Intel® High Definition Audio Specification
Document Change Notification

Date: June 2, 2008
Company: Intel Corporation
Address: 1900 Prairie City Rd.
City: Folsom State: CA
Country: USA Zip: 95630

Change Identification: **DCN No: HDA039-A**
Change Revision: 1.0
Document Revision: Intel® High Definition Audio 1.0

This document discloses changes to the Intel® High Definition Audio Specification and all information contained herein is provided under the terms of the "AZALIA" SPECIFICATION DEVELOPMENT AGREEMENT" also known as Intel® High Definition Audio Specification Developer Agreement, and all the terms of such agreement, including the confidentiality provisions, shall apply to this disclosure.

Title: HDMI ELD Memory Structure

Brief description of the functional changes:

This DCN provides the description of the HDMI EDID-Like Data (ELD) memory structure. This ELD memory structure is read by issuing the HDMI ELD Data command verb added in DCN No: HDA034-A.

Definition Text Formatting:

xxx Original text in existing specification or DCN released earlier.
yyy New text inserted by this new DCN.
~~zzz~~ Deleted text introduced by this new DCN.

New Definitions:

7.3.3.34 HDMI EDID-Like Data (ELD) Data

The audio software for the HDMI codec will need information about the audio capabilities of an attached HDMI sink device. This information is stored in the HDMI sink device's EDID. Typically, the EDID flows through a graphics adapter to graphics software, so the graphics adapter HW will not have knowledge of the EDID contents.

To that end, a new mechanism is defined here for passing the HDMI sink device's audio EDID information from the graphics software to the audio software. The data payload containing the audio information will be known as EDID-Like Data or ELD and will contain a subset of the HDMI sink device's EDID information. The size and contents of the ELD buffer will be determined by the HDMI Audio codec manufacturer.

The ELD information will be valid if the HDMI sink is attached and powered on and the ELD Valid bit is set. The Pin Widget that is associated with this HDMI widget will report if the device is attached and that the ELD memory is populated and valid by reporting Presence Detect of 1 and ELD Valid of 1 to a Pin Sense control command. As with the Presence Detect bit, the changes to the ELD Valid bit can also result in the generation of unsolicited responses.

Command Options:

Table 110. HDMI ELD Data

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F2Fh	Bits 7:0 Offset into ELD memory	Bit 31 ELD Valid indication Bits 30:8 *Reserved* Bits 7:0 Data: ELD data byte at specified offset into the ELD memory

Response Structure:

31	30:8	7:0
ELD Valid	*Reserved*	ELD Byte from memory

Figure 72. ELD Data Response Format

ELD Valid is a bit that indicates to software that the byte being returned is not valid.

ELD Byte [7:0] is the byte of configuration data specified by offset. For a non-existent ELD location, GET returns a value of 0. Note that the byte index will auto-increment after a Get command is completed.

Applies to:

- HDMI Pin Complex

7.3.3.34.1 ELD Memory Structure

The ELD memory structure is split into 3 blocks: header, baseline, and vendor defined. The header block contains the version and structure size information. The baseline block contains information about the HDMI sink device standard features which OS class driver can understand. The vendor defined block contains information about any HDMI sink device extended features that are specific to a particular vendor and may only be understood by a vendor specific driver.

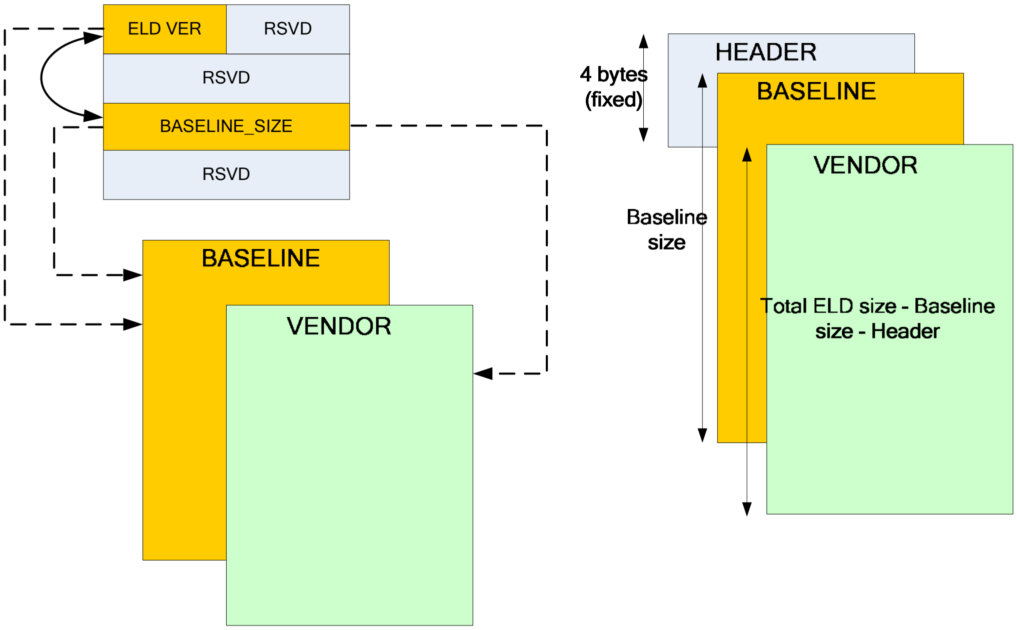

Figure 73. ELD Memory Structure

The size of the ELD memory structure is discovered by issuing HDMI DIP – Size command verb with ELD buffer size bit set. The header block is a fixed size of 4 bytes. The baseline block is variable size in multiple of 4 bytes, and its size is defined in the header block Baseline_ELD_Len field (in number of DWords). The vendor defined block is also variable size, and its size is the remaining bytes of the ELD memory structure, i.e. ELD buffer size – Header size of 4 bytes – (Baseline_ELD_Len * 4 bytes).

Header Block:

	Bit							
Byte offset into ELD memory	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	ELD_Ver					*Reserved*		
1	*Reserved*							
2	Baseline_ELD_Len							
3	*Reserved*							

Figure 74. Header Block of ELD Memory Structure

ELD_Ver[4:0] indicates the baseline ELD version number. Each version number has a fixed baseline ELD structure with a defined maximum number of bytes. It also indicates the CEA specification that the baseline ELD structure supports.

Table 112. ELD_Ver Encoding

Value	**Description**
00000b	*Reserved*

Value	**Description**
00001b	Indicates version 1, which is an obsolete ELD structure. Treated as reserved.
00010b	Indicates version 2, which supports CEA version 861-D or below. Maximum Baseline ELD size of 80 bytes (15 SAD count).
00011b – 11110b	*Reserved*
11111b	Indicates an ELD that has been partially populated through implementation specific mean of default programming before an external graphics driver is loaded. Only the field that is called out as “canned” field will be populated, and audio driver should ignore the non “canned” field.

Baseline_ELD_Len[7:0] indicates the length of the baseline structure in number of DW. There is a limit of the maximum length of the baseline structure supported per baseline ELD version number. It is required that the baseline structure length is equal or below the maximum number of bytes supported associated with the baseline ELD version.

Baseline Block:

	Bit							
Byte offset into ELD memory	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
4	CEA_EDID_Ver			MNL				
5	SAD_Count				Conn_Type		S_AI	HDCP
6	Aud_Synch_Delay							
7	*Rsvd*	RLRC	FLRC	RC	RLR	FC	LFE	FLR
8 to 15	Port_ID							
16 to 17	Manufacturer Name							
18 to 19	Product Code							
20 to 20 + MNL – 1	Monitor_Name_String							
20 + MNL to 20 + MNL + (3 * SAD_Count) – 1	CEA_SADs							
20 + MNL + (3 * SAD_Count) to 4 + Baseline_ELD_Len * 4 – 1	*Reserved*							

Figure 75. Baseline Block of ELD Memory Structure for ELD_Ver = 00010b

MNL[4:0] indicates the length of the monitor name string in number of bytes.

Table 113. MNL Encoding

Value	Description
00000b	Indicates the absence of a monitor name string.
00001b – 10000b	Indicates a monitor name string of length of 1 – 16 bytes.
10001b – 11111b	*Reserved*

CEA_EDID_Ver[2:0] indicates the CEA EDID Timing Extension version number of the HDMI sink device supports. There is a limit of the latest CEA EDID Timing Extension version number supported per baseline ELD version number. It is required that the version number is equal to or less than the supported version associated with the baseline ELD version. OS class driver and other function driver shall support the highest version number and below for any given CEA EDID Timing Extension version number. Changes to CEA EDID Timing Extension version number may affect the content of the info frame.

Table 114. CEA_EDID_Ver Encoding

Value	Description
000b	Indicates no CEA EDID Timing Extension block present.
001b	Indicates CEA-861.
010b	Indicates CEA-861-A.
011b	Indicates CEA-861-B, C, or D.
100b – 111b	*Reserved*

HDCP indicates the support for HDCP. If set to 1, it indicates the receiver supports HDCP over the HDMI link.

S_AI indicates the Supports_AI capability from the HDMI Vendor Specific Data Block. If set to 1, it indicates the sink supports at least one function that uses information carried by the ACP, ISRC1, or ISRC2 packets. A value of 0 indicates the sink does not support ACP, ISRC1, or ISRC2 packets.

Conn_Type[1:0] indicates the pin connection type of the device currently plugged in.

Table 115. Conn_Type Encoding

Value	Description
00b	Indicates a HDMI connection type.
01b – 11b	*Reserved*

SAD_Count[3:0] indicates the number of three byte CEA Short Audio Descriptors reported by the sink and present in the baseline structure. This field is a “canned” field that would be populated through implementation specific mean of default programming before the graphic driver is loaded, typically with value of 1 to indicate that there is one descriptor describing basic LPCM audio support.

Table 116. SAD_Count Encoding

Value	Description
0000b	Indicates no Short Audio Descriptors present.
0001b – 1111b	Indicates 1 – 15 Short Audio Descriptors present.

Aud_Synch_Delay[7:0] indicates the amount of latency added by the sink. It is expressed in terms of units of 2 ms of delay of video compared to audio for the given video mode being driven on the link.

Table 117. Aud_Synch_Delay Encoding

Value	Description
00h	Indicates that one or more down stream devices does not support the reporting of audio and video processing times, that the video and audio delay times for the given display mode are assumed equal, or that the reported processing times are both zero indicating that a down stream device has already compensated for the difference in processing times..
01h – FAh	Indicates that there is a difference in the amount of latency that video trails audio processing times that should be compensated for to align video and audio tracks. The resultant delay in milliseconds should be calculated by the following formula to retrieve the resultant delay that has to be compensated: delay in ms = value * 2. The maximum delay is 500 ms.
FBh – FFh	*Reserved*

FLR is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of front left and right transmission channels. This field is a "canned" field that would be populated through implementation specific mean of default programming before the graphic driver is loaded.

LFE is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of a low frequency effect transmission channel.

FC is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of a center front transmission channel.

RLR is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of rear left and right transmission channels.

RC is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of a center rear transmission channel.

FLRC is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of front left and right of center transmission channels.

RLRC is part of the Speaker Allocation Block byte. If set to 1, it indicates the presence of rear left and right of center transmission channels.

Port_ID[63:0] indicates the 8 bytes port identification value. This field is a "canned" field that would be populated through implementation specific mean of default programming before the graphic driver is loaded. The bytes orientation is little endian, i.e. the lowest significant byte is located at the lower byte offset of the ELD memory structure and most significant byte is located at the higher byte offset of the ELD memory structure.

Manufacturer Name indicates the 2 byte Manufacturer Name ID from the sink device base EDID.

Product Code indicates the 2 byte Product Code ID from sink device base EDID.

Monitor_Name_String indicates the ASCII string of monitor name extracted from 16 byte product description of the Source Product Description Info Frame. The bytes orientation is little endian, i.e. the lowest significant byte is located at the lower byte offset of the ELD memory structure and most significant byte is located at the higher byte offset of the ELD memory structure.

CEA_SADs indicates up to 15 entries of 3-byte CEA-861 Short Audio Descriptor reported by the sink device. This field is a "canned" field that would be populated through implementation specific mean of default programming before the graphic driver is loaded, typically with only one LPCM SAD entry to indicate the basic LPCM audio support. The bytes orientation is little endian, i.e. the lowest significant byte is located at the lower byte offset of the ELD memory structure and most significant byte is located at the higher byte offset of the ELD memory structure.

Vendor Defined Block:

The vendor defined block of the ELD memory structure byte offset starts from 4 + Baseline_ELD_Len * 4 to ELD buffer size – 1. This structure is vendor specific. OS class driver will not interpret this block. Only the associated vendor defined graphic/audio driver will be able to understand and enumerate these features based on the specific vendor ELD version number.