Society of Cable Telecommunications Engineers

ENGINEERING COMMITTEE
Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 21 2017

STANDARD FOR CARRIAGE OF VBI DATA IN CABLE DIGITAL TRANSPORT STREAMS

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Operational Practices (hereafter called "documents") are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability, best practices 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.

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

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, 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 document 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.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2017
140 Philips Road
Exton, PA 19341

Contents

List of Figures

List of Tables

1. Scope

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.* The approach builds upon a data structure defined in ATSC A/53 Part 4 (*Digital Television Standard: Part 4 – MPEG-2 Video System Characteristics)*, and is designed to be backwards-compatible with that method.

1.1 Document Structure

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

- **Section 1—**Provides the document Scope and structure
- **Section 2—**Lists Normative and Informative References
- **Section 3—**Defines the Compliance Notation for the document
- **Section 4—**Defines the acronyms and abbreviations used in this specification
- **Section 5—**Provides an overview of the VBI services supported
- **Section 6—**Specifies the video bitstream syntax and semantics for picture user data extensions

2. References

2.1 Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of subcommittee approval, the editions indicated were valid. All standards are subject to revision, and parties to agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

1. ISO/IEC 13818-2: 2000, *Generic Coding of Moving Pictures and Associated Audio,* International Standard (MPEG-2 Video).
2. ISO/IEC 13818-1:2007, *Generic Coding of Moving Pictures and Associated Audio*, , International Standard, (MPEG-2 Systems).
3. ITU-R BT. 1700 , Characteristics of composite video signals for conventional analogue televisions systems, 2005,

4. ITU-R BT.601-5, *Encoding Parameters of Digital Television for for Standard 4:3* and *Wide-Screen 16:9 Aspect Ratios*, 1995.
5. ATSC A/53 Part 4, *Digital Television Standard: Part 4 – MPEG-2 Video System Characteristics* (2009).
6. CEA-708-D, *Digital Television (DTV) Closed Captioning*, August 2008.
7. ANSI/CEA-608-E, *Line 21 Data Services*, April 2008.

2.2 Informative References

The following documents may provide valuable information to the reader but are not required when complying with this standard.

8. ANSI/SCTE 127, Carriage of Vertical Blanking Interval (VBI) Data in North American Digital Television Bitstreams, 2007.
9. EIA-516, *Joint EIA/CVCC Recommended Practice for Teletext: North American Basic Teletext Specification (NABTS)*, May 1988.
10. ITU-R BT.653-3, *Teletext Systems* (*World System Teletext)*, International Telecommunications Union, 1998.
11. SMPTE ST 12M-1, 2008 Section 10, VITC.
12. ANSI/SCTE 43, Digital Video Systems Characteristics standard for Cable Television, 2004.
13. SCTE 20 2012 Methods for carriage of closed captions and non-real time sampled video.

2.3 Published Materials

14. Code of Federal Regulations, Title 47 (Telecommunication), Part 73, Section 699.
15. NTC Report No. 7, Video Facility Testing Technical Performance Objectives, 1976.
16. *AMOL Signal Specification*, Nielsen Engineering and Technology, Document Number ACN 403-1122-000, Revision Level 1.4, January 16, 1995.
17. *AMOL II Signal Specification*, Nielsen Engineering and Technology, Document Number ACN 403-1193-024, Revision Level 3.2, May 19, 1995.
18. CEA-2020, Other VBI waveforms, *2006.*

3. Compliance Notation

"SHALL"	This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.
"SHALL NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but

	the full implications should be understood and the case carefully weighted before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

4. Acronyms and Abbreviations

ATSC	Advanced Television Systems Committee
AMOL	Automated Measurement of Lineups
bslbf	bit string left bit first
CCIR	International Radio Consultative Committee
CEA	Consumer Electronics Association
EIA	Electronic Industries Association
FCC	Federal Communications Commission
IEC	International Electrotechnical Commission
ISO	International Standards Organization
LL	Low Level
lsb	least significant bit
ML	Main Level
MP	Main Profile
MPEG	Moving Picture Experts Group
msb	most significant bit
NTC	Network Transmission Committee
NTSC	National Television System Committee
PAL	Phase Alternate Line
PAM	Pulse Amplitude Modulation
SID	Source Identification
uimsbf	unsigned integer most significant bit first
VBI	Vertical Blanking Interval
VITC	Vertical Interval Time Code
VITS	Vertical Interval Test Signal

5. 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 a number of NTSC VBI services beyond those supported by CEA-708 and ATSC A/53 Part 4. While ATSC A/53 Part 4 only standardizes carriage of CEA-708 closed captions (which includes carrying CEA-608 closed caption data from line 21), the VBI enhancements defined in this standard extend support to include:

a) CEA-608-compliant closed captioning for one or more VBI lines other than line 21 (Ref. [7] and Ref. [18])

b) Nielsen Source Identification (SID)/Automated Measurement Of Lineups (AMOL) signals (Ref. [16], Ref. [18] and Ref. [17])

c) North American Basic Teletext per the EIA-516 NABTS Specification (Ref. [9])

d) World System Teletext (WST) (Ref. [10])

e) Vertical Interval Time Code (VITC) (Ref. [11])

Note: a) and b) have been updated by CEA-2020 (Ref.[18]).

5.1 Use of Picture User Data

The use of additional new data types within video user data is discouraged unless the data is inseparable from the video content and unlikely to be replaced or modified prior to decoder's passive receipt of this data. In other words, it shall not be necessary to process the data before reception in the decoder and no processing should be required after reception other than using it for NTSC waveform reconstruction in the decoder.

The NTSC VBI data defined in this standard is an example of data that is by not any practical means separable from video and is unlikely to be replaced or modified. This type of VBI data is considered a part of the input video waveform that is digitized, compressed, transported, and reconstructed with an NTSC output (when the video mode is NTSC format).

The VBI data encoded in video user data syntax is carried only to allow reconstruction at a receiver of the full original NTSC waveform as it was input to the MPEG-2 encoder. In all other cases, applications residing in digital receivers should receive their pertinent data via elementary stream components of a digital service (i.e. via separate PID streams) and must not rely on the user data structure.

5.2 Closed Captioning

Digital television services in North America carry closed caption information as DTV Closed Captioning (DTVCC) standardized in the CEA-708 standard (Ref. [6]). CEA-708 provides carriage of CEA-608 caption data as a part of the DTVCC data structure. Placement of this structure within the video user data is standardized by ATSC A/53 Part 4 (Ref. [5]).

Many television services carry closed caption information in line 21, field 1 of the VBI. According to the CEA-608 standard, closed caption data is also carried in line 21 of field 2. As noted, these captions are also carried by CEA-708.

Note 1: ANSI/SCTE 43 (reference [12]) specifies the ordering of closed caption structures that conform to this specification and SCTE 20, 2012 (reference [13]).

Note 2: SCTE 20, 2012 (reference [13]) specifies another format for coding and transmission of closed captions that may be used in addition to what is specified here. Certain system service providers use the CEA-608 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 the standard line 21 closed caption usage.

5.3 Nielsen SID/AMOL Signals

AMOL encoding places information in the VBI and includes a SID code and a date/time stamp. It is used by broadcasters to verify that programs or commercials air at the intended times. VBI user data syntactic constructs support one or more VBI lines for this purpose.

5.4 Other VBI Standards

The encoding method described in this standard is applicable to VBI standards in addition to those mentioned here because it is a general purpose method for representing a basic VBI waveform. Most standards in current use use two-level luminance encoding, however this standard accommodates multi-level pulse-amplitude modulation (PAM) coding as well. The technique is applicable to both PAL and NTSC. If the MPEG-2 video syntax carried a video program in PAL format, the syntax described here can be used as-is to carry VBI data and reconstruct a PAL standard video waveform.

Note: ANSI/SCTE 127 (reference [8]) specifies a different format for coding and transmission of VBI information than what is specified here (luma-pam data).

6. 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 Video Standard, *ISO/IEC 13818-2*. The syntactic extensions to MPEG-2 Main Profile at Main Level operation for VBI services are implemented using the picture user data syntax defined in subsection 6.2.

6.1 Syntax Conventions and Definitions

6.1.1 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 6–1, 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 6–1. Bitstream Data Elements and Conditions

while (condition) {	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.
data_element	
. . .	
}	
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) {	If the condition is true, then the first group of data elements occurs next in the data stream.
data_element	
. . .	
} else {	If the condition is not true, then the second group of data elements occurs next in the data stream.
data_element	
. . .	
}	
for (i = 0; i < n; i++) {	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.
data_element	
. . .	
}	
/* 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][n]	the m+1, n+1th element of a two-dimensional array of data

While the syntax descriptions given in this document are expressed in procedural terms, it should not be assumed that subsection 6.2 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.

6.1.2 Reserved, Forbidden & Marker Bits

The terms *reserved* and *forbidden* are used in the description of some values of several fields in the coded bitstream. These are reproduced from Ref [1] section 5.3 for convenience here.

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—Indicates a one-bit field in which the value zero is forbidden (and it therefore shall have the value '1'). These marker bits are introduced at several points in the syntax to avoid start code emulation.

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.

6.1.3 Mnemonics

The following mnemonics are defined to describe the different data types used in the user data syntax described in subsection 6.2:

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.

6.1.4 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.

6.1.5 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 6-1.

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

Figure 6-1. Next Start Code Function Syntax

6.2 Bitstream Ordering of Picture User Data

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]
4. Note from the syntax definition in subsection 6.3 that a compliant bitstream will only contain one picture user data construct with the user_data_type_code value of 0x03 in any given picture header. Other user data constructs with other user_data_type_code values might be found, however.

6.3 Picture User Data Syntactic Extensions

The picture user data syntax to support VBI services is shown in Figure 6-2, where non-shaded cells highlight syntactic extensions to the ATSC/CEA syntax described in A/53 Part 4 Ref. [5] and CEA-708 Ref. [6]. Support for reconstruction of CEA-608 captions for insertion into the NTSC VBI by receivers that provide this function using SCTE 20 (Ref.[13]), requires dual carriage of CEA-608 captions using the ATSC/CEA syntax described in A/53 Part 4 Ref.[5] and CEA-708 Ref. [6] (user_data_type_code value of 0x03 and cc_type values of ‘00’ or ‘01’) and SCTE 20.

In Figure 6-2 (below) the ATSC/CEA syntax is reproduced as a service to the reader (and shown with shading). The definitive definition exists in ATSC A/53 Part 4 (Ref. [5]) and CEA-708 (Ref. [6]). As shown, the ATSC syntax is extensible via the definition of additional user_data_type_code values. This standard defines type code value 0x04 as additional_EIA_608_data() (see Figure 6-3) and type code value 0x05 as luma_PAM_data() (see Figure 6-4). The following sections define these data structures.

Note: Since the transfer of standardization of closed captions from EIA to CEA resulted in different document names, this document refers to both CEA-608 and CEA-708 by those terms. In the bitstream syntax below, however, the variable names will continue to use “EIA” for historical continuity.

---

[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 are 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.

	No. of bits	Mnemonic
user_data( ) {		
user_data_start_code	32	Bslbf
ATSC_identifier	32	Bslbf
user_data_type_code	8	Uimsbf
if (user_data_type_code == ‘0x03’) {		
reserved	1	’1’
process_cc_data_flag	1	Bslbf
zero_bit	1	’0’
cc_count	5	Uimsbf
reserved	8	‘1111 1111’
for (i=0 ; i < cc_count ; i++) {		
one_bits	5	‘1 1111’
cc_valid	1	Bslbf
cc_type	2	Bslbf
cc_data_1	8	Bslbf
cc_data_2	8	Bslbf
}		
marker_bits	8	‘1111 1111’
}		
} else if (user_data_type_code == ‘0x04’)		
additional_EIA_608_data()		
else if (user_data_type_code == ‘0x05’)		
luma_PAM_data()		
else		
ATSC_reserved_user_data		
next_start_code()		
}		

Figure 6-2. Extensions to ATSC Picture User Data Syntax

6.4 Additional CEA-608 Data

Figure 6-3 defines the additional_EIA_608_data() data structure.

additional_CEA_608_data() {		
marker_bits	3	‘111’
additional_cc_count	5	Uimsbf
for (i=0 ; i < additional_cc_count ; i++) {		
additional_cc_valid	1	Bslbf
additional_cc_line_offset	5	uimsbf
additional_cc_field_number	2	Uimsbf
additional_cc_data_1	8	Bslbf
additional_cc_data_2	8	Bslbf
}		
while(nextbits() != ‘0000 0000 0000 0000 0000 0001’)		
additional_type_4_data	8	
}		

Figure 6-3. Additional CEA 608 Data Structure and Syntax

additional_cc_count—A five-bit integer (values in the range [1:31]) indicating the number of lines of CEA-608 to be defined following the field. All such constructs must occur in the intended display order, assuming an interlaced display line and field display order.

additional_cc_valid—A Boolean flag that indicates, when set, that the two bytes of additional closed caption data to follow are valid. When the flag is false, the data shall be ignored by the decoder. The additional_cc_valid flag may be set to zero to create a place-holder for closed caption data to be inserted into the digital multiplex downstream from the encoder.

additional_cc_line_offset—A five-bit integer (values in the range [1:31]) giving the offset in lines from which the CEA-608 closed caption data originated relative to the base VBI frame line (line 9 of 525-line [NTSC and PAL/M] field 1, line 272 of 525-line field 2, line 5 of 625-line [all PAL except PAL/M] field 1, and line 318 of 625-line field 2), as specified in ITU-R BT. 1700 (Ref. [3]).

additional_cc_field_number—The number of the field, in display order, from which the CEA-608 data originated, interpreted in the following Table.

Table 6-2. Field Number for Additional CEA-608 Data

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

additional_cc_data_1—The first 8-bit byte of CEA-608 closed caption data to be reconstructed on the line identified by additional_cc_line_offset within the field identified by additional_cc_field_number.

additional_cc_data_2—The second 8-bit byte of CEA-608 closed caption data to be reconstructed.

additional_type_4_data—Reserved for future use by SCTE.

6.5 Luminance PAM Data Structure

Figure 6-4 defines the luma_PAM_data() structure.

luma_PAM_data() {		
marker_bits	3	'111'
luma_PAM_count	5	Uimsbf
for (i=0 ; i<luma_PAM_count ; i++) {		
luma_PAM_priority	2	Uimsbf
field_number	2	Uimsbf
start_sample	9	Uimsbf
bits_per_symbol	3	uimsbf
PAM_increment	6	Uimsbf
PAM_modulus	10	Uimsbf
low_amplitude_level	8	Uimsbf
high_amplitude_level	8	Uimsbf
line_offset	5	Uimsbf
pulse_shape	3	Uimsbf
if (pulse_shape == "rectangular") {		
symbol_to_transition_ratio	8	Uimsbf
}		
if (pulse_shape == "raised_cosine") {		
Reserved	3	Bslbf
PAM_alpha	5	Uimsbf
}		
if (pulse_shape == "PRC") {		
Reserved	8	Bslbf
}		
if (pulse_shape == "reserved") {		
Reserved	8	Bslbf
}		
marker_bit	3	'111'
word_count	5	Uimsbf
for (j=0 ; j<word_count ; j++) {		
marker_bit	2	'11'
luma_PAM_word	22	Bslbf
}		
marker_bit	1	'1'
remainder_count	5	Uimsbf

for (j=0 ; j< remainder_count ; j++) {		
symbol_bit	1	Bslbf
}		
while (!bytealigned())		
marker_bit	var	Bslbf
}		
while(nextbits() != ‘0000 0000 0000 0000 0000 0001’) {		
additional_type_5_data	8	
}		

Figure 6-4. Luminance PAM Data Structure and Syntax.

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

luma_PAM_priority—A number between 0 and 3 indicating the priority of constructs in picture reconstruction where different levels of hardware capability exist.

field_number—The number of the field, in display order, from which the VBI data originated, interpreted in Table 6-3.

Table 6-3. Field Number for Picture User Data

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

start_sample—A 9-bit unsigned integer (values in the range [0:511]) which indicates the sample of the reconstructed luminance line at which the transition into the first Luminance PAM symbol shall start. start_sample shall be in the same units as ITU-R BT.601-5 (Ref. [4]) samples and shall be relative to the first sample of ITU-R BT.601-5 reconstructed frames.

bits_per_symbol—A 3-bit enumerated type that specifies the number of bits per symbol according to the following table.

Table 6-4. Bits Per Symbol Encoding

bits_per_symbol	Meaning
000	Forbidden
001	One bit per symbol: two-level PAM encoding
010	Two bits per symbol: four-level PAM coding
011	Three bits per symbol: eight-level PAM coding
100	Four bits per symbol: sixteen-level PAM coding
101-111	reserved for future use

PAM_increment—A 6-bit unsigned integer (values in the range [1:63]) which indicates the Luminance PAM symbol clock increment value and takes on values that describe, together with PAM_modulus, the relationship of the Luminance PAM symbol clock to a 27 MHz reference. See the semantics of PAM_modulus for more details.

PAM_modulus—A 10-bit unsigned integer (values in the range [2:1023]) which indicates the Luminance PAM symbol clock modulus value and takes on values that describe, together with PAM_increment, the relationship of the Luminance PAM symbol clock to a 27 MHz reference. Specifically, PAM_increment and PAM_modulus are related to the Luminance PAM symbol rate as:

PAM_increment / PAM_modulus = Luminance PAM symbol rate / system_clock_frequency[2]

where

system_clock_frequency is specified in ISO/IEC 13818-1 as 27 MHz ± 30 ppm.

low_amplitude_level—An 8-bit unsigned integer (values in the range [1:254] which indicates the amplitude at which symbols of the lowest amplitude value shall be reconstructed in units of amplitude of ITU-R BT.601-5 (Ref. [4]) reconstructed frames.

high_amplitude_level— An 8-bit unsigned integer (values in the range [1:254] which indicates the amplitude at which symbols of the highest amplitude value shall be reconstructed in units of amplitude of ITU-R BT.601-5 (Ref. [4]) reconstructed frames.

line_offset—A five-bit integer (values in the range [1:31]) giving the offset in lines from which the Luminance PAM data originated relative to the base VBI frame line (line 9 of 525-line [NTSC and PAL/M] field 1, line 272 of 525-line field 2, line 5 of 625-line [all PAL except PAL/M] field 1, and line 318 of 625-line field 2), as specified in ITU-R BT. 1700 (Ref. [3]).

pulse_shape—A 3-bit unsigned integer which indicates the shape of the pulses which shall be used to reconstruct this line of Luminance PAM. The meaning of pulse_shape is given in Table 6-5.

---

[2] The value of nrz_increment must not exceed nrz_modulus-1.

Table 6-5. Pulse Shape.

pulse_shape	Meaning
000	**Rectangular**
001	**raised_cosine**
010	**PRC**—partial response coding
011-111	Reserved

The pulse shape of a PRC signaling system as described by:

$$H(\omega) = 2 * T * \cos(\omega T/2)$$

in the frequency domain , as represented in Figure 6-5 or

$$h(t) = \frac{4}{\pi} * \frac{\cos(\pi * t/T)}{(1 - 4 * t^2/T^2)}$$

in the time domain, as represented in Figure 6-6.

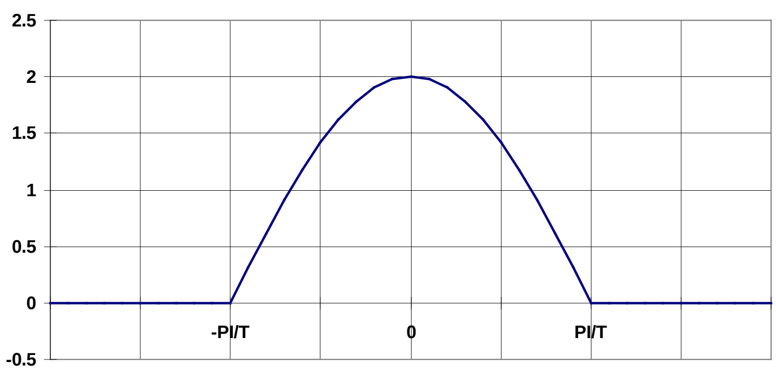

Figure 6-5 Frequency Response of PRC Filter (Linear Scale)

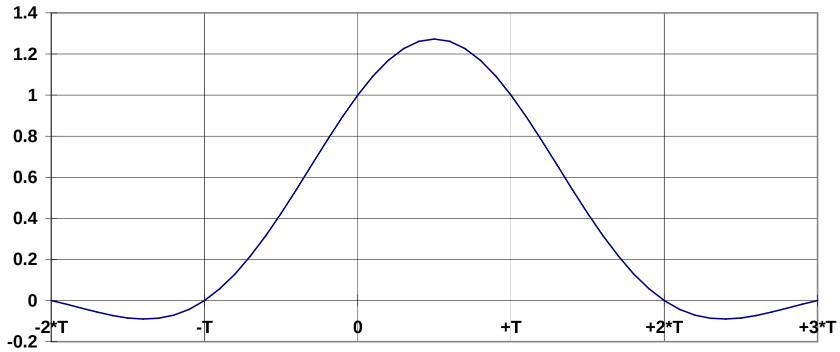

Figure 6-6 PRC Impulse Response

symbol_to_transition_ratio—An 8-bit unsigned integer (values in the range [16:255]) that indicates the ratio of each Luminance PAM symbol's duration to each symbol's transition duration between the amplitudes specified by low_amplitude_level and high_amplitude_level and having units of 2^{-4} (0.0625). This field describes symbols with a symbol to transition ratio ranging from 1.0 to 15.9375.

PAM_alpha—A 5-bit unsigned integer (values in the range [0:31]) which indicates the value of Alpha for the Raised Cosine filter whose pulse shape describes each Luminance PAM symbol with units of 2^{-5} (0.03125). This field describes values of Alpha from 0.03125 to 1.0. The meaning of PAM_alpha is defined in Table 6-6.

Table 6-6. PAM Alpha.

PAM_alpha	Alpha value
00000	1.0
00001-11111	PAM_alpha * 0.03125

word_count—A 5-bit unsigned integer (values in the range [0:31]) which indicates the number of marker_bit and luma_PAM_word pairs that follow this field.

The syntax defines the sequence of N-bit symbols representing PAM-encoded luminance values defining the VBI line. The value of N is given by the bits_per_symbol parameter. Each symbol is placed most-significant bit first. Symbols defined first represent samples to be reconstructed on the leftmost side of the video line as it is displayed from left to right. For reference, the sequence of N-bit symbols is called the *symbol bit list*.

To avoid start-code emulation, markers bit are inserted between each group of 22 bits taken from the symbol bit list.

luma_PAM_word—A string representing the next 22 bits of the symbol bit list.

remainder_count—A 5-bit unsigned integer (values in the range [0:21]) which indicates the number of symbol_bits that follow this field.

symbol_bit—A single bit of the symbol bit list.

Example: Given the following sequence of 3-bit symbols: 1, 1, 1, 7, 1, 1, 1, 1, 2, 3, 0, 4, 5, the word_count would be one, the luma_PAM_word would be (001 001 001 111 001 001 001 0), the remainder_count would be 17, and the remaining bits would be (01 010 011 000 100 101).

additional_type_5_data—Reserved for future use by SCTE.

6.6 Picture User Data Bandwidth

Decoders supporting this standard are expected to parse and reconstruct user_data() that has a data rate of not more than 38.4 Kbps over any one second, and has no more than 400 bytes of user_data() in any single picture.

Decoders supporting this standard are expected to parse user_data() that exceeds the above bandwidth limitation but does not exceed 800 Kbps over any one second and has no more than 8 Kbytes in any single picture in a manner such that proper operation of the decoder in other respects is not degraded. In this event, decoders are expected to continue to parse and reconstruct closed captioning appropriately, but may discard any additional portions of user_data(). Decoders need not function appropriately in the presence of user_data() data rates greater than 800 Kbps over one second or has more than 8 Kbytes of user_data() in any single picture.