US 20070011557A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2007/0011557 A1

Zheng et al. (43) Pub. Date: Jan. 11, 2007

(54) INTER-SEQUENCE PERMUTATION TURBO CODE SYSTEM AND OPERATION METHODS THEREOF

(75) Inventors: **Yan-Xiu Zheng**, Shulin City (TW); **Yu-Ted Su**, Hsinchu City (TW)

Correspondence Address:
LADAS & PARRY
26 WEST 61ST STREET
NEW YORK, NY 10023 (US)

(73) Assignee: **HIGHDIMENSION LTD.**

(21) Appl. No.: **11/176,829**

(22) Filed: **Jul. 7, 2005**

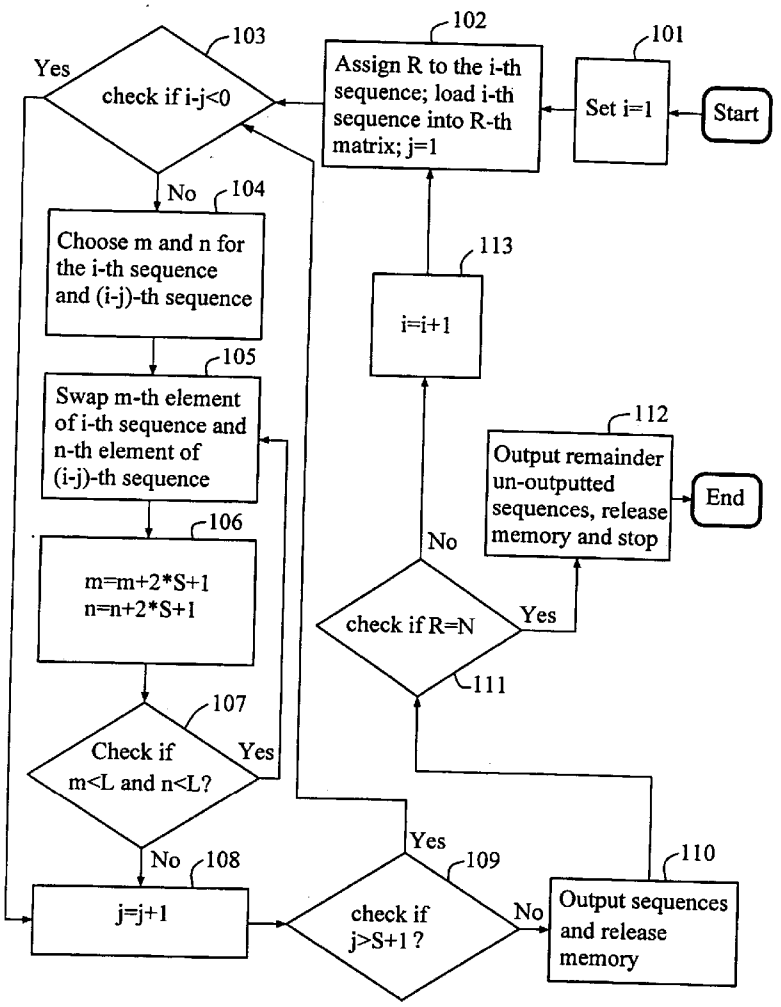
Publication Classification

(51) **Int. Cl.**
H03M 13/05 (2006.01)

(52) **U.S. Cl.** **714/752**

(57) **ABSTRACT**

A high performance real-time turbo code system is proposed. The proposed system exploits cooperative coding architecture and a proper decoding scheduling to achieve low error rate within a constrained latency. Permutation schemes and hardware embodiments utilizing the cooperative coding are also shown. Various memory saving techniques are provided to reduce memory usage in both encoder and decoder. The proposed system is compatible with 3rd generation mobile standards and cost of designing new parts exclusively for the proposed system can be minimized. This invention can provide substantial coding and system capacity gains for real-time applications in a wireless environment.



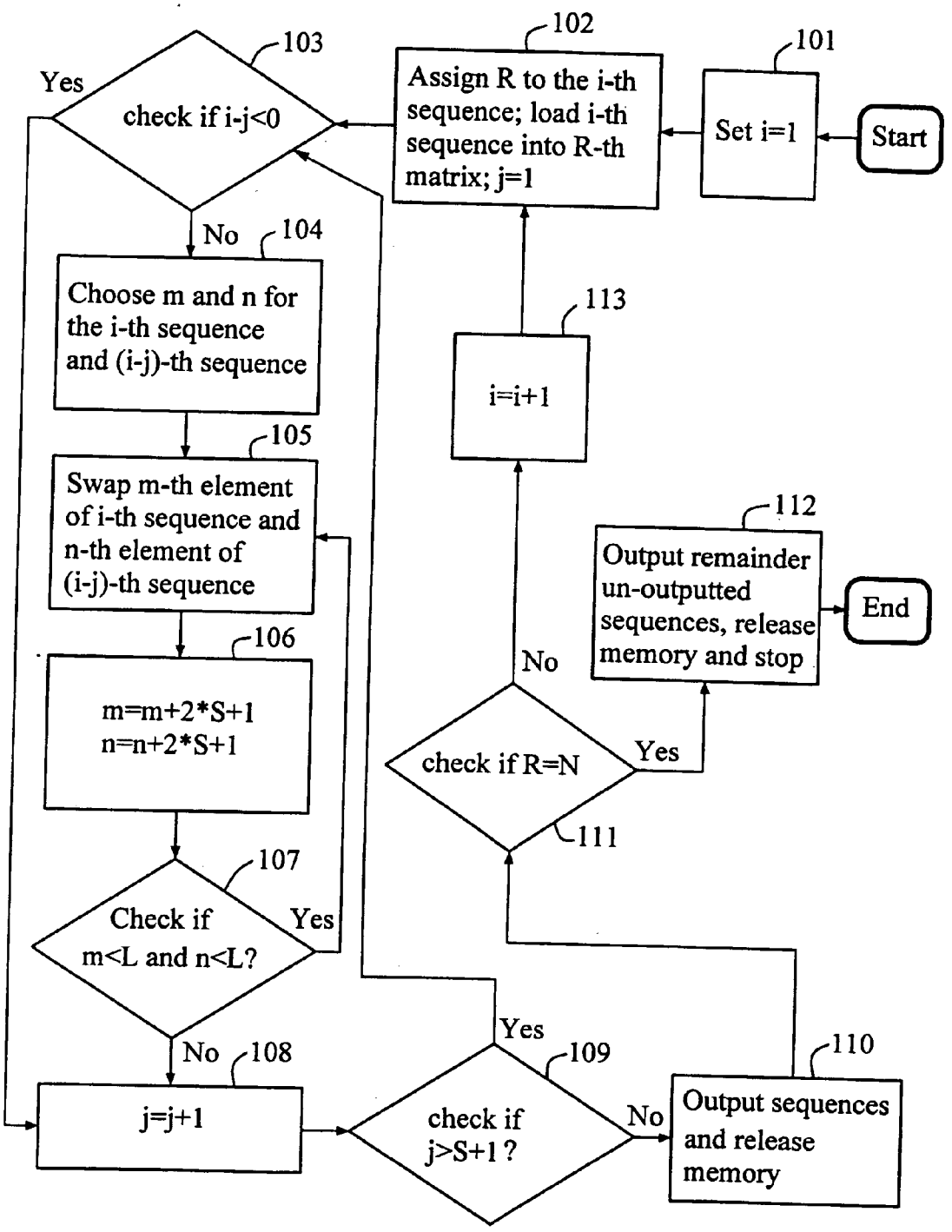

Start
Set i=1
101
Assign R to the i-th sequence; load i-th sequence into R-th matrix; j=1
102
check if i-j<0
103
Yes
No
Choose m and n for the i-th sequence and (i-j)-th sequence
104
Swap m-th element of i-th sequence and n-th element of (i-j)-th sequence
105
m=m+2*S+1
n=n+2*S+1
106
Check if m<L and n<L?
107
Yes
No
j=j+1
108
check if j>S+1 ?
109
Yes
No
Output sequences and release memory
110
check if R=N
111
Yes
No
i=i+1
113
Output remainder un-outputted sequences, release memory and stop
112
End

FIG.1

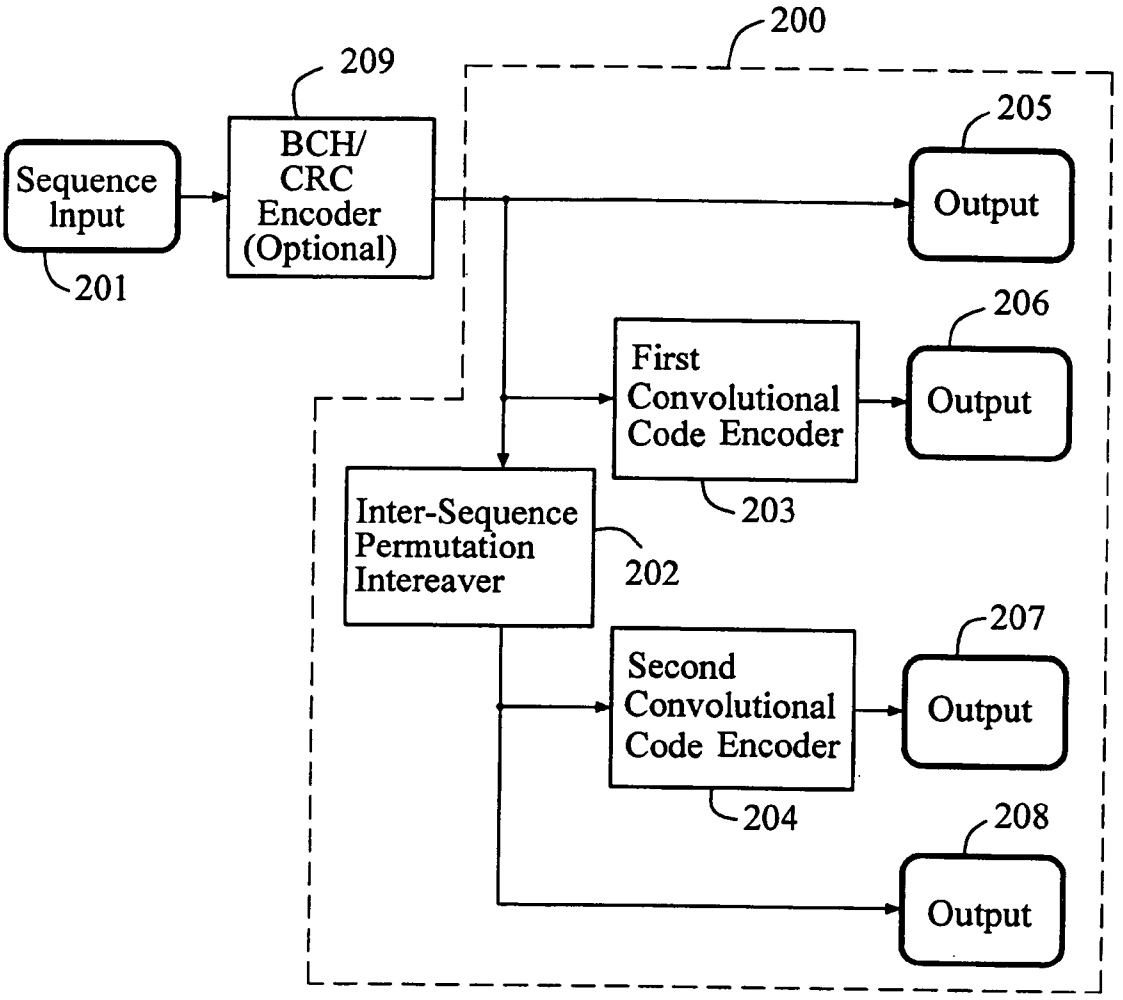

200
209
205
BCH/
CRC
Encoder
(Optional)
Sequence
Input
Output
201
206
First
Convolutional
Code Encoder
Output
203
Inter-Sequence
Permutation
Intereaver
202
207
Second
Convolutional
Code Encoder
Output
204
208
Output

FIG.2

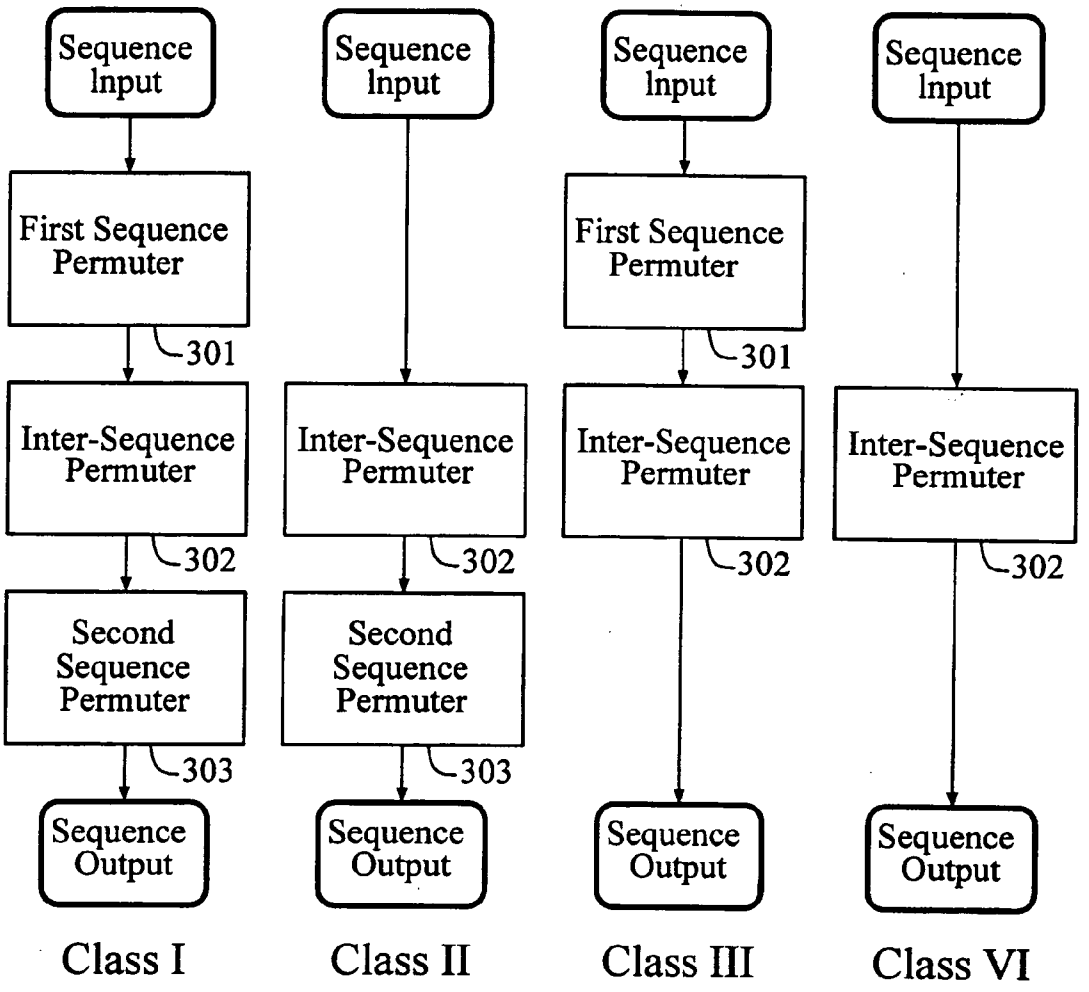

Sequence Input
First Sequence Permuter
301
Inter-Sequence Permuter
302
Second Sequence Permuter
303
Sequence Output
Class I
Sequence Input
Inter-Sequence Permuter
302
Second Sequence Permuter
303
Sequence Output
Class II
Sequence Input
First Sequence Permuter
301
Inter-Sequence Permuter
302
Sequence Output
Class III
Sequence Input
Inter-Sequence Permuter
302
Sequence Output
Class VI

FIG.3

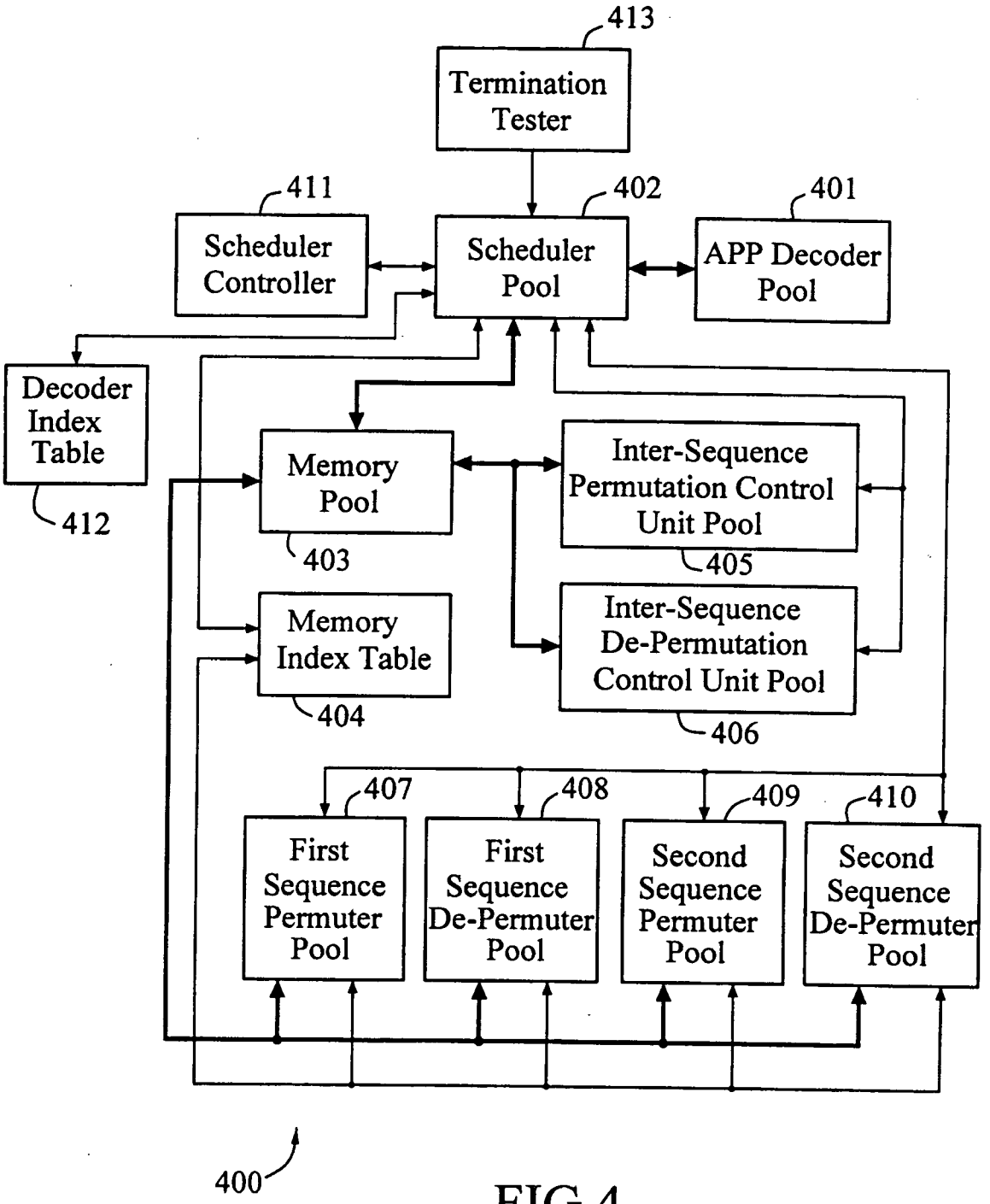

413
Termination Tester
411
Scheduler Controller
402
Scheduler Pool
401
APP Decoder Pool
Decoder Index Table
412
Memory Pool
403
Inter-Sequence Permutation Control Unit Pool
405
Memory Index Table
404
Inter-Sequence De-Permutation Control Unit Pool
406
407
First Sequence Permuter Pool
408
First Sequence De-Permuter Pool
409
Second Sequence Permuter Pool
410
Second Sequence De-Permuter Pool
400

FIG.4

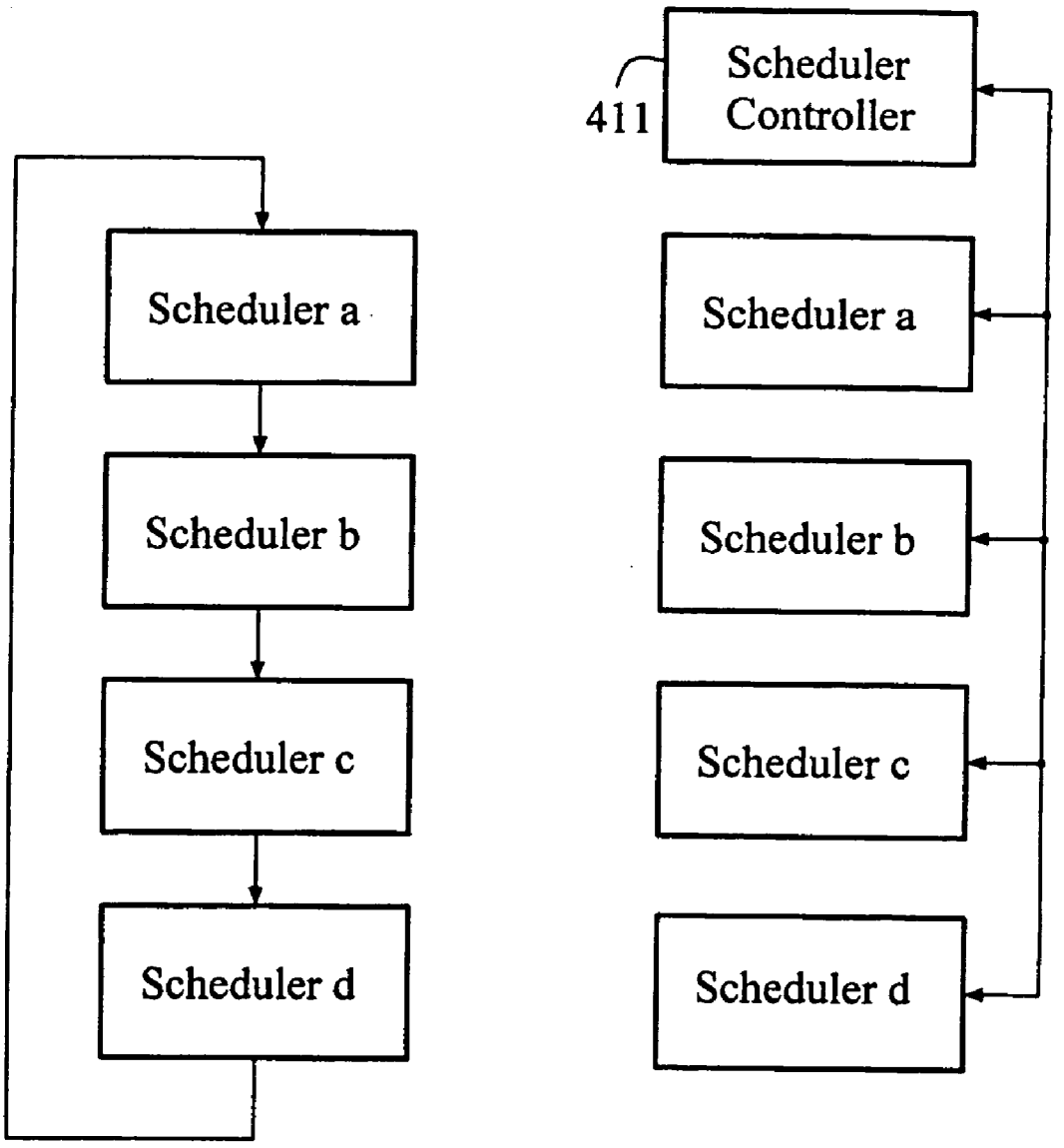

Scheduler a
Scheduler b
Scheduler c
Scheduler d
Ring Type
411
Scheduler Controller
Scheduler a
Scheduler b
Scheduler c
Scheduler d
Star Type

FIG.5

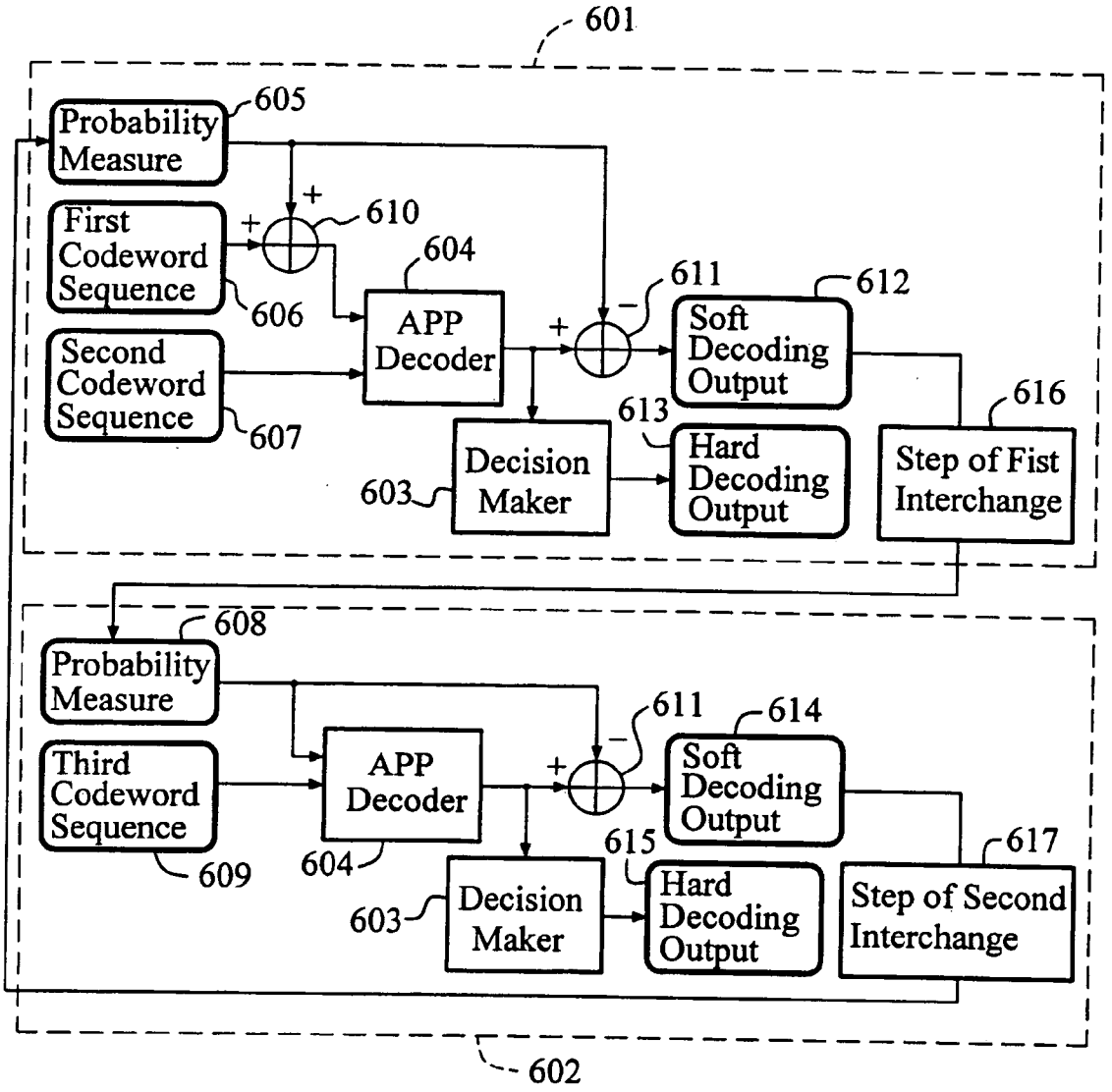

601
Probability Measure
605
First Codeword Sequence
606
Second Codeword Sequence
607
610
604
APP Decoder
611
612
Soft Decoding Output
613
Hard Decoding Output
603
Decision Maker
616
Step of Fist Interchange
608
Probability Measure
Third Codeword Sequence
609
APP Decoder
604
611
614
Soft Decoding Output
615
Hard Decoding Output
603
Decision Maker
617
Step of Second Interchange
602

FIG.6

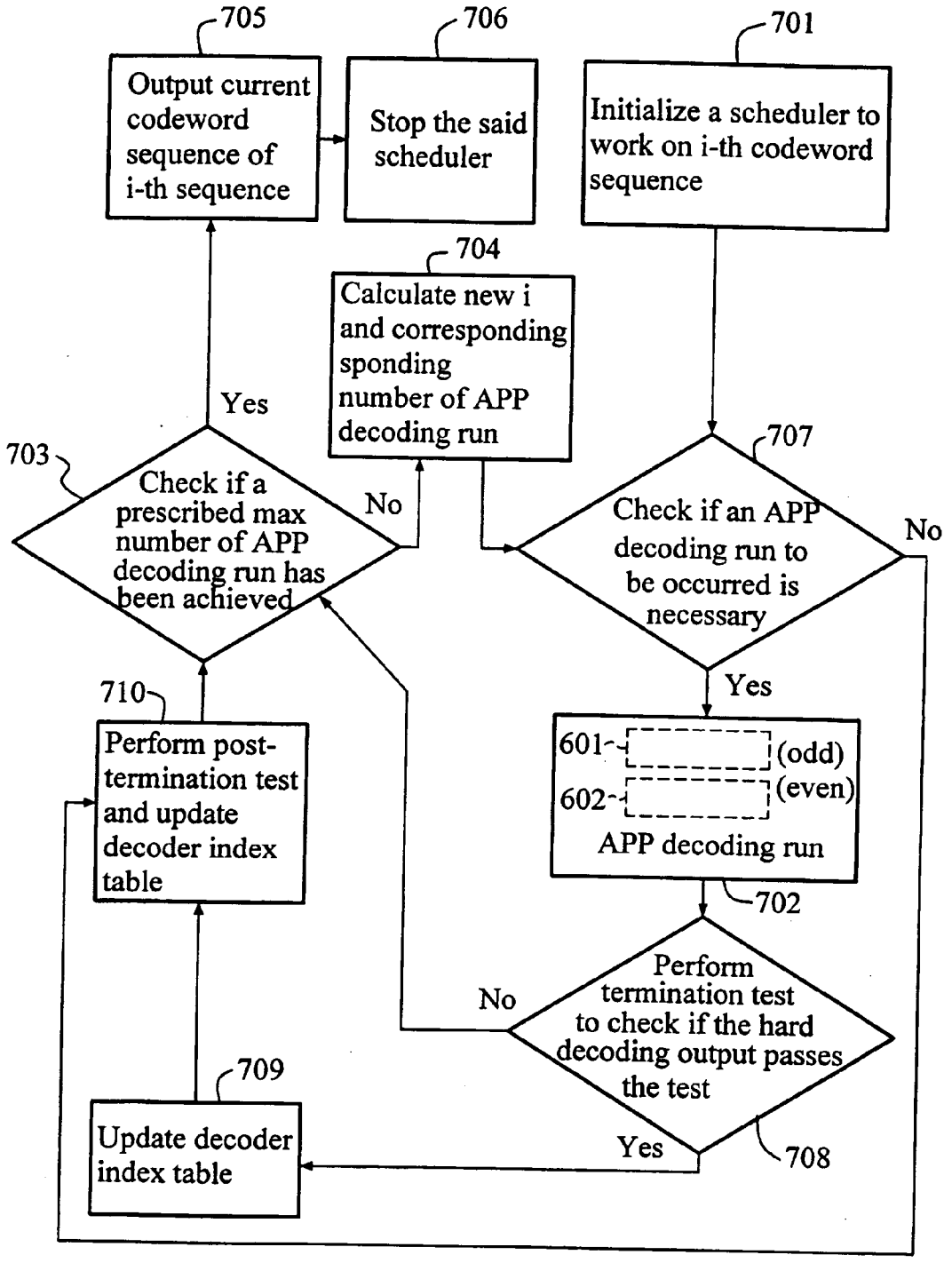

705
Output current codeword sequence of i-th sequence
706
Stop the said scheduler
701
Initialize a scheduler to work on i-th codeword sequence
704
Calculate new i and corresponding sponding number of APP decoding run
Yes
703
Check if a prescribed max number of APP decoding run has been achieved
No
707
Check if an APP decoding run to be occurred is necessary
No
Yes
710
Perform post-termination test and update decoder index table
601
(odd)
602
(even)
APP decoding run
702
Perform termination test to check if the hard decoding output passes the test
No
709
Update decoder index table
Yes
708

FIG.7

INTER-SEQUENCE PERMUTATION TURBO CODE SYSTEM AND OPERATION METHODS THEREOF

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to a turbo code system, more specifically, a turbo code system utilizing cooperative coding architecture and a proper decoding scheduling to achieve high performance real-time encoding and decoding.

[0003] 2. Description of Related Art

[0004] Turbo Code (TC) was invented in 1993, which renders extraordinary, near Shannon limit performance by applying the iterative decoding algorithm. Following researches on the area of Forward Error-Control (FEC) were inspired from this primitive code structure and decoding algorithm. We shall thus refer to any FEC system that utilizes the so-called "Turbo Principle" in decoding as a Turbo Code System (TCS).

[0005] Codeword length influences the performance of TCS. TCS with long codeword length performs excellently but decoder of the TCS has large decoding latency and hardware complexity. Moreover, the decoder requires considerable number of iterations to achieve desired performance. The TCS with moderate codeword length often gives unsatisfactory performance. The TCS with short codeword length (say <200) often only provides performance worse than that of conventional coding schemes.

[0006] Therefore, the conventional TCS renders high complexity, long decoding latency and large memory space consumption; thereby diminishing their applicability. Commercial FEC applications require affordable complexity, low decoding latency and low power consumption. Furthermore, for use in a future generation wireless communication system, it is preferred that any new enhancement be backward compatible with current air interface standard. It will be shown in the following that the present invention does satisfy all these requirements.

SUMMARY OF THE INVENTION

[0007] Cooperative decoding can improve performance of TCS with short codeword length. Besides, conventional A Posteriori Probability (APP) decoding modules and interleaving techniques can be applied and the feature of "Backward Compatible" is attainable. More than one schedulers can be applied for scheduling of cycles of APP decoding (or called APP decoding runs) or memory releasing.

[0008] Inter-Sequence Permutation (ISP) is a concept permuting between different sequences, and decoder of TCS can apply ISP to do cooperative decoding. A long sequence of codeword can be chopped into shorter sequences first, and by utilizing ISP, these shorter sequences can be subsequently decoded at decoder side simultaneously so as to achieve the goal of parallel decoding. The ISP algorithm permuting these sequences can be simple and require little effort. TCS applying ISP concept is called cooperative TCS and a turbo code applying the ISP concept is called ISP turbo code.

[0009] The proposed ISP turbo code can incorporate existing TC. Encoders of existing devices only need minor modification upon introducing the ISP permutation technique. CRC and BCH codes or the like are optional for termination test or error correction.

[0010] Memory usage would be the most critical implementation problem for the cooperative TCS. Decoding more than ten sequences at the same time requires large memory space for the temporary received samples. Moreover, an ISP between sequences also requires buffers storing probability measure for the nearby sequences. In the present invention, a termination test is used to halt decoding. In cooperative TCS, the termination test can be further used for providing more reliable probability measure and releasing memory. In summary, the termination test reduces power consumption and decoding latency, assists in the decoding of the other sequences, and makes the utilization of memory economic.

[0011] Proposed dynamic memory assignment decoder architecture can: i) reducing the average decoding latency and the computation power consumption; ii) minimizing the memory usage; iii) lowering down the average iterations at high error rate region; iv) parallel decoding; v) effective utilizing the APP decoders.

[0012] Physical architecture of the ISP turbo code system of the present invention comprises two parts, which are an ISP turbo code encoder and an ISP turbo code decoder.

[0013] The ISP turbo code encoder is used for generating a pre-permutation codeword sequence output before an ISP and a post-permutation codeword sequence output after the ISP from a sequence input, characterized in comprising an ISP interleaver within, wherein the said ISP interleaver is composed by an inter-sequence permuter and at least one conventional sequence permuter arranged in a one-by-one manner; and wherein the inter-sequence permuter of the ISP interleaver performing ISP comprises at least an ISP control unit and a memory pool, furthermore, an ISP algorithm is permanently embedded or temporally recorded in the ISP control unit controlling inputting to the memory pool, outputting from the memory pool, and execution of ISP between sequences stored in the memory pool.

[0014] The ISP turbo code decoder receiving the pre-permutation codeword sequence output and post-permutation codeword sequence output transmitted by the said ISP turbo code encoder, wherein the said ISP turbo code decoder decodes the said sequences by at least one a posteriori probability (APP) decoder therein, characterized in that decoding runs of the APP decoder is controlled by at least one scheduler and the decoding runs are performed in a loop manner so that the APP decoder can repeatedly be used in decoding.

[0015] Details of apparatus and operations mentioned above will be discussed in more detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The present invention is described below by way of examples with reference to the accompanying drawings which will make it easier for readers to understand the purpose, technical contents, characteristics and achievement of the present invention.

[0017] FIG. **1** is an operation flowchart of an example of inter-sequence permutation.

[0018] FIG. **2** is a schematic drawing of an example of ISP turbo code encoder.

[0019] FIG. **3** is a schematic drawing showing four possible arrangements of an ISP interleaver.

[0020] FIG. **4** is a schematic drawing of an example of ISP turbo code decoder.

[0021] FIG. **5** is a schematic drawing showing two types of scheduler arrangements.

[0022] FIG. **6** is a schematic drawing showing an example of operations of odd-numbered and even-numbered APP decoding runs.

[0023] FIG. **7** is a schematic drawing showing an example of operations of scheduling performed by a scheduler.

DETAILED DESCRIPTION OF THE INVENTION

[0024] The details to the exemplary embodiments of the invention will be described as follows and the same reference numbers are used throughout the drawings to refer to the same or like parts.

[0025] As stated above, ISP is a concept permuting between different sequences, which is easy to be comprehended for persons skilled in the art. For ease of understanding, we can take the sequences as matrices. One element in one matrix, if it is to be permuted, will be moved to one coordinate in the same or another matrix with the ordinal number within a span prior or successive to the element. In the present invention, preferably, one sequence is to be inter-sequence permuted with two other sequences which are consecutively before and after the sequence to be permuted. One exemplary procedure of ISP is given in FIG. **1**, wherein i is ordinal number of sequences, R and P are ordinal numbers of matrices, m and n are ordinal numbers of elements in the R-th and P-th matrices, where $0 \leqq m,n<L$ and L is the length of sequence, respectively, j is a variable used in following operations, M is memory size (in term of maximum number of sequences stored), S is a variable called ISP span which $S<M-1$, and N is total number of sequences, as follows:

[0026] step **101**: set initial value of i=1; then go to step **102**;

[0027] step **102**: Assign R for the i-th sequence, where R-th matrix is not in use; Register R-th matrix as in-use, load i-th sequence to R-th matrix with length L; j=1; then go to step **103**;

[0028] step **103**: if i−j<0, then go to step **108**; otherwise, go to step **104**;

[0029] step **104**: choose m and n for the i-th and (i−j)-th sequences, respectively, so that one element only swaps with one element of another sequence throughout the ISP process; then go to step **105**;

[0030] step **105**: swap m-th element of i-th sequence and n-th element of (i−j)-th sequence; then go to step **106**;

[0031] step **106**: m=m+2*S+1 and n−n+2*S+1; then go to step **107**;

[0032] step **107**: if m<L and n<L, then go to step **105**; otherwise, go to step **108**;

[0033] step **108**: j=j+1; then go to step **109**;

[0034] step **109**: if j<S+1, then go to step **103**; otherwise, go to step **110**;

[0035] step **110**: output sequences done by ISP and register the matrix corresponding to outputted sequence as non-use; then go to step **111**;

[0036] step **111**: if i=N, then output remaining un-outputted sequences, register the matrix corresponding to outputted sequence non-use and stop (step **112**); otherwise, i=i+1 (step **113**) then go to step **102**;

[0037] One should understand that the above example is just an illustration of one possible ISP algorithm. Many ISP algorithms are available to be used as long as they meet the definitions (one element in one matrix, if it is to be permuted, will be moved to one coordinate in the same or another matrix with the ordinal number within a span prior or successive to the element.)

[0038] FIG. **2** illustrates a schematic drawing showing a typical ISP turbo code encoder **200** according to the present invention. The ISP turbo code encoder generates a pre-permutation codeword sequence output before an ISP, and a post-permutation codeword sequence output after the ISP from a sequence input **201**, characterized by comprising an ISP interleaver **202** therein.

[0039] Illustrated in FIG. **3**, the ISP interleaver is composed by an inter-sequence permuter **302** and at least one conventional sequence permuter arranged in a one-by-one manner, which will be discussed more in detail below. The inter-sequence permuter **302** of the ISP interleaver **202** performing ISP comprises at least an ISP control unit and a memory pool; furthermore, an ISP algorithm is permanently embedded or temporally recorded in the ISP control unit controlling inputting to the memory pool, outputting from the memory pool, and execution of ISP between sequences stored in the memory pool.

[0040] Four possible classes of arrangement of the ISP interleaver **202** are illustrated, as follows:

[0041] Class I: comprising a first sequence permuter **301** utilizing a conventional sequence permuting algorithm, the inter-sequence permuter **302**, and a second sequence permuter **303** utilizing a conventional sequence permuting algorithm, wherein the conventional sequence permuting algorithm utilized in the first sequence permuter **301** and second sequence permuter **303** can be different or identical, and a sequence inputted into the ISP interleaver **202** is processed in the order of the first sequence permuter **301**, the inter-sequence permuter **302**, and then the second sequence permuter **303**.

[0042] Class II: comprising the inter-sequence permuter **302** and a second sequence permuter **303** utilizing a conventional sequence permuting algorithm, wherein a sequence inputted into the ISP interleaver **202** is processed in the order of the inter-sequence permuter **302** and then the second sequence permuter **303**.

[0043] Class III: comprising a first sequence permuter **301** utilizing a conventional sequence permuting algorithm and the inter-sequence permuter **302**, wherein a sequence inputted into the ISP interleaver **202** is processed in the order of the first sequence permuter **301** and then the inter-sequence permuter **302**.

[0044] Class IV: comprising the inter-sequence permuter **302**, wherein a sequence inputted into the ISP interleaver **202** is processed by the inter-sequence permuter **302**.

[0045] Back to FIG. **2**, the ISP turbo code encoder **200** comprises the ISP interleaver **202** and two convolutional code encoders, namely a first convolutional code encoder **203** and a second convolutional code encoder **204**, located in portions of the ISP turbo code encoder **200** before and after the ISP interleaver **202**, respectively.

[0046] As illustrated in the drawing, the pre-permutation codeword sequence output comprises two sequence outputs, which are the sequence output **205** of "original sequence from the sequence input" and sequence output **206** of "original sequence processed by the first convolutional code encoder **203**." Similarly, the post-permutation sequence output comprises two sequence outputs, which are the sequence output **207** of "original sequence processed by and in the order of the ISP interleaver **202** and the second convolutional code encoder **204**", and the sequence output **208** of "original sequence processed by the ISP interleaver **202**."

[0047] In practical application, only three sequence outputs out from the four sequence outputs **205**, **206**, **207** and **208** abovementioned are required, which can be chosen from only one of the two sets of sequence outputs: the sequence outputs **205**, **206** and **207**, or the sequence outputs **208**, **207** and **206**.

[0048] Further, as illustrated in FIG. **2**, the ISP turbo code encoder **200** is further provided with an optional encoder **209** located between the sequence input **201** and the ISP turbo code encoder **200**. For example, the encoder **209** can be a BCH or CRC encoder.

[0049] Now, please refer to FIG. **4** for an embodiment of an ISP turbo code decoder **400**. The present invention employs a distributed design so as to attain the goal of do-loop operation.

[0050] The decoder **400** comprises an APP decoder pool **401** composed of at least one APP decoder; a scheduler pool **402** composed of at least one scheduler; a memory pool **403** composed of a plurality of memory units storing sequences; a memory index table **404** storing relationship information between the memory units and probability measure sequences, and location of a specific sequences in the memory pool can be located by this table; an ISP control unit pool **405** composed of at least one ISP control unit; an inter-sequence de-permutation (ISDP) control unit pool **406** composed of at least one ISDP control unit; a first sequence permuter pool **407** composed of at least one first sequence permuter; a first sequence de-permuter pool **408** composed of at least one first sequence de-permuter; a second sequence permuter pool **409** composed of at least one second sequence permuter; and a second sequence de-permuter pool **410** composed of at least one second sequence de-permuter. A de-permuter runs like a permuter in reverse manner.

[0051] Wherein the scheduler pool **402** controls operations of the APP decoder pool **401**, the ISP control unit pool **405**, the ISDP control unit pool **406**, the first sequence permuter pool **407**, the first sequence de-permuter pool **408**, the second sequence permuter pool **409** and the second sequence de-permuter pool **410**. In detail, a scheduler controls each cycle of APP decoding (hereafter referred to as an "APP decoding run"), which relates to ISP, ISDP, conventional sequence permutation, or related arithmetic operation. Schedulers will be coordinated so that preferably all components in the ISP turbo code decoder **400** work and cooperate seamlessly. It will be discussed in more detail later.

[0052] The scheduler pool **402** provides and retrieves sequences into and from the memory pool **403**. The scheduler pool **402** provides and retrieves sequences to and from the APP decoder pool **401**. The scheduler pool **402** updates and retrieves information to and from the decoder index table **412** and memory index table **404**. The ISP control unit pool **405** and ISDP control unit pool **406** interchange sequences with the memory pool **403**. The first sequence permuter pool **407**, the first sequence de-permuter pool **408**, the second sequence permuter pool **409**, and the second sequence de-permuter pool **410** interchange sequences with the memory pool **403**. The scheduler pool **402** comprises at least one adder **610** and subtracter **611** (both are not shown in FIG. **4**).

[0053] Note that in all drawings in this specification, thick lines, e.g. between the APP decoder pool **401** and scheduler pool **402**, represent bus for transmitting sequences/control signals, and narrow lines, e.g. between the scheduler controller **411** and scheduler pool **402**, represent signal lines for transmitting control signals only.

[0054] Also note that arrangement of this embodiment would be modified according to the ISP interleaver **202** used in the ISP turbo code encoder **200**. What is illustrated is only for ISP interleaver **202** of Class I of FIG. **3**. If ISP interleaver **202** of Class II in FIG. **3** is used, the first sequence permuter pool **407** and the first sequence de-permuter pool **408** are not required. If ISP interleaver **202** of Class III in FIG. **3** is used, the second sequence permuter pool **409** and the second sequence de-permuter pool **410** are not required. If ISP interleaver **202** of Class IV in FIG. **3** is used, the first sequence permuter pool **407**, the first sequence de-permuter pool **408**, the second sequence permuter pool **409** and the second sequence de-permuter pool **410** are all not required.

[0055] FIG. **4** only illustrates relative relations between components therein. Therefore, no signal input/output is indicated. Operations of the components comprising signal input/output will be illustrated in FIG. **6**.

[0056] Further, the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance to scale or format of the sequences. For instance, if values in sequences are in logarithm-based, an adder and a subtracter are used.

[0057] With reference to FIG. **5**, schedulers in the scheduler pool **402** can be arranged in "ring type" or "star type", as illustrated. In the ring type, one scheduler is controlled by commands transmitted by a preceding scheduler. If the star type is used, a scheduler controller **411** is required, which is connected to all schedulers and coordinates operation of all schedulers. Operation of schedulers will be discussed in detail later.

[0058] Further, at least one decision maker **603** (shown in FIG. **6**) for outputting a hard decoding output sequence is included in the scheduler pool **402**. A "hard decoding output" is one kind of digitally altered decoding output of which all values are bits (symbols), so that ambiguous values are eliminated.

[0059] Further, the ISP turbo code decoder **400** comprises a decoder index table **412** for storing information on the relationship between necessity to perform APP decoding and codeword sequence numbers. The decoder index table at least is connected and interchanges information with the scheduler pool **402**. If a codeword sequence is marked as "unnecessary", then it will not go through APP decoding.

[0060] The scheduler pool **402** is connected to at least one termination tester **413** for performing a termination test, which is a test to check correctness or convergence of a sequence. Conventional tests such as CRC and sign check can be used.

[0061] Refer to FIG. **6** for a method of operation of APP decoding runs. An APP decoding run block **601** illustrates operation of odd-numbered APP decoding run, and an APP decoding run block **602** illustrates operation of even-numbered APP decoding run.

[0062] If the three codeword sequence outputs from the ISP turbo code encoder **200** are sequence output **205**, sequence output **206**, and sequence output **207**, then a pre-permutation codeword sequence received from the ISP turbo code encoder, i.e. original sequence from the sequence input and the sequence output of original sequence processed by the first convolutional code encoder, is processed in odd-numbered APP decoding runs and post-permutation codeword sequence received from the ISP turbo code encoder, i.e. the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, is processed in even-numbered APP decoding runs.

[0063] The original sequence from the sequence input is called a first codeword sequence **606**, the sequence output of original sequence processed by the first convolutional code encoder is called a second codeword sequence **607**, and the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder is called a third codeword sequence **609**.

[0064] For odd-numbered APP decoding run, it comprises the following steps:

[0065] Step of first APP decoder input: a first input of the APP decoder **604** is calculated by combining a sequence of a priori probability measure **605** and the first codeword sequence **606** through an adder **610** of the scheduler pool **402**, and then the process goes to the step of second APP decoder input.

[0066] Step of second APP decoder input: the second codeword sequence **607** is inputted into the APP decoder **604** as a second input, and then the process goes to the step of outputting first result.

[0067] Step of outputting first result: the APP decoder **604** outputs a first result probability measure sequence and then the process goes to the step of generating first soft decoding output;

[0068] step of generating first soft decoding output: a first sequence of soft decoding output **612** is calculated by eliminating the sequence of a priori probability measure **605** from the first result probability measure sequence through a subtracter **611** of the scheduler pool **402**, and then the process goes to the step of first interchange **616**;

[0069] step of first interchange **616** (details will be given later): the first sequence of soft decoding output **612** is the sequence of a priori probability measure **608** of subsequent even-numbered APP decoding run. However, since even-numbered APP decoding runs works on post-permutation codeword sequences, permutation must be performed on the first sequence of soft decoding output **612** before it can be used in the subsequent even-numbered APP decoding run.

[0070] For even-numbered APP decoding run, it comprises the following steps:

[0071] Step of third APP decoder input: an APP decoder **604** receives two inputs which are the sequence of a priori probability measure sequence **608** in step of first interchange and the third codeword sequence **609**, and outputs a second result probability measure sequence, wherein the APP decoder **604** can be or not be the same one as used in the odd-numbered APP decoding run; then the process goes to the step of outputting second result.

[0072] Step of outputting second result: a second sequence of soft decoding output **614** is calculated by eliminating the sequence of a priori probability measure **608** in the step of first interchange or the step of third APP decoder input from the second result probability measure sequence through the subtracter **611** of the scheduler pool, which can be or not be the same as used in the odd-numbered APP decoding run, and then the process goes to the step of second interchange **617**.

[0073] Step of second interchange **617** (details will be given later): the second sequence of soft decoding output **614** is the sequence of a priori probability measure **605** of subsequent odd-numbered APP decoding run. However, since odd-numbered APP decoding runs works on pre-permutation codeword sequences, de-permutation must be performed on the second sequence of soft decoding output **614** before it can be used in subsequent odd-numbered APP decoding run.

[0074] Alternatively, if the three codeword sequence outputs from the ISP turbo code encoder are sequence output **206**, sequence output **207**, and sequence output **208**, then pre-permutation codeword sequence received from the ISP turbo code encoder, i.e. the sequence output of original sequence processed by the first convolutional code encoder, is processed in even-numbered APP decoding runs and post-permutation codeword sequence received from the ISP turbo code encoder, i.e. the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, and the sequence output of original sequence processed by the ISP interleaver, is processed in odd-numbered APP decoding runs.

[0075] The sequence output of original sequence processed by the ISP interleaver **202** is called a first codeword sequence **606**, the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder is called a second codeword sequence **607**, and the sequence output of original sequence processed by the first convolutional code encoder is called a third codeword sequence **609**,

[0076] For odd-numbered APP decoding run, it comprises the following steps:

[0077] Step of first APP decoder input: a first input of the APP decoder **604** is calculated by combining a sequence of a priori probability measure **605** and the first codeword sequence **606** through an adder **610** of the scheduler pool **402**, and then the process goes to step of second APP decoder input.

[0078] Step of second APP decoder input: the second codeword sequence **607** is inputted into the APP decoder **604** as a second input, and then the process goes to the step of outputting first result.

[0079] Step of outputting first result: the APP decoder **604** outputs a first result probability measure sequence, and then the process goes to the step of generating first soft decoding output.

[0080] Step of generating first soft decoding output: a first codeword sequence of soft decoding output **612** is calculated by eliminating the sequence of a priori probability measure **605** from the first result probability measure sequence through a subtracter **611** of the scheduler pool **402**, and then the process goes to the step of first interchange **616**;

[0081] step of first interchange **616** (details will be given later): the first sequence of soft decoding output **612** is the sequence of a priori probability measure **608** of subsequent even-numbered APP decoding run. However, since even-numbered APP decoding runs works on pre-permutation codeword sequences, de-permutation must be performed on the first sequence of soft decoding output **612** before it can be used in the subsequent even-numbered APP decoding run.

[0082] For even-numbered APP decoding run, it comprises the following steps:

[0083] Step of third APP decoder input: an APP decoder **604** receives two inputs which are the sequence of a priori probability measure **608** in step of first interchange and the third codeword sequence **609** and outputs a second result probability measure sequence, wherein the APP decoder **604** can be or not be the same one as used in the odd-numbered APP decoding run; then the process goes to the step of outputting second result.

[0084] Step of outputting second result: a second sequence of soft decoding output **614** is calculated by eliminating the sequence of a priori probability measure **608** in the step of first interchange or the step of third APP decoder input from the second result probability measure sequence through the subtracter **611** of the scheduler pool **402**, which can be or not be the same as used in the odd-numbered APP decoding run, and then the process goes to the step of second interchange **617**;

[0085] Step of second interchange **617** (details will be given later): the second sequence of soft decoding output **614** is the sequence of a priori probability measure **605** of subsequent odd-numbered APP decoding run. However, since odd-numbered APP decoding runs works on post-permutation codeword sequences, permutation must be performed on the second sequence of soft decoding output **614** before it can be used in subsequent odd-numbered APP decoding run.

[0086] In the step of first interchange **616** and the step of second interchange **617**, "permutation" is performed according to any one of four classes of the ISP interleaver **202** used in the ISP turbo code encoder **200**, as follows:

[0087] If the ISP interleaver **202** of Class I in FIG. **3** is used in encoder side, the permutation is performed by and in the order of a first sequence permuter **301** in the first sequence permuter pool **407**, an ISP control unit in the ISP control unit pool **405** which works with the memory pool **403**, and a second sequence permuter **303** in the second sequence permuter pool **409**, in the process of the ISP interleaver **202**.

[0088] If the ISP interleaver **202** of Class II in FIG. **3** is used in encoder side, the permutation is performed by and in the order of an ISP control unit in the ISP control unit pool **405** which works with the memory pool **403**, and a second sequence permuter **303** in the second sequence permuter pool **409**, in the process of the ISP interleaver **202**.

[0089] If the ISP interleaver **202** of Class III in FIG. **3** is used in encoder side, the permutation is performed by and in the order of a first sequence permuter **301** in the first sequence permuter pool **407**, and an ISP control unit in the ISP control unit pool **405** which works with the memory pool **403**, in the process of the ISP interleaver **202**.

[0090] If the ISP interleaver **202** of Class IV in FIG. **3** is used in encoding side, the permutation is performed by an ISP control unit in the ISP control unit pool **405** which works with the memory pool **403**.

[0091] The "de-permutation" performed in the step of second interchange **617** or step of first interchange **616** is performed according to any one of four classes of the ISP interleaver **202** used in the ISP turbo code encoder **200**, as follows:

[0092] If the ISP interleaver **202** of Class I in FIG. **3** is used in encoder side, the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool **410**, an ISDP control unit in the ISDP control unit pool **406** which works with the memory pool **403**, and a first sequence de-permuter in the first sequence de-permuter pool **408**, in the reverse process of the ISP interleaver **202**.

[0093] If the ISP interleaver **202** of Class II in FIG. **3** is used in encoder side, the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool **410**, and an ISDP control unit in the ISDP control unit pool **406** which works with the memory pool **403**, in the reverse process of the ISP interleaver **202**.

[0094] If the ISP interleaver **202** of Class III in FIG. **3** is used in encoder side, the de-permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool **406** which works with the memory pool **403**, and a first sequence de-permuter in the first sequence de-permuter pool **408**, in the reverse process of the ISP interleaver **202**.

[0095] If the ISP interleaver **202** of Class IV in FIG. **3** is used in encoding side, the de-permutation is performed by an ISDP control unit in the ISDP control unit pool **406** which works with the memory pool **403**.

[0096] As stated above, the adder **610** and subtracter **611** can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences. For example, an adder and a subtracter are used when values in a sequence are in logarithm-based.

[0097] Finally referring to FIG. **7**, in order to control APP decoding runs by a scheduler, further steps in combination with the steps illustrated in FIG. **6** are further utilized. Essential steps of the further steps are described as follows:

[0098] Step of initialization **701**: a scheduler in the scheduler pool **402** is initialized to work on the i-th codeword sequence, and then the process goes to the step of APP decoding run **702**.

[0099] Step of APP decoding run **702**: an APP decoding run of the block **601** or **602** is performed, and then the process goes to the step of checking maximum APP decoding run **703**.

[0100] Step of checking maximum APP decoding run **703**: if a prescribed maximum number of APP decoding run has been achieved is checked; if achieved, the process goes to the step of first outputting **705**; if not achieved, the process goes to step of phasing **704**.

[0101] Step of first outputting **705**: since no more APP decoding run is available, an output result of the i-th codeword sequence is outputted if the result has not been outputted yet, and then the process goes to step of stopping **706**.

[0102] step of stopping **706**: stop the said scheduler;

[0103] Step of phasing **704**: new value of i and corresponding number of APP decoding run are calculated so that all sequences will go through all numbers of APP decoding runs, and then the process goes to the step of APP decoding run **702**.

[0104] To save time and resources, a termination test can be introduced. The test accelerates the speed to obtain a result. The test comprises the following steps:

[0105] Step of first necessity check **707**: the step **707** is performed between step **701** and step **702**. According to the decoder index table **412**, if an APP decoding run to be occurred is required is checked. The scheduler can check necessity for performing APP decoding of related sequences. If the APP decoding run to be occurred is required, the process goes to the step **702**. If the APP decoding run to be occurred is not required, the process goes to the step **703**. If step **707** exists, then the process goes to the step **707** directly instead of the step **702**.

[0106] The step of first decision making (not shown in FIG. **7**) is performed between the step of outputting first result of FIG. **6** and the step of generating first soft decoding output of FIG. **6**. Further, the first result probability measure sequence is inputted into the decision maker **603** of the scheduler pool and a first hard decoding output **613** is outputted. If the sequence outputs of ISP turbo code encoder are the sequence output of original sequence processed by the first convolutional code encoder, the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, and the sequence output of original sequence processed by the ISP interleaver, the first hard decoding output should be performed de-permutation to generate a de-permuted first hard decoding output because termination test which will be discussed below at step **708** works with sequences without permutation;

[0107] The step of second decision making (not shown in FIG. **7**) is performed between the step of third APP decoder input of FIG. **6** and the step of outputting second result of FIG. **6**. Further, the second result probability measure sequence is inputted into a decision maker **603** and outputted as a second hard decoding output **615**. If the sequence outputs of the ISP turbo code encoder are original sequence from the sequence input, sequence output of original sequence processed by the first convolutional code encoder, and sequence output original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, the second hard decoding output should be performed de-permutation to generate a de-permuted second hard decoding output because termination test which will be discussed below at step **708** works with sequences without permutation;

[0108] The decision maker **603** can be employed in odd-numbered APP decoding run block **601**, even-numbered decoding run block **602**, or both. Thus the step of first decision making and/or the step of second decision making do not need to exist in both the APP decoding runs. Further, the "de-permutation" is performed in accordance with type of ISP interleaver used in encoder side, as follows:

[0109] If the ISP interleaver of Class I in FIG. **3** is used in encoder side, as a result, the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver.

[0110] If the ISP interleaver of Class II in FIG. **3** is used in encoder side, the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, and an ISDP control unit in the ISDP control unit pool which works with the memory pool, in the reverse process of the ISP interleaver.

[0111] If the ISP interleaver of Class III in FIG. **3** is used in encoder side, the de-permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver.

[0112] If the ISP interleaver of Class IV in FIG. **3** is used in encoder side, the de-permutation is performed by an ISDP control unit in the ISDP control unit pool which works with the memory pool.

[0113] Following, the step of termination test **708** is performed between step **702** and step **703**. The termination test, which could be a conventional CRC test, is performed. That is, if the hard decoding output passes the test is checked. If the test is passed, then the process goes to the step of updating **709**. If the test is not passed or the APP decoding run block **601**, **602** does not have the hard decoding output, then the process goes to the step of checking maximum APP decoding run **703**.

[0114] The step of updating **709** updates the decoder index table **412** corresponding to pre-permutation codeword sequence according to a result of the termination test in the step **708**. Then the process goes to the step of checking maximum APP decoding run **703**.

[0115] In step **708**, if the probability measure sequence of hard decoding output **613** or **615** from the decision maker passes the test, then the probability measure sequence of hard decoding output **613** or **615**, or the de-permuted first hard decoding output or the de-permuted second hard decoding output can be directly outputted or used to calibrate the codeword sequence of soft decoding output **612** or **614**, respectively.

[0116] Further, a step of post-termination test **710** is performed between the step **709** and step **703**. A post-termination test is performed in the step **710**. That is, to check the decoder index table **412** if the post-permutation codeword sequence is required for successive APP decoding, and result thereof is used to update the decoder index table corresponding to the post-permutation codeword sequence. Then the process goes to the step of checking maximum APP decoding run **703**;

[0117] If the step of post-termination test **710** exists, then the process goes to the step **710** after step **707** when the APP decoding run to be occurred is not required. Also the process goes to the step **710** after step **709**.

[0118] In steps of termination test **708** and post-termination test **710**, the result of termination test and post-termination test can be used to release unnecessary information in the memory pool such as the codeword sequences and the probability measure sequences.

[0119] The operation illustrated in FIG. **7** take advantage of schedulers and/or termination/pos-termination tests, wherein schedulers can perform parallel processing on sequences, as illustrated in the "ring type" arrangement of schedulers for example, after initialization of the decoder, scheduler a receives first sequence and working on first APP decoding run thereof. After completion of the first APP decoding run, scheduler a passes information to scheduler b and scheduler b continues working on second APP decoding run of the first sequence, while the scheduler a receives a new sequence, and so on. We can see that schedulers are arranged to work on different sequences and different APP decoding runs in parallel automatically, which is one technical effect of the present invention. What illustrated above is one preferred operation of schedulers and of course modifications can be done by persons skilled in the art.

[0120] Further, termination tests can mark pre-permutation codeword sequences or post-permutation codeword sequences as "unnecessary to perform APP decoding" so that if all preceding sequences of a sequence to be performed are marked as "unnecessary to perform APP decoding, the APP decoding run to be performed can be skipped to save time and code sequences in the memory pool or memory index pool can be released to save resources.

[0121] Note that for ease of understanding, routine operations which are convention techniques such as memory capacity check and release are omitted in steps above. Persons skilled in the art should practice this invention with necessary modifications without departing from scope of the present invention.

1. An inter-sequence permutation (ISP) turbo code system, comprising:

an ISP turbo code encoder for receiving a sequence input and then generating a pre-permutation codeword sequence output before an ISP and a post-permutation codeword sequence output after the ISP, wherein the encoder includes:

an ISP interleaver having:

an inter-sequence permuter performs ISP and comprises at least one ISP control unit and a memory pool; an ISP algorithm is permanently embedded or temporally recorded in the at least one ISP control unit to control input to and output from the memory pool, and execution of the ISP between sequences stored in the memory pool; and

an ISP turbo code decoder for receiving the pre-permutation codeword sequence output and post-permutation codeword sequence output, wherein the decoder decodes the sequences by at least one a posteriori probability (APP) decoder therein, characterized in that decoding runs of the APP decoder are controlled by at least one scheduler and the decoding runs are performed in a do-loop manner so that the APP decoder can repeatedly be used in decoding.

2. The system as claimed in claim 1, wherein the ISP turbo code encoder comprises a first convolutional code encoder and a second convolutional code encoder, located in portions of the ISP turbo code encoder before and after the ISP interleaver, respectively.

3. The system as claimed in claim 2, wherein the ISP turbo code encoder outputs three codeword sequence outputs which are an original sequence from the sequence input, a sequence output of original sequence processed by the first convolutional code encoder, and a sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder.

4. The turbo code system as claimed in claim 2, wherein the ISP turbo code encoder outputs three codeword sequence outputs which are a sequence output of original sequence processed by the first convolutional code encoder, a sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, and a sequence output of original sequence processed by the ISP interleaver.

5. The turbo code system as claimed in claim 1, wherein the ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm, wherein the conventional sequence permuting algorithms utilized in the first sequence permuter and second sequence permuter can be different or identical, and a sequence inputted into the ISP interleaver is processed in the order of the first sequence permuter, the inter-sequence permuter, and then the second sequence permuter.

6. The turbo code system as claimed in claim 1, wherein the ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm, wherein a sequence inputted into the ISP interleaver is processed in the order of the inter-sequence permuter and then the second sequence permuter.

7. The turbo code system as claimed in claim 1, wherein the ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter, wherein a sequence inputted into the ISP interleaver is processed in the order of the first sequence permuter and then the inter-sequence permuter.

8. The turbo code system as claimed in claim 1, wherein the ISP interleaver comprises the inter-sequence permuter, wherein a sequence inputted into the ISP interleaver is processed by the inter-sequence permuter.

9. The turbo code system as claimed in claim 2, wherein the ISP turbo code encoder is further connected to a BCH or CRC encoder located between the sequence input and the ISP turbo code encoder.

10. The turbo code system as claimed in claim 5, wherein the ISP turbo code decoder comprises:

an APP decoder pool having at least one APP decoder;

a scheduler pool having at least one scheduler;

a memory pool having a plurality of memory units for storing sequences;

a memory index table for storing information on relationship between the memory units and received sequences;

an ISP control unit pool having at least one ISP control unit;

an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit;

a first sequence permuter pool having at least one first sequence permuter;

a first sequence de-permuter pool having at least one first sequence de-permuter;

a second sequence permuter pool having at least one second sequence permuter; and

a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the scheduler pool controls operations of the APP decoder pool, the ISP control unit pool, the ISDP control unit pool, the first sequence permuter pool, the first sequence de-permuter pool, the second sequence permuter pool and the second sequence de-permuter pool;

wherein the scheduler pool stores and retrieves sequences into and from the memory pool;

wherein the scheduler pool provides and retrieves sequences to and from the APP decoder pool;

wherein the scheduler pool updates and retrieves information to and from the decoder index table and memory index table;

wherein the ISP control unit pool and ISDP control unit pool interchange sequences with the memory pool;

wherein the first sequence permuter pool, the first sequence de-permuter pool, the second sequence permuter pool, and the second sequence de-permuter pool interchange sequences with the memory pool; and

wherein the scheduler pool comprises at least one adder and subtracter.

11. The turbo code system as claimed in claim 6, wherein the ISP turbo code decoder comprises:

an APP decoder pool having at least one APP decoder;

a scheduler pool having at least one scheduler;

a memory pool having a plurality of memory units for storing sequences;

a memory index table for storing information on relationship between the memory units and received sequences;

an ISP control unit pool having at least one ISP control unit;

an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit;

a second sequence permuter pool having at least one second sequence permuter; and

a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the scheduler pool controls operations of the APP decoder pool, the ISP control unit pool, the ISDP control unit pool, the second sequence permuter pool and the second sequence de-permuter pool;

wherein the scheduler pool stores and retrieves sequences into and from the memory pool;

wherein the scheduler pool provides and retrieves sequences to and from the APP decoder pool;

wherein the scheduler pool updates and retrieves information to and from the decoder index table and memory index table;

wherein the ISP control unit pool and ISDP control unit pool interchange sequences with the memory pool;

wherein the second sequence permuter pool and the second sequence de-permuter pool interchange sequences with the memory pool; and

wherein the scheduler pool comprises at least one adder and subtracter.

12. The turbo code system as claimed in claim 7, the ISP turbo code decoder comprises:

an APP decoder pool having at least one APP decoder;

a scheduler pool having at least one scheduler;

a memory pool having a plurality of memory units for storing sequences;

a memory index table for storing information on relationship between the memory units and received sequences;

an ISP control unit pool having at least one ISP control unit;

an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit;

a first sequence permuter pool having at least one first sequence permuter; and

a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the scheduler pool controls operations' of the APP decoder pool, the ISP control unit pool, the ISDP control unit pool, the first sequence permuter pool, and the first sequence de-permuter pool;

wherein the scheduler pool stores and retrieves sequences into and from the memory pool;

wherein the scheduler pool provides and retrieves sequences to and from the APP decoder pool;

wherein the scheduler pool updates and retrieves information to and from the decoder index table and memory index table;

wherein the ISP control unit pool and ISDP control unit pool interchange sequences with the memory pool;

wherein the first sequence permuter pool and first sequence de-permuter pool interchange sequences with the memory pool; and

wherein the scheduler pool comprises at least one adder and subtracter.

13. The turbo code system as claimed in claim 8, the ISP turbo code decoder comprises:

an APP decoder pool having at least one APP decoder;

a scheduler pool having at least one scheduler;

a memory pool having a plurality of memory units for storing sequences;

a memory index table storing information on relationship between the memory units and received sequences;

an ISP control unit pool having at least one ISP control unit;

an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit;

wherein the scheduler pool controls operations of the APP decoder pool, the ISP control unit pool, and the ISDP control unit pool;

wherein the scheduler pool stores and retrieves sequences into and from the memory pool;

wherein the scheduler pool provides and retrieves sequences to and from the APP decoder pool;

wherein the scheduler pool updates and retrieves information to and from the decoder index table and memory index table;

wherein the ISP control unit pool and ISDP control unit pool interchange sequences with the memory pool; and

wherein the scheduler pool comprises at least one adder and subtracter.

14. The turbo code system as claimed in claim 10, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

15. The turbo code system as claimed in claim 11, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

16. The turbo code system as claimed in claim 12, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

17. The turbo code system as claimed in claim 13, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

18. The turbo code system as claimed in claim 10, further comprises a scheduler controller connected to every scheduler in the scheduler pool if the scheduler pool is arranged in a star type.

19. The turbo code system as claimed in claim 11, further comprises a scheduler controller connected to every scheduler in the scheduler pool if the scheduler pool is arranged in a star type.

20. The turbo code system as claimed in claim 12, further comprises a scheduler controller connected to every scheduler in the scheduler pool if the scheduler pool is arranged in a star type.

21. The turbo code system as claimed in claim 13, further comprises a scheduler controller connected to every scheduler in the scheduler pool if the scheduler pool is arranged in a star type.

22. The turbo code system as claimed in claim 10, wherein the scheduler pool further comprises at least one decision maker used to output a hard decoding output sequence.

23. The turbo code system as claimed in claim 11, wherein the scheduler pool further comprises at least one decision maker used to output a hard decoding output sequence.

24. The turbo code system as claimed in claim 12, wherein the scheduler pool further comprises at least one decision maker used to output a hard decoding output sequence.

25. The turbo code system as claimed in claim 13, wherein the scheduler pool further comprises at least one decision maker used to output a hard decoding output sequence.

26. The turbo code system as claimed in claim 10, wherein the ISP turbo code decoder further comprises a decoder index table storing information on relationship between necessity to perform APP decoding and codeword sequence numbers, which connects and interchanges information with the scheduler pool.

27. The turbo code system as claimed in claim 11, wherein the ISP turbo code decoder further comprises a decoder index table storing information on relationship between necessity to perform APP decoding and codeword sequence numbers, which connects and interchanges information with the scheduler pool.

28. The turbo code system as claimed in claim 12, wherein the ISP turbo code decoder further comprises a decoder index table storing information on relationship between necessity to perform APP decoding and codeword sequence numbers, which connects and interchanges information with the scheduler pool.

29. The turbo code system as claimed in claim 13, wherein the ISP turbo code decoder further comprises a decoder index table storing information on relationship between necessity to perform APP decoding and codeword sequence numbers, which connects and interchanges information with the scheduler pool.

30. The turbo code system as claimed in claim 10, wherein the scheduler pool is further connected to at least one termination tester for performing a termination test.

31. The turbo code system as claimed in claim 11, wherein the scheduler pool is further connected to at least one termination tester for performing a termination test.

32. The turbo code system as claimed in claim 12, wherein the scheduler pool is further connected to at least one termination tester for performing a termination test.

33. The turbo code system as claimed in claim 13, wherein the scheduler pool is further connected to at least one termination tester for performing a termination test.

34-59. (canceled)

60. The turbo code system as claimed in claim 10, wherein only one of the ISP and ISDP control unit pool exists if the ISP algorithm and the ISDP algorithm are the same, and the existing control unit pool performs both ISP and ISDP.

61. The turbo code system as claimed in claim 11, wherein only one of the ISP and ISDP control unit pool exists if the ISP algorithm and the ISDP algorithm are the same, and the existing control unit pool performs both ISP and ISDP.

62. The turbo code system as claimed in claim 12, wherein only one of the ISP and ISDP control unit pool exists if the ISP algorithm and the ISDP algorithm are the same, and the existing control unit pool performs both ISP and ISDP.

63. The turbo code system as claimed in claim 13, wherein only one of the ISP and ISDP control unit pool exists if the ISP algorithm and the ISDP algorithm are the same, and the existing control unit pool performs both ISP and ISDP.

64. A method for generating probability measure sequences using the system as claimed in claim 3, wherein the ISP turbo code decoder uses codeword sequence outputs of the ISP turbo code encoder, wherein pre-permutation codeword sequence output received from the encoder, which comprises the original sequence from the sequence input and the sequence output of original sequence processed by the first convolutional code encoder, is processed in odd-numbered APP decoding runs, and post-permutation codeword sequence received from the ISP turbo code encoder, which comprises the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, is processed in even-numbered APP decoding runs,

whereby the original sequence from the sequence input is called a first codeword sequence, the sequence output of original sequence processed by a first convolutional code encoder is called a second codeword sequence, and the sequence output of original sequence processed by and in the order of the ISP interleaver and a second convolutional code encoder is called a third codeword sequence,

the method comprising the following steps for odd-numbered APP decoding run and even-numbered APP decoding run:

For the odd-numbered APP decoding run:

step of first APP decoder input: calculating a first input of the APP decoder by combining a sequence of a priori probability measure and the first codeword sequence through an adder of a scheduler pool;

step of second APP decoder input: inputting the second codeword sequence into the APP decoder as a second input;

step of outputting first result: outputting a first result probability measure sequence by the APP decoder;

step of generating first soft decoding output: calculating a first sequence of soft decoding output by eliminating the sequence of a priori probability measure from the first result probability measure sequence through a subtracter of the scheduler pool;

step of first interchange: outputting the first sequence of soft decoding output as a sequence of a priori probability measure of the subsequent even-numbered APP decoding run, wherein the even-numbered APP decoding runs work on post-permutation codeword sequences, such that permutation must be performed on the first sequence of soft decoding output before the first sequence of soft decoding output can be used in the subsequent even-numbered APP decoding run;

For the even-numbered APP decoding run:

step of third APP decoder input: receiving two inputs by an APP decoder wherein the inputs are the sequence of a priori probability measure in step of first interchange and the third codeword sequence, and outputting a second result probability measure sequence, wherein the APP decoder can be or not be the same as one used in the odd-numbered APP decoding run;

step of outputting second result: calculating a second sequence of soft decoding output by eliminating the sequence of a priori probability measure in step of first interchange or step of third APP decoder input from the second result probability measure sequence through the subtracter of the scheduler pool, wherein the subtracter can be or not be the same as one used in the odd-numbered APP decoding run;

step of second interchange: outputting the second sequence of soft decoding output as the sequence of a priori probability measure of the subsequent odd-numbered APP decoding run, wherein the odd-numbered APP decoding runs work on pre-permutation codeword sequences, de-permutation must be performed on the second sequence of soft decoding output before the second sequence of soft decoding output can be used in the subsequent odd-numbered APP decoding run.

65. A method for calculating probability measure sequences using the system as claimed in claim 4, wherein the ISP turbo code decoder uses codeword sequence outputs of the ISP turbo code encoder, wherein pre-permutation codeword sequence received from the ISP turbo code encoder, which comprises the sequence output of original sequence processed by the first convolutional code encoder, is processed in even-numbered APP decoding runs, and post-permutation codeword sequence received from the ISP turbo code encoder, which comprises the sequence output of original sequence processed by and in the order of the ISP interleaver and the second convolutional code encoder, and the sequence output of original sequence processed by the ISP interleaver, is processed in odd-numbered APP decoding runs,

whereby the sequence output of original sequence processed by the ISP interleaver is called a first codeword sequence, the sequence output of original sequence processed by and in the order of the ISP interleaver and a second convolutional code encoder is called a second

codeword sequence, and the sequence output of original sequence processed by a first convolutional code encoder is called a third codeword sequence,

the method comprising the following steps for odd-numbered APP decoding run and even-numbered APP decoding run:

For the odd-numbered APP decoding run:

step of first APP decoder input: calculating a first input of the APP decoder by combining a sequence of a priori probability measure and the first codeword sequence through an adder of a scheduler pool;

step of second APP decoder input: inputting the second codeword sequence into the APP decoder as a second input;

step of outputting first result: outputting a first result probability measure sequence by the APP decoder;

step of generating first soft decoding output: calculating a first sequence of soft decoding output by eliminating the sequence of a priori probability measure from the first result probability measure sequence through a subtracter of the scheduler pool;

step of first interchange: outputting the first sequence of soft decoding output as a sequence of a priori probability measure of the subsequent even-numbered APP decoding run, wherein the even-numbered APP decoding runs work on pre-permutation codeword sequences, such that de-permutation must be performed on the first sequence of soft decoding output before the first sequence of soft decoding output can be used in the subsequent even-numbered APP decoding run;

For the even-numbered APP decoding run:

step of third APP decoder input: receiving two inputs by an APP decoder wherein the inputs are the sequence of a priori probability measure in step of first interchange and the third codeword sequence, and outputting a second result probability measure sequence, wherein the APP decoder can be or not be the same as one used in the odd-numbered APP decoding run;

step of outputting second result: calculating a second sequence of soft decoding output by eliminating the sequence of a priori probability measure in step of first interchange or step of third APP decoder input from the second result probability measure sequence through the subtracter of the scheduler pool, wherein the subtracter can be or not be the same as one used in the odd-numbered APP decoding run;

step of second interchange: outputting the second sequence of soft decoding output as the sequence of a priori probability measure of the subsequent odd-numbered APP decoding run, wherein the odd-numbered APP decoding runs work on post-permutation codeword sequences, permutation must be performed on the second sequence of soft decoding output before the second sequence of soft decoding output can be used in the subsequent odd-numbered APP decoding run.

66. The method as claimed in claim 64, wherein the "permutation" performed in the step of first interchange is performed according to the ISP interleaver used in the ISP turbo code encoder, wherein the permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver recited is used in encoder side, a first ISP turbo code decoder is employed, and the permutation is performed by and in the order of a first sequence permuter in the first sequence permuter pool, an ISP control unit in the ISP control unit pool which works with the memory pool, and a second sequence permuter in the second sequence permuter pool, in the process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the permutation is performed by and in the order of an ISP control unit working in the ISP control unit pool which works with the memory pool, and a second sequence permuter in the second sequence permuter pool, in the process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the permutation is performed by and in the order of a first sequence permuter in the first sequence permuter pool, and an ISP control unit in the ISP control unit pool which works with the memory pool, in the process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the permutation is performed by an ISP control unit in the ISP control unit pool which works with the memory pool;

wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permu-

tation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

67. The method as claimed in claim 65, wherein the "permutation" performed in the step of second interchange is performed according to the ISP interleaver used in the ISP turbo code encoder, wherein the permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver is used in encoder side, a first ISP turbo code decoder is employed, and the permutation is performed by and in the order of a first sequence permuter in the first sequence permuter pool, an ISP control unit in the ISP control unit pool which works with the memory pool, and a second sequence permuter in the second sequence permuter pool, in the process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the permutation is performed by and in the order of an ISP control unit working in the ISP control unit pool which works with the memory pool, and a second sequence permuter in the second sequence permuter pool, in the process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the permutation is performed by and in the order of a first sequence permuter in the first sequence permuter pool, and an ISP control unit in the ISP control unit pool which works with the memory pool, in the process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the permutation is performed by an ISP control unit in the ISP control unit pool which works with the memory pool;

wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a

scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

68. The method as claimed in claim 64, wherein the "de-permutation" performed in the step of second interchange is performed according to ISP interleaver used in the ISP turbo code encoder, wherein the de-permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver is used in encoder side, a first ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, and an ISDP control unit in the ISDP control unit pool which works with the memory pool, in the reverse process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the de-permutation is performed by an ISDP control unit in the ISDP control unit pool which works with the memory pool.

wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

69. The method as claimed in claim 65, wherein the "de-permutation" performed in the step of first interchange is performed according to ISP interleaver used in the ISP turbo code encoder, wherein the de-permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver is used in encoder side, a first ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a

second sequence de-permuter in the second sequence de-permuter pool, and an ISDP control unit in the ISDP control unit pool which works with the memory pool, in the reverse process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the de-permutation is performed by an ISDP control unit in the ISDP control unit pool which works with the memory pool.

wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

70. The method as claimed in claim 64, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

71. The method as claimed in claim 65, wherein the adder and subtracter can be replaced by a multiplier and a divider respectively in accordance with scale or format of the sequences.

72. The method as claimed in claim 64, further comprising the following steps of scheduling:

step of initialization: initializing a scheduler in a scheduler pool to work on the i-th codeword sequence; going to step of APP decoding run;

step of APP decoding run: performing an APP decoding run according to number thereof;

going to step of checking maximum APP decoding run;

step of checking maximum APP decoding run: checking if a prescribed maximum number of APP decoding run has been achieved; if achieved, going to step of first outputting; if not achieved, going to step of phasing;

step of first outputting: outputting result of the i-th codeword sequence;

step of stopping: stopping the scheduler;

step of phasing: selecting a new value of i and corresponding number of APP decoding run; going to the step of APP decoding run.

73. The method as claimed in claim 65, further comprising the following steps of scheduling:

step of initialization: initializing a scheduler in a scheduler pool to work on the i-th codeword sequence; going to step of APP decoding run;

step of APP decoding run: performing an APP decoding run according to number thereof;

going to step of checking maximum APP decoding run;

step of checking maximum APP decoding run: checking if a prescribed maximum number of APP decoding run

has been achieved; if achieved, going to step of first outputting; if not achieved, going to step of phasing;

step of first outputting: outputting result of the i-th codeword sequence;

step of stopping: stopping the scheduler;

step of phasing: selecting a new value of i and corresponding number of APP decoding run; going to the step of APP decoding run.

74. The method as claimed in claim 72, further comprising the following steps:

step of first necessity check, which is between the step of initialization and the step of APP decoding run: according to a decoder index table, checking if an APP decoding run to be occurred is required; if required, going to the step of APP decoding run; if not required, going to the step of checking maximum APP decoding run;

if the step of first necessity check exists, then the step of phasing is directed to the step of first necessity check directly instead of the step of APP decoding run;

step of first decision making, which is between the step of outputting first result and the step of generating first soft decoding output further: inputting the first result probability measure sequence into a decision maker of the scheduler pool and outputting a first hard decoding output;

step of termination test, which is between the step of APP decoding run and the step of checking maximum APP decoding run: performing a termination test, which could be a conventional CRC test, to check if the first hard decoding output passes the test; if the test is passed, then going to a step of updating; if the test is not passed or the APP decoding run just performed does not have the hard decoding output, then going to the step of checking maximum APP decoding run;

step of updating: updating a decoder index table corresponding to the pre-permutation codeword sequence according to result of the first termination test; then going to the step of checking maximum APP decoding run.

75. The method as claimed in claim 73, further comprising the following steps:

step of first necessity check, which is between the step of initialization and the step of APP decoding run: according to a decoder index table, checking if an APP decoding run to be occurred is required; if required, going to the step of APP decoding run; if not required, going to the step of checking maximum APP decoding run;

if the step of first necessity check exists, then the step of phasing is directed to the step of first necessity check directly instead of the step of APP decoding run;

step of first decision making, which is between the step of outputting first result and the step of generating first soft decoding output further: inputting the first result probability measure sequence into a decision maker of the scheduler pool and outputting a first hard decoding output; the first hard decoding output should be performed de-permutation to generate a de-permuted first hard decoding output;

step of termination test, which is between the step of APP decoding run and the step of checking maximum APP decoding run: performing a termination test, which could be a conventional CRC test, to check if the de-permuted first hard decoding output passes the test; if the test is passed, then going to a step of updating; if the test is not passed or the APP decoding run just performed does not have the de-permuted hard decoding output, then going to the step of checking maximum APP decoding run;

step of updating: updating a decoder index table corresponding to the pre-permutation codeword sequence according to result of the first termination test; then going to the step of checking maximum APP decoding run.

76. The method as claimed in claim 74, further comprising the following steps:

step of post-termination test, which exists between the step of updating and the step of checking maximum APP decoding run: performing a post-termination test to check with the decoder index table if the post-permutation codeword sequence is required for successive APP decoding; a result of the test is used to update the decoder index table corresponding to the post-permutation codeword sequence or to release unnecessary post-permutation codeword sequence in the memory pool;

if the step of post-termination test exists, then directing "not required" output of the step of first necessity check and output of the step of updating to the step of post-termination test.

77. The method as claimed in claim 75, further comprising the following steps:

step of post-termination test, which exists between the step of updating and the step of checking maximum APP decoding run: performing a post-termination test to check with the decoder index table if the post-permutation codeword sequence is required for successive APP decoding; a result of the test is used to update the decoder index table corresponding to the post-permutation codeword sequence or to release unnecessary post-permutation codeword sequence in the memory pool;

if the step of post-termination test exists, then directing "not required" output of the step of first necessity check and output of the step of updating to the step of post-termination test.

78. The method as claimed in claim 74, further comprising the following step:

in the step of termination test, if the first hard decoding output from the decision maker passes the test, then outputting or calibrating the sequence of soft decoding output by using the first hard decoding output, or releasing unnecessary pre-permutation codeword sequence in the memory pool.

79. The method as claimed in claim 75, further comprising the following step:

in the step of termination test, if the de-permuted first hard decoding output from the decision maker passes the test, then outputting or calibrating the sequence of soft decoding output by using the de-permuted first hard decoding output, or releasing unnecessary pre-permutation codeword sequence in the memory pool.

80. The method as claimed in claim 72, further comprising the following steps:

step of first necessity check, which exists between the step of initialization and the step of APP decoding run: according to a decoder index table, checking if an APP decoding run to be occurred is required; if required, going to the step of APP decoding run; if not required, going to the step of checking maximum APP decoding run;

if the step of first necessity check exists, then going to the step of first necessity check directly instead of the step of APP decoding run after the step of phasing;

step of second decision making, which exists between the step of third APP decoder input and the step of outputting second result: inputting the second result probability measure sequence to a decision maker of the scheduler pool and outputting a second hard decoding output; the second hard decoding output is performed de-permutation to generate a de-permuted second hard decoding output;

step of termination test, which exists between the step of APP decoding run and the step of checking maximum APP decoding run: performing a termination test, which could be a conventional CRC test, to check if the de-permuted second hard decoding output passes the test; if the test is passed, then going to a step of updating; if the test is not passed or the APP decoding run just performed does not have the de-permuted second hard decoding output, then going to the step of checking maximum APP decoding run;

step of updating: updating the decoder index table corresponding to the pre-permutation codeword sequence according to results of the termination test; then going to the step of checking maximum APP decoding run.

81. The method as claimed in claim 73, further comprising the following steps:

step of first necessity check, which exists between the step of initialization and the step of APP decoding run: according to a decoder index table, checking if an APP decoding run to be occurred is required; if required, going to the step of APP decoding run; if not required, going to the step of checking maximum APP decoding run;

if the step of first necessity check exists, then going to the step of first necessity check directly instead of the step of APP decoding run after the step of phasing;

step of second decision making, which exists between the step of third APP decoder input and the step of outputting second result: inputting the second result probability measure sequence to a decision maker of the scheduler pool and outputting a second hard decoding output;

step of termination test, which exists between the step of APP decoding run and the step of checking maximum APP decoding run: performing a termination test, which could be a conventional CRC test, to check if the second hard decoding output passes the test; if the test is passed, then going to a step of updating; if the test is not passed or the APP decoding run just performed does not have the second hard decoding output, then going to the step of checking maximum APP decoding run;

step of updating: updating the decoder index table corresponding to the pre-permutation codeword sequence according to results of the termination test; then going to the step of checking maximum APP decoding run.

82. The method as claimed in claim 80, further comprising the following steps:

step of post-termination test, which exists between the step of updating and the step of checking maximum APP decoding run: performing a post-termination test to check with the decoder index table if the post-permutation codeword sequence is required for successive APP decoding; result of the test is used to update the decoder index table corresponding to the post-permutation codeword sequence or to release unnecessary post-permutation codeword sequence in the memory pool;

if the step of post-termination test exists, then directing "not required" output of the step of first necessity check and output of the step of updating to the step of post-termination test.

83. The method as claimed in claim 81, further comprising the following steps:

step of post-termination test, which exists between the step of updating and the step of checking maximum APP decoding run: performing a post-termination test to check with the decoder index table if the post-permutation codeword sequence is required for successive APP decoding; result of the test is used to update the decoder index table corresponding to the post-permutation codeword sequence or to release unnecessary post-permutation codeword sequence in the memory pool;

if the step of post-termination test exists, then directing "not required" output of the step of first necessity check and output of the step of updating to the step of post-termination test.

84. The method as claimed in claim 80, further comprising the following step:

in the step of termination test, if the de-permuted second hard decoding output corresponding to the pre-permutation codeword sequence from the decision maker passes the test, then outputting or directing the sequence of soft decoding output by using the de-permuted second hard decoding output, or releasing unnecessary pre-permutation codeword sequence in the memory pool.

85. The method as claimed in claim 81, further comprising the following step:

in the step of termination test, if the second hard decoding output corresponding to the pre-permutation codeword sequence from the decision maker passes the test, then outputting or directing the sequence of soft decoding output by using the second hard decoding output, or releasing unnecessary pre-permutation codeword sequence from the memory pool.

86. The method as claimed in claim 75, wherein the "de-permutation" performed in step of first decision making is performed according to ISP interleaver used in the ISP turbo code encoder; the de-permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver is used in encoder side, a first ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, and an ISDP control unit in the ISDP control unit pool which works with the memory pool, in the reverse process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the de-permutation is performed by an ISDP control unit in the ISDP control unit pool which works with the memory pool.

wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

87. The method as claimed in claim 80, wherein the "de-permutation" performed in step of second decision making is performed according to ISP interleaver used in the ISP turbo code encoder; the de-permutation can be performed in accordance with one of the following cases:

if a first ISP interleaver is used in encoder side, a first ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver;

if a second ISP interleaver is used in encoder side, a second ISP turbo code decoder is employed, and the de-permutation is performed by and in the order of a second sequence de-permuter in the second sequence de-permuter pool, and an ISDP control unit in the ISDP control unit pool which works with the memory pool, in the reverse process of the ISP interleaver;

if a third ISP interleaver is used in encoder side, a third ISP turbo code decoder is employed, and the de-

permutation is performed by and in the order of an ISDP control unit in the ISDP control unit pool which works with the memory pool, and a first sequence de-permuter in the first sequence de-permuter pool, in the reverse process of the ISP interleaver; and

if a fourth ISP interleaver is used in encoder side, a fourth ISP turbo code decoder is employed, and the de-permutation is performed by an ISDP control unit in the ISDP control unit pool which works with the memory pool, wherein the first ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm, the inter-sequence permuter, and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the first ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, a first sequence de-permuter pool having at least one first sequence de-permuter, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the second ISP interleaver comprises the inter-sequence permuter and a second sequence permuter utilizing a conventional sequence permuting algorithm;

wherein the second ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a second sequence permuter pool having at least one second sequence permuter, and a second sequence de-permuter pool having at least one second sequence de-permuter;

wherein the third ISP interleaver comprises a first sequence permuter utilizing a conventional sequence permuting algorithm and the inter-sequence permuter;

wherein the third ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table for storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit, a first sequence permuter pool having at least one first sequence permuter, and a first sequence de-permuter pool having at least one first sequence de-permuter;

wherein the fourth ISP interleaver comprises the inter-sequence permuter;

wherein the fourth ISP turbo code decoder comprises an APP decoder pool having at least one APP decoder, a scheduler pool having at least one scheduler, a memory pool having a plurality of memory units for storing sequences, a memory index table storing information on relationship between the memory units and received sequences, an ISP control unit pool having at least one ISP control unit, an inter-sequence de-permutation (ISDP) control unit pool having at least one ISDP control unit.

* * * * *