INTERNATIONAL STANDARD

ISO/IEC 15444-15

First edition
2019-10

Information technology — JPEG 2000 image coding system —

Part 15:
High-Throughput JPEG 2000

Technologies de l'information — Système de codage d'images JPEG 2000 —

Partie 15: JPEG 2000 à haut débit

Reference number
ISO/IEC 15444-15:2019(E)

Foreword

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

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

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

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

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

This document was prepared by ITU-T as ITU-T T.814 (06/2019) and drafted in accordance with its editorial rules. It was assigned to Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 15444 series can be found on the ISO website.

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

INTERNATIONAL STANDARD ISO/IEC 15444-15
RECOMMENDATION ITU-T T.814

Information technology – JPEG 2000 image coding system: High-throughput JPEG 2000

Summary

The computational complexity of the block-coding algorithm of Rec. ITU-T T.800 | ISO/IEC 15444-1 can be a challenge in some applications.

Rec. ITU-T T.814 | ISO/IEC 15444-15 specifies a high-throughput (HT) block-coding algorithm that can be used in place of the block-coding algorithm specified in Rec. ITU-T T.800 | ISO/IEC 15444-1.

The HT block-coding algorithm increases decoding and encoding throughput and allows mathematically lossless transcoding to and from the block-coding algorithm specified in Rec. ITU-T T.800 | ISO/IEC 15444-1. This is achieved at the expense of some loss in coding efficiency and substantial elimination of quality scalability.

The HT block-coding algorithm adopts a coding pass structure like that of the block-coding algorithm of Rec. ITU-T T.800 | ISO/IEC 15444-1. No more than three coding passes are required for any given code-block in the final codestream, and arithmetic coding is replaced with a combination of variable length coding tools, adaptive run-length coding and simple bit-packing. The algorithm involves three passes: a significance propagation pass (HT SigProp coding pass), a magnitude refinement pass (HT MagRef coding pass) and a cleanup pass (HT cleanup coding pass).

The HT MagRef coding pass is identical to that of the block-coding algorithm of Rec. ITU-T T.800 | ISO/IEC 15444-1, operating in the bypass mode, except that code bits are packed into bytes with a little-endian bit order. That is, the first code bit in a byte appears in its LSB, as opposed to its MSB.

The HT SigProp coding pass is also very similar to that of the block-coding algorithm of Rec. ITU-T T.800 | ISO/IEC 15444-1, operating in the BYPASS mode, with the following two differences:

- code bits are again packed into bytes of the raw bit-stream with a little-endian bit order, instead of big-endian bit packing order; and
- the significance bits associated with a set of four stripe columns are emitted first, followed by the associated sign bits, before advancing to the next set of stripe columns, instead of inserting any required sign bit immediately after the same sample's magnitude bit.

The HT cleanup coding pass is, however, significantly different from that of the block-coding algorithm of Rec. ITU-T T.800 | ISO/IEC 15444-1, and most of ITU-T T.814 | ISO/IEC 15444-15 is devoted to its description.

Aside from the block-coding algorithm itself and the parsing of packet headers, the HT block-coding algorithm preserves the syntax and semantics of other parts of the codestream specified in Rec. ITU-T T.800 | ISO/IEC 15444-1.

Recommendation ITU-T T.814 (2019) is a common text with ISO/IEC 15444-15:2019, both in their first edition.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T T.814	2019-06-13	16	11.1002/1000/13912

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Page

1 Scope ... 1

2 Normative references ... 1

3 Terms and definitions ... 1

4 Abbreviations and symbols ... 1

5 Convention ... 2

6 Conformance ... 2
- 6.1 HTJ2K codestream ... 2
- 6.2 HTJ2K decoding algorithm ... 2
- 6.3 JPH file ... 3

7 HT block decoding algorithm ... 3
- 7.1 Retrieving bit-streams from HT segments ... 3
- 7.2 Quad-based scanning pattern ... 9
- 7.3 HT Cleanup decoding algorithm ... 10
- 7.4 HT SigProp decoding procedure ... 20
- 7.5 HT MagRef decoding procedure ... 22
- 7.6 Sample output values ... 22

8 Constrained codestream sets ... 23
- 8.1 Overview ... 23
- 8.2 HTONLY, HTDECLARED and MIXED sets ... 23
- 8.3 SINGLEHT and MULTIHT sets ... 23
- 8.4 RGN and RGNFREE sets ... 23
- 8.5 HOMOGENEOUS and HETEROGENEOUS sets ... 23
- 8.6 LOCAL and FRAG sets ... 24
- 8.7 Bounded magnitude sets ... 24
- 8.8 CPF_N sets ... 25

9 Media types ... 26

Annex A (normative) – HTJ2K codestream syntax ... 27
- A.1 General ... 27
- A.2 SIZ marker segment ... 27
- A.3 CAP marker segment ... 27
- A.4 COD and COC marker segments ... 29
- A.5 RGN marker segment ... 30
- A.6 Corresponding Profile (CPF) marker segment ... 30

Annex B (normative) – HT data organization ... 32
- B.1 HT Sets ... 32
- B.2 HT segments ... 32
- B.3 Packets, Z_blk and S_blk ... 32

Annex C (normative) – CxtVLC tables ... 34

Annex D (normative) – JPH file format ... 54
- D.1 General ... 54
- D.2 JP2 Header box ... 54
- D.3 File Type box ... 54
- D.4 Colour Specification box ... 54
- D.5 Contiguous Codestream box ... 55
- D.6 Channel Definition box ... 55

Annex E (normative) – Media type specifications and registrations ... 57
- E.1 General ... 57
- E.2 JPH file ... 57
- E.3 Single HTJ2K codestream ... 57

Annex F (informative) – HT block encoding procedures ... 59
- F.1 Overview ... 59

F.2 Bit-planes, exponents, MagSgn values and EMB patterns ... 61
F.3 Cleanup pass encoding steps ... 62
F.4 Bit-stuffing and byte-stream termination procedures ... 65

Bibliography ... 72

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – JPEG 2000 image coding system: High-throughput JPEG 2000

1 Scope

This Recommendation | International Standard specifies an alternate block-coding algorithm that can be used in place of the block-coding algorithm specified in Rec. ITU-T T.800 | ISO/IEC 15444-1. This alternate block-coding algorithm offers a significant increase in throughput at the expense of slightly reduced coding efficiency, while a) allowing mathematically lossless transcoding to and from codestreams that use the block-coding algorithm specified in Rec. ITU-T T.800 | ISO/IEC 15444-1, and b) preserving codestream syntax and features specified in Rec. ITU-T T.800 | ISO/IEC 15444-1.

Recommendation ITU-T T.814 (2019) is a common text with ISO/IEC 15444-15:2019, both in their first edition.

2 Normative 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

– Recommendation ITU-T T.800 (2019) | ISO/IEC 15444-1:2019, *Information technology – JPEG 2000 image coding system: Core coding system*.

2.2 Paired Recommendations | International Standards equivalent in technical content

– Recommendation ITU-T H.273 (2016), *Coding-independent code points for video signal type identification*.

– ISO/IEC 23001-8:2016, *Information technology – MPEG systems technologies – Part 8: Coding-independent code points*.

2.3 Additional references

– ISO/IEC 15076-1, *Image technology colour management – Architecture, profile format and data structure – Part 1: Based on ICC.1:2010*.

3 Terms and definitions

For the purposes of this Recommendation | International Standard, the terms and definitions given in Rec. ITU-T T.800 | ISO/IEC 15444-1 apply.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the abbreviations and symbols defined in Rec. ITU-T T.800 | ISO/IEC 15444-1 and the following apply.

AZC All Zero Context

CUP Cleanup coding Pass

CPF Corresponding Profile

CxtVLC Context adaptive Variable Length Code

EMB Exponent Max Bound

FRAG Fragmented

HT High-Throughput

HTJ2K High-Throughput JPEG 2000

HTIRV	High-Throughput Irreversible
HTREV	High-Throughput Reversible
LSB	Least-Significant Bit
MAGB	Magnitude Bound
MagRef	Magnitude Refinement
MagSgn	Magnitude and Sign
MRP	MagRef coding pass
MSB	Most-Significant Bit
SigProp	Significance Propagation
U-VLC	Unsigned residual VLC
VLC	Variable Length Coding

5 Conventions and symbols

For the purposes of this Recommendation | International Standard, the symbols defined in Rec. ITU-T T.800 | ISO/IEC 15444-1 and the following apply:

Dcup[n]	Byte n of an HT Cleanup segment
Dref[n]	Byte n of an HT Refinement segment
Hblk	Height of a code-block, measured in samples
Lcup	Length in bytes of HT Cleanup segment
Lref	Length in bytes of HT Refinement segment
MEL	Adaptive run-length coding algorithm
MEL_E	MEL Exponent Table
Pcup	HT Cleanup segment prefix length
QH	Height of a code-block, measured in quads
QW	Width of a code-block, measured in quads
Scup	HT Cleanup segment suffix length
SPP	HT SigProp coding Pass
u_ext	U-VLC extension component
u_pfx	U-VLC prefix component
u_sfx	U-VLC suffix component
Wblk	Width of a code-block, measured in samples
Z_blk	Number of passes that can be processed within an HT Set

6 Conformance

6.1 HTJ2K codestream

A high-throughput JPEG 2000 (HTJ2K) codestream shall conform to Annex A.

6.2 HTJ2K decoding algorithm

The HTJ2K decoding algorithm processes an HTJ2K codestream as specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 together with any additional signalled capability, with the exception of HT code-blocks, as defined in Annex B, in which case the following applies:

- the HT code-blocks are processed according to clause 7; and

- the resulting number of magnitude bits N_b, the magnitude bits $\text{MSB}_i(b)$ and the sign bits s_b are processed according to Rec. ITU-T T.800 | ISO/IEC 15444-1, together with any additional signalled capability.

NOTE 1 – If the two most significant bits of Ccap[15] are 0 for a codestream, all code-blocks are HT code-blocks and the decoding procedures defined in Annexes C and D of Rec. ITU-T T.800 | ISO/IEC 15444-1 are not used.

NOTE 2 – The processing of HT code-blocks specified herein is compatible with the additional capabilities specified in Rec. ITU-T T.801 | ISO/IEC 15444-2.

NOTE 3 – The symbols N_b, $MSB_i(b)$ and s_b are defined in Rec. ITU-T T.800 | ISO/IEC 15444-1.

6.3 JPH file

A JPH file shall conform to Annex D.

7 HT block-decoding algorithm

7.1 Retrieving bit-streams from HT segments

7.1.1 General

This clause specifies the process for extracting bit-streams from an HT set and its associated parameters `Z_blk` and `S_blk`, as defined in Annex B.

If `Z_blk` equals `0`, no HT segments are available for the code-block, and so all sample output values for the block shall be 0.

There are at most two HT segments available to the HT block-decoding algorithm:

- The HT cleanup segment holds the coded bytes belonging to the HT cleanup coding pass (CUP);
- The HT refinement segment holds the coded bytes belonging to the HT significance propagation (SigProp) coding pass and, optionally, an HT magnitude refinement (MagRef) coding pass. The HT refinement segment is available if and only if `Z_blk` is greater than 1, while an HT MagRef coding pass is available if and only if `Z_blk` is equal to 3.

NOTE 1 – Multiple sets of HT cleanup and HT refinement segments can be found within the codestream for a given code-block, but the decoding procedure described here processes only `Z_blk` coding passes, whose coded bytes are found within one HT cleanup segment and, if `Z_blk` is greater than 1, the one HT refinement segment that follows this HT cleanup segment.

As illustrated in Figure 1, the HT segments are comprised of byte-streams, each an ordered sequence of bytes. From each byte-stream, a bit-stream, which is an ordered sequence of bits, can be unpacked as follows:

- The magnitude and sign (MagSgn) bit-stream is recovered from the MagSgn byte-stream, which extends forward from byte 0 of the HT cleanup segment for a total of `Pcup` bytes, with prefix length, `Pcup = Lcup – Scup`; where `Lcup` is the length (in bytes) of the HT cleanup segment, and `Scup` is a suffix length.
- The adaptive run-length coding algorithm (MEL) bit-stream is recovered from the MEL byte-stream, which extends forward from byte `Pcup` of the HT cleanup segment, for at most `Scup` bytes.
- The variable length coding (VLC) bit-stream is recovered from the VLC byte-stream, which extends backward from the last byte of the HT cleanup segment, for at most `Scup` bytes. The VLC and MEL byte-streams may overlap.
- If `Z_blk` is greater than 1, the SigProp bit-stream is recovered from the SigProp byte-stream, which extends forwards from byte 0 of the HT refinement segment, for at most `Lref` bytes, where `Lref` is the length of the HT refinement segment.
- If `Z_blk` is equal to 3, the MagRef bit-stream is recovered from the MagRef byte-stream, which extends backwards from the end (byte `Lref-1`) of the HT refinement segment, for at most `Lref` bytes. The MagRef and SigProp byte-streams may overlap.

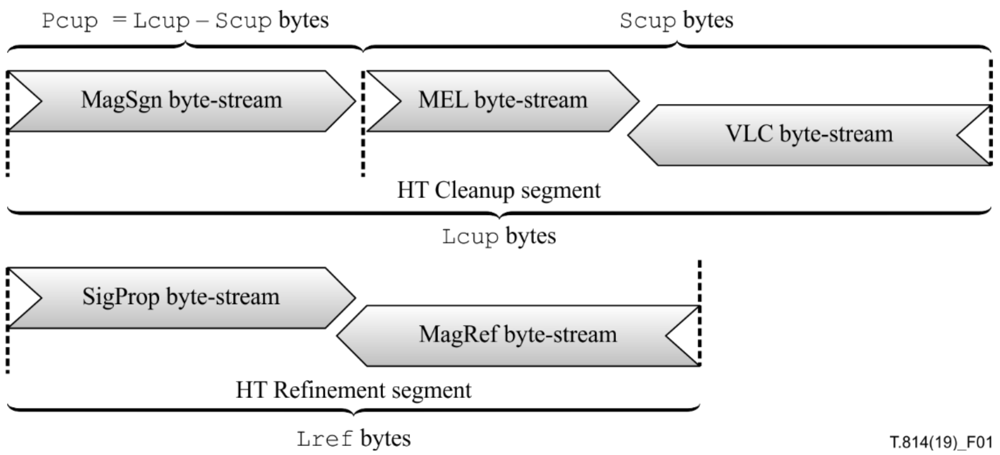

Figure 1 – HT segments and their byte-streams

The HT cleanup segment:

- shall have length `Lcup` such that $2 \leq$ `Lcup` < 65535;
- shall not contain any consecutive pair of bytes whose value, as a big-endian 16-bit unsigned integer, exceeds `0xFF8F`;
- shall not terminate with a byte whose value is `0xFF`.

The HT refinement segment:

- shall have length `Lref` satisfying $0 \leq$ `Lref` < 2047;
- shall also contain no consecutive pair of bytes whose value, as a big-endian 16-bit unsigned integer, exceeds `0xFF8F`;
- shall not terminate with a byte whose value is `0xFF`.

The suffix length `Scup` is found from the last two bytes of the HT cleanup segment as follows:

`Scup = (16 × Dcup[Lcup-1]) + (Dcup[Lcup-2] & 0x0F)`

where `Dcup[n]` denotes byte `n` of the HT cleanup segment, and where `n` takes value from `0` to `Lcup-1`.

After `Scup` is recovered from its last two bytes, `Dcup[n]` is accessed using the following procedure:

```
Procedure: modDcup
Returns: Modified Dcup array
State: Dcup, pos


if (pos == Lcup - 1)
  return 0xFF
else if (pos == Lcup - 2)
  return Dcup[pos] | 0x0F
else
  return Dcup[pos]
```

NOTE 2 – This procedure overwrites the last byte and the four least-significant bits (LSBs) of the second-last byte of the HT cleanup segment with 1s.

The value of `Scup` obtained in this way, shall satisfy:

$$2 \leq \texttt{Scup} \leq \min(\texttt{Lcup}, 4079)$$

Furthermore, the codestream shall be constructed such that, if `Scup < Lcup`, so that `Pcup > 0`, byte `Pcup-1` of the HT cleanup segment shall not have the value `0xFF`.

NOTE 3 – The `importMagSgnBit` procedure in clause 7.1.2 effectively synthesizes a byte equal to `0xFF` to replace any byte equal to `0xFF` that might have been discarded during encoding to satisfy this constraint.

Details of the procedures to be used in recovering each bit-stream from its respective byte-stream are provided in clauses 7.1.2 to 7.1.6.

Similar to the HT cleanup segment, `Dref[n]` denotes byte `n` of the HT refinement segment, where `n` takes values from `0` to `Lref-1`, except that no modification is made to the bytes of the HT refinement segment.

The procedure `error()` denotes a state resulting from a codestream that does not conform to this specification, and for which behaviour is undefined.

7.1.2 Magsgn bit-stream recovery

HT MagSgn bits are retrieved from the HT MagSgn byte-stream, as required by other elements of the decoding procedure, using the `importMagSgnBit` procedure in the following. This procedure is part of a state machine with state variables `MS_pos`, `MS_bits`, `MS_tmp` and `MS_last` that are initialized using the `initMS` procedure, prior to first use of the `importMagSgnBit` procedure for an HT code-block.

```
Procedure: initMS
State: MS_pos, MS_bits, MS_tmp, MS_last


MS_pos = 0
MS_bits = 0
MS_tmp = 0
MS_last = 0
```

```
Procedure: importMagSgnBit
Returns: next MagSgn bit
State: MS_pos, MS_bits, MS_tmp, MS_last


if (MS_bits == 0)
    MS_bits = (MS_last == 0xFF)? 7 : 8
    if (MS_pos < Pcup)
        MS_tmp = modDcup(Dcup, MS_pos)
        if ((MS_tmp & (1<<MS_bits)) != 0)
            error()
    else if (MS_pos == Pcup)
        MS_tmp = 0xFF
    else
        error()
    MS_last = MS_tmp
    MS_pos = MS_pos + 1
bit = MS_tmp & 1
MS_tmp = MS_tmp >> 1
MS_bits = MS_bits - 1
return bit
```


NOTE 1 – These procedures effectively unpack bits from the HT MagSgn byte-stream in little-endian order, skipping over stuffing bits that appear in the MSB position of any byte that follows a byte equal to `0xFF`.

NOTE 2 – The value of `Pcup` can be as small as 0.

NOTE 3 – The procedure in the foregoing effectively appends at most one byte equal to `0xFF` to the HT MagSgn byte-stream, which is sufficient to allow recovery of all required HT MagSgn bits if the codestream conforms to this Specification.

NOTE 4 – The `importMagSgnBit` procedure is designed such that the MSB of a byte that follows a byte equal to `0xFF` is 0, unless that byte does not contribute to the MagSgn bit-stream. This is intended to simplify decoder implementations.

7.1.3 MEL bit-stream recovery

MEL bits are retrieved from the MEL byte-stream, as required by other elements of the decoding procedure, using the `importMELBit` procedure in the following. This procedure is part of a state machine with state variables `MEL_pos`, `MEL_bits`, `MEL_tmp` that are initialized using the `initMEL` procedure, prior to first use of the `importMELBit` procedure for an HT code-block.

```
Procedure: initMEL
State: MEL_pos, MEL_bits, MEL_tmp


MEL_pos = Pcup
MEL_bits = 0
MEL_tmp = 0
```

```
Procedure: importMELBit
Returns: next MEL bit
State: MEL_pos, MEL_bits, MEL_tmp


if (MEL_bits == 0)
    MEL_bits = (MEL_tmp == 0xFF) ? 7 : 8
    if (MEL_pos < Lcup)
        MEL_tmp = modDcup(Dcup,MEL_pos)
        MEL_pos = MEL_pos + 1
    else
        MEL_tmp = 0xFF
MEL_bits = MEL_bits - 1
bit = (MEL_tmp >> MEL_bits) & 1
return bit
```


NOTE – These procedures effectively unpack bits from the MEL byte-stream in big-endian order, skipping over stuffing bits that appear in the MSB position of any byte that follows a byte equal to `0xFF`.

7.1.4 HT VLC bit-stream recovery

HT VLC bits are retrieved from the HT VLC byte-stream, as required by other elements of the decoding procedure, using the `importVLCBit` procedure in the following. This procedure is part of a state machine with state variables `VLC_pos`, `VLC_bits`, `VLC_tmp` and `VLC_last` that are initialized using the `initVLC` procedure in the following, prior to first use of the `importVLCBit` procedure for an HT code-block.

```
Procedure: initVLC
State: VLC_pos, VLC_bits, VLC_tmp, VLC_last


VLC_pos = Lcup-3
VLC_last = modDcup(Dcup ,Lcup-2)
VLC_tmp = VLC_last >> 4
VLC_bits = ((VLC_tmp & 7) < 7)?4:3
```

```
Procedure: importVLCBit
Returns: next VLC bit
State: VLC_pos, VLC_bits, VLC_tmp, VLC_last


if (VLC_bits == 0)
    if (VLC_pos >= Pcup)
        VLC_tmp = modDcup(Dcup, VLC_pos)
    else
        error()
    VLC_bits = 8
    if (VLC_last > 0x8F) and ((VLC_tmp & 0x7F) == 0x7F)
        VLC_bits = 7
    VLC_last = VLC_tmp
    VLC_pos = VLC_pos - 1
bit = VLC_tmp & 1
VLC_tmp = VLC_tmp >> 1
VLC_bits = VLC_bits – 1
return bit
```

NOTE – These procedures effectively unpack bits from the HT VLC byte-stream in little-endian order, while consuming bytes in reverse order, skipping over stuffing bits that appear in the MSB position of any byte whose 7 LSBs are all 1s if the byte that was last consumed was larger than 0x8F, and also skipping over the 12 bits that were replaced with 1s after using them to find the Scup value.

7.1.5 HT SigProp bit-stream recovery

If Z_blk is greater than or equal to 2, HT SigProp bits are retrieved from the HT SigProp byte-stream, as required by other elements of the decoding procedure, using the importSigPropBit procedure in the following. This procedure is part of a state machine with state variables SP_pos, SP_bits, SP_tmp and SP_last that are initialized using the initSP procedure in the following, prior to first use of the importSigPropBit procedure for an HT code-block.

```
Procedure: initSP
State: SP_pos, SP_bits, SP_tmp, SP_last

SP_pos = 0
SP_bits = 0

SP_tmp = 0

SP_last = 0
```