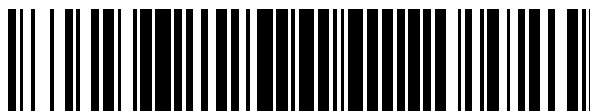


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 777 777**

51 Int. Cl.:

**G06F 13/42** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **14.06.2010 E 17153209 (6)**

97 Fecha y número de publicación de la concesión europea: **25.12.2019 EP 3193259**

54 Título: **Codificación de datos mediante la combinación de máscara de datos y la inversión del bus de datos**

30 Prioridad:

**13.07.2009 US 225102 P**

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

**06.08.2020**

73 Titular/es:

**RAMBUS INC. (100.0%)  
4453 N First Street, Suite 100  
San Jose, CA 95134, US**

72 Inventor/es:

**ABBASFAR, ALIAZAM**

74 Agente/Representante:

**ROEB DÍAZ-ÁLVAREZ, María**

**ES 2 777 777 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**

Codificación de datos mediante la combinación de máscara de datos y la inversión del bus de datos

5 La presente descripción se refiere a datos de codificación para transferir en un enlace de comunicación chip a chip.

Los dispositivos de circuito integrado se comunican entre sí mediante circuitos de entrada/salida (E/S) que están configurados para transmitir y recibir datos de varios bits a través de una pluralidad de líneas de señal. Cuando varios controladores de salida en los circuitos cambian de estado al mismo tiempo a alta velocidad para transmitir los datos de varios bits, la corriente cambiante que los controladores de salida extraen de una fuente de alimentación induce un voltaje que produce una perturbación de la fuente de alimentación, comúnmente conocida como ruido de salida de conmutación simultánea ("SSO") (o ruido de conmutación simultánea ("SSN")). El SSN puede causar un comportamiento transitorio no deseado entre los controladores de salida, los receptores de entrada o la lógica interna en los circuitos.

15 El documento US 2003/0158981 A1 describe un procedimiento y un sistema para transferir datos leídos desde un dispositivo de memoria que tiene un bus de datos y un pin de enmascaramiento de datos adaptado para recibir una señal de enmascaramiento durante las operaciones de escritura del dispositivo de memoria. El procedimiento incluye colocar una secuencia de palabras de datos leídos en el bus de datos y aplicar una señal de inversión del bus de datos en el pin de enmascaramiento de datos, indicando la señal de inversión del bus de datos si los datos contenidos en cada palabra de datos leídos se han invertido.

Resumen de la invención

25 La invención es como se define en las reivindicaciones independientes 1, 3, 5 y 7. Otros aspectos de la invención son como se establecen en las reivindicaciones dependientes.

Breve descripción de los dibujos

30 La figura 1 ilustra un sistema que incluye un codificador y un decodificador capaces de codificar y decodificar datos, según una realización. La figura 2 es un diagrama de flujo que ilustra un procedimiento de codificación de datos, según una realización. La figura 3 ilustra un ejemplo de un circuito codificador, según una realización. La figura 4 es un diagrama de flujo que ilustra un procedimiento de decodificación de datos, según una realización. La figura 5 ilustra un ejemplo de un circuito decodificador, según una realización. La figura 6A ilustra los pesos de Hamming (HW) presentes en los datos de 9 bits, incluidos los datos no codificados de 8 bits y un bit DBI. La figura 6B ilustra los pesos de Hamming presentes en los datos codificados de 9 bits incluidos un bit DBI y los datos codificados de 8 bits codificados según una realización.

40 Descripción detallada de las realizaciones

Las técnicas utilizadas para reducir el SSN en enlaces en paralelo incluyen la inversión del bus de datos (DBI), también conocida como inversión dinámica del bus. La DBI invierte algunos de los datos de varios bits que se transferirán en función de los bits anteriores transmitidos a través del enlace de comunicación en paralelo, por ejemplo, para reducir el SSN al disminuir el número de transiciones de conmutación del transmisor que se producen a través del enlace.

Un circuito usa una señal de máscara de datos (dm) para indicar a otro circuito que los datos enviados por un circuito deben ser ignorados por el otro circuito. Por ejemplo, la señal de máscara de datos puede ser utilizada por un controlador de memoria para indicar a un dispositivo de memoria que los datos transmitidos por el controlador de memoria al dispositivo de memoria deben ignorarse. La señal de máscara de datos puede ser un nivel de voltaje en un pin del circuito integrado del controlador de memoria o parte de un comando transmitido entre el controlador de memoria y el dispositivo de memoria.

Las realizaciones de la presente descripción incluyen un esquema de codificación de datos para la transmisión de datos de un circuito a otro circuito que combina codificación DBI y codificación no DBI y usa una señal de máscara de datos para indicar el tipo de codificación utilizada. En este caso, la expresión "codificación no DBI" se utiliza para referirse a cualquier esquema de codificación que sea diferente de DBI. La señal de máscara de datos en un primer estado indica que los datos transmitidos de un circuito a dicho otro circuito deben ser ignorados, y la señal de máscara de datos en un segundo estado indica que los datos transmitidos de un circuito a dicho otro circuito no deben ser ignorados. Si la señal de máscara de datos está en el segundo estado, un primer subconjunto de los datos se codifica con la inversión del bus de datos y un segundo subconjunto de datos se codifica de manera diferente a la inversión del bus de datos. El esquema de codificación según las realizaciones descritas en esta invención tiene la ventaja de

que el ruido SSO puede reducirse drásticamente cuando los datos codificados se transmiten de un circuito a otro circuito.

Ahora se hará referencia a varias realizaciones de la presente descripción, cuyos ejemplos se ilustran en las figuras adjuntas. Se observa que, siempre que sea posible, pueden usarse números de referencia parecidos o similares en las figuras y pueden indicar una funcionalidad parecida o similar. Las figuras representan realizaciones de la presente descripción solo con fines ilustrativos. Un experto en la materia reconocerá fácilmente a partir de la siguiente descripción que se pueden emplear realizaciones alternativas de las estructuras y procedimientos ilustrados en esta invención sin apartarse de los principios de la descripción descritos en esta invención.

La figura 1 ilustra un sistema que incluye un codificador y un decodificador capaces de codificar y decodificar datos, según una realización. El sistema de la figura 1 incluye dos circuitos que se comunican entre sí en un enlace de comunicación en paralelo 114. Por ejemplo, el circuito 102 puede ser un controlador de memoria, y el circuito 104 puede ser un dispositivo de memoria tal como una DRAM o SRAM. Sin embargo, los circuitos 102, 104 pueden ser otros tipos de circuitos que comunican datos entre sí.

El enlace de comunicación en paralelo 114 puede ser un bus en una placa de circuito en la que residen los circuitos 102, 104, y tiene un ancho de datos ( $n + 2$  bits) mayor que el número de bits ( $n$  bits) de datos  $D_n$  a transmitir desde el controlador de memoria 102 al dispositivo de memoria 104. En este caso,  $n$  es un número entero positivo mayor que uno ( $n > 1$ ). Los datos  $D_n$  de  $n$  bits tienen diferentes pesos de Hamming (HW) dependiendo del número de nivel lógico "1" en los datos  $D_n$  de  $n$  bits. En este caso, el peso de Hamming se refiere al número de bits distintos de cero (es decir, nivel lógico "1") en los datos  $D_n$  de  $n$  bits. Aunque el enlace 114 se muestra en esta invención como un enlace de comunicación en paralelo de un solo extremo, observe que el enlace 114 también puede ser un enlace configurado para transmitir señales diferenciales o señales de múltiples hilos.

El controlador de memoria 102 incluye un codificador 106 y un transmisor (Tx) 108. Los datos  $D_n$  pueden ser proporcionados al codificador 106 como, por ejemplo, datos en serie o en paralelo. El dispositivo de memoria 104 también incluye el decodificador 112 y un receptor (Rx) 110. Además, el dispositivo de memoria 104 también puede incluir otro codificador y un transmisor para transmitir datos codificados al controlador de memoria 102, y el controlador de memoria 102 también puede incluir un receptor y un decodificador para recibir y decodificar los datos codificados recibidos del dispositivo de memoria 104. Sin embargo, el codificador y transmisor del dispositivo de memoria 104 y el receptor y decodificador del controlador de memoria 102 no se muestran en la figura 1 ya que no son necesarios para explicar las realizaciones de la presente descripción.

El codificador 106 recibe los datos  $D_n$  de  $n$  bits y una señal de máscara de datos de 1 bit ( $dm$ ) de otros circuitos (no mostrados) u otros componentes del circuito (por ejemplo, máquinas de estado, no mostradas) dentro del controlador de memoria 102. Como se explicó anteriormente, el controlador de memoria 102 utiliza la señal de máscara de datos ( $dm$ ) para indicar al dispositivo de memoria 104 que ignore los datos recibidos desde el controlador de memoria 102 (es decir, para indicar la operación sin escritura). La señal  $dm$  puede ser un nivel de voltaje en un pin, que está desplazado o acompaña a los datos que se envían desde el controlador de memoria 102 al dispositivo de memoria. Como alternativa, la información de enmascaramiento transmitida por la señal  $dm$  puede incluirse como parte de un comando transmitido entre el controlador de memoria 102 y el dispositivo de memoria 104, y por lo tanto no es necesario que la señal de máscara de datos físicos  $dm$  esté presente en el enlace en paralelo 114. El codificador 106 incluye puertas lógicas combinatorias o tablas de consulta (LUT) que están configuradas para realizar la codificación de los datos  $D_n$  de  $n$  bits que recibe para convertir los datos de  $n$  bits a datos codificados  $DQ_n$  de  $n$  bits y un bit DBI (1 bit). El codificador 106 también convierte la señal de máscara de datos ( $dm$ ) en una señal de máscara de datos modificada (DM). Como se explicará con más detalle a continuación con referencia a las figuras 2 y 3, el codificador 106 utiliza la codificación DBI en algunos casos de los datos  $D_n$  de  $n$  bits y la codificación no DBI en otros casos de los datos  $D_n$  de  $n$  bits para codificar los datos  $D_n$  de  $n$  bits, dependiendo de los pesos de Hamming de los datos  $D_n$  de  $n$  bits. El codificador 106 también genera la señal de máscara de datos modificada (DM) para indicar al dispositivo de memoria 104 el tipo de codificación utilizada en los datos codificados  $DQ_n$  de  $n$  bits en esos casos diferentes.

Además, el codificador 106 también recibe una señal de modo 116. En una realización, cuando se afirma la señal de modo 116 (por ejemplo, lógica alta), el codificador 106 codifica los datos  $D_n$  de  $n$  bits utilizando el esquema de codificación según las realizaciones de la presente descripción como se describe en esta invención. Cuando no se afirma la señal de modo 116 (por ejemplo, lógica baja), el codificador 106 usa codificación DBI (que es independiente del uso de información de enmascaramiento) para codificar los datos  $D_n$  de  $n$  bits.

En una realización, el transmisor (Tx) 108 transmite los datos codificados  $DQ_n$ , el bit DBI y el bit DM a través del enlace en paralelo 114 al receptor (Rx) 110 del dispositivo de memoria 104. Al hacerlo, el transmisor (Tx) 108 extrae corriente de suministro  $I_{dd}$  de una fuente de alimentación (no mostrada) que proporciona el voltaje de alimentación ( $V_{dd}$ ), consume energía de la fuente de alimentación y genera ruido SSO. Sin embargo, el ruido SSO generado por el

transmisor (Tx) 108 puede reducirse según las realizaciones de la presente descripción.

En una realización, el receptor 110 recibe los datos codificados  $DQ_n$ , el bit DBI y el bit DM a través del enlace en paralelo 114 del controlador de memoria 102. Al hacerlo, el receptor 110 también extrae corriente de suministro Idd de una fuente de alimentación (no mostrada) que proporciona el voltaje de alimentación (Vdd), consume energía de la fuente de alimentación y genera ruido SSO. Nuevamente, el ruido SSO generado por el receptor (Rx) 110 puede reducirse según las realizaciones de la presente descripción.

El receptor 110 proporciona los datos codificados recibidos  $DQ_n$ , el bit DBI y el bit DM al decodificador 112. El decodificador 112 decodifica los datos codificados  $DQ_n$  y el bit DBI de nuevo a datos  $D_n$  de n bits y convierte la señal de máscara de datos modificada (DM) de nuevo a la señal de máscara de datos (dm), como se explicará con más detalle a continuación con referencia a las figuras 4 y 5.

Además, el decodificador 112 también recibe una señal de modo 516. En una realización, cuando se afirma la señal de modo 516 (por ejemplo, lógica alta), el decodificador 112 decodifica los datos codificados  $DQ_n$  de n bits usando el esquema de decodificación según las realizaciones de la presente descripción como se describe en esta invención. Cuando no se afirma la señal de modo 516 (es decir, lógica baja), la lógica del decodificador 112 utiliza la decodificación DBI (que es independiente del uso de información de enmascaramiento) para decodificar los datos codificados  $DQ_n$  de n bits.

La figura 2 es un diagrama de flujo que ilustra un procedimiento de codificación de datos, según una realización. Según el esquema de codificación de la realización de la figura 2, se usa una combinación de codificación DBI y codificación no DBI para codificar los datos  $D_n$  de n bits dependiendo del peso de Hamming de los datos  $D_n$  de n bits, es decir, el número de bits distintos de cero bits (es decir, nivel lógico "1") en los datos  $D_n$  de n bits.

Primero, se determina si se afirma la señal de máscara de datos (dm) (en este ejemplo "1" o lógica alta). Si se afirma la señal de máscara de datos (dm), los datos  $D_n$  de n bits se codifican de manera que los datos codificados [DBI,  $DQ_n$ ] de n + 1 bits son un patrón de datos predeterminado (DQM) de (n + 1) bits indicativo de una señal de máscara al dispositivo de memoria 104. Además, la señal de máscara de datos modificada (DM) se establece en "1" o lógica alta. Por otro lado, si la señal de máscara de datos (dm) no se afirma ("0" o lógica baja), se determina además si los datos  $D_n$  de n bits, cuando se codifica DBI, tienen un peso de Hamming inferior a un umbral predeterminado. En otras palabras, en la etapa 206 se determina además si los datos  $D_n$  de n bits no codificados tiene un peso de Hamming inferior a un primer umbral predeterminado o mayor que un segundo umbral predeterminado. Por ejemplo, cuando  $D_n$  son datos de 8 bits, se puede usar un umbral de peso de Hamming 3 de modo que los datos  $D_n$  de 8 bits se consideren datos de bajo peso de Hamming si los datos  $D_n$  de 8 bits, cuando se codifica DBI, tienen un peso de Hamming inferior a 4 (es decir, HW de  $D_n$  codificado DBI es 0, 1, 2 o 3), es decir, cuando los datos  $D_n$  de 8 bits no codificados tienen un peso de Hamming inferior a 4 (es decir, HW de  $D_n$  no codificado es 0, 1, 2 o 3) o mayor que 5 (es decir, HW de  $D_n$  no codificado es 6, 7 u 8).

Si los datos  $D_n$  de n bits codificados DBI no son datos de bajo peso de Hamming en la etapa 206, entonces los datos  $D_n$  de n bits se codifican de tal manera que los datos [DBI,  $DQ_n$ ] de n + 1 bits codificados son codificados DBI. Por ejemplo, cuando los datos  $D_n$  de n bits son datos de 8 bits y tienen un peso de Hamming de 4, no se produce inversión de los bits de datos y no se afirma el bit DBI (es decir, lógica baja). Por otro lado, si los datos  $D_n$  de 8 bits tienen un peso de Hamming de 5, se produce inversión de los bits de datos y se afirma el bit DBI (es decir, lógica alta). Por tanto, para datos de entrada  $D_n$  de 8 bits, los datos codificados [DBI,  $DQ_n$ ] de n + 1 bits con codificación DBI en la etapa 210 tendrán un peso de Hamming de solo 4. Además, no se afirma la señal de máscara de datos modificada (DM) ("0" o lógica baja) en la etapa 210.

Por otro lado, si los datos  $D_n$  de n bits codificados DBI son datos de bajo peso de Hamming en la etapa 206, entonces los datos  $D_n$  de n bits se codifican utilizando un esquema de codificación no DBI. En algunas realizaciones, todos o la mayoría de los datos codificados [DBI,  $DQ_n$ ] de n + 1 bits están codificados para ser datos que no son de bajo peso de Hamming. Por ejemplo, cuando los datos  $D_n$  de n bits son datos de 8 bits, los datos  $D_n$  de 8 bits están codificados con un esquema de codificación no DBI que da como resultado la mayor parte (252) de los datos codificados [DBI,  $DQ_n$ ] de 9 bits con pesos de Hamming de cuatro y simplemente algunos (4) de los datos codificados de 9 bits [DBI,  $DQ_n$ ] con pesos de Hamming de tres. Además, la señal de máscara de datos modificada (DM) se afirma ("1" o lógica alta) incluso cuando la señal de máscara de datos original (dm) no se afirma para indicar al dispositivo de memoria 104 que se utilizó un esquema de codificación no DBI para codificar los datos codificados [DBI,  $DQ_n$ ] de n + 1 bits. Por tanto, la señal de máscara de datos modificada (DM) se usa para indicar al dispositivo de memoria 104 si se utilizó la codificación DBI o no DBI para codificar los datos codificados [DBI,  $DQ_n$ ] de n + 1 bits. El esquema de codificación de la realización de la figura 2 reduce significativamente el ruido SSO porque los pesos de Hamming de los datos codificados [DBI,  $DQ_n$ ] están limitados a un pequeño intervalo de variaciones (por ejemplo, pesos de Hamming de 3 o 4 en el ejemplo de datos  $D_n$  de 8 bits).

## ES 2 777 777 T3

La tabla 1 a continuación ilustra un ejemplo de lógica combinatoria que puede usarse para codificar datos de entrada  $D_n$  de 8 bits a datos codificados [DBI,  $DQ_n$ ] de 9 bits según la realización como se muestra en la figura 2. La lógica combinatoria que se muestra en la tabla 1 es meramente ejemplar, y se puede usar otra lógica combinatoria para realizar la codificación como se ilustra en la figura 2.  $\bar{x}$

Tabla 1

HW de datos $D_n$ de 8 bits	Datos $D_n$ sin codificar de 8 bits ( $x = x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$ )	Datos codificados [DBI, $DQ_n$ ] de 9 bits  ( $x$ es complemento de $x$ )	Notas y lógica ("+" es operación OR)	HW de datos codificados [DBI, $DQ_n$ ]
0	0000 0000	1 0101 0000		3
1	$x_7 x_6 x_5 x_4$ 0000	0 1000 $\bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4$		4
	0000 $x_3 x_2 x_1 x_0$	0 0100 $\bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$		4
2	00 $x_5 x_4 x_3 x_2 x_1 x_0$	0 11 $x_5 x_4 x_3 x_2 x_1 x_0$	$y_3 = x_5 + x_2 + x_0$	4
	01 $x_5 x_4 x_3 x_2 x_1 x_0$	0 1010 $y_3 y_2 y_1 y_0$	$y_2 = x_5 + x_4 + x_1$	4
	10 $x_5 x_4 x_3 x_2 x_1 x_0$	01001 $y_3 y_2 y_1 y_0$	$y_1 = x_4 + x_3 + x_0$	4
	11 $x_5 x_4 x_3 x_2 x_1 x_0$	0 1001 0001	$y_0 = x_3 + x_2 + x_1$	3
3	$x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$	1 $x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$		4
4	$x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$	0 $x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$	(DBI sin inversión)	4
5	$x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$	1 $\bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$	DBI (inversión)	4
6	11 $x_5 x_4 x_3 x_2 x_1 x_0$	0 00 $x_5 x_4 x_3 x_2 x_1 x_0$	$z_3 = \bar{x}_5 + \bar{x}_2 + \bar{x}_0$	4
	10 $x_5 x_4 x_3 x_2 x_1 x_0$	0 0101 $z_3 z_2 z_1 z_0$	$z_2 = \bar{x}_5 + \bar{x}_4 + \bar{x}_1$	4
	01 $x_5 x_4 x_3 x_2 x_1 x_0$	0 0110 $z_3 z_2 z_1 z_0$	$z_1 = \bar{x}_4 + \bar{x}_3 + \bar{x}_0$	4
	00 $x_5 x_4 x_3 x_2 x_1 x_0$	0 0110 0001	$z_0 = \bar{x}_3 + \bar{x}_2 + \bar{x}_1$	3
7	$x_7 x_6 x_5 x_4$ 1111	0 0111 $\bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4$		4
	1111 $x_3 x_2 x_1 x_0$	0 1011 $\bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$		4
8	1111 1111	1 1010 0000		3
	Máscara de datos (dm) = 1	DQM = 1 1100 0000		3

- 10 Como se puede ver de la tabla 1 anterior, cuando se afirma la señal de máscara de datos (dm), los datos codificados [DBI,  $DQ_n$ ] son DQM (111000000). Por otro lado, cuando no se afirma la señal de máscara de datos (dm), entonces los datos  $D_n$  con pesos de Hamming de 4 o 5 se codifican usando DBI de modo que los bits de datos de  $D_n$  con un peso de Hamming de 4 no están invertidos y el bit DBI se establece en 0 mientras que los bits de datos  $D_n$  con un peso de Hamming de 5 están invertidos y el bit DBI se establece en 1. Los datos codificados DBI [DBI,  $DQ_n$ ] tienen un
- 15 peso de Hamming de 4 solamente. Además, cuando la señal de máscara de datos (dm) no se afirma, entonces los datos  $D_n$  con pesos de Hamming de 0, 1, 2 o 3 (inferior a 4) o 6, 7 u 8 (mayor que 5) se codifican utilizando un esquema de codificación no DBI como se muestra anteriormente en la tabla 1 que limita los pesos de Hamming de los datos codificados [DBI,  $DQ_n$ ] a pesos de Hamming de 3 o 4.
- 20 Más específicamente, como se muestra en la tabla 1, el patrón de datos de 8 bits en forma de 000000000 con HW de cero se codifica en el patrón de datos codificados de 9 bits 101010000 con HW de tres. Los datos de 8 bits en forma de  $x_7 x_6 x_5 x_4$  0000 con HW de uno se codifican en los datos de 9 bits en forma de 01000  $\bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4$  con HW de cuatro y los patrones de datos de 8 bits en forma de 0000  $x_3 x_2 x_1 x_0$  con HW de uno se codifican en los patrones de datos codificados de 9 bits en forma de 00100  $\bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$  con HW de cuatro, donde  $x_n$  es el enésimo bit de un bit menos
- 25 significativo del patrón de datos de 8 bits y  $\bar{x}_n$  es el complementos de  $x_n$ . Los patrones de datos de 8 bits en forma de 00  $x_5 x_4 x_3 x_2 x_1 x_0$  con HW de dos se codifican en los patrones de datos codificados de 9 bits en forma de 011  $x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cuatro, los patrones de datos de 8 bits en forma de 01  $x_5 x_4 x_3 x_2 x_1 x_0$  con HW de dos se codifican en los patrones de datos codificados de 9 bits en forma de 01010  $y_3 y_2 y_1 y_0$  con HW de cuatro, los patrones de datos de 8 bits en forma de 10  $x_5 x_4 x_3 x_2 x_1 x_0$  con HW de dos se codifican en los patrones de datos codificados
- 30 de 9 bits en forma de 01001  $y_3 y_2 y_1 y_0$  con HW de cuatro, y los patrones de datos de 8 bits en forma de 11  $x_5 x_4 x_3 x_2 x_1 x_0$  con HW de dos se codifican en un patrón de datos codificados de 9 bits 010010001 con HW de tres, donde  $\bar{x}_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits,  $y_3 = x_5 + x_2 + x_0$ ,  $y_2 = x_5 + x_4 + x_1$ ,  $y_1 = x_4 + x_3 + x_0$  e  $y_0 = x_3 + x_2 + x_1$  y donde + es la operación OR.
- 35 Para otro ejemplo, los patrones de datos de 8 bits en forma de  $x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de tres se codifican en

los patrones de datos codificados de 9 bits en forma de  $1 x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cuatro, donde  $x_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits. Los patrones de datos de 8 bits en forma de  $x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cuatro se codifican DBI en los patrones de datos codificados de 9 bits  $0 x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cuatro, donde  $x_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits. Los patrones de datos de 8 bits en forma de  $x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cinco también se codifican DBI en los patrones de datos codificados de 9 bits en forma de  $1 \bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$  con HW de cuatro, donde  $x_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits y  $\bar{x}_n$  es el complemento (inverso) de  $x$ .

Para otro ejemplo más, los patrones de datos de 8 bits en forma de  $11 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de seis se codifican en los patrones de datos codificados de 9 bits en forma de  $000 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de cuatro, los patrones de datos de 8 bits en forma de  $10 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de seis se codifican en los patrones de datos codificados de 9 bits  $00101 z_3 z_2 z_1 z_0$  con HW de cuatro, los patrones de datos de 8 bits en forma de  $01 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de seis se codifican en los patrones de datos codificados de 9 bits  $00110 z_3 z_2 z_1 z_0$  con HW de cuatro, y los patrones de datos de 8 bits en forma de  $00 x_5 x_4 x_3 x_2 x_1 x_0$  con HW de seis se codifican en los patrones de datos codificados de 9 bits  $001100001$  con HW de tres, donde  $x_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits,  $z_3 = \bar{x}_5 + \bar{x}_2 + \bar{x}_0$ ,  $z_2 = \bar{x}_5 + \bar{x}_4 + \bar{x}_1$ ,  $z_1 = \bar{x}_5 + \bar{x}_4 + \bar{x}_1$  y  $z_0 = \bar{x}_3 + \bar{x}_2 + \bar{x}_1$ ,  $\bar{x}_n$  es el complemento de  $x_n$ , y "+" es la operación OR. Los patrones de datos de 8 bits en forma de  $x_7 x_6 x_5 x_4 1111$  con HW de siete se codifican en los patrones de datos codificados de 9 bits en forma de  $00111$  de  $\bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4$  con HW de cuatro, y los patrones de datos de 8 bits en forma de  $1111 x_3 x_2 x_1 x_0$  con HW de siete se codifican en los patrones de datos codificados de 9 bits en forma de  $01011 \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$  con HW de cuatro, donde  $x_n$  es el enésimo bit de un bit menos significativo del patrón de datos de 8 bits, y  $\bar{x}_n$  es el complemento de  $x_n$ . Para otro ejemplo más, el patrón de datos de 8 bits  $11111111$  con HW de ocho se codifica en el patrón de datos codificados de 9 bits  $110100000$  con HW de tres. Además, el patrón de datos DQM se establece en  $111000000$ .

Aunque se supone que los datos  $D_n$  de  $n$  bits que se transmitirán desde el controlador de memoria 102 al dispositivo de memoria 104 son datos de 8 bits en los ejemplos anteriores, el uso de datos de 8 bits en esta invención es meramente ejemplar. Por tanto, el esquema de codificación descrito en esta invención puede usarse con cualquier número de bits de datos a transmitir desde el controlador de memoria 102 al dispositivo de memoria 104. Las asignaciones entre datos no codificados de 8 bits y datos codificados de 9 bits como los mostrados en la tabla 1 anterior pueden determinarse y almacenarse en una LUT (no mostrada) en un controlador de memoria 102. De esa manera, la codificación de datos según las realizaciones descritas en esta invención puede realizarse rápidamente.

La figura 3 ilustra un ejemplo de un circuito codificador, según una realización. El circuito codificador de la figura 3 es un ejemplo de un circuito configurado para realizar el esquema de codificación como se describió anteriormente con referencia a la figura 2, aunque pueden estar disponibles diferentes configuraciones de circuito con las mismas funciones para realizar el esquema de codificación de la figura 2. El codificador 106 incluye un codificador DBI 302, un codificador no DBI 304, una puerta AND 312, una puerta OR 314 y multiplexores 308, 310. El codificador DBI 302 está configurado para realizar codificación DBI en los datos de entrada  $D_n$  como se explicó anteriormente con la etapa 210 de la figura 2 y el codificador no DBI 304 está configurado para llevar a cabo una codificación no DBI en los datos de entrada  $D_n$  como se explicó anteriormente con la etapa 208 de la figura 2, por ejemplo, como se muestra anteriormente en la tabla 1. La señal de modo 116 se usa para indicar al codificador 116 si se debe usar (cuando se afirma el modo 116) el esquema de codificación según las realizaciones de esta invención (figura 2) o se debe usar (cuando no se afirma el modo 116) la codificación DBI (que es independiente del uso de información de enmascaramiento). Para fines de ilustración en esta invención del esquema de codificación de la figura 2, se supondrá afirmada la señal de modo 116 (lógica alta).

Volviendo al funcionamiento del circuito codificador 106, cuando se afirma la señal de máscara de datos (dm) 312 ("1" o lógica alta), la salida de la señal de máscara de datos modificada DM 326 de la puerta OR 314 también es "1" o lógica alta. Además, la señal de máscara de datos (dm) 312 también hace que el multiplexor 310 seleccione el patrón de bits predeterminado DQM 323 como su salida 316.

Por otro lado, si la señal de máscara de datos (dm) 312 no se afirma ("0" o lógica baja), el multiplexor 310 selecciona la salida 322 del multiplexor 308 como sus datos codificados [DBI, DQ<sub>n</sub>]. La salida 322 del multiplexor 308 es la salida 318 del codificador DBI 302 o la salida 320 del codificador no DBI 304, seleccionada según la salida 324 de la puerta AND 312. A este respecto, el codificador DBI 302 incluye lógica (no mostrada) configurado para determinar si los datos de entrada  $D_n$ , cuando están codificados DBI por el codificador 302 de DBI, tienen un peso de Hamming inferior a un umbral predeterminado. En otras palabras, el codificador DBI 302 también está configurado para determinar si los datos  $D_n$  de  $n$  bits no codificados tienen un peso de Hamming inferior a un primer umbral predeterminado o mayor que un segundo umbral predeterminado. Cuando los datos de entrada  $D_n$ , cuando están codificados DBI por el codificador DBI 302, tienen un peso de Hamming inferior a un umbral predeterminado, el codificador DBI 302 genera la señal de

bajo HW 306 para ser lógica alta ("1"), pero genera de otra manera la señal de bajo HW 306 para ser lógica baja ("0").

Dado que la señal de modo 116 se fija para ser lógica alta, la salida 324 de la puerta AND 312 se convierte en lógica alta si los datos de entrada  $D_n$ , cuando están codificados DBI por el codificador DBI 302, tienen un peso de Hamming inferior a un umbral predeterminado y por tanto se afirma la señal de bajo HW 306. Como resultado, el multiplexor 308 selecciona la salida 320 del codificador no DBI 304 que como su salida 322. Además, la salida 326 de la puerta OR 314 también se vuelve lógica alta (señal de máscara de datos modificada  $DM = 1$ ) aunque no se afirme la señal de máscara de datos original  $dm$  312. Por tanto, la señal de máscara de datos modificada  $DM$  326 a una lógica alta puede indicar al dispositivo de memoria 104 que se ha usado una codificación no DBI para codificar los datos de entrada  $D_n$ .

Por otro lado, la salida 324 de la puerta AND 312 se vuelve lógica baja si los datos de entrada  $D_n$ , cuando están codificados DBI por el codificador DBI 302, tienen un peso de Hamming no inferior a un umbral predeterminado y, por tanto, no se afirma la señal de bajo HW 306. Como resultado, el multiplexor 308 selecciona la salida 318 del codificador DBI 302 como su salida 322. Además, la salida 326 de la puerta OR 314 también se vuelve lógica baja (señal de máscara de datos modificada  $DM = 0$ ) cuando no se afirma la señal de máscara de datos original ( $dm$ ) 312. Por tanto, la señal de máscara de datos modificada  $DM$  326 a lógica baja puede indicar al dispositivo de memoria 104 que la codificación DBI se ha utilizado para codificar los datos de entrada  $D_n$ .

Finalmente, cuando la señal de modo 116 está en lógica baja ("0"), entonces la salida 324 de la puerta AND 312 está siempre en lógica baja independientemente de la señal de bajo HW 306. Por tanto, la salida de la señal de máscara de datos modificada ( $DM$ ) 326 de la puerta OR 314 sigue la señal de máscara de datos original ( $dm$ ) 312, y el multiplexor 308 siempre selecciona la salida 318 del codificador DBI 302. Por tanto, cuando la señal de máscara de datos original ( $dm$ ) 312 no se afirma, el multiplexor 310 selecciona la salida 322 de multiplexor 308, que es siempre la salida 318 del codificador DBI, lo que resulta en codificación DBI independientemente del peso de Hamming de los datos de entrada  $D_n$ . Cuando se afirma la señal de máscara de datos original ( $dm$ ) 312, el multiplexor 310 selecciona la señal  $DQM$  para indicar una señal de máscara al dispositivo de memoria 104. Por tanto, cuando la señal de modo 116 está en lógica baja, el codificador 106 realiza la codificación DBI (que es independiente del uso de información de enmascaramiento), y por tanto el codificador 106 tiene una compatibilidad completamente retroactiva con los esquemas de codificación DBI convencionales. Por ejemplo, la señal de modo 116 puede almacenarse como un valor en un registro de configuración programable o cableada por medio de un fusible o máscara de metal, etc.

La figura 4 es un diagrama de flujo que ilustra un procedimiento de decodificación de datos, según una realización. El esquema de decodificación de la realización de la figura 4 se usa para decodificar los datos  $[DBI, DQ_n]$  codificados usando una combinación de codificación DBI y codificación no DBI según el esquema de codificación como se explicó anteriormente con referencia a la figura 2.

Primero, se determina 402 si se afirma la señal de máscara de datos modificada ( $DM$ ) ("1" o lógica alta). Si la señal de máscara de datos modificada ( $DM$ ) no se afirma, eso significa que los datos  $[DBI, DQ_n]$  estaban codificados DBI (etapa 210 de la figura 2). Por tanto, los datos  $[DBI, DQ_n]$  se decodifican utilizando DBI y la señal de máscara de datos ( $dm$ ) se establece en "0" para indicar que los datos decodificados  $D_n$  no deben ignorarse. Por otro lado, si la señal de máscara de datos modificada ( $DM$ ) no se afirma, entonces se determina además 406 si los datos codificados  $[DBI, DQ_n]$  coinciden con el patrón de datos predeterminado  $DQM$ .

Si los datos codificados  $[DBI, DQ_n]$  coinciden con el patrón de datos predeterminado  $DQM$ , eso significa que se afirmó la señal de máscara de datos original ( $dm$ ) (etapa 204 de la figura 2). Por tanto, la señal de máscara de datos ( $dm$ ) se establece en "1" para indicar que los datos codificados  $DQ_n$  deben ignorarse y  $DQ_n$  no se tiene en cuenta por el dispositivo de memoria 104. Por otro lado, si los datos codificados  $[DBI, DQ_n]$  no coinciden con el patrón de datos predeterminado  $DQM$ , entonces eso significa que los datos  $[DBI, DQ_n]$  se codificaron utilizando una codificación no DBI (etapa 208 de la figura 2). Por tanto, los datos  $[DBI, DQ_n]$  se decodifican utilizando una decodificación no DBI y la señal de máscara de datos ( $dm$ ) se establece en "0".

Observe que la decodificación de los datos  $[DBI, DQ_n]$  puede realizarse simplemente invirtiendo la lógica combinatoria utilizada para codificar los datos  $DQ_n$ . Por ejemplo, cuando los datos  $DQ_n$  de 8 bits se codifican en datos  $[DBI, DQ_n]$  de 9 bits utilizando la lógica combinatoria que se muestra en la tabla 1, los datos  $[DBI, DQ_n]$  de 9 bits se pueden decodificar para recuperar los datos  $DQ_n$  de 8 bits, simplemente invirtiendo la lógica combinatoria que se muestra en la tabla 1. Por ejemplo, los datos de 9 bits en forma de  $01000 \bar{x}_7 \bar{x}_6 \bar{x}_5 \bar{x}_4$  con HW de cuatro se pueden decodificar en datos de 8 bits  $x_7 x_6 x_5 x_4 000$  con HW de uno. Numerosos otros ejemplos de inversión de la lógica combinatoria para la decodificación son evidentes a partir de la tabla 1. En algunas realizaciones, dichas asignaciones inversas de lógica también pueden determinarse y almacenarse en una LUT.

La figura 5 ilustra un ejemplo de un circuito decodificador, según una realización. El circuito decodificador de la figura 5 es un ejemplo de un circuito configurado para realizar el esquema de decodificación como se describió anteriormente

con referencia a la figura 4, aunque pueden estar disponibles diferentes configuraciones de circuito con las mismas funciones para llevar a cabo el esquema de decodificación de la figura 4. El decodificador 112 incluye un decodificador DBI 502, un decodificador no DBI 504, puertas AND 506, 530, una puerta OR 526, un comparador 514 y un multiplexor 510. El decodificador DBI 502 está configurado para realizar la decodificación DBI en los datos codificados [DBI, DQ<sub>n</sub>] como se explicó anteriormente con la etapa 404 de la figura 5 y el codificador no DBI 504 está configurado para realizar la decodificación no DBI en los datos codificados [DBI, DQ<sub>n</sub>] como se explicó anteriormente con la etapa 410 de la figura 4. La señal de modo 516 se usa para indicar al decodificador 112 si se debe usar (cuando se afirma la señal de modo 516) el esquema de decodificación según las realizaciones de esta invención (figura 4) o se debe usar (cuando no se afirma la señal de modo 516) la decodificación DBI (que es independiente del uso de información de enmascaramiento) Para fines de ilustración en esta invención del esquema de decodificación de la figura 4, se supondrá afirmada la señal de modo 516 (lógica alta).

Volviendo al funcionamiento del circuito decodificador 112, cuando la señal de máscara de datos modificada (DM) 512 no se afirma ("0" o lógica baja), eso significa que los datos [DBI, DQ<sub>n</sub>] fueron codificados DBI (etapa 210 de la figura 2). Por tanto, la salida de la señal de máscara de datos (dm) 532 de la puerta AND 530 se establece en lógica baja ("0"). Además, la salida 508 de la puerta AND 506 también está en lógica baja haciendo así que el multiplexor 510 seleccione la salida 518 del decodificador DBI 502 como su salida DQ<sub>n</sub> 522. Por tanto, los datos [DBI, DQ<sub>n</sub>] se decodifican usando DBI (etapa 404 de la figura 4).

Cuando se afirma la señal de máscara de datos modificada (DM) 512 ("1" o lógica alta), la señal de máscara de datos de salida (dm) 532 de la puerta AND 530 está en lógica alta si la salida 524 del comparador 512 está en lógica alta, es decir, cuando los datos codificados [DBI, DQ<sub>n</sub>] coinciden con el patrón de datos predeterminado DQM como en la etapa 406 de la figura 4. Cuando la señal de máscara de datos (dm) 532 está en lógica alta, los datos codificados no se tienen en cuenta (etapa 408 de la figura 4).

Por otro lado, si la salida 524 del comparador 512 está en lógica baja (es decir, cuando los datos codificados [DBI, DQ<sub>n</sub>] no coinciden con el patrón de datos predeterminado DQM), la salida 528 de la puerta OR 528 también está en lógica baja ("0") y, por tanto, la salida de la señal de máscara de datos (dm) 532 desde la puerta AND 530 también se establece en lógica baja. Además, la salida 508 de la puerta AND 506 también está en lógica alta, haciendo así que el multiplexor 510 seleccione la salida 520 del decodificador no DBI 504 como su salida DQ<sub>n</sub> 522. Por tanto, los datos [DBI, DQ<sub>n</sub>] se decodifican usando decodificación no DBI (etapa 410 de la figura 4).

Finalmente, cuando la señal de modo 516 está en lógica baja ("0"), entonces la salida 508 de la puerta AND 506 está siempre en lógica baja independientemente del estado de la señal de máscara de datos modificada (DM) 512, haciendo así que el multiplexor 308 siempre seleccione la salida 518 del codificador DBI 502 como su salida DQ<sub>n</sub> 522. Además, la salida 528 de la puerta OR 526 está siempre en lógica alta si la señal de modo 516 está en lógica baja, y por tanto la salida de la señal de máscara de datos (dm) 532 desde la puerta AND 530 sigue el estado de la señal de máscara de datos modificada (DM) 512. Por tanto, los datos [DBI, DQ<sub>n</sub>] siempre se decodifican utilizando la decodificación DBI cuando no se afirma la señal de modo 516. En otras palabras, cuando la señal de modo 516 está en lógica baja, el decodificador 112 realiza la decodificación DBI (que es independiente del uso de información de enmascaramiento) y, por tanto, el decodificador 112 tiene una compatibilidad completamente retroactiva con los esquemas de decodificación DBI convencionales. Por ejemplo, la señal de modo 516 puede almacenarse como un valor en un registro de configuración programable o cableada por medio de un fusible o máscara de metal etc.

Con la técnica de codificación según las realizaciones descritas en esta invención, el SSN puede reducirse significativamente y la corriente de CA extraída de la fuente de alimentación también puede reducirse significativamente en comparación con los procedimientos de codificación convencionales como DBI. Esto se explica con más detalle con referencia a las figuras 6A y 6B. La figura 6A ilustra los pesos de Hamming (HW) presentes en datos de 9 bits, incluidos datos no codificados de 8 bits y un bit DBI, y la figura 6B ilustra los pesos de Hamming presentes en datos codificados de 9 bits que incluyen un bit DBI y datos codificados de 8 bits codificados según una realización.

Con referencia a la figura 6A, el histograma HW ilustra la distribución de HW en los datos de 9 bits, incluidos los datos no codificados DBI (1 bit) y 8 bits, y los histogramas HW 602 cuando dichos datos no codificados de 9 bits se codifican utilizando DBI convencional ([DBI + datos codificados DBI de 8 bits]). Entre los posibles patrones de datos (2<sup>9</sup>) de los datos no codificados en paralelo de 9 bits, hay 1 patrón de datos con HW = 0, 9 patrones de datos con HW = 1, 36 patrones de datos con HW = 2, 84 patrones de datos con HW = 3, 126 patrones de datos con HW = 4, 126 patrones de datos con HW = 5, 84 patrones de datos con HW = 6, 36 patrones de datos con HW = 7, 9 patrones de datos con HW = 8 y 1 patrón de datos con HW = 9. La variación máxima posible en los HW en el patrón de datos no codificados de 9 bits es, por tanto, 9 (entre HW = 0 y HW = 9), lo que da como resultado un SSN significativo si se transmite sin codificar.



Aún con referencia a la figura 6A, los histogramas HW 602 ilustran que solo los pesos de Hamming de 0 a 4 estarían presentes cuando dichos datos de 8 bits se codifican utilizando DBI ([DBI + datos codificados DBI de 8 bits]), ya que los datos con pesos de Hamming de 5 a 8 se invertirían. Entre los posibles patrones de datos ( $2^8$ ) de los datos codificados en paralelo de 9 bits, hay 1 patrón de datos con HW = 0, 9 patrones de datos con HW = 1, 36 patrones de datos con HW = 2, 84 patrones de datos con HW = 3 y 128 patrones de datos con HW = 4. La variación máxima posible en los HW en el patrón de datos codificados DBI de 9 bits es, por tanto, 4 (entre HW = 0 y HW = 4), lo que da como resultado un SSN reducido en comparación a cuando los datos se transmiten sin codificar.

Con referencia a la figura 6B, el histograma HW 652 ilustra la distribución de HW en los datos de 9 bits ([DBI + datos codificados de 8 bits]) incluidos los codificados según el esquema combinado de codificación de máscara de datos y DBI como se describió anteriormente con referencia a la figura 2. Entre los posibles patrones de datos ( $2^8$ ) de los datos codificados en paralelo de 9 bits, hay 5 patrones de datos con HW = 3 (incluido el patrón DQM) y 252 patrones de datos con HW = 4. La variación máxima posible en los HW en el patrón de datos de 9 bits es, por tanto, solo 1 (entre HW = 3 y HW = 4), lo que da como resultado una reducción significativa de SSN en comparación con el SSN cuando los datos se transmiten sin codificar o están codificados por DBI convencional.

Estas ventajas se muestran más específicamente en la tabla 2 a continuación, que muestra la corriente (Idd) que extraería el transmisor (Tx) 108 (figura 1). Cuando los datos  $D_n$  se transmiten sin codificar, codificados por DBI convencional, y codificados con DM/DBI combinados según las realizaciones de esta invención, se hace referencia a la corriente de suministro Idd que se va a extraer cuando los datos  $D_n$  se transmiten sin codificar como referencia (100 %).

Tabla 2

Idd	Sin codificar	DBI convencional	DM/DBI combinado codificado
Idd, CC pico (%)	100 %	50 %	50 %
Idd, CC promedio (%)	100 %	40 %	45 %
Idd, pico-pico (%)	100 %	50 %	25 %
Idd, CA rms (%)	100 %	61 %	13 %

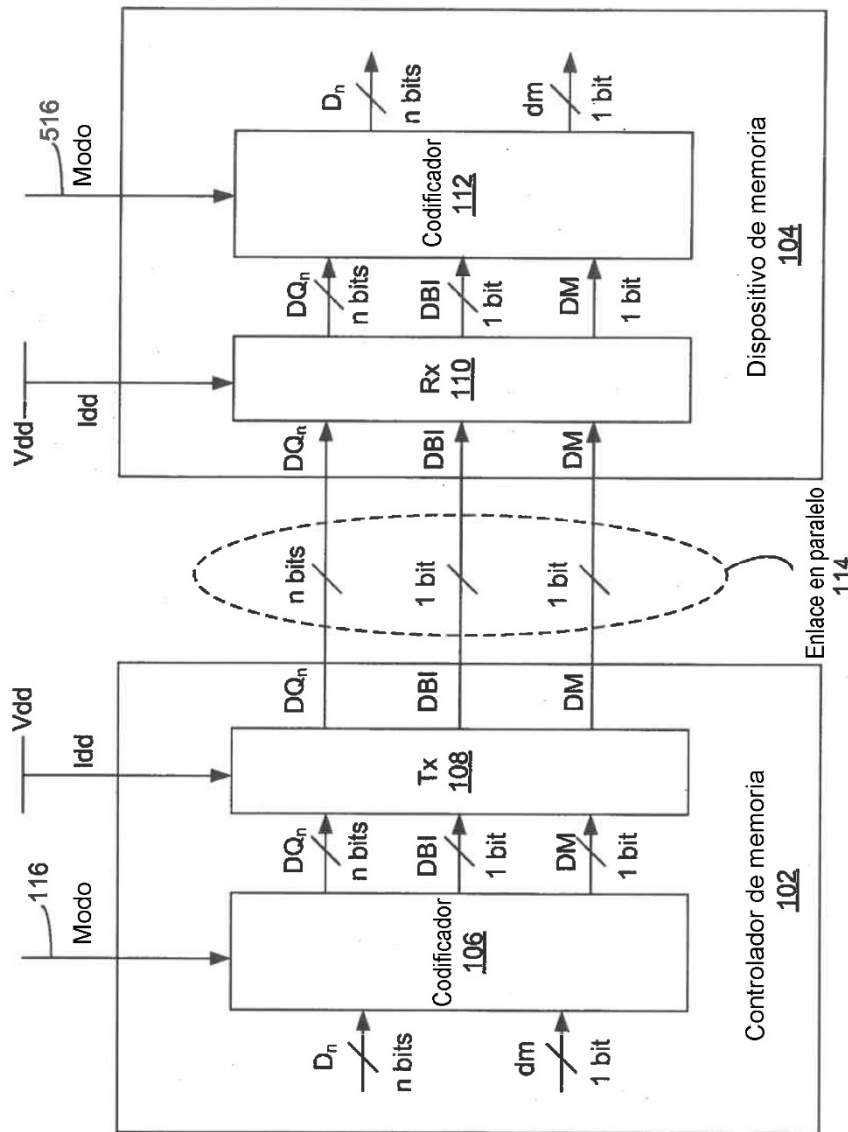
Como se muestra en la tabla 2 anterior, la corriente de suministro de CC promedio Idd aumenta ligeramente (del 40 % al 45 %) cuando se usa el esquema de codificación combinado DM/DBI descrito en esta invención, en comparación con la codificación DBI convencional. Esto se debe a que una gran cantidad de patrones de datos codificados se concentran con un alto peso de Hamming (HW = 4), lo que aumenta la corriente de CC promedio Idd extraída de la fuente de alimentación (no mostrada). Por otro lado, la corriente de suministro de CA extraída de la fuente de alimentación se reduce drásticamente, del 50 % al 25 % para la corriente de suministro CA pico a pico Idd del 61 % al 13 % para los valores CA rms (raíz cuadrática media), cuando se utiliza el esquema combinado de codificación DM/DBI según las realizaciones descritas en esta invención, en comparación con la codificación DBI convencional. Por tanto, el ruido SSO puede reducirse drásticamente cuando se usa el esquema combinado de codificación DM/DBI según las realizaciones descritas en esta invención.

Al leer esta descripción, los expertos en la materia apreciarán aún diseños estructurales y funcionales alternativos adicionales para codificar datos para reducir el ruido SSO, a través de los principios descritos de la presente descripción. Por tanto, aunque se han ilustrado y descrito realizaciones y aplicaciones particulares de la presente descripción, debe entenderse que la descripción no se limita a la construcción precisa y los componentes descritos en esta invención. Se pueden hacer varias modificaciones, cambios y variaciones que serán evidentes para los expertos en la materia en la disposición, funcionamiento y detalles del procedimiento y aparato de la presente descripción descrita en esta invención sin apartarse del alcance de la descripción como se define en las reivindicaciones adjuntas.

**REIVINDICACIONES**

1. Un controlador de memoria (102) para controlar el funcionamiento de un dispositivo de memoria DRAM (104), comprendiendo el controlador de memoria (102):
- 5 un codificador (106) para generar datos codificados (DQ) y una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ), donde el codificador (106) está configurado para:
- 10 (i) si los datos codificados (DQ) deben enmascarse por el dispositivo de memoria DRAM (104), generar los datos codificados (DQ) como un patrón de bits predeterminado y generar que la señal de máscara de datos (DM) tenga un primer estado;
- (ii) si los datos codificados (DQ) se deben invertir y escribir en una matriz del dispositivo de memoria DRAM (104), generar los datos codificados (DQ) como datos invertidos y generar que la señal de máscara de datos (DM) tenga un segundo estado.
- 15 (iii) si los datos codificados (DQ) se deben no invertir y escribir en la matriz del dispositivo de memoria DRAM (104), generar los datos codificados (DQ) como datos no invertidos y generar que la señal de máscara de datos (DM) tenga el primer estado; y
- un transmisor (108) configurado para transmitir los datos codificados (DQ) y la señal de máscara de datos (DM) al dispositivo de memoria DRAM (104).
- 20
2. El controlador de memoria (102) según la reivindicación 1, donde el codificador está adaptado para recibir una señal de modo (116), donde
- 25 (a) si la señal de modo define un primer estado, el codificador está configurado para realizar las etapas (i) a (iii); y (b) si la señal de modo define un segundo estado, el codificador está configurado para generar los datos codificados utilizando la inversión del bus de datos.
3. Un dispositivo de memoria DRAM (104), que comprende:
- 30 una matriz para almacenar datos;
- un receptor (110) configurado para recibir datos codificados (DQ) y una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ) desde un controlador de memoria (102); y
- 35 un decodificador (112) configurado para decodificar los datos codificados (DQ) y la señal de máscara de datos (DM) para generar datos (D<sub>n</sub>),
- donde el decodificador (112) está configurado además para:
- 40 (i) si la señal de máscara de datos (DM) está en un primer estado y los datos codificados coinciden con un patrón de bits predeterminado, enmascarar que los datos codificados (DQ) recibidos desde dicho controlador de memoria (102) se escriban en la matriz.
- (ii) si la señal de máscara de datos (DM) está en un segundo estado, invertir los datos codificados antes de escribir los datos en la matriz del dispositivo de memoria DRAM (104);
- 45 (iii) si la señal de máscara de datos (DM) está en el primer estado y los datos codificados no coinciden con el patrón de bits predeterminado, escribir los datos (D<sub>n</sub>) en la matriz del dispositivo de memoria DRAM (104) sin inversión.
4. El dispositivo de memoria DRAM (104), según la reivindicación 2, que comprende además un registro que especifica un modo para que la DRAM indique cómo usar los datos codificados (DQ) y la señal de máscara de datos (DM), donde
- 50 (a) si el modo define un primer estado, el decodificador está configurado para realizar las etapas (i) a (iii); y (b) si el modo define un segundo estado, el decodificador está configurado para invertir los datos codificados antes de escribir los datos en la matriz del dispositivo de memoria DRAM (104) independientemente de la señal de máscara de datos (DM).
- 55
5. Un procedimiento realizado por un controlador de memoria (102) para controlar el funcionamiento de un dispositivo de memoria DRAM (104), comprendiendo el procedimiento las etapas de:
- 60 (i) generar datos codificados (DQ) como un patrón de bits predeterminado y generar una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ) que tiene un primer estado, si el dispositivo de memoria DRAM (104) debe enmascarar los datos codificados (DQ);

- (ii) generar datos codificados (DQ) como datos invertidos y generar que una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ) tenga un segundo estado, si los datos codificados (DQ) se deben invertir y escribir en una matriz del dispositivo de memoria DRAM (104);
- 5 (iii) generar datos codificados (DQ) como datos no invertidos y generar que una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ) tenga el primer estado, si los datos codificados (DQ) se deben no invertir y escribir en una matriz del dispositivo de memoria DRAM (104); y
- (iv) transmitir los datos codificados (DQ) y la señal de máscara de datos (DM) al dispositivo de memoria DRAM (104).
- 10 6. El procedimiento según la reivindicación 5, donde el procedimiento comprende además recibir antes de la etapa (i) una señal de modo por el controlador de memoria;
- (a) si la señal de modo define un primer estado, realizar las etapas (i) a (iv); y
- (b) si la señal de modo define un segundo estado, generar datos codificados como datos invertidos en lugar de las
- 15 etapas (i) a (iii) antes de realizar la etapa (iv).
7. Un procedimiento realizado por un dispositivo de memoria DRAM (104) de recepción de datos de un controlador de memoria (102), comprendiendo el procedimiento las etapas de:
- 20 recibir datos codificados (DQ) y una señal de máscara de datos (DM) que acompaña a los datos codificados (DQ) desde un controlador de memoria (102); y
- decodificar los datos codificados (DQ) y la señal de máscara de datos (DM) para generar datos (D<sub>n</sub>),
- donde la decodificación comprende:
- 25 (i) si la señal de máscara de datos (DM) está en un primer estado y los datos codificados coinciden con un patrón de bits predeterminado, enmascarar los datos codificados (DQ) recibidos desde dicho controlador de memoria (102);
- (ii) si la señal de máscara de datos (DM) está en un segundo estado, invertir los datos antes de escribir los datos
- 30 en la matriz del dispositivo de memoria DRAM (104);
- (iii) si la señal de máscara de datos (DM) está en el primer estado y los datos no coinciden con el patrón de bits predeterminado, escribir los datos (DQ) en la matriz del dispositivo de memoria DRAM (104).
8. El procedimiento según la reivindicación 7, donde el procedimiento comprende además obtener de un
- 35 registro un modo para que la DRAM indique cómo usar los datos codificados (DQ) y la señal de máscara de datos (DM),
- (a) si el modo define un primer estado, realizar las etapas (i) a (iii); y
- (b) si el modo define un segundo estado, invertir los datos antes de escribir los datos en la matriz del dispositivo
- 40 de memoria DRAM (104) independientemente de la señal de máscara de datos (DM).



**FIG. 1**

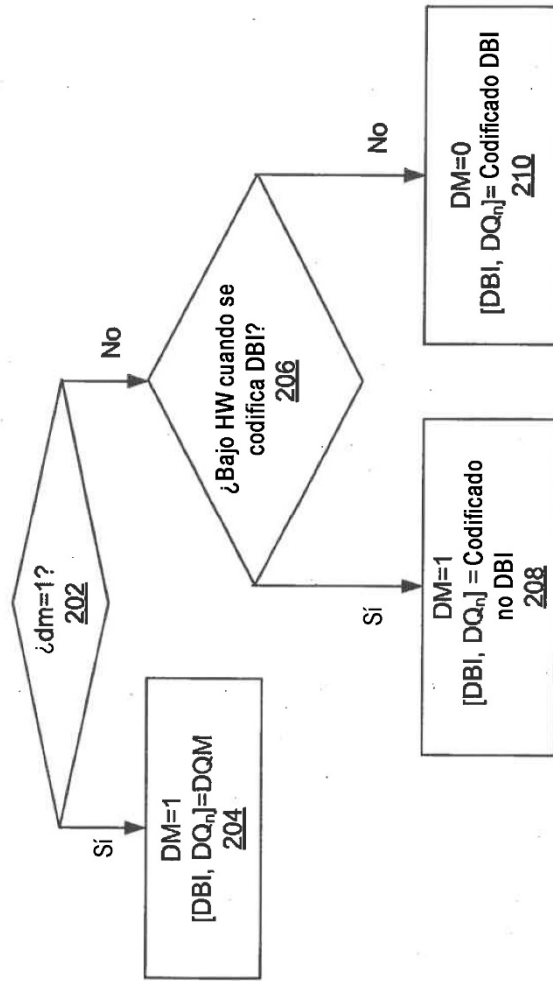


FIG. 2

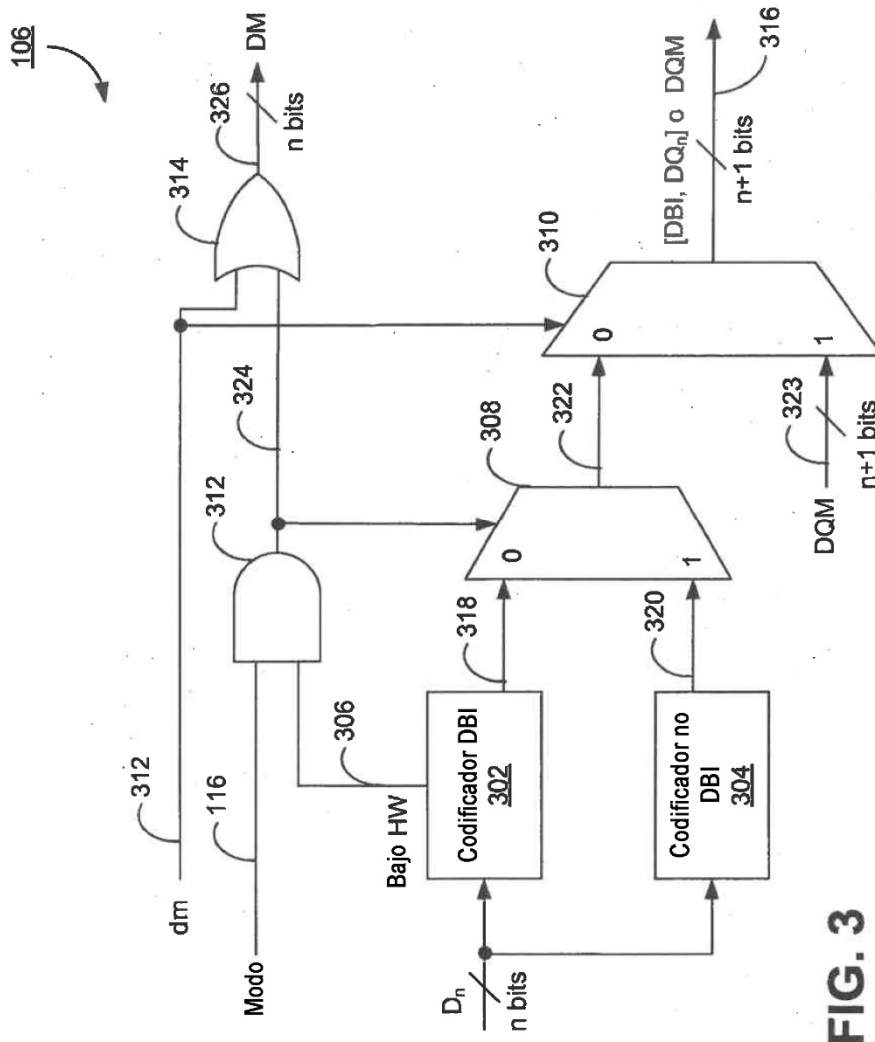


FIG. 3

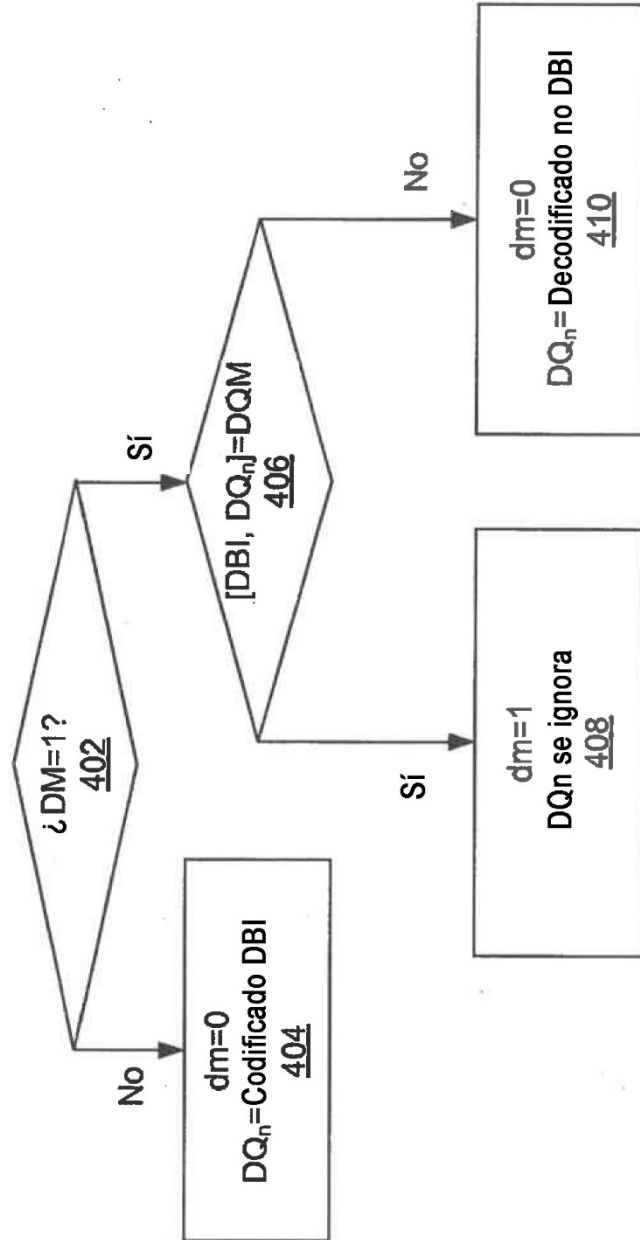


FIG. 4

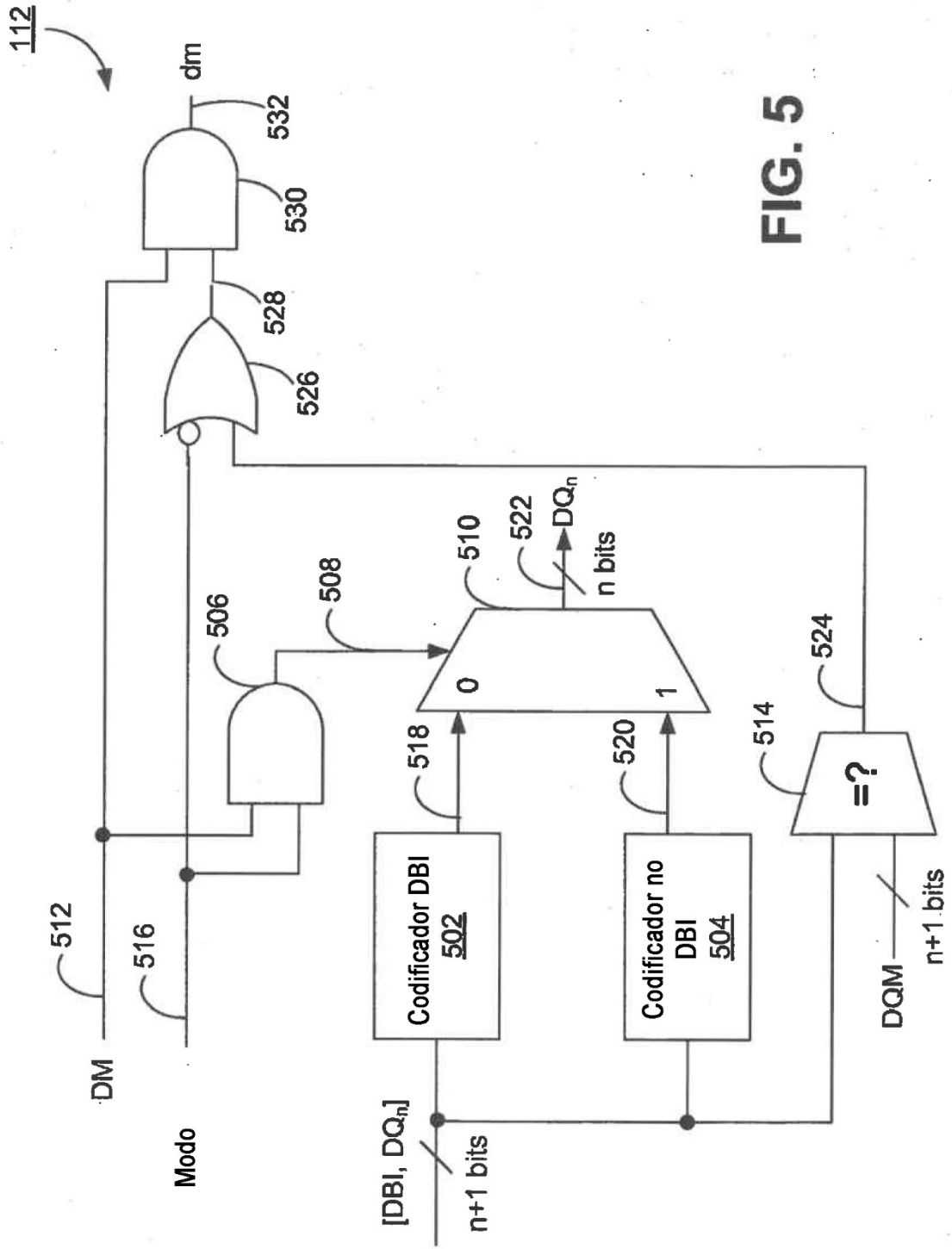
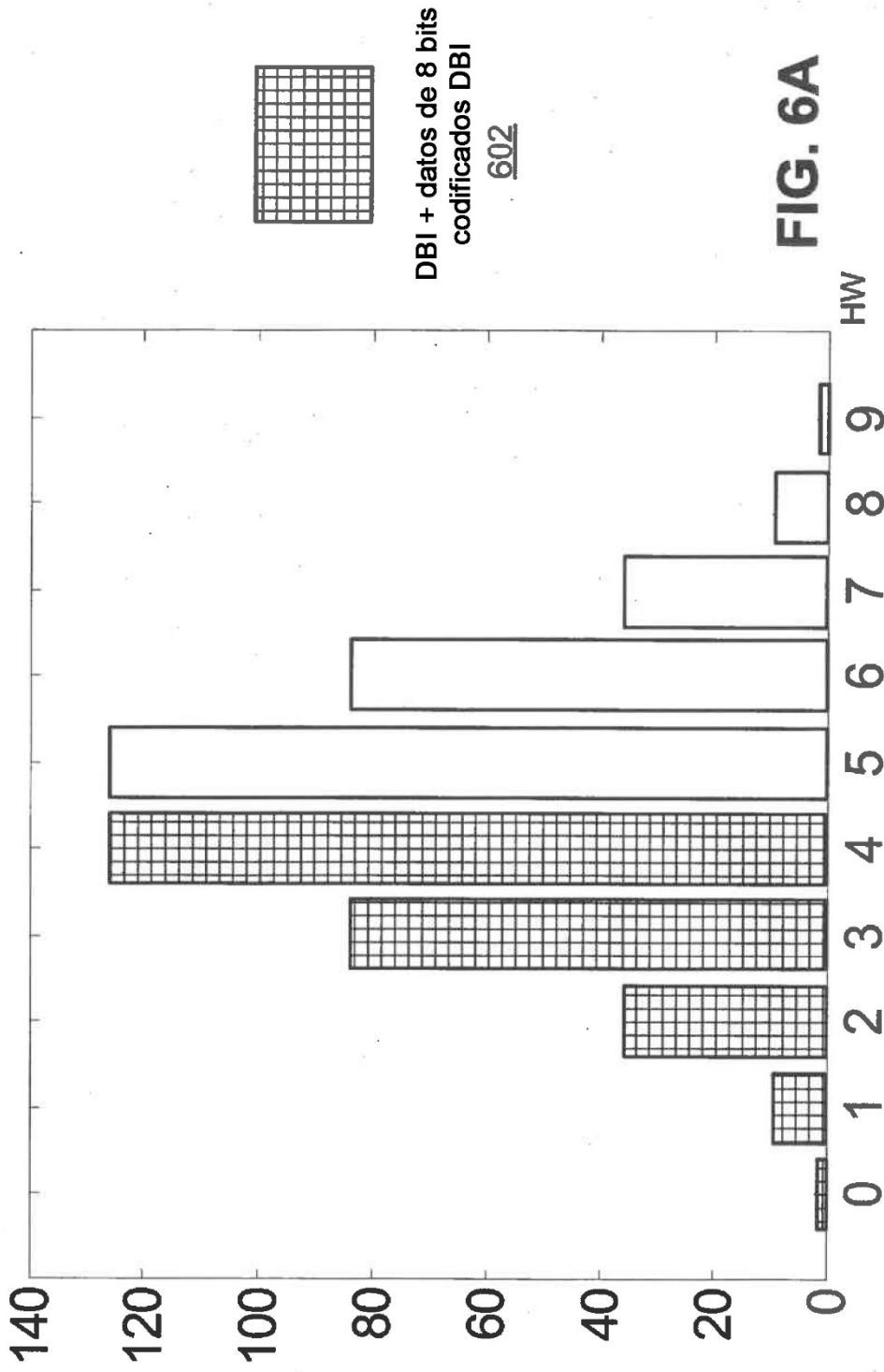


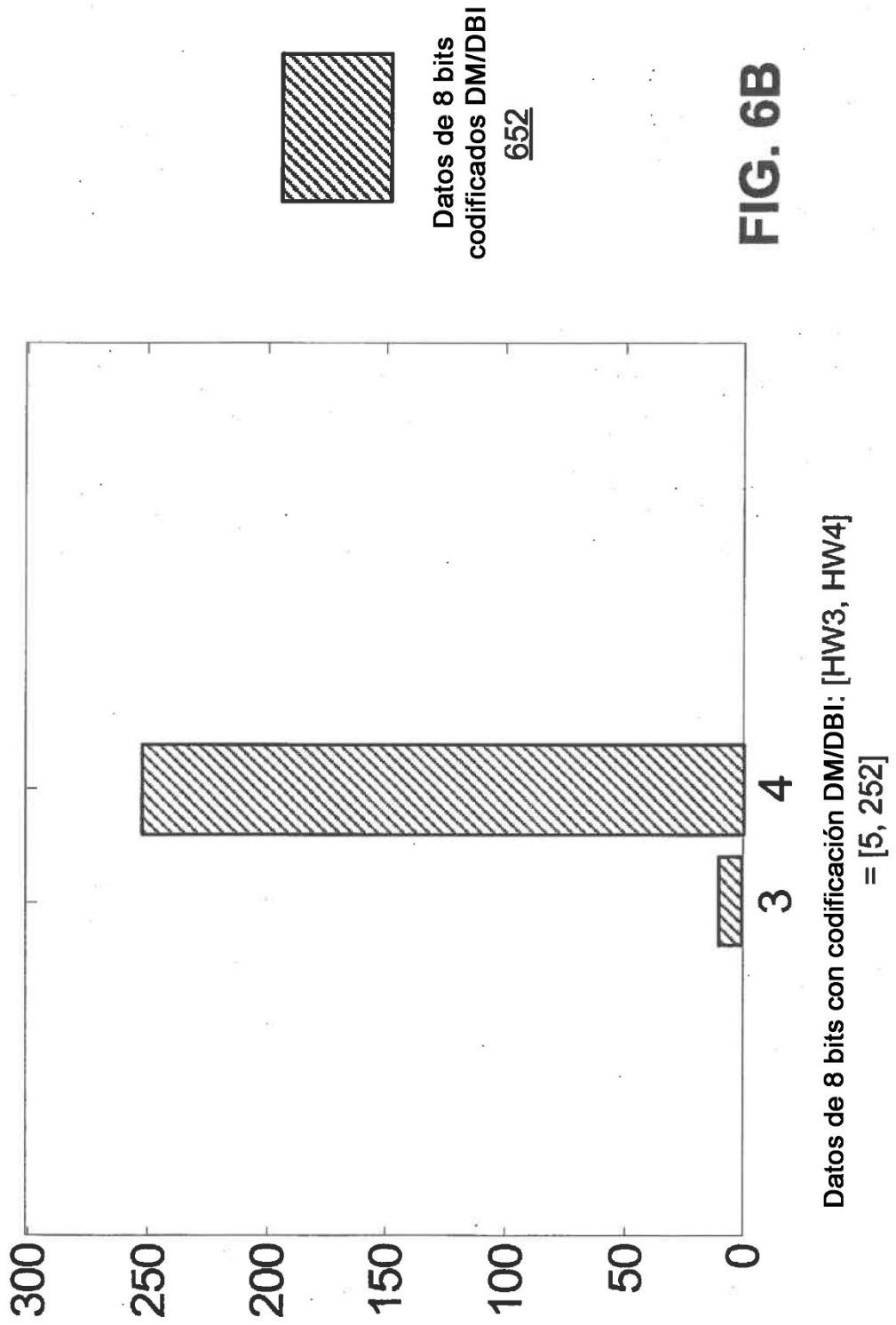
FIG. 5





Datos de 8 bits no codificados: [HW0, HW1, HW2, HW3, HW4, HW5, HW6, HW7, HW8, HW9]  
= [1, 9, 36, 84, 126, 126, 84, 36, 9, 1]

DBI + datos de 8 bits codificados DBI: [HW0, HW1, HW2, HW3, HW4]  
= [1, 9, 36, 84, 126]



**FIG. 6B**