Intel® I/O Controller Hub 6 (ICH6) High Definition Audio / AC '97

Programmer's Reference Manual (PRM)

May 2005

Document Number: 302349-003

Contents

Figures

Tables

Revision History

Revision Number	Description	Date
-001	• Initial public release	June 2004
-002	• Added *Codec Features and Requirements* section	July 2004
-003	• Sections 7.3 and 8 removed. For information about these sections please contact your Intel representative.	May 2005

§

1 AC ’97 Programming Model

1.1 Intel® ICH6: AC ’97 Software

Device Name	Vendor ID	Device ID	Subsystem Vendor ID	Subsystem Device ID	Base Class Code	Sub-Class Code	Prog. Interface	Revision ID	Bus Number (PCI Addr)	Device Number (PCI Addr)	Function Number (PCI Addr)	Microsoft PNP Device Node ID	Intel Desired Device Description (INF name) Name for: Windows* 95 Windows* 98 Windows NT* Windows 2000*
Intel® ICH	8086	2415	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	5	PCI\VEN_8086&DEV_2415 (subsystem will also provide additional information)	Intel® 82801AA AC '97 Audio Controller (displayed by driver provider’s INF)
Intel ICH	8086	2416	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	07h	03h	00h	ALL	00h	1Fh	6	PCI\VEN_8086&DEV_2416 (subsystem will also provide additional information)	Intel 82801AA AC '97 Modem Controller (displayed by driver provider’s INF)
Intel ICH-0	8086	2425	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	5	PCI\VEN_8086&DEV_2425 (subsystem will also provide additional information)	Intel® 82801AB AC '97 Audio Controller (displayed by driver provider’s INF)
Intel ICH-0	8086	2426	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	07h	03h	00h	ALL	00h	1Fh	6	PCI\VEN_8086&DEV_2426 (subsystem will also provide additional information)	Intel 82801AB AC '97 Modem Controller (displayed by driver provider’s INF)
Intel ICH3	8086	2445	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	5	PCI\VEN_8086&DEV_2445 (subsystem will also provide additional information)	Intel® “ICH3” AC '97 Audio Controller (displayed by driver provider’s INF)
Intel ICH3	8086	2446	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	6	PCI\VEN_8086&DEV_2446 (subsystem will also provide additional information)	Intel® ICH3 DT/Server / Mobile/Low End” AC '97 Modem Controller (displayed by driver provider’s INF)
Intel ICH4	8086	24C5	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	5	PCI\VEN_8086&DEV_24C5 (subsystem will also provide additional information)	Intel® “ICH4” AC '97 Audio Controller (displayed by driver provider’s INF)
Intel ICH4	8086	24C6	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	ALL	00h	1Fh	6	PCI\VEN_8086&DEV_24C6 (subsystem will also provide additional information)	Intel® ICH4 DT/Server / Mobile/Low End” AC '97 Modem Controller (displayed by driver provider’s INF)
Intel ICH5	8086	24D5	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	00	00h	1Fh	5	PCI\VEN_8086&DEV_24C6 (subsystem will also provide additional information)	Intel® ICH5 DT/Server / Mobile/Low End” AC '97 Modem Controller (displayed by driver provider’s INF)

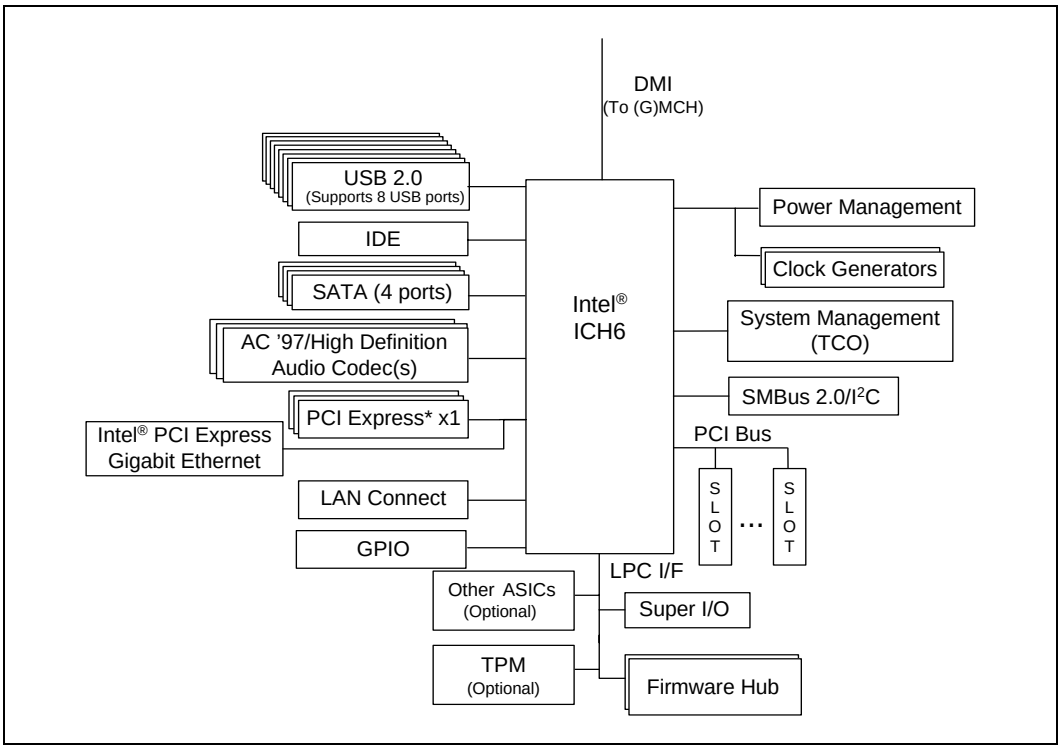
Device Name	Vendor ID	Device ID	Subsystem Vendor ID	Subsystem Device ID	Base Class Code	Sub-Class Code	Prog. Interface	Revision ID	Bus Number (PCI Addr)	Device Number (PCI Addr)	Function Number (PCI Addr)	Microsoft PNP Device Node ID	Intel Desired Device Description (INF name) Name for: Windows* 95 Windows* 98 Windows NT* Windows 2000*
Intel ICH6	8086	266E	Default is 00h. Value of this register varies according to the system	Default is 00h. Value of this register varies according to the system	04h	01h	00h	00	00h	1Eh	2	PCI\VEN_8086&DEV_24C6 (subsystem will also provide additional information)	Intel® ICH6 DT/Server / Mobile/Low End" AC '97 Modem Controller (displayed by driver provider's INF)

1.1.1 Introduction

This document was prepared to assist Independent Hardware Vendors (IHV) in supporting the Intel® I/O Controller Hub (ICH6) AC '97 Digital Controller feature set. This document also applies to the previous generation of Intel I/O Controller Hub components. New features for the ICH6 which are not supported in the ICH component have been distinguished by gray shading to improve readability. Please refer to the Applicable Component Table, above. This document also describes the general requirements to develop an audio mini-port driver that will make use of the AC '97 audio interface. The primary purpose of this document is to supplement the information provided in the *Intel® I/O Controller Hub 6 (ICH6) Family Datasheet* for use by IHVs and Intel customers developing their own driver interface.

This document also describes functions that the BIOS or Operating Systems (OS) must perform in order to ensure correct and reliable operation of the platform. This document will be supplemented from time to time with specification updates. The specification updates contain information relating to the latest programming changes. Check with your Intel representative for availability of specification updates.

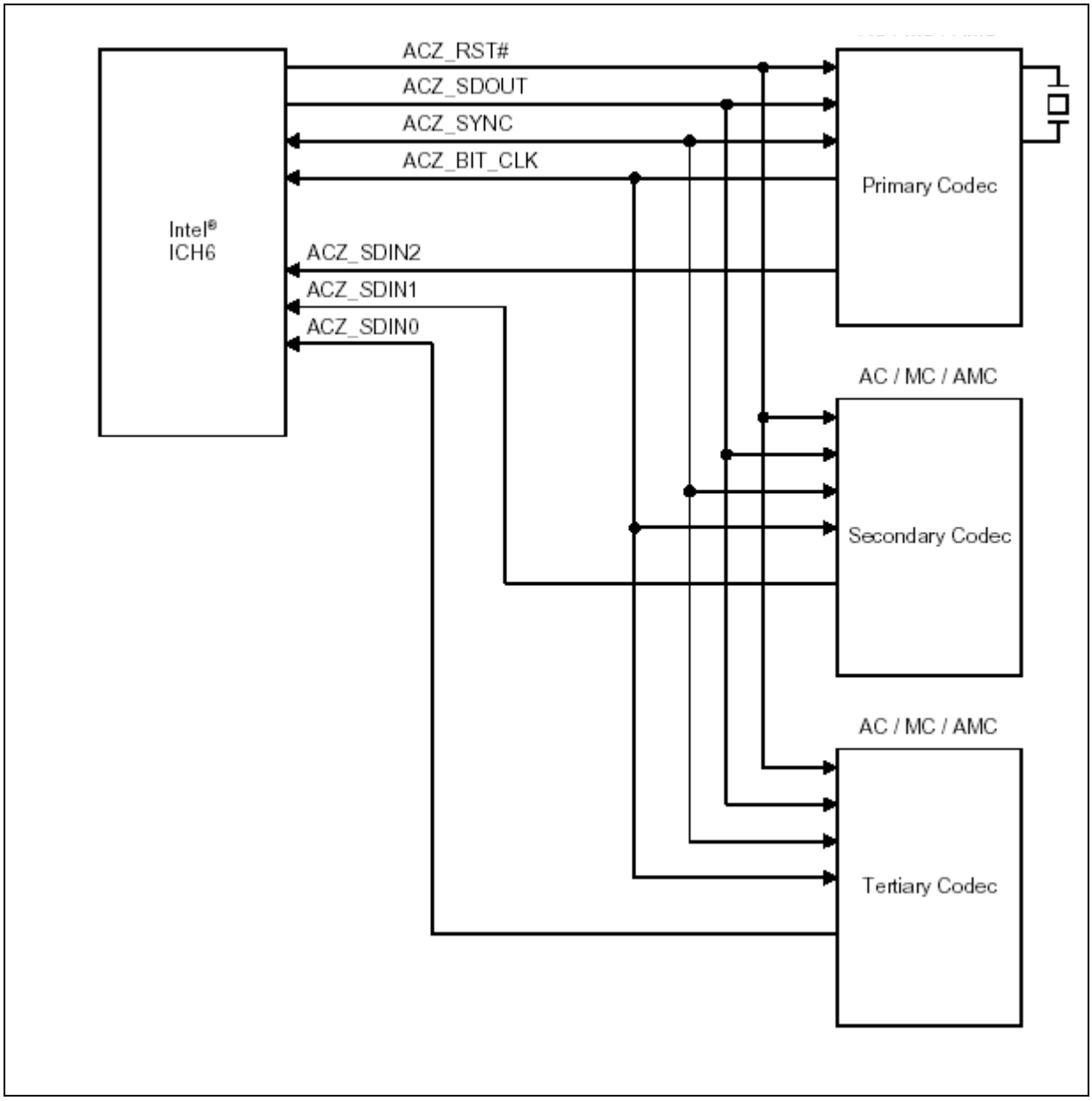
Figure 1-1. Block Diagram Intel® ICH6 Component



Note: In this document, "ICH6" stands for I/O Controller Hub 6. The ICH6 provides an AC '97-compliant controller. References to the "*AC '97 Component Specification*" refer to the *Audio Codec '97 Specificatio*n, Revision 2.1 the *Audio Codec '97 Specification*, Revision 2.2, and the *Audio Codec '97 Specification*, Revision 2.3. The ICH6 AC '97 controller implementation interfaces to *AC '97 Component Specification*, Revision 2.3-compliant codecs. The ICH6 supports up to three *AC '97 Component Specification*-compliant codecs on the AC-link interface. Figure 1-2, below, represents the typical configuration for the ICH6 AC '97 controller and companion codecs

1.1.2 Overview

Figure 1-2. Intel® ICH6 AC '97 Controller Connection to Its Companion Codecs



This scope of this document is limited to the specification of the software requirements and driver interface for the ICH6 AC '97 controller. Where possible, this document has pointers to additional considerations for supporting future proliferation or derivatives of the ICH6 controller. However, considerations for these future devices are subject to change. This document should be read in conjunction with the *Intel® I/O Controller Hub 6 (ICH6) Family Datasheet*.

1.1.3 Intel® ICH6 AC '97 Controller Compatibility

The ICH6 AC '97 controller is fully compatibility with the features found in the ICH1/2/3/4/5 versions. This allows for current drivers developed by ISVs and IHVs to work without modifications (see Section 1.1.3.4.) The ICH6 however, provides new capabilities not found in some of earlier ICH family of components. The following matrix provides a description of the available features for each of the ICHx components generation. This document specifically addresses new features on ICH6 while maintaining the original programming model reference for new developers working directly with ICH6 and not previously exposed to the ICH component.

Table 1-1. Audio Features Distribution Matrix

AC '97 Audio Controller Features	Intel® ICH	Intel® ICH2	Intel® ICH3	Intel® ICH4	Intel® ICH5	Intel® ICH6
16-bit Stereo PCM Output	X	X	X	X	X	X
16-bit Stereo PCM Input	X	X	X	X	X	X
16-bit Microphone Input	X	X	X	X	X	X
GPIO and Interrupt Support	X	X	X	X	X	X
Two 2.1/2.2/2.3 Codec Support	X	X	X	X	X	X
16-bit 2/4/6 Ch. Surround PCM Output		X	X	X	X	X
20-bit 2/4/6 Ch. Surround PCM Output				X	X	X
Dedicated S/P DIF DMA Output Ch.				X	X	X
Third 2.1/2.2/2.3 Codec Support				X	X	X
Memory Map Control and Status				X	X	X
Second 16-bit Stereo PCM Input				X	X	X
Second 16-bit Microphone Input				X	X	X
PCI 2.2 Power Management				X	X	X

The ICH6 AC '97 controller provides a set of new features that require significant software support. Section 1.1.3.1 through Section 1.1.3.6 provide summaries of these features.

The modem support infrastructure has not been changed in any generation of the I/O Controller Hub starting with the ICH.

1.1.3.1 AC '97 Component Specification 2.1, 2.2, 2.3 Compliant Codecs

The *AC '97 Component Specification* provides capability for up to four SDATA_IN signals for equal number of codec support. The ICH6 AC '97 controller provides support for up to three codecs to allow for Audio channel expansion without sacrificing the modem codec (MC) support. Also, the third codec capability enables a better mobile docking infrastructure.

1.1.3.2 Dedicated S/P DIF DMA Output Channel

The *AC '97 Component Specification*, Revision 2.3 provides the capability of steering an S/P DIF stream into the PCM channels for pass-through to a CE audio amplifier. The ICH6 provides this capability with an independent DMA channel that allows for a stream autonomous from the PCM audio available in other channels. This opens the possibility for an AC3 stream provided by DVD playback independent from system audio messages.

1.1.3.3 20 Bits Surround PCM Output

The *AC '97 Component Specification* provides a maximum bit resolution of 20 bits per sample. The ICH6 AC '97 controller DMA Engine fully exploits this capability to improve the audio output quality.

1.1.3.4 Memory Map Status and Control Registers

The ICH6 introduces a new PCI Memory Base Address register that allows for higher performance access to the controller registers while expanding the register space to access the new third codec support mechanism. All access and features can know be access via this new Memory BAR making the I/O Bar capabilities obsolete. However, the ICH6 controller maintains the I/O BAR capability to allow for the reuse of legacy code maintaining backward compatibility to deployed driver binaries.

Note: This document describes the programming interface using the new Memory BAR registers unless otherwise indicated. For usage under the Compatible Mode registers please refer to the *Intel® I/O Controller Hub 6 (ICH6) Family Datasheet*.

The default configuration for ICH6 Audio function is to use the PCI Memory Base Address register. The I/O BAR is therefore disabled unless system BIOS enables the simultaneous backward compatible capability on register:

Device 31 Function 5 Audio			
Offset	**Register**	**Default**	**Comments**
41h	CFG Configuration	00h	When cleared, the I/O space BARs at offset 10h and 14h become Read-Only registers. This is the default state for the I/O BARs. Initialized by BIOS when backward I/O Bar compatibility is required Memory BARs are always enable.

1.1.3.5 Second Independent Input DMA Engines

The ICH6 provides two new sets of input DMA engines that allow for the secondary or tertiary codecs to provide recording PCM data streams on the primary codec while simultaneously providing recording capabilities from the secondary or tertiary codec. A typical application is to provide independent input stream in am mobile docking configuration where an audio codec is located in the base system (notebook unit) and the secondary or tertiary codec is located in a docking unit for desktop replacement. The new DMA engines provide the infrastructure for s/w to select the input stream from either source for stereo or microphone recording. Also the capability of simultaneous input streams opens the possibilities for more futuristic applications where a microphone array can be created using two codecs.

1.1.3.6 PCI Local Bus Specification, Revision 2.2 Power Management

The ICH6 introduces *PCI Local Bus Specification*, Revision 2.2-compliant Power Management registers that allows for better OS power management support with reduced overhead to the BIOS programmers using ACPI control methodologies.

1.1.4 General Requirements

It is assumed that the reader has a working knowledge of AC '97 architecture and the ICH6 AC '97 controller implementation. Also, the reader should have an understanding of audio driver development for the target operating systems.

This document outlines the software specification for the AC '97 controller, and also includes details on the development of an audio device driver that will be used in Microsoft Windows* 98 family, Windows NT*, Windows* 2000 and Windows* XP, based on the Windows* Driver Model (WDM) interface.

1.2 Intel® ICH6 AC '97 Controller Theory of Operation

The ICH6 AC '97 controller interface is an implementation of the AC-link, with additional features to support the transaction and device power management. The ICH6 AC '97 controller includes DMA engines for high-performance data transfer to memory via a hub interface.

ICH6 AC '97 controller and AC-link support isochronous traffic, which emphasizes the timing of the data. This is critical to maintain the data stream from the audio and/or modem codec.

1.2.1 Intel® ICH6 AC '97 Initialization

1.2.1.1 System Reset

The ICH6 AC '97 circuitry is reset on power up by combining the PCIRST# signal with the AC Link RESET# signal. However, AC Link RESET# will not follow PCIRST# during a resume from sleep condition. During operation, the system can be reset by clearing the AC '97 cold reset bit in the Global Control/Status register (GLOB_CNT). This bit is maintained during ICH6 sleep mode and can be used by the driver to select warm or cold reset during a resume condition. If the codec is not present, i.e., ICH6 AC '97 is not supported, codec ready will never be seen by the controller. Once the reset has occurred, a read to Mixer register 00h/80h will indicate what type of hardware resides in the codec(s).

Software must ensure that codec ready bit is present for the appropriate Global Control/Status register (GLOB_CNT). Before writing any value on the Codec registers or initiating a DMA transfer, s/w must ensure that the analog portion of the codec has reached a ready status by reading the Audio Codec Register Power-Down Control/Status register (Index 26h) or Extended Modem Status and Control register (Index 3Eh) correspondingly.

1.2.1.2 Codec Topology

The following rules present the allowable codec configuration when attaching to the AC-link interface. To avoid improper driver loading, the system BIOS should determine the presence of the audio or modem codec attached on the AC-link, and enable the Audio or Modem function's PCI configuration space accordingly.

The following are the loading rules for ICH6:

1. Maximum of three codecs total on the link
2. Maximum of a single modem function, either as Modem Codec or a combination Audio/ Modem Codec

This information is used to disable (hide) the appropriate PCI function. To determine that a codec or codecs are attached to the link, the System BIOS follows an algorithmic approach (See the *Intel® I/O Controller Hub 6 (ICH6) Family Datasheet* for algorithm details and sample code).

Drivers can distribute output and input data in appropriate slots on available codec. For example a 6-channel data stream can be separate into three, 2-channel codecs, as long as the codecs are programmed to decode the appropriate slot output stream (SDATA_OUT). Similarly ICH6 provides two Stereo PCM input channels as well as two Microphone mono input DMA channels. These allow for separate input streams for stereo PCM and microphone recording from two different codecs simultaneously.

Software should match sample rates, when two codecs are teamed together. The codecs must have matching vendors, types, and be explicitly supported in software. Essentially, audio codecs must be programmed with a common sample rate. The selection of a common sample rate is based on each codec's capabilities.

1.2.1.3 BIOS PCI Configuration

The ICH6 AC '97 controller as previously indicated, exposes two PCI functions in the ICH6 (Bus 0, Device 30h). This allows for driver differentiation between these capabilities in the component.

- Function 2: ICH6 AC '97 Audio Controller
- Function 3: ICH6 AC '97 Modem Controller

As PCI devices there are a number of registers that are required to be initialized to enable these functions. The following table summarizes these requirements.

Table 1-2. Audio Registers (Sheet 1 of 2)

Device 31 Function 5 Audio			
Offset	**Register**	**Default**	**Comments**
04h-05h	Command (COM)	0000h	Bit 2: Bus Master Enable Bit 1: Memory Space Enable Bit 0: When enable in 41h I/O Space Enable
10h-13h	Native Audio Mixer Base Address (NAMBAR)	00000001h	When enable in 41h Address in the 64-K I/O space that allows 256 bytes of registers not in conflict with any other set
14h - 17h	Native Audio Bus Mastering Base Address (NABMBAR)	00000001h	When enable in 41h Address in the 64-K I/O space that allows 256 bytes of registers not in conflict with any other set
18h – 1Bh	Memory Audio Mixer Base Address (MMBAR)	00000000h	Address in the 4-GB memory space that allows 512 bytes of registers not in conflict with any other set

Table 1-2. Audio Registers (Sheet 2 of 2)

1Ch – 1Fh	Memory Bus Master Base Address Register (MBBAR)	00000000h	Address in the 4-GB memory space that allows 512 bytes of registers not in conflict with any other set
3Ch	Interrupt Line (INTLN)	00h	A hardware interrupt (0-Fh) that follows value assigned to PIRQB#. Has not effect on ICH6 it is used to indicate software the IRQ value assigned to the device.
41h	CFG Configuration	00h	When cleared, the I/O space BARs at offset 10h and 14h become read only registers. This is the default state for the I/O BARs. Initialize by BIOS when backward I/O Bar compatibility is required Memory BARs are always enable.

Table 1-3. Modem Registers

Device 31 Function 6 Modem			
Offset	**Register**	**Default**	**Comments**
04h-05h	Command (COM)	0000h	Bit 2: Bus Master Enable Bit 0: I/O Space Enable
10h-13h	Native Audio Mixer Base Address	00000001h	Address in the 64-K I/O space that allows 256 bytes of registers not in conflict with any other set
14h - 17h	Native Audio Bus Mastering Base Address	00000001h	Address in the 64-K I/O space that allows 256 bytes of registers not in conflict with any other set
3Ch	Interrupt Line (INTLN)	00h	A hardware interrupt (0-Fh) that follows value assigned to PIRQB#. Has not effect on ICH6 it is used to indicate software the IRQ value assigned to the device.

With the exception of register 41h on Device 30 Function 2, initialization of the PCI registers above is the responsibility of the PnP capable OS. If a PnP OS is not available in the system, it is then the BIOS's responsibility to configure all PCI devices including the registers above. Determination of the presence of PnP capable OS is usually made via a switch in the System Setup. However, the final configuration or the existence or not of this switch is implementation dependent.

The ICH6 AC '97 controllers also provide PCI Power Management functionality. PCI Power Management registers are available via the configuration space. Handling of the Power Management registers is responsibility of the OS PCI Bus driver following standard procedures. For further discussion on the usage model for this registers please review the power management section of this document.

1.2.1.4 Hardware Interrupt Routing

The audio and modem functions in the ICH6 internally share the same PCI IRQ (PIRQB#). The configuration software must take this into account and assign the same IRQ pin to both functions. Sharing IRQs increases the ISR latencies. Each ISR must determine if the interrupting device is the one serviced by the routine, as determined by the OS programming model. PIRQB# it is also exposed as a PCI IRQ.

In an environment where a high Quality of Service (QoS) is required, system designers must pay close attention to devices attached to the same PIRQ. Software driven signal processing functions, as in the case of software driven modem and audio, require maintaining a low latency interrupt service in order to maintain proper functionality. Software driver programmers need to pay close attention to the ISR latencies and make use of Deferred Procedure Calls (DPC) as much as possible.

1.2.2 DMA Engines

The ICH6 AC '97 controller uses a scatter gather mechanism to access memory. There are five, 16-bit DMA engines for Audio: two PCM Stereo In, two MIC mono in, and S/P DIF Out. There is one,20-bit PCM 2/4/6 channel surround. There are two, 16-bit DMA engines for Modem: In and Out. Audio and Modem registers are located in two separate PCI functions in the ICH6 components to allow for driver development flexibility.

1.2.2.1 Buffer Descriptor List

The buffer descriptor list is an array of up to 32 entries, each of which describes a data buffer. Each entry contains a pointer to a data buffer, control bits and the length of the buffer being pointed to the length is expressed as number of samples. The buffer length is restricted to 65536 samples at 16 or 20 bits per sample. A value of "0" in the buffer length indicates no samples to process. Each descriptor can point to a buffer of a different size.

Figure 1-3. Generic Form of Buffer Descriptor (One Entry in the List)

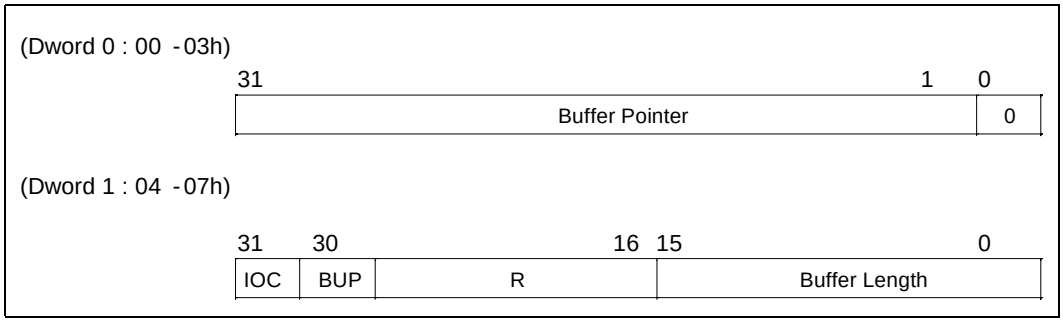


Table 1-4. BD Buffer Pointer (DWORD 0: 00-03h)

Bit	Description
31:1	**Buffer pointer**. This field points to the location of the data buffer. Since the samples can be as wide as 1 word, the buffer needs to be aligned to word boundaries to avoid having samples straddle dword boundaries.
0	**Reserved**. Must be 0 when writing this field.

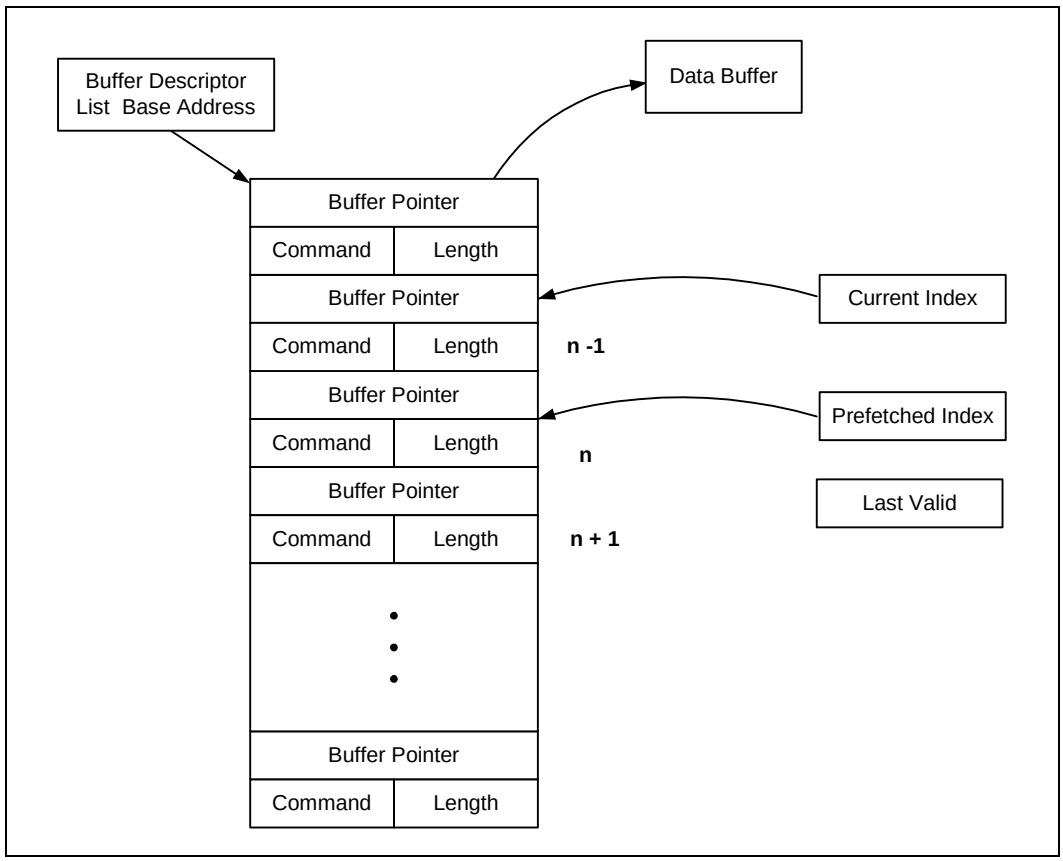
Table 1-5. BD Control and Length (DWORD 1: 04-07h)

Bit	Description
31	Interrupt on Completion (IOC) Disable. When this it is set, it means the controller should issue an interrupt upon completion of this buffer. It should also set the IOC bit in the Status register Enable
30	Buffer Underrun Policy (BUP) When this buffer is complete, if the next buffer is not yet ready, (last valid buffer has been processed) then continue to transmit the last valid sample. When this buffer is complete, if this is the last valid buffer, transmit 0's after this buffer is completely processed. This bit will typically be set only if this is the last buffer in the current stream.
29:16	**Reserved**. Must be 0 when writing this field.
15:0	**Buffer length**. This is the length of the data buffer in number of samples. The controller uses this data to determine the length of the buffer in bytes. A value of 0 indicates **no sample** to process.

1.2.2.2 DMA Initialization

The maximum length of the buffer descriptor list is fixed at 32 (this is limited by the size of the Index registers). The figure below describes the organization of the buffer descriptor list.

Figure 1-4. Buffer Descriptor List



The following steps describe the driver initialization process for a single DMA engine. The same process should be repeated for each DMA engine.

1. Create the buffer descriptor list structure in memory (non-paged-poll).
2. Write the Buffer Descriptor List Base Address register with the base address of the buffer descriptor list.

Table 1-6. Audio Descriptor List Base Address

Audio Buffer Descriptor List Base Address	I/O Address
PCM IN	MBBAR + 00h (PIBAR)
PCM OUT	MBBAR + 10h (POBAR)
MIC	MBBAR + 20h (MCBAR)
MIC 2	MBBAR + 40h (M2DBAR)
PCM2 IN	MBBAR + 50h (PI2BAR)
SPBAR	MBBAR + 60h (SPBAR)

Table 1-7. Modem Descriptor List Base Address

Modem Buffer Descriptor List Base Address	I/O Address:
Line IN	MBAR + 00h (MIBDBAR)
Line OUT	MBAR + 10h (MOBDBAR),

3. Set up the buffer descriptors and their corresponding buffers. Buffers are passed to the miniport driver as Memory Descriptor Lists (MDL). These MDL describe the physical page address of the virtual audio buffer. Multiple buffer descriptors may be required to represent a single, virtual buffer passed to the miniport driver
4. Once buffer descriptors are set in memory, software writes the Last Valid Index (LVI) register.

Table 1-8. Audio Last Valid Index

Audio Last Valid Index (LVI)	) I/O Address
PCM IN	MBBAR + 05h (PILVI),
PCM OUT	MBBAR + 15h (POLVI),
MIC	MBBAR + 25h (MCLVI)
MIC 2	MBBAR + 45h (M2LVI)
PCM2 IN	MBBAR + 55h (PI2LVI)
SPBAR	MBBAR + 65h (SPLVI)

Table 1-9. Modem Last Valid Index

Modem Last Valid Index (LVI)	I/O Address:
Line IN	MBAR + 05h (MILVI)
Line OUT	MBAR + 15h (MOLVI),

5. After LVI registers are updated, software sets the run bit in the Control register to execute the descriptor list.

1.2.2.3 DMA Steady State Operation

Software has two concurrent activities to perform while in normal operation: Preparing new buffers/buffer descriptors and marking processed buffer descriptors and buffers as free. Once the run bit is set in the Bus Master Control register bit 0, the bus master fetches the buffer descriptor.

1. Bus master starts processing the current buffer. Once current buffer is processed, depending upon the bits set in the command field, the interrupt is asserted and the interrupt bit is set.
2. Bus master increments the current and prefetch indices. It then starts executing the current buffer and schedules the next buffer to be prefetched.
3. Buffer service routine maintains a variable which points to the head of the list of descriptors to be processed. The descriptor list service routine performs the following activities:

```
// Update head of descriptors to be processed
        While (head!= current_index)
        {
            Mark head free;
            // check for end of descriptor list
            If head == base_address + (31 * 8);
                // last entry on the list, set head to top of the list
                head = base_address;
            Else
                // still inside the list, increment head to next entry
                head++;
```

Caution: This algorithm needs to be optimized to reduce the number of memory accesses during execution. The "while" statement could translate to several memory access if this code is not execute after each buffer descriptor update.

Also, the routine that prepares buffers maintains a variable that points to the entry *after* the tail of the list. This value is always the next entry after the LVI register. It follows the following algorithm:

```
// Update tail of descriptor list ready for execution and audio buffers // when
available for processing
        While ((tail == free) && (buffers_available > 0))
        {
            Prepare buffer descriptor indexed by tail;
            buffers_available--;
            //assign tail to Last Valid Index
            LVI = Tail;
            // check for end of descriptor list
            If (tail == base_address + 31 * 8);
                // last entry on the list, set tail to top of the list
                tail = base_address;
            Else
                // Advance tail to next value
                tail++;
            }
```

1.2.2.4 Stopping Transfers

There are two ways that transfers could be stopped:

1. By simply turning off the Bus Master run/pause bit. This will halt the current DMA transfer immediately. Data in the output FIFOs will continue to be read out until they empty. The registers will retain their current values and AC-link corresponding slots will be invalidated. Setting the run/pause bit will resume DMA activity.
2. Software can stop creating new buffers and hence not update the LVI register. The bus master will stop once the last valid buffer has been processed. All register information is maintained. During this condition the controller will transmit the last valid sample or 0's pending the status of the Buffer Underrun Policy (BUP) bit in the buffer descriptor entry. If the run/pause bit remains set, then any future update to the LVI register will cause the bus master operation to resume.

Note: **Software must ensure that the DMA controller halted bit is set before attempting to reset registers.**

1.2.2.4.1 FIFO Error Conditions

Two general conditions could result in the FIFO error bit 4 in the Status register being set. Pending the status of bit 3 in the control register it will also cause an interrupt.

1.2.2.4.2 FIFO Underrun

FIFO underrun will occur when the ICH6 AC '97 controller FIFO is drained:

1. As a result of system congestion. The DMA read transaction could still be pending as data has not returned from memory. In this case the controller will repeat last sample until new data is available in the FIFO.
2. As a result of DMA engine reaching the LVI, no further access to memory, therefore FIFO will drain. In this case the controller will transmit the last valid sample or 0's pending the status of the Buffer Underrun Policy (BUP) bit in the buffer descriptor entry. This condition is an error if software is not able to update the descriptor list before the DMA engine reaches the LVI. However, this condition could be as result of the completing processing the last buffer. It is up to the software driver to determine the final status of this condition. See Stopping Transfers, Section 1.2.2.4.

1.2.2.4.3 FIFO Overrun

FIFO overrun will occur when valid data is transmitted in proper AC-link slots and DMA FIFO remains full. Two conditions could result in the FIFO error bit 4 in the Status register being set. Pending the status of bit 3 in the Control register it will also cause an interrupt.

1. As a result of DMA engine not being able to update system memory with the content of the FIFO. This is a result of system congestion. In this case, all new samples received from the AC-link will be lost.
2. As a result of the DMA engine reaching the LVI, no further access to memory, therefore FIFO will not drain. This condition is an error if software is not able to update the descriptor list before the DMA engine reaches the LVI. However, this condition could be the natural result of the last buffer entry been processed. It is up to the software driver to determine the final status of this condition. See Stopping Transfers, Section 1.2.2.4.

1.2.3 Channel Arbitration

It is possible for up to eight ICH6 AC '97 DMA channels to be enabled at one time. A round-robin arbitration scheme is used to arbitrate between these. channels.

1.2.4 Data Buffers

1.2.4.1 Memory Organization of Data

Samples are packed in two samples for 2 channel (stereo), four samples for 4 channels surround and six samples for 6 channels surround.

The actual PCM data is “left-aligned” within the container. The sample itself is justified most significant; all extra bits are at the least-significant portion of the container. All non-valid data bits must be set to 0. With 20-bit data, the “top” 20 bits (31-12) of the dword contain the data. The bottom 12 bits (11-0) are not valid data.

The data must be sample aligned. 16-bit data would be word aligned and but 20-bit data is dword aligned, hence a 20-bit sample cannot start at the upper word of a dword.

1.2.4.2 PCM Buffer Restrictions

Below are the memory buffer restrictions for ICH6 PCM that applies for 2-, 4-, and 6-channel audio mode:

1. Buffer Descriptors Must Contain Integer Multiples of Framed Samples and Are Frame Aligned:

   Example: Two channel buffers must have a multiple of two samples. Four channel buffers must have 4, 8, 12. . . samples and six buffers channels have 6, 12, 18. . . samples. The controller does not support a frame (e.g., left and right samples for 2 channel) spanning multiple descriptors. Similarly, the controller does not support a buffer descriptor with a single sample (PCM out MONO is not supported). Also, odd length buffers are not allowed due to the sample alignment requirements.

2. Software Is Allowed to Create an Empty Frame (0 Samples) in a Buffer Descriptor with the Following Restriction:

   An empty buffer has to be part of a list of buffer descriptors and it cannot to be the first Buffer Descriptor or the last Buffer Descriptor of the list. A series of buffer descriptors with 0 samples are possible in the lists as long as they are not the first or the last. The last Buffer Descriptor in the list is determined by the LVI that is programmed.

1.2.4.3 FIFO Organization

The ICH6 AC '97 controller supports 16-bit samples on all channels-with exception of PCM Out, which also supports 20-bit samples.

Data will be written to the FIFO in sample pairs following the order of valid slots in a channel. For example, for audio PCM in, the controller will check the first valid slot and add it to the FIFO first entry as a word (16 bits). The next valid slot will be added as the second word entry in the FIFO to create the PCM stereo sample pair. This behavior works under the assumption that the first valid slot will be always the left channel (slot 3) followed by right channel in slot 4 in the same or subsequent frame. If the codec transmits data repeating the slot, it will cause the controller to

misplace the sample in the FIFO. Codecs compatible with the ICH6 AC '97 implementation should always maintain the indicated order, and never use the same slot twice to transmit samples to the controller. Figure 1-5 through Figure 1-7 present some ICH6 compatible and incompatible implementations, using as a reference a two-channel implementation.

Figure 1-5. Compatible Implementation with Left and Right Sample Pair in Slot 3/4 Every Frame

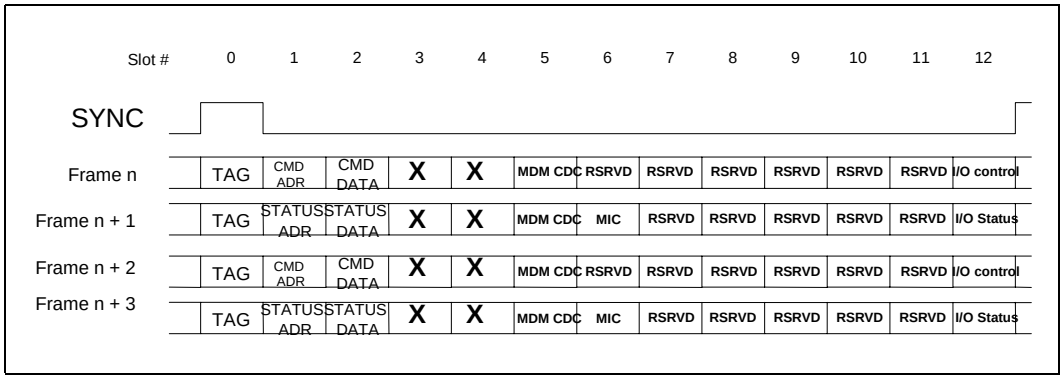


Figure 1-6. Compatible Sample Rate Conversion Slots 3 and 4 Alternating over Next Frame

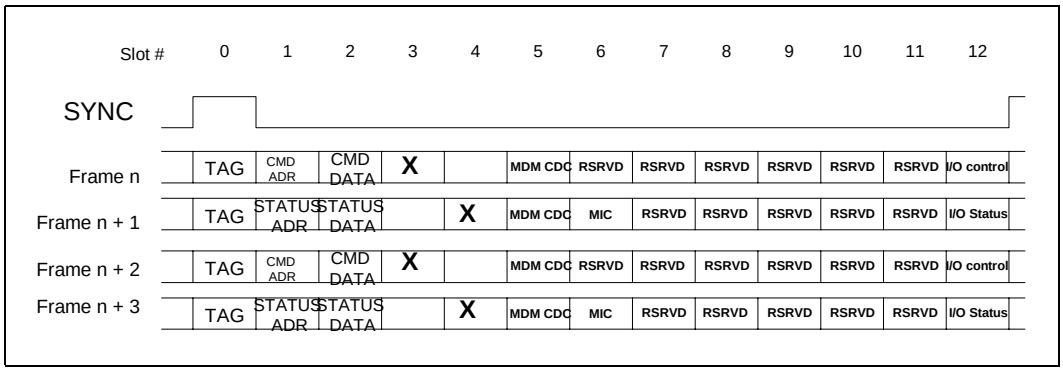


Figure 1-7. Incompatible Sample Rate Conversion with Repeating Slots over Next Frames

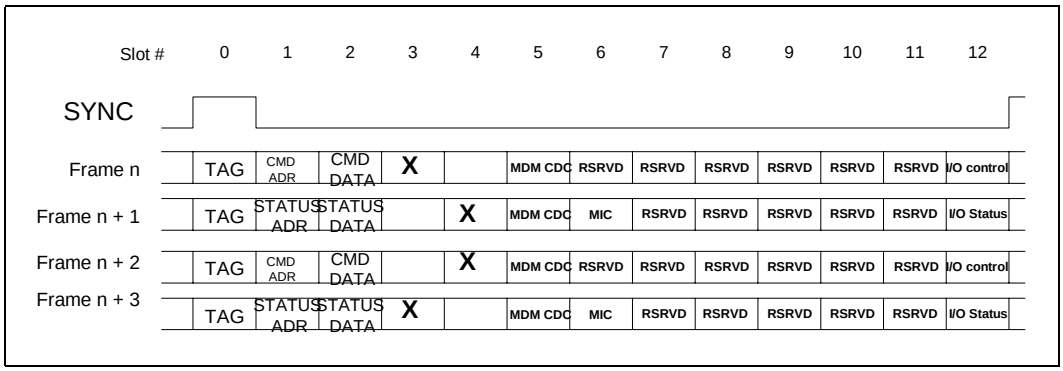


Table 1-10. FIFO Summary

Channel	# of Samples	FIFO Depth	FIFO Width	Comments
Mic In	1	2	32 bits	2 frame[1] dword
PCM In	2	4	32 bits	1 frames per dword
PCM Out	6, 4 or 2	24	32 bits	1 frames per dword (2 –16 bits ch), 1 frame per 2 dword (2 –20 bits ch) 1 frame per 2 dword (4 –16 bits ch) 1 frames per 4 dword (4 –20 bits ch) 1 frames per 3 dword (6 – 16 bits ch.) 1 frames per 6 dword (2 –20 bits ch)
Modem In	1	2	32 bits	2 frames dword
Modem Out	1	2	32 bits	2 frames dword

NOTES:
1. One audio frame worth of data for the specific DMA channel.

1.2.5 Multiple Codec/Driver Support

The ICH6 AC '97 controller is capable of supporting a three-codec implementation. Under this implementation all codecs share the SDATA_OUT signal while independent SDATA_IN[0:2] are used by the codec to supply data to the controller. ICH6 allows for a compatible behavior, where the three SDATA_IN are used, these signals are logically OR'd inside the digital controller, effectively creating one digital input data stream. However, ICH6 also allows for an independent SDATA_IN functionality. On independent functionality, the SDATA_IN Map Register (SDM) MBBAR + 80h is used to steer the content of the input slots to the appropriate controller DMA engine. This capability also allows for a more reliable enumeration algorithm of the available codecs.

Table 1-11. SDM Register Description

Bit	Type	Reset	Description
7:6	RW	00	PCM In 2, Microphone In 2 Data In Line (D21L): When the SE bit is set, these bits indicates which SDATA_IN line should be used by the hardware for decoding the input slots for PCM In 2 and Microphone In 2. When the SE bit is cleared, the value of these bits are irrelevant, and PCM In 2 and Mic In 2 DMA engines are not available. 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved
5:4	RW	00	PCM In 1, Microphone In 1 Data In Line (DI1L): When the SE bit is set, these bits indicates which SDATA_IN line should be used by the hardware for decoding the input slots for PCM In 1 and Microphone In 1. When the SE bit is cleared, the value of these bits are irrelevant, and the PCM In 1 and Mic In 1 engines use the OR'd SDATA_IN lines. 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved
3	RW	0	Steer Enable (SE): When set, the SDATA_IN lines are treated separately and not OR'd together before being sent to the DMA engines. When cleared, the SDATA_IN lines are OR'd together, and the "Microphone In 2" and "PCM In 2" DMA engines are not available.
2	RO	0	Reserved
1:0	RO	00	Last Codec Read Data Input (LDI): When a Codec register is read, this indicates which SDATA_IN the read data returned on. Software can use this to determine how the codecs are mapped. The values are: 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved

1.2.5.1 Codec Register Read

Codec register reads are presented in the AC-link in the next available frame after the controller receives the I/O transaction. Data will be returned to the controller pending codec availability. To avoid longer latencies than necessary, the codec must return data in the next available frame. Multiple frame transactions impose large system latencies, to the detriment of system performance.

Even when data is returned in the frame immediately after the read request is presented in the AC-link, the minimum latency is still on the order of 40 µs. To minimize the effect on the system caused by long latencies in the AC-link, software drivers must maintain a copy of the Codec register in memory (shadow) and use this data instead of accessing the codec.

Shadowing in memory is effective as long as the codec does not change the value of the registers itself. Therefore, the status of the GPIOs configured as inputs on the most recent frame is accessible to software by reading the register at offset 54h in the modem codec I/O space. Only the 16 MSBs are used to return GPI status. Reads from 54h will not be transmitted across the link. Instead, data received in slot 12 is stored internally in the controller, and the data from the most recent slot 12 is returned on reads from offset 54h.

The power-down in codec offset 26h and 3Eh Status registers are not supported by an automatic shadowing mechanism, as is the case for offset 54h. However, these registers are sparingly used. These registers are read only during power-down status determination.

Finally, codec ready status is required during system initialization. It is automatically reflected in the Global Status register (GSR) at *MBBAR + 30h (MBAR + 40h)*—bit 8, for the primary codec, bit 9, for the secondary, and bit 28, for the third codec. These three bits need not be saved in memory.

1.2.5.2 Codec Access Synchronization

All Codec register writes are posted transactions in the ICH6 AC '97 controller. The ICH6 AC '97 controller will indicate transaction completion to the host processor immediately following the request even when the transaction is actually pending for completion in the AC-link. This is done to improve system performance. However, it also imposes restrictions in the driver(s) operation. Also, register reads present synchronization issues.

Before a Codec register access is initiated, the driver must check the status of the codec access in progress (CAIP) bit 0 in the Codec Access register at MBBAR + 34h (MBAR + 44h.) If no write is in progress, this bit will be 0 and the act of reading the register sets this bit to 1. This reserves the driver the right to perform the I/O read or write access. Once the write is complete, hardware automatically clears the bit. The driver must also clear this bit if it decides not to perform a codec IO write after having read this bit. If the bit is already set, it indicates that another driver is performing a codec I/O writes across the link and the driver should try again later.

1.2.5.3 Data Request Synchronization in Audio Split Configurations

To support more than two channels of audio output, the *AC '97 Component Specification*, Revision 2.1 allows for a configuration where up to three audio codecs work concurrently to provide surround capabilities (refer to ac '97 component specification, Revision 2.2) To maintain data on demand capabilities the Intel controller, when configured for 4 or 6 audio channels, will wait for all the appropriate slot request bits to be set before sending data in the SDATA_OUT slots. This allows for a simple FIFO synchronization of the attached codecs.

If the codecs on the link are not compatible, or are not known to be compatible, with respect to sample rate conversion algorithms and FIFO depth requirements (for instance, all codecs being the same revision of the same model from the same vendor), Variable Rate support should not be used, and a fixed sample rate of 48 MHz is recommended to maintain synchronization across the codecs in use.

1.2.6 Power Management

Power management of the driver/codec interaction requires careful sequencing in the ICH6 AC '97 environment. In the ICH6 AC '97 environment it is possible to have two drivers sharing the same AC-link interface for two separate codecs or a single driver controlling two separate audio codecs. A driver forcing an aggressive sleep state in the link could have functional repercussions on the pairing codec. The Deep Sleep state in a device following ACPI compliance requirements is the D3 State. When a driver is requested to set its device to the D3 State, the driver should enter the most aggressive power saving mode possible. The D3 State is also often the precursor to a system wide core power removal. Therefore several considerations must be taken into account to maintain the device functionality and wake-up capability.

The procedure to be taken by an ICH6 AC '97 device driver varies depending on the system configuration. The following table enumerates the possible codec combinations supported by the ICH6 AC '97 controller.

The ICH6 audio/modem controller supports a maximum of two codecs. The following system implementations are possible (see Figure 4-5, "Response Inbound Ring Buffer" on page 109 for details.)

Table 1-12. Codecs Topologies

Config.						
1	AC	(Primary)				
2	MC	(Primary)				
3	AMC	(Primary)				Possible D3 state interactions
4	AC	(Primary)	+	MC	(Secondary)	Possible D3 State interactions
5	AC	(Primary)	+	AC	(Secondary)	Driver interaction concern
6	AMC	(Primary)	+	AC	(Secondary)	Possible D3 State Interactions
7	AC	(Primary)	+	AMC	(Secondary)	Possible D3 State Interactions

Note: The configuration above could be further limited by ICH6 AC '97 riser card configuration and loading. Refer to *Audio/Modem I/O Riser Specification* for details.

It is evident that Configurations 1 and 2, above, require no driver synchronization between ICH6 AC '97 codecs. Configuration 1 and 2 are single codec topologies, and therefore an aggressive power saving mode is possible including disabling of the actual AC-link without concern of affecting paired codec functionality.

Configuration 3 is a single codec topology that provides both functions audio and modem. In this configuration driver interaction is also critical if a separate set of drivers are in control of the audio and modem functions.

Configuration 4, however, is a two-codec topology. In this configuration an aggressive power saving mode requires detailed attention to cross interactions and the effect on AC-link functionality.

Configuration 5 is a two-codec audio topology. In this configuration concerns are on the proper power down sequence. However, no driver interaction is expected as only the audio driver executes power management functions.

Configuration 6 is also a two-codec topology with split audio and integrated modem support. This is the most complex interaction, as two different sets of driver will be operating in a complex topology.

Configuration 7 is identical to configuration 6, with the primary and secondary codecs switched.

In order to power manage ICH6 AC '97 codecs, there are two sets of PR bits that drivers need to be managed. One set is at offset *MMBAR + 26h* in the audio function, mapped to offset 26h in the primary codec, and the second set is at *MMBAR + 3Eh*, mapped to offset 3Eh in the modem function. Notice that register 3Eh does not provide link down functionality, this is provided in register 56h bit 12, (MLNK) modem link.

1.2.6.1 Power Management Transition Maps

The following paragraphs relate power management transition maps in the constraints of an ACPI system environment. Table 1-13 and Table 1-14 map codec PR bits transitions to specific ACPI "D" states for the device.

The following considerations were made in the generation of the following tables:

- Power management is defined in the framework of a desktop system. Further power savings are possible by implementing more aggressive power management typical of mobile environment policies (see aggressive power management section below). However these power savings are a trade off involving driver complexities and functional restrictions.
- Selection of a specific power policy below is pending the proper identification of the topology by the driver(s).
- Secondary codec is provided with external clocking mechanism and is not dependent on BIT_CLK to drive internal state machines when in power down mode.
- After warm or cold reset the device driver will bring all PR(x) bits to D0 state.
- Transition from/to any Dx state is accomplished by setting/resetting all appropriate PR(x) bit simultaneously. Codec should not place limitations on the PR(x) bits transition sequence represented above.
- Audio Codec Reg. 26h D15 EAPD (formerly PR<7> enable/disable function) is newly defined as control for an external audio power amp. Audio codec should provide an audio amp output pin (GPO) that provides off/on capability following this bit set/reset status.
- The modem tables assume Caller-ID capability during wake-up on ring, so VREF is on during D3.
- Modem D3 configuration is dependent on wake-up on ring event enable. If wake-up on ring is enabled, GPIO cannot go down in D3.

Note: When a codec section is powered back on the Power-Down Control/Status register (index 26h) should be read to verify that the section is ready before attempting any further operations.

Configuration Number 1: Single Audio Codec (Primary):

Table 1-13. Power State Mapping for Audio Single or Dual (Split) Codec Desktop Transition

PR<0:5> + (EAPD)								+12	+5 from +12	+3.3 Digital	+3.3 Vaux Digital	Comments
	EAPD	CLK	AC-Link	Mixer VREF.	Mixer	DAC	ADC					
Device State	7	5	4	3	2	1	0					
D0	0	0	0	0	0	0	0	On	On	On	On	All on
D1	0	0	0	0	0	***1***	***1***	On	On	On	On	-DAC, -ADC
D2	***1***	0	0	0	***1***	1	1	On	On	On	On	-Mix, -Amp
D3	1	***1***	***1***	***1***	1	1	1	Off	Off	Off	On	-Clock, -Vref

Configuration Number 2: Single Modem Codec (Primary):

Table 1-14. Power State Mapping for Modem Single Codec Desktop Transition

PR<A:D> + MLNK (other power control (PRx) bits do not apply for Intel® ICH6 implementation)						+12	+5 from +12	+3.3 Digital	+3.3 Vaux Digital	Comments
	Sdata_In	DAC1	ADC1	Vref	GPIO					
Device State	MLNK	D	C	B	A					
D0	0	0	0	0	0	On	On	On	On	All on
D1	0	***1***	***1***	0	0	On	On	On	On	-DAC, -ADC
D2	0	1	1	0	0	On	On	On	On	Same as D1
D3 (wake-up on ring)	***1***	1	1	0	0	Off	Off	Off	On	-Sdata_In,
D3	***1***	1	1	***1***	***1***	Off	Off	Off	On	-Sdata_In, -Vref, -GPIO

Table 1-15 and Table 1-16 represent the recommended Power Transition Tables for a Desktop System. These tables preclude the need for a driver to provide codec topology detection simplifying the initialization sequence. These tables do not provide the maximum power saving. However, they are believed to provide sufficient power saving for the Desktop applications. The OEM and IHV are free to provide further differentiation by allowing the deeper power savings obtained by identifying the codec Topology

Table 1-15. Power State Mapping for Audio in Dual Codec Desktop Transition

PR<0:5> + (EAPD)								+12	+5 from +12	+3.3 Digital	+3.3 Vaux Digital	Comments
	EAPD	CLK	AC-Link	Mixer Vref.	Mixer	DAC	ADC					
Device State	7	5	4	3	2	1	0					
D0	0	0	0	0	0	0	0	On	On	On	On	All on
D1	0	0	0	0	0	***1***	***1***	On	On	On	On	-DAC, -ADC
D2	***1***	0	0	0	***1***	1	1	On	On	On	On	-Mix, -Amp
D3	1	0	0	***1***	1	1	1	Off	Off	Off	On	-Clock, -Vref

NOTES:

1. PR(4) link down and PR(5) internal clocks disable are NOT recommended for desktop configuration. Setting these to power control bits could affect modem operation in an AC + MC configuration.
2. In a mobile system configuration, PR(4) and PR(5) could be used to provide further power savings. Driver designers should use D3 state codec semaphores in the ICH6 AC '97 controller to determine audio or modem codecs power status before setting PR(4) and PR(5) bits. Please refer to *Intel® I/O Controller Hub 6 (ICH6) Family Datasheet* for details. The miniport driver developed for the ICH6 AC '97 controller does not provide this capability.
3. In a dual audio codec transition, PR bits must be set in both of the Codec registers. The primary Audio Power Management register set is always located *MMBAR + 26h*, the secondary audio codec power management register set is located at *MMBAR + A6h*. Configuration software must sequence power down to the secondary codec first and then the primary codec. The process is reversed at resume when the primary codec is first restored and then the secondary.

Table 1-16. Power State Mapping for Modem in Dual Codec Desktop Transition

PR<A:D> + MLNK (other power control (PRx) bits do not apply for Intel® ICH6 implementation)						+12	+5 from +12	+3.3 Digital	+3.3 Vaux Digital	Comments
	Sdata_In	DAC1	ADC1	Vref	GPIO					
Device State	MLNK	D	C	B	A					
D0	0	0	0	0	0	On	On	On	On	All on
D1	0	***1***	***1***	0	0	On	On	On	On	-DAC, -ADC
D2	0	1	1	0	0	On	On	On	On	Same as D1
D3 (wake-up on ring)	***1***	1	1	0	0	Off	Off	Off	On	-Sdata_In,
D3	***1***	1	1	***1***	***1***	Off	Off	Off	On	-Sdata_In,-Vref, -GPIO

1.2.6.2 Power Management Topology Considerations

A set of drivers could always assume Configuration Numbers 3 and 4, above, and establish their power management policy based on tables 2-15 and 2-14. These are the safest configurations with a semi-aggressive power management style consistent with a desktop environment. However, even in a desktop environment, further power savings are possible when in Single Codec Configurations Numbers 1 and 2. For the tables above to be implemented, the audio driver needs to be able to determine the AC-link topology configuration.

1.2.6.2.1 Determining the Presence of a Secondary Codec

To determine that a secondary codec is present, the driver needs to check both of the Codec Ready bits located in GSR at:

Secondary Codec Ready:I/O Address: MBBAR + 30h (MBAR +40h) bits 8,9

If both of these bits are set to 1, it indicates that a secondary codec is active in the AC-link.

1.2.6.2.2 Determining the Presence of a Modem Function

In the case of an AMC configuration, only the primary codec ready bit will be indicated. In order to determine proper power down configuration, the audio driver needs to determine the presence of modem functionality in the codec. The audio driver could check the Extended Modem ID register at:

Extended Modem ID:I/O Address: MMBAR + 3Ch

The content of this register will be FFh if no modem function is present.

1.2.6.3 Resume Context Recovery

When the system is placed in a power-down state (S3 or greater) power is removed from the ICH6 AC '97 controller. In this state, DMA registers lose information regarding current position and pointer values. The device driver must be able to save the context of all DMA engines before acknowledging a D3 state. Device drivers must assume that D3 state request precedes a total system power lost therefore context for DMA engines. Upon system power resume, the device driver must restore the DMA Engine register accordingly to saved values prior the suspend event.

1.2.6.4 Aggressive Power Management

As indicated in previous sections it is possible to go into a more aggressive power savings mode by carefully synchronizing audio and modem driver interactions over the AC-link. This aggressive power savings is usually found in mobile environments where battery power is critical.

Driver synchronization is required in a dual codec configuration where the audio driver could cause a link down power condition by setting the PR4 and PR5 bits in the Audio Codec register. When PR4 and PR5 are set, the AC-link base clock BIT_CLK is stopped. If this action occurs while the modem codec is still in operating mode, it will cause malfunctions and possibly a system hang.

To avoid this and similar situations, the audio and modem driver could follow a protocol using the provided audio and modem D3 state bit semaphores, AD3 for audio and MD3 for modem. These bits are located at:

Codec Write Semaphore Registers:

NABMAR +30h Audio I/O Space and MBAR + 40h Modem I/O Space

bit 16 for Audio (AD3)

bit 17 for Modem (MD3)

ICH6 AC '97 drivers should set the appropriate bit after setting the codec in a D3 state. The audio codec could use this semaphore to determine if the modem codec is already in a D3 state and shut down the link by also asserting PR4 and PR5 in the Power Management register in the audio function/codec. The following sections review in detail the sequence of events for drivers/codec entering a D3 state and a resume to D0.

1.2.6.4.1 Primary Audio Requested to D3

The audio power management procedure attempting to get the audio codec to D3 state.

```
If MD3 == true    // (sleeping?)
    {
        Audio_Power_Manage_Reg = D3 + PR4 + PR5;
        // yes, sleep plus AC Link down
    }
Else
    {
        Audio_Power_Manage_Reg = D3;   // No, sleep keeping link up
    }
AD3 = true;          // set to "audio sleeping"
// setting the flag last avoids race condition during D0->D3 transit.
```

1.2.6.4.2 Secondary Modem Requested to D3

The modem power management procedure will try to get the modem codec to D3 state.

```
Secondary_codec = D3 + MLNK  // yes, sleep plus SDATA_IN1 low
MD3 = true
// setting the flag last avoids race condition during D0->D3 transit.
// MLNK corresponds to register Reg. 56h bit 12 (D12)
```

1.2.6.4.3 Secondary Modem Requested to D0

The modem power management procedure will try to get the modem codec to D0 state.

```
MD3 = false              // set to "modem awake"
//Setting the flag first avoid race condition during D3->D0 transit
If Modem_ready == True
    {
        Modem_Power_Manage_Reg = D0  // Bring back to fully awake,
    }
If AD3 == true         // (audio sleeping?)
    {
        Link_reset()     // cause a warm or cold reset
        While (!Modem_ready) // wait for modem ready
            {
                read modem codec ready bit every 400 ms
            }
        Modem_Power_Manage_Reg = D0  // Bring back to awake,
    }
```

1.2.6.4.4 Audio Primary Requested to D0

The audio power management procedure will attempt to get the audio codec to D0 state.

```
AD3 = false              // set to "audio awake"
//Setting the flag first avoid race condition during D3->D0 transit
    If Audio_ready == True
        {
            Audio_Power_Manage_Reg = D0;
            // Bring back to fully awake,
        }
    If MD3 == true;         // (modem sleeping?)
        {
            Link_reset();     // cause a warm or cold reset
            While (!Audio_ready); // wait for modem ready
                {
                    read audio codec ready bit every 100ms;
                }
            Audio_Power_Manage_Reg = D0;  // Bring back to awake,
        }
```

Appendix C provides a schematic representation of the wake-up circuit. This should be used as reference for ACPI and APM wake up code as it relates with the paragraph above.

1.2.6.4.5 Using a Cold or Warm Reset

In the pseudo code above there are several references to resetting the AC-link using “Link_reset()”. Drivers need to differentiate if the system enters a suspend event where core power is removed from the system before deciding to execute a Cold or Warm reset. A device is in a “D3 hot” state after the device is set in the lowest power consumption mode and core power is maintained. A device is in a “D3 cold” state when the device is set in the lowest power consumption mode and core power is removed.

In the ICH6 AC '97 implementation when core power is removed the cold reset bit is reset “0” This bit is located at:

MBBAR + 2Ch and MBAR + 3Ch

bit 1 **ICH6 AC '97 Cold Reset#.**

A driver requested to resume to a D0 state from a D3 state must check the status of the ICH6 AC '97 cold reset bit. If this bit has a value of 0, the driver will set it to 1 to deassert the AC_RESET# signal in the link, thus completing a cold reset. If the cold reset bit is set to 1 then a warm rest is required if the AC-link is down by the procedures indicated under aggressive power management. To execute an ICH6 AC '97 warm reset the driver must set to 1 the ICH6 AC '97 warm reset bit located at:

MBBAR + 2Ch and MBAR + 3Ch

bit 2 **ICH6 AC '97 Warm Reset#.**

A pseudo code representation is as follows:

```
void Link_reset(void)
    {
If Cold_Reset# == True// AC_RESET# asserted, D3 when cold!
        {
Cold_Reset# = False;  // De-assert AC_RESET# Wake-up!
        }
    Else
        {
        Warm_reset = True;    // D3 is Hot! Do warm reset
        }
    }
```

1.3 Surround Audio Support

The *AC '97 Component Specification* allows for up to six channels of audio supported in the AC-link. The audio device driver must determine the number of audio channels available in the codec(s) and properly enable the ICH6 AC '97 controller to support those channels in the link.

1.3.1 Determine Codec's Audio Channels

Upon determination of the link topology, system software must proceed to resolve the total number of audio channels available in the codec(s). Appendix A of the *AC '97 Component Specification,* Revision 2.1, defines Codec register index 28h to indicate the presence of surround, center and LFE channels as follows: *Extended Audio ID Register (Index 28h).*

Table 1-17. Audio Codec Extended Audio ID Register

Bit D6:	CDAC=1 indicates optional PCM Center DAC is supported
Bit D7:	SDAC=1 indicates optional PCM Surround DAC is supported
Bit D8:	LDAC=1 indicates optional PCM LFE DAC is supported
Bit D9:	AMAP=1 indicates optional slot/DAC mappings based on Codec ID (Refer to *AC '97 Component Specification, Revision 2.1*, Appendix D for description)
Bit D15-D14:	ID1, ID0 is a 2-bit field which indicates the Codec configuration: Primary is 00; Secondary is 01, 10, or 11 (Refer to *AC '97 Component Specification, Revision 2.1* Appendix C for details)

The *AC '97 Component Specification,* Revision 2.2, Section 5.8.2, provides detailed information on the usage model for multiple audio channels. This specification assumes the proper use of the AMAP bit describes the partition of the channels exposing critical functionality. If AMAP bit is not set to 1, a generic driver will not be able to determine the proper codec to slot distribution for a split audio codec configuration.

Based on Table 29 of the *AC '97 Component Specification*, Revision 2.2, the following table describes the available codec topology to audio channel distribution for ICH6 AC '97 controller.

Table 1-18. Single Codec Audio Channel Distribution

Single Audio Codec Configuration	
Primary	**Total Audio Channels**
L/R	2
L/R; S-L/R	4
L/R; S-L/R; C/LFE	6

Table 1-19. Multiple Codec Audio Channel Distribution

Split Audio Codec Configuration			
Primary	**Secondary**	**Tertiary**	**Total Channels**
L/R	-	-	2
L/R	S-L/R	-	4
L/R	S-L/R; C/LFE	-	6
L/R	S-L/R	C/LFE	6

Legend:

L/R: Left Stereo Channel (Slot 3); Right Stereo Channel (Slot 4)

S-L/R: Surround Left Channel (Slot 7); Surround Right Channel (Slot 8)

C/LFE: Center Channel (Slot 6); Low Frequency Enhancement "subwoofer" (Slot 9)

(-): Not Applicable for this configuration could be used for simultaneous S/PIDF output

1.3.2 Enabling Intel® ICH6 AC '97 Controller Audio Channels

The ICH6 indicates how many channels supports in its GSR. The ICH6 AC '97 controller defaults to PCM Stereo after system reset or suspend-resume from S3, S4 or S5. Audio drivers can determine the total number of channels and re-configure the controller to support either 4- or 6-channel audio for PCM Out only. PCM In is not configurable and always is set for 2-channels input. PCM MIC In is also not configurable and is always single channel (monaural.) The total numbers of audio channels supported are indicated at: *MBBAR + 30h: Global Status Registers.*

Table 1-20. CM 4/6 -PCM Channels Capability Bits

Bit	Description
21:20	**PCM 4/6 Capability:** These read-only bits indicate the capability to support more than 2 channels for the PCM OUT. These bits will both be 1 to indicate that the Intel® ICH6 supports both 4 and 6-channel PCM output.

The ICH6 AC '97 controller also provides support for an independent DMA channel to provide simultaneous S/PDIF output stream. This allows for concurrent PCM surround out with AC3 compress over S/PDIF interface. ICH6 AC '97 controller follows slot assignments described in Section 5.4.2.1 and Table 13 of the *AC '97 Component Specification*, Revision 2.2.

System software enables the number of channels it intends to provide to the codec at: *MBBAR + 2Ch: Global Control Registers.*

Table 1-21. AC-Link PCM 4/6 -Channels Enable Bits

21:20	PCM 4/6 Enable: Enables the PCM Output to be in 4 channel or 6-channel mode	PCM Slots Enable: (PCM OUT DMA)	S/PDIF Slots Enable: (S/PDIF Out DMA)
	00 = 2 channel mode (default).	3, 4 (L and R)	7, 8
	01 = 4 channel mode.	3, 4, 7, 8 (L, R, RL and RR)	6, 9
	10 = 6 channel mode	3, 4, 7, 8, 6, 9 (L, R, RL, RR, C and LFE)	10, 11
	11 = Reserved	(undefined)	(undefined)

1.4 20-Bit PCM Support

The ICH6 AC '97 controller provides support for 16- or 20-bit PCM out. Software can determine if 20-bit samples are supported in the controller by reading the sample capabilities bits in GLOB_STA registers as follows: *MBBAR + 30h: Global Status Registers.*

Table 1-22. Sample Capabilities

Bit	Description
23:22	**Sample Capabilities:** Indicates the capability to support any greater than 16-bit audio. 00 = 16-bit Audio only supported (ICH1, ICH2, and ICH3 value) 01 = 16- and 20-bit Audio supported (ICH6 value) 10 = Reserved (for 24-bit audio support) 11 = Reserved

After determination of the controller capabilities, software can enable 20-bits formats by programming the GLOB_CNT register as follows: *MBBAR + 2Ch: Global Control Registers.*

Table 1-23. PCM Out Mode Selector

Bit	Description
23:22	**PCM Out Mode (POM):** Enables the PCM out channel to use 16 or 20 bits. This does not affect the microphone, PCM In or S/PDIF DMA. When greater than 16-bits audio is used, the data structures are aligned as 32 bits per sample, with the **upper** order bits representing the data, and the **lower** order bits as don't care. 00 = 16 bit audio (default) 01 = 20 bit audio 10 = 24-bit audio (not supported. If set, indeterminate behavior will result. 11 = Reserved. If set, indeterminate behavior will result.

Note: Software must ensure that the PCM out DMA engine is stop and a new descriptor initialized before changing the value of PCM Out mode. These bits **must not** be modified when the PCM Out engine is running.

1.5 Independent S-P/DIF Output Capability

The ICH6 AC '97 controller provides an independent DMA engine for S-P/DIF output which operates independently of the six channel PCM Out stream. This allows the S-P/DIF data stream to be independent of the PCM stream, allowing usage models such as playing a DVD movie with output to S-P/DIF while at the same time using PCM In and Out in a telephony application.

The S-P/DIF DMA engine is initialized and is programmed in the same manner as the other DMA engines available, as described in Section 1.2.2.

Data from the S-P/DIF DMA engine may be output on several different pairs of slots, depending on the codec configuration. The SSM bits in the Global Control register control on which slots the S/P-DIF data is transmitted, as defined below. The default value (00) for this register is a reserved value, and so these bit **must** be appropriately programmed before the DMA engine is used. These bits are reset on controller reset, and are not affected by the reset of the S-P/DIF DMA engine.

Table 1-24. Global Control Register S-P/DIF Slot Map Bits

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>31:30</td><td>RW</td><td>00</td><td>S/PDIF Slot Map (SSM): If the run/pause bus master bit (bit 0 of offset 2Bh) is set, then the value in these bits indicate which slots S/PDIF data is transmitted on. Software must ensure that the programming here does not conflict with the PCM channels being used. If there is a conflict, unpredictable behavior will result – the hardware will not check.
<table>
<tr><th>Bits</th><th>Output Slots for S/PDIF Data</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>7 & 8</td></tr>
<tr><td>10</td><td>6 & 9</td></tr>
<tr><td>11</td><td>10 & 11</td></tr>
</table></td></tr>
</table>

1.6 Support for Double Rate Audio

The ICH6 AC '97 controller has the capability of supporting a stereo 96 kHz stream using the AC '97 Double Rate Audio (DRA) support. This capability is enabled by programming the controller to use four channel mode, which will place data on slots 3, 4, 7, and 8. A codec which is capable of treating the stream as PCM Left, Right, Left+1, and Right+1 data will interpret this as 96 kHz stream, providing higher quality audio.

Note that the current *AC '97 Component Specification*, Revision 2.3 http://www.intel.com/ial/scalableplatforms/audio/index.htm, provides support provides the Double Rate date on slots 3, 4, 10, and 11, a mode which is not supported by the ICH6 controller. Codecs will need to additionally accept data on slots 3, 4, 7, and 8 to utilize DRA with the ICH6.

1.7 Independent Input Channels Capability

ICH6 AC '97 controller provides capability for two new DMA channels dedicated to independent PCM and Microphone audio streams. These allow improved features that enable applications such as audio mobile docking and microphone arrays. Before the DMA transfer can occur, software must determine the proper codec topology in the AC-link in order to program proper volume values and other relevant properties.

1.7.1 Link Topology Determination

While the SDATA_Out lines are shared by all *AC '97 Component Specification*, Revision 2.2 compliant codecs, the input DMA engines must be properly steered to specific SDATA_IN[0:2] lines on the link. Software must determine the specific codec ID assignment to the SDATA_IN[0:2] lines to properly adjust codec properties for operation. Software must fill a table similar to the following:

Table 1-25. Topology Descriptor

Input Channels	SDATA_IN[0:2]	Codec ID[00:10]
PCM IN & MIC	TBD	TBD
PCM IN 2 & MIC 2	TBD	TBD

Software uses a recurrent algorithm using the SDATA_IN Map register to determine the values for the table above.

MBBAR + 80h: SDATA_IN Map Register.

Table 1-26. SDATA_IN Map (Sheet 1 of 2)

Bit	Description
7:6	PCM In 2, Microphone In 2 Data In Line (D21L): When the SE bit is set, these bits indicates which SDATA_IN line should be used by the hardware for decoding the input slots for PCM In 2 and Microphone In 2. When the SE bit is cleared, the value of these bits are irrelevant, and PCM In 2 and Mic In 2 DMA engines are not available. 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved
5:4	PCM In 1, Microphone In 1 Data In Line (DI1L): When the SE bit is set, these bits indicates which SDATA_IN line should be used by the hardware for decoding the input slots for PCM In 1 and Microphone In 1. When the SE bit is cleared, the value of these bits are irrelevant, and the PCM In 1 and Mic In 1 engines use the OR'd SDATA_IN lines. 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved

Table 1-26. SDATA_IN Map (Sheet 2 of 2)

3	Steer Enable (SE): When set, the SDATA_IN lines are treated separately and not OR'd together before being sent to the DMA engines. When cleared, the SDATA_IN lines are OR'd together, and the "Microphone In 2" and "PCM In 2" DMA engines are not available.
2	Reserved
1:0	Last Codec Read Data Input (LDI): When a Codec register is read, this indicates which SDATA_IN the read data returned on. Software can use this to determine how the codecs are mapped. The values are: 00 SDATA_IN0 01 SDATA_IN1 10 SDATA_IN2 11 Reserved

Software must follow the following steps to determine the codec topology:

1. Determine codec present by reading the codec ready bits GLOB_STA Global Status register.

MBBAR + 30h: SDATA_IN Map Register.

Table 1-27. Codec Ready Bits

Bit	Description
28	Tertiary Codec Ready (TRI) – RO. Provides the state of codec ready bit on SDATA_IN[2]
9	Secondary Codec Ready (TRI) – RO. Provides the state of codec ready bit on SDATA_IN[1]
8	Primary Codec Ready (PRI) – RO. Provides the state of codec ready bit on SDATA_IN[0]

2. Once codec present is determined, software will determine the ID of the codec attached to each SDATA_IN line in used by reading any Codec register at codec ID 00, 01 or 10, as indicated in Table 1-28, and determine the returning SDATA_IN line by reading bits 1:0 in SDATA_IN Map registers as shown in Table 1-26, above.

Table 1-28. MMBAR: Mixer Base Address Register

Codec ID 00 (offset)	Codec ID 00 (Offset)	Codec ID 00 (Offset)
00h to 7Fh	80h to FFh	100-7Fh

3. Base on the codec assignment and the specific application software will map the specific DMA engines to the appropriate SDATA_IN[0:2] lines by programming [7:6] for PCM/MIC and [5:4] for PCM/MIC2 in SDATA_IN MAP register.
4. Finally software will enable the steering mechanism by asserting bit [3] in SDATA_IN Map register before initiating a data transfer.

1.8 Intel® ICH6 AC '97 Modem Driver

The AC '97 Component Specification, Revision 2.3 allows for a modem codec to be connected to the AC-link interface. This allows for the development of a software stack that provides modem functionality, i.e. a soft modem. Currently there is no a single definition of how a soft-modem should be implemented under Microsoft* operating systems as is the case for audio in a WDM environment. Soft modem vendors have developed a variety of implementations for Windows 95/98/2K/XP, Windows NT 4.0 and Windows NT 5.0 operating systems. The design problems are not trivial for the soft-modem developer. This document does not attempt to describe solutions for each of these environments, but focuses instead on facilitating the development of the driver/hardware interface. At the time of this specification, Microsoft and Intel are engaging with the industry to define a common interface for the WDM environment.

1.8.1 Robust Host Based Generation of a Synchronous Data Stream

This section presents a method for synchronous modem data to be reliably generated on the host processor of a computer system where the host processor is running a non real-time operating system whose maximum response latency (interrupt, thread, etc.) exceeds the period at which the host processor generates consecutive buffers of modem data. For the purposes of this discussion we will assume that the host processor periodically, in response to interrupts, generates a buffer of modem data in memory which is then utilized or consumed in a synchronous fashion by hardware. This modem data would comprise a sequence of digital representations of the analog signal to be transmitted over a phone line in accordance with one of a variety of modem protocols, baud rates, etc., and could be transmitted to the ICH6 AC '97 DMA engines via the buffer descriptor list.as described in Section 1.2.2.1.

We will also assume, for the sake of simplicity, that the data are double buffered, so that failure to generate new data before the next period will result in stream underflow from the hardware's viewpoint, but other scenarios including multiply buffered designs as well as non-periodic processing models can be accommodated. The algorithm works by providing good data followed by spurious data, which is chosen or computed so as to be adequate to maintain connection with the other modem by, for example, transitioning seamlessly with respect to the phase of the carrier frequency and the baud rate, thereby avoiding a retrain. This enables the datapumps of the two modems to maintain synchronization in the face of infrequent hold-offs from processing experienced by the datapump of the host-based transmitting modem. The spurious data will, of course, cause a packet retransmission or other action by the controller, but to the receiving modem the incoming data signal will be indistinguishable from one corrupted by line conditions.

The first invocation of the host based modem task provides an initial buffer and one or more buffers of spurious data (henceforth, spurious buffers). The task chooses or computes each of the spurious buffer(s) based on signal state at end of immediately preceding buffer. Note that these buffers do not have to be computed on the fly but can be precomputed and indexed into at run time. Subsequent invocations overwrite the previously provided spurious data with good data so that under normal conditions the spurious data is never used or consumed by the DMA engine. In the event that the host based modem task does not generate the next buffer in time for the DMA engine to begin consuming it the DMA engine is able to begin consuming the spurious buffer and thereby maintain seamless connection with the other modem's datapump.

1.8.1.1 Spurious Data Algorithm

The following pseudo code presents a conceptual view the algorithm. `LastState()` is a function of which returns a unique integer as a function of, for example, the carrier phase and baud position of the last sample in the buffer. In an actual implementation this value would be computed during the course of generating the buffer. The `SpuriousBufferList` is an array of precomputed spurious buffers.

```
while (1)
{
    compute next buffer;
    pNextBuffer = &buffer;
    pSpuriousBuffer = &(SpuriousBufferList[LastState(buffer)]);
    wait for timer interrupt;
}
```

In this simplified scenario the device grabs the `pNextBuffer` address and stores it locally, using it to request the samples in the buffer one at a time. At the same time the device copies the `pSpuriousBuffer` into `pNextBuffer` so that when it is done with the current buffer it will get the spurious buffer unless the host software runs and overwrites `pNextBuffer` with a pointer to good data. In the next section we show how to implement the spurious data algorithm within the context of the ICH6 AC '97 buffer descriptor interface to hardware.

1.8.1.2 Intel® ICH6 AC '97 Spurious Data Implementation

The following pseudo code presents a modified version of the routine that prepares buffers and inserts them into the ICH6 AC '97 buffer descriptor list. In contrast to the version of this routine given in Section 2.2.3, in this version tail points to the last good (i.e., non-spurious) buffer in the list. Furthermore, because the ICH6 AC '97 DMA engine prefetches the next buffer descriptor we split the buffer generated by the datapump into two parts, with the second as small as practical and denote this size as `MinBufferLength` (here assumed to be eight samples = 4 dwords = 500 µs. at 16 kHz). To simplify, we assume that only a single buffer is generated by the datapump at a time and we ignore checking for the end of the descriptor list (i.e., the addition is implicitly modulo 32).

```
while (tail <= Prefetched_Index)
{
    tail++;  // Happens IFF Spurious Data was used
}
if (((tail <= LastValidIndex) || (tail == free)) &&
    (((tail+1) <= LastValidIndex) || ((tail+1) == free)))
{
    Descriptor.BufferPtr[tail] = &buffer;
    Descriptor.BufferLength[tail] = length(buffer) - MinBufferLength;
    Descriptor.BufferPtr[tail+1] =
        &buffer + length(buffer) - MinBufferLength;
    Descriptor.BufferLength[tail+1] = MinBufferLength;
    tail += 2;
 }
else
{
    ;    //error: no space for this data buffer
}
if ((tail <= LastValid index) || (tail == free))
    Descriptor.BufferPtr[tail] =
```

```
        &(SpuriousBufferList[LastState(buffer)]);
    Descriptor.BufferLength[tail] =
        SpuriousBufferLength[LastState(buffer)];
    LastValidIndex = tail;
    //Note: tail is NOT incremented, so next time we'll overwrite this
    //      descriptor, which is the whole point of this algorithm
}
else
{
    LastValidIndex = tail-1;//warning: no space for spurious data buffer
}
```

The above implementation can be improved upon in a number of ways. First, rather than adding a single (large) spurious buffer, a number of smaller ones could be chained together. In this way, the amount of spurious data actually transmitted would be reduced while still maintaining a given level of protection against long latencies for the host-based software. Additionally, the implementation could be extended to handle multiple buffers at once, inserting several buffers in a row and only splitting the last one and then appending a spurious buffer or buffers. Finally, the descriptor list is a circular buffer and a real implementation would have to check `tail and tail+1 against base_address + 31 * 8.`

§

2 AC ’97 Audio Controller Registers (D30:F2)

2.1 AC ’97 Audio PCI Configuration Space (Audio—D30:F2)

Note: Registers that are not shown should be treated as Reserved.

Table 2-1. AC ‘97 Audio PCI Register Address Map (Audio—D30:F2)

Offset	Mnemonic	Register Name	Default	Access
00–01h	VID	Vendor Identification	8086h	RO
02–03h	DID	Device Identification	266Eh	RO
04–05h	PCICMD	PCI Command	0000	R/W, RO
06–07h	PCISTS	PCI Status	0280h	R/WC, RO
08h	RID	Revision Identification	See register description	RO
09h	PI	Programming Interface	00	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	04h	RO
0Eh	HEADTYP	Header Type	00h	RO
10–13h	NAMBBAR	Native Audio Mixer Base Address	00000001h	R/W, RO
14–17h	NAMMBAR	Native Audio Bus Mastering Base Address	00000001h	R/W, RO
18–1Bh	MMBAR	Mixer Base Address (Mem)	00000000h	R/W, RO
1C–1Fh	MBBAR	Bus Master Base Address (Mem)	00000000h	R/W, RO
2C–2Dh	SVID	Subsystem Vendor Identification	0000h	R/WO
2E–2Fh	SID	Subsystem Identification	0000h	R/WO
34h	CAP_PTR	Capabilities Pointer	50h	RO
3Ch	INT_LN	Interrupt Line	00h	R/W
3Dh	INT_PN	Interrupt Pin	See register description	RO
40h	PCID	Programmable Codec ID	09h	R/W
41h	CFG	Configuration	00h	R/W
50–51h	PID	PCI Power Management Capability ID	0001h	RO
52–53h	PC	PC -Power Management Capabilities	C9C2h	RO
54–55h	PCS	Power Management Control and Status	0000h	R/W, R/WC

Note: Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the following BIOS programmed registers as BIOS may not be invoked following the D3-to-D0 transition. All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Dh – Subsystem Vendor ID (SVID)
- offset 2Eh–2Fh – Subsystem ID (SID)
- offset 40h – Programmable Codec ID (PCID)
- offset 41h – Configuration (CFG)
- Resume well registers **will not** be reset by the $D3_{HOT}$ to D0 transition:
- offset 54h–55h – Power Management Control and Status (PCS)
- Bus Mastering Register: Global Status Register, bits 17:16
- Bus Mastering Register: SDATA_IN MAP register, bits 7:3

2.1.1 VID—Vendor Identification Register (Audio—D30:F2)

Offset:	00–01h	Attribute:	RO
Default Value:	8086h	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	Vendor ID. This is a 16-bit value assigned to Intel.

2.1.2 DID—Device Identification Register (Audio—D30:F2)

Offset:	02–03h	Attribute:	RO
Default Value:	266Eh	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	Device ID.

2.1.3 PCICMD—PCI Command Register (Audio—D30:F2)

Address Offset:	04–05h	Attribute:	R/W, RO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCICMD is a 16-bit control register. Refer to the PCI 2.3 specification for complete details on each bit.

Bit	Description
15:11	Reserved. Read 0.
10	**Interrupt Disable (ID)** — R/W. 0 = The INTx# signals may be asserted and MSIs may be generated. 1 = The AC '97 controller's INTx# signal will be de-asserted and it may not generate MSIs.
9	Fast Back to Back Enable (FBE) — RO. Not implemented. Hardwired to 0.
8	SERR# Enable (SERR_EN) — RO. Not implemented. Hardwired to 0.
7	Wait Cycle Control (WCC) — RO. Not implemented. Hardwired to 0.
6	Parity Error Response (PER) — RO. Not implemented. Hardwired to 0.
5	VGA Palette Snoop (VPS). Not implemented. Hardwired to 0.
4	Memory Write and Invalidate Enable (MWIE) — RO. Not implemented. Hardwired to 0.
3	Special Cycle Enable (SCE). Not implemented. Hardwired to 0.
2	**Bus Master Enable (BME)** — R/W. Controls standard PCI bus mastering capabilities. 0 = Disable 1 = Enable
1	**Memory Space Enable (MSE)** — R/W. Enables memory space addresses to the AC '97 Audio controller. 0 = Disable 1 = Enable
0	**I/O Space Enable (IOSE)** — R/W. This bit controls access to the AC '97 Audio controller I/O space registers. 0 = Disable (Default). 1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit. **NOTE:** This bit becomes writable when the IOSE bit in offset 41h is set. If at any point software decides to clear the IOSE bit, software must first clear the IOS bit.

2.1.4 PCISTS—PCI Status Register (Audio—D30:F2)

Offset:	06–07h	Attribute:	RO, R/WC
Default Value	0280h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCISTA is a 16-bit status register. Refer to the PCI 2.3 specification for complete details on each bit.

Bit	Description
15	Detected Parity Error (DPE). Not implemented. Hardwired to 0.
14	Signaled System Error (SSE) — RO. Not implemented. Hardwired to 0.
13	**Master Abort Status (MAS)** — R/WC. Software clears this bit by writing a 1 to it. 0 = No master abort generated. 1 = Bus Master AC '97 2.3 interface function, as a master, generates a master abort.
12	Reserved — RO. Will always read as 0.
11	Signaled Target Abort (STA) — RO. Not implemented. Hardwired to 0.
10:9	DEVSEL# Timing Status (DEV_STS) — RO. This 2-bit field reflects the ICH6's DEVSEL# timing when performing a positive decode. 01b = Medium timing.
8	Data Parity Error Detected (DPED) — RO. Not implemented. Hardwired to 0.
7	Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1. This bit indicates that the ICH6 as a target is capable of fast back-to-back transactions.
6	UDF Supported — RO. Not implemented. Hardwired to 0.
5	66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0.
4	Capabilities List (CAP_LIST) — RO. Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h.
3	**Interrupt Status (IS)** — RO. 0 = This bit is 0 after the interrupt is cleared. 1 = This bit is 1 when the INTx# is asserted.
2:0	Reserved.

2.1.5 RID—Revision Identification Register (Audio—D30:F2)

Offset:	08h	Attribute:	RO
Default Value:	See bit description	Size:	8 Bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Revision ID — RO. Refer to the *Intel® ICH6/ICH6R/ICH6-M Specification Update* for the value of the Revision ID Register

2.1.6 PI—Programming Interface Register (Audio—D30:F2)

Offset:	09h	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Programming Interface — RO.

2.1.7 SCC—Sub Class Code Register (Audio—D30:F2)

Address Offset:	0Ah	Attribute:	RO
Default Value:	01h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Sub Class Code (SCC) — RO. 01h = Audio Device

2.1.8 BCC—Base Class Code Register (Audio—D30:F2)

Address Offset:	0Bh	Attribute:	RO
Default Value:	04h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Base Class Code (BCC) — RO. 04h = Multimedia device

2.1.9 HEADTYP—Header Type Register (Audio—D30:F2)

Address Offset:	0Eh	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Header Type — RO. Hardwired to 00h.

2.1.10 NAMBAR—Native Audio Mixer Base Address Register (Audio—D30:F2)

Address Offset:	10–13h	Attribute:	R/W, RO
Default Value:	00000001h	Size:	32 bits
Lockable:	No	Power Well:	Core

The Native PCI Mode Audio function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Native Audio Mixer software interface. The mixer requires 256 bytes of I/O space. Native Audio Mixer and Modem codec I/O registers are located from 00h to 7Fh and reside in the codec. Access to these registers will be decoded by the AC '97 controller and forwarded over the AC-link to the codec. The codec will then respond with the register value.

In the case of the split codec implementation, accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec.

Note: The tertiary codec cannot be addressed via this address space. The tertiary space is only available from the new MMBAR register. This register powers up as read only and only becomes write-able when the IOSE bit in offset 41h is set.

For description of these I/O registers, refer to the *Audio Codec '97 Component Specification, Version 2.3*.

Bit	Description
31:16	Hardwired to 0's.
15:8	**Base Address** — R/W. These bits are used in the I/O space decode of the Native Audio Mixer interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 mixer, the upper 16 bits are hardwired to 0, while bits 15:8 are programmable. This configuration yields a maximum I/O block size of 256 bytes for this base address.
7:1	Reserved. Read as 0's.
0	**Resource Type Indicator (RTE)** — RO. This bit defaults to 0 and changes to 1 if the IOSE bit is set (D30:F2:Offset 41h, bit 0). When 1, this bit indicates a request for I/O space.

2.1.11 NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D30:F2)

Address Offset:	14–17h	Attribute:	R/W, RO
Default Value:	00000001h	Size:	32 bits
Lockable:	No	Power Well:	Core

The Native PCI Mode Audio function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Native Mode Audio software interface.

Note: The DMA registers for S/PDIF* and Microphone In 2 cannot be addressed via this address space. These DMA functions are only available from the new MBBAR register. This register powers up as read only and only becomes write-able when the IOSE bit in offset 41h is set.

Bit	Description
31:16	Hardwired to 0's
15:6	**Base Address** — R/W. These bits are used in the I/O space decode of the Native Audio Bus Mastering interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For AC '97 bus mastering, the upper 16 bits are hardwired to 0, while bits 15:6 are programmable. This configuration yields a maximum I/O block size of 64 bytes for this base address.
5:1	Reserved. Read as 0's.
0	**Resource Type Indicator (RTE)** — RO. This bit defaults to 0 and changes to 1 if the IOSE bit is set (D30:F2:Offset 41h, bit 0). When 1, this bit indicates a request for I/O space.

2.1.12 MMBAR—Mixer Base Address Register (Audio—D30:F2)

Address Offset:	18–1Bh	Attribute:	R/W, RO
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

This BAR creates 512 bytes of memory space to signify the base address of the register space. The lower 256 bytes of this space map to the same registers as the 256-byte I/O space pointed to by NAMBAR. The lower 384 bytes are divided as follows:

- 128 bytes for the primary codec (offsets 00–7Fh)
- 128 bytes for the secondary codec (offsets 80–FFh)
- 128 bytes for the tertiary codec (offsets 100h–17Fh).
- 128 bytes of reserved space (offsets 180h–1FFh), returning all 0.

Bit	Description
31:9	**Base Address** — R/W. This field provides the lower 32-bits of the 512-byte memory offset to use for decoding the primary, secondary, and tertiary codec's mixer spaces.
8:3	Reserved. Read as 0's.
2:1	Type — RO. Hardwired to 00b to Indicate the base address exists in 32-bit address space
0	Resource Type Indicator (RTE) — RO. Hardwired to 0 to indicate a request for memory space.

2.1.13 MBBAR—Bus Master Base Address Register (Audio—D30:F2)

Address Offset:	1C–1Fh	Attribute:	R/W, RO
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

This BAR creates 256-bytes of memory space to signify the base address of the bus master memory space. The lower 64-bytes of the space pointed to by this register point to the same registers as the MBBAR.

Bit	Description
31:8	**Base Address** — R/W. This field provides the I/O offset to use for decoding the PCM In, PCM Out, and Microphone 1 DMA engines.
7:3	Reserved. Read as 0's.
2:1	Type — RO. Hardwired to 00b to indicate the base address exists in 32-bit address space
0	Resource Type Indicator (RTE) — RO. Hardwired to 0 to indicate a request for memory space.

2.1.14 SVID—Subsystem Vendor Identification Register (Audio—D30:F2)

Address Offset:	2C–2Dh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

The SVID register, in combination with the Subsystem ID register (D30:F2:2Eh), enable the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	**Subsystem Vendor ID** — R/WO.

2.1.15 SID—Subsystem Identification Register (Audio—D30:F2)

Address Offset:	2E–2Fh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

The SID register, in combination with the Subsystem Vendor ID register (D30:F2:2Ch) make it possible for the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	**Subsystem ID** — R/WO.

2.1.16 CAP_PTR—Capabilities Pointer Register (Audio—D30:F2)

Address Offset:	34h	Attribute:	RO
Default Value:	50h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates the offset for the capability pointer.

Bit	Description
7:0	Capabilities Pointer (CAP_PTR) — RO. This field indicates that the first capability pointer offset is offset 50h

2.1.17 INT_LN—Interrupt Line Register (Audio—D30:F2)

Address Offset:	3Ch	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates which PCI interrupt line is used for the AC '97 module interrupt.

Bit	Description
7:0	**Interrupt Line (INT_LN)** — R/W. This data is not used by the ICH6. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

2.1.18 INT_PN—Interrupt Pin Register (Audio—D30:F2)

Address Offset:	3Dh	Attribute:	RO
Default Value:	See Description	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates which PCI interrupt pin is used for the AC '97 module interrupt. The AC '97 interrupt is internally OR’d to the interrupt controller with the PIRQB# signal.

Bit	Description
7:0	AC '97 Interrupt Routing — RO. This reflects the value of D30IP.AAIP in chipset configuration space.

2.1.19 PCID—Programmable Codec Identification Register (Audio—D30:F2)

Address Offset:	40h	Attribute:	R/W
Default Value:	09h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register is used to specify the ID for the secondary and tertiary codecs for I/O accesses. This register is not affected by the $D3_{HOT}$ to D0 transition. The value in this register must be modified only before any AC ’97 codec accesses.

Bit	Description
7:4	Reserved.
3:2	**Tertiary Codec ID (TID**) — R/W. These bits define the encoded ID that is used to address the tertiary codec I/O space. Bit 1 is the first bit sent and Bit 0 is the second bit sent on ACZ_SDOUT during slot 0.
1:0	**Secondary Codec ID (SCID)** — R/W. These two bits define the encoded ID that is used to address the secondary codec I/O space. The two bits are the ID that will be placed on slot 0, bits 0 and 1, upon an I/O access to the secondary codec. Bit 1 is the first bit sent and bit 0 is the second bit sent on ACZ_SDOUT during slot 0.

2.1.20 CFG—Configuration Register (Audio—D30:F2)

Address Offset:	41h	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register is used to specify the ID for the secondary and tertiary codecs for I/O accesses. This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
7:1	Reserved—RO.
0	**I/O Space Enable (IOSE)** — R/W. 0 = Disable. The IOS bit at offset 04h and the I/O space BARs at offset 10h and 14h become read only registers. Additionally, bit 0 of the I/O BARs at offsets 10h and 14h are hardwired to 0 when this bit is 0. This is the default state for the I/O BARs. BIOS must explicitly set this bit to allow a legacy driver to work. 1 = Enable.

2.1.21 PID—PCI Power Management Capability Identification Register (Audio—D30:F2)

Address Offset:	50–51h	Attribute:	RO
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:8	Next Capability (NEXT) — RO. This field indicates that the next item in the list is at offset 00h.
7:0	Capability ID (CAP) — RO.This field indicates that this pointer is a message signaled interrupt capability

2.1.22 PC—Power Management Capabilities Register (Audio—D30:F2)

Address Offset:	52–53h	Attribute:	RO
Default Value:	C9C2h	Size:	16 bits
Lockable:	No	Power Well:	Core

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:11	PME Support — RO. This field indicates PME# can be generated from all D states.
10:9	Reserved.
8:6	Auxiliary Current — RO. This field reports 375 mA maximum Suspend well current required when in the $D3_{COLD}$ state.
5	Device Specific Initialization (DSI)—RO. This field indicates that no device-specific initialization is required.
4	Reserved — RO.
3	PME Clock (PMEC) — RO. This field indicates that PCI clock is not required to generate PME#.
2:0	Version (VER) — RO. This field indicates support for Revision 1.1 of the PCI Power Management Specification.

2.1.23 PCS—Power Management Control and Status Register (Audio—D30:F2)

Address Offset:	54–55h	Attribute:	R/W, R/WC
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Resume

Bit	Description
15	**PME Status (PMES)** — R/WC. This bit resides in the resume well. Software clears this bit by writing a 1 to it. 0 = PME# signal Not asserted by AC ‘97 controller. 1 = This bit is set when the AC ’97 controller would normally assert the PME# signal independent of the state of the PME_En bit.
14:9	Reserved — RO.
8	**Power Management Event Enable (PMEE)** — R/W. 0 = Disable. 1 = Enable. When set, and if corresponding PMES is also set, the AC '97 controller sets the AC97_STS bit in the GPE0_STS register
7:2	Reserved—RO.
1:0	**Power State (PS)** — R/W. This field is used both to determine the current power state of the AC ’97 controller and to set a new power state. The values are: 00 = D0 state 01 = not supported 10 = not supported 11 = $D3_{HOT}$ state When in the $D3_{HOT}$ state, the AC ’97 controller’s configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs.

2.2 AC ’97 Audio I/O Space (D30:F2)

The AC ’97 I/O space includes Native Audio Bus Master Registers and Native Mixer Registers. For the ICH6, the offsets are important as they will determine bits 1:0 of the TAG field (codec ID).

Audio Mixer I/O space can be accessed as a 16-bit field only since the data packet length on AC-link is a word. Any S/W access to the codec will be done as a 16-bit access starting from the first active byte. In case no byte enables are active, the access will be done at the first word of the qWord that contains the address of this request.

Table 2-2. Intel® ICH6 Audio Mixer Register Configuration (Sheet 1 of 2)

Primary Offset (Codec ID =00)	Secondary Offset (Codec ID =01)	Tertiary Offset (Codec ID =10)	NAMBAR Exposed Registers (D30:F2)
00h	80h	100h	Reset
02h	82h	102h	Master Volume
04h	84h	104h	Aux Out Volume
06h	86h	106h	Mono Volume
08h	88h	108h	Master Tone (R & L)
0Ah	8Ah	10Ah	PC_BEEP Volume
0Ch	8Ch	10Ch	Phone Volume
0Eh	8Eh	10Eh	Mic Volume
10h	90h	110h	Line In Volume
12h	92h	112h	CD Volume
14h	94h	114h	Video Volume
16h	96h	116h	Aux In Volume
18h	98h	118h	PCM Out Volume
1Ah	9Ah	11Ah	Record Select
1Ch	9Ch	11Ch	Record Gain
1Eh	9Eh	11Eh	Record Gain Mic
20h	A0h	120h	General Purpose
22h	A2h	122h	3D Control
24h	A4h	124h	AC ’97 RESERVED
26h	A6h	126h	Powerdown Ctrl/Stat
28h	A8h	128h	Extended Audio
2Ah	AAh	12Ah	Extended Audio Ctrl/Stat
2Ch	ACh	12Ch	PCM Front DAC Rate
2Eh	AEh	12Eh	PCM Surround DAC Rate
30h	B0h	130h	PCM LFE DAC Rate
32h	B2h	132h	PCM LR ADC Rate
34h	B4h	134h	MIC ADC Rate
36h	B6h	136h	6Ch Vol: C, LFE
38h	B8h	138h	6Ch Vol: L, R Surround
3Ah	BAh	13Ah	S/PDIF Control
3C–56h	BC–D6h	13C–156h	Intel RESERVED
58h	D8h	158h	AC ’97 Reserved

Table 2-2. Intel® ICH6 Audio Mixer Register Configuration (Sheet 2 of 2)

Primary Offset (Codec ID =00)	Secondary Offset (Codec ID =01)	Tertiary Offset (Codec ID =10)	NAMBAR Exposed Registers (D30:F2)
5Ah	DAh	15Ah	Vendor Reserved
7Ch	FCh	17Ch	Vendor ID1
7Eh	FEh	17Eh	Vendor ID2

NOTES:
1. Software should not try to access reserved registers
2. Primary Codec ID cannot be changed. Secondary codec ID can be changed via bits 1:0 of configuration register 40h. Tertiary codec ID can be changed via bits 3:2 of configuration register 40h.
3. The tertiary offset is only available through the memory space defined by the MMBAR register.

The Bus Master registers are located from offset + 00h to offset + 51h and reside in the AC '97 controller. Accesses to these registers do **not** cause the cycle to be forwarded over the AC-link to the codec. S/W could access these registers as bytes, word, dword or qword quantities, but reads must not cross dword boundaries.

In the case of the split codec implementation accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec, address offsets 80h–FFh for the secondary codec and address offsets 100h–17Fh for the tertiary codec.

The Global Control (GLOB_CNT) (D30:F2:2Ch) and Global Status (GLOB_STA) (D30:F2:30h) registers are aliased to the same global registers in the audio and modem I/O space. Therefore a read/write to these registers in either audio or modem I/O space affects the same physical register.

Bus Mastering registers exist in I/O space and reside in the AC '97 controller. The six channels, PCM in, PCM in 2, PCM out, Mic in, Mic 2, and S/PDIF out, each have their own set of Bus Mastering registers. The following register descriptions apply to all six channels. The register definition section titles use a generic "x_" in front of the register to indicate that the register applies to all six channels. The naming prefix convention used in Table 2-3 and in the register description I/O address is as follows:

PI = PCM in channel
PO = PCM out channel
MC = Mic in channel
MC2 = Mic 2 channel
PI2 = PCM in 2 channel
SP = S/PDIF out channel.

Table 2-3. Native Audio Bus Master Control Registers (Sheet 1 of 2)

Offset	Mnemonic	Name	Default	Access
00h	PI_BDBAR	PCM In Buffer Descriptor list Base Address	00000000h	R/W
04h	PI_CIV	PCM In Current Index Value	00h	RO
05h	PI_LVI	PCM In Last Valid Index	00h	R/W
06h	PI_SR	PCM In Status	0001h	R/WC, RO
08h	PI_PICB	PCM In Position in Current Buffer	0000h	RO
0Ah	PI_PIV	PCM In Prefetched Index Value	00h	RO
0Bh	PI_CR	PCM In Control	00h	R/W, R/W (special)
10h	PO_BDBAR	PCM Out Buffer Descriptor list Base Address	00000000h	R/W

Table 2-3. Native Audio Bus Master Control Registers (Sheet 2 of 2)

Offset	Mnemonic	Name	Default	Access
14h	PO_CIV	PCM Out Current Index Value	00h	RO
15h	PO_LVI	PCM Out Last Valid Index	00h	R/W
16h	PO_SR	PCM Out Status	0001h	R/WC, RO
18h	PO_PICB	PCM In Position In Current Buffer	0000h	RO
1Ah	PO_PIV	PCM Out Prefetched Index Value	00h	RO
1Bh	PO_CR	PCM Out Control	00h	R/W, R/W (special)
20h	MC_BDBAR	Mic. In Buffer Descriptor List Base Address	00000000h	R/W
24h	MC_CIV	Mic. In Current Index Value	00h	RO
25h	MC_LVI	Mic. In Last Valid Index	00h	R/W
26h	MC_SR	Mic. In Status	0001h	R/WC, RO
28h	MC_PICB	Mic. In Position In Current Buffer	0000h	RO
2Ah	MC_PIV	Mic. In Prefetched Index Value	00h	RO
2Bh	MC_CR	Mic. In Control	00h	R/W, R/W (special)
2Ch	GLOB_CNT	Global Control	00000000h	R/W, R/W (special)
30h	GLOB_STA	Global Status	See register description	R/W, R/WC, RO
34h	CAS	Codec Access Semaphore	00h	R/W (special)
40h	MC2_BDBAR	Mic. 2 Buffer Descriptor List Base Address	00000000h	R/W
44h	MC2_CIV	Mic. 2 Current Index Value	00h	RO
45h	MC2_LVI	Mic. 2 Last Valid Index	00h	R/W
46h	MC2_SR	Mic. 2 Status	0001h	RO, R/WC
48h	MC2_PICB	Mic 2 Position In Current Buffer	0000h	RO
4Ah	MC2_PIV	Mic. 2 Prefetched Index Value	00h	RO
4Bh	MC2_CR	Mic. 2 Control	00h	R/W, R/W (special)
50h	PI2_BDBAR	PCM In 2 Buffer Descriptor List Base Address	00000000h	R/W
54h	PI2_CIV	PCM In 2 Current Index Value	00h	RO
55h	PI2_LVI	PCM In 2 Last Valid Index	00h	R/W
56h	PI2_SR	PCM In 2 Status	0001h	R/WC, RO
58h	PI2_PICB	PCM In 2 Position in Current Buffer	0000h	RO
5Ah	PI2_PIV	PCM In 2 Prefetched Index Value	00h	RO
5Bh	PI2_CR	PCM In 2 Control	00h	R/W, R/W (special)
60h	SPBAR	S/PDIF Buffer Descriptor List Base Address	00000000h	R/W
64h	SPCIV	S/PDIF Current Index Value	00h	RO
65h	SPLVI	S/PDIF Last Valid Index	00h	R/W
66h	SPSR	S/PDIF Status	0001h	R/WC, RO
68h	SPPICB	S/PDIF Position In Current Buffer	0000h	RO
6Ah	SPPIV	S/PDIF Prefetched Index Value	00h	RO
6Bh	SPCR	S/PDIF Control	00h	R/W, R/W (special)
80h	SDM	SData_IN Map	00h	R/W, RO

Note: Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the registers shared with the AC '97 Modem (GCR, GSR, CASR). All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers and bits **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Fh – bits 6:0 Global Control (GLOB_CNT)
- offset 30h–33h – bits [29,15,11:10,0] Global Status (GLOB_STA)
- offset 34h – Codec Access Semaphore Register (CAS)
- Resume well registers and bits **will not** be reset by the $D3_{HOT}$ to D0 transition:
- offset 30h–33h – bits [17:16] Global Status (GLOB_STA)

2.2.1 *x*_BDBAR—Buffer Descriptor Base Address Register (Audio—D30:F2)

I/O Address:	NABMBAR + 00h (PIBDBAR), NABMBAR + 10h (POBDBAR), NABMBAR + 20h (MCBDBAR) MBBAR + 40h (MC2BDBAR) MBBAR + 50h (PI2BDBAR) MBBAR + 60h (SPBAR)	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Software can read the register at offset 00h by performing a single 32-bit read from address offset 00h. Reads across dword boundaries are not supported.

Bit	Description
31:3	**Buffer Descriptor Base Address[31:3]** — R/W. These bits represent address bits 31:3. The data should be aligned on 8-byte boundaries. Each buffer descriptor is 8 bytes long and the list can contain a maximum of 32 entries.
2:0	Hardwired to 0.

2.2.2 x_CIV—Current Index Value Register (Audio—D30:F2)

I/O Address:	NABMBAR + 04h (PICIV), NABMBAR + 14h (POCIV), NABMBAR + 24h (MCCIV) MBBAR + 44h (MC2CIV) MBBAR + 54h (PI2CIV) MBBAR + 64h (SPCIV)	Attribute:	RO
Default Value:	00h	Size:	8 bits

Lockable: No Power Well: Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 04h.

Bit	Description
7:5	Hardwired to 0
4:0	**Current Index Value [4:0]** — RO. These bits represent which buffer descriptor within the list of 32 descriptors is currently being processed. As each descriptor is processed, this value is incremented. The value rolls over after it reaches 31.

NOTE: Reads across dword boundaries are not supported.

2.2.3 x_LVI—Last Valid Index Register (Audio—D30:F2)

I/O Address:	NABMBAR + 05h (PILVI), NABMBAR + 15h (POLVI), NABMBAR + 25h (MCLVI) MBBAR + 45h (MC2LVI) MBBAR + 55h (PI2LVI) MBBAR + 65h (SPLVI)	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 05h.

Bit	Description
7:5	Hardwired to 0.
4:0	**Last Valid Index [4:0]** — R/W. This value represents the last valid descriptor in the list. This value is updated by the software each time it prepares a new buffer and adds it to the list.

NOTE: Reads across dword boundaries are not supported.

2.2.4 *x*_SR—Status Register (Audio—D30:F2)

I/O Address:	NABMBAR + 06h (PISR), NABMBAR + 16h (POSR), NABMBAR + 26h (MCSR) MBBAR + 46h (MC2SR) MBBAR + 56h (PI2SR) MBBAR + 66h (SPSR)	Attribute:	R/WC, RO
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 16-bit read to offset 06h. Reads across dword boundaries are not supported.

Bit	Description
15:5	Reserved.
4	**FIFO Error (FIFOE)** — R/WC. Software clears this bit by writing a 1 to it. 0 = No FIFO error. 1 = FIFO error occurs. **PISR Register:** FIFO error indicates a FIFO overrun. The FIFO pointers don't increment, the incoming data is not written into the FIFO, thus is lost. **POSR Register:** FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample. The ICH6 will set the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process.
3	**Buffer Completion Interrupt Status (BCIS)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = Set by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion (IOC) bit is set in the command byte of the buffer descriptor. It remains active until cleared by software.
2	**Last Valid Buffer Completion Interrupt (LVBCI)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = Last valid buffer has been processed. It remains active until cleared by software. This bit indicates the occurrence of the event signified by the last valid buffer being processed. Thus this is an event status bit that can be cleared by software once this event has been recognized. This event will cause an interrupt if the enable bit (D30:F2:NABMBAR + 0Bh, bit 2) in the Control Register is set. The interrupt is cleared when the software clears this bit. In the case of *Transmits* (PCM out, Modem out) this bit is set, after the last valid buffer has been fetched (not after transmitting it). While in the case of *Receives*, this bit is set after the data for the last buffer has been written to memory.
1	**Current Equals Last Valid (CELV)** — RO. 0 = Cleared by hardware when controller exists state (i.e., until a new value is written to the LVI register.) 1 = Current Index is equal to the value in the Last Valid Index Register (D30:F2:NABMBAR + 05h), and the buffer pointed to by the CIV has been processed (i.e., after the last valid buffer has been processed). This bit is very similar to bit 2, except this bit reflects the state rather than the event. This bit reflects the state of the controller, and remains set until the controller exits this state.
0	**DMA Controller Halted (DCH)** — RO. 0 = Running. 1 = Halted. This could happen because of the Start/Stop bit being cleared and the DMA engines are idle, or it could happen once the controller has processed the last valid buffer.

2.2.5 x_PICB—Position In Current Buffer Register (Audio—D30:F2)

I/O Address:	NABMBAR + 08h (PIPICB), NABMBAR + 18h (POPICB), NABMBAR + 28h (MCPICB) MBBAR + 48h (MC2PICB) MBBAR + 58h (PI2PICB) MBBAR + 68h (SPPICB)	Attribute:	RO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 16-bit read to offset 08h. Reads across dword boundaries are not supported.

Bit	Description
15:0	**Position In Current Buffer [15:0]** — RO. These bits represent the number of samples left to be processed in the current buffer. Once again, this means, the number of samples not yet read from memory (in the case of reads from memory) or not yet written to memory (in the case of writes to memory), irrespective of the number of samples that have been transmitted/received across AC-link.

2.2.6 x_PIV—Prefetched Index Value Register (Audio—D30:F2)

I/O Address:	NABMBAR + 0Ah (PIPIV), NABMBAR + 1Ah (POPIV), NABMBAR + 2Ah (MCPIV) MBBAR + 4Ah (MC2PIV) MBBAR + 5Ah (PI2PIV) MBBAR + 6Ah (SPPIV)	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Ah. Reads across dword boundaries are not supported.

Bit	Description
7:5	Hardwired to 0.
4:0	**Prefetched Index Value [4:0]** — RO. These bits represent which buffer descriptor in the list has been prefetched. The bits in this register are also modulo 32 and roll over after they reach 31.

2.2.7 x_CR—Control Register (Audio—D30:F2)

I/O Address:	NABMBAR + 0Bh (PICR), NABMBAR + 1Bh (POCR), NABMBAR + 2Bh (MCCR) MBBAR + 4Bh (MC2CR) MBBAR + 5Bh (PI2CR) MBBAR + 6Bh (SPCR)	Attribute:	R/W, R/W (special)
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Bh. Reads across dword boundaries are not supported.

Bit	Description
7:5	Reserved.
4	**Interrupt on Completion Enable (IOCE)** — R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. 0 = Disable. Interrupt will not occur. 1 = Enable.
3	**FIFO Error Interrupt Enable (FEIE)** — R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not. 0 = Disable. Bit 4 in the Status Register will be set, but the interrupt will not occur. 1 = Enable. Interrupt will occur.
2	**Last Valid Buffer Interrupt Enable (LVBIE)** — R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not. 0 = Disable. Bit 2 in the Status register will still be set, but the interrupt will not occur. 1 = Enable.
1	**Reset Registers (RR)** — R/W (special). 0 = Removes reset condition. 1 = Contents of all Bus master related registers to be reset, except the interrupt enable bits (bit 4,3,2 of this register). Software needs to set this bit but need not clear it since the bit is self clearing. This bit must be set only when the Run/Pause bit (D30:F2:2Bh, bit 0) is cleared. Setting it when the Run bit is set will cause undefined consequences.
0	**Run/Pause Bus Master (RPBM)** — R/W. 0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed). 1 = Run. Bus master operation starts.

2.2.8 GLOB_CNT—Global Control Register (Audio—D30:F2)

I/O Address:	NABMBAR + 2Ch	Attribute:	R/W, R/W (special)
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:30	**S/PDIF Slot Map (SSM)** — R/W. If the run/pause bus master bit (bit 0 of offset 2Bh) is set, then the value in these bits indicate which slots S/PDIF data is transmitted on. Software must ensure that the programming here does not conflict with the PCM channels being used. If there is a conflict, unpredictable behavior will result — the hardware will not check for a conflict. 00 = Reserved 01 = Slots 7 and 8 10 = Slots 6 and 9 11 = Slots 10 and 11
29:24	Reserved.
23:22	**PCM Out Mode (POM)** — R/W. Enables the PCM out channel to use 16 or 20-bit audio on PCM out. This does not affect the microphone of S/PDIF DMA. When greater than 16 bit audio is used, the data structures are aligned as 32-bits per sample, with the highest order bits representing the data, and the lower order bits as don't care. 00 = 16 bit audio (default) 01 = 20 bit audio 10 = Reserved. If set, indeterminate behavior will result. 11 = Reserved. If set, indeterminate behavior will result.
21:20	**PCM 4/6 Enable** — R/W. This field configures PCM Output for 2, 4 or 6 channel mode. 00 = 2-channel mode (default) 01 = 4-channel mode 10 = 6-channel mode 11 = Reserved
19:7	Reserved.
6	**ACZ_SDIN2 Interrupt Enable** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on the ACZ_SDIN2 causes a resume event on the AC-link. **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.
5	**ACZ_SDIN1 Interrupt Enable** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on the ACZ_SDIN1 causes a resume event on the AC-link. **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.
4	**ACZ_SDIN0 Interrupt Enable** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on ACZ_SDIN0 causes a resume event on the AC-link. **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.
3	**AC-LINK Shut Off (LSO)** — R/W. 0 = Normal operation. 1 = Controller disables all outputs which will be pulled low by internal pull down resistors. **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.

Bit	Description
2	**AC '97 Warm Reset** — R/W (special). 0 = Normal operation. 1 = Writing a 1 to this bit causes a warm reset to occur on the AC-link. The warm reset will awaken a suspended codec without clearing its internal registers. If software attempts to perform a warm reset while bit_clk is running, the write will be ignored and the bit will not change. This bit is self-clearing (it remains set until the reset completes and bit_clk is seen on the AC-link, after which it clears itself). **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.
1	**AC '97 Cold Reset#** — R/W. 0 = Writing a 0 to this bit causes a cold reset to occur throughout the AC '97 circuitry. All data in the controller and the codec will be lost. Software needs to clear this bit no sooner than the minimum number of ms have elapsed. 1 = This bit defaults to 0 and hence after reset, the driver needs to set this bit to a 1. The value of this bit is retained after suspends; hence, if this bit is set to a 1 prior to suspending, a cold reset is not generated automatically upon resuming. **NOTE:** This bit is in the core well and is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.
0	**GPI Interrupt Enable (GIE)** — R/W. This bit controls whether the change in status of any GPI causes an interrupt. 0 = Bit 0 of the Global Status Register is set, but no interrupt is generated. 1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register. **NOTE:** This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 reset.

NOTE: Reads across dword boundaries are not supported.

2.2.9 GLOB_STA—Global Status Register (Audio—D30:F2)

I/O Address: NABMBAR + 30h Attribute: RO, R/W, R/WC
Default Value: 00x0xxx01110000000000xxxxx00xxxbSize: 32 bits
Lockable: No Power Well: Core

Bit	Description
31:30	Reserved.
29	**ACZ_SDIN2 Resume Interrupt (S2RI)** — R/WC. This bit indicates a resume event occurred on ACZ_SDIN2. Software clears this bit by writing a 1 to it. 0 = Resume event did Not occur. 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
28	**ACZ_SDIN2 Codec Ready (S2CR)** — RO. Reflects the state of the codec ready bit on ACZ_SDIN2. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
27	**Bit Clock Stopped (BCS)** — RO. This bit indicates that the bit clock is not running. 0 = Transition is found on BIT_CLK. 1 = ICH6 detected that there has been no transition on BIT_CLK for four consecutive PCI clocks.
26	**S/PDIF Interrupt (SPINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = S/PDIF out channel interrupt status bits have been set.
25	**PCM In 2 Interrupt (P2INT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM In 2 channel status bits have been set.
24	**Microphone 2 In Interrupt (M2INT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the Mic in channel interrupts status bits has been set.
23:22	**Sample Capabilities** — RO. This field indicates the capability to support more greater than 16-bit audio. 00 = Reserved 01 = 16 and 20-bit Audio supported (ICH6 value) 10 = Reserved 11 = Reserved
21:20	**Multichannel Capabilities**— RO. This field indicates the capability to support more 4 and 6 channels on PCM Out.
19:18	Reserved.
17	**MD3** — R/W. Power down semaphore for Modem. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the AD3 bit to coordinate the entry of the two codecs into D3 state. This bit is not affected by $D3_{HOT}$ to D0 Reset.
16	**AD3** — R/W. Power down semaphore for Audio. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the MD3 bit to coordinate the entry of the two codecs into D3 state. This bit is not affected by $D3_{HOT}$ to D0 Reset.

Bit	Description
15	**Read Completion Status (RCS)** — R/WC. This bit indicates the status of codec read completions. 0 = A codec read completes normally. 1 = A codec read results in a time-out. The bit remains set until being cleared by software writing a 1 to the bit location. This bit is not affected by $D3_{HOT}$ to D0 Reset.
14	**Bit 3 of Slot 12** — RO. Display bit 3 of the most recent slot 12.
13	**Bit 2 of Slot 12** — RO. Display bit 2 of the most recent slot 12.
12	**Bit 1 of slot 12** — RO. Display bit 1 of the most recent slot 12.
11	**ACZ_SDIN1 Resume Interrupt (S1R1)** — R/WC. This bit indicates that a resume event occurred on ACZ_SDIN1. Software clears this bit by writing a 1 to it. 0 = Resume event did Not occur 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
10	**ACZ_SDIN0 Resume Interrupt (S0R1)** — R/WC. This bit indicates that a resume event occurred on ACZ_SDIN0. Software clears this bit by writing a 1 to it. 0 = Resume event did Not occur 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
9	**ACZ_SDIN1 Codec Ready (S1CR)** — RO. Reflects the state of the codec ready bit in ACZ_SDIN1. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
8	**ACZ_SDIN0 Codec Ready (S0CR)** — RO. Reflects the state of the codec ready bit in ACZ_SDIN0. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
7	**Microphone In Interrupt (MINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the Mic in channel interrupts status bits has been set.
6	**PCM Out Interrupt (POINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM out channel interrupts status bits has been set.
5	**PCM In Interrupt (PIINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM in channel interrupts status bits has been set.
4:3	Reserved
2	**Modem Out Interrupt (MOINT**) — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the modem out channel interrupts status bits has been set.
1	**Modem In Interrupt (MIINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the modem in channel interrupts status bits has been set.
0	**GPI Status Change Interrupt (GSCI)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = This bit reflects the state of bit 0 in slot 12, and is set when bit 0 of slot 12 is set. This indicates that one of the GPI's changed state, and that the new values are available in slot 12. This bit is not affected by AC '97 Audio Function $D3_{HOT}$ to D0 Reset.

NOTE: Reads across dword boundaries are not supported.

2.2.10 CAS—Codec Access Semaphore Register (Audio—D30:F2)

I/O Address:	NABMBAR + 34h	Attribute:	R/W (special)
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:1	Reserved.
0	**Codec Access Semaphore (CAS)** — R/W (special). This bit is read by software to check whether a codec access is currently in progress. 0 = No access in progress. 1 = The act of reading this register sets this bit to 1. The driver that read this bit can then perform an I/O access. Once the access is completed, hardware automatically clears this bit.

NOTE: Reads across dword boundaries are not supported.

2.2.11 SDM—SDATA_IN Map Register (Audio—D30:F2)

I/O Address:	NABMBAR + 80h	Attribute:	R/W, RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:6	**PCM In 2, Microphone In 2 Data In Line (DI2L)**— R/W. When the SE bit is set, these bits indicates which ACZ_SDIN line should be used by the hardware for decoding the input slots for PCM In 2 and Microphone In 2. When the SE bit is cleared, the value of these bits are irrelevant, and PCM In 2 and Mic In 2 DMA engines are not available. 00 = ACZ_SDIN0 01 = ACZ_SDIN1 10 = ACZ_SDIN2 11 = Reserved
5:4	**PCM In 1, Microphone In 1 Data In Line (DI1L)**— R/W. When the SE bit is set, these bits indicates which ACZ_SDIN line should be used by the hardware for decoding the input slots for PCM In 1 and Microphone In 1. When the SE bit is cleared, the value of these bits are irrelevant, and the PCM In 1 and Mic In 1 engines use the OR'd ACZ_SDIN lines. 00 = ACZ_SDIN0 01 = ACZ_SDIN1 10 = ACZ_SDIN2 11 = Reserved
3	**Steer Enable (SE)** — R/W. When set, the ACZ_SDIN lines are treated separately and not OR'd together before being sent to the DMA engines. When cleared, the ACZ_SDIN lines are OR'd together, and the "Microphone In 2" and "PCM In 2" DMA engines are not available.
2	Reserved — RO.
1:0	**Last Codec Read Data Input (LDI)** — RO. When a codec register is read, this indicates which ACZ_SDIN the read data returned on. Software can use this to determine how the codecs are mapped. The values are: 00 = ACZ_SDIN0 01 = ACZ_SDIN1 10 = ACZ_SDIN2 11 = Reserved

NOTE: Reads across dword boundaries are not supported.

The following pseudo code presents a modified version of the routine that prepares buffers and inserts them into the ICH6 AC '97 buffer descriptor list. In contrast to the version of this routine given in Section 2.2.3, in this version tail points to the last good (i.e., non-spurious) buffer in the list. Furthermore, because the ICH6 AC '97 DMA engine prefetches the next buffer descriptor we split the buffer generated by the datapump into two parts, with the second as small as practical and denote this size as `MinBufferLength` (here assumed to be eight samples = 4 dwords = 500 µs. at 16 kHz). To simplify, we assume that only a single buffer is generated by the datapump at a time and we ignore checking for the end of the descriptor list (i.e., the addition is implicitly modulo 32).

```
while (tail <= Prefetched_Index)
{
    tail++;  // Happens IFF Spurious Data was used
}
if (((tail <= LastValidIndex) || (tail == free)) &&
    (((tail+1) <= LastValidIndex) || ((tail+1) == free)))
{
    Descriptor.BufferPtr[tail] = &buffer;
    Descriptor.BufferLength[tail] = length(buffer) - MinBufferLength;
    Descriptor.BufferPtr[tail+1] =
        &buffer + length(buffer) - MinBufferLength;
    Descriptor.BufferLength[tail+1] = MinBufferLength;
    tail += 2;
 }
else
{
    ;    //error: no space for this data buffer
}
if ((tail <= LastValid index) || (tail == free))
    Descriptor.BufferPtr[tail] =
        &(SpuriousBufferList[LastState(buffer)]);
    Descriptor.BufferLength[tail] =
        SpuriousBufferLength[LastState(buffer)];
    LastValidIndex = tail;
    //Note: tail is NOT incremented, so next time we'll overwrite this
    //      descriptor, which is the whole point of this algorithm
}
else
{
    LastValidIndex = tail-1;//warning: no space for spurious data buffer
}
```

The above implementation can be improved upon in a number of ways. First, rather than adding a single (large) spurious buffer, a number of smaller ones could be chained together. In this way, the amount of spurious data actually transmitted would be reduced while still maintaining a given level of protection against long latencies for the host-based software. Additionally, the implementation could be extended to handle multiple buffers at once, inserting several buffers in a row and only splitting the last one and then appending a spurious buffer or buffers. Finally, the descriptor list is a circular buffer and a real implementation would have to check `tail and tail+1 against base_address + 31 * 8.`

The following pseudo code presents a modified version of the routine that prepares buffers and inserts them into the ICH6 AC '97 buffer descriptor list. In contrast to the version of this routine given in Section 2.2.3, in this version tail points to the last good (i.e., non-spurious) buffer in the list. Furthermore, because the ICH6 AC '97 DMA engine prefetches the next buffer descriptor we split the buffer generated by the datapump into two parts, with the second as small as practical and denote this size as `MinBufferLength` (here assumed to be eight samples = 4 dwords = 500 µs. at 16 kHz). To simplify, we assume that only a single buffer is generated by the datapump at a time and we ignore checking for the end of the descriptor list (i.e., the addition is implicitly modulo 32).

```
while (tail <= Prefetched_Index)
{
    tail++;  // Happens IFF Spurious Data was used
}
if (((tail <= LastValidIndex) || (tail == free)) &&
    (((tail+1) <= LastValidIndex) || ((tail+1) == free)))
{
    Descriptor.BufferPtr[tail] = &buffer;
    Descriptor.BufferLength[tail] = length(buffer) - MinBufferLength;
    Descriptor.BufferPtr[tail+1] =
        &buffer + length(buffer) - MinBufferLength;
    Descriptor.BufferLength[tail+1] = MinBufferLength;
    tail += 2;
 }
else
{
    ;    //error: no space for this data buffer
}
if ((tail <= LastValid index) || (tail == free))
    Descriptor.BufferPtr[tail] =
        &(SpuriousBufferList[LastState(buffer)]);
    Descriptor.BufferLength[tail] =
        SpuriousBufferLength[LastState(buffer)];
    LastValidIndex = tail;
    //Note: tail is NOT incremented, so next time we'll overwrite this
    //      descriptor, which is the whole point of this algorithm
}
else
{
    LastValidIndex = tail-1;//warning: no space for spurious data buffer
}
```

The above implementation can be improved upon in a number of ways. First, rather than adding a single (large) spurious buffer, a number of smaller ones could be chained together. In this way, the amount of spurious data actually transmitted would be reduced while still maintaining a given level of protection against long latencies for the host-based software. Additionally, the implementation could be extended to handle multiple buffers at once, inserting several buffers in a row and only splitting the last one and then appending a spurious buffer or buffers. Finally, the descriptor list is a circular buffer and a real implementation would have to check `tail and tail+1 against base_address + 31 * 8.`

§

3 AC ’97 Modem Controller Registers (D30:F3)

3.1 AC ’97 Modem PCI Configuration Space (D30:F3)

Note: Registers that are not shown should be treated as Reserved.

Table 3-1. AC ‘97 Modem PCI Register Address Map (Modem—D30:F3)

Offset	Mnemonic	Register	Default	Access
00–01h	VID	Vendor Identification	8086	RO
02–03h	DID	Device Identification	266Dh	RO
04–05h	PCICMD	PCI Command	0000h	R/W, RO
06–07h	PCISTS	PCI Status	0290h	R/WC, RO
08h	RID	Revision Identification	See register description	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	07h	RO
0Eh	HEADTYP	Header Type	00h	RO
10–13h	MMBAR	Modem Mixer Base Address	00000001h	R/W, RO
14–17h	MBAR	Modem Base Address	00000001h	R/W, RO
2C–2Dh	SVID	Subsystem Vendor Identification	0000h	R/WO
2E–2Fh	SID	Subsystem Identification	0000h	R/WO
34h	CAP_PTR	Capabilities Pointer	50h	RO
3Ch	INT_LN	Interrupt Line	00h	R/W
3Dh	INT_PN	Interrupt Pin	See register description	RO
50–51h	PID	PCI Power Management Capability ID	0001h	RO
52–53h	PC	Power Management Capabilities	C9C2h	RO
54–55h	PCS	Power Management Control and Status	0000h	R/W, R/WC

Note: Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the following BIOS programmed registers as BIOS may not be invoked following the D3-to-D0 transition. All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Dh – Subsystem Vendor ID (SVID)
- offset 2Eh–2Fh – Subsystem ID (SID)

Resume well registers **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 54h–55h – Power Management Control and Status (PCS)

3.1.1 VID—Vendor Identification Register (Modem—D30:F3)

Address Offset:	00–01h	Attribute:	RO
Default Value:	8086	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	Vendor ID.

3.1.2 DID—Device Identification Register (Modem—D30:F3)

Address Offset:	02–03h	Attribute:	RO
Default Value:	266Dh	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	Device ID.

3.1.3 PCICMD—PCI Command Register (Modem—D30:F3)

Address Offset:	04–05h	Attribute:	R/W, RO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCICMD is a 16-bit control register. Refer to the *PCI Local Bus Specification* for complete details on each bit.

Bit	Description
15:11	Reserved. Read 0.
10	**Interrupt Disable (ID)**— R/W. 0 = The INTx# signals may be asserted and MSIs may be generated. 1 = The AC '97 controller's INTx# signal will be de-asserted and it may not generate MSIs.
9	Fast Back to Back Enable (FBE) — RO. Not implemented. Hardwired to 0.
8	SERR# Enable (SERR_EN) — RO. Not implemented. Hardwired to 0.
7	Wait Cycle Control (WCC) — RO. Not implemented. Hardwired to 0.
6	Parity Error Response (PER) — RO. Not implemented. Hardwired to 0.
5	VGA Palette Snoop (VPS) — RO. Not implemented. Hardwired to 0.
4	Memory Write and Invalidate Enable (MWIE) — RO. Not implemented. Hardwired to 0.
3	Special Cycle Enable (SCE) — RO. Not implemented. Hardwired to 0.

Bit	Description
2	**Bus Master Enable (BME)** — R/W. This bit controls standard PCI bus mastering capabilities. 0 = Disable 1 = Enable
1	Memory Space Enable (MSE) — RO. Hardwired to 0, AC '97 does not respond to memory accesses.
0	**I/O Space Enable (IOSE)** — R/W. This bit controls access to the I/O space registers. 0 = Disable access. (default = 0). 1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit.

3.1.4 PCISTS—PCI Status Register (Modem—D30:F3)

Address Offset:	06–07h	Attribute:	R/WC, RO
Default Value:	0290h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCISTA is a 16-bit status register. Refer to the *PCI Local Bus Specification* for complete details on each bit.

Note: For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

Bit	Description
15	Detected Parity Error (DPE) — RO. Not implemented. Hardwired to 0.
14	Signaled System Error (SSE) —RO. Not implemented. Hardwired to 0.
13	**Master Abort Status (MAS)** — R/WC. 0 = Master abort Not generated by bus master AC '97 function. 1 = Bus Master AC '97 interface function, as a master, generates a master abort.
12	Reserved. Read as 0.
11	Signaled Target Abort (STA) — RO. Not implemented. Hardwired to 0.
10:9	DEVSEL# Timing Status (DEV_STS) — RO. This 2-bit field reflects the ICH6's DEVSEL# timing parameter. These read only bits indicate the ICH6's DEVSEL# timing when performing a positive decode.
8	Data Parity Error Detected (DPED) — RO. Not implemented. Hardwired to 0.
7	Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1. This bit indicates that the ICH6 as a target is capable of fast back-to-back transactions.
6	User Definable Features (UDF) — RO. Not implemented. Hardwired to 0.
5	66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0.
4	Capabilities List (CAP_LIST) — RO. Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h.
3	**Interrupt Status (INTS)** — RO. 0 = This bit is 0 after the interrupt is cleared. 1 = This bit is 1 when the INTx# is asserted.
2:0	Reserved

3.1.5 RID—Revision Identification Register (Modem—D30:F3)

Address Offset:	08h	Attribute:	RO
Default Value:	See bit description	Size:	8 Bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Revision ID — RO. Refer to Section 2.23 or the *Intel® ICH6/ICH6R/ICH6-M Specification Update* for the value of the Revision ID Register

3.1.6 PI—Programming Interface Register (Modem—D30:F3)

Address Offset:	09h	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Programming Interface — RO.

3.1.7 SCC—Sub Class Code Register (Modem—D30:F3)

Address Offset:	0Ah	Attribute:	RO
Default Value:	03h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Sub Class Code — RO. 03h = Generic Modem.

3.1.8 BCC—Base Class Code Register (Modem—D30:F3)

Address Offset:	0Bh	Attribute:	RO
Default Value:	07h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Base Class Code — RO. 07h = Simple Communications controller.

3.1.9 HEADTYP—Header Type Register (Modem—D30:F3)

Address Offset:	0Eh	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:0	Header Type — RO.

3.1.10 MMBAR—Modem Mixer Base Address Register (Modem—D30:F3)

Address Offset:	10–13h	Attribute:	R/W, RO
Default Value:	00000001h	Size:	32 bits

The Native PCI Mode Modem uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Modem Mixer software interface. The mixer requires 256 bytes of I/O space. All accesses to the mixer registers are forwarded over the AC-link to the codec where the registers reside.

In the case of the split codec implementation accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec.

Bit	Description
31:16	Hardwired to 0's.
15:8	**Base Address** — R/W. These bits are used in the I/O space decode of the Modem interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 Modem, the upper 16 bits are hardwired to 0, while bits 15:8 are programmable. This configuration yields a maximum I/O block size of 256 bytes for this base address.
7:1	Reserved. Read as 0
0	Resource Type Indicator (RTE) — RO. Hardwired to 1indicating a request for I/O space.

3.1.11 MBAR—Modem Base Address Register (Modem—D30:F3)

Address Offset:	14–17h	Attribute:	R/W, RO
Default Value:	00000001h	Size:	32 bits

The Modem function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Modem software interface. The Modem Bus Mastering register space requires 128 bytes of I/O space. All Modem registers reside in the controller, therefore cycles are **not** forwarded over the AC-link to the codec.

Bit	Description
31:16	Hardwired to 0's.
15:7	**Base Address** — R/W. These bits are used in the I/O space decode of the Modem interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 Modem, the upper 16 bits are hardwired to 0, while bits 15:7 are programmable. This configuration yields a maximum I/O block size of 128 bytes for this base address.
6:1	Reserved. Read as 0
0	Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating a request for I/O space.

3.1.12 SVID—Subsystem Vendor Identification Register (Modem—D30:F3)

Address Offset:	2C–2Dh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

The SVID register, in combination with the Subsystem ID register, enable the operating environment to distinguish one audio subsystem from the other(s). This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	**Subsystem Vendor ID** — R/WO.

3.1.13 SID—Subsystem Identification Register (Modem—D30:F3)

Address Offset:	2E–2Fh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

The SID register, in combination with the Subsystem Vendor ID register make it possible for the operating environment to distinguish one audio subsystem from another. This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	**Subsystem ID** — R/WO.

3.1.14 CAP_PTR—Capabilities Pointer Register (Modem—D30:F3)

Address Offset:	34h	Attribute:	RO
Default Value:	50h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates the offset for the capability pointer.

Bit	Description
7:0	Capabilities Pointer (CAP_PTR) — RO. This field indicates that the first capability pointer offset is offset 50h

3.1.15 INT_LN—Interrupt Line Register (Modem—D30:F3)

Address Offset:	3Ch	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates which PCI interrupt line is used for the AC '97 module interrupt.

Bit	Description
7:0	Interrupt Line (INT_LN) — R/W. This data is not used by the ICH6. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

3.1.16 INT_PIN—Interrupt Pin Register (Modem—D30:F3)

Address Offset:	3Dh	Attribute:	RO
Default Value:	See description	Size:	8 bits
Lockable:	No	Power Well:	Core

This register indicates which PCI interrupt pin is used for the AC ’97 modem interrupt. The AC ’97 interrupt is internally OR’d to the interrupt controller with the PIRQB# signal.

Bit	Description
7:3	Reserved
2:0	Interrupt Pin (INT_PN) — RO. This reflects the value of D30IP.AMIP in chipset configuration space.

3.1.17 PID—PCI Power Management Capability Identification Register (Modem—D30:F3)

Address Offset:	50h	Attribute:	RO
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:8	Next Capability (NEXT) — RO. This field indicates that this is the last item in the list.
7:0	Capability ID (CAP) — RO. This field indicates that this pointer is a message signaled interrupt capability.

3.1.18 PC—Power Management Capabilities Register (Modem—D30:F3)

Address Offset:	52h	Attribute:	RO
Default Value:	C9C2h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:11	PME Support — RO. This field indicates PME# can be generated from all D states.
10:9	Reserved.
8:6	Auxiliary Current — RO. This field reports 375 mA maximum Suspend well current required when in the $D3_{COLD}$ state.
5	Device Specific Initialization (DSI) — RO. This bit indicates that no device-specific initialization is required.
4	Reserved — RO.
3	PME Clock (PMEC) — RO. This bit indicates that PCI clock is not required to generate PME#.
2:0	Version (VS) — RO. This field indicates support for Revision 1.1 of the *PCI Power Management Specification.*

3.1.19 PCS—Power Management Control and Status Register (Modem—D30:F3)

Address Offset:	54h	Attribute:	R/W, R/WC
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Resume

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15	**PME Status (PMES)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set when the AC '97 controller would normally assert the PME# signal independent of the state of the PME_En bit. This bit resides in the resume well.
14:9	Reserved — RO.
8	**PME Enable (PMEE)** — R/W. 0 = Disable. 1 = Enable. When set, and if corresponding PMES is also set, the AC '97 controller sets the AC97_STS bit in the GPE0_STS register
7:2	Reserved — RO.
1:0	**Power State (PS)** — R/W. This field is used both to determine the current power state of the AC '97 controller and to set a new power state. The values are: 00 = D0 state 01 = not supported 10 = not supported 11 = $D3_{HOT}$ state When in the $D3_{HOT}$ state, the AC '97 controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs.

3.2 AC ’97 Modem I/O Space (D30:F3)

In the case of the split codec implementation accesses to the modem mixer registers in different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec. Table 3-2 shows the register addresses for the modem mixer registers.

Table 3-2. Intel® ICH6 Modem Mixer Register Configuration

Register		MMBAR Exposed Registers (D30:F3)
Primary	**Secondary**	**Name**
00h:38h	80h:B8h	Intel RESERVED
3Ch	BCh	Extended Modem ID
3Eh	BEh	Extended Modem Stat/Ctrl
40h	C0h	Line 1 DAC/ADC Rate
42h	*C2h*	*Line 2 DAC/ADC Rate*
44h	*C4h*	*Handset DAC/ADC Rate*
46h	C6h	Line 1 DAC/ADC Level Mute
48h	*C8h*	*Line 2 DAC/ADC Level Mute*
4Ah	*CAh*	*Handset DAC/ADC Level Mute*
4Ch	CCh	GPIO Pin Config
4Eh	CEh	GPIO Polarity/Type
50h	D0h	GPIO Pin Sticky
52h	D2h	GPIO Pin Wake Up
54h	D4h	GPIO Pin Status
56h	D6h	Misc. Modem AFE Stat/Ctrl
58h	D8h	AC ’97 Reserved
5Ah	DAh	Vendor Reserved
7Ch	FCh	Vendor ID1
7Eh	FEh	Vendor ID2

NOTES:
1. Registers in italics are for functions not supported by the ICH6
2. Software should not try to access reserved registers
3. The ICH6 supports a modem codec connected to ACZ_SDIN[2:0], as long as the Codec ID is 00 or 01. However, the ICH6 does not support more than one modem codec. For a complete list of topologies, see your ICH6 enabled Platform Design Guide.

The Global Control (GLOB_CNT) and Global Status (GLOB_STA) registers are aliased to the same global registers in the audio and modem I/O space. Therefore a read/write to these registers in either audio or modem I/O space affects the same physical register. Software could access these registers as bytes, word, dword quantities, but reads must not cross dword boundaries.

These registers exist in I/O space and reside in the AC '97 controller. The two channels, Modem in and Modem out, each have their own set of Bus Mastering registers. The following register descriptions apply to both channels. The naming prefix convention used is as follows:
MI = Modem in channel
MO = Modem out channel

Table 3-3. Modem Registers

Offset	Mnemonic	Name	Default	Access
00h–03h	MI_BDBAR	Modem In Buffer Descriptor List Base Address	00000000h	R/W
04h	MI_CIV	Modem In Current Index Value	00h	RO
05h	MI_LVI	Modem In Last Valid Index	00h	R/W
06h–07h	MI_SR	Modem In Status	0001h	R/WC, RO
08h–09h	MI_PICB	Modem In Position In Current Buffer	0000h	RO
0Ah	MI_PIV	Modem In Prefetch Index Value	00h	RO
0Bh	MI_CR	Modem In Control	00h	R/W, R/W (special)
10h–13h	MO_BDBAR	Modem Out Buffer Descriptor List Base Address	00000000h	R/W
14h	MO_CIV	Modem Out Current Index Value	00h	RO
15h	MO_LVI	Modem Out Last Valid	00h	R/W
16h–17h	MO_SR	Modem Out Status	0001h	R/WC, RO
18h–19h	MI_PICB	Modem In Position In Current Buffer	0000h	RO
1Ah	MO_PIV	Modem Out Prefetched Index	00h	RO
1Bh	MO_CR	Modem Out Control	00h	R/W, R/W (special)
3Ch–3Fh	GLOB_CNT	Global Control	00000000h	R/W, R/W (special)
40h–43h	GLOB_STA	Global Status	00300000h	RO, R/W, R/WC
44h	CAS	Codec Access Semaphore	00h	R/W (special)

NOTE:
1. MI = Modem in channel; MO = Modem out channel

Note: Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the registers shared with the AC '97 audio controller (GCR, GSR, CASR). All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers and bits **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 3Ch–3Fh – bits [6:0] Global Control (GLOB_CNT)
- offset 40h–43h – bits [29,15,11:10] Global Status (GLOB_STA)
- offset 44h – Codec Access Semaphore Register (CAS)

Resume well registers and bits **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 40h–43h – bits [17:16] Global Status (GLOB_STA)

3.2.1 x_BDBAR—Buffer Descriptor List Base Address Register (Modem—D30:F3)

I/O Address:	MBAR + 00h (MIBDBAR), MBAR + 10h (MOBDBAR)	Attribute:	R/W
Default Value:	00000000h	Size:	32bits
Lockable:	No	Power Well:	Core

Software can read the register at offset 00h by performing a single 32-bit read from address offset 00h. Reads across dword boundaries are not supported.

Bit	Description
31:3	**Buffer Descriptor List Base Address [31:3]** — R/W. These bits represent address bits 31:3. The entries should be aligned on 8-byte boundaries.
2:0	Hardwired to 0.

3.2.2 x_CIV—Current Index Value Register (Modem—D30:F3)

I/O Address:	MBAR + 04h (MICIV), MBAR + 14h (MOCIV),	Attribute:	RO
Default Value:	00h	Size:	8bits
Lockable:	No	Power Well:	Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 04h. Reads across dword boundaries are not supported.

Bit	Description
7:5	Hardwired to 0.
4:0	**Current Index Value [4:0]** — RO. These bits represent which buffer descriptor within the list of 16 descriptors is being processed currently. As each descriptor is processed, this value is incremented.

3.2.3 x_LVI—Last Valid Index Register (Modem—D30:F3)

I/O Address:	MBAR + 05h (MILVI), MBAR + 15h (MOLVI)	Attribute:	R/W
Default Value:	00h	Power Well:	Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 05h. Reads across dword boundaries are not supported.

Bit	Description
7:5	Hardwired to 0
4:0	**Last Valid Index [4:0]** — R/W. These bits indicate the last valid descriptor in the list. This value is updated by the software as it prepares new buffers and adds to the list.

3.2.4 *x*_SR—Status Register (Modem—D30:F3)

I/O Address:	MBAR + 06h (MISR), MBAR + 16h (MOSR)	Attribute:	R/WC, RO
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 16-bit read to offset 06h. Reads across dword boundaries are not supported.

Bit	Description
15:5	Reserved
4	**FIFO Error (FIFOE)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = FIFO error occurs. **Modem in:** FIFO error indicates a FIFO overrun. The FIFO pointers don't increment, the incoming data is not written into the FIFO, thereby being lost. **Modem out:** FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample. The ICH6 will set the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process.
3	**Buffer Completion Interrupt Status (BCIS)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = Set by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion (IOC) bit is set in the command byte of the buffer descriptor. Remains active until software clears bit.
2	**Last Valid Buffer Completion Interrupt (LVBCI)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = Set by hardware when last valid buffer has been processed. It remains active until cleared by software. This bit indicates the occurrence of the event signified by the last valid buffer being processed. Thus, this is an event status bit that can be cleared by software once this event has been recognized. This event will cause an interrupt if the enable bit in the Control Register is set. The interrupt is cleared when the software clears this bit. In the case of transmits (PCM out, Modem out) this bit is set, after the last valid buffer has been fetched (not after transmitting it). While in the case of Receives, this bit is set after the data for the last buffer has been written to memory.
1	**Current Equals Last Valid (CELV)** — RO. 0 = Hardware clears when controller exists state (i.e., until a new value is written to the LVI register). 1 = Current Index is equal to the value in the Last Valid Index Register, AND the buffer pointed to by the CIV has been processed (i.e., after the last valid buffer has been processed). This bit is very similar to bit 2, except, this bit reflects the state rather than the event. This bit reflects the state of the controller, and remains set until the controller exits this state.
0	**DMA Controller Halted (DCH)** — RO. 0 = Running. 1 = Halted. This could happen because of the Start/Stop bit being cleared and the DMA engines are idle, or it could happen once the controller has processed the last valid buffer.

3.2.5 x_PICB—Position in Current Buffer Register (Modem—D30:F3)

I/O Address:	MBAR + 08h (MIPICB), MBAR + 18h (MOPICB),	Attribute:	RO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 16-bit read to offset 08h. Reads across dword boundaries are not supported.

Bit	Description
15:0	**Position In Current Buffer[15:0]** — RO. These bits represent the number of samples left to be processed in the current buffer.

3.2.6 x_PIV—Prefetch Index Value Register (Modem—D30:F3)

I/O Address:	MBAR + 0Ah (MIPIV), MBAR + 1Ah (MOPIV)	Attribute:	RO
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Ah. Reads across dword boundaries are not supported.

Bit	Description
7:5	Hardwired to 0
4:0	**Prefetched Index Value [4:0]** — RO. These bits represent which buffer descriptor in the list has been prefetched.

3.2.7 *x*_CR—Control Register (Modem—D30:F3)

I/O Address:	MBAR + 0Bh (MICR), MBAR + 1Bh (MOCR)	Attribute:	R/W, R/W (special)
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Bh. Reads across dword boundaries are not supported.

Bit	Description
7:5	Reserved
4	**Interrupt on Completion Enable (IOCE)** — R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. 0 = Disable 1 = Enable
3	**FIFO Error Interrupt Enable (FEIE)** — R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not. 0 = Disable. Bit 4 in the Status Register will be set, but the interrupt will not occur. 1 = Enable. Interrupt will occur
2	**Last Valid Buffer Interrupt Enable (LVBIE)** — R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not. 0 = Disable. Bit 2 in the Status register will still be set, but the interrupt will not occur. 1 = Enable
1	**Reset Registers (RR)** — R/W (special). 0 = Removes reset condition. 1 = Contents of all registers to be reset, except the interrupt enable bits (bit 4,3,2 of this register). Software needs to set this bit. It must be set only when the Run/Pause bit is cleared. Setting it when the Run bit is set will cause undefined consequences. This bit is self-clearing (software needs not clear it).
0	**Run/Pause Bus Master (RPBM)** — R/W. 0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed). 1 = Run. Bus master operation starts.

3.2.8 GLOB_CNT—Global Control Register (Modem—D30:F3)

I/O Address:	MBAR + 3Ch	Attribute:	R/W, R/W (special)
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:6	Reserved.
6	**ACZ_SDIN2 Interrupt Enable (S2RE)** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on the ACZ_SDIN2 causes a resume event on the AC-link.
5	**ACZ_SDIN1 Resume Interrupt Enable (S1RE)** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on the ACZ_SDIN1 causes a resume event on the AC-link.
4	**ACZ_SDIN0 Resume Interrupt Enable (S0RE)** — R/W. 0 = Disable. 1 = Enable an interrupt to occur when the codec on ACZ_SDIN0 causes a resume event on the AC-link.
3	**AC-LINK Shut Off (LSO)** — R/W. 0 = Normal operation. 1 = Controller disables all outputs which will be pulled low by internal pull down resistors.
2	**AC ’97 Warm Reset** — R/W (special). 0 = Normal operation. 1 = Writing a 1 to this bit causes a warm reset to occur on the AC-link. The warm reset will awaken a suspended codec without clearing its internal registers. If software attempts to perform a warm reset while bit_clk is running, the write will be ignored and the bit will not change. This bit is self-clearing (it remains set until the reset completes and bit_clk is seen on the AC-link, after which it clears itself).
1	**AC ’97 Cold Reset#** — R/W. 0 = Writing a 0 to this bit causes a cold reset to occur throughout the AC ‘97 circuitry. All data in the controller and the codec will be lost. Software needs to clear this bit no sooner than the minimum number of ms have elapsed. 1 = This bit defaults to 0 and hence after reset, the driver needs to set this bit to a 1. The value of this bit is retained after suspends; hence, if this bit is set to a 1 prior to suspending, a cold reset is not generated automatically upon resuming. Note: This bit is in the Core well.
0	**GPI Interrupt Enable (GIE)** — R/W. This bit controls whether the change in status of any GPI causes an interrupt. 0 = Bit 0 of the Global Status Register is set, but no interrupt is generated. 1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register. **NOTE:** This bit is cleared by the AC ‘97 Modem function $D3_{HOT}$ to D0 reset.

NOTE: Reads across dword boundaries are not supported.

3.2.9 GLOB_STA—Global Status Register (Modem—D30:F3)

I/O Address:	MBAR + 40h	Attribute:	RO, R/W, R/WC
Default Value:	00300000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:30	Reserved.
29	**ACZ_SDIN2 Resume Interrupt (S2RI)** — R/WC. This bit indicates a resume event occurred on ACZ_SDIN2. 0 = Software clears this bit by writing a 1 to it. 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
28	**ACZ_SDIN2 Codec Ready (S2CR)** — RO. This bit reflects the state of the codec ready bit on ACZ_SDIN2. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
27	**Bit Clock Stopped (BCS)** — RO. This bit indicates that the bit clock is not running. 0 = Transition is found on BIT_CLK. 1 = ICH6 detects that there has been no transition on BIT_CLK for four consecutive PCI clocks.
26	**S/PDIF* Interrupt (SPINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = S/PDIF out channel interrupt status bits have been set.
25	**PCM In 2 Interrupt (P2INT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM In 2 channel status bits have been set.
24	**Microphone 2 In Interrupt (M2INT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the Mic in channel interrupts status bits has been set.
23:22	Sample Capabilities — RO. This field indicates the capability to support more greater than 16-bit audio. 00 = Reserved 01 = 16 and 20-bit Audio supported (ICH6 value) 10 = Reserved 11 = Reserved
21:20	**Multichannel Capabilities** — RO. This field indicates the capability to support 4 and 6 channels on PCM Out.
19:18	Reserved.
17	**MD3** — R/W. Power down semaphore for Modem. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the AD3 bit to coordinate the entry of the two codecs into D3 state. This bit is not affected by $D3_{HOT}$ to D0 Reset.
16	**AD3** — R/W. Power down semaphore for Audio. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the MD3 bit to coordinate the entry of the two codecs into D3 state. This bit is not affected by $D3_{HOT}$ to D0 Reset.
15	**Read Completion Status (RCS)** — R/WC. This bit indicates the status of codec read completions. Software clears this bit by writing a 1 to it. 0 = A codec read completes normally. 1 = A codec read results in a time-out. This bit is not affected by $D3_{HOT}$ to D0 Reset.

Bit	Description
14	**Bit 3 of Slot 12** — RO. Display bit 3 of the most recent slot 12.
13	**Bit 2 of Slot 12** — RO. Display bit 2 of the most recent slot 12.
12	**Bit 1 of Slot 12** — RO. Display bit 1 of the most recent slot 12.
11	**ACZ_SDIN1 Resume Interrupt (S1RI)** — R/WC. This bit indicates that a resume event occurred on ACZ_SDIN1. Software clears this bit by writing a 1 to it. 0 = Resume event did Not occur. 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
10	**ACZ_SDIN0 Resume Interrupt (S0RI)** — R/WC. This bit indicates that a resume event occurred on ACZ_SDIN0. Software clears this bit by writing a 1 to it. 0 = Resume event did Not occur. 1 = Resume event occurred. This bit is not affected by $D3_{HOT}$ to D0 Reset.
9	**ACZ_SDIN1 Codec Ready (S1CR)** — RO. This bit reflects the state of the codec ready bit in ACZ_SDIN1. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
8	**ACZ_SDIN0 Codec Ready (S0CR)** — RO. This bit reflects the state of the codec ready bit in ACZ_SDIN 0. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously. 0 = Not Ready. 1 = Ready.
7	**Microphone In Interrupt (MINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the Mic in channel interrupts status bits has been set.
6	**PCM Out Interrupt (POINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM out channel interrupts status bits has been set.
5	**PCM In Interrupt (PIINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the PCM in channel interrupts status bits has been set.
4:3	Reserved
2	**Modem Out Interrupt (MOINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the modem out channel interrupts status bits has been set.
1	**Modem In Interrupt (MIINT)** — RO. 0 = When the specific status bit is cleared, this bit will be cleared. 1 = One of the modem in channel interrupts status bits has been set.
0	**GPI Status Change Interrupt (GSCI)** — R/WC. 0 = Software clears this bit by writing a 1 to it. 1 = This bit reflects the state of bit 0 in slot 12, and is set when bit 0 of slot 12 is set. This indicates that one of the GPI's changed state, and that the new values are available in slot 12. This bit is not affected by AC '97 Audio Modem function $D3_{HOT}$ to D0 Reset.

NOTE: Reads across dword boundaries are not supported.

Note: On reads from a codec, the controller will give the codec a maximum of four frames to respond, after which if no response is received, it will return a dummy read completion to the processor (with all F's on the data) and also set the Read Completion Status bit in the Global Status Register.

3.2.10 CAS—Codec Access Semaphore Register (Modem—D30:F3)

I/O Address:	NABMBAR + 44h	Attribute:	R/W (special)
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:1	Reserved
0	**Codec Access Semaphore (CAS)** — R/W (special). This bit is read by software to check whether a codec access is currently in progress. 0 = No access in progress. 1 = The act of reading this register sets this bit to 1. The driver that read this bit can then perform an I/O access. Once the access is completed, hardware automatically clears this bit.

NOTE: Reads across dword boundaries are not supported.

The following pseudo code presents a modified version of the routine that prepares buffers and inserts them into the ICH6 AC '97 buffer descriptor list. In contrast to the version of this routine given in Section 2.2.3, in this version tail points to the last good (i.e., non-spurious) buffer in the list. Furthermore, because the ICH6 AC '97 DMA engine prefetches the next buffer descriptor we split the buffer generated by the datapump into two parts, with the second as small as practical and denote this size as `MinBufferLength` (here assumed to be eight samples = 4 dwords = 500 µs. at 16 kHz). To simplify, we assume that only a single buffer is generated by the datapump at a time and we ignore checking for the end of the descriptor list (i.e., the addition is implicitly modulo 32).

```
while (tail <= Prefetched_Index)
{
    tail++;  // Happens IFF Spurious Data was used
}
if (((tail <= LastValidIndex) || (tail == free)) &&
    (((tail+1) <= LastValidIndex) || ((tail+1) == free)))
{
    Descriptor.BufferPtr[tail] = &buffer;
    Descriptor.BufferLength[tail] = length(buffer) - MinBufferLength;
    Descriptor.BufferPtr[tail+1] =
        &buffer + length(buffer) - MinBufferLength;
    Descriptor.BufferLength[tail+1] = MinBufferLength;
    tail += 2;
 }
else
{
    ;    //error: no space for this data buffer
}
if ((tail <= LastValid index) || (tail == free))
    Descriptor.BufferPtr[tail] =
        &(SpuriousBufferList[LastState(buffer)]);
    Descriptor.BufferLength[tail] =
        SpuriousBufferLength[LastState(buffer)];
    LastValidIndex = tail;
    //Note: tail is NOT incremented, so next time we'll overwrite this
    //      descriptor, which is the whole point of this algorithm
}
```

```
else
{
    LastValidIndex = tail-1;//warning: no space for spurious data buffer
}
```

The above implementation can be improved upon in a number of ways. First, rather than adding a single (large) spurious buffer, a number of smaller ones could be chained together. In this way, the amount of spurious data actually transmitted would be reduced while still maintaining a given level of protection against long latencies for the host-based software. Additionally, the implementation could be extended to handle multiple buffers at once, inserting several buffers in a row and only splitting the last one and then appending a spurious buffer or buffers. Finally, the descriptor list is a circular buffer and a real implementation would have to check `tail and tail+1 against base_address + 31 * 8.`

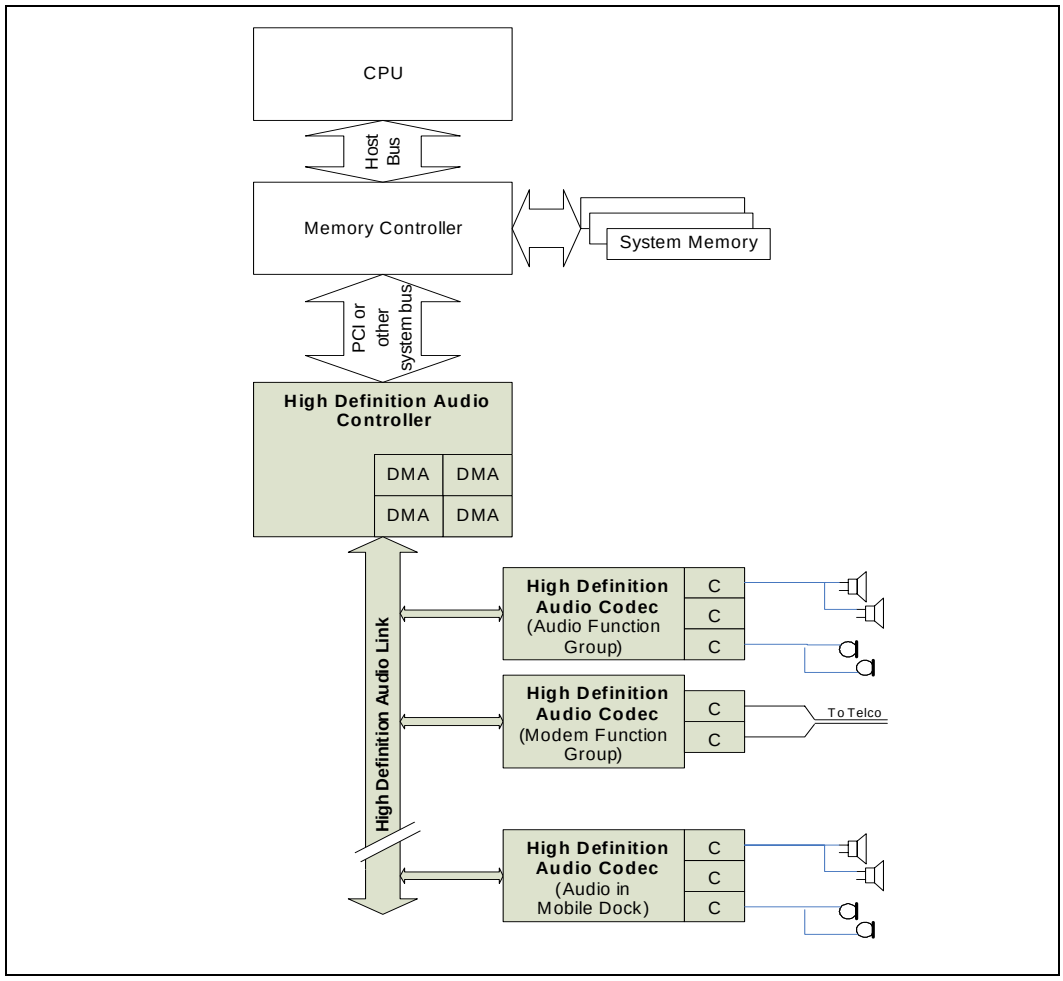
§

4 Intel® High Definition Audio Programming Model

4.1 Hardware System Overview

The purpose of this chapter is to introduce terminology specific to the Intel® High Definition Audio (Intel® HD Audio) architecture, which will be used throughout this specification, and to provide a qualitative introduction to or theory of operation for the Intel HD Audio architecture. This conceptual overview should give the reader a foundation for ease of navigating through the interface and syntactical details delivered in the balance of the specification. It is not the intent of this chapter to provide any syntactic, timing, or otherwise quantitative definitions.

Figure 4-1. Intel® High Definition Audio Hardware Overview



Controller: The Intel High Definition Audio controller is a bus mastering I/O peripheral, which is attached to system memory via PCI or other typical PC peripheral attachment host interface. It contains one or more DMA engines, each of which can be set up to transfer a single audio "stream" to memory from the codec or from memory to the codec depending on the DMA type. The controller implements all the memory mapped registers that comprise the programming interface as defined in Section 3.3 of the *Intel® High Definition Audio Specification*.

Link: The controller is physically connected to one or more codecs via the Intel High Definition Audio Link. The link conveys serialized data between the controller and the codecs. It is optimized in both bandwidth and protocol to provide a highly cost effective attach point for lost cost codecs. The link also distributes the sample rate time base, in the form of a link *bit clock (BCLK)*, which is generated by the controller and used by all codecs. The link protocol supports a variety of sample rates and sizes under a fixed data transfer rate.

Codec: One or more codecs connect to the link. A codec extracts one or more audio streams from the time multiplexed link protocol and converts them to an output stream through one or more converters (marked "C"). A converter typically converts a digital stream into an analog signal (or vise versa), but may also provide additional support functions of a modem and attach to a phone line, or it may simply de-multiplex a stream from the link and deliver it as a single (un-multiplexed) digital stream, as in the case of S/PDIF. The number and type of converters in a codec, as well as the type of jacks or connectors it supports, depend on the codec's intended function. The codec derives its sample rate clock from a clock broadcast (BCLK) on the link. Intel High Definition Audio Codecs are operated on a standardized command and control protocol as defined in Section 4.4 of the *Intel® High Definition Audio Specification.*

Acoustic Device: These devices include speakers, headsets, and microphones, and the specifications for them are outside the scope of this specification, except for discovery capabilities such as defined in Section 7.3.3.15 of the *Intel® High Definition Audio Specification*.

Packaging Alternatives: Figure 1 suggests that codecs can be packaged in a variety of ways, including integration with the controller, permanent attachment on the motherboard, modular ("add-in") attachment, or included in a separate subsystem such as a mobile docking station. In general the electrical extensibility and robustness of the link is the limiting factor in packaging options. This specification does not define or standardize packaging options beyond the standardized footprint of a codec as defined in Section 7.4.1 of the *Intel® High Definition Audio Specification*.

4.2 Theory of Operation

While the register interface is the concise description of the software interface to the Intel High Definition Audio controller, the implementation and interpretation of these various bits is not always clear from the definition of the bit(s). This Programming Model chapter is a supplement to the register definitions and provides narrative and interpretation guidance for the behaviors of the bits.

The software operation of the Intel High Definition Audio interface is divided into three categories: Codec Command and Control, Streaming Operation, and Link Initialization and Control. These three categories are described in detail in the following sections.

4.3 Controller Initialization

When the Intel High Definition Audio controller comes out of power-up reset after power-on, all controller registers will be in their power-on default state, and the link will be inactive.

4.3.0.1 Configuring a PCI or PCI Express* Interface

The first step in starting the controller is properly programming the PCI, PCI Express, or other system bus interface. Because this operation is specific to the controller implementation, the documentation for the specific controller should be followed. At the conclusion of this programming, the controller should be ready to transfer data on the system bus. For example, when using PCI, the Interrupt Line, Base Address, and other PCI Configuration space registers should be properly programmed.

4.3.1 Starting the Intel® High Definition Audio Controller

When the controller is first brought up, the CRST bit (Offset 08h, bit 0) will be 0 meaning that the controller is in reset. When the controller is in reset, the only bit which will accept writes is the CRST bit to take the controller out of CRST; all other registers will read their default values and writes will have no effect.

When a 1 is written to the CRST bit, the controller will go through the sequence of steps necessary to take itself out of reset. The link will be started, and state machines will initialize themselves. While the hardware is taking these steps, the CRST bit, if read, will still appear to be 0. When the initialization has been completed, a read of the CRST bit will return a 1 indicating that the controller is now ready to function. Therefore, after taking the controller out of reset, the software should wait until CRST is read as 1 before continuing.

Several of the Intel High Definition Audio controller registers maintain their values across resets and power transitions. These include the WAKEEN bits, the STATESTS bits, and any other registers with type “RSM” (Resume). These bits should be examined if necessary and then reset (STATESTS) or programmed appropriately (WAKEEN) for proper operation.

4.4 Codec Discovery

When the link is enabled by the assertion of CRST, the codecs will detect the de-assertion of the RESET# signal and request a status change and enumeration by the controller. As the controller hardware detects these requests, it will provide the codecs with their unique addresses and set the controller STATESTS bits to indicate that a Status Change event was detected on the appropriate SDATA_INx signals. Software can use these bits to determine the addresses of the codecs attached to the link. A 1 in a given bit position indicates that a codec at that associated address is present. For instance, a value of 05h means that there are codecs with addresses 0 and 2 attached to the link.

The software must wait 250 µs after reading CRST as a 1 before assuming that codecs have all made status change requests and have been registered by the controller. This gives codecs sufficient time to perform self-initialization.

If software wishes to get an interrupt when new codecs are attached, such as during a mobile docking event, the software can set the CIE bit in the INTCTL register to a 1 to enable controller interrupts which include the Status Change event. When the interrupt is received, the STATESTS bits can be examined to determine if a codec not previously identified has requested a status change.

4.5 Codec Command and Control

Once the attached codecs have been enumerated, commands can be sent to the codecs to determine their capabilities.

Codec Command and Control describes the mechanisms by which control information is sent to and received from the codecs. Command and Control data is low bandwidth, asynchronous data that is transmitted one command at a time on the Link. Timing is not ensured in any way, either inbound to the controller or outbound from the controller.

Codec Command and Control is handled by the controller via two key mechanisms, the Command Outbound Ring Buffer (CORB) and the Response Input Ring Buffer (RIRB).

Software is responsible for configuring the controller's CORB and RIRB via the CORB Control and RIRB Control registers.

4.5.1 Command Outbound Ring Buffer – CORB

The controller utilizes the CORB mechanism to pass commands to the codecs. The CORB is a circular buffer located in system memory that is used to pass commands (verbs) from software to codecs connected to the Intel High Definition Audio link. The controller uses DMA to fetch the outbound commands from the CORB and places them in the Command/Control bits at the start of each Link frame.

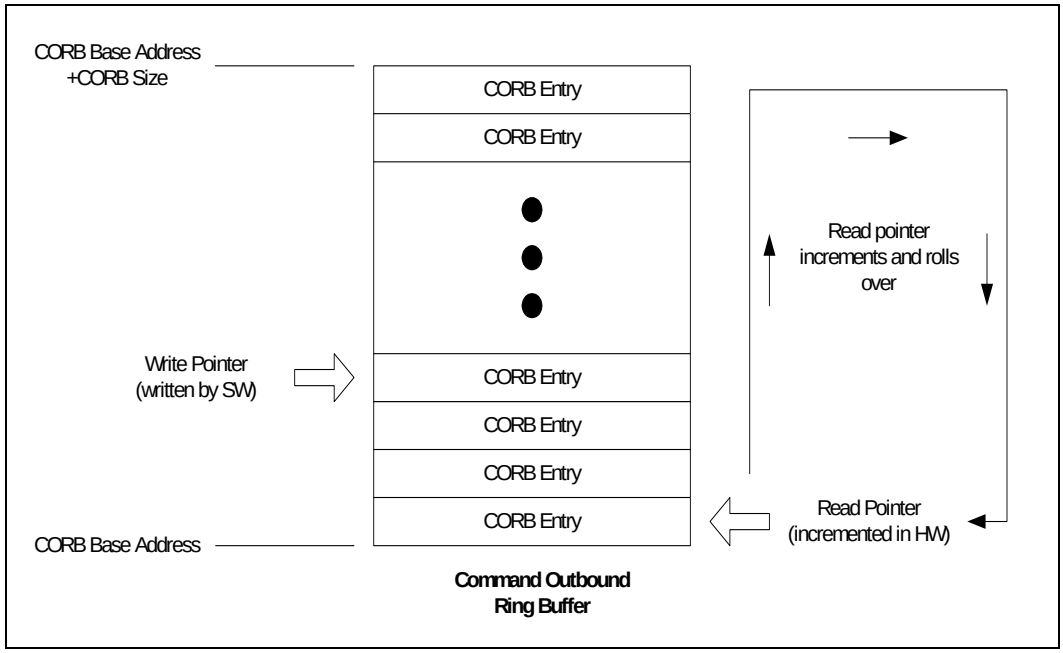
The size of the CORB is programmable to two entries (8 bytes), 16 entries (64 bytes), or 256 entries (1 KB) by using the CORBSIZE controller register. Software is responsible for choosing a CORB size based on the CORBSZCAP field and the capabilities of the system. In general, the software should choose the 256 entries option unless the system capabilities dictate a smaller memory footprint.

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.

When the CORB RUN bit is set, the DMA engine in the controller will continually compare the RP to the WP to determine if new commands are present for consumption. When the Read Pointer is not equal to the Write Pointer, the DMA engine runs until the pointers match, and the fetched commands are transmitted on the link. The DMA engine reads the commands from the CORB and sends them to the Codecs over the link.

Figure 4-2. Command Ring Buffer (CORB)



4.5.1.1 CORB Buffer Allocation

The CORB buffer in memory must be allocated to start on a 128-byte boundary and in memory configured to match the access type being used. For instance, a PCI Express controller using non-snooped accesses should allocate and access the CORB buffer in a way that maintains coherency between the processor cache and the physical memory.

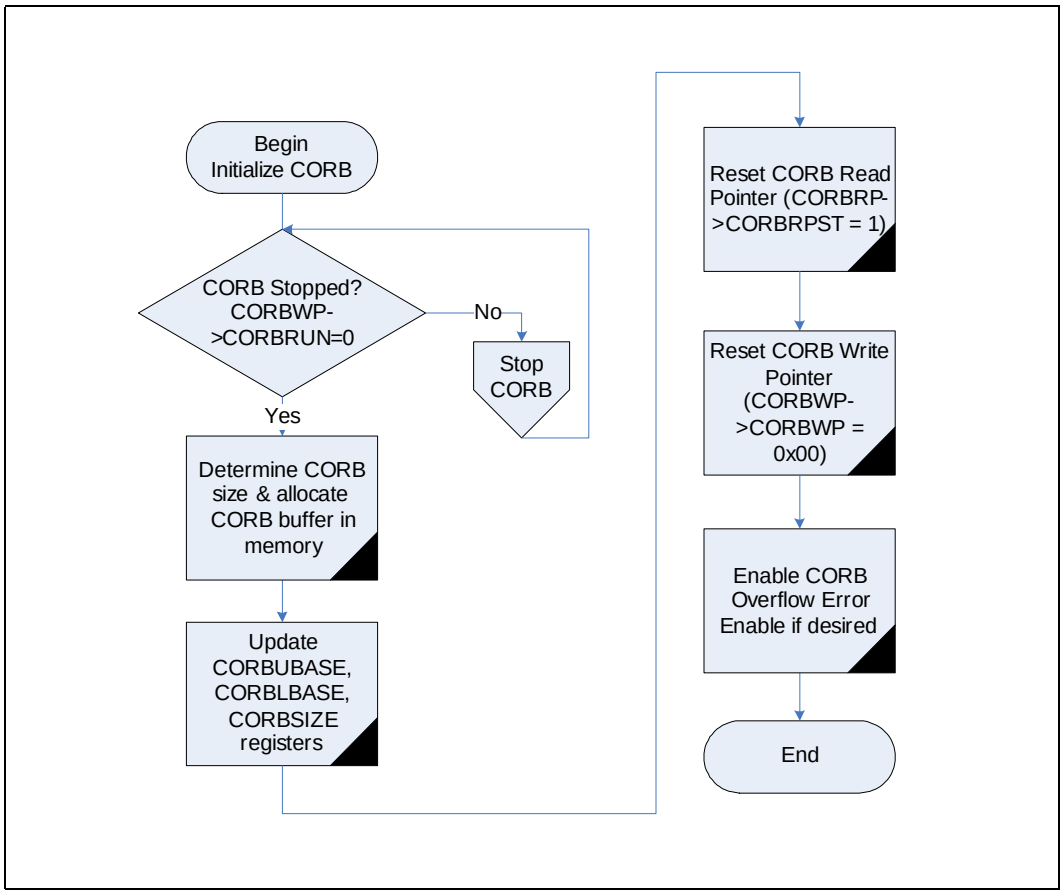
The location of the CORB is assigned by software and written to the controller's CORB Address Upper Base and Lower Base register. The lowest 7 bits of the Lower Base Address are always 0 to enforce the 128-byte alignment requirement.

4.5.1.2 CORB Entry Format

The verbs passed from the controller to the codecs are 32 bits long. Each entry in the CORB is also 32 bits long and matches the verb format. The verb format is opaque to the controller hardware and only has meaning to software and the codecs.

4.5.1.3 Initializing the CORB

Figure 4-3. CORB Initialization



To initialize the CORB, first the software must make sure that the CORB is stopped by making sure that the CORBRUN bit in the CORBCTL register is 0. The correct register size is determined using the CORBSIZE register, and the CORB memory is allocated from the appropriate heap and memory type.

The CORBBASE registers are programmed to the base of the allocated memory, and the CORBRPRST bit is used to reset the Read Pointer to 0. Software must write 0h to the Write Pointer to clear the Write Pointer. If desired, CORB error reporting may be enabled by setting the CMEIE bit. Lastly, the CORBRUN bit is set to 1 to enable CORB operation.

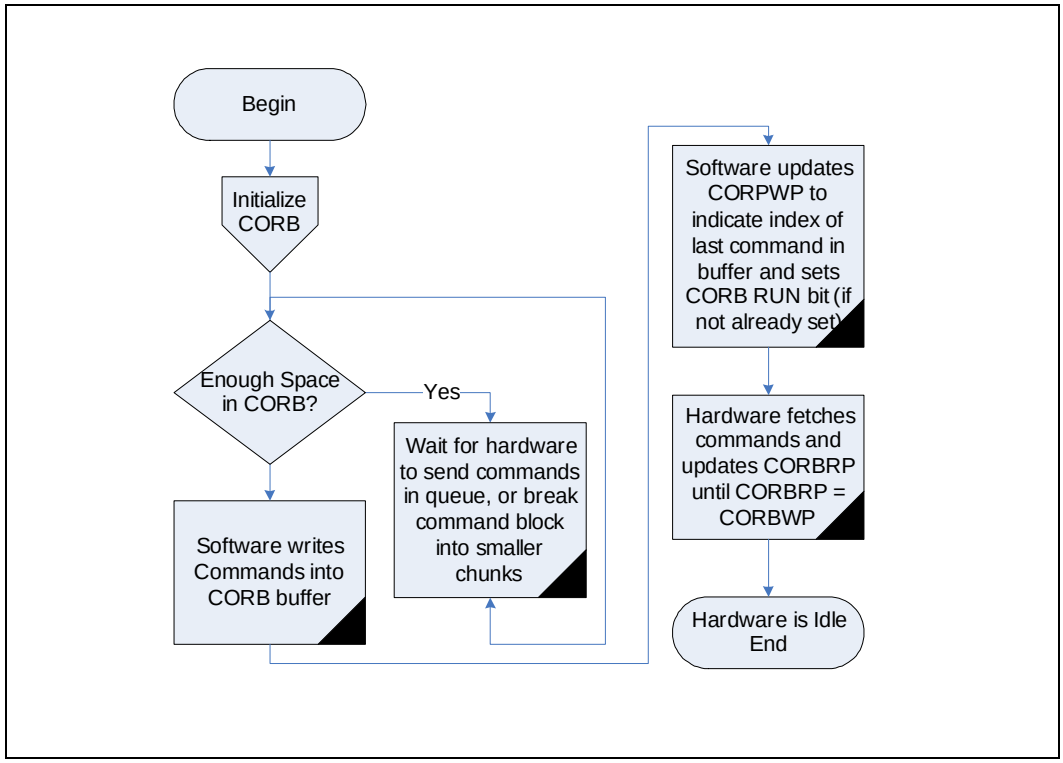
4.5.1.4 Transmitting Commands via the CORB

Transmission of commands via the CORB begins with checking to make sure there is sufficient space in the CORB. The difference between the CORBWP and CORBRP can be examined to determine the space available in the CORB. If the block of commands is larger than can fit in the CORB, it may be necessary to break the block of commands into multiple smaller blocks to send. The commands are written into the CORB starting at the location indicated by the index CORB

WP + 1, which is the first free space for a command. Note that in the case of the first block of commands, this means that the first commands will be placed at a offset of 4 bytes into the CORB buffer, as CORBWP will be 0, so CORBWP + 1 will indicate a 4-byte offset into the CORB.

CORBWP is then updated by software to reflect the index of the last command that has been written into the CORB. Hardware will then begin to transfer the commands over the link, updating CORBRP with each command fetched from memory. All commands have been sent when CORBRP is equal to CORBWP, at which point the controller will stop sending verbs until software repeats the process and sets CORBWP to a different value.

Figure 4-4. Transmitting Commands via the CORB



While the hardware is in the process of sending commands (CORBRP ? CORPWP and CORB RUN is set), software may add new commands into the CORB buffer after the index indicated by CORBWP, and then update CORBWP. Hardware must continue to send the newly added commands. Software must ensure that the newly added commands do not overflow the buffer; i.e., that no command added will overwrite commands that have not yet been sent, as indicated by the current value of the CORBRP.

4.5.1.5 Other CORB Programming Notes

If large numbers of commands are being sent to a codec, it is possible that the codec may be blocked from returning Unsolicited Responses because it is required to always respond to solicited verbs on the following frame. For this reason, it is recommended that the software occasionally insert breaks in the verbs being sent if a large block is being transmitted at one time.

If it is desired to insert breaks in the CORB, insert 00000000h commands, which are NULL commands. Since the codec will not have a solicited response to the NULL command, it provides an opportunity for the codec to respond with an unsolicited response. Note that when the software is matching responses with the commands, the NULL commands will not generate responses.

4.5.2 Response Inbound Ring Buffer - RIRB

The responses from the codecs are sent to the controller via the RIRB mechanism. The RIRB is a circular buffer located in system memory that is used to store responses from codecs connected to the Link. Responses can either be solicited (in response to a command from the controller) or unsolicited (sent by the codec to signal an event).

The size of the RIRB is programmable to two entries (16 bytes), 16 entries (128 bytes), or 256 entries (2 KB). The location of the RIRB is assigned by software and written to the controller's RIRB Address register.

The RIRB buffer in memory must be allocated on a 128-byte boundary and in memory configured to match the access type being used. For instance, a PCI-Express controller using non-snooped accesses should allocate and access the CORB buffer in a way that maintains coherency between the processor cache and the physical memory.

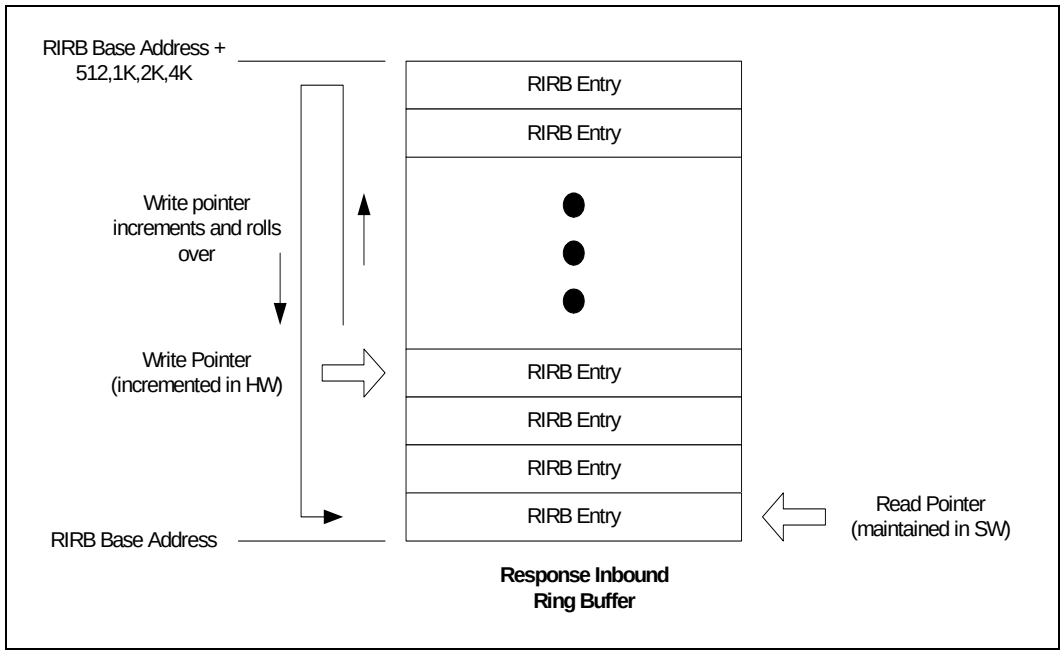
A Response can be sent to the controller from any one of the codecs, and the DMA engine in the controller writes the response to the RIRB. Solicited responses are returned by an individual codec in the subsequent frame and in the same order that the prompting commands were sent to that codec. Unsolicited responses may be injected by the codec in any frame where a solicited response is not present. The controller will write this stream of responses to the RIRB buffer. Software is responsible for separating responses from the individual codecs as well as separating unsolicited responses from solicited responses.

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).

There are two ways for the controller to notify software that the RIRB entries may be read:

- Interrupt: The controller will generate an interrupt after a programmable N number of Responses are written to the RIRB or when an empty Response slot is encountered on all SDATA_IN_X inputs, whichever occurs first. Software can then look at the Write Pointer and determine the entries added to the RIRB.
- Polling: Software may poll the hardware Write Pointer and compare it to the software maintained Read Pointer. If they are different, entries have been added to the RIRB, and software should read RIRB entries until up to the Write Pointer. Software is responsible for polling often enough to ensure that the hardware Write Pointer does not wrap around; if it does wrap, the responses will be lost.

Figure 4-5. Response Inbound Ring Buffer



4.5.2.1 RIRB Entry Format

Response passed from the Codecs to the controller is 32 bits long. Each entry in the RIRB is 64 bits long. In addition to the 32 bits representing the actual response data from the codec, the controller adds the following information to the RIRB entry:

- Codec # based on the SDATA_IN_x signals on which the Response was received
- Solicited versus unsolicited response indicator

Table 4-1. RIRB Entry Format

Offset	Length	Field	Description
0x00	4 bytes	Response	Response is the response data received from the codec
0x04	4 bytes	Resp_Ex	Response Extended contains information added to the response by the controller. Bits 3:0 is the Codec; i.e., the SDATA_INx line on which the response was received; this will correspond to the codec address. Bit 4 is a bit indicating whether the response is a solicited response (‘0’) or an unsolicited response (‘1’).

The bit definitions are as follows:

Codec
0000 = Response received on SDATA_IN_0
0001 = Response received on SDATA_IN_1

0010 = Response received on SDATA_IN_2
0011 = Response received on SDATA_IN_3
…
1111 = Response received on SDATA_IN_15

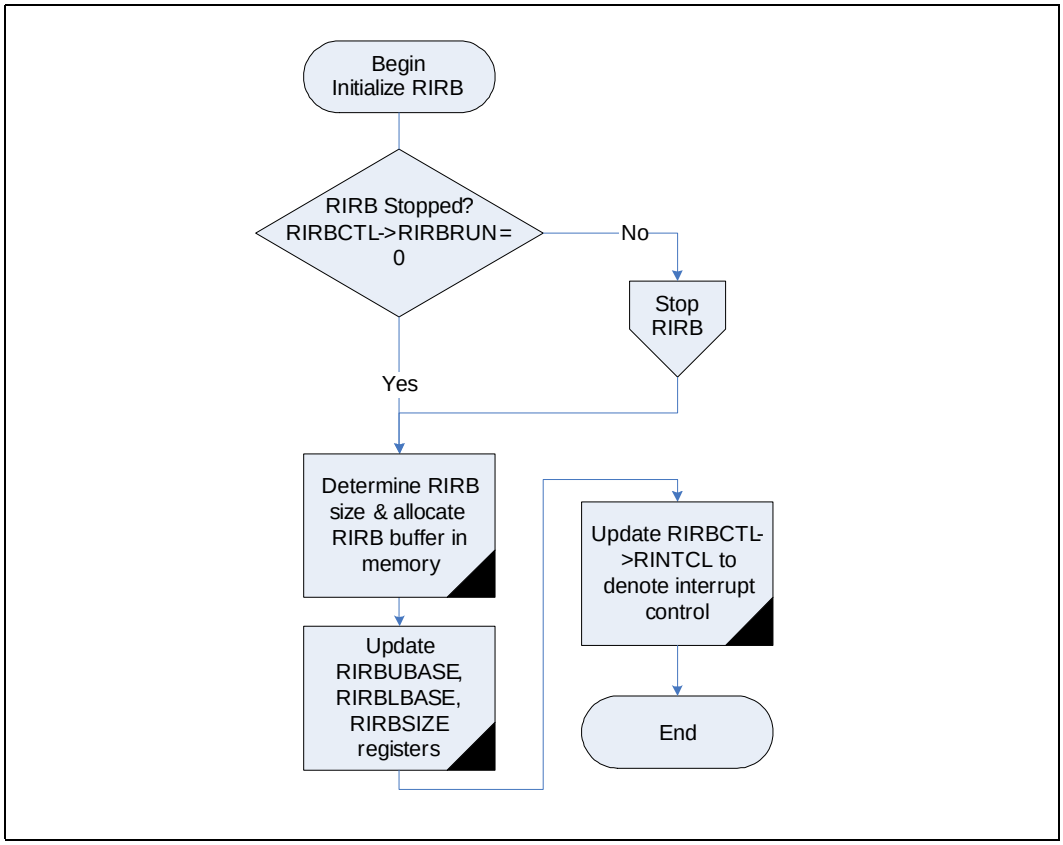
Sol/Unsol

0 = Solicited Response
1 = Unsolicited Response

4.5.2.2 Initializing the RIRB

RIRB initialization is very similarly to CORB initialization. The memory must be allocated correctly based on the RIRBSIZE register and from the heap appropriate for the system infrastructure. The RIRBUBASE, RIRBLBASE, and interrupt generation control registers are then updated appropriately.

Figure 4-6. Initializing the RIRB



4.6 Stream Management

The Intel High Definition Audio architecture uses the concept of streams and channels for organizing data for transmission on the Link. A stream is a virtual connection created between a system memory buffer(s) and the codec(s) rendering or capturing that data and which is driven by a single DMA channel through the link.

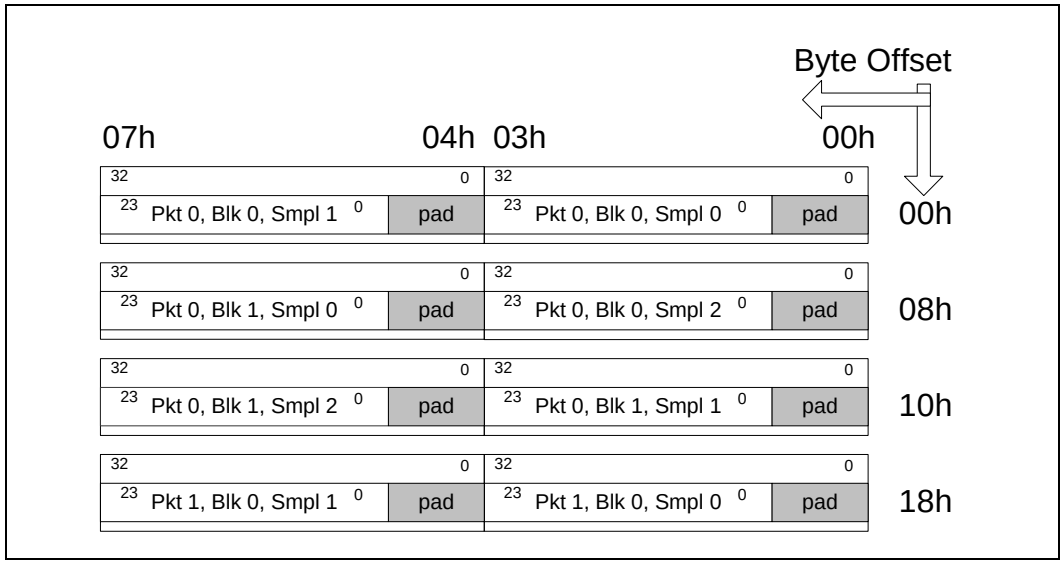
Software is responsible for creating and managing streams. Key parameters related to a stream such as stream parameters, number of channels, data format, bit depth, etc., are defined by software.

It is the responsibility of software to determine which converters (and associated hardware) on which codecs are associated with a given stream. For example, software may need to determine which jacks should be assigned to those converters.

4.6.1 Stream Data In Memory

Samples represent one channel of data to be played at one instant in time. In a 24-bit, three-channel, 96-kHz stream, one sample is 24-bits long. Samples are packed in *containers* which are 8-, 16-, or 32-bits wide; the smallest container size which will fit the sample size is used. In the case of a 24-bit sample, a 32-bit container would be used. Samples are padded with 0's at the LSB to left justify the sample within the container. Samples must be naturally aligned in memory. Samples in 16-bit containers must be Word aligned, and samples in 32-bit containers must be dword aligned.

Figure 4-7. A 24-bit, 3 Channel, 96 kHz Stream in Memory



Blocks are sets of samples to be rendered at one point in time. A block has a size of (container size * number of channels), so the 24-bit, three-channel, 96-kHz stream would have a block size of 12 bytes in memory.

The standard output Link rate is based on a 48-kHz frame time, therefore, in the case of a stream running faster than 48 kHz, multiple blocks must be transmitted at one time. Multiple blocks transmitted at the same time are *packets*. The 24-bit, three-channel, 96-kHz stream being used as an example would have a packet size of 24 bytes.

Packets are collected in memory into *buffers*, which are commonly (but are not required to be) whole pages of memory. Each individual buffer must contain an integer number of samples, but blocks and packets may be split across multiple buffers. The buffer must start on a 128-byte boundary and must contain at least one sample of data. For highest efficiency, the following guidelines should be met in buffer allocation:

- The buffer should have a length which is a multiple of 128.
- The buffer should contain at least one full packet of information.

There may be other system-dependent guidelines that can further increase memory efficiency, but buffers that do not meet at least the above guidelines may have a negative effect on system performance as bus and memory efficiency may be significantly affected.

4.6.2 Configuring and Controlling Streams

4.6.3 Starting Streams

To create a stream, the software must first determine the appropriate stream parameters, such as sample rate, bit depth, and number of channels. The controller and codec resources should be checked to make sure sufficient resources are available to support the desired stream format.

A data buffer and BDL buffer are then allocated from the proper memory pool, making sure that the proper buffer alignment and caching requirements are met. The BDL is constructed to describe the stream data buffer, including setting Interrupt on Completion bits at the points interrupts are desired. Software then allocates a Stream Descriptor (Input or Output, as appropriate), and configures the Descriptor with the stream format, BDL address, Cyclic Buffer Length, interrupt policy, and other necessary register settings.

The codec is next configured by configuring the appropriate Audio Input or Output Converters with the stream ID and format information. Other widgets in the audio path are configured so that the audio data flows to (or from) the Pin Complex Widgets, with Connection Select, Amplifiers, Processing Controls, and all other controls in the audio path set as appropriate. At this point, the codec is ready to accept data if the stream is an output stream or begins sending data on the link for an input stream.

The Stream Descriptor's RUN bit is then set to 1 to start the DMA engine and begin the transfer of data to or from the link.

4.6.4 Stopping Streams

To stop a stream, the software writes a 0 to the RUN bit in the Stream Descriptor. The RUN bit will not immediately transition to a 0. Rather, the DMA engine will continue receiving or transmitting data normally for the rest of the current frame but will stop receiving or transmitting data at the beginning of the next frame. When the DMA transfer has stopped and the hardware has idled, the RUN bit will then be read as 0. The run bit should transition from a 1 to a 0 within 40 µs.

4.6.5 Resuming Streams

If a stream which was previously running has been stopped, it can be restarted by setting the RUN bit back to 1. If the stream has been recently stopped, the RUN bit must be checked to make sure that it has transition back to a 0 to indicate that the hardware is ready to restart. When the RUN bit is again set to 1, the DMA engine will then restart at the point it left off.

4.6.6 Stream Steady State Operation

Once the stream has been started, the hardware will continue fetching data from the Cyclic Buffer described by the Buffer Descriptor List. For an output stream, software is responsible for making sure that valid data is present in the buffers before it is fetched by the hardware. For an input stream, data must be removed from the buffers before being overwritten by hardware. When the hardware has reached the end of the Cyclic Buffer, it will automatically wrap back around to the beginning of the buffer and continue to process the stream data until the stream is stopped by the software by clearing the RUN bit.

Software can either use interrupts at the end of selected buffers by setting the IOC bit in the BDL entry or can poll the stream position to determine when to process the stream data.

If interrupts are being used, some care must be taken to access the stream Status and Control registers in a safe manner. A recommended policy is that the Interrupt Service Routine only use byte access to read or write the Status register to clear the status bits. The ISR should not attempt to write to the stream Control register, as there may be synchronization issues between the ISR and the non-ISR code both trying to perform Read-Modify-Write cycles on the register.

After the RUN bit has been set, the buffer described by the BDL should not be changed by the software. The hardware may pre-fetch and/or cache an arbitrary number of BDL Entries from the list, so there is no way to ensure when or if any changes to the BDL list would be visible to the hardware. Even when the RUN bit has been cleared to pause the stream after it has been running, the hardware may still have pre-fetched descriptors that will not be flushed when the stream is restarted. Therefore, the software should only modify the BDL before the RUN bit has been set for the first time after a Stream Reset.

4.6.7 Synchronization

There are three different domains in which synchronization of streams is desired. They are multiple streams on different systems, multiple streams on different controllers in the same system, and multiple streams on the same controller.

The *Intel® High Definition Audio Specification* does not provide any mechanisms to aid in synchronizing streams on two different systems but does provide mechanisms to synchronize streams within a system. The wall clock can be used to synchronize between two separate controllers which do not share a common clock, and the stream start synchronization can be used to synchronize exactly two streams on the same controller.

4.6.7.1 Controller-to-Controller Synchronization

The *Intel® High Definition Audio Specification* requires that the controller provide a 'wall' clock, implemented in the Global Synchronization and Control Register, which is a monotonically increasing 32 bit counter. Whenever bit clock is running, this clock is also running. This time base may be used to account for drift between two different audio subsystems by performing micro-sample rate conversion operations on the audio data to keep the drift between streams running on the independent controllers to within a specified error.

4.6.7.2 Stream-to-Stream Start Synchronization

Using the SSYNC bits in combination with the stream RUN bits, multiple input and streams can be synchronized in time.

When the hardware is initialized, all the relevant stream descriptor's RUN bits are cleared. Also, the relevant SSYNC bits are cleared as well. To synchronously start a set of streams, software will set the relevant SSYNC bits to 1 to indicate the set of streams to be synchronously started and then set the stream's RUN bit to cause the DMA engine to fetch data. While the SSYNC bits are set, though, data will not be sent on to the link, essentially leaving the stream in a "pause" state.

Software then must wait until each output stream's FIFORDY bit is set, indicating that the DMA engines have fetched enough data to start the stream in the case of an output stream, or fetched enough buffer descriptors to have a place to put the incoming data in the case of an input stream. This ensures that all output streams will have sufficient data ready, or a place to put it, when the streams are started. Note that for input streams, FIFORDY is set to 1 as a function of whether one or more Input-Descriptors are available in the Input Stream buffer, independent of the descriptor's length value. If the first descriptor-length value is very small, it increases the likelihood that overrun condition will occur.

When all relevant FIFORDY bits are set, software clears the relevant SSYNC bits using a single write to the Stream Synchronization register causing the streams to begin flowing. All output streams will transmit their first sample on the link frame following the de-assertion of their SSYNC bit, and input streams will capture data from the link frame following the de-assertion of their SSYNC bit.

4.6.7.3 Stream-to-Stream Stop Synchronization

- The sequence starts while relevant streams are actively receiving (input) or transmitting (output). Their respective RUN bits are set to 1 while SSYNC bits are cleared to 0.
- To begin the synchronized stop, software writes a 1 to the relevant SSYNC bits. As a result of the SSYNC bits being set, the controller will stop receiving (input) or transmitting (output) in the beginning of the next frame for the relevant streams. Software may then clear each individual streams' RUN bits to 0 to stop the stream's DMA engine.
- Once software detects that all relevant RUN bits are cleared to 0, it can clear the SSYNC bits to a 0 to return the controller to the initial ready state. Software can then restart the streams using the same sequence described in Section 4.5.7.2, or it may start them individually without using the SSYNC bits.

4.6.8 Power Management

4.6.8.1 Power State Transitions

When the controller or codecs are transitioned to a D0 state from a D3 state, it is possible that the hardware may have had power removed, and, therefore, software must not assume that registers retain values previously programmed, with the exception of controller registers marked “RSM.” Software should therefore restore all codec registers on a transition to D0 from D3.

4.6.8.2 Power Optimization

While there is no requirement that software perform power optimizations, in many environments power savings are desired. Software may optimize at three levels. The easiest power management level is using the CRST bit in the controller to power up or down the entire controller and link at once. This method is obviously not always suitable, as any activity on the link to any codec will prevent software from entering this state. This state also takes the longest to resume from, as the entire subsystem including the controller and codec must be reinitialized before a stream can be started.

Software may also control power at the Function Group level by using the Power State Control at the Function Group level. This method is generally sufficient for most systems, especially where there are multiple function groups on the link, such as audio and modem. In this case, function groups not in use may be shut down without affecting the operation of other function groups.

To achieve optimal power conservation, software may use widget level power controls to shut down widgets not in use. In the example case of an audio function group, software may be able to shut down amplifiers and other power consuming components in the codecs without affecting active streams using other paths in the codec. The actual level of power savings may vary considerably, as different codec hardware implementations may make different power usage versus complexity tradeoffs.

4.6.9 Codec Wake

4.6.9.1 Codec Wake from System S0, Controller D0

When the system is in S0 and the controller is active, a codec will use an unsolicited response to indicate to software that it requires attention. For instance, a Modem Function Group may use a vendor defined unsolicited response on “Ring Indicate” to request attention from software. Software can then use the source of the unsolicited response and the Unsolicited Response Tag to know which Function Group requires service. The Audio Function Group definition does not currently define an unsolicited response to indicate a “Wake” situation; such a mechanism would be vendor defined.

4.6.9.2 Codec Wake from System S0, Controller D3

When the controller is in D3 (CRST is set to a 1), the link will not be running, and codecs will use a power state change request on the link (see Section 5.6) to indicate to the controller that they require service. Software can control which codecs may wake the system by setting the WAKEEN bits appropriately. On a wake event, software reads the STATESTS register (before taking the controller out of reset) to determine which codec(s) have requested a power state change. Software must then further query the function groups in the codec to determine which function group requested the wake service.

4.6.9.3 Codec Wake from System S3

When the system is in S3 at the time a codec issues a power state change request and the associated bit in the WAKEEN registers is set, the hardware will route the request to the system PME logic and the ACPI subsystem (or other power management mechanism as implemented in the system). This will cause the system to transition to S0. Software can control which codecs may wake the system by setting the WAKEEN bits appropriately. When software regains control, it can examine the STATESTS bits to determine which codec(s) have requested power state transitions and handle as necessary. Software must then further query the function groups in the codec to determine which function group requested the wake service.

4.6.9.4 Checking Wake Status on Resume

In the link bring up sequence, there is a time during the codec initialization when codecs may neither generate a power state change request on the link nor generate unsolicited responses. If it is vital that a wake event not be missed, then software should check the function group on any controller or codec transition from D3 to D0 to determine whether the function group has requested a wake. In most cases, this is not necessary as with a modem where the next ring will cause the necessary notification if the first notification is lost during the transition from D3 to D0. The Function Group control to check will be vendor unique and is not defined for the Audio Function Group.

§

5 Codec Features and Requirements

This chapter defines the architectural and physical requirements for Intel High Definition Audio codecs, including:

- Register definitions for codec parameters and controls
- Command and control Verb definitions
- Physical and mechanical (packaging) requirements
- Power Management requirements

Note that the term “codec” is often used herein to describe all devices (including modems) that connect to an Intel High Definition Audio Controller via the Intel High Definition Audio Link; the term is not limited strictly to audio codecs.

5.1 Codec Architecture

The *Intel® High Definition Audio Specification* defines a complete codec architecture that is fully discoverable and configurable so as to allow a software driver to control all typical operations of any codec. While this architectural objective is immediately intended for audio codecs, it is intended that such a standard software driver model not be precluded for modems and other codec types (e.g., HDMI, etc.). This goal of the architecture does not imply a limitation on product differentiation or innovative use of technology. It does not restrict the actual implementation of a given function but rather defines how that function is discovered and controlled by the software function driver.

5.1.1 Modular Architecture

The Intel High Definition Audio Codec Architecture provides for the construction and description of various codec functions from a defined set of parameterized modules (or building blocks) and collections thereof. Each such module and each collection of modules becomes a uniquely addressable *node*, each parameterized with a set of read-only capabilities or *parameters*, and a set of read-write commands or *controls* through which that specific module is connected, configured, and operated.

The codec Architecture organizes these nodes in a hierarchical or tree structure starting with a single *root node* in each physical codec attached to the Link. The root node provides the “pointers” to discover the one or more *function group(s)* which comprises all codecs. A function group is a collection of directed purpose modules (each of which is itself an addressable node) all focused to a single application/purpose, and that is controlled by a single software function driver; for example, an audio function group (AFG) or a modem function group.

Each of these directed purpose modules within a function group is referred to as a *widget*, such as an I/O Pin Widget or a D/A Converter Widget. A single function group may contain multiple instances of certain widget types (such as multiple Pin Widgets), enabling the concurrent operation of several channels. Furthermore, each widget node contains a configuration parameter which

identifies it as being “stereo” (two concurrent channels) or “mono” (single channel). Widgets within a single functional unit have a discoverable and configurable set of interconnection possibilities.

Figure 5-1. Module-Based Codec Architecture

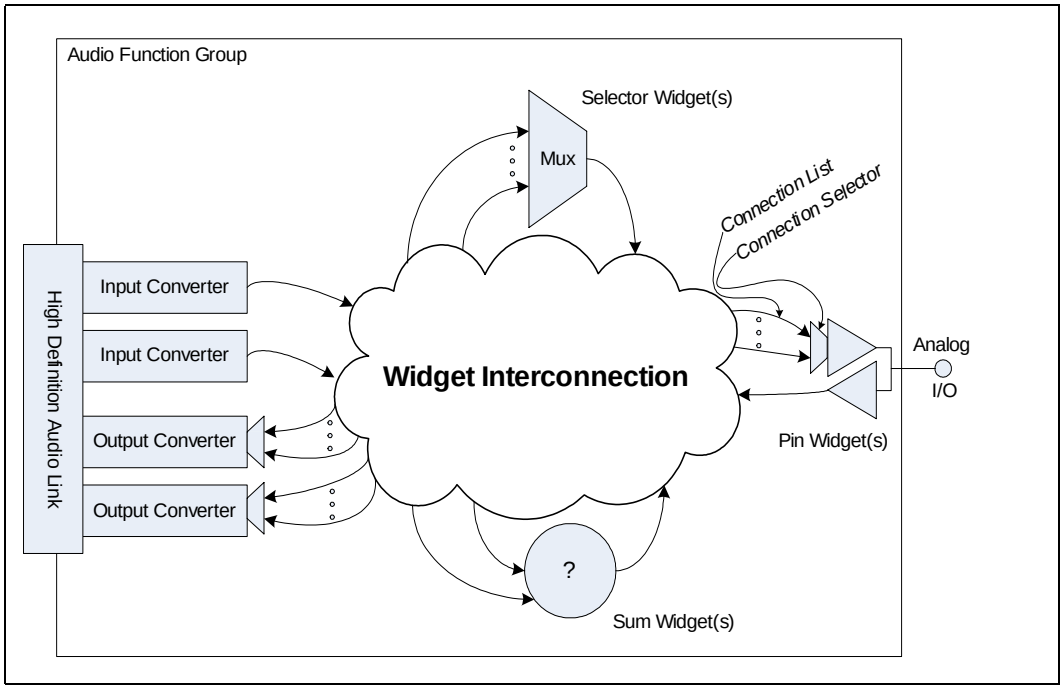


Figure 5-1 illustrates an Audio Function Group, showing some of the defined widgets and the concept of their interconnection. Some of these widgets have a digital side that is connected to the Intel High Definition Audio Link interface, in common with all other such widgets from all other function groups within this physical codec. Others of these widgets have a connection directly to the codec’s I/O pins. The remaining interconnections between widgets occur on-chip, and within the scope of a single function group.

Each widget drives its output to various points within the function group as determined by design (shown as an interconnect cloud in Figure 5-1). Potential inputs to a widget are specified by a connection list (configuration register) for each widget and a connection selector (command register) which is set to define which of the possible inputs is selected for use at a given moment (seeSection 5.1.2). The exact number of possible inputs to each widget is determined by design; some widgets may have only one fixed input while others may provide for input selection among several alternatives. Note that widgets that utilize only one input at a time (e.g., Pin Widget) have an implicit 1-of-n selector at their inputs if they are capable of being connected to more than one source, as shown in the Pin Widget example of Figure 5-1.

5.1.2 Node Addressing

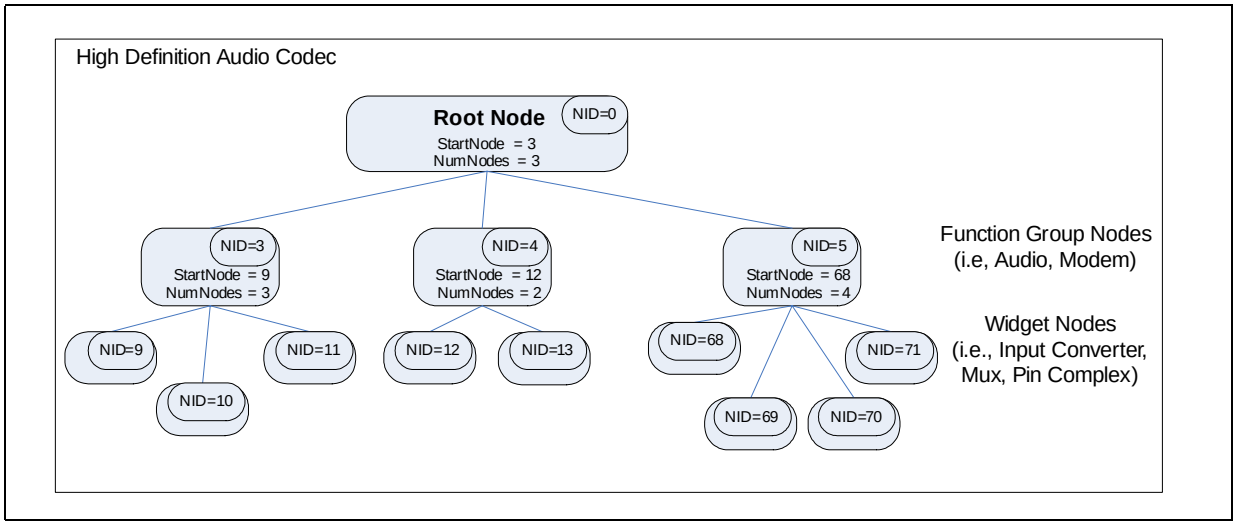
Each node in the codec architecture (i.e., root node, function group nodes, and widget nodes) may be uniquely addressed to access its various parameters and controls for purposes of enumeration and run-time control. Each physical codec connected to the Link is assigned a unique *codec address* (CAd) at initialization, which is thereafter used as part of this addressing mechanism.

Within a codec, nodes are organized in a three-level hierarchy: root node, function group nodes, and widget nodes. However, the addressing scheme for all of these nodes is completely flat, and is defined to simplify the software discovery of the codec's capabilities (seeFigure 5-2). Each node within a codec has a unique *node ID* (NID). The concatenation of CAd and NID provide a unique address allowing commands to reference a single specific node within the particular Intel High Definition Audio subsystem.

The root node is the top level node of any codec and is always addressed as node zero (NID = 0) at the codec address (CAd) assigned to this codec. The root node contains device level information including the number of function groups in this codec and the NID of the first function group.

Each codec contains at least one function group. All function groups within the codec are identified with sequential NIDs; the root node specifies the beginning NID in this sequential series. Each function group contains several parameters, including the starting NID of its particular collection of widgets; all widgets associated with this function group must be numbered sequentially starting at the reported NID.

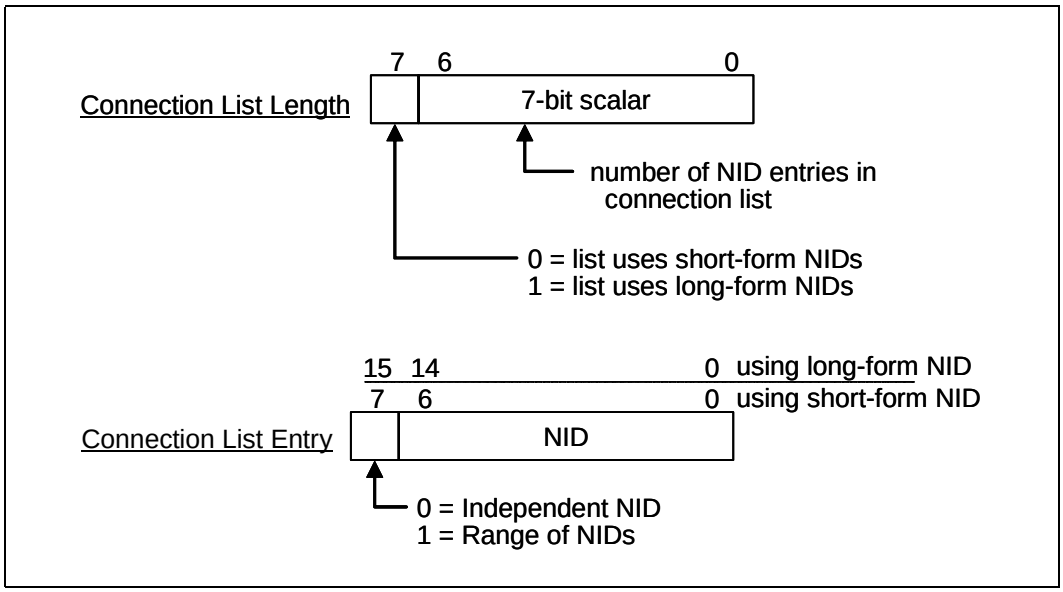
Figure 5-2. Codec Module Addressing Scheme



Nominally, a NID is represented in a "short-form" as a 7-bit integer. However, in very large codecs (e.g., containing more than 127 nodes), NIDs may be represented in a "long-form" as a 15-bit integer, as shown in Figure 5-3.

Each widget with inputs that are driven from the outputs of other widgets must have a Connection_List (parameter), or a list of NIDs that can be used as inputs. The number of entries (NIDs) in this list is specified in a Connect_List_Length register (parameter) as a 7-bit integer. The high order bit in the Connection List Length register indicates whether NIDs in this particular list will be represented in long-form or short-form (see Figure 5-3).

Figure 5-3. Connection Lists



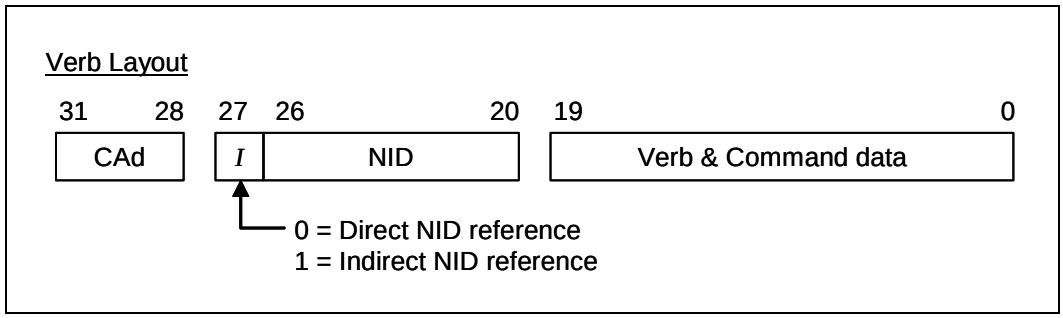
Each entry in the connection list is one NID; it may be an independent NID, indicating a single node, or may be part of a 2-tuple of NIDs delineating a continuous range of nodes. The high order bit of each entry indicates how it is to be interpreted (see Figure 5-3). If the range indicator is set, that list entry forms a range with the previous list entry; i.e., if the range bit were set on the third list entry, then the second and third entries form a range, and the first entry is an independent NID. The range indicator may not be set in the first entry of a connection list nor may it be set on any two sequential list entries. The connection list thus provides a set of optional inputs to the node. A connection selector (control) allows run-time control over which input is used.

Identifying a range of nodes allows for a reduction of enumeration data space in some cases. For example, in specifying the possible connection of four input pins to a single A/D converter, the connectivity list may have four entries, each identifying one of the input pins, or, if the NIDs of the input pins were sequential, the connection list could be reduced in size to two entries specifying a range comprising the same four input pins.

Based on this addressing scheme, 12 of the 32 bits of a verb are used to address a specific node in the system, as shown in Figure 5-4. The high order four bits in any verb are the codec address and specify a physical codec on the Link. Bits 26-20 (7 bits) are the node ID. Bit 27 allows for an indirect addressing mechanism (to be specified) for codecs that have more than 127 nodes to address and, therefore, use the long form (15-bits) of node addressing. The low order 19 bits of a Verb contain the actual command and payload data.

A CAd of 15 (all ones) is reserved for broadcasting to all codecs. A codec must respond to this CAd = 15 as well as to the one it was assigned during the link initialization sequence. However, there are not currently any commands or verbs defined that are intended for use in a broadcast manner.

Figure 5-4. Verb Addressing Fields



5.1.3 Widget Interconnection Rules

All connected widgets must adhere to the following connection rules:

1. Widgets are exclusive to a single function group; they may never be shared between function groups.
2. Connection_List_Length = 1 means there is only one (hard-wired) input possible and, therefore, there is no Connection_Selector field. The actual connection is read from the Connection List as usual.
3. When a stereo widget is an input to another stereo widget, the L-channel of the source always drives the L-channel of the sink: R-channel to R-channel.
4. When a mono widget is an input to a stereo widget, the source always drives both L- and R-channels of the sink.
5. When a stereo widget is an input to a mono widget, only the L-channel of the source drives the sink. There is one exception to this rule, which allows a mono Mixer Widget to mix the L- and R-channels of a single stereo widget into a mono channel. This exception occurs only when the sink widget is:

- A Mixer Widget, and
- Is identified as “mono,” and
- Contains exactly one entry in its connection list, and
- The single widget identified in its connection list (source) is “stereo.”

In this case, this mixer is required to have two inputs, the first being driven by the L-channel of the stereo source and the second by the R-channel.

6. Widgets may only interconnect within a function group. Widgets may not contain in their Connection Lists the addresses of widgets that are identified as being part of a different Function Group. Links between different function groups must be reported as external connections (Pin Complexes, in the example of an Audio Function Group) as if each function group was implemented in its own chip.

5.2 Qualitative Node Definition

This section provides a qualitative description of the standard modules or nodes that are defined in the Intel High Definition Audio Codec Architecture and which may be used in various combinations to construct codec functions. Each of these modules or nodes is formally defined by its own set of parameters (capabilities) and controls (command and status registers): however, since some parameters and controls are formatted to be used with multiple nodes types, it is easier to first understand nodes at the qualitative level provided in this section. Thereafter, the exact data type, layout, and semantics of each parameter and control are defined in Section 5.2.3.7. Finally, this is followed (Section 5.3.3.15) with tables specifying exactly which parameters and controls must be supported by which node types.

In the Intel High Definition Audio Architecture, enumeration or discovery proceeds in a top down manner. After controller initialization, or upon a hot-plug event, the controller provides a list of the connected codecs each of which has been assigned a codec address (CAd). Within each codec, the root node (NID = 0) contains parameters specifying the function groups contained within the codec. Each function group, in turn, contains parameters specifying the widgets contained within the function group, and each widget contains parameters and controls used for its discovery, configuration, and operation.

The discovery and arrangement of widget types differs for each type of function group. It is the responsibility of the function driver to query the parameters of each of its widgets, to build a topological graph of widget connection options, and to configure and operate the overall function group.

5.2.1 Root Node

A codec's root node is always at NID = 0 and contains device level information serving as the starting point in the enumeration of the codec. The root node contains the following parameters:

- Vendor ID
- Device ID
- Revision ID
- Number of Function Groups within the codec including NID of the first one.

The root node has no accessible controls or commands.

5.2.2 Function Groups

A function group is a collection of widgets which are all common to a single application/purpose and which are controlled by a single “Function Driver.” A codec contains one or more function group(s). While it is possible for a codec to contain more than one function group of a given type, this would not be typical. Currently defined function groups are:

- Audio Function Group
- Vendor Specific Modem Function Group
- HDMI Function Group (to be defined at a future time)

In addition to these function groups defined in this specification, it is possible for vendors to define other, proprietary function groups as part of a codec. Proprietary function group(s) will likely require proprietary parameters and controls and the verbs to access them. All of these (if used) must be defined within the Intel High Definition Audio codec architectural model. Any vendor specific access method must strictly adhere to the defined meanings of verb address fields shown in Figure 5-4 (“CAd” and “NID” fields); vendor defined verbs may use only those verb encodings specified for such use (Section 5.3.5) together with the related payloads, all of which is contained in verb bits 19:0.

5.2.2.1 Audio Function Group

The Audio Function Group (AFG) contains the audio functions in the codec and is enumerated and controlled by the audio function driver. An AFG may be designed/configured to support an arbitrary number of concurrent audio channels, both input and output. An AFG is a collection of zero or more of each of the following types of widgets (note that all widget types or all combinations of nodes may not be supported by the standard software driver - consult your driver provider for details):

- Audio Output Converter
- Audio Input Converter
- Pin Complex
- Mixer (Summing Amp)
- 1-of-N Input Selector (multiplexer)

The parameters contained within an AFG node are generally:

- Subsystem ID
- Total number of widgets to be enumerated within the AFG, including NID of the first one
- Optional default converter and amplifier parameters that apply to all AFG widgets unless specifically over-ridden
- Number of GPIO pins on this codec that are assigned to the audio function
- Power management and other audio capabilities and parameters

The controls supported by an AFG node include reset and power management controls and read/write access to the GPIO pins.

5.2.2.2 Vendor Specific Modem Function Group

The Vendor Specific Modem Function Group (VSM-FG) contains the modem functions in the codec and is enumerated and controlled by a modem function driver. A VSM-FG is nominally designed/configured to support a single modem. It is required to provide certain standard parameters and controls sufficient for positive function group identification and a few basic controls including information to load the appropriate function driver. Beyond this, implementation of this function group is vendor defined.

The following specification-defined Verb encodings must be supported by all VSM-FG nodes:

- Get Parameter; encoding F00h, Section 5.3.3.1
- Get/Set Power State; encoding F05h/705h, Section 5.3.3.9
- Get/Set Unsolicited Response; encoding F08h/708h, Section 5.3.3.14

- Get SubSystem ID; encoding F20h, Section 5.3.3.30
- Function RESET; encoding 7FFh, Section 5.3.3.28

All remaining Verb encodings have no standard definition for the VSM-FG node and may be arbitrarily defined by the modem vendor. However, it is strongly recommended that the vendor attempt to use specification-defined Verbs where possible in order to avoid confusing duplications in Verb definitions.

In addition, the following specification-defined parameters must be supported by all VSM-FG nodes:

- Subordinate Node Count; parameter # 04h, Section 5.3.4.3
- Function Group Type; parameter # 05h, Section 5.3.4.4
- Supported Power States; parameter # 0Fh, Section 5.3.4.12

All remaining parameter numbers are reserved. If other parameters are desired, they must be accessed through a Verb other than “Get Parameters” (F00h).

Whether a VSM-FG implementation contains any vendor defined widgets, or is wholly implemented as a single node, is strictly a vendor choice. A vendor may also elect to define each of several 16-bit registers as “virtual widgets nodes” and thus utilize the node address space as register addresses within this particular function group. Alternately, the vendor may map various Verb encodings to registers, or devise other methods of accessing vendor specific control registers. Any vendor specific access method must strictly adhere to the defined meanings of Verb address fields shown inFigure 5-4 (“CAd” and “NID” fields); vendor defined Verbs may use only those verb encodings specified for such use (Section 5.3.5), together with the related payloads, all of which is contained in Verb bits 19:0.

5.2.3 Widgets

A widget is the smallest enumerable and addressable module within a function group. A single function group may contain several instances of certain widgets. For each widget, there is defined a set of standard parameters (capabilities) and controls (command and status registers). Again, each widget is formally defined by its own set of parameters (capabilities) and controls (command and status registers); however, since some parameters and controls are formatted to be used with multiple different widget types, it is easier to first understand widgets at the qualitative level provided in this section. Thereafter, the exact data type, layout, and semantics of each parameter and control are defined in Section 5.2.3.7. Currently defined widgets are:

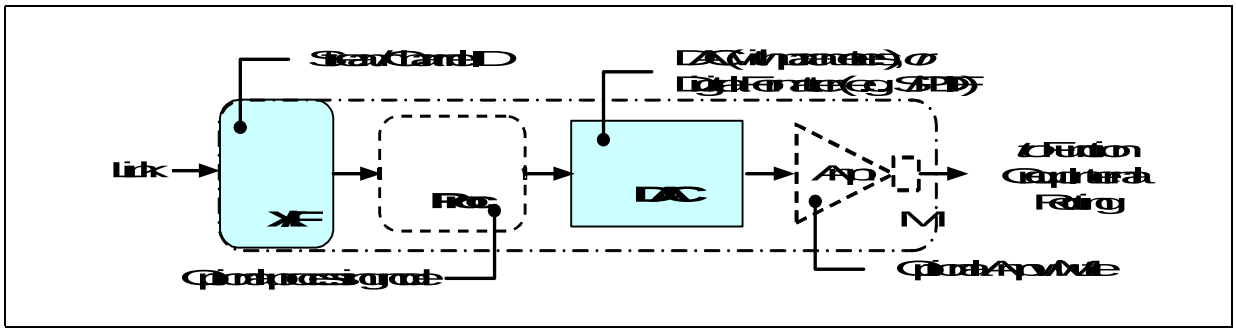
- Audio Output Converter Widget
- Audio Input Converter Widget
- Pin Widget
- Mixer (Summing Amp) Widget
- Selector (Multiplexer) Widget
- Power Widget

In addition to these standard widgets defined in this specification, it is possible for vendors to define other proprietary widgets for use in any proprietary function groups they define.

5.2.3.1 Audio Output Converter Widget

The Audio Output Converter Widget is primarily a DAC for analog converters or a digital sample formatter (e.g., for S/PDIF) for digital converters. Its input is always connected to the Intel High Definition Audio Link interface in the codec, and its output will be available in the connection list of other widget(s), such as a Pin Widget. This widget may contain an optional output amplifier, or a processing node, as defined by its parameters (Figure 5-5). Its parameters also provide information on the capabilities of the DAC and whether this is a mono or stereo (1- or 2-channel) converter. In order to save parameter space in an Audio Function Group incorporating several Audio Output Converter Widgets, the function group node may optionally contain defaults for many of the DAC and amplifier parameters; however, these defaults may be over-ridden with local parameters if necessary.

Figure 5-5. Audio Output Converter Widget

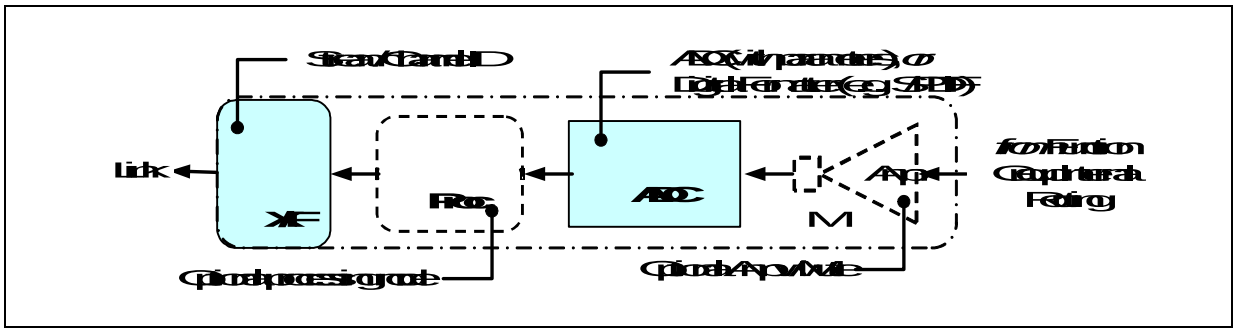


The Audio Output Converter Widget provides controls to access all its parametric configuration state, as well as to bind a stream and channel(s) on the Link to this converter. In the case of a 2-channel converter, only the “left” channel is specified; the “right” channel will automatically become the next larger channel number within the specified stream (see Section 5.3.3.11).

5.2.3.2 Audio Input Converter Widget

The Audio Input Converter Widget is composed primarily of an ADC for analog converters or a digital sample formatter (e.g., for S/PDIF) for digital converters. Its output is always connected to the Link interface in the codec, and its input will be selected from its own input connection list. This widget may contain an optional input amplifier, or a processing node, as defined by its parameters (Figure 5-6). Its parameters also provide information on the capabilities of the ADC, and whether this is a mono or stereo (1- or 2-channel) converter. In order to save parameter space in an Audio Function Group incorporating several Audio Input Converter Widgets, the function group node may optionally contain defaults for many of the ADC and amplifier parameters; however, these defaults may be over-ridden with local parameters if necessary.

Figure 5-6. Audio Input Converter Widget



The Audio Input Converter Widget provides controls to access all its parametric configuration state, as well as to bind a stream and channel(s) on the Link to this converter. In the case of a 2-channel converter, only the “left” channel is specified; the “right” channel will automatically become the next larger channel number within the specified stream (see Section 5.3.3.11.

5.2.3.3 Pin Widget

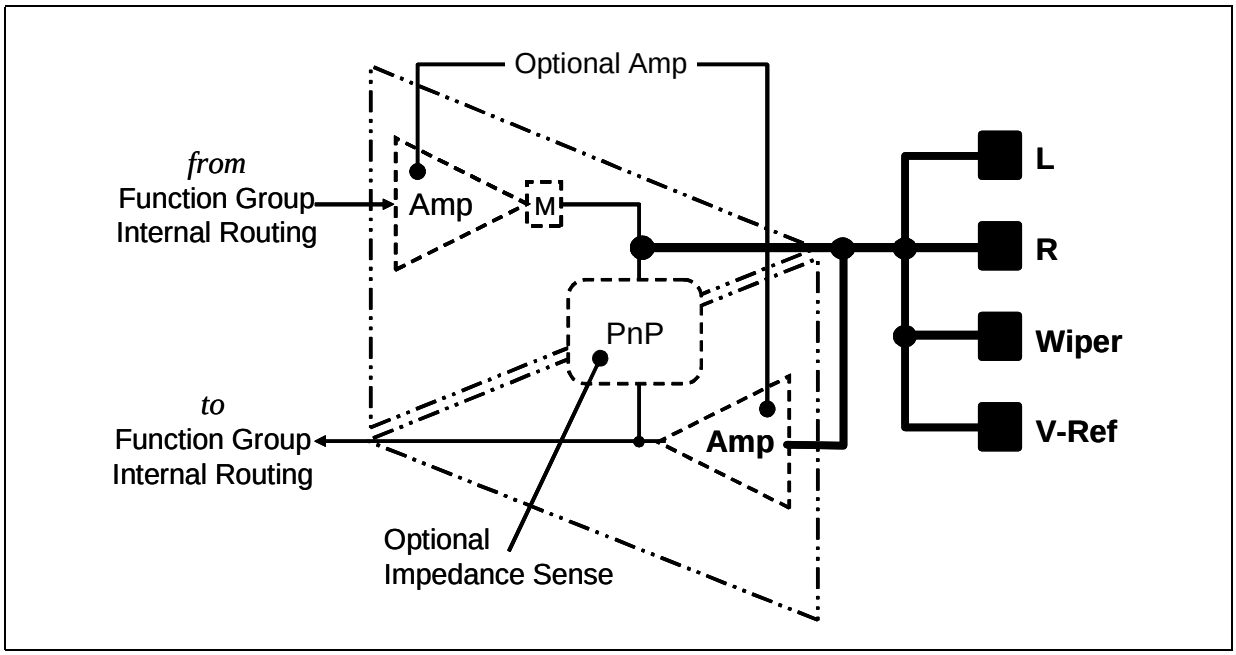
The Pin Widget provides the external (analog or digital) connection for the audio and other function groups. A Pin Widget further includes those signals directly related to the external connections, such as jack sense and Vref control signals (Figure 5-7). However, GPIO pins are not identified as part of a Pin Widget but are a resource of the function group (see Sections 1.2.2.1 and The Pin Widget's capabilities are highly parameterized defining optional support for:

- Input, output (or both), including the presence and capability of amplifier(s)
- Stereo or mono (1- or 2-channel)
- Plug (presence) detection
- Attached device impedance sensing
- VRef bias for microphone support

Every Pin Widget must contain a Configuration Default Register as defined inSection 5.3.3.31 In order to save parameter space in a function group incorporating several Pin Widgets, the function group node may optionally contain defaults for the amplifier parameters; however, these defaults may be over-ridden with local parameters if necessary.

The channel played on the external output pin (input to the Pin Widget) will be selected from its own input connection list; the channel on the external input pin will be available in the connection list of other widget(s), such as an Audio Input Converter Widget.

Figure 5-7. Pin Widget

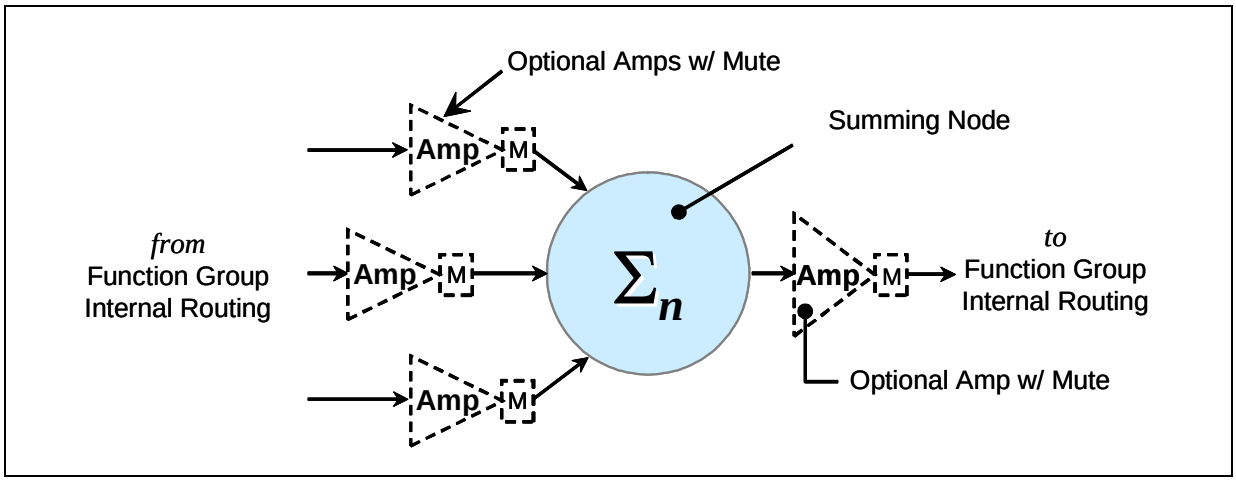


5.2.3.4 Mixer (Summing Amp) Widget

The Mixer Widget provides the facility to arbitrarily mix multiple channels (sources). It has two or more inputs and one output. Each input has an optional input amplifier (including an optional mute), which is optional to all inputs collectively; i.e., they must all have (or not have) the amplifier/mute. The output also has an optional amplifier and optional mute. Input and output amplifiers may be separately defined, but all input amplifiers (when present) must have the same parameters. In the event that input amplifiers with differing parameters are needed, or if amplifiers are used on only some of the Mixer Widget inputs, then the Mixer Widget specifies no input amplifiers, and separate amplifiers (based on Selector Widgets) are created with appropriate connections to the Mixer Widget.

The Mixer Widget may be 1- or 2-channel (mono or stereo) with particular connections rules defined inSection 5.1.3. In order to save parameter space in an Audio Function Group incorporating several Mixer Widgets, the function group node may optionally contain defaults for the amplifier parameters which may be used to define the Mixer Widget amplifiers; however, these defaults may be over-ridden with local parameters if necessary.

Figure 5-8. Mixer Widget



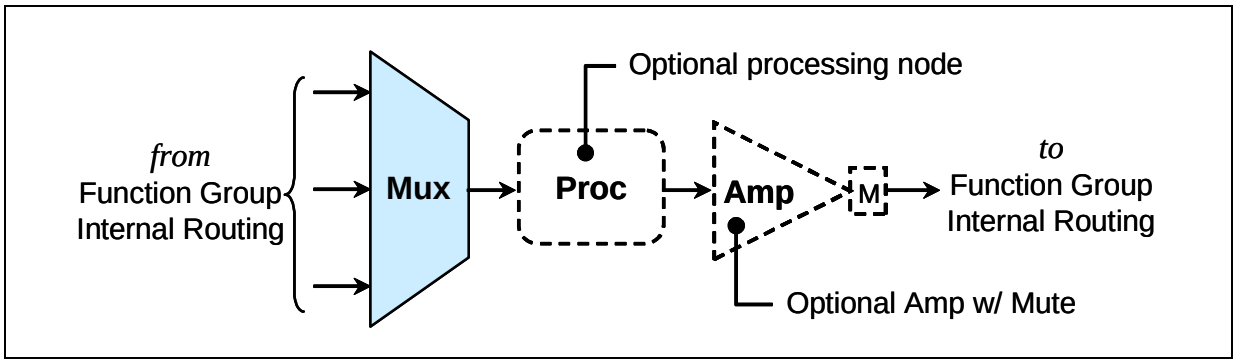
Inputs on the Mixer Widget are in its input Connection List and are hard wired (not selectable). Therefore, the Mixer Widget has no Connection Selector control. Input amplifiers/mutes are individually controlled by addressing their position (index) in the connection list. The Mixer Widget output will be available in the connection list of other widget(s).

5.2.3.5 Selector (Multiplexer) Widget

The Selector Widget provides a one-of-N signal selection. However, since the inputs to widgets generally have an implicit selector where needed, this widget may find infrequent use. A Selector Widget may be 1- or 2-channel and has one or more inputs and one output. Inputs on the Sum Widget are listed in its input Connection List; the output will be available in the Connection List of other widget(s). A selection among inputs is accomplished by a Connection Selector (control), the same as in other widgets with multiple inputs. The output may have an optional amplifier with mute. In order to save parameter space in an Audio Function Group incorporating several Selector Widgets, the function group node may optionally contain defaults for the amplifier parameters; however, these defaults may be over-ridden with local parameters if necessary.

Note that a degenerate (simple) Selector Widget (with only one input) allows a means of defining an arbitrarily placed amplifier or processing node.

Figure 5-9. Selector Widget



5.2.3.6 Power Widget

The Power Widget provides a convenient means to optimize power management within the Audio Function Group by providing a single point of power state control for an arbitrary group of audio widgets. The Power Widget has no connections with other widgets but still uses its Connection List to specify the set of widgets associated with and controlled hereby. These associations are defined at design time and are not dynamic. Writing power state control to this widget will effectively place all associated widgets in the prescribed power state. However, in no circumstance may the Power Widget place any Audio Widget in a power state higher than the current power state of the Audio Function Group. Note that while the Power Widget does have a Connection List, it does not contain a Connection Selector, since there are no direct connections to other widgets.

5.2.3.7 Volume Knob Widget

The Volume Knob Widget provides for mechanical volume control of specified output pins. The physical nature of the volume control is not specified - it could be an analog wheel (pot.), push buttons, etc. - however, it does have a declared capability of being absolute (e.g., pot.) or relative (e.g., push buttons). This widget has a connection list that describes which other widgets (presumably Pin Widgets) have their volume controlled by the Volume Knob Widget; since no dynamic connections are formed, there is no Connection Selector.

The Volume Knob can be set by software to directly control the associated “slave” amplifiers, or to send an unsolicited response, allowing the function driver to “read” the Volume Knob and then adjust the associated “slave” amplifiers indirectly.

5.2.3.8 Beep Generator Widget

The tone or Beep Generator Widget is an option used to generate an approximated sine wave by dividing the 48-kHz frame marker by a programmable amount. When the beep generator is actively generating a tone, its output drives all Pin Widgets which are currently defined output pins in a method of the vendor's choice, either by switching the pin to the beep signal or by mixing the tone into the currently playing stream. This node is never listed on any other node's connection list. The actual vendor-defined connection only persists while the Beep Generator is actively generating a tone. This widget may contain an optional amplifier.

This Beep Generator feature is independent of any optional “PC Beep Pin” or “Analog Beep Pin” input which is intended to receive an externally generated tone or sound. The presence of such a beep input pin is not exposed to software, nor defined in this specification. If used, this type of beep input would be connected through the codec to output pins in a vendor defined way, but such a connection may be maintained only while the Link reset (**RST#**) is asserted.

5.3 Codec Parameters and Controls

The foregoing function groups and widgets are formally specified in terms of parameters and controls, all of which are accessed by verbs. Parameters return static read-only information about the capabilities or configuration options of the codec, function group, or widget. Parameters are accessed, either with the “Get_Parameter” verb or, for connection lists, the “Get_Connection_List_Entry” verb. Controls have an effect on the behavior of the codec, such as setting a converter's data format or causing a reset to happen. Most controls are readable and writable using separate verbs defined for accessing each specific control, but some controls (such as RESET) are essentially write-only, and there is no associated verb to read a value.

All verbs are sent on the Link in the Command Field (first 40 bits) of outbound frames. Figure 5-9 shows the format of the Command Field; the first 8 bits are reserved and are transmitted as 0's. Commands additionally contain a 4-bit codec address (CAd) which is assigned at initialization time and identifies the target codec, together with an 8-bit node ID (NID) that identifies the target node within the codec. The 20-bit verbs vary in format and are each documented in subsequent sections.

Figure 5-10. Command Field Format

Bits 39:32	Bits 31:28	Bits 27:20	Bits 19:0
Reserved	CAd	NID	Verb

There are two types of verbs: those with 4-bit identifiers and 16 bits of data payload and those with 12-bit identifiers and 8 bits of payload. Because of the limited encoding space for the 4-bit identifiers, they are used sparingly for operations which need to convey data to the codec in 16-bit payloads. The values 0x7 and 0xF are not legal values for 4-bit verbs, as they select the extended 12-bit identifiers.

The SDO signal contains exactly one verb envelope in each frame. Since there is no valid bit for verbs, as there is for responses, an invalid verb is defined as all 0's sent to NID = 0. That is, if verb bits [27:0] are all 0's, the verb is invalid; otherwise, it is a valid verb and must have an associated response.

5.3.1 Required Verb Response

Link protocol requires that all commands have a valid response. A codec must return a valid response on the frame following the frame on which a command addressed to it was received. Responses are sent on the Link in the Response Field (first 36 bits) of inbound frames.

Figure 5-11 shows the format of the Response Field; reserved bits are transmitted as 0's. A 1 in the Valid bit position indicates the Response Field contains a valid response, which the controller will place in the RIRB; a 0 indicates there is no response. A 1 in the UnSol bit position is meaningful only when the Valid bit is set and indicates that the response is Unsolicited rather than in reply to a verb. Unsolicited responses may be used to signal the occurrence of asynchronous codec events to the software driver. The 32 actual response bits vary in format and are each documented in subsequent sections.

Figure 5-11. Response Field Format

Bit 35	Bit 34	Bits 33:32	Bits 31:0
Valid	UnSol	*Reserved*	Response

For Get commands, the response delivers the requested control status; for most Set commands, the response is usually all 0's. In both cases, a response must always be sent in the subsequent frame; out of order or delayed responses are not allowed.

In cases where a Get command actually initiates a sequence that will not be finished by the next link frame, a response must still be sent but may require a “Ready” bit or indication to allow software to properly interpret the response and poll for a completed response later if necessary. The only currently defined case where this is required is the Pin Sense “Execute” verb (Section Section 5.3.4.15) which returns a valid response with a value of 0h immediately and must be

queried at a later point when the Pin Sense operation is complete to determine the actual sensed value. In all other currently defined verbs, the valid response value is required on the following frame, and any vendor defined verbs must follow the same requirements.

Since verbs are function-specific, most verbs are not applicable to all nodes. If a verb is ever directed at a node for which that verb has no meaning (e.g., Set Stream/Channel directed at a Pin Widget), the node simply responds with all 0's. Hardware is not required to provide any form of error response to invalid or otherwise non-completing commands, other than returning a valid response of 0. In general, software may optionally verify completion of commands by reading back the command to see if the associated register holds the expected value.

5.3.2 Multiple SDI Operation

In certain applications, a codec may require more than a single **SDI** signal to support the required input bandwidth. In the simple case, there may be multiple functions on a single die, the aggregate input bandwidth of which exceeds the capacity of a single **SDI** signal. In this case, the designer may put multiple **SDI** signals on the codec but bind certain functions to a specific **SDI** signal. In this case, each group of functions must have its own root node; i.e., each **SDI** signal must be associated with a unique root node.

The more complex case comes when a single function's input bandwidth exceeds the capacity of a single **SDI** signal. In this case, multiple **SDI** signals must be fully sharable by any of the inputs within the function, and the function driver (software) alone binds input streams to specific **SDI** signals. In this case, there are multiple **SDI** signals associated with a single root node, and during codec initialization, each **SDI** is assigned a unique Cad. However, the hardware designer arbitrarily designates one of them as the “primary **SDI**,” and all verb responses must be returned on that single **SDI** signal. Verbs can (and will be) addressed to any of the CAds associated with that root node, but the responses must always come back on the primary **SDI** signal. In this case, the function driver discovers which **SDI** signals are associated to a given root node, and therefore sharable, by requesting the “Device ID” on each of the valid codec addresses (CAd), and then observing which **SDI** signal the response is returned upon. After mapping the “primary **SDI**” or “primary CAd,” as well as the associations between **SDI**s, it will carry out all further command and control functions on the primary address but will dynamically distribute the input data bandwidth between all associated **SDI**s. Unsolicited responses may be returned on any of the associated **SDI**s; however, it is recommended they also use only the primary **SDI**, unless unsolicited response latency is a factor.

In this case of multiple **SDI**s, the function driver assigns each input stream to a specific **SDI** as described in Section 5.3.4.12. However, this binding is dynamic and may change in mid-stream, as the function driver distributes the bandwidth of streams as they are created and terminated. It is therefore essential that the codec observe exactly the required timing of any assignment change. Note that these bindings are transparent to controllers that handle incoming traffic by stream ID and not by codec or **SDI** address.

5.3.3 Controls

5.3.3.1 Get Parameter

The **Get Parameter** command returns the value of the parameter indicated by the payload. Although all nodes support the Get Parameter verb, the specific parameters which may be requested depends on the node type. Specific requirement, by node type, on which parameters must be supported, is listed in Table 5-46 on page 174. The parameters are detailed in Section 5.3.4.

Command Options:

Table 5-1. Get Parameter Command

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F00h	The ID of the parameter to read	The parameter value

Applies to:

- All Nodes (Root Node, Function Group nodes, and Widget nodes)

5.3.3.2 Connection Select Control

For widgets that have multiple inputs, the Connection Select control determines which input is currently active. The index is in relation to the Connection List associated with the widget. The index is a zero-based offset into the Connection List.

If the Connection List Length value is 1, there is only one hard wired input possible and, therefore, there is no Connection Select control, and the verb for this control is not operable on that widget. The actual connection is read from the Connection List associated with the widget, as usual.

If an attempt is made to Set an index value greater than the number of list entries (index is equal to or greater than the Connection List Length property for the widget) the behavior is not predictable.

Command Options:

Table 5-2. Connection Select Control

	Verb ID	Payload (8 bits)	Response (32 bits)
Get	F01h	0	Bits 31:8 are 0 Bits 7:0 are the Connection index currently set
Set	701h	The Connection Index value to be set	0

Applies to:

- Input Converter
- Selector Widget
- Pin Complex
- Other

5.3.3.3 Get Connection List Entry

Returns the Connection List Entries at the index supplied. This command provides a way for the software to query the complete Connection List for a widget multiple entries at a time.

The requested index *n* is zero based. If the Long Form bit of the Connection List Length parameter (refer to Section 5.3.4.11) is 1, *n* must be even, and two long form Connection List entries will be returned. Therefore, requesting index 0 will return the values at offset 0 and 1, requesting index 2 will return the Connection List Entries at offset 2 and 3, etc. If the Long Form bit of the Connection List Length parameter is 0, *n* must be a multiple of four, and four short form Connection List entries will be returned. Therefore, requesting index 0 will return the values at offset 0, 1, 2, and 3, requesting index 4 will return the Connection List Entries at offset 4, 5, 6, and 7, etc.

If an index value is requested where the offset of the Connection List Entries would be greater than the number of Connection List Entries, the number of entries beyond the end of the list would be reported as 0's. For example, if the Connection List Length parameter is read as 0002h, indicating that there are two short form Connection List Entries in the list, requesting the Connection List Entry with *n* equal to 0 will return List Entry 0 in bits 7:0, List Entry 1 in bits 15:8, and 0's for the other list entries.

If an index value of greater than the number of list entries (n is equal to or greater than the Connection List Length property for the widget) the behavior is not predictable.

Command Options:

Table 5-3. Connection List Entry Control

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F02h	Offset index n	See response format below

Get Response With Short Form List Entries:

Figure 5-12. Response Format

31:16	24:16	15:8	7:0
Connection List Entry n+3	Connection List Entry n+2	Connection List Entry n+1	Connection List Entry n

Get Response With Long Form List Entries:

Figure 5-13. Response Format

31:16	15:0
Connection List Entry n+1	Connection List Entry n

The value returned contains the Connection List Entry *n* in the lower Word and Entry n+1 in the upper Word. If the list contains an odd number of list entries and the highest legal index was requested, the top Word would contain 0, as there is no valid list entry at index n+1.

Applies to:

- Audio Input Converter
- Mixer Widget
- Selector Widget

- Pin Widget
- Power Widget
- Other Widget

5.3.3.4 Processing State

For widgets that implement processing capabilities in the widget, the **Processing State** command provides a way to control the behavior of the widget.

All widgets that implement processing must support the value 00 (Processing Off) and 01 (Processing On). The Processing Parameters can be queried to determine whether the Processing Benign state is supported.

Command Options:

Table 5-4. Processing State

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F03h	0	Bits 31:8 are 0 Processing State is in bits 7:0
Set	703h	Processing State	0

The **Processing State** field may have the following values:

00h: Processing Off: The widget must not make any modifications to the stream passing through it.

01h: Processing On: The widget may perform any processing it wishes.

02h: Processing Benign: The widget must not make any modifications to the stream passing through it which are not linear and time invariant. For instance, speaker compensation or EQ processing is acceptable, but 3-D spatialization processing is not. If the widget does not support a benign mode of processing, it treats the “Benign” setting the same as the “Off” setting.

03-7Fh: *Reserved*

80-FFh: Vendor Specific. A vendor may use these modes to control the processing of the widget.

Applies to:

- Input Converter
- Output Converter
- Selector Widget
- Pin Complex Widget
- Other Widget

5.3.3.5 Coefficient Index

For widgets that have loadable processing coefficients, the **Coefficient Index** is a zero-based index into the processing coefficient list which will be either read or written using the Processing Coefficient control.

When the coefficient has been loaded, the Coefficient Index will automatically increment by one so that the next Load Coefficient verb will load the coefficient into the next slot.

If Coefficient Index is set to be greater than the number of “slots” in the processing coefficient list, unpredictable behavior will result if an attempt is made to Get or Set the processing coefficient.

Command Options:

Table 5-5. Coefficient Index

	Verb ID	Payload (16 Bits)	Response (32 Bits)
Get	D	0	Bits 31:16 are 0 index n is in bits 15:0
Set	5	Index n	0

Applies to:

- Input Converter (that has loadable processing coefficients)
- Output Converter (that has loadable processing coefficients)
- Selector Widget (that has loadable processing coefficients)
- Pin Complex Widget (that has loadable processing coefficients)
- Other Widget (that has loadable processing coefficients)

5.3.3.6 Processing Coefficient

The Processing Coefficient control is available on widgets that have processing capabilities, and have loadable coefficients. If the “ProcWidget” bit in the Audio Widget Capabilities parameter is set, indicating that the node has processing capabilities, the Processing parameters can be queried to determine the number of coefficients that can be loaded.

Processing Coefficient loads the value n into the widget's coefficient array at the index determined by the Coefficient Index control. When the coefficient has been loaded, the Coefficient Index will automatically increment by one so that the next Load Coefficient verb will load the coefficient into the next slot.

Command Options:

Table 5-6. Processing Coefficient

	Verb ID	Payload (16 Bits)	Response (32 Bits)
Get	Ch	0	The coefficient value n (in the lower 16 bits)
Set	4h	The value n of the 16 bit coefficient to set	0

Applies to:

- Input Converter (that has loadable processing coefficients)
- Output Converter (that has loadable processing coefficients)
- Selector Widget (that has loadable processing coefficients)
- Pin Complex Widget (that has loadable processing coefficients)
- Other Widget (that has loadable processing coefficients)

5.3.3.7 Amplifier Gain/Mute

The **Amplifier Gain/Mute** control determines the gain (or attenuation) of one or several amplifiers in a widget. Depending on the combination of bits set, anywhere from a single channel on a single amplifier to both channels on both input and output amplifiers, may be programmed at the same time. Each amplifier setting must be read individually, however.

Command Options:

Table 5-7. Amplifier Gain/Mute

	Verb ID	Payload (16 Bits)	Response (32 Bits)
Get	Bh	See Note A	See Note B
Set	3h	See Note C	0

Note A: Get Payload Format:

Figure 5-14. Amplifier Gain/Mute Get Payload

15	14	13	12:4	3:0
Get Output/Input	0	Get Left/Right	0	Index

Get Output/Input controls whether the request is for the input amplifier or output amplifier on a widget. If 1, the output amplifier is being requested; if 0 the input amplifier is being requested.

Get Left/Right controls whether the request is for the left channel amplifier or right channel amplifier on a widget. If 1, the left amplifier is being requested; if 0, the right amplifier is being requested.

Index specifies the input index of the amplifier setting to return if the widget has multiple input amplifies, such as a Sum Widget. The index corresponds to the input's offset in the Connection List. If the widget does not have multiple input amplifiers, or if “Get Output/Input” is a 1 (requesting the output amplifier), these bits have no meaning and are ignored. If the specified index is out of range, all 0's are returned in the response.

Note B: Get Response Format:

Figure 5-15. Amplifier Gain/Mute Get Response

31:8	7	6:0
0	Amplifier Mute	Amplifier Gain

The Get response returns the Gain and Mute settings for the amplifier requested in the Get payload. If the “Get Payload,” “Get Input/Output,” “Get Left/Right,” or “Index” bits are set to request the value of a Amplifier which does not exist in a widget, the response to the Get will be 00000000h.

Note C: Set Payload Format:

Figure 5-16. Amplifier Gain/Mute Set Payload

15	14	13	12	11:8	7	6:0
Set Output Amp	Set Input Amp	Set Left Amp	Set Right Amp	Index	Mute	Gain

Set Input Amp and **Set Output Amp** determine whether the value programmed refers to the input amplifier or the output amplifier in widgets which have both, such as Pin Widgets, Sum Widgets, and Selector Widgets. A 1 indicates that the relevant amplifier should accept the value being set. If both bits are set, both amplifiers are set; if neither bit is set, the command is effectively a no-op. Any attempt to set a non-existent amplifier is ignored.

Set Left and **Set Right** determine whether the left (channel 0) or right (channel 1) channel of the amplifier is being affected. A 1 indicates that the relevant amplifier should accept the value being set. If both bits are set, both amplifiers are set. Any attempt to set a non-existent amplifier is ignored. If the widget only supports a single channel, these bits are ignored and the value programmed applies to the left (channel 0) amplifier.

Index is only used when programming the input amplifiers on Selector Widgets and Sum Widgets, where each input may have an individual amplifier. The index corresponds to the input's offset in the Connection List. If the widget being programmed is not a Selector Widget or a Sum Widget, or the **Set Input Amp** bit is not set, this field is ignored. If the specified index is out of range, no action is taken.

Mute selects - gain, but the hardware implementation will determine the actual degree of mute provided. A value of 1 indicates the mute is active. Generally, mute should default to 1 on codec reset, but there may be circumstances where mute defaults to its off or inactive state. In particular, if an analog PC Beep Pin is used, the mutes of associated outputs must default to 0 to enable the beep signal prior to the codec coming out of reset. This bit is ignored by any amplifier that does not have a mute option.

Gain is a 7-bit “step” value specifying the amplifier gain, the actual dB value of which is determined by the “StepSize,” “Offset,” and “NumSteps” fields of the Output Amplifier Capabilities parameter for a given amplifier. After codec reset, this “Gain” field must default to the “Offset” value, meaning that all amplifiers, by default, are configured to 0 dB gain. If a value outside the amplifier's range is set, the results are undetermined.

Applies to:

- Input Converter
- Output Converter
- Pin Complex
- Selector Widgets
- Mixer Widgets
- Other Widgets

5.3.3.8 Converter Format

The **Converter Format** control determines the format the converter will use. This must match the format programmed into the Stream Descriptor on the controller so that the data format being transmitted on the link matches what is expected by the consumer of the data.

All of the field definitions of the format structure are applicable to the audio converter. It is important to note that even though the converter widget is only stereo or mono, the CHAN field is still necessary to indicate to the converter the size of a block in cases where MULT is greater than 1.

Command Options:

Table 5-8. Converter Format

	Verb ID	Payload (16 Bits)	Response (32 Bits)
Get	Ah	0	Bits 31:16 are 0 format is in bits 15:0
Set	2h	Format	0

Applies to:

- Input Converter
- Output Converter

5.3.3.9 Digital Converter Control

The **Digital Converter Controls 1** and **2** operate together to provide a set of bits to control the various aspects of the digital portion of the Converter Widget. The S/PDIF IEC Control (SIC) bits are supported in one of two ways.

In the first case referred to as “Codec Formatted SPDIF,” if a PCM bit stream of less than 32 bits is specified in the Converter Format control, then the S/PDIF Control bits, including the “V,” “PRE,” “/AUDIO,” and other such bits are embedded in the stream by the codec using the values (SIC bits) from the Digital Converter Control 1 and 2. On an input PCM stream of less than 32 bits, the codec strips off these SIC bits before transferring the samples to the system and places them in the Digital Converter Control 1 and 2 for later software access.

In the second case referred to as “Software Formatted (or Raw) SPDIF,” if a 32-bit stream is specified in the Converter Format control, the S/PDIF IEC Control (SIC) bits are assumed to be embedded in the stream by software, and the raw 32-bit stream is transferred on the link with no modification by the codec. Similarly, on a 32-bit input stream, the entire stream is transferred into the system without the codec stripping any bits. However, the codec must properly interpret the Sync Preamble bits of the stream and then send the appropriately coded preamble. The *IEC60958 Specification*, Section 4.3, *Preambles*, defines the preambles and the coding to be used. Software will specify the “B,” “M,” or “W” (also known as “X,” “Y,” or “Z”) preambles by encoding the last four bits of the preamble into the Sync Preamble section (bits 0-3) of the frame. The codec must examine the bits specified and encode the proper preamble based on the previous state. The previous state is to be maintained by the codec hardware. For more information on Preamble Coding, consult Section 4.3 of the *IEC 60958 Specification.*

Table 5-9. SPDIF Sync Preamble Bits

Preamble Bits Set by Software (Bits 3:0 of Frame)	Preamble Coding	
	Previous State = 0	Previous State = 1
1000b (“B” or “Z”)	11101000	00010111
0010b (“M” or “X”)	11100010	00011101
0100b (“W” or “Y”)	11100100	00011011

Command Options:

Table 5-10. S/PDIF Converter Control 1 and 2

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F0Dh[1]	0	Bits 31:16 are 0 Bits 15:0 are SIC bits
Set 1	70Dh	SIC bits [7:0]	0
Set 2	70Eh	SIC bits [15:8]	0

NOTES:

1. The Verb Code F0Eh is reserved for S/PDIF Converter Control 2 and may never be reassigned to anything else. However, it need not be implemented since standard software drivers must never use it. If a codec elects respond to this code, the response must be identical in all respects to the response to Verb Code F0Dh.

Figure 5-17. S/PDIF IEC Control (SIC) Bits

15	14:8	7	6	5	4	3	2	1	0
Rsvd	CC[6:0]	L	PRO	/AUDIO	COPY	PRE	VCFG	V	DigEn

CC[6:0] (Category Code): Programmed according to IEC standards, or as appropriate.

L (Generation Level): Programmed according to IEC standards, or as appropriate.

PRO (Professional): 1 indicates Professional use of channel status; 0 indicates Consumer.

/AUDIO (Non-Audio): 1 indicates data is non-PCM format; 0 indicates data is PCM.

COPY (Copyright): 1 indicates copyright is asserted; 0 indicates copyright is not asserted.

PRE (Preemphasis): 1 indicates filter preemphasis is 50/15 µs; 0 preemphasis is none.

VCFG (Validity Config.): Determines S/PDIF transmitter behavior when data is not being transmitted. When asserted, this bit forces the de-assertion of the S/PDIF “Validity” flag, which is bit 28 transmitted in each S/PDIF subframe. This bit is only defined for Output Converters and is defined as Reserved, with a Read Only value of 0 for Input Converters.

- If “V” = 0 and “VCFG”=0, then for each S/PDIF subframe (Left and Right) bit[28] “Validity” flag reflects whether or not an internal codec error has occurred (specifically whether the S/PDIF interface received and transmitted a valid sample from the Intel High Definition Audio Link). If a valid sample (Left or Right) was received and successfully transmitted, the “Validity” flag should be 0 for that subframe. Otherwise, the “Validity” flag for that subframe should be transmitted as “1.”

- If “V” = 0 and “VCFG” = 1, then for each S/PDIF subframe (Left and Right), bit[28] “Validity” flag reflects whether or not an internal codec transmission error has occurred. Specifically, an internal codec error should result in the “Validity” flag being set to 1. In the case where the S/PDIF transmitter is not receiving a sample or does not receive a valid sample from the Intel High Definition Audio Controller (Left or Right), the S/PDIF transmitter should set the S/PDIF “Validity” flag to 0 and pad each of the S/PDIF “Audio Sample Word” in question with 0's for the subframe in question. If a valid sample (Left or Right) was received and successfully transmitted, the “Validity” flag should be 0 for that subframe.
- If “V” = 1 and “VCFG” = 0, then each S/PDIF subframe (Left and Right) should have bit[28] “Validity” flag = 1. This tags all S/PDIF subframes as invalid.
- “V” = 1 and “VCFG” = 1 state is reserved for future use.
- Default state, coming out of reset, for “V” and “VCFG” should be 0 and 0 respectively.

V (Validity): This bit affects the “Validity flag,” bit[28] transmitted in each subframe, and enables the S/PDIF transmitter to maintain connection during error or mute conditions. The behavior of the S/PDIF transmitter with respect to this bit depends on the value of the “VCFG” bit.

DigEn (Digital Enable): Enables or disables digital transmission through this node. A 1 indicates that the digital data can pass through the node. A 0 indicates that the digital data is blocked from passing through the node, regardless of the state.

Applies to:

- Input Converter
- Output Converter

5.3.3.10 Power State

The **Power State** control determines the power state of the node to which it refers. There is no required power saving or maximum allowed power in any of the low power states; rather these states allow the vendor to reduce power by as much or as little as desired to meet their customer needs. However, power must never be reduced to a given circuit in a manner that would be inconsistent with the specified power recovery requirements of that power state.

Command Options:

Table 5-11. Power State

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F05h	0	Bits 31:8 are 0 PS-Act is in bits 7:4 PS-Set is in bits 3:0
Set	705h	PS-Set in bits 0:3 bits 4:7 are 0	0

PS-Set is a PowerState field which defines the current power setting of the referenced node. If the referenced node is of any type other than a Function Group node, the actual power state is a function of both this setting and the PowerState setting of the Function Group node under which the currently referenced node was enumerated (is controlled).

PS-Act is a PowerState field which indicates the actual power state of the referenced node. Within a Function Group type node, this field will always be equal to the PS-Set field (modulo the time required to execute a power state transition). Within any other type of node, this field will be the

lower power consuming state of either a) the PS-Set field of the currently referenced node or b) the PS-Set field of the Function Group node under which the currently referenced node was enumerated (is controlled).

All PowerState fields are defined as follows:

PowerState[1:0]:

00: Node Power state (D0) is fully on.

01: Node Power state (D1) allows for (does not require) the lowest possible power consuming state from which it can return to the “fully on” state (D0) within 10 ms, excepting analog pass through circuits (e.g., CD analog playback) which must remain fully on.

10: Node Power state (D2) allows for (does not require) the lowest possible power consuming state from which it can return to the “fully on” state (D0) within 10 ms. For modems, this is the “wake on ring” power state.

11: Node Power state (D3) allows for (does not require) lowest possible power consuming state under software control. Note that any low power state set by software must retain sufficient operational capability to properly respond to a subsequent software Power State command.

PowerState[3:2]: *Reserved*, always 0.

While Function Group nodes (Audio Function, Modem Function, etc.) and Power Widget nodes must support this control, other widget nodes may optionally implement this control to provide more fine-grained power management of the codec. For Audio Widgets, such as Input Converter or Output Converter Widgets, the Audio Widget Capabilities parameter (see Section 5.3.4.6) will define whether this control is supported.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group
- Power Widget
- Input Converter (Optional)
- Output Converter (Optional)
- Selector Widget (Optional)
- Mixer Widget (Optional)
- Pin Complex (Optional)

5.3.3.11 Converter Stream, Channel

Converter Stream, Channel controls the link Stream and Channel with which the converter is associated.

Command Options:

Table 5-12. Converter Control

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F06h	0	Bits 31:8 are 0 Stream is in bits 7:4 Channel is in bits 3:0
Set	706h	Stream is in bits 7:4 Channel is in bits 3:0	0

Stream is an integer representing the link stream used by the converter for data input or output. 0000b is stream 0, 0001b is stream 1, etc. Although the link is capable of transmitting any stream number, by convention stream 0 is reserved as unused so that converters whose stream numbers have been reset to 0 do not unintentionally decode data not intended for them.

Channel is an integer representing the lowest channel used by the converter. If the converter is a stereo converter, the converter will use the channel provided, as well as channel+1, for its data input or output.

Applies to:

- Input Converter
- Output Converter

5.3.3.12 Input Converter SDI Select

Input Converter SDI Select controls which Intel High Definition Audio Link **SDI** signal is used to transmit all data (samples) of a given stream; this control is only relevant in codecs which support multiple **SDI** signals (refer to Section 5.3.2). The control is addressed to, and set in, any Input Converter to which channel zero of a stream has been assigned. The codec then transmits all channels of the associated stream on the designated **SDI** signal, since a single stream may never be broken across multiple **SDI**s. This implies that Input Converters not assigned to channel zero of a stream must slave this control off that of the Input Converter that is assigned to channel zero of the same stream. This control is not valid if it is referenced to an Input Converter not assigned to channel zero of a stream; in this case, 0's are returned and no action is taken.

Command Options:

Table 5-13. SDI Select

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F04h	0	Bits 31:4 are 0 SDI-Select is in bits 3:0
Set	704h	bits 7:4 are 0 SDI-Select is in bits 3:0	0

SDI-Select is an integer representing the hardware assigned CAd of the SDI signal to be used by the converter for data input of its assigned stream.

Note that SDI assignment may be dynamically changed; operation of the Link frame in which this command is received will remain unchanged, but any SDI (re)assignment will be effective in the immediately subsequent frame - the one in which the command response would be returned.

5.3.3.13 Pin Widget Control

Pin Widget Control controls several aspects of the Pin Widget.

Command Options:

Table 5-14. Enable VRef

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F07h	0	Bits 31:8 are 0 Bits 7:0 are PinCntl
Set	707h	Bits 7:0 are PinCntl	0

PinCntl format:

Figure 5-18. Pin Cntl Format

7	6	5	4:3	2:0
H-Phn Enable	Out Enable	In Enable	Rsvd	VRefEn

H-Phn Enable disables/enables a low impedance amplifier associated with the output. The value 1 enables the amp. Enabling a non-existent amp is ignored.

Out Enable allows the output path of the Pin Widget to be shut off. The value 1 enables the path. Enabling a non-existent amp is ignored.

In Enable allows the input path of the Pin Widget to be shut off. The value 1 enables the path.

VRefEn: Voltage Reference Enable controls the VRef signal(s) associated with the Pin Widget. If more than one of the bits in the VRef[7:0] field of the Pin Capabilities parameter (Section 5.3.4.9) are non-zero, then this control allows the signal level to be selected.

The VRefEn field encoding selects one of the possible states for the VRef signal(s). If the value written to this control does not correspond to a supported value as defined in the Pin Capabilities parameter, the control must either retain the previous value or take the value of 000, which will put the control in a Hi-Z state and prevent damage to any attached components.

Table 5-15 enumerates the possible values for VRefEn which correlate to the values identified in the Pin Capabilities parameter (see Figure 5-32).

Table 5-15. VRefEn Values

VRefEn Encoding	VREF Signal Level
000b	Hi-Z
001b	50%
010b	Ground (0 V)
011b	*Reserved*
100b	80%
101b	100%
110b-111b	*Reserved*

Applies to:

- Pin Complex

5.3.3.14 Unsolicited Response

The **Unsolicited Response** control determines whether the node is enabled to send an unsolicited response, as well as what the tag will be for the response.

This control is only available for nodes which support Unsolicited Responses, as declared in the Function Group Type parameter (Section 5.3.4.4) and the Audio Widget Capabilities parameter (Section 5.3.4.6). The node should be queried to determine if it supports unsolicited responses before getting or setting this control.

Command Options:

Table 5-16. Connection Set Control

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F08h	0	Bits 31:8 are 0 EnableUnsol is bits 7:0
Set	708h	EnableUnsol is bits 7:0	0

EnableUnsol format:

Figure 5-19. EnableUnsol Format

7	6	5:0
Enable	0	Tag

Enable controls the actual generation of Unsolicited Responses. If Enable is a 1, Unsolicited Responses may be generated.

Tag is a 6-bit value which is opaque to the codec and is used by software to determine what codec node generated the unsolicited response. The value programmed into the Tag field is returned in the top 6 bits (31:26) of every Unsolicited Response generated by this node.

Applies to:

- All Nodes capable of generating Unsolicited Responses.

5.3.3.15 Pin Sense

The **Pin Sense** control returns the Presence Detect status and the impedance measurement of the device attached to the pin.

Some codecs may require that the impedance measurement be triggered by software; in that case, sending the Execute command will cause the impedance measurement to begin. The “Presence Detect” bit will always be accurate if that functionality is supported by the widget.

Note that the Pin Complex Widget may support the generation of an Unsolicited Response to indicate that the Sense Measurement (either the Presence Detect or the Impedance) value has changed, the generation of which implies that the measurement is complete.

Command Options:

Table 5-17. Pin Sense

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F09h	0	Presence Detect: bit 31 Impedance Sense: bits 30:0
Execute	709h	Right Chnl: bit 0 Rsvd: bits 1:7	0

Presence Detect is a bit indicating the state of the Presence Detect capability. A 1 indicates that there is “something” plugged into the jack associated with the Pin Complex. This bit will only be valid if the widget has Presence Detect capability as indicated by the “Presence Detect Capable” bit of the Pin Capabilities parameter (see Section 5.3.4.9).

Impedance returns the measured impedance of the widget. A returned value of 0x7FFF,FFFF (all 1’s) indicates that a valid sense reading is not available, or the sense measurement is busy (refer to Section 5.3.1) if it has been recently triggered. This field is only valid if the widget has Sense capability as indicated by the “Impedance Sense Capable” bit of the Pin Capabilities parameter.

- **Right Chnl:** Normally impedance sensing is done on the left channel or “tip” of the connector. However, Pin Widgets may optionally support sensing on the right channel or “ring” of the connector. When this bit is 1, the impedance value is taken on the right channel if the Pin Widget supports this; if not supported, this bit is ignored. When this bit is 0, the left channel is sensed.

Applies to:

- Pin Complex

5.3.3.16 EAPD/BTL Enable

EAPD/BTL Enable controls the EAPD pin and configures Pin Widgets into balanced output mode, when these features are supported. It also configures any widget to swap L and R channels when this feature is supported. When this control is referenced to a non-Pin Widget, bits 1 and 0 are not valid, and are considered reserved.

Command Options:

Table 5-18. EAPD/BTL Enable

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F0Ch	0	bits 31:3 are *Reserved* bit 2 = L-R Swap bit 1 = EAPD bit 0 = BTL
Set	70Ch	bits 7:3 are *Reserved* bit 2 = L-R Swap bit 1 is EAPD bit 0 is BTL	0

L-R Swap causes the left and right channels of the Widget to be swapped for both input and output paths if they exist. The value 1 enables swapping.

EAPD value is reflected on the EAPD pin; a 1 causes the external amplifier to power up, and a 0 causes it to power down. When the EAPD value = 1, the EAPD pin must be placed in a state appropriate to the current power state of the associated Pin Widget; i.e., the external amplifier must be powered down whenever the associated Pin Widget is put in a low power state, even though the EAPD value may remain 1. It is possible that more than one Pin Widget supports the EAPD function as indicated in bit 16 of the Pin Capabilities Parameter (Section 5.3.4.9); this would be true, for example, when external amps were supported for 4-channel output. In this case, each supporting Pin Widget must respond to this control; however, since there is only a single EAPD Pin, there is also only one logical value to set/get, and that value must be accessible, using this control, via any/all supporting Pin Widgets.

BTL controls the output configuration of a Pin Widget which has indicated support for balanced I/O (bit 6, Pin Capabilities Parameter). When this bit is 0, the output drivers are configured in normal, single-ended mode; when this bit is 1, they are configured in balanced mode. Note that in balanced mode, the Pin Widget has twice as many pins as it does in normal mode; i.e., a stereo Pin Widget in balanced mode has four signal pins (in addition to Vref pins). However, in both modes it must appear to software as a single Pin Widget. The additional pins must be reserved to this purpose; thus, in a configuration supporting BTL outputs the additional pins may not be enumerated as a separate Pin Widget.

Applies to:

- Any Audio Widget

5.3.3.17 GPI Data

GPI Data returns the current GPI pin status as a bit field.

Command Options:

Table 5-19. GPI Data

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F10h	0	Bits 31:8 are *Reserved* Bits 7:0 is Data
Set	710h	Data	0

Data is a bit field reporting the current state of the GPI signals.

Bits in the GPI control are not directly writable, but if the bit is configured as “Sticky” in the GPI Sticky control (below), writing a 1 to the bit will clear the bit.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.18 GPI Wake Enable Mask

GPI Wake controls the ability of a change on a GPI pin to cause a Status Change event on the Link.

Command Options:

Table 5-20. GPI Wake Mask

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F11h	0	Bits 31:8 are *Reserved* Bits 7:0 is Wake
Set	711h	Wake	0

Wake is a bit field representing the state of the GPI Wake bits.

The GPI Wake provides a mask for determining if a GPI change will generate a wake-up (0 = No, 1 = Yes). When the Link is powered down (**RST#** is asserted), a wake-up event will trigger a Status Change Request event on the link. When the Link is powered up, wake events would be Unsolicited Responses, the generation of which is separately controlled (see Section 5.3.3.19).

The default value after cold or register reset for this register (0000h) defaults to all 0’s specifying no wake-up event. Non-implemented GPI pins always return 0’s.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.19 GPI Unsolicited Enable Mask

GPI Unsolicited controls the ability of a change on a GPI pin to cause an Unsolicited Response on the Link.

Command Options:

Table 5-21. GPI Unsolicited Enable

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F12h	0	Bits 31:8 are *Reserved* Bits 7:0 is UnsolEnable
Set	712h	UnsolEnable	0

UnsolEnable is a bit field representing the state of the GPI Wake bits.

The GPI Unsolicited control provides a mask for determining if a change on a GPI line will generate a wake-up (0 = No, 1 = Yes). If enabled, and the Unsolicited Response control has been set to enable unsolicited responses on the function group, an unsolicited response will be sent when a GPI line changes state.

The default value after cold or register reset for this register (0000h) defaults to all 0's specifying no wake-up event. Non-implemented GPI pins always return 0's.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.20 GPI Sticky Mask

GPI Data returns the current GPI pin status as a bit field.

Command Options:

Table 5-22. GPI Sticky Mask

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F13h	0	Data
Set	713h	Bits 31:8 are *Reserved* Bits 7:0 is Data	0

Data is a bit field reporting the current state of the GPI signals.

The GPI Pin Sticky is a read/write register that defines GPI Input Type (0 = Non-Sticky, 1 = Sticky). GPI inputs configured as Sticky are cleared by writing a 0 to the corresponding bit of the GPI Data control (see Section 5.3.3.17) and by reset.

The default value after cold or register reset for this register (0000h) is all pins Non-Sticky. Unimplemented GPI pins always return 0s. Sticky is defined as edge-sensitive, Non-Sticky as Level-sensitive.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.21 GPO Data

GPO Data controls the state of the GPO pins.

Command Options:

Table 5-23. GPO Data

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F14h	0	Data
Set	714h	Bits 31:8 are *Reserved* Bits 7:0 is Data	0

Data is a bit field representing the state of the GPO signals on the codec.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.22 GPIO Data

GPIO Data sets and returns the data associated with the GPIO signals. The control is a bit field, with each bit corresponding to a GPIO pin starting from bit 0.

Command Options:

Table 5-24. GPIO Data

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F15h	0	Data
Set	715h	Bits 31:8 are *Reserved* Bits 7:0 is Data	0

Data is a bit field representing the current state of the GPIO signals on the codec.

For bits in the GPIO Data control that have corresponding bits in the GPIO Direction control set to 0, the pin behaves as an input and the bit value will be a read-only value indicating the state being sensed by the pin. Bits that are configured as outputs in the GPIO Data control are read/write in the GPIO Data control, and the value written will determine the value driven on the pin. Reads from output bits will reflect the value being driven by the GPIO pin.

Note that if the corresponding bit in the GPIO Enable control is not set, pins configured as outputs will not drive associated bit value (as the pin must be in a Hi-Z state), but the value returned on a read will still reflect the value that would be driven if the pin were to be enabled in the GPIO Enable control.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.23 GPIO Enable Mask

GPIO Enable controls whether a GPIO pin is enabled or not. The control is a bit field, with each bit corresponding to a GPIO pin starting from bit 0.

Command Options:

Table 5-25. GPIO Enable

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F16h	0	Enable
Set	716h	Bits 31:8 are *Reserved* Bits 7:0 is Enable	0

Enable is a bit field representing the configuration of the GPIO signals. For each GPIO pin, there is an associated bit in the Enable field. If the bit associated with a pin is 0, the pin is disabled, and must be in a Hi-Z state. If the bit is a 1, the GPIO pin is enabled and the pin's behavior will be determined by the GPIO Direction control.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.24 GPIO Direction

GPIO Direction determines whether a given GPIO is configured to drive or sense the connected signal. The control is a bit field, with each bit corresponding to a GPIO pin starting from bit 0.

Command Options:

Table 5-26. GPIO Direction

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F17h	0	Control
Set	717h	Bits 31:8 are *Reserved* Bits 7:0 is Control	0

Control is a bit field representing the configuration of the GPIO signals. If a bit position is a 0, the associated GPIO signal is configured as an input. If the bit is set to a 1, the associated GPIO signal is configured as an output.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.25 GPIO Wake Enable Mask

GPIO Wake controls the ability of a change on a GPIO pin configured as an input to cause a Status Change event on the Link.

Command Options:

Table 5-27. GPIO Wake Enable

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F18h	0	Bits 31:8 are *Reserved* Bits 7:0 is Wake
Set	718h	Wake	0

Wake is a bit field representing the state of the GPIO Wake bits.

The GPIO Wake provides a mask for determining if a change on a GPIO pin will generate a wake-up (0 = No, 1 = Yes). When the Link is powered down (**RST#** is asserted), a wake-up event will trigger a Status Change Request event on the link. When the Link is powered up, wake events would be Unsolicited Responses, the generation of which is separately controlled (see Section 5.3.3.26).

The default value after cold or register reset for this register (0000h) defaults to all 0's specifying no wake-up event. Non-implemented GPIO pins always return 0's.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.26 GPIO Unsolicited Enable Mask

GPIO Unsolicited controls the ability of a change on a GPIO pin configured as an input to cause an Unsolicited Response on the Link.

Command Options:

Table 5-28. GPIO Unsolicited Enable

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F19h	0	Bits 31:8 are *Reserved* Bits 7:0 is UnsolEnable
Set	719h	UnsolEnable	0

UnsolEnable is a bit field representing the state of the GPIO Unsolicited Enable bits.

The GPIO Unsolicited Enable Mask control provides a mask for determining if a change on a GPIO line will generate a wake-up (0 = No, 1 = Yes). If enabled, and the Unsolicited Response control has been set to enable unsolicited responses on the widget, an unsolicited response will be sent when a GPIO line changes state.

The default value after cold or register reset for this register (0000h) defaults to all 0's specifying no wake-up event. Non-implemented GPIO pins always return 0's.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.27 GPIO Sticky Mask

GPIO Data returns the current GPIO pin status as a bit field.

Command Options:

Table 5-29. GPIO Sticky Mask

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F1Ah	0	Data
Set	71Ah	Bits 31:8 are *Reserved* Bits 7:0 is Data	0

Data is a bit field reporting the current state of the GPIO signals.

The GPIO Sticky Mask defines GPIO Input Type (0 = Non-Sticky, 1 = Sticky) when a GPIO pin is configured as an input. GPIO inputs configured as Sticky are cleared by writing a 0 to the corresponding bit of the GPIO Data Control (see Section 5.3.3.22) and by reset.

The default value after cold or register reset for this register (0000h) is all pins Non-Sticky. Unimplemented GPIO pins always return 0's. Sticky is defined as Edge sensitive, Non-Sticky as Level-sensitive.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.28 Beep Generation

The **Beep Generation** command causes the Beep Generator Widget to beep or to stop beeping. The intended use of this command is for BIOS POST beeps, not for generating high quality audio output. Note that the Beep Generator Widget may have an optional output amplifier as defined in the "Audio Widget Capabilities" parameter (Section 5.3.4.6). If this amp is present, it is controlled with the normal amplifier controls.

Command Options:

Table 5-30. Beep Generation

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F0Ah	0	Bits 31:8 are *Reserved* Bits 7:0 are Divider
Set	70Ah	Bits 7:0 are Divider	0

Divider indicates the divisor for the 48-kHz link frame tick used to generate the beep.

The beep frequency generated is the result of dividing the 48-kHz link frame clock by 4 times the number specified in Divider, allowing tones from 47 Hz to 12 kHz. When a non-zero data value is sent to enable the Beep, the codec generates the beep tone on all Pin Complexes that are currently configured as outputs. The codec may either disable any normal streaming on the output pins, or it may mix the beep with the active output streams depending on the design of the codec.

A value of 00h in bits Divider disables internal PC Beep generation and enables normal operation of the codec.

The generated signal is not intended to be a high quality sine wave. The clock output rounded with a capacitor provides sufficient signal quality to provide beep code signaling.

Applies to:

- Beep Generator Widget

5.3.3.29 Volume Knob

Volume Knob provides the controls for an optional external hardware volume control.

Command Options:

Table 5-31. Volume Knob Control

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F0Fh	0	Bit 7 is Direct Bits 6:0 is Volume
Set	70Fh	Bit 7 is Direct Bits 6:0 is Volume	0

Direct causes the Volume Control Widget to directly control the hardware volume of the slave amplifiers. If “Direct” is set to a 0, the volume control will not directly affect the volume of the slaves amplifiers; rather, the software receives an unsolicited response, reads the volume control, and then programs the appropriate amplifiers correctly.

Volume is specified in steps, as is amplifier gain. If two amplifiers slaved to the Volume Knob control have different “StepSize” parameters, they are both adjusted by the same number of steps, implying a differing total dB treatment. If the Volume Knob control has more steps than a slave amplifier is capable of supporting (as indicated in the Volume Knob Capabilities parameter), the amplifier remains at its limiting value.

Applies to:

- Volume Knob Widget

5.3.3.30 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.

Figure 5-20. Subsystem ID Register

31:8	7:0
Subsystem ID	Assembly ID

Command Options:

Table 5-32. Subsystem ID

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

NOTES:
1. The Verb Codes F21h, F22h, and F23h are reserved for the Subsystem ID 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

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.3.31 Configuration Default

The **Configuration Default** is a 32-bit register required in each Pin Widget. It is used by software as an aid in determining the configuration of jacks and devices attached to the codec. At the time the codec is first powered on, this register is internally loaded with default values indicating the typical system use of this particular pin/jack. After this initial loading, it is completely codec opaque, and its state, including any software writes into the register, must be preserved across reset events. Its state need not be preserved across power level changes.

Command Options:

Table 5-33. Configuration Default

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F1Ch[1]	0	Config bits [31:0]
Set 1	71Ch	Config bits [7:0]	0
Set 2	71Dh	Config bits [15:8]	0
Set 3	71Eh	Config bits [23:16]	0
Set 4	71Fh	Config bits [31:24]	0

NOTES:
1. The Verb Codes F1Dh, F1Eh, and F1Fh are reserved for the Configuration Default 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 F1Ch.

Data Structure:

The Configuration Default register is defined as shown in Figure 5-21.

Figure 5-21. Configuration Data Structure

31:30	29:24	23:20	19:16	15:12	11:8	7:4	3:0
Port Connectivity	Location	Default Device	Connection Type	Color	Misc	Default Association	Sequence

Port Connectivity[1:0] indicates the external connectivity of the Pin Complex. Software can use this value to know what Pin Complexes are connected to jacks, internal devices, or not connected at all. The encodings of the Port Connectivity field are defined in Table 5-34

Location[5:0] indicates the physical location of the jack or device to which the pin complex is connected. This allows software to indicate, for instance, that the device is the “Front Panel Headphone Jack” as opposed to rear panel connections. The encodings of the Location field are defined in Table 5-35.

The Location field is divided into two pieces, the upper bits [5:4] and the lower bits [3:0]. The upper bits [5:4] provide a gross location, such as “External” (on the primary system chassis, accessible to the user), “Internal” (on the motherboard, not accessible without opening the box), on a separate chassis (such as a mobile box), or other.

The lower bits [3:0] provide a geometric location, such as “Front,” “Left,” etc., or provide special encodings to indicate locations such as mobile lid mounted microphones. An “x” in Table 5-35 indicates a combination that software should support. While all combinations are permitted, they are not all likely or expected.

Default Device[3:0] indicates the intended use of the jack or device. This can indicate either the label on the jack or the device that is hardwired to the port, as with integrated speakers and the like. The encodings of the Default Device field are defined in Table 5-36.

Connection Type[3:0] indicates the type of physical connection, such as a 1/8-inch stereo jack or an optical digital connector, etc. Software can use this information to provide helpful user interface descriptions to the user or to modify reported codec capabilities based on the capabilities of the physical transport external to the codec. The encodings of the Connection Type field are defined in Table 5-37.

Color[3:0] indicates the color of the physical jack for use by software. Encodings for the Color field are defined in Table 5-37.

Misc[3:0] is a bit field used to indicate other information about the jack. Currently, only bit 0 is defined. If bit 0 is set, it indicates that the jack has no presence detect capability, so even if a Pin Complex indicates that the codec hardware supports the presence detect functionality on the jack, the external circuitry is not capable of supporting the functionality. The bit definitions for the Misc field are in Table 5-39.

Default Association and **Sequence** are used together by software to group Pin Complexes (and therefore jacks) together into functional blocks to support multichannel operation. Software may assume that all jacks with the same association number are intended to be grouped together, for instance to provide six channel analog output. The Default Association can also be used by software to prioritize resource allocation in constrained situations. Lower Default Association values would be higher in priority for resources such as processing nodes or Input and Output Converters. Note that this is the default association only, and software can override this value if

required, in particular if the user provides additional information about the particular system configuration. A value of 0000b is reserved and should not be used. Software may interpret this value to indicate that the Pin Configuration data has not been properly initialized. A value of 1111b is a special value indicating that the Association has the lowest priority. Multiple different Pin Complexes may share this value, and each is intended to be exposed as independent devices.

Sequence indicates the order of the jacks in the association group. The lowest numbered jack in the association group should be assigned the lowest numbered channels in the stream, etc. The numbers need not be sequential within the group, only the order matters. Sequence numbers within a set of Default Associations must be unique.

Table 5-34. Port Connectivity

Port Connectivity	Encoding
The Port Complex is connected to a jack (1/8", ATAPI, etc.).	00b
No physical connection for Port.	01b
A fixed function device (integrated speaker, integrated mic, etc.) is attached.	10b
Both a jack and an internal device are attached. The Information provided in all other fields refers to the integrated device. The PD pin will reflect the status of the jack; the user will need to be queried to figure out what it is.	11b

Table 5-35. Location

Bits [5:4]	Bits [3:0]										
	0h: N/A	1h: Rear	2h: Front	3h: Left	4h: Right	5h: Top	6h: Bottom	7h: Special	8h: Special	9h: Special	Ah-Fh: Reserved
00b: External on primary chassis	x	x	x	x	x	x	x	x (Rear panel)	x (Drive bay)		
01b: Internal	x							x (riser)	x (HDMI)	x (ATAPI)	
10b: Separate chassis	x	x	x	x	x	x	x				
11b: Other	x						x	x (Mobile Lid–Inside) (e.g., mic inside mobile lid)	x (Mobile Lid–Outside)		

Table 5-36. Default Device

Default Device	Encoding
Line Out	0h
Speaker	1h
HP Out	2h
CD	3h
SPDIF Out	4h
Digital Other Out	5h
Modem Line Side	6h
Modem Handset Side	7h
Line In	8h
AUX	9h
Mic In	Ah
Telephony	Bh
SPDIF In	Ch
Digital Other In	Dh
Reserved	Eh
Other	Fh

Table 5-37. Connection Type

Connection Type	Encoding
Unknown	0h
1/8" stereo/mono	1h
1/4" stereo/mono	2h
ATAPI internal	3h
RCA	4h
Optical	5h
Other Digital	6h
Other Analog	7h
Multichannel Analog (DIN)	8h
XLR/Professional	9h
RJ-11 (Modem)	Ah
Combination	Bh
Other	Fh

Table 5-38. Color

Color	Encoding
Unknown	0h
Black	1h
Grey	2h
Blue	3h
Green	4h
Red	5h
Orange	6h
Yellow	7h
Purple	8h
Pink	9h
Reserved	A-Dh
White	Eh
Other	Fh

Table 5-39. Miscellaneous

Miscellaneous	Bit
Reserved	3
Reserved	2
Reserved	1
Jack Detect Override	0

Applies to:

Pin Widget

5.3.3.32 Stripe Control

The **Stripe Control** verb reports and controls the strip capability of an Audio Output Converter. This verb needs to be implemented only for an Audio Output Converter, and only if the Stripe bit of the Audio Widget Capabilities parameter is a 1.

Command Options:

Table 5-40. Stripe Control

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Get	F24h	0	Stripe bits [31:0]
Set 1	720h	Stripe bits [7:0]	0

Data Structure:

The Stripe Control register is defined as shown in Figure 5-22.

Figure 5-22. Stripe Control Register

31:23	22:20	7:2	1:0
Reserved	Stripe Capability [2:0]	*Reserved*	Stripe Control [1:0]

- **Stripe Capability[2:0]** is a bit field indicating on which **SDO** lines the Audio Output Converter Widget may receive data from the controller. Bit 0 should always be 1, indicating that the widget can receive data on **SDI**[0]. If bit 1 is set, it indicates that the widget can receive stream data on **SDI**[0:1]. If bit 2 is set, the widget can receive stream data on **SDI**[0:3]. These bits must be dynamically updated by the codec based on which SDI lines have a connection between the codec and the controller. This can be determined by the codec by observing which **SDO** lines toggle when commands are sent.
- **Stripe Control [1:0]** is an encoded field controlling which **SDO** lines a stream will be transmitted on. An Audio Output Converter Widget uses this information to know which **SDO** lines it should decode the output stream. This field should be programmed by software to match the Stripe Control field of the Stream Descriptor Control.

Applies to:

- Output Converter

5.3.3.33 Function Reset

The **Function Reset** command causes the functional unit, and all widgets associated with the functional unit, to return to their power-on reset values. Note that some controls such as the Configuration Default controls should not be reset with this command. It is also possible that certain other controls, such as Caller-ID, should not be reset.

This command does not affect the Link interface logic, which must be reset with the link **RST#** signal. Therefore, a codec must not initiate a Status Change request on the link.

When a codec receives the Function reset verb, the expected behavior is that the codec will issue a response to the verb to acknowledge receipt, and then reset the affected Function Group controls. The codec must be ready to respond to the verbs on the Link frame after the frame on which it returns its response to the Reset command.

Command Options:

Table 5-41. Function Reset

	Verb ID	Payload (8 Bits)	Response (32 Bits)
Execute	7FFh	0	0

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.4 Parameters

Parameters are static, Read Only values in the codec used to convey codec, function, and node capabilities to the software.

5.3.4.1 Vendor ID

Vendor ID returns the Device and Vendor ID values for the codec. These are each 16-bit values used to identify the codec to the PnP subsystem. Note that this is the first parameter read in enumerating a codec (root node) and as such will be used to determine the possible presence of multiple SDI signals (refer to Section 5.3.2). In the case of multiple SDIs with a single root node, the response to this parameter fetch must always be returned on the SDI which the codec designates as "primary," regardless of which CAd the request was generated on.

Parameter ID: 00h

Response Format:

Figure 5-23. Vendor ID Response Format

31:16	15:0
Vendor ID	Device ID

Applies to:

- Root Node

5.3.4.2 Revision ID

The **Revision ID** parameter returns the codec revision ID value for the codec. This is an 8-bit value used to identify the codec to the PnP subsystem.

Parameter ID: 02h

Response Format:

Figure 5-24. Revision ID Response Format

31:24	23:20	19:16	15:8	7:0
Rsvd	MajRev	MinRev	Revision ID	Stepping ID

MajRev (4 bits) is the major revision number (left of the decimal) of the *Intel® High Definition Audio Specification* to which the codec is fully compliant.

MinRev (4 bits) is the minor revision number (right of the decimal) or "dot number" of the *Intel® High Definition Audio Specification* to which the codec is fully compliant.

Revision ID (8 bits) is the vendor's revision number for this given Device ID.

Stepping ID (8 bits) is an optional vendor stepping number within the given Revision ID.

Applies to:

- Root Node

5.3.4.3 Subordinate Node Count

The **Subordinate Node Count** parameter provides information about the function group nodes associated with the codec (root node), as well as the widget nodes associated with a function group.

For a Root Node, the “Total Number of Nodes” parameter is the number of Function Group nodes in the codec, the starting address of which is provided by the “Starting Node Number” parameter.

For a Function Group node, the “Total Number of Nodes” parameter is the number of widget or functional nodes in the Functional Group, the starting address of which is provided by the “Starting Node Number” parameter.

Parameter ID: 04h

Response Format:

Figure 5-25. Subordinate Node Count Response Format

31:24	23:16	15:8	7:0
Reserved	Starting Node Number	Reserved	Total Number of Nodes

Applies to:

- Root Node
- Audio Function
- Modem Function Group
- Other Function Group

5.3.4.4 Function Group Type

The **Function Group Type** parameter returns a value describing what the type of node is being addressed. This parameter is primarily useful for identifying the type of Function Group a node represents (such as Audio versus Modem) but can also be used to identify the type of “Other” or vendor specific nodes.

Parameter ID: 05h

Response Format:

Figure 5-26. Function Group Type Response Format

31:9	8	7:0
Reserved	UnSol Capable	NodeType

UnSol Capable indicates (when = 1) that this node is capable of generating an unsolicited response and will respond to the Unsolicited Response verb Section 5.3.3.14). Note that if this node does not generate unsolicited responses, subordinate nodes (widgets) may still do so.

Table 5-42. Node Type

Node Type	Encoding
Reserved	00h
Audio Function Group	01h
Vendor Defined Modem Function Group	02h
Reserved	03-7Fh
Vendor defined Function Group	80-FFh

As shown, the upper 128 type encodings may be used to identify vendor specific nodes other than audio widgets. Vendor specific Audio Widgets – i.e., those vendor defined widgets that are enumerated hierarchically within an Audio Function Group – do not have this parameter, but must be identified as a "Vendor Defined Audio Widget" (type = Fh) in the "Audio Widget Capabilities" parameter (see Section 5.3.4.6), since their type will be queried by the audio function driver using that parameter.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Groups
- Vendor Defined Widgets outside an Audio Function Group

5.3.4.5 Audio Function Group Capabilities

The **Audio Parameter** returns a set of bit fields describing the audio capabilities of the Audio Function.

Parameter ID: 08h

Response Format:

Figure 5-27. Audio Function Group Capabilities Response Format

31:17	16	15:12	11:8	7:4	3:0
Rsvd	Beep Gen	*Rsvd*	Input Delay	*Rsvd*	Output Delay

BeepGen indicates the presence of an optional Beep Generator with this Audio Function Group (refer to Section 5.2.3.8).

Input Delay is a 4-bit value representing the number of samples between when the sample is received as an analog signal at the pin and when the digital representation is transmitted on the Intel High Definition Audio Link. This may be a "typical" value. If this is 0, the widgets along the critical path should be queried, and each individual widget must report its individual delay.

Output Delay is a four bit value representing the number of samples between when the sample is received from the Link and when it appears as an analog signal at the pin. This may be a "typical" value. If this is 0, the widgets along the critical path should be queried, and each individual widget must report its individual delay.

Applies to:

- Audio Function Group

5.3.4.6 Audio Widget Capabilities

The Audio Capabilities control returns a set of bit fields describing the audio capabilities of the widget.

Parameter ID: 09h

Response Format:

Figure 5-28. Audio Widget Capabilities Response Format

31:24	23:20	19:16	15:12	11	10	9	8	7	6	5	4	3	2	1	0
Rsvd	Type	Delay	*Rsvd*	L-R Swap	Power Cntrl	Digital	Conn List	Unsol Capable	Proc Widget	Stripe	Format Override	Amp Param Over-ride	Out Amp Present	In Amp Present	Stereo

Type defines the functionality of the widget node. This is an enumerated list.

Table 5-43. Widget Type

Widget Type	Encoding
Audio Output	0h
Audio Input	1h
Audio Mixer	2h
Audio Selector	3h
Pin Complex	4h
Power Widget	5h
Volume Knob Widget[1]	6h
Beep Generator Widget[2]	7h
Reserved	8h-Eh
Vendor defined audio widget	Fh

NOTES:
1. In the case of the Volume Knob Widget, none of the parameter bits [19:0] are used and may be omitted or set to 0. However, software assumes the capability of unsolicited responses and a connection list, as these are required by this widget type.
2. In the case of the Beep Generator Widget, the only meaningful parameter bits are 2 (“Out Amp Present”) and 3 (“Amp Param Override”). None of the other parameter bits are used and may be omitted or set to 0.

Any vendor defined widget that is enumerated hierarchically within an Audio Function Group must be identified as a vendor defined type (Fh) using this parameter.

Delay indicates the number of sample delays through the widget. This may be 0 if the delay value in the Audio Function Parameters is supplied to represent the entire path.

L-R Swap indicates the capability of swapping the left and right channels through the Audio Widget. If the Audio Widget is both input and output capable (e.g., Pin Widget), then swapping must be supported for both input and output paths. Default (0) is no swap capability.

PowerCntrl indicates that the Power State control is supported on this widget. This allows finer grained power management than just at the Function Group level for widgets which support it.

Digital indicates that a widget supports a digital stream. If the bit is a 1, it is a digital widget. For an Input or Output converter, for instance, this means the widget is translating between the Intel High Definition Audio Link and a digital format such as S/P-DIF or I2S rather than analog data.

ConnList indicates whether a connection list is present on the widget. If the bit is a 1, the Connection List Length parameter and the Connection List Entry controls should be queried to discover the input connections. This bit must be a 1 for Input Converters, Sum Widgets, and Selector Widgets. The bit may be a 0 for Output Converters if the only connection for the widget is to the Intel High Definition Audio Link.

If **Unsol Capable** is a 1, the audio widget supports unsolicited responses. The Unsolicited Response command can be used to configure and enable Unsolicited Response generation. When this parameter is associated with a Pin Widget, then setting this bit requires that the Pin Widget generate an unsolicited response (when enabled) whenever the “Presence Detect” bit (see Section 5.3.3.15) changes state.

If **ProcWidget** is a 1, the “Processing Controls” parameter should be queried for more information about the widget’s processing controls.

The **Stripe** bit defines whether the Audio Output Converter Widget supports striping. If Stripe is a 1, the Audio Output Converter Widget must support the Stripe Control verb. Stripe is only defined for Audio Output Converter Widgets; for all other widget types, this bit must be 0.

If **Format Override** is a 1, the widget contains format information, and the “Supported Formats” and “Supported PCM Bits, Rates” should be queried for the widget’s format capabilities. If this bit is a 0, then the Audio Function node must contain default amplifier parameters, and that node’s format related parameters should be queried to determine the format parameters. This bit is not applicable to, and is always 0 for, Pin Complex Widgets.

If **Amp Param Override** is a 1, the widget contains its own amplifier parameters. If this bit is a 0, then the Audio Function node must contain default amplifier parameters, and they should be used to define all amplifier parameters (both input and output) in this widget.

If **(In|Out) AmpPresent** is a 1, the widget contains an input or output amplifier, as indicated. The Amp Param Override bit should be examined to determine whether the widget contains default amplifier parameters or has amplifier parameters that need to be explicitly queried. The “In Amp Present” bit is only relevant for Sum Widgets, Input Converters, and Pin Complexes. The “Out Amp Present” bit is only relevant for Selector Widgets, Sum Widgets, Output Converter Widgets, and Pin Complex Widgets.

The **Stereo** bit determines if the widget is a stereo or mono widget. If the “Stereo” bit is a 1, the widget is a stereo widget.

Applies to:

- Input Converter Widget
- Output Converter Widget
- Selector Widget
- Mixer Widget
- Pin Widget

5.3.4.7 Supported PCM Size, Rates

The Supported Rates parameter returns a bit field describing the bit depth and sample rate capabilities of the widget when PCM formatted data is being rendered or captured.

Parameter ID: 0Ah

Response Format:

Figure 5-29. Supported PCM Size, Rates Return Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
rsvd	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	*rsvd*	B32	B24	B20	B16	B8
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rsvd	*rsvd*	*rsvd*	*rsvd*	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1

Table 5-44. Bit Depth and Sample Rate (Sheet 1 of 2)

Bit	Meaning	
B32	32-bit audio formats are supported	
B24	24-bit audio formats are supported	
B20	20-bit audio formats are supported	
B16	16-bit audio formats are supported	
B8	8-bit audio formats are supported	
	Rate (kHz)	**Mult/Div**
R1	8.0	1/6 * 48
R2	11.025	1/4 * 44.1
R3	16.0	1/3 * 48
R4	22.05	1/2 * 44.1
R5	32.0	2/3 * 48
R6	44.1	
R7	48.0	(Must be supported by all codecs)
R8	88.2	2/1 * 44.1
R9	96.0	2/1 * 48

Table 5-44. Bit Depth and Sample Rate (Sheet 2 of 2)

R10	176.4	4/1 * 44.1
R11	192.0	4/1 * 48
R12	384	8/1 * 48

Applies to:

- Audio Function Group (as default for all widgets within the Audio Function)
- Audio Input Converter
- Audio Output Converter

5.3.4.8 Supported Stream Formats

The Supported Stream Formats parameter returns a bit field describing the format capabilities of the widget.

Parameter ID: 0Bh

Response Format:

Figure 5-30. Supported Stream Formats Response Format

31:3	2	1	0
rsvd	AC3	Float32	PCM

The **PCM** bit set indicates that the widget supports PCM formatted data. The PCM formats supported are determined using the “Supported PCM Bits, Rates” parameter.

The **Float32** bit indicates that Float32 formatted data is supported. The “Supported PCM Bits, Rates” determine the sample rates, but only 32-bit data is supported.

The **AC3** bit indicates that Dolby* AC-3 formatted data is supported. The “Supported PCM Bits, Rates” parameter determines the sample rates, but only 16-bit data is supported.

Applies to:

- Audio Function Group (as default for all widgets within the Audio Function)
- Audio Input Converter
- Audio Output Converter

5.3.4.9 Pin Capabilities

The Pin Capabilities parameter returns a bit field describing the capabilities of the Pin Complex Widget.

Parameter ID: 0Ch

Response Format:

Figure 5-31. Pin Capabilities Response Format

31:17	16	15:8	7	6	5	4	3	2	1	0
Rsvd	EAPD Capable	VRef Control	*Rsvd*	Balanced I/O Pins	Input Capable	Output Capable	Headphone Drive Capable	Presence Detect Capable	Trigger Req'd	Impedance Sense Capable

EAPD Capable indicates the codec has an EAPD pin and that this Pin Widget provides support for controlling that pin.

VRef Control[7:0] is a bit field used to indicate what voltages may be produced on the associated VRef pin(s). If all bits in the bit field are 0, then VRef generation is not supported by the Pin Complex. Also, if the Input Capable bit is a 0, then the VRef bit field has no meaning and all bits must be 0.

If the Output Capable bit and any bits in the VRef field are set, then bit 0 (Hi-Z) must also be set to indicate that the VRef signal can be turned off to support output devices.

Figure 5-32 describes the VRef bit field. A 1 in any position indicates that the associated signal level is supported. All values of VRef are specified as a percentage of the analog voltage rail, AVdd.

Figure 5-32. VRef Bit Field

7:6	5	4	3	2	1	0
Rsvd	100%	80%	*Rsvd*	Ground	50%	Hi-Z

Balanced I/O Pins indicates that the Pin Complex Widget has balanced pins.

Input Capable indicates whether the pin complex supports input. If Input Capable is a 1, the pin is capable of input.

Output Capable indicates whether the pin complex supports output. If Output Capable is a 1, the pin is capable of output.

Headphone Drive Capable indicates that the pin has an amplifier with sufficient current drive to drive headphones. If Output Capable is a 0, then this bit has no meaning and must be 0.

Presence Detect Capable indicates whether the pin complex can perform presence detect to determine whether there is anything plugged in. Presence detect does not indicate *what* is plugged in, only that *something* is plugged in.

Trigger Required indicates whether a trigger is required for an impedance measurement (see Section 5.3.3.15).

Impedance Sense Capable indicates whether the pin complex supports impedance sense on the attached peripheral to determine what it is. More accurate (possibly sequenced) forms of peripheral discrimination may be supported independent of this capability; however, if this bit is a 1, then the codec must support at least the basic impedance test as described in Section 5.3.3.15

Applies to:

- Pin Widget

5.3.4.10 Amplifier Capabilities

The "Amplifier Properties" parameters return the parameters for the input or the output amplifier on a node. In the case of a Pin Widget, the terms input and output are relative to the codec itself; for all other widgets, these terms are relative to the node. The amplifier capabilities are indicated by the step size of the amplifier, the number of steps, the offset of the range with respect to 0 dB, and whether the amplifier supports mute.

Parameter ID: 0Dh for Input amplifiers; 12h for Output amplifiers

Response Format:

Figure 5-33. Amplifier Capabilities Response Format

31	30:23	22:16	15	14:8	7	6:0
Mute Capable	*Rsvd*	StepSize	*Rsvd*	NumSteps	*Rsvd*	Offset

StepSize (7 bits) indicates the size of each step in the gain range. Each individual step may be 0-32 dB specified in 0.25-dB steps. A value of 0 indicates 0.25-dB steps, while a value of 127d indicates a 32-dB step.

NumSteps (7 bits) indicates the number of steps in the gain range. There may be from 1 to 128 steps in the amplifier gain range. (0d means 1 step, 127d means 128 steps). A value of 0 (1 step) means that the gain is fixed and may not be changed.

Offset (7 bits) indicates which step is 0 dB. If there are two or more steps, one of the step values must correspond to a value of 0 dB. The "Offset" value reflects the value which, if programmed in to the Amplifier Gain control, would result in a gain of 0 dB.

Mute Capable (1 bit) reports if the respective amplifier is capable of muting. Muting implies a –infinity gain (no sound passes), but the actual performance is determined by the hardware.

Applies to:

- Audio Function Group (as default for all widgets within the Audio Function)
- Audio Input Converter
- Audio Output Converter
- Mixer Widget
- Selector Widget
- Pin Widget
- Other Widget

5.3.4.11 Connection List Length

Returns the length of the connection list for the widget.

Parameter ID: 0Eh

Response Format:

Figure 5-34. Connection List Length Response Format

31:8	7	6:0
Reserved	Long Form	Connection List Length

Long Form indicates whether the items in the connection list are long form or short form as described in Section 5.1.2.

Connection List Length is an integer indicating the number of items in the connection list. If Connection List Length is 1, there is only one hard-wired input possible, which is read from the Connection List, and there is no Connection Select Control (see Section 5.3.3.2).

Applies to:

- Audio Input Converter
- Mixer Widget
- Selector Widget
- Pin Widget
- Power Widget

5.3.4.12 Supported Power States

Returns a bit field describing the power states supported by the functional unit and widgets.

Parameter ID: 0Fh

Response Format:

Figure 5-35. Supported Power States Response Format

31:8	3	2	1	0
Reserved	D3Sup	D2Sup	D1Sup	D0Sup

Applies to:

- Audio Function
- Modem Function
- Other Functions
- Power Widget
- Other Widgets (optional)

5.3.4.13 Processing Capabilities

Parameter ID: 10h

Response Format:

Figure 5-36. Processing Capabilities Response Format

31:8	15:8	7:1	0
Reserved	NumCoeff	0	Benign

Benign will be a 1 if the processing on the widget can be placed in a mode which is linear and time invariant, such as equalization or speaker compensation processing. The Processing control can cause the processing to enter this state.

NumCoeff will indicate the number of coefficients to be loaded on the widget if the processing widget supports loadable parameters. If loadable coefficients are not supported, this value must be 0.

Applies to:

- Input Converter
- Output Converter
- Selector Widget
- Pin Complex
- Other Widget

5.3.4.14 GP I/O Count

Parameter ID: 11h

Response Format:

Figure 5-37. GP I/O Capabilities Response Format

31	30	29:24	23:16	15:8	7:0
GPIWake	*GPIUnsol*	*Reserved*	NumGPIs	NumGPOs	NumGPIOs

GPIOWake will be reported as 1 if the GPIs and GPIOs configured as inputs can cause a wake when there is a change in level on the pin by generating a Status Change event on the link when the link **RST#** is enabled. This capability must be enabled using the “GPI Wake Enable Mask” or ‘GPIO Wake Enable Mask’ controls.

GPIOUnsol will be reported as 1 if the GPIs and GPIOs configured as inputs can cause an Unsolicited Response to be generated when there is a change in level on the pin. This capability must be enabled using the “GPI Unsolicited Enable Mask” or “GPIO Unsolicited Enable Mask” controls.

NumGPIs is an integer representing the number of GPI pins supported by the function. There may be from 0 to 7 GPI bits in the function.

NumGPOs is an integer representing the number of GPOs supported by the function. There may be from 0 to 7 GPO bits in the function.

NumGPIOs is an integer representing the number of GPIOs supported by the function. There may be from 0 to 7 GPIO bits in the function.

Applies to:

- Audio Function Group
- Modem Function Group
- Other Function Group

5.3.4.15 Volume Knob Capabilities

Parameter ID: 13h

Response Format:

Figure 5-38. Volume Knob Capabilities Response Format

31:8	7	6:0
Reserved	Delta	NumSteps

Delta indicates if software can write a base volume to the Volume Control Knob. If this bit is a 1, software can write to the volume control; any volume changes in HW from that point are a delta from the written value. If this bit is a 0, then the volume is absolute (e.g., from a pot) and software cannot modify the value.

NumSteps is the total number of steps in the range of the volume knob. This is similar to the NumSteps parameter of amplifiers, but there is not a StepSize specified.

Applies to:

- Volume Knob Widget

5.3.5 Vendor Defined Verbs

Codec vendors may choose to use additional verbs not defined in this specification to provide additional functionality. A vendor may not use a vendor-defined verb to provide functionality that a specification defined verb already provides; the specification defined verb should always be used whenever possible.

A vendor may use verb encodings of F70 through FFF to implement vendor defined verbs on defined Audio Widget types.

On vendor defined widget types, such as modem widgets, any verb encoding that does not conflict with required verbs, such as the “Get Parameter” verb, may be used.

5.3.6 Required Parameter and Control Support

Table 5-45 specifies which parameters are required (R) for each specification-defined node. It also indicates optional (o) parameters which are used to declare the presence of optional features in the associated node. A shaded square in the table indicates that the subject parameter is not applicable to the subject node type. The squares marked with (a) indicate an alternative; the parameter is required in either the Audio Function Group (AFG), to be used as a default, or else in all of the indicated widgets. If these parameters are present in the AFG, they are only needed in the individual widgets that have non-default capabilities. Some parameters are marked with an asterisk (*) for the “Vendor_Specific_Audio_Widget indicating they are not required by the specification since a vendor specific node may largely define its own parameters. If, however, the vendor specific node implements features that can be defined by an existing parameter, then using the standard parameter is preferable to defining a new one.

Table 5-45. Required Support for Parameters (Sheet 1 of 2)

Required Parameter Support	Parameter ID	Root Node	Audio Function Group	Modem Function Group	Vendor Defined Function Group	Audio Output Converter	Audio Input Converter	Pin Complex Widget	Mixer (SumAmp)	Selector (Mux)	Power Widget	Volume Knob	Beep Generator	Vendor Defined Wdiget
Vendor ID	00	R												
Revision ID	02	R												
Subordinate Node Count	04	R	R	R	R									
Function Group Type	05		R	R	R									
Audio Function Group Capabilities	08		o											
Audio Widget Capabilities	09					R	R	R	R	R	R	R	R	R
Sample Size, Rate CAPs	0A		A			A	a							*
Stream Formats	0B		A			A	a							*
Pin Capabilities	0C							R						*
Input Amp Capabilities	0D		A				a	a	a	a				*
Output Amp Capabilities	12		A			a		a	a	a				*
Connection List Length	0E						R	R	R	R	R			*
Supported Power States	0F		R	R	o	o	o	o	o	o	R			*

Table 5-45. Required Support for Parameters (Sheet 2 of 2)

Processing Capabilities	10					o	o	o		o				*
GPI/O Count	11		o	o	o									
Volume Knob Capabilities	13											R		

Note that the Audio Function Group Capabilities parameter provides a default delay for the entire AFG to be used in lieu of adding specific delays listed for each widget in the Audio Widget Capabilities parameter. This is required if one or more widgets in the AFG opts to not report a correct delay in its Audio Widget Capabilities parameter; if all widgets do report an accurate delay number, the Audio Function Group Capabilities parameter is not required.

Table 5-46 specifies which verbs and controls are required (R) for each specification-defined node. It also indicates conditional (c) verbs which are required only if the respective optional capability is declared to be available. Another conditional verb (X) is required when the codec supports multiple **SDI** signals. A shaded square in the table indicates that the subject verb is not applicable to the subject node type. Some parameters are marked with an asterisk (*) for the "Vendor_Specific_Audio_Widget" indicating they are not required by the specification since a vendor specific node may largely define its own verbs. If, however, the vendor specific node implements controls that can be accessed with an existing verb, then using the standard verb is preferable to defining a new one.

Table 5-46. Required Support for Verbs (Sheet 1 of 2)

Required Verb Support	Get Code	Set Code	Root Node	Audio Function Group	Modem Function Group	Vendor Defined Function Group	Audio Output Converter	Audio Input Converter	Pin Complex Widget	Mixer (SumAmp)	Selector (Mux)	Power Widget	Volume Knob	Beep Generator	Vendor Defined Wdiget
Get Parameter	F00		R	R	R	R	R	R	R	R	R	R	R	R	R
Connection Select	F01	701						c	c		c				*
Get Connection List Entry	F02							R	R	R	R	R	R		*
Processing State	F03	##					c	c	c		c				*
Coefficient Index	D- -	5- -					c	c	c		c				*
Processing Coefficient	C- -	4- -					c	c	c		c				*
Amplifier Gain/Mute	B- -	3- -					c	c	c	c	c			c	*
Stream Format	A- -	2- -					R	R							*
Digital Converter 1	F0D	70D					c	c							*
Digital Converter 2	F0D	70E					c	c							*
Power State	F05	705		R	R	c	c	c	c	c	c	R			c
Channel/Stream ID	F06	706					R	R							*
SDI Select	F04	704					X	X							*

Table 5-46. Required Support for Verbs (Sheet 2 of 2)

Required Verb Support	Get Code	Set Code	Root Node	Audio Function Group	Modem Function Group	Vendor Defined Function Group	Audio Output Converter	Audio Input Converter	Pin Complex Widget	Mixer (SumAmp)	Selector (Mux)	Power Widget	Volume Knob	Beep Generator	Vendor Defined Wdiget
Pin Widget Control	F07	707							R						*
Unsolicited Enable	F08	708					c	c	c	c	c	c	c		*
Pin Sense	F09	709							c						*
EAPD/BTL Enable	F0C	70C							c						*
All GPI Controls	F10 thru F1A	710 thru 71A		c	c										
Beep Generation Control	F0A	70A												R	
Volume Knob Control	F0F	70F											R		
Subsystem ID, Byte 0	F20	720		R	R	R									
Subsystem ID, Byte 1	F20	721		R	R	R									
Subsystem ID, Byte 2	F20	722		R	R	R									
Subsystem ID, Byte 3	F20	723		R	R	R									
Config Default, Byte 0	F1C	71C							R						
Config Default, Byte 1	F1C	71D							R						
Config Default, Byte 2	F1C	71E							R						
Config Default, Byte 3	F1C	71F							R						
Stripe Control	F24	724					c								
RESET		7FF		R	R	R									

Note that the Connection Select control is not required when the Connection List Length Register value is 1 for this node. In that case, there is no Connection Select control.

§

6 Intel® High Definition Audio Audio Controller Registers (D27:F0)

The Intel High Definition Audio controller resides in PCI Device 27, Function 0 on bus 0. This function contains a set of DMA engines that are used to move samples of digitally encoded data between system memory and external codecs.

Note: All registers in this function (including memory-mapped registers) must be addressable in byte, word, and dword quantities. The software must always make register accesses on natural boundaries (i.e. dword accesses must be on dword boundaries; word accesses on word boundaries, etc.) In addition, the memory-mapped register space must not be accessed with the LOCK semantic exclusive-access mechanism. If software attempts exclusive-access mechanisms to the Intel High Definition Audio memory-mapped space, the results are undefined.

Note: Users interested in providing feedback on the *Intel® High Definition Audio Specification* or planning to implement the *Intel® High Definition Audio Specification* into a future product will need to execute the *Intel® High Definition Audio Specification Developer's Agreement*. For more information, send an e-mail request to nextgenaudio@intel.com.

6.1 Intel® High Definition Audio Audio PCI Configuration Space (High Definition Audio— D27:F0)

Note: Address locations that are not shown should be treated as Reserved.

Table 6-1. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 1 of 3)

Offset	Mnemonic	Register Name	Default	Access
00-01h	VID	Vendor Identification	8086h	RO
02-03h	DID	Device Identification	2668h	RO
04-05h	PCICMD	PCI Command	0000h	R/W, RO
06-07h	PCISTS	PCI Status	0010h	R/WC, RO
08h	RID	Revision Identification	See register description.	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	02h	RO
0Bh	BCC	Base Class Code	04h	RO
0Ch	CLS	Cache Line Size	00h	R/W
0Dh	LT	Latency Timer	00h	RO
0Eh	HEADTYP	Header Type	00h	RO

Table 6-1. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 2 of 3)

10-13h	AZBARL	High Definition Audio Lower Base Address (Memory)	00000004h	R/W, RO
14-17h	AZBARU	High Definition Audio Upper Base Address (Memory)	00000000h	R/W
2C-2Dh	SVID	Subsystem Vendor Identification	0000h	R/WO
2E-2Fh	SID	Subsystem Identification	0000h	R/WO
34h	CAPPTR	Capability List Pointer	50h	RO
3Ch	INTLN	Interrupt Line	00h	R/W
3Dh	INTPN	Interrupt Pin	See Register Description	RO
40h	AZCTL	High Definition Audio Control	00h	R/W, RO
44h	TCSEL	Traffic Class Select	00h	R/W
50-51h	PID	PCI Power Management Capability ID	6001h	RO
52-53h	PC	Power Management Capabilities	C842	RO
54-57h	PCS	Power Management Control and Status	00000000h	R/W, RO, R/WC
60-61h	MID	MSI Capability ID	7005h	RO
62-63h	MMC	MSI Message Control	0080h	R/W, RO
64-67h	MMLA	MSI Message Lower Address	00000000h	R/W, RO
68-6Bh	MMUA	SMI Message Upper Address	00000000h	R/W
6C-6Dh	MMD	MSI Message Data	0000h	R/W
70-71h	PXID	PCI Express Capability Identifiers	0010h	RO
72-73h	PXC	PCI Express Capabilities	0001h	RO
74-77h	DEVCAP	Device Capabilities	00000000h	RO
78-79h	DEVC	Device Control	0800h	R/W, RO
7A-7Bh	DEVS	Device Status	0010h	RO
100-103h	VCCAP	Virtual Channel Enhanced Capability Header	13010002h	RO
104-107h	PVCCAP1	Port VC Capability Register 1	00000001h	RO
108-10Bh	PVCCAP2	Port VC Capability Register 2	00000000h	RO
10C-10D	PVCCTL	Port VC Control	0000h	RO
10E-10Fh	PVCSTS	Port VC Status	0000h	RO
110-103h	VC0CAP	VC0 Resource Capability	00000000h	RO
114-117h	VC0CTL	VC0 Resource Control	800000FFh	R/W, RO
11A-11Bh	VC0STS	VC0 Resource Status	0000h	RO
11C-11Fh	VCiCAP	VCi Resource Capability	00000000h	RO
120-123h	VCiCLT	VCi Resource Control	00000000h	R/W, RO
126-127h	VCiSTS	VCi Resource Status	0000h	RO
130-133h	RCCAP	Root Complex Link Declaration Enhanced Capability Header	00010005h	RO
134-137h	ESD	Element Self Description	05000100h	RO

Table 6-1. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 3 of 3)

140-143h	L1DESC	Link 1 Description	00000001h	RO
148-14Bh	L1ADDL	Link 1 Lower Address	See Register Description	RO
14C-14Fh	L1ADDU	Link 1 Upper Address	See Register Description	RO

6.1.1 VID—Vendor Identification Register (High Definition Audio Controller—D27:F0)

Offset: 00-01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

Bit	Description
15:0	**Vendor ID** — RO. This is a 16-bit value assigned to Intel. Intel VID = 8086h

6.1.2 DID—Device Identification Register (High Definition Audio Controller—D27:F0)

Offset Address: 02–03h
Default Value: 2668h
Attribute: RO
Size: 16 bits

Bit	Description
15:0	Device ID — RO. This is a 16-bit value assigned to the Intel® ICH6 Intel® High Definition Audio Controller.

6.1.3 PCICMD—PCI Command Register (High Definition Audio Controller—D27:F0)

Offset Address: 04–05h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

Bit	Description
15:11	Reserved
10	**Interrupt Disable (ID)** — R/W. 0= The INTx# signals may be asserted. 1= The Intel® High Definition Audio controller's INTx# signal will be de-asserted Note that this bit does not affect the generation of MSI's.
9	Fast Back to Back Enable (FBE) — RO. Not implemented. Hardwired to 0.
8	SERR# Enable (SERR_EN) — RO. Not implemented. Hardwired to 0.
7	Wait Cycle Control (WCC) — RO. Not implemented. Hardwired to 0.
6	Parity Error Response (PER) — RO. Not implemented. Hardwired to 0.
5	VGA Palette Snoop (VPS). Not implemented. Hardwired to 0.

Bit	Description
4	Memory Write and Invalidate Enable (MWIE) — RO. Not implemented. Hardwired to 0.
3	Special Cycle Enable (SCE). Not implemented. Hardwired to 0.
2	**Bus Master Enable (BME)** — R/W. Controls standard PCI Express bus mastering capabilities for Memory and I/O, reads and writes. Note that this bit also controls MSI generation since MSI's are essentially Memory writes. 0 = Disable 1 = Enable
1	**Memory Space Enable (MSE)** — R/W. Enables memory space addresses to the Intel High Definition Audio Audio controller. 0 = Disable 1 = Enable
0	I/O Space Enable (IOSE)—RO. Hardwired to 0 since the Intel High Definition Audio controller does not implement I/O space.

6.1.4 PCISTS—PCI Status Register (High Definition Audio Controller—D27:F0)

Offset Address: 06–07h
Default Value: 0010h
Attribute: RO, R/WC
Size: 16 bits

Bit	Description
15	Detected Parity Error (DPE) —RO. Not implemented. Hardwired to 0.
14	SERR# Status (SERRS) — RO. Not implemented. Hardwired to 0.
13	Received Master Abort (RMA) — R/WC. Software clears this bit by writing a 1 to it. 0 = No master abort received. 1 = The Intel® High Definition Audio controller sets this bit when, as a bus master, it receives a master abort. When set, the High Definition Audio controller clears the run bit for the channel that received the abort.
12	Received Target Abort (RTA) — RO. Not implemented. Hardwired to 0.
11	Signaled Target Abort (STA) — RO. Not implemented. Hardwired to 0.
10:9	DEVSEL# Timing Status (DEV_STS) — RO. Does not apply. Hardwired to 0.
8	Data Parity Error Detected (DPED) — RO. Not implemented. Hardwired to 0.
7	Fast Back to Back Capable (FB2BC) — RO. Does not apply. Hardwired to 0.
6	Reserved.
5	66 MHz Capable (66MHZ_CAP) — RO. Does not apply. Hardwired to 0.
4	Capabilities List (CAP_LIST) — RO. Hardwired to 1. Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h.
3	Interrupt Status (IS) — RO. 0 = This bit is 0 after the interrupt is cleared. 1 = This bit is 1 when the INTx# is asserted. Note that this bit is not set by an MSI.
2:0	Reserved.

6.1.5 RID—Revision Identification Register (High Definition Audio Controller—D27:F0)

Offset: 08h Attribute: RO
Default Value: See bit description Size: 8 Bits

Bit	Description
7:0	Revision ID — RO. Refer to the Intel® ICH6/ICH6R/ICH6-M EDS Specification Update for the value of the Revision ID Register

6.1.6 PI—Programming Interface Register (High Definition Audio Controller—D27:F0)

Offset: 09h Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	Programming Interface — RO.

6.1.7 SCC—Sub Class Code Register (High Definition Audio Controller—D27:F0)

Address Offset: 0Ah Attribute: RO
Default Value: 02h Size: 8 bits

Bit	Description
7:0	Sub Class Code (SCC) — RO. 02h = Audio Device

6.1.8 BCC—Base Class Code Register (High Definition Audio Controller—D27:F0)

Address Offset: 0Bh Attribute: RO
Default Value: 04h Size: 8 bits

Bit	Description
7:0	Base Class Code (BCC) — RO. 04h = Multimedia device

6.1.9 CLS—Cache Line Size Register (High Definition Audio Controller—D27:F0)

Address Offset: 0Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

Bit	Description
7:0	Cache Line Size— R/W. Implemented as R/W register, but has no functional impact to the ICH6

6.1.10 LT—Latency Timer Register (High Definition Audio Controller—D27:F0)

Address Offset: 0Dh
Default Value: 00h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	Latency Timer— RO. Hardwired to 00

6.1.11 HEADTYP—Header Type Register (High Definition Audio Controller—D27:F0)

Address Offset: 0Eh
Default Value: 00h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	Header Type— RO. Hardwired to 00.

6.1.12 AZBARL—Intel® High Definition Audio Lower Base Address Register (High Definition Audio Controller—D27:F0)

Address Offset: 10h
Default Value: 00000004h
Attribute: R/W, RO
Size: 32 bits

Bit	Description
31:14	**Lower Base Address (LBA)**— R/W. Base address for the Intel® High Definition Audio controller's memory mapped configuration registers. 16 Kbytes are requested by hardwiring bits 13:4 to 0's.
13:4	RO. Hardwired to 0's
3	Prefetchable (PREF)—RO. Hardwired to 0 to indicate that this BAR is NOT prefetchable
2:1	Address Range (ADDRNG)—RO. Hardwired to 10b, indicating that this BAR can be located anywhere in 64-bit address space.
0	Space Type (SPTYP)—RO. Hardwired to 0. Indicates this BAR is located in memory space.

6.1.13 AZBARU—Intel® High Definition Audio Upper Base Address Register (High Definition Audio Controller—D27:F0)

Address Offset:	14h	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits

Bit	Description
31:0	**Upper Base Address (UBA)**— R/W. Upper 32 bits of the Base address for the Intel® High Definition Audio controller's memory mapped configuration registers.

6.1.14 SVID—Subsystem Vendor Identification Register (High Definition Audio Controller—D27:F0)

Address Offset:	2C–2Dh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits

The SVID register, in combination with the Subsystem ID register (D27:F0:2Eh), enable the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	Subsystem Vendor ID — R/WO.

6.1.15 SID—Subsystem Identification Register (High Definition Audio Controller—D27:F0)

Address Offset:	2E–2Fh	Attribute:	R/WO
Default Value:	0000h	Size:	16 bits

The SID register, in combination with the Subsystem Vendor ID register (D27:F0:2Ch) make it possible for the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

Bit	Description
15:0	**Subsystem ID** — R/WO.

6.1.16 CAPPTR—Capabilities Pointer Register (Audio—D30:F2)

Address Offset: 34h
Default Value: 50h
Attribute: RO
Size: 8 bits

This register indicates the offset for the capability pointer.

Bit	Description
7:0	Capabilities Pointer (CAP_PTR) — RO. This field indicates that the first capability pointer offset is offset 50h (Power Management Capability)

6.1.17 INTLN—Interrupt Line Register (High Definition Audio Controller—D27:F0)

Address Offset: 3Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

Bit	Description
7:0	**Interrupt Line (INT_LN)** — R/W. This data is not used by the ICH6. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

6.1.18 INTPN—Interrupt Pin Register (High Definition Audio Controller—D27:F0)

Address Offset: 3Dh
Default Value: See Description
Attribute: RO
Size: 8 bits

Bit	Description
7:4	Reserved.
3:0	Interrupt Pin— RO. This reflects the value of D27IP.ZIP (Chipset Config Registers: Offset 3110h:bits 3:0).

6.1.19 AZCTL—Intel® High Definition Audio Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 40h
Default Value: 00h
Attribute: R/W, RO
Size: 8 bits

Bit	Description
7:4	Reserved.
3	**BITCLK Detect Clear (CLKDETCLR)**— R/W. 0 = When a 0 is written to this bit, the clock detect circuit is operational and maybe enabled. 1 = Writing a 1 to this bit clears bit 1 (CLKDET#) in this register. CLKDET# bit remains clear when this bit is set to 1. **NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition.
2	**BITCLK Detect Enable (CLKDETEN)**— R/W. 0 = Latches the current state of bit 1 (CLKDET#) in this register 1 = Enables the clock detection circuit **NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition.
1	BITCLK Detected Inverted (CLKDET#)— RO. This bit is modified by hardware. It is set to 0 when the ICH6 detects that the BITCLK is toggling, indicating the presence of an AC '97 codec on the link **NOTE:** Bit 2 (CLKDETEN) and bit 3 (CLKDETCLR) in this register control the operation of this bit and must be manipulated correctly in order to get a valid CLKDET# indicator.
0	**High Definition Audio/AC '97 Signal Mode**— R/W. This bit selects the shared Intel® High Definition Audio/AC '97 signals. 0 = AC '97 mode is selected (Default) 1 = High Definition Audio mode is selected **NOTES:** 1. This bit has no affect on the visibility of the Intel High Definition Audio and AC '97 function configuration space. 2. This bit is in the resume well and only clear on a power-on reset. Software must not makes assumptions about the reset state of this bit and must set it appropriately.

6.1.20 TCSEL—Traffic Class Select Register (High Definition Audio Controller—D27:F0)

Address Offset:	44h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

This register assigned the value to be placed in the TC field. CORB and RIRB data will always be assigned TC0.

Bit	Description
7:3	Reserved.
2:0	**High Definition Audio Traffic Class Assignment (TCSEL)**— R/W. This register assigns the value to be placed in the Traffic Class field for input data, output data, and buffer descriptor transactions. 000 = TC0 001 = TC1 010 = TC2 011 = TC3 100 = TC4 101 = TC5 110 = TC6 111 = TC7 Note: These bits are not reset on $D3_{HOT}$ to D0 transition; however, they are reset by PLTRST#.

6.1.21 PID—PCI Power Management Capability ID Register (High Definition Audio Controller—D27:F0)

Address Offset:	50h	Attribute:	RO
Default Value:	6001h	Size:	16 bits

Bit	Description
15:8	Next Capability (Next)— RO. Hardwired to 60h. Points to the next capability structure (MSI)
7:0	Cap ID (CAP)— RO. Hardwired to 01h. Indicates that this pointer is a PCI power management capability.

6.1.22 PC—Power Management Capabilities Register (High Definition Audio Controller—D27:F0)

Address Offset: 52h
Default Value: C842h
Attribute: RO
Size: 16 bits

Bit	Description
15:11	PME Support— RO. Harwired to 11001b. Indicates PME# can be generated from D3 and D0 states.
10	D2 Support— RO. Hardwired to 0. Indicates that D2 state is not supported.
9	D1 Support—RO. Harwired to 0. Indicates that D1 state is not supported.
8:6	Aux Current—RO. Hardwired to 001b. Reports 55mA maximum suspend well current required when in the $D3_{COLD}$ state.
5	Device Specific Initialization (DSI)—RO. Hardwired to 0. Indicates that no device specific initialization is required.
4	Reserved
3	PME Clock (PMEC)—RO. Does not apply. Hardwired to 0.
2:0	Version—RO. Hardwired to 010b. Indicates support for version 1.1 of the PCI Power Management Specification.

6.1.23 PCS—Power Management Control and Status Register (High Definition Audio Controller—D27:F0)

Address Offset: 54h
Default Value: 00000000h
Attribute: RO, R/W, R/WC
Size: 32 bits

Bit	Description
31:24	Data—RO. Does not apply. Hardwired to 0.
23	Bus Power/Clock Control Enable— RO. Does not apply. Hardwired to 0.
22	B2/B3 Support—RO. Does not apply. Harwired to 0.
21:16	Reserved.
15	**PME Status (PMES)**—R/WC. 0 = Software clears the bit by writing a 1 to it. 1 = This bit is set when the Intel® High Definition Audio controller would normally assert the PME# signal independent of the state of the PME_EN bit (bit 8 in this register) This bit in the resume well and only cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately
14:9	Reserved
8	**PME Enable (PMEE)**—R/W. 0 = Disable 1 = when set and if corresponding PMES also set, the Intel High Definition Audio controller sets the AC97_STS bit in the GPE0_STS register (PMBASE +28h). The AC97_STS bit is shared by AC ’97 and Intel High Definition Audio functions since they are mutually exclusive. This bit in the resume well and only cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately
7:2	Reserved

Bit	Description
1:0	**Power State (PS)**—R/W. This field is used both to determine the current power state of the Intel High Definition Audio controller and to set a new power state. 00 = D0 state 11 = $D3_{HOT}$ state Others = reserved Note: - If software attempts to write a value of 01b or 10b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs. - When in the $D3_{HOT}$ states, the Intel High Definition Audio controller's configuration space is available, but the IO and memory space are not. Additionally, interrupts are blocked. - When software changes this value from $D3_{HOT}$ state to the D0 state, an internal warm (soft) reset is generated, and software must re-initialize the function.

6.1.24 MID—MSI Capability ID Register (High Definition Audio Controller—D27:F0)

Address Offset: 60h Attribute: RO
Default Value: 7005h Size: 16 bits

Bit	Description
15:8	Next Capability (Next)— RO. Harwired to 70h. Points to the PCI Express capability structure.
7:0	Cap ID (CAP)— RO. Hardwired to 05h. Indicates that this pointer is a MSI capability

6.1.25 MMC—MSI Message Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 62h Attribute: RO, R/W
Default Value: 0080h Size: 16 bits

Bit	Description
15:8	Reserved
7	64b Address Capability (64ADD)— RO. Hardwired to 1. Indicates the ability to generate a 64-bit message address
6:4	Multiple Message Enable (MME)—RO. Normally this is a R/W register. However since only 1 message is supported, these bits are hardwired to 000 = 1 message.
3:1	Multiple Message Capable (MMC)—RO. Hardwired to 0 indicating request for 1 message.
0	**MSI Enable (ME)**—R/W. 0 = an MSI may not be generated 1 = an MSI will be generated instead of an INTx signal.

6.1.26 MMLA—MSI Message Lower Address Register (High Definition Audio Controller—D27:F0)

Address Offset: 64h
Default Value: 00000000h
Attribute: RO, R/W
Size: 32 bits

Bit	Description
31:2	**Message Lower Address (MLA)**— R/W. Lower address used for MSI message.
1:0	Reserved.

6.1.27 MMUA—MSI Message Upper Address Register (High Definition Audio Controller—D27:F0)

Address Offset: 68h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31:0	**Message Upper Address (MUA)**— R/W. Upper 32-bits of address used for MSI message.

6.1.28 MMD—MSI Message Data Register (High Definition Audio Controller—D27:F0)

Address Offset: 6Ch
Default Value: 0000h
Attribute: R/W
Size: 16 bits

Bit	Description
15:0	**Message Data (MD)**— R/W. Data used for MSI message.

6.1.29 PXID—PCI Express* Capability ID Register (High Definition Audio Controller—D27:F0)

Address Offset: 70h
Default Value: 0010h
Attribute: RO
Size: 16 bits

Bit	Description
15:8	Next Capability (Next)— RO. Hardwired to 0. Indicates that this is the last capability structure in the list.
7:0	Cap ID (CAP)—RO. Hardwired to 10h. Indicates that this pointer is a PCI Express capability structure

6.1.30 PXC—PCI Express* Capabilities Register (High Definition Audio Controller—D27:F0)

Address Offset: 72h
Default Value: 0001h
Attribute: RO
Size: 16 bits

Bit	Description
15:14	Reserved
13:9	Interrupt Message Number (IMN)—RO. Hardwired to 0.
8	Slot Implemented (SI)—RO. Hardwired to 0.
7:4	Device/Port Type (DPT)—RO. Hardwired to 0. Indicates that this is a PCI Express Endpoint device.
3:0	Capability Version (CV)—RO. Hardwired to 0001b. Indicates version #1 PCI Express capability

6.1.31 DEVCAP—Device Capabilities Register (High Definition Audio Controller—D27:F0)

Address Offset: 74h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:28	Reserved
27:26	Captured Slot Power Limit Scale (SPLS)—RO. Hardwired to 0.
25:18	Captured Slot Power Limit Value (SPLV)—RO. Hardwired to 0.
17:15	Reserved
14	Power Indicator Present —RO. Hardwired to 0.
13	Attention Indicator Present—RO. Hardwired to 0.
12	Attention Button Present—RO. Hardwired to 0.
11:9	Endpoint L1 Acceptable Latency—RO.
8:6	Endpoint L0s Acceptable Latency—RO.
5	Extended Tag Field Support—RO. Hardwired to 0. Indicates 5-bit tag field support
4:3	Phantom Functions Supported—RO. Hardwired to 0. Indicates that phantom functions not supported
2:0	Max Payload Size Supported—RO. Hardwired to 0. Indicates 128B maximum payload size capability

6.1.32 DEVC—Device Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 78h
Default Value: 0800h
Attribute: R/W, RO
Size: 16 bits

Bit	Description
15	Reserved
14:12	Max Read Request Size—RO. Hardwired to 0 enabling 128B maximum read request size.
11	**No Snoop Enable (NSNPEN)**—R/W. 0 = The Intel® High Definition Audio controller will not set the No Snoop bit. In this case, isochronous transfers will not use VC1 (VCi) even if it is enabled since VC1 is never snooped. Isochronous transfers will use VC0. 1 = The Intel High Definition Audio controller is permitted to set the No Snoop bit in the Requester Attributes of a bus master transaction. In this case, VC0 or VC1 may be used for isochronous transfers. Note: This bit is not reset on $D3_{HOT}$ to D0 transition; however, it is reset by PLTRST#.
10	Auxiliary Power Enable—RO. Hardwired to 0, indicating that Intel High Definition Audio device does not draw AUX power
9	Phantom Function Enable—RO. Hardwired to 0 disabling phantom functions.
8	Extended Tag Field Enable—RO. Hardwired to 0 enabling 5-bit tag.
7:5	Max Payload Size—RO. Hardwired to 0 indicating 128B.
4	Enable Relaxed Ordering—RO. Hardwired to 0 disabling relaxed ordering.
3	Unsupported Request Reporting Enable—RO. Not implemented. Hardwired to 0.
2	Fatal Error Reporting Enable—RO. Not implemented. Hardwired to 0.
1	Non-Fatal Error Reporting Enable—RO. Not implemented. Hardwired to 0.
0	Correctable Error Reporting Enable—RO. Not implemented. Hardwired to 0.

6.1.33 DEVS—Device Status Register (High Definition Audio Controller—D27:F0)

Address Offset: 7Ah
Default Value: 0010h
Attribute: RO
Size: 16 bits

Bit	Description
15:6	Reserved
5	Transactions Pending—RO. 0 = Indicates that completions for all non-posted requests have been received 1 = Indicates that Intel® High Definition Audio controller has issued non-posted requests which have not been completed.
4	AUX Power Detected—RO. Hardwired to 1 indicating the device is connected to resume power
3	Unsupported Request Detected—RO. Not implemented. Hardwired to 0.
2	Fatal Error Detected—RO. Not implemented. Hardwired to 0.
1	Non-Fatal Error Detected—RO. Not implemented. Hardwired to 0.
0	Correctable Error Detected—RO. Not implemented. Hardwired to 0.

6.1.34 VCCAP—Virtual Channel Enhanced Capability Header (High Definition Audio Controller—D27:F0)

Address Offset: 100h
Default Value: 13010002h
Attribute: RO
Size: 32 bits

Bit	Description
31:20	Next Capability Offset—RO. Hardwired to 130h. Points to the next capability header, which is the Root Complex Link Declaration Enhanced Capability Header.
19:16	Capability Version—RO. Hardwired to 1h.
15:0	PCI Express Extended Capability—RO. Hardwired to 0002h.

6.1.35 PVCCAP1—Port VC Capability Register 1 (High Definition Audio Controller—D27:F0)

Address Offset: 104h
Default Value: 00000001h
Attribute: RO
Size: 32 bits

Bit	Description
31:12	Reserved.
11:10	Port Arbitration Table Entry Size—RO. Hardwired to 0 since this is an endpoint device.
9:8	Reference Clock—RO. Hardwired to 0 since this is an endpoint device.
7	Reserved.
6:4	Low Priority Extended VC Count—RO. Hardwired to 0. Indicates that only VC0 belongs to the low priority VC group
3	Reserved.
2:0	Extended VC Count—RO. Hardwired to 001b. Indicates that 1 extended VC (in addition to VC0) is supported by Intel® High Definition Audio controller.

6.1.36 PVCCAP2—Port VC Capability Register 2 (High Definition Audio Controller—D27:F0)

Address Offset: 108h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:24	VC Arbitration Table Offset—RO. Hardwired to 0 indicating that a VC arbitration table is not present.
23:8	Reserved.
7:0	VC Arbitration Capability—RO. Hardwired to 0. These bits are not applicable since the Intel® High Definition Audio controller reports a 0 in the Low Priority Extended VC Count bits in the PVCCAP1 register.

6.1.37 PVCCLT—Port VC Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 10Ch
Default Value: 0000h
Attribute: RO
Size: 16 bits

Bit	Description
15:4	Reserved.
3:1	VC Arbitration Select—RO. Hardwired to 0. Normally these bits are R/W. However, these bits are not applicable since the Intel® High Definition Audio controller reports a 0 in the Low Priority Extended VC Count bits in the PVCCAP1 register
0	Load VC Arbitration Table—RO. Hardwired to 0 since an arbitration table is not present.

6.1.38 PVCSTS—Port VC Status Register (High Definition Audio Controller—D27:F0)

Address Offset: 10Eh
Default Value: 0000h
Attribute: RO
Size: 16 bits

Bit	Description
15:1	Reserved.
0	VC Arbitration Table Status—RO. Hardwired to 0 since an arbitration table is not present.

6.1.39 VC0CAP—VC0 Resource Capability Register (High Definition Audio Controller—D27:F0)

Address Offset: 110h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:24	Port Arbitration Table Offset—RO. Hardwired to 0 since this field is not valid for endpoint devices
23	Reserved.
22:16	Maximum Time Slots—RO. Hardwired to 0 since this field is not valid for endpoint devices
15	Reject Snoop Transactions—RO. Hardwired to 0 since this field is not valid for endpoint devices.
14	Advanced Packet Switching—RO. Hardwired to 0 since this field is not valid for endpoint devices
13:8	Reserved.
7:0	Port Arbitration Capability—RO. Hardwired to 0 since this field is not valid for endpoint devices

6.1.40 VC0CTL—VC0 Resource Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 114h
Default Value: 800000FFh
Attribute: R/W, RO
Size: 32 bits

Bit	Description
31	VC0 Enable—RO. Hardwired to 1 for VC0.
30:27	Reserved.
26:24	VC0 ID—RO. Hardwired to 0 since the first VC is always assigned as VC0
23:20	Reserved.
19:17	Port Arbitration Select—RO. Hardwired to 0 since this field is not valid for endpoint devices
16	Load Port Arbitration Table—RO. Hardwired to 0 since this field is not valid for endpoint devices
15:8	Reserved.
7:0	**TC/VC0 Map**—R/W, RO. Bit 0 is hardwired to 1 since TC0 is always mapped VC0. Bits [7:1] are implemented as R/W bits.

6.1.41 VC0STS—VC0 Resource Status Register (High Definition Audio Controller—D27:F0)

Address Offset: 11Ah
Default Value: 0000h
Attribute: RO
Size: 16 bits

Bit	Description
15:2	Reserved.
1	VC0 Negotiation Pending—RO. Hardwired to 0 since this bit does not apply to the integrated Intel® High Definition Audio device
0	Port Arbitration Table Status—RO. Hardwired to 0 since this field is not valid for endpoint devices

6.1.42 VCiCAP—VCi Resource Capability Register (High Definition Audio Controller—D27:F0)

Address Offset: 11Ch
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:24	Port Arbitration Table Offset—RO. Hardwired to 0 since this field is not valid for endpoint devices.
23	Reserved.
22:16	Maximum Time Slots—RO. Hardwired to 0 since this field is not valid for endpoint devices
15	Reject Snoop Transactions—RO. Hardwired to 0 since this field is not valid for endpoint devices
14	Advanced Packet Switching—RO. Hardwired to 0 since this field is not valid for endpoint devices
13:8	Reserved
7:0	Port Arbitration Capability—RO. Hardwired to 0 since this field is not valid for endpoint devices

6.1.43 VCiCTL—VCi Resource Control Register (High Definition Audio Controller—D27:F0)

Address Offset: 120h
Default Value: 00000000h
Attribute: R/W, RO
Size: 32 bits

Bit	Description
31	**VCi Enable**—R/W. 0 = VCi is disabled 1 = VCi is enabled Note: This bit is not reset on $D3_{HOT}$ to D0 transition; however, it is reset by PLTRST#.
30:27	Reserved.
26:24	**VCi ID**—R/W. This field assigns a VC ID to the VCi resource. This field is not used by the ICH6 hardware, but it is R/W to avoid confusing software.
23:20	Reserved.
19:17	Port Arbitration Select—RO. Hardwired to 0 since this field is not valid for endpoint devices
16	Load Port Arbitration Table—RO. Hardwired to 0 since this field is not valid for endpoint devices
15:8	Reserved.
7:0	**TC/VCi Map**—R/W, RO. This field indicates the TCs that are mapped to the VCi resource. Bit 0 is hardwired to 0 indicating that it cannot be mapped to VCi. Bits [7:1] are implemented as R/W bits. This field is not used by the ICH6 hardware, but it is R/W to avoid confusing software.

6.1.44 VCiSTS—VCi Resource Status Register (High Definition Audio Controller—D27:F0)

Address Offset: 126h
Default Value: 0000h
Attribute: RO
Size: 16 bits

Bit	Description
15:2	Reserved.
1	VCi Negotiation Pending—RO. Does not apply. Hardwired to 0.
0	Port Arbitration Table Status—RO. Hardwired to 0 since this field is not valid for endpoint devices.

6.1.45 RCCAP—Root Complex Link Declaration Enhanced Capability Header Register (High Definition Audio Controller—D27:F0)

Address Offset: 130h
Default Value: 00010005h
Attribute: RO
Size: 32 bits

Bit	Description
31:20	Next Capability Offset—RO. Hardwired to 0 indicating this is the last capability.
19:16	Capability Version—RO. Hardwired to 1h.
15:0	PCI Express Extended Capability ID—RO. Hardwired to 0005h.

6.1.46 ESD—Element Self Description Register (High Definition Audio Controller—D27:F0)

Address Offset: 134h
Default Value: 05000100h
Attribute: RO
Size: 32 bits

Bit	Description
31:24	Port Number—RO. Hardwired to 05h indicating that Intel® High Definition Audio controller is assigned as Port #5.
23:16	Component ID—RO. This field returns the value of the ESD.CID field of the chip configuration section. ESD.CID is programmed by BIOS.
15:8	Number of Link Entries—RO. The Intel High Definition Audio only connects to one device, the ICH6 egress port. Therefore this field reports a value of 1h.
7:4	Reserved.
3:0	Element Type (ELTYP)—RO. The Intel High Definition Audio controller is an integrated Root Complex Device. Therefore, the field reports a value of 0h.

6.1.47 L1DESC—Link 1 Description Register (High Definition Audio Controller—D27:F0)

Address Offset: 140h
Default Value: 00000001h
Attribute: RO
Size: 32 bits

Bit	Description
31:24	Target Port Number—RO. The Intel® High Definition Audio controller targets the ICH6's RCRB Egress port, which is Port #0.
23:16	Target Component ID—RO. This field returns the value of the ESD.CID field of the chip configuration section. ESD.CID is programmed by BIOS.
15:2	Reserved.
1	Link Type—RO. Hardwired to 0 indicating Type 0.
0	Link Valid—RO. Hardwired to 1.

6.1.48 L1ADDL—Link 1 Lower Address Register (High Definition Audio Controller—D27:F0)

Address Offset: 148h
Default Value: See Register Description
Attribute: RO
Size: 32 bits

Bit	Description
31:14	Link 1 Lower Address—RO. Hardwired to match the RCBA register value in the PCI-LPC bridge (D31:F0:F0h).
13:0	Reserved.

6.1.49 L1ADDU—Link 1 Upper Address Register (High Definition Audio Controller—D27:F0)

Address Offset: 14Ch
Default Value: See Register Description
Attribute: RO
Size: 32 bits

Bit	Description
31:0	Link 1 Upper Address—RO. Hardwired to match the RCBA register value in the PCI-LPC bridge (D31:F0:F0h).

6.2 Intel® High Definition Audio Memory Mapped Configuration Registers (High Definition Audio— D27:F0)

The base memory location for these memory mapped configuration registers is specified in the AZBAR register (D27:F0:offset 10h and D27:F0:offset 14h). The individual registers are then accessible at AZBAR + Offset as indicated in the following table.

These memory mapped registers must be accessed in byte, word, or dword quantities.

Table 6-2. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 1 of 4)

AZBAR + Offset	Mnemonic	Register Name	Default	Access
00-01h	GCAP	Global Capabilities	4401h	RO
02h	VMIN	Minor Version	00h	RO
03h	VMAJ	Major Version	01h	RO
04-05h	OUTPAY	Output Payload Capability	003Ch	RO
06-07h	INPAY	Input Payload Capability	001Dh	RO
08-0Bh	GCTL	Global Control	00000000h	R/W
0C-0Dh	WAKEEN	Wake Enable	0000h	R/W
0E-0Fh	STATESTS	State Change Status	0000h	R/WC
10-11h	GSTS	Global Status	0000h	R/WC
20-23h	INTCTL	Interrupt Control	00000000h	R/W
24-27h	INTSTS	Interrupt Status	00000000h	RO
30-33h	WALCLK	Wall Clock Counter	00000000h	RO
34-37h	SYNC	Stream Synchronization	00000000h	R/W
40-43h	CORBLBASE	CORB Lower Base Address	00000000h	R/W, RO
44-47h	CORBUBASE	CORB Upper Base Address	00000000h	R/W
4A-4Bh	CORBRP	CORB Read Pointer	0000h	R/W
4Ch	CORBCTL	CORB Control	00h	R/W
4Dh	CORBST	CORB Status	00h	R/WC

Table 6-2. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 2 of 4)

4Eh	CORBSIZE	CORB Size	42h	RO
50-53h	RIRBLBASE	RIRB Lower Base Address	00000000h	R/W, RO
54-57h	RIRBUBASE	RIRB Upper Base Address	00000000h	R/W
58-59h	RIRBWP	RIRB Write Pointer	0000h	R/W, RO
5A-5Bh	RINTCNT	Response Interrupt Count	0000h	R/W
5Ch	RIRBCTL	RIRB Control	00h	R/W
5Dh	RIRBSTS	RIRB Status	00h	R/WC
5Eh	RIRBSIZE	RIRB Size	42h	RO
60-63h	IC	Immediate Command	00000000h	R/W
64-67h	IR	Immediate Response	00000000h	RO
68-69h	IRS	Immediate Command Status	0000h	R/W, R/WC
70-73h	DPLBASE	DMA Position Lower Base Address	00000000h	R/W, RO
74-77h	DPUBASE	DMA Position Upper Base Address	00000000h	R/W
80-82h	ISD0CTL	Input Stream Descriptor 0 (ISD0) Control	040000h	R/W, RO
83h	ISD0STS	ISD0 Status	00h	R/WC, RO
84-87h	ISD0LPIB	ISD0 Link Position in Buffer	00000000h	RO
88-8Bh	ISD0CBL	ISD0 Cyclic Buffer Length	00000000h	R/W
8C-8Dh	ISD0LVI	ISD0 Last Valid Index	0000h	R/W
8E-8F	ISD0FIFOW	ISD0 FIFO Watermark	0004h	R/W
90-91h	ISD0FIFOS	ISD0 FIFO Size	005Fh	RO
92-93h	ISD0FMT	ISD0 Format	0000h	R/W
98-9Bh	ISD0BDPL	ISD0 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
9C-9Fh	ISD0BDPU	ISD0 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
A0-A2h	ISD1CTL	Input Stream Descriptor 1(ISD01) Control	040000h	R/W, RO
A3h	ISD1STS	ISD1 Status	00h	R/WC, RO
A4-A7h	ISD1LPIB	ISD1 Link Position in Buffer	00000000h	RO
A8-ABh	ISD1CBL	ISD1 Cyclic Buffer Length	00000000h	R/W
AC-ADh	ISD1LVI	ISD1 Last Valid Index	0000h	R/W
AE-AFh	ISD1FIFOW	ISD1 FIFO Watermark	0004h	R/W
B0-B1h	ISD1FIFOS	ISD1 FIFO Size	005Fh	RO
B2-B3h	ISD1FMT	ISD1 Format	0000h	R/W
B8-BBh	ISD1BDPL	ISD1 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
BC-BFh	ISD1BDPU	ISD1 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
C0-C2h	ISD2CTL	Input Stream Descriptor 2 (ISD2) Control	040000h	R/W, RO
C3h	ISD2STS	ISD2 Status	00h	R/WC, RO
C4-C7h	ISD2LPIB	ISD2 Link Position in Buffer	00000000h	RO

Table 6-2. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 3 of 4)

C8-CBh	ISD2CBL	ISD2 Cyclic Buffer Length	00000000h	R/W
CC-CDh	ISD2LVI	ISD2 Last Valid Index	0000h	R/W
CE-CFh	ISD1FIFOW	ISD1 FIFO Watermark	0004h	R/W
D0-D1h	ISD2FIFOS	ISD2 FIFO Size	005Fh	RO
D2-D3h	ISD2FMT	ISD2 Format	0000h	R/W
D8-DBh	ISD2BDPL	ISD2 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
DC-DFh	ISD2BDPU	ISD2 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
E0-E2h	ISD3CTL	Input Stream Descriptor 3 (ISD3) Control	040000h	R/W, RO
E3h	ISD3STS	ISD3 Status	00h	R/WC, RO
E4-E7h	ISD3LPIB	ISD3 Link Position in Buffer	00000000h	RO
E8-EBh	ISD3CBL	ISD3 Cyclic Buffer Length	00000000h	R/W
EC-EDh	ISD3LVI	ISD3 Last Valid Index	0000h	R/W
EE-EFh	ISD3FIFOW	ISD3 FIFO Watermark	0004h	R/W
F0-F1h	ISD3FIFOS	ISD3 FIFO Size	005Fh	RO
F2-F3h	ISD3FMT	ISD3 Format	0000h	R/W
F8-FBh	ISD3BDPL	ISD3 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
FC-FFh	ISD3BDPU	ISD3 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
100-102h	OSD0CTL	Output Stream Descriptor 0 (OSD0) Control	040000h	R/W, RO
103h	OSD0STS	OSD0 Status	00h	R/WC, RO
104-107h	OSD0LPIB	OSD0 Link Position in Buffer	00000000h	RO
108-10Bh	OSD0CBL	OSD0 Cyclic Buffer Length	00000000h	R/W
10C-10Dh	OSD0LVI	OSD0 Last Valid Index	0000h	R/W
10E-10Fh	OSD0FIFOW	OSD0 FIFO Watermark	0004h	R/W
110-111h	OSD0FIFOS	OSD0 FIFO Size	00BFh	R/W
112-113h	OSD0FMT	OSD0 Format	0000h	R/W
118-11Bh	OSD0BDPL	OSD0 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
11C-11Fh	OSD0BDPU	OSD0 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
120-122h	OSD1CTL	Output Stream Descriptor 1 (OSD1) Control	040000h	R/W, RO
123h	OSD1STS	OSD1 Status	00h	R/WC, RO
124-127h	OSD1LPIB	OSD1 Link Position in Buffer	00000000h	RO
128-12Bh	OSD1CBL	OSD1 Cyclic Buffer Length	00000000h	R/W
12C-12Dh	OSD1LVI	OSD1 Last Valid Index	0000h	R/W
12E-12Fh	OSD1FIFOW	OSD1 FIFO Watermark	0004h	R/W
130-131h	OSD1FIFOS	OSD1 FIFO Size	00BFh	R/W
132-133h	OSD1FMT	OSD1 Format	0000h	R/W

Table 6-2. Intel® High Definition Audio Audio PCI Register Address Map (High Definition Audio—D27:F0) (Sheet 4 of 4)

138-13Bh	OSD1BDPL	OSD1 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
13C-13Fh	OSD1BDPU	OSD1 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
140-142h	OSD2CTL	Output Stream Descriptor 2 (OSD2) Control	040000h	R/W, RO
143h	OSD2STS	OSD2 Status	00h	R/WC, RO
144-147h	OSD2LPIB	OSD2 Link Position in Buffer	00000000h	RO
148-14Bh	OSD2CBL	OSD2 Cyclic Buffer Length	00000000h	R/W
14C-14Dh	OSD2LVI	OSD2 Last Valid Index	0000h	R/W
14E-14Fh	OSD2FIFOW	OSD2 FIFO Watermark	0004h	R/W
150-151h	OSD2FIFOS	OSD2 FIFO Size	00BFh	R/W
152-153h	OSD2FMT	OSD2 Format	0000h	R/W
158-15Bh	OSD2BDPL	OSD2 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
15C-15Fh	OSD2BDPU	OSD2 Buffer Description List Pointer-Upper Base Address	00000000h	R/W
160-162h	OSD3CTL	Output Stream Descriptor 3 (OSD3) Control	040000h	R/W, RO
163h	OSD3STS	OSD3 Status	00h	R/WC, RO
164-167h	OSD3LPIB	OSD3 Link Position in Buffer	00000000h	RO
168-16Bh	OSD3CBL	OSD3 Cyclic Buffer Length	00000000h	R/W
16C-16Dh	OSD3LVI	OSD3 Last Valid Index	0000h	R/W
16E-16Fh	OSD3FIFOW	OSD3 FIFO Watermark	0004h	R/W
170-171h	OSD3FIFOS	OSD3 FIFO Size	00BFh	R/W
172-173h	OSD3FMT	OSD3 Format	0000h	R/W
178-17Bh	OSD3BDPL	OSD3 Buffer Descriptor List Pointer-Lower Base Address	00000000h	R/W, RO
17C-17Fh	OSD3BDPU	OSD3 Buffer Description List Pointer-Upper Base Address	00000000h	R/W

6.2.1 GCAP—Global Capabilities Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 00h
Default Value: 4401h
Attribute: RO
Size: 16 bits

Bit	Description
15:12	Number of Output Stream Supported— RO. Hardwired to 0100b indicating that the Intel® ICH6 Intel® High Definition Audio controller supports 4 output streams.
11:8	Number of Input Stream Supported— RO. Hardwired to 0100b indicating that the ICH6 Intel High Definition Audio controller supports 4 input streams.
7:3	Number of Bidirectional Stream Supported— RO. Hardwired to 0 indicating that the ICH6 Intel High Definition Audio controller supports 0 bidirectional stream.
2	Reserved.
1	Number of Serial Data Out Signals— RO. Hardwired to 0 indicating that the ICH6 Intel High Definition Audio controller supports 1 serial data output signal.
0	64-bit Address Supported— RO. Hardwired to 1b indicating that the ICH6 Intel High Definition Audio controller supports 64-bit addressing for BDL addresses, data buffer addressees, and command buffer addresses.

6.2.2 VMIN—Minor Version Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 02h
Default Value: 00h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	Minor Version— RO. Hardwired to 0 indicating that the ICH6 supports minor revision number 00h of the *Intel® High Definition Audio Specification.*

6.2.3 VMAJ—Major Version Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 03h
Default Value: 01h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	Major Version— RO. Hardwired to 01h indicating that the ICH6 supports major revision number 1 of the *Intel® High Definition Audio Specification*.

6.2.4 OUTPAY—Output Payload Capability Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 04h
Default Value: 003Ch
Attribute: RO
Size: 16 bits

Bit	Description
15:7	Reserved.
6:0	Output Payload Capability— RO. Hardwired to 3Ch indicating 60 word payload. This field indicates the total output payload available on the link. This does not include bandwidth used for command and control. This measurement is in 16-bit word quantities per 48Mhz frame. The default link clock of 24.000MHz (the data is double pumped) provides 1000 bits per frame, or 62.5 words in total. 40 bits are used for command and control, leaving 60 words available for data payload. 00h = 0 word 01h = 1 word payload. ..... FFh = 256 word payload.

6.2.5 INPAY—Input Payload Capability Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 06h
Default Value: 001Dh
Attribute: RO
Size: 16 bits

Bit	Description
15:7	Reserved.
6:0	Input Payload Capability— RO. Hardwired to 1Dh indicating 29 word payload. This field indicates the total output payload available on the link. This does not include bandwidth used for response. This measurement is in 16-bit word quantities per 48Mhz frame. The default link clock of 24.000MHz provides 500 bits per frame, or 31.25 words in total. 36 bits are used for response, leaving 29 words available for data payload. 00h = 0 word 01h = 1 word payload. ..... FFh = 256 word payload.

6.2.6 GCTL—Global Control Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 08h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31:9	Reserved.
8	**Accept Unsolicited Response Enable**— R/W. 0 = Unsolicited responses from the codecs are not accepted. 1 = Unsolicited response from the codecs are accepted by the controller and placed into the Response Input Ring Buffer.
7:2	Reserved.
1	**Flush Control** — R/W. Writing a 1 to this bit initiates a flush. When the flush completion is received by the controller, hardware sets the Flush Status bit and clears this Flush Control bit. Before a flush cycle is initiated, the DMA Position Buffer must be programmed with a valid memory address by software, but the DMA Position Buffer bit 0 needs not be set to enable the position reporting mechanism. Also, all streams must be stopped (the associated RUN bit must be 0). When the flush is initiated, the controller will flush the pipelines to memory to guarantee that the hardware is ready to transition to a D3 state. Setting this bit is not a critical step in the power state transition if the content of the FIFIOs is not critical.
0	**Controller Reset #**— R/W. 0 = Writing a 0 to this bit causes the Intel® High Definition Audio controller to be reset. All state machines, FIFOs and non-resume well memory mapped configuration registers (not PCI configuration registers) in the controller will be reset. The Intel High Definition Audio link RESET# signal will be asserted, and all other link signals will be driven to their default values. After the hardware has completed sequencing into the reset state, it will report a 0 in this bit. Software must read a 0 from this bit to verify the controller is in reset. 1 = Writing a 1 to this bit causes the controller to exit its reset state and deassert the Intel High Definition Audio link RESET# signal. Software is responsible for setting/clearing this bit such that the minimum Intel High Definition Audio link RESET# signal assertion pulse width specification is met. When the controller hardware is ready to begin operation, it will report a 1 in this bit. Software must read a 1 from this bit before accessing any controller registers. This bit defaults to a 0 after Hardware reset, therefore, software needs to write a 1 to this bit to begin operation. Note: The CORB/RIRB RUN bits and all stream RUN bits must be verified cleared to 0 before writing a 0 to this bit in order to assure a clean re-start. When setting or clearing this bit, software must ensure that minimum link timing requirements (minimum RESET# assertion time, etc.) are met. When this bit is 0 indicating that the controller is in reset, all Intel High Definition Audio memory mapped registers are ignored as if the device is not present. The only exception is this bit itself, which will cause the controller to leave the reset state when a 1 is written to it.

6.2.7 WAKEEN—Wake Enable Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 0Ch
Default Value: 0000h
Attribute: R/W
Size: 16 bits

Bit	Description
15:3	Reserved.
2:0	**SDIN Wake Enable Flags**— R/W. These bits control which SDI signal(s) may generate a wake event. A 1b in the bit mask indicates that the associated SDIN signal is enabled to generate a wake. Bit 0 is used for SDI[0] Bit 1 is used for SDI[1] Bit 2 is used for SDI[2] Note: These bits are in the resume well and only cleared on a power on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately.

6.2.8 STATESTS—State Change Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 0Eh
Default Value: 0000h
Attribute: R/WC
Size: 16 bits

Bit	Description
15:3	Reserved.
2:0	**SDIN State Change Status Flags**— R/WC. Flag bits that indicate which SDI signal(s) received a state change event. The bits are cleared by writing 1's to them. Bit 0 = SDI[0] Bit 1 = SDI[1] Bit 2 = SDI[2] Note: These bits are in the resume well and only cleared on a power on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately.

6.2.9 GSTS—Global Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 10h
Default Value: 0000h
Attribute: R/WC
Size: 16 bits

Bit	Description
15:2	Reserved.
1	**Flush Status**— R/WC. This bit is set to 1 by hardware to indicate that the flush cycle initiated when the Flush Control bit (AZBAR + 08h, bit 1) was set has completed. Software must write a 1 to clear this bit before the next time the Flush Control bit is set to clear the bit.
0	Reserved.

6.2.10 INTCTL—Interrupt Control Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 20h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31	**Global Interrupt Enable (GIE)**—R/W. Global bit to enable device interrupt generation. When set to 1, the Intel® High Definition Audio function is enabled to generate an interrupt. This control is in addition to any bits in the bus specific address space, such as the Interrupt Enable bit in the PCI configuration space. **NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition.
30	**Controller Interrupt Enable (CIE)**— R/W. Enables the general interrupt for controller functions. When set to 1, the controller generates an interrupt when the corresponding status bit gets set due to a Response Interrupt, a Response Buffer Overrun, and State Change events. **NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition.
29:8	Reserved
7:0	**Stream Interrupt Enable (SIE)**—R/W. When set to 1, the individual streams are enabled to generate an interrupt when the corresponding status bits get set. A stream interrupt will be caused as a result of a buffer with IOC = 1in the BDL entry being completed, or as a result of a FIFO error (underrun or overrun) occurring. Control over the generation of each of these sources is in the associated Stream Descriptor. The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set. Bit 0: input stream 1 Bit 1: input stream 2 Bit 2: input stream 3 Bit 3: input stream 4 Bit 4: output stream 1 Bit 5: output stream 2 Bit 6: output stream 3 Bit 7: output stream 4

6.2.11 INTSTS—Interrupt Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 24h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31	Global Interrupt Status (GIS)—RO. This bit is an OR of all the interrupt status bits in this register. **NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition.
30	**Controller Interrupt Status (CIS)**— RO. Status of general controller interrupt. 1 = indicates that an interrupt occurred due to a Response Interrupt, a Response Buffer Overrun Interrupt, or a SDIN state change event. The exact cause can be determined by interrogating other registers. This bit is an OR of all of the stated interrupt status bits for this register. **NOTES:** 1. This bit is set regardless of the state of the corresponding interrupt enable bit, but a hardware interrupt will not be generated unless the corresponding enable bit is set. 2. This bit is not affected by the $D3_{HOT}$ to D0 transition.
29:8	Reserved
7:0	**Stream Interrupt Status (SIS)**—RO. 1 = indicates that an interrupt condition occurred on the corresponding stream. This bit is an OR of all of the stream's interrupt status bits. **NOTE:** These bits are set regardless of the state of the corresponding interrupt enable bits. The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set. Bit 0: input stream 1 Bit 1: input stream 2 Bit 2: input stream 3 Bit 3: input stream 4 Bit 4: output stream 1 Bit 5: output stream 2 Bit 6: output stream 3 Bit 7: output stream 4

6.2.12 WALCLK—Wall Clock Counter Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 30h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:0	Wall Clock Counter— RO. 32 bit counter that is incremented on each link BCLK period and rolls over from FFFF FFFFh to 0000 0000h. This counter will roll over to 0 with a period of approximately 179 seconds. This counter is enabled while the BCLK bit is set to 1. Software uses this counter to synchronize between multiple controllers. Will be reset on controller reset.

6.2.13 SSYNC—Stream Synchronization Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 34h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31:8	Reserved
7:0	**Stream Synchronization (SSYNC)**— R/W. When set to 1, these bits block data from being sent on or received from the link. Each bit controls the associated stream descriptor (i.e. bit 0 corresponds to the first stream descriptor, etc.) To synchronously start a set of DMA engines, these bits are first set to 1. The RUN bits for the associated stream descriptors are then set to 1 to start the DMA engines. When all streams are ready (FIFORDY =1), the associated SSYNC bits can all be set to 0 at the same time, and transmission or reception of bits to or from the link will begin together at the start of the next full link frame. To synchronously stop the streams, fist these bits are set, and then the individual RUN bits in the stream descriptor are cleared by software. If synchronization is not desired, these bits may be left as 0, and the stream will simply begin running normally when the stream's RUN bit is set. The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set. Bit 0: input stream 1 Bit 1: input stream 2 Bit 2: input stream 3 Bit 3: input stream 4 Bit 4: output stream 1 Bit 5: output stream 2

6.2.14 CORBLBASE—CORB Lower Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 40h
Default Value: 00000000h
Attribute: R/W, RO
Size: 32 bits

Bit	Description
31:7	**CORB Lower Base Address**— R/W. Lower address of the Command Output Ring Buffer, allowing the CORB base address to be assigned on any 128B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.
6:0	CORB Lower Base Unimplemented Bits— RO. Hardwired to 0. This required the CORB to be allocated with 128B granularity to allow for cache line fetch optimizations.

6.2.15 CORBUBASE—CORB Upper Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 44h Attribute: R/W
Default Value: 00000000h Size: 32 bits

Bit	Description
31:0	**CORB Upper Base Address**— R/W. Upper 32 bits of the address of the Command Output Ring buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.

6.2.16 CORBRP—CORB Read Pointer Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 4Ah Attribute: R/W
Default Value: 0000h Size: 16 bits

Bit	Description
15	**CORB Read Pointer Reset**—R/W. Software writes a 1 to this bit to reset the CORB Read Pointer to 0. The CORB DMA engine must be stopped prior to resetting the Read Pointer or else DMA transfer may be corrupted. This bit is always read as 0.
14:8	Reserved.
7:0	**CORB Write Pointer**— R/W. Software writes the last valid CORB entry offset into this field in dword granularity. The DMA engine fetches commands from the CORB until the Read pointer matches the Write pointer. Supports 256 CORB entries (256x4B = 1KB). This register field may be written when the DMA engine is running.

6.2.17 CORBCTL—CORB Control Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 4Ch Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:2	Reserved.
1	**Enable CORB DMA Engine**— R/W. 0 = DMA stop 1 = DMA run After software writes a 0 to this bit, the hardware may not stop immediately. The hardware will physically update the bit to 0 when the DMA engine is truly stopped. Software must read a 0 from this bit to verify that the DMA engine is truly stopped.
0	**CORB Memory Error Interrupt Enable**— R/W. If this bit is set the controller will generate an interrupt if the CMEI status bit (AZBAR + 4Dh: bit 0) is set.

6.2.18 CORBST—CORB Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 4Dh Attribute: R/WC
Default Value: 00h Size: 8 bits

Bit	Description
7:1	Reserved.
0	**CORB Memory Error Indication (CMEI)**— R/WC. If this bit is set, the controller has detected an error in the path way between the controller and memory. This may be an ECC bit error or any other type of detectable data error which renders the command data fetched invalid. Software can clear this bit by writing a 1 to it. However, this type of error leaves the audio subsystem in an unviable state and typically required a controller reset by writing a 0 to the Controller Reset # bit (AZBAR + 08h: bit 0).

6.2.19 CORBSIZE—CORB Size Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 4Eh Attribute: RO
Default Value: 42h Size: 8 bits

Bit	Description
7:4	CORB Size Capability—RO. Hardwired to 0100b indicating that the ICH6 only supports a CORB size of 256 CORB entries (1024B)
3:2	Reserved.
1:0	CORB Size— RO. Hardwired to 10b which sets the CORB size to 256 entries (1024B)

6.2.20 RIRBLBASE—RIRB Lower Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 50h Attribute: R/W, RO
Default Value: 00000000h Size: 32 bits

Bit	Description
31:7	**CORB Lower Base Address**— R/W. Lower address of the Response Input Ring Buffer, allowing the RIRB base address to be assigned on any 128B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.
6:0	RIRB Lower Base Unimplemented Bits— RO. Hardwired to 0. This required the RIRB to be allocated with 128B granularity to allow for cache line fetch optimizations.

6.2.21 RIRBUBASE—RIRB Upper Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 54h Attribute: R/W
Default Value: 00000000h Size: 32 bits

Bit	Description
31:0	**RIRB Upper Base Address**— R/W. Upper 32 bits of the address of the Response Input Ring Buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.

6.2.22 RIRBWP—RIRB Write Pointer Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 58h Attribute: R/W, RO
Default Value: 0000h Size: 16 bits

Bit	Description
15	**RIRB Write Pointer Reset**—R/W. Software writes a 1 to this bit to reset the RIRB Write Pointer to 0. The RIRB DMA engine must be stopped prior to resetting the Write Pointer or else DMA transfer may be corrupted. This bit is always read as 0.
14:8	Reserved.
7:0	RIRB Write Pointer (RIRBWP) — RO. Indicates the last valid RIRB entry written by the DMA controller. Software reads this field to determine how many responses it can read from the RIRB. The value read indicates the RIRB Write Pointer offset in 2 dword RIRB entry units (since each RIRB entry is 2 dwords long). Supports up to 256 RIRB entries (256 x 8 B = 2 KB). This register field may be written when the DMA engine is running.

6.2.23 RINTCNT—Response Interrupt Count Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 5Ah Attribute: R/W
Default Value: 0000h Size: 16 bits

Bit	Description
15:8	Reserved.
31:0	**N Response Interrupt Count**— R/W. 0000 0001b = 1 response sent to RIRB ..... 1111 1111b = 255 responses sent to RIRB 0000 0000b = 256 responses sent to RIRB The DMA engine should be stopped when changing this field or else an interrupt may be lost. Note that each response occupies 2 dwords in the RIRB. This is compared to the total number of responses that have been returned, as opposed to the number of frames in which there were responses. If more than one codecs responds in one frame, then the count is increased by the number of responses received in the frame.

6.2.24 RIRBCTL—RIRB Control Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 5Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

Bit	Description
7:3	Reserved.
2	**Response Overrun Interrupt Control**—R/W. If this bit is set, the hardware will generate an interrupt when the Response Overrun Interrupt Status bit (AZBAR + 5Dh: bit 2) is set.
1	**Enable RIRB DMA Engine**— R/W. 0 = DMA stop 1 = DMA run After software writes a 0 to this bit, the hardware may not stop immediately. The hardware will physically update the bit to 0 when the DMA engine is truly stopped. Software must read a 0 from this bit to verify that the DMA engine is truly stopped.
0	**Response Interrupt Control**— R/W. 0 = Disable Interrupt 1 = Generate an interrupt after N number of responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI[x] inputs (whichever occurs first). The N counter is reset when the interrupt is generated.

6.2.25 RIRBSTS—RIRB Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 5Dh
Default Value: 00h
Attribute: R/WC
Size: 8 bits

Bit	Description
7:3	Reserved.
2	**Response Overrun Interrupt Status**— R/WC. Software sets this bit to 1 when the RIRB DMA engine is not able to write the incoming responses to memory before additional incoming responses overrun the internal FIFO. When the overrun occurs, the hardware will drop the responses which overrun the buffer. An interrupt may be generated if the Response Overrun Interrupt Control bit is set. Note that this status bit is set even if an interrupt is not enabled for this event. Software clears this bit by writing a 1 to it.
1	Reserved.
0	**Response Interrupt**— R/WC. Hardware sets this bit to 1 when an interrupt has been generated after N number of Responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI[x] inputs (whichever occurs first). Note that this status bit is set even if an interrupt is not enabled for this event. Software clears this bit by writing a 1 to it.

6.2.26 RIRBSIZE—RIRB Size Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 5Eh
Default Value: 42h
Attribute: RO
Size: 8 bits

Bit	Description
7:4	RIRB Size Capability—RO. Hardwired to 0100b indicating that the ICH6 only supports a RIRB size of 256 RIRB entries (2048B)
3:2	Reserved.
1:0	RIRB Size— RO. Hardwired to 10b which sets the CORB size to 256 entries (2048B)

6.2.27 IC—Immediate Command Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 60h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31:0	**Immediate Command Write—R/W**. The command to be sent to the codec via the Immediate Command mechanism is written to this register. The command stored in this register is sent out over the link during the next available frame after a 1 is written to the ICB bit (AZBAR + 68h: bit 0)

6.2.28 IR—Immediate Response Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 64h
Default Value: 00000000h
Attribute: RO
Size: 32 bits

Bit	Description
31:0	Immediate Response Read (IRR)—RO. This register contains the response received from a codec resulting from a command sent via the Immediate Command mechanism. If multiple codecs responded in the same time, there is no guarantee as to which response will be latched. Therefore, broadcast-type commands must not be issued via the Immediate Command mechanism.

6.2.29 IRS—Immediate Command Status Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 68h
Default Value: 0000h
Attribute: R/W, R/WC
Size: 16 bits

Bit	Description
15:2	Reserved.
1	**Immediate Result Valid (IRV)**—R/WC. This bit is set to 1 by hardware when a new response is latched into the Immediate Response register (AZBAR + 64). This is a status flag indicating that software may read the response from the Immediate Response register. Software must clear this bit by writing a 1 to it before issuing a new command so that the software may determine when a new response has arrived.
0	**Immediate Command Busy (ICB)** —R/W. When this bit is read as 0, it indicates that a new command may be issued using the Immediate Command mechanism. When this bit transitions from a 0 to a 1 (via software writing a 1), the controller issues the command currently stored in the Immediate Command register to the codec over the link. When the corresponding response is latched into the Immediate Response register, the controller hardware sets the IRV flag and clears the ICB bit back to 0. **NOTE:** An Immediate Command must not be issued while the CORB/RIRB mechanism is operating, otherwise the responses conflict. This must be enforced by software.

6.2.30 DPLBASE—DMA Position Lower Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 70h
Default Value: 00000000h
Attribute: R/W, RO
Size: 32 bits

Bit	Description
31:7	**DMA Position Lower Base Address**—R/W. Lower 32 bits of the DMA Position Buffer Base Address. This register field must not be written when any DMA engine is running or the DMA transfer may be corrupted. This same address is used by the Flush Control and must be programmed with a valid value before the Flush Control bit (AZBAR+08h:bit 1) is set.
6:1	DMA Position Lower Base Unimplemented bits—RO. Hardwired to 0 to force the 128 byte buffer alignment for cache line write optimizations.
0	**DMA Position Buffer Enable**—R/W. When this bit is set to 1, the controller will write the DMA positions of each of the DMA engines to the buffer in the main memory periodically (typically once per frame). Software can use this value to know what data in memory is valid data.

6.2.31 DPUBASE—DMA Position Upper Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: AZBAR + 74h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Bit	Description
31:0	**DMA Position Upper Base Address**—R/W. Upper 32 bits of the DMA Position Buffer Base Address. This register field must not be written when any DMA engine is running or the DMA transfer may be corrupted.

6.2.32 SDCTL—Stream Descriptor Control Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 80h
Input Stream[1]: AZBAR + A0h
Input Stream[2]: AZBAR + C0h
Input Stream[3]: AZBAR + E0h
Output Stream[0]: AZBAR + 100h
Output Stream[1]: AZBAR + 120h
Output Stream[2]: AZBAR + 140h
Output Stream[3]: AZBAR + 160h
Attribute: R/W, RO

Default Value: 040000h
Size: 24 bits

Bit	Description
23:20	**Stream Number**—R/W. This value reflect the Tag associated with the data being transferred on the link. When data controlled by this descriptor is sent out over the link, it will have its stream number encoded on the SYNC signal. When an input stream is detected on any of the SDI signals that match this value, the data samples are loaded into FIFO associated with this descriptor. Note that while a single SDI input may contain data from more than one stream number, two different SDI inputs may not be configured with the same stream number. 0000 = Reserved 0001 = Stream 1 ..... 1110 = Stream 14 1111 = Stream 15
19	Bidirectional Direction Control—RO. This bit is only meaningful for bidirectional streams; therefore, this bit is hardwired to 0.
18	Traffic Priority—RO. Hardwired to 1 indicating that all streams will use VC1 if it is enabled through the PCI Express registers.
17:16	Stripe Control—RO. This bit is only meaningful for input streams; therefore, this bit is hardwired to 0.
15:5	Reserved
4	**Descriptor Error Interrupt Enable**—R/W. 0 = Disable 1 = An interrupt is generated when the Descriptor Error Status bit is set.

Bit	Description
3	**FIFO Error Interrupt Enable**—R/W. This bit controls whether the occurrence of a FIFO error (overrun for input or underrun for output) will cause an interrupt or not. If this bit is not set, bit 3in the Status register will be set, but the interrupt will not occur. Either way, the samples will be dropped.
2	**Interrupt on Completion Enable**—R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. If this bit is not set, bit 2 in the Status register will be set, but the interrupt will not occur.
1	**Stream Run (RUN)**—R/W. 0 = When cleared to 0, the DMA engine associated with this input stream will be disabled. The hardware will report a 0 in this bit when the DMA engine is actually stopped. Software must read a 0 from this bit before modifying related control registers or restarting the DMA engine. 1 = When set to 1, the DMA engine associated with this input stream will be enabled to transfer data from the FIFO to the main memory. The SSYNC bit must also be cleared in order for the DMA engine to run. For output streams, the cadence generator is reset whenever the RUN bit is set.
0	**Stream Reset (SRST)**—R/W. 0 = Writing a 0 causes the corresponding stream to exit reset. When the stream hardware is ready to begin operation, it will report a 0 in this bit. Software must read a 0 from this bit before accessing any of the stream registers. 1 = Writing a 1 causes the corresponding stream to be reset. The Stream Descriptor registers (except the SRST bit itself) and FIFO's for the corresponding stream are reset. After the stream hardware has completed sequencing into the reset state, it will report a 1 in this bit. Software must read a 1 from this bit to verify that the stream is in reset. The RUN bit must be cleared before SRST is asserted.

6.2.33 SDSTS—Stream Descriptor Status Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 83h
Input Stream[1]: AZBAR + A3h
Input Stream[2]: AZBAR + C3h
Input Stream[3]: AZBAR + E3h
Output Stream[0]: AZBAR + 103h
Output Stream[1]: AZBAR + 123h
Output Stream[2]: AZBAR + 143h
Output Stream[3]: AZBAR + 163h

Attribute: R/WC, RO

Default Value: 00h

Size: 8 bits

Bit	Description
7:6	Reserved.
5	FIFO Ready (FIFORDY)—RO. For output streams, the controller hardware will set this bit to 1 while the output DMA FIFO contains enough data to maintain the stream on the link. This bit defaults to 0 on reset because the FIFO is cleared on a reset. For input streams, the controller hardware will set this bit to 1 when a valid descriptor is loaded and the engine is ready for the RUN bit to be set.
4	**Descriptor Error**—R/WC. When set, this bit indicates that a serious error occurred during the fetch of a descriptor. This could be a result of a Master Abort, a parity or ECC error on the bus, or any other error which renders the current Buffer Descriptor or Buffer Descriptor list useless. This error is treated as a fatal stream error, as the stream cannot continue running. The RUN bit will be cleared and the stream will stopped. Software may attempt to restart the stream engine after addressing the cause of the error and writing a 1 to this bit to clear it.
3	**FIFO Error**—R/WC. This bit is set when a FIFO error occurs. This bit is set even if an interrupt is not enabled. The bit is cleared by writing a 1 to it. For an input stream, this indicates a FIFO overrun occurring while the RUN bit is set. When this happens, the FIFO pointers do not increment and the incoming data is not written into the FIFO, thereby being lost. For an output stream, this indicates a FIFO underrun when there are still buffers to send. The hardware should not transmit anything on the link for the associated stream if there is not valid data to send.
2	**Buffer Completion Interrupt Status**—R/WC. This bit is set to 1 by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion bit is set in the command byte of the buffer descriptor. It remains active until software clears it by writing a 1 to it.
1:0	Reserved.

6.2.34 SDLPIB—Stream Descriptor Link Position in Buffer Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 84h
Input Stream[1]: AZBAR + A4h
Input Stream[2]: AZBAR + C4h
Input Stream[3]: AZBAR + E4h
Output Stream[0]: AZBAR + 104h
Output Stream[1]: AZBAR + 124h
Output Stream[2]: AZBAR + 144h
Output Stream[3]: AZBAR + 164h

Attribute: RO

Default Value: 00000000h

Size: 32 bits

Bit	Description
31:0	Link Position in Buffer—RO. Indicates the number of bytes that have been received off the link. This register will count from 0 to the value in the Cyclic Buffer Length register and then wrap to 0.

6.2.35 SDCBL—Stream Descriptor Cyclic Buffer Length Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 88h
Input Stream[1]: AZBAR + A8h
Input Stream[2]: AZBAR + C8h
Input Stream[3]: AZBAR + E8h
Output Stream[0]: AZBAR + 108h
Output Stream[1]: AZBAR + 128h
Output Stream[2]: AZBAR + 148h
Output Stream[3]: AZBAR + 168h

Attribute: R/W

Default Value: 00000000h

Size: 32 bits

Bit	Description
31:0	**Cyclic Buffer Length**—R/W. Indicates the number of bytes in the complete cyclic buffer. This register represents an integer number of samples. Link Position in Buffer will be reset when it reaches this value. Software may only write to this register after Global Reset, Controller Reset, or Stream Reset has occurred. This value should be only modified when the RUN bit is 0. Once the RUN bit has been set to enable the engine, software must not write to this register until after the next reset is asserted, or transfer may be corrupted.

6.2.36 SDLVI—Stream Descriptor Last Valid Index Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 8Ch
Input Stream[1]: AZBAR + ACh
Input Stream[2]: AZBAR + CCh
Input Stream[3]: AZBAR + ECh
Output Stream[0]: AZBAR + 10Ch
Output Stream[1]: AZBAR + 12Ch
Output Stream[2]: AZBAR + 14Ch
Output Stream[3]: AZBAR + 16Ch
Attribute: R/W

Default Value: 0000h
Size: 16 bits

Bit	Description
15:8	Reserved.
7:0	**Last Valid Index**—R/W. The value written to this register indicates the index for the last valid Buffer Descriptor in BDL. After the controller has processed this descriptor, it will wrap back to the first descriptor in the list and continue processing. This field must be at least 1, i.e. there must be at least 2 valid entries in the buffer descriptor list before DMA operations can begin. This value should only modified when the RUN bit is 0.

6.2.37 SDFIFOW—Stream Descriptor FIFO Watermark Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 8Eh
Input Stream[1]: AZBAR + AEh
Input Stream[2]: AZBAR + CEh
Input Stream[3]: AZBAR + EEh
Output Stream[0]: AZBAR + 10Eh
Output Stream[1]: AZBAR + 12Eh
Output Stream[2]: AZBAR + 14Eh
Output Stream[3]: AZBAR + 16Eh
Attribute: R/W

Default Value: 0004h
Size: 16 bits

Bit	Description
15:3	Reserved.
2:0	**FIFO Watermark (FIFOW)**—R/W. Indicates the minimum number of bytes accumulated/free in the FIFO before the controller will start a fetch/eviction of data. 010 = 8B 011 = 16B 100 = 32B (Default) Others = Unsupported Note: When the bit field is programmed to an unsupported size, the hardware sets itself to the default value. Software must read the bit field to test if the value is supported after setting the bit field.

6.2.38 SDFIFOS—Stream Descriptor FIFO Size Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 90h Attribute:Input: RO
Input Stream[1]: AZBAR + B0h Output: R/W
Input Stream[2]: AZBAR + D0h
Input Stream[3]: AZBAR + F0h
Output Stream[0]: AZBAR + 110h
Output Stream[1]: AZBAR + 130h
Output Stream[2]: AZBAR + 150h
Output Stream[3]: AZBAR + 170h

Default Value: Input Stream: 0077h Size: 16 bits
Output Stream: 00BFh

Bit	Description
15:8	Reserved.
7:0	**FIFO Size**—RO (Input stream), R/W (Output stream). Indicates the maximum number of bytes that could be fetched by the controller at one time. This is the maximum number of bytes that may have been DMA'd into memory but not yet transmitted on the link, and is also the maximum possible value that the PICB count will increase by at one time. The value in this field is different for input and output streams. It is also dependent on the Bits per Samples setting for the corresponding stream. Following are the values read/written from/to this register for input and output streams, and for non-padded and padded bit formats: *Output Stream R/W value*: **Value** — **Output Streams** 0Fh = 16B — 8, 16, 20, 24, or 32 bit Output Streams 1Fh = 32B — 8, 16, 20, 24, or 32 bit Output Streams 3Fh = 64B — 8, 16, 20, 24, or 32 bit Output Streams 7Fh = 128B — 8, 16, 20, 24, or 32 bit Output Streams BFh = 192B — 8, 16, or 32 bit Output Streams FFh = 256B — 20, 24 bit Output Streams **NOTES:** 1. All other values not listed are not supported. 2. When the output stream is programmed to an unsupported size, the hardware sets itself to the default value (BFh). 3. Software must read the bit field to test if the value is supported after setting the bit field. *Input Stream RO value*: **Value** — **Input Streams** 77h = 120B — 8, 16, 32 bit Input Streams 9Fh = 160B — 20, 24 bit Input Streams **NOTE:** The default value is different for input and output streams, and reflects the default state of the BITS fields (in Stream Descriptor Format registers) for the corresponding stream.

6.2.39 SDFMT—Stream Descriptor Format Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 92h
Input Stream[1]: AZBAR + B2h
Input Stream[2]: AZBAR + D2h
Input Stream[3]: AZBAR + F2h
Output Stream[0]: AZBAR + 112h
Output Stream[1]: AZBAR + 132h
Output Stream[2]: AZBAR + 152h
Output Stream[3]: AZBAR + 172h

Attribute: R/W

Default Value: 0000h

Size: 16 bits

Bit	Description
15	Reserved.
14	**Sample Base Rate**—R/W 0 = 48 kHz 1 = 44.1 kHz
13:11	**Sample Base Rate Multiple**—R/W 000 = 48 kHz, 44.1 kHz or less 001 = x2 (96 kHz, 88.2 kHz, 32 kHz) 010 = x3 (144 kHz) 011 = x4 (192 kHz, 176.4 kHz) Others = Reserved.
10:8	**Sample Base Rate Devisor**—R/W. 000 = Divide by 1(48 kHz, 44.1 kHz) 001 = Divide by 2 (24 kHz, 22.05 kHz) 010 = Divide by 3 (16 kHz, 32 kHz) 011 = Divide by 4 (11.025 kHz) 100 = Divide by 5 (9.6 kHz) 101 = Divide by 6 (8 kHz) 110 = Divide by 7 111 = Divide by 8 (6 kHz)
7	Reserved.
6:4	**Bits per Sample (BITS)**—R/W. 000 = 8 bits. The data will be packed in memory in 8-bit containers on 16-bit boundaries 001 = 16 bits. The data will be packed in memory in 16-bit containers on 16-bit boundaries 010 = 20 bits. The data will be packed in memory in 32-bit containers on 32-bit boundaries 011 = 24 bits. The data will be packed in memory in 32-bit containers on 32-bit boundaries 100 = 32 bits. The data will be packed in memory in 32-bit containers on 32-bit boundaries Others = Reserved.
3:0	Number of Channels (CHAN)—R/W. Indicates number of channels in each frame of the stream. 0000 =1 0001 =2 ..... 1111 =16

6.2.40 SDBDPL—Stream Descriptor Buffer Descriptor List Pointer Lower Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 98h
Input Stream[1]: AZBAR + B8h
Input Stream[2]: AZBAR + D8h
Input Stream[3]: AZBAR + F8h
Output Stream[0]: AZBAR + 118h
Output Stream[1]: AZBAR + 138h
Output Stream[2]: AZBAR + 158h
Output Stream[3]: AZBAR + 178h

Attribute: R/W,RO

Default Value: 00000000h

Size: 32 bits

Bit	Description
31:7	**Buffer Descriptor List Pointer Lower Base Address**—R/W. Lower address of the Buffer Descriptor List. This value should only be modified when the RUN bit is 0, or DMA transfer may be corrupted.
6:0	Hardwired to 0 forcing alignment on 128-B boundaries.

6.2.41 SDBDPU—Stream Descriptor Buffer Descriptor List Pointer Upper Base Address Register (High Definition Audio Controller—D27:F0)

Memory Address: Input Stream[0]: AZBAR + 9Ch
Input Stream[1]: AZBAR + BCh
Input Stream[2]: AZBAR + DCh
Input Stream[3]: AZBAR + FCh
Output Stream[0]: AZBAR + 11Ch
Output Stream[1]: AZBAR + 13Ch
Output Stream[2]: AZBAR + 15Ch
Output Stream[3]: AZBAR + 17Ch

Attribute: R/W

Default Value: 00000000h

Size: 32 bits

Bit	Description
31:0	**Buffer Descriptor List Pointer Upper Base Address**—R/W. Upper 32-bit address of the Buffer Descriptor List. This value should only be modified when the RUN bit is 0, or DMA transfer may be corrupted.

§

7 Intel® High Definition Audio BIOS Considerations

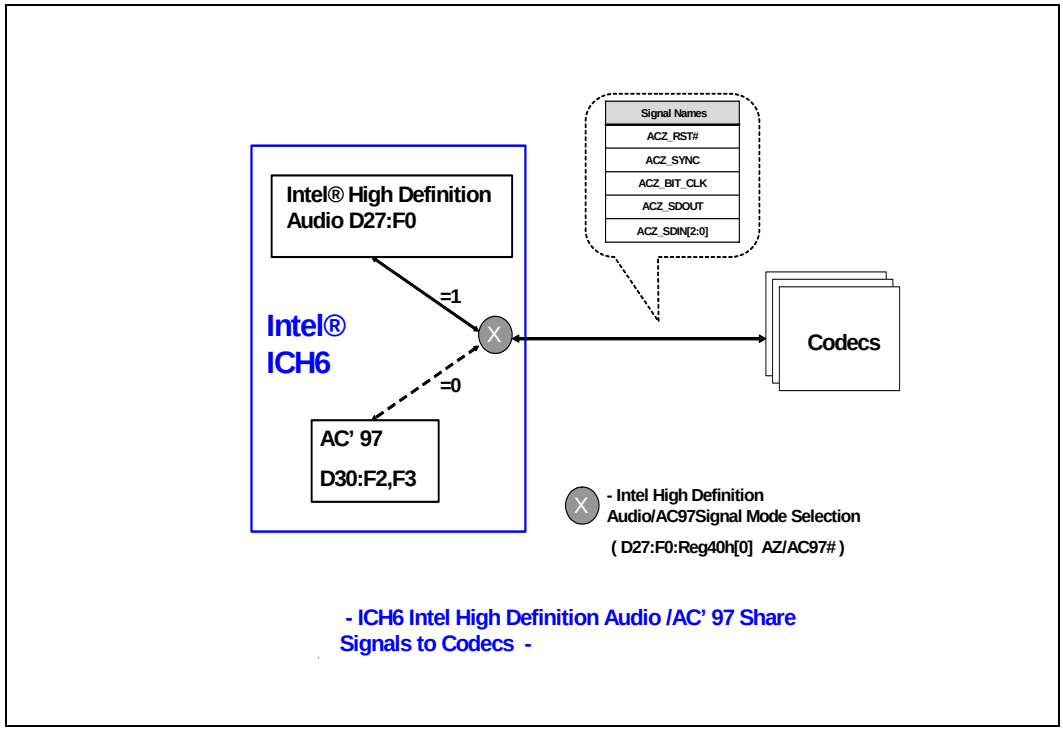
The Intel High Definition Audio controller (Bus #0, Device #27, Function #0) is an ICH6 internal PCI Express Endpoint device. Software may access the Intel High Definition Audio controller registers (including the memory mapped registers) by byte, word, dword quantities and on natural boundaries; dword accesses must be on dword boundaries, word accesses on word boundaries, etc.

This chapter describes BIOS requirements for Intel High Definition Audio controller support.

7.1 Intel® High Definition Audio/AC' 97 Signal Mode Selection

The Intel High Definition Audio controller and the AC'97 controllers (audio and modem) share the same physical signal pins to communicate with codecs as Figure 7-1 below shows. The **High Definition Audio/AC97# Signal Mode (AZ/AC97#)** bit at D27:F0:Reg40h[0] determines which one of the two controllers is connected to the codec.

Figure 7-1. Intel® ICH6 Intel® High Definition Audio/AC'97 Share Signals to Codecs



7.1.1 Intel® High Definition Audio/AC' 97 Codec Detection

Before PCI device enumeration during POST, BIOS must determine the type of codec present on the platform, then program the AZ/AC97# bit to select either Intel High Definition Audio or AC' 97 signal mode, and program the corresponding bit in the Function Disable register (RCBA+ 3418h[6:4]) to disable the other controller.

If the BIOS has inherent knowledge of which type of codec(s) will be connected to the ICH6's signals, it can set the AZ/AC97# bit accordingly.

ICH6 also provides a mechanism for software to detect the type of the codec present on the platform. Below is the Intel High Definition Audio register used in the codec detection.

Table 7-1. D27:F0:Reg40h - AZCTL - Intel® High Definition Audio Control

Bit	Type	Reset	Description
7:4	*RsvdP*	*0's*	*Reserved*
3	RW	0	**BITCLK Detect Clear (CLKDETCLR):** Writing a 1 to this bit clears the CLKDET# bit. The CLKDET# bit remains clear while this bit is set to 1. When a 0 is written to this bit, the clock detect circuit is operational and may be enabled.
2	RW	0	**BITCLK Detect Enable (CLKDETEN):** Writing a 1 to this bit enables the clock detection circuit. Writing a 0 latches the current state of the CLKDET# bit.
1	RO	0	**BITCLK Detected Inverted (CLKDET#):** This bit is modified by hardware. It is set to 0 when the ICH detects that the BITCLK signal is toggling, indicating the presence of an AC'97 codec on the link. Note that the CLKDETEN and CLKDETCLR bits control the operation of this bit and must be manipulated correctly in order to get a valid CLKDET# indicator.
0	RW	0	**High Definition Audio/AC97# Signal Mode (AZ/AC97#):** This bit selects the mode of the shared Intel® High Definition Audio/AC '97 signals. When set to 0 AC97 mode is selected. When set to 1 Intel High Definition Audio mode is selected. The bit defaults to 0 (AC97 mode) to protect against contention on BCLK when an AC97 codec is connected. Note that this bit has no affect on the visibility of the AC97 and Intel High Definition Audio function configuration space. That is controlled through individual function enable bits. This bit is in the resume well and only cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately.

In the Intel High Definition Audio environment the ICH6 drives BITCLK signal. The AZ/AC97# bit defaults to 0 (AC97 mode) after reset, so the BITCLK signal will be configured as an input. The ICH6 samples the BITCLK signal. If it is toggling the CLKDET# bit will be set to 0. The BIOS can read this bit and set the AZ/AC97# bit accordingly.

BIOS should perform this detection prior to PCI enumeration. The following is the BIOS software flow for codec detection:

1. Ensure that ICH6 RCBA base address register (D31:F0:Reg F0h) is initialized and enabled.
2. Ensure that both AC’97 and Intel High Definition Audio functions are present (RCBA+ 3418[6:4]=000b), which is the power on default.
3. Set IOSE bit (D30:F2:Reg41h[0]=1), program the AC'97 function's PCI BARs with temporary address values and enable IO BAR space via PCI command register.
4. Deassert AC_RESET# bit to take the link out of RESET# (NABMBAR at D30:F2:Reg14h + offset 2Ch[1]=1).
5. Wait ~20ms for AC'97 codec driven BIT_CLK startup.
6. Write a 0 to the Intel High Definition Audio/AC97# bit (D27:F0:Reg40h[0]=0) to ensure that AC’97 mode is selected.
7. Make sure that CLKDET# bit is cleared by writing a 1 and then a 0 to the CLKDETCLR bit.
8. Write a 1 to the CLKDETEN bit to enable the clock detection circuit.
9. Write a 0 to the CLKDETEN bit.
10. Read the CLKDET# bit.
11. If CLKDET# is clear(==0), then the codec(s) are AC'97. Disable and hide the Intel High Definition Audio function. Skip the steps below and exit.
12. If CLKDET# is set (==1), then the codec(s) are Intel High Definition Audio.
13. Reassert AC_RESET# bit to put the link back into reset state (NABMBAR at D30:F2:Reg14h + offset 2Ch[1]=0).
14. Clear the AC'97 BARs and disable memory/IO space through its PCI register 04h.
15. Hide the AC'97 functions (RCBA+ 3418h[6:5] = 11b)
16. Set the AZ/AC'97# bit to 1 to enable Intel High Definition Audio signal mode (D27:F0:Reg40h[0]=1b)
17. Program the Intel High Definition Audio AZBAR at PCI config space 10h-17h to a temporary address and enable it by setting PCI command register 04h[1]=1.
18. Deassert the Controller Reset# bit in Intel High Definition Audio to cause the link to start up (AZBAR+08h [0] = 1)
19. Clear STATESTS bits (AZBAR+0Eh [2:0]) by writing 1s to them.
20. Turn off the link by writing a 0 to the Controller Reset# bit in Intel High Definition Audio (AZBAR+08h [0] = 0). Poll Controller Reset# bit until it reads back as 0.
21. Turn on the link again by writing a 1 to Controller Reset# bit (AZBAR+08h [0] = 1). This causes a codec link re-enumeration. Wait for about 1 millisecond (ms). Poll Controller Reset# bit until it reads back as 1.
22. Read the STATESTS bits (AZBAR+0Eh [2:0]) which will indicate which SDIN lines have codecs on them. If there is one or more bits set to 1, Intel High Definition Audio codec(s) are present, go to step 24. Otherwise there is no codec present.
23. If there is no codec present, BIOS can disable the Intel High Definition Audio controller by

- Turning off the link by writing a 0 to the Controller Reset bit (AZBAR+08h [0] = 0).

- Clearing Intel High Definition Audio AZBAR register (offset 10h), then write 0 to PCI command register at offset 04h.
- Disabling Intel High Definition Audio controller via Function Disable register (set RCBA+ 3418h[4] =1).

Skip the following steps and exit.

24. For each Intel High Definition Audio codec present as indicated by AZBAR+0Eh[2:0], perform codec initialization as described in the next section.

7.1.2 Intel® High Definition Audio Codec Initialization

This section involves the programming interface on Intel High Definition Audio codec link. Readers are encouraged to read the relevant chapters of *Intel® High Definition Audio Specificat*ion for information regarding architecture overview, register interface, programming model and codec features & requirements.

Intel High Definition Audio allows flexible configurations of the inputs and outputs among its internal functional units and between codec and external jacks. Each pair of pins in the codec is assigned to an internal node in the codec, so the information related to the jack position, color coding, etc. is mapped to the node that is internally assigned to the pins and wired to a jack. This information will allow the audio driver to configure the audio codecs correctly

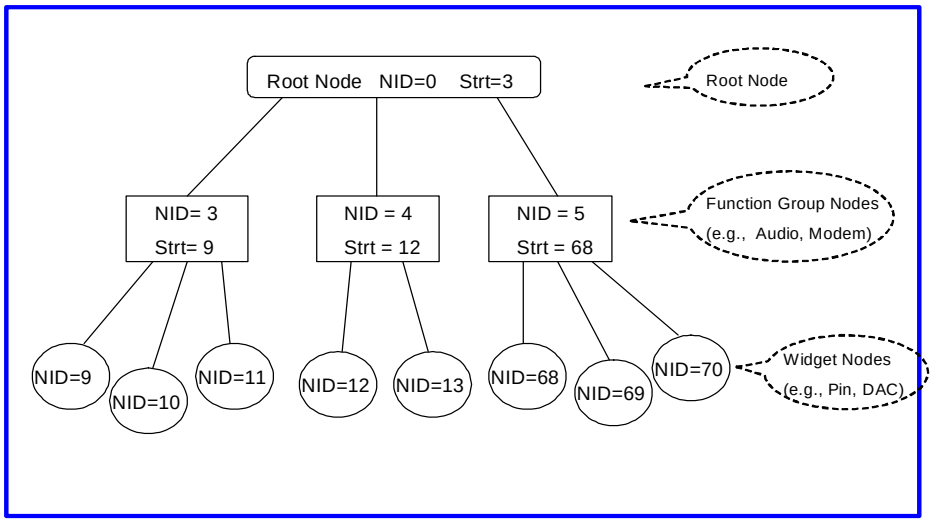
After BIOS has determined the presence of Intel High Definition Audio codec(s), it must follow the programming sequence given in this section to update the codec with correct jack information specific to the platform for Intel High Definition Audio driver to retrieve and use later. If the codecs are not initialized with this platform-specific information, the Intel High Definition Audio driver will use the default data in the codecs which may or may not match the pin/jack connections or jack locations of the platform.

7.1.2.1 Intel® High Definition Audio Codec Architecture Introduction

The *Intel® High Definition Audio Specification* defines a modular codec architecture that is fully discoverable and configurable by software. It provides for the construction and description of various codec functions from a defined set of parameterized modules (building blocks, or Widgets). Each such module and each collection of modules becomes a uniquely addressable Node, from which software can read capability parameters and to which it can send control commands. The root node is the top level node and always has a Node ID (NID) of 0. Each node contains information of the next level of nodes below it, in a tree structure as shown in Figure 7-2

For each Intel High Definition Audio codec present, a unique Codec Address (CAd) is assigned to the codec by hardware after reset during the codec link initialization and will be used for software to address each codec. For instance, the codec connected to SDI0 (as indicated by AZBAR+0Eh[0]=1) has its CAd=0, the codec connected to SDI1 (as indicated by AZBAR+0Eh[1]=1) has its CAd=1, and so on, each node in a codec has its pre-defined, unique Node ID (NID). The CAd+NID combination is used by a Verb to uniquely address a codec node.

Figure 7-2. Intel® High Definition Audio Codec Node Structure and Addressing



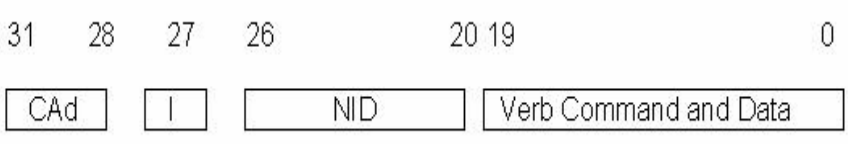
- High Definition Audio Codec Node Structure and Addressing-

A codec **verb** is a 32-bit dword command sent to a codec by software, which contains the following information:

- Codec address and Node ID of the target node in the codec
- Command to be performed by the target node
- Data payload (if any)

Below is the format of a Verb dword.

Figure 7-3. Format of a Verb Dword



There are two ways for software to send verbs to and receive response data from codecs over the Intel High Definition Audio codec link: Using CORB/RIRB (Command Output Ring Buffer / Response Input Ring Buffer), or using Immediate Command/Immediate Response register pair. See the *Intel® High Definition Audio Specification* for details of register description and programming interface.

7.1.2.2 Codec Verb Table

For each codec present on the Intel High Definition Audio codec link, a corresponding pre-defined "Codec Verb Table" must be available to BIOS. The Codec Verb Tables are based on codec specific information (coded datasheet) and platform design specific information (schematics) and are built by BIOS writers and platform designers. The table contains a list of 32-bit "Verb"s (command and data payload) to be sent to the corresponding codec over the Intel High Definition Audio codec link.

A software tool utility is available from Intel that helps with the generation of the codec verb tables for platforms in existence or platforms under design. Please contact your Intel PAE/FAE for details of this utility.

Below is a sample Intel High Definition Audio Codec Verb Table, defined in Intel x86 Assembly Language, for a platform with 1 codec at codec address 01h.

```
;Sample High Definition Audio Codec Verb Table
;Codec Address (CAd) = 01h
;Codec Vendor: XYZ Company
;VenID DevID:
dd 12345678h
;SubSystemID:
dw 4321h
;--------------------------------------------------
; FrontPanel_Supported?   ; 1=Supported ,  0=Not supported
db  01h
; # of Rear Panel Pin Complexes
dw 0009h
; # of Front Panel Pin Complexes
dw 0002h-----------------------------------------------
VerbTable0:
;Pin Complex 1       (NID 12h)
dd 11271C11h
dd 11271D81h
dd 11271E30h
dd 11271F00h
;Pin Complex 2       (NID 11h)
dd 11171C11h
dd 11171D01h
dd 11171E40h
dd 11171F00h
;Pin Complex 3       (NID 13h)
dd 11371C11h
dd 11371DA1h
dd 11371E60h
dd 11371F00h
;Pin Complex 4       (NID 32h)
dd 13271C11h
```

```
dd 13271D01h
dd 13271E00h
dd 13271F00h
;Pin Complex 5      (NID 15h)
dd 11571C11h
dd 11571D01h
dd 11571E00h
dd 11571F00h
;Pin Complex 6      (NID 31h)
dd 13171C11h
dd 13171D01h
dd 13171E00h
dd 13171F00h
;Pin Complex 9      (NID 19h)
dd 11971C11h
dd 11971DC4h
dd 11971E00h
dd 11971F00h
;Pin Complex 10      (NID 18h)
dd 11871C11h
dd 11871D04h
dd 11871E00h
dd 11871F00h
;Pin Complex 11      (NID 17h)
dd 11771C90h
dd 11771D3Fh
dd 11771E00h
dd 11771F00h

VerbTable0FP:

;Pin Complex 7      (NID 14h)              Front Panel Jack
dd 11471C02h
dd 11471D21h
dd 11471E10h
dd 11471F00h
;Pin Complex 8      (NID 16h)              Front Panel Jack
dd 11671C02h
dd 11671DA1h
dd 11671E10h
dd 11671F00h
```

7.1.2.3 Codec Initialization Programming Sequence

After BIOS has determined the presence of Intel High Definition Audio codec(s), it must follow the programming sequence given in this section to update the codec with correct jack information specific to the platform for Intel High Definition Audio driver to retrieve and use later.

There are two ways for software to send verbs to and receive response data from codecs over the Intel High Definition Audio codec link: Using CORB/RIRB (Command Output Ring Buffer / Response Input Ring Buffer), or using Immediate Command/Immediate Response register pair. The sequence below uses the latter which does not require the availability of a memory buffer.

BIOS should ensure that the Intel High Definition Audio AZBAR at PCI config space 10h-17h contains a valid address value and is enabled by setting PCI command register 04h[1]=1. BIOS should also ensure that the Controller Reset# bit of Global Control register in memory-mapped space (AZBAR+08h[0]) is set to 1 and read back as 1.

For each Intel High Definition Audio codec present as indicated by AZBAR+0Eh[2:0], BIOS should perform the codec initialization as described below:

1. Read the VenderID/DeviceID pair from the attached codec

- Poll the ICB bit of IRS register at AZBAR+68h[0] to make sure it returns 0.
- Write verb c00F0000h (dword) to the IC register at AZBAR+60h, where: ‘c’ (bits 31:28) represents the codec address (CAd).
- Write the bits of IRS register at AZBAR+68h[1:0] to 11b to send the verb to codec.
- Poll IRS register bits at AZBAR+68h[1:0] until it returns 10b indicating the verb has been sent to the codec and response data from codec is now valid.
- Read the IR register at AZBAR+64h, the dword data is the VID/DID value returned by the codec.

2. Check against internal list to determine if there is a stored verb table which matches the CAd/ VID/DID information.

Note: Note that steps 1 and 2 are BIOS implementation-specific steps and can be done in different ways. If a BIOS has prior knowledge of fixed platform/codec combination (e.g., for a BIOS having 3 stored verb tables for 3 known codecs at known codec addresses on a known platform), a simple pre-defined codec-to-table matching can be used and steps 1 and 2 can be eliminated. For a BIOS to support multiple codec/platform combinations, an internal match-list might be needed to match a platform/codec combination to a codec verb table.

3. If there is a match, send the entire list of verbs in the matching verb table one by one to the codec.

- Poll the ICB bit of IRS register at AZBAR+68h[0] to make sure it returns 0.
- Write the next verb (dword) in the table to the IC register at AZBAR+60h,
- Write the bits of IRS register at AZBAR+68h[1:0] to 11b to send the verb to codec.
- Poll the ICB bit of IRS register at AZBAR+68h[0] until it returns 0 indicating the verb has been sent to the codec.
- Repeat the steps until all the verbs in the table have been sent.

Note: Some verbs in the table may need to be qualified by certain platform-specific conditions. For example, for the sample table above, the verbs for Pin Complex 7 and 8 (NID=14,16 respectively) should be sent only if the Front Panel Jacks are present and connected on the platform, which may be indicated by a software flag that is controlled by certain GPIO pin state.

7.1.2.4 Codec Initialization Sample Code

This section shows an example of code implementation of the Intel High Definition Audio codec initialization sequence.

```
;-----------------------------------------------------------------------------

; Procedure:Initialize High Definition AudioCodecs
; Description:Initialize High Definition Audio Codecs by sending
;                       codec verbs to codecs.
;
; Input:
;                       ES      - 0000h with 4GB limit.
;                       STACK   - Available.
;                       High Definition Audio controller's AZBAR is initialized and enabled.
;  Codec verb tables are available and defined in the
;  same code segment.


; Output:
;   CF : 1 = Codec initialization failure
;   CF : 0 = Codec initialization success
; Registers modified:All except segment registers.

; Notes:

;  MKF_High Definition Audio_BASE_ADDRESS = the value of AZBAR register

;  MKF_MAX_NUM_AZAL_CODECS = 3 (max of 3 codecs supported)

;  High Definition Audio_MMIO_STATESTS = 0Eh

;  High Definition Audio_MMIO_IC = 60h

;  High Definition Audio_MMIO_IR = 64h

;  High Definition Audio_MMIO_ICS_ICB = 68h

;  VerbHeaderSize  =  11d

;-----------------------------------------------------------------------------

InitializeHigh Definition AudioCodecs  PROC NEAR PUBLIC

;       ebx will always hold the High Definition Audio base address

        mov     ebx, MKF_High Definition Audio_BASE_ADDRESS

;       ecx is the current codec address (only 15 codecs are supported in the

;          High Definition Audio spec so only the lower 4 bits are relevant)

        mov     ecx, MKF_MAX_NUM_AZAL_CODECS

;       dx is the map of SDI pins, and the bits will be cleared as the
```

```
;         associated codecs are serviced

      mov     dx, word ptr es:[ebx+High Definition Audio_MMIO_STATESTS]

InitCurrentCodec:

      dec     cx

      btr     dx, cx                    ; Test for 'cx'th codec

      jnc     NextSDI

;-------------------------------------------------------------------------

;1.    Ensure High Definition Audio device is enabled and BARs are programmed

;   a. Program High Definition Audio BARs with temporary values

;   b. Enable memory space and bus mastering

;   c. Deassert CRST#

;-------------------------------------------------------------------------

;       a. Set the AZ/AC'97# bit to 1 to enable High Definition Audio signal mode
(D27:F0:Reg40h[0]=1b)

      mov     ah, High Definition Audio_AZCTL_OFFSET

      mov     al, High Definition Audio_AZCTL_OFFSET_AZ_AC97

      _SET_PCI_FAR    High Definition Audio

;       b. Program the High Definition Audio AZBAR at PCI config space 10h-17h to a temepory
address

      mov     ah, PCI_BAR0

      mov     ebx, MKF_High Definition Audio_BASE_ADDRESS

      _WRITE_PCI_DWORD_FAR    High Definition Audio

;       c. Enable memory space and bus mastering for High Definition Audio

      mov     al, CMD_MEM_SPACE+CMD_BUS_MASTER

      _SET_PCI_FAR     High Definition Audio

;       d. Deassert the CRST bit in High Definition Audio to cause the link to start
up(AZBAR+08h[0]=1)

      or      byte ptr es:[ebx+High Definition Audio_MMIO_GCTL], High Definition
Audio_MMIO_GCTL_CRST


;-------------------------------------------------------------------------
```

```
;2.    Read the Vendor ID/Device ID pair from the attached codec
;
;  a. Poll the ICB bit in the ICS register at AZBAR+68h[0] until it returns 0
;  b. Write verb c00F0000h (dword) to the IC register at AZBAR+60h; where 'c'
;     (bits 31:28) respresents the codec address (CAd).
;  c. Set bits 1:0 of the IRS register at AZBAR+68h[1:0]
;  d. Poll ICS register bits at AZBAR+68h[1:0] until they return 10b indicating
;     the verb has been sent to the codec and response data from codec is now valid.
;  e. Read IR register at AZBAR+64h, the dword data is the VendorID/Device
;     ID value returned by the codec
;-------------------------------------------------------------------------------
;      a. Poll the ICB bit in the ICS register at AZBAR+68h[0] until it returns 0


      push    cx
      xor     cx, cx                    ; 64K cycles
PollICBBit:
      test    word ptr es:[ebx+High Definition Audio_MMIO_ICS], High Definition
Audio_MMIO_ICS_ICB
      jz      ICBBitClear               ; Poll ICB bit until it returns 0
      loop    PollICBBit
;      Add error handling code here
;      When we timeout, reset link per audio driver team request
      and     byte ptr es:[ebx+High Definition Audio_MMIO_GCTL], NOT High Definition
Audio_MMIO_GCTL_CRST
      or      byte ptr es:[ebx+High Definition Audio_MMIO_GCTL], High Definition
Audio_MMIO_GCTL_CRST
ICBBitClear:
      pop     cx


;      b. Write verb c00F0000h (dword) to the IC register at AZBAR+60h; where 'c'
```

```
;       (bits 31:28) respresents the codec address (CAd).
        mov     eax, ecx
        shl     eax, 28
        or      eax, 000F0000h
        mov     dword ptr es:[ebx+High Definition Audio_MMIO_IC], eax ; Write the verb
;       c. Set bits 1:0 of the IRS register at AZBAR+60h[1:0]
       or      word ptr es:[ebx+High Definition Audio_MMIO_ICS], BIT1+BIT0 ; Send the command

;       d. Poll ICS register bits at AZBAR+68h[1:0] until they return 10b indicating
;         the verb has been sent to the codec and response data from codec is now valid.
PollDataValid:
        mov     al, byte ptr es:[ebx+High Definition Audio_MMIO_ICS]
        cmp     al, 01b
        jne     PollDataValid
;       e. Read IR register at AZBAR+64h, the dword data is the VendorID/Device
;         ID value returned by the codec
        mov     eax, dword ptr es:[ebx+High Definition Audio_MMIO_IR] ; eax=vendorID/deviceID
;-------------------------------------------------------------------------
;3.     Check against the list of supported vendor ID/Device ID combinations
;       to determine if the received VID/DID is supported.
;-------------------------------------------------------------------------
        push    ecx

        call    CheckforValidCodec
        or      cx, cx
            jz      VerbTableDone               ; jump if VID/DID not supported
;-------------------------------------------------------------------------
;4.     If there is a match, send the entire list of verbs in the matching verb
;       table one by one to the codec
```

```
;   a. Poll the ICB bit of the ICS register at AZBAR+68h[0] until it returns 0.
;   b. Write the next verb (dword) in the table to the IC register at AZBAR+60h.
;   c. Write the bits of the ICS register at AZBAR+68h[1:0] to 11b to send the
;      verb to the codec.
;   d. Repeat steps 4a-4c until all verbs in the table have been sent for the
;      current codec.
;-------------------------------------------------------------------------------
;       a. Poll the ICB bit of the ICS register at AZBAR+68h[0] until it returns 0.

        push    cx
        xor     cx, cx
PollICBBit2:
        test    word ptr es:[ebx+High Definition Audio_MMIO_ICS], High Definition
Audio_MMIO_ICS_ICB
        jz      ICBBit2                 ; Poll ICB bit until it returns 0
        loop    PollICBBit2
;       Add error handling code here
ICBBit2:
        pop     cx
;       b. Write the current verb (dword) in the table to the IC register at AZBAR+60h.
        mov     eax, dword ptr cs:[si]
        mov     dword ptr es:[ebx+High Definition Audio_MMIO_IC], eax ; Write verb
;       c. Write the bits of the ICS register at AZBAR+68h[1:0] to 11b to send the
;          verb to the codec.
        or      word ptr es:[ebx+High Definition Audio_MMIO_ICS], BIT1+BIT0
;       d. Repeat steps 3a-3d until all verbs in the table have been sent for the
;          current codec.

        loop    PollICBBit2             ; Continue until all verbs written
```

```
VerbTableDone:
      pop     ecx
NextSDI:
      or      dx, dx
      jnz     InitCurrentCodec

High Definition AudioCodecComplete:
      ret
InitializeHigh Definition AudioCodecs  ENDP
;-----------------------------------------------------------------------------
; Procedure:CheckforValidCodec
; Description:Detects whether the vendor and device ID of the current codec
;             is supported based on whether the value is found at the start
;             of any of the codec verb tables.
; Input:   EAX     - Vendor and device ID of the current codec
;             ECX     - Current codec address
;             DS - BDA_DSEG.
;                     ES        - 0000h with 4GB limit.
;                     FS        - POST_DSEG.
;                     GS        - RUN_CSEG.
;                     STACK  - Available.
; Output:CX      - Size of codec verb table (in dwords) if a valid
;                                 codec is present.  Else cx = 0.
;             SI      - Adddress of the codec verb table (valid if CF=0)
; Modified:SI
;-----------------------------------------------------------------------------
CheckforValidCodec      PROC NEAR PUBLIC
      push    bx
```

```
        push    edx
        push    si

        xor     bx, bx
CheckNextCodecTable:
        mov     si, word ptr cs:[bx+offset CodecVerbTableList]
        cmp     dword ptr cs:[si], eax
        je      FoundValidCodec

;       end of table?
        add     bx, 2                   ; Next verb table entry
        cmp     bx, (offset CodecVerbTableListEnd - offset CodecVerbTableList)
        jb      CheckNextCodecTable
CodecNotValid:
            xor     cx, cx
        jmp     CodecCheckDone
FoundValidCodec:
        mov     edx, dword ptr cs:[si+VerbHeaderSize]       ; Get first verb
        shr     edx, 28
        cmp     edx, ecx                ; Is the codec address correct?
        jne     CodecNotValid
        add     si, 6
        call    GetVerbTableSize            ; Codec has valid DID/VID and addr
CodecCheckDone:
        pop     si
        pop     edx
        pop     bx
        ret
CheckforValidCodec      ENDP
```

```
;------------------------------------------------------------------------
; Procedure:GetVerbTableSize
; Description:Checks the front panel sensing GPIO to determine if front
;             panel jacks are present.  The routine returns the size of
;             the verb table (size may depend on whether front panel is
;             supported or if the codec supports front panel).
; Input:   SI     - Adddress of the front panel supported status byte
;             DS - BDA_DSEG.
;                     ES        - 0000h with 4GB limit.
;                     FS        - POST_DSEG.
;                     GS        - RUN_CSEG.
;                     STACK  - Available.
; Output:CX      - Size of codec verb table in dwords
;             SI     - Adddress of the codec verb table
; Modified:EBX, CX, SI
;------------------------------------------------------------------------
GetverbTableSize       PROC NEAR PUBLIC
      push    ebx

      mov     cl, byte ptr cs:[si]          ; al = Front panel support bit
      inc     si

      or      cl, cl
      mov     cx, word ptr cs:[si]          ; cx = length of rear panel table
      jz      FPSupportDone                 ; If front panel not supported
                                            ;  by the codec, no need to add
                                            ;  FP table size
;     TODO: OEMs must add code here to query the GPIO dedicated to front
;       panel sensing.
```

```
        jz      FPSupportDone
;       If control comes here, front panel jack is supported by the codec and
;         is present in the system, so add the size of the FP table.
        add     cx, word ptr cs:[si+2]          ; cx = rear panel table size +
                                                ;    front panel table size


FPSupportDone:


        add     si, 4                   ; si = start of codec verb table
        shl     cx, 2                   ; cx = # of Pin complexes * 4
                                        ;    = # of dwords in table
        pop     ebx
        ret
GetVerbTableSize        ENDP
CodecVerbTableList:
        dw offset VerbTable0
        dw offset VerbTable1
CodecVerbTableListEnd:
```

7.2 Intel® High Definition Audio Controller Configuration

Once the Intel High Definition Audio codec is determined to be present and Intel High Definition Audio controller is kept enabled via Function Disable register (RCBA+ 3418h[4]=1), BIOS should:

- Initialize the configuration space of Intel High Definition Audio controller as a regular PCI device (assign memory and interrupt resources and enable the device using standard PCI command register 04h).
- Initialize SSID/SVID registers at D27:F0:Reg2C-2Fh to OEM-specific IDs. This is similar to the SSID/SVIDs given to other ICH6 devices such as IDE, SATA, SMBus, USBs, etc.

7.2.1 Intel® High Definition Audio PME Event

Although it is a PCI Express Root Complex Integrated endpoint, the Intel High Definition Audio controller in the ICH6 is not capable of supporting the native PME software model. Its PME is supported in the same manner as a PCI PME.

The AC97_STS/AC97_EN bit-pair in ICH6 GPE0 register (PMBase+28h[5], PMBase+2Ch[5]) is shared between AC97 and Intel High Definition Audio PME event, depending on which one of the two devices is selected and enabled by BIOS. To support this PME feature in ACPI OS environment, BIOS needs to provide the proper _PRW object and _GPE._L05() control method in the ACPI name space. Below is an example of the ACPI name space for Intel High Definition Audio/AC97 PME support:

```
Scope (\_SB)
{

Device (PCI0)          // PCI Bus0
{
Name (_BBN, 0)   //   Bus Number of PCI0
Method(_PRT, 0) // _PRT package  for PCI0
……..
Device (AC97)     //  AC97 controller
{
Name(_ADR, 0x001E0002)           // Device30:Func2
Name ( _PRW, Package () {0x05,0x03} )    // PME Wake capability reporting
…….. }  // End AC97
Device (AZAL)               //  High Definition Audio controller
{
Name(_ADR, 0x001B0000)                        // Device27:Func0
Name ( _PRW, Package () {0x05,0x03} )    // PME Wake capability reporting
…….. // End AZAL
}   //  End Device PCI0
}   //  End \_SB scope
Scope (\_GPE)     // GPE event handlers
{
Method (_L05, 0)            //  High Definition Audio/AC97 PME event handler
{
//If High Definition Audio is the enabled controller
//       Notify (\_SB.PCI0.AZAL, 0x02)          // notify wake event
//Else
//       Notify (\_SB.PCI0.AC97, 0x02)          // notify wake event
// End of _L05

// End of \_GPE scope
```

§