

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 421 942**

51 Int. Cl.:

H03M 13/11 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.08.2005** **E 11177322 (2)**

97 Fecha y número de publicación de la concesión europea: **10.07.2013** **EP 2387157**

54 Título: **Codificación eficiente de códigos LDPC usando matrices de verificación de paridad estructuradas**

30 Prioridad:

09.08.2004 US 600005 P
03.12.2004 US 4359

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
06.09.2013

73 Titular/es:

MOTOROLA MOBILITY LLC (100.0%)
600 North US Highway 45
Libertyville, IL 60048, US

72 Inventor/es:

BLANKENSHIP, YUFEI W.;
CLASSON, BRIAN K. y
BLANKENSHIP, T KEITH

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 421 942 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 eficiente de códigos LDPC usando matrices de verificación de paridad estructuradas

5 Campo de la invención

La presente invención se refiere en general a codificar y decodificar datos y, en particular, a un método y aparato para codificar y decodificar datos utilizando códigos de verificación de paridad de baja densidad (LDPC).

10 Antecedentes de la invención

Como se describe en la Solicitud de Patente de Estados Unidos número de serie 10/839995, un código de verificación de paridad de baja densidad (LDPC) es un código de bloque lineal especificado por una matriz de verificación de paridad H. En general, un código LDPC se define sobre un campo Galois GF (q), q ≥ 2. Si q = 2, el código es un código binario. Todos los códigos de bloque lineal se pueden describir como el producto de un vector de información de k bits s_{1×k} con una matriz de generación de código G_{k×n} para producir una palabra clave de n bits x_{1×n}, donde la tasa de código es r = k/n. La palabra clave x es transmitida a través de un canal ruidoso, y el vector de señal recibida y se pasa al decodificador para estimar el vector de información s_{1×k}.

20 Dado un espacio n-dimensional, las filas de G abarcan el subespacio de palabra clave k-dimensional C, y las filas de la matriz de verificación de paridad H_{m×n} abarcan el espacio dual m-dimensional C[⊥], donde m = n-k. Dado que x = sG y GH^T = 0, se deduce que xH^T = 0 para todas las palabras clave en el subespacio C, donde "T" (o "T") denota transposición de matriz. En la explicación de los códigos LDPC, esto se escribe por lo general como

$$\mathbf{Hx}^T = \mathbf{0}^T, \tag{1}$$

25 donde 0 es un vector de fila de todo ceros, y la palabra clave x = [s p] = [s₀, s₁, ..., s_{k-1}, p₀, p₁, ..., p_{m-1}], donde p₀, ..., p_{m-1} son los bits de verificación de paridad; y s₀, ..., s_{k-1} son los bits sistemáticos, iguales a los bits de información dentro del vector de información.

30 Para un código LDPC, la densidad de entradas no cero en H es baja, es decir, solamente hay un pequeño porcentaje de 1s en H, permitiendo un mejor rendimiento de corrección de errores y una decodificación más simple que usando un H denso. Una matriz de verificación de paridad también se puede describir mediante un gráfico bipartito. El gráfico bipartito no es solamente una descripción gráfica del código, sino también un modelo para el decodificador. En el gráfico bipartito, un bit de palabra clave (por lo tanto cada columna de H) se representa por un nodo variable a la izquierda, y cada ecuación de verificación de paridad (por lo tanto cada fila de H) se representa por un nodo de verificación a la derecha. Cada nodo variable corresponde a una columna de H y cada nodo de verificación corresponde a una fila de H, haciéndose referencia de forma intercambiable a "nodo variable" y "columna" de H, así como "nodo de verificación" y "fila" de H. Los nodos variables solamente están conectados a nodos de verificación, y los nodos de verificación solamente están conectados a nodos variables. Para un código con n bits de palabra clave y m bits de paridad, el nodo variable v_i está conectado al nodo de verificación c_j por un borde si el bit de palabra clave i participa en la ecuación de verificación j, i = 0, 1, ..., n-1, j = 0, 1, ..., m-1. En otros términos, el nodo variable i se conecta al nodo de verificación j si la entrada h_{ji} de la matriz de verificación de paridad H es 1. Reflejando la ecuación (1), los nodos variables representan una palabra clave válida si todos los nodos de verificación tienen paridad par.

45 A continuación se muestra un ejemplo para ilustrar la relación entre la matriz de verificación de paridad, las ecuaciones de verificación de paridad y el gráfico bipartito. Sea n = 12, el código de tasa 1/2 se define por

$$\mathbf{H} = \left[\begin{array}{cccccc|cccccc} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \tag{2}$$

n

50 correspondiendo la porción lateral izquierda a k (= 6) bits de información s, correspondiendo la porción lateral derecha a m (= 6) bits de paridad o. Aplicando (1), H en (2) define 6 ecuaciones de verificación de paridad como sigue:

$$\begin{cases} x_0 + x_2 + x_6 + x_7 = 0 \\ x_1 + x_4 + x_7 + x_8 = 0 \\ x_2 + x_5 + x_6 + x_8 + x_9 = 0 \\ x_0 + x_3 + x_9 + x_{10} = 0 \\ x_1 + x_4 + x_{10} + x_{11} = 0 \\ x_3 + x_5 + x_6 + x_{11} = 0 \end{cases} \quad (3)$$

H también tiene el gráfico bipartito correspondiente representado en la figura 1.

5 Como se ha explicado anteriormente, el receptor obtiene una versión contaminada y de la palabra clave transmitida x. Para descodificar y determinar la secuencia de información original s, se aplica un algoritmo de descodificación iterativo, tal como propagación de creencia, en base al gráfico bipartito. Se pasa información blanda en el formato de relación de probabilidad log (LLRs) de los bits de palabra clave entre el banco de nodos variables y el banco de nodos de verificación. La iteración se para cuando se cumplen todas las ecuaciones de verificación o se alcanza un límite de iteración permitido máximo.

10 Un diseño de código LDPC estructurado empieza con una pequeña matriz base $m_b \times n_b$ H_b , hace z copias de H_b , e interconecta las z copias para formar una matriz $m \times n$ H grande, donde $m = m_b \times z$, $n = n_b \times z$. Usando la representación de matriz para crear H a partir de H_b , cada 1 en H_b es sustituido por una submatriz de permutación $z \times z$, y cada 0 en H_b es sustituido por una submatriz todo ceros $z \times z$. Este procedimiento mapea esencialmente cada borde de H_b a un borde vectorial de longitud z en H, cada nodo variable de H_b a un nodo variable de vector de longitud z en H, y cada nodo de verificación de H_b a un nodo de verificación de vector de longitud z en H. Los beneficios de vectorizar una matriz H_b pequeña para crear una matriz H grande son:

20 1. Usando unos valores diferentes de z, se puede diseñar códigos de tasa k_b/n_b , donde $k_b = n_b - m_b$, para muchos tamaños de secuencia de información diferentes $k = z \times k_b$ a partir de una sola matriz base H_b .

25 2. Los requisitos de memoria se reducen en gran medida. Con un diseño LDPC estructurado, solamente hay que guardar la matriz base H_b y la permutación para sus 1s, lo que requiere significativamente menos memoria dado que H_b es típicamente mucho más pequeño que H y la permutación puede ser muy simple.

30 3. La codificación y la descodificación pueden ser realizadas en grupos de bits más bien que por bits únicos. Por ejemplo, un grupo de z mensajes puede ser tomado de la memoria, permutado y pasado entre un nodo variable de vector y un nodo de verificación de vector.

Aunque la filosofía del diseño LDPC estructurado reduce en gran medida la complejidad de la implementación, no existe una técnica para diseñar la matriz base y asignar las matrices de permutación para un tamaño de H deseado dado que dé lugar a un código LDPC que tenga buen rendimiento de corrección de errores y pueda ser codificado y descodificado eficientemente. Por lo tanto, se necesita un método y aparato para diseñar un H estructurado y un método y aparato para codificar y descodificar datos utilizando la matriz H estructurada.

“LDPC coding for OFDMA PJY” IEEE 802.16 BROADBAND WIRELESS ACCESS WORKING GROUP, 1 Mayo 2004, páginas 0-11, XP002438609, describe codificación LDPC para OFDMA PHY para el estándar 802.16d.

40 Cláusulas que exponen aspectos adicionales y que son útiles para la comprensión de la presente invención

1. Un método para operar un transmisor que genera bits de verificación de paridad $p = (p_0, \dots, p_{m-1})$ en base a un conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$, incluyendo el método los pasos de:

45 recibir el conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$;

usar una matriz H para determinar los bits de verificación de paridad; y

50 transmitir los bits de verificación de paridad junto con el conjunto de símbolos corriente;

donde H es una expansión de una matriz base H_b , incluyendo H_b una sección H_{b1} y una sección H_{b2} , incluyendo H_{b2} una primera parte que tiene una columna h_b que tiene un peso impar superior a 2, e incluyendo una segunda parte H'_{b2} elementos de matriz para la fila i, columna j igual a

55 1 para $i = j$,

$$\mathbf{H}_{b_2} = [\mathbf{h}_b \mid \mathbf{H}'_{b_2}]$$

$$= \left[\begin{array}{c|cccc} h_b(0) & 1 & & & \\ h_b(1) & 1 & 1 & & \mathbf{0} \\ \cdot & & 1 & \ddots & \\ \cdot & & & \ddots & 1 \\ \cdot & & \mathbf{0} & & 1 & 1 \\ h_b(m_b - 1) & & & & & 1 \end{array} \right],$$

donde el vector h_b tiene un peso impar $w_h > = 3$.

5 9. Un aparato incluyendo:

medios de almacenamiento para almacenar una matriz H;

un receptor para recibir un vector de señal $y = (y_0 \dots y_{n-1})$; y

10 un microprocesador que usa una matriz H para determinar un conjunto de símbolos corriente (s_0, \dots, s_{k-1}) , donde H es una expansión de una matriz base H_b y donde H_b incluye una sección H_{b_1} y una sección H_{b_2} , y donde H_{b_2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, e incluyendo una segunda parte H'_{b_2} elementos de matriz para la fila i, columna j igual a

15 1 para $i = j$,

1 para $i = j+1$,

20 0 en otro caso; y

donde se usan dos submatrices idénticas para expandir 1s en cada columna de H'_{b_2} , y se usan submatrices pareadas para expandir un número par de 1s en h_b .

25 10. El aparato de cláusula 9 donde:

$$\mathbf{H}_{b_2} = [\mathbf{h}_b \mid \mathbf{H}'_{b_2}]$$

$$= \left[\begin{array}{c|cccc} h_b(0) & 1 & & & \\ h_b(1) & 1 & 1 & & \mathbf{0} \\ \cdot & & 1 & \ddots & \\ \cdot & & & \ddots & 1 \\ \cdot & & \mathbf{0} & & 1 & 1 \\ h_b(m_b - 1) & & & & & 1 \end{array} \right],$$

donde el vector h_b tiene un peso impar $w_h > = 3$.

30 **Breve descripción de los dibujos**

La figura 1 ilustra el gráfico bipartito de una matriz H (12, 6).

35 La figura 2 ilustra las relaciones entre la matriz base H_b , la matriz modelo H_{bm} , y la matriz expandida final H.

La figura 3 es un diagrama de bloques de un codificador.

La figura 4 es un diagrama de bloques de un descodificador.

La figura 5 es un diagrama de flujo que representa la operación del codificador de la figura 3.

La figura 6 es un diagrama de flujo que representa la operación del descodificador de la figura 4.

5

Descripción detallada de los dibujos

Para satisfacer dicha necesidad, se propone una matriz de verificación de paridad estructurada H , donde H es una expansión de una matriz base H_b y donde H_b incluye una sección H_{b1} y una sección H_{b2} , y donde H_{b2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, e incluyendo una segunda parte elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. La expansión de la matriz base H_b usa submatrices idénticas para 1s en cada columna de la segunda parte H'_{b2} , y la expansión usa submatrices pareadas para un número par de 1s en h_b .

10

La presente invención incluye un método para operar un transmisor que genera bits de verificación de paridad $p = (p_0, \dots, p_{m-1})$ en base a un conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$. El método incluye los pasos de recibir el conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$, y usar una matriz H para determinar los bits de verificación de paridad. Los bits de verificación de paridad son transmitidos junto con el conjunto de símbolos corriente. La matriz H es una expansión de una matriz base H_b donde H_b incluye una sección H_{b1} y una sección H_{b2} , y donde H_{b2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, y una segunda parte H'_{b2} incluyendo elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. La expansión de la matriz base H_b usa submatrices idénticas para 1s en cada columna de la segunda parte H'_{b2} , y donde la expansión usa submatrices pareadas para un número par de 1s en h_b .

15

20

La presente invención incluye adicionalmente un método para operar un receptor que estima un conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$. El método incluye los pasos de recibir un vector de señal recibida $y = (y_0 \dots y_{n-1})$ y usar una matriz H para estimar el conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$. La matriz H es una expansión de una matriz base H_b y donde H_b incluye una sección H_{b1} y una sección H_{b2} , incluyendo H_{b2} una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, e incluyendo una segunda parte H'_{b2} elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. La expansión de la matriz base H_b usa submatrices idénticas para 1s en cada columna de la segunda parte H'_{b2} , y donde la expansión usa submatrices pareadas para un número par de 1s en h_b .

25

30

La presente invención incluye adicionalmente un aparato incluyendo medios de almacenamiento para almacenar una matriz H , un microprocesador que utiliza una matriz H para determinar bits de verificación de paridad, donde H es una expansión de una matriz base H_b y donde H_b incluye una sección H_{b1} y una sección H_{b2} , incluyendo H_{b2} una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, e incluyendo una segunda parte H'_{b2} elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. La expansión de la matriz base H_b usa submatrices idénticas para 1s en cada columna de la segunda parte H'_{b2} , y donde la expansión usa submatrices pareadas para un número par de 1s en h_b .

35

40

La presente invención incluye un aparato incluyendo medios de almacenamiento para almacenar una matriz H , un receptor para recibir un vector de señal $y = (y_0 \dots y_{n-1})$, y un microprocesador que utiliza una matriz H para determinar un conjunto de símbolos corriente (s_0, \dots, s_{k-1}) . La matriz H es una expansión de una matriz base H_b incluyendo H_b una sección H_{b1} y una sección H_{b2} , y donde H_{b2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2. H_{b2} incluye una segunda parte H'_{b2} que tiene elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. Se usan dos submatrices idénticas para expandir 1s en cada columna de H'_{b2} , y se usan submatrices pareadas para expandir un número par de 1s en h_b .

45

Pasando ahora a los dibujos, donde los números análogos designan componentes análogos, la figura 3 es un diagrama de bloques de codificador 300 según una primera realización de la presente invención. Como se representa, el codificador 300 incluye un microprocesador 301 y una tabla de consulta 303. En la primera realización de la presente invención, el microprocesador 301 incluye un procesador de señal digital (DSP), tal como, aunque sin limitación, DSPs MSC8300 y DSP56300. Adicionalmente, la tabla de consulta 303 sirve como medio de almacenamiento para almacenar una matriz, e incluye memoria de lectura solamente; sin embargo, los expertos en la técnica reconocerán que también se puede utilizar otras formas de memoria (por ejemplo, memoria de acceso aleatorio, memoria de almacenamiento magnético, etc). En una segunda realización, la funcionalidad del microprocesador 301 y la tabla de consulta 303 puede estar incorporada a un circuito integrado específico de aplicación (ASIC) o matriz de puertas programable in situ (FPGA). En concreto, la tabla de consulta 303 se puede implementar en una forma de memoria correspondiente a la existencia o inexistencia de recorridos de señal en un circuito.

50

55

60

Como se ha explicado anteriormente, los datos codificados son enviados por lo general como una pluralidad de bits de verificación de paridad además de los bits sistemáticos, donde conjuntamente los bits de verificación de paridad y sistemáticos forman una palabra clave x . En la primera realización de la presente invención, se guarda una matriz de verificación de paridad H en la tabla de consulta 303, y a ella accede el microprocesador 301 para resolver la

65

ecuación (1). En concreto, el microprocesador 301 determina valores apropiados para los bits de verificación de paridad $p = (p_0, \dots, p_{m-1})$ en base al conjunto de símbolos corriente $s = (s_0, \dots, s_{k-1})$ y la matriz de verificación de paridad H . Los bits de verificación de paridad y el conjunto de símbolos se pasan entonces a un transmisor y son transmitidos a un receptor.

5 La figura 4 es un diagrama de bloques del descodificador 400 según una realización de la presente invención. Como se representa, el descodificador 400 incluye un microprocesador 401 y una tabla de consulta 403. En una primera realización de la presente invención, el microprocesador 401 incluye un procesador de señal digital (DSP), tal como, aunque sin limitación, DSPs MSC8300 y DSP56300. Adicionalmente, la tabla de consulta 403 actúa como medio de almacenamiento para almacenar la matriz H , e incluye memoria de lectura solamente. Sin embargo, los expertos en la técnica reconocerán que también se puede utilizar otras formas de memoria (por ejemplo, memoria de acceso aleatorio, memoria de almacenamiento magnético, etc). En una segunda realización, la funcionalidad del microprocesador 401 y la tabla de consulta 403 puede estar incorporada a un circuito integrado específico de aplicación (ASIC) o matriz de puertas programable in situ (FPGA). En particular, la tabla de consulta 403 se puede implementar en forma de memoria correspondiente a la existencia o inexistencia de recorridos de señal en un circuito.

20 El vector de señal recibida (recibida mediante un receptor) $y = (y_0, \dots, y_{n-1})$ corresponde a la palabra clave x transmitida a través de un canal ruidoso, donde los datos codificados x , como se ha explicado anteriormente, son un vector de palabra clave. En la primera realización de la presente invención, se guarda una matriz de verificación de paridad H en la tabla de consulta 403, y a ella accede el microprocesador 401 para descodificar y estimar el conjunto de símbolos corriente s (es decir, el conjunto de símbolos corriente (s_0, \dots, s_{k-1})). En concreto, el microprocesador 401 estima el conjunto de símbolos corriente (s_0, s_{k-1}) en base al vector de señal recibida $y = (y_0, \dots, y_{n-1})$ y la matriz de verificación de paridad H .

25 Como es conocido en la técnica, hay muchas formas en las que el descodificador 400 puede usar la matriz de verificación de paridad H en el microprocesador 401 para descodificación. Una forma es realizar una multiplicación de vector-matriz con H para determinar una configuración de error probable. Otra forma es usar H para construir un gráfico bipartito donde los bordes en el gráfico corresponden a 1s en H , y para procesar iterativamente y en el gráfico bipartito.

30 Para una LDPC estructurada, la submatriz zxz puede ser una matriz de permutación, una suma de matrices de permutación, o cualquier tipo de matriz binaria. Dado que una matriz de permutación P tiene un solo 1 en cada fila y un solo 1 en cada columna, la distribución de peso de la matriz expandida H es la misma que la matriz base H_b si se usa la submatriz de permutación. Por lo tanto, la distribución de peso de H_b se elige tan cerca de la distribución de peso final deseada como sea posible. La descripción siguiente es ilustrativa del caso donde las entradas de H_b son sustituidas por matrices de permutación, aunque se puede usar cualesquiera matrices. Si una submatriz de permutación P_{zxz} de un borde de vector tiene un 1 en la entrada (fila, columna) $(p(i), i)$, entonces el i -ésimo borde dentro del borde de vector es permutado a la posición $p(i)$ -ésima antes de que el borde de vector se conecte al nodo de verificación de vector. En otros términos, esta permutación hace el nodo variable i -ésimo dentro del nodo variable de vector relacionado conectado al nodo de verificación $p(i)$ -ésimo dentro del nodo de verificación de vector relacionado.

45 Las permutaciones incluyendo H pueden ser muy simples sin poner en peligro el rendimiento, tal como desplazamientos cíclicos simples y/o inversiones de bit. Por ejemplo, se puede usar un desplazamiento circular simple a la derecha. Con esta limitación, cada matriz H puede estar representada de forma única por una matriz modelo $m_b \times n_b$ H_{bm} , que se puede obtener

50 * sustituyendo cada 0 en H_b por -1 para denotar una submatriz todo ceros zxz , y

* sustituyendo cada $h_{ij} = 1$ en H_b por un tamaño de desplazamiento circular $p(i,j)$ donde $p(i,j)$ es no negativo.

Dado que los tiempos de desplazamiento circular a la izquierda $(x \bmod z)$ son equivalentes a los tiempos de desplazamiento circular a la derecha $((z-x) \bmod z)$, es adecuado explicar el desplazamiento circular a la derecha y designarlo un desplazamiento circular por razones de brevedad. Como se ha explicado previamente, hay un mapeado de uno a uno entre H y H_{bm} . Por lo tanto, H_{bm} es una representación abreviada de H si se da z . Notacionalmente, la matriz modelo se distingue de la matriz base por el subíndice 'bm', y la matriz expandida se distingue quitando el subíndice 'bm'. La relación entre las tres matrices se ilustra en la figura 2. Usando la estructura, el código tiene un rendimiento de corrección de errores similar a un aleatorio H de tamaño $m \times n$, mientras que la codificación y la descodificación se efectúan en base a un H_{bm} mucho más pequeño.

60 Por ejemplo, la matriz de la ecuación (2) puede ser usada como una matriz base H_b para crear una matriz modelo H_{bm} de la siguiente manera:

$$\mathbf{H}_{bm} = \left[\begin{array}{cccccc|cccccc} 1 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & 2 & 2 & -1 & 0 & 0 & -1 & -1 \\ 2 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & -1 & 0 & -1 & 1 & 0 & -1 & -1 & -1 & -1 & 0 \end{array} \right]_{m_b \times n_b} \quad (4)$$

Cuando $z = 3$, H_{bm} se convierte a una matriz H ($m_b \times z$) \times ($n_b \times z$) sustituyendo cada 1 por una submatriz todo ceros 3×3 y cada i por la submatriz P_i , $i = 0, 1, 2$, donde

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Obsérvese que P_0 es la matriz de identidad, y las columnas de P_i , $i > 0$ son las columnas de los tiempos i de desplazamiento circular a la derecha de P_0 .

Dado un vector $q = [q_0, q_1, q_2]$, $qP_0 = [q_0, q_1, q_2]$, $qP_1 = [q_2, q_0, q_1]$, $qP_2 = [q_1, q_2, q_0]$. En otros términos, qP_i da lugar a un desplazamiento circular a la derecha del vector q . Por otra parte, $P_i q^T$ da lugar a un desplazamiento circular superior de q^T , o un desplazamiento a la izquierda equivalentemente circular de q . Se aplican normas similares cuando se usa una matriz $z \times z$ Q : QP_i da lugar a un desplazamiento circular a la derecha de las columnas de Q , $P_i Q$ da lugar a un desplazamiento circular superior de las filas de Q .

Matriz base H

Para un código LDPC sin vectorización, una matriz H con una estructura de escalera modificada para la parte de paridad de H da lugar a una codificación eficiente sin poner en peligro el rendimiento. En general, suponiendo $x = [s \ p] = [s_0, s_1, \dots, s_{k-1}, p_0, p_1, \dots, p_{m-1}]$, una matriz H m -por- n se puede dividir en dos submatrices,

$$\mathbf{H} = [\mathbf{H}_1 \quad \mathbf{H}_2], \quad (5)$$

donde H_2 tiene una estructura de escalera modificada, y H_1 puede ser cualquier matriz binaria de tamaño m -por- k . Esta misma estructura puede ser usada para crear la matriz base H_b en un diseño LDPC estructurado. Igualmente, usando la estructura de escalera modificada, H_b se puede dividir en dos secciones, donde H_{b1} corresponde a los bits sistemáticos s , H_{b2} corresponde a los bits de verificación de paridad p :

$$\mathbf{H}_b = \left[\begin{array}{c} (\mathbf{H}_{b1})_{m_b \times k_b} \\ \vdots \\ (\mathbf{H}_{b2})_{m_b \times m_b} \end{array} \right]. \quad (6)$$

La sección H_{b2} puede ser dividida además en dos secciones, donde el vector h_b tiene peso impar w_h , y H_{b2} tiene una estructura de escalera:

$$\mathbf{H}_{b2} = [\mathbf{h}_b \mid \mathbf{H}'_{b2}]$$

$$= \begin{bmatrix} h_b(0) & 1 & & & \\ h_b(1) & 1 & 1 & & \mathbf{0} \\ \cdot & & 1 & \ddots & \\ \cdot & & & \ddots & 1 \\ \cdot & \mathbf{0} & & & 1 & 1 \\ h_b(m_b - 1) & & & & & 1 \end{bmatrix}. \quad (7)$$

La sección H_{b1} se puede construir aleatoriamente. Toda la matriz H_b tiene preferiblemente una distribución de peso lo más cerca posible de la distribución de peso deseada.

5 **Tamaños de desplazamiento**

Para convertir la matriz base H_b a la matriz modelo $m_b \times n_b$ H_{bm} (que se expande a H), hay que determinar los tamaños de desplazamiento circular $p(i,j)$ para cada 1 en H_b . En primer lugar, se puede especificar los tamaños de desplazamiento para H_2 . Después de determinar los tamaños de desplazamiento para la sección H_2 , se puede determinar los tamaños de desplazamiento de la sección H_1 para lograr un buen rendimiento general de H . La porción H_1 de la matriz base y los tamaños de desplazamiento de la porción H_1 de la matriz base (sección H_{bm1}) pueden ser asignados de muchas formas diferentes. Por ejemplo, se puede seleccionar valores aleatorios para los tamaños de desplazamiento y aceptarlos si los tamaños de desplazamiento no producen una degradación significativa del rendimiento. La degradación del rendimiento puede deberse a la introducción de números excesivos de ciclos de longitud corta o palabras clave de poco peso. También se puede usar otras técnicas disponibles en la técnica LDPC.

Los tamaños de desplazamiento circular $p(i,j)$ para un tamaño de H deseado dado deberán ser especificados para permitir la codificación eficiente sin poner en peligro el rendimiento de decodificación. Para facilitar la codificación, los desplazamientos pueden ser asignados de tal manera que todas, excepto una, las matrices de desplazamiento correspondientes a 1s en h_b se cancelen cuando se sumen, y todas las filas de vector de H'_{b2} se cancelen cuando se sumen. Esto es lo mismo que asignar tamaños de desplazamiento a h_b en pares excepto para una entrada, y asignar el mismo tamaño de desplazamiento a ambos 1s en cada columna de H'_{b2} . Por ejemplo, si $h_b = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1]^T$, es aceptable tener $h_{bm} = [3 \ -1 \ -1 \ 3 \ -1 \ -1 \ 4]^T$ como la columna correspondiente en la matriz modelo dado que el tamaño de desplazamiento 3 es asignado en pares. Dado que a todas las entradas no cero (ambos 1s) en cada columna H'_{b2} se les asignan los mismos tamaños de desplazamiento, cualquier opción de tamaño de desplazamiento es equivalente al tamaño de desplazamiento de 0 (es decir, submatrices de identidad) más una permutación de los bits dentro de la columna de vector. Así, a todos los tamaños de desplazamiento de H'_{b2} se les puede asignar 0 por razones de conveniencia, es decir, cada 1 en H'_{b2} es sustituido por una submatriz de identidad $z \times z$ al expandir a H .

Debido a la existencia de ciclos, los tamaños de desplazamiento de h_{bm} deberán ser asignados con cuidado. Se puede aplicar normas para evitar la formación de ciclos cortos o palabras clave de peso bajo. Una propiedad que puede ser usada para evitar ciclos es:

35 *Si los bordes $2c$ forman un ciclo de longitud $2c$ en la matriz base H_b , entonces los bordes de vector $2c$ correspondientes forman z ciclos de longitud $2c$ en la matriz expandida H si y solamente si*

$$\sum_{\substack{i=2j \\ j=0, \dots, c-1}} p(i) = \sum_{\substack{i=2j+1 \\ j=0, \dots, c-1}} p(i) \pmod{z}, \quad (8)$$

40 *donde z es el factor de expansión, $p(i)$ es el tamaño de desplazamiento circular del borde i en la matriz modelo H_{bm} , y los bordes $0, 1, 2, \dots, 2c-1$ (en este orden) forman un ciclo en H_b .*

Debido a la estructura de H_{b2} , hay ciclos entre h_b y H'_{b2} . Así, cualesquiera dos tamaños de desplazamiento idénticos en h_{bm} darían lugar a replicar los tiempos de ciclo z en la matriz expandida H según la propiedad anterior. Sin embargo, si estos dos desplazamientos están separados, entonces los ciclos tienen longitud larga, y tienen poco efecto en la decodificación iterativa. Por lo tanto, en una realización preferida, cuando h_b de la matriz base tiene tres 1s, para maximizar la longitud de ciclo, dos 1s a los que se les asignen iguales tamaños de desplazamiento pueden estar situados en la parte superior y la parte inferior de h_{bm} (lo más lejos posible), dejando al mismo tiempo un 1 en el medio de h_b con un tamaño de desplazamiento no pareado. Por ejemplo, $h_{bm} = [3 \ -1 \ 3 \ -1 \ -1 \ -1 \ 4]^T$ daría lugar a z

$$\mathbf{P}_{p(x,k_b)}\mathbf{v}(0) = \sum_{j=0}^{k_b-1} \sum_{i=0}^{m_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j) \quad (14)$$

5 donde x es el índice de fila de H_{bm} donde la entrada es no negativa y se usa un número impar de veces. En la realización preferida, las entradas superior e inferior de H_{bm} están pareadas, así $1 \leq x \leq m_b-2$. La ecuación (14) se resuelve para $\mathbf{v}(0)$ multiplicando ambos lados por \mathbf{P} . Para el caso especial considerado aquí donde $p(x,k_b)$ representa un desplazamiento circular, $\mathbf{P} = \mathbf{P}$. En otros términos, $\mathbf{v}(0)$ se obtiene por

$$\mathbf{v}(0) = \mathbf{P}_{z-p(x,k_b)} \sum_{j=0}^{k_b-1} \sum_{i=0}^{m_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j). \quad (15)$$

10 En general, las recursiones expresadas en las ecuaciones (16) y (17) pueden ser derivadas considerando la estructura de H'_{b2} ,

$$\mathbf{P}_{p(i,k_b+1)}\mathbf{v}(1) = \sum_{j=0}^{k_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 0, \quad (16)$$

Y

$$\mathbf{P}_{p(i,k_b+i+1)}\mathbf{v}(i+1) = \mathbf{P}_{p(i,k_b+i)}\mathbf{v}(i) + \sum_{j=0}^{k_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 1, \dots, m_b - 2, \quad (17)$$

15 donde

$$\mathbf{P}_{-1} \equiv \mathbf{0}_{z \times z}. \quad (18)$$

20 Así, todos los bits de paridad no en $\mathbf{v}(0)$ son determinados evaluando iterativamente las ecuaciones (16) y (17) con respecto a $0 \leq i \leq m_b-2$.

En una realización preferida donde los tamaños de desplazamiento de los 1s en H'_{b2} son todos cero, las ecuaciones (16) y (17) se pueden simplificar a las ecuaciones (19) y (20),

$$\mathbf{v}(1) = \sum_{j=0}^{k_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 0, \quad (19)$$

25 Y

$$\mathbf{v}(i+1) = \mathbf{v}(i) + \sum_{j=0}^{k_b-1} \mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 1, \dots, m_b - 2. \quad (20)$$

30 Así, como en el caso general, todos los bits de paridad no en $\mathbf{v}(0)$ son determinados evaluando iterativamente las ecuaciones (19) y (20) con respecto a $0 \leq i \leq m_b-2$.

35 Las ecuaciones (14), (19) y (20) describen el algoritmo de codificación. Estas ecuaciones también tienen una interpretación sencilla en términos de arquitecturas lógicas digitales estándar. En primer lugar, dado que los elementos no negativos $p(i,j)$ de H_{bm} representan tamaños de desplazamiento circular de un vector, todos los productos de la forma $\mathbf{P}_{p(i,j)}\mathbf{u}(j)$ pueden ser implementados por un desplazador en cilindro de tamaño z . Un tamaño de desplazamiento circular de cero puede no tener que ser desplazado en cilindro. Dado que un desplazador en cilindro que implementa todos los posibles desplazamientos circulares debe proporcionar conexiones de cada bit de entrada a todos los bits de salida, la velocidad con que se puede ejecutar depende de z . Para un z dado, la complejidad se puede reducir e incrementar la velocidad permitiendo solamente un subconjunto apropiado de todos los desplazamientos circulares posibles. Por ejemplo, H_{bm} podría ser construido solamente con tamaños de desplazamiento circular pares. Las sumas en las ecuaciones (14), (19) y (20) representan operaciones XOR (O exclusivo) a modo de vector que son conmutadas por puerta (es decir, no se actualizan) cuando $p(i,j) = -1$.

Para implementar las sumas en las ecuaciones (14), (19) y (20), las entradas $p(i,j)$ de H_{bm} , $0 \leq i \leq k_b$, $0 \leq j \leq m_b-1$, se pueden almacenar en una memoria de lectura solamente (ROM) de anchura $\log_2 z + 1$ bits. La secuencia de información agrupada puede ser almacenada en una memoria de tamaño z , que puede ser leída en orden secuencial. Cuando cada vector de información $u(j)$ es leído, la entrada correspondiente de la ROM H_{bm} puede ser leída, lo que indica al desplazador en cilindro el desplazamiento circular necesario. Después del desplazamiento circular, se actualiza un registro conteniendo una suma parcial. Con respecto a la ecuación (14), después de completar cada suma interior, el resultado puede ser usado para actualizar otro registro conteniendo la suma exterior. Cuando la suma exterior se ha completado, puede ser desplazada circularmente $z-p(x,k_b)$.

5
10
Suponiendo que el desplazamiento en cilindro pueda ser implementado en un solo ciclo de reloj, la codificación puede ser realizada en aproximadamente $(k_b+1)m_b$ ciclos de reloj. Este número se puede reducir a expensas de registros de anchura extra $z m_b-1$ calculando y almacenando las sumas de las ecuaciones (19) y (20), usando resultados que estén disponibles cuando se calcule la ecuación (14).

15 **Extensión de la matriz**

El procedimiento de extensión de código puede ser aplicado al código estructurado para llegar a un código de tasa más baja. El código de tasa progresivamente más baja puede ser usado en transmisiones sucesivas de un procedimiento de redundancia incremental (IR). Específicamente, si la matriz modelo de la primera transmisión es

20

$$\mathbf{H}_{bm}^{(1)} = \begin{bmatrix} \mathbf{H}_{bm1}^{(1)} & \mathbf{H}_{bm2}^{(1)} \end{bmatrix}, \quad (21)$$

entonces la matriz modelo para la segunda transmisión puede usar

25

$$\mathbf{H}_{bm}^{(2)} = \begin{bmatrix} \mathbf{H}_{bm1}^{(1)} & \mathbf{H}_{bm2}^{(1)} & \mathbf{0} \\ \mathbf{H}_{bm1}^{(2)} & \mathbf{H}_{bm2}^{(2)} & \mathbf{H}_{bm2}^{(2)} \end{bmatrix}, \quad (22)$$

etc, donde para cada transmisión i , la submatriz H tiene el formato de (9) y tiene el tamaño $m_b^{(i)} \times m_b^{(i)}$. La primera transmisión puede enviar $n_b^{(1)} = k_b + m_b^{(1)}$ grupos de bits, $[u(0), u(1), \dots, u(k_b-1), v^{(1)}(0), v^{(1)}(1), \dots, v^{(1)}(m_b^{(1)} - 1)]$, teniendo cada grupo el tamaño z . La descodificación después de la primera transmisión se lleva a cabo usando señales recibidas de $[u(0), u(1), \dots, u(k_b-1), v^{(1)}(0), v^{(1)}(1), \dots, v^{(1)}(m_b^{(1)} - 1)]$, y (21). La segunda transmisión puede enviar otros grupos de bits de tamaño z , $[v^{(2)}(0), v^{(2)}(1), \dots, v^{(2)}(m_b^{(2)}-1)]$, donde $m_2 = m_b^{(2)}z$, y los bits de la primera transmisión y la segunda transmisión conjuntamente, $[u(0), u(1) \dots, u(k_b-1), v^{(1)}(0), v^{(1)}(1), \dots, v^{(1)}(m_b^{(1)}-1), v^{(2)}(0), v^{(2)}(1), \dots, v^{(2)}(1) \dots, v^{(2)}(m_b^{(2)}-1)]$, son una palabra clave correspondiente a (22). Por lo tanto, la descodificación después de la segunda transmisión se lleva a cabo en base a (22) y la señal recibida combinada de la primera transmisión y la segunda transmisión. Este procedimiento se puede repetir durante más transmisiones. La descodificación después de la segunda transmisión se basa en un código de tasa $k_b/n_b^{(2)} = k_b / (n_b^{(1)} + m_b^{(2)})$, que es inferior a la de la primera transmisión. Este procedimiento se puede repetir durante más transmisiones, contribuyendo cada transmisión adicional a un código de tasa más baja y más fuerte.

40
45
50
La figura 5 es un diagrama de flujo que representa la operación del codificador 300, y en particular, el microprocesador 301. El flujo lógico comienza en el paso 501 donde un conjunto de símbolos corriente (s_0, \dots, s_{k-1}) es recibido por el microprocesador 301. En el paso 503, los valores de los bits de verificación de paridad se determinan en base al conjunto de símbolos corriente y H . En particular, los bits de verificación de paridad (p_0, \dots, p_{m-1}) se determinan como se ha descrito anteriormente, siendo H una expansión de una matriz base H_b . Como se ha explicado, H_b incluye una sección H_{b1} y una sección H_{b2} , y donde H_{b2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, y una segunda parte incluyendo elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso. Además, la expansión de la matriz base H_b (para producir H) usa submatrices idénticas para los 1s en cada columna de la segunda parte H'_{b2} , y donde la expansión usa submatrices pareadas para un número par de 1s en h_b . En el paso 505 el conjunto de símbolos corriente y los bits de verificación de paridad son transmitidos mediante transmisión por aire.

55
60
La figura 6 es un diagrama de flujo que representa la operación del descodificador 400, y en particular, el microprocesador 401. El flujo lógico comienza en el paso 601 donde se recibe el vector de señal recibida $y = (y_0, \dots, y_{n-1})$. En el paso 603, las estimaciones del conjunto de símbolos corriente s (es decir, el conjunto de símbolos corriente (s_0, \dots, s_{k-1})) son determinadas en base a H . Como se ha explicado, H es una expansión de una matriz base H_b y donde H_b incluye una sección H_{b1} y una sección H_{b2} , y donde H_{b2} incluye una primera parte incluyendo una columna h_b que tiene un peso impar superior a 2, y una segunda parte incluyendo elementos de matriz para la fila i , columna j igual a 1 para $i = j$, 1 para $i = j+1$, y 0 en otro caso.

Aunque la invención se ha mostrado y descrito en particular con referencia a una realización particular, los expertos

5 en la técnica entenderán que se puede hacer en ella varios cambios en la forma y los detalles sin apartarse del espíritu y alcance de la invención. Por ejemplo, aunque la invención se ha mostrado con el orden de s_i y p_i dentro de x definido, los expertos en la técnica reconocerán que otro orden de los bits dentro de x puede tener lugar dado que los bits de palabra clave pueden ser recogidos en cualquier orden a condición de que las columnas de H se reordenen consiguientemente. Adicionalmente, aunque la descripción anterior se ha representado y descrito en particular con referencia a códigos binarios (es decir, códigos definidos sobre el campo Galois $GF(2)$), los expertos en la técnica reconocerán que también se puede utilizar un GF arbitrario. Aunque los ejemplos dados anteriormente se representan en un formato, son posibles otros formatos que permitan un procedimiento similar de codificación y modificación de código. Por ejemplo, las filas de H pueden ser permutadas sin afectar al valor de los bits de verificación de paridad. En otro ejemplo, la estructura de escalera modificada puede ser usada para un subconjunto de los bits de verificación de paridad. En otro ejemplo, se puede llevar a la práctica pasos adicionales al expandir la matriz base a la matriz expandida. La matriz H también se puede usar en cualquier tipo de decodificador que se base en una matriz de verificación de paridad. Se ha previsto que tales cambios caigan dentro del alcance de las reivindicaciones siguientes.

10

REIVINDICACIONES

1. Un método de codificar LDPC una secuencia de información $s = (s_0, s_1, \dots, s_{k-1})$ de longitud $k = k_b \cdot z$ usando una matriz modelo $m_b \times n_b$ H_{bm} , con elementos de matriz $p(i,j)$, $i = 0, 1, \dots, m_b-1$ y $j = 0, 1, \dots, n_b-1$, para obtener una secuencia de paridad $p = (p_0, p_1, \dots, p_{m-1})$ $m = m_b \cdot z$ denotado por v , donde $v = [v(0) v(1) \dots v(m_b-1)]$ y cada elemento de v es un vector de columna como $v(i) = [p_{iz} p_{iz+1} \dots p_{(i+1)z-1}]^T$, $i = 0, 1, \dots, m_b-1$,

5 incluyendo el método

* dividir la secuencia de información s en $k_b = k_b = n_b - m_b$ grupos de z bits denotados por $u = [u(0) u(1) \dots u(k_b - 1)]$, donde cada elemento de u es un vector de columna como sigue

10

$$u(i) = [s_{iz} \quad s_{iz+1} \quad \dots \quad s_{(i+1)z-1}]^T$$

, $i = 0, 1, \dots, k_b-1$;

* determinar $v(0)$ por $v(0) =$

15

$$P_{z-p(x,k_b)} \sum_{j=0}^{k_b-1} \sum_{i=0}^{m_b-1} P_{p(i,j)} u(j)$$

donde x denota el índice de fila de h_{bm} donde la entrada tiene el único valor no negativo que se usa un número impar de veces dentro de h_{bm} y h_{bm} es la k_b -ésima columna de la matriz modelo H_{bm} , donde una matriz de verificación de paridad del código LDPC es una expansión de la matriz modelo H_{bm} reproduciendo cada $p(i, j)$ negativo por una matriz todo ceros $z \times z$ y cada $p(i,j)$ no negativo por una matriz de permutación $P_{p(i,j)}$ que es una matriz de identidad $z \times z$ cuyas columnas son circulares desplazadas por un tamaño de desplazamiento circular de $p(i,j)$ a la derecha;

20

* determinar $v(1), v(2), v(3), \dots, v(m_b-1)$ por

25

$$v(i) = \sum_{j=0}^{k_b-1} P_{p(i,j)} u(j) + P_{p(i,k_b)} v(0), \quad i = 0, \dots, m_b-1$$

$$v(i+1) = v(i) + \sum_{j=0}^{k_b-1} P_{p(i,j)} u(j) + P_{p(i,k_b)} v(0), \quad i = 1, \dots, m_b-2.$$

2. El método de la reivindicación 1, donde la matriz modelo incluye dos submatrices, $[H_{bm1} H_{bm2}]$, donde la primera matriz es de dimensión $m_b \times k_b$, y H_{bm2} es de dimensión $m_b \times m_b$ y además H_{bm2} tiene la estructura siguiente y hay $w_h \geq 3$ entradas no negativas en h_{bm}

30

$$H_{bm2} = [h_{bm} \ ; \ H'_{bm2}]$$

$$= \left[\begin{array}{c|cccc} p(0, k_b) & p(0, k_b+1) & & & \\ p(1, k_b) & p(1, k_b+1) & p(1, k_b+2) & & \\ \vdots & & p(2, k_b+2) & \ddots & \\ & & & \ddots & p(m_b-3, n_b-2) \\ p(m_b-1, k_b) & & & & p(m_b-2, n_b-2) \quad p(m_b-2, n_b-1) \\ & & & & p(m_b-1, n_b-1) \end{array} \right]$$

3. El método de la reivindicación 1, donde el producto de la forma $P_{p(i,j)} u(j)$ se implementa por un desplazador en cilindro de tamaño z .

35

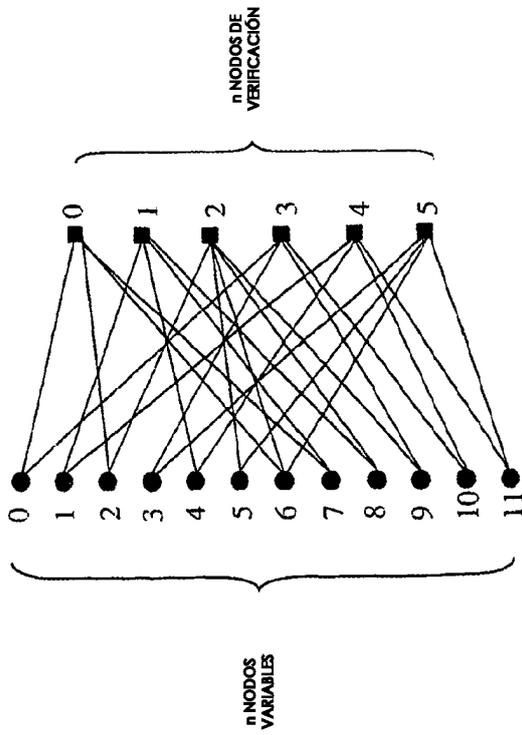


FIG. 1



FIG. 2

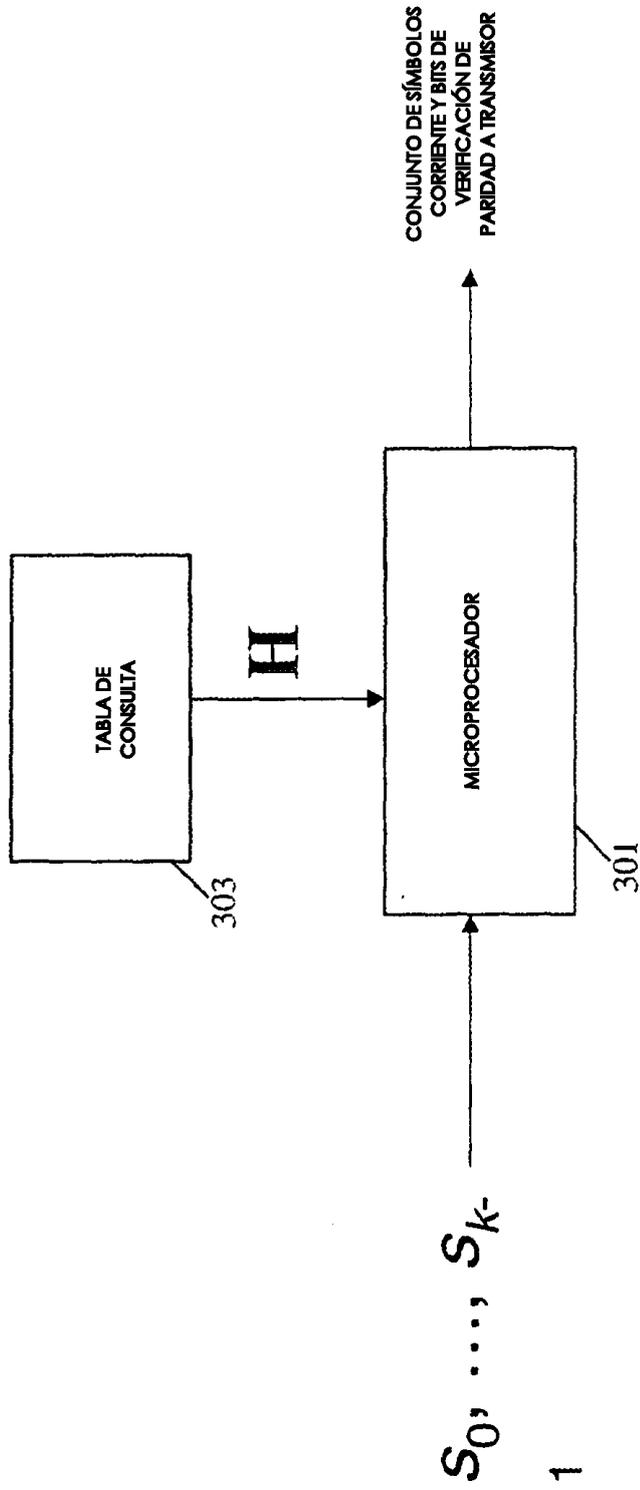


FIG. 3
300

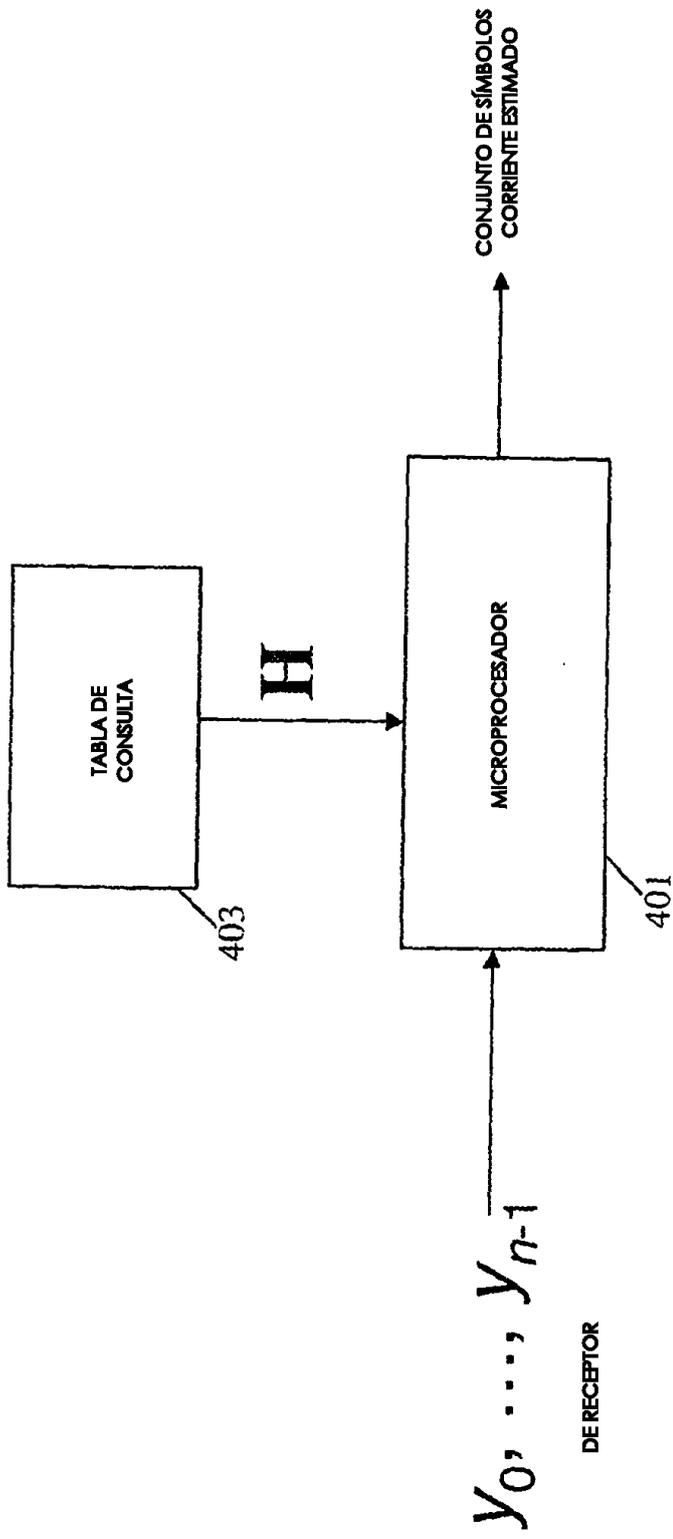


FIG. 4
400

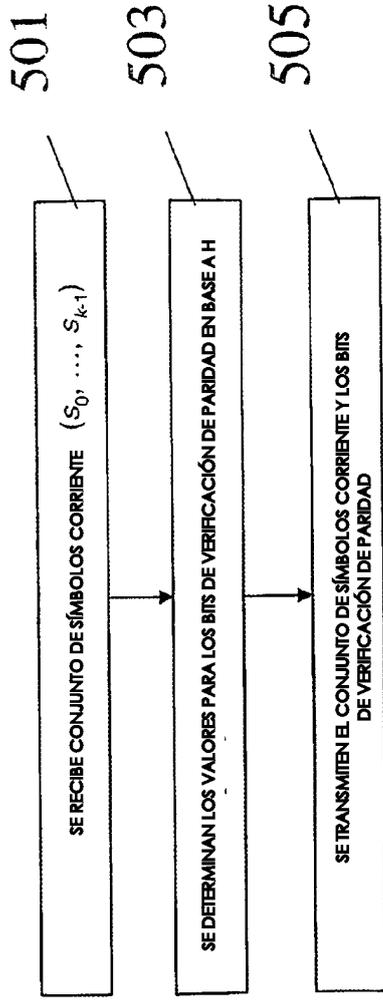


FIG. 5

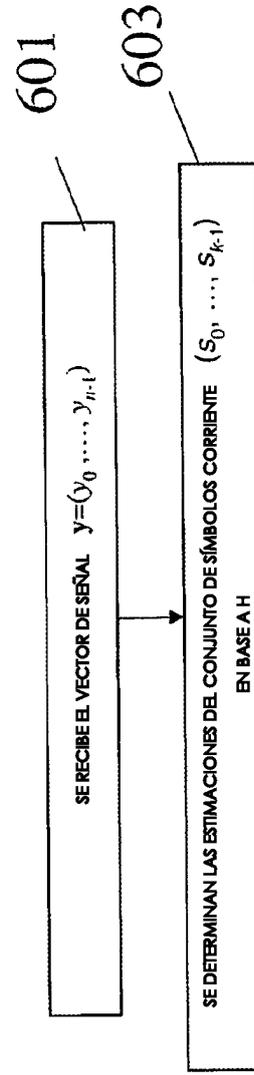


FIG. 6