FINAL
DRAFT

INTERNATIONAL
STANDARD

**ISO/IEC
FDIS
23094-2**

ISO/IEC JTC **1**/SC **29**

Secretariat: JISC

Voting begins on:
2021-06-14

Voting terminates on:
2021-08-09

Information technology – General video coding —

Part 2:
Low complexity enhancement video coding

Reference number
ISO/IEC FDIS 23094-2:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword ..... **v**

1 Scope **1**

2 Normative references **1**

3 Terms and definitions **1**

4 Abbreviated terms **7**

5 Conventions **7**
5.1 General 7
5.2 Arithmetic operators 7
5.3 Logical operators 8
5.4 Relational operators 8
5.5 Bit-wise operators 9
5.6 Assignment operators 9
5.7 Range notation 9
5.8 Mathematical functions 10
5.9 Order of operation precedence 10
5.10 Variables, syntax elements and tables 11
5.11 Text description of logical operations 12
5.12 Processes 13

6 Bitstream and picture formats, partitioning, scanning processes and neighbouring relationships **14**
6.1 Bitstream formats 14
6.2 Source, decoded and output picture formats 14
6.3 Partitioning of pictures 17
6.3.1 Organisation of the hierarchical structure 17
6.3.2 Partitioning of residuals plane 17

7 Syntax and semantics **18**
7.1 Method of specifying syntax in tabular form 18
7.2 Specification of syntax functions and descriptors 19
7.3 Syntax in tabular form 20
7.3.1 Syntax order 20
7.3.2 NAL unit and NAL unit header syntax 20
7.3.3 Process block syntax 21
7.3.4 Process payload – sequence configuration 22
7.3.5 Process payload – global configuration 22
7.3.6 Process payload – picture configuration 24
7.3.7 Process payload – encoded data 25
7.3.8 Process payload – encoded tiled data 26
7.3.9 Process payload – surface 28
7.3.10 Process payload – additional info 29
7.3.11 Process payload – filler 29
7.3.12 Byte alignment syntax 29
7.4 Semantics 30
7.4.1 General 30
7.4.2 NAL unit semantics 30
7.4.3 Data block unit configuration semantics 32

8 Decoding process **41**
8.1 General decoding process 41
8.2 Payload data block unit process 42
8.3 Picture enhancement decoding process 42
8.3.1 General enhancement decoding process 42
8.3.2 Decoding process for picture enhancement encoded data (payload_encoded_data) 42

8.3.3 Decoding process for picture enhancement encoded tiled data (payload_encoded_tiled_data) 45
8.3.4 Decoding process for enhancement sub-layer 1 (L-1) encoded data 50
8.3.5 Decoding process for enhancement sub-layer 2 (L-2) encoded data 52
8.4 Decoding process for the temporal prediction 54
8.4.1 General decoding process for temporal prediction 54
8.4.2 Tiled temporal refresh 54
8.5 Decoding process for the dequantisation 54
8.5.1 Decoding process for the dequantisation overview 54
8.5.2 Scaling process for transform coefficients 54
8.5.3 Derivation of dequantisation offset and stepwidth modifier 55
8.5.4 Derivation of quantisation matrix 56
8.6 Decoding process for the transform 59
8.6.1 General upscaling process description 59
8.6.2 Transform inputs and outputs, transform types, and residual samples derivation 64
8.6.3 2x2 directional decomposition transform 64
8.6.4 4x4 directional decomposition transform 65
8.7 Decoding process for the upscaling 66
8.7.1 Nearest sample upsampler kernel description 66
8.7.2 Bilinear upsampler kernel description 67
8.7.3 Cubic upsampler kernel description 70
8.7.4 Modified Cubic upsampler kernel description 73
8.7.5 Predicted residual process description 74
8.7.6 Adaptive Cubic upsampler kernel description 75
8.8 Decoding process for the residual reconstruction 75
8.8.1 Reconstructed residual of each block derivation 75
8.8.2 Residual reconstruction for L-1 block 76
8.8.3 Residual reconstruction for L-2 block 76
8.9 Decoding process for the L-1 filter 77
8.9.1 L-1 residual filter overview 77
8.9.2 Decoding process for filtering L-1 block 77
8.10 Decoding process for base decoder data extraction 79
8.11 Decoding process for dither filter 79

9 Parsing process 79
9.1 Parsing process inputs and outputs, process overview 79
9.1.1 Parsing process for entropy encoded transform coefficients 79
9.1.2 Parsing process for entropy encoded temporal signal coefficient group 82
9.2 Prefix Coding decoder 83
9.2.1 Prefix Coding decoder description 83
9.2.2 Prefix Coding decoder table generation 84
9.2.3 Prefix Coding decoder for tile data sizes 88
9.3 RLE decoder 89
9.3.1 RLE process inputs and outputs 89
9.3.2 RLE decoder for coefficient groups 89
9.3.3 RLE decoder description 90
9.3.4 RLE decoder for temporal signal coefficient group 91
9.3.5 RLE decoder for tile entropy_enabled_flag fields 93
9.4 Parsing process for 0-th order Exp-Golomb codes 94

Annex A (normative) **Profiles and levels 96**
Annex B (normative) **Byte stream format 98**
Annex C (normative) **Hypothetical reference decoder 101**
Annex D (normative) **Supplemental enhancement information 108**
Annex E (normative) **Video usability information 115**
Bibliography 119

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23094 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Information technology – General video coding —

Part 2:
Low complexity enhancement video coding

1 Scope

This document specifies low complexity enhancement video coding.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11578:1996, *Information technology — Open systems interconnection — Remote procedure call (RPC)*

ITU-T H.273 | ISO/IEC 23091-2:2019, *Information technology — Coding-independent code points — Part 2: Video*

ITU-T Recommendation T.35:2000, *Procedure for the allocation of ITU-T defined codes for non-standard facilities*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
access unit
AU
set of *NAL units* (3.35) that are associated with a particular output time, are consecutive in *decoding order* (3.20), and contain exactly one *coded picture* (3.9)

3.2
bitstream
sequence of bits, in the form of a *NAL unit stream* (3.36) or a *byte stream* (3.6), that forms the representation of *coded pictures* (3.9), and associated data forming one or more coded video sequences (CVSs)

3.3
block
MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients* (3.57)

3.4
byte
sequence of 8 bits, within which, when written or read as a sequence of bit values, the left-most and right-most bits represent the most and least significant bits, respectively

3.5
byte-aligned
position in a *bitstream* (3.2) in which the position is an integer multiple of 8 bits from the position of the first bit in the *bitstream*

Note 1 to entry: A bit, *byte* (3.4) or *syntax element* (3.53) is said to be *byte-aligned* when the position at which it appears in a *bitstream* (3.2) is *byte-aligned*.

3.6
byte stream
encapsulation of a *NAL unit stream* (3.36) containing *start code prefixes* (3.51) and *NAL units* (3.35)

3.7
chroma
sample array or single sample is representing one of the two colour difference signals related to the primary colours, represented by the symbols Cb and Cr

Note 1 to entry: The term *chroma* is used rather than the term chrominance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term chrominance.

3.8
chunk
entropy coded portion of data containing the quantised *transform coefficient* (3.57) belonging to a coefficient group

3.9
coded picture
coded representation (3.10) of a *picture* (3.40) containing all *TUs* (3.58) of the *picture*

3.10
coded representation
data element as represented in its coded form

3.11
coded video sequence
CVS
coded sequence of *access units* (3.1)

3.12
coding block
MxN *block* (3.3) of samples for some values of M and N

3.13
coding unit
CU
32 x 32 *block* (3.3) of samples resulting from the parsing of the entropy encoded *transform coefficients* (3.57) in the *decoding process* (3.22)

3.14
coefficient group
CG
syntactical structure containing coded data related to a specific set of *transform coefficients* (3.57)

3.15
component
array or single sample from one of the three arrays (*luma* (3.34) and two *chroma* (3.7)) that compose a *picture* (3.40) in 4:2:0, 4:2:2, or 4:4:4 colour format, or the array or a single sample of the array that compose a *picture* in monochrome format

3.16
data block
syntax structure (3.54) containing *bytes* (3.4) corresponding to a type of data

3.17
decoded base picture
decoded picture (3.18) derived by decoding a *coded picture* (3.9) with a base *decoder* (3.20)

3.18
decoded picture
picture (3.40) derived by decoding a *coded picture* (3.9), and which is either a decoded *frame* (3.29) or a decoded *field* (3.28)

3.19
decoded picture buffer
DPB
buffer holding *decoded pictures* (3.18) for reference or output reordering

3.20
decoder
embodiment of a *decoding process* (3.22)

3.21
decoding order
order in which *syntax elements* (3.53) are processed by the *decoding process* (3.22)

3.22
decoding process
process specified that reads a *bitstream* (3.2) and derives *decoded pictures* (3.18) from it

3.23
emulation prevention byte
byte (3.4) equal to 0x03 that may be present within a *NAL unit* (3.35), the presence of which ensures that no sequence of consecutive *byte-aligned* (3.5) *bytes* in the *NAL unit* contains a *start code prefix* (3.51)

3.24
encoder
embodiment of an *encoding process* (3.25)

3.25
encoding process
process that produces a *bitstream* (3.2) conforming to this document

3.26
enhancement layer
layer (3.32) within the *bitstream* (3.2) pertaining to the *residual planes* (3.47)

3.27
enhancement sub-layer
layer (3.32) of the *enhancement layer* (3.26)

3.28
field
assembly of alternate rows of a *frame* (3.29)

3.29
frame
array of *luma* (3.34) samples in monochrome format or array of *luma* samples and two corresponding arrays of *chroma* (3.7) samples in 4:2:0, 4:2:2, and 4:4:4 colour format, and which consists of two *fields* (3.28): a top field and a bottom field

3.30
instantaneous decoding refresh picture
IDR picture
picture (3.40) for which a *NAL unit* (3.35) contains a global configuration data block and does not refer to any other *picture* for operation of the *decoding process* (3.22) of this *picture* and for which no subsequent *pictures* in *decoding order* (3.21) refer to any *picture* that precedes it in *decoding order*

Note 1 to entry: An *IDR picture* shall occur at least when an IDR picture for the base *decoder* (3.20) occurs. The IDR picture for a base *decoder* is not specified in this document.

3.31
inverse transform
part of the *decoding process* (3.22) by which a set of *transform coefficients* (3.57) is converted into *residuals* (3.46)

3.32
layer
one of a set of syntactical structures in a non-branching hierarchical relationship

3.33
level
defined set of constraints on the values that may be taken by the *syntax elements* (3.53) and variables of this document

Note 1 to entry: The same set of *levels* is defined for all *profiles* (3.41), with most aspects of the definition of each *level* being in common across different *profiles*. Individual implementations may, within specified constraints, support a different *level* for each supported *profile*.

3.34
luma
sample array or single sample representing the monochrome signal related to the primary colours, represented by the symbol or subscript Y or L

Note 1 to entry: The term *luma* is used rather than the term luminance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term luminance. The symbol L is sometimes used instead of the symbol Y to avoid confusion with the symbol y as used for vertical location.

3.35
network abstraction layer unit
NAL unit
syntax structure (3.54) containing an indication of the type of data to follow and *bytes* (3.4) containing that data in the form of an *RBSP* (3.42) interspersed as necessary with *emulation prevention bytes* (3.23)

3.36
network abstraction layer unit stream
NAL unit stream
sequence of *NAL units* (3.35)

3.37
output order
order in which the *decoded pictures* (3.18) are output from the *decoded picture buffer* (3.19) (for the *decoded pictures* that are to be output from the *decoded picture buffer*)

3.38
partitioning
division of a set into subsets such that each element of the set is in exactly one of the subsets

3.39
plane
collection of data related to plane Y (*luma* (3.34)) or C (*chroma* (3.7))

3.40
picture
field (3.28) or *frame* (3.29)

3.41
profile
specified subset of the syntax of this document

3.42
raw byte sequence payload
RBSP
syntax structure (3.54) containing an integer number of *bytes* (3.4) that is encapsulated in a *NAL unit* (3.35) and which is either empty or has the form of a *string of data bits* (3.52) containing *syntax elements* (3.53) followed by an *RBSP stop bit* (3.43) and followed by zero or more subsequent bits equal to 0

3.43
raw byte sequence payload stop bit
RBSP stop bit
bit equal to 1 present within an *RBSP* (3.42) after a *string of data bits* (3.52)

Note 1 to entry: The location of the end of the *string of data bits* within an *RBSP* can be identified by searching from the end of the *RBSP* for the *RBSP stop bit*, which is the last non-zero bit in the *RBSP*.

3.44
reserved
value of a particular *syntax element* (3.53) for future use by ISO/IEC and not to be used in *bitstreams* (3.2) conforming to this document, but could be used in *bitstreams* conforming to future revised editions of this document

3.45
reserved_zeros
value of a particular *syntax element* (3.53) for future use by ISO/IEC and not to be used in *bitstreams* (3.2) conforming to this document, but could be used in bitstreams conforming to future revised editions of this document

Note 1 to entry: In this document, the value of any *reserved_zeros* bit is zero.

3.46
residual
difference between a prediction of a sample or data element and a reference of that same sample or data element

3.47
residual plane
collection of *residuals* (3.46)

3.48
run length encoding
RLE
method for encoding a sequence of values in which consecutive occurrences of the same value are represented as a single value together with its number of occurrences

3.49
sample aspect ratio
the ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the *luma* (3.34) sample array in a *picture* (3.40), which is specified for assisting the display process (not specified in this Specification) and expressed as h:v, where h is the horizontal width and v is the vertical height, in arbitrary units of spatial distance

3.50
source
video material or some of its attributes before encoding

3.51
start code prefix
unique sequence of three *bytes* (3.4) equal to 0 x 000001 embedded in the *byte stream* (3.6) as a prefix to each *NAL unit* (3.35)

Note 1 to entry: The location of a *start code prefix* can be used by a *decoder* (3.20) to identify the beginning of a new *NAL unit* and the end of a previous *NAL unit*. Emulation of *start code prefixes* is prevented within *NAL units* by the inclusion of *emulation prevention bytes* (3.23).

3.52
string of data bits
SODB
sequence of some number of bits representing *syntax elements* (3.53) present within a *raw byte sequence payload* (3.42) prior to the *raw byte sequence payload stop bit* (3.43), and within which the left-most bit is considered to be the first and most significant bit, and the right-most bit is considered to be the last and least significant bit

3.53
syntax element
element of data represented in the *bitstream* (3.2)

3.54
syntax structure
zero or more *syntax elements* (3.53) present together in the *bitstream* (3.2) in a specified order

3.55
tile
rectangular region of *TUs* (3.58) within a particular *picture* (3.40)

3.56
transform
part of the *decoding process* (3.22) by which a *block* (3.3) of *transform coefficients* (3.57) is converted to a *block* of spatial-domain values

3.57
transform coefficient
scalar quantity, considered to be in a transformed domain, that is associated with a particular index in an *inverse transform* (3.31) part of the *decoding process* (3.22)

3.58
transform unit
TU
MxN *block* (3.3) of samples resulting from a *transform* (3.56) in the *decoding process* (3.22)

3.59
unspecified
value of a particular *syntax element* (3.53) with no specified meaning in this document and that will not have a specified meaning in any future revised editions of this document

3.60
video coding layer NAL unit
VCL NAL unit
NAL units (3.35) that have reserved values of NalUnitType that are classified as VCL NAL units in this document

4 Abbreviated terms

CLVS	coded layer-wise video sequence
CPB	coded picture buffer
CPBB	coded picture buffer of the base
CPBL	coded picture buffer LCEVC
CVS	coded video sequence
DPBB	decoded picture buffer of the base
DUT	decoder under test
HBD	hypothetical base decoder
HDM	hypothetical demuxer
HRD	hypothetical reference decoder
HSS	hypothetical stream scheduler
I	intra
LCEVC	low complexity enhancement video coding
LSB	least significant bit
MSB	most significant bit
SEI	supplemental enhancement information
VUI	video usability information

5 Conventions

5.1 General

NOTE The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are defined more precisely, and additional operations are defined, such as exponentiation and real-valued division. Numbering and counting conventions generally begin from 0, e.g., “the first” is equivalent to the 0-th, “the second” is equivalent to the 1-th, etc.

5.2 Arithmetic operators

+	addition
–	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
x^y	exponentiation; specifies x to the power of y In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.

/	integer division with truncation of the result toward zero For example, 7 / 4 and −7 / −4 are truncated to 1 and −7 / 4 and 7 / −4 are truncated to −1.
÷	division in mathematical equations where no truncation or rounding is intended
	division in mathematical equations where no truncation or rounding is intended
$\sum_{i=x}^{y} f(i)$	summation of f(i) with i taking all integer values from x up to and including y
x % y	Modulus Remainder of x divided by y, defined only for integers x and y with x >= 0 and y > 0.

5.3 Logical operators

x && y	Boolean logical “and” of x and y
x || y	Boolean logical “or” of x and y
!	Boolean logical “not”
x ? y : z	if x is TRUE or not equal to 0, evaluates to the value of y; otherwise, evaluates to the value of z

5.4 Relational operators

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equal to
!=	not equal to

When a relational operator is applied to a syntax element or variable that has been assigned the value “na” (not applicable), the value “na” is treated as a distinct value for the syntax element or variable. The value “na” is considered not to be equal to any other value.

5.5 Bit-wise operators

& bit-wise “and”

When operating on integer arguments, operates on a two’s complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

| bit-wise “or”

When operating on integer arguments, operates on a two’s complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

^ bit-wise “exclusive or”

When operating on integer arguments, operates on a two’s complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

x >> y arithmetic right shift of a two’s complement integer representation of x by y binary digits

This function is defined only for non-negative integer values of y. Bits shifted into the most significant bits (MSBs) as a result of the right shift have a value equal to the MSB of x prior to the shift operation.

x << y arithmetic left shift of a two’s complement integer representation of x by y binary digits

This function is defined only for non-negative integer values of y. Bits shifted into the least significant bits (LSBs) as a result of the left shift have a value equal to 0.

5.6 Assignment operators

= assignment operator

++ increment, i.e., *x*++ is equivalent to *x* = *x* + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation

– – decrement, i.e., *x*– – is equivalent to *x* = *x* – 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation

+= increment by amount specified, i.e., x += 3 is equivalent to x = x + 3, and x += (–3) is equivalent to x = x + (–3)

–= decrement by amount specified, i.e., x –= 3 is equivalent to x = x – 3, and x –= (–3) is equivalent to x = x – (–3)

5.7 Range notation

x = y...z x takes on integer values starting from y to z, inclusive, with x, y, and z being integer numbers and z being greater than y

x = y to z x takes on integer values starting from y to z, inclusive, with x, y, and z being integer numbers and z being greater than y