ESS Technology, Inc.

ES1887
*Audio*Drive® Solution
Preliminary Data Sheet

DESCRIPTION

The ES1887 *Audio***Drive**® solution is a single mixed-signal chip that adds 16-bit stereo sound and FM music synthesis to personal computers. It is equipped with an embedded microcontroller, an OPL3™ superset **ESFM**™ music synthesizer, 16-bit stereo wave CODEC, 16-bit stereo system DAC, 16-bit stereo music DAC, hardware master volume control, MPU-401 UART mode serial port, dual game ports, two serial port interfaces to an external DSP and an external wavetable music synthesizer, DMA control logic with FIFO, and ISA bus interface logic. There are three external stereo inputs (typically line, CD audio, and auxiliary line) and a mono input for a microphone. All of this is embodied in a single chip that can be designed into a motherboard, add-on card, or integrated into other peripheral cards such as voice/fax/modem, VGA, LAN, I/O, and so forth.

The ES1887 *Audio***Drive**® solution can record, compress, and play back voice, sound, and music with two built-in 6-channel symmetric mixer controls. It supports full-duplex operation for simultaneous record and playback using two DMA channels. One of these channels supports bidirectional, 8-bit programmed I/O or DMA data transfers and the other supports 8-bit or 16-bit DMA playback.

The MPU-401 hardware is for interfacing to an external MIDI serial port. The ES1887 music DAC allows the use of an external wave-table synthesizer through the ES1887's third serial port. The PC speaker volume can be controlled by software.

Two software address selection modes allow for BIOS Plug and Play configuration. The dual game port supports two joysticks both having X,Y resistor value settings and two pushbutton switches.

A DSP serial interface allows an external DSP to take over ADC or DAC resources.

Advanced power management features include suspend/ resume from disk or host-independent, self-timed power-down and automatic wake-up.

The ES1887 is backward compatible and pin compatible with the ES1888.

The ES1887 *Audio***Drive**® solution is available in an industry-standard 100-pin Plastic Quad Flat Pack (PQFP) package.

FEATURES

- Single, high-performance, mixed-signal, 16-bit stereo VLSI chip
- Supports enhanced telegaming architecture for Windows™ and DOS game-over-modem
- High-quality, OPL-3 superset **ESFM**™ music synthesizer
- Patented **ESPCM**® compression
- New configurable DMA supports demand transfer and F-type

Record and Playback Features

- Record, compress, and play back voice, sound and music
- 16-bit stereo ADC and DAC
- Programmable sample rates from 4 kHz to 44.1 kHz for record and playback
- Stereo full-duplex operation for simultaneous record and playback
- 2- or 3-button hardware volume control for up, down, and mute
- 3 stereo DACs with independent sample rate and filter control for simultaneous game, music and system playback digital data streams

Inputs/Outputs

- 3 stereo inputs for line-in, CD audio, and auxiliary line-in, and a mono input for microphone
- MPU-401 (UART mode) interface for wavetable synthesizers and MIDI devices
- Integrated dual game port
- Software address mapping, and DMA and IRQ selections for BIOS Plug and Play
- Wavetable serial port interface for ES689/ES69x to access the music DAC
- Serial port interface to external DSP
- PC speaker input/output with volume control

Mixer Features

- 6-channel playback mixer and 6-channel record mixer with stereo inputs for line, CD audio, auxiliary line, music synthesizer, digital audio (wave files), and a mono input for microphone
- Mixer-controlled record and playback with programmable 6-bit (64 step) logarithmic master volume control

Power

- Advanced power management with self-timed power-down, automatic wake-up, and suspend/resume to and from disk
- Supports 3.3 V or 5.0 V operation

Compatibility

- Supports PC games and applications for Sound Blaster™ and Sound Blaster™ Pro
- Microsoft® Windows™ Sound System®
- Microsoft Windows®95 Hardware Design Guide Supplement

Operating Systems

- Microsoft Windows®95
- Microsoft Windows™
- Microsoft Windows for Workgroups™
- Microsoft Windows NT®
- IBM® OS/2® Warp™

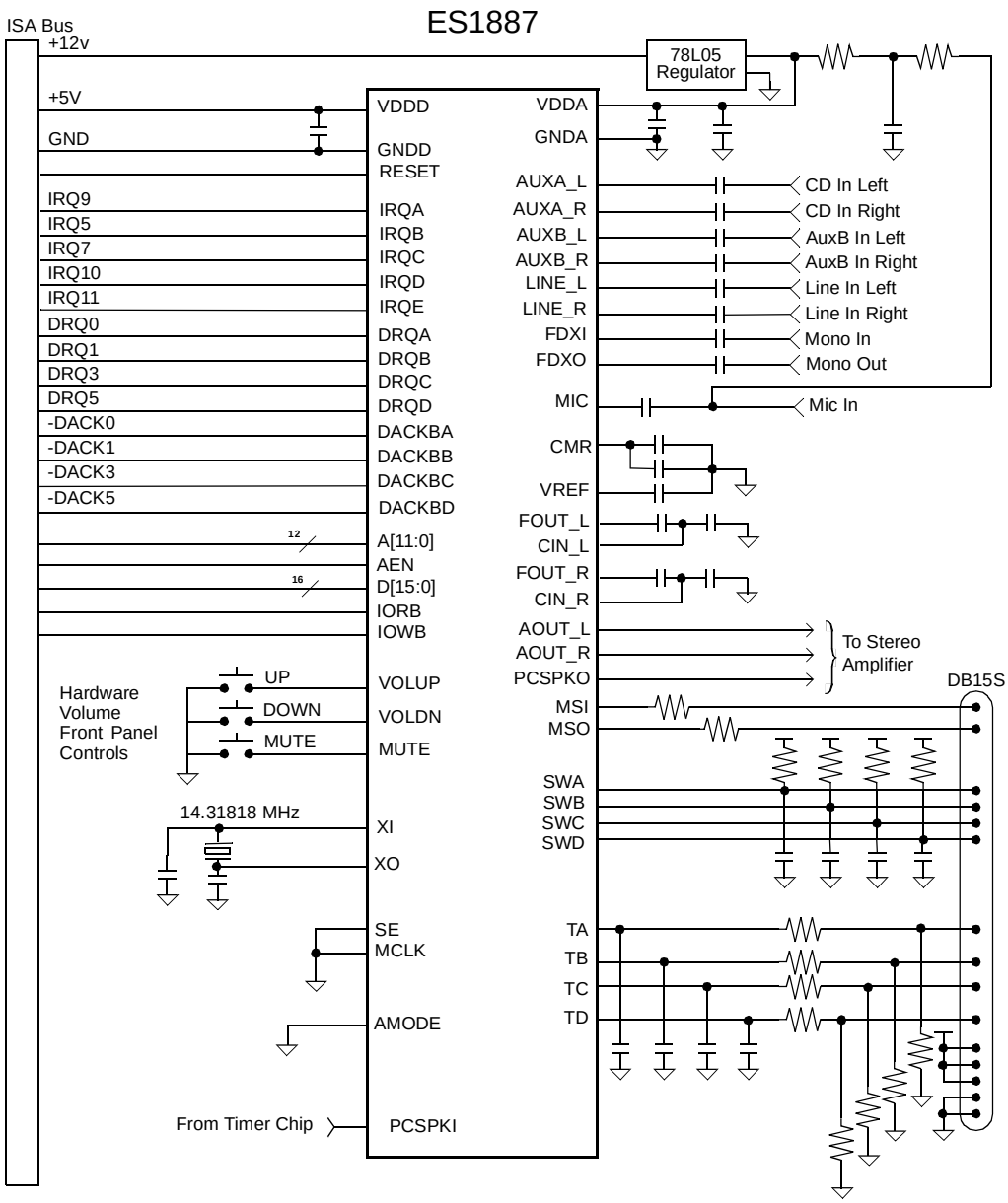


Figure 1 ES1887 Typical Application

CONTENTS

FIGURES

TABLES

PINOUT

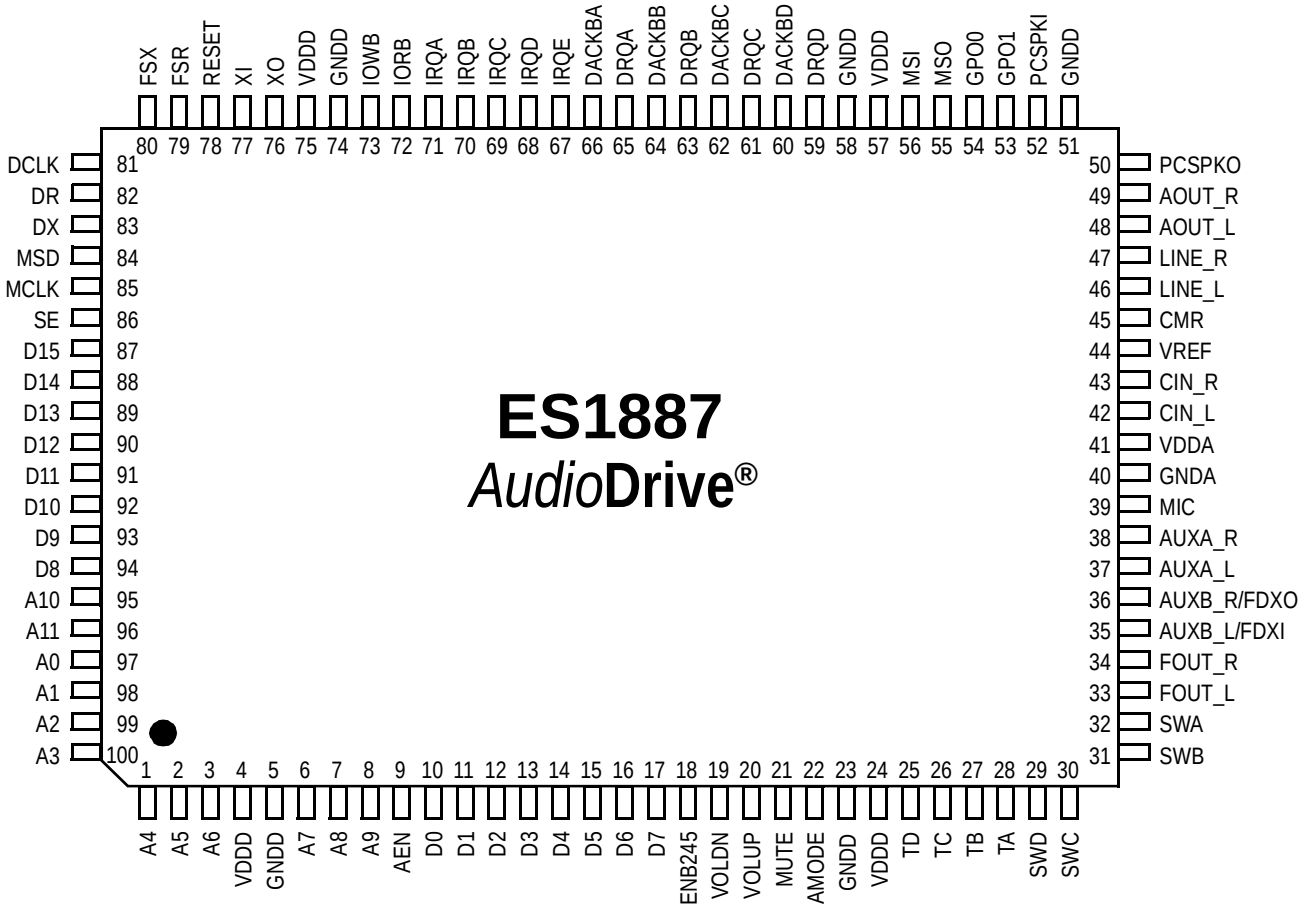


Figure 2 ES1887 Pinout

PIN DESCRIPTION

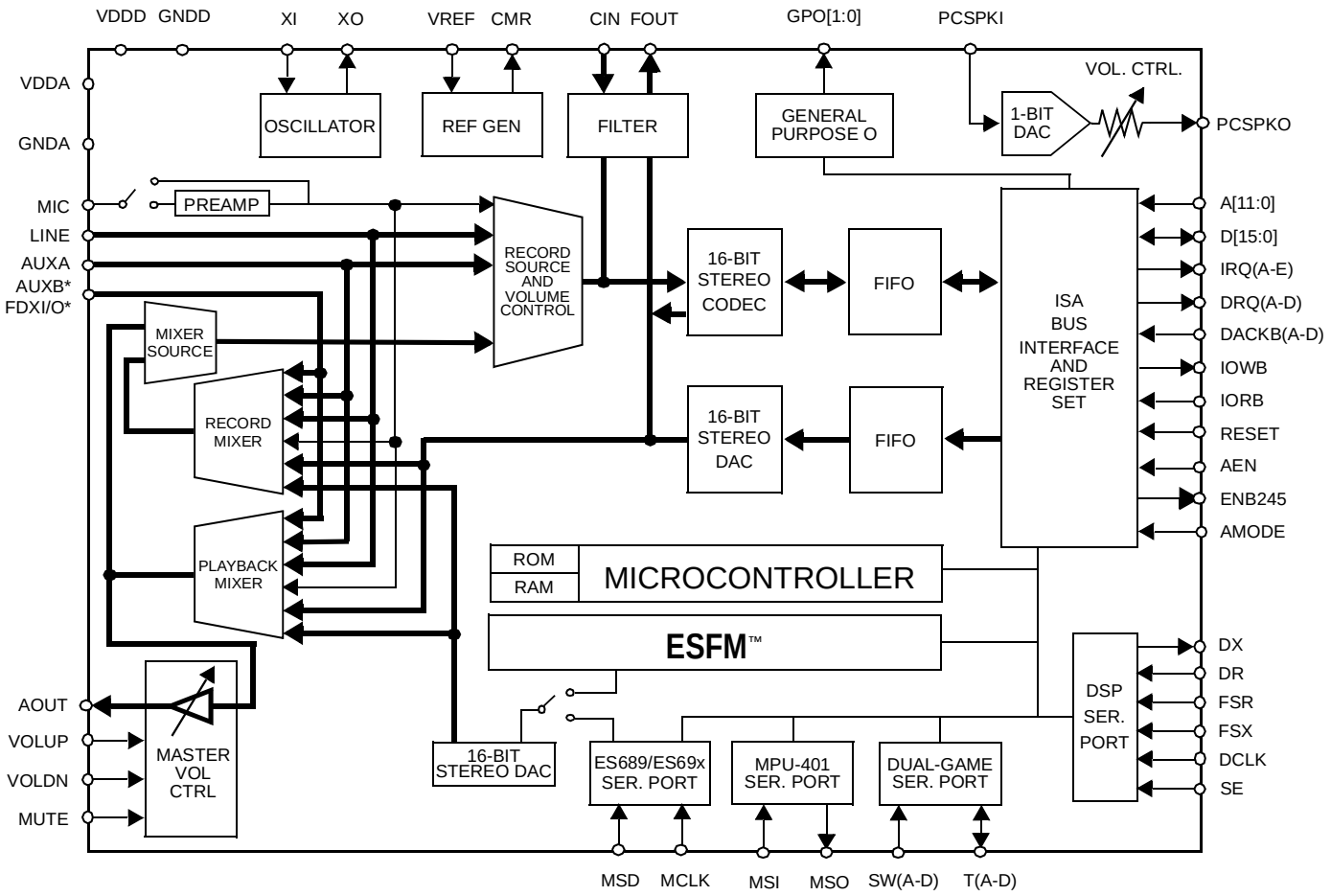
Name	Number	I/O	Definition
A[9:0]	8:6, 3:1, 100:97	I	ISA address bus.
VDDD	4, 24, 57, 75	I	Digital power supply (3.0V - 5.5V).
GNDD	5, 23, 51, 58, 74	I	Digital ground.
AEN	9	I	ISA address valid when active-low, DMA when high.
D[7:0]	17:10	I/O	ISA data bus. 24 mA drivers
ENB245	18	O	Active-low output when ES1887 is being read or written to. Intended to be connected to the enable control of an external 74LS245.
VOLDN	19	I	Active-low volume decrease button input.
VOLUP	20	I	Active-low volume increase button input.
MUTE	21	I	Active-low mute toggle button input.
AMODE	22	I	This pin has an internal pull-down device. The ES1887 is disabled following a hardware reset and must be configured by one of two methods (optioned by AMODE) of software address selection. 0: Read-Sequence-Key method 1: System-Control-Register method
T(A-D)	28:25	I/O	Joystick timer. Uses the digital power supply.
SW(A-D)	32:29	I	Joystick switch input. This pin has internal pull-up to VDDD.
FOUT_L	33	O	Filter output. This pin is normally AC-coupled to CIN_L. The output resistance is about 5k ohm.
FOUT_R	34	O	Filter output. This pin is normally AC-coupled to CIN_R. The output resistance is about 5k ohm.
AUXB_L	35	I	Multipurpose pin, AUXB_L or FDXI. 70k ohm pull-up to CMR.
FDXI		I	Multipurpose pin, AUXB_L or FDXI. When used as FDXI with DSP interface, provides a line level mono input.
AUXB_R	36	I	Multipurpose pin, AUXB_R or FDXO. 70k ohm pull-up to CMR.
FDXO		O	Multipurpose pin, AUXB_R or FDXO. When used as FDXO with DSP interface, is line level mono output, capable of driving a 5k ohm load.
AUXA_L	37	I	Aux A (CD) input left. 70k ohm pull-up to CMR.
AUXA_R	38	I	Aux A (CD) input right. 70k ohm pull-up to CMR.
MIC	39	I	Mic input to +26 dB internal preamp. 70k ohm pull-up to CMR.
GNDA	40	I	Analog ground.
VDDA	41	I	Analog supply voltage (4.5V to 5.5V). Should be greater than or equal to VDDD-0.3V.
CIN_L	42	I	Capacitive coupled input left. The input resistance is about 50k ohm.
CIN_R	43	I	Capacitive coupled input right. The input resistance is about 50k ohm.
VREF	44	I/O	2.25V reference generator.
CMR	45	I/O	2.25V reference buffer output.
LINE_L	46	I	Line input left. 70k ohm pull-up to CMR.
LINE_R	47	I	Line input right. 70k ohm pull-up to CMR.
AOUT_L	48	O	Analog output from master volume. This pin can drive a 10k ohm load.
AOUT_R	49	O	Analog output from master volume. This pin can drive a 10k ohm load.
PCSPKO	50	O	PC speaker analog output.

Name	Number	I/O	Definition
PCSPKI	52	I	PC speaker digital input. This pin has an internal pull-down.
GPO1	53	O	Output that is set low by external reset and thereafter controlled by bit 1 of port Audio_Base+7h. Available to system software for power management or other applications.
GPO0	54	O	Output that is set low by external reset and thereafter controlled by bit 0 of port Audio_Base+7h. Available to system software for power management or other applications.
MSO	55	O	MIDI serial output.
MSI	56	I	MIDI serial input. MSI has an internal pull-up device.
DRQ (A-D)	65, 63, 61, 59	O	ISA active-high DMA request.
DACKB (A-D)	66, 64, 62, 60	I	ISA active-low DMA acknowledge.
IRQ(A-E)	71:67	O	Active-high interrupt request to ISA bus. Unselected IRQ outputs are high impedance. IRQs are software configurable.
IORB	72	I	ISA active-low read strobe.
IOWB	73	I	ISA active-low write strobe.
XO	76	O	Output to external 14.31818 MHz crystal.
XI	77	I	14.31818 MHz clock input, or external crystal.
RESET	78	I	ISA active-high reset.
FSR	79	I	Frame sync for receive data. Software programmable to be active-high or active-low. This pin has a pull-down device.
FSX	80	I	Frame sync for transmit. Software programmable to be active-high or active-low. This pin has a pull-down device.
DCLK	81	I	Serial clock input. This pin has an internal pull-down device.
DR	82	I	Serial data receive. This pin has an internal pull-down device.
DX	83	O	Serial data transmit. Tri-state output. High impedance when not transmitting.
MSD	84	I	Serial data input from ES689/ES69x. This pin has a pull-down.
MCLK	85	I	Serial clock input from ES689/ES69x. This pin has a pull-down.
SE	86	I	Active-high to enable serial mode, i.e., enables an external DSP to control analog resources of the ES1887 through the DSP serial interface. This pin has an internal pull-down.
D[15:8]	87:94	I	ISA high byte input data bus. Used for system DAC when 16-bit DMA transfer mode is selected.
A[11:10]	96:95	I	ISA address bus. The ES1887 requires these pins to be low for all address decodes.

FUNCTIONAL DESCRIPTION

This section shows the overall structure of the ES1887 and discusses its major functional subunits.

The major subunits of the ES1887 are shown in Figure 3 and described briefly in the following paragraphs.



* Some of these pins are shared with other functions.

Figure 3 ES1887 Block Diagram

Digital Subsystems

- **RISC microcontroller** – game-compatible audio functions are performed by an embedded microcontroller.
- **Oscillator** – circuitry to support an external crystal.
- **ROM and RAM** – firmware ROM and data RAM to the embedded microcontroller.
- **MPU-401 serial port** – asynchronous serial port for MIDI devices such as a wavetable synthesizer or a music keyboard input.
- **FIFO** – RAM for a 256-byte FIFO data buffer for use with the first audio channel and RAM for a 64-byte FIFO data buffer for use with the second audio channel.
- **ISA bus interface** – provides interface to ISA bus address, data, and control signals.
- **Dual game port** – integrated dual game port for two joysticks.

- **Wavetable serial port** – serial port connection from the output of an ES689 or ES69x that eliminates the requirements for an external DAC.
- **DSP serial port** – interface to an optional external DSP for control of the CODEC.
- **ESFM music synthesizer** – high-quality, OPL3 superset FM synthesizer with 20 voices.
- **Hardware volume control** – 3-pushbutton inputs with internal pull-up devices for up/down/mute that can be used to adjust the master volume control.

  A software-selectable option allows the mute input to be omitted. The mute input is defined as the state when both up and down inputs are low. By default, this feature is disabled.

Analog Subsystems

- **Record and Playback Mixers**– six input stereo mixers. Each input has independent left and right 4-bit volume control:
  - Line In
  - Mic In
  - Aux A (CD-audio)
  - Aux B (or FDXI)
  - Digitized audio (wave files)
  - FM/ES689/ES69x music DAC
- **16-Bit stereo CODEC** – for audio record and playback of the first audio channel.
- **16-Bit stereo system DAC** – for audio playback of the second audio channel.
- **16-Bit stereo music DAC** – for **ESFM**™ or external wavetable synthesizer.
- **1-Bit DAC** – for PC speaker digital input.
- **Record source and input volume control** – input source and volume control for recording. The recording source can be selected from one of four choices:
  - Line In
  - Mic In
  - Aux A (CD-audio)
  - Mixer (playback or record)
- **Mixer source** – determines which mixer is used for the record source, either the playback or record mixer. Mixer register 7Ah bits 4:3 selects the mixer source.
- **Output volume and mute control** – The master volume is controlled either via programmed I/O or via volume control switch inputs. The master volume supports 6 bits per channel plus mute.
- **Reference generator** – analog reference voltage generator.
- **PC speaker volume control** – The PC speaker is supported with a 1-bit DAC with volume control. The analog output pin PCSPKO is intended to be externally mixed at the external amplifier.
- **Filter** – switched capacitor low-pass filter.
- **General purpose out** – outputs available to system software for power management or other applications.
- **Pre-amp** – 26 dB microphone pre-amplifier.

MIXER SCHEMATIC BLOCK DIAGRAM

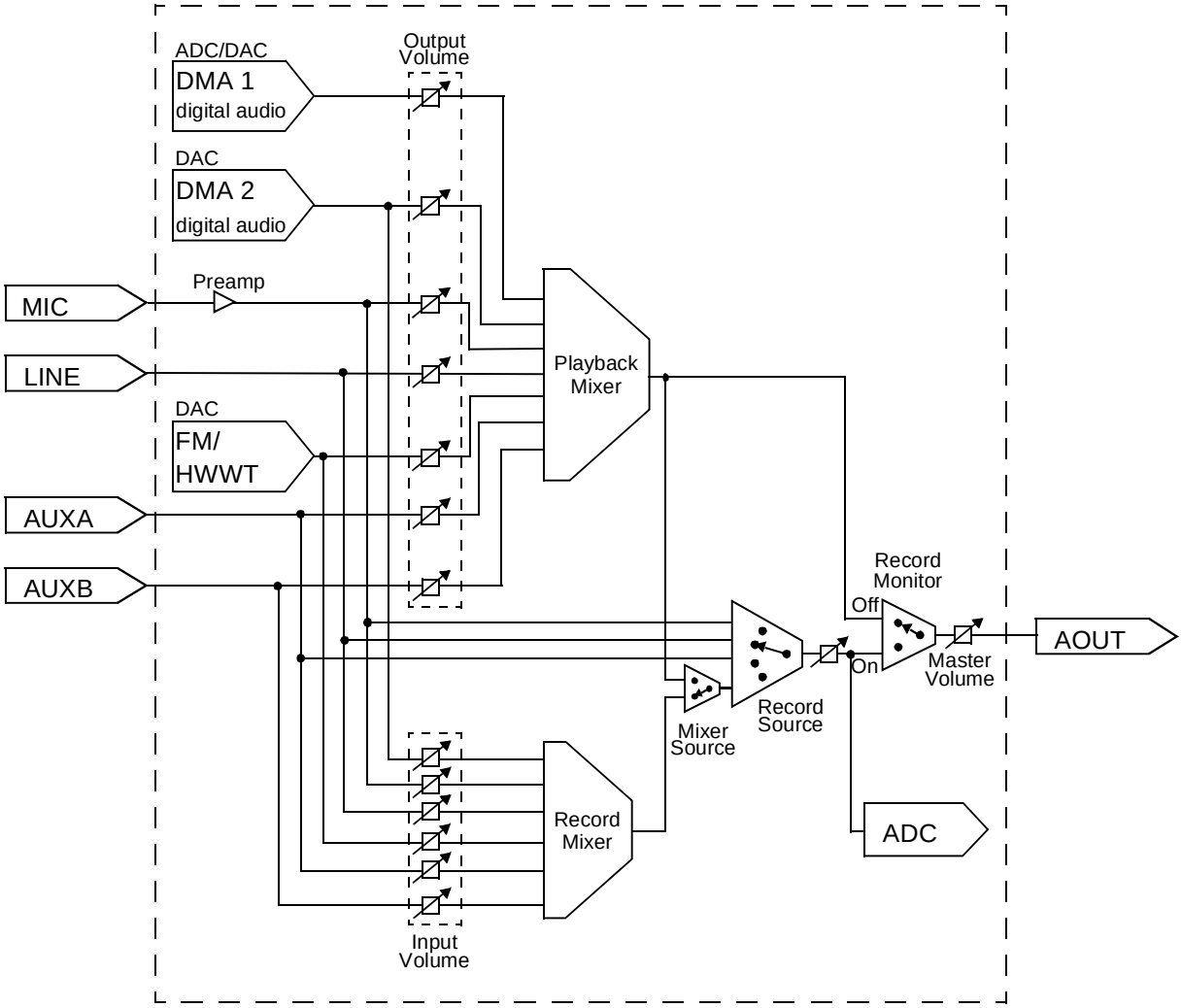


Figure 4 ES1887 Mixer Schematic Block Diagram

BUS INTERFACING

Table 1 shows the pins used to interface the ES1887 to the ISA bus.

Table 1 ES1887 ISA bus interface

Pin	I/O	Description
A[11:0]	I	ISA address bus.
AEN	I	ISA address valid when active-low, DMA when high.
D[15:8]	I	ISA data bus. Used for system DAC in 16-bit transfer mode.
D[7:0]	I/O	ISA data bus. 24 mA drivers.
IOWB	I	ISA active-low write strobe.
IORB	I	ISA active-low read strobe.
IRQ(A-E)	O/Hi Z	ISA interrupt request. 16 mA driver.
DACKB(A-D)	I	ISA active-low DMA acknowledge.
DRQ(A-D)	O/Hi Z	ISA active-high DMA request.
RESET	I	ISA active-high reset.

DIGITAL AUDIO

The ES1887 incorporates two digital audio channels.

Audio 1 The first audio channel. This channel is used for Sound Blaster Pro compatible DMA, Extended mode DMA, and programmed I/O. It can be used for either record or playback. This channel can be mapped to any of the three 8-bit ISA DMA channels: 0, 1, or 3.

Audio 2 The second audio channel. This channel is used for audio playback in full-duplex mode. This channel can be mapped to any of the three 8-bit ISA DMA channels or the 16-bit channel.

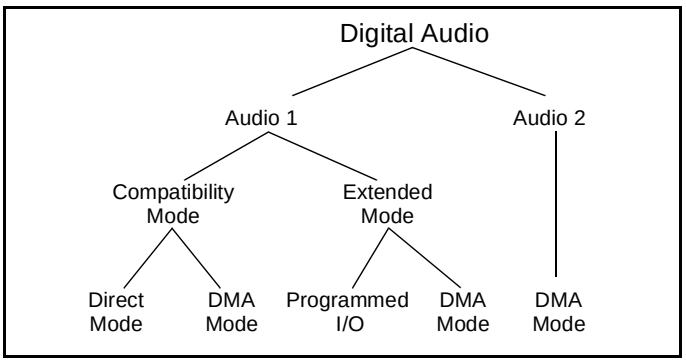


Figure 5 Data Transfer Modes

Programming Data transfers

Programming Data transfers can be quite complicated with the ES1887. Both Compatibility mode and Extended mode offer a variety of modes for conducting transfers. The commands to enable the different transfers varies depending on which audio channel is used and which mode (Compatibility or Extended) is used.

The biggest difference in available data transfer modes is between audio channel 1 and audio channel 2. This is illustrated in Figure 5. Audio 2 allows only for DMA mode. Audio 1 allows for Direct mode and DMA mode when using Compatibility mode. Audio 1 allows for Programmed I/O and DMA mode when using Extended mode.

Data Formats

See “Data Formats” on page 29.

Data Transfers in Compatibility Mode.

The first audio channel can be programmed using the standard Sound Blaster compatible commands. These commands are written to the chip through port Audio_Base+Ch.

When programming the first audio channel for transfers one of the following modes can be used:

- Direct Mode
- DMA Modes
  - Normal
  - Auto-initialize

In addition, both DMA normal mode and DMA auto-initialize mode can use a special high-speed mode.

Direct Mode

In direct mode, the timing for DMA transfers is handled by the application program. For example, the system timer can be reprogrammed to generate interrupts at the desired sample rate. At each system timer interrupt, the command 10h, 11h, 20h, or 21h is issued followed by the sample. Polling of the write buffer available flag (Audio_Base+Ch[bit 7]) is required before writing the command and between the command and the data.

NOTE: The switched capacitor filter is initialized by reset for an intended sample rate of 8 kHz. In direct mode, the application may wish to adjust this filter appropriate to the actual sample rate. The easiest way to do this is to program the timer with command 40h just as if the application were using DMA mode.

DMA Modes

In DMA mode, the programmable timer in the ES1887 controls the rate at which samples are sent to the CODEC. The timer is programmed using command 40h, which also sets up the programmable filters inside the ES1887. The ES1887 firmware maintains an internal FIFO (32 levels for 16-bit transfers, 64 levels for 8-bit transfers) that is filled by DMA transfers and emptied by the timed transfers to the DAC.

Before a DMA transfer, the application first programs the DMA controller for the desired transfer size and address, then programs the ES1887 with the same size information. At the end of the transfer, the ES1887 generates an interrupt request, indicating that the current block transfer is complete. The FIFO gives the application program sufficient time to respond to the interrupt and initiate the next block transfer.

The ES1887 supports both normal DMA mode and auto-initialize DMA mode.

Normal DMA Mode

In normal mode DMA transfers, the DMA controller must be initialized and the ES1887 must be commanded for every block that is transferred.

Auto-Initialize DMA Mode

In auto-initialize mode, the DMA transfer is continuous, in a circular buffer, and the ES1887 generates an interrupt for the transition between buffer halves. In this mode the DMA controller and ES1887 need to be set up only once.

High-Speed Mode

The ES1887 supports mono 8-bit DMA transfers at a rate up to 44 kHz. Mono 16-bit transfers are supported up to a rate of 22 kHz.

There is a special “high-speed mode” that allows 8-bit sampling up to 44 kHz for ADC. This mode uses commands 98h (“auto-initialization”) and 99h (“normal”). No automatic gain control (AGC) is performed. The input volume is controlled with command DDh.

Data Transfers in Extended Mode

The first audio channel is programmed using the controller registers internal to the ES1887. The commands written to the controller registers are written to the chip through port Audio_Base+Ch.

When programming the first audio channel for transfers, one of the following modes can be used:

- Programmed I/O
- DMA Modes
  - Normal (Single or Demand transfer)
  - Auto-Initialize (Single or Demand transfer)

In addition, both DMA normal mode and DMA auto-initialize mode use Single transfer or Demand transfer modes.

Programmed I/O

For some applications, DMA mode is not suitable or available for data transfer, and it is not possible to take exclusive control of the system for DAC and ADC transfers. In these situations, use I/O block transfers within an interrupt handler. The REP OUTSB instruction of the 80x86 family transfers data from memory to an I/O port specified by the DX register. The REP INSB instruction is the complementary function. Use ES1887 port Audio_Base+Fh for block transfers.

I/O transfers to FIFO are nearly identical to the DMA process, except that an I/O access to port Audio_Base+Fh replaces the DMA cycle. For details about programmed I/O operation see “Extended Mode Programmed I/O Operation” on page 37.

DMA Modes

Extended mode DMA supports both Normal and Auto-Initialize mode. In addition, Normal mode and Auto-Initialize mode both support Single and Demand transfer modes.

Using Single transfer, one byte is transferred per DMA request. Demand transfer reduces the number of DMA requests necessary to make a transfer by allowing two or four bytes to be transferred per DMA request. Thus there are multiple DMA acknowledges for each DMA request.

For a description of DMA mode including Normal DMA mode and Auto-Initialize DMA mode see “DMA Modes” on page 12.

Extended Mode Audio 1 Controller Registers

The following registers control operation of the first audio channel in Extended mode:

Address	Name
A1h	Audio 1 Sample Rate Generator register
A2h	Audio 1 Filter Clock Divider register
A4h	Audio 1 Transfer Count Reload register – low byte
A5h	Audio 1 Transfer Count Reload register – high byte
B1h	Legacy Audio Interrupt Control register
B2h	Audio 1 DRQ Control register
B4h	Input Volume Control register
B5h	Audio 1 DAC Direct Access register – low byte
B6h	Audio 1 DAC Direct Access register – high byte
B7h	Audio 1 Control 1 register
B8h	Audio 1 Control 2 register
B9h	Audio 1 Transfer Type register

Data Transfers Using the Second Audio Channel

The second audio channel is programmed using mixer registers 70h through 7Dh. The commands written to the mixer registers are written to the chip through ports Audio_Base+4h and Audio_Base+5h.

DMA mode is used when programming the second audio channel for transfers:

DMA Modes:

- Normal (Single or Demand transfer)
- Auto-Initialize (Single or Demand transfer)

In addition, both DMA normal mode and DMA auto-initialize mode use Single transfer or Demand transfer modes.

DMA Modes

DMA under the second audio channel supports both Normal and Auto-Initialize mode. In addition, Normal mode and Auto-Initialize mode both support Single and Demand transfer modes.

Using Single transfer, one byte is transferred per DMA request. Demand transfer reduces the number of DMA requests necessary to make a transfer by allowing two, four, or eight bytes to be transferred per DMA request. Thus there are multiple DMA acknowledges for each DMA request.

For a description of DMA mode including Normal DMA mode and Auto-Initialize DMA mode see “DMA Modes” on page 12.

Audio 2 Related Mixer Registers

The following registers control DMA operations for the second audio channel:

Address	Name
70h	Audio 2 Sample Rate Generator register
72h	Audio 2 Filter Clock Divider register
74h	Audio 2 Transfer Count Reload register – low byte
76h	Audio 2 Transfer Count Reload register – high byte
78h	Audio 2 Control 1 register
7Ah	Audio 2 Control 2 register
7Ch	Audio 2 DAC Volume Control register
7Dh	Audio 2 Configuration register

First Audio Channel “CODEC”

The CODEC of the first audio channel is not a true stereo CODEC, in that it cannot perform stereo DAC and ADC simultaneously. The first audio channel CODEC can be either a stereo DAC, a stereo ADC, or a mono CODEC. After reset, the CODEC is set up for DAC operations. Any ADC command causes a switch to the ADC “direction” and any subsequent DAC command switches the converter back to the DAC “direction.”

The DAC output is filtered and input to the mixer. After reset, input to the mixer from the first audio channel DAC is muted. This is to prevent pops. The ES1887 maintains a status flag to determine if the input to the mixer from the first audio channel DAC is enabled or disabled. The command D8h returns the status of the flag (00h = disabled and FFh = enabled). Use command D1h to enable input to the mixer from the first audio channel DAC and command D3h to disable the input.

To play a new sound without resetting beforehand, when the status of the analog circuits is not clear, mute the input to the mixer with command D3h, then set up DAC direction and level using the direct-to-DAC command:

10h + 80h

Wait 25 msec for the analog circuitry to settle before enabling the input to the mixer with command D1h.

A pop may be heard if the DAC level was left at a value other than mid-level (code 80h on an 8-bit scale) by the previous play operation. To prevent this, always finish a DAC transfer with a command to set the DAC level to mid-range:

10h + 80h

INTERRUPTS

There are four interrupt sources in the ES1887:

Table 2 ES1887 Interrupt sources

Interrupt Source	Description
Audio 1	This interrupt is used for the first audio channel (Sound Blaster-compatible DMA, Extended mode DMA, and Extended mode programmed I/O), as well as SB-compatible MIDI receive. This interrupt request is cleared by a hardware or software reset, or by an I/O read from Audio_Base+0Eh. The interrupt request can be polled by reading from Audio_Base+0Ch.
Audio2	This interrupt is used for the second audio channel. For details on masking interrupt sources see “Assigning Interrupt Sources” below. Audio 2 can be polled and cleared by reading or writing bit 7 of Mixer register 7Ah.
Hardware Volume	This interrupt occurs when one of the three hardware volume controls generates an event. Bit 1 of Mixer register 64h is the mask bit for this interrupt. The interrupt request can be polled by reading bit 4 of the same register. The interrupt request is cleared by writing any value to register 66h. Typically this interrupt, if used, is shared with an audio interrupt.
MPU-401	This interrupt occurs when a MIDI byte is received. It will go low when a byte is read from the MIDI FIFO and go high again quickly if there are additional bytes in the FIFO. The interrupt status is the same as the read data available status flag in the MPU-401 Status register. This interrupt is masked by bits 7:5 of Mixer register 40h.

As described in Table 2, each interrupt source has a register where the interrupt can be polled. As an alternative, all the interrupt source can be polled from register 7Fh (Interrupt Control and Status register). Bits 7:4 can be used to poll the interrupt sources as follows:

Bit 7 MPU-401
Bit 6 Hardware Volume
Bit 5 Audio 2
Bit 4 Audio 1

In addition register 7Fh is used to select the interrupt mode and IRQ channel.

Assigning Interrupt Sources

There are two ways of mapping or masking interrupt sources to the five interrupt output pins, IRQ(A–E). The first method is the same method that is used with the ES1888. The second method has been created to replace the ES1888 method. The ES1888 method has been retained to maintain backward compatibility. Any of bits 3:1 of Mixer register 7Fh will enable the new method (with the exception of the binary values 110 and 111 which are reserved).

Each interrupt pin can be in either an active or high-impedance state.

ES1888 Interrupt Method

To maintain backward compatibility to the ES1888, interrupts can be assigned as they are in the ES1888.

Table 3 ES1888 Interrupt Assignment Method

Audio 1	**Controller Register B1**	
	Map	Bits 3:2 select among IRQ(A-D)
	Output Enable	Bit 4
	Mask	Bits 6:5
Audio 2	**Mixer Register 7A**	
	Map	Fixed to IRQE
	Output Enable	Bit 4
	Mask	Bit 6
Hardware Volume	**Mixer Register 64h**	
	Map	Bit 2 selects IRQE Bit 1 share interrupt with Audio 1
	Output Enable	Bits 2 or 1enable output
	Mask	Bits 2 or 1 mask Hardware Volume
MPU-401	**Mixer Register 40h**	
	Map	Bits 7:5 select among IRQ(A-E)
	Output Enable	Bits 7:5
	Mask	Bits 7:5

New Interrupt Method

In the new interrupt method, all four sources must share a single IRQ output, IRQ(A–E), as determined by bits 3:1 of Mixer register 7Fh. Edge-generation logic ensures that each pending interrupt request will correctly signal the interrupt controller. Bit 0 of the same register is the single IRQ output enable.

The new interrupt method is enabled if any of bits 3:1 of Mixer register 7Fh are high. If all three bits are low, then the ES1887 acts exactly as the ES1888 described previously.

Mixer register 7Fh is reset to zero by hardware or software reset.

In the new interrupt method, each interrupt source has one or more mask bits:

Table 4 ES1887 Interrupt Method

Audio 1	**Controller Register B1**	
	Bit 6	Masks the interrupt for Extended mode DMA transfer complete.
	Bit 5	Masks the interrupt for Extended mode FIFO half-empty flag transition.
	Bits 3:2	These bits do not select the interrupt pin in the new interrupt method. They should be programmed to match the pin selected by mixer register 7Fh if possible. Some programs might read the B1h register to determine the interrupt number.
Audio 2	**Mixer Register 7A**	
	Bit 6	Masks this interrupt source.
Hardware Volume	**Mixer Register 64h**	
	Bit 1	Must be set high (share with audio) to enable this interrupt source.
MPU-401	**Mixer Register 40h**	
	Bits 7:5	Must be set to 010 (share with audio) to enable this interrupt source.

PERIPHERAL INTERFACING

DSP Interface

The ES1887 contains a synchronous serial interface for connection to a DSP serial interface. The typical application for this interface is a speakerphone.

Table 5 DSP Interface Pins

Pins	Descriptions
DCLK	Data clock input. The rate can vary, but a typical value is 2.048 MHz (8 kHz x 256). Input with pull-down.
DX	Data transmit. Active output when data is being transmitted serially from the ES1887; otherwise high impedance. Tri-state output.
DR	Serial data input with pull-down.
FSX	Frame sync input for transmit. Software programmable to be active-high or active-low. Input with pull-down.
FSR	Frame sync input for receive data. Software programmable to be active-high or active-low. Input with pull-down.

DSP Operating Modes

There are two DSP data transfer modes for the ES1887. The state of a single switch internal to the ES1887 determines which mode is enabled. This switch can route the first audio channel to the second audio channel DAC. When the first audio channel is routed to the second audio channel DAC, Telegaming mode is enabled. Otherwise the DSP is operating in its default mode.

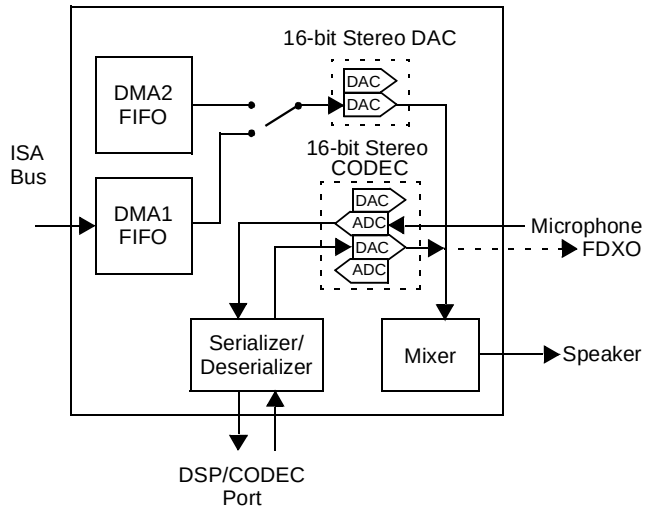


Figure 6 Telegaming Mode

Telegaming Mode

This mode is enabled when two conditions are present:

1. The DSP serial port must be enabled (i.e., either bit 7 of Mixer register 48h is high or the input pin SE is high).
2. Bit 1 of Mixer register 48h is high. This bit enables Telegaming mode.

In previous chips, when the DSP serial port is enabled, the Audio 1 CODEC is unavailable for use by the first audio channel. This means digital audio for Sound Blaster Pro compatible games is muted. Sound Blaster can use only the first audio channel for digital audio. The Audio 1 CODEC is used by the DSP.

In Telegaming mode, the first audio channel can be switched over to the Audio 2 DAC. Internally, the first audio channel is routed to the second audio channel DAC and the second audio channel has no function. In addition, the second audio channel mixer volume control is slaved to the first channel mixer volume control.

Default Mode

The default mode operates just as telegaming mode except that data from the first audio channel cannot be heard. Data sent through the second audio channel can be mixed as in Telegaming mode.

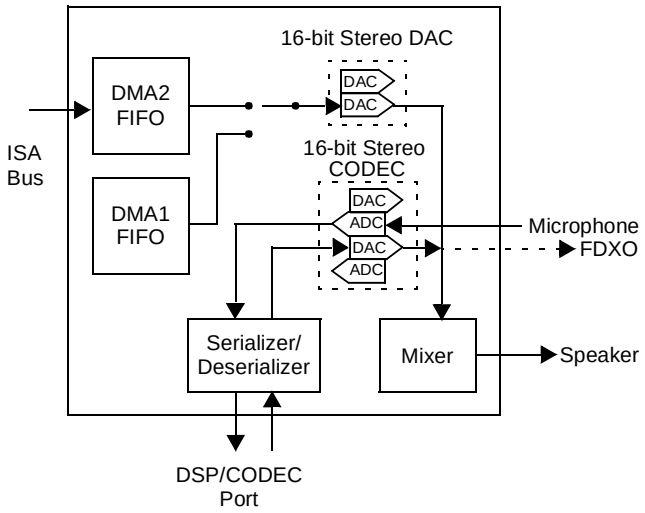


Figure 7 Default Mode

No Acoustic Echo Cancellation

The DSP cannot perform acoustic echo cancellation in either mode. Because the audio from the host does not pass directly through the DSP, there is no way for the DSP to compensate for acoustic echo. Therefore, using a headset for either the microphone or speakers or both is recommended.

DSP Digital Audio Playback

There are two choices for mixing the DSP digital audio playback data with other audio sources. The audio data can be mixed in the ES1887's internal playback mixer or external to the ES1887.

Mixing Internal to the ES1887

The DSP digital audio playback can be mixed within the ES1887 playback mixer. To select this method, set the Output Signal Control bits of the Mixer register 44h for mixer output. To do this, program bits 6:4 of mixer register 44h to 1, 0, and 0, respectively. The volume of the DSP digital audio playback is controlled by the DAC Play Volume Register (14h).

NOTE: In Telegaming mode, register 14h also controls the game-compatible first audio channel digital audio playback. If independent mixer volume control of the game-compatible and DSP digital audio data is necessary, use the second method.

Mixing External to the ES1887

The second method is to use the FDXO output pin and mix the DSP digital audio playback and the game-compatible digital audio playback in an external audio mixer. To select this method, set the Output Signal Control bits of Mixer register 44h for mixer output except DAC playback. To do this, program bits 6:4 of register 44h to 1, 0, and 1, respectively. In addition, set bit 1 of Mixer register 46h high to enable FDXO as an output when DSP serial mode is enabled.

The volume of the DSP digital audio playback is controlled within the DSP by scaling the data.

Table 6 Digital audio mixing methods in Serial mode

To mix DSP and digital audio...	...set the Output Signal Control bits of Register 44h to...			...and the volume of the DSP digital audio playback is controlled by...	In addition...
	Bit 6	Bit 5	Bit 4		
internal to the ES1887	1	0	0	DAC Play Volume Register 14h	N/A
external to the ES1887	1	0	1	scaling the data within the DSP	Set bit 1 of register 46h high.

DSP Interface Serial Data Format

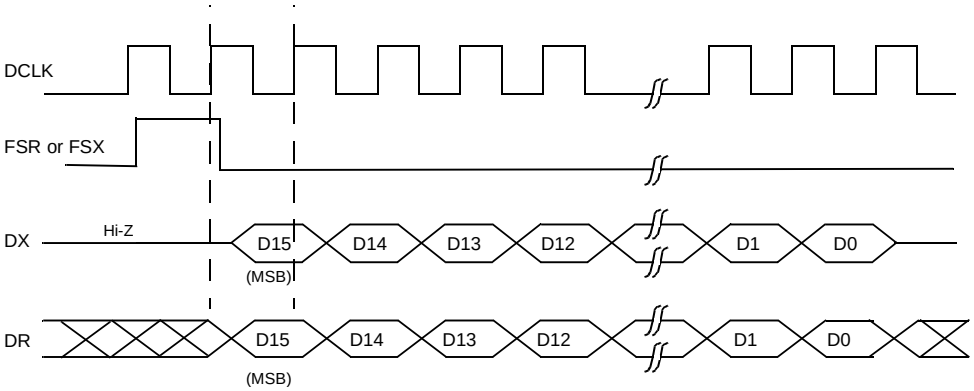


Figure 8 16-Bit Data, Positive Sync Pulse

Wavetable Interface

The ES1887 contains a synchronous serial interface for connection to an ES689/ES69x wavetable music synthesizer.

Table 7 Wavetable Interface Pins

Pins	Descriptions
MCLK	Serial clock from external ES689/ES69x music synthesizer (2.75 MHz). Input with pull-down.
MSD	Serial data from external ES689/ES69x music synthesizer. When both MCLK and MSD are active, the stereo DAC normally used by the FM synthesizer is acquired for use by the external ES689/ES69x. The normal FM output is blocked. Input with pull-down.

MPU-401 MIDI Interface

There are two separate MIDI interfaces in the ES1887. The Sound Blaster compatible command set and an MPU-401 “UART Mode” compatible serial port. MPU-401 is a superior method of MIDI serial I/O because it does not interfere with DAC or ADC Sound Blaster commands. Both methods of serial I/O share the same MSI and MSO pins. The ES1887 MPU-401 MIDI interface consists of a 23-byte receive FIFO and an 8-byte transmit FIFO.

By default after hardware reset, the MPU-401 interface is disabled. It must be configured using Mixer register 40h.

MPU-401 requires an interrupt channel for MIDI receive. This interrupt should be selected using Mixer register 40h. It can be different than the interrupt selected for audio interrupts or the interrupts can be shared as well. See “Assigning Interrupt Sources” on page 14.

If MPU-401 is enabled, a low-level signal on pin MSI prevents power-down and causes automatic wake-up if the ES1887 is powered down. Likewise, power-down is prevented if a byte is currently being received or transmitted.

Temporarily disabling MPU-401 using Mixer register 40h acts as a reset to the FIFOs.

The output of the transmit FIFO is serialized out the MSO pin. MIDI data can be received from the MSI pin of the ES1887.

Game/Joystick Interface

The ES1887 includes 8 pins for a dual joystick port. The digital game port address is decoded for TA, TB, TC, TD, SWA, SWB, SWC, and SWD. The MIDI serial input and output also come from the game port connector in most applications.

Four of these eight pins are inputs for the switches of the joysticks. The remaining 4 pins are "one-shot" timers that generate pulses of varying widths, where the width corresponds to the resistance of one of the joystick potentiometers.

Joystick/MIDI External Interface Connector

The joystick portion of the ES1887 reference design is identical to that on a standard PC game control adaptor or game port. The PC-compatible joystick can be connected to a 15-pin D-sub connector. It supports all standard PC joystick compatible software.

Figure 9 shows a typical application of a combined joystick and MIDI interface. Figure 10 show the required circuitry to connect the ES1887 MIDI I/O signals to MIDI current loop signals.

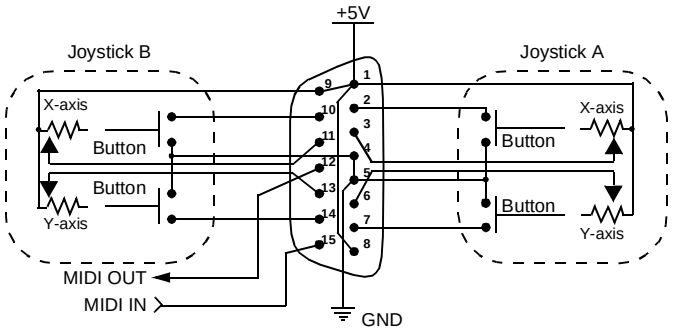


Figure 9 Dual Joystick/MIDI Connector

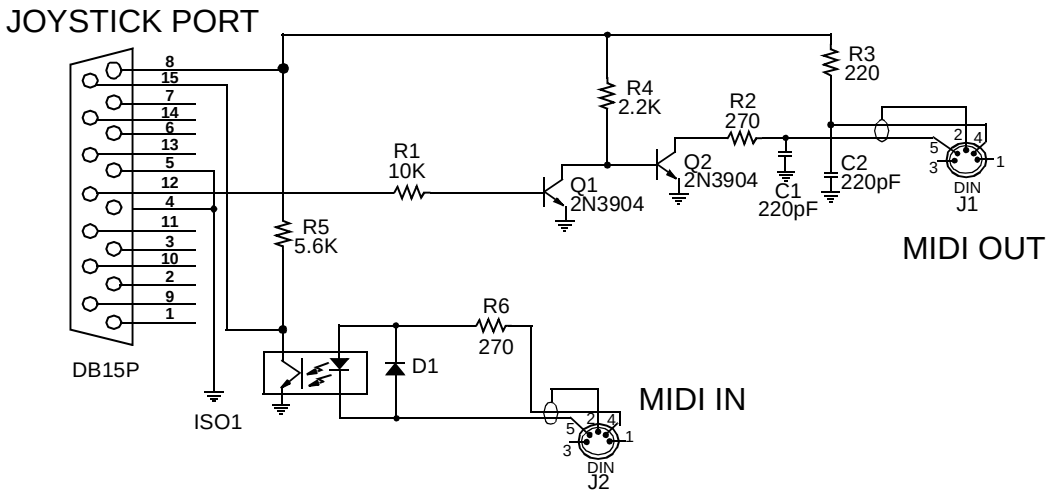


Figure 10 MIDI Serial Interface

Mono FDXI and FDXO

FDXI is shared with AUXB_L and FDXO is shared with AUXB_R. The ES1887 supports the use of FDXI and FDXO as an input to the ADC and output from the DAC when using the DSP serial port. It also supports a new function, where FDXI is a general mono input to the mixer, controlled by the AUXB volume register and FDXO is the mono output of the input volume stage (i.e., the recording source select (1Ch) and input volume control (B4h)).

Mono FDXI/O mode is useful with an external modem that has integrated a CODEC for speakerphone applications.

Bit 0 of Mixer register 46h enables FDXI as a mono input from the left channel filter input to the left channel ADC. When FDXI is a mono input to the mixer, its input impedance is cut in half to 25k ohms. Bit 1 of Mixer register 46h enables FDXO to FOUT_R (right channel filter output). When FDXO is enabled, it has a 5k ohm output impedance. When FDXO is not enabled, it is the AUXB_R input to the mixer and has a 50k ohm pull-up to CMR.

Master Volume

The master volume is controlled through programmed I/O or volume control switch inputs. The master volume supports 6-bits/channel plus mute.

Hardware Volume Controls

These three pins are inputs with internal pull-up devices. A software selectable option enables the mute input to be omitted. The mute input is defined as the state when both up and down inputs are low. By default, this feature is disabled. Otherwise, these pins operate on the master volume registers the same way they do in chips such as the ES1868.

PC Speaker

The PC Speaker is supported with a 1-bit DAC with volume control. The analog output pin PCSPKO is intended to be externally mixed at the external amplifier.

PC Speaker Volume Control

When the PCSPKI signal is high, a resistive path to analog ground is enabled. The value of the resistor is selected from among 7 choices to control the amplitude of the output signal.

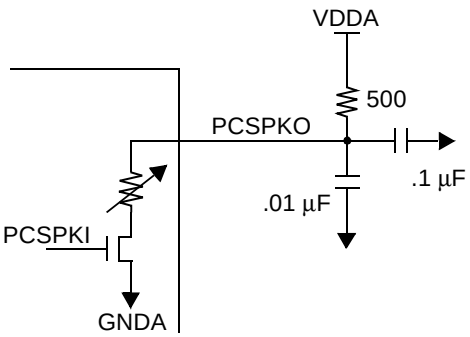


Figure 11 PC Speaker Volume Circuitry

With the external circuit shown in Figure 11, the amplitude of a square wave output on pin PCSPKO should be approximately VDDA/2 for maximum volume, i.e., the internal resistor is approximately 500 ohms (± 30%). The other levels are relative to this amplitude as follows:

off, -18dB, -15dB, -12dB, -9dB, -6dB, -3dB, +0dB

The purpose of the circuit, beyond volume control of the speaker, is to prevent digital noise from the PC speaker signal being mixed into the analog signal. This circuit provides a clean analog signal. The output can be either mixed with the AOUT_L and AOUT_R pins externally or it can be used to drive a simple transistor amplifier to drive an 8 ohm speaker dedicated to producing beeps.

ANALOG DESIGN CONSIDERATIONS

This section describes design considerations related to inputs and outputs of analog signals and related pins on the chip.

Game Port

The game port address Joystick_Base+1h is decoded for timer pins TA, TB, TC, and TD, and switch pins SWA, SWB, SWC, and SWD. The MIDI serial input and output also come from the game port connector in most applications.

Reference Generator

Reference generator pins VREF and CMR are shown bypassed to analog ground.

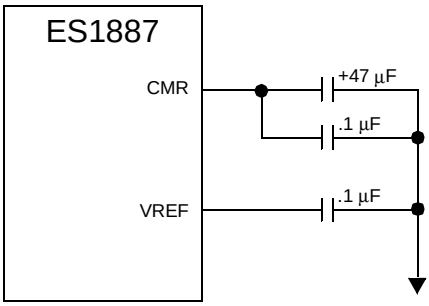


Figure 12 Reference Generator Pin Diagram

Switch-Capacitor Filter

The outputs of the FOUT_L and FOUT_R filters must be AC-coupled to the inputs CIN_L and CIN_R. This provides for DC blocking and an opportunity for low-pass filtering with capacitors to analog ground at these inputs.

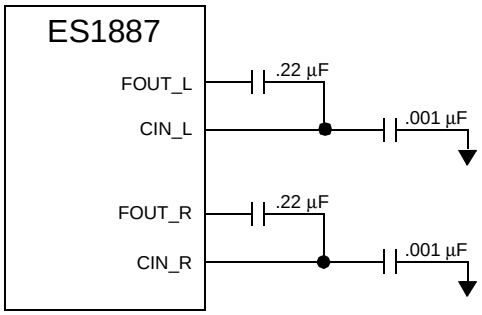


Figure 13 Switch-Capacitor Filter Pin Diagram

Audio Inputs and Outputs

Analog inputs MIC, LINE_L, LINE_R, AUXA_L, and AUXA_R should be capacitively coupled to their respective input signals. All have pull-up resistors to CMR.

ES1887 analog outputs AOUT_L and AOUT_R should be AC-coupled to an amplifier, volume control potentiometer, or line-level outputs.

SOFTWARE ADDRESS CONFIGURATION

Following a hardware reset, the ES1887 is disabled. Only those I/O necessary for the configuration process remain enabled. The ES1887 must first be configured using one of two modes of software address configuration. AMODE is an option input that selects between two different software address configuration modes. When AMODE is low, use one of the two Read-Sequence-Key address configuration methods. When AMODE is high, the System Control Register address configuration mode is selected.

NOTE: The System Control Register method of address selection is intended to be used only for applications where the ES1887 is on the system board. Plug-in cards should use the Read-Sequence-Key method.

ES1887 Read-Sequence-Key Method

This key sequence is always enabled (except if input pin AMODE=1). This is necessary because Mixer register 40h, used to restart the configuration sequence under the ES1888, is not accessible if the audio device is disabled. The sequence sets all bits of System Control Register 0 (register 0E1h), including bits that can be used to select the joystick and FM base addresses.

After a hardware reset the ES1887 is in a disabled state. A special sequence of I/O read operations addressing registers 229h, 22Bh, 22Dh, and 22Fh must be performed followed by three more I/O reads to set audio, joystick, and FM base address. The final read of the sequence enables the ES1887. The sequence will fail if it is interrupted by any I/O write (excluding DMA). The sequence will reset if it is interrupted by an I/O read of an incorrect address (except DMA).

1. Read 22Bh
2. Read 229h
3. Read 22Fh
4. Read 22Dh
5. Read 22Dh
6. Read 22Fh
7. Read 229h
8. Read Audio Base Address: 220h, 230h, 240h, 250h. Any other address leaves audio disabled (i.e. bit 2 of SCR 0 = 0).
9. Read Joystick Base Address: 200h, 201h, 202h, 203h. Any other address leaves joystick disabled (i.e. bit 3 of SCR 0 = 0).
10. Read FM Base Address: 388h, 398h, 3A8h, 3B8h. Any other address leaves FM at 388h (i.e. bits 7:6 of SCR 0 = 0).

NOTE: All interrupts must be disabled when the key is being executed, until after the FM base address has been set. An I/O read from other than 22xh should be done first in order to reset the key sequence.

ES1888 Compatible Read-Sequence-Key Method

The ES1887 is backward compatible to the ES1888 for address configuration. The ES1888 key sequence sets three bits of System Control Register 0 (bits 2:0) and is not always enabled. Specifically, it is enabled after hardware reset and after a 1 is written to bit 2 of Mixer register 40h.

As in the ES1887 method above, a special sequence of I/O read operations addressing registers 229h and 22Bh must be performed followed by a read of the proper base address register to enable the ES1887. The sequence will fail if it is interrupted by any I/O write (excluding DMA) or an I/O read of an address not in the sequence.

1. Read 229h
2. Read 229h
3. Read 229h. Three reads from 229h in a row guarantees reset of key sequence to beginning.
4. Read 22Bh
5. Read 229h
6. Read 22Bh
7. Read 229h
8. Read 229h
9. Read 22Bh
10. Read 229h
11. Read Audio Base Address 220h, 230h, 240h, 250h.

Once enabled, the ES1887 will not respond to this key sequence again. In order to change the address programming, bit 2 of Mixer register 40h must be written high. This generates a command to put the ES1887 into its disabled state as after a hardware reset. A read-key-sequence is then required to return to the enabled state.

System Control Register Method

This method of configuring the ES1887 is enabled if input pin AMODE=1. After a hardware reset, the ES1887 is in a disabled state. A system control register internal to the ES1887 must be written to in order to enable the chip and select the base I/O addresses.

1. Write register 0F9h to unlock the System Control register.
2. Write 0 to 0E0h (the index register).
3. Write configuration data to 0E1h (data register).

   Bits 7:6 FM Base Address: 388h, 398h, 3A8h, 3B8h.

   Bits 5:4 Joystick Base Address: 200h, 201h, 202h, 203h

Bit 3 Joystick enable.

Bit 2 Audio enable.

Bits 1:0 Audio Base Address: 220h, 230h, 240h, 250h.

NOTE: Bits 5:3 are ignored if bit 1 of Mixer register 40h is set. This bit, when high, enables the joystick at 201h. If bit 0 of Mixer register 40h is set, it enables FM alias at the base address specified by bits 7:6 at SCR 0.

4. Write register 0FBh to lock the System Control register.

I/O PORTS

Table 8 I/O Port for Joystick, Audio, FM, and MPU-401 Devices

Port	Read/Write	Function
Software Address Configuration		
0E0h	Write	System Control Index register
0E1h	Read/Write	System Control Data register. Sets the Addresses for the Joystick, Audio, and FM devices.
0F9h	Write	Unlocks the System Control register
0FBh	Write	Locks the System Control register
Joystick Device		
Base+1h	Read/write	Joystick.
Audio Device		
Base+0h - Base+3h	Read/write	20-voice FM synthesizer. Address and data registers.
Base+4h	Read/write	Mixer Address register (port for address of Mixer controller registers).
Base+5h	Read/write	Mixer Data register (port for data to/from Mixer controller registers).
Base+6h	Read/write	Audio reset and status flags.
Base+7h	Read/write	Power Management register. Suspend request and FM reset.
Base+8h - Base+9h	Read/write	11-voice FM synthesizer. Address and data registers.
Base+Ah	Read-only	Input data from read buffer for command/data I/O. Poll bit 7 of port Audio_Base+Eh to test whether the read buffer contents are valid.
Base+Ch	Read/write	Output data to write buffer for command/data I/O. Read embedded microcontroller status.
Base+Eh	Read-only	Data available flag from embedded microcontroller.
Base+Fh	Read/write	Address for I/O access to FIFO in Extended mode.
FM Device		
Base+0h - Base+3h	Read/write	20-voice FM synthesizer. Address and data registers.
MPU-401 Device		
Base+0h - Base+1h	Read/write	MPU-401 port (x=0,1,2, or 3) if enabled.

Port Descriptions

This section describes each port in detail.

System Control Registers

This section describes the system control registers.

System Control Index Register (0E0h, R/W)

Index							
7	6	5	4	3	2	1	0

Write 0 to this register before attempting to write to System Control Register 0.

System Control Register 0 (0E1h, R/W)

FM address		Joystick address		Joystick enable	Audio enable	Audio base address	
7	6	5	4	3	2	1	0

SCR 0 is reset to 0 by hardware reset.

Bits Definitions:

Bits	Name	Description
7:6	FM address	bit 7 bit 6 base address 0 0 388h 0 1 398h 1 0 3A8h 1 1 3B8h Bit 0 of Mixer register 40h, if set, enables FM alias at base address specified by bits 7:6 of SCR 0.
5:4	Joystick address	bit 5 bit 4 base address 0 0 200h 0 1 201h 1 0 202h 1 1 203h Bit 1 of Mixer register 40h, if set, enables joystick at 201h *if audio is also enabled*. This bit, when set, overrides address selected via bits 5:4 of SCR 0.
3	Joystick enable	1 = Enable joystick port. 0 = Disable joystick port. Bit 1 of Mixer register 40h, if set, enables joystick at 201h *if audio is also enabled*. This bit, when set, overrides enable selected via bit 3 of SCR 0.
2	Audio enable	1 = Enable audio port. 0 = Disable audio port.
1:0	Audio base address	bit 1 bit 0 base address 0 0 220h 0 1 230h 1 0 240h 1 1 250h SCR 0 is written during the ES1887 Read-Sequence-Key method if AMODE=0, but not the ES1888 Compatible Read-Sequence-Key method.

Lock System Control Register (0F9h, R/W)

Write: Locks System Control register 0.							
7	6	5	4	3	2	1	0

The “lock” command is an I/O write to address 0F9h. The data written is not significant.

Unlock System Control Register (0FBh, R/W)

Write: Unlocks System Control register 0.							
7	6	5	4	3	2	1	0

To access the System Control Register with the system software, the ES1887 must receive an “unlock” command. The unlock command is an I/O write to address 0FBh. The data written is not significant.

Audio Device

This section describes the audio ports.

Mixer Address Port (Audio_Base+4h, R/W)

x	A6	A5	A4	A3	A2	A1	A0
7	6	5	4	3	2	1	0

When commanding a mixer register (read or write), write the address of the mixer register to be command to this port.

Mixer Data Port (Audio_Base+5h, R/W)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

This port holds read data or write data for the register address written to the Mixer Address port.

Reset and Status Flags (Audio_Base+6h, W)

0	0	0	0	0	0	FIFO reset	SW reset
7	6	5	4	3	2	1	0

Bits Definitions:

Bits	Name	Description
7:2	–	Reserved. Always write 0.
1	FIFO reset	1 = Hold ES1887 FIFO in reset. 0 = Release ES1887 FIFO from reset. Bit 1 has no function in Compatibility mode.
0	SW reset	1 = Hold ES1887 in reset. 0 = Release ES1887 from reset.

Reset and Status Flags (Audio_Base+6h, R)

Act flag 2	Act flag 1	Act flag 0	Serial act flag	Digital power-down	MIDI modes	FIFO reset	SW reset
7	6	5	4	3	2	1	0

Bits 7:4 of port Audio_Base+6h can be used to monitor I/O activity to the ES1887.

Bit 7:5 are set high after any read from port Audio_Base+6h. Then specific I/O activity can set these bits low. When port Audio_Base+6h is read at a later time, these bits indicate whether I/O activity has occurred between the reads from Audio_Base+6h.

In addition, bit 4 can be used to indicate if the DSP or ES689/ES69x serial interface is in use. Bit 4 is set high if bit 7 or bit 5 of Mixer register 48h is high (software serial enable or serial reset). It is also set high if the ES689/ES69x serial interface is active, which is a combination of bit 4 of Mixer register 48h set high and MCLK (ES689/ES69x serial bit clock) being high periodically.

Bits Definitions:

Bits	Name	Description
7	Act flag 2	Set low by I/O reads/writes to MPU-401 or FM ports.
6	Act flag1	Set low by I/O reads/writes to audio ports Audio_Base+Ch and Audio_Base+Eh.
5	Act flag 0	Set low by I/O writes to audio ports Audio_Base+2h, Base+3h, Base+6h, and Base+Ch. Set low by I/O reads from audio ports Audio_Base+2h, Base+3h, and Base+Ah. Also set low by DMA accesses to ES1887.
4	Serial act flag	1 = Serial activity flag. High if DSP serial mode is enabled (SE input pin is high or bits 7or 5 of register 48h is high) or if an external ES689/ES69x is using MCLK/MSD to drive the music DAC.
3	Digital power-down	0 = The ES1887 digital section is currently powered-down (partial or full power-down). Power to the analog section is controlled by bit 3 of Audio_Base+7h.
2	MIDI mode	1 = The ES1887 is processing a MIDI command 30h, 31h, 34h, or 35h. In this mode, the ES1887 is monitoring the serial input. Powering-down may cause a loss of data. The ES1887 does not automatically wake up on serial input on the MSI pin.
1	FIFO reset	FIFO Reset bit.
0	SW reset	Software Reset bit.

Power Management Port (Audio_Base+7h, R/W)

Suspend request	0	FM synth reset	0	Analog power-down	Power-down request	GPO1	GPO0
7	6	5	4	3	2	1	0

Reading or writing port Audio_Base+7h does not automatically wake-up the ES1887.

Bits Definitions:

Bits	Name	Description
7	Suspend request	Pulse high, then low to request suspend.
6	–	Reserved. Always write 0.
5	FM synth reset	1 = Hold FM synthesizer in reset. 0 = Release FM synthesizer from reset.
4	–	Reserved. Always write 0.
3	Analog power-down	1 = Set Analog_Stays_On 0 = Clear Analog_Stays_On
2	Power-down request	Pulse high, then low to request power-down.
1	GPO1	1 = Set GPO1 high (Hardware reset condition). 0 = Clear GPO1.
0	GPO0	1 = Set GPO0 high. 0 = Clear GPO0 (Hardware reset condition).

Read Data Port (Audio_Base+Ah, R)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

Read data from the embedded audio microcontroller. Poll bit 7 of port Audio_Base+Eh to test whether the register contents are valid.

Write Data Port (Audio_Base+Ch, W)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

Write data to the embedded audio microcontroller. Sets bit 7 of port Audio_Base+Ch high (write buffer not available) until the data is processed by the ES1887.

Read Data Port (Audio_Base+Ch, R)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

Bits Definitions:

Bits	Name	Description
7	BUSY flag	1 = write buffer not available or ES1887 busy. 0 = write buffer available or ES1887 not busy.
6		1 = Data available in read buffer. 0 = Data not available in read buffer. This flag is reset by a read from port Audio_Base+Ah.
5		1 = Extended mode FIFO Full (256 bytes loaded).
4		1 = Extended mode FIFO Empty (0 bytes loaded).
3		1 = FIFO Half Empty, Extended mode flag.
2		1 = ES1887 microcontroller generated an interrupt request (e.g., from Compatibility mode DMA complete).
1		1 = Interrupt request generated by FIFO Half Empty flag change. Used by programmed I/O interface to FIFO in Extended mode.
0		1 = Interrupt request generated by DMA counter overflow in Extended mode.

Read Buffer Status Port (Audio_Base+Eh, R)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

A read from port Audio_Base+Eh will reset any interrupt request.

Bits Definitions:

Bits	Name	Description
7		1 = Data available in read buffer. 0 = Data not available in read buffer. This flag is reset by a read from port Audio_Base+Ah.

Programmed I/O Access to FIFO Port (Audio_Base+Fh, R/W)

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

This port can be used to replace Extended mode DMA transfers with programmed I/O transfers.

PROGRAMMING THE ES1887

Identifying the ES1887

There is no single command to identify the ES1887. To determine if the chip in question is an ES1887 a number of diagnostic steps must be taken. If any of these steps fails, then the chip is not an ES1887.

1. Send the E7h audio microcontroller command. The E7h command returns the Legacy ES688/ES1688 identification bytes: 68h 8xh. If x is greater than or equal to 8, continue to step 2. If x is less than 8 then the chip might be an ES688.
2. Check if you can change the value of bit 3 of register 64h. If yes, continue to step 3. If no, the chip might be an ES1688.

NOTE: Be sure to restore the value of this bit.

3. Check if the value of register 70h can be changed. If yes, continue to step 4. If no, the chip might be an ES1788.
4. Check if the value of bit 5 of register 64h can be changed. If yes, the chip in question is an ES1887. If no, the chip might be an ES1888.

NOTE: Be sure to restore the value of this bit.

Resetting the ES1887 by Software

The ES1887 embedded audio microcontroller can be reset in one of two ways: hardware reset or software reset. The hardware reset signal comes from the ISA bus. Software reset is controlled by bit 0 of port Audio_Base+6h.

To reset the ES1887 audio microcontroller by software:

1. Write a 1 to port Audio_Base+6h.
2. Delay at least 10 msec.
3. Write a 0 to port Audio_Base+6h.
4. In a loop that lasts at least 1 msec, poll port Audio_Base+Eh bit 7=1 for read data available.

   If bit 7=1, then read the byte from port Audio_Base+Ah. Exit loop if the content is 0AAh; otherwise, continue polling.

Both hardware reset and software reset:

- Disable Extended mode
- Reset the timer divider and filter registers for 8 kHz sampling
- Stop any DMA in progress
- Clear any active interrupt request
- Disable input to the mixer from the first audio channel DAC (see the D1h/ D3h commands)
- Reset Compatibility mode and Extended mode DMA counters to 2048 bytes
- Set CODEC direction to be DAC, with the DAC value set to mid-level
- Set input volume for 8-bit recording with Automatic Gain Control (AGC) to maximum
- Set input volume for 16-bit recording to mid-range

In addition to performing actions on the above list, a hardware reset resets all Mixer registers to default values.

Modes of Operation

The ES1887 can operate the first audio channel in one of two modes: Compatibility mode or Extended mode.

In both modes, a set of mixer and controller registers enables application software to control the analog mixer, record source, and output volume. Programming the ES1887 mixer is described later in this document. See “Programming the ES1887 Mixer” on page 42.

Table 9 lists the features of the two modes.

Compatibility Mode Description

The first mode, Compatibility mode, is compatible to the Sound Blaster Pro. This is the default mode after reset. In this mode, the ES1887 microcontroller is an intermediary in all functions between the ISA bus and the CODEC. The ES1887 microcontroller performs limited FIFO functions using 64 bytes of internal memory.

Extended Mode Description

The ES1887 also supports an Extended mode of operation. In this case a 256-byte FIFO is used as an intermediary between the ISA bus and the ADC and DAC Control registers, and various Extended mode controller registers are used for control. The ES1887 microcontroller is mostly idle in this mode. DMA control is handled by dedicated logic. Programming for Extended mode operation requires accessing various control registers with ES1887 commands. Some of these commands are also useful for Compatibility mode, such as those that configure DMA and IRQ channels.

Table 9 Comparison of Operation Modes

	Compatibility Mode (Sound Blaster Pro)	Extended Mode
Sound Blaster Pro compatible	Yes	No
FIFO Size	64 bytes (firmware managed)	256 bytes (hardware managed)
Mono 8-bit ADC, DAC	Yes, to 44 kHz	Yes, to 44 kHz
Mono 16-bit ADC, DAC	Yes, to 22 kHz[a]	Yes, to 44 kHz
Stereo 8-bit DAC	Yes, to 22 kHz	Yes, to 44 kHz
Stereo 8-bit ADC	Yes, to 22 kHz	Yes, to 44 kHz
Stereo 16-bit DAC	Yes, to 11 kHz[a]	Yes, to 44 kHz
Stereo 16-bit ADC	No	Yes, to 44 kHz
Signed/Unsigned Control	No	Yes
Automatic Gain Control during recording	Firmware controlled, to 22 kHz, mono only	No
Programmed I/O block transfer for ADC and DAC	No	Yes
FIFO status flags	No	Yes
Auto reload DMA	Yes	Yes
Time base for programmable timer	1 MHz or 1.5 MHz	800 kHz or 400 kHz
ADC and DAC jitter	± 2 microseconds	None

a. The support of 16-bit data is a superset feature of the ES1887.

Mixing Modes Not Recommended

In general, avoid mixing Extended mode commands with Compatibility mode commands where possible. The Audio 1 DAC Enable/Disable commands D1h and D3h are safe to use when using Extended mode to process ADC or DAC. However, there are other Compatibility mode commands that are likely to cause problems. The Extended mode commands may be used to set up the DMA or IRQ channels before entering the Compatibility mode.

Data Formats

This section briefly describes the different audio data formats used by the ES1887.

Compressed Data Formats

The ES1887 supports two types of compressed sound DAC operations: **ESPCM™**, which uses a variety of proprietary compression techniques developed by ESS Technology, Inc. and ADPCM, which is supported by other sound cards but is of a lower quality.

Both ADPCM and **ESPCM™** are only transferred using DMA transfer. The first block of a multiple-block transfer uses a different command than subsequent blocks. The first byte of the first block is called the reference byte.

Use Compatibility mode when transferring compressed data.

Sound Blaster Pro Compatible Data Formats

The ES1887 supports Sound Blaster Pro data formats in Compatibility mode. In addition, the ES1887 supports 16-bit data formats whereas the Sound Blaster Pro only supports 8-bit data formats.

There are four formats available from the combination of the following two options:

- 8-bit or 16-bit
- Mono or stereo

The 8-bit samples are unsigned, ranging from 0 to 0FFh, with the DC level around 80h.

16-bit samples are unsigned, least byte first, ranging from 0000h to 0FFFFh with the DC level around 8000h.

Stereo DMA Transfers in Compatibility Mode

Stereo DMA transfers are only available using DMA rather than direct mode commands.

To perform a stereo DMA transfer, first program bit 1 of Mixer register 0Eh to be high. Then set the timer divider to twice the per-channel sample rate.

The maximum stereo transfer rate for 8-bit data is 22 kHz per channel; so for this case, program the timer divider as if you were doing 44 kHz mono. The maximum stereo DAC

transfer rate for 16-bit data is 11 kHz per channel. Stereo ADC transfers for 16-bit data is not allowed in Compatibility mode.

For 8-bit data, the ES1887 expects the first byte transferred to be for the right channel, and subsequent bytes to alternate left, right etc.

For 16-bit data, the ES1887 expects the DMA transfers to be a multiple of 4, with repeating groups in the order:

1. left low byte
2. left high byte
3. right low byte
4. right high byte

ES1887 Data Formats (Extended mode and Audio 2)

There are eight formats available from the combination of the following three options:

- Mono or stereo
- 8-bit or 16-bit
- Signed or unsigned

For stereo data, the data stream always alternates channels in successive samples: first left, then right. This is the opposite of Compatibility mode. For 16-bit data, the low byte always precedes the high byte.

Sending Commands During DMA Operations

It is useful to understand the detailed operation of sending a command during DMA.

The ES1887 uses the Audio 1 FIFO for DMA to/from the CODEC transfers. When the FIFO is full (in the case of DAC, empty in the case of ADC), DMA requests are temporarily suspended and the BUSY flag (bit 7 of port Audio_Base+Ch) is cleared. This allows a window of opportunity to send a command to the ES1887. Commands such as D1h and D3h, which control the Audio 1 DAC mixer input enable/disable status, and command D0h, which suspends (i.e., pauses) DMA are acceptable to send during this window.

The ES1887 chip sets the BUSY flag when the command window is no longer open. Application software must send a command within 13 μsec after the BUSY flag goes high or the command will be confused with DMA data. This is normally easy to do if the polling is done with interrupts disabled.

As an example of sending a command during DMA, consider the case where the application desires to send command D0h in the middle of a DMA transfer. The application disables interrupts and polls the BUSY flag. Because of the FIFO and the rules used for determining the command window, it is possible for the current DMA transfer to complete while waiting for the busy flag to clear. In this event, the D0 command has no function, and there will be a pending interrupt request from the DMA completion.

This interrupt request can be cleared by reading port Audio_Base+Eh before enabling interrupts or having a way of signaling the interrupt handler that DMA is inactive so that it does not try to start a new DMA transfer.

Figure 14 shows timing considerations for sending a command.

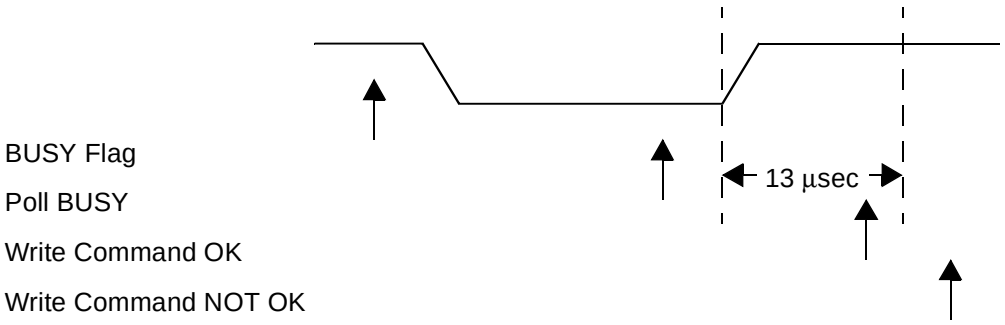


Figure 14 Command Transfer Timing

Compatibility Mode Programming

The following section describes Compatibility mode programming.

Compatibility Mode DAC Operation

1. Reset.

   Write 1h to port Audio_Base+6h.

   If you do not wish to reset the ES1887 before playing a new sound, and you are not certain of the status of the analog circuits, you can mute the input to the mixer with command D3h to prevent pops.

2. Enable stereo mode (optional).

   Set bit 1 of mixer register 0Eh high. Use only DMA mode. Be sure to clear bit 1 of Mixer register 0Eh after the DAC transfer.

3. Set sample rate and filter clock.

   Use commands 40h or 41h to set the sample rate and filter clock divider. If you want to set the filter clock independent from the sample rate use command 42h in addition to 40h or 41h.

   For stereo transfers, set the timer divider to twice the per-channel sample rate. The maximum stereo transfer rate for 8-bit data is 22 kHz per channel; so for this case, program the timer divider as if you were doing 44 kHz mono. The maximum stereo transfer rate for 16-bit data is 11 kHz per channel.

4. Set the block size. Only use this command (48h) with high-speed DMA transfer modes (commands 90h 91h).
5. Configure system interrupt controller and system DMA controller.
6. Start DMA.

   Start the DMA transfer by sending the command for the desired transfer type and data length. The uncompressed modes are shown in Table 10. See “Audio Microcontroller Command Summary” on page 61 for a description of the commands in addition to the commands for DMA transfers of compressed data.

Table 10 Uncompressed DAC Transfer Modes

DAC DMA Transfer Mode	Data Length	Command
Direct	8-bit	10h
	16-bit	11h
DMA mode normal	8-bit	14h
	16-bit	15h
high-speed	8-bit	91h
DMA mode auto-initialize	8-bit	1Ch
	16-bit	1Dh
high-speed	8-bit	90h

7. Delay approximately 100 msec to allow the analog circuits to settle, then enable the Audio 1 DAC input to mixer with command D1h.
8. During DMA: For auto-initialize, you do not need to send any commands to the ES1887 at interrupt time, except for reading Audio_Base+Eh to clear the interrupt request.

   For normal mode, you must initialize the system DMA controller with the address and count of the next block to transfer. You must update the ES1887 transfer block size if it changes. Use command 48h. To start the next transfer, use command D4h.

   To stop DMA after the current auto-initialize block is finished, use command D0h.

   Commands such as D1h and D3h which control the Audio 1 DAC mixer input enable/disable status, and command D0h, which suspends (i.e., pauses) DMA are acceptable to send during DMA transfers. These commands can only be sent during certain windows of opportunity. See “Stereo DMA Transfers in Compatibility Mode” on page 29.

9. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in register Audio_Base+Ch to be sure data transfer is completed. Delay 25 msec if possible to let the filter outputs settle to DC level, then disable the Audio 1 DAC input to the mixer with command D3h.
10. Issue another software reset to the ES1887 to leave the chip in a well-defined state.

Compatibility Mode ADC Operation

The ES1887 analog circuitry is switched from the DAC direction to the ADC direction by the first direct or DMA mode ADC command (2xh). If possible, discard the first 25-100 msec of samples, because pops may occur in the data due to the change from the DAC to ADC direction. In the ADC direction input to the mixer is automatically muted.

1. Reset.

   Write 1h to port Audio_Base+6h.

   If you do not wish to reset the ES1887 before playing a new sound, and you are not certain of the status of the analog circuits, you can mute the input to the mixer with command D3h to prevent pops.

2. Select the input source using register 0Ch.

   Sound Blaster Pro has three recording sources: microphone, line, and auxiliary A (CD). Microphone input is the default source after any reset.

   The ES1887 has four recording sources: microphone, line, auxiliary A, and mixer. Use Mixer register 1Ch to choose the additional source.

3. Program the input volume.

   The selected source passes through an input volume stage that can be programmed with 16 levels of gain from 0 to +22.5 db in steps of 1.5 db. In 8-bit recordings (other than “high-speed mode”), the volume stage is controlled by the ES1887 firmware for the purposes of automatic gain control (AGC). In 16-bit recordings as well as “high-speed mode” 8-bit recordings, the input volume stage is controllable from application software. Use command DDh to change the input volume level from 0 to 15. The reset default is mid-range, 8.

4. Enable stereo mode (optional).

   Set bit 1 of mixer register 0Eh high. Use only DMA mode. Be sure to clear bit 1 of Mixer register 0Eh after the ADC transfer.

5. Set sample rate and filter clock.

   Use commands 40h or 41h to set the sample rate and filter clock divider. If you want to set the filter clock independent from the sample rate use command 42h in addition to 40h or 41h.

   For stereo transfers, set the timer divider to twice the per-channel sample rate. The maximum stereo transfer rate for 8-bit data is 22 kHz per channel; so for this case, program the timer divider as if you were doing 44 kHz mono. The maximum stereo transfer rate for 16-bit data is 11 kHz per channel.

6. Set the block size. Only use this command (48h) with high-speed DMA transfer modes (commands 98h 99h).

7. Configure system interrupt controller and system DMA controller.

8. Start DMA.

   Start the DMA transfer by sending the command for the desired transfer type and data length. The uncompressed modes are shown in Table 11. See “Audio Microcontroller Command Summary” on page 61 for a description of the commands in addition to the commands for DMA transfers of compressed data.

Table 11 Uncompressed ADC Transfer Modes

ADC DMA Transfer Mode	Data Length	Command
Direct	8-bit	20h
	16-bit	21h
DMA mode normal	8-bit	24h
	16-bit	25h
high-speed	8-bit	99h
DMA mode auto-initialize	8-bit	2Ch
	16-bit	2Dh
high-speed	8-bit	98h

9. Delay approximately 100 msec to allow the analog circuits to settle, then enable the Audio 1 DAC input to mixer with command D1h.

10. During DMA: For auto-initialize, you do not need to send any commands to the ES1887 at interrupt time, except for reading Audio_Base+Eh to clear the interrupt request.

    For normal mode, you must initialize the system DMA controller with the address and count of the next block to transfer. You must update the ES1887 transfer block size if it changes. Use command 48h. To start the next transfer, use command D4h.

    To stop DMA after the current auto-initialize block is finished, use command D0h.

    Commands such as D0h, which suspends (i.e., pauses) DMA are acceptable to send during DMA transfers. These commands can only be sent during certain windows of opportunity. See “Writing Commands to the ES1887 Controller Registers” on page 34.

11. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in register Audio_Base+Ch to be sure data transfer is completed.
12. Issue another software reset to the ES1887 to leave the chip in a well-defined state.

The maximum sample rate for direct mode ADC is 22 kHz.

The maximum sample rate for DMA ADC for both 8-bit and 16-bit is 22 kHz, using commands 24h, 25h, 2Ch, or 2Dh.

There is a special “high-speed mode” for ADC that allows 8-bit sampling up to 44 kHz. This mode uses commands 98h (“auto-initialization) and 99h (“normal”). No AGC is performed: The input volume is controlled with command DDh.

Extended Mode Programming

This section describes Extended mode programming.

Commanding the ES1887 Controller Registers

Controller registers are written to and read from using commands sent to ports Audio_Base+Ch and Audio_Base+Ah.

Commands of the format Axh, Bxh, and Cxh, where x is a numeric value, are used for Extended mode programming of the first audio channel.

Commands of the format Ax and Bx are used to access the ES1887 controller registers. For convenience, the registers are named after the commands used to access them. For example, “controller register A4h”, the Audio 1 Transfer Count Reload (low-byte) register, is written to via “command A4h.”

Enabling Extended Mode Commands

After any reset, before using any Extended mode commands you must first send command C6h to enable Extended mode commands.

ES1887 Command/Data Handshaking Protocol

This section describes how to write commands to and read data from the ES1887 controller registers.

Writing Commands to the ES1887 Controller Registers

Commands written to the chip enter a write buffer. Before writing the command, you must make sure the buffer is not busy.

Bit 7 of port Audio_Base+Ch is the ES1887 BUSY flag. It is set when the write buffer is full or when the ES1887 is otherwise busy (for example, during initialization after reset or during Compatibility mode DMA requests).

To write a command or data byte to the ES1887 microcontroller:

1. Poll bit 7 of port Audio_Base+Ch until it is clear.
2. Write the command/data byte to port Audio_Base+Ch.

The following shows an example of writing to an ES1887 controller registers. To set up the Audio 1 Transfer Count Reload register to F800h, send the following command/ data bytes:

```
A4h, 00h; register A4 = 0
A5h, F8h; register A5 = F8
```

NOTE: The port Audio_Base+Ch write buffer is shared with Compatibility mode DMA write operations. When DMA is active, the BUSY flag is cleared during windows of time when a command can be received. Normally, the only commands that should be sent during DMA operations are Dxh commands: DMA pause/continue, Audio 1 DAC enable/disable, etc. In this situation it is recommended that interrupts be disabled between the time that the busy bit is polled and the command is written. Also, the time between these instructions should be minimized. For more information, See “Sending Commands During DMA Operations” on page 30.

Reading the Read Data Buffer of the ES1887

Command C0h is used to read the ES1887 Controller registers used for Extended mode. Send command C0h followed by the register number, Axh or Bxh. For example, to read register A4h, send the following command bytes:

```
C0h, A4h
```

Then poll the Read Data Buffer Status bit, before reading the register contents from port Audio_Base+Ah. The Read Data Buffer Status flag can be polled by reading bit 7 of port Audio_Base+Eh. When a byte is available the bit is set high.

NOTE: Any read of port Audio_Base+Eh also clears any active interrupt request from the ES1887. An alternative way of polling the read buffer status bit is via bit 6 of port Audio_Base+Ch, which is the same flag. The buffer status flag is cleared automatically by reading the byte from port Audio_Base+Ah.

Extended Mode DAC Operation

Follow the steps below to program the first audio channel for Extended mode DAC operation:

1. Reset.

   Write 3 to port Audio_Base+6h instead of 1 as in Compatibility mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to Compatibility mode. Reset disables the Audio 1 DAC input to the mixer. This is intended to mask any pops created during the setup of the DMA transfer.

2. After the reset, send command C6h to enable Extended mode commands.
3. Program direction and type: registers B8h, A8h, and B9h:

   Register B8h: Set bit 2 low for normal DMA mode, high for auto-initialize DMA mode. Leave bit 3 low for the CODEC to run in the DAC direction.

   Register A8h: modify only bits 1 and 0 of this register, read it first to preserve the remaining bits of this register:

   Bits 1:0 = 10 = mono.

   Bits 1:0 = 01 = stereo.

   Register B9h:

   Bits 1:0 = 00 = Single Transfer DMA.

   Bits 1:0 = 01 = Demand Transfer DMA: 2 bytes per DMA request.

   Bits 1:0 = 10 = Demand Transfer DMA: 4 bytes per DMA request.

4. Clocks and counters: registers A1h, A2h, A4h, and A5h:

   Register A1h = Sample Rate Clock Divider
   Register A2h = Filter Clock Divider
   Registers A4h/A5h = Audio 1 Transfer Count Reload register low/high, 2's complement

5. Initialize and Configure DAC: registers B6h and B7h. See Table 12.

   The first two commands written to registers B6h and B7h are designed to prevent pops.

   Registers B7h: Programs the FIFO (16-bit/8-bit, signed/unsigned, stereo/mono).

Table 12 Command Sequences for DMA Playback

Mono	Stereo	8-bits	16-bits	Unsigned	Signed	Sequence
X		X		X		Reg B6 = 80h Reg B7 = 51h Reg B7 = D0h
X		X			X	Reg B6 = 00h Reg B7 = 71h Reg B7 = F0h
X			X	X		Reg B6 = 80h Reg B7 = 51h Reg B7 = D4h
X			X		X	Reg B6 = 00h Reg B7 = 71h Reg B7 = F4h
	X	X		X		Reg B6 = 80h Reg B7 = 51h Reg B7 = 98h
	X	X			X	Reg B6 = 00h Reg B7 = 71h Reg B7 = B8h
	X		X	X		Reg B6 = 80h Reg B7 = 51h Reg B7 = 9Ch
	X		X		X	Reg B6 = 00h Reg B7 = 71h Reg B7 = BCh

6. Enable/Select DMA Channel and IRQ Channel, registers 7Fh, B1h, and B2h:

   Register 7Fh – Interrupt Control and Status register. Select the desired interrupt.

   Register B1h – Interrupt Configuration register. Make sure bits 4 and 6 are high, clear bits 7 and 5.

   Register B2h – DRQ Configuration register. Make sure bits 4 and 6 are high; clear bits 7 and 5.

7. Configure system interrupt controller and DMA controller
8. To Start DMA:

   Set bit 0 of register B8h high while preserving all other bits.

9. Delay approximately 100 msec to allow the analog circuits to settle, then enable the Audio 1 DAC input to mixer with command D1h.
10. During DMA:

    For auto-initialize, you do not need to send any commands to the ES1887 at interrupt time, except for reading Audio_Base+Eh to clear the interrupt request.

For normal mode, you must initialize the system DMA controller with the address and count of the next block to transfer. You must update the ES1887 transfer count registers if the count is changed. To start the next transfer, clear bit 0 of register B8h, then set it high again.

To stop DMA at any time, clear bit 0 of register B8h. To stop DMA after the current auto-initialize block is finished, clear bit 2 of register B8h, wait for the interrupt, and then clear bit 0 of the B8h.

11. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in register Audio_Base+Ch to be sure data transfer is completed. Delay 25 msec if possible to let the filter outputs settle to DC level, then disable the Audio 1 DAC input to the mixer with command D3h.
12. Finally, issue another software reset to the ES1887 to leave things in a well-defined state.

Extended Mode ADC Operation

Follow the steps below to program the first audio channel for Extended mode ADC operation:

NOTE: In Extended mode, there is no AGC performed while recording. If AGC is necessary, use 16-bit recordings and perform AGC in system software.

1. Reset.

   Write 3 to port Audio_Base+6h instead of 1 as in Compatibility mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to Compatibility mode. Reset disables the Audio 1 DAC input to the mixer. This is intended to mask any pops created during the setup of the DMA transfer.

2. Send command C6h to enable Extended mode commands.
3. Select the input source.

   The ES1887 has four recording sources: microphone, line, auxiliary A, and mixer. The mixer source can be the playback Mixer or the Record mixer. Bits 4:3 of mixer register 7Ah selects the mixer source. The Record mixer is the default. Microphone input is the default source after any reset. Select the source using the Mixer Control register 1Ch.

4. Program Input Volume register B4h.
5. Program direction and type: registers B8h, A8h.

   Register B8h: Set bit 3 high to program the CODEC for the ADC direction. Set bit 2 low for normal DMA mode, high for auto-initialize DMA mode.

   At this point, the direction of the analog circuits is ADC rather than DAC. Unless recording monitor is enabled, there will be no output from AOUT L/R until the direction is restored to DAC.

   Register A8h: modify only bits 3, 1, and 0 of this register; read it first to preserve the remaining bits of this register:

   Bits 1:0 = 10 = mono.

   Bits 1:0 = 01 = stereo.

   Bit 3 = 0 Disable record monitor for now.

   Register B9h:

   Bits 1:0 = 00 = Single Transfer DMA.

   Bits 1:0 = 01 = Demand Transfer DMA: 2 bytes per DMA request.

   Bits 1:0 = 10 = Demand Transfer DMA: 4 bytes per DMA request.

6. Clocks and counters: registers A1h, A2h, A4h, and A5h.
   a. Register A1h = Sample Rate Clock Divider. Set bit 7 high for sample rates greater than 22 kHz.
   b. Register A2h = Filter Clock Divider.
   c. Registers A4h/A5h = Audio 1 Transfer Count Reload register low/high, 2's complement.
7. Delay 100 msec to allow the analog circuits to settle.
8. Enable record monitor if desired:

   Register A8h Bit 3 = 1 Enable Record Monitor (optional).

9. Initialize and Configure ADC: register B7h. See Table 13. The first command sent to register B7h initializes the DAC and prevents pops.

Register B7h: Programs the FIFO (16-bit/8-bit, signed/ unsigned, stereo/mono).

Table 13 Command Sequence for DMA Record

Mono	Stereo	8-bits	16-bits	Unsigned	Signed	Sequence
x		x		x		Reg B7 = 51h Reg B7 = D0h
x		x			x	Reg B7 = 71h Reg B7 = F0h
x			x	x		Reg B7 = 51h Reg B7 = D4h
x			x		x	Reg B7 = 71h Reg B7 = F4h
	x	x		x		Reg B7 = 51h Reg B7 = 98h
	x	x			x	Reg B7 = 71h Reg B7 = B8h
	x		x	x		Reg B7 = 51h Reg B7 = 9Ch
	x		x		x	Reg B7 = 71h Reg B7 = BCh

10. Enable/Select DMA Channel and IRQ Channel, registers 7Fh, B1h, and B2h:

    Register 7Fh – Interrupt Control and Status register. Select the desired interrupt.

    Register B1h – Interrupt Configuration register. Make sure bits 4 and 6 are high, clear bits 7 and 5.

    Register B2h – DRQ Configuration register. Make sure bits 4 and 6 are high; clear bits 7 and 5.

11. Configure system interrupt controller and DMA controller.
12. To start DMA: Set bit 0 of register B8h high. Leave other bits unchanged.
13. During DMA:

    For auto-initialize, you do not need to send any commands to the ES1887 at interrupt time, except for reading Audio_Base+Eh to clear the interrupt request.

    For normal mode, initialize the system DMA controller with the address and count of the next block to transfer. Update the ES1887 Transfer Count registers if the count is changed. To start the next transfer, clear bit 0 of register B8h, then set it high again.

    To stop DMA at any time, clear bit 0 of register B8h. To stop DMA after the current auto-initialize block is finished, clear bit 2 of register B8h, wait for the interrupt, and then clear bit 0 of the B8h.

14. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state.
15. Finally, issue another software reset to the ES1887 to leave things in a well-defined state. This will return the ES1887 to the DAC direction and turn off the record monitor.

Extended Mode Programmed I/O Operation

The REP OUTSB instruction of the 80x86 family transfers data from memory to an I/O port specified by the DX register. The REP INSB instruction is the complementary function. Use ES1887 port Audio_Base+Fh for block transfers.

I/O transfers to FIFO are nearly identical to the DMA process, except that an I/O access to port Audio_Base+Fh replaces the DMA cycle. Some differences are described here.

To program in this mode, it is useful to understand how the FIFO Half-Empty flag generates an interrupt request: An interrupt request is generated on the rising edge of the FIFO Half-Empty flag. This flag can be polled by reading port Audio_Base+Ch. The meaning of this flag depends on the direction of the transfer:

DAC FIFOHE flag is set high if 0-127 bytes in FIFO.

ADC FIFOHE flag is set high if 128-256 bytes in FIFO.

Therefore, for DAC, an interrupt request is generated when the number of bytes in the FIFO changes from $\geq$ 128 to < 128. This indicates to the system processor that 128 bytes can be safely transferred without overfilling the FIFO. Before the first interrupt can be generated, the FIFO needs to be primed, or filled, with more than 128 bytes. Keep in mind that data may be taken out of the FIFO while it is being filled by the system processor. If that is the case, there may never be $\geq$ 128 bytes in the FIFO unless you transfer somewhat more than 128 bytes. A solution is to poll the ES1887 FIFOHE flag to be sure it goes low in the interrupt handler (or when priming the FIFO) and perhaps send a second block of 128 bytes.

For ADC, the interrupt request is generated when the number of bytes in the FIFO changes from < 128 to $\geq$ 128, indicating that the system processor can safely read 128 bytes from the FIFO. Before the first interrupt can be generated, the FIFO should be emptied (or mostly so) by reading from Audio_Base+Fh and polling the FIFOHE flag. It is not safe to indiscriminately use the FIFO reset, bit 1, of port Audio_Base+6h to clear the FIFO, because it may get ADC data out-of-sync.

As in DMA mode, bit 0 of register B8h enables transfers between the system and the FIFO inside the ES1887.

NOTE: The ES1887 is designed for I/O block transfer up to an ISA bus speed of 8.33 MHz.

Programmed I/O DAC Operation

Programmed I/O DAC operation is done just as explained under “Extended Mode DAC Operation” on page 35 with the following exceptions:

- In step 3, programming register B9h is unnecessary.
- In step 6, leave bits 7:5 of register B2h low. Set bit 5 of register B1h high to enable an interrupt on FIFO half-empty transitions. Keep bit 6 of register B1h low.
- In step 8, in addition to setting bit 0 of register B8h high, send the REP OUTSB instruction.

Programmed I/O ADC Operation

Programmed I/O ADC operation is done just as explained under “Extended Mode ADC Operation” on page 36 with the following exceptions:

- In step 3, programming register B9h is unnecessary.
- In step 6, leave bits 7:5 of register B2h low. Set bit 5 of register B1h high to enable an interrupt on FIFO half-empty transitions. Keep bit 6 of register B1h low.
- In step 8, in addition to setting bit 0 of register B8h high, send the REP OUTSB instruction.

Programming the Second Audio Channel

This section describes how to program the second audio channel.

Second Audio Channel DAC Operation

Follow the steps below to program the second audio channel for DAC operation:

1. Reset.

   Write 3 to port Audio_Base+6h instead of 1 as in Compatibility mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to Compatibility mode. On reset the playback mixer volume for the second audio channel is set to zero, register 7Ch. This masks any pops that might occur from the setup process.

2. Program transfer type: register 78h:

   Register 78h: Set bit 4 low for normal DMA mode, high for auto-initialize DMA mode.

   Bits 7:6 = 00 = Single Transfer DMA.

   Bits 7:6 = 01 = Demand Transfer DMA: 2 bytes per DMA request.

   Bits 7:6 = 10 = Demand Transfer DMA: 4 bytes per DMA request.

   Bits 7:6 = 11 = Demand Transfer DMA: 8 bytes per DMA request.

3. Clocks and counters: registers 70h, 72h, 74h, and 75h:

   Register 70h = Sample Rate Generator
   Register 72h = Filter Clock Divider
   Registers 74h/75h = Audio 2 Transfer Count Reload register low/high, 2's complement

4. Initialize and Configure DAC: register 7Ah

   Register 7Ah:

   Bit 2: set high for stereo, low for mono.

   Bit 1: set high for signed data, low for unsigned.

   Bit 0: set high for 16-bit samples, low for 8-bit.

5. Enable/Select DMA Channel and IRQ Channel, registers 7Fh, 7A, and 7Dh:

   Register 7Fh – Interrupt Control and Status register. Select the desired interrupt.

   Register 7Ah – Audio 2 Control 2 register.
   Bit 5 enables the DRQx/DACKBx pin pair selected in register 7Dh.

   Register 7Dh – Audio 2 Configuration register.
   Make sure bit 2 is high. Bits 1:0 select the DRQ.

   Bits 1:0 = 00 = DRQA/DACKBA.

   Bits 1:0 = 01 = DRQB/DACKBB.

   Bits 1:0 = 10 = DRQC/DACKBC.

   Bits 1:0 = 11 = DRQD/DACKBD.

6. Configure system interrupt controller and DMA controller
7. To Start DMA:

   Set bits 1:0 of register 78h high.

8. Delay approximately 100 msec to allow the analog circuits to settle, then set the Audio 2 DAC playback volume, register 7Ch.
9. During DMA:

   For auto-initialize, you do not need to send any commands to the ES1887 at interrupt time, except for reading Audio_Base+Eh to clear the interrupt request.

   For normal mode, you must initialize the system DMA controller with the address and count of the next block to transfer. You must update the ES1887 transfer count registers if the count is changed. To start the next transfer, clear bits 1:0 of register 78h, then set the bits high again.

   To stop DMA at any time, clear bit 0 of register B8h. To stop DMA after the current auto-initialize block is finished, clear bit 4 of register 78h, wait for the interrupt, and then clear bits 1:0 of 78h.

10. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in port Audio_Base+Ch to be sure data transfer is completed. Delay 25 msec if possible to let the filter outputs settle to DC level, then disable the Audio 2 DAC input to the mixer.
11. Finally, issue another software reset to the ES1887 to leave things in a well-defined state.

Full-Duplex DMA Mode (No DSP Serial Port)

The ES1887 supports stereo full-duplex DMA. In full-duplex (FD) mode, a second audio channel has been added to the ES1887. This second audio channel is programmed via Mixer registers.

1. Program the first audio channel as is in “Extended Mode ADC Operation” on page 36. Extended mode registers A1h and A2h define the sample rate and filter frequency for both record and playback. In other words, the record and playback must be at the same sample rate (synchronous).
2. Program the second audio channel. Mixer registers 74h and 76h are set to the 2's complement DMA transfer count. The second audio channel supports auto-initialize mode as well as normal mode. The playback buffer in system memory does not have to be

the same size as the record buffer. When the DMA transfer count rolls over to 0, it can generate an interrupt that is independent of the interrupt generated by the first audio channel.

If the record and playback buffers are the same size, then a single interrupt can be used. The DMA Transfer Count Reload registers (A4h, A5h, 74h, and 76h) are programmed with the same value for both channels. The second audio channel should be enabled first, before the record channel. For example, assume there are two half-buffers in a circular buffer. When the record channel completes filling the first half, it will generate an interrupt. We want to be sure that the playback channel is not still accessing the first half at the time of the interrupt. We can guarantee this by starting the playback channel first. It has a 32-word FIFO that will be filled quickly via DMA.

The recommended method is as follows:

Program both DMA controllers for auto-initialize DMA within separate circular buffers of the same size, N.

To exit full-duplex mode, clear bits 0 and 1 of Mixer register 78h.

1. Reset.

   Write 3 to port Audio_Base+6h instead of 1 as in Compatibility mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to Compatibility mode. Reset disables the Audio 1 DAC input to the mixer. This is intended to mask any pops created during the setup of the DMA transfer.

2. After the reset, send command C6h to enable Extended mode commands.

3. Program direction and type: registers B8h, A8h, and B9h:

   Register B8h: Set bit 2 high for auto-initialize DMA mode. Leave bit 3 low to program the CODEC for the DAC direction.

   Register A8h: modify only bits 1 and 0 of this register, read it first to preserve the remaining bits of this register:

   Bits 1:0 = 10 = mono.

   Register B9h:

   Bits 1:0 = 00 = Single Transfer DMA.

   Bits 1:0 = 01 = Demand Transfer DMA: 2 bytes per DMA request.

   Bits 1:0 = 10 = Demand Transfer DMA: 4 bytes per DMA request.

4. Clocks and counters: registers A1h, A2h, A4h, and A5h:

   Register A1h = Sample Rate Clock Divider
   Register A2h = Filter Clock Divider
   Registers A4h/A5h = DMA Counter Reload register low/high, 2's complement.

5. Initialize and Configure DAC: registers B6h and B7h.

   Register B6h: 80h for signed data and 00h for unsigned data. This command sets the register value to zero as the contents of B5h and B6h are unknown. This is to prevent pops.

   Register B7h: Set the data format 16-bit mono. See Table 12, “Command Sequences for DMA Playback,” on page 35.

6. Program transfer type: register 78h:

   Register 78h: Set bit 4 high for auto-initialize DMA mode.

   Bits 7:6 = 00 = Single Transfer DMA.

   Bits 7:6 = 01 = Demand Transfer DMA: 2 bytes per DMA request.

   Bits 7:6 = 10 = Demand Transfer DMA: 4 bytes per DMA request.

   Bits 7:6 = 11 = Demand Transfer DMA: 8 bytes per DMA request.

7. Clocks and counters: registers 70h, 72h, 74h, and 75h:

   Set the Sample rate the same as in A1h. Set the Transfer Count Reload to 64 bytes.

   Register 70h = Sample Rate Generator
   Register 72h = Filter Clock Divider
   Registers 74h/75h = Transfer Count Reload register low/high, 2's complement

8. Initialize and Configure DAC, register 7Ah:

   Bit 2: set low for mono.

   Bit 1: set high for signed data, low for unsigned.

   Bit 0: set high for 16-bit samples.

9. Enable/Select DMA Channel and IRQ Channel, registers 7Fh, 7A, B2h, and 7Dh:

   Register 7Fh – Interrupt Control and Status register.
   Select the desired interrupt.

   Register 7Ah – Audio 2 Control 2 register.
   Bit 5 enables the DRQx/DACKBx pin pair selected in register 7Dh.

   Register B2h – DRQ Configuration register.
   Make sure bits 4 and 6 are high; clear bits 7 and 5.

Register 7Dh – Audio 2 Configuration register.
Make sure bit 2 is high. Bits 1:0 select the DRQ.

Bits 1:0 = 00 = DRQA/DACKBA.

Bits 1:0 = 01 = DRQB/DACKBB.

Bits 1:0 = 10 = DRQC/DACKBC.

Bits 1:0 = 11 = DRQD/DACKBD.

10. Set bit 0 of register 78h. Since the playback FIFO is presumably empty, the value 0 is transferred to the playback DAC at each sample clock. A click or pop may be heard when full-duplex mode is enabled. To prevent this, use command D1 to enable the Audio 1 DAC input to the mixer after a suitable delay (about 25 msec).
11. Enable playback DMA by setting bit 1 of register 78h. After 64 bytes are transferred, bit 7 of 7Ah should go high. Poll this bit with a suitable time-out (for example, 10 msec).
12. After bit 7 of register 7Ah goes high, enable recording by setting bit 7 of register B7h and bit 0 of register B8h.
13. As usual, the first 50 to 100 msec of recorded data should be discarded until analog circuits have settled.

Programming the ES1887 Mixer

The ES1887 has a set of Mixer registers that are backward compatible with the Sound Blaster Pro. However, some of the registers have an “extended” or “alternate” way of accessing the registers to provide for greater functionality.

Commanding the ES1887 Mixer Registers

There are two I/O addresses used by the mixer: Audio_Base+4h is the address port; Audio_Base+5h is the data port. In the Sound Blaster Pro, Audio_Base+4h is write only, while Audio_Base+5h is read/write.

Writing Data to the ES1887 Mixer Registers

To set a mixer register, write its address to Audio_Base+4h, then write the data to Audio_Base+5h.

Reading Data from the ES1887 Mixer Registers

To read a mixer register, write its address to Audio_Base+4h, then read the data from Audio_Base+5.

Resetting the Mixer Registers

The Mixer registers are not affected by software reset. To reset the registers to initial conditions, write any value to mixer register 00h:

1. Write 00h to Audio_Base+4h (select mixer register 00h).
2. Write 00h to Audio_Base+5h (write 00h to the selected mixer register).

Extended Access to SB Pro Mixer Volume Controls

The Sound Blaster Pro Mixer Volume controls are mostly 3 bits per channel. Bits 0 and 4 are always high when read. The ES1887 offers an alternative way to write each Mixer register. Use the “Extended Access” registers for volume control of 4 bits/channel. If the Sound Blaster Pro compatible interface is used, bits 0 and 4 are cleared by a write and forced high on all reads. See Table 14 for a list of Sound Blaster Pro registers and the extended access counterparts.

Table 14 Sound Blaster Pro/Extended Access Registers

Register	Function	Extended Access Register for 4 Bits/Channel
04h	DAC Volume	14h
22h	Master Volume	32h
26h	FM Volume	36h
28h	CD (Aux) Volume	38h
2Eh	Line Volume	3Eh

For example, if you write 00h to Sound Blaster Pro register 04h, you will read back 11h because bits 0 and 4 are “stuck high” on reads. Inside the register, these bits are “stuck low,” so that writing 00h is the same as writing 11h.

If you write or read using address 14h instead of 04h, you have direct access to all 8 bits of this Mixer register.

Extended Access to Mic Mix Volume

If Sound Blaster Compatibility mode register address 0Ah is used to control Mic Mix Volume, only bits 2 and 1 are significant. Bit 0 is stuck high on read and stuck low on writes. Furthermore, this is a mono control, which prevents panning.

For extended access, use register address 1Ah instead. Register 1Ah offers 4-bits/channel for pan control of the mono microphone input to the mixer.

Access to this register via address 0Ah is mapped as follows:

Write to 0Ah	D2=0, D1=0	Mic Mix Volume = 00h
	D2=0, D1=1	Mic Mix Volume = 55h
	D2=1, D1=0	Mic Mix Volume = AAh
	D2=1, D1=1	Mic Mix Volume = FFh
Read from 0Ah	D2 = Mic Mix Volume register bit 3	
	D1 = Mic Mix Volume register bit 2	
	D0 = 1	
	Others are undefined.	

Extended Access to ADC Source Select

In Sound Blaster Compatibility mode in the Sound Blaster Pro mixer, there are three choices for recording source, set by bits 2 and 1 of Mixer register 0Ch. Note that bit 0 is set to 0 upon any write to 0Ch and set to one upon any read from 0Ch:

D2	D1	Source Selected
0	0	Microphone (default)
0	1	CD (Aux) Input
1	0	Microphone
1	1	Line input

For extended access, use register address 1Ch to select recording from the mixer as follows:

D2	D1	D0	Source Selected
x	0	x	Microphone (default)
0	1	x	CD (Aux) input
1	1	0	Line input
1	1	1	Mixer[a]

a. Mixer register 7A bits 4:3 determine if the mixer input source is the playback or record mixer.