

US00RE44466E

(19) **United States**  
(12) **Reissued Patent**  
**van Steenbrugge**

(10) **Patent Number:** **US RE44,466 E**  
(45) **Date of Reissued Patent:** **Aug. 27, 2013**

(54) **METHOD AND DEVICE FOR PACKAGING AUDIO SAMPLES OF A NON-PCM ENCODED AUDIO BITSTREAM INTO A SEQUENCE OF FRAMES**

4,375,083 A	2/1983	Maxemchuk
4,376,874 A	3/1983	Karban et al.
4,425,645 A	1/1984	Weaver et al.
4,429,384 A *	1/1984	Kaplinsky ..... 370/442
4,449,190 A	5/1984	Flanagan et al.
4,539,692 A	9/1985	Munter
4,701,937 A	10/1987	Wan et al.
4,703,477 A	10/1987	Adelmann et al.
4,905,234 A	2/1990	Childress et al.
4,920,534 A	4/1990	Adelmann et al.
5,029,200 A	7/1991	Haas et al.
5,262,964 A	11/1993	Bonsall et al.

(75) Inventor: **Bernard van Steenbrugge**, Eindhoven (NL)

(73) Assignee: **Koninklijke Philips Electronics N.V.**, Eindhoven (NL)

(21) Appl. No.: **10/170,822**

(22) Filed: **Jun. 13, 2002**

**Related U.S. Patent Documents**

Reissue of:

(64) Patent No.: **6,076,062**  
Issued: **Jun. 13, 2000**  
Appl. No.: **08/798,198**  
Filed: **Dec. 9, 1996**

(30) **Foreign Application Priority Data**

Dec. 7, 1995 (EP) ..... 95203376

(51) **Int. Cl.**  
**G10L 19/00** (2013.01)

(52) **U.S. Cl.**  
USPC ..... **704/500; 375/220; 375/240.03; 370/442; 370/468**

(58) **Field of Classification Search**  
USPC ..... 704/500-504, 200, 200.1, 229, 205, 704/203, 201, 212; 370/442, 468; 375/220, 375/216, 240.03; 381/2, 27  
See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

3,909,532 A 9/1975 Rabiner et al.  
4,008,375 A 2/1977 Lanier

**FOREIGN PATENT DOCUMENTS**

DE 4222150 A1 1/1994  
DE 4236989 A1 5/1994

(Continued)

**OTHER PUBLICATIONS**

Van Steenbrugge, Bernard, "The Transfer of Non\_PCM Encoded Audio Bitstreams", Draft Amendment to IEC958, Revised Nov. 6, 1995, pp. 1-25.

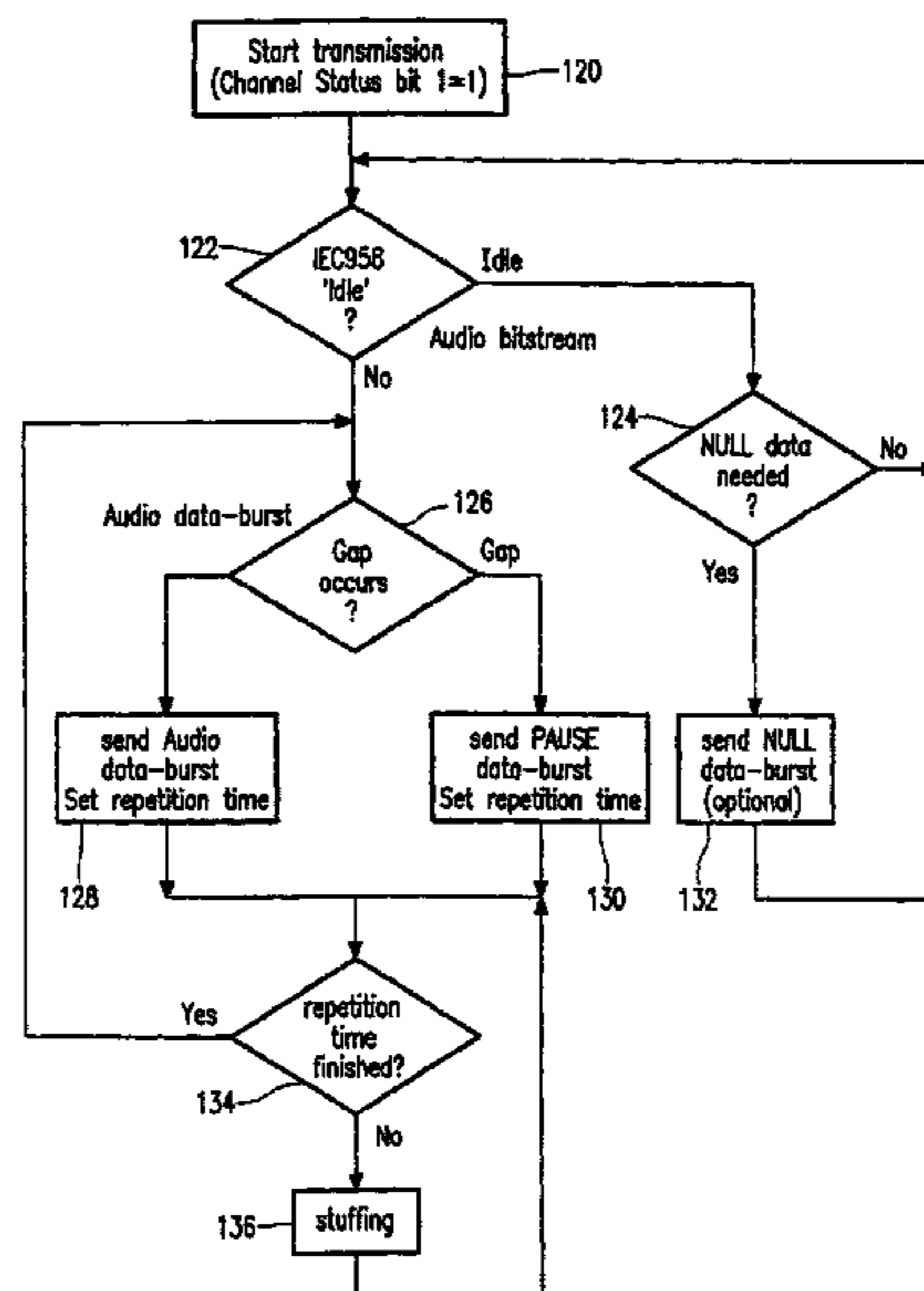
(Continued)

*Primary Examiner* — Vijay B Chawan

(57) **ABSTRACT**

An audio bitstream is read from a digital video disc DVD for transfer, subsequent to parsing thereof, via an IEC 958 protocolled interface, for use in a multi-channel audio reproduction apparatus. For each respective audio channel MPEG audio samples are packaged recurrently in burst payloads, and these burst payloads are packaged as user data in IEC958 format frames. Pause bursts are used for signalling absence of audio for all associated channels with, each pause burst representing such audio absence during a perceptibly acceptable time interval only.

**71 Claims, 7 Drawing Sheets**



(56)

## References Cited

## U.S. PATENT DOCUMENTS

5,297,181	A	3/1994	Barr et al.	
5,323,396	A *	6/1994	Lokhoff	370/468
5,359,626	A *	10/1994	Kloker et al.	375/220
5,451,942	A	9/1995	Beard et al.	
5,461,604	A	10/1995	Hasegawa	
5,530,655	A *	6/1996	Lokhoff et al.	375/240.03
5,539,829	A *	7/1996	Lokhoff et al.	381/2
5,544,247	A *	8/1996	Ten Kate	381/27
5,602,815	A	2/1997	Klappert et al.	
5,606,618	A *	2/1997	Lokhoff et al.	381/2
5,619,197	A	4/1997	Nakamura	
5,675,693	A *	10/1997	Kagoshima	386/68
5,684,786	A	11/1997	Schylander et al.	
5,687,191	A *	11/1997	Lee et al.	375/216
5,706,309	A	1/1998	Eberlein et al.	
5,748,835	A	5/1998	Lee	
5,812,760	A	9/1998	Mendenhall et al.	
5,841,385	A	11/1998	Xie	
5,875,396	A	2/1999	Stockton et al.	
5,987,418	A	11/1999	Gentit	
6,233,562	B1	5/2001	Sueyoshi et al.	
6,501,802	B1	12/2002	Williams	
6,504,994	B2	1/2003	Kawamura et al.	
6,667,950	B2	12/2003	Robers	
7,630,396	B2	12/2009	Goto et al.	
2002/0009044	A1	1/2002	Robers	
2005/0169320	A1	8/2005	Hosokubo	
2006/0143190	A1	6/2006	Haitsma et al.	

## FOREIGN PATENT DOCUMENTS

EP	0402973	A1 *	12/1990
EP	0618730	B1	10/1994
EP	0677932	A1	10/1995
EP	0686973	A1	12/1995
EP	0690630	B1	1/1996
EP	0701374	B1	3/1996
EP	0713628	B1	5/1996
EP	0724263	A2	7/1996
EP	0751520	B1	1/1997
EP	0847196	B1	6/1998
EP	1069780	A2	1/2001
EP	1434218	A1	6/2004
GB	2258102	A	1/1993
GB	2276796	A	10/1994
JP	6061958	A	3/1994
JP	7264144	A	10/1995
WO	WO9512254	*	5/1995
WO	9526597	A1	10/1995

## OTHER PUBLICATIONS

- Van Steenbrugge, Bernard, "The Transfer of Audio Bitstreams via the 'IEC 058' Interface Using the TDA 1315" Advanced Development Centre Broadcast Laboratory, July 21, 1995, pp. 1-10.
- Rault, J.B. et al, "The ISO/MPEG Audio Musicam Family", The Institution of Electrical Engineers, 1995, pp. 1-15.
- "Multichannel Digital Audio Decoding System for Consumer Products", Licensee Information Manual, Version 1.0, Oct. 10, 1995, pp. 1-33.
- Van Steenbrugge, Bernard, "Multi Channel Decoder Functional Specification", Draft, 1995, pp. 1-23.
- Stoll, G., "MPEG Audio Layer II: A Generic Coding Standard for Two and Multichannel Sound DFOR DVB, DAB and Computer Multimedia", International Broadcasting Convention, Conference Publication NNo. 413, Sep. 1995, pp. 1-9.
- "International Standard", CEI IEC 958, Digital Audio Interface, 1989, pp. 1-25.
- Van Steenbrugge, Bernard, "IEC958 to Convey Non\_PCM Audio Bitstreams, Including Amendment to IEC958", Revised Nov. 17, 1995, pp. 1-25.
- Anonymous, "Dynamic Range Compression and/or Expansion During Subband Decoding" Research Disclosure, Jul. 1995, p. 495.
- "AC-3 Data Stream in IEC958 Interface", Digital Audio Compression (AC-3) Standard, Annex B, Oct. 1995, pp. 115-126.
- AES Recommended Practice for Digital Audio Engineering—Serial Multichannel Audio Digital Interface (MADI), Audio Engineering Society Inc. AES 10, 1991, pp. 2-12.
- Hopkins, Digital Terrestrial HDTV for North America: The Grand Alliance HDTV System, IEEE Transactions on Consumer Electronics, vol. 40, No. 3, Aug. 1994, pp. 185-198.
- Digital Audio Input/Output Circuit (DAIO), TDA1315H, Product Specification, Jul. 17, 1995, Philips Semiconductors, 37 pages.
- Vernon, "Design and Implementation of AC-3 Coders", IEEE TR. Consumer Electronics, vol. 41, No. 3, Aug. 1995, pp. 1-6.
- Information Technology—Generic Coding of Moving Pictures and Associated Audio: Audio, ISO/IEC 13818-3, International Standard, 1994, 136 pages.
- Epanorama.net—Audio Documents, [http://www.epanorama.net/documents/audio\\_spdif.html](http://www.epanorama.net/documents/audio_spdif.html), Downloaded Mar. 10, 2003, 12 pages.
- Davis, "The AC-3 Multichannel Coder", Presented at the 95th Convention, Oct. 7-10 1993, pp. 1-6.
- MADI Multi Channel Audio Digital Interface, NTI-Pro Audio, Digilyer, [http://www.nti-instruments.com/x0-asp-plngcateid\\_219-pintlevel\\_4-plngpageid\\_661-x1-default.htm](http://www.nti-instruments.com/x0-asp-plngcateid_219-pintlevel_4-plngpageid_661-x1-default.htm), Downloaded Mar. 11, 2003, 2 pages.

\* cited by examiner



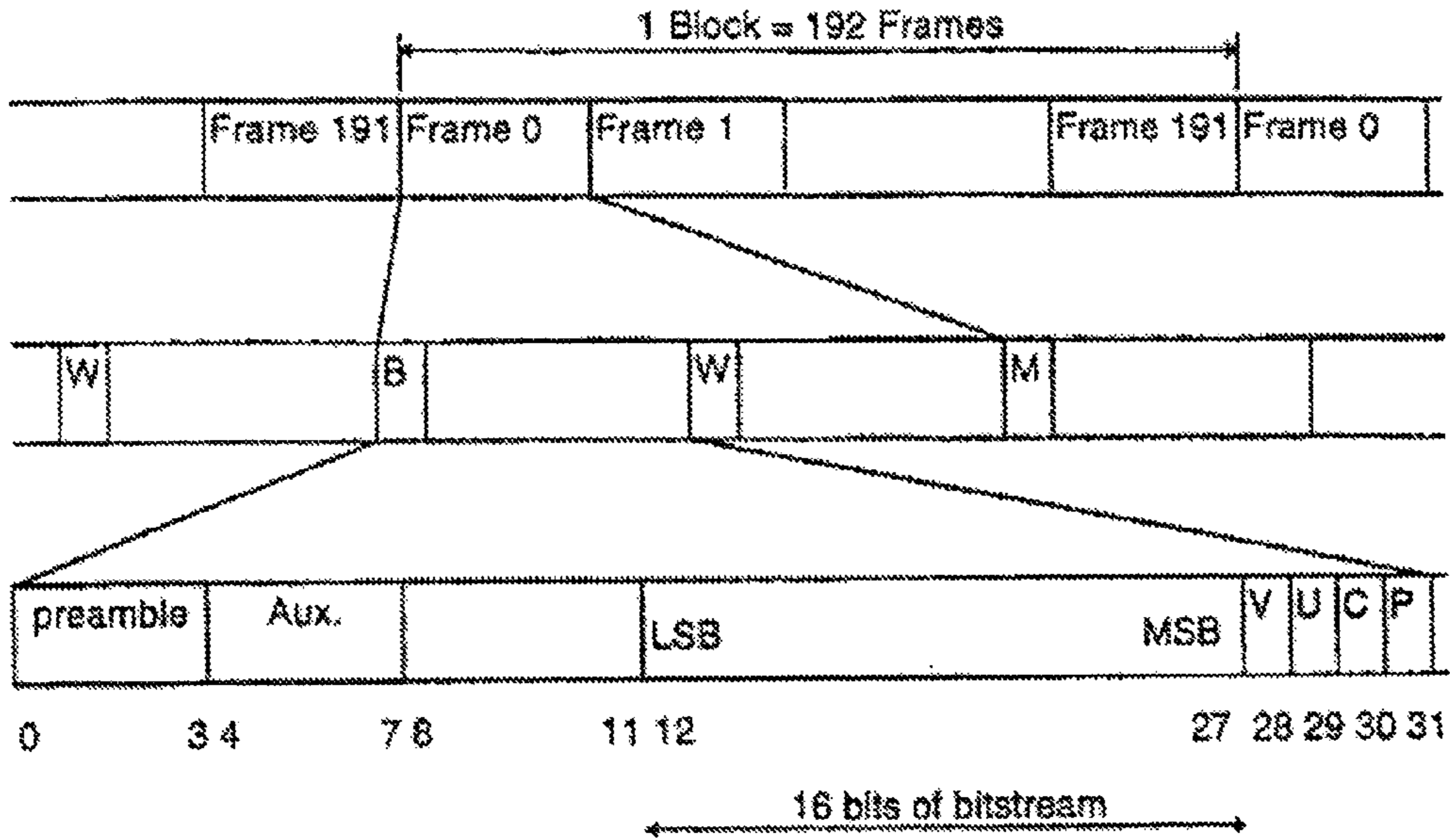


FIG. 1 (AMENDED)

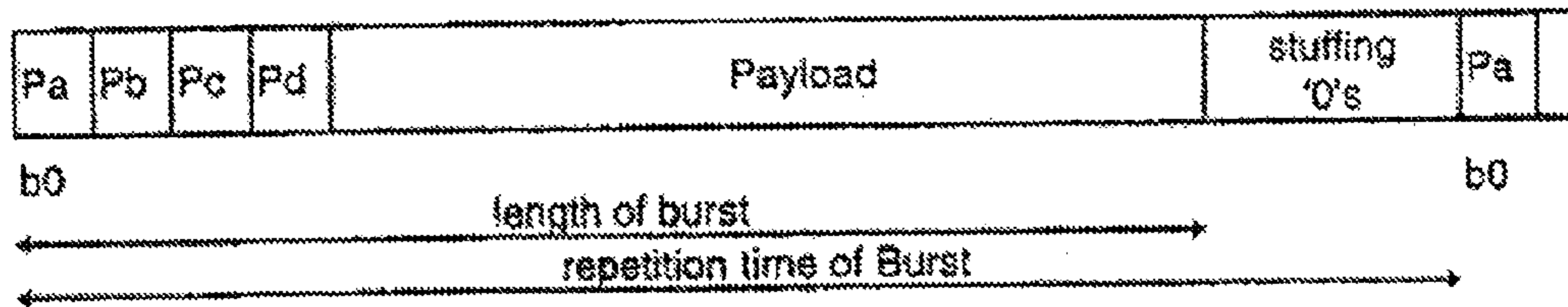


FIG. 2 (AMENDED)

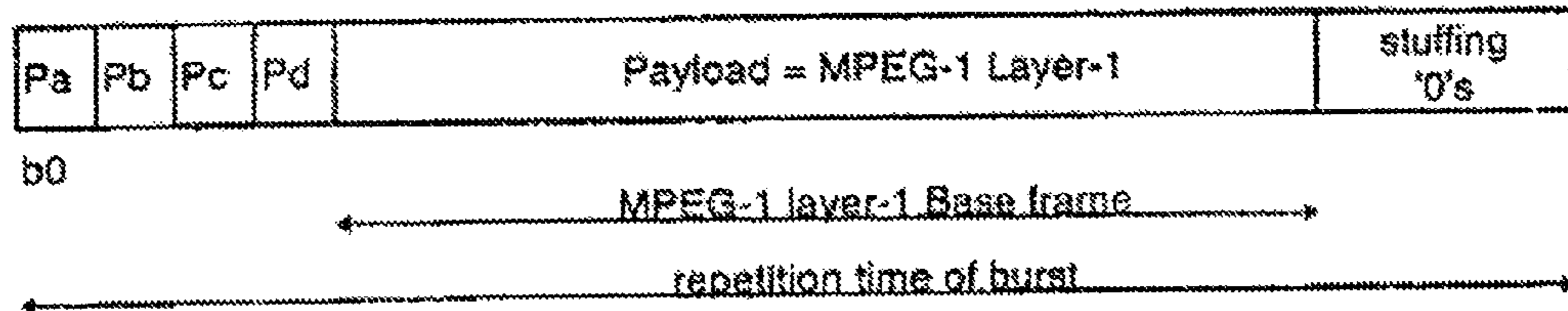
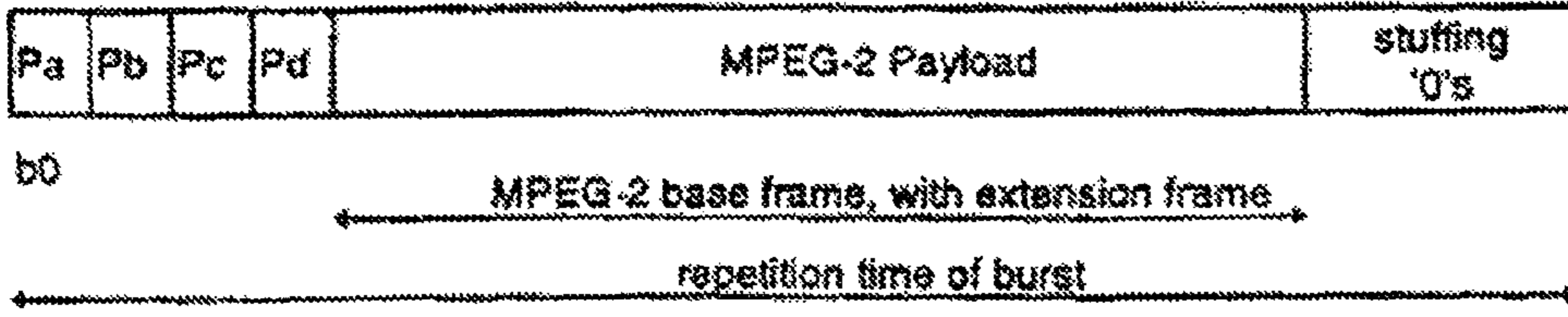
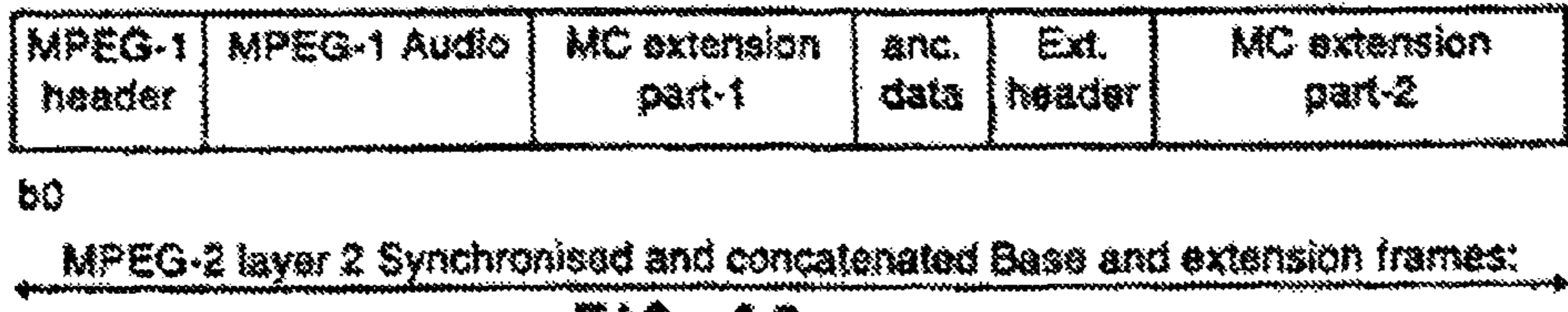
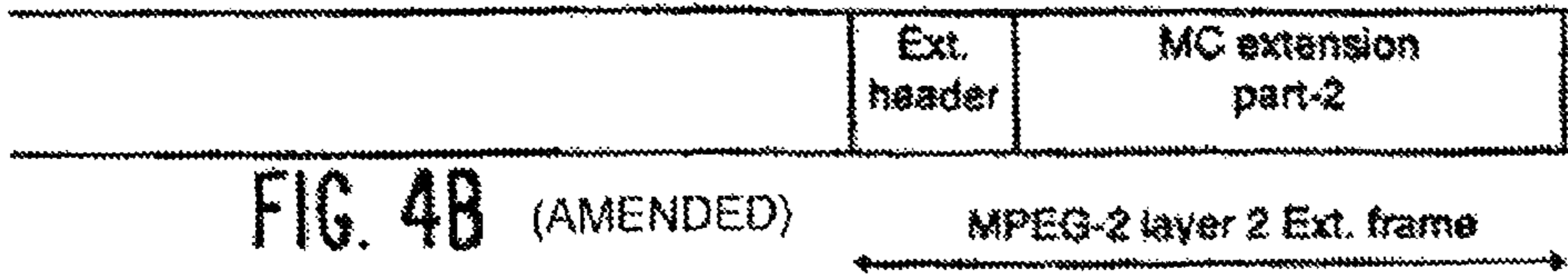
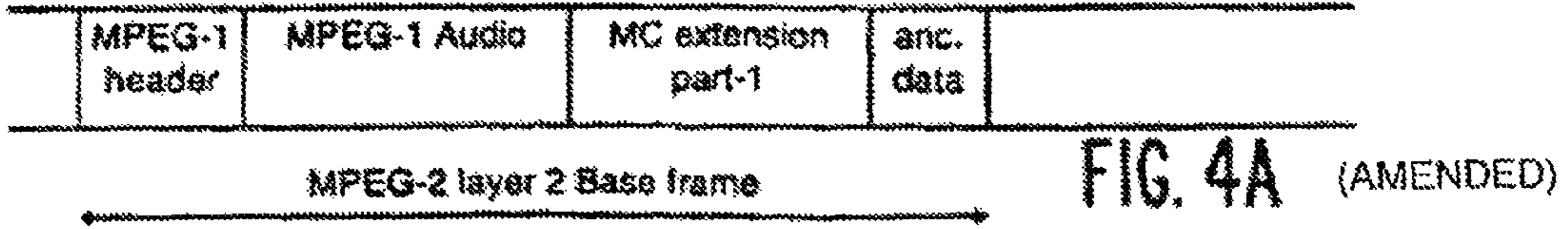
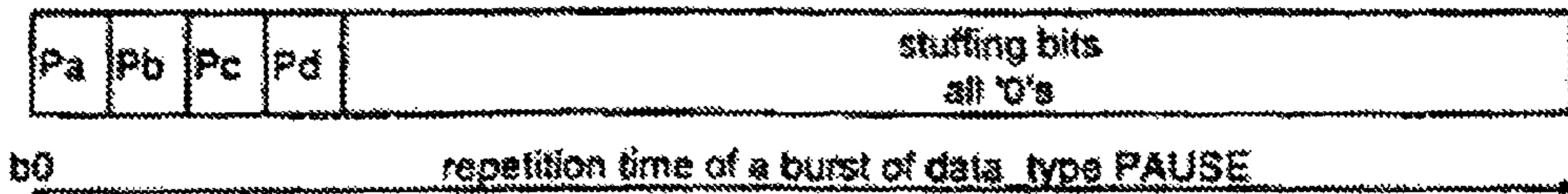


FIG. 3 (AMENDED)



**FIG. 4D** (AMENDED)



**FIG. 5** (AMENDED)

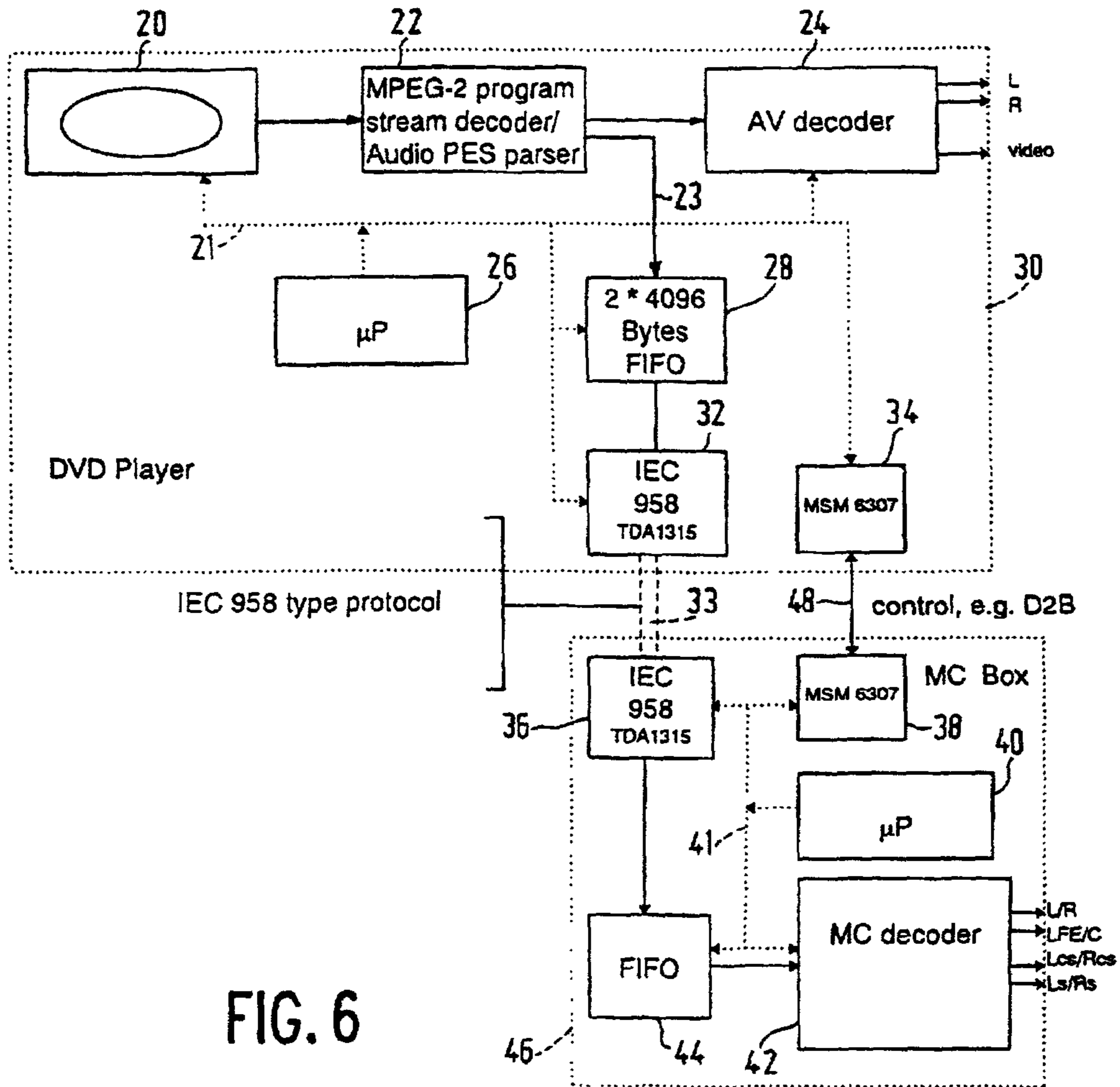


FIG. 6

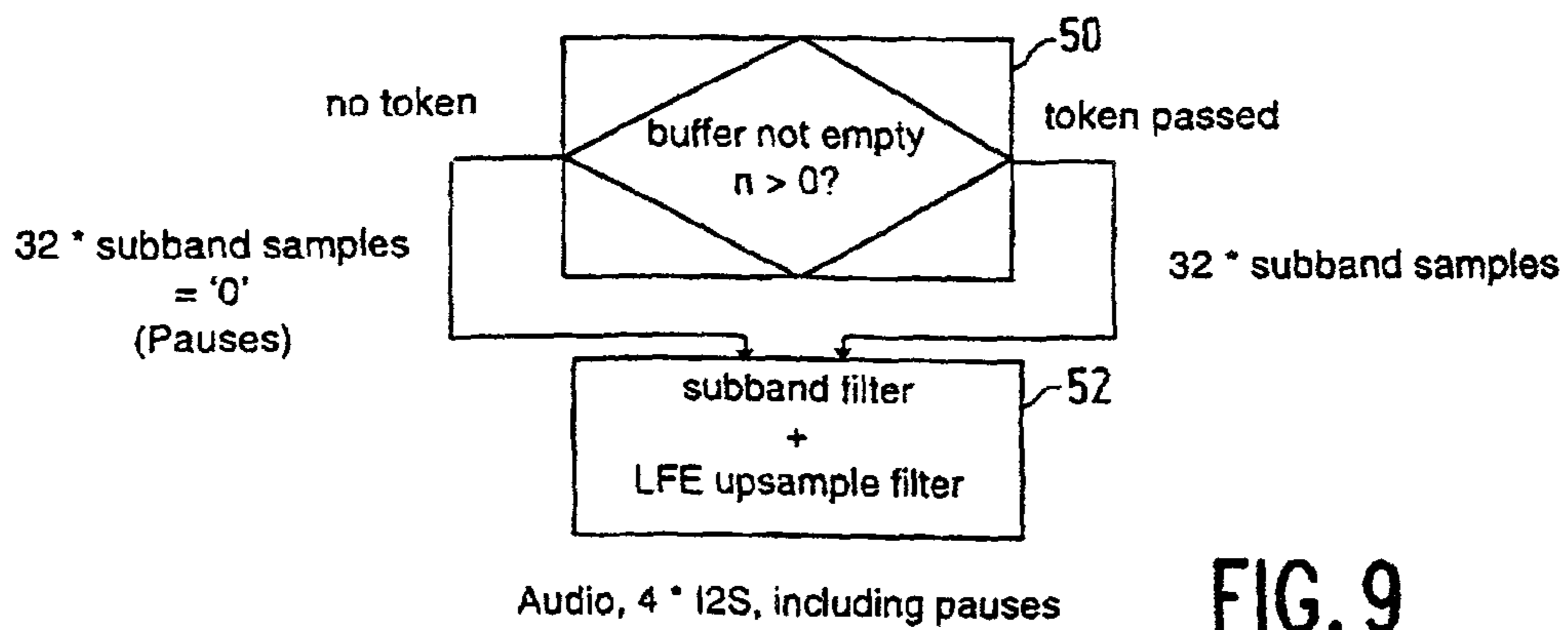


FIG. 9

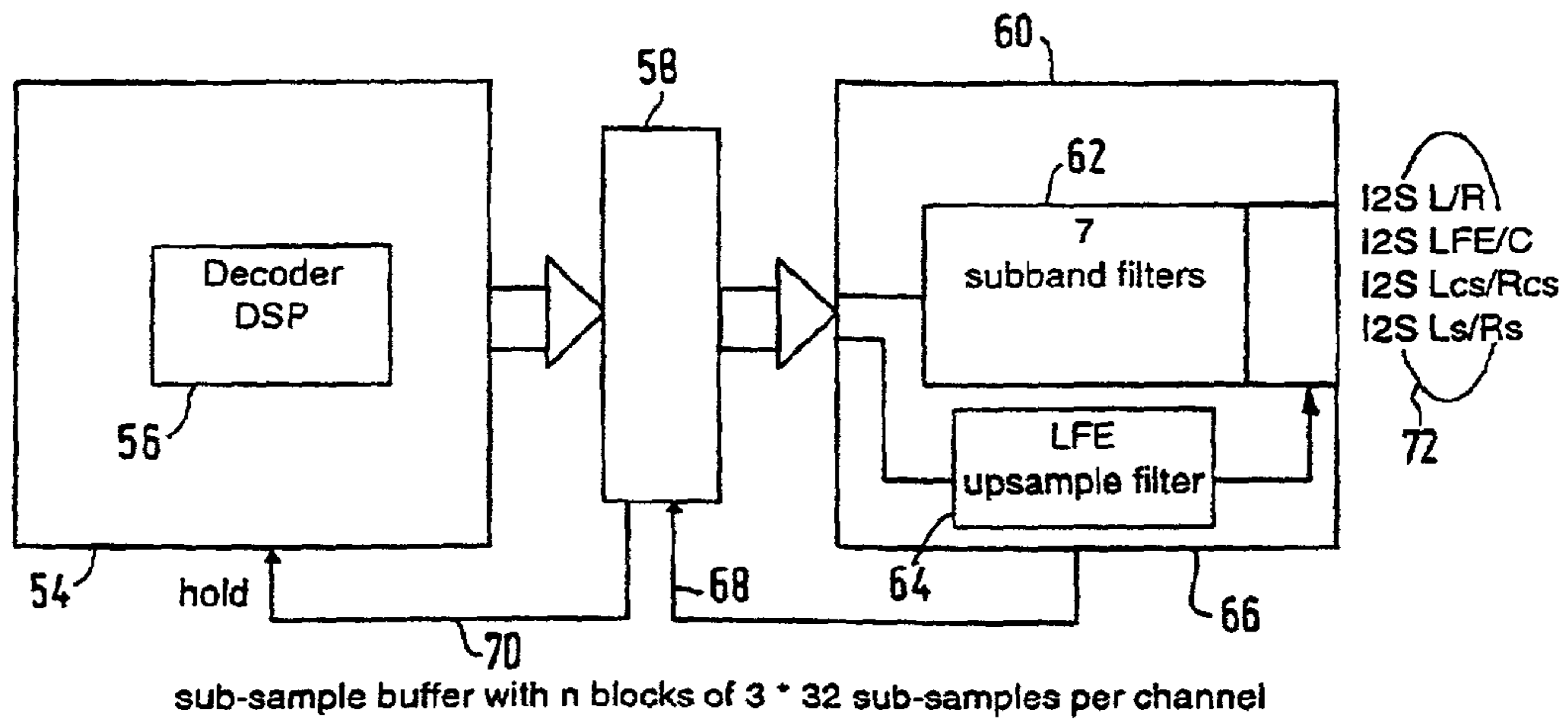


FIG. 7

7 \* subband filter.  
26.12/36 msec per Cycle

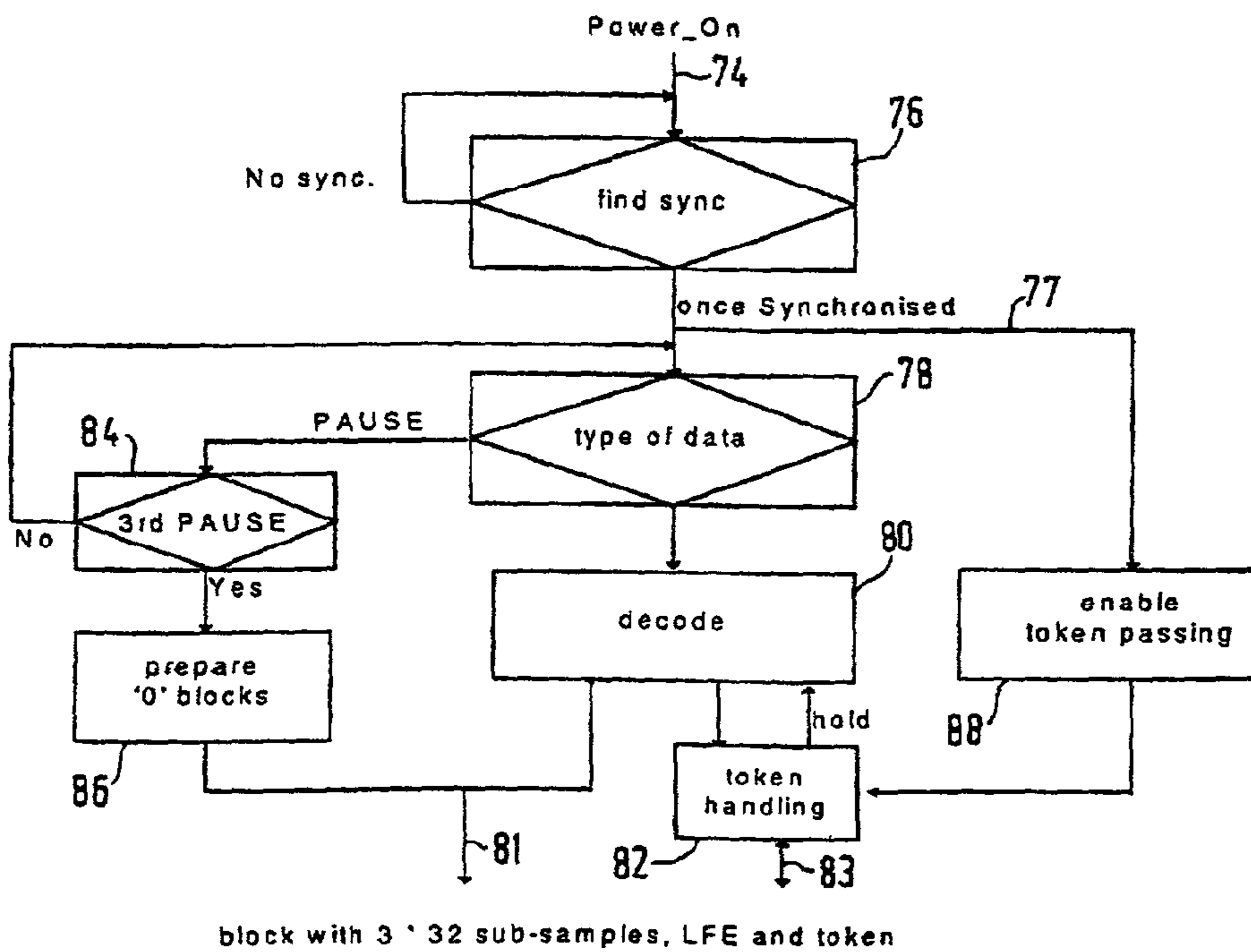


FIG. 8



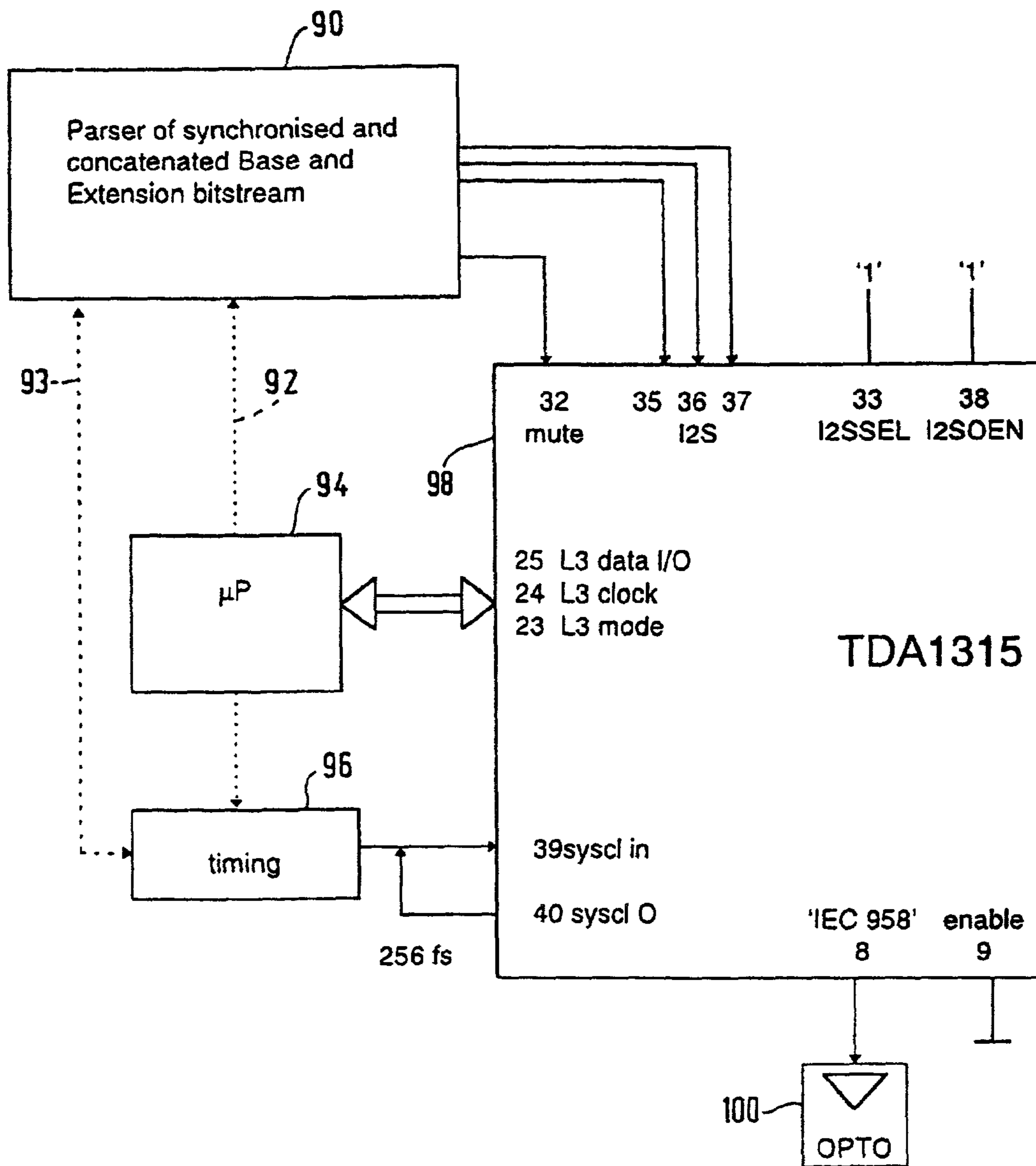


FIG. 10

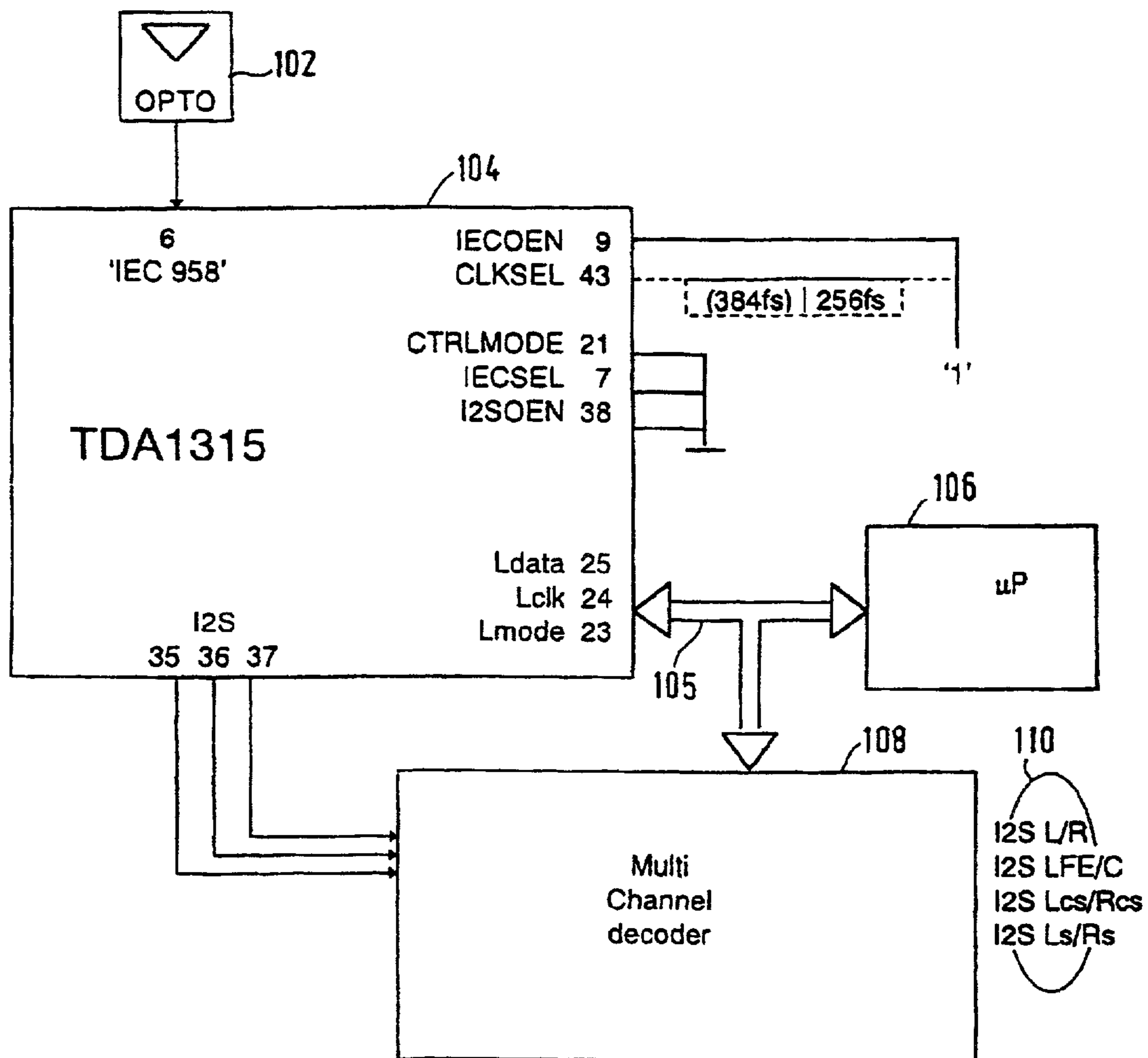


FIG. 11



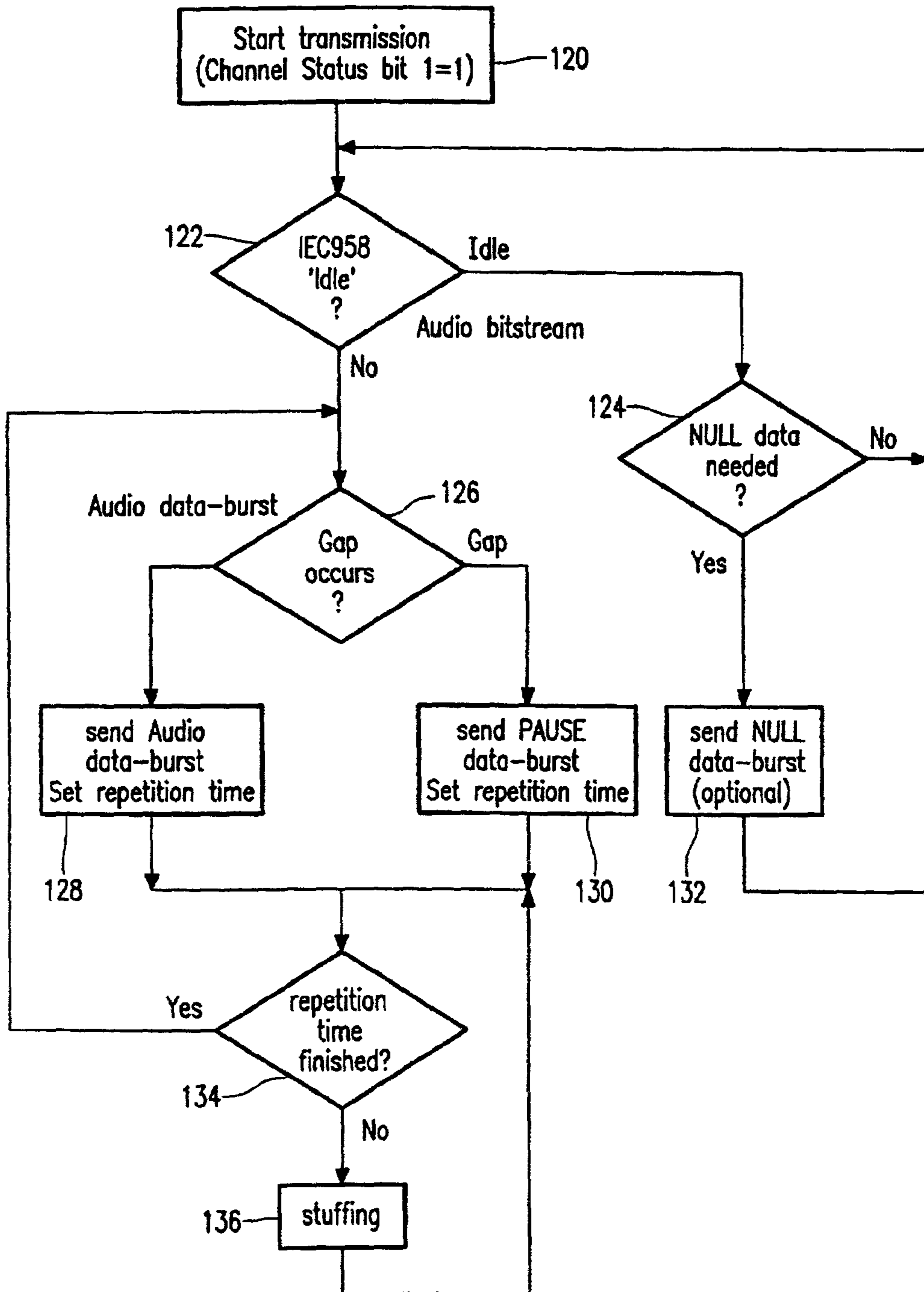


FIG. 12

**METHOD AND DEVICE FOR PACKAGING  
AUDIO SAMPLES OF A NON-PCM ENCODED  
AUDIO BITSTREAM INTO A SEQUENCE OF  
FRAMES**

**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.**

BACKGROUND OF THE INVENTION

The invention relates to a method of transferring a non-PCM encoded audio bitstream read from a digital medium, subsequent to parsing thereof, via an IEC 958 protocolled interface to a multi-channel audio reproduction apparatus.

Digital video disc standardizing is proceeding at an accelerated pace. Commercially available MPEG1 decoder circuit SAA2502 is able to decode compressed digital audio received as a continuous bit stream. Present-day MPEG2 technology has standardized 5 channels, to wit: Left, Right, Center, Left Surround, Right Surround, and furthermore a low frequency enhancement (LFE) channel. The MPEG2 bit stream is distributed into frames of 1152 samples for each of the actual channels, and player operation is controllable in a non-uniform manner on a frame-to-frame basis. For example, the number of actual channels may vary, and certain ones or all of them may be outputting silence.

General background to the invention is given by the following earlier documents, all being at least co-assigned to the present assignee and being herein incorporated by reference:

EP Patent 402 973, EP Patent Application 660 540, corresponding U.S. Pat. No. 5,323,396, issued Jun. 21, 1994, which is a continuation of U.S. application Ser. No. 07/532,462, abandoned; U.S. Pat. No. 5,606,618, issued Feb. 25, 1997; U.S. Pat. No. 5,530,655, issued Jun. 25, 1996; U.S. Pat. No. 5,539,829, issued Jul. 23, 1996; U.S. Pat. No. 5,777,992, issued Jul. 7, 1998; and pending U.S. application Ser. No. 08/488,536, describing a Musicam Layer 1 encoder and decoder for L and R signals;

EP 678 226, corresponding to U.S. Pat. No. 5,544,247, issued Aug. 6, 1996, describing encoding and decoding of L, R and C channels;

U.S. patent application Ser. Nos. 08/032,915, 08/180,004, 08/427,046, describing the matrixing of bitrate-reduced L, R, C, SL and SR signals.

Now, in a consumer application (SPDIF) of the above specified digital video disc, two subframes are specified that each can simultaneously carry 32 bit data words. This allows to transfer via the IEC 958 bitstream either 2-channel linear PCM audio, or a set of alternating bitstreams, but not those configurations simultaneously. The IEC 958 standard specifies a widely used method for interconnecting digital audio equipment with 2-channel linear PCM audio. A need has been encountered to allow transferring non-PcM encoded audio bitstreams for consumer applications in the same protocolled environment, and in particular pause bursts, in case one or more of the audio channels would represent silence. In particular, the granularity of such pause representation at the receiver side should be sufficiently brief from a perceptive standpoint.

SUMMARY OF THE INVENTION

Accordingly, amongst other things, it is an object of the present invention to extend present protocols to allow transfer of non-PCM encoded audio bitstreams for consumer applications in the same protocolled environment.

This and other objects, features and advantages according to the present invention are provided by a method for decoding a non-PCM encoded audio bit stream, which can be read from a digital medium. Preferably, the method includes the steps, executed for each respective audio channel, of recurrently packaging MPEG audio samples in burst payloads, and packaging the burst payloads as user data in IEC958 format frames, including pause bursts which signal the absence of audio for all associated channels. According to one aspect of the present invention, each pause burst represents such audio absence during a perceptively acceptable time interval only.

According to the state of the art, the above granularity could reach several tens of milliseconds, which the present inventor has found unacceptably long. According to the invention, the granularity is in the millisecond range which is acceptable in all circumstances encountered at present.

These and other objects, features and advantages according to the present invention are proved by a method for receiving a non-PCM encoded audio bitstream emanating from a parsed bitstream read from a digital versatile disc DVD via an IEC 958 protocolled interface for use in a multi-channel reproduction apparatus. Advantageously, the method includes steps for receiving the parsed bitstream as a sequence of frames each accommodating, for each applicable bitstream, a uniform number of data bits, storing each frame in an intermediate frame buffer; detecting presence or absence of data pertaining to a particular output channel. Moreover, in response to the detecting step, executing decoding and outputting decoded information for the particular channel. However, under control of one or more pause bursts received, as representing a sequential multiplicity of absence of the detecting, controlling a soft mute block. It will be appreciated that this feature provides for straightforward decoding of the pause bursts for subsequent representation by a soft mute block.

These and other objects, features and advantages according to the present invention are provided by a device for implementing the method described immediately above, either on the encoding or the decoding side. Further advantageous aspects of the invention are set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

These and further aspects and advantages of the invention will be discussed more in detail hereinafter with reference to the disclosure of preferred embodiments, and more in particular with reference to the following drawings, in which:

FIGS. 1-3, 4A, 4B, 4C, 4D and 5, show various information formats;

FIG. 6, a block diagram of connected DVD player and MC\_Box;

FIG. 7, a block diagram of a multi-channel audio decoder;

FIG. 8, a decoding flow chart of a digital signal processor;

FIG. 9, a decoding flow chart of a subband filter DSP;

FIG. 10, a block diagram of an IEC 958 transmitter station;

FIG. 11, a block diagram of an IEC 958 receiver station;

FIG. 12, a flow chart of bit stream transmission.



## DESCRIPTION OF THE PREFERRED EMBODIMENTS

## 1. Data Formats

For better detailing the invention, first various applicable information formats are described. FIG. 1 shows the IEC958 format, which according to the upper diagram consists of a concatenation of frames grouped into blocks of 192 frames each. The second diagram shows each frame to consist of two sub-frames. The first frame of a block has subframes labelled B (left) and W (right), further subframes are all labelled M. The third diagram shows the setup of a subframe. As shown, it has four preamble bits, four auxiliary bits, four unused bits, sixteen data bits or bitstream, and four flag bits V, U, C, P. The flag bits indicate the following:

- V indicates no deviation from the standard;
- U indicates user data with '0' default;
- C contains one bit of a channel status word, and
- P is a parity bit relating to bits 4 through 31.

A pair of subframes may contain one PCM word from each of left and right channels.

According to the present invention, for consumer applications, the channel status word as built from a sequence of C bits has the following meaning: bit bO with value 0 indicates consumer PCM audio, bit bi with value 1 indicates Non-linear PCM samples, bits 8 through 15 contain a category code. Furthermore, the MPEG header indicates audio sample rate and sample size in bits.

Now, audio bit streams as read from the DVD disc may contain gaps, that may be due to pauses in the audio, or to a trick mode of the related video source, such as the transition to a freeze picture produced in track mode. Now, during transfer in bursts on the IEC 958, these gaps in the bit stream may remain unused or rather, filled with bursts of the data\_type 'pause' to be described hereinafter. If the gap occurs in MPEG1 layer 1, or in MPEG 1 layer 2 or 3 data, or in MPEG2 without extension, or in MPEG2 data with extension audio bitstream, the gap will be filled with a sequence of bursts of data\_type 'pause'. These bursts may have the minimum allowable length therefor, corresponding to 32 sample periods. Preferably, this number is three times as long, so corresponding to 96 sample periods, which is the recurrency of LFE samples.

It is possible on this interface to simultaneously convey multiple multi-channel non\_PCM encoded data streams, for example relating to both a main audio service and to an associated audio service. In that case, the burst of the associated service occurs before the burst of the main service to which it is associated.

FIG. 2 describes the burst format attained by unpacking the user information of an IEC 958 block in a standard manner, or rather before the packing of an IEC 958 block. The burst has a fixed repetition time related to the number of audio samples for each channel encoded within that frame. Any unused bit between two bursts is set to zero. Each burst has four sixteen-bit preamble words, with the following meanings: Pa, Pb sync words, Pc burst info to be specified hereinafter, Pd length of payload in bits. Subsequently, the burst contains a payload field and is optionally terminated by stuffing zeroes to attain its pre-specified format. An advantageous, but not mandatory lower bound for the number of stuffing zeroes is 32. The payload also contains the MPEG header. The format of non-PCM encoded Audio bitstreams allows multiplexed conveying of more than one bitstream, wherein a burst can fill the space of stuffed zeroes from other bursts. The sampling frequency must be uniform across the bursts. The field Pc has the following codes:

Pc Bits	value	content	repetition of burst in number of sample periods
0-4	0	Null data	@4096
	1	AC-3 stream	1536
	2	SMPTE time stamp	
	3	MPEG1 layer 1 data	384
	4	MPEG1 layer 2/3 or MPEG 2 without extension	1152
	5	MPEG2 with extension	1152
	6	PAUSE	32 or 96
	7	ACX date	1024
	8	MPEG2 layer 1 low sample rate	384
	9	MPEG2 layer 2/3 low rate	1152
10-15	10-31	Reserved	

The content of further bits of Pc is irrelevant to the present invention. The provision of the relatively brief 'pause' burst allows a low granularity size of 'soft mute' intervals controlled thereby. The indication of the various burst type specifications by Pc bit values 3, 4, 5, 8, 9, allows an extremely flexible control policy.

FIG. 3 shows an MPEG1 layer 1 base frame that has a length of 384 sample periods (each of L and R). The various aspects of the format have been considered supra. The base frame for the payload of the MPEG1 layer 2/3 or MPEG2 without extension has the same shape, be it with a length of 1152 instead of 384 sample periods. MPEG2 allows transfer of five audio channels in parallel. In certain circumstances, the MPEG2 burst needs an extension that has been shown in FIGS. 4B and 4C.

Now, an MPEG2 frame comprises 1152 samples for each encoded channel. The burst as shown in the uppermost row, is headed by a burst\_preamble, followed by the payload, and stuffed with stuffing zero bits. The payload numbers up to 36768=1152\*32 bits. Furthermore, there are at least 32 stuffing zeroes and 64 bits for the Pa..Pd header. Bitstreams matching the MPEG layer 2 data type are:

- either encoded according to MPEG2 layer 2 or 3,
- or even encoded according to MPEG2 layer 1 'super-frame'.

A burst with an Audio frame consists of a synchronized and concatenated Base frame (MPEG1 compatible) and an extension frame. FIG. 4A shows the MPEG2 layer 2 base frame, with MPEG1 header, MPEGi Audio field, MC (multichannel) extension part i field, and a field for ancillary data. If extension is necessary, additional format according to FIG. 4B is appended that contains in MPEG2 layer 2 Extension frame an Extension header followed by the MC extension part 2. FIG. 4C of this shows the formats of FIGS. 4A and 4B both synchronized and concatenated. Likewise, the FIG. 4D shows the MPEG2 base frame, with extension frame as fractional payload inside the burst repetition time represented by the lower arrow, that in its turn needs a preamble and allows for trailed stuffing zeroes to the frame.

Now, by using units of 32 sample periods per subband filler, synchronization is maintained. In this respect, FIG. 5 shows a burst of data type 'PAUSE' inside its burst repetition time indicated by the bottom arrow. The length is 1024 bits=32 IEC frames, increased with the stuffing zeroes. As earlier, the four indications Pa, Pb, Pc, Pd are preappended. The user content is all zero. Another and preferred size of the burst allows for 3\*32=96 frames. The burst frame has of course a dummy content; the longer version allows better synchronization to the LFE feature that occurs every 96



frames. Due to the relatively small size of pause bursts, the transition between pause and non-pause has a small granularity size.

## 2. Hardware Embodiments

FIG. 6 is a block diagram of an interconnected DVD player 30 and an MCBox 46. In player 30, block 20 symbolizes a turn-table and associated read-out and feedback mechanisms, associated control signals being transferred by means of control path 21. Control processing is in microprocessor 26. Block 22 is an MPEG2 program stream decoder and audio parser that separates the massive bitstream received into standard stereo audio and video streams on the one hand to go to audio-video decoder 24, and furthermore, multi-channel bitrate reduced audio data on channel 23. Audio-Video decoder 24 operates in a standard manner for separating the bit stream into left and right audio channels and video as indicated. In fact, this type of reproduction is conform to the MPEG1 standardization. Relatively low-level consumer applications would do with the system as described thus far. DVD player 30 is implemented with a user control interface, such as hard buttons, soft keys, display.

For attaining full functionality of MPEG2, an external multichannel MC\_box 46 has been provided. To this effect, first in player 30, the MPEG data is configured according to the burst format described with respect to the earlier Figures. Next, this requires an output channel 33 for data according to the standardized IEC 958 protocol, and which is used to convey a non-PCM bitstream inclusive of various commands for the MC\_box. The channel may be based on galvanic interconnection or optical fibre. Optionally, interconnection is by a uni- or bidirectional channel 48, in particular for transmitting commands to the DVD player. The channel may be protocolled according to D2B described in U.S. Pat. No. 4,429,384 to the present assignee. Moreover as shown, a FIFO 28 is provided that by way of example accommodates 8 k Bytes as generally required for intermediate storage of MPEG data, a bus interface circuit 32 of commercially available type TDA1315, and a control interface circuit 34 of type MSM6307, organized according to the D2B protocol. Alternatively, block 32 receives commands from the microprocessor 26 on the data path, rather than on its control path.

Like the DVD player, MC\_box 46 has an internal control path 41, interface circuit 38 of type MSM6307, and control processing in microprocessor 40. In correspondence to FIFO 28, the MC\_box 46 has a relatively small FIFO 44. This stores the data of one bitstream while the previous one is decoded locally. The decoding pertains first to the burst level, and next to the sample level. The output from FIFO 44 feeds MC\_decoder 42 that may output up to seven audio channels as indicated: Left, Right, LFE/C, Left center surround, Right center surround, Left surround and Right surround. As shown, these are grouped on four I2S interfaces, according to a protocol described in U.S. Pat. No. 4,755,817 to the present assignee. Alternatively FIFO 44 plus decoder 42 are combined into a single hardware block and controlled directly by the commands contained in the IEC 958 data. Moreover, the MC\_Box attaches to the secondary control channel 48 by means of circuit 38.

FIG. 7 is a block diagram of a multi-channel audio decoder, as contained in block 42 in FIG. 11. First, the decoding proper is executed in block 56 according to the process described with reference to FIG. 8, and implemented with a Motorola DSP processor of the 56000 series architecture. Also, the dematrixing is executed in this processor. Block 54 symbolizes a control shell to the processor in question. The output of the first DSP processor is organized in blocks that each contain for each appropriate channel  $3 \times 32 = 96$  subsamples. For

such channel at the highest applicable sample frequency of 44.1 kHz, the block length corresponds to an interval of 2 msec, which is considered a sufficiently fine granularity to be practically imperceptible.

Block 58 is an intermediate buffer that can hold n blocks as specified supra, optimized as regards to cost versus allowable occurrence of over/underflow; expected value of n is about four. Line 70 transmits a stop/go signal to DSP shell 54 that functions as source; line 68 transmits a request signal from the data destination block 60.

Block 60 executes the demultiplexing function with respect to the maximum of seven channels received; it is based on a similar Motorola 56000 DSP processor. In particular, block 62 symbolizes the subband filtering, whereas block 64 symbolizes an LFE upsample filter. Again, the processor shell has been indicated by block 66. During each execution cycle, 32 subchannels per channel are filtered, and unloaded by means of a dual port RAM: the length of a cycle is thus for a sample frequency of 44.1 kHz:  $32/44, 1k = 0.725$  millisecc. The delay length of the RAM for each channel is advantageously equal to  $3 \times 32$  subsamples. Filtering takes place when  $3 \times 32$  subsamples have been received, otherwise the subband filter will output all zeroes signalling an audio pause, which thus has a reduced granularity with respect to prior art. The processor will contain a 'free running' function, and will continually output audio samples at uniform intervals. The first DSP 56 will continually produce audio samples in bursts of 1152 samples per channel, 12 groups of  $3 \times 32$  samples each for each and every channel. The real-time demand is on subband filter 62. If applicable, decoder 56/54 is put on "hold" to avoid an overflow of buffer 58.

The MC Box does not have a user control interface, but the data received on the IEC unidirectional interconnect 33 are used for effecting control, inclusive of the soft-mute and concealing feature according to the invention. If required, the D2B interconnect allows for sending control signals in the reverse direction. The multichannel decoder 60 can be controlled by decoder 54, such as by means of an I2C interface as disclosed in U.S. Pat. No. 4,689,740. This will be robust enough to recover from error conditions. However, no status output to a user is deemed necessary. If underflow occurs in buffer 58, the soft mute feature is controlled subsequently.

FIG. 8 shows a decoding flow chart of a processor, in particular digital signal processor 54 in FIG. 7. The received input bitstream is symbolized by 74, on which the decoder continually undertakes a synchronizing operation 76. Actual decoding starts in block 78 as soon as DSP 54 synchronizes and a next concatenated frame through the synchronizing words Pa, Pb is received. Frame item Pd gives the length of the payload. When synchronized, decoder 54 for each frame produces twelve groups of  $3 \times 32$  sub-samples for each channel. Decoder 56 is put on hold when the free area in the sub-sample buffer is not sufficient to store all sub-samples of that group (of  $3 \times 32$ ) sub-samples for each channel to avoid buffer overflow.

The hand-shake between the sub-sample buffer and decoder DSP 56 is implemented by a token that indicates the current owner of the block in question; this token is transferred when synchronization has been effected (77, 88). In the flowchart, block 78 detects either audio data, or a pause. Unless third pause, data type detecting is continued (78). Upon meeting a non-pause, decoding is continued in block 80, and the decoding result is outputted on line 81, subject to the reception of a blocking token to put the processor on hold via block 82. The handshake is between blocks 80 and 82. Bidirectional connection 83 allows reacting to the filling degree of buffer 58. When the third pause is received (84),



block **86** prepares zero output blocks for outputting on line **81** as an alternative to the decoding results from block **80**, to function as 'soft mute' information.

FIG. **8** is a decoding flow chart of subband DSP filter **62** in FIG. **7**. Each cycle, the subband DSP receives 32 subsamples per channel at its input; if no subsamples are available, the input will become all zeroes as a soft-mute. The subband filter DSP processes blocks of 32 sub-samples and produces 7 out of 8 signals for the four I2S interfaces shown in the Figure. The eighth signal LFE will be upsampled by block **64**. The filter operates according to a continuous process, producing audio at equidistant time intervals. After power-on, all outputs are muted by default; the output registers will contain zeroes until the subband filter is initialised after 512 sample periods.

In the implementation, block **50** detects whether the buffer **58** is not empty. If empty, zeroes are output in such a way as to maintain synchronization. If not empty and a token has been passed, the block output is at right, and 32 samples are outputted for each actual channel, plus a single LFE sample. If not empty and no token has been passed, the block output is at left, and 32 zero subband samples are outputted to emulate a pause. Both outputs from block **50** lead to the input of subband filter **62** and LFE upsample filter **64** in FIG. **7**.

The token indicates which processor is currently the owner of the block. An owner has read/write access to a block, non-owners can only read, such as read the token. Block ownership is only passed along by the owner of the block, render the actual owner to non-owner. After power-on, all tokens will be handed to the decoder DSP. Absent a token, the subband filter will clear all registers and will filter exclusively zeroes. When synchronizing on the Burst\_Preamble, the first token shall be passed to the subband filter DSP after an expected 'worst case' decoding time.

FIG. **10** is a block diagram of an IEC 958 transmitter station, of which a central part is the commercially available TDA 1315 circuit **98** interconnected as shown. Block **90** symbolizes the parser of synchronized and concatenated Base and Extension bitstreams (block **22** in FIG. **11**). Microprocessor **92** corresponds to microprocessor **26** in FIG. **11**. Microprocessor **92** interacts with interface circuit **98** along a three-wire L3 control bus protocolled according to U.S. Pat. No. 5,434,862 assigned to the present assignee, and connected to pins **23**, **24**, **25** as shown. The data output from block **90** is protocolled according to the I2S format and connected on pins **35**, **36**, **37** as shown. Input **32** receives a mute control signal from parser **90**, pin **33** an I2S selection signal and pin **38** an I2S output enable signal, these two continuously at logic 1. Timing control block **96** is controlled by microprocessor **92**, and handshakes along interconnection **93** with parser block **90**. Also, it handshakes on a sync cycle basis with TDA 1315 on pins **39**, **40**. Finally, the circuit outputs serial data according to IEC 958 on pin **8**, and receives an enable signal on pin **9** at a continuously low value. Block **100** is an electric to optical converter, allowing a remote position of the MC\_Box.

FIG. **11** is a block diagram of an IEC 958 receiver station. Data is received as 16 bit words over optical-to-electric converter **102** and transferred to IEC 958 input pin **6**. Standard control pins are ConTRLMODE at pin **21**, IECSElection pin **7**, and I2SOutputENable pin **38**, all three held at logic ground. Furthermore, there are IECOEoutputenable pin **9** and CLIOck SElection pin **43**, both held at logic high (1). The clock selection allows to select between 384 kHz and 256 kHz. The data output from TDA 1315 is by means of an I2S protocol on pins **35**, **36**, **37** to multichannel decoder **108**. This produces the four output bit streams as deemed in FIG. **11**.

Control interconnection between TDA 1315, the microprocessor (item **40** in FIG. **11**) and multi-channel decoder **108** is protocolled according to the I2S protocol referred supra.

FIG. **12** is a flow chart of bitstream transmission. In block **120**, transmission is started. Channel status bit **1** becomes "1". In block **122** IEC 958 "Idle" is detected. If "Idle", in block **124** it is detected whether NULL data is needed. If "No", the system reverts to block **122**. If "Yes", in block **132** a NULL data burst is sent; the latter feature is optional. If in block **122** an Audio bitstream is detected, in block **126** it is detected whether a Gap occurs. If "Gap", in block **120** a PAUSE data burst is sent. Also, the repetition time is set. If in block **126**, an Audio data-burst is detected, an audio data burst is sent. Also, the repetition time is set. Both after block **128** and after block **130**, in block **134** it is detected whether the repetition time has finished. If No, in block **136** "stuffing" is executed, and the system reverts to block **134**. If in block **134**, the repetition time has finished, the system reverts to block **126**.

PAUSE data-bursts are intended to fill small discontinuities in the bitstream, the gaps which may occur between two data-bursts of a non-PCM encoded audio data type. PAUSE data-bursts convey information of the audio decoder that a gap exists. The PAUSE data-bursts may also indicate the actual length of the audio gap, or that the non-PCM audio data stream has stopped. This information may be used by the audio decoder to minimise (or conceal) the existence of the audio gap, or in the case that the bitstream stops, to trigger a fade-out of the audio. A sequence of PAUSE data-bursts can also assist decoder synchronization prior to the beginning of a non-PCM audio bitstream. It is recommended to send a short sequence of PAUSE data-bursts immediately preceding the transmission of the first audio data-burst.

Data burst	stuff-ing	P	stuff-ing	P	Stuff-ing	P	stuff-ing	Data burst	stuff-ing	Data burst
R		R		R		R		R		R

In this example, P indicates a PAUSE burst, P+subsequent stuffing represents the repetition time of PAUSE, and the total gap in between the data bursts is three times as long. The length of Data burst+stuffing is the repetition time of the burst. The PAUSE burst is transferred with the same bit stream number as the bit stream number of the audio data stream which contains the gap that the PAUSE data-bursts are filling, or for which synchronization is being assisted.

The PAUSE data-burst contains the burst\_preamble and a 32-bit payload. The first 16 bits of the payload contain the audio\_gap\_length parameter. The remaining bits are reserved, and should all be set to '0'. The audio\_gap\_length parameter is an optional indication of the actual audio gap length. This is the length, measured in IEC958 frames, between the first bit of Pa of the first PAUSE data-burst and the first bit of Pa of the next Audio data-burst. The detailed use of the PAUSE data-burst depends on the data-type of the Audio data-burst. For example, gaps between AC-3 data bursts are filled with a sequence of very short PAUSE bursts, and the repetition time of PAUSE data-bursts between data-bursts of an MPEG type is related to the algorithm. The gap\_length parameter of the first PAUSE data-burst of the sequence may be used to indicate the length of the audio gap which is being bridged by the sequence of PAUSE data-bursts. The PAUSE data-bursts in the sequence which follows the initial PAUSE data-burst do not have a gap\_length specified (gap\_length=0). A gap may be filled with one single sequence of PAUSE data-bursts with a single indication of audio\_gap\_length. For example, a gap



corresponding to an audio gap of 768 samples may be filled with one sequence of PAUSE data-bursts with an indication of gap-length=768 in the initial PAUSE data-burst. Or this gap could be filled with a number of smaller sequences of PAUSE data-bursts, with the initial PAUSE data-burst in each sequence indicating the gap\_length bridged by that sequence (E.g. one sequence with a gap-length of 200 samples, followed by a sequence with gap-length of 568, together bridging a gap of 768 sample periods).

The information about the full length of the audio gap in the first PAUSE data-burst will allow the decoder to perform the best concealment. However, if the data source does not have the information about the full audio gap length at the time the gap begins, then it may signal an initial value for gap\_length. If the data source then determines that the audio gap will be longer than the initial indication, then another sequence of PAUSE data-bursts may be initiated (following the first sequence by the repetition time) with another gap\_length value to signal the decoder that the audio gap is being extended. If the gap is further extended, additional sequences may be initiated.

Audio decoders may use the gap\_length information to optimise their concealment of the audio gap. The inclusion of non-zero values of gap\_length is optional, data sources are not required to indicate the length of the audio gap.

The data type PAUSE has a sequence of four control words Pa, Pb, Pc, Pd followed by the Payload and the Stuffing.

“Gaps” are discontinuities in a bitstreams, and may be due to switching between bitstreams. The length of the gaps depend on the timing of switching from one to the other bitstream, and may have any value. The length of a gap however, depends on the decoder which must be able to conceal the gaps. Therefore the transmitter shall adjust the length of the gap to a multiple of the repetition time of PAUSE data-bursts. The PAUSE data-burst has its repetition time, which gives the time of transmission of Pa of the next data-burst.

Some AC-3 decoders may be capable of “concealing” audio gaps. The indication of the audio gap length (gap\_length) which may be included in the payload of the PAUSE data-burst allows the decoder to know how long an audio gap will need to be concealed, and thus allow the decoder to optimise the concealment process for the actual audio gap length. AC-3 decoders will most easily conceal audio gaps which have a length equal to an integral multiple of 256 samples. Thus audio gaps of length 256, 768, etc IEC958 frames are preferred, as follows:

data-type of Audio data-burst	repetition time of PAUSE data-burst	
	mandatory	recommended
AC-3 data		3 IEC958 frames
MPEG-1 Layer 1 data	32 IEC958 frames	
MPEG-1 Layer 2 or 3 data or MPEG-2 without extension	32 IEC958 frames	
MPEG-2 data with extension	32 IEC958 frames	
MPEG-2, layer-1 Low sample rate	64 IEC958 frames	
MPEG-2, layer-2 or 3 Low sample rate	64 IEC958 frames	

The AC-3 bitstream consists of a sequence of AC-3-frames. The data-type of a AC 3 data-burst is 01 h. An AC-3 frame contains 1536 samples for each encoded channel. The data-burst is headed with a burst\_preamble, followed by the burst\_payload. The burst-payload of each data-burst of AC-3 data shall contain one complete AC-3-frame. The length of the

AC-3-data-burst will depend on the encoded bit rate (which determines the AC-3-frame length). An AC-3 data burst with reference instant R comprises again four control words Pa, Pb, Pc, Pd an AC-3 burst\_payload, and stuffing.

I claim:

1. A method of transferring a non-PCM encoded multichannel bitstream to digital audio reproduction apparatus via an interface which is adapted for transfer of PCM encoded bitstreams, the non-PCM encoded multichannel bitstream resulting from parsing a multichannel bitstream received from a digital video disc (DVD), comprising the steps of:

- (i) recurrently packaging audio samples of the non-PCM encoded multichannel bitstream in data burst payloads;
- (ii) packing the data burst payloads as user data in successive frames having a predetermined format including pause bursts signifying the absence of audio in all channels, each pause burst having a duration in the range below 10 milliseconds; and
- (iii) decoding and supplying the packaged data burst payloads of the respective channels to respective channel inputs of said audio reproduction apparatus.

2. A method as claimed in claim 1, wherein each pause burst has a repetition interval of 32 or a multiple of 32 of the periods of said audio samples.

3. A method as claimed in claim 1, wherein the presence of a pause burst is indicated by a burst header in the non-PCM encoded multichannel bitstream.

4. A method as claimed in claim 1, wherein each pause burst has a repetition period not over one-fifth that of any non-pause burst in the non-PCM encoded multichannel bitstream.

5. A method as claimed in claim 1, wherein the data bursts have a repetition period of 384 periods of the audio samples.

6. A method as claimed in claim 1, further comprising adding stuffing zeros to the data burst payloads in order to maintain a uniform size burst format, the bursts being multiplexed among the multiple channels in a recurrent manner, relevant data from other channels being superposed on stuffing zero positions in channels having stuffing zeros.

7. A coupling device for transferring a non-PCM encoded multichannel bitstream to digital audio reproduction apparatus via an interface which is adapted for transfer of PCM encoded bitstreams, the non-PCM encoded multichannel bitstream resulting from parsing a multichannel bitstream read from a digital video disc (DVD); comprising:

means for recurrently packaging audio samples of the non-PCM encoded multichannel bitstream in data burst payloads;

means for packing the data burst payloads as user data in successive frames having a predetermined format including pause bursts signifying the absence of audio in all channels, each pause burst having a duration in the millisecond range below 10 milliseconds; and

means for decoding and supplying the packaged data burst payloads of the respective channels to respective channel inputs of said audio reproduction apparatus.

8. A method of transferring a non-PCM encoded multichannel bitstream to digital audio reproduction apparatus via an interface which is adapted for transfer of PCM encoded bitstreams, the non-PCM encoded multichannel bitstream resulting from parsing a multichannel bitstream read from a digital video disc (DVD), the parsed bitstream being received



## 11

as a sequence of frames which accumulates, for each channel, a uniform number of data bits, comprising the steps of:

- storing each frame in an intermediate frame buffer;
- detecting the presence of data pertaining to a particular output channel or the absence of such data as represented by one or more sequential pause bursts;
- upon detecting the presence of said data in said detecting step, decoding and outputting decoded information for the particular channel; and
- upon detecting one or more sequential pause bursts in the detecting step, controlling a soft mute block to conceal the absence of said data.

9. A coupling device for transferring a non-PCM encoded multichannel bitstream to digital audio reproduction apparatus via an interface which is adapted for transfer of PCM encoded bitstreams, the non-PCM encoded multichannel bitstream resulting from parsing a multichannel bitstream read from a digital video disc (DVD), comprising:

- means for receiving said parsed bitstream as a sequence of frames which each accommodate, for each channel, a uniform number of data bits;
- an intermediate frame buffer for storing each frame;
- detection means for detecting the presence of data pertaining to a particular output channel or the absence of such data as represented by one or more sequential pause bursts;
- decoding means controlled by said detection means for decoding and outputting from a stored frame decoded information for said particular channel; and
- means responsive to detection by said detection means of said one or more sequential pause bursts to control an audio soft mute block of said one or more sequential pause bursts to conceal the absence of data in said mute block.

10. A method of packaging audio samples of a non-PCM encoded audio bitstream into a sequence of IEC958 format frames, the method comprising acts of:

- recurrently packaging audio samples of a plurality of audio channels of a non-PCM encoded audio bitstream into burst payloads;
- packaging by a processor the burst payloads into IEC958 format frames, resulting in a sequence of IEC958 format frames, the packaging including:
  - generating pause bursts indicating absence of audio data in one of the audio channels, the pause bursts representing such audio absence for a predetermined time interval; and
  - inserting the pause bursts into the sequence of IEC958 format frames; and
- outputting the sequence of IEC958 format frames including the generated pause bursts.

11. The method of claim 10, wherein the audio samples are MPEG audio samples, and the pause bursts have a repetition time interval that is equal to at most one MPEG LFE sample recurrency time interval.

12. The method of claim 11, wherein the pause bursts have a repetition time interval equal to 32 audio sample periods or an integer multiple of 32 audio sample periods.

13. The method of claim 12, wherein the presence of the pause burst is indicated in a header of the burst.

14. The method of claim 12, wherein the pause burst has a repetition time that is at least five times shorter than the repetition time of any non-pause burst.

15. The method of claim 12, wherein the bursts include MPEG-1 layer data bursts with a repetition time interval of 384 audio sample periods.

## 12

16. The method of claim 12, wherein the bursts include MPEG-1 layer 2 or 3, or MPEG-2 without extension bursts with a repetition time interval of 1152 audio sample periods.

17. The method of claim 12, wherein the bursts include MPEG-2 extension bursts with a repetition time interval of 1152 audio sample periods.

18. The method of claim 12, wherein the bursts include MPEG2 layer 1 low sample rate bursts with a repetition time interval of 384 audio sample periods.

19. The method of claim 12, wherein the bursts include MPEG2 layer 2 or 3 low sample rate bursts with a repetition time interval of 1152 audio sample periods.

20. The method of claim 12, comprising acts of:
- adding stuffing zeroes for maintaining a uniform size among all the bursts;
  - multiplexing the bursts among the channels in a recurrent manner; and
  - superposing relevant data when stuffing a burst format at a uniform size among all bursts.

21. The method of claim 10, wherein the presence of the pause burst is indicated in a header of the burst.

22. The method of claim 10, wherein the pause burst has a repetition time that is at least five times shorter than the repetition time of any non-pause burst.

23. The method of claim 10, wherein the bursts include MPEG-1 layer data bursts with a repetition time interval of 384 audio sample periods.

24. The method of 23, wherein the presence of the pause burst is indicated in a header of the burst.

25. The method of claim 23, wherein the pause burst as a repetition time that is at least five times shorter than the repetition time of non-pause burst.

26. The method of claim 10, wherein the bursts include MPEG-1 layer 2 or 3, or MPEG-2 without extension bursts with a repetition time interval of 1152 audio sample periods.

27. The method of claim 10, wherein the bursts include MPEG-2 extension bursts with a repetition time interval of 1152 audio sample periods.

28. The method of claim 10, wherein the bursts include MPEG2 layer 1 low sample rate bursts with a repetition time interval of 384 audio sample periods.

29. The method of claim 10, wherein the bursts include MPEG2 layer 2 or 3 low sample rate bursts with a repetition time interval of 1152 audio sample periods.

30. The method of claim 10, comprising acts of:
- adding stuffing zeroes for maintaining a uniform size among all the bursts;
  - multiplexing the bursts among the channels in a recurrent manner; and
  - superposing relevant data when stuffing a burst format at a uniform size among all bursts.

31. The method of claim 10, wherein the non-PCM encoded bitstream is an AC-3 bitstream.

32. The method of claim 31, wherein the non-PCM encoded audio bursts have a repetition time interval of 1536 audio sample periods.

33. The method of claim 32, wherein the repetition time interval of the pause bursts equals 3 IEC958 frames.

34. The method of claim 31, wherein the repetition time interval of the pause bursts equals 3 IEC958 frames.

35. The method of claim 31, wherein the non-PCM encoded bitstream is an MPEG audio bitstream.

36. The method of claim 35, wherein the non-PCM encoded audio bursts have a repetition time interval of 1152 audio sample periods.

37. The method of claim 36, wherein the repetition time interval of the pause bursts equals 32 IEC958 frames.



38. The method of claim 35, wherein the repetition time interval of the pause bursts equals 32 IEC958 frames.

39. The method of claim 31, comprising acts of:

adding stuffing zeroes for maintaining a uniform size among all the bursts;

multiplexing the bursts among the channels in a recurrent manner; and

superposing relevant data when stuffing a burst format at a uniform size among all bursts.

40. A device for packaging audio samples of a non-PCM encoded audio bitstream into a sequence of IEC958 format frames, the device comprising:

packaging means for recurrently packaging data words of a multitude of audio channels of a non-PCM encoded audio bitstream into burst payloads, and for packaging the burst payloads into IEC958 format frames, resulting in a sequence of IEC958 format frames;

pause burst generating means for generating pause bursts and inserting those pause bursts into the sequence of IEC958 format frames, such pause bursts each indicating absence of audio data in one of the audio channels, each pause burst representing the absence of audio during a predetermined time interval; and

output means for outputting the sequence of IEC958 format frames including the inserted pause bursts.

41. A method of receiving a sequence of IEC958 format frames and outputting audio data for reproduction by an audio channel, the method comprising acts of:

receiving a bit stream for a plurality of audio channels as a sequence of IEC958 format frames, each frame accommodating a uniform number of data bits for each respective audio channel;

detecting by a processor presence or absence of audio data for each output channel, the absence of audio data being indicated by a series of one or more pause bursts;

upon detecting the absence of audio data for a given output channel, the processor outputting a concealing block for the given output channel for concealing the absence of audio data for the given output channel; and

alternatively, upon detecting the presence of audio data for the given output channel, the processor outputting the audio data for the given output channel.

42. A device for receiving a parsed non-PCM encoded bitstream and outputting audio data for reproduction by an audio channel, the device comprising:

reception means for receiving a non-PCM encoded bitstream for a plurality of audio channels as a sequence of frames, each frame accommodating for each applicable channel a uniform number of data bits;

detection means for detecting a presence or absence of audio data in the frames pertaining to a particular audio output channel;

decoding means for decoding and outputting decoded audio information for the particular audio channel, depending on the detection of audio data;

concealing means for generating and outputting a concealing block for the particular audio channel, for concealing the absence of audio data, depending on the detection of the absence of audio data for the particular audio channel.

43. A method of packaging data words of a non-PCM encoded multichannel bitstream into a sequence of successive frames having a predetermined format, the method comprising acts of:

recurrently packaging data words of a non-PCM encoded multichannel bitstream into data burst payloads;

packaging by a processor the data burst payloads as well as pause bursts into successive frames having a predetermined format the pause bursts signifying the absence of audio samples for a duration; and

outputting the successive frames having the predetermined format including the data burst payloads and the pause bursts.

44. A coupling device for transferring a non-PCM encoded multichannel bitstream to an audio reproduction apparatus, the device comprising:

means for recurrently packaging data words of the non-PCM encoded multichannel bitstream into data burst payloads;

means for packaging the data burst payloads as well as pause bursts into successive frames that have a predetermined format, the pause bursts signifying the absence of audio samples for a duration; and

means for outputting the successive frames that have the predetermined format including the inserted pause bursts.

45. A processor for packaging audio samples of a non-PCM encoded multichannel bitstream into a sequence of successive frames having a predetermined format, the processor comprising:

a portion for recurrently packaging audio samples of a non-PCM encoded multichannel bitstream into data burst payloads;

a portion for packaging the data burst payloads as well as pause bursts into successive frames having a predetermined format, the pause bursts signifying the absence of audio samples, each pause burst representing a pause having a duration in the millisecond range below 1000 milliseconds; and

a portion for outputting the successive frames having the predetermined format including the data burst payloads and the pause bursts.

46. A nontransitory storage device storing an audio signal comprising:

a plurality of successive frames having a predetermined format and including:

audio samples of a non-PCM encoded multichannel bitstream recurrently packaged into data burst payloads; and

pause bursts signifying the absence of audio samples for one or more of the channels of the bitstream and having a duration in the millisecond range below 1000 milliseconds.

47. A method of packaging audio samples of a non-PCM encoded audio bitstream into a sequence of format frames, the method comprising acts of:

recurrently packaging audio samples of a plurality of audio channels of a non-PCM encoded audio bitstream into burst payloads;

packaging by a processor the burst payloads into format frames according to a digital audio standard, resulting in a sequence of format frames, the packaging including:

generating pause bursts indicating absence of audio data in one or more of the audio channels for a predetermined time interval; and

inserting the pause bursts into the sequence of format frames; and

outputting the sequence of format frames including the inserted pause bursts.

48. A device for packaging audio samples of a non-PCM encoded audio bitstream into a sequence of IEC958 format frames, the device comprising:



*packaging means for recurrently packaging audio samples of a multitude of audio channels of a non-PCM encoded audio bitstream into burst payloads, and for packaging the burst payloads into format frames according to a digital audio standard, resulting in a sequence of IEC958 format frames;*

*pause burst generating means for generating pause bursts and inserting those pause bursts into the sequence of format frames, such pause bursts indicating absence of audio data in at least one of the audio channels during a predetermined time interval; and*

*output means for outputting the sequence of IEC958 format frames including the inserted pause bursts.*

49. A method of receiving a sequence of format frames and outputting audio data for reproduction by an audio channel, the method comprising acts of:

*receiving a bit stream for a plurality of audio channels as a sequence of format frames according to an audio standard each frame accommodating a uniform number of data bits for each respective audio channel;*

*detecting by a processor presence or absence of audio data for each output channel, the absence of audio data being indicated by a series of one or more pause bursts;*

*upon detecting the absence of audio data, the processor outputting a concealing block for the channel for concealing the absence of audio data for the channel; and*

*alternatively, upon detecting the presence of audio data, the processor outputting the audio data for the particular channel in question.*

50. A device for receiving a non-PCM encoded bitstream and outputting multichannel audio data for subsequent reproduction, the device comprising:

*reception means for receiving a non-PCM encoded bitstream for a plurality of audio channels as a sequence of frames, each frame accommodating data bits;*

*detection means for detecting a presence or absence of audio data in the frames;*

*decoding means for decoding and outputting decoded audio information depending on the detection of audio data;*

*concealing means for concealing the absence of audio data depending on the detection of the absence of audio data.*

51. The device of claim 50, wherein the absence of audio is detected by detecting presence of one or more pause frames.

52. The device of claim 51, wherein the concealing means mutes the audio output.

53. The device of claim 52, wherein the mute is a soft mute.

54. The device of claim 50, wherein the concealing means mutes the audio output.

55. The device of claim 54, wherein the mute is a soft mute.

56. The device of claim 50, wherein a pause burst comprises a burst preamble and a 32 bit payload.

57. The device of claim 56, wherein the burst preamble consists of four control words.

58. The device of claim 57, wherein information in a third control word of the four control words indicates that a packet type is of a pause-type.

59. A device for receiving a non-PCM encoded bitstream and outputting multichannel audio data for subsequent reproduction, the device comprising:

*a receiver configured to receive a non-PCM encoded bitstream for a plurality of audio channels as a sequence of frames, each frame accommodating data bits;*

*a detector configured to detect a presence or absence of audio data in the frames;*

*a decoder configured to decode audio information and output decoded audio information depending on the detection of audio data and to conceal the absence of audio data depending on the detection of the absence of audio data.*

60. The device of claim 59, wherein the absence of audio is detected by detecting presence of one or more pause frames.

61. The device of claim 60, wherein the concealing means mutes the audio output.

62. The device of claim 61, wherein the mute is a soft mute.

63. The device of claim 59, wherein the concealing means mutes the audio output.

64. The device of claim 63, wherein the mute is a soft mute.

65. The device of claim 59, wherein a pause burst comprises a burst preamble and a 32 bit payload.

66. The device of claim 65, wherein the burst preamble consists of four control words.

67. The device of claim 66, wherein information in a third control word of the four control words indicates that a packet type is of a pause-type.

68. A device for packaging audio samples of a non-PCM encoded audio bitstream into a sequence of frames designed for transmission of PCM encoded audio streams, the device comprising:

*a processor configured to*

*control a process of recurrently packaging audio samples of a non-PCM encoded audio bitstream into burst payloads,*

*control a process of packaging the burst payloads into a sequence of IEC958 format frames;*

*control a process of generating one or more pause bursts and inserting the one or more pause bursts into the sequence of IEC958 format frames, the one or more pause bursts indicating absence of audio data in one of the audio channels; and*

*control a process of outputting the sequence of IEC958 format frames including the inserted one or more pause bursts.*

69. The device of claim 68, wherein a pause burst comprises a burst preamble and a 32 bit payload.

70. The device of claim 69, wherein the burst preamble consists of four control words.

71. The device of claim 70, wherein information in a third control word of the four control words indicates that a packet type is of a pause-type.