Intel® High Definition Audio Specification
Document Change Notification

Date: December 8, 2005
Company: Intel Corporation
Address: 1900 Prairie City Rd.
City: Folsom State: CA
Country: USA Zip: 95630

Change Identification: **DCN No: HDA006-A**
Document Revision: Intel® High Definition Audio 1.0

Title: Clarification to Sub-system Identification reporting

Previously the definition of the Subsystem ID was not sufficient for proper driver loading on some popular OSes. There was a need for more details and a further refinement of the information into well defined fields that are used for ALL codecs. This change provides this clarity.

Examples of device PnP strings that could be used with the new definition would be:

String	Usage
HDAUDIO\Func&Ven_yyyy&Dev_zzzz	Universal driver from 3rd party, but must run against ALL designs using the specified codec
HDAUDIO\Func&Ven_yyyy&Dev_zzzz&Subsys_aa	Universal driver from 3rd party with a specific and known audio design specified by the Assembly ID
HDAUDIO\Func_xx&Ven_yyyy&Dev_zzzz&Subsys_mmmmssaa	Driver loaded only on one specific board and silicon; normally used for only a broken board
HDAUDIO\Func	Class driver from OS vendor only
HDAUDIO\Func_xx&Ven_yyyy&Dev_zzzz&Subsys_mmmmssaa&Rev_bbbb	Driver loaded only on one specific board and silicon; normally used for only a broken board and silicon
HDAUDIO\Func&Ven_yyyy&Dev_zzzz&Rev_bbbb	Single codec driver tied to a specific version, typically not used
HDAUDIO\Func&Ven_yyyy	Universal driver for all of one vendors codecs, not allowed

HDAUDIO **Class code for drivers loaded as clients of a bus driver**
Func_XX **Function group type where 01 is Audio**
Ven_YYYY **Vendor PCI SIG ID that is returned by the Codec**
Dev_ZZZZ **Device ID of the function Group returned by the Codec**
Subsys_MMMMSSAA **Sub-System ID returned by the Codec (to be changed to Implementation ID)**
Rev_BBBB **Revision returned by Codec, but stepping ID is not used**

Current Definition:

Section 7.3.3.30 of the Intel® High Definition specification version 1.0 defines:

7.3.3.30 Subsystem ID

This set of controls provides read/write access to the 32-bit Subsystem ID register contained in each Functional Group. This register is used to identify the functional group to the software PnP subsystem. The Assembly ID (8 bits) is intended primarily for modems; when used, its value is loaded from a "strapping option" or other board-specific mechanism at power-up time. The Subsystem ID (24 bits) is intended to be "hardwired" into the silicon. The silicon vendor is responsible for defining the values used in both fields. It is recommended that this control default to a non-zero value. System BIOS or other means may also be used to write to this register to set it. In such cases, the register should be set to its proper value at all times the operating system or application software may read the register. The ability for software to write this register is not a requirement as long as operating system requirements for unique Subsystem ID's can be met through other means.

In the case where the SSID is determined through external means, for instance the codec reading an external EEPROM to load register defaults, this register may return a value of 0xFFFFFFFF for up to 7 ms after the de-assertion of the Link **RST#** signal before changing to reflect the proper value.

31:8	**7:0**
Subsystem ID	Assembly ID

Figure 1. Subsystem ID Register

New Definition:

7.3.3.30 Implementation Identification

This set of controls provides read/write access to a 32-bit Implementation Identification (IID) register contained in each Functional Group. This register is used to identify the functional group and all the surrounding logic to a software PnP subsystem. The Implementation Identification is comprised of three fields that are intended to be used to better identify the proper software and drivers to be used and may be used either together or separately. The Implementation Identification value must always be used in conjunction with the Vendor and Device identification as the Implementation Identification value is not guaranteed to be unique by itself.

System BIOS or other means may also be used to write to this register to set it. In such cases, the register should be set to its proper value at all times the operating system or application software may read the register. The ability for software to write this register is not a requirement as long as operating system requirements for unique identification could be met through other means.

In the case where the Board Implementation ID is determined through external means, for instance the codec reading an external EEPROM to load defaults, this register may return a value of 0xFFFFFFFF for up to 7 Milliseconds after the de-assertion of the Link **RST#** signal before changing to reflect the proper value.

<table>
<tr><th>31:16</th><th>15:9</th><th>7:0</th></tr>
<tr><td colspan="2">Board Implementation ID (BIID)</td><td rowspan="2">Assembly ID (AssyID)</td></tr>
<tr><td>Board Manufacturer Identification (BMID)</td><td>Board SKU (BSKU)</td></tr>
</table>

Figure 2. Implementation Identification Value

The intent for the Implementation Identification value is to provide identification of all the aspects of the design that may affect the software and drivers that should be used. This includes not only the implementation of the function group, but external hardware and board design. The value returned is intended to have a default that would in most cases by overwritten by system BIOS. In cases where it is not possible for the system BIOS to provide an appropriate value, such as when an add in card is used, the default value is augmented with the Assembly ID that must be changeable independent of the function group "hardwired" identification. This Assembly ID (8 bits) is intended primarily for add in cards; when used, its value is loaded from a "strapping option" or other board-specific mechanism at power-up time.

The Board Implementation Identification (24 bits) is intended to have a "hardwired" default in the silicon. The silicon vendor is responsible for defining the values used in both the Board Manufacturer Identification and Board SKU fields. It is recommended that this control default to a non-zero and appropriate value.

The Board Manufacturer Identification field is intended to be used to uniquely identify the manufacturer of the board where the function group (codec) is attached. Although not required by this specification, it is expected that this field would contain the 16 bit identification that is assigned to the board manufacturer by the PCI SIG. In cases where the value for this field can not be written by software, a reasonable attempt should be made by the codec manufacturer to provide a BMID that is different than the codec manufacturers' identification.

The Board SKU field is intended to be used by the manufacturer of the board where the function group (codec) is attached to identify the specific board design. The combination of BMID and BSKU should uniquely identify the audio design by that manufacturer. Although it is permissible for the BSKU to be the same for a family of boards that have identical audio hardware designs it is recommended that the SKU be different for each board.

Command Options:

Table 1. Implementation Idendification

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F20h[1]	0	Implementation Identification bits [31:0]
Set 1	720h	Implementation ID bits [7:0]	0
Set 2	721h	Implementation ID bits [15:8]	0
Set 3	722h	Implementation ID bits [23:16]	0
Set 4	723h	Implementation ID bits [31:24]	0

Applies to:

(a) Audio Function Group
(b) Modem Function Group
(c) Other Function Group

[1] The Verb Codes F21h, F22h, and F23h are reserved for the Implementation Identification register and must not be reassigned to anything else. However, they need not be implemented since standard software drivers will not use them. If a codec elects to respond to these codes, the response must be identical in all respects to the response to Verb Code F20h.