

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 660 581**

51 Int. Cl.:

**H03M 13/11** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **24.04.2015** E 15290110 (4)

97 Fecha y número de publicación de la concesión europea: **22.11.2017** EP 3086474

54 Título: **Método para controlar un nodo de comprobación de un decodificador NB-LDPC y nodo de comprobación correspondiente**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**23.03.2018**

73 Titular/es:  
**TECHNISCHE UNIVERSITÄT KAISERSLAUTERN (33.3%)  
Postfach 3049  
67663 Kaiserslautern, DE;  
CREONIC GMBH (33.3%) y  
UNIVERSITE DE BRETAGNE SUD (33.3%)**

72 Inventor/es:  
**BOUTILLON, EMMANUEL;  
SCHLÄFER, PHILIPP y  
LEHNIGK-EMDEN, TIMO**

74 Agente/Representante:  
**ISERN JARA, Jorge**

ES 2 660 581 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**

Método para controlar un nodo de comprobación de un decodificador NB-LDPC y nodo de comprobación correspondiente

5 1. Campo técnico

La presente invención se refiere a un método para controlar un nodo de comprobación de un decodificador para decodificar códigos de LDPC no binarios y al nodo de comprobación correspondiente

10 2. Antecedentes de la técnica

Los códigos de LDPC (Comprobación de Paridad de Baja Densidad) son códigos de corrección de errores conocidos para acercarse al límite de transmisión teórico de Shannon. Debido a su rendimiento en términos de inmunidad de ruido, los códigos de LDPC, y en particular códigos de LDPC no binarios, hallan uso en el campo de transmisión y almacenamiento de datos digitales.

Los códigos de LDPC no binarios se construyen en un campo de Galois no binario de orden q, escrito de manera convencional como GF(q). Un código de LDPC en GF(q) se identifica por una matriz de paridad dispersa H de M x N dimensiones, los elementos de los cuales pertenecen a GF(q), siendo N el número de GF(q) elementos en la palabra de código y siendo M el número de restricciones de paridad. Para un campo de Galois GF(4) que consiste en 4 elementos {0,  $\alpha^0$ ,  $\alpha^1$ ,  $\alpha^2$ }, la matriz de paridad para N=6 y M=3, por ejemplo, es de la siguiente manera:

$$H = \begin{pmatrix} 0 & \alpha^0 & \alpha^0 & 0 & \alpha^2 & 0 \\ \alpha^1 & 0 & 0 & \alpha^0 & 0 & \alpha^2 \\ \alpha^0 & 0 & \alpha^2 & 0 & 0 & \alpha^1 \end{pmatrix}$$

25 Esta matriz puede mostrarse análogamente por un grafo bipartito (grafo de Tanner) que tiene nodos N variables que reciben los símbolos de la palabra de código y M nodos de comprobación. Cada columna de la matriz de paridad está asociada con un nodo variable y cada línea de la matriz está asociada con un nodo de comprobación. En el caso de un código de LDPC normal, cada nodo de comprobación está conectado, por medio de  $d_c$  ramales, a un único número  $d_c$  de nodos variables. De manera similar, cada nodo variable está conectado, por medio de  $d_v$  ramales, a un único número  $d_v$  de nodos de comprobación. En el caso de un código de LDPC regular, el número  $d_c$  o  $d_v$  de ramales varía de acuerdo con el nodo variable o nodo de comprobación.

30 La decodificación de estos códigos de LDPC es iterativa y consiste intercambiar mensajes mediante estos ramales, procesando cada nodo variable o de comprobación los mensajes recibidos y, después del procesamiento, entregando otros mensajes, como pueda ser el caso, a nodos de comprobación o nodos variables a los que están conectados mediante ramales. Los mensajes intercambiados representan funciones de densidad de probabilidad de los símbolos de código y se encuentran por lo tanto en forma de vectores de tamaño q. Tal decodificación se describe en el documento titulado "Decoding Algorithms for Non binary LDPC Codes Over GF(q)" por D. Declercq y M. Fossorier, Communications IEEE Transactions in Volume 55, Tema 4, abril de 2007, páginas 633-643.

35 La invención se refiere específicamente al método para controlar nodos de comprobación del decodificador. Con referencia a la Figura 1, cada nodo de comprobación CN recibe una pluralidad de mensajes entrantes  $U_i$  y entrega mensajes salientes  $V_i$ , con  $i, j \in [1...d_c]$ , de manera que

45

$$V_j = \sum_{i \neq j} U_i$$

En el ejemplo de la figura 1, el nodo de comprobación recibe  $d_c=4$  mensajes entrantes  $U_1, U_2, U_3, U_4$  y entrega 4 mensajes salientes  $V_1, V_2, V_3, V_4$  tal como:

50

$$\begin{cases} V_1 = U_2 + U_3 + U_4 \\ V_2 = U_1 + U_3 + U_4 \\ V_3 = U_1 + U_2 + U_4 \\ V_4 = U_1 + U_2 + U_3 \end{cases}$$

en el que "+" es el operador de adición en el campo de Galois GF(q).

Cada mensaje entrante y saliente es un vector de tamaño  $q$ , que asocia una densidad de probabilidad con cada elemento del campo de Galois. Cada nodo de comprobación realiza cálculos en los mensajes entrantes y entrega mensajes salientes a los nodos variables.

5 Las funciones de probabilidad de densidad por lo tanto se intercambian entre los nodos de comprobación y los nodos variables, y a la inversa, hasta que se ha decodificado completamente la palabra de código o se ha realizado un número predefinido de iteraciones. Se hace referencia a continuación a un algoritmo de "propagación de creencia" o algoritmo BP.

10 Implementar un algoritmo de este tipo es relativamente complejo y requiere un elevado número de cálculos en el nodo de comprobación. Con tal algoritmo, la complejidad aumenta con el tamaño del  $GF(q)$ . Una implementación sencilla del algoritmo de BP tiene una complejidad de  $O(q^2)$ .

15 Para reducir esta complejidad, es conocido trabajar en el campo logarítmico para evitar convertir las multiplicaciones en adiciones. Los datos intercambiados entre nodos son relaciones de probabilidad logarítmica (LLR).

20 Para calcular las funciones de densidad de probabilidad del mensaje saliente, otra solución conocida para reducir dicha complejidad consiste en tener en cuenta solamente las  $n_m$  funciones de densidad de probabilidad más grandes en los mensajes entrantes,  $n_m < q$ . Este algoritmo, denominado como EMS (Mínimo-Suma Extendido), se describe en detalle en el documento titulado "Algorithmes simplifiés pour le décodage de codes LDPC non binaires" de A.Voiclla, D.Declercq, M.Fossorler et F.Verdler, GRETSI, Louvain-la-Neuve, Bélgica, septiembre de 2005. Esta solución puede combinarse con la solución anterior de manera que los datos intercambiados son relaciones de probabilidad logarítmica.

25 De acuerdo con este algoritmo, los mensajes entrantes  $U_i$  se filtran y ordenan antes de que se procesen por el nodo de comprobación. Los mensajes suministrados al nodo de comprobación son entonces listas que comprenden únicamente  $n_m$  valores de función de densidad de probabilidad o LLR ordenados en orden descendente, asociándose cada función de densidad de probabilidad o valor de LLR con un elemento del campo de Galois  $GF(q)$ . Reduciéndose el número de función de probabilidad o valores de LLR tenidos en cuenta por el nodo de comprobación ( $n_m < q$ ), la complejidad de este algoritmo se reduce pero, sin embargo, permanece alta.

Actualmente, el algoritmo de EMS proporciona el mejor compromiso entre complejidad de hardware y rendimiento de comunicaciones.

35 Para conseguir el rendimiento requerido de las aplicaciones de hoy en día, ejecutar los algoritmos en software no es suficiente. Se vuelven obligatorias arquitecturas de hardware especializadas. La mayor complejidad en el algoritmo de EMS es el cálculo del Nodo de Comprobación (CN). Las arquitecturas del estado de la técnica aplican un denominado esquema hacia delante/atrás (FWBW) para procesar el nodo de comprobación. Esta arquitectura se desvela en "Log-domain decoding of LDPC codes over  $GF(q)$ ", H. Wymeersch, H. Steendam y M. Moeneclaey, Proc. IEEE International Conference on Communications, vol. 2 de junio de. 2004, págs. 772-776.

40 Se lleva a cabo el cálculo en serie para reducir el coste de hardware y para permitir la reutilización de resultados intermedios durante el cálculo. Sin embargo este esquema introduce alta latencia y degrada el rendimiento. Este efecto aumenta significativamente cuando crece el tamaño de  $GF(q)$ .

45 Un fin de la invención es proporcionar un esquema para permitir la reducción de estos

50 Se observa que el documento US2013/0212451 desvela un método de decodificación de un LDPC no binario. En este método, el procesamiento del Nodo de Comprobación añade LLR y añade elementos de GF, excluyendo ambos un borde  $j$ . Para producir el mensaje de C2V en ese borde. La fórmula (10) muestra añadir todos las LLR entrantes de unos nodos de comprobación, y restar el LLR de borde del borde  $j$  para obtener el mensaje de C2V en el borde  $j$ . Esto es, sin embargo, una etapa de cálculo intermedio no usada realmente en el procesamiento de CN.

### 3. Sumario de la invención

55 De acuerdo con la invención, se propone un nuevo método para controlar un nodo de comprobación en un decodificador NB-LDPC (Comprobación de Paridad de Baja Densidad No Binaria) que permite paralelismo aumentado del cálculo del nodo de comprobación. El procesamiento aplicado por el nodo de comprobación se denomina procesamiento de Nodo de Comprobación (CN) basado en Síndrome (SYN). Aunque se consigue ligeramente mejor rendimiento de comunicaciones que los algoritmos de decodificación conocidos del hardware del estado de la técnica, el procesamiento de CN de SYN tiene una complejidad inferior y el paralelismo aumentado del cálculo del nodo de comprobación posibilita arquitecturas de decodificador de baja latencia y alto rendimiento.

65 La invención se refiere a un método para controlar un nodo de comprobación de un decodificador para decodificar códigos de LDPC no binarios, recibiendo dicho nodo de comprobación  $d_c$  listas de entrada  $U_i$  de  $n_m$  elementos y entregando  $d_c$  listas de salida  $V_i$  de  $n'_m$  elementos, con  $i \in [1 \dots d_c]$ , con  $d_c > 2$ , cada elemento de las listas de entrada o

de salida, denominado respectivamente elemento de entrada y salida, que comprende un valor de fiabilidad asociado a un símbolo de un campo de Galois  $GF(q)$  con  $q > n_m$  y  $q > n'_m$ , ordenándose sustancialmente los elementos de entrada y elementos de salida de acuerdo con los valores de fiabilidad respectivamente en dicha lista de entrada y lista de salida, dicho método estando caracterizado porque incluye las siguientes etapas:

- 5
- añadir  $d_c$  elementos de entrada de listas de entrada  $U_i$  para generar una pluralidad de sumas denominadas síndromes, perteneciendo cada uno de dichos elementos de entrada a una lista de entrada distintiva entre las  $d_c$  listas de entrada  $U_i$  y comprendiendo cada síndrome un valor de fiabilidad que es la suma de los valores de fiabilidad de dichos elementos de entrada y un símbolo del campo de Galois que es la suma de los símbolos de
  - 10 dichos elementos de entrada en el campo de Galois,
  - para cada lista de salida  $V_i$ , aplicar una descorrelación a los síndromes restando el elemento de entrada de la lista de entrada  $U_i$  de los síndromes para generar síndromes descorrelacionados, y
  - para cada lista de salida  $V_i$  seleccionar, como elementos de salida de dicha lista de salida  $V_i$ , los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos y generados para dicha lista de salida  $V_i$ .

15 Este método permite una implementación paralela masiva, en particular para las operaciones de descorrelación, dando como resultado un procesamiento de alto rendimiento y baja latencia.

20 El valor de fiabilidad se deriva preferentemente de un valor de LLR (relación de probabilidad logarítmica). El valor de fiabilidad más alto se proporciona por el valor de LLR más bajo y a la inversa el valor de fiabilidad más bajo se proporciona por el valor de LLR más alto.

25 En una realización particular, en la etapa de adición, cada síndrome se genera basándose en como máximo  $k$  elementos de entrada diferentes de los elementos de entrada que tienen el valor de fiabilidad más alto (LLR=0), con  $k < d_c$ .

30 En esta realización, únicamente los síndromes con los valores de fiabilidad más altos se usan para generar las listas de salida  $V_i$ , que hace el cálculo de todos los demás síndromes superfluo. Únicamente se generan los síndromes con pocas desviaciones (con pocos elementos de entrada que no tienen el valor de fiabilidad más alto (LLR=0)). Da como resultado que el número de síndromes a generarse se reduce significativamente.

35 En una realización particular, en la etapa de adición, cada síndrome se genera basándose en elementos de entrada cuya distancia, denominada distancia de fiabilidad, de los elementos de entrada que tienen los valores de fiabilidad más altos es inferior a una distancia de fiabilidad máxima. Los elementos de entrada con los valores de fiabilidad más bajos no se usan para la generación de los síndromes. También contribuye a reducir el número de síndromes a generarse.

40 En una realización particular, la distancia de fiabilidad máxima depende de  $k$ . La distancia de fiabilidad máxima se enlaza al número de desviaciones para un síndrome. Ventajosamente, cuanto más alto es el número de desviaciones para un síndrome, menor es la distancia de fiabilidad máxima.

45 En una realización particular, los elementos de salida de la lista de salida  $V_i$  se seleccionan ordenando los síndromes descorrelacionados generados para la lista de salida  $V_i$  de acuerdo con los valores de fiabilidad y seleccionando los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos (valores de LLR más bajos). En esta realización, se realizan operaciones de ordenación en los síndromes descorrelacionados para generar las listas de salida  $V_i$ .

50 En otra realización particular, para una lista de salida  $V_i$  a generarse, la descorrelación se aplica a síndromes generados del elemento de entrada de la lista de entrada  $U_i$  que tienen el valor de fiabilidad más alto (LLR=0). En esta realización, únicamente se consideran los síndromes que usan el elemento más fiable (LLR = 0) de la lista de entrada  $U_i$  para generar la lista de salida  $V_i$ . Todos los otros síndromes no se usan. Con esta realización, no se requiere operación de resta de LLR.

55 En una realización particular vinculada con la realización anterior, antes de la etapa de descorrelación, los síndromes se ordenan de acuerdo con los valores de fiabilidad de dichos síndromes de manera que, después de la etapa de descorrelación, los síndromes descorrelacionados generados para la lista de salida  $V_i$  se ordenan de acuerdo con los valores de fiabilidad, y los elementos de la lista de salida  $V_i$  son los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos. En esta realización, la operación de ordenación puede hacerse antes de la etapa de descorrelación. Permite reducir significativamente el número de ordenadores de  $d_c$  a 1.

60 En otra realización particular, antes o después o en paralelo con la generación de síndrome, el método comprende adicionalmente las etapas de:

- 65
- preseleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,

- evaluar dichas sondas preseleccionadas para seleccionar y ordenar un número predeterminado de las sondas preseleccionadas, dichas sondas, denominadas sondas finales, ordenándose de acuerdo con los valores de fiabilidad; y
- seleccionar síndromes generados basándose en dichas sondas finales, ordenándose dichos síndromes en el orden de las sondas finales.

Con esta realización, los elementos de salida de las listas de salida  $V_i$  no se ordenan exactamente de acuerdo con los valores de fiabilidad pero no es obligatorio que el nodo variable reciba estas listas de salida. En esta realización, únicamente se usa una cantidad reducida de los elementos de entrada, denominados sondas, para seleccionar los síndromes a descorrelacionarse.

En una variante, antes de la generación de síndrome, el método comprende adicionalmente las etapas de:

- preseleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda, y
- evaluar dichas sondas preseleccionadas para seleccionar y ordenar un número predeterminado de las sondas preseleccionadas, dichas sondas, denominadas sondas finales, ordenándose de acuerdo con los valores de fiabilidad;

y en el que, en la etapa de generación de síndrome, los síndromes se generan basándose en dichas sondas finales, ordenándose dichos síndromes en el orden de las sondas finales.

En esta variante, únicamente se generan los síndromes basados en las sondas finales. Por lo tanto se calcula una cantidad más reducida de síndromes.

En una realización particular, en cada lista de entrada  $U_i$ , las sondas se distribuyen igualmente.

Para  $GF(q)$  con valores altos de  $q$  (mayores que 64), es posible distribución desigual.

En una realización particular, la sonda es el elemento de entrada que tiene el valor de fiabilidad más alto en el grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda y el valor de fiabilidad de la sonda es dicho valor de fiabilidad más alto.

En otra realización, el valor de fiabilidad de la sonda es una combinación de los valores de fiabilidad de los  $p$  elementos de entrada vecinos. El valor de fiabilidad de la sonda es, por ejemplo, el valor promedio de los valores de fiabilidad de los  $p$  elementos de entrada vecinos.

La invención también se refiere a un nodo de comprobación de un decodificador para decodificar códigos de LDPC no binarios que incluye:

- $d_c$  entradas para recibir  $d_c$  listas de entrada  $U_i$  de  $n_m$  elementos, denominados elementos de entrada, con  $i \in [1 \dots d_c]$ ,  $n_m > 1$  y  $d_c > 2$ , comprendiendo cada elemento de entrada un valor de fiabilidad asociado a un símbolo de un Campo de Galois  $GF(q)$  con  $q > n_m$ , ordenándose sustancialmente los elementos de entrada de acuerdo con los valores de fiabilidad en dicha lista de entrada,
- $d_c$  salidas para entregar  $d_c$  listas de salida  $V_i$  de  $n'_m$  elementos denominados elementos de salida, con  $q > n'_m$ , comprendiendo cada elemento de salida un valor de fiabilidad asociado a un símbolo de un Campo de Galois  $GF(q)$ , ordenándose sustancialmente los elementos de salida de acuerdo con los valores de fiabilidad en dicha lista de salida,
- un calculador de síndrome para añadir  $d_c$  elementos de entrada de las listas de entrada  $U_i$  para generar una pluralidad de sumas denominadas síndromes, perteneciendo cada uno de dicho elemento de entrada a una lista de entrada distintiva entre las  $d_c$  listas de entrada  $U_i$  y comprendiendo cada síndrome un valor de fiabilidad que es la suma de los valores de fiabilidad de dichos elementos de entrada y un símbolo del campo de Galois que es la suma de los símbolos de dichos elementos de entrada en el campo de Galois,
- $d_c$  descorreladores para aplicar, para cada lista de salida  $V_i$ , una descorrelación a los síndromes restando el elemento de entrada de la lista de entrada  $U_i$  de los síndromes para generar síndromes descorrelacionados, y
- medios para seleccionar, para cada lista de salida  $V_i$ , como elementos de salida para dicha lista de salida  $V_i$ ,  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos y generados para dicha lista de salida  $V_i$ .

En una realización particular, el nodo de comprobación comprende  $d_c$  ordenadores para ordenar los síndromes descorrelacionados de acuerdo con los valores de fiabilidad, estando especializado cada uno de dichos ordenadores para ordenar los síndromes descorrelacionados generados para una lista de salida especializada  $V_i$ .

En una realización particular, el nodo de comprobación comprende un ordenador para ordenar los síndromes generados por el calculador de síndrome de acuerdo con los valores de fiabilidad, aplicándose la descorrelación, para una lista de salida  $V_i$  a generarse, a síndromes generados del elemento de entrada de la lista de entrada  $U_i$  que tienen los valores de fiabilidad más altos.

- 5 En una realización particular, el nodo de comprobación comprende adicionalmente:
- un selector de sondas para seleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,
  - 10 - un ordenador de sondas para ordenar las sondas de acuerdo con los valores de fiabilidad, y
  - un selector de síndrome que selecciona síndromes generados basándose en dichas sondas, ordenándose dichos síndromes en el orden de las sondas ordenadas.

- 15 En una realización particular, el nodo de comprobación comprende adicionalmente:
- un selector de sondas para seleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,
  - 20 - un ordenador de sondas para ordenar las sondas de acuerdo con los valores de fiabilidad,

y en el que el calculador de síndrome se acciona para generar síndromes basándose en dichas sondas ordenadas, ordenándose dichos síndromes en el orden de las sondas ordenadas.

#### 25 4. Breve descripción de los dibujos

La invención puede entenderse mejor con referencia a la siguiente descripción y dibujos, dados a modo de ejemplo y que no limitan el alcance de protección y en los que:

- 30 - la Figura 1 es una representación esquemática de un nodo de comprobación;
- la Figura 2 es un diagrama de flujo que ilustra el método basado en síndrome de acuerdo con la invención;
- la Figura 3 es una representación esquemática de un nodo de comprobación que implementa el método basado en síndrome de la Figura 2;
- 35 - la Figura 4 es una representación esquemática de síndromes;
- la Figura 5 son gráficos que muestran los valores de LLR promedio de síndromes de diferentes conjuntos de desviación;
- la Figura 6 es un diagrama de flujo que ilustra las etapas de una primera variante del método basado en síndrome de acuerdo con la invención;
- 40 - la Figura 7 es una representación esquemática de un nodo de comprobación que implementa el método basado en síndrome de la Figura 6;
- la Figura 8 es una representación esquemática de un descorrelador de la comprobación ilustrada por la Figura 7;
- la Figura 9 es una representación esquemática de un ordenador para ordenar síndromes de los conjuntos de desviación  $D_1$  y  $D_2$ ;
- 45 - la Figura 10 es una representación esquemática de una porción del ordenador de la Figura 9;
- la Figura 11 es una vista esquemática que muestra la distribución de sondas;
- la Figura 12 es un diagrama de flujo que ilustra las etapas de una segunda variante del método basado en síndrome de acuerdo con la invención usando sondas;
- la Figura 13 son curvas que ilustran resultados del método de la Figura 12;
- 50 - la Figura 14 es una representación esquemática de un nodo de comprobación que implementa el método basado en síndrome de la Figura 12;
- la Figura 15 es una representación esquemática de un evaluador de sonda del nodo de comprobación de la Figura 14;
- la Figura 16 es una representación esquemática de un calculador de síndrome del nodo de comprobación de la Figura 14;
- 55 - la Figura 17 es una representación esquemática de un selector de síndrome del nodo de comprobación de la Figura 14;
- la Figura 18 es una representación esquemática de una variante de la comprobación de la Figura 14; y
- 60 - la Figura 19 es un diagrama de flujo que ilustra las etapas del método basado en síndrome implementadas por el nodo de comprobación de la Figura 18.

#### 5. Descripción de las realizaciones

65 En la descripción que sigue, se considera un nodo de comprobación, que recibe como listas de entrada de  $n_m$  elementos ordenados en orden descendente o ascendente y que entrega como listas de salida de  $n'_m = n_m$  elementos ordenados análogamente en orden descendente o ascendente. Se considera análogamente que este

nodo de comprobación funciona en el campo logarítmico, los datos intercambiados entre nodos son entonces valores de LLR. Por supuesto,  $n'_m$  puede ser diferente de  $n_m$ .

5 Más específicamente, la invención se describirá en lo sucesivo más adelante con referencia a un nodo de comprobación que recibe, como mensajes entrantes,  $d_c$  entradas  $U_i$  y que entrega, como mensajes salientes,  $d_c$  salidas  $V_i$ , con  $i \in [1 \dots d_c]$ . Cada entrada  $U_i$  y salida  $V_i$  es una tupla (lista ordenada) de  $n_m$  valores de LLR cada uno asociado a un símbolo de  $GF(q)$ , con  $n_m < q$ , ordenándose los elementos de la tupla de acuerdo con un orden ascendente o descendente de sus valores de LLR. Los  $n_m$  símbolos de la tupla son los  $n_m$  símbolos más fiables (símbolos que tienen los valores de LLR más bajos) como se define en el algoritmo de EMS del estado de la técnica.  
10 Los símbolos (o elementos de campo de Galois) con el valor de fiabilidad más alto tienen un  $LLR=0$ .

A continuación,  $U_i[j]$  designa el elemento de orden  $j$  de la lista de entrada  $U_i$  y  $V_i[j]$  designa el elemento de orden  $j$  de la lista de salida  $V_i$ , con  $j \in [0 \dots n_m - 1]$ .

15 Antes de describir en detalle el método de la invención para un nodo de comprobación con  $d_c$  entradas y  $d_c$  salidas, se describe brevemente el principio del método inventivo con un nodo de comprobación como se ilustra por la Figura 1, con cuatro entradas  $U_1, U_2, U_3, U_4$  y cuatro salidas  $V_1, V_2, V_3, V_4$ .

20 Como se ha mencionado anteriormente, las cuatro salidas son como sigue:

$$\begin{cases} V_1 = U_2 + U_3 + U_4 \\ V_2 = U_1 + U_3 + U_4 \\ V_3 = U_1 + U_2 + U_4 \\ V_4 = U_1 + U_2 + U_3 \end{cases}$$

Puede reescribirse como sigue:

25

$$\begin{cases} V_1 = U_1 + U_2 + U_3 + U_4 - U_1 \\ V_2 = U_1 + U_2 + U_3 + U_4 - U_2 \\ V_3 = U_1 + U_2 + U_3 + U_4 - U_3 \\ V_4 = U_1 + U_2 + U_3 + U_4 - U_4 \end{cases}$$

en el que  $-$  es el operador de resta en el Campo de Galois  $GF(q)$  y para valores de LLR.

30 El principio básico de la invención es calcular en primer lugar la suma  $U_1+U_2+U_3+U_4$ , denominado síndrome, común a todas las salidas  $V_1, V_2, V_3, V_4$  antes de llevar a cabo la resta apropiada (descorrelación) para obtener las 4 salidas  $V_1, V_2, V_3, V_4$ . Permite hacer una multitud de operaciones en paralelo.

35 La Figura 2 representa un diagrama de flujo básico del método de la invención y la Figura 3 representa una arquitectura básica para implementar dicho método.

De acuerdo con la Figura 2, el método comprende las siguientes etapas:

- Etapa S1: añadir  $d_c$  elementos de entrada de listas de entrada  $U_i$ , para generar una pluralidad de sumas denominadas síndromes; cada uno de los elementos de entrada añadidos pertenece a una lista de entrada distintiva  $U_i$  entre las  $d_c$  listas de entrada; cada síndrome comprende un valor de fiabilidad que es la suma de los valores de fiabilidad de los elementos de entrada y un símbolo del campo de Galois que también es la suma de los símbolos de los elementos de entrada en el campo de Galois; los valores de fiabilidad son valores de LLR.
- 40
- Etapa S2: para cada lista de salida  $V_i$ , se aplica una descorrelación a los síndromes restando el elemento de entrada de la lista de entrada  $U_i$  de los síndromes para generar síndromes descorrelacionados, y
- 45
- Etapa S3: para cada lista de salida  $V_i$ , seleccionar, como elementos de salida de dicha lista de salida  $V_i$  los  $n_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos y generados para dicha lista de salida  $V_i$ .

50 En la etapa S1, se calcula una pluralidad de síndromes. El conjunto de síndromes se denomina  $S$ . Los síndromes individuales se distinguen por los elementos que se eligen para la suma. Esta etapa se implementa por el calculador de síndrome 10 representado en la Figura 3.

Si  $x_i$  designa un elemento  $U_i[j]$  de la lista de entrada  $U_i$ , con  $j \in [0..n_m]$  e  $i \in [1..d_c]$ , el síndrome generado a partir de los elementos de entrada  $(x_1 \dots x_{d_c})$ , denominado  $SYN(x_1 \dots x_{d_c})$ , es como sigue:

$$SYN(x_1 \dots x_{d_c}) = \{LLR(x_1 \dots x_{d_c}), GF(x_1 \dots x_{d_c})\}$$

5 con

$$\begin{cases} LLR(x_1 \dots x_{d_c}) = \sum_{t=1}^{d_c} LLR[x_t], \\ GF(x_1 \dots x_{d_c}) = \sum_{t=1}^{d_c} GF[x_t], \end{cases}$$

10 El conjunto de síndromes  $S$  que comprende todos los posibles síndromes comprende  $n_m^{d_c}$  síndromes y se define como sigue:

$$S = \{SYN(x_1 \dots x_{d_c}); \forall x_i \in U_i \text{ y } \forall i \in [1..d_c]\}$$

15 En la etapa S2, se aplica una descorrelación a los síndromes restando, para cada lista de salida  $V_i$  a generarse, el elemento de entrada de la lista de entrada  $U_i$  de los síndromes para generar síndromes descorrelacionados. Esta etapa se implementa por los descorreladores 11 representados en la Figura 3. El efecto técnico de esta etapa es la descorrelación de las salidas del nodo de comprobación de las entradas del nodo de comprobación.

20 Esta etapa consiste en generar un conjunto de síndromes especializados  $S^i$  para cada salida  $V_i$ , que no tiene correlación con la entrada  $U_i$ :

$$S^i = \{SYN^i(x_1 \dots x_{d_c}); \forall x_i \in U_i \text{ y } \forall i \in [1..d_c]\}$$

con

$$SYN^i(x_1 \dots x_{d_c}) = \{R^i(x_1 \dots x_{d_c}), G^i(x_1 \dots x_{d_c})\}$$

25 y

$$\begin{cases} LLR^i(x_1 \dots x_{d_c}) = LLR(x_1 \dots x_{d_c}) - LLR[x_i], \\ GF^i(x_1 \dots x_{d_c}) = GF(x_1 \dots x_{d_c}) - GF[x_i] \end{cases}$$

30 Cada conjunto  $S^i$  comprende  $n_m^{d_c-1}$  síndromes.

35 Una vez que se calculan los conjuntos  $S^i$ , los síndromes descorrelacionados en los conjuntos  $S^i$  se ordenan en la etapa S3 de acuerdo con su fiabilidad de síndrome representada por los valores de LLR. La etapa de ordenación se lleva a cabo por los ordenadores 12 representados en la Figura 3. Cada lista de salida  $V_i$  está constituida entonces por los  $n_m$  síndromes descorrelacionados más fiables del conjunto  $S^i$ .

40 Este método de procesamiento es una alternativa a los procesamientos de nodo de comprobación del estado de la técnica. Es el primer enfoque para la decodificación de campo de Galois de alto orden, que permite implementaciones paralelas masivas y por lo tanto alto rendimiento y baja latencia. Una vez que se calcula el conjunto de síndromes  $S$ , las etapas de descorrelación y las etapas de ordenación para cada salida  $V_i$  pueden ejecutarse en paralelo. El cálculo de síndrome puede hacerse también en paralelo. Permite tener procesamiento de baja latencia.

45 Sin embargo, sin tratamiento especial, el cálculo del conjunto de síndromes  $S$  y la ordenación de  $S^i$  introducen una alta complejidad. Tiene que reducirse para hacer el algoritmo atractivo para implementaciones de hardware. Para ese fin, se proponen diferentes mejoras en lo sucesivo. Más específicamente, se proponen diferentes enfoques para simplificaciones de la generación del conjunto de síndromes y la ordenación mientras se mantiene el rendimiento de las comunicaciones.

50 De acuerdo con una primera realización ventajosa, se reduce el número de síndromes del conjunto  $S$ . Para el cálculo de salida únicamente se usan los valores de  $S$  más fiables que hacen el cálculo de todos los demás síndromes superfluo. Por lo tanto una reducción inteligente de la cardinalidad de  $S$ , indicado  $|S|$ , puede reducir significativamente la complejidad global del algoritmo sin sacrificar el rendimiento de las comunicaciones.



La primera etapa para una reducción de  $|S|$  es la separación de síndromes con alta fiabilidad de los de baja fiabilidad. El conjunto de síndromes  $S$  puede definirse como la unión de  $d_c+1$  subconjuntos  $D_k$  (también denominado conjuntos de desviación), con  $k \in [0 \dots d_c]$ , de manera que:

$$S = \bigcup_{k=0}^{d_c} D_k$$

5 Cada subconjunto  $D_k$  contiene únicamente síndromes que se desvían en exactamente  $k$  elementos del elemento más fiable. La Figura 4 representa gráficamente dos síndromes ejemplares, uno con dos desviaciones (círculos llenos) y uno con tres desviaciones (círculos abiertos). Las desviaciones designan los elementos de entrada usados para generar el síndrome y no tienen un valor de LLR igual a cero (valor de fiabilidad más alto). El subconjunto  $D_0$  contiene únicamente un síndrome, que es la suma de los elementos más fiables de todas las entradas. El

subconjunto  $D_1$  contiene  $d_c - n_m$  síndromes. Más en general, el subconjunto  $D_k$  contiene  $\binom{d_c}{k} \cdot (n_m)^k$  síndromes.

15 La Figura 5 muestra los valores de LLR promedio de los síndromes en los conjuntos de desviación ordenados  $D_k$ . Puede observarse, que la distribución de LLR fiable depende de la Relación de Señal a Ruido (SNR) expresada por  $E_b/N_0$ . Sin embargo, los síndromes con más de dos desviaciones, por ejemplo,  $D_k$  para  $k > 2$  tienen una baja fiabilidad de este tipo (valor de LLR alto) que raramente pueden contribuir a la generación de las salidas  $V_i$ . Por lo tanto, ventajosamente, podemos limitar el cálculo de subconjuntos  $D_k$  a los de

$$|D_k| = \begin{cases} \binom{d_c}{k} \cdot (d_k)^k & \text{si } d_k \geq 1; k > 0 \\ 1 & \text{si } k = 0 \\ 0 & \text{sino} \end{cases}$$

20 Combinando ambas técnicas propuestas reduce estrictamente la cardinalidad de  $S$  y por lo tanto la complejidad computacional. Se calculan los síndromes más fiables y únicamente se eliminan los más poco fiables. La parametrización para el número de desviaciones y sus distancias de fiabilidad máxima es una etapa crítica en el algoritmo. Usar por ejemplo únicamente  $D_0, D_1,$  y  $D_2$  con distancias de fiabilidad fijas  $d_0=0; d_1=n_m-1; d_2=2, d_c=4$  y  $n_m=13$ , comprimen  $|S|$  de 28561 a 73. Para un código en GF(64) esto es un buen equilibrio entre complejidad y rendimiento de comunicaciones.

30 Otra manera de reducir la complejidad es simplificar la etapa de ordenación. Una gran desventaja del procesamiento presentado anteriormente en el presente documento es que cada conjunto de síndrome  $S^i$  debe ordenarse por separado para emitir los  $n_m$  síndromes descorrelacionados más fiables. Este es el caso debido a la etapa de descorrelación aplicada antes. Para evitar la ordenación de los conjuntos de síndromes descorrelacionados  $S^i$ , puede elegirse un enfoque sencillo pero eficaz. En lugar de descorrelacionar cada valor, únicamente se consideran los síndromes que usan el elemento más fiable (LLR = 0) del borde de salida actualmente manejado. Todos los otros síndromes no se usan para la salida actual  $V_i$ . Mediante este enfoque el orden de los síndromes no se cambia por la etapa de descorrelación y es suficiente ordenar un conjunto  $S$  en lugar de los  $d_c$  conjuntos  $S^i$ . Además, los valores de LLR no se modifican en la etapa de descorrelación que ahorra una resta con valor real para cada elemento en la salida  $V_i$ . Finalmente únicamente debe almacenarse el elemento de entrada más fiable y no los conjuntos de entrada completos para la baja cantidad de desviaciones de descorrelación. La Figura 5 muestra que considerar síndromes con una y dos desviaciones funciona bien para GF(64).

Por lo tanto, de acuerdo con una realización ventajosa, el conjunto  $S$  está limitado a la unión de  $D_0, D_1$  y  $D_2$ :

$$S = \bigcup_{k=0}^2 D_k$$

45 Otro parámetro para reducción de  $|S|$  es la distancia de fiabilidad máxima permitida  $d_k$  de elementos que contribuyen al conjunto de desviación  $D_k$ . La distancia de fiabilidad describe la posición del elemento en la lista de entrada con relación al elemento más fiable (LLR=0). En la Figura 4, el elemento más fiable tiene el índice cero. Elementos menos fiables tienen índices más altos que describen su ordenación en la lista ordenada de LLR.

50 Para el cálculo de  $D_k$  únicamente se consideran elementos con índices menores o iguales a  $d_k$ . La distancia de fiabilidad máxima permitida para una cierta desviación puede establecerse dinámicamente basándose en el valor de LLR de los elementos, o está fijada, como un parámetro predefinido. Para cada desviación puede establecerse una

distancia de fiabilidad máxima diferente, por ejemplo cuanto más alto es el número de desviaciones permitidas, menor es la distancia de fiabilidad máxima de las desviaciones,  $d_1 \geq d_2 \geq \dots \geq d_{d_c}$ . En la Figura 4,  $d_2$  se fija a 8 y  $d_3$  se fija a 2. La distancia de fiabilidad máxima  $d_1$  no se muestra en esta figura pero es igual a  $n_m - 1$ .

5 Usar este esquema mantiene implícitamente los mejores síndromes en cada  $D_k$  y elimina los menos fiables. La cardinalidad de los subconjuntos  $D_i$  puede calcularse como sigue:

10 Esta descorrelación simplificada permite también que se aplique para la etapa de ordenación antes de la etapa de descorrelación, como se ilustra por las Figuras 6 y 7. Reduce el número de ordenadores de  $d_c$  a 1. La Figura 6 muestra un diagrama de flujo de esta realización y la Figura 7 es una implementación de esta realización.

15 En la Figura 6, la operación de ordenación que se ejecutó en la etapa de generación de salida  $S_3$  en la Figura 2 se opera en este punto antes de la etapa de descorrelación  $S_3$ . La operación de ordenación, denominada  $S_{10}$ , se aplica en los síndromes del conjunto  $S$ . En la Figura 7, el nodo de comprobación comprende un calculador de síndrome 200 para generar el conjunto de síndromes  $S$ , un ordenador 220 para ordenar los síndromes de  $S$  de acuerdo con sus valores de LLR y  $d_c$  descorreladores 210 para aplicar la descorrelación a los conjuntos ordenados  $S^i$ .

20 La Figura 8 muestra las operaciones esquemáticas de un descorrelador de la Figura 7. En la Figura 8, cada síndrome se indica con una información adicional SRC acerca de la que los bordes de entrada contribuyen al síndrome con una desviación. SRC en la Figura 8 almacena los bordes de entrada donde tuvieron lugar las desviaciones y  $ADDR_i$  representa el borde de salida actual. Una comparación sencilla evalúa si una desviación del borde actual estaba implicada en el cálculo del síndrome y por lo tanto si el síndrome es válido o no para el borde actual. Únicamente si no tuvo lugar desviación en el borde actual, el síndrome descorrelacionado se marca como  
25 válido y se usa para la salida  $V_i$ .

Se ha de indicar que la expresión "borde" se usa en este punto de manera intercambiable con el término entrada ya que cada entrada corresponde a un borde en el grafo de Tanner.

30 Incluso aunque la ordenación se haya reducido al conjunto de síndromes  $S$ , existe más potencial para la simplificación. La ordenación  $S$  puede dividirse en ordenar los conjuntos de desviación  $D_i$  y unirlos. Especialmente para  $D_1$  la ordenación puede simplificarse adicionalmente. Esto se consigue debido al conocimiento anterior que tenemos de los datos de entrada. Conocemos implícitamente que las listas  $U_i$  se ordenan de acuerdo con sus LLR. La ordenación de  $D_1$  puede por lo tanto limitarse a unir  $d_c$  conjuntos ordenados. Para las desviaciones de orden superior  $D_i$ , para  $i \geq 2$ , la ordenación puede simplificarse también debido a las listas de entrada ordenadas. Un ejemplo del circuito para ordenar los síndromes de  $D_1$  y  $D_2$ , con  $d_c=4$ , se muestra en la Figura 9. La parte P1 se usa para ordenar los síndromes de  $D_1$  y la parte P2 se usa para ordenar los síndromes de  $D_2$  con  $d_2=2$ .

40 Como puede observarse a partir de la Figura 9 y como se ha mencionado, la ordenación de los síndromes de  $D_1$  se simplifica puesto que las listas  $U_i$  ya se han ordenado de acuerdo con sus LLR y el LLR del síndrome es el LLR del elemento de entrada que no tiene un LLR igual a cero.

45 Para los síndromes de  $D_2$  (parte P2 de la Figura 9), las sublistas ordenadas tienen que generarse en primer lugar por seis circuitos  $SC_p$  que generan subconjuntos de elementos de entrada ordenados, un circuito  $CS_1$  para generar una sublista ordenada de los elementos de entrada  $U_1[1]$ ,  $U_1[2]$ ,  $U_2[1]$  y  $U_2[2]$ , un circuito  $CS_2$  para generar una sublista ordenada de los elementos de entrada  $U_1[1]$ ,  $U_1[2]$ ,  $U_3[1]$  y  $U_3[2]$ , un circuito  $CS_3$  para generar una sublista ordenada de los elementos de entrada  $U_1[1]$ ,  $U_1[2]$ ,  $U_4[1]$  y  $U_4[2]$ , un circuito  $CS_4$  para generar una sublista ordenada de los elementos de entrada  $U_2[1]$ ,  $U_2[2]$ ,  $U_3[1]$  y  $U_3[2]$ , un circuito  $CS_5$  para generar una sublista ordenada de los elementos de entrada  $U_2[1]$ ,  $U_2[2]$ ,  $U_4[1]$  y  $U_4[2]$ , y un circuito  $CS_6$  para generar una sublista ordenada de los  
50 elementos de entrada  $U_3[1]$ ,  $U_3[2]$ ,  $U_4[1]$  y  $U_4[2]$ .

55 La Figura 10 muestra el circuito  $CS_1$  usado para generar la sublista ordenada de los elementos de entrada  $U_1[1]$ ,  $U_1[2]$ ,  $U_2[1]$  y  $U_2[2]$ . El circuito de la Figura 10 puede extenderse fácilmente a otras desviaciones y distancias de fiabilidad. Una vez que se ordenan las sublistas, pueden generarse las salidas uniéndolas secuencialmente como se muestra en la Figura 9.

En vista de la memoria descriptiva anterior, surgen tres beneficios notables a partir del método de acuerdo con la invención:

- 60
- Reducción significativa de  $|S|$ .
  - No restas de LLR y no almacenamiento para  $U_i$  en la etapa de descorrelación.

65 El método de la invención puede simplificarse adicionalmente. Considerando el decodificador de NB-LDPC como una totalidad, puede observarse, que no se requiere una ordenación exacta de las salidas del nodo de comprobación. Cuando un nodo variable ha calculado los mensajes de probabilidad a posteriori (APP) como la suma de los valores de canal y mensajes desde los nodos de comprobación, tienen que reordenarse de todas maneras.

Por lo tanto es suficiente una salida de nodo de comprobación aproximadamente ordenada y que no perjudica el rendimiento de comunicaciones del decodificador. Por lo tanto, se propone en lo sucesivo un nuevo método que usa la robustez frente a las salidas de nodo de comprobación ordenadas para reducir adicionalmente la complejidad de los algoritmos.

5 Para permitir esta ordenación aproximada, de modo que se elijan denominados sondas entre los elementos de las listas de entrada  $U_i$  y se ordenen de acuerdo con su LLR. La LLR de cada sonda se considera como representativa de las LLR del grupo de  $p$  elementos vecinos que incluyen dicha sonda. La Figura 11 muestra la distribución de sondas en la lista de entrada  $U_1$  usada para  $D_1$ .

10 En este ejemplo, las sondas se distribuyen igualmente en las listas de entrada que funcionarán bien para la configuración investigada. Las sondas consideradas son los elementos indexados 1, 4, 7 y 10. Sin embargo, para otros parámetros de  $d_c$  y  $n_m$  pueden usarse otras distribuciones.

15 En esta figura, la sonda es el elemento de entrada que tiene el valor de fiabilidad más bajo en el grupo de  $p$  elementos de entrada vecinos que comprenden la sonda y el valor de fiabilidad de la sonda es el valor de fiabilidad más bajo de los elementos de este grupo. En el ejemplo de la Figura 11, el valor de LLR de la sonda 1 ( $=U_1[3]$ ) se considera como representativo de los valores de LLR de todos los elementos del grupo sonda1\_grupo que comprende los elementos  $U_1[1]$ ,  $U_1[2]$  y  $U_1[3]$ . Por supuesto, el valor de LLR de la sonda también puede ser una combinación de los valores de LLR de los elementos del grupo, por ejemplo el valor promedio de los valores de LLR de los elementos del grupo.

25 En esta realización, una vez que se seleccionan las sondas, se evalúan para seleccionar un número reducido de sondas ordenadas, y a continuación los síndromes en el conjunto  $S$  (que es por ejemplo  $D_0 \cup D_1 \cup D_2$  con  $d_2=2$ ) se seleccionan basándose en estas sondas ordenadas, ordenándose los síndromes seleccionados en el orden de las sondas ordenadas.

El diagrama de flujo de una realización de este tipo se ilustra por Figura 12. El método comprende las siguientes etapas:

- 30
- Etapa S200: preseleccionar sondas en las listas de entrada  $U_i$  como se ha descrito anteriormente; esta preselección puede ser una predefinida como se describe en la Figura 11; en esta figura, los elementos  $U_i[j]$  con  $i=1;2;3;4$  y  $j=3;6;9;12$  se preseleccionan; por lo que se preseleccionan 16 sondas;
  - 35 - Etapa S210: calcular síndromes para generar el conjunto de síndromes  $S$ ;
  - Etapa S220: evaluar las sondas preseleccionadas para seleccionar y ordenar un número predeterminado reducido de las sondas preseleccionadas, dichas sondas, denominadas sondas finales, ordenándose de acuerdo con los valores de fiabilidad; la etapa S220 puede llevarse a cabo antes o después o en paralelo con la etapa S210;
  - 40 - Etapa S230: seleccionar síndromes a partir del conjunto  $S$  basándose en las sondas finales; los síndromes seleccionados se ordenan en el orden de las sondas finales; y
  - Etapa S240: aplicar una descorrelación a los síndromes ordenados para generar las listas de salida  $V_i$ .

45 Este esquema conduce a alguna incertidumbre en el conjunto de síndromes emitidos a partir de la etapa S230 pero es lo suficientemente cercano a la solución exacta para no degradar el rendimiento de las comunicaciones de los decodificadores.

La Figura 13 muestra los valores de LLR de los síndromes aproximadamente ordenados usados para el cálculo de salida de CN. La aproximación funciona bien para LLR bajos (alta fiabilidad) y únicamente para las salidas menos fiables existe una diferencia en comparación con los resultados de ordenación óptimos.

50 A continuación, se proporciona una implementación de hardware de la solución inventiva con uso de sondas. La arquitectura es independiente del código de NB-LDPC real usado, únicamente se proporcionan los parámetros  $d_c = 4$ ,  $q = 64$  y  $n_m = 13$ .

55 La Figura 14 muestra una vista general del hardware del nodo de comprobación. Para conseguir baja latencia y alto rendimiento, el paralelismo de entrada del nodo de comprobación se elige para que sea seis GF(q), tuplas de LLR y una entrada de GF(q) adicional para el elemento más fiable (LLR=0). Todos los elementos de entrada  $U_i[j]$  pueden leerse por lo tanto en dos ciclos de reloj. La evaluación de las sondas puede procesarse en paralelo con el cálculo real del conjunto de síndromes  $S$ . En este ejemplo, el conjunto de síndromes  $S$  se reduce a  $D_0 \cup D_1 \cup D_2$  con  $d_2=2$ .  $D_0$  incluye 1 síndrome,  $D_1$  incluye 48 síndromes y  $D_2$  incluye 24 síndromes.

El nodo de comprobación comprende

- 65
- un evaluador de sonda PE1 para determinar un número reducido de sondas usadas para seleccionar síndromes de  $D_1$ ,

- un evaluador de sonda PE2 para determinar un número reducido de sondas usadas para seleccionar síndromes de  $D_2$ ,
- un calculador de síndrome SC para calcular los síndromes de  $D_0$ ,  $D_1$  y  $D_2$ ,
- un selector de síndrome SS1 para seleccionar, entre los síndromes de  $D_1$  generados por el calculador de síndrome SC, 6 síndromes;
- un selector de síndrome SS2 para seleccionar, entre los síndromes de  $D_2$  generados por el calculador de síndrome SC, 3 síndromes; y
- $d_c$  descorreladores que reciben el síndrome de  $D_0$  y los síndromes de  $D_1$  y  $D_2$  seleccionados por los selectores de síndrome SS1 y SS2.

En esta realización, una vez que se calcula  $S=D_0 \cup D_1 \cup D_2$  y se ordenan las sondas, los subconjuntos más fiables seleccionados por los selectores de síndrome SS1 y SS2 se usan para la descorrelación. El paralelismo con el que se procesan los síndromes tiene un impacto significativo en el rendimiento global. Se ha elegido que sean tres veces tres síndromes. En cada ciclo de reloj se procesan dos conjuntos de síndromes vecinos de  $D_1$  y un conjunto de  $D_2$ , en un total de nueve síndromes. Por lo tanto, después de un máximo de cuatro ciclos de reloj todos los bordes de salida se rellenan con  $n_m$  mensajes válidos. El paralelismo de salida del CN se elige simétricamente para las entradas para que sean seis GF(q), tuplas de LLR y un mensaje de GF(q) para  $D_0$ .

Un ejemplo del evaluador de sonda PE1 se ilustra por la Figura 15. Este evaluador de sondas procesa dos sondas por borde de entrada en cada ciclo de reloj. Para  $d_c = 4$  tienen que ordenarse un total de ocho LLR, que se realiza mediante una red de ordenación de latencia optimizada. Ya que cada dos sondas están ya ordenadas implícitamente puesto que pertenecen a la misma lista de entrada, pueden aplicarse algunas simplificaciones en la red.

El evaluador de sonda PE1 recibe las sondas  $U_1[1]$ ,  $U_1[4]$ ,  $U_2[1]$ ,  $U_2[4]$ ,  $U_3[1]$ ,  $U_3[4]$ ,  $U_4[1]$ ,  $U_4[4]$  en un primer ciclo y las sondas  $U_1[7]$ ,  $U_1[10]$ ,  $U_2[7]$ ,  $U_2[10]$ ,  $U_3[7]$ ,  $U_3[10]$ ,  $U_4[7]$ ,  $U_4[10]$  en un segundo ciclo. Estas sondas se procesan por una red de ordenación. El resultado de la ordenación no es una lista ordenada de LLR, sino en su lugar las posiciones de las entradas desde donde provenían. Se ordenan en un registro y la misma tarea se realiza una segunda vez para la segunda mitad de las LLR de entrada en el siguiente ciclo de reloj. Empezando desde el segundo ciclo de reloj, cada siguiente ciclo de reloj se emiten las posiciones de las dos sondas más pequeñas. Para realizar esta tarea, se utiliza un ordenador adicional, que selecciona las dos sondas más pequeñas a partir de los registros. Una vez que se usa una sonda para una generación de salida, se elimina desplazando el contenido del registro en consecuencia. El evaluador de sonda PE2 para  $D_2$  es una versión simplificada de el de para  $D_1$  ya que considera únicamente cuatro entradas. Además necesita generar únicamente una salida por ciclo de reloj. La salida del evaluador de sonda se usa como componentes de señal de control para el selector de síndrome.

La Figura 16 representa parcialmente un ejemplo del calculador de síndrome SC. El cálculo de síndrome se lleva a cabo completamente en paralelo. Debido a las restricciones en las distancias de fiabilidad de desviación ( $d_0=0$ ,  $d_1=n_m-1$ ,  $d_2=2$ ), el hardware requerido está estrictamente limitado. Un esquema sofisticado para el cálculo permite coste de hardware reducido adicional. En lugar de calcular cada síndrome como una suma de  $d_c$  entradas, se usan resultados intermedios para minimizar el número de cálculos explícitos. El cálculo de  $D_0$  implica  $d_c-1$  GF(q) adiciones, para  $D_1$  se requiere un total de  $d_c-n_m$  GF(q) adiciones. Por lo tanto para  $D_0$  y  $D_1$  únicamente se requieren GF(q) cálculos. Para  $D_2$  el esquema de procesamiento de  $D_1$  puede extenderse fácilmente y requiere únicamente una adición con valor real por síndrome además de las operaciones de GF(q). En comparación con el procesamiento del estado de la técnica, existe un ahorro significativo en complejidad computacional. La salida del calculador de síndrome es un número de subconjuntos ordenados usados como entrada para los selectores de síndrome.

La Figura 17 representa parcialmente un ejemplo del selector de síndrome SS1 para  $D_1$ . El selector de síndrome recibe las direcciones de los conjuntos de síndrome más fiables desde el evaluador de sonda PE1 y los síndromes  $D_i$  ordenados en subconjuntos desde el calculador de síndrome. El hardware completo consiste en un árbol multiplexor que elige un conjunto de los tres síndromes más fiables de los subconjuntos. Para conseguir seis mensajes de  $D_1$ , el árbol multiplexor de la Figura 17 se ha de duplicar en esta unidad. Para  $D_2$  es suficiente que un único árbol elija entre los seis subconjuntos de  $D_2$ .

La descorrelación tiene que realizarse individualmente para cada borde de salida del nodo de comprobación. El paralelismo de salida del descorrelador es seis mensajes por ciclo de reloj. Se procesan dos veces tres síndromes desde  $D_1$  y otros tres desde  $D_2$  por ciclo de reloj. Por construcción, los mensajes de un conjunto siempre tienen desviaciones en los mismos bordes. Por lo tanto es suficiente comprobar uno de los mensajes en un conjunto si es válido o no, que se indica con una bandera de validez. Si únicamente una parte de los conjuntos recibidos es válida, se reorganizan por multiplexores de tal manera que, únicamente se usan mensajes válidos para la salida. En el mejor caso todos los síndromes recibidos en un ciclo de reloj son válidos. Como el paralelismo de salida es únicamente seis, los síndromes en exceso se almacenan en un registro adicional y se reutilizan en el siguiente ciclo de reloj. Antes de que se envíen los mensajes al nodo variable, se aplica la descorrelación real que es una resta del valor de GF(q) más fiable del borde de entrada actual.

En la implementación de hardware de la Figura 14, todos los síndromes de  $D_1$  y  $D_2$  se generan por el calculador de síndrome SC y únicamente se seleccionan nueve de ellos por los selectores de síndrome SS1 y SS2.

5 En una variante ilustrada por la Figura 18 y la Figura 19, se propone únicamente generar los síndromes que se descorrelacionarán. En esta variante, las señales de control ctrl generadas por los evaluadores de sonda se transmiten al calculador de síndrome SC de manera que el último únicamente genera el síndrome de  $D_0$  y los nueve síndromes de  $D_1$  y  $D_2$  a descorrelacionar. Los selectores de síndrome SS1 y SS2 se eliminan.

10 El método ejecutado por el nodo de comprobación en esta variante puede resumirse en las siguientes etapas representadas la Figura 19:

- Etapa S300: preseleccionar sondas en las listas de entrada  $U_i$  como se ha descrito anteriormente,
- Etapa S310: evaluar las sondas para seleccionar y ordenar un número predeterminado de las sondas preseleccionadas, denominadas sondas finales, ordenándose dichas sondas de acuerdo con los valores de fiabilidad;
- 15 - Etapa S320: calcular un número reducido de síndromes basándose en las sondas finales; y
- Etapa S340: aplicar una descorrelación a los síndromes ordenados para generar las listas de salida  $V_i$ .

## REIVINDICACIONES

1. Método para controlar un nodo de comprobación de un decodificador para decodificar códigos de LDPC no binarios, recibiendo dicho nodo de comprobación  $d_c$  listas de entrada  $U_i$  de  $n_m$  elementos ( $U_{i[j]}$ ) y entregando  $d_c$  listas de salida  $V_i$  de  $n'_m$  elementos ( $V_{i[j]}$ ), con  $i \in [1 \dots d_c]$ , con  $d_c > 2$ , cada elemento de las listas de entrada o de salida, denominado respectivamente elemento de entrada y de salida, que comprende un valor de fiabilidad ( $LLR(U_{i[j]})$ ,  $LLR(V_{i[j]})$ ) asociado a un símbolo ( $GF(U_{i[j]})$ ,  $GF(V_{i[j]})$ ) de un Campo de Galois  $GF(q)$  con  $q > n_m$  y  $q > n'_m$ , ordenándose sustancialmente los elementos de entrada y elementos de salida de acuerdo con los valores de fiabilidad respectivamente en dicha lista de entrada y lista de salida, estando dicho método caracterizado por que incluye las siguientes etapas:
- añadir  $d_c$  elementos de entrada de las listas de entrada  $U_i$  para generar una pluralidad de sumas denominadas síndromes, perteneciendo cada uno de dichos elementos de entrada a una lista de entrada distintiva entre las  $d_c$  listas de entrada  $U_i$  y comprendiendo cada síndrome un valor de fiabilidad que es la suma de los valores de fiabilidad de dichos elementos de entrada y un símbolo del campo de Galois que es la suma de los símbolos de dichos elementos de entrada en el campo de Galois,
  - para cada lista de salida  $V_i$ , aplicar una descorrelación a los síndromes restando el elemento de entrada de la lista de entrada  $U_i$ , de los síndromes para generar síndromes descorrelacionados, y
  - para cada lista de salida  $V_i$ , seleccionar, como elementos de salida de dicha lista de salida  $V_i$ , los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos y generados para dicha lista de salida  $V_i$ .
2. Método de acuerdo con la reivindicación 1, en el que, en la etapa de adición, cada síndrome se genera basándose en como máximo  $k$  elementos de entrada diferentes de los elementos de entrada que tienen los valores de fiabilidad más altos, con  $k < d_c$ .
3. Método de acuerdo con la reivindicación 1 o 2, en el que, en la etapa de adición, cada síndrome se genera basándose en elementos de entrada cuya distancia, denominada distancia de fiabilidad, de los elementos de entrada que tienen los valores de fiabilidad más altos es inferior a una distancia de fiabilidad máxima.
4. Método de acuerdo con las reivindicaciones 2 y 3, en el que dicha distancia de fiabilidad máxima depende de  $k$ .
5. Método de acuerdo con una cualquiera de las reivindicaciones anteriores, en el que los elementos de salida de la lista de salida  $V_i$  se seleccionan ordenando los síndromes descorrelacionados generados para la lista de salida  $V_i$  de acuerdo con los valores de fiabilidad y seleccionando los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos.
6. Método de acuerdo con una cualquiera de las reivindicaciones 1 a 4, en el que, para una lista de salida  $V_i$  a generarse, la descorrelación se aplica a síndromes generados del elemento de entrada de la lista de entrada  $U_i$  que tienen los valores de fiabilidad más altos.
7. Método de acuerdo con la reivindicación 6, en el que, antes de la etapa de descorrelación, los síndromes se ordenan de acuerdo con los valores de fiabilidad de dichos síndromes de manera que, después de la etapa de descorrelación, los síndromes descorrelacionados generados para la lista de salida  $V_i$  se ordenan de acuerdo con los valores de fiabilidad, y los elementos de la lista de salida  $V_i$  son los  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos.
8. Método de acuerdo con una cualquiera de las reivindicaciones 1 a 4, en el que, antes o después o en paralelo con la generación de síndrome, el método comprende adicionalmente las etapas de:
- preseleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,
  - evaluar dichas sondas preseleccionadas para seleccionar y ordenar un número predeterminado de las sondas preseleccionadas, dichas sondas, denominadas sondas finales, ordenándose de acuerdo con los valores de fiabilidad; y
  - seleccionar síndromes generados basándose en dichas sondas finales, ordenándose dichos síndromes en el orden de las sondas finales.
9. Método de acuerdo con una cualquiera de las reivindicaciones 1 a 4, en el que, antes de la generación de síndrome, el método comprende adicionalmente las etapas de:
- preseleccionar el elemento de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda, y
  - evaluar dichas sondas preseleccionadas para seleccionar y ordenar un número predeterminado de las sondas preseleccionadas, dichas sondas, denominadas sondas finales, ordenándose de acuerdo con los valores de fiabilidad;

y en el que, en la etapa de generación de síndrome, los síndromes se generan basándose en dichas sondas finales, ordenándose dichos síndromes en el orden de las sondas finales.

5 10. Método de acuerdo con la reivindicación 8 o 9, en el que la sonda es el elemento de entrada que tiene el valor de fiabilidad más alto en el grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda y el valor de fiabilidad de la sonda es dicho valor de fiabilidad más alto.

10 11. Método de acuerdo con la reivindicación 8 o 9, en el que el valor de fiabilidad de la sonda es una combinación de los valores de fiabilidad de los  $p$  elementos de entrada vecinos.

12. Nodo de comprobación de un decodificador para decodificar códigos de LDPC no binarios que incluye:

- $d_c$  entradas para recibir  $d_c$  listas de entrada  $U_i$  de  $n_m$  elementos ( $U_{i[j]}$ ), denominados elementos de entrada, con  $i \in [1 \dots d_c]$ ,  $n_m > 1$  y  $d_c > 2$ , comprendiendo cada elemento de entrada un valor de fiabilidad ( $LLR(U_{i[j]})$ ) asociado a un símbolo ( $GF(U_{i[j]})$ ) de un Campo de Galois  $GF(q)$  con  $q > n_m$ , ordenándose sustancialmente los elementos de entrada de acuerdo con los valores de fiabilidad en dicha lista de entrada,
- $d_c$  salidas para entregar  $d_c$  listas de salida  $V_i$  de  $n'_m$  elementos ( $V_{i[j]}$ ) denominados elementos de salida, con  $q > n'_m$ , comprendiendo cada elemento de salida un valor de fiabilidad ( $LLR(V_{i[j]})$ ) asociado a un símbolo ( $GF(V_{i[j]})$ ) de un Campo de Galois  $GF(q)$ , ordenándose sustancialmente los elementos de salida de acuerdo con los valores de fiabilidad en dicha lista de salida,

caracterizado por que incluye adicionalmente:

- un calculador de síndrome para añadir  $d_c$  elementos de entrada de listas de entrada  $U_i$  para generar una pluralidad de sumas denominadas síndromes, perteneciendo cada uno de dicho elemento de entrada a una lista de entrada distintiva entre las  $d_c$  listas de entrada  $U_i$  y comprendiendo cada síndrome un valor de fiabilidad que es la suma de los valores de fiabilidad de dichos elementos de entrada y un símbolo del campo de Galois que es la suma de los símbolos de dichos elementos de entrada en el campo de Galois,
- $d_c$  descorreladores para aplicar, para cada lista de salida  $V_i$ , una descorrelación a los síndromes restando el elemento de entrada de la lista de entrada  $U_i$  de los síndromes para generar síndromes descorrelacionados, y
- medios para seleccionar, para cada lista de salida  $V_i$ , como elementos de salida para dicha lista de salida  $V_i$ ,  $n'_m$  síndromes descorrelacionados que tienen los valores de fiabilidad más altos y generados para dicha lista de salida  $V_i$ .

35 13. Nodo de comprobación de acuerdo con la reivindicación 12, en el que comprende  $d_c$  ordenadores para ordenar los síndromes descorrelacionados de acuerdo con los valores de fiabilidad, estando especializado cada uno de dichos ordenadores para ordenar los síndromes descorrelacionados generados para una lista de salida especializada  $V_i$ .

40 14. Nodo de comprobación de acuerdo con la reivindicación 12, en el que comprende un ordenador para ordenar los síndromes generados por el calculador de síndrome de acuerdo con los valores de fiabilidad, aplicándose la descorrelación, para una lista de salida  $V_i$  a generarse, a síndromes generados del elemento de entrada de la lista de entrada  $U_i$  que tienen los valores de fiabilidad más altos.

45 15. Nodo de comprobación de acuerdo con la reivindicación 12, en el que comprende adicionalmente:

- un selector de sonda para seleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,
- un ordenador de sonda para ordenar las sondas de acuerdo con los valores de fiabilidad, y
- un selector de síndrome que selecciona síndromes generados basándose en las sondas ordenadas, ordenándose dichos síndromes en el orden de las sondas ordenadas.

55 16. Nodo de comprobación de acuerdo con la reivindicación 12, en el que comprende adicionalmente:

- un selector de sonda para seleccionar elementos de entrada, denominados sondas, en las listas de entrada  $U_i$ , teniendo cada sonda un valor de fiabilidad representativo para un grupo de  $p$  elementos de entrada vecinos que comprende dicha sonda,
- un ordenador de sonda para ordenar las sondas de acuerdo con los valores de fiabilidad,

60 y en el que el calculador de síndrome se acciona para generar síndromes basándose en las sondas ordenadas, ordenándose dichos síndromes en el orden de las sondas ordenadas.

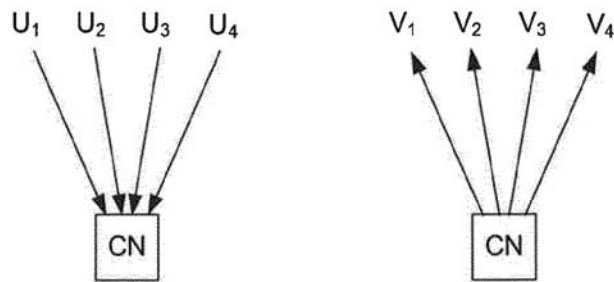


FIG.1

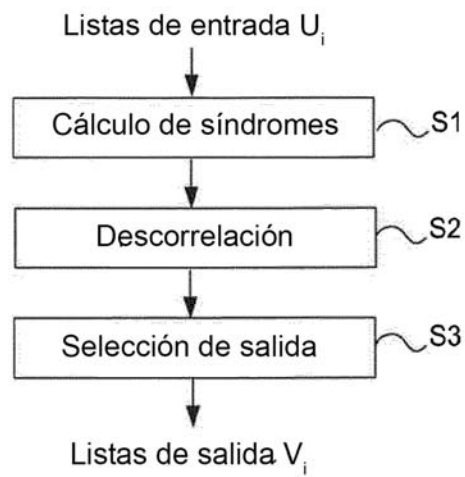


FIG.2

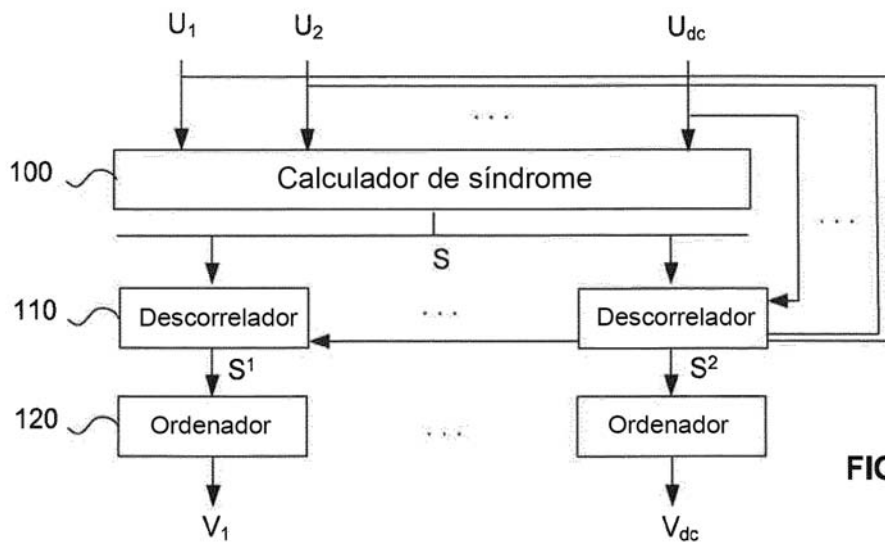


FIG.3



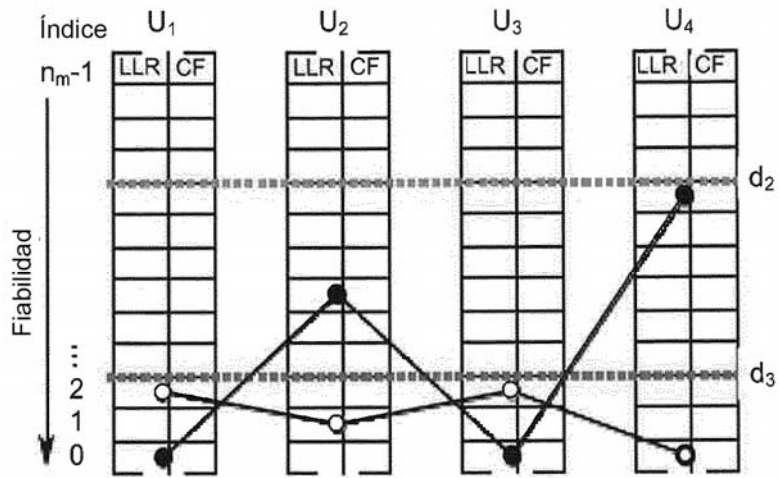


FIG.4

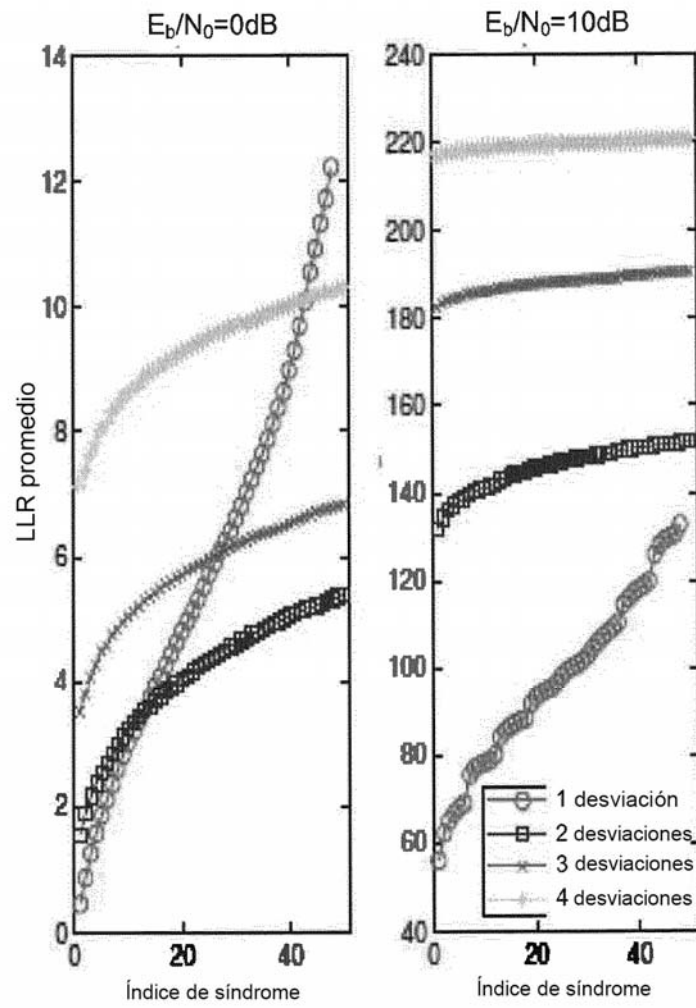


FIG.5

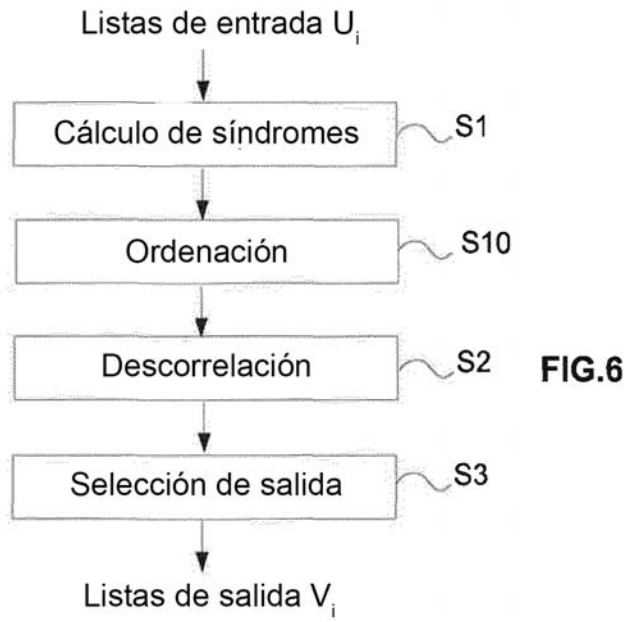


FIG.6

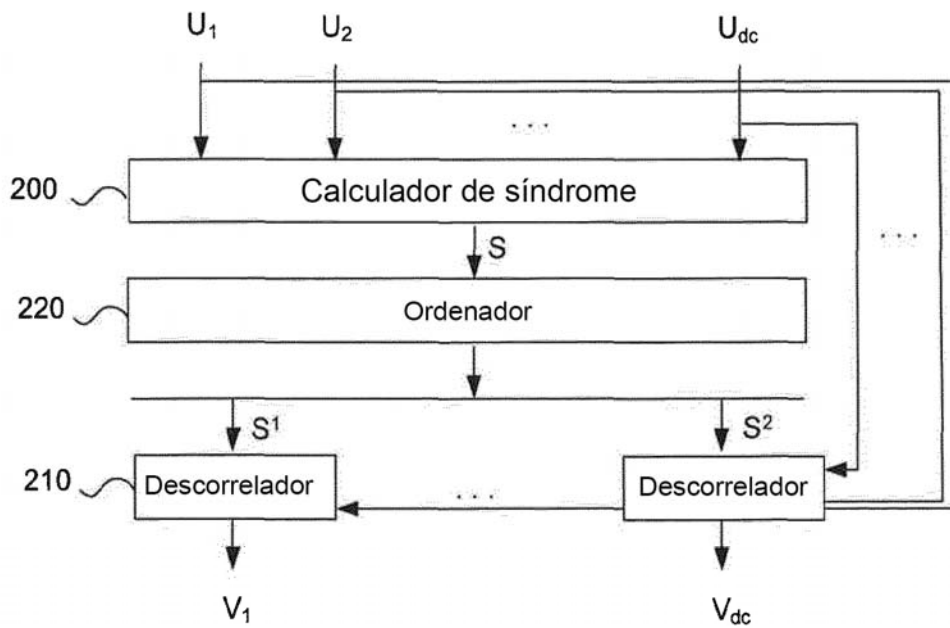


FIG.7

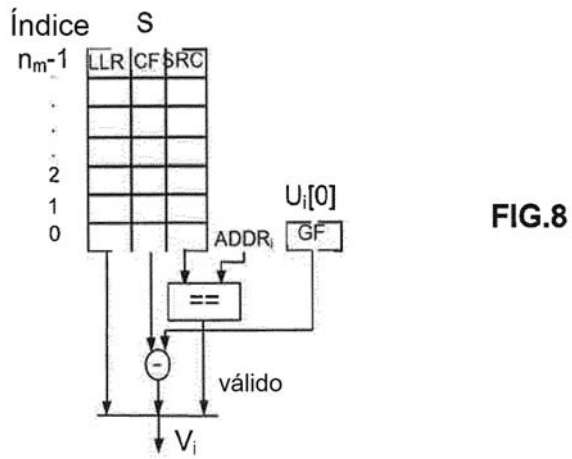


FIG. 8

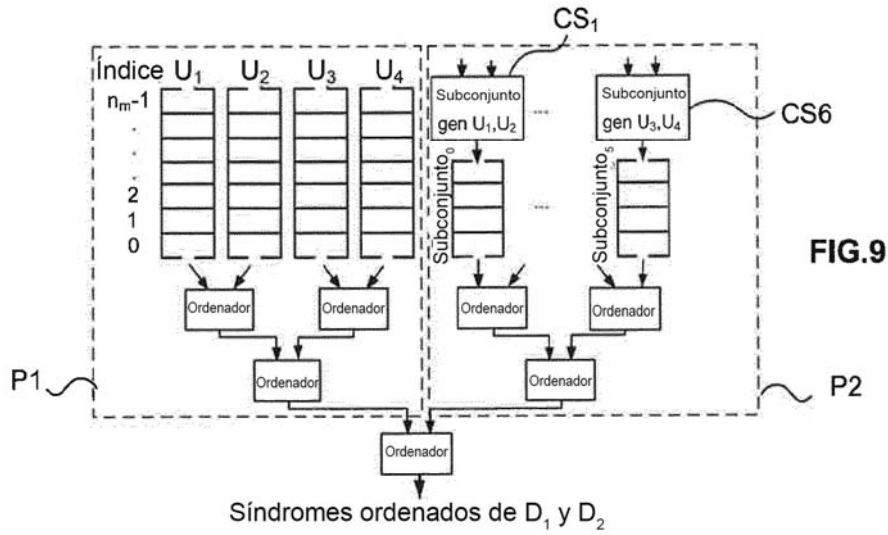


FIG. 9

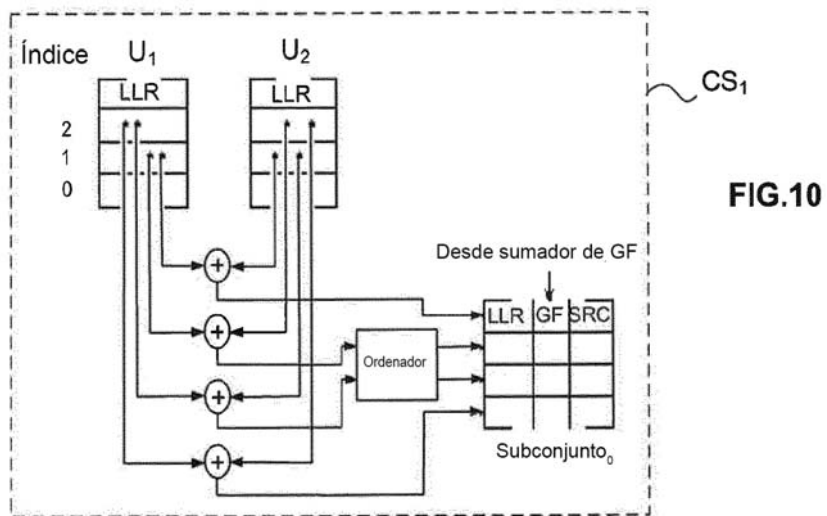


FIG. 10

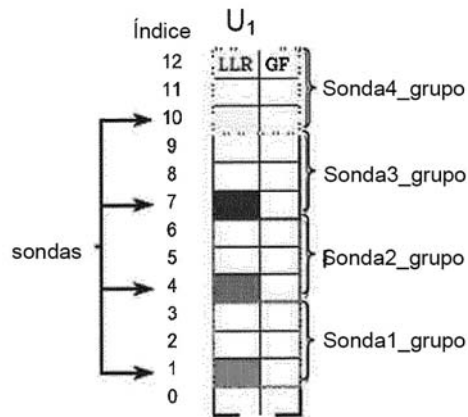


FIG.11

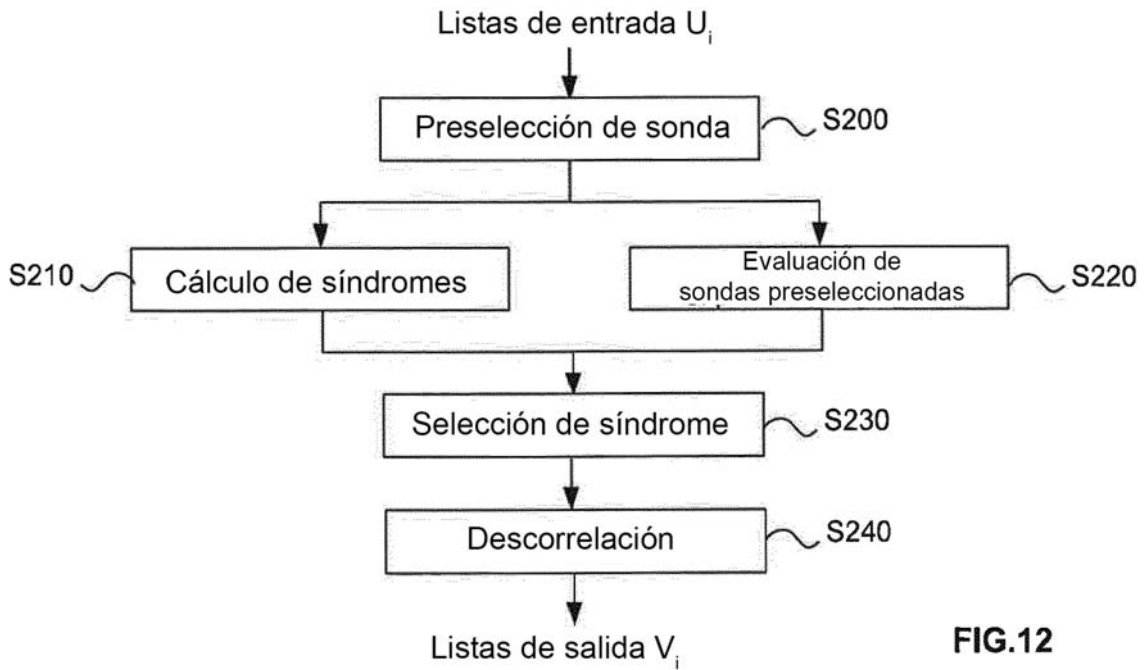


FIG.12

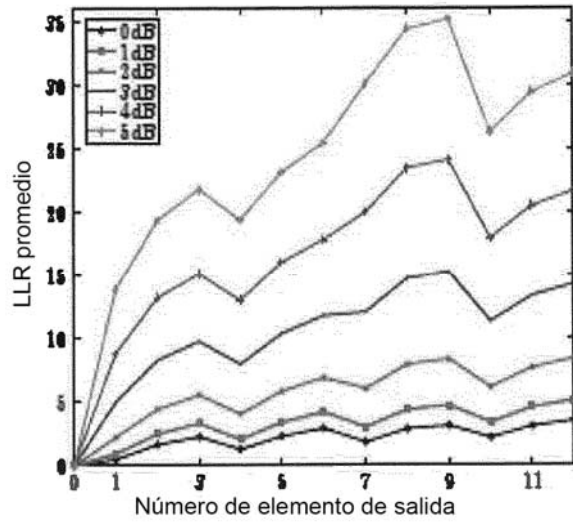


FIG.13

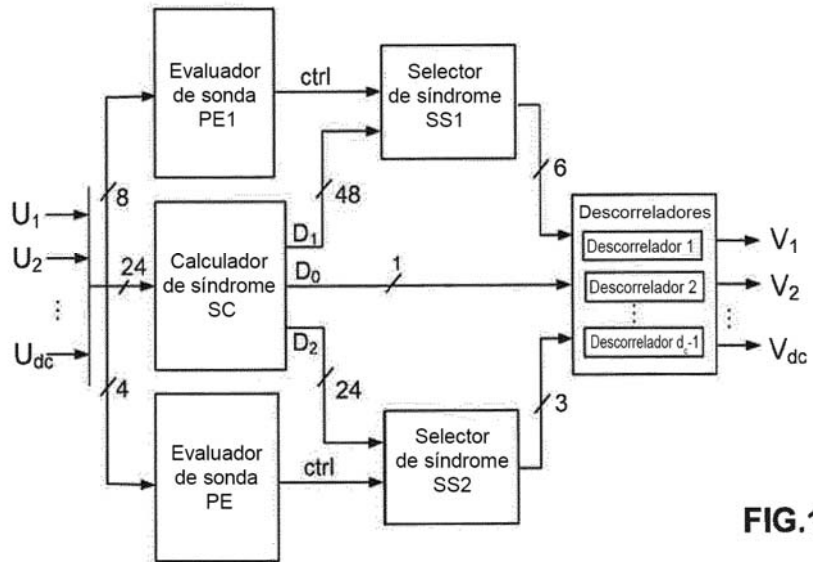


FIG.14

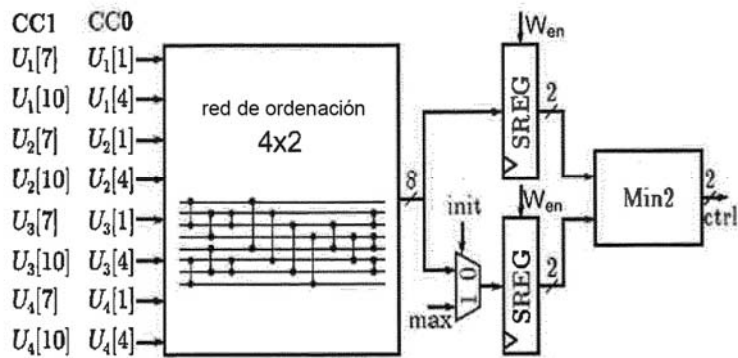


FIG.15

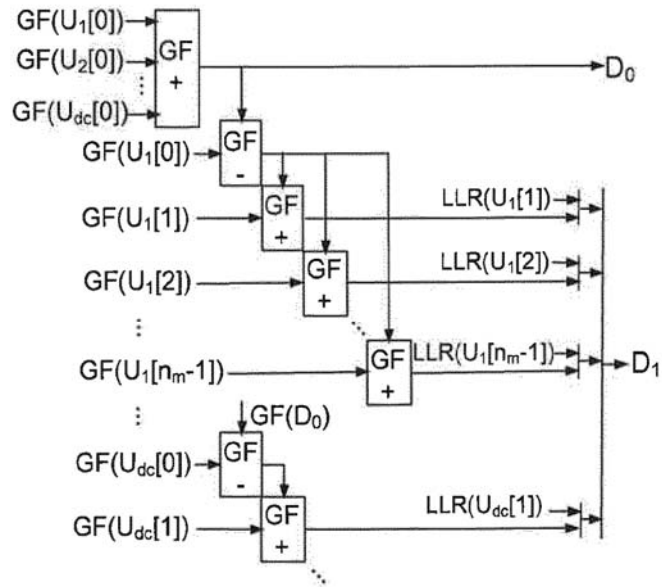


FIG. 16

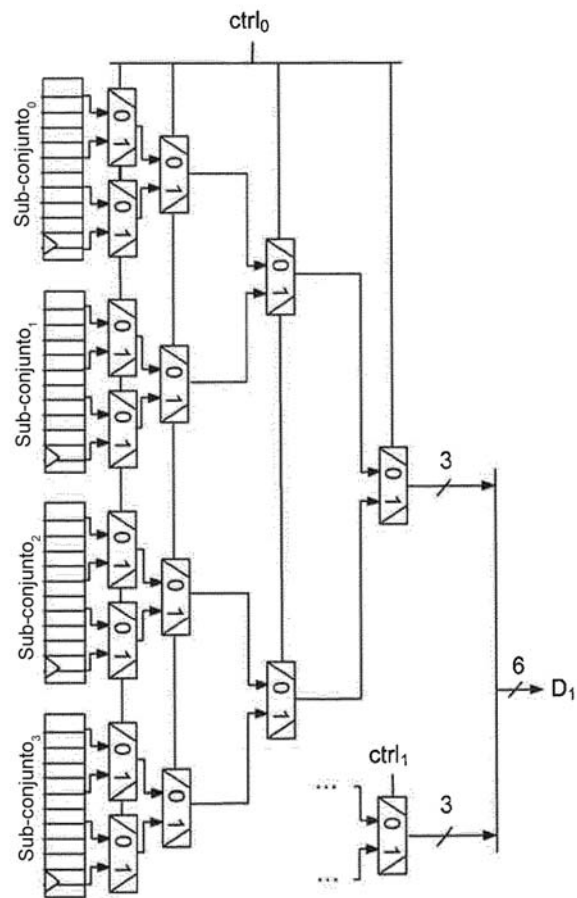


FIG. 17

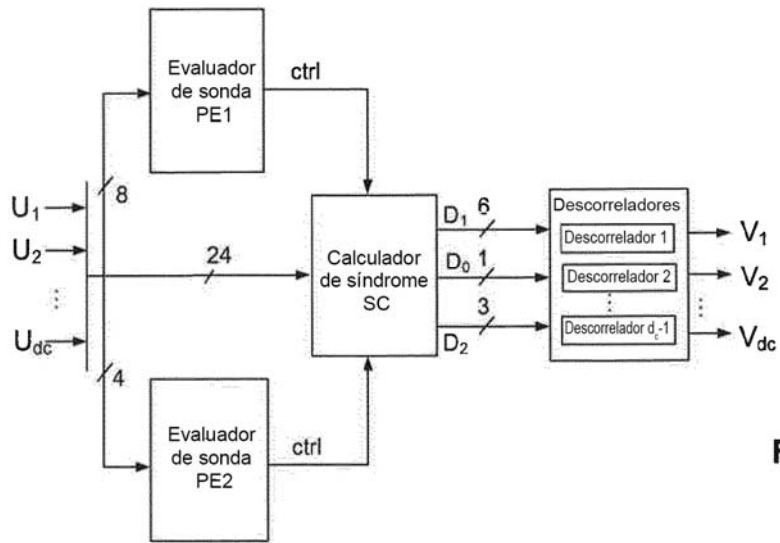


FIG.18

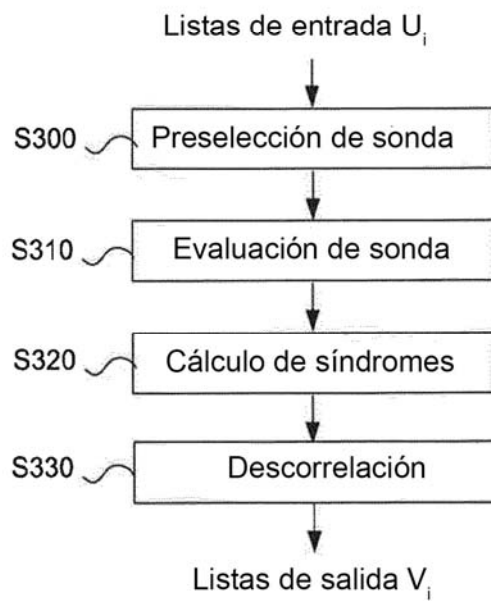


FIG.19