DESIGN GUIDE
ES1888 AudioDrive®

ESS Technology, Inc.

Revision C - October 1995

ESS Technology, Inc. • 48401 Fremont Blvd • Fremont CA 94538 • (510) 492-1088 • FAX: (510) 492-1098

Contents

ES1888 AudioDrive® Design Guide

1.0 Introduction

ESS Technology has developed the ES1888 AudioDrive®, a single chip solution for adding 16-bit stereo audio and four-operator FM music synthesis to personal computers. It has integrated all the major blocks of audio into a single chip that can be designed into a motherboard.

1.01 ES1888 Versus ES1688 Overview

The ES1888 is a feature upgrade to the ES1688 design. In addition to the ES1688 features, the ES1888 adds: full duplex operation using two DMA channels, dual joystick timers, hardware volume control, and 6 bit (64 steps) master volume control. The ES1888 is not pin compatible to the ES1688.

The master volume control has been upgraded for 6 bit, 64 steps. To remain backward compatible, there are now three register sets that control the master volume:

1) 6-bits of resolution are available using mixer extension registers 60H and 62H.
2) 4-bits of resolution are still available using mixer extension register 32H for compatibility with the ES688 and ES1688. However, the steps have been adjusted to allow for more attenuation at lower volume levels.
3) 3-bits of resolution are still available using mixer register 22H for game compatibility.

1.02 The ES1888 Features

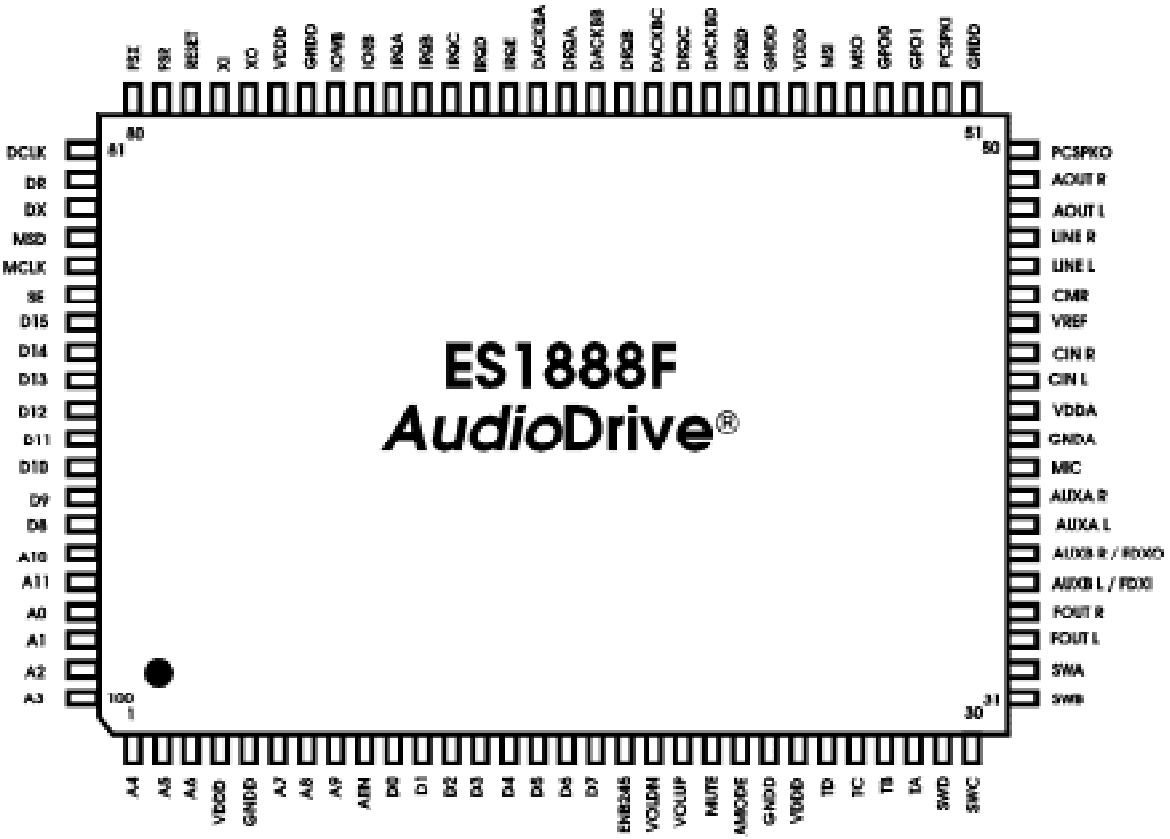
The ES1888 AudioDrive can record, compress, and playback voice, sound and music with built-in mixer controls. It consists of an embedded microprocessor, a game compatible 16-bit stereo A/D and D/A, music 16-bit stereo D/A, system 16-bit stereo D/A, 20-voice 72 operator FM music synthesizer, MIDI serial port compatible with MPU401 UART mode, dual joystick timers, hardware volume control, DMA control, and ISA bus interface logic. A DSP serial interface allows an external DSP to take over analog resources such as the D/A or A/D converters. Control of I/O address, DMA, and interrupt selection is controled by system software. Interface to analog inputs is extremely simple. There are stereo inputs for CD-audio, line-in, and an external music synthesis chip or TV, and a mono microphone input to an internal pre-amp. A digital PC speaker input is converted to an analog signal with volume control and is available as an analog output signal. Advanced power management features such as Suspend/Resume to disk, self timed power down, auto-wakeup, and partial power-down are supported.

The ES1888 AudioDrive has a 6 bit master volume control with 64 total steps.

The ES1888 AudioDrive is compatible with Sound Blaster PRO™ version 3.01 voice and music functions as documented in the Sound Blaster Series Developer Kit.

This design guide should be used as a reference document only. The designs shown in this document are examples only and are not meant to be either the most current version of a design or production worthy design. Contact your ESS sales representative for the latest information on reference designs.

1.1 Pinout


ES1888F
AudioDrive®
80
81
51
50
100
1
30
31
PS2
PS0
RESET
XI
XO
VDDD
GNDD
IOW#
IOR#
IRQA
IRQB
IRQC
IRQD
IRQE
DACKSA
DRQA
DACKSB
DRQB
DACKSC
DRQC
DACKSD
DRQD
GNDD
VDDD
MSI
MSO
GPO0
GPO1
PCSPKI
GNDD
DCLK
DR
DX
MSD
MCLK
SE
D15
D14
D13
D12
D11
D10
D9
D8
A10
A11
A0
A1
A2
A3
PCSPKO
AOUT R
AOUT L
LINE R
LINE L
CMR
VREF
CIN R
CIN L
VDDA
GNDA
MIC
AUXA R
AUXA L
AUXB R / RDXO
AUXB L / RDXI
FOUT R
FOUT L
SWA
SWB
A4
A5
A6
VDDD
GNDD
A7
A8
A9
AEN
D0
D1
D2
D3
D4
D5
D6
D7
ENESS
VOLDN
VOLUP
MUTE
AMODE
GNDD
VDDD
TP
TC
TB
EA
SWD
SWC

1.2 ES1888 Pin Description

1.2.1 Digital Pins

Name	Number	I/O	Description
VDDD	4,24,57,75	I	Digital Supply Voltage (3.0V to 5.5V)
GNDD	5,23,51,58 74	I	Digital Ground
VOLDN	19	I	Active low volume decrease button input.
VOLUP	20	I	Active low volume increase button input.
MUTE	21	I	Active low mute toggle button input.
GPO0	54	O	Output that is set low by external reset and thereafter controlled by bit 0 of port 2x7H. Available to system software for power management or other applications.
GPO1	53	O	Output that is set high by external reset and thereafter controlled by bit 1 of port 2x7H. Available to system software for power management or other applications.
MSI	56	I	MIDI serial input. Schmitt trigger input with internal pull-up resistor. Either MPU401 or SoundBlaster formats.
MSO	55	O	MIDI serial data output.
RESET	78	I	Active high reset from ISA bus.
XO	76	O	Crystal oscillator output.
XI	77	I	Crystal oscillator input.
IORB	72	I	Active low read strobe from ISA bus.
IOWB	73	I	Active low write strobe from ISA bus.
A0-A9	1,2,3,6,7,8, 97,98,99,100	I	Address inputs from ISA bus.
A10-A11	95,96	I	Address inputs from ISA bus. The ES1888 requires these pins to be low for all address decodes
AEN	9	I	Active low address enable from ISA bus.
D0-D7	10-17	I/O	Bi-directional data bus. These pins have weak pullup devices to prevent these inputs from floating when not driven.
D8-D15	87-94	I	High byte input data bus. This is used for the system D/A when the 16 bit DMA transfer mode is selected.

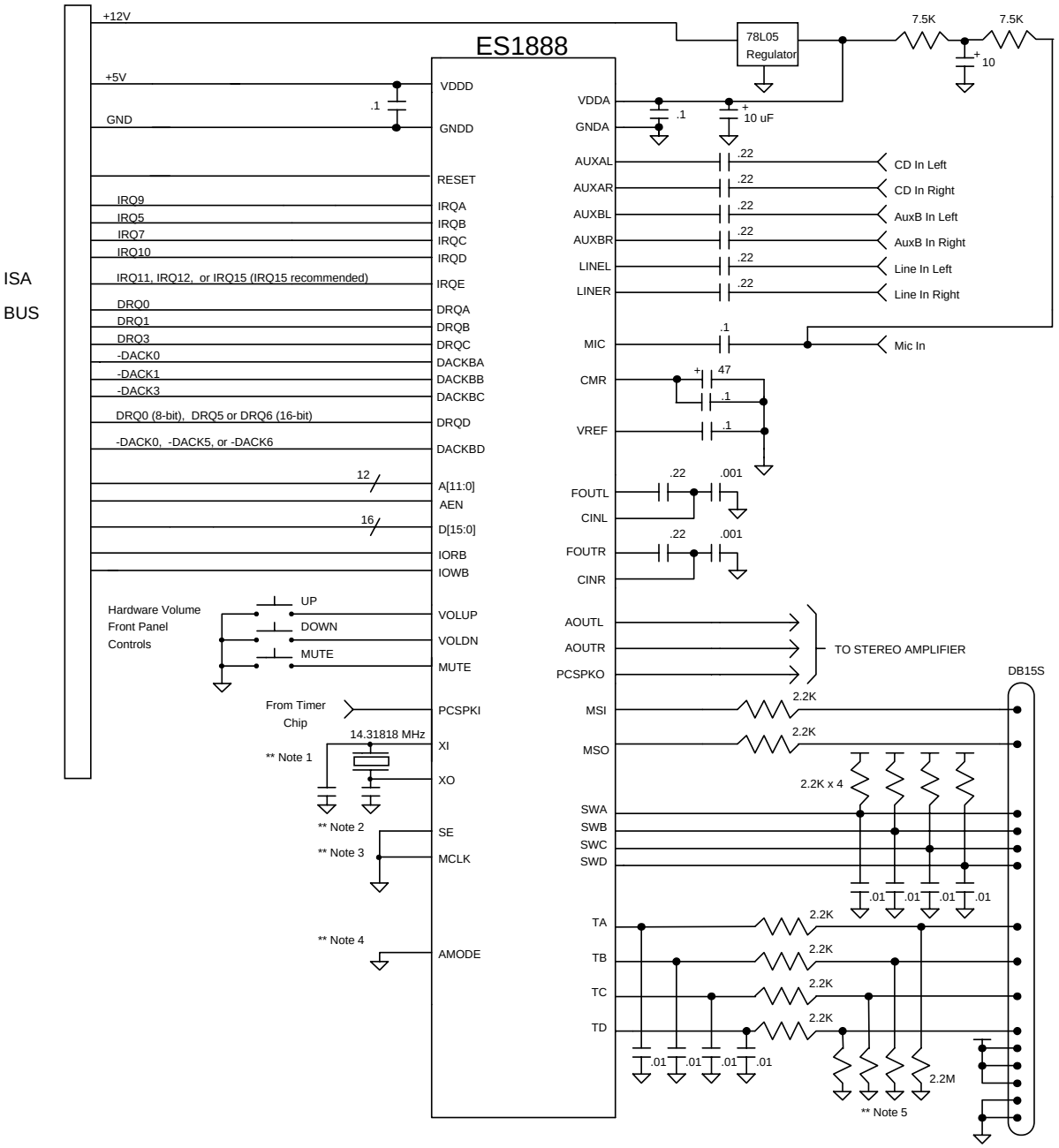
Name	Number	I/O	Description
ENB245	18	O	Active low output when ES1888 is being read or written to. Intended to be connected to the enable control of an external 74LS245.
SWA, B, C, D	29-32	I	Active low joystick switch setting inputs. These SW pins have a 2K pull-up resistor which is pulled to ground by the two switches on each joystick. The joystick port is at address 201.
TA, B, C, D	25-28	I/O	Joystick timer pins. These pins connects to the XY positioning variable resistors for the two joysticks.
AMODE	22	I	Input pin with internal pull-down device. The ES1888 is disabled following a hardware reset and must be configured by one of two methods (optioned by AMODE) of software address selection: 0: Read-Sequence-Key method 1: System-Control-Register method
IRQA,B,C,D	68-71	O	Active high interrupt request to ISA bus. Unselected IRQ outputs are high impedance. IRQs are software configurable.
IRQE	67	O	Active high interrupt request to ISA bus. Reserved for MPU401 use or hardware volume control.
DRQA,B,C	65,63,61	O	Active high DMA request to ISA bus. Unselected DRQ outputs are high impedance. When DMA is not active, the selected DRQ output has a pulldown device that holds the DRQ line inactive unless another device that shares the same DRQ line can source enough current to make the DRQ line active. DRQs are software configurable.
DACKBA, B,C	66, 64, 62	I	Active low DMA acknowledge inputs from ISA bus.
DRQD	59	O	DMA request for the system DAC. This can be selected to be either 8 bit or 16 bit DMA transfers.
DACKBD	60	I	Active low DMA acknowledge for the system DAC.
PCSPKI	52	I	Normally low digital PC speaker signal input. This signal is converted to an analog signal with volume control and appears on analog output PCSPKO.

Name	Number	I/O	Description
FSR	79	I	Input with internal pull-down. Frame Sync for Receive data from external DSP. Programmable for active high or active low.
FSX	80	I	Input with internal pull-down. Frame Sync for Transmit request from external DSP. Programmable for active high or active low.
DCLK	81	I	Input with internal pull-down. Serial data clock from external DSP. Typically 2.048 MHz.
DR	82	I	Input with internal pull-down. Data Receive pin from external DSP.
DX	83	O	Tri-state output. Data Transmit to external DSP. High impedance when not transmitting.
MSD	84	I	Input with internal pull-down. Music Serial Data from external ES689 Music Synthesizer.
MCLK	85	I	Input with internal pull-down. Music Serial Clock from external ES689 Music Synthesizer.
SE	86	I	Input with internal pull-down. Active high to enable serial mode, i.e., enables an external DSP to control analog resources of the ES1888 through the DSP serial interface.

1.2.2 Analog Pins

Name	Number	I/O	Description
VDDA	41	I	Analog supply voltage (4.5V to 5.5V). Should be greater than or equal to VDDD-0.3V.
GNDA	40	I	Analog Ground
MIC	39	I	Microphone input. MIC has an internal pullup resistor to CMR.
LINE L, R	46,47	I	Line input left,right. LINE L,R have internal pullup resistors to CMR.
AUXA L,R	37,38	I	Auxilliary input left, right. AUXA L,R have internal pullup resistors to CMR. Normally intended for connection to an internal or external CD-ROM analog output.
AUXB L,R	35,36	I	Auxilliary input left, right. AUXB L,R have internal pullup resistors to CMR. Normally intended for connection to an external music synthesizer or other line level source. These pins have dual function with the FDXI, FDXO pins (see below).
FDXO	36	O	Normally connected to CMR via an internal resistor. Can be programmed to connect internal to FOUT R pin during DSP serial mode. This pin has dual function with the AUXB R input pin.
FDXI	35	I	Input with internal pullup to CMR. Alternate input to left channel filter stage in DSP serial mode. This pin has dual function with the AUXB L input pin.
FOUT L,R	33,34	O	Filter outputs left, right. A.C. coupled externally to CIN L,R in order to remove DC offsets. These outputs have internal series resistors of about 5K ohms. Capacitors to analog ground on these pins can be used to create a lowpass filter pole that removes switching noise introduced by the switched-capacitor filters.
CIN L,R	42,43	I	Capacitive coupled inputs left, right. These inputs have internal pullup resistors to CMR of approximately 50K ohms.
VREF	44	O	Reference generator resistor divider output. Should be bypassed to analog ground with 0.1 uf capacitor.
CMR	45	O	Buffered reference output. Should be bypassed to analog ground with a 47 uf electrolytic capacitor with a .1 uf capacitor in parallel.
AOUT L,R	48,49	O	Line level stereo outputs, left, right.
PCSPKO	50	O	Analog output of PCSPKI with volume control.

1.3 Typical Application



**Note 1 : Use a crystal for the ES1888 if the accuracy of the ISA bus OSC signal is not sufficient. The XI pin can be driven by an external clock if the clock has CMOS logic levels.

**Note 2 : In designs where the DSP interface is not used, it is recommended to tie SE pin low.

**Note 3: In designs where the ES689 serial interface is not used, it is recommended to tie MCLK pin low.

**Note 4: AMODE selects one of the two software address configuration methods. In this example, the Read-Sequence-Key method is selected.

Figure 1.3.1

1.3.2 Typical Application Description

In this application, the DMA and Interrupt channel configuration is controlled by system software (for example, they are selected by an application called in AUTOEXEC.BAT. All IRQ and DRQ outputs after external reset are disabled and must be configured by software.

The 14.32 MHz crystal oscillator is connected to XI and XO.

The 24 mA drivers on D[7:0] can directly drive the ISA bus without a buffer. D[15:8] are input only.

The PC speaker digital signal is brought in on a header pin and connected to the ES1888.

The game port address 0x201 is associated with the timer pins TA, TB, TC, TD, and the joystick buttons SWA, SWB, SWC, and SWD. The MIDI serial input and output also come from the game port connector in most applications.

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

Analog inputs MIC, LINE L/R, and AUXA L/R are capacitively coupled to their respective inputs. All have pullup resistors to CMR. In this application AUXB L/R is not used and is left floating.

The outputs of the filters, FOUT L/R, are A.C. coupled to the inputs CIN L/R, which provides for D.C. blocking, and an opportunity for low pass filtering with capacitors to analog ground at these inputs.

ES1888 analog outputs AOUT L and AOUT R are intended to be A.C. coupled to an amplifier, volume control potentiometer, or line level outputs. They are external mixed with analog output PCSPKO.

2.0 ES1888 Programmer's Guide

2.1 Two Modes of Operation: Compatibility Mode and Extended Mode

As shown in figure 2.1.1, there are two ways to access the A/D and D/A converters inside the ES1888. The first mode is called Compatibility Mode, where the ES1888 is compatible with both the Sound Blaster™ and the Sound Blaster PRO™. This is the default mode after any reset. In this mode the ES1888 processor is an intermediary in all functions between the ISA bus and the A/D and D/A. The blocks labeled "FIFO/DMA Control" and "256-Byte FIFO" are inactive. The ES1888 processor will perform limited FIFO functions using 64 bytes of internal memory.

The ES1888 also supports another mode of operation called Extended Mode. In this case a 256-byte FIFO is used as an intermediary between the ISA bus and the A/D and D/A control registers. The ES1888 processor is mostly idle in this mode. DMA control is handled by dedicated logic. New commands have been added to access the various control registers needed for Extended Mode operation. Some of these commands are also useful for compatibility mode, such as those that configure DMA and IRQ channels. Table 2.1.1 compares features of the two modes of operation.

In both modes, a set of mixer control registers allows application software to control the analog mixer, record source, and output volume. Programming the ES1888 Enhanced Mixer is described in chapter 3.

Programming in Extended Mode is described in chapter 4.
Programming for Power Management is described in chapter 5.
The MIDI Interface is described in chapter 6.
Sofware Address Configuration is described in chapter 7.
The DSP Interface is described in chapter 8.

Figure 2.1.1. Block Diagram of the ES1888

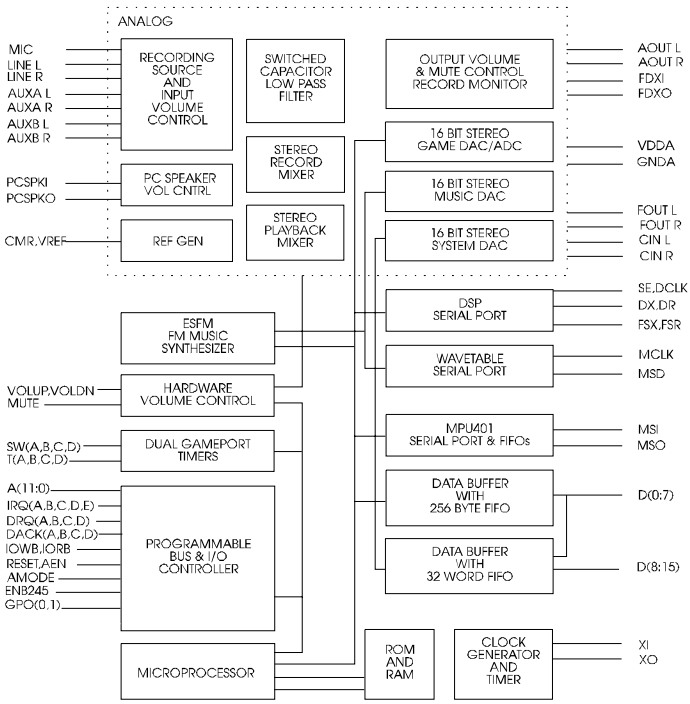


Table 2.1.1 Comparision of Operation Modes

	Compatibility Mode (Sound Blaster Pro)	**Extended Mode**
Sound Blaster PRO™ Compatible	Yes	No
FIFO Size	64 Bytes (firmware managed)	256 Bytes (hardware managed)
Mono 8-bit A/D, D/A	Yes, to 44 KHz	Yes, to 44 KHz
Mono 16-bit A/D, D/A	Yes, to 22 KHz	Yes, to 44 KHz
Stereo 8-bit D/A	Yes, to 22 KHz	Yes, to 44 KHz
Stereo 8-bit A/D	Yes, to 22 KHz	Yes, to 44 KHz
Stereo 16-bit D/A	Yes, to 11 KHz	Yes, to 44 KHz
Stereo 16-bit A/D	No	Yes, to 44 KHz
Signed/Unsigned Control	No	Yes
Automatic Gain Control during recording	Firmware controlled, to 22 KHz, mono only.	No
Programmed I/O Block Transfer for A/D and D/A	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
A/D and D/A jitter	+/- 2 usec	None

2.2 Summary of Port Addresses and Functions

Port	Read/Write	Function
0E0,0E1,0F9,0FB		Used by Software Address Configuration, if enabled
201	Read/Write	Decoded along with IORB and IOWB to access internal joystick port.
2x0-2x3	Read/Write	20-voice FM Synthesizer. Address and data registers.
2x4	Read/Write	Mixer address register
2x5	Read/Write	Mixer data register.
2x6	Write	Bit 0: 1: hold ES1888 in reset 0: release ES1888 from reset Bit 1: 1: hold ES1888 FIFO in reset 0: release ES1888 FIFO from reset Note: Bit 1 is "don't care" for compatibility mode.
	Read	Activity flags and other status for power management.
2x7	Read/Write	Power Management Register.
2x8-2x9	Read/Write	11-voice FM Synthesizer. Address and data registers.
2xA	Read Only	Input data from read buffer. Poll bit 7 of port 2xE to test whether the read buffer contents are valid.
2xC	Write	Write data to write buffer. Sets write buffer not available flag until data is processed by the ES1888.
	Read	Bit 7: 1: write buffer not available or ES1888 busy. 0: write buffer is available and ES1888 not busy. Bit 6: same as bit 7 of port 2xE. Bit 5: 1: Extended Mode FIFO Full (256 bytes loaded) Bit 4: 1: Extended Mode FIFO Empty (0 bytes loaded) Bit 3: 1: FIFO Half Empty, Extended Mode Flag. Bit 2: 1: ES1888 processor generated an interrupt request (e.g., from Compatibility Mode DMA complete) Bit 1: 1: Interrupt request generated by FIFO Half Empty flag change. Used by progammed I/O interface to FIFO in Extended Mode. Bit 0: 1: Interrupt request generated by DMA Counter Overflow in Extended Mode.
2xE	Read Only	Bit 7: 1: data available in read buffer. 0: data not available in read buffer. This flag is reset by a read from port 2xA. A read from port 2xE will reset any IRQ request.

2xF	Read/Write	Address for I/O access to FIFO in Extended Mode.
3x0-3x1	Read/Write	MPU-401 port (x=0,1,2 or 3) if enabled.
388-38B	Read/Write	Same as 2x0-2x3, i.e. 20-voice FM Synthesizer

2.3 Reset

There are two sources of reset: Hardware Reset and Software Reset. The hardware reset signal comes from the ISA bus. Software reset is controlled by bit 0 of port 2x6H.

To reset the ES1888 by software:

1) write a 1 to port 2x6H.
2) delay a short period, for example, by reading back 2x6H.
3) write a 0 to port 2x6H.
4) in a loop that lasts at least 1 msec, poll port 2xEH bit 7=1 for read data available. If bit 7=1, then read the byte from port 2xAH. Exit loop if the content is 0AAH. Otherwise, continue polling.

.
Both Hardware and Software Reset will:

- Disable Extended Mode
- Reset the timer divider and filter registers for 8 KHz sampling
- Stop any DMA in progress
- Clear any active interrupt request
- Disable voice input of mixer (see D1H/ D3H commands)
- Reset Compatibility Mode and Extended Mode DMA Counters to 2048 bytes
- Set analog direction to be D/A, with the DAC value set to mid-level
- Set input volume for 8-bit recording with AGC to maximum
- Set input volume for 16-bit recording to mid-range

In addition to the above list, a Hardware Reset will reset all mixer registers to default values.

2.4 Command / Data Handshaking Protocol with the ES1888 Processor

Bit 7 of port 2xCH is the ES1888 BUSY flag. It is set when the write buffer is full or when the ES1888 is otherwise busy (for example, during initialization after reset or during Compatibility Mode DMA requests). To write a command or data byte to the ES1888 processor, poll this bit until it is clear, then write the command/data byte to port 2xCH.

Note: the port 2xCH Write Buffer is shared with Compatibility Mode DMA write operations. When DMA is active, the BUSY flag will be cleared during time windows when a command can be received. Normally, the only commands that should be sent during DMA operations are 0DxH commands: DMA pause/continue, Voice Enable/Disable, etc. In this situation it is recommended that interrupts be disabled between the time that the busy bit is polled and the command is written. Also, the time between these instructions should be minimized. For more information, see the section below entitled "Sending Commands During Compatibility Mode DMA Operations".

The read buffer status flag is polled by reading bit 7 of port 2xEH. When a byte is available it will be set high. Note that any read of port 2xEH will also clear any active interrupt request from the ES1888. An alternative way of polling the read buffer status bit is via bit 6 of port 2xCH which is the same flag. The buffer status flag is cleared automatically by reading the byte from port 2xAH.

2.5 Compatibility Mode D/A Operation

After reset, the analog circuitry is setup for D/A operations. Any A/D command will cause a switch to the A/D "direction", and any subsequent D/A command will switch the ES1888 back to the D/A "direction". The D/A output is filtered and connected to the voice input of the mixer. After reset, the voice input to the mixer is muted. The purpose is to prevent pops. The ES1888 maintains a status flag called the Voice Enable/Disable flag that indicates when the voice channel is muted. Use command D1H to enable the voice channel and command D3H to disable the voice channel.

If you do not wish to reset the ES1888 before playing a new sound, and you are not certain of the status of the analog circuits, you can mute the voice input to the mixer with command D3H, then setup D/A direction and level using the direct-to-DAC command:

10H + 80H

Then waiting 25 msec for the analog circuitry to settle before enabling the voice channel with command D1H.

You may still hear a pop sound 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, you should always finish a D/A transfer with a command to set the DAC level to mid-range:

10H + 80H

2.5.1 8-bit, 16-bit, and Compressed Formats

8-bit samples are unsigned, ranging from 0 to 0FFH, with the D.C. level around 80H.

16-bit samples are unsigned, least byte first, ranging from 0000H to 0FFFFH with the D.C. level around 8000H.

The ES1888 supports two types of compressed sound D/A operations: ESPCM, which uses a variety of proprietary compression techniques developed by ESS Technology, and ADPCM, which is supported by many other sound cards but is of a lower quality.

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

2.5.2 Direct Mode D/A vs. DMA Mode D/A

In direct mode, the timing for D/A 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 (or 11H for 16-bit data) is issued followed by the sample. Polling of the write buffer available flag is required before writing the command and between the command and the data.

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

In DMA mode, the programmable timer in the ES1888 controls the rate at which samples are sent to the DAC. The timer is programmed using command 40H which also sets up the programmable filters inside the ES1888. The ES1888 firmware maintains an internal FIFO (32 levels for 16-bit transfers, 64 levels for 8-bit transfers) that is filled by DMA transfers and emptied by the timed transfers to the DAC.

Before a DMA transfer, the application first programs the DMA controller for the desired transfer size and address, then programs the ES1888 with the same size information. At the end of the transfer the ES1888 will generate 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.

In "normal mode" DMA transfers, the DMA controller must be initialized and the ES1888 be commanded for every block that is transferred. In "auto-initialize mode", the DMA transfer is continuous, in a circular buffer, and the ES1888 generates an interrupt for the transition between buffer halves. In this mode the DMA controller and ES1888 need to be set up only once.

The ES1888 supports mono 8-bit transfers to DAC at a rate up to 44KHz. Mono 16-bit transfers are supported up to a rate of 22KHz.

2.5.3 Stereo D/A Transfers in Compatibility Mode

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

To perform a stereo D/A transfer, first program bit 1 of mixer register 0EH to be high. Then set the timer divider to twice the per-channel sample rate. The maximum stereo transfer rate for 8-bit data is 22 KHz per channel, so for this case program the timer divider as if you were doing 44 KHz mono. The maximum stereo transfer rate for 16-bit data is 11 KHz per channel.

For 8-bit data, the ES1888 expects the first byte transferred to be for the right channel, and subsequent bytes to alternate left, right etc.

For 16-bit data, the ES1888 expects the DMA transfers to be a multiple of 4, with repeating groups in the order :

left low byte
left high byte
right low byte
right high byte

Be sure to clear bit 1 of mixer register 0EH when you are done with the D/A transfer.

2.6 Compatibility Mode A/D Operation

The ES1888 analog circuitry is switched from the D/A direction to the A/D direction by the first direct or DMA mode A/D command. It is a good practice to discard the first 25 to 100 msec of samples if possible because pops might occur in the data due to the change from the D/A to A/D direction. In the A/D direction the voice input to the mixer is automatically muted.

The ES1888 has four recording sources: Microphone, Line, Aux/CD, and Mixer. Microphone input is the source after any reset. Select the source using the mixer control register 0CH/1CH.

The selected source passes through an input volume stage that can be programmed with 16 levels of gain from 0 to +22.5 db in steps of 1.5 db. In 8-bit recordings (other than "high speed mode") the volume stage is controlled by the ES1888 firmware for the purposes of automatic gain control. 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 set the input volume level from 0 to 15. The reset default is mid-range, 8.

The ES1888 supports direct mode A/D, "normal mode" DMA for A/D, as well all "auto-initialize mode" DMA for A/D. The differences between the various types are described above for D/A.

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

The maximum sample rate for direct mode A/D is 22 KHz.

The maximum sample rate for DMA A/D for both 8-bit and 16-bit is 22 KHz, using commands 24H, 25H, 2CH or 2DH.

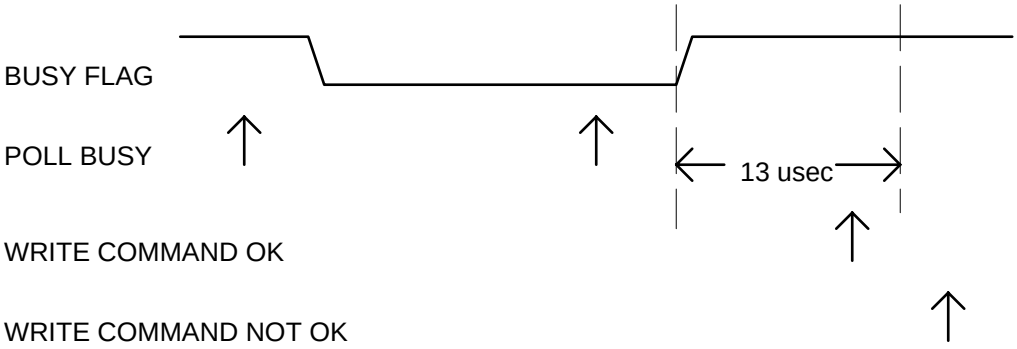
There is a special "high speed mode" for A/D that allows 8-bit sampling up to 44KHz. This mode uses commands 98H ("auto-initialize")and 99H ("normal"). No AGC is performed: the input volume is controlled with command DDH.

2.7 Sending Commands During Compatibility Mode DMA Operations

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

The ES1888 has an internal 64-byte FIFO used for DMA to the D/A converter and from the A/D converter. When the FIFO is full (in the case of D/A, empty in the case of A/D), DMA requests are temporarily suspended and the BUSY flag (bit 7 of port 22CH) is cleared. This allows a window of opportunity to send a command to the ES1888. You should restrict yourself to commands such as D1H and D3H which control the mixer voice enable/disable status, and command D0H, which suspends (i.e., pauses) DMA.

The ES1888 chip will set the busy flag when the command window is no longer open. Application software must send a command within 13 usec after the busy flag goes high or the command will be confused with DMA data. This is normally easy to do if the polling is done with interrupts disabled.



As an example of sending a command during DMA, consider the case where the application desires to send command D0H in the middle of a DMA transfer. The application disables interrupts and polls the busy flag. Because of the FIFO and the rules used for determining the command window, it is possible for the current DMA transfer to complete while waiting for the busy flag to clear. In this event, the D0 command has no function, and *there will be a pending interrupt request* from the DMA completion.

You can clear this interrupt request by reading port 22EH before enabling interrupts or have a way of signaling the interrupt handler that DMA is inactive, so that it does not try to start a new DMA transfer.

2.8 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 write	Direct write 16-bit DAC. Data is 16-bit unsigned format, first low byte then high byte.
14H	2 write	Start normal mode DMA for 8-bit DAC transfer. Data is transfer count - 1, least byte first. Stereo DAC transferi if stereo flag is set in mixer register 0EH. Maximum sample rate is 44 KHz mono, 22 KHz stereo.
15H	2 write	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 read	Direct mode 16-bit ADC, returns least byte first. Data is 16-bit unsigned format. Input volume controlled via command DDH.
24H	2 write	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 write	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.

COMMAND	DATA BYTE(S) WRITE/READ	FUNCTION
2DH		Start auto-initialize mode DMA for 16-bit ADC transfer. Block size must be previously set by command 48H. Input volume is controlled via 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 2xE. Command 31H will also generate an interrupt request for each byte received. Exit MIDI input mode by executing a write to port 2xC. 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 2xC will be transmitted as MIDI serial output data. The only way to exit this mode is a 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 D/A, 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 D/A, 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 lowpass filter -3db point and the filter clock rate is approximately 1:82.

COMMAND	DATA BYTE(S) WRITE/READ	FUNCTION
48H	2 write	Set block size-1 for high speed mode and auto-init mode transfer, least byte first.
64H	2 write	Start ESPCM 4.3-bit (low compression) format DMA transfer to DAC. Data is transfer count - 1, least byte first.
65H	2 write	Same as command 64H, except with reference byte flag.
66H	2 write	Start ESPCM 3.4-bit (medium compression) format DMA transfer to DAC. Data is transfer count - 1, least byte first.
67H	2 write	Same as command 66H, except with reference byte flag.
6AH	2 write	Start ESPCM 2.5-bit (high compression) format DMA transfer to DAC. Data is transfer count - 1, least byte first.
6BH	2 write	Same as command 6AH, except with reference byte flag.
6EH	2 write	Start ESPCM 4.3-bit (low compression) format A/D, compression, and DMAtransfer. Data is transfer count - 1, least byte first.
6FH	2 write	Same as command 6EH, except with reference byte flag.
74H	2 write	Start ADPCM 4-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
75H	2 write	Same as command 74H, except with reference byte flag.
76H	2 write	Start ADPCM 2.6-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
77H	2 write	Same as command 76H, except with reference byte flag.
7AH	2 write	Start ADPCM 2-bit format DMA transfer to DAC. Data is transfer count - 1, least byte first.
7BH	2 write	Same as command 7AH, except with reference byte flag.
80H	2 write	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.

COMMAND	DATA BYTE(S) WRITE/READ	FUNCTION
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		ES1888 extension commands.
C1H		Resume after suspend.
C6H		Enable ES1888 extension commands Ax, Bx. Must be issued after every reset.
C7H		Disable ES1888 extension commands Ax, Bx.
CEH	1 read	Read GPO0/1 Power Management Register
CFH	1 write	Write GPO/1 Power Management Register
D0H		Pause DMA. Internal FIFO operations will 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 voice D/A input to mixer.
D3H		Disable voice D/A input to mixer.
D4H		Continue DMA after command D0H.
D5H	1 read	For compatibility with the ES488, always returns "1".
D6H	1 read	ES488 command not supported by ES1888. Use mixer registers instead.
D7H	1 write	ES488 command not supported by ES1888. Use mixer registers instead.
D8H	1 read	Return voice D/A enable status: 0=disabled FFH=enabled
DCH	1 read	Return current input gain, 0-15, (valid during 16-bit A/D and 8-bit “high speed mode” A/D).

COMMAND	DATA BYTE(S) WRITE/READ	FUNCTION
DDH	1 write	Write current input gain, 0-15, (valid during 16-bit A/D and 8-bit "high speed mode" A/D).
DEH	1 read	ES488 command not supported by ES1888. Use mixer registers instead.
DFH	1 write	ES488 command not supported by ES1888. Use mixer registers instead.
E1H	2 read	Return version number high (3), followed by version number low (1). This indicates Sound Blaster PRO™compatibility.
E7H	2 read	Returns ES1888 Identification bytes: 68H 8xH, where x is the version code. Sound cards other than the ES1888 will either ignore this command or perform an unknown function, in which case the sound card should be reset after this command is used. The version code, x , is less than 8 for the ES688, and greater than or equal to 8 for the ES1888.
F2H		Generate an interrupt for test purposes.
FDH		Forces power down. Software or Hardware reset required for wakeup.

2.9 Additional ES1888 Features and Registers

2.9.1 Hardware Volume Control

There are 3 pins activated for hardware volume control. Each has an internal pullup device:

VOLUP	active low volume increase button input
VOLDN	active low volume decrease button input
MUTE	active low mute button input

Pressing the up or down button will immediately cause one increment or decrement. Holding the button for 400 msec consecutively will initiate auto-increment/auto-decrement mode. In this mode the volume will be incremented or decremented as long as the button remains pressed, at a rate of 10 changes per second.

The mute button toggles the mute status without changing the counter value. If mute is enabled, then any increment or decrement will disable the mute.

2.9.2 ES1888 Mixer Extension Registers

Register 60H Left Master Volume Counter

7	6	5	4	3	2	1	0
0	1: Mute	Left Master Volume					

Register 62H Right Master Volume Counter

7	6	5	4	3	2	1	0
0	1: Mute	Right Master Volume					

Register 64H Master Volume Control

7	6	5	4	3	2	1	0
0	0	1	Read Only HWV int. request	Reserved	1: Enable HWV int. using IRQE	1: Enable HWV int. share w/ audio int.	1: Disable SB PRO Master Vol. Emulation

Bit 0 1: Disable SB PRO master volume emulation
Bit 1 1: Enable H/W volume interrupt request to share Audio IRQ
Bit 2 1: Enable H/W volume interrupt request to use IRQE (potentially sharing with MPU-401)
Bit 3 (read only) Reserved: 1 for ES1888 Mode
Bit 4 (read only) H/W volume interrupt request
Bit 5 (read only) always returns 1. Write value is not significant.
Reset value = 00

Register 66H (write only) clear H/W volume interrupt request

2.9.3 SB PRO Master Volume Emulation

SB PRO emulation for master volume means that the 6-bit volume counters can be written via the SB PRO mixer register 22H (or 32H for ES688 compatibility). SB PRO emulation is enabled by default, and can be disabled by setting bit 0 of mixer extension register 62H.

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

Mute	Master Volume	Value Read @ 32H	Value Read @ 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 SB PRO emulation is enabled, then a mixer reset will cause both left and right channels to be set to their power-on defaults, namely 54 (36H).

If SB PRO emulation is enabled, then a write to mixer register 22H (or 32H) will cause both the left and right master volume registers to be changed as follows:

Value Written to 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

2.9.4 ES1888 Alternate DMA Interface Specification

ALTERNATE DMA MIXER EXTENSION REGISTERS

70H — ALT DAC Sample Rate divider (just like existing extended mode command A1):

Bit 7: 1: high rate (>22KHz)
0: low rate (<22KHz)

This signal generates the RC filter control signal FILTSEL2. It also determines if xtal/36 or xtal/18 is used for the sample rate divider input clock.

Bits 6-0: 2's complement divider that determines sample rate used for alternate dac.

72H — Filter clock divider (just like existing extended mode command A2).

74H — 2's complement DMA transfer count reload register, low byte.

Note: when suspend request bit is set, reading this register returns the current counter contents.

76H — 2's complement DMA transfer count reload register, high byte

Note: when suspend request bit is set, reading this register returns the current counter contents.

78H Control 1:

Bit 0: 1: enable transfer from FIFO to D/A.
0: disable transfer from FIFO to D/A. D/A will receive code 0. FIFO will be flushed.

Bit 1: 1: enable DMA transfer into FIFO.
0: disable DMA transfer into FIFO. This causes the dma counter to be reloaded from the reload register.

This bit will be cleared automatically at the completion of a non-auto-init transfer.

Bit 2: 1: suspend alternate DMA request/hold DMA.

If DMA is in progress, it will halt at the nearest 4-byte boundary, and bit 3 will be cleared.

If DMA is not in progress when this bit is set high (ie, bit 1 is low), then this bit high will hold DMA requests from beginning when bit 1 is set high, until this bit is cleared.

This is used for resume after suspend:
1) bit 1 is cleared.
2) bit 2 is set.
3) 74H/76H are set to count-in-progress.
4) bit 1 is set, thereby latching count-in-progress into counter.
5) 74H/76H are set to reload value
6) bit 2 is cleared, allowing dma to continue.

Bit 3: (read only)
If bit 2 is set high, this flag goes low when a suspend request has completed.

If bit 2 is low, this flag goes low when the FIFO is empty.

Bit 4: 1: auto-init DMA. DMA continues after transfer count expires.
0: DMA halts when transfer count expires.

Bit 5: 1: 16-bit DMA transfer
0: 8-bit DMA transfer

Bits 7-6: 00: single transfer
01: 2 DACKs per DRQ demand transfer
10: 4 DACKs per DRQ demand transfer
11: 8 DACKs per DRQ demand transfer

7AH Control 2:

Bit 0: 1: 16-bit samples
0: 8-bit samples

Bit 1: 1: stereo
0: mono

Bit 2: 1: Signed
0: Unsigned

Bit 4-3: 00: record from output mixer
01: record from record mixer
10: (test) record from FM dac
11: (test) record from ALT dac

Bit 5: 1: enable DRQD output.

Bit 6: 1: enable IRQE output for alt dac dma.

Bit 7: 1: interrupt request active
Write a 0 to this bit to clear the alt dac interrupt request.
Write 1 to generate a test interrupt.

7CH ALT DAC Volume:

Bits 3-0: right channel volume
Bits 7-4: left channel volume

3.0 Programming the ES1888 Enhanced Mixer

The ES1888 has a set of mixer registers that is backward compatible with the Sound Blaster PRO, but with an extended, alternate way of accessing the registers to provide for greater functionality.

Keep in mind that the mixer is read and translated by the ES1888 processor, so that commands written to the mixer registers do not take effect instantly or necessarily in the same order as written.

There are 2 I/O addresses used by the mixer: 2x4H is the address port, 2x5H is the data port. In the Sound Blaster PRO™, 2x4H is write only, while 2x5H is read/write. To set a mixer register, write it's address to 2x4H, followed by the data to 2x5H. To read the register, read from 2x5H after setting the address into 2x4H.

The mixer registers are not affected by software reset. To reset the registers to initial conditions, write any value to mixer address 0:

Write 0 to 2x4H (set mixer address to 0)
Write 0 to 2x5H (write 0 to address 0 to reset mixer)

The Sound Blaster PRO mixer volume controls are mostly 3 bits per channel. Bits 0 and 4 are always stuck high when read. The ES1888 offers an alternative way to write each mixer register: if the address bit 4 is high, all 8 bits of the register are readable and writeable. This is called "Extended Access". If the address bit 4 is low, the interface is Sound Blaster PRO™ compatible, and bits 0 and 4 are cleared by a write, and forced high on all reads.

The Sound Blaster PRO registers that have 3 bits per channel:

Register	**Function**	**Extended Access Reg for 4 bits Per 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 PRO register 04H, you will read back 11H because bits 0 and 4 are stuck high on reads. Inside the register, these bits are stuck low, so that writing 00H is the same as writing 11H.

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

3.1 Extended Access to Mic Mix Volume

If you use register address 0AH to control Mic Mix Volume, only bits 2 and 1 are significant. Bit 0 is stuck high on read and stuck low on writes. Furthermore, this is a mono control: you can't pan. In the ES1888 mixer, the Mic Mix Volume register is 8 bits wide, with the 4 upper bits for left volume, and the 4 lower bits for right volume. Access to this register via address 0AH is mapped as follows:

Write to 0AH:	D2=0, D1=0	Mic Mix Volume = 00H
	D2=0, D1=1	Mic Mix Volume = 55H
	D2=1, D1=0	Mic Mix Volume = AAH
	D2=1, D1=1	Mic Mix Volume = FFH

Read from 0AH: D2 = Mic Mix Volume register bit 3
D1 = Mic Mix Volume register bit 2
D0 = 1
others are undefined.

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

Register	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value
1AH	Mic Mix Vol Left				Mic Mix Vol Right				00H

3.2 Extended Access to ADC Source Select

In the 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

Use register address 1CH for extended access to select recording from the mixer:

D2	D1	D0	Source Selected
X	0	X	Microphone (Default)
0	1	X	CD (Aux) Input
1	1	0	Line Input
1	1	1	Mixer

3.3 Filter Control Registers

The Sound Blaster PRO mixer has three bits that control input and output filters. They are shown in figure 3.1 as bits F2, F1 and F0. They have no function in the ES1888 and their values are ignored.

3.4 Mixer Stereo Control Bit

This is bit 1 of register 0EH. It is normally zero. Set this bit high to enable Sound Blaster PRO™ compatible stereo D/A functions. In this case, you program the D/A 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 D/A converter in Compatibility Mode. It should not be used in Extended Mode.

After any write to this mixer register (0EH), the ES1888 will expect the next 8-bit DMA sample to be for the **right** channel. Subsequent samples will then alternate left, right, left, right.

For 16-bit Stereo DMA to D/A, the data is always expected to be in the order:

Left Low, Left High, Right Low, Right High

And DMA transfers should be in multiples of 4 bytes.

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

3.5 Reading Mixer Address Register, Port 2x4H

The ES1888 provides a way to read back the mixer address register. This is useful for a "hot key" application that needs to change the mixer while preserving the address register.

Register	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value
2x4H (read)	X	X	A5	A4	A3	A2	A1	MXD	N/A

Table 3.1. Sound Blaster PRO Compatible Mixer Map

Register	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value
00H	Write: Reset Mixer								
02H									
04H	Voice Volume Left			X	Voice Volume Right			X	88H
06H									
08H									
0AH	X	X	X	X	X	Mix Mix Vol		X	00H
0CH	X	X	F1	X	F0	ADC Source		X	00H
0EH	X	X	F2	X	X	X	Stereo	X	00H
20H									
22H	Master Vol Left			X	Master Vol Right			X	88H
24H									
26H	FM Volume Left			X	FM Volume Right			X	88H
28H	CD (AuxA) Vol Left			X	CD (AuxA) Vol Right			X	00H
2AH									
2CH									
2EH	Line Volume Left			X	Line Volume Right			X	00H

Note : Filter control bits F2, F1 and F0 have no equivalent function in the ES1888 and are ignored.

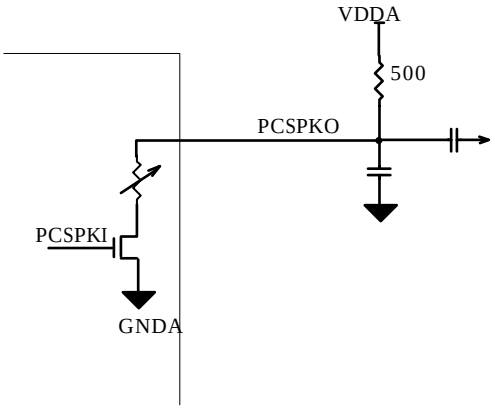
Table 3.2. Extended Access Mixer Map

Register	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value
00H	Write: Reset Mixer								
12H									
14H	Voice Volume Left				Voice Volume Right				88H
16H									
18H									
1AH	Mic Mixer Volume Left				Mic Mixer Volume Right				00H
1CH	X	X	F1	X	F0	ADC Source		X	00H
1EH	X	X	F2	X	X	X	Stereo	X	00H
30H									
32H	Master Volume Left				Master Volume Right				88H
34H									
36H	FM Volume Left				FM Volume Right				88H
38H	CD (AuxA) Volume Left				CD (AuxA) Vol ume Right				00H
3AH	AuxB Volume Left				AuxB Volume Right				00H
3CH						PC Speaker Volume			04H
3EH	Line Volume Left				Line Volume Right				00H

Note : Filter control bits F2, F1 and F0 have no equivalent function in the ES1888 and are ignored.

3.6 PC Speaker Volume Control

A new digital pin will be added to receive a normally low, 1-bit, digital speaker signal. Within the analog section, when this signal is high, a resistive path to analog ground is enabled. The value of the resistor is selected from among 7 choices to control the amplitude of the output signal:



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

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

The purpose of the circuit, beyond volume control of the speaker, is to prevent digital noise from the PC speaker signal being mixed into the 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.

4.0 Programming in Extended Mode

4.1 Introduction to Extended Mode Commands

Commands Ax, Bx, Cx are extensions to the ES488 command set specifically for the ES1888. Commands Ax and Bx are used to access internal registers of the ES1888. The registers are named after the commands for convenience. For example, "register A4", the DMA counter low byte register, is written to via "command A4".

4.1.1 Command C6H Enables Extended Mode Commands

After any reset, you must send command C6H before using any other Extended Mode Command.

4.1.2 Writing ES1888 Internal Registers

Commands Ax and Bx are used to write to ES1888 internal registers. For example, to setup the FIFO DMA counter reload register to F800H, send the following command/data bytes:

A4H, 00H; register A4 = 0
A5H, F8H; register A5=F8

Of course, use the normal polling procedure before sending command or data bytes to port 2xCH. Also, be sure to send command C6H after every reset if you are going to use Extended Mode Commands

4.1.3 Reading ES1888 Internal Registers

Command C0H is used to read the ES1888 internal registers used for Extended Mode. Send command C0H followed by the register number, Ax or Bx. For example, to read register A4H, send the following command bytes:

C0H, A4H

Then poll the read data available bit before reading the register contents.

4.1.4 Mixing Extended Mode Commands with Compatibility Mode Commands

In general, this should be avoided where possible. The Voice Enable/Disable commands D1H and D3H are safe when using Extended Mode to process A/D or D/A conversion. Other Compatibility Mode commands are likely to cause problems. The Extended Mode commands may be used to setup just the DMA or IRQ channels before entering the Compatibility Mode.

4.2 DMA and Interrupt Control Registers

4.2.1 Interrupt Control Register

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
B1	Proc-essor IRQ	Enable IRQ on DMA Cntr Ovf	Enable IRQ on FIFOHE Change	General IRQx tri-state Enable	IS1 Reg	IS0 Reg	X	X	Bits 7-5 Cleared

On hardware reset, the IRQs are disabled, and the ES1888 processor needs to program the select registers bits 3 and 2 of register B1. Bit 4 of this register must be set high to enable one of the four tri-state outputs. When bit 4 is low, all four IRQ outputs are high-impedance.

On any reset, all IRQ sources are disabled by clearing bits 7-5. Bits 1 and 0 are don't-care.

Bit 7, the processor IRQ, is reserved for Compatibility Mode, and should be left zero for Extended Mode. Bit 6 should be set high to receive interrupts for each overflow of the ES1888 DMA counter in Extended Mode. Bit 5 should be set high to receive interrupts for FIFO half-empty transitions when doing block I/O to/from the FIFO in Extended Mode.

The IS1/IS0 Selection Registers (Bits 3 and 2 of register B1H) are interpreted as follows:

IS1	**IS0**	**IRQx Pin**	**Recommended ISA IRQ Channel**
0	0	IRQA	IRQ9
0	1	IRQB	IRQ5
1	0	IRQC	IRQ7
1	1	IRQD	IRQ10

4.2.2 DMA Control Register

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
B2	Proc-essor DRQ	Enable DRQ for Extended DMA	Enable DRQ for Compat. DMA	Enable pulldown inactive DRQ	DS1 Reg	DS0 Reg	X	X	Bits 7-5 Cleared

On hardware reset, the DRQs are all disabled, and the ES1888 processor needs to program the select registers bits 3 and 2. The selected DRQ output is either active or inactive. It is inactive if bits D6 and D5 are both low. The inactive DRQ can either be high-impedance or have an active pulldown device, determined by the value of bit 4 of register B2. The active pulldown device is required for Compatibility Mode to prevent false DRQ requests during the interval between the time that the application programs the system DMA controller and the time the application informs the ES1888 to proceed with DMA. The active pulldown device should be able to sink enough current to override any pullup device on the system motherboard. It is designed to simulate a resistor < 500 ohms. Unless there is a specific reason, bit 4 should be set high.

On any reset, all DRQ sources are disabled by clearing bits 7-5. Bits 1 and 0 are don't-care.

Bit 7, the processor DRQ, as well as bit 5, are reserved for Compatibility Mode, and should be left zero for Extended Mode. Bit 6 should be set high to enable DRQ outputs and DACKB inputs for DMA transfers in Extended Mode. It should be set low for block I/O to/from the FIFO in Extended Mode.

The DS1/DS0 Selection Registers (Bits 3 and 2 of register B2H) are interpreted as follows:

DS1	**DS0**	**DRQx/ DACKBx Pin**	**Recommended ISA DMA Channel**
0	0	None.	- - - - - - - - - - - -
0	1	DRQA, DACKBA	DMA channel 0
1	0	DRQB, DACKBB	DMA channel 1
1	1	DRQC, DACKBC	DMA channel 3

4.3 Miscellaneous Registers for Extended Mode Operation

4.3.1 Extended Mode Sample Rate Generator Register

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
A1	1: > 22 Khz 0: <= 22 khz	Sample Rate Divider							unknown

This register should be programmed for the sample rate for all D/A and A/D operations in Extended Mode.

Clock Source For Sample Rate Generator:
Reg. A1 bit 7=1: 795.5 KHz
Reg. A1 bit 6=0: 397.7 KHz

The sample rate is determined by the 2's complement divider in bits 7-0:
Sample_Rate = 397.7 KHz / (128-X); Reg A1 bit 7 = 0
or
Sample_Rate = 795.5 KHz / (256-X); Reg A1 bit 7 = 1

Where X is the value in bits 7-0 of register A1.

4.3.2 Filter Divider Register

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
A2	Filter Clock Divider								Setup for 8 KHz Sampling

This register controls the low-pass frequency of the switch-capacitor filters inside the ES1888. Generally, the filter rolloff 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. Then use the formula below to determine the closest divider:

Filter_Clock_Frequency = 7.16 MHz / (256-Filter_Divider_Register)

4.3.3 DMA Transfer Counter Reload Registers

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
A4	DMA Transfer Counter Reload --- Low Byte								00H
A5	DMA Transfer Counter Reload --- High Byte								F8H

The FIFO Control Logic of the ES1888 has a 16-bit counter for controlling transfers to/from the FIFO. These registers are the re-load value for that counter, that is, the value that gets copied into the counter after each overflow (plus at the beginning of the initial DMA transfer). The counter will be incremented after each successful byte is transferred via DMA. Since the counter counts up towards FFFF and then overflows, the reload value is in 2's complement form.

For Auto-Initialize DMA, the counter is used simply to generate interrupt requests to the system processor: in this mode DMA continues indefinitely as far as the ES1888 is concerned. In a typical application the counter is programmed to be one-half of the DMA buffer maintained by the system processor. In this case an interrupt is generated whenever DMA switches from one half of the circular buffer to the other.

For Normal Mode DMA, DMA requests will be halted at the time that the counter overflows, until a new DMA transfer is commanded by the system processor. Again, an interrupt request will be generated to the system processor if bit 6 of register B1 is set high.

4.3.4 Input Volume Control Register

Register	D7	D6	D5	D4	D3	D2	D1	D0	S/W Reset
B4	Input Volume Right				Input Volume Left				FFH

This register controls the stereo Input Volume controls. Each channel has 4 bits of control, with a resolution of +1.5 db per step. For recording sources other than the microphone, the input volume varies from -6 db to +16.5 db. For the microphone, the input volume varies from +0 db (no gain) to +22.5 db.

4.4 Programming the FIFO for DMA Playback

4.4.0 Data Formats

There are 8 formats available from the combination of the following 3 options:

- o Mono or Stereo
- o 8-bit or 16-bit
- o Signed or Unsigned

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

4.4.1 Outline Of Programming Steps for DMA Playback

4.4.1.1 Reset:

Write 3 to register 2x6H instead of 1 as in Compatibility Mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to compatibility mode. After the reset, send command C6H to enable extended mode commands.

Reset disables the voice input to the mixer. This is intended to mask any pops created during the setup of the DMA transfer.

4.4.1.2 Program Direction and Type: registers B8H, A8H, and B9H

Register B8H: 0 for normal D/A transfer,
4 for auto-initialize D/A transfer.

Register A8H: modify only bits 1 and 0 of this register, read it first to preserve the remaining bits of this register:

Bits 1,0 = 1,0 Mono
Bits 1,0 = 0,1 Stereo

Register B9H: 0 Single Transfer DMA
1 Demand Transfer DMA: 2 bytes per DMA request
2 Demand Transfer DMA: 4 bytes per DMA request

4.4.1.3 Clocks and Counters: registers A1H, A2H, A4H and A5H

Register A1H = Sample Rate Clock Divider

Register A2H = Filter Clock Divider

Registers A4H/A5H = DMA Counter Reload Register low/high, 2's complement.

4.4.1.4 Initialize and Configure DACs: registers B6H and B7H

The DACs must be configured and initialized with a command sequence depending on the data format:

Mono	Stereo	8-bit	16-bit	Unsigned	Signed	Sequence
X		X		X		Reg B6 = 80H, Reg B7 = 51H, Reg B7 = D0H
X		X			X	Reg B6 = 00H, Reg B7 = 71H, Reg B7 = F0H
X			X	X		Reg B6 = 80H, Reg B7 = 51H, Reg B7 = D4H
X			X		X	Reg B6 = 00H, Reg B7 = 71H, Reg B7 = F4H
	X	X		X		Reg B6 = 80H, Reg B7 = 51H, Reg B7 = 98H
	X	X			X	Reg B6 = 00H, Reg B7 = 71H, Reg B7 = B8H
	X		X	X		Reg B6 = 80H, Reg B7 = 51H, Reg B7 = 9CH
	X		X		X	Reg B6 = 00H, Reg B7 = 71H, Reg B7 = BCH

4.4.1.5 Enable/Select DMA Channel and IRQ Channel: registers B1H and B2H.

Register B1H: Interrupt configuration register.
Make sure bit 4 is high and bit 6 is high,
clear bits 7 and 5.

Register B2H: DRQ configuration register.
Make sure bit 4 is high and bit 6 is high,
clear bits 7 and 5.

4.4.1.6 Configure system interrupt controller and DMA controller.

4.4.1.7 Recommended: delay approximately 100 msec before enabling voice input to mixer.

4.4.1.8 Enable Voice to mixer with command D1H.

4.4.1.9 To Start DMA: Set bit 0 of register B8H high while preserving all other bits.

4.4.1.10 During DMA: for auto-intialize, you do not need to send any commands to the ES1888 at interrupt time, except for reading 2xEH to clear the interrupt request.

For normal mode, you need to initialize the system DMA controller with the address and count of the next block to transfer. You need to update the ES1888 transfer count registers if the count is changed. Then, to start the next transfer, clear bit 0 of register B8H, then set it high again.

To stop DMA at any time, just clear bit 0 of register B8H.

To stop DMA after the current auto-initialize block is finished, clear bit 2 of register B8H, wait for the interrupt, and then clear bit 0 of the B8H.

4.4.1.11 After DMA is finished, restore the system interrupt controller and DMA controller to their idle state. Monitor the FIFO Empty status flag in register 2xCH to be sure data transfer is completed. Delay 25 msec if possible to let the filter outputs settle to D.C. level, then mute the Voice input to mixer with command D3H.

4.4.1.12 Finally, issue another software reset to the ES1888 to leave things in a well-defined state.

4.5 Programming the FIFO for DMA Record

4.5.0 Data Formats

There are 8 formats available from the combination of the following 3 options:

- o Mono or Stereo or Mono Differential
- o 8-bit or 16-bit
- o Signed or Unsigned

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

4.5.1 No AGC for 8-bit Recordings

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

4.5.2 Outline Of Programming Steps for DMA Recording

4.5.2.1 **Reset**: Write 3 to register 2x6H instead of 1 as in Compatability Mode. Bit 1 high specifically resets the FIFO to be empty. The remainder of the software reset is identical to compatability mode. After the reset, send command C6H to enable extended mode commands.

4.5.2.2 Select the input source using the Mixer register 0CH.

4.5.2.3 Program Input Volume: register B4H

4.5.2.4 Program Direction and Type: registers B8H, A8H

Register B8H: OAH for normal A/D transfer,
OEH for auto-initialize A/D transfer.

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

Register A8H: modify only bits 3, 1 and 0 of this register, read it first to preserve the remaining bits of this register:

Bits 1,0 = 1,0 Mono
0,1 Stereo

Bit 3 = 0 Disable Record Monitor for now

Register B9H: 0: Single Transfer DMA
1: Demand Transfer, 2 bytes per DMA request
2: Demand Transfer, 4 bytes per DMA request

4.5.2.5 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=DMA Counter Reload Register low/high, 2's complement

4.5.2.6 Delay 100 msec to allow the analog circuits to settle.

4.5.2.7 Enable Record Monitor if desired:

Register A8H Bit 3=1: Enable Record Monitor (optional)

4.5.2.8 Initialize and Configure ADC: register B7H

The ADCs must be configured and initialized with a command sequence depending on the data format:

Mono	Stereo	8-bit	16-bit	Unsigned	Signed	Sequence
X		X		X		Reg B7 = 51H, Reg B7 = D0H
X		X			X	Reg B7 = 71H, Reg B7 = F0H
X			X	X		Reg B7 = 51H, Reg B7 = D4H
X			X		X	Reg B7 = 71H, Reg B7 = F4H
	X	X		X		Reg B7 = 51H, Reg B7 = 98H
	X	X			X	Reg B7 = 71H, Reg B7 = B8H
	X		X	X		Reg B7 = 51H, Reg B7 = 9CH
	X		X		X	Reg B7 = 71H, Reg B7 = BCH

4.5.2.9 Enable/Select DMA Channel and IRQ Channel: registers B1H and B2H.

Register B1H: Interrupt configuration register.
Make sure bit 4 is high and bit 6 is high.
Clear bits 7 and 5.

Register B2H: DRQ configuration register
Make sure bit 4 is high and bit 6 is high.
Clear bits 7 and 5.

4.5.2.10 Configure system interrupt controller and DMA controller.

4.5.2.11 To Start DMA: Set bit 0 of register B8H high. Leave other bits unchanged.

4.5.2.12 During DMA: for auto-intialize, you do not need to send any commands to the ES1888 at interrupt time, except for reading 2xEH to clear the interrupt request.

For normal mode, you need to initialize the system DMA controller with the address and count of the next block to transfer. You need to update the ES1888 transfer count registers if the count is changed. Then, to start the next transfer, clear bit 0 of register B8H, then set it high again.

To stop DMA at any time, just clear bit 0 of register B8H.

To stop DMA after the current auto-initialize block is finished, clear bit 2 of register B8H, wait for the interrupt, and then clear bit 0 of the B8H.

4.5.2.13 After DMA is finished, restore the system interrupt controller and DMA controller to their idle state.

4.5.2.14 Finally, issue another software reset to the ES1888 to leave things in a well-defined state. This will return the ES1888 to the D/A direction and turn off the record monitor.

4.6 Programming the FIFO for I/O Block Transfer

For some applications, DMA is not suitable or available for data transfer, and it is not possible to take exclusive control of the system for D/A and A/D 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 ES1888 port 2xFH for block transfers.

I/O transfers to FIFO are nearly identical to the DMA process described above, except that an I/O access to port 2xFH replaces the DMA cycle. Some differences are described here.

The DRQ control register B2H bits 7-5 should all be low. This is because no actual DRQ/DACKB cycle is needed.

The IRQ control register B1H must have bit 5 high to enable an interrupt on FIFO half-empty transitions. Bit 6 should be low becaues we don't want an interrupt from the DMA counter.

To progam in this mode you need 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 2xCH. The meaning of this flag depends on the direction of the transfer:

D/A	FIFOHE flag is set high if 0-127 bytes in FIFO
A/D	FIFOHE flag is set high if 128-256 bytes in FIFO

So, for D/A, 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 you transfer somewhat more than 128 bytes. A solution is to poll the ES1888 FIFOHE flag to be sure it goes low in the interrupt handler (or when priming the FIFO) and perhaps send a second block of 128 bytes.

For A/D, 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 2xFH and polling the FIFOHE flag. Its not safe to indiscriminately use the FIFO reset bit 1 of port 2x6H to clear the FIFO, because it may get A/D data out-of-sync

As in DMA mode, bit 0 of register B8H enables transfers between the system and the FIFO inside the ES1888.

Note: The ES1888 is designed for I/O block transfer up to a ISA bus speed of 8.33 MHz.

5.0 ES1888 Power Management Features

5.1 Overview

There are three general categories of power management applications supported by the ES1888:

In the first category, the power supply remains connected to the chip during power-down. The chip has a **partial power-down mode** where the analog remains active while the digital circuits are mostly inactive, as well as a complete power-down that reduces the current to less than 50 microamps. The decision to power-down partially or fully is made by the system processor. To assist the system processor, **activity flags** are available that can be monitored by the system processor to track I/O activity to/from the ES1888. After a predetermined idle period, the ES1888 can be commanded to power-down partially or fully.

If the oscillator clock is provided from an external circuit, **automatic wake-up** upon I/O activity is available. With this feature, the act of reading or writing to a ES1888 I/O port will cause the chip to immediately power-up without losing context from partial or fully powered-down states. If the oscillator clock is provided by a crystal, automatic wake-up from partial power-down is still available because the oscillator will continue to run as long as the ES1888 is not fully powered-down. Once the ES1888 is fully-powered down, however, automatic wake-up is not available with a crystal oscillator because of the start-up requirements of the crystal oscillator. It is then the responsibility of the system software to provide for a start-up period for the oscillator before returning control to the application programs that may access the ES1888. In any case, there is no loss of context.

In the second category, power is maintained as in the first category, except the decision to power-down is made by the ES1888 itself. In **self-timed power-down**, the ES1888 processor waits for a pre-programmed period of I/O inactivity between successive commands and before entering partial or full power-down state. This mode is otherwise similar to the first category except that the decision to power-down is made by the ES1888 processor rather than the system processor.

In the third category, power is removed from the ES1888 during its **suspended state**. Before removing power, the entire context of the processor and registers must be uploaded to the system processor and saved. After restoring power and generating a hardware reset, the opposite **resume** operation must download the context.

5.2 Inputs and Outputs During Power-Down

When powered-down, digital inputs that do not have pull-up or pull-down devices should be driven high or low, that is, they should not be floating. Examples of such pins are A[11:0] and AEN.

Some input pins have circuitry that provides a pulldown device when the ES1888 digital circuits are powered up. During power down, these inputs have a feedback device that latches the input state and prevents leakage current into the pin. The pulldown device is disabled.

The pins that have this feature are: AMODE, SE, DR

Output pins such as DRQx and IRQx will be frozen in their state at power down.

GPO0 and GPO1 may change state during full power-down if so programmed (see section below on programming GPO0 and GPO1 to reflect power-down status).

The MSI pin has an internal pull-up device, so this pin can be left floating during power-down.

The internal inverter connected to pins XI and XO will continue to operate when the digital part of the ES1888 is powered-down as long as 1) SCLK is high, and 2) the analog part is powered-up. When the chip is fully powered-down, the inverter becomes high-impedance with a weak pullup on the XO pin.

VREF will go low when the analog circuitry is powered-down.

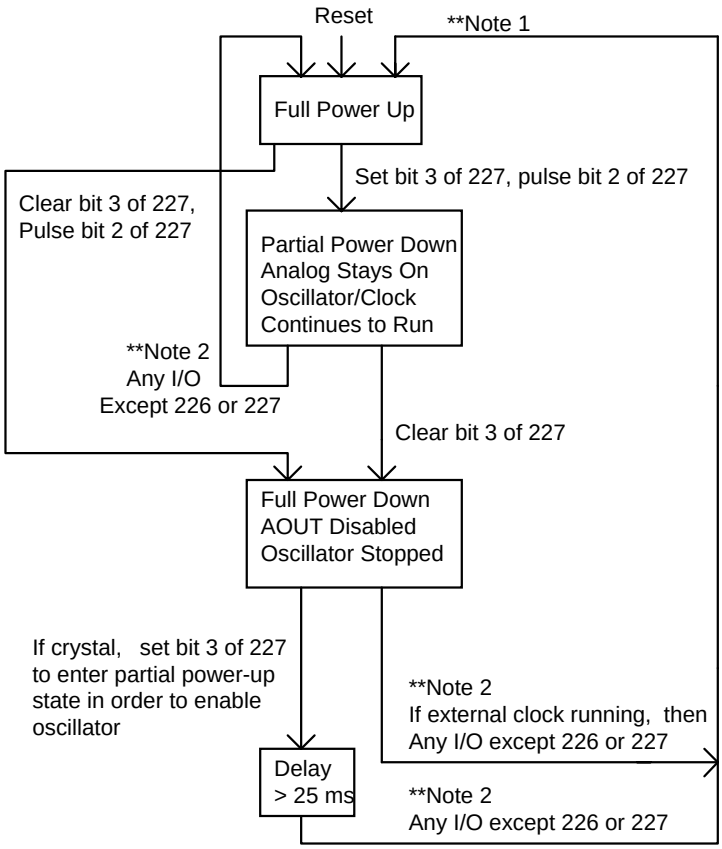
CMR is pulled low by an internal transistor during analog power-down.

The AOUT L/R pins will be held at approximately the idle voltage level with a high-impedance resistor divider.

5.3 Partial Power down (Analog Stays On) vs. Full Power down

The total current used by the ES1888 can be reduced by a factor of 2 or more by putting the ES1888 in a partial power-down state. The crystal oscillator, if used, will continue to operate. The analog circuitry remains powered-up so that Aux A/B, Line, and Mic audio sources can continue to be heard. FM and D/A audio are automatically muted. There should be no pop when returning from partial power-down to a full power-up state. To be in partial power-down mode, bit 3 of register 227H must be high before pulsing bit 2 high, then low, and bit 3 must remain high.

In the full power-down state the oscillator is stopped and the analog circuitry is powered-down. The AOUT L/R pins are left at approximately the reference voltage by a high value resistor divider.



**Note 1: after return to full power-up state from full power-down state, the AOUT L/R analog output pins will not be enabled for 48 to 64 msec. The chip should remain in the full power-up state for at least 64 msec to be sure the AOUT pins are enabled before changing to the partial power-down state. Otherwise the AOUT pins may never get enabled. For this reason it is not possible to go directly from full power-down to partial power-down and have AOUT enabled.

**Note 2: a Low input on any of the three hardware volume control pins (VOLUP, VOLDN, or MUTE) will also act as a wake up event.

5.4 Automatic Wakeup

Automatic wakeup is the method of returning to the full power-up state by I/O activity. The alternative way to power up is via hardware or software reset. With automatic wakeup the context is preserved. Any I/O access to any of the ES1888 port addresses other than 226H or 227H will cause an automatic wakeup.

Auto wakeup will also be triggered by DMA accesses. However, it is unlikely that this will occur if power down is triggered by a period of I/O inactivity, which includes DMA accesses and the I/O operations required to set up the DMA transfer.

Automatic wakeup requires that XI is driven by a clock with stable frequency. If a crystal connected to XI and XO, then the ES1888 cannot be fully powered down and have automatic wakeup work correctly. The reason is that the oscillator will require some time (typically greater than 25 msec) to stabilize.

5.5 Port 226H

Register	D7	D6	D5	D4	D3	D2	D1	D0
226H (write)	0	0	0	0	0	0	FIFO Reset	S/W Reset

Register	D7	D6	D5	D4	D3	D2	D1	D0
226H (read)	Act Flag 2	Act Flag 1	Act Flag 0	Serial Act	0 if Power Down	MIDI Mode	FIFO Reset	S/W Reset

Reading port 226H returns the following information:

Bit 0 Software Reset Register

Bit 1 FIFO Reset Register

Bit 2 1: The ES1888 is processing a MIDI command 30H, 31H, 34H or 35H. In this mode the ES1888 is monitoring serial input. Powering down may cause loss of data. Note that the ES1888 does not automatically wakeup on serial input on the MSI pin.

Bit 3 0: The ES1888 digital section is currently powered down. In this state the power to the analog section is controlled by bit 3 of port 227H.

Bit 4 1: Serial activity flag. High if DSP serial mode is enabled (SE input pin high or bit 7 of mixer extension register 48H is high), or, if an external ES689 is using MCLK/MSD to drive the FM D/A converters.

Activity Flags: These 3 flags are reset low by certain I/O activity. They are all set high each time that this port, 226H, is read:

Bit 5 Activity Flag 0: set low by any DMA read/write or the following:

read/write	222H-223H
write	226H
read	22AH
write	22CH

Bit 6 Activity Flag 1: set low by:

read	22CH
read	22EH

Bit 7 Activity Flag 2: set low by any read/write of an FM port or MPU401 port:

228H-229H, 388H-389H, 3x0H-3x1H.

5.6 Port 227H -- Power Management Register

Register	D7	D6	D5	D4	D3	D2	D1	D0
227H	Suspnd Request	X	1:Reset FM synth	0	Analog Stays On	Power Down Rqst	GPO1	GPO0

Reading or writing port 227H will not auto wakeup the ES1888.

Bit 0 1: Set GPO0 = 1
0: Clear GPO0 = 0 (hardware reset condition)

Bit 1 1: Set GPO1 = 1 (hardware reset condition)
0: Clear GPO1 = 0

Bit 2 Power down Request. Pulse high, then low, to request power down.

Bit 3 1: set Analog_Stays_On
0: clear Analog_Stays_On

Bit 4 Reserved, should be set low.

Bit 5 1: FM Synthesizer Reset
0: Release FM Synthesizer Reset

Bit 6 Unknown (read only).

Bit 7 Suspend Request. Pulse high, then low, to request suspend.

5.7 Example: Powering Down the ES1888 Using System Software Timer Interrupt

In this example it is assumed that the ES1888 is using an external clock source that continues to run even when the ES1888 is fully powered down.

From a timer interrupt routine, you read 226H to monitor activity. After 1 minute of I/O inactivity you conclude you want to power down the ES1888 completely, and then return from the timer interrupt. The ES1888 will wakeup automatically by any I/O access to the ES1888 by any application.

First, check to see if the ES1888 is already powered down (bit 3 of port 226H = 0). If so, there is nothing to do.

Next, check if the ES1888 is being held in reset by reading bit 0 of port 226H. If bit 0 is high, you will have to release the reset before you can power down: clear bit 0 of port 226H, then delay 1 msec or more for the ES1888 processor to complete its initialization routines.

Next, check to see if the ES1888 is in MIDI serial interface mode by testing bit 2 of port address 226H. If so, you may decide not to power down. While the ES1888 can power down when in MIDI mode, it will not automatically wakeup if serial data comes in to the MSI pin, and such data will be lost.

Next, send a power-down request to the chip by clearing bit 3 in register 227H, then pulsing bit 2 first high, then low. The other bits of this register should be preserved. The ES1888 processor will see the rising edge of bit 2 of register 227H as an interrupt request to power-down.

5.8 Self-Timed Power-Down

The ES1888 processor can be programmed to monitor I/O activity in place of the system processor, and after a programmable period of inactivity, enter either a partial or full power-down state.

The ES1888 will need to use the activity flags in register 226H and therefore if this feature is enabled, the system processor will not be able to monitor I/O activity.

To enable Self-Timed Power-Down:

1) Send command C6H to enable access to this feature via the BDH command.
2) Send command BDH.
3) Send the time out value N, where the time period is N x 8 seconds.
4) Send command C7H to disable access to this feature via the BDH command.

If the value N is zero, self-timed power-down is disabled.

Whether the ES1888 enters partial or full power-down is determined by bit 3 of register 227H.

Even if self-timed power-down is enabled, the ES1888 can be commanded to power-down via bit 2 of register 227H.

There is one limitation to this feature: the timing of inactivity only occurs between commands sent to the ES1888. It is possible for a program to leave the ES1888 in a state where timing will not happen. For example, if a program exits without a DMA transfer being completed. Most programs are well-behaved in this respect and leave the ES1888 in a well defined state.

5.9 Suspend/Resume

The term "suspend" is used here to describe the process of uploading the context of the ES1888 and removing digital and analog power to the chip. The term "resume" describes the process of applying power to the ES1888 and downloading the context.

In firmware version 10 of the ES1888, 782 (decimal) bytes are required to store the entire context of the ES1888.

It is possible to suspend the ES1888 regardless of its current state, including suspending in the middle of a DMA transfer. The suspend process is initiated by pulsing bit 7 of port address 227H high, then low. This will interrupt the ES1888 processor and begin a sequence of upload operations.

A hardware reset is required during the resume procedure, before downloading the context. Downloading the context is initiated with command C1H.

An example assembly language program that implements suspend and resume from a TSR is available.

5.10 Using the General Purpose Outputs to Indicate Power down

The ES1888 has the ability to have one or both of the general purpose outputs GPO0 and GPO1 change state when the ES1888 is powered-down.

After hardware reset, this feature is disabled and the general purpose outputs are not affected by power-down. An internal register in the ES1888 must be programmed to enable this feature.

Specifically, the GPO Power Down Control Register is set via command 0CFH and read via command 0CEH. It should be set once by system software after system reset. This register will remain unaffected by soft resets. Using this register, one or both of the general purpose outputs can be programmed to be inverted from its normal state during power-down. The "normal" state of each pin is set by the appropriate bits in register 227H. A further feature allows the inverted outputs to return to their normal state immediately after power up, or after a programmed delay after power up.

Register	D7	D6	D5	D4	D3	D2	D1	D0
GPO Power Down Control	1: restore GPO1 timed	1: invert GPO1 at PDN	1: restore GPO0 timed	1: invert GPO0 at PDN	0	T2	T1	T0

After hardware reset, all bits of this register are 0. This means that GPO0 and GPO1 are unaffected by the power down status, i.e., they remain in the state programmed into register 227H. If bit 6 (GPO1) or bit 4 (GPO0) are high, then the corresponding bit will be inverted from the normal state (register 227H) during power down.

Bit 7 (GPO1) and bit 5 (GPO0) if low indicate that the corresponding output will return to its normal state immediately after the ES1888 wakes up from power down. If high, the corresponding output will return to its normal state after a time period elapses. The time period is determined by the bits T2, T1, T0: A 16 Hz counter will start at 0 and proceed up until it matches the 3-bit number formed by T2,T1 and T0. The maximum delay is 7 * 67 msec or about 469 msec.

Note: "Power down" as used in this document refers to full power-down, i.e., when both the analog and digital parts of the ES1888 are powered-down.

5.11 External Amplifier Power Management and Pop Prevention

Normally an external stereo amplifier chip is used in a ES1888 design in order to directly drive speakers. There are two power management problems associated with an external amplifier:

1) The amplifier itself will draw current unless it can be powered down.
2) Suspend/Resume will cause pops because power is removed from the ES1888 and then re-applied.

Amplifiers such as the SGS/Thomson TDA7233 have a mute input which reduces current to 400 uA and also reduces pops from the suspend/resume process. This part is a mono amplifier, so two are required. Connect GPO0 to the active low MUTE input of the TDA7233. In this case the amplifier will be muted after hardware reset. In an program activated from the AUTOEXEC.BAT file, program the ES1888 so that GPO0 will be high when powered-up and low when fully powered-down. Program a delay of about 133 msec between power-down and power-up states, before GPO0 will return high so that the ES1888 analog circuits can settle.

5.12 Power Management and the Internal FM Synthesizer

The ES1888 FM synthesizer is a fully static design. This means that the clock can be stopped to power-down the circuitry without loss of the state.

Also, for suspend/resume applications, the entire context of the synthesizer can be read back. The details of this procedure are beyond the scope of this document and are covered in an application note concerning suspend/resume.

6.0 Joystick/MIDI Interface

6.1 MPU-401 UART Mode

There are 2 separate MIDI interfaces in the ES1888. The Sound Blaster compatible command set and a MPU-401 "UART Mode" compatible serial port. MPU-401 is a superior method of MIDI serial I/O because it does not interfere with D/A or A/D Sound Blaster commands. Both methods of serial I/O share the same MSI and MSO pins. The MPU-401 interface consists of separate 8-byte FIFOs for receive and transmit.

By default after hardware reset, the MPU-401 interface is disabled. It must be configured using mixer extension register 40H. This register is described in chapter 7.0.

MPU-401 requires an interrupt channel for MIDI receive. This interrupt should be selected using mixer extension register 40H. It should be different than the interrupt selected for audio DMA interrupts.

If MPU-401 is enabled, a low level signal on pin MSI will prevent power-down and cause an autowake event if the ES1888 is powered-down. Likewise, power-down is prevented if a byte is currently being received or transmitted.

Temporarilly disabling MPU-401 using mixer extension register 40H acts as a reset to the FIFOs.

6.2 Joystick/MIDI External Interface

The joystick portion of the ES1888 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 your PC already has a game card or port, either remove the game card or disable the joystick port in the reference design by removing the joystick enable jumper. Disabling the joystick port does not affect its use as a MIDI port.

If you need to run two joysticks, a joystick conversion cable is required. This cable uses a 15-pin D-sub male connector on one end, and a 15-pin D-sub female connector 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 end, pins 12 and 15 should be left without connection. On the female connector end, pin 15 is internally connected to pin 8 , and pin 12 is internally connected to pin 4. The dual joystick port and MIDI port takes up only one slot in your PC leaving room for other cards. The dual joystick/MIDI connector configuration is shown below.

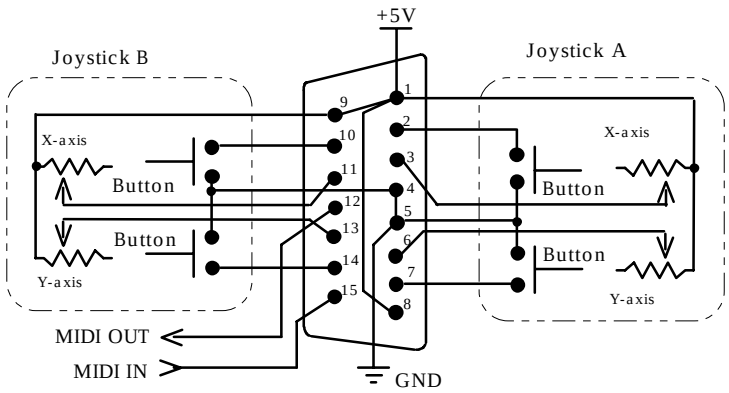


Figure 6.2.1 Dual Joystick/MIDI Connector

The MIDI Serial Interface Adaptor from the Joystick/MIDI Connector is shown in below.

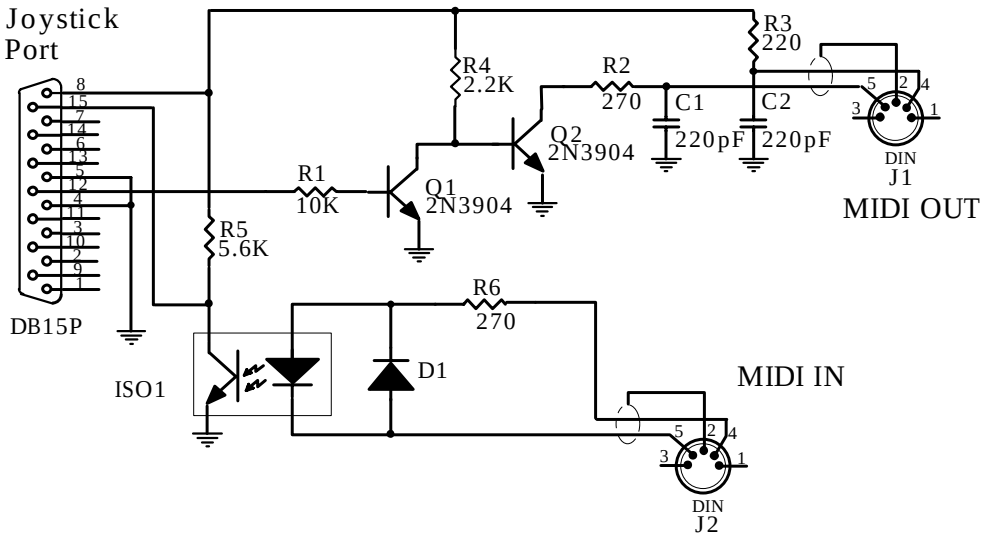


Figure 6.2.2 MIDI Serial Interface Adaptor from Joystick/MIDI Connector

7.0 Software Address Configuration, MPU-401 Configuration, Joystick Disable

Following a hardware reset, the ES1888 is disabled. The part must first be configured using one of two modes of software address configuration. AMODE is a option input that selects between two different software methods of address configuration. When AMODE is low, the Read-Sequence-Key address configuration is selected. When AMODE is high, the System-Control-Register address configuration is selected.

A mixer extension register has been added to allow system software to configure the MPU-401 port, to enable or disable the joystick address decode as well as the decode of 388H-38BH for the intenal FM synthesizer. (The internal synthesizer will still respond to the FM register addresses which are offsets from the ES1888 base address even if disabled at 388-38BH).

AMODE	**Mode**
0	Read-Sequence-Key address selection.
1	System-Control-Register address selection.

7.1 Software Address Selection

There are two methods of selecting the base address of the ES1888 by system software. In each of these two modes, after a hardware reset all I/O will be disabled except as necessary for the configuration process.

These methods of address selection are intended to be used only for applications where the ES1888 is on the system board. Plug-in cards should not use software address selection.

7.1.1 Method 1. Read-Sequence-Key

After hardware reset, or after the reset-sequence command bit is written high (bit 2 of mixer extension register 40H), the ES1888 is in a disabled state. A special sequence of I/O read operations addressing registers 229H and 22BH must be performed followed by a read of the proper base address register. The final read of the sequence selects the base address and enables the ES1888. The sequence will fail if the sequence is interrupted by any I/O write (excluding DMA) or an I/O read of an address not in the sequence.

The key sequence is:

```
Read 229H
Read 229H
Read 229H       ; 3 reads from 229H in a row guarantees reset of key sequence to beginning

Read 22BH       ; advance to state 1
Read 229H       ; advance to state 2
Read 22BH       ; advance to state 3
Read 229H       ; advance to state 4
Read 229H       ; advance to state 5
Read 22BH       ; advance to state 6
Read 229H       ; advance to state 7

Read 220H, 230H, 240H or 250H   ; selects base address and enables ES1888.
```

Once enabled, the ES1888 will not respond to the key sequence again. In order to change the address programming, bit 2 of mixer extension register 40H must be written high. This generates a command to put the ES1888 into its disabled state as after a hardware reset. A read-key-sequence is then required to return to the enabled state.

7.1.2 Method 2. System-Control-Register

After hardware reset the chip is in a disabled state. A system control register internal to the ES1888 must be written in order to enable the ES1888 and select the base I/O address. This register is accessable by system software at any time if the ES1888 receives an "unlock" command. The unlock command is an I/O write to address 0FBH. The corresponding "lock" command is an I/O write to address 0F9H. For both the lock and unlock commands, the data written is not significant.

When unlocked, the ES1888 decodes addresses 0E0H and 0E1H as write-only registers. 0E0H is the index register, 0E1H is the data register. Presently, only one system control register is defined and has the index value 0. To program this register, first write 0 to 0E0H (the index register), followed by the configuration data to 0E1H.

System Control Register 0

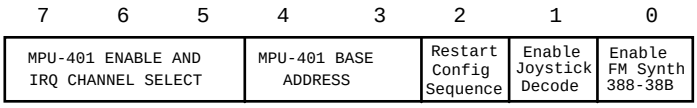
7	6	5	4	3	2	1	0
0	0	0	0	0	CE	AS1	AS0

CE Set high to enable ES1888 I/O decoding.
AS1,AS0 Select register bits to select base address.

Bits 7-3 are reserved and should be written zero.

After external reset, all bits are zero.

7.2 Mixer Extension Register 40h -- FM/Joystick/MPU-401 Configuration



Bit 0 (R/W) This bit when high enables the decode of register addresses 388H-38BH for the internal FM synthesizer. It does not affect FM synthesizer addresses which are offsets of the base register address.

Bit 1 (R/W) This bit when high enables the decode of register address 201H for read and write to access the internal joystick port.

Bit 2 (Write Only) Only used for the Read-Sequence-Key method of software address selection. In this case a write of this bit high will put the ES1888 in its disabled state in anticipation of a new read-sequence-key to change the base register address.

Bits 4-3 (R/W) Selects MPU-401 Base Address:

Bit 4 3	Address
0 0	300H
0 1	310H
1 0	320H
1 1	330H

Bits 7-5 (R/W) Enables MPU-401 and selects interrupt channel:

7 6 5	Function
0 0 0	MPU401 Disabled (hardware reset default)
0 0 1	MPU401 Enabled/ No IRQ
0 1 0	Share IRQ with audio
0 1 1	IRQE
1 0 0	IRQA
1 0 1	IRQB
1 1 0	IRQC
1 1 1	IRQD

Bits 1 and 0 are cleared by hardware reset. Bits 7-3 are cleared by hardware reset. This register is not affected by a mixer reset command.

8.0 DSP / ES689 Serial Interface

The ES1888 contains a synchronous serial interface to an external DSP as well as an ES689 Music Synthesizer. The following new pins are added to support this feature:

Pin	Input/Output	Function
SE	Input w/pulldown	Active high signal from an external DSP to enable Serial Mode.
MCLK	Input w/pulldown	Serial clock from external ES689 music synthesizer (2.75 MHz).
MSD	Input w/pulldown	Serial data from external ES689 music synthesizer. When both MCLK and MSD are active, the stereo D/A converters that are normally used by the FM synthesizer are acquired for use by the external ES689. The normal FM output is blocked.
DX	Tri-state output	Data Transmit. Active output when data is being transmitted serially from the ES1888, otherwise high impedance.
DR	Input w/pulldown	Serial data input.
DCLK	Input w/pulldown	Data Clock. The rate can vary, but a typical value is 2.048 MHz (8 KHz x 256).
FSR	Input w/pulldown	Frame Sync Receive. FSR is either active high or active low based on bit 3 of mixer extension register 48H. The FSR pulse signals the arrival of 8 or 16 bits of data to pin DR.
FSX	Input w/pulldown	Frame Sync Transmit. FSX is either active high or active low based on bit 3 of mixer extension register 48H. The FSX pulse is a request from the external DSP to begin transmission of 8 or 16 bits of data out of pin DX.

8.1 Concurrent Operation of Serial Interface and Other Audio Applications

For the ES689 interface, the ES1888 detects activity from the ES689 by the combination of MCLK being active and MSD toggling within a period of 64 MCLKs. Unless bit 4 of mixer extension register 48H is zero, this activity means that the normal FM output is replaced by the signal coming in the serial port. All other ES1888 audio functions are unaffected.

For DSP serial mode (such as enabled by the SE pin), when it is not necessary to use the ES1888 DMA channel as part of the DSP application, then a concurrently running audio application such as a game, or a Windows audio application can run without knowledge of the DSP application's acquisition of the analog section. Of course, since the D/A and A/D converters are acquired by the DSP, they are not available for the audio application. In this case sound output from the game or other application will be muted. For example, in a game playing audio data, the DMA from the game will continue to operate normally, but the data will be blocked from the D/A converter.

8.2 Serial Data Format

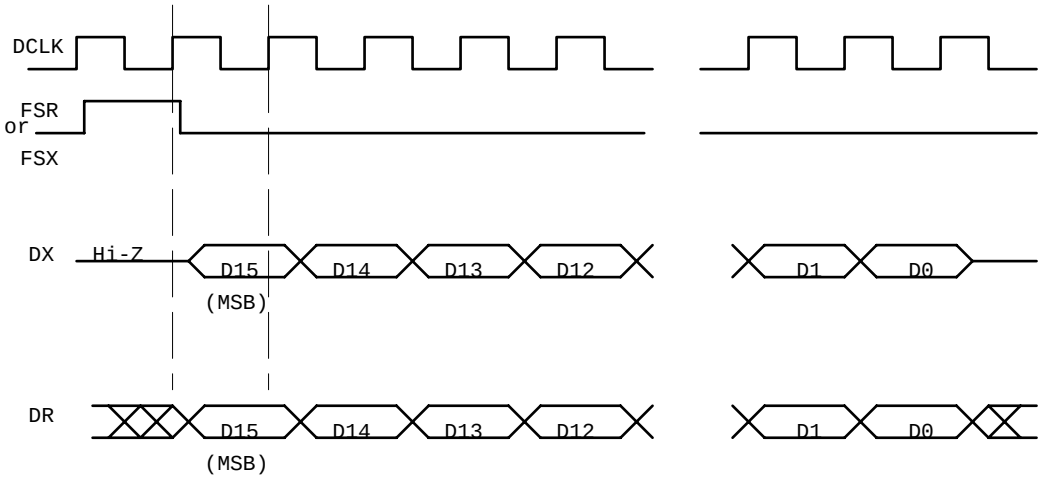
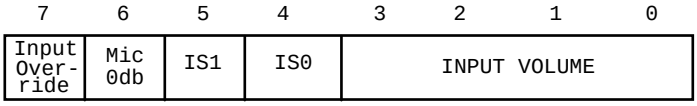


Figure 8.2.1 16-bit data, Positive Sync Pulse

8.3 Mixer Extension Registers

This section describes the new registers related to the DSP and ES689 serial interface. These registers are accessed via I/O addresses 2x4H and 2x5H.

8.3.1 Mixer Extension Register 42H Serial Mode Input Control



Bit 7: If 1, IS1/IS0 and INPUT VOLUME replace normal values as programmed by the system application when the ES1888 is in Serial Mode. Note input volume is mono, and both channels will get this value. If 0, IS1/IS0 and INPUT VOLUME are unchanged during Serial Mode.

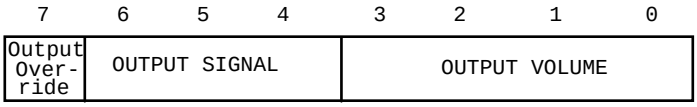
Bit 6: If high, the 26 db microphone preamp is bypassed during Serial Mode (if bit 7 is high).

Bits 5-4: IS1/IS0 select the input source during Serial Mode (if bit 7 is high). These values override the normal mixer settings:

IS1	IS0	Input Source
0	0	LINE
0	1	AUXA (CD)
1	0	MIC
1	1	MIXER

Bits 3-0: Input volume. If bit 7 is high during Serial Mode, this value overrides the input volume settings set via command B4H.

8.3.2 Mixer Extension Register 44H Serial Mode Output Control



Bit 7: If 1, Output Volume during Serial Mode is taken from this register rather than then normal Mixer Master Volume Register. Note that the Output Signal control is always in force during Serial Mode regardless of the state of this bit.

Bits 6-4: These bits control the signal routed to the speaker outputs AOUT L/R.

Bits 6 5 4	Signal
0 0 0	Mute
0 0 1	FDXI monitor in both channels
0 1 0	FDXO monitor in both channels
0 1 1	FDXI monitor in left channel, FDXO in right channel
1 0 0	Mixer output
1 0 1	Mixer output - Wave (ie, Wave input muted)
1 1 0	Mixer output - Wave - FM
1 1 1	Reserved

Bits 3-0: Output volume. Replaces normal mixer Master Volume setting if bit 7 is high during Serial Mode.

8.3.3 Mixer Extension Register 46H Serial Mode Miscellaneous Analog Control

7	6	5	4	3	2	1	0
Analog Control Over-ride	0	Left A/D	Right A/D	AC1	AC0	FDXO Enable	FDXI Enable

Bit 7: If 1, bits 6-0 of this register take effect during Serial Mode. If 0, they do not ever take effect.

Bit 6: Reserved. Should be set to 0.

Bit 5: 1: Left channel combined A/D and D/A converter is in A/D mode.
0: Left channel combined A/D and D/A converter is in D/A mode.

Bit 4: 1: Right channel combined A/D and D/A converter is in A/D mode.
0: Right channel combined A/D and D/A converter is in D/A mode.

Bit 3-2: Analog Control bits 1,0. This special control signals control interconnections in the analog circuitry. They should be set appropriate to the application:

Application	AC1	AC0
Stereo Wave Playback or Record	0	0
Mono Wave Playback or Record	1	1
Full Duplex (Mono Record & Playback)	1	0

Bit 1: 1: Enables FDXO output connection to output pin FOUT R (right channel filter output).
0: FDXO has 50K pullup to CMR.

Bit 0: 1: Enables FDXI input connection to left channel filter input and thus to the input of the left channel A/D converter.
0: FDXI input has 50K pullup to CMR. The left channel filter input and A/D comes from input volume stage as usual.

8.3.4 Mixer Extension Register 48H Serial Mode Miscellaneoous Control

7	6	5	4	3	2	1	0
S/W SE	2's Comp	Serial Reset	Enable ES689 Intfc	Active Low Sync	0	0	0

Bit 7: 1: Force Serial Mode regardless of state of SE pin.
0: Serial Mode controlled by SE pin.

Bit 6: 1: Data format is 2's complement.
0: Data format is unsigned.

Bit 5: 1: Reset serial register left/right toggle flags.
0: Release reset.

Serial Reset also inhibits FDXO connection to FOUT_R, and "zeros" all shift registers.

Bit 4: 1: Enable ES689 to acquire FM D/A converters when serial activity present on pins MCLK and MSD.
0: Prevent ES689 from acquiring FM D/A converters.

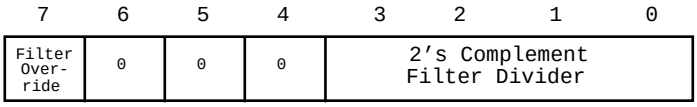
Bit 3: 1: Sync pulses (FSR, FSX) are active low.

Bit 2: Reserved. Always write 0.

Bit 1: Reserved. Always write 0.

Bit 0: Reserved. Always write 0.

8.3.5 Mixer Extension Register 4CH Serial Mode Filter Divider



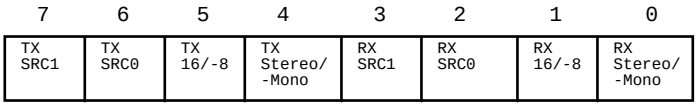
This register controls the filter clock rate during Serial Mode. It is a 2's complement value that divides down the serial clock. The ratio of the filter -3 db frequency to the filter clock is approximately 1:41.

Examples:

02H (-14)	External Serial Clock 2.048 MHz / 14 / 41 = 3568 Hz for 8000 Hz Sample Rate.
0EH (-2)	Internal Serial Clock 1.591 MHz / 2 / 41 = 19.4 KHz for 44,100 Sample Rate. Note that the sample rate divider is an integer multiple of the filter divide for 44,100, which gives maximum performance of D/A and A/D converters!

Bit 7: If 1, during Serial Mode the filter clock is generated by dividing down the serial clock.
If 0, during Serial Mode the filter clock is generated as usual.

8.3.6 Mixer Extension Register 4EH Serial Mode Format/Source/Target



7	6	5	4	3	2	1	0
TX SRC1	TX SRC0	TX 16/-8	TX Stereo/ -Mono	RX SRC1	RX SRC0	RX 16/-8	RX Stereo/ -Mono

Bit 7-6: Transmit Register Source

Bit 7	6	Source
0	0	None: Transmit Register held at "Zero" code
0	1	FIFO
1	0	Left A/D Converter, or stereo A/D transmission
1	1	Right A/D Converter

Bit 5: 1: Transmit length is 16 bits, unsi gned
0: Transmit length is 8 bits, unsigned

Bit 4: 1: Transmit mode is stereo. Left and right channels alternate, with left channel data preceding right channel data.
0: Transmit mode is mono.

Bit 3-2: Receive Register Target

Bit 7	6	Target
0	0	None: Receive Register held at "Zero" code
0	1	FIFO
1	0	D/A Converter (if mono, right channel receives data, left channel receives complement of data)
1	1	FM DAC (if mono, right channel receives data, left channel receives complement of data)

Bit 1: 1: Receive length is 16 bits, unsigned
0: Receive length is 8 bits, unsigned

Bit 0: 1: Receive mode is stereo. Left and right channels alternate, with left channel data preceding right channel data.
0: Receive mode is mono.

ES1888 D.C. Electrical Characteristics

Conditions: VDDD = 5 Volts TA = 0 ^{o}C to 70 ^{o}C

Symbol	Parameters	Min	Max	Unit	Conditions
IILH1	Input Leakage Current High: AMODE,MSD,MCLK,SE,DR,PCSPKI (**Note1) FSR,FSX,DCLK	25	100	uA	Vin = 5.0 V
IILH2	Input Leakage Current High: Other Inputs	0	10	uA	Vin = 5.0 V
IILL1	Input Leakage Current Low: D[7:0] (**Note2)	5	20	uA	Vin = 0 V
IILL2	Input Leakage Current Low: MSI,VOLUP,VOLDN,MUTE	100	500	uA	Vin = 0 V
IILL3	Input Leakage Current Low: Other Inputs	0	10	uA	Vin = 0 V
ICC1	VDDD active		60	mA	VDDD = max. Oscillator rate at 14.318 MHz.
ICC2	VDDA active		40	mA	VDDA = max.
VIL	Input Low Voltage		0.8	V	VDDD = max.
VIH1	Input High Voltage: All except XI	2.0		V	VDDD = min.
VIH2	Input High Voltage: XI	3.0		V	VDDD = min.
VOL1	Output Low Voltage: All except D[7:0], DRQx, IRQx		0.4	V	IOL = 4 mA, VDDD = min.
VOH1	Output High Voltage: All except D[7:0], DRQx, IRQx	2.4		V	IOH = -3 mA, VDDD = max.
VOL2	Output Low Voltage: D[7:0], DRQx, IRQx		0.4	V	IOL = 16 mA, VDDD = min.
VOH2	Output High Voltage: D[7:0], DRQx, IRQx	2.4		V	IOH = -12 mA, VDDD = max.
VOL3	Output Low Voltage: Select DRQx when DMA inactive (**Note 3)		0.4	V	IOL = 0.8 mA

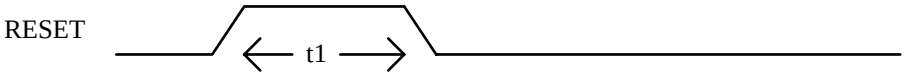
Note 1: These pins have a weak pull-down device that is active except when the ES1888 is powered-down. At the time of power-down, these inputs are latched and held at their current value with a weak keeper feedback that drives the pins.

Note 2: D[7:0] have weak pull-up devices. Their purpose is to pull these pins high if the data bus is not driven by any device. For complete power-down, D[7:0] should be floating or driven high or these devices will source current. Inputs D[15:8] do not have pull-up devices.

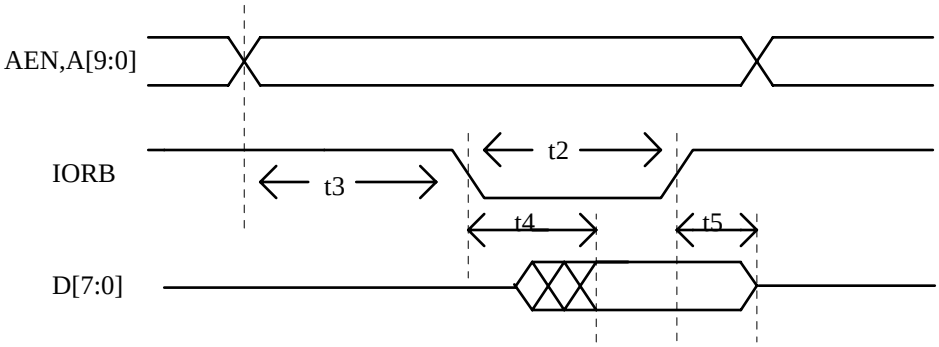
Note 3: Unselected DRQx outputs are high impedance. Selected outputs when DMA is not active have weaker pull-down devices that hold the DRQ inactive unless a stronger driver that shares this DRQ can source enough current to bring this DRQ active.

ES1888 Timing Diagrams

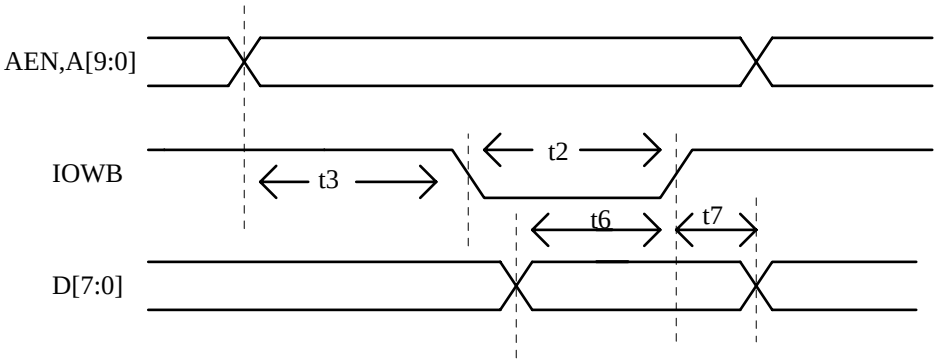
Reset Timing



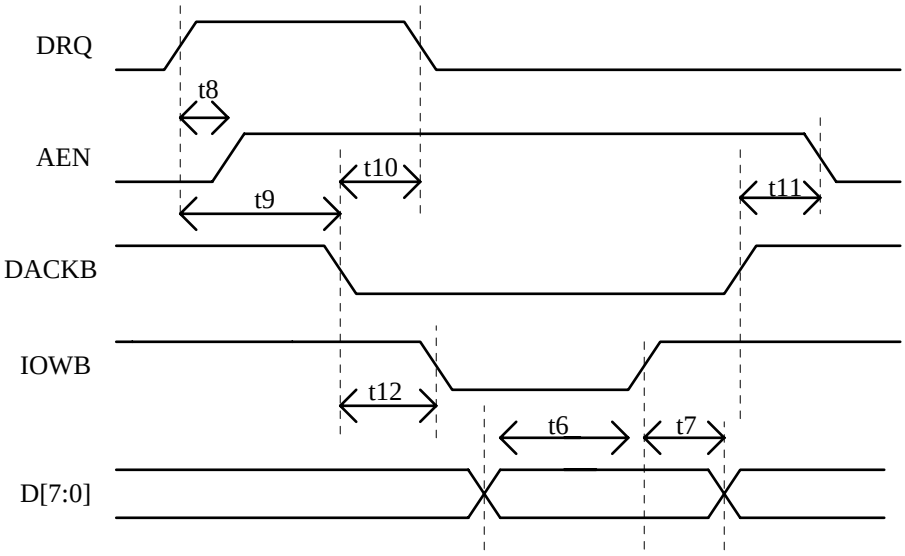
I/O Read Cycle



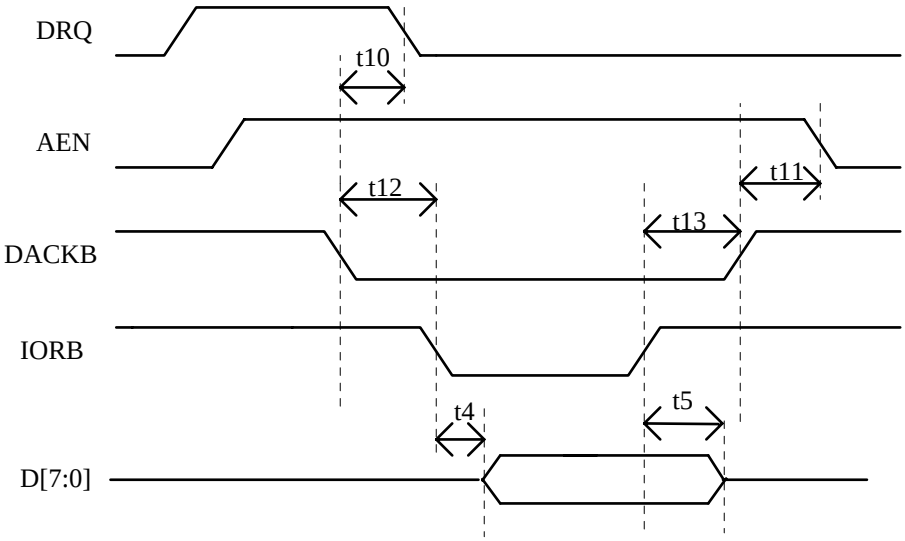
I/O Write Cycle



Compatibility Mode DMA Write Cycle **

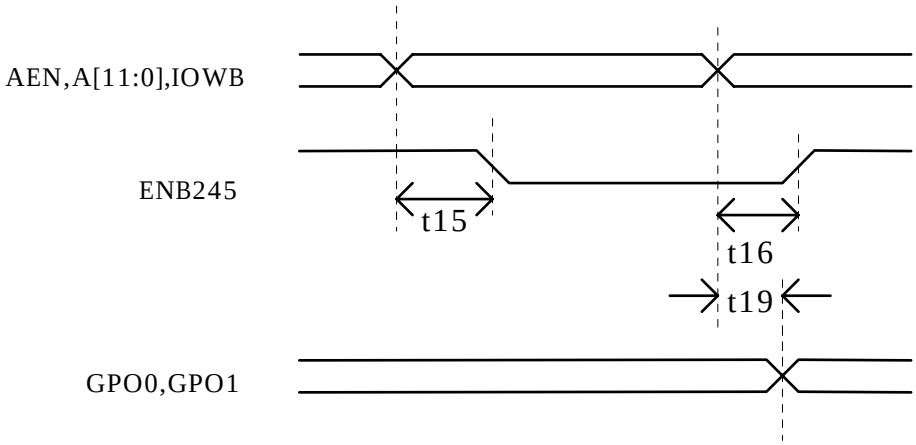


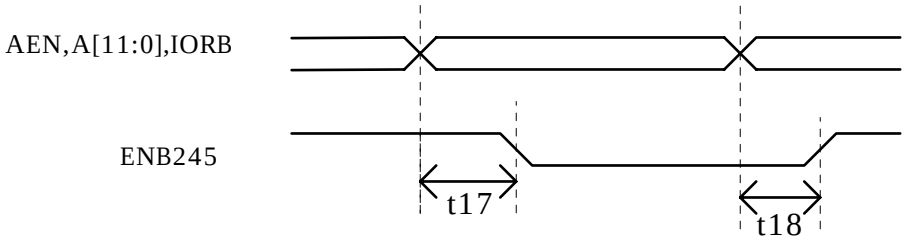
Compatibility Mode DMA Read Cycle **



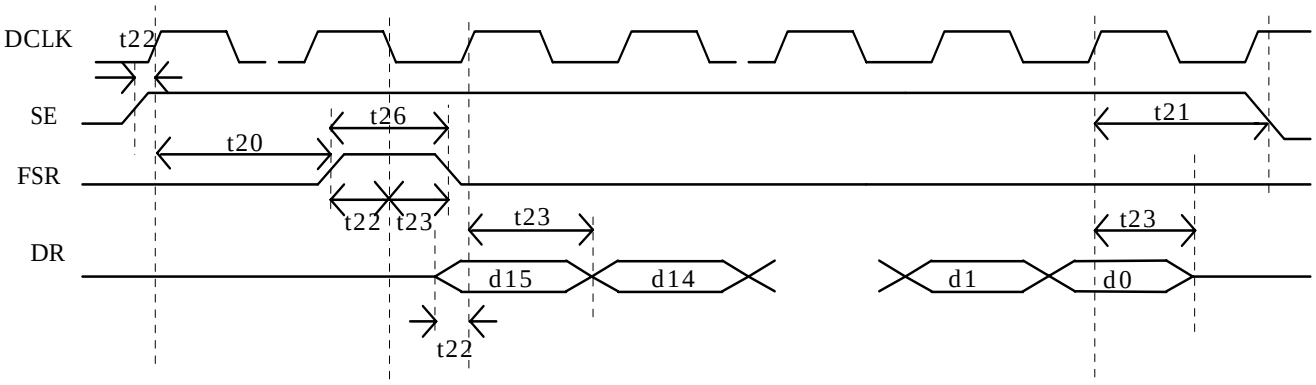
** Note: In Compatibility Mode DMA, the DMA request is reset by the acknowledge going low. In Extended Mode DMA and the alternate DMA channel, the DMA request is reset when the acknowledge signal is low *and* the correct command signal is low, either IORB (for DMA read from I/O device) or IOWB (for DMA write to I/O device). For either of these two modes, the time T10 is relative to the later of the falling edge of the acknowledge signal or the command signal.

Miscellaneous Outputs Signals

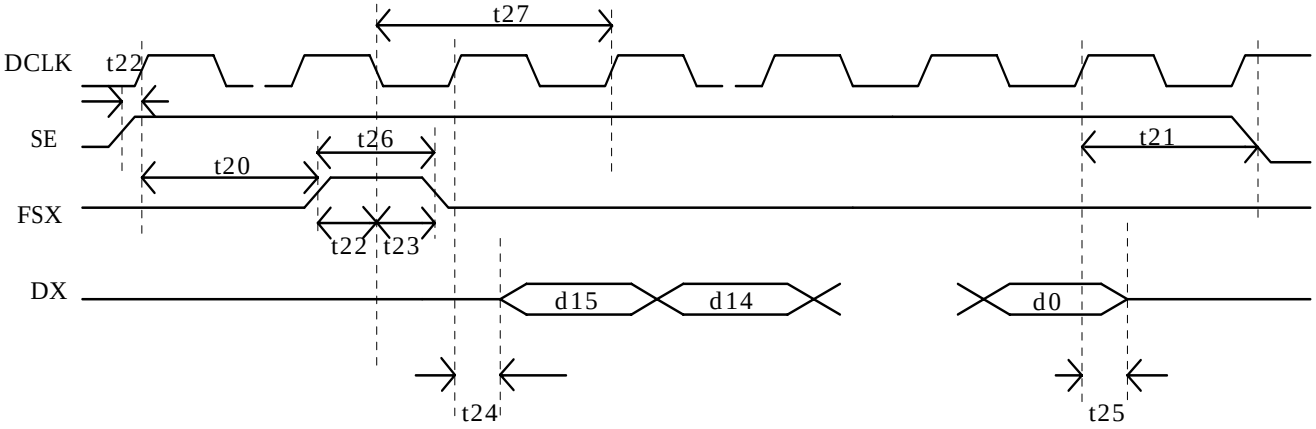




Serial Mode Receive Operation



Serial Mode Transmit Operation



ES1888 Timing Characteristics

Conditions: V_{DDD} = 4.5 to 5.5 volts
$T_A = 0^o$ C to 70^o C

Symbol	Parameter	Min	Typ	Max	Units
T1	Reset Pulse Width	300			ns
T2	IORB, IOWB Pulse Width	100			ns
T3	Address Setup Time	10			ns
T4	Read Data Access Time			70	ns
T5	Read Data Hold Time			10	ns
T6	Write Data Setup Time	5			ns
T7	Write Data Hold Time	10			ns
T8	DMA Request to AEN High	0			ns
T9	DMA Request to DMA ACK	10			ns
T10	DMA ACK to Request Release (**Note 1)			30	ns
T11	DMA ACK High to AEN Low	0			ns
T12	DMA ACK to IOWB, IORB Low	0			ns
T13	IOWB, IORB to DMA ACK Release	20			ns
T14	Crystal Frequency, XI/XO		14.318		MHz
T15	AEN, A[11:0], IOWB to ENB245 Low			20	ns
T16	AEN, A[11:0], IOWB to ENB245 High			15	ns
T17	AEN, A[11:0], IORB to ENB245 Low			25	ns
T18	AEN, A[11:0], IORB to ENB245 High			25	ns
T19	AEN, A[11:0], IOWB, IORB to GPO0, GPO1 Delays			20	ns
T20	SE High to Valid FSR, FSX Edge	2			DCLK
T21	SE Release Time to Last DX, DR Data Bit	1			DCLK

T22	SE, FSX, FSR Setup Time to DCLK Edge	15			ns
T23	SE, FSX, FSR, DR Hold Time to DCLK Edge	10			ns
T24	DX Delay Time from DCLK Edge			20	ns
T25	DX Hold Time from DCLK Edge	10			ns
T26	FSR, FSX Pulse Width	60 ns	500ns		ns
T27	DCLK Clock Frequency		2.048		MHz

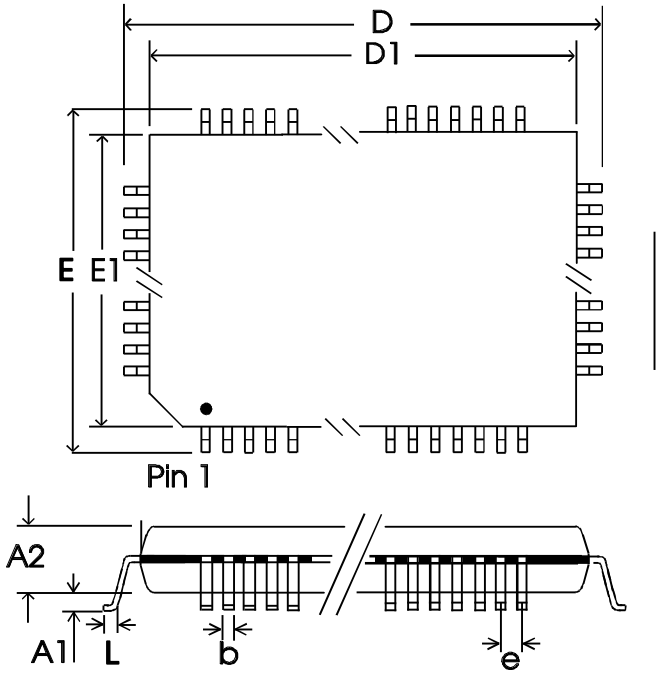
Note 1: In Compatibility Mode DMA, the DMA request is reset by the acknowledge going low. In Extended Mode DMA and the alternate DMA channel, the DMA request is reset when the acknowledge signal is low *and* the correct command signal is low, either IORB (for DMA read from I/O device) or IOWB (for DMA write to I/O device).

ES1888 A.C. Electrical Characteristics

Conditions: V_{DDA} = 4.5 to 5.5 volts

Parameter	Min	Typ	Max	Unit (Conditions)
Reference Voltage: CMR, VREF		2.25		Volts (V_{DDA} = 5.0 V)
MIC input voltage range	10		125	mVp-p
Input Impedance: LINE L/R, AUXA L/R, AUXB L/R, MIC	30K		100K	Ohms
Input voltage range: LINE L/R, AUXA L/R, AUXB L/R	0.5		V_{DDA}-0.5	Volts
MIC Preamp Gain		26		dB
FOUT L/R Output Impedance	3.5K	5K	6.5K	Ohms
CIN L/R Input Impedance	35K	50K	65K	Ohms
Programmable Input Volume Range	0		22.5	dB
Programmable Output Volume Range	-46.5		+10	dB
AOUT L/R full-scale output range	0.5		V_{DDA}-1.0	Volts
AOUT L/R Max load for full-scale output		5K		Ohms

ES1888 Mechanical Dimensions



Symbol	Description	Millimeters		
		Min.	Nom.	Max.
D	Lead-to Lead, X-axis	23.65	23.90	24.15
D1	Package's Outside, X-axis	19.90	20.00	20.10
E	Lead-to Lead, Y-axis	17.65	17.90	18.15
E1	Package's Outside, Y-axis	13.90	14.00	14.10
A1	Board Standoff	0.10	0.25	0.36
A2	Package Tickness	2.57	2.71	2.87
b	Lead Width	0.20	0.30	0.40
e	Lead Pitch		0.65	
L	Lead Length	0.65	0.80	0.95
	Coplanarity			0.102
	No. of Leads in X-axis		30	
	No. of Leads in Y-axis		20	
	No. of Leads Total		100	
	Package Type		PQFP	
	Tray Type		JEDEC 6X11	

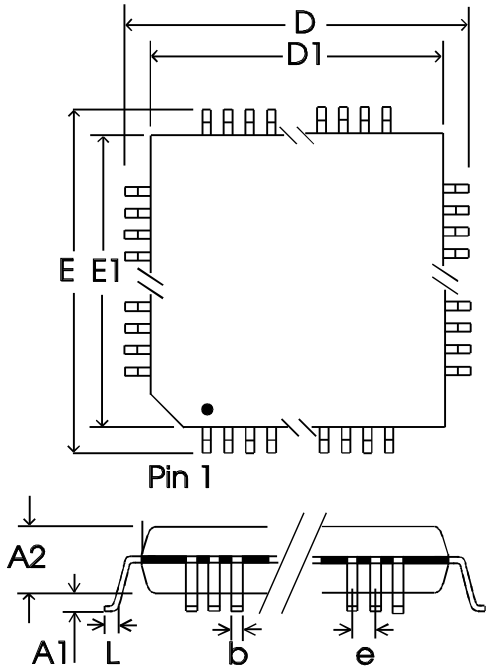
schematics page 1

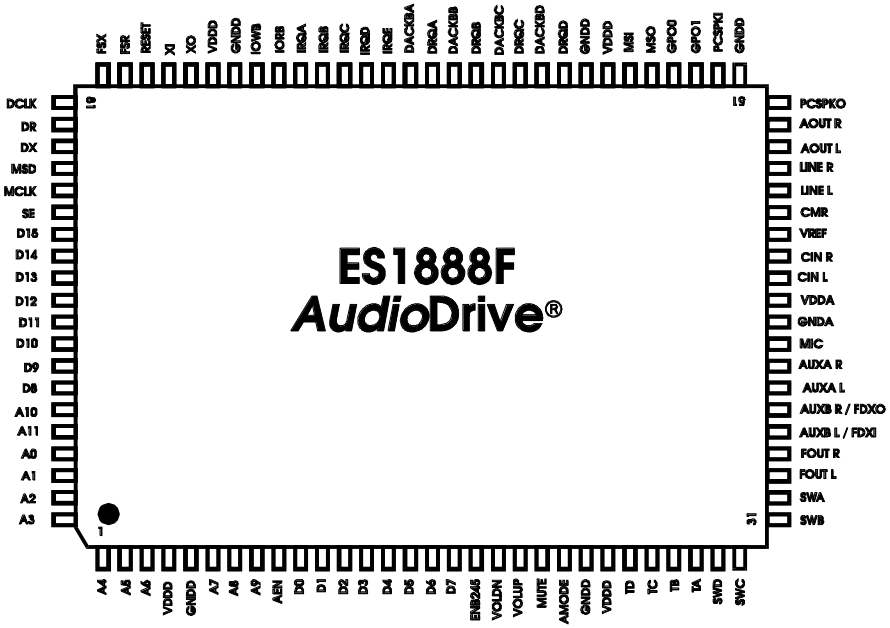
schematics page2

schametics page 3

blank page

ES1888F PACKAGE DIMENSIONS





Symbol	Description	Millimeters		
		Min.	Nom.	Max.
D	Lead-to Lead, X-axis	17.65	17.90	18.15
D1	Package's Outside, X-axis	13.90	14.00	14.10
E	Lead-to Lead, Y-axis	23.65	23.90	24.15
E1	Package's Outside, Y-axis	19.90	20.00	20.10
A1	Board Standoff	0.10	0.25	0.36
A2	Package Tickness	2.57	2.71	2.87
b	Lead Width	0.20	0.30	0.40
e	Lead Pitch		0.65	
L	Lead Length	0.65	0.80	0.95
	Coplanarity			0.102
	No. of Leads in X-axis		30	
	No. of Leads in Y-axis		20	
	No. of Leads Total		100	
	Package Type		PQFP	
	Tray Type		JEDEC 6X11	