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: HDA012-A**
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: Clarification of CORB and RIRB pointer use

Brief description of the functional changes:

The definition in the HD Audio specification could be a clearer if examples were provided. This DCN is to add that clarification.

Current Definitions:

Section 4.4.1 of the Intel® High Definition specification version 1.0 defines:

Two pointers are maintained in the hardware, Write Pointer (WP) and Read Pointer (RP). WP is used by the software to indicate to the hardware the last valid command in the CORB, while the hardware uses RP to indicate to the software the last command that has been fetched. WP and RP both measure the offset into the buffer in terms of commands. Since commands are 4 bytes long, the byte offset into the CORB buffer indicated by RP or WP is WP*4 or RP*4.

To add commands to the CORB, the software places commands into the CORB at the end of the list of commands already in the list, which is at byte offset (WP + 1) * (4 bytes). When software has finished writing a new group of commands, it updates the WP to be equal to the offset of the last valid command in the buffer. When the CORB is first initialized, WP = 0, so the first command to be sent will be placed at offset (0 + 1) * 4 = 4 bytes, and WP would be updated to be 1.

Section 4.4.2 of the Intel® High Definition specification version 1.0 defines:

As with the CORB, a Read Pointer and a Write Pointer are used in the RIRB. In the RIRB, though, the RP is kept only by software to remember the last response the software read from the response buffer; there is no hardware representation of the RP. The Controller keeps a WP in hardware to indicate the offset of the last response which has been written into the response buffer. The Controller blindly writes to the RIRB whenever a response (solicited or unsolicited) is returned. As with the CORB, the WP indicates the offset in the response buffer in units of responses. Since each response is 8 bytes, the byte offset into the buffer is (WP * 8 bytes).

New Definition:

Portions of section 4.4.1 updated to:

Two pointers are maintained in the hardware, Write Pointer (WP) and Read Pointer (RP). WP is used by the software to indicate to the hardware the last valid command in the CORB, while the hardware uses RP to indicate to the software the last command that has been fetched. WP and RP both measure the offset into the buffer in terms of commands. Since commands are 4 bytes long, the byte offset into the CORB buffer indicated by RP or WP is WP*4 or RP*4.

To add commands to the CORB, the software places commands into the CORB at the end of the list of commands already in the list, which is at byte offset (WP + 1) * (4 bytes). When software has finished writing a new group of commands, it updates the WP to be equal to the offset of the last valid command in the buffer. When the CORB is first initialized, WP = 0, so the first command to be sent will be placed at offset (0 + 1) * 4 = 4 bytes, and WP would be updated to be 1.

Example hardware sequence for pointer usage:

```
Loop:
If (RP != WP) && ('run' bit is set) && ('link is running')
        RP++
        Get DWORD from buffer offset RP*4bytes
        Send DWORD (at beginning of next new frame)
goto Loop
```

Portions of section 4.4.2 updated to:

As with the CORB, a Read Pointer and a Write Pointer are used in the RIRB. In the RIRB, though, the RP is kept only by software to remember the last response the software read from the response buffer; there is no hardware representation of the RP. The Controller keeps a WP in hardware to indicate the offset of the last response which has been written into the response buffer. The Controller blindly writes to the RIRB whenever a response (solicited or unsolicited) is returned.   As with the CORB, the WP indicates the offset in the response buffer in units of responses. Since each response is 8 bytes, the byte offset into the buffer is (WP * 8 bytes).

Example hardware sequence for pointer usage:

```
Loop:
If ("new response received") && ('run' bit is set) && ('Valid bit' is set)
        WP++
        Write response + flags (QWORD) into buffer offset WP*8bytes
Goto loop
```