BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS	BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE	CHAMBRES DE RECOURS DE L'OFFICE EUROPEEN DES BREVETS

Internal distribution code:

(A) [ ] Publication in OJ
(B) [ ] To Chairmen and Members
(C) [ ] To Chairmen
(D) [X] No distribution

Datasheet for the decision
of 16 January 2012

Case Number:	T 2048/08 - 3.5.02
Application Number:	04254015.3
Publication Number:	1494358
IPC:	H03M 13/11
Language of the proceedings:	EN

Title of invention:
Method and system for generating parallel decodable low density parity check (LDPC) codes

Applicant:
DTVG LICENSING, INC.

Opponent:
-

Headword:
-

Relevant legal provisions:
EPC Art. 56

Relevant legal provisions (EPC 1973):
-

Keyword:
"Inventive step - no"

Decisions cited:
-

Catchword:
-

Case Number: T 2048/08 **-** 3.5.02

DECISION
of the Technical Board of Appeal 3.5.02
of 16 January 2012

Appellant:
(Applicant)

DTVG LICENSING, INC
2230 East Imperial Highway
El Segundo CA 90245 (US)

Representative:

Jackson, Richard Eric
Carpmaels & Ransford
One Southampton Row
London WC1B 5HA (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 29 May 2008 refusing European patent application No. 04254015.3 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman: M. Ruggiu
Members: M. Rognoni
R. Moufang

Summary of Facts and Submissions

I. The appellant (applicant) appealed against the decision of the examining division refusing European patent application No. 04 254 015.3.

II. In the contested decision, the examining division found, ***inter alia***, that the subject-matter of the method claim 1 and the subject-matter of the corresponding apparatus claim 9 then on file lacked novelty with respect to the following document:

D1: US-A-2003/0033575.

III. In a communication dated 7 September 2011 summoning the appellant to oral proceedings, the Board addressed a number of issues concerning the appellant's request filed with the statement of grounds of appeal.

IV. With a letter dated 16 December 2011, the appellant filed a set of claims constituting a replacement main request and a set of claims 1 to 14 of an auxiliary request.

V. At the oral proceedings, which were held before the Board on 16 January 2012, the appellant withdrew the main request filed with the letter dated 16 December 2011.

VI. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of claims 1 to 14 of the auxiliary request filed with letter dated 16 December 2011.

VII. Claim 1 of the auxiliary request reads as follows:

"A method for supporting decoding of low density parity check (LDPC) codes, the method comprising:

constructing a mapped matrix corresponding to a parity check matrix associated with the LDPC codes, wherein the mapped matrix satisfies a plurality of parallel decodable conditions for permitting a lumped memory structure (901, 1101) that is accessible by a plurality of processors (903, 1103) operating in parallel; and

storing the mapped matrix in memory (901, 1101) for decoding the LDPC codes by the processors (903, 1103), wherein the parallel decodable conditions include:

none of any two different entries in an identical row of the mapped matrix connects to identical bit nodes or identical check nodes;

for a bit node group BG_i representing all bit nodes connected to edges in an ith row of the mapped matrix, and for BG_i and BG_j where $i \neq j$, ensuring that BG_i and BG_j either do not have any common node or are identical; and

for a check node group CG_i representing all the check nodes connected to edges in an ith row, and for CG_i and CG_j where $i \neq j$, ensuring that CG_i and CG_j either do not have any common node or are identical".

Claims 2 to 6 are directly or indirectly dependent on claim 1.

Claim 7 relates to a computer-readable medium bearing instructions for performing the method of claim 1. Claim 8 is directed to a decoding apparatus corresponding essentially to the method of claim 1. Claims 9 to 14 are directly or indirectly dependent on claim 8. As these claims have no bearing on the outcome of the appeal, their wording need not be specified.

VIII. The appellant essentially argued that D1 neither taught nor suggested to construct a mapped matrix corresponding to the parity check matrix associated with an LDPC code and fulfilling the three conditions set out in claim 1 of the auxiliary request. In D1, edge values were arranged in vectors which were subjected to a rotation each time the decoder switched from bit node processing to check node processing. As the arrangement of the edge values in the edge memory changed constantly, it did not define a mapped matrix according to the present invention. By providing a mapped matrix for edge values based on the parity check matrix of an LDPC code as specified in claim 1, the present invention permitted efficient decoding at the same time as permitting different numbers of parallel decoding engines to be implemented. For these reasons, the subject-matter of claim 1 of the auxiliary request was new and involved an inventive step.

Reasons for the Decision

1. The appeal is admissible.

2.1 Claim 1 according to the appellant's auxiliary request relates to a method for supporting decoding of low density parity check (LDPC) codes comprising the following steps:

(a) constructing a mapped matrix corresponding to a parity check matrix associated with the LDPC codes,

(b) wherein the mapped matrix satisfies a plurality of parallel decodable conditions for permitting a lumped memory structure that is accessible by a plurality of processors operating in parallel; and

(c) storing the mapped matrix in memory for decoding the LDPC codes by the processors,

(d) wherein the parallel decodable conditions include:

(d_1) none of any two different entries in an identical row of the mapped matrix connects to identical bit nodes or identical check nodes;

(d_2) for a bit node group BG_i representing all bit nodes connected to edges in an ith row of the mapped matrix, and for BG_i and BG_j where $i \neq j$, ensuring that BG_i and BG_j either do not have any common node or are identical; and

(d_3) for a check node group CG_i representing all the check nodes connected to edges in an ith row, and for CG_i and CG_j where $i \neq j$, ensuring that CG_i and CG_j either do not have any common node or are identical.

2.2 As explained in paragraph [55] of the filed application, the ***"mapped matrix"*** is stored in the edge memory and, in fact, ***"represents the memory structure.***

This memory can be physically different memory devices that are functionally equivalent to the mapped matrix". As to the three conditions (d_1), (d_2) and (d_3) to be satisfied by the entries of the mapped matrix, they are *"necessary to yield efficient parallel processing"*.

2.3 Paragraph [57] shows an example of an 8 x 16 parity check matrix used to construct a 9 x 4 mapped matrix which supports parallel decoding with four parallel processors according to the present application. As explained in paragraph [58], each nonzero entry of the parity check matrix is associated with an edge value. The edge values are labelled from e_1 to e_{36} starting at the top of the first column and continuing from top to bottom of the first column and of each successive column. The edge values are then mapped one-to-one to a matrix 9 x 4, so that the twelve edge values required to decode the first four bit nodes are located in the first three rows and the edge values for each of the following three groups of four bit nodes are located in the next two rows, respectively. As a result of the fact that the mapped matrix satisfies the conditions (d_1), (d_2) and (d_3) specified in claim 1, the twelve bit nodes or the eight check nodes can be processed in groups of four by reading into four parallel processors the four edge values located in one of the three (or two) corresponding rows.

In particular, *"all the edge values associated with bit node {1,2,3,4} reside in the first three rows of the edge matrix. Thus, in the next access cycle, either the second row or the third row will continue to be accessed. If the process begins with any other row, additional intermediate results will need to be* *buffered. After all the rows associated with bit node {1,2,3,4} are processed and the edge values are updated, the other groups of bit nodes are processed. Likewise, the above procedure is carried out at the check nodes. That is, the check nodes are processed in groups, and after all the edges associated with this group have been updated, the next group is processed. In each access cycle, precisely one value from the edge matrix is fetched for each node being actively processed, thereby ensuring that none of the processors 903 is over loaded (or under utilized, for that matter)"* (application as originally filed, paragraphs [63] and [64]).

3.1 D1 relates, ***inter alia***, to a method for supporting parallel decoding of low parity density check (LDPC) codes.

Figure 13 of D1 shows a 12 x 15 parity check matrix obtained by replacing each 1 in the 4 x 5 matrix H of Figure 7 with a cyclic permutation of a 3 x 3 identity matrix. The edge values corresponding to the thirty-six nonzero entries of the parity check matrix of Figure 13 are mapped as shown in Figure 14 (top diagram), so that the nine edge values associated with the first three bit nodes are arranged in the first three columns, the nine edges of the next three bit nodes are arranged in the following three columns and the edge values of each of the remaining three groups of three bit nodes are located in two of the following columns, respectively.

The decoder according to D1 stores the edge values in a vector edge memory and, as explained in paragraph [0143] of D1 (third sentence), the message edge memory vectors ($y_{1,i}$, $y_{2,i}$, $y_{3,i}$) ***"are read out in order and delivered to the vector node processor 1508 for variable node processing"***, whereby "i" in the given example varies from 1 to 12 and corresponds to the column index of the edge value arrangement in the top diagram of Figure 14. It is evident that the twelve vectors ($y_{1,i}$, $y_{2,i}$, $y_{3,i}$) constitute a one-to-one mapping that maps the nonzero entries of the parity check matrix of Figure 13 and thus correspond functionally to a "mapped matrix".

3.2 The edge values required to process the first group of three bit nodes are identified by the first group of three vectors ($y_{1,1}$, $y_{2,1}$, $y_{3,1}$), ($y_{1,2}$, $y_{2,2}$, $y_{3,2}$) and ($y_{1,3}$, $y_{2,3}$, $y_{3,3.}$). The processing of the following three bit nodes is based on the following edge value vectors ($y_{1,4}$, $y_{2,4}$, $y_{3,4}$), ($y_{1,5}$, $y_{2,5}$, $y_{3,5}$) and ($y_{1,6}$, $y_{2,6}$, $y_{3,6.}$), whereas the edge values needed to process the remaining three groups of three bit nodes correspond to the remaining three groups of two vectors. Similarly, in order to process the first group of check nodes it is required to address only vectors ($y_{1,1}$, $y_{2,1}$, $y_{3,1}$), ($y_{1,4}$, $y_{2,4}$, $y_{3,4}$) and ($y_{1,7}$, $y_{2,7}$, $y_{3,7}$).

As pointed out in paragraph [0143] of D1, the ***"1st element (message) of each message vector (set of messages) is delivered to the 1st node processor; the 2nd message is delivered to the 2nd processor; and the 3rd message is delivered to the 3rd processor, respectively"***.

3.3 Hence, the method for supporting decoding of LDPC codes known from D1 comprises the following steps expressed in the wording of claim 1:

(a) constructing a mapped matrix (see Figure 14 of D1) corresponding to a parity check matrix associated with the LDPC codes;

(b) wherein the mapped matrix satisfies a plurality of parallel decodable conditions for permitting a lumped memory structure that is accessible by a plurality of processors 1508 operating in parallel (see Figure 15); and

(c) storing the mapped matrix in memory 1506 for decoding the LDPC codes by the processors 1508.

3.4 As to the parallel decodable conditions according to claim 1, each edge value of a vector ($y_{1,i}$, $y_{2,i}$, $y_{3,i}$) (*i.e.* each column of the edge value arrangement shown in Figure 14) is associated with a different bit node and check node. In fact, each vector contains the edge values corresponding to nonzero elements of a cyclic permutation of a 3 x 3 identity matrix and thus relating to different bit nodes or check nodes (see feature (d_1)).

Furthermore, the first three bit nodes represent all the bit nodes connected to edges in the first vector. The same bit nodes represent also all the bit nodes connected to edges in the second or third vector. As these vectors do not contain any edge values associated with other bit nodes, they fulfil the second condition set out in claim 1 for the rows of a mapped matrix (see feature (d_2)). The same considerations apply to the other bit node groups or to the check node groups.

Hence, all the vectors defined in D1 satisfy also conditions (d_2) and (d_3).

In summary, D1 teaches to define as vectors groups of edge values which correspond to the nonzero elements of a parity check matrix and are required for processing a certain group of bit nodes or check nodes, whereby such vectors satisfy the parallel decodable conditions defined in claim 1 for the rows of the mapped matrix according to the present invention.

3.5 D1 does not show how the edge values are mapped to a memory and, in particular, does not teach that the edge values defining a vector should be stored as rows of a matrix. On the contrary, in Figure 14 they are arranged as columns of a matrix.

However, it is specified in paragraph [0143] that *"the message edge memory 1506 vectors (all 0s at this point) are read out in order and delivered to the vector node processor 1508 for variable node processing. The vector node processor 1508 then outputs the received values alone along each edge from a variable node, we will use y to denote these first messages to indicate this. Thus, the outgoing vectors would be ($y_{1,i}$, $y_{2,i}$, $y_{3,i}$) for i=1, . . . ,12 in increasing order"* (underlining added).

As it is generally known to read out a memory array row by row, it would be obvious for a person skilled in the art wishing to implement the edge value memory of the decoder disclosed in D1 to consider the possibility of storing the edge values identified by a vector in a corresponding row of a memory array.

4.1 The appellant has essentially argued that D1 did not teach to construct a mapped matrix corresponding to a parity check matrix of an LDPC code by defining an arrangement of edge values which was not modified in the course of bit node and check node processing. In fact, in D1 the edge value vectors were subjected to a rotation when the decoder switched from bit node processing to check node processing and vice versa.

4.2 It is true that in D1 the arrangement of the edge values within the vectors is changed during processing. However, even after rotation, the vectors shown in D1 fulfil the conditions set out in claim 1 for the rows of the mapped matrix. Furthermore, it should be noted that the wording of claim 1 does not exclude that the mapping of the edge values might be modified in the course of processing, as long as conditions (d_1), (d_2) and (d_3) remain fulfilled. In fact, it appears from the example given in paragraph [59] of the application, that for check node processing some reordering of the edge values in a row should take place before they are sent to their respective processors because the edge values associated with a particular check node, and thus to be processed by a particular processor, do not occupy the same column in the different rows, as is the case for the edge values of the bit nodes. Thus, for check node processing the mapped matrix of the present invention does not establish a direct correspondence between a column and a check node processor as in the case of bit node processing.

5.1 In summary the Board considers that it was obvious to a person skilled in the art wishing to implement the parallel decoder disclosed in D1 to arrange the values of each vector in a corresponding row of a mapped matrix. In doing so, the skilled person would have arrived at the claimed subject-matter.

5.2 Hence, the subject-matter of claim 1 does not involve an inventive step within the meaning of Article 56 EPC.

6. In the result, the Board finds that the appellant's auxiliary request does not satisfy the requirements of the EPC and that a patent cannot be granted on the basis thereof.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
C. Moser	M. Ruggiu