

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 498 096**

51 Int. Cl.:

G11C 7/10 (2006.01)

G11C 11/4193 (2006.01)

G11C 16/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.03.2007** **E 10171414 (5)**

97 Fecha y número de publicación de la concesión europea: **16.07.2014** **EP 2242058**

54 Título: **Esquema de control de sistema de memoria Flash**

30 Prioridad:

31.03.2006 US 788083 P

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

24.09.2014

73 Titular/es:

**MOSAID TECHNOLOGIES INCORPORATED
(100.0%)
11 HINES ROAD, SUITE 203
OTTAWA, ON K2K 2X1, CA**

72 Inventor/es:

KIM, JIN-KI

74 Agente/Representante:

CURELL AGUILÁ, Mireia

ES 2 498 096 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

Esquema de control de sistema de memoria Flash.

5 **Referencia cruzada a solicitudes relacionadas**

La presente solicitud reivindica la prioridad de la solicitud de patente provisional US n.º 60/788.083, presentada el 31 de marzo de 2006.

10 **Campo de la invención**

La presente invención se refiere en general a memorias Flash. Más particularmente, la presente invención se refiere a un sistema de memoria Flash de múltiples dispositivos para aplicaciones de almacenamiento de gran capacidad.

15

Antecedentes de la invención

La memoria Flash es un tipo de memoria no volátil usado comúnmente y de manera extendida como medios de almacenamiento de gran capacidad para electrónica de consumo, tal como, por ejemplo, cámaras digitales y reproductores portátiles digitales de música. La densidad de un chip de memoria Flash disponible en la actualidad puede ser de hasta 32 Gbits (4 GB), lo cual resulta adecuado para su uso en unidades de almacenamiento Flash USB populares puesto que el tamaño de un chip Flash es pequeño.

20

La llegada de cámaras digitales de 8 megapíxeles y de dispositivos portátiles digitales de entretenimiento con capacidades musicales y de vídeo ha estimulado la demanda de capacidades ultra-altas para almacenar las grandes cantidades de datos a las que no puede llegar el dispositivo de memoria Flash individual. Por lo tanto, múltiples dispositivos de memoria Flash se combinan entre sí formando un sistema de memoria para incrementar de manera eficaz la capacidad de almacenamiento disponible. Por ejemplo, para tales aplicaciones puede que se requieran densidades de almacenamiento Flash de 20 GB.

25

30

La Figura 1 es un diagrama de bloques de un sistema de memoria Flash 10 de la técnica anterior en el que se ha integrado un sistema de anfitrión 12. El sistema de memoria Flash 10 incluye un controlador de memorias Flash 14 en comunicación con el sistema de anfitrión 12, y múltiples dispositivos de memoria no volátil 16. El sistema de anfitrión incluirá un dispositivo de procesamiento, tal como un microcontrolador, microprocesador, o un sistema de ordenador. El sistema de memoria Flash 10 de la Figura 1 está configurado para incluir un canal 20, donde dispositivos de memoria 16 se conectan en paralelo al canal 20. Aquellos expertos en la materia entenderán que el sistema de memoria 10 tendrá conectados al mismo, un número cualquiera de dispositivos de memoria.

35

El canal 20 incluye un conjunto de buses comunes, los cuales incluyen líneas de datos y de control que están conectadas a todos sus dispositivos de memoria correspondientes. Aunque no se muestra, cada dispositivo de memoria se habilita/deshabilita con una señal respectiva de selección de chip proporcionada por el controlador de memorias Flash 14. El controlador de Flash 14 es responsable de emitir órdenes y datos, por medio del canal, a un dispositivo de memoria seleccionado sobre la base del funcionamiento del sistema de anfitrión 12. Datos leídos de los dispositivos de memoria se transfieren, por medio del canal de vuelta al controlador de memorias Flash 14 y al sistema de anfitrión 12. Al sistema de memoria Flash 10 se le hace referencia en general como configuración *multi-drop*, en la cual los dispositivos de memoria 16 están conectados en paralelo con respecto al canal 20.

40

45

En el sistema de memoria Flash 10, los dispositivos de memoria no volátil 16 son idénticos entre sí, y típicamente se implementan como dispositivos de memoria Flash NAND. Aquellos expertos en la materia entenderán que la memoria Flash está organizada en bancos, y cada banco está organizado en bloques para facilitar el borrado de bloques. Los dispositivos de memoria Flash NAND con mayor disponibilidad comercial están configurados para tener dos bancos de memoria. Antes de ofrecer una descripción del funcionamiento del sistema de memoria Flash 10, se expone una breve visión general de un núcleo de memoria con un solo dispositivo de memoria Flash NAND.

50

La Figura 2 es un diagrama de bloques general de un banco de una memoria Flash NAND conocida. El banco 30 está organizado en $k+1$ bloques. Cada bloque está compuesto por cadenas de celdas de memoria NAND, presentando hasta $i+1$ celdas de memoria Flash conectadas en serie entre sí. Por consiguiente, las líneas de palabra WL0 a WLi están conectadas a las puertas de cada celda de memoria Flash de la cadena de celdas de memoria. Un dispositivo de selección de cadenas conectado a la señal SSL (línea de selección de cadenas) conecta selectivamente la cadena de celdas de memoria a una línea de bit, mientras que un dispositivo de selección de tierra conectado a la señal GSL (línea de selección de tierra) conecta selectivamente la cadena de celdas de memoria a una línea de fuente, tal como VSS. El dispositivo de selección de cadenas y el dispositivo de selección de tierra son transistores de n canales. Hay $j+1$ líneas de bit comunes para todos los bloques del banco 30, y cada línea de bit está conectada a una cadena de celdas de memoria NAND en cada uno de los bloques [0] a [k]. Cada señal de línea de palabra (WL0 a WLi), SSL y GSL está conectada al mismo dispositivo de transistor correspondiente en cada cadena de celdas de memoria NAND del bloque. Tal como deberían conocer aquellos expertos en la materia, a los

60

65

datos almacenados en las celdas de memoria Flash a lo largo de una línea de palabra se les hace referencia como página de datos.

5 Conectado a cada línea de bit fuera del banco 30 se encuentra un registro de datos 32 para almacenar una página de datos de escritura a programar en una página de celdas de memoria Flash. El registro de datos 32 incluye también circuitos de detección para detectar datos leídos de una página de celdas de memoria Flash. Durante operaciones de programación, los registros de datos llevan a cabo operaciones de verificación de programas para garantizar que los datos se han programado correctamente en las celdas de memoria Flash conectadas a la línea de palabra seleccionada. La programación dentro de un bloque se inicia típicamente en la página correspondiente a 10 WL0, y prosigue secuencialmente hasta WLi para llenar el bloque actual. A continuación, la programación continúa con la WL0 de un bloque nuevo. Dentro de un dispositivo, los bloques se programan de manera secuencial.

Volviendo al sistema de memoria Flash 10 de la Figura 1, existen cuestiones específicas que tendrán un impacto negativo en el rendimiento del sistema. Algunas son físicas mientras que otras son arquitectónicas.

15 La configuración del sistema de memoria Flash 10 impone limitaciones físicas sobre el rendimiento. Con el alto número de señales paralelas que se extienden por el sistema, la integridad de señal correspondiente a las señales que transportan se verá deteriorada por la interferencia cruzada, el desfase de la señal, y el ruido de conmutación simultánea (SSN). El consumo de energía en una configuración de este tipo se convierte en un problema en la medida en la que cada pista de señal entre el controlador de flash y los dispositivos de memoria flash se carga y descarga frecuentemente para la señalización. Con frecuencias cada vez mayores del reloj del sistema, el consumo de energía también se incrementará.

25 Desde un punto de vista arquitectónico, las operaciones de programación ocuparán demasiado tiempo. Una de las funciones principales del controlador de Flash 14 es gestionar la escritura de datos en los dispositivos de memoria del sistema. En el contexto de las memorias Flash, a la escritura de datos se le hace referencia más comúnmente como programación de datos. Existen dos cuestiones significativas relacionadas con la programación Flash. En primer lugar, la programación Flash es lenta con respecto a memorias volátiles, tales como la DRAM y SRAM, y otras memorias no volátiles, tales como las unidades de disco duro. La programación de datos en celdas de memoria Flash requiere voltajes altos y una secuencia de programación escalonada para obtener una distribución ajustada de voltajes de umbral programados. En un dispositivo de memoria Flash NAND que tiene dos bancos de memoria, se programan simultáneamente dos páginas de datos, una para cada banco. Puesto que existe solamente un registro de datos por cada banco, las operaciones de programación adicionales deben esperar hasta que las páginas actuales hayan sido programadas satisfactoriamente. Por lo tanto, la programación de grandes cantidades de datos en los dispositivos Flash 16 puede requerir una cantidad significativa de tiempo.

Una segunda cuestión en relación con el sistema de memoria Flash 10 convencional es la estructura lineal de archivos de los datos de programación. La Figura 3 es una ilustración de la estructura convencional de archivos para un sistema de memoria Flash 50 que tiene cuatro dispositivos de memoria. En la Figura 3, cada dispositivo de memoria 52, 54, 56 y 58 tiene un total de n páginas físicas de espacio de almacenamiento, que se dividen entre un número cualquiera de bloques. En el ejemplo mostrado en la presente, se supone que las n páginas se dividen equitativamente entre dos bancos. La mayoría de sistemas de memoria Flash almacenarán un archivo de datos compuesto por una serie de páginas de datos, linealmente dentro de un dispositivo de memoria. Por ejemplo, la primera página del archivo de datos se almacena en la Página 0 del dispositivo 52, y las páginas de datos sucesivas se almacenan progresivamente en páginas posteriores. Una vez que el dispositivo 52 está lleno, entonces los archivos de datos adicionales a almacenar en el sistema 50 comienzan en la Página 0 del dispositivo 54, y así sucesivamente. La flecha 60 muestra el patrón de almacenamiento de datos que se describe en el sistema de memoria Flash 50.

50 Esta estructura lineal de archivos junto con el tiempo de programación relativamente prolongado por cada página del archivo de datos por dispositivo de memoria, da como resultado un sistema de memoria Flash que requiere un tiempo significativo para almacenar datos. Otra cuestión que está en relación con la estructura lineal de archivos es la fiabilidad de los dispositivos, y más específicamente, el desgaste por programación/borrado de un dispositivo de memoria con respecto a los otros dispositivos de memoria del sistema. El desgaste por programación/borrado se refiere a un deterioro progresivo de una celda de memoria Flash debido a las operaciones acumuladas de programación y borrado. El efecto de dichas operaciones acumuladas de programación y borrado es la alteración de las características de programación y borrado de la celda de memoria más allá de parámetros óptimos. Cuando las celdas de memoria se deterioran, son necesarios voltajes más altos de programación y borrado para programar y borrar las celdas de memoria con respecto a los voltajes de umbral deseados. Finalmente, las celdas de memoria no conseguirán funcionar correctamente. Este es el motivo por el que las memorias Flash tienen un número limitado nominal de ciclos de borrado y una programación, el cual se sitúa entre 10.000 y 100.000 ciclos.

Si, por ejemplo, el primer dispositivo de memoria 52 de la Figura 3 soporta más ciclos de programación y borrado que cualquiera de los otros dispositivos de memoria, el dispositivo de memoria 52 probablemente fallará antes de los otros. Cuando el dispositivo de memoria 52 falla, el sistema completo 50 ya no es utilizable puesto que los dispositivos de memoria están encapsulados juntos y la sustitución de un único dispositivo de memoria es poco

práctica. Esto es una pérdida indeseable de dispositivos de memoria, ya que los dispositivos restantes del sistema sigue siendo útiles y les puede quedar una vida significativa.

Una arquitectura técnica inherente de la mayoría de memorias Flash es que la unidad más pequeña de memoria que se puede borrar es un bloque de memoria. Esto significa que si, tan siquiera se va a modificar una página dentro del bloque, el bloque completo se debe re-programar junto con la página nueva. A esto se le hace referencia como re-programación de bloques, la cual requiere un tiempo de programación significativo y, por tanto, tiene un impacto negativo en el rendimiento del sistema.

El documento US 2005/0114589 da a conocer técnicas de nivelación de desgaste para sistemas de EEPROM Flash.

Por lo tanto, los sistemas de memoria Flash conocidos en la actualidad presentan un rendimiento lento en cuanto a la programación de datos, y, debido al desgaste desigual por programación y borrado en los dispositivos, el sistema completo presentará una vida útil limitada al fallo del primer dispositivo de memoria.

Es por ello deseable proporcionar una arquitectura de un sistema de memoria Flash de alta velocidad que presente un esquema para aumentar al máximo la vida útil del sistema.

Sumario de la invención

Es un objetivo de la presente invención eliminar o mitigar por lo menos una de las desventajas de los esquemas de control previos para sistemas de memoria Flash. En particular, es un objetivo de la presente invención mejorar el rendimiento de la programación de sistemas de memoria Flash mediante la intercalación de operaciones de programación en cada dispositivo de memoria Flash del sistema.

La invención proporciona un método según la reivindicación 1.

En una primera disposición, se proporciona un método para controlar un primer y un segundo dispositivos de memoria Flash conectados a un canal. El método incluye ejecutar una primera operación en el primer dispositivo de memoria Flash como respuesta a una primera orden, e iniciar una segunda operación en el segundo dispositivo de memoria Flash como respuesta a una segunda orden, mientras el primer dispositivo de memoria Flash está ejecutando la primera operación.

En una forma de realización de esta disposición, el primer dispositivo de memoria Flash y el segundo dispositivo de memoria Flash están conectados en serie entre sí, y la segunda orden se traslada al segundo dispositivo de memoria Flash a través del primer dispositivo de memoria Flash antes de la etapa de inicio. En otras formas de realización, la etapa de ejecutar la primera operación incluye programar por lo menos una página de un archivo de datos en el primer dispositivo de memoria. La segunda operación incluye iniciar la programación de por lo menos otra página del archivo de datos en el segundo dispositivo de memoria, mientras el método incluye además iniciar una tercera operación en el tercer dispositivo de memoria Flash como respuesta a una tercera orden. La segunda operación incluye iniciar una operación de lectura de datos en el segundo dispositivo de memoria o iniciar una operación de borrado en el segundo dispositivo de memoria. Todavía en otra forma de realización, la ejecución de la primera operación incluye una de entre una operación de lectura y una operación de borrado en el primer dispositivo de memoria.

En una segunda disposición, se proporciona un método para una programación de nivelación de desgaste de alta velocidad en un sistema de memoria Flash que tiene una pluralidad de dispositivos de memoria Flash. El método incluye recibir un archivo de datos que tiene k páginas, siendo k un entero mayor que 0; seleccionar un perfil de programación correspondiente al tamaño de k y parámetros de configuración del sistema de memoria Flash; y programar por lo menos una de las k páginas del archivo de datos y cada uno de por lo menos dos de la pluralidad de dispositivos de memoria Flash de acuerdo con el perfil de programación seleccionado.

Según una forma de realización de esta disposición, los parámetros de configuración incluyen j dispositivos de memoria Flash, y cada uno de los j dispositivos de memoria Flash tiene i páginas por bloque, donde j e i son valores enteros mayores que 0. La etapa de selección incluye calcular una función de techo de z , donde $z=k/i$, y el perfil de programación incluye una única estructura de archivos para almacenar k páginas del archivo de datos en z de j dispositivos de memoria Flash cuando z es inferior o igual a j . La etapa de programación incluye proporcionar secuencialmente órdenes de programación a cada uno de los z dispositivos de memoria Flash para programar las k páginas, donde cada orden de programación programa la por lo menos una de las k páginas.

En otra forma de realización de esta disposición, el perfil de programación incluye una estructura de archivos múltiple cuando z es mayor que j . La estructura de archivos múltiple incluye almacenar n unidades de j^i páginas del archivo de datos en j dispositivos de memoria Flash, y almacenar $k-(m*(j^i))$ páginas del archivo de datos en z de j dispositivos de memoria Flash cuando z es inferior o igual a j , donde m es un valor entero mayor que 0. La etapa de programación incluye proporcionar secuencialmente órdenes de programación a cada uno de los j dispositivos de memoria Flash para programar las j^i páginas del archivo de datos, donde cada orden de programación programa la

por lo menos una de las k páginas. La etapa de programación incluye además proporcionar secuencialmente órdenes de programación a cada uno de los z dispositivos de memoria Flash para programar las $k - (m \cdot (j^i))$ páginas, donde cada orden de programación programa la por lo menos una de las k páginas.

5 En una tercera disposición, se proporciona una arquitectura de almacenamiento de archivos de datos para un sistema de memoria que tiene por lo menos dos dispositivos de memoria conectados al mismo canal. La arquitectura de almacenamiento de archivos de datos incluye porciones del archivo de datos almacenados en dos de los por lo menos dos dispositivos de memoria. Según formas de realización del presente aspecto, las porciones son sustancialmente iguales entre sí en cuanto a tamaño, y las porciones se almacenan en cada uno de los por lo menos dos dispositivos de memoria del sistema de memoria.

10 En una cuarta disposición, se proporciona un método para una programación de nivelación de desgaste de alta velocidad en un sistema de memoria Flash que tiene j dispositivos de memoria Flash, donde cada uno de los j dispositivos de memoria Flash tiene i páginas por bloque, donde j e i son valores enteros mayores que 0. El método incluye recibir un archivo de datos que tiene k páginas, siendo k un entero mayor que 0; proporcionar órdenes para programar k páginas dentro de z de los j dispositivos de memoria si una función de techo de $z = k/i$ es inferior o igual a j ; proporcionar órdenes para programar j^i páginas dentro de j dispositivos de memoria si la función de techo de $z = k/i$ es mayor que j ; actualizar k fijando $k = k - (j^i)$; y repetir la etapa de programar k páginas actualizadas.

15 En una quinta disposición, se proporciona un sistema de memoria Flash. El sistema de memoria Flash incluye un controlador, un primer dispositivo de memoria Flash, y un segundo dispositivo de memoria Flash. El controlador tiene un canal para proporcionar una primera orden y una segunda orden. El primer dispositivo de memoria Flash está acoplado al canal para ejecutar una primera operación como respuesta a la primera orden. El segundo dispositivo de memoria Flash está acoplado al canal para iniciar una segunda operación como respuesta a la segunda orden mientras el primer dispositivo de memoria Flash está ejecutando la primera operación.

20 Según formas de realización de esta disposición, el primer dispositivo de memoria Flash y el segundo dispositivo de memoria Flash están conectados en serie entre sí, y la segunda orden se traslada al segundo dispositivo de memoria Flash a través del primer dispositivo de memoria Flash. La primera operación incluye una operación de programación, y el primer dispositivo de memoria Flash programa por lo menos una página de un archivo de datos. La segunda operación incluye otra operación de programación, y el segundo dispositivo de memoria Flash programa por lo menos otra página del archivo de datos.

25 Otros aspectos y características de la presente invención se pondrán de manifiesto para aquellos con conocimientos habituales en la materia, tras revisar la siguiente descripción de formas de realización específicas de la invención conjuntamente con las figuras adjuntas.

Breve descripción de los dibujos

30 A continuación se describirán, únicamente a título de ejemplo, formas de realización de la presente invención, en referencia a las Figuras adjuntas, en las que:

la Fig. 1 es un diagrama de bloques de un sistema de memoria Flash de la técnica anterior;

35 la Fig. 2 es un esquema de un núcleo de memoria Flash NAND de la técnica anterior;

la Fig. 3 es una ilustración de una estructura de archivos convencional para un sistema de memoria Flash;

40 la Fig. 4 es un diagrama de bloques de un sistema de memoria Flash según una forma de realización de la presente invención;

la Fig. 5 es un diagrama de temporización de órdenes de programa emitidas por el controlador de Flash de la Figura 4, según una forma de realización de la presente invención;

45 la Fig. 6 es un diagrama de temporización que muestra órdenes de programa recibidas por cada dispositivo de memoria del sistema de memoria Flash de la Figura 4;

la Fig. 7 es un diagrama de flujo que ilustra un método de programación intercalada de alta velocidad según una forma de realización de la presente invención;

50 la Fig. 8 es una ilustración gráfica de una estructura de archivos en el sistema de memoria Flash de la Figura 4 resultante del método de programación intercalada de alta velocidad de la Figura 7;

55 la Fig. 9 es un diagrama de flujo que ilustra un método de programación de nivelación de desgaste de alta velocidad, según una forma de realización de la presente invención; y

la Fig. 10 es un diagrama de temporización que muestra operaciones intercaladas de programación y lectura.

Descripción detallada

5 Una arquitectura de un sistema de memoria Flash que tiene dispositivos de memoria Flash conectados en serie para lograr una programación de datos de alta velocidad. La programación de datos de alta velocidad se logra intercalando páginas de los datos entre los dispositivos de memoria del sistema, de tal manera que, en dispositivos de memoria diferentes, se almacenan páginas de datos diferentes. Un controlador de memoria emite órdenes de programación para cada dispositivo de memoria en un flujo continuo de bits que tiene una o más líneas de señales.
 10 A medida que cada dispositivo de memoria recibe una orden de programación, o bien da inicio a una operación de programación o bien traslada la orden al siguiente dispositivo de memoria. Por lo tanto, los dispositivos de memoria del sistema Flash programan secuencialmente páginas de datos una tras otra, minimizando así el retardo de la programación de cada página de datos en el sistema de memoria Flash. El controlador de memoria ejecuta un algoritmo de control del nivel de desgaste para optimizar el rendimiento y la resistencia de la programación para
 15 datos de cualquier tamaño.

La Figura 4 es un diagrama de bloques de un sistema de memoria Flash que tiene dispositivos de memoria conectados en serie, para ejecutar operaciones de programación de datos de alto nivel con control del nivel de desgaste, según una forma de realización de la presente invención. El sistema de memoria Flash 100 incluye un controlador de memorias Flash 102 en comunicación con el sistema de anfitrión 104, y cuatro dispositivos de memoria Flash conectados en serie 106, 108, 110 y 112. Aunque en la presente forma de realización se muestran cuatro dispositivos de memoria, las formas de realización de la presente invención resultarán eficaces para sistemas de memoria que tengan por lo menos dos dispositivos de memoria. Cada uno de los cuatro dispositivos de memoria Flash tiene un circuito de interfaz de entrada/salida serie para facilitar el funcionamiento en serie entre dispositivos de memoria. Se describe un ejemplo de un dispositivo de memoria Flash de este tipo en la solicitud de patente US, de propiedad conjunta, n.º de serie 11/324.023, presentada el 30 de diciembre de 2005, y en la solicitud de patente US, de propiedad conjunta, n.º de serie 11/496.278, presentada el 31 de julio de 2006, cuyo contenido se incorpora a la presente a título de referencia. El dispositivo de memoria Flash de la solicitud de patente US 11/324.023 se le hace referencia como dispositivo de múltiples enlaces en serie independientes (MISL). Como en el sistema mostrado en la Figura 1, el sistema de anfitrión incluirá un dispositivo de procesado, tal como un microcontrolador, microprocesador, o un sistema de ordenador.

El dispositivo de memoria Flash 106 es el primer dispositivo de la cadena, y recibe órdenes, tales como órdenes de lectura, programación y borrado, del controlador de memorias Flash 102. En el dispositivo Flash MISL de la solicitud de patente US n.º 11/324.023, todas las órdenes, datos e información de direcciones se reciben en forma de un flujo continuo de bits en serie. Las órdenes incluyen información de datos, información de direcciones, y cualquier otra información requerida por el dispositivo de memoria para ejecutar una operación particular. En la medida en la que cada dispositivo de memoria Flash debería incluir circuitería lógica de flujo pasante, cualquier orden recibida no destinada a un dispositivo particular se traslada al siguiente dispositivo de memoria Flash, y así sucesivamente, hasta que el dispositivo de memoria Flash pretendido actúa sobre la misma. El último dispositivo de memoria Flash 112 de la cadena tiene salidas conectadas al controlador de memorias Flash 102, para proporcionar datos de lectura como respuesta a una orden de lectura. La forma de realización de la Figura 4 mostrada en este momento tiene un canal para enviar y recibir datos desde la cadena de dispositivos de memoria Flash. Aquellos expertos en la materia entenderán que el controlador de memorias Flash 102 tendrá opcionalmente múltiples canales para dar acomodo a un número correspondiente de cadenas de dispositivos de memoria Flash. Puesto que cada dispositivo de memoria Flash está posicionado preferentemente muy próximo a otro dispositivo, se minimizarán los hilos conductores que interconectan entre sí los dispositivos de memoria Flash. Por lo tanto, no existen limitaciones físicas sobre el rendimiento en relación con la longitud de las líneas de bus, como en el sistema de memoria Flash 10 de la Figura 1.

Tal como se ha descrito previamente, el controlador de memorias Flash 102 es responsable de emitir órdenes de programación para cada dispositivo de memoria Flash. Para la programación de datos de alta velocidad según una forma de realización de la presente invención, páginas de un archivo de datos se programan en dispositivos de memoria diferentes. Esto se realiza emitiendo órdenes de programación en serie una tras otra para iniciar operaciones de programación en cada dispositivo de memoria Flash en una rápida sucesión. Las Figuras 5 y 6
 55 ayudarán a ilustrar cómo se materializa dicha programación de alta velocidad.

La Figura 5 es un diagrama de temporización que ilustra la emisión de órdenes de programación por parte del controlador de memorias Flash 102 para aumentar al máximo la velocidad de programación total correspondiente al sistema de memoria Flash 100, de acuerdo con una forma de realización de la presente invención. En el ejemplo mostrado en este momento, se emiten en serie cuatro órdenes de programación Datos[0], Datos[1], Datos[2] y Datos[3] en periodos de tiempo respectivos t1 a t4 por parte del controlador de memorias Flash 102 para programar un archivo de datos. El número de identificación entre corchetes para cada orden de programación indica el orden secuencial en el que se emitió la orden de programación desde el controlador de memorias Flash 102. Cada orden de programación incluirá, aunque sin carácter limitativo, una orden 200 y por lo menos una página de datos 202 para un dispositivo de memoria específico. La orden 200 incluye una dirección en la cual se va a programar la página de datos, y un identificador de dispositivo para hacer coincidir la orden de programación con el dispositivo de memoria

específico. Los dispositivos de memoria Flash 106, 108, 110 y 112 son sensibles, respectivamente, a las órdenes de programación Datos[0], Datos[1], Datos[2] y Datos[3]. Puesto que los dispositivos de memoria Flash están conectados en serie y las órdenes de programación se emiten en serie, las operaciones de programación fundamental de cada dispositivo de memoria Flash se solapan con las operaciones del dispositivo de memoria Flash subsiguiente, con la excepción del último dispositivo de memoria Flash que recibe una orden de programación.

Si se supone que cada orden de programación requiere aproximadamente 85 microsegundos para la transferencia al dispositivo de memoria (tiempo t1), entonces la secuencia total para transferir las cuatro órdenes de programación requerirá 4×85 microsegundos = 340 microsegundos. El tiempo requerido para programar la por lo menos una página de datos por cada dispositivo de memoria está fijado, y se supone que es aproximadamente 200 microsegundos. Por lo tanto, el tiempo transcurrido total para programar todos los datos en el sistema de memoria Flash 100 será aproximadamente 340 microsegundos + 200 microsegundos = 540 microsegundos. El tiempo de programación de 200 microsegundos es aportado por el último dispositivo de memoria para recibir una orden de programación. Por contraposición, la programación de 4 páginas en el mismo dispositivo de memoria ocupará 1.140 microsegundos. Las operaciones de programación con solape de los dispositivos de memoria Flash se muestran más claramente en el diagrama de temporización extendido de la Figura 6.

La Figura 6 es un diagrama de temporización extendido que muestra las operaciones de dispositivos de memoria Flash 106, 108, 110 y 112 como respuesta a órdenes de programación respectivas Datos[0], Datos[1], Datos[2] y Datos[3]. Los periodos de tiempo t1 a t4 se corresponden con los mismos periodos de tiempo mostrados en la Figura 5. A continuación se describirá, en referencia al diagrama de flujo de la Figura 7, la secuencia de programación de la Figura 6. A la forma de realización de control de la programación del sistema Flash descrita en la Figura 7 se le hará referencia como método de programación intercalada de alta velocidad, por el cual, entre dispositivos de memoria diferentes, se intercalan operaciones de programación. El método comienza en la etapa 300 en la que se proporcionan en serie por lo menos dos órdenes de programación al primer dispositivo de memoria 106 del sistema. La primera orden de programación se corresponde con la orden de programación Datos[0] mientras que la segunda orden de programación se corresponde con la orden de programación Datos[1]. En la etapa 302, el dispositivo de memoria 106 recibe la orden de programación Datos[0] durante un primer periodo de tiempo t1, al cual le sucede una operación de programación en la etapa 304. Durante el periodo de tiempo t1, la orden de programación Datos[0] se transfiere al dispositivo de memoria 106 mientras los dispositivos de memoria 108, 110 y 112 permanecen en un estado sin operación (NOP).

En cuanto la orden de programación Datos[0] se ha transferido al dispositivo de memoria 106, el dispositivo de memoria 108 recibe la segunda orden de programación Datos[1] durante un segundo periodo de tiempo t2 en la etapa 306. En la etapa 308 se prosigue con una operación de programación, y, puesto que la operación de programación para el dispositivo de memoria 106 ya ha comenzado al final del periodo de tiempo t1, existe un periodo de tiempo en el que los dos dispositivos de memoria 106 y 108 están ejecutando operaciones de programación al mismo tiempo. El proceso se repetiría de la misma manera para la siguiente orden de programación y dispositivo de memoria. En el presente ejemplo, las operaciones de programación en los dispositivos de memoria 106 y 108 se habrán completado antes del final del periodo de tiempo t4. La operación de programación en el dispositivo de memoria 110 se completará a medida que continúe la operación de programación en el dispositivo de memoria 112.

El ejemplo anterior presenta un escenario en el que el primer dispositivo de memoria que va a recibir una orden es el 106. Alternativamente, uno cualquiera de los dispositivos de memoria del sistema puede ser el primer dispositivo de memoria que va a recibir la primera orden de programación. Una quinta orden de programación se emite al dispositivo de memoria 106 después del periodo de tiempo t4, siempre que el dispositivo de 106 haya finalizado su operación de programación. Aquellos expertos en la materia entenderán que dispositivos de memoria Flash diferentes presentarán tiempos de programación variables. En el ejemplo de la Figura 6 mostrado en la presente, el dispositivo de memoria 106 ha acabado de programar sus datos en el comienzo del periodo de tiempo t4, por lo tanto está preparado para recibir la siguiente orden de programación Datos[4] en cuanto el dispositivo de memoria 112 haya acabado de recibir la orden de programación Datos[3]. Si el dispositivo de memoria Flash tiene un tiempo de programación fundamental muy prolongado, entonces será necesario que el controlador de memorias Flash espere hasta que el dispositivo de memoria 106 haya completado las operaciones de programación antes de emitir la orden de programación Datos[4]. Típicamente, los dispositivos de memoria Flash proporcionarán una señal de estado preparado al controlador de memorias Flash para indicar cuándo se ha completado una operación de programación.

La Figura 8 es una ilustración gráfica de una estructura de archivos de dispositivos de memoria Flash 106, 108, 110 y 112 después de que se haya ejecutado la secuencia de programación de alta velocidad descrita en la Figura 6 y la Figura 7 para órdenes de programación Datos[0], Datos[1], Datos[2], Datos[3] y Datos[4]. La estructura de archivos mostrada en la Figura 8 supone que los bloques de memoria de los dispositivos 106, 108 y 112 están vacíos antes de la programación. Este es el motivo por el que las páginas se programan en la página física 0 de estos dispositivos. Por otro lado, en el dispositivo de memoria 110, el bloque de memoria puede tener otros datos en las páginas físicas 0 y 1. Por lo tanto, los Datos[2] se programan en la siguiente página disponible del bloque, que es la página física 2. Tal como se muestra en la Figura 8, la página física 1 del dispositivo de memoria 106 es la última

página a programar en la secuencia de programación de alta velocidad mostrada en la presente. La siguiente secuencia de programación comenzará a programar datos en la página física 1 del dispositivo de memoria 108, y proseguirá de la misma manera que la descrita previamente para la secuencia de programación mostrada en la Figura 6. Por lo tanto, se obtiene la máxima velocidad de programación del sistema de memoria Flash 100 cuando se emiten en serie órdenes de programación a dispositivos de memoria conectados en serie, consecutivos. En otras palabras, una estructura de archivos en la cual se distribuyen datos por el número máximo de dispositivos de memoria en el sistema de memoria Flash interconectado en serie dará como resultado la velocidad de programación más alta de un archivo de datos. Debería indicarse que no se requiere necesariamente que los datos se programen en el mismo número de página física por todos los dispositivos de memoria.

Aunque la programación de alta velocidad es beneficiosa para sistemas que usan el sistema de memoria Flash 100, algunos sistemas pueden requerir una resistencia del sistema de memoria Flash 100 aumentada al máximo. La estructura de archivos mostrada en la Figura 8 no aumentará al máximo el desgaste por programación/borrado que pasa por todos los dispositivos de memoria, debido principalmente a la arquitectura de borrado de bloques de los dispositivos de memoria Flash. Por ejemplo, si se programa con intercalación un archivo de datos de 20 páginas en un bloque en cada dispositivo de memoria, entonces existe una posibilidad de que la totalidad de los cuatro dispositivos de memoria requieran llevar a cabo un borrado de bloques antes de actualizar el archivo de datos. Por contraposición, si la totalidad de las 20 páginas se almacenase en un bloque de un dispositivo de memoria, entonces es necesario borrar solamente ese bloque.

Por lo tanto, según otra forma de realización de la invención, el método antes descrito de control de programación de alta velocidad se ajusta para minimizar el desgaste por programación/borrado, o para optimizar el rendimiento de la programación y el desgaste por programación/borrado. Más específicamente, el controlador de Flash 102 de la Figura 4 ejecuta un algoritmo de control de programación para optimizar el rendimiento de la programación y el desgaste por programación/borrado sobre la base de criterios predeterminados. Estos criterios predeterminados incluirán características de los datos a programar y características de los dispositivos de memoria del sistema de memoria Flash. Las características de los datos incluyen el número de páginas de los datos a programar, y las características de los dispositivos de memoria incluyen el número de páginas por bloque.

La Figura 9 es un diagrama de flujo que ilustra una forma de realización para programar múltiples dispositivos de memoria Flash en un sistema de memoria Flash con alta velocidad y con control de nivelación del desgaste. A esta forma de realización de control de la programación se le hará referencia como método de programación de nivelación del desgaste de alta velocidad. La nivelación del desgaste hace referencia a un esquema para prolongar la vida del sistema de memoria Flash. La forma de realización descrita en este momento optimizará la nivelación del desgaste y el rendimiento para cualquier archivo de datos a programar, usando una variedad de perfiles de programación. Un perfil de programación se corresponde en general con una secuencia de programación para almacenar páginas de un archivo de datos con una estructura específica de almacenamiento de archivos. Finalmente, el perfil de programación distribuye páginas de archivo de datos por los dispositivos de memoria del sistema de memoria Flash. El método descrito en este momento es ejecutable por un controlador de memorias, tal como el controlador de Flash 102 de la Figura 4, en un sistema de dispositivos de memoria conectados en serie.

El método de programación de nivelación del desgaste de alta velocidad comienza en la etapa 400 en donde una variable i se fija al número de páginas por bloque de cada dispositivo de memoria del sistema de memoria Flash, y una variable j se fija al número de dispositivos de memoria en el sistema de memoria Flash. Se presupone que todos los dispositivos de memoria del sistema de memoria Flash son idénticos entre sí, y tienen el mismo tamaño de bloque. Esta información se pre-programa en el controlador de memorias. En la etapa 402, el controlador de memoria recibe un archivo de datos compuesto por un número de páginas k , para la programación. Prosiguiendo hacia la etapa 404, se realiza un cálculo para determinar si k es inferior o igual a i . Si k es inferior o igual a i , lo cual significa que el archivo de datos es inferior o igual a un bloque de espacio de almacenamiento en un dispositivo de memoria, entonces la totalidad de las k páginas del archivo de datos se programa en un bloque de un dispositivo de memoria en la etapa 406. Este es un ejemplo de un perfil de programación que presenta una única estructura de archivos. El controlador de memorias selecciona el dispositivo de memoria específico en el cual se programará el archivo de datos de acuerdo con uno o más parámetros de selección. Por ejemplo, un parámetro de selección es el dispositivo de memoria con el número más alto de ciclos restantes de programación/borrado, mientras que otro parámetro de selección incluye el último dispositivo de memoria en el que se va a programar.

Por otro lado, si k es mayor que i , lo cual significa que el archivo de datos incluye más páginas que un bloque de espacio de almacenamiento en un dispositivo de memoria, entonces el método prosigue a la etapa 408. En la etapa 408, se realiza un cálculo para determinar si k/i es inferior o igual a j . Debería indicarse que el cálculo de k/i debería generar únicamente números enteros. Puesto que el presente método determina el número mínimo de bloques requeridos para almacenar los k datos, un resultado no entero que presente una parte entera y una parte decimal (es decir, un número real) indica que se requiere un bloque más que el número entero. Esto se realiza a través de funciones matemáticas conocidas, tales como una función de techo. Aquellos expertos en la materia entenderán que una función de techo devuelve el entero más pequeño que no es menor que el número real. Por otro lado, un resultado entero directo de k/i no requiere un procesamiento matemático adicional. Desde este momento en adelante, la referencia al resultado de k/i presupondrá que se ha aplicado al mismo una función de techo.

Si el valor entero de k/i es inferior a j , el número de dispositivos de memoria en el sistema de memoria Flash, entonces las k páginas del archivo de datos se programan con intercalación entre k/i dispositivos de memoria en la etapa 410. Este es otro ejemplo de un perfil de programación que presenta una única estructura de archivos. La programación con intercalación proseguirá tal como se ha descrito previamente en el método de la Figura 7. En un ejemplo práctico, si $i=32$, $j=4$ y $k=61$, entonces $k/i=1,90$. Puesto que k/i da como resultado un número real, una función de techo aplicada a 1,90 da como resultado el número entero 2. Por tanto, en la etapa 410, la totalidad de las $k=61$ páginas se programan con intercalación por 2 dispositivos de memoria. El controlador de memorias selecciona dos dispositivos de memoria cualesquiera del sistema de memoria Flash, ya estén conectados directamente o conectados indirectamente entre sí. Dos dispositivos de memoria conectados indirectamente pueden tener por lo menos un dispositivo de memoria intermedio conectado entre ellos. Puesto que el archivo de datos se programa en dos dispositivos de memoria, aproximadamente la mitad de las páginas se programa en un dispositivo de memoria y el resto se almacena en el otro dispositivo de memoria.

Si k/i es por lo menos j , el número de dispositivos de memoria del sistema, entonces se usarán secuencias diferentes de programación con optimización para programar grupos de tamaño diferente del archivo de datos. Más específicamente, los archivos de datos muy grandes se tratan como múltiples unidades de archivos de datos más pequeños, los cuales se programan de acuerdo con una cualquiera de las secuencias de programación previamente descritas. Continuando en la etapa 412, todas las posiciones de las páginas en un bloque de cada dispositivo de memoria se programan con j^i páginas de las k páginas del archivo de datos, según la secuencia de programación intercalada de la etapa 410. Siguiendo en la etapa 414, el número de páginas k se actualiza fijándolo igual a $k-(j^i)$. Por lo tanto, se calcula el número restante de páginas a programar. El método realiza un bucle de vuelta a la etapa 404 para repetir las secuencias de procesado y programación del árbol de decisiones basándose en el valor actualizado de k . En resumen, el presente método programará de manera iterativa múltiples unidades de j^i páginas del archivo de datos usando la misma estructura de archivos para cada unidad de j^i páginas, y a continuación programará las $k-(j^i)$ páginas restantes usando una estructura de archivos diferente. Por tanto, el presente método tiene un perfil de programación compuesto por múltiples estructuras de archivos para el archivo de datos.

A continuación se usará un ejemplo práctico para ilustrar esta forma de realización. Si $i=32$, $j=4$ y $k=192$, entonces las primeras 128 páginas se programarán por todos los dispositivos de memoria según se describe en la etapa 412. En la etapa 414, k se actualiza de manera que es $192-(128)=64$. A continuación, las 64 páginas restantes se programan por dos dispositivos de memoria según se describe en la etapa 410. Tal como se ha mencionado previamente, se seleccionan dos dispositivos de memoria cualesquiera para programar las 64 páginas restantes. Aunque la forma de realización descrita en este momento ilustra un método en el que a la programación le sucede el recálculo de k , la secuencia completa la puede determinar el controlador de Flash de antemano y antes de que comiencen cualesquiera operaciones de programación utilizando los cálculos antes descritos. Aunque se presupone que las primeras páginas del archivo de datos a programar son las j^i páginas, las primeras páginas a programar pueden ser en cambio las $k-(j^i)$ páginas, seguidas por las múltiples unidades de j^i páginas.

El método de programación de nivelación del desgaste de alta velocidad descrito en este momento se ha descrito para un sistema de memoria Flash que tiene un único canal, tal como la forma de realización mostrada en la Figura 4. Las formas de realización previamente descritas de la invención se ejecutan en un sistema de memoria Flash que tiene dos o más canales. En dichas formas de realización alternativas, por lo menos dos archivos de datos se programan simultáneamente; un primer archivo de datos en un canal y un segundo archivo de datos en otro canal.

Adicionalmente, la presente forma de realización se ha descrito en funcionamiento con dispositivos de memoria que tienen un único banco de memoria. Evidentemente, se pueden usar dispositivos de memoria que tengan dos o más bancos de memoria. Con dos bancos de memoria, habrá dos memorias intermedias de páginas disponibles para almacenar hasta dos páginas de datos. En una configuración de dispositivos con múltiples bancos, hay disponibles varias opciones de programación. En una primera opción, todas las páginas de datos a programar en un dispositivo de memoria se programan en un bloque dentro de un banco del dispositivo de memoria. La operación sería análoga a un dispositivo de memoria con solamente un banco de memoria. En una segunda opción, dos páginas del archivo de datos se cargan simultáneamente en un dispositivo de memoria, o en una única orden de programación. Esto hará que aumente eficazmente el rendimiento de la programación ya que cada dispositivo de memoria programará dos páginas al mismo tiempo. En una tercera opción, se intercalan operaciones de programación entre bancos de los dispositivos de memoria. Por ejemplo, la programación de páginas en dos dispositivos de memoria se producirá en la siguiente secuencia: dispositivo 1 [banco 1], dispositivo 2 [banco 1], dispositivo 1 [banco 2], y dispositivo 2 [banco 2]. La secuencia de programación debería resultar evidente para dispositivos de memoria que tengan más de 2 bancos.

Además, aunque el método de programación de nivelación del desgaste de alta velocidad descrito previamente se ha descrito en sistemas de memoria Flash que tienen dispositivos de memoria conectados en serie, o en cadena (*daisy*), las formas de realización son aplicables a sistemas Flash configurados en *multi-drop* tal como el mostrado en la Figura 1. Esto se realiza habilitando el dispositivo de memoria apropiado y proporcionando los datos de órdenes correspondientes sobre los buses comunes en tiempos diferentes.

Las formas de realización previamente descritas de la invención ilustran ejemplos en los que se intercalan operaciones de programación en el sistema de memoria Flash de la Figura 4. Según otra forma de realización de la invención, el sistema de memoria Flash de la Figura 4 ejecuta operaciones intercaladas de programación y lectura.

5 La Figura 10 es un diagrama de temporización que ilustra operaciones intercaladas de lectura y programación de alta velocidad para el sistema de memoria Flash de la Figura 4. En este ejemplo, los dispositivos de memoria 106 y 112 se van a programar con datos, mientras que los dispositivos de memoria 108 y 110 van a proporcionar datos de lectura. Durante el periodo de tiempo t1, el dispositivo de memoria 106 recibe una orden de programación Datos[0], y a continuación da inicio inmediatamente a secuencias internas del núcleo para programar los datos. En el
10 comienzo del periodo de tiempo t2, el dispositivo de memoria 108 recibe una orden de lectura, y prosiguen operaciones de transferencia interna de datos (xfer). Una operación de transferencia interna de datos ocupará, por ejemplo, 20 microsegundos, para leer los datos y cargar los registros de datos, tiempo tras el cual el controlador de Flash emitirá la orden de lectura para el dispositivo de memoria 110 en el comienzo del periodo de tiempo t3. El dispositivo de memoria 108 entra en un periodo interno de NOP 500 para permitir el flujo pasante de una orden a un
15 dispositivo de memoria de aguas abajo antes de dar salida a datos desde sus registros de datos.

El dispositivo de memoria 110 comenzará su operación de transferencia interna de datos después de recibir su orden de lectura respectiva. No obstante, ahora que las líneas de señal entre los dispositivos de memoria 108 y 110 no están siendo utilizadas, el dispositivo de memoria de aguas arriba 108 comenzará a dar salida a los datos en sus
20 registros de datos al dispositivo de memoria 110, el cual deja pasar los datos al dispositivo de memoria 112. Esto se realiza, por ejemplo, de una manera secuencial. Al final del periodo de tiempo t3, el dispositivo de memoria 108 habrá acabado de dar salida a todos sus datos de lectura, permitiendo así que el dispositivo de memoria 110 comience a dar salida a sus datos de lectura en el inicio del periodo de tiempo t4. Puesto que el dispositivo de memoria 110 no puede dar salida a sus datos hasta que el dispositivo de memoria 108 haya acabado de dar salida a
25 sus datos, se entra en un periodo de NOP interno 502. Al final del periodo de tiempo t4, las líneas de señal entre los dispositivos de memoria 110 y 112 no estarán siendo utilizadas ya que a todos los datos del dispositivo de memoria 110 se les ha dado salida a través del dispositivo de memoria 112. Por tanto, en el comienzo del periodo de tiempo t5, el dispositivo de memoria 112 recibirá la orden de programación Datos[1].

30 Aunque se han ilustrado operaciones intercaladas de lectura y programación, las operaciones intercaladas de lectura, programación y borrado son ejecutables en cualquier combinación.

El método previamente descrito de programación intercalada de alta velocidad se usa para aumentar al máximo el rendimiento de la programación en un sistema de memoria Flash que tiene dispositivos de memoria conectados en serie. La programación intercalada de alta velocidad se aplica a archivos de datos de cualquier tamaño. No obstante,
35 para mejorar la resistencia de todos los dispositivos de memoria, se usa un método de programación de nivelación del desgaste de alta velocidad para distribuir las páginas de un archivo de datos con una estructura de archivos basada en el tamaño del archivo de datos. Aunque las formas de realización van dirigidos a dispositivos de memoria Flash, las formas de realización de la invención son aplicables a otros dispositivos de memoria en los cuales se programen o escriban páginas de archivos de datos en por lo menos dos dispositivos de memoria.

En la descripción previa, a efectos explicativos, se exponen numerosos detalles para proporcionar una comprensión minuciosa de las formas de realización de la invención. No obstante, resultará evidente para aquellos expertos en la materia que estos detalles específicos no son necesarios para llevar a la práctica la invención. En otros casos,
45 estructuras y circuitos eléctricos bien conocidos se muestran en forma de diagramas de bloques con el fin de no complicar la invención. Por ejemplo, no se proporcionan detalles específicos en relación con si las formas de realización de la invención descritas en la presente se implementan como una rutina de software, un circuito de hardware, microprogramas o una combinación de los mismos.

Formas de realización de la invención se pueden representar como un producto de software almacenado en un soporte legible por máquina (al que se hace referencia también como soporte legible por ordenador, soporte legible por procesador, o soporte utilizable en ordenador que tiene un código de programa legible por ordenador materializado en el mismo). El soporte legible por máquina puede ser cualquier soporte tangible adecuado,
50 incluyendo un soporte de almacenamiento magnético, óptico, o eléctrico, incluyendo un disquete, un disco compacto-memoria de solo lectura (CD-ROM), un dispositivo de memoria (volátil o no volátil), o mecanismo de almacenamiento similar. El soporte legible por máquina puede contener varios conjuntos de instrucciones, secuencias de código, información de configuración, u otros datos, que, cuando se ejecutan, consiguen que un procesador lleve a cabo etapas de un método según una forma de realización de la invención. Aquellos con conocimientos habituales en la materia apreciarán que, en el soporte legible por máquina, también se pueden
55 almacenar otras instrucciones y operaciones necesarias para implementar la invención descrita. El software que se ejecuta desde el soporte legible por máquina se puede comunicar por interfaz con circuitería para llevar a cabo las tareas descritas.

Las formas de realización de la invención antes descritas están destinadas únicamente a ser ejemplos. Aquellos
65 expertos en la materia pueden llevar a cabo alteraciones, modificaciones y variaciones en las formas de realización

particulares, sin apartarse con respecto al alcance de la invención, el cual queda definido meramente por las reivindicaciones adjuntas a la misma.

REIVINDICACIONES

1. Método para programación en un sistema de memoria Flash que tiene una pluralidad de dispositivos de memoria Flash, comprendiendo el método:

5 recibir (402) un archivo de datos que tiene una pluralidad de páginas;
si la totalidad de la pluralidad de páginas del archivo de datos puede caber en un único bloque de uno de entre la pluralidad de dispositivos de memoria Flash, programar la totalidad de la pluralidad de páginas del archivo de datos en el bloque único de uno de entre la pluralidad de dispositivos de memoria Flash;

10 si no, si la totalidad de la pluralidad de páginas del archivo de datos puede caber en una pluralidad de bloques incluyendo un bloque en cada uno de entre la pluralidad de dispositivos de memoria, programar con intercalación (410) la totalidad de la pluralidad de páginas del archivo de datos por la pluralidad de bloques incluyendo un bloque en cada uno de entre la pluralidad de dispositivos de memoria; y

15 si no, programar con intercalación un subconjunto de la pluralidad de páginas del archivo de datos en una pluralidad de bloques incluyendo un bloque en cada uno de entre la pluralidad de dispositivos de memoria y eliminar (414) del archivo de datos el subconjunto de la pluralidad de páginas del archivo de datos.

20 2. Método según la reivindicación 1, que comprende además repetir la programación con intercalación del subconjunto de la pluralidad de páginas del archivo de datos en la pluralidad de bloques incluyendo un bloque en cada uno de entre la pluralidad de dispositivos de memoria y eliminar del archivo de datos el subconjunto de la pluralidad de páginas del archivo de datos.

25 3. Método según la reivindicación 1, que comprende además seleccionar el bloque único de uno de entre la pluralidad de dispositivos de memoria Flash según el número más alto de ciclos restantes de programación/borrado que quedan para el dispositivo de memoria.

30 4. Método según la reivindicación 1, que comprende además seleccionar el bloque único de uno de entre la pluralidad de dispositivos de memoria Flash según el último dispositivo de memoria programado.

35 5. Método según la reivindicación 1, en el que la programación con intercalación comprende proporcionar secuencialmente órdenes de programación a cada uno de entre la pluralidad de dispositivos de memoria.

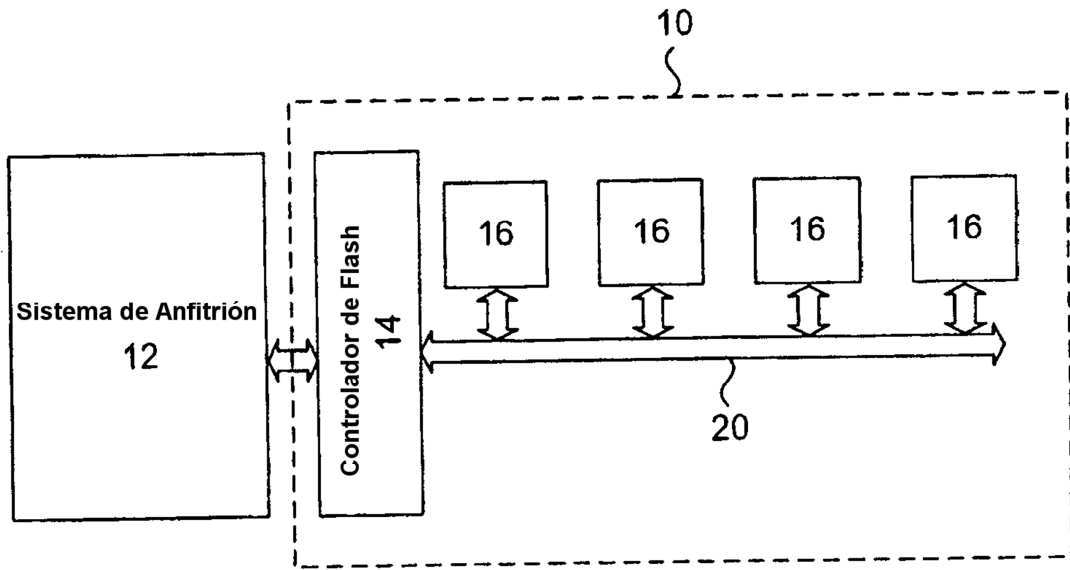


Figura 1 (Técnica anterior)

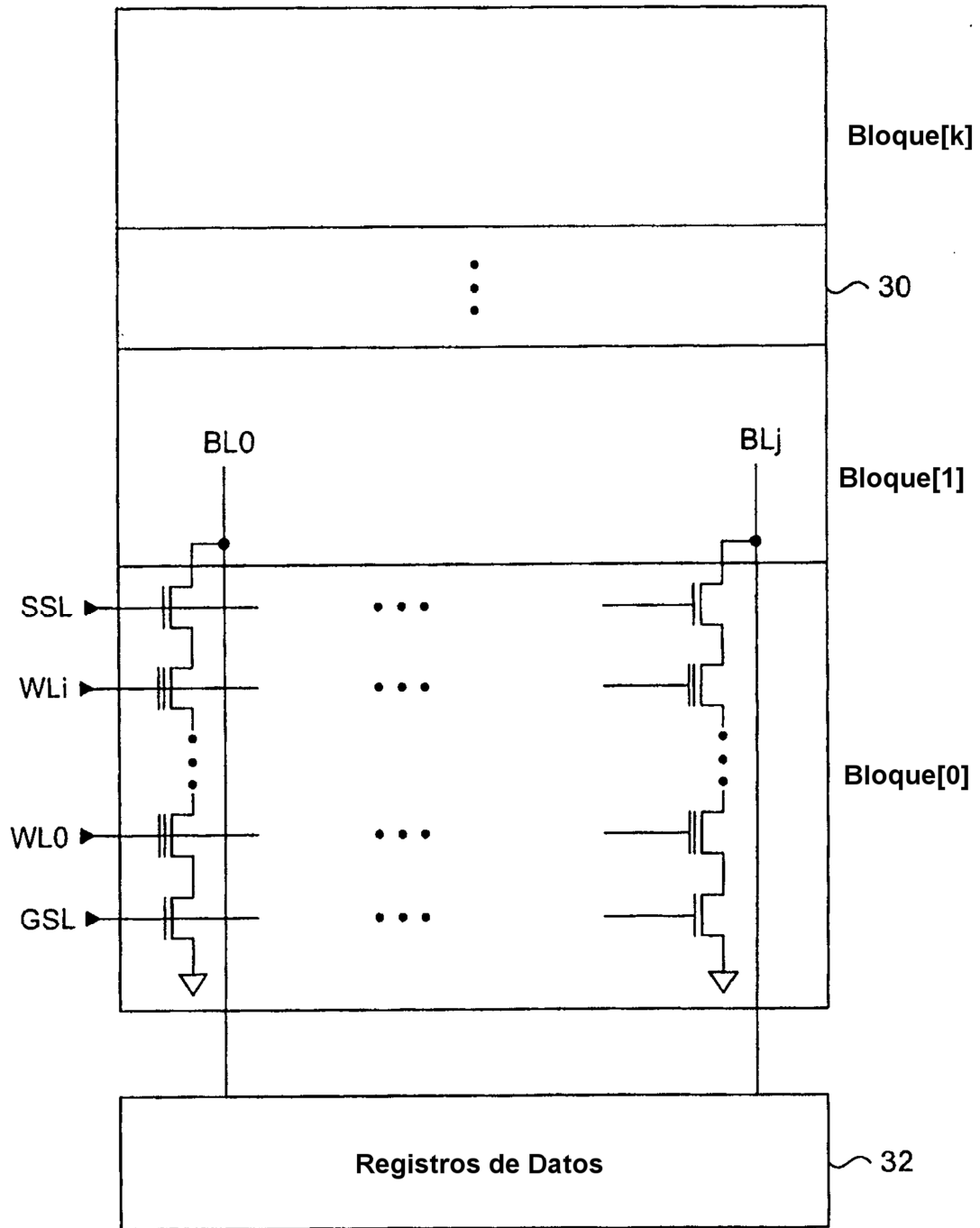


Figura 2 (Técnica anterior)

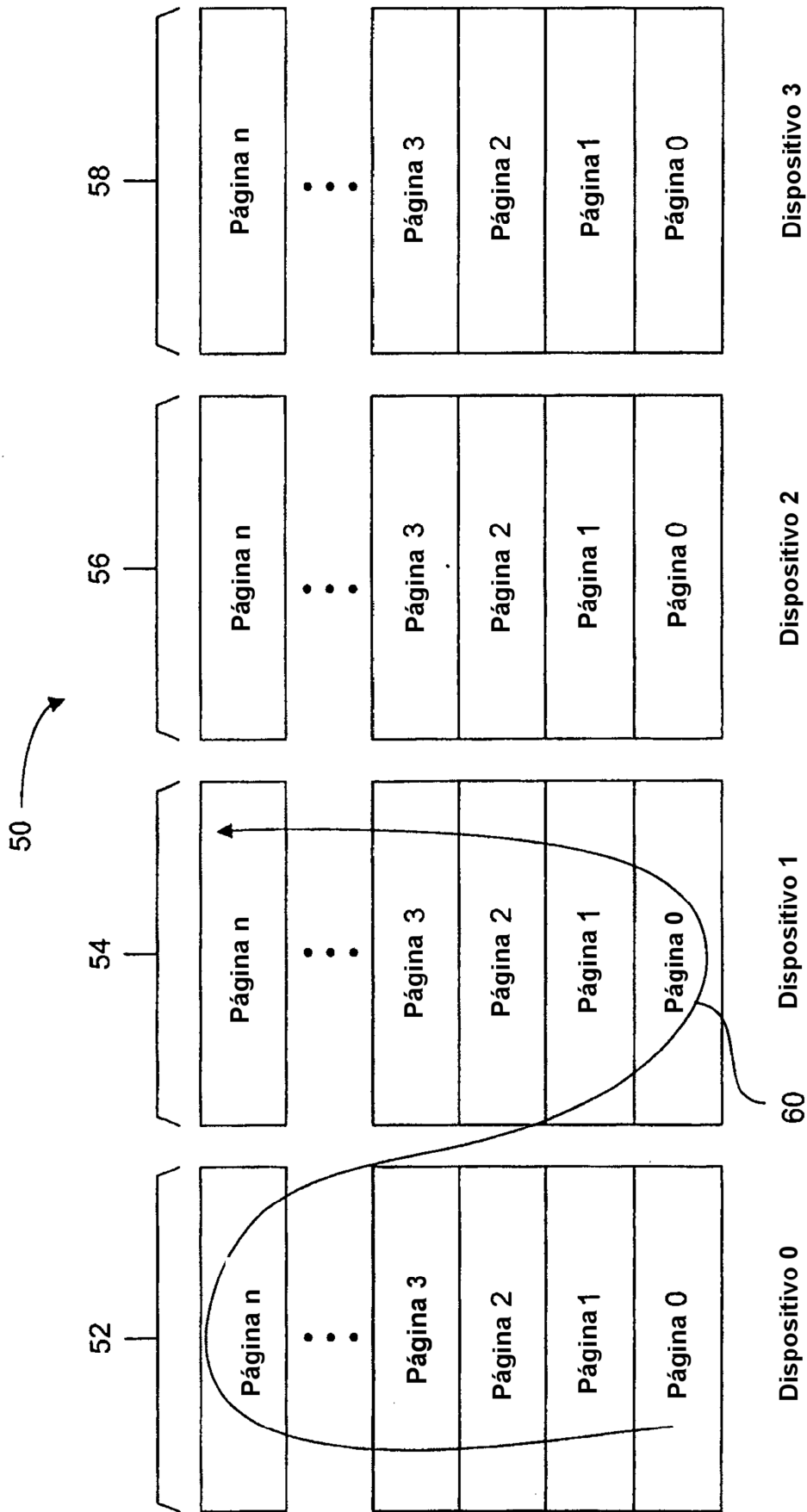


Figura 3 (Técnica anterior)

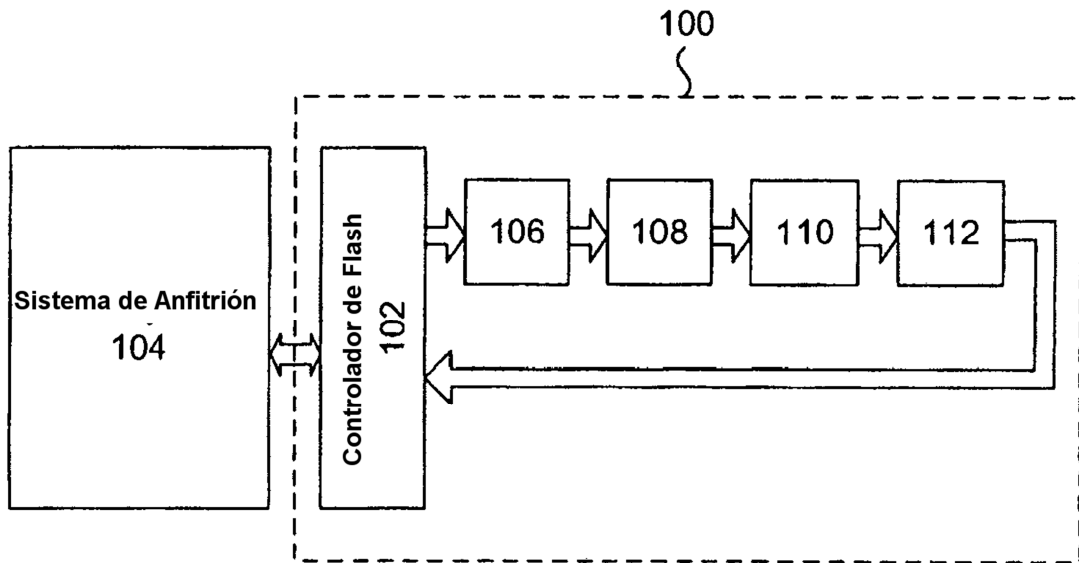


Figura 4

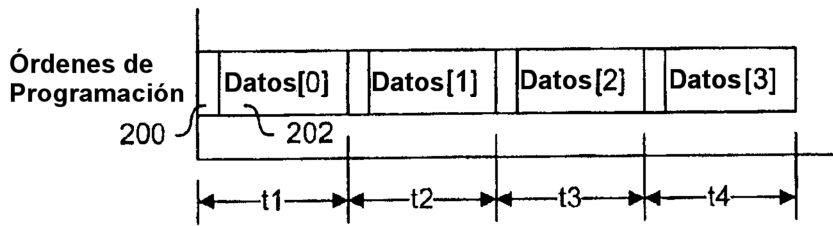


Figura 5

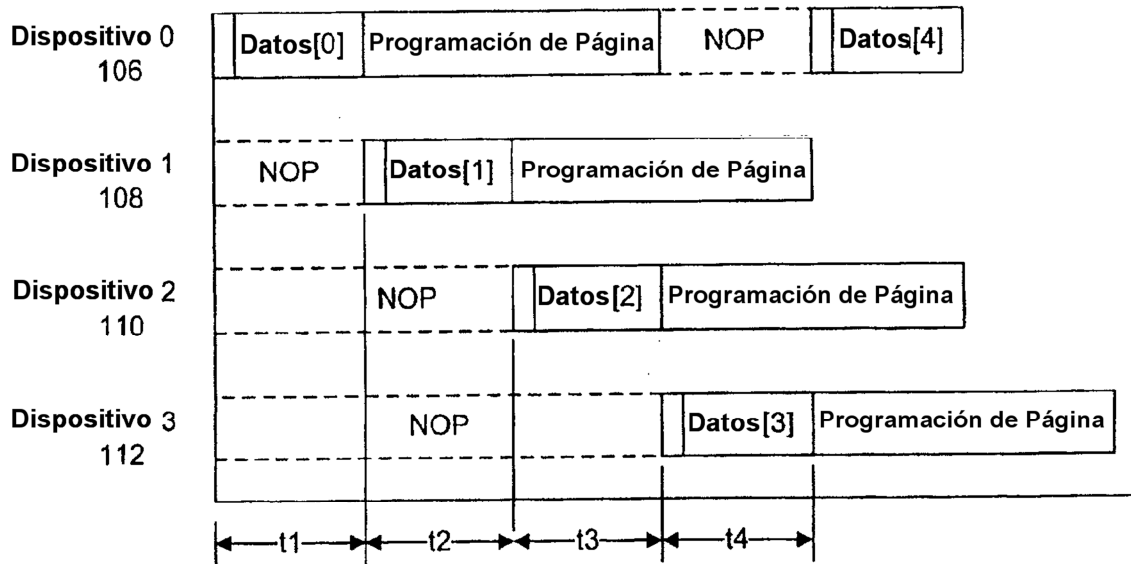


Figura 6

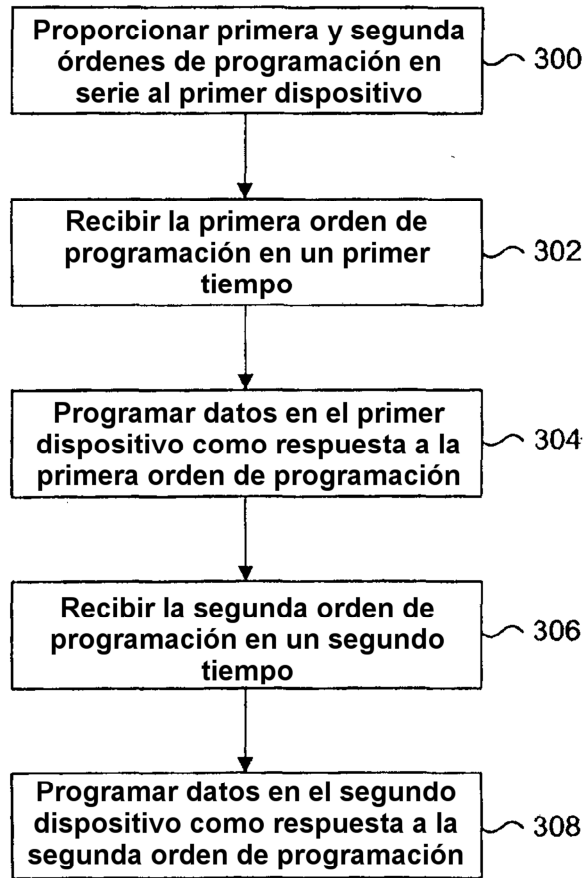


Figura 7

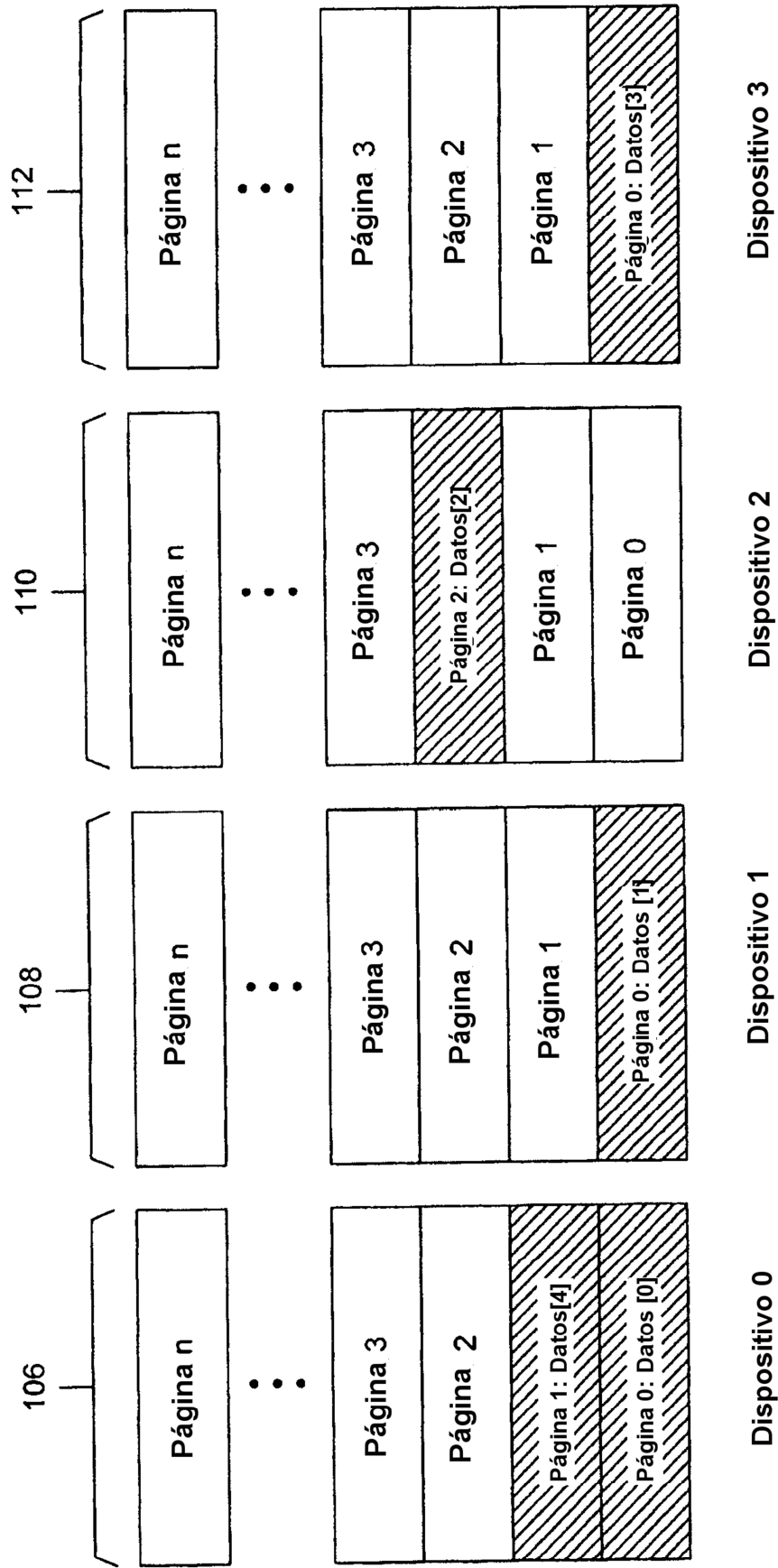


Figura 8

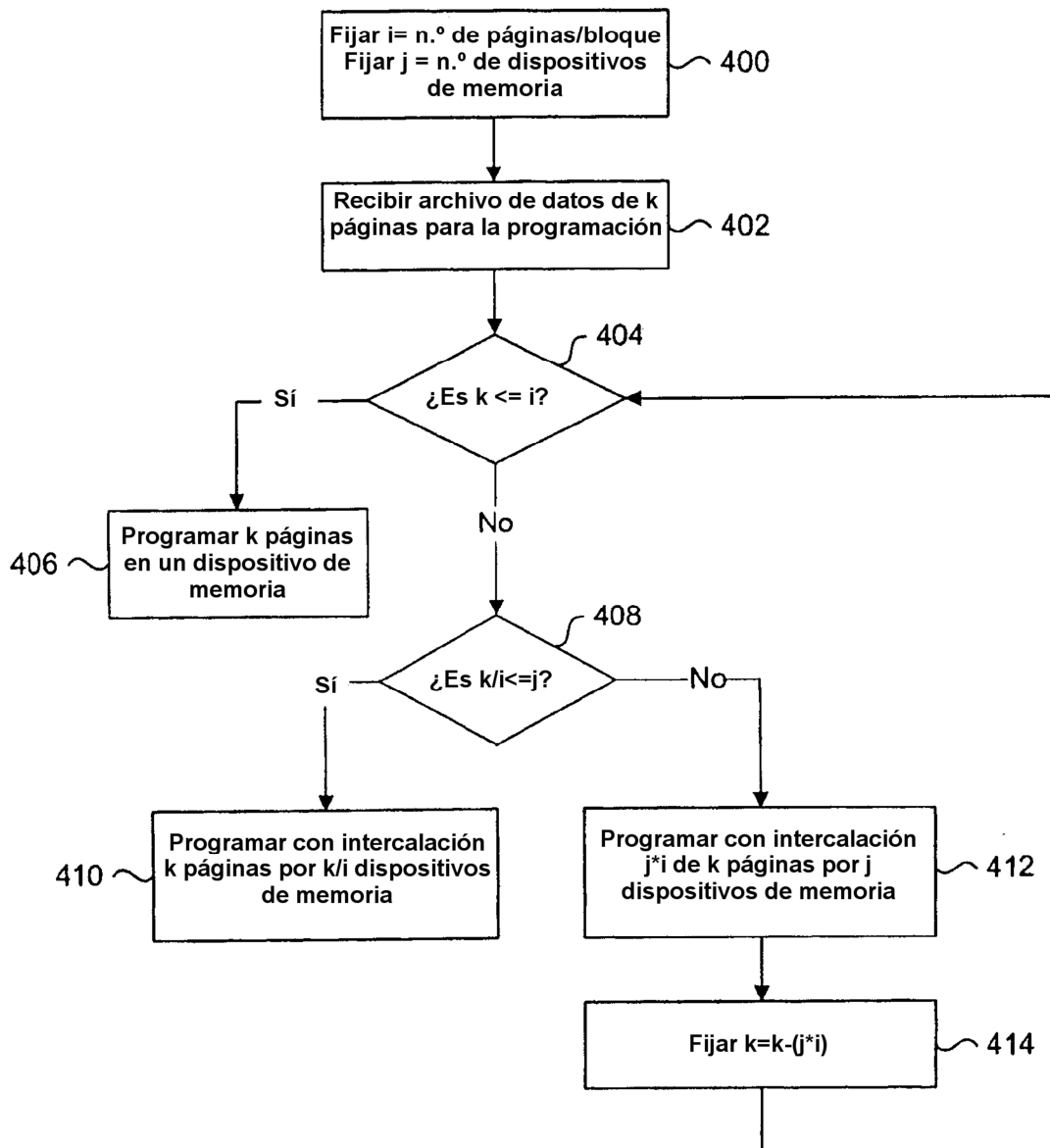


Figura 9

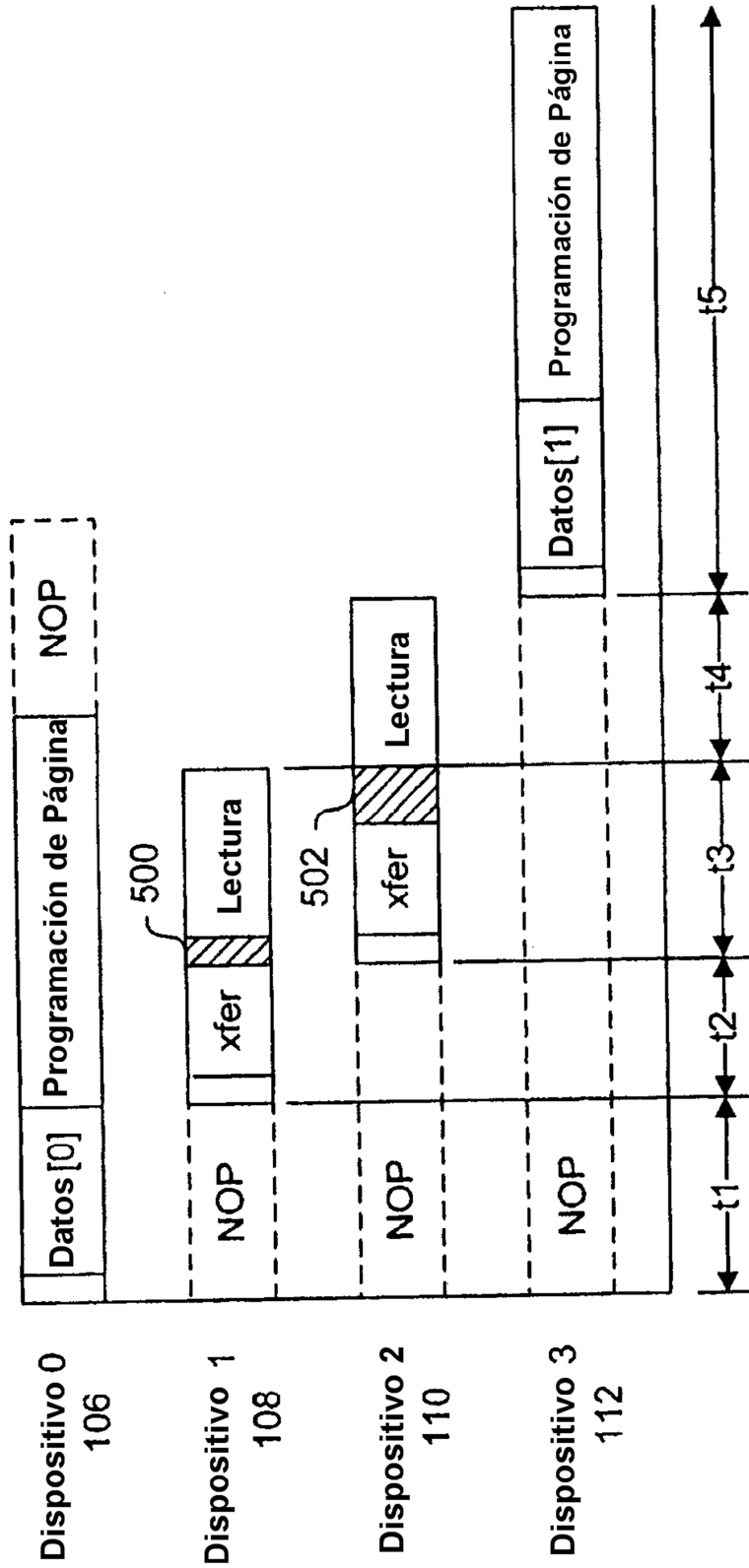


Figura 10