

SMPTE STANDARD

VC-3 Picture Compression
and Data Stream Format

Page 1 of 108 pages

Table of Contents	Page
Foreword	4
Intellectual Property	4
Introduction.....	4
1 Scope	6
2 Conformance Notation	6
3 Normative References	6
4 Overview (Informative).....	7
5 Notation.....	8
5.1 Values	8
5.2 Arithmetic Operators	8
5.3 Logical Operators	9
5.4 Relational Operators	9
5.5 Bitwise Operators	9
5.6 Assignment	9
5.7 Precedence Order of Operators.....	9
5.8 Pseudocode Operations.....	10
5.9 Definition of Terminology	11
5.9.1 AC coefficient.....	11
5.9.2 alpha	11
5.9.3 amplitude	11
5.9.4 bitstream	11
5.9.5 bitrate.....	11
5.9.6 block	11
5.9.7 byte-aligned	11
5.9.8 byte	11
5.9.9 CBR	11
5.9.10 coding unit	11
5.9.11 component.....	11
5.9.12 compressed payload	11
5.9.13 compression	11
5.9.14 compression ID.....	11
5.9.15 DC coefficient	12
5.9.16 decoder.....	12
5.9.17 decoding process.....	12
5.9.18 dequantization	12
5.9.19 display process.....	12
5.9.20 encoder.....	12
5.9.21 encoding (process)	12

5.9.22	flag	12
5.9.23	Frame	12
5.9.24	field	12
5.9.25	frame rate	12
5.9.26	HD raster	12
5.9.27	header.....	12
5.9.28	macroblock	13
5.9.29	macroblock scan line	13
5.9.30	parameter	13
5.9.31	picture	13
5.9.32	progressive	13
5.9.33	quantize, quantization.....	13
5.9.34	raster.....	13
5.9.35	resolution independent (RI) raster	13
5.9.36	run.....	13
5.9.37	variable length coding (VLC)	13
5.9.38	VBR	13
5.9.39	Video Codec 3 (VC-3)	13
5.9.40	zigzag scanning order	13
5.10	Acronym Definitions	14
5.10.1	DCT	14
5.10.2	IDCT	14
5.10.3	LSB	14
5.10.4	MB	14
5.10.5	MSB	14
5.10.6	VLC	14
5.10.7	VLD	14
6	Video Sampling Structure	14
6.1	HD Raster Constraints	14
6.2	RI Raster Constraints	15
6.3	Frame Structure	15
7	Compressed Frame Format	27
7.1	Compressed Frame Size	28
7.2	Header Definition	30
7.2.1	Header Prefix.....	30
7.2.2	Coding Control A	31
7.2.3	Image Geometry	32
7.2.4	Compression ID.....	34
7.2.5	Coding Control B	35
7.2.6	Time Code Control	36
7.2.7	Time Code Data Area.....	36
7.2.8	User Data Control.....	37
7.2.9	User Data Payload	38
7.2.10	Macroblock Scan Indices Control	38
7.2.11	Macroblock Scan Indices Payload.....	38
7.3	Compressed Payload	39
7.3.1	Compressed Data	39
7.3.1.1	DCT-Compressed Macroblock Data	40
7.3.1.2	RLE-Compressed Alpha Macroblock Data	42
7.3.1.3	Macroblock Scan-Line Padding	44
7.3.2	Compressed Payload Padding.....	44
7.4	EOF Signature	45
7.4.1	Verification of Data Integrity (Informative).....	45
7.5	Illustrative Example (Informative)	46

8	Decoding	47
8.1	Macroblock Decoding	47
8.1.1	Video	47
8.1.2	Alpha	50
8.2	DCT block decoding	50
8.2.1	Entropy Decoding	50
8.2.2	Macroblock Header and Quantization Scale Factor	51
8.2.3	VLC Codeword Tables	51
8.2.4	DC Coefficient Decoding	51
8.2.5	AC Coefficient Entropy Decoding	53
8.2.5.1	AC Coefficient Codeword Tables	53
8.2.5.2	AC Coefficient Decoding Pseudo Code	53
8.2.5.3	AC Coefficient Decoding Pseudo Code Process (Informative)	53
8.2.6	Inverse Zig-zag	55
8.2.7	Inverse Quantization	55
8.2.8	Discrete Cosine Transform	56
8.2.8.1	8x8 Inverse DCT	56
8.2.8.2	8x8 Forward DCT (informative)	56
8.2.8.3	Video Sample Level Adjustment	57
8.3	RLE Block Decoding	57
8.4	Best Practices (Informative)	58
Annex A	AC Coefficient Entropy Codeword Format (Normative)	59
Annex B	User Data Control and Payload Field Usage (Informative)	62
Annex C	Compression IDs (Normative)	63
Annex D	Quantization Weighting Tables (Normative)	65
Annex E	VLC Tables (Normative)	76
Annex F	System Overview (Informative)	101
Annex G	Compressed Bitrates (Informative)	103
Annex H	VC-3 Profiles and Levels (Informative)	106
Annex I	Bibliography (Informative)	107
Annex J	Revision History	108

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 2019-1 was prepared by Technology Committee 10E.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

The original scope of the VC-3 standard covered HD resolutions 1920 x 1080 (4:2:2 and RGB, 8 and 10 bit), and 1280 x 720 (4:2:2, 8-bit) as well as 1440 x 1080 and 960 x 720 (4:2:2, 8 bit) encoded according to ITU-R 709 as a constant bitrate codec. This backwards compatible revision adds five new, spatially resolution-independent, progressive-only compression IDs which also support 8-, 10- or 12-bit component depth, 4:4:4 and 4:2:0 $YCbCr$ color subsampling, ITU Rec BT.2020 and provides for the optional inclusion of Alpha information (both loss-based and lossless for select compression IDs) and variable bitrate operation.

This revision of the standard also introduces the notion of a Profiles and Levels classification (Annex H). Encodings using compression IDs defined in prior revisions of this standard are collectively referred to as HD Profile. Encodings complying with this standard revision are referred to as RI (resolution independent) Profile.

The RI Profile is based on generalizations (removing or relaxing constraints in definitions) and extensions (backwards compatible additions) of some of the progressive HD scan types. Without applying the generalizations and extensions the bitstream formation of the RI profile matches the bitstream formation of the HD profile, except for header versions and compression IDs.

This revision of the VC-3 standard supports $YC_B C_R$ and RGB video source data as follows:

$YC_B C_R(A)$:

- Rasters:
 - Full raster (1920x1080, 1280x720)
 - Thin raster (1440x1080, 960x720), HD-only
 - RI (1x1 up to 16384x16384)
- Sampling:
 - HD
 - 4:2:2 at 8- and 10-bits
 - 4:4:4 at 10-bits
 - RI
 - 4:2:0(:4) at 8-, 10- and 12-bits
 - 4:2:2(:4) at 8-, 10- and 12-bits
 - 4:4:4(:4) at 10- and 12-bits
- Progressive
- Interlaced (HD-only)

RGB(A):

- Rasters:
 - Full raster (1920x1080)
 - RI (1x1 up to 16384x16384)
- Sampling:
 - HD: 10-bits
 - RI: 10- and 12-bits
- Progressive only

1 Scope

This standard specifies the compressed data format and decoding process for VC-3 compressed video data. VC-3 supports video up to 16384 x 16384 raster resolutions as $YC_B C_R(A)$ (4:4:4(:4), 4:2:2(:4), 4:2:0(:4)) at 8-, 10- and 12-bit or as RGB(A) at 10- or 12-bit sampling depth.

VC-3 is an intra-frame compression format, independent of any predefined frame rate, which supports both constant (CBR) and variable (VBR) bitrate operations. The compression parameters are described by the contents of a header data region and attributes associated with a Compression ID described within this standard. The standard does not specify the encoding process nor does it define video frame rates although examples are used in this document.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 274:2008) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 274M-2008). Documents with the same root number (e.g. 274) and publication year (e.g. 2008) are functionally identical.

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

SMPTE ST 12-1:2014, Time and Control Code

SMPTE ST 274:2008, Television – 1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates

SMPTE ST 296:2012, 1280 x 720 Progressive Image 4:2:2 and 4:4:4 Sample Structure – Analog and Digital Representation and Analog Interface

SMPTE ST 2036-1:2014, Ultra High Definition Television – Image Parameter Values for Program Production

Recommendation ITU-R BT.709-5 (04/02), Parameter Values for the HDTV Standards for Production and International Programme Exchange.

Recommendation ITU-R BT.2020-1 (06/2014), Parameter Values for Ultra-High Definition Television Systems for Production and International Program Exchange

4 Overview (Informative)

This section provides a brief overview of the VC-3 decoding process. Annex F System Overview is an informative annex where one can find a description of the system and encoding process for VC-3.

The decoding process is depicted in Figure 1. The VC-3 bitstream contains a header region followed by entropy codes, which encode the quantized Discrete Cosine Transform (DCT) coefficients for video and optional DCT-encoded or RLE-encoded alpha blocks. As shown in the figure, the header region of the VC-3 compressed bitstream is parsed to determine information about the compressed bitstream.

For Video data, entropy decoding is then performed, followed by inverse quantization. The inverse quantized coefficients are then transformed to the spatial domain via the Inverse Discrete Transform (IDCT), producing an initial output picture.

For DCT-based Alpha the decoding process follows the Video path.

For lossless Alpha data, the bitstream is decoded using a predictive, differential run-length encoding scheme, followed by zigzag reordering stage.

The VC-3 bitstream is intra-frame encoded, meaning that each compressed frame is independent of all other frames in the encoded bitstream.

The remainder of this standard describes the video sampling structure (Section 6), the compressed frame format (Section 7), and the decoding process (Section 8). In addition there are annexes that provide details regarding the entropy codeword format (Annex A), a suggested usage of the User Data Payload area (Annex B), the Compression ID definition tables (Annex C), the quantization tables (Annex D), the Variable Length Codeword (VLC) tables (Annex E), and compressed frame bitrates (Annex G).

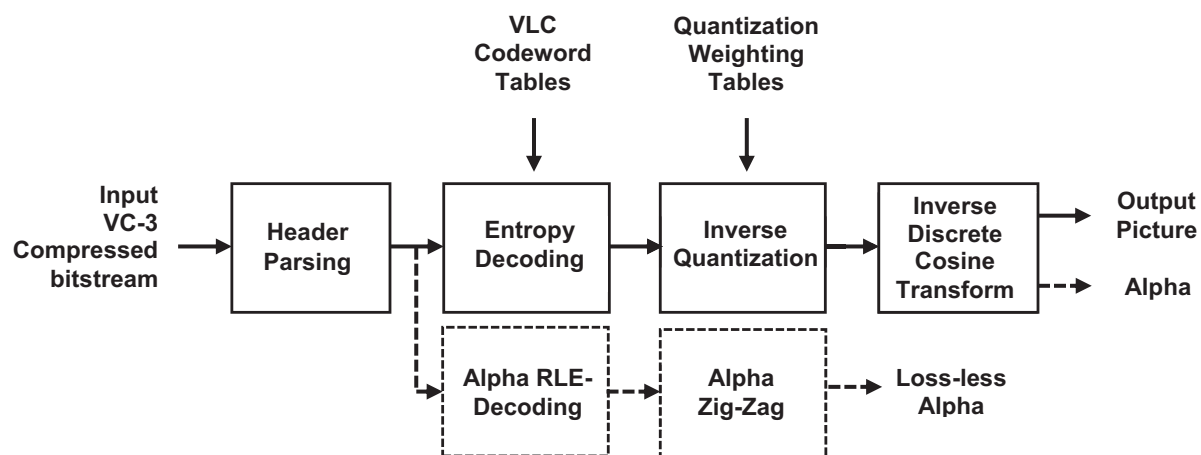


Figure 1 – Overview of the decompression process (dashed elements are optional)

5 Notation

The following notation is used in this document.

5.1 Values

d	Decimal value d
xb	Binary value x
$0xH$	Hexadecimal value H

5.2 Arithmetic Operators

$+$	Addition
$-$	Subtraction (as a binary operator) or negation (as a unary operator)
$++$	Increment
$*$ or $\bullet$	Multiplication
$-$	Division
$ x $	Absolute value:
	$ x = x$, when $x > 0$
	$ x = 0$, when $x = 0$
	$ x = -x$, when $x < 0$
$sgn(x)$	Sign:
	$sgn(x) = 1$, when $x \geq 0$
	$sgn(x) = -1$, when $x < 0$

$\lfloor x \rfloor$	Floor: Returns the largest integral value $\leq x$
$\lfloor x \rfloor_i$	Round to multiple: Returns the multiple of i closest to x

5.3 Logical Operators

TRUE/FALSE convention: This document uses the convention that a variable or expression evaluating to a nonzero value is equivalent to a condition that is TRUE and a variable or expression evaluating to a zero value is equivalent to a condition that is FALSE.

$\&\&$	logical AND
$\ \ $	logical OR

5.4 Relational Operators

$>$	Greater than
$\geq$	Greater than or equal to
$<$	Less than
$\leq$	Less than or equal to
$=$	Equal to
$\neq$	Not equal to

5.5 Bitwise Operators

A twos complement number representation is assumed where the bitwise operators are used.

$\gg$	Shift right with sign extension.
$\ll$	Shift left with zero fill.
$\&$	Bitwise AND.

5.6 Assignment

$=$	Assignment operator.
-----	----------------------

5.7 Precedence Order of Operators

The precedence order of operators is defined as follows:

Operators	Type of operation	Associativity
$()$	Expression	Left to Right
$++$	Postfix operator	Right to Left
$-$	Unary	Left to Right

$*, /$	Multiplicative	Left to Right
$+, -$	Additive, Subtractive	Left to Right
$<< >>$	Shift	Left to Right
$>, \geq, <, \leq$	Relational	Left to Right
$=$	Equality	Left to Right
$\&\&, $	AND, OR	Left to Right
$=$	Assignment operator	Right to Left

Operators are listed in descending order of precedence. If several operators appear in the same line, they have equal precedence. When several operators of equal precedence appear at the same level in an expression, evaluation proceeds according to the associativity of the operator either from right to left or from left to right.

5.8 Pseudocode Operations

The following operations are used in the pseudo code to define the decoding process.

- `//` is a comment to the line end

- A group of statements is treated functionally as a single (compound) statement if and only if it is enclosed in curly brackets `{ ... }`.

- `while (condition)`
 `(statement)`

specifies repeated execution of *statement* until condition is no longer true.

- `for(initial statement; condition; subsequent statement)`
 `(primary statement)`

specifies evaluation of *initial statement* followed by evaluation of *condition*, and if *condition* is true, specifies repeated execution of *primary statement* followed by *subsequent statement* until *condition* is no longer true

- `if (condition)`
 `(statement)`
 else
 `(alternate statement)`

statement is executed if *condition* is true, *alternate statement* is executed otherwise.

- `switch (discriminator)`
 {
 case `(value1)` [, `(value2)`, `(value3)`, `(value4)`]: `(statement1)`; break;
 case `(value5)` [, `(value6)`, `(value7)`, `(value8)`]: `(statement2)`; break;
 }

`(statement1)` is executed if `(discriminator)` matches any of the `(values)` following the case label.

5.9 Definition of Terminology

For the purposes of this standard, the following definitions apply:

5.9.1 AC coefficient

Any DCT coefficient for which the frequency in one or both dimensions is non-zero.

5.9.2 alpha

The opacity of a pixel. A logical value of 0.0 denotes complete transparency, while a logic value of 1.0 denotes complete opacity.

5.9.3 amplitude

The absolute value of a DCT coefficient.

5.9.4 bitstream

An ordered series of bits that forms the coded representation of the data.

5.9.5 bitrate

The (bit) data rate at which the decoder consumes the bits of the input bitstream.

5.9.6 block

An 8 x 8 matrix of samples, or 64 equivalent transform coefficients.

5.9.7 byte-aligned

A bit in a coded bitstream is byte-aligned if its position is a multiple of 8 bits from the first bit in the stream.

5.9.8 byte

A sequence of 8 bits.

5.9.9 CBR

Constant bitrate encoding. The output data per time segment is constant.

5.9.10 coding unit

A self-contained VC-3 data segment. A coding unit is comprised of a Header, a Compressed Payload, and an End of Frame signature data section. Progressive frames require a single coding unit, while interlaced frames require two coding units, one for each field.

5.9.11 component

A matrix, block or single sample from one of the three matrices (one luma matrix and two color difference matrices or an RGB triplet) that make up a picture.

5.9.12 compressed payload

The section of data which contains the compressed data of a coding unit.

5.9.13 compression

Reduction in the number of bits used to represent an item of data.

5.9.14 compression ID

A unique identifier associated with a sample bit depth and encoding parameters such as variable length codeword tables and quantization weights. For HD bitstreams the Compression ID also implies the raster size and compressed frame size. This implication does not apply to RI-based bitstreams.

5.9.15 DC coefficient

The DCT coefficient for which the frequency is zero in both dimensions.

5.9.16 decoder

An embodiment of a decoding process.

5.9.17 decoding process

The process defined whereby a compressed bitstream is converted to a picture. The decoding process does not include the display rendering process, which can convert these samples to images of another colorspace, can apply format specific black and white levels, color primaries, matrix coefficients sample aspect ratios, etc., and can display the images with frequency and timing different from the sampled rate.

5.9.18 dequantization

The process of rescaling the quantized transform coefficients after their representation in the bitstream has been decoded and before they are presented to the inverse transform.

5.9.19 display process

The (non-normative) process by which reconstructed frames are displayed.

5.9.20 encoder

An embodiment of an encoding process.

5.9.21 encoding (process)

A process which reads a stream of input pictures and produces a valid coded bitstream.

5.9.22 flag

A variable which can take one of only two values is called a flag.

5.9.23 Frame

A frame contains lines of spatial information of a video signal. For progressive video, these lines contain samples starting from one time instant and continuing through successive lines to the bottom of the frame. For interlaced video, a frame consists of two fields.

5.9.24 field

For an interlaced video signal a "field" refers to the assembly of alternate lines. Therefore an interlaced frame is composed of two fields, a top field, containing the spatially top-most line of the image, and a bottom field, containing the spatially bottom-most line of the image. One of these fields will commence one field period later than the other.

5.9.25 frame rate

The rate at which frames are output from the decoding process.

5.9.26 HD raster

Rasters conforming to either 1920x1080p, 1920x1080i, 1280x720p (full raster), or 1440x1080p, 1440x1080i, 960x720p (thin raster).

5.9.27 header

A block of data in the coded bitstream containing the coded representation of a number of data elements pertaining to the coded data that follow the header in the bitstream.

5.9.28 macroblock

For 4:2:0 video data, the four 8 by 8 blocks of luma data and the two corresponding 8 by 8 blocks of color-difference data corresponding to a 16 by 16 section of the luma component of the picture, resulting in a total of 6 DCT blocks. For 4:2:2 video data, the four 8 by 8 blocks of luma data and the four corresponding 8 by 8 blocks of color-difference data corresponding to a 16 by 16 section of the luma component of the picture, resulting in a total of 8 DCT blocks. For (RGB) 4:4:4 data, the macroblock consists of a 16 by 16 section of sample data, resulting in a total of 12 DCT blocks.

5.9.29 macroblock scan line

A row of macroblocks running the entire width of the video raster.

5.9.30 parameter

A variable within the syntax which can take one of a range of values.

5.9.31 picture

Source, or decoded image data. A source or decoded picture consists of three rectangular matrices of 8-, 10- or 12-bit numbers representing the luma and two color-difference signals or representing the red, green and blue signals. See the relevant normative reference for the specific definition.

5.9.32 progressive

The property of frames where all the samples of the frame represent the same instance in time and lines are stored in consecutive order.

5.9.33 quantize, quantization

A process in which the continuous range of values of an input signal is divided into non-overlapping (but not necessarily equal) sub-ranges, and a discrete, unique value is assigned to each sub-range. A unique index is generated to represent this value.

5.9.34 raster

A picture with a particular width, height and frame layout.

5.9.35 resolution independent (RI) raster

A raster whose size is not predefined and shall be additionally specified. This term is used to differentiate from the pre-defined HD rasters.

5.9.36 run

The number of zero coefficients preceding a non-zero coefficient, in the scan order.

5.9.37 variable length coding (VLC)

A reversible procedure for coding, that assigns shorter code-words to symbols of higher probability and longer code-words to symbols of lower probability. The acronym VLC also indicates a variable length code.

5.9.38 VBR

Variable bitrate encoding. The amount of output data per time segment can vary.

5.9.39 Video Codec 3 (VC-3)

The name of the standard described here.

5.9.40 zigzag scanning order

A specific sequential ordering of the transform coefficients from (approximately) the lowest spatial frequency to the highest.

5.10 Acronym Definitions

The following acronyms are commonly used:

5.10.1 DCT

Discrete Cosine Transform

5.10.2 IDCT

Inverse Discrete Cosine Transform

5.10.3 LSB

Least Significant Bit.

5.10.4 MB

MacroBlock.

5.10.5 MSB

Most Significant Bit.

5.10.6 VLC

Variable Length Code.

5.10.7 VLD

Variable Length Decoding.

6 Video Sampling Structure

The VC-3 compression standard supports the following features:

6.1 HD Raster Constraints

The constraints in this section match the definitions of the previous revision (SMPTE ST 2019-1:2014) of this standard.

- 1080 line interlaced video
- 1080 line progressive video
- 720 line progressive video

For 1080 line Video raster sizes, the video sampling structure of $YC_B C_R$ and RGB video shall be defined by SMPTE ST 274; for 720 line raster sizes, the video sampling structure of $YC_B C_R$ video shall be defined by SMPTE ST 296.

For $YC_B C_R$ video, the decoder's video output signal shall consist of Luminance (Y) and color-difference components (C_B and C_R) with a 4:2:2 (8-, 10-bit) or 4:4:4 (10-bit) sampling ratio. The component sample levels are tabulated in Table 1.

For RGB video, the decoder's output signal shall consist of 10-bit RGB components. The video sampling structure for RGB video shall always be 1920 samples per line and 1080 lines per frame (no thin raster support). The frame layout for RGB video shall be progressive. The component sample levels are tabulated in Table 2.

The decoder's video output signal for RGB video shall consist of Red, Green and Blue components. The RGB *encoder* may choose, on a macroblock basis, between encoding RGB or internally transcoded $Y_{CB}C_R$ values in the bitstream. Only the RGB format shall be returned by the *decoder*.

For HD rasters only CBR operation shall be permitted.

6.2 RI Raster Constraints

The constraints in this section apply to the additions made in this revision of this standard.

- Samples per line and lines per frame of the raster shall be no less than 1 and no more than 16384.
- For $Y_{CB}C_R$ 4:2:2 video the number of samples per line shall be even.
- For $Y_{CB}C_R$ 4:2:0 video the number of lines and the number of samples per line shall be even.
- Progressive video

The video sampling structure shall adhere to SMPTE ST 274 or SMPTE ST 2036-1 using VC3 in-band identification methods. The characteristics for other sampling structure mappings should be defined in an out-of-band bitstream container and will not be verified and ascertained by this decoder.

For $Y_{CB}C_R(A)$ video the decoder's video output signal shall consist of Luminance (Y) and color-difference components (C_B and C_R) with 4:2:0(:4) (8-, 10- or 12-bit), 4:2:2(:4) (8-, 10- or 12-bit) or 4:4:4(:4) (10- and 12-bit) sampling ratios respectively. The component sample levels are tabulated in Table 1.

For RGB(A) video the decoder's output signal shall either consist of 10- or 12-bit RGB(A) components. The component sample levels are tabulated in Table 2.

The decoder's video output signal for RGB(A) video shall consist of Red, Green and Blue components and an optional Alpha component. The RGB(A) *encoder* may choose between encoding the video as RGB (RGB encoding mode) or $Y_{CB}C_R$ ($Y_{CB}C_R$ encoding mode) values in the bitstream if the input is signaled to comply with Recommendation ITU-R BT.709 or Recommendation ITU-R BT.2020. Only the RGB format shall be returned by the *decoder*. The component sample levels for $Y_{CB}C_R$ encoding are tabulated in Table 2.

RI rasters may be encoded as constant or variable bitrate bitstreams. The operation mode shall be constant for the bitstream and shall be signaled in the Compressed Frame Header (see Section 7.2.2).

To avoid confusion and retain full backwards encoding compatibility, a VC-3 encoder shall use the HD profile encoding, whenever applicable, instead of the (otherwise identical) RI profile.

6.3 Frame Structure

Table 1 and Table 2 list the component sample levels for all encodings supported by this standard.

Table 1 – YC_BC_R Video: Sample levels for 8-, 10- and 12-bit source sampling bit-depth (Informative)

Sampling bit depth			8-bit (0 ... 255)	10-bit (0 ... 1023)	12-bit (0 ... 4095)
Sample levels (BT 709 and BT 2020)	Y	Peak Range	1 ... 254	4 ... 1019	16 ... 4079
		White Level	235	940	3760
		Black Level	16	64	256
		Number of levels between white and black (inclusive)	220	877	3505
	C _B , C _R	Gray Level	128	512	2048
		Excursion	+/- 112	+/- 448	+/- 1792
Sample levels (out-of-band)	Y	Peak Range	0 ... 255	0 ... 1023	0 ... 4095
	C _B , C _R	Gray level	128	512	2048
		Excursion	+/- 127	+/- 511	+/- 2047
Alpha	A	Peak Range	0 ... 255	0 ... 1023	0 ... 4095

Table 2 – RGB Video: Sample levels for 10- and 12-bit source sampling bit-depth (Informative)

Sampling bit depth			10-bit (0...1023)	12-bit (0 ... 4095)
Sample levels (RGB encoding mode)	R,G,B	Peak Range	4...1019	16 ... 4079
		White Level	940	3760
		Black Level	64	256
		Number of levels between white and black (inclusive)	877	3505
Sample levels (YC _B C _R encoding mode)	Y	Peak Range	4...1019	16 ... 4079
		White Level	940	3760
		Black Level	64	256
		Number of levels between white and black (inclusive)	877	3505
	C _B , C _R	Gray Level	512	2048
		Excursion	+/- 448	+/- 1792
Sample levels (out-of-band)	R,G,B	Peak Range	0 ... 1023	0 ... 4095
Alpha	A	Peak Range	0 ... 1023	0 ... 4095

As shown in Figure 2, the active video raster shall be subdivided into macroblocks, which are 16x16 blocks of spatially adjacent samples. A macroblock scan line shall be defined as a row of macroblocks

running the entire width of the video raster. The following variables are defined to specify the video raster subdivision into macroblocks:

N_W = the number of macroblocks per macroblock scan line; that is, the *width* of the macroblock scan line.

N_S = the number of macroblock scan lines in a frame or field.

N_T = the total number of macroblocks in a frame or field. Note that $N_T = N_W * N_S$.

l = macroblock index; $l = 0, 1, 2, \dots, N_T$, running through consecutive macroblock scan lines.

All rasters for which the number of active lines is not a multiple of 16 shall require additional (augmentation) lines to increase the total number of lines in the bitstream to the next multiple of 16 (Figure 2 below). During decoding the data contained in these augmentation lines shall be discarded.

RI rasters with a number of active samples per line not divisible by 16 shall require additional augmentation samples (padding) to increase the total number of samples per line in the bitstream to the next multiple of 16 (Figure 2 below). During decoding the data contained in these augmentation samples shall be discarded.

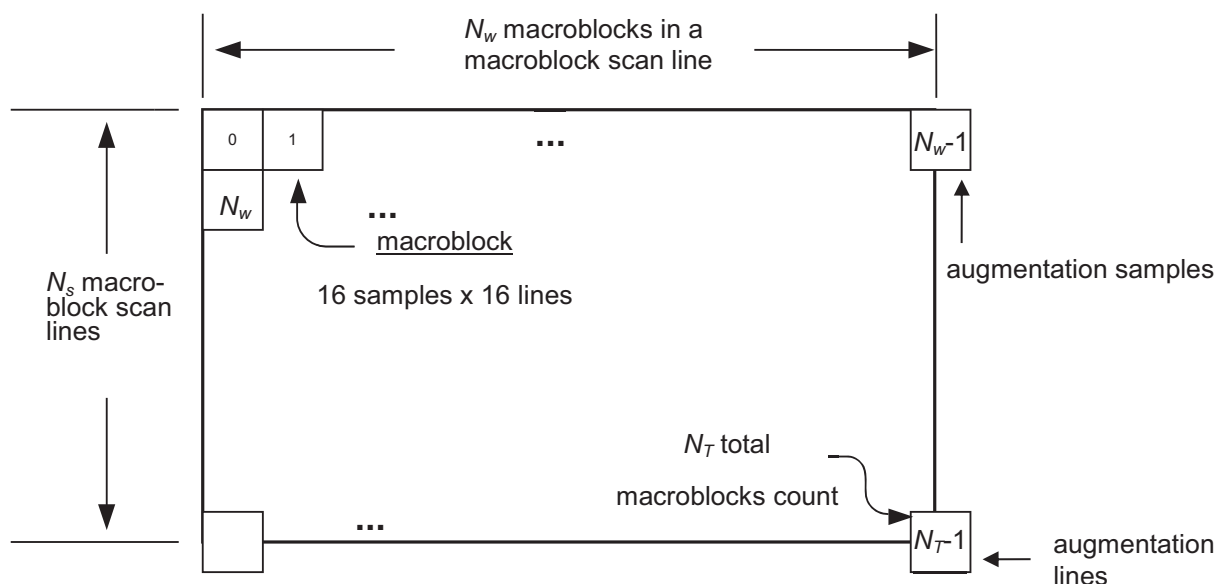


Figure 2 – Macroblock subdivision of the video raster

The macroblock subdivisions for the HD raster sizes supported by VC-3 shall be as shown in Figure 3 thru Figure 8 and tabulated in Table 3. These figures specify how to map individual macroblocks to a specific range of sample locations in the video raster.

As indicated in Table 3, three HD video rasters (bottom three rows) contain fewer samples per line than required by SMPTE ST 274 for 1080 line video rasters and SMPTE ST 296 for 720 line rasters. For these three cases, the video sampling structure shall be defined as a $\frac{3}{4}$ horizontally subsampled raster resulting in 1440 luma and 720 color-difference samples per line for 1080i and 1080p video and resulting in 960

luma and 480 color-difference samples per line for 720p video. These three sub-sampled rasters shall be denoted as 1080p - thin raster, 1080i - thin raster, and 720p - thin raster.

Note: Interlaced video can be partitioned on a field basis as required for field encoding (Figure 4), or on a frame basis as required for frame encoding (Figure 6).

Table 3 – Macroblock related parameters for the subdivision of the HD video raster

Source scan type	Lines per frame	Samples per line	Number of macroblocks per scan line: N_W	Number of macroblock scan lines per frame or field: N_S	Number of macroblocks per frame or field: N_T
Progressive	1080	1920	120	68 (frame)	8160 (frame)
Interlaced	1080	1920	120	34 (field)	4080 (field)
Progressive	720	1280	80	45 (frame)	3600 (frame)
Progressive	1080	1440	90	68 (frame)	6120 (frame)
Interlaced	1080	1440	90	68 (frame) ¹⁾	6120 (frame) ¹⁾
Progressive	720	960	60	45 (frame)	2700 (frame)

¹⁾ Note: Supported only for CID 1260; the MACF constraint to frame-based encoding for this CID is in Section 7.2.2.

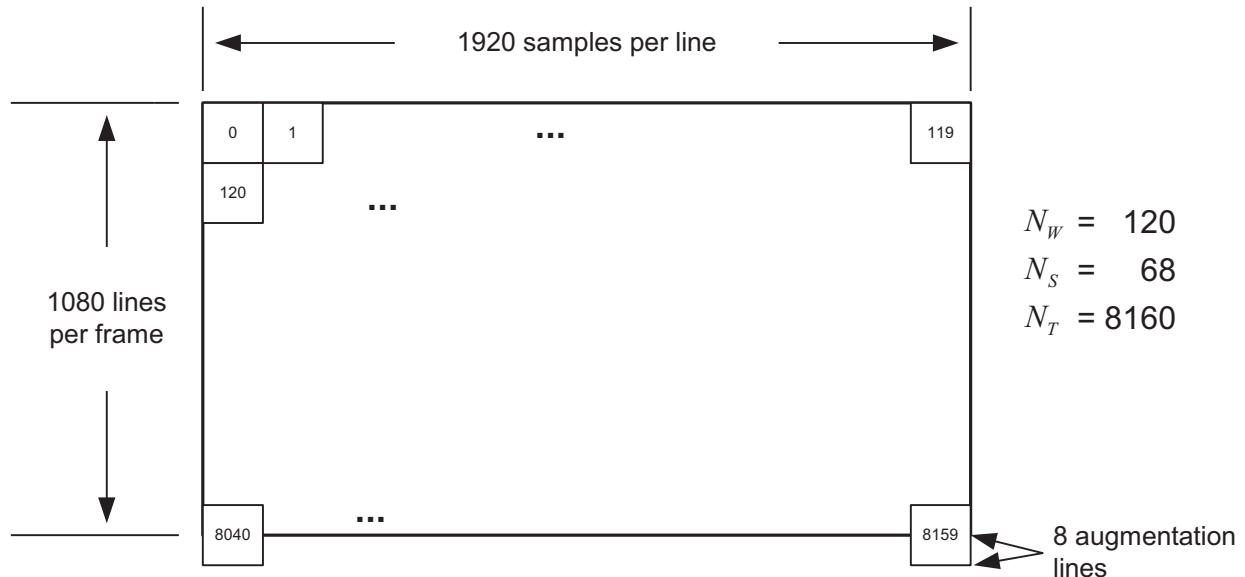


Figure 3 – 1080p raster subdivision into macroblocks

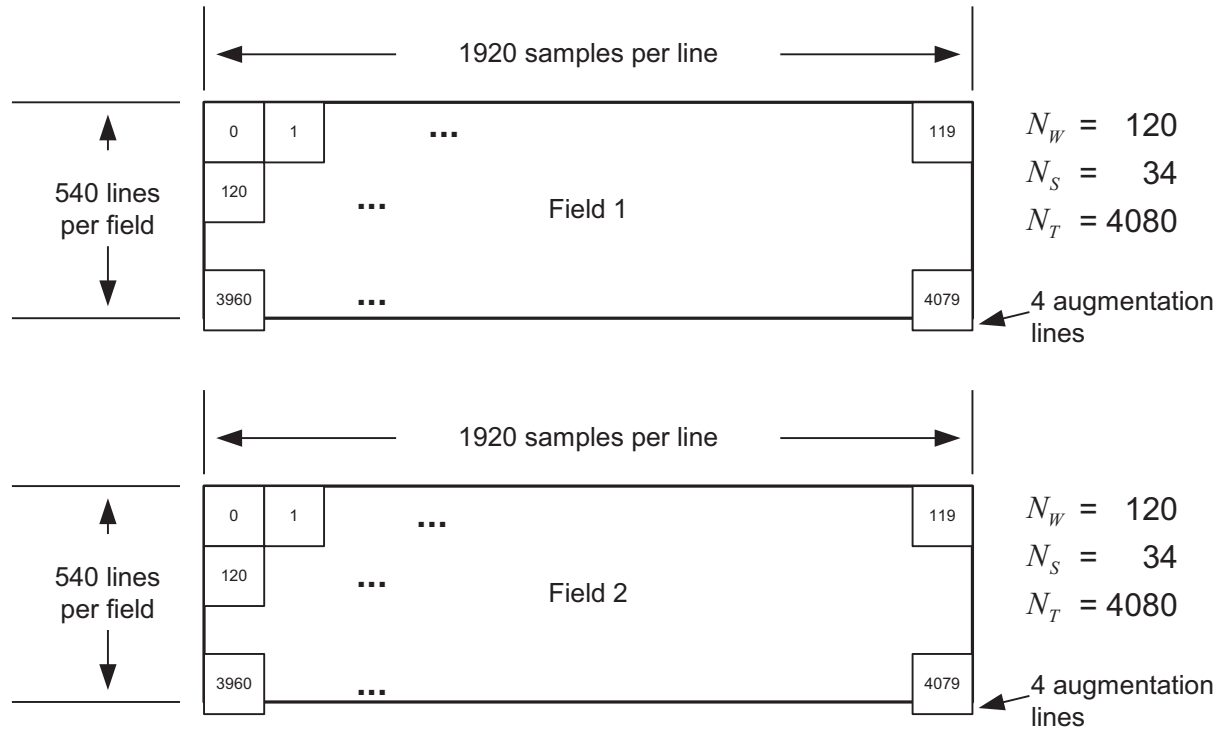


Figure 4 – 1080i raster sub-division into macroblocks

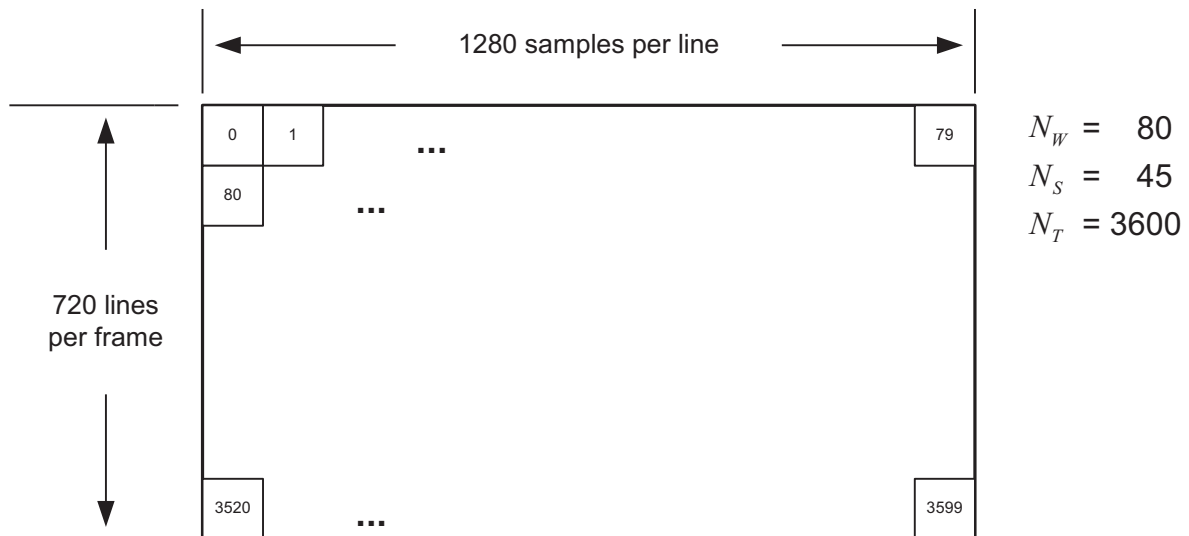


Figure 5 – 720p raster subdivision into macroblocks

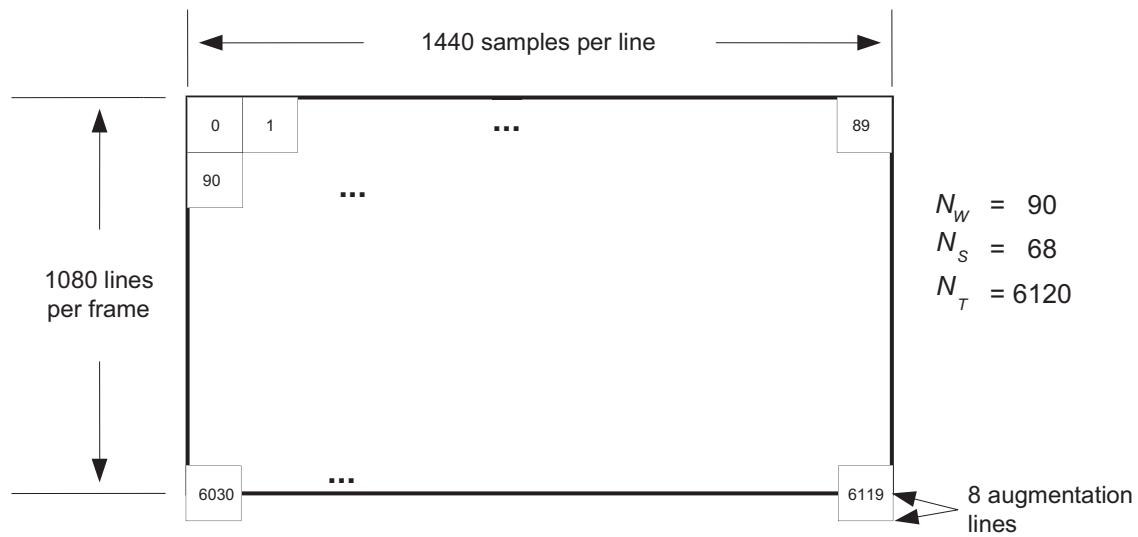


Figure 6 – 1080p – thin raster subdivision into macroblocks

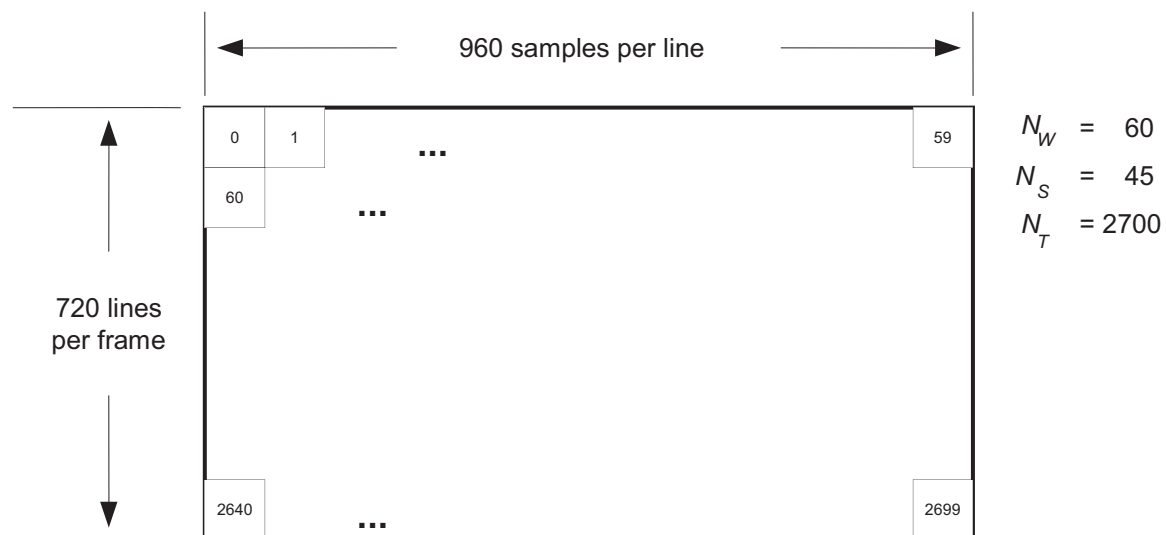


Figure 7 – 720p – thin raster subdivision into macroblocks

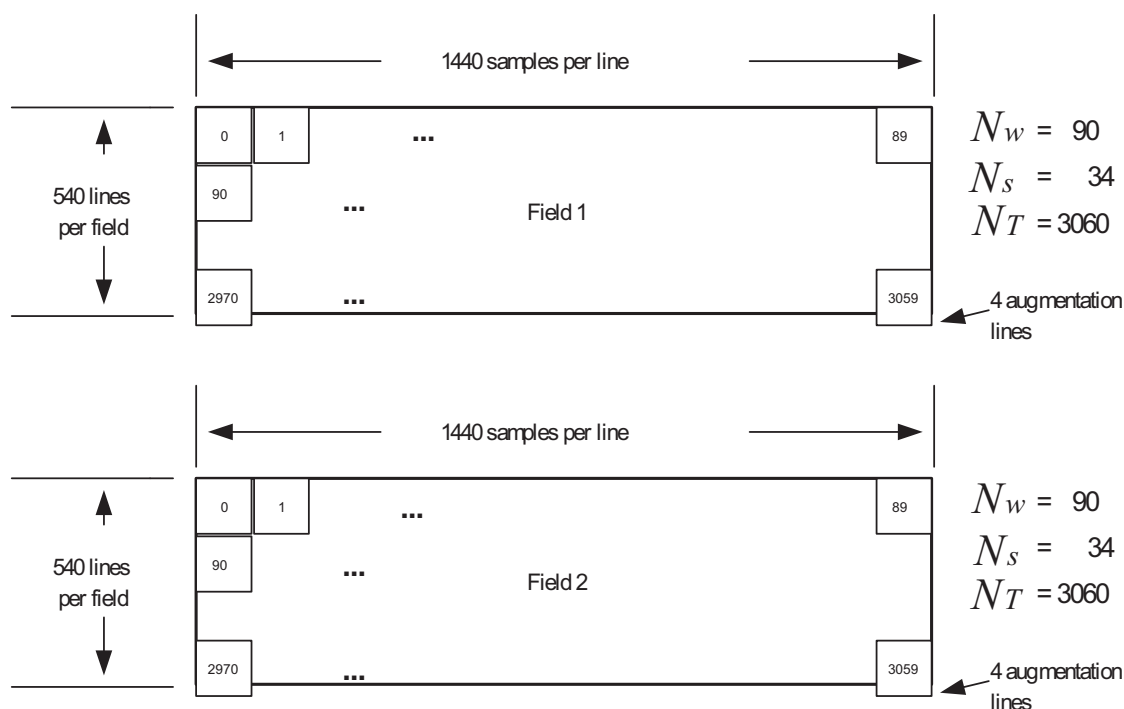


Figure 8 – 1080i – thin raster subdivision into macroblocks - field encoding

For RI rasters the macroblocks are organized in the same manner. The number of macroblocks in a row is the (augmented) raster width divided by 16, the number of macroblocks in a column is the (augmented) raster height divided by 16. The (optional) Alpha plane shall follow the definitions of the primary video.

For 4:2:0 ($Y C_B C_R$) sampling, each macroblock shall be further subdivided into 8x8 pixel DCT blocks, as shown in Figure 9. Each DCT block shall consist of 8 pixels x 8 lines. Due to the 4:2:0 chroma sub-sampling ratio a DCT shall span 16 samples x 16 lines. The DCT blocks shall be ordered according to the indices listed in Table 4.

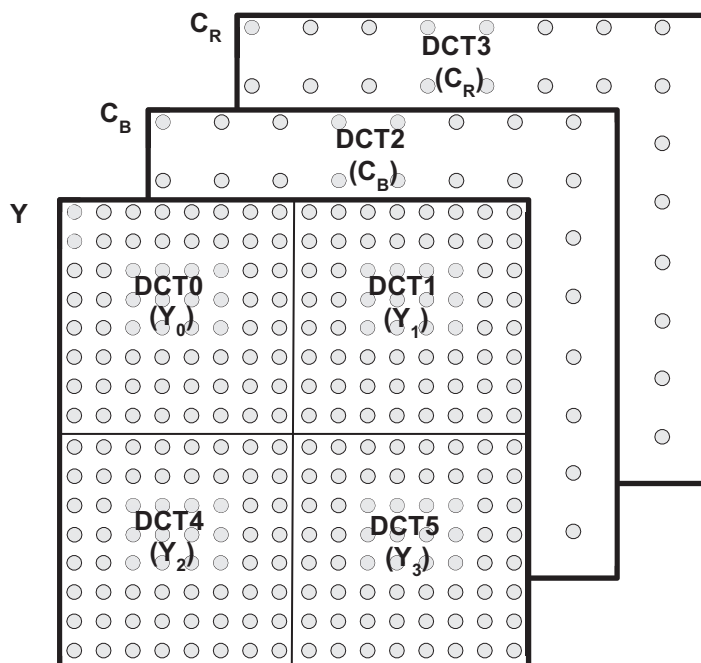


Figure 9 – Macroblock subdivision into 8x8 DCT blocks for YCbCr 4:2:0 sampling

Table 4 – DCT Block and DCT Index Correspondence for YCbCr 4:2:0 sampling

DCT Index k	DCT Block
0	Y_0
1	Y_1
2	C_{B0}
3	C_{R0}
4	Y_2
5	Y_3

For 4:2:2 (YCbCr) sampling, each macroblock shall be subdivided into 8x8 pixel DCT blocks, as shown in Figure 10. Each DCT block shall consist of 8 pixels x 8 lines. Due to the 4:2:2 chroma sub-sampling ratio a color-difference DCT block shall span 16 samples x 8 lines. The DCT blocks shall be ordered according to the indices listed in Table 5.

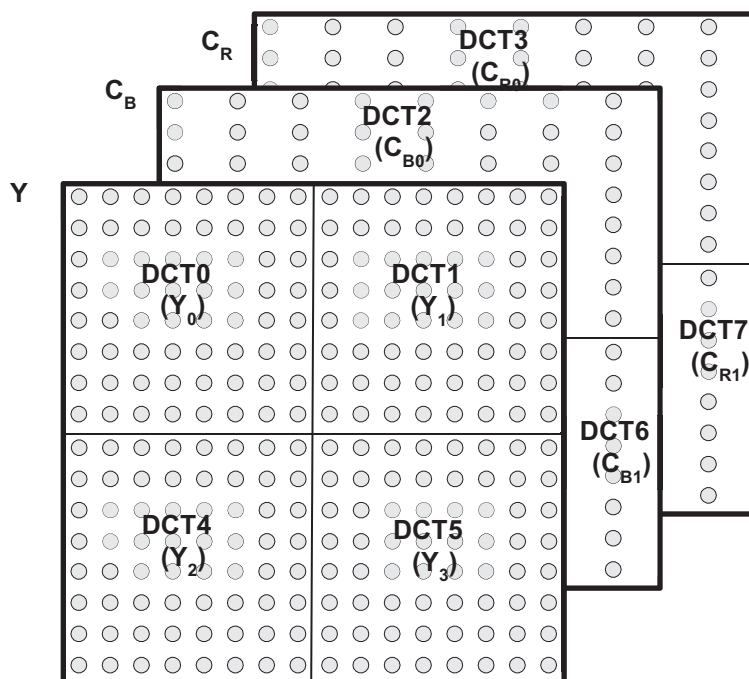


Figure 10 – Macroblock subdivision into 8x8 DCT blocks for $YC_B C_R$ 4:2:2 sampling

Table 5 – DCT Block and DCT Index Correspondence for $YC_B C_R$ 4:2:2 sampling

DCT Index k	DCT Block
0	Y_0
1	Y_1
2	C_{B0}
3	C_{R0}
4	Y_2
5	Y_3
6	C_{B1}
7	C_{R1}

For 4:4:4 $YC_B C_R$ (RI) and RGB sampling, each macroblock shall be further subdivided into 8x8 pixel DCT blocks, as shown in Figure 11. Each DCT block shall consist of 8 pixels x 8 lines. The DCT blocks shall be ordered according to the indices listed in Table 6.

For RGB source sampling complying to ITU-R.BT 709 or ITU-R.BT 2020 the encoder can choose between encoding the original RGB values or equivalent $YC_B C_R$ values on a macroblock basis: If the macroblock header bit-field $ACF = 0b$, the macroblock shall be encoded in *RGB Mode* and Ch1 shall be the red channel, Ch2 shall be the green channel, and Ch3 shall be the blue channel. If the macroblock header bit-field $ACF = 1b$, the macroblock shall be encoded in *$YC_B C_R$ Mode* and Ch1 shall be the Luma channel, Ch2 shall be the C_B channel, and Ch3 shall be the C_R channel. See Section 7.3.1.1 for the definition of the ACF field.

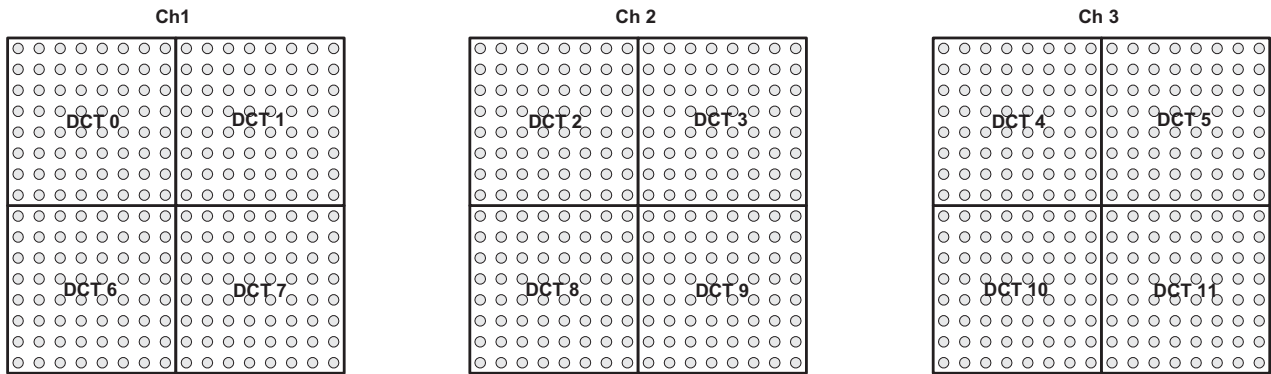


Figure 11 – Macroblock subdivision into 8x8 DCT blocks for 4:4:4 YC_BC_R and RGB sampling

Table 6 – DCT Block and DCT Index Correspondence for 4:4:4 YC_BC_R and RGB sampling

DCT Index k	DCT Block (RGB mode)	DCT Block (YC _B C _R mode)
0	R ₀	Y ₀
1	R ₁	Y ₁
2	G ₀	C _{B0}
3	G ₁	C _{B1}
4	B ₀	C _{R0}
5	B ₁	C _{R1}
6	R ₂	Y ₂
7	R ₃	Y ₃
8	G ₂	C _{B2}
9	G ₃	C _{B3}
10	B ₂	C _{R2}
11	B ₃	C _{R3}

The variable k denotes the DCT block index number, which ranges from 0 to 7. Figure 12 depicts the 8x8 block of video samples, $x_k^f(i, j)$ which comprises a DCT block. The index i and line index j range from 0 to 7.

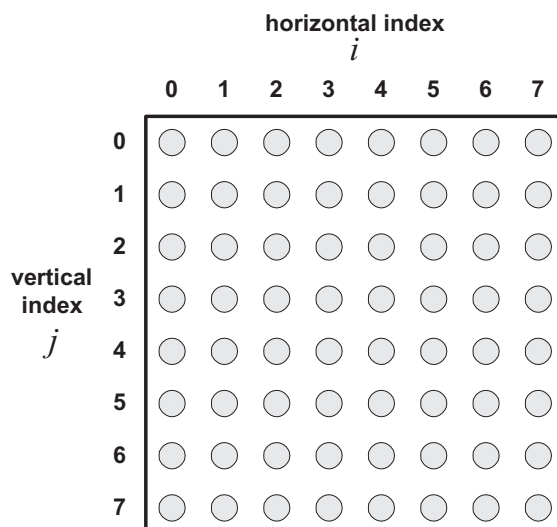


Figure 12 – $x_k^l(i, j)$: A block of 8x8 video sample data

For Compression ID 1260, two different macroblock modes are permissible. In Frame Macroblock Mode, the macroblock shall be partitioned into 8x8 DCT blocks, where each DCT block is comprised of data from both fields. This is depicted in Figure 13 where the top field is indicated by the lines with solid fill, and the bottom field is indicated by lines with no fill. In Field Macroblock Mode, depicted in Figure 14, the DCT blocks shall be comprised of only lines from a single field either top or bottom.

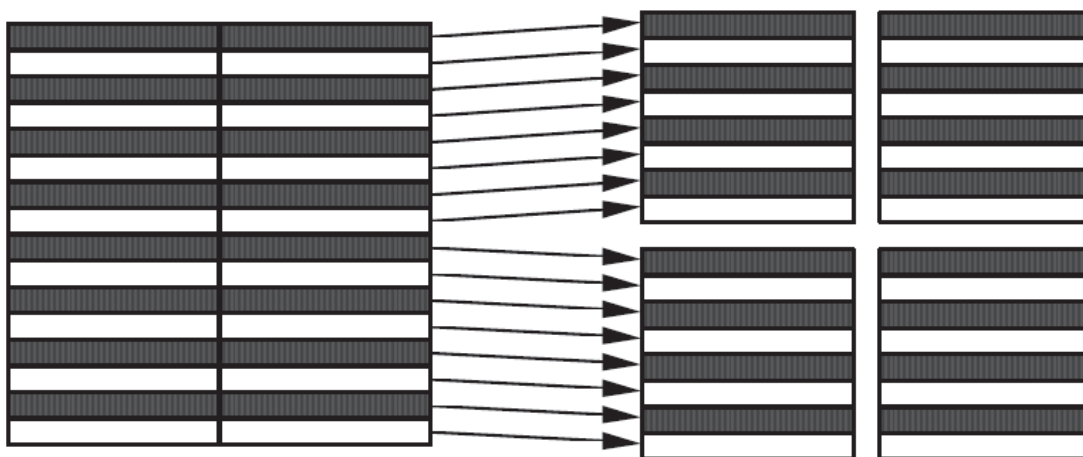


Figure 13 – Frame Macroblock Mode

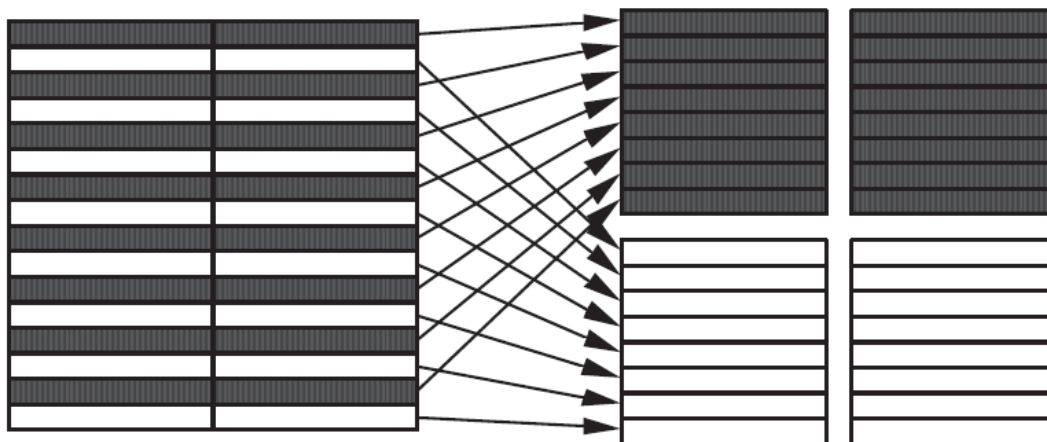


Figure 14 – Field Macroblock Mode

Alpha is broken down into analogous Alpha macroblocks sized 16 samples x 16 lines, matching the location of the macroblocks of the primary video.

For DCT-based alpha compression these macroblock shall be further subdivided into 4 DCT blocks, as shown in Figure 15. Each DCT block shall consist of 8 pixels x 8 lines. The DCT blocks for the Alpha channel shall be ordered as listed in Table 7 and shall use the same quantization matrix as defined for the Y component of the compression ID.

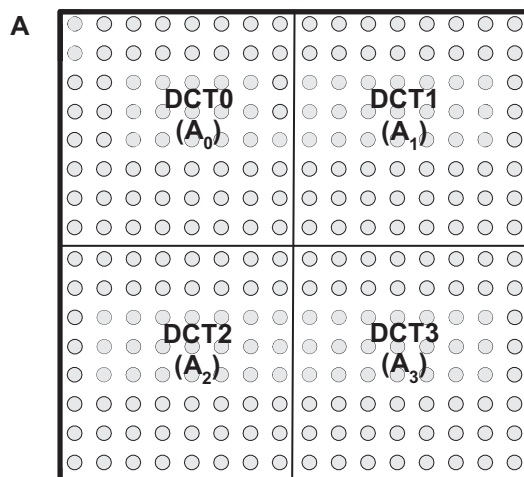
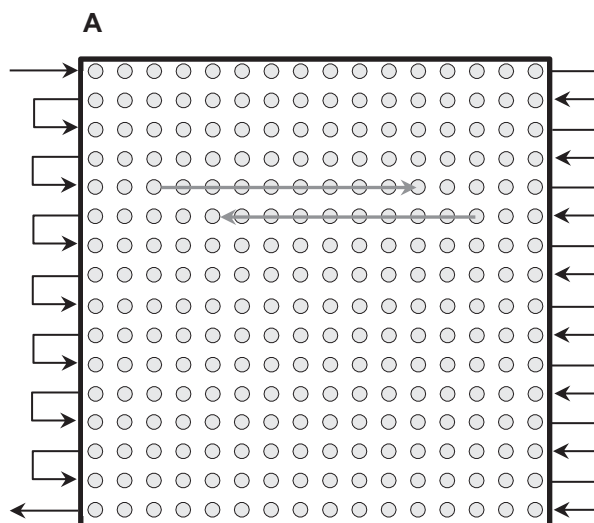


Figure 15 – Macroblock subdivision into 8x8 DCT blocks for Alpha sampling

Table 7 – DCT Block and DCT Index Correspondence for Alpha sampling

DCT Index k	DCT Block
0	A_0
1	A_1
2	A_2
3	A_3

For lossless differential RLE-compression the Alpha block shall be scanned as shown in Figure 16 (Alpha Zig-Zag).

**Figure 16 – Zig-zag scan of 16x16 Alpha blocks (Differential RLE encoding)**

7 Compressed Frame Format

A VC-3 compressed bitstream shall be comprised of continuously concatenated coding units (Figure 17). A coding unit shall consist of a Header, a Compressed Payload, and an End-Of-Frame (EOF) signature of 4 bytes. Table C.1 and Table C.2 specify the Compressed Frame, Coding Unit, and Compressed Payload sizes for the range of VC-3 bitstreams with the corresponding Compression IDs.

The Header shall be as described in Section 7.2. It shall contain parameters required to decode Compressed Data contained in the Compressed Payload, which is described in Section 7.3. The Compressed Payload shall contain the Coding Unit's compressed data, see Section 7.3.1. If the length of the Compressed Data is less than the Compressed Payload Size for CBR encoding, the remainder of the Compressed Payload Area shall be filled with padding bytes as detailed in Section 7.3.2. The padding bytes shall be stored in the Compressed Payload Padding area. This padding step shall be omitted for VBR coding.

The compressed data for a single video frame shall require either one or two coding units:

- Progressive video shall require one coding unit and be encoded using frame encoding.
- Interlaced video shall require either one coding unit (frame encoding) or two coding units (field encoding, see Figure 18).

In the case of frame encoding, a single coding unit shall contain the compressed data representing all the video data of a single frame. In the case of the field encoding, two coding units shall hold the compressed data for a single frame. The first coding unit shall contain the compressed representation of the first video field, while the second coding unit shall contain the representation of the second video field. The second coding unit shall be concatenated onto the end of the first to create a compressed frame.

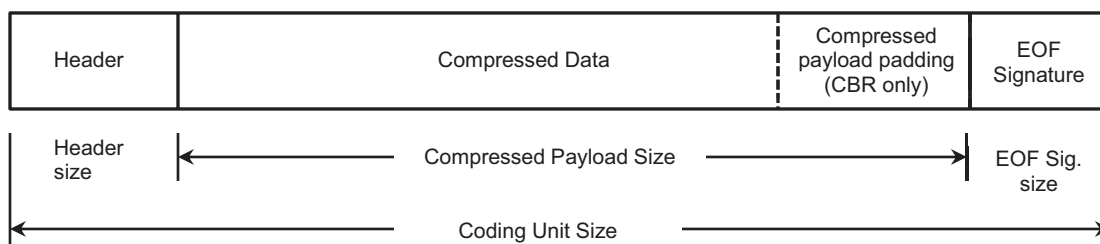


Figure 17 – Coding unit data structure

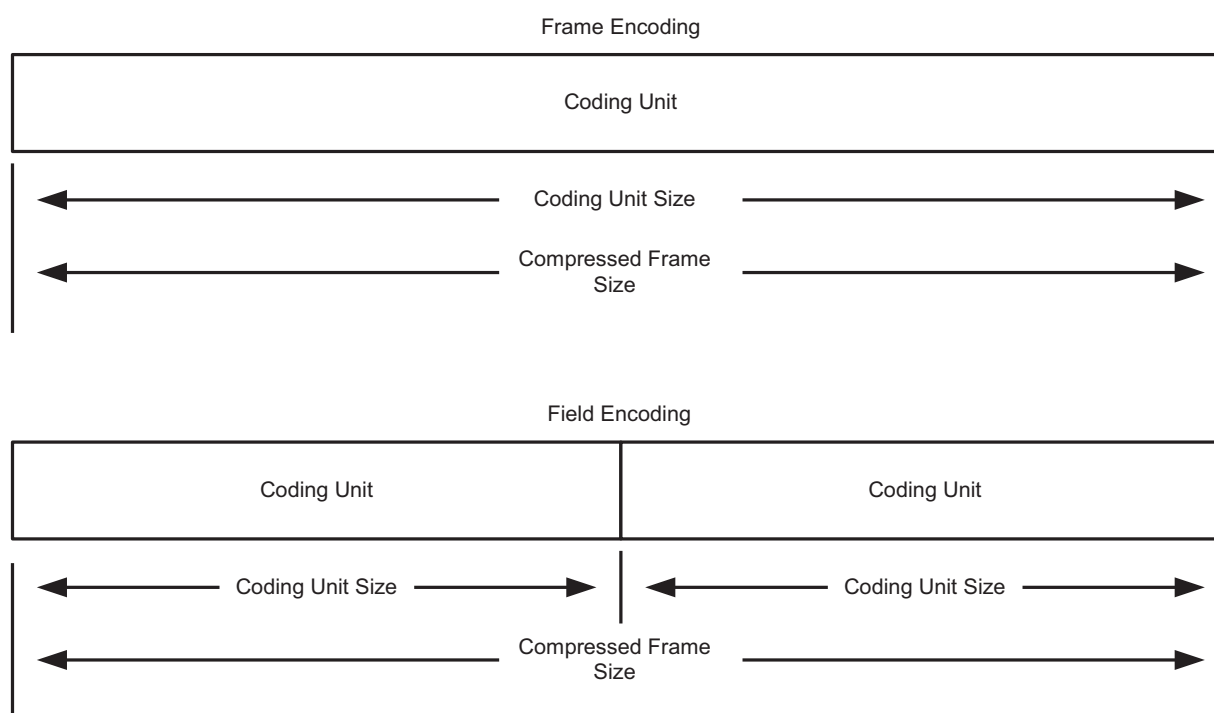


Figure 18 – Compressed frame data structure

7.1 Compressed Frame Size

For HD rasters the number of compressed bytes per frame C_{ref} for a particular Compression ID shall be as specified in Table C.1. HD rasters can only be compressed in CBR mode.

For RI rasters and CBR encoding C_{ref} shall be calculated according to

$$C_{ref}(f_A) = \left[(1 + f_A) \cdot \left(C_0 * \frac{N_T}{M_T} \right) \right]_{4096}, \min(C_{ref}) = 8192 \quad (7.1)$$

where C_0 and M_T denote the reference frame size and macroblock count for the Compression ID, taken from Table C.2, N_T is the total number of macroblocks calculated according to Section 6 and $f_A = 0.5$ is a constant, encoding-independent augmentation factor if alpha is present. The minimum number accepted for C_{ref} is 8192.

For RI rasters and VBR encoding C_{ref} shall denote the maximum compressed frame size permitted, except if lossless alpha compression is used. For this case, $C_{ref}(0)$ shall denote the maximum compressed frame size of non-Alpha components, to which the variable size alpha encoding is then added.

Note 1: The following pseudo code calculates the compressed frame size of an RI raster of dimensions *width* and *height*, according to (7.1).

```

RISize(width, height,  $f_A$ ,  $C_0$ )
{
     $N_W = \text{width}/16$ ;
    if (width % 16 != 0)
         $N_W++$ ;                // Padding required

     $N_S = \text{height}/16$ ;
    if (height % 16)
         $N_S++$ ;                // Padding required

    size =  $(1+f_A) * C_0 * (N_W * N_S) / 8160$ ;
    rem = size % 4096;

    if (rem >= 2048)           // Round-up limit
        size += 4096-rem;     // Round UP
    else
        size -= rem;          // Round DOWN

    if (size < 8192)           // Lower size limit
        size = 8192;

    return size;
}

```

Note 2: The video frame rate at which the compression is applied will determine the compressed video *bitrate*. For example, the Compressed Frame Size of Compression ID 1241 is 917504 bytes. At a video frame rate of 29.97 frames per second, the compressed bitrate is $8 * 29.97 * 917504 = 219.98$ Mbps. The informative Annex G, "Compressed Bitrates" provides examples of bitrates as a function of Compression ID and video frame rate.

Note 3: For lossless Alpha/VBR the maximum compressed frame size is approximately given by

$$C_{maxVBR} \approx C_{ref}(0) + N_T \cdot C_A$$

where C_A is the size of an Alpha macroblock stored as packed raw according to Section 7.3.1.2.

7.2 Header Definition

Header size shall depend on Compression ID. For HD rasters the contents of the 640 byte Header area shall be as summarized in Table 8. For RI rasters the header size shall depend on the raster height. If the raster height is less or equal 1088, the header size shall be 640 bytes. For each additional 16 lines (whole or partial), 4 bytes shall be added to the header size, which shall be recorded in the header prefix.

The following subsections define the header data regions. The byte offsets shown in Table 8 and the figures of this section shall be relative to the start of the coding unit. The content of the reserved areas shall be 0b. Decoder implementations should not process these regions.

Table 8 – Content of header data regions

Starting Offset	Data Region Length (bytes)	Header Data Region Contents	Subsection
0x000	0x005	Header Prefix	7.2.1
0x005	0x003	Coding Control A	7.2.2
0x008	0x010	Reserved	—
0x018	0x00B	Image Geometry	7.2.3
0x023	0x005	Reserved	—
0x028	0x004	Compression ID	7.2.4
0x02C	0x001	Coding Control B	7.2.5
0x02D	0x003	Reserved	—
0x030	0x001	Time Code Control	7.2.6
0x031	0x008	Time Code	7.2.7
0x039	0x026	Reserved	—
0x05F	0x001	User Data Control	7.2.8
0x060	0x104	User Data Payload	7.2.9
0x164	0x003	Reserved	—
0x167	0x009	Macroblock Scan Indices Control	7.2.10
0x170	At least 0x110	Macroblock Scan Indices Payload	7.2.11

7.2.1 Header Prefix

The header prefix shall consist of the five bytes given in Table 9.

Table 9 – Header Prefix data area contents

Byte Offset	Contents
0x000	HS ₀
0x001	HS ₁
0x002	HS ₂
0x003	HS ₃
0x004	HVN

HS: Header Size. The value of HS shall depend on values of Compression ID, samples per line and active lines per frame.

$$HS = (HS_0 \ll 24) + (HS_1 \ll 16) + (HS_2 \ll 8) + HS_3$$

HVN: Header Version Number. The value of HVN shall be dependent upon Compression ID as follows

0x01: Header version 1: Compression ID 1235, 1237, 1238, 1241, 1242, 1243, 1244, 1250, 1251, 1252, 1253

0x02: Header version 2: Compression ID 1256, 1258, 1259, 1260

0x03: Header version 3: Compression ID 1270, 1271, 1272, 1273, 1274

For HVN values 0x01 and 0x02 the values of HS₀, HS₁, HS₂, and HS₃ shall be 0x00, 0x00, 0x02, and 0x80 respectively.

HVN values 0x01 and 0x02 are HD profile.

HVN value 0x03 is RI profile.

Note: An encoder creating e.g. a Compression ID 1235 bitstream can write HVN=0x01 into the bitstream even when implemented according to this version of the standard. The HVN signals compliance of the bitstream to the definitions of specific Compression IDs, not to a specific version of this standard.

7.2.2 Coding Control A

Coding Control A area shall hold the following data fields as shown in Figure 19.

Byte Offset	msb						lsb	
0x005	0	0	0	VBR	0	0	FFC ₁	FFC ₀
0x006	1	0	MACF	CRCF	0	0	0	0
0x007	1	0	1	0	0	PMA	LLA	ALP

Figure 19 – Coding Control A data area

VBR: Variable Bitrate Encoding Flag

The value of the VBR flag shall be constant throughout the bitstream and signal the absence of the Compressed Payload Padding in the bitstream.

- 0b: Constant bitrate encoding (CBR).
- 1b: Variable bitrate encoding (VBR). Only CID 1270-1274.

FFC: Field/Frame Count

- 01b: Progressive frame or interlaced frame in frame encoded mode.
- 10b: Field 1 of interlaced frame in field encoded mode.
- 11b: Field 2 of interlaced frame in field encoded mode.

MACF: Macroblock Adaptive Control Flag:

The MACF bit-field shall control the indication of adaptive Macroblock Field/Frame selection for macroblocks.

- 0b: Macroblocks do not use field macroblock mode
- 1b: Macroblocks may use either field or frame macroblock mode

MACF == 1 b shall always occur for CID 1260 and shall be 0 b for all other Compression IDs.

CRCF: CRC flag

- 0b: CRC shall not be present in EOF signature
- 1b: CRC shall be present in EOF signature

ALP: Alpha flag (RI-only, shall be 0b for HD rasters)

- 0b: Alpha channel not present
- 1b: Alpha channel present; only CID=1270, 1271, 1272 and 1273

LLA: Lossless Alpha flag (only CID=1270; shall be 0b for all others)

- 0b: Alpha, if present, is encoded using DCT-based compression
- 1b: Alpha, if present, is encoded using differential RLE compression.

PMA: Pre-multiplied Alpha (RI only; shall be 0b for HD rasters)

- 0b: Alpha has not been applied to video channels
- 1b: Alpha has been applied to the video channels prior to encoding.
Requires out-of-band encoding (CLV=11b, see Section 7.2.5, Coding Control B).

7.2.3 Image Geometry

The Image Geometry area shall hold the following data fields as shown in Figure 20.

Byte Offset	msb					lsb		
0x018	ALPF ₁							
0x019	ALPF ₀							
0x01A	SPL ₁							
0x01B	SPL ₀							
0x01C	0	0	0	0	PARC ₁	PARN ₁		
0x01D	NAL ₁							
0x01E	NAL ₀							
0x01F	PARC ₀							
0x020	PARN ₀							
0x021	SBD			1	1	0	0	0
0x022	1	0	0	0	1	SST	0	0

Figure 20 – Image Geometry data area

ALPF: For HD rasters the Active Lines Per Frame value shall depend on the compression ID. The range of permissible values shall be as listed in the “Active lines” column of Table C.1.

For RI rasters the active lines-per-frame value shall not depend on a compression ID. The value of active lines-per-frame shall be no less than 1 and no more than 16384.

$$\text{ALPF} = (\text{ALPF}_1 \ll 8) + \text{ALPF}_0.$$

SPL: For HD rasters the samples-per-line value shall depend on the compression ID. The range of permissible values shall be as listed in the “Samples per line” column of Table C.1.

For RI the samples-per-line value shall not depend on a compression ID.

The value of samples-per-line shall be no less than 1 and no more than 16384.

If the sample-per-line value is not divisible by 16 each line shall be sample-padded in the encoded bitstream to the next multiple of 16.

$$\text{SPL} = (\text{SPL}_1 \ll 8) + \text{SPL}_0.$$

NAL: Number of Active Lines shall have the same value as ALPF.

$$\text{NAL} = (\text{NAL}_1 \ll 8) + \text{NAL}_0.$$