INTERNATIONAL STANDARD

ISO/IEC 23002-2

First edition
2008-06-01

Information technology — MPEG video technologies —

Part 2:
Fixed-point 8×8 inverse discrete cosine transform and discrete cosine transform

Technologies de l'information — Technologies vidéo MPEG —

Partie 2: Transformation de cosinus discret inverse 8×8 *point fixe et transformation de cosinus discret*

Reference number
ISO/IEC 23002-2:2008(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 23002-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23002 consists of the following parts, under the general title *Information technology — MPEG video technologies*:

— *Part 1: Accuracy requirements for implementation of integer-output* 8×8 *inverse discrete cosine transform*

— *Part 2: Fixed-point* 8×8 *inverse discrete cosine transform and discrete cosine transform*

— *Part 3: Representation of auxiliary video and supplemental information*

Introduction

A number of visual-coding-related specifications (see Ref. [1] to [6] in the Bibliography) include a requirement for decoders to implement an integer-output 8×8 inverse discrete cosine transform (IDCT) for the generation of inverse-transformed samples with a nominal range from 0 to $(2^B)–1$, or sample differences with a nominal range from $–2^B$ to $(2^B)–1$, for some integer number of bits B, where B is greater than or equal to 8.

This part of ISO/IEC 23002 provides the following benefits.

- It provides an example IDCT (and also an example forward DCT) approximation method to ease the implementation community in their design of decoders and encoders.

- It can help to ensure that decoders are implemented in full conformance with relevant video and image coding specifications (such as those listed in Refs. [2] to [6] in the Bibliography). Decoders that are designed to use the specified method will be assured to conform to the IDCT conformance requirements of the relevant image and video coding standards.

- It specifies a single deterministic result as the output of an image or video decoding process, such that video analysis tools can operate on decoded video with precisely predictable results. This provides the assurance to source material providers, for example, of exactly what results will be obtained from a video target detector, segmentation mask operator, or other classification, analysis, or post-processing process that operates on the decoded video. Such certainty is not achievable without a deterministically-specified decoding result.

- It can improve the quality of delivered video and image representations, as encoders designed to target their encoding process for the IDCT approximation specified herein can be assured that the decoding process will be free of encoder-decoder drift error on all decoders that conform to this part of ISO/IEC 23002.

Information technology — MPEG video technologies —

Part 2:
Fixed-point 8×8 inverse discrete cosine transform and discrete cosine transform

1 Scope

This part of ISO/IEC 23002 specifies a particular implementation of an integer-output 8×8 IDCT that fully conforms to the accuracy requirements specified in ISO/IEC 23002-1 (see Ref. [7] in the Bibliography) and additionally meets or exceeds all accuracy requirements specified for IDCT precision in a number of international video coding standards (see Ref. [2] to [6] in the Bibliography). It additionally provides a (non-normative) specification of an integer-output 8×8 forward DCT based on the same factorization structure.

2 Terms and definitions

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

2.1
array
indexed collection of **scalar values**

2.2
discrete cosine transform
DCT
transformation that produces an output **matrix** of **transform coefficients** from an input matrix of **samples** with results similar to those produced by the process specified in Annex A

2.3
fixed-point
numerical representation that has a fixed number of digits after the radix point

NOTE Equivalent to an integer representation herein, for which the number of digits after the radix point is zero.

2.4
forward discrete cosine transform
forward DCT
discrete cosine transform

NOTE Inclusion of the word "forward" in the term is used when a contrast with the concept of an **inverse discrete cosine transform** is intended to be emphasized.

2.5
inverse discrete cosine transform
IDCT
transformation that produces an output **matrix** of **samples** from an input matrix of **transform coefficients** with results similar to those produced by the process specified in Clause 5

2.6
matrix
array with a two-dimensional index

2.7
sample
entry in a **matrix** that is the input of a **discrete cosine transform** or the output of an **inverse discrete cosine transform**

2.8
scalar value
integer or a real-valued number

2.9
transform coefficient
entry in a **matrix** that is the output of a **forward discrete cosine transform** or the input of an **inverse discrete cosine transform**

3 Abbreviated terms

For purposes of this document, the following listed abbreviations apply.

DC: referring to the entry in a matrix of transform coefficients associated with the index coordinate pair [0][0].

DCT: discrete cosine transform

IDCT: inverse DCT

LSB: least significant bit

MSB: most significant bit

4 Conventions

4.1 Arithmetic operators

The following arithmetic operators are defined.

+	Addition
–	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	Multiplication
x^y	Exponentiation. Specifies x to the power of y.
$\frac{x}{y}$	Division. Specifies division of x by y, producing a real-valued number result.

When an order of operations is not indicated explicitly by use of parenthesis, the following rules apply:

– exponentiation is considered to take place before multiplication and division;

– multiplication and division operations are considered to take place before addition and subtraction;

– multiplication and division operations in sequence are evaluated sequentially from left to right;

– addition and subtraction operations in sequence are evaluated sequentially from left to right.

4.2 Logical operators

A logical operator is defined as follows:

$c\ ?\ d : e$ If the condition c is TRUE, evaluates to the value of d; otherwise, evaluates to the value of e.

4.3 Relational operators

The following relational operators are defined.

$==$	Equal to
$<$	Less than
$>$	Greater than
$<=$	Less than or equal to
$>=$	Greater than or equal to

The relational operators return the value TRUE if the expressed condition is fulfilled and otherwise return the value FALSE.

4.4 Bit-wise operators

The following bit-wise operators are defined.

$x >> y$ Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for positive integer values of y. Bits shifted into the MSBs as a result of the right shift shall 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 positive integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

4.5 Assignment operators

An assignment operator is defined as follows.

$c = d$ Assigns the variable c equal to the value of d.

The assignment operator is considered to take effect after all other operations in an equation have been completed.

4.6 Range notation

The following notation is used to specify a range of values.

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

4.7 Index notation for arrays

Square parentheses are used to indicate the indexing of arrays. For example, s[5] denotes the entry at index 5 in the array s.

When a matrix represents samples in a spatial sampling grid or is depicted as a spatial grid of numbers in text, the first (left-most) component of the index is considered to be the vertical component of the index and the second (right-most) component of the index is considered to be the horizontal component of the index. For example, entry f[3][5] in a matrix f of samples would denote the entry at vertical position 3 and horizontal position 5. When the matrix represents transform coefficients, the first (left-most) component of the index is considered to represent a vertical frequency index and the second (right-most) component of the index is considered to represent a horizontal frequency index. For example, entry $F[3][5]$ in a matrix F of transform coefficients would denote the entry with vertical frequency index 3 and horizontal frequency index 5.

5 Specification of fixed point 8×8 IDCT

5.1 General

The process defined in this subclause is a transformation of an 8×8 input matrix of transform coefficients $F[v][u]$ with $u = 0..7$ and $v = 0..7$ into an 8×8 output matrix of samples $f[y][x]$ with $x = 0..7$ and $y = 0..7$.

Each entry in the input matrix of transform coefficients $F[v][u]$ with $u = 0..7$ and $v = 0..7$ shall be an integer in the range of $-2^{(B+3)}$ to $2^{(B+3)}-1$, inclusive. Unless specified otherwise by a referencing specification, the value of B shall be equal to 8.

The computation of output samples $f[y][x]$ using transform coefficients $F[v][u]$ shall be performed in a manner mathematically equivalent to the process described below.

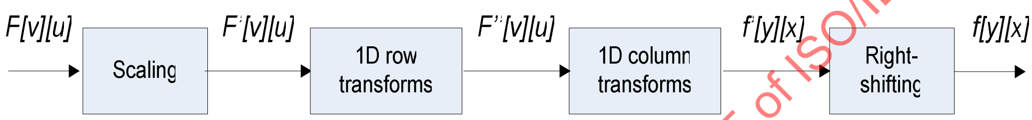

Figure 1 – Structure of 8×8 IDCT process

Figure 1 illustrates the overall transform process. All matrices in this figure contain integer values. Input coefficients $F[v][u]$ are first pre-scaled, producing a matrix of scaled coefficients $F'[v][u]$. Then a sequence of one-dimensional transforms is applied to all rows $F'[v][\cdot]$ with $v = 0..7$, producing a matrix of intermediate values $F''[v][u]$. Then, another sequence of one-dimensional transforms is applied to all columns $F''[\cdot][u]$ with $u = 0..7$, producing a matrix of scaled sample values $f'[y][x]$. Then the scaled sample values are right-shifted, producing output values $f[y][x]$.

5.2 Scaling

Scaling of transform coefficients $F[v][u]$ is performed in a sequence of steps as follows:

$$F'[v][u] = F[v][u] * S[v][u]; \quad v = 0..7,\ u = 0..7; \tag{1}$$

$$F'[0][0] = F'[0][0] + 4096; \tag{2}$$

where $S[v][u]$ are elements of the following symmetric matrix:

$$S=\begin{bmatrix} 1024 & 1138 & 1730 & 1609 & 1024 & 1609 & 1730 & 1138 \\ 1138 & 1264 & 1922 & 1788 & 1138 & 1788 & 1922 & 1264 \\ 1730 & 1922 & 2923 & 2718 & 1730 & 2718 & 2923 & 1922 \\ 1609 & 1788 & 2718 & 2528 & 1609 & 2528 & 2718 & 1788 \\ 1024 & 1138 & 1730 & 1609 & 1024 & 1609 & 1730 & 1138 \\ 1609 & 1788 & 2718 & 2528 & 1609 & 2528 & 2718 & 1788 \\ 1730 & 1922 & 2923 & 2718 & 1730 & 2718 & 2923 & 1922 \\ 1138 & 1264 & 1922 & 1788 & 1138 & 1788 & 1922 & 1264 \end{bmatrix}. \tag{3}$$

An input range constraint is that all results of Equations 1 and 2 are in the range of $-2^{(B+14)}$ to $2^{(B+14)}-1$, inclusive.

5.3 Computation of one-dimensional transforms

Input to one-dimensional transforms can be either a row $F'[v][\cdot]$ in a matrix of scaled transform coefficients $F'[v][u]$, or a column $F''[\cdot][u]$ in a matrix of intermediate values $F''[v][u]$. In both cases, the specified input locations are also used for storing output of the one-dimensional transform.

The computation of one-dimensional transforms includes generation of products by pairs of transform factors and combines them together using the transform factorization.

5.3.1 Computation of products by the first pair of transform factors: *mul_1(y, z)*

This process receives a pair of variables (y, z) and performs the following sequence of steps.

$$y2 = (y >> 3) - (y >> 7); \quad (4)$$
$$y3 = y2 - (y >> 11); \quad (5)$$
$$z = y2 + (y3 >> 1); \quad (6)$$
$$y = y - y2; \quad (7)$$

where $y2$ and $y3$ are intermediate variables that can subsequently be discarded.

An input range constraint is that all results of Equations 4 through 7 are in the range of $-2^{(B+17)}$ to $2^{(B+17)}-1$, inclusive.

5.3.2 Computation of products by the second pair of transform factors: *mul_2(y, z)*

This process receives a pair of variables (y, z) and performs the following sequence of steps.

$$y2 = (y >> 9) - y; \quad (8)$$
$$z = y >> 1; \quad (9)$$
$$y = (y2 >> 2) - y2; \quad (10)$$

where $y2$ is an intermediate variable that can subsequently be discarded.

An input range constraint is that all results of Equations 8 through 10 are in the range of $-2^{(B+17)}$ to $2^{(B+17)}-1$, inclusive.

5.3.3 Computation of products by the third pair of transform factors: *mul_3(y, z)*

This process receives a pair of variables (y, z) and performs the following sequence of steps.

$$y2 = y + (y >> 5); \quad (11)$$
$$y3 = y2 >> 2; \quad (12)$$
$$y = y3 + (y >> 4); \quad (13)$$
$$z = y2 - y3; \quad (14)$$

where $y2$ and $y3$ are intermediate variables that can subsequently be discarded.

An input range constraint is that all results of Equations 11 through 14 are in the range of $-2^{(B+17)}$ to $2^{(B+17)}-1$, inclusive.

5.3.4 Computation of one-dimensional transform butterfly

This process receives either a row or a column of coefficients, denoted by $G[u]$, $u = 0..7$, and performs the following sequence of steps:

$$x1 = G[1]; \quad (15)$$
$$x3 = G[3]; \quad (16)$$
$$x5 = G[5]; \quad (17)$$
$$x7 = G[7]; \quad (18)$$

$xa = x1 + x7;$ (19)
$xb = x1 - x7;$ (20)

$x1 = xa + x3;$ (21)
$x3 = xa - x3;$ (22)
$x7 = xb + x5;$ (23)
$x5 = xb - x5;$ (24)

$mul\_1(x3, xa);$ // as defined in subclause 5.3.1 (25)
$mul\_1(x5, xb);$ // as defined in subclause 5.3.1 (26)
$x3 = x3 - xb;$ (27)
$x5 = x5 + xa;$ (28)

$mul\_2(x1, xa);$ // as defined in subclause 5.3.2 (29)
$mul\_2(x7, xb);$ // as defined in subclause 5.3.2 (30)
$x1 = x1 + xb;$ (31)
$x7 = x7 - xa;$ (32)

$x0 = G[0];$ (33)
$x2 = G[2];$ (34)
$x4 = G[4];$ (35)
$x6 = G[6];$ (36)

$mul\_3(x2, xa);$ // as defined in subclause 5.3.3 (37)
$mul\_3(x6, xb);$ // as defined in subclause 5.3.3 (38)
$x2 = x2 - xb;$ (39)
$x6 = x6 + xa;$ (40)

$xa = x0 + x4;$ (41)
$xb = x0 - x4;$ (42)

$x0 = xa + x6;$ (43)
$x6 = xa - x6;$ (44)
$x4 = xb + x2;$ (45)
$x2 = xb - x2;$ (46)

$G[0] = x0 + x1;$ (47)
$G[1] = x4 + x5;$ (48)
$G[2] = x2 + x3;$ (49)
$G[3] = x6 + x7;$ (50)
$G[4] = x6 - x7;$ (51)
$G[5] = x2 - x3;$ (52)
$G[6] = x4 - x5;$ (53)
$G[7] = x0 - x1;$ (54)

where $x0, x1, x2, x3, x4, x5, x6, x7, xa$, and xb are intermediate variables that can subsequently be discarded.

An input range constraint is that all results of Equations 15 through 54 are in the range of $-2^{(B+17)}$ of $2^{(B+17)}-1$, inclusive.

5.4 Right-shifting

Right-shifting of scaled samples $f'[y][x]$ is performed as follows:

$$f[y][x] = f'[y][x] >> 13; \qquad y = 0..7,\ x = 0..7; \tag{55}$$

Annex A
(informative)

Fixed point 8×8 forward DCT

A.1 General

This annex does not specify any requirements for conformance to this part of ISO/IEC 23002. It describes a particular fixed-point 8×8 forward DCT approximation as an example to aid in the design of implementations of products that include forward DCT approximation.

The process defined in this annex is a transformation of an 8×8 matrix of input samples $f[y][x]$ with $x = 0..7$ and $y = 0..7$, into an 8×8 matrix of transform coefficients $F[v][u]$ with $u = 0..7$ and $v = 0..7$.

The values of input samples are assumed to be in the range of -2^{B} to $2^{B}-1$, inclusive. The values of the output coefficients $F[v][u]$ produced by this transform will then be in the range of $-2^{(B+3)}$ to $2^{(B+3)}-1$, inclusive. Unless specified otherwise by a referencing specification, the value of B is considered to be equal to 8.

The computation of output coefficients $F[v][u]$ is performed in a manner mathematically equivalent to the process described below.

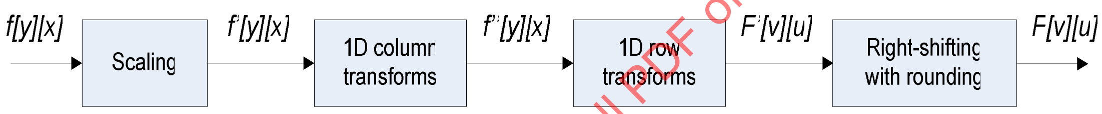

Figure 2 – Structure of 8×8 DCT process

Figure 2 illustrates the overall transform process. All matrices in this figure contain integer values. Input samples $f[y][x]$ are first pre-scaled, resulting in a matrix of scaled samples $f'[y][x]$. Then a sequence of eight one-dimensional transforms is applied to all columns $f'[\cdot][x]$ with $x = 0..7$, producing a matrix of intermediate values $f''[y][x]$. Then, another sequence of eight one-dimensional transforms is applied to all rows $f''[y][\cdot]$ with $y = 0..7$, producing a matrix of scaled transform coefficients $F'[v][u]$. Then, transform coefficients $F'[v][u]$ are right-shifted and rounded, producing a matrix of transform coefficients $F[v][u]$.

A.2 Scaling

Scaling of input samples $f[y][x]$ is performed as follows:

$$f'[y][x] = f[y][x] << 7; \qquad y = 0..7,\ x = 0..7; \qquad (56)$$

A.2.1 Computation of one-dimensional forward transforms

This process receives either a column or a row of coefficients, denoted by $G[u]$, $u = 0..7$, and performs the following sequence of steps:

$$x0 = G[0] + G[7]; \qquad (57)$$
$$x1 = G[0] - G[7]; \qquad (58)$$
$$x4 = G[1] + G[6]; \qquad (59)$$
$$x5 = G[1] - G[6]; \qquad (60)$$
$$x2 = G[2] + G[5]; \qquad (61)$$
$$x3 = G[2] - G[5]; \qquad (62)$$
$$x6 = G[3] + G[4]; \qquad (63)$$
$$x7 = G[3] - G[4]; \qquad (64)$$

$mul\_1(x3, xa);$ // as defined in subclause 5.3.1 (65)
$mul\_1(x5, xb);$ // as defined in subclause 5.3.1 (66)

$$x3 = x3 + xb; \qquad (67)$$
$$x5 = x5 - xa; \qquad (68)$$

$mul\_2(x1, xa);$ // as defined in subclause 5.3.2 (69)
$mul\_2(x7, xb);$ // as defined in subclause 5.3.2 (70)
$x1 = x1 - xb;$ (71)
$x7 = x7 + xa;$ (72)

$xa = x1 + x3;$ (73)
$x3 = x1 - x3;$ (74)
$xb = x7 + x5;$ (75)
$x5 = x7 - x5;$ (76)

$x1 = xa + xb;$ (77)
$x7 = xa - xb;$ (78)

$xa = x0 + x6;$ (79)
$x6 = x0 - x6;$ (80)
$xb = x4 + x2;$ (81)
$x2 = x4 - x2;$ (82)

$x0 = xa + xb;$ (83)
$x4 = xa - xb;$ (84)

$mul\_3(x2, xa);$ // as defined in subclause 5.3.3 (85)
$mul\_3(x6, xb);$ // as defined in subclause 5.3.3 (86)
$x2 = xb + x2;$ (87)
$x6 = x6 - xa;$ (88)

$G[0] = x0;$ (89)
$G[1] = x1;$ (90)
$G[2] = x2;$ (91)
$G[3] = x3;$ (92)
$G[4] = x4;$ (93)
$G[5] = x5;$ (94)
$G[6] = x6;$ (95)
$G[7] = x7;$ (96)

where $x0, x1, x2, x3, x4, x5, x6, x7, xa,$ and xb are intermediate variables that can subsequently be discarded. When the values of the input samples $f[y][x]$ are in the assumed range (the range of -2^B to 2^B-1, inclusive), all values computed as the result of Equations 57 to 96 are in the range of $-2^{(B+13)}$ to $2^{(B+13)}-1$, inclusive.

This process uses computations of products by pairs of transform factors as defined in subclauses 5.3.1, 5.3.2, and 5.3.3.

A.2.2 Right-shifting with rounding

Right-shifting of transform coefficients $F'[v][u]$, if desired, can be performed as follows:

$$F[v][u] = (F'[v][u] * S[v][u] + (1 << 19) - ((F'[v][u] >= 0) ? 0 : 1)) >> 20; \quad v = 0..7,\ u = 0..7; \tag{97}$$

where $S[v][u]$ is an entry in the matrix of transform scale factors specified in Equation 3 of subclause 5.2.

When the values of the input samples $f[y][x]$ are in the assumed range of -2^B to 2^B-1, inclusive, the intermediate values computed as a result of operations prior to the right shift in Equation 97 are in the range of $-2^{(B+23)}$ to $2^{(B+23)}-1$, inclusive, and the values computed as a result of Equation 97 are in the range of $-2^{(B+3)}$ to $2^{(B+3)}-1$, inclusive.

When using two's complement arithmetic and a register size of *n* bits for representation of $F'[v][u]$, subtraction of $((F'[v][u] >= 0) ? 0 : 1)$ is equivalent to addition of $(F'[v][u] >> (n-1))$.

