a

AD1812

SoundPort Controller

Technical Reference

82-000782-02

You may contact the Computer Products Division in the following ways:

- By contacting your local Analog Devices Sales Representative or Authorized Distributor
- For Marketing information, call (617) 461-3881 in Norwood, Massachusetts, USA
- For Applications Engineering information, call (617) 461-3672 in Norwood, Massachusetts, USA ; applications e-mail address is as follows:
  dsp_applications@analog.com
- The Norwood office may also be reached by FAX: (617) 461-3010, Telex: 924491, TWX: 710/394-6577, Cables: ANALOG NORWOODMASS
- The Computer Products Division runs a Bulletin Board Service that can be reached at speeds up to 14,400 baud, no parity, 8 bits data, 1 stop bit, by dialing (617) 461-4258. This BBS supports: V.32bis, error correction (V.42 and MNP classes 2, 3, and 4), and data compression (V.42bis and MNP class 5).
- The Computer Products Division runs an FTP site that can be reached by typing:
  ftp ftp.analog.com or ftp 137.71.23.21
- By writing to:
  Analog Devices
  Computer Products Division
  One Technology Way, P.O. Box 9106
  Norwood, MA 02062-9106
  USA
- The Computer Products Division Technical Publications staff appreciates your comments on this and other product documentation; the publication group's e-mail address is as follows:
  dsp_techpubs@analog.com

AD1812 SoundPort Controller
Technical Reference

PRINTED IN USA Second Edition (Latest Revision : October 27, 1995)

Contents

LIST OF FIGURES & TABLES

List Of Figures & Tables

Chapter 1 Introduction

1.1 Overview

The purpose of the AD1812 SoundPort Controller User's Manual is to provide an audience of programmers, engineers, and OEMs with the information required to program the chip, design PC motherboards (and plug-in boards) using the chip, and manufacture AD1812 based boards.

The AD1812 SoundPort Controller integrates codec, synthesis, and bus interface functions into a single chip audio sub-system — adding low cost 16-bit stereo audio to any ISA bus equipped PC. Motherboard or PC plug-in designs using the AD1812 are compatible with virtually all applications written to meet Sound Blaster® Pro, Ad Lib™, MIDI MPU-401, Windows 95®, and the Microsoft® Windows™ Sound System standards.

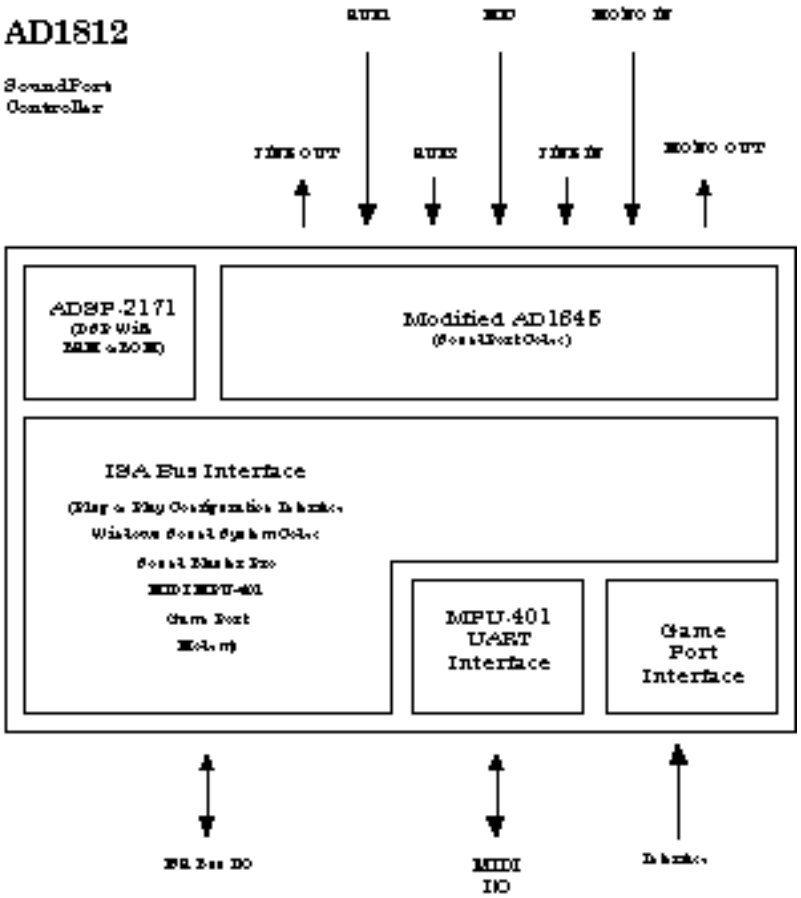


Figure 1.1 AD1812 SoundPort Controller Block Diagram

Figure 1.1 shows the AD1812 audio sub-system combines the following components:

- ADSP-2171 DSP with RAM & ROM
- AD1845 SoundPort Stereo Codec
- Plug & Play Compatible ISA Bus Interface

This combination provides five Plug & Play logical devices:

- Windows Sound System Codec (application-level compatible)
- Sound Blaster Pro (register level compatible
- AdLib (OPL-3 FM parameter compatible music synthesizer)
- MIDI MPU-401 Port
- Game/Joystick Port
- Modem

The AD1812 contains a register set that corresponds to those used for Sound Blaster Pro, Windows Sound System, MIDI MPU-401, Game/Joystick Port, and Plug & Play Configuration.

All logical devices in the AD1812 are compliant with the Intel/Microsoft Plug & Play Specification. Devices (Windows Sound System, MIDI, etc. …) are automatically configured at system start. Through the Plug & Play mechanism, the operating system can reconfigure the AD1812, avoiding conflicts with other hardware devices. The AD1812 also can be programmed in a non-Plug & Play environment.

The PC plug-in card shown in Figure 1.2 illustrates how few additional components are needed to build an AD1812 board. Only a power Amp and a few discrete components are needed to build an AD1812 plug-in card — even fewer are required to add an AD1812 to a motherboard design. Table 1.1 contains a summary of the controller's features.

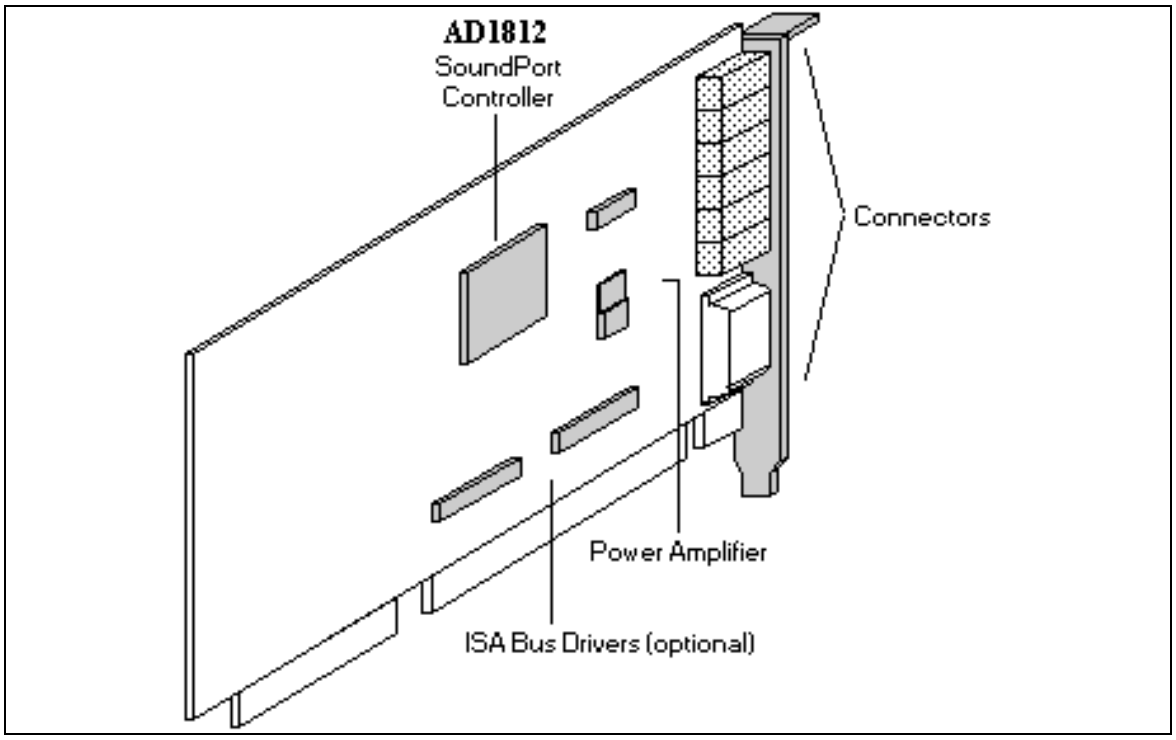


Figure 1.2 AD1812 PC Plug-In Card

Table 1.1 AD1812 SoundPort Controller Features

• Single-Chip Integration	- 16-bit Stereo Codec - Music Synthesizer - 16-bit ISA Bus Interface - Power Management - Game Port
• Industry Standards Compliance	- Sound Blaster Pro™ - Microsoft® Windows™ Sound System - General MIDI & MPU-401 - Plug & Play Configurable
• Available Software Support	- Microsoft® Windows™ 3.1 Drivers - Microsoft® Windows™ 95 Drivers - Windows™ Control & Diagnostic Applets
• Integrated Codec Advantages	- Full-Duplex Operation - Dynamic Sample Rate Variation from 5 kHz to 50 kHz (in 1 Hz increments)

1.2 AD1812 System Architecture

As shown in Figure 1.3, the AD1812 SoundPort Controller's architecture connects a set of Plug & Play logical devices to the PC ISA bus. These devices include a Sound Blaster Pro compatible device (SB), an AdLib compatible device (ADLIB), a Windows Sound System device (WSS), a MIDI MPU-401 compatible device (MIDI), and a Game Port device (GAME). The AD 1812 also provides a Plug and Play ISA bus interface for an external modem chipset. To provide these devices, the SoundPort Controller's architecture combines a codec, digital signal processor, and an ISA bus interface on a single chip.

The AD1812's PC ISA bus interface (INTERFACE) connects the DOS games register set, Windows Sound System register set, and music synthesis hardware to the ISA bus using a fully compatible Plug & Play (PnP) configuration interface. Hardware support in the AD1812 includes address decoding for on-chip devices, control & signal interpretation, DMA selection & control logic, IRQ selection & control logic, and interface configuration logic.

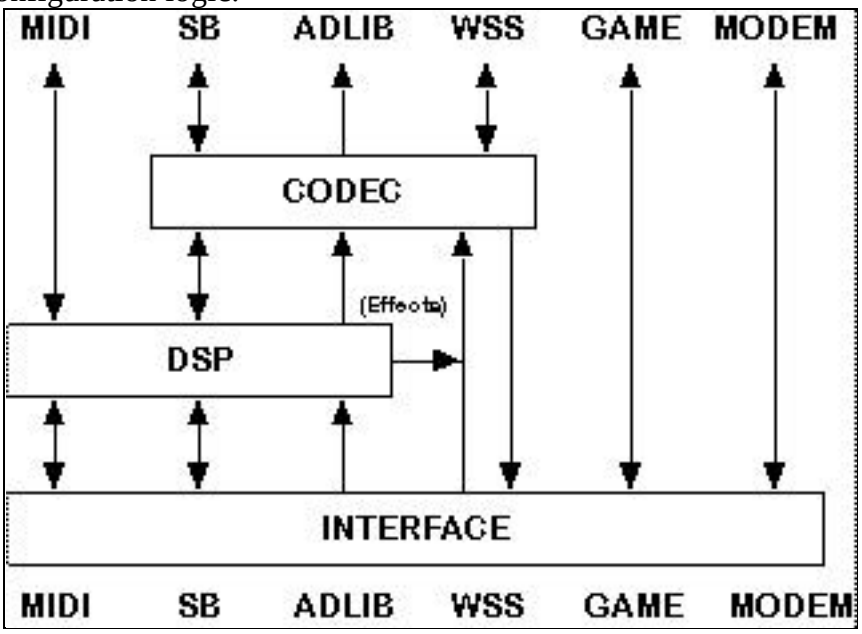


Figure 1.3 AD1812 SoundPort Controller System Architecture

- The three functional units (Codec, DSP, & Interface) that make up the AD1812 controller have the following architectural features.

- *Codec*
  A modified AD1845 stereo-audio 16-bit ΣΔ codec is integrated into the AD1812 controller. This codec provides support for business audio and multi-media applications with stereo audio converters, complete on-chip filtering (some external capacitors required) MPC Level-2 compliant analog mixing, programmable gain & attenuation, a variable sample-rate frequency generator, and FIFO buffers.

The codec's architecture includes a stereo pair of ΣΔ analog-to-digital converters (ADC) and a stereo pair of ΣΔ digital-to-analog converters (DAC). You can select inputs to the ADC from four stereo pairs of analog signals: line (LINE), microphone (MIC), auxiliary line #1 (AUX1), and post-mixed DAC output. An analog mixer lets you mix mono (MONO_IN), MIC, AUX1, LINE and auxiliary line #2 (AUX2) inputs with the DACs' output. Independent gain for each channel going into the ADCs is available through a software-controlled programmable gain stage. This architecture also lets you digitally mix the ADCs' output with the DACs' input. Also, the codec includes a variable sample rate frequency generator that lets you instantaneously change the codec's sample rate with a resolution of 1 Hz. Creating audio special effects (like Doppler-effect bullet-shots) without any clicks or pops is easy with this feature. The codec uses the variable sample rate frequency generator to derive all internal clocks from a single, 14.31818 MHz clock input.

The integrated codec supports a DMA request/grant (or a Programmed I/O mode) architecture for transferring data on the ISA bus. Dual DMA count registers in the AD1812 provide for full-duplex operation, enabling simultaneous capture and playback on separate DMA channels (8 or 16-bit).

- *DSP*
  An ADSP-2171 Digital Signal Processor (DSP with internal ROM & RAM) is integrated into the AD1812 controller. Software running on this internal signal processor provides sound effects and music synthesis by emulating the responses of the Yamaha YM262 (OPL-3) FM synthesis chip. Using this technique, the internal signal processor delivers 20-voice, 11-note polyphony. Music synthesized on the signal processor is converted on an additional pair of ΣΔ DACs and sent to the analog mixer on the codec.

- *Interface*
  An ESC615 ISA bus interface chip (with added Plug & Play interface) is integrated into the AD1812 controller. This interface's architecture also supports Sound Blaster Pro DMA transfers (separate channel from codec).

The controller's interface complies with the Intel/Microsoft Plug & Play specification, making all on-chip features automatically configurable. For compatibility with non-Plug & Play systems, you can disable the Plug & Play protocol and use a non-Plug & Play software configuration utility. In either case, the controller is completely software configurable. To provide Sound Blaster Pro, AdLib, and MIDI MPU-401 support, the interface includes registers that emulate those used by these devices.

These three architectural components (Codec, DSP, & Interface) are combined in the controller to support a broad range of PC audio conversion operations. An AD1812 based system can do the following audio operations simultaneously:

- Synthesize music and sound effects for games
- Convert compressed digital wave files into analog wave forms
- Digitize, format, and compress analog signals
- Receive and transmit MIDI MPU-401 UART information
- Communicate with an IBM compatible joystick

1.3 AD1812 Development

Virtually all applications developed for Sound Blaster, Windows Sound System, AdLib, and MIDI MPU-401 platforms run on the AD1812 SoundPort Controller. Follow the same development process for the controller as you would use for these other devices. This section provides information on related development kits, hardware/software specifications, and reference texts.

For information on AD1812 signal timing, mechanical, and electrical specifications, see the Analog Devices:

- *AD1812 SoundPort Controller Data Sheet*

As the AD1812 contains Sound Blaster (compatible) and Windows Sound System logical devices, you may find the following related development kits useful when developing AD1812 applications.

- *Developer Kit for Sound Blaster Series*, 2nd ed. ©1993, Creative Labs, Inc., 1901 McCarthy Blvd., Milpitas, CA 95035

- *Microsoft Windows Sound System Driver Development Kit (CD)*, Version 2.0, ©1993, Microsoft Corp., One Microsoft Way, Redmond, WA 98052

Because the AD1812 complies with the following related specifications, you can use them as an additional reference to AD1812 operations beyond the material in this manual.

- *Plug & Play ISA Specification, Version 1.0a*, ©1993,1994, Intel Corp. & Microsoft Corp., One Microsoft Way, Redmond, WA 98052

- *Multimedia PC Level 2 Specification*, ©1993, Multimedia PC Marketing Council, 1730 M St. NW, Suite 707, Washington, DC 20036

- *MIDI 1.0 Detailed Specification & Standard MIDI Files 1.0*, ©1994, MIDI Manufacturers Association, PO Box 3173 La Habra, CA 90632-3173

- *Recommendation G.711—Pulse Code Modulation (PCM) Of Voice Frequencies* (μ-Law & A-Law Companding), The International Telegraph and Telephone Consultative Committee IX Plenary Assembly Blue Book, Volume III - Fascicle III.4, General Aspects Of Digital Transmission Systems; Terminal Equipments, Recommendations G.700 - G.795, (Geneva, 1988), ISBN 92-61-03341-5

- *IMA Digital Audio Doc-Pac* (IMA-ADPCM), ©1992, Interactive Multimedia Association, 48 Maryland Avenue, Suite 202, Annapolis, MD 21401-8011

The following reference texts can serve as additional sources of information on developing applications that run on the AD1812.

- S. De Furia & J. Scacciaferro, *The MIDI Implementation Book*, (©1986, Third Earth, Pompton Lake)

- C. Petzold, *Programming Windows: the Microsoft guide to writing applications for Windows 3.1*, 3rd. ed. (©1992, Microsoft Press, Redmond)

- K. Pohlmann, *Principles of Digital Audio*, (©1989, Sams, Indianapolis)

- A. Stolz, *The Sound Blaster Book*, (©1993, Abacaus, Grand Rapids)

- J. Strawn, *Digital Audio Engineering An Anthology*, (©1985, Kaufmann, Los Altos)

- T. Yamamoto, *MIDI Guidebook*, 4th. ed., (©1987, 1989, Roland Corp., Japan)

1.4 Manual Conventions

Table 1.2 lists the text and layout conventions used in this manual. In addition to the conventions shown, please note the following euphemisms and acronyms:

- *Sound Blaster*, *Sound Blaster Pro*, *AdLib*, and *Music Synthesis* refer to DSP-based emulation of these devices through a combination of DSP software and dedicated hardware.

- *Windows Sound System Codec* refers to a modified AD1845 Codec integrated into the AD1812 SoundPort Controller that (with custom driver software) supports Windows Sound System applications.

- *DSP* stands for Digital Signal Processor.

Table 1.2 Manual Text & Symbol Conventions

Item	*Convention (how shown or used)*
Trademarks:	Registered trademarks (®) and trademarks (™) used in this document are *honored* in the front matter of this document
Registers:	Registers are documented with a labeled figure and a bit definition table
Bits:	Bits are indicated as part of a range of bits from upper bit to lower bit, [15:0] indicates bits 15 through 0.
Decimal base	Decimal base numbers are shown without a prefix: 2050
Hexadecimal base	Hexadecimal base numbers are shown with a 0x prefix: 0x000F
Binary base	Binary base numbers are shown with a 0b prefix: 0b1010,1010
Addresses: (PC and AD1812)	Addresses are shown as a hexadecimal numbers: 0x201 (Text with address indicates the type of address)
Bold Text:	Bold text is used to indicate **very important text**
Italic Text:	Italic text is used to indicate *important text*
Courier Text:	Courier text is used to indicate `program listings`
Tables:	Tables are shown in grid boxes with title at top
Figures:	Figures are shown in a box with title at bottom
Note:	Notes indicate information crucial to step, operation, or options
Caution:	Cautions indicate information crucial to avoiding component damage
Footnote:	Footnotes indicate information that is not required to operate the part but can be of some assistance

1.5 Manual Organization

This manual documents the AD1812 SoundPort Controller. Areas of particular interest to programmers and designers include the following:

- Chapter 2 AD1812 Programming
  The programming reference provides an overview of AD1812 programming, lists Plug & Play resource data for the AD1812's built-in logical devices, and describes how to configure/program the AD1812 when it is in non-Plug & Play mode. Also, this chapter includes descriptions of Windows Sound System Codec programming.

- Chapter 3 AD1812 Registers
  The register reference provides an overview of AD1812 registers with a map of all AD1812 registers and describes each register in detail. The manual presents register's descriptions in functional groups (Plug & Play, Windows Sound System, Sound Blaster Pro, MIDI, AdLib, and Game Port).

- Chapter 4 AD1812 Reference Design
  The design reference provides specifications for producing an AD1812-based PC plug-in card. These specifications include a functional overview, an architectural description, and board schematics.

1.6 Manual Errata

This is the second edition of the AD1812 SoundPort Controller Technical Reference. Errata in this manual is documented on the Computer Products Division Bulletin Board Service which can be reached at speeds up to 14,400 baud, no parity, 8 bits data, 1 stop bit, dialing (617) 461-4258. This BBS supports: V.32bis, error correction (V.42 and MNP classes 2, 3, and 4), and data compression (V.42bis and MNP class 5).

Chapter 2 AD1812 Programming

2.1 Overview

Programming the AD1812 SoundPort Controller consists of setting the controller's configuration and programming the controller's logical devices (Windows Sound System, Sound Blaster, AdLib, MIDI MPU-401, modem, and Game Port). This chapter describes AD1812 Plug & Play mode configuration data, non-Plug & Play mode configuration process, and procedures for programming the integrated Windows Sound System codec.

For information on the Plug & Play mode configuration process, see the *Plug & Play ISA Specification Version 1.0a (May 5, 1994)*. All the AD1812's logical devices comply with Plug & Play resource definitions described in the specification.

For information on vendor specific Plug & Play registers and all Windows Sound System codec registers, see Chapter 3, *AD1812 Registers*.

The next several sections use Plug & Play related terminology to describe configuration and programming topics. Definitions of terms are as follows:

- *Plug & Play Logical Devices*—Devices that meet the Plug & Play specification for runtime configuration in an ISA bus PC.
- *System Boot*—Steps a PC system goes through at power up (BIOS, POST, & BOOT)
- *System Resources*—Standard PC system resources (i.e. I/O addresses, interrupt channels, and DMA channels)
- *Plug & Play Resource ROM*—Read-Only-Memory in the AD1812 containing a list of possible resource settings for the on-chip logical devices.
- *Plug & Play Device IDs*—Product identifier that the system uses to find and start corresponding Plug & Play device drivers.
- *Plug & Play Resource Manager*—PC system software (Plug & Play BIOS or after Boot Plug & Play System Support) that provides runtime system configuration services.
- *Active Plug & Play Devices*—Active logical devices respond to all ISA bus cycles as per its normal operation. Inactive logical devices do *not* respond to *nor* drive any ISA bus signals.

- *Dependent functions*—Encoded interdependent options within a logical device's configuration resource data. The Plug & Play system software weighs these options, when configuring the system, to achieve the highest rating. Device configurations involving dependent functions can have the following ratings:

  A) *Good Plug & Play Configuration*—Configuration that best serves the software using the device

  B) *Acceptable Plug & Play Configuration*—Configuration that satisfactorily serves the software using the device

  C) *Sub-optimal Plug & Play Configuration*— Configuration that barely serves the software using the device

Note: The *Good*, *Acceptable*, and *Sub-optimal* ratings of dependent function configurations come from a variety of sources (i.e. most commonly recognizable I/O base addresses for devices, most useful DMA channel assignments, etc. …). Not all devices have what could be considered *Good* configurations, but all devices do have *Acceptable* configurations.

2.2 AD1812 Plug & Play Device Configuration

The operating system configures/re-configures AD1812 Plug & Play Logical Devices after system boot♣. To complete this configuration, the system reads resource data from the AD1812's on-chip resource ROM and from any other Plug & Play cards in the system, then arbitrates the configuration of system resources with a heuristic algorithm. The algorithm maximizes the number of *active* devices and the *acceptability* of their configurations.

The system considers all Plug & Play logical device resource data at the same time and makes a conflict-free assignment of resources to the devices. If the system cannot assign a conflict-free resource to a device, the system does not configure or activate the device. All configured devices are activated.

The system's Plug & Play support selects all necessary drivers, starts them, and outputs a list of system resources allocated to each logical device. Optionally, you can re-assign system resources at runtime with a Plug & Play Resource Manager. The custom setup created using the manager can be saved and used automatically on following system boots.

♣ There are no "boot-devices" among the Plug & Play Logical Devices in the AD1812. Non-Plug & Play BIOS systems configure the AD1812's Logical Devices after boot using drivers. Depending on BIOS implementations, Plug & Play BIOS systems *may* configure the AD1812's Logical Devices before POST or after Boot. See the *Plug & Play ISA Specification Version 1.0a* for more information on configuration control.

Plug & Play Device IDs (embedded in the logical device's resource data) provide the system with the information required to find and load the correct device drivers. In one case, a custom driver is required; the AD1812 Window Sound System driver from Analog Devices is required for correct operation. In all other cases (Sound Blaster, AdLib, MIDI, Game Port and modem), the Analog Devices custom driver is preferred, but the system can use generic drivers (that ship with all Plug & Play systems) if the custom driver is unavailable. Table 2.1 lists the AD1812's logical devices and compatible Plug & Play device drivers.

Table 2.1 AD1812 Logical Devices And Compatible Plug & Play Device Drivers

Device Name	*ADI (Device ID) & Driver Name*	*Compatible (Device ID) & Driver Name*
Window Sound System	(ADS7140) ADS7140 Windows Sound System	*(None)* *None*
Sound Blaster Pro (Compatible)	(ADS7141) ADS7141 Sound Blaster Pro	(PNPB002) PNPB002 Sound Blaster Pro
AdLib (Compatible)	(ADS7142) ADS7142 Yamaha OPL3-compatible FM synthesis device	(PNPB020) PNPB020 Yamaha OPL3-compatible FM synthesis device
MIDI (Compatible)	(ADS7143) ADS7143 MPU401 compatible	(PNPB006) PNPB006 MPU401 compatible
Game Port	(ADS7144) ADS7144 Joystick/Game port	(PNPB02F) PNPB02F Joystick/Game port
Modem	(ADS7145) ADS7145 Modem	(PNP0501) PNP0501 Compatible COM port

When the AD1812's PnP pin is asserted, the chip is in Plug & Play mode. The configuration process for the logical devices on the AD1812 is described in the *Plug & Play ISA Specification Version 1.0a (May 5, 1994)*. The specification describes how to transfer the logical devices from their startup *Wait For Key* state to the *Config* state and how to assign I/O ranges, interrupt channels, and DMA channels.

Tables 2.2 and 2.3 list the Plug & Play ISA bus registers and configuration data for the AD1812 in Plug & Play mode. The resource data presented in these tables corresponds to data described in the AD1812 Resource ROM.

Table 2.2 Plug & Play ISA Bus Registers (PnP pin asserted)

Port Name	*ISA Address*	*Type*
ADDRESS	0x279 (Printer status port)	Write-only
WRITE_DATA	0xA79 (Printer status port + 0x800)	Write-only
READ_DATA	Relocatable in range 0x203 - 0x3FF	Read-only

Table 2.3 Plug & Play AD1812 Logical Devices-Descriptors-Configurations

Device	*Descriptor*	*Configuration*
Windows Sound System (LDN==0)	I/O Port Address Descriptor 0 (0x60-0x61)	The Windows Sound System address range is from 0x0008 to 0xFFF8. The most commonly used address is 0x530. The range is 8 bytes long and must be aligned to an 8 byte memory boundary.
	Interrupt Request Level Select 0 (0x70)	The Windows Sound System requires one of the following IRQ channels: 3, 4, 5, 7, 9, 10, 11, or 12.
	DMA Channel Select 0 (0x74), DMA Channel Select 1 (0x75)	The Windows Sound System requires one or two DMA channels. Possible configurations are as follows: *Two 16-bit DMA channels*, selected from channels 5, 6, or 7. This is the Plug & Play *Good* configuration. *One 16-bit DMA playback channel and one 8-bit DMA capture channel*, selected from 16-bit channels 5, 6, or 7 and 8-bit channels 0, 1, or 3. This is a Plug & Play *Acceptable* configuration. *Two 8-bit DMA channels*, selected from channels 0, 1, or 3. This is a Plug & Play *Acceptable* configuration. *One 16-bit DMA channel*, selected from channels 5, 6, or 7. Because the channel is shared for capture and playback operations, simultaneous playback and capture is *not* possible in this configuration. This a Plug & Play *Sub-optimal* configuration. *One 8-bit DMA channel*, selected from channels 0, 1, or 3. Because the channel is shared for capture and playback operations, simultaneous playback and capture is *not* possible in this configuration. This a Plug & Play *Sub-optimal* configuration.

Table 2.3 Plug & Play AD1812 Logical Devices-Descriptors-Configurations (continued)

Device	*Descriptor*	*Configuration*
Sound Blaster Compatible (LDN==1)	I/O Port Address Descriptor 0 (0x60-0x61)	The Sound Blaster address range is from 0x010 to 0x3F0. The range is 16 bytes long and must be aligned to a 16 byte memory boundary. In Plug & Play terms, a Sound Blaster Configuration that has a base address of *0x300 or 0x330* is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x010 to 0x3F0 (other than 0x220 or 0x240) because most games will not recognize these non-standard ports.
	Interrupt Request Level Select 0 (0x70)	The Sound Blaster requires one of the following IRQ channels: 3, 4, 5, 7, 9, 10, 11, or 12.
	DMA Channel Select 0 (0x74)	The Sound Blaster requires one 8-bit DMA channel. Possible choices are (0,1,3).
AdLib (LDN==2)	I/O Port Address Descriptor 0 (0x60-0x61)	The AdLib address range is from 0x008 to 0x3F8. The range is 4 bytes long and must be aligned to an 8 byte memory boundary. In Plug & Play terms, an AdLib Configuration that has a base address of *0x388* is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x008 to 0x3F8 (other than 0x388) because most games will not recognize these non-standard ports.

Table 2.3 Plug & Play AD1812 Logical Devices-Descriptors-Configurations (continued)

Device	*Descriptor*	*Configuration*
MIDI MPU-401 (LDN==3)	I/O Port Address Descriptor 0 (0x60-0x61)	The MIDI address range is from 0x008 to 0x3F8. The range is 4 bytes long and must be aligned to a 16 byte memory boundary. In Plug & Play terms, a MIDI Configuration that has a base address of *0x300* or *0x330* is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x010 to 0x3F0 (other than 0x300 or 0x330) because most serial devices will not recognize these non-standard ports.
	Interrupt Request Level Select 0 (0x70)	The MIDI requires one of the following IRQ channels: 3, 4, 5, 7, 9, 10, 11, or 12.
Game Port (LDN==4)	I/O Port Address Descriptor 0 (0x60-0x61)	The Game Port address range is from 0x001 to 0x3FF. The range is 1 byte long. In Plug & Play terms, a Game Port Configuration that has a base address of 0x201 is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x001 to 0x3FF (other than 0x201) because most games will not recognize these non-standard ports.
Modem (LDN==5)	I/O Port Address Descriptor 0 (0x60-0x61)	The Modem device address range is from 0x008 to 0x3F8. The range is 8 bytes long and must be aligned to an eight byte memory boundary. In Plug & Play terms, a Modem Configuration that has a base address of 0x3F8 (with IRQ4), 0x2F8 (with IRQ3), 0x3E8 (with IRQ4), or 0x2E8 (with IRQ3) is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x008 to 0x3F8 (other than above combinations) because most serial devices will not recognize these non-standard ports.
	Interrupt Request Level Select 0 (0x70)	The Modem requires one of the following IRQ channels: 3, 4, 5, 7, 9, 10, 11, or 12 (see I/O Port Address for address and interrupt combinations).

2.3 AD1812 Non-Plug & Play Device Configuration

Configuring the AD1812 in non-Plug & Play mode is similar to configuring the device when it is in Plug & Play mode. In non-Plug & Play mode, the logical devices (Windows Sound System, Sound Blaster, AdLib, MIDI, & Game Port) on the AD1812 are locked in *Config* mode, ready to be assigned configuration data. Note that in non-Plug & Play mode all configuration decisions are left to your configuration routine rather than the operating system's Plug & Play configuration process.

Use the following procedure to configure the AD1812 SoundPort Controller when the controller is in Non-Plug & Play mode (PnP pin is de-asserted).

1. Write the value 0x00 to the ADDRESS register (PC I/O Address 0x234)
   (Sets the AD1812 register index to Set RD_DATA port register)

   Write the value 0x87 to the WRITE_DATA register (PC I/O Address 0x235)
   (Sets the address for the READ_DATA register to PC I/O Address 0x21F (or any valid RD I/O address). See Table 2.4 for a description of how the address value in WRITE_DATA is calculated and address options.)

2. Write the value 0x02 to the ADDRESS register (PC I/O Address 0x234)
   (Sets the AD1812 register index to Config Control register)

   Write the value 0x01 to the WRITE_DATA register (PC I/O Address 0x235)
   (Resets all Plug & Play logical devices to power up values)

Note: Each AD1812 logical device varies slightly in configurable features. The AD1812 Logical Device Number register (AD1812 register index 0x07) lets you select a device to configure. *Once you have selected a device, AD1812 register indices 0x30 through 0xFF correspond to the configuration registers of that device.* The AD1812 register indices below 0x30 are always available because they are not indexed by the logical device register.

The series of steps that follow demonstrate the logical device selection/configuration process for the Windows Sound System. Use the data in Tables 2.4 through 2.10 (at the end of this procedure) to complete configuration steps for the other devices.

3. Write the value 0x07 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to Logical Device Number (LDN) register)

   Write the value 0x00 to the WRITE_DATA register (PC I/O Address 0x239)
   (Selects the Windows Sound System Plug & Play device to configure—Logical Device Number 0x00—other valid device numbers are 0x01 (Sound Blaster), 0x02 (AdLib), 0x03 (MIDI), and 0x04 (Game Port))

4. Write the value 0x60 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to IO_BASE, upper byte register for Windows Sound System)

   Write the value 0x05 to the WRITE_DATA register (PC I/O Address 0x239)
   (Sets upper byte of the IO_BASE register [upper byte of the logical device's PC I/O address] to 0x05)

   Write the value 0x61 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to IO_BASE, lower byte register for Windows Sound System)

   Write the value 0x30 to the WRITE_DATA register (PC I/O Address 0x239)
   (Sets lower byte of the IO_BASE register [lower byte of the logical device's PC I/O address] to 0x30—Step 4 sets the Windows Sound System address to PC I/O Address 0x530. For other I/O Address range options, see Table 2.5.)

5. Write the value 0x70 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to Interrupt level select 0 register for Windows Sound System)

   Write the value 0x0A to the WRITE_DATA register (PC I/O Address 0x239)
   (Sets the interrupt level to 10—see Table 2.5 for additional Interrupt request level options)

6. Write the value 0x74 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to DMA channel select 0 *capture* register for Windows Sound System)

   Write the value 0x05 to the WRITE_DATA register (PC I/O Address 0x239)
   (Sets the DMA channel select 0 to channel 5, 16-bit DMA *capture*—see Table 2.5 for additional DMA options)

Note: If you do not enable the *capture* DMA channel all DMA (both *capture* & *playback*) occurs on the *playback* DMA channel. Also, note that simultaneous capture and playback is *not* possible using single channel DMA mode.

7. Write the value 0x75 to the ADDRESS register (PC I/O Address 0x238)
   (Sets the AD1812 register index to DMA channel select 1 *playback* register for Windows Sound System)

   Write the value 0x06 to the WRITE_DATA register (PC I/O Address 0x239)
   (Sets the DMA channel select 0 to channel 6, 16-bit DMA *playback*—see Table 2.5 for additional DMA options)

At this point, you have configured the Windows Sound System device. The device is ready to be activated or an I/O port conflict check can be performed.

In Tables 2.4 through 2.10 the following acronyms describe register and bit types: (RO) Read Only, (WO) Write Only, (WM) Write only Momentary, and (RW) Read/Write.

Table 2.4 Non-Plug & Play ISA Bus Registers (PnP pin de-asserted)

Port Name	*PC I/O Address*	*Type*
ADDRESS	0x234	Write Only (WO)
WRITE_DATA	0x235	Write Only (WO)
READ_DATA	Relocatable in the range 0x203 - 0x3FF	Read Only (RO)

Table 2.5 Non-Plug & Play AD1812 ADDRESS Register Values

Register Name	*ADDRESS*	*Definition (value)*
Set RD_DATA Port	0x00 WO	Sets the address-value of the READ_DATA port. A write to this register must be performed before any reads. The contents of RD_DATA (bits [7:0]) correspond to part of (bits [9:2]) the PC I/O address of the READ_DATA port. The upper bits [15:10] of the PC I/O address contain zeros and the lower bits [1:0] contain ones. *These bits are not user accessible, **shown in gray type in example**↓.* The location of READ_DATA is Relocatable in the range 0x203 to 0x3FF. **Example**: To set the PC I/O address of the READ_DATA port to 0x21F, write 0x87 to the RD_DATA port register. The example below indicates the relationship between the bits in each register. Bit [7] of RD_DATA is set to 1 always. READ_DATA Address: 0b**0000,00**10,0001,11**11** = 0x21F RD_DATA contents: 0b1000,0111 = 0x87
Config Control	0x02 WM	Resets all logical devices. Write (1) to bit [0:0] to reset all AD1812 logical devices. The contents of the configuration registers are set to power up values. Logical devices are inactive and the I/O ranges are disabled (set to zero).
Logical Device Number (LDN)	0x07 RW	Selects the current logical device. All reads and writes of I/O, DMA, and interrupt configuration registers access the logical devices indexed by this register. AD1812 logical device number indices are: 0x00 (Window Sound System) 0x01 (Sound Blaster) 0x02 (AdLib Music synthesis) 0x03 (MIDI port) 0x04 (Game port) 0x05 (Modem)

Table 2.5 Non-Plug & Play AD1812 ADDRESS Register Values (continued)

Register Name	*ADDRESS*	*Definition (value)*
Powerdown	0x20 RW	Controls powerdown operations. (For bit definitions, see Chapter 3, *AD1812 Registers*)

Table 2.6 Non-Plug & Play AD1812 ADDRESS Register Values (Windows Sound System, LDN==0)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address. Write upper byte to 0x60—IO_BASE, bits [15:8] Write lower byte to 0x61—IO_BASE, bits [7:0] Note: The Windows Sound System address range is from 0x0008 to 0xFFF8. The most commonly used address is 0x530. The range is 8 bytes long and must be aligned to an 8 byte memory boundary.
Interrupt request level select 0	0x70 RW	Holds the PC interrupt level selection. Write IRQ value to bits [3:0] to set the PC interrupt level, where valid IRQ values are: 3, 4, 5, 7, 9, 10, 11, & 12 Note: An IRQ value of 0 represents no interrupt selection. The most commonly used IRQs are 10 & 11.

Table 2.6 Non-Plug & Play AD1812 ADDRESS Register Values (Window Sound System, LDN==0) (continued)

Register Name	*ADDRESS*	*Definition (value)*
Interrupt request type select 0	0x71 RO	Holds the PC interrupt sensitivity selection. This Read Only (RO) register contains the value 0x02, indicating that the interrupt sensitivity is set to active-high & edge-sensitive. The contents of this register cannot be changed and it is included only for Plug & Play compatibility.
DMA channel select 0	0x74 RW	Holds the *capture* DMA channel selection. Write channel value to bits [2:0], where valid channel values are: 0, 1, or 3 (for 8-bit DMA) 4 (for no DMA) 5, 6, or 7 (for 16-bit DMA) Note: In Plug & Play terms, a Windows Sound System Configuration that includes two 16-bit DMA channels is *Good*, one 16-bit & one 8-bit DMA channel or two 8-bit DMA channels is *Acceptable*, and one 16-bit DMA channel or one 8-bit DMA channel is *Sub-optimal*. If you do not enable the *capture* DMA channel all DMA (both *capture* & *playback*) occurs on the *playback* DMA channel. Also, note that simultaneous capture and playback is *not* possible using single channel DMA mode
DMA channel select 1	0x75 RW	Holds the *playback* DMA channel selection. Write channel value to bits [2:0], where valid channel values are: 0, 1, or 3 (for 8-bit DMA) 4 (for no DMA) 5, 6, or 7 (for 16-bit DMA)

Table 2.7 Non-Plug & Play AD1812 ADDRESS Register Values (Sound Blaster, LDN==1)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address (10-bit descriptor, bits [15:10] are zero). Write 2-bits to 0x60—IO_BASE, bits [9:8]. Write lower byte to 0x61—IO_BASE, bits [7:0]. Note: The Sound Blaster Pro address range is from 0x010 to 0x3F0. The range is 16 bytes long and must be aligned to a 16 byte memory boundary. In Plug & Play terms, a Sound Blaster Configuration that has a base address of 0x220 or 0x240 is *Acceptable* and *Sub-optimal* if between 0x010 to 0x3F0 (other than 0x220 or 0x240) because most games will not recognize these non-standard ports.

Table 2.7 Non-Plug & Play AD1812 ADDRESS Register Values (Sound Blaster, LDN==1) (continued)

Register Name	*ADDRESS*	*Definition (value)*
Interrupt request level select 0	0x70 RW	Holds the PC interrupt level selection. Write IRQ value to bits [3:0] to set the PC interrupt level, where valid IRQ values are: 3, 4, 5, 7, 9, 10, 11, & 12. Note: An IRQ value of 0 represents no interrupt selection. The most commonly used IRQs are 5 & 7.
Interrupt request type select 0	0x71 RO	Holds the PC interrupt sensitivity selection. This Read Only (RO) register contains the value 0x02, indicating that the interrupt sensitivity is set to active-high & edge-sensitive. The contents of this register cannot be changed and it is included only for Plug & Play compatibility.
DMA channel select 0	0x74 RW	Holds the PC DMA channel selections. Write channel value to bits [2:0], where valid channel values are: 1, 0, or 3 4 (for no DMA) The most commonly used DMA channel is 1.

Table 2.8 Non-Plug & Play AD1812 ADDRESS Register Values (AdLib Music Synthesis, LDN==2)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).

Table 2.8 Non-Plug & Play AD1812 ADDRESS Register Values (AdLib Music Synthesis, LDN==2) (continued)

Register Name	*ADDRESS*	*Definition (value)*
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address (10-bit descriptor, bits [15:10] are zero). Write 2-bits to 0x60—IO_BASE, bits [9:8]. Write lower byte to 0x61—IO_BASE, bits [7:0]. Note: The AdLib address range is from 0x008 to 0x3F8. The range is 4 bytes long and must be aligned to an 8 byte memory boundary. In Plug & Play terms, an AdLib Configuration that has a base address of 0x388 is *Acceptable* and *Sub-optimal* if between 0x008 to 0x3F8 (other than 0x388) because most games will not recognize these non-standard ports.

Table 2.9 Non-Plug & Play AD1812 ADDRESS Register Values (MIDI, LDN==3)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).

Table 2.9 Non-Plug & Play AD1812 ADDRESS Register Values (MIDI, LDN==3) (continued)

Register Name	*ADDRESS*	*Definition (value)*
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address (10-bit descriptor, bits [15:10] are zero). Write 2-bits to 0x60—IO_BASE, bits [9:8] Write lower byte to 0x61—IO_BASE, bits [7:0] Note: The MIDI address range is from 0x008 to 0x3F8. The range is 4 bytes long and must be aligned to a 16 byte memory boundary. In Plug & Play terms, a MIDI Configuration that has a base address of 0x330 is *Acceptable* and *Sub-optimal* if between 0x010 to 0x3F0 (other than 0x330) because most games will not recognize these non-standard ports.
Interrupt request level select 0	0x70 RW	Holds the PC interrupt level selection. Write IRQ value to bits [3:0] to set the PC interrupt level, where valid IRQ values are: 3, 4, 5, 7, 9, 10, 11, & 12 Note: An IRQ value of 0 represents no interrupt selection; IRQ 9 is the most commonly used.
Interrupt request type select 0	0x71 RO	Holds the PC interrupt sensitivity selection. This Read Only (RO) register contains the value 0x02, indicating that the interrupt sensitivity is set to active-high & edge-sensitive. The contents of this register cannot be changed and it is included only for Plug & Play compatibility.

Table 2.10 Non-Plug & Play AD1812 ADDRESS Register Values (Game Port, LDN==4)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address (10-bit descriptor, bits [15:10] are zero). Write 2-bits to 0x60—IO_BASE, bits [9:8]. Write lower byte to 0x61—IO_BASE, bits [7:0]. Note: The Game Port address range is from 0x001 to 0x3FF. The range is 1 byte long. In Plug & Play terms, a Game Port Configuration that has a base address of 0x201 is *Acceptable* and *Sub-optimal* if between 0x001 to 0x3FF (other than 0x201) because most games will not recognize these non-standard ports.

Table 2.11 Non-Plug & Play AD1812 ADDRESS Register Values (Modem, LDN==5)

Register Name	*ADDRESS*	*Definition (value)*
Activate	0x30 RW	Activates the device on the ISA bus. Write (1) to bit [0] to activate the device. Note: Do *not* enable Activate and I/O Range Check at the same time. Bits [7:1] of this register return (0) on reads. An inactive device does *not* respond to writes to its I/O address, interrupt (IRQ), or DMA request (DRQ).
I/O Range Check	0x31 RW	Checks for I/O port conflicts with other devices on ISA bus. Write (1) to bit [0] to select 0x55 return on device reads in the programmed range. Write (0) to bit [0] to select 0xAA return on device reads in the programmed range. Write (1) to bit [1] to enable I/O check. Note: Do *not* enable I/O Range Check and Activate at the same time. Bits [7:2] of this register return (0) on reads.
I/O port base address	0x60 RW 0x61 RW	Holds PC I/O base address (10-bit descriptor, bits [15:10] are zero). Write 2-bits to 0x60—IO_BASE, bits [9:8]. Write lower byte to 0x61—IO_BASE, bits [7:0]. Note: The Modem device address range is from 0x008 to 0x3F8. The range is 8 bytes long and must be aligned to an eight byte memory boundary. In Plug & Play terms, a Modem Configuration that has a base address of 0x3F8 (with IRQ4), 0x2F8 (with IRQ3), 0x3E8 (with IRQ4), or 0x2E8 (with IRQ3) is *Acceptable*. The configuration is *Sub-optimal* if using an address between 0x008 to 0x3F8 (other than above combinations) because most serial devices will not recognize these non-standard ports.

Table 2.11 Non-Plug & Play AD1812 ADDRESS Register Values (Modem, LDN==5) (continued)

Register Name	*ADDRESS*	*Definition (value)*
Interrupt request level select 0	0x70 RW	Holds the PC interrupt level selection. Write IRQ value to bits [3:0] to set the PC interrupt level, where valid IRQ values are: 3, 4, 5, 7, 9, 10, 11, & 12 (see I/O Port Address for address and interrupt combinations).
Interrupt request type select 0	0x71 RO	Holds the PC interrupt sensitivity selection. This Read Only (RO) register contains the value 0x02, indicating that the interrupt sensitivity is set to active-high & edge-sensitive. The contents of this register cannot be changed and it is included only for Plug & Play compatibility.

2.4 AD1812 Windows Sound System Codec Programming

Programmable sample rate, gain, attenuation, muting, mixing, and companding features of the AD1812's integrated codec provide many performance options when using the part. Built-in support for Direct Memory Accessing (DMA) and Programmed I/O (PIO) also provides several options for data transfer.

This section on codec programming provides an overview of the programmable features of the codec. The codec programming topics in this section include the following:

- *Codec Data Formats & Sequencing*—This section covers the data types and data transfer sequences supported by the codec.

- *Codec DMA & PIO Transfers*—This section provides procedural overviews of DMA & PIO transfers and modes.

- *Codec I/O Mixing, Gain, & Attenuation*—This section provides an overview of the programmable controls for the codec's mixing, gain, attenuation, and mute features.

- Codec Autocalibration—This section provides a procedural overview of codec autocalibration.

- *Codec Sample Rate Operations*—This section provides a procedural overview of the codec's programmable sample rate features.

- *Codec Powerdown Operations*—This section provides a procedural overview of the codec's programmable powerdown features.

2.4.1 Codec Data Formats & Sequencing

This section describes the data formats supported by the AD1812's integrated codec and the order in which the data can be sent during PIO and DMA transfers. Some terms that this section uses in describing the codec data formats and data transfer sequences are defined as follows:

- *Companding*—Compression/Decompression. The AD1812 supports µ-Law, A-Law, and IMA-ADPCM voice band audio compression/decompression.

- *Big Endian 16-Bit Signed (Two's Complement)*—16-bit Linear data format; the most significant (Upper) byte is sent first.

- *Little Endian 16-Bit Signed (Two's Complement)*—16-bit Linear data format; the least significant (Lower) byte is sent first.

- *Pulse Code Modulation (PCM)*—8-bit Linear data format

- *µ-Law & A-Law PCM*—8-bit Companded formats of PCM data (Defined by CCITT G.711 recommendation).

- *Interactive Multimedia Association-Adaptive Differential Pulse Code Modulation (IMA-ADPCM)*—4-bit Companded format of PCM Data

The codec supports three linear and three companded data formats. You can transfer data in any of the following formats:

- Linear, Big Endian 16-Bit Signed (Two's Complement)
- Linear, Little Endian 16-Bit Signed (Two's Complement)
- Linear, Unsigned, 8-Bit PCM
- µ-Law Companded, 8-Bit PCM
- A-Law Companded, 8-Bit PCM
- IMA-ADPCM Companded, 4-Bit PCM

Regardless of the data format used, the AD1812's codec always transfers 32-bits of data (two 16-bit words). The number of samples sent in each 32-bit packet (with no unused bits) varies with the data format (4, 8, or 16-bit, Signed, Unsigned, or Companded).

Program the DMA counters in the AD1812's codec with the number of SAMPLES to be transferred. The sample counters in the codec count the number of samples transferred, *not* the number of 32-bit packets.

Each time a 32-bit packet is transferred, the codec decrements the counters by the number of samples in the packet. To determine the number of transfers required to cause a counter underflow—indicated when the codec sets the INT bit—for a programmed sample count (Number of Samples) and data format (Samples per Packet), use the following formula:

$$Number\ of\ Transfers = \left\{ TRUNC \left(\frac{Number\ of\ Samples}{\left(\frac{Number\ of\ Samples}{Packet} \right)} \right) + 1 \right\}$$

Note: During playback DMA if the number of samples to be transferred is *not* evenly divisible by the number of samples in a packet, *fill the unused portion of the last packet with either a mid-scale value or the value of the last sample*

Example: If you program the transfer count to 7 (*for 7 samples*) in 8-bit mono format, how many packet transfers are required to send/receive the samples?

The codec sends/receives 4 samples in each 32-bit packet in 8-bit mono format. (For a complete list of sample to packet ratios in supported formats, see tables 2.11, 2.12, and 2.13.). From the truncated transfer count formula, you can calculate the number of packet transfers required to send the samples as follows:

$$Number\ of\ Transfers = \left\{ TRUNC \left(\frac{7\ Samples}{\left(\frac{4\ Samples}{Packet} \right)} \right) + 1 \right\} = 2$$

After two 32-bit transfers (four samples per transfer) in this case, the Current Count register underflows—and the codec generates an interrupt (sets INT bit).

Figures 2.1 through 2.6 display the relationship between 32-bit data transfers used by the codec and the number of samples per transfer sent in stereo, mono, 4-bit, 8-bit, and 16-bit modes. It is useful to compare the information displayed in these figures with data transfer sequences listed in Tables 2.12 through 2.14.

As shown in the tables, the number of samples passed in a 32-bit codec data transfer varies depending on the format of the data sent. Note that in mono mode the codec only uses the left channel.

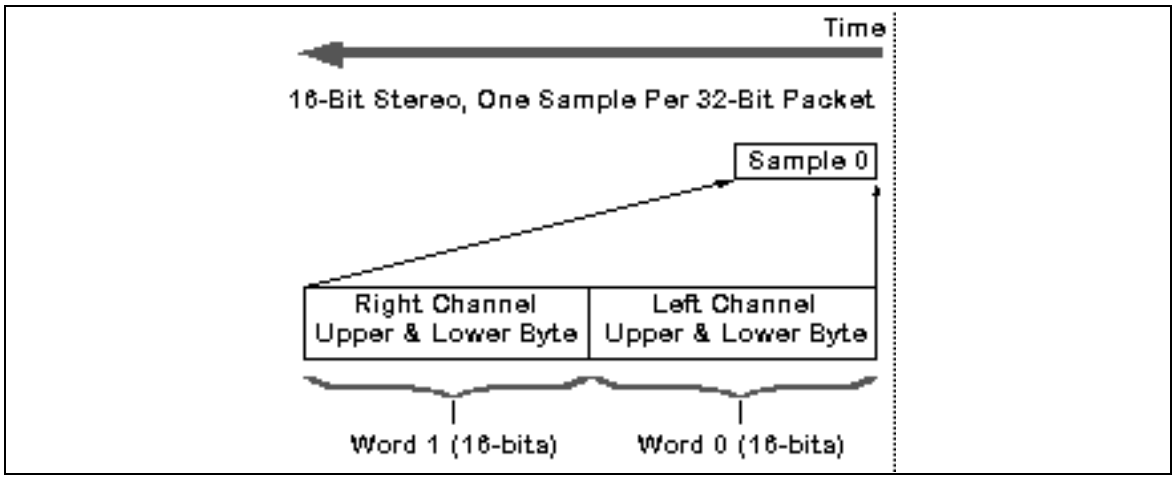


Figure 2.1 Stereo 16-Bit (Linear Big & Little Endian) Audio Data

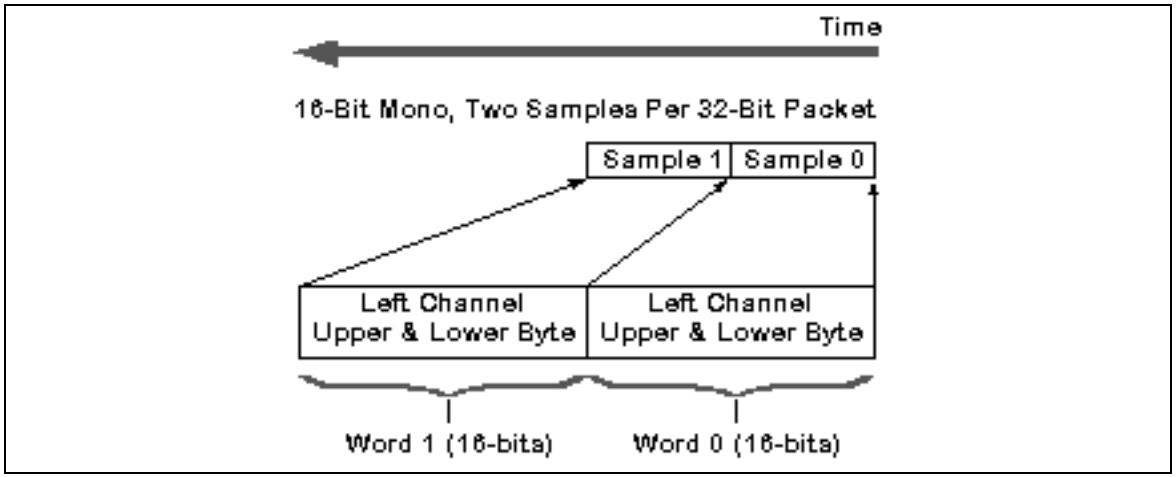


Figure 2.2 Mono 16-Bit (Linear Big & Little Endian) Audio Data

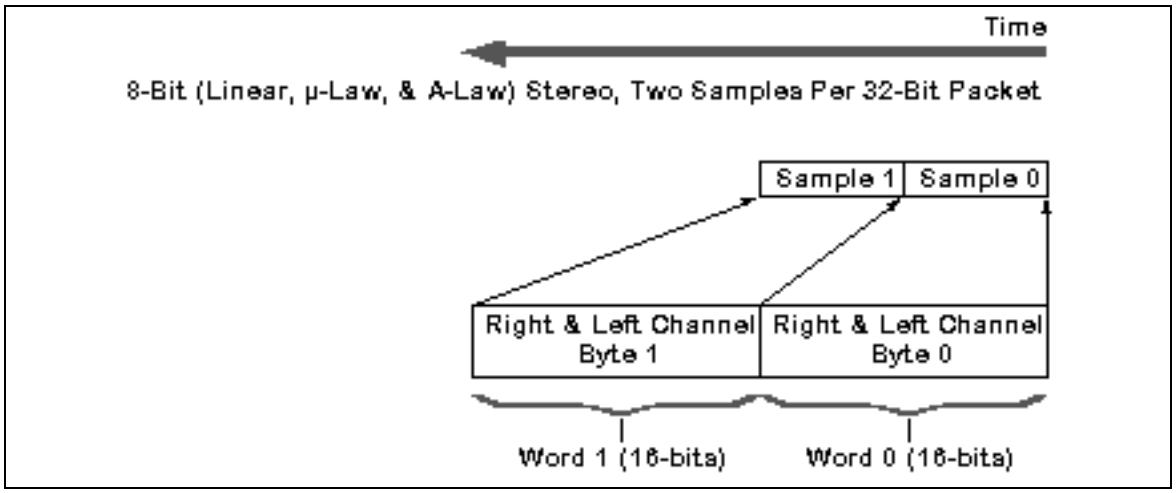


Figure 2.3 Stereo 8-Bit (Linear, μ-Law, & A-Law) Audio Data

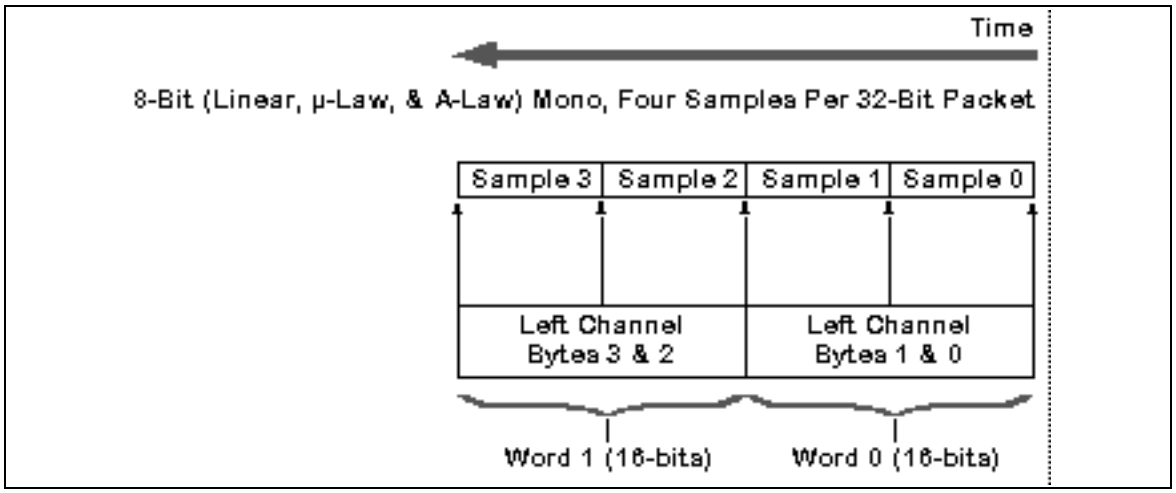


Figure 2.4 Mono 8-Bit (Linear, µ-Law, & A-Law) Audio Data

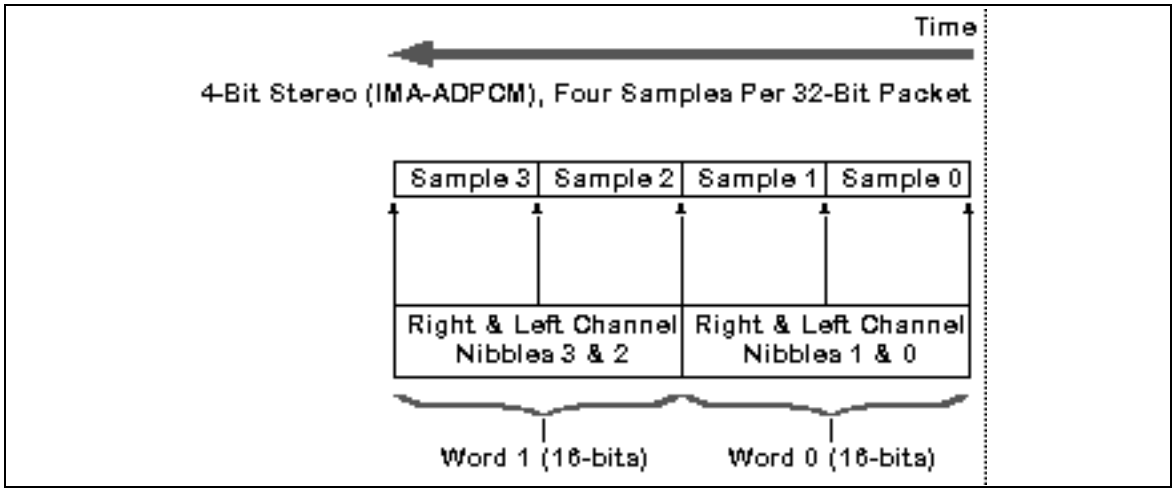


Figure 2.5 Mono 4-Bit (IMA-ADPCM) Audio Data

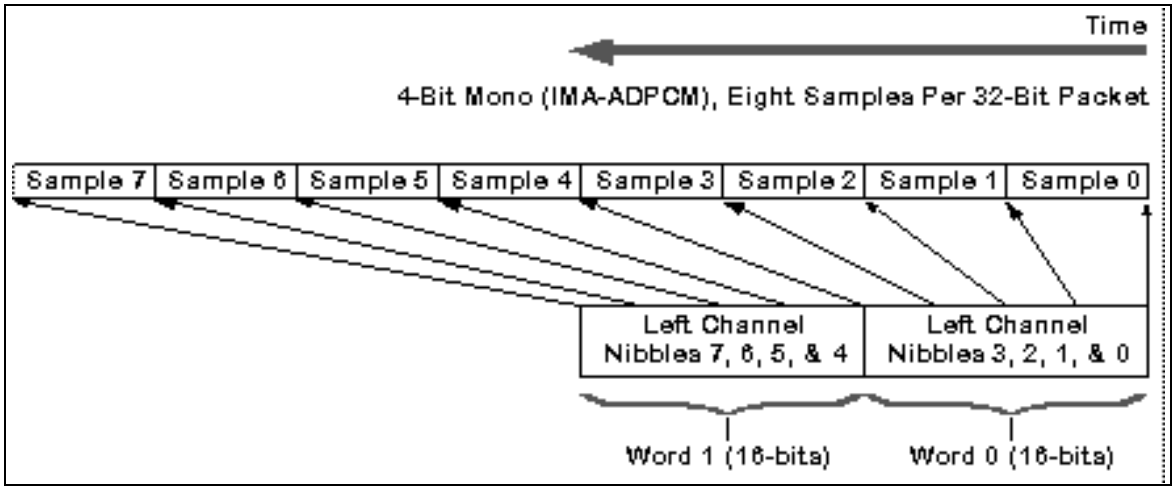


Figure 2.6 Mono 4-Bit (IMA-ADPCM) Audio Data

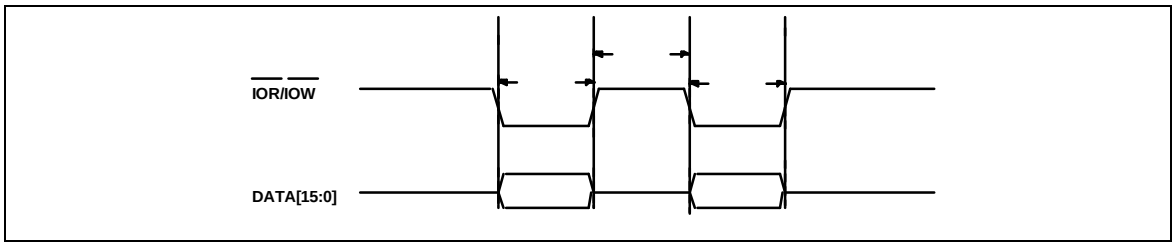


Figure 2.7 Codec Transfers 16-Bit Interface

Table 2.12 Codec Transfers 16-Bit Interface, No Byte Swap (P/CINF8=0, P/CBSW=0)

Format	*Word 1 (16-bit)*				*Word 0 (16-bit)*			
	MSB			*LSB*	*MSB*			*LSB*
Mono, 16-bit *Little Endian*	Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 0 Left Channel		Lower 8-bits of Sample 0 Left Channel	
Stereo, 16-bit *Little Endian*	Upper 8-bits of Sample 0 Right Channel		Lower 8-bits of Sample 0 Right Channel		Upper 8-bits of Sample 0 Left Channel		Lower 8-bits of Sample 0 Left Channel	
Mono, 8-bit *Linear PCM* *µ-Law PCM* *A-Law PCM*	Sample 3, 8-bits Left Channel		Sample 2, 8-bits Left Channel		Sample 1, 8-bits Left Channel		Sample 0, 8-bits Left Channel	
Stereo, 8-bit *Linear PCM* *µ-Law PCM* *A-Law PCM*	Sample 1, 8-bits Right Channel		Sample 1, 8-bits Left Channel		Sample 0, 8-bits Right Channel		Sample 0, 8-bits Left Channel	
Mono, 4-bit *IMA-ADPCM*	Sample 7, 4-bits Left Channel	Sample 6, 4-bits Left Channel	Sample 5, 4-bits Left Channel	Sample 4, 4-bits Left Channel	Sample 3, 4-bits Left Channel	Sample 2, 4-bits Left Channel	Sample 1, 4-bits Left Channel	Sample 0, 4-bits Left Channel
Stereo, 4-bit *IMA-ADPCM*	Sample 3, 4-bits Right Channel	Sample 3, 4-bits Left Channel	Sample 2, 4-bits Right Channel	Sample 2, 4-bits Left Channel	Sample 1, 4-bits Right Channel	Sample 1, 4-bits Left Channel	Sample 0, 4-bits Right Channel	Sample 0, 4-bits Left Channel
Mono, 16-bit *Big Endian*	Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 0 Left Channel		Upper 8-bits of Sample 0 Left Channel	
Stereo, 16-bit *Big Endian*	Lower 8-bits of Sample 0		Upper 8-bits of Sample 0		Lower 8-bits of Sample 0		Upper 8-bits of Sample 0	

Format	*Word 1 (16-bit)*		*Word 0 (16-bit)*	
	MSB	*LSB*	*MSB*	*LSB*
Mono, 16-bit Little Endian	Upper 8-bits of Sample 1 Left Channel	Lower 8-bits of Sample 1 Left Channel	Upper 8-bits of Sample 0 Left Channel	Lower 8-bits of Sample 0 Left Channel
	Right Channel	Right Channel	Left Channel	Left Channel

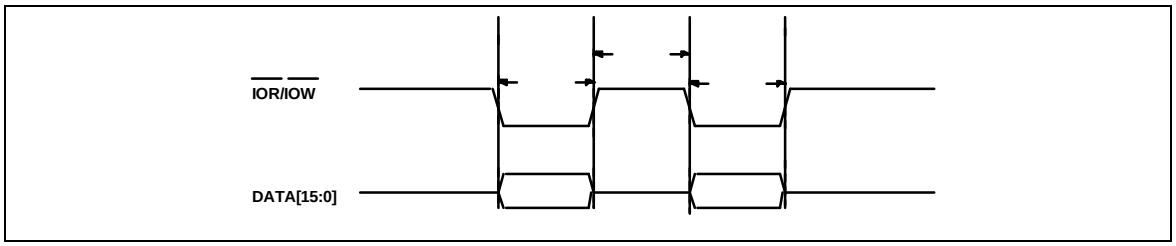


Figure 2.8 Codec Transfers 16-Bit Interface

Table 2.13 Codec Transfers 16-Bit Interface, With Byte Swap (P/CINF8=0, P/CBSW=1)

Format	*Word 1 (16-bit)*				*Word 0 (16-bit)*			
	MSB			*LSB*	*MSB*			*LSB*
Mono, 16-bit *Little Endian*	Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 0 Left Channel		Upper 8-bits of Sample 0 Left Channel	
Stereo, 16-bit *Little Endian*	Lower 8-bits of Sample 0 Right Channel		Upper 8-bits of Sample 0 Right Channel		Lower 8-bits of Sample 0 Left Channel		Upper 8-bits of Sample 0 Left Channel	
Mono, 8-bit *Linear PCM* *µ-Law PCM* *A-Law PCM*	Sample 2, 8-bits Left Channel		Sample 3, 8-bits Left Channel		Sample 0, 8-bits Left Channel		Sample 1, 8-bits Left Channel	
Stereo, 8-bit *Linear PCM* *µ-Law PCM* *A-Law PCM*	Sample 1, 8-bits Left Channel		Sample 1, 8-bits Right Channel		Sample 0, 8-bits Left Channel		Sample 0, 8-bits Right Channel	
Mono, 4-bit *IMA-ADPCM*	Sample 5, 4-bits Left Channel	Sample 4, 4-bits Left Channel	Sample 7, 4-bits Left Channel	Sample 6, 4-bits Left Channel	Sample 1, 4-bits Left Channel	Sample 0, 4-bits Left Channel	Sample 3, 4-bits Left Channel	Sample 2, 4-bits Left Channel
Stereo, 4-bit *IMA-ADPCM*	Sample 2, 4-bits Right Channel	Sample 2, 4-bits Left Channel	Sample 3, 4-bits Right Channel	Sample 3, 4-bits Left Channel	Sample 0, 4-bits Right Channel	Sample 0, 4-bits Left Channel	Sample 1, 4-bits Right Channel	Sample 1, 4-bits Left Channel
Mono, 16-bit *Big Endian*	Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 0 Left Channel		Lower 8-bits of Sample 0 Left Channel	
Stereo, 16-bit *Big Endian*	Upper 8-bits of Sample 0		Lower 8-bits of Sample 0		Upper 8-bits of Sample 0		Lower 8-bits of Sample 0	

Format	*Word 1 (16-bit)*		*Word 0 (16-bit)*	
	MSB	*LSB*	*MSB*	*LSB*
Mono, 16-bit Little Endian	Lower 8-bits of Sample 1 Left Channel	Upper 8-bits of Sample 1 Left Channel	Lower 8-bits of Sample 0 Left Channel	Upper 8-bits of Sample 0 Left Channel
	Right Channel	Right Channel	Left Channel	Left Channel

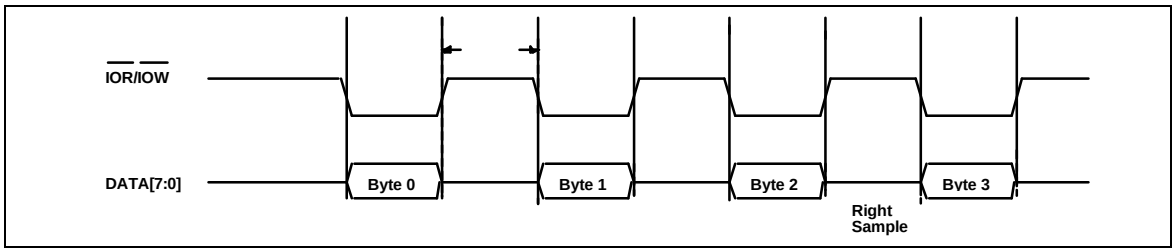


Figure 2.9 Codec Transfers 8-Bit Interface

Table 2.14 Codec Transfers 8-Bit Interface (P/CINF8=1)

Format	*Byte 3*		*Byte 2*		*Byte 1*		*Byte 0*	
	MSB	*LSB*	*MSB*	*LSB*	*MSB*	*LSB*	*MSB*	*LSB*
Mono, 16-bit Little Endian	Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 0 Left Channel		Lower 8-bits of Sample 0 Left Channel	
Stereo, 16-bit Little Endian	Upper 8-bits of Sample 0 Right Channel		Lower 8-bits of Sample 0 Right Channel		Upper 8-bits of Sample 0 Left Channel		Lower 8-bits of Sample 0 Left Channel	
Mono, 8-bit Linear PCM µ-Law PCM A-Law PCM	Sample 3, 8-bits Left Channel		Sample 2, 8-bits Left Channel		Sample 1, 8-bits Left Channel		Sample 0, 8-bits Left Channel	
Stereo, 8-bit Linear PCM µ-Law PCM A-Law PCM	Sample 1, 8-bits Right Channel		Sample 1, 8-bits Left Channel		Sample 0, 8-bits Right Channel		Sample 0, 8-bits Left Channel	
Mono, 4-bit IMA-ADPCM	Sample 7, 4-bits Left Channel	Sample 6, 4-bits Left Channel	Sample 5, 4-bits Left Channel	Sample 4, 4-bits Left Channel	Sample 3, 4-bits Left Channel	Sample 2, 4-bits Left Channel	Sample 1, 4-bits Left Channel	Sample 0, 4-bits Left Channel
Stereo, 4-bit IMA-ADPCM	Sample 3, 4-bits Right Channel	Sample 3, 4-bits Left Channel	Sample 2, 4-bits Right Channel	Sample 2, 4-bits Left Channel	Sample 1, 4-bits Right Channel	Sample 1, 4-bits Left Channel	Sample 0, 4-bits Right Channel	Sample 0, 4-bits Left Channel
Mono, 16-bit Big Endian	Lower 8-bits of Sample 1 Left Channel		Upper 8-bits of Sample 1 Left Channel		Lower 8-bits of Sample 0 Left Channel		Upper 8-bits of Sample 0 Left Channel	
Stereo, 16-bit Big Endian	Lower 8-bits of Sample 0		Upper 8-bits of Sample 0		Lower 8-bits of Sample 0		Upper 8-bits of Sample 0	

	Right Channel	Right Channel	Left Channel	Left Channel

2.4.2 Codec DMA & PIO Data Transfers

You can transfer data to and from the AD1812's codec using the chip's built-in support for Direct Memory Accessing (DMA) and Programmed Input/Output (PIO). This section describes the order of operations for both transfer methods. Note these procedures are subject to signal timing restrictions described in the *AD1812 SoundPort Controller Data Sheet.*

To initiate a DMA process using the AD1812's integrated codec, use the following steps:

1) Enter Mode Change Enable state by setting the MCE bit of Codec Index Address register. (You must put the codec in MCE state to modify the contents of Codec Indexed register 0x09—*except for the PEN and CEN bits*).

2) Select dual or single channel DMA by writing a 0 for dual or a 1 for single to the SDC bit in Codec Indexed register 0x09.

   Note that if a single DMA channel is configured for the AD1812 Codec, that channel is the playback channel; all DMA transfers (playback or capture) occur on the playback channel. The codec supports single channel DMA, dual channel DMA (full-duplex capture/playback), single channel DMA/PIO (one of each at the same time) and dual channel PIO modes. Use the PIO procedure (follows this DMA procedure) to set up a PIO channel.

3) Use steps (a) *and/or* (b) to set up DMA channels:

   a) Select (while the playback channel is disabled, PEN=0) the playback data format, stereo/mono, byte swapping, and interface with the PFMT, PC/L, PS/M, PBSW, and PINF8 bits in Codec Indexed register 0x08.

   b) Select (while the capture channel is disabled, CPEN=0) the capture data format, stereo/mono, byte swapping, and interface with the CFMT, CC/L, CS/M, CBSW, and CINF8 bits in Codec Indexed register 0x1C.

4) Select the sample rate with the FU and FL bits in Codec Indexed registers 0x16 and 0x17 (FU must be loaded before FL).

5) Load the Lower then Upper Base Count registers (Codec Indexed registers 0x0F then 0x0E) with the number of samples to playback. For capture, load the Capture Lower then Upper Base Count registers (Codec Indexed registers 0x1F then 0x1E).

   (Continued — Notes on this step continued on next page)

(Continued — Notes on this step continued from previous page)

Note that the DMA count registers select the number of *samples* sent or received. For a description of the relationship between data formats, samples, and the 32-bit transfers used by the codec, see the previous section, *Codec Data Formats & Sequencing*.

Also note that the codec only loads the current count registers when either: *(1)* You write to an Upper Base Count register (0x0E or 0x1E), or *(2)* A DMA counter interrupt occurs.

Finally, note that once DMA transfers are enabled the codec decrements by the number of samples sent or received in a 32-bit codec transfer. On the next data transfer after the number of samples in the current count is reached (underflow), the codec issues an interrupt (indicates this by setting the INT bit) and automatically reloads the current count registers.

6) Enable playback (set PEN bit & clear PPIO bit) and/or capture (set CEN bit & Clear CPIO bit) in Codec Indexed register 0x09.

   Note that you can abort the DMA transfer at any point by disabling DMA (clear the PEN and/or CEN bits) without entering MCE state. See note in step 9 on terminating a DMA process.

7) Start DMA by exiting Mode Change Enable state (clear the MCE bit of Codec Index Address register) with playback and/or capture control bits set (PEN and/or CEN).

8) Clear the codec interrupt that occurs after the DMA transfer is complete (count underflow) by writing to the Codec Status register (clearing INT *sticky* bit).

   Note that (*unless the TRD bit in the Codec Index Address register is set*) DMA continues whether or not you clear the INT bit. Clear the INT bit so the bit can indicate when the next batch of data is needed.

9) End the DMA process by disabling playback (clear PEN bit) and/or capture (clear CEN bit) in Codec Indexed register 0x09.

   Note that you must service the last sample in the last 32-bit codec transfer after the DMA process is terminated to complete the process. *If you disable a DMA channel (clear the PEN or CEN bits) while the AD1812 DMA request pin is active, the last complete 32-bit codec transfer (and all the samples within that transfer) must be serviced before the AD1812 deasserts the DMA request pin.*

To initiate a PIO process using the AD1812's integrated codec, use the following steps:

1) Enter Mode Change Enable state by setting the MCE bit of Codec Index Address register. (You must put the codec in MCE state to modify the contents of Codec Indexed register 0x09—*except for the PEN and CEN bits*).

2) Use steps (a) *and/or* (b) to set up a PIO channel:

   a) Select (while the playback channel is disabled, PEN=0) the playback data format, stereo/mono, byte swapping, and interface with the PFMT, PC/L, PS/M, PBSW, and PINF8 bits in Codec Indexed register 0x08.

   b) Select (while the capture channel is disabled, CPEN=0) the capture data format, stereo/mono, byte swapping, and interface with the CFMT, CC/L, CS/M, CBSW, and CINF8 bits in Codec Indexed register 0x1C.

3) Select the sample rate with the FU and FL bits in Codec Indexed registers 0x16 and 0x17 (FU must be loaded before FL).

4) Enable capture or playback PIO by setting either the playback bits (PEN=PPIO=1) or capture bits (CEN=CPIO=1) in Codec Indexed register 0x09.

   Note that the codec supports single channel PIO and single channel DMA/PIO (one of each at the same time) modes. Use the DMA procedure (precedes this PIO procedure) to set up a DMA channel.

5) Start PIO by exiting Mode Change Enable state (clear the MCE bit of Codec Index Address register) with playback and/or capture control bits set (PEN and/or CEN).

6) Poll the capture data ready (CRDY=1) or playback data ready (PRDY=1) bits in the Codec Status register. Read or write when data is ready.

   Note that the codec clears the appropriate data ready bit when playback data is *valid* (just written, do *not* overwrite) and/or capture data is *fresh* (just received, *not* yet read).

   Also note that the DMA counters can be used to count the number of samples sent or received during a PIO process and generate codec interrupts. Clear the IEN bit in Codec Indexed register 0x0A, if you want to inhibit codec interrupts (PC interrupts only, not the INT bit) during a PIO process.

7) End the PIO process by clearing the playback and/or capture control bits (PEN and/or CEN).

2.4.3 Codec I/O Mixing, Gain, & Attenuation

The AD1812's integrated codec provides control for gain, attenuation, and digital mixing through a set of registers. This section provides a overview of these audio controls. For information on each of the codec registers mentioned in this section, see Chapter 3, *AD1812 Registers*.

Figure 2.10 shows a block diagram of the AD1812 controller's codec and indicates the indices of the control registers for each stage.

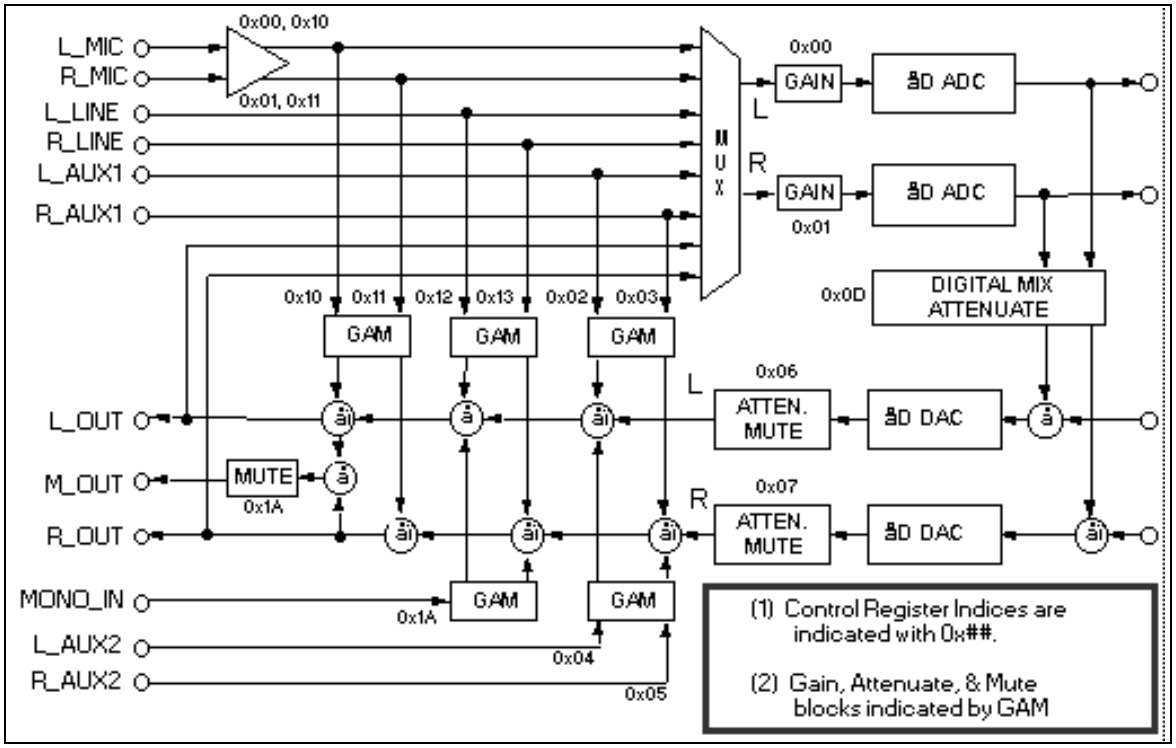


Figure 2.10 Map of AD1812 Integrated Codec (With Control Registers)

The following are descriptions of the control registers indicated in Figure 2.10. Use the figure and the following descriptions to correlate registers with mix, gain, and attenuate control functions:

- *Left Input Control (Index 0x00)*
  The LMGE bit of this register controls a 0 or 20 dB gain on the L_MIC input. The LIG bits of this register control a 0 to 22.5 dB gain on the left channel ADC input. The LSS bits of this register select the input source for the pre-ADC left channel gain stage.

- *Right Input Control (Index 0x01)*
  The RMGE bit of this register controls a 0 or 20 dB gain on the R_MIC input. The RIG bits of this register control a 0 to 22.5 dB gain on the right channel ADC input. The RSS bits of this register select the input source for the pre-ADC right channel gain stage.

- *Left AUX1 Input Control (Index 0x02)*
  The LMX1 bit of this register controls a mute on the L_AUX1 input. The LX1A bits of this register control a 12 to -34.5 dB gain-attenuation on the L_AUX1 input to the L_OUT output mixer.

- *Right AUX1 Input Control (Index 0x03)*
  The RMX1 bit of this register controls a mute on the R_AUX1 input. The RX1A bits of this register control a 12 to -34.5 dB gain-attenuation on the R_AUX1 input to the R_OUT output mixer.

- *Left AUX2 Input Control (Index 0x04)*
  The LMX2 bit of this register controls a mute on the L_AUX2 input. The LX2A bits of this register control a 12 to -34.5 dB gain-attenuation on the L_AUX2 input to the L_OUT output mixer.

- *Right AUX2 Input Control (Index 0x05)*
  The RMX2 bit of this register controls a mute on the R_AUX2 input. The RX2A bits of this register control a 12 to -34.5 dB gain-attenuation on the R_AUX2 input to the R_OUT output mixer.

- *Left Output Control (Index 0x06)*
  The LDM bit of this register controls a mute on the left channel DAC's output. The LDA bits of this register control a 0 to -94.5 dB attenuation on the left channel DAC's output passed to the L_OUT output mixer.

- *Right Output Control (Index 0x07)*
  The RDM bit of this register controls a mute on the right channel DAC's output. The RDA bits of this register control a 0 to -94.5 dB attenuation on the right channel DAC's output passed to the R_OUT output mixer.

- *Digital Mix/Attenuation (Index 0x0D)*
  The DME bit of this register controls digital mixing of left and right channel ADCs output passed to left and right channel DACs input. The DMA bits of this register control a 0 to -94.5 dB attenuation on the mixer's output.

- *Alternate Feature Enable/Left MIC Input Control (Index 0x10)*
  The OL bit of this register controls the analog output line level, scaling outputs for 2.8 Vpp or 2Vpp full scale signal. The LMC bits of this register control a 12 to -34.5 dB gain-attenuation on the L_MIC input (post LIG gain) to the L_OUT output mixer.

- *MIC Mix Enable/Right MIC Input Control (Index 0x11)*
  The RMME and LMME bits of this register control passing the R_MIC and L_MIC inputs to the R_OUT and L_OUT output mixers. The RMG bits of this register control a 12 to -34.5 dB gain-attenuation on the R_MIC input (post RIG gain) to the R_OUT output mixer.

- *Left Line Gain, Attenuate, Mute, Mix (Index 0x12)*
  The LLM bit of this register controls a mute on the L_LINE input. The LLG bits of this register control a 12 to -34.5 dB gain-attenuation on the L_LINE input to the L_OUT output mixer.

- *Right Line Gain, Attenuate, Mute, Mix (Index 0x13)*
  The RLM bit of this register controls a mute on the R_LINE input. The RLG bits of this register control a 12 to -34.5 dB gain-attenuation on the R_LINE input to the R_OUT output mixer.

- *Mono Control (Index 0x1A)*
  The MIM bit of this register controls a mute on the MONO_IN input. The MOM bit of this register controls a mute on the M_OUT output. The MIA bits of this register control a 0 to -45 dB attenuation on the M_IN input to the R_OUT and L_OUT output mixers.

2.4.4 Codec Autocalibration

Codec autocalibration of the ADCs and DACs provides greater accuracy by minimizing DC offsets. *At power on reset, the AD1812's codec always autocalibrates before you can access it and then clears the ACAL bit.* This section describes how to autocalibrate the codec, *if you determine it necessary* at some time after power on reset.

Autocalibration occurs when the codec exits Mode Change Enable state (controlled with MCE bit in Codec Index Address register) with Autocalibration Enabled (ACAL bit set in Codec Indexed register 0x09). If an autocalibration is *not* performed on exiting the Mode Change Enable state, the codec uses the ADC and DAC offset compensations retained from the most recent autocalibration.

You can monitor the completion of autocalibration by polling for a Low-High-Low transition of the Autocalibrate-In-Progress (ACI) bit in Codec Indexed register 0x0B (Test and Initialization register). This bit is set during autocalibration and cleared on completion. As an alternative to polling, you can assume autocalibration is complete 384 sample periods after it starts.

Note that data transfers enabled during autocalibration do *not* begin until the completion of autocalibration.

To perform an autocalibration of the AD1812's integrated codec, use the following steps:

1) Set (write-1-to) the Mode Change Enable (MCE) bit in the WSS Codec Status register.

2) Set (write-1-to) the Autocalibration Enable (ACAL) bit in Codex Indexed register 0x09.

3) Clear (write-0-to) the Mode Change Enable (MCE) bit

   After you have toggled the MCE bit, the Autocalibrate-In-Progress (ACI) bit transitions from LO to HI immediately. This bit remains HI for 384 sample periods. Polling the ACI bit for its transitions from Low to High and then from High to Low provides a flag for completion of autocalibration.

   Note that during autocalibration the codec automatically mutes the left and right DAC outputs, AUX1 and AUX2 inputs, and disables the digital mix.

   Also note that while the autocalibration sequence is in progress, data output from the ADCs is *meaningless* and inputs to the DACs are *ignored*.

2.4.5 Codec Sample Rate Operations

To modify the sample rate used by the AD1812's integrated codec, change the rate in the Upper and then Lower Frequency Select registers (writing to codec Indexed registers 0x16 and then 0x17). This feature of the integrated codec lets you modify the sample rate selection in 1 Hz increments *on the fly* without entering MCE mode.

Note: You must write to the Lower Frequency Select register (0x17) *last* when changing the frequency because writes to Codec Indexed register 0x17 force the codec to update the current sample rate (16-bit register).

2.4.6 Codec Powerdown Operations

Two types of registers control codec powerdown operations. The Plug & Play Powerdown register (PnP Index 0x20) controls codec and DSP powerdown. Several codec indexed registers (Powerdown Control-Index 0x1B & Total Powerdown-Index 0x1D) also control codec powerdown.

The powerdown functions that correspond between these two types of registers can be controlled from either location, but the Plug & Play registers override the codec registers. If the codec has been powered down using the Plug & Play Powerdown register's SP_PWNDWN or TOT_PWRDWN bits, the codec must be powered up by clearing those bits; the codec Total Powerdown register's TOTPWD bit cannot override the Plug & Play powerdown command. Table 2.15 lists the register bits that correspond.

Table 2.15 Powerdown Bits (Plug & Play Versus Codec)

Plug & Play Registers	*Codec Registers*	*Description (Type)*
N/A	Codec Index 0x1B, bit 5, MIXPWD	1 Powers Codec DAC & Mixer down (RW)
N/A	Codec Index 0x1B, bit 6, DACPWD	1 Powers Codec DAC down (RW)
N/A	Codec Index 0x1B, bit 7, ADCPWD	1 Powers Codec ADC down (RW)
N/A	Codec Index 0x1D, bit 0, TOTPWD	1 Powers Codec down (RW)
PnP Index 0x20, bit 1, SP_PU_RDY	N/A	1 Indicates Codec powered up (RO)
PnP Index 0x20, bit 2, SP_PD_RDY	N/A	1 Indicates Codec powered down (RO)
PnP Index 0x20, bit 7, TOT_PWRDWN	N/A	1 Powers chip down (RW)

When the AD1812 is initially powered up, the RESET pin should be asserted (high). Asserting the RESET pin keeps the AD1812 in its minimum power consumption state. All analog and digital sections are shut down. The codec's parallel interface does not function; all bi-directional signal lines are in high-impedance tri-state. The crystal on pins XTALI and XTALO must be powered during the RESET pulse.

Exit signal imposed powerdown by de-asserting the $\overline{\text{PWRDWN}}$ and RESET pins. On return from powerdown, initial power up, or reset, the AD1812's codec enters initialization, returning all registers to their initialization values. During initialization, the codec autocalibrates itself. This technique powers up all the analog amplifiers without any *clicks* and *pops*.

While initializing, the codec ignores writes and returns 0x8080 on reads. You can monitor for the end of initialization (approximately 300 ms in duration) by polling the index register for any value other than 0x8080.

The AD1812's codec has five software programmable Advanced Power Down modes. You can control these powerdown modes from the Powerdown Control and Total Powerdown registers. Table 2.16 provides a summary of power down modes, recovery periods, and power dissipation.

Table 2.16 Power Down Mode Summary

Power Down Mode	*TOTPWD Bit*	*ADCPWD Bit*	*DACPWD Bit*	*MIXPWD Bit*	*Power-Down Delay*	*Power-Up Delay*	*Power Use*
Total Power-Down	1	x	x	x	3 ms	512 ms	tbd
Standby	0	1	x	1	1/Fs	1/Fs	tbd
Mixer Power-Down	0	0	x	1	1/Fs	1/Fs	tbd
Mixer Only	0	1	1	0	1/Fs	1/Fs	tbd
ADC Power-Down	0	1	0	0	1/Fs	1/Fs	tbd
DAC Power-Down	0	0	1	0	1/Fs	1/Fs	tbd

Note: An "X" indicates that the bit does not influence the powerdown mode

2.4.7 Codec Comparison (AD1845 Vs. AD1812's Integrated Codec)

The codec integrated into the AD1812 is a modified version of the Analog Devices AD1845 SoundPort codec. Table 2.17 provides programmers and hardware designers already familiar with the AD1845 with comparison information on the two codecs.

Table 2.17 Comparison of AD1845 Vs. AD1812's Codec

Feature	*AD1845*	*AD1812's Integrated Codec*
Autocalibration	On powerup, the ACAL bit is set, but you can skip autocalibration by clearing it before exiting MCE.	On powerup, the codec always autocalibrates before you have access to the codec and the ACAL bit is cleared.
Sample Rate	Sample rate selected with register 0x08 or (when the FREN bit is set) with registers 0x16 and 0x17	Sample rate only selected with registers 0x16 and 0x17.
Host Interface	Host interface consists of four 8-bit registers, addressed as base+0, +1, +2, & +3. All capture and playback data is passed through the 8-bit interface.	Host interface consists of four 16-bit registers, addressed as base+0, +2, +4, & +6. Capture and playback data can be passed in 8 or 16-bit transfer modes (selected by CINF8 and PINF8 bits). Control information is written to or read from the lower bytes of the interface registers.
Data Format Selection	In Mode1, data format is selected with register 0x08. In Mode2, Playback format is selected with register 0x08. In Mode2, Capture format is selected with register 0x1B.	Because AD1812's codec is always in "Mode2": Playback format is selected with register 0x08. Capture format is selected with register 0x1B.
Data Formats	The AD1845 supports the following playback/capture data formats: • 16-bit, signed Little Endian • 8-bit, unsigned PCM • 8-bit, µ-Law PCM • 8-bit, A-Law PCM	The AD1812's codec supports the following playback/capture data formats: • 16-bit, signed Little Endian • 16-bit, signed Big Endian • 8-bit, unsigned PCM • 8-bit, µ-Law PCM • 8-bit, A-Law PCM • 4-bit, IMA-ADPCM

Table 2.17 Comparison of AD1845 Vs. AD1812's Codec (*continued*)

Feature	*AD1845*	*AD1812's Integrated Codec*
Data Transfer	The AD1845 uses the following techniques for data transfer: • DMA counters are programmed with the number of samples to transfer. Counters decrement by one with each transfer • Transfer 1 sample per transfer, varying the size (number of bits) of the transfer. • Store uncompressed data in fixed depth (16 position) FIFO buffer.	The AD1812 uses the following techniques for data transfer: • DMA counters are programmed with the number of samples to transfer. Counters decrement by the number of samples (varies with format) in each transfer. • Transfer 32-bits per transfer, varying the number of samples (based on data format) in each transfer. • Store compressed data in FIFO buffer. Because the number of samples in the buffer depends on the format, the number of samples in the buffer varies from 16 (16-bit, linear, stereo) to 128 (4-bit, IMA-ADPCM, mono).
Bit Differences	AD1845 INITD Bit	This bit (0x0A, [0]) is now reserved. The INITD bit's functionality is not needed because the AD1812 does not transition between Mode1 and Mode2.
	AD1845 BUF8 Bit	This bit (0x0C, [5]) is now reserved. The BUF8 bit's functionality is now in the Plug & Play powerdown register
	AD1845 MODE2 Bit	This bit (0x0C, [6]) is always set (1), the AD1812 always in Mode2.
	AD1845 FREN bit	This bit (0x1B, [3]) is always set (1).

2.5 AD1812 Programming Summary

The AD1812 SoundPort Controller contains five Plug & Play logical devices (Windows Sound System, Sound Blaster Pro, AdLib, MIDI, & Game Port). Programming the logical devices on the AD1812 consists of configuring each device then sending programming information to it. You can configure these devices in Plug & Play Mode (AD1812 PnP pin asserted) or in Non-Plug & Play Mode (AD1812 PnP pin de-asserted).

Windows Sound System codec programming is also covered in this chapter. Programming procedures for data transfer, mixing, attenuation, and gain are described. The information in this chapter includes the following:

- AD1812 Programming Overview
- AD1812 Plug & Play Resource Data
- AD1812 Non-Plug & Play Configuration Procedure
- AD1812 Windows Sound System Codec Programming

The Plug & Play Logical Devices in the AD1812 comply fully with the *Plug & Play ISA Specification Version 1.0a (May 5, 1994).*

Chapter 3 AD1812 Registers

3.1 Overview

The AD1812 has many direct registers (PC I/O addressable) and indirect registers (indexed through the direct registers). This chapter describes all the direct registers and the set of indirect registers unique to the AD1812.

Indirect (indexed) registers not unique to the AD1812 are described in other documents. For example, the AD1812 provides similar functionality and a register set that corresponds to the Sound Blaster Pro PC plug-in card. This manual lists the PC I/O ports (registers) for the Sound Blaster Pro logical device in the AD1812, but not the indexed registers in the Sound Blaster Pro. For a description of the indexed registers in a Sound Blaster Pro, see the *Developer Kit for Sound Blaster Series* programmer's reference Sound Blaster:

Register descriptions include the following:

- Register Figure: The figure shows the register name, address (direct or indirect), bit names, and reset values for each bit (0, 1, or *X* for undetermined).

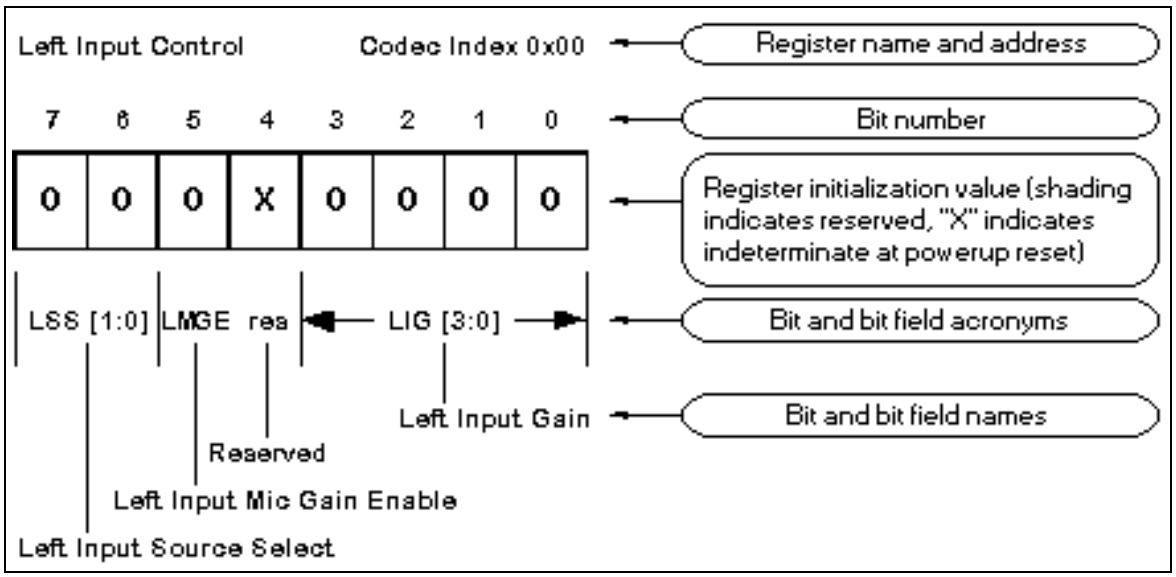


Figure 3.1 Register Diagram Example

Register Bit Table: The table lists the register's bits (in ascending order) by name and gives a usage description and bit type: (RO) Read Only, (WO) Write Only, (STKY) Sticky, and (RW) Read/Write. Table 3.1 lists the AD1812 registers that are directly available from the ISA Bus (PC I/O addressable ports).

Table 3.1 Map of AD1812 ISA Bus Registers

Register Type—Register Name	*Register PC I/O Address*
Plug & Play	*(PnP pin asserted)*
• ADDRESS	0x279
• WRITE_DATA	0xA79
• READ_DATA	Relocatable in range 0x203 - 0x3FF
Non-Plug & Play	*(PnP pin de-asserted)*
• ADDRESS	0x234
• WRITE_DATA	0x235
• READ_DATA	Relocatable in range 0x203 - 0x3FF
Windows Sound System	
• WSS Codec Index Address(r,w)	0x(WSS Base) Relocatable in range 0x0008 - 0xFFF8
• WSS Codec Indexed Data(r,w)	0x(WSS Base+2)
• WSS Codec Status (r,w)	0x(WSS Base+4)
• Codec PIO Data (r,w)	0x(WSS Base+6)
Sound BlasterPro	
• Music0: Address (w), Status (r)	0x(SB Base) Relocatable in range 0x010 - 0x3F0
• Music0: Data (w)	0x(SB Base+1)
• Music1: Address (w)	0x(SB Base+2)
• Music1: Data (w)	0x(SB Base+3)
• Mixer Address(w)	0x(SB Base+4)
• Mixer Data(w)	0x(SB Base+5)
• Reset (w)	0x(SB Base+6)
• Music0: Address (w)	0x(SB Base+8)
• Music0: Data (w)	0x(SB Base+9)
• Input Data (r)	0x(SB Base+A)
• Status (r), Output Data (w)	0x(SB Base+C)
• Status (r)	0x(SB Base+E)
AdLib	
• Music0: Address (w), Status (r)	0x(Adlib Base) Relocatable in range 0x008 - 0x3F8
• Music0: Data (w)	0x(Adlib Base+1)
• Music1: Address (w)	0x(Adlib Base+2)
• Music1: Data (w)	0x(Adlib Base+3)
MIDI MPU-401	
• MIDI Data (r/w)	0x(MIDI Base) Relocatable in range 0x008 - 0x3F8
• MIDI Status (r), Command (w)	0x(MIDI Base+1)
Game Port	
• Game Port I/O	0x(Game Base) Relocatable in range 0x001 - 0x3FF

3.2 AD1812 Plug & Play And Non-Plug & Play Registers

This section describes the Plug & Play and Non-Plug & Play registers unique to the AD1812. For descriptions of direct and indirect Plug & Play (and corresponding Non-Plug & Play) registers, see the *Plug & Play ISA Specification Version 1.0a (May 5, 1994)* from Intel and Microsoft.

3.2.1 Plug & Play And Non-Plug & Play ISA Bus Registers (Ports)

The AD1812 has two configuration modes, Plug & Play (PnP pin asserted) and Non-Plug & Play (PnP pin de-asserted). Tables 3.2 and 3.3 list the Plug & Play and Non-Plug & Play ISA Bus Registers. For programming information (i.e. using Plug & Play registers to configure the AD1812), see Chapter 2, *AD1812 Programming*

Table 3.2 Plug & Play ISA Bus Registers (PnP pin asserted)

Register	*Description*
ADDRESS	0x279
WRITE_DATA	0xA79
READ_DATA	Relocatable in range 0x203 - 0x3FF

Table 3.3 Non-Plug & Play ISA Bus Registers (PnP pin de-asserted)

Register	*Description*
ADDRESS	0x234
WRITE_DATA	0x235
READ_DATA	Relocatable in range 0x203 - 0x3FF

3.2.2 Plug & Play And Non-Plug & Play Indexed Registers

The AD1812 supports all required registers described in the *Plug & Play ISA Specification Version 1.0a (May 5, 1994)* from Intel and Microsoft. This section contains descriptions of the AD1812's vendor defined Plug & Play registers. For descriptions of the complete set of Plug & Play registers see the Intel/Microsoft specification. As with the ISA bus registers—for programming information, see Chapter 2, *AD1812 Programming*

In Plug & Play mode (PnP pin asserted) and Non-Plug & Play mode (PnP pin de-asserted), your SoundPort Controller configuration routine uses the ADDRESS, WRITE_DATA, and READ_DATA registers (ports) to configure the chip. Writes to WRITE_DATA and reads from READ_DATA access the internal Plug & Play register pointed to by the address in the ADDRESS register (port).

Table 3.4 Map of AD1812 Plug & Play—Indexed Registers

Indexed Plug & Play Register	*Index*	*Reset State*
Set RD_DATA Port	0x00	0x00
Serial Isolation	0x01	0x00
Config Control	0x02	0x00
Wake [CSN]	0x03	0x00
Resource Data	0x04	0x00
Status	0x05	0x00
Card Select Number	0x06	0x00
Logical Device Number (LDN)	0x07	0x00
Powerdown	0x20	0x00
Activate (LDN Indexed)	0x30	0x00
I/O Range Check (LDN Indexed)	0x31	0x00
I/O port base address descriptor 0 (LDN Indexed)	0x60 - 061	0x00
Interrupt level select 0 (LDN Indexed)	0x70	0x00
Interrupt type select 0 (LDN Indexed)	0x71	0x02
DMA channel select 0 (LDN Indexed)	0x74	0x04
DMA channel select 1 (LDN Indexed)	0x75	0x04

For Plug & Play indexed registers with addresses of 0x30 and up, the contents of the Logical Device Number register applies a second tier of indexing. Figure 3.2 shows how the indexing scheme applies to Plug & Play register types. Table 3.4 lists the internal Plug & Play registers, their indices, and reset values.

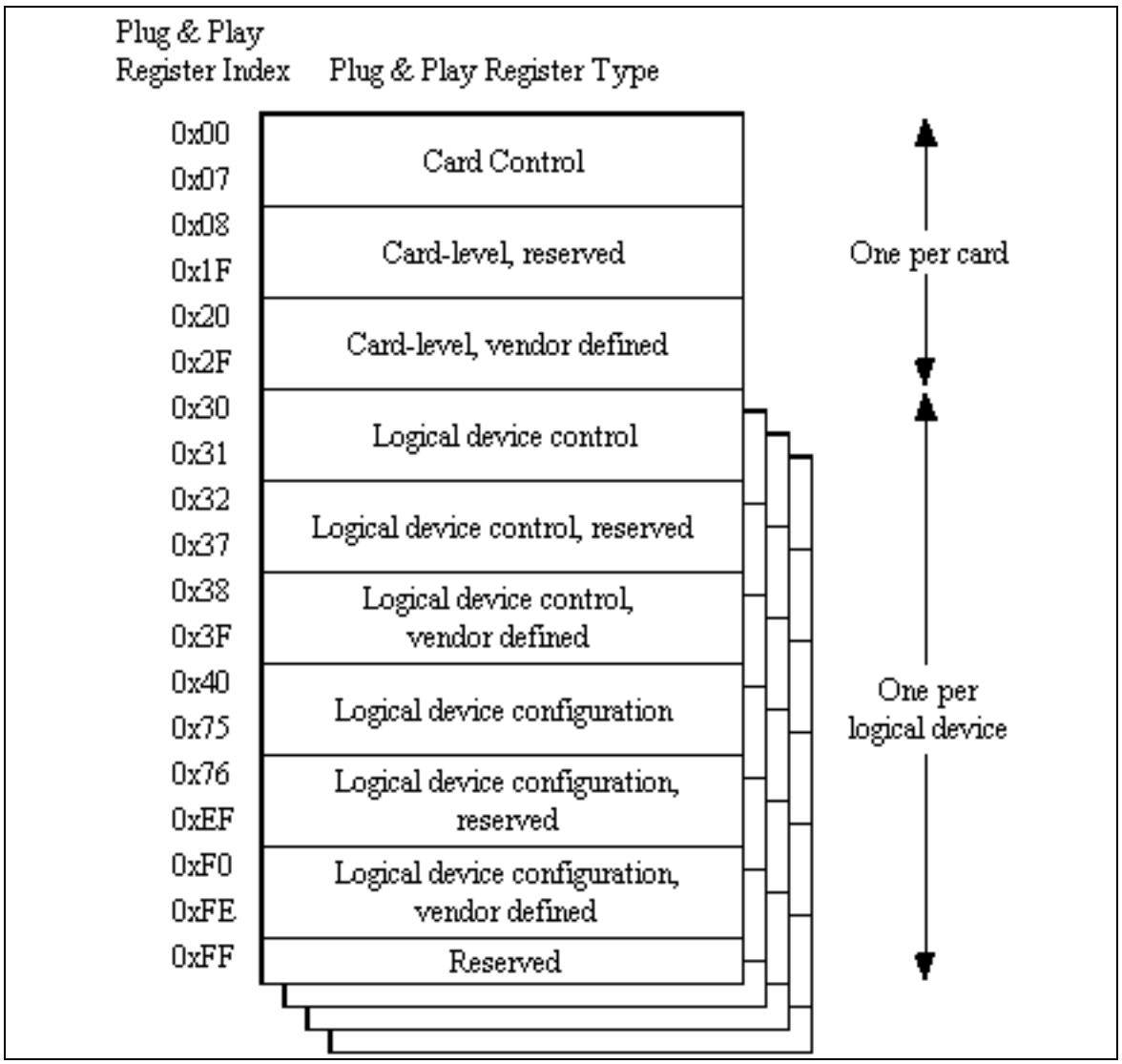
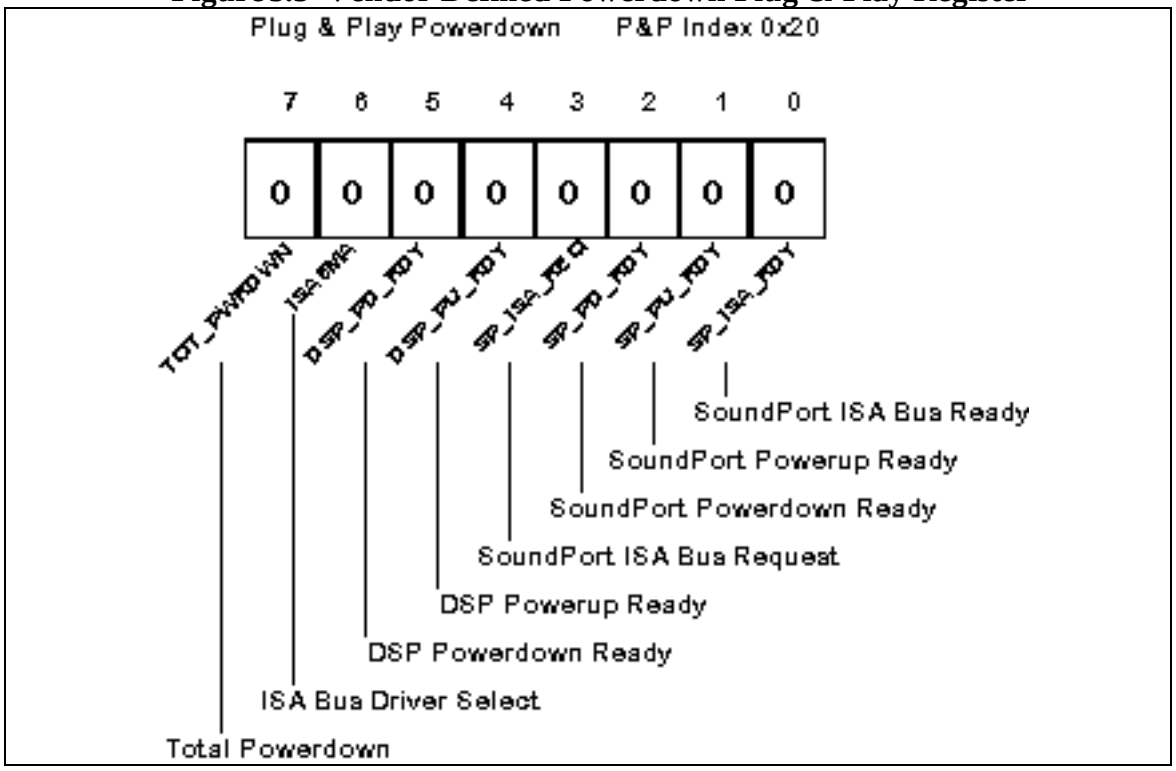


Figure 3.2 AD1812 Plug & Play Register Indexing

Figure 3.3 Vendor Defined Powerdown Plug & Play Register



Bits	*Description (Plug & Play Indexed (0x20) Powerdown Register*
[0] SP_ISA_RDY (RW)	SoundPort ISA Bus Ready 1 Indicates SoundPort is in Windows Sound System mode. 0 Indicates SoundPort is in Sound Blaster mode. Note: Compare the contents of SP_ISA_RDY and SP_ISA_REQ (must be identical, either 0 or 1) before using the Sound Port.
[1] SP_PU_RDY (RO)	SoundPort Powerup Ready 1 Indicates the 1845 is powered up.
[2] SP_PD_RDY (RO)	SoundPort Powerdown Ready 1 Indicates the 1845 is powered down.

Bits	*Description (Plug & Play Indexed (0x20) Powerdown Register (continued)*
[3] SP_ISA_REQ (RW)	SoundPort ISA Bus Request 1 Requests the SoundPort in Windows Sound System mode. 0 Requests the SoundPort in Sound Blaster mode. The codec changes modes within approximately 1 ms of writing to SP_ISA_REQ and in Windows Sound System mode is delivered in MCE mode with outputs muted. Note: Compare the contents of SP_ISA_RDY and SP_ISA_REQ (must be identical, either 0 or 1) before writing to SP_ISA_RDY.
[4] DSP_PU_RDY (RO)	DSP Powerup Ready 1 when the DSP is powered up.
[5] DSP_PD_RDY (RO)	DSP Powerdown Ready 1 when the DSP is powered down.
[6] ISA8MA (RW)	ISA 8 mA Driver Enable Writing a 1 selects 8 mA line drivers for ISA bus signals. Writing a 0 selects 24 mA line drivers.
[7] TOT_PWRDWN (RW)	Total Powerdown Writing a 1 powers down the chip. Writing a 0 powers up again.

3.3 AD1812 Windows Sound System Registers

This section describes the Windows Sound System's four directly accessible ISA bus registers and 32 indirectly accessible indexed registers. These registers let you transfer data to and from the codec and control mixing and gain.

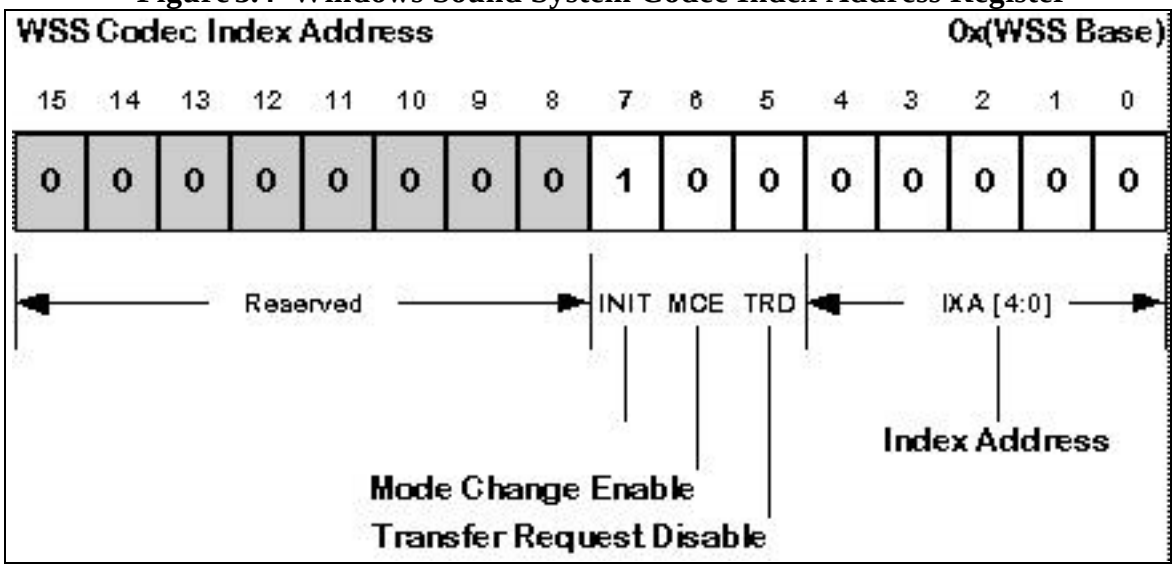
3.3.1 Windows Sound System ISA Bus Registers (Ports)

The AD1812's Windows Sound System contains four direct accessible ISA bus registers (ports). To access these registers, you address them using their ISA bus address. Table 3.5 lists the Windows Sound System direct registers and their bit acronyms.

Table 3.5 Map of Windows Sound System Register Bits

Direct Address	*Bit 15*	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
WSS Base	res	res	res	res	res	res	res	res	INIT	MCE	TRD	IXA4	IXA3	IXA2	IXA1	IXA0
WSS Base+2	res	res	res	res	res	res	res	res	IXD7	IXD6	IXD5	IXD4	IXD3	IXD2	IXD1	IXD0
WSS Base+4	res	res	res	res	res	res	res	res	CU/L	CL/R	CRDY	SOUR	PU/L	PL/R	PRDY	INT
WSS Base+6 (read)	CD15	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
WSS Base+6 (write)	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Figure 3.4 Windows Sound System Codec Index Address Register



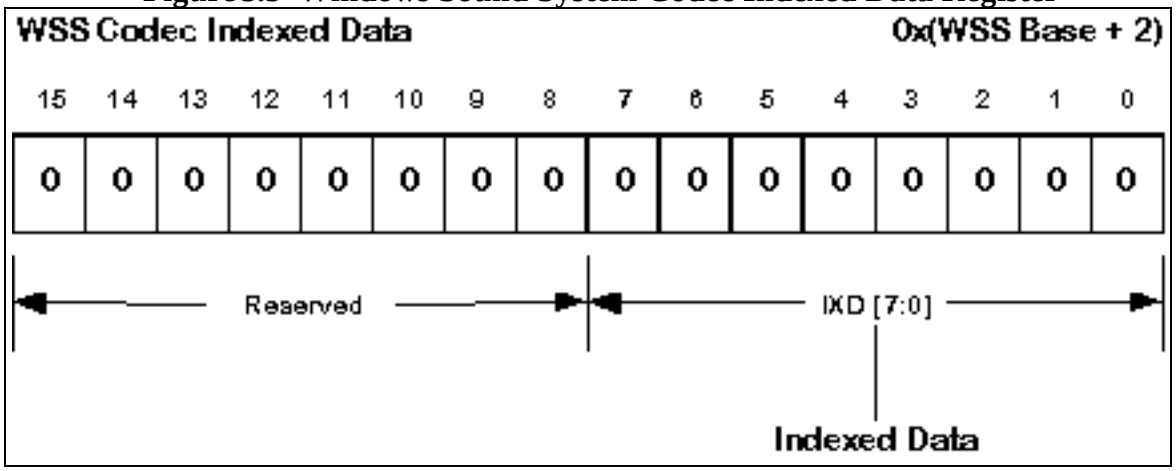
After reset this register contains 0x40 (8-bit mode or 16-bit mode). During codec initialization, this register is read only and contains 0x80 (8-bit mode or 16-bit mode).

Bits	*Description (WSS Codec Index Address—0x(WSS Base))*
[4:0] IXA [4:0] (RW)	Index Address. These bits hold the index address of the Codec register accessible through the Codec Indexed Data Register.
[5] TRD (RW)	Transfer Request Disable 0 Enables codec transfers during a codec interrupt. 1 Disables codec transfers during a codec interrupt. This bit disables (1) or enables (0) codec DMA transfers during a codec interrupt (interrupt indicated by the WSS Codec Status register's INT bit being set (1)). (This assumes codec DMA transfers were enabled—the WSS Codec Indexed (0x09) Interface Configuration register's PEN or CEN bits are set.) Playback or capture DMA requests pending when the TRD bit is set (1) are allowed to complete. After completing a pending request, the codec processes the data in the FIFO at the sample rate.

Bits	*Description (WSS Codec Index Address—0x0 (WSS Base))* *(continued)*
(continued) [5] TRD (RW)	Note: *If the codec processes all the data in the FIFO* (runs out of data), the codec has one of the following responses—*either*: • *Generates* mid-scale inputs for the DAC. The codec generates these inputs if the WSS Coded Indexed (0x10) Alternate Feature Enable /Left Mic Mix Gain register's DACZ bit is set. *or* • *Repeats* the previous valid sample as input for the DAC. In this case, the codec's ADC output buffer retains the last valid output. Resume processing of playback and/or capture requests by clearing (set to 0) either the TRD bit or the WSS Codec Status register's INT bit. Note: The codec does not report over-run or under-run errors while transfers are disabled during a codec interrupt.
[6] MCE (RW)	Mode Change Enable. 0 Disables changes to the codec's current functional mode. 1 Enables changes to the codec's current functional mode. This bit enables (1) or disables (0) changes to the codec's current functional mode. To change the codec's current functional mode (selected with WSS Codec Indexed (0x08, 0x09, 0x1C, & 0x1D) registers), you must first set (1) this bit, make the mode changes in the indexed registers, and then clear (0) the bit. (For noted exceptions, see the indexed registers' descriptions.) The codec mutes DAC outputs while the MCE bit is set (1). After the bit is cleared (0), the codec sets DAC outputs to the state specified by the WSS Codec Indexed (0x06) Left DAC Control register's LDA bits and WSS Codec Indexed (0x07) Right DAC Control register's RDA bits. After exiting the mode change state, the DACs' analog outputs mute and the ADCs output mid-scale values for 128 sample cycles, allowing the reference and filters time to settle. During the 128 cycle delay, the WSS Codec Indexed (0x0B) Test and Initialization register's ACI bit is set (1), providing a flag for system software to poll rather than count the cycles.

Bits	*Description (WSS Codec Index Address—0x(WSS Base)) (continued)*
(continued) [6] MCE (RW)	Note: If the WSS Codec Indexed (0x09) Interface Configuration register's ACALbit is set (1) when exiting mode change state, the codec goes through an autocalibration. (For information on the autocalibration sequence, see Chapter 2, *AD1812 Programming*.
[7] INIT (RO)	Codec Initialization. 0 Codec can respondto parallel bus cycles. 1 Codec does not responseto parallel bus cycles. This bit is set(1) by the codecwhen the codec's state dis-allows responseto parallel bus cycles.
[15:8] Reserved	Reserved for future expansion. Always write a zero to these bits.

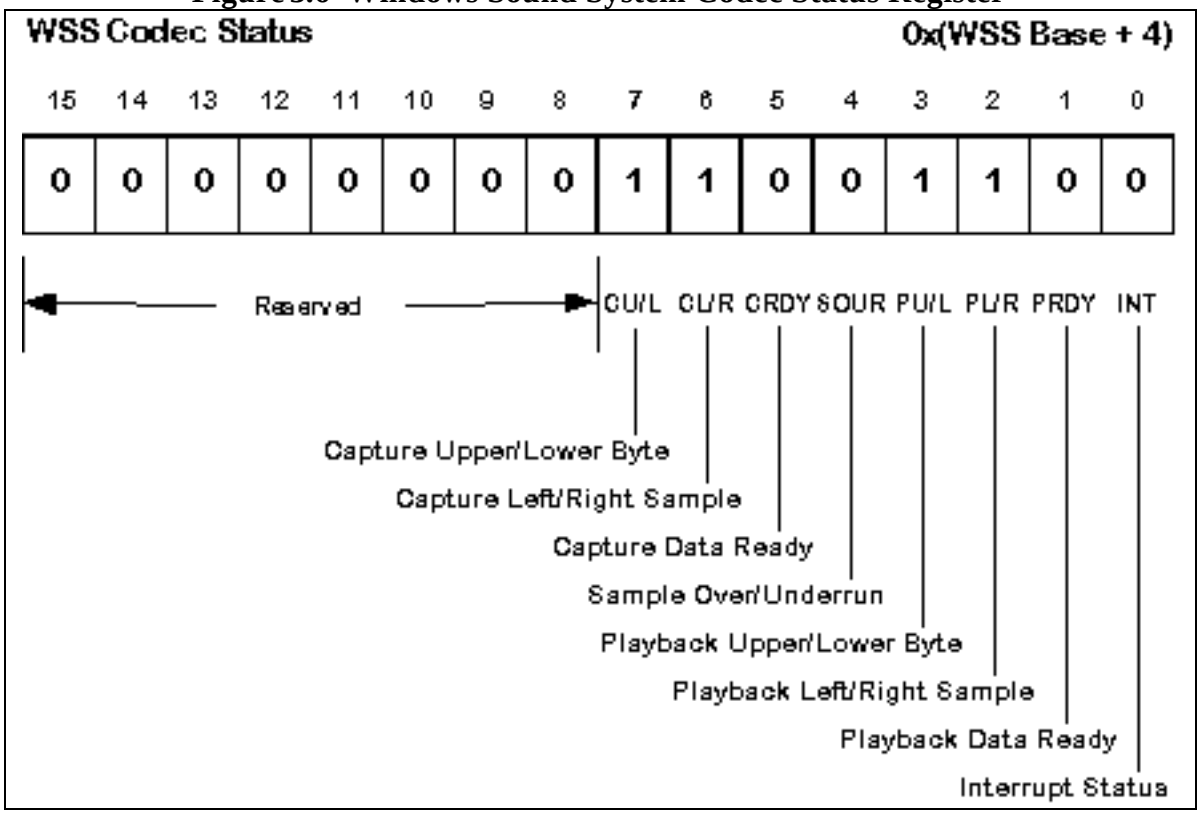
Figure 3.5 Windows Sound System Codec Indexed Data Register



During codec initialization, this register is read only and contains 0x80 (8-bit mode or 16-bit mode).

Bits	*Description (WSS Codec Indexed Data—0x(WSS Base+2))*
[7:0] IXD [7:0] (RW)	Indexed Data Register. These bits hold the contents of the indexed codec register pointed to by the WSS Codec Index Address register.
[15:8] Reserved	Reserved for future expansion. Always write a zero to these bits.

Figure 3.6 Windows Sound System Codec Status Register



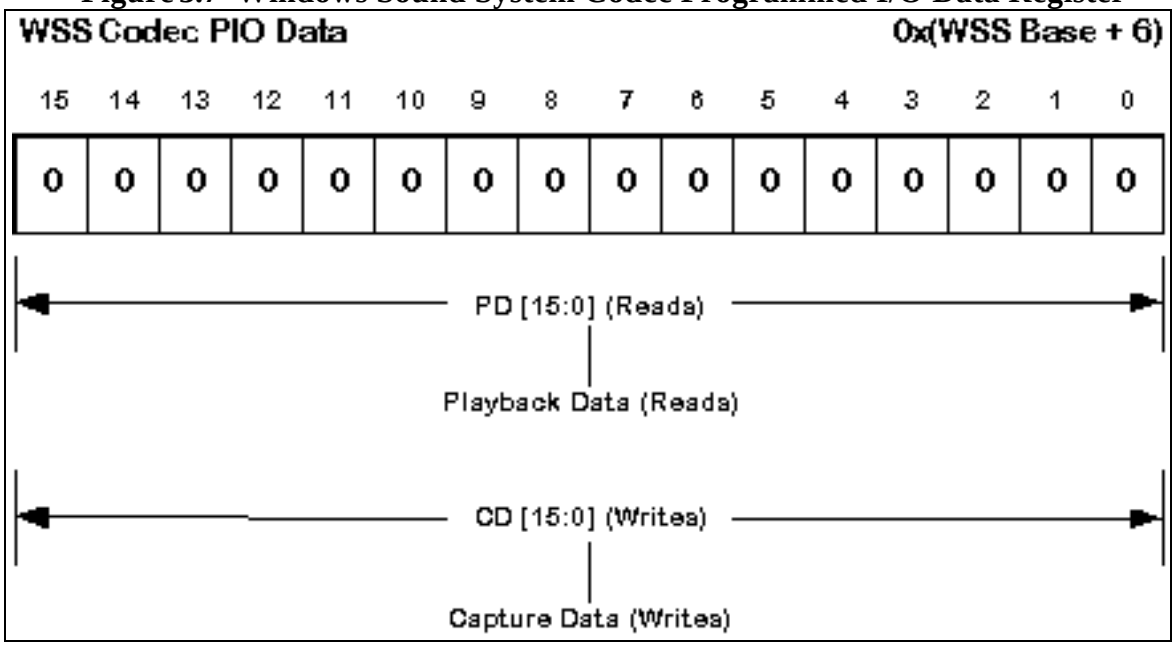
During codec initialization, this register is read only and contains 0x80 (8-bit mode) or 0x88 (16-bit mode). The register contains 0xCC after reset. Note that the PRDY, CRDY, and INT bits of this register *can* change asynchronously to ISA bus accesses. Reads of these bits during bit transition will *not* always return the post transition bit value. For timing information used when reading this register, see the *AD1812 Data Sheet*.

Bits	*Description (WSS Codec Status—0x(WSS Base+4))*
[0] INT (STKY)	Codec Interrupt. 0 No codec interrupt. 1 Active codec interrupt. This bit indicates that a codec interrupt is active (1) or inactive (0). The bit is *sticky*, cleared (0) whenever *any* value is written to this register. Note: *Underflow* conditions generate codec interrupts. An underflow of either the DMA current count or timer registers generates a codec interrupt.

Bits	*Description (WSS Codec Index Address—0xWSS Base+4)) (continued)*
[1] PRDY (RO)	Playback Data Ready 0 DAC data for playback valid. 1 DAC data for playback invalid. This bit indicates that the DAC data for playback is valid (0) or invalid (1). When the PRDY bit is set (1), the Windows Sound System Programmed I/O Playback Data register (0xWSS Base + 6) is ready for more data. When the PRDY bit is cleared (0), the data in the playback register is valid and you should not overwrite the data.
[2] PL/R (RO)	Playback Left/Right Sample 0 Playback data needed for right channel 1 Playback data needed for left channel or mono. This bit indicates PIO playback data is needed for the left channel DAC (1), right channel DAC (0), or mono (1).
[3] PU/L (RO)	Playback Upper/Lower Byte 0 Word 0 (if PINF8=0) or lower 8 bits (if PINF8=1) of playback data needed. 1 Word 1 (if PINF8 =0) or upper 8 bits (if PINF8=1) of playback data needed. If using the 16 bit interface, (PINF8=1) with any format (16, 8 or 4-bit), Word 0 or Word 1 is transferred according to the order shown in Tables 2.11 and 2.12. If using the 8-bit interface (PINF8=1) with any 16 bit format, the codec transfers bytes according to the order shown in Table 2.14. Note that if using the 8-bit interface (PINF8=1) with 8 or 4-bit formats, data is only needed when PU/L=1 (always upper).

Bits	*Description (WSS Codec Index Address—0x(WSS Base+4)) (continued)*
[4] SOUR (RO)	Sample Over/Underrun. 0 Most recent sample was serviced in time. 1 Most recent sample was not serviced in time. On a sample-by-sample basis, this bit indicates that the most recent sample was (0) or was not (1) serviced in time. If the sample was not serviced in time, a capture overrun or playback underrun condition occurs. The codec indicates these conditions with bits in the WSS Codec Indexed (0x0B) Test and Initialization register; setting (1) the COR bit on capture overruns and setting (1) the PUR bit on playback underruns.
[5] CRDY (RO)	Capture Data Ready. 0 ADC data is fresh. 1 ADC data is stale. Use this bit when using direct programmed I/O transfers to indicate whether the ADC data is fresh (0) or stale (1). If the data is fresh, read it. Do *not* re-read stale data.
[6] CL/R (RO)	Capture Left/Right Sample. 0 PIO capture data waiting for right channel ADC. 1 PIO capture data waiting for left channel ADC or mono. This bit indicates PIO capture data is waiting for the right channel ADC (0), left channel ADC (1), or mono (1).
[7] CU/L (RO)	Capture Upper/Lower Byte. 0 Word 0 (if PINF8=0) or lower 8 bits (if PINF8=1) of capture data is ready. 1 Word 1 (if PINF8 =0) or upper 8 bits (if PINF8=1) of capture data is ready. If using the 16 bit interface, (PINF8=1) with any format (16, 8 or 4-bit), Word 0 or Word 1 is transferred according to the order shown in Tables 2.11 and 2.12. If using the 8-bit interface (PINF8=1) with any 16 bit format, the codec transfers bytes according to the order shown in Table 2.14. Note that if using the 8-bit interface (PINF8=1) with 8 or 4-bit formats, data is only ready when CU/L=1 (always upper).
[15:8]	Reserved for future expansion. Always write a zero to these bits.

Figure 3.7 Windows Sound System Codec Programmed I/O Data Register



During codec initialization, this register is read only and contains 0x80 (8-bit mode) or 0x88 (16-bit mode).The PIO Data registers are two registers mapped to the same address. Writes send data to the PIO Playback Data register (PD[15:0]). Reads receive data from the PIO Capture Data Register (CD[15:0]).

Bits	*Description (WSS Codec PIO Data—0x(WSS Base+6))*
[15:0] CD [15:0] (R)	*Capture Data* (R). These bits hold the capture data read during Programmed I/O transfers. When you read from this register, the codec increments the state machine to read from the next byte or word in the sample. Note that the contents of the WSS Codec Status register indicate the identity (upper/lower byte, left/right sample) of the next byte to be read. After all bytes in the sample are read, the codec points the state machine and WSS Codec Status register to the last byte or word in the sample, and until a new sample is received the register returns the last byte in the sample on reads. When the codec receives this new sample, it points the state machine to the first byte/word in the sample. The ready bit indicates a new sample is available.
PD [15:0] (W)	*Playback Data* (W). These bits hold the playback data written during Programmed I/O transfers. When you write to this register, the codec increments the playback byte tracking state machine for the write of the next byte in the sample. After all bytes in the sample are written, the codec ignores subsequent writes to this register until the sample is sent to the DACs; at that point the codec resets the state machine and sets the ready bit in the status register.

3.3.2 Windows Sound System Indexed Registers

The AD1812's Windows Sound System contains 32 indexed registers. To access these registers, you address them through an index address register (WSS base address) and write/read data through an index data register (WSS Base address + 2). Table 3.6 lists the internal codec registers, their indices, and reset values (an "x" digit in the reset value indicates the bit is indeterminate (not set) on reset). Table 3.7 shows a map of the Index register bits. Note that you can always access the WSS Codec Status and WSS Codec PIO Data registers because these registers are directly accessible without indexing.

Table 3.6 Map of AD1812 Windows Sound System Codec—Indexed Registers

Windows Sound System Codec Indexed Register	*Index*	*Reset State*
Left Input Control	0x00	0x80
Right Input Control	0x01	0x80
Left Aux #1 Input Control	0x02	0x9F
Right Aux #1 Input Control	0x03	0x9F
Left Aux #2 Input Control	0x04	0x9F
Right Aux #2 Input Control	0x05	0x9F
Left Output Control	0x06	0xBF
Right Output Control	0x07	0xBF
Playback Data Format	0x08	0x08
Interface Configuration	0x09	0x00
Pin Control	0x0A	0x05
Test and Initialization	0x0B	0x20
Miscellaneous Information	0x0C	0xCA
Digital Mix/Attenuation	0x0D	0x00
Playback Upper Base Count	0x0E	0x00
Playback Lower Base Count	0x0F	0x00
Alternate Feature Enable/Left MIC Input Control	0x10	0x80
MIC Mix Enable/Right MIC Input Control	0x11	0x00
Left Line Gain, Attenuate, Mute, Mix	0x12	0x9F
Right Line Gain, Attenuate, Mute, Mix	0x13	0x9F
Lower Timer	0x14	0x00
Upper Timer	0x15	0x00
Upper Frequency Select	0x16	0x2A
Lower Frequency Select	0x17	0xF8
Capture Playback Timer	0x18	0x30
Revision ID	0x19	0x80
Mono Control	0x1A	0xC0
Power-Down Control	0x1B	0x08
Capture Data Format	0x1C	0x50
Total Power-Down	0x1D	0x00
Capture Upper Base Count	0x1E	0x00

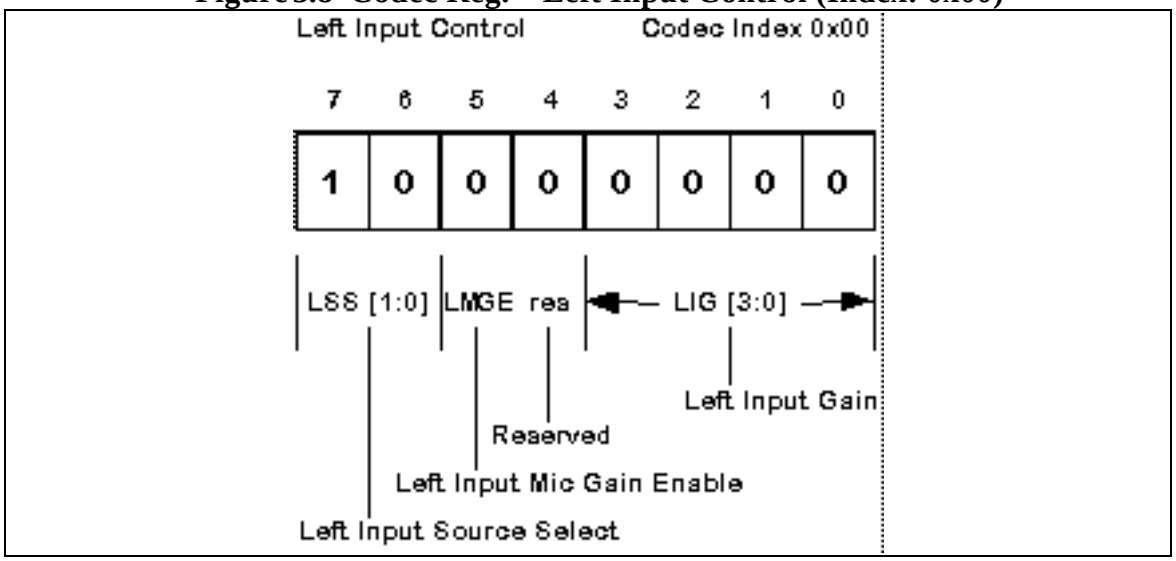
Capture Lower Base Count	0x1F	0x00

Table 3.7 Map of Window Sound System Indexed Register Bits

Indirect Address	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0x00	LSS1	LSS0	LMGE	res	LIG3	LIG2	LIG1	LIG0
0x01	RSS1	RSS0	RMGE	res	RIG3	RIG2	RIG1	RIG0
0x02	LMX1	res	res	LX1A4	LX1A3	LX1A2	LX1A1	LX1A0
0x03	RMX1	res	res	RX1A4	RX1A3	RX1A2	RX1A1	RX1A0
0x04	LMX2	res	res	LX2A4	LX2A3	LX2A2	LX2A1	LX2A0
0x05	RMX2	res	res	RX2A4	RX2A3	RX2A2	RX2A1	RX2A0
0x06	LDM	res	LDA5	LDA4	LDA3	LDA2	LDA1	LDA0
0x07	RDM	res	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0
0x08	PFMT1	PFMT0	PC/L	PS/M	PBSW	PINF8	res	res
0x09	CPIO	PPIO	res	res	ACAL	SDC	CEN	PEN
0x0A	XCTL1	XCTL0	res	res	res	res	IEN	res
0x0B	COR	PUR	ACI	DRS	ORR1	ORR0	ORL1	ORL0
0x0C	MID	res	res	res	ID3	ID2	ID1	ID0
0x0D	DMA5	DMA4	DMA3	DMA2	DMA1	DMA0	res	DME
0x0E	UB7	UB6	UB5	UB4	UB3	UB2	UB1	UB0
0x0F	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0
0x10	OL	TE	LMG4	LMG3	LMG2	LMG1	LMG0	DACZ
0x11	LMME	RMME	RMG4	RMG3	RMG2	RMG1	RMG0	res
0x12	LLM	res	res	LLG4	LLG3	LLG2	LLG1	LLG0
0x13	RLM	res	res	RLG4	RLG3	RLG2	RLLG1	RLG0
0x14	TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0
0x15	TU7	TU6	TU5	TU4	TU3	TU2	TU1	TU0
0x16	FU7	FU6	FU5	FU4	FU3	FU2	FU1	FU0
0x17	FL7	FL6	FL5	FL4	FL3	FL2	FL1	FL0
0x18	res	TI	CI	PI	CU	CO	PO	PU
0x19	V2	V1	V0	res	res	CID2	CID1	CID0
0x1A	MIM	MOM	res	res	MIA3	MIA2	MIA1	MIA0
0x1B	ADCPWD	DACPWD	MIXPWD	res	res	res	res	res

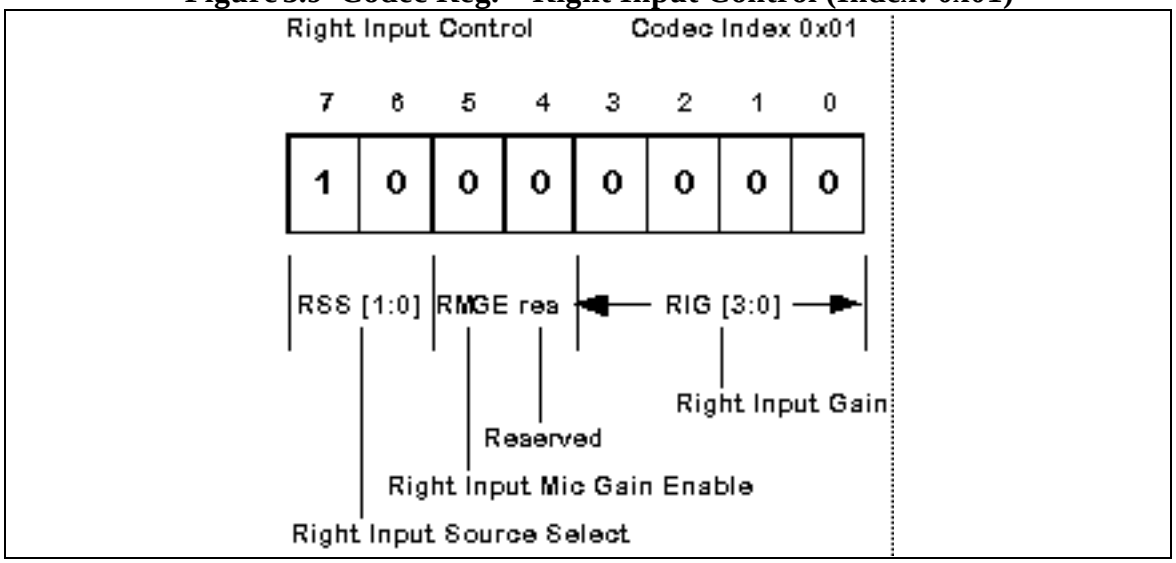
0x1C	CFMT1	CFMT0	CC/L	CS/M	CBSW	CINF8	res	res
0x1D	res	res	res	res	res	res	res	TOTPWD
0x1E	CUB7	CUB6	CUB5	CUB4	CUB3	CUB2	CUB1	CUB0
0x1F	CLB7	CLB6	CLB5	CLB4	CLB3	CLB2	CLB1	CLB0

Figure 3.8 Codec Reg.—Left Input Control (Index: 0x00)



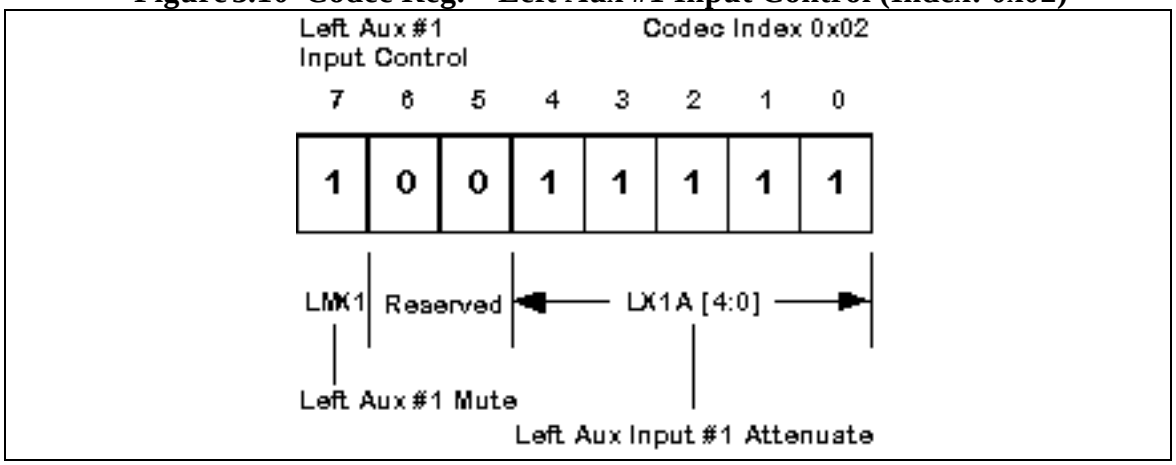
Bits	*Description (Left Input Control—WSS Codec Index 0x00)*
[3:0] LIG [3:0] (RW)	Left Input Gain select. These bits hold the gain select for the left input. Using these bits, you can select gains from 0 dB (LIG=0x0) to +22.5 dB (LIG=0xF) in +1.5 dB increments. The following equation lets you determine the value to load into LIG [3:0]: (Gain dB) / (1.5 dB) = LIG For 18 dB gain, example: 18 dB / 1.5 dB = 12 = 0b1100 = LIG
[4] res	Reserved for future expansion. Always write a zero to this bit.
[5] LMGE (RW)	Left Input Microphone Gain Enable 0 Disables a 20 dB gain of the left mic input signal. 1 Enables a 20 dB gain of the left mic input signal. This bit enables (1) or disables (0) a 20 dB gain of the left mic input signal.
[7:6] LSS [1:0] (RW)	Left Input Source Select These bits hold the selection of input source for the left gain stage (preceding the left ADC) according to the following table: LSS1 LSS0 Left Input Source 0 0 Left Line Source Selected 0 1 Left Aux. #1 Source Selected 1 0 Left Microphone Source Selected 1 1 Left Line Post-Mixed DAC Output Source Selected

Figure 3.9 Codec Reg.—Right Input Control (Index: 0x01)



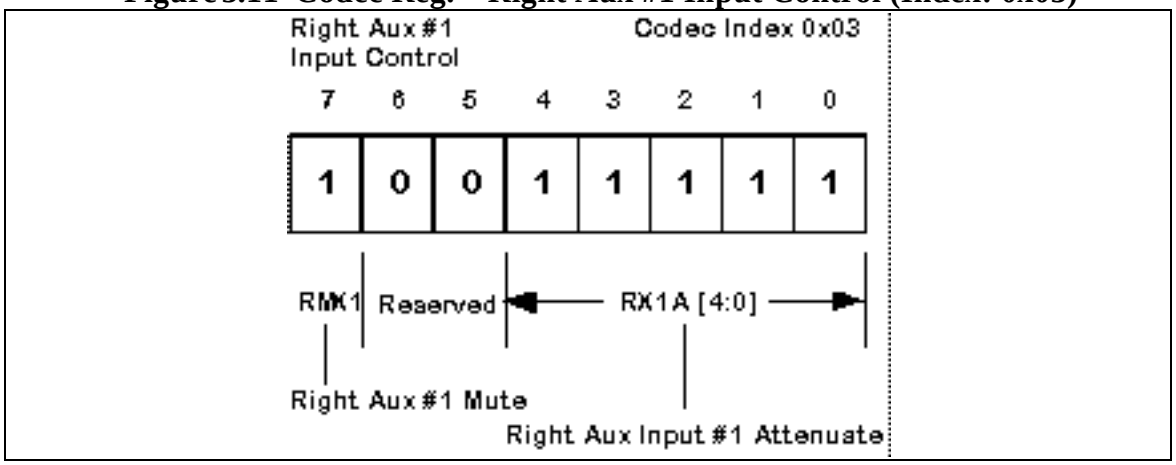
Bits	*Description (Right Input Control—WSS Codec Index 0x01)*
[3:0] RIG [3:0] (RW)	Right Input Gain select. These bits hold the gain select for the right input. Using these bits, you can select gains from 0 dB (RIG=0x0) to +22.5 dB (RIG=0xF) in +1.5 dB increments. The following equation lets you determine the value to load into RIG [3:0]: (Gain dB) / (1.5 dB) = RIG For 18 dB gain, example: 18 dB / 1.5 dB = 12 = 0b1100 = RIG
[4] res	Reserved for future expansion. Always write a zero to this bit.
[5] RMGE (RW)	Right Input Mic Gain Enable 0 Disables a+20 dB gain of the right mic input signal. 1 Enables a+20 dB gain of the right mic input signal. This bit enables (1) or disables (0) a+20 dB gain of the right mic input signal.
[7:5] RSS [1:0] (RW)	Right Input Source Select. These bits hold the selection of input source for the right channel gain stage (preceding the right ADC) according to the following table: RSS1 RSS0 Right Input Source 0 0 Right Line Source Selected 0 1 Right Aux #1 Source Selected 1 0 Right Microphone Source Selected 1 1 Right Post-Mixed DAC Output Source Selected

Figure 3.10 Codec Reg.—Left Aux #1 Input Control (Index: 0x02)



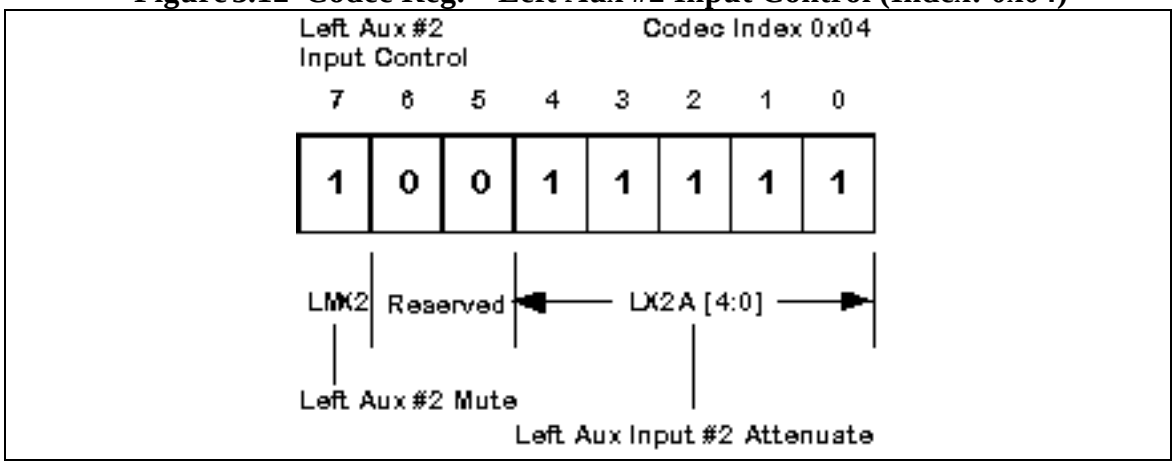
Bits	*Description* *(Left Aux Input #1 Attenuate Select—WSS Codec Index 0x02)*
[4:0] LX1A [4:0] (RW)	Left Auxiliary Input #1 Attenuate Select. These bits hold the gain/attenuate select for the left #1 auxiliary input. Using these bits, you can select gain/attenuates from 12 dB (LX1A=0x00) to -34.5 dB (LX1A=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into LX1A [4:0]: ((Gain *or* -Atten. dB) - 12 dB) / (-1.5 dB) = LX1A For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = LX1A Note: For 0 dB gain, set LX1A [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] LMX1 (RW)	Left Auxiliary #1 Mute 0 Un-mutes the left #1 auxiliary input source. 1 Mutes the left #1 auxiliary input source. This bit mutes (1) or un-mutes (0) the left #1 auxiliary input source.

Figure 3.11 Codec Reg.—Right Aux #1 Input Control (Index: 0x03)



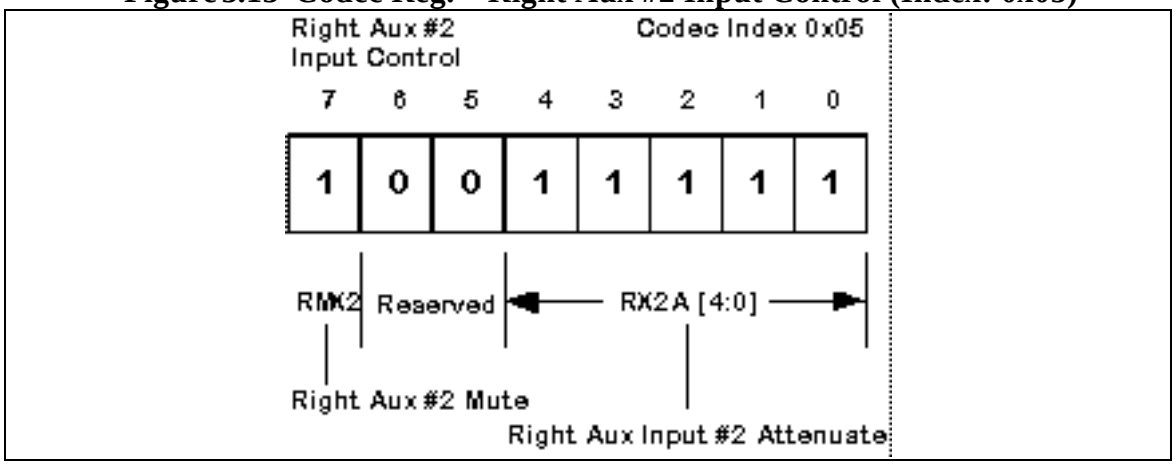
Bits	*Description* *(Right Aux Input #1 Attenuate Select—WSS Codec Index 0x03)*
[4:0] RX1A [4:0] (RW)	Right Auxiliary Input #1 Attenuate Select. These bits hold the gain/attenuate select for the right #1 auxiliary input. Using these bits, you can select gain/attenuates from 12 dB (RX1A=0x00) to -34.5 dB (RX1A=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into RX1A [4:0]: ((Gain *or* -Atten. dB) - 12 dB) / (-1.5 dB) = RX1A For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = RX1A Note: For 0 dB gain, set RX1A [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] RMX1 (RW)	Right Auxiliary #1 Mute 0 Un-mutes the right #1 auxiliary input source. 1 Mutes the right #1 auxiliary input source This bit mutes (1) or un-mutes (0) the right #1 auxiliary input source.

Figure 3.12 Codec Reg.—Left Aux #2 Input Control (Index: 0x04)



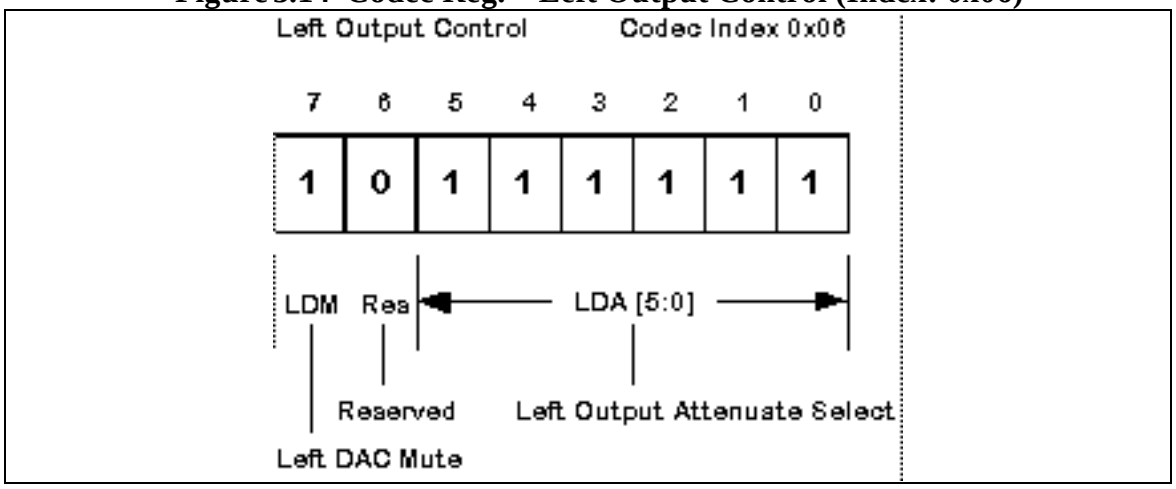
Bits	*Description* *(Left Aux Input #2 Attenuate Select—WSS Codec Index 0x04)*
[4:0] LX2A [4:0]	Left Auxiliary Input #2 Attenuate Select. These bits hold the gain/attenuate select for the left #2 auxiliary input. Using these bits, you can select gain/attenuates from 12 dB (LX2A=0x00) to -34.5 dB (LX2A=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into LX2A [4:0]: ((Gain *or* -Atten. dB) - 12 dB) / (-1.5 dB) = LX2A For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = LX2A Note: For 0 dB gain, set LX2A [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] LMX2	Left Auxiliary #2 Mute 0 Un-mutes the left auxiliary #2 input source. 1 Mutes the left auxiliary #2 input source This bit mutes (1) or un-mutes (0) the left auxiliary #2 input source. This bit powers up set.

Figure 3.13 Codec Reg.—Right Aux #2 Input Control (Index: 0x05)



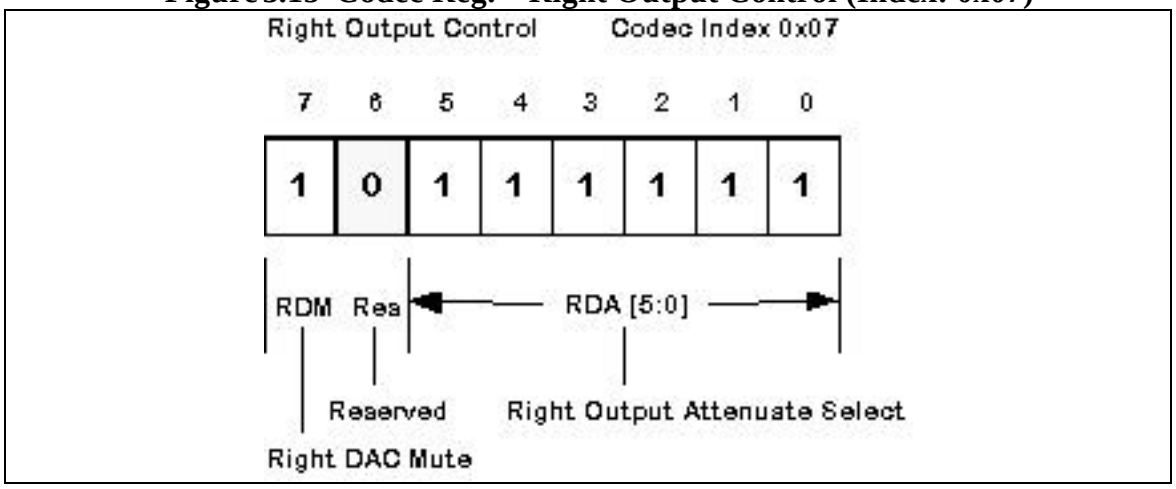
Bits	*Description* *(Right Aux Input #2 Attenuate Select—WSS Codec Index 0x05)*
[4:0] RX2A [4:0] (RW)	Right Auxiliary Input #2 Attenuate Select. These bits hold the gain/attenuate select for the right #2 auxiliary input. Using these bits, you can select gain/attenuates from 12 dB (RX2A=0x00) to -34.5 dB (RX2A=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into RX2A [4:0]: ((Gain *or* -Atten. dB) - 12 dB) / (-1.5 dB) = RX2A For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = RX2A Note: For 0 dB gain, set RX2A [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] RMX2 (RW)	Right Auxiliary #2 Mute 0 Un-mutes the right #2 auxiliary input source. 1 Mutes the right #2 auxiliary input source. This bit mutes (1) or un-mutes (0) the right #2 auxiliary input source.

Figure 3.14 Codec Reg.—Left Output Control (Index: 0x06)



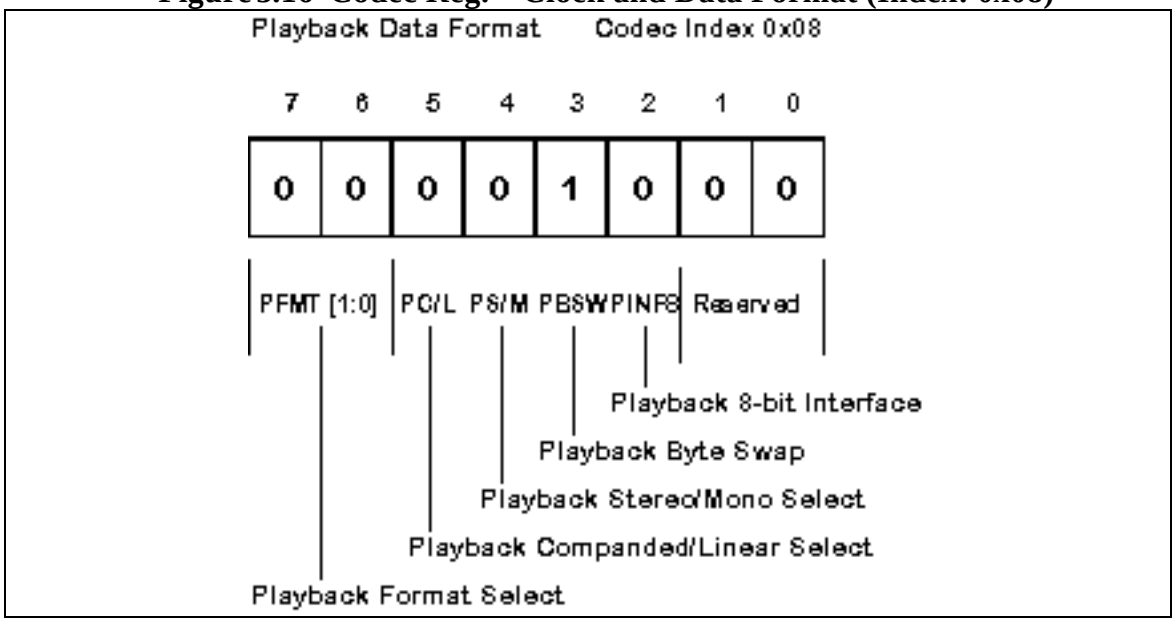
Bits	*Description* *(Left Output Control—WSS Codec Index 0x06)*
[5:0] LDA [5:0] (RW)	Left Output Attenuate select. These bits hold the attenuate select for the left output. Using these bits, you can select attenuation from 0 dB (LDA=0x00) to -94.5 dB (LDA=0x3F) in 1.5 dB increments. The following equation lets you determine the value to load into LDA [5:0]: (Attenuation dB) / (-1.5 dB) = LDA For -12 dB att., example: -12 dB / -1.5 dB = 8 = 0b01000 = LDA
[6] res	Reserved for future expansion. Always write a zero to this bit.
[7] LDM (RW)	Left DAC Mute 0 Un-mutes the left DAC output. 1 Mutes the left DAC output. This bit mutes (1) or un-mutes (0) the left DAC output.

Figure 3.15 Codec Reg.—Right Output Control (Index: 0x07)



Bits	*Description* *(Right Output Control—WSS Codec Index 0x07)*
[5:0] RDA [5:0] (RW)	Right Output Attenuate select. These bits hold the attenuate select for the right output. Using these bits, you can select attenuation from 0 dB (RDA=0x00) to -94.5 dB (RDA=0x3F) in -1.5 dB increments. The following equation lets you determine the value to load into RDA [5:0]: (Attenuation dB) / (-1.5 dB) = RDA For -12 dB att., example: -12 dB / -1.5 dB = 8 = 0b01000 =RDA
[6] res	Reserved for future expansion. Always write a zero to this bit.
[7] RDM (RW)	Right DAC Mute 0 Un-mutes the right DAC output. 1 Mutes the right DAC output. This bit mutes (1) or un-mutes (0) the right DAC output.

Figure 3.16 Codec Reg.—Clock and Data Format (Index: 0x08)

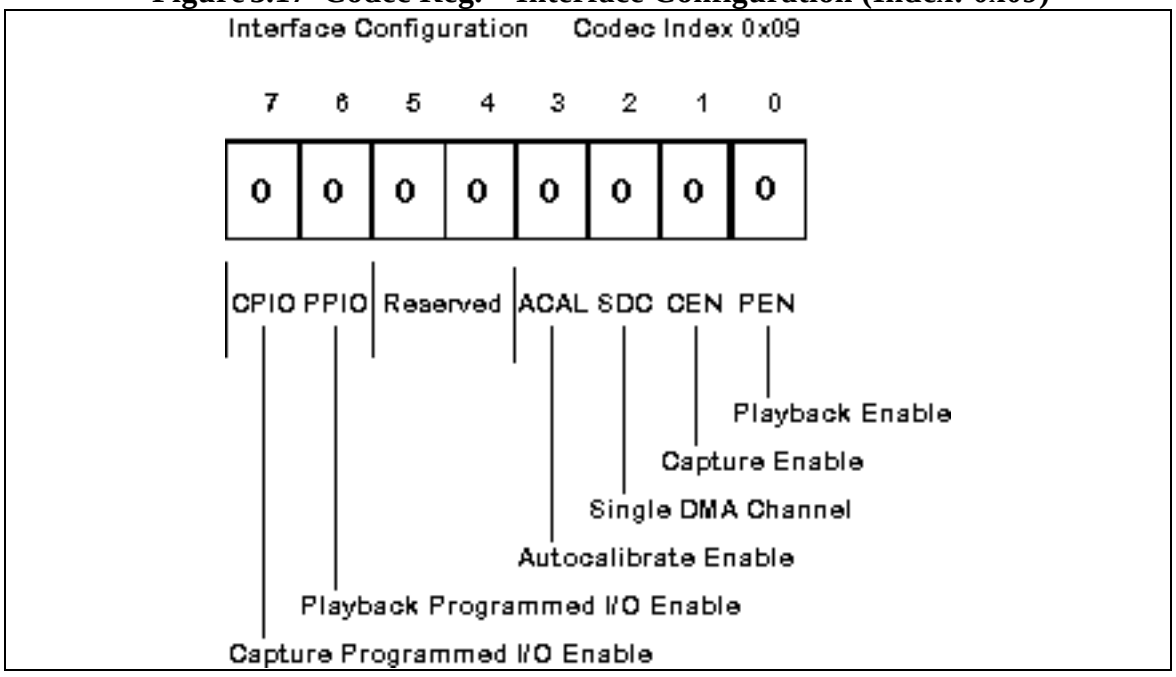


NOTE: To change the bits in this register, place the codec in the Mode Change Enable (MCE) state or set PEN=0.

Bits	*Description* *(Playback Data Format—WSS Codec Index 0x08)*
[1:0] res	Reserved for future expansion. Always write a zero to this bit.
[2] PINF8 (RW)	Playback 8-bit Interface 0 Playback channel is in 16-bit mode. 1 Playback channel is in 8-bit mode This bit indicates that the playback channel is in 8-bit (1) or 16-bit (0) mode. Note: The PFMT bits in this register select among 8-bit or 16-bit interface on the codec; programming this bit's contents must be coordinated with programming the Plug & Play configuration registers selection of 8-bit or 16-bit transfers and DMA channels.
[3] PBSW (RW)	Playback Byte Swap 0 The byte order of output words is at default. 1 The byte order of output words is swapped. For 16-bit interface (PINF8=0), this bit swaps (1) or leaves at default (0) the byte order of output words.

Bits	*Description* *(Clock and Data Format—WSS Codec Index 0x08)* *(Continued)*
[4] PS/M (RW)	Playback Stereo/Mono Select. 0 Mono output audio data stream format is mono. 1 Stereo output audio data stream format is stereo. This bit selects stereo (1) or mono (0) formatting for the output audio data streams. In stereo, the codec alternates samples between channels to provide left and right channel output. For mono, the codec plays the same sample on both channels.
[5] PC/L (RW)	Playback Companded Select /Linear. 0 Linear-digital Representation format for output data. 1 Non-linear, companded format for output data. Use this bit to select a linear-digital, representation format (0) or a non-linear, companded format (1) for output data. The PC/L bit works in concert with the PFMT bits for output format selection.
[7:6] PFMT [1:0] (RW)	Playback Format Select. Use these bits and the PC/L bit to select the format for output data according to the following table. PFMT1 PFMT0 PC/L Playback Audio Data Type 0 0 0 8-bit, unsigned PCM (Linear) 0 0 1 8-bit, µ-Law companded PCM 0 1 0 16-bit, signed Little Endian (Linear) 0 1 1 8-bit, A-Law companded PCM 1 0 0 reserved 1 0 1 4-bit, IMA-ADPCM Companded 1 1 0 16-bit, signed Big Endian (Linear) 1 1 1 reserved

Figure 3.17 Codec Reg.—Interface Configuration (Index: 0x09)



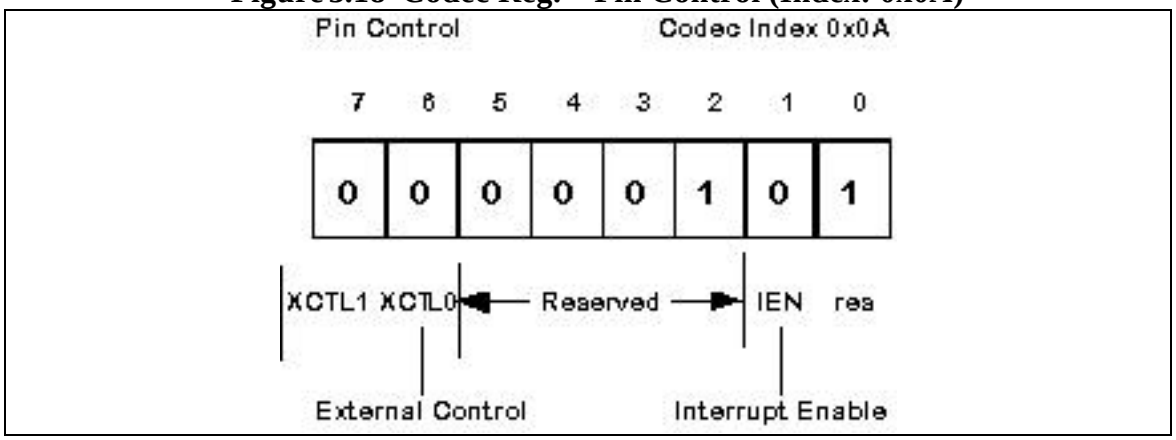
NOTE: Placing the Codec in the Mode Change Enable (MCE) state is not required when changing the CEN and PEN bits in this register.

Bits	*Description* *(Interface Configuration—WSS Codec Index 0x09)*
[0] PEN (RW)	Playback Enable 0 Disables data playback. 1 Enables data playback. This bit enables (1) or disables (0) data playback. If the PPIO bit is set (1), PEN enables programmed I/O playback mode. If the PPIO bit is cleared (0), PEN enables DMA playback mode.
[1] CEN (RW)	Capture Enable 0 Disables data capture. 1 Enables data capture. This bit enables (1) or disables (0) data capture. If the CPIO bit is set (1), CEN enables programmed I/O capture mode. If the CPIO bit is cleared (0), CEN enables DMA capture mode.

Bits	*Description* *(Interface Configuration—WSS Codec Index 0x09)* *(Continued)*
[2] SDC (RW)	Single DMA Channel 0 Dual channel DMA mode selected. 1 Single channel DMA mode selected. This bit selects single channel (1) or dual channel (0) DMA mode. Playback and capture DMA occur on the playback channel in single channel mode. If playback and capture DMA are enabled at the same time (PEN=CEN=1), the codec only performs playback DMA operations. Note that in single channel mode the codec cannot perform simultaneous playback and capture DMA operations. When the SDC is set (1), the single DMA and/or PIO channels are selected for playback and/or capture as follows: Capture Playback SDC CPIO PPIO CEN PEN off DMA 1 0 0 1 1 DMA PIO 1 0 1 1 1 PIO DMA 1 1 0 1 1 PIO PIO 1 1 1 1 1 DMA off 1 0 0 1 0 off DMA 1 0 0 0 1
[3] ACAL (RW)	Autocalibrate Enable 0 Disables codec autocalibration on exiting from a mode change operation. 1 Enables codec autocalibration on exiting from a mode change operation. This bit enables (1) or disables (0) codec autocalibration on exiting from a mode change operation (occurs whenever MCE bit toggles from 1 to 0). Note that the codec performs an autocalibration on the first exit from MCE after the RESET pin is pulsed low regardless of the state of ACAL.
[5:4] res	Reserved for future expansion. Always write zeros to these bits.

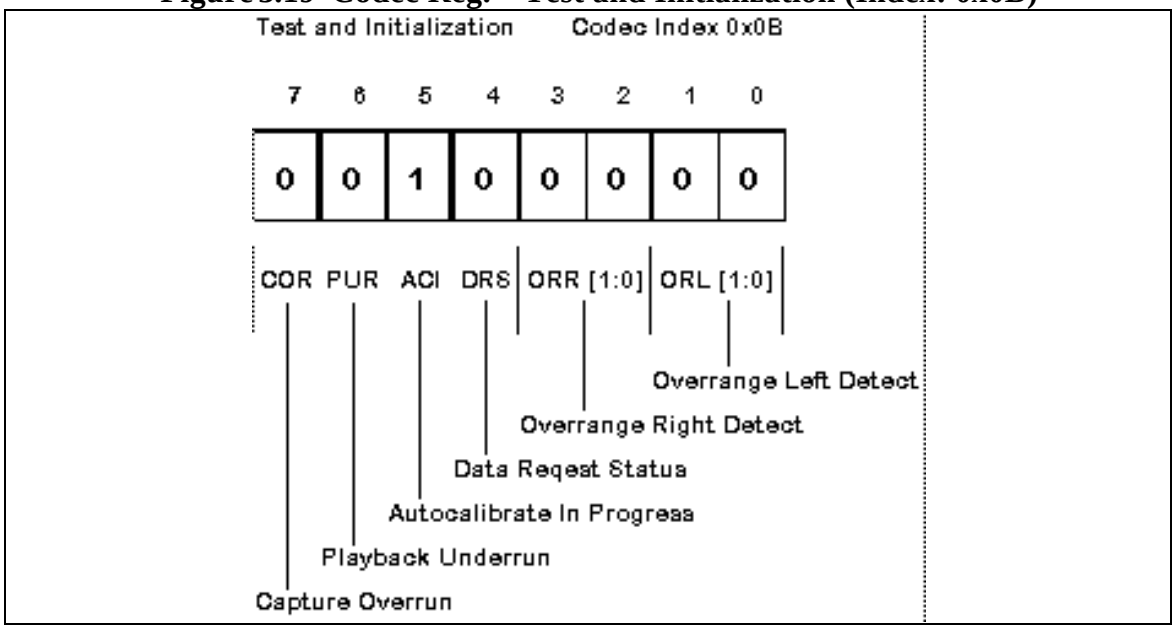
Bits	*Description* *(Interface Configuration—WSS Codec Index 0x09)* *(Continued)*
[6] PPIO (RW)	Playback PIO Enable 0 DMA mode for playback data transfer selected. 1 Programmed I/O mode for playback data transfer selected. This bit selects Programmed I/O (1) or DMA (0) mode for playback data transfers.
[7] CPIO (RW)	Capture PIO Enable 0 DMA mode for capture data transfer selected. 1 Programmed I/O mode for capture data transfer selected. This bit selects Programmed I/O (1) or DMA (0) mode for capture data transfers.

Figure 3.18 Codec Reg.—Pin Control (Index: 0x0A)



Bits	*Description* *(Pin Control—WSS Codec Index 0x0A)*
[0] res	Reserved for future expansion. Always write zeros to these bits.
[1] IEN (RW)	Interrupt Enable 0 Disables codec interrupts. 1 Enables codec interrupts. This bit enables (1) or disables (0) codec interrupts. These interrupts occur when the transfer count in the Base Count register matches the number of samples to transfer. A (1) in the INT bit in WSS Codec Status register indicates an active codec interrupt.
[4:2] res	Reserved for future expansion. Always write zeros to these bits.
[6:5] XCTL[1:0] (RW)	External Control 0 The XCTL[1:0] pin is at TTL logic low. 1 The XCTL[1:0] pin is at TTL logic high. This bit indicates that a TTL logic high (1) or logic low (0) in on the SoundPort Controller's XCTL[1:0] pin.

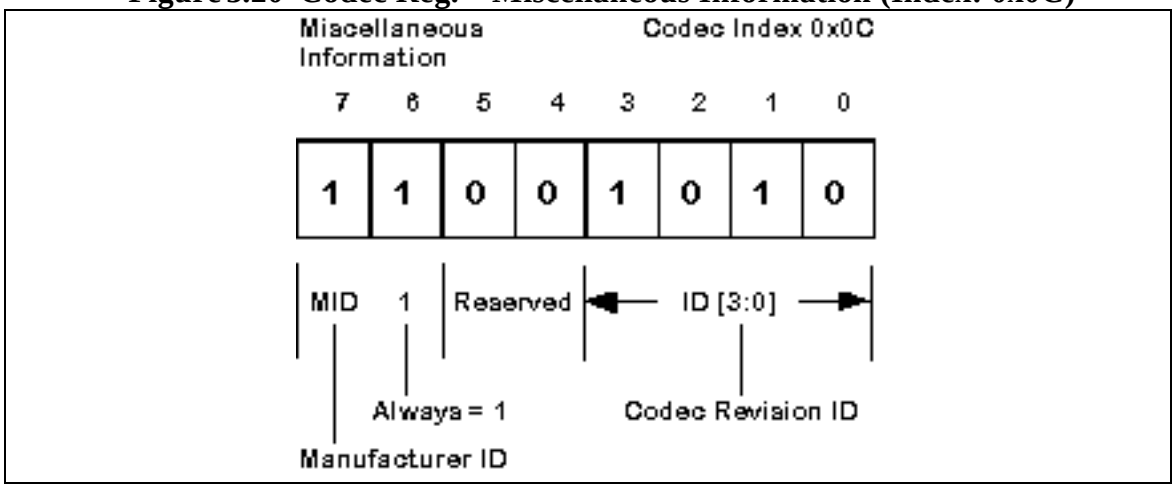
Figure 3.19 Codec Reg.—Test and Initialization (Index: 0x0B)



Bits	*Description* *(Test and Initialization—WSS Codec Index 0x0B)*
[1:0] ORL [1:0] (RO)	Overrange Left Detect These bits indicate over/under range detection on the current left capture channel sample according to the following table: ORL Over/Under Range Detection 0 Less than –1 dB underrange 1 Between –1 dB and 0 dB underrange 2 Between 0 dB and +1 dB overrange 3 Greater than +1 dB overrange
[3:2] ORR [1:0]	Overrange Right Detect These bits indicate over/under range detection on the current right capture channel sample according to the following table: ORR Over/Under Range Detection 0 Less than –1 dB underrange 1 Between –1 dB and 0 dB underrange 2 Between 0 dB and +1 dB overrange 3 Greater than +1 dB overrange
[4] DRS (RO)	Data Request Status 0 A DMA capture/playback request is *not* being serviced. 1 A DMA capture/playback request is being serviced This bit that the codec is currently servicing a DMA capture / playback request (1) or that no request is being serviced (0).

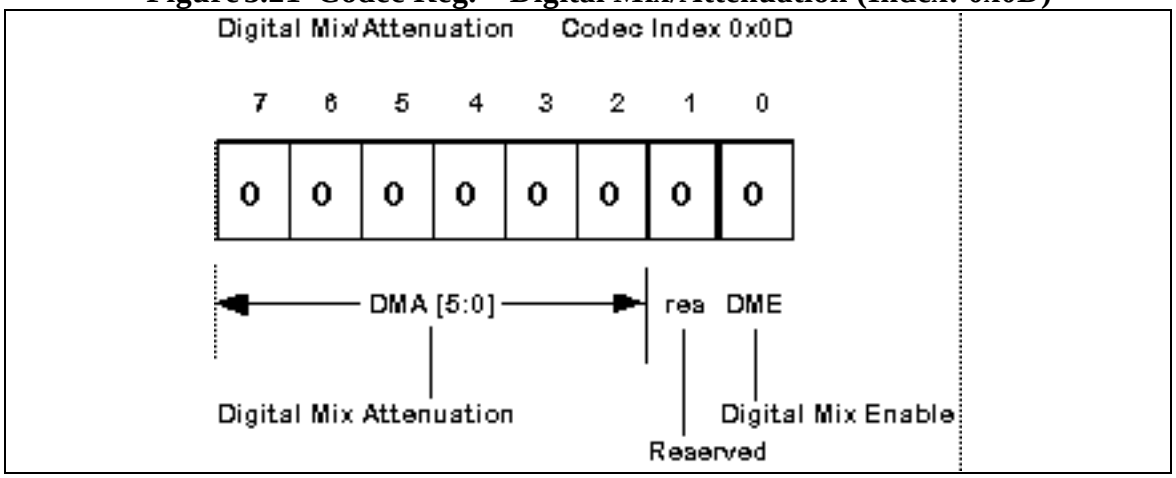
Bits	*Description* *(Test and Initialization—WSS Codec Index 0x0B)* *(Continued)*
[5] ACI (RO)	Autocalibrate-In-Progress 0 Autocalibration is not in progress. 1 Autocalibration is in progress. This bit indicates an autocalibration is in progress (1) or is not in progress (0). This bit is high (1) for 128 sample periods after exiting MCEmode if an autocalibration is not in progress and 384 sample periods if an autocalibration is in progress.
[6] PUR (RO)	Playback Underrun 0 No underrun. 1 Underrun. The codec sets (1) this bit when playback data is not written within 1 sample period after the playback FIFO empties. The codec clears (0) this bit immediately after a four byte playback sample is written. When PUR is set, the playback channel has "run out" of data and either plays back a mid-scale value (DACZ=1) or repeats the last sample (DACZ=0).
[7] COR (RO)	Capture Overrun 0 No overrun. 1 Overrun. The codec sets (1) this bit when capture data is not read within 1 sample period after the capture FIFO fills. When COR is set, the FIFO is full and the codec discards any new data generate. The codec clears (0) this bit immediately after a four byte capture sample is read. Note: The WSS CodecSOUR bit contains the logical OR of the COR and PUR bits

Figure 3.20 Codec Reg.—Miscellaneous Information (Index: 0x0C)



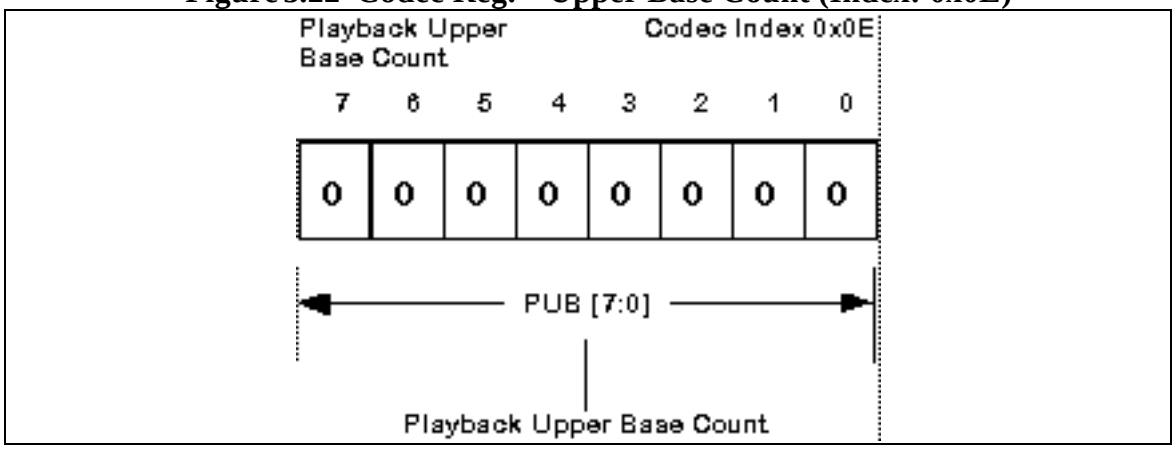
Bits	*Description (Miscellaneous Information—WSS Codec Index 0x0C)*
[3:0] ID [3:0] (RO)	Codec Revision ID. These bits indicate the revision level of the codec. The initial revision level of the codec is 0xA.
[5:4] res	Reserved for future expansion. Always write 0's to these bits.
[6] 1	This bit is always 1.
[7] MID	Manufacturer ID Bit. This bit is set to 1.

Figure 3.21 Codec Reg.—Digital Mix/Attenuation (Index: 0x0D)



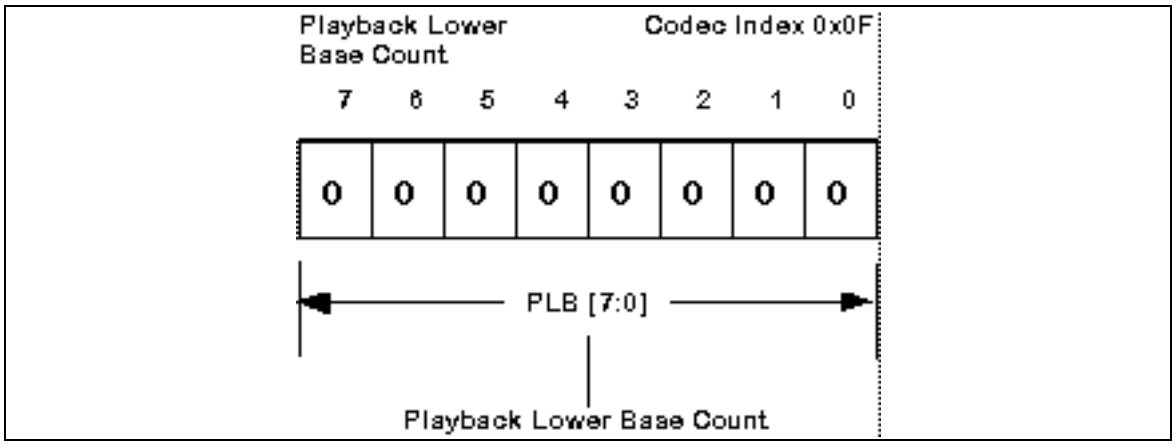
Bits	*Description* *(Digital Mix/Attenuation—WSS Codec Index 0x0D)*
[0] DME (RW)	Digital Mix Enable 0 Mutes digital mixing. 1 Un-mutes digital mixing. This bit enables (1) or mutes (0) digital mixing of the ADCs' output with the DACs' input. When mixing is enabled, the codec digitally mixes the data from the ADCs with the other data delivered to the DACs. If a capture overrun (COR=1) occurs during capture (CEN=1), the codec uses the last sample captured before overrun for the digital mix. If a playback underrun (PUR=1) occurs during playback (PEN=1) with DACZ set (1), the codec adds a mid-scale zero to the digital mix data. If the underrun occurs with DACZ not set (1), the codec repeats the last valid sample for the digital mix.
[1] res	Reserved for future expansion. Always write a zero to this bit.
[7:2] DMA [5:0] (RW)	Digital Mix Attenuation. These bits hold the attenuate select for the ADC data mixed with the DAC input. Using these bits, you can select attenuation from 0 dB (DMA=0x00) to -94.5 dB (DMA=0x3F) in-1.5 dB increments. The following equation lets you determine the value to load into DMA [5:0]: (Attenuation dB) / (-1.5 dB) = DMA For -12 dB att., example: -12 dB / -1.5 dB = 8 = 0b01000 = DMA

Figure 3.22 Codec Reg.—Upper Base Count (Index: 0x0E)



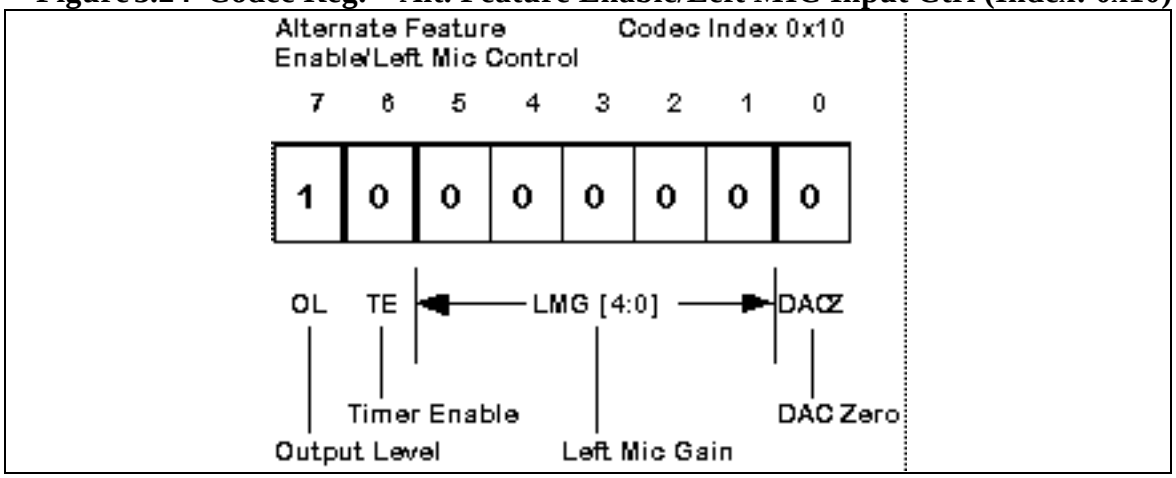
Bits	*Description (Playback Upper Base Count—WSS Codec Index 0x0E)*
[7:0] PUB [7:0] (RW)	Playback Upper Base Count. These bits hold the upper byte of the 16-bit base count (eight most significant bits). Note that reads from this register return the same value written, *not* the current count. *Also note that the current count is not loaded into the counter until you write the upper base count register (write register 0x0F first, then 0x0E)*

Figure 3.23 Codec Reg.—Lower Base Count (Index: 0x0F)



Bits	*Description (Playback Lower Base Count—WSS Codec Index 0x0F)*
[7:0] PLB [7:0] (RW)	Playback Lower Base Count. These bits hold the lower byte of the 16-bit base count (eight least significant bits). Note that reads from this register return the same value written, *not* the current count.

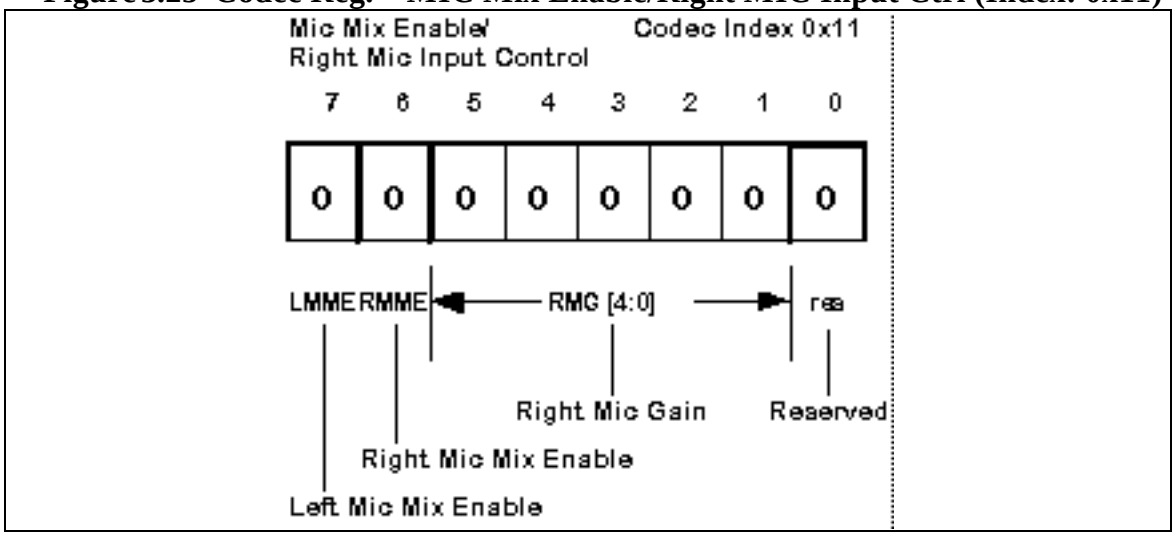
Figure 3.24 Codec Reg.—Alt. Feature Enable/Left MIC Input Ctrl (Index: 0x10)



Bits	*Description* *(Alt. Fea. Enable/Left Mic Input Ctrl—WSS Codec Index 0x10)*
[0] DACZ (RW)	DAC Zero. 0 Output the previous valid sample. 1 Output a mid-scale value. During an underrun condition, this bit directs the DAC to output a mid-scale value (1) or output the previous valid sample (0).
[5:1] LMG [4:0] (RW)	Left Mic Gain. These bits hold the gain/attenuate select for the left microphone input. Using these bits, you can select gain/attenuates from 12 dB (LMG=0x00) to -34.5 dB (LMG=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into LMG [4:0]: ((Gain *or* -Atten. dB) - 12 dB) / (-1.5 dB) = LMG For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = LMG Note: For 0 dB gain, set LMG [4:0] = 0b01000

Bits	*Description* *(Alt. Fea. Enable/Left Mic Input Ctrl—WSS Codec Index 0x10) (Continued)*
[6] TE (RW)	Timer Enable 0 Disables the timer. 1 Enables the timer. This bit enables (1) or disables (0) the 16-bit programmable timer loaded by the Upper and Lower Timer Count registers (Indices 0x14 & 0x15). When you enable the timer, the codec loads the timer count and generates interrupts at the specified period.
[7] OL (RW)	Output Level 0 2 Vpp full scale analog output. 1 2.8 Vpp full scale analog output. This bit selects the analog output level attenuation as either full scale 2.8 Vpp signal applying no attenuation (1) or full scale 2 Vpp signal applying -3 dB attenuation (0).

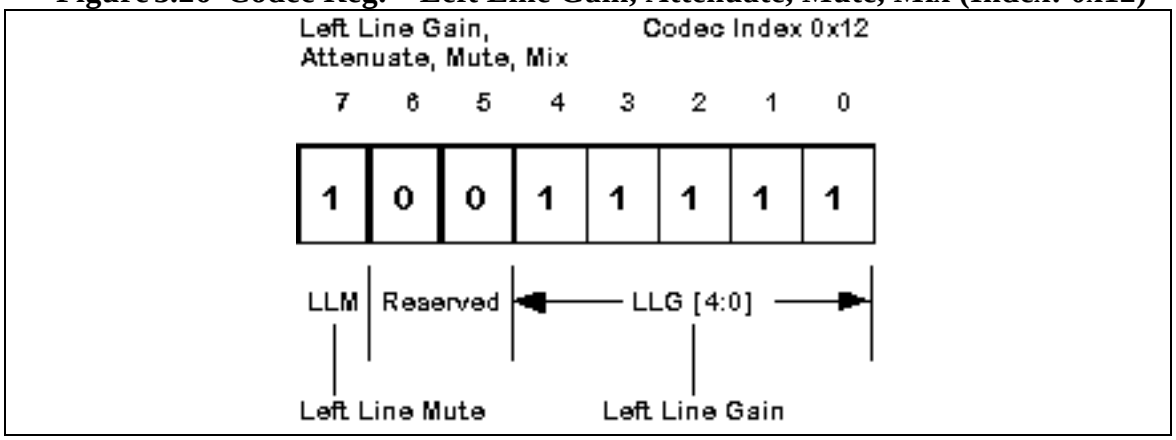
Figure 3.25 Codec Reg.—MIC Mix Enable/Right MIC Input Ctrl (Index: 0x11)



Bits	*Description* *(Mic Mix Enable/Right Mic Input Ctrl—WSS Codec Index 0x11)*
[0] res	Reserved for future expansion. Always write 0's to these bits.
[5:1] RMG [4:0] (RW)	Right Mic Gain These bits hold the gain/attenuate select for the right microphone input. Using these bits, you can select gain/attenuates from 12 dB (RMG=0x00) to -34.5 dB (RMG=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into RMG [4:0]: ((Gain *or* Atten. dB) - 12 dB) / (-1.5 dB) = RMG For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = RMG Note: For 0 dB gain, set RMG [4:0] = 0b01000
[6] RMME (RW)	Right Mic Mix Enable 0 Mutes mixing of right microphone input with DAC output on the R_OUT line. 1 Enables mixing of right microphone input with DAC output on the R_OUT line. This bit enables (1) or mutes (0) mixing of right microphone input with DAC output on the R_OUT line. Note the RMME and LMME bits mute when cleared (0) and enable when set (1); these bits operate opposite of all other mute/enable bits in the codec.

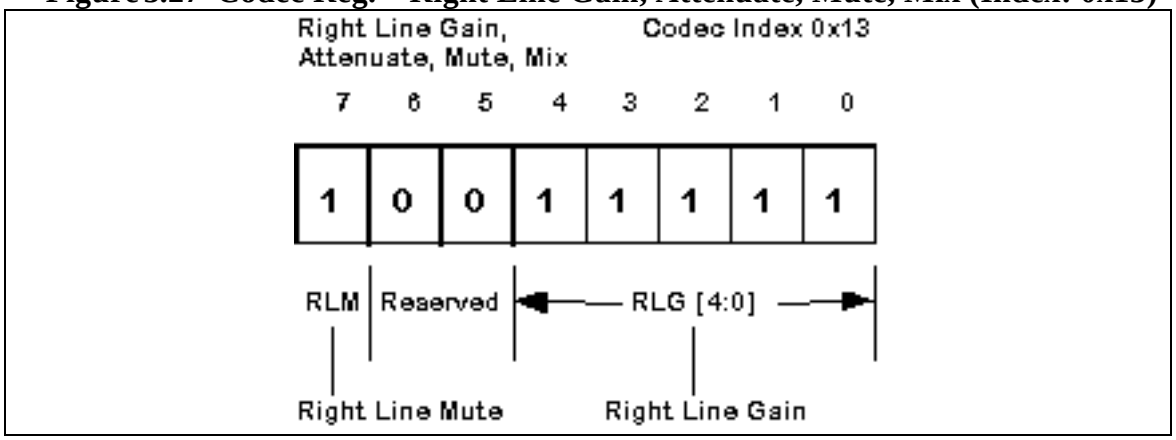
Bits	*Description* *(Mic Mix Enable/Right Mic Input Ctrl—WSS Codec Index 0x11)* *(Continued)*
[7] LMME (RW)	Left Mic Mix Enable 0 Mutes mixing of left microphone input with DAC output on the L_OUT line. 1 Enables mixing of left microphone input with DAC output on the L_OUT line. This bit enables (1) or mutes (0) mixing of left microphone input with DAC output on the L_OUT line. Note the RMME and LMME bits mute when cleared (0) and enable when set (1); these bits operate opposite of all other mute/enable bits in the codec.

Figure 3.26 Codec Reg.—Left Line Gain, Attenuate, Mute, Mix (Index: 0x12)



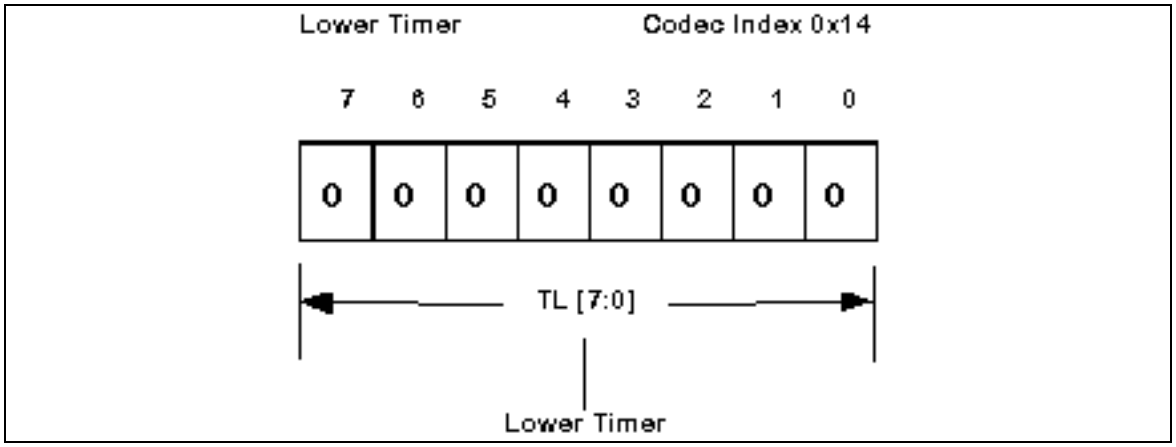
Bits	*Description* *(Left Line Gain, Atten., Mute, Mix—WSS Codec Index 0x12)*
[4:0] LLG [4:0] (RW)	Left Line Mix Gain. These bits hold the gain/attenuate select for the left line mix gain. Using these bits, you can select gain/attenuates from 12 dB (LLG=0x00) to -34.5 dB (LLG=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into LLG [4:0]: ((Gain *or* Atten. dB) - 12 dB) / (-1.5 dB) = LLG For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = LLG Note: For 0 dB gain, set LLG [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] LLM (RW)	Left Line Mute 0 Un-mutes left line input into the output mixer. 1 Mutes left line input into the output mixer. This bit mutes (1) or un-mutes (0) the left line input into the output mixer.

Figure 3.27 Codec Reg.—Right Line Gain, Attenuate, Mute, Mix (Index: 0x13)



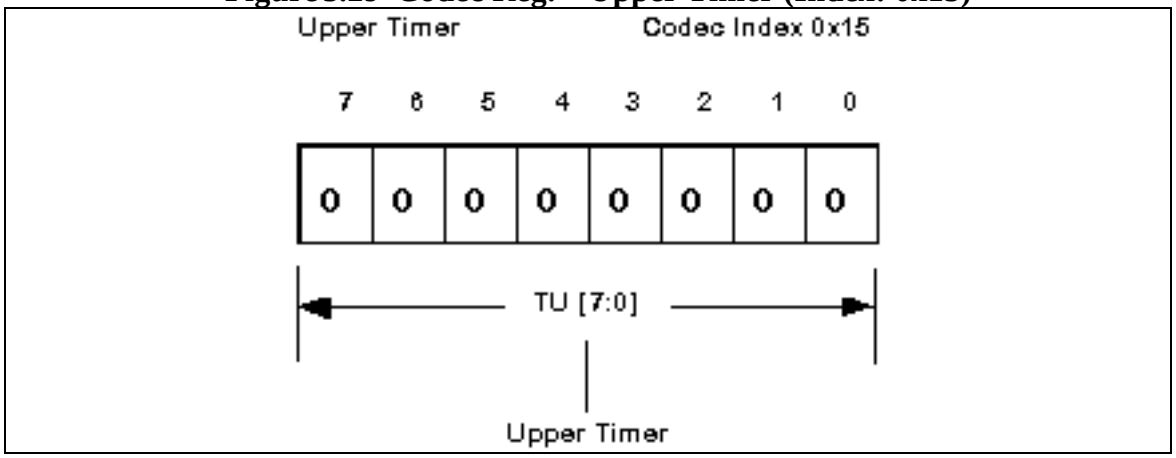
Bits	*Description* *(Right Line Gain, Atten., Mute, Mix—WSS Codec Index 0x13)*
[4:0] RLG [4:0] (RW)	Right Line Mix Gain. These bits hold the gain/attenuate select for the right line mix gain. Using these bits, you can select gain/attenuates from 12 dB (RLG=0x00) to -34.5 dB (RLG=0x1F) in -1.5 dB increments. The following equation lets you determine the value to load into RLG [4:0]: ((Gain-*or*-Atten. dB) - 12 dB) / (-1.5 dB) = RLG For +3 dB gain, example: (3dB - 12 dB) / (-1.5 dB) = 6 = 0b00110 = RLG Note: For 0 dB gain, set RLG [4:0] = 0b01000
[6:5] res	Reserved for future expansion. Always write zeros to these bits.
[7] RLM (RW)	Right Line Mute 0 Un-mutes right line input into the output mixer. 1 Mutes right line input into the output mixer. This bit mutes (1) or un-mutes (0) the right line input into the output mixer.

Figure 3.28 Codec Reg.—Lower Timer (Index: 0x14)



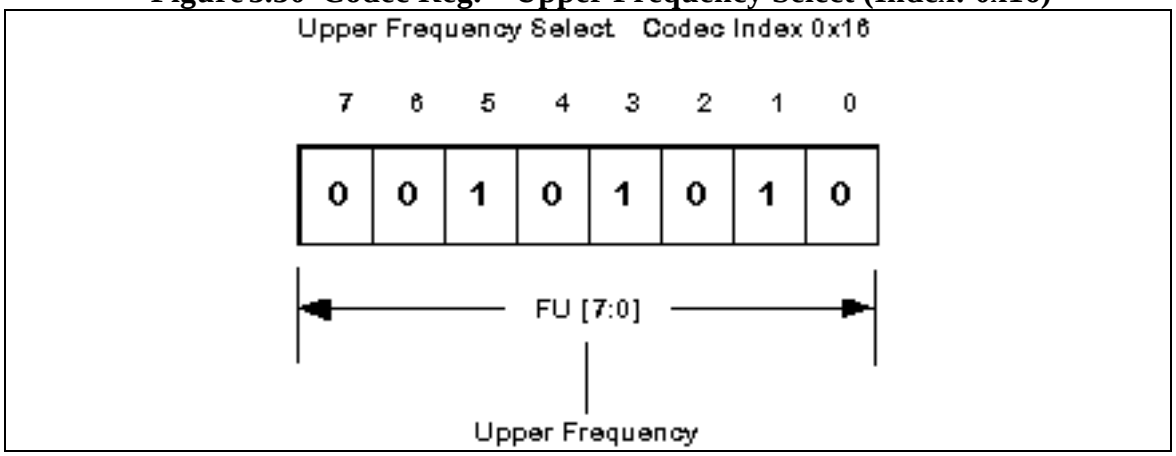
Bits	*Description* *(Lower Timer—WSS Codec Index 0x14)*
[7:0] TL [7:0] (RW)	Lower Timer. These bits hold the lower byte of the 16-bit timer (eight least significant bits). Note that reads from this register return the same value written, *not* the current count. The time base is calculated from the following formula: (PC system clock) / 144 = Time Base. Example: 14.3818 MHz / 144 = 10.057 µs

Figure 3.29 Codec Reg.—Upper Timer (Index: 0x15)



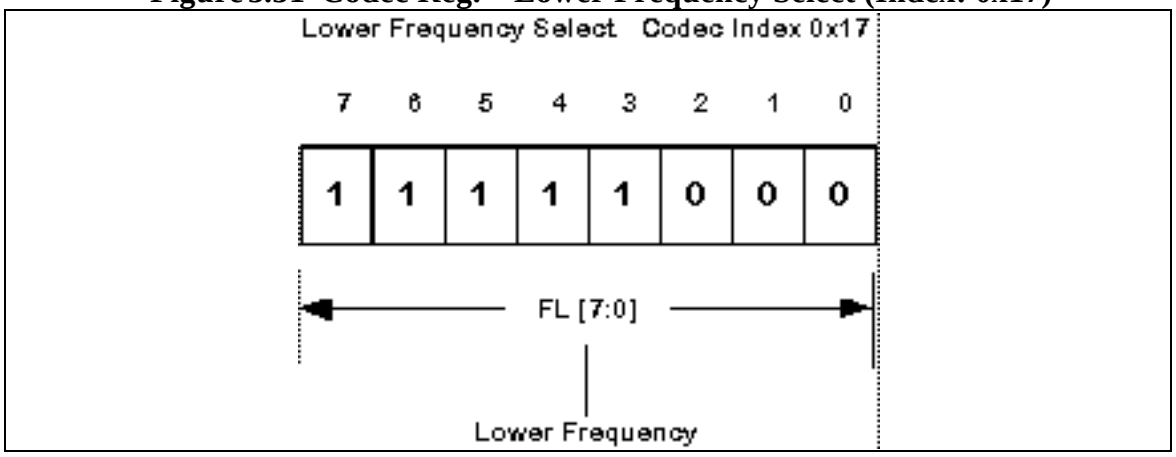
Bits	*Description* *(Upper Timer—WSS Codec Index 0x15)*
[7:0] TU [7:0] (RW)	Upper Timer. These bits hold the upper byte of the 16-bit timer (eight most significant bits). Note that reads from this register return the same value written, *not* the current count. The time base is calculated from the following formula: (PC system clock) / 144 = Time Base. Example: 14.3818 MHz / 144 = 10.057 µs. *Also note that the timer is not loaded until you write the upper timer register (write register 0x14 first, then 0x15)*

Figure 3.30 Codec Reg.—Upper Frequency Select (Index: 0x16)



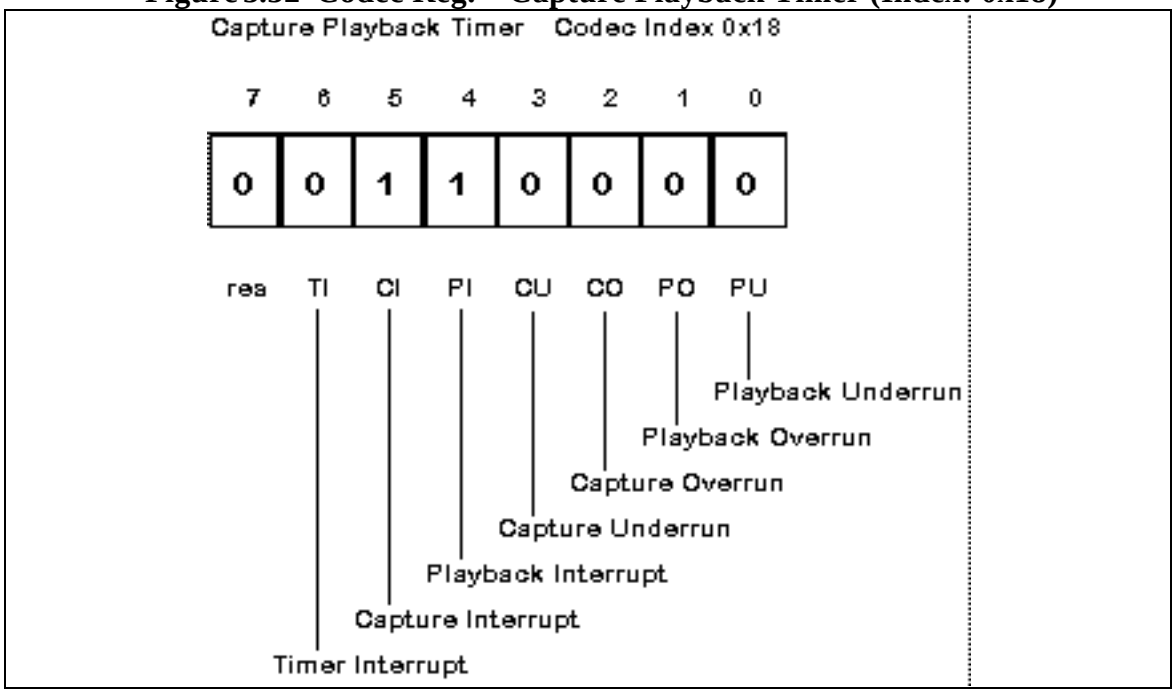
Bits	*Description* *(Upper Frequency Select—WSS Codec Index 0x16)*
[7:0] FU [7:0] (RW)	Upper Frequency Select. These bits hold the upper byte of the 16-bit sampling frequency selection (eight most significant bits). Using these bits and the bits in the Lower Frequency Select register, you can select sampling frequencies from 5 kHz (FU [7:0] & FL [7:0] = 0x0FA0) to 50 kHz (FU[7:0] & FL [7:0] = 0xC350) in 1 kHz increments. To find the value to load into FU[7:0] & FL [7:0], convert the decimal sampling frequency (between 5 to 50 kHz) to a two byte binary number. To sample at 8.0 kHz for example: 8000 = 0b0001,1111,0100,0000 = FU [7:0] & FL [7:0] Some commonly used sample rates are listed below: Quality — Sampling Frequency — FU [7:0] — FL [7:0] Voice — 8.0 kHz — 0001 1111 — 0100 0000 (default) Radio — 11.025 kHz — 0010 1011 — 0001 0001 Tape — 22.05 kHz — 0101 0110 — 0010 0010 CD — 44.1 kHz — 1010 1100 — 0100 0100 DAT — 48.0 kHz — 1011 1011 — 1000 0000 *Also note that the timer is not loaded until you write the upper timer register (write register 0x14 first, then 0x15)* *[Load FU before FL]*

Figure 3.31 Codec Reg.—Lower Frequency Select (Index: 0x17)



Bits	*Description (Lower Frequency Select—WSS Codec Index 0x17)*
[7:0] FL [7:0]	Lower Frequency Select. These bits hold the lower byte of the 16-bit sampling frequency selection (eight least significant bits). Using these bits and the bits in the Upper Frequency Select register, you can select sampling frequencies from 4 kHz (FU [7:0] & FL [7:0] = 0x0FA0) to 50 kHz (FU[7:0] & FL [7:0] = 0xC350) in 1 kHz increments. *Note that the sample rate is not loaded into the until you write the lower frequency register (write register 0x16 first, then 0x17)* To find the value to load into FU[7:0] & FL [7:0], convert the decimal sampling frequency (between 5 to 50 kHz) to a two byte binary number. To sample at 8.0 kHz for example: 8000 = 0b0001,1111,0100,0000 = FU [7:0] & FL [7:0] Some commonly used sample rates are listed below: Quality — Sampling Frequency — FU [7:0] — FL [7:0] Voice — 8.0 kHz — 0001 1111 — 0100 0000 (default) Radio — 11.025 kHz — 0010 1011 — 0001 0001 Tape — 22.05 kHz — 0101 0110 — 0010 0010 CD — 44.1 kHz — 1010 1100 — 0100 0100 DAT — 48.0 kHz — 1011 1011 — 1000 0000

Figure 3.32 Codec Reg.—Capture Playback Timer (Index: 0x18)

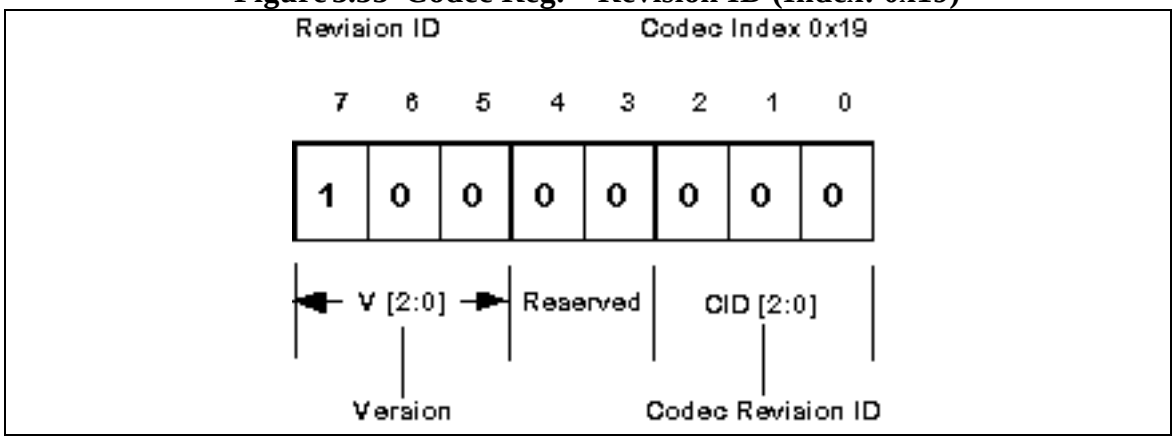


Bits	*Description* *(Capture Playback Timer—WSS Codec Index 0x18)*
[0] PU (RO)	Playback Underrun. 0 No underrun. 1 Underrun. The codec sets (1) this bit when the DAC runs out of data and a sample has been missed. The codec clears (0) this bit when the DAC receives a fresh sample.
[1] PO (RO)	Playback Overrun. 0 No overrun. 1 Overrun. The codec sets (1) this bit when a bus write is ignored because the FIFO is full. The codec clears (0) this bit when there is space in the FIFO for the next write.
[2] CO (RO)	Capture Overrun. 0 No overrun. 1 Overrun. The codec sets (1) this bit when a sample from the ADC is ignored because the ADC FIFO is full. The codec clears (0) this bit when there is space in the ADC FIFO for the next write.

Bits	*Description* *(Capture Playback Timer—WSS Codec Index 0x18)* *(Continued)*
[3] CU (RO)	Capture Underrun 0 No underrun. 1 Underrun. The codec sets (1) this bit when it has to repeat the last valid byte sent for a bus read from an empty ADC FIFO. The codec clears (0) this bit when the ADC FIFO has data for the next read.
[4] PI (R) (Write 0)	Playback Interrupt 0 No interrupt. 1 Interrupt is pending from the playback DMA count registers. The codec sets (1) this bit to indicate that an interrupt is pending from the playback DMA count registers. Clear (write 0 to) the PI bit to clear the playback DMA interrupt.
[5] CI (R) (Write 0)	Capture Interrupt. 0 No interrupt. 1 Interrupt is pending from the capture DMA count registers. The codec sets (1) this bit to indicate that an interrupt is pending from the capture DMA count registers. Clear (write 0 to) the CI bit to clear the capture DMA interrupt.
[6] TI (R) (Write 0)	Timer Interrupt. 0 No interrupt. 1 Interrupt is pending from the timer count registers. The codec sets (1) this bit to indicate that an interrupt is pending from the timer count registers. Clear (write 0 to) the TI bit to clear the timer interrupt.
[7] res	Reserved for future expansion. Always write zeros to these bits.

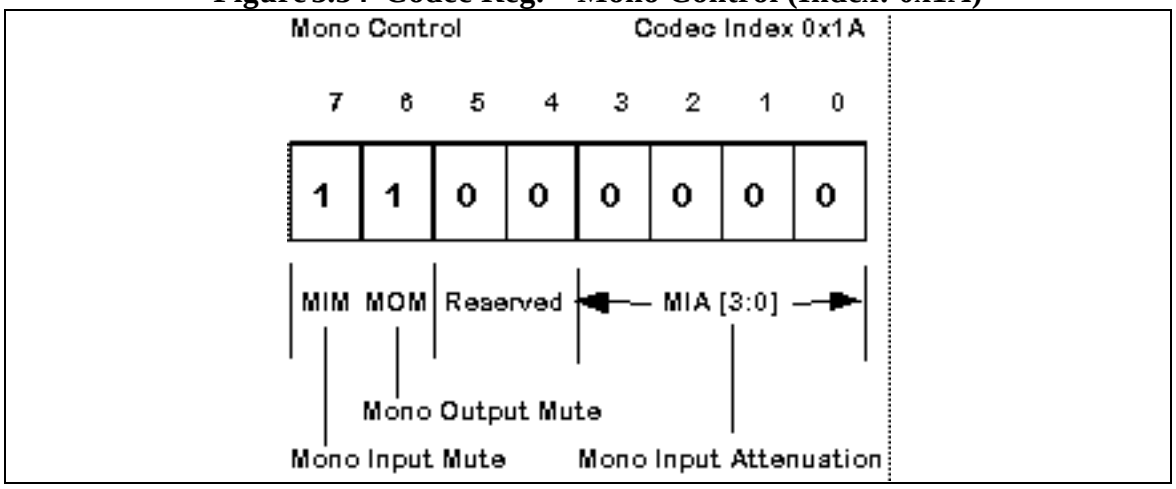
Note: You can clear all three interrupts (PI, CI & TI) by writing to the Codec Status register. The INT bit in the Codec Status register is a logical OR of these three interrupts. While you can clear (0) the PI, CI & TI bits, setting (1) these bits manually has no effect.

Figure 3.33 Codec Reg.—Revision ID (Index: 0x19)



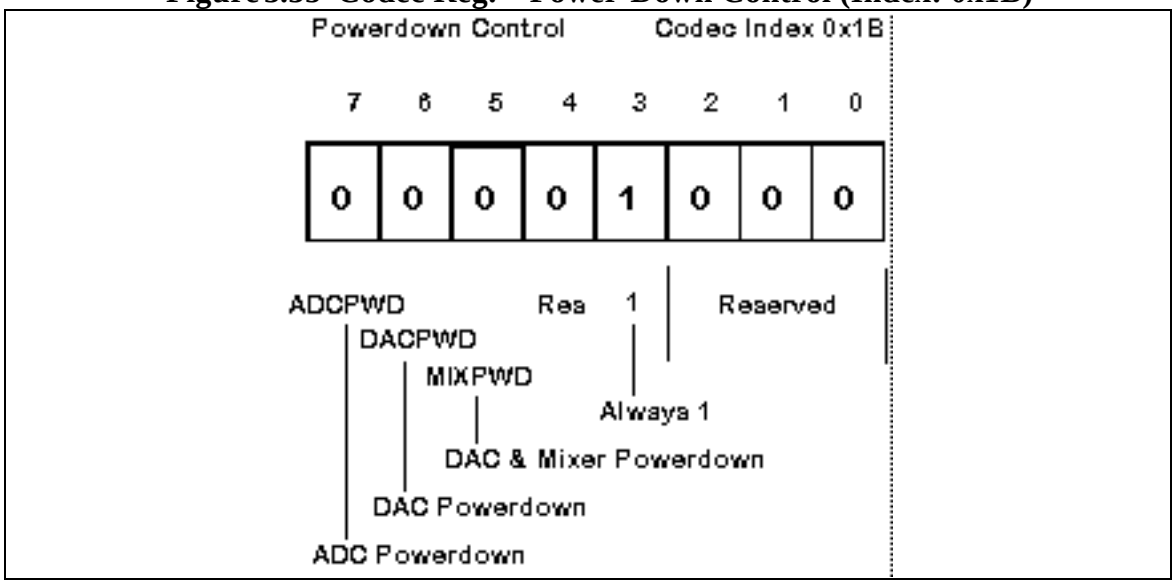
Bits	*Description* *(Revision ID—WSS Codec Index 0x19)*
[2:0] CID [2:0]	Revision ID Number
[4:3] res	Reserved for future expansion. Always write zeros to these bits.
[7:5] V [2:0]	Version Number Indicates the version of the Codec.

Figure 3.34 Codec Reg.—Mono Control (Index: 0x1A)



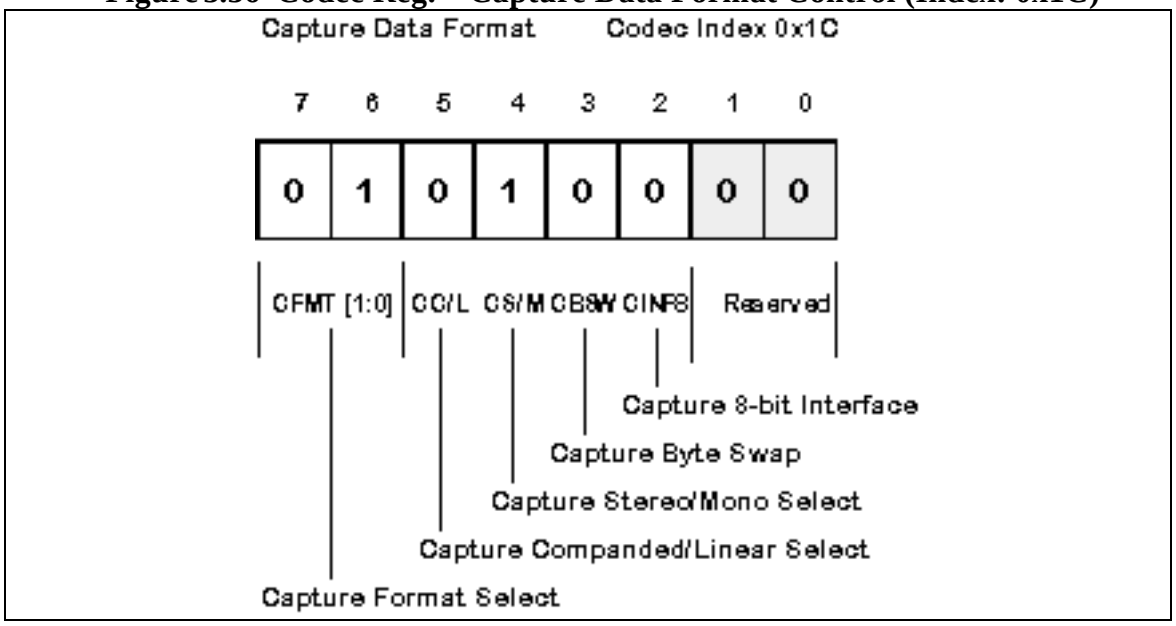
Bits	*Description* *(Mono Control—WSS Codec Index 0x1A)*
[3:0] MIA [3:0] (RW)	Mono Input Attenuation select. These bits hold the attenuate select for the mono input. Using these bits, you can select attenuates from 0 dB (MIA=0x0) to -45 dB (MIA=0xF) in 3 dB increments. The following equation lets you determine the value to load into MIA [3:0]: (Attenuation dB) / (-3 dB) = MIA For -9 dB attenuation, example: -9 dB / -3 dB = 3 = 0b0011 = MIA
[5:4] res	Reserved for future expansion. Always write zeros to these bits.
[6] MOM (RW)	Mono Output Mute. 0 Un-mutes Mono output signal. 1 Mutes Mono output signal. This bit mutes (1) or un-mutes (0) the mono output (M_OUT) signal.
[7] MIM (RW)	Mono Input Mute. 0 Un-mutes Mono input signal. 1 Mutes Mono input signal. This bit mutes (1) or un-mutes (0) the mono input (M_IN) signal.

Figure 3.35 Codec Reg.—Power-Down Control (Index: 0x1B)



Bits	*Description* *(Powerdown Control—WSS Codec Index 0x1B)*
[2:0] res	Reserved for future expansion. Always write zeros to these bits.
[3] 1	This bit is always 1.
[4]	Reserved
[5] MIXPWD (RW)	DAC and Mixer Power Down. Setting this bit powers the DAC and mixer down. When the DAC is powered down, the DAC sample clock is turned off.
[6] DACPWD (RW)	DAC Power Down. Setting this bit powers the DAC down. When the DAC is powered down, the DAC sample clock is turned off.
[7] ADCPWD (RW)	ADC Power Down. Setting this bit powers the ADC down. When the ADC is powered down, the ADC sample clock is turned off.

Figure 3.36 Codec Reg.—Capture Data Format Control (Index: 0x1C)

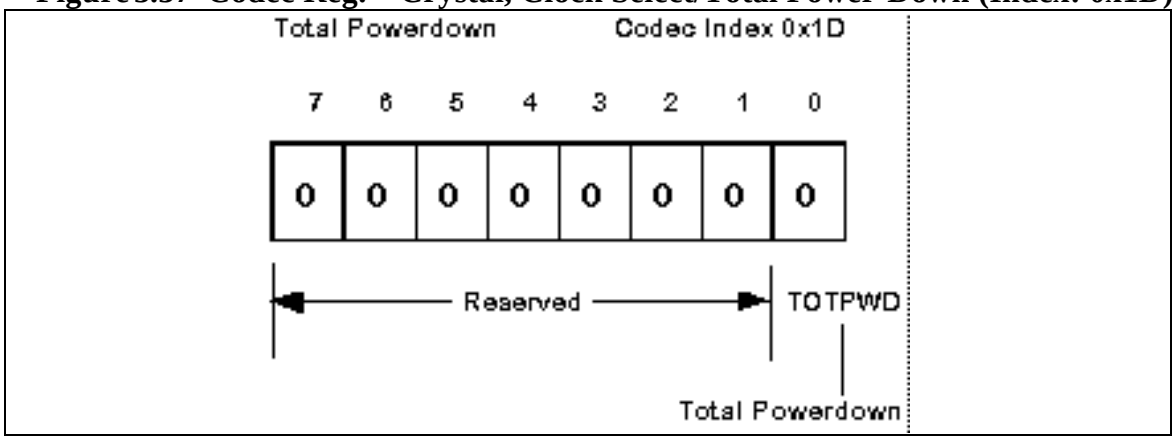


NOTE: To change the bits in this register, place the codec in the Mode Change Enable (MCE) state or set CEN=0.

Bits	*Description* *(Capture Data Format Control—WSS Codec Index 0x1C)*
[1:0] res	Reserved for future expansion. Always write zeros to these bits.
[2] CINF8 (RW)	Capture 8-bit Interface. 0 16-bit capture mode. 1 8-bit capture mode. This bit indicates that the capture channel is in 8-bit (1) or 16-bit (0) mode. Note: The CFMT bits in this register select among 8-bit or 16-bit interface on the codec; programming this bit's contents must be coordinated with programming the Plug & Play configuration registers selection of 8-bit or 16-bit transfers and DMA channels.
[3] CBSW (RW)	Capture Byte Swap. 0 Default capture word byte order. 1 Swaps capture word byte order. For 16-bit data transfers (CINF8=0), this bit swaps (1) or leaves at default (0) the byte order of capture words.

Bits	*Description* *(Capture Data Format Control—WSS Codec Index 0x1C)* *(Continued)*
[4] CS/M (RW)	Capture Stereo/Mono Select 0 Mono input format. 1 Stereo input format. This bit selects stereo (1) or mono (0) formatting for the input audio data streams. In stereo, the codec alternates samples between channels to provide left and right channel input. For mono, the codec captures samples on the left channel.
[5] CC/L (RW)	Capture Companded Select /Linear 0 Linear-digital representation format. 1 Non-linear, companded format. Use this bit to select a linear-digital, representation format (0) or a non-linear, companded format (1) for input data. The CC/L bit works in concert with the CFMT bits for input format selection.
[7:6] CFMT [1:0]	Capture Format Select. Use these bits and the CC/L bit to select the format for input data according to the following table. CFMT1 CFMT0 CC/L Playback Audio Data Type 0 0 0 8-bit, unsigned PCM (Linear) 0 0 1 8-bit, µ-Law companded PCM 0 1 0 16-bit, signed Little Endian (Linear) 0 1 1 8-bit, A-Law companded PCM 1 0 0 reserved 1 0 1 4-bit, IMA-ADPCM Companded 1 1 0 16-bit, signed Big Endian (Linear) 1 1 1 reserved

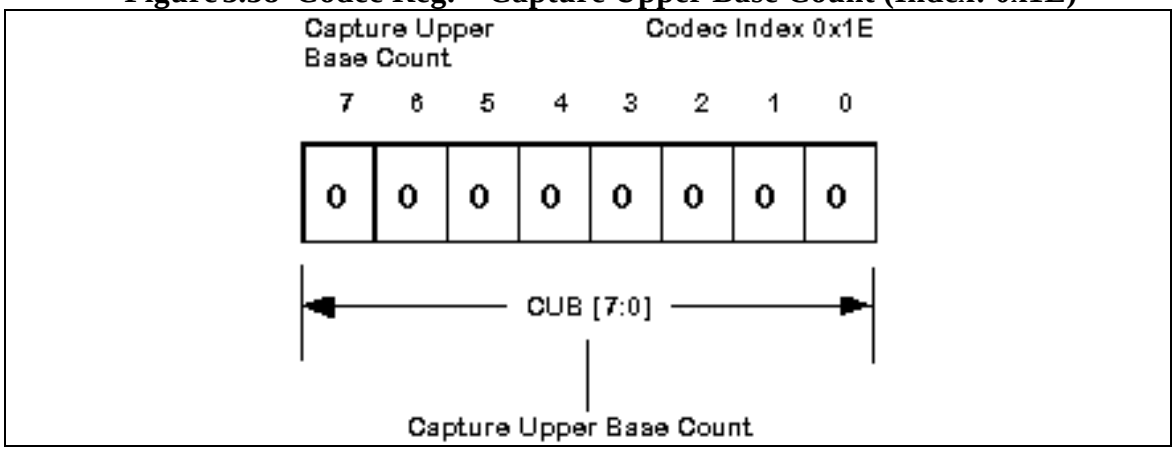
Figure 3.37 Codec Reg.—Crystal, Clock Select/Total Power-Down (Index: 0x1D)



NOTE: To change the bits in this register, place the codec in the Mode Change Enable (MCE) state.

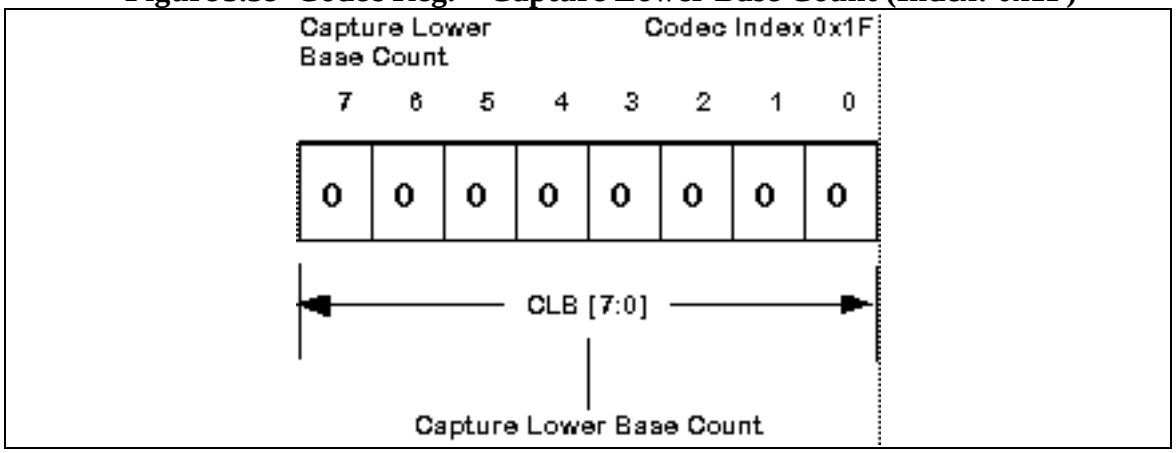
Bits	*Description (Crystal, Clock Sel./Total Powerdown—WSS Codec Index 0x1D)*
[0] TOTPWD (RW)	Total Power Down. Setting this bit powers the DAC, ADC, and mixer down. When the DAC and ADC are powered down, the DAC and ADC sample clocks are turned off. Note: When the codec has set (1) this bit, only the digital interface on-chip remains active awaiting the power up signal.
[7:1] res	Reserved for future expansion. Always write zeros to these bits.

Figure 3.38 Codec Reg.—Capture Upper Base Count (Index: 0x1E)



Bits	*Description (Capture Upper Base Count—WSS Codec Index 0x1E)*
[7:0] CUB [7:0] (RW)	Capture Upper Base Count. These bits hold the upper byte of the 16-bit base count (eight most significant bits). Note that reads from this register return the same value written, *not* the current count. *Also note that the current counter is not loaded until you write the upper base count register (write register 0x1F first, then 0x1E)*

Figure 3.39 Codec Reg.—Capture Lower Base Count (Index: 0x1F)



Bits	*Description (Capture Lower Base Count—WSS Codec Index 0x1F)*
[7:0] CLB [7:0] (RW)	Capture Lower Base Count. These bits hold the lower byte of the 16-bit base count (eight least significant bits). Note that reads from this register return the same value written, *not* the current count.

3.4 AD1812 Sound Blaster Pro ISA Bus Registers (Ports)

The AD1812 contains a set of ISA Bus registers (ports) that correspond to those used by the Sound Blaster Pro audio card from Creative labs. Table 3.8 lists the ISA Bus Sound Blaster Pro registers. For complete information on using these registers, see the *Developer Kit for Sound Blaster Series*, 2nd ed. ©1993, Creative Labs, Inc., 1901 McCarthy Blvd., Milpitas, CA 95035.

Table 3.8 Sound Blaster Pro ISA Bus Registers

Register Name	*Address*
Music0: Address (w), Status (r)	0x(SB Base) Relocatable in range 0x010 - 0x3F0
Music0: Data (w)	0x(SB Base+1)
Music1: Address (w)	0x(SB Base+2)
Music1: Data (w)	0x(SB Base+3)
Mixer Address (w)	0x(SB Base+4)
Mixer Data (w)	0x(SB Base+5)
Reset (w)	0x(SB Base+6)
Music0: Address (w)	0x(SB Base+8)
Music0: Data (w)	0x(SB Base+9)
Input Data (r)	0x(SB Base+A)
Status (r), Output Data (w)	0x(SB Base+C)
Status (r)	0x(SB Base+E)

3.5 AD1812 AdLib ISA Bus Registers (Ports)

The AD1812 contains a set of ISA Bus registers (ports) that correspond to those used by the AdLib audio card from AdLib Multimedia. Table 3.9 lists the ISA Bus AdLib registers. Because the AdLib registers are a subset of those in the Sound Blaster card, you can find complete information on using these registers in the *Developer Kit for Sound Blaster Series*, 2nd ed. ©1993, Creative Labs, Inc., 1901 McCarthy Blvd., Milpitas, CA 95035.

Table 3.9 AdLib ISA Bus Registers

Register Name	*Address*
Music0: Address (w), Status (r)	0x(Adlib Base) Relocatable in range 0x008 - 0x3F8
Music0: Data (w)	0x(Adlib Base+1)
Music1: Address (w)	0x(Adlib Base+2)
Music1: Data (w)	0x(Adlib Base+3)

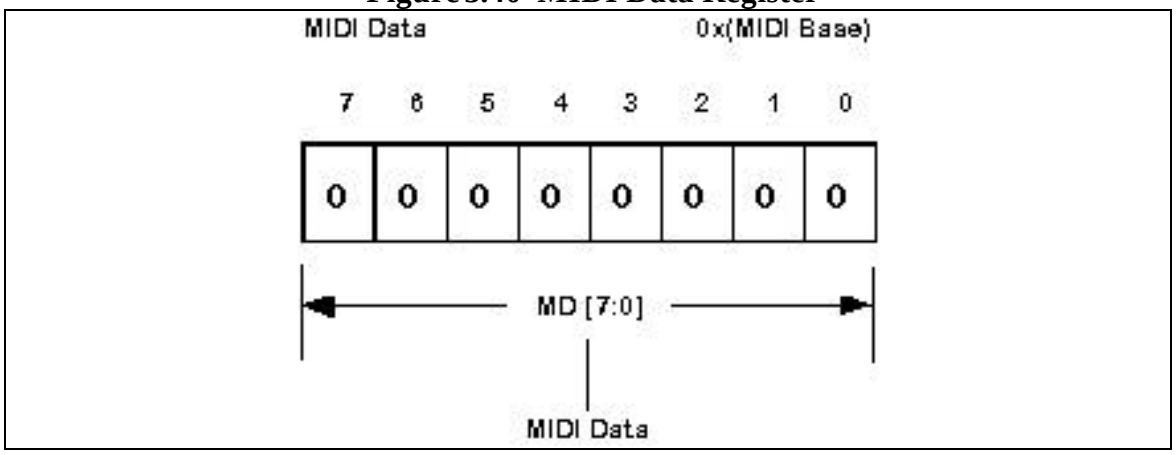
3.6 AD1812 MIDI MPU-401 ISA Bus Registers (Ports)

The AD1812 contains a set of ISA Bus registers (ports) that correspond to those used by the ISA bus MIDI audio interface cards. Table 3.10 lists the ISA Bus MIDI registers and Figures 3.39 and 3.40 show them. These registers support commands and data transfers described in the *MIDI 1.0 Detailed Specification & Standard MIDI Files 1.0*, ©1994, MIDI Manufacturers Association, PO Box 3173 La Habra, CA 90632-3173.

Table 3.10 MIDI ISA Bus Registers

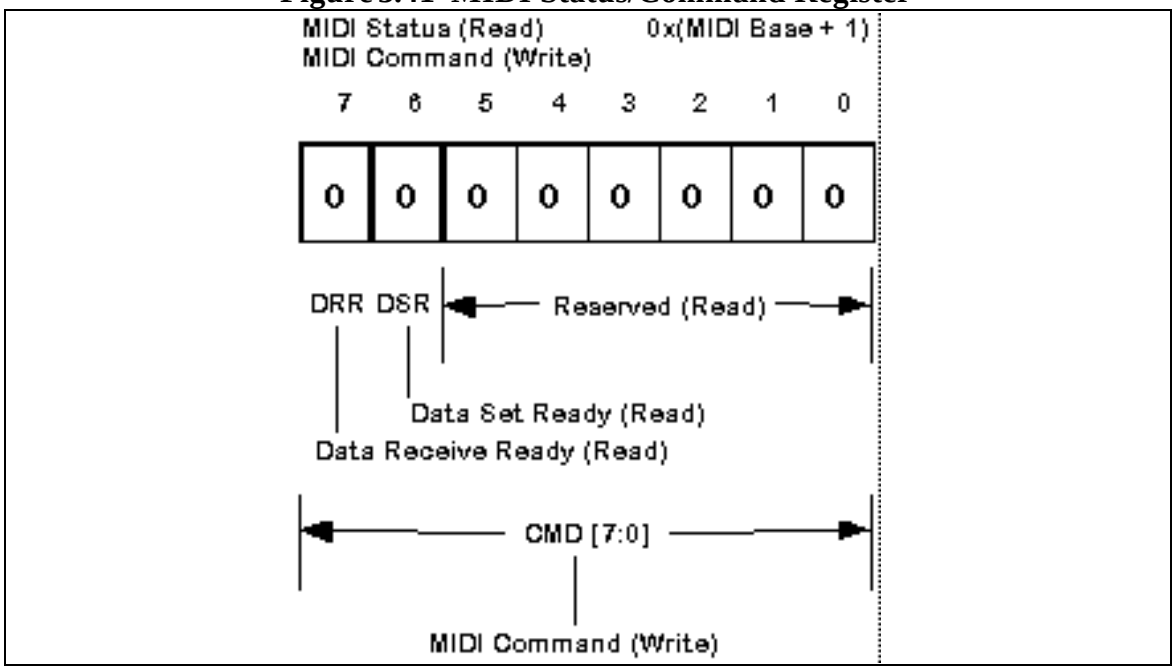
Register Name	*Address*
MIDI Data (r/w)	0x(MIDI Base) Relocatable in range 0x008 - 0x3F8
MIDI Status (r), Command (w)	0x(MIDI Base+1)

Figure 3.40 MIDI Data Register



Bits	*Description* *(MIDI Data—0x(MIDI Base))*
[7:0] MD [7:0] (RW)	MIDI Data. Use these bits for reading/writing MIDI data to the MIDI MPU-401 UART.

Figure 3.41 MIDI Status/Command Register



Bits	*Description (MIDI Status/Command—0x(MIDI Base+1))*
[5:0] (R) res	On reads from this register, these bits do not indicate anything; ignore them.
[6] (R) DSR	Data Set Ready. When read, this bit indicates that you can (0) or cannot (1) write to the MIDI Data register. (Full=1, Empty=0)
[7] (R) DRR	Data Receive Ready When read, this bit indicates that you can (0) or cannot (1) read from the MIDI Data register. (Un-readable=1, Readable=0)
[7:0] (W) CMD [7:0]	MIDI Command Write MIDI commands to this register. Note: The AD1812*only* supports the MIDI 0xFF (reset) and 0x3F (pass-through mode) commands. The controller powers up set for intelligent MIDI mode, but must be put in pass-through mode. To start MIDI operations, send a reset command (0xFF) and then send a pass-through mode command (0x3F). The MIDI data register contains an acknowledge byte (0xFE) after each successful command transfer.

3.7 AD1812 Game Port

The AD1812 contains a Game Port ISA Bus register that correspond to the game port described in the Plug & Play specification. Table 3.11 show the address range for the port.

Table 3.11 Game Port ISA Bus Register (Port)

Register Name	*Address*
Game Port I/O	0x(Game Base) Relocatable in range 0x001 - 0x3FF

3.8 AD1812 Register Summary

The AD1812 SoundPort Controller contains registers that correspond to those found in Sound Blaster Pro, AdLib, MIDI, and Game Port Plug & Play devices. Figure 3.41 shows an overview of the direct and indirect registers in the controller.

This chapter provides detailed information on the vendor specific Plug & Play Powerdown register, all registers in the Windows Sound System codec, and MIDI interface registers. For detailed information on Sound Blaster and AdLib registers, see the *Developer Kit for Sound Blaster Series*, 2nd ed. ©1993, Creative Labs, Inc., 1901 McCarthy Blvd., Milpitas, CA 95035.

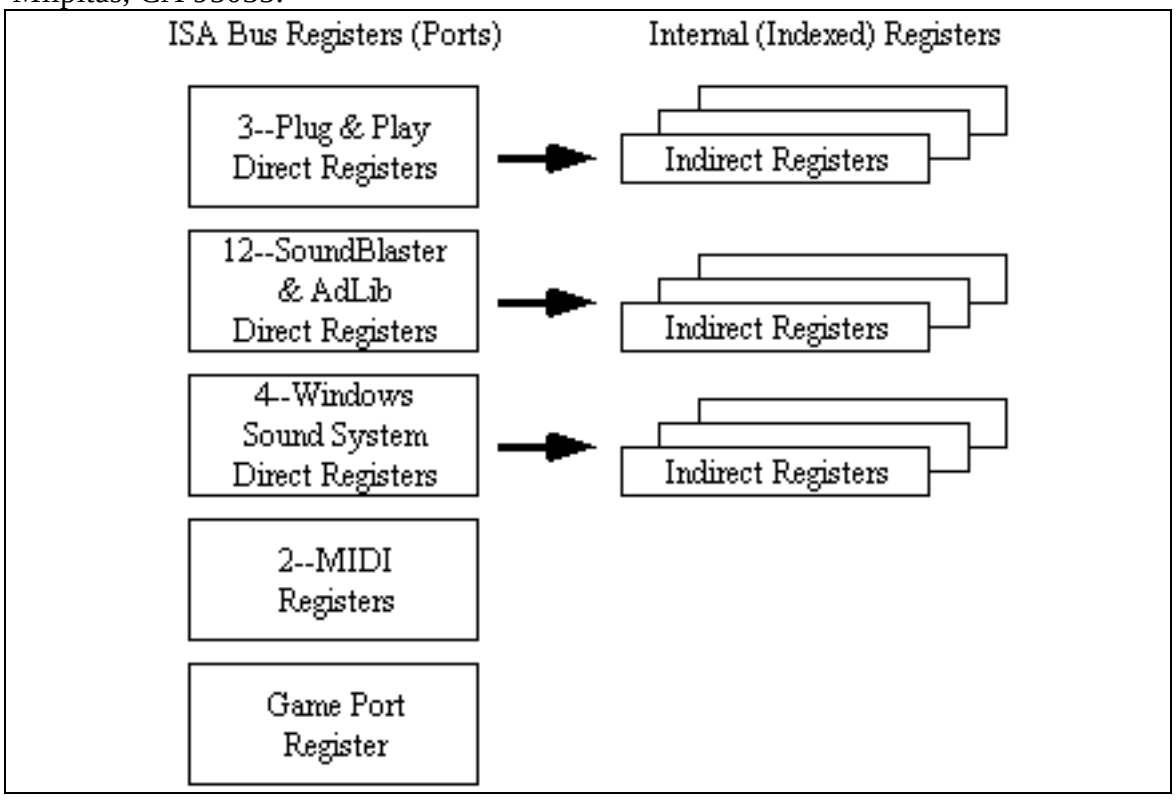


Figure 3.42 Register Overview

Chapter 4 AD1812 Reference Design

4.1 Overview

A reference design using the AD1812 provides a starting point for motherboard and PC plug-in card OEM's beginning their designs using the SoundPort Controller. The AD1812 reference design described in this chapter shows how few components are required for building an AD1812 based PC plug-in card.

The AD1812 evaluation kit (available through your Analog Devices representative) comes with the reference design board described here plus software drivers for DOS, Windows 3.1 and Windows 95 applications. To really get an idea of the AD1812's capabilities, you should participate in a demonstration with the evaluation kit.

The AD1812 reference design board's architecture supports all the controller's features. The board is a 16-bit audio plug-in card for ISA bus personal computers. The I/O features of the card include the following:

- Stereo jacks (four) provide connections for MIC IN, CD-ROM IN (external), LINE IN, and LINE OUT signals.
- Headers provide audio connections for an internal CD-ROM.
- A D-type (DB15) connector lets you connect Joystick /Game and MIDI devices
- A single jumper for selection of Plug & Play or Non-Plug & Play configuration mode, all other board configuration is performed by software.

4.2 Reference Design Architecture

This section provides some detail on the AD1812 reference design board's architecture. For more detailed information, examine the board's schematics in the next section, *Design Information*.

AD1812 SoundPort Controller based designs (plug-in & motherboard) provide software and hardware compatibility with SoundBlaster, AdLib, Windows Sound System, MIDI, and Game Port industry standards. The controller provides these feature through a set of internal integrated device and external I/O ports. Figure 4.1 shows an overview of the reference design board's architecture.

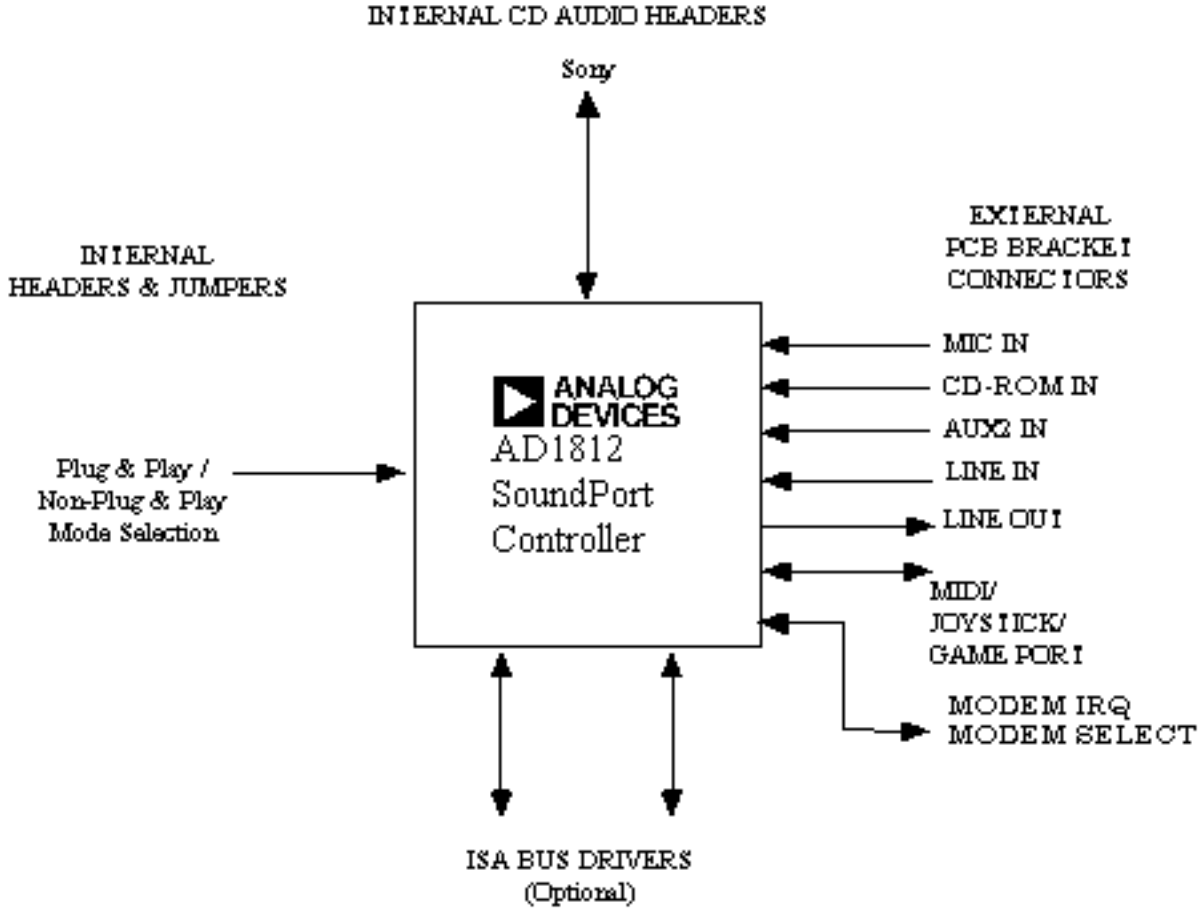


Figure 4.1 SoundPort Reference Design Board I/O

As shown in Figure 4.1 the reference design board's features include the following:

- MICROPHONE INPUT. This input contains an op-amp buffer with a gain of 4 dB and, for microphones with larger signals, you can apply a software controlled 20 dB gain block as well. You can mix this analog input with a digital signal from the ISA bus.

- CD-ROM INPUT. This input has an input impedance of approximately 10 kΩ with a maximum full-scale input level of 2 Vrms. This input jack (and internal CD audio headers) support Panasonic, Mitsumi, and Sony CD-ROMs. You can mix this analog input with a digital signal from the ISA bus.

- LINE INPUT. This input has an input impedance of approximately 10 kΩ with a maximum full-scale input level of 2 Vrms. You can mix this analog input with a digital signal from the ISA bus.

- LINE OUTPUT. This output has a maximum full-scale output level of 2 Vrms. This outputs provide a high-quality line output for use with equipment containing line-level inputs and provides exceptional audio quality when driving speakers designed for the PC (game speakers).

- MIDI/JOYSTICK/GAME PORT. This D-type connector lets you connect a Joystick, other Game Port device, or MIDI devices to your PC.

- Mono Input Header (on-board header JP4). This header connects to the SoundPort Controller's mono input. You can mix this analog input with a digital signal from the ISA bus.

- Plug & Play / Non-Plug & Play Selection Jumper (on-board jumper JP10). This jumper selects Plug & Play mode (jumper not installed) or Non-Plug & Play mode (jumper connects pins 1 & 2). In either mode, the reference design board is completely software configurable with driver software. For more information on the configuration process, see Chapter 2, AD1812 Programming.

Two other SoundPort Controller features that could be useful on your own board are an optional Wave Blaster header and an on-board modem. The reference design would have to be modified if you wanted to include these features.

- Wave Blaster. Adding an external Wave Blaster card lets the SoundPort Controller board use the external wavetable for true instrument quality sounds.

- Modem. The AD1812 can support an external modem chipset. This chipset can provide V.34/DSVD Full Duplex Modem/Speakerphone operation.

4.3 Design Infomation

This section includes a set of schematics for the AD1812 Reference Design Board and component side silk-screen drawing. Complete production files (schematics, pcb layout, & fabrication instructions schematic) are available in Data Exchange Format (DXF) on the Analog Devices, Computer Products Division, Applications Bulletin Board Service (BBS).

To access these files (in the Development Tools section of the BBS), contact the BBS by modem at speeds up to 14,400 baud, no parity, 8 bits data, 1 stop bit by dialing (617) 461-4258. This BBS supports: V.32bis, error correction (V.42 and MNP classes 2, 3, and 4), and data compression (V.42bis and MNP class5).

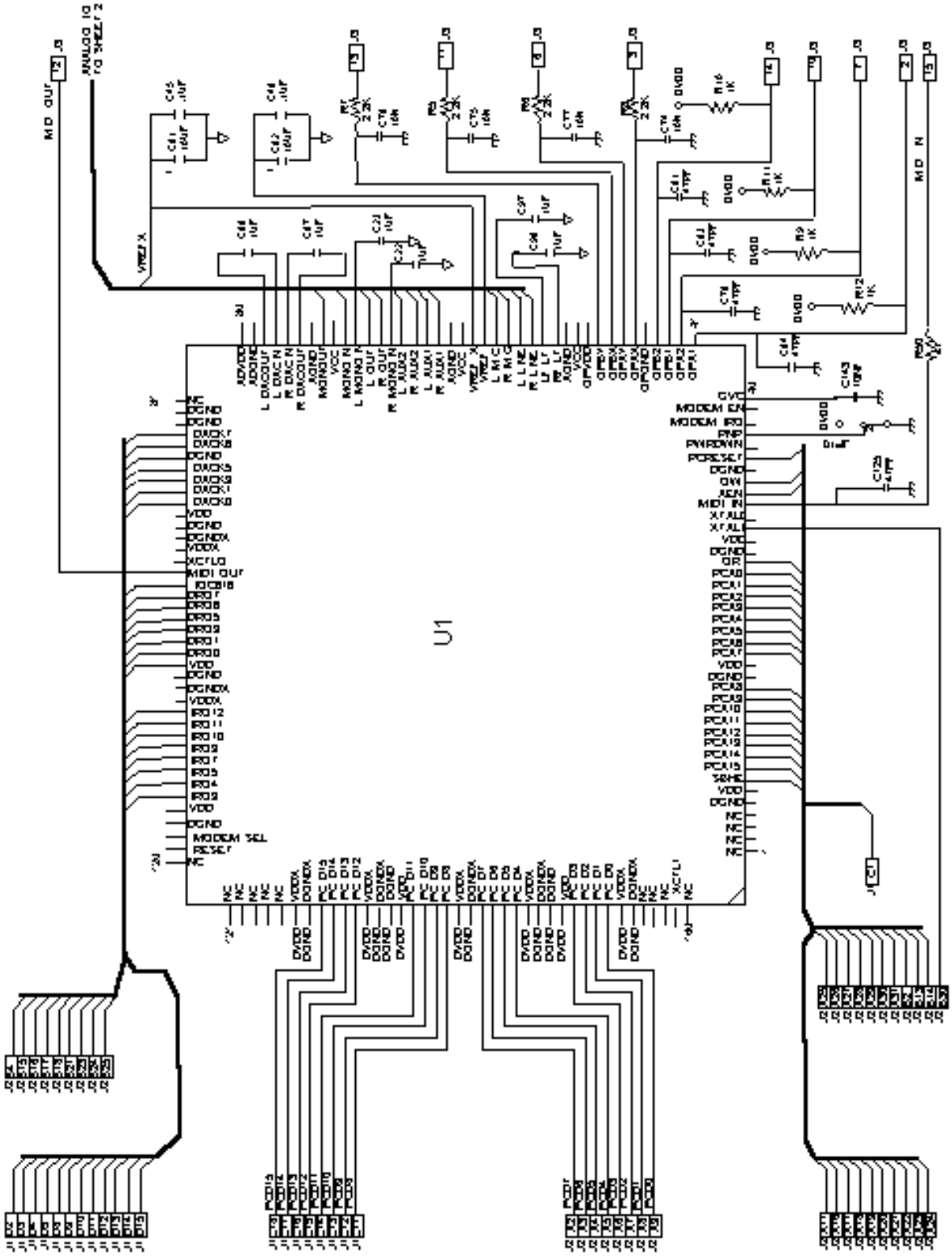


Figure 4.2 AD1812 Reference Design Board, Schematic Page 1 of 4

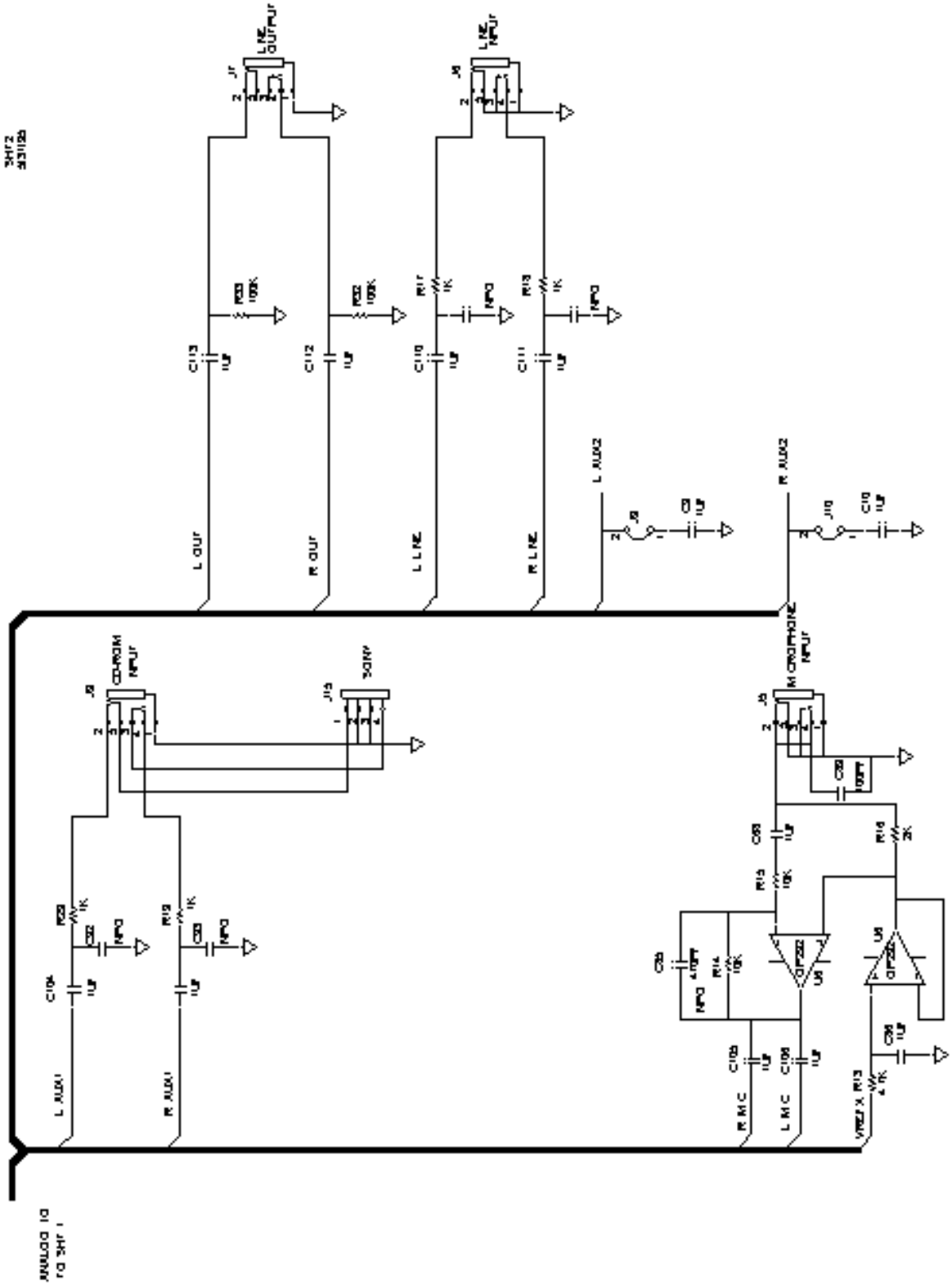


Figure 4.3 AD1812 Reference Design Board, Schematic Page 2 of 4

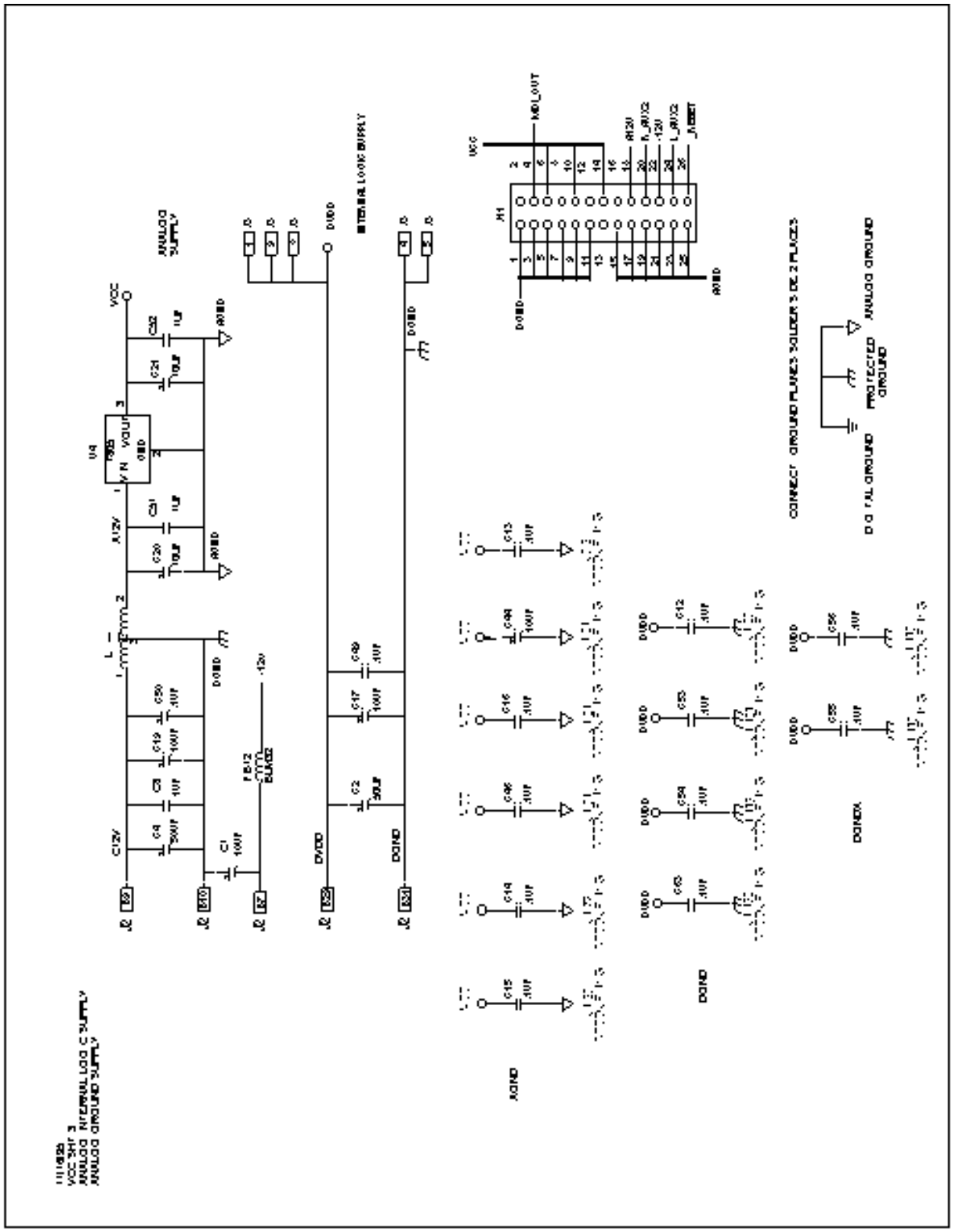


Figure 4.4 1812 Reference Design Board, Schematic Page 3 of 4

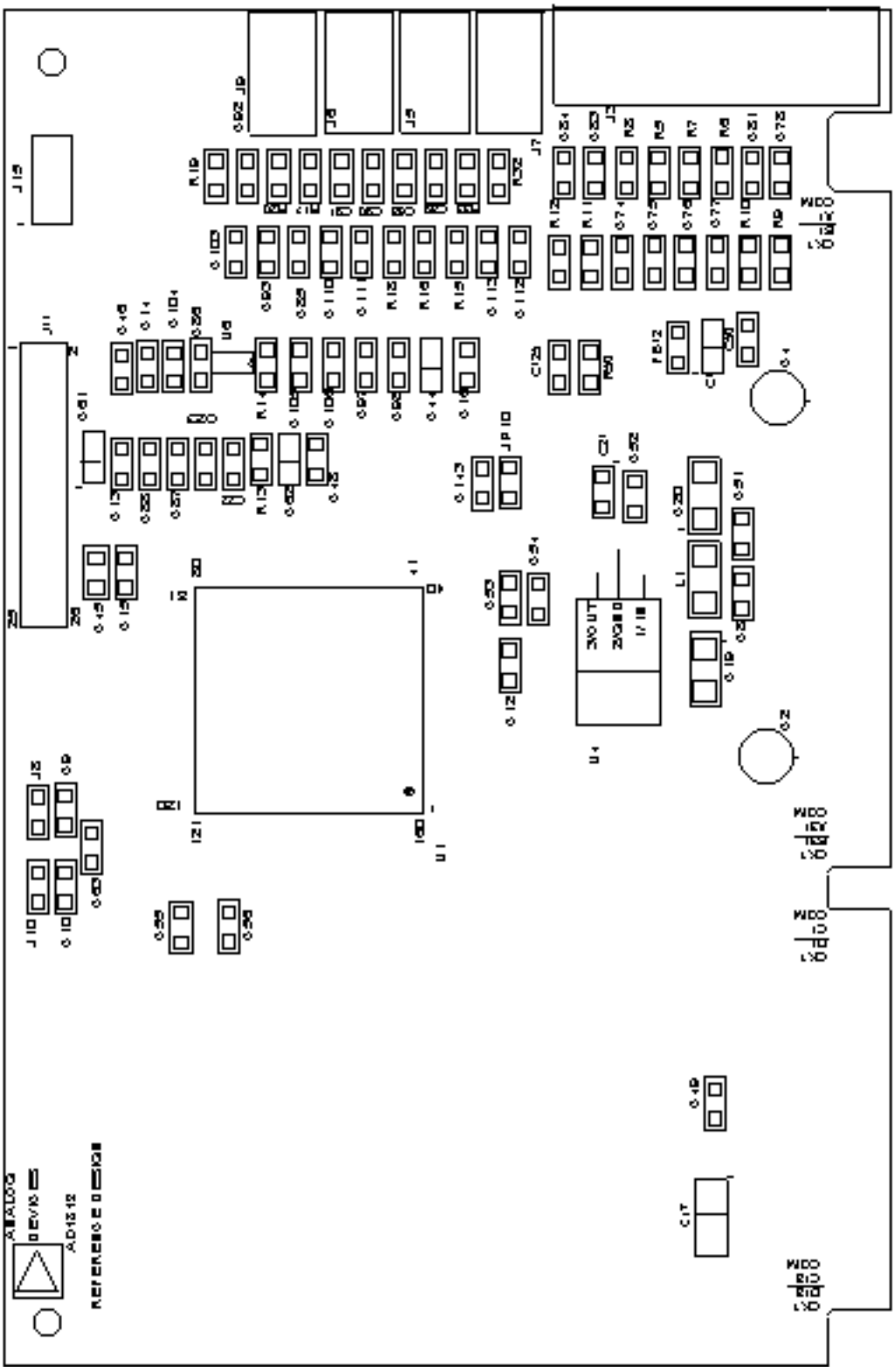


Figure 4.5 AD1812 Reference Design Board, Component Side silk-screen

Index

C

S

T