



US00RE46437E

(19) **United States**
(12) **Reissued Patent**
Kim et al.

(10) **Patent Number:** **US RE46,437 E**
(45) **Date of Reissued Patent:** **Jun. 13, 2017**

(54) **DTV TRANSMITTING SYSTEM AND
METHOD OF PROCESSING DTV SIGNAL**

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

(72) Inventors: **Byoung Gill Kim**, Seoul (KR); **In
Hwan Choi**, Gwacheon-si (KR); **Jin
Woo Kim**, Seoul (KR); **Jong Moon
Kim**, Gwangmyeong-si (KR); **Won
Gyu Song**, Seoul (KR); **Hyoung Gon
Lee**, Seoul (KR); **Kook Yeon Kwak**,
Anyang-si (KR)

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

(21) Appl. No.: **14/514,056**

(22) Filed: **Oct. 14, 2014**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **8,315,334**
Issued: **Nov. 20, 2012**
Appl. No.: **13/181,439**
Filed: **Jul. 12, 2011**

U.S. Applications:

(63) Continuation of application No. 14/326,296, filed on
Jul. 8, 2014, which is an application for the reissue of
(Continued)

Foreign Application Priority Data

Apr. 6, 2007 (KR) 10-2007-0033986

(51) **Int. Cl.**
H04L 27/00 (2006.01)
H04N 5/38 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC **H04L 1/0047** (2013.01); **H04L 1/0065**
(2013.01); **H04L 1/0071** (2013.01);
(Continued)

(58) **Field of Classification Search**
CPC H04L 1/0047; H04L 1/0065; H04L 1/006;
H04L 1/0057; H04L 1/0071;
(Continued)

References Cited

U.S. PATENT DOCUMENTS

5,596,604 A * 1/1997 Cioffi et al. 375/260
6,618,452 B1 9/2003 Huber et al.
(Continued)

FOREIGN PATENT DOCUMENTS

EP 1791368 5/2007
KR 1020010043503 5/2001
(Continued)

OTHER PUBLICATIONS

Korean Intellectual Property Office Application Serial No. 10-2007-
0033986, Notice of Allowance dated May 29, 2013, 2 pages.

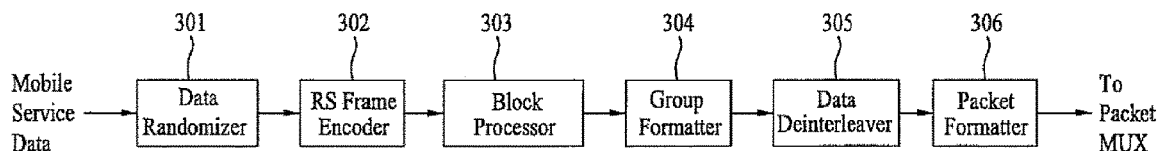
Primary Examiner — Christina Y Leung

(74) *Attorney, Agent, or Firm* — Dentons US LLP

(57) ABSTRACT

A digital television (DTV) transmitting system is provided that includes an encoder, a group formatter, a packet formatter and a transmission unit. The group formatter forms data groups where the plurality of second known data sequences are spaced 16 segments apart within at least one of the data groups that includes a transmission parameter inserted between the first known data sequence and the plurality of second known data sequences and the first known data sequence has a first M-symbol sequence and a second M-symbol sequence, the first M-symbol sequence and the second M-symbol sequence have a first pattern, each of the plurality of second known data sequences has a second pattern other than the first pattern, and the second pattern is positioned from a last symbol to a previous N symbol in each of the plurality of second known data sequences.

1 Claim, 26 Drawing Sheets



US RE46,437 E

Page 2

Related U.S. Application Data			(56)	References Cited		
Pat. No. 8,315,334, which is a continuation of application No. 11/774,545, filed on Jul. 6, 2007, now Pat. No. 8,009,766.			U.S. PATENT DOCUMENTS			
(60)	Provisional application No. 60/911,507, filed on Apr. 12, 2007.		6,628,926	B1	9/2003	van de Beek et al.
			6,996,418	B2 *	2/2006	Teo et al. 455/562.1
			7,006,848	B2 *	2/2006	Ling et al. 455/562.1
			7,030,935	B2	4/2006	Choi et al.
			7,593,474	B2	9/2009	Jeong et al.
			7,711,029	B2 *	5/2010	Guey 375/133
(51)	Int. Cl.		7,839,950	B2	11/2010	Choi et al.
	<i>H04L 1/00</i> (2006.01)		8,009,766	B2 *	8/2011	Kim et al. 375/316
	<i>H04N 21/438</i> (2011.01)		2001/0004390	A1	6/2001	Pukkila et al.
	<i>H04L 25/03</i> (2006.01)		2001/0008542	A1 *	7/2001	Wiebke H04J 13/0044 375/141
	<i>H04N 5/44</i> (2011.01)		2003/0223516	A1	12/2003	Zhang et al.
	<i>H04N 21/61</i> (2011.01)		2003/0223519	A1	12/2003	Jeong et al.
	<i>H04L 25/02</i> (2006.01)		2004/0001554	A1 *	1/2004	Sun et al. 375/260
	<i>H04N 21/2389</i> (2011.01)		2004/0071234	A1	4/2004	Li
			2005/0135308	A1 *	6/2005	Vijayan et al. 370/330
(52)	U.S. Cl.		2005/0226414	A1	10/2005	Lee et al.
	CPC <i>H04L 25/0232</i> (2013.01); <i>H04L 25/03159</i> (2013.01); <i>H04N 5/4401</i> (2013.01); <i>H04N 21/4382</i> (2013.01); <i>H04N 21/6131</i> (2013.01); <i>H04L 1/006</i> (2013.01); <i>H04L 1/0057</i> (2013.01); <i>H04L 1/0061</i> (2013.01); <i>H04L 25/03273</i> (2013.01); <i>H04L 25/03292</i> (2013.01); <i>H04L 2027/0026</i> (2013.01); <i>H04N 21/2389</i> (2013.01)		2007/0081484	A1	4/2007	Wang
			2007/0121681	A1	5/2007	Kang et al.
			2007/0230580	A1	10/2007	Kim et al.
			2008/0024688	A1	1/2008	Mizoguchi
			2009/0037792	A1	2/2009	Choi et al.
			2009/0044072	A1	2/2009	Oh et al.
			2009/0052587	A1	2/2009	Song et al.
			2009/0080574	A1	3/2009	Choi et al.
			2009/0083606	A1	3/2009	Choi et al.
			2010/0050047	A1	2/2010	Choi et al.
			2010/0309969	A1	12/2010	Kim et al.
(58)	Field of Classification Search		2010/0310015	A1	12/2010	Choi et al.
	CPC H04L 1/0061; H04L 25/232; H04L 25/03273; H04L 25/03159; H04L 25/03292; H04L 2027/0026; H04N 5/4401; H04N 21/6131; H04N 21/4382; H04N 21/2389		2011/0267545	A1 *	11/2011	Kim H04L 1/0047 348/720
	USPC 375/240.07, 240.27, 295, 265, 285, 296, 375/346, 350; 348/723, 724		FOREIGN PATENT DOCUMENTS			
	See application file for complete search history.		KR	1020070055117	5/2007	
			WO	99/59331	11/1999	
			WO	2005/101655	10/2005	
			WO	2005109878	11/2005	
			* cited by examiner			

FIG. 1

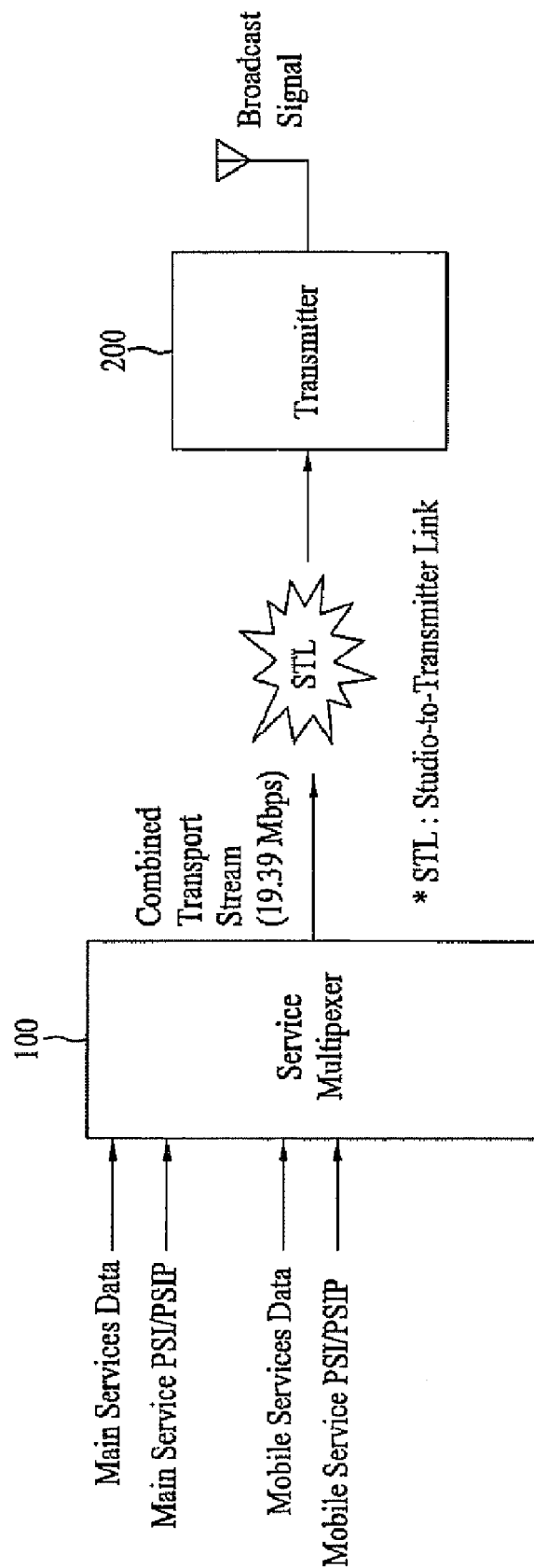


FIG. 2

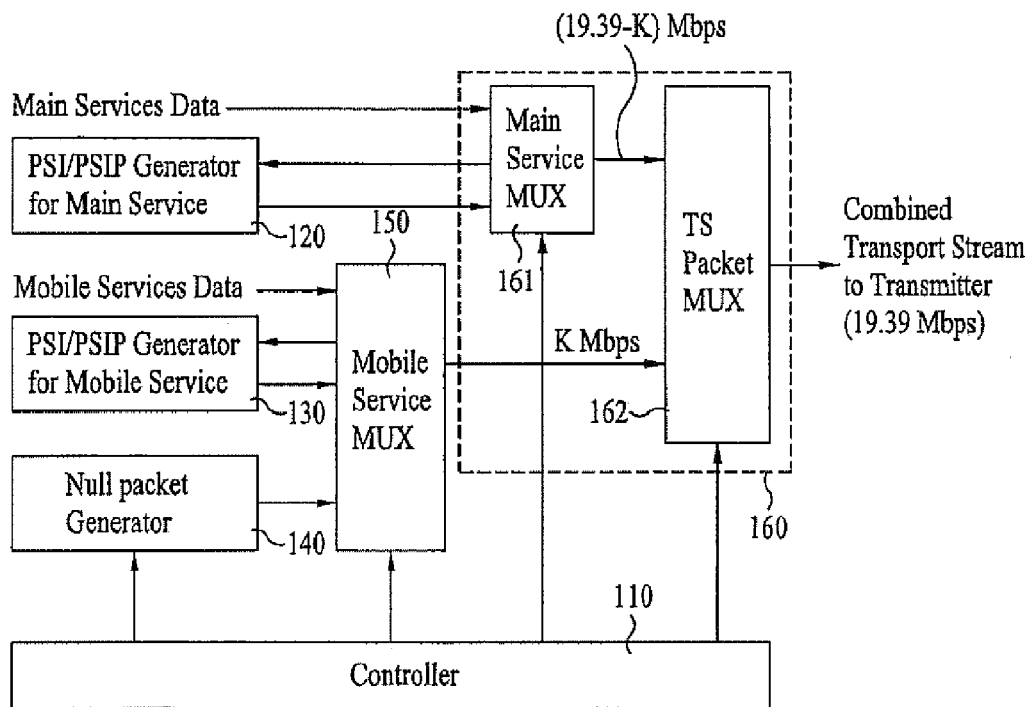


FIG. 3

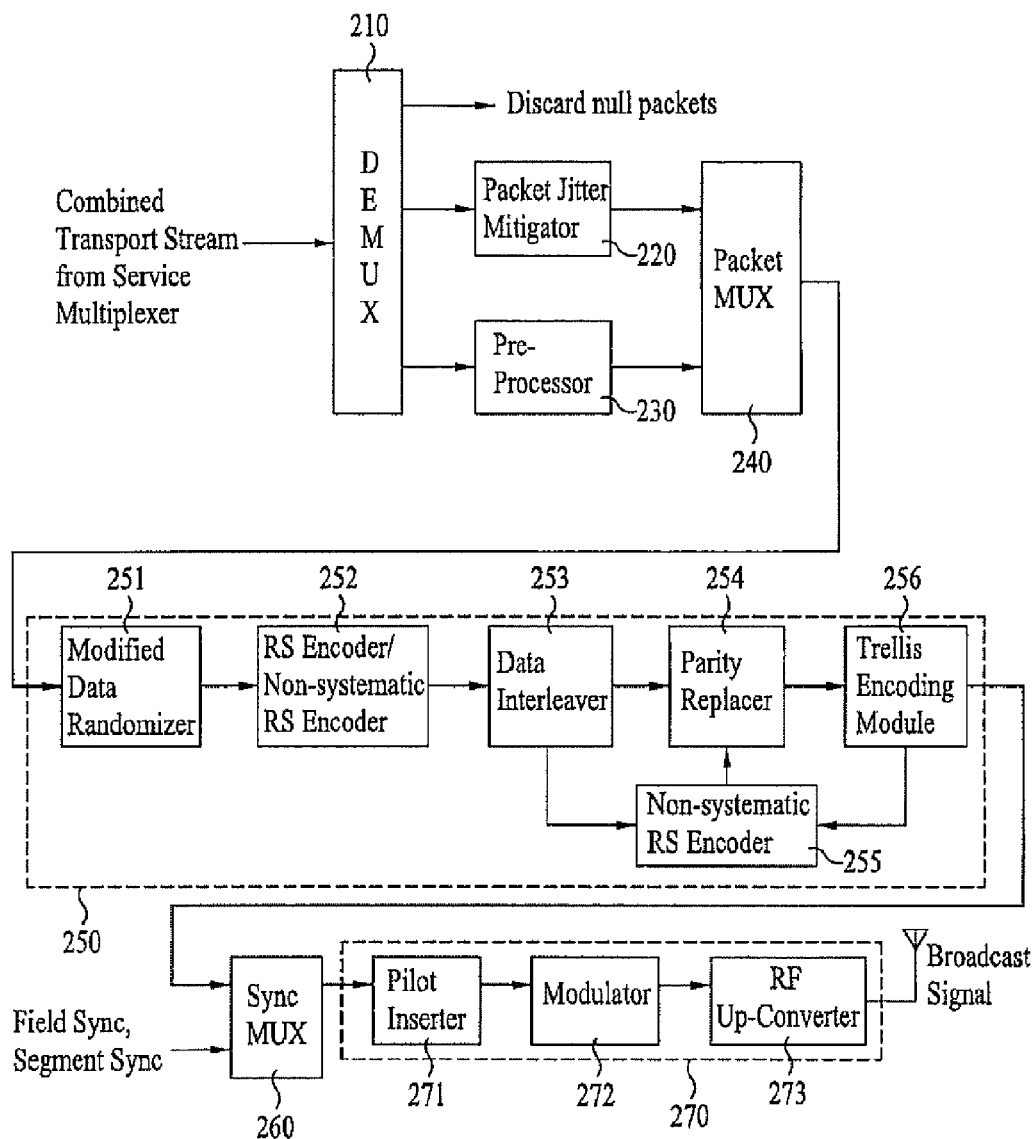


FIG. 4

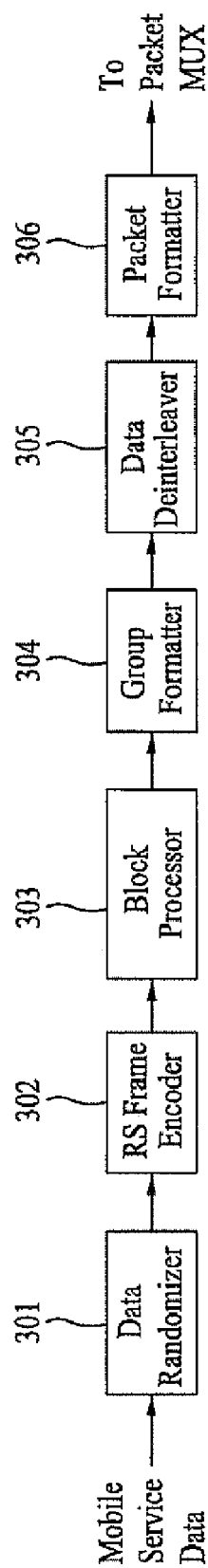


FIG. 5

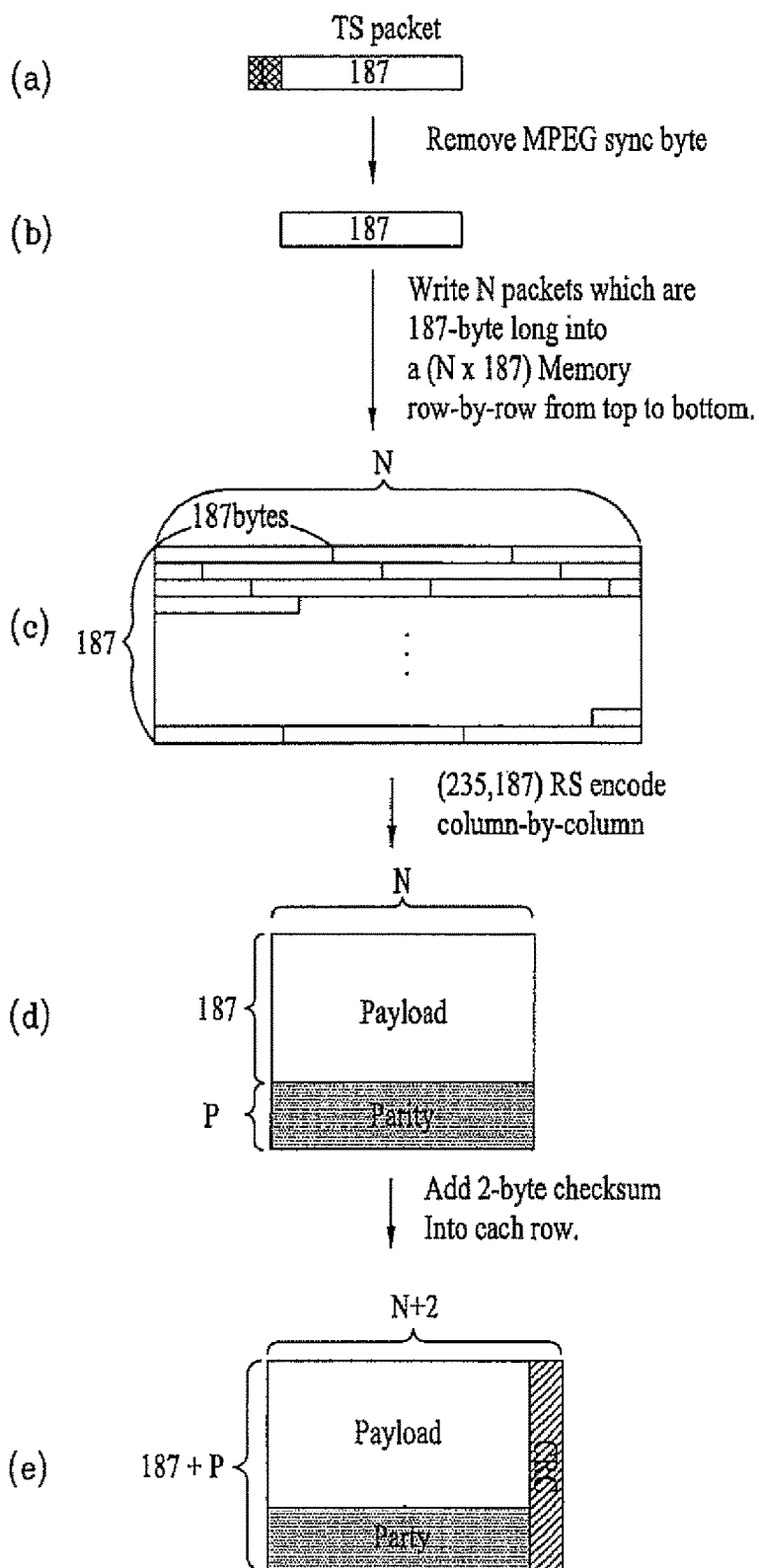


FIG. 6A

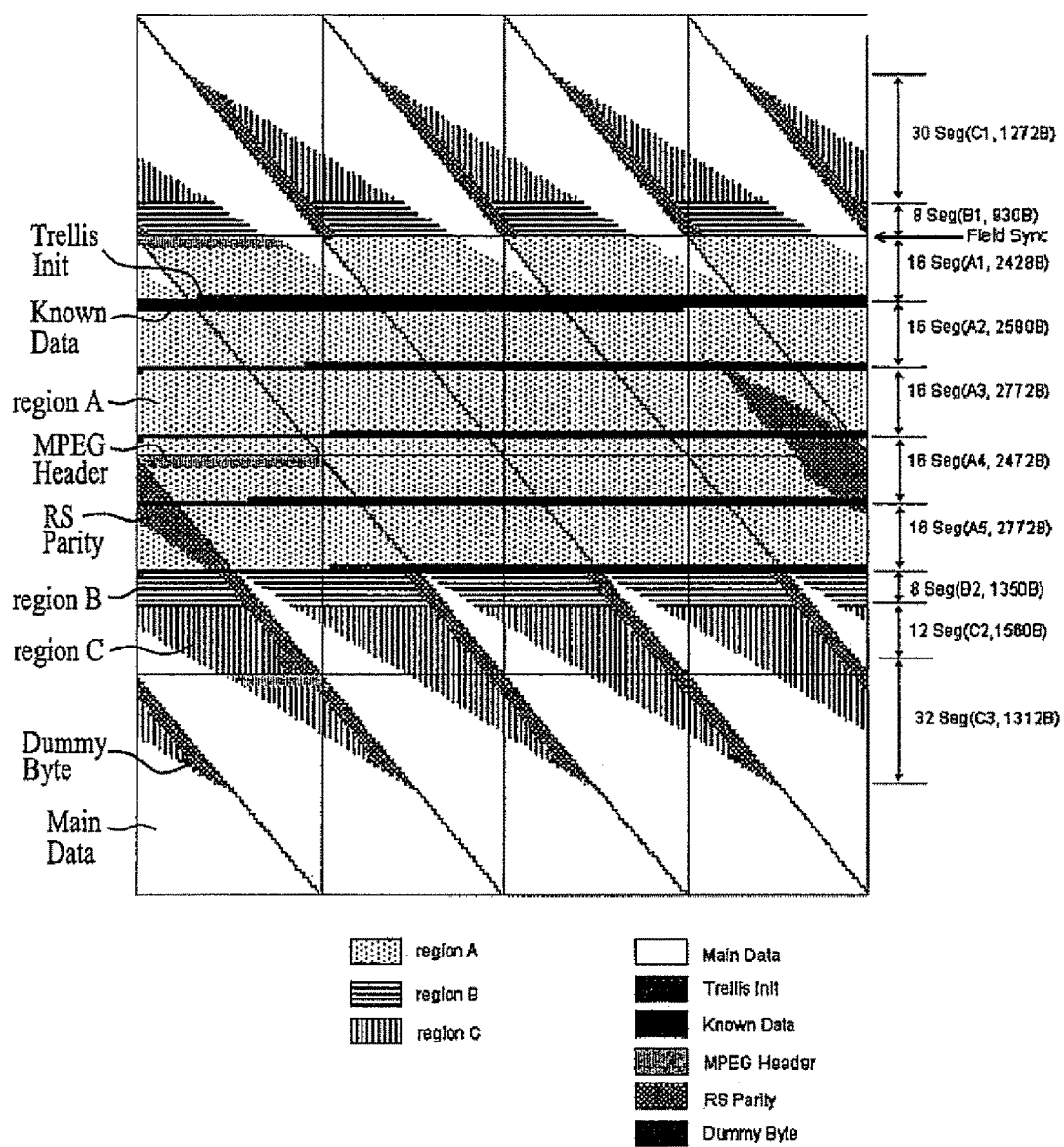


FIG. 6B

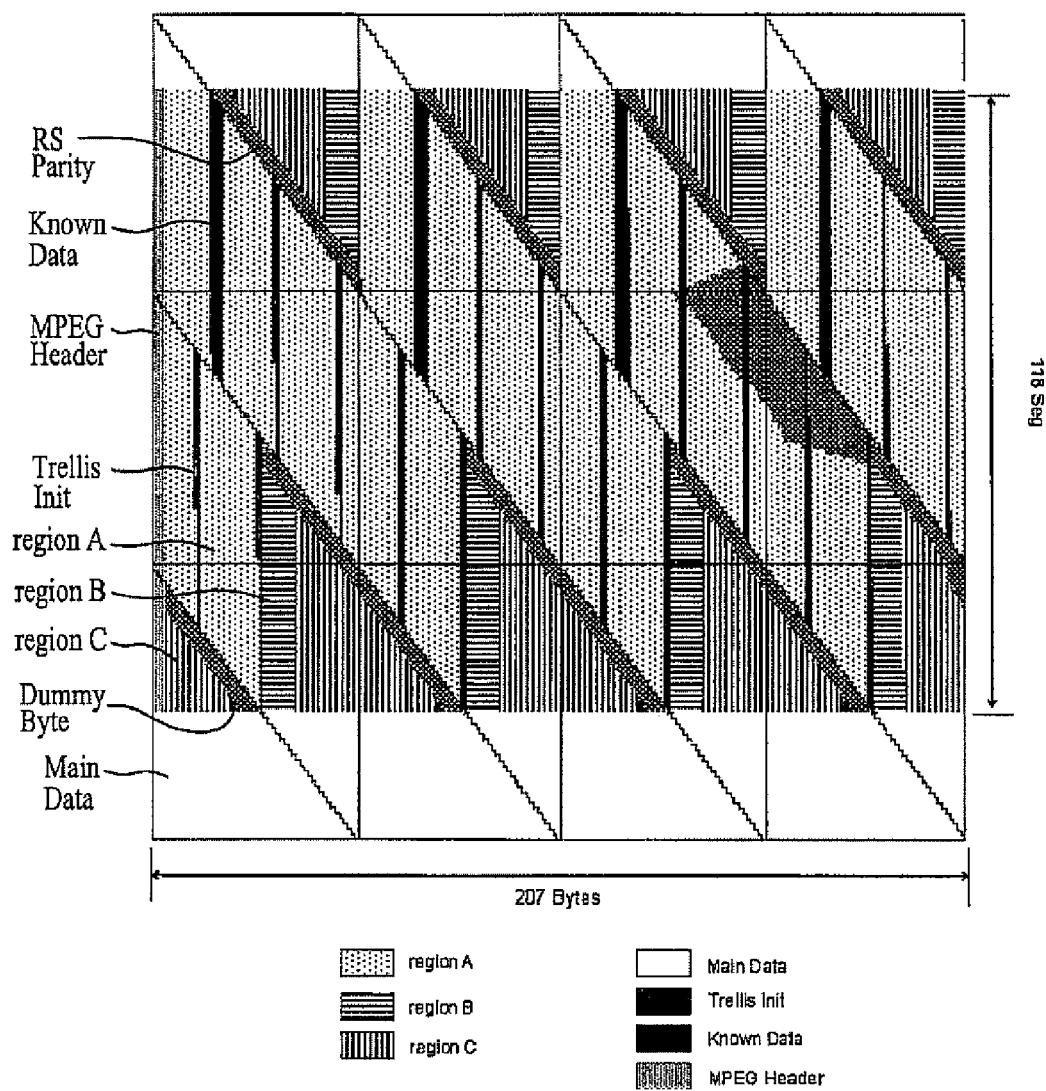


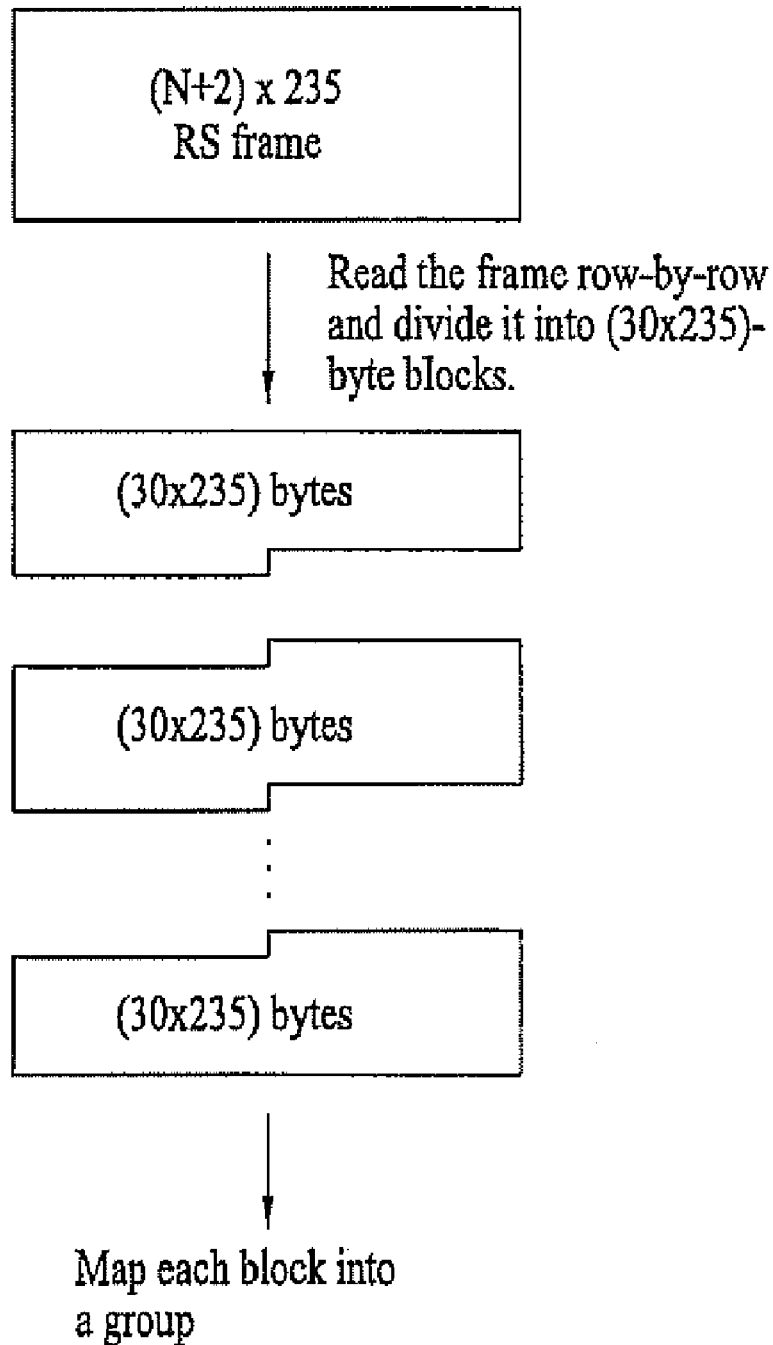
FIG. 7

FIG. 8

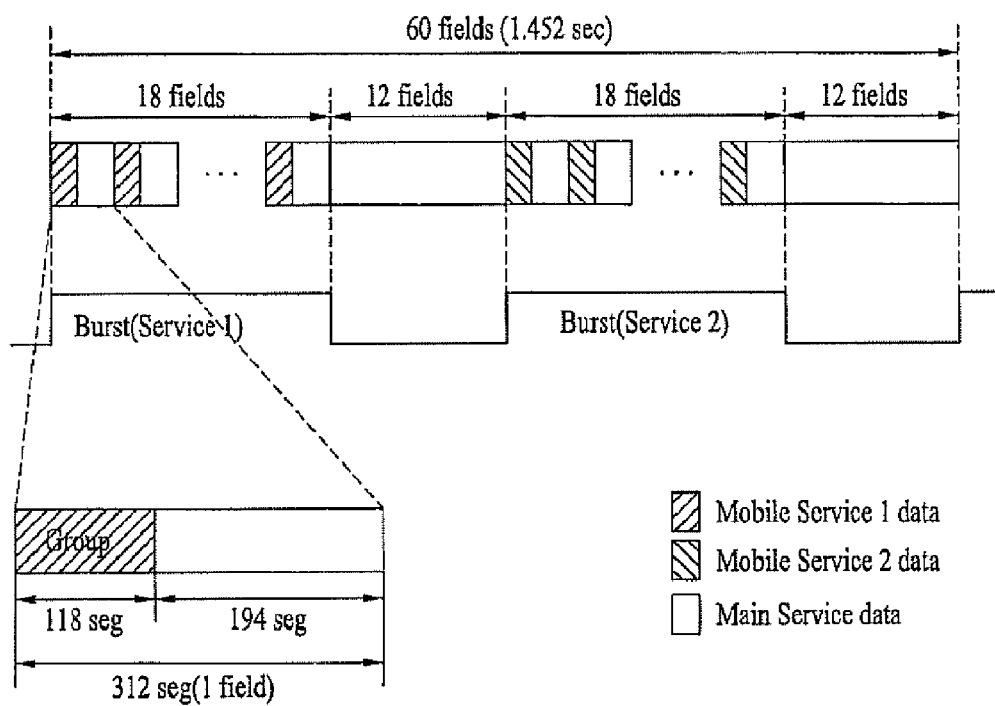


FIG. 9

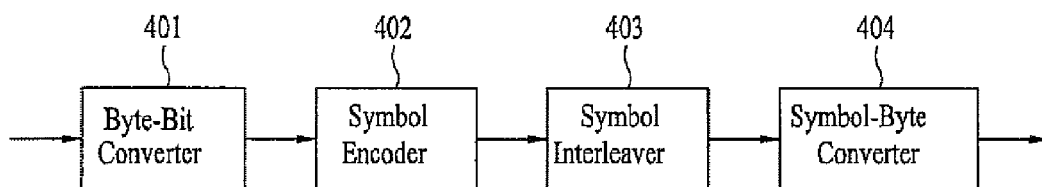


FIG. 10

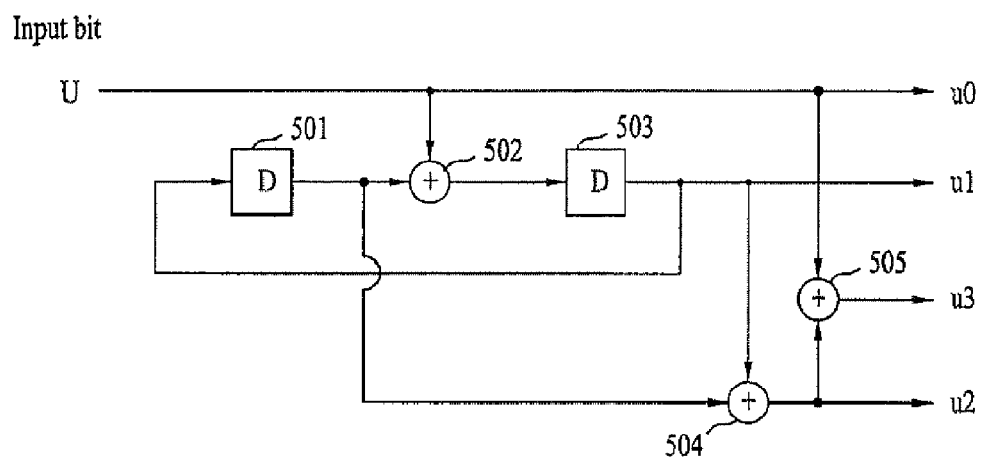


FIG. 11

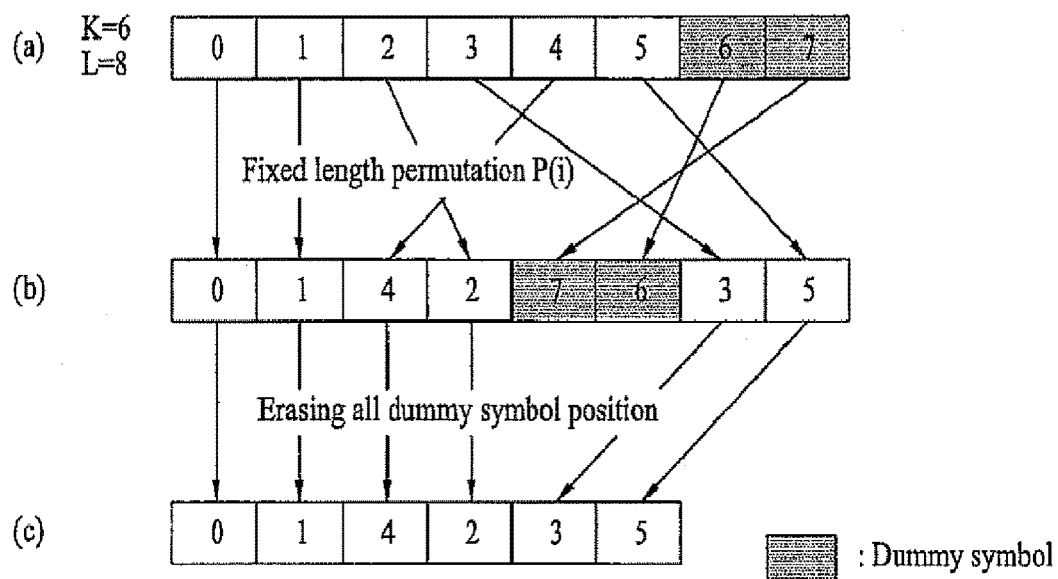


FIG. 12A

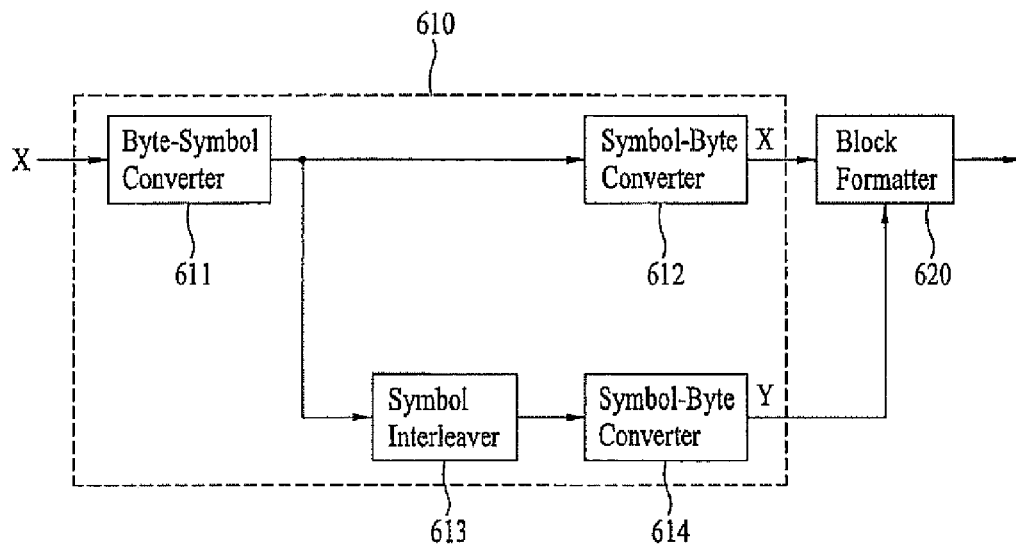


FIG. 12B

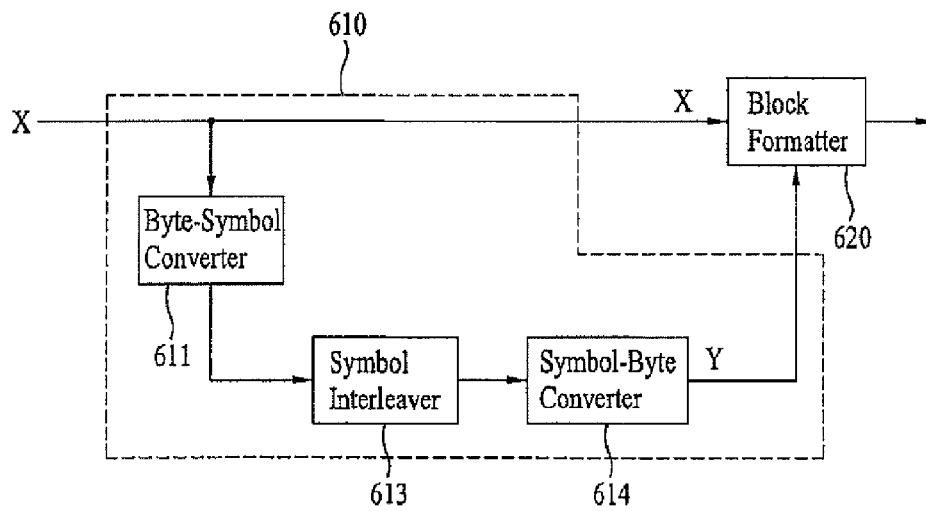


FIG. 13

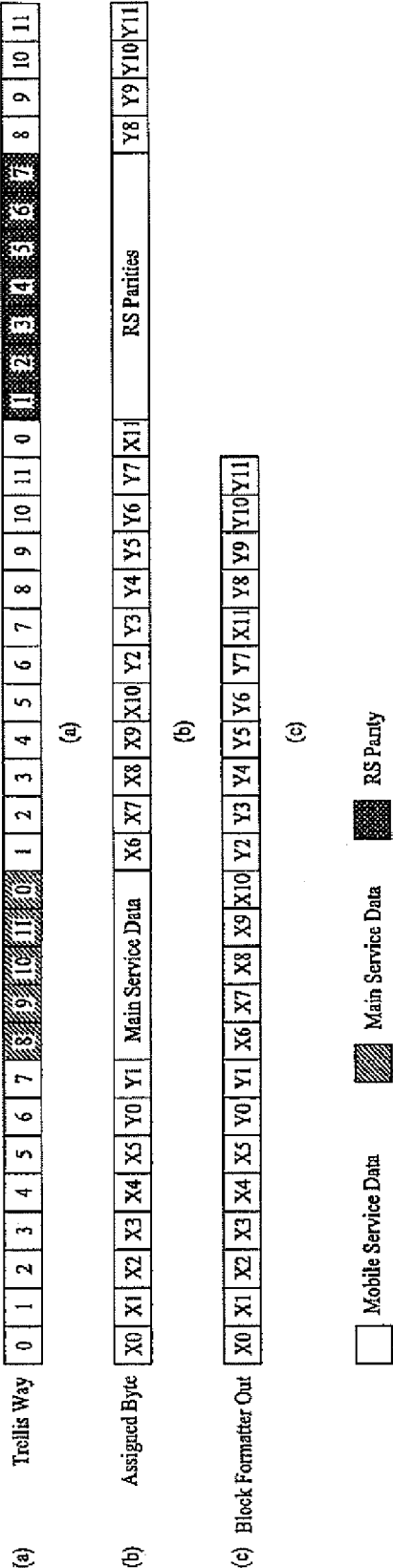


FIG. 14

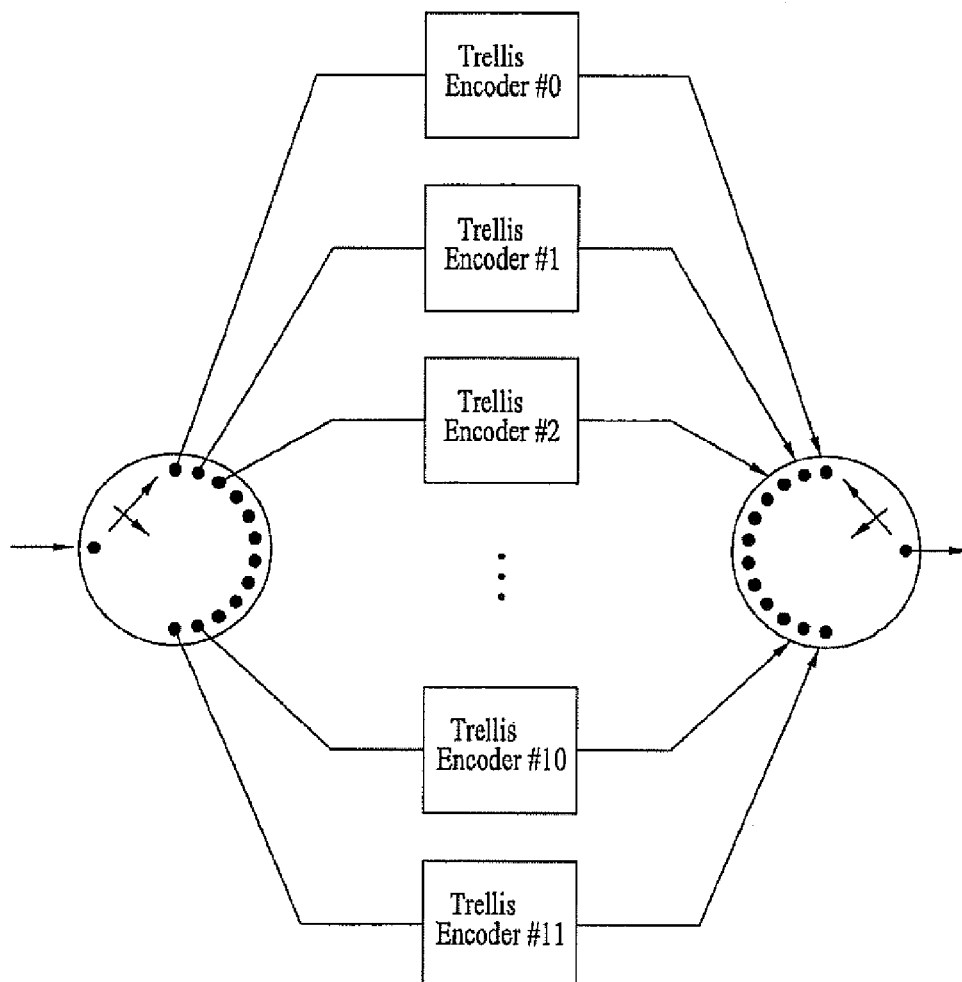


FIG. 15A

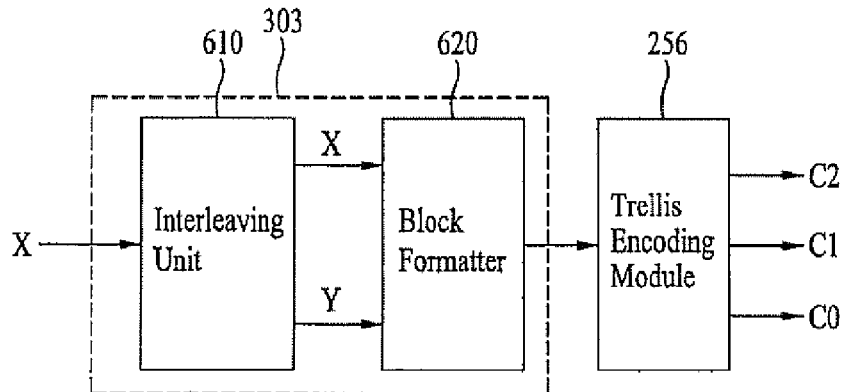


FIG. 15B

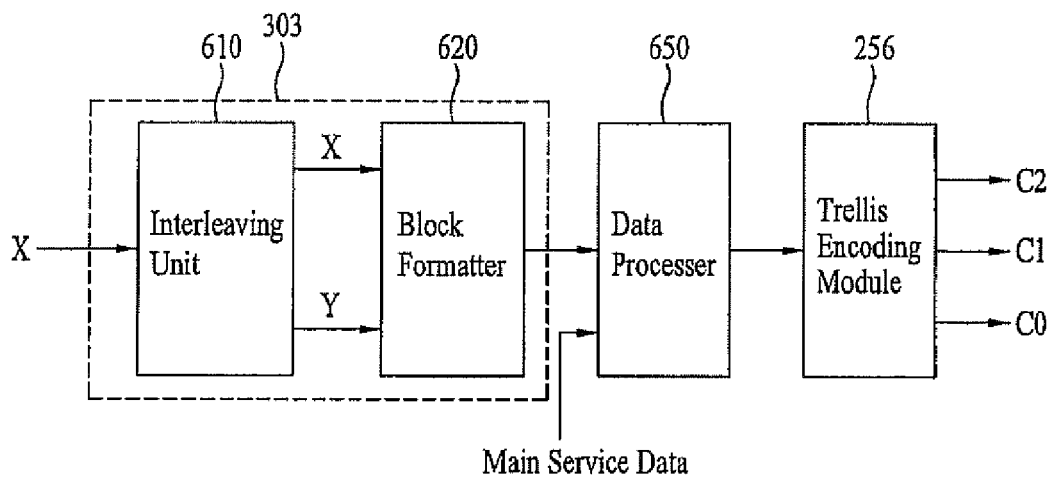


FIG. 16

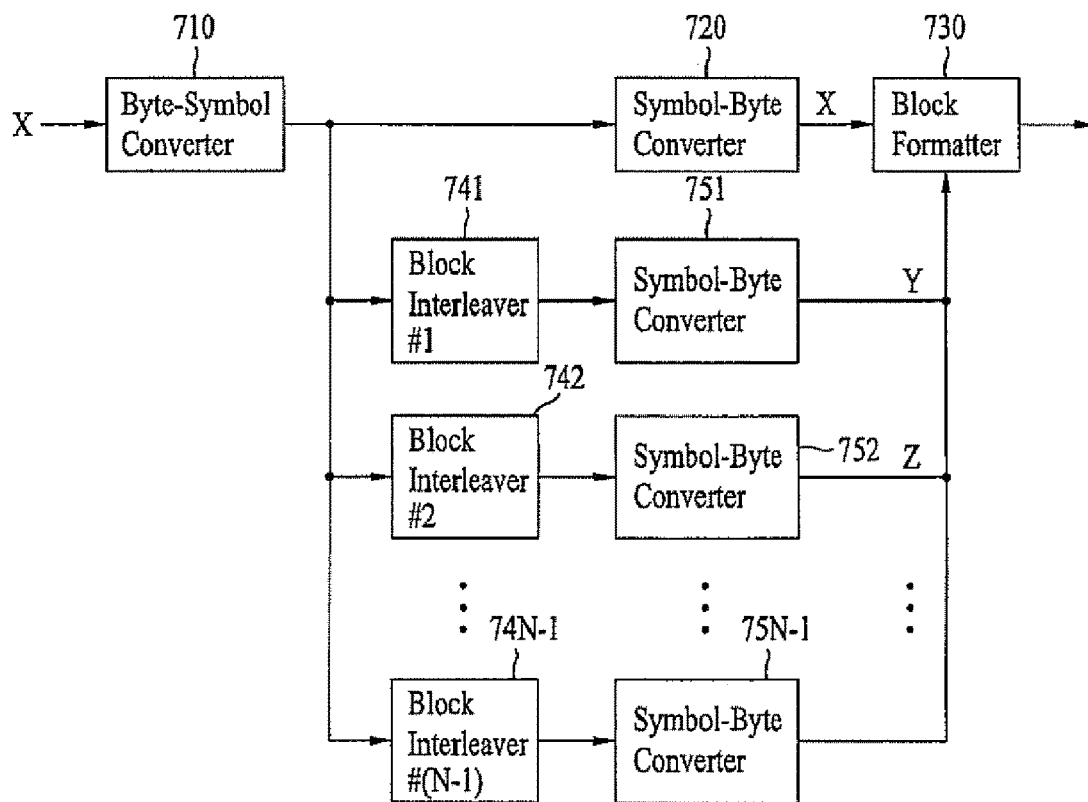


FIG. 17

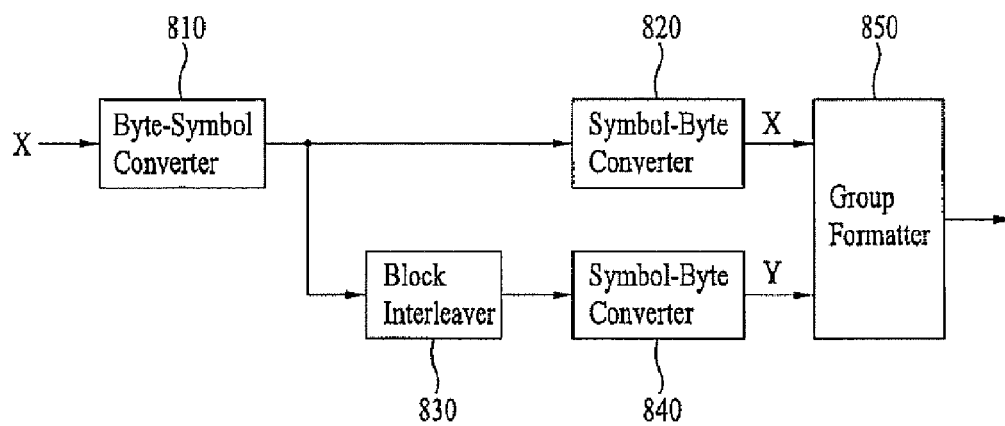


FIG. 18

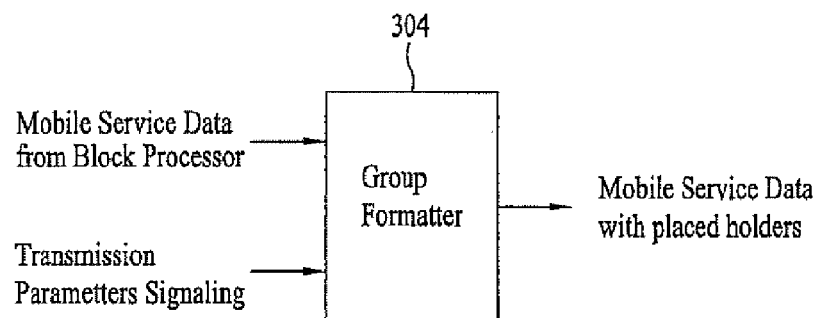


FIG. 19

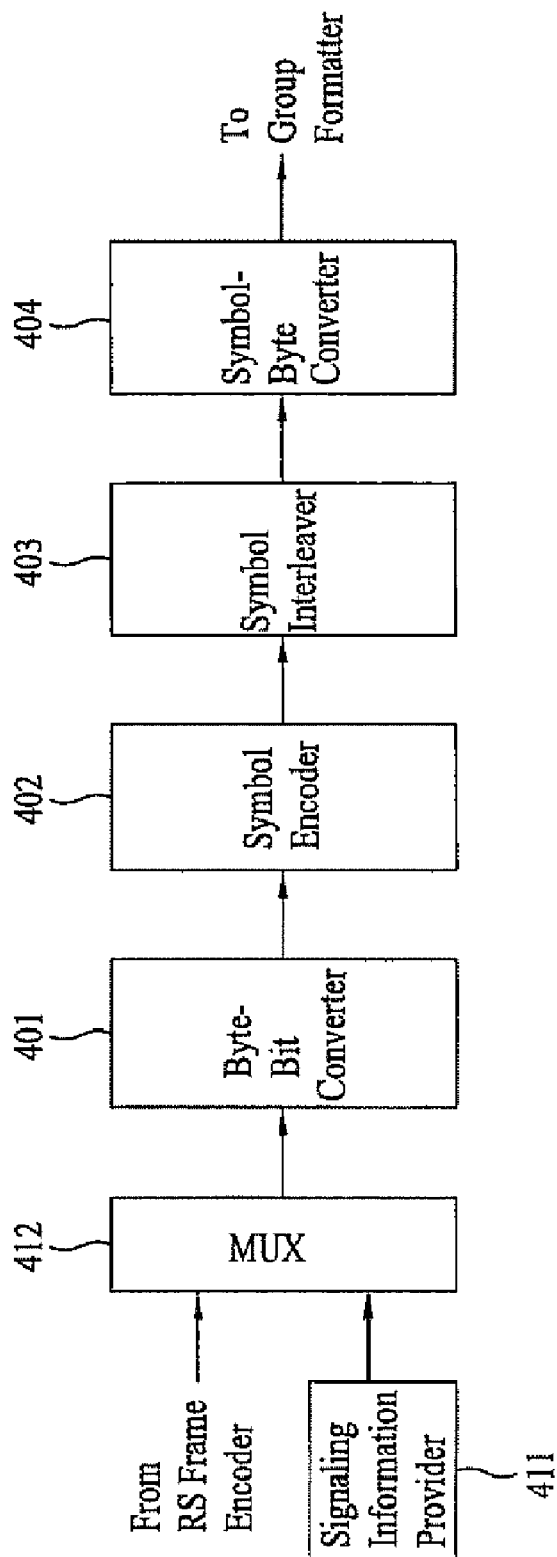


FIG. 20

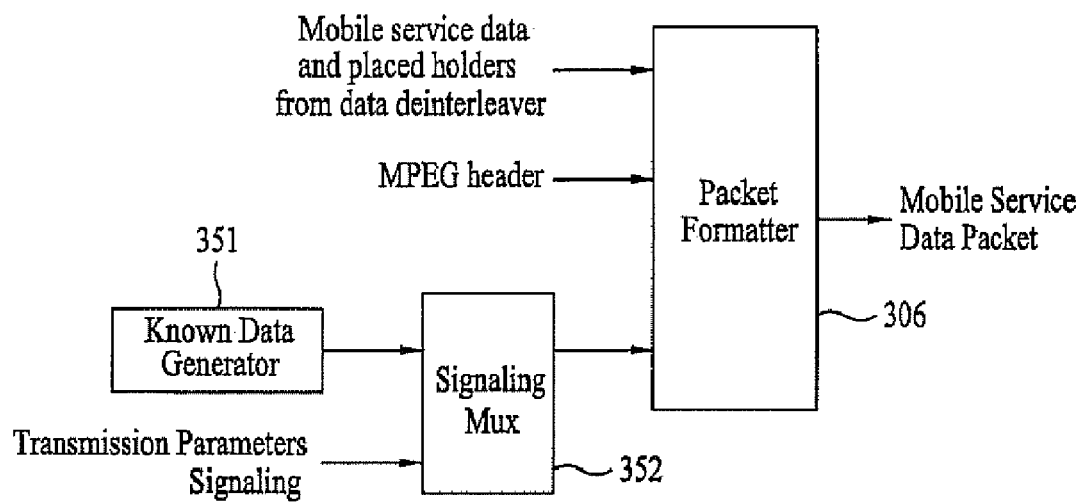


FIG. 21

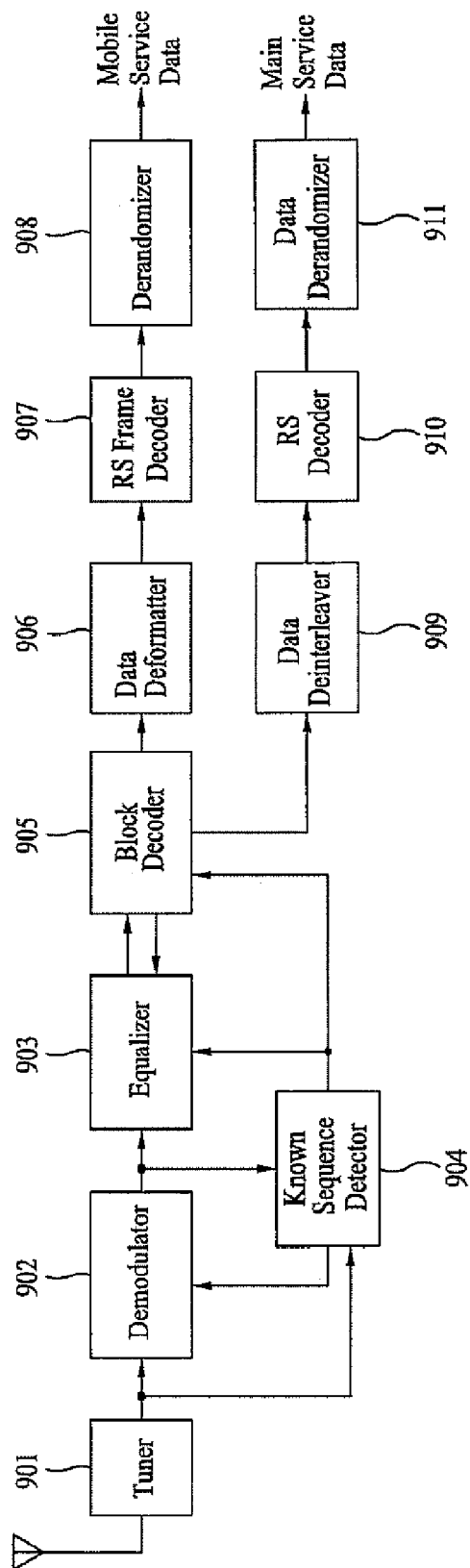


FIG. 23

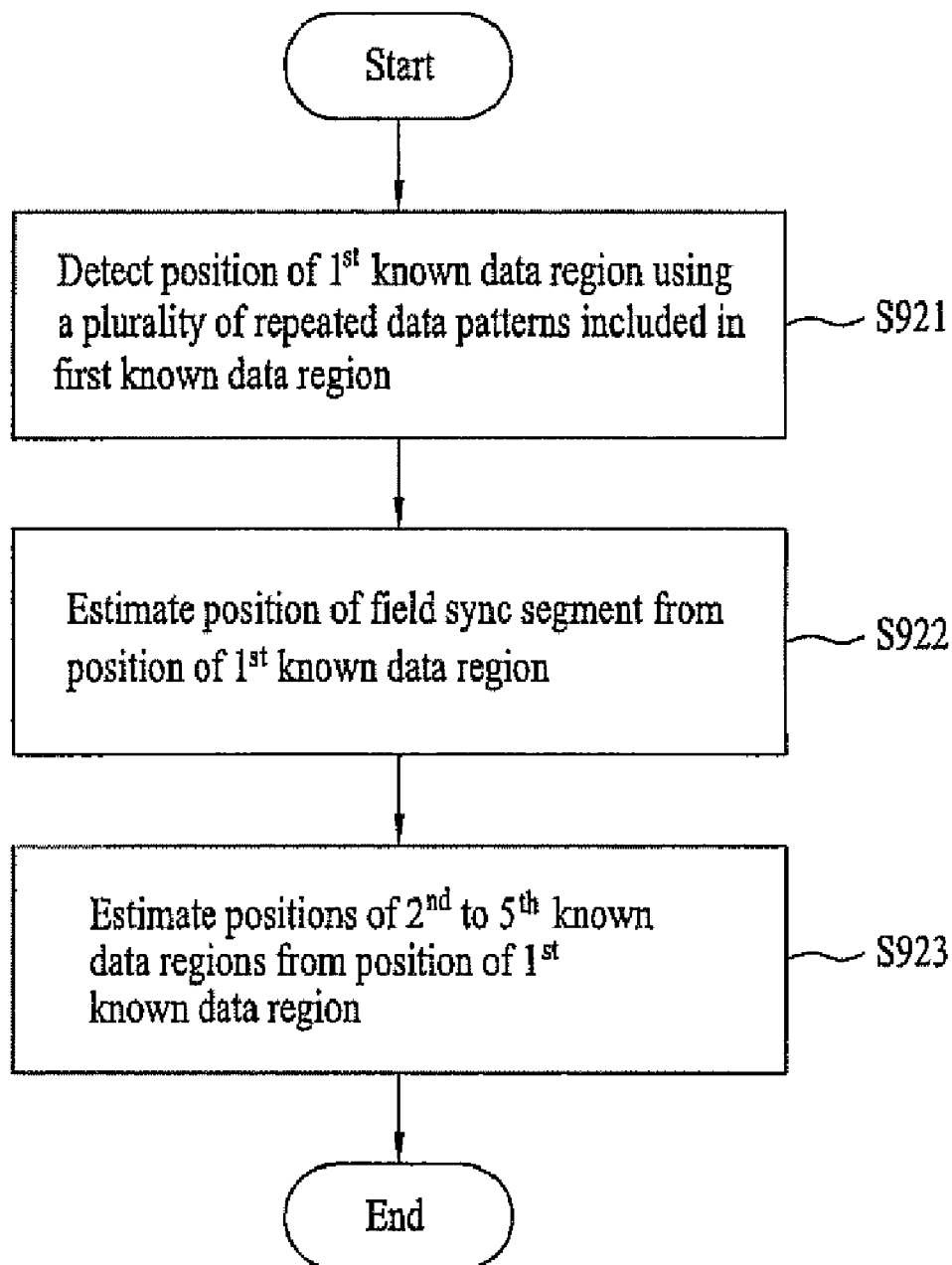


FIG. 24

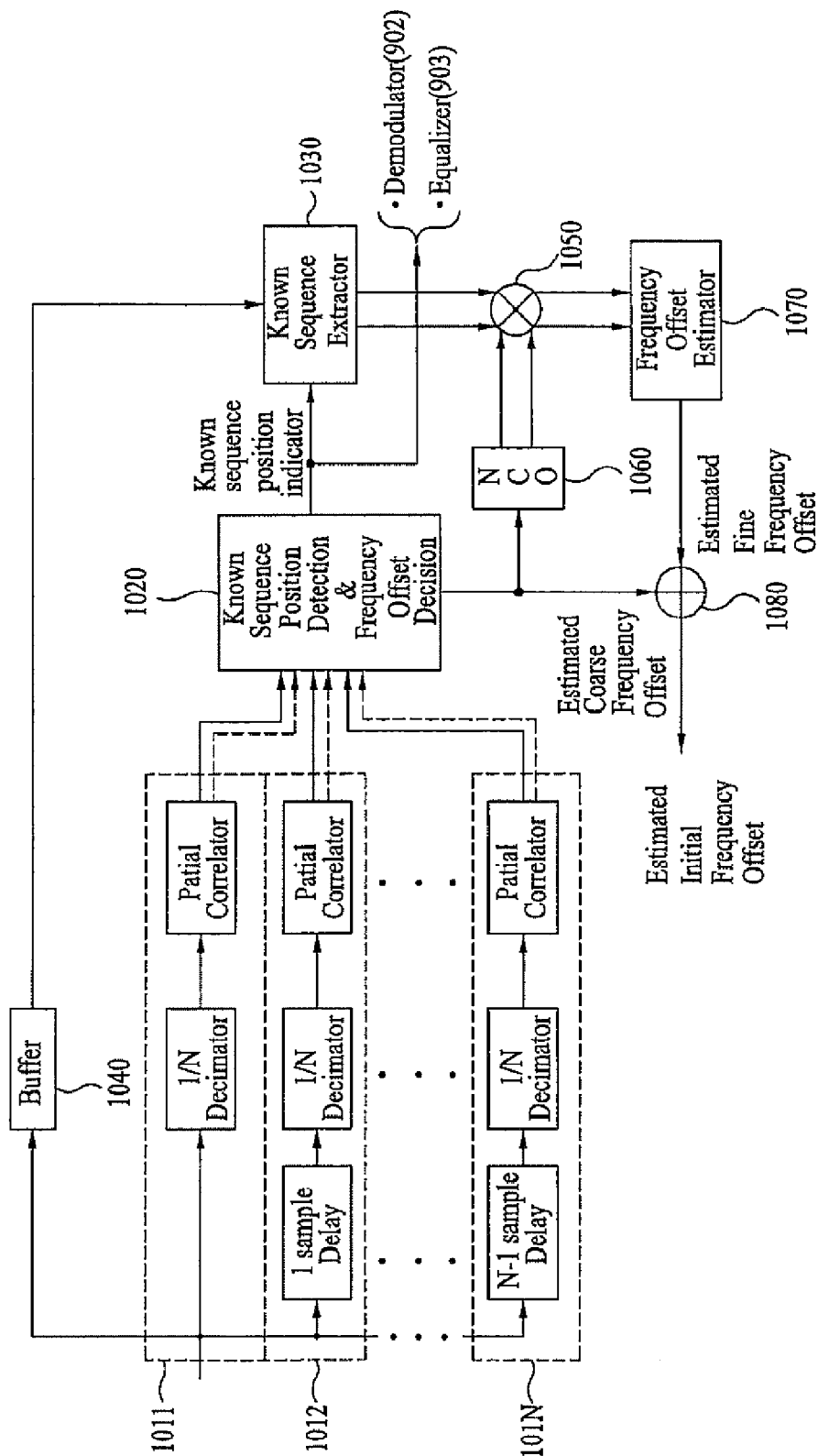


FIG. 25

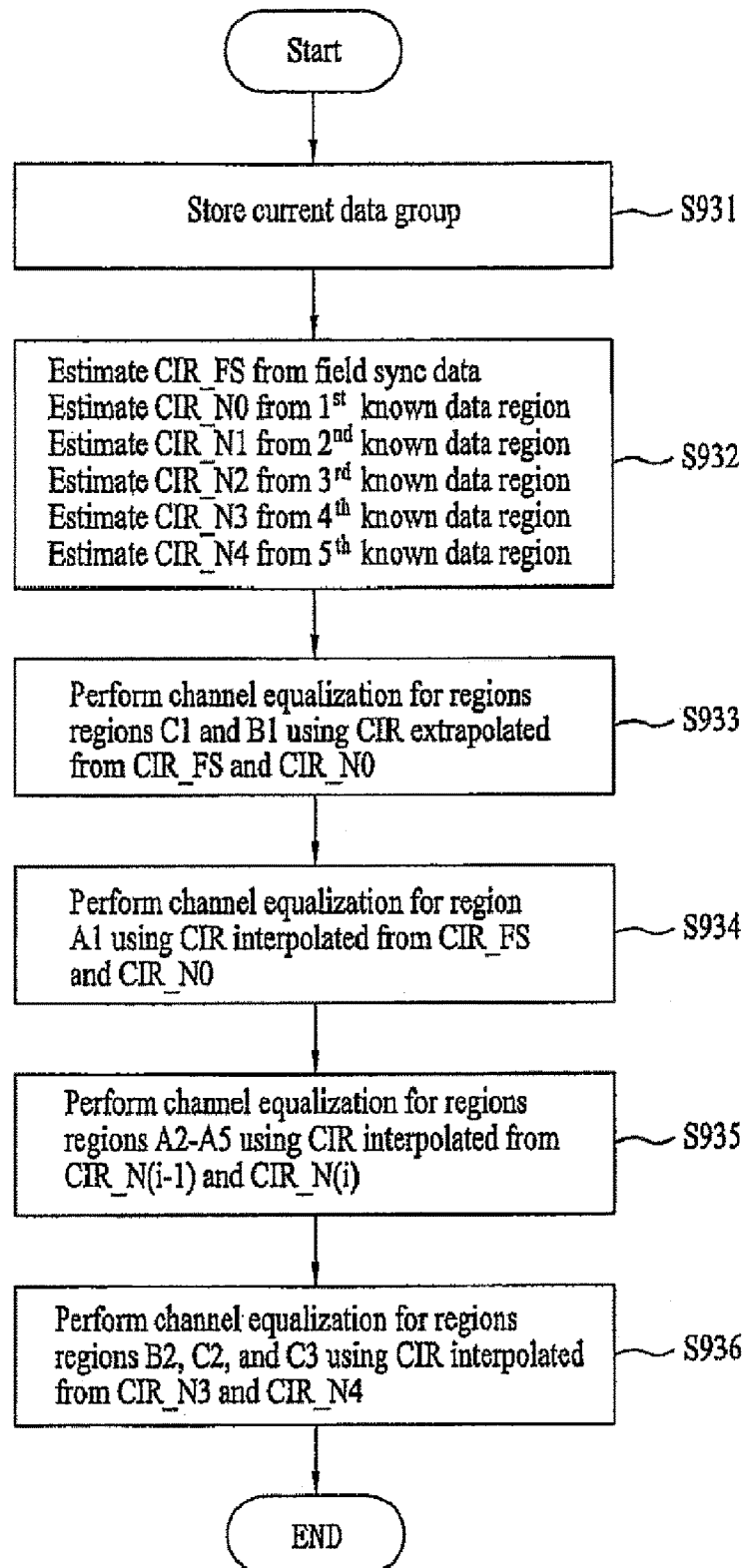


FIG. 26

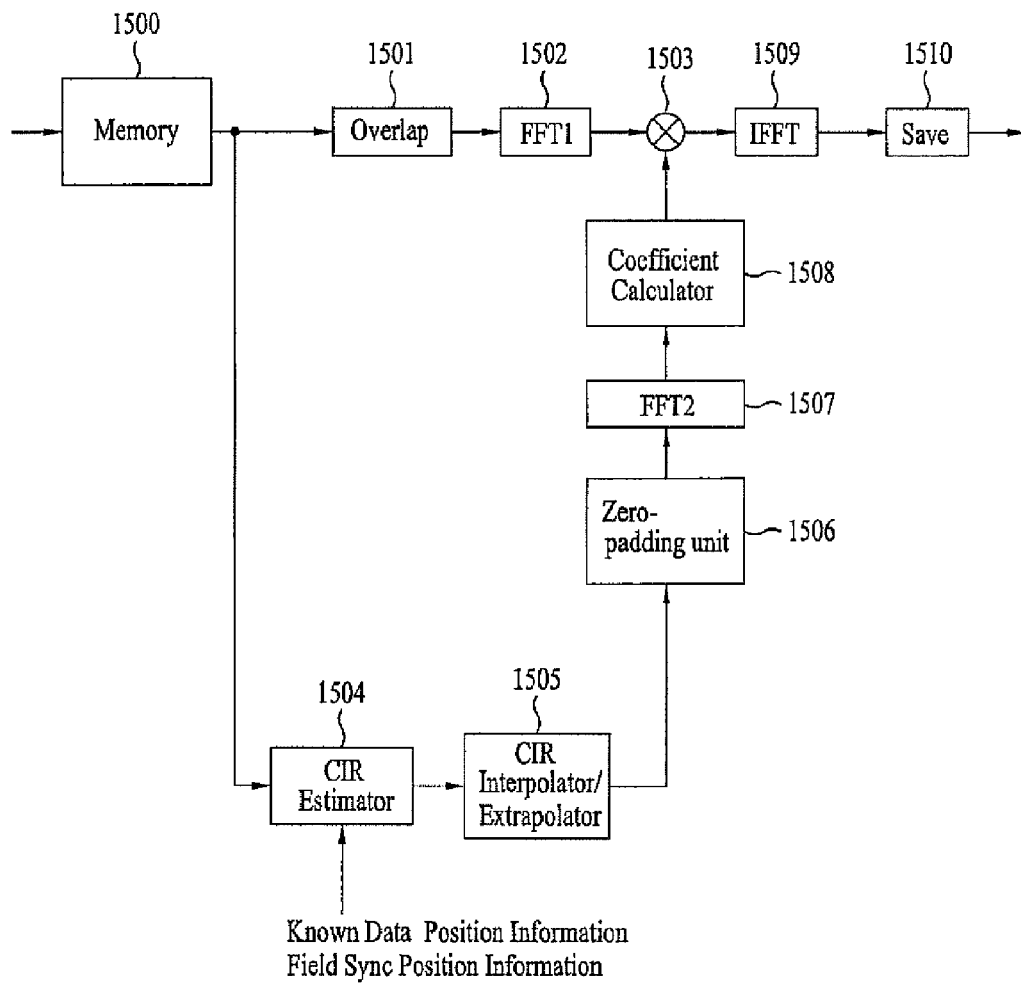
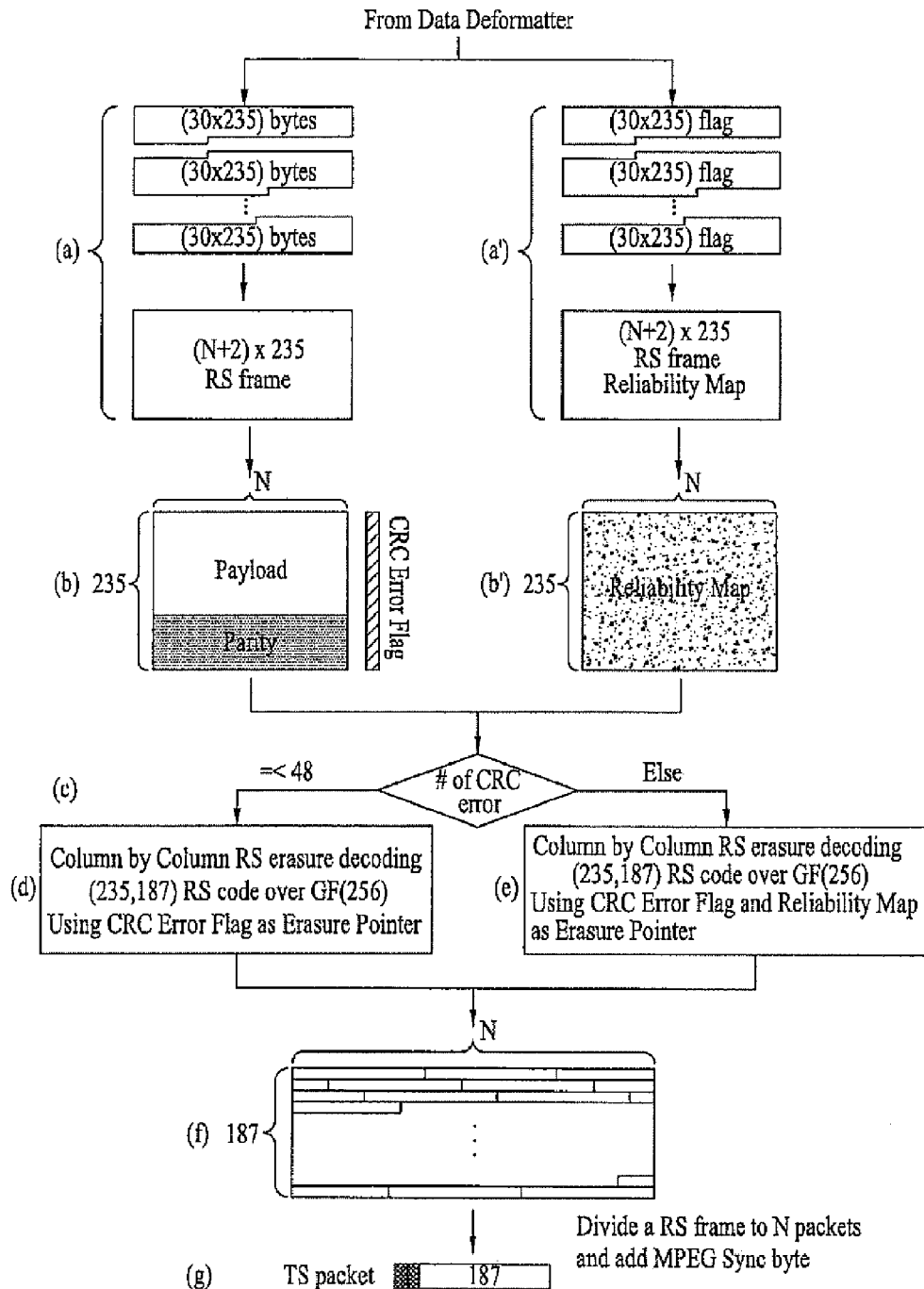


FIG. 27



DTV TRANSMITTING SYSTEM AND METHOD OF PROCESSING DTV SIGNAL

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. 14/326,296, filed on Jul. 8, 2014, currently pending, which is reissue application of U.S. Pat. No. 8,315,334 B2, issued from U.S. application Ser. No. 13/181,439, filed Jul. 12, 2011, which is a continuation of U.S. application Ser. No. 11/774,545, filed on Jul. 6, 2007, now U.S. Pat. No. 8,009,766, which claims the benefit of earlier filing date and right to priority to Korean Patent Application No. 10-2007-0033986, filed on Apr. 6, 2007, and claims the benefit of U.S. Provisional Application No. 60/911,507, filed on Apr. 12, 2007, the contents of *all of* which are [all] hereby incorporated by reference herein in their [entirety] *entireties*.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a digital television (DTV) systems and methods of processing television signals.

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 receiving systems, 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 DTV systems and methods of processing television signals 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 DTV systems and methods of processing television signals that are highly resistant to channel changes and noise.

Another object of the present invention is to provide DTV systems and methods of processing television signals that can enhance the receiving performance of a digital broadcast 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 DTV systems and methods of processing television signals that can also enhance the receiving performance of a digital broadcast 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 area within a data area.

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 digital television (DTV) receiving system includes a known data detector configured to detect a known data position and a field sync data position from a data group including field sync data, known data, and mobile service data; a first converter configured to convert the data included in the data group into frequency domain data; a channel impulse response (CIR) estimator configured to estimate CIRs of the data included in the data group using the field sync data and the known data identified by the known data and field sync data positions, respectively; and a CIR calculator configured to interpolate or extrapolate the estimated CIRs. The DTV receiving system further includes a second converter configured to convert the interpolated or extrapolated CIRs into frequency domain CIRs; a coefficient calculator configured calculate equalization coefficients using the frequency domain CIRs; and a multiplier configured to compensate channel distortion of the frequency domain data in the data group by multiplying the frequency domain data with the equalization coefficients.

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 block diagram showing a general structure of a digital broadcasting system according to an embodiment of the present invention;

FIG. 2 illustrates a block diagram showing an example of a service multiplexer of FIG. 1;

FIG. 3 illustrates a block diagram showing an example of a transmitter of FIG. 1;

FIG. 4 illustrates a block diagram showing an example of a pre-processor of FIG. 3;

FIG. 5(a) to FIG. 5(e) illustrate error correction encoding and error detection encoding processed according to an embodiment of the present invention;

FIG. 6A and FIG. 6B illustrate data configuration before and after a data deinterleaver in a digital broadcast transmitting system according to the present invention;

FIG. 7 illustrates a process of dividing a RS frame for configuring a data group according to the present invention;

FIG. 8 illustrates exemplary operations of a packet multiplexer for transmitting the data group according to the present invention;

FIG. 9 illustrates a block diagram showing a structure of a block processor according to the present invention;

FIG. 10 illustrates a detailed block diagram of a symbol encoder shown in FIG. 9;

FIG. 11(a) to FIG. 11(c) illustrate a variable length interleaving process of a symbol interleaver shown in FIG. 9;

FIG. 12A and FIG. 12B illustrate block diagrams showing structures of a block processor according to another embodiment of the present invention;

FIG. 13(a) to FIG. 13(c) illustrate block encoding and trellis encoding processes according to the present invention;

FIG. 14 illustrates a block diagram showing a trellis encoding module according to the present invention;

FIG. 15A and FIG. 15B a block processor and a trellis encoding module connected to one another according to the present invention;

FIG. 16 illustrates a block processor according to another embodiment of the present invention;

FIG. 17 illustrates a block processor according to yet another embodiment of the present invention;

FIG. 18 illustrates an example of a group formatter inserting and transmitting a transmission parameter;

FIG. 19 illustrates an example of a block processor inserting and transmitting a transmission parameter;

FIG. 20 illustrates an example of a packet formatter inserting and transmitting a transmission parameter;

FIG. 21 illustrates a block diagram showing a structure of a digital broadcast receiving system according to the present invention;

FIG. 22 illustrates a part of data region A of the signal frame shown in FIG. 6A;

FIG. 23 is a flow chart illustrating a method of detecting positions of known data according to one aspect of the present invention;

FIG. 24 illustrates a detailed block diagram of a known sequence detector according to one aspect of the present invention;

FIG. 25 is a flow chart illustrating a channel equalization method according to one aspect of the present invention;

FIG. 26 illustrates a channel equalizer according to one aspect of the present invention; and

FIG. 27 illustrates an example of an error correction decoding process according to 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, in the present invention, mobile service data may include at least one of mobile service data, pedestrian service data, and handheld service data, and are collectively referred to as mobile service data for simplicity. Herein, the mobile service data not only correspond to mobile/pedestrian/handheld service data (M/P/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/P/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 serviced 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 & 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.

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 (i.e., 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.

General Description of a Transmitting System

FIG. 1 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,

5

in this case, the plurality of transmitters receives the same signal. 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.

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 mobile service data and program specific information (PSI)/program and system information protocol (PSIP) table data for each mobile service and encapsulates the received data to each transport stream (TS) packet. Also, the service multiplexer **100** receives at least one type of main service data and PSI/PSIP table data for each main service so as to encapsulate the received data to a TS packet. Subsequently, the TS packets are multiplexed according to a predetermined multiplexing rule and outputs the multiplexed packets to the transmitter **200**.

Service Multiplexer

FIG. 2 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 PSI/PSIP generator **120** for the main service, a PSI/PSIP generator **130** for the mobile service, a null packet generator **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. 2, at least one type of compression encoded main service data and the PSI/PSIP table data generated from the PSI/PSIP generator **120** for the main service are inputted to the main service multiplexer **161** of the transport multiplexer **160**. The main service multiplexer **161** encapsulates each of the inputted main service data and PSI/PSIP table data to MPEG-2 TS packet forms. Then, the MPEG-2 TS packets are multiplexed and outputted to the TS packet multiplexer **162**. Herein, the data packet being outputted from the main service multiplexer **161** will be referred to as a main service data packet for simplicity.

Thereafter, at least one type of the compression encoded mobile service data and the PSI/PSIP table data generated from the PSI/PSIP generator **130** for the mobile service are inputted to the mobile service multiplexer **150**. The mobile service multiplexer **150** encapsulates each of the inputted mobile service data and PSI/PSIP table data to MPEG-2 TS packet forms. Then, the MPEG-2 TS packets are multiplexed and outputted to the TS packet multiplexer **162**.

6

Herein, the data packet being outputted from the mobile service multiplexer **150** will be referred to as a mobile service data packet for simplicity. 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.

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/IEC13818-1 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 inverted, or only a portion of the synchronization byte may be inverted. 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. In this case, the mobile service multiplexer **150** multiplexes and outputs null data packets generated from the null packet generator **140** so that the output data rate can reach K Mbps. More specifically, in order to match the output data rate of the mobile service multiplexer **150** to a constant data rate, the null packet generator **140** generates null data packets, which are then outputted to the mobile service multiplexer **150**. 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 $\frac{1}{2}$, 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. Therefore, in order to match the data rate of the data that are finally outputted from the service multiplexer **100** to a constant data rate (e.g., 19.39 Mbps), an amount of null data packets corresponding to the amount of lacking data rate is generated from the null packet generator **140** and outputted to the mobile service multiplexer **150**.

Accordingly, the mobile service multiplexer **150** encapsulates each of the mobile service data and the PSI/PSIP table data that are being inputted to a MPEG-2 TS packet form. Then, the above-described TS packets are multiplexed with the null data packets and, then, outputted to the TS packet multiplexer **162**. Thereafter, the TS packet multiplexer **162** 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** and transmits the multiplexed data packets to the transmitter **200** at a data rate of 19.39 Mbps.

According to an embodiment of the present invention, the mobile service multiplexer **150** receives the null data packets. However, this is merely exemplary and does not limit the scope of the present invention. In other words, according to another embodiment of the present invention, the TS packet multiplexer **162** may receive the null data packets, so as to match the data rate of the finally outputted data to a constant data rate. Herein, the output path and multiplexing rule of the null data packet is controlled by the controller **110**. The controller **110** controls the multiplexing processed performed by the mobile service multiplexer **150**, the main service multiplexer **161** of the transport multiplexer **160**, and the TS packet multiplexer **162**, and also controls the null data packet generation of the null packet generator **140**. At this point, the transmitter **200** discards the null data packets transmitted from the service multiplexer **100** instead of transmitting the null data packets.

Further, in order to allow the transmitter **200** to discard the null data packets transmitted from the service multiplexer **100** instead of transmitting them, identification information for identifying the null data packet is required. Herein, the identification information may use values pre-decided in accordance with an agreement between the transmitting system and the receiving system. For example, the value of the synchronization byte within the header of the null data packet may be modified so as to be used as the identification information. Alternatively, a transport_error_indicator flag may also be used as the identification information.

In the description of the present invention, an example of using the transport_error_indicator flag as the identification information will be given to describe an embodiment of the present invention. In this case, the transport_error_indicator flag of the null data packet is set to '1', and the transport-

_error_indicator flag of the remaining data packets are reset to '0', so as to identify the null data packet. More specifically, when the null packet generator **140** generates the null data packets, if the transport_error_indicator flag from the header field of the null data packet is set to '1' and then transmitted, the null data packet may be identified and, therefore, be discarded. In the present invention, any type of identification information for identifying the null data packets may be used. Therefore, the scope of the present invention is not limited only to the examples set forth in the description of the present invention.

According to another embodiment of the present invention, a transmission parameter may be included in at least a portion of the null data packet, or at least one table or an operations and maintenance (OM) packet (or OMP) of the PSI/PSIP table for the mobile service. In this case, the transmitter **200** extracts the transmission parameter and outputs the extracted transmission parameter to the corresponding block and also transmits the extracted parameter to the receiving system if required. More specifically, a packet referred to as an OMP is defined for the purpose of operating and managing the transmitting system. For example, the OMP is configured in accordance with the MPEG-2 TS packet format, and the corresponding PID is given the value of 0x1FFA. The OMP is configured of a 4-byte header and a 184-byte payload. Herein, among the 184 bytes, the first byte corresponds to an OM_type field, which indicates the type of the OM packet.

In the present invention, the transmission parameter may be transmitted in the form of an OMP. And, in this case, among the values of the reserved fields within the OM_type field, a pre-arranged value is used, thereby indicating that the transmission parameter is being transmitted to the transmitter **200** in the form of an OMP. More specifically, the transmitter **200** may find (or identify) the OMP by referring to the PID. Also, by parsing the OM_type field within the OMP, the transmitter **200** can verify whether a transmission parameter is included after the OM_type field of the corresponding packet. The transmission parameter corresponds to supplemental data required for processing mobile service data from the transmitting system and the receiving system.

Herein, the transmission parameter may include data group information, region information within the data group, RS frame information, super frame information, burst information, turbo code information, and RS code information. The burst information may include burst size information, burst period information, and time information to next burst. The burst period signifies the period at which the burst transmitting the same mobile service is repeated. The data group includes a plurality of mobile service data packets, and a plurality of such data groups is gathered (or grouped) to form a burst. A burst section signifies the beginning of a current burst to the beginning of a next burst. Herein, the burst section is classified as a section that includes the data group (also referred to as a burst-on section), and a section that does not include the data group (also referred to as a burst-off section). A burst-on section is configured of a plurality of fields, wherein one field includes one data group.

The transmission parameter may also include information on how signals of a symbol domain are encoded in order to transmit the mobile service data, and multiplexing information on how the main service data and the mobile service data or various types of mobile service data are multiplexed. The information included in the transmission parameter is merely exemplary to facilitate the understanding of the present invention. And, the adding and deleting of the information included in the transmission parameter may be

easily modified and changed by anyone skilled in the art. Therefore, the present invention is not limited to the examples proposed in the description set forth herein. Furthermore, the transmission parameters may be provided from the service multiplexer 100 to the transmitter 200. Alternatively, the transmission parameters may also be set up by an internal controller (not shown) within the transmitter 200 or received from an external source.

Transmitter

FIG. 3 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 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, or a null data packet. For example, the demultiplexer 210 uses the PID within the received data packet so as to identify the main service data packet and the mobile service data packet. Then, the demultiplexer 210 uses a transport_error_indicator field to identify the null data packet. 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 a transmission parameter is included in the null data packet, then the transmission parameter is first extracted and outputted to the corresponding block. Thereafter, the null data packet is discarded.

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. The pre-processor 230 may also refer to the transmission parameter 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 areas within the data group.

Pre-Processor within Transmitter

FIG. 4 illustrates a block diagram showing an example of the pre-processor 230 according to the present invention. The pre-processor 230 includes a data randomizer 301, a RS frame encoder 302, a block processor 303, a group formatter 304, a data deinterleaver 305, a packet formatter 306. The data randomizer 301 within the above-described pre-processor 230 randomizes the mobile service data packet including the mobile service data that is inputted through the demultiplexer 210. Then, the data randomizer 301 outputs the randomized mobile service data packet to the RS frame encoder 302. At this point, since the data randomizer 301 performs the randomizing process on the mobile service data, the randomizing process that is to be performed by the data randomizer 251 of the post-processor 250 on the mobile service data may be omitted. The data randomizer 301 may also discard the synchronization byte within the mobile service data packet and perform the randomizing process. This is an option that may be chosen by the system designer. In the example given in the present invention, the random-

izing process is performed without discarding the synchronization byte within the mobile service data packet.

The RS frame encoder 302 groups a plurality of mobile the synchronization byte within the mobile service data packets that is randomized and inputted, so as to create a RS frame. Then, the RS frame encoder 302 performs at least one of an error correction encoding process and an error detection encoding process in RS frame units. 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 frame encoder 302 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.

More specifically, when the RS frame encoder 302 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. 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 RS encoding is one of forward error correction (FEC) methods. The FEC corresponds to a technique for compensating errors that occur during the transmission process. 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 frame encoder 302 refers to a pre-determined transmission parameter and/or the transmission parameter provided from the service multiplexer 100 so as to perform operations including RS frame configuration, RS encoding, CRC encoding, super frame configuration, and row permutation in super frame units.

Pre-Processor within RS Frame Encoder

FIG. 5(a) to FIG. 5(e) illustrate error correction encoding and error detection encoding processed according to an embodiment of the present invention. More specifically, the RS frame encoder 302 first divides the inputted mobile service data bytes to units of a predetermined length. The predetermined length is decided by the system designer. And, in the example of the present invention, the predetermined length is equal to 187 bytes, and, therefore, the 187-byte unit will be referred to as a packet for simplicity. For example, when the mobile service data that are being inputted, as shown in FIG. 5(a), correspond to a MPEG

11

transport packet stream configured of 188-byte units, the first synchronization byte is removed, as shown in FIG. 5(b), so as to configure a 187-byte unit. Herein, the synchronization byte is removed because each mobile service data packet has the same value.

Herein, the process of removing the synchronization byte may be performed during a randomizing process of the data randomizer 301 in an earlier process. In this case, the process of the removing the synchronization byte by the RS frame encoder 302 may be omitted. Moreover, when adding synchronization bytes from the receiving system, the process may be performed by the data derandomizer instead of the RS frame decoder. Therefore, if a removable fixed byte (e.g., synchronization byte) does not exist within the mobile service data packet that is being inputted to the RS frame encoder 302, or if the mobile service data that are being inputted are not configured in a packet format, the mobile service data that are being inputted are divided into 187-byte units, thereby configuring a packet for each 187-byte unit.

Subsequently, as shown in FIG. 5(c), N number of packets configured of 187 bytes is grouped to configure a RS frame. At this point, the RS frame is configured as a RS frame having the size of N(row)*187(column) bytes, in which 187-byte packets are sequentially inputted in a row direction. In order to simplify the description of the present invention, the RS frame configured as described above will also be referred to as a first RS frame. More specifically, only pure mobile service data are included in the first RS frame, which is the same as the structure configured of 187 N-byte rows. Thereafter, the mobile service data within the RS frame are divided into an equal size. Then, when the divided mobile service data are transmitted in the same order as the input order for configuring the RS frame, and when one or more errors have occurred at a particular point during the transmitting/receiving process, the errors are clustered (or gathered) within the RS frame as well. In this case, the receiving system uses a RS erasure decoding method when performing error correction decoding, thereby enhancing the error correction ability. At this point, the N number of columns within the N number of RS frame includes 187 bytes, as shown in FIG. 5(c).

In this case, a (Nc,Kc)-RS encoding process is performed on each column, so as to generate Nc-Kc(=P) number of parity bytes. Then, the newly generated P number of parity bytes is added after the very last byte of the corresponding column, thereby creating a column of (187+P) bytes. Herein, as shown in FIG. 5(c), Kc is equal to 187 (i.e., Kc=187), and Nc is equal to 187+P (i.e., Nc=187+P). For example, when P is equal to 48, (235,187)-RS encoding process is performed so as to create a column of 235 bytes. When such RS encoding process is performed on all N number of columns, as shown in FIG. 5(c), a RS frame having the size of N(row)*(187+P) (column) bytes may be created, as shown in FIG. 5(d). In order to simplify the description of the present invention, the RS frame having the RS parity inserted therein will be referred to as a second RS frame. More specifically, the second RS frame having the structure of (187+P) rows configured of N bytes may be configured.

As shown in FIG. 5(c) or FIG. 5(d), each row of the RS frame 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. 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 frame encoder 302 may perform CRC encoding on the mobile service data being RS encoded so as to create (or

12

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. 5(e) 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 1 below corresponds to an exemplary equation for generating a 2-byte CRC checksum for each row being configured of N number of bytes.

$$g(x)=x^{16}+x^{12}+x^5+1$$

Equation 1

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. In order to simplify the understanding of the present invention, the RS frame having the RS parity and CRC checksum added therein will hereinafter be referred to as a third RS frame. More specifically, the third RS frame corresponds to (187+P) number of rows each configured of (N+2) number of bytes. As described above, when the process of RS encoding and CRC encoding are completed, the (N*187)-byte RS frame is expanded to a (N+2)*(187+P)-byte RS frame. Furthermore, the RS frame that is expanded, as shown in FIG. 5(e), is inputted to the block processor 303.

As described above, the mobile service data encoded by the RS frame encoder 302 are inputted to the block processor 303. The block processor 303 then encodes the inputted mobile service data at a coding rate of G/H (wherein, G is smaller than H (i.e., G<H)) and then outputted to the group formatter 304. More specifically, the block processor 303 divides the mobile service data being inputted in byte units into bit units. Then, the G number of bits is encoded to H number of bits. Thereafter, the encoded bits are converted back to byte units and then outputted. For example, if 1 bit of the input data is coded to 2 bits and outputted, then G is equal to 1 and H is equal to 2 (i.e., G=1 and H=2). Alternatively, if 1 bit of the input data is coded to 4 bits and outputted, then G is equal to 1 and H is equal to 4 (i.e., G=1 and H=4). Hereinafter, the former coding rate will be referred to as a coding rate of 1/2 (1/2-rate coding), and the latter coding rate will be referred to as a coding rate of 1/4 (1/4-rate coding), for simplicity.

Herein, when using the 1/4 coding rate, the coding efficiency is greater than when using the 1/2 coding rate, and may, therefore, provide greater and enhanced error correction ability. For such reason, when it is assumed that the data encoded at a 1/4 coding rate in the group formatter 304, which is located near the end portion of the system, are allocated to an area in which the receiving performance may be deteriorated, and that the data encoded at a 1/2 coding rate are allocated to an area having excellent receiving performance, the difference in performance may be reduced. At this point, the block processor 303 may also receive signaling information including transmission parameters. Herein, the signaling information may also be processed with either 1/2-rate coding or 1/4-rate coding as in the step of processing

mobile service data. Thereafter, the signaling information is also considered the same as the mobile service data and processed accordingly.

Meanwhile, the group formatter inserts mobile service data that are outputted from the block processor 303 in corresponding areas within a data group, which is configured in accordance with a pre-defined rule. Also, with respect to the data deinterleaving process, each place holder or known data (or known data place holders) are also inserted in corresponding areas within the data group. At this point, the data group may be divided into at least one hierarchical area. Herein, the type of mobile service data being inserted in each area may vary depending upon the characteristics of each hierarchical area. Additionally, each area may, for example, be divided based upon the receiving performance within the data group. Furthermore, one data group may be configured to include a set of field synchronization data.

In an example given in the present invention, a data group is divided into A, B, and C regions in a data configuration prior to data deinterleaving. At this point, the group formatter 304 allocates the mobile service data, which are inputted after being RS encoded and block encoded, to each of the corresponding regions by referring to the transmission parameter. FIG. 6A illustrates an alignment of data after being data interleaved and identified, and FIG. 6B illustrates an alignment of data before being data interleaved and identified. More specifically, a data structure identical to that shown in FIG. 6A is transmitted to a receiving system. Also, the data group configured to have the same structure as the data structure shown in FIG. 6A is inputted to the data deinterleaver 305.

As described above, FIG. 6A illustrates a data structure prior to data deinterleaving that is divided into 3 regions, such as region A, region B, and region C. Also, in the present invention, each of the regions A to C is further divided into a plurality of regions. Referring to FIG. 6A, region A is divided into 5 regions (A1 to A5), region B is divided into 2 regions (B1 and B2), and region C is divided into 3 regions (C1 to C3). Herein, regions A to C are identified as regions having similar receiving performances within the data group. Herein, the type of mobile service data, which are inputted, may also vary depending upon the characteristic of each region.

In the example of the present invention, the data structure is divided into regions A to C based upon the level of interference of the 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, 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 (e.g., region A). 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 (e.g., region B and region C).

Hereinafter, examples of allocating data to region A (A1 to A5), region B (B1 and B2), and region C (C1 to C3) will now be described in detail with reference to FIG. 6A. The data group size, the number of hierarchically divided regions

within the data group and the size of each region, and the number of mobile service data bytes that can be inserted in each hierarchically divided region of FIG. 6A are merely examples given to facilitate the understanding of the present invention. Herein, the group formatter 304 creates a data group including places in which field synchronization data bytes are to be inserted, so as to create the data group that will hereinafter be described in detail.

More specifically, region A is a region within the data group in which a long known data sequence may be periodically inserted, and in which includes regions wherein the main service data are not mixed (e.g., A1 to A5). Also, region A includes a region (e.g., A1) located between a field synchronization region and the region in which the first known data sequence is to be inserted. The field synchronization region has the length of one segment (i.e., 832 symbols) existing in an ATSC system.

For example, referring to FIG. 6A, 2428 bytes of the mobile service data may be inserted in region A1, 2580 bytes may be inserted in region A2, 2772 bytes may be inserted in region A3, 2472 bytes may be inserted in region A4, and 2772 bytes may be inserted in region A5. Herein, trellis initialization data or known data, MPEG header, and RS parity are not included in the mobile service data. As described above, when region A includes a known data sequence at both ends, the receiving system uses channel information that can obtain known data or field synchronization data, so as to perform equalization, thereby providing enforced equalization performance.

Also, region B includes a region located within 8 segments at the beginning of a field synchronization region within the data group (chronologically placed before region A1) (e.g., region B1), and a region located within 8 segments behind the very last known data sequence which is inserted in the data group (e.g., region B2). For example, 930 bytes of the mobile service data may be inserted in the region B1, and 1350 bytes may be inserted in region B2. Similarly, trellis initialization data or known data, MPEG header, and RS parity are not included in the mobile service data. In case of region B, the receiving system may perform equalization by using channel information obtained from the field synchronization region. Alternatively, the receiving system may also perform equalization by using channel information that may be obtained from the last known data sequence, thereby enabling the system to respond to the channel changes.

Region C includes a region located within 30 segments including and preceding the 9th segment of the field synchronization region (chronologically located before region A) (e.g., region C1), a region located within 12 segments including and following the 9th segment of the very last known data sequence within the data group (chronologically located after region A) (e.g., region C2), and a region located in 32 segments after the region C2 (e.g., region C3). For example, 1272 bytes of the mobile service data may be inserted in the region C1, 1560 bytes may be inserted in region C2, and 1312 bytes may be inserted in region C3. Similarly, trellis initialization data or known data, MPEG header, and RS parity are not included in the mobile service data. Herein, region C (e.g., region C1) is located chronologically earlier than (or before) region A.

Since region C (e.g., region C1) is located further apart from the field synchronization region which corresponds to the closest known data region, the receiving system may use the channel information obtained from the field synchronization data when performing channel equalization. Alternatively, the receiving system may also use the most recent channel information of a previous data group. Furthermore,

in region C (e.g., region C2 and region C3) located before region A, the receiving system may use the channel information obtained from the last known data sequence to perform equalization. However, when the channels are subject to fast and frequent changes, the equalization may not be performed perfectly. Therefore, the equalization performance of region C may be deteriorated as compared to that of region B.

When it is assumed that the data group is allocated with a plurality of hierarchically divided regions, as described above, the block processor 303 may encode the mobile service data, which are to be inserted to each region based upon the characteristic of each hierarchical region, at a different coding rate. For example, the block processor 303 may encode the mobile service data, which are to be inserted in regions A1 to A5 of region A, at a coding rate of $\frac{1}{2}$. Then, the group formatter 304 may insert the $\frac{1}{2}$ -rate encoded mobile service data to regions A1 to A5.

The block processor 303 may encode the mobile service data, which are to be inserted in regions B1 and B2 of region B, at a coding rate of $\frac{1}{4}$ having higher error correction ability as compared to the $\frac{1}{2}$ -coding rate. Then, the group formatter 304 inserts the $\frac{1}{2}$ -rate coded mobile service data in region B1 and region B2. Furthermore, the block processor 303 may encode the mobile service data, which are to be inserted in regions C1 to C3 of region C, at a coding rate of $\frac{1}{4}$ or a coding rate having higher error correction ability than the $\frac{1}{2}$ -coding rate. Then, the group formatter 304 may either insert the encoded mobile service data to regions C1 to C3, as described above, or leave the data in a reserved region for future usage.

In addition, the group formatter 304 also inserts supplemental data, such as signaling information that notifies the overall transmission information, other than the mobile service data in the data group. Also, apart from the encoded mobile service data outputted from the block processor 303, the group formatter 304 also inserts MPEG header place holders, non-systematic RS parity place holders, main service data place holders, which are related to data deinterleaving in a later process, as shown in FIG. 6A. 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 and C based upon the input of the data deinterleaver, as shown in FIG. 6A. 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.

Furthermore, the group formatter 304 either inserts known data generated in accordance with a pre-determined method or inserts known data place holders for inserting the known data in a later process. Additionally, place holders for initializing the trellis encoding module 256 are also inserted in the corresponding regions. For example, the initialization data place holders may be inserted in the beginning of the known data sequence. Herein, the size of the mobile service data that can be inserted in a data group may vary in accordance with the sizes of the trellis initialization place holders or known data (or known data place holders), MPEG header place holders, and RS parity place holders.

The output of the group formatter 304 is inputted to the data deinterleaver 305. And, the data deinterleaver 305 deinterleaves data by performing an inverse process of the data interleaves on the data and place holders within the data group, which are then outputted to the packet formatter 306. More specifically, when the data and place holders within the data group configured, as shown in FIG. 6A, are deinterleaved by the data deinterleaver 305, the data group being

outputted to the packet formatter 306 is configured to have the structure shown in FIG. 69.

The packet formatter 306 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 306 groups the remaining portion and replaces the 4-byte MPEG header place holder with an MPEG header having a null packet PID (or an unused PID from the main service data packet). Also, when the group formatter 304 inserts known data place holders, the packet formatter 306 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 306 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.

The packet multiplexer 240 multiplexes the mobile service data packet outputted from the pre-processor 230 and the main service data packet outputted from the packet jitter mitigator 220 in accordance with a pre-defined multiplexing method. Then, the packet multiplexer 240 outputs the multiplexed data packets to the data randomizer 251 of the post-processor 250. Herein, the multiplexing method may vary in accordance with various variables of the system design. One of the multiplexing methods of the packet formatter 240 consists of providing a burst section along a time axis, and, then, transmitting a plurality of data groups during a burst-on section within the burst section, and transmitting only the main service data during the burst-off section within the burst section. Herein, the burst section indicates the section starting from the beginning of the current burst until the beginning of the next burst.

At this point, the main service data may be transmitted during the burst-on section. The packet multiplexer 240 refers to the transmission parameter, such as information on the burst size or the burst period, so as to be informed of the number of data groups and the period of the data groups included in a single burst. Herein, the mobile service data and the main service data may co-exist in the burst-on section, and only the main service data may exist in the burst-off section. Therefore, a main data service section transmitting the main service data may exist in both burst-on and burst-off sections. At this point, the main data service section within the burst-on section and the number of main data service packets included in the burst-off section may either be different from one another or be the same.

When the mobile service data are transmitted in a burst structure, in the receiving system receiving only the mobile service data turns the power on only during the burst section, thereby receiving the corresponding data. Alternatively, in the section transmitting only the main service data, the power is turned off so that the main service data are not received in this section. Thus, the power consumption of the receiving system may be reduced.

Detailed Embodiments of the RS Frame Structure and Packet Multiplexing

Hereinafter, detailed embodiments of the pre-processor 230 and the packet multiplexer 240 will now be described. According to an embodiment of the present invention, the N value corresponding to the length of a row, which is included in the RS frame that is configured by the RS frame encoder 302, is set to 538. Accordingly, the RS frame encoder 302 receives 538 transport stream (TS) packets so as to configure a first RS frame having the size of 538*187 bytes. There-

after, as described above, the first RS frame is processed with a $(235,187)$ -RS encoding process so as to configure a second RS frame having the size of 538×235 bytes. Finally, the second RS frame is processed with generating a 16-bit checksum so as to configure a third RS frame having the sizes of 540×235 .

Meanwhile, as shown in FIG. 6A, the sum of the number of bytes of regions A1 to A5 of region A, in which $\frac{1}{2}$ -rate encoded mobile service data are to be inserted, among the plurality of regions within the data group is equal to 13024 bytes $(=2428+2580+2772+2472+2772 \text{ bytes})$. Herein, the number of byte prior to performing the $\frac{1}{2}$ -rate encoding process is equal to 6512 $(=13024/2)$. On the other hand, the sum of the number of bytes of regions B1 and B2 of region B, in which $\frac{1}{2}$ -rate encoded mobile service data are to be inserted, among the plurality of regions within the data group is equal to 2280 bytes $(=930+1350 \text{ bytes})$. Herein, the number of byte prior to performing the $\frac{1}{2}$ -rate encoding process is equal to 570 $(=2280/4)$.

In other words, when 7082 bytes of mobile service data are inputted to the block processor 303, 6512 byte are expanded to 13024 bytes by being $\frac{1}{2}$ -rate encoded, and 570 bytes are expanded to 2280 bytes by being $\frac{1}{2}$ -rate encoded. Thereafter, the block processor 303 inserts the mobile service data expanded to 13024 bytes in regions A1 to A5 of region A and, also, inserts the mobile service data expanded to 2280 bytes in regions B1 and B2 of region B. Herein, the 7082 bytes of mobile service data being inputted to the block processor 303 may be divided into an output of the RS frame encoder 302 and signaling information. In the present invention, among the 7082 bytes of mobile service data, 7050 bytes correspond to the output of the RS frame encoder 302, and the remaining 32 bytes correspond to the signaling information data. Then, $\frac{1}{2}$ -rate encoding or $\frac{1}{4}$ -rate encoding is performed on the corresponding data bytes.

Meanwhile, a RS frame being processed with RS encoding and CRC encoding from the RS frame encoder 302 is configured of 540×235 bytes, in other words, 126900 bytes. The 126900 bytes are divided by 7050-byte units along the time axis, so as to produce 18 7050-byte units. Thereafter, a 32-byte unit of signaling information data is added to the 7050-byte unit mobile service data being outputted from the RS frame encoder 302. Subsequently, the RS frame encoder 302 performs $\frac{1}{2}$ -rate encoding or $\frac{1}{4}$ -rate encoding on the corresponding data bytes, which are then outputted to the group formatter 304. Accordingly, the group formatter 304 inserts the $\frac{1}{2}$ -rate encoded data in region A and the $\frac{1}{4}$ -rate encoded data in region B.

The process of deciding an N value that is required for configuring the RS frame from the RS frame encoder 302 will now be described in detail. More specifically, the size of the final RS frame (i.e., the third RS frame), which is RS encoded and CRC encoded from the RS frame encoder 302, which corresponds to $(N+2) \times 235$ bytes should be allocated to X number of groups, wherein X is an integer. Herein, in a single data group, 7050 data bytes prior to being encoded are allocated. Therefore, if the $(N+2) \times 235$ bytes are set to be the exact multiple of 7050 $(=30 \times 235)$, the output data of the RS frame encoder 302 may be efficiently allocated to the data group. According to an embodiment of the present invention, the value of N is decided so that $(N+2)$ becomes a multiple of 30. For example, in the present invention, N is equal to 538, and $(N+2) (=540)$ divided by 30 is equal to 18. This indicates that the mobile service data within one RS frame are processed with either $\frac{1}{2}$ -rate encoding or $\frac{1}{4}$ -rate encoding. The encoded mobile service data are then allocated to 18 data groups.

FIG. 7 illustrates a process of dividing the RS frame according to the present invention. More specifically, the RS frame having the size of $(N+2) \times 235$ is divided into 30×235 byte blocks. Then, the divided blocks are mapped to a single group. In other words, the data of a block having the size of 30×235 bytes are processed with one of a $\frac{1}{2}$ -rate encoding process and a $\frac{1}{4}$ -rate encoding process and are, then, inserted in a data group. Thereafter, the data group having corresponding data and place holders inserted in each hierarchical region divided by the group formatter 304 passes through the data deinterleaver 305 and the packet formatter 306 so as to be inputted to the packet multiplexer 240.

FIG. 8 illustrates exemplary operations of a packet multiplexer for transmitting the data group according to the present invention. More specifically, the packet multiplexer 240 multiplexes a field including a data group, in which the mobile service data and main service data are mixed with one another, and a field including only the main service data. Thereafter, the packet multiplexer 240 outputs the multiplexed fields to the data randomizer 251. At this point, in order to transmit the RS frame having the size of 540×235 bytes, 18 data groups should be transmitted. Herein, each data group includes field synchronization data, as shown in FIG. 6A. Therefore, the 18 data groups are transmitted during 18 field sections, and the section during which the 18 data groups are being transmitted corresponds to the burst-on section.

In each field within the burst-on section, a data group including field synchronization data is multiplexed with main service data, which are then outputted. For example, in the embodiment of the present invention, in each field within the burst-on section, a data group having the size of 118 segments is multiplexed with a set of main service data having the size of 194 segments. Referring to FIG. 8, during the burst-on section (i.e., during the 18 field sections), a field including 18 data groups is transmitted. Then, during the burst-off section that follows (i.e., during the 12 field sections), a field consisting only of the main service data is transmitted. Subsequently, during a subsequent burst-on section, 18 fields including 18 data groups are transmitted. And, during the following burst-off section, 12 fields consisting only of the main service data are transmitted.

Furthermore, in the present invention, the same type of data service may be provided in the first burst-on section including the first 18 data groups and in the second burst-on section including the next 18 data groups. Alternatively, different types of data service may be provided in each burst-on section. For example, when it is assumed that different data service types are provided to each of the first burst-on section and the second burst-on section, and that the receiving system wishes to receive only one type of data service, the receiving system turns the power on only during the corresponding burst-on section including the desired data service type so as to receive the corresponding 18 data fields. Then, the receiving system turns the power off during the remaining 42 field sections so as to prevent other data service types from being received. Thus, the amount of power consumption of the receiving system may be reduced. In addition, the receiving system according to the present invention is advantageous in that one RS frame may be configured from the 18 data groups that are received during a single burst-on section.

According to the present invention, the number of data groups included in a burst-on section may vary based upon the size of the RS frame, and the size of the RS frame varies in accordance with the value N. More specifically, by adjusting the value N, the number of data groups within the

burst section may be adjusted. Herein, in an example of the present invention, the (235,187)-RS encoding process adjusts the value N during a fixed state. Furthermore, the size of the mobile service data that can be inserted in the data group may vary based upon the sizes of the trellis initialization data or known data, the MPEG header, and the RS parity, which are inserted in the corresponding data group.

Meanwhile, since a data group including mobile service data in-between the data bytes of the main service data 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 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. The standard for repositioning the audio data packets in the main service data performed by the packet jitter mitigator 220 will now be described. Herein, it is assumed that the packet jitter mitigator 220 knows the same multiplexing information as that of the packet multiplexer 240, which is placed further behind the packet jitter mitigator 220.

Firstly, if one audio data packet exists in the main service data section (e.g., the main service data section positioned between two data groups) within the burst-on section, the audio data packet is positioned at the very beginning of the main service data section. Alternatively, if two audio data packets exist in the corresponding data section, one audio data packet is positioned at the very beginning and the other audio data packet is positioned at the very end of the main service data section. Further, if more than three audio data packets exist, one audio data packet is positioned at the very beginning of the main service data section, another is positioned at the very end of the main service data section, and the remaining audio data packets are equally positioned between the first and last audio data packets. Secondly, during the main service data section placed immediately before the beginning of a burst-on section (i.e., during a burst-off section), the audio data packet is placed at the very end of the corresponding section.

Thirdly, during a main service data section within the burst-off section after the burst-on section, the audio data packet is positioned at the very end of the main service data section. Finally, the data packets other than audio data

packets are positioned in accordance with the inputted order in vacant spaces (i.e., spaces that are not designated for the audio data packets). Meanwhile, 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 randomize only a portion of the data packet. For example, if it is assumed that a randomizing process has already been performed in advance on the mobile service data packet by the pre-processor 230, the data randomizer 251 deletes the synchronization byte from the 4-byte MPEG header included in the mobile service data packet and, then, performs the randomizing process only on the remaining 3 data bytes of the MPEG header. Thereafter, the randomized data bytes are outputted to the RS encoder/non-systematic RS encoder 252. More specifically, the randomizing process is not performed on the remaining portion of the mobile service data excluding the MPEG header. In other words, the remaining portion of the mobile service data packet is directly outputted to the RS encoder/non-systematic RS encoder 252 without being randomized. Also, the data randomizer 251 may or may not perform a randomizing process on the known data (or known data place holders) and the initialization data place holders included in 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.

21

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 place holder and not to the actual known data. Herein, the initialization data place holder has been included in the data by the group formatter within the pre-processor **230** in an earlier process. Therefore, the process of generating initialization data and replacing the initialization data place holder of the corresponding memory with the generated initialization data are required to be performed immediately before the inputted known data sequence is trellis-encoded.

Additionally, a value of the trellis memory initialization data is decided and generated based upon a memory status of the trellis encoding module **256**. Further, due to the newly replaced initialization data, a process of newly calculating the RS parity and replacing the RS parity, which is outputted from the data interleaver **253**, with the newly calculated RS parity is required. Therefore, the non-systematic RS encoder **255** receives the mobile service data packet including the initialization data place holders, which are to be replaced with the actual initialization data, from the data interleaver **253** and also receives the initialization data from the trellis encoding module **256**.

Among the inputted mobile service data packet, the initialization data place holders are replaced with the initialization data, 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 place holders 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**.

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

22

modulated data are transmitted to each receiving system through the radio frequency (RF) up-converter **273**.
Block Processor

FIG. 9 illustrates a block diagram showing a structure of a block processor according to the present invention. Herein, the block processor includes a byte-bit converter **401**, a symbol encoder **402**, a symbol interleaver **403**, and a symbol-byte converter **404**. The byte-bit converter **401** divides the mobile service data bytes that are inputted from the RS frame encoder **112** into bits, which are then outputted to the symbol encoder **402**. The byte-bit converter **401** may also receive signaling information including transmission parameters. The signaling information data bytes are also divided into bits so as to be outputted to the symbol encoder **402**. Herein, the signaling information including transmission parameters may be processed with the same data processing step as that of the mobile service data. More specifically, the signaling information may be inputted to the block processor **303** by passing through the data randomizer **301** and the RS frame encoder **302**. Alternatively, the signaling information may also be directly outputted to the block processor **303** without passing through the data randomizer **301** and the RS frame encoder **302**.

The symbol encoder **402** corresponds to a G/H-rate encoder encoding the inputted data from G bits to H bits and outputting the data encoded at the coding rate of G/H. According to the embodiment of the present invention, it is assumed that the symbol encoder **402** performs either a coding rate of $\frac{1}{2}$ (also referred to as a $\frac{1}{2}$ -rate encoding process) or an encoding process at a coding rate of $\frac{1}{4}$ (also referred to as a $\frac{1}{4}$ -rate encoding process). The symbol encoder **402** performs one of $\frac{1}{2}$ -rate encoding and $\frac{1}{4}$ -rate encoding on the inputted mobile service data and signaling information. Thereafter, the signaling information is also recognized as the mobile service data and processed accordingly.

In case of performing the $\frac{1}{2}$ -rate coding process, the symbol encoder **402** receives 1 bit and encodes the received 1 bit to 2 bits (i.e., 1 symbol). Then, the symbol encoder **402** outputs the processed 2 bits (or 1 symbol). On the other hand, in case of performing the $\frac{1}{4}$ -rate encoding process, the symbol encoder **402** receives 1 bit and encodes the received 1 bit to 4 bits (i.e., 2 symbols). Then, the symbol encoder **402** outputs the processed 4 bits (or 2 symbols).

FIG. 10 illustrates a detailed block diagram of the symbol encoder **402** shown in FIG. 9. The symbol encoder **402** includes two delay units **501** and **503** and three adders **502**, **504**, and **505**. Herein, the symbol encoder **402** encodes an input data bit U and outputs the coded bit U to 4 bits (u0 to u4). At this point, the data bit U is directly outputted as uppermost bit u0 and simultaneously encoded as lower bit u1u2u3 and then outputted. More specifically, the input data bit U is directly outputted as the uppermost bit u0 and simultaneously outputted to the first and third adders **502** and **505**. The first adder **502** adds the input data bit U and the output bit of the first delay unit **501** and, then, outputs the added bit to the second delay unit **503**. Then, the data bit delayed by a pre-determined time (e.g., by 1 clock) in the second delay unit **503** is outputted as lower bit u1 and simultaneously fed-back to the first delay unit **501**. The first delay unit **501** delays the data bit fed-back from the second delay unit **503** by a pre-determined time (e.g., by 1 clock). Then, the first delay unit **501** outputs the delayed data bit to the first adder **502** and the second adder **504**. The second adder **504** adds the data bits outputted from the first and second delay units **501** and **503** as a lower bit u2. The third

23

adder 505 adds the input data bit U and the output of the second delay unit 503 and outputs the added data bit as a lower bit u3.

At this point, if the input data bit U corresponds to data encoded at a 1/2-coding rate, the symbol encoder 402 configures a symbol with u1u0 bits from the 4 output bits u0u1u2u3. Then, the symbol encoder 402 outputs the newly configured symbol. Alternatively, if the input data bit U corresponds to data encoded at a 1/2-coding rate, the symbol encoder 402 configures and outputs a symbol with bits u1u0 and, then, configures and outputs another symbol with bits u2u3. According to another embodiment of the present invention, if the input data bit U corresponds to data encoded at a 1/2-coding rate, the symbol encoder 402 may also configure and output a symbol with bits u1u0, and then repeat the process once again and output the corresponding bits. According to yet another embodiment of the present invention, the symbol encoder outputs all four output bits U u0u1u2u3. Then, when using the 1/2-coding rate, the symbol interleaver 403 located behind the symbol encoder 402 selects only the symbol configured of bits u1u0 from the four output bits u0u1u2u3. Alternatively, when using the 1/2-coding rate, the symbol interleaver 403 may select the symbol configured of bits u1u0 and then select another symbol configured of bits u2u3. According to another embodiment, when using the 1/4-coding rate, the symbol interleaver 403 may repeatedly select the symbol configured of bits u1u0.

The output of the symbol encoder 402 is inputted to the symbol interleaver 403. Then, the symbol interleaver 403 performs block interleaving in symbol units on the data outputted from the symbol encoder 402. Any interleaver performing structural rearrangement (or realignment) may be applied as the symbol interleaver 403 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. 11 illustrates a symbol interleaver according to an embodiment of the present invention. Herein, the symbol interleaver according to the embodiment of the present invention corresponds to a variable length symbol interleaver that may be applied even when a plurality of lengths is provided for the symbol, so that its order may be rearranged. Particularly, FIG. 11 illustrates an example of the symbol interleaver when K=6 and L=8. Herein, K indicates a number of symbols that are outputted for symbol interleaving from the symbol encoder 402. And, L represents a number of symbols that are actually interleaved by the symbol interleaver 403.

In the present invention, the symbol interleaver 403 should satisfy the conditions of $L=2^n$ (wherein n is an integer) and of $L \geq K$. If there is a difference in value between K and L, (L-K) number of null (or dummy) symbols is added, thereby creating an interleaving pattern. Therefore, K becomes a block size of the actual symbols that are inputted to the symbol interleaver 403 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 403. The example of what is described above is illustrated in FIG. 11.

More specifically, FIG. 11(a) to FIG. 11(c) illustrate a variable length interleaving process of a symbol interleaver shown in FIG. 9. The number of symbols outputted from the symbol encoder 402 in order to be interleaved is equal to 6 (i.e., K=6). In other words, 6 symbols are outputted from the symbol encoder 402 in order to be interleaved. And, the

24

actual interleaving unit (L) is equal to 8 symbols. Therefore, as shown in FIG. 11(a), 2 symbols are added to the null (or dummy) symbol, thereby creating the interleaving pattern. Equation 2 shown below describes the process of sequentially receiving K number of symbols, the order of which is to be rearranged, and obtaining an L value satisfying the conditions of $L=2^n$ (wherein n is an integer) and of $L \geq K$, thereby creating the interleaving so as to realign (or rearrange) the symbol order.

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

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

Herein, $L \geq K$, $L=2^n$, and n and S are integers. Referring to FIG. 11, it is assumed that S is equal to 89, and that L is equal to 8, and FIG. 11 illustrates the created interleaving pattern and an example of the interleaving process. As shown in FIG. 11(b), the order of K number of input symbols and (L-K) number of null symbols is rearranged by using the above-mentioned Equation 2. Then, as shown in FIG. 11(c), the null byte places are removed, so as to rearrange the order, by using Equation 3 shown below. Thereafter, the symbol that is interleaved by the rearranged order is then outputted to the symbol-byte converter.

$$\text{if } P(i) > K-1, \text{ then } P(i) \text{ place is removed and rearranged} \quad \text{Equation 3}$$

Subsequently, the symbol-byte converter 404 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 group formatter 304.

FIG. 12A illustrates a block diagram showing the structure of a block processor according to another embodiment of the present invention. Herein, the block processor includes an interleaving unit 610 and a block formatter 620. The interleaving unit 610 may include a byte-symbol converter 611, a symbol-byte converter 612, a symbol interleaver 613, and a symbol-byte converter 614. Herein, the symbol interleaver 613 may also be referred to as a block interleaver.

The byte-symbol converter 611 of the interleaving unit 610 converts the mobile service data X outputted in byte units from the RS frame encoder 302 to symbol units. Then, the byte-symbol converter 611 outputs the converted mobile service data symbols to the symbol-byte converter 612 and the symbol interleaver 613. More specifically, the byte-symbol converter 611 converts each 2 bits of the inputted mobile service data byte (=8 bits) to 1 symbol and outputs the converted symbols. This is because the input data of the trellis encoding module 256 consist of symbol units configured of 2 bits. The relationship between the block processor 303 and the trellis encoding module 256 will be described in detail in a later process. At this point, the byte-symbol converter 611 may also receive signaling information including transmission parameters. Furthermore, the signaling information bytes may also be divided into symbol units and then outputted to the symbol-byte converter 612 and the symbol interleaver 613.

The symbol-byte converter 612 groups 4 symbols outputted from the byte-symbol converter 611 so as to configure a byte. Thereafter, the converted data bytes are outputted to the block formatter 620. Herein, each of the symbol-byte converter 612 and the byte-symbol converter 611 respectively performs an inverse process on one another. Therefore, the yield of these two blocks is offset. Accordingly, as shown in FIG. 12B, the input data X bypass the byte-symbol converter 611 and the symbol-byte converter 612 and are

25

directly inputted to the block formatter 620. More specifically, the interleaving unit 610 of FIG. 12B has a structure equivalent to that of the interleaving unit shown in FIG. 12A. Therefore, the same reference numerals will be used in FIG. 12A and FIG. 12B.

The symbol interleaver 613 performs block interleaving in symbol units on the data that are outputted from the byte-symbol converter 611. Subsequently, the symbol interleaver 613 outputs the interleaved data to the symbol-byte converter 614. Herein, any type of interleaver that can rearrange the structural order may be used as the symbol interleaver 613 of the present invention. In the example given in the present invention, a variable length interleaver that may be applied for symbols having a wide range of lengths, the order of which is to be rearranged. For example, the symbol interleaver of FIG. 11 may also be used in the block processor shown in FIG. 12A and FIG. 12B.

The symbol-byte converter 614 outputs the symbols having the rearranging of the symbol order completed, in accordance with the rearranged order. Thereafter, the symbols are grouped to be configured in byte units, which are then outputted to the block formatter 620. More specifically, the symbol-byte converter 614 groups 4 symbols outputted from the symbol interleaver 613 so as to configure a data byte. As shown in FIG. 13, the block formatter 620 performs the process of aligning the output of each symbol-byte converter 612 and 614 within the block in accordance with a set standard. Herein, the block formatter 620 operates in association with the trellis encoding module 256.

More specifically, the block formatter 620 decides the output order of the mobile service data outputted from each symbol-byte converter 612 and 614 while taking into consideration the place (or order) of the data excluding the mobile service data that are being inputted, wherein the mobile service data include main service data, known data, RS parity data, and MPEG header data.

According to the embodiment of the present invention, the trellis encoding module 256 is provided with trellis encoders. FIG. 14 illustrates a block diagram showing the trellis encoding module 256 according to the present invention. In the example shown in FIG. 14, 12 identical trellis encoders are combined to the interleaver in order to disperse noise. Herein, each trellis encoder may be provided with a pre-coder.

FIG. 15A illustrates the block processor 303 being concatenated with the trellis encoding module 256. In the transmitting system, a plurality of blocks actually exists between the pre-processor 230 including the block processor 303 and the trellis encoding module 256, as shown in FIG. 3. Conversely, the receiving system considers the pre-processor 230 to be concatenated with the trellis encoding module 256, thereby performing the decoding process accordingly. However, the data excluding the mobile service data that are being inputted to the trellis encoding module 256, wherein the mobile service data include main service data, known data, RS parity data, and MPEG header data, correspond to data that are added to the blocks existing between the block processor 303 and the trellis encoding module 256. FIG. 15B illustrates an example of a data processor 650 being positioned between the block processor 303 and the trellis encoding module 256, while taking the above-described instance into consideration.

Herein, when the interleaving unit 610 of the block processor 303 performs a 1/2-rate encoding process, the interleaving unit 610 may be configured as shown in FIG. 12A (or FIG. 12B). Referring to FIG. 3, for example, the data processor 650 may include a group formatter 304, a

26

data deinterleaver 305, a packet formatter 306, a packet multiplexer 240, and a post-processor 250, wherein the post-processor 250 includes a data randomizer 251, a RS encoder/non-systematic RS encoder 252, a data interleaver 253, a parity replacer 254, and a non-systematic RS encoder 255.

At this point, the trellis encoding module 256 symbolizes the data that are being inputted so as to divide the symbolized data and to send the divided data to each trellis encoder in accordance with a pre-defined method. Herein, one byte is converted into 4 symbols, each being configured of 2 bits. Also, the symbols created from the single data byte are all transmitted to the same trellis encoder. Accordingly, each trellis encoder pre-codes an upper bit of the input symbol, which is then outputted as the uppermost output bit C2. Alternatively, each trellis encoder trellis-encodes a lower bit of the input symbol, which is then outputted as two output bits C1 and C0. The block formatter 620 is controlled so that the data byte outputted from each symbol-byte converter can be transmitted to different trellis encoders.

Hereinafter, the operation of the block formatter 620 will now be described in detail with reference to FIG. 9 to FIG. 12. Referring to FIG. 12A, for example, the data byte outputted from the symbol-byte converter 612 and the data byte outputted from the symbol-byte converter 614 are inputted to different trellis encoders of the trellis encoding module 256 in accordance with the control of the block formatter 620. Hereinafter, the data byte outputted from the symbol-byte converter 612 will be referred to as X, and the data byte outputted from the symbol-byte converter 614 will be referred to as Y, for simplicity. Referring to FIG. 13(a), each number (i.e., 0 to 11) indicates the first to twelfth trellis encoders of the trellis encoding module 256, respectively.

In addition, the output order of both symbol-byte converters are arranged (or aligned) so that the data bytes outputted from the symbol-byte converter 612 are respectively inputted to the 0th to 5th trellis encoders (0 to 5) of the trellis encoding module 256, and that the data bytes outputted from the symbol-byte converter 614 are respectively inputted to the 6th to 11th trellis encoders (6 to 11) of the trellis encoding module 256. Herein, the trellis encoders having the data bytes outputted from the symbol-byte converter 612 allocated therein are merely examples given to simplify the understanding of the present invention. Furthermore, according to an embodiment of the present invention, and assuming that the input data of the block processor 303 correspond to a block configured of 12 bytes, the symbol-byte converter 612 outputs 12 data bytes from X0 to X11, and the symbol-byte converter 614 outputs 12 data bytes from Y0 to Y11.

FIG. 13(b) illustrates an example of data being inputted to the trellis encoding module 256. Particularly, FIG. 13(b) illustrates an example of not only the mobile service data but also the main service data and RS parity data being inputted to the trellis encoding module 256, so as to be distributed to each trellis encoder. More specifically, the mobile service data outputted from the block processor 303 pass through the group formatter 304, from which the mobile service data are mixed with the main service data and RS parity data and then outputted, as shown in FIG. 13(a). Accordingly, each data byte is respectively inputted to the 12 trellis encoders in accordance with the positions (or places) within the data group after being data-interleaved.

Herein, when the output data bytes X and Y of the symbol-byte converters 612 and 614 are allocated to each

respective trellis encoder, the input of each trellis encoder may be configured as shown in FIG. 13(b). More specifically, referring to FIG. 13(b), the six mobile service data bytes (X0 to X5) outputted from the symbol-byte converter 612 are sequentially allocated (or distributed) to the first to sixth trellis encoders (0 to 5) of the trellis encoding module 256. Also, the 2 mobile service data bytes Y0 and Y1 outputted from the symbol-byte converter 614 are sequentially allocated to the 7th and 8th trellis encoders (6 and 7) of the trellis encoding module 256. Thereafter, among the 5 main service data bytes, 4 data bytes are sequentially allocated to the 9th and 12th trellis encoders (8 to 11) of the trellis encoding module 256. Finally, the remaining 1 byte of the main service data byte is allocated once again to the first trellis encoder (0).

It is assumed that the mobile service data, the main service data, and the RS parity data are allocated to each trellis encoder, as shown in FIG. 13(b). It is also assumed that, as described above, the input of the block processor 303 is configured of 12 bytes, and that 12 bytes from X0 to X11 are outputted from the symbol-byte converter 612, and that 12 bytes from Y0 to Y11 are outputted from the symbol-byte converter 614. In this case, as shown in FIG. 13(c), the block formatter 620 arranges the data bytes that are to be outputted from the symbol-byte converters 612 and 614 by the order of X0 to X5, Y0, Y1, X6 to X10, Y2 to Y7, X11, and Y8 to Y11. More specifically, the trellis encoder that is to perform the encoding process is decided based upon the position (or place) within the transmission frame in which each data byte is inserted. At this point, not only the mobile service data but also the main service data, the MPEG header data, and the RS parity data are also inputted to the trellis encoding module 256. Herein, it is assumed that, in order to perform the above-described operation, the block formatter 620 is informed of (or knows) the information on the data group format after the data-interleaving process.

FIG. 16 illustrates a block diagram of the block processor performing an encoding process at a coding rate of 1/N according to an embodiment of the present invention. Herein, the block processor includes (N-1) number of symbol interleavers 741 to 74N-1, which are configured in a parallel structure. More specifically, the block processor having the coding rate of 1/N consists of a total of N number of branches (or paths) including a branch (or path), which is directly transmitted to the block formatter 730. In addition, the symbol interleaver 741 to 74N-1 of each branch may each be configured of a different symbol interleaver. Furthermore, (N-1) number of symbol-byte converter 751 to 75N-1 each corresponding to each (N-1) number of symbol interleavers 741 to 74N-1 may be included at the end of each symbol interleaver, respectively. Herein, the output data of the (N-1) number of symbol-byte converter 751 to 75N-1 are also inputted to the block formatter 730.

In the example of the present invention, N is equal to or smaller than 12. If N is equal to 12, the block formatter 730 may align the output data so that the output byte of the 12th symbol-byte converter 75N-1 is inputted to the 12th trellis encoder. Alternatively, if N is equal to 3, the block formatter 730 may arranged the output order, so that the data bytes outputted from the symbol-byte converter 720 are inputted to the 1st to 4th trellis encoders of the trellis encoding module 256, and that the data bytes outputted from the symbol-byte converter 751 are inputted to the 5th to 8th trellis encoders, and that the data bytes outputted from the symbol-byte converter 752 are inputted to the 9th to 12th trellis encoders. At this point, the order of the data bytes outputted from each symbol-byte converter may vary in accordance with the

position within the data group of the data other than the mobile service data, which are mixed with the mobile service data that are outputted from each symbol-byte converter.

FIG. 17 illustrates a detailed block diagram showing the structure of a block processor according to another embodiment of the present invention. Herein, the block formatter is removed from the block processor so that the operation of the block formatter may be performed by a group formatter. More specifically, the block processor of FIG. 17 may include a byte-symbol converter 810, symbol-byte converters 820 and 840, and a symbol interleaves 830. In this case, the output of each symbol-byte converter 820 and 840 is inputted to the group formatter 850.

Also, the block processor may obtain a desired coding rate by adding symbol interleavers and symbol-byte converters. If the system designer wishes a coding rate of 1/N, the block processor needs to be provided with a total of N number of branches (or paths) including a branch (or path), which is directly transmitted to the block formatter 850, and (N-1) number of symbol interleavers and symbol-byte converters configured in a parallel structure with (N-1) number of branches. At this point, the group formatter 850 inserts place holders ensuring the positions (or places) for the MPEG header, the non-systematic RS parity, and the main service data. And, at the same time, the group formatter 850 positions the data bytes outputted from each branch of the block processor.

The number of trellis encoders, the number of symbol-byte converters, and the number of symbol interleavers proposed in the present invention are merely exemplary. And, therefore, the corresponding numbers do not limit the spirit or scope of the present invention. It is apparent to those skilled in the art that the type and position of each data byte being allocated to each trellis encoder of the trellis encoding module 256 may vary in accordance with the data group format. Therefore, the present invention should not be understood merely by the examples given in the description set forth herein. The mobile service data that are encoded at a coding rate of 1/N and outputted from the block processor 303 are inputted to the group formatter 304. Herein, in the example of the present invention, the order of the output data outputted from the block formatter of the block processor 303 are aligned and outputted in accordance with the position of the data bytes within the data group.

Signaling Information Processing

The transmitter 200 according to the present invention may insert transmission parameters by using a plurality of methods and in a plurality of positions (or places), which are then transmitted to the receiving system. For simplicity, the definition of a transmission parameter that is to be transmitted from the transmitter to the receiving system will now be described. The transmission parameter includes data group information, region information within a data group, the number of RS frames configuring a super frame (i.e., a super frame size (SFS)), the number of RS parity data bytes (P) for each column within the RS frame, whether or not a checksum, which is added to determine the presence of an error in a row direction within the RS frame, has been used, the type and size of the checksum if the checksum is used (presently, 2 bytes are added to the CRC), the number of data groups configuring one RS frame—since the RS frame is transmitted to one burst section, the number of data groups configuring the one RS frame is identical to the number of data groups within one burst (i.e., burst size (BS)), a turbo code mode, and a RS code mode.

Also, the transmission parameter required for receiving a burst includes a burst period—herein, one burst period corresponds to a value obtained by counting the number of fields starting from the beginning of a current burst until the beginning of a next burst, a positioning order of the RS frames that are currently being transmitted within a super frame (i.e., a permuted frame index (PFI)) or a positioning order of groups that are currently being transmitted within a RS frame (burst) (i.e., a group index (GI)), and a burst size. Depending upon the method of managing a burst, the transmission parameter also includes the number of fields remaining until the beginning of the next burst (i.e., time to next burst (TNB)). And, by transmitting such information as the transmission parameter, each data group being transmitted to the receiving system may indicate a relative distance (or number of fields) between a current position and the beginning of a next burst.

The information included in the transmission parameter corresponds to examples given to facilitate the understanding of the present invention. Therefore, the proposed examples do not limit the scope or spirit of the present invention and may be easily varied or modified by anyone skilled in the art. According to the first embodiment of the present invention, the transmission parameter may be inserted by allocating a predetermined region of the mobile service data packet or the data group. In this case, the receiving system performs synchronization and equalization on a received signal, which is then decoded by symbol units. Thereafter, the packet deformatter may separate the mobile service data and the transmission parameter so as to detect the transmission parameter. According to the first embodiment, the transmission parameter may be inserted from the group formatter **304** and then transmitted.

According to the second embodiment of the present invention, the transmission parameter may be multiplexed with another type of data. For example, when known data are multiplexed with the mobile service data, a transmission parameter may be inserted, instead of the known data, in a place (or position) where a known data byte is to be inserted. Alternatively, the transmission parameter may be mixed with the known data and then inserted in the place where the known data byte is to be inserted. According to the second embodiment, the transmission parameter may be inserted from the group formatter **304** or from the packet formatter **306** and then transmitted.

According to a third embodiment of the present invention, the transmission parameter may be inserted by allocating a portion of a reserved region within a field synchronization segment of a transmission frame. In this case, since the receiving system may perform decoding on a receiving signal by symbol units before detecting the transmission parameter, the transmission parameter having information on the processing methods of the block processor **303** and the group formatter **304** may be inserted in a reserved field of a field synchronization signal. More specifically, the receiving system obtains field synchronization by using a field synchronization segment so as to detect the transmission parameter from a pre-decided position. According to the third embodiment, the transmission parameter may be inserted from the synchronization multiplexer **240** and then transmitted.

According to the fourth embodiment of the present invention, the transmission parameter may be inserted in a layer (or hierarchical region) higher than a transport stream (TS) packet. In this case, the receiving system should be able to receive a signal and process the received signal to a layer higher than the TS packet in advance. At this point, the

transmission parameter may be used to certify the transmission parameter of a currently received signal and to provide the transmission parameter of a signal that is to be received in a later process.

In the present invention, the variety of transmission parameters associated with the transmission signal may be inserted and transmitted by using the above-described methods according to the first to fourth embodiment of the present invention. At this point, the transmission parameter may be inserted and transmitted by using only one of the four embodiments described above, or by using a selection of the above-described embodiments, or by using all of the above-described embodiments. Furthermore, the information included in the transmission parameter may be duplicated and inserted in each embodiment. Alternatively, only the required information may be inserted in the corresponding position of the corresponding embodiment and then transmitted. Furthermore, in order to ensure robustness of the transmission parameter, a block encoding process of a short cycle (or period) may be performed on the transmission parameter and, then, inserted in a corresponding region. The method for performing a short-period block encoding process on the transmission parameter may include, for example, Kerdock encoding, BCH encoding, RS encoding, and repetition encoding of the transmission parameter. Also, a combination of a plurality of block encoding methods may also be performed on the transmission parameter.

The transmission parameters may be grouped to create a block code of a small size, so as to be inserted in a byte place allocated within the data group for signaling and then transmitted. However, in this case, the block code passes through the block decoded from the receiving end so as to obtain a transmission parameter value. Therefore, the transmission parameters of the turbo code mode and the RS code mode, which are required for block decoding, should first be obtained. Accordingly, the transmission parameters associated with a particular mode may be inserted in a specific section of a known data region. And, in this case, a correlation of with a symbol may be used for a faster decoding process. The receiving system refers to the correlation between each sequence and the currently received sequences, thereby determining the encoding mode and the combination mode.

Meanwhile, when the transmission parameter is inserted in the field synchronization segment region or the known data region and then transmitted, and when the transmission parameter has passed through the transmission channel, the reliability of the transmission parameter is deteriorated. Therefore, one of a plurality of pre-defined patterns may also be inserted in accordance with the corresponding transmission parameter. Herein, the receiving system performs a correlation calculation between the received signal and the pre-defined patterns so as to recognize the transmission parameter. For example, it is assumed that a burst including 5 data groups is pre-decided as pattern A based upon an agreement between the transmitting system and the receiving system. In this case, the transmitting system inserts and transmits pattern A, when the number of groups within the burst is equal to 5. Thereafter, the receiving system calculates a correlation between the received data and a plurality of reference patterns including pattern A, which was created in advance. At this point, if the correlation value between the received data and pattern A is the greatest, the received data indicates the corresponding parameter, and most particularly, the number of groups within the burst. At this point, the number of groups may be acknowledged as 5. Hereinafter, the process of inserting and transmitting the transmission

parameter will now be described according to first, second, and third embodiments of the present invention.

First Embodiment

FIG. 18 illustrates a schematic diagram of the group formatter 304 receiving the transmission parameter and inserting the received transmission parameter in region A of the data group according to the present invention. Herein, the group formatter 304 receives mobile service data from the block processor 303. Conversely, the transmission parameter is processed with at least one of a data randomizing process, a RS frame encoding process, and a block processing process, and may then be inputted to the group formatter 304. Alternatively, the transmission parameter may be directly inputted to the group formatter 304 without being processed with any of the above-mentioned processes. In addition, the transmission parameter may be provided from the service multiplexer 100. Alternatively, the transmission parameter may also be generated and provided from within the transmitter 200. The transmission parameter may also include information required by the receiving system in order to receive and process the data included in the data group. For example, the transmission parameter may include data group information, and multiplexing information.

The group formatter 304 inserts the mobile service data and transmission parameter which are to be inputted to corresponding regions within the data group in accordance with a rule for configuring a data group. For example, the transmission parameter passes through a block encoding process of a short period and is, then, inserted in region A of the data group. Particularly, the transmission parameter may be inserted in a pre-arranged and arbitrary position (or place) within region A. If it is assumed that the transmission parameter has been block encoded by the block processor 303, the block processor 303 performs the same data processing operation as the mobile service data, more specifically, either a $\frac{1}{2}$ -rate encoding or $\frac{1}{2}$ -rate encoding process on the signaling information including the transmission parameter. Thereafter, the block processor 303 outputs the processed transmission parameter to the group formatter 304. Thereafter, the signaling information is also recognized as the mobile service data and processed accordingly.

FIG. 19 illustrates a block diagram showing an example of the block processor receiving the transmission parameter and processing the received transmission parameter with the same process as the mobile service data. Particularly, FIG. 19 illustrates an example showing the structure of FIG. 9 further including a signaling information provider 411 and multiplexer 412. More specifically, the signaling information provider 411 outputs the signaling information including the transmission parameter to the multiplexer 412. The multiplexer 412 multiplexes the signaling information and the output of the RS frame encoder 302. Then, the multiplexer 412 outputs the multiplexed data to the byte-bit converter 401.

The byte-bit converter 401 divides the mobile service data bytes or signaling information byte outputted from the multiplexer 412 into bits, which are then outputted to the symbol encoder 402. The subsequent operations are identical to those described in FIG. 9. Therefore, a detailed description of the same will be omitted for simplicity. If any of the detailed structures of the block processor 303 shown in FIG. 12, FIG. 15, FIG. 16, and FIG. 17, the signaling

information provider 411 and the multiplexer 412 may be provided behind the byte-symbol converter.

Second Embodiment

Meanwhile, when known data generated from the group formatter in accordance with a pre-decided rule are inserted in a corresponding region within the data group, a transmission parameter may be inserted in at least a portion of a region, where known data may be inserted, instead of the known data. For example, when a long known data sequence is inserted at the beginning of region A within the data group, a transmission parameter may be inserted in at least a portion of the beginning of region A instead of the known data. A portion of the known data sequence that is inserted in the remaining portion of region A, excluding the portion in which the transmission parameter is inserted, may be used to detect a starting point of the data group by the receiving system. Alternatively, another portion of region A may be used for channel equalization by the receiving system.

In addition, when the transmission parameter is inserted in the known data region instead of the actual known data. The transmission parameter may be block encoded in short periods and then inserted. Also, as described above, the transmission parameter may also be inserted based upon a pre-defined pattern in accordance with the transmission parameter. If the group formatter 304 inserts known data place holders in a region within the data group, wherein known data may be inserted, instead of the actual known data, the transmission parameter may be inserted by the packet formatter 306. More specifically, when the group formatter 304 inserts the known data place holders, the packet formatter 306 may insert the known data instead of the known data place holders. Alternatively, when the group formatter 304 inserts the known data, the known data may be directly outputted without modification.

FIG. 20 illustrates a block diagram showing the structure of a packet formatter 306 being expanded so that the packet formatter 306 can insert the transmission parameter according to an embodiment of the present invention. More specifically, the structure of the packet formatter 306 further includes a known data generator 351 and a signaling multiplexer 352. Herein, the transmission parameter that is inputted to the signaling multiplexer 352 may include information on the length of a current burst, information indicating a starting point of a next burst, positions in which the groups within the burst exist and the lengths of the groups, information on the time from the current group and the next group within the burst, and information on known data.

The signaling multiplexer 352 selects one of the transmission parameter and the known data generated from the known data generator 351 and, then, outputs the selected data to the packet formatter 306. The packet formatter 306 inserts the known data or transmission parameter outputted from the signaling multiplexer 352 into the known data place holders outputted from the data interleaver 305. Then, the packet formatter 306 outputs the processed data. More specifically, the packet formatter 306 inserts a transmission parameter in at least a portion of the known data region instead of the known data, which is then outputted. For example, when a known data place holder is inserted at a beginning portion of region A within the data group, a transmission parameter may be inserted in a portion of the known data place holder instead of the actual known data.

Also, when the transmission parameter is inserted in the known data place holder instead of the known data, the transmission parameter may be block encoded in short

periods and inserted. Alternatively, a pre-defined pattern may be inserted in accordance with the transmission parameter. More specifically, the signaling multiplexer **352** multiplexes the known data and the transmission parameter (or the pattern defined by the transmission parameter) so as to configure a new known data sequence. Then, the signaling multiplexer **352** outputs the newly configured known data sequence to the packet formatter **306**. The packet formatter **306** deletes the main service data place holder and RS parity place holder from the output of the data interleaver **305**, and creates a mobile service data packet of 188 bytes by using the mobile service data, MPEG header, and the output of the signaling multiplexer. Then, the packet formatter **306** outputs the newly created mobile service data packet to the packet multiplexer **240**.

In this case, the region A of each data group has a different known data pattern. Therefore, the receiving system separates only the symbol in a pre-arranged section of the known data sequence and recognizes the separated symbol as the transmission parameter. Herein, depending upon the design of the transmitting system, the known data may be inserted in different blocks, such as the packet formatter **306**, the group formatter **304**, or the block processor **303**. Therefore, a transmission parameter may be inserted instead of the known data in the block wherein the known data are to be inserted.

According to the second embodiment of the present invention, a transmission parameter including information on the processing method of the block processor **303** may be inserted in a portion of the known data region and then transmitted. In this case, a symbol processing method and position of the symbol for the actual transmission parameter symbol are already decided. Also, the position of the transmission parameter symbol should be positioned so as to be transmitted or received earlier than any other data symbols that are to be decoded. Accordingly, the receiving system may detect the transmission symbol before the data symbol decoding process, so as to use the detected transmission symbol for the decoding process.

Receiving System

FIG. **21** illustrates a block diagram of a digital broadcast receiving system according to one aspect of the present invention. In this broadcast receiving system, known data inserted into a mobile service region by a broadcast transmitter are used for carrier sync recovery, frame sync recovery, and channel equalization in order to enhance the performance of the receiving system. The broadcast receiving system shown in FIG. **21** includes a tuner **901**, a demodulator **902**, an equalizer **903**, a known sequence detector **904**, and a block decoder **905**. It further includes a data deformatter **906**, a Reed-Solomon (RS) frame decoder **907**, and a data derandomizer **908**. The broadcast receiving system may further include a data interleaver **909**, a Reed-Solomon (RS) decoder **910**, and a data derandomizer **911**. Hereinafter, the data deformatter **906**, the RS frame decoder **907**, and the data derandomizer **908** will be referred to as a mobile service data processor, and the data interleaver **909**, the RS decoder **910**, and the data derandomizer **911** will be referred to as a main service processor.

The tuner **901** tunes to a channel to receive an intermediate frequency (IF) broadcast signal including mobile service data and main service data, and outputs the signal to the demodulator **902** and to the known sequence detector **904**. The demodulator **902** converts the IF signal into a baseband signal by performing automatic gain control, carrier recovery, and timing recovery, and outputs the baseband signal to the equalizer **903** and to the known sequence detector **904**.

The equalizer **903** compensates channel distortion of the demodulated signal and outputs the channel-equalized signal to the block decoder **905**.

The known sequence detector **904** detects position information of known data inserted by a broadcast transmitter (not illustrated) from the signal outputted from the tuner **901** or from the demodulated signal, and outputs the position information and the known data to the demodulator **902** and the equalizer **903**. In addition, the known sequence detector **904** may further detect information required for distinguishing the mobile and main service data and output this information to the block decoder **905**. The information detected by the known sequence detector **903** can be used by any component of the broadcast receiving system shown in FIG. **21**, e.g., the data deformatter **906** and the RS frame decoder **907**.

The demodulator **902** uses the known data and the position information of the known data for timing recovery and carrier recovery, and the equalizer **903** uses the known data and the position information of the known data during the channel equalization. In addition, the equalizer **903** may further use a decoding result provided from the block decoder **905** for the channel equalization.

The broadcast transmitter periodically inserts known data into a data frame, as shown in FIG. **6A**, and FIG. **22** illustrates a part of region A of the data frame which includes a field sync segment and three known data regions. The data included in each region are indicated with a number of symbols. One byte of data includes four symbols, and each symbol includes two bits of data.

Referring to FIG. **22**, the top region of the data frame represents a field sync segment including 832 symbols, in which a known data sequence may be included. Each known data region includes a data region including initialization data required to initialize one or more trellis decoders. As an example, this data region may have a length of 48 symbols. Assuming that the broadcast receiving system includes 12 trellis decoders, the 48-symbol initialization data region may include 24 symbols of initialization data and 24 symbols of known data, because two symbols of initialization data are necessary to initialize one trellis decoder.

The first known data region located about 16 segments below the field sync segment includes a first sub-region, a second sub-region (slashed), a third sub-region (CIR TS), a fourth sub-region (reserved TS), and a fifth sub-region (ACQ TS). The first sub-region includes initialization data required to initialize one or more trellis decoders in the broadcast receiving system, and 48 data symbols may be assigned to this sub-region. The second sub-region (slashed) is provisionally assigned for aligning the third sub-region located next to the second sub-region with corresponding sub-regions (CIR TS) included in other known data regions, e.g., second and third known data regions shown in FIG. **22**. The second sub-region may include known data which can be used by the broadcast receiving system. The third sub-region (CIR TS) has a length of 1424 data symbols and includes a known data pattern to be used for channel equalization. As described above, a start position of the third sub-region is aligned with CIR TS sub-regions of the second and third known data regions.

The fourth sub-region (reserved TS) has a length of 252 data symbols and includes various transmission parameters. In one aspect of the present invention, the fourth sub-region includes information identifying a coding rate for each data region of a data group. Hereinafter, the mode used for indicating a code rate for each data region will be referred

to as a Serial Concatenated Convolution Code (SCCC) mode. An example of the SCCC mode is illustrated in the following table.

SCCC		SCCC Code Rate		
Mode	Region A	Region B	Region C	
0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	
1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	
2	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	
3	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	
4	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	
5	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	

For example, if the SCCC mode identified by the information included in the fourth sub-region indicates 2, the code rates for region A, region B, and region C are $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{2}$, respectively. The broadcast transmitting system generates different patterns indicating six SCCC modes, respectively, according to protocol, and may insert a corresponding pattern into the fourth sub-region of the first known data region. By detecting this pattern from the fourth sub-region, the broadcast receiving system can detect a SCCC mode before performing block decoding by the block decoder 905. In the fifth sub-region (ACQ TS), a known data pattern required for acquiring initial carrier frequency synchronization is periodically inserted. According to one aspect of the present invention, each known data pattern repeated in the fifth sub-region includes 576 symbols.

Referring back to FIG. 22, the second known data region located about 16 segments below the first known data region includes a first sub-region, a second sub-region (slashed), and a third sub-region (CIR TS). The first sub-region includes initialization data required to initialize one or more trellis decoders in the broadcast receiving system, and 48 data symbols may be assigned this sub-region. The second sub-region (slashed) is provisionally assigned for aligning the third sub-region located next to the second sub-region with corresponding sub-regions (CIR TS) included in other known data regions, e.g., first and third known data regions shown in FIG. 22. The second sub-region may include known data which can be used by the broadcast receiving system. The third sub-region (CIR TS) has a length of 1424 data symbols and includes a known data pattern to be used for channel equalization. A start position of the third sub-region is aligned with CIR TS sub-regions of the first and third known data regions.

Finally, the third known data region located about 16 segments below the second known data region includes a first sub-region including initialization data required to initialize one or more trellis decoders in the broadcast receiving system, and a second sub-region (CIR TS) including a known data pattern to be used for channel equalization. 12 data bytes may be assigned to the first sub-region, and only two symbols of each data byte are assigned for trellis decoder initialization. The remaining 24 symbols in the first sub-region can be used for channel equalization.

Referring to FIG. 22, region A1 located between the field sync segment and the first known data region includes 11892 symbols of normal data, region A2 located between the first and second known data regions includes 10480 symbols of normal data, and region A3 located between second and third known data regions includes 11888 symbols of normal data. The normal data include main service data, mobile service data, RS parity data, and MPEG header data.

FIG. 23 is a flow chart illustrating a method of detecting positions of known data according to one aspect of the present invention. The known sequence detector 904 initially detects the position of the first known data region using a plurality of repeated data patterns included in the fifth sub-region of the first known data region (S921). Next, the known sequence detector 904 can estimate the position of the field sync segment from the detected position of the first known data region according to protocol (S922), and can further estimate positions of the second, third, fourth, and fifth known data regions in a similar manner (S923). The estimated position information of the field sync and known data regions are provided to the demodulator 902 and the channel equalizer 903. For example, the position information of the field sync and CIR TS region of each known data region is provided to the equalizer 903. Then the equalizer 903 estimates a channel impulse response (CIR) using these position information.

The known sequence detector 904 may further estimate an initial frequency offset value using the plurality of repeated data patterns included in the fifth sub-region of the first known data region, while the position of the first known data region is detected. Then the demodulator 902 can compensate a carrier frequency offset using the known data position information and the estimated initial frequency offset value. FIG. 24 illustrates a detailed block diagram of the known sequence detector 904 shown in FIG. 21. It will be assumed that the input signal is oversampled at a sampling rate of N. The known sequence detector 904 includes N partial correlation units 1101 to 101N, a known sequence position detection and frequency offset decision unit (hereinafter, detection unit) 1020, a known sequence extractor 1030, a buffer 1040, and a multiplier 1050, an Numerically Controlled Oscillator (NCO) 1060, a frequency offset estimator 1070, and an adder 1080.

The first partial correlation unit 1011 includes a 1/N decimator, and a partial correlator. The second partial correlation unit 1012 includes a 1 sample delay, a 1/N decimator, and a partial correlator. The Nth partial correlation unit 101N includes a N-1 sample delay, a 1/N decimator, and a partial correlator. Each partial correlation unit changes a phase of an oversampled symbol in accordance with a phase of an original symbol, decimates the phase-changed symbol, and calculates a partial correlation value of the decimated symbol. Namely, the input signal is decimated at a rate of 1/N for each sampling phase and each decimated signal is inputted to a corresponding partial correlator.

For example, if an input signal is oversampled at a rate of 2 (N=2), one symbol includes 2 samples, only two partial correlation units 1011 and 1012 are required, and each decimator has a rate of $\frac{1}{2}$. The decimator included in the partial correlation unit 1011 decimates each sample between two symbol locations. In addition, the 1 sample delay included in the partial correlation unit 1012 delays input samples for one sample, and the decimator included in the partial correlation unit 1012 decimates each sample between two symbol locations among the delayed input samples. Each partial correlation unit 1011 and 1012 outputs a correlation value and an estimated coarse frequency offset value corresponding to the correlation value to detection unit 1020 for each predetermined period of VSB symbols.

The detection unit 1020 stores the output of the partial correlators for a predetermined period, determines a position having the maximum correlation value as a position of known data, and further determines an estimated frequency offset value corresponding to the determined position as a coarse frequency offset value. The known sequence position

indicator represents a position of the first known data region having the plurality of repeated data patterns. The positions of the field sync segment and the second to fifth known data regions can be estimated based on the position of the first known data region. Furthermore, positions of each CIR TS sub-region (1424 symbols) can be estimated. The known sequence position indicator is provided to the known sequence extractor **1030**, the demodulator **902**, and the equalizer **903**, and the estimated coarse frequency offset value is provided to the adder **1080** and the NCO **1060**.

The buffer **1040** temporarily stores the input signal while each partial correlation unit performs known data position detection and coarse frequency offset estimation, and outputs the stored to the known sequence extractor **1030**. The known sequence extractor **1030** extracts known data from the output of the buffer **1040** using the known sequence position indicator, and outputs the extracted known data to the multiplier **1050**. The NCO **1060** generates a complex signal corresponding to the coarse frequency offset value outputted from the detection unit **1020** and outputs the complex signal to the multiplier **1050**. The multiplier **1050** compensates a coarse frequency offset of the known data extracted from the known sequence extractor **1030** by multiplying the extracted known data with the complex signal generated by the NCO **1060**.

The frequency offset estimator **1070** estimates a fine frequency offset value from the known data outputted from the multiplier **1050**. Finally, the adder **1080** estimates an initial frequency offset value by adding the estimated coarse frequency offset value and the estimated fine frequency offset value. In the present invention, both of coarse and fine frequency offsets are compensated in order to obtain a more precise initial frequency offset value. If the input signal includes a data frame shown in FIG. 22, the known sequence detector **904** can estimate an initial frequency offset value by calculating correlation of the plurality of repeated data patterns included in the first known data region.

The known sequence detector **904** outputs the position information of the field sync segment and a known data pattern inserted in the CIR TS sub-region in each of the first, second, third known data regions shown in FIG. 22 to the demodulator **902** and the equalizer **903**, respectively.

Reference will now be made in detail to channel equalization of the equalizer **903** which uses the information provided from the known sequence detector **904**. Referring back to FIG. 6A, a long known data sequence can be periodically added to region A, but not to regions R and C. According to one aspect of the invention, a data group for channel equalization is divided into regions A, B, and C, as shown in FIG. 6A. Region A is divided into A1 to A5, region B is divided into B1 and B2, and region C is divided into C1 to C3. Therefore, a different channel impulse response (CIR) is estimated and applied for each data region in the data group. In addition, the CIR is estimated using the reference known data and reference field sync which are pre-known to the broadcast receiving system to perform channel equalization with better stability.

Hereinafter, an CIR estimated from the field sync will be referred to as CIR_FS and CIRs estimated from the five known data sequences shown in FIG. 6A will be referred to as CIR_N0, CIR_N1, CIR_N2, CIR_N3, and CIR_N4. According to one aspect of the invention, the CIRs estimated from the field sync and the known data sequences are used for channel equalization. Any one of the CIRs can be used, or a CIR generated by interpolating or extrapolating more than one CIRs can be used based on characteristic of each data region in the data group. Typical examples of the

interpolation and extrapolation are linear interpolation and linear extrapolation, and other schemes may be used as well.

FIG. 25 is a flow chart illustrating a channel equalization method according to one aspect of the present invention. An input data group (current data group) is temporarily stored (**S931**), and CIRs of the field sync data and the known data sequences included in the stored data group are estimated (**S932**). In step **S931**, a portion of the data group can be stored, because regions B1, C1, and A1 are received before the first known data region and the channel equalization for regions B1, C1, and A1 are performed using a CIR estimated from the first known data region.

In step **S933**, the channel equalization for region C1 of a current data group may be performed using a CIR which is estimated by extrapolating CIR_FS and CIR_N0 estimated from the current data group. Alternatively, the channel equalization for region C1 of the current data group can be performed using any one of CIR_N4 estimated from a previous data group and CIR_FS estimated from the current data group. If the extrapolated CIR is used, the data included in region C1 must be stored until CIR_N0 is estimated because the region C1 data are inputted before the first known data region in the current group. The data included in region C1 can be stored in a memory in the broadcast receiving system, or in an external storage.

In step **S933**, the channel equalization for region B1 of a current data group may be performed using a CIR which is estimated by extrapolating CIR_FS and CIR_N0 estimated from the current data group. In further example, the channel equalization for region B1 can be performed using CIR_FS of the current data group. Similarly, the data included in the region B1 must be stored until CIR_N0 (or CIR_FS) is estimated because the B1 region data are inputted before the first known data region. The data included in region B1 can be stored in a memory in the broadcast receiving system, or in an external storage.

In step **S934**, the channel equalization for region A1 of a current data group may be performed using a CIR which is estimated by interpolating CIR_FS and CIR_N0 estimated from the current data group. Alternatively, the channel equalization for region A1 can be performed using any one of CIR_FS and CIR_N0 estimated from the current data group. If region A1 is equalized using the interpolated CIR, the data included in region A1 must be stored until CIR_N0 (or CIR_FS) is estimated. The data included in region A1 can be stored in a memory in the broadcast receiving system, or in an external storage.

For Ai, (i=2, 3, 4, and 5) of a current data group, the channel equalization may be performed using a CIR which is estimated by interpolating CIR_N(i-1) and CIR_N(i) of the current data group (**S935**). Alternatively, the channel equalization for Ai may be performed using any one of CIR_N(i-1) and CIR_N(i). For example, channel equalization for A2 (i=2) is performed using a CIR estimated by interpolating CIR_N1 and CIR_N2 of the current data group or using any one of CIR_N1 and CIR_N2 of the current data group.

In step **S936**, the channel equalization for regions B2, C2, and C3 may be performed using a CIR estimated by interpolating CIR_N3 and CIR_N4 estimated from the current data group. Alternatively, the channel equalization may be performed using CIR_N4 estimated from the current data group.

FIG. 26 illustrates the channel equalizer **903** shown in FIG. 21 according to one aspect of the present invention. The channel equalizer **903** includes a memory **1500**, an overlap unit **1501**, a first Fast Fourier Transform (FFT) unit

1502, a multiplier 1503, a CIR estimator 1504, a CIR calculator (interpolator or extrapolator) 1505, a zero-padding unit 1506, a second FFT unit 1507, a coefficient calculator 1508, an Inverse Fast Fourier Transform (IFFT) unit 1509, and a save unit 1510.

An input signal is temporarily stored in the memory 1509 and is outputted to the overlap unit 1501. For example, the memory 1500 is used when the channel equalization for at least one of regions C1, B1, and A1 in a data group is performed using a CIR estimated by extrapolating or interpolating CIR_FS and CIR_N0 estimated from field sync data and known data which are received after regions C1, B1, and A1. In other words, the memory 1500 must store the data in regions C1, B1, and A1 until CIR_FS and CIR_N0 are estimated.

The overlap unit 1501 overlaps the data outputted from the memory 1500 at a predetermined overlap rate, and outputs the overlapped data to the first FFT unit. The first FFT unit converts the overlapped data in a time domain into frequency domain data using a FFT algorithm, and outputs the converted data to the multiplier 1503. The multiplier 1503 compensates channel distortion of the frequency domain data outputted from the first FFT unit 1502 by performing complex multiplication on the frequency domain data with equalization coefficients calculated from the coefficient calculator 1508. The IFFT unit 1509 converts the channel-equalized data in the frequency domain into time domain data using an IFFT algorithm. The save unit 1510 saves the equalized data in the time domain by extracting effective data in reverse order of the overlap unit 1501.

The CIR estimator 1504 estimates CIRs of a channel (i.e., CIR_N0 to CIR_N4) using the known data actually received from the broadcast transmitting system and reference known data pre-known to the broadcast receiving system by protocol. For this reason, the CIR estimator 1504 receives known data position information from the known sequence detector 904. Similarly, the CIR estimator 1504 further estimates other CIR of the channel (i.e., CIR_FS) using the field sync data actually received from the broadcast transmitting system and reference field sync data pre-known to the receiving system by protocol. Therefore, it further receives field sync position information from the known sequence detector 904. The reference known data and the reference field sync data may be pre-stored in the CIR estimator or can be provided from any other component in the broadcast receiving system shown in FIG. 21.

The CIR estimator 1504 may estimate the CIRs using a Least Square (LS) algorithm. The CIRs estimated by the CIR estimator 1504 are provided to the CIR calculator 1505. The CIR calculator 1505 bypasses or interpolates/extrapolates the CIRs based on region information of a data region to be equalized. It is assumed that the CIR calculator 1505 knows a position of data to be equalized in a data frame or data group. For example, if region B1 is to be equalized, the CIR calculator 1505 extrapolates CIR_FS and CIR_N0 estimated from a current data group. In a further example, if region A1 is to be equalized, the CIR calculator 1505 interpolates CIR_FS and CIR_N0 estimated from the current data group. The channel equalizer 903 may further include a first CIR cleaner (not illustrated) which operates only during the extrapolation, and a second CIR cleaner (not illustrated) which operates only during the interpolation. The CIRs estimated using the known data may include jitters due to noise, and these jitters may be removed by the first and second CIR cleaner.

The CIR interpolation is performed in the CIR calculator 1505 by multiplying two CIRs estimated by the CIR esti-

mator 1504 with coefficients values and adding the multiplied CIRs. A portion of noise in the interpolated CIRs may be offset. Therefore, the CIR calculator 1505 uses original CIRs including noise when performing CIR interpolation.

For example, if the CIR calculator 1505 performs CIR interpolation, the first CIR cleaner does not operate, and second CIR cleaner performs CIR cleaning on the interpolated CIR. On the other hand, the CIR extrapolation is performed in the CIR calculator 1505 by using a difference between two CIRs estimated by the CIR estimator 1504, where the extrapolated CIR is located outside of the estimated CIRs. In this case, the noise included in the CIRs may be amplified. Therefore, the first CIR cleaner performs CIR cleaning on the estimated CIRs and outputs cleaned CIRs to the CIR calculator 1505. During the extrapolation, the second CIR cleaner does not operate, and the extrapolated CIR is outputted to the zero-padding unit 1506.

When the second FFT 1507 converts an input CIR in the time domain into a frequency domain CIR, the length of the input CIR may not conform to a FFT size (e.g., the length of the input CIR is less than the FFT size). For this reason, the zero-padding unit 1506 adds zeros such that the CIR length conforms to the FFT size. Any one of the interpolated, extrapolated, and estimated CIRs can be zero-padded.

The second FFT unit 1504 converts the time domain CIR from the zero-padding unit 1506 into a frequency domain CIR, which is outputted to the coefficient calculator 1508. The coefficient calculator 1508 calculates equalization coefficients using the converted the frequency domain CIR, and outputs the coefficients to the multiplier 1503. According to one aspect of the present invention, the coefficient calculator 1508 can use a Minimum Mean Square Error (MMSE) algorithm.

Meanwhile, the channel equalized data from the equalizer 903 inputs to block decoder 905. The block decoder 905 performs on the inputted data as inverse processes of the transmitting system if the data being inputted to the block decoder 905 after being channel equalized from the equalizer 903 correspond to the mobile service data having additional encoding and trellis encoding performed thereon by the transmitting system, trellis decoding and additional decoding processes. Alternatively, if the data being inputted to the block decoder 905 correspond to the main service data having only trellis encoding performed thereon, and not the additional encoding, only the trellis decoding process is performed on the inputted data as the inverse process of the transmitting system.

The data group decoded by the block decoder 905 is inputted to the data deformatter 906, and the main service data are inputted to the data deinterleaver 909. According to another embodiment, the main data may also bypass the block decoder 905 so as to be directly inputted to the data deinterleaver 909. In this case, a trellis decoder for the main service data should be provided before the data deinterleaver 909. When the block decoder 905 outputs the data group to the data deformatter 906, the known data, trellis initialization data, and MPEG header, which are inserted in the data group, and the RS parity, which is added by the RS encoder/non-systematic RS encoder or non-systematic RS encoder of the transmitting system, are removed. Then, the processed data are outputted to the data deformatter 906. 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. If the transmitting system includes signaling information in the data group upon transmission, the signaling information is outputted to the data deformatter 906.

41

More specifically, if the inputted data correspond to the main service data, the block decoder **905** performs Viterbi 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 mobile service data, the block decoder **905** outputs a hard decision value or a soft decision value with respect to the inputted mobile service data. In other words, if the inputted data correspond to the mobile service data, the block decoder **905** performs a decoding process on the data encoded by the block processor and trellis encoding module 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 external code. And, the block processor and the trellis encoder may be viewed as an internal code. In order to maximize the performance of the external code when decoding such concatenated codes, the decoder of the internal code should output a soft decision value. Therefore, the block decoder **905** may output a hard decision value on the mobile service data. However, when required, it may be more preferable for the block decoder **905** to output a soft decision value.

Meanwhile, the data deinterleaver **909**, the RS decoder **910**, and the derandomizer **911** are blocks required for receiving the main service data. Therefore, the above-mentioned blocks may not be required in the structure of a digital broadcast receiving system that only receives the mobile service data. The data deinterleaver **909** performs an inverse process of the data interleaver included in the transmitting system. In other words, the data deinterleaver **909** deinterleaves the main service data outputted from the block decoder **905** and outputs the deinterleaved main service data to the RS decoder **910**. The RS decoder **910** performs a systematic RS decoding process on the deinterleaved data and outputs the processed data to the derandomizer **911**. The derandomizer **911** receives the output of the RS decoder **910** and generates a pseudo random data byte identical to that of the randomizer included in the digital broadcast transmitting system. Thereafter, the derandomizer **911** 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.

Meanwhile, the data being outputted from the block decoder **905** to the data deformatter **906** are inputted in the form of a data group. At this point, the data deformatter **906** already knows the structure of the data that are to be inputted and is, therefore, capable of identifying the signaling information, which includes the system information, and the mobile service data from the data group. Thereafter, the data deformatter **906** outputs the identified signaling information to a block for processing signaling information (not shown) and outputs the identified mobile service data to the RS frame decoder **907**. More specifically, the RS frame decoder **907** receives only the RS encoded and CRC encoded mobile service data that are transmitted from the data deformatter **906**.

The RS frame encoder **907** performs an inverse process of the RS frame encoder included in the transmitting system so as to correct the error within the RS frame. Then, the RS frame decoder **907** adds the 1-byte MPEG synchronization service data packet, which had been removed during the RS frame encoding process, to the error-corrected mobile service data packet. Thereafter, the processed data packet is outputted to the derandomizer **908**. The operation of the RS frame decoder **907** will be described in detail in a later

42

process. The derandomizer **908** performs a derandomizing process, which corresponds to the inverse process of the randomizer included in the transmitting system, on the received mobile service data. Thereafter, the derandomized data are outputted, thereby obtaining the mobile service data transmitted from the transmitting system. Hereinafter, detailed operations of the RS frame decoder **907** will now be described.

FIG. **27** illustrates a series of exemplary step of an error correction decoding process of the RS frame decoder **907** according to the present invention. More specifically, the RS frame decoder **907** groups mobile service data bytes received from the data deformatter **906** so as to configure an RS frame. The mobile service data correspond to data RS encoded and CRC encoded from the transmitting system. FIG. **27(a)** illustrates an example of configuring the RS frame. More specifically, the transmitting system divided the RS frame having the size of $(N+2)*235$ to $30*235$ byte blocks. When it is assumed that each of the divided mobile service data byte blocks is inserted in each data group and then transmitted, the receiving system also groups the $30*235$ mobile service data byte blocks respectively inserted in each data group, thereby configuring an RS frame having the size of $(N+2)*235$. For example, when it is assumed that an RS frame is divided into 18 $30*235$ byte blocks and transmitted from a burst section, the receiving system also groups the mobile service data bytes of 18 data groups within the corresponding burst section, so as to configure the RS frame. Furthermore, when it is assumed that N is equal to 538 (i.e., $N=538$), the RS frame decoder **907** may group the mobile service data bytes within the 18 data groups included in a burst so as to configure a RS frame having the size of $540*235$ bytes.

Herein, when it is assumed that the block decoder **905** outputs a soft decision value for the decoding result, the RS frame decoder **907** 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 the 18 data groups included in a single burst, the RS frame having the size of $540*235$ 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 1 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 1 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 1 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 1 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 540×235 bytes, the reliability map is also configured to have the size of 540×235 bytes. FIG. 27(a') illustrates the process steps of configuring the reliability map according to the present invention. Meanwhile, if a RS frame is configured to have the size of $(N+2) \times 235$ bytes, the RS frame decoder 907 performs a CRC syndrome checking process on the corresponding RS frame, thereby verifying whether any error has occurred in each row. Subsequently, as shown in FIG. 27(b), a 2-byte checksum is removed to configure an RS frame having the size of $N \times 235$ 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 $N \times 235$ number of the reliability information bytes remain, as shown in FIG. 27(b').

After performing the CRC syndrome checking process, the RS frame decoder 907 performs RS decoding 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. 27(c), the CRC error flag corresponding to each row within the RS frame is verified. Thereafter, the RS frame decoder 907 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 a 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.

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 235 N-byte rows, as shown in FIG. 27(d). Thereafter, as shown in FIG. 27(f), 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 907 compares the absolute value of the soft decision value of the block decoder

905 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, 8 bits, each being determined by the code of the soft decision value, are grouped to form 1 data byte. Accordingly, the reliability information on this 1 data byte is indicated on the reliability map. Therefore, as shown in FIG. 27(e), 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. 27(f).

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

45

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 configured of 187 N-byte rows (or packets) maybe obtained, as shown in FIG. 27(f). Furthermore, the RS frame having the size of N*187 bytes is sequentially outputted in N number of 187-byte units. Herein, as shown in FIG. 27(g), the 1-byte MPEG synchronization byte that was removed by the transmitting system is added at the end of each 187-byte packet, thereby outputting 188-byte mobile service data packets.

As described above, the digital broadcasting system and the data processing method according to the present invention have the following advantages. More specifically, the digital broadcasting receiving system and method according to the present invention is highly protected against (or resistant to) any error that may occur when transmitting mobile service data through a channel. And, the present invention is also highly compatible to the conventional 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.

Additionally, 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, by multiplexing mobile service data with main service data into a burst structure, the power consumption of the receiving system may be reduced. Furthermore, 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. An apparatus for transmitting a broadcast signal, the apparatus comprising:

an encoder configured to encode mobile data for Forward Error Correction (FEC);

a group formatter configured to form data groups, wherein at least one the data groups includes the encoded mobile data, a first known data sequence, and a plurality of second known data sequences;

a packet formatter configured to format packets including data in the data groups; and

a transmission unit configured to transmit the broadcast signal including the packets,

wherein the plurality of second known data sequences are spaced 16 segments apart within the at least one of the data groups,

wherein the at least one of the data groups includes a transmission parameter inserted between the first known data sequence and one of the plurality of second known data sequences,

46

wherein the first known data sequence has a first M-symbol sequence and a second M-symbol sequence, wherein the first M-symbol sequence and the second M-symbol sequence have a first pattern,

wherein each of the plurality of second known data sequences has a second pattern other than the first pattern, and

wherein the second pattern is positioned from a last symbol to a previous N symbol in each of the plurality of second known data sequences.]

[2. The apparatus of claim 1, further comprising a block processor configured to perform block processing on the encoded mobile data.]

[3. A method for transmitting a broadcast signal, the method comprising:

encoding mobile data for Forward Error Correction (FEC);

forming data groups, wherein at least one of the data groups includes the encoded mobile data, a first known data sequence, and a plurality of second known data sequences;

formatting packets including data in the data groups; and transmitting the broadcast signal including the packets, wherein the plurality of second known data sequences are spaced 16 segments apart within the at least one of the data groups,

wherein the at least one of the data groups includes a transmission parameter inserted between the first known data sequence and one of the plurality of second known data sequences,

wherein the first known data sequence has a first M-symbol sequence and a second M-symbol sequence, wherein the first M-symbol sequence and the second M-symbol sequence have a first pattern,

wherein each of the plurality of second known data sequences has a second pattern other than the first pattern, and

wherein the second pattern is positioned from a last symbol to a previous N symbol in each of the plurality of second known data sequences.]

[4. The method of claim 3, further comprising performing block processing on the encoded mobile data.]

5. A method for processing broadcast data in a transmitter, the method comprising:

first encoding the broadcast data,

wherein the broadcast data is carried in packets and the packets are mapped to frames which are units for forward error correction (FEC) encoding,

wherein the packets are mapped to a frame in order and the packets are split in byte units and mapped to the frame;

second encoding the first-encoded broadcast data by an error detection encoding method;

third encoding the second-encoded broadcast data based on different code rates;

interleaving the third-encoded broadcast data,

wherein dummy data is included in a memory for the interleaving and a number of interleaving data added by the dummy data are larger than a number of data to be interleaved in the interleaving operation;

modulating signal frames including known data and the interleaved broadcast data; and

transmitting a broadcast signal carrying the modulated signal frames,

wherein the known data are used for channel estimation, and

47

wherein the known data include at least two pre-defined patterns and are inserted in predetermined positions in the signal frames,

wherein the signal frames carry different regions for the broadcast data, the different regions having broadcast 5 data encoded in the different code rates, and

wherein the signal frames further include signaling information and the signaling information includes information related to the different code rates.

* * * * *

10

48