**OPEN SOURCE SOFTWARE NOTICE**

The open source software notice for the open source software used in this openEuler distribution (“openEuler distribution”) is provided in this document. The open source software licenses are granted by the respective right holders. With regard to the respective open source software contained in the openEuler distribution, the applicable open source software license will prevail all other license agreement, including but not limited to the openEuler Software License, in case of any conflicts.

**Warranty Disclaimer**

**THE OPEN SOURCE SOFTWARE IN THIS OPENEULER DISTRIBUTION IS PROVIDED ON AN "AS IS" BASIS, WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.  
See the applicable licenses for more details.**

**Copyright Notice and License Texts**

Software: libtheora 1.1.1

**Copyright notice:**

copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

copyright notice, this list of conditions and the following disclaimer.

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors. 4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the NetBSD Foundation, Inc. and its contributors. 4. Neither the name of The NetBSD Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Copyright","Copyright (C) 2002-2007 Xiph.Org Foundation" VALUE "OriginalFilename",TH\_ENC\_INTERNAL\_NAME ".dll" VALUE "ProductName","libtheora" VALUE "ProductVersion",TH\_VERSION\_STRING END END BLOCK "VarFileInfo" BEGIN 0x0409, 1200 == US English, Unicode\*/ VALUE "Translation",0x0409,1200 END END

Copyright","Copyright (C) 2002-2007 Xiph.Org Foundation" VALUE "OriginalFilename",TH\_DEC\_INTERNAL\_NAME ".dll" VALUE "ProductName","libtheora" VALUE "ProductVersion",TH\_VERSION\_STRING END END BLOCK "VarFileInfo" BEGIN 0x0409, 1200 == US English, Unicode\*/ VALUE "Translation",0x0409,1200 END END

Copyright © 2004-2007 Xiph.org Foundation. All rights reserved. Ogg, Theora, Vorbis, Xiph.org Foundation and their logos are trademarks (�) of the Xiph.org Foundation. This document is set in LATEX.

Copyright © 2004 Scott James Remnant <scott@netsplit.com>.

Copyright © 2002-2009Xiph.Org Foundation</string> key>CSResourcesFileMapped</key> true/> dict> plist>

Copyright © 2002-2009Xiph.Org Foundation</string> key>CFBundleIconFile</key> string></string> key>CFBundleIdentifier</key> string>org.xiph.theora</string> key>CFBundleInfoDictionaryVersion</key> string>6.0</string> key>CFBundlePackageType</key> string>FMWK</string> key>CFBundleSignature</key> string>????</string> key>CFBundleVersion</key> string>1.0d6</string> key>CFBundleShortVersionString</key> string>1.1alpha1svn</string> key>NSHumanReadableCopyright</key>

Copyright \textcopyright{} 2004-2007 Xiph.org Foundation. All rights reserved. Ogg, Theora, Vorbis, Xiph.org Foundation and their logos are trademarks texttrademark) of the \href{http://www.xiph.org}{Xiph.org Foundation}.

Copyright Statement

Copyright Notice

Copyright 2002 Free Software Foundation, Inc.

Copyright 2001, 2002 Free Software Foundation, Inc.

Copyright 2001 Free Software Foundation, Inc. -\*- Autoconf -\*-

Copyright 2001 Free Software Foundation, Inc.

Copyright 1999, 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright 1999, 2000, 2001 Free Software Foundation, Inc.

Copyright 1999, 2000 Free Software Foundation, Inc. Written by Tom Tromey <tromey@cygnus.com>.

Copyright 1999, 2000 Free Software Foundation, Inc.

Copyright 1997, 2000, 2001 Free Software Foundation, Inc.

Copyright 1997, 1999, 2000, 2001 Free Software Foundation, Inc.

Copyright 1996, 1998, 2000, 2001 Free Software Foundation, Inc.

Copyright 1996, 1997, 2000, 2001 Free Software Foundation, Inc.

Copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc. This file is free software; the Free Software Foundation gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

Copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc. This Makefile.in is free software; the Free Software Foundation gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

Copyright 1991 by the Massachusetts Institute of Technology

Copyright (c) 2000 The NetBSD Foundation, Inc. All rights reserved.

Copyright (c) 1987, 1993, 1994 The Regents of the University of California. All rights reserved.

Copyright (C) Thomas Östreich - June 2001

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Copyright (C) The Internet Society (2006).

Copyright (C) QNX Software Systems Ltd. 1995 All rights reserved. Please send any comments/suggetions to: latex@qnx.com

Copyright (C) 2008 Free Software Foundation, Inc. This config.status script is free software; the Free Software Foundation gives unlimited permission to copy, distribute and modify it."

Copyright (C) 2008 Free Software Foundation, Inc. This config.lt script is free software; the Free Software Foundation gives unlimited permision to copy, distribute and modify it."

Copyright (C) 2004, 2005, 2007, 2008 Free Software Foundation, Inc. Written by Gary V. Vaughan, 2004

Copyright (C) 2004, 2005, 2007 Free Software Foundation, Inc. Written by Scott James Remnant, 2004.

Copyright (C) 2004, 2005, 2006, 2007 Free Software Foundation, Inc. Written by Gary V. Vaughan <gary@gnu.org>

Copyright (C) 2004 Free Software Foundation, Inc. Written by Scott James Remnant, 2004

Copyright (C) 2003 Commonwealth Scientific and Industrial Research Organisation (CSIRO) Australia

Copyright (C) 2002-2009 Xiph.org Foundation

Copyright (C) 2002 Thomas Östreich

Copyright (C) 1999-2006, 2007, 2008 Free Software Foundation, Inc. Written by Thomas Tanner, 1999

Copyright (C) 1999 Rainer Johanni <Rainer@Johanni.de>

Copyright (C) 1996, 1997, 1999, 2000, 2002 Free Software Foundation, Inc. Originally by Fran,cois Pinard <pinard@iro.umontreal.ca>, 1996.

Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004, 2005, 2006, 2007, 2008 Free Software Foundation, Inc. Written by Gordon Matzigkeit, 1996

Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004, 2005, 2006, 2007 2008 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Copyright (C) 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Free Software Foundation, Inc. This configure script is free software; the Free Software Foundation gives unlimited permission to copy, distribute and modify it. ACEOF exit fi cat >config.log <<\_ACEOF This file contains any messages produced by compilers while running configure, to aid debugging if configure makes a mistake.

Copyright (C) 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Free Software Foundation, Inc. This configure script is free software; the Free Software Foundation gives unlimited permission to copy, distribute and modify it.

Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001 Free Software Foundation, Inc.

Copyright (C) 1989-1994, 1996-1999, 2001 Free Software Foundation, Inc. This file is part of the GNU C Library.

Copyright (C) 1989,90,91,92,93,94,96,97,98 Free Software Foundation, Inc. This file is part of the GNU C Library.

Copyright (C) 1987,88,89,90,91,92,93,94,96,97,98 Free Software Foundation, Inc. This file is part of the GNU C Library.

Copyright (C) 1987,88,89,90,91,92,93,94,95,96,98,99,2000,2001 Free Software Foundation, Inc. This file is part of the GNU C Library.

Copyright (C) 1987, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99 Free Software Foundation, Inc.

COPYRIGHTED 0x00020000

COPYRIGHT:} Copyright statement.

COPYRIGHT: Copyright statement.

COPYRIGHT field for such a URL. item{ORGANIZATION:} Studio name, Publisher, or other organization involved in the creation of the video.

COPYRIGHT field for such a URL. ORGANIZATION: Studio name, Publisher, or other organization involved in the creation of the video. DIRECTOR: Director or Filmmaker credit, similar to ARTIST. PRODUCER: Producer credit for the video. COMPOSER: Music credit for the video. ACTOR: Acting credit for the video. TAG: subject or category tag, keyword, or other content classification labels. The value of each instance of this field SHOULD be treated as a single label, with multiple instances of the field for multiple tags. The value of a single field SHOULD NOT be parsed into multiple tags based on some internal delimeter. DESCRIPTION: General description, summary, or blurb.

COPYRIGHT (C) 2002-2009,2009 \* by the Xiph.Org Foundation and contributors http://www.xiph.org/ \*

COPYRIGHT (C) 2002-2009 \*</span> a name="l00009"></a>00009 <span class="comment"> \* by the Xiph.Org Foundation http://www.xiph.org/ \*</span> a name="l00010"></a>00010 <span class="comment"> \* \*</span> a name="l00011"></a>00011 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</span> a name="l00012"></a>00012 <span class="comment"></span> a name="l00013"></a>00013 <span class="comment"> function:</span> a name="l00014"></a>00014 <span class="comment"> last mod: $Id: theora.h,v 1.8 2004/03/15 22:17:32 derf Exp $</span> a name="l00015"></a>00015 <span class="comment"></span> a name="l00016"></a>00016 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/</span> a name="l00017"></a>00017 a name="l00062"></a>00062 <span class="preprocessor">#if !defined(\_O\_THEORA\_CODEC\_H\_)</span> a name="l00063"></a><

COPYRIGHT (C) 2002-2009 \*</span> a name="l00009"></a>00009 <span class="comment"> \* by the Xiph.Org Foundation http://www.xiph.org/ \*</span> a name="l00010"></a>00010 <span class="comment"> \* \*</span> a name="l00011"></a>00011 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</span> a name="l00012"></a>00012 <span class="comment"></span> a name="l00013"></a>00013 <span class="comment"> function:</span> a name="l00014"></a>00014 <span class="comment"> last mod: $Id: theora.h,v 1.8 2004/03/15 22:17:32 derf Exp $</span> a name="l00015"></a>00015 <span class="comment"></span> a name="l00016"></a>00016 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/</span> a name="l00017"></a>00017 a name="l00021"></a>00021 <span class="preprocessor">#if !defined(\_O\_THEORA\_THEORAENC\_H\_)</span> a name="l00022"><

COPYRIGHT (C) 2002-2009 \*</span> a name="l00009"></a>00009 <span class="comment"> \* by the Xiph.Org Foundation http://www.xiph.org/ \*</span> a name="l00010"></a>00010 <span class="comment"> \* \*</span> a name="l00011"></a>00011 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</span> a name="l00012"></a>00012 <span class="comment"></span> a name="l00013"></a>00013 <span class="comment"> function:</span> a name="l00014"></a>00014 <span class="comment"> last mod: $Id: theora.h,v 1.8 2004/03/15 22:17:32 derf Exp $</span> a name="l00015"></a>00015 <span class="comment"></span> a name="l00016"></a>00016 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/</span> a name="l00017"></a>00017 a name="l00021"></a>00021 <span class="preprocessor">#if !defined(\_O\_THEORA\_THEORADEC\_H\_)</span> a name="l00022"><

COPYRIGHT (C) 2002-2009 \*</span> a name="l00009"></a>00009 <span class="comment"> \* by the Xiph.Org Foundation http://www.xiph.org/ \*</span> a name="l00010"></a>00010 <span class="comment"> \* \*</span> a name="l00011"></a>00011 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</span> a name="l00012"></a>00012 <span class="comment"></span> a name="l00013"></a>00013 <span class="comment"> function:</span> a name="l00014"></a>00014 <span class="comment"> last mod: $Id: theora.h,v 1.17 2003/12/06 18:06:19 arc Exp $</span> a name="l00015"></a>00015 <span class="comment"></span> a name="l00016"></a>00016 <span class="comment"> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/</span> a name="l00017"></a>00017 a name="l00018"></a>00018 <span class="preprocessor">#ifndef \_O\_THEORA\_H\_</span> a name="l00019"></a>00019 <span c

COPYRIGHT (C) 2002-2009 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT (C) 2002-2009 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT (C) 2002-2009 \* by the Xiph.Org Foundation and contributors http://www.xiph.org/ \*

COPYRIGHT (C) 2002-2007 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT (C) 2002-2004 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT (C) 1999-2006 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT (C) 1999-2006 \* by the Xiph.Org Foundation http://www.xiph.org/ \*

COPYRIGHT "Copyright (C) 2005. Some Rights Reserved." define LICENSE "Creative Commons Attribution-ShareAlike 2.5"

(c) Using FTYPE, NMBS, NBS, and BCODED, decode the macro block coding modes into MBMODES using the procedure given in Section 7.4. d) If FTYPE is non-zero (inter frame), using PF, NMBS, MBMODES, NBS, and BCODED, decode the motion vectors into MVECTS using the procedure given in Section 7.5.1. e) Using NBS, BCODED, and NQIS, decode the block-level qi values into QIIS using the procedure given in Section 7.6. f) Using NBS, NMBS, BCODED, and HTS, decode the DCT coefficients into NCOEFFS and NCOEFFS using the procedure given in Section 7.7.3. g) Using BCODED and MBMODES, undo the DC prediction on the DC coefficients stored in COEFFS using the procedure given in Section 7.8.2. 2. Otherwise:

(c) Truncate C to a 16-bit representation by dropping any higher-order bits. d) Assign DQC[ci ] the value C.

(c) Set the string COMMENTS[ci ] to the contents of these octets.

(c) Read a 1-bit unsigned integer as MAG.

(c) Read a 1-bit unsigned integer as ISLEAF. d) If ISLEAF is one: i. If the number of entries in table HTS[hti ] is already 32, stop. The stream is undecodable. ii. Read a 5-bit unsigned integer as TOKEN. iii. Add the pair (HBITS, TOKEN) to Huffman table HTS[hti ]. e) Otherwise: i. Add a ‘0’ to the end of HBITS. ii. Decode the ‘0’ sub-tree using this procedure, starting from step 1b.

(c) Otherwise, if MSCHEME is not 7, assign the entries of MALPHABET the values in the corresponding column of Table 7.19. d) For each consecutive macro block in coded order (cf. Section 2.4)— indexed by mbi : i. If a block bi in the luma plane of macro block mbi exists such that BCODED[bi ] is 1: A. If MSCHEME is not 7, read one bit at a time until one of the Huffman codes in Table 7.19 is recognized, and assign MBMODES[mbi ] the value MALPHABET[mi ], where mi is the index of the Huffman code decoded. B. Otherwise, read a 3-bit unsigned integer as MBMODES[mbi ]. ii. Otherwise, if no luma-plane blocks in the macro block are coded, assign MBMODES[mbi ] the value 0 (INTER NOMV).

(c) Otherwise, if MBMODES[mbi ] is 4 (INTER MV LAST2): i. Assign (MVX, MVY) the value LAST2. ii. Assign LAST2 the value LAST1. iii. Assign LAST1 the value (MVX, MVY). d) Otherwise, if MBMODES[mbi ] is 3 (INTER MV LAST), assign MVX, MVY) the value LAST1. e) Otherwise, if MBMODES[mbi ] is 2 (INTER MV): i. Decode a single motion vector into MVX and MVY using the procedure described in Section 7.5.1. ii. Assign LAST2 the value LAST1. iii. Assign LAST1 the value (MVX, MVY). f) Otherwise (MBMODES[mbi ] is 5: INTER GOLDEN NOMV, 1: INTRA, or 0: INTER NOMV), assign MVX and MVY the value zero. g) If MBMODES[mbi ] is not 7 (not INTER MV FOUR), then for each coded block bi in macro block mbi : i. Assign MVECTS[bi ] the value (MVX, MVY).

(c) Otherwise, assign P[3] the value zero. 10. Otherwise, assign P[3] the value zero. 11. If none of the values P[0], P[1], P[2], nor P[3] are non-zero, then assign DCPRED the value LASTDC[rfi ]. 12. Otherwise: a) Assign the array W and the variable PDIV the values from the row of Table 7.47 corresonding to the values of each P[i ].

(c) Otherwise, assign P[2] the value zero. 8. Otherwise, assign P[2] the value zero. 9. If block bi is not along the right edge nor the bottom edge of the coded frame: a) Assign bj the coded-order index of block bi ’s lower-right neighbor, i.e., one row down and one column to the right. b) If BCODED[bj ] is not zero: i. Assign mbj the index of the macro block containing block bj . ii. If the value of the Reference Frame Index column of Table 7.46 corresonding to MBMODES[mbj ] equals rfi : A. Assign P[3] the value 1. B. Assign PBI[3] the value bj . iii. Otherwise, assign P[3] the value zero.

(c) Otherwise, assign P[1] the value zero. 6. Otherwise, assign P[1] the value zero. 7. If block bi is not along the the bottom edge of the coded frame: a) Assign bj the coded-order index of block bi ’s lower neighbor, i.e., in the same column but one row down. b) If BCODED[bj ] is not zero: i. Assign mbj the index of the macro block containing block bj . ii. If the value of the Reference Frame Index column of Table 7.46 corresonding to MBMODES[mbj ] equals rfi : A. Assign P[2] the value 1. B. Assign PBI[2] the value bj . iii. Otherwise, assign P[2] the value zero.

(c) Otherwise, assign P[0] the value zero. 4. Otherwise, assign P[0] the value zero. 5. If block bi is not along the left edge nor the bottom edge of the coded frame: a) Assign bj the coded-order index of block bi ’s lower-left neighbor, i.e., one row down and one column to the left. b) If BCODED[bj ] is not zero: i. Assign mbj the index of the macro block containing block bj . ii. If the value of the Reference Frame Index column of Table 7.46 corresonding to MBMODES[mbj ] equals rfi :

(c) Otherwise, assign COEFFS[bi ][ti ] the value −6. d) Assign TIS[bi ] the value TIS[bi ] + 1. e) Assign NCOEFFS[bi ] the value TIS[bi ]. 11. Otherwise, if TOKEN is 17: a) Read a 1-bit unsigned integer as SIGN. b) Read a 1-bit unsigned integer as MAG.

(c) Otherwise, assign COEFFS[bi ][ti ] the value −5. d) Assign TIS[bi ] the value TIS[bi ] + 1. e) Assign NCOEFFS[bi ] the value TIS[bi ]. 10. Otherwise, if TOKEN is 16: a) Read a 1-bit unsigned integer as SIGN. b) If SIGN is zero, assign COEFFS[bi ][ti ] the value 6.

(c) Otherwise, assign COEFFS[bi ][ti ] the value −4. d) Assign TIS[bi ] the value TIS[bi ] + 1. e) Assign NCOEFFS[bi ] the value TIS[bi ]. 9. Otherwise, if TOKEN is 15: a) Read a 1-bit unsigned integer as SIGN. b) If SIGN is zero, assign COEFFS[bi ][ti ] the value 5.

(c) Otherwise, assign COEFFS[bi ][ti ] the value −3. d) Assign TIS[bi ] the value TIS[bi ] + 1. e) Assign NCOEFFS[bi ] the value TIS[bi ]. 8. Otherwise, if TOKEN is 14: a) Read a 1-bit unsigned integer as SIGN. b) If SIGN is zero, assign COEFFS[bi ][ti ] the value 4.

(c) If ry1 is less than zero, assign ry1 the value zero.

(c) If ry is less than zero, assign ry the value zero. d) For each value of bx from 0 to 7, inclusive:

(c) If ci equals zero, assign QSCALE the value DCSCALE[qi ]. d) Else, assign QSCALE the value ACSCALE[qi ]. e) Assign QMAT[ci ] the value max(QMIN, min((QSCALE ∗ BM[ci ]//100) ∗ 4, 4096)).

(c) If SIGN is zero, assign COEFFS[bi ][ti + 5] the value 1. d) Otherwise, assign COEFFS[bi ][ti + 5] the value −1. e) Assign TIS[bi ] the value TIS[bi ] + 6. f) Assign NCOEFFS[bi ] the value TIS[bi ]. 22. Otherwise, if TOKEN is 28: a) Read a 1-bit unsigned integer as SIGN. b) Read a 2-bit unsigned integer as RLEN.

(c) If SIGN is zero, assign COEFFS[bi ][ti + 4] the value 1. d) Otherwise, assign COEFFS[bi ][ti + 4] the value −1.

(c) If SIGN is zero, assign COEFFS[bi ][ti + 3] the value 1. d) Otherwise, assign COEFFS[bi ][ti + 3] the value −1. e) Assign TIS[bi ] the value TIS[bi ] + 4. f) Assign NCOEFFS[bi ] the value TIS[bi ]. 20. Otherwise, if TOKEN is 26: a) For each value of tj from ti to (ti + 3), assign COEFFS[bi ][tj ] the value zero. b) Read a 1-bit unsigned integer as SIGN.

(c) If SIGN is zero, assign COEFFS[bi ][ti + 2] the value 1. d) Otherwise, assign COEFFS[bi ][ti + 2] the value −1. e) Assign TIS[bi ] the value TIS[bi ] + 3. f) Assign NCOEFFS[bi ] the value TIS[bi ]. 19. Otherwise, if TOKEN is 25: a) For each value of tj from ti to (ti + 2), assign COEFFS[bi ][tj ] the value zero. b) Read a 1-bit unsigned integer as SIGN.

(c) If SIGN is zero, assign COEFFS[bi ][ti + 1] the value 1. d) Otherwise, assign COEFFS[bi ][ti + 1] the value −1. e) Assign TIS[bi ] the value TIS[bi ] + 2. f) Assign NCOEFFS[bi ] the value TIS[bi ]. 18. Otherwise, if TOKEN is 24: a) For each value of tj from ti to (ti + 1), assign COEFFS[bi ][tj ] the value zero. b) Read a 1-bit unsigned integer as SIGN.

(c) If P[0] is non-zero, assign DCPRED the value (DCPRED + W[0] ∗ COEFFS[PBI[0]][0]). d) If P[1] is non-zero, assign DCPRED the value (DCPRED + W[1] ∗ COEFFS[PBI[1]][0]). e) If P[2] is non-zero, assign DCPRED the value (DCPRED + W[2] ∗ COEFFS[PBI[2]][0]). f) If P[3] is non-zero, assign DCPRED the value (DCPRED + W[3] ∗ COEFFS[PBI[3]][0]). g) Assign DCPRED the value (DCPRED//PDIV). h) If P[0], P[1], and P[2] are all non-zero: i. If |DCPRED−COEFFS[PBI[2]][0]| is greater than 128, assign DCPRED the value COEFFS[PBI[2]][0]. ii. Otherwise, if |DCPRED − COEFFS[PBI[0]][0]| is greater than 128, assign DCPRED the value COEFFS[PBI[0]][0]. iii. Otherwise, if |DCPRED − COEFFS[PBI[1]][0]| is greater than 128, assign DCPRED the value COEFFS[PBI[1]][0].

(c) If P is less than zero, assign RECP[FY + by][FX + 1] the value zero. d) Otherwise, if P is greater than 255, assign RECP[FY + by][FX + 1] the value 255. e) Otherwise, assign RECP[FY + by][FX + 1] the value P. f) Assign P the value (RECP[FY + by][FX + 2] − lflim(R, L)). g) If P is less than zero, assign RECP[FY + by][FX + 2] the value zero. h) Otherwise, if P is greater than 255, assign RECP[FY + by][FX + 2] the value 255. i) Otherwise, assign RECP[FY + by][FX + 2] the value P.

(c) If P is less than zero, assign RECP[FY + 1][FX + bx ] the value zero. d) Otherwise, if P is greater than 255, assign RECP[FY + 1][FX + bx ] the value 255. e) Otherwise, assign RECP[FY + 1][FX + bx ] the value P. f) Assign P the value (RECP[FY + 2][FX + bx ] − lflim(R, L)).

(c) If MVSIGN is 1, assign MVX the value −MVX. d) Read a 5-bit unsigned integer as MVY. e) Read a 1-bit unsigned integer as MVSIGN. f) If MVSIGN is 1, assign MVY the value −MVY.

(c) If MOREQIS is zero, set NQIS to 2. d) Otherwise: i. Read in a 6-bit unsigned integer as QIS[2]. ii. Set NQIS to 3. 7. If FTYPE is 0, read a 3-bit unsigned integer. These bits are reserved. If this value is not zero, stop. This frame is not decodable according to this specification. VP3 Compatibility The precise format of the frame header is substantially different in Theora than in VP3. The original VP3 format includes a larger number of unused, reserved bits that are required to be zero. The original VP3 frame header also can contain only a single qi value, because VP3 does not support block-level qi values and uses the same qi value for all the coefficients in a frame.

(c) For each value of tj from ti to (ti +RLEN−1), assign COEFFS[bi ][tj ] the value zero. d) Assign TIS[bi ] the value TIS[bi ] + RLEN. 3. Otherwise, if TOKEN is 9: a) Assign COEFFS[bi ][ti ] the value 1. b) Assign TIS[bi ] the value TIS[bi ] + 1.

(c) For each value of tj from ti to (ti +RLEN−1), assign COEFFS[bi ][tj ] the value zero. d) Assign TIS[bi ] the value TIS[bi ] + RLEN. 2. Otherwise, if TOKEN is 8: a) Read in a 6-bit unsigned integer as RLEN. b) Assign RLEN the value (RLEN + 1).

(c) For each value of ri from 0 to 7: i. Assign RES[ri ][ci ] the value (X[ri ] + 8) >> 4.

(c) For each value of ci from 0 to 7: i. Assign RES[ri ][ci ] the value X[ci ]. 2. For each value of ci from 0 to 7: a) For each value of ri from 0 to 7: i. Assign Y[ri ] the value RES[ri ][ci ]. b) Compute X, the 1D inverse DCT of Y using the procedure described in Section 7.9.3.

(c) For each consecutive value of sbi from 0 to (NSBS − 1), remove the bit at the head of the string BITS and assign it to SBPCODED[sbi ]. d) Assign NBITS the total number of super blocks such that SBPCODED[sbi ] equals zero. e) Read an NBITS-bit bit string into BITS, using the procedure described in Section 7.2.1. This represents the list of fully coded super blocks. f) For each consecutive value of sbi from 0 to (NSBS − 1) such that SBPCODED[sbi ] equals zero, remove the bit at the head of the string BITS and assign it to SBFCODED[sbi ]. g) Assign NBITS the number of blocks contained in super blocks where SBPCODED[sbi ] equals one. Note that this might not be equal to 16 times the number of partially coded super blocks, since super blocks which overlap the edge of the frame will have fewer than 16 blocks in them. h) Read an NBITS-bit bit string into BITS, using the procedure described in Section 7.2.2. i) For each block in coded order—indexed by bi : i. Assign sbi the i

(c) For each consecutive value of bi from 0 to (NBS − 1) such that BCODED[bi ] is non-zero and QIIS[bi ] equals qii : i. Remove the bit at the head of the string BITS and add its value to QIIS[bi ]. VP3 Compatibility For VP3 compatible streams, only one qi value can be specified in the frame header, so the main loop of the above procedure, which would iterate from 0 to −1, is never executed. Thus, no bits are read, and each block uses the one qi value defined for the frame.

(c) Assign RLEN the value (RLEN + 6). d) For each value of tj from ti to (ti +RLEN−1), assign COEFFS[bi ][tj ] the value zero. e) If SIGN is zero, assign COEFFS[bi ][ti + RLEN] the value 1. f) Otherwise, assign COEFFS[bi ][ti + RLEN] the value −1. g) Assign TIS[bi ] the value TIS[bi ] + RLEN + 1. h) Assign NCOEFFS[bi ] the value TIS[bi ]. 23. Otherwise, if TOKEN is 29: a) Read a 1-bit unsigned integer as SIGN. b) Read a 3-bit unsigned integer as RLEN.

(c) Assign RLEN the value (RLEN + 10). d) For each value of tj from ti to (ti +RLEN−1), assign COEFFS[bi ][tj ] the value zero. e) If SIGN is zero, assign COEFFS[bi ][ti + RLEN] the value 1. f) Otherwise, assign COEFFS[bi ][ti + RLEN] the value −1. g) Assign TIS[bi ] the value TIS[bi ]+RLEN+1. Assign NCOEFFS[bi ] the value TIS[bi ]. 24. Otherwise, if TOKEN is 30: a) Assign COEFFS[bi ][ti ] the value zero. b) Read a 1-bit unsigned integer as SIGN.

(c) Assign QIS[0] the value 63. d) For each value of bi from 0 to (NBS − 1), assign BCODED[bi ] the value zero. 3. Assign RPYW and RPYH the values (16 ∗ FMBW) and (16 ∗ FMBH), respectively. 4. Assign RPCW and RPCH the values from the row of Table 7.89 corresponding to PF. 5. Using ACSCALE, DCSCALE, BMS, NQRS, QRSIZES, QRBMIS, NBS, BCODED, MBMODES, MVECTS, COEFFS, NCOEFFS, QIS, QIIS, RPYW, RPYH, RPCW, RPCH, GOLDREFY, GOLDREFCB, GOLDREFCR, PREVREFY, PREVREFCB, and PREVREFCR, reconstruct the complete frame into RECY, RECCB, and RECCR using the procedure given in Section 7.9.4. 6. Using LFLIMS, RPYW, RPYH, RPCW, RPCH, NBS, BCODED, and QIS, apply the loop filter to the reconstructed frame in RECY, RECCB, and RECCR using the procedure given in Section 7.10.3. 7. If FTYPE is zero (intra frame), assign GOLDREFY, GOLDREFCB, and GOLDREFCR the values RECY, RECCB, and RECCR, respectively. 8. Assign PREVREFY, PREVREFCB, and PREVREFCR the values RECY, RECCB, and RECCR, respectively.

(c) Assign NCOEFFS[bi ] the value TIS[bi ]. 7. Otherwise, if TOKEN is 13: a) Read a 1-bit unsigned integer as SIGN.

(c) Assign NCOEFFS[bi ] the value TIS[bi ]. 6. Otherwise, if TOKEN is 12: a) Assign COEFFS[bi ][ti ] the value −2. b) Assign TIS[bi ] the value TIS[bi ] + 1.

(c) Assign NCOEFFS[bi ] the value TIS[bi ]. 5. Otherwise, if TOKEN is 11: a) Assign COEFFS[bi ][ti ] the value 2. b) Assign TIS[bi ] the value TIS[bi ] + 1.

(c) Assign NCOEFFS[bi ] the value TIS[bi ]. 4. Otherwise, if TOKEN is 10: a) Assign COEFFS[bi ][ti ] the value −1. b) Assign TIS[bi ] the value TIS[bi ] + 1.

(c) Assign MAG the value (MAG + 9). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ].

(c) Assign MAG the value (MAG + 7). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ]. 12. Otherwise, if TOKEN is 18: a) Read a 1-bit unsigned integer as SIGN.

(c) Assign MAG the value (MAG + 69). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ]. 17. Otherwise, if TOKEN is 23: a) Assign COEFFS[bi ][ti ] the value zero. b) Read a 1-bit unsigned integer as SIGN.

(c) Assign MAG the value (MAG + 37). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ]. 16. Otherwise, if TOKEN is 22: a) Read a 1-bit unsigned integer as SIGN. b) Read a 9-bit unsigned integer as MAG.

(c) Assign MAG the value (MAG + 21). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ]. 15. Otherwise, if TOKEN is 21: a) Read a 1-bit unsigned integer as SIGN. b) Read a 5-bit unsigned integer as MAG.

(c) Assign MAG the value (MAG + 2). d) Read a 1-bit unsigned integer as RLEN. e) Assign RLEN the value (RLEN + 2). f) For each value of tj from ti to (ti +RLEN−1), assign COEFFS[bi ][tj ] the value zero. g) If SIGN is zero, assign COEFFS[bi ][ti + RLEN] the value MAG. h) Otherwise, assign COEFFS[bi ][ti + RLEN] the value −MAG. i) Assign TIS[bi ] the value TIS[bi ]+RLEN+1. Assign NCOEFFS[bi ] the value TIS[bi ].

(c) Assign MAG the value (MAG + 13). d) If SIGN is zero, assign COEFFS[bi ][ti ] the value MAG. e) Otherwise, assign COEFFS[bi ][ti ] the value −MAG. f) Assign TIS[bi ] the value TIS[bi ] + 1. g) Assign NCOEFFS[bi ] the value TIS[bi ]. 14. Otherwise, if TOKEN is 20: a) Read a 1-bit unsigned integer as SIGN. b) Read a 4-bit unsigned integer as MAG.

(c) Assign LASTDC[2] the value zero. d) For each block of color plane pli in raster order, with coded-order index bi : i. If BCODED[bi ] is non-zero: A. Compute the value DCPRED using the procedure outlined in Section 7.8.1. B. Assign DC the value (COEFFS[bi ][0] + DCPRED). C. Truncate DC to a 16-bit representation by dropping any higher-order bits. D. Assign COEFFS[bi ][0] the value DC. E. Assign mbi the index of the macro block containing block bi . F. Assign rfi the value of the Reference Frame Index column of Table 7.46 corresponding to MBMODES[mbi ]. G. Assign LASTDC[rfi ] the value DC.

(c) Assign BY the vertical pixel index of the lower-left corner of block bi . d) If BCODED[bi ] is non-zero:

(C) COPYRIGHT 1994-2009 \* by the Xiph.Org Foundation and contributors http://www.xiph.org/ \*.

**License:** BSD  
  
BSD Zero Clause License  
Copyright (C) 2006 by Rob Landley <rob@landley.net>  
  
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.  
  
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.