

Information technology — JPEG 2000 image coding system —

Part 2: Extensions

ICS 35.040

National foreword

This British Standard reproduces verbatim ISO/IEC 15444-2:2004 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of picture, audio, multimedia and hypermedia information, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC title page, pages ii to ix, a blank page, pages 1 to 321 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 23 June 2004

© BSI 23 June 2004

BS ISO/IEC 15444-2:2004

INTERNATIONAL **ISO/IEC**
STANDARD **15444-2**

First edition
2004-05-15

**Information technology — JPEG 2000
image coding system: Extensions**

*Technologies de l'information — Système de codage d'image
JPEG 2000: Extensions*



Reference number
ISO/IEC 15444-2:2004(E)

AVA Documents
Document Sharing Hub

CONTENTS

	<i>Page</i>
1 Scope	1
2 References	1
2.1 Identical Recommendations International Standards	1
2.2 Additional references	2
3 Definitions	3
4 Abbreviations	4
5 Symbols	4
6 General description	5
6.1 Extensions specified by this Recommendation International Standard	5
6.1.1 Syntax	5
6.1.2 Variable DC offset	5
6.1.3 Variable scalar quantization	5
6.1.4 Trellis coded quantization	5
6.1.5 Visual masking	5
6.1.6 Arbitrary decomposition	5
6.1.7 Arbitrary wavelet transformation	6
6.1.8 Single sample overlap discrete wavelet transformations	6
6.1.9 Multiple component transformations	6
6.1.10 Non-linear transformation	6
6.1.11 Region of interest	6
6.1.12 File format	6
6.1.13 Metadata definitions	6
6.2 Relation between extensions	7
Annex A – Compressed data syntax, extension	8
A.1 Extended capabilities	8
A.2 Extensions to ITU-T Rec. T.800 ISO/IEC 15444-1 marker segment parameters	8
A.2.1 Image and tile size (SIZ), extended	9
A.2.2 Start of tile-part (SOT) extended	9
A.2.3 Coding style (COD, COC), extended	10
A.2.4 Quantization (QCD, QCC), extended	12
A.2.5 Region of interest (RGN), extended	14
A.3 Extended marker segments	15
A.3.1 Variable DC offset (DCO)	15
A.3.2 Visual masking (VMS)	16
A.3.3 Downsampling factor styles (DFS)	17
A.3.4 Arbitrary decomposition styles (ADS)	18
A.3.5 Arbitrary transformation kernels (ATK)	19
A.3.6 Component bit depth definition (CBD)	21
A.3.7 Multiple component transformation definition (MCT)	22
A.3.8 Multiple component transform collection (MCC)	23
A.3.9 Multiple component transform ordering (MCO)	26
A.3.10 Non-linearity point transformation (NLT)	27
A.3.11 Quantization default, precinct (QPD)	29
A.3.12 Quantization precinct component (QPC)	30
Annex B – Variable DC offset, extension	33
B.1 Variable DC offset flow	33
B.2 Inverse DC offset	33
B.3 Forward DC offset (informative)	33
Annex C – Variable scalar quantization, extension	35
C.1 Variable scalar quantization	35
C.2 Variable scalar dequantization for irreversible filters	35
C.3 Variable scalar quantization for irreversible filters (informative)	36

	<i>Page</i>
Annex D – Trellis coded quantization extensions	37
D.1 Introduction to TCQ.....	37
D.2 Sequence definition	39
D.3 Forward TCQ quantization (informative)	39
D.4 Inverse quantization (normative)	41
D.4.1 Full TCQ dequantization.....	41
D.4.2 Approximate dequantization	43
D.5 Lagrangian rate allocation (informative).....	44
Annex E – Visual masking, extensions.....	49
E.1 Introduction to visual masking (informative).....	49
E.2 Point-wise extended non-linearity (informative).....	49
E.3 Decoding with visual masking	51
E.4 Encoding with visual masking (informative)	52
E.5 Setting parameters (informative).....	52
E.6 Compatibility with other technologies (informative)	53
Annex F – Arbitrary decomposition of tile-components, extensions.....	54
F.1 Wavelet sub-bands	54
F.1.1 Tier 1: Number of decomposition levels.....	54
F.1.2 Tier 2: Resolution formation	54
F.1.3 Tier 3: Sub-level decompositions.....	54
F.1.4 Tier 4: Horizontal and vertical splits to variable sub-level depths	54
F.1.5 Complete sub-band notation	55
F.1.6 HorOrient, VerOrient and PrimeOrient sub-band operators	55
F.2 Equation, text and decomposition updates	55
F.2.1 Updates to N_{LL} references	56
F.2.2 Context updates.....	56
F.2.3 Extension to ITU-T Rec. T.800 ISO/IEC 15444-1 Equation B-14.....	56
F.2.4 Remaining updates	57
F.2.5 Updates to decomposition structure	61
F.3 Inverse discrete wavelet transformation for general decompositions.....	66
F.3.1 Modified IDWT procedure	67
F.3.2 Modified 2D_SR procedure	68
F.3.3 Modified 2D_INTERLEAVE procedure	69
F.4 Forward discrete wavelet transformation for general decompositions (informative).....	73
F.4.1 Modified FDWT procedure.....	73
F.4.2 Modified 2D_SD procedure.....	74
F.4.3 Modified 2D_DEINTERLEAVE procedure	75
Annex G – Whole-sample symmetric transformation of images, extensions	80
G.1 Wavelet transformation parameters, definitions and normalizations	80
G.2 Whole-sample symmetric (WS) wavelet transformations reconstruction	80
G.2.1 Normalization of WS wavelet transformations	80
G.2.2 One-dimensional sub-band reconstruction procedure for WS wavelet transformations.....	81
G.3 Whole-sample symmetric (WS) wavelet transformation decomposition (informative).....	84
G.3.1 The 1D_SD_WS procedure (informative)	84
G.3.2 The 1D_FILTD_WS one-dimensional decomposition procedure (informative)	84
G.4 Examples of WS wavelet transformations (informative)	86
G.4.1 Reversible WS wavelet transformations ($WT\_Typ = REV$) (informative)	86
G.4.2 Irreversible WS wavelet transformations ($WT\_Typ = IRR$) (informative).....	87
Annex H – Transformation of images using arbitrary wavelet transformations.....	89
H.1 Wavelet transformation parameters and normalizations	89
H.1.1 Normalization of ARB wavelet transformations.....	89
H.1.2 Compatibility of ARB and WS wavelet transformations	89

	<i>Page</i>
H.2 Arbitrary (ARB) wavelet transformation reconstruction procedures	90
H.2.1 The extended 1D_SR_ARB procedure	90
H.2.2 The 1D_SCALER procedure	91
H.2.3 The 1D_STEPR procedure	92
H.2.4 Extension procedures	92
H.2.5 One-dimensional reconstruction update filtering procedures	94
H.3 Arbitrary (ARB) wavelet transformation decomposition procedures (informative)	95
H.3.1 Extended 1D_SD_ARB procedure (informative)	95
H.3.2 The 1D_STEPR procedure (informative)	96
H.3.3 Extension procedures (informative)	97
H.3.4 One-dimensional decomposition update procedures (informative)	97
H.3.5 1D_SCALED procedure (informative)	98
H.4 Examples of ARB wavelet transformations (informative)	99
H.4.1 Examples of arbitrary wavelet transformations (Filt_Cat = ARB) (informative)	99
H.4.2 Example of a structure for lifting implementation of half-sample symmetric wavelet transformations (informative)	101
Annex I – Single sample overlap discrete wavelet transform, extensions	103
I.1 Introduction to single sample overlapping	103
I.2 The code-block anchor points (CBAP) extension	103
I.2.1 Division of resolution levels in precincts	103
I.2.2 Division of the sub-bands into codeblocks	104
I.2.3 Resolution level-position-component-layer progression	105
I.2.4 Position-component-resolution level-layer progression	106
I.2.5 Component-position-resolution level-layer progression	106
I.3 The SSO extension	107
I.3.1 Single sample overlap inverse discrete wavelet transformation (SSO-IDWT)	107
I.3.2 Single sample overlap forward discrete wavelet transformation (informative)	110
I.3.3 Selection of single sample overlap parameters (informative)	112
I.3.4 SSO examples (informative)	113
I.4 The TSSO extension	115
I.4.1 Signalling for the TSSO	115
I.4.2 Partitioning of the image into single-sample overlapping tiles	115
I.4.3 Reconstruction of images samples from reconstructed tiles	116
I.5 Combining the SSO and TSSO extensions (informative)	116
Annex J – Multiple component transformations, extension	117
J.1 Introduction to multiple component transformation concepts	117
J.2 Overview of inverse processing	117
J.2.1 Inverse multiple component transformation (MCO_TRANSFORM)	118
J.2.2 Multiple component transformation stage (MCC_TRANS)	119
J.2.3 Transformation component collection (CC_TRANS)	121
J.3 Transformations	123
J.3.1 Array-based transforms	124
J.3.2 Wavelet-based transformation	132
Annex K – Non-linear transformation	134
K.1 Signalling the use of the non-linear transformations	134
K.1.1 Decoded component reconstruction	134
K.1.2 Bit depth and interaction with the multiple component transformation	134
K.1.3 Marker interpretation	135
K.2 Non-linear transformation specifications	135
K.2.1 Gamma-style non-linearity	135
K.2.2 LUT-style reverse non-linearity transformation	137
Annex L – Region of interest coding and extraction, extensions	139
L.1 Decoding of ROI	139
L.2 Description of the Scaling based method	139
L.2.1 Encoding with ROI (informative)	139

	<i>Page</i>
L.3 Region of interest mask generation	140
L.3.1 Rectangular mask generation on the reference grid	141
L.3.2 Elliptic mask generation on the reference grid	141
L.3.3 Region of Interest mask generation of whole-sample symmetric filter banks	142
L.3.4 Region of Interest mask generation of arbitrary optional filter banks	142
L.3.5 Fast generation of a rectangular mask (informative)	143
L.4 Remarks on region of interest coding	145
L.4.1 Usage together with Maxshift method described in ITU-T T.800 ISO/IEC 15444-1	145
L.4.2 Multi-component remark (informative)	145
L.4.3 Implementation Precision remark (informative)	145
Annex M – JPX extended file format syntax	146
M.1 File format scope	146
M.2 Introduction to JPX	146
M.2.1 File identification	146
M.2.2 File organization	146
M.2.3 Greyscale/Colour/multi-component specification	147
M.2.4 Specification of opacity information	147
M.2.5 Metadata	147
M.2.6 Storage of a codestream within JPX	147
M.2.7 Combining multiple codestreams	147
M.3 Greyscale/Colour/Palette/multi-component specification architecture	148
M.3.1 Extensions to the Colour Specification box header	148
M.3.2 Extensions to the Enumerated method	148
M.3.3 Any ICC method	148
M.3.4 Vendor Colour method	148
M.3.5 Palettized colour	149
M.3.6 Using multiple methods	149
M.3.7 Interactions with the decorrelating multiple component transformation	149
M.4 Fragmenting the codestream between one or more files	149
M.5 Combining multiple codestreams	151
M.5.1 Mapping codestreams to compositing layers	151
M.5.2 Sharing header and metadata information between codestreams and compositing layers	152
M.5.3 Composition	152
M.6 Using reader requirements masks to determine how a file can be used	154
M.6.1 Types of expressions	155
M.6.2 Expression representation	155
M.6.3 Testing an Implementation against Requirements Expressions	160
M.7 Extensions to the JPX file format and the registration of extensions	161
M.7.1 Registration elements	161
M.7.2 Differentiation between publication and registration	162
M.7.3 Items which can be extended by registration	162
M.7.4 Published items	164
M.7.5 Registration process	165
M.7.6 Timeframes for the registration process	165
M.8 Differences from the JP2 binary definition	165
M.9 Conformance	166
M.9.1 Interpretation of JPX data structures	166
M.9.2 Support for JPX feature set	166
M.10 Key to graphical descriptions (informative)	168
M.11 Defined boxes	168
M.11.1 Reader Requirements box	171
M.11.2 Data Reference box	174
M.11.3 Fragment Table box (superbox)	174
M.11.4 Cross-Reference box	175
M.11.5 JP2 Header box (superbox)	176
M.11.6 Codestream Header box (superbox)	178
M.11.7 Compositing Layer Header box (superbox)	179
M.11.8 Contiguous Codestream box	191
M.11.9 Media Data box	191

	<i>Page</i>
M.11.10 Composition box (superbox).....	191
M.11.11 Association box (superbox).....	194
M.11.12 Number List box	196
M.11.13 Label box	197
M.11.14 Binary Filter box	197
M.11.15 Desired Reproductions box (superbox).....	198
M.11.16 ROI Description box	199
M.11.17 Digital Signature box	200
M.11.18 XML box	202
M.11.19 MPEG-7 Binary box	203
M.11.20 Free box	203
M.12 Dealing with unknown boxes.....	203
M.13 Using the JPX file format in conjunction with other multi-media standards (informative)	203
Annex N – JPX file format extended metadata definition and syntax	204
N.1 Introduction to extended metadata	204
N.2 Additional references for extended metadata	204
N.3 Scope of metadata definitions	204
N.3.1 Image Creation metadata.....	205
N.3.2 Content Description metadata	205
N.3.3 History metadata	205
N.3.4 Intellectual Property Rights metadata	205
N.3.5 Fundamental metadata types and elements	205
N.4 Metadata syntax	205
N.4.1 Metadata schema definition language	205
N.4.2 Namespace	205
N.4.3 Document type definition information.....	206
N.4.4 XML Schema information	206
N.5 Defined boxes	206
N.5.1 Image Creation metadata box.....	206
N.5.2 Content Description metadata box	207
N.5.3 History box.....	207
N.5.4 Intellectual Property Rights box.....	208
N.5.5 Image Identifier box	208
N.6 Metadata definitions.....	208
N.6.1 Image Creation metadata.....	209
N.6.2 Content Description metadata	220
N.6.3 History metadata	226
N.6.4 Intellectual Property Rights metadata	229
N.6.5 Image Identifier metadata	236
N.7 Fundamental type and element definitions.....	237
N.7.1 Defined types	237
N.7.2 Defined attributes.....	254
N.7.3 Defined elements.....	255
N.8 JPX extended metadata document type definition	255
N.9 JPX extended metadata XML Schema	265
Annex O – Examples and guidelines, extensions	281
O.1 Arbitrary decomposition examples	281
O.2 Odd Tile Low Pass First (OTLPF) convention	303
O.2.1 Example one (even tile sizes).....	304
O.2.2 Example two (odd tile sizes).....	304
O.2.3 Example three (TSSO/OTLPF).....	304
O.3 Multiple component collection example	305
O.3.1 Array-based multiple component transform example	305
O.3.2 Unitary decorrelation transformation factorization and reversible decorrelation transformation	311
O.3.3 Dependency transformation, irreversible and reversible.....	315
O.4 Background to enhancement of quantization	317

	<i>Page</i>
Bibliography	317
Index	319
Patent statement	321

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 15444-2:2004 was prepared jointly by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.801.

ISO/IEC 15444 consists of the following parts, under the general title *Information technology — JPEG 2000 image coding system*:

- *Part 1: Core coding system*
- *Part 2: Extensions*
- *Part 3: Motion JPEG 2000*
- *Part 4: Conformance testing*
- *Part 5: Reference software*
- *Part 6: Compound image file format*
- *Part 9: Interactivity tools, APIs and protocols*
- *Part 12: ISO base media file format*

The following parts are under preparation:

- *Part 8: Secure JPEG 2000*
- *Part 10: Extensions for three-dimensional data and floating point data*
- *Part 11: Wireless JPEG 2000*

**INTERNATIONAL STANDARD
ITU-T RECOMMENDATION**

Information technology – JPEG 2000 image coding system: Extensions

1 Scope

This Recommendation | International Standard defines a set of lossless (bit-preserving) and lossy compression methods for coding continuous-tone, bi-level, grey-scale, colour digital still images, or multi-component images.

This Recommendation | International Standard:

- specifies extended decoding processes for converting compressed image data to reconstructed image data;
- specifies an extended codestream syntax containing information for interpreting the compressed image data;
- specifies an extended file format;
- specifies a container to store image metadata;
- defines a standard set of image metadata;
- provides guidance on extended encoding processes for converting source image data to compressed image data;
- provides guidance on how to implement these processes in practice.

2 References

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*.
- ITU-T Recommendation T.82 (1993) | ISO/IEC 11544:1993, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*.
- ITU-T Recommendation T.83 (1994) | ISO/IEC 10918-2:1995, *Information technology – Digital compression and coding of continuous-tone still images: Compliance testing*.
- ITU-T Recommendation T.84 (1996) | ISO/IEC 10918-3:1997, *Information technology – Digital compression and coding of continuous-tone still images: Extensions*.
- ITU-T Recommendation T.84 (1996)/Amd.1 (1999) | ISO/IEC 10918-3:1997/Amd.1:1999, *Information technology – Digital compression and coding of continuous-tone still images: Extensions – Amendment 1: Provisions to allow registration of new compression types and versions in the SPIFF header*.
- ITU-T Recommendation T.86 (1998) | ISO/IEC 10918-4:1999, *Information technology – Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF Compression types and Registration Authorities (REGAUT)*.
- ITU-T Recommendation T.87 (1998) | ISO/IEC 14495-1:2000, *Information technology – Lossless and near-lossless compression of continuous-tone still images – Baseline*.

- ITU-T Recommendation T.88 (2000) | ISO/IEC 14492:2001, *Information technology – Lossy/lossless coding of bi-level images*.
- ITU-T Recommendation T.800 (2002) | ISO/IEC 15444-1:2003, *Information technology – JPEG 2000 image coding system: Core coding system*.

2.2 Additional references

- ITU-T Recommendation T.42 (1996), *Continuous-tone colour representation method for facsimile*.
- ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
- ISO 8601:2000, *Data elements and interchange formats – Information interchange – Representation of dates and times*.
- ISO 3166-1:1997, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*.
- ISO 3166-2:1998, *Codes for the representation of names of countries and their subdivisions – Part 2: Country subdivision code*.
- ISO/IEC 11578:1996, *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)*.
- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*.
- ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*.
- ISO/IEC 15938, *MPEG-7*.
- ISO 10126-2:1991, *Banking – Procedures for message encipherment (wholesale) – Part 2: DEA algorithm*.
- IEEE Standard 754-1985 R1990, *IEEE Standard for Binary Floating-Point Arithmetic*.
- IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm*.
- IETF RFC 1766 (1995), *Tags for the Identification of Languages*.
- IETF RFC 2279 (1998), *UTF-8, A transformation format of ISO 10646*.
- IETF RFC 2630 (1999), *Cryptographic Message Syntax*.
- IETF RFC 2313 (1998), *PKCS #1: RSA Encryption, Version 1.5*.
- International Color Consortium, *ICC profile format specification*. ICC.1.
- IEC 61966-2-1:1999, *Multimedia systems and equipment – Colour measurement and management: Part 2-1: Colour management – Default RGB colour space – sRGB, plus its Amendment 1: 2003*.
- Digital Imaging Group, *Flashpix digital image file format*. Version 1.0.1, 10 July 1997.
- PIMA 7666, *Photography-Electronics still picture imaging-Reference Output Medium Metric RGB Color encoding: ROMM-RGB*.
- PIMA 7667:2001, *Photography-Electronics still picture imaging-Extended sRGB color encoding e-sRGB*.
- Federal Information Processing Standard Publication (FIPS PUB) 186-2, *Digital Signature Standard (DSS)*. <<http://www.itl.nist.gov/fipspubs/fip186-2.pdf>>
- ANSI X9.30.2-1997, *Public Key Cryptography for the Financial Services Industry – Part 2: The Secure Hash Algorithm (SHA-1)*. <<http://www.itl.nist.gov/fipspubs/fip180-1.htm>>
- W3C, *Extensible Markup Language (XML 1.0)*, 2nd edition Rec-xml-2000106, <<http://www.w3.org/TR/REC-xml>>.
- W3C, *Namespaces in XML*, Rec-xml-names-19990114, <<http://www.w3.org/TR/1999/REC-xml-names>>.
- W3C, *XML Schema Part 1: Structures*, Rec-xmlschema-1-20010502, <<http://www.w3.org/TR/xmlschema-1>>.
- W3C, *XML Schema Part 2: Datatypes*, Rec-xmlschema-2-20010502, <<http://www.w3.org/TR/xmlschema-2>>.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. The definitions defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 clause 3 also apply to this Recommendation | International Standard except for the terms decomposition level, sub-band and resolution which are redefined in this clause.

- 3.1 attribute:** An XML construct that is a name-value pair extending or qualifying the meaning of an element.
- 3.2 cell:** An optional subdivision of a tile used for low-memory encoding and decoding.
- 3.3 component:** Compressed data from the codestream representing a single set of two-dimensional data.
- 3.4 component collection:** A subset of intermediate components used as inputs to a multiple component transformation stage, and a subset of intermediate components obtained as outputs from a multiple component transformation stage. The subset's constituent components may occur in an arbitrary order, i.e., permuted with respect to their order of appearance in the set of input or output intermediate components.
- 3.5 component reconstruction arrays:** A general term that refers to any of the following; decorrelation transformation array, dependency transformation array or offset array.
- 3.6 compositing:** The act of combining two compositing layers into a single, non-redundant set of image channels.
- 3.7 compositing layer:** A set of non-redundant channels drawn from one or more codestreams that shall be treated as a group. The set of compositing layers within the JPX file may then be combined by compositing or animation instructions to form a rendered result. For example, one layer may be a simple RGBA codestream. Another layer may consist of R, G and B channels generated by the application of a palette to one component from codestream 1, and an opacity channel directly extracted from codestream 2.
- 3.8 deadzone:** The interval within which all sub-band coefficients are quantized to 0.
- 3.9 decomposition level:** A collection of sub-bands where each coefficient has the same spatial impact or span with respect to the original samples. These include the LL, LH, HL, HH, LX, HX, XL, and XH sub-band splits out of decomposition sublevels.
- 3.10 decomposition sub-level:** A collection of sub-bands that result from splits of a sub-band from a lower decomposition sub-level or splits of either LL, LX or XL sub-bands from a higher decomposition level.
- 3.11 decorrelation transformation array:** An array of coefficients that maps the input components of a component collection to the output components of the collection via a multiple component decorrelation transformation.
- 3.12 dependency transformation array:** An array of coefficients that maps the input components of a component collection to the output components of the collection via a multiple component dependency transformation.
- 3.13 element:** An XML construct that consists of a start tag and an end tag with data enclosed within.
- 3.14 HX sub-band:** The sub-band obtained by forward horizontal high-pass analysis filtering and no vertical analysis filtering. This sub-band contributes to reconstruction with inverse horizontal high-pass synthesis filtering and no vertical synthesis filtering.
- 3.15 intermediate component:** A single two-dimensional array of data involved in a stage of a multiple component transformation.
- 3.16 JPX baseline:** A specific subset of the features of the JPX file format.
- 3.17 JPX baseline reader:** An application that correctly interprets all files that conform to the definition of a JPX baseline file.
- 3.18 JPX file:** The name of file in the file format described in this Recommendation | International Standard. Structurally, a JPX file is a contiguous sequence of boxes.
- 3.19 LX sub-band:** The sub-band obtained by forward horizontal low-pass analysis filtering and no vertical analysis filtering. This sub-band contributes to reconstruction with inverse horizontal low-pass synthesis filtering and no vertical synthesis filtering.
- 3.20 metadata:** Additional data associated with the image data beyond the image data.
- 3.21 namespace:** A collection of names, identified by a URI, that allows XML documents of different sources to use the same element names within a single document to avoid element name conflicts.
- 3.22 offset array:** An array of coefficients containing offsets which are added to intermediate components during multiple component transformation of a component collection.

3.23 reconstructed image component: The set of output intermediate components from the final transformation stage in the inverse multiple component transformation process.

3.24 rendered result: The result generated by combining the compositing layers in the JPX file, either by composition or animation.

3.25 resolution: The spatial relation of samples to a physical space. In this Recommendation | International Standard, the decomposition levels of the wavelet transformation create resolutions that differ by powers of two in either just horizontal, just vertical or both horizontal and vertical directions. The last (highest) decomposition level includes either an LL, LX or XL sub-band which is considered to be a lower resolution. Therefore, there is one more resolution level than decomposition levels.

3.26 sub-band: A group of transformation coefficients resulting from the sequence of low-pass and high-pass filtering operations, either just horizontally, just vertically or both horizontally and vertically.

3.27 spatially reconstructed component: A component which has been extracted from the codestream and passed through the decoding and inverse wavelet transformation process as specified by this Recommendation | International Standard. The set of spatially reconstructed components is the set of input components to the first transformation stage in the inverse multiple component transformation process.

3.28 transformation stage: A set of component collections and associated multiple component transformations.

3.29 visual masking: Visual masking is a mechanism where artefacts are masked by the image acting as a background signal.

3.30 XH sub-band: The sub-band obtained by no forward horizontal analysis filtering and vertical high-pass analysis filtering. This sub-band contributes to reconstruction with vertical high-pass synthesis filtering and no inverse horizontal synthesis filtering.

3.31 XL sub-band: The sub-band obtained by no forward horizontal analysis filtering and vertical low-pass analysis filtering. This sub-band contributes to reconstruction with vertical low-pass synthesis filtering and no inverse horizontal synthesis filtering.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply. The abbreviations defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 clause 4 also apply to this Recommendation | International Standard.

CCITT	International Telegraph and Telephone Consultative Committee, now ITU-T
DPI	Dots per inch
IPR	Intellectual Property Rights
UUID	Universal Unique Identifier

5 Symbols

For the purposes of this Recommendation | International Standard, the following symbols apply. The symbols defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 clause 4 also apply to this Recommendation | International Standard.

ADS	Arbitrary decomposition styles marker
ATK	Arbitrary transformation kernels marker
CBD	Component bit depth definition marker
DCO	Variable DC offset marker
DFS	Downsample factor styles marker
MCC	Multiple component collection transformation marker
MCO	Multiple component transformation ordering marker
MCT	Multiple component transformation definition marker
NLT	Non-linearity point transformation marker
VMS	Visual masking marker

6 General description

The purpose of this clause is to give an overview of this Recommendation | International Standard. Terms defined in previous clauses in this Recommendation | International Standard will also be introduced. (Terms defined in clauses 3 and 4 in ITU-T Rec. T.800 | ISO/IEC 15444-1 continue to apply in this Recommendation | International Standard.)

This Recommendation | International Standard defines a set of lossless (bit-preserving) and lossy compression methods for coding continuous-tone, bi-level, grey-scale, colour digital still images, or multi-component images. This set of methods extends the elements in the core coding system described in ITU-T Rec. T.800 | ISO/IEC 15444-1. Extensions which pertain to encoding and decoding are defined as procedures which may be used in combination with the encoding and decoding processes described in ITU-T Rec. T.800 | ISO/IEC 15444-1. Each encoding or decoding extension shall only be used in combination with particular coding processes and only in accordance with the requirements set forth herein. These extensions are backward compatible in the sense that decoders which implement these extensions will also support configuration subsets that are currently defined by ITU-T Rec. T.800 | ISO/IEC 15444-1. This Recommendation | International Standard also defines extensions to the compressed data format, i.e., interchange format and the abbreviated formats.

6.1 Extensions specified by this Recommendation | International Standard

The following extensions are specified in this Recommendation | International Standard.

6.1.1 Syntax

An extension of the code stream syntax is described in Annex A. This extension provides all the codestream signalling in this Recommendation | International Standard. Further, it anticipates signalling needed for future specifications that include this Recommendation | International Standard as a normative reference. In addition to the codestream syntax defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, the following capabilities are supported: variable DC offset, variable scalar quantization, trellis coded quantization, visual masking, arbitrary decomposition, arbitrary transformation kernels, single sample overlap, multiple component transformations, non-linear transformation, arbitrary regions of interest. These extended markers conform to the same rules as the syntax in ITU-T Rec. T.800 | ISO/IEC 15444-1.

6.1.2 Variable DC offset

An extension which provides for variable DC offset is described in Annex B. Variable DC offset may be used to generate a better data distribution for input to the ICT or RCT multi component transformation, defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, and/or the wavelet transformation. Images with very skewed sample distributions may benefit from a non-default DC offset.

6.1.3 Variable scalar quantization

An extension that provides for variable scalar quantization is described in Annex C. This extension allows smaller or larger deadzones to be used with the scalar quantizer. This technique may improve visual appearance of low level texture.

6.1.4 Trellis coded quantization

An extension of the quantization is described in Annex D. This extension provides for trellis coded quantization (TCQ). The TCQ algorithm applies spatial-varying scalar quantization to its input sequence by choosing one of four scalar quantizers for each sample. Quantizer indices from supersets of these quantizers along with quantizer transitions in the form of a trellis provide all information necessary to reconstruct TCQ encoded wavelet coefficients.

6.1.5 Visual masking

An extension which provides for visual masking is described in Annex E. Visual masking is a mechanism where artefacts are masked by the image acting as a background signal. The main goal is to improve the image quality, especially for displays. The first effect of this technique is to improve the image quality, where the improvement becomes greater as the image becomes more complex. The second main effect of this technique is that for a given fixed bit-rate, the image quality is more robust against variations in image complexity. This is accomplished at the encoder via an extended non-linearity interposed between the transformation stage and the quantization stage.

6.1.6 Arbitrary decomposition

An extension providing for arbitrary decomposition of the tile component is described in Annex F. This extension can control the bandpass extent of wavelet sub-bands and thus provide control over the decorrelation process in order to

tune compression performance. This extension also allows for transcoding of other wavelet based compression algorithms into codestreams of this Recommendation | International Standard.

6.1.7 Arbitrary wavelet transformation

Extensions that provide for transformation of image tile components using user defined wavelet filters are described in Annexes G and H. Annex G describes whole sample filters while Annex H describes arbitrary filters.

6.1.8 Single sample overlap discrete wavelet transformations

Extensions providing for block based wavelet transformations are described in Annex I. These extensions consist of one method for tile based wavelet transformation without tiling artefacts and one method for cell-based wavelet transformation.

6.1.9 Multiple component transformations

An extension which provides for multiple component transformations is described in Annex J. This extension specifies two types of multiple component transformations:

- 1) A specification of a multiple component transformation which uses linear transformations of bands to reduce the correlation of each band. This is similar to the most common colour transformations.
- 2) A specification of a wavelet transformation along the component direction.

6.1.10 Non-linear transformation

Annex K specifies two non-linear point transformations that are used after decoding processes and inverse multiple component transformations to map reconstructed values back to their proper range. These transformations, gamma and look-up table (LUT) style non-linearities, may be employed by encoders prior to multiple component transformation and encoding to increase compression efficiency. A common usage of these transformations might be to perceptually flatten a scanner or sensor with a linear response, from 12 bits to 8 bits precision prior to compression.

6.1.11 Region of interest

An extension which provides for Region of interest coding using the scaling based method is described in Annex L. The scaling based method provides for having different scaling values for different regions of interest. The extension also specifies how to generate the masks in the wavelet domain that describe the set of wavelet coefficient belonging to each Region of interest.

6.1.12 File format

An extension of the file format is described in Annex M. This extension provides for the exchange of compressed image files between application environments. This extension is an optional file format, called JPX, that applications may choose to use to contain JPEG 2000 compressed image data. JPX is an extension to the JP2 file format defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex I. The format:

- 1) specifies a binary container for both image and metadata;
- 2) specifies a mechanism to indicate image properties, such as the tone-scale or colour space of the image;
- 3) specifies a mechanism by which readers may recognize the existence of intellectual property rights information in the file;
- 4) specifies a mechanism by which metadata (including vendor specific information) can be included in files specified by this Recommendation | International Standard;
- 5) specifies a mechanism by which multiple codestreams can be combined into a single work, by methods such as compositing and animation.

6.1.13 Metadata definitions

Metadata definitions are described in N. Metadata is additional information that is associated with the primary data (the image). In the context of this Specification, it is "additional data linked with the image data beyond the pixels which define the image." Metadata, to be most valuable for the owner(s) and user(s) of an image, needs to be consistently maintained throughout the image lifecycle. In today's environment of image editing applications, rapid transmission via the Internet, and high quality photographic printers, the lifecycle of a digital image may be very long as well as complex.

6.2 Relation between extensions

The relations, at decoder side, between the extensions listed above are given in Figure 6-1. Technologies described in ITU-T Rec. T.800 | ISO/IEC 15444-1 are indicated in the boxes.

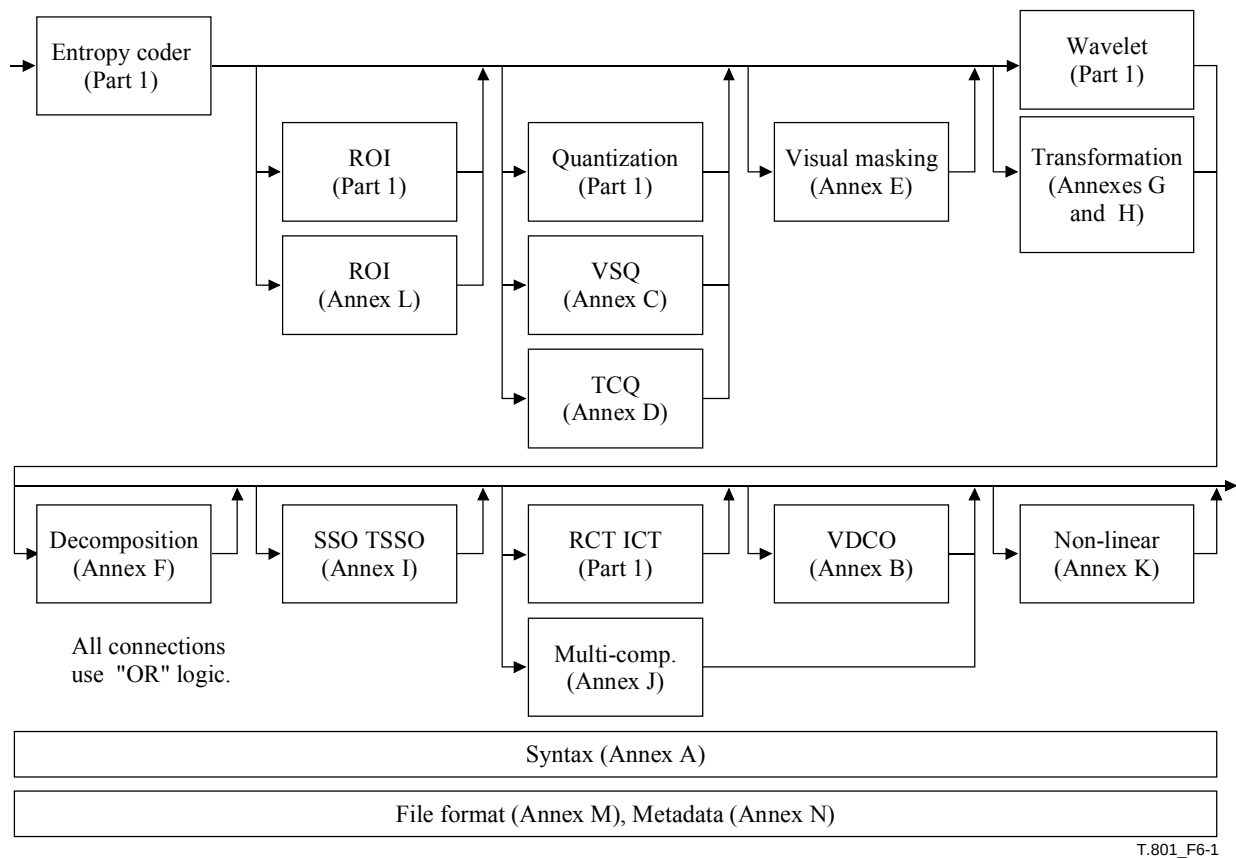


Figure 6-1 – Decoder block diagram

Annex A

Compressed data syntax, extension

(This annex forms an integral part of this Recommendation | International Standard)

In this annex and all of its subclauses, the flow charts and tables are normative only in the sense that they are defining an output that alternative implementations shall duplicate.

This annex specifies the marker and marker syntax extensions to the ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A syntax. These markers provide all the codestream signalling in this Recommendation | International Standard. Further, it anticipates signalling needed for future specifications that include this Recommendation | International Standard as a normative reference.

All positive (unsigned) integer values of parameters are placed in the codestream as unsigned integers. All other (signed) integers are expressed in two's complement. Unless otherwise indicated, all values are in big endian order.

Some marker segment parameters have values defined with bits. In some cases there are bits, denoted by "x," that do not correspond to any parameter and are unused by all parameters. The codestream shall have zero value bits for those cases. The decoder shall ignore these bits.

A.1 Extended capabilities

The syntax in this annex supports the extensions in this Recommendation | International Standard. These marker segments conform to the same rules as the syntax in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A. The addition of parameter values to some marker segments in ITU-T Rec. T.800 | ISO/IEC 15444-1 and the addition of new marker segments signals the information specific to the extensions in this Recommendation | International Standard. In every marker segment the first two bytes after the marker shall be an unsigned value that denotes the length in bytes of the marker segment parameters (including the two bytes of this length parameter but not the two bytes of the marker itself). When a marker segment that is not specified in this Recommendation | International Standard or in ITU-T Rec. T.800 | ISO/IEC 15444-1 is encountered in a codestream, the decoder shall use the length parameter to discard the marker segment. Table A.1 shows the marker segments affected by this Recommendation | International Standard.

Table A.1 – Syntax support for extensions

Extension	Extended ITU-T Rec. T.800 ISO/IEC 15444-1 marker segments	New marker segments
All extensions	SIZ	–
Variable DC offset	–	DCO
Variable scalar quantization	QCD, QCC, SOT	QPD, QPC
Trellis coded quantization	QCD, QCC, SOT	QPD, QPC
Visual masking	–	VMS
Single sample offset transform	SIZ, COD, COC	–
Arbitrary decomposition styles	COD, COC	DFS, ADS
Arbitrary transformation kernels	COD, COC	ATK
Multiple component transform	COD	CBD, MCT, MCC, MCO
Non-linearity point transformation	–	NLT
Arbitrary shaped region of interest	RGN	–

A.2 Extensions to ITU-T Rec. T.800 | ISO/IEC 15444-1 marker segment parameters

This clause describes the extensions to marker segments defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A.

A.2.1 Image and tile size (SIZ), extended

The capability parameter Rsiz of ITU-T Rec. T.800 | ISO/IEC 15444-1 denotes the necessity or usefulness of extensions of this Recommendation | International Standard for decoding the codestream. Table A.2 is used to define this parameter.

Table A.2 – Capability Rsiz parameter, extended

Value (bits)		Capability
MSB	LSB	
0000	0000 0000 0000	Capabilities specified in ITU-T Rec. T.800 ISO/IEC 15444-1 only
1000	xxxx xxxx xxxx	At least one of the extended capabilities specified in this Recommendation International Standard is present
1000	xxx0 xxxx xxx1	Variable DC offset capability is required to decode this codestream ^{a) b)}
1000	xxxx xxxx xx1x	Variable scalar quantization capability is required to decode this codestream ^{a)}
1000	xxxx xxxx x1xx	Trellis coded quantization capability is useful to decode this codestream ^{c)}
1000	xxxx xxxx 1xxx	Visual masking capability is useful to decode this codestream ^{c)}
1000	xxxx xxx1 xxxx	Single sample overlap capability is required to decode this codestream ^{a)}
1000	xxxx xx1x xxxx	Arbitrary decomposition style capability is required to decode this codestream ^{a)}
1000	xxxx x1xx xxxx	Arbitrary transformation kernel capability is required to decode this codestream ^{a)}
1000	xxxx 1xxx xxxx	Whole sample symmetric transformation kernel capability is required to decode this codestream ^{a)}
1000	xxx1 xxxx xxxx	Multiple component transformation capability is required to decode this codestream ^{a)}
1000	xx1x xxxx xxxx	Non-linear point transformation capability is useful to decode this codestream ^{c)}
1000	x1xx xxxx xxxx	Arbitrary shaped region of interest capability is required to decode this codestream ^{a)}
1000	1xxx xxxx xxxx	Precinct-dependent quantization is required to decode this codestream ^{a)}
		All other values reserved
a)		"Required to decode" implies that no useful data or image can be reconstructed without the use of this capability.
b)		Shall not be used with the multiple component transformation.
c)		"Useful to decode" implies that use of this capability would improve the quality of the reconstructed data or image; however, the data or image may be decoded without its use.

A.2.2 Start of tile-part (SOT) extended

If Rsiz indicates that the precinct-dependent quantization capability is used, then the SOT marker segment from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A, is extended to allow 1-65535 tile-parts. Table A.3 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.5 and Table A.4 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.6.

Table A.3 – Start of tile-part parameter values, extended

Parameter	Size (bits)	Values
SOT	16	0xFF90
Lsot	16	10
Isot	16	0-65 534
Psot	32	0, or $14 \cdot (2^{32} - 1)$
TPsot	16	0-65535
TNsot	16	Table A.4

Table A.4 – Number of tile-parts, TNsot, parameter value, extended

Value	Number of tile-parts
0	Number of tile-parts of this tile in the codestream is not defined in this header
1-65535	Number of tile-parts of this tile in the codestream

A.2.3 Coding style (COD, COC), extended

Geometric manipulation is enabled with two bits in the Scod parameter shown in Table A.5.

If the Rsiz field of the SIZ marker segment indicates that the multiple component transformations are used, then Table A.8 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.17.

If the Rsiz field of the SIZ marker segment indicates that the arbitrary transformation kernels are used, then Table A.10 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.20.

If the Rsiz field of the SIZ marker segment indicates that the single sample overlap transformation capability is necessary, then an extra 8 bit field is added to ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.13 after the transformation field as shown in Table A.7. The SSO values are found in Table A.11.

If the Rsiz field of the SIZ marker segment indicates that the arbitrary decomposition styles are used then the maximum number of decomposition levels field definitions are found in Table A.9 rather than in ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.13. This is shown in Table A.7.

Table A.5 – Coding style parameter values for the Scod parameter

Values (bits) MSB LSB	Coding style
xxxx xxx0	Entropy coder, precincts with PP _x = 15 and PP _y = 15
xxxx xxx1	Entropy coder with precincts defined below
xxxx xx0x	No SOP marker segments used
xxxx xx1x	SOP marker segments may be used
xxxx x0xx	No EPH marker used
xxxx x1xx	EPH marker shall be used
xxxx 0xxx xxxx 1xxx	Offset in the horizontal dimension, $z_x = 0$ (CBAP) Offset in the horizontal dimension, $z_x = 1$
xxx0 xxxx xxx1 xxxx	Offset in the vertical dimension, $z_y = 0$ (CBAP) Offset in the vertical dimension, $z_y = 1$
	All other values reserved

Table A.6 – Coding style parameter values of the SGcod parameter

Parameters (in order)	Size (bits)	Values	Meaning of SGcod values
Progression order	8	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.16	Progression order
Number of layers	16	1-65535	Number of layers
Multiple component transformation	8	Table A.8	Multiple component transformation usage

Table A.7 – Coding style parameter values of the SPcod and SPcoc parameters, extended

Parameters (in order)	Size (bits)	Values (bits) MSB LSB	Meaning of SPcod values
Maximum number of decomposition levels	8	Table A.9	Decomposition mapping and levels
Code-block width	8	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.18	Code-block width exponent offset value, xcb
Code-block height	8	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.18	Code-block height exponent offset value, ycb
Code-block style	8	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.19	Style of the code-block coding passes
Transform	8	Table A.10	Wavelet transformation used
SSO overlap	16	Table A.11	SSO overlap values
Precinct size	variable	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.21	If Scod or Scoc = xxxx xxx0, this parameter is not present, otherwise this indicates precinct width and height. The first parameter (8 bits) corresponds to the N_L LL, N_L LX, or N_L XL sub-band. Each successive parameter corresponds to each successive resolution in order.

Table A.8 – Multiple component transformation for the SGcod parameters

Values (bits) MSB LSB	Multiple component transformation type
0000 0000	No multiple component transformation specified.
0000 0001	Component transformation used on components 0, 1, 2 for coding efficiency. Irreversible component transformation used with irreversible filters. Reversible component transformation used with reversible filters.
0000 0x10	Array-based multiple component transformation is used. May be combined with wavelet-based multiple component transform.
0000 01x0	Wavelet-based multiple component transformation is used. May be combined with array-based multiple component transform.
	All other values reserved.

Table A.9 – Decomposition for the SPcod and SPcoc parameters, extended

Values (bits) MSB LSB	Decomposition type
0000 0000 to 0010 0000	Number of levels of wavelet decomposition, dyadic decomposition, N_L (Zero implies no transform.).
1000 0001 to 1111 1111	If in the main header: Downsampling factor style index value (1 to 127). (See A.3.3.) If in tile-part header: Arbitrary decomposition style index value (1 to 127). (See A.3.4.)
	All other values reserved

Table A.10 – Transformation for the SPcod and SPcoc parameters, extended

Values (bits) MSB LSB	Transformation type
0000 0000	ITU-T Rec. T.800 ISO/IEC 15444-1 9-7 irreversible wavelet transform.
0000 0001	ITU-T Rec. T.800 ISO/IEC 15444-1 5-3 reversible wavelet transform.
0000 0010 to 1111 1111	Arbitrary transformation kernel definition index value (2-255). Definitions are found in the appropriate ATK marker segment (see A.3.5).

Table A.11 – SSO parameters, extended

Values (bits) MSB LSB	SSO size (see Annex I)
0xxx xxxx xxxx xxxx 1xxx xxxx xxxx xxxx	SSO not used SSO used
x0xx xxxx xxxx xxxx x1xx xxxx xxxx xxxx	TSSO not used for every tile in the image (shall be the same in every COD marker segment) TSSO used for every tile in the image (shall be the same in every COD marker segment)
xx0x xxxx xxxx xxxx xx1x xxxx xxxx xxxx	Vovlp = 0 (shall be the same in every COD marker segment) Vovlp = 1 (shall be the same in every COD marker segment)
xxx0 xxxx xxxx xxxx xxx1 xxxx xxxx xxxx	Hovlp = 0 (shall be the same in every COD marker segment) Hovlp = 1 (shall be the same in every COD marker segment)
xxxx 0xxx xxxx xxxx x1xx 1xxx xxxx xxxx	TBDWT is not used (shall be the same in every COD marker segment) TBDWT is used (shall be the same in every COD marker segment)
xxxx xxxx xxxx 0000 to xxxx xxxx xxxx 1111	Cell width exponent value, $XC = 2^{\text{value}}$
xxxx xxxx 0000 xxxx to xxxx xxxx 1111 xxxx	Cell height exponent value, $YC = 2^{\text{value}}$
	All other values reserved

A.2.4 Quantization (QCD, QCC), extended

If Rsiz indicates that the variable scalar quantization (see Annex C) capability is used, then the deadzone adjustment is signalled in modified QCD and QCC marker segments from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A. If Rsiz indicates that trellis coded quantization is used, then these values are also signalled via the extended QCD and QCC marker segments from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A. This shall only be used with irreversible transformations.

Table A.12 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.28, and Table A.13 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.30.

Table A.12 – Quantization default values for the Sqcd, Sqcc, Sqpd, and Sqpc parameters, extended

Values (bits) MSB LSB	Quantization style	SPqxx size (bits)	SPqxx usage
xxx0 0000	No quantization	8	ITU-T Rec. T.800 ISO/IEC 15444-1 Annex A
xxx0 0001	Scalar derived (values signalled for N_{LL} sub-band only). Use ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-5.	16	ITU-T Rec. T.800 ISO/IEC 15444-1 Annex A
xxx0 0010	Scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	ITU-T Rec. T.800 ISO/IEC 15444-1 Annex A
xxx0 0011	Variable deadzone and scalar derived (values signalled for N_{LL} sub-band only). Use ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-5.	32	Table A.13
xxx0 0100	Variable deadzone derived and scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	Table A.14 then Table A.15
xxx0 0101	Variable deadzone and scalar expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	32	Table A.13
xxx0 1001	Trellis coded quantization derived (values signalled for N_{LL} sub-band only). Use ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-5.	16	ITU-T Rec. T.800 ISO/IEC 15444-1 Annex A
xxx0 1010	Trellis coded quantization expounded (values signalled for each sub-band). There are as many step sizes signalled as there are sub-bands.	16	ITU-T Rec. T.800 ISO/IEC 15444-1 Annex A
000x xxxx to 111x xxxx	Number of guard bits 0-7.		
	All other values reserved.		

Table A.13 – Quantization values (irreversible transformation only), extended

Values (bits)								Deadzone adjustment and quantization step size values
MSB				LSB				
0000	0000	0000	0000	xxxx	xxxx	xxxx	xxxx	Variable deadzone, num_nzb, value −32 768-32 767 (see Equation C-1)
1111	1111	1111	1111	xxxx	xxxx	xxxx	xxxx	
xxxx	xxxx	xxxx	xxxx	xxxx	x000	0000	0000	Mantissa, μ_b , of the quantization step size value 0-2 047 (see ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-3)
xxxx	xxxx	xxxx	xxxx	xxxx	x111	1111	1111	
xxxx	xxxx	xxxx	xxxx	0000	0xxx	xxxx	xxxx	Exponent, ε_b , of the quantization step size value 0-31 (see ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-3)
xxxx	xxxx	xxxx	xxxx	1111	1xxx	xxxx	xxxx	

Table A.14 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended

Values (bits) MSB LSB	Deadzone adjustment values (one for each sub-band)
0000 0000 0000 0000 to 1111 1111 1111 1111	First two bytes of SPqcx are the deadzone adjustment, num_nzb, value −32 768-32 767 (see Equation C-1)

Table A.15 – SPqcd, SPqcc, SPqpd, and SPqpc parameters (irreversible transformation only), extended

Values (bits) MSB LSB	Quantization step size values
xxxxx x000 0000 0000 to xxxxx x111 1111 1111	After first two bytes of SPqcx are the mantissa, μ_b , of the quantization step size value 0-2 047 (see ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-3)
0000 0xxx xxxxx xxxxx to 1111 1xxx xxxxx xxxxx	After first two bytes of SPqcx are the exponent, ϵ_b , of the quantization step size value 0-31 (see ITU-T Rec. T.800 ISO/IEC 15444-1 Equation E-3)

A.2.5 Region of interest marker (RGN), extended

If Rsiz indicates that an arbitrary region of interest is used (see Annex L), then the description of a coefficient shift and a mask are signalled in a modified RGN marker segment from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A. Table A.16 replaces ITU-T Rec. T.800 | ISO/IEC 15444-1 Table A.25. If there is RGN marker segment in the main header with a Srgn = 0, there shall not be any RGN marker segment anywhere in the codestream with a non-zero Srgn value for the component given by the corresponding Crgn value. Likewise, if there is RGN marker segment in the main header with a non-zero Srgn value, there shall not be any RGN marker segment anywhere in the codestream with a Srgn = 0 for the component given by the corresponding Crgn value.

When used in both the main header and the first tile-part header, the RGN in the first tile part header overrides the main for that tile. Also, an RGN specifying a single component (Crgn $\neq$ 65 535) overrides on specifying all components (Crgn = 65 535). Thus the order of precedence is the following:

Tile-part RGN (Crgn $\neq$ 65 535) > Tile-part RGN (Crgn = 65 535) > Main RGN (Crgn $\neq$ 65 535) > Main RGN (Crgn = 65 535)

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Table A.16 – Region-of-interest parameter values for the Srgn parameter

Values	ROI style (Srgn)	SPrgn usage
0	Implicit ROI (maximum shift)	ITU-T Rec. T.800 ISO/IEC 15444-1 Table A.26
1	Arbitrary region of interest, rectangle	Table A.18
2	Arbitrary region of interest, ellipse	Table A.18
	All other values reserved	

Table A.17 – Component index parameter value for the Crgn parameter

Parameter	Size (bits)	Values	Components index parameter
Component	16	0-16 383 16 394-65 354 65 535	Specifies component to which these region of interest descriptions apply Reserved Region of interest descriptions apply to all components

Table A.18 – Region-of-interest values from SPrgn parameter (Srgn = 1 or Srgn = 2)

Parameter	Size (bits)	Values	Meaning of SPrgn parameter
Binary shift	8	0-255	Binary shifting of coefficients in the region of interest above the background.
XArgn (left)	32	$0-(2^{32}-1)$	Horizontal reference grid point from the origin of the first point. (In the case of the ellipse, Srgn = 2, this value shall not exceed the width of the image.)
YArgn (top)	32	$0-(2^{32}-1)$	Vertical reference grid point from the origin of the first point. (In the case of the ellipse, Srgn = 2, this value shall not exceed the height of the image.)
XBrng (right)	32	$0-(2^{32}-1)$	Horizontal reference grid point from the origin of the second point.
YBrng (bottom)	32	$0-(2^{32}-1)$	Vertical reference grid point from the origin of the second point.

A.3 Extended marker segments

Table A.19 lists the markers specified in this Recommendation | International Standard.

Table A.19 – List of markers and marker segments

	Symbol	Code	Main header ^{a)}	Tile-part header ^{a)}
Variable DC offset	DCO	0xFF70	optional	optional
Visual masking	VMS	0xFF71	optional	optional
Downsampling factor style	DFS	0xFF72	optional	optional
Arbitrary decomposition style	ADS	0xFF73	optional	optional
Arbitrary transformation kernels	ATK	0xFF79	optional	optional
Component bit depth	CBD	0xFF78	optional	optional
Multiple component transformation definition	MCT	0xFF74	optional	optional
Multiple component collection	MCC	0xFF75	optional	optional
Multiple component transformation ordering	MCO	0xFF77	optional	optional
Non-linearity point transformation	NLT	0xFF76	optional	optional
Quantization default, precinct	QPD	0xFF5A	optional	optional
Quantization component, precinct	QPC	0xFF5B	optional	optional
^{a)} Required means the marker or marker segment shall be in this header if this extension is used. Optional means it may be used in the header if this extension is used.				

A.3.1 Variable DC offset (DCO)

Function: Describes the variable DC offset for every component.

Usage: Present only if the variable DC offset capability bit in the Rsiz parameter (see A.2.1) is a one value. Main and first tile-part header of a given tile. Optional in both the main and tile-part headers. No more than one shall appear in any header. If present in the main header, it describes the variable DC offset for every component in every tile. If present in the first tile-part header of a given tile, it describes the variable DC offset for every component in that tile only. When used in both the main header and the first tile-part header, the DCO in the first tile part header overrides the main for that tile. Thus the order of precedence is the following:

Tile-part DCO > Main DCO

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Shall not be used with the multiple component transform.

Length: Variable depending on the number of components.

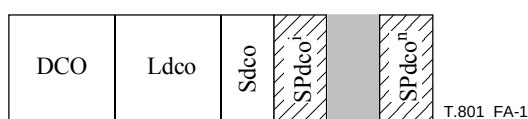


Figure A.1 – Variable DC offset syntax

DCO: Marker code. Table A.20 shows the size and parameter values for coding style component marker segment.

Ldco: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Ldco = \begin{cases} 3 + Csiz & Sdco = 0 \\ 3 + 2 \cdot Csiz & Sdco = 1 \\ 3 + 4 \cdot Csiz & Sdco = 2 \\ 3 + 8 \cdot Csiz & Sdco = 3 \end{cases} \quad (A-1)$$

where $Csiz$ is from ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A.

NOTE – If $Ldco$ were to be larger than 65 535, then the DCO marker segment cannot be used. Instead multiple component transformation functionality could be used.

Sdco: Variable DC offset type definition.

SPdcoⁱ: Variable DC offset for the i th component. There is one SPdco parameter for every component in the image.

Table A.20 – Variable DC offset parameter values

Parameter	Size (bits)	Values
DCO	16	0xFF70
Ldco	16	5-32 770
Sdco	8	Table A.21
SPdco ⁱ	variable	Table A.21

Table A.21 – Variable DC offset parameter values for the Sdco parameter

Values (bits) MSB LSB	Offset type definition
0000 0000	Offsets are 8 bit unsigned integers
0000 0001	Offsets are 16 bit signed integers
0000 0010	Offsets are 32-bit floating point (IEEE Std. 754-1985 R1990)
0000 0011	Offsets are 64-bit floating point (IEEE Std. 754-1985 R1990)
	All other values reserved

A.3.2 Visual masking (VMS)

Function: Describes the visual masking for all tile-components in the image or tile.

Usage: Present only if the visual masking capability bit in the Rsiz parameter (see A.2.1) has the value one. Optionally used in the main and/or the first tile-part header of a given tile. No more than one VMS marker segment for a component shall appear in any header. When used in both the main header and the first tile-part header, the VMS marker segment in the first tile part header overrides the one in the main header for that tile. A VMS marker segment specifying a single component ($Cvms \neq 65\,535$) overrides on specifying all components ($Cvms = 65\,535$). Thus the order of precedence is the following:

Tile-part VMS ($Cvms \neq 65\,535$) > Tile-part VMS ($Cvms = 65\,535$) > Main VMS ($Cvms \neq 65\,535$) > Main VMS ($Cvms = 65\,535$)

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Length: Fixed.

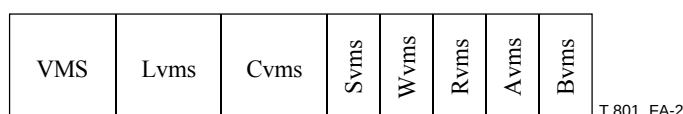


Figure A.2 – Visual masking syntax

- VMS:** Marker code. Table A.22 shows the size and parameter values for coding style, default marker segment.
- Lvms:** Length of marker segment in bytes (not including the marker). Fixed at 7 bytes.
- Cvms:** The index of the component to which this marker segment applies. Could be all components.
- Svms:** Minimal resolution level and respect block boundaries flag.
- Wvms:** Window width variable, *win_width* (see E.6).
- Rvms:** Bits retained variable, *bits_retained* (see E.6).
- Avms:** Value of the numerator of the α parameter, $\alpha = Avms/128$ (see E.6).
- Bvms:** Value of the numerator of the β parameter, $\beta = Bvms/128$ (see E.6).

Table A.22 – Visual masking parameter values

Parameter	Size (bits)	Values
VMS	16	0xFF71
Lvms	16	9
Cvms	16	Table A.23
Svms	8	Table A.24
Wvms	8	0-8
Rvms	8	0-255
Avms	8	0-255
Bvms	8	0-255

Table A.23 – Component parameter value for the Cvms parameter

Values	Component index parameter
0-16 383	Specifies component to which these region of interest descriptions apply
16 394-65 354	Reserved
65 535	Region of interest descriptions apply to all components

Table A.24 – Visual masking for the Svms parameters

Values (bits) MSB LSB	Visual masking parameters
x000 0000 to x001 0000	Minimum resolution level value, <i>minlevel</i> (0-32) (see E.6)
0xxx xxxx 1xxx xxxx	Variable <i>respect_block_boundaries</i> = 0 (see E.6) Variable <i>respect_block_boundaries</i> = 1 (see E.6)
	All other values reserved

A.3.3 Downsampling factor styles (DFS)

Function: Describes the arbitrary decomposition pattern for the lowest resolution sub-band for all tiles of a given component.

Usage: Present only if the custom decomposition style bit in the Rsiz parameter (see A.2.1) is a one value. Main header. Assigned to a component by an index in the main header COD or COC markers.

Length: Variable.

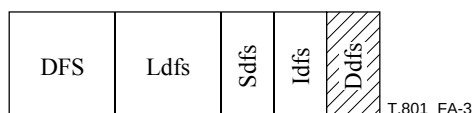


Figure A.3 – Downsampling factor styles syntax

- DFS:** Marker code. Table A.26 shows the size and values of the symbol and parameters for coding style, default marker segment.
- Ldfs:** Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Ldfs = 4 + \left\lceil \frac{Idfs}{4} \right\rceil \quad (A-2)$$

- Sdfs:** The index of this DFS marker segment. This marker segment is associated with a component via the parameter in the COD or COC marker segments found in the main header.
- Idfs:** Number of elements in the string defining the number of decomposition sub-levels.
- Ddfs:** String defining the number of decomposition sub-levels. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.

Table A.25 – Downsampling factor styles parameter values

Parameter	Size (bits)	Values
DFS	16	0xFF72
Ldfs	16	5-65 535
Sdfs	16	0-15
Idfs	8	0-255
Ddfs	variable	String of elements

A.3.4 Arbitrary decomposition styles (ADS)

Function: Describes the arbitrary decomposition pattern for a tile-component or all tile-components within a single tile.

Usage: Present only if the custom decomposition style capability bit in the Rsiz parameter (see A.2.1) is a one value. Shall not be used to describe the decomposition described in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex F. Main and first tile-part header of a given tile. There may be up to 127 such marker segments with unique index values. If an index value is found in a tile-part header, then it is used instead of an ADS marker segment in the main header with the same index value. These are assigned to a particular tile-component via the parameter in the COD or COC marker segments found only in a specific tile-part header.

Length: Variable.

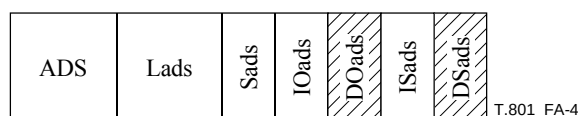


Figure A.4 – Arbitrary decomposition styles syntax

- ADS:** Marker code. Table A.26 shows the size and values of the symbol and parameters for coding style, default marker segment.
- Lads:** Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lads = 5 + \left\lceil \frac{IOads + ISads}{4} \right\rceil \quad (A-3)$$

- Sads:** The index of this ADS marker segment. This marker segment is associated with a component via the parameter in the COD or COC marker segments found in that tile-part header.
- IOads:** Number of elements in the string defining the number of decomposition sub-levels.
- DOads:** String defining the number of decomposition sub-levels. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.

ISads: Number of elements in the string defining the arbitrary decomposition structure.

DSads: String defining the arbitrary decomposition structure. The two bit elements are packed into bytes in big endian order. The final byte is padded to a byte boundary.

Table A.26 – Arbitrary decomposition styles parameter values

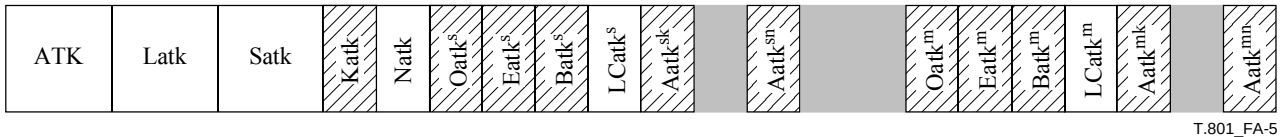
Parameter	Size (bits)	Values
ADS	16	0xFF73
Lads	16	3-65 535
Sads	8	1-127
IOads	8	0-255
DOads	variable	String of elements
ISads	8	0-255
DSads	variable	String of elements

A.3.5 Arbitrary transformation kernels (ATK)

Function: Describes a transformation kernel and an index that allows assignment to tile-components.

Usage: Present only if the arbitrary transformation kernel capability bit in the Rsiz parameter (see A.2.1) is a one value. Main and first tile-part header of a given tile. May be up to 254 marker segments in any header. A marker segment in the tile-part header with the same index as one in the main header overrides the main header marker segment.

Length: Variable.



T.801_FA-5

Figure A.5 – Arbitrary transformation default syntax

ATK: Marker code. Figure A.5 shows the size and values of the symbol and parameters for arbitrary transformation marker segment.

Latk: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$\text{Latk} = \begin{cases} 5 + 2\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(1 + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{IRR}, \text{Filt\_Cat} = \text{ARB} \\ 5 + \text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(1 + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{IRR}, \text{Filt\_Cat} = \text{WS} \\ 5 + 3\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(\text{Natk} + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{REV}, \text{Filt\_Cat} = \text{ARB} \\ 5 + 2\text{Natk} + \text{sizeof}(\text{Coeff\_Typ}) \left(\text{Natk} + \sum_{s=0}^{\text{Natk}-1} \text{LCatk}^s \right) & \text{WT\_Typ} = \text{REV}, \text{Filt\_Cat} = \text{WS} \end{cases} \quad (\text{A-4})$$

where $\text{sizeof}(\text{Coeff\_Typ})$ is the size (in bytes) of the Satk parameter that takes values of the type Coeff_Typ .

Satk: Index of the ATK marker segment; the type, Coeff_Typ , of the scaling factor and lifting step parameters; the wavelet filter category, Filt_Cat ; wavelet transformation type, WT_Typ , the initial odd or even subsequence, m_{init} . The values "0000 0000" and "0000 0001" are not available for this index having been assigned to the 9-7 irreversible wavelet filter and the 5-3 reversible wavelet filter respectively in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A.

- Katk:** The scaling factor, K . Present for irreversible transformation only, $WT\_Typ = IRR$.
- Natk:** Number of lifting steps, N_{LS} .
- Oatk^s:** Offset for lifting step s , off_s . The index, s , ranges from $s = 0$ to $Natk - 1$. Present only if $Filt\_Cat = ARB$.
- Eatk^s:** The base two scaling exponent for lifting step s , ε_s . Present only with reversible transformation, $WT\_Typ = REV$. The index, s , ranges from $s = 0$ to $Natk - 1$.
- Batk^s:** Additive residue for lifting step, s . Present for reversible transformations ($WT\_Typ = REV$) only. The index, s , ranges from $s = 0$ to $Natk - 1$.
- LCatk^s:** Number of lifting coefficients signalled for lifting step s . Provides the range, k , for $Aatk^{sk}$. The index, s , ranges from $s = 0$ to $Natk - 1$.
- Aatk^{sk}:** The k th lifting coefficient for the lifting step s , $\alpha_{s,k}$. The index, s , ranges from $s = 0$ to $Natk - 1$. The index, k , ranges from $k = 0$ to $LCatk - 1$.

Table A.27 – Arbitrary transformation parameter values

Parameter	Size (bits)	Values
ATK	16	0xFF79
Latk	16	9-65 535
Satk	16	Table A.28
Katk	0 8 16 32 64 128	$WT\_Typ = REV$ $WT\_Typ = IRR, Coeff\_Typ = 0$ $WT\_Typ = IRR, Coeff\_Typ = 1$ $WT\_Typ = IRR, Coeff\_Typ = 2$ $WT\_Typ = IRR, Coeff\_Typ = 3$ $WT\_Typ = IRR, Coeff\_Typ = 4$
Natk	8	0-255
Oatk ^s	0 8	$Filt\_Cat = WS$ -128-127; $Filt\_Cat = ARB$
Eatk ^s	0 8	$WT\_Typ = IRR$ 0-255; $WT\_Typ = REV$
Batk ^s	0 8 16 32 64 128	$WT\_Typ = IRR$ $Coeff\_Typ = 0$ $Coeff\_Typ = 1$ $Coeff\_Typ = 2$ $Coeff\_Typ = 3$ $Coeff\_Typ = 4$
LCatk ^s	8	0-255
Aatk ^{sk}	8 16 32 64 128	$Coeff\_Typ = 0$ $Coeff\_Typ = 1$ $Coeff\_Typ = 2$ $Coeff\_Typ = 3$ $Coeff\_Typ = 4$

Table A.28 – Arbitrary transformation values for the Satk parameter

Values (bits) MSB LSB	Index
xxxx xxxx 0000 0010 to xxxx xxxx 1111 1111	Index of this marker segment (2-255)
xxxx x000 xxxx xxxx xxxx x001 xxxx xxxx xxxx x010 xxxx xxxx xxxx x011 xxxx xxxx xxxx x100 xxxx xxxx	Parameters 8-bit signed integer, <i>Coeff_Typ</i> = 0 Parameters 16-bit signed integer, <i>Coeff_Typ</i> = 1 Parameters 32-bit floating point (IEEE Std. 754-1985 R1990), <i>Coeff_Typ</i> = 2 Parameters 64-bit floating point (IEEE Std. 754-1985 R1990), <i>Coeff_Typ</i> = 3 Parameters 128-bit floating point (IEEE Std. 754-1985 R1990), <i>Coeff_Typ</i> = 4
xxxx 0xxx xxxx xxxx x1xx 1xxx xxxx xxxx	Arbitrary filters, <i>Filt_Cat</i> = <i>ARB</i> WS filters, <i>Filt_Cat</i> = <i>WS</i>
xxx0 xxxx xxxx xxxx xxx1 xxxx xxxx xxxx	Irreversible filter, <i>WT_Typ</i> = <i>IRR</i> Reversible filter, <i>WT_Typ</i> = <i>REV</i>
xx0x xxxx xxxx xxxx xx1x xxxx xxxx xxxx	Modify even-indexed subsequence in first reconstruction step, <i>m_{init}</i> = 0 Modify odd-indexed subsequence in first reconstruction step, <i>m_{init}</i> = 1
x0xx 0xxx xxxx xxxx x1xx xxxx xxxx xxxx	Boundary extension method used in lifting steps is constant, <i>Exten</i> = <i>CON</i> Boundary extension method used in lifting steps is whole-sample symmetric, <i>Exten</i> = <i>WS</i>
	All other values reserved

A.3.6 Component bit depth definition (CBD)

Function: Defines the bit depth of reconstructed image components coming out of any multiple component transformation process.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see A.2.1) is a one value. Main header. The CBD marker segment is required if the multiple component transformation processes are used. At most there can be one CBD in the main header.

The presence of a CBD marker segment in a codestream alters the procedures used to determine the precision of output image components and the interpretation of the SIZ marker. See Annex J for further details.

Length: Variable depending on the number of reconstructed image component bit depths signalled.

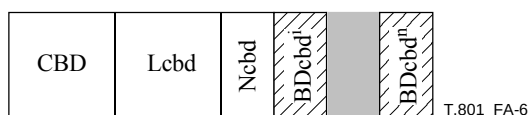


Figure A.6 – Component bit depth definition syntax

CBD: Marker code. Table A.29 shows the size and parameter values for component bit depth definition syntax.

Lcbd: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lmct = \begin{cases} 5 & \text{Ncbd} = 1xxx xxxx xxxx xxxx \\ 4 + \text{Ncbd} & \text{Ncbd} = 1xxx xxxx xxxx xxxx \end{cases} \quad (\text{A-5})$$

Ncbd: Number of component bit depths included in marker segment. Table A.30 shows the value for the Ncbd parameter.

BDcbdⁱ: Bit depth and sign of the reconstructed image components in the order in which they are created as determined by the MCC and MCO marker segments. Either one value is signalled for all components (see Table A.30) or an individual bit depth is given for each component.

Table A.29 – Component bit depth definition parameter values

Parameter	Size (bits)	Values
CBD	16	0xFF78
Lcbd	16	5-16 388
Ncbd	16	Table A.30
BDcbd ⁱ	8	Table A.31

Table A.30 – Component bit depth definition values for the Ncbd parameter

Values (bits) MSB LSB	Number of reconstructed image component bit depths in marker
x000 0000 0000 0001 to x100 0000 0000 0000	Number of reconstructed image components (1-16 384)
0xxx xxxx xxxx xxxx 1xxx xxxx xxxx xxxx	Bit depths included, one per reconstructed image component One bit depth included, applies to <i>all</i> reconstructed image components
	All other values reserved

Table A.31 – Component bit depth definition values for the BDcbdⁱ parameter

Values (bits) MSB LSB	Reconstructed image component bit depths
x000 0000 to x010 0101	Component sample bit depth = value + 1. From 1 bit deep through 38 bits deep respectively.
0xxx xxxx	Component sample values are unsigned values
1xxx xxxx	Component sample values are signed values
	All other values reserved

A.3.7 Multiple component transformation definition (MCT)

Function: Defines one multiple component transformation array per marker segment. The type and index of the array defined in this marker distinguishes it from other MCT marker segments in a given header. This array can be assigned to a collection of components within the MCC marker segment.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see A.2.1) is a one value. Main and first tile-part header of a given tile. There may be up to 255 MCT marker segments, or series of marker segments, in the main header. There may be up to 255 MCT marker segments, or series of marker segments, in any tile-part header. An MCT marker segment in a tile-part header with the same index (Imct) as one in the main header overrides the main header MCT segment for that tile.

A series of MCT marker segments (defined as having the same Imct value in the same header and a Ymct > 0) shall all appear in the same header in order of (consecutive) Zmct parameter values.

To apply the transformation array included in an MCT marker segment, an MCC marker segment must exist that associates the MCT marker segment with a component collection. This association is made through the array definition index of the MCT marker segment and the Tmccⁱ fields of MCC marker segments. If no such MCC marker segment exists, then the transformation array included in the MCT marker segment shall not be used in the decoding process.

Length: Variable depending on the size of the array.

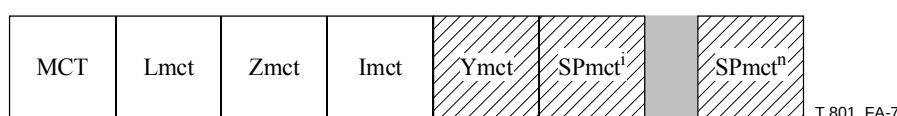


Figure A.7 – Multiple component transformation definition syntax

- MCT:** Marker code. Table A.32 shows the size and parameter values for multiple component transformation definition marker segment.
- Lmct:** Length of marker segment in bytes (not including the marker).
- Zmct:** Index of this marker segment in a series of MCT marker segments. All the marker segments in the series have the same Imct parameter value present in this header. The data in each subsequent MCT marker segment shall be appended, in order, to make on stream of SPmctⁱ parameter values. The Ymct parameter values are present only in the first marker segment in the series (Zmct = 0).
- Imct:** Multiple component transformation index value, array type, and parameter size. An MCT marker segment, or series, with a given Imct value in the tile-part header overrides a main header MCT marker segment, or series, with the same Imct value.
- Ymct:** Index of the last number of MCT marker segment in the series. For every series of MCT marker segments (i.e., MCT marker segments in this header with the same Imct parameter value), there shall be MCT marker segment with Zmct parameter values of 0 to Ymct. The last MCT marker segment will have Zmct = Ymct. This value is present only in the first marker segment in the series (Zmct = 0).
- SPmctⁱ:** Parameters for the multiple component transformation definition. One parameter value for each element in the array. See J.2 to determine the number of array elements and their order in the marker segment. The number of elements in a row and the number of rows (elements in a column) are determined by the type of array and the number of the input and output components to which it is assigned.

Table A.32 – Multiple component transformation definition parameter values

Parameter	Size (bits)	Values
MCT	16	0xFF74
Lmct	16	6-65 535
Zmct	16	0-65 535
Imct	16	Table A.33
Ymct	0 16	If Zmct > 0 0-65 535
SPmct ⁱ	variable	Table A.33

Table A.33 – Multiple component transformation definition values for the Imct parameter

Values (bits) MSB LSB	Index of the array definition, type and parameter type
xxxx xxxx 0000 0001 to xxxx xxxx 1111 1111	Index of the array definition, 1-255
xxxx xx00 xxxx xxxx	Dependency transformation array type
xxxx xx01 xxxx xxxx	Decorrelation transformation array type
xxxx xx10 xxxx xxxx	Offset array type
xxxx 00xx xxxx xxxx	Array elements are 16 bit signed integers
xxxx 01xx xxxx xxxx	Array elements are 32 bit signed integers
xxxx 10xx xxxx xxxx	Array elements are 32-bit floating point (IEEE Std. 754-1985 R1990)
xxxx 11xx xxxx xxxx	Array elements are 64-bit floating point (IEEE Std. 754-1985 R1990)
	All other values reserved

A.3.8 Multiple component transform collection (MCC)

Function: Describes the collection of input intermediate components, the collection of output intermediate components, and the associated wavelets or arrays for a multiple component transform. This marker segment can appear in the main header and can be referred to or overridden by an MCC marker in a tile-part header.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see A.2.1) is a one value. Main and first tile-part header of a given tile. There may be up to 255 MCC marker segments, or series of marker segments, in the main header. There may be up to 255 MCC marker segments, or series of marker segments, in any tile-part header. An MCC marker segment in a tile-part header with the same index ($Imcc$) as one in the main header overrides the main header MCC segment for that tile.

A series of MCC marker segments (defined as having the same $Imcc$ value in the same header and a $Ymcc > 0$) shall all appear in the same header in order of (consecutive) $Zmcc$ parameter values.

Length: Variable depending on the number of component collections.

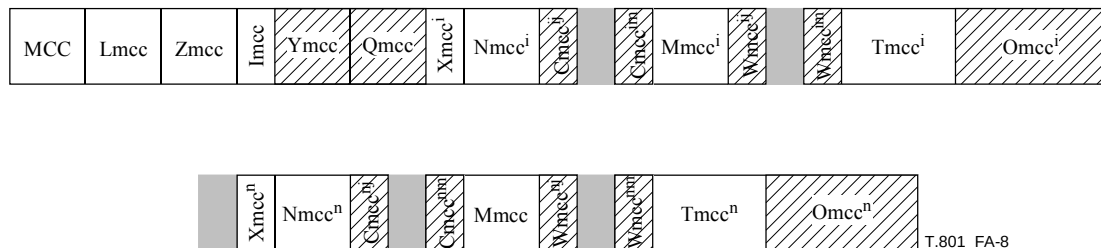


Figure A.8 – Multiple component collection syntax

- MCC:** Marker code. Table A.34 shows the size and parameter values for multiple component collection marker segment.
- Lmcc:** Length of marker segment in bytes (not including the marker).
- Zmcc:** Index of this marker segment in a series of MCC marker segments. All the marker segments in the series have the same $Imcc$ parameter value present in this header. The data in each subsequent MCC marker segment shall be appended, in order, to make one stream of the other parameters. The $Ymcc$ and $Qmcc$ parameter appears only in the first marker segment ($Zmcc = 0$).
- Imcc:** Index of this marker segment. An MCC marker segment, or series, with a given $Imcc$ value in the tile-part header overrides a main header MCC marker segment, or series, with the same $Imcc$ value.
- Ymcc:** Index of the last number of MCC marker segment in the series. For every series of MCC marker segments (i.e., MCC marker segments in this header with the same $Imcc$ parameter value), there shall be MCC marker segment with $Zmcc$ parameter values of 0 to $Ymcc$. The last MCC marker segment will have $Zmcc = Ymcc$. This value is present only in the first marker segment in the series ($Zmcc = 0$).
- Qmcc:** The number of collections in the MCC marker segment. This value is present only in the first marker segment in the series ($Zmcc = 0$).
- Xmccⁱ:** Indicates type of multiple component transform used for the i th component collection (wavelet or array-based decorrelation or array-based dependency). Defines the interpretation applied to $Tmcc$.
- Nmccⁱ:** Indicates the number of input components for the i th component collection and defines the number of bits (8 or 16) used to represent the component indices in i th collection.
- Cmcc^{ij}:** Input intermediate component indices included the i th component collection. The number of indices in the i th component collection is $Nmcc^i$. Each index denotes an input intermediate component. The order of the indices defines the ordering applied to the input intermediate components prior to application of the inverse transform.
- Mmccⁱ:** Indicates the number of output intermediate components for the i th component collection and defines the number of bits (8 or 16) used to represent the component indices in i th collection. If anything other than an array-based irreversible decorrelation transform is used, $Mmcc^i$ must equal $Nmcc^i$.
- Wmcc^{ij}:** Intermediate component indices included the i th output component collection. The number of indices in the i th component collection is $Mmcc^i$. All output intermediate component indices in a given MCC marker segment shall appear only once across all collections in that MCC marker.
- Tmccⁱ:** For array-based component collection transforms, $Tmcc^i$ assigns arrays defined in an MCT marker segment to the i th component collection. An MCT marker segment with the right type and index in the first tile-part header of a tile is used before an MCT marker segment with the right type and index in the main header. $Tmcc^i$ also indicates the reversibility of array-based component transforms.

For wavelet-based component collection transforms, $Tmcc^i$ assigns a wavelet kernel defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A or an ATK marker segment and the number of wavelet decomposition levels for the i th component collection (only the dyadic decomposition of ITU-T Rec. T.800 | ISO/IEC 15444-1 is supported). An ATK marker segment with the proper index in the first tile-part header of a tile is used before an ATK marker segment with the proper index in the main header. $Tmcc^i$ also contains the index of an MCT marker segment that contains component additive offsets.

Omccⁱ: Present in the MCC marker segment only for those component collections that use a wavelet-based transform. $Omcc^i$ indicates the reference grid offset to apply in the component dimension for the i th component collection (see J.2.2).

Table A.34 – Multiple component collection parameter values

Parameter	Size (bits)	Values
MCC	16	0xFF75
Lmcc	16	5-65 535
Zmcc	16	0-65 535
Imcc	8	1-255
Ymcc	0 16	If $Zmcc > 0$ 0-65 535
Qmcc	0 16	If $Zmcc > 0$ 0-16 383
$Xmcc^i$	8	Table A.35
$Nmcc^i$	16	Table A.36
$Cmcc^{ij}$	8 16	0-255 0-16 383
$Mmcc^i$	16	Table A.37
$Wmcc^{ij}$	8 16	0-255 0-16 383
$Tmcc^i$	24	Table A.35
$Omcc^i$	32	0-4 294 967 295

Table A.35 – Multiple component collection values for the $Xmcc^i$ parameter

Values (bits) MSB LSB	Coding style	$Tmcc^i$ parameter
xxxx xx00 xxxx xx01 xxxx xx11	Component collection transform is array-based dependency transform Component collection transform is array-based decorrelation transform Component collection transform is wavelet-based transform	Table A.38 Table A.38 Table A.39
	All other values reserved	

Table A.36 – Multiple component collection values for the $Nmcc^i$ parameter

Values (bits) MSB LSB	Coding style
0xxx xxxx xxxx xxxx to 1xxx xxxx xxxx xxxx	Input component collection indices ($Cmcc^i$) are 8 bit integers Input component collection indices ($Cmcc^i$) are 16 bit integers
x000 0000 0000 0001 to x100 0000 0000 0000	Number of input components in i th component collection (1-16 384)

Table A.37 – Multiple component collection values for the Mmccⁱ parameter

Values (bits) MSB LSB	Coding style
0xxx xxxx xxxx xxxx to 1xxx xxxx xxxx xxxx	Output component collection indices (Wmcc ⁱ) are 8 bit integers Output component collection indices (Wmcc ⁱ) are 16 bit integers
x000 0000 0000 0001 to x100 0000 0000 0000	Number of output components in ith component collection (1-16 384)

Table A.38 – Multiple component collection values for the Tmccⁱ parameter (array-based)

Values (bits) MSB LSB	Coding style
xxxx xxxx xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 0000 0001 to xxxx xxxx xxxx xxxx 1111 1111	No assigned transform array (NULL array) Decorrelation or dependency transform array index
xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 0000 0001 xxxx xxxx to xxxx xxxx 1111 1111 xxxx xxxx	No assigned offset array (NULL array) Offset array index
xxxx xxx0 xxxx xxxx xxxx xxxx xxxx xxx1 xxxx xxxx xxxx xxxx	Decorrelation or dependency transform is irreversible Decorrelation or dependency transform is reversible
	All other values reserved

Table A.39 – Multiple component collection values for the Tmccⁱ parameter (wavelet-based)

Values (bits) MSB LSB	Coding style
xxxx xxxx xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 0000 0001 xxxx xxxx xxxx xxxx 0000 0010 to xxxx xxxx xxxx xxxx 1111 1111	ITU-T Rec. T.800 ISO/IEC 15444-1 9-7 irreversible filter ITU-T Rec. T.800 ISO/IEC 15444-1 5-3 reversible filter Index of the ATK marker segment containing the wavelet kernel for component collection
xxxx xxxx 0000 0000 xxxx xxxx xxxx xxxx 1111 1111 xxxx xxxx	Index of MCT marker segment containing additive offsets for component collection
xx00 0000 xxxx xxxx xxxx xxxx xx10 0000 xxxx xxxx xxxx xxxx	Number of dyadic wavelet decomposition levels used in the component collection (0-32)
	All other values reserved

A.3.9 Multiple component transform ordering (MCO)

Function: Describes the order in which multiple component transforms are applied during inverse multiple component transform processing.

Usage: Present only if the multiple component transformation capability bit in the Rsiz parameter (see A.2.1) is a one value. At most one MCO marker segment in main and first tile-part header of a given tile. If used in the main header, this marker segment defines the default ordering of multiple component transform stages for all tiles. If used in the first tile-part header, then the component transform order established by the MCO marker segment overrides any default ordering defined by a main header MCO marker segment.

Length: Variable depending on the number of multiple component transform stages used.

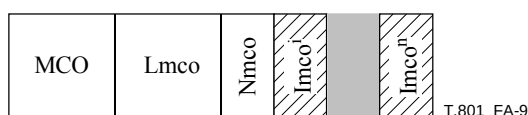


Figure A.9 – Multiple component transform ordering syntax

MCO: Marker code. Table A.40 shows the size and parameter values for the multiple component transform ordering marker segment.

Lmco: Length of marker segment in bytes (not including the marker). The length is given by the following expression:

$$Lmco = 3 + Nmco \quad (A-6)$$

Nmco: Number of multiple component transform stages specified for inverse transform processing. If $Nmco = 0$, then no multiple component transform processing is used for the current tile and no $Imco^i$ parameters shall appear. Otherwise, $Nmco$ specifies the number of MCC marker segment identifiers that will follow.

$Imco^i$: Index of the MCC marker segment containing the component collection information for the i th inverse multiple component transform stage (see A.3.8).

Table A.40 – Multiple component intermediate collection parameter values

Parameter	Size (bits)	Values
MCO	16	0xFF77
Lmco	16	3-258
Nmco	8	0-255
$Imco^i$	8	0-255

A.3.10 Non-linearity point transformation (NLT)

Function: Describes either a gamma or LUT non-linearity to be applied to a single component or all components.

Usage: Present only if the non-linearity point transformation capability bit in the Rsiz parameter (see A.2.1) is a one value. Main and first tile-part header of a given tile. There may be no more than one marker segment per component plus one default in any header.

When used in the main header, the defined non-linearity can be established as a default for all components or established as a default for a single component. When used in a tile-part header, it can be used to establish a default for all components in the tile or to set the non-linearity transformation for a single component in that tile. Thus, the order of precedence is the following:

Tile-part NLT > Tile-part NLT default > Main NLT > Main NLT default

where the "greater than" sign, >, means that the greater overrides the lesser marker segment.

Length: Variable depending on the value of $Tnlt$.

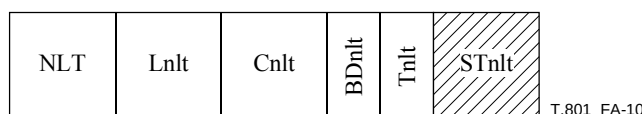


Figure A.10 – Non-linearity point transformation syntax

NLT: Marker code. Table A.41 shows the size and values of the symbol and parameters for non-linearity point transformation marker segment.

Lnlt: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lnlt = 6 + \begin{cases} 15 & Tnlt = 1 \\ 11 + (N_{points} \cdot \psi_{Tval}) & Tnlt = 2 \end{cases} \quad (A-7)$$

$$\psi_{Tval} = \begin{cases} 1 & PTval \in [1, 8] \\ 2 & PTval \in [9, 16] \\ 4 & PTval \in [17, 32] \end{cases}$$

- Cnlt:** The index of the component to which this marker segment relates. The components are indexed 0, 1, 2, etc. If this value is 65 535, then this marker segment applies to all components. Table A.42 shows the value for the Cnlt parameter.
- BDnlt:** Bit depth and sign of the decoded image component, Z_i , after processing of the i th reconstructed image component by the non-linearity. If Cnlt = 65 535, then this value applies to all components. Table A.43 shows the values for the BDnlt parameter.
- Tnlt:** Non-linearity type. Table A.44 shows the value for the Scod parameter.
- STnlt:** Parameter values associated with the non-linearity as controlled by the Tnlt field.

Table A.41 – Non-linearity transformation parameter values

Parameter	Size (bits)	Values
NLT	16	0xFF76
Lnlt	16	12-65 535
Cnlt	16	Table A.42
BDnlt	8	Table A.43
Tnlt	8	Table A.44
STnlt	variable	Table A.44

Table A.42 – Non-linearity transformation parameter values for the Cnlt parameter

Values	Components index parameter
0-16 383	Defines component to which these non-linearity transformation descriptions in this marker segment apply
65 535	Non-linearity transformation descriptions in this marker segment apply to all components
	All other values reserved

Table A.43 – Decoded image component bit depth parameter values for the BDnlt parameter

Values (bits) MSB LSB	Decoded image component bit depth
x000 0000 to x010 0101	Component sample bit depth = value + 1. From 1 bit deep through 38 bits deep respectively.
0xxx xxxx	Component sample values are unsigned values
1xxx xxxx	Component sample values are signed values
	All other values reserved

Table A.44 – Non-linearity transformation parameter values of the Tnlt parameter

Values (bits) MSB LSB	Meaning of Tnlt values	STnlt usage
0000 0000	No non-linearity transformation applied	—
0000 0001	Gamma-style non-linearity transformation	Table A.45
0000 0010	LUT-style non-linearity transformation	Table A.46
	All other values reserved	—

Table A.45 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 1)

Parameter (in order)	Size (bits)	Values	Meaning of STnlt values
E	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65\,535\}}{65\,535}$	Non-linearity exponent (8-bit integer + 16-bit fraction)
S	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65\,535\}}{65\,535}$	Non-linearity toe slope (8-bit integer + 16-bit fraction)
T	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65\,535\}}{65\,535}$	Non-linearity threshold (8-bit integer + 16-bit fraction)
A	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65\,535\}}{65\,535}$	Non-linearity continuity parameter A (8-bit integer + 16-bit fraction)
B	24	$\{0, \dots, 255\} + \frac{\{0, \dots, 65\,535\}}{65\,535}$	Non-linearity continuity parameter B (8-bit integer + 16-bit fraction)

Table A.46 – Non-linearity transformation parameter values of the STnlt parameter (Tnlt = 2)

Parameters (in order)	Size (bits)	Values	Meaning of STnlt values
Npoints	16	1-8 191	(Number of points – 1) in the LUT-style non-linearity definition (all other values reserved)
Dmin	32	$0-(2^{32}-1)$	Dmin = parameter value / $(2^{32}-1)$
Dmax	32	$1-(2^{32}-1)$	Dmax = parameter value / $(2^{32}-1)$
PTval	8	0000 0001 to 0010 0000	Precision of Tvalue parameter in bits (1-32). This also implies how many bytes are used to express the Tvalue (all other values reserved)
Tvalue	8, PTval ≤ 8; 16, 9 ≤ PTval ≤ 16 32, PTval > 16	variable	Run of table values for the LUT-style non-linearity. The (Npoints + 1) parameters are unsigned integers. The actual value of Tvalue is Tvalue = parameter value / $(2^{PTval}-1)$

A.3.11 Quantization default, precinct (QPD)

Function: Describes the quantization default used for compressing all components of a particular resolution level and precinct. The parameter values can be overridden for an individual component, resolution level, and precinct by a QPC marker segment which, if present, must appear in a tile-part header prior to any packets for that component, resolution level, and precinct.

Usage: Main and any tile-part header. Several QPD marker segments may appear in any tile-part header, but only one for each resolution level and precinct. If a QPD is used in a tile-part header it overrides the quantization characteristics defined by either QCD or QCC marker segments for all components of the resolution level and precinct indexed by the QPD within the scope of the particular tile. Thus, the quantization characteristics of a particular resolution level, precinct pair is determined by the presence of QCD, QCC, QPD or QPC markers in the following order of precedence:

Any tile-part QPC > Any tile-part QPD > First tile-part QCC > First tile-part QCD >
Main QPC > Main QPD > Main QCC > Main QCD

When QPD marker segments are used, they must appear in tile-part headers before any packets are found for the indexed resolution level and precinct.

Length: Variable depending on the number of quantized sub-bands within the resolution level indexed.



Figure A.11 – Quantization default, precinct syntax

QPD: Marker code. Table A.47 shows the size and values of the symbol and parameters for quantization default, precinct marker segment.

Lqpd: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lqpd = \begin{cases} 5 + \text{num\_subbands\_lev} & \text{no\_quant AND PLqpd} < 128 \\ 7 & \text{quant\_derived AND PLqpd} < 128 \\ 5 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND PLqpd} < 128 \\ 6 + \text{num\_subbands\_lev} & \text{no\_quant AND PLqpd} \geq 128 \\ 8 & \text{quant\_derived AND PLqpd} \geq 128 \\ 6 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND PLqpd} \geq 128 \end{cases} \quad (\text{A-8})$$

where *num_subbands_lev* can be derived from F.2.4 for each resolution level and whether this marker segment has *no_quant*, *quant_derived*, and *quant_expounded* is signalled in the *Sqpd* parameter.

NOTE – The *Lqpd* can be used to determine how many quantization step sizes are present in the marker segment. However, there is not necessarily a correspondence with the number of sub-bands present because the sub-bands can be truncated with no requirement to correct this marker segment.

PLqpd: The resolution level index for the quantization values signalled. Equation A-9 shows how this marker segment is constructed based on the resolution level index, *lev*, as well as the precinct index, *prec*.

$$PLqpd = \begin{cases} lev & prec < 256 \\ 128 + lev & prec \geq 256 \end{cases} \quad (\text{A-9})$$

The resolution level index, *lev*, can range from 0 to N_L , where N_L is the number of decomposition levels defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, A.6.1.

PPqpd: The precinct index for the quantization values signalled. The size of this marker segment parameter will be one byte when the *PLqpd* parameter is less than 128, but two bytes when *PLqpd* is greater than or equal to 128. This parameter will then just hold the precinct index, *prec*. The precinct index, *prec*, can range from 0 to *numprecincts* – 1, where *numprecincts* is the number of precincts at resolution level *lev* and is also defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, B.6.

Sqpd: Quantization style for all components at the resolution level, *lev*, and precinct, *prec*.

SPqpd¹: Quantization step size value for the *i*th sub-band at the resolution level, *lev*, in the order defined for *lev* in F.2.4. The number of parameters is at least as large as the number of sub-bands in the tile-component with the greatest number of sub-bands at resolution level, *lev*.

Table A.47 – Quantization default, precinct parameter values

Parameter	Size (bits)	Values
QPD	16	0xFF5A
Lqpd	16	6-101
PLqpd	16	0-32 or 128-160
PPqpd	8 16	0-255, <i>PLqpd</i> < 128 0-65 535, <i>PLqpd</i> ≥ 128
Sqpd	8	Table A.12
SPqpd ¹	variable	Table A.12

A.3.12 Quantization precinct component (QPC)

Function: Describes the quantization used for compressing a particular component, resolution level, and precinct.

Usage: Main and any tile-part header. Several QPC marker segments may appear in any tile-part header, but only one for each component, resolution level, and precinct. If a QPC is used in a tile-part header it overrides the quantization characteristics defined by QCD, QCC, or QPD marker segments for the triplet indexed by the QPC within the scope of the particular tile. Thus, the quantization characteristics of a particular component, resolution level, and precinct is determined by the presence of QCD, QCC, QPD or QPC markers in the following order of precedence:

Any tile-part QPC > Any tile-part QPD > First tile-part QCC > First tile-part QCD >
Main QPC > Main QPD > Main QCC > Main QCD

When QPC marker segments are used, they must appear in tile-part headers before any packets are found for the indexed component, resolution level, and precinct.

Length: Variable depending on the number of quantized sub-bands within the resolution level indexed.

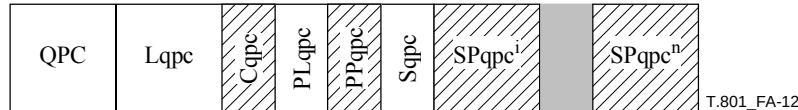


Figure A.12 – Quantization precinct component syntax

QPC: Marker code. Table A.48 shows the size and values of the symbol and parameters for quantization component marker segment.

Lqpc: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lqpc = \begin{cases} 6 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 8 & \text{quant\_derived AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 6 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} < 257 \text{ AND Plqpc} < 128 \\ 7 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 9 & \text{quant\_derived AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 7 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} \geq 257 \text{ AND Plqpc} < 128 \\ 7 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 9 & \text{quant\_derived AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 7 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} < 257 \text{ AND Plqpc} \geq 128 \\ 8 + \text{num\_subbands\_lev} & \text{no\_quant AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \\ 10 & \text{quant\_derived AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \\ 8 + 2 \cdot \text{num\_subbands\_lev} & \text{quant\_expounded AND Csiz} \geq 257 \text{ AND Plqpc} \geq 128 \end{cases} \quad (\text{A-10})$$

where *num_subbands_lev* can be derived from F.2.4 for each resolution level and whether this marker segment has *no_quant*, *quant_derived*, and *quant_expounded* is signalled in the *Sqpc* parameter.

NOTE – The *Lqpc* can be used to determine how many step sizes are present in the marker segment. However, there is not necessarily a correspondence with the number of sub-bands present because the sub-bands can be truncated with no requirement to correct this marker segment.

Cqpc: The index of the component to which this marker segment relates. The components are indexed 0, 1, 2, etc. (Either 8 or 16 bits depending on *Csiz* value.)

PLqpc: The resolution level index for the quantization values signalled. Equation A-11 shows how this marker segment is constructed based on the resolution level index, *lev*, as well as the precinct index, *prec*.

$$PLqpc = \begin{cases} lev & prec < 256 \\ 128 + lev & prec \geq 256 \end{cases} \quad (\text{A-11})$$

The resolution level index, *lev*, can range from 0 to N_L , where N_L is the number of decomposition levels defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, A.6.1.

PPqpc: The precinct index for the quantization values signalled. The size of this marker segment parameter will be one byte when the PLqpc parameter is less than 128, but two bytes when PLqpc is greater than or equal to 128. This parameter will then just hold the precinct index, *prec*. The precinct index, *prec*, can range from 0 to *numprecincts* - 1, where *numprecincts* is the number of precincts at resolution level *lev* and is also defined in ITU-T Rec. T.800 | ISO/IEC 15444-1, B.6.

Sqpc: Quantization style for this component, resolution level, *lev*, and precinct, *prec*.

SPqpcⁱ: Quantization value for the *i*th sub-band at resolution level, *lev*, in the defined order for *lev* in F.2.4. The number of parameters is at least as large as the number of sub-bands in the relevant tile-component with the greatest number of sub-bands at resolution level, *lev*.

Table A.48 – Quantization precinct component parameter values

Parameter	Size (bits)	Values
QPC	16	0xFF5B
Lqpc	16	5-199
Cqpc	8 16	0-255; if Csiz < 257 0-16 383; Csiz ≥ 257
PLqpc	16	0-32 or 128-160
PPqpc	8 16	0-255, PLqpd < 128 0-65 535, PLqpd ≥ 128
Sqpc	8	Table A.12
SPqpc ⁱ	variable	Table A.12

Annex B

Variable DC offset, extension

(This annex forms an integral part of this Recommendation | International Standard)

In this annex and all of its subclauses, the flow charts and tables are normative only in the sense that they are defining an output that alternative implementations shall duplicate. This annex describes an extension to ITU-T Rec. T.800 | ISO/IEC 15444-1 that can be used alone or in conjunction with any of the other extensions in this Recommendation | International Standard, except the Multiple Component Transformation in Annex J. The capabilities of the codestream are defined by the SIZ marker segment parameter Rsiz (see A.2.1).

This annex specifies variable DC offset that converts the signed values resulting from the decoding process to the proper reconstructed samples. It can be applied to with both signed and unsigned component data.

B.1 Variable DC offset flow

DC offset occurs external to any component transformations, i.e., prior to component transformations during encoding and after component transformations during decoding. Figure B.1 shows the flow of DC offset in the system with a multiple component transformation from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex G.

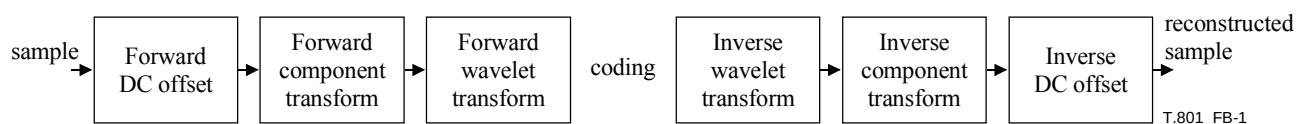


Figure B.1 – Placement of the DC offset with multiple component transformation

Figure B.2 shows the flow of DC offset in the system without a multiple component transformation from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex G.

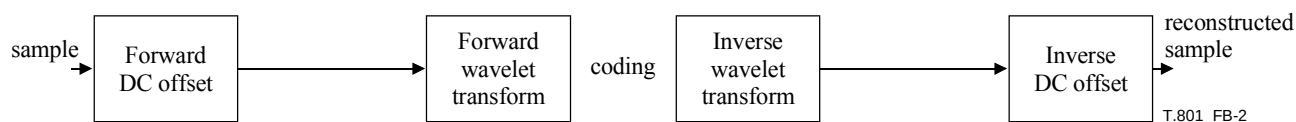


Figure B.2 – Placement of the DC offset without multiple component transformation

B.2 Inverse DC offset

If the DCO marker segment is present in the main or tile-part header (see A.3.1), then the DC offset, O_i , is specified by that marker. All samples of a given component are offset by adding the same quantity to each sample as follows:

$$I'(x, y) = I(x, y) + O_i \quad (\text{B-1})$$

If no DCO marker segment is present, then the DC offset is performed as described in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex G.

NOTE – If I' is not of the same precision or dynamic range as the output data it may be rounded to the closest precision and clipped in bounds.

B.3 Forward DC offset (informative)

The variable DC offset allows user control of the actual offset value. The offset, O_i , may be chosen as any value, but it is suggested that it be within the dynamic range and precision of the original data. The default value for unsigned data is:

$$O_i = 2^{Ssiz^i} \quad (\text{B-2})$$

where Ss_{iz}^i comes from ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex A. For signed data the default in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex G, is to have no offset ($O_i = 0$). Any other value must be signalled in a DCO marker segment in either the main or tile-part header. All input data is then offset by subtracting the fixed offset from all samples in the tile component. When using a variable DC offset in conjunction with a reversible transformation, the offset should be an integer value.

$$I(x, y) = I'(x, y) - O_i \quad (\text{B-3})$$

When a non-default offset is used, care must be taken to adjust the number of guard bits to account for any potential increase in bit depth of the offset data. If the offset, O_i , is chosen within the dynamic range of the original data, then increasing the number of guard bits, G , by one will be sufficient to handle any potential increase.

For most images the default offset setting gives good compression performance. However for some images that have sharply peaked histograms, with a small amount of highly contrasting data, significantly improved performance can be obtained if the offset is set nearer the mode of the histogram.

Annex C

Variable scalar quantization, extension

(This annex forms an integral part of this Recommendation | International Standard)

In this annex and all of its subclauses, the flow charts and tables are normative only in the sense that they are defining an output that alternative implementations shall duplicate. This annex describes an extension to ITU-T Rec. T.800 | ISO/IEC 15444-1 that can be used alone or in conjunction with any of the other extensions in this Recommendation | International Standard. The capabilities of the codestream are defined by the SIZ marker segment parameter Rsiz (see A.2.1).

Variable scalar quantization extends the default scalar quantization described in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex E, to allow deadzones of variable width (up to four times the step size). Variable scalar quantization shall only be used with irreversible filters.

C.1 Variable scalar quantization

All terminology and variables described in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex E, remain the same for variable scalar quantization. An additional parameter nz_b is used to convey the adjusted deadzone size. The adjusted deadzone size is $2(1-nz_b)\Delta_b$. When $nz = 0$ this is equivalent to the scalar quantizer in ITU-T Rec. T.800 | ISO/IEC 15444-1, Annex E, with a deadzone of twice the stated step size. When $nz_b > 0$ then the deadzone is smaller, and when $nz_b < 0$ then the deadzone is larger. The value of nz_b must lie in the range $[-1, 1)^{1)}$.

When $nz = 0$ for all sub-bands, there is no need to transmit adjusted deadzone factors. However, if $nz_b \neq 0$ for at least one sub-band, then extended QCD, QCC, QPD, and/or QPC marker segments shall appear in either the main header or the first tile-part header of a given tile (see A.2.4). The value nz_b is represented as:

$$nz_b = \frac{\text{num } nz_b}{2^{15}} \quad (\text{C-1})$$

When an extended QCD, QCC, QPD, QPC marker segment appears, the adjustment factors are either signalled for each sub-band explicitly, or else signalled only for the LL sub-band. The former is known as expounded deadzone adjustment and the latter is known as derived deadzone adjustment. In the latter case, all the deadzone adjustment factors nz_b are derived implicitly from the single deadzone adjustment factor nz_0 corresponding to the LL band, according to:

$$nz_b = nz_0 \quad (\text{C-2})$$

C.2 Variable scalar dequantization for irreversible filters

For generalized scalar dequantization with irreversible filters the reconstructed values are computed as:

$$Rq_b(u, v) = \begin{cases} (\bar{q}_b(u, v) + r2^{M_b-N(u,v)} - nz_b)\Delta_b & \bar{q}_b(u, v) > 0 \\ (\bar{q}_b(u, v) - r2^{M_b-N(u,v)} - nz_b)\Delta_b & \bar{q}_b(u, v) < 0 \\ 0 & \bar{q}_b(u, v) = 0 \end{cases} \quad (\text{C-3})$$

where nz_b is the transmitted deadzone adjustment factor from the extended QCD, QCC, QPD, QPC marker segments and all other parameters are as described in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex E. If there is no extended QCD, QCC, QPD, QPC marker segment applicable to a component in the codestream, then $nz_b = 0$ for all sub-bands. When $nz_b = 0$ this formula is identical to the scalar dequantization used with irreversible filters in ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex E.

The variable scalar quantizer shall only be used with irreversible transforms.

¹⁾ The "[" means the first value is included and the ")" means the last value is excluded.

C.3 Variable scalar quantization for irreversible filters (informative)

The quantized coefficients, $q_b(u, v)$, are computed from the unquantized coefficients, $a_b(u, v)$, by:

$$q_b(u, v) = \begin{cases} \text{sign}(a_b(u, v)) \left\lfloor \frac{|a_b(u, v)| + nz_b \Delta_b}{\Delta_b} \right\rfloor & |a_b(u, v)| \geq -nz_b \Delta_b \\ 0 & |a_b(u, v)| < -nz_b \Delta_b \end{cases} \quad (\text{C-4})$$

where Δ_b is the quantization step size included in the QCD, QCC, QPD, QPC marker segments of ITU-T Rec. T.800 | ISO/IEC 15444-1 Annex A, and nz_b is a deadzone adjustment parameter included in the extended QCD, QCC, QPD, QPC marker segments. If $nz_b = 0$ for all sub-bands, then it need not be signalled. If nz_b is identical for all sub-bands, then the derived signalling may be used in the extended QCD, QCC, QPD, QPC marker segments.

Annex D

Trellis coded quantization extensions

(This annex forms an integral part of this Recommendation | International Standard)

In this annex and all of its subclauses, the flow charts and tables are normative only in the sense that they are defining an output that alternative implementations shall duplicate. This annex describes an extension to ITU-T Rec. T.800 | ISO/IEC 15444-1 that can be used alone or in conjunction with any of the other extensions in this Recommendation | International Standard. The capabilities of the codestream are defined by the SIZ marker segment parameter Rsiz (see A.2.1).

This annex specifies the trellis coded quantization (TCQ) option for encoding and reconstructing a sequence of wavelet coefficients. TCQ shall only be used with irreversible transformations.

D.1 Introduction to TCQ

The TCQ algorithm applies spatial-varying scalar quantization to its input sequence by choosing one of four scalar quantizers for each sample. Quantizer indices from supersets of these quantizers along with quantizer transitions in the form of a trellis provide all information necessary to reconstruct TCQ encoded wavelet coefficients.

Figure D.1 depicts the four separate scalar quantizers (D_0 , D_1 , D_2 , and D_3) used for this Recommendation | International Standard. Included with this figure is information regarding scalar quantized indices (m_{D_i}), reconstruction levels ($R_{D_i}(m_{D_i})$), and eventual union quantizer indices ($Q_{D_i}(m_{D_i})$) for each scalar quantizer.

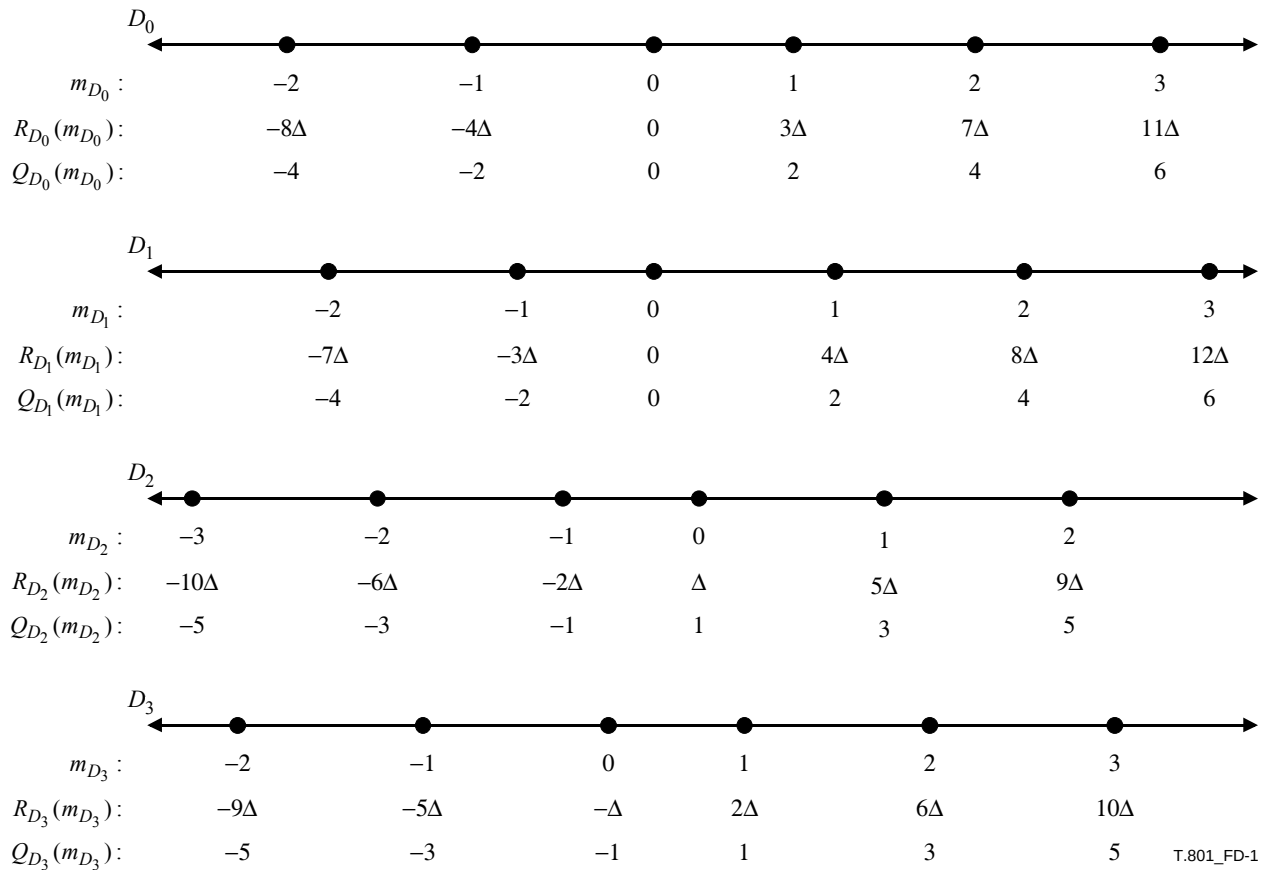


Figure D.1 – Scalar quantizers used for TCQ

Figure D.2 shows the combination of these scalar quantizers into union quantizers, A_0 and A_1 , along with the indices available with each quantizer and the corresponding reconstruction levels ($R_{A_i}(m_{A_i})$).

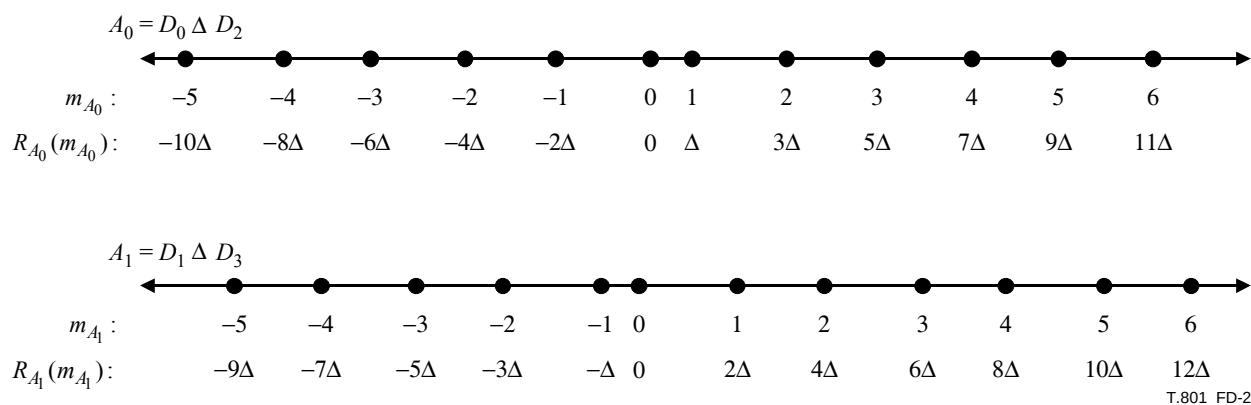


Figure D.2 – Union quantizers for TCQ

The eight state trellis directed graph with possible quantizer transitions is shown in Figure D.3, flowing left to right, where each node represents a possible trellis state. Columns of nodes represent stages which are ordered from left to right. There are exactly $K+1$ stages if K data points are quantized. Each node in Figure D.3 is labelled as $N_{k,s}$, where k corresponds to the stage index for the node and s is the trellis state of the node.

Trellis State:

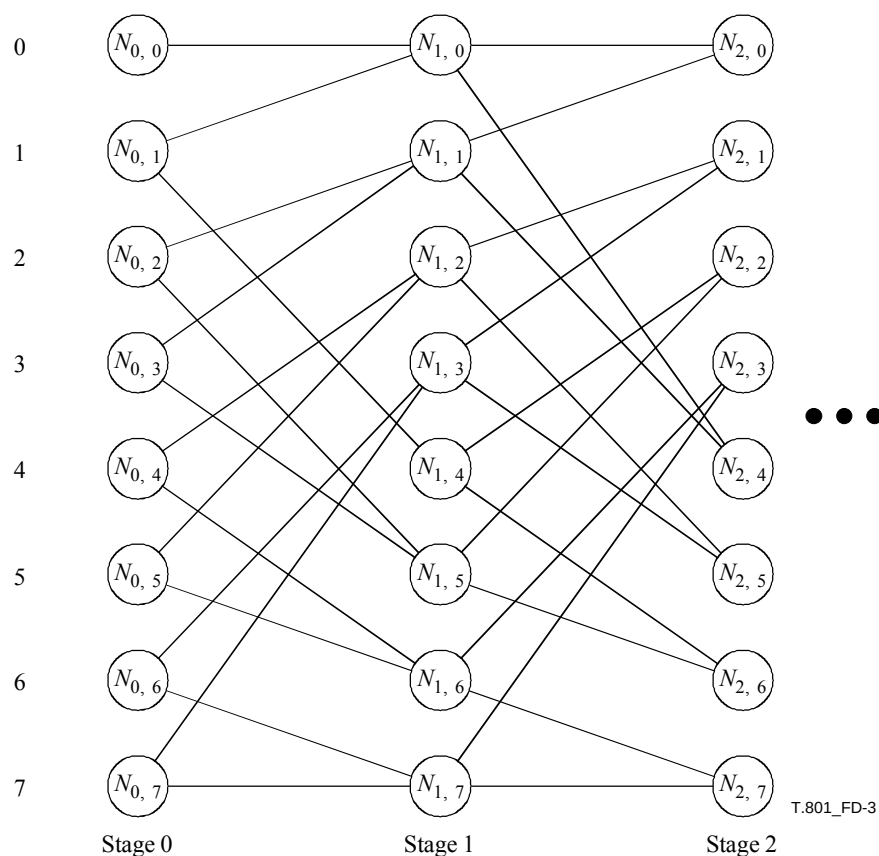


Figure D.3 – Trellis showing node indices