Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2333109 C 2007/06/19

(11)(21) **2 333 109**

(12) **BREVET CANADIEN**
CANADIAN PATENT

(13) **C**

(86) Date de dépôt PCT/PCT Filing Date: 2000/04/14
(87) Date publication PCT/PCT Publication Date: 2000/10/26
(45) Date de délivrance/Issue Date: 2007/06/19
(85) Entrée phase nationale/National Entry: 2000/11/23
(86) N° demande PCT/PCT Application No.: FI 2000/000322
(87) N° publication PCT/PCT Publication No.: 2000/064057
(30) Priorité/Priority: 1999/04/16 (FI990863)

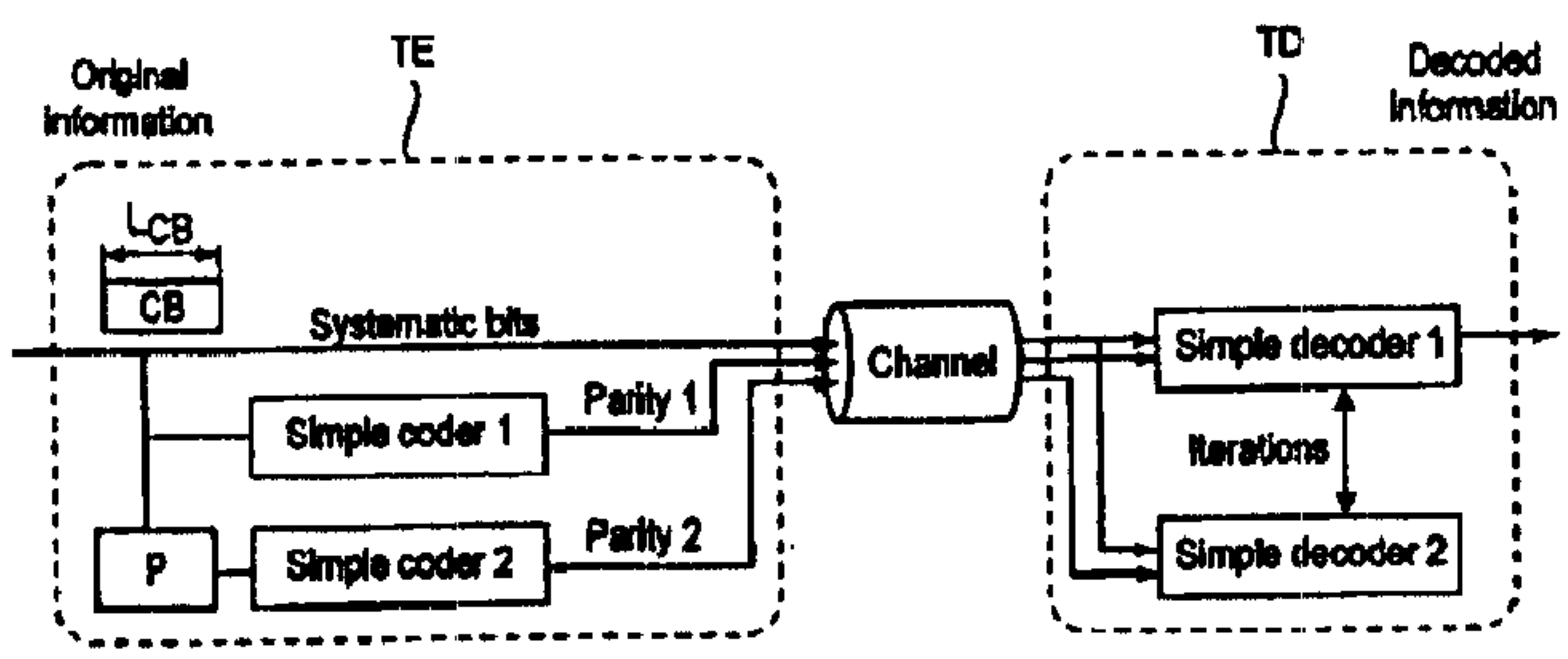
(51) Cl.Int./Int.Cl. *H03M 13/05* (2006.01), *H03M 13/29* (2006.01), *H03M 13/45* (2006.01), *H04L 1/00* (2006.01)

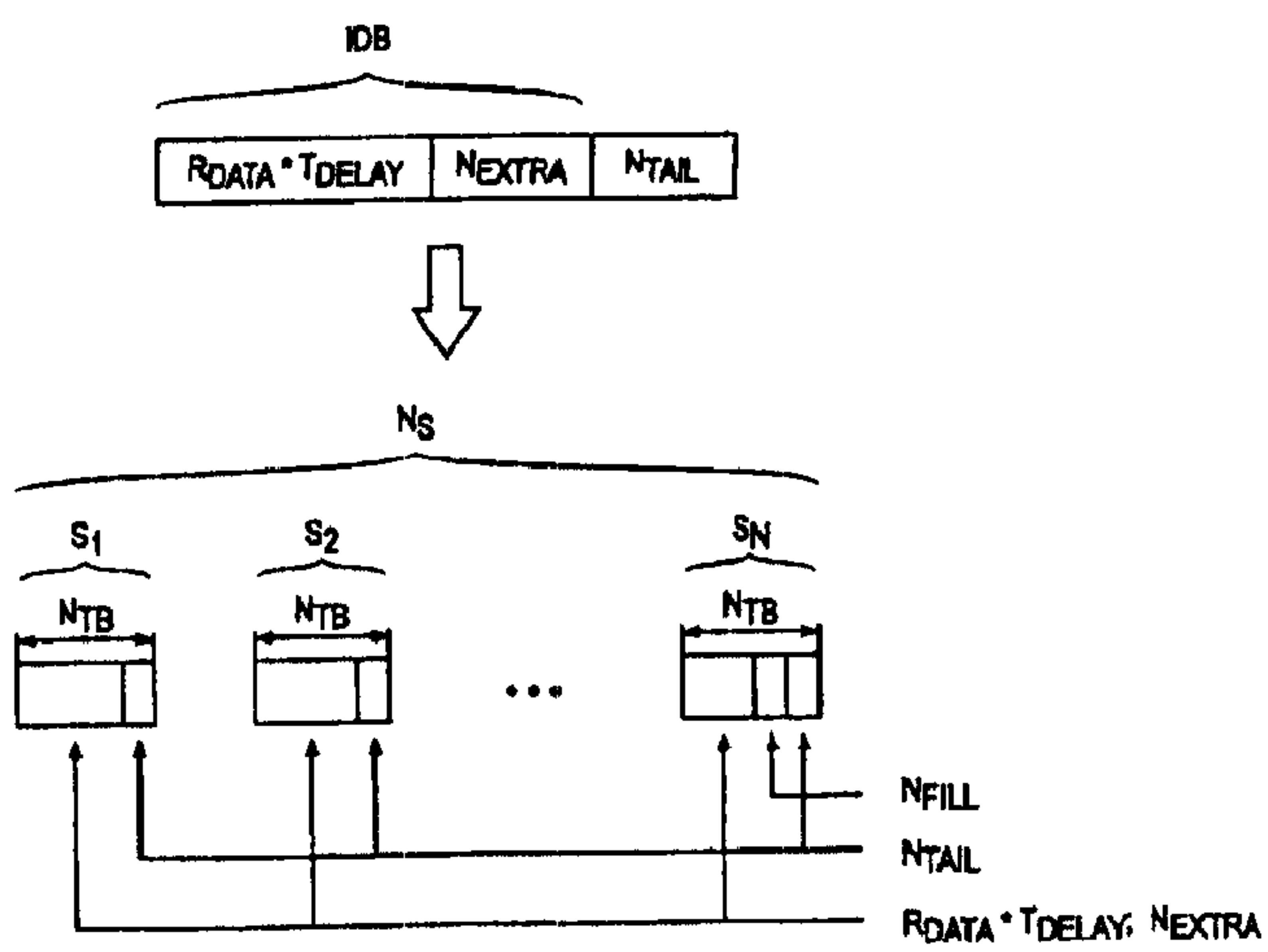
(72) Inventeurs/Inventors:
NIEMINEN, ESKO, FI;
PIRTTIAHO, LAURI, FI

(73) Propriétaire/Owner:
NOKIA NETWORKS OY, FI

(74) Agent: OGILVY RENAULT LLP/S.E.N.C.R.L.,S.R.L.

(54) Titre : MECANISME DE SEGMENTATION POUR CODEUR DE BLOCS
(54) Title: SEGMENTATION MECHANISM FOR A BLOCK ENCODER





(57) Abrégé/Abstract:
A method for encoding an input data block (IDB) with a block encoder (TE). The block encoder is capable of processing consecutive coding blocks (CB) whose size has an upper limit (L_{CB}) which is smaller than the size of the input data block (IDB).

CA 2333109 C 2007/06/19

(11)(21) 2 333 109

(13) C

(57) Abrégé(suite)/Abstract(continued):
The method comprises: (1) determining the length of the input data block (IDB) before encoding any of its data with the block encoder (TE); (2) dividing the input data block (IDB) to a plurality of segments (S_1 ... S_N) wherein all segments are of substantially equal size and no segment is larger than the upper limit (L_{CB}); and (3) processing each segment (S_1 ... S_N) with the block encoder (TE). If the last segment (S_N) is shorter than the remaining segments (S_1, S_2...), fill bits (N_{FILL}) can be added to the last segment (S_N) such that its length equals that of the remaining segments (S_1, S_2...).

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [7] : H03M 13/05	A1	(11) International Publication Number: WO 00/64057 (43) International Publication Date: 26 October 2000 (26.10.00)

(21) International Application Number: PCT/FI00/00322

(22) International Filing Date: 14 April 2000 (14.04.00)

(30) Priority Data:
990863 16 April 1999 (16.04.99) FI

(71) Applicant *(for all designated States except US)*: NOKIA NETWORKS OY [FI/FI]; Keilalahdentie 4, FIN–02150 Espoo (FI).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: NIEMINEN, Esko [FI/FI]; Maahisentie 3 G 10, FIN–90550 Oulu (FI). PIRTTIAHO, Lauri [FI/FI]; Allitie 2 B 29, FIN–90150 Oulu (FI).

(74) Agent: KOLSTER OY AB; Iso Roobertinkatu 23, P.O. Box 148, FIN–00121 Helsinki (FI).

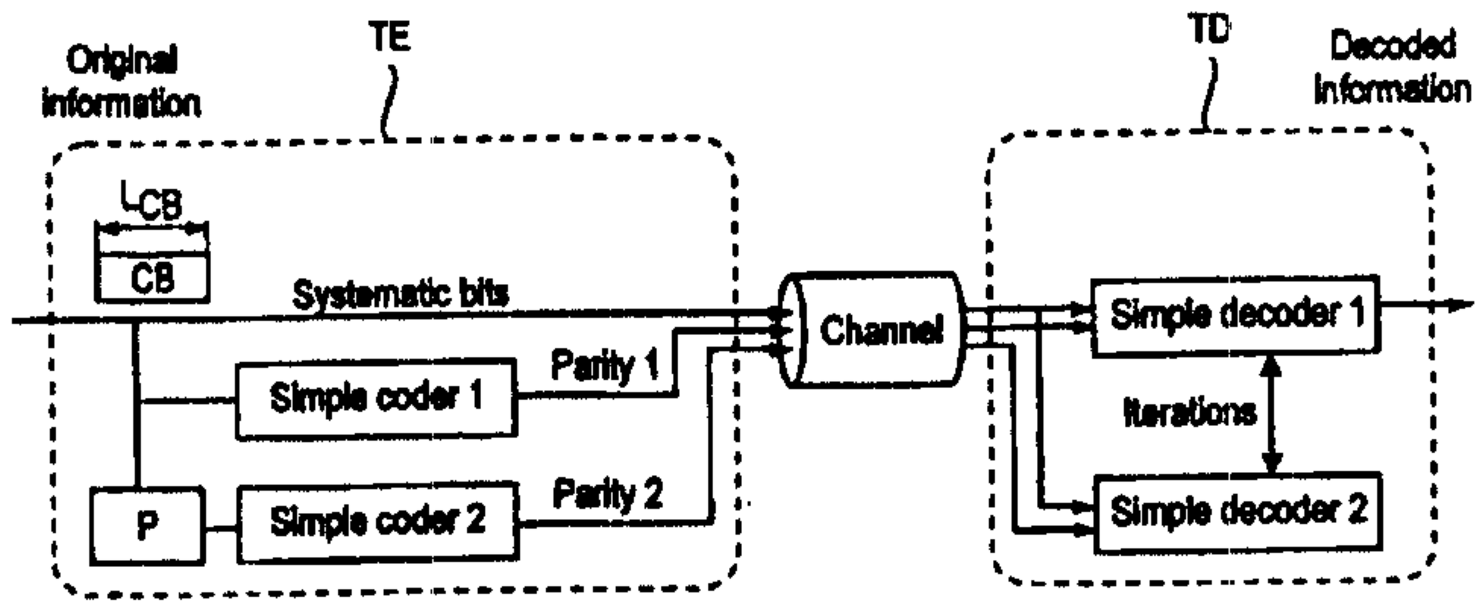
(81) Designated States: AE, AG, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), DM, DZ, EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (Utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

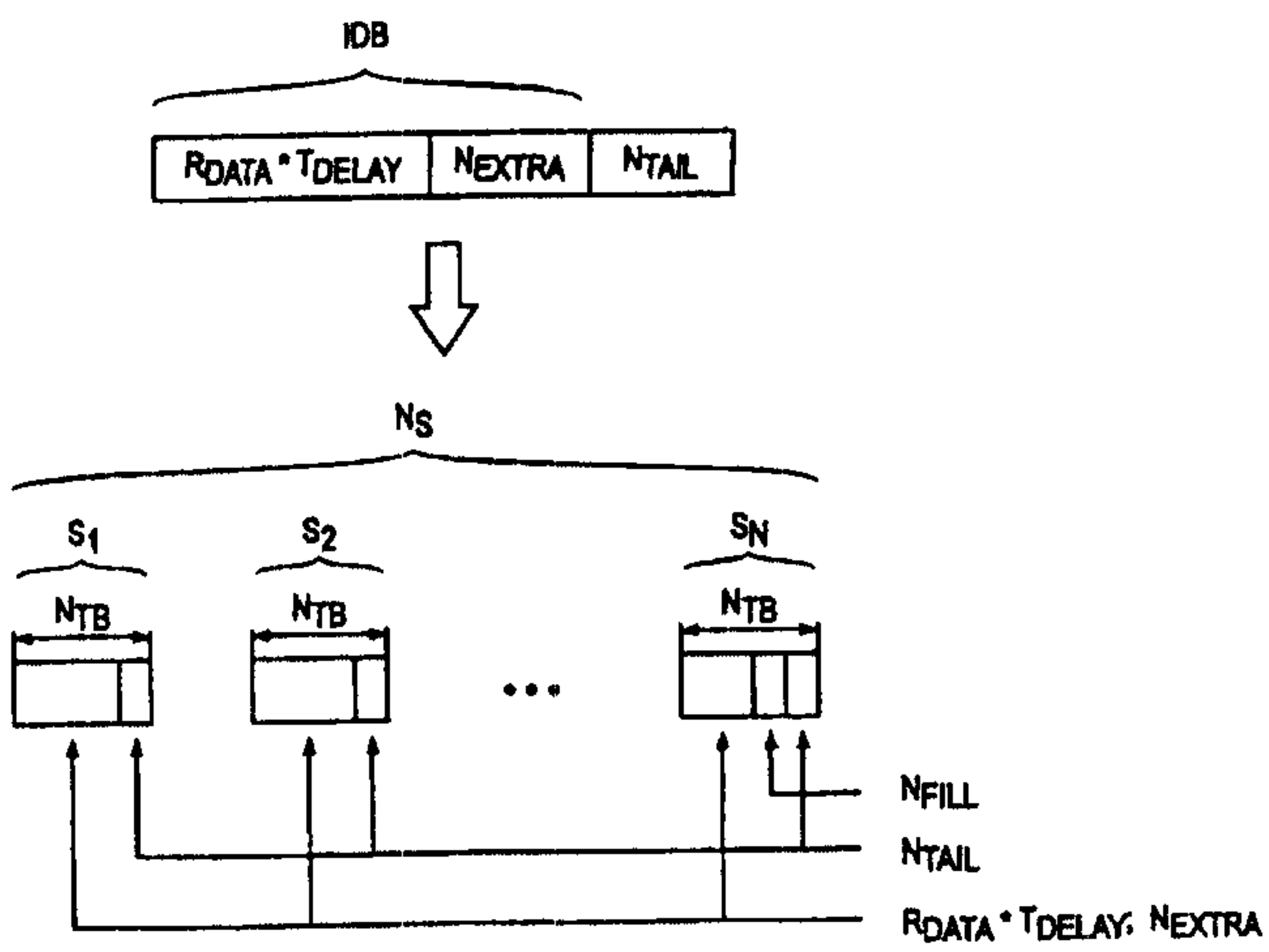
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SEGMENTATION MECHANISM FOR A BLOCK ENCODER

(57) Abstract

A method for encoding an input data block (IDB) with a block encoder (TE). The block encoder is capable of processing consecutive coding blocks (CB) whose size has an upper limit (L_{CB}) which is smaller than the size of the input data block (IDB). The method comprises: (1) determining the length of the input data block (IDB) before encoding any of its data with the block encoder (TE); (2) dividing the input data block (IDB) to a plurality of segments (S_1 ... S_N) wherein all segments are of substantially equal size and no segment is larger than the upper limit (L_{CB}); and (3) processing each segment (S_1 ... S_N) with the block encoder (TE). If the last segment (S_N) is shorter than the remaining segments (S_1, S_2...), fill bits (N_{FILL}) can be added to the last segment (S_N) such that its length equals that of the remaining segments (S_1, S_2...).





SEGMENTATION MECHANISM FOR A BLOCK ENCODER

Background of the invention

The invention relates to methods and equipment for block encoders. As is well known, block encoders are frequently used for error correction.

An example of a block encoder is a so-called turbo (en) coder, which is disclosed in C. Berrou, A. Glavieux, P. Thitimajshima: Near Shannon limit error-correcting coding and decoding: Turbo-codes, IEEE International Conference on Communications, ICC 1993, Geneva, Switzerland 23-26 May, 1993, Vol. 2, pp. 1064-1070.

Unlike streaming encoders, block encoders process one or more data blocks at a time. An input data block whose size exceeds the block size of the block encoder must be divided into smaller segments such that no segment is larger than the block size of the block encoder. This is why block encoders are particularly suitable for applications with a fixed input block size. A problem with block encoders is that they do not easily lend themselves to applications having a variable (dynamic) input block size. In other words, what to do with the last few segments of the input data block, remains an open question.

Disclosure of the invention

An object of the invention is to provide a mechanism for using block encoders with applications having a variable (dynamic) input block size. The mechanism should be generic in order to be applicable to a wide variety of block encoders.

This object is achieved with a method and equipment which are characterized by what is disclosed in the attached independent claims. Preferred embodiments of the invention are disclosed in the attached dependent claims.

A straightforward solution would be to fill the segment of the input data block to the block size of the block encoder. Assuming a block size of 8 kilobits (kb), a 14-kb input data block would be divided into a first segment of 8 kb and second (last) segment with a net size of 6 kb and 2 kb of fill (padding) bits. A benefit of this straightforward solution is that the block encoder does not have to adapt to varying input block sizes.

The invention is based on the idea that for an input data block whose size exceeds the block size of the block encoder:

1) before coding is started, the size of the input data block is determined; and
2) the input data block is divided into segments of approximately equal size such that no segment is larger than the block size of the block encoder.

According to a preferred embodiment of the invention, the input data block is divided into the least possible number of segments. In other words, the segments are as large as possible.

According to an alternative embodiment, the input data block is divided into 2^n segments where n is a positive integer.

According to yet another preferred embodiment of the invention, if dividing the input data block produces a last segment which is shorter than the remaining segments, the input data block or the last segment is padded with a few fill bits until the length of the last segment equals that of the remaining segments. However, in contrast to the straightforward solution, the last segment is not padded to the full block size of the block encoder (unless the remaining segments happen to be of that size too).

Brief description of the drawings

The invention will be described in more detail by means of preferred embodiments with reference to the appended drawing wherein:

Figure 1 is a block chart of a turbo encoder; and

Figure 2 illustrates dividing an input data block to a number of segments for the turbo encoder.

Figure 1 is a block chart of a turbo encoder TE which is connected to a corresponding turbo decoder TD via a (transmission) channel. A typical turbo encoder conveys the original information directly to the channel. These bits are called systematic bits. Additionally, the turbo encoder adds redundancy (parity) with simple encoders 1 and 2, the latter of which is preceded by an interleaver P, which permutes the bits of the original information. However, details of the block encoder are not relevant for understanding the invention, and reference is made to relevant literature.

Detailed description of the invention

Various embodiments of the invention will be described in connection with a turbo encoder, an example of which is disclosed in reference 1.

However, details of the turbo encoder, or any other encoder, are not relevant for understanding the invention.

Implementing a turbo encoder (and the corresponding decoder) can be facilitated by limiting the length of the coding block. A reasonable value for

the length of the coding block is 8192 bits including the user data, a possible error detection field (CRC) and the termination. The following naming conventions will be used:
N_{TAIL} = the number of termination bits
T_{DELAY} (seconds) = the length of the user data block
R_{DATA} (bits per second) = the user data rate of the service
N_{EXTRA} = the number of other bits added to the original user data (CRC etc.)
L_{CB} = max length of the coding block.

The following condition has to be satisfied:

$$R_{DATA} * T_{DELAY} + N_{EXTRA} + N_{TAIL} \leq L_{CB} \qquad [1]$$

If this condition is not satisfied the data to be encoded must be segmented so that each separate segment satisfies the condition. The number of segments N_S has to satisfy the condition:

$$\text{round\_up}((R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S) + N_{TAIL} \leq L_{CB} \qquad [2]$$

It is preferable to choose the smallest N_S satisfying the inequality [2]. N_S can be calculated from:

$$N_S = \text{round\_up}((R_{DATA} * T_{DELAY} + T_{EXTRA}) / (L_{CB} - N_{TAIL})) \qquad [3]$$

It may happen that all the encoding blocks do not end up being of the same length, i.e. that $(R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S$ is not an integer. In such a case, there are at least two possible solutions, which will be called algorithms A and B. In algorithm A, the last segment is allowed to have a different length than the other segments. In algorithm B, a number N_{FILL} of fill bits (e.g. zeroes) are added to the input data so that $(R_{DATA} * T_{DELAY} + N_{EXTRA} + N_{FILL}) / N_S$ is the smallest possible integer. (Alternatively, the fill bits can be appended to the last segment after segmentation.)

Algorithm A

Algorithm A allows the last segment to be shorter than the other segments. Algorithm A uses the following Inputs:
R_{DATA} = the user data rate (bits per second)
T_{DELAY} = encoding user data block length (seconds)
N_{EXTRA} = extra data to be appended to the user data before encoding (bits)

N_{TAIL} = number of tail bits to be appended to the encoding blocks

Algorithm A produces the following outputs:
N_S = number of segments
N_{TB} = number of bits in the turbo encoder input blocks except the last one
N_{LAST_TB} = number of bits in the last turbo encoder input block

In algorithm A the following computations will be performed:
Let N_S = round_up((R_{DATA} * T_{DELAY} + N_{EXTRA}) / (L_{CB} – N_{TAIL}))
Let N_{TB} = round_up((R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S) + N_{TAIL};
Let N_{REM} = remainder of (R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S;
If N_{REM} is not equal to zero then N_{LAST_TB} = N_{TB} – N_S + N_{REM} else N_{LAST_TB} = N_{TB}.
End.

If algorithm A is used, an adaptive turbo interleaver is needed since the last input segment to the turbo encoder may be shorter than the others. The number of systematic bits in the output of the encoder is

$$R_{DATA} * T_{DELAY} + N_{EXTRA} + N_S * N_{TAIL} \qquad [4]$$

Thus there are no additional bits other than the ones due to the termination of each segment.

Algorithm B

As shown in Figure 2, in algorithm B all input segments to the turbo encoder will be of equal size. The inputs to algorithm B are:
R_{DATA} = the user data rate (bits per second)
T_{DELAY} = encoding user data block length (seconds)
N_{EXTRA} = extra data to be appended to the user data before encoding (bits)
N_{TAIL} = number of tail bits to be appended to the encoding blocks

The outputs from algorithm B are:
N_S = number of segments
N_{TB} = number of bits in the turbo encoder input blocks
N_{FILL} = number of fill bits (e.g. zero) in the last turbo encoder input block

In algorithm B, the following computations will be performed:
Let N_S = round_up((R_{DATA} * T_{DELAY} + N_{EXTRA}) / (L_{CB} – N_{TAIL}))
Let N_{TB} = round_up ((R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S) + N_{TAIL}
Let N_{REM} = remainder of (R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S

If $N_{REMX} \neq 0$ then insert $N_{FILL} = (N_s - N_{REM})$ zero bits to the end of the input data else $N_{FILL} = 0$.

All input segments to the turbo encoder are of equal size and therefore the same turbo interleaver can be used for all segments. In this case the number of systematic bits over an entire channel interleaving block at the output of the turbo encoder is:

$$N_S * (\text{round\_up}((R_{DATA} * T_{DELAY} + N_{EXTRA}) / N_S) + N_{TAIL}) \qquad [5]$$

Thus there may be some additional bits other than the termination bits.

Modification to algorithms A and B

In the above algorithms A and B, the length of input segment to the turbo encoder is maximised by choosing the smallest possible number of segments N_S. In some cases it may be preferable to use a number of segments N_S which is a power of 2, but this will shorten the input segments to the turbo encoder. In this case, the first step of the above algorithms A and B would be replaced by the following three steps:

Let $n_s = \text{round\_up}((R_{DATA} * T_{DELAY} + N_{EXTRA}) / (L_{CB} - N_{TAIL}))$;

Let $m = \text{round\_up}(\log_2 n_s)$;

Let $N_s = 2^m$.

CLAIMS,

1. 1. A method for encoding an input data block (IDB) with a block encoder (TE), said block encoder being capable of processing consecutive coding blocks (CB) whose size has an upper limit (L_{CB}) which is smaller than the size of the input data block (IDB);

characterized in that the method comprises the steps of:

determining the length of the input data block (IDB) before encoding any of its data with said block encoder (TE);

dividing the input data block (IDB) to a plurality of segments(S_1 ... S_N) wherein all segments are of substantially equal size and no segment is larger than said upper limit (L_{CB}); and

processing each segment (S_1 ... S_N) with said block encoder (TE).

2. A method according to claim 1, characterized in that if the last segment (S_N) is shorter than the remaining segments (S_1, S_2...), fill bits (N_{FILL}) are added to the last segment (S_N) such that its length equals that of the remaining segments (S_1, S_2...).

3. A method according to claim 1, characterized in that if the length of the input data block (IDB) is not an exact multiple of said upper limit (L_{CB}), fill bits (N_{FILL}) are added to the input data block (IDB) such that its length is an exact multiple of said upper limit (L_{CB}).

4. A method according to any one of claims 1 to 3, characterized in that the number of segments is 2^n where n is a positive integer.

5. A segmentation device for segmenting an input data block (IDB) for processing with a block encoder (TE), said block encoder being capable of processing consecutive coding blocks (CB) whose size has an upper limit (L_{CB}) which is smaller than the size of the input data block (IDB);

characterized in that the segmentation device is arranged to:

determine the length of the input data block (IDB) before applying any of its data to said block encoder (TE);

divide the input data block (IDB) to a plurality of segments(S_1 ... S_N) wherein all segments are of substantially equal size and no segment is larger than said upper limit (L_{CB}); and to

apply each segment (S_1... S_N) to said block encoder (TE).

Fig. 1

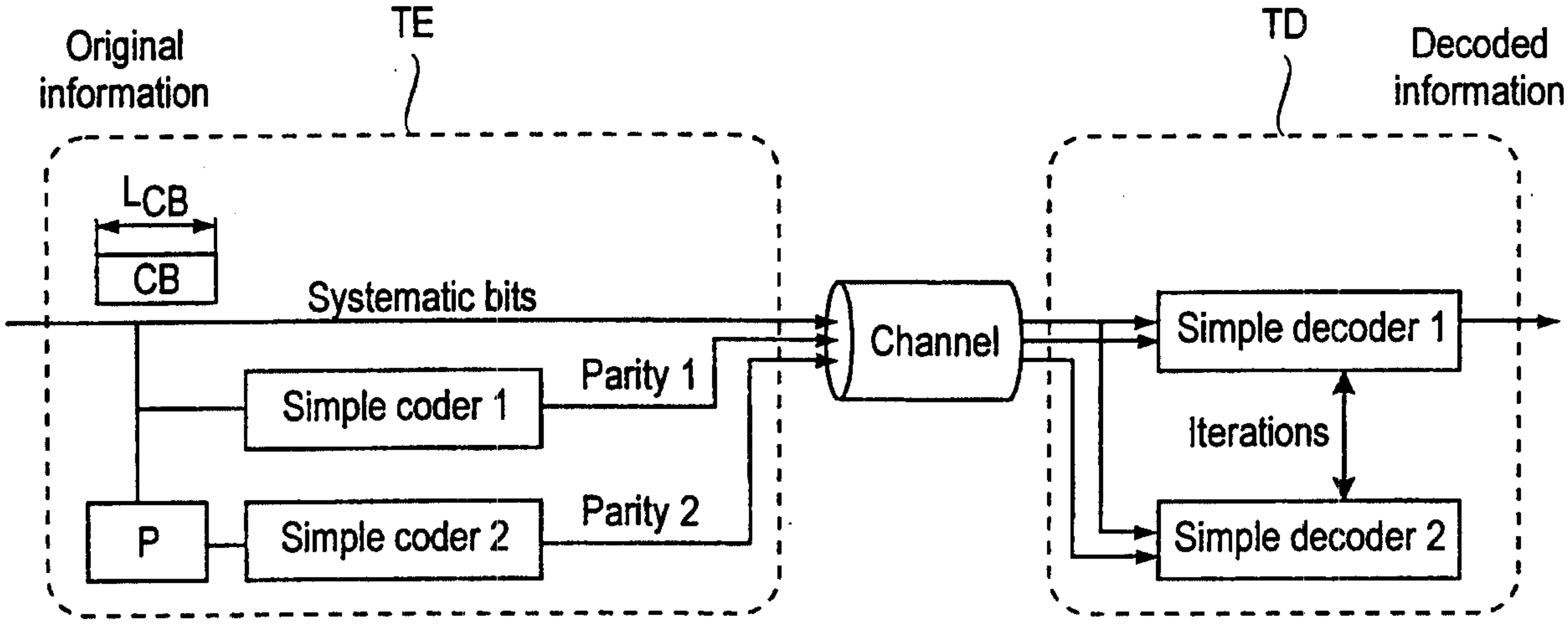
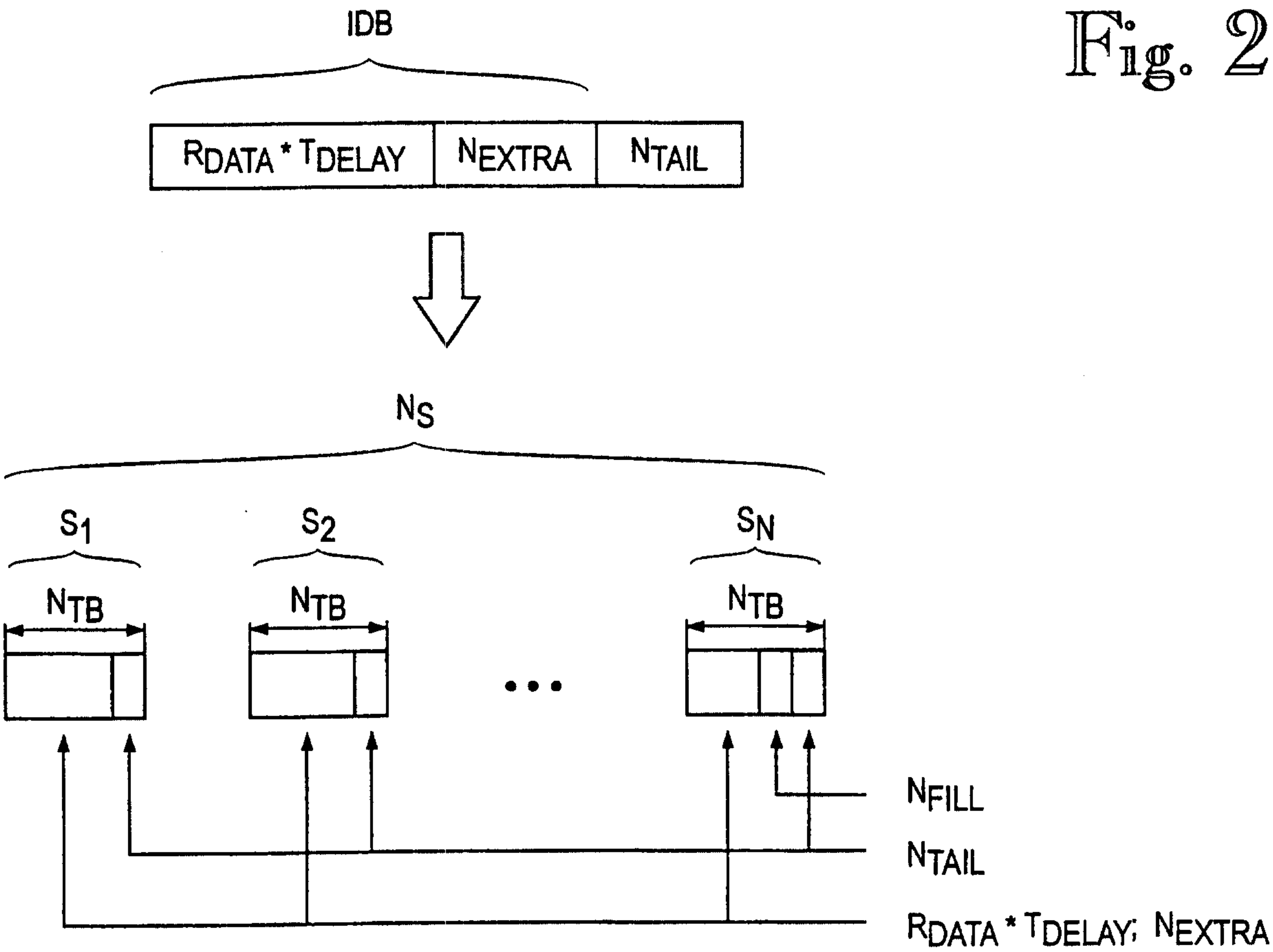
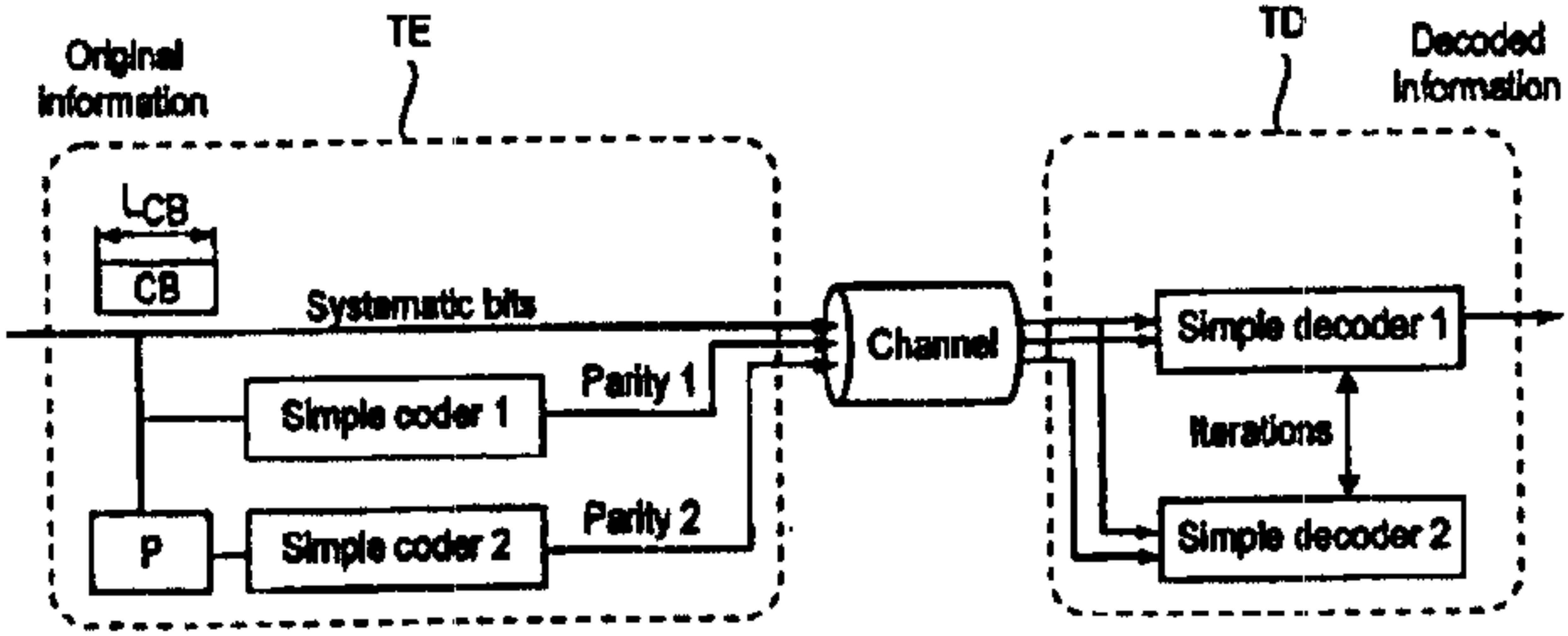
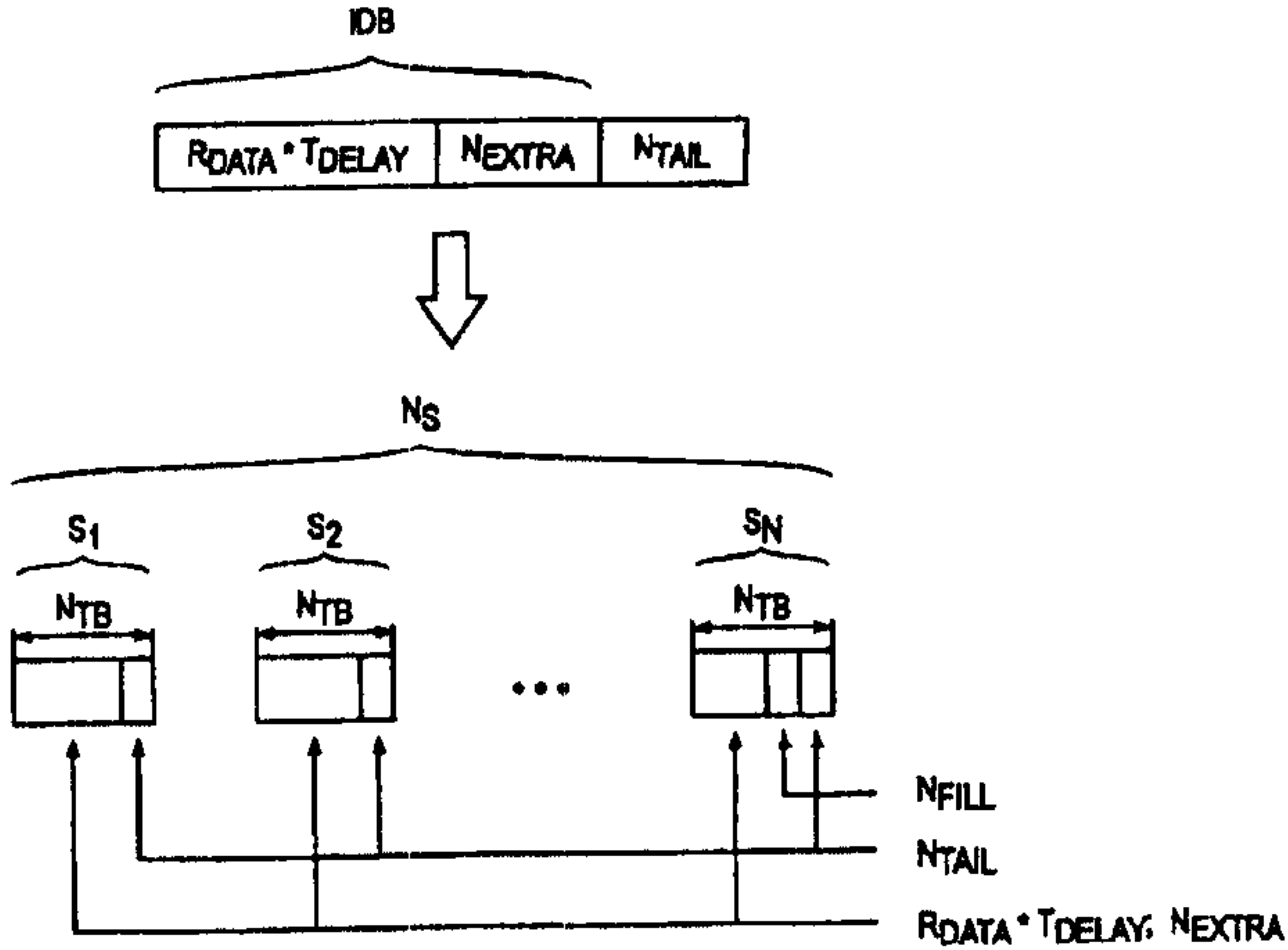

Original information
TE
TD
Decoded information
L_{CB}
CB
Systematic bits
Channel
Simple coder 1
Parity 1
Simple decoder 1
Iterations
P
Simple coder 2
Parity 2
Simple decoder 2

Fig. 2


IDB
$R_{DATA} * T_{DELAY}$
N_{EXTRA}
N_{TAIL}
N_S
S_1
S_2
S_N
N_{TB}
N_{TB}
N_{TB}
...
N_{FILL}
N_{TAIL}
$R_{DATA} * T_{DELAY}$; N_{EXTRA}


TE
TD
Original information
Decoded information
L_{CB}
CB
Systematic bits
Channel
Simple decoder 1
Parity 1
Simple coder 1
Iterations
P
Simple coder 2
Parity 2
Simple decoder 2


IDB
$R_{DATA} * T_{DELAY}$
N_{EXTRA}
N_{TAIL}
N_S
S_1
S_2
S_N
N_{TB}
N_{TB}
N_{TB}
N_{FILL}
N_{TAIL}
$R_{DATA} * T_{DELAY}, N_{EXTRA}$