

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



A standard 1D barcode is located at the bottom of the page, spanning most of the width. It is used for document tracking and identification.

**(43) International Publication Date
26 May 2006 (26.05.2006)**

PCT

(10) International Publication Number
WO 2006/055086 A1

(51) International Patent Classification:
H03M 13/11 (2006.01) *H03M 13/00* (2006.01)

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(21) International Application Number: PCT/US2005/033342

(22) International Filing Date: 19 September 2005 (19.09.2005)

(25) Filing Language: English

(20) Publication Language: English
(30) Priority Data:

(71) **Applicant (for all designated States except US): THOM-SON**

(71) **Applicant (for all designated States except US): THOM-
SON LICENSING [FR/FR]; 46, Quai A. Le Gallo,
F-92100 Boulogne Billancourt (FR).**

(72) Inventors; and
(73) Inventor's Assignee (See U.S. Law) GAO W

(75) **Inventors/Applicants (for US only):** **GAO, Wen** [CN/US]; 73 Warwick Road, West Windsor, NJ 08550 (US). **RAMASWAMY, Kumar** [IN/US]; 71 Sayre Drive, Princeton, NJ 08540 (US). **STEWART, John, Sidney** [US/US]; 9145 Doe Springs Drive, Indianapolis, IN 46278 (US).

(74) **Agents:** **TRIPOLI, Joseph, S.** et al.; c/o Thomson Licensing Inc., 2 Independence Way, Suite 200, Princeton, NJ 08540 (US).

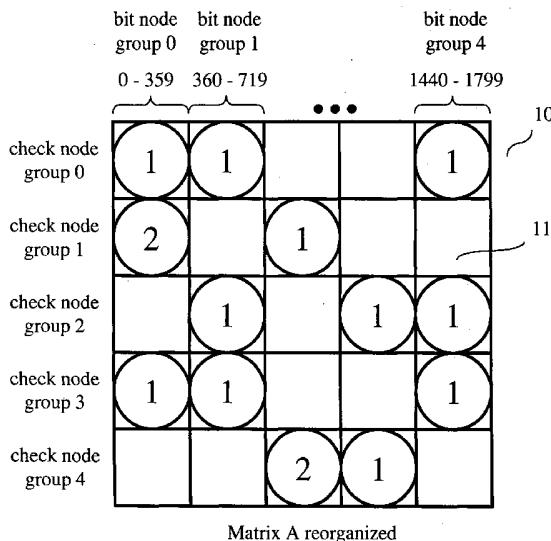
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A LOW DENSITY PARITY CHECK (LDPC) DECODER



WO 2006/055086 A1

(57) Abstract: A satellite receiver comprises a front-end, demodulator and an LDPC decoder. The front-end receives a DVB-S2 LDPC coded signal and provides a down-converted signal to the demodulator. The latter demodulates the down-converted signal and provides a demodulated signal to the LDPC decoder. The LDPC decoder has a partially parallel architecture and partitions the bit node messages into $N/360$ groups and the check node messages into q groups, where $q = M/360$. Each group is processed by 360 bit node processors or 360 check node processors, respectively. Illustratively, the LDPC decoder includes a memory that is partitioned such that messages associated with bit node groups are consecutively addressed. Alternatively, the LDPC decoder includes a memory that is partitioned such that messages associated with check node groups are consecutively addressed.

A LOW DENSITY PARITY CHECK (LDPC) DECODER

BACKGROUND OF THE INVENTION

[0001] The present invention generally relates to communications systems and, more particularly, to a receiver that processes low density parity check (LDPC) encoded data.

5 [0002] In recent years, LDPC codes have increased in popularity because of its near-Shannon limit error-correcting capability. For example, the second generation of digital video broadcast standard (DVB-S2) has adopted LDPC codes as the major error-correcting code replacing the convolutional codes used in the first generation DVB standards (e.g., see European Telecommunications Standards Institute (ETSI) Draft EN 302307, v.1.1.1, June 10 2004).

[0003] In general, an (N, K) LDPC code is a parity check code, where K is the number of bits to be encoded, N is the size (length) of the resulting coded block and $(N-K)$ are the additional error correction bits added by the code. An (N, K) LDPC code can be expressed in matrix form as the set of solutions, x , of the following matrix equation: $Hx^T = 0^T$. This 15 equation is also referred to as the "parity check equation," where the superscript T refers to the transpose of the associated matrix, and H is referred to as the "parity check matrix" having dimensions $M \times N$, where N , as noted above, corresponds to the size of the resulting coded block and $M = N - K$. The modifier "low-density" conveys the fact that the fraction of nonzero elements in the parity check matrix, H , is small, and in particular it is linear in the 20 code block length N . (In contrast, "random" linear block codes are those for which the expected number of ones grows on the order of N^2 .)

[0004] As known in the art, an LDPC code can also be represented by a bipartite graph, which is useful for understanding the LDPC decoding process. In the context of the parity check matrix, H , having dimensions $M \times N$, the corresponding bipartite graph contains N bit 25 nodes (also called variable nodes or message nodes) corresponding to the N columns of the parity check matrix and also contains M check nodes corresponding to the M rows of the parity check matrix. Each check node connects to one, or more, bit nodes. Specifically, an edge (or a branch) connects a check node m to a variable node n if and only if $H_{m,n} = 1$, where $0 \leq n < N$ and $0 \leq m < M$. For a bipartite graph, the term "degree of the bit node" (or 30 bit node degree) refers to the number of check nodes to which the bit node is connected. Similarly, the term "degree of the check node" (or check node degree) refers to the number

of bit nodes to which the check node is connected. It should also be observed that the check node degree and the bit node degree also correspond to the number of "1"s in the respective rows and columns of the parity check matrix, H . Turning briefly to FIG. 1, an illustrative parity check matrix 5 and a corresponding bipartite graph 6 are shown, where $N = 7$ and $M = 3$.

5 Illustratively, the bit node degree for bit node x_7 is one and the check node degree for check node c_3 is four.

[0005] As noted above, the bipartite graph is useful for understanding the LDPC decoding process. In this context, in the LDPC decoder a check node is associated with a check node processor and a bit node is associated with a bit node processor. Unfortunately, 10 while the decoding algorithm for an LDPC decoder is conceptually simple, the architecture of an LDPC decoder for a large code block or a near-random parity check matrix poses significant implementation challenges. There are three known architectures for implementing an LDPC decoder. The first one is a fully parallel architecture, in which all 15 the check nodes, bit nodes and their connections are mapped into hardware. This architecture results in a very high-speed decoder. However, this architecture is not practical for decoding LDPC codes with long block length due to high hardware complexity. The second one is a serial architecture, in which only one check node processing unit (CPU) and one bit node processing unit (BPU) are implemented and reused multiple times to accomplish all the decoding operations. Unfortunately, since all processing is done in a 20 serial fashion, the serial architecture results in a decoder with very low speed. Finally, the third one is a partially parallel architecture, which is a middle ground between the first and second architectures. Here, multiple bit node processing units (BPUs) and multiple check node processing units (CPUs) are implemented and reused to, in effect, trade-off between 25 hardware complexity and decoding latency for the desired LDPC decoder. Unfortunately, no consistent design approach exists for efficiently implementing a partially parallel LDPC decoder.

SUMMARY OF THE INVENTION

[0006] We have observed that it is possible to reduce the complexity of an LDPC decoder by exploiting certain properties of LDPC parity check matrices and, thus, design a 30 more efficient LDPC decoder having a partially parallel architecture. Therefore, and in accordance with the principles of the invention, a receiver performs an LDPC decoding method comprising the steps of: receiving LDPC encoded data; and processing the received

LDPC encoded data to provide decoded data; wherein the processing step partitions the bit node messages into Y groups and the check node messages into q groups, where q varies as a function of the code rate.

[0007] In an embodiment of the invention, a satellite receiver comprises a front-end, a 5 demodulator and an LDPC decoder. The front-end receives a DVB-S2 LDPC coded signal and provides a down-converted signal to the demodulator. The latter demodulates the down-converted signal and provides a demodulated signal to the LDPC decoder. The LDPC decoder has a partially parallel architecture and partitions the bit node messages into $N/360$ groups and the check node messages into q groups, where $q = M/360$. Each group is 10 processed by 360 bit node processors or 360 check node processors, respectively. Illustratively, the LDPC decoder includes a memory that is partitioned such that messages associated with bit node groups are consecutively addressed.

[0008] In an embodiment of the invention, a satellite receiver comprises a front-end, a demodulator and an LDPC decoder. The front-end receives a DVB-S2 LDPC coded signal 15 and provides a down-converted signal to the demodulator. The latter demodulates the down-converted signal and provides a demodulated signal to the LDPC decoder. The LDPC decoder has a partially parallel architecture and partitions the bit node messages into $N/360$ groups and the check node messages into q groups, where $q = M/360$. Each group is processed by 360 bit node processors or 360 check node processors, respectively. 20 Illustratively, the LDPC decoder includes a memory that is partitioned such that messages associated with check node groups are consecutively addressed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 illustrates a parity check matrix and a bipartite graph with respect to LDPC coding;

25 [0010] FIG. 2 shows Table One, which illustrates some DVB-S2 LDPC coding parameters;

[0011] FIGs. 3-5 shows some known observations about the DVB-S2 LDPC parity check matrices;

30 [0012] FIG. 6 shows Table Two which further illustrates some observations about DVB-S2 LDPC coding;

[0013] FIGs. 7-12 illustrate the reorganization of a parity check matrix in accordance with the principles of the invention;

[0014] FIG. 13 shows a portion of an illustrative communications system embodying the principles of the invention;

[0015] FIG. 14 shows an illustrative embodiment of a receiver in accordance with the principles of the invention;

5 [0016] FIG. 15 shows an illustrative embodiment of an LDPC decoder in accordance with the principles of the invention;

[0017] FIGs. 16 and 17 show an illustrative memory structure for use in the LDPC decoder in accordance with the principles of the invention;

10 [0018] FIG. 18 shows an illustrative flow chart in accordance with the principles of the invention for use in the LDPC decoder of FIG. 15;

[0019] FIG. 19 illustrates message passing with respect to the embodiment shown in FIG. 15;

[0020] FIG. 20 shows an illustrative memory structure for use in the LDPC decoder in accordance with the principles of the invention;

15 [0021] FIG. 21 illustrates the operation of a cyclic shifter of FIG. 15;

[0022] FIG. 22 shows an illustrative check node processing unit for use in the LDPC decoder of FIG. 15;

[0023] FIGs. 23 and 24 show an illustrative bit node processing unit for use in the LDPC decoder of FIG. 15;

20 [0024] FIGs. 25-28 show another illustrative embodiment in accordance with the principles of the invention; and

[0025] FIG. 29 shows another illustrative embodiment in accordance with the principles of the invention.

DETAILED DESCRIPTION

25 [0026] Other than the inventive concept, the elements shown in the figures are well known and will not be described in detail. For example, other than the inventive concept, satellite transponders, downlink signals, symbol constellations, carrier recovery, interpolation, phase-locked loops (PLLs), a radio-frequency (rf) front-end, or receiver section, such as a low noise block downconverter, formatting and encoding methods (such as

30 Moving Picture Expert Group (MPEG)-2 Systems Standard (ISO/IEC 13818-1), LDPC coding, etc.) for generating transport bit streams and decoding methods such as log-likelihood ratios, soft-input-soft-output (SISO) decoders, Viterbi decoders are well-known

and not described herein. In addition, the inventive concept may be implemented using conventional programming techniques, which, as such, will not be described herein. Also, familiarity with satellite-based systems (e.g., DBV-S2) and the above-noted ETSI, Draft EN 302307, v.1.1.1, June 2004 is assumed and is not described in detail herein. Finally, like-
5 numbers on the figures represent similar elements.

[0027] Before continuing with a description of the inventive concept, a brief review of the prior art decoding algorithm for an LDPC decoder is provided. It should be noted that the decoding algorithm for an LDPC decoder is sometimes, as known in the art, called the message passing algorithm or the belief propagation algorithm. The message passing
10 algorithm itself seems rather simple. In particular, define a set of check nodes, $M_n = \{m : H_{m,n} = 1\}$ and a set of bit nodes, $N_m = \{n : H_{m,n} = 1\}$. Let $u_{m,n}^{(l)}$ be the message from check node m to bit node n during the l th iteration, let $v_{n,m}^{(l)}$ be the message from the bit node n to check node m during the l th iteration, and let $\lambda_n^{(l)}$ denote an estimate of the a posteriori log-likelihood ratio (LLR) of the n th bit after l iterations. If the channel
15 observation of an LDPC code block is denoted as the vector \mathbf{r} , the message passing algorithm is as follows. At initialization, the following is computed:

$$\lambda_n^{(0)} = \log \frac{\Pr(\mathbf{r} | b_n = 0)}{\Pr(\mathbf{r} | b_n = 1)}, \text{ and} \quad (1)$$

$$\text{let } v_{n,m}^{(0)} = \lambda_n^{(0)}, \text{ for all } n \in \{0, \dots, N-1\} \text{ and } m \in M_n. \quad (2)$$

After initialization, i.e., for iterations $l = 1, 2, \dots, l_{\max}$ the following computations are
20 performed for each check node update and bit node update. For each check node update:

for $m \in \{0, 1, \dots, M-1\}$ and $n \in N_m$, compute

$$u_{m,n}^{(l)} = \left(\prod_{i \in N_m - \{n\}} \text{sign} \left\{ v_{i,m}^{(l-1)} \right\} \right) f \left(\sum_{i \in N_m - \{n\}} f \left(|v_{i,m}^{(l-1)}| \right) \right), \quad (3)$$

$$\text{where } f(x) = \log \frac{e^x + 1}{e^x - 1}.$$

And, for each bit node update:

25 for $n \in \{0, 1, \dots, N-1\}$ and $m \in M_n$, compute

$$\begin{aligned} \lambda_n^{(l)} &= \lambda_n^{(0)} + \sum_{m \in M_n} u_{m,n}^{(l)} \\ v_{n,m}^{(l)} &= \lambda_n^{(l)} - u_{m,n}^{(l)} \end{aligned} \quad (4)$$

The hard-decision and termination criterion for the decoding algorithm are as follows:

$$b_n = 0 \text{ if } \lambda_n^{(l)} > 0, \text{ otherwise } b_n = 1, \quad (5)$$

where a check is made if the bit sequence b_0, b_1, \dots, b_{N-1} satisfies all parity check equations defined by the parity check matrix, H . If so, the iteration terminates, otherwise, let $l \leftarrow (l + 1)$ and continue the iteration until the maximum number of iterations is reached.

5 [0028] As noted above, the message passing algorithm itself is rather simple. However, actual implementations of an LDPC decoder are not always simple due to the hardware constraints, the length of LDPC codes, and the near-random connections between bit nodes and check nodes. This is particularly illustrated by the LDPC codes used in a DVB-S2 satellite system, which will be used to illustrate the inventive concept. However, the
10 inventive concept is not so limited and is applicable to any type of LDPC decoder whether a part of a satellite system or not.

[0029] In DVB-S2 there are four possible modulation schemes: QPSK (quadrature phase shift keying), 8-PSK, 16-APSK (amplitude phase shift keying) and 32-APSK. Before modulation, data is encoded using a serial concatenated code scheme where an LDPC code is
15 the inner code and a BCH (Bose-Chaudhuri-Hochquenghem) code is the outer code. Except for QPSK modulation, the LDPC codeword bits are also interleaved before modulation. With respect to the coding process, the BCH code is a very weak code, which is used to correct the residual errors after the LDPC decoding process in order to achieve 10^{-7} packet error rates. With respect to the LDPC coding, there are two types of LDPC codes. The first
20 type is referred to herein as a "*normal* LDPC code", which has a code block length of 64800 bits. The second type is a short LDPC code, which has a code block length of 16200 bits. Since the two types of codes have similar structures, the *normal* LDPC code will be described herein. For convenience only, and unless stated otherwise, any subsequent references to the term "LDPC code" means a *normal* LDPC code. However, use of the term
25 "LDPC code" in the claims is not so limited.

[0030] For a DVB-S2 system (e.g., see the above-noted ETSI, Draft EN 302307, v.1.1.1, June 2004), there are a number of different LDPC code rates available as shown in Table One of FIG. 2. The first column of the table, labeled "rate", lists these different LDPC code rates. The next column, "K", lists the amount of data encoded in an LDPC coded block at
30 that particular LDPC code rate. In the context of the DVB-S2 system, this data includes the earlier mentioned BCH-coded data. For example, for a 1/4 code rate an un-encoded data block has a size of 16008 bits (not shown in Table One). This un-encoded data block is then

BCH-coded into a BCH-coded block of 16,200 bits (the respective value for K in Table One for an LDPC 1/4 code rate). This BCH-coded block is then LDPC-coded at the particular code rate. Since in this example the LDPC code rate is 1/4, the size of the resulting LDPC-coded block is 68,400 bits (not shown in Table One). It should be noted that the 5 corresponding receiver determines the code rate from data contained in a predefined portion of the received DVB-S2 signal format.

[0031] As can be observed from Table One, there are 11 possible code rates, ranging from 1/4 to 9/10. However, codes with different rates have different parity check matrices as defined in DVB-S2. As such, high rate code words can not be obtained by puncturing of the 10 low rate code words. Thus, large code block length and multiple code rates make the hardware implementation of the LDPC decoder very complicated.

[0032] As noted earlier, there are three major architectures for LDPC decoder implementations. In the context of DVB-S2, the LDPC code block length is 64,800 bits, which is rather large. In addition, the DVB-S2 decoder requires low latency. Hence, a fully 15 parallel or serial architecture is not suitable for decoder implementations and a partially parallel architecture needs to be designed. However, there is no consistent design approach to implementing an efficient partially parallel LDPC decoder.

[0033] To overcome this difficulty, and in accordance with the principles of the invention, it is possible to reduce the complexity of an LDPC decoder by exploiting certain 20 properties of the DVB-S2 parity check matrices. For irregular LDPC codes, one desirable rule is that the distribution of the check node degree (D_C) be as uniform as possible. With regard to the DVB-S2 LDPC codes, it can be determined that for each associated parity 25 check matrix each check node has the same check node degree (D_C) except for the first check node of the parity check matrix, which has a degree of ($D_C - 1$). Thus, the LDPC codes in DVB-S2 follow the above-noted rule.

[0034] In addition, it is known that all the DVB-S2 parity check matrices are of the form $[A \mid T]$, as illustrated in FIG. 3. The matrix A is a rectangular matrix of dimensions $M \times K$, where $M = N - K$. Matrix A is further illustrated in FIG. 4. It should be noted that matrix A itself can be treated as a parity-check matrix, which consists of two submatrices, A_1 and 30 A_2 , where A_1 is a matrix with dimension $M \times L$, and A_2 is a matrix with dimension $M \times (K - L)$. The bit nodes in matrix A_1 have the same degree, denoted as DV_1 and, likewise, the degree of the bit nodes in matrix A_2 are the same and fixed at $DV_2 = 3$.

Turning now to matrix T, this matrix is a special $M \times M$ lower triangular matrix, as show in FIG. 5. This type of structure is sometimes called a staircase structure, which provides bit nodes of degree 2, i.e., $DV_3 = 2$, for a given degree distribution. Furthermore, it should be noted that this lower triangle structure enables fast LDPC encoding (e.g., see the ETSI, Draft 5 EN 302307, v.1.1.1, June 2004).

[0035] Turning now to FIG. 6, Table Two illustrates the values for the above-mentioned L , DV_1 , q and D_C for the different DVB-S2 code rates. The first two columns of Table Two, labeled "rate" and "K" are identical to the columns shown in Table One of FIG. 1.

[0036] In accordance with the principles of the invention, further analysis of the structure of matrix A sheds additional light on implementing an LDPC decoder. In particular, for every group of 360 bit nodes $\{360 \times k, \dots, 360 \times k + 359\}$, the check nodes they involve can be specified by the check nodes of the first bit node with an index $(360 \times k)$. For example, if the first bit node in the group involves a set of check nodes $\{C_1, C_2, \dots, C_{DV}\}$, where DV is the degree of the bit nodes, the bit node with index $(360 \times k + m)$ involves a set of check nodes, given as:

$$\{c \mid c = (x + m \times q) \bmod M, x \in \{C_1, C_2, \dots, C_{DV}\}\}, \quad (6)$$

where $q = \frac{M}{360}$.

[0037] In light of the above-described observations, and in accordance with the principles of the invention, the bit nodes and check nodes are each organized into multiple groups in order to carry out the bit node update or check node update operations simultaneously. With respect to this particular example, and as illustrated by equation (6), every 360 bit nodes $\{360 \times k, \dots, 360 \times k + 359\}$ can be processed as one group, i.e., the bit nodes are grouped consecutively, such as,

for $n \in \{0, 1, \dots, (K / 360) - 1\}$, the n -th bit node group will contain the bit node

$$\{360n, 360n+1, \dots, 360n+358, 360n+359\}.$$

These bit nodes are also referred to herein as systematic-bit nodes.

[0038] With respect to the check nodes, the check nodes are re-arranged into q groups as follows (where, as noted above, $q = \frac{M}{360}$, i.e., q varies as a function of the code rate):

Group 0: $\{0, q, 2 \times q, 3 \times q, \dots, 359 \times q\}$;

Group 1: $\{1, 1 + q, 1 + 2 \times q, 1 + 3 \times q, \dots, 1 + 359 \times q\}$;

Group q-2: $\{q - 2, q - 2 + q, \dots, q - 2 + 359 \times q\}$; and

Group q-1: $\{q - 1, q - 1 + q, \dots, q - 1 + 359 \times q\}$.

[0039] Since an LDPC-coded block has a size of $N = 64,800$ bits, a smaller size A 5 matrix will be described below to further illustrate the inventive concept. FIG. 7 shows a matrix 10 (a matrix of form A) re-organized in accordance with the principles of the invention. The matrix 10 is for an LDPC code having the following parameters:

$$N = 10 \times 360 = 3600;$$

$$M = 5 \times 360 = 1800;$$

10 $q = 5$;

$$DV_1 = 4; \text{ and}$$

$$L = 360.$$

[0040] Each square, 11, represents a submatrix of dimensions 360×360 . Other than the 15 inventive concept, it should be noted that the notation shown in FIG. 7 is known in the art with respect to similar code constructions (e.g., see David J. C. Mackay, Simon T. Wilson and Matthew C. Davey, "Comparison of Constructions of Irregular Gallager Codes", IEEE Transactions on Communications, Vol. 47, pp. 1449-1454, Oct. 1999; and D. Sridhara, T. Fuja and R. M. Tanner, "Low density parity check codes from permutation matrices," Conf. On Info. Sciences and Sys., The John Hopkins University, March 2001). In particular, a 20 blank square represents an all-zero matrix and an integer in a circle within a square represents a number of cyclic identity matrices superposed on the surrounding square. The number one represents a single cyclic identity matrix having a particular offset while the number two represents a combination of two cyclic identity matrices. This is further illustrated in FIGs. 8 and 9. Turning first to FIG. 8, this figure illustrates different offsets of 25 in the context of a left-shifted cyclic identity matrix. Matrix 21 illustrates the identity matrix. This is also referred to herein as a cyclic identity matrix with no shift, i.e., having an offset of zero. Moving from left to right in FIG. 8, matrix 21 is left-shifted once resulting in matrix 22. If one compares the position of element 24 in matrix 22 with its previous position in matrix 21, it can be observed that element 24 appears in the same row, but has been 30 shifted one column to the left (with the columns, in effect, wrapping around). As such, matrix 22 is a cyclic identity matrix having an offset of one. Matrix 22 is again left-shifted once resulting now in matrix 23. Again, it can be observed from FIG. 8 that element 24 has

shifted one column to the left from its previous position in matrix 22. Since matrix 23 is the result of two left shifts, matrix 23 is a cyclic identity matrix having an offset of two. Other offsets can be derived in a similar fashion and, although not shown in FIG. 8, right-shifting operations could also be equivalently performed in the other direction. For convenience, a 5 left-shifted cyclic identity matrix is denoted herein as the matrix $I^{(y)}$, where the value of the superscript represents the value of the offset.

10 [0041] Referring now to FIG. 9, the concept of a combined cyclic identity matrix is illustrated. A combined cyclic identity matrix is a combination of two or more cyclic identity matrices. With respect to FIG. 9, this figure illustrates combinations of two cyclic identity matrices. In particular, matrix 26 is a combination of matrices 21 and 22 of FIG. 8, matrix 27 is a combination of matrices 22 and 23 of FIG. 8, and matrix 28 is a combination 15 of matrices 21 and 23 of FIG. 8. Other combinations can be derived in a similar fashion.

[0042] In light of the above, FIG. 10 again illustrates matrix 10 of FIG. 7 (a matrix of form A) with the patterns of the particular left-shifted cyclic identity matrices and combined cyclic identity matrices shown. If there are lines within a submatrix, this represents that the corresponding submatrix elements on which the line crosses have a value of "1" and the other submatrix elements have values of "0". For a submatrix having no lines inside it, this represent an all zero submatrix.

20 [0043] As a result, it can be observed from FIGs. 7 and 10 that for all LDPC codes, the A matrix of the parity check matrix comprises three types of sub-matrices of dimension 360 x 360:

- a zero matrix;
- a cyclic identify matrix, $I^{(y)}$, for $0 \leq y \leq 359$; and
- a combined cyclic identity matrix, $I^{(x)} + I^{(y)}$, for $x \neq y$, and $0 \leq x, y \leq 359$.

25 [0044] Now, there is another way to describe an LDPC parity check matrix. In particular, let $H(m, n)$ denote the submatrix of the A matrix of the parity check matrix corresponding to check node group m , and bit node group n , and only show the nonzero submatrices. For the n -th row in the address of the Parity Bit Accumulators Table (other than the inventive concept, addresses of parity bit accumulators are described in ETSI, Draft 30 EN 302307, v.1.1.1, June 2004), a set of submatrices is obtained corresponding to the n -th bit node group. For a given number x in the n -th row of the table, the corresponding check node group is $m = x \bmod q$, the value for the left cyclic shift number is $y = \lfloor x / q \rfloor$, and the

corresponding submatrix is $\mathbf{H}(m, n) = \mathbf{I}^{(y)}$. For example, from FIG. 6, for a rate 1/2 code, the value of $q = 90$, and from Annex B of the ETSI, Draft EN 302307, v.1.1.1, June 2004, the zero-th row ($n = 0$) of the address of the Parity Bit Accumulators Table is:

54, 9318, 14392, 27561, 26909, 10219, 2534, 8597.

5 (In terms of the nomenclature in the ETSI draft, the zero-th row of the parity bit accumulator table for a rate 1/2 code corresponds to the rows of the parity matrix for the zero-th bit node for which there is a "1" in the column.)

[0045] Hence the corresponding submatrices for the A matrix are:

10 54: $\mathbf{H}(54,0) = \mathbf{I}^{(0)}$;

9318: $\mathbf{H}(48,0) = \mathbf{I}^{(103)}$;

14392: $\mathbf{H}(82,0) = \mathbf{I}^{(159)}$;

27561: $\mathbf{H}(21,0) = \mathbf{I}^{(306)}$;

26909: $\mathbf{H}(89,0) = \mathbf{I}^{(298)}$;

10219: $\mathbf{H}(49,0) = \mathbf{I}^{(113)}$;

15 2534: $\mathbf{H}(14,0) = \mathbf{I}^{(28)}$; and

9597: $\mathbf{H}(47,0) = \mathbf{I}^{(95)}$.

[0046] Likewise, consider the first row ($n = 1$) from the same Parity Bit Accumulators Table (again, from Annex B of the ETSI, Draft EN 302307, v.1.1.1, June 2004):

55, 7263, 4635, 2530, 28130, 3033, 23830, 3651.

20 [0047] Hence the corresponding submatrices for the A matrix are:

55: $\mathbf{H}(55,1) = \mathbf{I}^{(0)}$;

3033, 7263: $\mathbf{H}(63,1) = \mathbf{I}^{(33)} + \mathbf{I}^{(80)}$;

4635: $\mathbf{H}(45,1) = \mathbf{I}^{(51)}$;

2530: $\mathbf{H}(10,1) = \mathbf{I}^{(28)}$;

28130: $\mathbf{H}(50,1) = \mathbf{I}^{(312)}$;

23830: $\mathbf{H}(70,1) = \mathbf{I}^{(264)}$; and

3651: $\mathbf{H}(51,1) = \mathbf{I}^{(40)}$.

[0048] It should be noted that the calculations for 3033 and 7263 with respect to the first row both result in the same submatrix $\mathbf{H}(63,1)$ but with different cyclic identity matrices, $\mathbf{I}^{(33)}$ and $\mathbf{I}^{(80)}$, respectively. Hence, the submatrix $\mathbf{H}(63,1)$ is the summation of these two cyclic identity matrices as shown above.

[0049] As noted earlier, all DVB-S2 parity check matrices are of the form $[A \mid T]$. In accordance with the principles of the invention, in matrix T the bit nodes are grouped in a different way, compared with the bit nodes in matrix A. For $n \in \{(K/360), \dots, (N/360)-1\}$, the n-th bit node group contains the bit node:

5
$$K + (n - K/360) + \{0, q, 2 \times q, 3 \times q, \dots, 359 \times q\}.$$

The discontinuity of the parity-bit nodes in one bit node group is due to the re-order of the parity-check equations. An example of the resulting T matrix is shown in FIG. 11. It can be observed from FIG. 11 that there are three possible squares of dimension 360×360 in matrix T:

10 - a zero matrix;
- a identity matrix, $I^{(0)}$; and
- a square $H(0, (N/360)-1)$ that contains a special submatrix of dimension 360×360 , shown in FIG. 12.

[0050] The above-described rearrangement of the parity check matrix is now used to 15 implement an LDPC decoder in accordance with the principles of the invention. An illustrative portion of a communications system in accordance with the principles of the invention is shown in FIG. 13. As can be observed from FIG. 13, a signal 104 is received by a receiver 105. Signal 104 conveys information representative of control signaling, content (e.g., video), etc. In the context of this example, it is assumed that signal 104 represents a 20 DVB-S2 downlink satellite signal after reception by an antenna (not shown). Receiver 105 processes signal 104 in accordance with the principles of the invention (described below) and provides a signal 106 for conveying particular content to a multi-media endpoint as represented by television (TV) 90 for display thereon.

[0051] Turning now to FIG. 14, an illustrative portion of receiver 105 in accordance with 25 the principles of the invention is shown. Receiver 105 includes front end filter 110, analog-to-digital (A/D) converter 115, demodulator 120, LDPC decoder 125 and BCH decoder 135. Front end filter 110 down-converts (e.g., from the satellite transmission bands) and filters received signal 104 to provide a near baseband signal to A/D converter 115, which samples the down converted signal to convert the signal to the digital domain and provide signal 116, 30 which is a sequence of samples, to demodulator 120. The latter performs demodulation of signal 116 (including carrier recovery) and provides a demodulated signal 121 to LDPC decoder 125, which, in accordance with the principles of the invention, decodes the

demodulated signal point stream 121 to provide signal 126, which represents a BCH-coded signal, or data stream. Signal 126 is applied to BCH decoder 135 for recovery of the transmitted data as represented by signal 136. At least some of the data from signal 136 is eventually provided (not shown in FIG. 14) to TV 90 via signal 106. (In this regard, receiver 5 105 may additionally process the data before application to TV 90 and/or directly provide the data to TV 90.)

[0052] In accordance with the principles of the invention, an illustrative embodiment of LDPC decoder 125 is shown in FIG. 15. LDPC decoder 125 comprises log-likelihood ratio (LLR) computing element 205, LLR buffer 210, multiplexer (mux) 215, edge memory 220, 10 cyclic shifters 225 and 235, a plurality of check node processing units (group CPU processing) 230, a plurality of bit node processing units (group BPU processing) 240, iteration termination decision element 245 and controller 290. The latter is representative of a stored-program controlled processor (e.g., a microprocessor and associated memory) or a state machine, etc.

15 [0053] LLR computing element 205 receives the demodulated signal point stream signal 121 and computes the LLR as known in the art to provide signal 206, which represents the calculated LLR values that are representative of the received LDPC coded blocks. In particular, LLR computing element 205 computes the LLR of codeword bits as

$$\lambda_n^{(0)} = \log \frac{\Pr(\mathbf{r} | b_n = 0)}{\Pr(\mathbf{r} | b_n = 1)} \text{ based on the modulation scheme and the signal to noise ratio of the}$$

20 received signal. For simplicity, lookup tables (not shown) are used to implement this function. In addition, LLR computing element 205 also de-interleaves the LLR values before they are sent, via signal 206, to LLR buffer 210 (as noted earlier, the LDPC coded bit stream was interleaved before modulation unless QPSK modulation was used). LLR buffer 210 is a storage element and comprises, e.g., a double buffer structure to alternately store the 25 data representative of the received LDPC coded blocks. As such, when one buffer is being filled, data from the other buffer is processed, via signal 211, for decoding of the previously received LDPC coded block. An illustrative memory structure 315 for use in an LLR buffer for the systematic bit nodes of matrix A is shown in FIG. 16; while an illustrative memory structure 320 for use in the LLR buffer for the bit nodes of matrix T is shown in FIG. 17. 30 From FIGs. 16 and 17, the number of memory words required is $N / 360 = 64800 / 360 = 180$, where the bit width of one memory word is $360 \times 6 = 2160$ bits

if it is assumed that 6 bits are required to store the initial channel information ($\lambda_n^{(0)}$). As such, LLR buffer 210 provides LDPC coded blocks via signal 211 to mux 215. The latter is controlled by a processor as represented by controller 290 which controls the various elements of LDPC decoder 125 as represented for simplicity by the dashed arrows. Mux 215 provides, via signal 216, either of three types of data to edge memory 220: a received LDPC coded block for decoding (via signal 211); bit node processing data (via signal 241), or check node processing data (via signal 236).

[0054] Reference now should be made to FIG. 18, which shows an illustrative flow chart of an overall process that is used in LDPC decoder 125 for performing LDPC decoding. In step 405, an LDPC coded block is provided from LLR buffer 210 to edge memory 220 for storage therein. In steps 410 and 415, LDPC decoding is performed. In particular, check node updates (step 410) and bit node updates (step 415) operate on the data stored in edge memory 220 (described below). In step 420, a check is made if the decoding process should be terminated, e.g., from equation (5), above. If the process is terminated, execution returns to step 405 to begin decoding the next LDPC coded block, otherwise, decoding continues with another round of check node and bit node updates via steps 410 and 415. It should be noted that for simplicity error conditions are not shown in the flow chart of FIG. 18.

[0055] As noted above, edge memory 220 stores the LDPC coded data and is accessed in both the check node update and bit node update steps shown in FIG. 18. Edge memory 220 is representative of a storage element. While edge memory 220 can be implemented using registers, which allow for fast access (albeit with higher design complexity), preferably a bank of memory is a more suitable implementation given the length of the LDPC coded blocks. In the LDPC decoding process, messages are passing through the edges of the bipartite graph between bit nodes and check nodes. This is conceptually illustrated in FIG. 19, which shows a portion of an illustrative bipartite graph. For example, a bit node n is coupled to a check node m by an edge 40, which enables the passing of messages therebetween as represented by bit node message 41 and check node message 42. Since the memory used in the LDPC decoding process is associated with the edges between the check nodes and the bit nodes, this memory is referred to herein as edge memory. As such, edge memory 220 stores the messages from check nodes to bit nodes $\{u_{m,n}^{(l)}\}$, via signal 236, or the messages from bit nodes to check nodes $\{v_{n,m}^{(l)}\}$, via signal 241. In particular, and as can be observed from the flow chart of FIG. 18, LDPC decoder 125 has at least two phases: a check

node update phase (e.g., step 410 of FIG. 18) and a bit node update phase (e.g., step 415 of FIG. 18). At the beginning of the check-node update phase, $\{v_{n,m}^{(l-1)}\}$ is stored in a memory location of edge memory 220; while at the end of the check-node update phase, $\{u_{m,n}^{(l)}\}$ is computed and stored in the same memory location. Likewise, in the bit node update phase, 5 $\{u_{m,n}^{(l)}\}$ is read out and $\{v_{n,m}^{(l)}\}$ is computed and stored into the same memory location. Thus, in a partially parallel architecture, the same memory location is used to store $\{v_{n,m}^{(l)}\}$ or $\{u_{m,n}^{(l)}\}$ depending on the phase of the LDPC decoder.

[0056] In accordance with the principles of the invention, edge memory 220 can be organized in terms of the bits nodes or in terms of the check nodes in accordance with the 10 above-described reorganization of the parity matrix. It should be noted that the overall amount of memory required is the same for both cases since the number of edges is fixed for a particular parity check matrix.

[0057] In one embodiment, edge memory 220 is illustratively organized in terms of bit nodes. In this case, one memory word is used to store all the messages corresponding to a 15 bit node group are stored in consecutive address locations, which makes the bit node update simple. An illustrative memory structure 325 for use in edge memory 220 is shown in FIG. 20. Since the memory of edge memory 220 is organized in terms of bit nodes, this memory may also be referred to as a bit node memory bank.

[0058] Returning back to FIG. 15, data stored in edge memory 220 is provided to either 20 a bit node processing path or a check node processing path via signal 221. With respect to the check node processing path, this path is active in the check node update phase (step 410 of FIG. 18). In particular, data (whether the initial LDPC coded data or the subsequent message data, $\{v_{n,m}^{(l-1)}\}$) is provided to group CPU processing 230 via cyclic shifter 225. Since 25 edge memory 220 is organized in terms of bit nodes, cyclic shifter 225 cyclically shifts the data in the memory words such that the data for one check node group are aligned. This is illustrated in FIG. 21, which shows the amount of cyclic shift for check node group 0/bit node group 0. Group CPU processing 230 comprises 360 check node processing units (described further below) for computing $\{u_{m,n}^{(l)}\}$ and for providing $\{u_{m,n}^{(l)}\}$ to cyclic shifter 235, which again reorients the data in the memory words such that the data for one bit node 30

group are aligned. Cyclic shifter 235 provides $\{u_{m,n}^{(l)}\}$, via signal 236, to edge memory 220 via mux 215 and signal 216. It should be noted that one cyclic shifter may be used instead of two by multiplexing its operation in the time domain. Turning now to the bit node processing path, this path is active in the bit node update phase (step 415 of FIG. 18). In 5 particular, data, $\{u_{m,n}^{(l)}\}$ is provided to group BPU processing 240. Group BPU processing 240 illustratively comprises 360 bit node processing units (described further below) for computing $\{v_{n,m}^{(l)}\}$ and for providing $\{v_{n,m}^{(l)}\}$, via signal 241, to edge memory 220 via mux 215 and signal 216.

[0059] As described above, group CPU processing 230 comprises 360 check node processing units. An illustrative check node processing unit (CPU) 230-J, where $0 < J \leq 360$, is shown in FIG. 22. CPU 230-J processes a set of input messages $\{e_0, e_1, \dots, e_{D_c-1}\}$ to provide a corresponding set of output messages $\{e'_0, e'_1, \dots, e'_{D_c-1}\}$. As noted earlier, equation (3) is used in LDPC decoding for generating the set of output messages. However, the complexity of each check node processing unit increases if the exact formula in equation (3) 15 is implemented. Indeed, since the maximum possible check node degree is 30, the implementation of the adder array and all the functions $f(\bullet)$ become very complicated even though the function $f(\bullet)$ can be implemented by a simple look-up table. In order to reduce the complexity of a check node processing unit, the CPU 230-J implements the following approach. In particular, assume that $\{e_0, e_1, \dots, e_{D_c-1}\}$ are the set of input messages to a 20 check node processing unit (CPU). Then, compute

$$s_k = \text{sign}(e_k); \quad (7)$$

$$S = \prod_{k=0}^{D_c-1} s_k, \quad (8)$$

Now, select the 3 smallest values in the set $|e_0|, |e_1|, \dots, |e_{D_c-1}|\}$ and let their corresponding indices be m_0, m_1, m_2 . After this, the following four values are computed:

$$25 \quad \lambda_0 = g(|e_{m_0}|, |e_{m_2}|), \quad (9)$$

$$\lambda_1 = g(|e_{m_0}|, |e_{m_1}|), \quad (10)$$

$$\lambda_2 = g(|e_{m_0}|, |e_{m_1}|), \quad (11)$$

$$\lambda_3 = g(g(|e_{m_0}|, |e_{m_1}|), |e_{m_2}|), \quad (12)$$

where

$$g(x, y) \equiv \text{sign}(x)\text{sign}(y)\{\min(|x|, |y|) - h(|x| - |y|)\}, \text{ and} \quad (13)$$

$$h(x) = \ln(1 + e^{-x}). \quad (14)$$

Then, a set of output messages $\{e'_0, e'_1, \dots, e'_{D_C-1}\}$ are computed as follows:

$$5 \quad e'_k = s_k \times S \times \begin{cases} \lambda_0, & \text{for } k = m_0 \\ \lambda_1, & \text{for } k = m_1 \\ \lambda_2, & \text{for } k = m_2 \\ \lambda_3, & \text{for } k \neq m_0, m_1, m_2 \end{cases}, \quad (15)$$

where $0 < k \leq D_C-1$. In the above-described approach, only the three smallest values of the input messages are used to compute the output messages of the CPU. Simulations have shown that the performance loss due to this approximation is negligible for all the LDPC codes in DVB-S2.

10 [0060] Turning now to the bit node processing, group BPU processing 240 comprises 360 bit node processing units. An illustrative bit node processing unit (BPU) 240-I, where $0 < I \leq 360$, is shown in FIG. 23. BPU 240-I processes a set of input messages $\{e_0, e_1, \dots, e_{D_V-1}\}$ to provide a corresponding set of output messages $\{e'_0, e'_1, \dots, e'_{D_V-1}\}$. The bit node processing operation is rather simple and is further illustrated in FIG. 24. In FIG. 24, 15 the term *LLR* denotes the log-likelihood ratio of the associated bit node which is provided via signal 211 from LLR buffer 210.

20 [0061] The final element of LDPC decoder 125 is iteration termination decision element 245 which implements the above-described step 420 of FIG. 18. As can be observed from FIG. 15 and 23, signal 242 is provided from the bit node processing path to iteration termination decision element 245 for use therein. If the LDPC decoding is terminated, the resulting LDPC decoded data is provided via signal 126 to BCH decoder 135, described above. Iteration termination decision element 245 provides signaling (not shown) to controller 290 with respect to continuing the LDPC decoding process or starting anew, e.g., for the next LDPC coded block.

25 [0062] As noted earlier, DVB-S2 supports a number of code rates with predefined parity matrices and the receiver determines the code rate from data contained in a predefined portion of the received DVB-S2 signal format. In this context, controller 290 uses the determined modulation type to select different look-up tables (not shown) for the earlier-described LLR computations and different interleaving schemes (as defined in DVB-S2).

Controller 290 also configures LDPC decoder 125 in accordance with the principles of the invention to process received LDPC coded signals at the different code rates in accordance with parity matrices reorganized in accordance with the principles described earlier. For example, the above noted sub-matrix calculations ($H(m, n)$) can be stored *a priori* in a 5 memory (e.g., configuration memory 295 of FIG. 15) for subsequent use in processing the received LDPC coded data for the different parameters as illustrated in Table Two of FIG. 6.

[0063] Another illustrative embodiment of LDPC decoder 125 is shown in FIG. 25. This arrangement is similar to that shown in FIG. 15 and functions in a similar fashion (e.g., see FIGs. 18, 22, 23 and 24) except that edge memory 220 is organized in terms of check 10 nodes (and can also be referred to as a check node memory bank). As such, the two cyclic shifters 225 and 235 are now positioned before and after group BPU processing 240. Again, it should be noted that once cyclic shifter may be used instead of two by simply multiplexing its operation in the time domain.

[0064] In terms of organizing edge memory 220 in terms of check nodes, the check node 15 memory corresponding to one check node group is put together and uses one memory location, e.g., a word, to store all the messages corresponding to a particular cyclic identity matrix. In other words, one memory word stores all the messages sent through the edges associated with a particular cyclic identity matrix. As observed earlier, and shown in Table Two of FIG. 6, for LDPC codes the degree of all check nodes, D_c , is the same except for the 20 0-th check node, which has the check node degree ($D_c - 1$). As an illustration, consider the following example using a 1/2 rate LDPC code where, from Table Two of FIG. 6, $D_c = 7$, and the 63-rd check node group corresponds to the following square matrix, which are submatrices in the parity-check matrix (calculated as illustrated earlier):

$$\begin{aligned}
 H(63,1) &= I^{(33)} + I^{(80)}; \\
 H(63,9) &= I^{(0)}; \\
 H(63,29) &= I^{(324)}; \\
 H(63,34) &= I^{(132)}; \\
 H(63,62) &= I^{(0)}; \text{ and} \\
 H(63,63) &= I^{(0)}.
 \end{aligned}$$

[0065] An illustrative memory bank 305 within edge memory 220 corresponding to the 30 63-rd check node group is shown in FIG. 26. If each row of memory bank 305 is treated as a single memory word, then only $D_c = 7$ memory words are required as shown for a check

node group. Indeed, all the memory banks for all the check nodes can then be put together and addressed in a linear fashion. This illustrative memory structure 310 for edge memory 220 is shown in FIG. 27. The address space of edge memory 220 for memory structure 310 is $\{0,1,2,\dots, qD_C - 1\}$. In other words, the size of the address space is $(q \times D_C)$. The required memory space for the different LDPC code rates is shown in Table Three of FIG. 28. As can be observed from Table Three, the maximum number of memory words required is 792. If it is again assumed that each bit node message or check node message is six bits wide, than the bit width of a memory word is $360 \times 6 = 2160$ bits.

5 [0066] Another illustrative embodiment of the inventive concept is shown in FIG. 29. In this illustrative embodiment an integrated circuit (IC) 705 for use in a receiver (not shown) includes an LDPC decoder 720 and at least one register 710, which is coupled to bus 751. Illustratively, IC 705 is an integrated demodulator/decoder. However, only those portions of IC 705 relevant to the inventive concept are shown. For example, analog-digital converters, filters, decoders, etc., are not shown for simplicity. Bus 751 provides communication to, and from, other components of the receiver as represented by processor 750. Register 710 is representative of one, or more, registers, of IC 705, where each register comprises one, or more, bits as represented by bit 709 for controlling the operation of IC 705. The registers, or portions thereof, of IC 705 may be read-only, write-only or read/write. LDPC decoder 720 is coupled to register 710 via internal bus 711, which is representative of other signal paths and/or components of IC 705 for interfacing LDPC decoder 720 to register 710 as known in the art. In accordance with the principles of the invention, LDPC decoder 720 includes the above-described group CPUs and group BPUs. In the context of FIG. 14, IC 705 receives an IF signal 701 (e.g., signal 116 of FIG. 14) for processing via an input pin, or lead, of IC 705. A derivative of this signal, 702, is applied to LDPC decoder 720 for LDPC decoding in accordance with the principles of the invention as described above (e.g., FIGs. 15 and 24). LDPC decoder 720 provides signal 721, which is an LDPC decoded bit stream. IC 705 provides one, or more, recovered signals, as represented by signal 706. For example signal 706 is representative of signal 136 from a BCH decoder (not shown) of IC 705. In another variation of IC 705, signal 706 is representative of signal 106 of FIG. 13.

10 20 25 30 [0067] As described above, and in accordance with the principles of the invention, an LDPC decoder is described and shown that is capable of handling a variety of different code rates. Also, it should be noted that the above-described circularly shifted identity matrix

could be equivalently generalized to a permutation matrix. In this case, the above-described cyclic shifter is replaced with a permutation network.

[0068] In view of the above, it should be noted that although described in the context of a satellite communications system, the inventive concept is not so limited. For example, the 5 elements of FIG. 13 may represent other types of systems and other forms of multi-media endpoints. For example, satellite radio, terrestrial broadcast, cable TV, etc. Also, although described herein in the context of a single demodulator, it should be realized that the inventive concept is applicable to multi-modulation receivers, where information may be conveyed on different signal layers. For example, layered modulation receivers, hierarchical 10 modulation receivers, or combinations thereof. Indeed, the invention is applicable to any type of receiver in which LDPC decoding is performed. As such, the inventive concept is not limited to the decoding of DVB-S2 LDPC codes.

[0069] As such, the foregoing merely illustrates the principles of the invention and it will thus be appreciated that those skilled in the art will be able to devise numerous alternative 15 arrangements which, although not explicitly described herein, embody the principles of the invention and are within its spirit and scope. For example, although illustrated in the context of separate functional elements, these functional elements may be embodied on one or more integrated circuits (ICs). Similarly, although shown as separate elements, any or all of the elements may be implemented in a stored-program-controlled processor, e.g., a digital signal 20 processor (DSP) or microprocessor that executes associated software, e.g., corresponding to one or more of the elements shown in FIGs. 14, 15 and/or 24, etc. Further, although shown as separate elements, the elements therein may be distributed in different units in any combination thereof. For example, receiver 105 may be a part of TV 90 or receiver 105 may be located further upstream in a distribution system, e.g., at a head-end, which then 25 retransmits the content to other nodes and/or receivers of a network. It is therefore to be understood that numerous modifications may be made to the illustrative embodiments and that other arrangements may be devised without departing from the spirit and scope of the present invention as defined by the appended claims.

CLAIMS

1. A method for use in a receiver, the method comprising:
receiving low density parity check (LDPC) encoded data; and
processing the received LDPC encoded data using check node messages and bit node
5 messages to provide decoded data;
wherein the processing step partitions the bit node messages into Y groups and the
check node messages into q groups, where q varies as a function of a code rate associated with
the received LDPC encoded data.
- 10 2. The method of claim 1, wherein the received LDPC encoded data is derived from a
received digital video broadcasting system-2 signal.
- 15 3. The method of claim 1, wherein the received LDPC encoded data is representative
of an (N, K) LDPC code, where $M = N - K$, and $q = Y(N - K)/N$.
- 15 4. The method of claim 3, wherein $Y = N/360$.
- 20 5. The method of claim 1, wherein the received LDPC encoded data is representative
of an (N, K) LDPC code having a parity matrix of dimensions $M \times N$, and wherein the
processing step includes:
processing each group of check node messages with J processors; and
processing each group of bit node messages with J processors;
wherein J represents dimensions of a square sub-matrix, such that an integral number
of square sub-matrices fit into the parity matrix.
- 25 6. The method of claim 5, wherein the processing the check node messages step
includes the steps of:
cyclically shifting each group of check node messages;
processing each cyclically shifted group of check node message with J processors to
30 provide a group of new messages; and
cyclically shifting each group of new messages to form a group of bit node messages.

7. The method of claim 5, wherein the processing the bit node messages step includes the steps of:

 cyclically shifting each group of bit node messages;

 processing each cyclically shifted group of bit node message with J processors to

5 provide a group of new messages; and

 cyclically shifting each group of new messages to form a group of check node messages.

8. The method of claim 1, wherein the received LDPC encoded data is representative

10 of an (N, K) LDPC code having a parity matrix of dimensions $M \times N$, and wherein the processing step includes:

 processing each group of check node messages with J processors; and

 processing each group of bit node messages with J processors;

 wherein $J = N / Y$.

15

9. The method of claim 8, wherein the processing the check node messages step includes the steps of:

 cyclically shifting each group of check node messages;

 processing each cyclically shifted group of check node message with J processors to

20 provide a group of new messages; and

 cyclically shifting each group of new messages to form a group of bit node messages.

10. The method of claim 8, wherein the processing the bit node messages step includes the steps of:

25 cyclically shifting each group of bit node messages;

 processing each cyclically shifted group of bit node message with J processors to

provide a group of new messages; and

 cyclically shifting each group of new messages to form a group of check node messages.

30

11. Apparatus for use in a receiver, the apparatus comprising:

 a demodulator for providing low density parity check (LDPC) encoded data;

 an LDPC decoder for decoding the LDPC encoded data to provide decoded data;

wherein the LDPC decoder processes the LDPC encoded data by partitioning bit node messages into Y groups and check node messages into q groups, where q varies as a function of a code rate associated with the LDPC encoded data.

5 12. The apparatus of claim 11, wherein the LDPC encoded data is derived from a received digital video broadcasting system-2 signal.

13. The apparatus of claim 11, wherein the LDPC encoded data is representative of an (N, K) LDPC code, where $M = N - K$, and $q = Y(N - K)/N$.

10 14. The apparatus of claim 13, wherein $Y = N/360$.

15 15. The apparatus of claim 11, wherein the LDPC encoded data is representative of an (N, K) LDPC code having a parity matrix of dimensions $M \times N$, and wherein the LDPC encoder comprises:

J processors for processing each group of bit node messages; and
J processors for processing each group of check node messages;
wherein J represents dimensions of a square sub-matrix, such that an integral number of square sub-matrices fit into the parity matrix.

20 16. The apparatus of claim 11, wherein the LDPC encoded data is representative of an (N, K) LDPC code having a parity matrix of dimensions $M \times N$, and wherein the LDPC encoder comprises:

J processors for processing each group of bit node messages; and
J processors for processing each group of check node messages;
wherein $J = N/Y$.

30 17. The apparatus of claim 11, wherein the LDPC decoder comprises:
a memory for storing the check node messages and the bit node messages; and
a cyclic shifter for shifting the check node messages;
a group of bit node processors for processing the cyclically shifted check node messages to provide new messages;
a cyclic shifter for shifting the new messages to form new bit node messages;

a group of check node processors for processing the bit node messages to provide new check node messages for storage in the memory.

wherein the memory is structured such that new bit node messages are stored consecutively.

5

18. The apparatus of claim 11, wherein the LDPC decoder comprises:
a memory for storing the check node messages and the bit node messages; and
a group of bit node processors for processing the check node messages to provide new bit node messages for storage in the memory; and

10 a cyclic shifter for shifting the bit node messages;
a group of check node processors for processing the cyclically shifted bit node messages to provide new messages; and
a cyclic shifter for shifting the new messages to form new check node messages;
wherein the memory is structured such that new check node messages are stored
15 consecutively.

19. The apparatus of claim 11, wherein the LDPC decoder comprises a memory organized such that groups of bit node messages are stored consecutively.

20 20. The apparatus of claim 11, wherein the LDPC decoder comprises a memory organized such that groups of check node messages are stored consecutively.

1/29

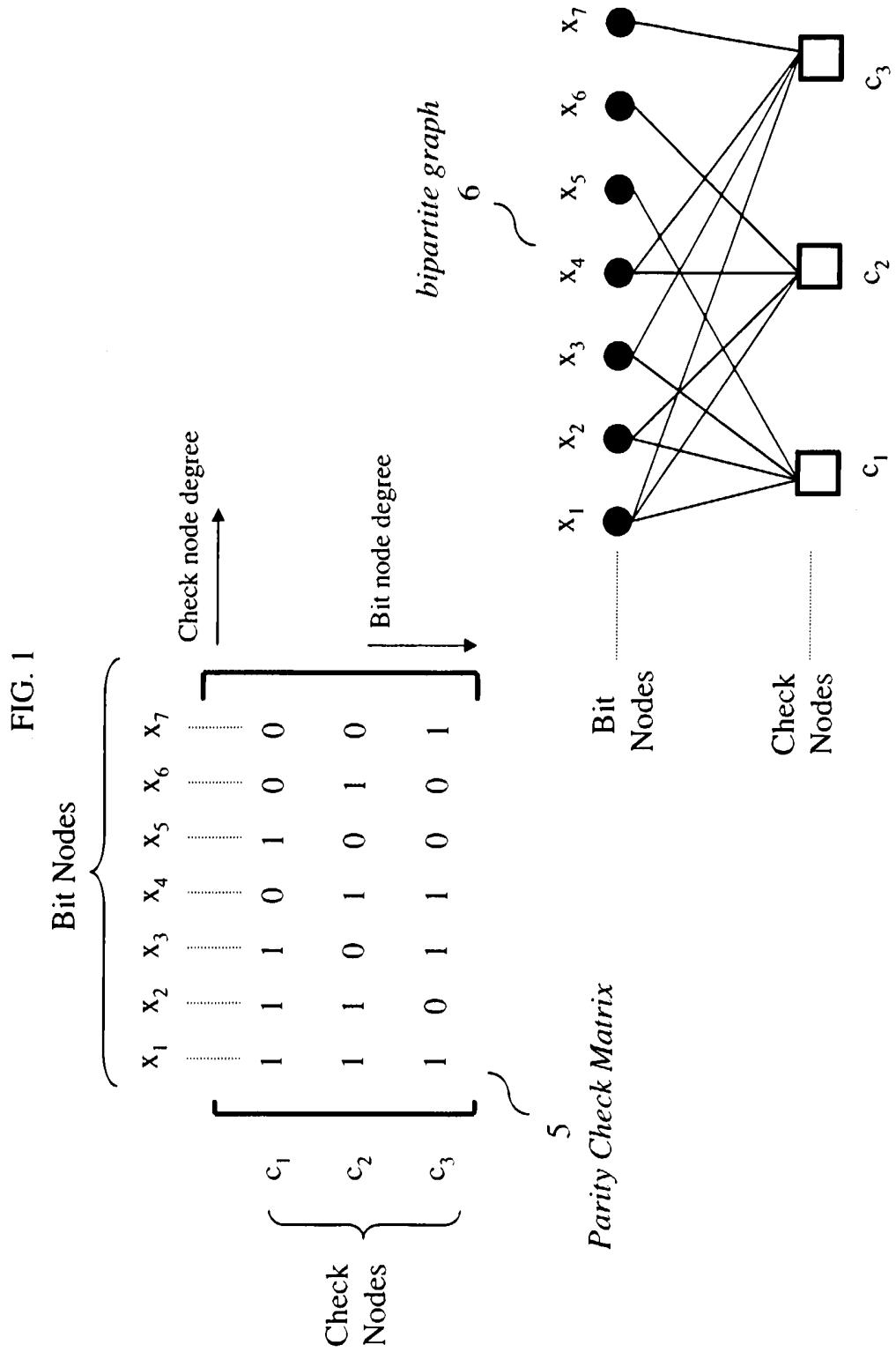


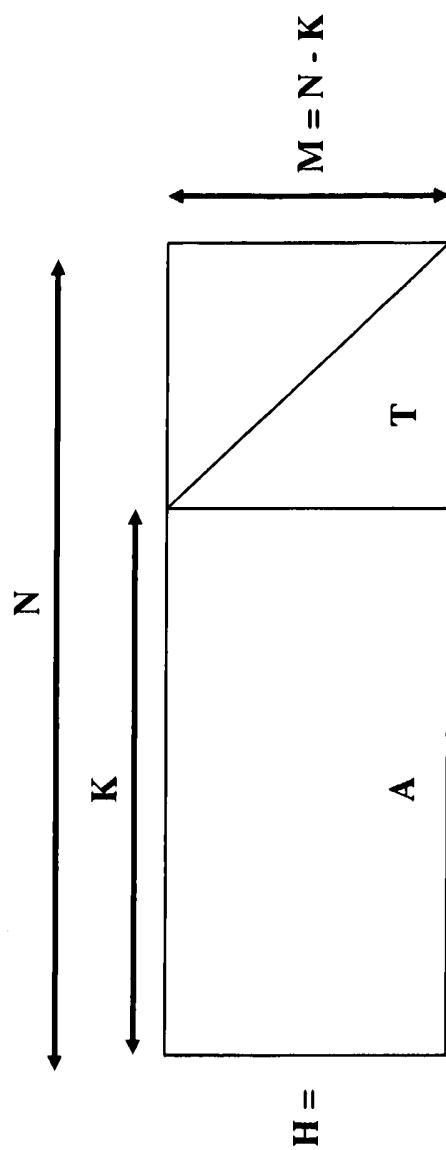
FIG. 2

Table One

Rate	K
1/4	16200
1/3	21600
2/5	25920
1/2	32400
3/5	38880
2/3	43200
3/4	48600
4/5	51840
5/6	54000
8/9	57600
9/10	58320

3/29

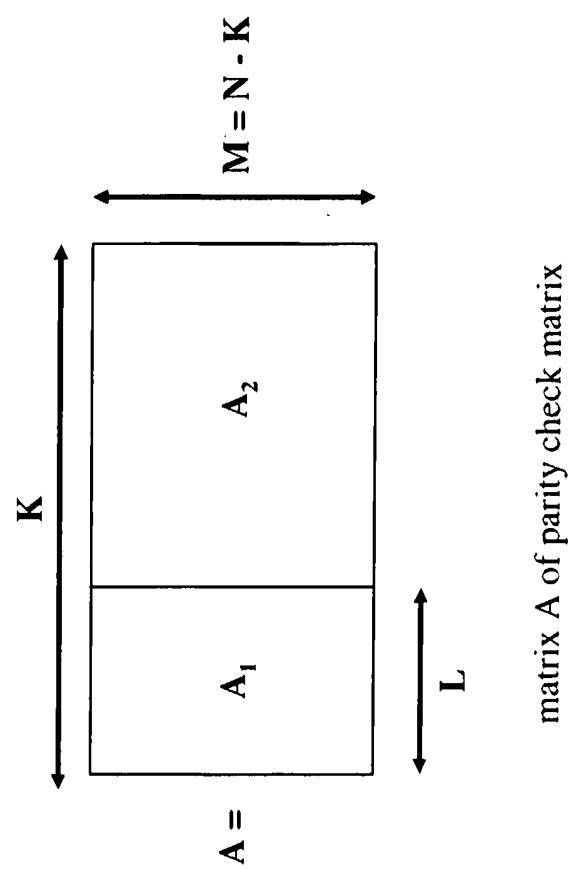
FIG. 3



Parity check matrix of LDPC codes in DVB-S2

4/29

FIG. 4



matrix A of parity check matrix

5/29

FIG. 5

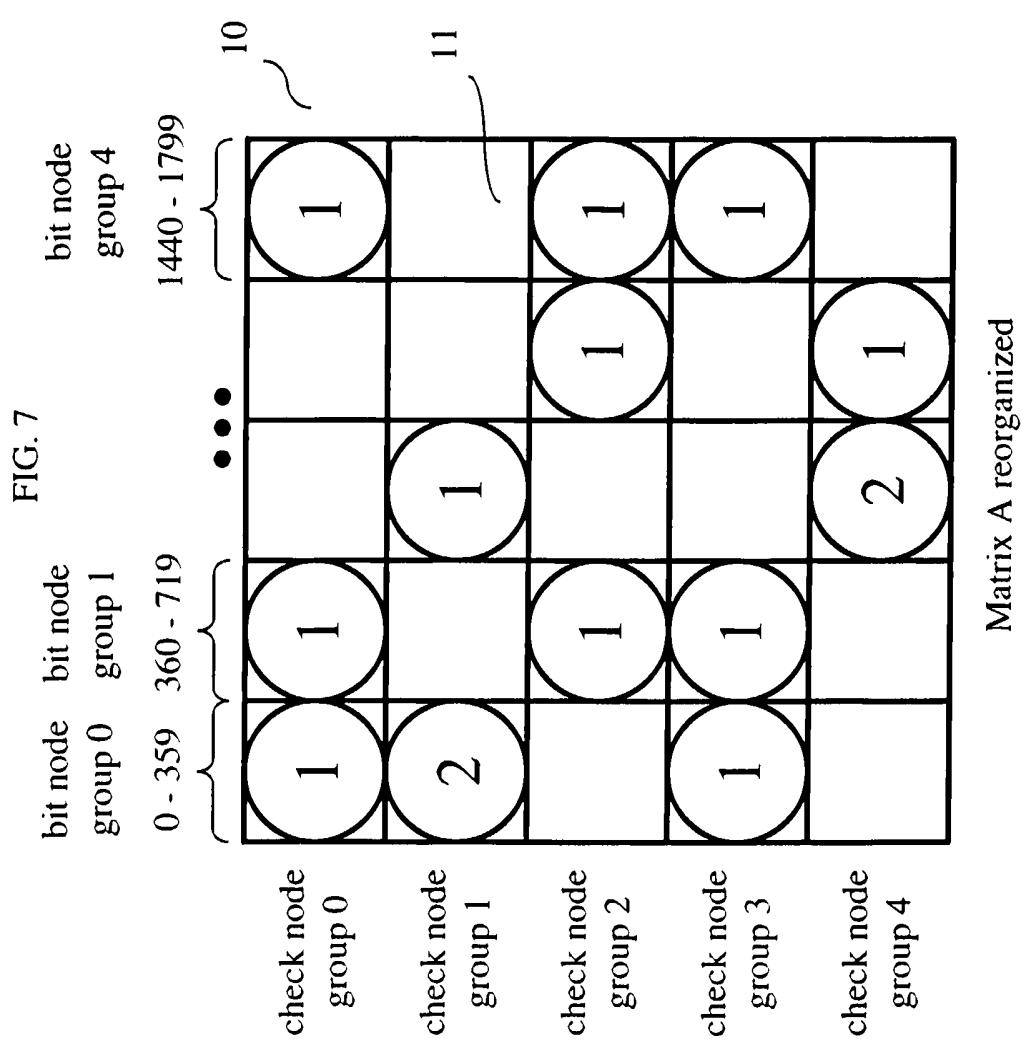
$$\begin{bmatrix} 1 & & & & & & & \\ 1 & 1 & & & & & & \\ & & \bullet & \bullet & \bullet & & & \\ & & 1 & & & & & \\ & & & \bullet & \bullet & \bullet & & \\ & & & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix}$$

matrix T of parity check matrix

6/29

FIG. 6
Table Two

Rate	K	L	DV ₁	q	D _C
1/4	16200	15	12	135	4
1/3	21600	20	12	120	5
2/5	25920	19	12	108	6
1/2	32400	36	8	90	7
3/5	38880	36	12	72	11
2/3	43200	12	13	60	10
3/4	48600	15	12	45	14
4/5	51840	18	10	36	18
5/6	54000	15	13	30	22
8/9	57600	20	4	20	27
9/10	58320	18	4	18	30



8/29

FIG. 8
Cyclic Identity Matrices

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \xrightarrow{\text{left shift}} \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & 1 & \\ & & & 1 & \\ 1 & & & & \end{bmatrix} \xrightarrow{\text{left shift}} \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & 1 & \\ & & & 1 & \\ 1 & & & & \end{bmatrix} \xrightarrow{\text{left shift}} \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & 1 & \\ & & & 1 & \\ 1 & & & & \end{bmatrix}$$

offset of zero (Identity Matrix) 21

offset of one 22

offset of two 23

24

9/29

FIG. 9
 Combined Cyclic Identity Matrices
 (e.g., two Cyclic Identity Matrices combined together)

$$\begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & & 1 & \\ & 1 & & & 1 & \\ & & 1 & & 1 & \\ & & & 1 & & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

offset of zero and one combined 26
 offset of one and two combined 27
 offset of zero and two combined 28

10/29

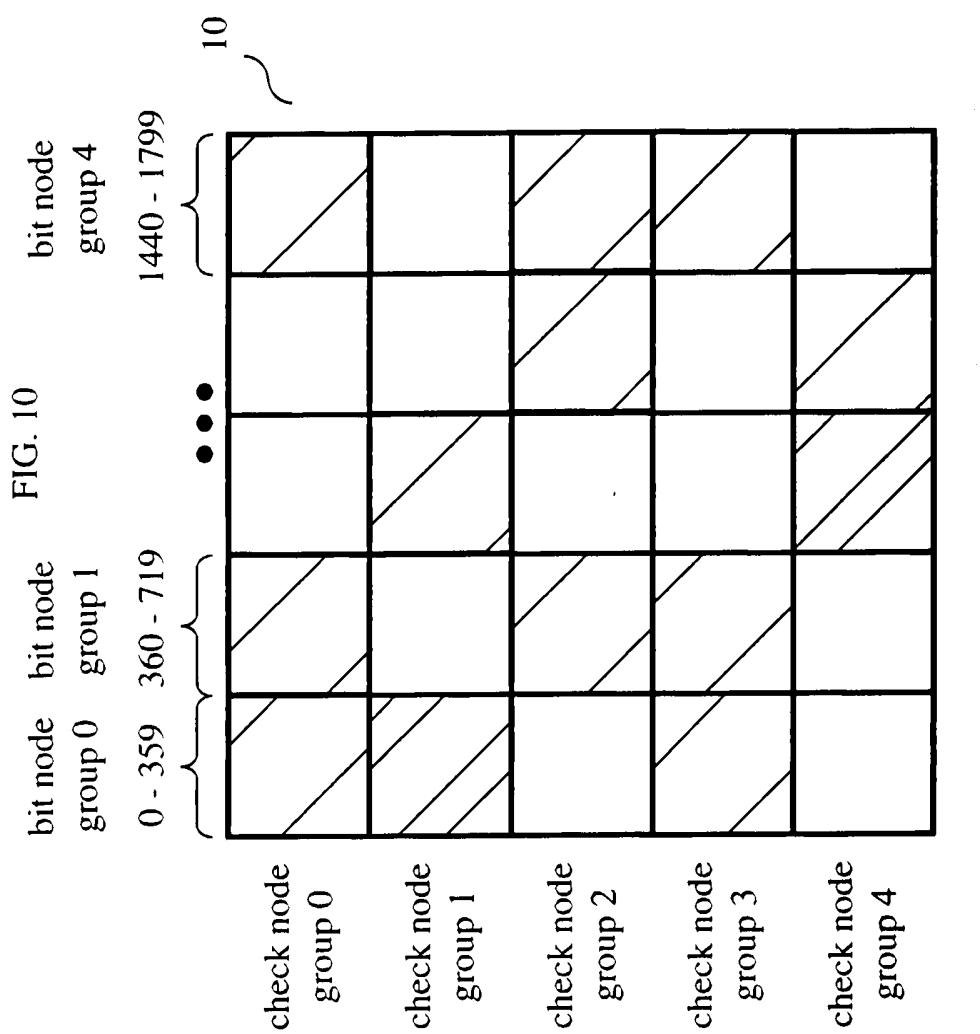
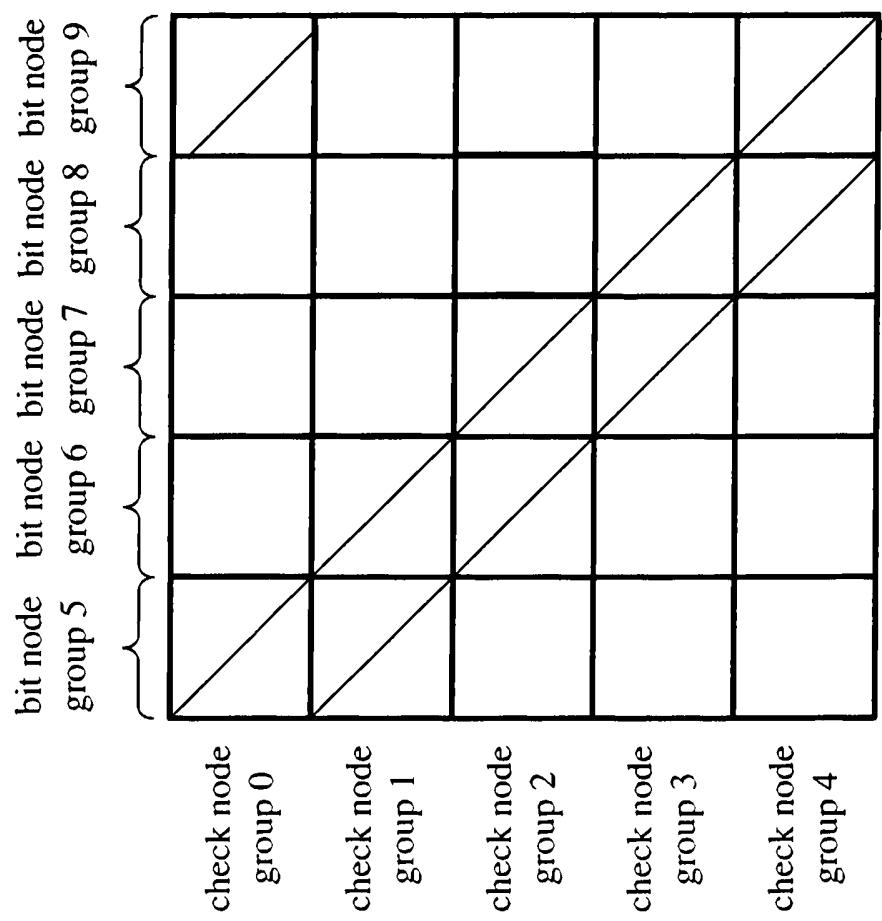


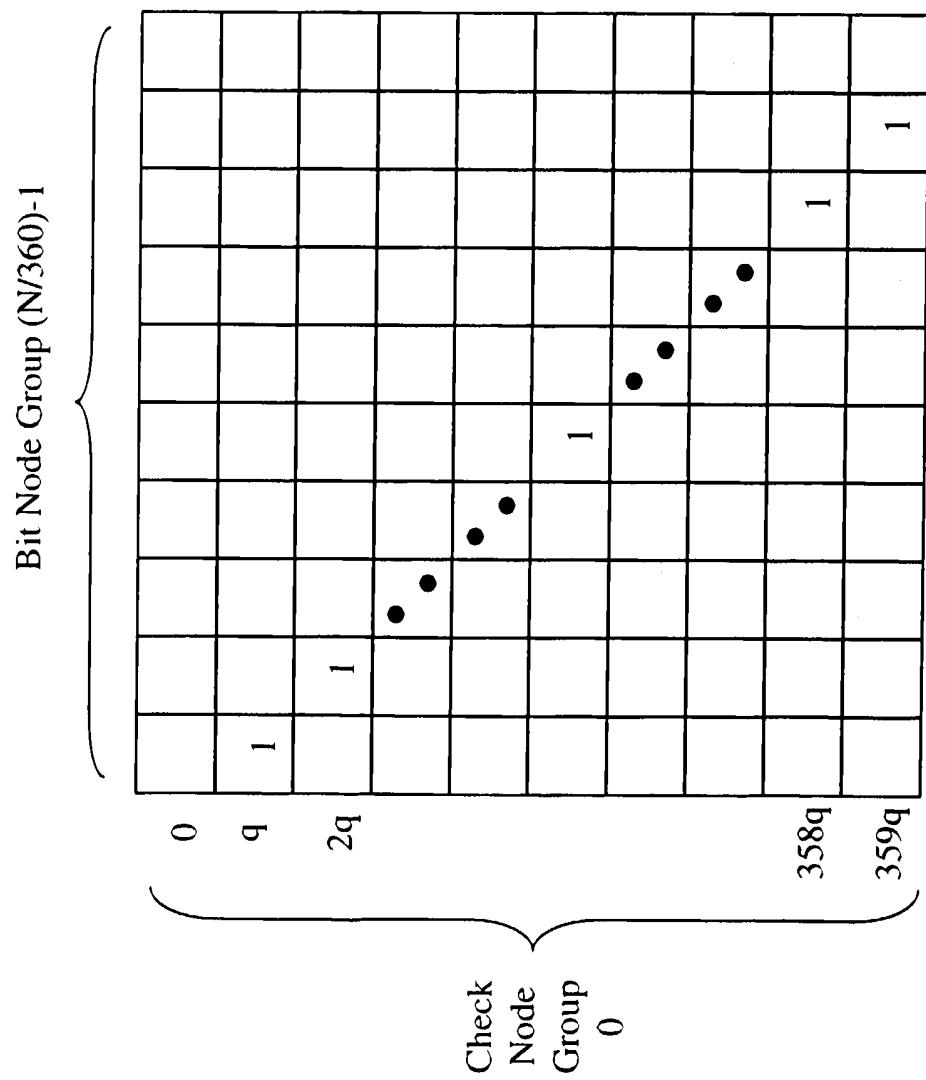
FIG. 11



Matrix T reorganized

12/29

FIG. 12



13/29

FIG. 13

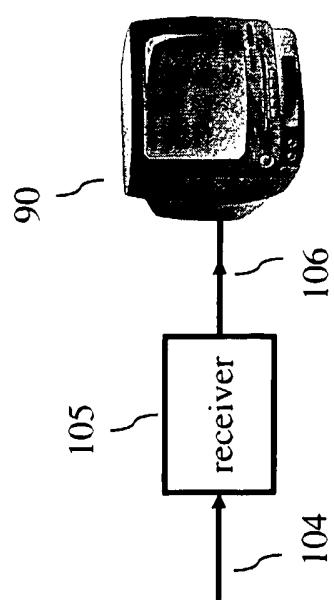
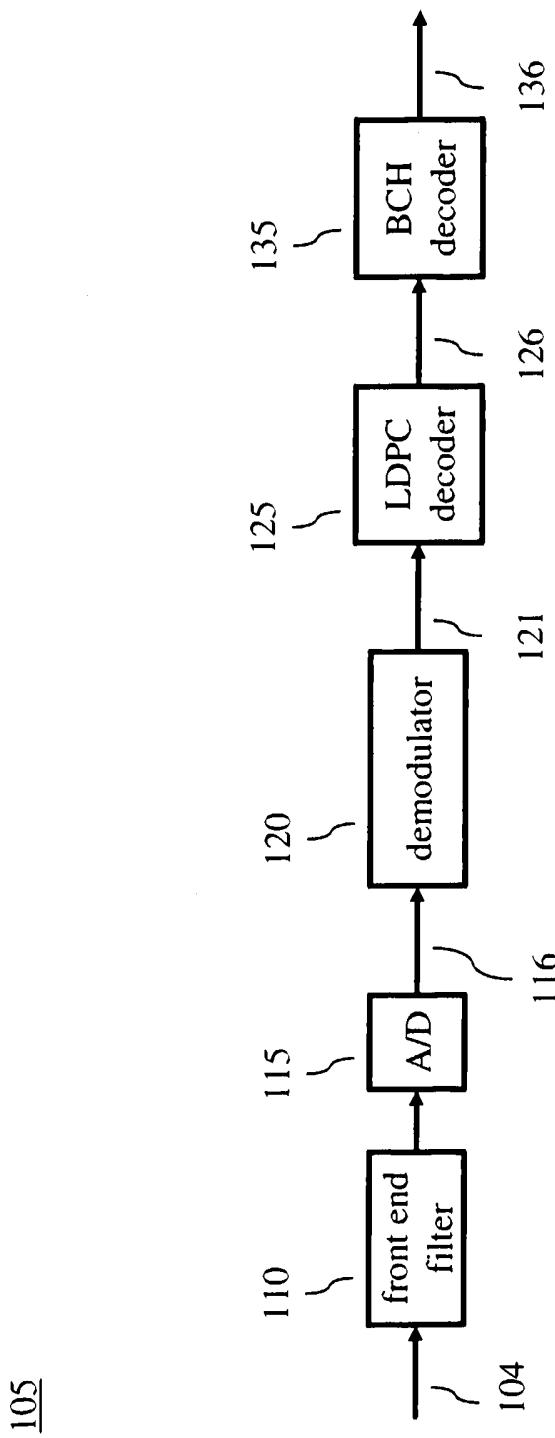
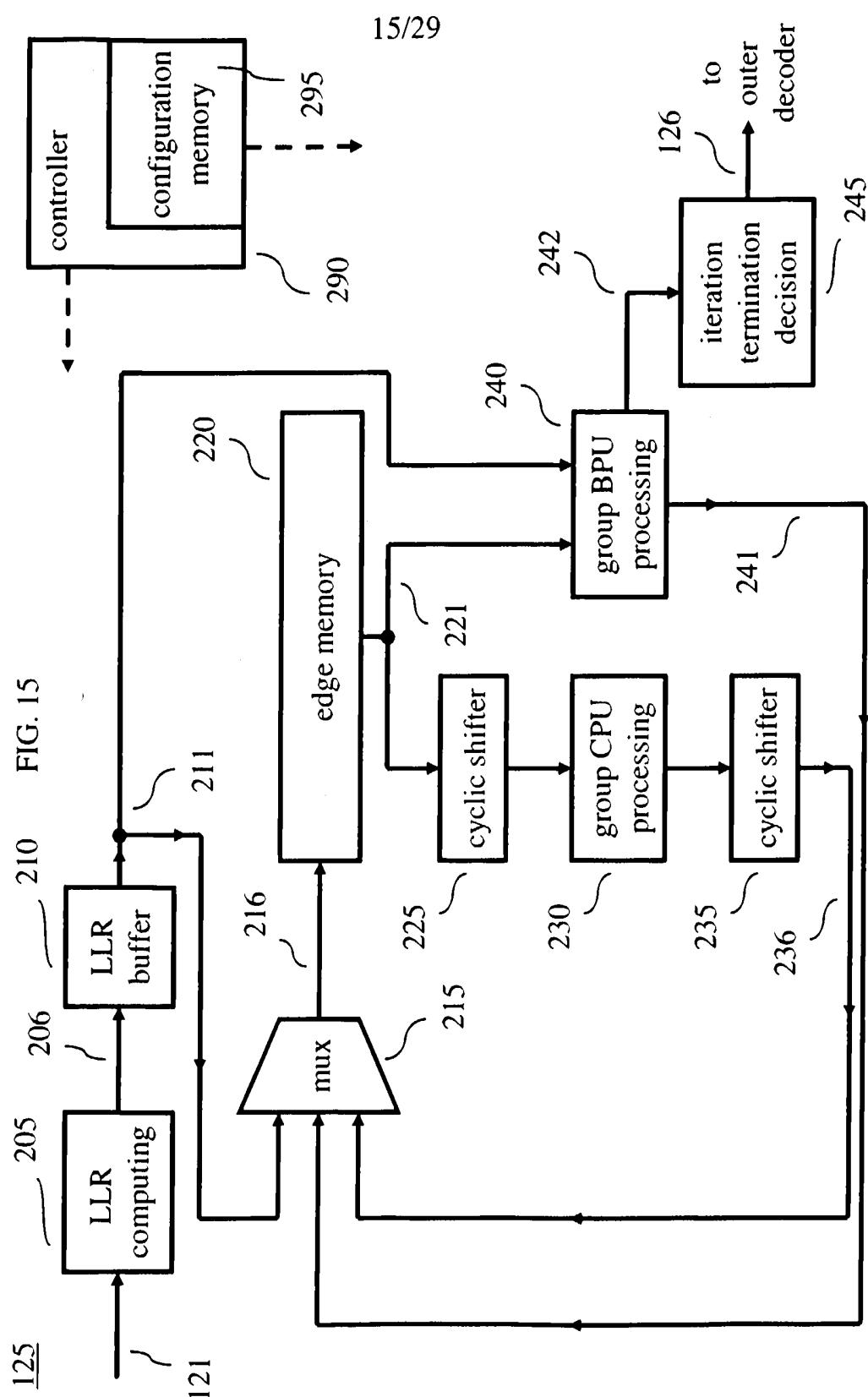


FIG. 14





16/29

FIG. 16
315

Bit Node Group 0	0	1	2	...	358	359
Bit Node Group 1	360	361	362	...	718	719
Bit Node Group 2	720	721	722	...	1078	1079
	•	•	•	•	•	•
	•	•	•	•	•	•
	•	•	•	•	•	•
Bit Node Group m	360m	360m+1	360m+2	...	360m+358q	360m+359q
	•	•	•	•	•	•
	•	•	•	•	•	•
	•	•	•	•	•	•
Bit Node Group K/360-2	K-720	K-719	K-718	...	K-362	K-361
Bit Node Group K/360-1	K-360	K-359	K-358	...	K-2	K-1

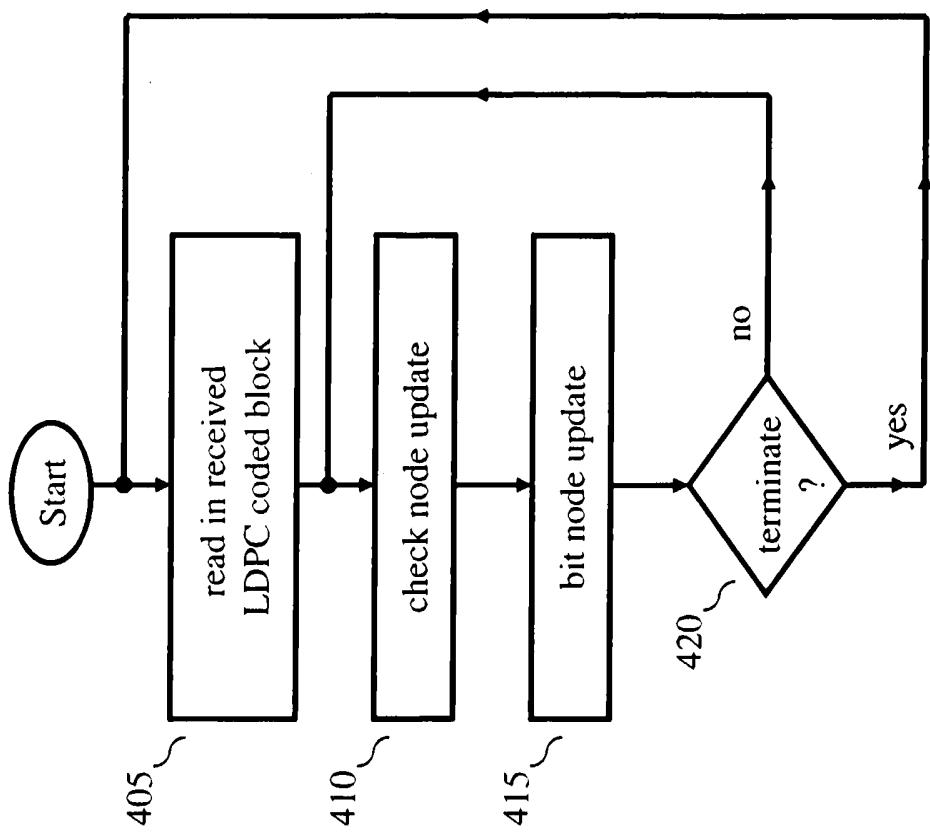
FIG. 17

17/29

Bit Node Group K/360	K	K+q	K+2q	•••	K+358q	K+359q
Bit Node Group K/360+1	K+1	K+1+q	K+1+2q	•••	K+1+358q	K+1+359q
Bit Node Group K/360+2	K+2	K+2+q	K+2+2q	•••	K+2+358q	K+2+359q
Bit Node Group K/360+m	K+m	K+m+q	K+m+2q	•••	K+m+358q	K+m+359q
Bit Node Group N/360-2	K-720	K-719	K-718	•••	K+q-2+358q	K+q-2+359q
Bit Node Group N/360-1	K-360	K-359	K-358	•••	K+q-1+358q	K+q-1+359q

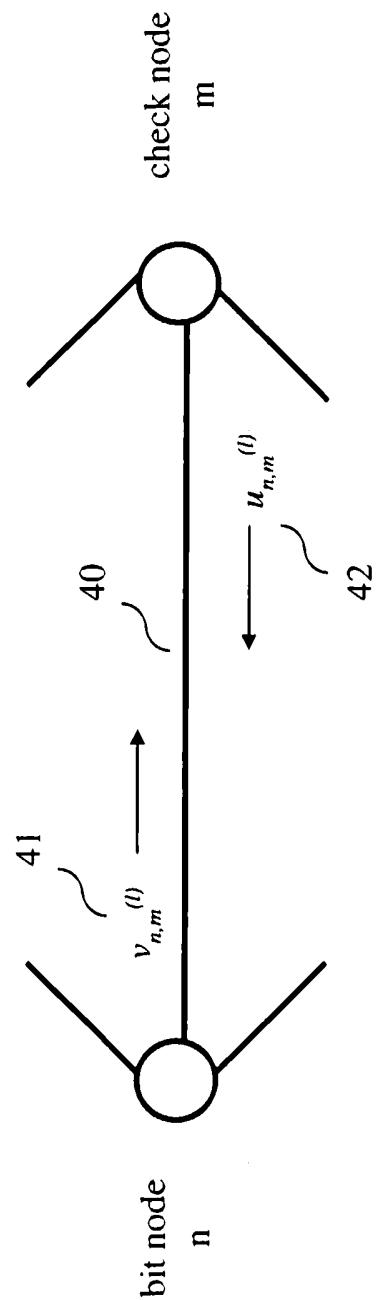
18/29

FIG. 18



19/29

FIG. 19



20/29

325

FIG. 20

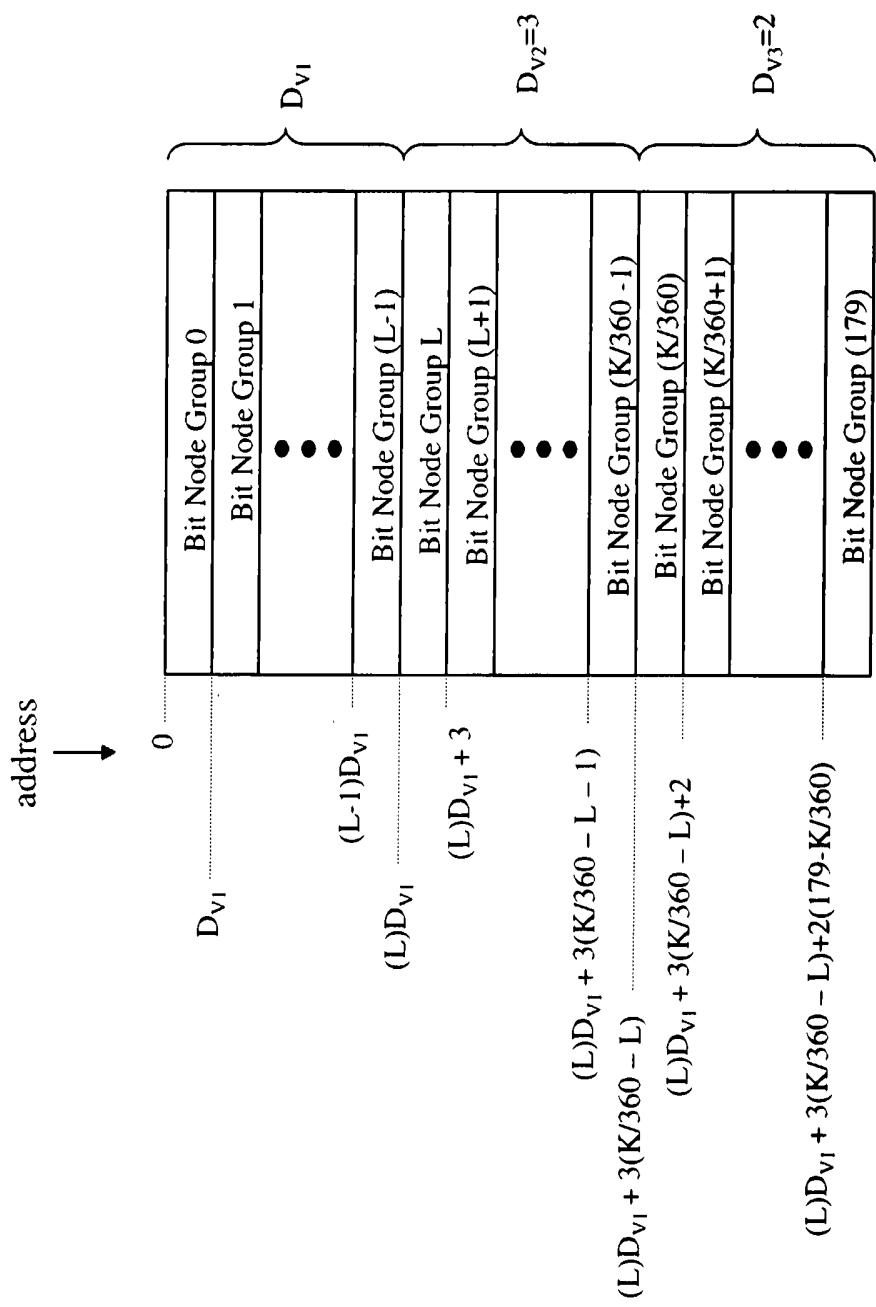
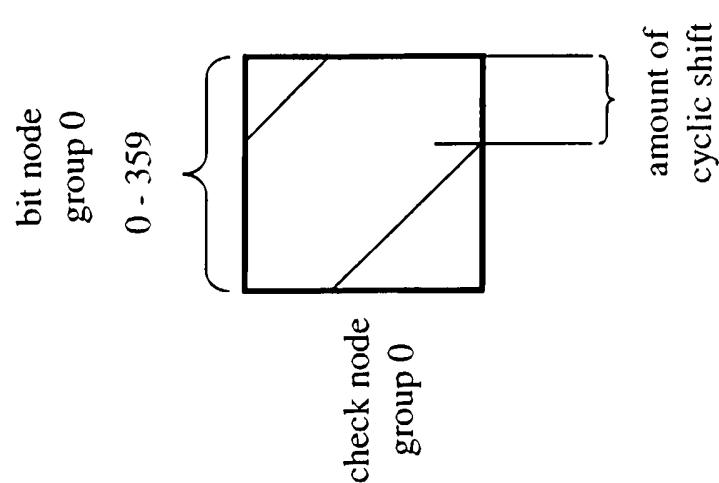
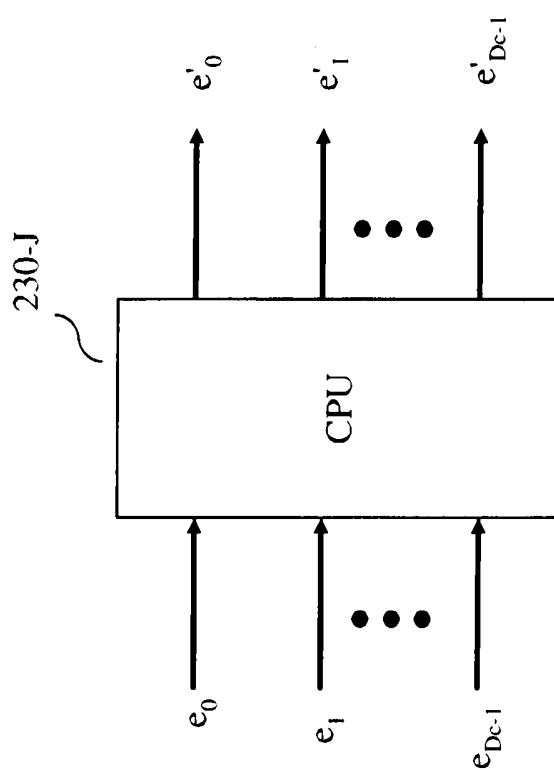


FIG. 21



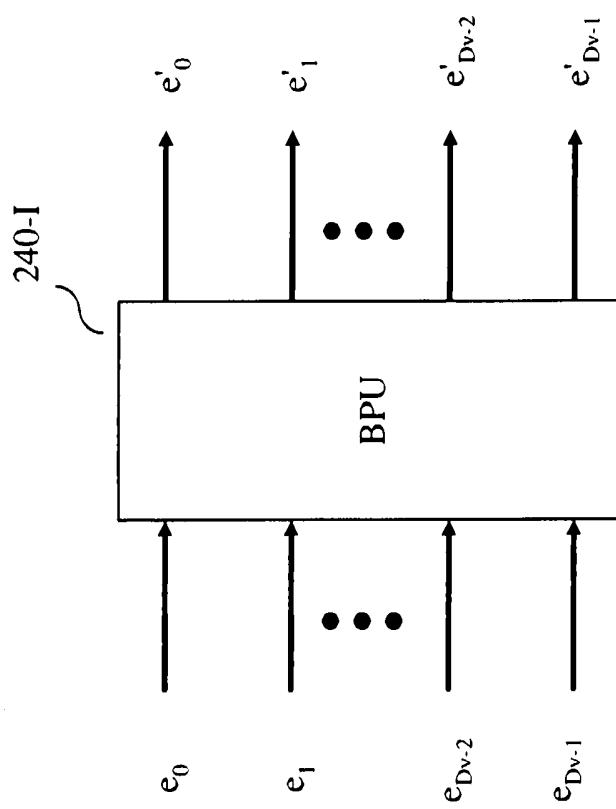
22/29

FIG. 22



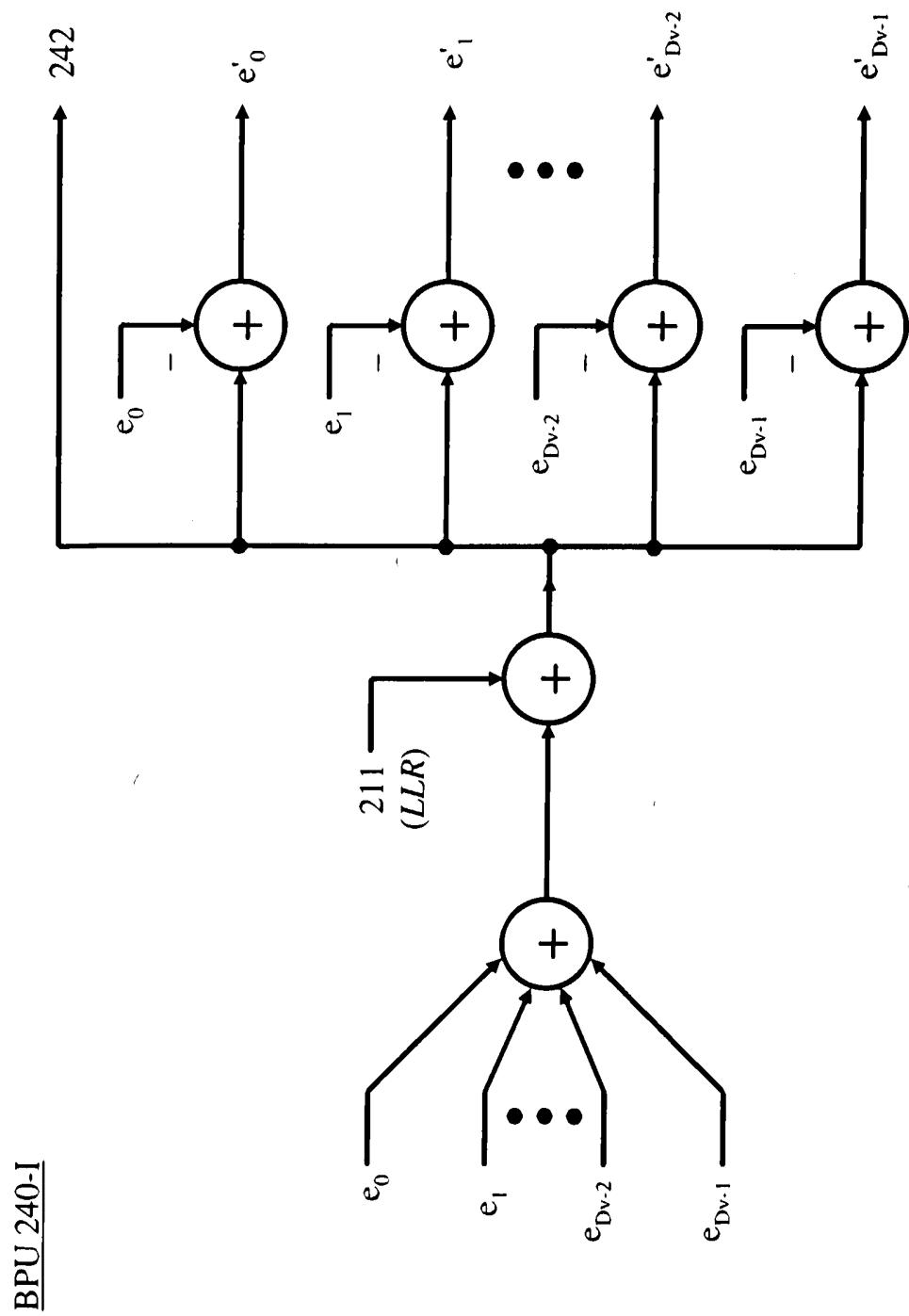
23/29

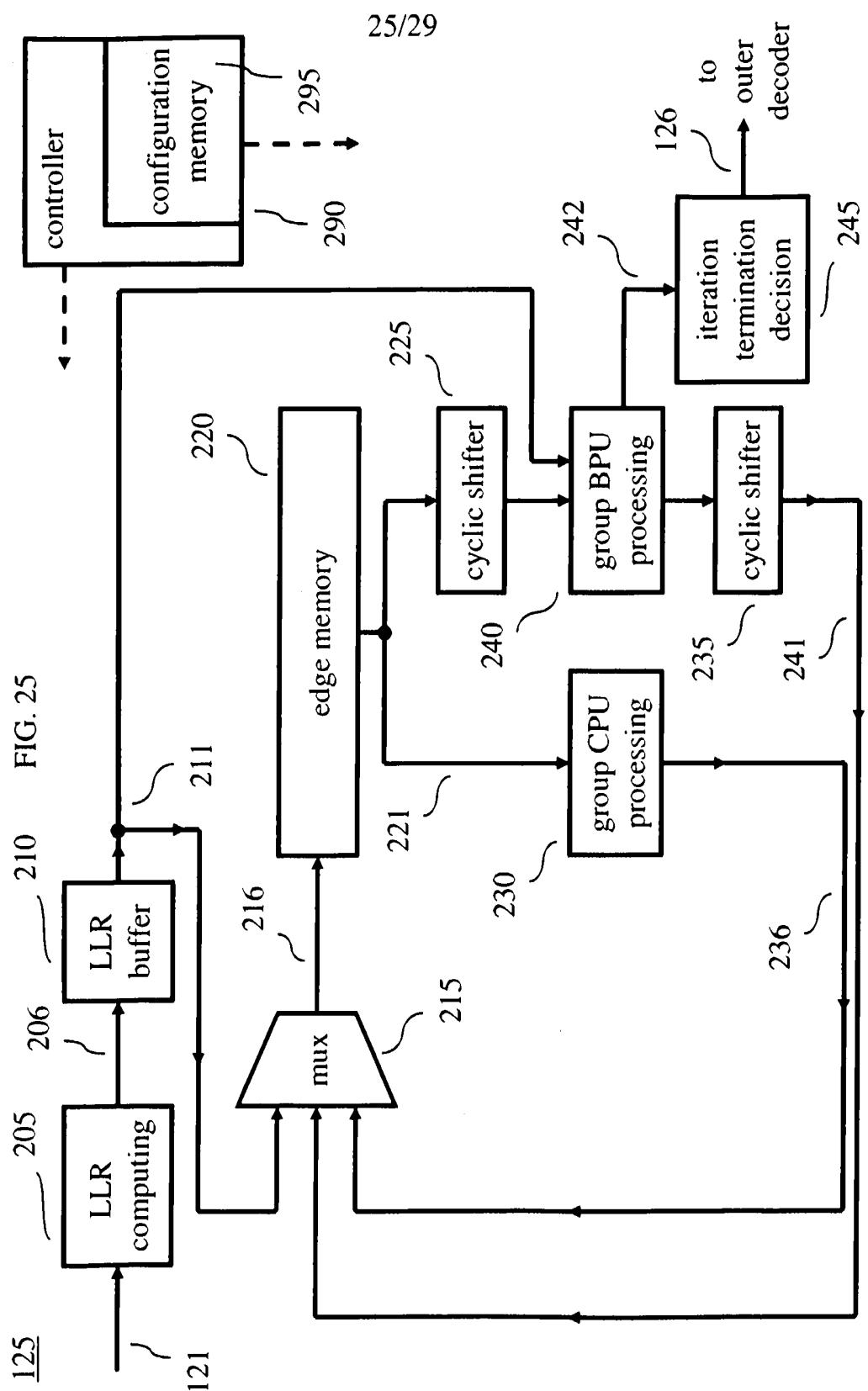
FIG. 23



24/29

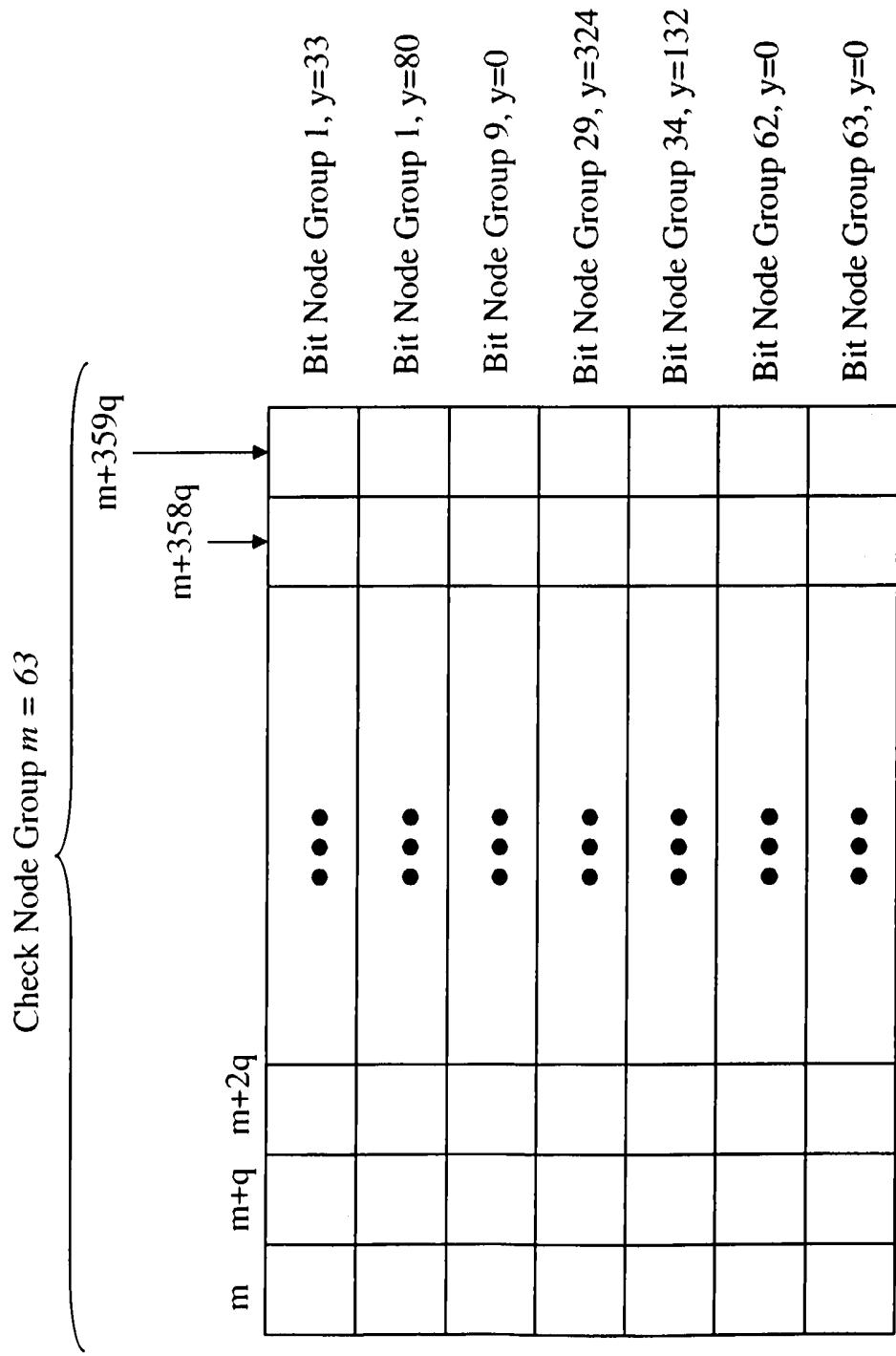
FIG. 24





305

FIG. 26



27/29

310

address

Check Node Group 0	0	0	q	2q	...	358q	359q
Check Node Group 1	D_C	1	$1+q$	$1+2q$...	$1+358q$	$1+359q$
Check Node Group 2	$2D_C$	2	$2+q$	$2+2q$...	$2+358q$	$2+359q$
				
Check Node Group m	mD_C	m	$m+q$	$2+2q$...	$m+358q$	$m+359q$
				
Check Node Group $q-2$	$(q-2)D_C$	$q-2+q$	$q-2+2q$...		$q-2+358q$	$q-2+359q$
Check Node Group $q-1$	$(q-1)D_C$	$q-1+q$	$q-1+2q$...		$q-1+358q$	$q-1+359q$

28/29

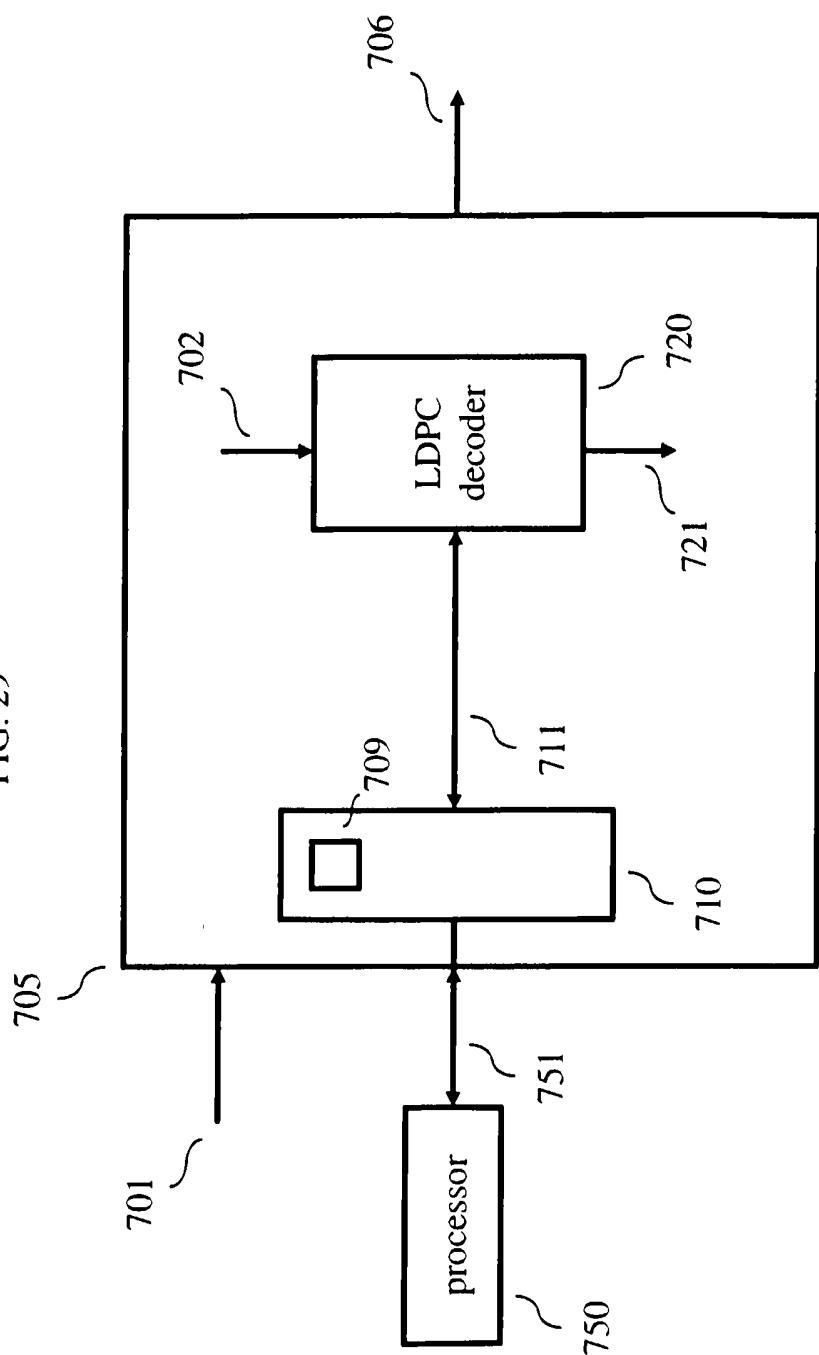
FIG. 28

Table Three

Rate	q	D _c	Memory Space (qxD _c)
1/4	135	4	540
1/3	120	5	600
2/5	108	6	648
1/2	90	7	630
3/5	72	11	792
2/3	60	10	600
3/4	45	14	630
4/5	36	18	648
5/6	30	22	660
8/9	20	27	540
9/10	18	30	540

29/29

FIG. 29



INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/033342

A. CLASSIFICATION OF SUBJECT MATTER
H03M13/11 H03M13/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>"DIGITAL VIDEO BROADCASTING (DVB); SECOND GENERATION FRAMING STRUCTURE, CHANNEL CODING AND MODULATION SYSTEMS FOR BROADCASTING, INTERACTIVE SERVICES, NEWS GATHERING AND OTHER BROADBAND SATELLITE APPLICATIONS"</p> <p>ETSI STANDARDS, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE, SOPHIA-ANTIPOLIS, FR, no. V111, June 2004 (2004-06), pages 1-74, XP002311764</p> <p>ISSN: 0000-0001</p> <p>cited in the application paragraph '5.3.2!; figure 1</p> <p>-----</p> <p style="text-align: center;">-/-</p>	1,11

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

6 December 2005

Date of mailing of the international search report

30/12/2005

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Van Staveren, M

INTERNATIONAL SEARCH REPORT

National Application No
/US2005/033342

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	D.J.C. MACKAY, S.T. WILSON AND M.C. DAVEY: "Comparison of Constructions of Irregular Gallager Codes" IEEE TRANSACTIONS ON COMMUNICATIONS, vol. 47, October 1999 (1999-10), pages 1449-1454, XP002357502 cited in the application the whole document	1,11
A	D. SRIDHARA, T. FUJA, R.M. TANNER: "LOW DENSITY PARITY CHECK CODES FROM PERMUTATION MATRICES" 2001 CONFERENCE ON INFORMATION SCIENCES AND SYSTEMS, 21 March 2001 (2001-03-21), page 1, XP002357503 cited in the application the whole document	1,11
A	RYAN E W: "An introduction to LDPC codes" CRC HANDBOOK FOR CODING AND SIGNAL PROCESSING FOR RECORDING SYSTEM, 19 August 2003 (2003-08-19), pages 1-23, XP002336953 the whole document	1,11
A	HU X-Y ET AL: "EFFICIENT IMPLEMENTATION OF THE SUM-PRODUCT ALGORITHM FOR DECODING LDPC CODES" GLOBECOM'01. 2001 IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE. SAN ANTONIO, TX, NOV. 25 - 29, 2001, IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE, NEW YORK, NY : IEEE, US, vol. VOL. 2 OF 6, 25 November 2001 (2001-11-25), pages 1036-1036E, XP001099262 ISBN: 0-7803-7206-9 the whole document	1,11