TechNote

ADV202 Output Formats and Attribute Data Format

ADV202 Output format and Attribute Data format

The following document will contain information about the ADV202's compressed output data formats that are programmable with the Encode JPEG2000 parameter COD_STYLE and the Attribute data formats that are programmable with the Encode JPEG2000 parameter ATTRTYPE.

It is recommended to refer to the *Getting Started with the ADV202* application note for reference as how to configure the ADV202 using these parameters.

1. ADV202 Output format – COD_STYLE

The ADV202 allows compressed data to be output in the following formats:

1. ADV202 Raw Format
2. J2C Format
3. JP2 Format

For any of these formats, the ADV202 will insert a ADV202 specific header at the beginning of the code stream. This is followed by a JPEG2000 compliant header. The contents of the JPEG2000 specific header will differ with the format chosen, the ADV202 header format is the same for all output formats.

The ADV202 header is required at the start of the codestream in order for the ADV202 to start decoding a compressed datastream.

The ADV202 header can be removed from the .j2c or jp2 compressed files if a JPEG2000 compliant decoder is required for decoding, as for example the Kakadu JPEG2000 software decoder.
Kakadu is fully compatible with the ADV202 output dataformats .j2c and jp2 after removing the ADV202 header. For full color support the .jp2 format must be used.

ADV202 Output Formats and Attribute Data Format

ADV202 output stream format

The following figures show the output data streams for the 3 possible output formats the ADV202 can provide.

ADV202 header	JPEG2000 compliant header	Codeblock 1 Attribute Data	Codeblock 2 Attribute Data	…… ….	Codeblock n Attribute Data	Compressed Data

Figure 1 ADV202 Raw Format output – COD_STYLE = 0

ADV202 header	JPEG2000 compliant header	JPEG2000 packed packet headers	Codeblock Compressed Data	…… ….	JPEG2000 packed packet headers	Codeblock Compressed Data

Figure 2 J2C Format - COD_STYLE = 1

ADV202 header	JPEG2000 compliant header	JPEG2000 packed packet headers	Codeblock Compressed Data	…… ….	JPEG2000 packed packet headers	Codeblock Compressed Data

Figure 3 Jp2 Format - COD_STYLE = 2

ADV202 header

As mentioned before, the ADV202 will insert a ADV202 specific header at the beginning of the code stream. The following table contains detailed information about this ADV202 header.

Big Endian Byte Offset	Number of Bytes	Description
0x0	4	**Field header** 0xFFFFFFF0 = even field delimiter 0xFFFFFFF1 = odd field delimiter
0x4	4	**Compressed image index** in captured stream. Starting with 0, and incrementing for each successive image that is captured during the encode session.

Big Endian Byte Offset	Number of Bytes	Description
0x8	1	**Output code stream format** – COD_STYLE parameter setting: bit 0 = ADV202 Raw Format bit 1 = J2C Format bit 2 = Jp2 Format bit 3 = 1 = ADV202 header not generated bit 3 = 0 = ADV202 header generated bit 4 to 7 only valid with J2c or Jp2 format, otherwise these bits must all be 0. bit 4 = 1 = Include EPH bit 4 = 0 = EPH not included bit 5 = 1 = Include SOP bit 5 = 0 = SOP not included bit 6 = 1 = Include PLT bit 6 = 0 = PLT not included bit 7 = 1 = include PPT bit 7 = 0 = Packet headers with Packet Body
0x9	1	**VFORMAT parameter:** 0 = NTSC 4:2:2 1 = PAL 4:2:2 2 = 1080i Luminance 3 = 1080i Chrominance 4 = Custom specific 5 = 720p/60 Luminance 6 = 720p/60 Chrominance 8 = NTSC de-interlaced 9 = PAL de-interlaced
0xA	1	reserved (0x00)
0xB	1	**Header version**: 0x02 for encode firmware with App ID = 0xFF82
0xC	4	**Size of compressed data**

Big Endian Byte Offset	Number of Bytes	Description
		Number of 32-bit words in compressed data and attribute data. This does not include this 16 byte ADV202 header.

Table 1 ADV202 header

JPEG2000 compliant header

Detailed information about JPEG2000 markers can be found in the ISO/IEC15444-1 standard. The JPEG2000 compliant header contains main and tile headers from the JPEG2000 standard. It consists of all the parameters needed to decode an image correctly, including the quantisation stepsizes.
The table below lists the most common markers that are inserted into the compressed codestream by the ADV202 .

Postition	JPEG2000 compliant marker	Description	ADV202 Raw Format	J2C format	Jp2 format
		First marker after ADV202 header	FF4F	FF4F	00 00 00 0C
	FF4F	SOC – Start of Code	yes	yes	yes
	FF52	COD Contains information about the coding style	yes	yes	yes
	FF5C	QCD Contains information about the Quantisation used.	yes	yes	yes
	FF90	SOT This marker follows immediately the SOC marker and contains information about the input image/tile size	yes	yes	yes
	FF93	SOD Following this marker is the actual compressed	no	yes	yes

Postition	JPEG2000 compliant marker	Description	ADV202 Raw Format	J2C format	Jp2 format
		data.			
	FFD9	EOC – End of codestream Can be found at the end of a field/frame and will always be the last marker in the codestream.	yes	yes	yes

Table 2 JPEG2000 markers in the ADV202 output stream

Codeblock Attribute Data for Raw Format output

Applies to COD_STYLE programmed to 0, ie ADV202 Raw format output.
COD_STYLE is programmed with the encode parameter as described in the *Getting Started with the ADV202* application note.

A 32-bit word description will follow the JPEG2000 compliant header as shown in figure 1. This word will describe every codeblock that is contained within the output codestream.

Bits	Name	Description
31		Reserved
30:28	LEV	Transform Level 0 = 1st transform level which has the largest subbands
27:26	COMP	Component 0 = Y 1 = Cb 2 = Cr
25:22	NZBP	Number of leading zero bitplanes
21:16	NCP	Number of coding passes in codeblock
15:0	BYTELEN	Number of bytes of compressed data for this codeblock

Table 3 Raw format – Codeblock Attribute data

PPT, PLT, SOP, EPH markers for j2c or jp2 output format

Bits [7:4] of the COD_STYLE parameter control the insertion of Packed Packet Headers into the compressed codestream if the output format is either .jp2 or .j2c.
These bits can be programmed with the COD_STYLE parameter as described in the *Getting Started with the ADV202* application note.

The Packed Packet Headers are described in detail in the JPEG2000 standard [ISO/IEC 15444-1].

PPT and PLT are inserted at the start of a tile. SOP and EPH are inserted at the start of each packet.

These markers are required for applications that require multi-layer decoding, or decoding of certain regions of the input image only since packet headers contain information about layer, decomposition level and component. They can also be used for error detection.

Packed Packet Header	Marker	Description	Function
PPT	0xFF61	Packed Packet Header - Tile part header	
PLT	0xFF58	Packet Length – Tile part header	Should be enabled for Multi-layer applications. Can be used for error detection.
SOP	0xFF91	Start of Packet	Can be used for error detection.
EPH	0xFF92	End of Packet Header	Can be used for error detection.

Table 4 PPT,PLT, SOP and EPH markers

2. ATTRIBUTE DATA FORMAT

The ADV202 allows output of additional information about the compressed data stream to be output over the ATTR FIFO. This is controlled with the ATTRTYPE encode parameter as described in the *Getting Started with the ADV202* application note.

The host must enable both DMA channels when using this feature, one assigned to the CODE FIFO, and the other to the ATTR FIFO. This is achieved in configuring registers EDMOD0 and EDMOD1. Refer to the *Getting Started with the ADV202* application note on instructions on how to configure the ADV202.

The ATTRTYPE encode parameter is used to select between:

1. Use ADV-JP2000 attribute data format
2. Log2 Convex Hull without Adjustments
3. Log2 Convex Hull with Quantization Adjust
4. ADV202 field header output to ATTR FIFO
5. ADV202 field header and packet byte length output to ATTR FIFO

If the ATTRTYPE parameter is set to 1, 2, or 3, then the ADV202 will be configured to output codeblock attribute data via the ATTR FIFO. The first of these words will contain some codeblock description information. The format for the rest of the attribute data is dependent on the ATTRTYPE parameter value. These words can contain distortion metrics, distortion slopes and byte length [one per coding pass].

If ATTRTYPE is set to 4, then the 16-byte ADV202 compressed image header will be output to the ATTR FIFO instead of being placed at the start of each compressed image. This will allow systems to obtain this header information without the need to parse through the code stream data.

If ATTRTYPE is set to 5, then a more detailed byte-length summary will be output to the ATTR FIFO to support external merging of HD Luminance and HD Chrominance streams from two different ADV202s. This setting must be used for multi-chip applications if stream merging of Y and CbCr data is required.

Attribute Datastream Format for ATTR_TYPE =1, 2 or 3

The basic structure of a attribute data stream is illustrated as such:

Attribute Address	**ATTR data output sequence**
ATTR_ADDR	Codeblock Attribute Header Word
ATTR_ADDR + 4	Codeblock Attribute Data Word 1
ATTR_ADDR + 8	Codeblock Attribute Data Word 2
ATTR_ADDR + 12	Codeblock Attribute Data Word 3

Table 5 Attribute Data Stream Structure [in 32-bit words]

The sequence of the codeblocks in the attribute data stream will always be such that a subband's codeblocks will be in raster order. For example, the 3rd codeblock with the same SBID, LEV, and COMP [see table below] parameters will represent the 3rd codeblock in that particular subband as the codeblocks appear in raster order. Thus the X and Y coordinates of the codeblock within the subband are implicit by the sequence of the codeblocks, and therefore need not be specified.

The sequence of codeblocks can change from field-to-field as the sequence is based on when a codeblock's entropy coding has finished. But the subband raster-order rule will always be true.

Codeblock Attribute Header Word

The header word contains all the parameters that the packetizer routine will need to have. It includes the number of data words in the attribute stream (does not include this header word), the format of the data words, codeblock identification parameters, and other necessary packet header parameters. The bit assignments for this word are as follows:

Bits	Name	Description
31	CHF	Codeblock header flag Set to 1 to indicate this is a header word.
30:25	NDWRD	Number of Attribute Data words in this codeblock attribute data stream
24:21		Reserved
20:19	ADFMT	Attribute Data Format [see below]
18		Reserved
17:16	SBID	Subband ID 0 = LL 1 = LH 2 = HL 3 = HH
15		Reserved
14:12	LEV	Transform Level 0 = Transform level 1 [lowest] 1 = Transform level 2 2 = Transform level 3 3 = Transform level 4 4 = Transform level 5 5 = Transform level 6
11:10	COMP	Component 0 = Y

		1 = Cb 2 = Cr
9:6	NZBP	Number of leading zero bitplanes
5:0	NCP	Number of coding passes in codeblock

Table 5 Codeblock Attribute Header format

Codeblock Attribute Data Words

The data stream will contain one 32-bit word for each coding pass in the codeblock, thus making NDWRD equal to NCP.

Codeblock Attribute Data Word if ATTR_TYPE = 1

In this format, the data word will contain a distortion metric for the coding pass, a termination point flag (T), and the byte length of the coding pass. The distortion metric is a 15-bit floating point value with 1 sign bit (S), 7 exponent bits, and 7 mantissa bits. The termination flag bit is included to indicate that the coding pass is terminated.

	0	**S**	**Distortion Metric exp**	**Distortion Metric mant**	**T**	**Byte Length**
Bit number	31	30	29:23	22:16	15	14:0

Codeblock Attribute Data Word if ATTR_TYPE = 2 or ATTR_TYPE = 3

When using this mode, the Entropy Coder performs the convex hull algorithm to identify which coding passes are not valid truncation points. To perform this task, logarithmic distortion slopes are computed for each truncation point, and these values are provided in the attribute data word as a twos-complement number with 7.7 precision (for ATTRTYPE==3, the distortion-slope will be compensated for the quantization stepsize that was assigned to this codeblock's subband).

If a coding pass has been invalidated as a truncation point, then the valid bit (V) for that coding pass will be set to 0.

The T flag indicates if the coding pass is terminated.

	0	S	Distortion Slope integer	Distortion Slope fraction	T	Byte Length
Bit number	31	30	29:23	22:16	15	14:0

ATTR_TYPE = 4

If ATTRTYPE is set to 4, then the 16 byte ADV202 compressed image header will be output to the ATTR FIFO instead of being placed at the start of each compressed image. This will allow systems to obtain this header information without the need to parse through the codestream data.

ATTR_TYPE = 5

This ATTRTYPE setting is used for merging Y and CbCr streams in multi-chip mode.
If ATTRTYPE is set to 5, then a more detailed byte-length summary will be output to the ATTR FIFO to support external merging of HD Luminance and HD Chrominance streams from two different ADV202s.

This can also be used to obtain the packet lengths without parsing the codestream for the PLT markers.

The ADV202 16-byte header is not placed in the output codestream for this mode (just like for the ATTRTYPE==4 mode described above).

The following is an example that illustrates the format of the attribute data for this mode:

If the ADV202 is capturing HD Luminance, then the JPEG2000 Main Header will be for all 3 components in the final merged YCbCr codestream. Likewise, if the ADV202 is capturing HD Chrominance, the Main Header and Tile Header will be omitted. For both of these HD cases, there will be no EOC markers at the end of the codestream. For standard definition VFORMATS [refer to *Getting Started with the ADV202* application note], the headers will be correct and there will be an EOC marker present at the end of the codestream.

The Attribute data will first provide the 16 byte ADV202 Header and the number of compressed words will include the zero padding at the end of the codestream. Then, the number of byte length counts is given, followed by the byte count of the first valid truncation point (which will be the sum of the byte lengths for the Main Header, Tile Header, and the first packet). Finally, the byte lengths of each successive truncation point (i.e. packet byte lengths) will be provided.

Description	ATTR data bytes		ATTR data output
ADV202 Header	byte 0-3	Field Delimiter	0xFFFFFFF0
	byte 4-7	Field Index	0x00000000
	byte 8-11	Cod_style and Vformat	0x01020002
	byte 12-15	Number of words in compressed image	0x00000150
ATTR data	byte 16-19	Number of byte length counts for this image	0x00000006
	byte 20-23	Byte Length Sum of Main Header, Tile Header and Packet 1	0x000000C8 [200]
	byte 24-27	Byte Length of Packet 2	0x00000064 [100]
	byte 28-31	Byte Length of Packet 3	0x00000096 [150]
	byte 32-35	Byte Length of Packet 4	0x000000C8 [200]
	byte 36-39	Byte Length of Packet 5	0x000000FA [250]
	word 40-43	Byte Length of Packet 6	0x000000190 [400]

Table 7 ATTR data output for ATTR_TYPE=5 – example

CODE data byte length	CODE data output
150 bytes	Main header + Tile Header
50 bytes	packet 1
100 bytes	packet 2
150 bytes	packet 3
200 bytes	packet 4
250 bytes	packet 5
400 bytes	packet 6
28 bytes	zero padding to DMA Burst boundary

Table 8 CODE data output for ATTR_TYPE=5 – example

Filename: output_format_1.doc
Directory: C:\drive\adv_jpg\appnotes\adv202output format
Template: C:\Documents and Settings\cbako\Application Data\Microsoft\Templates\Normal.dot
Title: ADV202 – How to use the ADV202 overview
Subject:
Author: cbako
Keywords:
Comments:
Creation Date: 8/16/2005 11:17 AM
Change Number: 63
Last Saved On: 10/10/2005 1:53 PM
Last Saved By: cbako
Total Editing Time: 1,134 Minutes
Last Printed On: 10/10/2005 1:54 PM
As of Last Complete Printing
Number of Pages: 11
Number of Words: 2,609 (approx.)
Number of Characters: 12,680 (approx.)