SMPTE REGISTERED DISCLOSURE DOCUMENT

High Density Image Encoding for ARRIRAW Files

The attached document is a Registered Disclosure Document prepared by the proponent identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Errors in this document are to be reported to the proponent identified below, with a copy to eng@smpte.org.

All other inquiries in respect of this document, including inquiries as to intellectual property requirements that can be attached to use of the disclosed technology, are to be addressed to the proponent identified below.

Proponent contact information:

Brian Gaffney
Codex Inc.
3450 Cahuenga Blvd. West
Unit 103
Los Angeles, CA 90068
United States

Email: brian.gaffney@codex.online

Approved April 9, 2020

Table of Contents Page

Introduction

Codex High Density Image Encoding (HDE) can be used to compress raw Bayer data (e.g. from a digital image sensor) in such a way that decoded data is identical to the raw data. It has been widely tested and implemented with ARRI cameras and HDE-encoded ARRIRAW file sizes are typically 60% of those of the original files. HDE is designed to provide high encode and decode throughput for use with systems with high data rates (e.g. high resolution or high frame rate cameras), real-time playback, faster than real-time data transfers and low-latency decoding. A wide variety of third-party applications can be used to decode files containing HDE bitstreams for data management, transcoding, color grading and visual effects purposes.

HDE bitstreams can be decoded using the Codex High Density Encoding SDK that is available from the Codex Partner Program. The SDK allows transcoding between files containing HDE bitstreams and equivalent files containing raw data. The SDK, certification of applications implementing HDE decoding and information about encoding HDE bitstreams are available through the Codex Partner Program. Contact details are below.

Codex Partner Management Team Codex Digital
60 Poland Street London
W1F 7NT U.K.
Email: support@codex.online
Partner Agreement Application: https://codex.online/partner-program

1 Scope

This document describes the structure of data encoded with Codex High Density Image Encoding (HDE), along with algorithms for decoding the stored data.

It is the intent of this RDD to describe the structure and encoding of all fields in the HDE bitstream, such that users of the RDD can develop applications to decode HDE bitstreams and correctly identify their structure. It is specifically not the intent of this document to support development of applications encoding or creating HDE bitstreams, or the implementation of file formats based on HDE bitstreams.

2 References

ISO/IEC 8859-1:1997, 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No.1

3 Acronyms

Acronyms used in this document are listed below.

ASCII: American Standard Code for Information Interchange. Note: Refers to ISO/IEC 8859-1:1997 above.
HDE: High Density (Image) Encoding.

4 Trademarks

Codex Digital and HDE are registered trademarks of Codex Digital Ltd.
ARRI is a registered trademark of Arnold & Richter Cine Tecknik GmbH & Co. Betriebs (KG).

5 Definition of Field Encoding Techniques

Within the following sections, each data field is encoded according to one of the following techniques.

uint: Unsigned binary representation of a value written in the order from most significant to least significant bit.

zigzag-encoded: For an n-bit encoded value *zz*, signed integer *d* is found according to Equation 1.

- >>> represents a zero-extended right-shift
- ^ is an exclusive or operation
- & is a logical and operation

$$d = (zz >>> 1) \wedge -(zz \,\&\, 1)$$

Equation 1. Decoding a value from a zigzag-encoded value.

6 Definition of HDE bitstream

The HDE bitstream consists of a global header, followed by a non-zero number of plane parameters sub-elements, then a variable number of rows, each row containing a variable number of groups. Each group can be decoded to 16 samples.

Structure 1. HDE bitstream.

Name	Count
Bitstream header	1
Plane parameters	Number of planes
Rows	Sum of number of rows in each plane

Bitstream header: This sub-element appears once. The syntax of this sub-element is specified in section 6.1.

Plane parameters: This sub-element appears a number of times equal to the **Number of planes**, with each set of parameters in decoding order relating to the corresponding plane in decoding order. The syntax of these sub-elements is specified in section 6.2.

Rows: The rows of sample groups that comprise the image. Row syntax is specified in section 6.3. The rows are ordered according to the image plane to which they belong, and the **Plane interleave type**, specified in section 6.1.

6.1 Bitstream Header

Structure 2. Bitstream header.

Name	Encoding	Size (bits)
Four CC	uint	32
Version	uint	8
Encoder ID	uint	32
Group type	uint	8
Number of planes	uint	8
Plane interleave type	uint	8
Plane configuration	uint	8

Four CC: Allows the identification of the start of the bitstream. This field shall have the value `0x48 44 45 30` ('HDE0' in ASCII).

Version: Identifies the version of HDE compression in use. The value of this field is interpreted according to the following Table 1.

Table 1. Values for Version.

Version	Notes
0	Reserved
1	Bitstream compliant with this document

Encoder ID: Identifies the device and/or software that created this bitstream. Compliant encoders are assigned unique identifiers by Codex Digital Ltd. Codex HDE SDK uses the value `0x43 58 53 30` ('CXS0' in ASCII).

Group type: Specifies the shape of the sample groups and the order in which the samples within are encoded. The value of this field is interpreted according to the following Table 2.

Table 2. Values for Group type.

Group type	Group shape (width x height)	Scan order
0	16 x 1	Raster (left-to-right)

Number of planes: Specifies the number of sample planes comprising the full image. This field shall not have value equal to 0.

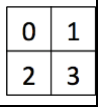
Plane interleave type: Specifies the order in which group rows are encoded. The value of this field is interpreted according to the following Table 3.

Table 3. Values for Plane interleave type.

Plane interleave type	Interleaving	Row ordering
0	No interleaving	All rows comprising the first plane are encoded in raster order, followed by all rows comprising the second plane (if present) and so on until all planes are encoded. i.e. P_1R_1, P_1R_2 … P_1R_N, P_2R_1 … P_2R_N, ... (where N is the **Plane height** of that plane)
1	Planar interleaving	The first row of the first plane is encoded in raster order, followed by the first row of the second plane (if present), and so on until the first row of each plane is encoded. This is then followed by the second row of each plane and so on until all rows are encoded. If a plane comprises fewer rows than the maximum number of rows among all planes, then rows from that plane are omitted from iterations beyond the last row of that plane. i.e. P_1R_1, P_2R_1 … P_NR_1, P_1R_2 … P_NR_2, ... (where N is the **Number of planes**)

Plane configuration: Specifies the arrangement of planes as they appear in the image. The value of this field is interpreted according to the following Table 4.

Table 4. Values for Plane configuration.

Plane configuration	Configuration
0	Every (x,y) raster coordinate has N components, corresponding with N equal-sized planes. Example: RGB image.
1	Every (x,y) raster coordinate has 1 component, number of planes must be 4, and the mapping of planes to pixels is as shown. Example: Bayer image. 

6.2 Plane Parameters

Each set of plane parameters shall contain fields according to the following specification.

Structure 3. Plane parameters.

Name	Encoding	Size (bits)
Plane ID	uint	8
Plane width	uint	32
Plane height	uint	32
Sample size	uint	8

Plane ID: Identifies the image component represented by this plane. The value of this field is interpreted according to the following Table 5.

Table 5. Values for Plane ID.

Plane ID as Hexadecimal	Plane ID as ASCII	Plane ID
0x3F	?	Undefined
0x52	R	Red channel (Bayer or RGB)
0x47	G	Green channel (Bayer or RGB), G_R (green in row with red) in Bayer images with two green channels
0x42	B	Blue channel (Bayer or RGB)
0x67	g	G_B (green in row with blue) in Bayer images with two green channels
0x41	A	Alpha channel
0x59	Y	Luminance channel
0x55	U	Blue chrominance channel (C_b)
0x56	V	Red chrominance channel (C_r)

Plane width: Specifies the width of the plane to which this set of parameters relates. This field shall not have value equal to 0.

Plane height: Specifies the height of the plane to which this set of parameters relates. This field shall not have value equal to 0.

Sample size: Specifies the size in bits of each sample of the reconstructed output image plane to which this set of parameters relates. This field shall have value between 1 and 16 inclusive.

6.3 Row

Each encoded row shall contain fields and sub-elements according to the following specification.

CEILING: The ceiling function of *x* is the unique integer *[x]* such that: *[x] -1 < x < [x]*

Structure 4. Row.

Name	Count
Row header	1
Groups	*CEILING (Plane width / 16)*

Row header: This sub-element appears once. The syntax of this sub-element is specified in section 6.3.1.

Groups: The sample groups that comprise this row. The syntax of these sub-elements is specified in section 6.3.2.

Implementation note: The total number of encoded samples can be greater than the number of samples in the decoded image. The "filler" values that are assigned to samples beyond the rightmost edge of a row are not defined by this specification and any values that can be encoded in the last group are valid.

6.3.1 Row Header

Each encoded row contains fields and sub-elements according to the following specification.

Structure 5. Row header.

Name	Encoding	Size (bits)
Start code	uint	16
Row index	uint	16
Row size	uint	24
Row header parity	uint	8

Start code: This field shall have value equal to **`0x52 4F`** ('RO' in ASCII).

Row index: This field is the lower 16 bits of an incrementing index identifying this row among all rows within the current image (i.e. if this index would exceed 65535, it instead restarts from 0).

Row size: The total encoded size of the Groups sub-element comprising the essence of this row in bytes.

Row header parity: A byte that contains parity information pertaining to the other fields in the row header. The whole 8 bytes of the Row Header shall have an even number of bits set when combined via bitwise AND with each of masks A to G from the following Table 6.

Table 6. Masks used to decode Row header parity.

Mask	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
A	0xFF	0xFF	0xFF	0xFE	0x00	0x00	0x00	0x40
B	0xFF	0xFF	0x00	0x01	0xFF	0xFC	0x00	0x20
C	0xFF	0x00	0xFF	0x01	0xFE	0x03	0xF8	0x10
D	0xF0	0xF0	0xF0	0xF1	0xE1	0xE3	0xC7	0x08
E	0xCC	0xCC	0xCC	0xCD	0x99	0x9B	0x36	0x84
F	0xAA	0xAA	0xAA	0xAB	0x55	0x56	0xAD	0x82
G	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Implementation note: The Start code and Row index are intended to facilitate error recovery; The Row header parity is intended to facilitate detection of bitstream corruption.

6.3.2 Group

Each encoded group shall contain fields and sub-elements according to the following specification.

Structure 6. Group.

Name	Encoding	Size (bits)
Sample size indicator	uint	8
Sample type mask	uint	16 (or not present)
Samples [ ]	*	*

Sample size indicator: Specifies the encoding of this group according to the Sample size indicator table (Table 8), in terms of parameters chosen by the encoder referred to as **Large sample size** and **Small sample size**.

Sample type mask: This field is only present if the encoding is **Reordered Zigzag-encoded differential**. There are eight 1s and eight 0s in the **Sample type mask**.

Samples: The encoded values that are used to construct the output image. This field consists of 16 elements. The samples can be encoded in three different ways described in the following Table 7. Elements are always tightly packed and written from most- to least-significant bit.

Table 7. Encoding strategies for Samples.

Encoding	Occurs if
Raw sample	Sample size indicator = 0
Zigzag-encoded differential	Sample size indicator != 0 and Large sample size = Small sample size
Reordered Zigzag-encoded differential	Sample size indicator != 0 and Large sample size != Small sample size

Raw sample encoding: Each element has a bit width equal to the **Sample size** of the plane to which this group belongs. Decoded samples are equal to encoded elements and do not require further processing.

Zigzag-encoded differential encoding: Each element has a bit width equal to **Large sample size**. For group *G*, differentials $D_{G,X}$ are derived by decoding the Zigzag-encoded elements as in section 5. Samples $P_{G,X}$ are derived according to Equation 2. The previous sample can be in the previous **Group**. For the first sample in a **Row**, the previous sample is always equal to 0.

$$P_{0,0} = D_{0,0} \ (+ 0)$$
$$P_{0,1} = D_{0,1} + P_{0,0}$$
$$P_{1,0} = D_{1,0} + P_{0,15}$$

Equation 2. Decoding samples from differentials.

Reordered Zigzag-encoded differential encoding: The first eight elements in decoding order have bit width equal to **Large sample size**. The remaining eight elements have bit width equal to **Small sample size**. These must be reordered according to the **Sample type mask** (see Figure 1 below) before being decoded in the same way as in **Zigzag-encoded differential encoding**.

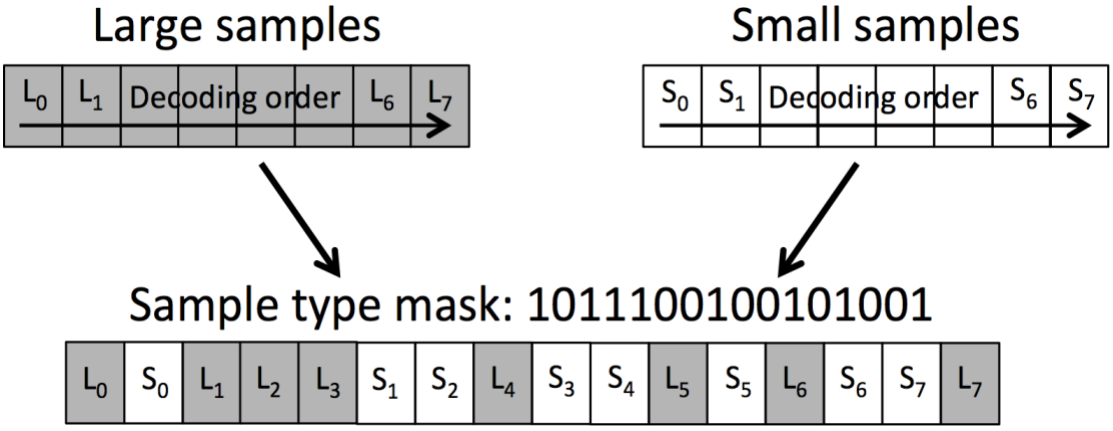


Figure 1. Reordering using Sample type mask.

7 Sample Size Indicator Table

Table 8. Values for Sample size indicator.

Sample size indicator	Large sample size (bits)	Small sample size (bits)
0	Special - see section 6.3.2.	
1	0	0
2 to 3	1	0 to 1
4 to 6	2	0 to 2
7 to 10	3	0 to 3
11 to 15	4	0 to 4
29 to 36	7	0 to 7
37 to 45	8	0 to 8
46 to 55	9	0 to 9
56 to 66	10	0 to 10
67 to 78	11	0 to 11
79 to 91	12	0 to 12
92 to 105	13	0 to 13
106 to 120	14	0 to 14
121 to 136	15	0 to 15
137 to 153	16	0 to 16
154 to 171	17	0 to 17
172 to 255	Reserved	