BBC RD 1996/2

BBC

Research and Development Report

MPEG-2:
Overview of the systems layer

P.A. Sarginson, B.Sc.

Research & Development Department
Policy & Planning Directorate
THE BRITISH BROADCASTING CORPORATION

*MPEG-2 : Overview of the systems layer

P.A. Sarginson, B.Sc.

Summary

The MPEG-2 Systems specification describes how MPEG-compressed video and audio data streams may be multiplexed together with other data to form a single data stream suitable for digital transmission or storage. This Report introduces the principles and terminology of the MPEG-2 Systems layer. The Report covers three main areas: Firstly, the structure of the multiplexes; secondly, the service information that may be present; thirdly, the system of time stamps and clock references used to synchronise related components of a programme at the decoder.

*Two alternative multiplexes are specified for the MPEG-2 systems layer. The **programme stream** is biased towards the storage and replay of a single programme from a digital storage device while the **transport stream** is intended for the simultaneous delivery of a number of programmes over potentially error-prone channels. Both multiplexes facilitate the inclusion of Programme Specific Information detailing the programme(s) and elementary streams present. The multiplexes also use a system of time stamps and clock references to ensure the synchronous replay of related elementary streams and correct buffer behaviour at a decoder. There are many optional syntax elements and many opportunities to include private (user-defined) syntax extensions enabling either multiplex to be optimised to suit a particular application.*

* This Report is based on two papers prepared for the Institution of Electrical Engineers (IEE) and is published with permission. The papers were written and used by BBC R&D Engineer, P.A. Sarginson; firstly, for the IEE Colloquium at Savoy Place (Jan. 1995):- *MPEG-2: A tutorial introduction to the systems layer*, and published as *IEE Digest 1995/012*; and secondly, for the IEE Residential Course, Durham University (July 1995), from a paper entitled:- *An introduction to the MPEG 2 'transport stream' multiplex.*

Issued under the Authority of

Research & Development Department
Policy & Planning Directorate
BRITISH BROADCASTING CORPORATION

General Manager
Research & Development Department

(R023/025) 1996

MPEG-2: Overview of the systems layer

P.A. Sarginson, B.Sc.

MPEG-2: Overview of the systems layer

P.A. Sarginson, B.Sc.

1. INTRODUCTION

The MPEG-2 Systems Specification[1] describes how MPEG-compressed video and audio data streams may be multiplexed together to form a single data stream. This Report aims to introduce the terminology and the fundamental principles of the MPEG-2 Systems layer.

Programme – Within a broadcast context, a 'programme' is usually taken to mean one of the many entertainment entities shown by a broadcaster during the course of a day such as 'The 9 o'clock news' or 'Eastenders'.* Confusingly, MPEG uses the word 'programme' to mean a *single broadcast service or channel*. So, BBC1** is a programme as far as MPEG is concerned.

Elementary Stream – A programme comprises one or more elementary streams. An elementary stream is the name given to a *single*, digitally-coded and possibly MPEG-compressed *component* of a programme; for example, coded video or audio. The simplest type of programme is a radio service comprising a single audio elementary stream. A traditional television broadcast comprises three elementary streams; one carrying coded video, another carrying coded stereo audio and another carrying teletext data. Future television broadcasts are certain to comprise many more elementary streams. For example, there may be additional video elementary streams carrying the same picture at alternative resolutions and there may be a choice of audio and teletext elementary streams each in a different language.

The MPEG-2 Systems layer describes how elementary streams comprising one or more programmes are multiplexed together to form a single data stream suitable for digital transmission or storage.

The output of an MPEG-2 multiplexer (see Fig. 1) is a contiguous stream of 8-bit-wide data bytes. There are no constraints on the data rate but clearly it must at least equal the total of the combined data rates of the contributing elementary streams. The multiplex may be of fixed or variable data rate and may contain fixed or variable data rate elementary streams.

In addition to the elementary streams, a variety of additional information may be included in the multiplex:

- A system of *time stamps* is specified, ensuring that related elementary streams are replayed in synchronism at a decoder.
- *Tables* of service information may be included which detail network parameters, programmes within the multiplex and the nature of the various elementary streams.
- Support is provided for the control of scrambling, for conditional access, which may be applied to one or more of the elementary streams; though it should be noted that neither the scrambling algorithm nor the access control is specified by MPEG.

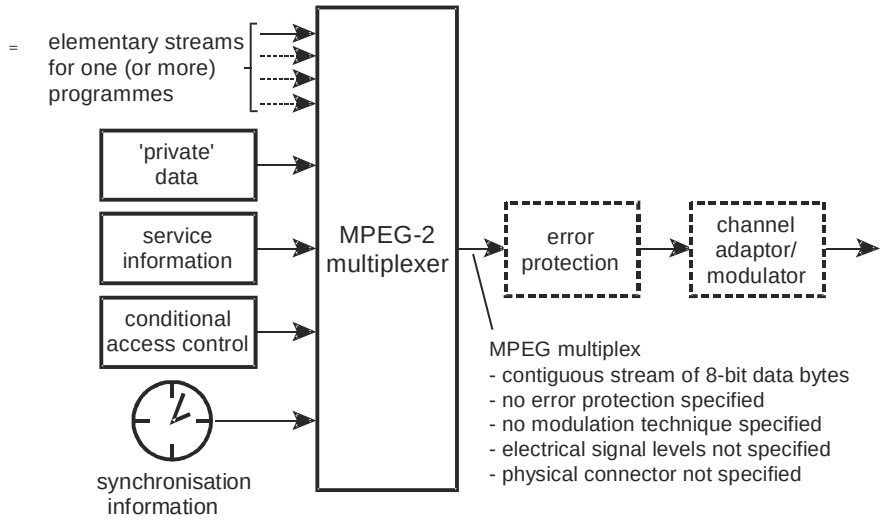


Fig. 1 - An MPEG-2 multiplexer.

= *Audio and video elementary streams have been compressed using the appropriate MPEG algorithm.*

* Two typical listed transmissions on a BBC television channel.

** A BBC television channel.

- A number of additional ‘private data’ channels may be accommodated. Private data is a data stream whose content is not specified by MPEG. Such data streams may be used to carry data services such as: teletext; additional service information specific to a particular network; commands intended to control modulation, and network distribution equipment; any other type of data required by a particular application.

There are some notable multiplex characteristics not specified by MPEG:

- There is no form of error protection within the multiplex. Error protection and the subsequent modulation of the MPEG-2 multiplex is chosen to suit the characteristics of the channel or storage medium and is not specified by MPEG.

- There is no electrical or physical specification for the MPEG-2 multiplex, so a designer may use the signal levels or connector types that best suit the application.

2. PROGRAMME AND TRANSPORT STREAMS

The MPEG-2 Systems Specification defines two alternative multiplexes. One is called a ‘transport stream’, the other is called a ‘programme stream’. Each is optimised for a different set of applications.

2.1 The programme stream

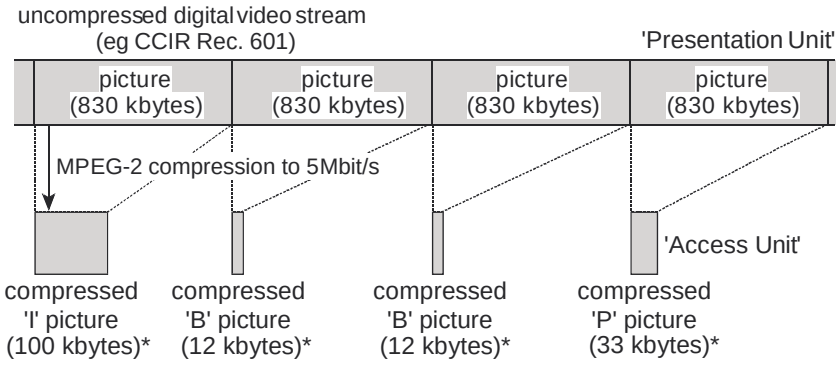
This multiplex is based on the established MPEG-1 multiplex. Like the MPEG-1 multiplex, it can accommodate a *single programme* only and is intended for the storage and retrieval of programme material from digital storage media.

A programme stream is intended for use in error-free environments because it is rather susceptible to errors. There are two reasons for this. Firstly, the programme stream comprises a succession of relatively long and variable length packets. Each packet begins with a packet header. An error occurring in the packet header may cause the loss of the entire packet. As programme stream packets may contain many kilobytes of data, the loss of a single packet may represent the loss or corruption of an entire video frame. Secondly, the variability of the packet lengths means that a decoder cannot predict where one packet will finish and a new one will start. Instead, it must read and interpret the packet length field contained in each packet header. If this packet length field is corrupted by an error, the decoder will lose synchronism with the stream, again resulting in the loss of at least one packet.

2.2 The transport stream

A multiplex devised for *multi-programme* applications such as broadcasting, so that a single transport stream can accommodate many independent programmes. It comprises a succession of 188-byte-long packets called **transport packets**. The use of short, fixed-length packets means that the transport stream is not as susceptible to errors as the programme stream. Further, each 188-byte-long packet is easily given additional error protection by processing it through a standard error protection process such as Reed-Solomon encoding. The improved error resilience of the transport stream means that it has a better chance of surviving the error-prone channels to be found in a broadcast environment, for example.

It might seem that the transport stream is clearly the better of the two multiplexes with its increased error resilience and ability to carry many simultaneous programmes. However, it should be remembered that the transport stream is a more sophisticated multiplex than the programme stream and is consequently more difficult to create and to demultiplex. It is also less like the well-established MPEG-1 multiplex.



** The actual size depends on target bit-rate and complexity of picture*

Fig. 2 - The compression of ‘presentation units’ to yield ‘access units’.

(N.B. The ‘video elementary’ stream consists of a succession *of ‘access units’.)*

3. PRESENTATION UNITS AND ACCESS UNITS

Fig. 2 shows an uncompressed digital video sequence being MPEG-coded to a target data rate of 5 Mbit/s. Each *picture** in its uncompressed form is termed a **presentation unit**. The coder compresses each presentation unit to give a *coded picture* which is termed an **access unit**. Note that video access units are not all the same size. Their size depends on whether they represent an 'I', 'P' or 'B' picture[2, 3] and also on how difficult the picture was to code.

The result of the MPEG-encoding of a video sequence is a succession of video access units; it is this succession of access units that comprises the **video elementary stream**.

Similarly, the result of the MPEG-encoding of audio is a succession of audio access units;[4] it is this succession of access units that comprises an **audio elementary stream**. Each audio access unit typically contains a few tens of milliseconds of compressed audio.

4. PACKETISED ELEMENTARY STREAMS

The next stage in the creation of either of the MPEG-2 multiplexes is to convert each elementary stream into a **Packetised Elementary Stream** (PES). A Packetised Elementary Stream consists entirely of **PES-packets**, as shown in Fig. 3.

A PES-packet consists of a **header** and a **payload**. The payload simply consists of data bytes taken sequentially from the original elementary stream. There is no requirement to align the start of access units and the start of PES-packet payloads. Thus, a new access unit may start at any point in the payload of a PES-packet and it is possible for *several* small access units to be contained in a *single* PES-packet.

PES-packets may be of variable length subject to a maximum length of 64 kBytes. (There is one exception to this: in the case of a video packetised elementary stream when carried in a transport stream, where PES-packets may be of unbounded length.) The designer of an MPEG-2 multiplexer may choose to exploit this flexibility in a number of ways. The possibilities include simply using a fixed PES-packet length or varying the length in order to align the start of access units with the start of PES-packet payloads.

4.1 PES packet header

Fig. 4 shows the fields comprising a **PES-packet header**. The first four (i.e. The 'PES-packet start code prefix' plus the 'stream_id') comprise the PES-packet start code. This combination of 32 bits is guaranteed not to arise in the packetised elementary stream other than at the start of a PES-packet. (There is a general exception to this; the contents of private data are not constrained by MPEG, and as such they may contain emulations of the PES-packet start code.)

4.2 Stream_id byte

The **stream_id byte** distinguishes PES-packets belonging to one elementary stream from those of another within the same programme. MPEG specifies the permitted values for this field, including 32 available for audio elementary streams and 16 for video elementary streams.

4.3 The flags

Flags 1 and **Flags 2** are 'indicator bits' which show the presence or absence of the various optional fields that may be included in a PES-packet header. These optional fields convey additional information about the packetised elementary stream, such as: whether scrambled or not, relative priority, copyright information, and an optional error-check field for the packet.

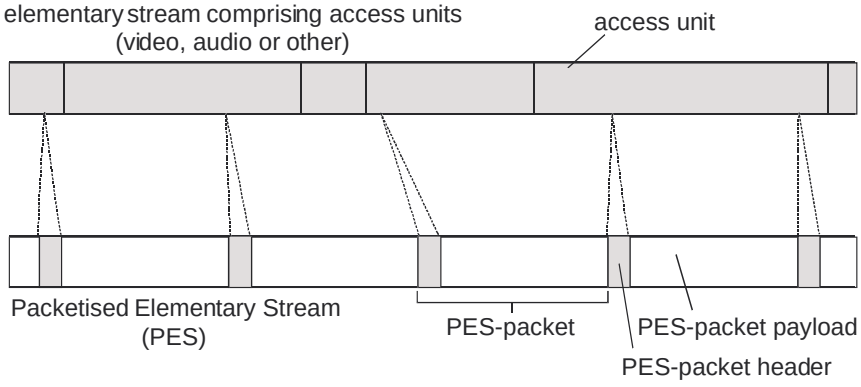


Fig. 3 - Conversion of an elementary stream to a Packetised Elementary Stream.

* A 'picture' can be represented by a frame *or* field. See Ref. 2 for details.

4.4 Time stamps

Of particular importance are the two most significant bits of *flags 2* marked *P* and *D* in Fig. 4. When set, these indicate the presence of a **Presentation Time Stamp** (PTS) field and a **Decoding Time Stamp** (DTS) field respectively, within the PES-packet header. Time stamps are the mechanism provided by the MPEG-2 systems layer to ensure *correct synchronism* between related elementary streams at a decoder.

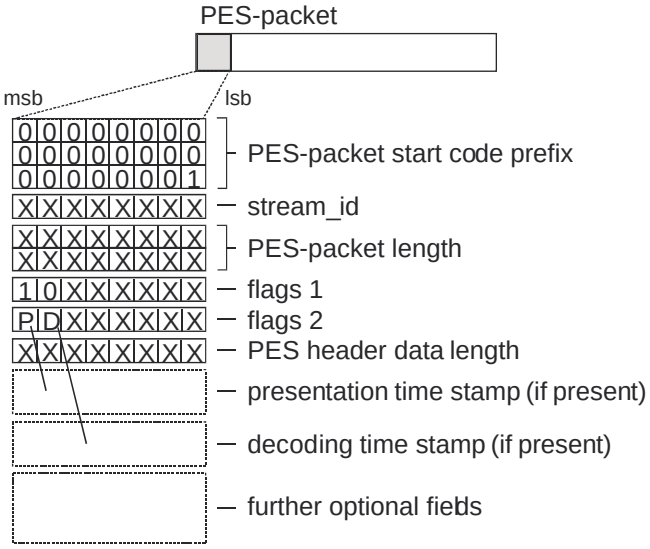


Fig. 4 - A PES-packet header.

4.5 Data length field

The **PES header data length field** is the last of the mandatory bytes in the PES-packet header. Its value gives the number of bytes of optional PES-packet header data present in the PES-packet header before the first byte of the PES-packet payload is reached. (There are 25 optional PES-packet header fields which between them may total nearly 200 bytes of additional data.)

5. THE PROGRAMME STREAM MULTIPLEX

In a programme stream, PES-packets that are derived from the contributing elementary streams are organised into 'packs' (see Fig. 5). A pack comprises a 'pack-header', an optional 'system-header' and any number of PES-packets taken from any of the contributing elementary streams, in any order. There is no constraint on the length of a pack, except that a pack header must occur at least every 0.7 seconds within the program stream, as the pack header contains important timing information (the **system clock reference**). The system header contains a summary of the characteristics of the programme stream such as: its maximum data rate; the number of contributing video and audio elementary streams; further timing information. A decoder may use the information contained in a system header to determine whether it is capable of decoding the programme stream or not.

6. THE TRANSPORT STREAM MULTIPLEX

The transport stream multiplex consists entirely of short, fixed length transport packets (Fig. 6). A transport packet (see Fig. 7) is always 188 bytes long. It comprises a **4-byte header** followed by an **adaptation field** or a **payload** or both. In a transport stream, the PES-packets from the various elementary streams are each divided among the payload parts of a number of transport packets.

Fig. 8 shows how each PES-packet is divided into the payloads of a number of transport packets. The process is subject to two constraints:

1. The first byte of each PES-packet must become the first byte of a transport packet payload.

2. Only data taken from one PES-packet may be carried in any one transport packet.

A PES-packet is unlikely to fill the payloads of an integer number of transport packets, exactly. As shown in Fig. 8, it will often be the case that there are insufficient bytes to completely fill the payload of the final transport packet. So as not to contravene the two constraints of the previous paragraph, the excess space is deliberately wasted by including an adaptation field of appropriate length for this particular transport packet. This wastage can be minimised through careful choice of PES-packet length. This also provides an argument for the use of long PES-packet lengths, as this will ensure that a greater proportion of transport packets are completely filled.

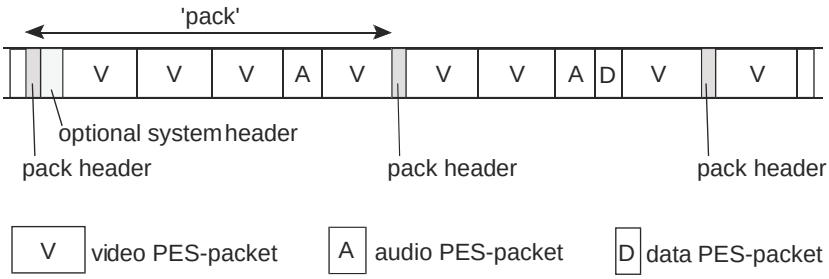


Fig. 5 - Structure of the MPEG-2 'Programme Stream' multiplex.

All the packetised elementary streams that are to be multiplexed together are converted to transport packets in this way. The resulting transport packets are then output sequentially to form an MPEG-2 transport stream. Transport packets containing service information, and 'null' transport packets used to soak-up any spare multiplex capacity, may also appear in the transport stream. There are no constraints on the order in which transport packets appear within the multiplex except that the chronological order of packets belonging to the same elementary stream must be preserved.

6.1 The transport packet header

A transport packet begins with a 4-byte header; the structure of this is shown in Fig. 9. Of the various fields it contains, four are particularly important:

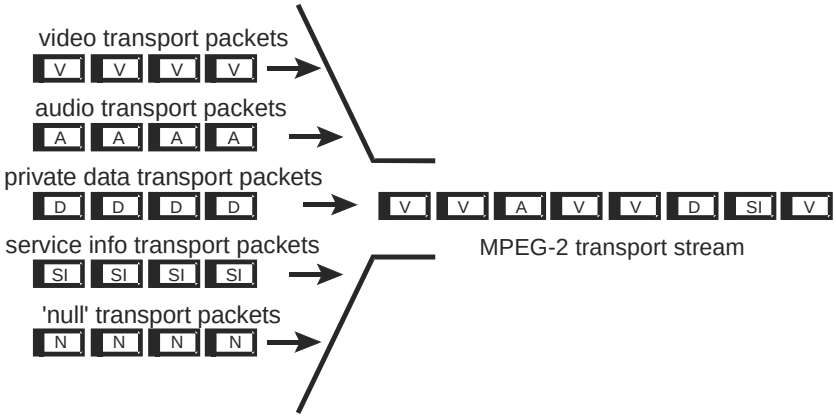


Fig. 6 - Transport packets derived from an almost limitless number of elementary streams may be freely mixed to form a transport stream.

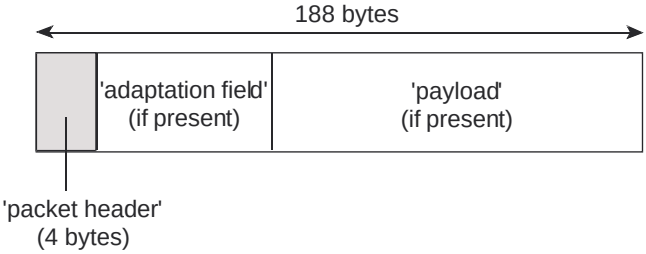


Fig. 7 - The structure of a transport packet.

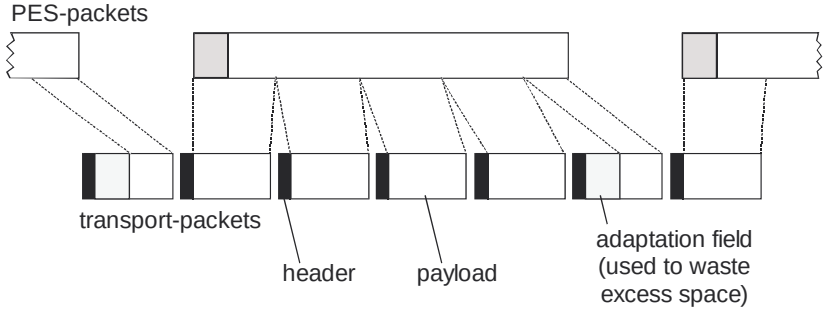


Fig. 8 - Dividing a PES-packet into a number of transport packets.

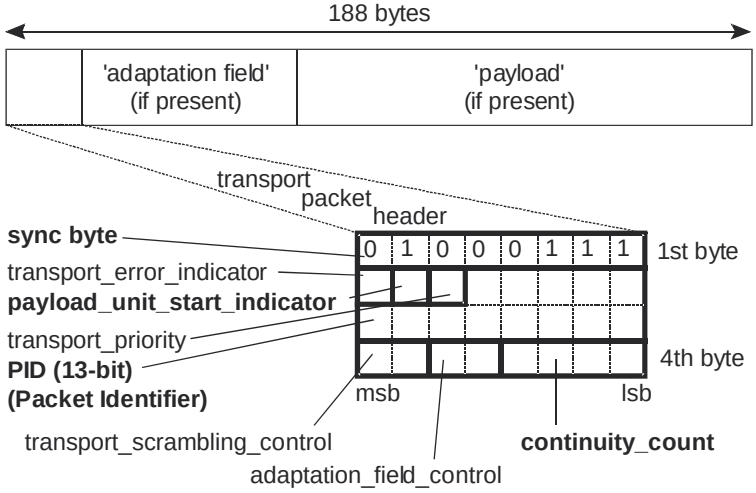


Fig. 9 - The structure of a transport packet header.

- The first byte of the packet header is a **sync-byte** having the value 47 (hex). This value is *not* unique within a transport packet and can quite naturally occur in other fields of the transport packet. However, the fact that a sync byte will occur every 188 bytes within a transport stream enables a suitably designed decoder to lock onto this repetition and hence to identify the start of each new transport packet.

- A single transport stream may carry many different programmes, each comprising many packetised elementary streams. The 13-bit **Packet Identifier** (PID) field is used to distinguish transport packets containing the data of one elementary stream from those carrying the data of other elementary streams. Of the 2^{13} possible values, 17 are reserved for special purposes. This leaves 8,175 values that may be assigned to the various elementary streams and represents the maximum number of elementary streams that can be accommodated in a single transport stream. It is the responsibility of the multiplexer to ensure that each elementary stream is awarded a unique PID value. Other than this, MPEG does not constrain which PID value is assigned to a particular elementary stream.

- The **payload_unit_start_indicator** is set to show that there is something special about the first byte of the payload part of the transport packet. For example, the bit is set if the first byte of the payload is also the first byte of a PES-packet.

- Finally, the **continuity count field** is incremented between successive transport packets belonging to the same elementary stream. This enables a decoder to detect the loss or gain of a transport packet and hopefully conceal the errors that might otherwise result from such an event.

6.2 Programme specific information (PSI)

In a transport stream, each transport packet is tagged with an appropriate PID value indicating to which elementary stream its payload belongs. There may be many elementary streams comprising many different programmes. How can a decoder determine which elementary streams belong to each programme? The answer is to include additional information within the transport stream to explicitly state the relationship between the available programmes and the PID values of their component elementary streams. Such information is called **Programme Specific Information** (PSI) and must be present in every transport stream.

The Programme Specific Information specified for the MPEG-2 systems layer comprises four types of table:– The **Programme Map Table** (PMT), the **Programme Association Table** (PAT), the **Network Information Table** (NIT) and the **Conditional Access Table** (CAT). *Note*, that each of these tables may be carried as the payload of one or more transport packets and hence accommodated in a transport stream.

6.2.1 Programme map table (PMT)

Every programme carried in a transport stream has a Programme Map Table associated with it. This table gives details about the programme and the elementary streams that comprise it. Fig.10 shows an example programme map table. From this table, a decoder can determine that the coded video elementary stream is being carried in transport packets tagged with PID = 726 and that an English language audio elementary stream is available from transport packets tagged with the PID value 57.

The basic programme map table may be embellished with some of the many **descriptors** specified within the MPEG-2 systems specification. They convey further information about a programme or its component elementary streams. The descriptors include video encoding parameters, audio encoding parameters, language identification, pan-and-scan information, conditional access details, copyright information and so on. A broadcaster or other user may define additional private descriptors if required.

6.2.2 Programme association table (PAS)

A complete list of all the programmes available in a transport stream is maintained in the Programme Association Table. This table is easily found as it always has the PID value 0. Each programme is listed along with the PID value of the transport packets that contain its Programme Map Table. For example, in the Programme Association Table of Fig. 11, the Programme Map Table for programme 3 is found in transport packets with PID value 1127.

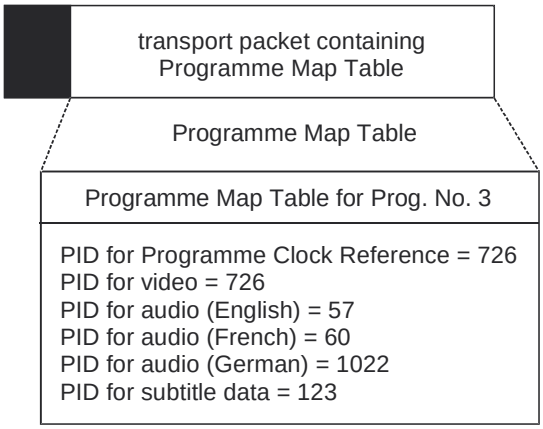


Fig. 10 - Representation of a Programme Map Table.

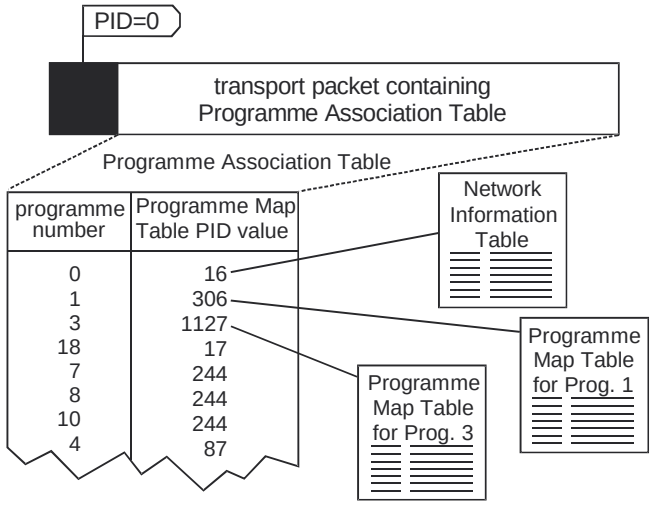


Fig. 11 - A Programme Association Table.

It is permitted for a single Programme Map Table to contain the details of more than one programme. This is of particular advantage when each programme definition is short, as the combined Programme Map Table may still fit within the payload of a single transport packet which is more efficient than the use of a separate transport packet for each of the programme definitions.

6.2.3 Network information table (NIT)

The programme number zero has special meaning within the Programme Association Table. It is used to point the way to the Network Information Table. This table is optional and its contents are private (i.e. defined by the broadcaster/user and not MPEG). Where present, the table is intended to provide information about the physical network carrying the transport stream such as channel frequencies, satellite transponder details, modulation characteristics, service originator, service name and details of alternative networks available.

6.2.4 Conditional access table (CAT)

Finally, there is the Conditional Access Table. If any elementary streams within a transport stream are scrambled, then a Conditional Access Table must be present. The table provides details of the scrambling system(s) in use and provides the PID values of transport packets that contain the conditional access management and entitlement information. The format of this information is not specified within the MPEG-2 Systems specification as it depends on the type of scrambling system employed.

6.3 Further programme specific information

Programme Specific Information is also defined for use in the programme stream multiplex. As a programme stream may only carry a single programme, all elementary streams present in the multiplex must belong to the same programme. A table called a **Programme Stream Map**' is defined for use in the programme stream and states the type (audio, video, other) of information carried in each elementary stream. It may be embellished with MPEG-defined or private 'descriptors' to provide virtually limitless information about the programme or any of its elementary streams.

7. VIDEO CODER AND DECODER BUFFERS

In Section 3 it was stated that the access units resulting from the MPEG-2 encoding of video are not of constant size. Their length depends on whether they represents an 'I', 'P' or 'B' picture, and on the complexity of the picture material. A video encoder therefore generates a variable number of bits per picture. This variable bit-rate is generally smoothed to a fixed bit-rate by a First-In-First-Out (FIFO) memory known as the **coder buffer**.

Typically (for a 'Main Profile @ Main Level' MPEG-2 *video coder*)[3] the coder buffer has a capacity of 229 376 bytes. The memory locations can be thought of as being arranged around the circumference of a clock face. The 'clock' has two hands called the write pointer and the **read pointer**. The write pointer does not move around the clock with constant angular velocity but jumps around the clock in a series of unequal steps. Each step results from the writing of a complete access unit into the buffer and the size of each step corresponds to the size of the access unit. Following some way behind is the read pointer. But the read pointer travels with constant angular velocity so as to read the data from the buffer at a *constant* bit-rate. A feedback process in the coder ensures that the write pointer is always kept ahead of the read pointer but not so far ahead that there is a danger of it overtaking (that is, lapping) the read pointer.

The function of the coder buffer is therefore to smooth the output of a video coder to a *constant bit-rate*. This now constant bit-rate **video elementary** stream is conveyed via the MPEG-2 transport stream multiplex to a *video decoder*. The decoder also contains a buffer (the **decoder buffer**). The coder buffer and decoder buffer are (theoretically) the same size. In the decoder buffer, it is the write pointer that travels with constant angular velocity as it writes the incoming constant bit-rate elementary stream into the decoder buffer. Following behind, is the read pointer which moves around the circular buffer in an irregular manner, depending on the access unit sizes, making one movement every picture period as each complete access unit is removed from the decoder buffer, and is then decoded and displayed.

In practice, the coder buffer read pointer and decoder

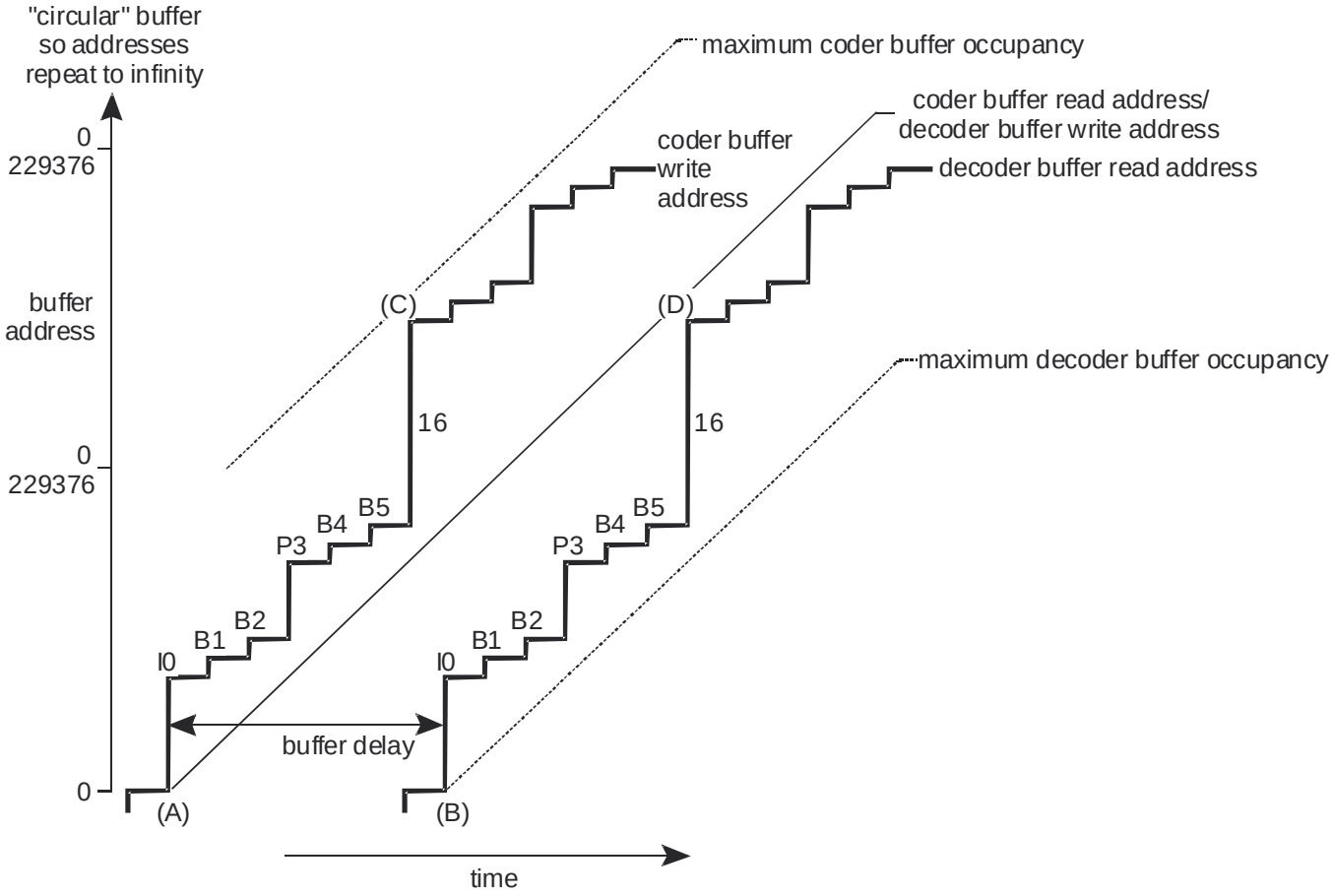


Fig. 12 - Diagram showing video coder and encoder buffer occupancy.

buffer write pointer will not move with truly constant angular velocity but in a succession of small rapid steps. This is because the transport stream conveys the video elementary stream from the coder buffer to the decoder buffer in small packets and not as an uninterrupted fixed-rate data stream However, the effect of packetising the video elementary stream may be neglected for the purposes of this description.

Fig. 12 shows the behaviour of both the coder and decoder buffers on the same diagram. The leftmost of the two stepped waveforms is the coder buffer write address (that is, the position of the coder buffer write pointer within the coder buffer). The first seven steps are labelled I0, B1, B2, P3, B4, B5, I6. These represent the first seven access units being written into the coder buffer. It is assumed that the coder writes each access unit into the buffer very rapidly, resulting in a near-instantaneous rise in buffer occupancy. Thus, the coder buffer occupancy increases as a series of steps (whose heights correspond to the size of each access unit), as each access unit is written into the coder buffer.

Note the dotted line labelled, ‘maximum coder buffer occupancy’. If the coder buffer write address exceeds this line, then the coder buffer has overflowed. (In clock analogy terms, the write pointer has reached a whole revolution ahead of the read pointer and is consequently over-writing data that the read pointer has not yet read.)

In the middle of the diagram is a smooth diagonal line. This is the coder buffer read address which increases at a constant rate with respect to time, corresponding to reading data from the coder buffer at a constant bit-rate. If the coder buffer read address were ever to exceed the coder buffer write address then buffer underflow would occur – it would be empty – (so the read pointer would have caught up with the write pointer).

In Fig. 12, the coder buffer read address line also doubles as the decoder write address, which increases at a constant rate with respect to time, as well. Hence, the diagram is showing that a byte read from coder buffer address *n* is immediately written into decoder buffer address *n*. In practice, it would *take* a *finite and fixed time* to transfer a byte from the coder buffer to the decoder buffer, as the data is conveyed via the multiplex. However, this may be neglected for the purpose of this explanation.

The decoder removes complete access units from the decoder buffer in order to decode and display each frame. The decoder buffer read address is shown in the diagram. This is a stepped waveform where the rise of each step corresponds to a complete access unit being rapidly removed from the decoder buffer.

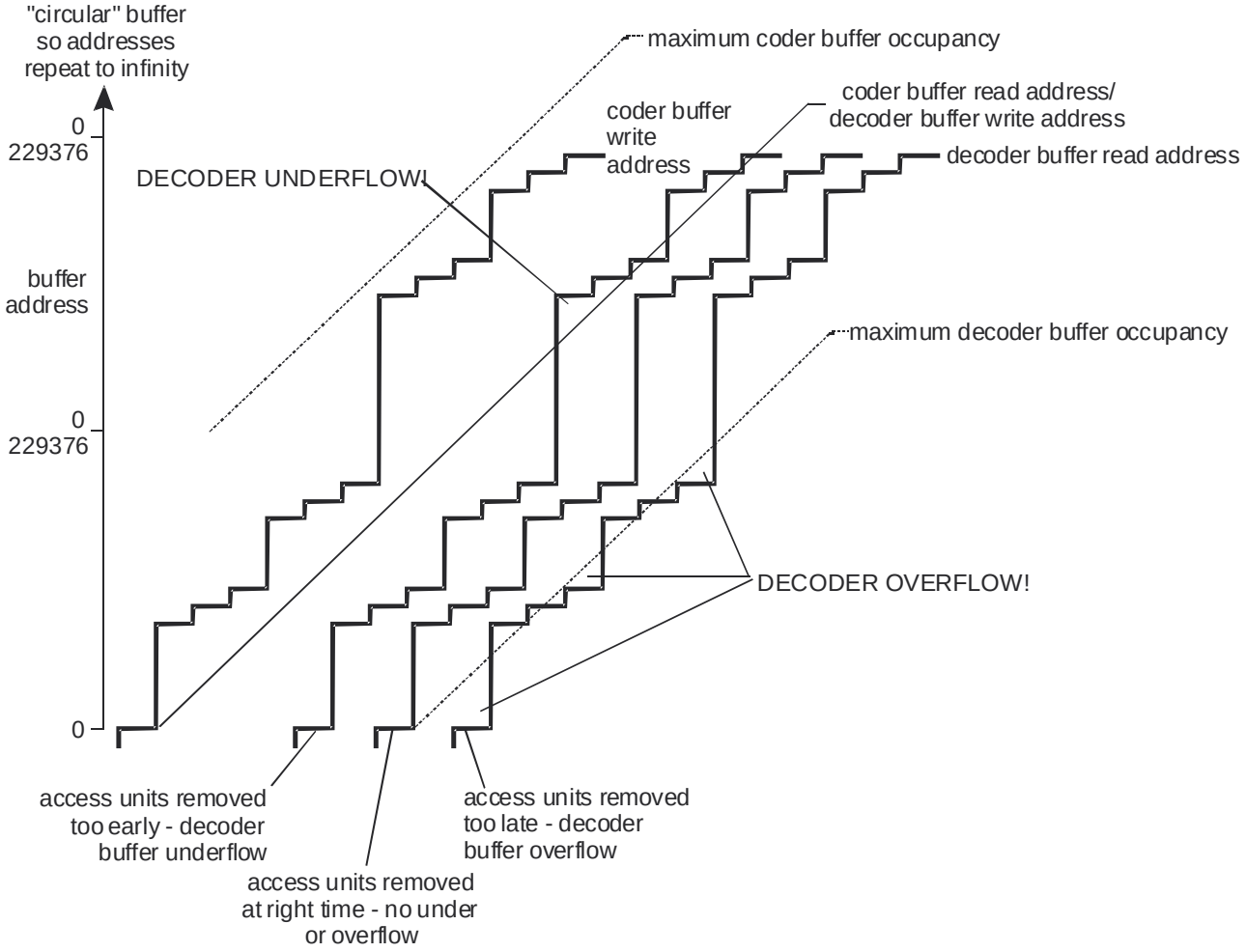


Fig. 13 - Decoder buffer underflow and overflow.

There are a number of important points to note from the diagram:

- The shape of the coder write address and the decoder read address waveforms are identical.

- There is a constant delay between an access unit being written into the coder buffer and being removed from the decoder buffer. The delay is called the **buffer delay** and is equal to the size of the buffer (in bytes) divided by the (constant) rate at which data is transferred from the coder buffer to the decoder buffer (in bytes-per-second).

- A point on the coder write address waveform, where the coder buffer is almost empty (for example, point (A) in Fig. 12), corresponds to a point on the decoder read address waveform where the decoder buffer is almost full (point (B) in Fig.12). Conversely, a coder buffer almost-full condition (point (C)) corresponds to a decoder buffer almost-empty condition (point (D)).

It is essential that the decoder removes access units from the decoder buffer at precisely the right moment. Fig. 13 shows what happens if the decoder reads access units from the buffer too early or too late: too early, results in decoder buffer underflow; too late, results in overflow.

In practice, MPEG-2 defines a system of time stamps and clock references that enable a decoder to determine the precise moment at which an access unit should be removed from the decoder buffer. A video decoder taking account of the timestamps is guaranteed to be presenting decoded pictures in synchronism with any associated decoded audio and will not suffer decoder buffer underflow or overflow.

8. TIME STAMPS AND CLOCK REFERENCES: The basics

Fig. 14 *(overleaf)* shows a simple decoder capable of extracting and replaying video and audio from either a programme stream or a transport stream. The switches are controlled so as to identify the transport packets in a transport stream (or PES-packets in a programme stream) that contain data from the wanted video or audio elementary streams. A switch only permits data

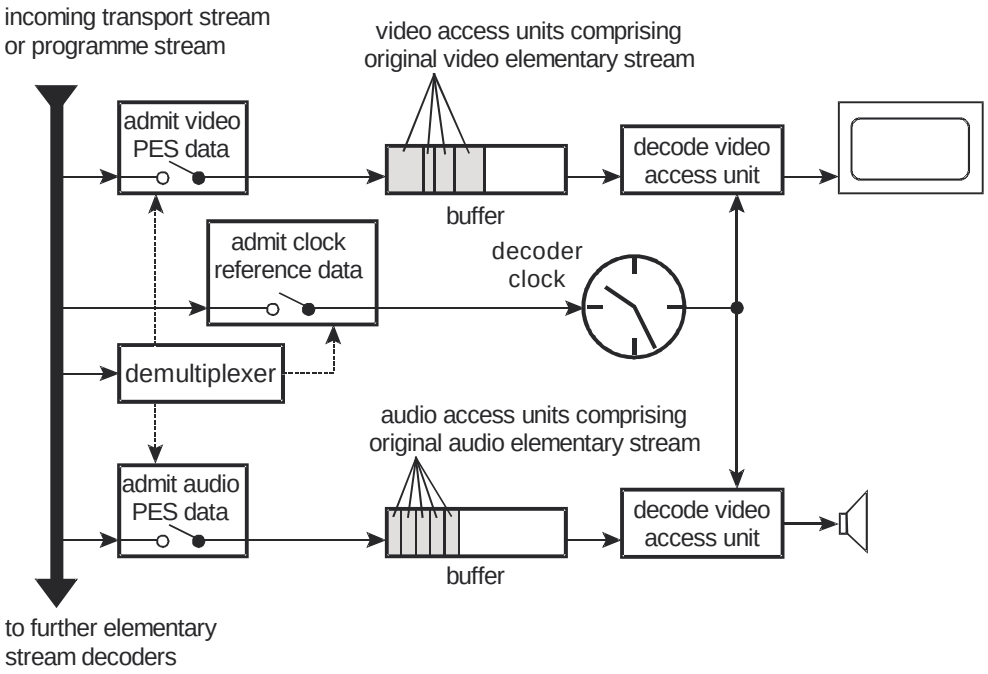


Fig. 14 - A possible decoder.

bytes from the wanted elementary stream to pass into the buffer. Any bytes that are not a part of the original elementary stream are not allowed to pass. As each buffer fills, so the access units, comprising the original elementary stream, accumulate. It should be recollected that a video access unit is simply a single compressed picture. When instructed to do so, the video decoder takes a complete video access unit from the buffer, decodes it and displays the decoded picture on the screen. Similarly, the audio decoder can decode each audio access unit to provide some milliseconds of audio.

How is a decoder enabled to decode each access unit? A time stamp explicitly sets the exact time at which an access unit should be removed from the buffer and decoded. A decoder making correct use of the time stamps will accurately decode the access units in synchronism with the other elementary streams comprising a programme, and its buffer will never underflow (empty) or overflow.

A time stamp is a value representing a time. One function of the multiplexer is to assign time stamps to the access units as they are produced by a coder. In Fig. 15, the video coder has generated three access units and they have been assigned the time stamps 10:27, 10:28 and 10:29 respectively. So in this (rather low frame rate) example, the first access unit is to be decoded at 27 minutes past 10, the second at 28 minutes past 10 and the third at 29 minutes past 10.

In order to allocate time stamps, the coder must have a

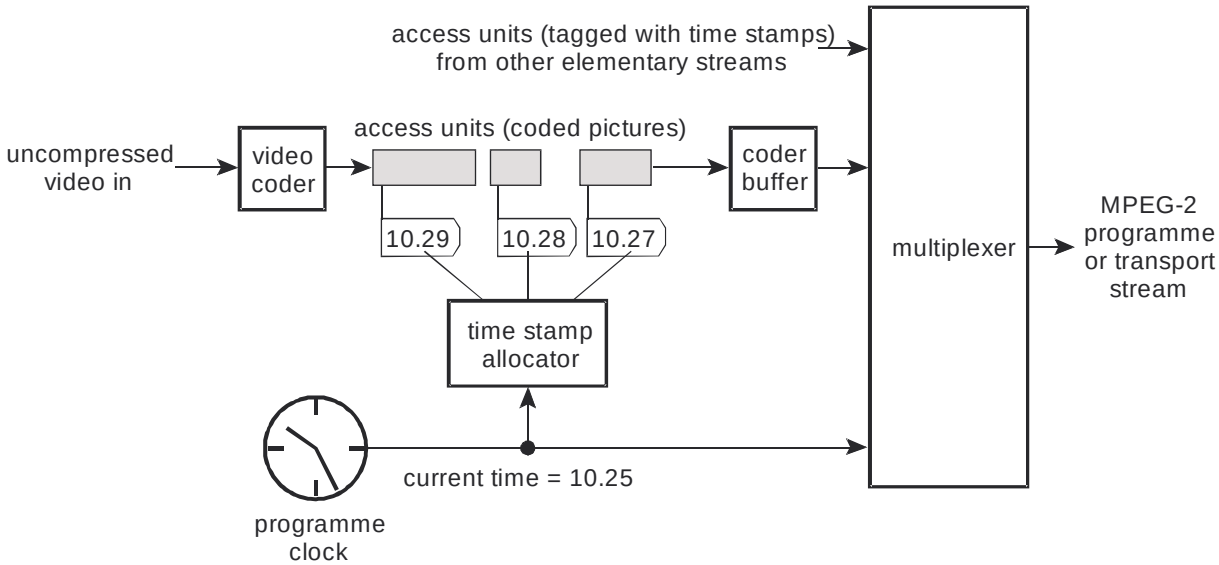


Fig. 15 - Time stamps are allocated to access units by the multiplexer.
(Note the allowances made to access unit times for the decoder clock).

time reference. The current time is maintained by an accurate clock located in the multiplexer. Whenever a new access unit is generated by the coder, the current time is inspected and a time stamp based on the current time is assigned to the new access unit. Note that the time stamp is not simply equal to the current time; it cannot be. Recall that the time stamp represents the time at which the access unit *will* be decoded at the decoder. This clearly must occur at some time in the future. So when a time stamp is first allocated to an access unit by the multiplexer it will always be deliberately offset into the future with respect to the current time.

How far into the future a time stamp should be offset is a subject in itself and depends on a number of factors, including the size of the coder and decoder buffers and the bit rate at which the elementary stream is carried in the multiplex. The offset must be sufficiently large to allow enough time for the access unit to pass through the coder buffer, be conveyed to the decoder via the multiplex, and be fully written into the decoder buffer.

In order to interpret the time stamps, the decoder requires its own accurate clock to determine the appropriate time to decode a particular access unit. It is essential that the decoder's clock is in exact synchronism with the clock at the multiplexer. This is achieved by including regular samples of the current time, according to multiplexer's clock, in the multiplex. The decoder uses these samples to check and correct the decoder clock.

9. TIME STAMPS AND CLOCK REFERENCES: The details

The clocks used at the multiplexer and decoder do not measure time in hours and minutes but in units of 27 MHz expressed as a 42-bit binary number and there is no requirement that the clocks should be related to any national time standard.

In a programme stream which can carry only a single programme, the clock at the multiplexer is called the **System Clock**. Access units in all the elementary streams of the programme are assigned time stamps based on this system clock. Regular samples of the system clock are carried in the programme stream* to enable the decoder clock to maintain synchronism with the system clock. These samples are called **System Clock References** (SCR) and they are encoded in optional fields in the pack headers of the programme stream. A System Clock Reference must appear in the programme stream at least once every 0.7 seconds.

The alternative MPEG-2 multiplex, the transport stream, may contain many independent programmes. Each programme has its own independent clock called a **Programme Clock** which need not be synchronised with the clocks of other programmes. (It is, though, permitted for programmes to share Programme Clocks). Access units comprising a programme are assigned time stamps based on the associated programme clock. Samples of each programme clock called **Programme Clock References** (PCR) are carried in the transport stream. They enable a decoder to synchronise its clock with the programme clock of the programme to be decoded. A Programme Clock Reference for each Programme Clock must appear in the transport stream at least every 0.1 seconds. Exactly which transport packets are carrying the programme clock reference values for a particular programme may be found in the programme map table defining the programme. (See 'PID for Programme Clock Reference' in Fig. 9).

Note the potential confusion here. In a *programme stream*, the clock is called a **System Clock** and a sample of its value is called a **System Clock Reference**. In a *transport stream* the clocks are called **Programme Clocks** and samples of their values are called **Programme Clock References**.** Apart from these differences in naming, there is very little difference between the clocks and their function in either the programme stream or transport stream. But note, from the footnotes on this and the previous page, where the synchronising references are placed for each type of 'stream'.

Time stamps themselves are 33-bit binary values expressed in units of 90 kHz.

9.1 Presentation time stamp (PTS)

There are, in fact, two types of time stamp. The first and most fundamental is the **Presentation Time Stamp** (PTS). It specifies the time at which an access unit should be removed from the decoder buffer, decoded and presented to the viewer. MPEG assumes that this can be performed instantaneously. In practice, the transfer of data into the decoder and the decoding process itself is certain to take some time and it is the responsibility of the decoder designer to compensate for these practicalities.

For many types of elementary stream, Presentation Time Stamps are the only time stamps that will be needed. However, in the significant case of an elementary stream carrying MPEG-coded video then a second type of time stamp called a **Decoding Time Stamp** (DTS) may be required.

* Carried in the PES-packet header; see Section 5.

** Programme clock references are actually carried in the 'adaptation fields' of transport packets.

9.2 Decoding time stamp (DTS)

A **Decoding Time Stamp (DTS)** specifies the time at which an access unit should be removed from the decoder buffer and decoded, but *not* presented to the viewer. Instead, the decoded picture is temporarily stored for presentation at a later time. Such treatment is only necessary for the I and P picture types of an MPEG-coded video sequence where they are separated by **B** type pictures. (See Ref. 2 for an explanation of I, P and B picture types).

A Decoding Time Stamp never occurs on its own and is always accompanied by a Presentation Time Stamp. Here, the Presentation Time Stamp specifies the time at which the decoded picture is eventually released from temporary storage and is presented to the viewer. Thus the Presentation Time Stamp will always be greater (i.e. further in to the future) than its associated Decoding Time Stamp, as the 'presentation' of the picture to the viewer will occur at a later time.

9.3 Time stamp allocation

It is not necessary to allocate time stamps to every access unit. A decoder will usually know in advance the rate at which access units are to be decoded and it is sufficient to provide the occasional time stamp simply to ensure that the decoding process maintains long-term synchronism. The constraint specified by MPEG is that a time stamp should occur at least every 0.7 seconds in a video or audio packetised elementary stream.

In either type of MPEG-2 multiplex, the time stamps are carried as optional fields in the headers of PES-packets (as shown in Fig. 4). If an access unit has a time stamp associated with it then the time stamp is encoded in the header of the PES-packet in which that access unit commences.

10. DISCUSSION

MPEG-2 is sometimes referred to as a 'toolkit' that gives a very full specification of video and audio compression algorithms and an extensive multiplex syntax. Only a limited area of the system is likely to be required for most applications; for example, only the programme stream would, in all probability, be used for storage and retrieval systems. (The programme stream has a specification that is essentially the same as that of the MPEG-1 specification, so that compatibility between the two systems is maintained.) The full specification of MPEG-2 was ready by November, 1994; it was publicly released in early 1995. So only a short time has elapsed since then, and various manufacturing companies are now endeavouring to produce equipment that uses the parts of the multiplex's 'toolkit' that are relevant to their design requirements. The designers can construct the coders in any way that is cost-effective, provided any decoder that is designed strictly to MPEG-2 specification can correctly decode the compressed/multiplexed data. This means that manufacturers of television sets, computer equipment, digital home video etc. can be assured that their MPEG-2 decoding equipment will be valid.

11. REFERENCES

1. ISO/IEC, 1994. Generic Coding of Moving Pictures and Associated Audio: Systems, (MPEG-2 Systems Specification), November, ISO/IEC 13818-1.

2. ELY, S.R., 1996. MPEG VIDEO CODING: A basic tutorial introduction. BBC Research & Development Department Report No. 1996/3.

3. ISO/IEC, 1994. Generic Coding of Moving Pictures and Associated Audio: Video, (MPEG-2 Video Specification), May, ISO/IEC 13818-2.

4. STOLL, G. *and* GILCHRIST, N.H.C., 1996. ISO/IEC MPEG-2 AUDIO: Bit-rate-reduced coding for two channel and multichannel sound. BBC Research & Development Department Report No. 1996/4.

Published by BBC RESEARCH & DEVELOPMENT DEPARTMENT, Kingswood Warren, Tadworth, Surrey, KT20 6NP