

(19) World Intellectual Property  
Organization  
International Bureau



(43) International Publication Date  
8 January 2004 (08.01.2004)

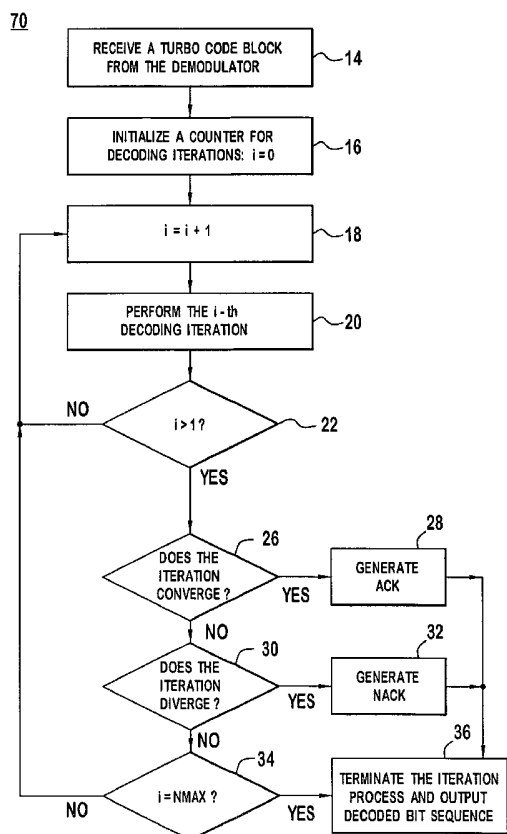
PCT

(10) International Publication Number  
WO 2004/004133 A1

- (51) International Patent Classification<sup>7</sup>: **H03M 13/00**
- (21) International Application Number:  
PCT/US2003/019643
- (22) International Filing Date: 20 June 2003 (20.06.2003)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/392,200 28 June 2002 (28.06.2002) US  
10/334,490 30 December 2002 (30.12.2002) US
- (71) Applicant: **INTERDIGITAL TECHNOLOGY CORPORATION** [US/US]; 300 Delaware Avenue, Suite 527, Wilmington, DE 19801 (US).
- (72) Inventors: **SHIN, Sung-Hyuk**; 1531 8th Street, Fort Lee, NJ 07024 (US). **BOLOURCHI, Nader**; 20 Bonnie Way, Larchmont, NY 10538 (US).
- (74) Agents: **VOLPE, Anthomy, S.** et al.; Volpe and Koenig P.C., United Plaza, Wuite 1600, 30 South 17th Street, Philadelphia, PA 19103 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A FAST H-ARQ ACKNOWLEDGEMENT GENERATION METHOD USING A STOPPING RULE FOR TURBO DECODING



(57) Abstract: A stopping rule for Turbo Decoding (14) is applied for both good and bad code blocks in a received data stream. The iteration (26) either converges or diverges. Rule testing may be used for H-ARQ acknowledgement generation: if the iteration converges, an ACK (28) is generated and if the iteration diverges, a NACK (32) is generated. Optionally, the maximum number of decoding iterations (34) may dynamically selected based on MCS levels.

WO 2004/004133 A1



**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.*

[0001] A FAST H-ARQ ACKNOWLEDGEMENT GENERATION  
METHOD USING A STOPPING RULE FOR TURBO DECODING

[0002] FIELD OF INVENTION

[0003] The present invention is related to data communication systems. More particularly, the present invention is directed to an improved Turbo decoder in a data communication system.

[0004] BACKGROUND

[0005] Turbo codes are used for data communication systems [such as a High Speed Downlink Shared Channel (HS-DSCH) in High Speed Downlink Packet Access (HSDPA) in wireless communication systems] as a forward error connection (FEC) scheme. Decoding of Turbo codes is iterative in nature. That is, each Turbo code block is decoded several times. In general, there is a tradeoff between the Turbo code performance, which improves with the number of decoding iterations, and the decoding delay and computational complexity. Conventionally, the number of decoding iterations is fixed (for example, at 4 or 8 iterations). However, some Turbo code blocks may need only a few decoding iterations to successfully decode the code blocks, (i.e. to converge), before reaching the last decoding iteration and further iterations are not necessary. In such a case, if the Turbo decoder stops the redundant decoding iterations for the good blocks, it reduces the decoding delay and power consumption without degrading performance.

[0006] To prevent an endless loop when the stopping rule is never satisfied, the decoder stops after a maximum number of iterations. Several stopping rules for Turbo decoding have been addressed in the prior art. However, prior art stopping rules are focused on the case where decoding iterations converge (e.g., for good Turbo coded blocks).

[0007] SUMMARY

[0008] The present invention not only implements a stopping rule for good code blocks, but also includes a stopping rule for bad code blocks which fail to be

correctly decoded even at the last decoding iteration. This benefits data communication systems such as HSDPA which employ an H-ARQ (hybrid-automatic repeat request) protocol, since the H-ARQ protocol requests bad blocks to be retransmitted. It is particularly applicable with HS-DSCHs with H-ARQ that may require raw block error rates (BLERs) before retransmission on the order of  $10^{-1}$ , which leads to frequent occurrences of bad Turbo coded blocks for HS-DSCH. It should be noted that although the present invention will focus on HSDPA as an example, other data communication system using Turbo coding and an H-ARQ technique may also be used in accordance with the teachings of the present invention.

[0009] The H-ARQ protocol used for HSDPA sends the transmitter an acknowledgement (ACK/NACK) of each H-ARQ process where generation of the acknowledgement is typically based on the cyclic redundancy check (CRC) check result of the individual H-ARQ process. There is some delay in deriving the CRC result, which may be on the order of 10msec. The CRC processing delay may cause H-ARQ performance degradation. As an alternative to the H-ARQ acknowledgement generation, the result of the stopping rule testing may be used to determine whether a given H-ARQ process is in error (NACK generation) or error-free (ACK generation).

[0010] In addition, HSDPA employs adaptive modulation and coding (AMC) as a link adaptation technique. The modulation and coding format can be changed on a radio frame basis in accordance with variations in the channel conditions, subject to system restrictions. In order to more efficiently implement the Turbo decoder with a stopping rule, the maximum number of Turbo decoding iterations may be dynamically selected depending on a code rate and modulation type for the HS-DSCH.

[0011] The present invention provides the advantage of a reduction in the decoding delay and computational complexity at the user equipment (UE) receiver. In addition, reduction in the decoding delay leads to earlier availability of H-ARQ acknowledgements at the Node B, which improves HSDPA performance.

[0012] BRIEF DESCRIPTION OF THE DRAWING(S)

[0013] The present invention will be described with reference to the drawing figures wherein like numerals represent like elements throughout and wherein:

[0014] Figures 1 and 2 are flow diagrams useful in describing alternative techniques of the present invention.

[0015] Figure 3 is a modified block diagram showing apparatus utilized to perform the turbo decoding technique of the present invention.

[0016] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0017] The stopping rule known as Sign Change Ratio (SCR) is implemented for Turbo decoding in accordance with the present invention. This rule depends upon the sign changes of the extrinsic information provided by the component decoders in the Turbo decoder between the  $(k-1)^{\text{th}}$  and  $k^{\text{th}}$  iterations for both good and bad Turbo code blocks. The conventional SCR stopping rule attempts to determine, by checking the sign changes, when the iteration converges and then terminates the iteration process. This SCR stopping rule is applied only to good received code blocks. However, in accordance with the present invention, the SCR stopping rule is applied to bad code blocks as well. This especially benefits HSDPA systems employing a H-ARQ protocol, since the H-ARQ protocol requests bad H-ARQ processes consisting of Turbo code block(s) to be retransmitted. It should be noted that although the present invention will focus on the SCR based stopping rule as an example, other stopping criterion may also be used in accordance with the teachings of the present invention. By way of example, other known stopping criteria include: (a) CRC wherein, after each decoding iteration, CRC bits are checked for errors and the iteration is stopped if there is no CRC error and (b) Cross Entropy wherein after each iteration, the cross entropy between log-likelihood ratios of the component decoders is calculated and the iteration is terminated if the estimated cross entropy is less than a given threshold.

[0018] To see the behavior of iterative decoding in the Turbo decoder, Turbo code simulations were performed with a fixed number of iterations  $k$  where  $k$  is (set to 8). Table 1 shows typical samples of the simulation results in terms of the number of sign changes at each iteration for good Turbo code blocks, and Table 2 shows typical samples of the simulation results in terms of the number of sign changes at each iteration for bad Turbo code blocks. As observed in Table 1, with good code blocks the number of sign changes between  $(k-1)$  and  $k$  (for  $k > 1$ ) iterations converges before the last (8<sup>th</sup>) iteration. In this case, if the stopping rule is applied, the average number of iterations would be reduced to approximately 4.

Blocks	# of sign changes between $(k-1)$ and $k$ iterations							Stopped iteration
	K=2	K=3	K=4	K=5	K=6	K=7	K=8	
1	3	0	0	0	0	0	0	K=3
2	8	3	0	0	0	0	0	K=4
3	16	9	0	0	0	0	0	K=4
4	4	8	7	3	0	0	0	K=6
5	11	2	0	0	0	0	0	K=4
6	18	20	11	10	0	0	0	K=6
7	19	5	0	0	0	0	0	K=4
8	16	9	0	0	0	0	0	K=4
9	4	5	3	0	0	0	0	K=5
10	10	0	0	0	0	0	0	K=3

Table 1. Typical samples of TC simulation results in terms of the number of sign changes for successful decoded (good) blocks when 16 QAM,  $\frac{3}{4}$  rate, BLER=10%

[0019] In Table 2, it is shown that with bad code blocks the number of sign changes never converges.

Blocks	# of sign changes between (k-1) and k iterations							Stopped iteration
	K=2	K=3	K=4	K=5	K=6	K=7	K=8	
1	30	39	29	37	46	49	31	K=3
2	24	36	39	39	38	35	28	K=3
3	33	27	24	23	24	14	17	K=8
4	11	11	12	20	21	37	34	K=5
5	9	14	9	8	11	9	16	K=3
6	18	10	7	9	17	14	7	K=5
7	3	34	39	38	39	23	25	K=3
8	16	14	34	36	12	22	35	K=4

Table 2. Typical samples of TC simulation results in terms of the number of sign changes for unsuccessfully decoded (bad) blocks when 16 QAM,  $\frac{3}{4}$  rate, BLER=10%

[0020] In the present invention, it is proposed that the iterative decoding process is terminated if either the iteration converges or the iteration diverges. Otherwise the decoding ceases after a maximum number of iterations.

[0021] Referring to Figure 1, a flowchart of the method 10 in accordance with the present invention for Turbo decoding is shown. The method 10 commences by receiving a Turbo code block from a demodulator (step 14). A counter for decoding iterations is then initialized ( $i=0$ ) (step 16) and then the counter incremented ( $i=i+1$ ) (step 18). The  $i$ th decoding iteration is performed (step 20) and it is determined whether or not this is the first iteration (step 22). If it is the first iteration, the procedure 10 reverts to step 18. If not, the method 10 makes a determination of whether or not the iteration converges or diverges.

[0022] If the SCR is considered as the stopping criterion, then the iteration convergence and divergence can be defined as follows. If the number of the sign changes between the  $(k-1)$ th iteration and  $k$ th iteration (for  $k > 1$ ) becomes zero, the iteration is determined to be converging. If the number of the sign changes between the  $(k-1)$ th iteration and  $k$ th iteration (for  $k > 2$ ) is greater than that between the  $(k-2)$ th iteration and  $(k-1)$ th iteration, the iteration is determined to

be diverging. Accordingly, at step 26, it is determined whether the iteration converges. If so, the iteration process is terminated after generation of an ACK (step 28) and the decoded sequence is output (step 36). If not, it is determined whether the iteration diverges (step 30). If the iteration diverges, the iteration process is terminated after generation of NACK (step 32) and the decoded sequence is output (step 36). If the iteration does not diverge, it is determined whether the maximum number of iterations ( $i=N_{max}$ ) has been reached (step 34). If so, the iteration process is terminated and the decoded bit sequence is output (step 36). If not, the process returns to step 18 whereby the counter is incremented ( $i=i+1$ ) and steps 20-36 are repeated. It should be noted that the maximum number of iterations  $N_{max}$  may be dynamically selected as a function of the applied code rate and modulation type. For example, the higher the code rate and the higher the order of the modulation type, the less the maximum number of iterations  $N_{max}$ .

[0023] Figure 2 is a flow chart of an alternative method 70 in accordance with the present invention for Turbo decoding. In this embodiment 70, the results of the stopping rule are used for H-ARQ acknowledgement generation. The like steps of the method 70 shown in Figure 2 are numbered the same as the steps of the procedure 10 shown in Figure 1 and therefore will not be further described with reference to Figure 2.

[0024] In accordance with this embodiment of the present invention, after a determination of whether or not the iteration converges, an acknowledgement (ACK) or non-acknowledgement (NACK) for H-ARQ is generated. More specifically, referring to step 26, if it is determined that the iteration converges (step 26), an ACK is generated (step 28) assuming that an H-ARQ process has a single Turbo code block. When there are multiple Turbo code blocks in an H-ARQ process, the ACK for the H-ARQ process will be generated, if all the iterations with all the code blocks converge. The iteration process is then terminated and the decoded bit sequence is output (step 36). If the iteration does not converge as determined at step 26, it is then determined whether or not the iteration diverges (step 30). If so, a NACK is generated (step 32) for the H-ARQ process carrying

the decoded block, the iteration process is terminated and the decoded bit sequence is output (step 36). When there are multiple Turbo code blocks in an H-ARQ process, if any one code block is determined to be diverging (generating NACK), then all the iterations with other relevant code blocks may be terminated as well. If the iteration does not diverge, as determined at step 30, it is determined whether or not the iteration has reached the maximum number of iterations  $N_{max}$  (step 34). If so, the iteration process is terminated and the decoded sequence is output (step 36). If the maximum number of iterations  $N_{max}$  has not been reached, as determined at step 34, the counter is incremented (step 18) and steps 20-36 are repeated. Accordingly, if the iteration process does not converge or diverge, the H-ARQ acknowledgement generation will be based on CRC check results as in the prior art. The use of the Turbo decoding aided H-ARQ acknowledgement generation may reduce H-ARQ processing delay at the receiving station, taking into account the delay in CRC processing (on the order of 10 msec).

[0025] In Figure 3, a block diagram of the Turbo decoder structure 100 is shown, including the stopping rule decision unit. In general, the Turbo decoder 100 consists of (2) two SISO (soft input soft output) modules, SISO1 106 and SISO2 108. Each SISO provides soft-valued log-likelihood ratios (LLR) for the other SISO through the Turbo internal interleaver/de-interleaver 110, 112. After each iteration, a stopping rule decision unit 114 checks whether the decoding iteration converges or diverges, or neither. If the decision turns out to be either "converged" or "diverged", the iteration is stopped and either "Ack" or "Nack" indication depending on convergence or divergence is generated for H-ARQ processing. Otherwise, the decoder continues the iteration.

[0026] More specifically, the Turbo decoder 100 processes soft-valued input data 102 in each Turbo code block in a transmission. The input 102 to the Turbo decoder is passed through a demultiplexer 104 which separates the input into three sequences: systematic bit sequence, parity bit 1 sequence, and parity bit 2 sequence. The systematic bit sequence and parity bit 1 sequence are initially sent to the SISO 1 decoder 106 (soft-input soft-output decoder), along with a

priori information data derived from the SISO 2 decoder 108. The SISO 1 decoder 106 generates log-likelihood ratios (LLRs) (i.e. extrinsic information plus systematic information) of the information bits. The LLRs from the SISO 1 decoder 106 are permuted by a Turbo internal interleaver 110 and passed to the SISO 2 decoder 108. Along with the interleaved LLRs, the parity bit 2 sequence is fed into the SISO 2 decoder 108. The extrinsic information output of the SISO 2 decoder are deinterleaved in accordance with the Turbo internal deinterleaver 112 performing an inverse permutation with respect to the Turbo internal interleaver 110. The permuted extrinsic information is then fed back as the a priori information of the SISO 1 decoder 106 to repeat the process. After each iteration, the stopping rule decision unit 114 determines whether the iteration converged, diverged or neither converged or diverged. If the decision turns out to be either "converged" or "diverged," the iteration is stopped, the decoded bit sequence is output at 116, and a corresponding H-ARQ acknowledgement is provided at 114a for H-ARQ processing. Otherwise the process continues to be iterated.

[0027] The present invention provides the advantage of a reduction in the decoding delay and computation complexity at the receiving station. In addition, a decrease of the decoding delay leads to make H-ARQ acknowledgements available earlier at the transmission, which improves H-ARQ performance.

[0028] Although the present invention has been described in detail, it is to be understood that the invention is not limited thereto, and that various changes can be made therein without departing from the spirit and scope of the invention, which is defined by the attached claims.

\* \* \*

## CLAIMS

What is claimed is:

1. A method employed in conjunction with turbo decoding for reducing a number of iterations, comprising:
  - a) initializing an iteration count upon receipt of a turbo code block;
  - b) increasing the iteration count responsive to performance of step (a);
  - c) performing an iteration;
  - d) increasing the iteration count responsive to performance of step (c);
  - e) determining if a latest iteration converges;
  - f) generating an acknowledgement responsive to step (e); and
  - g) terminating performance of further iterations.
2. The method of claim 1 wherein step (f) includes an acknowledge signal (ACK) that the last iteration converged.
3. The method of claim 1 further includes:
  - (h) outputting a decoded sequence.
4. A method employed in conjunction with turbo decoding for reducing a number of iterations, comprising:
  - a) initializing an iteration count upon receipt of a turbo code block;
  - b) increasing the iteration count responsive to performance of step (a);
  - c) performing an iteration;
  - d) increasing the iteration count responsive to performance of step (c);
  - e) determining if a latest iteration diverges;
  - f) generating an acknowledgement responsive to step (e); and
  - g) terminating performance of further iterations.
5. The method of claim 4 wherein step (f) includes generating an acknowledge signal (NACK) that the last iteration diverged.

6. The method of claim 4 further includes:

(h) outputting a decoded sequence.

7. A method employed in conjunction with turbo decoding iterations, comprising:

a) initializing an iteration count upon receipt of a turbo code block;

b) increasing the iteration count responsive to performance of step (a);

c) performing an iteration;

d) increasing the iteration count responsive to performance of step (c);

e) determining if a latest iteration converges;

f) determining if the latest iteration diverges in the event that there was not convergence at step (e);

g) generating an acknowledgement responsive to step (f); and

h) terminating performance of further iterations.

8. The method of claim 7 wherein step (g) includes generating an acknowledgement (NACK) which indicates that the latest iteration diverged.

9. The method of claim 7 which further includes:

i) outputting a decoded sequence.

10. The method of claim 1 wherein performing an iteration comprises:

h) generating log-likelihood ratios (LLRs) based on a systematic bit sequence and a parity bit 1 sequence of the inputted coded data and a priori data;

i) permuting the LLRs;

j) generating extrinsic information based on the permuted LLRs and a parity 2 sequence of the coded input data;

k) de-interleaving the extrinsic information; and

l) providing the de-interleaved extrinsic information as a priori data employed at step (h).

11. The method of claim 10 wherein step (e) includes:
  - m) making a convergence determination based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).
  
12. The method of claim 10 wherein step (e) includes:
  - m) making a divergence determination based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).
  
13. The method of claim 11 wherein step (f) performs an acknowledgement based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).
  
14. The method of claim 12 wherein step (f) performs an acknowledgement based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).
  
15. A method employed in conjunction with turbo decoding for reducing a number of iterations, comprising:
  - a) generating log-likelihood ratios (LLRs) based on a systematic bit sequence and a parity bit 1 sequence of inputted coded data and a priori data;
  - b) permuting the LLRs;
  - c) generating extrinsic information based on the permuted LLRs and a parity 2 sequence of the coded input data;
  - d) de-interleaving the extrinsic information; and
  - e) providing the de-interleaved extrinsic information as a priori data employed at step (h).
  
16. The method of claim 15 wherein step (e) includes:

m) making a convergence determination based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).

17. The method of claim 15 wherein step (e) making a determination based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).

18. The method of claim 15 wherein step (f) performs an acknowledgement based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated at step (k).

19. Apparatus employed in turbo decoding for reducing a number of decoding iterations, comprising:

means for initializing an iteration count upon receipt of a turbo code block;

means for increasing the iteration count responsive to said initializing means;

means for performing an iteration;

means for increasing the iteration count responsive to said performing means;

means for determining if a latest iteration converges;

means for generating an acknowledgement responsive to said determining means; and

means for terminating performance of further iterations.

20. The apparatus of claim 19 wherein said means for generating an acknowledgement includes means to generate an acknowledge signal (ACK) that the last iteration converged.

21. The apparatus of claim 19 further includes:

means for outputting a decoded sequence responsive to said means for terminating.

22. Apparatus employed in turbo decoding for reducing a number of decoding iterations, comprising:

means for initializing an iteration count upon receipt of a turbo code block;

means for increasing the iteration count responsive to said initializing means;

means for performing an iteration;

means for increasing the iteration count responsive to said performing means;

means for determining if a latest iteration diverges;

means for generating an acknowledgement responsive said means for generating an acknowledgement; and

means for terminating performance of further iterations.

23. The apparatus of claim 22 wherein said means for generating an acknowledgement includes means for generating an acknowledge signal (NACK) that the last iteration diverged.

24. The method of claim 22 further includes:

means for outputting a decoded sequence responsive to said means for terminating.

25. Apparatus for performing turbo decoding iterations, comprising:

means for initializing an iteration count upon receipt of a turbo code block;

means for increasing the iteration count responsive to performance of said initializing means;

means for performing a decoding iteration;

means for increasing the iteration count responsive to said performing means;

means for determining if a latest iteration converges;

means for determining if the latest iteration diverges in the event that convergence was not detected by said convergence determining means.

means for generating an acknowledgement responsive to said divergence determining means; and

means for terminating performance of further iterations responsive to said acknowledge generating means.

26. The apparatus of claim 25 wherein acknowledgement generating means includes means for generating an acknowledgement (NACK) which indicates that the latest iteration diverged.

27. The apparatus of claim 25 further includes:

means for outputting a decoded sequence responsive to said acknowledgement generating means.

28. The apparatus of claim 25 wherein performing an iteration comprises:

means for generating log-likelihood ratios (LLRs) based on a systematic bit sequence and a parity bit 1 sequence of the inputted coded data and a priori data;

means for permuting the LLRs;

means for generating extrinsic information based on the permuted LLRs and a parity 2 sequence of the coded input data;

means for de-interleaving the extrinsic information; and

means for providing the de-interleaved extrinsic information as a priori data to said means for generating LLRs.

29. The apparatus of claim 28 wherein said means for determining convergence includes:

means for making a convergence determination based upon the LLRs generated at step (i) and the de-interleaved extrinsic information generated.

30. The apparatus of claim 28 wherein said means for determining convergence includes:

means for making a divergence determination based upon the LLRs generated and the de-interleaved extrinsic information generated.

31. The apparatus of claim 29 wherein said means for determining divergence comprises means for generating an acknowledgement based upon the LLRs generated and the de-interleaved extrinsic information generated.

32. The apparatus of claim 30 wherein said means for determining divergence comprises means for generating an acknowledgement based upon the LLRs generated and the de-interleaved extrinsic information generated.

33. Apparatus for turbo decoding which reduces a number of decoding iterations, comprising:

means for generating log-likelihood ratios (LLRs) based on a systematic bit sequence and a parity bit 1 sequence of inputted coded data and a priori data;

means for permuting the LLRs;

means for generating extrinsic information based on the permuted LLRs and a parity 2 sequence of the coded input data;

means for de-interleaving the extrinsic information; and

means for providing the de-interleaved extrinsic information as a priori data to said means for generating LLRs.

34. The apparatus of claim 33 wherein said means for generating extrinsic information includes:

means for making a convergence determination based upon the LLRs generated and the de-interleaved extrinsic information generated.

35. The apparatus of claim 33 wherein said means for generating extrinsic information comprises means for making a determination based upon the LLRs generated and the de-interleaved extrinsic information generated.

36. The apparatus of claim 33 wherein said means for determining divergence includes means for generating an acknowledgement based upon the LLRs generated and the de-interleaved extrinsic information generated.

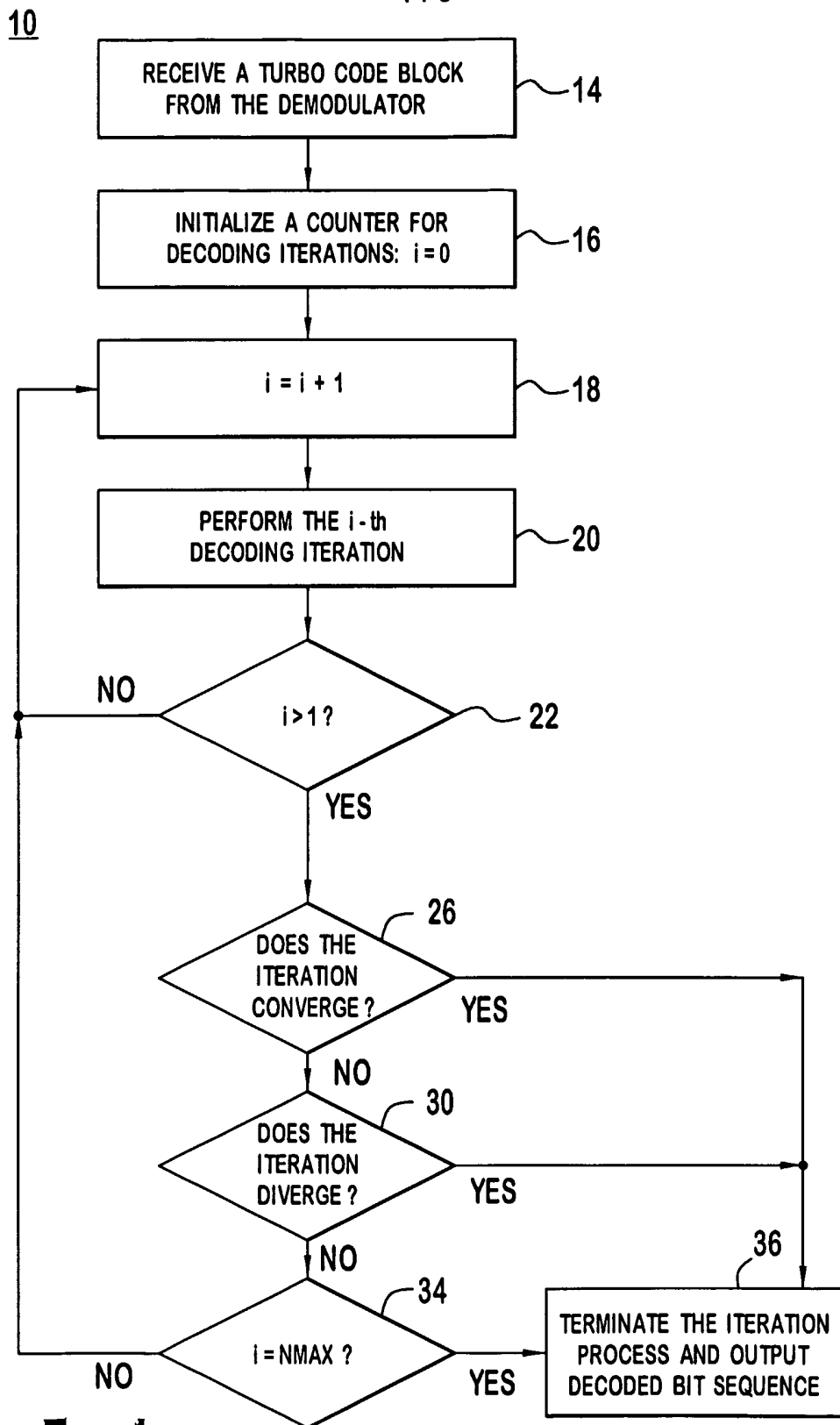


FIG. 1

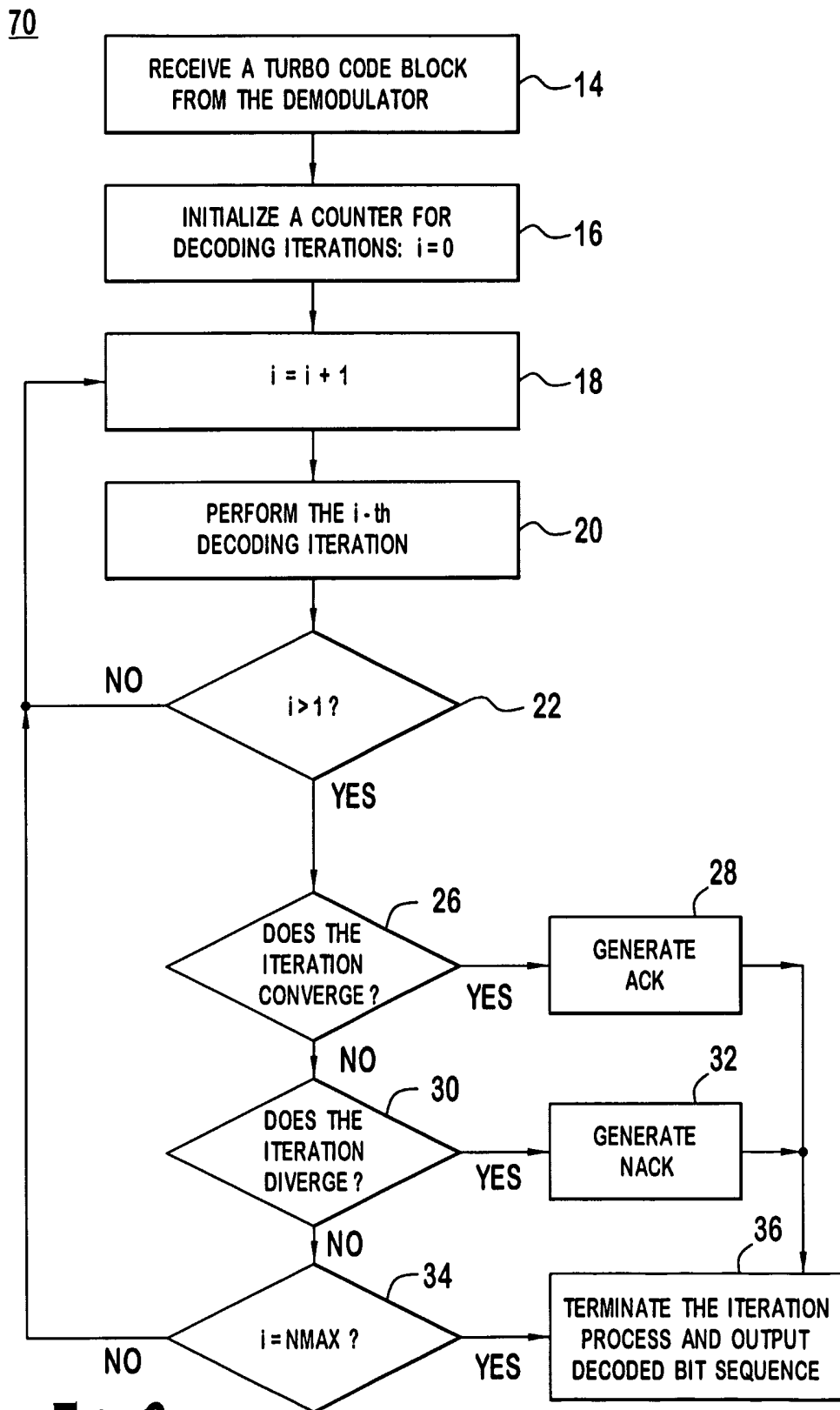
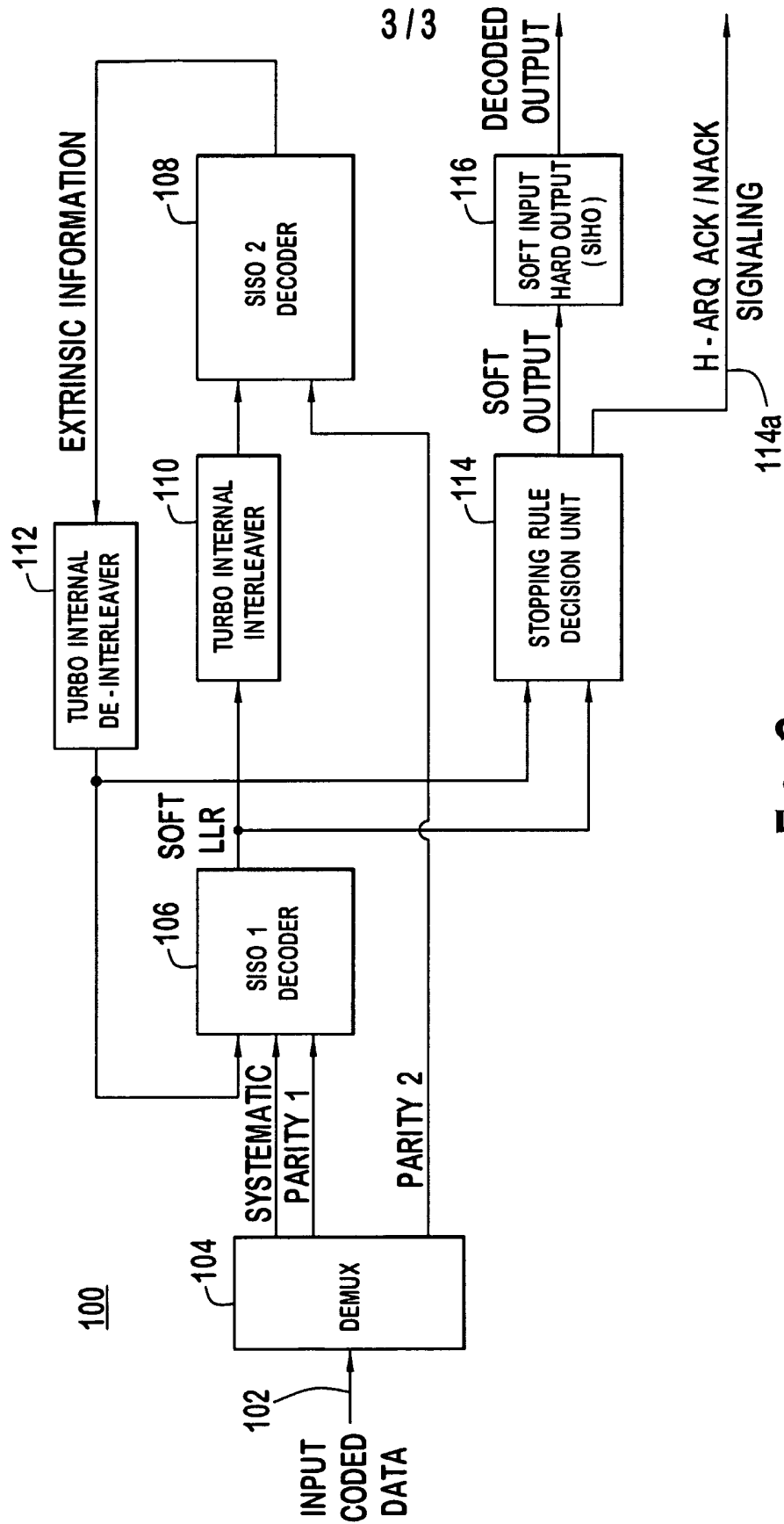


FIG. 2



3/3

FIG. 3

**INTERNATIONAL SEARCH REPORT**

International application No.

PCT/US03/19643

<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
IPC(7) : H03M 13/00 US CL : 714/755		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 714/755		
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 practicable, search terms used) EAST Database		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,518,892 B2 (SHEN et al.) 11 February 2003, columns 2-12	1-36
Y, P	US 6,557,139 B2 (Bohnke) 29 April 2003, columns 5-10	1-36
Y, P	US 6,487,694 B1 (Bajwa) 26 November 2002, columns 5-20	1-36
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:		
"A"	document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier application or patent published on or after the international filing date	"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
"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)	"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
"O"	document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search	Date of mailing of the international search report	
14 August 2003 (14.08.2003)	12 SEP 2003	
Name and mailing address of the ISA/US	Authorized officer	
Mail Stop PCT, Attn: ISA/US	Albert B. Cady	
Commissioner for Patents	Telephone No. 703-305-3900	
P.O. Box 1450		
Alexandria, Virginia 22313-1450		
Facsimile No. (703)305-3230		