US00RE46527E

(19) **United States**

(12) **Reissued Patent**

Song et al.

(10) **Patent Number: US RE46,527 E**

(45) **Date of Reissued Patent: Aug. 29, 2017**

(54) **TRANSMITTING/RECEIVING SYSTEM AND METHOD OF PROCESSING BROADCAST SIGNAL IN TRANSMITTING/RECEIVING SYSTEM**

(71) Applicant: **LG ELECTRONICS INC.**, Seoul (KR)

(72) Inventors: **Won Gyu Song**, Seoul (KR); **Byoung Gill Kim**, Seoul (KR); **Hyoung-Gon Lee**, Seoul (KR); **In Hwan Choi**, Seoul (KR); **Jin Woo Kim**, Seoul (KR)

(73) Assignee: **LG ELECTRONICS INC.**, Seoul (KR)

(21) Appl. No.: **14/548,049**

(22) Filed: **Nov. 19, 2014**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **8,503,347**
Issued: **Aug. 6, 2013**
Appl. No.: **12/751,974**
Filed: **Mar. 31, 2010**

U.S. Applications:

(66) Substitute for application No. 12/751,974, filed on Mar. 31, 2010, now Pat. No. 8,503,347, which is an (Continued)

(51) **Int. Cl.**

H04H 20/71 (2008.01)
H03C 1/52 (2006.01)
H04L 1/00 (2006.01)

(52) **U.S. Cl.**

CPC ***H04L 1/0071*** (2013.01); ***H04L 1/005*** (2013.01); ***H04L 1/007*** (2013.01); ***H04L 1/0065*** (2013.01);
(Continued)

(58) **Field of Classification Search**

CPC H04L 1/005; H04L 1/0065; H04L 1/007; H04L 1/0071; H04L 1/0072; H04L 1/0057; H04L 1/006
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,781,975 B1 * 8/2004 Baier et al. 370/335
6,868,093 B1 * 3/2005 Bohm et al. 370/506
(Continued)

FOREIGN PATENT DOCUMENTS

KR 2009-0021106 2/2009

OTHER PUBLICATIONS

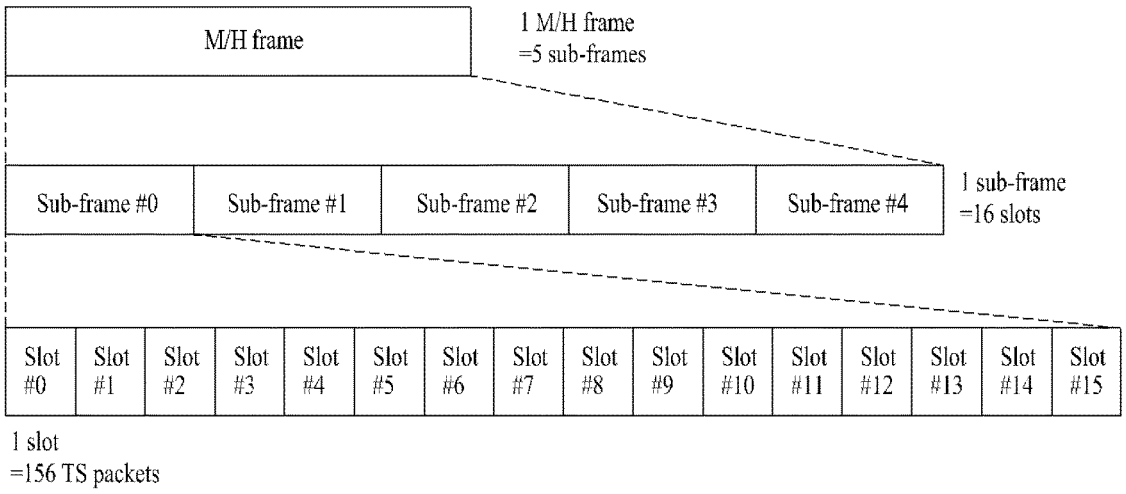
A Coding and Modulation Scheme for HDTV Services in DMB-T; Mar. 2004, IEEE.
(Continued)

Primary Examiner — Ovidio Escalante
(74) *Attorney, Agent, or Firm* — Dentons US LLP

(57) **ABSTRACT**

A transmitting system, a receiving system, and a method for processing a broadcast signal are disclosed. The receiving system comprises a tuner, a channel equalizer, a turbo decoder, a demultiplexer, a first error correction decoder, a block deinterleaver, and a second error correction decoder. The tuner receives a broadcast signal including a data group. The data group comprises mobile service data, regularly spaced known data sequences, and signaling data. The turbo decoder performs turbo decoding for the signaling data included in the channel equalized broadcast signal in the channel equalizer. The block deinterleaver performs block deinterleaving for the turbo-decoded FIC data in a block unit of TNoG (the number of all data groups assigned to one subframe)×51 bytes.

10 Claims, 51 Drawing Sheets



Related U.S. Application Data

application for the reissue of Pat. No. Re. 45,834, which is an application for the reissue of Pat. No. 8,503,347.

(60) Provisional application No. 61/165,500, filed on Mar. 31, 2009.

(52) **U.S. Cl.**
CPC ***H04L 1/0072*** (2013.01); *H04L 1/006* (2013.01); *H04L 1/0057* (2013.01)

(58) **Field of Classification Search**
USPC 370/312, 201, 278, 286, 328, 329, 350, 370/536, 542; 375/270, 229, 232, 234, 375/262, 265, 267, 277, 316, 321, 326, 375/340, 246
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,394,987	B2 *	7/2008	Hong et al.	398/106
8,116,400	B2 *	2/2012	Limberg	375/270
8,135,034	B2	3/2012	Song et al.	
8,762,816	B2	6/2014	Choi et al.	
2002/0023246	A1 *	2/2002	Jin	714/755
2007/0087687	A1	4/2007	Park et al.	
2007/0094567	A1 *	4/2007	Park et al.	714/753
2007/0121748	A1 *	5/2007	Park et al.	375/265
2008/0134007	A1 *	6/2008	Lee	H04L 1/0041 714/776
2008/0253468	A1 *	10/2008	Yu	375/240.28
2009/0013356	A1 *	1/2009	Doerr et al.	725/62
2009/0034556	A1 *	2/2009	Song et al.	370/471
2009/0067561	A1 *	3/2009	Kuppens	375/362
2009/0083606	A1	3/2009	Choi et al.	
2009/0265751	A1	10/2009	Limberg	
2009/0323823	A1	12/2009	Limberg	
2010/0037120	A1 *	2/2010	Limberg	714/758
2010/0054376	A1 *	3/2010	Park et al.	375/341
2010/0061491	A1 *	3/2010	Choi et al.	375/341
2010/0100793	A1	4/2010	Limberg	
2010/0149427	A1	6/2010	Limberg	
2010/0195712	A1 *	8/2010	Yu et al.	375/240.01
2010/0231803	A1 *	9/2010	Citta et al.	348/723
2010/0232550	A1	9/2010	Lee et al.	
2010/0254494	A1	10/2010	Song et al.	
2011/0047434	A1	2/2011	Walker et al.	

OTHER PUBLICATIONS

Hardware Implementation of DVB-RCTT Modulator; Feb. 2, 2006.
Construction of a digital-TV receiver for the second-generation satellite broadcasting, DVB-S2; Jun. 20, 2007.
Performance Analysis of Different Reen-Solomon Erasure Decoding Strategies at the DVB-H Link Layer; 2006 IEEE.
Portable Receivers for Digital Multimedia Broadcasting; May 2004, IEEE.
Mobile Multimedia Broadcasting Standards; Aug. 2008.

* cited by examiner

FIG. 1

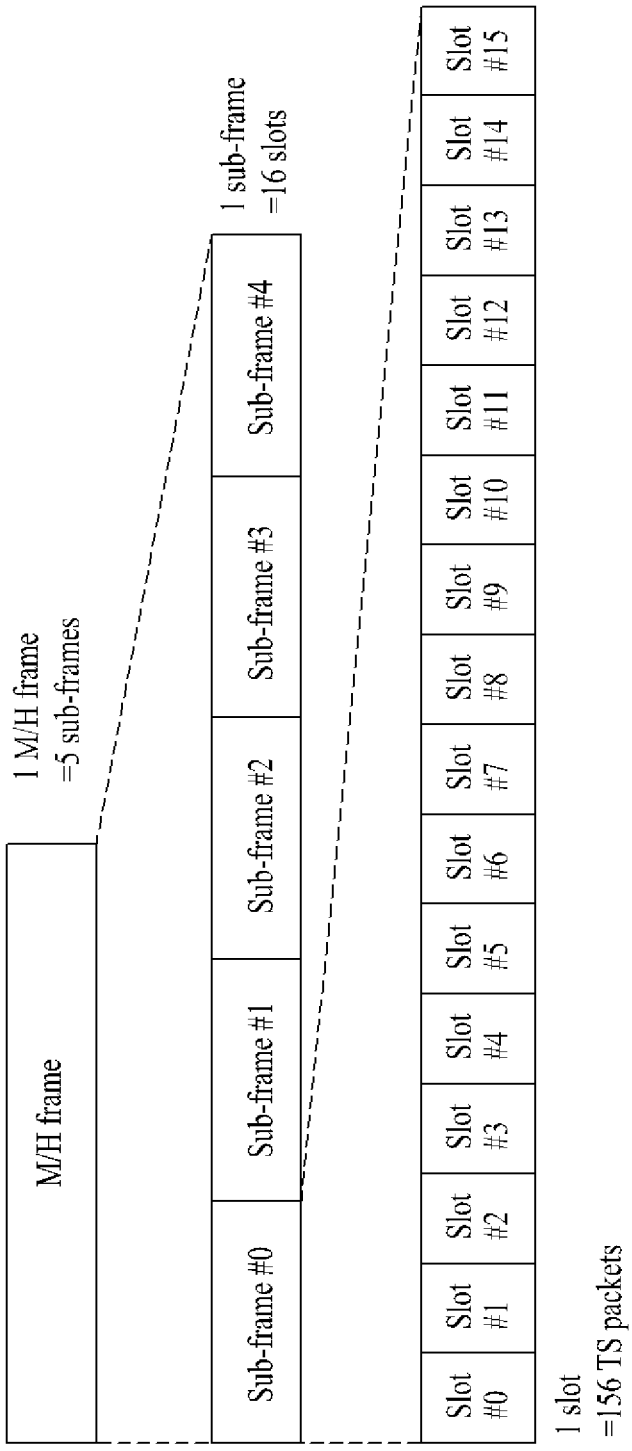

M/H frame
1 M/H frame
=5 sub-frames
Sub-frame #0
Sub-frame #1
Sub-frame #2
Sub-frame #3
Sub-frame #4
1 sub-frame
=16 slots
Slot #0
Slot #1
Slot #2
Slot #3
Slot #4
Slot #5
Slot #6
Slot #7
Slot #8
Slot #9
Slot #10
Slot #11
Slot #12
Slot #13
Slot #14
Slot #15
1 slot
=156 TS packets

FIG. 2

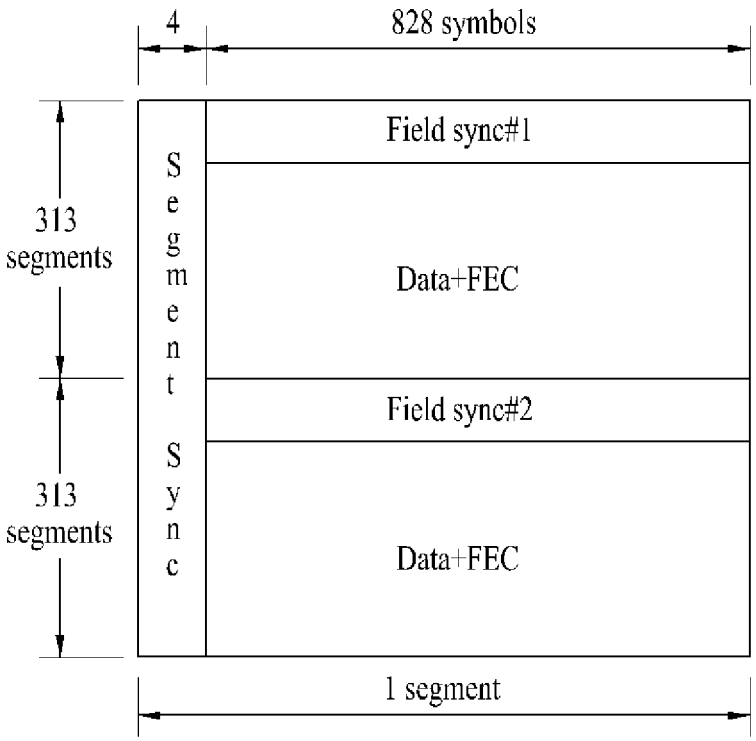

4
828 symbols
313 segments
313 segments
Segment Sync
Field sync#1
Data+FEC
Field sync#2
Data+FEC
1 segment

FIG. 3

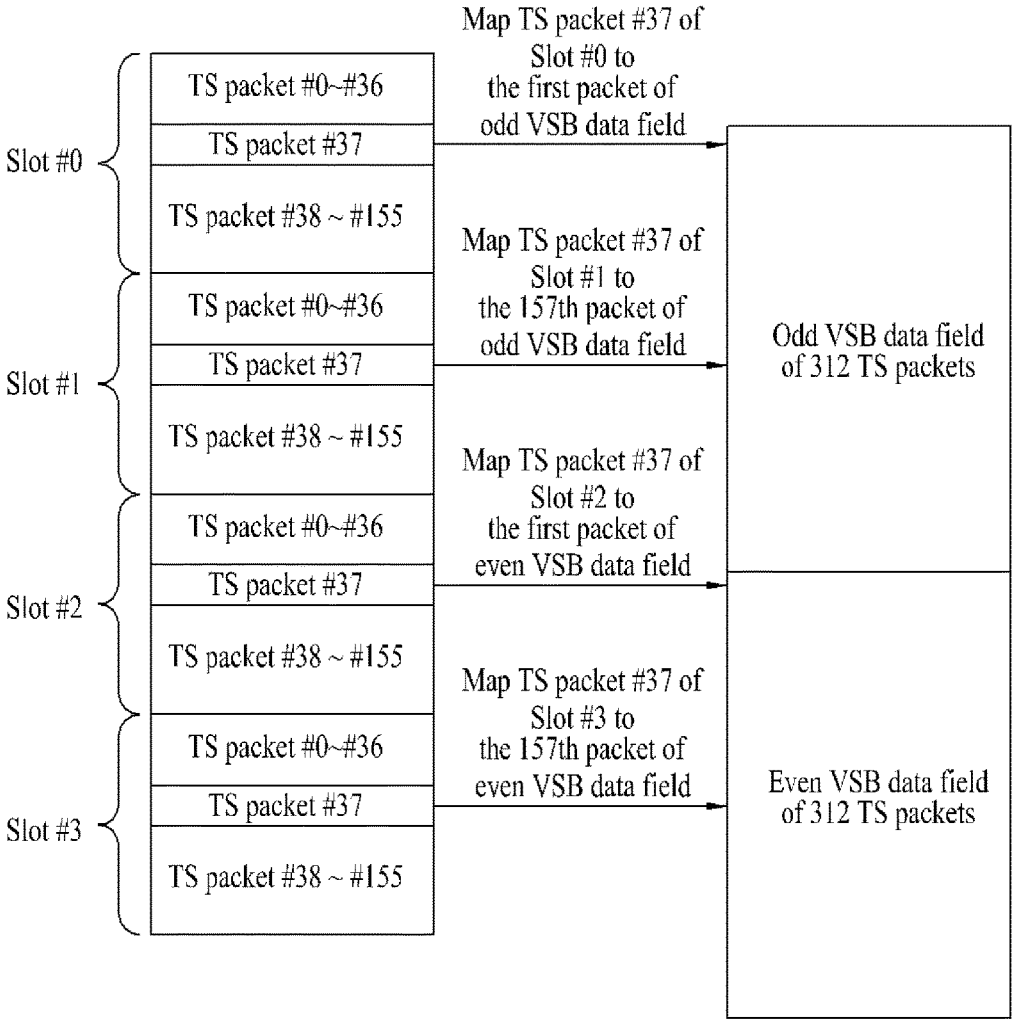

Slot #0
Slot #1
Slot #2
Slot #3
TS packet #0~#36
TS packet #37
TS packet #38 ~ #155
Map TS packet #37 of Slot #0 to the first packet of odd VSB data field
Map TS packet #37 of Slot #1 to the 157th packet of odd VSB data field
Map TS packet #37 of Slot #2 to the first packet of even VSB data field
Map TS packet #37 of Slot #3 to the 157th packet of even VSB data field
Odd VSB data field of 312 TS packets
Even VSB data field of 312 TS packets

FIG. 4

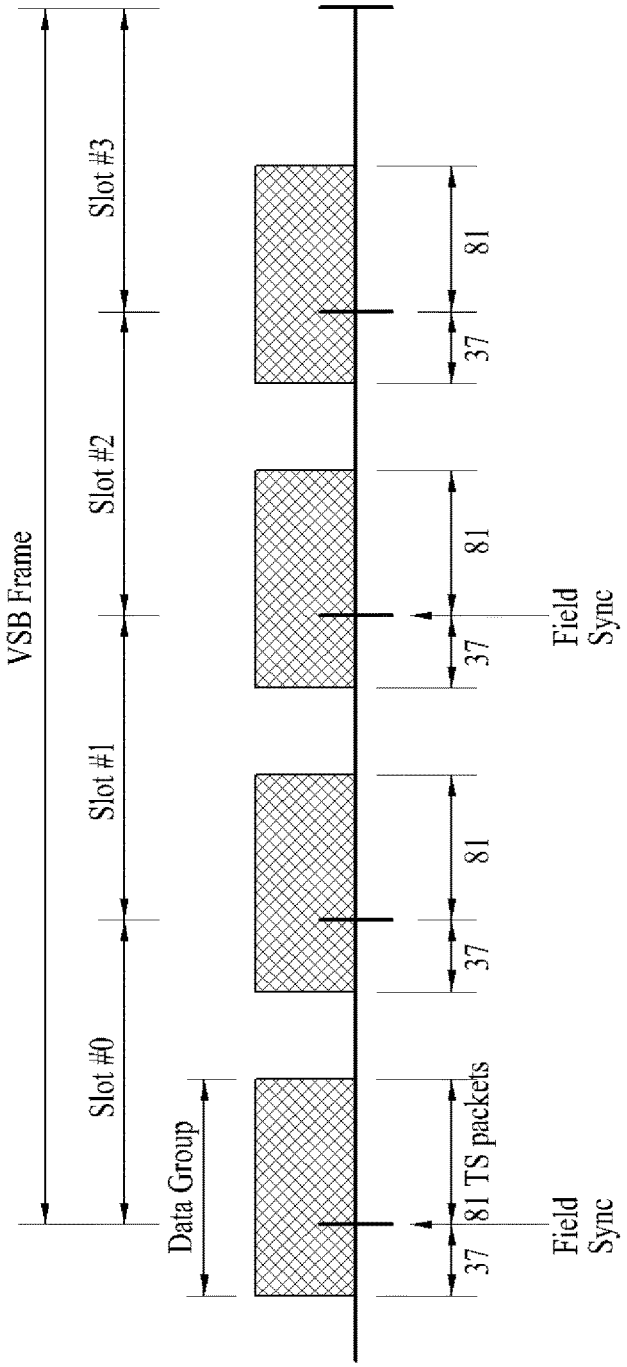

VSB Frame
Slot #3
Slot #2
Slot #1
Slot #0
Data Group
37
81 TS packets
81
37
81
37
81
37
Field Sync
Field Sync

FIG. 5

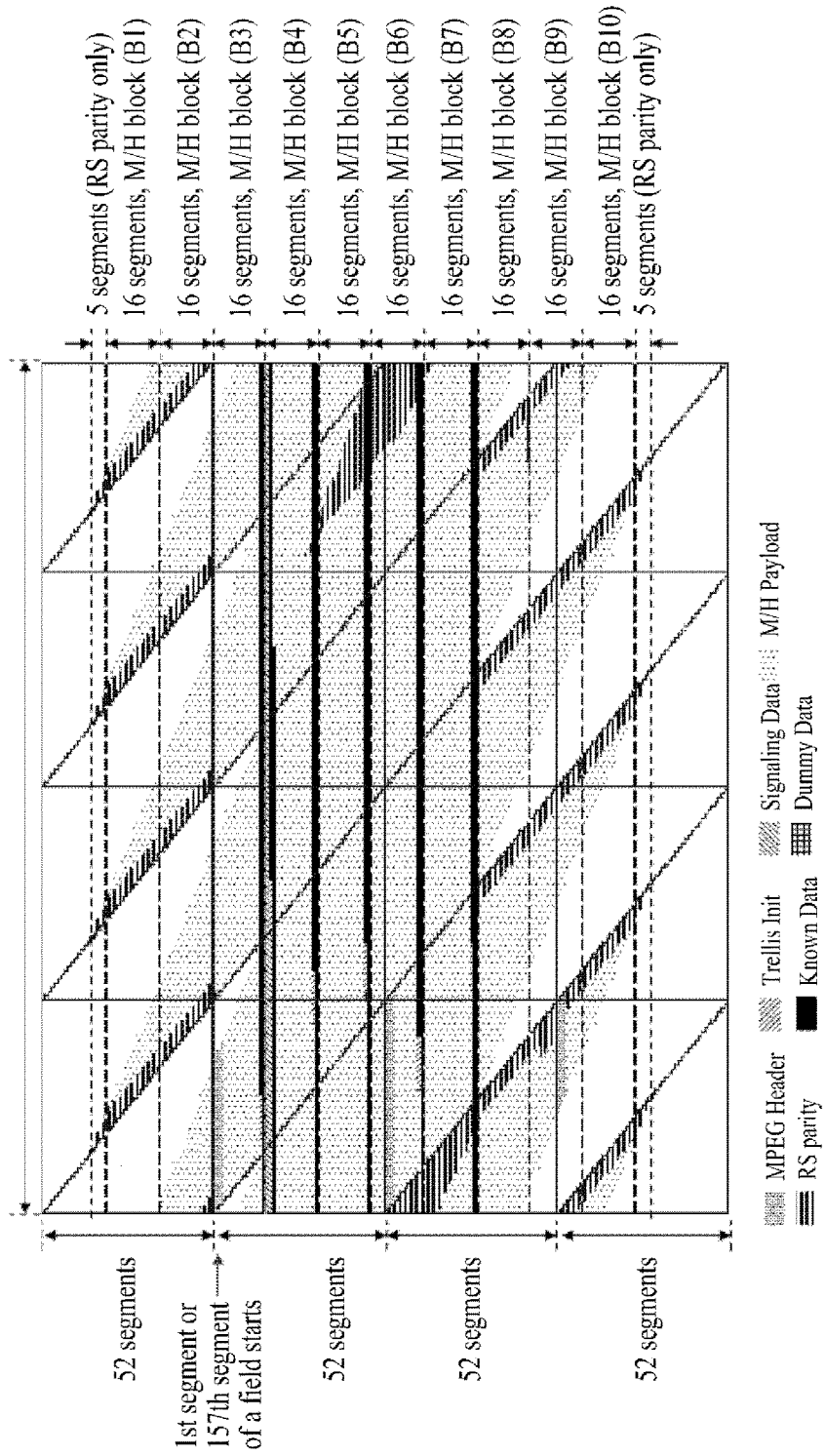

5 segments (RS parity only)
16 segments, M/H block (B1)
16 segments, M/H block (B2)
16 segments, M/H block (B3)
16 segments, M/H block (B4)
16 segments, M/H block (B5)
16 segments, M/H block (B6)
16 segments, M/H block (B7)
16 segments, M/H block (B8)
16 segments, M/H block (B9)
16 segments, M/H block (B10)
5 segments (RS parity only)
52 segments
1st segment or 157th segment of a field starts
52 segments
52 segments
52 segments
MPEG Header
RS parity
Trellis Init
Known Data
Signaling Data
Dummy Data
M/H Payload

FIG. 6

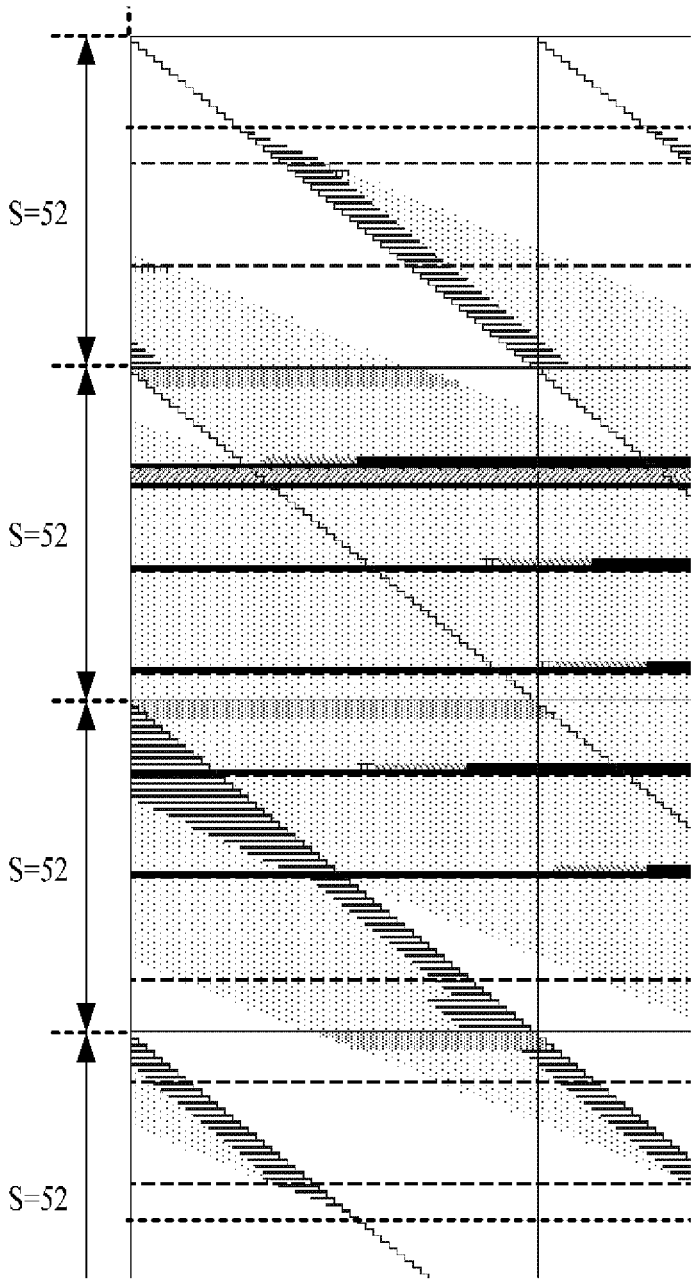


FIG. 7

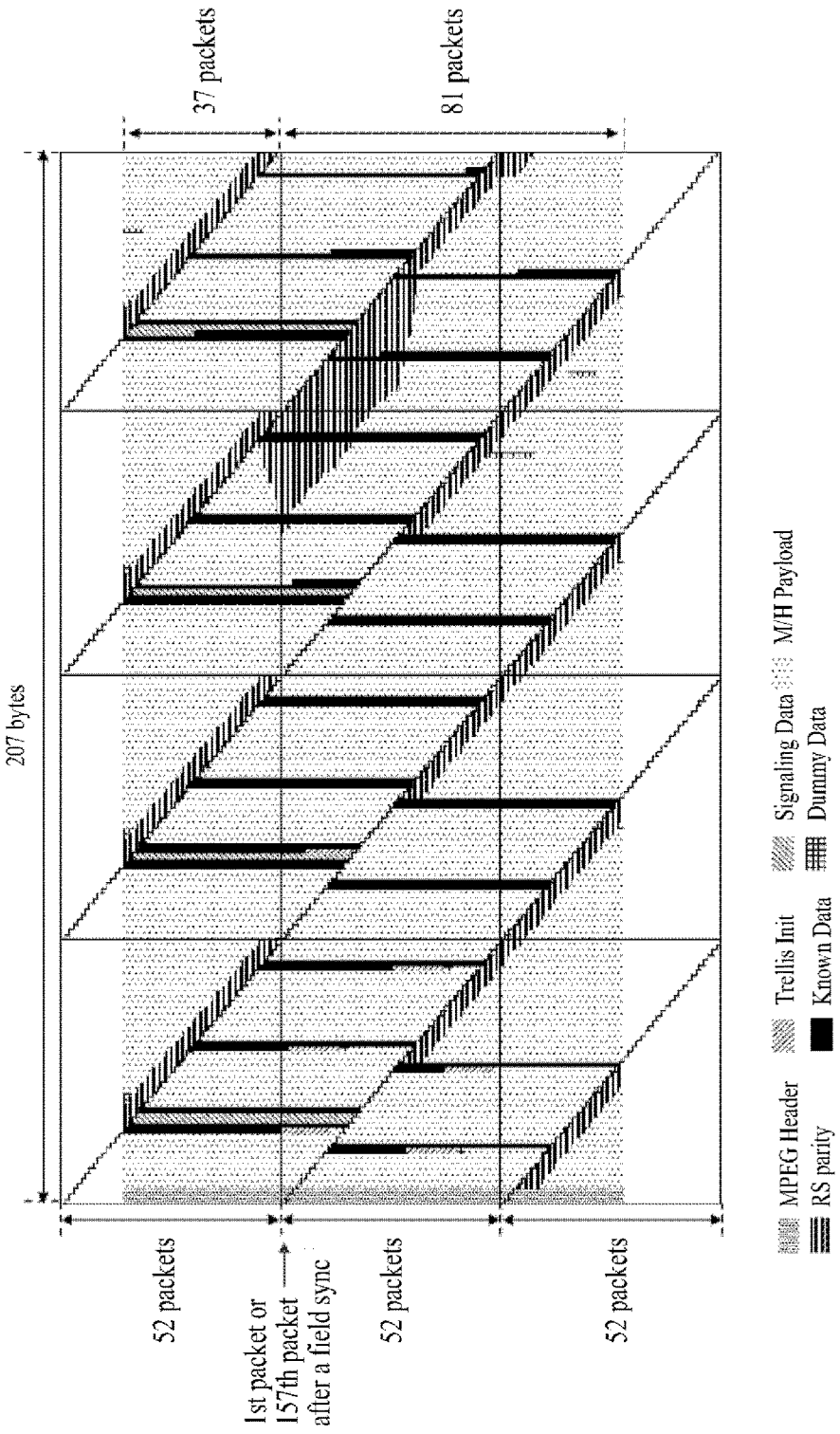

37 packets
81 packets
207 bytes
52 packets
1st packet or
157th packet
after a field sync
52 packets
52 packets
MPEG Header
RS parity
Trellis Init
Known Data
Signaling Data
Dummy Data
M/H Payload

FIG. 8

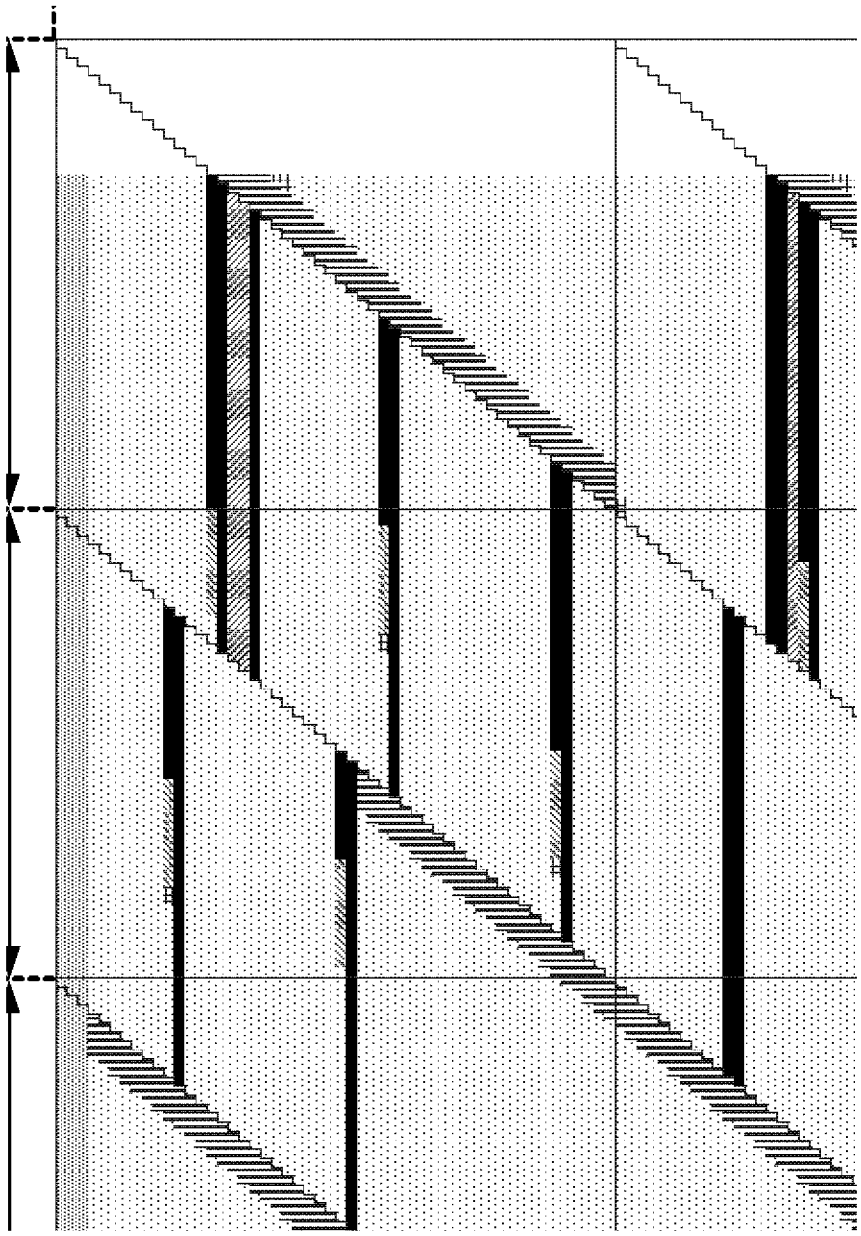


FIG. 9

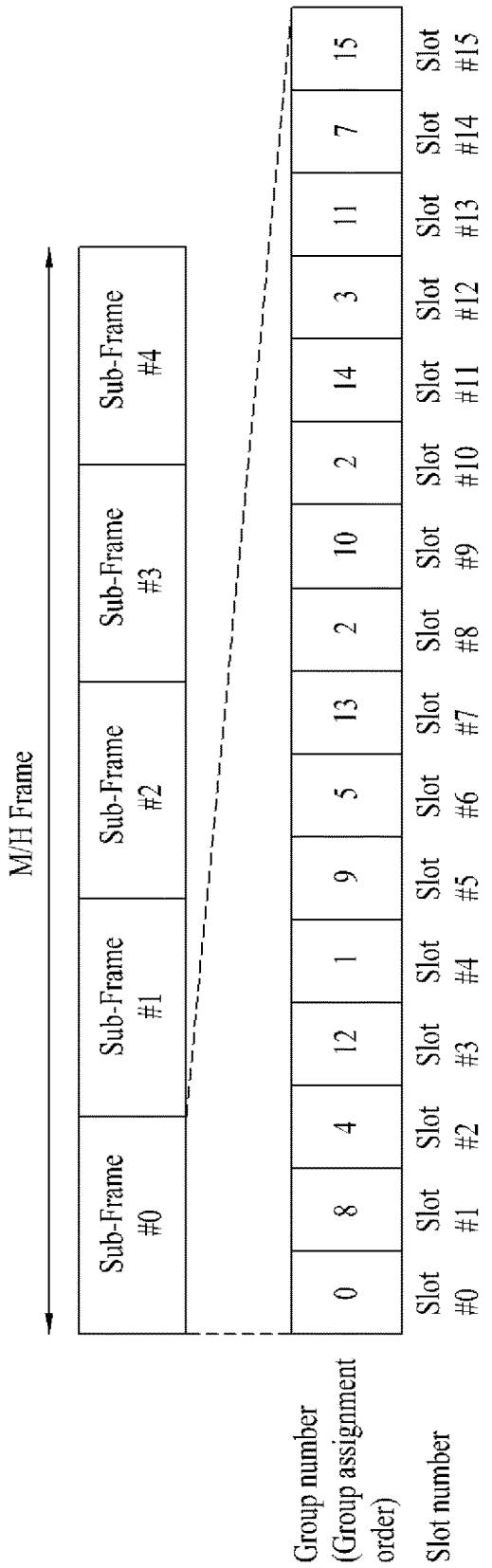

M/H Frame
Sub-Frame #0
Sub-Frame #1
Sub-Frame #2
Sub-Frame #3
Sub-Frame #4
Group number (Group assignment order)
Slot number
0 8 4 12 1 9 5 13 2 10 2 14 3 11 7 15
Slot #0 Slot #1 Slot #2 Slot #3 Slot #4 Slot #5 Slot #6 Slot #7 Slot #8 Slot #9 Slot #10 Slot #11 Slot #12 Slot #13 Slot #14 Slot #15

FIG. 10

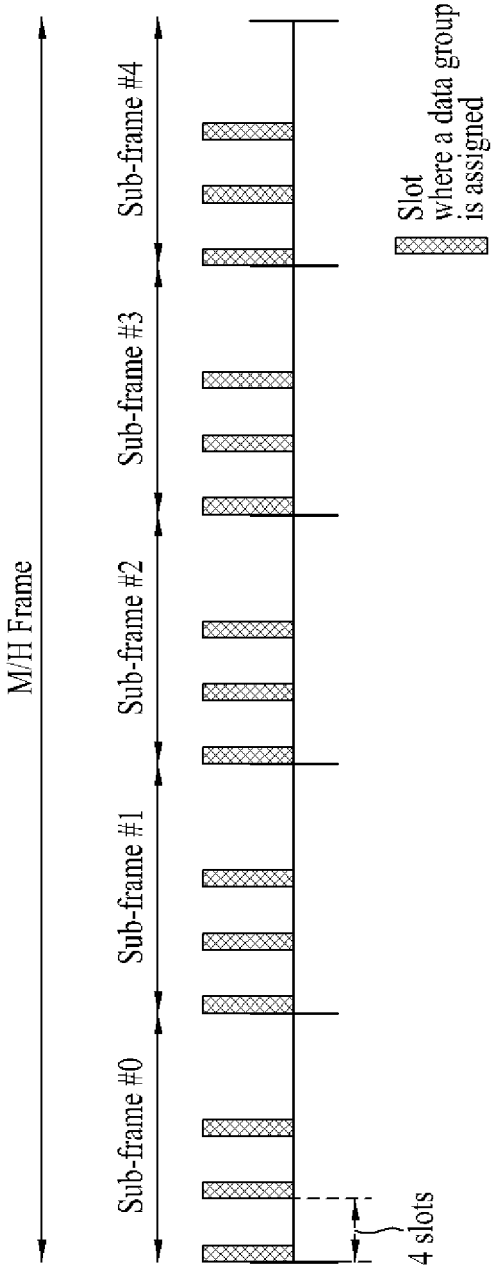

M/H Frame
Sub-frame #0
Sub-frame #1
Sub-frame #2
Sub-frame #3
Sub-frame #4
4 slots
Slot
where a data group
is assigned

FIG. 11

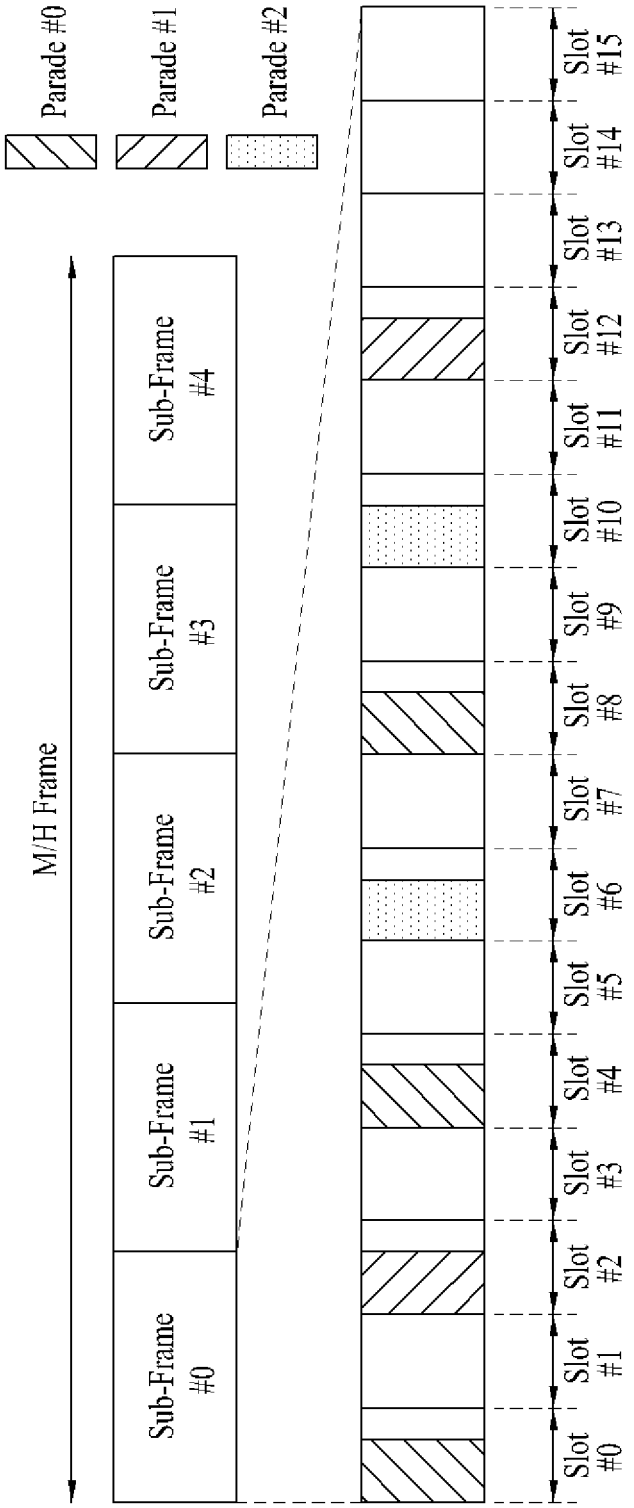

Parade #0
Parade #1
Parade #2
M/H Frame
Sub-Frame #0
Sub-Frame #1
Sub-Frame #2
Sub-Frame #3
Sub-Frame #4
Slot #0
Slot #1
Slot #2
Slot #3
Slot #4
Slot #5
Slot #6
Slot #7
Slot #8
Slot #9
Slot #10
Slot #11
Slot #12
Slot #13
Slot #14
Slot #15

FIG. 12

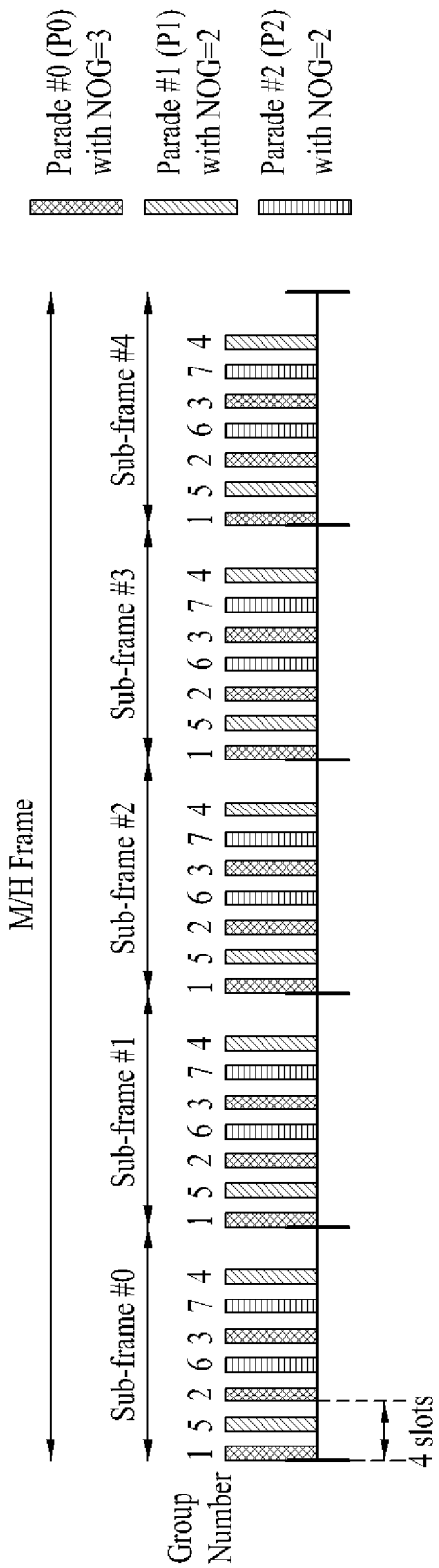

Parade #0 (P0) with NOG=3
Parade #1 (P1) with NOG=2
Parade #2 (P2) with NOG=2
M/H Frame
Sub-frame #0
Sub-frame #1
Sub-frame #2
Sub-frame #3
Sub-frame #4
Group Number
1 5 2 6 3 7 4
4 slots

FIG. 13

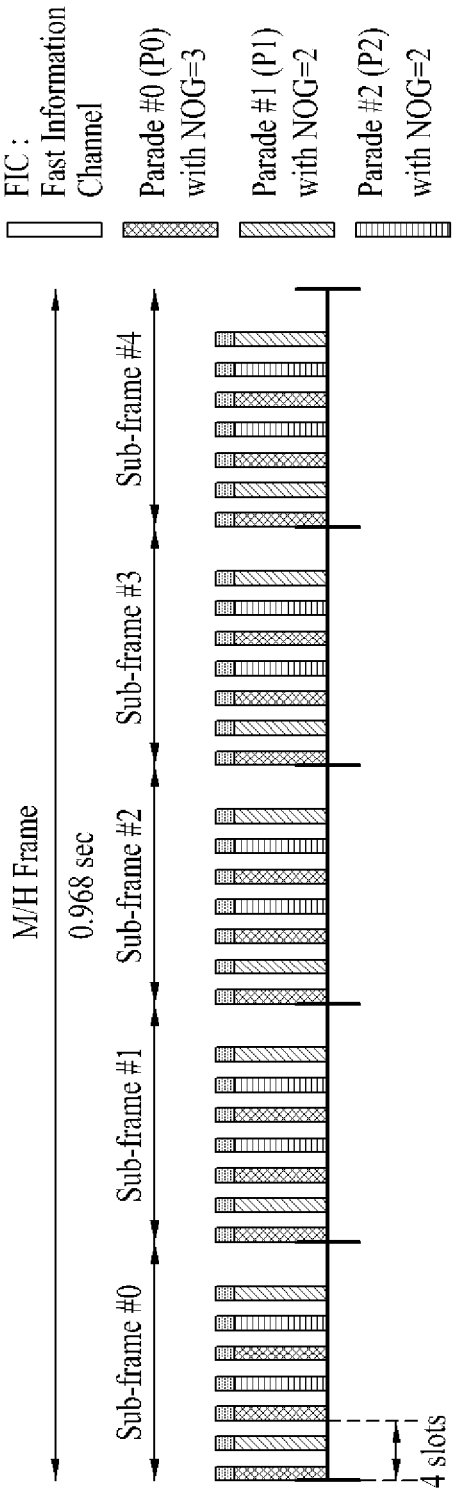

FIC : Fast Information Channel
Parade #0 (P0) with NOG=3
Parade #1 (P1) with NOG=2
Parade #2 (P2) with NOG=2
M/H Frame
0.968 sec
Sub-frame #0
Sub-frame #1
Sub-frame #2
Sub-frame #3
Sub-frame #4
4 slots

FIG. 14

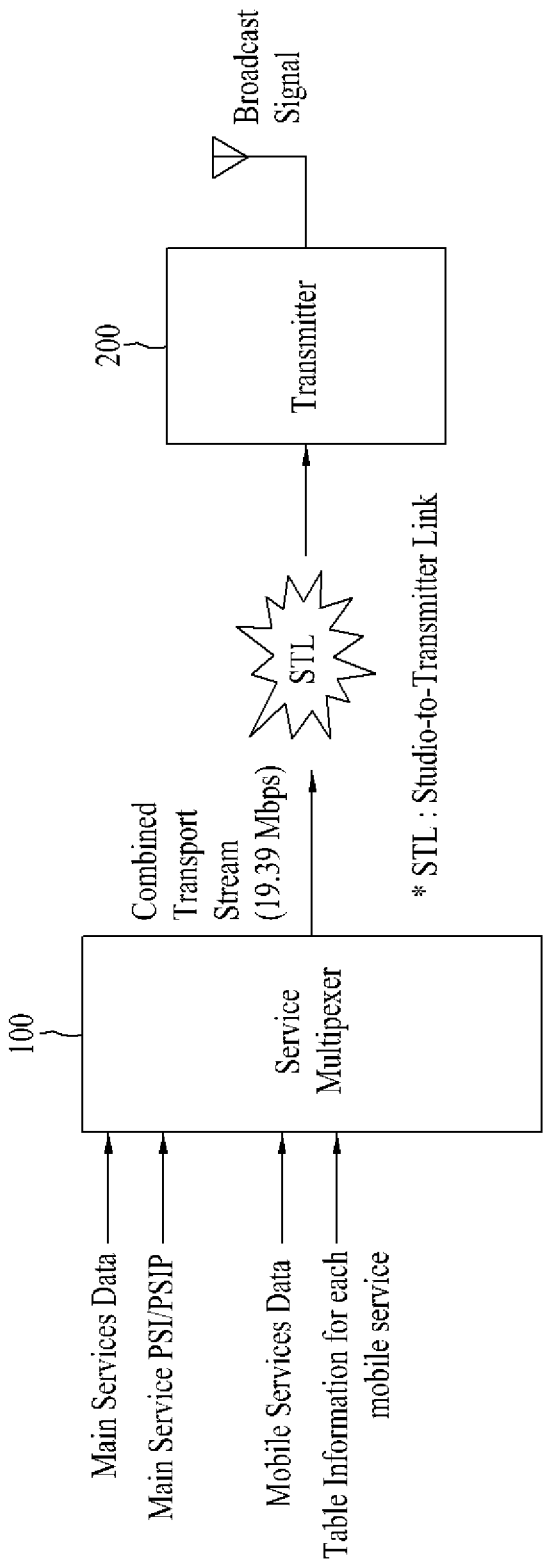

Broadcast Signal
200
Transmitter
STL
* STL : Studio-to-Transmitter Link
Combined Transport Stream (19.39 Mbps)
100
Service Multiplexer
Main Services Data
Main Service PSI/PSIP
Mobile Services Data
Table Information for each mobile service

FIG. 15

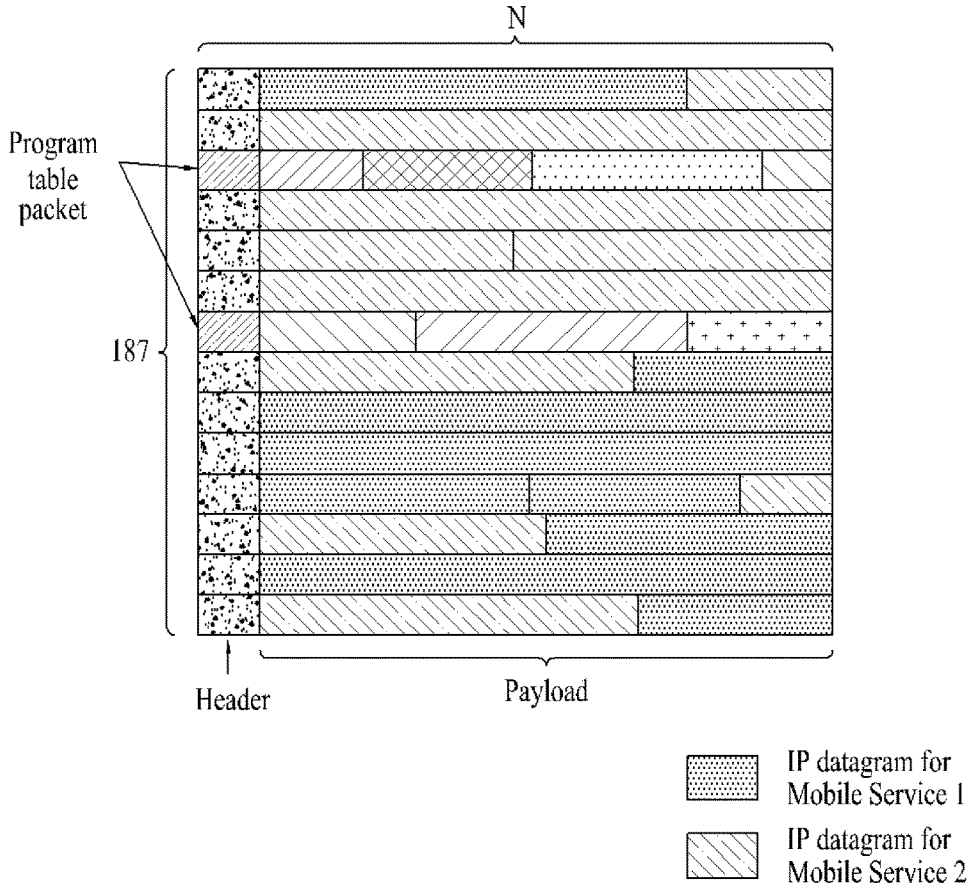


FIG. 16

Type Indicator (3)	Error Indicator (1)	Stuff Indicator (1)	Pointer Field (11)	Payload (N-2) bytes

FIG. 17

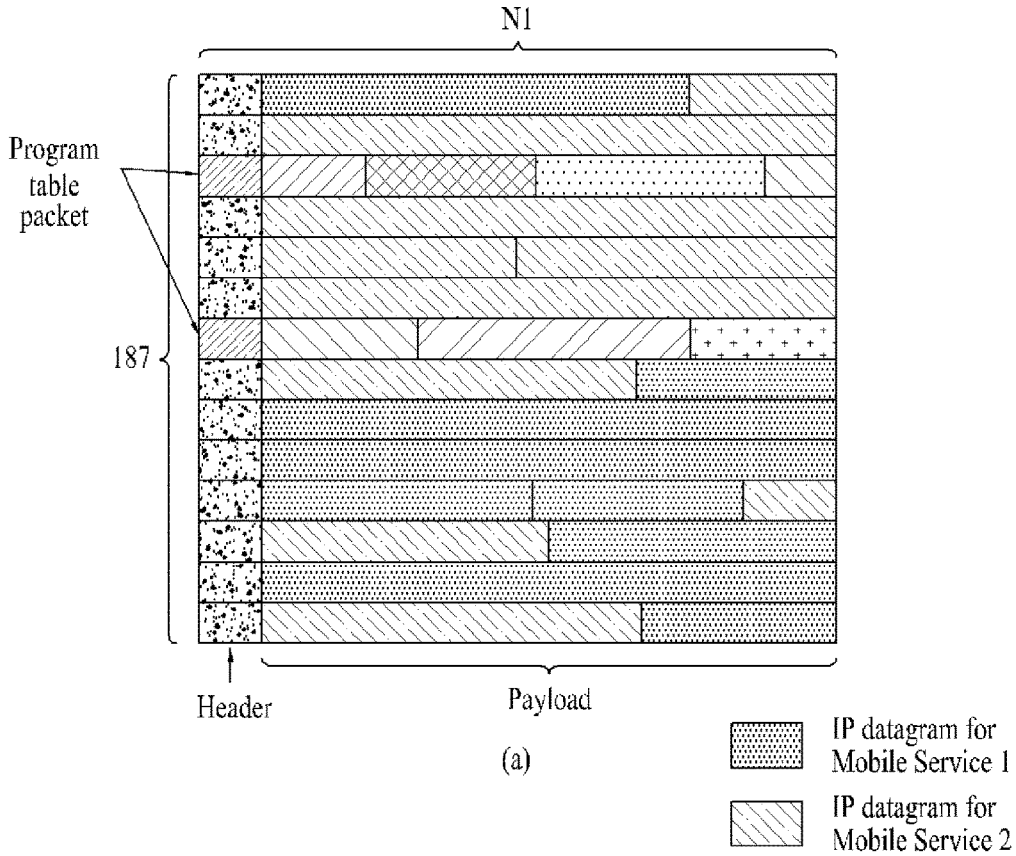

N1
Program table packet
187
Header
Payload
IP datagram for Mobile Service 1
IP datagram for Mobile Service 2
(a)
N2
187
Header
Payload
(b)

FIG. 18

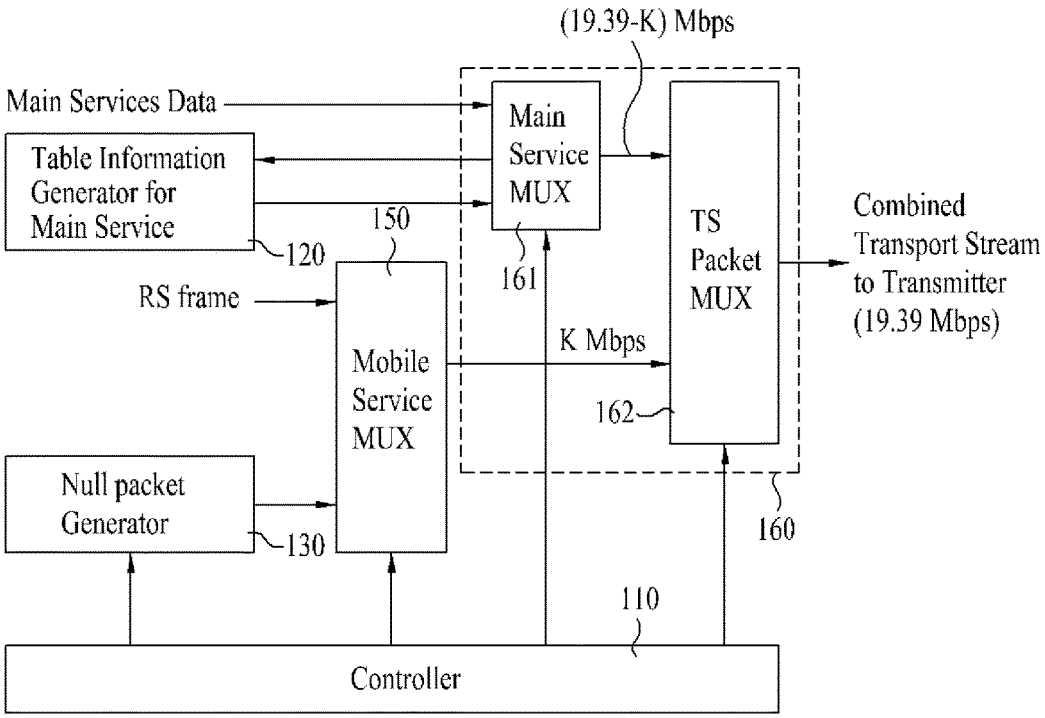

(19.39-K) Mbps
Main Services Data
Table Information Generator for Main Service
120
150
RS frame
Null packet Generator
130
Mobile Service MUX
Main Service MUX
161
K Mbps
TS Packet MUX
162
160
Combined Transport Stream to Transmitter (19.39 Mbps)
110
Controller

FIG. 19

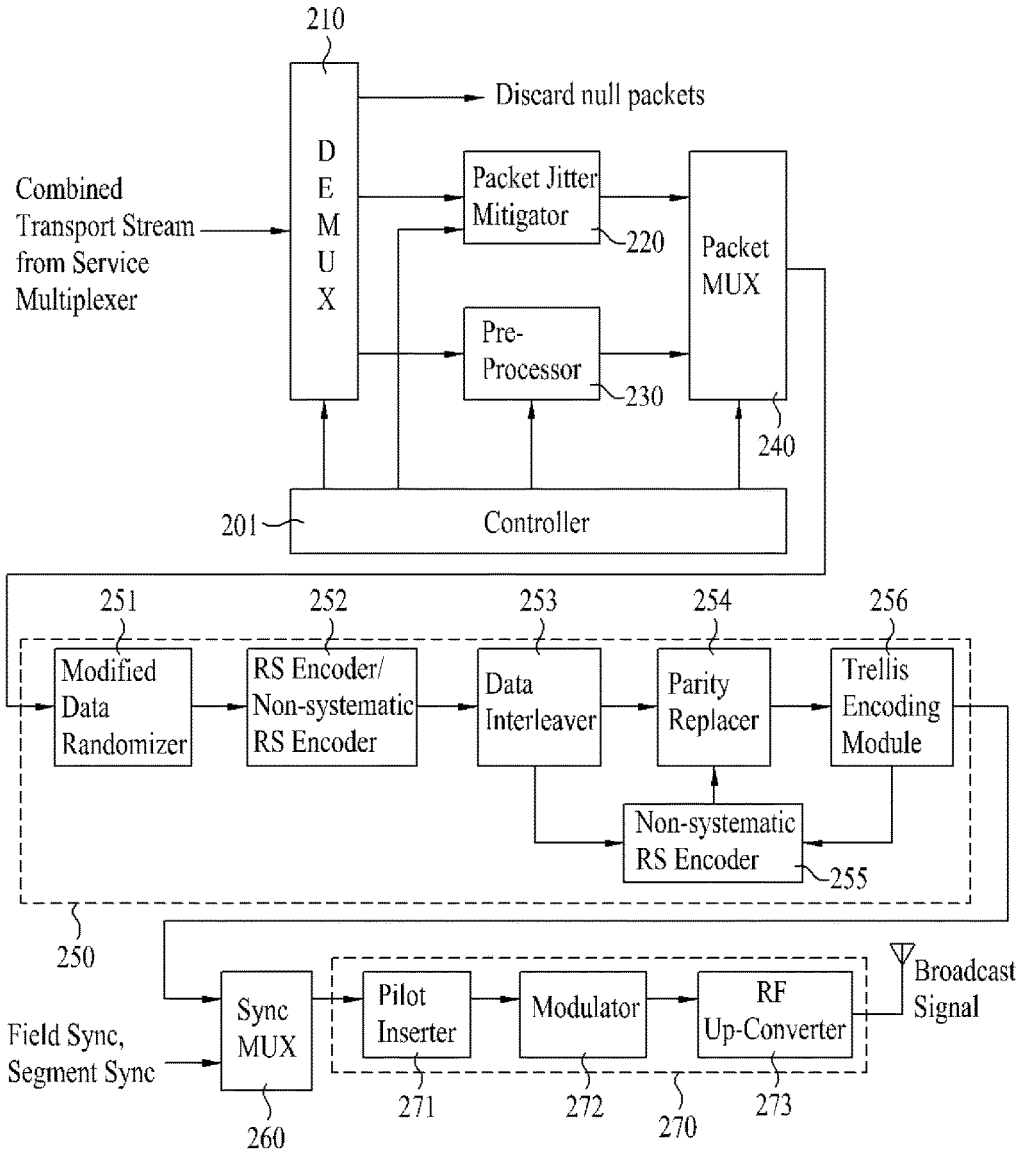

210
Discard null packets
Combined Transport Stream from Service Multiplexer
D E M U X
Packet Jitter Mitigator
220
Packet MUX
Pre-Processor
230
240
201
Controller
251
252
253
254
256
Modified Data Randomizer
RS Encoder/ Non-systematic RS Encoder
Data Interleaver
Parity Replacer
Trellis Encoding Module
Non-systematic RS Encoder
255
250
Field Sync, Segment Sync
Sync MUX
260
Pilot Inserter
271
Modulator
272
270
RF Up-Converter
273
Broadcast Signal

FIG. 20

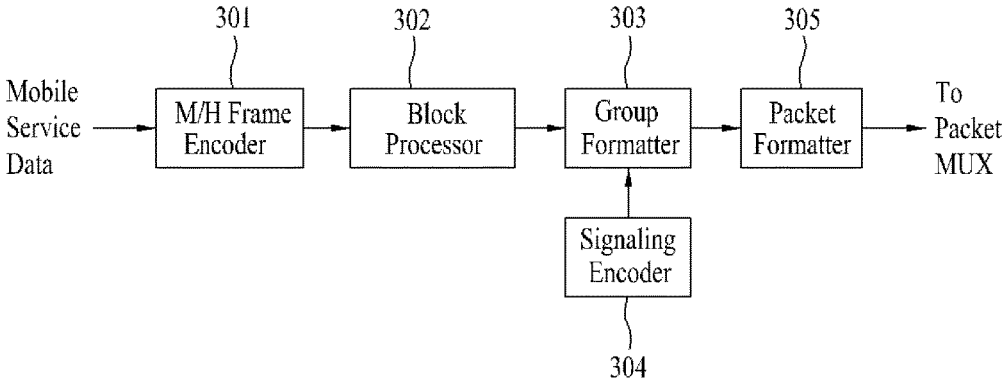

301
302
303
305
Mobile Service Data
M/H Frame Encoder
Block Processor
Group Formatter
Packet Formatter
To Packet MUX
Signaling Encoder
304

FIG. 21

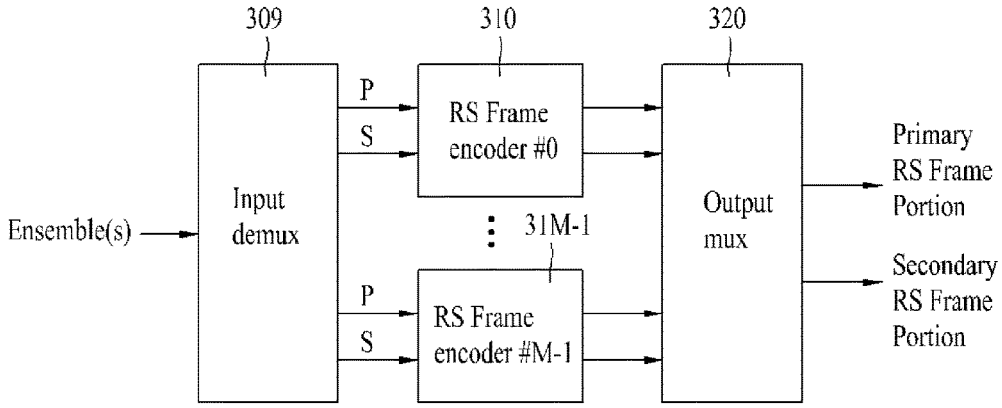

309
310
320
Ensemble(s)
Input demux
P
S
RS Frame encoder #0
31M-1
P
S
RS Frame encoder #M-1
Output mux
Primary RS Frame Portion
Secondary RS Frame Portion

FIG. 22

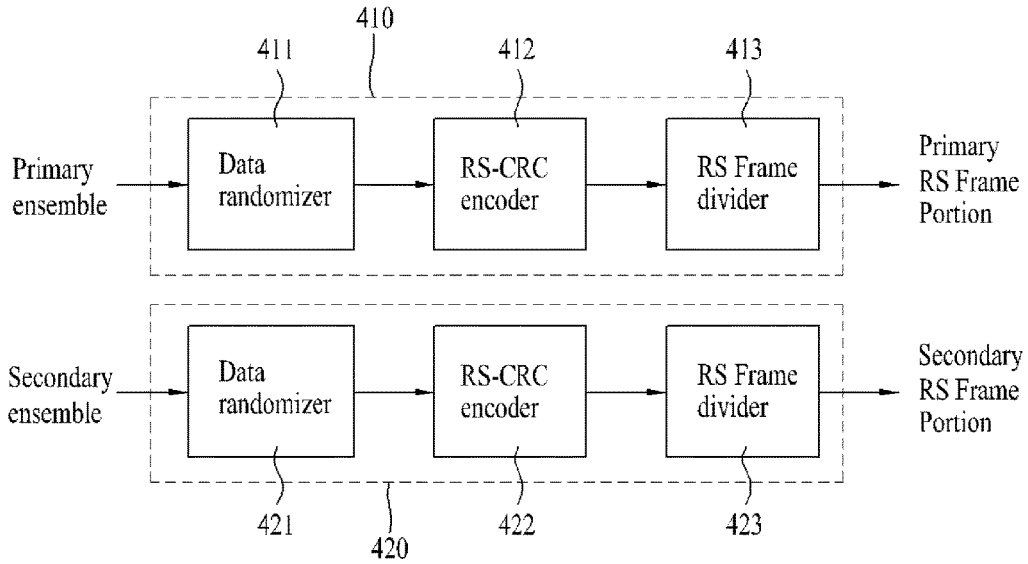

410
411
412
413
Primary ensemble
Data randomizer
RS-CRC encoder
RS Frame divider
Primary RS Frame Portion
Secondary ensemble
Data randomizer
RS-CRC encoder
RS Frame divider
Secondary RS Frame Portion
421
420
422
423

FIG. 23

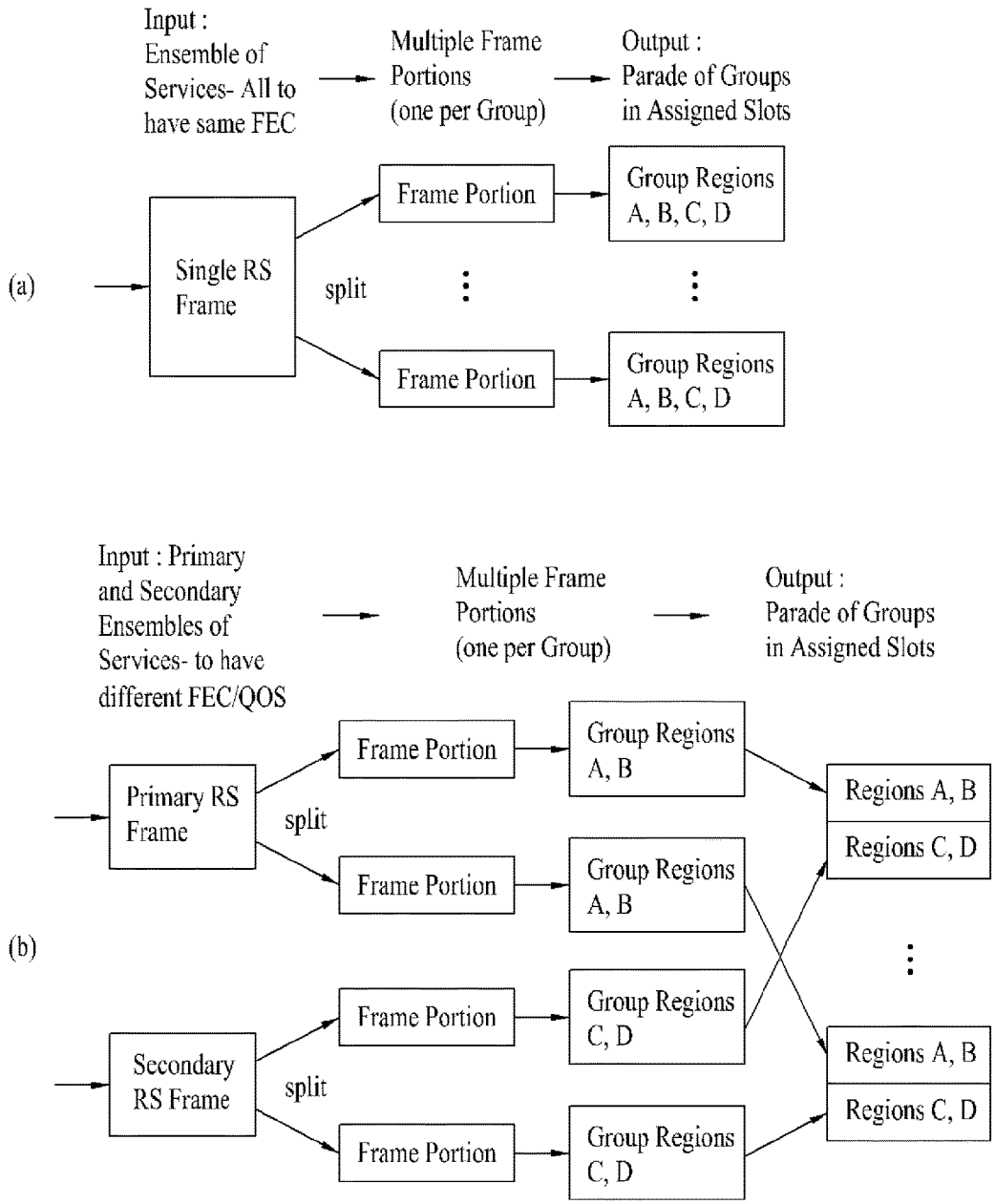

(a)
Input : Ensemble of Services- All to have same FEC
Multiple Frame Portions (one per Group)
Output : Parade of Groups in Assigned Slots
Single RS Frame
split
Frame Portion
Frame Portion
Group Regions A, B, C, D
Group Regions A, B, C, D
(b)
Input : Primary and Secondary Ensembles of Services- to have different FEC/QOS
Multiple Frame Portions (one per Group)
Output : Parade of Groups in Assigned Slots
Primary RS Frame
Secondary RS Frame
split
split
Frame Portion
Frame Portion
Frame Portion
Frame Portion
Group Regions A, B
Group Regions A, B
Group Regions C, D
Group Regions C, D
Regions A, B
Regions C, D
Regions A, B
Regions C, D

FIG. 24

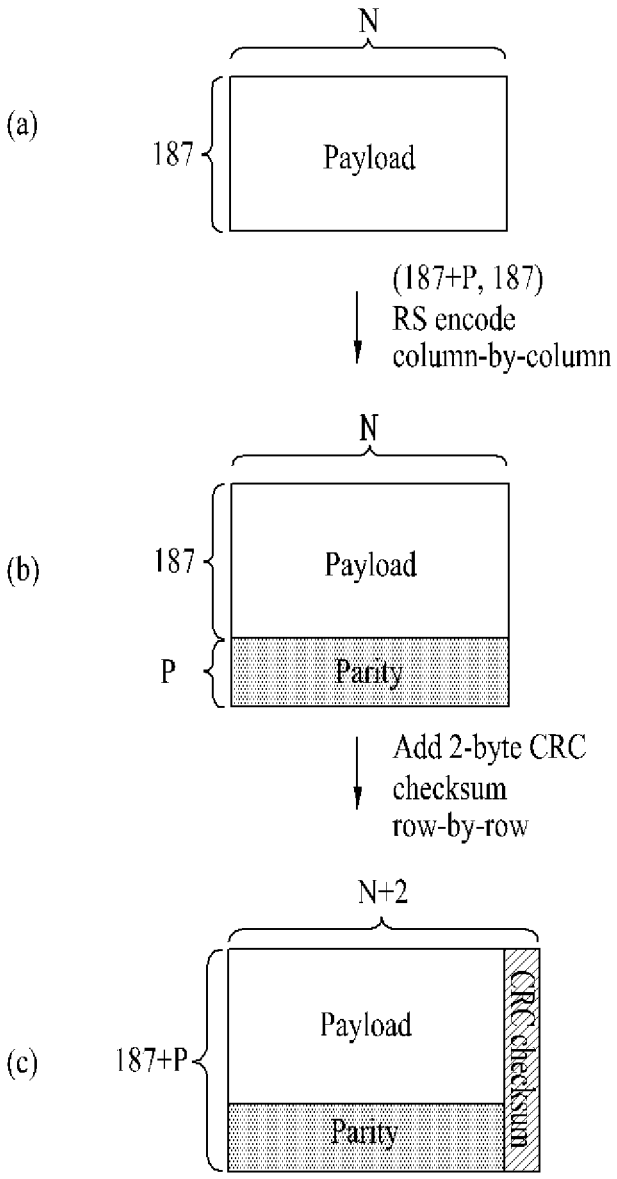

(a)
N
187
Payload
(187+P, 187)
RS encode
column-by-column
(b)
N
187
Payload
P
Parity
Add 2-byte CRC
checksum
row-by-row
(c)
N+2
187+P
Payload
CRC checksum
Parity

FIG. 25

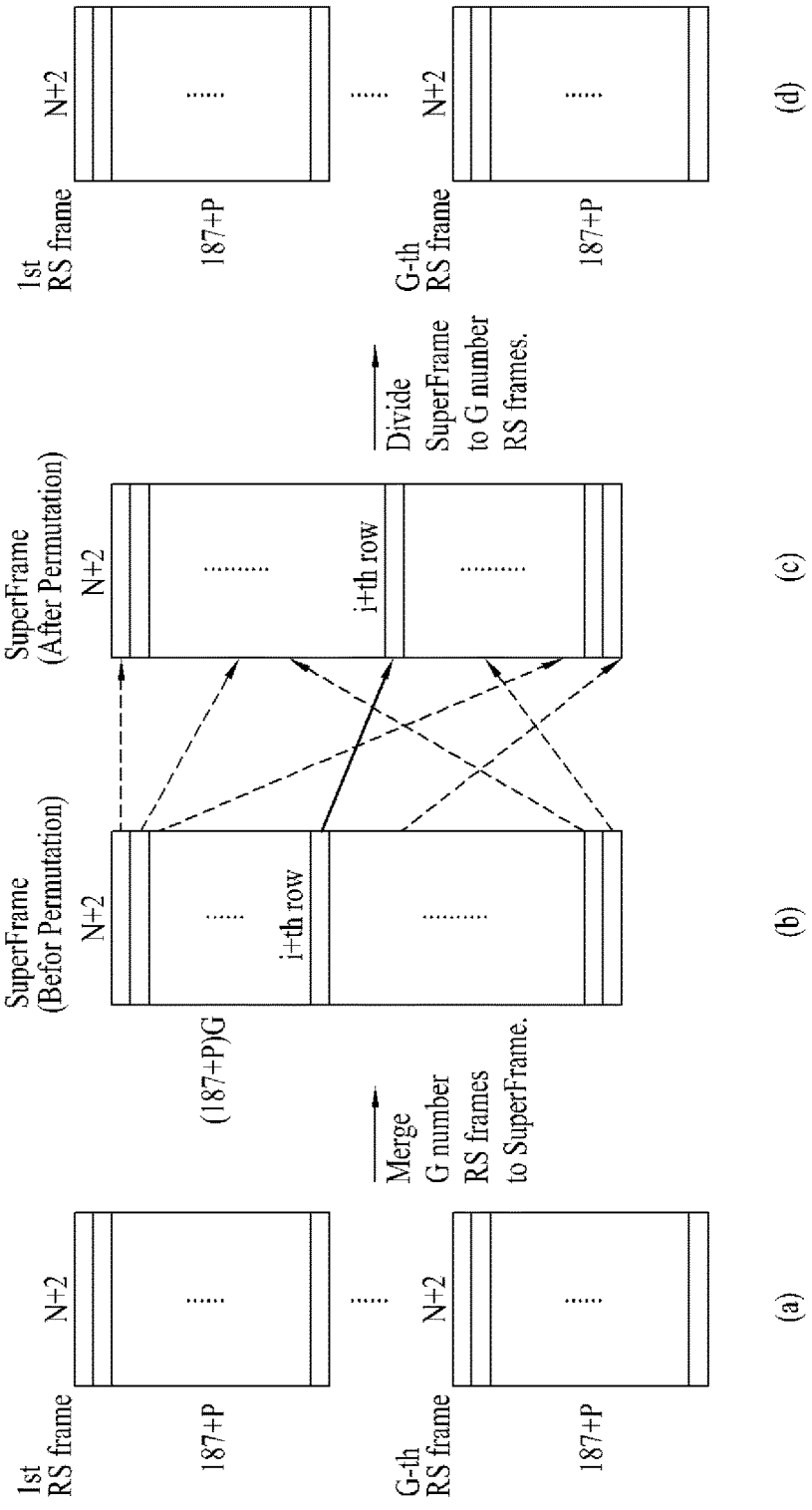

1st RS frame
G-th RS frame
187+P
N+2
(187+P)G
SuperFrame (Befor Permutation)
SuperFrame (After Permutation)
i+th row
Merge G number RS frames to SuperFrame.
Divide SuperFrame to G number RS frames.
(a)
(b)
(c)
(d)

FIG. 26

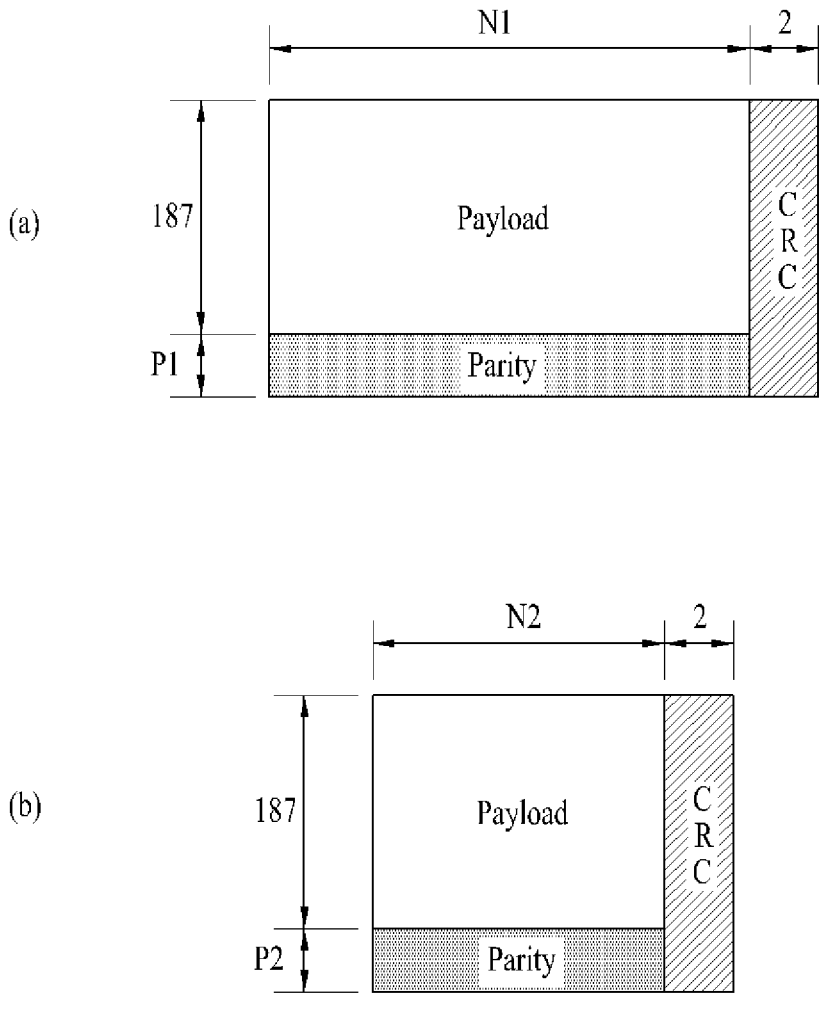

(a)
N1
2
187
Payload
C
R
C
P1
Parity
(b)
N2
2
187
Payload
C
R
C
P2
Parity

FIG. 27

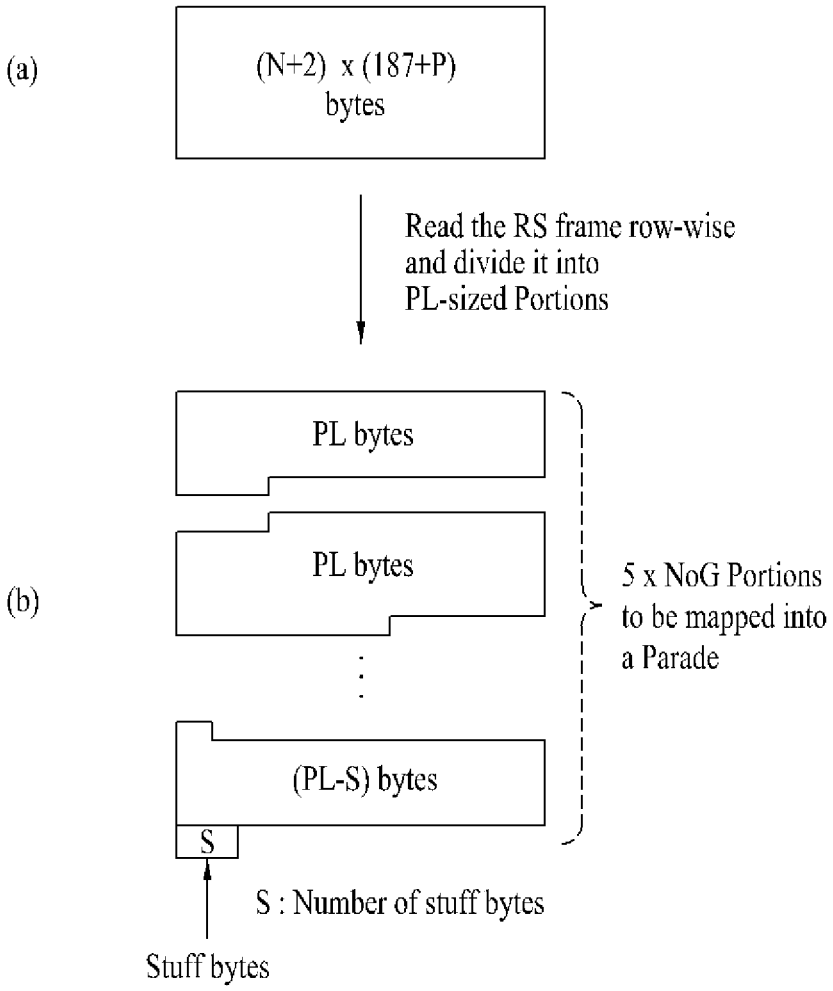

(a)
(N+2) x (187+P) bytes
Read the RS frame row-wise and divide it into PL-sized Portions
(b)
PL bytes
PL bytes
(PL-S) bytes
S
5 x NoG Portions to be mapped into a Parade
S : Number of stuff bytes
Stuff bytes

FIG. 28

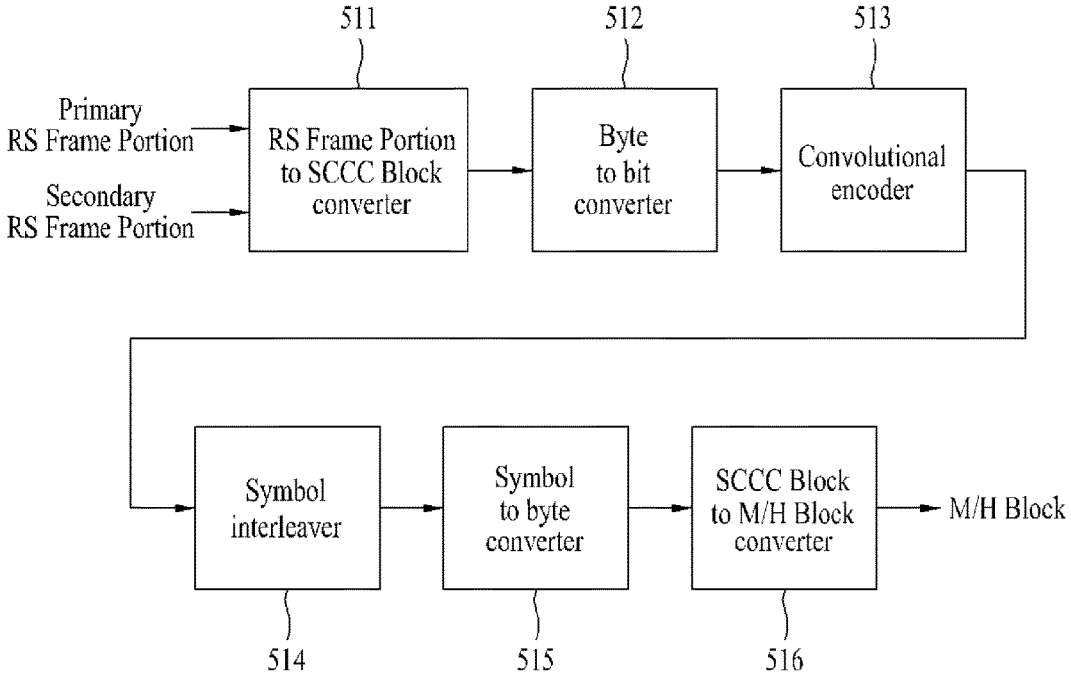

511
512
513
Primary RS Frame Portion
Secondary RS Frame Portion
RS Frame Portion to SCCC Block converter
Byte to bit converter
Convolutional encoder
Symbol interleaver
Symbol to byte converter
SCCC Block to M/H Block converter
M/H Block
514
515
516

FIG. 29

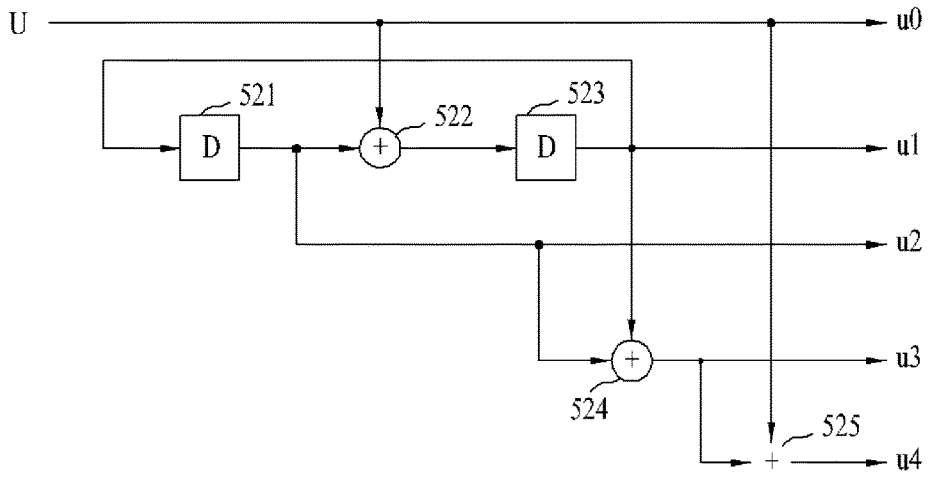

U
u0
521
522
523
D
D
u1
u2
u3
524
525
u4

FIG. 30

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	2110	2111
P'(i)	0	89	267	534	890	1335	1869	2492	3204	4005	799	1778	2846	4003	1153	2488	...	2809	2272
P(i)	0	89	267	534	890	1335	1869	799	1778	1153	1329	526	79	2037	253	874	...	444	2048

FIG. 31

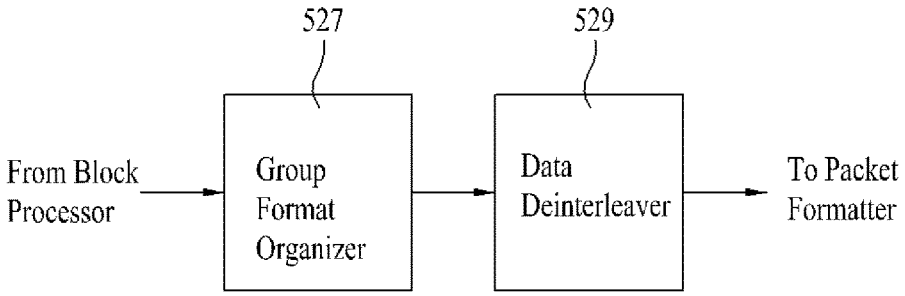
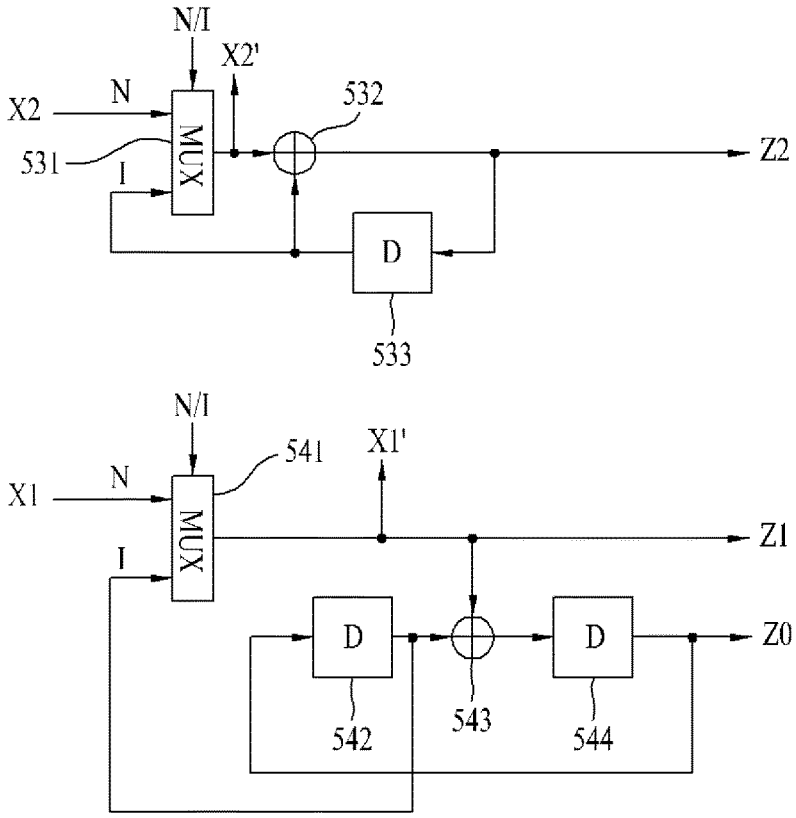


FIG. 32



N : Normal mode, I : Initialization mode

FIG. 33

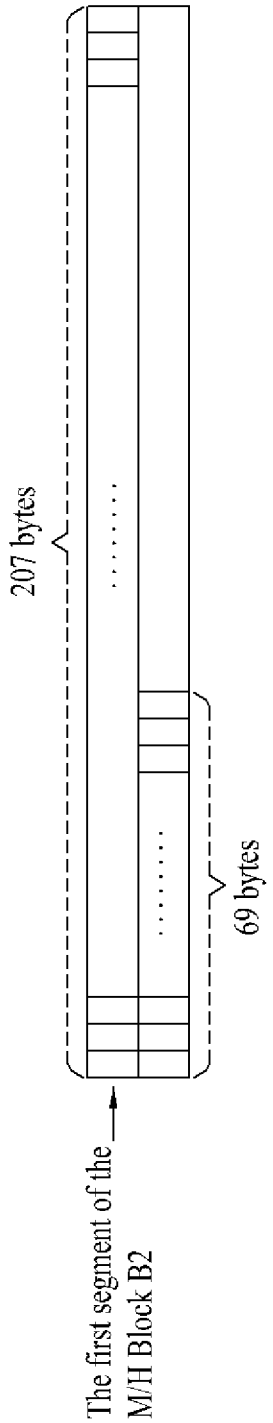


FIG. 34

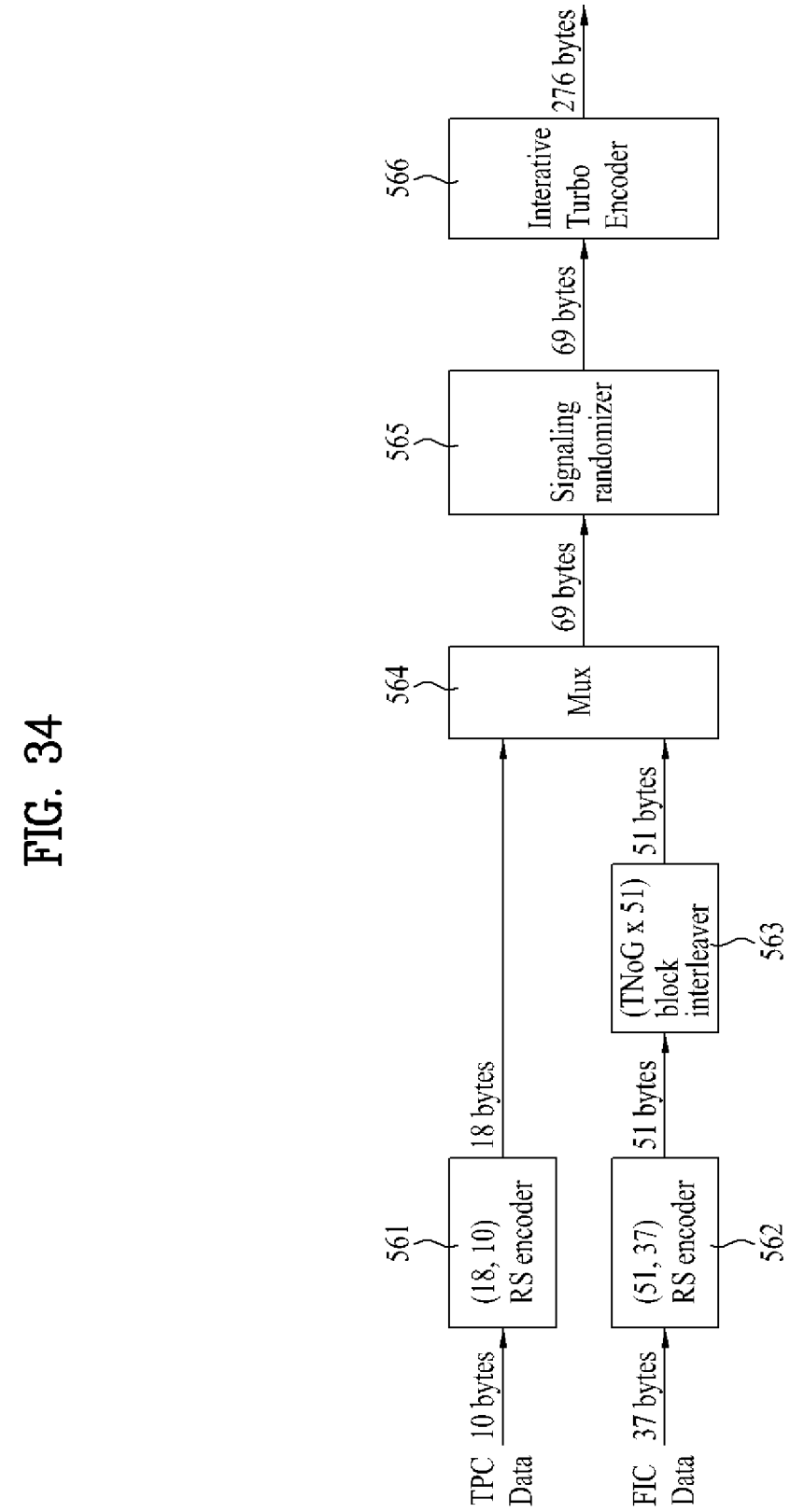

TPC Data
10 bytes
FIC Data
37 bytes
561
(18, 10) RS encoder
562
(51, 37) RS encoder
18 bytes
51 bytes
563
(TNoG x 51) block interleaver
51 bytes
564
Mux
69 bytes
565
Signaling randomizer
69 bytes
566
Interative Turbo Encoder
276 bytes

FIG. 35

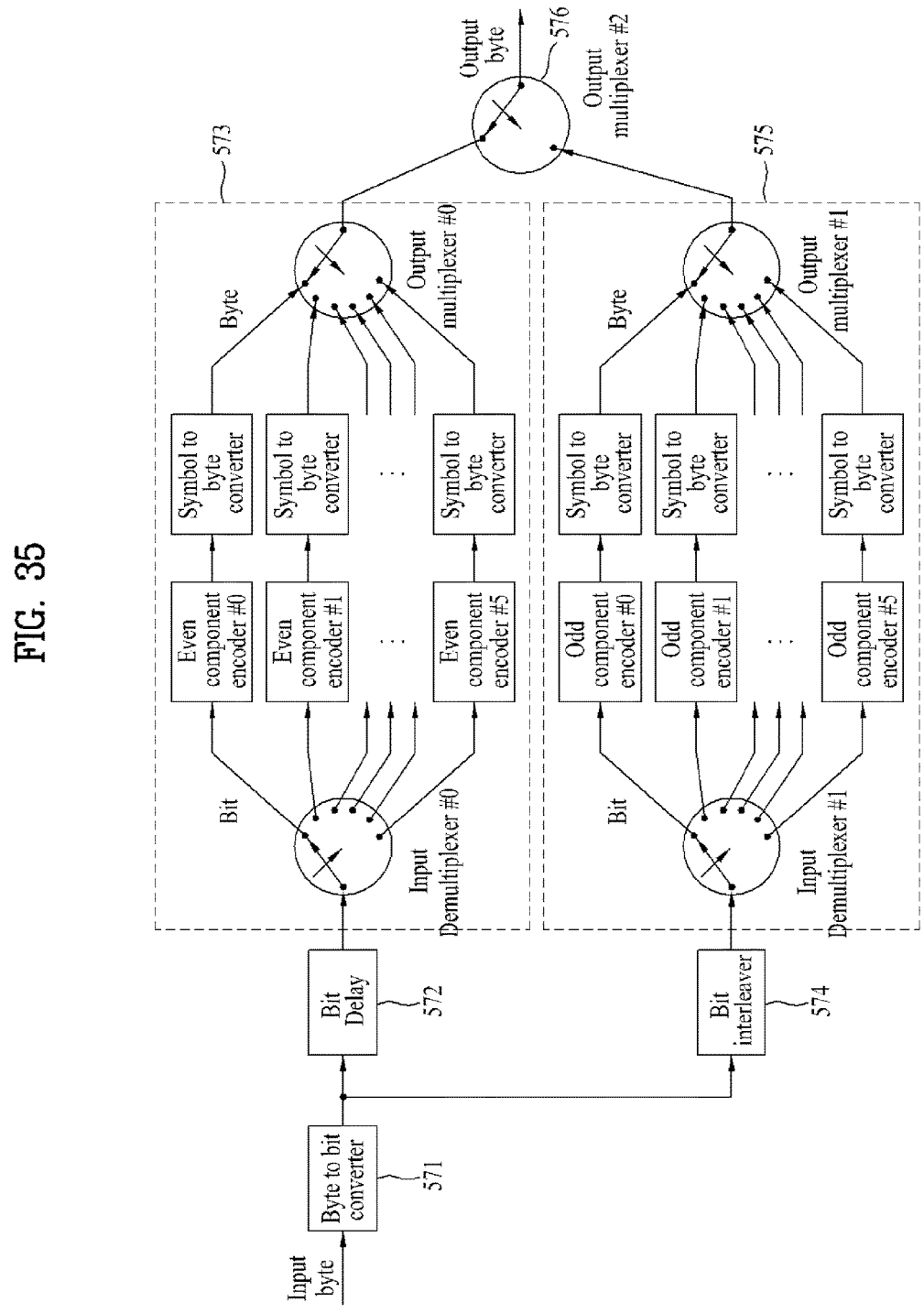
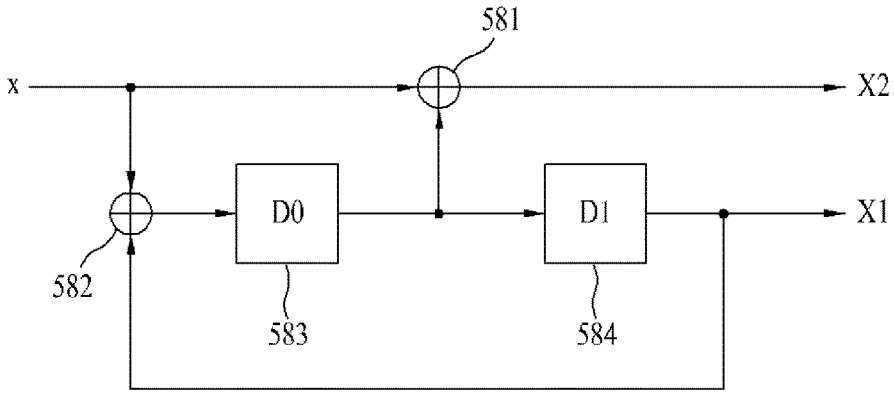

Input byte
Byte to bit converter
571
Bit Delay
572
Bit interleaver
574
Input Demultiplexer #0
Bit
Even component encoder #0
Even component encoder #1
Even component encoder #5
Symbol to byte converter
Symbol to byte converter
Symbol to byte converter
Byte
Output multiplexer #0
573
Input Demultiplexer #1
Bit
Odd component encoder #0
Odd component encoder #1
Odd component encoder #5
Symbol to byte converter
Symbol to byte converter
Symbol to byte converter
Byte
Output multiplexer #1
575
Output multiplexer #2
576
Output byte

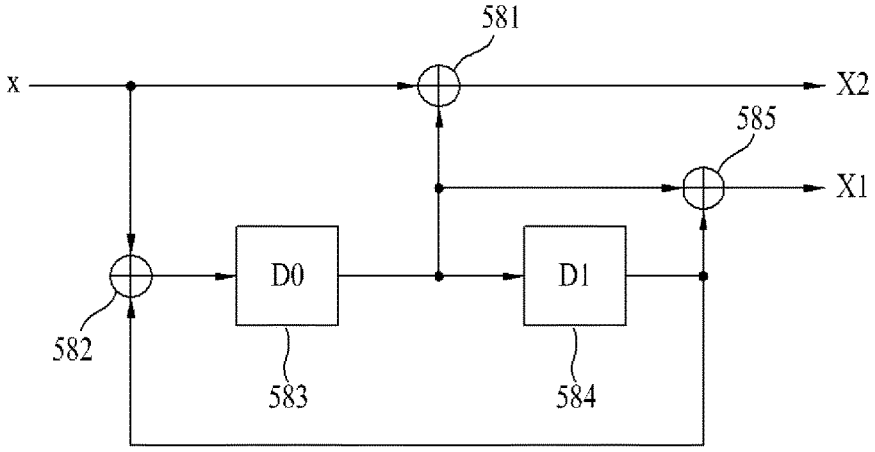
FIG. 36

(a)



even component encoder

(b)



odd component encoder

FIG. 37

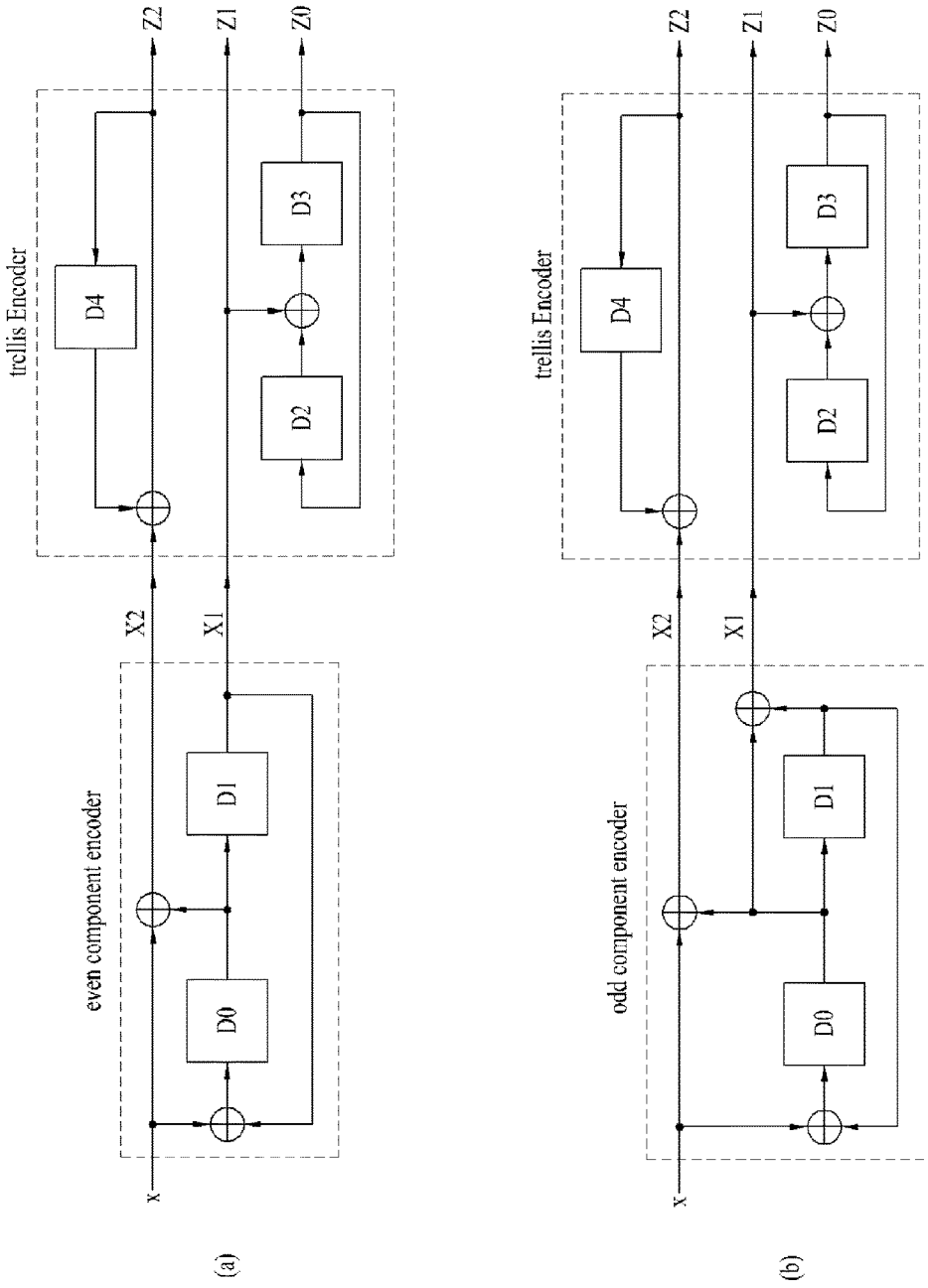

Z2
Z1
Z0
trellis Encoder
D4
D3
D2
X2
X1
even component encoder
D1
D0
x
(a)
Z2
Z1
Z0
trellis Encoder
D4
D3
D2
X2
X1
odd component encoder
D1
D0
x
(b)

FIG. 38

Syntax	No. of bits	Format
TPC_data {		
Sub-Frame_number	3	uimsbf
Slot_number	4	uimsbf
Parade_id	7	uimsbf
starting_Group_number	4	uimsbf
number_of_Groups	3	uimsbf
Parade_repetition_cycle	3	uimsbf
RS_Frame_mode	2	bslbf
RS_code_mode_primary	2	bslbf
RS_code_mode_secondary	2	bslbf
SCCC_Block_mode	2	bslbf
SCCC_outer_code_mode_A	2	bslbf
SCCC_outer_code_mode_B	2	bslbf
SCCC_outer_code_mode_C	2	bslbf
SCCC_outer_code_mode_D	2	bslbf
FIC_version	5	uimsbf
Parade_continuity_counter	4	uimsbf
TNoG	5	uimsbf
reserved	26	bslbf
}		

FIG. 39

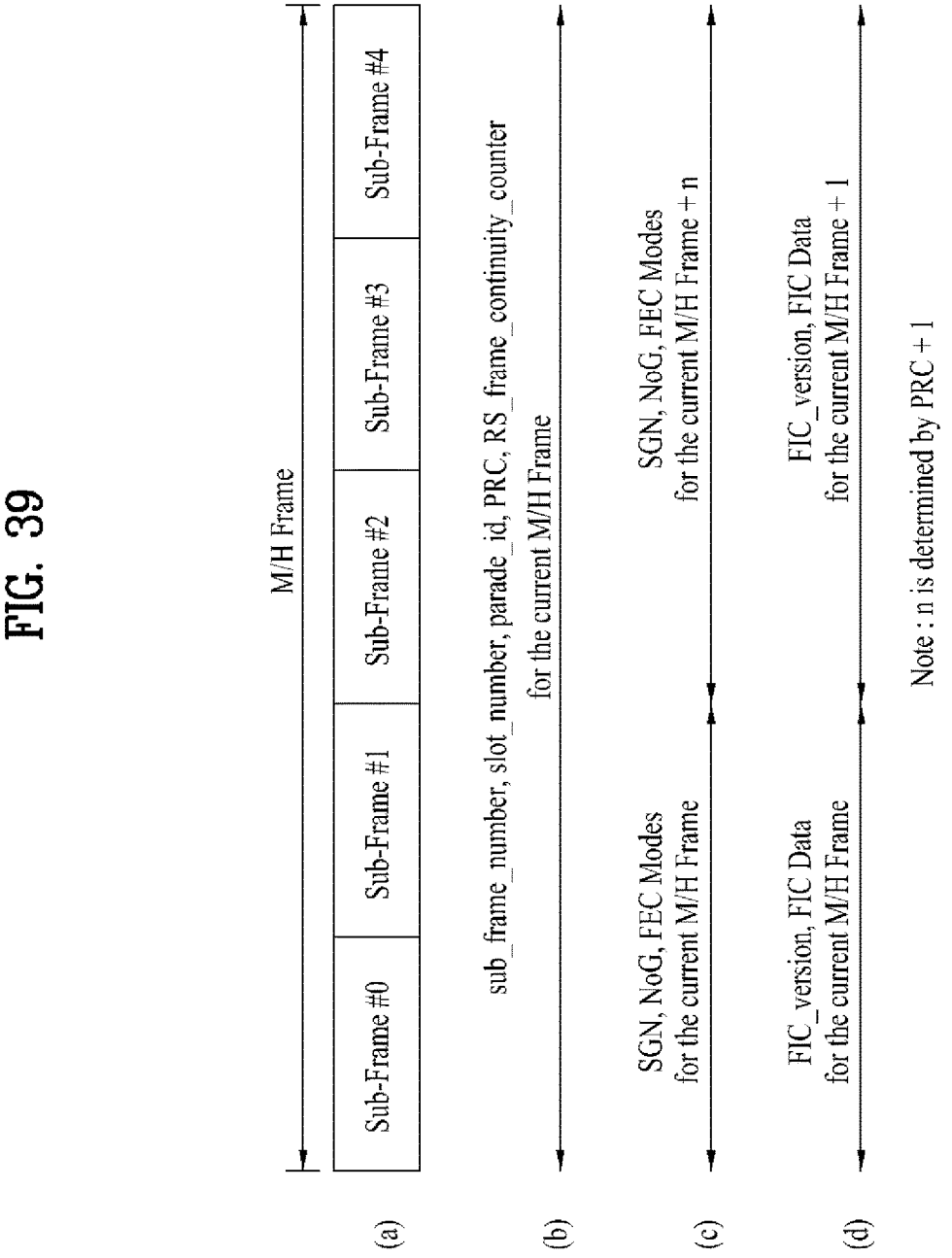

M/H Frame
Sub-Frame #0
Sub-Frame #1
Sub-Frame #2
Sub-Frame #3
Sub-Frame #4
(a)
sub_frame_number, slot_number, parade_id, PRC, RS_frame_continuity_counter
for the current M/H Frame
(b)
SGN, NoG, FEC Modes
for the current M/H Frame
SGN, NoG, FEC Modes
for the current M/H Frame + n
(c)
FIC_version, FIC Data
for the current M/H Frame
FIC_version, FIC Data
for the current M/H Frame + 1
(d)
Note : n is determined by PRC + 1

FIG. 40

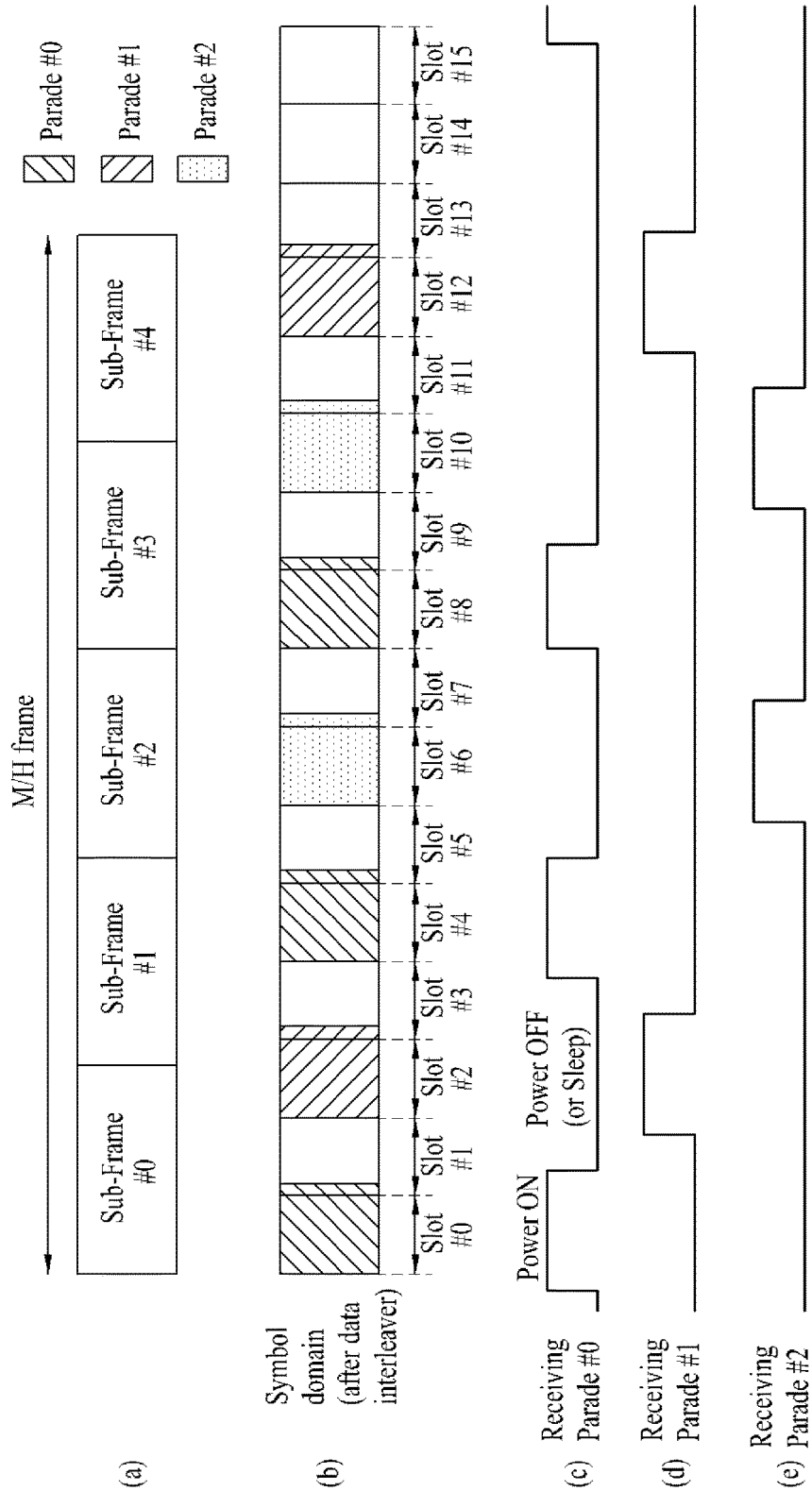

(a)
M/H frame
Sub-Frame #0
Sub-Frame #1
Sub-Frame #2
Sub-Frame #3
Sub-Frame #4
Parade #0
Parade #1
Parade #2
(b)
Symbol domain (after data interleaver)
Slot #0
Slot #1
Slot #2
Slot #3
Slot #4
Slot #5
Slot #6
Slot #7
Slot #8
Slot #9
Slot #10
Slot #11
Slot #12
Slot #13
Slot #14
Slot #15
(c)
Receiving Parade #0
Power ON
Power OFF (or Sleep)
(d)
Receiving Parade #1
(e)
Receiving Parade #2

FIG. 41

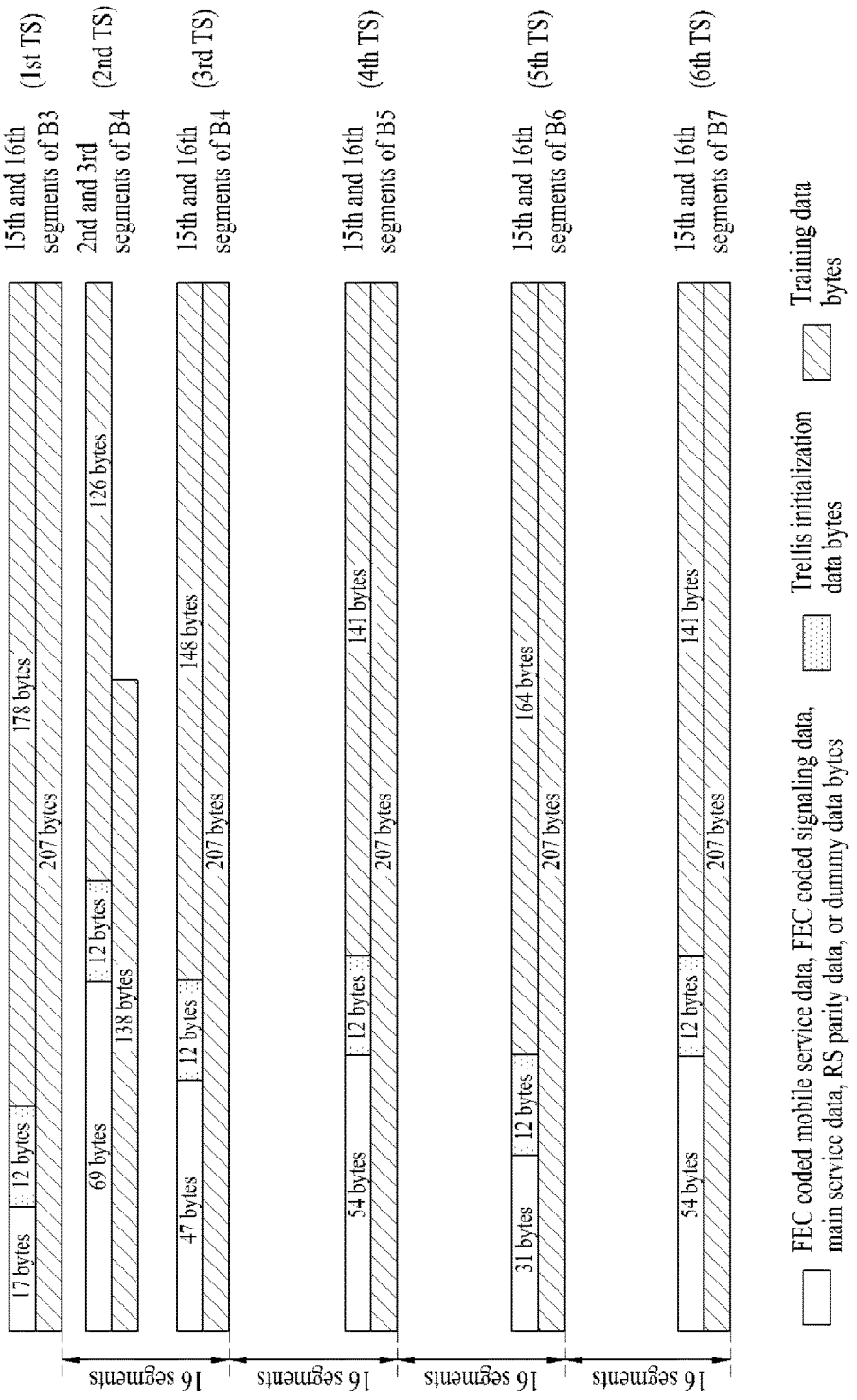

15th and 16th segments of B3 (1st TS)
2nd and 3rd segments of B4 (2nd TS)
15th and 16th segments of B4 (3rd TS)
15th and 16th segments of B5 (4th TS)
15th and 16th segments of B6 (5th TS)
15th and 16th segments of B7 (6th TS)
17 bytes
12 bytes
178 bytes
207 bytes
69 bytes
12 bytes
126 bytes
138 bytes
47 bytes
12 bytes
148 bytes
207 bytes
54 bytes
12 bytes
141 bytes
207 bytes
31 bytes
12 bytes
164 bytes
207 bytes
54 bytes
12 bytes
141 bytes
207 bytes
16 segments
16 segments
16 segments
16 segments
FEC coded mobile service data, FEC coded signaling data, main service data, RS parity data, or dummy data bytes
Trellis initialization data bytes
Training data bytes

FIG. 42

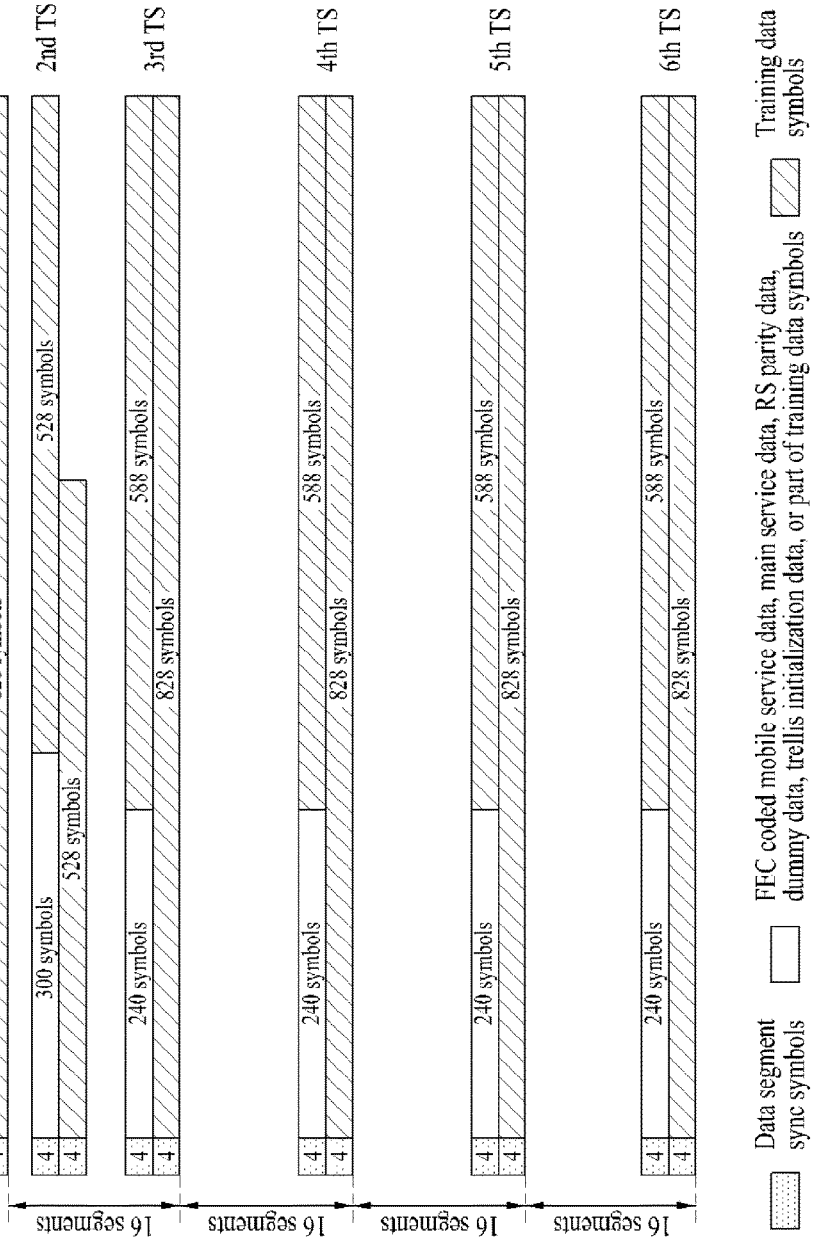

1st TS
2nd TS
3rd TS
4th TS
5th TS
6th TS
4
240 symbols
588 symbols
828 symbols
300 symbols
528 symbols
16 segments
Data segment sync symbols
FEC coded mobile service data, main service data, RS parity data, dummy data, trellis initialization data, or part of training data symbols
Training data symbols

FIG. 43

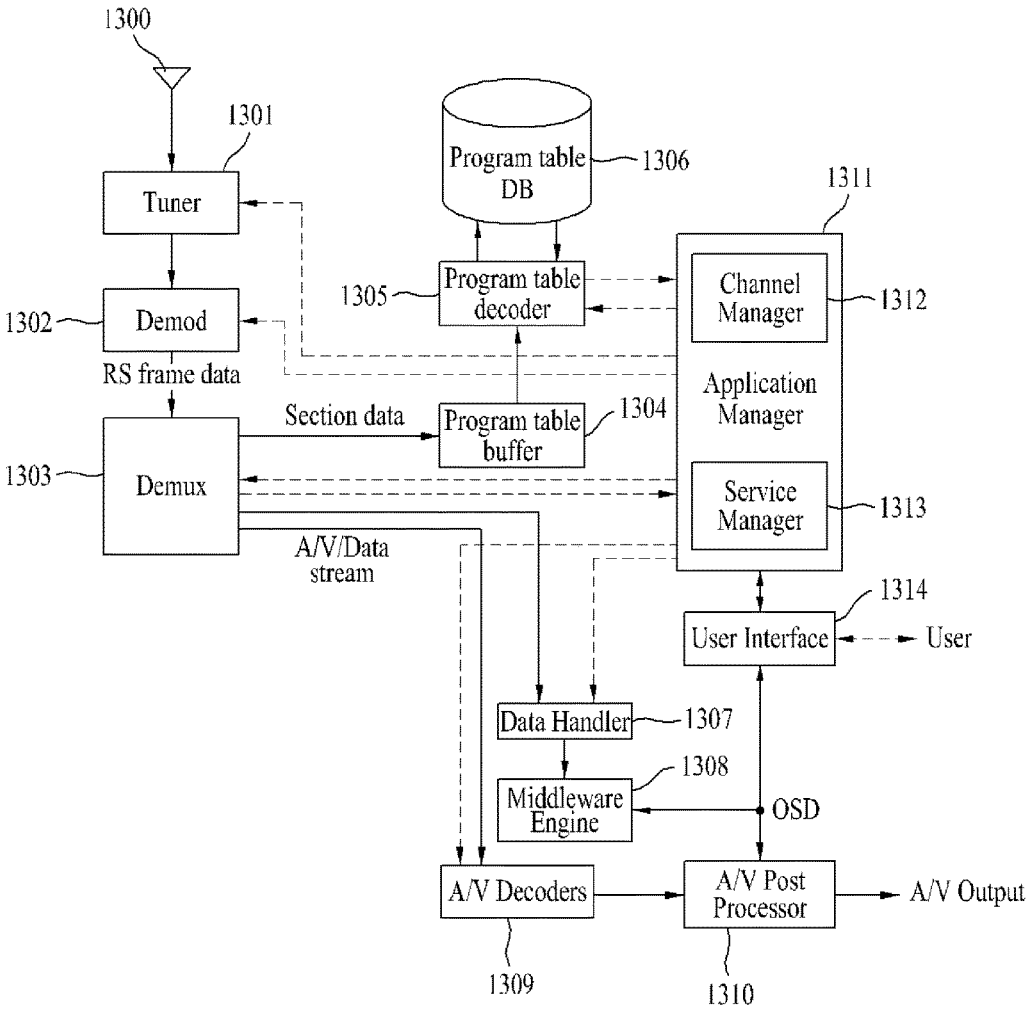

1300
1301
Tuner
Program table DB
1306
1311
1305
Program table decoder
Channel Manager
1312
1302
Demod
RS frame data
Application Manager
Section data
Program table buffer
1304
1303
Demux
Service Manager
1313
A/V/Data stream
1314
User Interface
User
Data Handler
1307
1308
Middleware Engine
OSD
A/V Decoders
A/V Post Processor
A/V Output
1309
1310

FIG. 44

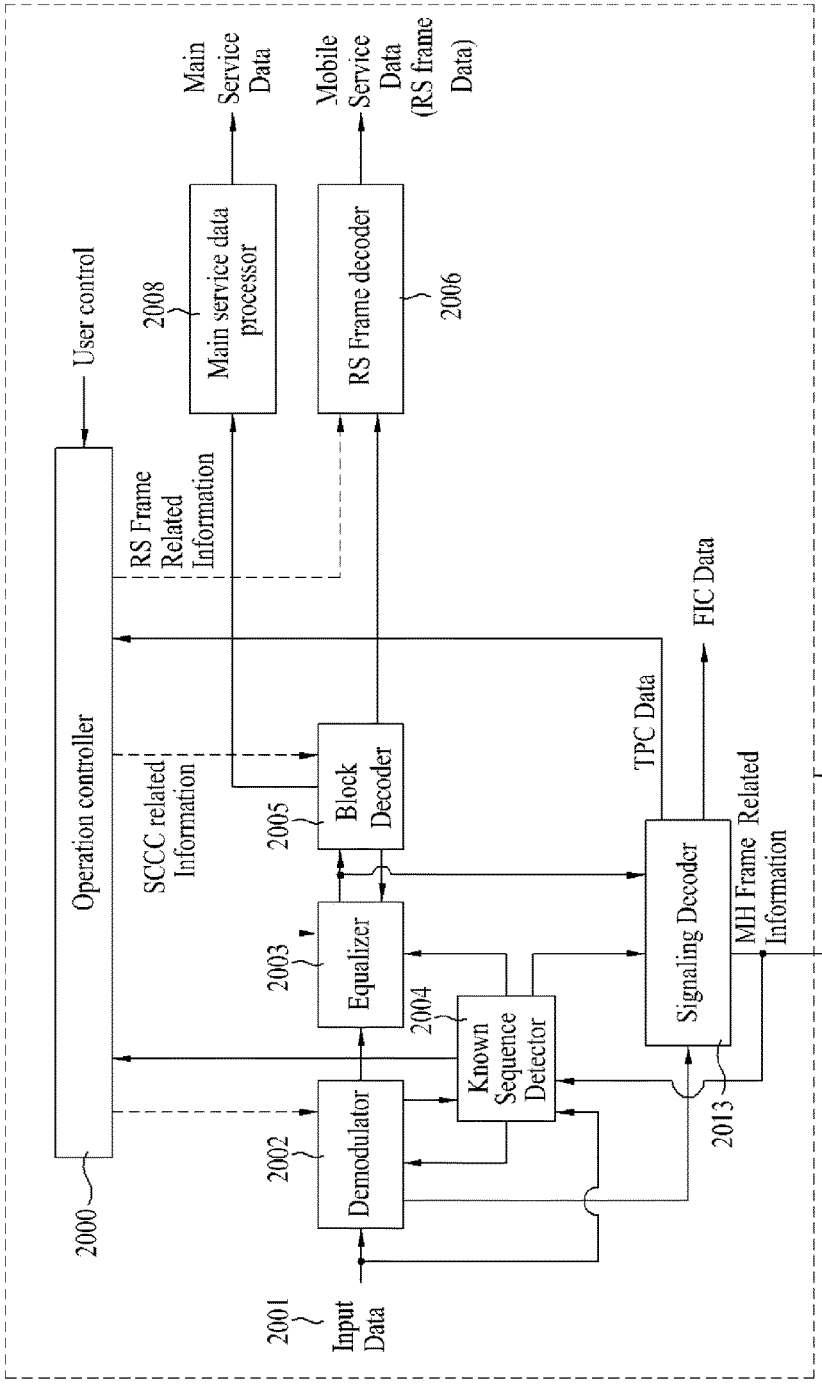

Main Service Data
Mobile Service Data (RS frame Data)
User control
2008
Main service data processor
RS Frame decoder
2006
RS Frame Related Information
FIC Data
TPC Data
Power On/Off
Operation controller
SCCC related Information
2005
Block Decoder
Signaling Decoder
MH Frame Related Information
2003
Equalizer
2004
Known Sequence Detector
Power Controller
2000
2002
Demodulator
2013
5000
2001
Input Data

FIG. 45

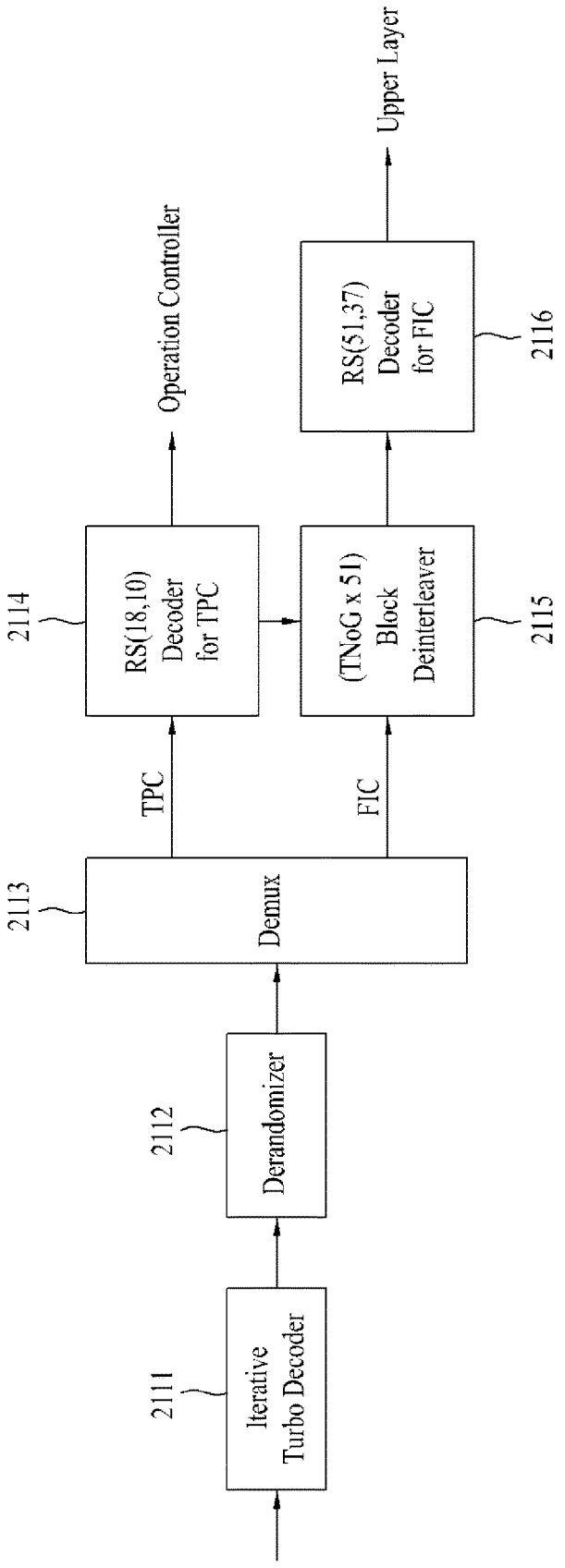

Iterative Turbo Decoder
2111
Derandomizer
2112
Demux
2113
TPC
FIC
RS(18,10) Decoder for TPC
2114
(TNoG x 51) Block Deinterleaver
2115
RS(51,37) Decoder for FIC
2116
Operation Controller
Upper Layer

FIG. 46

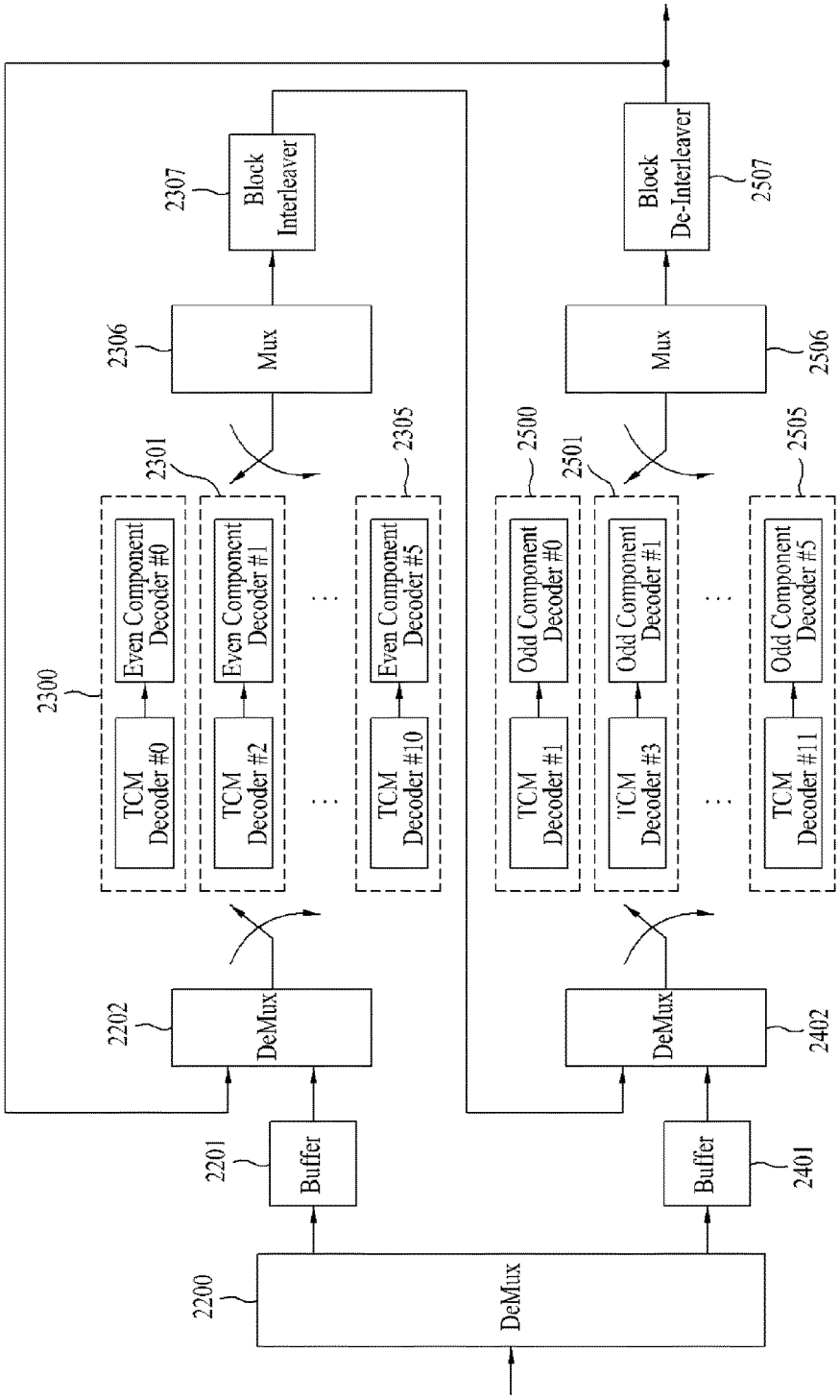

2200
DeMux
2201
Buffer
2202
DeMux
2300
TCM Decoder #0
Even Component Decoder #0
TCM Decoder #2
Even Component Decoder #1
TCM Decoder #10
Even Component Decoder #5
2301
2305
2306
Mux
2307
Block Interleaver
2401
Buffer
2402
DeMux
2500
TCM Decoder #1
Odd Component Decoder #0
2501
TCM Decoder #3
Odd Component Decoder #1
TCM Decoder #11
Odd Component Decoder #5
2505
2506
Mux
2507
Block De-Interleaver

FIG. 47

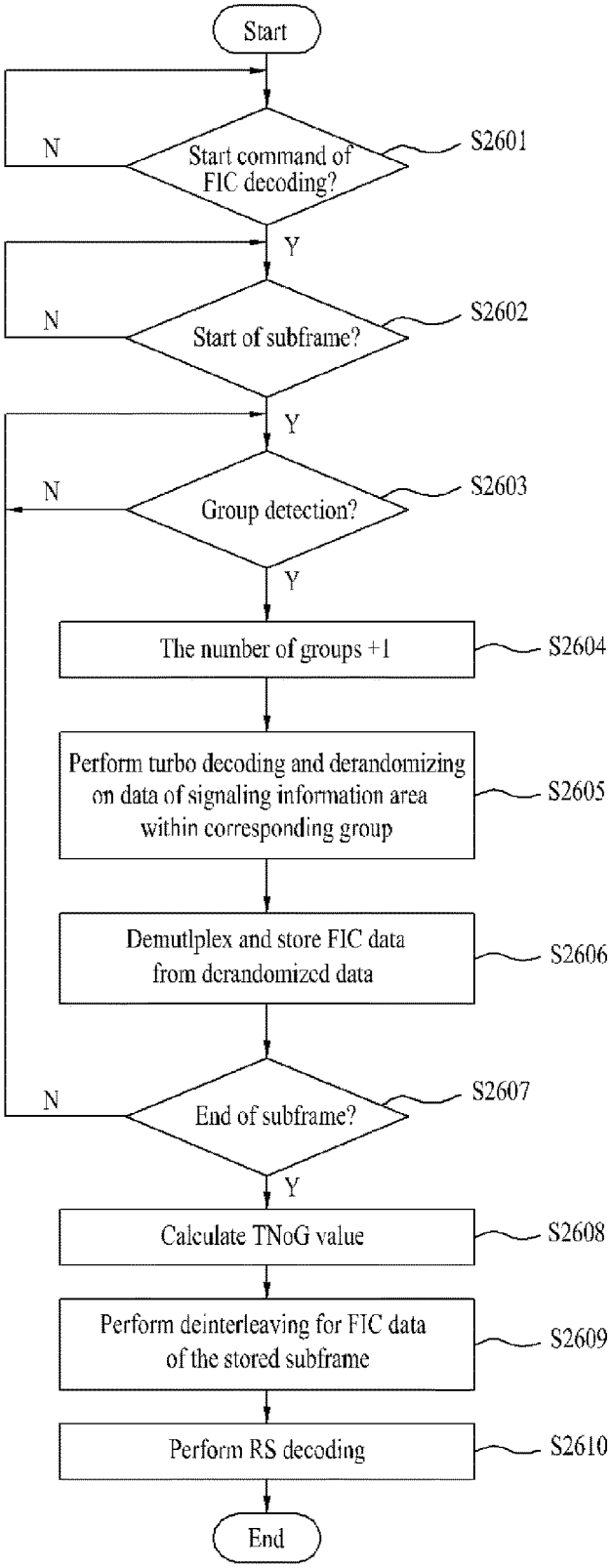

Start
Start command of FIC decoding?
S2601
N
Y
Start of subframe?
S2602
N
Y
Group detection?
S2603
N
Y
The number of groups +1
S2604
Perform turbo decoding and derandomizing on data of signaling information area within corresponding group
S2605
Demutlplex and store FIC data from derandomized data
S2606
End of subframe?
S2607
N
Y
Calculate TNoG value
S2608
Perform deinterleaving for FIC data of the stored subframe
S2609
Perform RS decoding
S2610
End

FIG. 48

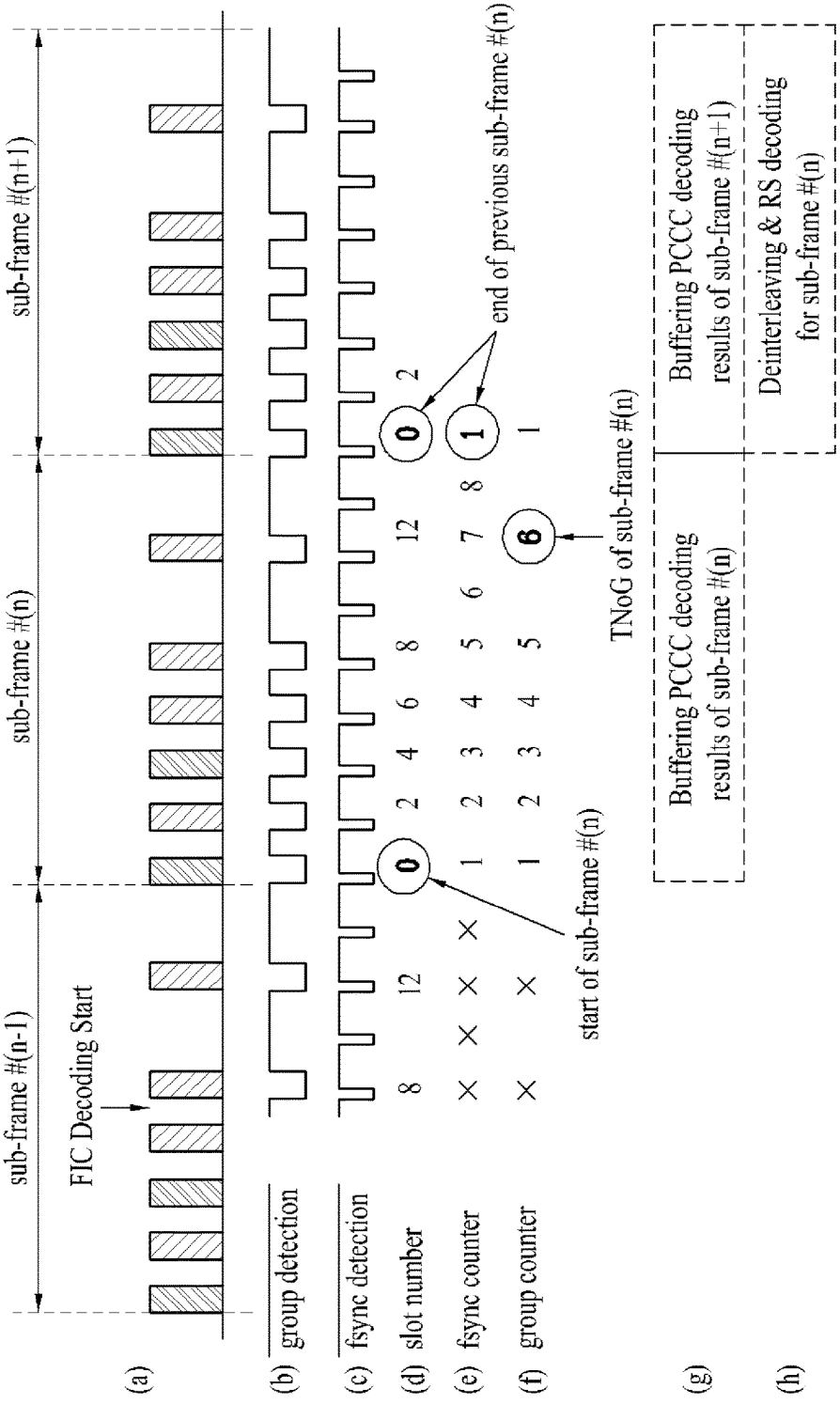

sub-frame #(n-1)
sub-frame #(n)
sub-frame #(n+1)
FIC Decoding Start
(a)
(b) group detection
(c) fsync detection
(d) slot number
(e) fsync counter
(f) group counter
8 12 0 2 4 6 8 12 0 2
× × × 1 2 3 4 5 6 7 8 1
× × 1 2 3 4 5 6 1
start of sub-frame #(n)
TNoG of sub-frame #(n)
end of previous sub-frame #(n)
Buffering PCCC decoding results of sub-frame #(n)
Buffering PCCC decoding results of sub-frame #(n+1)
(g)
Deinterleaving & RS decoding for sub-frame #(n)
(h)

FIG. 49

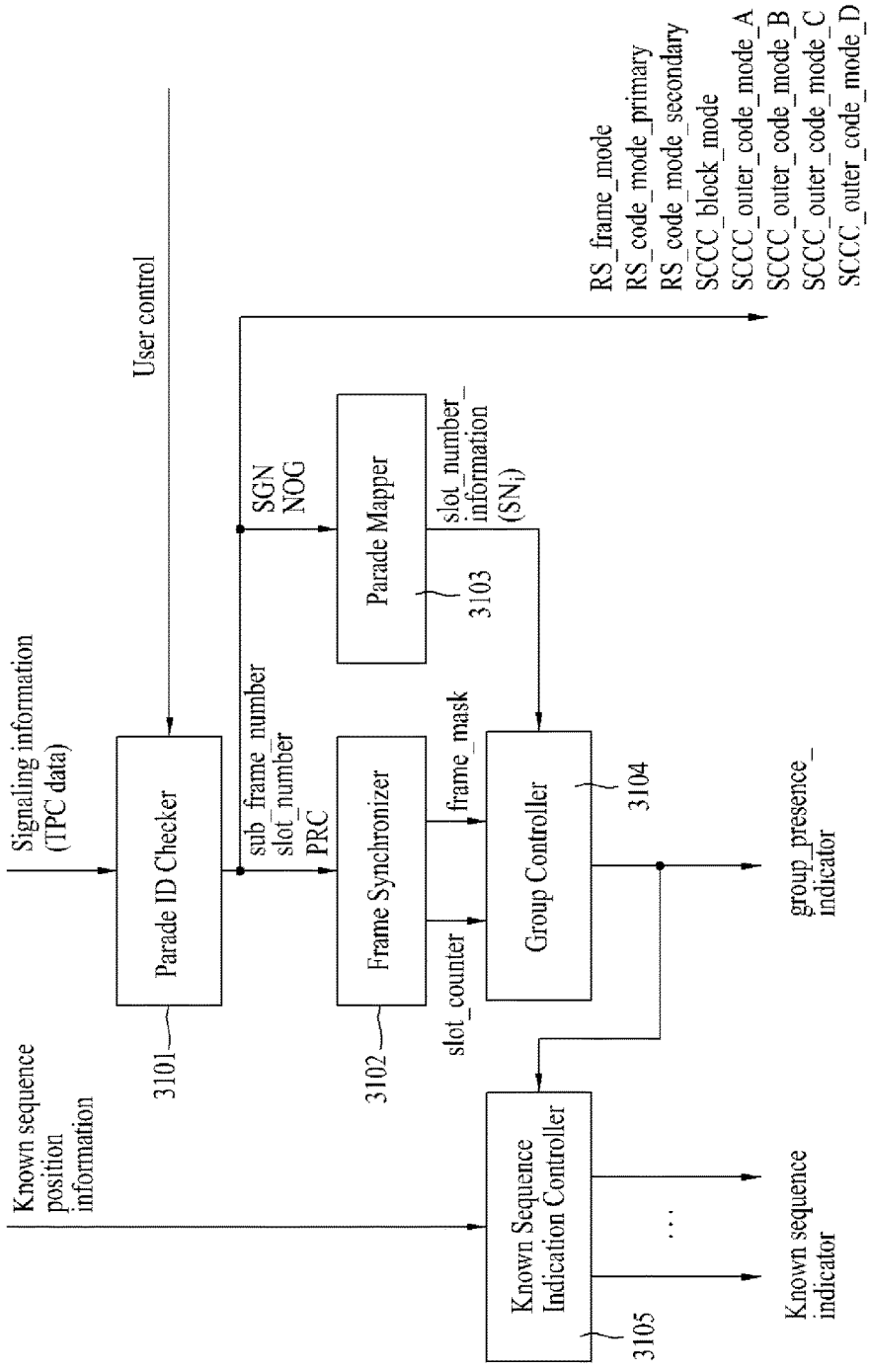

Signaling information (TPC data)
Known sequence position information
Parade ID Checker
3101
User control
sub_frame_number
slot_number
PRC
SGN
NOG
Parade Mapper
3103
slot_number_information (SN_l)
Frame Synchronizer
3102
slot_counter
frame_mask
Group Controller
3104
Known Sequence Indication Controller
3105
RS_frame_mode
RS_code_mode_primary
RS_code_mode_secondary
SCCC_block_mode
SCCC_outer_code_mode_A
SCCC_outer_code_mode_B
SCCC_outer_code_mode_C
SCCC_outer_code_mode_D
group_presence_indicator
Known sequence indicator

FIG. 50

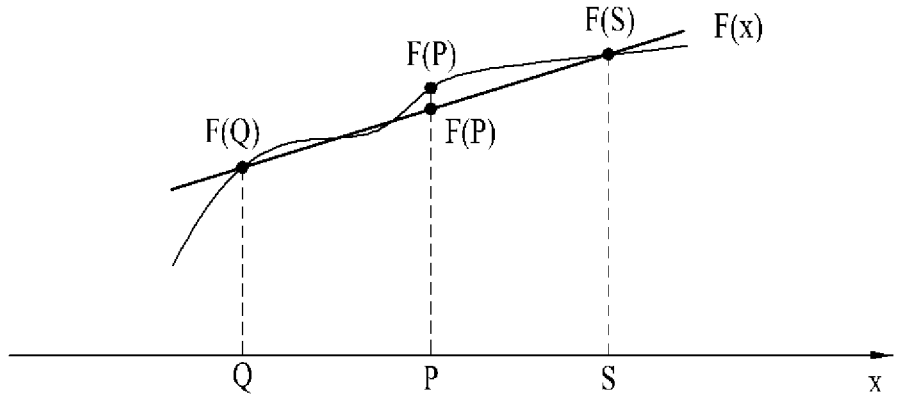


FIG. 51

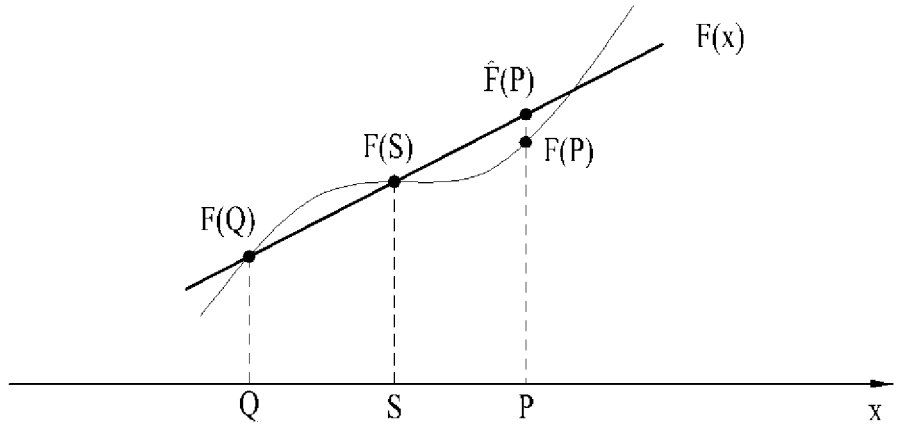


FIG. 52

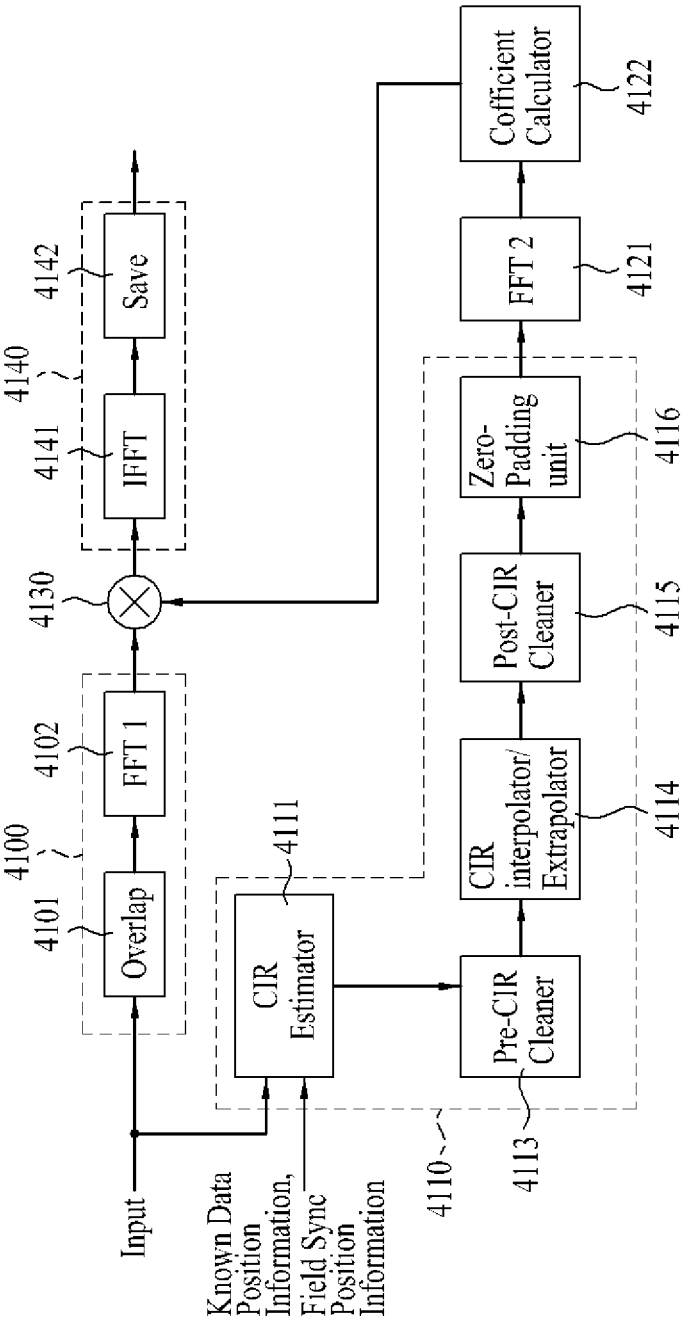

Input
Known Data Position Information, Field Sync Position Information
4110
4100
4101
Overlap
4102
FFT 1
4130
4140
4141
IFFT
4142
Save
4111
CIR Estimator
4113
Pre-CIR Cleaner
CIR interpolator/ Extrapolator
4114
Post-CIR Cleaner
4115
Zero-Padding unit
4116
FFT 2
4121
Cofficient Calculator
4122

FIG. 53

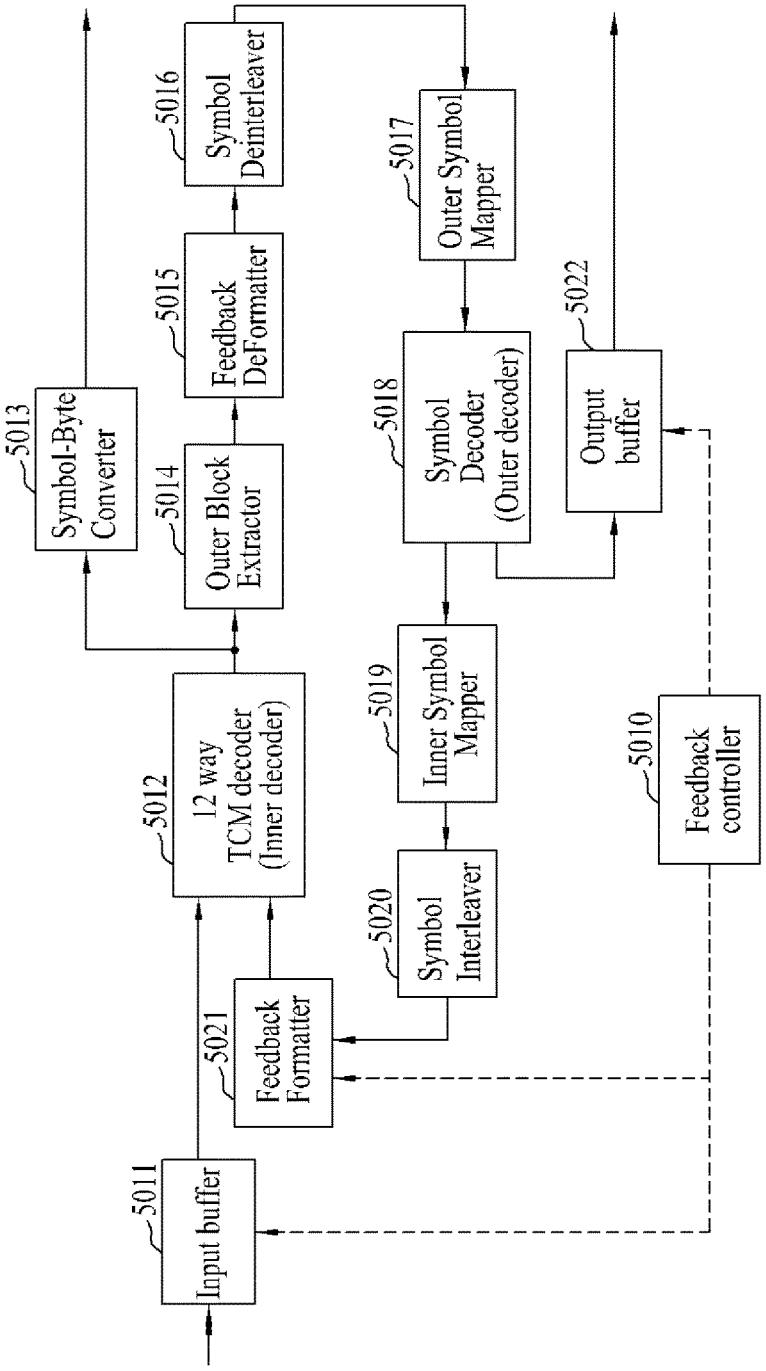

5011
Input buffer
5021
Feedback Formatter
5012
12 way TCM decoder (Inner decoder)
5013
Symbol-Byte Converter
5014
Outer Block Extractor
5015
Feedback DeFormatter
5016
Symbol Deinterleaver
5020
Symbol Interleaver
5019
Inner Symbol Mapper
5018
Symbol Decoder (Outer decoder)
5017
Outer Symbol Mapper
5022
Output buffer
5010
Feedback controller

FIG. 54

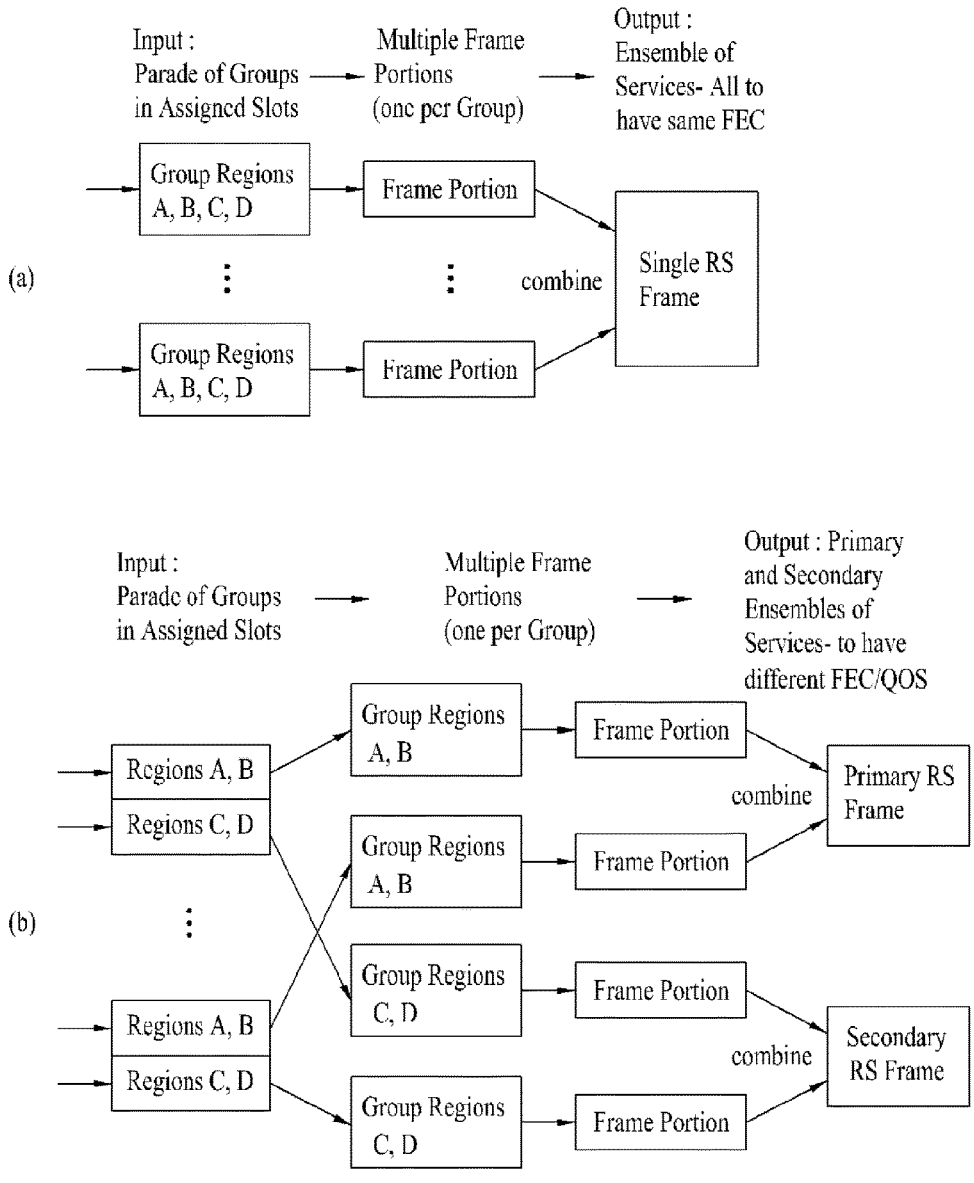

(a)
Input :
Parade of Groups
in Assigned Slots
Multiple Frame
Portions
(one per Group)
Output :
Ensemble of
Services- All to
have same FEC
Group Regions
A, B, C, D
Frame Portion
combine
Single RS
Frame
Group Regions
A, B, C, D
Frame Portion
(b)
Input :
Parade of Groups
in Assigned Slots
Multiple Frame
Portions
(one per Group)
Output : Primary
and Secondary
Ensembles of
Services- to have
different FEC/QOS
Regions A, B
Regions C, D
Group Regions
A, B
Frame Portion
combine
Primary RS
Frame
Group Regions
A, B
Frame Portion
Group Regions
C, D
Frame Portion
Regions A, B
Regions C, D
combine
Secondary
RS Frame
Group Regions
C, D
Frame Portion

FIG. 55

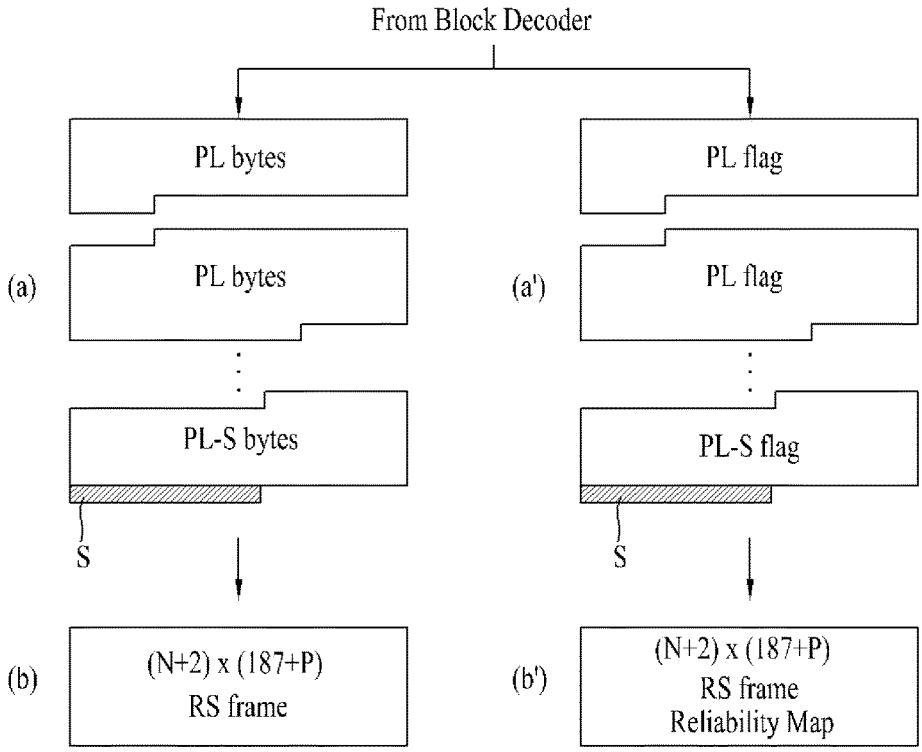
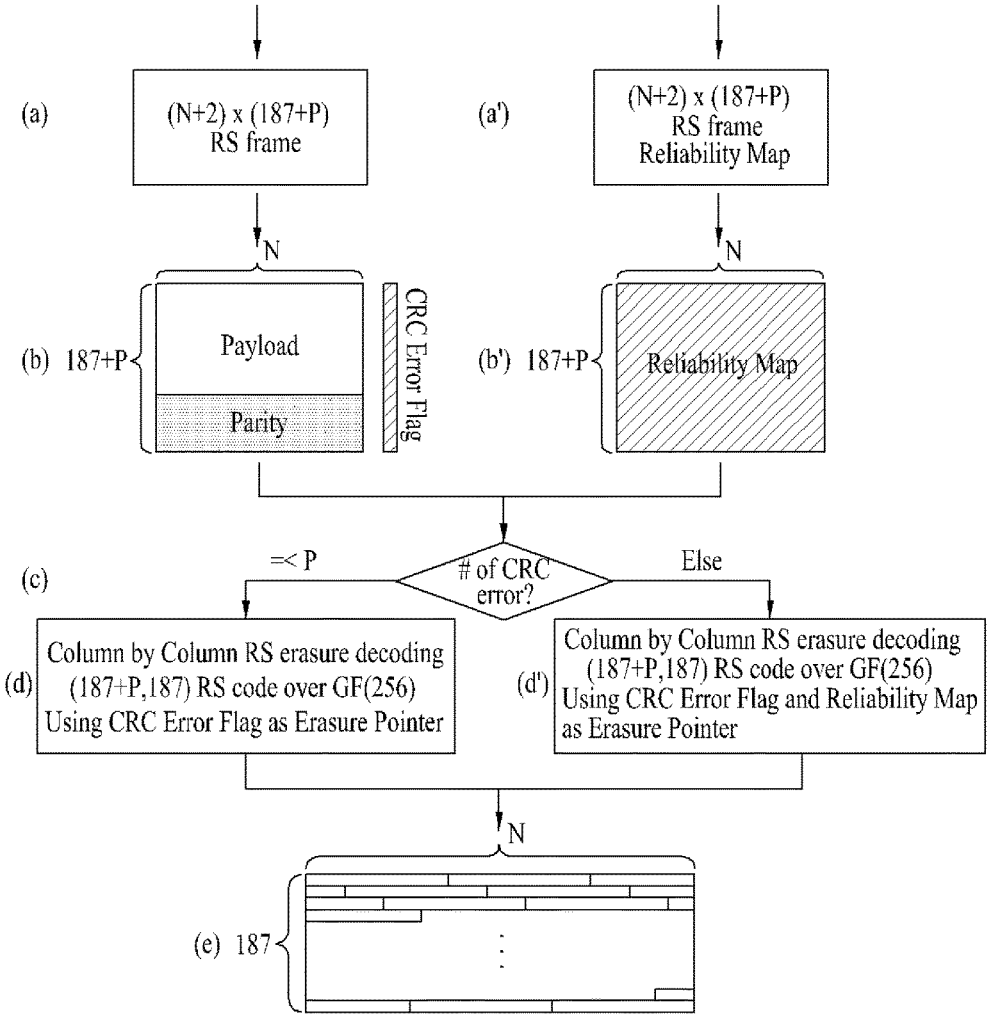

From Block Decoder
PL bytes
PL flag
(a)
PL bytes
(a')
PL flag
PL-S bytes
PL-S flag
S
S
(b)
(N+2) x (187+P)
RS frame
(b')
(N+2) x (187+P)
RS frame
Reliability Map

FIG. 56


(a)
(N+2) x (187+P) RS frame
(a')
(N+2) x (187+P) RS frame Reliability Map
N
N
(b) 187+P
Payload
Parity
CRC Error Flag
(b') 187+P
Reliability Map
(c)
=< P
# of CRC error?
Else
(d)
Column by Column RS erasure decoding (187+P,187) RS code over GF(256) Using CRC Error Flag as Erasure Pointer
(d')
Column by Column RS erasure decoding (187+P,187) RS code over GF(256) Using CRC Error Flag and Reliability Map as Erasure Pointer
N
(e) 187

TRANSMITTING/RECEIVING SYSTEM AND METHOD OF PROCESSING BROADCAST SIGNAL IN TRANSMITTING/RECEIVING SYSTEM

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue; a claim printed with strikethrough indicates that the claim was canceled, disclaimed, or held invalid by a prior post-patent action or proceeding.

This application *is a continuation reissue application of U.S. Reissue application Ser. No. 14/514,078, filed on Oct. 14, 2014, which is a reissue application of U.S. Pat. No. 8,503,347 B2, issued from U.S. application Ser. No. 12/751,974, filed on Mar. 31, 2010, which* claims the benefit of U.S. Provisional Application No. 61/165,500, filed on Mar. 31, 2009, *the contents of all of* which [is hereby incorporated] *are incorporated* by reference [as if fully set forth] herein *in their entireties.*

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a digital broadcasting system for transmitting and receiving a digital broadcast signal, and more particularly, to a transmitting system for processing and transmitting the digital broadcast signal, and a receiving system for receiving and processing the digital broadcast signal and, a method of processing data in the transmitting system and the receiving system.

2. Discussion of the Related Art

The Vestigial Sideband (VSB) transmission mode, which is adopted as the standard for digital broadcasting in North America and the Republic of Korea, is a system using a single carrier method. Therefore, the receiving performance of the digital broadcast receiving system may be deteriorated in a poor channel environment. Particularly, since resistance to changes in channels and noise is more highly required when using portable and/or mobile broadcast receivers, the receiving performance may be even more deteriorated when transmitting mobile service data by the VSB transmission mode.

SUMMARY OF THE INVENTION

Accordingly, the present invention is directed to a transmitting system and a receiving system and a method of processing broadcast signal that substantially obviate one or more problems due to limitations and disadvantages of the related art.

An object of the present invention is to provide a transmitting system and a receiving system and a method of processing broadcast signal that are highly resistant to channel changes and noise.

Another object of the present invention is to provide a transmitting system and a receiving system and a method of processing broadcast signal that can enhance the receiving performance of the receiving system by performing additional encoding on mobile service data and by transmitting the processed data to the receiving system.

A further object of the present invention is to provide a transmitting system and a receiving system and a method of processing broadcast signal that can also enhance the receiving performance of the receiving system by inserting known data already known in accordance with a pre-agreement between the receiving system and the transmitting system in a predetermined region within a data region.

A further aspect of the present invention is to provide a transmitting system and a receiving system and a method of processing broadcast signal that can enhance the decoding performance of signaling data included in a data group.

Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objectives and other advantages of the invention may be realized and attained by the structure particularly pointed out in the written description and claims hereof as well as the appended drawings.

To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, a method of processing a broadcast signal in a receiving system may include receiving a broadcast signal including a data group, the data group comprising mobile service data, regularly spaced known data sequences, and signaling data, the signaling data including fast information channel (FIC) data and transmission parameter channel (TPC) data, the FIC data including information for fast mobile service acquisition, and wherein the TPC data including a number of a slot through which the data group is transmitted and version information of the FIC data, channel equalizing the broadcast signal by using at least one of the known data sequences, performing turbo decoding for the signaling data included in the channel equalized broadcast signal, demultiplexing the TPC data and the FIC data from the turbo-decoded signaling data, performing first error correction decoding for the demultiplexed TPC data, performing block deinterleaving for the demultiplexed FIC data in a block unit of TNoG×51 bytes, herein TNoG equals to the number of all data groups assigned to one subframe, and performing second error correction decoding for the block deinterleaved FIC data.

The signaling data may be assigned to a signaling information area located between first known data sequence and second known data sequence of the known data sequences.

The step of performing block deinterleaving may include acquiring TNoG information from the error correction decoded TPC data, and performing block deinterleaving using the acquired TNoG information.

A time period where the broadcast signal is received is an M/H frame, which includes five subframes, each subframe includes 16 slots, and N ($16 \geq N \geq 0$) number data groups are received through the subframe, the N number of data groups being respectively received through N number of slots.

The step of performing block deinterleaving may include detecting a start of next subframe if FIC decoding start command is input, counting the number of data groups transmitted to the subframe of which start is detected, detecting an end of the subframe, acquiring TNoG information from the number of the counted data groups if the end of the subframe is detected, and performing block deinterleaving using the acquired TNoG information.

The start of the subframe detects by using a slot number included in the error correction decoded TPC data.

The end of the subframe detects by using a slot number included in the error correction decoded TPC data.

The end of the subframe detects by counting the number of field synchronization transmitted to the subframe.

In another aspect of the present invention, a receiving system may include a tuner, a channel equalizer, a turbo decoder, a demultiplexer, a first error correction decoder, a block deinterleaver, and a second error correction decoder. The tuner receives a broadcast signal including a data group. The data group comprises mobile service data, regularly spaced known data sequences, and signaling data. The signaling data includes fast information channel (FIC) data and transmission parameter channel (TPC) data, the FIC data including information for fast mobile service acquisition, and the TPC data including a number of a slot through which the data group is transmitted and version information of the FIC data. The channel equalizer performs channel equalizing on the broadcast signal by using at least one of the known data sequences. The turbo decoder performs turbo decoding for the signaling data included in the channel equalized broadcast signal. The demultiplexer demultiplexes the TPC data and the FIC data from the turbo-decoded signaling data. The first error correction decoder performs first error correction decoding for the demultiplexed TPC data. The block deinterleaver performs block deinterleaving on the demultiplexed FIC data in a block unit of TNoG×51 bytes. Herein, TNoG equals to the number of all data groups assigned to one sub-frame. The second error correction decoder performs second error correction decoding for the block deinterleaved FIC data.

It is to be understood that both the foregoing general description and the following detailed description of the present invention are exemplary and explanatory and are intended to provide further explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this application, illustrate embodiment(s) of the invention and together with the description serve to explain the principle of the invention. In the drawings:

FIG. **1** illustrates a structure of a M/H frame for transmitting and receiving mobile service data according to the present invention;

FIG. **2** illustrates an exemplary structure of a VSB frame;

FIG. **3** illustrates a mapping example of the positions to which the first 4 slots of a sub-frame are assigned with respect to a VSB frame in a space region;

FIG. **4** illustrates a mapping example of the positions to which the first 4 slots of a sub-frame are assigned with respect to a VSB frame in a time region;

FIG. **5** illustrates an alignment of data after being data interleaved and identified;

FIG. **6** illustrates an enlarged portion of the data group shown in FIG. **5** for a better understanding of the present invention;

FIG. **7** illustrates an alignment of data before being data interleaved and identified;

FIG. **8** illustrates an enlarged portion of the data group shown in FIG. **7** for a better understanding of the present invention;

FIG. **9** illustrates an exemplary assignment order of data groups being assigned to one of 5 sub-frames according to the present invention;

FIG. **10** illustrates an example of assigning a single parade to an M/H frame according to the present invention;

FIG. **11** illustrates an example of assigning 3 parades to an M/H frame according to the present invention;

FIG. **12** illustrates an example of expanding the assignment process of 3 parades to 5 sub-frames within an M/H frame;

FIG. **13** illustrates a data transmission structure according to an embodiment of the present invention, wherein signaling data are included in a data group so as to be transmitted;

FIG. **14** illustrates a block diagram showing a general structure of a transmitting system according to an embodiment of the present invention;

FIG. **15** is a diagram illustrating an example of RS frame payload according to the present invention;

FIG. **16** is a diagram illustrating a structure of an M/H header within an M/H service data packet according to the present invention;

FIG. **17**(a) and FIG. **17**(b) are diagrams illustrating another example of RS frame payload according to the present invention; and

FIG. **18** illustrates a block diagram showing an example of a service multiplexer of FIG. **14**;

FIG. **19** illustrates a block diagram showing an embodiment of a transmitter of FIG. **14**;

FIG. **20** illustrates a block diagram showing an example of a pre-processor of FIG. **19**;

FIG. **21** illustrates a conceptual block diagram of the M/H frame encoder of FIG. **20**;

FIG. **22** illustrates a detailed block diagram of an RS frame encoder of FIG. **21**;

FIG. **23**(a) and FIG. **23**(b) illustrate a process of one or two RS frame being divided into several portions, based upon an RS frame mode value, and a process of each portion being assigned to a corresponding region within the respective data group;

FIG. **24**(a) to FIG. **24**(c) illustrate error correction encoding and error detection encoding processes according to an embodiment of the present invention;

FIG. **25**(a) to FIG. **25**(d) illustrate an example of performing a row permutation (or interleaving) process in super frame units according to the present invention;

FIG. **26**(a) and FIG. **26**(b) illustrate an example which a parade consists of two RS frames

FIG. **27**(a) and FIG. **27**(b) illustrate an exemplary process of dividing an RS frame for configuring a data group according to the present invention;

FIG. **28** illustrates a block diagram of a block processor according to an embodiment of the present invention;

FIG. **29** illustrates a detailed block diagram of a convolution encoder of the block processor;

FIG. **30** illustrates a symbol interleaver of the block processor;

FIG. **31** illustrates a block diagram of a group formatter according to an embodiment of the present invention;

FIG. **32** illustrates a block diagram of a trellis encoder according to an embodiment of the present invention;

FIG. **33** illustrates an example of assigning signaling information area according to an embodiment of the present invention;

FIG. **34** illustrates a detailed block diagram of a signaling encoder according to the present invention;

FIG. **35** illustrates a detail block diagram of an iterative turbo encoder according to an embodiment of the present invention;

FIG. **36**(a) illustrates a detail block diagram of an even component encoder according to an embodiment of the present invention;

FIG. **36**(b) illustrates a detail block diagram of an odd component encoder according to an embodiment of the present invention;

FIG. **37**(a) illustrates an exemplary case in which a trellis encoder is serially concatenated with the even component encoder shown in FIG. **36**(a);

FIG. **37**(b) illustrates an exemplary case in which a trellis encoder is serially concatenated with the odd component encoder shown in FIG. **36**(b);

FIG. **38** illustrates an example of a syntax structure of TPC data according to the present invention;

FIG. **39** illustrates an example of a transmission scenario of the TPC data and the FIC data level according to the present invention;

FIG. **40** illustrates an example of power saving of in a receiver when transmitting 3 parades to an M/H frame level according to the present invention;

FIG. **41** illustrates an example of a training sequence at the byte level before trellis encoding according to the present invention;

FIG. **42** illustrates an example of a training sequence at the symbol level after trellis encoding according to the present invention;

FIG. **43** illustrates a block diagram of a receiving system according to an embodiment of the present invention;

FIG. **44** is a block diagram showing an example of a demodulating unit in the receiving system;

FIG. **45** illustrates a detail block diagram of a signaling decoder according to an embodiment of the present invention;

FIG. **46** illustrates a detail block diagram of an iterative turbo decoder according to an embodiment of the present invention;

FIG. **47** illustrates a flow chart illustrating a process of decoding FIC data without using TPC data in accordance with one embodiment of the present invention;

FIG. **48** illustrates a detailed embodiment of a process of extracting TNoG in accordance with the present invention;

FIG. **49** is a block diagram showing an example of an operation controller of FIG. **44**;

FIG. **50** illustrates an example of linear interpolation according to the present invention;

FIG. **51** illustrates an example of linear extrapolation according to the present invention;

FIG. **52** illustrates a block diagram of a channel equalizer according to an embodiment of the present invention;

FIG. **53** illustrates a block diagram of a block decoder according to an embodiment of the present invention;

FIG. **54**(a) and FIG. **54**(b) illustrate an exemplary process of configuring one or two RS frame by collecting a plurality of portions according to the present invention; and

FIG. **55** and FIG. **56** illustrate process steps of error correction decoding according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts. In addition, although the terms used in the present invention are selected from generally known and used terms, some of the terms mentioned in the description of the present invention have been selected by the applicant at his or her discretion, the detailed meanings of which are described in relevant parts of the description herein. Furthermore, it is required that the present invention is understood, not simply by the actual terms used but by the meaning of each term lying within.

Among the terms used in the description of the present invention, main service data correspond to data that can be received by a fixed receiving system and may include audio/video (A/V) data. More specifically, the main service data may include A/V data of high definition (HD) or standard definition (SD) levels and may also include diverse data types required for data broadcasting. Also, the known data correspond to data pre-known in accordance with a pre-arranged agreement between the receiving system and the transmitting system.

Additionally, among the terms used in the present invention, "M/H (or MH)" corresponds to the initials of "mobile" and "handheld" and represents the opposite concept of a fixed-type system. Furthermore, the M/H service data may include at least one of mobile service data and handheld service data, and will also be referred to as "mobile service data" for simplicity. Herein, the mobile service data not only correspond to M/H service data but may also include any type of service data with mobile or portable characteristics. Therefore, the mobile service data according to the present invention are not limited only to the M/H service data.

The above-described mobile service data may correspond to data having information, such as program execution files, stock information, and so on, and may also correspond to A/V data. Most particularly, the mobile service data may correspond to A/V data having lower resolution and lower data rate as compared to the main service data. For example, if an A/V codec that is used for a conventional main service corresponds to a MPEG-2 codec, a MPEG-4 advanced video coding (AVC) or scalable video coding (SVC) having better image compression efficiency may be used as the A/V codec for the mobile service. Furthermore, any type of data may be transmitted as the mobile service data. For example, transport protocol expert group (TPEG) data for broadcasting real-time transportation information may be transmitted as the main service data.

Also, a data service using the mobile service data may include weather forecast services, traffic information services, stock information services, viewer participation quiz programs, real-time polls and surveys, interactive education broadcast programs, gaming services, services providing information on synopsis, character, background music, and filming sites of soap operas or series, services providing information on past match scores and player profiles and achievements, and services providing information on product information and programs classified by service, medium, time, and theme enabling purchase orders to be processed. Herein, the present invention is not limited only to the services mentioned above.

In the present invention, the transmitting system provides backward compatibility in the main service data so as to be received by the conventional receiving system. Herein, the main service data and the mobile service data are multiplexed to the same physical channel and then transmitted.

Furthermore, the transmitting system according to the present invention performs additional encoding on the mobile service data and inserts the data already known by the receiving system and transmitting system (e.g., known data), thereby transmitting the processed data.

Therefore, when using the transmitting system according to the present invention, the receiving system may receive the mobile service data during a mobile state and may also receive the mobile service data with stability despite various distortion and noise occurring within the channel.

M/H Frame Structure

In the embodiment of the present invention, the mobile service data are first multiplexed with main service data in M/H frame units and, then, modulated in a VSB mode and transmitted to the receiving system.

At this point, one M/H frame consists of K1 number of sub-frames, wherein one sub-frame includes K2 number of slots. Also, each slot may be configured of K3 number of data packets. In the embodiment of the present invention, K1 will be set to 5, K2 will be set to 16, and K3 will be set to 156 (i.e., K1=5, K2=16, and K3=156). The values for K1, K2, and K3 presented in this embodiment either correspond to values according to a preferred embodiment or are merely exemplary. Therefore, the above-mentioned values will not limit the scope of the present invention.

FIG. **1** illustrates a structure of an M/H frame for transmitting and receiving mobile service data according to the present invention. In the example shown in FIG. **1**, one M/H frame consists of 5 sub-frames, wherein each sub-frame includes 16 slots. In this case, the M/H frame according to the present invention includes 5 sub-frames and 80 slots. Also, in a packet level, one slot is configured of 156 data packets (i.e., transport stream packets), and in a symbol level, one slot is configured of 156 data segments. Herein, the size of one slot corresponds to one half (½) of a VSB field. More specifically, since one 207-byte data packet has the same amount of data as a data segment, a data packet prior to being interleaved may also be used as a data segment.

At this point, two VSB fields are grouped to form a VSB frame.

FIG. **2** illustrates an exemplary structure of a VSB frame, wherein one VSB frame consists of 2 VSB fields (i.e., an odd field and an even field). Herein, each VSB field includes a field synchronization segment and 312 data segments.

The slot corresponds to a basic time period for multiplexing the mobile service data and the main service data. Herein, one slot may either include the mobile service data or be configured only of the main service data.

If one M/H frame is transmitted during one slot, the first 118 data packets within the slot correspond to a data group. And, the remaining 38 data packets become the main service data packets. In another example, when no data group exists in a slot, the corresponding slot is configured of 156 main service data packets.

Meanwhile, when the slots are assigned to a VSB frame, an offset exists for each assigned position.

FIG. **3** illustrates a mapping example of the positions to which the first 4 slots of a sub-frame are assigned with respect to a VSB frame in a space region. And, FIG. **4** illustrates a mapping example of the positions to which the first 4 slots of a sub-frame are assigned with respect to a VSB frame in a time region.

Referring to FIG. **3** and FIG. **4**, a 38^{th} data packet (TS packet #**37**) of a 1^{st} slot (Slot #**0**) is mapped to the 1^{st} data packet of an odd VSB field. A 38^{th} data packet (TS packet #**37**) of a 2^{nd} slot (Slot #**1**) is mapped to the 157^{th} data packet of an odd VSB field. Also, a 38^{th} data packet (TS packet #**37**) of a 3^{rd} slot (Slot #**2**) is mapped to the 1^{st} data packet of an even VSB field. And, a 38^{th} data packet (TS packet #**37**) of a 4^{th} slot (Slot #**3**) is mapped to the 157^{th} data packet of an even VSB field. Similarly, the remaining 12 slots within the corresponding sub-frame are mapped in the subsequent VSB frames using the same method.

Meanwhile, one data group may be divided into at least one or more hierarchical regions. And, depending upon the characteristics of each hierarchical region, the type of mobile service data being inserted in each region may vary. For example, the data group within each region may be divided (or categorized) based upon the receiving performance.

In an example given in the present invention, a data group is divided into regions A, B, C, and D in a data configuration after data interleaving.

FIG. **5** illustrates an alignment of data after being data interleaved and identified. FIG. **6** illustrates an enlarged portion of the data group shown in FIG. **5** for a better understanding of the present invention. FIG. **7** illustrates an alignment of data before being data interleaved and identified. And, FIG. **8** illustrates an enlarged portion of the data group shown in FIG. **7** for a better understanding of the present invention. More specifically, a data structure identical to that shown in FIG. **5** is transmitted to a receiving system. In other words, one data packet is data-interleaved so as to be scattered to a plurality of data segments, thereby being transmitted to the receiving system. FIG. **5** illustrates an example of one data group being scattered to 170 data segments. At this point, since one 207-byte packet has the same amount of data as one data segment, the packet that is not yet processed with data-interleaving may be used as the data segment.

FIG. **5** shows an example of dividing a data group prior to being data-interleaved into 10 M/H blocks (i.e., M/H block **1** (B**1**) to M/H block **10** (B**10**)). In this example, each M/H block has the length of 16 segments. Referring to FIG. **5**, only the RS parity data are allocated to a portion of 5 segments before the M/H block **1** (B**1**) and 5 segments behind the M/H block **10** (B**10**). The RS parity data are excluded in regions A to D of the data group.

More specifically, when it is assumed that one data group is divided into regions A, B, C, and D, each M/H block may be included in any one of region A to region D depending upon the characteristic of each M/H block within the data group. At this point, according to an embodiment of the present invention, each M/H block may be included in any one of region A to region D based upon an interference level of main service data.

Herein, the data group is divided into a plurality of regions to be used for different purposes. More specifically, a region of the main service data having no interference or a very low interference level may be considered to have a more resistant (or stronger) receiving performance as compared to regions having higher interference levels. Additionally, when using a system inserting and transmitting known data in the data group, wherein the known data are known based upon an agreement between the transmitting system and the receiving system, and when consecutively long known data are to be periodically inserted in the mobile service data, the known data having a predetermined length may be periodically inserted in the region having no interference from the main service data (i.e., a region wherein the main service data are not mixed). However, due to interference from the main service data, it is difficult to periodically insert known data and also to insert consecutively long known data to a region having interference from the main service data.

Referring to FIG. **5**, M/H block **4** (B**4**) to M/H block **7** (B**7**) correspond to regions without interference of the main service data. M/H block **4** (B**4**) to M/H block **7** (B**7**) within the data group shown in FIG. **5** correspond to a region where no interference from the main service data occurs. In this example, a long known data sequence is inserted at both the beginning and end of each M/H block. In the description of the present invention, the region including M/H block **4** (B**4**)

to M/H block **7** (B**7**) will be referred to as "region A (=B**4**+B**5**+B**6**+B**7**)". As described above, when the data group includes region A having a long known data sequence inserted at both the beginning and end of each M/H block, the receiving system is capable of performing equalization by using the channel information that can be obtained from the known data. Therefore, the strongest equalizing performance may be yielded (or obtained) from one of region A to region D.

In the example of the data group shown in FIG. **5**, M/H block **3** (B**3**) and M/H block **8** (B**8**) correspond to a region having little interference from the main service data. Herein, a long known data sequence is inserted in only one side of each M/H block B**3** and B**8**. More specifically, due to the interference from the main service data, a long known data sequence is inserted at the end of M/H block **3** (B**3**), and another long known data sequence is inserted at the beginning of M/H block **8** (B**8**). In the present invention, the region including M/H block **3** (B**3**) and M/H block **8** (B**8**) will be referred to as "region B(=B**3**+B**8**)". As described above, when the data group includes region B having a long known data sequence inserted at only one side (beginning or end) of each M/H block, the receiving system is capable of performing equalization by using the channel information that can be obtained from the known data. Therefore, a stronger equalizing performance as compared to region C/D may be yielded (or obtained).

Referring to FIG. **5**, M/H block **2** (B**2**) and M/H block (B**9**) correspond to a region having more interference from the main service data as compared to region B. A long known data sequence cannot be inserted in any side of M/H block **2** (B**2**) and M/H block **9** (B**9**). Herein, the region including M/H block **2** (B**2**) and M/H block **9** (B**9**) will be referred to as "region C(=B**2**+B**9**)". Finally, in the example shown in FIG. **5**, M/H block **1** (B**1**) and M/H block **10** (B**10**) correspond to a region having more interference from the main service data as compared to region C. Similarly, a long known data sequence cannot be inserted in any side of M/H block **1** (B**1**) and M/H block **10** (B**10**).

Herein, the region including M/H block **1** (B**1**) and M/H block **10** (B**10**) will be referred to as "region D (=B**1**+B**10**)". Since region C/D is spaced further apart from the known data sequence, when the channel environment undergoes frequent and abrupt changes, the receiving performance of region C/D may be deteriorated.

FIG. **7** illustrates a data structure prior to data interleaving. More specifically, FIG. **7** illustrates an example of 118 data packets being allocated to a data group. FIG. **7** shows an example of a data group consisting of 118 data packets, wherein, based upon a reference packet (e.g., a 1st packet (or data segment) or 157th packet (or data segment) after a field synchronization signal), when allocating data packets to a VSB frame, 37 packets are included before the reference packet and 81 packets (including the reference packet) are included afterwards.

In other words, with reference to FIG. **5**, a field synchronization signal is placed (or assigned) between M/H block **2** (B**2**) and M/H block **3** (B**3**). Accordingly, this indicates that the slot has an off-set of 37 data packets with respect to the corresponding VSB field.

The size of the data groups, number of hierarchical regions within the data group, the size of each region, the number of M/H blocks included in each region, the size of each M/H block, and so on described above are merely exemplary. Therefore, the present invention will not be limited to the examples described above.

FIG. **9** illustrates an exemplary assignment order of data groups being assigned to one of 5 sub-frames, wherein the 5 sub-frames configure an M/H frame. For example, the method of assigning data groups may be identically applied to all M/H frames or differently applied to each M/H frame. Furthermore, the method of assigning data groups may be identically applied to all sub-frames or differently applied to each sub-frame. At this point, when it is assumed that the data groups are assigned using the same method in all sub-frames of the corresponding M/H frame, the total number of data groups being assigned to an M/H frame is equal to a multiple of '5'.

According to the embodiment of the present invention, a plurality of consecutive data groups is assigned to be spaced as far apart from one another as possible within the M/H frame. Thus, the system can be capable of responding promptly and effectively to any burst error that may occur within a sub-frame.

For example, when it is assumed that 3 data groups are assigned to a sub-frame, the data groups are assigned to a 1st slot (Slot #**0**), a 5th slot (Slot #**4**), and a 9th slot (Slot #**8**) in the sub-frame, respectively. FIG. **9** illustrates an example of assigning 16 data groups in one sub-frame using the above-described pattern (or rule). In other words, each data group is serially assigned to 16 slots corresponding to the following numbers: 0, 8, 4, 12, 1, 9, 5, 13, 2, 10, 6, 14, 3, 11, 7, and 15. Equation 1 below shows the above-described rule (or pattern) for assigning data groups in a sub-frame.

$$j=(4i+0)\bmod 16 \qquad \text{Equation 1}$$

Herein,
0=0 if i<4,
0=2 else if i<8,
0=1 else if i<12,
0=3 else.

Herein, j indicates the slot number within a sub-frame. The value of j may range from 0 to 15 (i.e., $0 \le j \le 15$). Also, value of i indicates the data group number. The value of i may range from 0 to 15 (i.e., $0 \le i \le 15$).

In the present invention, a collection of data groups included in an M/H frame will be referred to as a "parade". Based upon the RS frame mode, the parade transmits data of at least one specific RS frame.

The mobile service data within one RS frame may be assigned either to all of regions A/B/C/D within the corresponding data group, or to at least one of regions A/B/C/D. In the embodiment of the present invention, the mobile service data within one RS frame may be assigned either to all of regions A/B/C/D, or to at least one of regions A/B and regions C/D. If the mobile service data are assigned to the latter case (i.e., one of regions A/B and regions C/D), the RS frame being assigned to regions A/B and the RS frame being assigned to regions C/D within the corresponding data group are different from one another.

In the description of the present invention, the RS frame being assigned to regions A/B within the corresponding data group will be referred to as a "primary RS frame", and the RS frame being assigned to regions C/D within the corresponding data group will be referred to as a "secondary RS frame", for simplicity. Also, the primary RS frame and the secondary RS frame form (or configure) one parade. More specifically, when the mobile service data within one RS frame are assigned either to all of regions A/B/C/D within the corresponding data group, one parade transmits one RS frame. In this case, also the RS frame will be referred to as a "primary RS frame". Conversely, when the mobile service

data within one RS frame are assigned either to at least one of regions A/B and regions C/D, one parade may transmit up to 2 RS frames.

More specifically, the RS frame mode indicates whether a parade transmits one RS frame, or whether the parade transmits two RS frames. Table 1 below shows an example of the RS frame mode.

TABLE 1

RS frame mode (2 bits)	Description
00	There is only one primary RS frame for all group regions
01	There are two separate RS frames. Primary RS frame for group regions A and B Secondary RS frame for group regions C and D
10	Reserved
11	Reserved

Table 1 illustrates an example of allocating 2 bits in order to indicate the RS frame mode. For example, referring to Table 1, when the RS frame mode value is equal to '00', this indicates that one parade transmits one RS frame. And, when the RS frame mode value is equal to '01', this indicates that one parade transmits two RS frames, i.e., the primary RS frame and the secondary RS frame. More specifically, when the RS frame mode value is equal to '01', data of the primary RS frame for regions A/B are assigned and transmitted to regions A/B of the corresponding data group. Similarly, data of the secondary RS frame for regions C/D are assigned and transmitted to regions C/D of the corresponding data group.

As described in the assignment of data groups, the parades are also assigned to be spaced as far apart from one another as possible within the sub-frame. Thus, the system can be capable of responding promptly and effectively to any burst error that may occur within a sub-frame.

Furthermore, the method of assigning parades may be identically applied to all sub-frames or differently applied to each sub-frame. According to the embodiment of the present invention, the parades may be assigned differently for each M/H frame and identically for all sub-frames within an M/H frame. More specifically, the M/H frame structure may vary by M/H frame units. Thus, an ensemble rate may be adjusted on a more frequent and flexible basis.

FIG. **10** illustrates an example of multiple data groups of a single parade being assigned (or allocated) to an M/H frame. More specifically, FIG. **10** illustrates an example of a plurality of data groups included in a single parade, wherein the number of data groups included in a sub-frame is equal to '3', being allocated to an M/H frame. Referring to FIG. **10**, 3 data groups are sequentially assigned to a sub-frame at a cycle period of 4 slots. Accordingly, when this process is equally performed in the 5 sub-frames included in the corresponding M/H frame, 15 data groups are assigned to a single M/H frame. Herein, the 15 data groups correspond to data groups included in a parade. Therefore, since one sub-frame is configured of 4 VSB frame, and since 3 data groups are included in a sub-frame, the data group of the corresponding parade is not assigned to one of the 4 VSB frames within a sub-frame.

For example, when it is assumed that one parade transmits one RS frame, and that a RS frame encoder located in a later block performs RS-encoding on the corresponding RS frame, thereby adding 24 bytes of parity data to the corresponding RS frame and transmitting the processed RS frame, the parity data occupy approximately 11.37% (=24/(187+24)×100) of the total code word length. Meanwhile, when one sub-frame includes 3 data groups, and when the data groups included in the parade are assigned, as shown in FIG. **10**, a total of 15 data groups form an RS frame. Accordingly, even when an error occurs in an entire data group due to a burst noise within a channel, the percentile is merely 6.67% (=1/15×100). Therefore, the receiving system may correct all errors by performing an erasure RS decoding process. More specifically, when the erasure RS decoding is performed, a number of channel errors corresponding to the number of RS parity bytes may be corrected. By doing so, the receiving system may correct the error of at least one data group within one parade. Thus, the minimum burst noise length correctable by a RS frame is over 1 VSB frame.

Meanwhile, when data groups of a parade are assigned as described above, either main service data may be assigned between each data group, or data groups corresponding to different parades may be assigned between each data group. More specifically, data groups corresponding to multiple parades may be assigned to one M/H frame.

Basically, the method of assigning data groups corresponding to multiple parades is very similar to the method of assigning data groups corresponding to a single parade. In other words, data groups included in other parades that are to be assigned to an M/H frame are also respectively assigned according to a cycle period of 4 slots.

At this point, data groups of a different parade may be sequentially assigned to the respective slots in a circular method. Herein, the data groups are assigned to slots starting from the ones to which data groups of the previous parade have not yet been assigned.

For example, when it is assumed that data groups corresponding to a parade are assigned as shown in FIG. **10**, data groups corresponding to the next parade may be assigned to a sub-frame starting either from the 12^{th} slot of a sub-frame. However, this is merely exemplary. In another example, the data groups of the next parade may also be sequentially assigned to a different slot within a sub-frame at a cycle period of 4 slots starting from the 3^{rd} slot.

FIG. **11** illustrates an example of transmitting 3 parades (Parade #**0**, Parade #**1**, and Parade #**2**) to an M/H frame. More specifically, FIG. **11** illustrates an example of transmitting parades included in one of 5 sub-frames, wherein the 5 sub-frames configure one M/H frame.

When the 1^{st} parade (Parade #**0**) includes 3 data groups for each sub-frame, the positions of each data groups within the sub-frames may be obtained by substituting values '0' to '2' for i in Equation 1. More specifically, the data groups of the 1^{st} parade (Parade #**0**) are sequentially assigned to the 1^{st}, 5^{th}, and 9^{th} slots (Slot #**0**, Slot #**4**, and Slot #**8**) within the sub-frame. Also, when the 2^{nd} parade includes 2 data groups for each sub-frame, the positions of each data groups within the sub-frames may be obtained by substituting values '3' and '4' for i in Equation 1.

More specifically, the data groups of the 2^{nd} parade (Parade #**1**) are sequentially assigned to the 2^{nd} and 12^{th} slots (Slot #**3** and Slot #**11**) within the sub-frame.

Finally, when the 3^{rd} parade includes 2 data groups for each sub-frame, the positions of each data groups within the sub-frames may be obtained by substituting values '5' and '6' for i in Equation 1. More specifically, the data groups of the 3^{rd} parade (Parade #**2**) are sequentially assigned to the 7^{th} and 11^{th} slots (Slot #**6** and Slot #**10**) within the sub-frame.

As described above, data groups of multiple parades may be assigned to a single M/H frame, and, in each sub-frame, the data groups are serially allocated to a group space having 4 slots from left to right. Therefore, a number of groups of one parade per sub-frame (NOG) may correspond to any one

integer from '1' to '8'. Herein, since one M/H frame includes 5 sub-frames, the total number of data groups within a parade that can be allocated to an M/H frame may correspond to any one multiple of '5' ranging from '5' to '40'.

FIG. **12** illustrates an example of expanding the assignment process of 3 parades, shown in FIGS. **11**, to 5 sub-frames within an M/H frame.

FIG. **13** illustrates a data transmission structure according to an embodiment of the present invention, wherein signaling data are included in a data group so as to be transmitted.

As described above, an M/H frame is divided into 5 sub-frames. Data groups corresponding to a plurality of parades co-exist in each sub-frame. Herein, the data groups corresponding to each parade are grouped by M/H frame units, thereby configuring a single parade.

Three parades (Parade #**0**, Parade #**1**, Parade #**2**) also exist in one M/H frame of FIG. **13**. At this time, a part (e.g., 37 bytes/data group) of each data group is used to forward fast information channel (FIC) information of mobile service data, which is encoded separately from RS code. An FIC region within a signaling information area assigned to each data group constitutes one FIC segment.

Meanwhile, in this embodiment, a collection of services is defined by concept of M/H ensemble. One M/H ensemble has the same QoS, and is coded with the same FEC code. Also, the ensemble has unique identifier (i.e., ensemble id), and is a collection of consecutive RS frames having the same FEC code.

As shown in FIG. **13**, FIC segment corresponding to each data group describes service information of M/H ensemble to which corresponding data group belongs.

In other words, the transmitting/receiving system according to one embodiment of the present invention manages two data channels. One data channel is an RS frame data channel for contents transmission, and the other data channel is a fast information channel (FIC) for service acquisition. The present invention is intended that mapping information between ensemble and mobile service is signaled using FIC chunk, which is split in a FIC segment unit and then transmitted through the FIC, whereby the receiving system can perform fast service acquisition.

General Description of the Transmitting System

FIG. **14** illustrates a block diagram showing a general structure of a digital broadcast transmitting system according to an embodiment of the present invention.

Herein, the digital broadcast transmitting includes a service multiplexer **100** and a transmitter **200**. Herein, the service multiplexer **100** is located in the studio of each broadcast station, and the transmitter **200** is located in a site placed at a predetermined distance from the studio. The transmitter **200** may be located in a plurality of different locations. Also, for example, the plurality of transmitters may share the same frequency. And, in this case, the plurality of transmitters receives the same signal. This corresponds to data transmission using Single Frequency Network (SFN). Accordingly, in the receiving system, a channel equalizer may compensate signal distortion, which is caused by a reflected wave, so as to recover the original signal. In another example, the plurality of transmitters may have different frequencies with respect to the same channel. This corresponds to data transmission using Multi Frequency Network (MFN).

A variety of methods may be used for data communication each of the transmitters, which are located in remote positions, and the service multiplexer. For example, an interface standard such as a synchronous serial interface for transport of MPEG-2 data (SMPTE-310M). In the SMPTE-310M interface standard, a constant data rate is decided as an output data rate of the service multiplexer. For example, in case of the 8VSB mode, the output data rate is 19.39 Mbps, and, in case of the 16VSB mode, the output data rate is 38.78 Mbps. Furthermore, in the conventional 8VSB mode transmitting system, a transport stream (TS) packet having a data rate of approximately 19.39 Mbps may be transmitted through a single physical channel. Also, in the transmitting system according to the present invention provided with backward compatibility with the conventional transmitting system, additional encoding is performed on the mobile service data. Thereafter, the additionally encoded mobile service data are multiplexed with the main service data to a TS packet form, which is then transmitted. At this point, the data rate of the multiplexed TS packet is approximately 19.39 Mbps.

At this point, the service multiplexer **100** receives at least one type of main service data and table information (e.g., PSI/PSIP table data) for each main service and encapsulates the received data into a transport stream (TS) packet.

Also, according to an embodiment of the present invention, the service multiplexer **100** receives at least one type of mobile service data and table information (e.g., PSI/PSIP table data) for each mobile service and encapsulates the received data into a transport stream (TS) packet.

According to another embodiment of the present invention, the service multiplexer **100** receives a RS frame (or RS frame payload), which is configured of at least one type of mobile service data and table information for each mobile service, and encapsulates the received RS frame data into mobile service data packets of a transport stream (TS) packet format.

And, the service multiplexer **100** multiplexes the encapsulated TS packets for main service and the encapsulated TS packets for mobile service based upon a predetermined multiplexing rule, thereby outputting the multiplexed TS packets to the transmitter **200**.

At this point, the RS frame payload (or RS frame) has the size of N (row)×187 (column), as shown in FIG. **15**. Herein, N represents the length of a row (i.e., number of columns), and 187 corresponds to the length of a column (i.e., number of rows).

In the present invention, for convenience of description, each row of the N bytes will be referred to as M/H service data packet (or M/H TP packet). The M/H service data packet includes M/H header of 2 bytes, a stuffing region of k bytes, and M/H payload of N-2-k bytes. At this time, k has a value of 0 or a value greater than 0. In this case, the M/H header of 2 bytes is only one example, and corresponding bytes can be varied depending on a designer. Accordingly, the present invention will not be limited to such example.

At this time, as the M/H service data packet includes M/H header, the M/H header may not reach N bytes.

In this case, stuffing bytes can be assigned to the remaining payload part of the corresponding M/H service data packet. For example, after program table information is assigned to one M/H service data packet, if the length of the M/H service data packet is N-20 bytes including the M/H header, the stuffing bytes can be assigned to the remaining 20 bytes. In this case, the value k becomes 20, and the M/H payload region within the corresponding M/H service data packet includes N-2-20 bytes.

The RS frame payload is generated by collecting signaling table information corresponding to one or more mobile services and/or IP datagram of the mobile service data. For example, signaling table information for two kinds of

mobile services called news (for example, IP datagram for mobile service **1**) and the stocks (for example, IP datagram for mobile service **2**) and IP datagram of mobile service data can be included in one frame payload.

More specifically, in the transmitting system (e.g., mobile broadcast station), the mobile service data (e.g., A/V steaming) are packetized based upon a real time protocol (RTP) method. The RTP packet is then packetized once again based upon a user datagram protocol (UDP) method. Thereafter, the RTP/UDP packet is in turn packetized based upon an IP method, thereby being packetized into RTP/UDP/IP packet data. In the description of the present invention, the packetized RTP/UDP/IP packet data will be referred to as an IP datagram for simplicity.

Furthermore, service information for receiving mobile services may be provided in the form of a signaling table. And, a service signaling channel transmitting such signaling table is packetized based upon a UDP method. And, the packetized UDP data are then packetized based upon an IP method, thereby being packetized into UDP/IP data. In the description of the present invention, the packetized UDP/IP packet data will also be referred to as an IP datagram for simplicity. According to an embodiment of the present invention, the service signaling channel is encapsulated into an IP datagram having a well-known destination IP address and a well-known destination UDP port number.

More specifically, one RS frame payload includes an IP datagram of mobile service data for at least one or more mobile services and an IP datagram of a service signaling channel for receiving the mobile service data.

According to the embodiment of the present invention, among a service map table (SMT), a guide access table (GAT), a cell information table (CIT), a service labeling table (SLT), and a rating region table (RRT), the present invention transmits at least one signaling table through the service signaling channel. Herein, the signaling tables presented in the embodiment of the present invention are merely examples for facilitating the understanding of the present invention. Therefore, the present invention is not limited only to the exemplary signaling tables that can be transmitted through the service signaling channel.

The SMT provides signaling information on ensemble levels. Also, each SMT provides IP access information for each mobile service belonging to the corresponding ensemble including each SMT. Furthermore, the SMT provides IP stream component level information required for the corresponding mobile service.

The RRT transmits information on region and consultation organs for program ratings. More specifically, the RRT provides content advisory rating information.

The GAT provides information on SG providers, which transmit the service guides. Also, the GAT provides service guide bootstrapping information required for accessing the SG. The CIT provides channel information of each cell, which corresponds to the frequency domain of a broadcast signal. Herein, a cell refers to a scope affected (or influenced) by a transmitter based upon a physical frequency in a multi-frequency network (MFN) environment (or condition). More specifically, the CIT provides information on a carrier wave frequency of an adjacent cell in the current transmitter (or transmitting system). Therefore, based upon the CIT information, a receiver (or receiving system) can travel from one transmitter's (or exciter's) coverage area to another.

The SLT provides minimum required information for an exclusive usage of a channel scan process. More specifically, according to the embodiment of the present invention, other than the SMT, by using the SLT for the exclusive usage of the channel scan process, so as to configure a set of minimum information for the channel scan process, the channel scanning speed may be increased.

According to an embodiment of the present invention, each signaling table is divided into at least one section. Then, each section is encapsulated to a UDP/IP header, thereby being transmitted through the service signaling channel. In this case, the number of UDP/IP packets being transmitted through the service signaling channel may vary based upon the number of signaling tables being transmitted through the service signaling channel and the number of sections in each signaling table.

At this point, all UDP/IP packets transmitted through the service signaling channel have the same number of well-known target IP addresses and well-known target UDP port numbers. For example, when it is assumed that the SMT, RRT, and GAT are transmitted through the service signaling channel, the target IP address and target UDP port number of all UDP/IP packets transmitting the SMT, RRT, and GAT are identical to one another. Furthermore, the target IP address and the target UDP port number respectively correspond to well-known values, i.e., values pre-known by the receiving system based upon an agreement between the receiving system and the transmitting system.

Therefore, the identification of each signaling table included in the service signaling data is performed by a table identifier. The table identifier may correspond to a table id field existing in the corresponding signaling table or in the header of the corresponding signaling table section. And, when required, identification may be performed by further referring to a table_id_extension field.

FIG. **16** is a diagram illustrating examples of fields allocated to the M/H header region within the M/H service data packet according to the present invention. Examples of the fields include type_indicator field, error_indicator field, stuff_indicator field, and pointer field.

The type_indicator field can allocate 3 bits, for example, and represents a type of data allocated to payload within the corresponding M/H service data packet. In other words, the type_indicator field indicates whether data of the payload is IP datagram or program table information. At this time, each data type constitutes one logical channel. In the logical channel which transmits the IP datagram, several mobile services are multiplexed and then transmitted. Each mobile service undergoes demultiplexing in the IP layer.

The error_indicator field can allocate 1 bit, for example, and represents whether the corresponding M/H service data packet has an error. For example, if the error_indicator field has a value of 0, it means that there is no error in the corresponding M/H service data packet. If the error_indicator field has a value of 1, it means that there may be an error in the corresponding M/H service data packet.

The stuff_indicator field can allocate 1 bit, for example, and represents whether stuffing byte exists in payload of the corresponding M/H service data packet. For example, if the stuff_indicator field has a value of 0, it means that there is no stuffing byte in the corresponding M/H service data packet. If the stuff_indicator field has a value of 1, it means that stuffing byte exists in the corresponding M/H service data packet.

The pointer field can allocate 11 bits, for example, and represents position information where new data (i.e., new signaling information or new IP datagram) starts in the corresponding M/H service data packet.

For example, if IP datagram for mobile service **1** and IP datagram for mobile service **2** are allocated to the first M/H

service data packet within the RS frame payload as illustrated in FIG. **15**, the pointer field value represents the start position of the IP datagram for mobile service **2** within the M/H service data packet.

Also, if there is no new data in the corresponding M/H service data packet, the corresponding field value is expressed as a maximum value exemplarily. According to the embodiment of the present invention, since 11 bits are allocated to the pointer field, if 2047 is expressed as the pointer field value, it means that there is no new data in the packet. The point where the pointer field value is 0 can be varied depending on the type_indicator field value and the stuff_indicator field value.

It is to be understood that the order, the position, and the meaning of the fields allocated to the header within the M/H service data packet illustrated in FIG. **16** are exemplarily illustrated for understanding of the present invention. Since the order, the position and the meaning of the fields allocated to the header within the M/H service data packet and the number of additionally allocated fields can easily be modified by those skilled in the art, the present invention will not be limited to the above example.

FIG. **17**(a) and FIG. **17**(b) illustrate another examples of RS frame payload according to the present invention. FIG. **17**(a) illustrates an example of primary RS frame payload to be allocated to regions A/B within the data group, and FIG. **17**(b) illustrates an example of secondary RS frame payload to be allocated to regions C/D within the data group.

In FIG. **17**(a) and FIG. **17**(b), a column length (i.e., the number of rows) of the RS frame payload to be allocated to the regions A/B and a column length (i.e., the number of rows) of the RS frame payload to be allocated to the regions C/D are 187 equally. However, row lengths (i.e, the number of columns) may be different from each other.

According to the embodiment of the present invention, when the row length of the primary RS frame payload to be allocated to the regions A/B within the data group is N1 bytes and the row length of the secondary RS frame payload to be allocated to the regions C/D within the data group is N2 bytes, a condition of N1>N2 is satisfied. In this case, N1 and N2 can be varied depending on the transmission parameter or a region of the data group, to which the corresponding RS frame payload will be transmitted.

For convenience of the description, each row of the N1 and N2 bytes will be referred to as the M/H service data packet. The M/H service data packet within the RS frame payload to be allocated to the regions A/B within the data group can be comprised of M/H header of 2 bytes, a stuffing region of k bytes, and M/H payload of N1-2-k bytes. At this time, k has a value of 0 or a value greater than 0. Also, the M/H service data packet within the RS frame payload to be allocated to the regions C/D within the data group can be comprised of M/H header of 2 bytes, a stuffing region of k bytes, and M/H payload of N2-2-k bytes. At this time, k has a value of 0 or a value greater than 0.

In the present invention, the primary RS frame payload for the regions A/B within the data group and the secondary RS frame payload for the regions C/D within the data group can include at least one of IP datagrams of signaling table information and mobile service data. Also, one RS frame payload can include IP datagram corresponding to one or more mobile services.

Corresponding parts of FIG. **15** can be applied to the other parts, which are not described in FIG. **17**(a) and FIG. **17**(b).

Meanwhile, the value of N, which corresponds to the number of columns within an RS frame payload, can be decided according to Equation 2.

$$N = \left\lfloor \frac{5 \times NoG \times PL}{187 + P} \right\rfloor - 2 \qquad \text{Equation 2}$$

Herein, NoG indicates the number of data groups assigned to a sub-frame. PL represents the number of SCCC payload data bytes assigned to a data group. And, P signifies the number of RS parity data bytes added to each column of the RS frame payload. Finally, $\lfloor x \rfloor$ is the greatest integer that is equal to or smaller than X.

More specifically, in Equation 2, PL corresponds to the length of an RS frame portion. The value of PL is equivalent to the number of SCCC payload data bytes that are assigned to the corresponding data group. Herein, the value of PL may vary depending upon the RS frame mode, SCCC block mode, and SCCC outer code mode. Table 2 to Table 5 below respectively show examples of PL values, which vary in accordance with the RS frame mode, SCCC block mode, and SCCC outer code mode. The SCCC block mode and the SCCC outer code mode will be described in detail in a later process.

TABLE 2

SCCC outer code mode				
for Region A	for Region B	for Region C	for Region D	PL
00	00	00	00	9624
00	00	00	01	9372
00	00	01	00	8886
00	00	01	01	8634
00	01	00	00	8403
00	01	00	01	8151
00	01	01	00	7665
00	01	01	01	7413
01	00	00	00	7023
01	00	00	01	6771
01	00	01	00	6285
01	00	01	01	6033
01	01	00	00	5802
01	01	00	01	5550
01	01	01	00	5064
01	01	01	01	4812
		Others		Reserved

Table 2 shows an example of the PL values for each data group within an RS frame, wherein each PL value varies depending upon the SCCC outer code mode, when the RS frame mode value is equal to '00', and when the SCCC block mode value is equal to '00'. For example, when it is assumed that each SCCC outer code mode value of regions A/B/C/D within the data group is equal to '00' (i.e., the block processor **302** of a later block performs encoding at a coding rate of 1/2), the PL value within each data group of the corresponding RS frame may be equal to 9624 bytes. More specifically, 9624 bytes of mobile service data within one RS frame may be assigned to regions A/B/C/D of the corresponding data group.

TABLE 3

SCCC outer code mode	PL
00	9624
01	4812
Others	Reserved

Table 3 shows an example of the PL values for each data group within an RS frame, wherein each PL value varies

depending upon the SCCC outer code mode, when the RS frame mode value is equal to '00', and when the SCCC block mode value is equal to '01'.

TABLE 4

SCCC outer code mode		
For Region A	for Region B	PL
00	00	7644
00	01	6423
01	00	5043
01	01	3822
Others		Reserved

Table 4 shows an example of the PL values for each data group within a primary RS frame, wherein each PL value varies depending upon the SCCC outer code mode, when the RS frame mode value is equal to '01', and when the SCCC block mode value is equal to '00'. For example, when each SCCC outer code mode value of regions A/B is equal to '00', 7644 bytes of mobile service data within a primary RS frame may be assigned to regions A/B of the corresponding data group.

TABLE 5

sCCC outer code mode		
For Region C	for Region D	PL
00	00	1980
00	01	1728
01	00	1242
01	01	990
Others		Reserved

Table 5 shows an example of the PL values for each data group within a secondary RS frame, wherein each PL value varies depending upon the SCCC outer code mode, when the RS frame mode value is equal to '01', and when the SCCC block mode value is equal to '00'. For example, when each SCCC outer code mode value of regions C/D is equal to '00', 1980 bytes of mobile service data within a secondary RS frame may be assigned to regions C/D of the corresponding data group.

Service Multiplexer

FIG. **18** illustrates a block diagram showing an example of the service multiplexer. The service multiplexer includes a controller **110** for controlling the overall operations of the service multiplexer, a table information generator **120** for the main service, a null packet generator **130**, an OM packet encapsulator **140**, a mobile service multiplexer **150**, and a transport multiplexer **160**.

The transport multiplexer **160** may include a main service multiplexer **161** and a transport stream (TS) packet multiplexer **162**.

Referring to FIG. **18**, at least one type of compression-encoded main service data and table data generated from the table information generator **120** for the main services are inputted to the main service multiplexer **161** of the transport multiplexer **160**. According to the embodiment of the present invention, the table information generator **120** generates PSI/PSIP table data, which is configured in the form of an MPEG-2 private section.

The main service multiplexer **161** respectively encapsulates each of the main service data and the PSI/PSIP table data, which are being inputted, to MPEG-2 TS packet formats, thereby multiplexing the encapsulated TS packets and outputting the multiplexed packets to the TS packet multiplexer **162**. Herein, the data packet being outputted from the main service multiplexer **161** will hereinafter be referred to as a main service data packet for simplicity.

The mobile service multiplexer **150** receives and respectively encapsulates at least one type of compression-encoded mobile service data and the table information (e.g., PSI/PSIP table data) for mobile services to MPEG-2 TS packet formats. Then, the mobile service multiplexer **150** multiplexes the encapsulated TS packets, thereby outputting the multiplexed packets to the TS packet multiplexer **162**. Hereinafter, the data packet being outputted from the mobile service multiplexer **150** will be referred to as a mobile service data packet for simplicity.

Alternatively, the mobile service multiplexer **150** receives and encapsulates an RS frame payload, which is generated by using at least one type of compression-encoded mobile service data and the signaling table information for mobile services, to MPEG-2 TS packet formats. Then, the mobile service multiplexer **150** multiplexes the encapsulated TS packets, thereby outputting the multiplexed packets to the TS packet multiplexer **162**. Hereinafter, the data packet being outputted from the mobile service multiplexer **150** will be referred to as a mobile service data packet for simplicity.

According to an embodiment of the present invention, the mobile service multiplexer **150** encapsulates an RS frame payload, which is inputted in any one of the formats shown in FIG. **15**, FIG. **17**(a), or FIG. **17**(b), to a TS packet format.

At this point, the transmitter **200** requires identification information in order to identify and process the main service data packet and the mobile service data packet. Herein, the identification information may use values pre-decided in accordance with an agreement between the transmitting system and the receiving system, or may be configured of a separate set of data, or may modify predetermined location value with in the corresponding data packet.

As an example of the present invention, a different packet identifier (PID) may be assigned to identify each of the main service data packet and the mobile service data packet. More specifically, by assigning a PID, which does not use for the main service data packet, to the mobile service data packet, the transmitter **200** refers to a PID of data packet inputted, thereby can identify each of the main service data packet and the mobile service data packet.

In another example, by modifying a synchronization data byte within a header of the mobile service data, the service data packet may be identified by using the synchronization data byte value of the corresponding service data packet. For example, the synchronization byte of the main service data packet directly outputs the value decided by the ISO/IEC 13818-1 standard (i.e., 0x47) without any modification. The synchronization byte of the mobile service data packet modifies and outputs the value, thereby identifying the main service data packet and the mobile service data packet. Conversely, the synchronization byte of the main service data packet is modified and outputted, whereas the synchronization byte of the mobile service data packet is directly outputted without being modified, thereby enabling the main service data packet and the mobile service data packet to be identified.

A plurality of methods may be applied in the method of modifying the synchronization byte. For example, each bit of the synchronization byte may be inversed, or only a portion of the synchronization byte may be inversed.

As described above, any type of identification information may be used to identify the main service data packet and the

mobile service data packet. Therefore, the scope of the present invention is not limited only to the example set forth in the description of the present invention.

Meanwhile, a transport multiplexer used in the conventional digital broadcasting system may be used as the transport multiplexer **160** according to the present invention. More specifically, in order to multiplex the mobile service data and the main service data and to transmit the multiplexed data, the data rate of the main service is limited to a data rate of (19.39−K) Mbps. Then, K Mbps, which corresponds to the remaining data rate, is assigned as the data rate of the mobile service. Thus, the transport multiplexer which is already being used may be used as it is without any modification.

Herein, the transport multiplexer **160** multiplexes the main service data packet being outputted from the main service multiplexer **161** and the mobile service data packet being outputted from the mobile service multiplexer **150**. Thereafter, the transport multiplexer **160** transmits the multiplexed data packets to the transmitter **200**.

However, in some cases, the output data rate of the mobile service multiplexer **150** may not be equal to K Mbps. For example, when the service multiplexer **100** assigns K Mbps of the 19.39 Mbps to the mobile service data, and when the remaining (19.39−K) Mbps is, therefore, assigned to the main service data, the data rate of the mobile service data that are multiplexed by the service multiplexer **100** actually becomes lower than K Mbps. This is because, in case of the mobile service data, the pre-processor of the transmitting system performs additional encoding, thereby increasing the amount of data. Eventually, the data rate of the mobile service data, which may be transmitted from the service multiplexer **100**, becomes smaller than K Mbps.

For example, since the pre-processor of the transmitter performs an encoding process on the mobile service data at a coding rate of at least 1/2, the amount of the data outputted from the pre-processor is increased to more than twice the amount of the data initially inputted to the pre-processor. Therefore, the sum of the data rate of the main service data and the data rate of the mobile service data, both being multiplexed by the service multiplexer **100**, becomes either equal to or smaller than 19.39 Mbps.

In order to set the final output data rate of the mobile service multiplexer **150** to K Mbps, the service multiplexer **100** of the present invention may perform various exemplary operations.

According to an embodiment of the present invention, the null packet generator **130** may generate a null data packet, which is then outputted to the mobile service multiplexer **150**. Thereafter, the mobile service multiplexer **150** may multiplex the null data packet and the mobile service data packets, so as to set the output data rate to K Mbps.

At this point, the null data packet is transmitted to the transmitter **200**, thereby being discarded. More specifically, the null data packet is not transmitted to the receiving system. In order to do so, identification information for identifying the null data is also required. Herein, the identification information for identifying the null data may also use a value pre-decided based upon an agreement between the transmitting system and the receiving system and may also be configured of a separate set of data. And, the identification information for identifying the null data may also change a predetermined position value within the null data packet and use the changed value. For example, the null packet generator **130** may modify (or change) a synchronization byte value within the header of the null data packet, thereby using the changed value as the identification information. Alternatively, the transport_error_indicator flag may be set to '1', thereby being used as the identification information. According to the embodiment of the present invention, the transport_error_indicator flag within the header of the null data packet is used as the identification information for identifying the null data packet. In this case, the transport_error_indicator flag of the null data packet is set to '1', and the transport_error_indicator flag for each of the other remaining data packets is reset to '0', so that the null data packet can be identified (or distinguished).

More specifically, when the null packet generator **130** generated a null data packet, and if, among the fields included in the header of the null data packet, the transport_error_indicator flag is set to '1' and then transmitted, the transmitter **200** may identify and discard the null data packet corresponding to the transport_error_indicator flag.

Herein, any value that can identify the null data packet may be used as the identification information for identifying the null data packet. Therefore, the present invention will not be limited only to the example proposed in the description of the present invention.

As another example of setting (or matching) the final output data rate of the mobile service multiplexer **150** to K Mbps, an operations and maintenance (OM) packet (also referred to as OMP) may be used. In this case, the mobile service multiplexer **150** may multiplex the mobile service data packet, the null data packet, and the OM packet, so as to set the output data rate to K Mbps.

Meanwhile, signaling data, such as transmission parameters, are required for enabling the transmitter **200** to process the mobile service data.

According to an embodiment of the present invention, the transmission parameter is inserted in the payload region of the OM packet, thereby being transmitted to the transmitter.

At this point, in order to enable the transmitter **200** to identify the insertion of the transmission parameter in the OM packet, identification information that can identify the insertion of the transmission parameter in the type field of the corresponding OM packet (i.e., OM_type field).

More specifically, an operations and maintenance packet (OMP) is defined for the purpose of operating and managing the transmitting system. For example, the OMP is configured in an MPEG-2 TS packet format, and the value of its respective PID is equal to '0x1FFA'. The OMP consists of a 4-byte header and a 184-byte payload. Among the 184 bytes, the first byte corresponds to the OM type field indicating the type of the corresponding OM packet (OMP). And, the remaining 183 bytes correspond to an OM payload field, wherein actual data are inserted.

According to the present invention, among the reserved field values of the OM type field, a pre-arranged value is used, thereby being capable of indicating that a transmission parameter has been inserted in the corresponding OM packet. Thereafter, the transmitter **200** may locate (or identify) the corresponding OMP by referring to the respective PID. Subsequently, by parsing the OM type field within the OMP, the transmitter **200** may be able to know (or recognize) whether or not a transmission parameter has been inserted in the corresponding OM packet.

The transmission parameters that can be transmitted to the OM packet include M/H frame information (e.g., M/H frame_index), FIC information (e.g., next_FIC_version_number), parade information (e.g., number_of_parades, parade_id, parade_repetition_cycle, and ensemble_id), group information (e.g., number_of_group and start_group_number), SCCC information (e.g., SCCC_block_mode and SCCC_outer_code_mode), RS frame information (e.g.,

RS_Frame_mode and RS_frame_continuity_counter), RS encoding information (e.g., RS_code_mode), and so on.

At this point, the OM packet in which the transmission parameter is inserted may be periodically generated by a constant cycle, so as to be multiplexed with the mobile service data packet.

The multiplexing rules and the generation of null data packets of the mobile service multiplexer **150**, the main service multiplexer **161**, and the TS packet multiplexer **160** are controlled by the controller **110**.

The TS packet multiplexer **162** multiplexes a data packet being outputted from the main service multiplexer **161** at (19.39-K) Mbps with a data packet being outputted from the mobile service multiplexer **150** at K Mbps. Thereafter, the TS packet multiplexer **162** transmits the multiplexed data packet to the transmitter **200** at a data rate of 19.39 Mbps.

Transmitter

FIG. **19** illustrates a block diagram showing an example of the transmitter **200** according to an embodiment of the present invention. Herein, the transmitter **200** includes a controller **201**, a demultiplexer **210**, a packet jitter mitigator **220**, a pre-processor **230**, a packet multiplexer **240**, a post-processor **250**, a synchronization (sync) multiplexer **260**, and a transmission unit **270**.

Herein, when a data packet is received from the service multiplexer **100**, the demultiplexer **210** should identify whether the received data packet corresponds to a main service data packet, a mobile service data packet, a null data packet, or an OM packet.

For example, the demultiplexer **210** uses the PID within the received data packet so as to identify the main service data packet, the mobile service data packet, and the null data packet. Then, the demultiplexer **210** uses a transport_error_ indicator field to identify the null data packet.

If an OM packet is included in the received data packet, the OM packet may identify using the PID within the received data packet. And by using the OM_type field included in the identified OM packet, the demultiplexer **210** may be able to know whether or not a transmission parameter is included in the payload region of the corresponding OM packet and, then, received.

The main service data packet identified by the demultiplexer **210** is outputted to the packet jitter mitigator **220**, the mobile service data packet is outputted to the pre-processor **230**, and the null data packet is discarded. If the transmission parameter is included in the OM packet, the corresponding transmission parameter is extracted, so as to be outputted to the corresponding blocks. Thereafter, the OM packet is discarded. According to an embodiment of the present invention, the transmission parameter extracted from the OM packet is outputted to the corresponding blocks through the controller **201**.

The pre-processor **230** performs an additional encoding process of the mobile service data included in the service data packet, which is demultiplexed and outputted from the demultiplexer **210**. The pre-processor **230** also performs a process of configuring a data group so that the data group may be positioned at a specific place in accordance with the purpose of the data, which are to be transmitted on a transmission frame. This is to enable the mobile service data to respond swiftly and strongly against noise and channel changes.

According to one embodiment of the present invention, RS frame payload of FIG. **15** (or (a) and (b) of FIG. **17**) is encapsulated into TS packet by the service multiplexer **100** and transmitted to the transmitter. In this case, mobile service data within the mobile service data packet become a part of data of the RS frame payload. In the present invention, for convenience of description, M/H header data of 2 bytes of each M/H service data packet, stuffing data of k bytes, and M/H payload data of N-2-k bytes will be referred to as mobile service data. According to one embodiment of the present invention, the M/H payload data are signaling table and/or IP datagram of mobile service data.

The pre-processor **230** may also refer to the transmission parameter extracted in the OM packet when performing the additional encoding process. Also, the pre-processor **230** groups a plurality of mobile service data packets to configure a data group. Thereafter, known data, mobile service data, RS parity data, and MPEG header are allocated to pre-determined regions within the data group.

Pre-Processor within Transmitter

FIG. **20** illustrates a block diagram showing the structure of a pre-processor **230** according to the present invention. Herein, the pre-processor **230** includes an M/H frame encoder **301**, a block processor **302**, a group formatter **303**, a signaling encoder **304**, and a packet formatter **305**.

The M/H frame encoder **301**, which is included in the pre-processor **230** having the above-described structure, data-randomizes the mobile service data that are inputted to the demultiplexer **210**, thereby forming at least one RS frame belonging to an ensemble.

The M/H frame encoder **301** may include at least one

RS frame encoder. More specifically, RS frame encoders may be provided in parallel, wherein the number of RS frame encoders is equal to the number of parades within the M/H frame. As described above, the M/H frame is a basic time cycle period for transmitting at least one parade. Also, each parade consists of one or two RS frames.

FIG. **21** illustrates a conceptual block diagram of the M/H frame encoder **301** according to an embodiment of the present invention. The M/H frame encoder **301** includes an input demultiplexer (DEMUX) **309**, M number of RS frame encoders **310** to **31M-1**, and an output multiplexer (MUX) **320**. Herein, M represent the number of parades included in one M/H frame.

The demultiplexer **309** output the inputted mobile service data packet to a corresponding RS frame encoder among M number of RS frame encoders in ensemble units.

According to an embodiment of the present invention, each RS frame encoder forms an RS frame payload using mobile service data inputted and performs an error correction encoding process in RS frame payload units, thereby forming an RS frame. Also, each RS frame encoder divides the error-correction-encoded RS frame into a plurality of portions, in order to assign the error-correction-encoded RS frame data to a plurality of data groups. Based upon the RS frame mode of Table 1, data within one RS frame may be assigned either to all of regions A/B/C/D within multiple data groups, or to at least one of regions A/B and regions C/D within multiple data groups.

When the RS frame mode value is equal to '01', i.e., when the data of the primary RS frame are assigned to regions A/B of the corresponding data group and data of the secondary RS frame are assigned to regions C/D of the corresponding data group, each RS frame encoder creates a primary RS frame and a secondary RS frame for each parade. Conversely, when the RS frame mode value is equal to '00', when the data of the primary RS frame are assigned to all of regions A/B/C/D, each RS frame encoder creates a RS frame (i.e., a primary RS frame) for each parade.

Also, each RS frame encoder divides each RS frame into several portions. Each portion of the RS frame is equivalent to a data amount that can be transmitted by a data group. The

output multiplexer (MUX) **320** multiplexes portions within M number of RS frame encoders **310** to **310**M-**1** are multiplexed and then outputted to the block processor **302**.

For example, if one parade transmits two RS frames, portions of primary RS frames within M number of RS frame encoders **310** to **310**M-**1** are multiplexed and outputted. Thereafter, portions of secondary RS frames within M number of RS frame encoders **310** to **310**M-**1** are multiplexed and transmitted.

The input demultiplexer (DEMUX) **309** and the output multiplexer (MUX) **320** operate based upon the control of the controller **201**. The controller **201** may provide necessary (or required) FEC modes to each RS frame encoder. The FEC mode includes the RS code mode, which will be described in detail in a later process.

FIG. **22** illustrates a detailed block diagram of an RS frame encoder among a plurality of RS frame encoders within an M/H frame encoder.

One RS frame encoder may include a primary encoder **410** and a secondary encoder **420**. Herein, the secondary encoder **420** may or may not operate based upon the RS frame mode. For example, when the RS frame mode value is equal to '00', as shown in Table 1, the secondary encoder **420** does not operate.

The primary encoder **410** may include a data randomizer **411**, a Reed-Solomon-cyclic redundancy check (RS-CRC) encoder (**412**), and a RS frame divider **413**. And, the secondary encoder **420** may also include a data randomizer **421**, a RS-CRC encoder (**422**), and a RS frame divider **423**.

More specifically, the data randomizer **411** of the primary encoder **410** receives mobile service data of a primary RS frame payload belonging to a primary ensemble outputted from the output demultiplexer (DEMUX) **309**. Then, after randomizing the received mobile service data, the data randomizer **411** outputs the randomized data to the RS-CRC encoder **412**.

The RS-CRC encoder **412** forms an RS frame payload belonging to the randomized primary ensemble, and performs forward error collection (FEC)-encoding in the RS frame payload unit using at least one of a Reed-Solomon (RS) code and a cyclic redundancy check (CRC) code. The RS-CRC encoder **412** outputs the FEC-encoded RS frame to the RS frame divider **413**.

The RS-CRC encoder **412** groups a plurality of mobile service data that is randomized and inputted, so as to form a RS frame payload. Then, the RS-CRC encoder **412** performs at least one of an error correction encoding process and an error detection encoding process in RS frame payload units, thereby forming an RS frame. Accordingly, robustness may be provided to the mobile service data, thereby scattering group error that may occur during changes in a frequency environment, thereby enabling the mobile service data to respond to the frequency environment, which is extremely vulnerable and liable to frequent changes. Also, the RS-CRC encoder **412** groups a plurality of RS frame so as to create a super frame, thereby performing a row permutation process in super frame units. The row permutation process may also be referred to as a "row interleaving process". Hereinafter, the process will be referred to as "row permutation" for simplicity. In the present invention, the row permutation process is optional.

More specifically, when the RS-CRC encoder **412** performs the process of permuting each row of the super frame in accordance with a pre-determined rule, the position of the rows within the super frame before and after the row permutation process is changed. If the row permutation process is performed by super frame units, and even though the section having a plurality of errors occurring therein becomes very long, and even though the number of errors included in the RS frame, which is to be decoded, exceeds the extent of being able to be corrected, the errors become dispersed within the entire super frame. Thus, the decoding ability is even more enhanced as compared to a single RS frame.

At this point, as an example of the present invention, RS-encoding is applied for the error correction encoding process, and a cyclic redundancy check (CRC) encoding is applied for the error detection process in the RS-CRC encoder **412**. When performing the RS-encoding, parity data that are used for the error correction are generated. And, when performing the CRC encoding, CRC data that are used for the error detection are generated. The CRC data generated by CRC encoding may be used for indicating whether or not the mobile service data have been damaged by the errors while being transmitted through the channel. In the present invention, a variety of error detection coding methods other than the CRC encoding method may be used, or the error correction coding method may be used to enhance the overall error correction ability of the receiving system.

Herein, the RS-CRC encoder **412** refers to a pre-determined transmission parameter provided by the controller **201** so as to perform operations including RS frame configuration, RS encoding, CRC encoding, super frame configuration, and row permutation in super frame units.

FIG. **23**(a) and FIG. **23**(b) illustrate a process of one or two RS frame being divided into several portions, based upon an RS frame mode value, and a process of each portion being assigned to a corresponding region within the respective data group. According to an embodiment of the present invention, the data assignment within the data group is performed by the group formatter **303**.

More specifically, FIG. **23**(a) shows an example of the RS frame mode value being equal to '00'. Herein, only the primary encoder **410** of FIG. **22** operates, thereby forming one RS frame for one parade. Then, the RS frame is divided into several portions, and the data of each portion are assigned to regions A/B/C/D within the respective data group. FIG. **23**(b) shows an example of the RS frame mode value being equal to '01'. Herein, both the primary encoder **410** and the secondary encoder **420** of FIG. **22** operate, thereby forming two RS frames for one parade, i.e., one primary RS frame and one secondary RS frame. Then, the primary RS frame is divided into several portions, and the secondary RS frame is divided into several portions. At this point, the data of each portion of the primary RS frame are assigned to regions A/B within the respective data group. And, the data of each portion of the secondary RS frame are assigned to regions C/D within the respective data group.

Detailed Description of the RS Frame

FIG. **24**(a) illustrates an example of an RS frame being generated from the RS-CRC encoder **412** according to the present invention.

When the RS frame payload is formed, as shown in FIG. **24**(a), the RS-CRC encoder **412** performs a (Nc,Kc)-RS encoding process on each column, so as to generate Nc−Kc (=P) number of parity bytes. Then, the RS-CRC encoder **412** adds the newly generated P number of parity bytes after the very last byte of the corresponding column, thereby creating a column of (187+P) bytes. Herein, as shown in FIG. **24**(a), Kc is equal to 187 (i.e., Kc=187), and Nc is equal to 187+P (i.e., Nc=187+P). Herein, the value of P may vary depending

upon the RS code mode. Table 6 below shows an example of an RS code mode, as one of the RS encoding information.

TABLE 6

RS code mode	RS code	Number of Parity Bytes (P)
00	(211, 187)	24
01	(223, 187)	36
10	(235, 187)	48
11	Reserved	Reserved

Table 6 shows an example of 2 bits being assigned in order to indicate the RS code mode. The RS code mode represents the number of parity bytes corresponding to the RS frame payload.

For example, when the RS code mode value is equal to '10', (235,187)-RS-encoding is performed on the RS frame payload of FIG. **24**(a), so as to generate 48 parity data bytes. Thereafter, the 48 parity bytes are added after the last data byte of the corresponding column, thereby creating a column of 235 data bytes.

When the RS frame mode value is equal to '00' in Table 1 (i.e., when the RS frame mode indicates a single RS frame), only the RS code mode of the corresponding RS frame is indicated. However, when the RS frame mode value is equal to '01' in Table 1 (i.e., when the RS frame mode indicates multiple RS frames), the RS code mode corresponding to a primary RS frame and a secondary RS frame. More specifically, it is preferable that the RS code mode is independently applied to the primary RS frame and the secondary RS frame.

When such RS encoding process is performed on all N number of columns, a size of N(row)×(187+P)(column) bytes may be generated, as shown in FIG. **24**(b).

Each row of the RS frame payload is configured of N bytes. However, depending upon channel conditions between the transmitting system and the receiving system, error may be included in the RS frame payload. When errors occur as described above, CRC data (or CRC code or CRC checksum) may be used on each row unit in order to verify whether error exists in each row unit.

The RS-CRC encoder **412** may perform CRC encoding on the mobile service data being RS encoded so as to create (or generate) the CRC data. The CRC data being generated by CRC encoding may be used to indicate whether the mobile service data have been damaged while being transmitted through the channel.

The present invention may also use different error detection encoding methods other than the CRC encoding method. Alternatively, the present invention may use the error correction encoding method to enhance the overall error correction ability of the receiving system.

FIG. **24**(c) illustrates an example of using a 2-byte (i.e., 16-bit) CRC checksum as the CRC data. Herein, a 2-byte CRC checksum is generated for N number of bytes of each row, thereby adding the 2-byte CRC checksum at the end of the N number of bytes. Thus, each row is expanded to (N+2) number of bytes. Equation 3 below corresponds to an exemplary equation for generating a 2-byte CRC checksum for each row being configured of N number of bytes.

$$g(x)=x^{16}+x^{12}+x^{5}+1 \quad \text{Equation 3}$$

The process of adding a 2-byte checksum in each row is only exemplary. Therefore, the present invention is not limited only to the example proposed in the description set forth herein. As described above, when the process of RS encoding and CRC encoding are completed, the (N×187)-byte RS frame payload is converted into a (N+2)×(187+P)-byte RS frame. Based upon an error correction scenario of a RS frame formed as described above, the data bytes within the RS frame are transmitted through a channel in a row direction. At this point, when a large number of errors occur during a limited period of transmission time, errors also occur in a row direction within the RS frame being processed with a decoding process in the receiving system. However, in the perspective of RS encoding performed in a column direction, the errors are shown as being scattered. Therefore, error correction may be performed more effectively. At this point, a method of increasing the number of parity data bytes (P) may be used in order to perform a more intense error correction process. However, using this method may lead to a decrease in transmission efficiency. Therefore, a mutually advantageous method is required. Furthermore, when performing the decoding process, an erasure decoding process may be used to enhance the error correction performance.

Additionally, the RS-CRC encoder **412** according to the present invention also performs a row permutation (or interleaving) process in super frame units in order to further enhance the error correction performance when error correction the RS frame.

FIG. **25**(a) to FIG. **25**(d) illustrates an example of performing a row permutation process in super frame units according to the present invention. More specifically, G number of RS frames RS-CRC-encoded is grouped to form a super frame, as shown in FIG. **25**(a). At this point, since each RS frame is formed of (N+2)×(187+P) number of bytes, one super frame is configured to have the size of (N+2)×(187+P)×G bytes.

When a row permutation process permuting each row of the super frame configured as described above is performed based upon a pre-determined permutation rule, the positions of the rows prior to and after being permuted (or interleaved) within the super frame may be altered. More specifically, the i^{th} row of the super frame prior to the interleaving process, as shown in FIG. **25**(b), is positioned in the i^{th} row of the same super frame after the row permutation process, as shown in FIG. **25**(c). The above-described relation between i and j can be easily understood with reference to a permutation rule as shown in Equation 4 below.

$$j=G(i \bmod (187+P))+\lfloor i/(187+P) \rfloor$$

$$i=(187+P)(j \bmod G)+\lfloor j/G \rfloor$$

where $0 \le i, j \le (187+P)G-1$; or

where $0 \le i, j < (187+P)G$

Herein, each row of the super frame is configured of (N+2) number of data bytes even after being row-permuted in super frame units.

When all row permutation processes in super frame units are completed, the super frame is once again divided into G number of row-permuted RS frames, as shown in FIG. **25**(d), and then provided to the RS frame divider **413**. Herein, the number of RS parity bytes and the number of columns should be equally provided in each of the RS frames, which configure a super frame. As described in the error correction scenario of a RS frame, in case of the super frame, a section having a large number of error occurring therein is so long that, even when one RS frame that is to be decoded includes an excessive number of errors (i.e., to an extent that the errors cannot be corrected), such errors are scattered throughout the entire super frame. Therefore, in

comparison with a single RS frame, the decoding performance of the super frame is more enhanced.

The above description of the present invention corresponds to the processes of forming (or creating) and encoding an RS frame, when a data group is divided into regions A/B/C/D, and when data of an RS frame are assigned to all of regions A/B/C/D within the corresponding data group. More specifically, the above description corresponds to an embodiment of the present invention, wherein one RS frame is transmitted using one parade. In this embodiment, the secondary encoder **420** does not operate (or is not active).

Meanwhile, 2 RS frames are transmitting using one parade, the data of the primary RS frame may be assigned to regions A/B within the data group and be transmitted, and the data of the secondary RS frame may be assigned to regions C/D within the data group and be transmitted. At this point, the primary encoder **410** receives the mobile service data that are to be assigned to regions A/B within the data group, forms the primary RS frame payload, and then performs RS-encoding and CRC-encoding on the primary RS frame payload, thereby forming the primary RS frame. Similarly, the secondary encoder **420** receives the mobile service data that are to be assigned to regions C/D within the data group, forms the secondary RS frame payload, and then performs RS-encoding and CRC-encoding on the secondary RS frame payload thereby forming the secondary RS frame. More specifically, the primary RS frame and the secondary RS frame are generated independently.

FIG. **26** illustrates examples of receiving the mobile service data that are to be assigned to regions A/B within the data group, so as to form the primary RS frame payload, and receives the mobile service data that are to be assigned to regions C/D within the data group, so as to form the secondary RS frame payload, thereby performing error correction encoding and error detection encoding on each of the first and secondary RS frame payloads.

More specifically, FIG. **26**(a) illustrates an example of the RS-CRC encoder **412** of the primary encoder **410** receiving mobile service data of the primary ensemble that are to be assigned to regions A/B within the corresponding data group, so as to create an RS frame payload having the size of N1(row)×187(column). Then, in this example, the primary encoder **410** performs RS-encoding on each column of the RS frame payload created as described above, thereby adding P1 number of parity data bytes in each column. Finally, the primary encoder **410** performs CRC-encoding on each row, thereby adding a 2-byte checksum in each row, thereby forming an primary RS frame.

FIG. **26**(b) illustrates an example of the RS-CRC encoder **422** of the secondary encoder **420** receiving mobile service data of the secondary ensemble that are to be assigned to regions C/D within the corresponding data group, so as to create an RS frame payload having the size of N2(row)×187 (column). Then, in this example, the secondary encoder **420** performs RS-encoding on each column of the RS frame payload created as described above, thereby adding P2 number of parity data bytes in each column. Finally, the secondary encoder **420** performs CRC-encoding on each row, thereby adding a 2-byte checksum in each row, thereby forming an secondary RS frame.

At this point, each of the RS-CRC encoders **412** and **422** may refer to a pre-determined transmission parameter provided by the controller **201**, the RS-CRC encoders **412** and **422** may be informed of M/H frame information, FIC information, RS frame information (including RS frame mode information), RS encoding information (including RS code mode), SCCC information (including SCCC block mode information and SCCC outer code mode information), data group information, and region information within a data group. The RS-CRC encoders **412** and **422** may refer to the transmission parameters for the purpose of RS frame configuration, error correction encoding, error detection encoding. Furthermore, the transmission parameters should also be transmitted to the receiving system so that the receiving system can perform a normal decoding process. At this point, as an example of the present invention, the transmission parameter is transmitted through transmission parameter channel (TPC) to a receiving system. The TPC will be described in detail in a later.

The data of the primary RS frame, which is encoded by RS frame units and row-permuted by super frame units from the RS-CRC encoder **412** of the primary encoder **410**, are outputted to the RS frame divider **413**. If the secondary encoder **420** also operates in the embodiment of the present invention, the data of the secondary RS frame, which is encoded by RS frame units and row-permuted by super frame units from the RS-CRC encoder **422** of the secondary encoder **420**, are outputted to the RS frame divider **423**. The RS frame divider **413** of the primary encoder **410** divides the primary RS frame into several portions, which are then outputted to the output multiplexer (MUX) **320**. Each portion of the primary RS frame is equivalent to a data amount that can be transmitted by one data group. Similarly, the RS frame divider **423** of the secondary encoder **420** divides the secondary RS frame into several portions, which are then outputted to the output multiplexer (MUX) **320**.

Hereinafter, the RS frame divider **413** of the primary RS encoder **410** will now be described in detail. Also, in order to simplify the description of the present invention, it is assumed that an RS frame payload having the size of N(row)×187 (column), as shown in FIG. **24**(a) to FIG. **24**(c), that P number of parity data bytes are added to each column by RS-encoding the RS frame payload, and that a 2-byte checksum is added to each row by CRC-encoding the RS frame payload. As a result, an RS frame having the size of (N+2) (row)×187+P (column) is formed. Accordingly, the RS frame divider **413** divides (or partitions) the RS frame having the size of (N+2) (row)×187+P (column) into several portions, each having the size of PL (wherein PL corresponds to the length of the RS frame portion).

At this point, as shown in Table 2 to Table 5, the value of PL may vary depending upon the RS frame mode, SCCC block mode, and SCCC outer coder mode. Also, the total number of data bytes of the RS-encoded and CRC-encoded RS frame is equal to or smaller than 5×NoG×PL. In this case, the RS frame is divided (or partitioned) into ((5×NoG)−1) number of portions each having the size of PL and one portion having a size equal to smaller than PL. More specifically, with the exception of the last portion of the RS frame, each of the remaining portions of the RS frame has an equal size of PL. If the size of the last portion is smaller than PL, a stuffing byte (or dummy byte) may be inserted in order to fill (or replace) the lacking number of data bytes, thereby enabling the last portion of the RS frame to also be equal to PL. Each portion of an RS frame corresponds to the amount of data that are to be SCCC-encoded and mapped into a single data group of a parade.

FIG. **27**(a) and FIG. **27**(b) respectively illustrate examples of adding S number of stuffing bytes, when an RS frame having the size of (N+2) (row)×(187+P) (column) is divided into 5×NoG number of portions, each having the size of PL. More specifically, the RS-encoded and CRC-encoded RS frame, shown in FIG. **27**(a), is divided into several portions, as shown in FIG. **27**(b). The number of divided portions at

the RS frame is equal to (5×NoG). Particularly, the first ((5×NoG)−1) number of portions each has the size of PL, and the last portion of the RS frame may be equal to or smaller than PL. If the size of the last portion is smaller than PL, a stuffing byte (or dummy byte) may be inserted in order to fill (or replace) the lacking number of data bytes, as shown in Equation 5 below, thereby enabling the last portion of the RS frame to also be equal to PL.

$$S=(5\times NoG\times PL)-((N+2)\times(187+P)) \quad \text{Equation 5}$$

Herein, each portion including data having the size of PL passes through the output multiplexer **320** of the M/H frame encoder **301**, which is then outputted to the block processor **302**.

At this point, the mapping order of the RS frame portions to a parade of data groups in not identical with the group assignment order defined in Equation 1. When given the group positions of a parade in an M/H frame, the SCCC-encoded RS frame portions will be mapped in a time order (i.e., in a left-to-right direction).

For example, as shown in FIG. **11**, data groups of the 2^{nd} parade (Parade #**1**) are first assigned (or allocated) to the 13^{th} slot (Slot #**12**) and then assigned to the 3^{rd} slot (Slot #**2**). However, when the data are actually placed in the assigned slots, the data are placed in a time sequence (or time order, i.e., in a left-to-right direction). More specifically, the 1^{st} data group of Parade #**1** is placed in Slot #**2**, and the 2^{nd} data group of Parade #**1** is placed in Slot #**12**.

Block Processor

Meanwhile, the block processor **302** performs an SCCC outer encoding process on the output of the M/H frame encoder **301**. More specifically, the block processor **302** receives the data of each error correction encoded portion. Then, the block processor **302** encodes the data once again at a coding rate of 1/H (wherein H is an integer equal to or greater than 2 (i.e., H≥2)), thereby outputting the 1/H-rate encoded data to the group formatter **303**. According to the embodiment of the present invention, the input data are encoded either at a coding rate of 1/2 (also referred to as "1/2-rate encoding") or at a coding rate of 1/4 (also referred to as "1/4-rate encoding"). The data of each portion outputted from the M/H frame encoder **301** may include at least one of mobile service data, RS parity data, CRC data, and stuffing data. However, in a broader meaning, the data included in each portion may correspond to data for mobile services. Therefore, the data included in each portion will all be considered as mobile service data and described accordingly.

The group formatter **303** inserts the mobile service data SCCC-outer-encoded and outputted from the block processor **302** in the corresponding region within the data group, which is formed in accordance with a pre-defined rule. Also, in association with the data deinterleaving process, the group formatter **303** inserts various place holders (or known data place holders) in the corresponding region within the data group. Thereafter, the group formatter **303** deinterleaves the data within the data group and the place holders.

According to the present invention, with reference to data after being data-interleaved, as shown in FIG. **5**, a data groups is configured of **10** M/H blocks (B**1** to B**10**) and divided into 4 regions (A, B, C, and D). Also, as shown in FIG. **5**, when it is assumed that the data group is divided into a plurality of hierarchical regions, as described above, the block processor **302** may encode the mobile service data, which are to be inserted to each region based upon the characteristic of each hierarchical region, at different coding rates. For example, the block processor **302** may encode the mobile service data, which are to be inserted in region A/B within the corresponding data group, at a coding rate of 1/2. Then, the group formatter **303** may insert the 1/2-rate encoded mobile service data to region A/B. Also, the block processor **302** may encode the mobile service data, which are to be inserted in region C/D within the corresponding data group, at a coding rate of 1/4 having higher (or stronger) error correction ability than the 1/2-coding rate. Thereafter, the group formatter **303** may insert the 1/2-rate encoded mobile service data to region C/D. In another example, the block processor **302** may encode the mobile service data, which are to be inserted in region C/D, at a coding rate having higher error correction ability than the 1/4-coding rate. Then, the group formatter **303** may either insert the encoded mobile service data to region C/D, as described above, or leave the data in a reserved region for future usage.

According to another embodiment of the present invention, the block processor **302** may perform a 1/H-rate encoding process in SCCC block units. Herein, the SCCC block includes at least one M/H block. At this point, when 1/H-rate encoding is performed in M/H block units, the M/H blocks (B**1** to B**10**) and the SCCC block (SCB**1** to SCB**10**) become identical to one another (i.e., SCB**1**=B**1**, SCB**2**=B**2**, SCB**3**=B**3**, SCB**4**=B**4**, SCB**5**=B**5**, SCB**6**=B**6**, SCB**7**=B**7**, SCB**8**=B**8**, SCB**9**=B**9**, and SCB**10**=B**10**). For example, the M/H block **1** (B**1**) may be encoded at the coding rate of 1/2, the M/H block **2** (B**2**) may be encoded at the coding rate of 1/4, and the M/H block **3** (B**3**) may be encoded at the coding rate of 1/2. The coding rates are applied respectively to the remaining M/H blocks.

Alternatively, a plurality of M/H blocks within regions A, B, C, and D may be grouped into one SCCC block, thereby being encoded at a coding rate of 1/H in SCCC block units. Accordingly, the receiving performance of region C/D may be enhanced. For example, M/H block **1** (B**1**) to M/H block **5** (B**5**) may be grouped into one SCCC block and then encoded at a coding rate of 1/2. Thereafter, the group formatter **303** may insert the 1/2-rate encoded mobile service data to a section starting from M/H block **1** (B**1**) to M/H block **5** (B**5**). Furthermore, M/H block **6** (B**6**) to M/H block **10** (B**10**) may be grouped into one SCCC block and then encoded at a coding rate of 1/4. Thereafter, the group formatter **303** may insert the 1/4-rate encoded mobile service data to another section starting from M/H block **6** (B**6**) to M/H block **10** (B**10**). In this case, one data group may consist of two SCCC blocks.

According to another embodiment of the present invention, one SCCC block may be formed by grouping two M/H blocks. For example, M/H block **1** (B**1**) and M/H block **6** (B**6**) may be grouped into one SCCC block (SCB**1**). Similarly, M/H block **2** (B**2**) and M/H block **7** (B**7**) may be grouped into another SCCC block (SCB**2**). Also, M/H block **3** (B**3**) and M/H block **8** (B**8**) may be grouped into another SCCC block (SCB**3**). And, M/H block **4** (B**4**) and M/H block **9** (B**9**) may be grouped into another SCCC block (SCB**4**). Furthermore, M/H block **5** (B**5**) and M/H block **10** (B**10**) may be grouped into another SCCC block (SCB**5**). In the above-described example, the data group may consist of 10 M/H blocks and 5 SCCC blocks. Accordingly, in a data (or signal) receiving environment undergoing frequent and severe channel changes, the receiving performance of regions C and D, which is relatively more deteriorated than the receiving performance of region A, may be reinforced. Furthermore, since the number of mobile service data symbols increases more and more from region A to region D, the error correction encoding performance becomes more and more deteriorated. Therefore, when grouping a plurality of

M/H block to form one SCCC block, such deterioration in the error correction encoding performance may be reduced.

As described-above, when the block processor **302** performs encoding at a 1/H-coding rate, information associated with SCCC should be transmitted to the receiving system in order to accurately recover the mobile service data. Table 7 below shows an example of a SCCC block mode, which indicating the relation between an M/H block and an SCCC block, among diverse SCCC block information.

TABLE 7

	SCCC Block Mode			
	00	01	10	11
	Description			
SCB	One M/H Block per SCCC Block SCB input, M/H Block	Two M/H Blocks per SCCC Block SCB input, M/H Blocks	Reserved	Reserved
SCB1	B1	B1 + B6		
SCB2	B2	B2 + B7		
SCB3	B3	B3 + B8		
SCB4	B4	B4 + B9		
SCB5	B5	B5 + B10		
SCB6	B6	—		
SCB7	B7	—		
SCB8	B8	—		
SCB9	B9	—		
SCB10	B10	—		

More specifically, Table 4 shows an example of 2 bits being allocated in order to indicate the SCCC block mode. For example, when the SCCC block mode value is equal to '00', this indicates that the SCCC block and the M/H block are identical to one another. Also, when the SCCC block mode value is equal to '01', this indicates that each SCCC block is configured of 2 M/H blocks.

As described above, if one data group is configured of 2 SCCC blocks, although it is not indicated in Table 7, this information may also be indicated as the SCCC block mode. For example, when the SCCC block mode value is equal to '10', this indicates that each SCCC block is configured of 5 M/H blocks and that one data group is configured of 2 SCCC blocks. Herein, the number of M/H blocks included in an SCCC block and the position of each M/H block may vary depending upon the settings made by the system designer. Therefore, the present invention will not be limited to the examples given herein. Accordingly, the SCCC mode information may also be expanded.

An example of a coding rate information of the SCCC block, i.e., SCCC outer code mode, is shown in Table 8 below.

TABLE 8

SCCC outer code mode (2 bits)	Description
00	Outer code rate of SCCC block is ½ rate
01	Outer code rate of SCCC block is ¼ rate
10	Reserved
11	Reserved

More specifically, Table 8 shows an example of 2 bits being allocated in order to indicate the coding rate information of the SCCC block. For example, when the SCCC outer code mode value is equal to '00', this indicates that the coding rate of the corresponding SCCC block is 1/2. And, when the SCCC outer code mode value is equal to '01', this indicates that the coding rate of the corresponding SCCC block is 1/4.

If the SCCC block mode value of Table 7 indicates '00', the SCCC outer code mode may indicate the coding rate of each M/H block with respect to each M/H block. In this case, since it is assumed that one data group includes 10 M/H blocks and that 2 bits are allocated for each SCCC block mode, a total of 20 bits are required for indicating the SCCC block modes of the 10 M/H modes. In another example, when the SCCC block mode value of Table 7 indicates '00', the SCCC outer code mode may indicate the coding rate of each region with respect to each region within the data group. In this case, since it is assumed that one data group includes 4 regions (i.e., regions A, B, C, and D) and that 2 bits are allocated for each SCCC block mode, a total of 8 bits are required for indicating the SCCC block modes of the 4 regions. In another example, when the SCCC block mode value of Table 7 is equal to '01', each of the regions A, B, C, and D within the data group has the same SCCC outer code mode.

Meanwhile, an example of an SCCC output block length (SOBL) for each SCCC block, when the SCCC block mode value is equal to '00', is shown in Table 9 below.

TABLE 9

		SIBL	
SCCC Block	SOBL	½ rate	¼ rate
SCB1 (B1)	528	264	132
SCB2 (B2)	1536	768	384
SCB3 (B3)	2376	1188	594
SCB4 (B4)	2388	1194	597
SCB5 (B5)	2772	1386	693
SCB6 (B6)	2472	1236	618
SCB7 (B7)	2772	1386	693
SCB8 (B8)	2508	1254	627
SCB9 (B9)	1416	708	354
SCB10 (B10)	480	240	120

More specifically, when given the SCCC output block length (SOBL) for each SCCC block, an SCCC input block length (SIBL) for each corresponding SCCC block may be decided based upon the outer coding rate of each SCCC block. The SOBL is equivalent to the number of SCCC output (or outer-encoded) bytes for each SCCC block. And, the SIBL is equivalent to the number of SCCC input (or payload) bytes for each SCCC block. Table 10 below shows an example of the SOBL and SIBL for each SCCC block, when the SCCC block mode value is equal to '01'.

TABLE 10

		SIBL	
SCCC Block	SOBL	½ rate	¼ rate
SCB1 (B1 + B6)	528	264	132
SCB2 (B2 + B7)	1536	768	384
SCB3 (B3 + B8)	2376	1188	594
SCB4 (B4 + B9)	2388	1194	597
SCB5 (B5 + B10)	2772	1386	693

In order to do so, as shown in FIG. **28**, the block processor **302** includes a RS frame portion-SCCC block converter **511**, a byte-bit converter **512**, a convolution encoder **513**, a symbol interleaver **514**, a symbol-byte converter **515**, and an

SCCC block-M/H block converter **516**. The convolutional encoder **513** and the symbol interleaver **514** are virtually concatenated with the trellis encoding module in the post-processor in order to configure an SCCC block. More specifically, the RS frame portion-SCCC block converter **511** divides the RS frame portions, which are being inputted, into multiple SCCC blocks using the SIBL of Table 9 and Table 10 based upon the RS code mode, SCCC block mode, and SCCC outer code mode. Herein, the M/H frame encoder **301** may output only primary RS frame portions or both primary RS frame portions and secondary RS frame portions in accordance with the RS frame mode.

When the RS Frame mode is set to '00', a portion of the primary RS Frame equal to the amount of data, which are to be SCCC outer encoded and mapped to 10 M/H blocks (B**1** to B**10**) of a data group, will be provided to the block processor **302**. When the SCCC block mode value is equal to '00', then the primary RS frame portion will be split into 10 SCCC Blocks according to Table 9. Alternatively, when the SCCC block mode value is equal to '01', then the primary RS frame will be split into 5 SCCC blocks according to Table 10.

When the RS frame mode value is equal to '01', then the block processor **302** may receive two RS frame portions. The RS frame mode value of '01' will not be used with the SCCC block mode value of '01'. The first portion from the primary RS frame will be SCCC-outer-encoded as SCCC Blocks SCB**3**, SCB**4**, SCB**5**, SCB**6**, SCB**7**, and SCB**8** by the block processor **302**. The SCCC Blocks SCB**3** and SCB**8** will be mapped to region B and the SCCC blocks SCB**4**, SCB**5**, SCB**6**, and SCB**7** shall be mapped to region A by the group formatter **303**. The second portion from the secondary RS frame will also be SCCC-outer-encoded, as SCB**1**, SCB**2**, SCB**9**, and SCB**10**, by the block processor **302**. The group formatter **303** will map the SCCC blocks SCB**1** and SCB**10** to region D as the M/H blocks B**1** and B**10**, respectively. Similarly, the SCCC blocks SCB**2** and SCB**9** will be mapped to region C as the M/H blocks B**2** and B**9**.

The byte-bit converter **512** identifies the mobile service data bytes of each SCCC block outputted from the RS frame portion-SCCC block converter **511** as data bits, which are then outputted to the convolution encoder **513**. The convolution encoder **513** performs one of 1/2-rate encoding and 1/4-rate encoding on the inputted mobile service data bits.

FIG. **29** illustrates a detailed block diagram of the convolution encoder **513**. The convolution encoder **513** includes two delay units **521** and **523** and three adders **522**, **524**, and **525**. Herein, the convolution encoder **513** encodes an input data bit U and outputs the coded bit U to 5 bits (u**0** to u**4**). At this point, the input data bit U is directly outputted as uppermost bit u**0** and simultaneously encoded as lower bit u**1**u**2**u**3**u**4** and then outputted. More specifically, the input data bit U is directly outputted as the uppermost bit u**0** and simultaneously outputted to the first and third adders **522** and **525**.

The first adder **522** adds the input data bit U and the output bit of the first delay unit **521** and, then, outputs the added bit to the second delay unit **523**. Then, the data bit delayed by a pre-determined time (e.g., by 1 clock) in the second delay unit **523** is outputted as a lower bit u**1** and simultaneously fedback to the first delay unit **521**. The first delay unit **521** delays the data bit fed-back from the second delay unit **523** by a pre-determined time (e.g., by 1 clock). Then, the first delay unit **521** outputs the delayed data bit as a lower bit u**2** and, at the same time, outputs the fed-back data to the first adder **522** and the second adder **524**. The second adder **524** adds the data bits outputted from the first and second delay units **521** and **523** and outputs the added data bits as a lower bit u**3**. The third adder **525** adds the input data bit U and the output of the second delay unit **523** and outputs the added data bit as a lower bit u**4**.

At this point, the first and second delay units **521** and **523** are reset to '0', at the starting point of each SCCC block. The convolution encoder **513** of FIG. **29** may be used as a 1/2-rate encoder or a 1/4-rate encoder. More specifically, when a portion of the output bit of the convolution encoder **513**, shown in FIG. **29**, is selected and outputted, the convolution encoder **513** may be used as one of a 1/2-rate encoder and a 1/4-rate encoder. Table 11 below shown an example of output symbols of the convolution encoder **513**.

TABLE 11

		¼rate	
Region	½ rate	SCCC block mode = '00'	SCCC block mode = '01'
A, B	(u0, u1)	(u0, u2), (u1, u3)	(u0, u2), (u1,u4)
C, D		(u0, u1), (u3, u4)	

For example, at the 1/2-coding rate, 1 output symbol (i.e., u**0** and u**1** bits) may be selected and outputted. And, at the 1/4-coding rate, depending upon the SCCC block mode, 2 output symbols (i.e., 4 bits) may be selected and outputted. For example, when the SCCC block mode value is equal to '01', and when an output symbol configured of u**0** and u**2** and another output symbol configured of u**1** and u**4** are selected and outputted, a 1/4-rate coding result may be obtained.

The mobile service data encoded at the coding rate of 1/2 or 1/4 by the convolution encoder **513** are outputted to the symbol interleaver **514**. The symbol interleaver **514** performs block interleaving, in symbol units, on the output data symbol of the convolution encoder **513**. More specifically, the symbol interleaver **514** is a type of block interleaver. Any interleaver performing structural rearrangement (or realignment) may be applied as the symbol interleaver **514** of the block processor. However, in the present invention, a variable length symbol interleaver that can be applied even when a plurality of lengths is provided for the symbol, so that its order may be rearranged, may also be used.

FIG. **30** illustrates a symbol interleaver according to an embodiment of the present invention. Particularly, FIG. illustrates an example of the symbol interleaver when B=2112 and L=4096. Herein, B indicates a block length in symbols that are outputted for symbol interleaving from the convolution encoder **513**. And, L represents a block length in symbols that are actually interleaved by the symbol interleaver **514**. At this point, the block length in symbols B inputted to the symbol interleaver **514** is equivalent to 4×SOBL . More specifically, since one symbol is configured of 2 bits, the value of B may be set to be equal to 4×SOBL.

In the present invention, when performing the symbol-interleaving process, the conditions of L=2 (wherein m is an integer) and of L≥B should be satisfied. If there is a difference in value between B and L, (L−B) number of null (or dummy) symbols is added, thereby creating an interleaving pattern, as shown in P'(i) of FIG. **30**. Therefore, B becomes a block size of the actual symbols that are inputted to the symbol interleaver **514** in order to be interleaved. L becomes an interleaving unit when the interleaving process is performed by an interleaving pattern created from the symbol interleaver **514**.

Equation 6 shown below describes the process of sequentially receiving B number of symbols, the order of which is to be rearranged, and obtaining an L value satisfying the conditions of $L=2^m$ (wherein m is an integer) and of $L \geq B$, thereby creating the interleaving so as to realign (or rearrange) the symbol order.

In relation to all places, wherein $0 \leq i \leq B-1$,

$$P'(i)=\{89 \times i \times (i+1)/2\} \bmod L \quad \text{Equation 6}$$

Herein, $L \geq B$, $L=2^m$, wherein m is an integer .

As shown in P'(i) of FIG. **30**, the order of B number of input symbols and (L−B) number of null symbols is rearranged by using the above-mentioned Equation 6. Then, as shown in P(i) of FIG. **30**, the null byte places are removed, so as to rearrange the order. Starting with the lowest value of i, the P(i) are shifted to the left in order to fill the empty entry locations. Thereafter, the symbols of the aligned interleaving pattern P(i) are outputted to the symbol-byte converter **515** in order. Herein, the symbol-byte converter **515** converts to bytes the mobile service data symbols, having the rearranging of the symbol order completed and then outputted in accordance with the rearranged order, and thereafter outputs the converted bytes to the SCCC block-M/H block converter **516**. The SCCC block-M/H block converter **516** converts the symbol-interleaved SCCC blocks to M/H blocks, which are then outputted to the group formatter **303**.

If the SCCC block mode value is equal to '00', the SCCC block is mapped at a one-to-one (1:1) correspondence with each M/H block within the data group. In another example, if the SCCC block mode value is equal to '01', each SCCC block is mapped with two M/H blocks within the data group. For example, the SCCC block SCB**1** is mapped with (B**1**, B**6**), the SCCC block SCB**2** is mapped with (B**2**, B**7**), the SCCC block SCB**3** is mapped with (B**3**, B**8**), the SCCC block SCB**4** is mapped with (B**4**, B**9**), and the SCCC block SCB**5** is mapped with (B**5**, B**10**). The M/H block that is outputted from the SCCC block-M/H block converter **516** is configured of mobile service data and FEC redundancy. In the present invention, the mobile service data as well as the FEC redundancy of the M/H block will be collectively considered as mobile service data.

Group Formatter

The group formatter **303** inserts data of M/H blocks outputted from the block processor **302** to the corresponding M/H blocks within the data group, which is formed in accordance with a pre-defined rule. Also, in association with the data-deinterleaving process, the group formatter **303** inserts various place holders (or known data place holders) in the corresponding region within the data group. More specifically, apart from the encoded mobile service data outputted from the block processor **302**, the group formatter **303** also inserts MPEG header place holders, non-systematic RS parity place holders, main service data place holders, which are associated with the data deinterleaving in a later process, as shown in FIG. **5**.

Herein, the main service data place holders are inserted because the mobile service data bytes and the main service data bytes are alternately mixed with one another in regions B to D based upon the input of the data deinterleaver, as shown in FIG. **5**. For example, based upon the data outputted after data deinterleaving, the place holder for the MPEG header may be allocated at the very beginning of each packet. Also, in order to configure an intended group format, dummy bytes may also be inserted. Furthermore, the group formatter **303** inserts initialization data (i.e., trellis initialization byte) of the trellis encoding module **256** in the corresponding regions. For example, the initialization data may be inserted in the beginning of the known data sequence. The initialization data is used for initializing memories within the trellis encoding module **256**, and is not transmitted to the receiving system.

Additionally, the group formatter **303** may also insert signaling information, which are encoded and outputted from the signaling encoder **304**, in corresponding regions within the data group. At this point, reference may be made to the signaling information when the group formatter **303** inserts each data type and respective place holders in the data group. The process of encoding the signaling information and inserting the encoded signaling information to the data group will be described in detail in a later process.

After inserting each data type and respective place holders in the data group, the group formatter **303** may deinterleave the data and respective place holders, which have been inserted in the data group, as an inverse process of the data interleaver, thereby outputting the deinterleaved data and respective place holders to the packet formatter **305**. The group formatter **303** may include a group format organizer **527**, and a data deinterleaver **529**, as shown in FIG. **31**. The group format organizer **527** inserts data and respective place holders in the corresponding regions within the data group, as described above. And, the data deinterleaver **529** deinterleaves the inserted data and respective place holders as an inverse process of the data interleaver.

The packet formatter **305** removes the main service data place holders and the RS parity place holders that were allocated for the deinterleaving process from the deinterleaved data being inputted. Then, the packet formatter **305** groups the remaining portion and inserts the 3-byte MPEG header place holder in an MPEG header having a null packet PID (or an unused PID from the main service data packet). Furthermore, the packet formatter **305** adds a synchronization data byte at the beginning of each 187-byte data packet. Also, when the group formatter **303** inserts known data place holders, the packet formatter **303** may insert actual known data in the known data place holders, or may directly output the known data place holders without any modification in order to make replacement insertion in a later process. Thereafter, the packet formatter **305** identifies the data within the packet-formatted data group, as described above, as a 188-byte unit mobile service data packet (i.e., MPEG TS packet), which is then provided to the packet multiplexer **240**.

Based upon the control of the controller **201**, the packet multiplexer **240** multiplexes the data group packet-formatted and outputted from the packet formatter **306** and the main service data packet outputted from the packet jitter mitigator **220**. Then, the packet multiplexer **240** outputs the multiplexed data packets to the data randomizer **251** of the post-processor **250**. More specifically, the controller **201** controls the time-multiplexing of the packet multiplexer **240**. If the packet multiplexer **240** receives 118 mobile service data packets from the packet formatter **305**, 37 mobile service data packets are placed before a place for inserting VSB field synchronization. Then, the remaining 81 mobile service data packets are placed after the place for inserting VSB field synchronization. The multiplexing method may be adjusted by diverse variables of the system design. The multiplexing method and multiplexing rule of the packet multiplexer **240** will be described in more detail in a later process.

Also, since a data group including mobile service data in-between the data bytes of the main service data is multiplexed (or allocated) during the packet multiplexing

process, the shifting of the chronological position (or place) of the main service data packet becomes relative. Also, a system object decoder (i.e., MPEG decoder) for processing the main service data of the receiving system, receives and decodes only the main service data and recognizes the mobile service data packet as a null data packet.

Therefore, when the system object decoder of the receiving system receives a main service data packet that is multiplexed with the data group, a packet jitter occurs.

At this point, since a multiple-level buffer for the video data exists in the system object decoder and the size of the buffer is relatively large, the packet jitter generated from the packet multiplexer **240** does not cause any serious problem in case of the video data. However, since the size of the buffer for the audio data in the object decoder is relatively small, the packet jitter may cause considerable problem. More specifically, due to the packet jitter, an overflow or underflow may occur in the buffer for the main service data of the receiving system (e.g., the buffer for the audio data). Therefore, the packet jitter mitigator **220** re-adjusts the relative position of the main service data packet so that the overflow or underflow does not occur in the system object decoder.

In the present invention, examples of repositioning places for the audio data packets within the main service data in order to minimize the influence on the operations of the audio buffer will be described in detail. The packet jitter mitigator **220** repositions the audio data packets in the main service data section so that the audio data packets of the main service data can be as equally and uniformly aligned and positioned as possible. Additionally, when the positions of the main service data packets are relatively re-adjusted, associated program clock reference (PCR) values may also be modified accordingly. The PCR value corresponds to a time reference value for synchronizing the time of the MPEG decoder. Herein, the PCR value is inserted in a specific region of a TS packet and then transmitted.

In the example of the present invention, the packet jitter mitigator **220** also performs the operation of modifying the PCR value. The output of the packet jitter mitigator **220** is inputted to the packet multiplexer **240**. As described above, the packet multiplexer **240** multiplexes the main service data packet outputted from the packet jitter mitigator **220** with the mobile service data packet outputted from the pre-processor **230** into a burst structure in accordance with a pre-determined multiplexing rule. Then, the packet multiplexer **240** outputs the multiplexed data packets to the data randomizer **251** of the post-processor **250**.

If the inputted data correspond to the main service data packet, the data randomizer **251** performs the same randomizing process as that of the conventional randomizer. More specifically, the synchronization byte within the main service data packet is deleted. Then, the remaining 187 data bytes are randomized by using a pseudo random byte generated from the data randomizer **251**. Thereafter, the randomized data are outputted to the RS encoder/non-systematic RS encoder **252**. On the other hand, if the inputted data correspond to the mobile service data packet, the data randomizer **251** may not perform a randomizing process on the mobile service data packet.

The RS encoder/non-systematic RS encoder **252** performs an RS encoding process on the data being randomized by the data randomizer **251** or on the data bypassing the data randomizer **251**, so as to add 20 bytes of RS parity data. Thereafter, the processed data are outputted to the data interleaver **253**. Herein, if the inputted data correspond to the main service data packet, the RS encoder/non-systematic RS encoder **252** performs the same systematic RS encoding process as that of the conventional broadcasting system, thereby adding the 20-byte RS parity data at the end of the 187-byte data. Alternatively, if the inputted data correspond to the mobile service data packet, the RS encoder/non-systematic RS encoder **252** performs a non-systematic RS encoding process. At this point, the 20-byte RS parity data obtained from the non-systematic RS encoding process are inserted in a pre-decided parity byte place within the mobile service data packet.

The data interleaver **253** corresponds to a byte unit convolutional interleaver. The output of the data interleaver **253** is inputted to the parity replacer **254** and to the non-systematic RS encoder **255**.

Meanwhile, a process of initializing a memory within the trellis encoding module **256** is primarily required in order to decide the output data of the trellis encoding module **256**, which is located after the parity replacer **254**, as the known data pre-defined according to an agreement between the receiving system and the transmitting system. More specifically, the memory of the trellis encoding module **256** should first be initialized before the received known data sequence is trellis-encoded.

At this point, the beginning portion of the known data sequence that is received corresponds to the initialization data and not to the actual known data. Herein, the initialization data has been included in the data by the group formatter within the pre-processor **230** in an earlier process. Therefore, the process of replacing the initialization data with memory values within the trellis encoding module **256** are required to be performed immediately before the inputted known data sequence is trellis-encoded.

More specifically, the initialization data are replaced with the memory value within the trellis encoding module **256**, thereby being inputted to the trellis encoding module **256**. At this point, the memory value replacing the initialization data are process with (or calculated by) an exclusive OR (XOR) operation with the respective memory value within the trellis encoding module **256**, so as to be inputted to the corresponding memory. Therefore, the corresponding memory is initialized to '0'. Additionally, a process of using the memory value replacing the initialization data to re-calculate the RS parity, so that the re-calculated RS parity value can replace the RS parity being outputted from the data interleaver **253**, is also required.

Therefore, the non-systematic RS encoder **255** receives the mobile service data packet including the initialization data from the data interleaver **253** and also receives the memory value from the trellis encoding module **256**.

Among the inputted mobile service data packet, the initialization data are replaced with the memory value, and the RS parity data that are added to the mobile service data packet are removed and processed with non-systematic RS encoding. Thereafter, the new RS parity obtained by performing the non-systematic RS encoding process is outputted to the parity replacer **255**. Accordingly, the parity replacer **255** selects the output of the data interleaver **253** as the data within the mobile service data packet, and the parity replacer **255** selects the output of the non-systematic RS encoder **255** as the RS parity. The selected data are then outputted to the trellis encoding module **256**.

Meanwhile, if the main service data packet is inputted or if the mobile service data packet, which does not include any initialization data that are to be replaced, is inputted, the parity replacer **254** selects the data and RS parity that are outputted from the data interleaver **253**. Then, the parity replacer **254** directly outputs the selected data to the trellis

encoding module **256** without any modification. The trellis encoding module **256** converts the byte-unit data to symbol units and performs a 12-way interleaving process so as to trellis-encode the received data. Thereafter, the processed data are outputted to the synchronization multiplexer **260**.

FIG. **32** illustrates a detailed diagram of one of 12 trellis encoders included in the trellis encoding module **256**. Herein, the trellis encoder includes first and second multiplexers **531** and **541**, first and second exclusive OR (XOR) gates **532** and **542**, and first to third memories **533**, **542**, and **544**.

More specifically, the first to third memories **533**, **542**, and **544** are initialized by the memory value instead of the initialization data from the parity replacer **254**. More specifically, when the first symbol (i.e., two bits), which are converted from initialization data (i.e., each trellis initialization data byte), are inputted, the input bits of the trellis encoder will be replaced by the memory values of the trellis encoder, as shown in FIG. **32**.

Since 2 symbols (i.e., 4 bits) are required for trellis initialization, the last 2 symbols (i.e., 4 bits) from the trellis initialization bytes are not used for trellis initialization and are considered as a symbol from a known data byte and processed accordingly.

When the trellis encoder is in the initialization mode, the input comes from an internal trellis status (or state) and not from the parity replacer **254**. When the trellis encoder is in the normal mode, the input symbol (X**2**X**1**) provided from the parity replacer **254** will be processed. The trellis encoder provides the converted (or modified) input data for trellis initialization to the non-systematic RS encoder **255**.

More specifically, when a selection signal designates a normal mode, the first multiplexer **531** selects an upper bit X**2** of the input symbol. And, when a selection signal designates an initialization mode, the first multiplexer **531** selects the output of the first memory **533** and outputs the selected output data to the first XOR gate **532**. The first XOR gate **532** performs XOR operation on the output of the first multiplexer **531** and the output of the first memory **533**, thereby outputting the added result to the first memory **533** and, at the same time, as a most significant (or uppermost) bit Z**2**. The first memory **533** delays the output data of the first XOR gate **532** by 1 clock, thereby outputting the delayed data to the first multiplexer **531** and the first XOR gate **532**.

Meanwhile, when a selection signal designates a normal mode, the second multiplexer **541** selects a lower bit X**1** of the input symbol. And, when a selection signal designates an initialization mode, the second multiplexer **541** selects the output of the second memory **542**, thereby outputting the selected result to the second XOR gate **543** and, at the same time, as a lower bit Z**1**. The second XOR gate **543** performs XOR operation on the output of the second multiplexer **541** and the output of the second memory **542**, thereby outputting the added result to the third memory **544**. The third memory **544** delays the output data of the second XOR gate **543** by 1 clock, thereby outputting the delayed data to the second memory **542** and, at the same time, as a least significant (or lowermost) bit Z**0**. The second memory **542** delays the output data of the third memory **544** by 1 clock, thereby outputting the delayed data to the second XOR gate **543** and the second multiplexer **541**.

The select signal designates an initialization mode during the first two symbols that are converted from the initialization data.

For example, when the select signal designates an initialization mode, the first XOR gate **532** performs an XOR operation on the value of the first memory **533**, which is provided through the first multiplexer **531**, and on a memory value that is directly provided from the first memory **533**. That is, the first XOR gate **532** performs an XOR operation on 2 bits having the same value. Generally, when only one of the two bits belonging to the operand is '1', the result of the XOR gate is equal to '1'. Otherwise, the result of the XOR gate becomes equal to '0'. Therefore, when the value of the first memory **533** is processed with an XOR operation, the result is always equal to '0'. Furthermore, since the output of the first XOR gate **532**, i.e., '0', is inputted to the first memory **533**, the first memory **533** is initialized to '0'.

Similarly, when the select signal designates an initialization mode, the second XOR gate **543** performs an XOR operation on the value of the second memory **542**, which is provided through the second multiplexer **541**, and on a memory value that is directly provided from the second memory **542**. Therefore, the output of the second XOR gate **543** is also always equal to '0'. Since the output of the second XOR gate **543**, i.e., '0', is inputted to the third memory **544**, the third memory **544** is also initialized to '0'. The output of the third memory **544** is inputted to the second memory **542** in the next clock, thereby initializing the second memory **542** to '0'. In this case also, the select signal designates the initialization mode.

More specifically, when the first symbol being converted from the initialization data byte replaces the values of the first memory **533** and the second memory **542**, thereby being inputted to the trellis encoder, each of the first and third memories **533** and **544** within the trellis encoder is initialized to '00'. Following the process, when the second symbol being converted from the initialization data byte replaces the values of the first memory **533** and the second memory **542**, thereby being inputted to the trellis encoder, each of the first, second, and third memories **533**, **542**, and **544** within the trellis encoder is initialized to '000'.

As described above, 2 symbols are required to initialize the memory of the trellis encoder. At this point, while the select signal designates an initialization mode, the output bits (X**2**'X**1**') of the first and second memories **533** and **542** are inputted to the non-systematic RS encoder **255**, so as to perform a new RS parity calculation process.

The synchronization multiplexer **260** inserts a field synchronization signal and a segment synchronization signal to the data outputted from the trellis encoding module **256** and, then, outputs the processed data to the pilot inserter **271** of the transmission unit **270**. Herein, the data having a pilot inserted therein by the pilot inserter **271** are modulated by the modulator **272** in accordance with a pre-determined modulating method (e.g., a VSB method). Thereafter, the modulated data are transmitted to each receiving system though the radio frequency (RF) up-converter **273**.

Multiplexing Method of Packet Multiplexer

Data of the error correction encoded and 1/H-rate encoded primary RS frame (i.e., when the RS frame mode value is equal to '00') or primary/secondary RS frame (i.e., when the RS frame mode value is equal to '01'), are divided into a plurality of data groups by the group formatter **303**. Then, the divided data portions are assigned to at least one of regions A to D of each data group or to an M/H block among the M/H blocks B**1** to B**10**, thereby being deinterleaved. Then, the deinterleaved data group passes through the packet formatter **305**, thereby being multiplexed with the main service data by the packet multiplexer **240** based upon a de-decided multiplexing rule. The packet multiplexer **240** multiplexes a plurality of consecutive data groups, so that the data groups are assigned to be spaced as far apart from

one another as possible within the sub-frame. For example, when it is assumed that 3 data groups are assigned to a sub-frame, the data groups are assigned to a 1st slot (Slot #**0**), a 5th slot (Slot #**4**), and a 9th slot (Slot #**8**) in the sub-frame, respectively.

As described-above, in the assignment of the plurality of consecutive data groups, a plurality of parades are multiplexed and outputted so as to be spaced as far apart from one another as possible within a sub-frame. For example, the method of assigning data groups and the method of assigning parades may be identically applied to all sub-frames for each M/H frame or differently applied to each M/H frame.

FIG. **10** illustrates an example of a plurality of data groups included in a single parade, wherein the number of data groups included in a sub-frame is equal to '3', and wherein the data groups are assigned to an M/H frame by the packet multiplexer **240**. Referring to FIG. **10**, 3 data groups are sequentially assigned to a sub-frame at a cycle period of 4 slots. Accordingly, when this process is equally performed in the 5 sub-frames included in the corresponding M/H frame, 15 data groups are assigned to a single M/H frame. Herein, the 15 data groups correspond to data groups included in a parade.

When data groups of a parade are assigned as shown in FIG. **10**, the packet multiplexer **240** may either assign main service data to each data group, or assign data groups corresponding to different parades between each data group. More specifically, the packet multiplexer **240** may assign data groups corresponding to multiple parades to one M/H frame. Basically, the method of assigning data groups corresponding to multiple parades is very similar to the method of assigning data groups corresponding to a single parade. In other words, the packet multiplexer **240** may assign data groups included in other parades to an M/H frame according to a cycle period of 4 slots. At this point, data groups of a different parade may be sequentially assigned to the respective slots in a circular method. Herein, the data groups are assigned to slots starting from the ones to which data groups of the previous parade have not yet been assigned. For example, when it is assumed that data groups corresponding to a parade are assigned as shown in FIG. **10**, data groups corresponding to the next parade may be assigned to a sub-frame starting either from the 12th slot of a sub-frame.

FIG. **11** illustrates an example of assigning and transmitting 3 parades (Parade #**0**, Parade #**1**, and Parade #**2**) to an M/H frame. For example, when the 1st parade (Parade #**0**) includes 3 data groups for each sub-frame, the packet multiplexer **240** may obtain the positions of each data groups within the sub-frames by substituting values '0' to '2' for i in Equation 1. More specifically, the data groups of the 1st, parade (Parade #**0**) are sequentially assigned to the 1st, 5th, and 9th slots (Slot #**0**, Slot #**4**, and Slot #**8**) within the sub-frame. Also, when the 2nd parade includes 2 data groups for each sub-frame, the packet multiplexer **240** may obtain the positions of each data groups within the sub-frames by substituting values '3' and '4' for i in Equation 1. More specifically, the data groups of the 2nd parade (Parade #**1**) are sequentially assigned to the 2nd and 12th slots (Slot #**3** and Slot #**11**) within the sub-frame. Finally, when the 3rd parade includes 2 data groups for each sub-frame, the packet multiplexer **240** may obtain the positions of each data groups within the sub-frames by substituting values '5' and '6' for i in Equation 1. More specifically, the data groups of the 3rd parade (Parade #**2**) are sequentially assigned and outputted to the 7th and 11th slots (Slot #**6** and Slot #**10**) within the sub-frame.

As described above, the packet multiplexer **240** may multiplex and output data groups of multiple parades to a single M/H frame, and, in each sub-frame, the multiplexing process of the data groups may be performed serially with a group space of 4 slots from left to right. Therefore, a number of groups of one parade per sub-frame (NOG) may correspond to any one integer from '1' to '8'. Herein, since one M/H frame includes 5 sub-frames, the total number of data groups within a parade that can be allocated to an M/H frame may correspond to any one multiple of '5' ranging from '5' to '40'.

Signaling Information Encoding

The present invention assigns signaling information areas for inserting signaling information to some areas within each data group. FIG. **33** illustrates an example of assigning signaling information areas for inserting signaling information starting from the 1st segment of the 4th M/H block (B**4**) to a portion of the 2nd segment. More specifically, 276 (=207+69) bytes of the 4th M/H block (B**4**) in each data group are assigned as the signaling information area. In other words, the signaling information area consists of 207 bytes of the 1st segment and the first 69 bytes of the 2nd segment of the 4th M/H block (B**4**). For example, the 1st segment of the 4th M/H block (B**4**) corresponds to the 17th or 173rd segment of a VSB field.

The signaling information that is to be inserted in the signaling information area is FEC-encoded by the signaling encoder **304**, thereby inputted to the group formatter **303**. The signaling information may include a transmission parameter which is included in the payload region of an OM packet, and then received to the demultiplexer **210**.

The group formatter **303** inserts the signaling information, which is FEC-encoded and outputted by the signaling encoder **304**, in the signaling information area within the data group. Herein, the signaling information may be identified by two different types of signaling channels: a transmission parameter channel (TPC) and a fast information channel (FIC).

Herein, the TPC data corresponds to signaling information including transmission parameters, such as RS frame information, RS encoding information, FIC information, data group information, SCCC information, and M/H frame information and so on. However, the TPC data presented herein is merely exemplary. And, since the adding or deleting of signaling information included in the TPC may be easily adjusted and modified by one skilled in the art, the present invention will, therefore, not be limited to the examples set forth herein.

Furthermore, the FIC data is provided to enable a fast service acquisition of data receivers, and the FIC data includes cross layer information between the physical layer and the upper layer(s).

FIG. **34** illustrates a detailed block diagram of the signaling encoder **304** according to the present invention. Referring to FIG. **34**, the signaling encoder **304** includes a TPC encoder **561**, an FIC encoder **562**, a block interleaver **563**, a multiplexer **564**, a signaling randomizer **565**, and an iterative turbo encoder **566**.

The TPC encoder **561** receives 10-bytes of TPC data and performs (18,10)-RS encoding on the 10-bytes of TPC data, thereby adding 8 bytes of RS parity data to the 10 bytes of TPC data. The 18 bytes of RS-encoded TPC data are outputted to the multiplexer **564**.

The FIC encoder **562** receives 37-bytes of FIC data and performs (51,37)-RS encoding on the 37-bytes of FIC data, thereby adding 14 bytes of RS parity data to the 37 bytes of FIC data. Thereafter, the 51 bytes of RS-encoded FIC data

are inputted to the block interleaver **563**, thereby being interleaved in predetermined block units. Herein, the block interleaver **563** corresponds to a variable length block interleaver. The block interleaver **563** interleaves the FIC data within each sub-frame in TNoG(column)×51(row) block units and then outputs the interleaved data to the multiplexer **564**. Herein, the TNoG corresponds to the total number of data groups being assigned to a sub-frame. The block interleaver **563** is synchronized with the first set of FIC data in each sub-frame.

The block interleaver **563** writes 51 bytes of incoming (or inputted) RS codewords in a row direction (i.e., row-by-row) and left-to-right and up-to-down directions and reads 51 bytes of RS codewords in a column direction (i.e., column-by-column) and left-to-right and up-to-down directions, thereby outputting the RS codewords.

The multiplexer **564** multiplexes the RS-encoded TPC data from the TPC encoder **561** and the block-interleaved FIC data from the block interleaver **563** along a time axis. Then, the multiplexer **564** outputs 69 bytes of the multiplexed data to the signaling randomizer **565**. The signaling randomizer **565** randomizes the multiplexed data and outputs the randomized data to the iterative turbo encoder **566**. The signaling randomizer **565** may use the same generator polynomial of the randomizer used for mobile service data. Also, initialization occurs in each data group.

The iterative turbo encoder **566** corresponds to an inner encoder performing iterative turbo encoding in a PCCC method on the randomized data (i.e., signaling information data).

For example, if the iterative turbo encoder **566** performs encoding of data at a coding rate of 1/4, 69 bytes applied to the iterative turbo encoder **566** are extended to 276 bytes by the iterative turbo-encoding process, such that the iterative turbo encoder **566** outputs the resultant 276 bytes. The 276 bytes generated from the iterative turbo encoder **566** are transferred to the group formatter **303**, such that they are inserted into a signaling information area of a corresponding data group.

FIG. **35** is a detailed block diagram illustrating the iterative turbo encoder **566** according to one embodiment of the present invention. The iterative turbo encoder **566** shown in FIG. **35** is used as a PCCC encoder having a coding rate of 1/4 (i.e., a 1/4-coding-rate PCCC encoder), which performs iterative turbo encoding on input data of one bit (i.e., 1-bit input data) using the PCCC scheme so as to convert the 1-bit input data into 4-bit output data.

The iterative turbo encoder **566** includes 6 identical even component encoders and 6 identical odd component encoders. In other words, the iterative turbo encoder **566** includes 6-way even component encoders and 6-way odd component encoders.

Referring to FIG. **35**, the iterative turbo encoder **566** includes a byte-to-bit converter **571**, a bit delay **572**, an even encoder **573**, a bit interleaver **574**, an odd encoder **575**, and a multiplexer **576**.

The even encoder **573** includes a demultiplexer, 6 even component encoders, 6 odd symbol-to-byte converters, and a multiplexer. In this case, the demultiplexer performs switching of input bits such that it outputs the switched input bits to any one of 6 branches (or 6 paths). The 6 even component encoders are connected to 6 branches (or 6 paths) of the demultiplexer, respectively. The 6 symbol-to-byte converters are connected to the 6 even component encoders, respectively. The multiplexer temporally multiplexes output data of the 6 symbol-to-byte converters.

The odd encoder **575** includes a demultiplexer, 6 odd component encoders, 6 symbol-to-byte converters, and a multiplexer. In this case, the demultiplexer performs switching of input bits such that it outputs the switched input bits to any one of 6 branches (or 6 paths). The 6 odd component encoders are connected to 6 branches (or 6 paths) of the demultiplexer, respectively. The 6 symbol-to-byte converters are connected to the 6 odd component encoders, respectively. The multiplexer temporally multiplexes output data of the 6 symbol-to-byte converters. In this case, the coding rate of each component encoder contained in each of the even encoder **573** and the odd encoder **575** is set to 1/H1 (where H1≥1). For convenience of description and better understanding of the present invention, H1 is set to 2 in this embodiment of the present invention.

In other words, output data bits generated from the byte-to-bit converter **571** are simultaneously input to the even encoder **573** and the odd encoder **575**, and each component encoder contained in each of the even encoder **573** and the odd encoder **575** has a coding rate of 1/2, so that an overall coding rate of the PCCC encoder shown in FIG. **35** is set to 1/4.

The byte-to-bit converter **571** classifies an input data byte into data bits, and outputs the data bits to the bit delay **572** and the bit interleaver **574**. For example, 8 bits contained in one byte are sequentially output from a most significant bit (MSB). The bit delay **572** delays input data bits, and outputs the delayed data bits to the even encoder **573**. The bit interleaver **574** performs interleaving of input data bits in units of a predetermined PCCC block, and outputs the interleaved input data bits to the odd encoder **575**. In accordance with one embodiment of the present invention, a length of the PCCC block of the bit interleaver **574** is equal to the length of 552 bits (=69×8), and an interleaving rule of the bit interleaver **574** conforms to the interleaving rule of the symbol interleaver of a block processor.

The bit delay **572** delays input data bits by 552 bits so that the delayed input data bits can be matched with the delay of 552 bits in the bit interleaver **574**. As a result, the bit delay **572** delays input data bits by 552 bits, and outputs the delayed result to the even encoder **573**.

The same data bits are bit-interleaved and then encoded by the odd encoder **575**, whereas the data bits generated from the byte-to-bit converter **571** are directly encoded by the even encoder **573**. Thus, the 6-way even component encoders are parallel-concatenated with the 6-way odd component encoders as a result of the action of the bit interleaver **574**.

The demultiplexer of the even encoder **573** outputs the delayed data bits by the bit delay **572** to any one of the 6 even component encoders. The even component encoder being received data bit through the demultiplexer encodes the received data bit at a coding rate of 1/2, such that they(i.e., 2 output data bits) output the encoded result to a corresponding symbol-to-byte converter. That is, the even component encoder encodes one bit (1 bit) into 2 bits and outputs the 2 bits, such that an output unit of the even component encoder is considered to be a symbol. In other words, one symbol is configured of 2 bits. The symbol-to-byte converter collects input symbols, constructs a byte composed of the collected input symbols, and outputs the constructed byte to the multiplexer. That is, the symbol-to-byte converter collects four symbols generated from the even component encoder, such that it constructs the byte composed of the collected four symbols. The multiplexer outputs the data byte generated from the symbol-to-byte converter to the multiplexer **576**. That is, the multiplexer

contained in the even encoder **573** is used as a (6×1) multiplexer, such that it selects one of output signals of the 6 even component encoders and outputs the selected output signal to the multiplexer **576**.

The demultiplexer of the odd encoder **575** receives the interleaved data bits from the bit interleaver **574**, and outputs the received data bits to any one of the 6 odd component encoders. The odd component encoder being received data bits through the demultiplexer encodes the received data bit at a coding rate of 1/2, and outputs the encoded result to a corresponding symbol-to-byte converter. In other words, the odd component encoder encodes a single bit into 2 bits and outputs the 2 bits, such that an output unit of the odd component encoder is considered to be a symbol. That is, a single symbol is composed of 2 bits. The symbol-to-byte converter collects input symbols, constructs a byte composed of the collected input symbols, and outputs the constructed byte to the multiplexer. The multiplexer outputs the byte generated from the symbol-to-byte converter to the multiplexer **576**. That is, the multiplexer contained in the odd encoder **575** is used as a (6×1) multiplexer, such that it selects one of the output signals of the 6 odd component encoders and outputs the selected output signal to the multiplexer **576**.

The multiplexer **576** is used as a (2×1) multiplexer, selects either of the output data of the even encoder **573** and the other output data of the odd encoder **575**, and outputs the selected output data to the group formatter **303**. In other words, the multiplexer **576** alternately selects the output data of the even encoder **573** and the other output data of the odd encoder **575**, and outputs the selected result to the group formatter **303**.

In this case, the PCCC encoder shown in FIG. **35** is virtually parallel-concatenated with a trellis encoder of the trellis encoding module **256** according to a parallel turbo-code scheme, such that it forms a parallel turbo code. Indeed, although the transmission system includes a plurality of blocks between the signaling encoder **304** and the trellis encoding module **256**, the receiving system considers two blocks to be concatenated with each other, and decodes the plurality of blocks in consideration of the above two blocks concatenated with each other.

Therefore, output bytes of each component encoder contained in the iterative turbo encoder **566** are transmitted to each of 12 trellis encoders contained in the trellis encoding module. For example, output bytes of the sixth even component encoder of the even encoder **573** may be arranged to be input to the sixth trellis encoder of the trellis encoding module. That is, each component encoder is paired with a specific trellis encoder from among 12 trellis encoders contained in the trellis encoding module. In addition, the demultiplexer and the multiplexer contained in the even encoder **573** and the odd encoder **575**, and the multiplexer **576** are each used to map an output byte of each component encoder to a corresponding trellis encoder of the trellis encoding module **256**.

Detailed structures of the even component encoder may be equal to or different from those of the odd component encoder. For convenience of description and better understanding of the present invention, it is assumed that detailed structures of the even component encoder may be different from those of the odd component encoder.

FIG. **36**(a) is a detailed block diagram illustrating the even component encoder according to an embodiment of the present invention. FIG. **36**(b) is a detailed block diagram illustrating the odd component encoder according to an embodiment of the present invention.

The even component encoder shown in FIG. **36**(a) includes two adders **581** and **582** and two memories **583** and **584**, such that it encodes an input bit 'x' and outputs the encoded result as 2 bits 'X**2**' and 'X**1**'.

In other words, the adder **581** adds the input bit 'x' and output data of the memory **583**, and outputs the added result as a high-order bit 'X**2**'.

In addition, the adder **582** adds the input bit 'x' and output data of the memory **584**, and outputs the added result to the memory **583**. The memory **583** delays output data of the adder **582** by a predetermined time (e.g., one clock), and outputs the delayed result to the adder **581** and the memory **584**. The memory **584** delays output data of the memory **583** by a predetermined time (e.g., one clock), and outputs the delayed result to the adder **581** and at the same time outputs the same delayed result as a low-order bit 'X**1**'.

The odd component encoder shown in FIG. **36**(b) includes three adders **581**, **582**, and **585** and two memories **583** and **584**, such that it encodes an input bit 'x' and outputs the encoded result as 2 bits X**2** and X**1**. In other words, the adder **581** adds the input bit 'x' and output data of the memory **583**, and outputs the added result as a high-order bit 'X**2**'.

In addition, the adder **582** adds the input bit 'x' and output data of the memory **584**, and outputs the added result to the memory **583**. The memory **583** delays output data of the adder **582** by a predetermined time (e.g., one clock), and outputs the delayed result to the adder **581** and the memory **584**. The memory **584** delays output data of the memory **583** by a predetermined time (e.g., one clock), and outputs the delayed result to the adder **581** and the other adder **585**. The adder **585** adds the output data of the memory **583** and the output data of the memory **584**, and outputs the added result as a low-order bit 'X**1**'.

FIG. **37**(a) illustrates an exemplary case in which a trellis encoder is serially concatenated with the even component encoder shown in FIG. **36**(a).

In fact, although a plurality of blocks are located between the even component encoder and the trellis encoder, the receiving system considers two blocks to be concatenated with each other, so that it decodes data. In other words, the trellis encoder performs precoding on the high-order bit 'X**2**' generated from the even component encoder of FIG. **36**(a), and outputs the precoded result as a most significant bit 'Z**2**'. In addition, the trellis encoder performs trellis-encoding on the low-order bit 'X**1**', so that it outputs the trellis-encoded result as two output bits Z**1** and Z**0**.

FIG. **37**(b) illustrates an exemplary case in which a trellis encoder is serially concatenated with the odd component encoder shown in FIG. **36**(b).

In fact, although a plurality of blocks are located between the even component encoder and the trellis encoder, the receiving system considers two blocks to be concatenated with each other, so that it decodes data. The trellis encoder performs precoding on the high-order bit 'X**2**' generated from the odd component encoder of FIG. **36**(b), and outputs the precoded result as a most significant bit 'Z**2**'. In addition, the trellis encoder performs trellis-encoding on the low-order bit 'X', so that it outputs the trellis-encoded result as two output bits Z**1** and Z**0**.

FIG. **38** illustrates an example of a syntax structure of TPC data being inputted to the TPC encoder **561**. The TPC data are inserted in the signaling information area of each data group and then transmitted. The TPC data may include a sub-frame_number field, a slot_number field, a parade_id field, a starting_group_number (SGN) field, a number_of_groups (NoG) field, a parade_repetition_cycle (PRC) field,

an RS_frame_mode field, an RS_code_mode_primary field, an RS_code_mode_secondary field, an SCCC_block_mode field, an SCCC_outer_code_mode_A field, an SCCC_outer_code_mode_B field, an SCCC_outer_code_mode_C field, an SCCC_outer_code_mode_D field, an FIC_version field, a parade_continuity_counter field, and a TNoG field.

The Sub-Frame_number field corresponds to the current Sub-Frame number within the M/H frame, which is transmitted for M/H frame synchronization. The value of the Sub-Frame_number field may range from 0 to 4. The Slot_number field indicates the current slot number within the sub-frame, which is transmitted for M/H frame synchronization. Also, the value of the Sub-Frame_number field may range from 0 to 15. The Parade_id field identifies the parade to which this group belongs. The value of this field may be any 7-bit value. Each parade in a M/H transmission shall have a unique Parade_id field.

Communication of the Parade_id between the physical layer and the management layer may be performed by means of an Ensemble_id field formed by adding one bit to the left of the Parade_id field. If the Ensemble_id field is used for the primary Ensemble delivered through this parade, the added MSB shall be equal to '0'. Otherwise, if the Ensemble_id field is used for the secondary ensemble, the added MSB shall be equal to '1'. Assignment of the Parade_id field values may occur at a convenient level of the system, usually in the management layer. The starting_group_number (SGN) field shall be the first Slot_number for a parade to which this group belongs, as determined by Equation 1 (i.e., after the Slot numbers for all preceding parades have been calculated). The SGN and NoG shall be used according to Equation 1 to obtain the slot numbers to be allocated to a parade within the sub-frame.

The number_of_Groups (NoG) field shall be the number of groups in a sub-frame assigned to the parade to which this group belongs, minus 1, e.g., NoG=0 implies that one group is allocated (or assigned) to this parade in a sub-frame. The value of NoG may range from 0 to 7. This limits the amount of data that a parade may take from the main (legacy) service data, and consequently the maximum data that can be carried by one parade. The slot numbers assigned to the corresponding Parade can be calculated from SGN and NoG, using Equation 1. By taking each parade in sequence, the specific slots for each parade will be determined, and consequently the SGN for each succeeding parade. For example, if for a specific parade SGN=3 and NoG=3 (010b for 3-bit field of NoG), substituting i=3, 4, and 5 in Equation 1 provides slot numbers **12**, **2**, and **6**.

The Parade_repetition_cycle (PRC) field corresponds to the cycle time over which the parade is transmitted, minus 1, specified in units of M/H frames, as described in Table 12.

TABLE 12

PRC	Description
000	This parade shall be transmitted once every M/H frame.
001	This parade shall be transmitted once every 2 M/H frames.
010	This parade shall be transmitted once every 3 M/H frames.
011	This parade shall be transmitted once every 4 M/H frames.
100	This parade shall be transmitted once every 5 M/H frames.
101	This parade shall be transmitted once every 6 M/H frames.
110	This parade shall be transmitted once every 7 M/H frames.
111	Reserved

For example, if PRC field value is equal to '001', this indicates that the parade shall be transmitted once every 2 M/H frame.

The RS_Frame_mode field shall be as defined in Table 1. The RS_Frame_mode field represents that one parade transmits one RS frame or two RS frames.

The RS_code_mode_primary field shall be the RS code mode for the primary RS frame. Herein, the RS_code_mode_primary field is defined in Table 6.

The RS_code_mode_secondary field shall be the RS code mode for the secondary RS frame. Herein, the RS_code_mode_secondary field is defined in Table 6.

The SCCC_Block_mode field represents how M/H blocks within a data group are assigned to SCCC block. The SCCC_Block_mode field shall be as defined in Table 7.

The SCCC_outer_code_mode_A field corresponds to the SCCC outer code mode for Region A within a data group. The SCCC outer code mode is defined in Table 8.

The SCCC_outer_code_mode_B field corresponds to the SCCC outer code mode for Region B within the data group. The SCCC_outer_code_mode_C field corresponds be the SCCC outer code mode for Region C within the data group. And, the SCCC_outer_code_mode_D field corresponds to the SCCC outer code mode for Region D within the data group.

The FIC_version field represents a version of FIC data.

The Parade_continuity_counter field counter may increase from 0 to 15 and then repeat its cycle. This counter shall increment by 1 every (PRC+1) M/H frames. For example, as shown in Table 12, PRC=011 (decimal 3) implies that Parade_continuity_counter increases every fourth M/H frame.

The TNoG field may be identical for all sub-frames in an M/H Frame.

However, the information included in the TPC data presented herein is merely exemplary. And, since the adding or deleting of information included in the TPC may be easily adjusted and modified by one skilled in the art, the present invention will, therefore, not be limited to the examples set forth herein.

Since the TPC data (excluding the Sub-Frame_number field and the Slot_number field) for each parade do not change their values during an M/H frame, the same information is repeatedly transmitted through all M/H groups belonging to the corresponding parade during an M/H frame. This allows very robust and reliable reception of the TPC data. Because the Sub-Frame_number and the Slot_number are increasing counter values, they also are robust due to the transmission of regularly expected values.

Furthermore, the FIC data is provided to enable a fast service acquisition of data receivers, and the FIC information includes cross layer information between the physical layer and the upper layer(s).

FIG. **39** illustrates an example of a transmission scenario of the TPC data and the FIC data. The values of the Sub-Frame_number field, Slot_number field, Parade_id field, Parade_repetition_cycle field, and Parade_continuity_counter field may corresponds to the current M/H frame throughout the 5 sub-frames within a specific M/H frame. Some of TPC parameters and FIC data are signaled in advance.

The SGN, NoG and all FEC modes may have values corresponding to the current M/H frame in the first two sub-frames. The SGN, NoG and all FEC modes may have values corresponding to the frame in which the parade next appears throughout the 3^{rd}, 4^{th} and 5^{th} sub-frames of the current M/H frame. This enables the M/H receivers to receive (or acquire) the transmission parameters in advance very reliably.

For example, when Parade_repetition_cycle='000', the values of the 3rd, 4th, and 5th sub-frames of the current M/H frame correspond to the next M/H frame. Also, when Parade_repetition_cycle='011', the values of the 3rd, 4th, and 5th sub-frames of the current M/H frame correspond to the 4th M/H frame and beyond.

The FIC_version field and the FIC_data field may have values that apply to the current M/H Frame during the 1st sub-frame and the 2nd sub-frame, and they shall have values corresponding to the M/H frame immediately following the current M/H frame during the 3rd, 4th, and 5th sub-frames of the current M/H frame.

Meanwhile, the receiving system may turn the power on only during a slot to which the data group of the designated (or desired) parade is assigned, and the receiving system may turn the power off during the remaining slots, thereby reducing power consumption of the receiving system. Such characteristic is particularly useful in portable or mobile receivers, which require low power consumption. For example, it is assumed that data groups of a 1st parade with NOG=3, a 2nd parade with NOG=2, and a 3rd parade with NOG=3 are assigned to one M/H frame, as shown in FIG. **40**. It is also assumed that the user has selected a mobile service included in the 1st parade using the keypad provided on the remote controller or terminal. In this case, the receiving system turns the power on only during a slot that data groups of the 1st parade is assigned, as shown in FIG. **40**, and turns the power off during the remaining slots, thereby reducing power consumption, as described above. At this point, the power is required to be turned on briefly earlier than the slot to which the actual designated data group is assigned (or allocated). This is to enable the tuner or demodulator to converge in advance.

Assignment of Known Data (or Training Signal)

In addition to the payload data, the M/H transmission system inserts long and regularly spaced training sequences into each group. The regularity is an especially useful feature since it provides the greatest possible benefit for a given number of training symbols in high-Doppler rate conditions. The length of the training sequences is also chosen to allow fast acquisition of the channel during bursted power-saving operation of the demodulator. Each group contains 6 training sequences. The training sequences are specified before trellis-encoding. The training sequences are then trellis-encoded and these trellis-encoded sequences also are known sequences. This is because the trellis encoder memories are initialized to pre-determined values at the beginning of each sequence. The form of the 6 training sequences at the byte level (before trellis-encoding) is shown in FIG. **41**. This is the arrangement of the training sequence at the group formatter **303**.

The 1st training sequence is located at the last 2 segments of the 3rd M/H block (B**3**). The 2nd training sequence may be inserted at the 2nd and 3rd segments of the 4th M/H block (B**4**). The 2nd training sequence is next to the signaling area, as shown in FIG. **5**. Then, the 3rd training sequence, the 4th training sequence, the 5th training sequence, and the 6th training sequence may be placed at the last 2 segments of the 4th, 5th, 6th, and 7th M/H blocks (B**4**, B**5**, B**6**, and B**7**), respectively. As shown in FIG. **41**, the 1st training sequence, the 3rd training sequence, the 4th training sequence, the 5th training sequence, and the 6th training sequence are spaced 16 segments apart from one another. Referring to FIG. **41**, the dotted area indicates trellis initialization data bytes, the lined area indicates training data bytes, and the white area includes other bytes such as the FEC-coded M/H service data bytes, FEC-coded signaling data, main service data bytes, RS parity data bytes (for backwards compatibility with legacy ATSC receivers) and/or dummy data bytes.

FIG. **42** illustrates the training sequences (at the symbol level) after trellis-encoding by the trellis encoder. Referring to FIG. **42**, the dotted area indicates data segment sync symbols, the lined area indicates training data symbols, and the white area includes other symbols, such as FEC-coded mobile service data symbols, FEC-coded signaling data, main service data symbols, RS parity data symbols (for backwards compatibility with legacy ATSC receivers), dummy data symbols, trellis initialization data symbols, and/or the first part of the training sequence data symbols. Due to the intra-segment interleaving of the trellis encoder, various types of data symbols will be mixed in the white area.

After the trellis-encoding process, the last 1416 (=588+828) symbols of the 1st training sequence, the 3rd training sequence, the 4th training sequence, the 5th training sequence, and the 6th training sequence commonly share the same data pattern. Including the data segment synchronization symbols in the middle of and after each sequence, the total length of each common training pattern is 1424 symbols. The 2nd training sequence has a first 528-symbol sequence and a second 528-symbol sequence that have the same data pattern. More specifically, the 528-symbol sequence is repeated after the 4-symbol data segment synchronization signal. At the end of each training sequence, the memory contents of the twelve modified trellis encoders shall be set to zero(0).

Receiving System

FIG. **43** is a block diagram illustrating a receiving system according to an embodiment of the present invention.

The receiving system of FIG. **43** includes an antenna **1300**, a tuner **1301**, a demodulating unit **1302**, a demultiplexer **1303**, a program table buffer **1304**, a program table decoder **1305**, a program table storage unit **1306**, a data handler **1307**, a middleware engine **1308**, an A/V decoder **1309**, an A/V post-processor **1310**, an application manager **1311**, and a user interface **1314**. The application manager **1311** may include a channel manager **1312** and a service manager **1313**.

In FIG. **43**, solid lines indicate data flows and dotted lines indicate control flows.

The tuner **1301** tunes to a frequency of a specific channel through any of an antenna, a cable, or a satellite and down-converts the frequency to an Intermediate Frequency (IF) signal and outputs the IF signal to the demodulating unit **1302**.

In one embodiment of the present invention, the tuner **1301** may select a frequency of a specific mobile broadcasting channel from among broadcasting channels transmitted via the antenna **1300**. For example, if it is assumed that the receiving system is a terminal having both a communication function such as a phone function and a broadcast function such as a mobile broadcasting function, the antenna **1300** may be used as a broadcasting antenna, and an additional communication antenna may also be included in the receiving system. That is, the broadcasting antenna may be physically different than the communication antenna. For another example, one antenna may be used as both the broadcasting antenna and the communication antenna. For still another example, a plurality of antennas having different polarization characteristics may be used as a substitute for the broadcasting antenna, so that a multi-path diversity scheme is made available. In this case, although a quality of a received broadcast signal increases in proportion to the number of used antennas, power consumption excessively

increases and the size of a space occupied by an overall system also increases. Therefore, it is preferable that a proper number of diversity antennas be used in consideration of the above-mentioned limitations.

Herein, the tuner **1301** is controlled by the channel manager **1312** in the application manager **1311** and reports the result and strength of a broadcast signal of the tuned channel to the channel manager **1312**. Data received through the frequency of the specific channel includes main service data, mobile service data, a transmission parameter, and program table information for decoding the main service data and the mobile service data.

The demodulating unit **1302** performs VSB demodulation, channel equalization, etc., on the signal output from the tuner **1301** and identifies and separately outputs main service data and mobile service data. The demodulating unit **1302** will be described in detail in a later.

On the other hand, the transmitter can transmit signaling information (or TPC information) including transmission parameters by inserting the signaling information into at least one of a field synchronization region, a known data region, and a mobile service data region. Accordingly, the demodulating unit **1302** can extract the transmission parameters from the field synchronization region, the known data region, and the mobile service data region.

The transmission parameters may include M/H frame information, sub-frame information, slot information, parade-related information (for example, a parade_id, a parade repeat period, etc.), information of data groups in a sub-frame, RS frame mode information, RS code mode information, SCCC block mode information, SCCC outer code mode information, FIC version information, etc.

The demodulating unit **1302** performs block decoding, RS frame decoding, etc., using the extracted transmission parameters. For example, the demodulating unit **1302** performs block decoding of each region in a data group with reference to SCCC-related information (for example, SCCC block mode information or an SCCC outer code mode) included in the transmission parameters and performs RS frame decoding of each region included in the data group with reference to RS-related information (for example, an RS code mode).

In the embodiment of the present invention, an RS frame including mobile service data demodulated by the demodulating unit **1302** is input to the demultiplexer **1303**.

That is, data inputted to the demultiplexer **1303** has an RS frame payload format as shown in FIG. **17**(a) or FIG. **17**(b). More specifically, the RS frame decoder of the demodulating unit **1302** performs the reverse of the encoding process performed at the RS frame encoder of the transmission system to correct errors in the RS frame and then outputs the error-corrected RS frame payload to a data derandomizer. The data derandomizer then performs derandomizing on the error-corrected RS frame payload through the reverse of the randomizing process performed at the transmission system to obtain an RS frame payload as shown in FIG. **17**(a) or FIG. **17**(b).

The demultiplexer **1303** may receive RS frame payloads of all parades and may also receive only an RS frame payload of a parade including a mobile service that the user desires to receive through power supply control. For example, when RS frame payloads of all parades are received, the demultiplexer **1303** can demultiplex a parade including a mobile service that the user desires to receive using a parade_id.

When one parade carries two RS frames, the demultiplexer **1303** needs to identify an RS frame carrying an ensemble including mobile service data to be decoded from a parade containing a mobile service that the user desires to receive. That is, when a received single parade or a parade demultiplexed from a plurality of parades carries a primary ensemble and a secondary ensemble, the demultiplexer **1303** selects one of the primary and secondary ensembles.

In an embodiment, the demultiplexer **1303** can demultiplex an RS frame carrying an ensemble including mobile service data to be decoded using an ensemble_id created by adding one bit to a left position of the parade_id.

The demultiplexer **1303** refers to the M/H header of the M/H service data packet within the RS frame payload belonging to the ensemble including the mobile service data that are to be decoded, thereby identifying when the corresponding M/H service data packet is the signaling table information or the IP datagram of the mobile service data. Alternatively, when the signaling table information and the mobile service data are both configured in the form of IP datagrams, the demultiplexer **1303** may use the IP address in order to identify the IP datagram of the program table information and the mobile service data.

Herein, the identified signaling table information is outputted to the program table buffer **1304**. And, audio/video/data streams are separated from the IP datagram of mobile service data that are to be selected among the IP datagrams of the identified mobile service data, thereby being respectively outputted to the A/V decoder **1309** and/or the data handler **1307**.

According to an embodiment of the present invention, when the stuff_indicator field within the M/H header of the M/H service data packet indicates that stuffing bytes are inserted in the payload of the corresponding M/H service data packet, the demultiplexer **1303** removes the stuffing bytes from the payload of the corresponding M/H service data packet. Then, the demultiplexer **1303** identifies the program table information and the mobile service data. Thereafter, the demultiplexer **1303** identifies A/V/D streams from the identified mobile service data.

The program table buffer **1304** temporarily stores the section-type program table information and then outputs the section-type program table information to the program table decoder **1305**.

The program table decoder **1305** identifies tables using a table_id and a section_length in the program table information and parses sections of the identified tables and produces and stores a database of the parsed results in the program table storage unit **1306**. For example, the program table decoder **1305** collects sections having the same table identifier (table_id) to construct a table. The program table decoder **1305** then parses the table and produces and stores a database of the parsed results in the program table storage unit **1306**.

The A/V decoder **1309** decodes the audio and video streams outputted from the demultiplexer **1303** using audio and video decoding algorithms, respectively. The decoded audio and video data is outputted to the A/V post-processor **1310**.

Here, at least one of an AC-3 decoding algorithm, an MPEG 2 audio decoding algorithm, an MPEG 4 audio decoding algorithm, an AAC decoding algorithm, an AAC+ decoding algorithm, an HE AAC decoding algorithm, an AAC SBR decoding algorithm, an MPEG surround decoding algorithm, and a BSAC decoding algorithm can be used as the audio decoding algorithm and at least one of an MPEG 2 video decoding algorithm, an MPEG 4 video decoding algorithm, an H.264 decoding algorithm, an SVC

decoding algorithm, and a VC-1 decoding algorithm can be used as the audio decoding algorithm.

The data handler **8507** processes data stream packets required for data broadcasting among data stream packets separated (or identified) by the demultiplexer **1303** and provides the processed data stream packets to the middleware engine **1310** to allow the middleware engine **1310** to be multiplexed them with A/V data. In an embodiment, the middleware engine **1310** is a Java middleware engine.

The application manager **1311** receives a key input from the TV viewer and displays a Graphical User Interface (GUI) on the TV screen in response to a viewer request through a User Interface (UI). The application manager **1311** also writes and reads information regarding overall GUI control of the TV, user requests, and TV system states to and from a memory (for example, NVRAM or flash memory). In addition, the application manager **1311** can receive parade-related information (for example, a parade_id) from the demodulating unit **1302** to control the demultiplexer **1303** to select an RS frame of a parade including a required mobile service. The application manager **1311** can also receive an ensemble_id to control the demultiplexer **1303** to select an RS frame of an ensemble including mobile service data to be decoded from the parade. The application manager **1311** also controls the channel manager **1312** to perform channel-related operations (for example, channel map management and program table decoder operations).

The channel manager **1312** manages physical and logical channel maps and controls the tuner **1301** and the program table decoder **1305** to respond to a channel-related request of the viewer. The channel manager also requests that the program table decoder **1305** parse a channel-related table of a channel to be tuned and receives the parsing results from the program table decoder **1305**.

Demodulating Unit within Receiving System

FIG. **44** illustrates an example of a demodulating unit in a digital broadcast receiving system according to the present invention. The demodulating unit of FIG. **44** uses known data information, which is inserted in the mobile service data section and, then, transmitted by the transmitting system, so as to perform carrier synchronization recovery, frame synchronization recovery, and channel equalization, thereby enhancing the receiving performance. Also the demodulating unit may turn the power on only during a slot to which the data group of the designated (or desired) parade is assigned, thereby reducing power consumption of the receiving system.

Referring to FIG. **44**, the demodulating unit includes an operation controller **2000**, a demodulator **2002**, an equalizer **2003**, a known sequence detector **2004**, a block decoder **2005**, and a RS frame decoder **2006**. The demodulating unit may further include a main service data processor **2008**. The main service data processor **2008** may include a data deinterleaver, a RS decoder, and a data derandomizer. The demodulating unit may further include a signaling decoder **2013**. The receiving system also may further include a power controller **5000** for controlling power supply of the demodulating unit.

More specifically, a frequency of a particular channel tuned by a tuner down converts to an intermediate frequency (IF) signal. Then, the down-converted data **2001** outputs the down-converted IF signal to the demodulator **2002** and the known sequence detector **2004**. At this point, the down-converted data **2001** is inputted to the demodulator **2002** and the known sequence detector **2004** via analog/digital converter ADC (not shown). The ADC converts pass-band analog IF signal into pass-band digital IF signal.

The demodulator **2002** performs self gain control, carrier recovery, and timing recovery processes on the inputted pass-band digital IF signal, thereby modifying the IF signal to a base-band signal. Then, the demodulator **2002** outputs the newly created base-band signal to the equalizer **2003** and the known sequence detector **2004**.

The equalizer **2003** compensates the distortion of the channel included in the demodulated signal and then outputs the error-compensated signal to the block decoder **2005**.

At this point, the known sequence detector **2004** detects the known sequence position information inserted by the transmitting end from the input/output data of the demodulator **2002** (i.e., the data prior to the demodulation process or the data after the demodulation process). Thereafter, the position information along with the symbol sequence of the known data, which are generated from the detected position, is outputted to the operation controller **2000**, the demodulator **2002**, the equalizer **2003**, and the signaling decoder **2013**. Also, the known sequence detector **2004** outputs a set of information to the block decoder **2005**. This set of information is used to allow the block decoder **2005** of the receiving system to identify the mobile service data that are processed with additional encoding from the transmitting system and the main service data that are not processed with additional encoding.

In addition, although the connection status is not shown in FIG. **44**, the information detected from the known sequence detector **2004** may be used throughout the entire receiving system and may also be used in the RS frame decoder **2006**.

The data demodulated in the demodulator **2002** or the data equalized in the channel equalizer **2003** is inputted to the signaling decoder **2013**. The known data position information detected in the known sequence detector **2004** is inputted to the signaling decoder **2013**.

Signaling Information Decoding

The signaling decoder **2013** extracts and decodes signaling information (e.g., TPC information, and FIC information), which inserted and transmitted by the transmitting end, from the inputted data, the decoded signaling information provides to blocks requiring the signaling information.

More specifically, the signaling decoder **2013** extracts and decodes TPC data and FIC data, which inserted and transmitted by the transmitting end, from the equalized data, and then the decoded TPC data and FIC data outputs to the operation controller **2000**, the known sequence detector **2004**, and the power controller **5000**. For example, the TPC data and FIC data is inserted in a signaling information region of each data group, and then is transmitted to a receiving system.

The signaling decoder **2013** performs signaling decoding as an inverse process of the signaling encoder shown in FIG. **34**, so as to extract TPC data and FIC data. For example, the signaling decoder **2013** decodes the inputted data using the PCCC method and derandomizes the decoded data, thereby dividing the derandomized data into TPC data and FIC data. At this point, the signaling decoder **2013** performs RS-decoding on the divided TPC data, so as to correct the errors occurring in the TPC data. Subsequently, the signaling decoder **2013** deinterleaves the divided FIC data and then performs RS-decoding on the deinterleaved FIC data, so as to correct the error occurring in the FIC data. The error-corrected TPC data are then outputted to the operation controller **2000**, the known sequence detector **2004**, and the power controller **5000**.

The TPC data may also include a transmission parameter which is inserted into the payload region of an OM packet by the service multiplexer **100**, and then is transmitted to transmitter **200**.

Herein, the TPC data may include RS frame information, SCCC information, M/H frame information, and so on, as shown in FIG. **38**. The RS frame information may include RS frame mode information and RS code mode information. The SCCC information may include SCCC block mode information and SCCC outer code mode information. The M/H frame information may include M/H frame index information, and the TPC data may include sub-frame count information, slot count information, parade_id information, SGN information, NoG information, and so on.

At this time, the signaling information area within the data group can be identified using known data information output from the known data detector **2004**. The signaling information area is located from the first segment of M/H block B**4** within the data group to a part of the second segment as shown in FIG. **33**. Namely, in the present invention, 276 (=207+69) bytes of the M/H block B**4** of each data group are assigned to an area for inserting signaling information. In other words, the signaling information area includes 207 bytes of the first segment of the M/H block B**4** and first 69 bytes of the second segment of the M/H block B**4**. The first known data sequence (i.e., first training sequence) is inserted into the last two segments of the M/H block B**3**, and the second known data sequence (i.e., second training sequence) is inserted into the second and third segments of the M/H block B**4**. At this time, since the second known data sequence is received subsequently to the signaling information area, the signaling decoder **2013** can decode the signaling information of the signaling information area by extracting the same from the data output from the demodulator **2002** or the channel equalizer **2003**.

FIG. **45** is a block diagram illustrating a signaling decoder **2013** shown in FIG. **44** according to one embodiment of the present invention.

Referring to FIG. **45**, the signaling decoder **2013** performs iterative turbo decoding and RS decoding on signaling information area's data selected from among equalized data. The signaling decoder **2013** outputs transmission parameters (i.e., TPC data) to the operation controller **2000**, and outputs a fast information channel (FIC) data to an upper layer.

For this operation, the signaling decoder **2013** may include an iterative turbo decoder **2111**, a derandomizer **2112**, a demultiplexer **2113**, an RS decoder **2114**, a block deinterleaver **2115**, and an RS decoder **2116**.

FIG. **46** is a detailed block diagram illustrating the iterative turbo decoder **2111**. Referring to FIG. **46**, upon receiving the signaling information area's data from among the equalized data, a demultiplexer (DeMux) **2200** discriminates symbols corresponding to respective branches of the signaling encoder **304** of the transmission system, and outputs the discriminated symbols to buffers **2201** and **2401**, respectively.

The buffers **2201** and **2401** store input data corresponding to the signaling information area, and respectively repeatedly output the stored input data to the demultiplexers **2202** and **2402** during the turbo decoding process.

In accordance with one embodiment of the present invention, it is assumed that output data of the even encoder **573** in the signaling encoder of the transmission system is processed to be input to 0^{th}, 2^{nd}, . . . , 10^{th} trellis encoders (i.e., even number trellis encoders), and output data of the odd encoder **575** is processed to be input to 1^{st}, 3^{rd}, . . . , 11^{th} trellis encoders. In this case, the demultiplexer **2202** outputs output data of the buffer **2201** to a trellis decoder (i.e., TCM decoder) corresponding to the even number trellis encoder. The demultiplexer **2202** receives data fed back from the block deinterleaver **2507**, and outputs the feed-back data to the same trellis decoder (i.e., TCM decoder) corresponding to the even number trellis encoder.

In this case, output data of each trellis decoder (TCM decoder) corresponds to a log likelihood ratio (LLR) value. The LLR value is a result from taking a logarithm of a soft decision value. More specifically, the LLR value corresponds to a log likelihood ratio (LLR) between a likelihood of input bit being equal to '1' and a likelihood of input bit being equal to '0'. An initial value of the LLR is set to zero. The LLR value is transferred to the even component decoder corresponding to the even component encoder contained in the signaling encoder of the transmission system. Input/output (I/O) data of the even component decoder is such an LLR value as well. In this case, since a single even number trellis decoder interoperates with a single even component decoder, an even component encoder and an even number trellis encoder are considered as a single encoder (effective component encoder) as shown in FIG. **37** (a). Hence, the even number trellis decoder and the even component decoder can be merged into a single effective component decoder. In the case where the two decoders configure a single decoder, decoding performance will be enhanced although complexity increases due to the increased number of states.

Output signals of the even component decoders **2300** to **2305** are sequentially transferred to the multiplexer **2306** and are then transferred to the block interleaver **2307**. The block interleaver **2307** has the same configuration as a block interleaver used for the signaling encoder of the transmitting side.

The LLR value block-interleaved by the block interleaver **2307** is fed back to the demultiplexer **2402**. The demultiplexer **2402** outputs the LLR value to a corresponding trellis decoder (i.e., TCM decoder) from among six trellis decoders, and at the same time transmits output data of the buffer **2401** to the trellis decoder. For example, provided that the LLR value fed back from the block deinterleaver **2507** is an LLR value of the first decoder **2500**, the demultiplexer **2402** outputs this feed-back LLR value and the output data of the buffer **2401** to the trellis decoder of the first decoder **2500**.

The above-mentioned rules are equally applied to the demultiplexer **2202**. The odd number trellis decoder and the odd component decoder can be operated in the same manner as in the even number trellis decoder and the even component decoder. Likewise, the odd number trellis decoder and the odd component decoder can be implemented as a single effective component decoder.

Output signals of the odd number decoders **2500** to **2505** are sequentially transferred to the multiplexer **2506**, and are then forwarded to the block deinterleaver **2507**. The block deinterleaver **2507** is an inverse process of the block interleaver. Thus, the LLR value block-deinterleaved by the block deinterleaver **2507** is input to the demultiplexer **2202** to accomplish the iterative turbo decoding.

After the iterative turbo decoding has been repeatedly performed at a predetermined level, the iterative turbo-decoded result is output to the derandomizer **2112**.

The derandomizer **2112** performs derandomizing of the iterative turbo-decoded data, and outputs the derandomized result to the demultiplexer **2113**. The demultiplexer (Demux) **2113** discriminates between TPC data composed of 18 bytes and FIC data composed of 51 bytes on the basis of the derandomized data.

Here, the TPC data is output to the RS decoder **2114** corresponding to an RS (18, 10) of a GF **256**. The RS decoder **2114** receives a result of hard decision from the iterative turbo decoder **2111** so as to perform general RS decoding, or the RS decoder **2114** receives the result of soft decision from the iterative turbo decoder **2111** so as to perform RS erasure decoding. TPC data (i.e., transmission parameter information) error-corrected by the RS decoder **2114** is output to the operation controller **2000**. In this case, the RS decoder **2114** further transmits the decision result to the operation controller **2000**, so that it prevents the occurrence of operational failure which may be generated from misjudgment of the transmission parameter. Also, since some information of the TPC data is repeatedly transmitted to each group, decoding performance can be improved using such a feature. For example, in case of FEC mode information such as SCCC or RS, since information of next M/H frame is repeatedly transmitted to three sub frames at the rear of one M/H frame, even though decoding is successfully performed once within the three subframes, there is no problem in receiving the next M/H frame.

The FIC data discriminated by the demultiplexer **2113** is output to a (TNoG×51) block deinterleaver **2115**. The block deinterleaver **2115** is an inverse process of the (TNoG×51) block interleaver of the signaling encoder of the transmitting side.

For example, the (TNoG×51) block interleaver of the transmitting side is a variable-length block interleaver, and interleaves FIC data contained in each subframe in units of a (TNoG (columns)×51 (rows)) block. In this case, 'TNoG' is indicative of a total number of data groups allocated to a subframe contained in a single M/H frame.

The FIC data block-deinterleaved by the block deinterleaver **2115** is input to the RS decoder **2116** corresponding to the RS (51, 37) of the GF **256**. In the same manner as in the RS decoder **2114** for TPC data, the RS decoder **2116** is able to use both the hard decision value and the soft decision value, and FIC data error-corrected by the RS decoder **2116** is output to an upper layer.

Meanwhile, TNoG value required by the block deinterleaver **2115** can be acquired from the TPC data output from the RS decoder **2114**. To this end, the block deinterleaver **2115** includes a controller.

However, since TNoG of next M/H frame is transmitted to three subframes at the rear of one M/H frame, information of TNoG of the current subframe may not be obtained through TPC data decoding. For example, if the broadcast receiver is turned on at the third subframe (sub-frame #**2**) and starts to perform FIC decoding to obtain channel information, and performs FIC block deinterleaving using TNoG within the TPC data, the broadcast receiver cannot decode the FIC data until it reaches the next M/H frame.

Accordingly, the present invention suggests a method for decoding FIC data by acquiring TNoG even without using RS-decoded TPC data.

FIG. **47** is a flow chart illustrating a process of decoding FIC data without using TPC data in accordance with one embodiment of the present invention.

FIG. **48** illustrates a detailed embodiment of a process of extracting TNoG in accordance with the present invention.

The process of acquiring TNoG according to the present invention may be performed by the signaling decoder **2013**, or may be performed by the operation controller **2000**. According to one embodiment of the present invention, TNoG is acquired by the signaling decoder **2013**. In particular, according to one embodiment of the present invention, a controller is provided at the block deinterleaver **2115** within the signaling decoder **2013**, and acquires TNoG. This is only one example, and the controller may be provided outside the block deinterleaver **2115**.

In other words, if a command to start FIC decoding is input (S**2601**), the signaling decoder **2013** searches start of next subframe (S**2602**). For example, it is supposed that a command to start FIC decoding is input at the middle of the n−1th subframe within one M/H frame as shown in (a) of FIG. **48**. In this case, start of the nth subframe is searched.

Namely, if the command to start FIC decoding is input, in order to extract start of the subframe, it is identified whether a data group exists in a corresponding slot. For example, 16 slots are assigned to one subframe. At this time, since known data exist in the data group, it is identified whether the data group exists in the corresponding slot as shown in (b) of FIG. **48** through correlation between a pre-determined pattern of known data and received data. As another example, information as to whether the data group exists in the corresponding slot may be provided from the operation controller **2000**.

At this time, if it is identified whether the data group exists in the corresponding slot, the signaling decoder **2013** performs turbo decoding, signaling derandomizing, and demultiplexing for data of the signaling information area within the data group to split TPC data, and performs RS decoding for the split TPC data. Then, the signaling decoder **2013** acquires a slot number from the RS-decoded TPC data as shown in (d) of FIG. **48**.

The slot number becomes 0 at a start slot of each sub frame, and has a value of 15 at the last slot of the corresponding subframe. Accordingly, start of the subframe can be identified by using the slot number.

In other words, the signaling decoder **2013** repeatedly performs the signaling decoding process until the slot number having a value of 0 is detected from the TPC data. If the data group within the subframe is assigned and transmitted as shown in (a) of FIG. **48** and the command to start FIC decoding is input at the middle of the n−1th subframe, start of the nth subframe is detected through the signaling decoding process. If the start of the subframe is detected, the group counter value is reset to 0.

If start of the nth subframe is detected through the above process, the signaling decoder **2013** detects a data group from the nth sub frame (S**2603**).

The presence of the data group may be identified using the correlation between the known data pattern and the received data, or may be provided from the operation controller **2000**.

If the data group is detected in the step S**2603**, the group counter value increases by 1 as shown in (f) of FIG. (S**2604**). The turbo decoder **2111** and the derandomizer **2112** perform turbo decoding and derandomizing for data of the signaling information area within the data group (S**2605**). Subsequently, the demultiplexer **2113** performs demultiplexing for the derandomized data to split TPC data from FIC data, and the RS decoder **2114** performs RS decoding for the split TPC data. The slot number is acquired from the RS-decoded TPC data. Also, the split FIC data (i.e., 51 bytes) are stored in a buffer (not shown) of the block deinterleaver **2115** (S**2606**).

The steps S**2604**, S**2605**, and S**2606** are performed whenever the data group is detected from the subframe to increase the group counter value by 1, and the buffer of the block deinterleaver **2115** stores the split FIC data by the demultiplexer **2113**.

This process is performed until the end of the subframe is detected. According to one embodiment of the present invention, the end of the subframe is detected using the slot number such as (d) of FIG. **48**. According to another

embodiment of the present invention, the end of the subframe is detected using the field synchronizing counter value such as (e) of FIG. **48**.

If the end of the subframe is identified (S**2607**), TNoG is calculated using the group counter value (S**2608**).

The TNoG value is applied to the FIC data stored in the buffer of the block deinterleaver **2115** to perform block deinterleaving (S**2609**). The block deinterleaved FIC data are input to the RS decoder **2116** and then RS-decoded by the RS decoder **2116**. In case of (g) and (h) of FIG. **48**, turbo decoding and derandomizing are performed for the FIC data included in each data group of the nth sub frame for the nth subframe interval and then stored in the buffer of the block deinterleaver **2115**. The TNoG calculated in the step S**2608** is applied to the FIC data of the nth subframe stored in the buffer of the block deinterleaver **2115** to perform block deinterleaving, RS decoding is performed for the block deinterleaved FIC data.

Meanwhile, in the step S**2607**, the end of the subframe may be detected using either the slot number such as (d) of FIG. **48** or the field synchronizing counter value such as (e) of FIG. **48**.

In other words, if the slot number acquired from the RS-decoded TPC data becomes 0, it means that a new subframe starts. Accordingly, if the slot number becomes 0, it is determined that the previous subframe has ended. In this case, since the group counter value increases by 1 in the step S**2604**, the value obtained by subtracting 1 from the group counter value becomes the TNoG value.

However, next subframe is the first subframe of new M/H frame, and data group may not exist in next M/H frame due to PRC. Under the circumstances, if the end of the subframe is detected using the slot number, TNoG cannot be identified until M/H frame where data group exists is detected, whereby FIC decoding time may be delayed. In this case, start of the subframe can be determined using the new slot number, and the number of field synchronization values can be counted to identify the end of the subframe. This is because that eight field synchronization values in one subframe and field synchronization values are transmitted regardless of the presence of the data group. For example, if the field synchronization counter value is 8, it is determined as the end of the subframe. In this case, the group counter value becomes the TNoG value.

The field synchronization values can also be detected through correlation. The field synchronization values can be detected from at least one of the signaling decoder **2013**, the operation controller **2000**, the demodulator **2002**, the known data detector **2004**, and a block which is not shown.

Meanwhile, the power controller **5000** is inputted the M/H frame-associated information from the signaling decoder **2013**, and controls power of the tuner and the demodulating unit. Alternatively, the power controller **5000** is inputted a power control information from the operation controller **2000**, and controls power of the tuner and the demodulating unit.

According to the embodiment of the present invention, the power controller **5000** turns the power on only during a slot to which a slot of the parade including user-selected mobile service is assigned. The power controller **5000** then turns the power off during the remaining slots.

For example, it is assumed that data groups of a 1^{st} parade with NOG=3, a 2^{nd} parade with NOG=2, and a 3^{rd} parade with NOG=3 are assigned to one M/H frame, as shown in FIG. **40**. It is also assumed that the user has selected a mobile service included in the 1^{st} parade using the keypad provided on the remote controller or terminal. In this case, the power controller **5000** turns the power on during a slot that data groups of the 1^{st} parade is assigned, as shown in FIG. **40**, and turns the power off during the remaining slots, thereby reducing power consumption.

The demodulator **2002** uses the known data symbol sequence during the timing and/or carrier recovery, thereby enhancing the demodulating performance. Similarly, the equalizer **2003** uses the known data so as to enhance the equalizing performance. Moreover, the decoding result of the block decoder **2005** may be fed-back to the equalizer **2003**, thereby enhancing the equalizing performance.

Demodulator and Known Sequence Detector

At this point, the transmitting system may receive a data frame (or VSB frame) including a data group which known data sequence (or training sequence) is periodically inserted therein, as shown in FIG. **5**. Herein, the data group is divided into regions A to D, as shown in FIG. **5**. More specifically, in the example of the present invention, each region A, B, C, and D are further divided into M/H blocks B**4** to B**7**, M/H blocks B**3** and B**8**, M/H blocks B**2** and B**9**, M/H blocks B**1** and B**10**, respectively.

Referring to FIG. **41** and FIG. **42**, known data sequence having the same pattern are included in each known data section that is being periodically inserted. Herein, the length of the known data sequence having identical data patterns may be either equal to or different from the length of the entire (or total) known data sequence of the corresponding known data section (or block). If the two lengths are different from one another, the length of the entire known data sequence should be longer than the length of the known data sequence having identical data patterns. In this case, the same known data sequences are included in the entire known data sequence.

As described above, when the known data are periodically inserted in-between the mobile service data, the channel equalizer of the receiving system may use the known data as training sequences, which may be used as accurate discriminant values. According to another embodiment of the present invention, the channel equalizer estimates a channel impulse response. Herein, the known data may be used in the process. According to yet another embodiment of the present invention, the channel equalizer may use the known data for updating filter coefficients (i.e., equalization coefficients).

Meanwhile, when known data sequence having the same pattern is periodically inserted, each known data sequence may be used as a guard interval in a channel equalizer according to the present invention. Herein, the guard interval prevents interference that occurs between blocks due to a multiple path channel. This is because the known data sequence located behind a mobile service data section (i.e., data block) may be considered as being copied in front of the mobile service data section.

The above-described structure is referred to as a cyclic prefix. This structure provides circular convolution in a time domain between a data block transmitted from the transmitting system and a channel impulse response. Accordingly, this facilitates the channel equalizer of the receiving system to perform channel equalization in a frequency domain by using a fast fourier transform (FFT) and an inverse fast fourier transform (IFFT).

More specifically, when viewed in the frequency domain, the data block received by the receiving system is expressed as a multiplication of the data block and the channel impulse response. Therefore, when performing the channel equalization, by multiplying the inverse of the channel in the frequency domain, the channel equalization may be performed more easily.

The known sequence detector **2004** detects the position of the known data being periodically inserted and transmitted as described above. At the same time, the known sequence detector **2004** may also estimate initial frequency offset during the process of detecting known data. In this case, the demodulator **2002** may estimate with more accuracy carrier frequency offset from the information on the known data position information and initial frequency offset estimation value, thereby compensating the estimated carrier frequency offset.

Meanwhile, when known data is transmitted, as shown in FIG. **5**, the known sequence detector **2004** detects a position of second known data region by using known data of the second known data region that the same pattern is repeated twice.

At this point, since the known sequence detector **2004** is well-informed of the data group structure, when the position of the second known data region is detected, the known sequence detector **2004** can estimate positions of the first, third, fourth, fifth, and sixth known data regions of a corresponding data group by counting symbols or segments based upon the second known data region position. If the corresponding data group is a data group including field synchronization segment, the known sequence detector **2004** can estimate the position of the field synchronization segment of the corresponding data group, which is positioned chronologically before the second known data region, by counting symbols or segments based upon the second known data region position. Also, the known sequence detector **2004** may estimate the known data position information and the field synchronization position information from the parade including mobile service selected by a user based on the M/H frame-associated information outputted from the signaling decoder **2013**.

At least one of the estimated known data poison information and field synchronization information is provided to the demodulator **2002**, the channel equalizer **2003**, the signaling decoder **2013**, and the operation controller **2000**.

Also, the known sequence detector **2004** may estimate initial frequency offset by using known data inserted in the second known data region (i.e., ACQ known data region). In this case, the demodulator **2002** may estimate with more accuracy carrier frequency offset from the information on the known data position information and initial frequency offset estimation value, thereby compensating the estimated carrier frequency offset.

Operation Controller

The operation controller **2000** receives the known data position information and the transmission parameter information and then forwards M/H frame time information, a presence or non-presence of a data group of a selected parade, position information of known data within the data group, power control information and the like to each block of the demodulating unit. The operation controller **2000**, as shown in FIG. **44**, controls operations of the demodulator **2002**, the channel equalizer **2003**, the block decoder **2005** and the RS frame decoder **2006**. And, the operation controller **2000** is able to overall operations of the demodulating unit (not shown in the drawing). Moreover, the operation controller **2000** can be implemented with the separate block or can be included within a prescribed one of the blocks of the demodulating unit shown in FIG. **44**.

FIG. **49** is an overall block diagram of an operation controller **2000**.

Referring to FIG. **49**, the operation controller **2000** can include a parade ID checker **3101**, a frame synchronizer **3102**, a parade mapper **3103**, a group controller **3104** and a known sequence indication controller **3105**.

The operation controller **2000** receives known data position information from the known sequence detector **2004** and receives transmission parameter information from the signaling decoder **2013**. The operation controller **2000** then generates a control signal necessary for a demodulating unit of a receiving system. For instance, the known data position information detected by the known sequence detector **2004** is inputted to the known sequence indication controller **3105**. And, the transmission parameter information (i.e., TPC data) decoded by the signaling decoder **2013** is inputted to the parade ID checker **3101**.

The parade ID checker **3101** compares a parade ID (parade ID selected by a user) contained in the user control signal to a parade ID inputted from the signaling decoder **2013**. If the two parade IDs are not identical to each other, the parade ID checker stands by until a next transmission parameter is inputted from the signaling decoder **2013**.

If the two parade IDs are identical to each other, the parade ID checker **3101** outputs the transmission parameter information to the blocks within the operation controller **2000** and the overall system.

If it is checked that the parade ID in the transmission parameter information inputted to the parade ID checker **3101** is identical to the parade ID selected by a user, the parade ID checker **3101** outputs starting_group_number (SGN) and number_of_groups (NOG) to the parade mapper **3103**, outputs sub_frame number, slot_number and parade_repetition_cycle PRC) to the frame synchronizer **3102**, outputs SCCC_block_mode, SCCC_outer_code_mode_A, SCCC_outer_code_mode_B, SCCC_outer_code_mode_C and SCCC_outer_code_mode_D to the block decoder **2005**, and outputs RS_frame_mode, RS_code_mode_primary and Rs_code_mode_secondary to the RS frame decoder **2006**.

The parade mapper **3103** receives the SGN and the NOG from the parade ID checker as inputs, decides a data group is carried by which one of sixteen slots within a Sub-frame, and then outputs the corresponding information. Data group number transmitted every sub-frame is set to an integer consecutive between SGN and (SGN+NOG−1). For instance, if SGN=3 and NOG=4, four groups, of which group numbers are 3, 4, 5 and 6, are transmitted for the corresponding sub-frames, respectively. The parade mapper finds a slot number j for transmitting a data group according to Equation 1 with a group number i obtained from SGN and NOG.

In the above example, in case of SGN=3 and NOG=4, if they are inserted in Equation 1, slot numbers of groups transmitted according to the above formula sequentially become 12, 2, 6 and 10.

The parade mapper **3103** then outputs the found slot number information. The slot number information may be outputted to the signaling decoder **2013**. In this case, the signaling decoder **2013** may identify a start of a subframe or a end of the subframe by using the slot number information.

For example of outputting slot numbers, a method of using a bit vector having 16 bits is available.

A bit vector SNi (i=0~15) can be set to 1 if there exists a group transmitted for an slot. A bit vector SNi (i=0~15) can be set to 0 if a group transmitted for an i^{th} slot does not exist. And, this bit vector can be outputted as slot number information.

The frame synchronizer **3102** receives the sub_frame_number, slot-number and PRC from the parade ID checker and then sends slot_counter and frame_mask signals as outputs. The slot_counter is the signal indicating a slot_

number at a current timing point at which a receiver is operating. And, the frame_mask is the signal indicating whether a corresponding parade is transmitted for a current frame. The frame synchronizer **3102** performs a process for initializing slot_counter, sub_frame_number and frame_counter in receiving signaling information initially. A counter value of a current timing point is generated from adding a delayed slot number L according to a time taken to decode signaling from demodulation together with the signaling information inputted in this process. After completion of the initialization process, slot_counter is updated every single slot period, updates sub_frame_counter every period of the slot_counter value, and updates frame_counter every period of the sub_frame_counter. By referring to the frame_counter information and the PRC information, a frame_mask signal is generated. For example, if a corresponding parade is being transmitted for a current frame, '1' is outputted as the frame_mask. Otherwise, it is able to output '0'.

The group controller **3104** receives the slot number information from the parade mapper **3103**. The group controller **3104** receives the slot_counter and frame_mask information from the frame synchronizer **3102**. The group controller **3104** then outputs group_presence_indicator indicating whether an M/H group is being transmitted. For instance, if the slot number information inputted from the parade mapper **3103** corresponds to 12, 2, 6 and 10, when the frame_mask information inputted from the frame synchronizer **3102** is 1 and the slot_counter inputted from the frame synchronizer **3102** includes 2, 6, 10 and 12, '1' is outputted as the group_presence_indicator. Otherwise, it is able to output 0.

The group_presence_indicator may be outputted to the signaling decoder **2013**. In this case, the signaling decoder **2013** may use the group_presence_indicator to identify whether a data group exits.

The known sequence indication controller **3105** outputs position information of another known data, group start position information and the like with position information of specific inputted known data. In this case, since the known data are present at a previously appointed position within the data group, if position data of one of a plurality of known data sequences, it is able to know data position information of another known sequence, data group start position information and the like. The known sequence indication controller **3105** can output known data and data group position information necessary for the demodulating unit of the receiving system using the group_presence_indicator information only if the data group is transmitted. Alternatively, the known sequence detector **2004** can perform operations of the known sequence indication controller **3105**.

Channel Equalizer

The data demodulated by the demodulator **2002** by using the known data are inputted to the equalizer **2003**. Additionally, the demodulated data may also be inputted to the known sequence detector **2004**. At this point, a data group that is inputted for the equalization process may be divided into region A to region D, as shown in FIG. **5**. More specifically, according to the embodiment of the present invention, region A includes M/H block B**4** to M/H block B**7**, region B includes M/H block B**3** and M/H block B**8**, region C includes M/H block B**2** and M/H block B**9**, and region D includes M/H block B**1** and M/H block B**10**. In other words, one data group is divided into M/H blocks from B**1** to B**10**, each M/H block having the length of 16 segments. Also, a long training sequence (i.e., known data sequence) is inserted at the starting portion of the M/H blocks B**4** to B**8**. Furthermore, two data groups may be allocated (or assigned) to one VSB field. In this case, field synchronization data are positioned in the 37^{th} segment of one of the two data groups.

The present invention may use known data, which have position and content information based upon an agreement between the transmitting system and the receiving system, and/or field synchronization data for the channel equalization process.

The channel equalizer **2003** may perform channel equalization using a plurality of methods. According to the present invention, the channel equalizer **2003** uses known data and/or field synchronization data, so as to estimate a channel impulse response (CIR), thereby performing channel equalization.

Most particularly, an example of estimating the CIR in accordance with each region within the data group, which is hierarchically divided and transmitted from the transmitting system, and applying each CIR differently will also be described herein.

At this point, a data group can be assigned and transmitted a maximum the number of 4 in a VSB frame in the transmitting system. In this case, all data group do not include field synchronization data. In the present invention, the data group including the field synchronization data performs channel-equalization using the field synchronization data and known data. And the data group not including the field synchronization data performs channel-equalization using the known data.

For example, the data of the M/H block B**3** including the field synchronization data performs channel-equalization using the CIR calculated from the field synchronization data area and the CIR calculated from the first known data area. Also, the data of the M/H blocks B**1** and B**2** performs channel-equalization using the CIR calculated from the field synchronization data area and the CIR calculated from the first known data area. Meanwhile, the data of the M/H blocks B**1** to B**3** not including the field synchronization data performs channel-equalization using CIRS calculated from the first known data area and the third known data area.

As described above, the present invention uses the CIR estimated from the known data region in order to perform channel equalization on data within the data group. At this point, each of the estimated CIRs may be directly used in accordance with the characteristics of each region within the data group. Alternatively, a plurality of the estimated CIRs may also be either interpolated or extrapolated so as to create a new CIR, which is then used for the channel equalization process.

Herein, when a value F(Q) of a function F(x) at a particular point Q and a value F(S) of the function F(x) at another particular point S are known, interpolation refers to estimating a function value of a point within the section between points Q and S. Linear interpolation corresponds to the simplest form among a wide range of interpolation operations.

FIG. **50** illustrates an example of linear interpolation. More specifically, in a random function F(x), when given the values F(Q) and F(S) each from points x=Q and x=S, respectively, the approximate value $\hat{F}(P)$ of the F(x) function at point x=P may be estimated by using Equation 7 below. In other words, since the values of F(Q) and F(S) respective to each point x=Q and x=S are known (or given), a straight line passing through the two points may be calculated so as to obtain the approximate value $\hat{F}(P)$ of the corresponding function value at point P. At this point, the straight line

passing through points (Q,F(Q)) and (S,F(S)) may be obtained by using Equation 7 below.

$$\hat{F}(x) = \frac{F(S) - F(Q)}{S - Q}(x - Q) + F(Q) \quad \text{Equation 7}$$

Accordingly, Equation 8 below shows the process of substituting p for x in Equation 7, so as to calculate the approximate value $\hat{F}(P)$ of the function value at point P.

$$\hat{F}(P) = \frac{F(S) - F(Q)}{S - Q}(P - Q) + F(Q) \quad \text{Equation 8}$$

$$\hat{F}(P) = \frac{S - P}{S - Q}F(Q) + \frac{P - S}{S - Q}F(S)$$

The linear interpolation method of Equation 8 is merely the simplest example of many other linear interpolation methods. Therefore, since any other linear interpolation method may be used, the present invention will not be limited only to the examples given herein.

Alternatively, when a value F(Q) of a function F(x) at a particular point Q and a value F(S) of the function F(x) at another particular point S are known (or given), extrapolation refers to estimating a function value of a point outside of the section between points Q and S. Herein, the simplest form of extrapolation corresponds to linear extrapolation.

FIG. **51** illustrates an example of linear extrapolation. As described above, for linear extrapolation as well as linear interpolation, in a random function F(x), when given the values F(Q) and F(S) each from points x=Q and x=S, respectively, the approximate value $\hat{F}(P)$ of the corresponding function value at point P may be obtained by calculating a straight line passing through the two points. Herein, linear extrapolation is the simplest form among a wide range of extrapolation operations. Similarly, the linear extrapolation described herein is merely exemplary among a wide range of possible extrapolation methods. And, therefore, the present invention is not limited only to the examples set forth herein.

FIG. **52** illustrates a block diagram of a channel equalizer according to an embodiment of the present invention. Referring to FIG. **52**, the channel equalizer includes a first frequency domain converter **4100**, a channel estimator **4110**, a second frequency domain converter **4121**, a coefficient calculator **4122**, a distortion compensator **4130**, and a time domain converter **4140**. Herein, the channel equalizer may further include a remaining carrier phase error remover, a noise canceller (NC), and a decision unit.

The first frequency domain converter **4100** includes an overlap unit **4101** overlapping inputted data, and a fast fourier transform (FFT) unit **4102** converting the data outputted from the overlap unit **4101** to frequency domain data.

The channel estimator **4110** includes a CIR estimator **4111**, a first cleaner **4113**, a CIR calculator **4114**, a second cleaner, and a zero-padding unit. herein, the channel estimator **4110** may further include a phase compensator compensating a phase of the CIR which estimated in the CIR estimator **4111**.

The second frequency domain converter **4121** includes a fast fourier transform (FFT) unit converting the CIR being outputted from the channel estimator **4110** to frequency domain CIR.

The time domain converter **4140** includes an IFFT unit **4141** converting the data having the distortion compensated by the distortion compensator **4130** to time domain data, and a save unit **4142** extracting only valid data from the data outputted from the IFFT unit **4141**. The output data from the save unit **4142** corresponds to the channel-equalized data.

If the remaining carrier phase error remover is connected to an output terminal of the time domain converter **4140**, the remaining carrier phase error remover estimates the remaining carrier phase error included in the channel-equalized data, thereby removing the estimated error. If the noise remover is connected to an output terminal of the time domain converter **4140**, the noise remover estimates noise included in the channel-equalized data, thereby removing the estimated noise.

More specifically, the receiving data demodulated in FIG. **52** are overlapped by the overlap unit **4101** of the first frequency domain converter **4100** at a pre-determined overlapping ratio, which are then outputted to the FFT unit **4102**. The FFT unit **4102** converts the overlapped time domain data to overlapped frequency domain data through by processing the data with FFT. Then, the converted data are outputted to the distortion compensator **4130**.

The distortion compensator **4130** performs a complex number multiplication on the overlapped frequency domain data outputted from the FFT unit **4102** included in the first frequency domain converter **4100** and the equalization coefficient calculated from the coefficient calculator **4122**, thereby compensating the channel distortion of the overlapped data outputted from the FFT unit **4102**. Thereafter, the compensated data are outputted to the IFFT unit **4141** of the time domain converter **4140**. The IFFT unit **4141** performs IFFT on the overlapped data having the channel distortion compensated, thereby converting the overlapped data to time domain data, which are then outputted to the save unit **4142**. The save unit **4142** extracts valid data from the data of the channel-equalized and overlapped in the time domain, and outputs the extracted valid data.

Meanwhile, the received data are inputted to the overlap unit **4101** of the first frequency domain converter **4100** included in the channel equalizer and, at the same time, inputted to the CIR estimator **4111** of the channel estimator **4110**.

The CIR estimator **4111** uses a training sequence, for example, data being inputted during the known data section and the known data in order to estimate the CIR. If the data to be channel-equalizing is the data within the data group including field synchronization data, the training sequence using in the CIR estimator **4111** may become the field synchronization data and known data. Meanwhile, if the data to be channel-equalizing is the data within the data group not including field synchronization data, the training sequence using in the CIR estimator **4111** may become only the known data.

For example, the CIR estimator **4111** estimates CIR using the known data correspond to reference known data generated during the known data section by the receiving system in accordance with an agreement between the receiving system and the transmitting system. For this, the CIR estimator **4111** is provided known data position information from the known sequence detector **2004**. Also the CIR estimator **4111** may be provided field synchronization position information from the known sequence detector **2004**.

The estimated CIR passes through the first cleaner (or pre-CIR cleaner) **4113** or bypasses the first cleaner **4113**, thereby being inputted to the CIR calculator (or CIR interpolator-extrapolator) **4114**. The CIR calculator **4114** either interpolates or extrapolates an estimated CIR, which is then outputted to the second cleaner (or post-CIR cleaner) **4115**.

The first cleaner **4113** may or may not operate depending upon whether the CIR calculator **4114** interpolates or extrapolates the estimated CIR. For example, if the CIR calculator **4114** interpolates the estimated CIR, the first cleaner **4113** does not operate. Conversely, if the CIR calculator **4114** extrapolates the estimated CIR, the first cleaner **4113** operates.

More specifically, the CIR estimated from the known data includes a channel element that is to be obtained as well as a jitter element caused by noise. Since such jitter element deteriorates the performance of the equalizer, it preferable that a coefficient calculator **4122** removes the jitter element before using the estimated CIR. Therefore, according to the embodiment of the present invention, each of the first and second cleaners **4113** and **4115** removes a portion of the estimated CIR having a power level lower than the predetermined threshold value (i.e., so that the estimated CIR becomes equal to '0'). Herein, this removal process will be referred to as a "CIR cleaning" process.

The CIR calculator **4114** performs CIR interpolation by multiplying CIRs estimated from the CIR estimator **4111** by each of coefficients, thereby adding the multiplied values. At this point, some of the noise elements of the CIR may be added to one another, thereby being cancelled. Therefore, when the CIR calculator **4114** performs CIR interpolation, the original (or initial) CIR having noise elements remaining therein. In other words, when the CIR calculator **4114** performs CIR interpolation, the estimated CIR bypasses the first cleaner **4113** and is inputted to the CIR calculator **4114**. Subsequently, the second cleaner **4115** cleans the CIR interpolated by the CIR interpolator-extrapolator **4114**.

Conversely, the CIR calculator **4114** performs CIR extrapolation by using a difference value between two CIRs, so as to estimate a CIR positioned outside of the two CIRs. Therefore, in this case, the noise element is rather amplified. Accordingly, when the CIR calculator **4114** performs CIR extrapolation, the CIR cleaned by the first cleaner **4113** is used. More specifically, when the CIR calculator **4114** performs CIR extrapolation, the extrapolated CIR passes through the second cleaner **4115**, thereby being inputted to the zero-padding unit **4116**.

Meanwhile, when a second frequency domain converter (or fast fourier transform (FFT**2**)) **4121** converts the CIR, which has been cleaned and outputted from the second cleaner **4115**, to a frequency domain, the length and of the inputted CIR and the FFT size may not match (or be identical to one another). In other words, the CIR length may be smaller than the FFT size. In this case, the zero-padding unit **4116** adds a number of zeros '0's corresponding to the difference between the FFT size and the CIR length to the inputted CIR, thereby outputting the processed CIR to the second frequency domain converter (FFT**2**) **4121**. Herein, the zero-padded CIR may correspond to one of the interpolated CIR, extrapolated CIR, and the CIR estimated in the known data section.

The second frequency domain converter **4121** performs FFT on the CIR being outputted from the zero padding unit **4116**, thereby converting the CIR to a frequency domain CIR. Then, the second frequency domain converter **4121** outputs the converted CIR to the coefficient calculator **4122**.

The coefficient calculator **4122** uses the frequency domain CIR being outputted from the second frequency domain converter **4121** to calculate the equalization coefficient. Then, the coefficient calculator **4122** outputs the calculated coefficient to the distortion compensator **4130**. Herein, for example, the coefficient calculator **4122** calculates a channel equalization coefficient of the frequency domain that can provide minimum mean square error (MMSE) from the CIR of the frequency domain, which is outputted to the distortion compensator **4130**.

The distortion compensator **4130** performs a complex number multiplication on the overlapped data of the frequency domain being outputted from the FFT unit **4102** of the first frequency domain converter **4100** and the equalization coefficient calculated by the coefficient calculator **4122**, thereby compensating the channel distortion of the overlapped data being outputted from the FFT unit **4102**.

Block Decoder

Meanwhile, if the data being inputted to the block decoder **2005**, after being channel-equalized by the equalizer **2003**, correspond to the data having both block encoding and trellis encoding performed thereon (i.e., the data within the RS frame, the signaling information data, etc.) by the transmitting system, trellis decoding and block decoding processes are performed on the inputted data as inverse processes of the transmitting system. Alternatively, if the data being inputted to the block decoder **2005** correspond to the data having only trellis encoding performed thereon (i.e., the main service data), and not the block encoding, only the trellis decoding process is performed on the inputted data as the inverse process of the transmitting system.

The trellis decoded and block decoded data by the block decoder **2005** are then outputted to the RS frame decoder **2006**. More specifically, the block decoder **2005** removes the known data, data used for trellis initialization, and signaling information data, MPEG header, which have been inserted in the data group, and the RS parity data, which have been added by the RS encoder/non-systematic RS encoder or non-systematic RS encoder of the transmitting system. Then, the block decoder **2005** outputs the processed data to the RS frame decoder **2006**. Herein, the removal of the data may be performed before the block decoding process, or may be performed during or after the block decoding process.

Meanwhile, the data trellis-decoded by the block decoder **2005** are outputted to the data deinterleaver of the main service data processor **2008**. At this point, the data being trellis-decoded by the block decoder **2005** and outputted to the data deinterleaver may not only include the main service data but may also include the data within the RS frame and the signaling information. Furthermore, the RS parity data that are added by the transmitting system after the preprocessor **230** may also be included in the data being outputted to the data deinterleaver.

According to another embodiment of the present invention, data that are not processed with block decoding and only processed with trellis encoding by the transmitting system may directly bypass the block decoder **2005** so as to be outputted to the data deinterleaver. In this case, a trellis decoder should be provided before the data deinterleaver. More specifically, if the inputted data correspond to the data having only trellis encoding performed thereon and not block encoding, the block decoder **2005** performs Viterbi (or trellis) decoding on the inputted data so as to output a hard decision value or to perform a hard-decision on a soft decision value, thereby outputting the result.

Meanwhile, if the inputted data correspond to the data having both block encoding process and trellis encoding process performed thereon, the block decoder **2005** outputs a soft decision value with respect to the inputted data.

In other words, if the inputted data correspond to data being processed with block encoding by the block processor **302** and being processed with trellis encoding by the trellis encoding module **256**, in the transmitting system, the block

decoder **2005** performs a decoding process and a trellis decoding process on the inputted data as inverse processes of the transmitting system. At this point, the RS frame encoder of the pre-processor included in the transmitting system may be viewed as an outer (or external) encoder. And, the trellis encoder may be viewed as an inner (or internal) encoder. When decoding such concatenated codes, in order to allow the block decoder **2005** to maximize its performance of decoding externally encoded data, the decoder of the internal code should output a soft decision value.

FIG. **53** illustrates a detailed block diagram of the block decoder **2005** according to an embodiment of the present invention. Referring to FIG. **53**, the block decoder **2005** includes a feedback controller **4010**, an input buffer **4011**, a trellis decoding unit (or 12-way trellis coded modulation (TCM) decoder or inner decoder) **4012**, a symbol-byte converter **4013**, an outer block extractor **4014**, a feedback deformatter **4015**, a symbol deinterleaver **4016**, an outer symbol mapper **4017**, a symbol decoder **4018**, an inner symbol mapper **4019**, a symbol interleaver **4020**, a feedback formatter **4021**, and an output buffer **4022**. Herein, just as in the transmitting system, the trellis decoding unit **4012** may be viewed as an inner (or internal) decoder. And, the symbol decoder **4018** may be viewed as an outer (or external) decoder.

The input buffer **4011** temporarily stores the mobile service data symbols being channel-equalized and outputted from the equalizer **2003**. (Herein, the mobile service data symbols may include symbols corresponding to the signaling information, RS parity data symbols and CRC data symbols added during the encoding process of the RS frame.) Thereafter, the input buffer **4011** repeatedly outputs the stored symbols for M number of times to the trellis decoding unit **4012** in a turbo block (TDL) size required for the turbo decoding process.

The turbo decoding length (TDL) may also be referred to as a turbo block. Herein, a TDL should include at least one SCCC block size. Therefore, as defined in FIG. **5**, when it is assumed that one M/H block is a 16-segment unit, and that a combination of 10 M/H blocks form one SCCC block, a TDL should be equal to or larger than the maximum possible combination size. For example, when it is assumed that 2 M/H blocks form one SCCC block, the TDL may be equal to or larger than 32 segments (i.e., 828×32=26496 symbols). Herein, M indicates a number of repetitions for turbo-decoding pre-decided by the feed-back controller **4010**.

Also, M represents a number of repetitions of the turbo decoding process, the number being predetermined by the feedback controller **4010**.

Additionally, among the values of symbols being channel-equalized and outputted from the equalizer **2003**, the input symbol values corresponding to a section having no mobile service data symbols (including RS parity data symbols during RS frame encoding and CRC data symbols) included therein, bypass the input buffer **4011** without being stored. More specifically, since trellis-encoding is performed on input symbol values of a section wherein SCCC block-encoding has not been performed, the input buffer **4011** inputs the inputted symbol values of the corresponding section directly to the trellis encoding module **4012** without performing any storage, repetition, and output processes. The storage, repetition, and output processes of the input buffer **4011** are controlled by the feedback controller **4010**. Herein, the feedback controller **4010** refers to SCCC-associated information (e.g., SCCC block mode and SCCC outer code mode), which are outputted from the signaling decoder **2013** or the operation controller **2000**, in order to control the storage and output processes of the input buffer **4011**.

The trellis decoding unit **4012** includes a 12-way TCM decoder. Herein, the trellis decoding unit **4012** performs 12-way trellis decoding as inverse processes of the 12-way trellis encoder.

More specifically, the trellis decoding unit **4012** receives a number of output symbols of the input buffer **4011** and soft-decision values of the feedback formatter **4021** equivalent to each TDL, so as to perform the TCM decoding process.

At this point, based upon the control of the feedback controller **4010**, the soft-decision values outputted from the feedback formatter **4021** are matched with a number of mobile service data symbol places so as to be in a one-to-one (1:1) correspondence. Herein, the number of mobile service data symbol places is equivalent to the TDL being outputted from the input buffer **4011**.

More specifically, the mobile service data being outputted from the input buffer **4011** are matched with the turbo decoded data being inputted, so that each respective data place can correspond with one another. Thereafter, the matched data are outputted to the trellis decoding unit **4012**. For example, if the turbo decoded data correspond to the third symbol within the turbo block, the corresponding symbol (or data) is matched with the third symbol included in the turbo block, which is outputted from the input buffer **4011**. Subsequently, the matched symbol (or data) is outputted to the trellis decoding unit **4012**.

In order to do so, while the regressive turbo decoding is in process, the feedback controller **4010** controls the input buffer **4011** so that the input buffer **4011** stores the corresponding turbo block data. Also, by delaying data (or symbols), the soft decision value (e.g., LLR) of the symbol outputted from the symbol interleaver **4020** and the symbol of the input buffer **4011** corresponding to the same place (or position) within the block of the output symbol are matched with one another to be in a one-to-one correspondence. Thereafter, the matched symbols are controlled so that they can be inputted to the TCM decoder through the respective path. This process is repeated for a predetermined number of turbo decoding cycle periods. Then, the data of the next turbo block are outputted from the input buffer **4011**, thereby repeating the turbo decoding process.

The output of the trellis decoding unit **4012** signifies a degree of reliability of the transmission bits configuring each symbol. For example, in the transmitting system, since the input data of the trellis encoding module correspond to two bits as one symbol, a log likelihood ratio (LLR) between the likelihood of a bit having the value of '1' and the likelihood of the bit having the value of '0' may be respectively outputted (in bit units) to the upper bit and the lower bit. Herein, the log likelihood ratio corresponds to a log value for the ratio between the likelihood of a bit having the value of '1' and the likelihood of the bit having the value of '0'. Alternatively, a LLR for the likelihood of 2 bits (i.e., one symbol) being equal to "00", "01", "10", and "11" may be respectively outputted (in symbol units) to all 4 combinations of bits (i.e., 00, 01, 10, 11). Consequently, this becomes the soft decision value that indicates the degree of reliability of the transmission bits configuring each symbol. A maximum a posteriori probability (MAP) or a soft-out Viterbi algorithm (SOYA) may be used as a decoding algorithm of each TCM decoder within the trellis decoding unit **4012**.

The output of the trellis decoding unit **4012** is inputted to the symbol-byte converter **4013** and the outer block extractor **4014**.

The symbol-byte converter **4013** performs a hard-decision process of the soft decision value that is trellis decoded and outputted from the trellis decoding unit **4012**. Thereafter, the symbol-byte converter **4013** groups 4 symbols into byte units, which are then outputted to the data deinterleaver of the main service data processor **2008** of FIG. **44**. More specifically, the symbol-byte converter **4013** performs hard-decision in bit units on the soft decision value of the symbol outputted from the trellis decoding unit **4012**. Therefore, the data processed with hard-decision and outputted in bit units from the symbol-byte converter **4013** not only include main service data, but may also include mobile service data, known data, RS parity data, and MPEG headers.

Among the soft decision values of TDL size of the trellis decoding unit **4012**, the outer block extractor **4014** identifies the soft decision values of B size of corresponding to the mobile service data symbols (wherein symbols corresponding to signaling information, RS parity data symbols that are added during the encoding of the RS frame, and CRC data symbols are included) and outputs the identified soft decision values to the feedback deformatter **4015**.

The feedback deformatter **4015** changes the processing order of the soft decision values corresponding to the mobile service data symbols. This is an inverse process of an initial change in the processing order of the mobile service data symbols, which are generated during an intermediate step, wherein the output symbols outputted from the block processor **302** of the transmitting system are being inputted to the trellis encoding module **256** (e.g., when the symbols pass through the group formatter, the data deinterleaver, the packet formatter, and the data interleaver). Thereafter, the feedback deformatter **2015** performs reordering of the process order of soft decision values corresponding to the mobile service data symbols and, then, outputs the processed mobile service data symbols to the symbol deinterleaver **4016**.

This is because a plurality of blocks exist between the block processor **302** and the trellis encoding module **256**, and because, due to these blocks, the order of the mobile service data symbols being outputted from the block processor **302** and the order of the mobile service data symbols being inputted to the trellis encoding module **256** are not identical to one another. Therefore, the feedback deformatter **4015** reorders (or rearranges) the order of the mobile service data symbols being outputted from the outer block extractor **4014**, so that the order of the mobile service data symbols being inputted to the symbol deinterleaver **4016** matches the order of the mobile service data symbols outputted from the block processor **302** of the transmitting system. The reordering process may be embodied as one of software, middleware, and hardware.

The symbol deinterleaver **4016** performs deinterleaving on the mobile service data symbols having their processing orders changed and outputted from the feedback deformatter **4015**, as an inverse process of the symbol interleaving process of the symbol interleaver **514** included in the transmitting system. The size of the block used by the symbol deinterleaver **4016** during the deinterleaving process is identical to interleaving size of an actual symbol (i.e., B) of the symbol interleaver **514**, which is included in the transmitting system. This is because the turbo decoding process is performed between the trellis decoding unit **4012** and the symbol decoder **4018**. Both the input and output of the symbol deinterleaver **4016** correspond to soft decision values, and the deinterleaved soft decision values are outputted to the outer symbol mapper **4017**.

The operations of the outer symbol mapper **4017** may vary depending upon the structure and coding rate of the convolution encoder **513** included in the transmitting system. For example, when data are 1/2-rate encoded by the convolution encoder **513** and then transmitted, the outer symbol mapper **4017** directly outputs the input data without modification. In another example, when data are 1/4-rate encoded by the convolution encoder **513** and then transmitted, the outer symbol mapper **4017** converts the input data so that it can match the input data format of the symbol decoder **4018**. For this, the outer symbol mapper **4017** may be inputted SCCC-associated information (i.e., SCCC block mode and SCCC outer code mode) from the signaling decoder **2013**. Then, the outer symbol mapper **4017** outputs the converted data to the symbol decoder **4018**.

The symbol decoder **4018** (i.e., the outer decoder) receives the data outputted from the outer symbol mapper **4017** and performs symbol decoding as an inverse process of the convolution encoder **513** included in the transmitting system. At this point, two different soft decision values are outputted from the symbol decoder **4018**. One of the outputted soft decision values corresponds to a soft decision value matching the output symbol of the convolution encoder **513** (hereinafter referred to as a "first decision value"). The other one of the outputted soft decision values corresponds to a soft decision value matching the input bit of the convolution encoder **513** (hereinafter referred to as a "second decision value").

More specifically, the first decision value represents a degree of reliability the output symbol (i.e., 2 bits) of the convolution encoder **513**. Herein, the first soft decision value may output (in bit units) a LLR between the likelihood of 1 bit being equal to '1' and the likelihood of 1 bit being equal to '0' with respect to each of the upper bit and lower bit, which configures a symbol. Alternatively, the first soft decision value may also output (in symbol units) a LLR for the likelihood of 2 bits being equal to "00", "01", "10", and "11" with respect to all possible combinations. The first soft decision value is fed-back to the trellis decoding unit **4012** through the inner symbol mapper **4019**, the symbol interleaver **4020**, and the feedback formatter **4021**. On the other hand, the second soft decision value indicates a degree of reliability the input bit of the convolution encoder **513** included in the transmitting system. Herein, the second soft decision value is represented as the LLR between the likelihood of 1 bit being equal to '1' and the likelihood of 1 bit being equal to '0'. Thereafter, the second soft decision value is outputted to the outer buffer **4022**. In this case, a maximum a posteriori probability (MAP) or a soft-out Viterbi algorithm (SOYA) may be used as the decoding algorithm of the symbol decoder **4018**.

The first soft decision value that is outputted from the symbol decoder **4018** is inputted to the inner symbol mapper **4019**. The inner symbol mapper **4019** converts the first soft decision value to a data format corresponding the input data of the trellis decoding unit **4012**. Thereafter, the inner symbol mapper **4019** outputs the converted soft decision value to the symbol interleaver **4020**. The operations of the inner symbol mapper **4019** may also vary depending upon the structure and coding rate of the convolution encoder **513** included in the transmitting system.

The symbol interleaver **4020** performs symbol interleaving, as shown in FIG. **30**, on the first soft decision value that is outputted from the inner symbol mapper **4019**. Then, the symbol interleaver **4020** outputs the symbol-interleaved first

soft decision value to the feedback formatter **4021**. Herein, the output of the symbol interleaver **4020** also corresponds to a soft decision value.

With respect to the changed processing order of the soft decision values corresponding to the symbols that are generated during an intermediate step, wherein the output symbols outputted from the block processor **302** of the transmitting system are being inputted to the trellis encoding module (e.g., when the symbols pass through the group formatter, the data deinterleaver, the packet formatter, the RS encoder, and the data interleaver), the feedback formatter **4021** alters (or changes) the order of the output values outputted from the symbol interleaver **4020**. Subsequently, the feedback formatter **4020** outputs values to the trellis decoding unit **4012** in the changed order. The reordering process of the feedback formatter **4021** may configure at least one of software, hardware, and middleware.

The soft decision values outputted from the symbol interleaver **4020** are matched with the positions of mobile service data symbols each having the size of TDL, which are outputted from the input buffer **4011**, so as to be in a one-to-one correspondence. Thereafter, the soft decision values matched with the respective symbol position are inputted to the trellis decoding unit **4012**. At this point, since the main service data symbols or the RS parity data symbols and known data symbols of the main service data do not correspond to the mobile service data symbols, the feedback formatter **4021** inserts null data in the corresponding positions, thereby outputting the processed data to the trellis decoding unit **4012**. Additionally, each time the symbols having the size of TDL are turbo decoded, no value is fed-back by the symbol interleaver **4020** starting from the beginning of the first decoding process. Therefore, the feedback formatter **4021** is controlled by the feedback controller **4010**, thereby inserting null data into all symbol positions including a mobile service data symbol. Then, the processed data are outputted to the trellis decoding unit **4012**.

The output buffer **4022** receives the second soft decision value from the symbol decoder **4018** based upon the control of the feedback controller **4010**. Then, the output buffer **4022** temporarily stores the received second soft decision value. Thereafter, the output buffer **4022** outputs the second soft decision value to the RS frame decoder **2006**. For example, the output buffer **4022** overwrites the second soft decision value of the symbol decoder **4018** until the turbo decoding process is performed for M number of times. Then, once all M number of turbo decoding processes is performed for a single TDL, the corresponding second soft decision value is outputted to the RS frame decoder **2006**.

The feedback controller **4010** controls the number of turbo decoding and turbo decoding repetition processes of the overall block decoder, shown in FIG. **53**. More specifically, once the turbo decoding process has been repeated for a predetermined number of times, the second soft decision value of the symbol decoder **4018** is outputted to the RS frame decoder **2006** through the output buffer **4022**. Thus, the block decoding process of a turbo block is completed. In the description of the present invention, this process is referred to as a regressive turbo decoding process for simplicity.

At this point, the number of regressive turbo decoding rounds between the trellis decoding unit **4012** and the symbol decoder **4018** may be defined while taking into account hardware complexity and error correction performance. Accordingly, if the number of rounds increases, the error correction performance may be enhanced. However, this may lead to a disadvantageous of the hardware becoming more complicated (or complex).

Meanwhile, the main service data processor **2008** corresponds to block required for receiving the main service data. Therefore, the above-mentioned blocks may not be necessary (or required) in the structure of a digital broadcast receiving system for receiving mobile service data only.

The data deinterleaver of the main service data processor **2008** performs an inverse process of the data interleaver included in the transmitting system. In other words, the data deinterleaver deinterleaves the main service data outputted from the block decoder **2005** and outputs the deinterleaved main service data to the RS decoder. The data being inputted to the data deinterleaver include main service data, as well as mobile service data, known data, RS parity data, and an MPEG header. At this point, among the inputted data, only the main service data and the RS parity data added to the main service data packet may be outputted to the RS decoder. Also, all data outputted after the data derandomizer may all be removed with the exception for the main service data. In the embodiment of the present invention, only the main service data and the RS parity data added to the main service data packet are inputted to the RS decoder.

The RS decoder performs a systematic RS decoding process on the deinterleaved data and outputs the processed data to the data derandomizer.

The data derandomizer receives the output of the RS decoder and generates a pseudo random data byte identical to that of the randomizer included in the digital broadcast transmitting system. Thereafter, the data derandomizer performs a bitwise exclusive OR (XOR) operation on the generated pseudo random data byte, thereby inserting the MPEG synchronization bytes to the beginning of each packet so as to output the data in **188**-byte main service data packet units.

RS Frame Decoder

The data outputted from the block decoder **2005** are in portion units. More specifically, in the transmitting system, the RS frame is divided into several portions, and the mobile service data of each portion are assigned either to regions A/B/C/D within the data group or to any one of regions A/B and regions C/D , thereby being transmitted to the receiving system. Therefore, the RS frame decoder **2006** groups several portions included in a parade so as to form an RS frame. Alternatively, the RS frame decoder **2006** may also group several portions included in a parade so as to form two RS frames. Thereafter, error correction decoding is performed in

RS frame units.

For example, when the RS frame mode value is equal to '00', then one parade transmits one RS frame. At this point, one RS frame is divided into several portions, and the mobile service data of each portion are assigned to regions A/B/C/D of the corresponding data group, thereby being transmitted. In this case, the RS frame decoder **2006** extracts mobile service data from regions A/B/C/D of the corresponding data group, as shown in FIG. **54**(a). Subsequently, the RS frame decoder **2006** may perform the process of forming (or creating) a portion on a plurality of data group within a parade, thereby forming several portions. Then, the several portions of mobile service data may be grouped to form an RS frame. Herein, if stuffing bytes are added to the last portion, the RS frame may be formed after removing the stuffing byte.

In another example, when the RS frame mode value is equal to '01', then one parade transmits two RS frames (i.e., a primary RS frame and a secondary RS frame). At this

point, a primary RS frame is divided into several primary portions, and the mobile service data of each primary portion are assigned to regions A/B of the corresponding data group, thereby being transmitted. Also, a secondary RS frame is divided into several secondary portions, and the mobile service data of each secondary portion are assigned to regions C/D of the corresponding data group, thereby being transmitted.

In this case, the RS frame decoder **2006** extracts mobile service data from regions A/B of the corresponding data group, as shown in FIG. **54**(b). Subsequently, the RS frame decoder **2006** may perform the process of forming (or creating) a primary portion on a plurality of data group within a parade, thereby forming several primary portions. Then, the several primary portions of mobile service data may be grouped to form a primary RS frame. Herein, if stuffing bytes are added to the last primary portion, the primary RS frame may be formed after removing the stuffing byte. Also, the RS frame decoder **2006** extracts mobile service data from regions C/D of the corresponding data group. Subsequently, the RS frame decoder **2006** may perform the process of forming (or creating) a secondary portion on a plurality of data group within a parade, thereby forming several secondary portions. Then, the several secondary portions of mobile service data may be grouped to form a secondary RS frame. Herein, if stuffing bytes are added to the last secondary portion, the secondary RS frame may be formed after removing the stuffing byte.

More specifically, the RS frame decoder **2006** receives the RS-encoded and/or CRC-encoded mobile service data of each portion from the block decoder **2005**. Then, the RS frame decoder **2006** groups several portions, which are inputted based upon RS frame-associated information outputted from the signaling decoder **2013** or the operation controller **2000**, thereby performing error correction. By referring to the RS frame mode value included in the RS frame-associated information, the RS frame decoder **2006** may form an RS frame and may also be informed of the number of RS code parity data bytes and the code size. Herein, the RS code is used to configure (or form) the RS frame. The RS frame decoder **2006** also refers to the RS frame-associated information in order to perform an inverse process of the RS frame encoder, which is included in the transmitting system, thereby correcting the errors within the RS frame. Thereafter, the RS frame decoder **2006** adds 1 MPEG synchronization data byte to the error-correction mobile service data packet. In an earlier process, the 1 MPEG synchronization data byte was removed from the mobile service data packet during the RS frame encoding process. Finally, the RS frame decoder **2006** performs a derandomizing process on the processed mobile service data packet.

FIG. **55** illustrates, when the RS frame mode value is equal to '00', an exemplary process of grouping several portion being transmitted to a parade, thereby forming an RS frame and an RS frame reliability map.

More specifically, the RS frame decoder **2006** receives and groups a plurality of mobile service data bytes, so as to form an RS frame. According to the present invention, in transmitting system, the mobile service data correspond to data RS-encoded in RS frame units. At this point, the mobile service data may already be error correction encoded (e.g., CRC-encoded). Alternatively, the error correction encoding process may be omitted.

It is assumed that, in the transmitting system, an RS frame having the size of (N+2)×(187+P) bytes is divided into M number of portions, and that the M number of mobile service data portions are assigned and transmitted to regions A/B/C/D in M number of data groups, respectively. In this case, in the receiving system, each mobile service data portion is grouped, as shown in FIG. **55**(a), thereby forming an RS frame having the size of (N+2)×(187+P) bytes. At this point, when stuffing bytes (S) are added to at least one portion included in the corresponding RS frame and then transmitted, the stuffing bytes are removed, thereby configuring an RS frame and an RS frame reliability map. For example, as shown in FIG. **27**, when S number of stuffing bytes are added to the corresponding portion, the S number of stuffing bytes are removed, thereby configuring the RS frame and the RS frame reliability map.

Herein, when it is assumed that the block decoder **2005** outputs a soft decision value for the decoding result, the RS frame decoder **2006** may decide the '0' and '1' of the corresponding bit by using the codes of the soft decision value. 8 bits that are each decided as described above are grouped to create 1 data byte. If the above-described process is performed on all soft decision values of several portions (or data groups) included in a parade, the RS frame having the size of (N+2)×(187+P) bytes may be configured.

Additionally, the present invention uses the soft decision value not only to configure the RS frame but also to configure a reliability map.

Herein, the reliability map indicates the reliability of the corresponding data byte, which is configured by grouping 8 bits, the 8 bits being decided by the codes of the soft decision value.

For example, when the absolute value of the soft decision value exceeds a pre-determined threshold value, the value of the corresponding bit, which is decided by the code of the corresponding soft decision value, is determined to be reliable. Conversely, when the absolute value of the soft decision value does not exceed the pre-determined threshold value, the value of the corresponding bit is determined to be unreliable. Thereafter, if even a single bit among the 8 bits, which are decided by the codes of the soft decision value and group to configure one data byte, is determined to be unreliable, the corresponding data byte is marked on the reliability map as an unreliable data byte.

Herein, determining the reliability of one data byte is only exemplary. More specifically, when a plurality of data bytes (e.g., at least 4 data bytes) are determined to be unreliable, the corresponding data bytes may also be marked as unreliable data bytes within the reliability map. Conversely, when all of the data bits within the one data byte are determined to be reliable (i.e., when the absolute value of the soft decision values of all 8 bits included in the one data byte exceed the predetermined threshold value), the corresponding data byte is marked to be a reliable data byte on the reliability map. Similarly, when a plurality of data bytes (e.g., at least 4 data bytes) are determined to be reliable, the corresponding data bytes may also be marked as reliable data bytes within the reliability map. The numbers proposed in the above-described example are merely exemplary and, therefore, do not limit the scope or spirit of the present invention.

The process of configuring the RS frame and the process of configuring the reliability map both using the soft decision value may be performed at the same time. Herein, the reliability information within the reliability map is in a one-to-one correspondence with each byte within the RS frame. For example, if a RS frame has the size of (N+2)×(187+P) bytes, the reliability map is also configured to have the size of (N+2)×(187+P) bytes. FIG. **55**(a ') and FIG. **55**(b

') respectively illustrate the process steps of configuring the reliability map according to the present invention.

Subsequently, the RS frame reliability map is used on the RS frames so as to perform error correction.

FIG. **56** illustrates example of the error correction processed according to embodiments of the present invention. FIG. **56** illustrates an example of performing an error correction process when the transmitting system has performed both RS encoding and CRC encoding processes on the RS frame.

As shown in FIG. **56**(a) and FIG. **56**(a '), when the RS frame having the size of (N+2)×(187+P) bytes and the RS frame reliability map having the size of (N+2)×(187+P) bytes are created, a CRC syndrome checking process is performed on the created RS frame, thereby verifying whether any error has occurred in each row. Subsequently, as shown in FIG. **56**(b), a 2-byte checksum is removed to configure an RS frame having the size of N×(187+P) bytes. Herein, the presence (or existence) of an error is indicated on an error flag corresponding to each row. Similarly, since the portion of the reliability map corresponding to the CRC checksum has hardly any applicability, this portion is removed so that only N×(187+P) number of the reliability information bytes remain, as shown in FIG. **56**(b ').

After performing the CRC syndrome checking process, as described above, a RS decoding process is performed in a column direction. Herein, a RS erasure correction process may be performed in accordance with the number of CRC error flags. More specifically, as shown in FIG. **56**(c), the CRC error flag corresponding to each row within the RS frame is verified. Thereafter, the RS frame decoder **2006** determines whether the number of rows having a CRC error occurring therein is equal to or smaller than the maximum number of errors on which the RS erasure correction may be performed, when performing the RS decoding process in a column direction. The maximum number of errors corresponds to P number of parity bytes inserted when performing the RS encoding process. In the embodiment of the present invention, it is assumed that 48 parity bytes have been added to each column (i.e., P=48).

If the number of rows having the CRC errors occurring therein is smaller than or equal to the maximum number of errors (i.e., 48 errors according to this embodiment) that can be corrected by the RS erasure decoding process, a (235, 187)-RS erasure decoding process is performed in a column direction on the RS frame having (187+P) number of N-byte rows (i.e., 235 N-byte rows), as shown in FIG. **56**(d). Thereafter, as shown in FIG. **56**(e), the 48-byte parity data that have been added at the end of each column are removed. Conversely, however, if the number of rows having the CRC errors occurring therein is greater than the maximum number of errors (i.e., 48 errors) that can be corrected by the RS erasure decoding process, the RS erasure decoding process cannot be performed. In this case, the error may be corrected by performing a general RS decoding process. In addition, the reliability map, which has been created based upon the soft decision value along with the RS frame, may be used to further enhance the error correction ability (or performance) of the present invention.

More specifically, the RS frame decoder **2006** compares the absolute value of the soft decision value of the block decoder **2005** with the pre-determined threshold value, so as to determine the reliability of the bit value decided by the code of the corresponding soft decision value. Also, bits, each being determined by the code of the soft decision value, are grouped to form one data byte. Accordingly, the reliability information on this one data byte is indicated on the reliability map. Therefore, as shown in FIG. **56**(c), even though a particular row is determined to have an error occurring therein based upon a CRC syndrome checking process on the particular row, the present invention does not assume that all bytes included in the row have errors occurring therein. The present invention refers to the reliability information of the reliability map and sets only the bytes that have been determined to be unreliable as erroneous bytes. In other words, with disregard to whether or not a CRC error exists within the corresponding row, only the bytes that are determined to be unreliable based upon the reliability map are set as erasure points.

According to another method, when it is determined that CRC errors are included in the corresponding row, based upon the result of the CRC syndrome checking result, only the bytes that are determined by the reliability map to be unreliable are set as errors. More specifically, only the bytes corresponding to the row that is determined to have errors included therein and being determined to be unreliable based upon the reliability information, are set as the erasure points. Thereafter, if the number of error points for each column is smaller than or equal to the maximum number of errors (i.e., 48 errors) that can be corrected by the RS erasure decoding process, an RS erasure decoding process is performed on the corresponding column. Conversely, if the number of error points for each column is greater than the maximum number of errors (i.e., 48 errors) that can be corrected by the RS erasure decoding process, a general decoding process is performed on the corresponding column.

More specifically, if the number of rows having CRC errors included therein is greater than the maximum number of errors (i.e., 48 errors) that can be corrected by the RS erasure decoding process, either an RS erasure decoding process or a general RS decoding process is performed on a column that is decided based upon the reliability information of the reliability map, in accordance with the number of erasure points within the corresponding column. For example, it is assumed that the number of rows having CRC errors included therein within the RS frame is greater than 48. And, it is also assumed that the number of erasure points decided based upon the reliability information of the reliability map is indicated as 40 erasure points in the first column and as erasure points in the second column. In this case, a (235,187)-RS erasure decoding process is performed on the first column. Alternatively, a (235,187)-RS decoding process is performed on the second column. When error correction decoding is performed on all column directions within the RS frame by using the above-described process, the 48-byte parity data which were added at the end of each column are removed, as shown in FIG. **56**(e).

As described above, even though the total number of CRC errors corresponding to each row within the RS frame is greater than the maximum number of errors that can be corrected by the RS erasure decoding process, when the number of bytes determined to have a low reliability level, based upon the reliability information on the reliability map within a particular column, while performing error correction decoding on the particular column. Herein, the difference between the general RS decoding process and the RS erasure decoding process is the number of errors that can be corrected. More specifically, when performing the general RS decoding process, the number of errors corresponding to half of the number of parity bytes (i.e., (number of parity bytes)/2) that are inserted during the RS encoding process may be error corrected (e.g., 24 errors may be corrected). Alternatively, when performing the RS erasure decoding process, the number of errors corresponding to the number

of parity bytes that are inserted during the RS encoding process may be error corrected (e.g., 48 errors may be corrected).

After performing the error correction decoding process, as described above, a RS frame payload configured of 187 N-byte rows (or packet) may be obtained as shown in FIG. **56**(e). At this point, the RS frame payload having the size of N×187 bytes is performed a derandomizing process, which corresponds to the inverse process of the randomizer included in the transmitting system and then the derandomized data are outputted, thereby obtaining the mobile service data transmitted from the transmitting system. In the present invention, the RS frame decoder **2006** may perform the data derandomizing function.

An RS frame decoder may be configured of M number of RS frame decoders provided in parallel, wherein the number of RS frame encoders is equal to the number of parades (=M) within an M/H frame, a multiplexer for multiplexing each portion and being provided to each input end of the M number of RS frame decoders, and a demultiplexer for demultiplexing each portion and being provided to each output end of the M number of RS frame decoders.

As described above, the transmitting system and the receiving system and the broadcast signal processing method of the same according to the present invention have the following advantages. When transmitting mobile service data through a channel, the present invention may be robust against errors and backward compatible with the conventional digital broadcast receiving system. Moreover, the present invention may also receive the mobile service data without any error even in channels having severe ghost effect and noise. Furthermore, by inserting known data in a particular position (or place) within a data region and transmitting the processed data, the receiving performance of the receiving system may be enhanced even in a channel environment that is liable to frequent changes.

Also, according to the present invention, as the TNoG information is acquired using slot information and data group information during block deinterleaving of the turbo-decoded FIC data, the FIC data can be decoded even though the TNoG information of the current subframe is not obtained through TPC data decoding. Accordingly, decoding performance of the FIC data can be improved.

Finally, the present invention is even more effective when applied to mobile and portable receivers, which are also liable to a frequent change in channel and which require protection (or resistance) against intense noise.

It will be apparent to those skilled in the art that various modifications and variations can be made in the present invention without departing from the spirit or scope of the inventions. Thus, it is intended that the present invention covers the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

What is claimed is:

[1. A method for processing a broadcast signal in a receiving system, the method comprising:

receiving the broadcast signal including a data group, the data group comprising mobile service data, known data sequences, and signaling data, the signaling data including fast information channel (FIC) data and transmission parameter channel (TPC) data, wherein the FIC data includes information for fast mobile service acquisition and the TPC data includes a number of a slot through which the data group is transmitted and version information of the FIC data;

channel equalizing the broadcast signal by using at least one of the known data sequences;

performing turbo decoding on the signaling data included in the channel equalized broadcast signal;

demultiplexing the TPC data and the FIC data from the turbo-decoded signaling data;

performing first error correction decoding on the demultiplexed TPC data;

performing block deinterleaving on the demultiplexed FIC data in a block unit of TNoG×51 bytes, wherein TNoG equals a number of data groups assigned to one subframe; and

performing second error correction decoding on the block deinterleaved FIC data,

wherein a time period during which the broadcast signal is received is a mobile/handheld (M/H) frame that includes five subframes,

wherein each of the five subframes includes 16 slots,

wherein N number of data groups are received through the corresponding subframe via N number of slots,

wherein N is greater than 0 and less than or equal to 16, and

wherein performing the block deinterleaving comprises:

detecting a start of a next subframe when an FIC decoding start command is received;

counting a number of data groups transmitted to the next subframe the start of which was detected;

detecting an end of the next subframe;

acquiring TNoG information from the counted data groups when the end of the next subframe is detected; and

performing the block deinterleaving using the acquired TNoG information.**]**

[2. The method of claim **1**, wherein the signaling data are assigned to a signaling information area located between a first known data sequence and a second known data sequence of the known data sequences.**]**

[3. The method of claim **1**, wherein performing block deinterleaving further comprises:

acquiring the TNoG information from the error correction decoded TPC data.**]**

[4. The method of claim **1**, wherein detecting the start of the next subframe comprises using a slot number included in the error correction decoded TPC data.**]**

[5. The method of claim **1**, wherein detecting the end of the next subframe comprises using a slot number included in the error correction decoded TPC data.**]**

[6. The method of claim **1**, wherein detecting the end of the next subframe comprises counting a number of field synchronizations transmitted to the subframe.**]**

[7. The method of claim **1**, wherein performing the first error correction decoding comprises performing a Reed-Solomon (RS) process with (18, 10) RS code on the demultiplexed TPC data.**]**

[8. The method of claim **1**, wherein performing the second error correction decoding comprises performing a Reed-Solomon (RS) decoding process with (51, 37) RS code on the block deinterleaved FIC data.**]**

[9. A receiving system comprising:

a tuner for receiving a broadcast signal including a data group, the data group comprising mobile service data, known data sequences, and signaling data, the signaling data including fast information channel (FIC) data and transmission parameter channel (TPC) data, wherein the FIC data includes information for fast mobile service acquisition and the TPC data includes a number

of a slot through which the data group is transmitted and version information of the FIC data;

a channel equalizer for channel equalizing the broadcast signal by using at least one of the known data sequences;

a turbo decoder for performing turbo decoding on the signaling data included in the channel equalized broadcast signal;

a demultiplexer for demultiplexing the TPC data and the FIC data from the turbo-decoded signaling data;

a first error correction decoder for performing first error correction decoding on the demultiplexed TPC data;

a block deinterleaver for performing block deinterleaving on the demultiplexed FIC data in a block unit of TNoG×51 bytes, wherein TNoG equals a number of all data groups assigned to one subframe; and

a second error correction decoder for performing second error correction decoding on the block deinterleaved FIC data,

wherein a time period during which the broadcast signal is received is a mobile/handheld (M/H) frame that includes five subframes,

wherein each of the five subframes includes 16 slots,

wherein N number data groups are received through the corresponding subframe via N number of slots,

wherein N is greater than 0 and less than or equal to 16, and

wherein the block deinterleaver includes a controller for detecting a start of a next subframe when an FIC decoding start command is received, counting a number of data groups transmitted to the next subframe the start of which was detected, detecting an end of the next subframe and acquiring TNoG information from the counted data groups when the end of the next subframe is detected.]

[10. The receiving system of claim 9, wherein the signaling data are assigned to a signaling information area located between a first known data sequence and a second known data sequence of the known data sequences.]

[11. The receiving system of claim 9, wherein the controller is further for acquiring the TNoG information from the error correction decoded TPC data.]

[12. The receiving system of claim 9, wherein the controller is further for detecting the start of the next subframe using a slot number included in the error correction decoded TPC data.]

[13. The receiving system of claim 9, wherein the controller is further for detecting the end of the next subframe using a slot number included in the error correction decoded TPC data.]

[14. The receiving system of claim 9, wherein the controller is further for detecting the end of the next subframe by counting the number of field synchronizations transmitted to the subframe.]

[15. The receiving system of claim 9, wherein the first error correction decoder is further for performing a Reed-Solomon (RS) decoding process with (18, 10) RS code on the demultiplexed TPC data.]

[16. The receiving system of claim 9, wherein the second error correction decoder is further for performing a Reed-Solomon (RS) decoding process with (51, 37) RS code on the block deinterleaved FIC data.]

17. A method of processing data in a broadcast transmitter, the method comprising:

encoding broadcast service data in data units;

performing permuting on the encoded broadcast service data;

performing block interleaving on the permuted broadcast service data;

performing convolutional interleaving on the block-interleaved broadcast service data;

encoding first signaling data for signaling the broadcast service data, the first signaling data including encoding information related to the broadcast service data; and

transmitting a frame including the convolutional interleaved broadcast service data, the encoded first signaling data and second signaling data, the second signaling data including information for service acquisition,

wherein each of the data units includes a header and a payload,

wherein the header includes stuffing indication information indicating whether stuffing data are included in the data unit,

wherein the payload includes the broadcast service data,

wherein the frame is composed of multiple sub-frames that are concatenated together in time,

wherein the first signaling data further include sub-frame information related to the sub-frames,

wherein the frame further includes known data, wherein the known data include a first pattern of known data and a second pattern of known data other than the first pattern of known data, and wherein at least one of the first pattern of known data and the second pattern of known data are used for channel estimation.

18. The method of claim 17, wherein the broadcast service data in the frame are categorized into a first type of broadcast service data and a second type of broadcast service data.

19. The method of claim 18, wherein the known data are inserted into specific positions of the frame.

20. The method of claim 17, wherein the header further includes information for identifying a data type of the payload.

21. The method of claim 17, wherein the first signaling data further include information to indicate a change of the second signaling data.

22. A broadcast transmitter for processing data, the broadcast transmitter comprising:

a first encoder to encode broadcast service data in data units;

a permuting unit to perform permuting on the encoded broadcast service data;

a first interleaver to perform block interleaving on the permuted broadcast service data;

a second encoder to encode first signaling data for signaling the broadcast service data, the first signaling data including encoding information related to the broadcast service data;

a second interleaver to perform convolutional interleaving on the block-interleaved broadcast service data; and

a transmitter to transmit a frame including the convolutional interleaved broadcast service data, the encoded first signaling data and second signaling data, the second signaling data including information for service acquisition,

wherein each of the data units includes a header and a payload,

wherein the header includes stuffing indication information indicating whether stuffing data are included in the data unit,

wherein the payload includes the broadcast service data,

wherein the frame is composed of multiple sub-frames that are concatenated together in time,

wherein the first signaling data further include sub-frame information related to the sub-frames,

wherein the frame further includes known data, wherein the known data include a first pattern of known data and a second pattern of known data other than the first pattern of known data, and wherein at least one of the first pattern of known data and the second pattern of known data are used for channel estimation.

23. The broadcast transmitter of claim 22, wherein the broadcast service data in the frame are categorized into a first type of broadcast service data and a second type of broadcast service data.

24. The broadcast transmitter of claim 23, wherein the known data are inserted into specific positions of the frame.

25. The broadcast transmitter of claim 22, wherein the header further includes information for identifying a data type of the payload.

26. The broadcast transmitter of claim 22, wherein the first signaling data further include information to indicate a change of the second signaling data.

* * * * *