Un valor de corriente medido mayor que el valor umbral de corriente superior corresponde a un "0" lógico y un valor de corriente medido menor que el valor de corriente umbral inferior corresponde a un "1" lógico, o viceversa.

En la operación normal de escritura, la E/S 411 recibe datos desde una fuente exterior y una celda de memoria resistiva 210 direcciona donde se almacenan los datos. La E/S 411 convierte la dirección en coordenadas de fila y columna y envía las coordenadas al decodificador de fila 412 y al decodificador de columna 413, respectivamente. El decodificador de fila 412 decodifica las coordenadas de fila y proporciona una tensión de escritura a una línea de palabras correspondiente. El decodificador de columna 413 decodifica las coordenadas de columna y conecta una línea de bits correspondiente al amplificador de detección y controlador de escritura 414. La E/S 411 transmite instrucciones al amplificador de detección y controlador de escritura 414 para escribir un "0" o un "1" en la celda de memoria resistiva 210 localizada en la intersección de la línea de palabras activada y la línea de bits. El circuito de escritura del amplificador de detección y controlador de escritura 414 cambia la resistencia del elemento resistivo 212 a un estado de alta resistencia o baja resistencia en función de si se debe almacenar un "1" o un "0".

En algunas realizaciones, el controlador de memoria 410 incluye un controlador de ECC 415. El controlador de ECC 415 detecta errores de bits de datos entre los datos almacenados en las celdas de datos 310 y los errores de bit de anticipación entre los bits almacenados en las celdas de anticipación 330. En la operación de escritura descrita anteriormente, el controlador de ECC 415 genera bits de ECC correspondientes a los datos que deben escribirse en las celdas de memoria resistiva 210. El controlador de ECC 415 también genera bits de ECC correspondientes a los bits de anticipación que deben escribirse en las celdas de memoria resistiva 210. El número de bits de ECC generados para cada unidad de almacenamiento de datos o bits de anticipación se determina de acuerdo con uno de los esquemas de codificación de ECC descritos anteriormente. Los bits de ECC generados se transmiten desde el controlador de ECC 415 al amplificador de detección y controlador de escritura 414. El circuito de escritura del amplificador de detección y controlador de escritura 414 escribe los datos en las celdas de datos 310, los bits de anticipación en las celdas de anticipación 330 y los bits de ECC generados en las celdas de ECC 320.

En la operación de lectura descrita anteriormente, el controlador de ECC 415 lee los datos almacenados en las celdas de datos 310 y que corresponden a los bits de ECC almacenados en las celdas de ECC 320. El controlador de ECC 415 usa los bits de ECC para realizar una detección de error en los datos de lectura para detectar los errores de bit de datos. En una realización, el controlador de ECC 415 determina el número de errores de bit de datos por unidad de almacenamiento de datos. Si el número de errores de bit de datos por unidad de almacenamiento es menor que el número de bits de ECC correspondiente a la unidad de almacenamiento, el controlador de ECC 415 usa los bits de ECC para corregir los errores de bit de datos. El controlador de ECC 415 transmite los datos corregidos al amplificador de detección y controlador de escritura 414 y el circuito de escritura vuelve a escribir los datos en las celdas de datos 310 y en las celdas de ECC 320 respectivas. En algunas realizaciones, el circuito de escritura escribe toda la unidad de almacenamiento de datos, incluyendo los bits de datos corregidos, en las celdas de datos 310. En otras realizaciones, el circuito de escritura escribe solo los bits corregidos en las celdas de datos 310.

En algunas realizaciones, el controlador de memoria 410 incluye un controlador de anticipación 416. El controlador de anticipación 416 detecta una corrupción de datos significativa y/o el desfase de datos sistemático de los datos almacenados en las celdas de datos 310. Por ejemplo, el controlador de anticipación 416 lee los bits de anticipación almacenados en las celdas de anticipación 330 y que corresponden a los bits de ECC almacenados en las celdas de ECC 320, y determina el número de errores de bit de anticipación almacenados en las celdas de anticipación 330 usando los bits de ECC. Si el número de errores de bit de anticipación es igual o superior a un número umbral de errores de bit, el controlador de anticipación 416 inicia una actualización fuerte de los datos almacenados en las celdas de datos 310. Un número umbral de ejemplo de errores de bit de anticipación puede ser el 50 % de los bits de anticipación almacenados en las celdas de anticipación 330. Por lo tanto, en el presente ejemplo, cuando hay 32 bits de anticipación almacenados en las celdas de anticipación 330, el controlador de anticipación 416 inicia una actualización fuerte cuando el controlador de anticipación 416 determina que el número de errores de bit de anticipación es igual o mayor que 16 bits.

La actualización fuerte incluye leer todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva 200, corregir cualquiera de los errores de bit corregibles, reemplazar cualquier bit que tenga errores incorregibles, y escribir de nuevo todos los bits de datos y de anticipación (incluidos los bits corregidos y reemplazados) en la matriz de memoria no volátil resistiva 200. Específicamente, los bits de datos y de anticipación que no tienen errores se actualizan en la operación de actualización fuerte. Cuando los bits de anticipación almacenados contienen errores significativos, indicando de este modo que es probable que los datos almacenados en las celdas de datos 310 también contengan errores significativos, los bits de datos y de anticipación que no tienen errores pueden haber perdido un margen de celda de memoria significativo y, por lo tanto, pueden estar cerca de fallar. En consecuencia, la operación de actualización fuerte actualiza todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva 200 con el fin de restaurar el margen dentro de las celdas de memoria que almacenan los bits de datos y de anticipación. La operación de actualización fuerte se describe con mayor detalle a continuación haciendo referencia a la **figura 7**.

Si el número de errores de bit de anticipación es menor que el número umbral de errores, el controlador de anticipación 416 inicia una actualización débil de los datos almacenados en las celdas de datos 310. La actualización débil incluye leer todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva 200, corregir cualquiera de los errores de bit corregibles, reemplazar cualquiera de los bits que tenga errores no corregibles, y escribir de nuevo solo los datos corregidos/reemplazados y los bits de anticipación en la matriz de memoria no volátil resistiva 200. Por consiguiente, cuando los bits de anticipación no contienen errores significativos, los recursos del sistema de memoria usados para realizar la operación de actualización pueden minimizarse escribiendo solo los datos corregidos/reemplazados y los bits de anticipación en lugar de la totalidad de la matriz de memoria no volátil resistiva 200 de los bits de datos y de anticipación. La operación de actualización débil se describe con mayor detalle a continuación haciendo referencia a la **figura 8**.

La **figura 5** representa un diagrama de flujo de un método de ejemplo 500, consecuente con algunas realizaciones y aspectos de la presente divulgación. El método 500 puede implementarse, por ejemplo, para detectar y corregir errores en la memoria no volátil resistiva. En algunas realizaciones, el método 500 puede implementarse mediante un sistema de memoria tal como, por ejemplo, el sistema de memoria 400 ilustrado en la **figura 4**.

En algunas realizaciones, el método de ejemplo 500 incluye realizar una operación de lectura de margen en los bits de anticipación almacenados en una matriz de memoria no volátil resistiva (510). Un controlador de memoria (por ejemplo, el controlador de memoria 410 de **figura 4**) incluido en el sistema de memoria lee los bits de anticipación almacenados en celdas de memoria (por ejemplo, en las celdas de anticipación 330 de la **figura 3**) de una matriz de memoria resistiva (por ejemplo, la matriz de memoria resistiva 200 de las **figuras 2-4**) usando la operación de lectura de margen descrita anteriormente haciendo referencia a la **figura 4**. El controlador de memoria lee los bits de anticipación en diversas unidades de almacenamiento, tales como páginas, bloques, sectores, palabras o cualquier otra unidad de almacenamiento conocida en la técnica. Cada unidad de almacenamiento puede incluir cualquier número de bits de anticipación.

En algunas realizaciones, el método de ejemplo 500 incluye determinar un número de errores entre los bits de anticipación almacenados en una matriz de memoria no volátil resistiva (520). El controlador de memoria lee los bits de ECC almacenados en celdas de memoria (por ejemplo, las celdas de ECC 320 de la **figura 3**) de la matriz de memoria resistiva y usa los bits de ECC almacenados para detectar errores en los bits de anticipación. El controlador de memoria detecta errores determinando qué bits de anticipación han cambiado de estados lógicos (por ejemplo, de un "1" lógico a un "0" lógico y viceversa) debido a las altas temperaturas o a la EMI. Los bits de ECC almacenados pueden incluir códigos de Hamming, códigos Bose, Chaudhuri y Hocquenghem (BCH), bits de paridad, códigos Reed-Solomon, códigos Turbo, códigos de verificación de paridad de baja densidad (LDPC), códigos convolucionales o cualquier otro código de corrección de errores conocido en la técnica.

En algunas realizaciones, el controlador de memoria detecta errores entre los bits de anticipación almacenados comparando los bits de anticipación almacenados con un patrón. El controlador de memoria escribe los bits de anticipación en la matriz de memoria resistiva en uno de los numerosos patrones, tal como un patrón de tablero de ajedrez de estados lógicos alternos (por ejemplo, 01010101). El controlador de memoria está además programado para leer los bits de anticipación con patrón y compararlos con un patrón esperado. Se producen errores en los bits de anticipación cuando el patrón de lectura no coincide con el patrón esperado.

El controlador de memoria determina el número de errores de bit de anticipación durante diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede determinar el número de errores de bit de anticipación durante un ciclo de lectura, en respuesta al encendido del sistema de memoria, durante un modo inactivo del sistema de memoria, o en respuesta a una orden recibida. En el caso de un modo inactivo, el controlador de memoria puede determinar el número de errores de bit de anticipación durante un solo modo inactivo o durante un número predeterminado de modos inactivos. En consecuencia, el controlador de memoria realiza la detección de errores cuando el impacto en el rendimiento del sistema es mínimo. En el caso de un encendido, el controlador de memoria puede determinar el número de errores de bit de anticipación en respuesta a un solo encendido o a un número predeterminado de encendidos. El número predeterminado de encendidos puede

determinarse usando diversos criterios. Por ejemplo, el número de encendidos puede determinarse basándose en el número esperado de ciclos de alimentación que experimentará el sistema de memoria durante las pruebas de producción. En consecuencia, cualquier defecto sistemático en los datos que se precargan en el sistema de memoria durante la producción puede corregirse automáticamente antes de enviarse desde una instalación de producción. En el caso de una orden recibida, el sistema de memoria puede recibir la orden desde un sistema de procesamiento acoplado comunicativamente al sistema de memoria, desde un usuario tal como un operador de sistema o un técnico/ingeniero de pruebas, y/o desde un sistema de pruebas tal como un sistema de pruebas de desarrollo o de producción. La orden puede ser una orden para realizar la detección y corrección de errores, una orden de lectura, una orden de reinicio del sistema, una orden de reinicio de encendido (POR) o una orden de reactivación.

En algunas realizaciones, el controlador de memoria incluye un controlador de anticipación (por ejemplo, el controlador de anticipación 416 de la **figura 4**) que determina un número de errores entre los bits de anticipación almacenados en la matriz de memoria no volátil resistiva. Por ejemplo, el controlador de anticipación lee los bits de anticipación de la matriz de memoria resistiva y determina el número de errores entre los bits de anticipación.

En algunas realizaciones, el método de ejemplo 500 incluye comparar el número de errores de bit de anticipación con un número umbral de errores de bit de anticipación (530). El umbral puede determinarse basándose en un número de factores que incluyen, por ejemplo, los requisitos de confiabilidad del sistema de memoria, los requisitos de seguridad, los requisitos de cliente, la disponibilidad de los recursos de sistema de memoria, el tamaño de la matriz de memoria resistiva, etc. Un número de umbral de ejemplo de los errores de bit de anticipación es el 50 % de los bits de anticipación almacenados en las celdas de anticipación de la matriz de memoria no volátil resistiva. Cuando el controlador de memoria determina que el número de errores de bit de anticipación es menor que el número umbral de errores de bit de anticipación (530-sí), el controlador de memoria realiza una actualización débil de solo las unidades de almacenamiento de datos que tienen errores de bit de datos (540). Cuando el controlador de memoria determina que el número de errores de bit de anticipación es igual o mayor que el número umbral de errores de bit de anticipación (530-no), el controlador de memoria realiza una actualización fuerte de todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva (550).

La **figura 6** representa un diagrama de flujo de un método de ejemplo 600, consecuente con algunas realizaciones y aspectos de la presente divulgación. El método 600 puede implementarse, por ejemplo, para detectar y corregir errores en la memoria no volátil resistiva. En algunas realizaciones, el método 600 puede implementarse mediante un sistema de memoria tal como, por ejemplo, el sistema de memoria 400 ilustrado en la **figura 4**.

Al igual que en método de ejemplo 500, el método de ejemplo 600 incluye realizar una operación de lectura de margen en los bits de anticipación almacenados en una matriz de memoria no volátil resistiva (610) y determinar un número de errores entre los bits de anticipación almacenados en la matriz de memoria no volátil resistiva (620). En algunas realizaciones, el método de ejemplo 600 incluye comparar el número de errores de bit de anticipación con un primer número umbral de errores de bit de anticipación (630). Cuando el controlador de memoria determina que el número de errores de bit de anticipación es menor que el primer número umbral de errores de bit de anticipación (630-sí), el controlador de memoria no realiza una operación de actualización (640). Cuando el controlador de memoria determina que el número de errores de bit de anticipación es igual o mayor que el primer número umbral de errores de bit de anticipación (630-no), el controlador de memoria compara el número de errores de bit de anticipación con un segundo número umbral de errores de bit de anticipación (650). El segundo número umbral de errores de bit de anticipación es mayor que el primer umbral. El controlador de memoria realiza una actualización débil de los datos almacenados en la matriz de memoria no volátil resistiva (660) cuando el número de errores de bit de anticipación es menor que el segundo número umbral de errores de bit de anticipación (650-sí) y realiza una actualización fuerte de todos los bits de anticipación y datos almacenados en la matriz de memoria no volátil resistiva (670) cuando el número de los errores de bit de anticipación es igual o mayor que el segundo número umbral de errores de bit de anticipación (650-no).

Cuando se usan los umbrales superior e inferior de bits de anticipación, el sistema de memoria usa menos recursos debido a que el sistema de memoria está configurado para dar cabida a un número de errores de bit de anticipación por debajo del primer umbral. En consecuencia, las operaciones de actualización solo se realizan si las celdas de anticipación contienen errores por encima del primer umbral. El operador del sistema de memoria puede personalizar el primer umbral basándose en un número de factores que incluyen, por ejemplo, los requisitos de confiabilidad del sistema de memoria, los requisitos de seguridad, los requisitos de cliente, la disponibilidad de los recursos de sistema de memoria, el tamaño de la matriz de memoria resistiva, etc.

La **figura 7** muestra un diagrama de flujo de un método de ejemplo 700, consecuente con algunas realizaciones y aspectos de la presente divulgación. El método 700 puede implementarse, por ejemplo, para realizar una actualización fuerte de todos los bits de anticipación y los datos almacenados en la matriz de memoria no volátil resistiva. En algunas realizaciones, el método 700 puede implementarse mediante un sistema de memoria tal como, por ejemplo, el sistema de memoria 400 ilustrado en la **figura 4**.

En algunas realizaciones, el método de ejemplo 700 incluye leer los datos con corrección de ECC y detectar errores de bits de datos entre los datos almacenados en una matriz de memoria no volátil resistiva (710). Un controlador de memoria (por ejemplo, el controlador de memoria 410 de la **figura 4**) incluido en el sistema de memoria lee los datos almacenados en celdas de memoria (por ejemplo, las celdas de datos 310 de la **figura 3**) de una matriz de memoria resistiva (por ejemplo, la matriz de memoria resistiva 200 de las **figuras 2-4**) usando la operación de lectura normal descrita anteriormente. El controlador de memoria lee los datos de diversas unidades de almacenamiento, tal como páginas, bloques, sectores, palabras o cualquier otra unidad de almacenamiento conocida en la técnica. Cada unidad de almacenamiento puede comprender cualquier número de bits de datos. El controlador de memoria lee además los bits de ECC almacenados en celdas de memoria (por ejemplo, las celdas de ECC 320 de la **figura 3**) de la matriz de memoria no volátil resistiva y usa los bits de ECC almacenados para detectar errores de bit de datos en las unidades de almacenamiento de datos. El controlador de memoria detecta errores en los bits de datos en las unidades de almacenamiento de datos determinando qué bits de datos han cambiado de estados lógicos (por ejemplo, de un "1" lógico a un "0" lógico y viceversa) debido a las altas temperaturas o a la EMI. Los bits de ECC almacenados pueden incluir códigos de Hamming, códigos Bose, Chaudhuri y Hocquenghem (BCH), bits de paridad, códigos Reed-Solomon, códigos Turbo, códigos de verificación de paridad de baja densidad (LDPC), códigos convolucionales o cualquier otro código de corrección de errores conocido en la técnica.

El controlador de memoria determina el número de errores de bit de datos durante diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede determinar el número de errores de bits de datos durante un ciclo de lectura, en respuesta al encendido del sistema de memoria, durante un modo inactivo del sistema de memoria, o en respuesta a una orden recibida. En el caso de un modo inactivo, el controlador de memoria puede determinar el número de errores de bit de datos durante un solo modo inactivo o durante un número predeterminado de modos inactivos. En consecuencia, el controlador de memoria realiza la detección de errores cuando el impacto en el rendimiento de sistema es mínimo. En el caso de un encendido, el controlador de memoria puede determinar el número de errores de bit de datos en respuesta a un solo encendido o a un número predeterminado de encendidos. El número predeterminado de encendidos puede determinarse usando diversos criterios. Por ejemplo, el número de encendidos puede determinarse basándose en el número esperado de ciclos de alimentación que experimentará el sistema de memoria durante las pruebas de producción. En consecuencia, cualquier error en los datos que se precargan en el sistema de memoria durante la fabricación puede corregirse automáticamente antes de enviarse desde una instalación de producción. En el caso de una orden recibida, el sistema de memoria puede recibir la orden desde un sistema de procesamiento acoplado comunicativamente al sistema de memoria, desde un usuario tal como un operador de sistema o un técnico/ingeniero de pruebas, y/o desde un sistema de pruebas tal como un sistema de pruebas de desarrollo o de producción. La orden puede ser una orden para realizar la detección y corrección de errores, una orden de lectura, una orden de reinicio del sistema, una orden de reinicio de encendido (POR) o una orden de reactivación.

En algunas realizaciones, el controlador de memoria incluye un controlador de ECC (por ejemplo, el controlador de ECC 415 de la **figura 4**) que determina el número de errores de bit de datos en los datos almacenados en la matriz de memoria no volátil resistiva. Por ejemplo, el controlador de ECC lee los datos y los bits de ECC de la matriz de memoria resistiva y determina el número de errores de bit de datos basándose en un esquema de codificación de ECC.

En algunas realizaciones, el método de ejemplo 700 incluye realizar una escritura fuerte de los bits de datos y de anticipación de la matriz de memoria no volátil resistiva y verificar los datos escritos con una operación de lectura de verificación (720). El controlador de memoria realiza la operación de escritura fuerte escribiendo de nuevo los bits de datos y de anticipación en las celdas de memoria, de tal manera que la distribución de corriente de las celdas de la memoria es mayor que la de un ciclo de escritura normal. Para aumentar la distribución de corriente de una celda de memoria, el controlador de memoria programa el valor lógico para esa celda de memoria de tal manera que el valor de corriente leído de la celda de memoria se desplace más del valor umbral que para un ciclo de escritura normal. El aumento de la distribución de corriente dentro de las celdas de memoria requiere tiempo adicional de escritura y consumo de energía por parte del controlador de memoria, pero ayuda a proteger los bits de datos y de anticipación para evitar que se corrompan al aumentar el margen de celda de memoria. En consecuencia, el aumento de la distribución de corriente dentro de las celdas de memoria mejora la tolerancia a fallos al hacer que sea más difícil para las celdas de memoria cambiar los estados lógicos.

Para realizar la operación de escritura fuerte, el controlador de memoria determina un número de bits de ECC necesarios para corregir los errores de bit de datos y/o los errores de bit de anticipación leídos de la matriz de memoria no volátil resistiva. El número de bits de ECC almacenados en la matriz de memoria no volátil resistiva depende del esquema de codificación de ECC. Por ejemplo, el esquema de codificación de ECC puede asignar un número de bits de ECC por unidad de almacenamiento (por ejemplo, por página, por bloque, por sector, por palabra, etc.). El número de bits de ECC asignados a cada unidad de almacenamiento puede ser flexible y depende de los requisitos de diseño, las capacidades de sistema, los requisitos de tiempo y rendimiento, etc. El controlador de memoria o el controlador de ECC determinan el número de bits de ECC asignados a cada unidad de almacenamiento necesaria para corregir los errores de bit de datos en la unidad de almacenamiento. Por ejemplo, cinco bits de ECC se asignan a cada dirección de datos almacenados en la matriz de memoria no volátil resistiva.

Por lo tanto, en el ejemplo, el controlador de memoria o el controlador de ECC determinan cuántos de los cinco bits de ECC son necesarios para corregir los errores de bit de datos en la dirección de datos correspondiente. El controlador de memoria o el controlador de ECC determinan además el número de bits de ECC asignados a cada bit de anticipación (o unidad de almacenamiento o bits de anticipación) necesarios para corregir los errores de los bits de anticipación.

El controlador de memoria o el controlador de ECC determina si el número de los bits de ECC necesarios para corregir los errores de bit de datos para cada unidad de almacenamiento y/o los errores de bit de anticipación es menor que un número umbral de bits de ECC. El número umbral de bits de ECC se determina en general basándose en las capacidades de corrección de errores del esquema de codificación de ECC implementado por el controlador de memoria o el controlador de ECC. Sin embargo, pueden considerarse otros factores al determinar el número umbral de bits de ECC.

Cuando se determina que no se necesitan bits de ECC para corregir los errores de datos en una unidad de almacenamiento o para corregir un error de bit de anticipación (es decir, que no hay errores de bit de datos en la unidad de almacenamiento o no hay un error de bit de anticipación), entonces el controlador de memoria lee la unidad de almacenamiento de datos o el bit de anticipación y escribe de nuevo en la unidad de almacenamiento o el bit de anticipación en la matriz de memoria no volátil resistiva usando una operación de escritura fuerte (720). Leer bits de datos y de anticipación que no contienen errores y escribirlos de nuevo en la matriz de memoria no volátil resistiva usando la operación de escritura fuerte aumenta el margen dentro de las celdas de memoria que almacenan los bits de datos y de anticipación.

Cuando el controlador de memoria determina que se necesitan bits de ECC para corregir los errores de datos en una unidad de almacenamiento o para corregir un error de bit de anticipación, y el número de bits de ECC necesarios es menor que el número umbral de bits de ECC, el controlador de memoria o el controlador de ECC corrige los errores y escribe los bits corregidos en la matriz de memoria no volátil resistiva usando la operación de escritura fuerte. El controlador de memoria o el controlador de ECC corrigen los errores invirtiendo los valores de los bits de datos o el bit de anticipación. Por lo tanto, por ejemplo, si una celda de datos o una celda de anticipación incluye un valor lógico de "0" y el controlador de memoria o el controlador de ECC determina que el valor lógico es incorrecto usando los bits de ECC, entonces el controlador de memoria o el controlador de ECC invierte el valor lógico a "1".

El controlador de memoria escribe los bits corregidos en la matriz de memoria no volátil resistiva transmitiendo los bits corregidos a un circuito de escritura incluido en un amplificador de detección y controlador de escritura (por ejemplo, el amplificador de detección y controlador de escritura 414 de la **figura 4**). El controlador de memoria puede escribir los bits corregidos durante diversas etapas de la operación de sistema de memoria. Por ejemplo, en una realización, el controlador de memoria realiza la escritura en respuesta a un encendido posterior del sistema de memoria (por ejemplo, el siguiente encendido) o durante un modo inactivo posterior del sistema de memoria (por ejemplo, el siguiente modo inactivo). Cuando el controlador de memoria escribe los bits corregidos en respuesta a un encendido posterior, la escritura se realiza durante la inicialización de encendido o poco después de la misma. Esperar para escribir los bits corregidos hasta que el uso del sistema sea mínimo, tal como en respuesta a un encendido o durante un modo inactivo, minimiza el impacto en el rendimiento de sistema de memoria.

Cuando el controlador de memoria determina que los bits de ECC son necesarios para corregir los errores de datos en una unidad de almacenamiento o para corregir un error de bit de anticipación, y el número de bits de ECC necesarios es igual o mayor que el número umbral de bits de ECC, el controlador de memoria o el controlador de ECC reemplaza la unidad de almacenamiento o el bit de anticipación. Reemplazar una unidad de almacenamiento de datos o un bit de anticipación incluye reemplazar la unidad de almacenamiento de datos o el bit de anticipación con una unidad de almacenamiento de datos correspondiente o un bit de anticipación almacenado en una matriz de memoria no volátil basada en transistores. Las matrices de memoria no volátiles basadas en transistores incluyen, por ejemplo, memoria de solo lectura programable (PROM), memoria flash, memoria de solo lectura programable borrrable eléctricamente (EEPROM) o memoria eFUSE.

La unidad de almacenamiento de datos o el bit de anticipación almacenado en la matriz de memoria no volátil basada en transistores puede ser una copia (es decir, una unidad de almacenamiento de copia de seguridad o un bit de anticipación de copia de seguridad) de la unidad de almacenamiento de datos o el bit de anticipación en la matriz de memoria no volátil resistiva que se está reemplazando. El controlador de memoria escribe los bits de datos y de anticipación almacenados en la matriz de memoria resistiva en la matriz de memoria no volátil basada en transistores para crear las unidades de almacenamiento de copia de seguridad de los bits de datos y de anticipación. La escritura puede producirse durante diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede escribir los bits de datos y de anticipación en la matriz de memoria no volátil basada en transistores antes de un ciclo térmico de alta temperatura (por ejemplo, un proceso de soldadura por onda de montaje en placa), después se escriben los bits de datos y de anticipación en la matriz de memoria no volátil resistiva, durante un estado inactivo del sistema de memoria, en respuesta a una orden recibida, o en respuesta a un encendido del sistema de memoria.

El controlador de memoria lee la unidad de almacenamiento de copia de seguridad o el bit de anticipación de copia de seguridad almacenado en la matriz de memoria no volátil basada en transistores como parte del proceso de sustitución. En algunas realizaciones, el controlador de memoria controla tanto la matriz de memoria no volátil resistiva como la matriz de memoria no volátil basada en transistores. En tales realizaciones, el controlador de memoria lee directamente los datos almacenados en la matriz de memoria no volátil basada en transistores. En otras realizaciones, la matriz de memoria no volátil basada en transistores se controla por un controlador de memoria separado. En tales realizaciones, el controlador de memoria asociado con la matriz de memoria no volátil resistiva transmite una solicitud al controlador de memoria asociado con la matriz de memoria no volátil basada en transistores. El controlador de memoria asociado con la matriz de memoria no volátil basada en transistores recibe la solicitud, realiza una operación de lectura para recuperar la unidad de almacenamiento de copia de seguridad almacenada de datos o el bit de anticipación de copia de seguridad, y transmite la copia de seguridad recuperada al controlador de memoria asociado con la matriz de memoria no volátil resistiva.

El controlador de memoria escribe la unidad de almacenamiento de copia de seguridad o el bit de anticipación de copia de seguridad en la matriz de memoria no volátil resistiva usando la operación de escritura fuerte. El controlador de memoria reemplaza la unidad de almacenamiento de datos o el bit de anticipación en diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede reemplazar la unidad de almacenamiento de datos o el bit de anticipación inmediatamente después de determinar el número de bits de ECC necesarios para corregir los errores igual o superior al número de bits de ECC de umbral. Como otro ejemplo, el controlador de memoria puede reemplazar la unidad de almacenamiento de datos o el bit de anticipación en respuesta a un encendido posterior del sistema de memoria (por ejemplo, el siguiente encendido), o durante un modo inactivo posterior del sistema de memoria (por ejemplo, el siguiente modo inactivo).

Una vez que la operación de escritura fuerte se ha realizado para los bits de datos o para el bit de anticipación, el controlador de memoria verifica que la operación de escritura fuerte se ha realizado con éxito realizando una operación de lectura de margen descrita anteriormente haciendo referencia a la **figura 4**. La operación de lectura de margen compara la distribución de corriente de los datos escritos o el bit de anticipación con el valor de corriente de umbral superior y/o inferior para verificar que la distribución de corriente del bit de datos o de anticipación escrito ha aumentado. Una vez que el controlador de memoria verifica que se ha realizado con éxito la operación de escritura fuerte, el controlador de memoria determina si todas las celdas de datos y las celdas de anticipación se han actualizado (730). Si el controlador de memoria determina que todas las celdas de datos y las celdas de anticipación se han actualizado (730-Sí), entonces la operación de actualización fuerte se completa (740). Si el controlador de memoria determina que algunas celdas de datos o celdas de anticipación aún necesitan actualizarse (730-No), entonces el controlador de memoria se mueve a la siguiente celda de datos/de anticipación y continúa la operación de actualización fuerte (750).

La **figura 8** representa un diagrama de flujo de un método de ejemplo 800, consecuente con algunas realizaciones y aspectos de la presente divulgación. El método 800 puede implementarse, por ejemplo, para realizar una actualización débil de los datos almacenados en la matriz de memoria no volátil resistiva. La actualización débil incluye actualizar solo la unidad de almacenamiento de datos que tiene errores de bit de datos. En algunas realizaciones, el método 800 puede implementarse mediante un sistema de memoria tal como, por ejemplo, el sistema de memoria 400 ilustrado en la **figura 4**.

En algunas realizaciones, el método 800 incluye leer los datos con corrección de ECC y detectar los errores de bits de datos entre los datos almacenados en una matriz de memoria no volátil resistiva (810). Un controlador de memoria (por ejemplo, el controlador de memoria 410 de la **figura 4**) incluido en el sistema de memoria lee los datos almacenados en las celdas de memoria (por ejemplo, las celdas de datos 310 de la **figura 3**) de una matriz de memoria resistiva (por ejemplo, la matriz de memoria resistiva 200 de las **figuras 2-4**). El controlador de memoria lee datos en diversas unidades de almacenamiento, tal como páginas, bloques, sectores, palabras o cualquier otra unidad de almacenamiento conocida en la técnica. Cada unidad de almacenamiento puede comprender cualquier número de bits de datos. El controlador de memoria lee además los bits de ECC almacenados en las celdas de memoria (por ejemplo, las celdas de ECC 320 de la **figura 3**) de la matriz de memoria resistiva y usa los bits de ECC almacenados para detectar los errores de bit de datos en los datos almacenados. El controlador de memoria detecta errores de bits de datos en las unidades de almacenamiento de datos determinando qué bits de datos han cambiado de estados lógicos (por ejemplo, de un "1" lógico a un "0" lógico y viceversa) debido a las altas temperaturas o a la EMI. Los bits de ECC almacenados pueden incluir códigos de Hamming, códigos Bose, Chaudhuri y Hocquenghem (BCH), bits de paridad, códigos Reed-Solomon, códigos Turbo, códigos de verificación de paridad de baja densidad (LDPC), códigos convolucionales o cualquier otro código de corrección de errores conocido en la técnica.

El controlador de memoria detecta los errores de bit de datos durante las diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede detectar los errores de bit de datos durante un ciclo de lectura, en respuesta al encendido del sistema de memoria, durante un modo inactivo del sistema de memoria, o en respuesta a una orden recibida. En el caso de un modo inactivo, el controlador de memoria puede detectar los errores de bit de datos durante un solo modo inactivo o durante un número predeterminado de modos inactivos. En consecuencia, el controlador de memoria realiza la detección de errores cuando el impacto en el rendimiento del

sistema es mínimo. En el caso de un encendido, el controlador de memoria puede detectar los errores de bit de datos en respuesta a un solo encendido o a un número predeterminado de encendidos. El número predeterminado de encendidos puede determinarse usando diversos criterios. Por ejemplo, el número de encendidos puede determinarse basándose en el número esperado de ciclos de alimentación que experimentará el sistema de memoria durante las pruebas de producción. En consecuencia, cualquier defecto en los datos que se precargan en el sistema de memoria durante la fabricación pueden corregirse automáticamente antes de enviarse desde una instalación de producción. En el caso de una orden recibida, el sistema de memoria puede recibir la orden desde un sistema de procesamiento acoplado comunicativamente al sistema de memoria, desde un usuario tal como un operador de sistema o un técnico/ingeniero de pruebas, y/o desde un sistema de pruebas tal como un sistema de pruebas de desarrollo o de producción. La orden puede ser una orden para realizar la detección y corrección de errores, una orden de lectura, una orden de reinicio de sistema, una orden de reinicio de encendido (POR) o una orden de reactivación.

En algunas realizaciones, el controlador de memoria incluye un controlador de ECC (por ejemplo, el controlador de ECC 415 de la **figura 4**) que detecta los errores de bit de datos en los datos almacenados en la matriz de memoria no volátil resistiva. Por ejemplo, el controlador de ECC lee los bits de datos y de ECC de la matriz de memoria resistiva y detecta los errores de bit de datos basándose en un esquema de codificación de ECC.

En algunas realizaciones, el método 800 incluye determinar un número de bits de ECC necesarios para corregir los errores de bit de datos (820). El número de bits de ECC almacenados en la matriz de memoria no volátil resistiva depende del esquema de codificación de ECC. Por ejemplo, el esquema de codificación de ECC puede asignar un número de bits de ECC por unidad de almacenamiento (por ejemplo, por página, por bloque, por sector, por palabra, etc.). El número de bits de ECC asignados a cada unidad de almacenamiento puede ser flexible y depende de los requisitos de diseño, las capacidades del sistema, los requisitos de tiempo y rendimiento, etc. El controlador de memoria o el controlador de ECC determinan el número de bits de ECC asignados a cada unidad de almacenamiento necesario para corregir los errores de bit de datos en la unidad de almacenamiento. Por ejemplo, cinco bits de ECC se asignan a cada dirección de datos almacenados en la matriz de memoria no volátil resistiva. Por lo tanto, en el ejemplo, el controlador de memoria o el controlador de ECC determinan cuántos de los cinco bits de ECC son necesarios para corregir los errores de bit de datos en la dirección de datos correspondiente.

El controlador de memoria o el controlador de ECC determinan si el número de los bits de ECC necesarios para corregir los errores de bit de datos para cada unidad de almacenamiento es menor que un número umbral de bits de ECC (830). Para la operación de actualización débil, el número umbral de bits de ECC se selecciona para que sea menor que las capacidades de corrección de errores de bit de ECC máximas del esquema de codificación de ECC implementado por el controlador de memoria o el controlador de ECC. Ya que se ha determinado que los bits de anticipación almacenados no contienen errores significativos (530-sí), los recursos de sistema de memoria usados para realizar la operación de actualización débil pueden minimizarse al escribir solo datos que tengan errores o daños significativos en lugar de escribir toda la memoria no volátil resistiva de datos.

Cuando el controlador de memoria determina que el número de bits de ECC necesarios para corregir los errores de datos en una unidad de almacenamiento es menor que el umbral (830-Sí), el controlador de memoria y/o el controlador de ECC no realizan una actualización débil de los datos y continúa a la siguiente celda de datos (840).

Cuando el controlador de memoria determina que el número de bits de ECC necesario para corregir los errores de datos en una unidad de almacenamiento es mayor o igual al umbral (830-No), y el número de bits de ECC necesarios es menor que los bits de ECC máximos disponibles para la corrección de errores, el controlador de memoria o el controlador de ECC realiza una escritura de actualización débil corrigiendo los errores y escribiendo los bits corregidos en la matriz de memoria no volátil resistiva (850). El controlador de memoria o el controlador de ECC corrigen los errores invirtiendo los valores de los errores de bit de datos. Por lo tanto, por ejemplo, si una celda de datos incluye un valor lógico de "0" y el controlador de memoria o el controlador de ECC determina que el valor lógico es incorrecto usando los bits de ECC, entonces el controlador de memoria o el controlador de ECC invierte el valor lógico a "1".

El controlador de memoria escribe los bits de datos corregidos en la matriz de memoria no volátil resistiva usando una operación de escritura normal descrita anteriormente haciendo referencia a la **figura 4**. El controlador de memoria escribe los bits corregidos en la matriz de memoria no volátil resistiva transmitiendo los bits corregidos a un circuito de escritura incluido en un amplificador de detección y controlador de escritura (por ejemplo, el amplificador de detección y controlador de escritura 414 de la **figura 4**). El controlador de memoria puede escribir los bits corregidos durante diversas etapas de la operación de sistema de memoria. Por ejemplo, en una realización, el controlador de memoria realiza la escritura en respuesta a un encendido posterior del sistema de memoria (por ejemplo, el siguiente encendido) o durante un modo inactivo posterior del sistema de memoria (por ejemplo, el siguiente modo inactivo). Cuando el controlador de memoria escribe los bits corregidos en respuesta a un encendido posterior, la escritura se realiza durante la inicialización de encendido o poco después. Esperar para escribir los bits corregidos hasta que el uso del sistema sea mínimo, tal como en respuesta a un encendido o durante un modo inactivo, minimiza el impacto en el rendimiento de sistema de memoria. En otra realización, el controlador de

5 memoria realiza la escritura inmediatamente después de corregir los errores de bit de datos. Los tiempos de escritura de las celdas de memoria no volátiles resistivas suelen ser mucho más cortos que los tiempos de escritura de las celdas de memoria no volátil basadas en transistores (por ejemplo, las celdas de memoria flash). Ya que solo las unidades de almacenamiento de datos que tienen errores de bit de datos se escriben durante una actualización débil, la actualización débil puede realizarse durante la operación normal del sistema de memoria sin afectar negativamente al sistema de memoria.

10 Cuando el controlador de memoria determina que el número de bits de ECC necesarios para corregir errores de datos en una unidad de almacenamiento es mayor o igual que el umbral (830-No), y el número de bits de ECC necesarios es igual o mayor que los bits de ECC máximos disponibles para la corrección de errores, el controlador de memoria o el controlador de ECC realizan una escritura de actualización débil reemplazando la unidad de almacenamiento (850). Reemplazar una unidad de almacenamiento de datos incluye reemplazar la unidad de almacenamiento de datos con una unidad de almacenamiento correspondiente de datos almacenados en una matriz de memoria no volátil basada en transistores. Las matrices de memoria no volátil basadas en transistores incluyen, por ejemplo, memoria de solo lectura programable (PROM), memoria flash, memoria de solo lectura programable borrrable eléctricamente (EEPROM) o memoria eFUSE.

20 La unidad de almacenamiento de datos almacenados en la matriz de memoria no volátil basada en transistores puede ser una copia (es decir, una unidad de almacenamiento de copia de seguridad) de la unidad de almacenamiento de datos en la matriz de memoria no volátil resistiva que se está reemplazando. El controlador de memoria puede escribir los datos almacenados en la matriz de memoria resistiva en la matriz de memoria no volátil basada en transistores para crear las unidades de datos de almacenamiento de copia de seguridad. La escritura puede producirse durante diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede escribir los datos en la matriz de memoria no volátil basada en transistores antes de un ciclo térmico de alta temperatura (por ejemplo, un proceso de soldadura por onda de montaje en placa), después de que los datos se escriban en la matriz de memoria no volátil resistiva, durante un estado de inactividad del sistema de memoria, en respuesta a una orden recibida, o en respuesta a un encendido del sistema de memoria.

30 El controlador de memoria lee la unidad de almacenamiento de copia de seguridad almacenada en la matriz de memoria no volátil basada en transistores como parte del proceso de sustitución. En algunas realizaciones, el controlador de memoria controla tanto la matriz de memoria no volátil resistiva como la matriz de memoria no volátil basada en transistores. En tales realizaciones, el controlador de memoria lee directamente los datos almacenados en la matriz de memoria no volátil basada en transistores. En otras realizaciones, la matriz de memoria no volátil basada en transistores se controla por un controlador de memoria separado. En tales realizaciones, el controlador de memoria asociado con la matriz de memoria no volátil resistiva transmite una solicitud al controlador de memoria asociado con la matriz de memoria no volátil basada en transistores. El controlador de memoria asociado con la matriz de memoria no volátil basada en transistores recibe la solicitud, realiza una operación de lectura para recuperar la unidad de almacenamiento de copia de seguridad almacenada de los bits de datos o de anticipación de copia de seguridad, y transmite la copia de seguridad recuperada al controlador de memoria asociado con la matriz de memoria no volátil resistiva.

40 El controlador de memoria escribe el almacenamiento de copia de seguridad en la matriz de memoria no volátil resistiva usando la operación de escritura normal descrita anteriormente haciendo referencia a la **figura 4**. El controlador de memoria reemplaza la unidad de almacenamiento de datos en diversas etapas de la operación de sistema de memoria. Por ejemplo, el controlador de memoria puede reemplazar la unidad de almacenamiento de datos inmediatamente después de determinar el número de bits de ECC que se necesitan para corregir los errores iguales o superiores al número de bits de ECC de umbral, y el número de bits de ECC necesarios es igual o mayor que los bits de ECC máximos disponibles para la corrección de errores. Como otro ejemplo, el controlador de memoria puede reemplazar la unidad de almacenamiento de datos en respuesta a un encendido posterior del sistema de memoria (por ejemplo, el siguiente encendido), o durante un modo inactivo posterior del sistema de memoria (por ejemplo, el siguiente modo inactivo).

50 Una vez que la operación de actualización débil se ha realizado para los bits de datos o el bit de anticipación, el controlador de memoria verifica que se ha realizado con éxito la operación de actualización débil realizando una operación de lectura de margen descrita anteriormente haciendo referencia a la **figura 4**. La lectura de margen para la operación de actualización débil usa los valores de actuales de umbral superior e inferior que están más cerca de un valor de corriente de umbral normal que para la lectura de margen de la operación de escritura fuerte. Una vez que se completa la operación de actualización débil para la celda de datos (850), el controlador de memoria y/o el controlador de ECC se mueven a la siguiente celda de datos y la operación de actualización débil continúa.

55 La **figura 9** ilustra una comparación entre una operación de lectura normal y una operación de lectura de margen consecuente con las realizaciones de la presente divulgación. Como se muestra en la **figura 9**, la corriente de umbral normal 910 divide la distribución de corriente de una celda de memoria en dos regiones: una región superior y una región inferior. La región superior y las regiones inferiores corresponden a valores lógicos diferentes (por ejemplo, "0" lógico y "1" lógico). En una operación de escritura normal, el controlador de memoria (por ejemplo, el

controlador de memoria 410, **figura 4**) programa la celda de memoria de tal manera que la distribución de corriente de la celda caiga a ambos lados de la corriente de umbral normal 910. En una operación de lectura normal, el controlador de memoria mide la corriente a través de la celda de memoria y la compara con la corriente de umbral normal 910 para determinar el valor lógico de la celda de memoria.

5 En una operación de actualización débil, el controlador de memoria usa una corriente de umbral débil inferior 920a y una corriente de umbral débil superior 920b como límites para la región inferior y la región superior, respectivamente. Como se muestra en la **figura 9**, la corriente de umbral débil inferior 920a y la corriente de umbral débil superior 920b se distribuyen a ambos lados de la corriente de umbral normal 910, de tal manera que hay un margen entre cada corriente de umbral débil 920a y 920b y una corriente de umbral normal 910. En una escritura de actualización débil, el controlador de memoria programa la celda de memoria de tal manera que la distribución de corriente de la celda sea menor que la corriente de umbral débil inferior 920a o mayor que la corriente de umbral débil superior 920b.

15 En una operación de lectura de verificación débil (por ejemplo, una lectura de margen), el controlador de memoria mide la corriente a través de la celda de memoria y la compara con la corriente de umbral débil inferior 920a y la corriente de umbral débil superior 920b para determinar el valor lógico de la celda de memoria.

20 En una operación de actualización fuerte, el controlador de memoria usa una corriente de umbral fuerte inferior 930a y una corriente de umbral fuerte superior 930b como límites para la región inferior y la región superior, respectivamente. Como se muestra en la **figura 9**, la corriente de umbral fuerte inferior 930a y la corriente de umbral fuerte superior 930b se distribuyen a ambos lados de la corriente de umbral normal 910, de tal manera que existe un margen entre las dos corrientes de umbral fuertes 930a y 930b y la corriente de umbral normal 910. Además, la corriente de umbral fuerte inferior 930a y la corriente de umbral fuerte superior 930b se distribuyen más lejos de la corriente de umbral normal 910 que la corriente de umbral débil inferior 920a y la corriente de umbral débil superior 920b, respectivamente. En una escritura de actualización fuerte, el controlador de memoria programa la celda de memoria de tal manera que la distribución de corriente de la celda es menor que la corriente de umbral fuerte inferior 930a o mayor que la corriente de umbral fuerte superior 930b. En consecuencia, la operación de escritura de actualización fuerte proporciona más margen de celda de memoria que la operación de escritura de actualización débil.

30 En una operación de lectura de verificación fuerte (por ejemplo, una lectura de margen), el controlador de memoria mide la corriente a través de la celda de memoria y la compara con la corriente de umbral fuerte inferior 930a y la corriente de umbral fuerte superior 930b para determinar el valor lógico de la celda de memoria.

35 Las corrientes de umbral fuertes 930a y 930b pueden usarse para detectar cambios en la distribución de corriente de una celda de memoria. Por ejemplo, una operación de lectura de margen que usa una corriente de umbral fuerte inferior 930a y una corriente de umbral fuerte superior 930b puede detectar que la distribución de corriente de una celda de memoria se ha cambiado más cerca de la corriente de umbral normal 910 y, por lo tanto, que la calidad de los datos almacenados en la celda de memoria se está degradando. La detección temprana de la degradación de la calidad de los datos es útil, por ejemplo, cuando se usan los errores de bit de anticipación como indicadores de grandes cantidades de datos debido a que la degradación en la calidad de los bit de anticipación indica que la calidad de los datos correspondientes también se está degradando y, por lo tanto, deberían actualizarse.

40 La **figura 9** ilustra además las curvas de distribución de ejemplo 940a y 940b que muestran el número de bits almacenados en una celda de memoria en comparación con la distribución de corriente de la celda. Las curvas de distribución 940a y 940b son solo a modo de ejemplo y un experto en la materia reconocerá que son posibles otras curvas de distribución.

45 En la descripción anterior, se han descrito haciendo referencia a los dibujos adjuntos diversas realizaciones y aspectos a modo de ejemplo. Sin embargo, será evidente que pueden realizarse diversas modificaciones y cambios en los mismos, y que pueden implementarse realizaciones y aspectos adicionales, sin alejarse del alcance más amplio de la invención como se expone en las siguientes reivindicaciones. Por consiguiente, la memoria descriptiva y los dibujos deben considerarse en sentido ilustrativo en lugar de restrictivo.

REIVINDICACIONES

1. Un sistema de memoria, que comprende:

una matriz de memoria no volátil resistiva (200) configurada para almacenar datos, bits de anticipación y bits de código de corrección de errores (ECC) correspondientes a los datos almacenados y a los bits de anticipación almacenados, en la que los bits de anticipación indican la calidad de los bits de datos almacenados en la matriz de memoria no volátil, y un número total de bits de anticipación es menor que un número total de los bits de datos, **caracterizado por que** un controlador de memoria (410) está configurado para:

(510, 610) realizar una operación de lectura en los bits de anticipación y en los bits de ECC para detectar errores de bit de datos en unidades de los datos almacenados y errores de bit de anticipación entre los bits de anticipación almacenados;

(520, 620) detectar un número de los errores de bit de anticipación;

(530, 650) comparar el número de los errores de bit de anticipación con un número umbral para los errores de bit de anticipación;

(550, 670) realizar una actualización fuerte de todos los bits de datos y los bits de anticipación almacenados en la matriz de memoria no volátil resistiva cuando el número de los errores de bit de anticipación es igual o superior al número umbral para los errores de bit de anticipación; y

(540, 660) realizar una actualización débil actualizando solo las unidades de los datos almacenados que tienen los errores de bit de datos y los bits de anticipación almacenados que tienen los errores de bit de anticipación cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación.

2. El sistema de memoria de la reivindicación 1, en el que la matriz de memoria no volátil resistiva comprende al menos una de entre una memoria de acceso aleatorio de cambio de fase (PCRAM), una memoria de acceso aleatorio resistiva (RRAM), o una memoria de acceso aleatorio de puente conductor (CGRAM).

3. El sistema de memoria de la reivindicación 1, en el que: el controlador de memoria está configurado además para:

(820) determinar un número de los bits de ECC necesarios para corregir los errores de bit de datos; y

(830) comparar el número de los bits de ECC con un número umbral para los bits de ECC;

en el que el controlador de memoria realiza la actualización débil (850) cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación mientras que el número de los bits de ECC necesarios para corregir los errores de bit de datos en las unidades de datos almacenados es igual o mayor que el número umbral para los bits de ECC, y no realiza la actualización débil (840) cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación mientras que el número de los bits de ECC necesarios para corregir los errores de bit de datos es mayor que cero y menor que el número umbral para los bits de ECC.

4. El sistema de memoria de la reivindicación 1, en el que el controlador de memoria está configurado además para:

realizar una operación de lectura de margen usando un valor de corriente umbral fuerte superior (930b) y un valor de corriente umbral fuerte inferior (930a) para verificar que se ha realizado con éxito la actualización fuerte; y

realizar la operación de lectura de margen usando un valor de corriente umbral débil superior (920b) y un valor de corriente umbral débil inferior (920a) para verificar que se ha realizado con éxito la actualización débil;

en el que la diferencia entre el valor de corriente umbral fuerte superior y el valor de corriente umbral fuerte inferior es mayor que la diferencia entre el valor de corriente umbral débil superior y el valor de corriente umbral débil inferior.

5. El sistema de memoria de la reivindicación 1, en el que la operación de lectura en los bits de anticipación es una operación de lectura de margen con una pluralidad de referencias de umbral a comparar para determinar los valores lógicos de los bits de anticipación.

6. El sistema de memoria de la reivindicación 1, en el que los bits de datos y de anticipación se almacenan respectivamente en diferentes matrices de memoria no volátil resistivas formadas en un chip de semiconductor, almacenados en diferentes matrices de memoria no volátil resistivas formadas en chips de semiconductor separados incluidos en un paquete de circuito integrado (CI), o almacenados en diferentes matrices de memoria no volátil resistivas formadas en chips de semiconductor separados incluidos en paquetes de CI separados.

7. Un método para corregir errores en un sistema de memoria que incluye una matriz de memoria no volátil resistiva (200), que comprende:

realizar una operación de lectura en los bits de anticipación y en los bits de código de corrección de errores (ECC) para detectar errores de bit de datos en unidades de datos almacenados en la matriz de memoria no volátil resistiva y errores de bit de anticipación entre los bits de anticipación almacenados en la matriz de memoria no volátil resistiva, **caracterizado por que** los bits de ECC se almacenan en la matriz de memoria no volátil resistiva y corresponden a las unidades de los datos almacenados y a los bits de anticipación almacenados, y los bits de anticipación indican la calidad de los bits de datos almacenados en la matriz de memoria no volátil, y un número total de los bits de anticipación es menor que un número total de los bits de datos; y que comprende además:

- (520) detectar un número de los errores de bit de anticipación;
- (530) comparar el número de los errores de bit de anticipación con un número umbral para los errores de bit de anticipación;
- (550) realizar una actualización fuerte de todos los bits de datos y de anticipación almacenados en la matriz de memoria no volátil resistiva cuando el número de los errores de bit de anticipación es igual o superior al número umbral para los errores de bit de anticipación; y
- (540) realizar una actualización débil actualizando solo las unidades de los datos almacenados que tienen los errores de bit de datos y los bits de anticipación que tienen los errores de bit de anticipación cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación.

8. El método de la reivindicación 7, en el que la matriz de memoria no volátil resistiva comprende al menos una de entre una memoria de acceso aleatorio de cambio de fase (PCRAM), una memoria de acceso aleatorio resistiva (RRAM) y una memoria de acceso aleatorio de puente conductor (CBRAM).

9. El método de la reivindicación 7, en el que la detección del número de los errores de bit de anticipación es en respuesta al encendido del sistema de memoria, durante un modo inactivo del sistema de memoria, o en respuesta a una orden recibida.

10. El método de la reivindicación 7, en el que la detección del número de los errores de bit de anticipación es en respuesta a un número predeterminado de encendidos, durante un número predeterminado de modos inactivos, o en respuesta a un número predeterminado de órdenes recibidas.

11. El método de la reivindicación 7, que comprende además:

- (820) determinar un número de ECC y
- (830) comparar el número de los bits de ECC con un número umbral para los bits de ECC;
- realizar la actualización débil (850) cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación mientras el número de los bits de ECC necesarios para corregir los errores de bit de datos en las unidades de datos almacenados es igual o mayor que el número umbral para los bits de ECC, y no realizar la actualización débil (840) cuando el número de los errores de bit de anticipación es menor que el número umbral para los errores de bit de anticipación, mientras el número de los bits de ECC necesarios para corregir los errores de bit de datos es mayor que cero y menor que el número umbral para los bits de ECC.

12. El método de la reivindicación 11, en el que realizar la actualización fuerte incluye:
(720) reemplazar las unidades de los datos almacenados y los bits de anticipación con las unidades de datos correspondientes y los bits de anticipación almacenados en una matriz de memoria no volátil basada en transistores.

13. El método de la reivindicación 7 incluye además:

- realizar una operación de lectura de margen usando un valor de corriente umbral fuerte superior (930b) y un valor de corriente umbral fuerte inferior (930a) para verificar que se ha realizado con éxito la operación de escritura fuerte; y
- realizar la operación de lectura de margen usando un valor de corriente umbral débil superior (920b) y un valor de corriente umbral débil inferior (920a) para verificar que se ha realizado con éxito la actualización débil, en el que la diferencia entre el valor de corriente umbral fuerte superior y el valor de corriente umbral fuerte inferior es mayor que la diferencia entre el valor de corriente umbral débil superior y el valor de corriente umbral débil inferior.

14. El método de la reivindicación 7, en el que la operación de lectura en los bits de anticipación es una operación de lectura de margen con una pluralidad de referencias de umbral a comparar para determinar los valores lógicos de los

bits de anticipación.

15. El método de la reivindicación 7, en el que realizar la actualización fuerte o la actualización débil incluye realizar la actualización fuerte o la actualización débil en respuesta a un encendido del sistema de memoria o durante un modo inactivo del sistema de memoria.

5

100

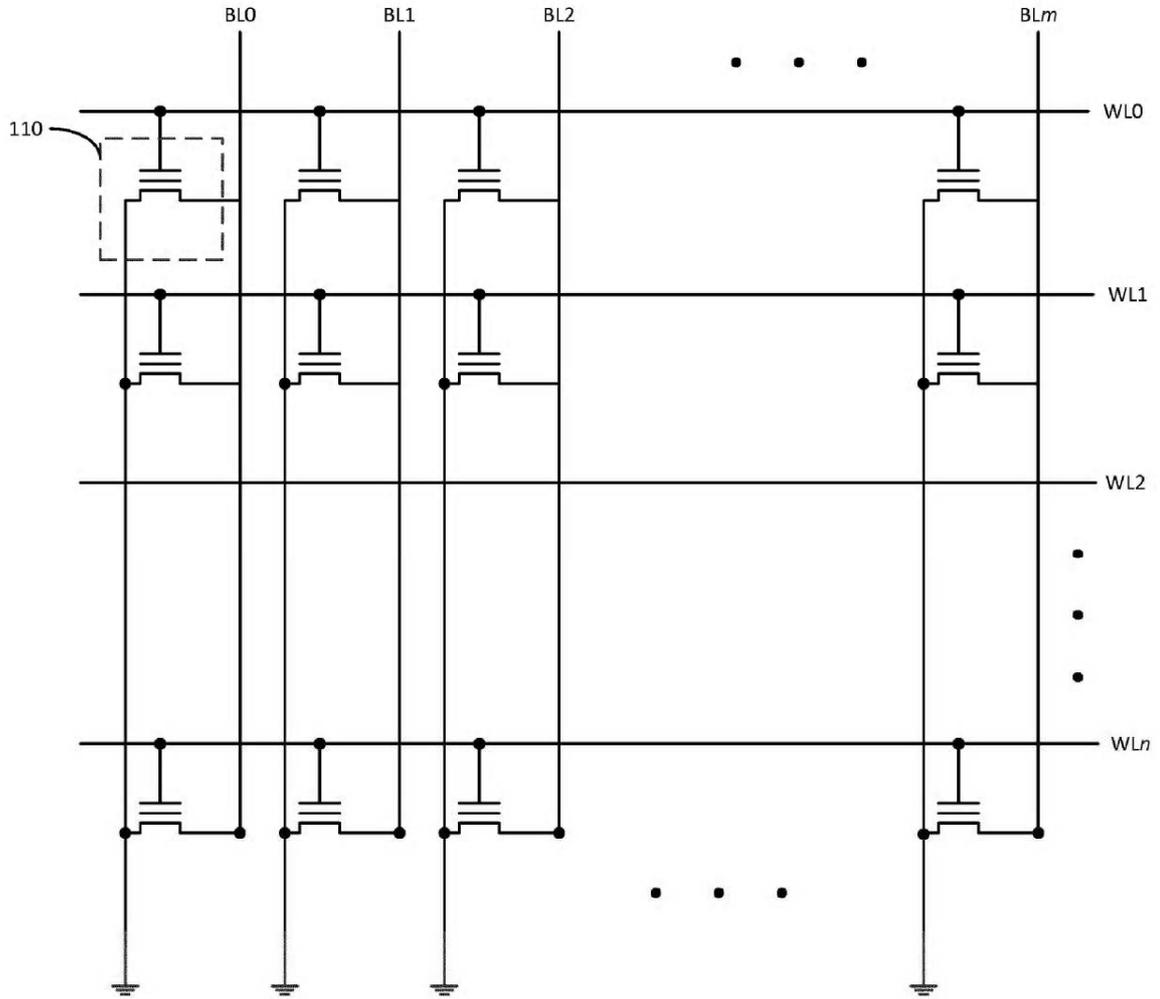


FIG. 1
(Técnica anterior)

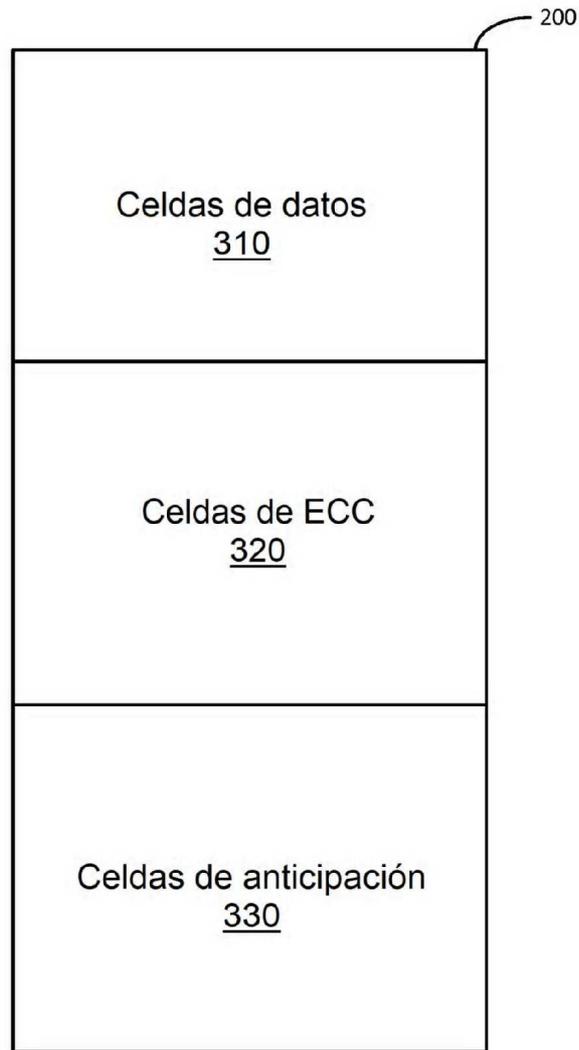


FIG. 3

400

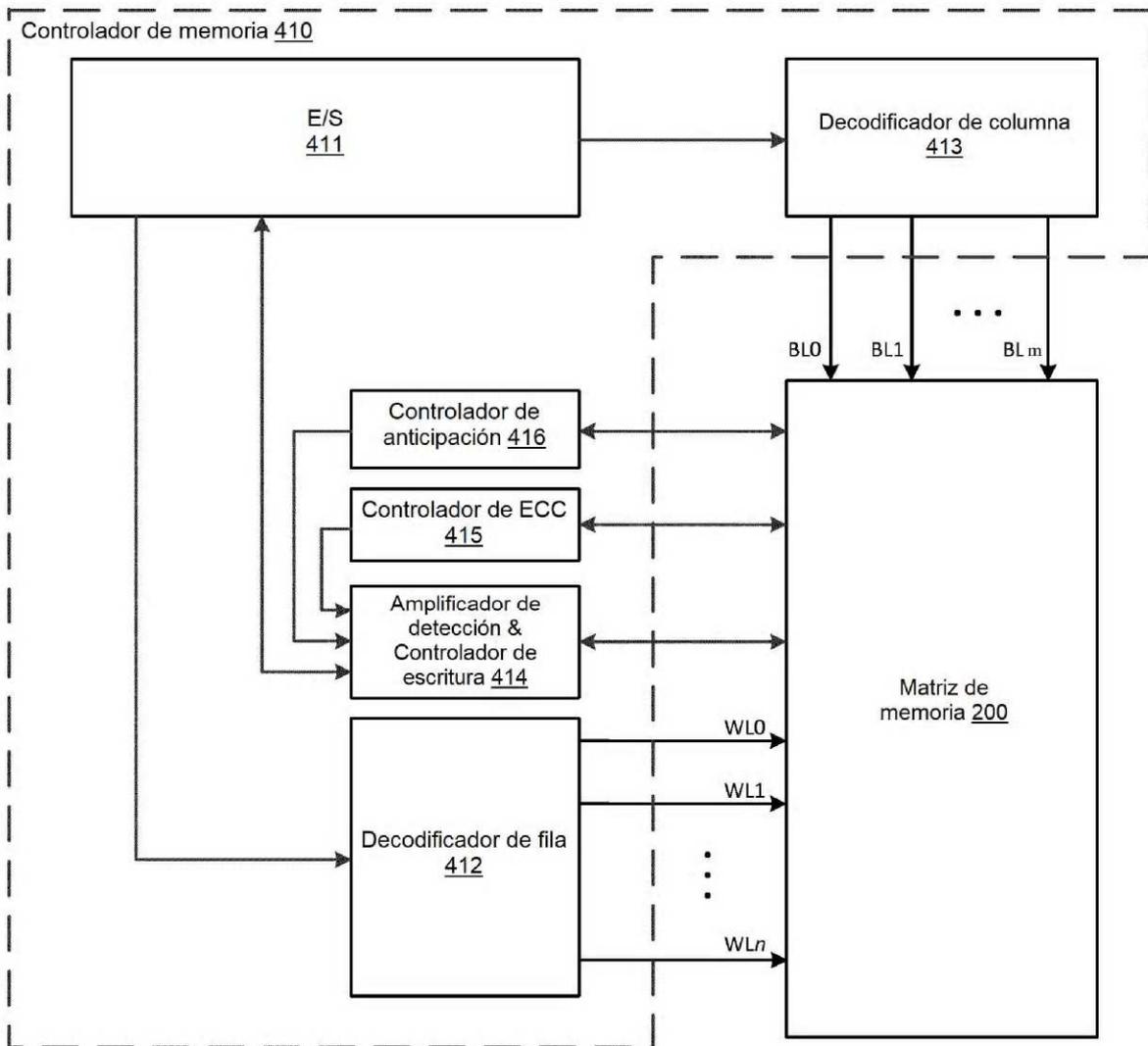


FIG. 4

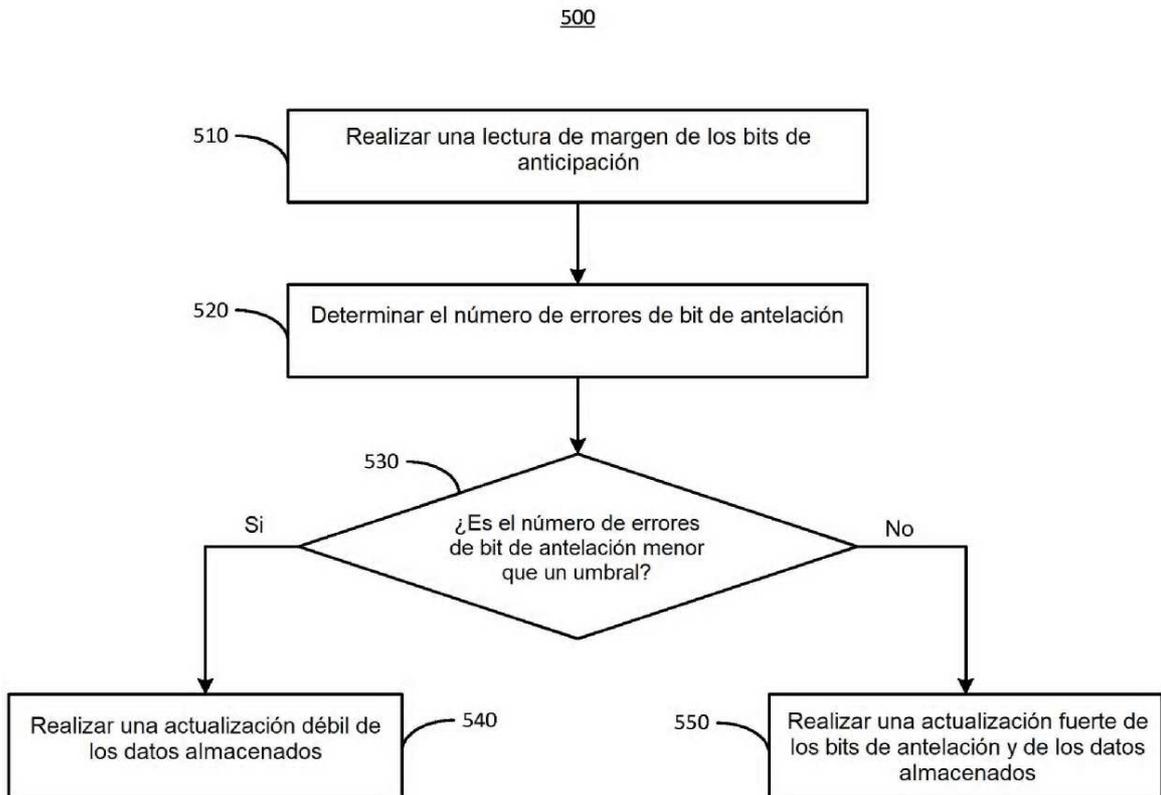


FIG. 5

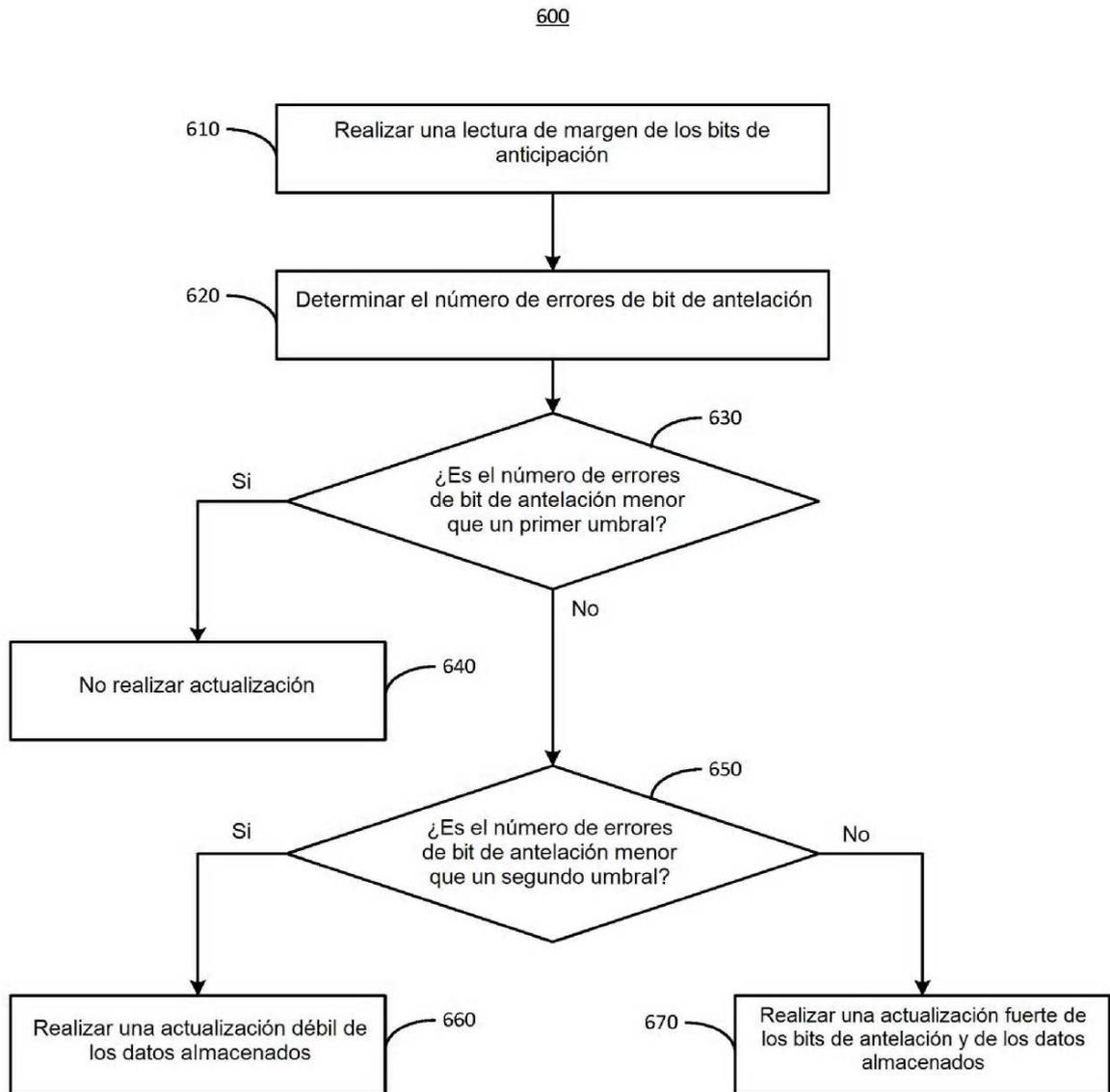


FIG. 6

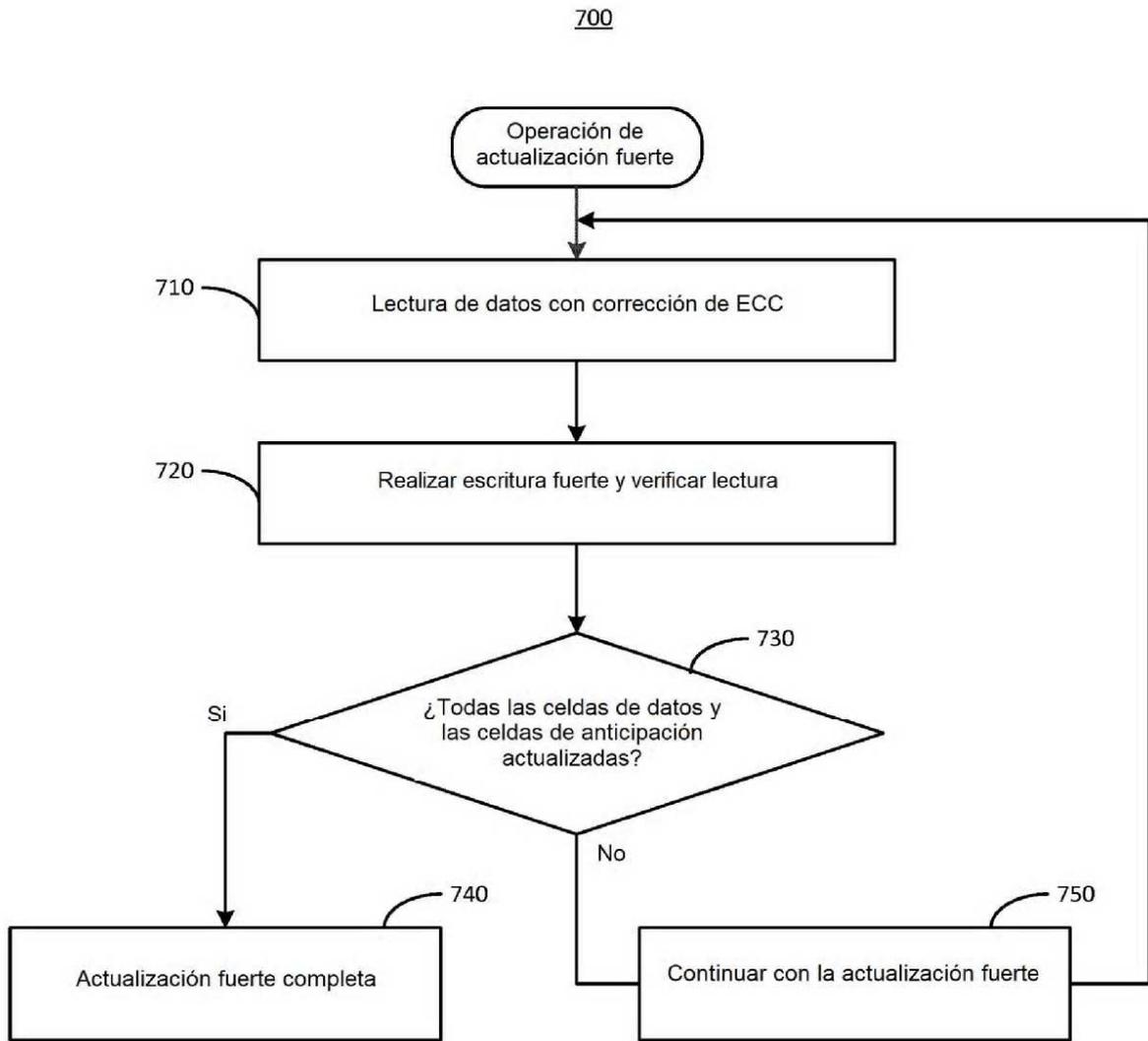


FIG. 7

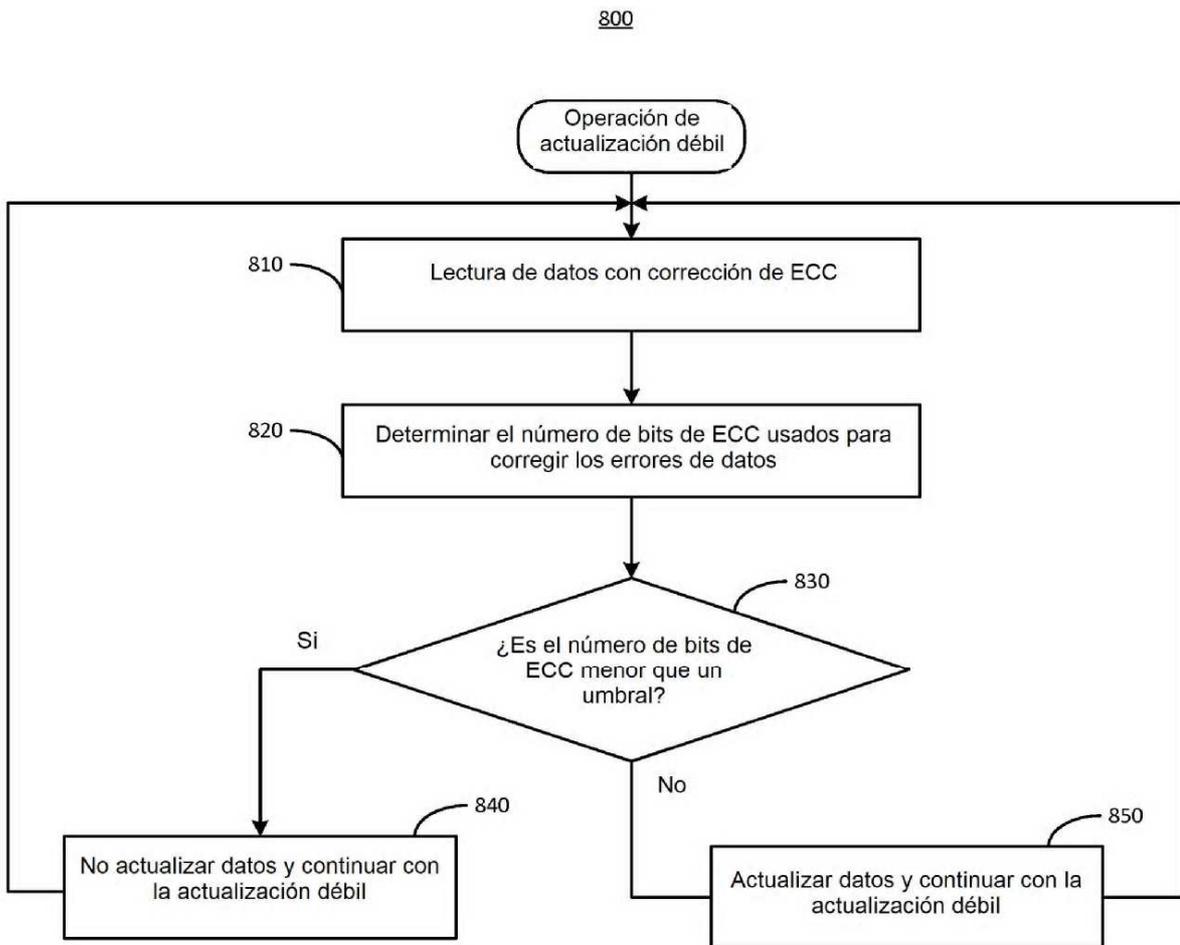


FIG. 8

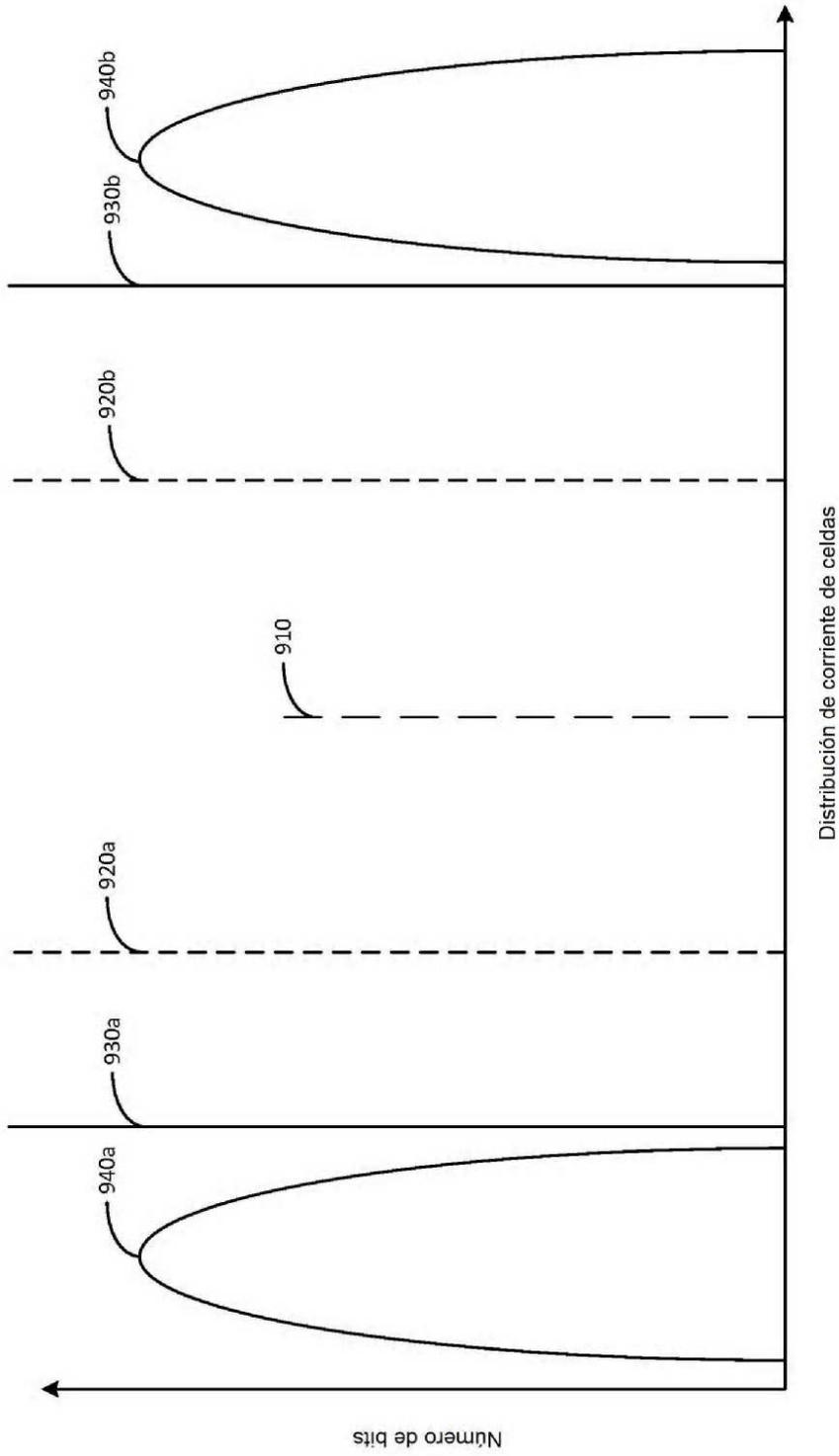


FIG. 9