(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 December 2004 (29.12.2004)

PCT

(10) International Publication Number
WO 2004/114675 A1

(51) International Patent Classification[7]: H04N 7/64, H03M 13/15, H04N 7/62

(21) International Application Number: PCT/US2004/019010

(22) International Filing Date: 16 June 2004 (16.06.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/479,600 18 June 2003 (18.06.2003) US

(71) Applicant *(for all designated States except US)*: **THOMSON LICENSING S.A.** [FR/FR]; 46, Quai A. Le Gallo, F-92648 Boulogne (FR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MARKMAN, Ivonete** [BR/US]; 11388 Royal Court, Carmel, IN 46032 (US). **LIU, Weixiao** [CN/US]; 9353 D College Drive, Indianapolis, IN 46240 (US).

(74) Agents: **TRIPOLI, Joseph, S.** et al.; c/o Thomson Licensing Inc., 2 Independence Way, Suite 200, Princeton, NJ 08540 (US).

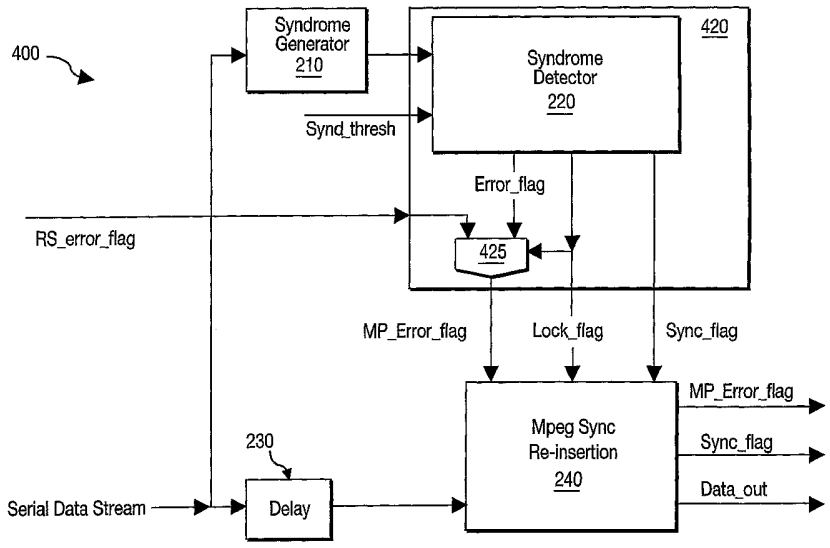
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR ERROR DETECTION OF COMPRESSED VIDEO IN A DIGITAL MEDIA RECEIVER



(57) Abstract: A method (500) and apparatus (400) for identifying errors in an MPEG-2 packet delivered to a receiver transport layer. Processing a stream of MPEG-2 packets received within Reed-Solomon (RS) codewords and delivering them to the transport layer together with a combined error flag (MP_Error_flag) value pertaining to each particular packet based upon the output of a Syndrome Detector (220) and the outputs (RS_error_flag) of a Reed-Solomon decoder. An apparatus (400) for processing and for delivering to a transport layer a combined error flag (MP_Error_flag) value associated with every bit of particular packet, the combined error flag value being logically dependent upon the synchronization error flag (Error_flag) value and upon the existence of any uncorrectable RS decoder errors affecting the particular packet.

WO 2004/114675 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR ERROR DETECTION OF COMPRESSED VIDEO IN A DIGITAL MEDIA RECEIVER

1. Field of the Invention

The present invention relates to transmitting and receiving multimedia data including digital video and audio, and more particularly to a method and apparatus for reliably identifying errors in an MPEG-2 stream broadcast over such a digital transmission system to the receiver transport layer.

2. Description of the Related Art

Digital transmission systems offer consumers high-quality multimedia data including compressed audio and video streams. For broadcasters, the compression of data allows several programs to be delivered over the same analog bandwidth required for a single channel. The audio and video components of a program are compressed at the source and time-multiplexed with other programs and system information needed to recreate the original program. The digital multiplex is processed by a physical layer and transmitted to the consumer. At the consumer end, the receiver processes the signal to recover the multiplexed digital streams, extracts the program of interest, and decodes the compressed audio and video for presentation on a video/audio display such as a television.

To promote the development of interoperable components from different manufacturers, the MPEG-2 international compression standard was developed. The standard does not specify the techniques for encoding, multiplexing, and decoding the bit streams, but only the format of the data. This allows the manufacturers the opportunity to differentiate their products via the way in which they use resources such as silicon, processor power, and memory, and through their ability to conceal or recover from errors. The standard is composed of three primary parts covering systems, video, and audio. The video and audio parts specify the format of the compressed video and audio data, while the systems part specifies the formats for multiplexing the audio and video data for one or more programs as well as information necessary for recovery of the programs.

The ANSI/SCTE 07 2000 (formerly SCTE DVS 031) and ITU-T J.83B standards, which are nearly identical, describe a digital transmission system for cable distribution of video, sound and data services. In particular, the ANSI/SCTE 07 2000

standard describes the adopted standard for digital cable transmission in the U.S. In both standards, the data format input to the physical layer (channel coding and modulation) is assumed to be MPEG-2 transport.

FIG. 1 is a block diagram showing the basic physical layer of the digital cable transmission system. As shown in FIG. 1, in the physical layer, the "MPEG framing" is the outermost layer of processing, and is conventionally viewed as being computationally separate and distinct from the Forward Error Correction (FEC) (including a Reed-Solomon) "Encode"/"Decode" layer. At the transmitter end, the MPEG framing block is followed by the Forward Error Correction (FEC) encoder (Encode) and the 64 or 256 Quadrature Amplitude Modulator (QAM). An FEC system is a class of methods for controlling errors in a one-way communication system such as an MPEG-2 stream. An FEC encoder sends extra information (e.g., "redundant" parity bits) along with the data, which can be used by the receiver to check and correct the data.

Errors, known and unknown, are inherent in transport stream delivery and can occur at any time. Unknown errors such as bit corruption or data loss can occur at any bit position of the stream, and may mislead the transport into unusual behavior. The FEC encoder comprises concatenated systems including a Reed-Solomon (RS) encoder, an interleaver capable of several modes, a randomizer and a trellis encoder. It produces high coding gain at moderate complexity and overhead. The FEC system is optimized for quasi error free operation at a threshold output error event rate of one error event per 15 minutes. At the receiver, the corresponding functions of demodulation and FEC decoding, including a Reed-Solomon (RS) decoder, are performed, followed by the MPEG framing processing block. The Reed-Solomon decoder processes each RS block or codeword and attempts to correct errors and recover the original data. The number and type of errors that can be corrected depends on the characteristics of the Reed-Solomon code.

The MPEG framing processing block at the receiver delivers an MPEG-2 transport data stream comprising a continuous stream of fixed length (188 bytes) packets that are transmitted in serial fashion, most significant bit (MSB) first. The so-called MPEG-2 "link" header of each packet contains fields for packet synchronization and identification, error indication, and conditional access. The subsequent adaptation header carries synchronization and timing information for

decoding and presentation process. The payload (1496 bits) can contain any multimedia data including compressed video and audio streams.

The packets of the MPEG-2 transport layer comprise 188 bytes, beginning with a four-byte transport packet header, the header having one (first) byte for synchronization purposes (called "sync byte" and having a fixed value of 47Hex), and three subsequent bytes containing service identification, scrambling and control information. The four-byte transport packet header is followed by 184 bytes of MPEG-2 or auxiliary data.

A broadcast MPEG-2 stream may contain several multiplexed programs of audio and video data, along with the necessary system data, and each MPEG-2 packet is identified by a unique "program ID" tag (PID) within the packet header.

The MPEG-2 sync-byte is intended to facilitate MPEG-2 packet boundary delineation at a decoder. However, unlike many other digital transmission standards, the method used for MPEG-2 synchronization in the digital cable transmission system physical layer is de-coupled from the Forward Error Correction (FEC) synchronization. First, the MPEG-2 packet does not contain an integer number of FEC frames, nor even Reed-Solomon (RS) codewords. Reed-Solomon (RS) Coding, using a (128,122) code, provides block encoding and decoding to correct up to three 7-bit symbols within an RS block. The data portion of an RS codeword comprises 122 7-bit symbols (122x7 = 854 bits) and the MPEG packet (excluding the sync-byte) comprises 187 8-bit bytes (187x8 = 1496 bits). Thus, the ratio of RS codewords to MPEG-2 packets is nearly but not exactly two to one. Hence, the MPEG-2 packets and the FEC frames, or the MPEG-2 packets and RS codewords are asynchronous with respect to each other. A single MPEG-2 packet could be carried within (and split between) two consecutive RS codewords or within three consecutive RS codewords (e.g., at least two consecutive RS codewords). A particular MPEG-2 packet can be carried within a portion of one RS codeword and a portion (or the entirety) of another (adjacent) RS codeword. And a particular MPEG-2 packet could be carried with a portion of a first RS codeword, the entirety of a second RS codeword, and a portion of a third (consecutive) RS codeword. Similarly, one RS codeword could spread between 2 MPEG-2 packets.

This synchronization de-coupling feature was intended to introduce the flexibility, for example, to enable the system to carry Asynchronous Transfer Mode (ATM) packets easily without interfering with ATM synchronization.

At the transmitter, the MPEG-2 packet sync-byte is replaced inside the MPEG framing block at the transmission site by a parity checksum that is a coset of an FIR parity check linear block code to make use of the information bearing capacity of the sync-byte. The parity checksum, which is a coset of an FIR (finite impulse response) parity check linear block code (LBC or FIR-PCC) is substituted for this sync-byte, supplying improved packet delineation functionality, and error detection capability independent of the FEC layer. The parity checksum is computed over the adjacent 187 bytes, which constitute the immediately preceding (prior) MPEG-2 packet content (minus its sync-byte). The parity checks of the block code are computed at the receiver by observing the output of a finite impulse response (FIR), linear time-invariant (binary) filter. The parity check structure is based on a PN sequence generated by a (binary) primitive polynomial.

Hence, the MPEG framing block at the receiver site needs to decode this parity check block code in order to recover the sync-byte and then lock to it. The MPEG framing block then delivers MPEG packet synchronization to the downstream receiver blocks, including the transport block. The sync-byte checksum at the transmitter site is computed by passing the 1496 payload bits through a linear feedback shift register (LFSR) as described by the following equation:

$$f(X) = [1 + X^{1497}b(X)]/g(X), \text{ where } g(X) = 1 + X + X^5 + X^6 + X^8 \text{ and } b(X) = 1 + X + X^3 + X^7.$$

An offset of 67Hex is added to this checksum result for improved autocorrelation properties, and causes a 47Hex result to be produced during a syndrome decode operation when a valid code word is present. This structure allows for a computationally efficient implementation of the parity check FIR filter, in a recursive manner, that is generally self-synchronizing and therefore supports simultaneous packet synchronization and error detection. The decoder computes a sliding checksum on the serial data stream, using the detection of a valid code word to detect the start of a packet.

A parity check matrix is used by the decoder to identify a valid checksum. The code has been designed such that when the appropriate 188 bytes of bitstream (including the checksum) are multiplied against the parity check matrix, a positive match is indicated when the calculated product produces a 47Hex result. (Note that the checksum is calculated based on the previous 187 bytes and not the 187 bytes

yet to be received by the MPEG-2 sync decoder. This is in contrast to the conventional notion of an MPEG packet structure, in that the sync-byte is usually described as the first byte of a received packet.)

FIG. 2 shows an example of a prior art MPEG framing block 200 at the receiver end. As shown in FIG. 2, the output of this block 200 may include the "Data_out" stream (in serial or parallel format), a "sync" signal (Sync_flag) identifying the position of the sync-byte in the "Data_out" stream, and an "error" signal (Error_flag) identifying whether the packet is considered invalid (uncorrectable errors) or error-free, as determined by the regular detection of the sync-byte checksum by the Syndrome Detector 220. Outputs of the MPEG framing block 200 may also include (not shown) an output "clock", and a "valid-data" signal identifying when data is present at the output "data" stream.

The data stream input to the MPEG framing block 200 at the receiver end is serialized (Serial Data Stream) and is sent through the Syndrome Generator 210. Following the Syndrome Generator 210, the Syndrome Detector 220 compares the Syndrome Generator's output with 47Hex for a number of packets, N, and a programmable threshold, synd_thresh, establishes whether a sync-byte has actually been detected. For example, if during N packets, the number of Syndrome Generator 210 outputs equal to 47Hex is greater than or equal to synd_thresh, then a sync-byte has been detected. A Lock_flag indicates whether or not periodic sync-bytes have been detected within the data stream, for example, by being 1 or 0, respectively. A Sync_flag indicates the sync-byte position within the data stream by, for example, being 1 during the sync-byte and 0 otherwise. Once a locked alignment condition is established, the absence of a valid code word at the expected location in the Serial Data Stream will indicate a packet error. The Error_flag of the previous packet can then be set to 1; otherwise, the packet is considered error free and the Error_flag is set to 0. In the absence of a locked condition, the Error_flag may be set to 1.

On a parallel path, the original data stream is appropriately delayed (see Delay 230 in FIG. 2) and is sent to the MPEG Sync Re-insertion block 240 where the predetermined sync-byte value is inserted in place of the parity checksum that was created at the transmitter-end MPEG framing block. Hence, the output data stream (Data_out) output by the receiver-end MPEG framing block 200 is a restored standard MPEG-2 transport stream. This data output (Data_out) can be in either

serial or parallel mode. Two additional signals not shown in FIG. 2 are also sent to the transport layer: the "clock" and the "valid" or "enable signal associated with the data.

After a lock (synchronization) detection, the MPEG sync re-inserter 240 within the MPEG framing block 200 of the prior art inserts the predetermined sync-byte value into the sync-byte position identified by the parity check block decoder, outputs the Sync_flag signal, the Error_flag signal, the valid and clock signals, and sends the data stream (and the Error_flag) to the transport layer. The Sync_flag and the Error_flag sent to the transport layer are the same as created by the syndrome detector 220.

The Error_flag is delivered by the physical layer to the transport layer (via Error_flag_out) and identifies whether the packet is considered invalid (uncorrectable errors) or error free. However, in the ANSI/SCTE 07 2000 (formerly SCTE DVS 031) and ITU-T J.83B standards, due to the decoupling of the MPEG-2 synchronization and the FEC synchronization, particularly, the RS decoder synchronization, the value of the Error_flag depends upon identification of packet errors by the prior art Syndrome Decoder (210 & 220 in FIG. 2), according to the flowchart in FIG. 3.

FIG. 3 is a flowchart describing the method 300 of generation of the Error_flag within the Syndrome Detector 220 of FIG. 2. In each flowchart (e.g., FIG. 3, 5, and 6), "Y" denotes "YES" and marks each branch of a decision step that is used when the comparison or statement indicated within the associated diamond (decision block) is TRUE. Conversely, "N" denotes "NO" and marks each branch of a decision step that is used when the comparison or statement indicated within the associated diamond (decision block) is FALSE.

The method of FIG. 3 comprises a loop Pktloop that begins at the Start and that repeats for each MPEG-2 packet until the end of the received Serial Data Steam is detected (e.g., "End of Stream?" equals "YES"), and includes steps (S301, S302, S304, and S305) that are performed for each MPEG-2 packet. Step S301 is a decision branch step of determining whether (Y) or not (N) the Lock_flag output of the Syndrome Detector of Fig. 2 is equal to "1" (which indicates MPEG-2 synchronization). Step S302 is a decision branch step of determining whether (Y) or not (N) the output of Syndrome Generator of Fig. 2 is equal to 47Hex at the supposed sync-byte location (which confirms MPEG-2 synchronization). Step S304 is a decision branch step of detecting (Y) or not detecting (N) the End of the Data Stream input to the prior art MPEG framing block 200 of Fig. 2, which upon being detected

(Y) would terminate (End) the loop Pktloop; if the End of the Data Stream is not detected (N) in step S304, then loop Pktloop continues to repeat and step S305 is performed, wherein the index packet_count is incremented.

The method of loop Pktloop further includes essentially performing the alternative steps of generating the Error_flag (See FIG.2) value pertaining to a particular (prior) MPEG-2 packet (packet_count-1) equal to '1' (AS310) to indicate an error, or equal to '0' (AS320) to indicate no error, based upon the value of Lock_flag and upon the value of the output of the Syndrome Generator (210 of FIG. 2) at the sync-byte location of the sync-byte (first byte) of the current (packet_count) MPEG-2 packet following the particular (prior) MPEG-2 packet (packet_count-1).

Because of the properties of the syndrome encoder (see FIG. 1) and decoder (shown as 210 & 220 in FIG. 2) within the MPEG framing blocks of the transmitter and receiver, respectively, when the Error_flag value is generated, it always refers to the condition of the previous (packet_count-1) packet. A Delay (230 in FIG. 2) in the Serial Data Stream into the MPEG Sync Re-insertion block 240 provides for the final alignment between the data and the error flag (Error_flag_out, being the same as Error_flag), as shown in FIG 2.

If the MPEG-2 syndrome detector 220 is locked, there is a high likelihood that no errors exist. However, the parity check block code used for the sync-byte detection is not very powerful in identifying errors. It is possible that there are a number of errors in a packet, even though the location of the sync-byte (Sync_flag) is found.

A Reed-Solomon encoder in the physical layer (See FIG. 1) outputs a flag, RS_error_flag, denoting an RS decoding error that is associated with every bit of a particular RS codeword. The RS_error_flag is '0' whenever the RS codeword is free of errors after the RS correction of errors and '1' if an uncorrectable number of errors are found in the RS codeword. Due to the de-coupling feature of the MPEG-2 synchronization and the FEC synchronization, particularly, the RS decoder synchronization, the boundaries between RS codewords are not aligned with the MPEG-2 packet boundaries, and data bits carried by two or three adjacent (consecutive) RS codewords can be found within the same MPEG-2 packet. Thus, the RS_error_flag can be '0' and '1' at different portions of the same MPEG-2 packet. Since an RS codeword can potentially spread between two MPEG-2 packets, an uncorrectable RS codeword can potentially create two uncorrectable MPEG-2

packets, even while the MPEG-2 syndrome detector (220 of FIG. 2) is locked and while the value of the Error_flag that it outputs indicates no error.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for processing and for delivering to a transport layer, a stream of fixed-length packets received within a series of Reed-Solomon (RS) codewords, each packet including a header portion containing a sync-byte, the apparatus comprising: an RS decoder for performing RS decoding of the series of RS codewords and for outputting a first RS error flag value associated with the bits of a first RS codeword and a second RS error flag value associated with the bits of a second RS codeword and a third RS error flag value associated with the bits of a third RS codeword, the first, second and third RS codewords being consecutive in the series; wherein the bits of a particular packet are carried within at least the first and second RS codewords; a Synchronization Detector for detecting the position of the sync-byte in each packet, and for outputting a synchronization error flag value associated with the bits of the particular packet; wherein the apparatus is adapted to generate a combined error flag value associated with the bits of the particular packet, the combined error flag value being logically dependent upon the synchronization error flag value and upon at least the first RS error flag value and the second RS error flag value. When the bits of a particular packet are contained within the bits of the first, second and a third RS codeword, the combined error flag value will be logically dependent upon the synchronization error flag value and upon the first, second and third RS error flag values. If any one or more of the (first, second or third) RS codewords that contain any of the bits of a particular packet contain an uncorrectable number of errors, even if the Synchronization Detector indicates a lock condition and no error, the combined error flag value output to the transport level (in place of the error-output of the Synchronization Detector) will indicate an error associated with all the bits of the particular packet.

BRIEF DESCRIPTION OF THE DRAWINGS

The above features of the present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

FIG. 1 is a block diagram depicting the physical layer at the transmitting and receiving ends of a prior art digital cable transmission system;

FIG. 2 is a block diagram of a prior art MPEG framing block at the receiver end of a prior art digital transmission system;

FIG. 3 is a flowchart describing the generation of the Error_flag within the prior art Syndrome Detector of FIG. 2;

FIG. 4 is a block diagram of an MPEG framing block at the receiver end of a digital transmission system according to an embodiment of the present invention;

FIG. 5 is a flowchart that describes the method performed to generate a combined error flag based upon the output of a Syndrome Generator and the output of a Reed-Solomon decoder according to an embodiment of the present invention; and

FIG. 6 is a flowchart that describes an algorithm performed to determine a value of the combined error flag according to an embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 4 is a block diagram of the MPEG framing block 400 of the present invention. The MPEG framing block 400 of the present invention is similar to the MPEG framing block 200 of the prior art in FIG. 2, except that the disclosed method and the Error Combining circuit 425 for generating the Error_flag takes advantage of information available from both the RS decoder and from the MPEG-2 syndrome detector 220.

MPEG framing block 400 receives an additional input RS_error_flag, which is an output of an RS decoder, identifying whether there are no errors (value of '0') or uncorrectable errors (value of '1') in the current RS codeword. The value of RS_error_flag is associated with every bit of a particular codeword and is sent to the MPEG framing block 400 along with the Serial Data Stream. This RS_error_flag is transmitted to the Sync & Error Detector block 420 where it is combined with the information of the conventional Error_flag and Lock_flag (e.g., outputs from the prior art Syndrome Detector 220) to generate the error flag MP_error_flag pertaining to the current MPEG-2 packet. In some embodiments of the invention, the method of combining the RS_error_flag and the conventional Error_flag and Lock_flag outputs (available from the prior art Syndrome Detector 220) can be performed by a distinct Error Combining circuit 425 added to the prior art MPEG framing block 200 as

depicted in FIG. 4.

The output MP_error_flag output by the Error Combining circuit 425 (and of the Sync & Error Detector block 420) is the logical (and sequential) combination of the inputs RS_error_flag and the conventional Error_flag (and implicitly, Lock_flag, which may be a function of Error_flag) and is exemplarily defined according to Truth Table 1 or Truth Table 2.

Inverse of Lock_flag	Error_flag(0)	RS_error_flag_0	RS_error_flag_1	MP_error_flag(0)
0	0	0	0	0
0	0	0	1	1
0	0	1	X	1
0	1	X	X	1
1	X	X	X	1

TRUTH TABLE 1

In Truth Tables 1 and 2, "X" denotes "Don't Care" (e.g., the value at X does not affect the result) and that both a "1" or a "0" in that position produce the same result. In Truth Table 1, the MP_error_flag(0) column represents the value of the MP_error_flag output of Error Combining circuit 425 available after the end of (after the sync-byte of the MPEG-2 packet following after) the current MPEG-2 packet, in the case where the current MPEG-2 packet is carried within two (not three) consecutive RS codewords. The "Inverse of Lock_flag" (first) column represents the value of the inverse of the Lock_flag output of Syndrome Detector 220; the Error_flag(0) column represents the value of the Error_flag output of Syndrome Detector 220 after the end of (at the checksum-encoded sync-byte following after) the current MPEG-2 packet; the RS_Error_flag_0 column represents the value of the RS_error_flag output (from the FEC layer) pertaining to the RS codeword containing at least the first bit of the current MPEG-2 packet; the RS_Error_flag_1 column

represents the value of the RS_error_flag output (from the FEC layer) pertaining to the (next consecutive) RS codeword containing at least the last bit of the current MPEG-2 packet.

Inverse of Lock_flag	Error_flag(0)	RS_error_flag_0	RS_error_flag_1	RS_error_flag_2	MP_error_flag(0)
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	X	1
0	0	1	X	X	1
0	1	X	X	X	1
1	X	X	X	X	1

TRUTH TABLE 2

In Truth Table 2, the MP_error_flag(0) column represents the value of the MP_error_flag output of Error Combining circuit 425 available after the end of (after the sync-byte of the MPEG-2 packet following after) the current MPEG-2 packet, in the case where the current MPEG-2 packet is carried within three (not only two) consecutive RS codewords. The Lock-flag first column represents the value of the inverse of the Lock_flag output of Syndrome Detector 220; the Error_flag(0) column represents the value of the Error_flag output of Syndrome Detector 220 after the end of (at the checksum-encoded sync-byte following after) the current MPEG-2 packet; the RS_Error_flag_0 column represents the value of the RS_error_flag output (from the FEC layer) pertaining to the RS codeword containing at least the first bit of the current MPEG-2 packet (e.g., the first bit of its sync-byte); the RS_Error_flag_1 column represents the value of the RS_error_flag output (from the FEC layer) pertaining to the (next consecutive) RS codeword containing at least the center bit of the current MPEG-2 packet; the RS_Error_flag_2 column represents the value of the RS_error_flag output (from the FEC layer) pertaining to the (next consecutive) RS

codeword containing at least the last bit of the current MPEG-2 packet. As previously noted, because RS codewords in the MPEG-2 specification are smaller than MPEG-2 packets, the RS codeword containing the first bit of an MPEG-2 packet cannot be the same RS codeword that contains the last bit of the same MPEG-2 packet, and therefore the MPEG-2 packet will be comprised of data carried within at least two consecutive RS codewords.

As illustrated in Truth Tables 1 and 2, the Error Combining circuit 425 effectively performs a cumulative logical OR operation upon inputs Error_flag and RS_error_flag, including at least RS_error_flag_0 and RS_error_flag_1 (and also including RS_error_flag_2 when applicable) (and in some embodiments, upon the inverse of the Lock_flag signal, which may be derived from the Error_flag signal) to produce the output MP_Error_flag, which is finally available after processing the last bit of the MPEG-2 packet and the subsequent checksum-encoded sync-byte (the first byte of the next MPEG-2 packet). Because the output of the Error Combining circuit 425 may depend upon at least two (e.g. three) consecutive sequential values of RS_error_flag per each MPEG-2 packet, the Error Combining circuit 425 is not depicted graphically as a simple "OR" logic gate within the Sync & Error Detector block 420 shown in FIG. 4. Persons skilled in the art of digital circuit design will recognize that there is a wide variety of known circuits which can be adapted to implement the function of Error Combining circuit 425 and the results of Truth Tables 1 and 2. The circuits available to implement the function of the Error Combining circuit 425 and Truth Table 1 are known by persons skilled in the art to include a combinatorial logic circuit, a sequential logic circuit, a latch, a multiplexor, a shift register, a microprocessor, a Programmable Logic Array, a finite state machine, and an ASIC core.

The method performed by the Error Combining circuit 425 (and of the Sync & Error Detector block 420) can be summarized according to the following rules:

A) If the prior art MPEG-2 Syndrome Detector 220 detects that synchronization is unlocked (e.g., Lock_flag = 0), or synchronization is locked (e.g., Lock_flag = 1) and Error_flag = 1, the value of MP_error_flag pertaining to a particular MPEG packet is output as 1, indicating uncorrectable errors, regardless of the applicable value(s) of the RS_error_flag.

B1) If the prior art MPEG-2 Syndrome Detector 220 detects that MPEG-2

synchronization is locked (e.g., Lock_flag =1), and Error_flag = 0, the contemporaneous value(s) of the RS decoder error flag, RS_error_flag, is used in determining the validity of a particular MPEG-2 packet (and the value of MP_Error_flag at the end of a particular MPEG-2 packet). After MPEG-2 synchronization is locked (e.g., Lock_flag =1) and Error_flag = 0, in the MPEG framing block 400: if the value of RS_error_flag is '1' during any portion of a particular MPEG-2 packet, then that the whole MPEG-2 packet is invalidated and MP_Error_flag is set to '1' for the duration of the entire packet;

B2) If MPEG-2 synchronization is locked (e.g., Lock_flag =1) and Error_flag = 0, in the MPEG framing block 400: and if the value of RS_error_flag is '0' during the entire MPEG-2 packet, then the value of MP_error_flag is set to '0' for the duration of the entire MPEG-2 packet.

FIG. 5 and FIG. 6 depict flowcharts that together describe an exemplary embodiment of the invention in the form of a method 500 and an exemplary enabling algorithm 600 to be performed to generate the MP_error_flag in the current invention in accordance with Truth Tables 1 and 2. The method 500 in FIG. 5 is similar to the prior art method 300 in FIG. 3, except that the output MP_error_flag value (analogous to Error_flag in FIG. 3) assumes the value of the intermediate RS_check_flag (once the Syndrome Detector 220 is locked and the sync-byte value 47Hex appears as expected at the sync-byte location in the Serial Data Stream).

The intermediate RS_check_flag represents a one-bit memory recording the accumulated error (RS_error_flag =1) detected in the value of RS_error_flag during the processing of all the RS codewords containing any of the bits of a particular MPEG-2 packet. RS_check_flag may be reset to 0 at the beginning of each new MPEG-2 packet, but is set to "1" when (immediately, or after) the value of RS_error_flag is equal to "1" during any part (even the first bit, or the last bit, or any other bit) of a particular MPEG-2 packet.

The algorithm 600 used for generation of the RS_check_flag is described by the flowchart in FIG. 6. Note that when the RS_check_flag is generated, it refers to the condition of the previous packet (packet_count -1), as shown in FIG. 6, so it is aligned with the Error_flag in FIG. 5. Additionally, a Delay 230 in the data stream

provides for the final alignment between the Data and the Error_flag, as shown in FIG. 4.

FIG. 5 is a flowchart describing the method 500 of generation of the MP_error_flag within the Sync & Error Detector block 420 of FIG. 4. As in the method 300 of FIG. 3, the method 500 of FIG. 5 comprises a loop Pktloop5 that repeats for each MPEG-2 packet until the end of the received Serial Data Steam is detected (e.g., "End of Stream?" equals "YES"), and includes steps (S301, S302, S304, S305) that are performed for each MPEG-2 packet.

The method 500 of FIG. 5 is distinguished from the method 300 of FIG. 3 essentially by performing (in method 500 of FIG. 5) the alternative steps of generating the value of an error flag, MP_Error_flag, pertaining to a particular (prior) MPEG-2 packet (packet_count-1) equal to the value of RS_check_flag (AS510), as above described, to indicate a (possible) applicable RS decoder error, or equal to "1" (AS520) to indicate a synchronization error, based upon consideration of the value(s) of the RS_error_flag that appertain to the bits of the particular (prior, packet_count -1) MPEG-2 packet as well as the value of the output of the Syndrome Generator (210 of FIG. 2) at the sync-byte location of the sync-byte (first byte) of the next (current, packet_count) MPEG-2 packet following the particular (prior, packet_count-1) MPEG-2 packet.

Because of the properties of the syndrome encoder (see FIG. 1) and decoder (shown as 210 & 220 in FIG. 2) within the MPEG framing blocks of the transmitter and receiver, respectively, when the MP_Error_flag value is generated, it refers to the condition of the previous (packet_count-1) MPEG-2 packet. A Delay (230 in FIG. 4) in the Serial Data Stream into the MPEG Sync Re-insertion block (240 in FIG. 4) provides for the final alignment between the data and the error flag (Error_flag_out, being the same as Error_flag), as shown in FIG 4.

FIG. 6 is a flowchart describing an exemplary algorithm 600 which may be used to generate value of the intermediate RS_check_flag which is used in the method 500 of FIG. 5 to determine the appropriate value of the MP_error_flag to be output by the Sync & Error Detector block 420 of FIG. 4, which is subsequently output (in place of Error_flag) by the MPEG framing block 400 of FIG. 4.

As in the method 500 of FIG 5, the algorithm of FIG. 6 comprises the loop Pktloop6 that repeats for each MPEG-2 packet until the End of the received Serial Data Steam is detected (e.g., in step S304, "End of Stream?" equals "YES"), and

further includes a nested loop BitLoop containing a comparison/decision step Detect1 that is performed for each bit of each MPEG-2 packet. The algorithm 600 of FIG. 6 includes a nested loop BitLoop (nested within the packet loop Pktloop6) that repeats for each bit within a particular MPEG-2 packet until the last bit of the MPEG-2 packet is detected (e.g., in step S606, "End of Packet?" equals "YES"), and includes a comparison/decision step Detect1 that is performed for each bit of each MPEG-2 packet. Step S601 is an index (packet_count) initialization step, that sets packet_count, which is used in Pktloop6 equal to 0. Step S602 is a flag and index (rs_check and bit_count) initialization step, that sets both rs_check and bit_count, which are used within Pktloop6, equal to 0. Step S304 is a decision branch step of detecting (Y) or not detecting (N) the End of the Data Stream input to the MPEG framing block 400 of Fig. 4, which upon being detected (Y) would terminate (End) the loop Pktloop6; if the End of the Data Stream is not detected (N) in step S304, then loop Pktloop6 continues to repeat. The index packet_count is incremented, in step S305, each time that loop PktLoop6 is performed.

Within the nested loop BitLoop, for each bit of a particular (prior, packet_count – 1) MPEG-2 packet, the value of RS_error_flag associated with that bit is compared (in decision step Detect1) with "1" in order to determine the existence of an RS decoding error that is applicable to the particular (prior, packet_count – 1) MPEG-2 packet. If the value of RS_error_flag pertaining to at least one bit within the particular MPEG-2 packet is equal to "1" (thus indicating an RS decoding error potentially affecting that packet), the value of rs_check is set equal to "1" (S604). The index bit_count is incremented, in step S605, each time that loop BitLoop is performed, and may be used in step S606 to enable the detection of the last bit (End) of the current Packet. Upon the detection of the End of the particular MPEG-2 packet (e.g., in step S606, "End of Packet?" = YES), the value of RS_check_flag pertaining to that particular MPEG-2 packet is finally set equal to rs_check (S608).

Thus, if the value of RS_error_flag pertaining to at least one bit within the particular MPEG-2 packet is equal to "1", then the value of RS_check_flag pertaining to that particular MPEG-2 packet is set equal to "1". Thence, upon the performance of the alternative step AS510 in the method of FIG. 5 (which assumes no detected MPEG-2 synchronization error), the value of MP_Error_flag, pertaining to a particular (prior) MPEG-2 packet (packet_count-1) is set to "1", indicating an error, and that "combined error" flag MP_Error_flag is delivered to the transport layer in the place of

the Error_flag generated by the Syndrome Detector 220 of the prior art.

Conversely, if the value of RS_error_flag pertaining to every bit within the particular MPEG-2 packet is equal to "0", then the value of RS_check_flag pertaining to that particular MPEG-2 packet remains equal to "0" (it's initialized value). Thence, upon the performance of the alternative step AS510 in the method of FIG. 5 (which assumes no detected MPEG-2 synchronization error), the value of MP_Error_flag, pertaining to a particular (prior) MPEG-2 packet (packet_count-1) is set to "0", indicating no error, and that "combined error" flag MP_Error_flag is delivered to the transport layer in the place of the Error_flag generated by the Syndrome Detector 220 of the prior art.

These and other features of the present invention may be readily ascertained by one of ordinary skill in the pertinent art based on the principles disclosed herein. It is to be understood that the principles of the present invention may be implemented in various forms of hardware, software, firmware, special purpose processors, or combinations thereof.

The present invention is implemented as a combination of hardware and software. Moreover, a software implementation may be implemented as an application program tangibly embodied on a program storage unit or fixed media. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units ("CPU"), a random access memory ("RAM"), and input/output ("I/O") interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU.

Exemplary embodiments of the invention have been explained above and are shown in the figures. However, the present invention is not limited to the exemplary embodiments described above, and it is apparent that variations and modifications can be effected by those skilled in the art within the spirit and scope of the present invention.

It is to be further understood that, because some of the constituent system components and methods depicted in the accompanying drawings may be implemented in software (e.g., adapted to be executed by a personal computer or a

set-top box), the actual connections between the system components or the method blocks may differ depending upon the manner in which the software programmed to implement the present invention is programmed. Given the principles of the present invention disclosed herein, one of ordinary skill in the pertinent art will be able to contemplate these and similar implementations or configurations of the present invention.

Although the illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various changes and modifications may be effected therein by one of ordinary skill in the pertinent art without departing from the scope or spirit of the present invention. All such changes and modifications are intended to be included within the scope of the present invention as set forth in the appended claims.

Therefore, the exemplary embodiments should be understood not as limitations but as examples. The scope of the present invention is not determined by the above description but by the accompanying claims, and variations and modifications may be made to the embodiments of the invention without departing from the scope of the invention as defined by the appended claims and equivalents.

Claims:

1. An apparatus (400) for processing and for delivering to a transport layer, a stream of fixed-length packets received within a series of Reed-Solomon (RS) codewords, each packet including a header portion containing a sync-byte, the apparatus comprising:

an RS decoder for performing RS decoding of the series of RS codewords and for outputting a first RS error flag value associated with the bits of a first RS codeword and a second RS error flag value associated with the bits of a second RS codeword and a third RS error flag value associated with the bits of a third RS codeword, the first, second and third RS codewords being consecutive in the series; wherein the bits of a particular packet are carried within at least the first and second RS codewords;

a Synchronization Detector (420) for detecting the position of the sync-byte in each packet, and for outputting a synchronization error flag (Error_flag) value associated with the bits of the particular packet;

wherein the apparatus (400) is adapted to generate a combined error flag (MP_Error_flag) value associated with the bits of the particular packet, the combined error flag (MP_Error_flag) value being logically dependent upon the synchronization error flag (Error_flag) value and upon at least the first RS error flag value and the second RS error flag value.

2. The apparatus of claim 1 wherein the combined error flag (MP_Error_flag) value is logically dependent upon the first RS error flag value, the second RS error flag value, and the third RS error flag value.

3. The apparatus of claim 1 wherein the bits of a particular packet are carried within the first, second, and third RS codewords.

4. The apparatus of claim 3, wherein the apparatus generates the combined error flag (MP_Error_flag) value based upon the contents of a memory for storing at least one of the first, second and third RS error flag values.

5. The apparatus of claim 1 wherein the apparatus is further adapted to deliver the combined error flag (MP_Error_flag) value to the transport layer with the packet associated with the combined error flag (MP_Error_flag) value.

6. The apparatus of claim 5, wherein the apparatus generates the combined error flag (MP_Error_flag) value based upon the contents of a latch adapted to store at least one of the first RS error flag value and the second RS error flag value.

7. The apparatus of claim 5, further comprising an MPEG Sync-Byte Re-insertion circuit (240) for inserting a predetermined value into the sync-byte location indicated by an MPEG synchronization signal (Sync_flag), and further adapted to output the combined error flag value (MP_Error_flag) to the transport layer.

8. The apparatus of claim 1, wherein the Synchronization Detector (420) is an MPEG-2 sync-byte detector that includes a Syndrome Detector (220) for detecting a checksum-encoded sync-byte.

9. The apparatus of claim 1 wherein if at least one of the first RS error flag value, and the second RS error flag value indicate the existence of an error, then the combined error flag (MP_Error_flag) value will indicate the existence of an error.

10. The apparatus of claim 1 wherein if at least one of the first, second and third RS error flag values indicates the existence of an error, then the combined error flag (MP_Error_flag) value will indicate the existence of an error.

11. The apparatus of claim 1 wherein at least one of the first RS error flag value, the second RS error flag value, and the synchronization error flag (Error_flag) value are binary values.

12. The apparatus of claim 11 wherein all of the first RS error flag value, the second RS error flag value, the third RS error flag value and the synchronization error flag (Error_flag) value are binary values.

13. The apparatus of claim 12 wherein the binary logic value "1" indicates the existence of an error in each of the first RS error flag, the second RS error flag, and the synchronization error flag (Error_flag).

14. The apparatus of claim 1 further comprising an error combining circuit (425) adapted to generate the combined error flag (MP_Error_flag) having the combined error flag value associated with every bit of the particular packet.

15. The apparatus of claim 14 wherein the error combining circuit (425) logically combines the synchronization error flag (Error_flag) value with both the first RS error flag value and the second RS error flag value, and generates the combined error flag (MP_Error_flag) value based upon the synchronization error flag (Error_flag) value and the at least one of the first RS error flag value and the second RS error flag value.

16. The apparatus of claim 14 wherein the error combining circuit (425) logically combines the synchronization error flag (Error_flag) value with all three of first, second and third RS error flag values, and generates the combined error flag (MP_Error_flag) value based upon the synchronization error flag (Error_flag) value and all three of the first, second and third RS error flag values.

17. The apparatus of claim 14 wherein the error combining circuit (425) includes a multiplexor, wherein the synchronization error flag (Error_flag) and at least one of the first RS error flag value and the second RS error flag value are input to the multiplexor.

18. The apparatus of claim 1 wherein the error combining circuit (425) includes a multiplexor, wherein the synchronization error flag (Error_flag) and at least one of the first, second and third RS error flag values are input to the multiplexor.

19. The apparatus of claim 14 wherein the error combining circuit (425) includes a logical OR gate, wherein the synchronization error flag (Error_flag) and at least one of the first RS error flag value and the second RS error flag value are input to the OR gate.

20. The apparatus of claim 14 wherein the error combining circuit (425) includes a logical OR gate, wherein the synchronization error flag (Error_flag) and at least one of the first, second and third RS error flag values are input to the OR gate.

21. The apparatus of claim 14 wherein the error combining circuit (425) includes a sequential logic circuit, wherein the synchronization error flag (Error_flag) and both of the first RS error flag value and the second RS error flag value are input to the sequential logic circuit.

22. The apparatus of claim 21 wherein the error combining circuit (425) includes a sequential logic circuit, wherein the synchronization error flag (Error_flag) and all of the first, second and third RS error flag values are input to the sequential logic circuit.

23. The apparatus of claim 14 wherein the error combining circuit (425) includes a finite state machine.

24. A method for processing a stream of fixed-length packets received within a series of Reed-Solomon (RS) codewords, each packet including a header portion containing a sync-byte, the method comprising:

generating the value of an error flag pertaining to a particular packet based upon the output of a Syndrome Generator indicating alternatively the presence and the absence of a synchronization error and the output of a Reed-Solomon decoder indicating alternatively the presence and the absence of an uncorrectable RS decoding error.

25. The method of claim 24, wherein a second value of the error flag indicates the absence of an RS decoder error and the absence of a synchronization error pertaining to the particular packet.

26. The method of claim 24, wherein a second value of the error flag indicates that the packet synchronization is locked and without synchronization error,

and that the packet was not received within any RS codewords containing an uncorrectable number or errors.

27. The method of claim 24 further comprising the steps of outputting the stream of fixed-length packets and outputting the error flag to a transport layer.

28. The method of claim 27 wherein the step of outputting the stream of fixed-length packets to a transport layer is performed only if the value of the error flag based upon the output of a Syndrome Detector and the output of a Reed-Solomon decoder indicates the absence of an RS decoder error and the absence of a synchronization error pertaining to the particular packet.

29. The method of claim 27 further comprising the step of inserting a predetermined sync-byte value into the sync-byte position within the header of the particular packet.

30. The method of claim 27 wherein the step of inserting a predetermined sync-byte value into the sync-byte position within the header of the particular packet is performed only if the value of the error flag based upon the output of a Syndrome Detector and the output of a Reed-Solomon decoder indicates the absence of an RS decoder error and the absence of a synchronization error pertaining to the particular packet.

31. The method of claim 27 wherein the step of inserting a predetermined sync-byte value into the sync-byte position within the header of the particular packet is performed only if the value of the error flag based upon the output of a Syndrome Detector and the output of a Reed-Solomon decoder indicates the absence of an RS decoder error and the absence of a synchronization error pertaining to the particular packet and the absence of a sync-byte checksum error pertaining to the particular packet.

32. A computer program product for a set-top-box that comprises a set of instructions, which, when loaded into the set-top-box, causes the set-top-box to carry out the method, for processing a stream of fixed length packets, claimed in claim 24.

33. A computer program product for a television set that comprises a set of instructions, which, when loaded into the television set, causes the television set to carry out the method, for processing a stream of fixed length packets, claimed in claim 24.

34. A method for error detection of a compressed video bitstream transmitted in a fixed length packet transport scheme, said method comprising:

determining if reception of fixed length packets are synchronized;

if reception of fixed length packets are synchronized, then determining whether an individual packet is indicated as having uncorrectable errors; and

if said individual packet is indicated as having said uncorrectable errors, then marking said packet as having an error.

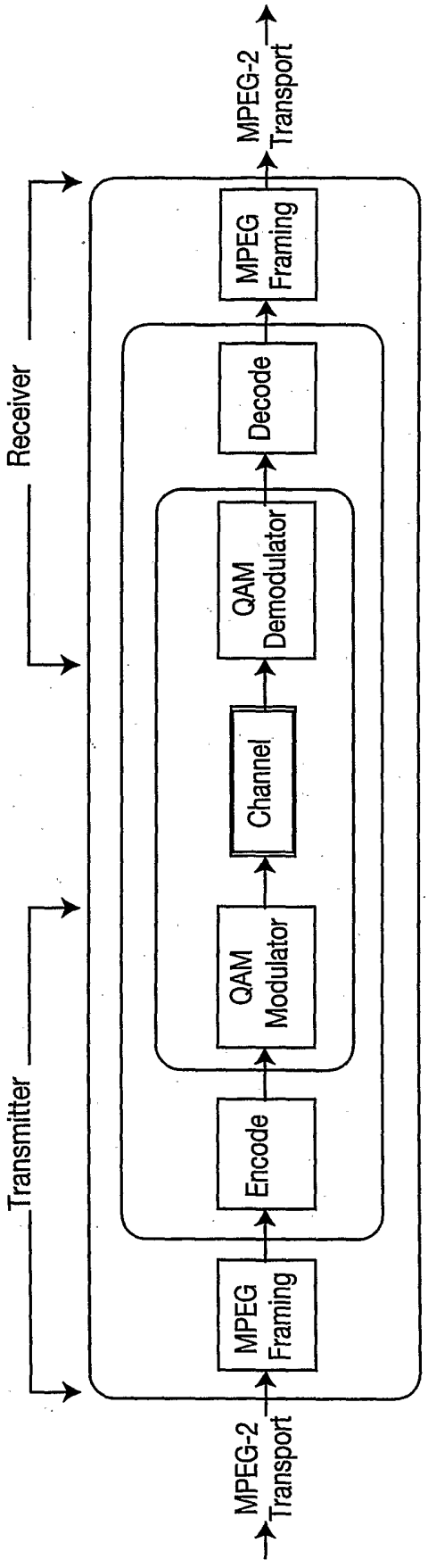

MPEG-2 Transport
MPEG Framing
Encode
QAM Modulator
Channel
QAM Demodulator
Decode
MPEG Framing
MPEG-2 Transport
Transmitter
Receiver

FIG. 1

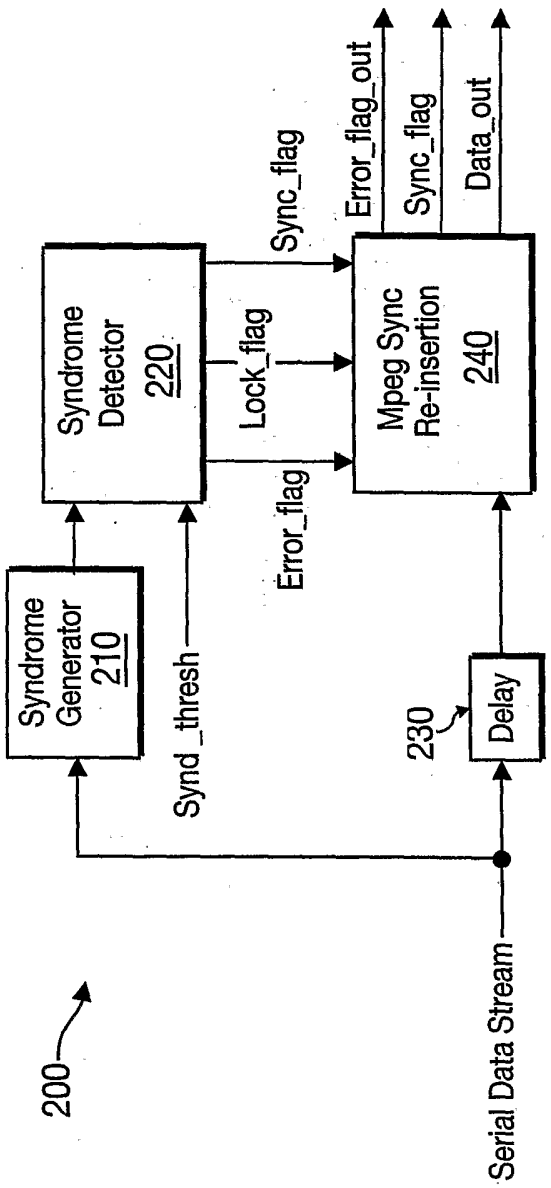

200
Syndrome Generator 210
Syndrome Detector 220
Mpeg Sync Re-insertion 240
Delay
230
Synd_thresh
Error_flag
Lock_flag
Sync_flag
Error_flag_out
Sync_flag
Data_out
Serial Data Stream

FIG. 2

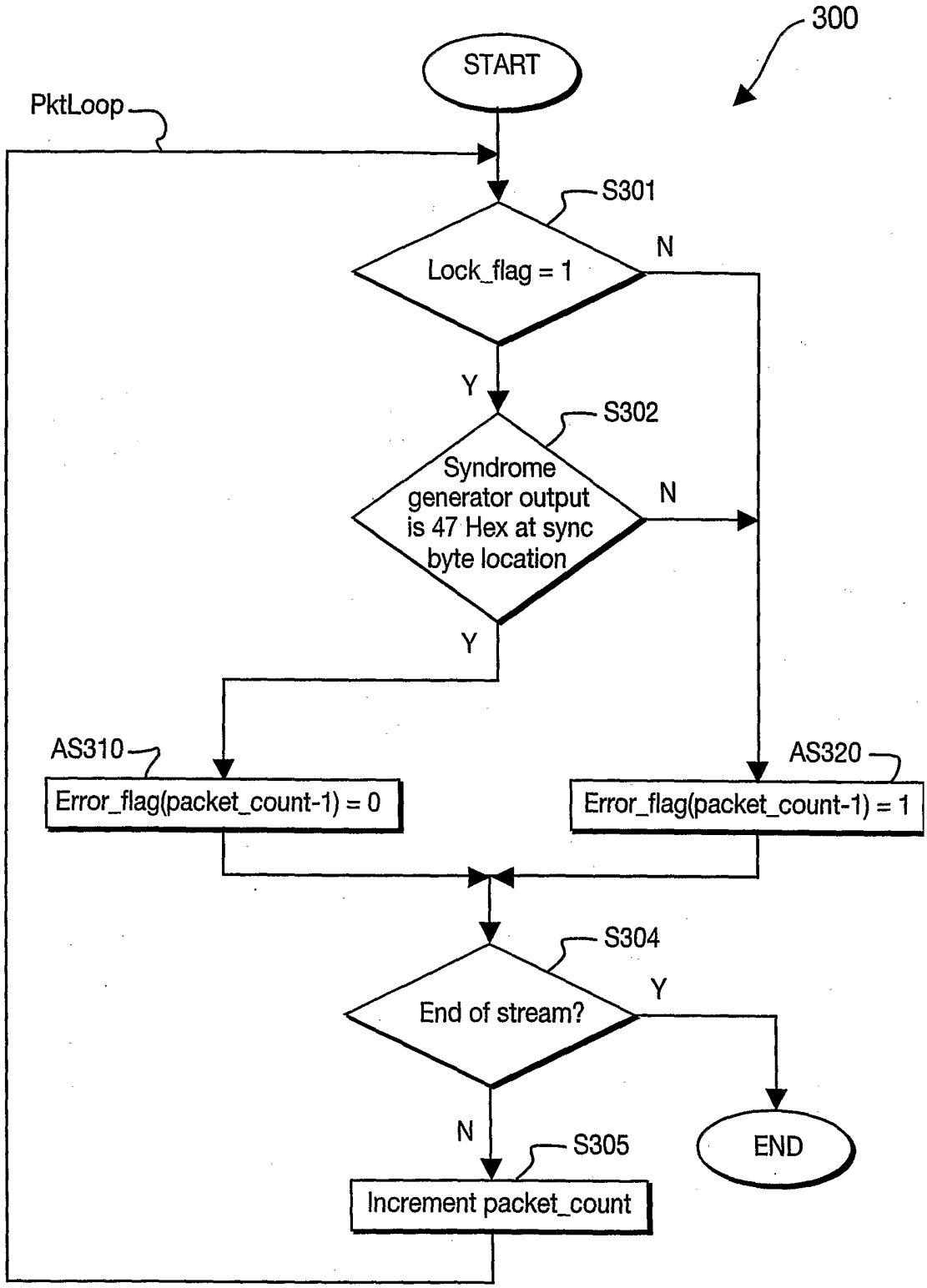

300
START
PktLoop
S301
Lock_flag = 1
N
Y
S302
Syndrome generator output is 47 Hex at sync byte location
N
Y
AS310
Error_flag(packet_count-1) = 0
AS320
Error_flag(packet_count-1) = 1
S304
End of stream?
Y
N
S305
Increment packet_count
END

FIG. 3

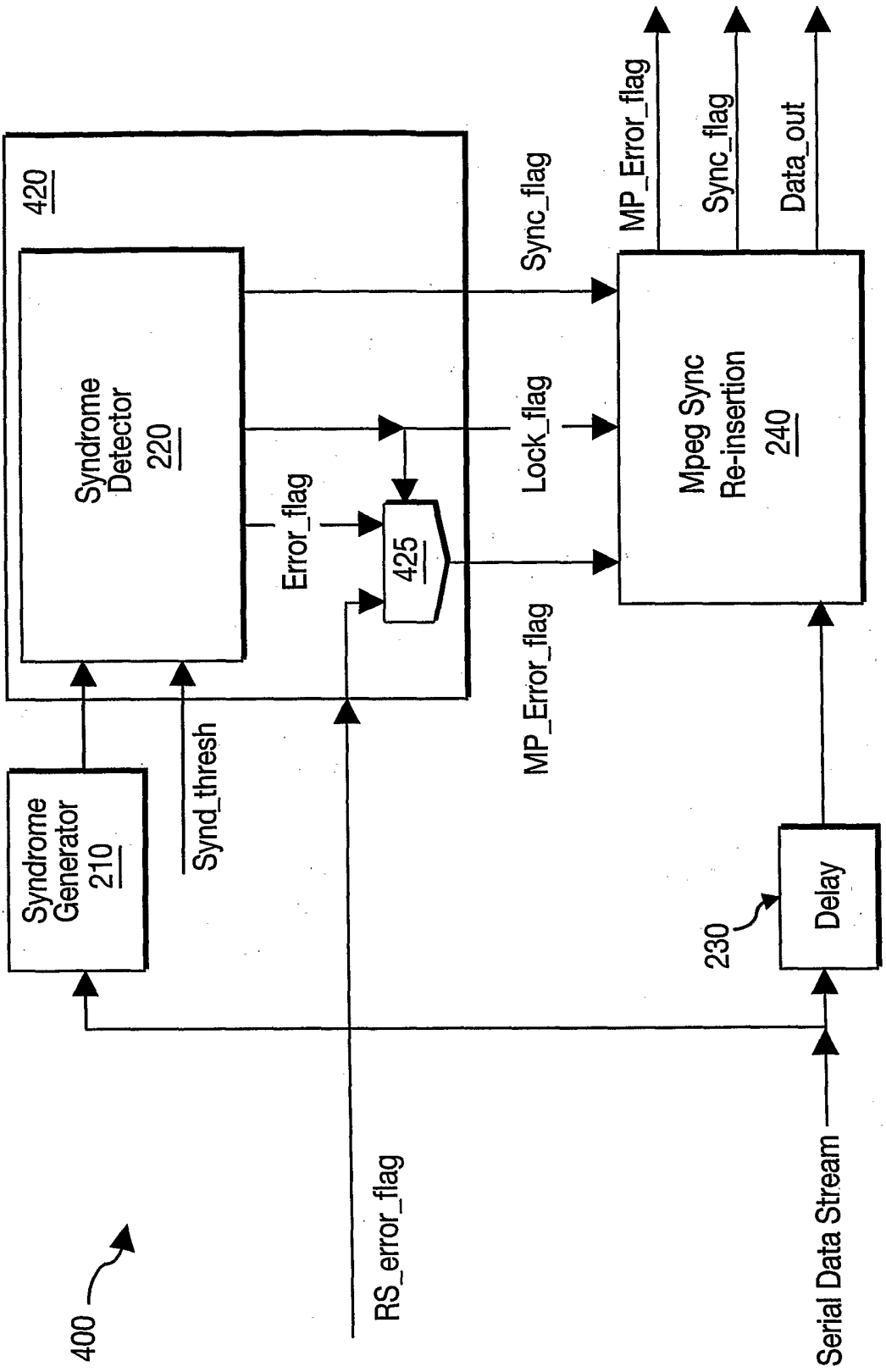

400
420
Syndrome Detector 220
Syndrome Generator 210
Synd_thresh
Error_flag
425
RS_error_flag
Sync_flag
Lock_flag
MP_Error_flag
Mpeg Sync Re-insertion 240
MP_Error_flag
Sync_flag
Data_out
230
Delay
Serial Data Stream

FIG. 4

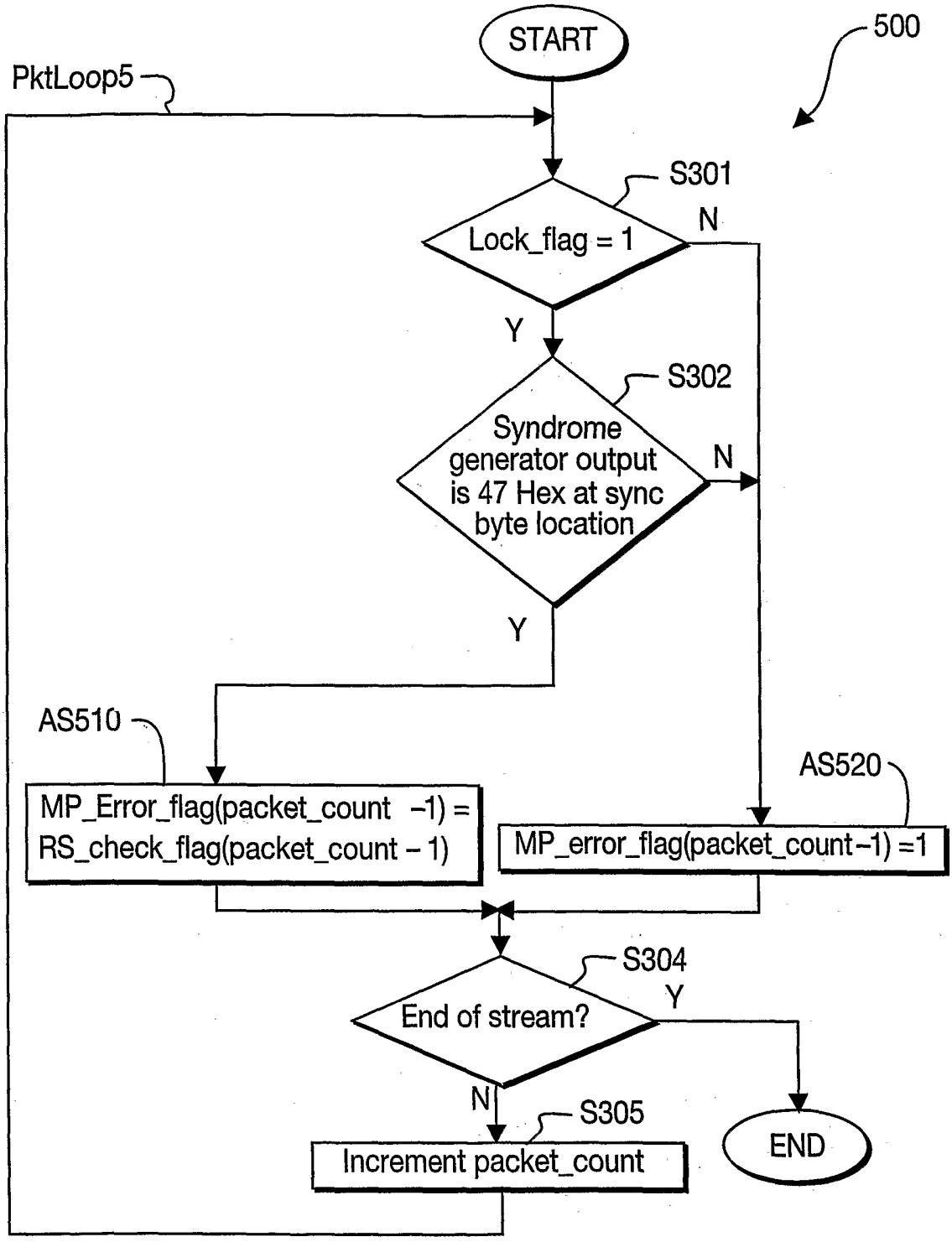

START
500
PktLoop5
S301
Lock_flag = 1
N
Y
S302
Syndrome generator output is 47 Hex at sync byte location
N
Y
AS510
MP_Error_flag(packet_count −1) = RS_check_flag(packet_count − 1)
AS520
MP_error_flag(packet_count−1) =1
S304
End of stream?
Y
N
S305
Increment packet_count
END

FIG. 5

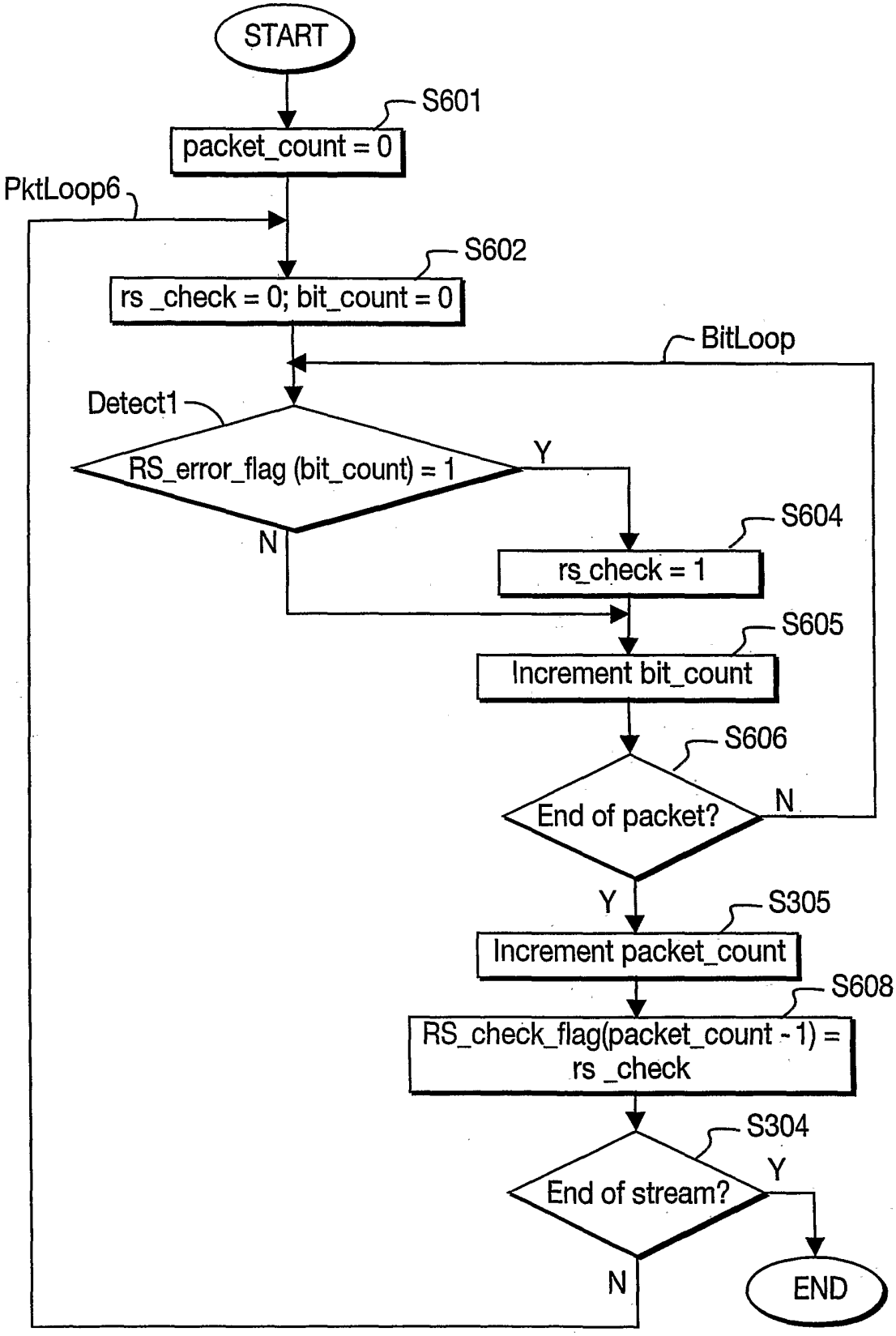

START
S601
packet_count = 0
PktLoop6
S602
rs _check = 0; bit_count = 0
BitLoop
Detect1
RS_error_flag (bit_count) = 1
Y
N
S604
rs_check = 1
S605
Increment bit_count
S606
End of packet?
N
Y
S305
Increment packet_count
S608
RS_check_flag(packet_count - 1) = rs _check
S304
End of stream?
Y
N
END

FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2004/019010

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/64 H03M13/15 H04N7/62

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H03M H04N H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
X	US 5 703 887 A (HEEGARD CHRIS ET AL) 30 December 1997 (1997-12-30) abstract; claim 1; figures 1,5-8 column 2, line 37 - column 3, line 2 column 4, lines 15-40 column 5, lines 13-38 column 8, line 28 - column 9, line 20 column 12, lines 18-67 column 14, lines 28-35 column 15, lines 1-9	1-34
X	EP 1 011 202 A (MATSUSHITA ELECTRIC IND CO LTD) 21 June 2000 (2000-06-21) abstract; claims 1,2; figures 1,2 paragraphs '0012!, '0052! - '0064!	24,32-34

-/--

[X] Further documents are listed in the continuation of box C

[X] Patent family members are listed in annex

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 November 2004	01/12/2004

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax. (+31-70) 340-3016	Cakiroglu, S

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2004/019010

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
X	US 5 910 960 A (CLAYDON ANTHONY PETER J ET AL) 8 June 1999 (1999-06-08) abstract; figures 24,25 column 2, lines 26-41 column 8, lines 41-54 column 19, lines 47-51 column 31, line 30 - column 34, line 18 column 35, lines 19-31 column 36, lines 43-48; table 15	34
A	US 2002/038458 A1 (HAWLEY ROBERT ALLEN ET AL) 28 March 2002 (2002-03-28) abstract; figures 3,5 paragraphs '0013! - '0017!, '0031!, '0039!	1-34
A	US 6 487 692 B1 (MORELOS-ZARAGOZA ROBERT) 26 November 2002 (2002-11-26) abstract; claim 1	1
A	EP 1 235 378 A (SONY CORP) 28 August 2002 (2002-08-28) the whole document	1-34
A	US 2002/087995 A1 (PEDLOW LEO MARK) 4 July 2002 (2002-07-04) the whole document	1-34

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US2004/019010

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5703887 A	30-12-1997	CA 2165604 A1	24-06-1996
		JP 8256336 A	01-10-1996
EP 1011202 A	21-06-2000	CN 1263383 A ,C	16-08-2000
		EP 1011202 A2	21-06-2000
		JP 3290639 B2	10-06-2002
		JP 2000232377 A	22-08-2000
		KR 2000052456 A	25-08-2000
		TW 510086 B	11-11-2002
		US 6553537 B1	22-04-2003
US 5910960 A	08-06-1999	US 5717715 A	10-02-1998
		US 5635864 A	03-06-1997
		US 5692020 A	25-11-1997
		US 5761210 A	02-06-1998
US 2002038458 A1	28-03-2002	NONE	
US 6487692 B1	26-11-2002	NONE	
EP 1235378 A	28-08-2002	JP 2002084261 A	22-03-2002
		EP 1235378 A1	28-08-2002
		WO 0221755 A1	14-03-2002
		US 2003129999 A1	10-07-2003
US 2002087995 A1	04-07-2002	NONE	