ES1946 Solo-1E™
PCI *Audio*Drive® Notebook Solution
Data Sheet

DESCRIPTION

The ES1946 Solo-1E™ PCI *Audio*Drive® solution implements a single-chip PCI audio solution, providing high-quality audio processing while maintaining full legacy DOS game compatibility. With a dynamic range over 80 dB, the ES1946 complies with the Microsoft® PC 98 / PC 99 specifications and meets WHQL audio requirements. The ES1946 forms a complete audio subsystem on a single chip for notebook platforms.

The ES1946 operates at 3.3 volt digital supply and 5 volt analog supply. Combined with its built in power management support, the ES1946 is an ideal audio solution for notebooks.

The ES1946 includes an I²S Zoom Video port interface which allows playback of MPEG audio using the 16-bit stereo music DAC; thus reducing the total audio solution chip count.

The ES1946 incorporates a microcontroller, **ESFM**™ music synthesizer, 16-bit stereo wave ADC and DAC, 16-bit stereo music DAC, MPU-401 UART mode serial port, I²S Zoom Video port, serial port interface to external wavetable music synthesizer, dual game port, hardware master volume control, DMA control logic with FIFO, and PCI bus interface logic. There are three stereo inputs (typically line-in, CD audio, and auxiliary line) and a mono microphone input.

The ES1946 also incorporates *Spatializer*® VBX™ technology, provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc. This processor expands the stereo sound field emitted by two speakers to create a resonant 3-D sound environment.

The ES1946 integrates ESS' field-proven hardware design for DOS game compatibility with hardware FM synthesis (**ESFM**™ synthesis) and three methods for legacy audio control interface: PC/PCI, Distributed DMA, and Transparent DMA. Transparent DMA requires no sideband signals from PC core logic chipsets in addition to the standard PCI 2.2 bus. TDMA is compatible with the Intel® Pentium®, Pentium II®, and Celeron™ chipsets as well as standard PCI add-in cards.

The ES1946 provides a serial EEPROM interface for ease of programming the Subsystem ID and Subsystem Vendor ID.

The ES1946 can record, compress, and play back voice, sound, and music with built-in mixer controls. It supports stereo full-duplex operation for simultaneous record and playback. The **ESFM**™ synthesizer has extended capabilities within native mode operation providing superior sound and power-down capabilities.

The ES1946 is compliant with Advanced Power Management (APM) 1.2, Advanced Configuration and Power Interface (ACPI) 1.0, and PCI Power Management Interface (PPMI) 1.0.

It is available in an industry-standard 100-pin Thin Quad Flat Pack (TQFP) package.

Features

- Single, high-performance, mixed-signal, 16-bit stereo VLSI chip
- 3.3 volt PCI parallel bus interface, revision 2.2
- Full native DOS games compatibility, via three technologies:
  - TDMA
  - DDMA
  - PC/PCI
- High-quality **ESFM**™ music synthesizer
- Dynamic range (SNR) over 80 dB
- Serial EEPROM interface for SID and SVID resource
- Integrated *Spatializer*® 3-D VBX™ stereo audio effects technology provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc.

Record and Playback Features

- Record, compress, and play back voice, sound, and music
- 16-bit stereo ADC and DAC
- Programmable independent sample rates from 6 kHz up to 48 kHz for record and playback
- Full-duplex operation for simultaneous record and playback
- 2-wire hardware volume control for up, down, and mute

Inputs and Outputs

- Stereo inputs for line-in, auxiliary A (CD audio), and auxiliary B, and a mono input for microphone
- I²S Zoom Video port interface for MPEG audio at up to 48 kHz
- MPU-401 (UART mode) interface for wavetable synthesizers and MIDI devices
- Integrated dual game port
- Separate mono input and mono output for speakerphone

Mixer Features

- 7-channel mixer with stereo inputs for line, CD audio, auxiliary line, music synthesizer, digital audio (wave files), and mono inputs for microphone and speakerphone
- Programmable 6-bit logarithmic master volume control

Power

- Advanced power management meets APM 1.2, ACPI 1.0, and PPMI 1.0 standards
- 3.3 volt digital supply operation

Compatibility

- Supports PC games and applications for Sound Blaster™ and Sound Blaster™ Pro
- Supports Microsoft® Windows™ Sound System®
- Meets PC 98 / PC 99 and WHQL specifications

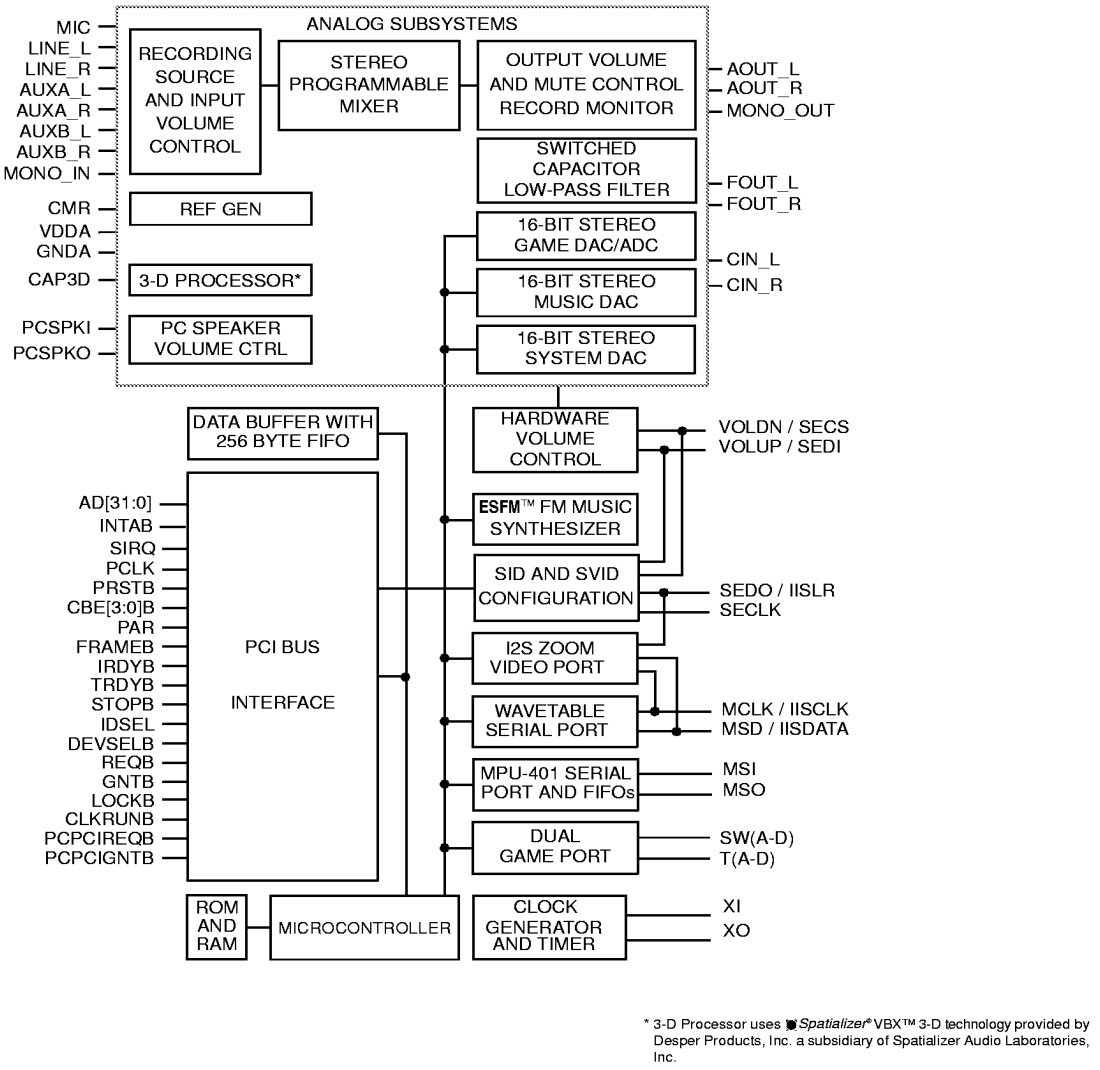


Figure 1 Block Diagram

CONTENTS

FIGURES

TABLES

PINOUT

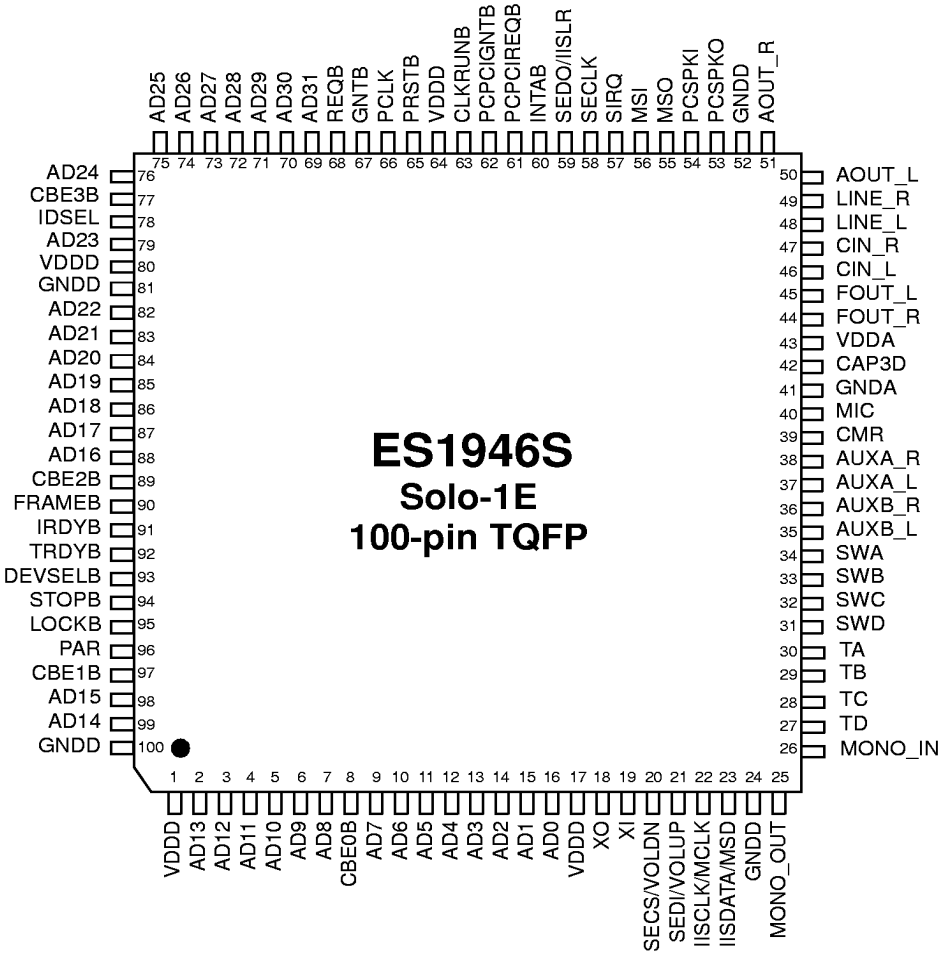


Figure 2 ES1946 Pinout

PIN DESCRIPTION

Name	Number	I/O	Description
VDDD	1,17,64,80	I	Digital supply voltage (3.0V to 3.6V).
AD[31:0]	69–76,79, 82–88,98, 99, 2–7,9–16	I/O	Address and data lines from the PCI bus.
CBE[3:0]B	77,89,97,8	I/O	PCI command/byte enable.
XO	18	O	Oscillator output. Connect to external 14.318 MHz crystal.
XI	19	I	Oscillator/external clock input. Connect to external 14.318 MHz crystal or clock source (must be CMOS levels).
VOLDN	20	I	Active-low volume decrease button input with internal pull-up.
SECS		O	Serial EEPROM chip select.
VOLUP	21	I	Active-low volume increase button input with internal pull-up.
SEDI		O	Serial data output to EEPROM.
MCLK	22	I	Input with internal pull-down. Music serial clock from external wavetable music synthesizer (ES692).
IISCLK		I	Serial shift clock for I²S interface. IIS and hardware wavetable cannot co-exist.
MSD	23	I	Input with internal pull-down. Music serial data from external wavetable music synthesizer (ES692).
IISDATA		I	Serial data for I²S interface. IIS and hardware wavetable cannot co-exist.
GNDD	24,52,81,100	I	Digital ground.
MONO_OUT	25	O	Mono output with source select and volume control (including mute). This pin can drive an external 5k Ω AC load.
MONO_IN	26	I	Mono input to mixer and ADC. This pin has an internal pull-up to CMR.
T(A-D)	30:27	I[a]/O	Joystick timer pins. These pins connect to the X-Y positioning variable resistors for the two joy-sticks. Five volt tolerant pins.
SW(A-D)	34:31	I[a]	Active-low joystick switch setting inputs. These SW pins have internal pull-up resistors. Five volt tolerant pins.
AUXB_L	35	I	Auxiliary B input, left. AUXB_L has an internal pull-up resistor to CMR.
AUXB_R	36	I	Auxiliary B input, right. AUXB_R has an internal pull-up resistor to CMR.
AUXA_L	37	I	Auxiliary A input, left. AUXA_L has an internal pull-up resistor to CMR. Normally intended for connection to an internal or external CD-ROM analog output.
AUXA_R	38	I	Auxiliary A input, right. AUXA_R has an internal pull-up resistor to CMR. Normally intended for connection to an internal or external CD-ROM analog output.
CMR	39	O	Common mode reference voltage (2.25 V ± 5%). Bypass this pin to analog ground with 47 µF electrolytic in parallel with a .1 µF capacitor.
MIC	40	I	Microphone input. MIC has an internal pull-up resistor to CMR.
GNDA	41	I	Analog ground.
CAP3D	42	I	Bypass capacitor to analog ground for 3-D effects.
VDDA	43	I	Analog supply voltage (5 V ± 5%). Must be greater than or equal to VDDD - 0.3 V.
FOUT_R,	44	O	Filter output, right. AC-coupled externally to CIN_R to remove DC offsets.
FOUT_L	45	O	Filter output, left. AC-coupled externally to CIN_L to remove DC offsets.
CIN_L	46	I	ADC and first channel DAC mixer input. This pin has an internal 50k Ω pull-up resistor to CMR.
CIN_R	47	I	ADC and first channel DAC mixer input. This pin has an internal 50k Ω pull-up resistor to CMR.
LINE_L	48	I	Line input, left. LINE_L has an internal pull-up resistor to CMR.

a. 1[a] pins are 5 volt tolerant pins with 3.3 volt digital supply on VDDD pins

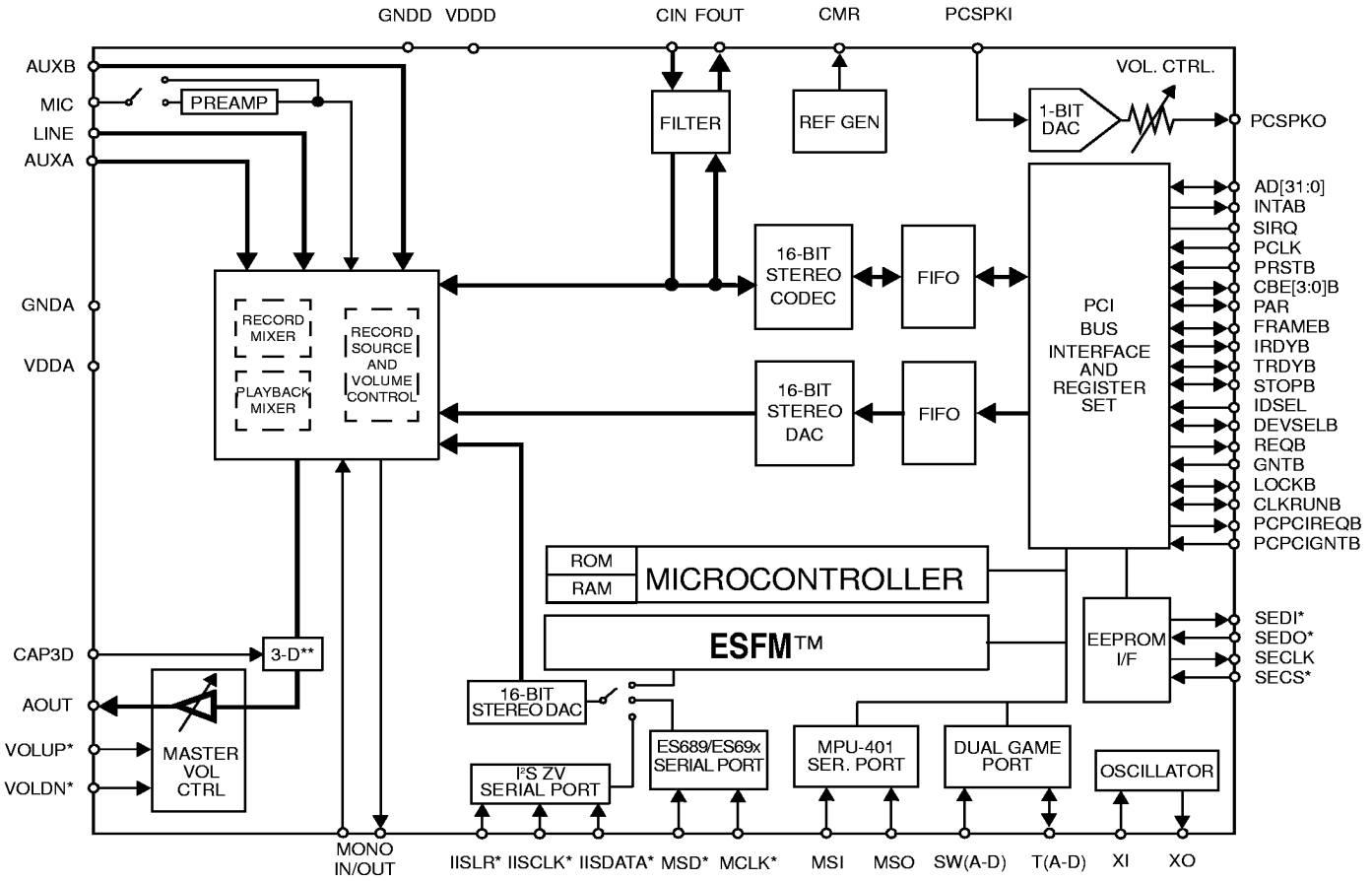
Name	Number	I/O	Description
LINE_R	49	I	Line input, right. LINE_R has an internal pull-up resistor to CMR.
AOUT_L	50	O	Line-level stereo output, left. This pin can drive a 5k Ω AC load.
AOUT_R	51	O	Line-level stereo output, right. This pin can drive a 5k Ω AC load.
PCSPKO	53	O	Analog output of PCSPKI with volume control.
PCSPKI	54	I[a]	Normally low digital PC speaker signal input. This signal is converted to an analog signal with volume control and appears on analog output PCSPKO. Five volt tolerance input.
MSO	55	O	MIDI serial data output.
MSI	56	I[a]	MPU-401 MIDI serial input. Schmitt trigger input with internal pull-up resistor. Five volt tolerance input.
SIRQ	57	I/O	Serialized IRQ.
SECLK	58	O	Clock for serial EEPROM. Pull-up to VDDD via external 47k Ω resistor. Pull-up resistor is required even though EEPROM is not used.
SEDO	59	I	Serial data input from EEPROM.
IISLR		I	Left/Right strobe for I²S interface.
INTAB	60	O	PCI interrupt request.
PCPCIREQB	61	O	PC/PCI serialized DREQ output. (Optional motherboard PC/PCI implementation.)
PCPCIGNTB	62	I	PC/PCI serialized DACK input. (Optional motherboard PC/PCI implementation.)
CLKRUNB	63	I/O	PCI clock state for power management.
PRSTB	65	I	PCI reset.
PCLK	66	I	PCI clock. This clock times all PCI transactions.
GNTB	67	I	PCI busmaster grant.
REQB	68	O	PCI busmaster request, tri-state output.
IDSEL	78	I	PCI device select for configuration.
FRAMEB	90	I/O	PCI cycle frame.
IRDYB	91	I/O	PCI initiator ready.
TRDYB	92	I/O	PCI target ready.
DEVSELB	93	I/O	PCI device select.
STOPB	94	I/O	PCI stop transaction.
LOCKB	95	I/O	PCI lock.
PAR	96	I/O	PCI parity.

a. 1[a] pins are 5 volt tolerant pins with 3.3 volt digital supply on VDDD pins

FUNCTIONAL DESCRIPTION

This section shows the overall structure of the ES1946 and discusses its major functional subunits. The ES1946 operates at 3.3 volt digital supply voltage and 5 volt analog supply voltage. It requires that all digital I/O be at 3.3 volt logic level with the exceptions of PCSPKI, MSI, SW (A-D) and T (A-D) pins which are 5 volt tolerant inputs.

The major subunits of the ES1946 are shown in Figure 3 and described briefly in the following paragraphs.



* Some of these pins are shared with other functions.

** 3-D Processor uses *Spatializer* ® VBX™ 3-D technology provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc.

Figure 3 ES1946 Block Diagram

Digital Subsystems

- **RISC microcontroller** – game-compatible audio functions are performed by an embedded microcontroller.
- **ROM and RAM** – firmware ROM and data RAM to the embedded microcontroller.
- **Oscillator** – circuitry to support an external crystal.
- **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.
- **PCI bus interface** – provides interface to 3.3 volt PCI bus signals. The PCI 2.2 compliant interface supports bus master/slave.
- **Dual game port** – integrated dual game port for two joysticks.
- **MPU-401 serial port** – asynchronous serial port for MIDI devices such as a wavetable synthesizer or a music keyboard input.
- **EEPROM interface** – serial port connection from a 93LC66/46 EEPROM providing Subsystem ID and Subsystem Vendor ID.
- **ESFM™ music synthesizer** – high-quality, OPL3 superset FM synthesizer with 20 voices.
- **Hardware volume control** – 2 pushbutton inputs with internal pull-up devices for up/down/mute that can be used to adjust the master volume control.

  The mute input is defined as the state when both up and down inputs are low simultaneously.

Analog Subsystems

- **Record and Playback Mixers** – seven input stereo mixers. Each input has independent left and right 4-bit volume control:
  - Line In
  - Mic In
  - Aux A (CD-audio)
  - Aux B
  - Digitized audio (wave files)
  - I²S/FM music DAC
  - MONO_IN/MONO_OUT
- **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™**/I²S Zoom video port or **ESFM™**/external wavetable sythesizer.
- **1-Bit DAC** – for PC speaker digital input.
- **3-D Processor** – a 3-D processor using *Spatializer®* VBX™ stereo 3-D audio effects technology, provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc.
- **I²S Zoom Video serial port** – supports sample rates up to 48 kHz for MPEG audio. Cannot co-exist with hardware wavetable.
- **Wavetable serial port** – serial port connection from the output of an ES689 or ES69x that eliminates the need for an external DAC. Cannot co-exist with I² S.
- **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)
  - Record Mixer
- **Output volume and mute control** – The master volume is controlled either by programmed I/O or by volume control switch inputs. The master volume supports 6 bits per channel.
- **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.
- **Pre-amp** – 20 dB microphone pre-amplifier.

MIXER SCHEMATIC BLOCK DIAGRAM

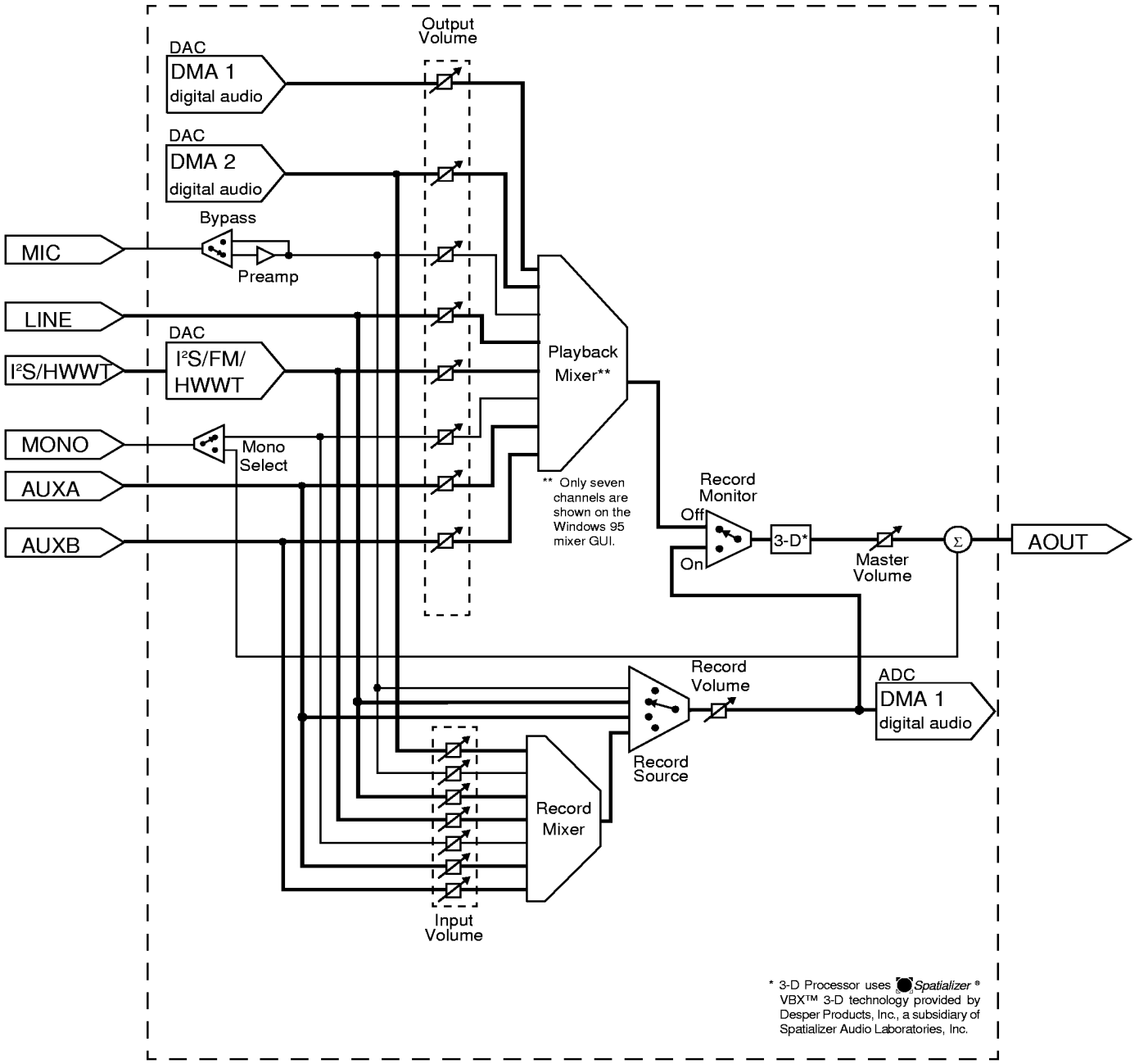
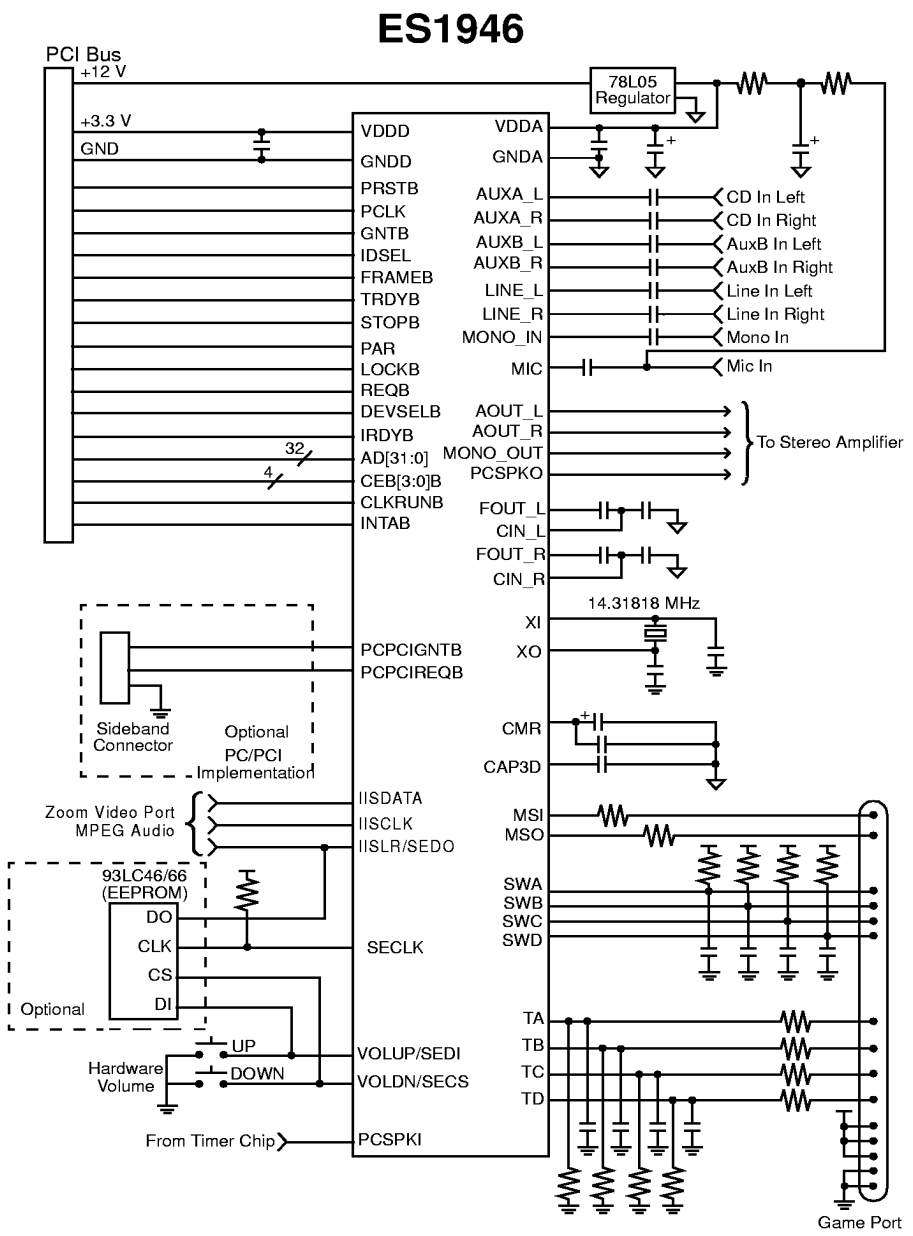


Figure 4 ES1946 Mixer Schematic Block Diagram

TYPICAL APPLICATION



BUS INTERFACING

The ES1946 is compliant with PCI parallel bus interface, version 2.2. This section discusses interfacing to the PC bus, and items relating to configuration for the bus.

Table 1 shows the pins used to interface the ES1946 to the PCI bus.

Table 1 PCI Bus Interface Pins

Pins	Descriptions
AD[31:0]	Address and data lines from the PCI bus.
CBE[3:0]B	PCI command/byte enable.
CLKRUNB	PCI clock state for power management. (optional).
DEVSELB	PCI device select.
FRAMEB	PCI cycle frame.
GNTB	PCI busmaster grant.
IDSEL	PCI device select for configuration.
INTAB	PCI interrupt request.
IRDYB	PCI initiator ready.
LOCKB	PCI lock.
PAR	PCI parity.
PCLK	PCI clock. This clock times all PCI transactions.
PCPCIGNTB	PC/PCI serialized DACK input. (Motherboard implementation).
PCPCIREQB	PC/PCI serialized DREQ output (Motherboard implementation).
PRSTB	PCI reset.
REQB	PCI busmaster request, tri-state output.
STOPB	PCI stop transaction.
TRDYB	PCI target ready.

DIGITAL AUDIO

DMA Simulation Modes

Native PCI mode or PCI Bus Master DMA is used to simulate DMA when the ES1946 is configured as a native PCI device under Windows. See “Programming for Native PCI Audio” on page 25 for more information.

Legacy mode is used for native DOS or DOS box applications. The ES1946 uses TDMA, DDMA, or PCPCI to simulate DMA. See “Programming for DOS Game Compatibility” on page 24 for more information. Once the ES1946 is programmed, the Legacy mode used is transparent to an application using DMA.

Audio Channels

The ES1946 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.

In DOS mode, this channel uses TDMA, DDMA, or PCPCI to emulate ISA DMA on the PCI bus. Since most DOS games default to DMA channel 1, the first audio channel should ideally be assigned to ISA channel 1. However, it is possible to map audio 1 to one of three DMA channels (0,1,3) through the PCI configuration registers.

In Windows mode, this channel can use PCI Bus Master DMA. No ISA resources are required.

Audio 2 The second audio channel. This channel is used for audio playback in full-duplex mode. Audio 2 uses PCI Bus Mastering with burst transfers to minimize PCI bus access.

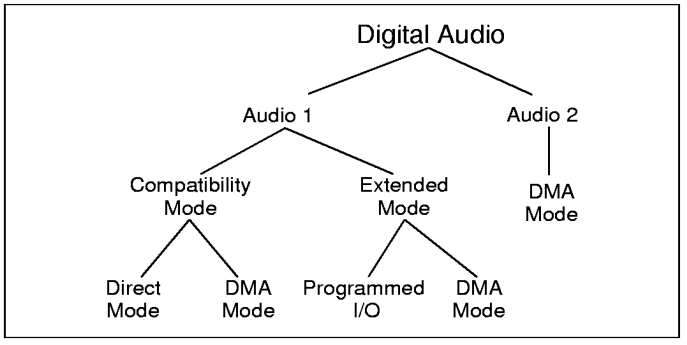


Figure 5 Data Transfer Modes

Data Formats

See “Data Formats” on page 26.

Audio 1 DMA Transfers in legacy DOS Compatibility Mode

The first audio channel is programmed using standard commands which support Sound Blaster functionality. These commands are written to the chip through port SBBase+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, 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 (SBBase+Ch [bit 7]) is required before writing the command and between writing 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. Do this by programming the timer with command 40h just as if the application were using DMA mode.

DMA Modes

In DMA mode, the programmable timer in the ES1946 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 ES1946. The ES1946 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 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 ES1946 with the same size information. At the end of the transfer, the ES1946 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 ES1946 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 ES1946 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 ES1946 generates an interrupt for the transition between buffer halves. In this mode the DMA controller and ES1946 only need to be set up once.

High-Speed Mode

The ES1946 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.

The special “High-Speed mode” allows 8-bit sampling up to 44 kHz for ADC, using commands 98h (auto-initialize) and 99h (normal). No automatic gain control (AGC) is performed. The input volume is controlled with command DDh.

Audio 1 DMA Transfers in Extended Mode

The first audio channel is programmed using the controller registers internal to the ES1946. The commands written to the controller registers are written to the chip through port SBBase+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 ES1946 port SBBase+Fh for block transfers.

I/O transfers to FIFO are nearly identical to the DMA process, except that an I/O access to port SBBase+Fh replaces the DMA cycle. For details about programmed I/O operation see “Extended Mode Programmed I/O Operation” on page 33.

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.

Single Transfer

One byte is transferred per DMA request.

Demand Transfer

To reduce the number of DMA requests necessary to make a transfer, two or four bytes are transferred per DMA request (DRQ). Using Demand transfer enables 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 13.

Extended Mode Audio 1 Controller Registers

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

Table 2 Extended Mode Audio 1 Controller Registers

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
B7h	Audio 1 Control 1 register
B8h	Audio 1 Control 2 register
B9h	Audio 1 Transfer Type register

Audio 1 DMA Transfers in Native PCI Mode

Unlike DOS game environments, the ES1946 can be completely controlled by drivers. This way the ES1946 can perform Bus Master DMA for first channel audio data transfers under Windows (or other operating systems).

Data Transfers Using the Second Audio Channel

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

The second audio channel always uses PCI Bus Master transfers instead of ISA-like DMA. IOBase+0h – IOBase+6h control Audio 2 Bus Master DMA. Both normal and auto-initialize modes are available, as in ISA-DMA. DMA counts for Audio 2 must be in multiples of 16 bytes, so that the ES1946 can performs 4 DWord burst transfers.

Audio 2 Related Mixer Registers

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

Table 3 Audio 2 Related Mixer Registers

Address	Name
70h	Audio 2 Sample Rate register
71h	Audio 2 Mode 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

First DMA Channel CODEC

The CODEC of the first audio channel cannot perform stereo DAC and ADC simultaneously. It can either be a stereo DAC or a stereo ADC. 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 sent to the mixer. After reset, input to the mixer from the first audio channel DAC is muted to prevent pops. The ES1946 maintains a status flag to determine if the input to the mixer from the first audio channel DAC is enabled or disabled. Command D8h returns the status of the flag (0h=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 milliseconds for the analog circuitry to settle before enabling the voice channel with command D1h.

Pop sounds may still occur 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 ES1946, shown in Table 4.

Table 4 ES1946 Interrupt Sources

Interrupt Source	Description
Audio 1	An interrupt used for the first DMA channel (Sound Blaster compatible DMA, Extended mode DMA, and Extended mode programmed I/O), as well as MIDI receive that supports Sound Blaster functionality. Controller register B1h controls use of this interrupt for Extended mode DMA and programmed I/O. This interrupt request is cleared by hardware or software reset, or an I/O read from port SBBase+0Eh. The interrupt request can be polled by reading from port SBBase+0Ch or IOBase+07h.
Audio 2	An optional interrupt for the second DMA channel. The ES1946 can operate in full-duplex mode using two DMA channels. The Audio 2 interrupt is masked by bit 6 of mixer register 7Ah. It can be polled and cleared by reading or writing bit 7 of register 7Ah.
Hardware Volume	Hardware volume activity interrupt. This interrupt occurs when one of the two hardware volume controls changes state. Bit 1 of mixer register 64h is the mask bit for this interrupt. The interrupt request can be polled by reading bit 3 of register 64h. The interrupt request is cleared by writing any value to register 66h.
MPU-401	The MPU-401 interrupt occurs when a MIDI byte is received. It goes low when a byte is read from the MIDI FIFO and goes 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. The MPU-401 interrupt is masked by bit 6 of mixer register 64h.

Interrupt Status Register

Port IOBase+7h of the configuration device can be read to quickly find out which ES1946 interrupt sources are active. The bits are:

Table 5 Interrupt Status Bits in IOBase+7h

Bit	Description
4	Audio 1 interrupt request
5	Audio 2 interrupt request AND'ed with bit 6 of mixer register 7Ah
6	Hardware volume interrupt request AND'ed with bit 1 of mixer register 64h
7	MPU-401 receive interrupt request AND'ed with bit 6 of mixer register 64h

Interrupt Mask Register

Port IOBase+7h can be used to mask any of the four interrupt sources, with the exception of the Audio 1 interrupt. The Audio 1 interrupt request can be enabled by bit 4 of IOBase+7h, or when bit 15 of PCI Configuration register 40h (Legacy Audio Control register) is 0.

The mask bits can be used to force the interrupt source to be zero, without putting the interrupt pin in a high-impedance state. Each bit is AND'ed with the corresponding interrupt source. This register is set to all zeros by hardware reset.

The Interrupt Status register (ISR) is not affected by the state of the Interrupt Mask register (IMR). That is, the ISR reflects the status of the interrupt request lines before being masked by the IMR.

The IMR is useful because the ES1946 shares interrupts. For example, assume that Audio 1, Audio 2, Hardware Volume, and MPU-401 all share the same interrupt in Windows. When returning from Windows to DOS, the Hardware Volume, MPU-401, and Audio 2 interrupts can be masked by setting the appropriate bits to 0.

A second use is within an interrupt handler. The first thing the interrupt handler can do is mask all the interrupt sources mapped to the interrupt handler. Then, the ISR can be polled to decide which sources to process. Just before exiting the interrupt handler, the IMR can be restored. If an unprocessed interrupt remains active, it generates an interrupt request because the interrupt pin was de-asserted during the masked period and then was asserted when the interrupt sources were unmasked. Also, while the interrupts are masked, the individual interrupt sources can change state any number of times without generating a false interrupt request.

PERIPHERAL INTERFACING

I²S Serial Interface

Three input pins, IISDATA, IISCLK, and IISLR, are used for a serial interface between an external device and the stereo music DAC within the ES1946. IISDATA, IISCLK, and IISLR can be left floating or connected to ground if the serial interface is not used.

A typical applications of the I²S serial interface is MPEG audio or digital CD audio.

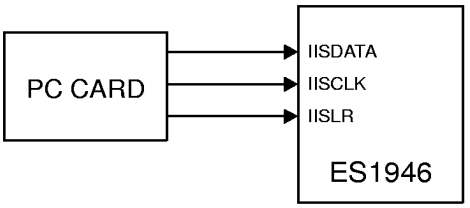


Figure 6 I²S Implementation in ES1946

Table 6 I²S Interface Pins

Pin	Description
IISDATA	Serial data for I²S interface. This pin has an internal pull-down to GNDD.
IISCLK	Serial shift clock for I²S interface. This pin has an internal pull-down to GNDD.
IISLR	Left/Right signal for I²S interface. This pin has an internal pull-down to GNDD.

I²S Serial Interface Software Enable

Bit 0 of mixer register 7Fh enables the data bus connection to the I²S interface.

I²S Serial Interface Timing

This section discusses the I²S serial interface signals. The signals when the port is configured for use with an ES689/ES69x wavetable synthesizer are defined in the Wavetable Interface section.

Three signals are used for I²S:

IISCLK The shift clock. The maximum rate is 6.4 MHz. The minimum number of IISCLK periods per IISLR period is 32. Any number greater than or equal to 32 is acceptable.

IISLR Sample synchronization signal. The maximum sample rate is 50 kHz.

IISDATA Serial data.

Within the ES1946, IISLR and IISDATA are sampled on the rising edge of IISCLK. See Figure 18 and Figure 19 for detailed I²S timing.

Wavetable Interface

The ES1946 contains a synchronous serial interface for connection to a wavetable music synthesizer.

Table 7 Wavetable Interface Pins

Pin	Description
MCLK	Serial clock from external ES689/ES69x music synthesizer. Input with pull-down.
MSD	Serial data from external ES689/ES69x music synthesizer. When both MCLK and MSD are active, the stereo DACs normally used by the FM synthesizer are acquired for use by the external ES689/ES69x. The normal FM output is blocked. Input with pull-down.

Joystick / MPU-401 Interface

MPU-401 UART Mode

There is one MIDI interface in the ES1946, 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.

MPU-401 requires an interrupt channel for MIDI receive. This interrupt should be selected using mixer register 64h. It should be different than the interrupt selected for audio DMA interrupts.

Joystick / MIDI External Interface

The joystick portion of the ES1946 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. If the system already has a game card or port, remove the game card.

If multiple joysticks are required, use a joystick conversion cable. This cable uses a 15-pin D-sub male connector on one end, and two 15-pin D-sub female connectors on the other end. All signals on this cable have direct pin-to-pin connection, except for pins 12 and 15. On the male connector, pins 12 and 15 should be left without connection. On the female connectors, pin 15 is internally connected to pin 8, and pin 12 is internally connected to pin 4. The dual joystick and MIDI port take up only one slot in the system, leaving room for other cards. Figure 7 shows the dual joystick/MIDI connector configuration.

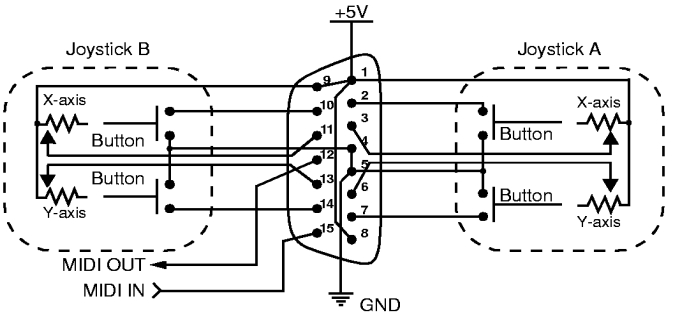


Figure 7 Dual Joystick/MIDI Connector

Figure 8 shows the MIDI serial interface adaptor from the joystick/MIDI connector.

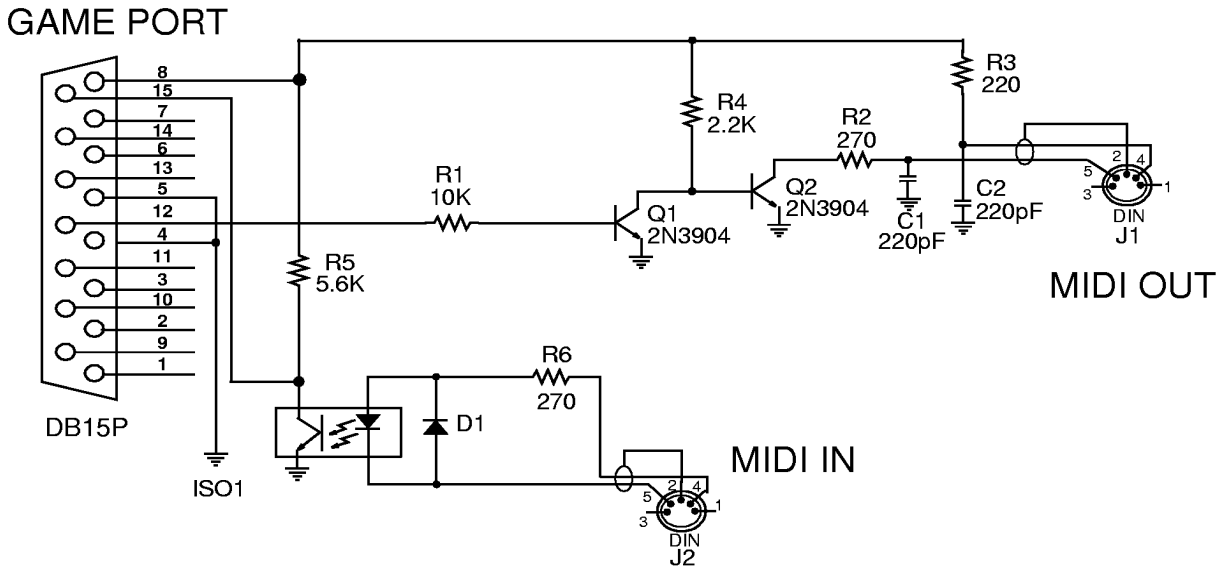


Figure 8 MIDI Serial Interface

MONO_IN and MONO_OUT

MONO_IN is a line-level analog input. MONO_IN is an input to the playback mixer and the record mixer. The mixer volumes are controlled by mixer registers 6Dh (playback) and 6Fh (record).

Alternately, MONO_IN can be mixed with AOUT_L and AOUT_R after the master volume stage. Bit 0 of mixer register 7Dh, when high, enables MONO_IN to be mixed directly (unity gain) with AOUT_L and AOUT_R.

MONO_OUT is a line-level mono output. During power-down or during opamp calibration, MONO_OUT is held at CMR (as are AOUT_L and AOUT_R) by an internal, high-impedance resistor divider. MONO_OUT can be selected from among four sources by bits 2 and 1 of mixer register 7Dh.

Mixer Register 7Dh		MONO_OUT Source
Bit 2	Bit 1	
0	0	Mute (CMR)
0	1	First channel filter output (actually CIN_R pin)
1	0	Second channel DAC, right channel
1	1	Mono mix of record level stage outputs

Normally bits 2:1 are both zero, so that MONO_OUT is muted.

When bit 2 is 0 and bit 1 is 1, MONO_OUT is a buffered version of input pin CIN_R. CIN_R is typically the right channel DAC output, filtered by the first channel switched-capacitor filter. If the right channel is used for ADC, CIN_R will be the right channel ADC input. MONO_OUT can be used in this application as digitized audio playback through the first channel DMA, right channel DAC.

When bit 2 is 1 and bit 1 is 0, MONO_OUT is a buffered version of the second channel, right channel DAC. In this case, the second channel DMA can play digitized audio through MONO_OUT.

When bit 2 is 1 and bit 1 is 1, MONO_OUT is a buffered version of a mono mix of the record level stage left and right outputs. This gives the utmost flexibility in the source or sources of MONO_OUT. The record source select and record levels can be programmed to generate any combination of sources and volumes for MONO_OUT.

Spatializer® VBX™ Audio Processor

The ES1946 incorporates an embedded *Spatializer®* VBX™ stereo audio processor provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc. It is positioned between the output of the playback mixer and the master volume controls and it produces a wider perceived stereo effect.

The 3-D effect is enabled by register 50h bit 3. The amount of effect is controlled by directly programming 3-D Effect Level register 52h.

Hardware and Master Volume Control

Two external pins, VOLUP and VOLDN, can be connected to external momentary switches to ground to implement hardware master volume controls. Pressing one of these buttons produces a low signal to one of the inputs and thereby changes the master volume.

MUTE is emulated by the state where both VOLUP and VOLDN inputs are low simultaneously.

The up and down buttons produce a single step change in volume when they are first pressed. If these buttons are held down, they enter a fast-scrolling mode. The single step change can be either one volume unit (.75 dB) or three volume units (2.25 dB). In scrolling mode, the step change is always one volume unit.

The two inputs have debounce circuitry within the ES1946. Hold each input low for 40 milliseconds or more for it to be recognized as a valid button press. Hold each input high for 40 milliseconds or more between button presses. A software option allows the debounce time to be reduced from 40 milliseconds to 10 microseconds.

Normally the hardware volume controls directly change the master volume registers and produce an interrupt at each change. However, the ES1946 can be programmed so that the hardware volume controls do not directly change the master volume registers. This is called “split mode”, in which the hardware volume control counters are split from the master volume registers. Pressing a hardware volume control button changes the hardware volume counters and produces an interrupt. The host software can read the hardware volume counters and update the master volume registers as needed. Split mode is enabled by bit 7 of mixer register 64h.

For support of mixer master volume control, a write to mixer registers 22h or 32h translates automatically into writes to the master volume registers. Since register 22h only has 3-bit resolution per channel, and register 32h only has 4-bit resolution per channel, a translation circuit is included in the ES1946 that translates 3- or 4-bit volume

values into the 6-bit volume + mute that is used in the master volume registers. Support of these mixer registers can be defeated under software control.

Reading master volume registers 22h or 32h also requires a translation circuit to translate 6-bit + mute master volumes into 3- or 4-bit master volume numbers for registers 22h or 32h.

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 power is enabled. The value of the resistor is selected from among 7 choices to control the amplitude of the output signal.

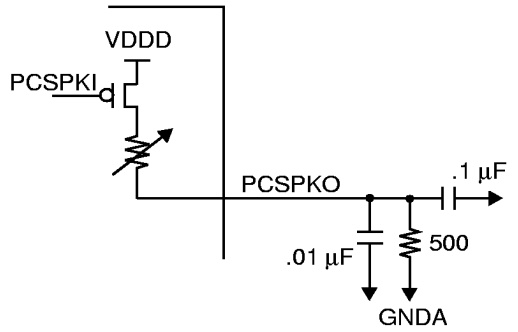


Figure 9 PC Speaker Volume Circuitry

With the external circuit shown in Figure 9, the amplitude of a square wave output on pin PCSPKO should be approximately VDDD/2 for maximum volume, i.e., the internal resistor is approximately 500 ohms (± 30%). The other levels are relative to this amplitude as follows:

mute, -24dB, -21dB, -18dB, -15dB, -12dB, -9dB, -6dB

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.

Serial EEPROM Interface

The ES1946 gets Subsystem ID (SID) and Subsystem Vendor ID (SVID) from an external EEPROM device. The external EEPROM is accessed by the ES1946 immediately after the PRSTB pin becomes inactive. The ES1946 supports 93LC46 and 93LC66 serial EEPROMs.

The EEPROM interface is shared with the hardware volume controls. When the EEPROM interface is active, the volume controls are deactivated. See Figure 10.

EEPROM ROM FORMAT

The first byte should be the EEPROM identifier

Device	1st Byte
93LC46	46h
93LC66	66h

The second byte is the signature byte for the ES1946. The signature byte must be 38h.

Example EEPROM Code

```
; SVID.DAT
;
; Example of EEPROM (93LC46) Data
;
; SVID = 1766h        Sub-system Vendor ID (HC Corp)
; SID  = 1480h        Sub-system ID (Model VC1480)
;
46H, 38H              ; Signature for 93LC46/1946
14H, 80H              ; SID
17H, 66H              ; SVID
;
; End of SVID.DAT
```

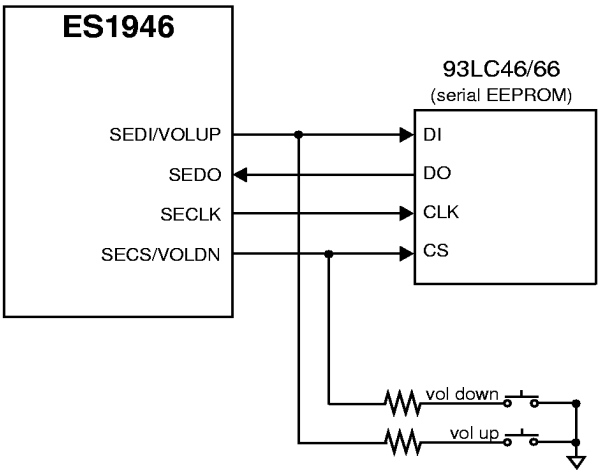


Figure 10 Serial EEPROM – Typical Application

Programming the EEPROM

After reset, the ES1946 reads six bytes from the EEPROM. The ES1946 can read, write, or erase all the bits stored in the EEPROM. The ES1946 accesses the EEPROM by means of three I/O ports.

IOBase+2Dh	Device Select
IOBase+2Eh	EEPROM Data
IOBase+2Fh	EEPROM Command

EEPROM Access Example

```
/* This example illustrates usage of EEPROM i/f */

#define READ  0x8
#define EWEN  0x3
#define EWDS  0x0
#define WRITE 0x4
#define WRAL  0x1
#define ERASE 0xc
#define ERAL  0x2

/* select device type.  must be called before other routine is
called, if EEPROM didn't contain right Device Identifier (first
byte) at boot time. */
void select_eeprom (int dev)
{
  switch (dev) {
        case 0x46:
     outp (iobase+0x2d, 0x01);
     break;
        default:
     outp (iobase+0x2d, 0x03);
     break;
  }
  pause (2);
}

/* n bytes write to EEPROM */
void eepromWrite (int eeaddr, int *eedata, int n)
{
   int i;
   unsigned int index;
   char line[10];

   outp (iobase+0x2f, EWEN); //enable erase/write
   pause (1);
   gotoAddr (eeaddr);

   for (i=0; i<n; i++) {
     outp (iobase+0x2e, *eedata++);
     outp (iobase+0x2f, WRITE);
     pause (10);            // wait 10 msec
   }

   outp (iobase+0x2f, EWDS); //disable erase/write
}

/* n bytes read from EEPROM */
void eepromRead (int eeaddr, int *eedata, int n)
{
   int i;
   int *p;

   gotoAddr(eeaddr);

   for (i=0, p=eedata; i<n; i++) {
     outp(iobase+0x2f, READ);
     pause (1);
     *p++ = inp (iobase+0x2e);
     pause (1);
   }
}

/* erase all data */
void eepromEraseAll(void)
{
   int i;

   outp(iobase+0x2f, EWEN); //enable erase/write
   pause(1);
   outp(iobase+0x2f, ERAL);
   pause(15); //wait 15 msec
      outp(iobase+0x2f, EWDS); //disable erase/write
}

void gotoAddr (int addr)
{
   int i;

   inp (iobase+0x2f); //reset eeprom address
   pause (1);
   for (i=0; i<addr; i++) {
     outp (iobase+0x2f, READ);
     pause (1);
     (void) inp (iobase+0x2e);
     pause (1);
   }
}
```

ANALOG DESIGN CONSIDERATIONS

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

MONO_IN and MONO_OUT

MONO_IN and MONO_OUT are bound by digital signals on either side. When laying out a PCB, MONO_IN and MONO_OUT must be in the analog ground plane and well isolated from the digital ground plane.

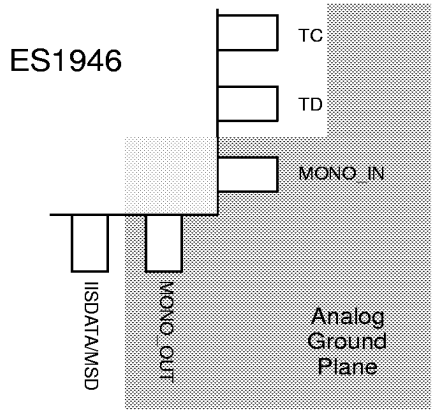


Figure 11 Analog Ground Plane Diagram

Reference Generator

Reference generator pin CMR is shown bypassed to analog ground.

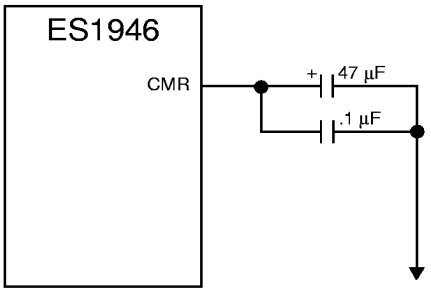


Figure 12 Reference Generator Pin Diagram

Switched-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.

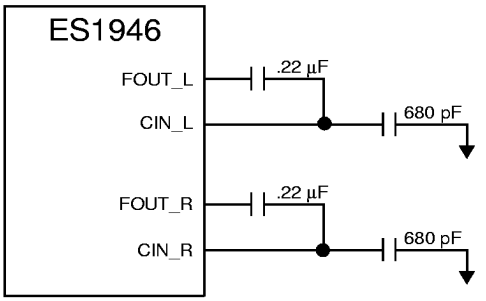


Figure 13 Switched-Capacitor Filter Pin Diagram

Audio Inputs and Outputs

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

ES1946 analog outputs MONO_OUT, AOUT_L and, AOUT_R should be AC-coupled to an amplifier, volume control potentiometer, or line-level outputs.

PROGRAMMING THE ES1946

Identifying the ES1946

The ES1946 device can be identified using the PCI standard configuration register. Offset 00h contains the registered Vendor ID (VID), which for ESS is 125Dh. Offset 02h contains the assigned Device ID (DID), which for the ES1946 is 1969h.

In addition, the ES1946 has a Subsystem Vendor ID (SVID) and Subsystem ID (SID). These two registers default values are read from the EEPROM immediately after PRSTB becomes inactive. If the EEPROM signature check fails (if no EEPROM is connected, for example), the default values become 125Dh and 8889h respectively. If the ES1946 is mounted on the motherboard, the system BIOS can program these two registers so that the EEPROM is not needed.

Resetting and Initializing the ES1946

The ES1946 chip can be reset in one of two ways: hardware or software reset.

The hardware reset signal comes from the PCI bus and it initializes:

- PCI configuration registers
- the microcontroller
- internal FIFOs
- the ESFM synthesizer
- the mixer registers
- the analog mixer
- CODECs

The software reset is controlled by bit 0 of port SBBase+6h and it initializes:

- the microcontroller
- the ESFM synthesizer
- the analog mixer
- the CODEC

To reset the ES1946 by software:

1. Set bit 0 of SBBase+6h.
2. Delay a short period by reading back SBBase+6h.
3. Clear bit 0 of SBBase+6h.
4. In a loop lasting at least 1 millisecond, poll port SBBase+Eh bit 7 for Read Data Available. Exit the loop only if bit 7 is high and SBBase+Ah returns 0AAh.

Table 8 Hardware and Software Reset Initializations

Initialization	Hardware Reset	Software Reset
Disable Extended Mode	Yes	Yes
Reset the timer divider and filter for 8 kHz	Yes	Yes
Stop any DMA transactions in progress	Yes	Yes
Clear any active interrupt requests	Yes	Yes
Disable voice input of mixer	Yes	Yes
Compatibility/Extended mode DMA counters to 2048 bytes	Yes	Yes
Set audio 1 CODEC direction to DAC	Yes	Yes
Set DAC volumes to mid-level	Yes	Yes
Set input volume for 8-bit recording with AGC to maximum	Yes	Yes
All other mixer registers to default values	Yes	No
Internal FIFO in extended mode	Yes	No[1]
PCI Configuration registers	Yes	No

1. Bit 1 of SBBase+6h can reset internal FIFOs.

After the system powers up, the PCI BIOS initializes the header portion (00h–3Fh) of the PCI configuration space. The PCI BIOS can also program the SID and SVID registers.

After the PCI BIOS initializes the device, the ES1946 is in Native mode. All of the PCI configuration registers should be set up properly before the ES1946 accesses any other registers.

Programming for DOS Game Compatibility

The ES1946 is fully compatible to legacy DOS games with Sound Blaster Pro functionality. To achieve high compatibility to a legacy ISA device on the PCI bus, two major issues need to be addressed. The first issue concerns ISA DMA and the second issue concerns ISA IRQ.

ISA DMA

To emulate ISA DMA on the PCI bus, the ES1946 can employ three different protocols:

TDMA	Transparent DMA, a chipset independent mechanism
DDMA	Distributed DMA, must be supported by the PCI chipset
PC/PCI DMA	PC to PCI DMA, must be supported by the PCI chipset

Once one of the three DMA protocols is set up, the ES1946 is seen as an ISA device.

TDMA

In TDMA, the ES1946 snoops PCI bus transactions to a legacy DMA controller device then performs a PCI bus master transaction to complete DMA.

DDMA

In DDMA, the central resource (PCI chipset) includes a DMA remap engine. All transactions to legacy DMACs are remapped to each client (such as the ES1946) by the remap engine. The ES1946 then performs a PCI bus master transaction.

PC/PCI DMA

In PC/PCI DMA, the central resource (PCI chipset) performs PC/PCI cycles, which use sideband signals to the standard PCI bus. The ES1946 then acts as a slave device during DMA.

ISA IRQ

The second issue concerns ISA IRQ. The ISA IRQ is edge triggered while PCI IRQ is level sensitive. By configuring the IRQ policy bits in PCI Configuration register 50h, the ES1946 can emulate ISA IRQ.

Setting bit 15 of Legacy Audio Control register (LACR, PCI Configuration register 40h) to 0, allows the ES1946 to decode legacy audio addresses.

Selecting DMA/IRQ Policy

Because PCI chipsets do not all support the same DMA protocols, DMA policy should be selected according to the chipset in use. Table 9 and Table 10 list the recommended DMA policies for Intel chipsets for add-on cards and motherboards respectively. To find out which DMA policy to use with non-Intel chipset, contact your ESS FAE. DMA policy is configured in PCI Configuration register 50h, bits [10:8].

Table 9 ISA DMA Policy Bits for Add-On Cards

Chipset	Protocol	DMA Policy		
		Bit 10	Bit 9	Bit 8
Intel 430FX (Triton)	TDMA	0	0	1
Intel 430HX (Triton-2)	TDMA	0	0	1
Intel 430VX (Triton-3)	TDMA	0	0	1
Intel 430TX	DDMA	0	0	0
Intel 440LX/EX	TDMA	1	0	0
Intel 440BX	DDMA	0	0	0
	TDMA	1	0	0

Table 10 ISA DMA Policy Bits for Motherboards

Chipset	Protocol	DMA Policy		
		Bit 10	Bit 9	Bit 8
Intel 430FX (Triton)	TDMA	0	0	1
Intel 430HX (Triton-2)	TDMA	0	0	1
Intel 430VX (Triton-3)	TDMA	0	0	1
Intel 430TX	DDMA	0	0	0
Intel 440LX/EX	PCPCI	0	1	0
	TDMA	1	0	0
Intel 440BX	PCPCI	0	1	0
	DDMA	0	0	0
	TDMA	1	0	0

To emulate ISA IRQ on the PCI bus, program the IRQ emulation policy bits, unless SERIRQ is used. Table 11 lists the program IRQ Policy bits. Program the policy bits according to the IRQ level selected for the ES1946's INTAB pin. IRQ policy is configured in PCI Configuration register 50h, bits [14:13].

Table 11 ISA IRQ Emulation Policy Bits

PCI IRQ (INTAB Pin)	IRQ Policy	
	Bit 14	Bit 13
IRQ 5/7/9/10	0	1
IRQs other than IRQ 5/7/9/10	1	0

PCI Configuration register 3Ch is set by the PCI BIOS, and it indicates which IRQ the ES1946 INTAB pin is using. PCI Configuration register 40h (Legacy Audio Control register), bits[9:8] indicate which IRQ is used by the DOS game.

Programming for Native PCI Audio

When the ES1946 is configured as a native PCI device, the audio channels must be configured properly, as shown in the procedures below. The ES1946 is a bus master device.

Configuring Audio 1 in Native Mode

1. Set up the DDMA Control register.
   The PCI BIOS reserves I/O regions (16 bytes), and stores them in the VCBase register. Copy the VCBase register (PCI Configuration register 18h) to the DDMA Control register (PCI Configuration register 60h), with bit 0 set.
2. Select the DMA/IRQ policy.
   The DMA policy should be set for DDMA (PCI Configuration register 50h, bits 10:8 = 000). The ISA IRQ emulation should be disabled (PCI Configuration register 50h, bits 14:13 = 00).
3. Program the legacy DOS compatible module.
   See “Extended Mode Audio 1 DAC Operation” on page 30.
4. Program the DMA controller.
   Instead of using 8237 (DMAC), the program should use the built-in DMAC inside the ES1946. The built-in DMAC is accessed as I/O address range DDMABase+0h to DDMABase+Fh. Use the following sequence to program the built-in DMAC.
   1. Master reset.
      Write any data to DDMABase+Dh.
   2. Mask DMA.
      Write 1 to DDMABase+Fh.
   3. Set up DMA mode.
      Write the mode value to DDMA Base+Bh.
   4. Setup DMA base address and counts.
      Write the base address to DDMABase+0h.
      Write the base count to DDMABase+4h.
   5. Unmask DMA.
      Write 0 to DDMABase+Fh.

Configuring Audio 2 in Native Mode

Unlike programming for Audio 1 in native mode, all I/O spaces are allocated by PCI BIOS.

1. Program the legacy ESS *Audio*Drive® controller module.
   See “Second Audio Channel DAC Operation” on page 33.
2. Program the DMA controller.
   Instead of using 8237 (DMAC), the program should use the built-in DMAC inside the ES1946. The built-in DMAC is accessed as I/O address range IOBase+0h to IOBase+6h. Use the following sequence to program the built-in DMAC.
   1. Disable DMA.
      Clear bit 1 of IOBase+6h. Preserve all other bits.
   2. Set up DMA mode.
      Program bit 3 of IOBase+6h. Preserve all other bits.
   3. Set up DMA base address and counts.
      Write the base address to IOBase+0h.
      Write the base count to IOBase+4h.
   4. Enable DMA.
      Set bit 1 of IOBase+6h. Preserve all other bits.

Modes of Operation

The ES1946 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 ES1946 Enhanced Mixer is described later in this document. See “Programming the ES1946 Mixer” on page 35.

Compatibility Mode Description

The first mode, Compatibility mode, is compatible to legacy DOS. This is the default mode after reset. In this mode, the ES1946 microcontroller is an intermediary in all functions between the PCI bus and the CODEC. The ES1946 microcontroller performs limited FIFO functions using 64 bytes of internal memory.

Extended Mode Description

The ES1946 also supports an Extended mode of operation. In this case, a 256-byte FIFO is used as an intermediary between the PCI bus and the ADC and DAC Control registers, and various Extended mode controller registers are used for control. The ES1946 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 ES1946 commands.

Table 12 Comparison of Operation Modes

	Compatibility Mode (Legacy DOS)	Extended Mode
Legacy DOS compatible	Yes	No
FIFO Size	64 bytes (firmware managed)	256 bytes (hardware managed)
Mono 8-bit ADC, DAC	Yes, to 44 kHz	Yes, to 48 kHz
Mono 16-bit ADC, DAC	Yes, to 22 kHz	Yes, to 48 kHz
Stereo 8-bit DAC	Yes, to 22 kHz	Yes, to 48 kHz
Stereo 8-bit ADC	Yes, to 22 kHz	Yes, to 48 kHz
Stereo 16-bit DAC	Yes, to 11 kHz	Yes, to 48 kHz
Stereo 16-bit ADC	No	Yes, to 48 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	Depends on XTAL

Mixing Modes Not Recommended

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

Data Formats

This section briefly describes the different audio data formats used by the ES1946.

Compressed Data Formats

The ES1946 supports ADPCM in compressed sound DAC operations.

ADPCM is transferred only 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.

Legacy DOS Compatible 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 0h to 0FFh, with the DC-levels around 80h.

16-bit samples are unsigned, ranging from 0000h to 0FFFFh, with the DC-levels 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 set bit 1 of mixer register 0Eh 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 it were for 44 kHz mono. The maximum stereo transfer rate for 16-bit data is 11 kHz per channel. Stereo ADC transfers for 16-bit data are not allowed in Compatibility mode.

For 8-bit data, the ES1946 expects the first byte transferred to be for the right channel, and subsequent bytes to alternate left, right etc.

For 16-bit data, the ES1946 expects 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

ES1946 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. 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 ES1946 uses the Audio 1 FIFO for DMA transfers to/from the CODEC. When the FIFO is full (in the case of DAC) or empty (in the case of ADC), DMA requests are temporarily suspended and the Busy flag (bit 7 of port SBBase+Ch) is cleared. This opens a window of opportunity to send a command to the ES1946. Commands such as D1h and D3h which control the Audio 1 DAC mixer input enable/disable status, and command D0h, which suspends or pauses DMA, are acceptable to send during this window.

The ES1946 chip sets the Busy flag when the command window is no longer open. Application software must send a command within 13 microseconds after the Busy flag goes high or the command will be confused with DMA data. Sending a command within the command window is easy if polling is done with interrupts disabled.

As an example of sending a command during DMA, consider the case where the application wants 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 D0h command has no function, and a pending interrupt request from the DMA completion is generated.

The interrupt request can be cleared by reading port SBBase+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.

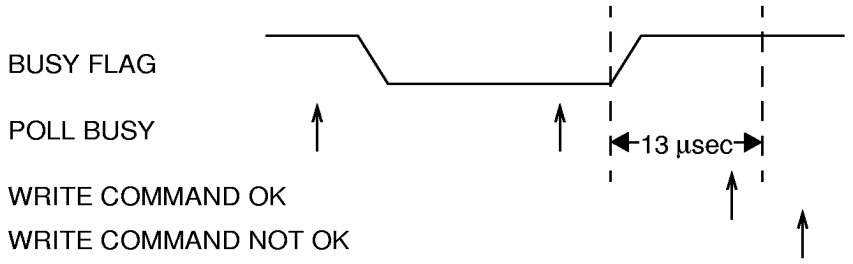


Figure 14 Command Transfer Timing

Compatibility Mode Programming

This section describes Compatibility mode programming.

Compatibility Mode DAC Operation

1. Reset

   Write 1h to port SBBase+6h.

   To play a new sound without resetting the ES1946 beforehand, when the status of the analog circuits is not clear, 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. 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. To set the filter clock to be 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 first timer divider as if you were transferring data at 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 and 91h).
5. Configure the 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 13. See Table 33 for a description of the commands in addition to the commands for DMA transfers of compressed data.

Table 13 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 300 milliseconds 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 mode, it is not necessary to send any commands to the ES1946 at interrupt time, except to read SBBase+Eh to clear the interrupt request.

   For Normal mode, initialize the system DMA controller with the address and count of the next 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 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 26.
9. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in port SBBase+Ch to be sure that data transfer is completed. Delay 25 milliseconds to let the filter outputs settle to DC-levels, then disable the Audio 1 DAC input to the mixer with command D3h.
10. Issue another software reset to the ES1946 to initialize the appropriate registers.

Compatibility Mode ADC Operation

ES1946 analog circuitry is switched from the DAC direction to the ADC direction by the first direct or DMA mode ADC command (2xh). Discard the first 25 to 100 milliseconds of samples because pops might occur in the data due to the change from the DAC to ADC direction. In the ADC direction the voice input to the mixer is automatically muted.

1. Reset

   Write 1h to port SBBase+6h.

   To play a new sound without resetting the ES1946 beforehand, when the status of the analog circuits is not clear, 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 ES1946 has four recording sources: microphone, line, auxiliary A (CD), 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 in steps of 1.5 dB. In 8-bit recordings (other than High-Speed mode), the volume stage is controlled by the ES1946 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. 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 to be 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 first timer divider as if you were transferring data at 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 and 99h).
7. Configure the 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 14. See Table 33 for a description of the commands in addition to the commands for DMA transfers of compressed data.

Table 14 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 milliseconds 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 mode, it is not necessary to send any commands to the ES1946 at interrupt time, except to read SBBase+Eh to clear the interrupt request.

    For Normal mode, initialize the system DMA controller with the address and count of the next 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 DMA, are acceptable to send during DMA transfers. These commands can only be sent during certain windows of opportunity. See “Writing Commands to ES1946 Controller Registers” on page 30.

11. After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in port SBBase+Ch to be sure that data transfer is completed.
12. Issue another software reset to the ES1946 to initialize the appropriate registers.

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-initialize) and 99h (normal). No AGC is performed as the input volume is controlled with command DDh.

Extended Mode Programming

This section describes Extended mode programming.

Commanding ES1946 Controller Registers

Controller registers are written to and read from using commands sent to ports SBBase+Ch and SBBase+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 ES1946 controller registers. For convenience, the registers are named after the commands used to access them. For example “register A4h,” the Audio 1 Transfer Count Reload (low-byte) register, is written to by “command A4h.”

Enabling Extended Mode Commands

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

ES1946 Command/Data Handshaking Protocol

This section describes how to write commands to and read data from the ES1946 controller registers.

Writing Commands to ES1946 Controller Registers

Commands written to the ES1946 enter a write buffer. Before writing the command, make sure the buffer is not busy.

Bit 7 of port SBBase+Ch is the ES1946 Busy flag. It is set when the write buffer is full or when the ES1946 is otherwise busy (for example, during initialization after reset or during Compatibility mode DMA requests).

To write a command or data byte to the ES1946 microcontroller:

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

The following is an example of writing to ES1946 controller registers. To set up the Audio 1 Transfer Count Reload register to F800h, send the following command/data bytes:

A4h, 00h; register A4h = 0h

A5h, F8h; register A5h = F8h

NOTE: The port SBBase+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 such as DMA pause/continue and Audio 1 DAC enable/disable. In this situation it is recommended to disable interrupts between the time that the Busy bit is polled and the command is written. Also, minimize the time between these instructions. See “Sending Commands During DMA Operations” on page 27 for more information.

Reading the Read Data Buffer of the ES1946

Command C0h is used to read the ES1946 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, bit 7 of port SBBase+Eh, before reading the register contents from port SBBase+Ah.

The Read-Data-Buffer-Status flag can be polled by reading bit 7 of port SBBase+Eh. When a byte is available, the bit is set high.

NOTE: Any read of port SBBase+Eh also clears any active interrupt request from the ES1946. An alternate way of polling the Read-Data-Buffer-Status bit is through bit 6 of port SBBase+Ch, which is the same flag. The Read-Data-Buffer-Status flag is cleared automatically by reading the byte from port SBBase+Ah.

Extended Mode Audio 1 DAC Operation

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

1. Reset

   Write 3h to port SBBase+6h, instead of 1h as in Compatibility mode. Bit 1 high specifically clears the FIFO. 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: read this register to preserve the bits and then modify only bits 1 and 0:

   Bits 1:0 10: Mono

   Bits 1:0 01: Stereo

   Set register B9h:

   Bits 1:0 00: Single transfer DMA.

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

   Bits 1:0 11: 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 byte, two's complement

5. Initialize and configure DACs: registers B6h and B7h: See Table 15.

   Register B6h: 80h for signed data and 00h for unsigned data. This also initializes the CODEC for DAC transfer.

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

Table 15 Command Sequences for DMA Playback

Mono	Stereo	8-bits	16-bits	Unsigned	Signed	Sequence
X		X		X		Reg B6h = 80h, Reg B7h = 51h, Reg B7h = D0h
X		X			X	Reg B6h = 00h, Reg B7h = 71h, Reg B7h = F0h
X			X	X		Reg B6h = 80h, Reg B7h = 51h, Reg B7h = D4h
X			X		X	Reg B6h = 00h, Reg B7h = 71h, Reg B7h = F4h
	X	X		X		Reg B6h = 80h, Reg B7h = 51h, Reg B7h = 98h
	X	X			X	Reg B6h = 00h, Reg B7h = 71h, Reg B7h = B8h
	X		X	X		Reg B6h = 80h, Reg B7h = 51h, Reg B7h = 9Ch
	X		X		X	Reg B6h = 00h, Reg B7h = 71h, Reg B7h = BCh

6. Enable/select DMA channel and IRQ channel, registers B1h, and B2h:

   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 milliseconds to allow analog circuits to settle, then enable the Audio 1 DAC input to mixer with command D1h.
10. During DMA

    For Auto-Initialize transfers, read SBBase+Eh to clear the interrupt request. Do not send any other commands to the ES1946 at interrupt time.

    For Normal mode, initialize the system DMA controller with the address and count of the next block to transfer. Update the ES1946 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 a DMA transaction in progress, clear bit 0 of register B8h. To stop a DMA transaction after the current auto-initialize block is finished, clear bit 2 of register B8h, wait for the interrupt, and then clear bit 0 of register 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 port SBBase+Ch to be sure data transfer is completed. A delay of 25 milliseconds is required to let the filter outputs settle to DC-levels, then disable the first DMA DAC input to the mixer with command D3h.

12. Finally:

    Issue another software reset to the ES1946 to initialize the appropriate registers.

Extended Mode Audio 1 ADC Operation

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

1. Reset

   Write 3h to port SBBase+6h instead of 1h as in Compatibility mode. Bit 1 high specifically clears the FIFO. 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 ES1946 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 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, and 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. Unless the recording monitor is enabled, there will be no output from AOUT_L or AOUT_R until the CODEC direction is restored to DAC.

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

   Bits 1:0 10: Mono

   Bits 1:0 01: Stereo

   Bit 3 0: Disable Record Monitor

   Register B9h:

   Bits 1:0 00: Single Transfer DMA

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

   Bits 1:0 11: Demand Transfer:
   4 bytes per DMA request

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

   Register A1h: Sample Rate Clock Divider. Set bit 7 high for sample rates greater than 22 kHz.

   Register A2h: Filter Clock Divider.

   Registers A4h/A5h: Audio 1 Transfer Count Reload register low/high, two's complement

7. Delay 300 milliseconds to allow 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 16. 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 16 Command Sequence for DMA Record

Mono	Stereo	8-bits	16-bits	Unsigned	Signed	Sequence
X		X		X		Reg B7h = 51h, Reg B7h = D0h
X		X			X	Reg B7h = 71h, Reg B7h = F0h
X			X	X		Reg B7h = 51h, Reg B7h = D4h
X			X		X	Reg B7h = 71h, Reg B7h = F4h
	X	X		X		Reg B7h = 51h, Reg B7h = 98h
	X	X			X	Reg B7h = 71h, Reg B7h = B8h
	X		X	X		Reg B7h = 51h, Reg B7h = 9Ch
	X		X		X	Reg B7h = 71h, Reg B7h = BCh

10. Enable/select DMA channel and IRQ channel, registers B1h, and B2h:

    Register B1h: Interrupt Configuration register.
    Verify that bits 4 and 6 are high. Clear bits 7 and 5.

    Register B2h: DRQ Configuration register:
    Verify that 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 transfers, do not send any commands to the ES1946 at interrupt time, except for reading SBBase+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 ES1946 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 a DMA transaction in progress, clear bit 0 of register B8h. To stop a DMA transaction after the current auto-initialize block is finished, clear bit 2 of register B8h, wait for the interrupt, and then clear bit 0 of register 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 ES1946 to initialize the appropriate registers. This returns the ES1946 CODEC to the DAC direction and turns 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 ES1946 port SBBase+Fh for block transfers.

I/O transfers to FIFO are nearly identical to the DMA process, except that an I/O access to port SBBase+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 SBBase+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 operations, an interrupt request is generated when the number of bytes in the FIFO changes from >= 128 to < 128. This indicates to the system processor that 128 bytes can be safely transferred without over-filling 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 >= 128 bytes in the FIFO unless somewhat more than 128 bytes is transferred. Polling the ES1946 FIFOHE flag to be sure it goes low in the interrupt handler (or when priming the FIFO) and perhaps sending a second block of 128 bytes is a solution to this problem.

For ADC, the interrupt request is generated when the number of bytes in the FIFO changes from < 128 to >= 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 SBBase+Fh and polling the FIFOHE flag. It is not safe to use FIFO reset bit 1 of port SBBase+6h indiscriminately 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 ES1946.

Programmed I/O DAC Operation

Programmed I/O DAC operation is done just as explained under “Extended Mode Audio 1 DAC Operation” on page 30 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 command.

Programmed I/O ADC Operation

Programmed I/O ADC operation is done just as explained under “Extended Mode Audio 1 ADC Operation” on page 31 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 command.

Second Audio Channel DAC Operation

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

1. Reset

   Write 3h to port SBBase+6h, instead of 1h as in Compatibility mode. Bit 1 high specifically clears the FIFO. 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 during the setup process.

2. Program transfer type: mixer register 78h:

   Mixer register 78h: set bit 4 low for Normal DMA mode, high for Auto-Initialize DMA mode.

3. Clocks and counters: registers 70h, 74h and 76h:

   Register 70h: Sample Rate Generator
   Registers 74h/76h: Audio 2 Transfer Count Reload register low/high, two's complement

4. Initialize and configure Audio 2 DAC: register 7Ah.

   Register 7Ah:

   Bit 2: set high for signed data, low for unsigned.

   Bit 1: set high for stereo, low for mono.

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

5. Enable IRQ channel, register 7Ah and port IOBase+7h:

   Register 7Ah: Audio 2 Control 2 register.
   Bit 6 unmasks channel 2 IRQ.

   Port IOBase+7h: IRQ control register.
   Bit 5 unmasks channel 2 IRQ.

6. Program IOBase+0h, IOBase+4h, and IOBase+6h.
7. To start DMA:

   Set bits 1:0 of register 78h high.

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

   For Auto-Initialize transfers, read SBBase+Eh to clear the interrupt request. Do not send any other commands to the ES1946 at interrupt time.

   For Normal mode, initialize IOBase+0h and IOBase+4h with the address and count of the next block to transfer. Update the ES1946 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 a DMA transaction in progress, clear bit 0 of register B8h. To stop a DMA transaction 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 register 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 SBBase+Ch to be sure data transfer is completed. A delay of 25 milliseconds is required to let the filter outputs settle to DC-levels, then disable the Audio 2 DAC input to the mixer.

11. Finally:

    Issue another software reset to the ES1946 to initialize the appropriate registers.

Full-Duplex DMA Mode

The ES1946 supports stereo full-duplex DMA. In full-duplex (FD) mode, a second audio channel has been added to the ES1946. The second audio channel is programmed through mixer registers. The CODEC is used for recording and the Audio 2 DAC is used for playback.

1. Program the first audio channel as in “Extended Mode Audio 1 ADC Operation” on page 31. Extended mode registers A1h and A2h define the sample rate and filter frequency for both record and playback.
2. Program the second audio channel. Mixer registers 74h and 76h are set to the two's complement DMA transfer count. The second audio channel supports both Auto-Initialize and Normal modes. 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 zero, 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. Program the DMA Transfer Count Reload registers (A4h, A5h, 74h, and 76h) are programmed with the same value for both channels. Enable the second audio channel before enabling 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 generates an interrupt. To ensure that the playback channel is not accessing the first half at the time of the interrupt, start the playback channel first. It has a 32-word FIFO that fills quickly through DMA.

Programming the ES1946 Mixer

The ES1946 has a set of mixer registers that support the Sound Blaster Pro. However, some of the registers can be accessed in an "extended" or "alternate" way, providing for greater functionality.

Writing and Reading Data from the Mixer Registers

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

Writing Data to the ES1946 Mixer Registers

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

Reading Data from the ES1946 Mixer Registers

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

Resetting the Mixer Registers

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

1. Write 00h to SBBase+4h
   (select mixer register to 00h).
2. Write 00h to SBBase+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. See the Sound Blaster Support register 04h in Table 28 for details. Bits 0 and 4 are always high when read. The ES1946 offers an alternate way to write each mixer register. Use the "extended access" registers for volume control of 4 bits per channel. If the legacy DOS compatible interface is used, bits 0 and 4 are cleared by a write and forced high on all reads. See Table 17 for a list of Sound Blaster Pro registers and the extended access counterparts.

Table 17 Sound Blaster Pro/Extended Access Registers

Register	Function	Extended Access Register for 4 bits/Channel
04h	Voice 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.

A write or read to address 14h instead of 04h allows direct access to all 8 bits of this mixer register.

Extended Access to Mic Mix Volume

If Legacy DOS 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 reads and stuck low on writes. Since this is a mono control, panning is not supported.

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

Mic Mix Volume **(1Ah, R/W)**

Mic mix volume left				Mic mix volume right			
7	6	5	4	3	2	1	0

Access to register 1Ah through 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 Legacy DOS 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 zero 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
0	0	0	Microphone (default)
0	0	1	Reserved
0	1	0	CD (Aux) input
0	1	1	Reserved
1	0	0	Microphone
1	0	1	Record mixer
1	1	0	Line input
1	1	1	Reserved

Sound Blaster Pro Volume Emulation

Sound Blaster Pro emulations for master volume means that the 6-bit volume counters can be written through the Sound Blaster Pro mixer register 22h (or 32h). Sound Blaster Pro emulation is enabled by default, and can be disabled by setting bit 0 of mixer register 64h.

The master volume registers 60h and 62h can always be read, regardless of whether Sound Blaster Pro volume emulation is enabled, using the Sound Blaster Pro mixer registers 22h (and 32h). The following 6-bit to 4-bit translation table is used.

Table 18 SB Pro Read Volume Emulation

Mute	Master Volume	Value Read at 32h	Value Read at 22h
1	xx	0	1
0	0-24	1	1
0	25-30	2	3
0	31-34	3	3
0	35-38	4	5
0	39-42	5	5
0	43-46	6	7
0	47-50	7	7
0	51-54	8	9
0	55	9	9
0	56-57	10	11
0	58	11	11
0	59-60	12	13
0	61	13	13
0	62	14	15
0	63	15	15

If Sound Blaster Pro volume emulation is enabled, then a mixer reset causes both left and right channels to be set to their power-on defaults, namely 54 (36h).

If Sound Blaster Pro volume emulation is enabled, then a write to mixer register 22h (or 32h) causes both left and right master volume registers to be changed as follows:

Table 19 SB Pro Write Volume Emulation

Value written to 22h or 32h	Mute	6-Bit Volume
0	1	24
1	0	24
2	0	30
3	0	34
4	0	38
5	0	42
6	0	46
7	0	50
8	0	54
9	0	55
10	0	56
11	0	58
12	0	59
13	0	61
14	0	62
15	0	63

Record and Playback Mixer

The ES1946 has stereo mixers for playback and record. Each stereo mixer has eight input sources, each with independent 4-bit left and right volume controls. For each 4-bit volume control, level 0 is mute and level 15 is maximum volume. The ES1946 mixers use a dual slope method for selecting volume. Each increase of one step in volume from settings 1 to 8 results in a +3 dB increase. Each increase of one step in volume from settings 8 to 15 results in a +1.5 dB increase.

Table 20 Extended Access Mixer Volume Values

4-Bit Value	Volume in Decibels (dB)			
	Audio 1[a] (Record)	Audio 2 (Playback)	Mic, Music DAC	AuxA, AuxB, Line, Mono-In
15	+3.0	0	+10.5	+3.0
14	+1.5	-1.5	+9.0	+1.5
13	0	-3.0	+7.5	0
12	-1.5	-4.5	+6.0	-1.5
11	-3.0	-6.0	+4.5	-3.0
10	-4.5	-7.5	+3.0	-4.5
9	-6.0	-9.0	+1.5	-6.0
8	-7.5	-10.5	0	-7.5
7	-10.5	-13.5	-3.0	-10.5
6	-13.5	-16.5	-6.0	-13.5
5	-16.5	-19.5	-9.0	-16.5
4	-19.5	-22.5	-12.0	-19.5
3	-22.5	-25.5	-15.0	-22.5
2	-25.5	-28.5	-18.0	-25.5
1	-28.5	-31.5	-21.0	-28.5
0	mute	mute	mute	mute

a. Audio 1 DAC mixer input is gated by the Sound Blaster "Speaker On" control. This control bit is toggled by the D1 (on) and D3 (off) Sound Blaster commands.

Table 21 Mixer Input Volume Registers

Mixer Input	Playback Volume Register	Record Volume Register
Audio 1	14h	–
Audio 2	7Ch	69h
Microphone	1Ah	68h
Music DAC (FM)	36h	6Bh
AuxA (CD)	38h	6Ah
AuxB	3Ah	6Ch
Line	3Eh	6Eh
Mono In	6Dh	6Fh

PROGRAMMING MODEL

Register Types

There are four types of registers or I/O ports in the ES1946:

- PCI configuration registers (00h – C4h).

  These registers are accessed through the PCI bus interface. These registers are accessed through the PCI configuration mechanism as described in the *PCI Local Bus Specification* available from the PCI Special Interest Group.

- I/O mapped registers or I/O ports

  These registers are mapped into I/O address space in the PCI system. They are accessed through the PCI I/O read/write cycle as described in the *PCI Local Bus Specification* available from the PCI Special Interest Group.

- Mixer registers (00h – 7Fh).

  These registers are accessed through I/O ports SBBase+4h and SBBase+5h. SBBase+4h is written with the register address. Then the register can be read written through SBBase+5h. These registers control many functions other than the mixer.

- Controller registers (A0h – BFh).

  These registers are used to control Extended mode DMA playback and record through the first audio channel. Controller registers are accessed through an extension to the Sound Blaster common interface. This interface uses I/O ports SBBase+Ah, SBBase+Ch, and SBBase+Eh to transfer read data, write data/commands, and status respectively.

PCI Configuration Registers

Register Summary

Table 22 PCI Configuration Registers Summary

31 16	15 0	Offset
Device ID	Vendor ID	00h
Device status	Command	04h
Base class code / Sub-class code	Programming interface / Revision ID	08h
Reserved / Header type	Master latency timer / Reserved	0Ch
I/O space base address		10h
SB I/O space base address for native PCI audio		14h
VC I/O space base address for native PCI audio		18h
MPU-401 I/O space base address for native PCI audio		1Ch
Game port I/O space base address for native PCI audio		20h
Reserved		24h
Reserved		28h
Subsystem ID (read/write-protected)	Subsystem vendor ID (read/write-protected)	2Ch
Reserved		30h
Reserved	Capability pointer	34h
Reserved		38h
Maximum latency / Minimum grant	Interrupt pin / Interrupt line	3Ch
Reserved	Legacy audio control	40h
Reserved		44h
Reserved		48h
Reserved		4Ch
ES1946 configuration		50h
Reserved		54h
Reserved		58h
Reserved		5Ch
Reserved	Distributed DMA control	60h
Shadow SID	Shadow SVID	6Ch
Power-Management capabilities	Next-Item pointer / Capability ID	C0h
Reserved	Power-Management control/status	C4h

All reserved locations are read-only with a default value of zero.

Register Descriptions

Vendor ID (00h, 01h, R)

Vendor ID
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
15:0	Vendor ID	The default value after reset is 125Dh, indicating ESS as the manufacturer of this device.

Device ID (02h, 03h, R)

Device ID
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
15:0	Device ID	The default value after reset is 1969h.

Command (04h, 05h, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	BME	0	IOS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000h.

Bit Definitions:

Bits	Name	Description
15:3	–	Reserved. Returns 0 when read.
2	BME	Bus Master enable.
1	–	Reserved. Returns 0 when read.
0	IOS	I/O Space access enable.

Device Status (06h, 07h, R/W)

0	0	MAS	RTA	STA	DT		DPE	FBC	UDF	66M	ACPI	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Writing 1 clears a bit; writing 0 has no effect on a bit.

The default value after reset is 0290h.

Bit Definitions:

Bits	Name	Description
15:14	–	Reserved. Returns 0 when read.
13	MAS	Master abort status (read/write-clear).
12	RTA	Received target abort status (read/write-clear).
11	STA	Signaled target abort. Read-only 0.
10:9	DT	DEVSEL timing. Read-only 01h. 01h = medium.
8	DPE	Data parity error detected. Read-only 0.
7	FBC	Fast back-to-back capable. Read-only 1.
6	UDF	UDF supported. Read-only 0.
5	66M	66 MHz capable. Read-only 0.
4	ACPI	ACPI capable. Read-only 1.
3:0	–	Reserved. Returns 0 when read.

Revision ID (08h, R)

Revision ID
7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
7:0	Revision ID	Identifies the revision of this device. The default value after reset is 02h.

Programming Interface (09h, R)

Programming interface
7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
7:0	PI	Identifies the programming interface of this device. The default value after reset is 00h.

Sub-Class Code (0Ah, R)

Sub-Class code
7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
7:0	SCC	The default value after reset (assigned by the PCI-SIG) is 01h, indicating an audio device with a multimedia base class.

Base Class Code (0Bh, R)

Base class code							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	BCC	The default value after reset (assigned by the PCI-SIG) is 04h, indicating a multimedia device.

Master Latency Timer (0Dh, R/W)

Master latency timer count value				0	0	0	0
7	6	5	4	3	2	1	0

The default value after reset is 00h.

Bit Definitions:

Bits	Name	Description
7:4	MLTCV	Master latency timer count value.
3:0	–	Reserved. Returns 0 when read.

Header Type (0Eh, R)

Header type							
7	6	5	4	3	2	1	0

The default value after reset is 00h.

Bit Definitions:

Bits	Name	Description
7:0	HEDT	Header type. A value of 00h indicates a single-function PCI device.

I/O Base (10h – 13h, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOSB										0	0	0	0	0	ISI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000xxx1h. Bits 15:6 are uninitialized on reset.

Bit Definitions:

Bits	Name	Description
31:16	–	Read-only 0000h.
15:6	IOSB	I/O space base address. The ES1946 claims 64 bytes of I/O space.
5:1	–	Read-only 00000b.
0	ISI	I/O space indicator. Read-only 1.

SB Base for Native-PCI-Audio (14h – 17h, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBSB												0	0	0	ISI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000xxx1h. Bits 15:4 are uninitialized on reset.

Bit Definitions:

Bits	Name	Description
31:16	–	Read-only 0000h.
15:4	SBSB	SB space base address for native PCI audio. The ES1946 claims 16 bytes of Sound Blaster I/O space.
3:1	–	Read-only 000b.
0	ISI	I/O space indicator. Read-only 1.

VC Base for Native-PCI-Audio (18h – 1Bh, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VCSB												0	0	0	ISI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000xxx1h. Bits 15:4 are uninitialized on reset.

Bit Definitions:

Bits	Name	Description
31:16	–	Read-only 0000h.
15:4	VCSB	VC space base address for native PCI audio. The ES1946 claims 16 bytes of driver I/O space. The driver can use allocated I/O ports for the DDMA base address.
3:1	–	Read-only 000b.
0	ISI	I/O space indicator. Read-only 1.

MPU Base for Native-PCI-Audio (1Ch – 1Fh, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUSB														0	ISI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000xxxxh. Bits 15:2 are uninitialized on reset.

Bit Definitions:

Bits	Name	Description
31:16	–	Read-only 0000h.
15:2	MPUSB	MPU space base address for native PCI audio. The ES1946 claims 4 bytes of MPU-401 I/O space.
1	–	Read-only 0.
0	ISI	I/O space indicator. Read-only 1.

GP Base for Native-PCI-Audio (20h – 23h, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPSB														0	ISI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000xxxxh. Bits 15:2 are uninitialized on reset.

Bit Definitions:

Bits	Name	Description
31:16	–	Read-only 0000h.
15:2	GPSB	GP space base address for native PCI audio. The ES1946 claims 4 bytes of game port I/O space.
1	–	Read-only 0.
0	ISI	I/O space indicator. Read-only 1.

Subsystem Vendor ID (2Ch, 2Dh, R)

Subsystem vendor ID															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

If there is an EEPROM, the power-on default is read from the EEPROM, otherwise the default value is 125Dh.

To load the Subsystem Vendor ID in BIOS POST code, the configuration register (50h - 53h) bit 12 must be set and the new SVID must be written to the shadow register 6Ch - 6Dh.

Bit Definitions:

Bits	Name	Description
15:0	SVID	Subsystem Vendor ID. (read only)

Subsystem ID (2Eh, 2Fh, R)

Subsystem ID															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

If there is an EEPROM, the power-on default is read from the EEPROM, otherwise the default value is 8898h.

To load the Subsystem ID BIOS POST code, the configuration register (50h - 53h) bit 12 must be set and the new SID must be written to the shadow register 6Eh - 6Fh.

Bit Definitions:

Bits	Name	Description
15:0	SID	Subsystem ID. (read only)

ACPI Capability Pointer (34h, R)

ACPI capability pointer							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	ACPICP	ACPI Cap_Ptr. The default value is C0h. Read-only.

Interrupt Line (3Ch, R/W)

Interrupt line							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	IL	Interrupt line. Valid values are 0 – 15, 255. Bits 7:5 echo the value of bit 4. The default value is 255.

Interrupt Pin (3Dh, R)

Interrupt pin							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	IP	Interrupt pin. The default value is 01h, indicating INTA. Read-only.

Minimum Grant (3Eh, R)

Minimum grant							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	MG	Min_Gnt. The default value is 02h, corresponding to 500 ns. Read-only

Maximum Latency (3Fh, R)

Maximum latency							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	ML	Max_Lat. The default value is 18h, corresponding to 6 ms.

Legacy Audio Control (40h, 41h, R/W)

LA	R	MIDIIRQ			SBIRQ			DMACH	IA	MQ	MI	GP	FM	SB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 907Fh. This is also a control register in PCI mode.

Bit Definitions:

Bits	Name	Description
15	LA	Legacy audio address decode disable. 1 = Disable Legacy Audio mode (default). 0 = Enable Legacy Audio mode.
14	–	Reserved. When bit 16 of PCI configuration register 50h – 53h is 1, write 1 to this bit. Otherwise write 0.
13:11	MIDIIRQ	MIDI IRQ select.

Bit 13	Bit 12	Bit 11	IRQ Selection
0	0	0	IRQ 5
0	0	1	IRQ 7
0	1	0	IRQ 9 (default)
0	1	1	IRQ 10
1	0	0	IRQ 11
1	0	1	IRQ 12
1	1	0	IRQ 13
1	1	1	IRQ 14

10:8 SBIRQ Sound Blaster IRQ select.

Bit 10	Bit 9	Bit 8	IRQ Selection
0	0	0	IRQ 5 (default)
0	0	1	IRQ 7
0	1	0	IRQ 9
0	1	1	IRQ 10
1	0	0	IRQ 11
1	0	1	IRQ 12
1	1	0	IRQ 13
1	1	1	IRQ 14

Bits	Name	Description
7:6	DMACH	Sound Blaster DMA channel select.

Bit 7	Bit 6	DMA Channel Selection
0	0	Channel 0
0	1	Channel 1 (default)
1	0	Reserved.
1	1	Channel 3

5	IA	I/O address aliasing control. 1 = 10-Bit I/O address (default). 0 = 16-Bit I/O address.
4	MQ	MPU-401 IRQ enable/mask. 1 = Enable MPU-401 IRQ (default). 0 = Disable MPU-401 IRQ.
3	MI	MPU-401 I/O enable. 1 = Enable MPU-401 I/O (default). 0 = Disable MPU-401 I/O.
2	GP	Game port enable. 1 = Enable game port (default). 0 = Disable game port.
1	FM	FM synthesis enable. 1 = Enable FM synthesis (default). 0 = Disable FM synthesis.
0	SB	Sound Blaster enable. 1 = Enable Sound Blaster channel (default). 0 = Disable Sound Blaster channel.

Legacy Audio Support

The ES1946 supports the following legacy audio addresses.

Table 23 Supported Legacy Audio Addresses

Legacy Audio Resources	I/O Address Base
Sound Blaster Pro	220h/240h
FM synthesis	388h
MPU-401	300h/320h/330h/340h
DMA	Channel 0, 1, 3
IRQ	5, 7, 9, 10, 11–14
Joystick	201h

ES1946 Configuration (50h – 53h, R/W)

R	CPE	Reserved	ISAIRQ
31	30	29 28 27 26 25 24 23 22 21 20 19 18 17	16

R	IRQP	SID-SEL	R	DMAP	R	0	0	MPUBA	SBBA	R	R
15	14	13	12	11 10 9	8	7	6	5	4	3	2	1	0

The default value after reset is 00000000h.

Bit Definitions:

Bits	Name	Description
31	–	Reserved.
30	CPE	CLKRUN protocol enable.
29:17	–	Reserved.
16	ISAIRQ	ISA serial IRQ enable.
15	–	Reserved.
14:13	IRQP	ISA IRQ emulation policy.

0	0	Emulation is disabled.
0	1	PCI IRQ is one of IRQ 5/7/9/10.
1	0	PCI IRQ is not one of IRQ 5/7/9/10.
1	1	Reserved.

12	SIDSEL	SID/SVID select. 1 = Enable shadow ID register 6C - 6E. 0 = Get SID information from EEPROM (default).
11	–	Reserved.
10:8	DMAP	DMA policy.

Bit 10	Bit 9	Bit 8	DMA Policy
0	0	0	Distributed DMA
0	0	1	Transparent DMA
0	1	0	PC/PCI DMA
0	1	1	Reserved
1	0	0	Transparent DMA 1
1	0	1	Transparent DMA 2
1	1	0	Transparent DMA 3
1	1	1	Transparent DMA 4

7	–	Reserved. Write 0.
6:5	–	Reserved. Returns 0 when read.
4:3	MPUBA	MPU base address select.

Bit 4	Bit 2	MPU-401 I/O
0	0	330h
0	1	300h
1	0	320h
1	1	340h

2	SBBA	SB base address select. 0 = Sound Blaster decode is 220h. 1 = Sound Blaster decode is 240h.
1	-	Reserved. Write 0.
0	-	Reserved.

Distributed DMA Control (60h, 61h, R/W)

DIOSB	0	0	0	DE
15 14 13 12 11 10 9 8 7 6 5 4	3	2	1	0

The default value after reset is 0000h.

Bit Definitions:

Bits	Name	Description
15:4	DIOSB	Distributed DMA base address.
3:1	–	Reserved. Always write 0.
0	DE	Distributed DMA enable. 1 = Enable distributed DMA. 0 = Disable distributed DMA.

Shadow SVID (6C-6Dh, R/W)

SVID
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

This shadow Subsystem Vendor ID register can be read and written only when the ES1946 configuration register (50h - 53h) bit 12 is 1.

Bit Definitions:

Bits	Name	Description
15:0	SVID	Subsystem Vendor ID

Shadow SID (6E-6Fh, R/W)

SID
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

This shadow Subsystem ID register can be read and written only when the ES1946 configuration register (50h - 53h) bit 12 is 1.

Bit Definitions:

Bits	Name	Description
15:0	SID	Subsystem ID

Capability ID (C0h, R)

Capability ID
7 6 5 4 3 2 1 0

Bit Definitions:

Bits	Name	Description
7:0	CID	Read-only. This register identifies the linked list item as the register for PCI power management. The default value (assigned by the PCI-SIG) is 01h, indicating the unique ID for the PCI location of the capabilities pointer and the value.

Next-Item Pointer (C1h, R)

Next-Item pointer							
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:0	NIP	Read-only. The default value is 00h, indicating that there are no more items in the linked list of the PCI power management capabilities.

Power-Management Capabilities (C2h, C3h, R)

0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0621h.

Bit Definitions:

Bits	Name	Description
15:11		PME Support. This five-bit field indicates the power states in which the function may assert PME. A value of 0 for any bit indicates that the function is not capable of asserting the PME signal while in that power state. Bit [15] = 0. PME# cannot be asserted from $D3_{cold}$. Bit [14] = 0. PME# cannot be asserted from $D3_{hot}$. Bit [13] = 0. PME# cannot be asserted from D2. Bit [12] = 0. PME# cannot be asserted from D1. Bit [11] = 0. PME# cannot be asserted from D0. Value of bits 15:11 = 00000.
10		D2 Support. This bit indicates that this function supports the D2 power management state. Value of bit 10 = 1.
9		D1 Support. This bit indicates that this function supports the D1 power management state. Value of bit 9 = 1.
8:6	–	Reserved. Value of bits 8:6 = 000.
5		DSI. The Device Specific Initialization bit indicates special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. Value of bit 5 = 1.
4		Auxiliary power source. Value of bit 4 = 0.
3		PME clock. This bit indicates that no PCI clock is required for the function to generate PME. Value of bit 3 = 0.
2:0		Version. This 3-bit field indicates that this function complies with Revision 1.0 of the PCI Power Management Interface specification. Value of bits 2:0 = 001.

Power-Management Control/Status (C4h, C5h, R/W)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	PS	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The default value after reset is 0000h.

Bit Definitions:

Bits	Name	Description
15:2	–	Reserved. Returns 0 when read.
1:0	PS	Power state. This 2-bit field is used both to determine the current power state of a function, and to set the function into a new power state.

Bit 1	Bit 0	Power State
0	0	Normal mode (D0)
0	1	Microcontroller halted (D1)
1	0	Microcontroller halted and analog off (D2)
1	1	Microcontroller halted, analog off, and oscillator off (D3)

I/O Ports

These registers are divided into 5 groups. Each group has a corresponding base address. The registers are addressed by the base address plus hexadecimal offset. Base addresses are determined as shown in Table 24.

Table 24 I/O Port Base Addresses

Group	PCI Native Mode	Legacy Mode
IOBase	PCICONF+10h	—
SBBase	PCICONF+14h	220/240[a]
FMBase	PCI Config + 14h (share with SBBase)	338
MPUBase	PCICONF+1Ch	300/320/330/340[b]
GPBase	PCICONF+20h	200h
DDMABase	PCICONF+60h	—

a. Selected by PCICONF+50h[2].
b. Selected by PCICONF+50h[3].

Port Summary

Table 25 I/O Port Summary

Port	Read/ Write	Function
I/O Device		
IOBase+0h–IOBase+3h	Read/write	Audio 2 base/current DMA address.
IOBase+4h–IOBase+5h	Read/write	Audio 2 base/current DMA count.
IOBase+6h	Read/write	ES1946 mode register.
IOBase+7h	Read/write	IRQ control register.
IOBase+2Dh	Read/write	EEPROM type
IOBase+2Eh	Read/write	EEPROM data register
IOBase+2Fh	Read/write	EEPROM command/address register
Audio/FM Device		
SBBase+0h–SBBase+3h	Read/write	20-voice FM synthesizer. Address and data registers. (FMBase+0h–FMBase+3h)
SBBase+4h	Read/write	Mixer address register (port for address of mixer controller registers).
SBBase+5h	Read/write	Mixer data register (port for data to/from mixer controller registers).
SBBase+6h	Read/write	Audio reset and status flags.
SBBase+7h	Read/write	Power Management register. Suspend request and FM reset.
SBBase+8h–SBBase+9h	Read/write	11-voice FM synthesizer. Address and data registers. Alias of SBBase+0h–SBBase+3h.
SBBase+Ah	Read-only	Input data from read buffer for command/data I/O. Poll bit 7 of port SBBase+Eh to test whether the read buffer contents are valid.
SBBase+Ch	Read/write	Output data to write buffer for command/data I/O. Read embedded processor status.
SBBase+Eh	Read-only	Data available flag from embedded processor.
SBBase+Fh	Read/write	Address for I/O access to FIFO in Extended mode.
MPU-401 Device		
MPUBase+0h–MPUBase+1h	Read/write	MPU-401 port if enabled.
Game Port Device		
GPBase+1h	Read/write	Joystick.
DMAC Device		
DDMABase+0h–DDMABase+2h	Read/write	DMA current/base address.
DDMABase+4h–DDMABase+5h	Read/write	DMA current/base count.
DDMABase+8h	Read/write	DMA command/status.
DDMABase+9h	Write	DMA request.
DDMABase+Bh	Write	DMA mode.
DDMABase+Dh	Write	DMA master clear.
DDMABase+Fh	Read/write	DMA mask.

Port Descriptions

This section describes the various devices and I/O port descriptions.

I/O Device

This device controls the Audio 2 DMA controller internal to the ES1946 and other functions.

Audio 2 Base/Current DMA Address

(IOBase+0h – IOBase+3h, R/W)

A2DMAA (high word)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A2DMAA (low word)												0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write to the DMA base address register, read from the DMA current address register. The current address is automatically copied from the base register when DMA starts. The current address is then incremented.

The default value is xxxxxxx0h.

NOTE: The internal counter counts bits [19:0], then concatenates that with the upper bits of the base register.

Bit Definitions:

Bits	Name	Description
31:4	A2DMAA	Audio 2 DMA address. Write to base address. Read from current address.
3:0	–	Read-only 0h.

Audio 2 Base/Current DMA Count

(IOBase+4h – IOBase+5h, R/W)

A2DMAC												0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write to the DMA base count register, read from the DMA current count register. The current count is automatically copied from the base register when DMA starts. The current count is then decremented for each byte transferred.

The default value is xxx0h.

NOTE: The ISA DMAC (8237) sets n-1 to the count register. But this register needs n.

Bit Definitions:

Bits	Name	Description
15:4	A2DMAC	Audio 2 DMA count. Write to base count. Read from current count.
3:0	–	Read-only 0h.

Mode

(IOBase+6h, R/W)

0	0	0	0	Auto-Init en	BCLK sel	DMA en	DIR
7	6	5	4	3	2	1	0

The default value is 00h.

Bit Definitions:

Bits	Name	Description
7:4	–	Reserved. Always write 0.
3	Auto-Init en	Auto-Initialize enable for Audio 2 DMA. 1 = Enable auto-initialization. 0 = Disable auto-initialization.
2	BCLK sel	BCLK select. 1 = PCLK/3. 0 = PCLK/4.
1	DMA en	Audio 2 DMA enable. 1 = Enable DMA. 0 = Disable DMA.
0	DIR	Audio 2 DMA Direction. 0 = Memory to DAC. Read-only.

IRQ Mask

(IOBase+7h, W)

MPUIRQ	HVIRQ	A2IRQ	A1IRQ	0	0	0	0
7	6	5	4	3	2	1	0

The default value is 00h.

Bit Definitions:

Bits	Name	Description
7	MPUIRQ	IRQ enable for MPU-401 IRQ.
6	HVIRQ	IRQ enable for hardware volume IRQ.
5	A2IRQ	IRQ enable for audio 2 IRQ.
4	A1IRQ	IRQ enable for audio 1 IRQ.
3:0	–	Reserved. Always write 0.

IRQ Status Mask

(IOBase+7h, R)

MPUIRQ	HVIRQ	A2IRQ	A1IRQ	0	0	0	0
7	6	5	4	3	2	1	0

The default value is 00h.

Bit Definitions

Bits	Name	Description
7	MPUIRQ	IRQ status for MPU401
6	HVIRQ	IRQ status for hardware volume.
5	A2IRQ	IRQ status for audio 2 IRQ.
4	A1IRQ	IRQ status for audio 1 IRQ.
3:0	–	Reserved. Always Read 0.

EEPROM Type (IOBase+2Dh, R/W)

0	0	0	0	0	0	Type	
7	6	5	4	3	2	1	0

During normal operation, it is not required to program this register. The ES1946 can auto-detect EEPROM type and read the first six bytes for SID/SVID after reset.

In order for the EEPROM auto-detect to function correctly, the first two bytes of the EEPROM must be programmed with the correct ID. See Table 26. If the EEPROM is not correctly programmed, 93LC66 is used as the default, but the subsequent SID and SVID bytes won't be recognized.

Table 26 EEPROM Auto-detect ID Bytes

EEPROM	First Byte	Second Byte
93LC46	46	38
93LC66	66	38

The default value is 00h.

Bit Definitions:

Bits	Name	Description
7:2	–	Reserved. Always write 0.
1:0	–	EEPROM type select.

Bit 1	Bit 0	Device
0	0	auto-detect
0	1	93LC46
1	0	Reserved.
1	1	93LC66

EEPROM Data (IOBase+2Eh, R/W)

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

The default value undefined.

Bit Definitions:

Bits	Name	Description
7:0	Data	Data port to/from the EEPROM.

EEPROM Command/Address (IOBase+2Fh, W)

0	0	0	0	Command			
7	6	5	4	3	2	1	0

This register is used to modify the contents of an external EEPROM. See "Programming the EEPROM" on page 21. The internal address counter is incremented by issuing a command.

Bit Definitions:

Bits	Name	Description
7:4	–	Reserved. Always write 0.
3:0	Command	See Table 27 for a decode of the command bits.

Table 27 EEPROM Access Command Summary

Bit 3	Bit 2	Bit 1	Bit 0	Symbol	Command
0	0	0	0	EWDS	Erase and write disable
0	0	0	1	WRAL	Write all
0	0	1	0	ERAL	Erase all
0	0	1	1	EWEN	Erase and write enable
0	1	0	0	WRITE	Write word
0	1	0	1	–	Reserved
0	1	1	0	–	Reserved
0	1	1	1	–	Reserved
1	0	0	0	READ	Read word
1	0	0	1	–	Reserved
1	0	1	0	–	Reserved
1	0	1	1	–	Reserved
1	1	0	0	ERASE	Erase word
1	1	0	1	–	Reserved
1	1	1	0	–	Reserved
1	1	1	1	–	Reserved

EEPROM Address Reset (IOBase+2Fh, R)

Address counter reset							
7	6	5	4	3	2	1	0

Reading this register resets the internal address counter.

Audio/FM Device

The FM synthesizer operates in two different modes: Emulation mode and Native mode. In Emulation mode the FM synthesizer is fully compatible with the OPL3 FM synthesizer. In Native mode the FM synthesizer has increased capabilities and performance for more realistic music. The following register descriptions are for Emulation mode only.

FM Status **(SBBase+0h, R)**

IRQ	FT1	FT2	0	0	0	0	0
7	6	5	4	3	2	1	0

Reading this register returns the overflow flags for timers 1 and 2 and the “interrupt request” from these timers (this is not a real interrupt request but is supported as a status flag for backward compatibility with the OPL3 FM synthesizer). This register can also be accessed from FMBase+0h.

FM Low Bank Address **(SBBase+0h, W)**

A7	A6	A5	A4	A3	A2	A1	A0
7	6	5	4	3	2	1	0

Low bank register address. This register can also be accessed from FMBase+0h.

NOTE: Any write to this register will also put the FM synthesizer in Emulation mode if it is currently in Native mode.

FM Low Bank Data Write **(SBBase+1h, W)**

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

FM register write. The data written to SBBase+1h is written to the current address FM register. Note that register writes must follow the timing requirements of the OPL3 FM synthesizer. This register can also be accessed from FMBase+1h.

FM High Bank Address **(SBBase+2h, W)**

A7	A6	A5	A4	A3	A2	A1	A0
7	6	5	4	3	2	1	0

High bank register address. This register can also be accessed from FMBase+2h.

FM High Bank Data Write **(SBBase+3h, W)**

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

FM register write. Writing to this register in Emulation mode is the same as writing to register SBBase+1h. This register can also be accessed from FMBase+3h.

Mixer Address Register **(SBBase+4h, R/W)**

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

The ES1946 provides a means to read back the Mixer Address register. Reading back this register is useful for a “hot-key” application that needs to change the mixer while preserving the address register.

Mixer Data Register **(SBBase+5h, R/W)**

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

Reset **(SBBase+6h, W)**

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

Bit Definitions:

Bits	Name	Description
7:2	–	Reserved. Always write 0.
1	FIFO reset	1 = Hold ES1946 FIFO in reset. 0 = Release ES1946 FIFO from reset. Bit 1 has no function for Compatibility mode.
0	SW reset	1 = Hold ES1946 in reset. 0 = Release ES1946 from reset.

NOTE: The FM synthesizer can be reset by bit 5 of SBBase+7h.

Status Flags **(SBBase+6h, R)**

R	Act flag 1	Act flag 0	Serial act flag	R	MIDI modes	FIFO reset	SW reset
7	6	5	4	3	2	1	0

Bits 6:5 of port SBBase+6h can be used to monitor I/O activity to the ES1946.

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

Bit 4 can be used to indicate if the 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 set high periodically.

Bit Definitions:

Bits	Name	Description
7	–	Reserved.
6	Act flag1	Set low by I/O reads/writes to audio ports SBBase+4h and SBBase+5h.
5	Act flag 0	Set low by I/O writes to audio ports SBBase+0h–SBBase+3h, SBBase+6h, and SBBase+Ch. Set low by I/O reads from audio ports SBBase+0h–SBBase+3h, and SBBase+Ah. Also set low by DMA accesses to ES1946.
4	Serial act flag	1 = Serial activity flag. High if an external ES689/ES69x is using MCLK/MSD to drive the music DAC.
3	–	Reserved.
2	MIDI mode	1 = The ES1946 is processing a controller command 30h, 31h, 34h, or 35h and is waiting for the command to complete. Powering-down may cause a loss of data. The ES1946 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 Register (SBBase+7h, R/W)

Suspend request	R	FM synth reset	R	R	R	R	R
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	Suspend request	Pulse high, then low to request suspend.
6	–	Reserved.
5	FM synth reset	1 = Hold FM synthesizer in reset. 0 = Release FM synthesizer from reset.
4	–	Reserved.
4:0	–	Reserved.

Read Data Register (SBBase+Ah, R)

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

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

Write Data Register (SBBase+Ch, W)

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

Write data to embedded audio processor. Sets bit 7 of port SBBase+Ch high (write buffer not available) until data is processed by the ES1946. This register cannot be written when SBBase+Ch bit 7 is high.

Buffer Status Register (SBBase+Ch, R)

BUSY flag	RDAV	FIFO full	FIFO empty	FIFO half	IRQ flag2	IRQ flag1	IRQ flag0
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	BUSY flag	1 = Write buffer not available or ES1946 busy. 0 = Write buffer available or ES1946 not busy.
6	RDAV	1 = Data available in read buffer. 0 = Data not available in read buffer. This flag is reset by a read from port SBBase+Ah.
5	FIFO full	1 = Extended mode FIFO Full (256 bytes loaded).
4	FIFO empty	1 = Extended mode FIFO Empty (0 bytes loaded).
3	FIFO half	1 = FIFO Half Empty, Extended mode flag.
2	IRQ flag2	1 = ES1946 processor generated an interrupt request (e.g., from Compatibility mode DMA complete).
1	IRQ flag1	1 = Interrupt request generated by FIFO Half Empty flag change. Used by programmed I/O interface to FIFO in Extended mode.
0	IRQ flag0	1 = Interrupt request generated by DMA counter overflow in Extended mode.

Read Buffer Status Register (SBBase+Eh, R)

RDAV	Same as SBBase+Ch[6:0]						
7	6	5	4	3	2	1	0

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

Bit Definitions:

Bits	Name	Description
7	RDAV	1 = Data available in read buffer. 0 = Data not available in read buffer. This flag is reset by a read from port SBBase+Ah.
6:0	–	Same as SBBase+Ch[6:0].

Programmed I/O Access to FIFO Register (SBBase+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 with programmed I/O.

MPU-401 Device

MPU-401 Data **(MPUBase+0h, R/W)**

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

This register is used to read data from the MPU-401 receive FIFO or a command acknowledge byte (0FEh). This register is also used to write data to the MPU-401 transmit FIFO.

MPU-401 Command **(MPUBase+1h, W)**

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

The MPU-401 device accepts only two commands:

FFh	Reset/return to Smart mode. This command generates an acknowledge byte if received when already in Smart mode.
3Fh	Go to UART mode. This command generates an acknowledge byte if received while in Smart mode. It is ignored if the device is already in UART mode.

MPU-401 Status **(MPUBase+1h, R)**

-RR	-TR	X	X	X	X	X	X
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	-RR	0 = read data available in the receive FIFO, or pending acknowledge byte to be read (0FEh).
6	-TR	0 = there is room in the transmit FIFO to accept another byte.

Game Port Device

The joystick device uses only a single I/O port.

Joystick **(GPBase+1h W)**

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0

Any value written to the GPBase+1h port will restart the timing sequence. This should be done before reading the timer status flags.

Joystick **(GPBase+1h, R)**

SWD	SWC	SWB	SWA	TD	TC	TB	TA
7	6	5	4	3	2	1	0

SW(A-D) return the current state of the joystick switch inputs. T(A-D) return the current state of the four one-shot timers connected to the X and Y resistors of the dual joysticks.

DMAC Device

The DMAC device is used for the Audio 1 DMA controller internal to the ES1946.

DDMA Current/Base Address (DDMABase+0h – DDMABase+2h, R/W)

23	22	21	20	19	18	17	16								
DMA current/base address															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write to the DMA base address register, read from the DMA current address register. The current address is automatically copied from the base register when DMA starts. The current address is then incremented or decremented according to bit 5 of DDMABase+Bh.

DDMA Current/Base Count (DDMABase+4h – DDMABase+5h, R/W)

DMA current/base count															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write to the DMA base count register, read from the DMA current count register. The current count is automatically copied from the base register when DMA starts. The current count is then decremented for each byte transferred.

DMA Command (DDMABase+8h, W)

DACKPOL	DREQPOL	0	0	0	EN	0	0
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	DACKPOL	DACK signal polarity.
6	DREQPOL	DREQ signal polarity.
5:3	–	Reserved. Always write 0.
2	EN	Controller enable.
1:0	–	Reserved. Always write 0.

DMA Status (DDMABase+8h, R)

RQ				TC			
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:4	RQ	0000 = DREQ is asserted. 1111 = DREQ is negated.
3:0	TC	0000 = Has not reached terminal count (TC). 1111 = Has reached TC.

DMA Mode (DDMABase+Bh, W)

TMODE		DIR	AI	TTYPE		0	0
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:6	TMODE	

Bit 7	Bit 6	Transfer Mode
0	0	Demand transfer
0	1	Single transfer
1	0	Block transfer
1	1	Reserved

Bits	Name	Description
5	DIR	Transfer direction. 1 = Address decrement. 0 = Address increment.
4	AI	Auto-Initialize.
3:2	TTYPE	

Bit 3	Bit 2	Transfer Type
0	0	Verify transfer
0	1	Write transfer
1	0	Read transfer
1	1	Illegal

Bits	Name	Description
1:0	–	Reserved. Always write 0.

DMA Master Clear (DDMABase+Dh, W)

DMA master clear							
7	6	5	4	3	2	1	0

Write any value to this register to reset DMAC.

DMA Mask (DDMABase+Fh, R/W)

0	0	0	0	0	0	0	Mask
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:1	–	Reserved. Always write 0.
0	Mask	Mask the DREQ.

Mixer Registers

There are two types of mixer registers. Sound Blaster Pro Support mixer registers fully support the Sound Blaster Pro. ESS mixer registers are specific to ESS Technology, Inc. ES1946 *Audio***Drive®** chips, although many registers are shared throughout the *Audio***Drive®** family of chips.

Sound Blaster Pro Support Mixer Registers

This section provides a summary of mixer registers which support Sound Blaster Pro in the ES1946 and some comments on the characteristics of these registers.

Table 28 Sound Blaster Support Register Summary

Reg	D7	D6	D5	D4	D3	D2	D1	D0	Description
00h	Write: reset mixer								Mixer reset
04h	DAC play volume left			X	DAC play volume right			X	DAC playback volume
0Ah	X	X	X	X	X	Mic mix volume		X	Mic mix volume
0Ch	X	X	F1 *	X	F0 *	ADC source		X	See note for F0, F1.
0Eh	X	X	F2 *	X	X	X	Stereo	X	See note for F2.
22h	Master volume left			X	Master volume right			X	Master volume
26h	FM volume left			X	FM volume right			X	Music DAC volume
28h	AuxA (CD) volume left			X	AuxA (CD) volume right			X	AuxA (CD) volume
2Eh	Line volume left			X	Line volume right			X	Line volume

* Sound Blaster Filter Control bits F2, F1, and F0 have no equivalent function in the ES1946 and are ignored.

Filter Control Bits

The Sound Blaster Pro mixer has three bits that control input and output filters. They are labeled F0, F1, and F2 in Table 28 and Table 29. They have no equivalent function in the ES1946 and their values are ignored.

Mixer Stereo Control Bit

Bit 1 of register 0Eh is the Mixer Stereo Control bit. It is normally zero. Set this bit high to enable legacy DOS compatible stereo DAC functions. In this case, program the DAC sample rate to be twice the sample rate of each channel. For example, for 22 kHz stereo, program the "sample rate" to be 44 kHz using command 40h.

This bit enables stereo only for DMA transfer to the DAC in Compatibility mode. It should not be used in Extended mode.

Clear this bit after completing the stereo DMA transfer, because this bit is unaffected by software reset (only mixer reset).

See also "Stereo DMA Transfers in Compatibility Mode" on page 26.

ESS Mixer Registers

This section provides a summary of the ESS mixer registers followed by a detailed description of each register.

Table 29 ESS Mixer Registers Summary

Reg	D7	D6	D5	D4	D3	D2	D1	D0	Remark
00h	Write: reset mixer								Reset mixer
14h	Audio 1 play volume left				Audio 1 play volume right				Audio 1 play volume
1Ah	Mic mix volume left				Mic mix volume right				Mic mix volume
1Ch	x	x	F1	Mute	F0	Extended record source			
32h	Master volume left				Master volume right				Master volume
36h	FM volume left				FM volume right				FM volume
38h	AuxA (CD) volume left				AuxA (CD) volume right				AuxA (CD) volume
3Ah	AuxB volume left				AuxB volume right				AuxB volume
3Ch					PC speaker volume				PC speaker volume
3Eh	Line volume left				Line volume right				Line volume
48h	0	0	0	ES689/ 69x I/F enable	0	0	0	0	Serial mode miscellaneous control

Table 29 ESS Mixer Registers Summary (Continued)

Reg	D7	D6	D5	D4	D3	D2	D1	D0	Remark
50h	0	0	0	0	1:Enable 3-D	0:Reset	0	0	3-D enable
52h	0	0	3-D level						3-D level
60h	0	1:Mute	Left master volume						Left master volume and mute
61h	0	1:Mute	Left volume counter						Left hardware volume counter
62h	0	1:Mute	Right master volume						Right master volume and mute
63h	0	1:Mute	Right volume counter						Right hardware volume counter
64h	1:Split mode	MPU-401 int mask	1:Count by 3	Read-only HWV int request	Mode		HWV int mask	Disable SB Pro master vol control	Master volume control
66h	Clear hardware volume interrupt request								Write-only
68h	Mic record volume left				Mic record volume right				Mic record volume
69h	Audio 2 record volume left				Audio 2 record volume right				Audio 2 record volume
6Ah	AuxA (CD) record volume left				AuxA (CD) record volume right				AuxA (CD) record volume
6Bh	Music DAC record volume left				Music DAC record volume right				Music DAC record volume
6Ch	AuxB record volume left				AuxB record volume right				AuxB record volume
6Dh	Left Mono_In play mix				Right Mono_In play mix				Mono_In play mix
6Eh	Line record volume left				Line record volume right				Line record volume
6Fh	Mono_In record volume left				Mono_In record volume right				Mono_In record volume
70h	Master clock	Two's complement rate divider							Audio 2 sample rate
71h	0	0	1:New reg A1h	1:4x mode	1	1:SCF1 bypass	1:Async mode	1:FM mix	Audio 2 mode
74h	Two's complement transfer count – low byte								Audio 2 transfer count reload
76h	Two's complement transfer count – high byte								
78h	0	0	0	1: Auto-initialize	0	0	Enable second channel DMA	Enable FIFO to 2nd chan DAC	Audio 2 control 1
7Ah	2nd channel IRQ	IRQ mask	0	0	0	Data sign	Stereo /Mono	16-bit/ 8-bit	Audio 2 control 2
7Ch	Left channel volume				Right channel volume				Audio 2 DAC mixer volume
7Dh	0	0	0	0	Enable mic amp	Mono_Out source select		Enable Mono_In mix w/ AOUTL/R	Mic preamp, Mono_In and Mono_Out
7Fh	Reserved			Music digital record	I²S Data Activity	I²S clock activity	x	I²S connect to music DAC	I²S interface

Register Detailed Descriptions

Reset Mixer (00h, W)

Write: reset mixer							
7	6	5	4	3	2	1	0

A write to this register resets the mixer registers to their default values.

Table 30 Mixer Register Default Values

Register Name	Address	Value	
Sound Blaster Registers			
DAC playback volume	04h	99h	middle
Mic mix volume	0Ah	11h	mute
Record source	0Ch	11h	mic
Stereo select	0Eh	11h	mono
Master volume	22h	99h	middle
Music DAC volume	26h	99h	middle
AuxA (CD) volume	28h	11h	mute
Line volume	2Eh	11h	mute
Extended Mode Registers			
Audio 1 play volume	14h	88h	middle
Mic mix volume	1Ah	11h	mute
Record source	1Ch	11h	mic
Master volume	32h	88h	middle
FM volume	36h	88h	middle
AuxA (CD) volume	38h	00h	mute
AuxB volume	3Ah	00h	mute
PC speaker volume	3Ch	04h	middle
Line volume	3Eh	11h	mute

Audio 1 Play Volume (14h, R/W)

Audio 1 play volume left				Audio 1 play volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the first audio channel. On reset, this register assumes the value of 88h.

Mic Mix Volume (1Ah, R/W)

Mic mix volume left				Mic mix volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the Mic input. On reset, this register assumes the value of 00h.

Extended Record Source (1Ch, R/W)

x	x	F1	Mute	F0	Extended record source		
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:6	–	No function.
5	F1	Sound Blaster Filter Control bit F1 has no equivalent function in the ES1946 and is ignored.
4	Mute	1 = Mutes the input to the filters for recording. This does not affect MONO_OUT.
3	F0	Sound Blaster Filter Control bit F0 has no equivalent function in the ES1946 and is ignored.
2:0	Extended record source	Selects the record source in Extended mode.

Bit 2	Bit 1	Bit 0	Record Source
0	0	0	Microphone
0	0	1	Reserved
0	1	0	AuxA (CD)
0	1	1	Reserved
1	0	0	Microphone
1	0	1	Record mixer
1	1	0	Line
1	1	1	Reserved

Master Volume Register (32h, R/W)

Master volume left				Master volume right			
7	6	5	4	3	2	1	0

On reset, this register assumes the value of 88h.

This register provides backward-compatible access to master volume. New applications can also use registers 60h and 62h, which have more resolution.

FM Volume Register (36h, R/W)

FM volume left				FM volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the music DAC. On reset, this register assumes the value of 88h.

AuxA (CD) Volume Register (38h, R/W)

AuxA (CD) volume left				AuxA (CD) volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the CD audio input. On reset, this register assumes the value of 00h.

AuxB Volume Register (3Ah, R/W)

AuxB volume left				AuxB volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the auxiliary line input. On reset, this register assumes the value of 00h.

PC Speaker Volume Register (3Ch, R/W)

Reserved					PC speaker volume		
7	6	5	4	3	2	1	0

This register controls the PC speaker volume. Bits 2:0 select the attenuation level in steps of -3 dB. On reset, this register assumes the value of 04h.

Line Volume Register (3Eh, R/W)

Line volume left				Line volume right			
7	6	5	4	3	2	1	0

This register controls the playback volume of the line input. On reset, this register assumes the value of 00h.

Serial Mode Miscellaneous Control (48h, R/W)

0	0	0	ES689/ ES69x interface enable	0	0	0	0
7	6	5	4	3	2	1	0

Bits Definitions:

Bits	Name	Description
7:5	–	Reserved. Always write 0.
4	ES689/ ES69x interface enable	1 = Enable ES689/ES69x serial interface to use the music DAC. MCLK must also go high at least once every 20 µsec or the DAC will revert to FM. The mixer volume for the music DAC is controller by mixer register 36h. 0 = Disable ES689/ES69x serial interface.
3:0	–	Reserved. Always write 0.

3-D Enable and Mode (50h, R/W)

0	0	0	0	Enable 3-D	Reset	0	0
7	6	5	4	3	2	1	0

3-D effect uses *Spatializer*® VBX™ technology, provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc.

Bit Definitions:

Bits	Name	Description
7:4	0	Reserved. Always write 0.
3	Enable 3-D	1 = Enable 3-D effect. This bit is ignored if the record monitor is enabled, register A8h bit 3. 0 = Disable 3-D effect (effect unit bypassed).
2	Reset	1 = Release from reset. 0 = Reset 3-D effect.
1	–	Reserved. Always write 0.
0	–	Reserved. Always write 0.

3-D Level (52h, R/W)

0	0	3-D level					
7	6	5	4	3	2	1	0

3-D effect uses *Spatializer*® VBX™ technology, provided by Desper Products, Inc., a subsidiary of Spatializer Audio Laboratories, Inc.

Reset to zero by hardware reset.

Bit Definitions:

Bits	Name	Description
7:6	0	Reserved. Always write 0.
5:0	3-D level	0 is minimum effect; 3Fh is maximum effect.

Left Master Volume and Mute (60h, R/W)

0	1:Mute	Left master volume					
7	6	5	4	3	2	1	0

Right Master Volume and Mute (62h, R/W)

0	1:Mute	Right master volume					
7	6	5	4	3	2	1	0

Registers 60h and 62h are the actual logarithmic volume values presented to the analog hardware in 0.75 dB steps. These registers can be modified under five circumstances:

- Hardware reset. Each register is reset to 36h.
- Direct write to mixer address 60h or 62h.
- If bit 0 of mixer register 64h is low, then writing to mixer registers 22h or 32h updates 60h and 62h.
- If bit 0 of mixer register 64h is low, then a mixer reset (writing to mixer register 00h) loads these registers with 36h.
- If hardware volume controls are enabled and bit 7 of mixer register 64h is low, then the hardware volume controls can directly modify the contents of these registers.

Reading mixer registers 22h or 32h actually reads a value calculated from the current contents of 60h and 62h. See “Sound Blaster Pro Volume Emulation” on page 36.

Left Hardware Volume Control Counter (61h, R/W)

0	1:Mute	Left volume counter					
7	6	5	4	3	2	1	0

Right Hardware Volume Control Counter (63h, R/W)

0	1:Mute	Right volume counter					
7	6	5	4	3	2	1	0

These registers only exist if bit 7 of mixer register 64h is high. If bit 7 is low, these registers are combined with registers 60h and 62h and cannot be independently written or read.

If bit 7 of mixer register 64h is high, these registers have no connection with registers 60h or 62h. They are the hardware volume counters and mute. It is the responsibility of the host software to read these registers and update the master volume registers 60h and 62h.

Master Volume Control (64h, R/W)

Split mode	MPU-401 int mask	Count by 3	Read-only HMV int request	Mode		HWV int mask	Disable SB Pro master vol emulation
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	Split mode	1 = Split counter registers from volume registers and access them independently. 0 = Slave counter and volume registers together.
6	MPU-401 int mask	This bit is AND'ed with the MPU-401 interrupt request. If it is low, the MPU-401 interrupt request stays low. This bit is cleared by hardware reset.
5	Count by 3	This bit is cleared by hardware reset. 1 = Count up and down by 3 for each push of Up or Down buttons. 0 = Count up and down by 1 for each push of Up or Down buttons.
4	HMV int request	Read-only interrupt request from hardware volume event.
3:2	Mode	Selects operation mode:

Bit 3	Bit 2	Operating Mode
0	0	Reserved
0	1	2-wire mode: both Up and Down inputs being low together act the same as Mute input low. (default)
1	0	Reduced debounce (10 µsec vs. 40 msec), 2-wire mode, auto-increment and -decrement disabled.
1	1	Hardware volume control disabled

1	HWV int mask	This bit is AND'ed with the hardware volume interrupt request before being OR'd with the first channel audio interrupt request. If this bit is low, the hardware volume interrupt request does not get OR'd with the first channel audio interrupt request. This bit is cleared by hardware reset.
0	Disable SB Pro master vol emulation	When low, a write to Sound Blaster Pro master volume registers 22h or 32h is translated into a write to hardware master volume registers 60h and 62h. Also, if low, a mixer reset (writing to mixer register 0h) causes registers 60h and 62h to be reset to default value 36h. When high, the Sound Blaster Pro master volume registers are, in effect, read-only. This bit is cleared by hardware reset.

Clear Hardware Volume Interrupt Request (66h, R/W)

Clear Hardware Master Control							
7	6	5	4	3	2	1	0

Any write to this register resets the hardware volume interrupt request.

Mic Record Volume (68h, R/W)

Left Mic record				Right Mic record			
7	6	5	4	3	2	1	0

This register controls the record volume for the mic input. Set low by hardware reset but not by mixer reset.

Audio 2 Record Volume (69h, R/W)

Left Audio 2 record				Right Audio 2 record			
7	6	5	4	3	2	1	0

This register controls the record volume for the second audio channel input. Set low by hardware reset but not by mixer reset.

AuxA (CD) Record Volume (6Ah, R/W)

Left AuxA (CD) record				Right AuxA (CD) record			
7	6	5	4	3	2	1	0

This register controls the record volume for the CD input. Set low by hardware reset but not by mixer reset.

Music DAC Record Volume (6Bh, R/W)

Left music DAC record				Right music DAC record			
7	6	5	4	3	2	1	0

This register controls the record volume for the music DAC input (FM, wavetable, or I²S). Set low by hardware reset but not by mixer reset.

AuxB Record Volume (6Ch, R/W)

Left AuxB record				Right AuxB record			
7	6	5	4	3	2	1	0

This register controls the record volume for the auxiliary line input. Set low by hardware reset but not by mixer reset.

Mono_In Playback Volume (6Dh, R/W)

Left Mono_In play mix				Right Mono_In play mix			
7	6	5	4	3	2	1	0

This register controls the playback volume for the mono input.

Line Record Volume (6Eh, R/W)

Left Line record				Right Line record			
7	6	5	4	3	2	1	0

This register controls the record volume for the line input. Set low by hardware reset but not by mixer reset.

Mono_In Record Volume (6Fh, R/W)

Left Mono_In record				Right Mono_In record			
7	6	5	4	3	2	1	0

This register controls the record volume for the mono input. Set low by hardware reset but not by mixer reset.

Audio 2 Sample Rate Generator (70h, R/W)

Clock source	Sample rate divider						
7	6	5	4	3	2	1	0

Program this register for the sample rate for Audio 2 DAC operations in extended mode.

The sample rate is determined by the two's complement divider in bits 6:0.

Sample_Rate = Clock_Source / (128 - Sample_Rate_Divider)

This register is reset to zero by hardware reset.

Bits Definitions:

Bits	Name	Description
7	Clock source	1 = 768 kHz (used to generate 48 kHz, 32 kHz, 16 kHz, 8 kHz, etc.). 0 = 793.8 kHz (used to generate 44.1 kHz, 22.05 kHz, etc.).
6:0	Sample rate divider	Signed sample rate divider of master clock. For example:

Value	Sample Rate	Condition
20h	8000	bit 7 = 1
70h	48000	bit 7 = 1
6Eh	44100	bit 7 = 0

Audio 2 Mode (71h, R/W)

0	0	New reg A1h	4x mode	–	SCF1 bypass	Async mode	FM mix
7	6	5	4	3	2	1	0

This register is reset to zero by hardware reset.

Bit Definitions:

Bits	Name	Description
7:6	0	Reserved. Always write 0.
5	New reg A1h	1 = Register A1h behaves in the same manner as mixer register 70h, which gives more accurate sample rates that are divisors of 48 kHz. 0 = Enables register A1h to behave exactly as in previous ESS *Audio***Drive**® chips.
4	4x mode	1 = 2nd channel DAC is in 4x oversampling mode. 0 = 2nd channel DAC is not oversampling.
3	–	Reserved.
2	SCF1 bypass	1 = 1st channel CODEC switched capacitor filter is bypassed. 0 = 1st channel CODEC SCF is not bypassed.
1	Async mode	1 = 2nd channel DAC is asynchronous to the sample rate of the 1st channel. 0 = 2nd channel DAC is slaved to the sample rate and filter rate of the 1st channel.
0	FM mix	1 = 2nd channel DMA is slaved to the FM synthesizer sample rate and the DMA data is digitally mixed to the FM synthesizer output.

Audio 2 Transfer Count Reload (74h, R/W)

Two's complement transfer count – low byte							
7	6	5	4	3	2	1	0

Audio 2 Transfer Count Reload (76h, R/W)

Two's complement transfer count – high byte							
7	6	5	4	3	2	1	0

NOTE: Audio 2 transfer counts by words not bytes.

Audio 2 Control 1 (78h, R/W)

0	0	0	Auto-initialize	0	0	Enable 2nd chan DMA	Enable FIFO to 2nd chan DAC
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:6	0	Reserved. Always write 0.
5	0	Reserved.
4	Auto-initialize	1 = Auto-initialize mode. After the transfer counter rolls over to 0, it is automatically reloaded and DMA continues. The second channel interrupt flag will be set high. 0 = Normal mode. After the transfer counter rolls over to 0, it is reloaded but DMA stops. Bit 1of this register is cleared. The 2nd channel interrupt flag will be set high.
3:2	0	Reserved. Always write 0.
1	Enable 2nd chan DMA	1 = 2nd channel DMA enabled for data to be written into the 2nd channel FIFO (32 words deep). 0 = Second channel DMA not enabled. This bit is cleared when the transfer counter rolls over to zero, if not in Auto-initialize mode.
0	Enable FIFO to 2nd Chan DAC	1 = Data transfer from the FIFO to the 2nd channel DAC is enabled. 0 = Data transfer not enabled.

Audio 2 Control 2 (7Ah, R/W)

2nd chan IRQ	IRQ mask	0	0	0	Signed	Stereo/ mono	16-bit/ 8-bit
7	6	5	4	3	2	1	0

This register is reset to zero by hardware or software reset.

Bit Definitions:

Bits	Name	Description
7	2nd channel IRQ	This latch is set high when the DMA counter rolls over to zero, or when a 1 is written to this bit. The latch is cleared by writing a zero to this bit, or by hardware or software reset.
6	IRQ mask	This bit is AND'ed with bit 7 to produce the second DMA channel interrupt request.
5:3	0	Reserved. Always write 0.
2	Signed	1 = Data is in signed, two's complement format. 0 = Unsigned data.
1	Stereo/ mono	1 = Stereo data. 0 = Mono data.
0	16-bit/ 8-bit	1 = 16-bit samples. 0 = 8-bit samples.

Audio 2 DAC Mixer Volume (7Ch, R/W)

Left channel volume				Right channel volume			
7	6	5	4	3	2	1	0

This register is reset to zero by hardware reset.

Mic Preamp, MONO_IN and MONO_OUT (7Dh, R/W)

0	0	0	0	Enable mic amp	MONO_OUT source select		Enable MONO_IN mix with AOUT_L/R
7	6	5	4	3	2	1	0

This register is reset to 08h by hardware reset.

Bit Definitions:

Bits	Name	Description
7:4	0	Reserved. Always write 0.
3	Enable mic amp	1 = Enable +20 dB microphone preamp gain. 0 = Mic preamp is 0 dB.
2:1	MONO_OUT source select	Selects MONO_OUT source.
0	Enable MONO_IN mix with AOUT_L/R	1 = MONO_IN is mixed with AOUT_L and AOUT_R after the master volume, bypassing the mixers. 0 = The MONO_IN mixer bypass is disabled.

Bit 2	Bit 1	Source Select
0	0	Mute (CMR)
0	1	1st channel DAC, right channel playback, after filter stage.
1	0	2nd channel DAC, right channel output.
1	1	Mono mix of left and right record level stage outputs. MONO_OUT is controlled by record source select and record level registers.

I²S Interface (7Fh, R/W)

Reserved	Music digital record	I²S Data Activity	I²S clock activity	x	Enable I²S connect to music DAC
7 6 5	4	3	2	1	0

The I²S interface and the ES689/ES69X hardware wavetable interface share the same pins (#22 and #23); therefore, the I²S interface and ES689/ES69X interface are mutually exclusive.

Bit Definitions:

Bits	Name	Description
7:5	–	Reserved.
4	Music digital record	1 = Enable direct digital recording of Music DAC data (including FM, ES689/ES69x, or I²S). In this mode, the first DMA channel must be enabled for stereo recording. The sample rate is determined by the music DAC sample rate rather than by controller register A1h.
3	–	This bit is set high if IISDATA has been high at least once since it was cleared by software.
2	I²S clock activity	This bit is set high if both IISCLK and IISLR have been high at the same time at least once since the last time it was last cleared by software.
1	–	Reserved.
0	Enable I²S connect to Music DAC	1 = Enable I²S serial interface to acquire control of music DAC. 0 = Allow FM synthesizer or ES689/ES69x serial interface to use DAC. This bit in conjunction with bit 4 of register 48h controls which source uses the music DAC.

Table 31 Music DAC Source

Reg. 48h [4]	Reg 7Fh [0]	Music DAC Source
0	0	FM Synthesizer
0	1	I²S Interface
1	0	ESFM or HWWT by auto -detect of HWWT I/F
1	1	I²S Interface

Controller Registers

This is a summary and description of the controller registers. These registers are written to and read from using commands of the format Axh or Bxh. To enable access to these registers send the command C6h.

Table 32 ESS Controller Registers Summary

Reg	D7	D6	D5	D4	D3	D2	D1	D0	Description
A1h	Clock source	Sample rate divider							
A2h	Filter clock divider								S/W reset, setup for 8 kHz sampling
A4h	DMA transfer counter reload – low byte								
A5h	DMA transfer counter reload – high byte								
A8h	0	0	0	1	Enable record monitor	0	Mono/stereo select		Analog control
B1h	Game compatible IRQ	Enable IRQ ovf Ext mode DMA cntr	Enable IRQ for FIFO1 HE status edge	x	Reserved				Legacy audio interrupt control
B2h	Game compatible DRQ	Enable DRQ for Ext mode DMA	Enable DRQ game compatible DMA	x	Reserved				Audio DRQ control
B4h	Right channel record level				Left channel record level				Record level
B7h	Enable FIFO to/from CODEC	Reserved. Set opposite polarity of bit 3	Signed	1	Stereo/ Mono mode select	16-bit/8-bit mode select	0	1	Audio 1 control 1
B8h	0	0	0	0	CODEC mode	Auto-initialize	DMA read/ write	Trans-fer enable	Audio 1 control 2
B9h	0	0	0	0	0	0	Transfer type		Audio 1 transfer type
BAh	0		Reserved	Sign	Adjust magnitude				Left channel ADC offset adjust
BBh	0			Sign	Adjust magnitude				Right channel ADC offset adjust

Controller Register Descriptions

Audio 1 Sample Rate Generator (A1h, R/W)

Clock source	Sample rate divider						
7	6	5	4	3	2	1	0

Program this register for the sample rate for Audio 1 DAC operations in extended mode.

This register is reset to zero by hardware reset.

How the sample rate is programmed varies depending of the value of bit 5 of mixer register 71h.

If Bit 5 = 1

If bit 5 of mixer register 71h = 1 then this register is programmed as the Audio 2 Sample Rate Generator register 70h.

The sample rate is determined by the two's complement divider in bits 6:0.

Sample_Rate = Clock_Source / (128 - Sample_Rate_Divider)

Bits Definitions:

Bits	Name	Description
7	Clock source	1 = 768 kHz (used to generate 48 kHz, 32 kHz, 16 kHz, 8 kHz, etc.). 0 = 793.8 kHz (used to generate 44.1 kHz, 22.05 kHz, etc.).
6:0	Sample rate divider	Signed sample rate divider of master clock. For example:

Value	Sample Rate	Condition
20h	8000	bit 7 = 1
70h	48000	bit 7 = 1
6Eh	44100	bit 7 = 0

If Bit 5 = 0

If bit 5 of mixer register 71h = 0 then this register is programmed as follows:

The sample rate is determined by the two's complement divider in bits 6:0.

Sample_Rate = Clock_Source / (128 - Sample_Rate_Divider)

Bits Definitions:

Bits	Name	Description
7	Clock source	1 = 793.8 kHz (for rates > 22.05 kHz). 0 = 396.9 kHz (for rates <= 22.05 kHz).
6:0	Sample rate divider	Signed sample rate divider of master clock.

Audio 1 Filter Clock Divider (A2h, R/W)

Filter clock divider							
7	6	5	4	3	2	1	0

This register controls the low-pass frequency of the switch-capacitor filters inside the ES1946. Generally, the filter roll-off should be positioned at 80% - 90% of the Sample_Rate/2 frequency. The ratio of the roll-off frequency to the filter clock frequency is 1:82. In other words, first determine the desired roll-off frequency by taking 80% of the Sample_Rate divided by 2, then multiply by 82 to find the desired filter clock frequency. Use the formula below to determine the closest divider:

Filter_Clock_Frequency = 7.16 MHz / (256-Filter_Divider_Register)

DMA Transfer Count Reload (A4h, R/W)

DMA transfer count reload – low byte							
7	6	5	4	3	2	1	0

On reset, this register assumes the value of 00h.

DMA Transfer Count Reload (A5h, R/W)

DMA transfer count reload – high byte							
7	6	5	4	3	2	1	0

On reset, this register assumes the value of F8h.

The FIFO control logic of the ES1946 has a 16-bit counter for controlling transfers to and from the FIFO. These registers are the reload value for that counter which is the value that gets copied into the counter after each overflow (plus at the beginning of the initial DMA transfer). The counter is incremented after each successful byte is transferred by DMA. Since the counter counts up towards FFFFh and then overflows, the reload value is in two's complement form.

For Auto-Initialize mode DMA, the counter is used to generate interrupt requests to the system processor. In this mode, the ES1946 allows continuous DMA. In a typical application the counter is programmed to be one-half of the DMA buffer maintained by the system processor. In this application an interrupt is generated whenever DMA switches from one half of the circular buffer to the other.

For Normal mode DMA, DMA requests are halted at the time that the counter overflows, until a new DMA transfer is commanded by the system processor. Again, an interrupt request is generated to the system processor if bit 6 of register B1h is set high.

Analog Control (A8h, R/W)

0	0	0	1	Record monitor enable	0	Stereo/mono select	
7	6	5	4	3	2	1	0

When programming the FIFO for DMA playback modify only bits 1:0. When programming the FIFO for DMA record modify only bits 3, 1, and 0. Read this register first to preserve the remaining bits.

Bit Definitions:

Bits	Name	Description
7:5	0	Reserved. Always write 0.
4	1	Reserved. Always write 1.
3	Record monitor enable	1 = Enable record monitor. This bit is ignored unless register B8h bit 3 is high (the CODEC is in ADC mode). 0 = Disable record monitor.
2	0	Reserved. Always write 0.
1:0	Stereo/ mono select	Select operation mode of first DMA converters.

Bit 1	Bit 0	Mode
0	0	Reserved
0	1	Stereo
1	0	Mono
1	1	Reserved

Legacy Audio Interrupt Control (B1h, R/W)

Game compatible IRQ	Enable IRQ ovf Ext mode DMA cntr	Enable IRQ for FIFO1 HE status edge	x	Reserved			
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	Game compatible IRQ	Reserved for Compatibility mode. Leave zero for Extended mode.
6	Enable IRQ ovf Ext mode DMA cntr	Set high to receive interrupts for each overflow of the ES1946 DMA counter in Extended mode.
5	Enable IRQ for FIFO1 HE status edge	Set high to receive interrupts for FIFO Half-Empty transitions when doing block I/O to/ from the FIFO in Extended mode.
4	–	No function. The audio device activate bit serves the purpose of enabling the interrupt pin.
3:0	–	Reserved.

DRQ Control (B2h, R/W)

Game compatible DRQ	Enable DRQ for Extended mode DMA	Enable DRQ game compatible DMA	x	Reserved			
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	Game compatible DRQ	Reserved for Compatibility mode. Leave zero for Extended mode.
6	Enable DRQ for Extended mode DMA	1 = Enable DRQ outputs and DACKB inputs for DMA transfers in Extended mode. 0 = Enable block I/O to/from the FIFO in Extended mode.
5	Enable DRQ game compatible DMA	Reserved for Compatibility mode. Leave zero for Extended mode.
4	–	No function. The DRQ lines always drive (there is no enable). If neither bit 5 nor bit 6 are set high, the first audio DRQ is always low.
3:0	–	Reserved.

Record Level (B4h, R/W)

Right channel record level				Left channel record level			
7	6	5	4	3	2	1	0

Register B4h allows for independent right and left record level. Each channel has 16 levels (excluding mute). The amount of gain or attenuation for each level is different for microphone than for all other sources. The record levels are listed in the following table.

Record Level	Gain for Mic	Gain for Other Sources
0	+0 dB	-6.0 dB
1	+1.5 dB	-4.5 dB
2	+3.0 dB	-3.0 dB
3	+4.5 dB	-1.5 dB
4	+6.0 dB	0 dB
5	+7.5 dB	+1.5 dB
6	+9.0 dB	+3.0 dB
7	+10.5 dB	+4.5 dB
8	+12.0 dB	+6.0 dB
9	+13.5 dB	+7.5 dB
10	+15.0 dB	+9.0 dB
11	+16.5 dB	+10.5 dB
12	+18.0 dB	+12.0 dB
13	+19.5 dB	+13.5 dB
14	+21.0 dB	+15.0 dB
15	+22.5 dB	+16.5 dB

Audio 1 Control 1 (B7h, R/W)

Enable FIFO to/from CODEC	Set oppo-site bit 3	Signed	1	FIFO stereo mode	FIFO 16-bit mode	0	Generate load signal
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7	Enable FIFO to/from CODEC	1 = Enable first DMA FIFO connection to DAC or ADC. This allows transfers to/from the FIFO and the analog circuitry. 0 = Disable first DMA FIFO connection to DAC or ADC.
6	Set opposite bit 3	Reserved function. This bit must be set to the opposite polarity of bit 3: high for mono and low for stereo.
5	Signed	1 = First DMA FIFO two's complement mode (signed data). 0 = First DMA FIFO unsigned (offset 8000).
4	1	Reserved. Always write 1.
3	FIFO stereo mode	1 = First DMA FIFO stereo mode. 0 = First DMA FIFO mono mode. Bit 6 must be set at the opposite polarity of this bit: high for mono, low for stereo.
2	FIFO 16-bit mode	1 = First DMA FIFO 16-bit mode. 0 = First DMA FIFO 8-bit mode.
1	0	Reserved. Always write 0.
0	Generate load signal	Write 1. Generates a load signal that copies DAC Direct Access Holding register to DAC on the next sample rate clock edge (sample rate is determined by Extended mode register A1h). This bit is cleared after the holding register is copied to the DAC.

Audio 1 Control 2 (B8h, R/W)

0	0	0	0	CODEC mode	Auto-initialize	DMA read enable	DMA transfer enable
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:4	0	Reserved. Always write 0.
3	CODEC mode	1 = first DMA converters in ADC mode. 0 = first DMA converters in DAC mode.
2	Auto-initialize	1 = auto-initialize mode. 0 = normal DMA mode.
1	DMA read enable	1 = first DMA is read (e.g. for ADC operation). 0 = first DMA is write (e.g. for DAC operation).
0	DMA transfer enable	First DMA active-low reset. When high, first DMA is allowed to proceed.

Audio 1 Transfer Type (B9h, R/W)

0	0	0	0	0	0	DMA transfer type select	
7	6	5	4	3	2	1	0

Bit Definitions:

Bits	Name	Description
7:2	0	Reserved. Always write 0.
1:0	DMA transfer type select	Selects the DMA transfer type for the first DMA:

Bit 1	Bit 0	Transfer Type	Bytes/DMA Request
0	0	Single	1
0	1	Demand	2
1	0	Demand	4
1	1	Reserved	–

Left Channel ADC Offset Adjust (BAh, R/W)

0	0	Disable time delay on analog wake-up	Sign	Adjust magnitude			
7	6	5	4	3	2	1	0

This register is reset to zero by hardware reset and is unaffected by software reset.

Bit Definitions:

Bits	Name	Description
7:6	0	Reserved. Always write 0.
5	Disable time delay on analog wake-up	Normally, the AOUT_L and AOUT_R pins are muted for 100 msec ± 20 msec after hardware reset or after the analog sub-systems wake from power-down. Set high to disable delay. This bit is cleared by hardware reset.
4:0	Sign/Adjust magnitude	See the explanation for bits 4:0 following register BBh.

Right Channel ADC Offset Adjust (BBh, R/W)

0	0	0	Sign	Adjust magnitude			
7	6	5	4	3	2	1	0

This register is reset to zero by hardware reset and is unaffected by software reset.

Bit Definitions:

Bits	Name	Description
7:5	0	Reserved. Always write 0.
4:0	Sign/Adjust magnitude	See the following explanation for bits 4:0.

Bits 4 (sign) and 3:0 (adjust magnitude) of the ADC Offset Adjust registers cause a constant value to be added to the ADC converter output, as shown in the following:

Code	Offset	Code	Offset
00h	0	10h	-64
01h	+64	11h	-128
02h	+128	12h	-192
03h	+192	13h	-256
04h	+256	14h	-320
05h	+320	15h	-384
06h	+384	16h	-448
07h	+448	17h	-512
08h	+512	18h	-576
09h	+576	19h	-640
0Ah	+640	1Ah	-704
0Bh	+704	1Bh	-768
0Ch	+768	1Ch	-832
0Dh	+832	1Dh	-896
0Eh	+896	1Eh	-960
0Fh	+960	1Fh	-1024

Formula:

bit 4 = 0: offset = 64 * bits[3:0]

bit 4 = 1: offset = -64 * (bits[3:0] +1)

To calculate the offset adjust code, first measure the ADC offset for both right and left channels before adjustment by following these steps:

1. Program Extended mode registers BAh and BBh bits 4:0 to be zero (no digital offset).
2. Select a zero-amplitude (or low amplitude) recording source.
3. Set the recording volume to minimum by setting Extended mode register B4h to zero.
4. Make a stereo 16-bit two's complement recording at 11 kHz sample rate of 2048 stereo samples (2048 stereo samples = 4096 words = 8192 bytes, which is about 190 milliseconds).
5. Use the last 1024 stereo samples to calculate a long term average for both left and right channels.
6. With this average DC offset, calculate the best digital offset to bring the sum closest to zero, using the codes and offsets listed in the table above.

AUDIO MICROCONTROLLER COMMAND SUMMARY

Table 33 Command Summary

Command	Data Byte(s) Write/Read	Function
10h	1 write	Direct write 8-bit DAC. Data is 8-bit unsigned format.
11h	2 writes	Direct write 16-bit DAC. Data is 16-bit unsigned format, first low byte then high byte.
14h	2 writes	Start Normal mode DMA for 8-bit DAC transfer. Data is transfer count - 1, least byte first. Stereo DAC transfer if stereo flag is set in mixer register 0Eh. Maximum sample rate is 44 kHz mono, 22 kHz stereo.
15h	2 writes	Start Normal mode DMA for 16-bit DAC transfer. Data is transfer count - 1, least byte first. Stereo DAC transfer if stereo flag is set in mixer register 0EH. Maximum sample rate is 22 kHz mono, 11 kHz stereo.
1Ch		Start Auto-Initialize mode DMA for 8-bit DAC transfer. Block size must be previously set by command 48h. Stereo DAC transfer if stereo flag is set in mixer register 0Eh. Maximum sample rate is 44 kHz mono, 22 kHz stereo.
1Dh		Start Auto-Initialize mode DMA for 16-bit DAC transfer. Block size must be previously set by command 48h. Stereo DAC transfer if stereo flag is set in mixer register 0Eh. Maximum sample rate is 22 kHz mono, 11 kHz stereo.
20h	1 read	Direct mode 8-bit ADC. Data is 8-bit unsigned. Firmware controlled input volume for AGC.
21h	2 reads	Direct mode 16-bit ADC, returns least byte first. Data is 16-bit unsigned format. Input volume controlled by command DDh.
24h	2 writes	Start Normal mode DMA for 8-bit ADC transfer. Data is transfer count - 1, least byte first. Firmware controlled input volume for AGC. Maximum sample rate is 22 kHz: use command 99h for higher rates up to 44 kHz.
25h	2 writes	Start Normal mode DMA for 16-bit ADC transfer. Data is transfer count - 1, least byte first. Input volume controlled via command DDh. Maximum sample rate is 22 kHz.
2Ch		Start Auto-Initialize mode DMA for 8 bit ADC transfer. Block size must be previously set by command 48h. Firmware controlled input volume for AGC. Maximum sample rate is 22 kHz: use command 98h for higher rates up to 44 kHz.
2Dh		Start Auto-Initialize mode DMA for 16-bit ADC transfer. Block size must be previously set by command 48h. Input volume is controlled by command DDh. Maximum sample rate is 22 kHz.
30h/31h		MIDI input mode. Detects MIDI serial input data and transfers to data register, setting Data-Available flag in register SBBase+Eh. Command 31h will also generate an interrupt request for each byte received. Exit MIDI input mode by executing a write to port SBBase+Ch. The data written is ignored. A software reset will also exit this mode.
34h/35h		MIDI UART mode. Acts like commands 30h/31h, except that any data written to SBBase+Ch will be transmitted as MIDI serial output data. The only way to exit this mode is through software reset.
38h	1 write	MIDI output. Transmit one byte.
40h	1 write	Set time constant, X, for timer used for DMA mode DAC/ADC transfers: rate = 1 MHz / (256-X) X must be less than or equal to 233. For stereo DAC, program sample rate for twice the per-channel rate.
41h	1 write	Alternate set time constant, X: rate = 1.5 MHz / (256-X) This command provides more accurate timing for certain rates such as 22,050. X must be less than equal to 222. For stereo DAC, program sample rate for twice the per-channel rate.
42h	1 write	Set filter clock independently of timer rate. (note that the filter clock is automatically set by commands 40h/41h) Filter clock rate: rate = 7.16E6 / (256-X) The relationship between the low-pass filter -3 dB point and the filter clock rate is approximately 1:82.
48h	2 writes	Set block size-1 for high speed mode and auto-init mode transfer, least byte first.
74h	2 writes	Start ADPCM 4-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
75h	2 writes	Same as command 74h, except with reference byte flag.

Table 33 Command Summary (Continued)

Command	Data Byte(s) Write/Read	Function
76h	2 writes	Start ADPCM 2.6-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
77h	2 writes	Same as command 76h, except with reference byte flag.
7Ah	2 writes	Start ADPCM 2-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
7Bh	2 writes	Same as command 7Ah, except with reference byte flag.
80h	2 writes	Generate silence period. Data is number of samples - 1.
90h		Start Auto-Initialize DMA 8-bit transfer to DAC. Transfer count must be previously set by command 48h.
91h		Start DMA 8-bit transfer to DAC. Transfer count must be previously set by command 48h.
98h		Start High-Speed mode, Auto-Initialize, DMA 8-bit transfer from ADC. Transfer count must be previously set by command 48h. There is no AGC. Input volume is controlled with command DDh. Maximum sample rate is 44 kHz.
99h		Start High-Speed mode, DMA 8-bit transfer from ADC. Transfer count must be previously set by command 48h. There is no AGC. Input volume is controlled with command DDh. Maximum sample rate is 44 kHz.
Axh, Bxh, Cxh		(where x = 0 to Fh) ES1946 extension commands. Many of these commands are used to access the ES1946's controller registers. For information on these registers, see the register descriptions.
C0h	1 write,1 read	Read controller registers A0h to BFh. Write this command followed by the register number, then read register contents from SBBase+Ah.
C1h		Resume after suspend.
C6h		Enable ES1946 Extension commands Axh, Bxh. Must be issued after every reset.
C7h		Disable ES1946 Extension commands Axh, Bxh.
D0h		Pause DMA. Internal FIFO operations continue until the FIFO is empty (DAC transfer) or full (ADC transfer). It is not necessary to use this command to stop DMA if the transfer is completed normally and the end-of-DMA interrupt is generated.
D1h		Enable Audio 1 DAC input to mixer.
D3h		Disable Audio 1 DAC input to mixer.
D4h		Continue DMA after command D0h.
D8h	1 read	Return Audio 1 DAC enable status: 0=disabled; FFh=enabled
DCh	1 read	Return current input gain, 0-15, (valid during 16-bit ADC and 8-bit "high speed mode" ADC).
DDh	1 write	Write current input gain, 0-15, (valid during 16-bit ADC and 8-bit "high speed mode" ADC).
E1h	2 reads	Return version number high (3), followed by version number low (1). This indicates legacy DOS compatibility.

POWER MANAGEMENT

The ES1946 is a high-performance device with low power consumption. In addition to the low-power deep sub-micron CMOS mixed-signal technology used to process the ES1946, various features are designed into the device to provide benefits from popular power-saving techniques.

CLKRUN Protocol

The PCI CLKRUN feature is one of the primary methods of power management on the PCI bus interface of the ES1946 for the notebook computer. The protocol is defined in the *PCI Mobile Design Guide*, published by the PCI Special Interest Group (PCISIG). To use this feature, a PCI sideband signal, CLKRUN, must be supported by the chipset. All PCI bus signals must be in leakage control state to shut down leak current at I/O buffers, as specified in the *PCI Mobile Design Guide*.

PCI Power Management Interface (PPMI)

The *PCI Bus Power Management Interface Specification* (PPMI), also published by the PCISIG, establishes the infrastructure required to let the operating system control the power of PCI functions. This is done by defining standard PCI interface and operations to manage the power of PCI functions on the bus. PCI functions can be assigned one of five power management states, and the PCI bus itself can be assigned one of four power management states.

The ES1946 (as a PCI function) supports the following five power-management states:

- D0 – full power
- D1 – embedded controller is halted
- D2 – D1 and analog functions are off
- $D3_{hot}$ – D2 and oscillator is off
- $D3_{cold}$ – power supply is off

To minimize the power consumed by the ES1946 in $D3_{hot}$ state, the PCI bus should be in B2 or B3 state, where the PCI clock is stopped.

All PPMI registers are located in the PCI configuration space. This allows the operating system to identify the power management capabilities of the ES1946 without a special driver.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Ratings	Symbol	Value	Units
Analog supply voltage	VDDA	-0.3 to 7.0	V
Digital supply voltage	VDDD	-0.3 to 7.0	V
Input voltage	VIN	-0.3 to 7.0	V
Operating temperature range	TA	0 to 70	Deg C
Storage temperature range	TSTG	-50 to 125	Deg C

WARNING: Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. Operation beyond the “Operating Conditions” is not recommended and extended exposure beyond the “Operating Conditions” may affect device reliability.

Thermal Characteristics

The ES1946 is designed to operate at 3.3 volt digital supply voltage and 5 volt analog voltage and at temperatures between 0°C and + 70°C.

The ES1946 digital and analog characteristics operate under the following conditions:

VDDD	3.0 V to 3.6 V
VDDA[a]	4.75 V to 5.25 V
TA	25 °C

a. VDDA must always be greater than VDDD - 0.3 V.

Operating Current (Typical)

State	Digital	Analog
D0	27 mA	52 mA
D1	11 mA	52 mA
D2	10 mA	10 µA
D3 (Bus B3 PCLK stopped)	15 µA	10 µA

Table 34 Digital Characteristics

Parameter	Symbol	Min	Max	Unit	Conditions
Input high voltage. All inputs except: MSD, MCLK, SW(A-D), VOLUP, VOLDN, XI	VIH1	VDDD x 0.5	VDDD + 0.5	V	VDDD = max
Input high voltage: MSD, MCLK	VIH2	2.0	VDDD + 0.5	V	VDDD = max
Input high voltage: SW(A-D), VOLUP, VOLDN	VIH3	VDDD - 0.2	VDDD + 0.5	V	VDDD = max
Input high voltage: XI	VIH4	VDDD x 0.7	VDDD + 0.5	V	VDDD = max
Input low voltage. All inputs except: MSD, MCLK, SW(A-D), VOLUP, VOLDN	VIL1	-0.5	VDDD x 0.3	V	VDDD = min
Input low voltage: MSD, MCLK	VIL2	-0.5	0.5	V	VDDD = min
Input low voltage: SW(A-D), VOLUP, VOLDN	VIL3	-0.5	0.2	V	VDDD = min
Input low voltage:XI	VIL4	-0.5	VDDD X 0.2	V	VDDD = min
Output high voltage. All output except: XO, T (A - D)	VOH1	VDDD x 0.9		V	VDDD = max IOH = -.5 mA
Output high voltage: XO, T (A - D)	VOH2	VDDD x 0.7			VDDD = max IOH = -1mA
Output low voltage. All output except: XO, T (A - D)	VOL1		VDDD x 0.1	V	VDDD = min IOL = 1.5 mA
Output low voltage: XO, T(A - D)	VOL2		VDDD x 0.2		VDDD = min IOL = 2 mA

Table 35 Analog Characteristics

Parameter	Pins	Min	Typ	Max	Unit
Reference voltage	CMR		2.25		V
Input Impedance	LINE_L, LINE_R, AUXB_L, AUXB_R, MIC, MONO_IN		125		kΩ
	AUXA_L, AUXA_R, CIN_L, CIN_R		50		kΩ
Output impedance	FOUT_L, FOUT_R	3.5	5	6.5	kΩ
	AOUT_L, AOUT_R, MONO_OUT max load for full-scale output	10			kΩ
Input voltage	MIC – preamp ON			150	mVp-p
	– preamp OFF			2.8	Vp-p
	LINE_L, LINE_R, AUXA_L, AUXA_R, AUXB_L, AUXB_R, MONO_IN			3.4	Vp-p
Output voltage	AOUT_L, AOUT_R, MONO_OUT full-scale output range	3.4			Vp-p
Mic preamp gain	MIC		20		dB
Input Impedance	AUXA L,R				

TIMING DIAGRAMS

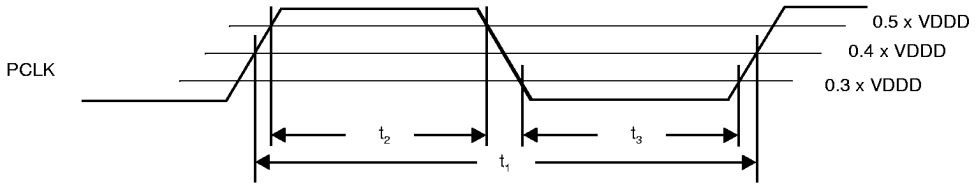


Figure 15 PCLK Timing

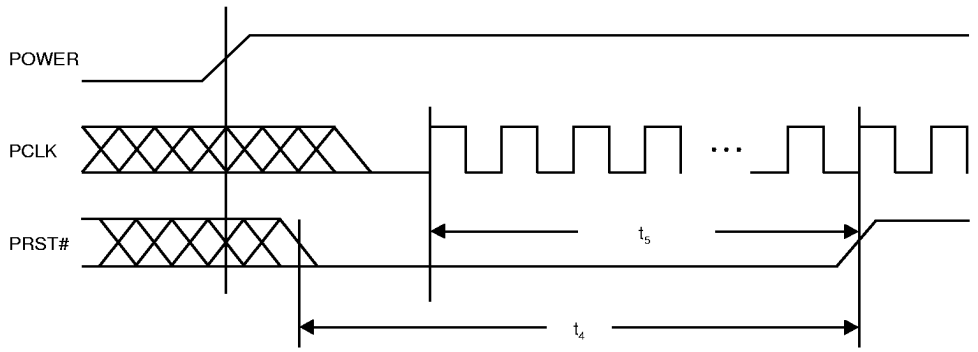


Figure 16 PRST Timing

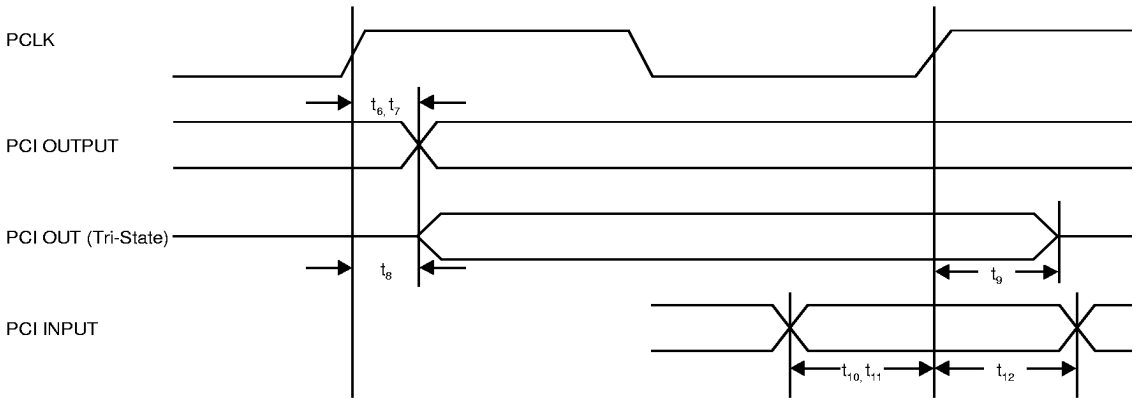


Figure 17 PCI Signal Timing[1]

[1] PCI signals include: AD[31:0], CBE[3:0]B, PAR, FRAMEB, IRDYB, TRDYB, STOPB, IDSEL, DEVSELB, REQB, GNTB, LOCKB, CLKRUNB, PCPCIREQB, and PCPCIGNTB

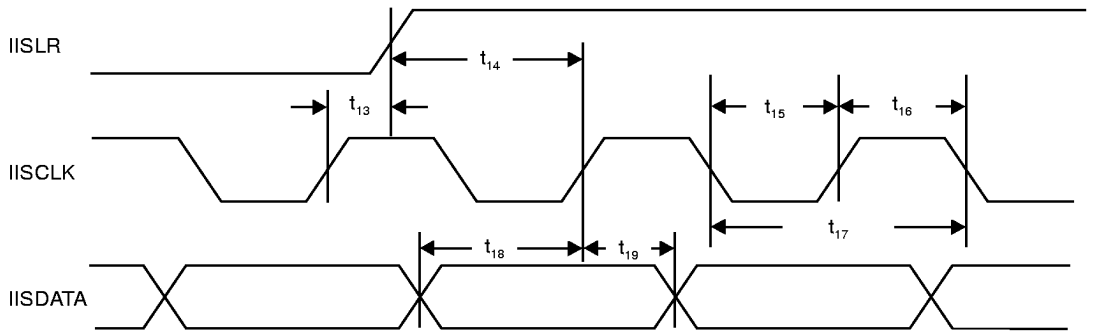


Figure 18 Serial Input Timing for I²S Interface

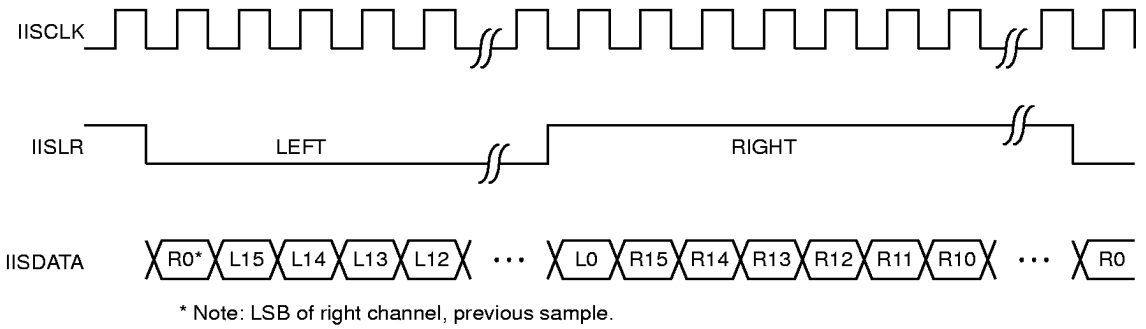


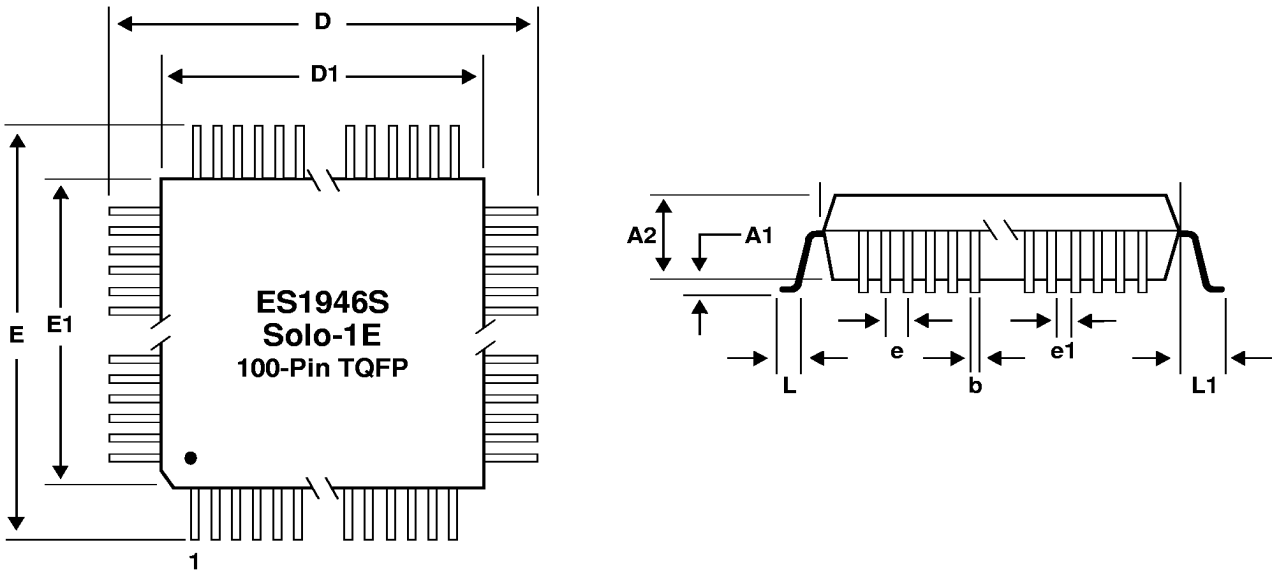
Figure 19 I²S Digital Input Format

TIMING CHARACTERISTICS

Table 36 Timing Characteristics

Symbol	Parameter	Min	Max	Units
t_1	PCLK cycle time	30	∞	ns
t_2	PCLK high time	11		ns
t_3	PCLK low time	11		ns
–	PCLK slew rate	1	4	V/ns
t_4	PRST# active time after power stable	1		ms
t_5	PRST# active time after PCLK stable	1000		t_1
t_6	PCLK to signal valid delay (except REQB)	2	11	ns (0 pF load)
t_7	PCLK to signal valid delay (REQB)	2	12	ns
t_8	Float to active delay	2		ns
t_9	Active to float delay		28	ns
t_{10}	Input setup time to PCLK (except GNTB)	7		ns
t_{11}	Input setup time to PCLK (GNTB)	12		ns
t_{12}	Input hold time from PCLK	0		ns
t_{13}	IISLR input hold time	2		ns
t_{14}	IISLR input setup time	32		ns
t_{15}	IISCLK low time	22		ns
t_{16}	IISCLK high time	22		ns
t_{17}	ISCLK cycle time	108		ns
t_{18}	IISDATA input setup time	32		ns
t_{19}	IISDATA input hold time	2		ns

MECHANICAL DIMENSIONS



Symbol	Description	Millimeters		
		Min	Nom	Max
D	Lead to lead, X-axis	15.75	16.00	16.25
D1	Package's outside, X-axis	13.90	14.00	14.10
E	Lead to lead, Y-axis	15.75	16.00	16.25
E1	Package's outside, Y-axis	13.90	14.00	14.10
A1	Board standoff	0.05	0.10	0.15
A2	Package thickness	1.35	1.40	1.45
b	Lead width	0.17	0.22	0.27
e	Lead pitch	-	0.50 BSC	-
e1	Lead gap	0.24	-	-
L	Foot length	0.45	0.60	0.75
L1	Lead length	0.93	1.00	1.07
-	Foot angle	0°		7°
-	Coplanarity	-	-	0.102
-	Leads in X-axis	-	25	-
-	Leads in Y-axis	-	25	-
-	Total leads	-	100	-
-	Package type	-	TQFP	-

Figure 20 ES1946 Mechanical Dimensions

APPENDIX A: ES689/ES69x DIGITAL SERIAL INTERFACE

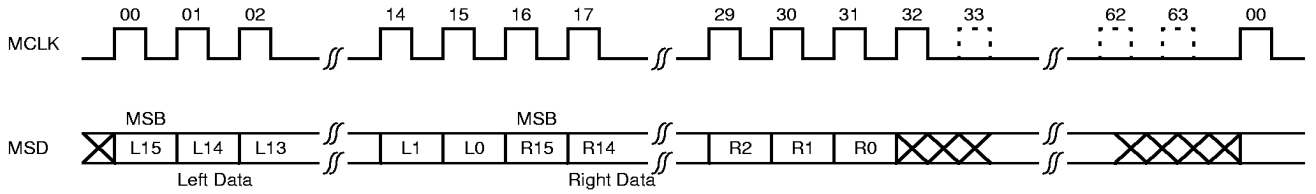
In order for the ES689/ES69x to acquire the FM DAC, bit 4 of mixer register 48h inside theES1946 must be set high. When bit 4 is set high, activity on the MCLK signal causes the ES1946 to connect the FM DAC to the ES689/ES69x. If MCLK stays low for more than a few sample periods, the ES1946 reconnects the FM DAC to the FM synthesizer.

After reset, the ES689/ES69x transmits samples continuously. In this mode, bit 4 of mixer register 48h must be set/cleared to assign the current owner of the FM DAC.

The ES689/ES69x can be programmed to enter Activity-Detect mode using system exclusive command 4.

For more information on system exclusive commands, see the appropriate ES689/ES69x Data Sheet. In this mode, the ES689/ES69x blocks the serial port output (i.e., sets MSD and MCLK low) if no MIDI input is detected on the MSI pin for a period of 5 seconds. It resumes output of data on the serial port as soon as a MIDI input is detected on the MSI pin. This is the recommended mode of operation.

The I²S interface and the ES689/ES69x interface share the same pins; therefore, these two interfaces cannot co-exist on the same design.



Bit Clock Rate (MCLK):	2.75 MHz
Sample Rate:	42,968.75 Hz
MCLK Clocks per Sample:	33 clocks (+ 31 missing clocks)
MSD Format:	16 bits, unsigned (offset 8000h), MSB first

MSD changes after rising edge of MCLK. Hold time relative to MCLK rising edge is 0-25 nanoseconds.

APPENDIX B: I²S ZV INTERFACE REFERENCE

(Excerpted from *PCMCIA Document Number 0135 – Release 010 1/15/96*)

Overview

The following diagram shows the system level concept of the ZV Port. The diagram demonstrates how TV in a window can be achieved in a portable computer with a low cost PC Card. An MPEG or teleconferencing card can also be plugged into the PC Card slot.

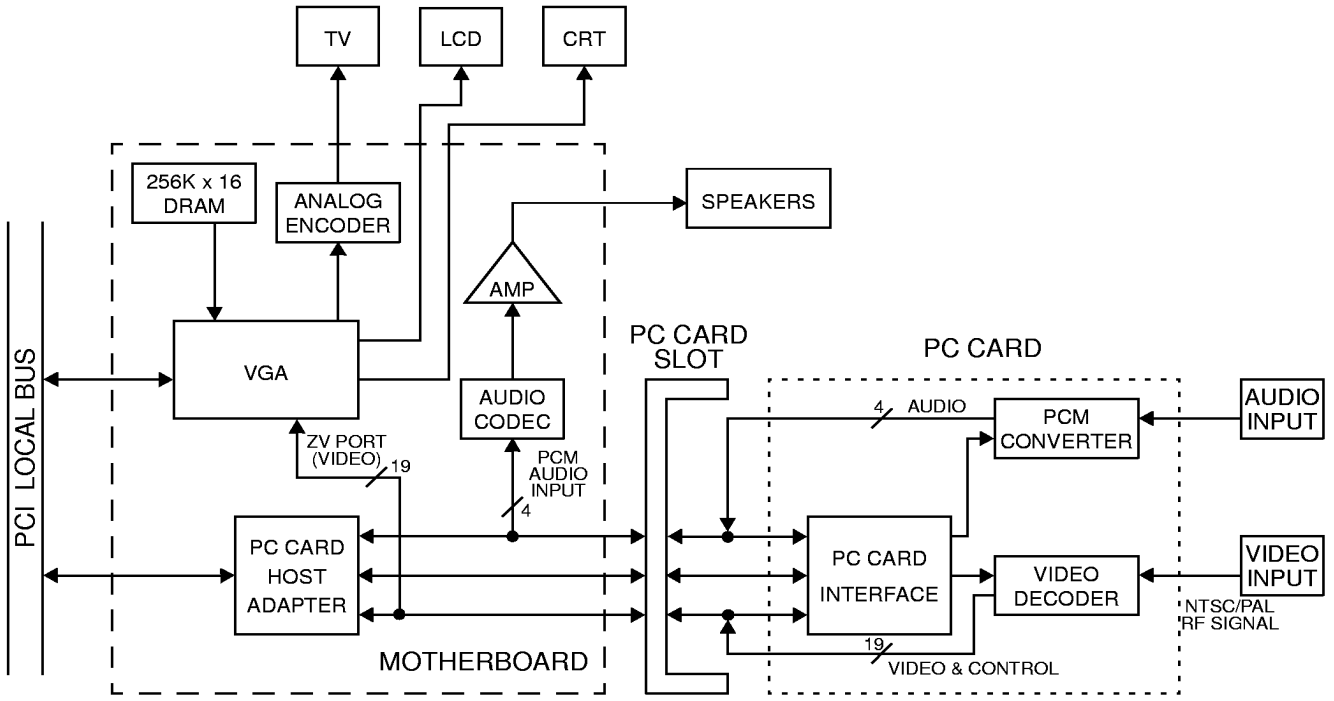


Figure 21 Example ZV Port Implementation

The Audio Interface

The ZV Port compliant PC Card sends audio data to the host computer using Pulse Code Modulation (PCM). Audio data is transferred using serial I²S format. The audio circuitry in the host system is primarily a PCM DAC.

The PCM audio DAC is a complete stereo digital-to-analog system including digital-interpolation, delta-sigma digital-to-analog conversion, digital de-emphasis and analog filtering. Only the normal power supply decoupling components and one resistor and capacitor per channel for analog signal reconstruction are required.

The DAC accepts data at standard audio frequencies including 48 kHz, 44.1 kHz, 32 kHz, and 22 kHz. Audio data is input via the serial data input pin, SDATA. The Left/Right Clock (LRCLK) defines the channel and delineation of data. The Serial Clock (SCLK) clocks the audio data into the input data buffer. The Master Clock (MCLK) is used to operate the digital interpolation filter and the delta-sigma modulator.

NOTE: MCLK is not required in some I²S designs.

Table 37 Common Clock Frequencies

LRCLK (KHz)	MCLK(MHz)	
	256x	384x
22	5.632	8.448
32	8.192	12.2880
44.1	11.2896	16.9344
48	12.2880	18.4320

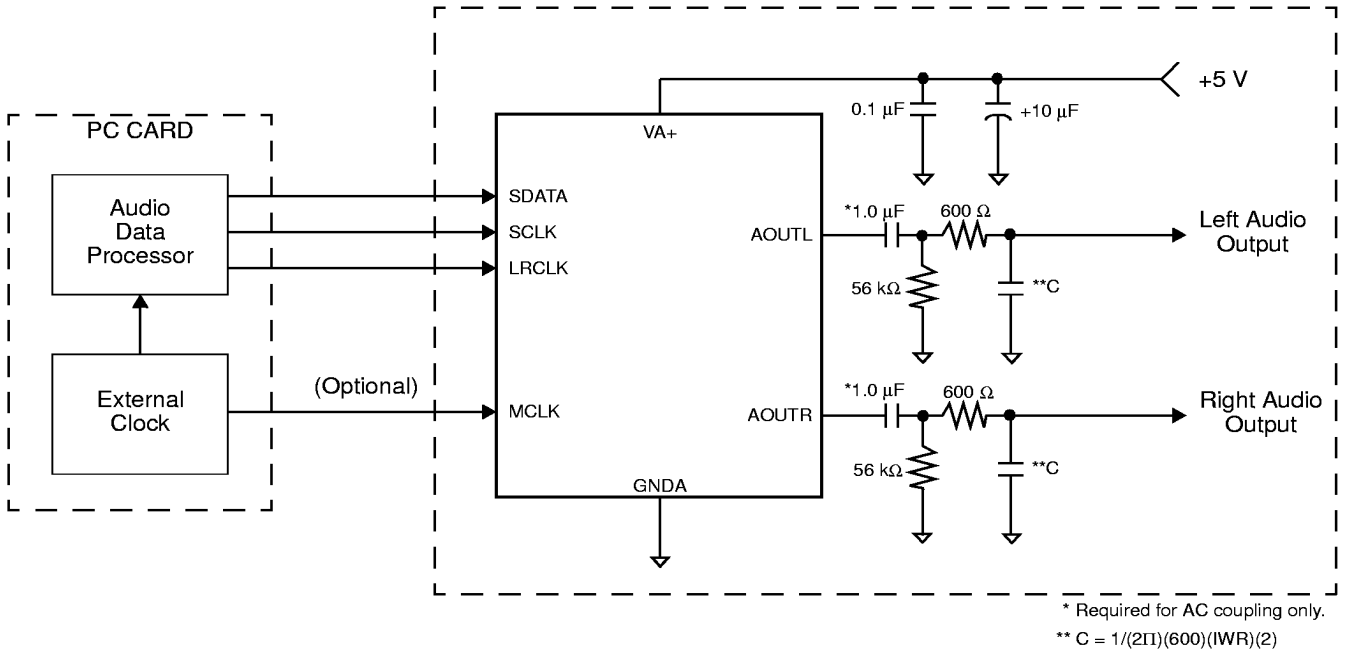


Figure 22 Typical ZV Port Audio Implementation

Audio Interface Timing

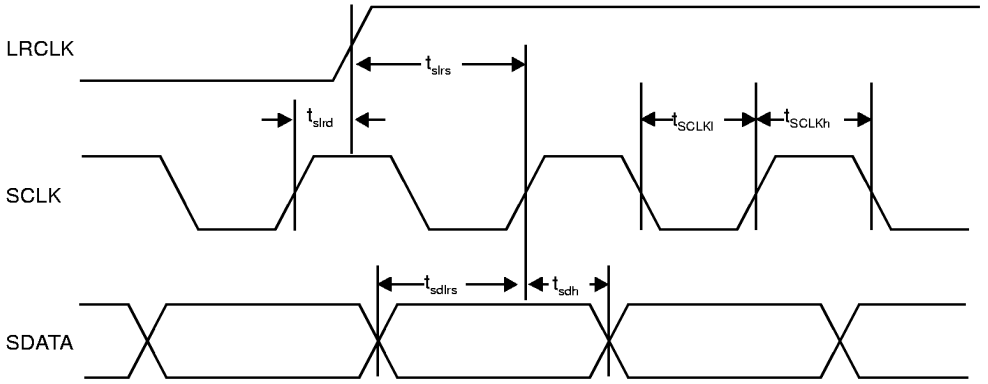


Figure 23 Audio Interface Timing

Table 38 AC Parameters for Audio Signals

Symbol	Parameter	Min
t_{slrd}	LRCLK delay	2 ns
t_{slrs}	LRCLK setup	32 ns
t_{SCLKl}	bit clock low	22 ns
t_{SCLKh}	bit clock high	22 ns
t_{sdlrs}	data setup	32 ns
t_{sdh}	data hold	2 ns

LRCLK

This signal determines which audio channel (left/right) is currently being input on the audio Serial Data input line. LRCLK is low to indicate the left channel and high to indicate the right channel. Typical frequency values for this signal are 48 kHz, 44.1 kHz, 32 kHz, and 22 kHz.

SCLK

This signal is the serial digital audio PCM clock.

SDATA

This signal is the digital PCM signal that carries audio information. Digital audio data is transferred using I²S format.

MCLK

This signal is the Master clock for the digital audio. MCLK is asynchronous to LRCLK, SDATA, and SCLK.

The MCLK must be either 256x or 384x the desired Input Word Rate (IWR). IWR is the frequency at which words for each channel are input to the DAC and is equal to the LRCLK frequency. The following table illustrates several standard audio word rates and the required MCLK and LRCLK frequencies. Typically, most devices operate with 384 Fs master clock.

The ZV Port audio DAC should support a MCLK frequency of 384 Fs. This results in the frequencies shown below.

LRCLK (kHz) Sample Frequency	SCLK (MHz) 32 x Fs	MCLK (MHz) 384x Fs
22	0.704	8.448
32	1.0240	12.2880
44.1	1.4112	16.9344
48	1.5360	18.4320

I²S Format

The I²S format is shown in Figure 24 below. The digital audio data is left-channel MSB-justified to the high-to-low going edge of the LRCLK plus one SCLK delay.

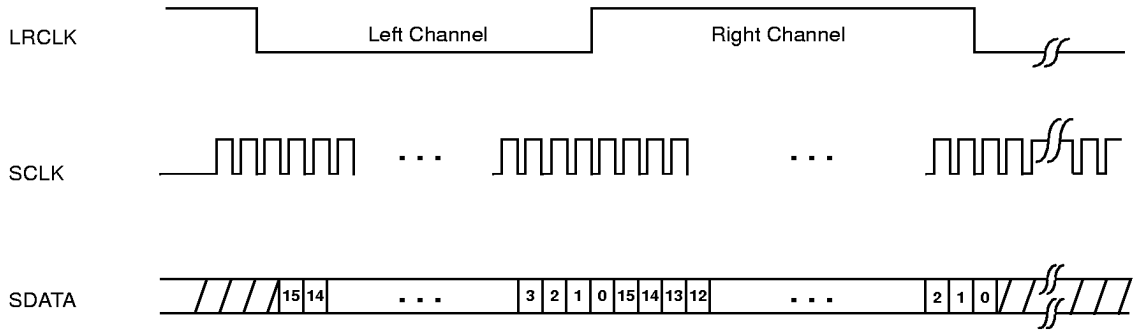


Figure 24 I²S Digital Input Format

ZV Port Pin Assignments

Table 39 shows the function of various PC Card signals when the ZV Port custom interface mode is set in the PC Card Host Adapter. PC Card signals not mentioned in the table below remain unchanged from the 16-bit PC Card I/O and Memory interface.

Table 39 ZV Port Interface Pin Assignments

PC Card Pin Number	I/O and Memory Interface Signal Name	I/O and Memory I/O[a]	ZV Port Interface Signal Name	ZV Port I/O[a]	Comments
8	A10	I	HREF	O	Horizontal Sync to ZV Port
10	A11	I	VSYNC	O	Vertical Sync to ZV Port
11	A9	I	Y0	O	Video Data to ZV Port YUV:4:2:2 format
12	A8	I	Y2	O	Video Data to ZV Port YUV:4:2:2 format
13	A13	I	Y4	O	Video Data to ZV Port YUV:4:2:2 format
14	A14	I	Y6	O	Video Data to ZV Port YUV:4:2:2 format
19	A16	I	UV2	O	Video Data to ZV Port YUV:4:2:2 format
20	A15	I	UV4	O	Video Data to ZV Port YUV:4:2:2 format
21	A12	I	UV6	O	Video Data to ZV Port YUV:4:2:2 format
22	A7	I	SCLK	O	Audio SCLK PCM Signal
23	A6	I	MCLK	O	Audio MCLK PCM Signal
24:25	A[5:4]	I	RESERVED	RFU	Put in three state by Host Adapter No connection in PC Card
26:29	A[3:0]	I	ADDRESS[3:0]	I	Used for accessing PC Card
33	IOIS16#	O	PCLK	O	Pixel Clock to ZV Port
46	A17	I	Y1	O	Video Data to ZV Port YUV:4:2:2 format
47	A18	I	Y3	O	Video Data to ZV Port YUV:4:2:2 format
48	A19	I	Y5	O	Video Data to ZV Port YUV:4:2:2 format
49	A20	I	Y7	O	Video Data to ZV Port YUV:4:2:2 format
50	A21	I	UV0	O	Video Data to ZV Port YUV:4:2:2 format
53	A22	I	UV1	O	Video Data to ZV Port YUV:4:2:2 format
54	A23	I	UV3	O	Video Data to ZV Port YUV:4:2:2 format
55	A24	I	UV5	O	Video Data to ZV Port YUV:4:2:2 format
56	A25	I	UV7	O	Video Data to ZV Port YUV:4:2:2 format
60	INPACK#	O	LRCLK	O	Audio LRCLK PCM Signal
62	SPKR#	O	SDATA	O	Audio PCM Data Signal

a. “I” indicates signal is input to PC Card, “O” indicates signal is output from PC Card

APPENDIX C: SCHEMATICS

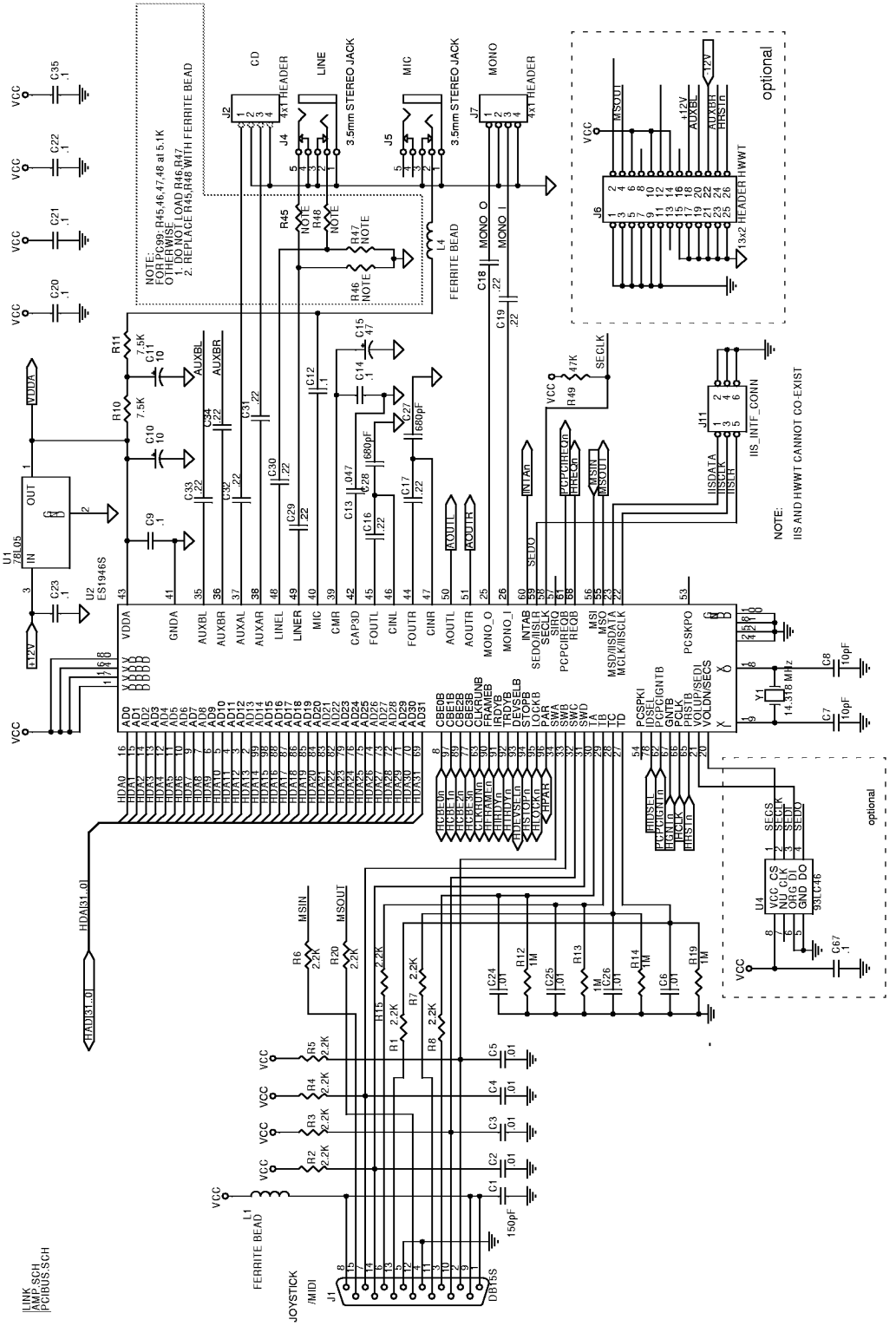


Figure 25 ES1946 Schematic

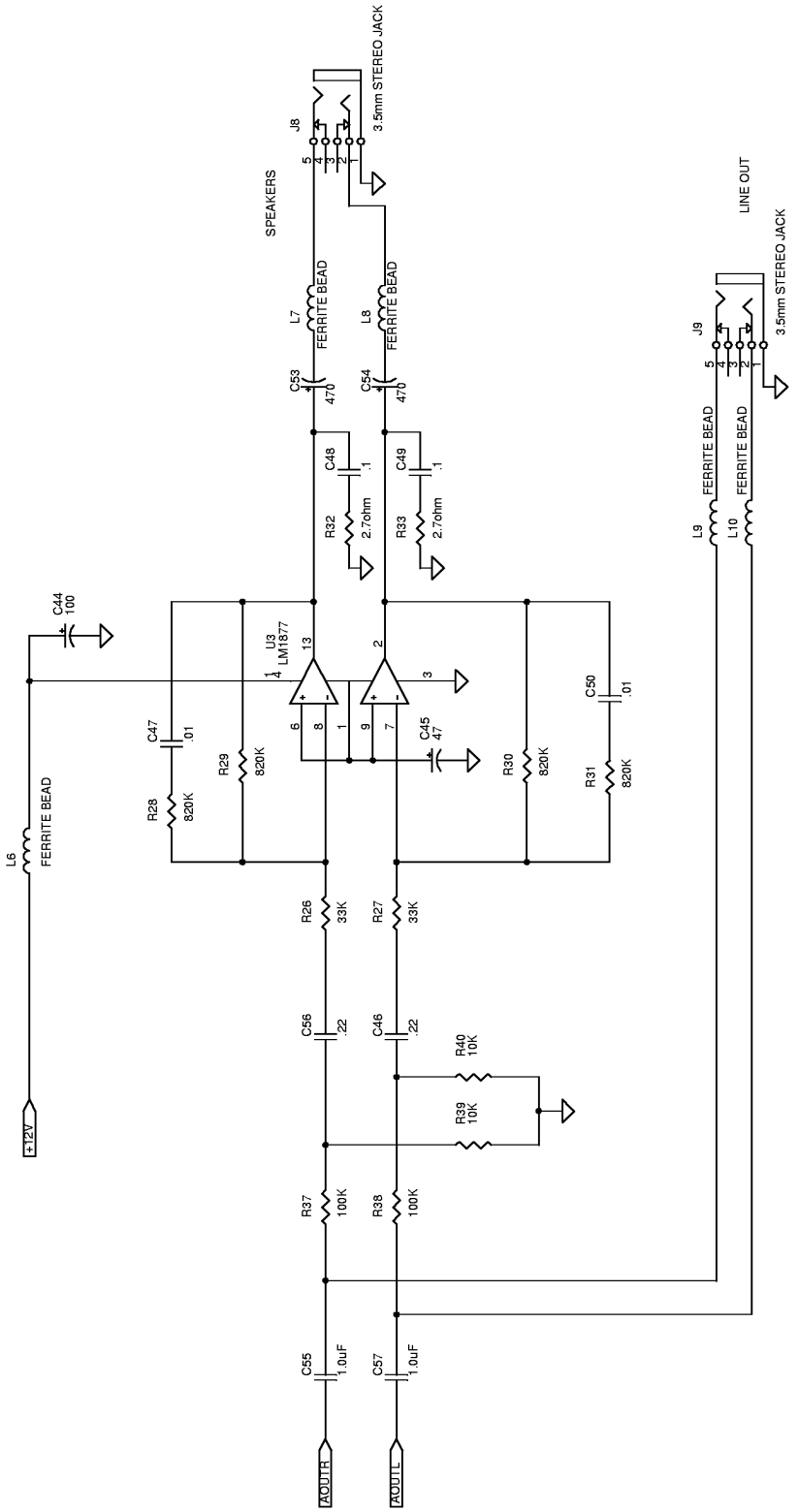


Figure 26 Amplifier

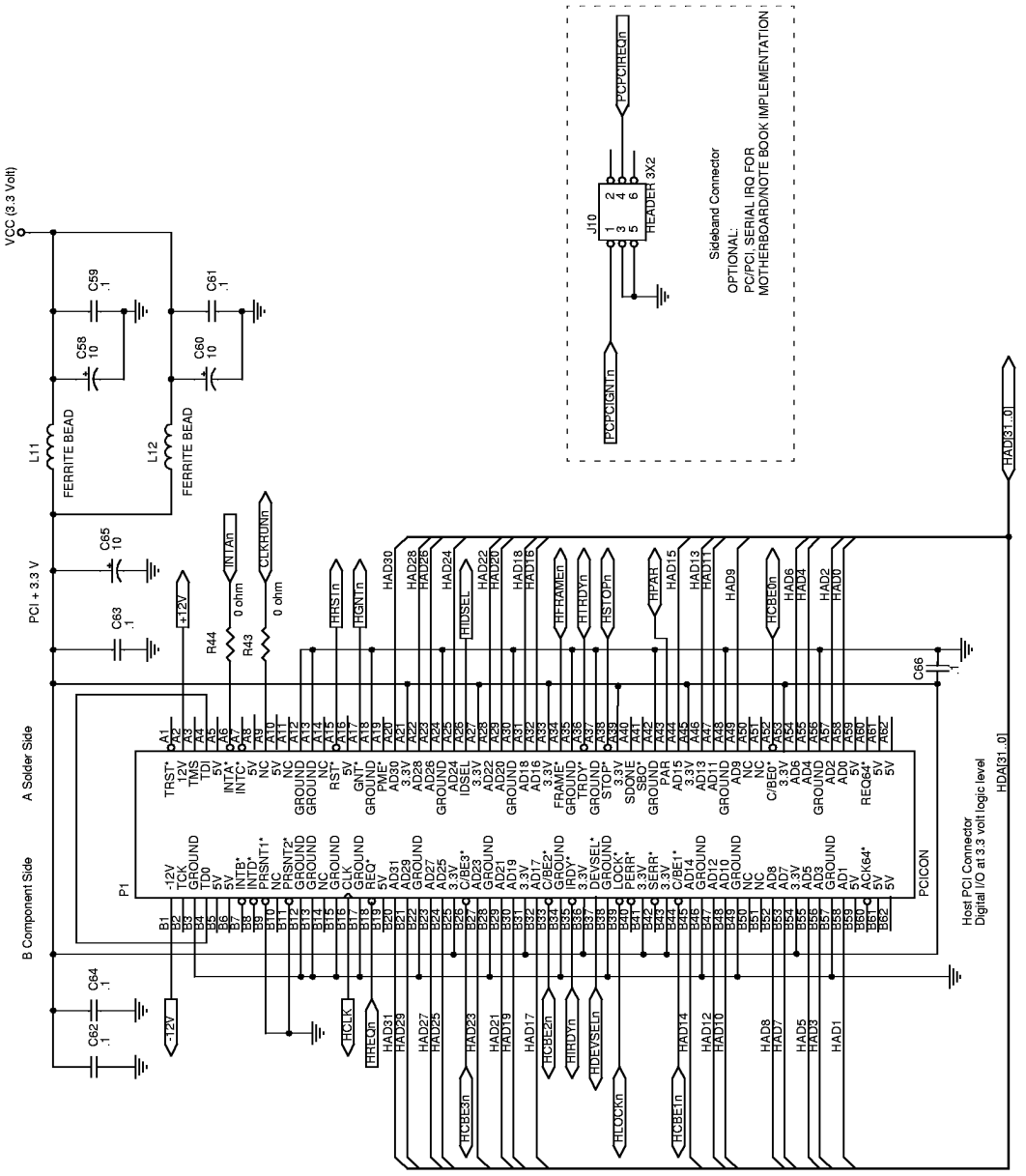


Figure 27 PC Interface

APPENDIX D: BILL OF MATERIALS

Table 40 ES1946 Bill of Materials (BOM)

Item	Quantity	Reference	Part
1	1	C1	150 pF
2	10	C2,C3,C4,C5,C6,C24,C25,C26,C47,C50	.01 µF
3	2	C7,C8	10 pF
4	17	C9,C12,C14,C20,C21,C22,C23,C35,C48,C49,C59,C61,C62, C63,C64,C66,C67	.1 µF
5	5	C10,C11,C58,C60,C65	10 µF
6	1	C13	.047 µF
7	2	C15,C45	47 µF
8	12	C16,C17,C18,C19,C29,C30,C31,C32,C33,C34,C46,C56	.22 µF
9	2	C27,C28	680 pF
10	1	C44	100 µF
11	2	C53,C54	470 µF
12	2	C55,C57	1.0 µF
13	1	J1	DB15S (game port)
14	1	J2	4x1 HEADER (CD-audio)
15	4	J4,J5,J8,J9	3.5 mm STEREO JACK (Line, Mic, Spkr, Line-Out)
16	1	J6	13x2 HEADER (optional HWWT)
17	1	J7	4x1 HEADER (Mono-In / Mono-Out)
18	2	J10, J11	3x2 HEADER (I2S, sideband connector)
19	11	L1,L4,L6,L7,L8,L9,L10,L11,L12,R45,R48	FERRITE BEAD[a]
20	10	R1,R2,R3,R4,R5,R6,R7,R8,R15,R20	2.2K
21	2	R10,R11	7.5K
22	4	R12,R13,R14,R19	1M
23	2	R26,R27	33K
24	4	R28,R29,R30,R31	820K
25	2	R32,R33	2.7ohm
26	2	R37,R38	100K
27	2	R39,R40	10K
28	2	R43, R44	0 ohm
29	2	R46,R47	Note[b]
30	1	R49	47K
31	1	U1	78L05
32	1	U2	ES1946S
33	1	U3	LM1877
34	1	U4	93LC46
35	1	Y1	14.318 MHz

a. For PC99 system design, R45 and R48 are 5.1 Kohm resistors. Otherwise use ferrite bead.
b. For PC99 system design, R46 and R47 are 5.1 Kohm resistors. Otherwise do not load.

APPENDIX E: LAYOUT GUIDELINES

PCB Layout

Notebook, Motherboard, Pen-based, and PDA portable computers have the following similarity in PCB layout design:

1. Multi-layer (usually 4 to 8 layer).
2. Double-sided SMT.
3. CPU, corelogic (chipset), system memory, VGA controller, and video memory reside in the same PCB.

This is a very noisy environment for adding an audio circuit. The following are the guidelines for PCB layout for an ESS *Audio***Drive®** chip application.

Component Placement

The audio circuit-related components, including the audio I/O jack and connector, must be grouped in the same area.

There are two possible placements for these audio components:

A grouped on one side of the PCB.

B separated on both sides of the PCB.

In Case B, audio component grouping will take less space.

Analog Ground Plane

Audio circuits require two layers of analog ground planes for use as shielding for all analog traces.

In component placement case A (Figure 28), the first layer of analog ground plane is on the analog component side, the second analog ground plane is on the inner layer, and the analog traces are embedded between these two planes.

In component placement case B (Figure 29), the analog ground planes are on both sides of the PCB, and the analog traces are shielded in the middle.

Case A:

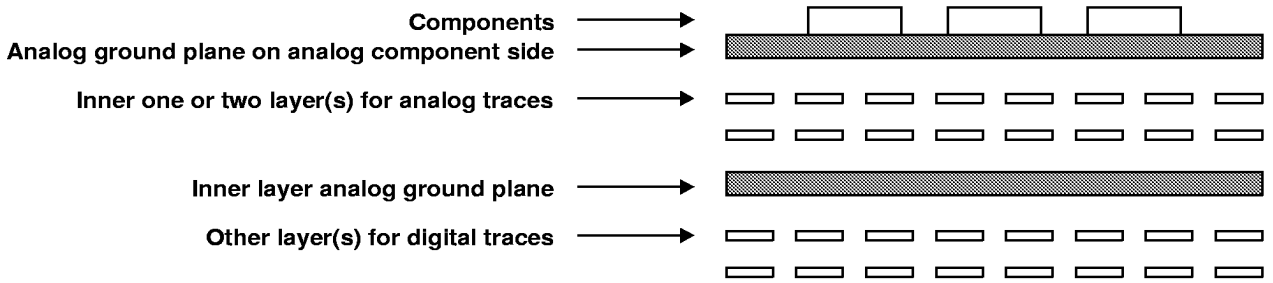


Figure 28 Analog Components on One Side of the PCB

Case B:

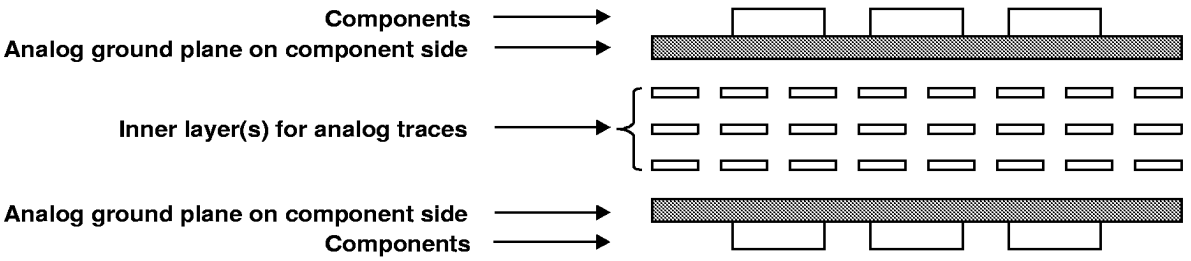


Figure 29 Analog Components on Both Sides of the PCB

Special Notes

The analog traces should be placed as short as possible.

The MIC-IN circuit is the most sensitive of the audio circuits, and requires proper and complete shielding.

APPENDIX F: SID/SVID SAMPLE CODE

```
/*
 * Set SID/SVID
 *
 * ---------------------------------------------------------------------
 * Use 32 bits compiler
 *  Use PCI configuration mechanism Type-1 of  PCI Rev2.1 Spec.
 * ---------------------------------------------------------------------
 * Copyright (C) 1998,  ESS Technology, Inc.
 *
 */
typedef unsigned short ushort;
typedef unsigned long ulong;

#define PCI_EN 0x80000000
#define ESS_VENDORID 0x125d
#define SOLO1_DEVID 0x1969

/* SID & SVID you want to set */
#define MY_SID 0x1766
#define MY_SVID 0x1480

/* global var for debug */
static int debug = 1;

/*---------------------------------------------------------------------*/
static void OUTPDW (ushort port, ulong val)
{
  __asm {
    mov dx,port;
    mov eax,val;
    out dx,eax;
  }
}
static ulong INPDW (ushort port)
{
  ulong retval;
  __asm {
    mov dx,port;
    in eax,dx;
    mov retval,eax;
  }
  return (retval);
}

static void dwPciConfigWrite (ulong busdevfunc, ulong ofs, ulong val)
{
  ulong cmd = PCI_EN | busdevfunc | (ofs & 255);
  OUTPDW (0xcf8, cmd);
  OUTPDW (0xcfc, val);
}
static ulong dwPciConfigRead (ulong busdevfunc, ulong ofs)
{
  ulong cmd = PCI_EN | busdevfunc | (ofs & 255);
  OUTPDW (0xcf8, cmd);
  return INPDW (0xcfc);
}
ulong FindDevice( ushort vid, ushort did )
{
```

```
  ulong busdevfunc = ~0;
  ulong pcibus, pcidev;
  union {
    ulong device_vendor;
    struct {
      ushort vendor;
      ushort device;
    } dv;
  } dv_id;

  /* For BIOS implementation, the search for Solo is optional because bus/dev are known. */
  for (pcibus=0; pcibus<0x10; pcibus++) {
    for (pcidev=0; pcidev<0x20; pcidev++) {
      dv_id.device_vendor =
            dwPciConfigRead((pcibus<<16)|(pcidev<<11), 0x0);
      if (dv_id.dv.vendor == 0xffff)
            continue;
      if (debug) {
            printf ("device_vendor=%04x device_id=%04x bus=%d slot=%d\n",
                  dv_id.dv.vendor, dv_id.dv.device, pcibus, pcidev>>3);
      }
      if (dv_id.dv.vendor == vid && dv_id.dv.device == did) {
            busdevfunc = (pcibus<<16)|(pcidev<<11);
      }
    }
  }

  if (debug && busdevfunc==~0) {
    printf ("Cannot find Solo\n");
  }
  return busdevfunc;
}

int main (int argc, char **argv)
{
  ulong SoloBusDevFunc = ~0;
  ulong dwSIDSVID;

  SoloBusDevFunc = FindDevice( ESS_VENDORID, SOLO1_DEVID );

  /* check if Solo-1 exist */
  if (!(~SoloBusDevFunc)) return 1;

/* Enable programmable SID/SVID */
  dwPciConfigWrite (SoloBusDevFunc,0x50,
    dwPciConfigRead (SoloBusDevFunc, 0x50) | 0x1001 );

  /* Write to SID/SVID */
  dwPciConfigWrite (SoloBusDevFunc,0x2c,
    MY_SVID | (MY_SID<<16));

  /* Write to shadow-SID/SVID */
  dwPciConfigWrite (SoloBusDevFunc,0x6c,
    MY_SVID | (MY_SID<<16));

  /* Make SID/SVID read-only */
  dwPciConfigWrite (SoloBusDevFunc,0x50,
    dwPciConfigRead (SoloBusDevFunc, 0x50) & ~1 );
```

```
  /* Read back SID/SVID */
  if (debug) {
    dwSIDSVID = dwPciConfigRead (SoloBusDevFunc, 0x2c);
        printf ("SID_SVID = %08lx\n", dwSIDSVID);
  }

  return 0;
}
```

ORDERING INFORMATION

Part Number	Package
ES1946S	100-pin TQFP