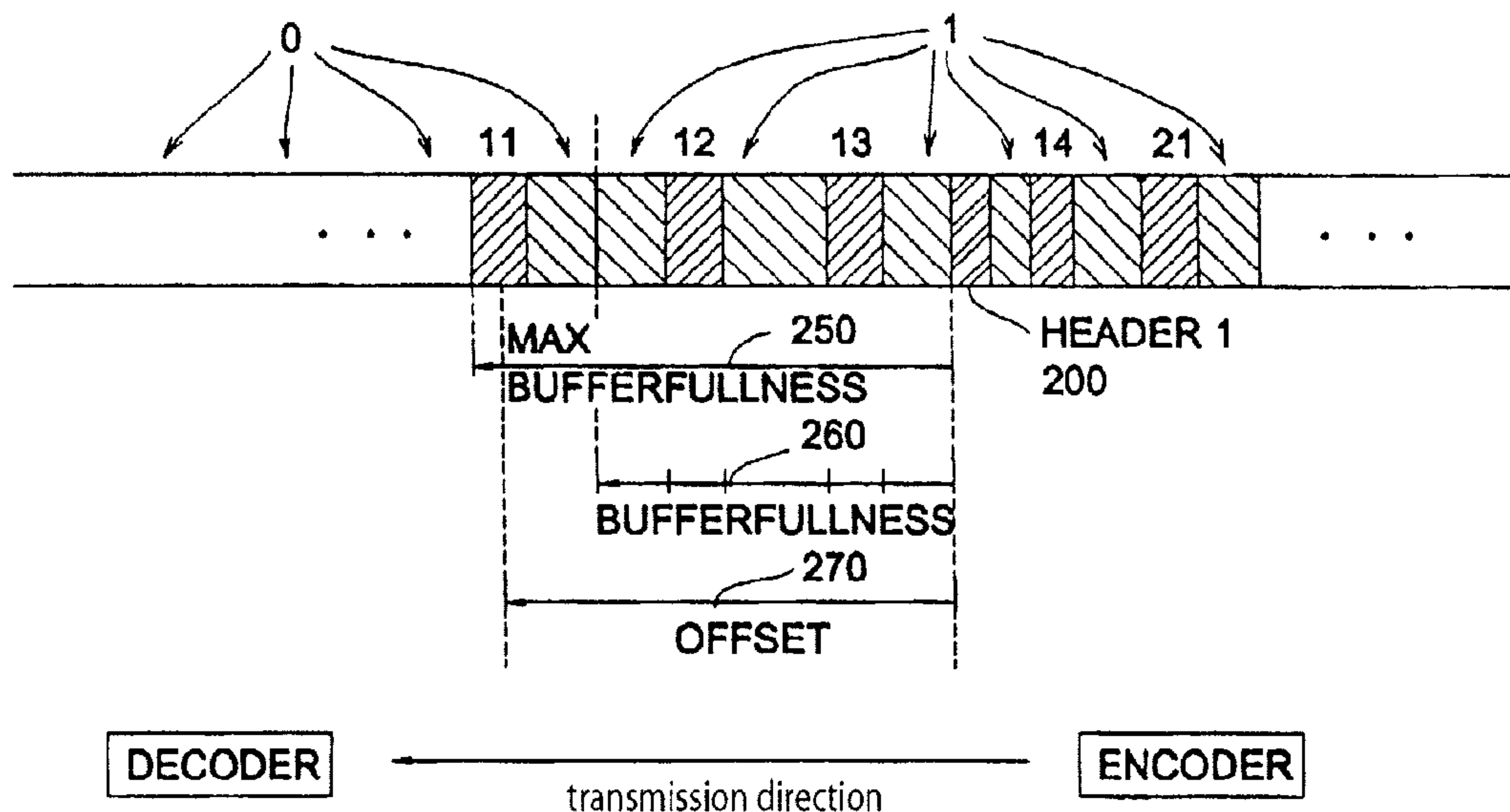




(86) Date de dépôt PCT/PCT Filing Date: 2002/01/14
 (87) Date publication PCT/PCT Publication Date: 2002/08/15
 (45) Date de délivrance/Issue Date: 2008/04/15
 (85) Entrée phase nationale/National Entry: 2003/07/15
 (86) N° demande PCT/PCT Application No.: EP 2002/000294
 (87) N° publication PCT/PCT Publication No.: 2002/063611
 (30) Priorité/Priority: 2001/01/18 (DE101 02 159.3)

(51) Cl.Int./Int.Cl. *H03M 7/00* (2006.01),
G10L 19/14 (2006.01), *H03M 7/30* (2006.01),
H04N 7/50 (2006.01)
 (72) Inventeurs/Inventors:
SPERSCHNEIDER, RALPH, DE;
TEICHMANN, BODO, DE;
LUTZKY, MANFRED, DE;
GRILL, BERNHARD, DE
 (73) Propriétaire/Owner:
FRAUNHOFER-GESELLSCHAFT ZUR FOERDERUNG
DER ANGEWANDTEN FORSCHUNG E.V., DE
 (74) Agent: GIERCZAK, EUGENE J. A.

(54) Titre : PROCÉDE ET DISPOSITIF POUR PRODUIRE OU DECODER UN FLUX DE DONNEES MODULABLE
 COMPTE TENU D'UNE BANQUE DE BITS, CODEUR ET CODEUR MODULABLE
 (54) Title: METHOD AND DEVICE FOR GENERATING AND/OR DECODING A SCALABLE DATA STREAM WITH
 PROVISION FOR A BIT SAVINGS BANK, ENCODER AND SCALABLE ENCODER



(57) **Abrégé/Abstract:**

In a method for generating a scalable data stream, when a block (11) of output data of a first encoder is present, this block of output data is written into the scalable data stream. If output data (0) of a second encoder is present for a preceding period of time, this output data for the preceding section is written in transmission direction behind the block (11) of output data of the first encoder into the data stream. When the output data (1) of the scalable encoder for the current section is present, the output data of the second encoder is written into the bit stream subsequent to the output data of the first encoder. A determining data block (200) is generated and written into the bit stream delayed by a period of time (250) which corresponds to the size of the bit savings bank of the second encoder. Finally, buffer information (260) is written into the bit stream, which indicates, where the beginning of the output data of the second encoder for the current section regarding the determining data block is, wherein the buffer information (260) corresponds to the bit savings bank level. Thus, it is possible to simply signalize a bit savings bank in a scalable data stream. The maximum size of the bit savings bank may further be adjusted depending on the intended decoder delay and be communicated to a decoder by positioning the determining data block in the scalable data stream without an effort of additional bits in order to reduce the initial delay of the decoder.



Method and Device for Generating and/or Decoding a Scalable Data Stream with Provision for a Bit Savings Bank, Encoder and Scalable Encoder

Abstract

In a method for generating a scalable data stream, when a block (11) of output data of a first encoder is present, this block of output data is written into the scalable data stream. If output data (0) of a second encoder is present for a preceding period of time, this output data for the preceding section is written in transmission direction behind the block (11) of output data of the first encoder into the data stream. When the output data (1) of the scalable encoder for the current section is present, the output data of the second encoder is written into the bit stream subsequent to the output data of the first encoder. A determining data block (200) is generated and written into the bit stream delayed by a period of time (250) which corresponds to the size of the bit savings bank of the second encoder. Finally, buffer information (260) is written into the bit stream, which indicates, where the beginning of the output data of the second encoder for the current section regarding the determining data block is, wherein the buffer information (260) corresponds to the bit savings bank level. Thus, it is possible to simply signalize a bit savings bank in a scalable data stream. The maximum size of the bit savings bank may further be adjusted depending on the intended decoder delay and be communicated to a decoder by positioning the determining data block in the scalable

data stream without an effort of additional bits in order to reduce the initial delay of the decoder.

**Method and Device for Generating and/or Decoding a
Scalable Data Stream with Provision for a Bit Savings
Bank, Encoder and Scalable Encoder**

5

Description

The present invention relates to scalable encoders and decoders and in particular to the generation of scalable data streams.

10

Scalable encoders are shown in EP 0 846 375 B1. In general, scalability is understood as the possibility of decoding a partial section of a bit stream representing an encoded data signal, e.g. an audio signal or a video
15 signal into a useful signal. This property is particularly desirable when e.g. a data transmission channel fails to provide the complete bandwidth necessary for transmitting a complete bit stream. On the other hand, an incomplete decoding is possible on a decoder with reduced complexity.
20 Generally, different discrete scalability layers are defined in practice.

An example of a scalable encoder as defined in Subpart 4 (General Audio) of Part 3 (Audio) of the MPEG-4 Standard
25 (ISO/IEC 14496-3; 1999 Subpart 4) is shown in Fig. 1. An audio signal $s(t)$ to be encoded is fed into the scalable encoder on the input side. The scalable encoder shown in Fig. 1 contains a first encoder 120, which is an MPEG Celp encoder. The second encoder 140 is an AAC encoder, which
30 provides high-quality audio encoding and is defined in the Standard MPEG-2 AAC (ISO/IEC 13818). The Celp encoder 120 provides a first scaling layer via an output line 16,

while the AAC encoder 140 provides a second scaling layer via a second output line 18, to a bit stream multiplexer (BitMux) 20. On the output side the bit stream multiplexer then outputs an MPEG-4-LATM bit stream 22 (LATM = Low-
5 Overhead MPEG-4 Audio Transport Multiplex). The LATM format is described in Section 6.5 of Part 3 (Audio) of the first supplement to the MPEG-4 Standard (ISO/IEC 14496-3:1999/AMD1:2000).

10 The scalable audio encoder further includes some further elements. First, there exists a delay stage 240 in the AAC branch and a delay stage 26 in the Celp branch. With both delay stages it is possible to set an optional delay for the respective branch. A downsampling stage 28 is
15 downstream of the delay stage 26 of the Celp branch to adjust the sampling rate of the input signal $s(t)$ to the sampling rate requested by the Celp encoder. An inverse Celp decoder 30 is downstream to the Celp encoder 120, wherein the Celp encoded/decoded signal is then supplied
20 to an upsampling stage 32. The upsampled signal is then supplied to a further delay stage 34, which is termed "Core Coder Delay" in the MPEG-4 Standard.

The stage CoreCoderDelay 34 has the following function. If
25 the delay is set to zero, the first encoder 140 and the second encoder 120 process exactly the same samples of the audio input signal in a so-called superframe. A superframe might e.g. consist of three AAC frames, which together represent a certain number of samples No. x to No. y of
30 the audio signal. The superframe further includes e.g. 8 CELP blocks, which represent the same number of samples

and also the same samples No. x to No. y if CoreCoderDelay = 0.

If, however, a CoreCoderDelay D is set as a time value other than zero, the three blocks of AAC frames nevertheless represent the same samples No. x to No. y . The eight blocks of CELP frames, in contrast, represent the samples No. $x - F_s D$ to No. $y - F_s D$, wherein F_s is the sampling frequency of the input signal.

10

The current time sections of the input signal in a superframe for the AAC blocks and the CELP blocks can thus be either identical, when CoreCoderDelay $D = 0$, or be shifted relative to each other by CoreCoderDelay, when D is not equal to zero. For the following implementations, however, it will be assumed, on the grounds of simplicity and without restriction of generality, that CoreCoderDelay = 0, so that the current time section of the input signal for the first encoder and the current time section for the second encoder are identical. In general, however, the only requirement for a superframe is, that the AAC block(s) and the CELP block(s) in a superframe represent the same number of samples, wherein it is not necessary for the samples themselves to be identical to one another, but they may also be shifted relative to each other by CoreCoderDelay.

It should be noted that the Celp encoder, depending on the configuration, may process a section of the input signal $s(t)$ faster than the AAC encoder 140. In the AAC branch a block decision stage 26 is downstream to the optional delay stage 240 which establishes among other things

30

whether short or long windows should be used for windowing the input signal $s(t)$, wherein short windows must be chosen for strongly transient signals, while long windows are preferred for less transient signals since the
5 relationship between the amount of payload data and page information is better than for short windows.

By the block decision stage 26 a fixed delay by e. g. 5/8 times a block is performed in the present example. This is
10 referred to as a look-ahead function in the art. The block decision stage must already look ahead a certain time to be able to determine whether there are transient signals in future that must be encoded with short windows. After that the corresponding signal in the Celp branch as well
15 as the signal in the AAC branch are fed to means for converting the time-related illustration to a spectral illustration, which is designated as MDCT 36 or 38, respectively, in Fig. 1 (MDCT = modified discrete cosine transform). The output signals of the MDCT blocks 36, 38
20 are then supplied to a subtracter 40.

At this point, samples belonging together regarding time must be present, i.e. the delay must be identical in both
branches.

25 The following block 44 determines whether it is more favorable to supply the input signal itself to the AAC encoder 140. This is enabled via the bypass branch 42. If it is determined, however, that the differential signal at
30 the output of the subtracter 40 is smaller regarding energy than the signal output by the MDCT block 38, then not the original signal but the differential signal is

taken to be encoded by the AAC encoder 140 to finally form the second scaling layer 18. This comparison may be performed band by band, which is indicated by frequency-selective switching means (FSS) 44. The exact functions of the individual elements are known in the art and are described for example in the MPEG-4 standard as well as in further MPEG standards.

One main feature in the MPEG-4 standard and in other encoder standards, respectively, is that the transmission of the compressed data signal is to be performed with a constant bit rate via a channel. All high-quality audio codecs operate based on blocks, i.e. they process blocks of audio data (order 480 - 1024 samples) to pieces of a compressed bit stream, which are also referred to as frames. The bit stream format must here be set up so that a decoder without a priory information where a frame starts is able to recognize the beginning of a frame in order to start the output of decoded audio signal data with a lowest possible delay. Thus, each header or determining data block of a frame starts with a certain synchronization word which may be searched for in a continuous bit stream. Further common components within the data stream apart from the determining data block are the main data or "payload data" of the individual layers in which the actual compressed audio data is contained.

Fig. 4 shows a bit stream format with a fixed frame length. In this bit stream format the headers or determining data blocks are inserted equidistantly into the bit stream. The side information associated with this header and the main data follow immediately afterwards.

The length, i.e. the number of bits, for the main data is the same in each frame. Such a bit stream format as it is shown in Fig. 4 is for example used in the MPEG layer 2 or the MPEG-CELP.

5

Fig. 5 shows another bit stream format with a fixed frame length and a backpointer. In this bit stream format the header and the side information are arranged equidistantly as in the format illustrated in Fig. 4. The start of the associated main data is, however, only performed exceptionally directly following a header. In most cases the start is in one of the preceding frames. The number of bits by which the start of the main data is shifted in the bit stream is transferred by the page information variable backpointer. The end of these main data may lie within this frame or within a preceding frame. The length of the main data is therefore not constant any more. Therefore, the number of bits with which a block is encoded may be adjusted to the characteristics of the signal. Simultaneously, a constant bit rate may be achieved, however. This technology is called "bit savings bank" and increases the theoretical delay within the transmission chain. Such a bit stream format is for example used in the MPEG layer 3 (MP3). The technology of the bit savings bank is further described in the standard MPEG layer 3.

10
15
20
25

Generally, the bit savings bank represents a buffer of bits which may be used to provide more bits for encoding a block of time sample as is actually allowed by the constant output data rate. The technology of the bit savings bank takes into account that some blocks of audio samples may be encoded with less bits than predetermined

30

by the constant transmission rate, so that through these blocks the bit savings bank is filled, while again other blocks of audio samples comprise psychoacoustic characteristics which do not allow such a high compression so that for these blocks the available bits would actually not be enough for a low-interference or interference-free encoding, respectively. The additional bits needed are taken from the bit savings bank so that the bit savings bank is emptied with such blocks.

10

Such an audio signal may, however, be also transmitted by a format with a variable frame length, as it is shown in Fig. 6. With the bit stream format "variable frame length", as it is illustrated in Fig. 6, the fixed sequence of the bit stream elements header, page information and main data is maintained, as with the "fixed frame length". As the length of the main data is not constant, the bit savings bank technology may also be used here, there are, however, no backpointers needed as in Fig. 5. One example for a bit stream format, as it is illustrated in Fig. 6, is the transport format ADTS (audio data transport stream), as it is defined in the standard MPEG 2 AAC.

25 It is to be noted that the above-mentioned encoders are no scalable encoders but include only one single audio encoder.

In MPEG 4 the combination of different encoder/decoders to a scalable encoder/decoder is provided. It is therefore possible and sensible to combine one CELP voice encoder as the first encoder with an AAC encoder for the further

30

scaling layer(s) and pack the same into one bit stream. The purpose of this combination is that the possibility remains open either to decode all scaling layers and therefore reach a best possible audio quality, or parts of the same, maybe even only the first scaling layer, with the correspondingly restricted audio quality. Reasons for only decoding the lowest scaling layer may be that due to a bandwidth of the transmission channel which is too small, the decoder only received the first scaling layer of the bit stream. Because of this the parts of the first scaling layer in the bit stream are favored over the second and the further scaling layers in the transmission, whereby the transmission of the first scaling layer is guaranteed with capacity bottlenecks in the transmission network, while the second scaling layer may be lost completely or in part.

A further reason may be that a decoder wants to achieve a lowest possible codec delay and therefore decodes only the first scaling layer. It is to be noted that the codec delay of a Celp code is generally significantly smaller than the delay of the AAC code.

In MPEG 4 version 2 the transport format LATM is standardized, which may among other things also transmit scalable data streams.

In the following, reference is made to Fig. 2a. Fig. 2a is a schematical illustration of the samples of the input signal $s(t)$. The input signal may be divided into different successive sections 0, 1, 2, 3, wherein each section comprises a certain fixed number of time samples.

Usually, the AAC encoder 140 (Fig. 1) processes a whole section 0, 1, 2 or 3 in order to provide an encoded data signal for this section. The CELP encoder 120 (Fig. 1), however, processes usually a smaller amount of time samples per encoding step. Thus, it is shown as an example in Fig. 2b, that the CELP encoder or generally speaking the first encoder or encoder 1 comprises a block length which is one fourth of the block length of the second encoder. It is to be noted that this division is completely random. The block length of the first encoder may also be half as long, might, however, also be one eleventh of the block length of the second encoder. Thus, the first encoder will generate four blocks (11, 12, 13, 14) from the section of the input signal, from which the second encoder provides one block of data. In Fig. 2c a common LATM bit stream format is shown.

One superframe may comprise several ratios of number of AAC frames to number of CELP frames, as it is illustrated in tabular form in MPEG 4. Thus, a superframe may for example comprise one AAC block and 1 to 12 CELP blocks, 3 AAC blocks and 8 CELP blocks but also e.g. for example more AAC blocks than CELP blocks, depending on the configuration. An LATM frame which comprises an LATM determining data block includes a superframe or also several superframes.

The generation of the LATM frame opened by the header 1 is described as an example. First, the output data blocks 11, 12, 13, 14 of the Celp encoder 12 (Fig. 1) are generated and buffered. In parallel, the output data block of the AAC encoder designated with "1" in Fig. 2c is generated.

Then, when the output data block of the AAC encoder has been generated, first of all the determining data block (header 1) is written. Depending on the convention, the output data block of the first encoder which was generated first, designated with 11 in Fig. 2c, may be written, i.e. transmitted, directly following header 1. Usually (regarding the few necessary signaling information) an equidistant distance of the output data blocks of the first encoder is selected for a further writing and/or transmitting of the data stream, as it is illustrated in Fig. 2c. This means, that after writing and/or transmitting block 11 the second output data block 12 of the first encoder, then the third output data block 13 of the first encoder and then the fourth output data block 14 of the first encoder are written and/or transmitted in equidistant distances. The output data block 1 of the second encoder is filled into the remaining gaps during the transmission. Then, an LATM frame is fully written, i.e. fully transmitted.

20

One disadvantage of the bit stream formats illustrated in Fig. 4 to 6 is the fact that they are only known for simple encoders, not, however, for scalable encoders and in particular not for scalable encoders having a bit savings bank function.

25

As it is known, the bit savings bank is used so that the variable output data rate which a psychoacoustic encoder generates inherently may be adjusted to a constant output data rate. In other words, the number of bits an audio encoder needs depends on the signal characteristics. If the signal is comprised such that it may be quantized in

30

relatively coarse way, then a relatively low amount of bits is needed for encoding this signal. If the signal is, however, comprised such that it has to be quantized very finely, a relatively low amount of bits is needed for encoding this signal. If the signal is, however, comprised such that it needs to be quantized very finely in order not to introduce audible interferences, then a larger amount of bits is needed for encoding this signal.

10 In order to achieve a constant output data rate, a medium amount of bits is determined for one section of a signal to be encoded. If the actually needed amount of bits for encoding a section is smaller than the determined number of bits, then the bits which are not needed may be placed
15 into the bit savings bank. Thus, the bit savings bank is filled. If, however, a section of a signal to be encoded is comprised such that a larger number than the determined number of bits is needed for encoding in order not to introduce audible interferences into the signal, then the
20 additionally needed bits may be taken from the bit savings bank. That way, the bit savings bank is emptied. Thereby it may be guaranteed that a constant output data rate is maintained and at the same time no audible interferences are introduced into the audio signal. A precondition for
25 this is that the bit savings bank is selected to be sufficiently large.

In the standard MPEG AAC (13818-7:1997) a bit savings bank is referred to as "bit reservoir". The maximum size of the
30 bit savings bank for channels with a constant data rate may be calculated by subtracting the average amount of bits per block from the maximum decoder input buffer size.

Its value is usually firmly preset to a value of 10,240 bits according to the standard MPEG AAC with a transmission rate of 96 kBit/s for a stereo signal with a sampling rate of 48 kHz. The maximum value of the bit savings bank, i.e. the size of the bit savings bank is sized so that also under bad conditions, i.e. also when the signal comprises many sections which may not be encoded with the determined number of bits, audible interferences need to be introduced into the audio signal in order to maintain the constant output data rate. This is only possible when the bit savings bank is sized sufficiently large so that it is emptied at no time.

On the decoder side this has the following consequence. After the decoder has to consider that both the case of a full bit savings bank and the case of an empty bit savings bank may occur in the course of decoding an audio signal, the decoder needs to buffer a number of bits corresponding to the size of the bit savings bank before it starts decoding at all. Thereby it is guaranteed that the decoder does not run out of bits during decoding the audio signal. If a decoder would immediately decode a signal encoded with the bit savings bank function when it has received the same, then the bits for the output would already run out when the first block to be decoded by accident needed a smaller number than the determined number for encoding, i.e. when the bit savings bank was filled up by the first block. In other words, the bit savings bank function inevitably leads to a delay within the decoder, wherein this delay corresponds to the size of the bit savings bank.

For the preceding example the size of the bit savings bank is 10,240 bits. This leads to an inherent initial delay due to the bit savings bank of about 0.1 s. The delay gets larger, the larger the maximum size of the bit savings bank is selected and the smaller the transmission rate is selected.

If, for example, real-time transmissions of a telephone call are considered, in which a continuous change of speakers takes place, then already due to the bit savings bank a delay of the mentioned size occurs with each change of speaker. Such a delay is extraordinarily disturbing for both communication partners and typically leads to the fact that one speaker, because he does not immediately hear a reaction of the other speaker, that the one speaker repeats the question again, which contributes to a further confusion. Therefore, it is determined that a product designed this way is not suitable for real-time applications and would not have a chance of a breakthrough in the market, respectively.

It is the object of the present invention to provide an encoder comprising a bit savings bank function through which a smaller transmission delay may be achieved.

25

This object is achieved by an encoder according to claim 5 or by a scalable encoder according to claim 6.

It is a further object of the present invention to provide a method and a device for generating a scalable data stream in which a bit savings bank function may be signaled.

30

This object is achieved by a method according to claim 1 or by a device according to claim 7.

5 It is a further object of the present invention to provide a method and a device for decoding a scalable data stream in which a bit savings bank function is signaled.

This object is achieved by a method according to claim 8
10 or by a device according to claim 9.

The present invention is based on the findings that the present concept of the fixed set bit savings bank size must be discarded in order to achieve a reduced-delay
15 decoding. According to the invention, this is achieved by making the maximum size of the bit savings bank of an encoder adjustable, wherein depending on the application and depending on the intended decoder function a certain adjustment of the bit savings bank is achieved. For the
20 case of a one-directional data transmission only a large bit savings bank may be selected in order to satisfy highest possible audio quality requirements, while for the case of a bi-directional communication in which a frequent change of transmitter and receiver and a frequent change
25 of speakers takes place, respectively, a smaller bit savings bank size is to be adjusted. So that the decoder may profit from a smaller bit savings bank size adjustment, the bit savings bank size must be transmitted to the decoder in some way. This may on the one hand be
30 achieved by the transmission of additional information in the data stream, it may however also be performed implicitly without the transmission of additional side

information and signaling information, respectively, as it is illustrated in particular with reference to the scalable case.

5 One advantage of the present invention is that now direct influence may be taken on the decoder delay via the adjustment of the maximum size of the bit savings bank. If the maximum size of the bit savings bank is selected smaller, then the decoder may also insert a smaller delay
10 before it starts decoding without risking the danger that it may run out of output data during decoding which needs to be prevented in any case. The "price" which has to be paid for this is that one or the other section of the audio signal was not encoded with 100 % of the audio
15 quality, as the bit savings bank was empty and no additional bits were available any more. Usually, an audio encoder reacts in this case by violating the psychoacoustic masking threshold when quantizing and, in order to make do with the available number of bits,
20 selects a coarser quantization as is really needed. The main advantage of the smaller delay of the decoder is, however, guaranteed. The reduction of the size of the bit savings bank in order to reach a smaller delay also on the decoder side is therefore achieved with a lower audio
25 quality, wherein this lower audio quality only occurs now and then in the audio signal, and when the audio signal is simple to decode it may not occur at all. As a result, the inflexibility regarding the bit savings bank according to the prior art is overcome, which may be over-dimensioned
30 for many applications in order to encode all possible cases with a high audio quality, so that a use of encoders for a bi-directional communication with frequently

changing speakers becomes possible which was not conceivable up to now due to the large fixedly adjusted bit savings bank.

5 The inventive variability of the bit savings bank and the accompanying variability of the delay on the decoder side is especially of an advantage in the case of a scalable audio encoder, as now also here a reduced-delay decoding may not only be achieved of the first lowest scaling
10 layers but also a reduced-delay decoding of higher scaling layers which are for example generated by an AAC encoder may be achieved. In particular in the scalable case only one scaling layer is influenced by the variable adjustment of the bit savings bank, while the other scaling layer(s)
15 remain unaffected. It is thus possible to act upon individual scaling layers deliberately without causing any changes in the other scaling layers.

As it was already discussed it is necessary to communicate
20 the freely selectable and the freely selected bit savings bank size, respectively, to the decoder. This was not necessary in the prior art, as a fixed bit savings bank size was always agreed upon, so that a decoder introduced the corresponding delay for example by dimensioning its
25 input buffer knowing the bit savings bank size which was firmly agreed on.

In particular for scalable encoders and scalable data stream an adjustable bit savings bank size without
30 additional side information may be achieved simply by positioning a determining data block within the scalable data stream. According to the invention, the determining

data block is positioned within the bit stream so that the decoder needs to receive as many bits for the respective layer as it is determined by the average block length when it receives the determining data block.

5

After receiving a frame, the decoder may start decoding without calculating or inserting a delay. This is achieved due to the fact that already within the scalable data stream the determining data block is written in a delayed manner regarding the first and the second scaling layer, i.e. preferably delayed by a period of time which corresponds to the adjustment of the bit savings bank. Thereby it is achieved that the encoder may select any bit savings bank size depending on the requirement and that the selected bit savings bank size simply implicitly signalizes to the decoder, for it to enter the determining data block in the bit stream in a delayed manner with regard to the payload data.

20 In other words, the consequence is that the determining data block is not written at the first possible point of time anymore, i.e. delay-optimized, as in the prior art, but at the latest possible point of time, without delaying the AAC block. The current level of the bit savings bank may then be signaled by the so-called backpointer, where the data of a preceding section end and where the data of the current section begin.

This is true both for the scalable case in which only output data of one individual encoder occur in the bit stream, and also for the scalable case, in which data of at least two different encoders occur in the scalable bit

30

stream. If a superframe, i.e. a section in the bit stream comprising a first number of output data blocks of a first encoder and a second number of output data blocks of a second encoder which relate to the same number of samples
5 of a input signal, comprises a plurality of blocks of an encoder, then the number of blocks of the one encoder which are associated with a determining data block can simply be signaled by the fact that offset information is transferred with the bit stream. The offset information
10 may also be interpreted by the decoder as backpointer in order to know which data of the bit stream now belong to a determining data block and therefore correspond to a time section of the input signal if necessary considering the variable core coder delay.

15

One main advantage of this arrangement is that the decoder, when it receives an inventive data stream, must not calculate and insert a delay, but that the delay was already considered by the positioning of the determining
20 data block alone on the encoder side. The decoder can therefore output a frame immediately after the reception. This also provides the possibility to signalize an adjusted maximum bit savings bank size in a simple way, i.e. without additional bits. As the signalization may be
25 performed in a simple and without efforts, i.e. by the position of the determining data block, it is also possible easily and in particular without access to the decoder to vary the bit savings bank size in order to be able to adjust the transmission delay as desired.

30

In the following, preferred embodiments of the present invention are explained in more detail referring to the accompanying drawings, in which:

- 5 Fig. 1a shows a scalable encoder according to MPEG 4 which comprises the present invention;
- Fig. 1b shows a decoder according to the present invention;
- 10 Fig. 2a shows a schematical illustration of an input signal which is divided into successive time sections;
- 15 Fig. 2b shows a schematical illustration of an input signal which is divided into successive time sections, wherein the ratio of the block length of the first encoder to the block length of the second encoder is illustrated;
- 20 Fig. 2c shows a schematical illustration of a scalable data stream with a high delay in decoding the first scaling layer;
- 25 Fig. 2d shows a schematical illustration of a scalable data stream with a low delay in decoding the first scaling layer;
- 30 Fig. 2e shows a schematical illustration of an inventive scalable data stream wherein the determining data block is delayed with reference to the payload data;

Fig. 3 shows a detailed illustration of the inventive scalable data stream regarding the example of a Celp encoder as the first encoder and an AAC encoder as the second encoder with a bit savings bank function;

Fig. 4 shows an example for a bit stream format with a fixed frame length;

Fig. 5 shows an example for a bit stream format with a fixed frame length and a backpointer; and

Fig. 6 shows an example of a bit stream format with a variable frame length.

In the following, Fig. 2d is referred to in comparison to Fig. 2c in order to explain a bit stream with a small delay of the first scaling layer for purposes of comparison. As in Fig. 2c the scalable data stream contains successive determining data blocks which are referred to as header 1 and header 2. In the preferred embodiment of the present invention which is implemented according to the MPEG 4 standard the determining data blocks are LATM headers. Like in the prior art in the transmission direction from an encoder to a decoder, which is illustrated in Fig. 2d with an arrow 202, behind the LATM header 200 the parts hatched from top right to bottom left of the output data block of the AAC encoder are located which are inserted in gaps remaining between the output data blocks of the first encoder.

In contrast to the prior art, there are not only output data blocks of the first encoder within the frame started by the LATM header 200 anymore, which belong to this frame, like for example the output data blocks 13 and 14, but also the output data blocks 21 and 22 of the following section of input data. In other words, in the example illustrated in Fig. 2d, the two output data blocks of the first encoder, which are designated with 11 and 12, are present in the bit stream in the transmission direction (arrow 202) before the LATM header 200. In the example illustrated in Fig. 2d the offset information 204 indicate an offset of two output data blocks of the output data blocks of the first encoder. When Fig. 2d is compared to Fig. 2c it may be seen that the decoder may already decode the lowest scaling layer earlier by a time which exactly corresponds to this offset than it is the case in Fig. 2c, if the decoder is only interested in the first scaling layer. The offset information, which may for example be signalized in the form of a "core frame offset", serve to determine the position of the first output data block 11 in the bit stream.

For the case of core frame offset = zero, the bit stream indicated in Fig. 2c results. If, however, core frame offset > zero, then the corresponding output data block of the first encoder 11 is transmitted earlier by the number of core frame offset at the output data blocks of the first encoder. In other words, the delay between the first output data block of the first encoder after the LATM header and the first AAC frame results from core coder delay (Fig. 1) + core frame offset x core block length (block length of encoder 1 in Fig. 2b). As it becomes

clear from the comparison of Fig. 2c and 2d, for the case of core frame offset = zero (Fig. 2c), the output data blocks 11 and 12 of the first encoder are transmitted after the LATM header 200. By the transmission of core
5 frame offset = 2 the output data blocks 13 and 14 may follow after the LATM header 200, whereby the delay with a pure CELP decoding, i.e. the decoding of the first scaling layer, is reduced by two CELP block lengths. An offset of three blocks would be optimum in the example. An offset of
10 one or two blocks brings, however, already a delay advantage.

Through this bit stream structure it is possible for the Celp encoder to transmit the generated Celp block directly
15 after the encoding. In this case no additional delay is added to the CELP encoder by the bit stream multiplexer (20). Thus, for this case no additional delay is added to the Celp delay by the scalable combination, so that the delay is at its minimum.

20

It is noted that the case illustrated in Fig. 2d is only exemplary. This way, different ratios of the block length of the first encoder to the block length of the second encoder are possible, which may for example vary from 1:2
25 to 1:12, may however also take different ratios.

In the extreme case this means (1:12 for MPEG 4 AAC/CELP), that for the same time section of the input signal for which the AAC encoder generates an output data block, the
30 Celp encoder generates twelve output data blocks. The delay advantage by the data stream illustrated in Fig. 2d in contrast to the data stream illustrated in Fig. 2c may

in this case easily take magnitudes from one fourth up to half a second. This advantage will be increased the greater the ratio between the block length of the second encoder and the block length of the first encoder becomes, wherein in the case of an AAC encoder as the second encoder a block length as great as possible is aimed at due to the ratio which is then more favorable from payload information to side information, if the encoding signal admits it.

10

In Fig. 2c a scalable data stream according to the LATM format is illustrated in which the data blocks of the first encoder have to be buffered, i.e. delayed. In the format of Fig. 2 this results from the fact, as it was discussed, that the header may only be written when the output data of the second encoder are present, as the header includes information about the length and the number of bits, respectively, within the output data block of the second encoder.

20

Thus, in Fig. 2d for purposes of illustration an improvement is already illustrated regarding the fact that the output data blocks of the first encoder are already written into the bit stream earlier in order to reduce the delay when a decoder only wants to decode the lowest scaling layer. Nevertheless, the determining data block is still located before the output data block of the second encoder, which is designated with "1" in Fig. 2d.

30 In Fig. 2e now, compared to Fig. 2c, the inventive scalable data stream is illustrated, wherein the determining data block (header 1 200) is not immediately

written anymore when it is available, i.e. before the output data block of the first encoder which is designated with "11", but in which the determining data block 200 is written into the data stream delayed by a period of time in relation to the case of Fig. 2c. This period of time equals the maximum size of the bit savings bank (max bufferfullness 250) in a preferred embodiment of the present invention. Therefore the output data block of the second encoder for the current section of the input signal, designated by the determining data block 200, starts a number of bits equal to bufferfullness 260 before the determining data block in the transmission direction from an encoder to a decoder, whereas it can be seen from Fig. 2c that the AAC data have started behind the determining data block.

From the point of view of the decoder the pointer 260 is therefore a backpointer.

For the case, that the first encoder provides a larger number of blocks for a number of samples than the second decoder, wherein in the example illustrated in Fig. 2e the ratio of four blocks of output data of the first encoder to a block of output data of the second encoder is only exemplary for the same number of samples, based on the determining data block, as in the case of Fig. 2e, a core frame offset is signaled, so that a decoder knows which blocks of output data of the first encoder for example belong to a block of output data of the second encoder or are related to each other via core coder delay, respectively.

If now Fig. 2d is compared to Fig. 2e, then it may be seen that also in Fig. 2e an offset 204 is present. The offset 204 of Fig. 2d which has a value of 2 in Fig. 2d would increase to a value of 5 with regard to the case of Fig. 5 2e, as the determining data block 200 in Fig. 2e compared to Fig. 2d has been shifted backwards by three output data blocks of the first encoder.

In the following, reference is made to Fig. 1a again. In 10 addition to the scalable encoder already described in the description introduction, the inventive scalable encoder illustrated in Fig. 1a contains a block bit savings bank control 50 and a control line 52 from the AAC encoder 14 to the bit stream multiplexer 20, via which the maximum 15 size of the bit savings bank which was adjusted by the bit savings bank control 50, may be communicated to the bit stream multiplexer so that the same may perform the bit stream formatting required in Fig. 2e.

20 In Fig. 1b a schematical block diagram of a scalable decoder may be found which is complementary to the scalable encoder in Fig. 1a. The scalable bit stream which is supplied to the encoder via a line 60 is fed into an input buffer/bit stream demultiplexer 62 of the decoder. 25 Here, the bit stream is divided, to extract the required blocks for a CELP decoder 64 and an AAC decoder 66. The inventive decoder further includes an AAC delay stage 68 which serves for introducing a delay corresponding to the bit savings bank size, so that the AAC decoder 66 never 30 runs out of data to put out. According to the invention, this AAC delay stage is now implemented variably, wherein the delay is controlled depending on the bit savings bank

information, which are extracted from the bit stream by the bit stream demultiplexer 62 and supplied to the AAC delay stage 68 via a bit savings bank information line 60. Depending on the bit savings bank level now the delay of the AAC delay stage 68 is adjusted. If a small bit savings bank is adjusted by bit savings bank control means 50 of Fig. 1a, then also the AAC delay stage 68 may be adjusted to a small delay, so that a reduced-delay decoding of the second scaling layer may be achieved.

10

The scalable decoder of Fig. 1b further includes MDCT means 72 to transform the time domain output signals of the CELP decoder 64 into the frequency domain, and an upsampling stage upstream to the same. The spectrum is delayed by the delay stage 74, which compensates time differences present between the two branches, so that at means 76 which are referred to as adder/ FSS^{-1} , the same ratios are present. Means 66 basically performs the analog function to the subtractor 40 and the FSS 44 of Fig. 1a. After block 76 the spectral values are transformed by means 78 for performing a back-transformation from the frequency domain into the time domain, so that at an output 80 either only the second scaling layer or the first and the second scaling layer are present in the time domain. At an output 82, however, only the first scaling layer is present in the time domain generated by the CELP decoder 64.

25

In the following, reference is made to Fig. 3, which is similar to Fig. 2, illustrates, however, the special implementation referring to the example of MPEG 4. In the first row again a current time section is shown hatched.

30

In the second row the windowing which is used with the AAC encoder is illustrated schematically. As it is known, an overlap-and-add of 50 % is used so that a window usually comprises double the length of time samples than the current time section which is illustrated hatched in the top row of Fig. 3. In Fig. 3 the delay t_{dip} is further illustrated, which corresponds to block 26 of Fig. 1 and comprises a size of $5/8$ of the block length in the selected example. Typically, a block length of the current time section of 960 samples is used so that the delay t_{dip} of $5/8$ of the block length comprises 600 samples. For example, the AAC encoder provides a bit stream of 24 kbit/s, while the CELP encoder schematically illustrated below provides a bit stream comprising a rate of 8 kbit/s. The overall bit rate is then 32 kbit/s.

As it may be seen from Fig. 3, the output data blocks zero and one of the CELP encoder correspond to the current time section for the first encoder. The output data block comprising the number 2 of the CELP encoder already corresponds to the next time section. The same holds true for the CELP block with the number 3. In Fig. 3, the delay of the downsampling stage 28 and the CELP encoder 12 is further illustrated by an arrow which is designated by the reference numeral 302. From this, the delay designated by core coder delay and illustrated by an arrow 304 in Fig. 3 results as the delay which needs to be adjusted by stage 34 so that at the subtraction location 40 of Fig. 1 equal ratios are present. This delay may alternatively be generated by block 26. For example:

core coder delay =

= t_{dip} - Celp encoder delay - downsampling delay =

= $600 - 120 - 117 = 363$ samples.

5

For the case without a bit savings bank function and for the case, respectively, that the bit savings bank (bit mux outputbuffer) is full, which is indicated by the variable $bufferfullness = \max$, the case indicated in Fig. 2d results. In contrast to Fig. 2d in which four output data blocks of the first encoder are generated corresponding to one output data block of the second encoder, in Fig. 3 two output data blocks of the CELP encoder designated with "0" and "1" are generated for an output data block of the second encoder which is drawn in black in the two last rows of Fig. 3. According to the invention, now, however, not the output data block of the CELP encoder with the number "0" is written behind a first LATM header 306 anymore, but the output data block of the CELP encoder with the number "one", as the output data block with the number "zero" has already been transmitted back to the decoder. In the equidistant grid distance provided for the CELP data blocks, the CELP block 1 is then followed by the CELP block 2 for the next time section, wherein then for the completion of a frame the rest of the data of the output data block of the AAC encoder is written into the data stream until a next LATM header 308 for the next time section follows.

30

The present invention may simply be combined with the bit savings bank function, as it is illustrated in the last row of Fig. 3. For the case, that the variable

"bufferfullness" which indicates the filling of the bit savings bank, is smaller than the maximum value, this means, that the AAC frame for the directly preceding time section needed more bits than it is actually admissible.

5 This means, that behind the LATM header 306 the CELP frames are written as before, that however first the at least one output data block of the AAC encoder needs to be written from one or several preceding time sections in the bit stream before the writing of the output data block of

10 the AAC encoder for the current time section may be started. From the comparison of the last two rows of Fig. 3 which are designated by "1" and "2" it may be seen that the bit savings bank function also directly leads to a delay in the encoder for the AAC frame. So the data for

15 the AAC frame of the current time domain, which is designated by 310 in Fig. 3, is however present at the same point of time as in case "1", can however only be written into the bit stream after the AAC data 312 for the directly preceding time section have been written into the

20 bit stream. Depending on the bit savings bank level of the AAC encoder therefore the initial position of the AAC frame is shifted.

The bit savings bank level is to be transferred in the

25 LATM element StreamMuxConfig by the variable "bufferfullness". The variable bufferfullness is calculated from the variable bit reservoir divided by the 32-fold of the actually present channel number of the audio channels.

30

It is to be noted that the pointer designated by the reference numeral 314 in Fig. 3, whose length = max

bufferfullness - bufferfullness, is a forward-pointer which points to the future as it were, while the pointer illustrated in Fig. 5 is a backpointer which points to the past as it were. The reason for this is that according to
5 the present embodiment the LATM header is always written into the bit stream after the current time section has been processed by the AAC encoder, although AAC data may still have to be written into the bit stream from preceding time sections.

10

It is further noted that the pointer 314 is deliberately drawn interrupted below the Celp block 2 as it does neither consider the length of the CELP block 2 nor the length of the CELP block 1 as this data has of course
15 nothing to do with the bit savings bank of the AAC encoder. Further, no header data and bits of possibly present further layers are considered.

In the decoder first of all an extraction of the CELP
20 frames from the bit stream is performed which is easily possible as the same are for example arranged equidistantly and comprise a fixed length.

In the LATM header, however, length and distance of all
25 Celp blocks may be signaled so that in every case a direct decoding is possible.

Thereby, the parts of the output data of the AAC encoder of the directly preceding time section which were so to
30 speak separated by the CELP block 2 are jointed again and the LATM header 306 so to speak moves to the beginning of the pointer 314, so that the decoder knowing the length of

the pointer 314 knows when the data of the directly preceding time section are over in order to then decode the directly preceding time section together with the Celp data blocks present for the same with full audio quality
5 when these data is completely read in.

In contrast to the case illustrated in Fig. 2c, in which an LATM header is followed both by the output data blocks of the first encoder as well as the output data block of
10 the second encoder, now on the one hand a shift from the output data blocks of the first encoder forward in the bit stream may be performed by the variable core frame offset, while by the arrow 314 (max bufferfullness -
bufferfullness) a shift of the output data block of the
15 second encoder to the back of the scalable data stream may be achieved, so that the bit savings bank function may be implemented easily and safely also in the scalable data stream, while the basic raster of the bit stream is maintained by the successive LATM determining data blocks
20 which are always written when the AAC encoder has encoded a time section and which therefore may serve as a reference point also when a major part of the data in the frame designated by an LATM header originate on the one hand from the next time section (regarding the CELP
25 frames) or, however, from the preceding time section (regarding the AAC frame), as it is illustrated in the last row in Fig. 3, wherein the respective shifts are communicated, however, to a decoder by two variables additionally to be transmitted in the bit stream.

30

For purposes of illustration the last row of Fig. 3 describes the case, as it has been discussed, in which the

LATM header 306 is written into the bit stream immediately after it has been generated, so that the LATM header 306 is followed by output data of the second encoder 312 of the preceding time section, wherein the output data of the second encoder for the current time section which the LATM header 306 refers to only follow after a distance in the transmission direction behind the LATM header, wherein the distance is given by the difference between max bufferfullness and bufferfullness, as it is illustrated in Fig. 3.

In contrast to this, according to the present invention, as it is illustrated referring to Fig. 2e, the LATM header 306 is not written anymore when it has been generated but is written delayed by a period of time which corresponds to max bufferfullness. According to the invention, the LATM header 306 would therefore stand behind a position 330 within the bit stream depending on the value of bufferfullness and the forward-pointer 314 is replaced by a backward-pointer (260 in Fig. 2e).

According to the invention the arrangement selected in the Fig. 2c and 2d and also in Fig. 3 is discarded in which a CELP block immediately follows the LATM header.

25

Instead of that, preferably the following priority distribution is preferred when writing data into the scalable bit stream in order to achieve a reduced-delay decoding of the first scaling layer as well as a reduced-delay decoding of the second scaling layer.

30

The output data blocks of the first encoder enjoy a high priority. Always when an output data block of the first encoder is completely written, this output data block is written into the bit stream. From this the equidistant
5 raster of output data blocks of the first encoder automatically results which further have an equal length when using a CELP encoder.

If no output data of the first encoder to be written are
10 currently present, output data of the AAC encoder for the preceding time section of the input signal is written into the bit stream until no corresponding data is present anymore. Only then the writing of the output data of the AAC encoder for the current section is started. The
15 writing of this output data into the bit stream is obviously always interrupted when the output data of the first encoder are available again, as it may be seen in Fig. 2e.

20 The writing of the output data of the AAC encoder for the current time section is further also interrupted when an LATM header is complete and the same has been delayed by max bufferfullness 350 (Fig. 2e). The scalable bit stream is complete when the corresponding values for
25 bufferfullness 260 and offset 270 have been entered into the bit stream either separately or via the determining data block.

In the following, reference is made to a decoding of a bit
30 stream generated this way. When the decoder is only interested in the first scaling layer, i.e. the output data blocks of the first encoder (CELP encoder), then it

will simply take one CELP block after the other from the bit stream and decode the same, without consideration for the LATM header or the AAC data. As the CELP blocks are preferably written into the bit stream immediately after
5 their creation, a reduced-delay decoding of the CELP blocks is guaranteed.

When the decoder wishes a decoding both of the first as well as the second scaling layer, i.e. wants to achieve an
10 audio signal with a high quality, then he need to achieve the association between the CELP blocks and the several AAC block(s) for a superframe, i.e. for a certain number of samples, wherein if necessary a core coder delay (34 of Fig. 1a) is to be considered when the current time section
15 of the input signal of the AAC encoder regarding a superframe is shifted from the current time section of the CELP encoder.

This is performed by the decoder buffering the bit stream
20 until it hits an LATM header, e.g. the header 200 of Fig. 2e. Knowing the offset 270, the decoder may then determine which output data blocks of the first encoder belong to the LATM header 200. Considering the variable bufferfullness the decoder further knows where in the data
25 stored in the decoder input buffer the AAC frame of the time section begins that the LATM header refers to. In the case of bufferfullness equal max already the whole interesting AAC frame is contained in the decoder input buffer. In the case of bufferfullness equal 0, the
30 interesting AAC frame begins immediately behind the LATM header, so that the decoder may begin to decode without delay using the data already stored in the input buffer or

also using a part of the data stored in the input buffer and using a directly arriving part of data which stands behind the LATM header in the transmission direction. The bit savings bank size is therefore signaled only
5 implicitly by the position of the determining data block with reference to the payload data in the bit stream, without any side information being required. In this case also the stage with a variable delay in the decoder (block 68 of Fig. 1b) and the line 70 of Fig. 1b are disposed of.

Claims

1. Method for generating a scalable data stream from at least one block of output data of a first encoder and at least one block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal in the first encoder, wherein the number of samples defines a current section of the input signal for the first encoder, and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal in the second encoder, wherein the number of samples represents a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, comprising:

when a block of output data of the first encoder is available, writing the at least one block of output data of the first encoder into the scalable data stream;

if output data of the second encoder for a directly preceding section of the input signal for the second encoder is available, writing the output data of the second encoder for the directly preceding section of the input signal for the second encoder in the transmission direction behind a block of output data of the first encoder;

if output data of the second encoder for the current section of the second encoder is available, writing the output data of the second encoder in the transmission direction behind the output data of the second encoder for a directly preceding section of the input signal for the second encoder into the scalable data stream;

generating a header data block, when the block of output data of the second encoder for the current section of the second encoder is complete, and writing the header data block delayed by a period of time with regard to the generation of the header data block into the scalable data stream, wherein the period of time is smaller or equal to a delay which corresponds to the maximum size of the bit savings bank of the second encoder; and

writing buffer information into the scalable data stream which indicates where the beginning of the output data of the second encoder for the current section of the input signal for the second encoder is with regard to the header data block.

2. Method according to claim 1,

wherein the period of time is equal to a delay which corresponds to the maximum size of the bit savings bank for the second encoder, and

wherein the buffer information corresponds to the current level of the bit savings bank for the current section of the input signal for the second encoder.

3. Method according to claim 1 or 2,

wherein in case of a concurrent availability of a header data block, a block of output data of the first encoder or a block of output data of the second encoder, a writing of data into the scalable data stream is determined by priorities associated to the data,

wherein the header data block is written into the scalable data stream with a high priority,

wherein the blocks of output data of the first encoder are written into the scalable data stream with a lower priority, and

wherein the at least one block of output data of the second encoder for a directly preceding section of the input signal is written with a higher priority into the scalable data stream than the at least one block of output data of the second encoder for the current section.

4. Method according to claim 1, 2, or 3, wherein the first encoder provides at least two blocks for a number of samples, wherein the method further comprises:

writing offset information into the scalable data stream, which indicates, how many blocks of output data of the first encoder in transmission direction before the header data block belong to the current section of the first encoder.

5. Scalable encoder, comprising:

a first encoder for generating a block of output data for the first encoder;

a second encoder comprising a bit savings bank, wherein the bit savings bank comprises a maximum size, the second encoder being operative to generate a block of output data for the second encoder, wherein the second encoder further comprises means for adjusting the maximum size of the bit savings bank depending on an initial delay provided for an audio decoder;

a bit stream multiplexer for generating a scalable data stream, wherein the bit stream multiplexer is implemented to

write the block of output data for the first encoder into a scalable data stream,

write the block of output data for the second encoder into the scalable data stream;

generate a header data block after the block of output data of the second encoder has been output by the second encoder,

write the header data block into the scalable data stream delayed by a period of time with regard to the generation of the header data block, wherein the period of time corresponds the maximum size of the bit savings bank, and

write buffer information into the scalable data stream which indicates the beginning of the output data of the second encoder in the scalable data stream in the transmission direction of the scalable data stream before the header data block, wherein the buffer information corresponds to a current level of the bit savings bank.

6. Device for generating a scalable data stream from at least one block of output data of a first encoder and at least one block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal into the first encoder, wherein the number of samples defines a current section of the input signal for the first encoder and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal into the second encoder, wherein the number of samples represents a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, comprising:

means for writing a block of output data of the first encoder into the scalable data stream, when a block of output data of the first encoder is available;

means for writing output data of the second encoder for a directly preceding section of the input signal for the second encoder in transmission direction behind a block of output data of the first encoder if the output data of the second encoder for the directly preceding section of the input signal are available for the second encoder;

means for writing output data of the second encoder for the current section of the time signal for the second encoder in transmission direction behind the output data of the second encoder for a directly preceding section of the input signal for the second encoder into the scalable data stream when the output data of the second encoder is available for the current section of the second encoder;

means for generating a header data block when the block of output data of the second encoder for the current section of the second encoder is complete, and for writing the header data block delayed by a period of time with regard to the generation of the header data block, wherein the period of time is smaller or equal to a delay which corresponds to the maximum size of the bit savings bank of the second encoder; and

means for writing buffer information into the scalable data stream which indicates where the beginning of the output data of the second encoder is for the current section of the second encoder with regard to the header data block.

7. Method for decoding a scalable data stream from at least one block of output data of a first encoder and at least one block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal into the first encoder, wherein the number of samples define a current section of the input signal for the first decoder and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal into the second encoder, wherein the number of

samples represents a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal, and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, wherein the scalable data stream comprises output data of the first encoder, output data of the second encoder for a directly preceding section, output data of the second encoder for the current section, a header data block and buffer information, comprising:

buffering the scalable data stream;

reading the block of output data of the first encoder for the current section of the first encoder;

reading the header data block and the buffer information from the buffered data stream;

determining the beginning of the block of output data of the second encoder for the current section of the second encoder using the buffer information; and

decoding the block of output data of the first encoder and the block of output data of the second encoder, the step of decoding including the step of considering the adjustable period of time by which the current section of the first encoder and the current section of the second encoder are time-shifted in relation to each other.

8. Device for decoding a scalable data stream from at least one block of output data of a first encoder and at least one

block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal into the first encoder, wherein the number of samples define a current section of the input signal for the first encoder and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal into the second encoder, wherein the number of samples represent a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, wherein the scalable data stream comprises output data of the first encoder, output data of the second encoder for a directly preceding section, output data of the second encoder for a current section, a header data block and buffer information, comprising:

means for buffering the scalable data stream;

means for reading the block of output data of the first encoder for the current section of the first encoder;

means for reading the header data block and the buffer information from the buffered data stream;

means for determining the beginning of the block of output data of the second encoder for the current section of the second encoder using the buffer information; and

means for decoding the block of output data of the first encoder and the block of output data of the second encoder the means for decoding being operative to consider the adjustable period of time by which the current section of the first encoder and the current section of the second encoder are time-shifted to each other.

9. Method for decoding a scalable data stream from at least one block of output data of a first encoder and at least one block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal into the first encoder, wherein the number of samples define a current section of the input signal for the first decoder and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal into the second encoder, wherein the number of samples represents a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal, and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, wherein the scalable data stream comprises output data of the first encoder, output data of the second encoder for a directly preceding section, output data of the second encoder for the current section, a header data block and buffer information, comprising:

buffering the scalable data stream;

reading the block of output data of the first encoder for the current section of the first encoder;

reading the header data block and the buffer information from the buffered data stream; and

determining the beginning of the block of output data of the second encoder for the current section of the second encoder using the buffer information in order to obtain extracted blocks for a first decoder for decoding the block of output data of the first encoder and a second decoder for decoding the block of output data of the second encoder from the scaled data stream.

10. Device for decoding a scalable data stream from at least one block of output data of a first encoder and at least one block of output data of a second encoder, wherein the second encoder includes a bit savings bank which is defined by a maximum size and a current level of a buffer of bits, wherein the at least one block of output data of the first encoder represents a number of samples of the input signal into the first encoder, wherein the number of samples define a current section of the input signal for the first encoder and wherein the at least one block of output data of the second encoder represents a number of samples of the input signal into the second encoder, wherein the number of samples represent a current section of the input signal for the second encoder, wherein the number of samples for the first encoder and the number of samples for the second encoder are equal and wherein the current sections for the first and the second encoder have the same portion of the input signal or relate to portions of the input signal, which are shifted in time in relation to each other by an adjustable period of time, wherein the scalable data stream

comprises output data of the first encoder, output data of the second encoder for a directly preceding section, output data of the second encoder for a current section, a header data block and buffer information, comprising:

means for buffering the scalable data stream;

means for reading the block of output data of the first encoder for the current section of the first encoder;

means for reading the header data block and the buffer information from the buffered data stream; and

means for determining the beginning of the block of output data of the second encoder for the current section of the second encoder using the buffer information in order to obtain extracted blocks for a first decoder for decoding the block of output data of the first encoder and a second decoder for decoding the block of output data of the second encoder from the scaled data stream.

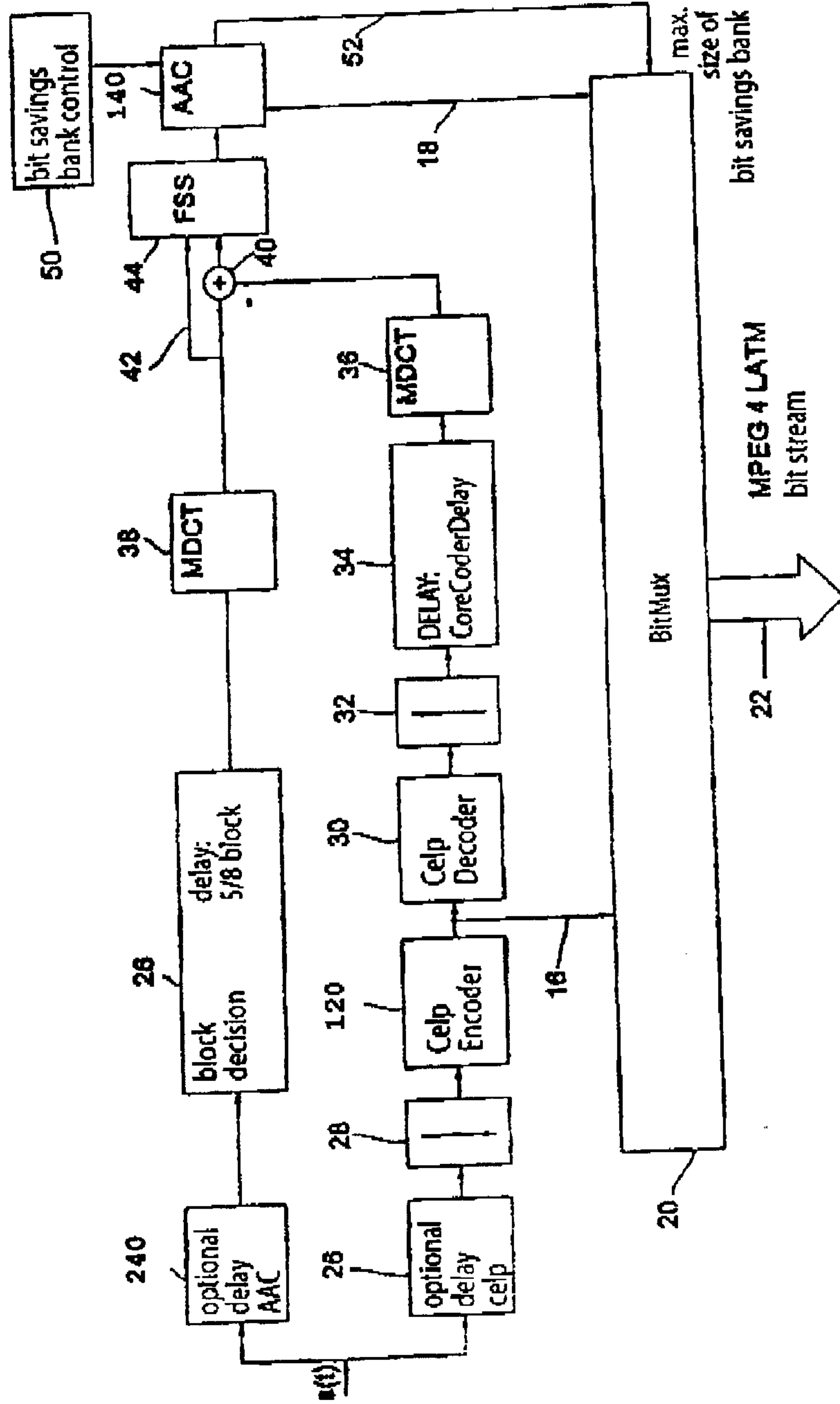


Fig. 1a

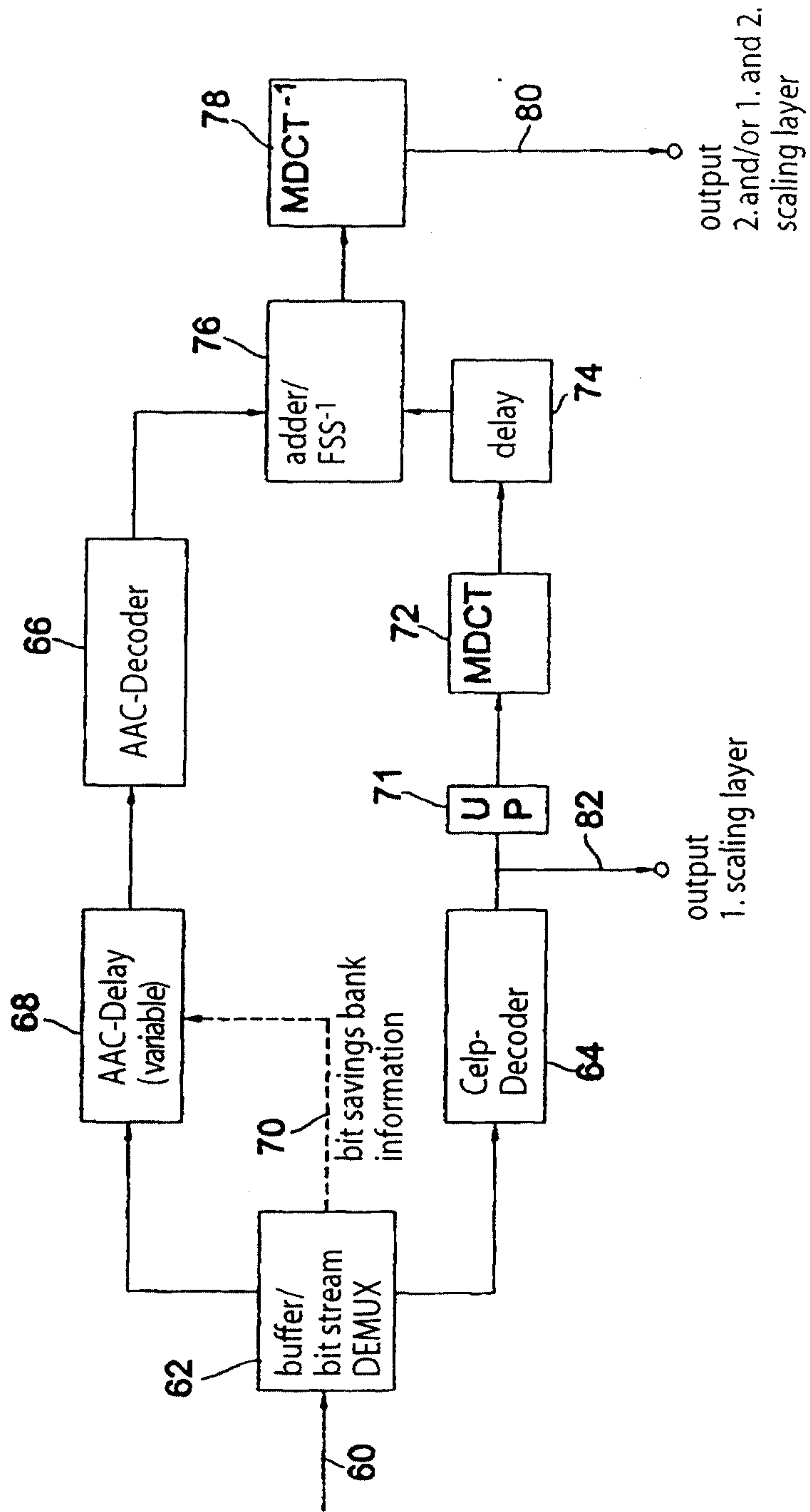


Fig. 1b

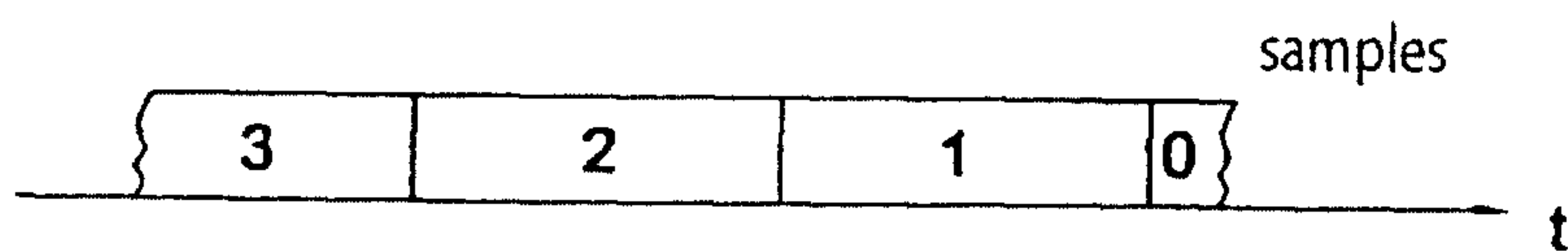


Fig. 2a

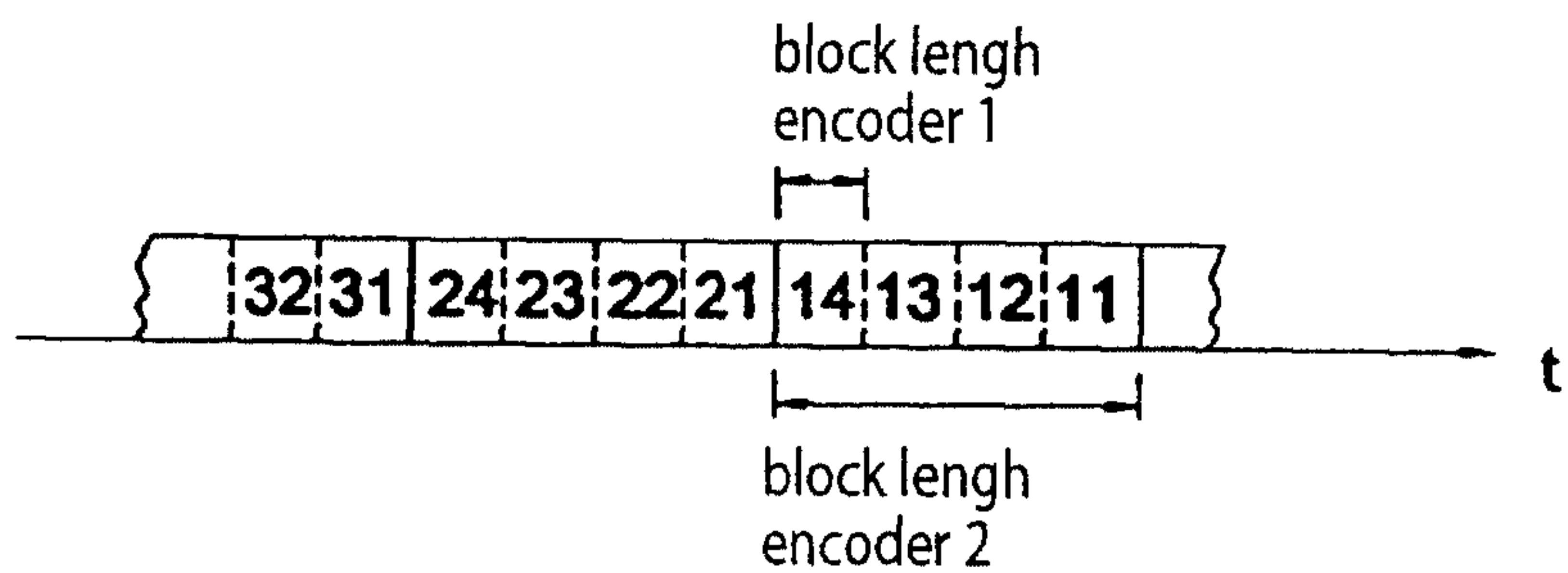


Fig. 2b

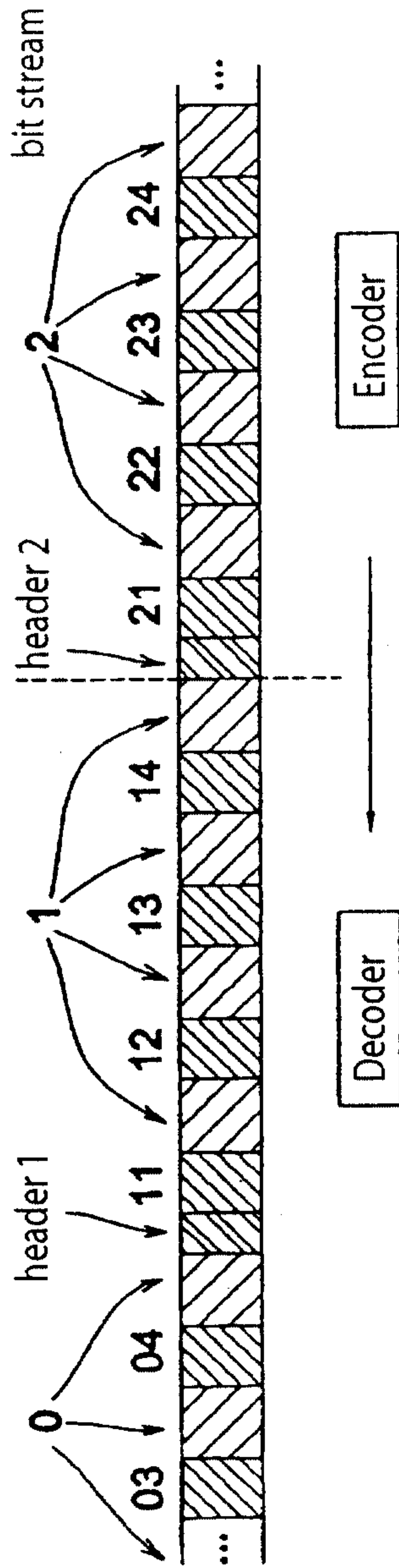


Fig. 2c (PRIOR ART)

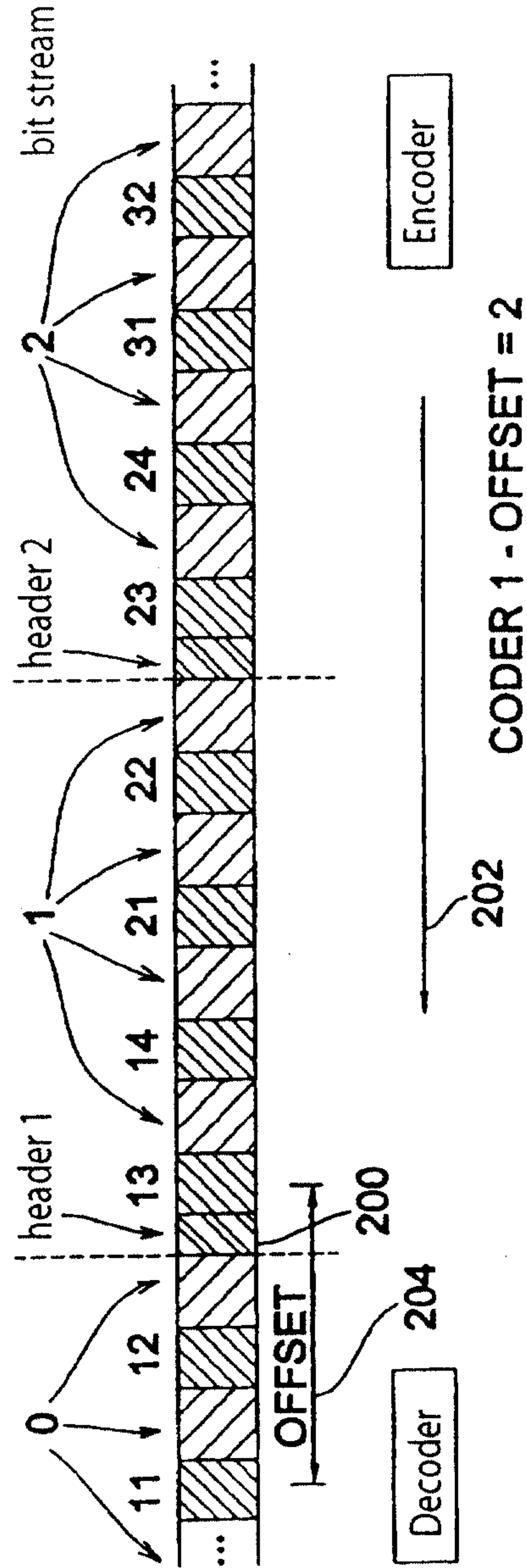


Fig. 2d

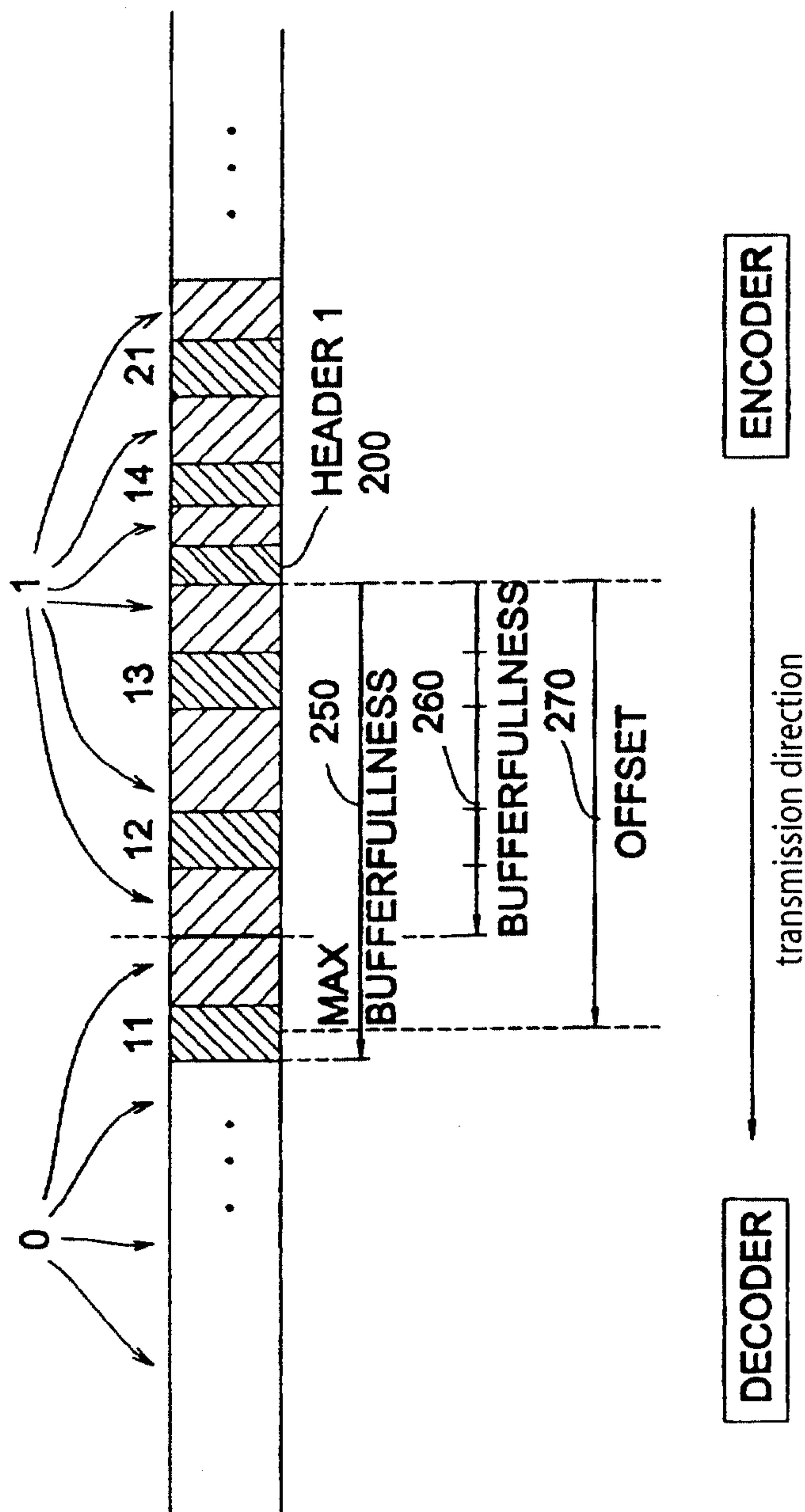


Fig. 2e

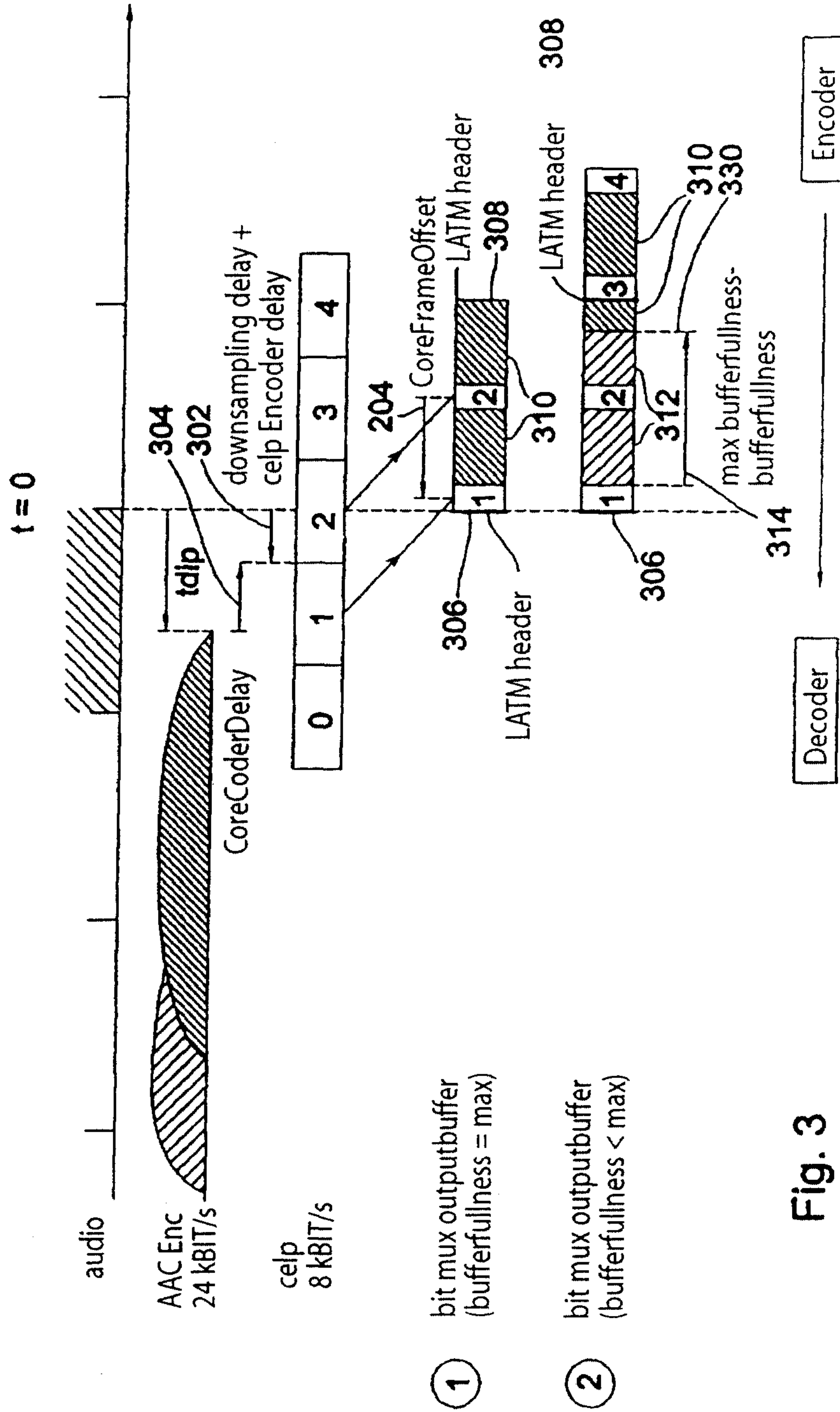


Fig. 3

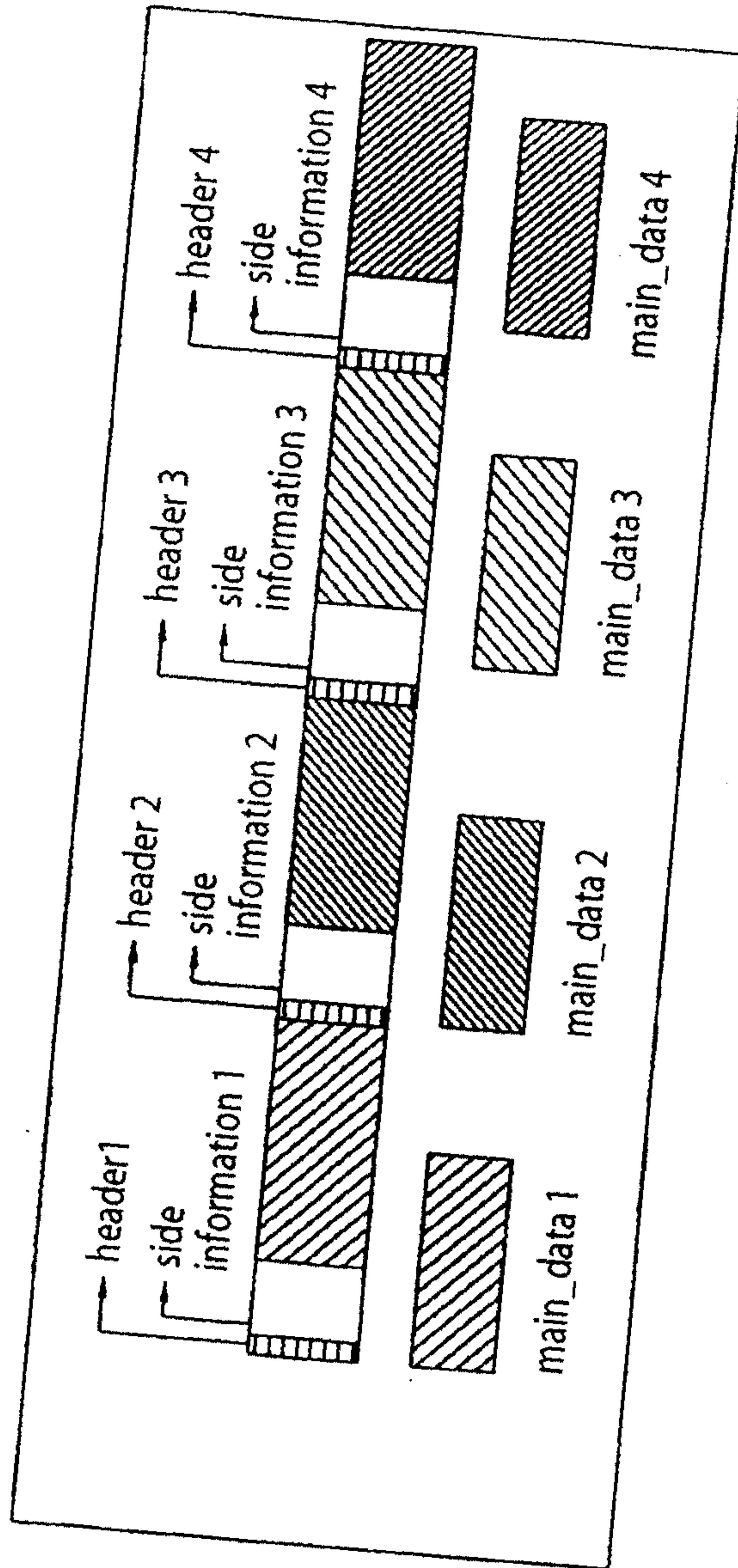


Fig. 4

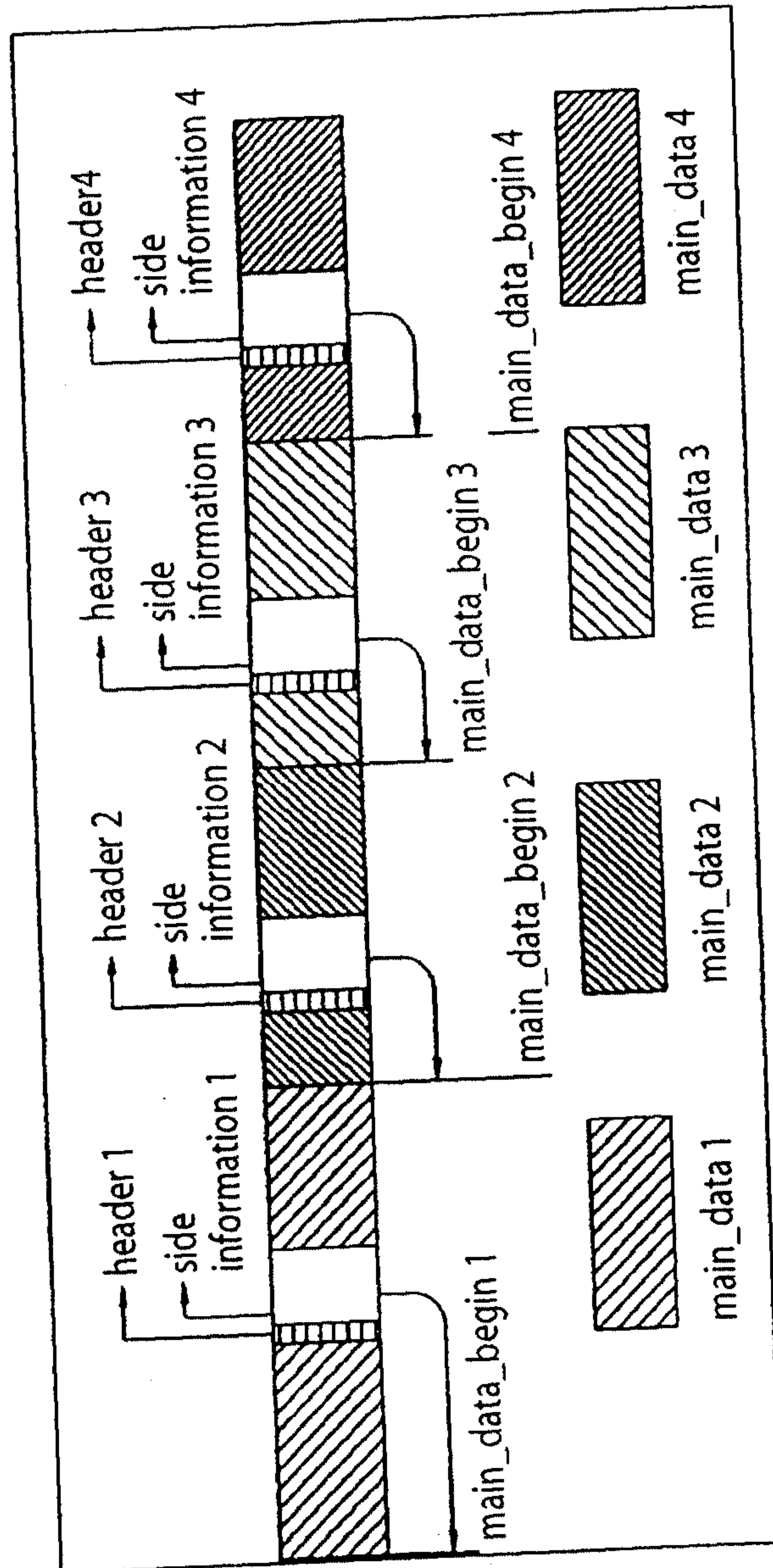


Fig. 5

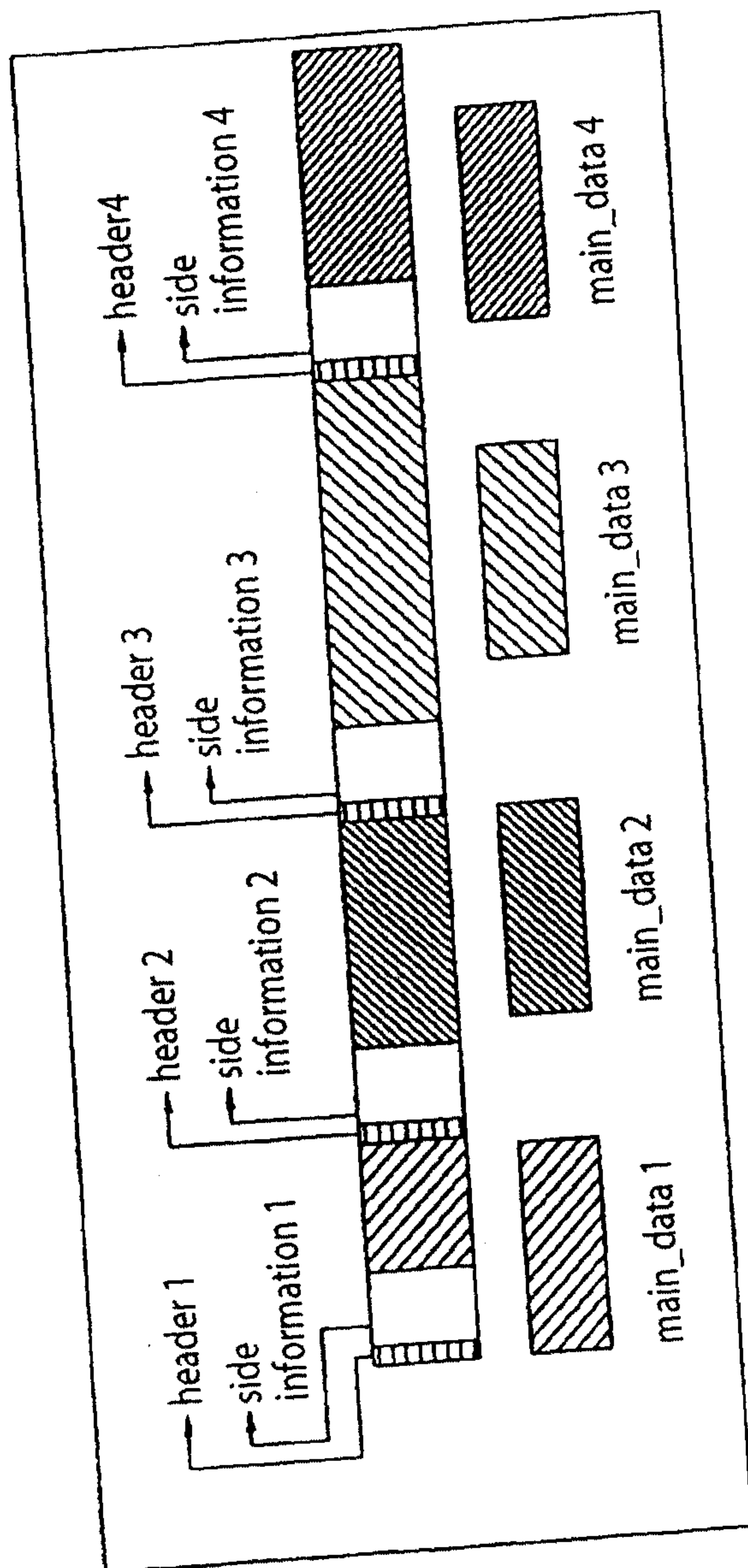


Fig. 6

