ENGINEERING COMMITTEE
Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 20 2004

METHODS FOR
CARRIAGE OF CLOSED CAPTIONS
AND NON-REAL TIME SAMPLED VIDEO

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members, whether used domestically or internationally.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the Standards. Such adopting party assumes all risks associated with adoption of these Standards, and accepts full responsibility for any damage and/or claims arising from the adoption of such Standards.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this standard have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at http://www.scte.org.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

This document defines a standard for the carriage of Vertical Blanking Interval (VBI) services in MPEG-2 compliant bitstreams constructed in accordance with *ISO/IEC 13818-2* (Reference [1]).

1.2 Revisions

This version is the initial release.

1.3 Scope

The sections in this standard describing video user data extensions to MPEG-2 are organized as follows:

- **Section 1**—Provides an introduction
- **Section 2**— Lists normative and informative references
- **Section 3**—Defines the acronyms and abbreviations used in this specification
- **Section 4**—Provides an overview of the VBI services supported
- **Section 5**—Specifies the video bitstream syntax for picture user data extensions
- **Section 6**—Describes the video bitstream semantics for picture user data extensions

2 References

2.1 Normative References

1. ISO/IEC 13818-2:2000: Information technology -- Generic coding of moving pictures and associated audio information – Part 2: Video (MPEG-2 Video)

2. ITU-R BT.601-5, Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios, ITU Radiocommunication Assembly, 1995

3. EIA 608-B, Line 21 Data Services, 2000

2.2 Informative References

4. ITU-R BT.470-6, Conventional Television Systems, ITU Radiocommunication Assembly, 1998

3. Acronyms and Abbreviations

bslbf	bit string left bit first
CEA	Consumer Electronics Association
FCC	Federal Communications Commission
IEC	International Electrotechnical Commission
ISO	International Standards Organization
ITU	International Telecommunication Union
LL	Low Level
lsb	least significant bit
ML	Main Level
MP	Main Profile
MPEG	Moving Picture Experts Group
msb	most significant bit
uilsbfopbl	unsigned integer least significant bit first odd parity bit last
NTSC	National Television System Committee
PAL	Phase Alternate Line
uimsbf	unsigned integer most significant bit first
VBI	Vertical Blanking Interval
VITS	Vertical Interval Test Signal

4. Video User Data Extensions

The video user data extensions to the MPEG-2 syntax and semantics described in this standard provide a means to support VBI services. This new feature allows carriage and reinsertion of VBI data into the same field from which it originates. The VBI enhancements relative to MPEG-2 main and simple profiles include:

        a) Closed captioning for single and multiple lines

        b) Non-real-time sampled video

4.1 Closed Captioning

Many television services carry closed caption information in line 21 (field 1 as well as field 2) of the VBI (Reference [3]). The user data syntax described herein supports the carriage and decoder reinsertion of closed captioning information. This feature requires logic to parse, store and reorder closed caption data as well as to synthesize the Federal Communications Commission (FCC) standard closed caption waveform.

Certain system service providers use the closed caption format to carry additional data in VBI lines other than line 21. The user data syntactic constructs described in this document allow multiple VBI lines per display field, including any standard closed caption usage described in the above paragraph. A compatible decoder shall be able to process and rebuild up to four lines per display field.

4.2 Non-Real-Time Sampled Video

Primarily intended to support Vertical Interval Test Signals (VITS), non-real-time sampled video can potentially be used for other VBI services as well. Non-real-time sampled video supports the carriage and decoder reinsertion of VITS information, with the maximum refresh rate of once in every 22 frames of video (32 pixels of data * 22 = 704 pixels). User data syntactic constructs include logic to parse and store multiple lines of VBI video luma and chroma for this purpose.

5.   Picture User Data Syntactic Extensions

The method used in this document for describing video bitstream syntax is the same as that used in the MPEG-2 International Standard, *ISO/IEC 13818-2 (Reference [1])*. The syntactic extensions to MPEG-2 MP@ML operation for VBI services are implemented using the picture user data syntax defined in subsection 5.2.

5.1 Syntax Conventions and Definitions

5.1.1 Method of Describing Bitstream Syntax

Those *ISO/IEC 13818-2* conventions and definitions that appear in VBI user data syntax are reviewed in the remainder of this subsection.

As exemplified in Table 5–1, this syntax resembles C-code and uses the convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true.

Table 5–1. Bitstream Data Elements and Conditions

while (condition) {
   data_element
...
}
	If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.

do { **data_element** ...	The data element always occurs at least once.
} while (condition)	The data element is repeated until the condition is not true.
if (condition) { **data_element** ...	If the condition is true, then the first group of data elements occurs next in the data stream.

} else {
   data_element
...
}
	If the condition is not true, then the second group of data elements occurs next in the data stream.

for (i = 0; i < n; i++) {
   data_element
...
}
	The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to one for the second occurrence, and so forth.

/* comment ... */	Explanatory comment that may be deleted entirely without in any way altering the syntax.

Each data item in the bitstream appears in bold type and is described by its name, its length in bits, and a mnemonic for its type and order of transmission. The action caused by a decoded data element in a bitstream depends on the value of that data element and on data elements previously

decoded. The constructs in normal type in the above table are used to express the conditions when data elements are present.

A group of data elements may contain nested conditional constructs. For compactness, the { } are omitted when only one data element follows. Array data is represented as follows:

    data_element[n] the n+1th element of an array of data
    data_element[m][the m+1, n+1th element of a two-dimensional array of data
    n]

While the syntax descriptions given in this document are expressed in procedural terms, it should not be assumed that subsection 5.1 implements a satisfactory decoding procedure. In particular, it defines a correct and error-free input bitstream for compatible encoders. Actual decoders must include means to look for start codes in order to begin decoding correctly, and to identify errors, erasures and insertions while decoding. Neither the methods to identify these situations nor the actions to be taken are specified in this document.

5.1.2 Reserved, Forbidden & Marker Bits

The terms *reserved* and *forbidden* are used in the description of some values of several fields in the coded bitstream.

 Reserved—Indicates that the value may be used in the future for ISO/IEC-defined extensions.

 Forbidden—Indicates a value that shall never be used (usually in order to avoid emulation of start codes).

 marker_bit— A marker_bit is a 1-bit field that has the value '1'.

5.1.3 Operators

  +   Addition.
  -    Subtraction (as a binary operator) or negation (as a unary operator).
  ++   Increment.
  - -   Decrement.
  >   Greater than.
  >=   Greater than or equal to.
  <   Less than.
  <=   Less than or equal to.
  ==   Equal to.
  !=   Not equal to.
  =   Assignment operator.

5.1.4 Mnemonics

The following mnemonics are defined to describe the different data types used in the user data syntax described in subsection 5.2:

    bslbf   Bit string, left bit first, where "left" is the order in which bit strings are written in the specification. Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.
    uimsbf   Unsigned integer, most significant bit first.
    uilsbfopbl Unsigned integer least significant bit first odd parity bit last

5.1.5 Start Codes

Start codes are specific bit patterns that do not otherwise occur in the video stream. Each start code consists of the 24-bit start code prefix string '0000 0000 0000 0000 0000 0001' followed by an 8-bit integer that identifies the type of start code as described in ISO/IEC 13818-2. Start codes are always byte aligned, and may be preceded by any number of zero stuffing bits.

5.1.6 Definition of Functions

The following utility functions for picture coding algorithms are defined:.

bytealigned()	returns 1 if the next bit in the bitstream is the first bit in a byte. Otherwise it returns 0.
nextbits()	permits comparison of a bit string with the next bits to be decoded in the bitstream.
next_start_code()	removes any zero bit and zero byte stuffing and locates the next start code as defined in Figure 5-1.

next_start_code() {	No. of bits	Mnemonic
while (!bytealigned())		
zero_bit	1	0
while (nextbits() != '0000 0000 0000 0000 0000 0001')		
zero_byte	8	0000 0000
}		

Figure 5–1. **Next Start Code Function Syntax**

5.2 Picture User Data Syntax

Picture user data syntax to support VBI services is shown in Figure 5-2.

user_data(2) {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
user_data_type_code	8	uimsbf
if (user_data_type_code == '0x03') {		
'1000 000'	7	bslbf – See Note 1
vbi_data_flag	1	bslbf
if (vbi_data_flag) {		
cc_count	5	uimsbf
for (i=0 ; i<cc_count ; i++) {		
cc_priority	2	uimsbf
field_number	2	uimsbf
line_offset	5	uimsbf
cc_data_1[1:8]	8	uilsbfopbl
cc_data_2[1:8]	8	uilsbfopbl
marker_bit	1	bslbf
}		
non_real_time_video_count	4	uimsbf
for (i=0 ; i<non_real_time_video_count ; i++) {		
non_real_time_video_priority	2	uimsbf
sequence_number	2	uimsbf
non_real_time_video_field_number	1	uimsbf
line_offset	5	uimsbf
if (sequence_number != '00') {		
segment_number	5	uimsbf
for (i=0 ; i<32 ; i++) {		
non_real_time_video_y_data[7:0]	8	uimsbf
}		
for (i=0 ; i<16 ; i++) {		
non_real_time_video_cb_data[7:0]	8	uimsbf
non_real_time_video_cr_data[7:0]	8	uimsbf
}		
}		
}		
}		
reserved	n	bslbf
}		
next_start_code()		
}		

Figure 5–2. **Picture User Data Syntax**

Note 1: Please note that implementations that preceded this standard set these bits to '0000 000' instead of '1000 000'. It is recommended that receivers ignore the setting of the leading bit when the other 6 bits are set to '000 000'.

6. Picture User Data Semantic Extensions

The semantic extensions for Vertical Blanking Interval (VBI) service enhancements are specified in this section.

6.1 User Data Identification

user_data_start_code-A 32-bit start code as defined in ISO/IEC 13818-2 to have the value B2 in hexadecimal**.**

user_data_type_code—An eight-bit code for picture user data, 03 in hexadecimal indicates the presence of picture user data of the type specified in this standard. Note from the syntax definition in subsection 5.2 that the compatible encoder shall send no more than one picture user data construct after any given picture header.

6.2 VBI Data Constructs

vbi_data_flag—Indicates that one or more VBI data constructs follow (closed captions, non-real-time video) in accordance with the semantics given in subsections 6.2.1 through 6.2.2
.

> **NOTE:** In order to comply with this standard, an encoder shall satisfy all the following general requirements with regard to VBI data:
>
> 1. The picture user data shall be packed in decode order, storing the VBI data to be reconstructed from a given picture in the picture user data of the same picture.
>
> 2. The VBI data for the repeated field shall be transported with the picture that transports the video data for the field to be repeated.
>
> 3. For a given picture and VBI data type, all the VBI data for the first display field shall be followed by all the VBI data for the second display field followed by all the VBI data for the third (repeated) display field, if present. Also, for a given picture, VBI data type, and field, all the VBI data for the first line shall be followed by all the VBI data for the second line, etc.[1]

6.2.1 Closed Captioning

cc_count—A five-bit integer (values in the range [0:31]) indicating the number of closed caption constructs following the field. All such constructs must occur in the intended display order, assuming an interlaced display.

cc_priority—A number between 0 and 3 indicating the priority of constructs in picture reconstruction where different levels of hardware capability exist. Priority value of 0 is highest and priority value of 3 is lowest. For closed caption constructs, up to four lines per display field (including Line 21) can be labeled as priority zero.

field_number—The number of the field, in display order, from which the VBI data originated, interpreted in Table 6-1. Please note that this standard uses the MPEG defined

---

[1] As an example, a three-field film-mode picture with a f2,f1,f2 display order and closed captions on lines 14, 16, and 21 of field 2 and lines 15 and 21 of field 1 shall be sent in order: d1-14, d1-16, d1-21, d2-15, d2-21, d3-14, d3-16, d3-21, where f1 is the odd field, f2 is the even field, d1 is the first display field, d2 is the second display field, and d3 is the third display field.

association of top field with field 1 or odd field and bottom field with field 2 or even field.

Table 6–1. Field Number for Picture User Data

Value	Meaning
00	Forbidden
01	1st display field
10	2nd display field
11	3rd display field (the repeated field in film mode).

line_offset—A five-bit integer giving the offset in lines from which the VBI data originated, relative to the base VBI frame line (line 10 of NTSC field 1, line 273 of NTSC field 2, line 6 of PAL field 1, and line 319 of PAL field 2). Consult Reference [4] for more information.

cc_data_1[1:8]— Data for the 1st closed caption character for this field such that the first transmitted bit is the first bit on the video line as displayed from left to right.

cc_data_2[1:8]—Data for the 2nd closed caption character for this field such that the first transmitted bit is the first bit on the video line as displayed from left to right.
The encoder shall compress closed caption data to the 16-bit representation and shall pack these bits into picture data starting with the least significant bit of the 1st character and ending with the most significant bit of the 2nd character.

6.2.2 Non-Real-Time Sampled Video

non_real_time_video_count—Indicates the number of non-real-time video constructs that follow. This field can have values of 0 through 15. All such constructs must occur in the intended display order, assuming an interlaced display.

non_real_time_video_priority—A number between 0 and 3 used by the decoder to determine if it is required to reconstruct the particular non-real-time VBI line. For non-real-time sampled video, only a single line may be labeled as priority zero. Thus the decoder that can reconstruct one line need only reconstruct priority 0.

sequence_number—Numbers each sequence of non-real-time video segments, starting from 1 and counting to 3, before rolling over to 1 again. A sequence_number of 0 indicates the non-real-time-sampled video line is not to be reconstructed (is inactive) until a segment is received with a non-zero sequence_number and therefore the corresponding fields do not follow for this construct.

The sequence number shall be incremented by one between sequences.

non_real_time_video_field_number—A one-bit number indicating the field into which the decoder must reconstruct the non-real video line as interpreted in Table 6–2.

Table 6–2. Non Real Time Video Field Number

Value	Meaning
0	Odd field
1	Even field

segment_number—The number of the non-real-time sampled video segment starting with 0001.

> The encoder shall segment non-real-time sampled video into 64-byte segments and transport each as an array of 32 luminance (Y) samples followed by an array of 16 chrominance sample pairs (Cb, Cr), starting with the most significant bit of the leftmost sample (Reference [2]). All segments of the sequence shall be transmitted in order before any segment of a new sample of the same non-real-time video line.

non_real_time_video_y_data[7:0]—The non-real-time sampled video luminance data for
> this segment such that the first bit is the most significant bit of the code word.

non_real_time_video_cb_data[7:0]—The non-real-time sampled video chrominance
> Cb data for this segment such that the first bit is the most significant bit of the code word.

non_real_time_video_cr_data[7:0]—The non-real-time sampled video chrominance
> Cr data for this segment such that the first bit is the most significant bit of the code word.