INTERNATIONAL STANDARD

ISO/IEC 23091-2

First edition
2019-07

Information technology — Coding-independent code points —

Part 2: **Video**

Technologies de l'information — Points de code indépendants du codage —
Partie 2: Vidéo

Reference number
ISO/IEC 23091-2:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 ISO documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO 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).

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.

This document was prepared by Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. The technically identical text is published as ITU-T H.273 (12/2016).

Together with ISO/IEC 23091-1 and ISO/IEC 23091-3, this first edition of ISO/IEC 23091-2 cancels and replaces ISO/IEC 23001-8:2016, which has been technically revised.

A list of all parts in the ISO/IEC 23091 series can be found on the ISO website.

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.

Introduction

In a number of specifications, there is a need to identify some characteristics of video (or still image) media content that are logically independent of the compression format. These characteristics may include, for example, aspects that relate to the sourcing or presentation, or the role of the video (or still image) media component. These characteristics have typically been documented by fields that take an encoded value or item selected from an enumerated list, herein called code points.

These code points are typically defined in the specification of compression formats to document these characteristics of the media. In past practices, the definition of these fields has been copied from document to document, sometimes with new values being added in later documents (and sometimes with later amendments specified to add new entries to existing documents).

This past practice has raised a number of issues, including the following:

a) A lack of a formal way to avoid conflicting assignments being made in different documents.

b) Having additional values defined in later specifications that may be practically used with older compression formats, but without clear formal applicability of these new values to older documents.

c) Any update or correction of code point semantics can incur significant effort to update all documents in which the code point is specified, instead of enabling a single central specification to apply across different referencing specifications.

d) The choice of reference for other specifications (such as container or delivery formats) not being obvious; wherein a formal reference to a compression format document appears to favour that one format over others, and also appears to preclude definitions defined in other compression format specifications.

e) Burdensome maintenance needs to ensure that a reference to material defined in a compression format specification is maintained appropriately over different revisions of the referenced format specification, as the content of a compression format specification may change over time and is ordinarily not intended as a point of reference for defining such code points.

This document provides a central definition of such code points for video and image applications to address these issues. This document can be used to provide universal descriptions to assist interpretation of video and image signals following decoding, or to describe the properties of these signals before they are encoded.

Information technology — Coding-independent code points —

Part 2: Video

1 Scope

This document defines various code points and fields that establish properties of a video (or still image) representation and are independent of the compression encoding and bit rate. These properties could describe the appropriate interpretation of decoded data or could, similarly, describe the characteristics of such a signal before the signal is compressed by an encoder that is suitable for compressing such an input signal.

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 11664-1, *Colorimetry — Part 1: CIE standard colorimetric observers*

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:

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

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

3.1
chroma
sample array or single sample 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.

3.2
component
array or single sample from one of the three arrays [*luma* (3.3) and two *chroma* (3.1)] that compose a *picture* 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

3.3
luma
sample array or single sample is 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.

3.4
picture
array of *luma* (3.3) samples in monochrome format or array of *luma* samples and two corresponding arrays of *chroma* (3.1) samples in 4:2:0, 4:2:2, and 4:4:4 colour format

3.5
reserved
values of a particular code point that are for future use by ITU-T | ISO/IEC and shall not be used in identifiers conforming to this version of this document, but which may be used in a manner yet to be specified in some future extensions of this document by ITU-T | ISO/IEC

3.6
unspecified
values of a particular code point that have no specified meaning in this version of this document and will not have a specified meaning in the future as an integral part of future versions of this document

4 Abbreviated terms

LSB	least significant bit
MSB	most significant bit

5 Conventions

NOTE The mathematical operators used in this document are similar to those used in the C programming language. However, integer division and arithmetic shift operations are specifically defined. Numbering and counting conventions generally begin from 0.

5.1 Arithmetic operators

+	addition
−	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
x^y	exponentiation, x to the power of y (in other contexts, such notation may be used for super-scripting 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 formulae where no truncation or rounding is intended

$\frac{x}{y}$ division in mathematical formulae 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

5.2 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 (defined only for non-negative integer values of y; bits shifted into the 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 (defined only for non-negative integer values of y; bits shifted into the LSBs as a result of the left shift have a value equal to 0)

5.3 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 given, i.e., x += 3 is equivalent to x = x + 3, and x += (–3) is equivalent to x = x + (–3)

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

5.4 Relational, logical, and other operators

= = equality operator

!= not equal to operator

!x logical negation "not"

>	larger than operator
<	smaller than operator
>=	larger than or equal to operator
<=	smaller than or equal to operator
&&	conditional/logical "and" operator, performs a logical "and" of its Boolean operators, but only evaluates the second operand when necessary
||	conditional/logical "or" operator, performs a logical "or" of its Boolean operators, but only evaluates the second operand when necessary
a ? b : c	ternary conditional, if condition a is true, then the result is equal to b; otherwise the result is equal to c

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

5.5 Mathematical functions

$$\mathrm{Abs}(x)=\begin{cases} x & ; \quad x >= 0 \\ -x & ; \quad x < 0 \end{cases} \tag{1}$$

Ceil(x) the smallest integer greater than or equal to x. (2)

$$\mathrm{Clip1}_Y(x) = \mathrm{Clip3}(0, (1 << \mathrm{BitDepth}_Y) - 1, x), \tag{3}$$

where $\mathrm{BitDepth}_Y$ is the representation bit depth of the corresponding luma colour component signal.

$$\mathrm{Clip1}_C(x) = \mathrm{Clip3}(0, (1 << \mathrm{BitDepth}_C) - 1, x), \tag{4}$$

where $\mathrm{BitDepth}_C$ is the representation bit depth of the corresponding chroma colour component signal C. In general, $\mathrm{BitDepth}_C$ may be distinct for different chroma colour components signals C – e.g. for C corresponding to Cb or Cr.

$$\mathrm{Clip3}(x, y, z)=\begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} \tag{5}$$

Floor(x) the largest integer less than or equal to x. (6)

Ln(x) the natural logarithm of x. (7)

Log10(x) the base-10 logarithm of x. (8)

Round(x) = Sign(x) * Floor(Abs(x) + 0.5). (9)

$$\mathrm{Sign}(x)=\begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases} \tag{10}$$

$$\mathrm{Sqrt}(x) = \sqrt{x}\,. \tag{11}$$

5.6 Order of operations

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

— Operations of a higher precedence are evaluated before any operation of a lower precedence.

— Operations of the same precedence are evaluated sequentially from left to right.

Table 1 specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

Operations (with operands x, y, and z)
"x++", "x– –"
"!x", "–x" (as a unary prefix operator)
"x^y"
"x * y", "x / y", "x ÷ y", "$\frac{x}{y}$", "x % y"
"x + y", "x – y" (as a two-argument operator), "$\sum_{i=x}^{y} f(i)$"
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x = = y", "x != y"
"x & y"
"x | y"
"x && y"
"x | | y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x –= y"

6 Specified code points

This clause identifies the code points defined in this document, as listed in Table 2 with cross-references to the subclause in which each is specified.