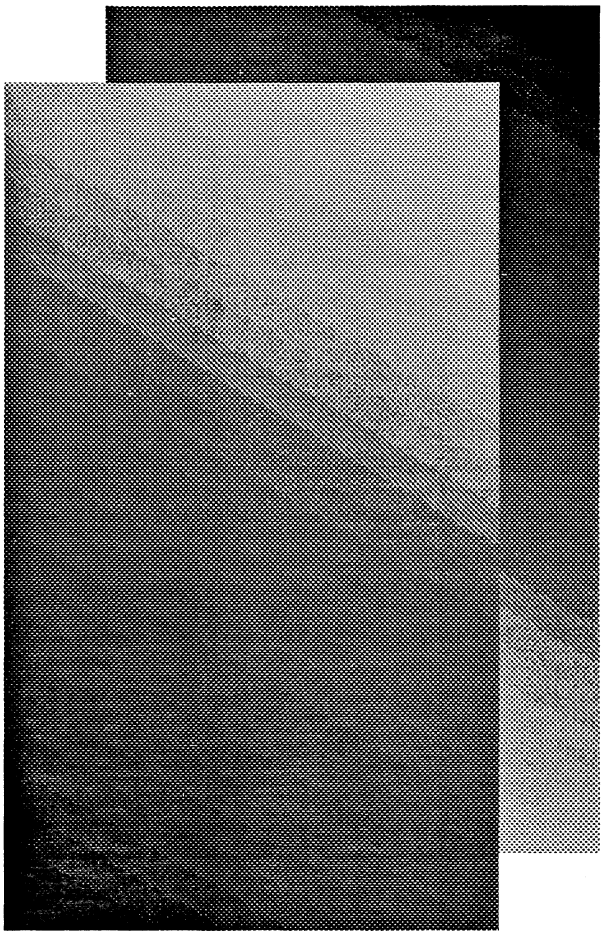


Thunder Board Programmer's Reference

Media Vision

Thunder Board Programmer's Reference

Media Vision, Inc.

3185 Laurelview Court

Fremont, CA 94538

Technical Support (510) 770-9905

BBS (510) 770-0968

Fax (510) 770-8648

June 1992

Trademarks and Copyrights

Media Vision believes this information is accurate and reliable. However, it is subject to change without notice. Media Vision grants to you the right to reproduce and distribute both the run time modules and compiled versions of source, including but not limited to sample program code provided either on disk or provided after purchase by electronic means or through supplemental disk, provided that you (a) distribute the run time modules and compiled source code of the PC as an integral part of your software product, (b) state in your documentation that your product is compatible with the Thunder Board, and (c) agree to indemnify, hold harmless, and defend Media Vision from and against any claims or lawsuit.

This software is protected by both the United States copyright law and international copyright treaty provisions. You may not reproduce any part of the documentation nor software program except in accordance with the above provisions, and for backing up your software for protection from accidental loss.

Media Vision Software License Statement

The software described in this manual is protected by US and international copyright laws. You must not copy the software for any purpose other than making archival copies for the sole purpose of backing-up our software for protection against loss.

The software must not be used on two or more machines at the same time.

Limited Warranty

(2) Media Vision warrants to you that the Software will perform substantially in accordance with the Documentation for a period of one year after delivery to you. You must report all defects and return the Software to Media Vision with a copy of your sales receipt within such period to be eligible for warranty service. If the Software fails to comply with this warranty, your sole and exclusive remedy, at Media Vision's option and cost, will be to either provide all corrections required for any errors, or replace the Software.(3) Media Vision warrants to you that the Hardware will be free from significant defects in materials and workmanship for a period of one year from the date of purchase. Media Vision's sole and exclusive remedy with respect to defective Hardware will be, at Media Vision's option, to repair or replace such Hardware, if it is determined to be defective by Media Vision in its sole discretion, or refund the purchase price for the Hardware.

(4) MEDIA VISION DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE, HARDWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES MEDIA VISION WILL PROVIDE FOR BREACH OF WARRANTY. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, MEDIA VISION MAKES NO WARRANTS, EXPRESS OR IMPLIED, AS TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE.

(5) Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights. You may have other rights which vary from state to state.

Limit of Liability

(6) IN NO EVENT WILL MEDIA VISION BE LIABLE TO YOU FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF A MEDIA VISION REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY PARTY.

(7) Some states do not allow the exclusion of limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Media Vision, Inc.

47221 Fremont Boulevard

Fremont, CA 94538

Technical Support (510) 770-9905

BBS (510) 770-0968

FAX (510) 770-9592

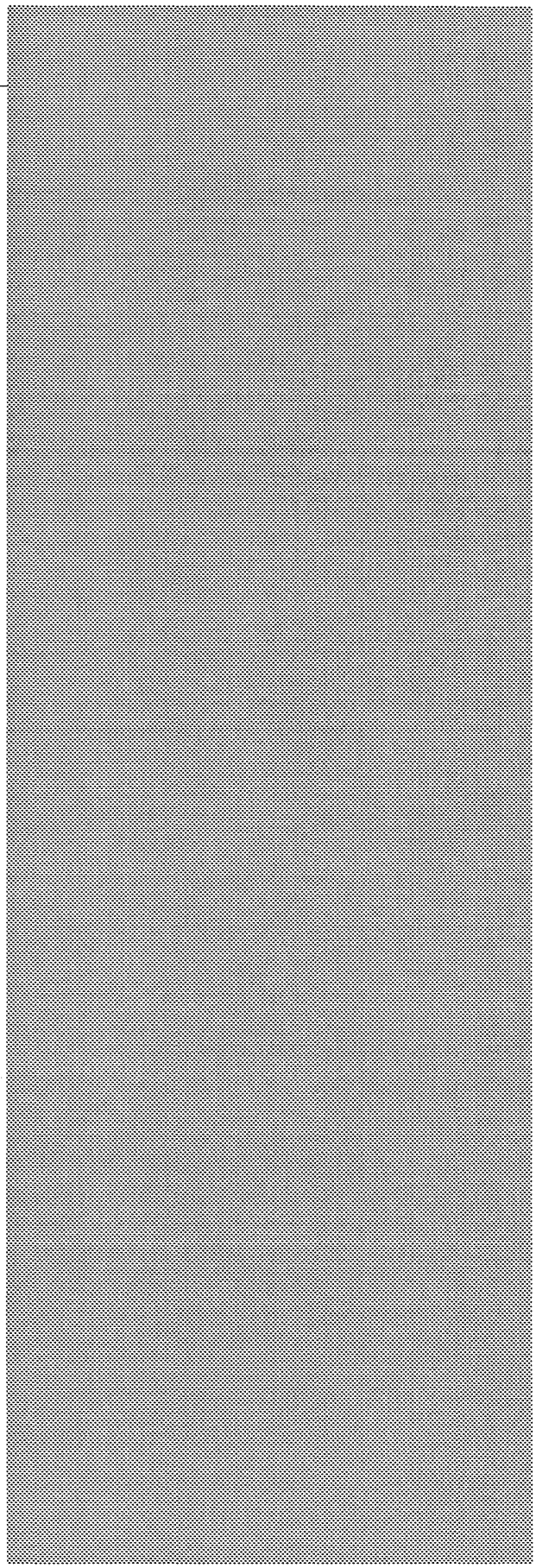
Printed in the United States of America.

This book was written, illustrated, and produced using FrameMaker® workstation publishing software and the ITC Times®, Helvetica® Condensed, Palatino® and Courier® families of typefaces.

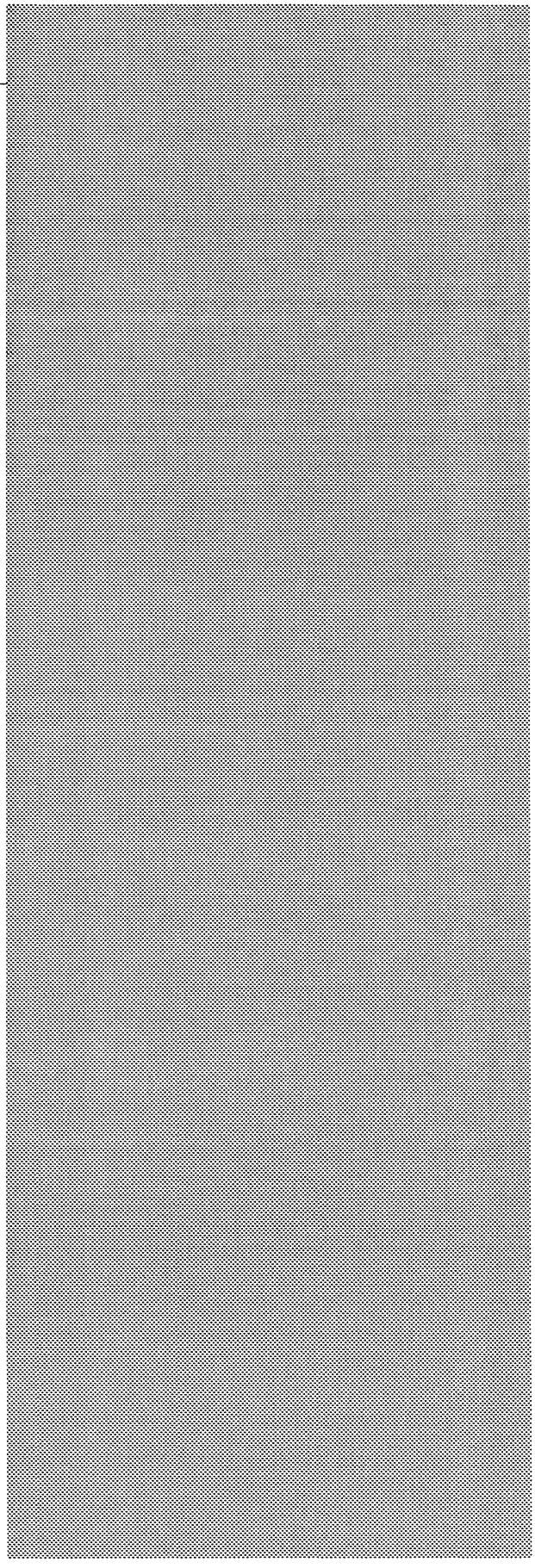
Program team: Doug Cody, Jim Gifford, and Debbie Gronski. Technical writing by Tansy Associates.

Table of Contents

List of Tables



List of Figures



1 *Introduction*

The Thunder Board Programmer's Reference is for experienced PC programmers who want to create applications that incorporate audio capabilities. This manual assumes that you are well grounded in the essentials of the PC and are proficient in hardware -level programming.

This manual covers the following topics:

Digital audio programming basics

- Working with pulse code modulation (PCM) data and data compression
- Maximum PCM sampling rates and performance considerations
- PCM programming using direct memory access

Programming the Enhanced Sound Processor

- Understanding the programming interface to the Thunder Board's digital audio processor, called the Enhanced Sound Processor (ESP)
- Thunder Board digital audio programming procedures

ESP command reference
A guide to all current programming commands for the ESP digital audio processor.

FM synthesizer programming basics

- Features and capabilities of the Thunder Board's FM synthesizer
- Synthesizer sound modes and sound mode configurations
- FM synthesis techniques

Programming the FM synthesizer

- Understanding the FM synthesizer's hardware programming interface
- Reading and writing to registers
- Understanding operator cell and channel number assignments
- Using timers

- Thunder Board FM synthesizer programming procedures

FM synthesizer register functions

A guide to the Thunder Board's FM synthesizer register functions.

Examples of programming the Thunder Board for DMA transfers
Sample algorithms, written in pseudo code, demonstrating how to perform DMA transfers to and from the Thunder Board's ESP.

Rate to time conversion tables
Tables to help compute attack/decay/release values for the FM synthesizer.

Key scaling levels
Table of attenuation values for setting KSL/Total Level registers of the FM synthesizer.

Standard pitch values
Tables of American standard pitch values for setting Block and F-Numbers of the FM synthesizer.

For further information on programming the Thunder Board, call Media Vision technical support at 1-800-348-7116, or use our dial-up bulletin board system by calling 1-510-770-0968.

Thunder Board Overview

The Thunder Board is a PC- compatible sound board that utilizes the sound support built into MS-DOS based games and the Windows 3.1 graphical operating environment. The Thunder Board lets you make or play synthesized music, digital sound effects, and voice recordings in high fidelity, 22kHz audio. Because it is fully AdLib and Sound Blaster compatible, the Thunder Board reliably delivers the sound effects and voice recordings of virtually any PC game.

Features of the Thunder Board include:

- 11- voice FM synthesizer

  The synthesizer plays eleven instrument sounds simultaneously to deliver an orchestra full of rich, dynamic music. Two music modes are built-in: nine melodic instruments, or six melodic and five percussive instruments.

- Digitized voice output channel (DAC) for playing digitized, recorded audio

  Features of the DAC include:

  Programmable playback rate of 4khz to 22kHz

Real-time data decompression (2-to-1, 2.6-to-1, and 4-to-1 compression ratios) at up to 22kHz

Polled mode operation for direct CPU output to the DAC circuitry

DMA mode output with little CPU overhead

Continuous DMA output for achieving higher sample rates

One watt amplifier for driving a personal headset, 8 ohm speakers, or an auxiliary line input to a stereo amplifier

- Digitizing input channel (ADC) for recording microphone level inputs

  Features of the ADC include:

  Programmable sampling rates of 4khz to 22kHz

  Real-time 2-to-1 compression of sampled data up to 12khz

  Polled mode operation for direct CPU input from the ADC circuitry

  DMA mode for minimal CPU intervention

  Continuous DMA input for achieving higher sampling rates

  Automatic gain control (AGC) circuitry to set input levels during recording

- Half-size, 8-bit slot form factor

  Compatible with most IBM PC, XT, AT, '386, '486, PS/2 or clone computers

- Thumb wheel for manual volume control
- Microphone input jack
- Mono line out jack
- Two IBM compatible joystick ports
- Selectable FM synthesizer and joy stick I/O addresses
- Adlib and Sound Blaster compatibility

The following figure shows the physical layout of the Thunder Board including placement of DIP switches, input/output jacks, and key processing components.

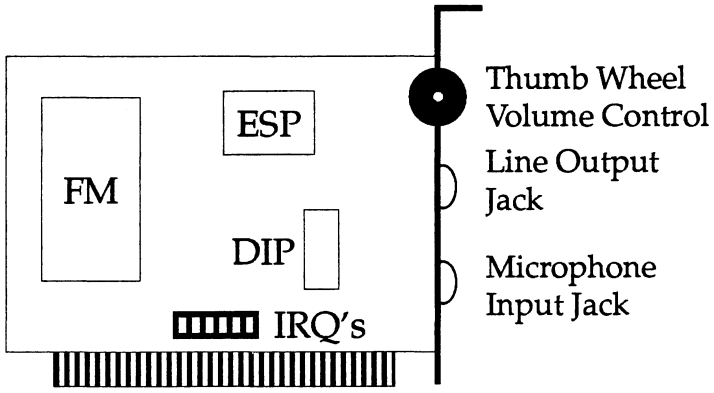


Figure 1 Thunder Board Hardware Diagram

2 *Digital Audio Programming Basics*

This chapter describes:

- The relationship between the quality of an audio sample and pulse code modulated (PCM) compression ratios
- Performance considerations when using data compression
- Alternative approaches for reading and writing PCM data to and from the Thunder Board
- Block oriented programming using DMA

PCM data and data compression

The Thunder Board uses a single byte as it's basic unit of data. This byte may represent up to four different bytes of sampled PCM data.

When recording, the Thunder Board's Analog to Digital Converter (ADC) creates an 8 bit sample byte. This *sample* is a measurement of the voltage level of the input device at the time the sample was taken. In output mode, the Thunder Board's Digital to Analog Converter (DAC) accepts an 8 bit *sample value* to drive the speaker line to a given voltage level. Since the hardware supports only 8 bit ADC/DAC values, the data must conform to this standard.

The Thunder Board's Enhanced Sound Processor (ESP) reads and writes data to and from the ADC/DAC hardware, and performs compression and decompression on -the- fly. The Thunder Board's compression format is known as *Adaptive Delta Pulse Code Modulation* (ADPCM) and is compatible with the Sound Blaster compressed format. With ADPCM, the sample value is represented as a scalable delta between itself and the previous sample (not the true, absolute value read from the ADC). In other words, only the difference between samples is saved, not the true sample. With this technique, the true physical samples may be represented by 2, 3 or 4 bits.

While data compression can improve efficiency, representing a sample with fewer bits increases the probability of sample errors. This error rate translates directly into audible noise. Higher compression rates generate greater amounts of noise. Two- and three-bit compression always have high noise levels. Four bit compression introduces little noise, and produces acceptable quality audio.

The following table lists data formats supported by the Thunder Board:

Bits per sample	Compression ratio
8	No compression
4	2-to-1
3	2.6-to-1
2	4-to-1

Table 1 Thunder Board compression ratios.

Thunder Board can play (output) all four data formats, but can only record 8-bit and 4-bit data. You can record in 2- and 3-bit formats by sampling in 8-bit format and using your own software algorithm to perform compression.

Determining file size requirements based on sample rate

To determine the file size required to capture audio samples, multiply the *sample size* by the *capture period* and adding 2,000 bytes of additional space for the file header. This algorithm is expressed as follows:

(capture period * sample size) + 2,000 = file size

The capture period is the length of time you will be capturing data and is expressed in seconds.

Sample size is equal to the sample rate (expressed in thousands of Hertz, or Hz) divided by the compression ratio you're using. For example, the sample size for un-compressed 8-bit data recorded at 22 kHz is 22,000 bytes per second. Sample sizes for compressed data formats are shown in the following table. These examples assume a 22kHz sample rate; you may specify other sample rates.

Sample rate	÷ Compression ratio	= Sample size
22kHz	1-to-1 (8-bit samples)	22,000 bytes/second
22kHz	2-to-1 (4-bit samples)	11,000 bytes/second
22kHz	2.6-to-1 (3-bit samples)	8,462 bytes/second
22kHz	4-to-1 (2-bit samples)	5,500 bytes/second

Table 2 Sample sizes for given compression ratios.

Performance considerations for using compression

Higher compression ratios result in lower actual transfer rates between the DMA and the Thunder Board. This slower transfer rate, combined with having less data to move, frees up more time for the CPU to do other work. As a result, it may be advantageous to use compression with CPU intensive applications such as live action video games.

Maximum sample rates for PCM I/O

Each of the four PCM data formats have different play and record rates. The following table lists the maximum samples rates for each format. The minimum sample rate for all formats is 4k.

Data format	Recording	Playback, non-continuous DMA	Playback, continuous DMA
8 bit Data	22khz	22kHz	22kHz
4 bit Data	12khz	20khz	22kHz
3 bit Data	N/A	21khz	22kHz
2 bit Data	N/A	21khz	22kHz

Table 3 Maximum Thunder Board PCM sample rates.

Setting the sample rate higher than the maximum value for a given data format will cause a pitch bend (lower pitch) during record and playback.

Programming approaches for PCM data transfers

There are three approaches for reading and writing PCM data to and from the Thunder Board: *direct, automatic* and *enhanced.*

- Direct data I/O using the CPU

  The direct transfer mode offers the easiest approach to transferring data, but the trade off is almost 100 percent CPU usage.

  In this approach, the CPU writes or reads each sample at a high, fixed rate. This generally leaves little time for the CPU to do much else. The direct mode is most appropriate for simple sounds such as blips, beeps, and barks.

- Automatic transfers using DMA

  The automatic transfer mode involves a more elaborate programming approach, but uses the PC's DMA controller to perform the majority of the work.

  Automatic mode assumes the data comes in blocks. It involves programming the PC's DMA controller and interrupt controller; hooking an interrupt vector; and servicing interrupts. This approach is not difficult, just complex.

- Enhanced automatic transfers using continuous DMA

  The enhanced transfer mode is almost identical to the automatic mode, but you program the DMA controller and the ESP to transfer data continuously, without interruption. When the last block is loaded into the DMA buffer, a special command is issued to the ESP to finish the block, then stop processing.

Both the automatic and enhanced modes are appropriate for lengthy, sustained transfers.

Since the functionality of the automatic and enhanced modes are similar, all future references to automatic transfers will refer to both approaches, unless specifically stated.

Use the following procedure to program automatic data transfers:

1. **Hook the correct IRQ vector.**

   The IRQ you use is dependent upon the jumpers have been set on the Thunder Board. The factory default is IRQ7, Int 0Fh.

2. **Enable the interrupt via the interrupt mask register.**

   The bit corresponding to the specific IRQ number is set to a 0 to enable the interrupt and a 1 to disable it. For example, bit 7 set to 0 enables IRQ7 interrupts.

3. **Program channel one of the DMA controller.**

   For information on programming the DMA controller, see "Programming the PC's DMA Controller" on page 3-4.

4. **Program the ESP to start transferring data.**

   For information on programming the ESP, see Chapter 3, "Programming the Enhanced Sound Processor." and Chapter 4, "ESP Command Reference."

ADPCM Compression Data Stream Format

As implemented on the Thunder Board, adaptive delta pulse code modulation (ADPCM) interprets each compressed datum as the delta between two samples. Since this interpretation is dependent on a prior sample, the first data byte must be a true, 8-bit PCM value.

When the Thunder Board generates ADPCM recordings, the data stream consists of a leading 8-bit, un-compressed PCM sample followed by the ADPCM data. Likewise, during ADPCM playback, the first byte in the data stream is assumed to be an uncompressed PCM sample.

In Chapter 4, "ESP Command Reference," reference byte always refers to the leading 8-bit PCM value in the data stream.

3 *Programming the Enhanced Sound Processor*

This chapter describes the Thunder Board hardware programming interface and outlines the four basic steps involved in programming the Thunder Board's Enhanced Sound Processor:

- Resetting and initializing the hardware
- Writing commands and data
- Reading data
- Handling interrupts

Hardware programming interface

The Thunder Board logically occupies a range of PC I/O addresses at one of several selectable locations. These locations, which are jumper selectable, are 210H, 220H, 230H, 240H, 250H, or 260H. The factory default setting is pre-configured for 220H.

Within this range of addresses, there are four specific I/O locations for addressing the Enhanced Sound Processor (ESP):

- 2x6H for resetting the ESP
- 2xAH for reading data from the ESP
- 2xCH for writing commands and data to the ESP, and reading the command ready status bit
- 2xEH for reading the Data Available status bit, and clearing interrupt generated by the ESP

I/O addresses 200H through 207H are used for the standard IBM-compatible joystick port.

Resetting the ESP

Start your application by resetting the ESP. Performing this procedure guarantees that it is in an initialized, functional state:

1. **Write a 1 to the RESET PORT, 2x6H.**

   When this bit is held for up to 3 micro-seconds, it activates the reset line to the ESP causing an immediate hardware reset.

2. **Wait for 3 micro-seconds.**

   Since many IBM compatibles run at different speeds, the best way to guarantee a 3 micro-second wait is to perform 3 I/O reads from port 2x6H (or any other I/O address above 100h). Because the I/O bus runs at the standard 8Mhz speed, all CPUs (286s, 386s, or 486s) will be slowed down to this speed to assure a proper bus I/O operation.

3. **Write a 0 to the RESET PORT.**

   This frees the ESP from the reset state.

4. **Read I/O address 2xEH until Bit 7 (the most significant bit) is set equal to 1.**

   This is the DATA AVAILABLE state bit.

5. **Read address 2xAH to fetch the ESP's READY BYTE.**

   The ESP returns a 0AAh in the READ DATA PORT upon completing a Power-On-Reset.

Note: If the ESP has not returned a 0AAh after a maximum of 100 micro-seconds, verify that you are using the correct I/O address and try again.

If you are using the correct I/O address, it is possible that the hardware has failed to perform the Power-On-Reset initialization. In this case, attempt to locate a working I/O address by continuing the reset sequence using all possible I/O address ranges. If this does not work, declare a hardware failure and take whatever steps are necessary in your application.

Writing commands and data to the ESP

Write commands and data to the ESP through the WRITE COMMAND/DATA PORT using the following handshaking procedure:

- Read the WRITE COMMAND/DATA PORT at I/O address 2xCh.

  Bit 7 (the most significant bit) will be "0" if the ESP is ready to receive a command or data byte. If the bit is a "1", then the ESP has not yet read the last command/data byte sent from the CPU.

Once bit 7 is "0", the CPU may write the next command/data byte to address 2xCh.

Reading data from the ESP

Read data from the ESP using this procedure:

1. **Read the DATA AVAILABLE STATUS PORT at I/O address 2xEh until bit 7 is set to 1.**

   This indicates that a byte from the ESP is available.

2. **Read the READ DATA PORT at address 2xAh to fetch the data byte.**

   Bit 7 of 2xEh is automatically cleared (set to 0) each time the CPU reads port 2xAh.

Handling interrupts generated by the ESP

The ESP generates a hardware interrupt when it completes a block transfer to or from the CPU. In automatic mode, this interrupt allows the CPU to be notified that the block transfer is complete and that the ESP is available for additional work. In enhanced mode, the IRQ signal tells the CPU that the current block operation has completed and that the ESP is continuing on to the next block. For information on automatic and enhanced mode DMA, see "Programming approaches for PCM data transfers" on page 2-3.

To receive a hardware interrupt, your application must load the proper interrupt vector table entry using a routine address. Assuming that the DMA controller has been programmed, and that the ESP has been programmed to process a block of data, this interrupt routine will be called using the PC's hardware interrupt system.

1. **Save any registers used by your program.**

2. **Read the DATA AVAILABLE STATUS PORT at address 2xEh.**

   This acknowledges the interrupt on the Thunder Board.

3. **Write an EOI (End Of Interrupt) command, 20h, to I/O address 20H.**

   This acknowledges the interrupt within the motherboard's interrupt system circuitry and allows further interrupts to occur.

4. **Perform additional buffer management to continue or complete the DMA process.**

   In order to finish an automatic DMA process, acknowledge the interrupts and return.

   In the enhanced mode, send a Finish current enhanced DMA transfer command (DAh) during the last block being transferred. For information on this

command, see "Finish current enhanced DMA transfer (DAh)" on page 4-10.

An alternative approach is to write a Halt DMA transfer command (D0h) to the WRITE COMMAND/DATA PORT. This immediately stops the ESP from performing additional data transfers. For information on the Halt DMA transfer command, see "Halt DMA transfer (D0h)" on page 4-9.

Note: A Halt DMA command leaves the ESP state intact, so the process may be restarted using a Continue DMA command. You typically use this command sequence to pause and then resume DMA transfers.

Programming the PC's DMA Controller

The scope of the following discussion is limited to programming the PC's 8237 DMA controller for memory transfers to and from the Thunder Board. For more details on the 8237, refer to the 8237 data sheet. Information on other aspects of programming the 8237 can be found in many of the popular PC programming books available in computer book stores.

Before programming DMA memory transfers, you should understand two significant limitations of the 8237 and the PC's DMA architecture: 1) DMA cannot cross a 64k boundary and, 2) the DMA controller uses a zero-based counting scheme.

Accommodating the 64k boundary limitation

The DMA controller is a 16-bit device and can only read from a 64k address space. Since the PC addresses a minimum of 1 megabyte, PC designers had to add hardware to support this smaller memory model. As a result, programming is more difficult due to restrictions placed on the location and length of the DMA buffer.

A special page *register* was created with the original PC and carried forward with the AT. This register receives a value indicating which 64k page of memory the DMA controller will read from or write to. The 1 megabyte

address space is divided into 16 separate pages. The following table lists the legal page register values for the 16 pages of memory:

Page Register	Segment:offset
0	0000:0000 thru 0000:ffff
1	1000:0000 thru 1000:ffff
2	2000:0000 thru 2000:ffff
3	3000:0000 thru 3000:ffff
4	4000:0000 thru 4000:ffff
5	5000:0000 thru 5000:ffff
6	6000:0000 thru 6000:ffff
7	7000:0000 thru 7000:ffff
8	8000:0000 thru 8000:ffff
9	9000:0000 thru 9000:ffff
A	A000:0000 thru A000:ffff
B	B000:0000 thru B000:ffff
C	C000:0000 thru C000:ffff
D	D000:0000 thru D000:ffff
E	E000:0000 thru E000:ffff
F	F000:0000 thru F000:ffff

Note: The page register on AT class machines has been increased to allow access to the full 16 megabytes of address space so the page register can be programmed with values of 00h to FFh.

It is important to keep track of which page of memory your program is using. Programs must load a proper starting address and block length. If the starting address plus the block length exceeds 64k, the DMA controller will wrap back to offset 0000 within the current page; it will not increment the page register and continue through memory.

Understanding the DMA controller zero-based counting scheme

You encounter the second limitation when programming the block size. Since 64k (64*1024 = 65536 = 10000h) is really represented using 17 bits, the DMA controller was designed to use a zero based counting scheme. As a result, a count of zero represents a one byte transfer.

The Thunder Board currently uses only DMA channel 1 to transfer PCM data. This may change in the future.

The table below lists DMA registers that are used to transfer data to and from the Thunder Board.

I/O address	Description
00AH	Write Single Mask Register. Used to enable and disable the DMA channel.
083H	Page Register. Described in detail above.
00CH	Clear Byte Pointer Flip-Flop. Used to clear an internal flag in the controller for loading 16-bit values.
002H	Base Address. Contains the 16-bit starting address of the DMA buffer.
003H	Base Word Count. Contains a 16-bit value representing block length.
00BH	Write Mode Register. Used to indicate the mode of transfer.

Your program must set up DMA registers before initiating a DMA transfer using an ESP command. The ESP assumes that the DMA controller is ready to go when a DMA command is sent from the CPU.

DMA programming procedure

Use the following procedure to program DMA registers for memory transfer:

1. **Send a 05h to I/O address 00AH (Write Single Mask Register) to disable channel 1.**

   Channel 1 must be disabled until all the other registers are set up.

2. **Output the page number to I/O address 083H (Page Register).**

3. **Clear the byte pointer flip/flop by writing any value to I/O address 00CH (Clear Byte Pointer Flip-Flop).**

   The flip/flop controls the order of byte loading in the Base Address and Base Word Count registers. Clearing the flip/flop sets up these registers to receive the 16-bit values in the order of low byte first and high byte second.

4. **Output the base address to I/O address 002H (Base Address).**

   Send the low byte first, then the high byte of the base address. Remember, this is a zero-based number.

5. **Output the base word count to I/O address 003H (Base Word Count).**

   Send the low byte first, then the high byte. Again, this is a zero-based number.

6. **Output the DMA transfer mode.**

   Send one of the values listed below to I/O address 00BH (Write Mode Register).

Value	DMA transfer mode
49h	Non-continuous, playback.
59h	Continuous, playback.
45h	Non-continuous, record.
55h	Continuous, record.

7. **Enable DMA channel 1.**

   Send a 01h to I/O address 00AH (Write Single Mask Register). This enables the DMA channel for full operation.

Note: For all DMA transfers, remember to program the sample rate and speaker on/off state before initiating the transfer.

Programming the PC Motherboard's Interrupt Controller

The Thunder Board's ESP interrupts the CPU at the end of DMA transfers using the PC's 8259 interrupt controller. The 8259 allows up to eight different interrupt requests to be sent to the CPU from external devices. The Thunder Board uses interrupt request lines (IRQ's) 2, 3, 5, or 7, depending upon the physical jumper selection on the board. These IRQ's are addressed at I/O addresses 20H and 21H.

I/O address 20H indicates which IRQ line is active (which caused the interrupt). Read bits 0 through 7 of this address to determine the state of IRQ's 0 through 7 respectively. A "1" value in any of the bits indicates an active IRQ. Writing to address 20H acknowledges and clears the active interrupt.

I/O address 21H holds the interrupt mask to the IRQ lines, allowing the CPU to enable or disable any of the eight IRQ's. Write to bits 0 through 7 of this address to enable or disable IRQ's 0 through 7, respectively. A "1" value in a bit disables the corresponding IRQ; a "0" value enables the IRQ.

4 *ESP Command Reference*

This chapter is a reference to the Thunder Board's Enhanced Sound Processor (ESP) commands and command options. You use the ESP to capture and playback digital audio samples.

ESP command overview

You normally program the ESP by sending one byte commands from the CPU to the ESP, and receiving one byte responses from the ESP. Some commands require additional bytes.

The following table is a list of ESP commands. Note that the high nibble of the command is the command type, and that the low nibble is variable.

Bits:	7654	3	2	1	0	Description
	0xH	x	x	x	x	Reserved.
	1xH	x	x	x	x	Set DAC output mode for 8 and 2 bit playback.
	2xH	x	x	x	x	Set ADC input mode for 8 bit recording.
	3xH	x	x	x	x	Reserved.
	4xH	x	x	x	x	Set sample rate and continuous DMA block length.
	5xH	x	x	x	x	Reserved.
	6xH	x	x	x	x	Multi DAC Output.
	7xH	x	x	x	x	Set DAC output mode for 4 and 2.6 bit playback.
	8xH	x	x	x	x	Output silence and record 4 bit ADPCM.
	9xH	x	x	x	x	Reserved.
	AxH	x	x	x	x	Reserved.
	BxH	x	x	x	x	Reserved.
	CxH	x	x	x	x	Reserved.
	DxH	x	x	x	x	Control DMA and speaker.
	ExH	x	x	x	x	Test read/write and board ID.
	FxH	x	x	x	x	Test IRQ and ESP ROM.

Table 4 ESP commands

Set DAC output mode for 8 and 2 bit playback Command 1xH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	DMA transfers	2-Bit Decompress	Reference Byte
0	—	—	8-Bit Data	Direct Mode

Send 8 bit data to the DAC - direct mode (10h)

Transfers non-compressed PCM data directly to the DAC.

Parameters: One byte (8 bit PCM data)

Returns: None

Send 8 bit data to the DAC - automatic DMA mode (14h)

The two bytes following this command sets the block length for the DMA transfer. This value should be set to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 8 bit data to the DAC - enhanced DMA mode (1Ch)

Enables continuous DMA transfers. See the command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Send 2 bit compressed data to the DAC - automatic DMA mode (16h)

The two bytes following this command sets the block length for the automatic DMA transfer. This value should be set to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 2 bit compressed data to the DAC - enhanced DMA mode (1Eh)

Enables continuous DMA transfers. See the command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Send 2 bit compressed data with reference byte to the DAC - automatic DMA mode (17h)

The reference byte is the first byte in the block of data to be transferred. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 2 bit compressed data with reference byte to the DAC - enhanced DMA mode (1Fh)

The reference byte is the first byte in the block of data to be transferred. See the command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Note: On all DMA transfers, remember to program the sample rate and speaker on/off state before initiating the transfer.

Note: See "Programming approaches for PCM data transfers" on page 2-3 for more information on the various DMA transfer modes.

Set ADC input mode for 8 bit recording Command 2xH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	DMA Transfers	—	—
0	—	—	—	Direct Mode

Receive 8 bit PCM sample from the ADC - direct mode (20h)

Transfers a one byte (8 bit) sample from the ADC.

Parameters: None

Returns: One Byte (8 bit PCM sample)

Receive 8 bit PCM samples from the ADC - automatic DMA mode (24h)

The two bytes following this command set the block length for the DMA transfer from the ADC and causes a multiple byte return using automatic DMA mode. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: Multiple bytes

Receive 8 bit PCM samples from the ADC - enhanced DMA mode (2Ch)

Enables continuous DMA transfers. See the command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: ADC sample data stream

Note: On all DMA transfers, remember to program the sample rate and speaker on/off state before initiating the transfer.

Note: See "Programming approaches for PCM data transfers" on page 2-3 for more information on the various DMA transfer modes.

Set sample rate and continuous DMA block length Command 4xH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	—	—	—
0	—	—	—	—

Set sample rate for DMA transfers (40h)

This value indicates the sample rate and is calculated as 256-(1000000/Sample Rate). For example, to calculate the sample rate for 11025khz: 256-(1000000/11025) = 165 (0A5h).

Parameters: One byte

Returns: None

Set block transfer length for enhanced DMA transfers (48h)

Indicates the block length in a 16 bit word. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Set DAC output mode for 4 and 2.6 bit playback Command 7xH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	DMA Mode	2.6 Bit	Reference Byte
0	—	—	4 Bit	—

Send 4 bit compressed data to the DAC - automatic DMA mode (74h)

The two bytes following this command set the block length for the DMA transfer. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 4 bit compressed data to the DAC - enhanced DMA mode (7Ch)

Enables continuous DMA transfers. See command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Send 4 bit compressed data with reference byte to the DAC - automatic DMA mode (75h)

The reference byte is the first byte in the block of data to be transferred. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 4 bit compressed data with reference byte to the DAC - enhanced DMA mode (7Dh)

Reference byte is the first byte in the block of data. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte. See command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Set block length, send 2.6 bit compressed data to the DAC - automatic DMA mode (76h)

Sets the block length for the DMA transfer. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 2.6 bit compressed data to the DAC - enhanced DMA mode (7Eh)

Enables continuous DMA transfers. See command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Send 2.6 bit compressed data with reference byte to the DAC - automatic DMA mode (77h)

Reference byte is the first byte in the block of data. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Send 2.6 bit compressed data with reference byte to the DAC - enhanced DMA mode (7Fh)

Reference byte is the first byte in the block of data. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte. See command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Note: On all DMA transfers, remember to program the sample rate and speaker on/off state before initiating the transfer.

Note: See "Programming approaches for PCM data transfers" on page 2-3 for more information on the various DMA transfer modes.

Output silence and record 4 bit ADPCM Command 8xH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	DMA Mode	—	Reference Byte
0	—	—	—	—

Output silence (80h)

The two bytes following this command set the block length to specify a silence period. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Receive 4 bit compressed data from the ADC - automatic DMA mode (84h)

The two bytes following this command set the block length for the DMA transfer. Set this value to one less that the desired block length. For example, for a block length of 9 bytes, set the block length to 8.

Parameters: Two bytes

Byte 1. The low byte of the length word.

Byte 2. The high byte of the length word.

Returns: None

Receive 4 bit compressed data with reference byte from the DAC - enhanced DMA mode (8Dh)

Reference byte is the first byte in the block of data. See "ADPCM Compression Data Stream Format" on page 2-4 for more information on the reference byte. See command "Set block transfer length for enhanced DMA transfers (48h)" on page 4-5 to set the block length.

Parameters: None

Returns: None

Note: On all DMA transfers, remember to program the sample rate and speaker on/off state before initiating the transfer.

Note: See "Programming approaches for PCM data transfers" on page 2-3 for more information on the various DMA transfer modes.

Control DMA and Speaker Command DxH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Continuous DMA Mode	—	Speaker OFF	—
0	—	—	Speaker ON	—

Halt DMA transfer (D0h)

Pauses the transfer of PCM data using the DMA controller.

Parameters: None

Returns: None

Resume DMA transfer (D4h)

Resumes the transfer of PCM data using DMA.

Parameters: None

Returns: None

Turn speaker on (D1h)

Causes the DAC output to be connected to the speaker, allowing PCM data playback to be heard.

Parameters: None

Returns: None

Turn speaker off (D3h)

Causes the DAC output to be disconnected from the speaker (muted).

Parameters: None

Returns: None

Finish current enhanced DMA transfer (DAh)

Causes the board to finish the current block, then cease transferring. Issue this command while the DMA is transferring the last block of PCM data to the Thunder Board.

Parameters: None

Returns: None

Test read/write and board ID Command ExH

State	Bit #3	Bit #2	Bit #1	Bit #0
1	—	—	—	Return the Version ID
0	—	—	—	Read/Write Test

Read/Write Test (E0h)

This commands sends one byte, then reads back the inverted value. You can use any value.

Parameters: One byte

Returns: One byte

Return version number of the ESP (E1h)

Causes the ESP to return the version number of the board in two bytes, 02h (major byte) and 00h (minor byte).

Issue the command once and the ESP returns the current revision of Sound Blaster's DSP, version 02.00. Issue it a second time (immediately following the first), and the Thunder Board returns its the two byte ID.

Parameters: None

Returns: Two bytes

If this is the first issuance of the command:

1. Sound Blaster's DSP Major version.
2. Sound Blaster's DSP Minor version.

If this is the second issuance IMMEDIATELY following the first issuance:

1. ESP major version.
2. ESP minor version.

Test IRQ and ESP ROM Command FxH

This command is used by Media Vision manufacturing to test IRQ settings and ESP ROM level.

State	Bit #3	Bit #2	Bit #1	Bit #0
1	Return 8 bit checksum	—	Generate an IRQ	—
0	—	—	—	—

Generate an IRQ now (F2h)

Causes the ESP to generate an IRQ immediately and may be used to locate the jumpered IRQ selection on the Thunder Board.

Parameters: None

Returns: None

Return a checksum of the ESP ROM (F8h)

Generates a zero or non-zero value indicating the state of the ROM. A zero value indicates the ROM is OK; an non-zero value indicates the ROM has been corrupted.

The ESP adds the entire contents of ROM into a 16 bit number. This value will be 2's complement, then the low order byte will be returned to the CPU.

Parameters: None

Returns: One byte

5 *FM Synthesizer Programming Basics*

This chapter provides information you should understand before attempting to program the Thunder Board's FM synthesizer. Topics covering in this chapter include:

- Features and capabilities of the Thunder Board's FM synthesizer
- Understanding channel and sound modes
- Understanding FM synthesis modes including *FM synthesis* (serial connection) and *additive synthesis* (parallel connection)

Thunder Board FM synthesizer capabilities

The Thunder Board's FM synthesizer can produce complex musical waveforms while introducing very little PC processor overhead. Unlike PCM waveform generation, in which you must simulate a complex waveform, FM synthesis uses pre-programmed counters and waveforms to generate sound.

The Thunder Board hardware features a Yamaha 3812 FM synthesizer chip to simulate the timbre of musical instruments. Timbre, or sound quality, results from the complexity of sound harmonics. The Thunder Board's frequency modulation technique lets you create rich harmonics and musical sounds by programmatically controlling a few simple parameters.

The key features of the FM synthesizer include:

- A single computational element with 18 storage registers that provides the equivalent of 18 operators
- Programmatic control over individual operators, operator pairs, and all operators at once

- Control over a wide range of parameters for creating sound with rich texture. Parameters you can control include:

  Frequency

  Frequency multiplier

  Envelope type (percussive or non-percussive)

  Envelope amplitude (total level)

  Envelope key scaling level (KSL)

  ADSR (attack/decay/sustain/release) rate

  Key scaling rate (KSR)

  Sustain amplitude level

  Feedback (modulating operator only)

  Tremolo depth and enable

  Vibrato depth and enable

  Waveform selection (sine or non-sine)

FM synthesizer channel modes

The 18 operator cells of the FM synthesizer can be configured in two different ways:

- Nine channel FM mode

  This mode combines the 18 operator cells in pairs to simultaneously generate nine FM voices. Each FM sound uses two operator cells.

- Six channel FM mode (plus five percussion instruments)

  Operator cells 1 through 12 are combined in 6 pairs. Each pair produces one FM voice. Operator cells 13 through 18 produce the sounds of five different percussion instruments.

Operator connection modes

The FM synthesizer operators are paired to create channels, or voices. The Thunder Board FM synthesizer supports two connection modes: FM synthesis (serial connection) and additive synthesis (parallel connection).

FM synthesis (serial connection)

In FM synthesis (serial connection), a sine wave carrier frequency is modulated by a second sine wave signal at the same or a closely related frequency. Both of these frequencies, as well as many of the partial frequencies (harmonics) that are created, are within the audible range. Since it contains many harmonics, the resulting sound spectrum is complex and creates notes that are rich in timbre.

Harmonics are predictable because they appear in accordance with well-known formulae, called Bessel functions. Bessel functions describe the combination of two sine waves by modulation.

The FM synthesis technique requires that two cells be connected in series, as shown in the following figure:

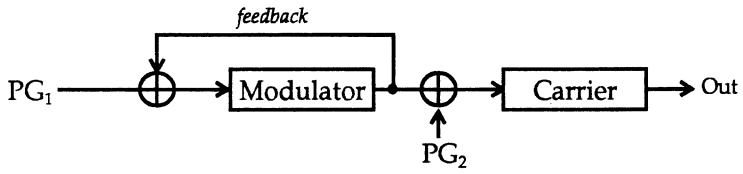


Figure 2 FM Synthesis (Serial Connection)

The output of the carrier cell (which generates the base frequency) is *modulated*, or altered, by the modulator cell. The modulator cell imparts a richness in timbre when it alters the steady frequency produced by the carrier cell.

Each operator cell has three components:

- Phase generator
- Envelope generator
- Sine table

The figure below depicts the characteristics of operator cells.

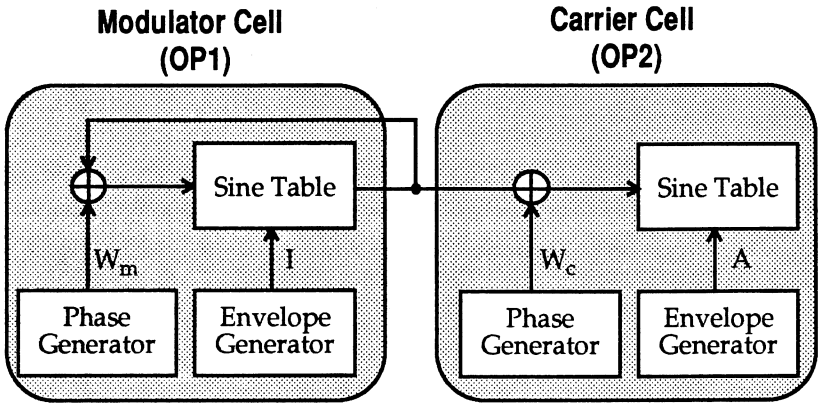


Figure 3 Operator Cell Components

Each operator creates sine waves independently. The frequency of each sine wave is controlled by the phase generator; the amplitude is controlled by the envelope generator.

Operators are combined in series. The output of the modulator cell alters the carrier cell. The waveform that results from the output of the carrier cell contains the fundamental frequency of the carrier cell, plus harmonics that result from the interaction of the two cells. The harmonics are at frequencies equal to the carrier frequency, plus and minus integer multiples of the modulator frequency.

The relative strength of the harmonics depends on the amplitude of the modulator cell output. By changing the frequency and amplitude of the modulator cell, while keeping the frequency of the carrier cell constant, you can dramatically change the timbre of the FM synthesized sound created by these two operators.

The formula below explains the interaction of the four parameters, which you control programmatically, that control FM sound production:

$$F(t) = A \sin(w_c t + I \sin(w_m t))$$

A	output amplitude
I	modulation index (modulator cell amplitude)
w_c	carrier cell frequency
w_m	modulator cell frequency

Additive synthesis (parallel connection)

With additive synthesis (parallel connection), the output equals the sum of the two operators. Operator 1 optionally has feedback, which is useful for creating interesting harmonics. The following figure illustrates the interaction of cells operating in parallel.

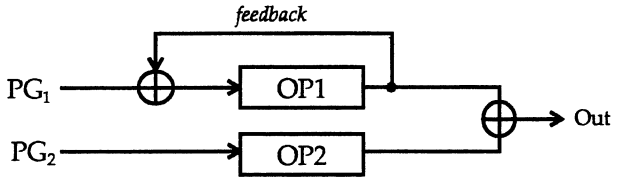


Figure 4 Additive Synthesis

The formula below describes composite sine wave synthesis:

$$F(t) = A_1 \sin(w_1 t) + A_2 \sin(w_2 t)$$

A_1 cell 1 amplitude
w_1 cell 1 frequency
A_2 cell 2 amplitude
w_2 cell 2 frequency

6 *Programming the FM Synthesizer*

This chapter describes the hardware programming interface of the Thunder Board's FM synthesizer and outlines four important concepts involved in programming the synthesizer:

- Reading and writing to status registers
- Understanding operator and cell number relationships
- Using timers
- Understanding the strategy for producing synthesized music

FM synthesizer hardware programming interface

The FM synthesizer application programming interface implements the concepts of I/O ports and indices. Indices are the actual FM synthesizer hardware registers you program; the I/O ports are the mechanisms though which you access them. The following figure depicts the relationship between I/O ports and indices.

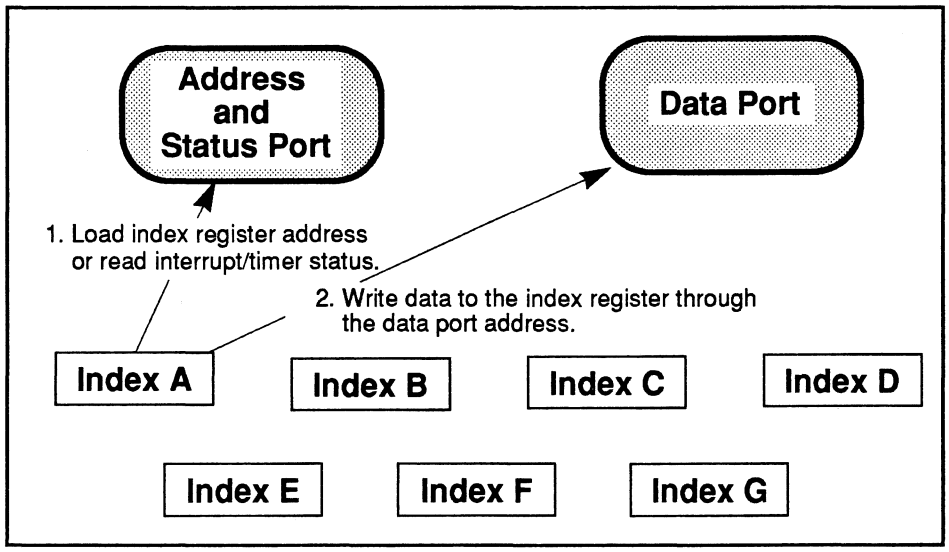


Figure 5 FM synthesizer API structure

The following table describes the channel I/O addresses used to program the FM synthesizer.

Channel	I/O address(es)	Description
	388H	*FM synthesizer address and status* port. Used for two purposes: To select the register address and to read the FM synthesizer status register. Preferred interface supported by AdLib and Yamaha.
	389H	*FM synthesizer data* port. Used to write data to the selected register. Preferred interface supported by AdLib and Yamaha.
	2x8H	*FM synthesizer address and status* port. Alternative address for Sound Blaster compatibility.
	2x9H	*FM synthesizer data* port. Alternative address for Sound Blaster compatibility.

Reading and writing to FM synthesizer ports

The bits of the FM synthesizer address and status port contain different types of data depending upon whether the port is in read or write mode.

In read mode, the FM synthesizer address and status port contains the following data:

D7	D6	D5	D4	D3	D2	D1	D0
Interrupt Request	Timer 1 Flag	Timer 2 Flag	Not Used	Not Used	Not Used	Not Used	Not Used

In read mode, use this port to determine the current state of the FM synthesizer Interrupt line and Timers. When either Timer overflows, the corresponding Timer Flag (D5 or D6) and the Interrupt Request (D7) are set to 1. For more information on handling interrupts, see "Handling interrupts generated by the ESP" on page 3-3.

In write mode, the FM synthesizer address and status port accepts the following data:

D7	D6	D5	D4	D3	D2	D1	D0
FM Address Select	FM Address Select	FM Address Select	FM Address Select	FM Address Select	FM Address Select	FM Address Select	FM Address Select

In write mode, use this port to specify FM registers to load when your program writes to the FM synthesizer data port.

The FM synthesizer data port is used to load (write) sound generation parameters into the FM synthesizer. The table below shows a bit map of FM synthesizer data port.

D7	D6	D5	D4	D3	D2	D1	D0
FM Register Data	FM Register Data	FM Register Data	FM Register Data	FM Register Data	FM Register Data	FM Register Data	FM Register Data

Use this port to load FM registers with register values. For more information on FM registers and register values, see Chapter 7, "FM Synthesizer Register Functions."

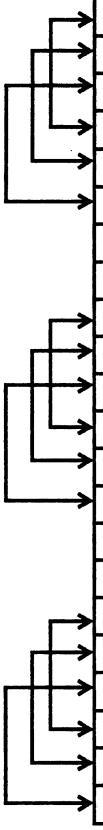
Understanding operator cell number and channel number

The FM synthesizer can be configured for up to nine FM channels (also known as voices). Each channel is produced by combining a pair of operator cells.

When the FM synthesizer is set to melody mode, all 18 operators are dedicated to the nine FM channels. To set the FM synthesizer to melody mode, set CSM Mode/Keyboard Split Register (08h) bit D7 to and the Depth/Percussion/ Instruments Register (0BDh) bit D5 to 0.

In percussion mode, operators 1 through 12 are paired to create six FM channels; operators 13 through 18 produce five percussion sounds, including bass drum and high hat. For more information on enabling percussion mode and percussion instruments, see "Depth / Percussion / Instruments Register BDh" on page 7-22.

The table below shows how operator cells are paired to create channels when the synthesizer is in melody mode. Remember, in percussion mode, operator cells 13 through 18 are reserved for percussion instruments.



Operator	Channel	Offset	Function
1	1	00	modulator
2	2	01	modulator
3	3	02	modulator
4	1	03	carrier
5	2	04	carrier
6	3	05	carrier
--	--	06	scratch register
--	--	07	scratch register
7	4	08	modulator
8	5	09	modulator
9	6	0A	modulator
10	4	0B	carrier
11	5	0C	carrier
12	6	0D	carrier
--	--	0E	scratch register
--	--	0F	scratch register
13	7	10	modulator*
14	8	11	modulator*
15	9	12	modulator*
16	7	13	carrier*
17	8	14	carrier*
18	9	15	carrier*

Figure 6 Operator Cell Pairings To Create Channels

* Used for either FM synthesis (in melody mode) or percussion sound production (in percussion mode)

Note: The Key On bits (D5) of Block and F-Number Registers B6h, B7h, and B8h must be set to 0 before enabling percussion instruments.

Using timers

The Thunder Board's FM synthesizer provides two timers, Timer 1 and Timer 2, which provide 80 and 320 microseconds resolution respectively.

You control timers by loading an initial count value, activating the timer, counting and counting to 256.

When 256 is reached:

- The Timer 1 status flag in the FM synthesizer Test and Status Register (01h) is set to 1
- All operator cells are set to Key On (sound enabled) and then immediately set to Key Off
- The timer is reloaded with the register value and counting begins again

Follow this procedure in order to use timers:

1. **Write an initial count value (value n) to a Timer register (02h or 03h).**
2. **Set the appropriate Start and Stop Timer bit (D1 or D0 of 04h) to start the Timer.**

   When the timer starts, the Timer register value is loaded into the timer and counting begins (from initial count upwards).

For more information on Timers, see "Timer 1 Register 02h" on page 7-2 and "Timer 2 Register 03h" on page 7-2. For more information on the Key On function, see "AM/VIB/EG/KSR/MULTIPLE Registers 20h to 35h" on page 7-7.

Programming strategy

The procedure below integrates the concepts presented in this chapter and Chapter 5, "FM Synthesizer Programming Basics." Follow these steps to produce computer synthesized music:

1. **Identify the channel and operators you want to program.**
2. **Initialize the synthesizer.**

   Select FM synthesis mode (FM synthesis or additive synthesis) and melody or percussion mode.
3. **Program each operator with the appropriate ADSR curves and miscellaneous modifier bits.**
4. **Determine the note you want to produce.**

   Load the frequency and octave information into the appropriate registers.
5. **Start playing notes.**

   Toggle the *Key On* bit for the channel and operators you've chosen.
6. **Wait until all notes have been played and start again.**

Note: If you follow the procedure to produced a note, but don't produce the one you want, check the ADSR setting, frequency, octave scale, connection bit, frequency multiplier, and CSM mode/keyboard split, in that order.

7 FM Synthesizer Register Functions

This chapter describes the Thunder Board's FM synthesizer registers and how to program register functions. Before beginning to use the register functions described here, be sure to read Chapter 6, "Programming the FM Synthesizer."

Test Register 01h

The Test register enables and disables the Wave select command and is also used by Yamaha to test the FM synthesizer chip. Aside from D5, all other bits of this register are reserved by Yamaha for testing and should always be set to 0.

D7	D6	D5	D4	D3	D2	D1	D0
Test	Test	Enable Wave Select	Test	Test	Test	Test	Test

Test (D7, D6 and D4 to D0)

Reserved. These bits should always be set to 0.

	Dn	Description
Settings:	1	Reserved, do not use.
	0	Clear test bits.
Default:	0	

Enable Wave Select (D5)

Set this bit to 1 to turn the Wave Select command (registers E0h to F5h) on; set D5 to 0 turn Wave Select off.

	D5	Description
Settings:	1	Turn Wave Select on.
	0	Turn Wave Select off.
Default:	0	

Timer 1 Register 02h

Timer 1 is a general purpose timer that can also be used to control composite speech synthesis. It is an 8-bit, presettable counter with a resolution of 80 microseconds. For more information on using timers, see "Using timers" on page 6-39.

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

To activate the timer, load it with an initial count value. The Timer 1 count value is calculated as follows:

$$T1_{overflow} = (256-N) * 80 \text{ microseconds}$$

Where:

N Timer 1 count values

	Dn	Description
Settings:	1	Set bits to indicate Timer 1 value.
	0	Set bits to indicate Timer 1 value.
Default:	N/A	

Timer 2 Register 03h

Timer 2 is identical to Timer 1 except that its resolution is 320 microseconds, or four times that of Timer 1.

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

To activate the timer, load it with an initial count value. The Timer 2 count value is calculated as follows:

$$T2_{overflow} = (256-N) * 320 \text{ microseconds}$$

Where:

N Timer 2 count values

	Dn	Description
Settings:	1	Set bits to indicate Timer 2 value.
	0	Set bits to indicate Timer 2 value.
Default:	N/A	

Control Timer Register 04h

The Control Timer register resets IRQ status flags; resets and masks timer status flags; and starts and stops Timer 1 and Timer 2.

D7	D6	D5	D4	D3	D2	D1	D0
IRQ Reset	Mask T1	Mask T2	--	--	--	ST2	ST1

Reset IRQ (D7)

Set this bit to 1 to clear status register flags for IRQ, Timer 1, and Timer 2. This bit automatically resets itself to 0 after flags are cleared.

	D7	Description
Settings:	1	Clear status register flags.
	0	Inactive.
Default:	0	

Note: See "Using timers" on page 6-39 for an explanation of how the Thunder Board responds to the FM synthesizer IRQ and Timer status flags.

Mask Timer 1 (D6)

Set this bit to 1 to mask the Timer 1 status flag, preventing it from being set when Timer 1 overflows. Because the Thunder Board has its own interrupt enable bits for FM synthesis, using this mask is not recommended.

	D6	Description
Settings:	1	Mask Timer 1 flag.
	0	Inactive.
Default:	0	

Mask Timer 2 (D5)

Set this bit to 1 to mask the Timer 2 status flag, preventing it from being set when Timer 2 overflows.

	D5	Description
Settings:	1	Mask Timer 2 flag.
	0	Inactive.
Default:	0	

Start or Stop Timer 2 (D1)

Set this bit to 1 to cause the count value to be loaded into Timer 2 and begin counting. Set to 0 to stop and reset Timer 2.

	D1	Description
Settings:	1	Load register value into Timer 2 and begin counting.
	0	Stop and reset Timer 2.
Default:	0	

Start or Stop Timer 1 (D0)

Set this bit to 1 to cause the count value to be loaded into Timer 1and begin counting. Set to 0 to stop and reset Timer 1.

	D0	Description
Settings:	1	Load register value into Timer 1 and begin counting.
	0	Stop and reset Timer 1.
Default:	0	

CSM Mode / Keyboard Split Register 08h

This register controls whether the synthesizer operates in music mode or composite speech mode (CSM), as well as the location of the keyboard split for keyboard rate scaling.

D7	D6	D5	D4	D3	D2	D1	D0
CSM	Keyboard Split (SEL)						

Composite Sine Wave Mode (D7)

Set this bit to 1 to turn composite sine wave mode on or to 0 to turn melody mode on.

In music mode, the FM synthesizer can be programmed to operate in one of two sub-modes: melody mode (9 FM channels) or percussion mode (6 FM channels and 5 percussion sounds). Melody mode is the default configuration.

All 18 operators must be turned off (Key On = 0) before you switch to the composite speech mode. You produce composite speech sound by momentarily switching to Key On.

	D7	Description
Settings:	1	Set synthesizer to composite speech mode.
	0	Set synthesizer to music mode.
Default:	0	

Note: There is no functional use for composite speech mode. Yamaha has chosen not to support this function in later revisions of the FM synthesizer chip.

Keyboard Split Point or SEL (D6)

Set this bit to 1 to set the keyboard split point.

In the context of the FM synthesizer, keyboard split refers to the point where keyboard scaling occurs within each octave.

The *Split Number* controls the amount of key scaling applied to a note. As shown in the following table, the Split Number is a function of both the octave (block number) and the split point within each octave. For both split point settings (SEL = 0 or 1), key scaling increases by a factor of two as the octave (block number) increases. An exception to this rule occurs at the split point within an octave. Beyond the split point, key scaling increases by 1.

Octave	0		1		2		3		4		5		6		7	
Block Data	0		1		2		3		4		5		6		7	
F-Num MSB	1		1		1		1		1		1		1		1	
F-Num 2nd	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Split Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Octave	0		1		2		3		4		5		6		7	
Block Data	0		1		2		3		4		5		6		7	
F-Num MSB	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
F-Num 2nd	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Split Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

For more information on key scaling, see "AM/VIB/EG/KSR/MULTIPLE Registers 20h to 35h" on page 7-7.

For both SEL bit settings, the keyboard split point occurs in the middle of each octave. Note that the most significant bit changes from 0 to 1 in the middle of the octave.

When set to 1, keyboard scaling increases smoothly (in jumps of 2) from octave to octave, and increases by 1 within each octave. Keyboard scaling increases in a roughly linear fashion across the entire keyboard. When set to 0, notes of the lower half of the keyboard have no scaling while notes of the upper half scale smoothly (in jumps of 2).

Since the split number influences the note's attack/decay/release rate, SEL=1 results in a natural sounding scaling effect while SEL=0 scaling provides a special effect.

	D6	Description
Settings:	1	Scale keyboard by jumps of 2 from octave to octave.
	0	Do not scale lower half of keyboard, scale by jumps of 2 for upper half.
Default:	0	

AM/VIB/EG/KSR/MULTIPLE Registers 20h to 35h

This register group controls operator characteristics that produce changes in timbre. One register is dedicated to each of the operators.

D7	D6	D5	D4	D3	D2	D1	D0
Tremolo (AM)	Vibrato	Envelope (EG) Type	Key Scaling Rate (KSR)	Multiple Bit 3	Multiple Bit 2	Multiple Bit 1	Multiple Bit 0

Tremolo Modulation (D7)

Set this bit to 1 to apply amplitude modulation (tremolo) to an operator.

The tremolo frequency is 3.7 Hz and the modulation depth is either 7% or 14%, depending on the setting of the Tremolo Depth (bit D7) in register BDh.

	D7	Description
Settings:	1	Apply amplitude modulation.
	0	No amplitude modulation.
Default:	0	

Vibrato Modulation (D6)

Set this bit to 1 to apply vibrato modulation to an operator.

The vibrato is a frequency of 6.4 Hz (twice that of the tremolo frequency) and the modulation depth is either 4.8 dB or 1 dB, depending on the setting of the Vibrato Depth (bit D6) in register at BDh.

	D6	Description
Settings:	1	Apply vibrato modulation.
	0	No vibrato modulation.
Default:	0	

Envelope Type (D5)

Set this bit to 1 to select a continuing sound envelope shape or to 1 to select a diminishing sound envelope shape.

Diminishing sounds are sometimes referred to as *percussive* sounds while continuing sounds are called *non-percussive*. The figure below shows the two envelope types.

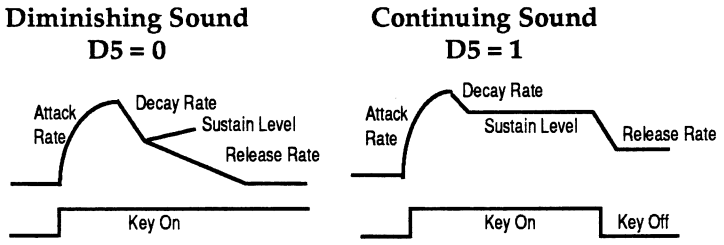


Figure 7 Envelope Types

	D5	Description
Settings:	1	Set envelope shape to continuing sound.
	0	Set envelope shape to diminishing sound.
Default:	0	

Key Scaling Rate (D4)

Set this bit to 1 to set the key scaling rate to maximum scaling and to 0 to set it to minimum scaling. You control the rate of key scaling in order to imitate string instruments.

To the human ear, notes of string instruments appear to shorten at higher frequencies. Key scaling lets you set the rate of attack, decay, and release of a note. A higher (more vertical) rate of attack, decay, and release results in a gradual shortening of the envelope length as higher frequency notes are played. The figure below illustrates this phenomenon.

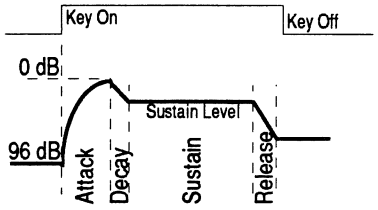


Figure 8 Envelope Waveform

Using values from the table below, use this formula to calculate the actual rate:

```
Actual_rate = 4 * Unscaled_rate + KSR_adjustment
```

Where:

Unscaled_rate	Unscaled attack/decay/release rate (Keyboard Split Number).
KSR_adjustment	KSR adjustment value from table.

Unscaled_rate (keyboard split #)	KSR_adjustment	
	Minimal D4=0	Maximal D4=1
0	0	0
1	0	1
2	0	2
3	0	3
4	1	4
5	1	5
6	1	6
7	1	7
8	2	8
9	2	9
10	2	10
11	2	11
12	3	12
13	3	13
14	3	14
15	3	15

Table 5 Key Scaling Rate/Keyboard Split Number Cross Reference

	D4	Description
Settings:	1	Set key scaling rate to maximum.
	0	Set key scaling rate to minimum.
Default:	0	

Multiple Bits or MUL (D3 - D0)

Set these four bits in order to switch from the fundamental frequency of a note to a nearby harmonic frequency.

To understand the effect of the Multiple bits, the following formula shows FM synthesis with the multiplication factor explicitly included:

$$F(t) = A \sin (m_c w_c t + I \sin (m_m w_m t))$$

Where:

A	output amplitude
m_c	multiplication factor for carrier
w_c	frequency for carrier cell
m_m	multiplication factor for modulator
w_m	frequency for modulator cell

	D3	D2	D1	D0	Description
Settings:	1	1	1	1	Set multiplication factor to 15.
	1	1	1	0	Set multiplication factor to 15.
	1	1	0	1	Set multiplication factor to 12.
	1	1	0	0	Set multiplication factor to 12.
	1	0	1	1	Set multiplication factor to 10.
	1	0	1	0	Set multiplication factor to 10.
	1	0	0	1	Set multiplication factor to 9.
	1	0	0	0	Set multiplication factor to 8.
	0	1	1	1	Set multiplication factor to 7.
	0	1	1	0	Set multiplication factor to 6.
	0	1	0	1	Set multiplication factor to 5.
	0	1	0	0	Set multiplication factor to 4.
	0	0	1	1	Set multiplication factor to 3.
	0	0	1	0	Set multiplication factor to 2.
	0	0	0	1	Set multiplication factor to 1.
	0	0	0	0	Set multiplication factor to .5.
Default:	0	0	0	0	

KSL / Total Level Registers 40h to 55h

The KSL/Total Level registers control he operator envelope level (amplitude). By changing the operator envelope of the carrier operator, you control the output level (strength) of a note. By changing the modulator operator envelope level, relative to the carrier operator, you control the harmonic richness (timbre) of the sound. One register is dedicated to each of the operators.

D7	D6	D5	D4	D3	D2	D1	D0
KSL Bit 1	KSL Bit 0	Total Level Bit 5	Total Level Bit 4	Total Level Bit 3	Total Level Bit 2	Total Level Bit 1	Total Level Bit 0

Key Scaling Level (D7 and D6)

Set these two bits to specify the rate at which the output amplitude decreases from the starting level as pitch increases. Use Key Scaling Level to simulate the behavior of acoustic instruments.

	D7	D6	Description
Settings:	1	1	Set degree of attenuation to 6 dB/octave.
	0	1	Set degree of attenuation to 3 dB/octave.
	1	0	Set degree of attenuation to 1.5 dB/octave.
	0	0	Set degree of attenuation to 0 dB/octave.
Default:	0	0	Set degree of attenuation to 0 dB/octave.

Total Level (D5 to D0)

Set these six bits to specify operator strength. You can vary output amplitude from full strength to 47.25 dB attenuation (reduction). The following table illustrates the amount of attenuation controlled by each bit.

	D5	D4	D3	D2	D1	D0
Degree of Attenuation	24 dB	12 dB	6 dB	3 dB	1.5 dB	.75 dB

To determine the total amount of attenuation, add up the decibel values for each of the bits set to 1.

The carrier cell output level governs how dominant this channel is relative to other FM channels produced by the FM synthesizer.Varying the modulator cell output level changes the amount of modulation. The greater the modulation, the greater the number and strength of harmonics. Strong harmonics result in rich timbre.

	Dn	Description
Settings:	1	Turn attenuation on for Bit (n).
	0	Turn attenuation on for Bit (n).
Default:	0	

Attack / Decay Rate Registers 60h to 75h

This set of registers sets the rising and decaying times for a sound. One register is dedicated to each of the operators.

D7	D6	D5	D4	D3	D2	D1	D0
Attack Rate Bit 3	Attack Rate Bit 2	Attack Rate Bit 1	Attack Rate Bit 0	Decay Rate Bit 3	Decay Rate Bit 2	Decay Rate Bit 1	Decay Rate Bit 0

Attack Rate (D7 to D4)

Set these four bits to specify the attack rate value for the sound. To determine the actual rate of attack, see Appendix B, "FM Rate to Time Conversion Tables."

	D7	D6	D5	D4	Description
Settings:	1	1	1	1	Set attack rate value to 15.
	1	1	1	0	Set attack rate value to 14.
	1	1	0	1	Set attack rate value to 13.
	1	1	0	0	Set attack rate value to 12.
	1	0	1	1	Set attack rate value to 11.
	1	0	1	0	Set attack rate value to 10.
	1	0	0	1	Set attack rate value to 9.
	1	0	0	0	Set attack rate value to 8.
	0	1	1	1	Set attack rate value to 7.
	0	1	1	0	Set attack rate value to 6.
	0	1	0	1	Set attack rate value to 5.
	0	1	0	0	Set attack rate value to 4.
	0	0	1	1	Set attack rate value to 3.
	0	0	1	0	Set attack rate value to 2.
	0	0	0	1	Set attack rate value to 1.
	0	0	0	0	Set attack rate value to 0.
Default:	0	0	0	0	

Decay Rate (D3 to D0)

Set these four bits to specify the decay rate value for the sound. To determine the actual rate of decay, see Appendix B, "FM Rate to Time Conversion Tables."

Note: The Decay Rate time is the same as the Release Rate time for the same rate value.

	D3	D2	D1	D0	Description
Settings:	1	1	1	1	Set decay rate value to 15.
	1	1	1	0	Set decay rate value to 14.
	1	1	0	1	Set decay rate value to 13.
	1	1	0	0	Set decay rate value to 12.
	1	0	1	1	Set decay rate value to 11.
	1	0	1	0	Set decay rate value to 10.
	1	0	0	1	Set decay rate value to 9.
	1	0	0	0	Set decay rate value to 8.
	0	1	1	1	Set decay rate value to 7.
	0	1	1	0	Set decay rate value to 6.
	0	1	0	1	Set decay rate value to 5.
	0	1	0	0	Set decay rate value to 4.
	0	0	1	1	Set decay rate value to 3.
	0	0	1	0	Set decay rate value to 2.
	0	0	0	1	Set decay rate value to 1.
	0	0	0	0	Set decay rate value to 0.
Default:	0	0	0	0	

Sustain Level / Release Rate Registers 80h to 95h

This set of registers sets the sustain level for a continuing sound and the release rate for both continuing and diminishing sounds. One register is dedicated to each of the operators.

D7	D6	D5	D4	D3	D2	D1	D0
Sustain Level Bit 3	Sustain Level Bit 2	Sustain Level Bit 1	Sustain Level Bit 0	Release Rate Bit 3	Release Rate Bit 2	Release Rate Bit 1	Release Rate Bit 0

Sustain Level (D7 to D4)

Set the Sustain Level bits to specify the amplitude of a continuing sound prior to decay. You specify Sustain Level relative to peak amplitude of the note. The valid range of settings varies between 0 dB (same as the peak) and 46 dB attenuation (almost a vertical drop-off from the peak). The following table illustrates the sustaining decibel level controlled by each bit.

	D7	D6	D5	D4
Degree of Attenuation	24 dB	12 dB	6 dB	3 dB

To determine the total sustain level, add up the decibel values for each of the bits set to 1.

	Dn	Description
Settings:	1	Turn sustain level on for Bit(n).
	0	Turn sustain level on for Bit(n).
Default:	0	

Release Rate (D3 to D0)

Set these four bits to specify the the speed at which a sound fades away. You can apply Release Rate to both continuing and diminishing sounds.

For continuing sounds, the Release Rate governs the speed at which sound fades away after the transition from Key On (D5 0f Axh =1) to Key Off (D5 = 0). For diminishing sounds, the Release Rate determines the rate at which sound fades away after the Sustain Level signal level is reached. To see understand the relationship between Release Rate values and actual release times, see Appendix B, "FM Rate to Time Conversion Tables."

Note: The Release Rate time is the same as the Decay Rate time for the same rate value.

	D3	D2	D1	D0	Description
Settings:	1	1	1	1	Set release rate value to 15.
	1	1	1	0	Set release rate value to 14.
	1	1	0	1	Set release rate value to 13.
	1	1	0	0	Set release rate value to 12.
	1	0	1	1	Set release rate value to 11.
	1	0	1	0	Set release rate value to 10.
	1	0	0	1	Set release rate value to 9.
	1	0	0	0	Set release rate value to 8.
	0	1	1	1	Set release rate value to 7.
	0	1	1	0	Set release rate value to 6.
	0	1	0	1	Set release rate value to 5.
	0	1	0	0	Set release rate value to 4.
	0	0	1	1	Set release rate value to 3.
	0	0	1	0	Set release rate value to 2.
	0	0	0	1	Set release rate value to 1.
	0	0	0	0	Set release rate value to 0.
Default:	0	0	0	0	

Block and F-Number Registers A0h to A8h to B0h to B8h

Together, these two groups of 9 registers select the octave, and note within the octave, associated with the two operator cells that are paired in FM synthesis to form a channel. The Key On bit of register group B0h to B8h controls when notes are played.

The Block Number specifies the octave for a pair of operator cells. The 10-bit F-Number, which spans Ax and Bx registers, specifies the note within the octave.

The 10 bit F-Number is formed by combining bits D1 and D0 from the B0h to B8h register group with its matching register in the A0h to A8h group. F-Numbers are configured as follows

- F-Number (8 lower bits) (D7 to D0 bits in Axh)
- F-Number (2 high bits) (D1 and D0 bits in Bxh)

Register Group A0h to A8h

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

Register Group B0h to B8h

D7	D6	D5	D4	D3	D2	D1	D0
--	--	Key On	Block Num Bit 2	Block Num Bit 1	Block Num Bit 0	F-Num Bit 9	F-Num Bit 8

F-Num (D7 to D0 of register Axh plus D1 and D0 of register Bxh)

Set these ten bits to specify the frequency number for a note within an octave. The F-Num for a given note is the same for all octaves.

Ten bit F-Nums map into 1,024 frequency values. Many of these values extend outside an octave or specify frequencies that fall between commonly accepted pitch values. Consequently, only a relatively small number of the F-Num values make sense for a given octave. The following table shows the F-Num for the fourth octave in both decimal and in binary:

	Freq. Octave 4	F-Number										
		Decimal Value	High		Low							
			D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
C#	277.2	363	0	1	0	1	1	0	1	0	1	1
D	293.7	385	0	1	1	0	0	0	0	0	0	1
D#	311.1	408	0	1	1	0	0	1	1	0	0	0
E	329.6	432	0	1	1	0	1	1	0	0	0	0
F	349.2	458	0	1	1	1	0	0	1	0	1	0
F#	370	485	0	1	1	1	1	0	0	1	0	1
G	392	514	1	0	0	0	0	0	0	0	1	0
G#	415.3	544	1	0	0	0	1	0	0	0	0	0
A	440	577	1	0	0	1	0	0	0	0	0	1
A#	466.2	611	1	0	0	1	1	0	0	0	1	1
B	493.9	647	1	0	1	0	0	0	0	1	1	1
C	523.3	686	1	0	1	0	1	0	1	1	1	0

Table 6 F-Numbers For Octave 4

You can multiply the operator frequency by the Multiple bits found in register group 20h through 35h to yield a set of frequencies different than those specified by the Block Number and F-Num alone. Note that while the Block Number and F-Num applies to both operators in a pair, the Multiple can be applied to the modulator, the carrier, or to both the modulator and carrier.

For a complete list of standard pitch values for all notes within the eight octave range of the FM synthesizer, see Appendix C, "FM Key Scaling Level Tables."

	D1	D2	D7	D6	D5	D4	D3	D2	D1	D0	Description
Settings:	0	1	0	1	1	0	1	0	1	1	Set F-Num to decimal 363 (C#).
	0	1	1	0	0	0	0	0	0	1	Set F-Num to decimal 385 (D).
	0	1	1	0	0	1	1	0	0	0	Set F-Num to decimal 408 (D#).
	0	1	1	0	1	1	0	0	0	0	Set F-Num to decimal 432 (E).
	0	1	1	1	0	0	1	0	1	0	Set F-Num to decimal 458 (F).
	0	1	1	1	1	0	0	1	0	1	Set F-Num to decimal 485 (F#).
	1	0	0	0	0	0	0	0	1	0	Set F-Num to decimal 514 (G).
	1	0	0	0	1	0	0	0	0	0	Set F-Num to decimal 544 (G#).
	1	0	0	1	0	0	0	0	0	1	Set F-Num to decimal 577 (A).
	1	0	0	1	1	0	0	0	1	1	Set F-Num to decimal 611 (A#).
	1	0	1	0	0	0	0	1	1	1	Set F-Num to decimal 647 (B).
	1	0	1	0	1	0	1	1	1	0	Set F-Num to decimal 686 (C).
Default:	0	0	0	0	0	0	0	0	0	0	

Key On (D5 of register Bxh)

Set this bit to control output of a given operator pair. There is one Key On bit for each of the nine operator pairs (in FM synthesis melody mode). For a discussion of operator cell pairs, see "Understanding operator cell number and channel number" on page 6-3.

Note: Key On bits 13 through 18 must be cleared before you switch from melody mode to percussion mode.

	D5	Description
Settings:	1	Turn channel on for operator pair (n).
	0	Turn channel off for operator pair (n).
Default:	0	

Block Number (D4 through D2 of register Bxh)

Set these three bits to specify the octave you are programming. The FM synthesizer supports an eight octave range.

	D4	D3	D2	Description
Settings:	1	1	1	Set to octave 8.
	1	1	0	Set to octave 7.
	1	0	1	Set to octave 6.
	1	0	0	Set to octave 5.
	0	1	1	Set to octave 4.
	0	1	0	Set to octave 3.
	0	0	1	Set to octave 2.
	0	0	0	Set to octave 1.
Default:	0	0	0	

Depth / Percussion / Instruments Register BDh

This register group controls tremolo modulation and vibrato depth for melody instruments; toggles the FM synthesizer between melody and percussion mode; and enables percussion instruments of the FM synthesizer.

D7	D6	D5	D4	D3	D2	D1	D0
Tremolo Depth	Vibrato Depth	Percussion Mode Enable	Bass Drum Enable	Snare Drum Enable	Tom Tom Enable	Top Cymbal Enable	Hi-Hat Enable

Tremolo Depth (D7)

Set this bit to control the amount of tremolo modulation (changes to amplitude) for an operator pair. Tremolo must be enabled (D7 of register group 20h through 35h) for this setting to take effect.

	D7	Description
Settings:	1	Set tremolo modulation to 4.8 decibels.
	0	Set tremolo modulation to 1 decibels.
Default:	0	

Vibrato Depth (D6)

Set this bit to control the amount of vibrato modulation (changes to pitch) for an operator pair. Vibrato must be enabled (D6 or register group 20h through 35h) for this setting to take effect.

	D6	Description
Settings:	1	Set vibrato modulation to 14 percent.
	0	Set vibrato modulation to 7 percent.
Default:	0	

Percussion Mode Enable (D5)

Set this bit to 1 to turn percussion mode on or to 0 to turn melody mode on. You must set this bit on in order to enable individual percussion instruments.

	D5	Description
Settings:	1	Set FM synthesizer to percussion mode.
	0	Set FM synthesizer to melody mode.
Default:	0	

Percussion Instrument Enable (D4 through D0)

Set these five bits to enable the bass drum, snare drum, Tom Tom, Top cymbal, and Hi-Hat cymbals. The essential characteristics of the percussion instruments are pre-programmed into the FM synthesizer, but you must still program other characteristics, such as the F-Number and Attack Rate, to control the frequency and timbre. The Percussion Mode Enable bit (D5), must be set to 1 in order to activate the individual percussion instruments.

The table below shows FM synthesizer percussion instruments, the particular bits of register BDh that control them, and their associated operator cell number.

Percussion Instrument	Control Bit	Operator Cell Number(s)
Bass Drum	D4	13 and 16
Snare Drum	D3	17
Tom Tom	D2	15
Top Cymbal	D1	18
Hi-Hat Cymbal	D0	14

Before enabling a percussion instrument, you must turn off the corresponding Key On bits (D5 = 0) in registers B6h through B8h. These three registers control channels 7 through 9 (operators 13 through 18), which are now reserved for percussion instrument sounds.

	Dn	Description
Settings:	1	Enable percussion instrument.
	0	Disable percussion instrument.
Default:	0	

Feedback / Connection Registers C0h - C8h

This group of registers determines the amount of operator cell feedback and whether pairs of operators are connected serially (FM synthesis) or in parallel (additive synthesis).

D7	D6	D5	D4	D3	D2	D1	D0
⋮	⋮	⋮	⋮	Feedback Bit 2	Feedback Bit 1	Feedback Bit 0	Connection

Feedback (D3 to D1)

Set these four bits to specify the level of feedback for the first operator cell in a channel. The FM synthesizer supports eight fixed amplitude settings.

When an operator pair is connected serially (FM synthesis), feedback applies to the *modulator* operator; when connected in parallel (additive synthesis), feedback applies to one operator only. For more information on connection modes, see "Operator connection modes" on page 5-2.

	D3	D2	D1	Description
Settings:	1	1	1	Set feedback to 4π.
	1	1	0	Set feedback to 2π.
	1	0	1	Set feedback to π.
	1	0	0	Set feedback to π/2.
	0	1	1	Set feedback to π/4.
	0	1	0	Set feedback to π/8.
	0	0	1	Set feedback to π/16.
	0	0	0	Set feedback to 0.
Default:	0	0	0	

Connection (D0)

Set this bit to specify how pairs of operator cells are connected. The 18 operator cells of the FM synthesizer are organized as nine pairs of operator cells.You can connect these serially (for FM synthesis) or in parallel (for additive synthesis).

The following figures illustrate the the differences between FM synthesis and additive synthesis.

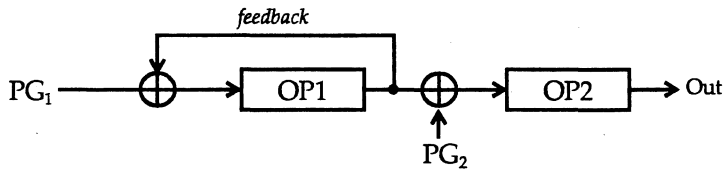


Figure 9 FM Synthesis (Serial) Connection

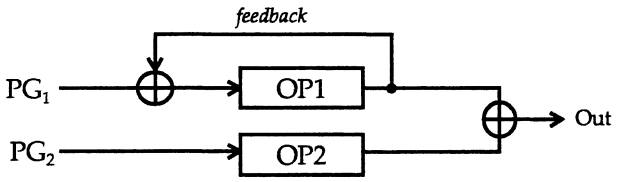


Figure 10 Additive Synthesis (Parallel) Connection

	D0	Description
Settings:	1	Set connection to additive synthesis (parallel).
	0	Set connection to FM synthesis (serial).
Default:	0	

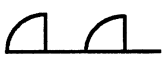
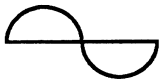
Waveform Registers E0h to F5h

This group of 18 registers is used to select the waveform generated by the operator cells.

D7	D6	D5	D4	D3	D2	D1	D0
:	:	:	:	:	:	Wave Select Bit 1	Wave Select Bit 0

Wave Select (D1 to D0)

Set these two bits to control the shape of the waveform generated by operator cells.

	D1	D0	Description	
Settings:	1	1	Set waveform to:	
	1	0	Set waveform to:	
	0	1	Set waveform to:	
	0	0	Set waveform to:	
Default	0	0		

A *Examples of Programming the Thunder Board for DMA Transfers*

The most common approach to playing or recording PCM data is to use an *automatic* mode of data transfer. This approach provides two major benefits:

- The CPU may perform other tasks during the DMA transfer
- Lengthy recordings may be broken down into more manageable chunks

The typical block oriented approach is to transfer one block at a time to and from the Thunder Board. The Thunder Board's ESP generates an interrupt at the end of each block, and the next block of data is then transferred. This continues until the last block is processed.

There are three phases to the block processing: first in chain, middle of chain, and last in chain. The following discussions on continuous versus non-continuous DMA transfer describe the details required to make this approach work.

Using non-continuous DMA for playback

```
First in Chain activity:
  Hook the interrupt vector.
  Setup the interrupt mask to allow interrupts on this IRQ.
  Program the DMA controller using non auto-initialize mode.
  Set the sample rate by programming the time constant.
  If using compression:
    Program the ESP for the first transfer with reference byte.
  Else:
    Program the ESP for the first transfer without reference byte.

Middle of Chain activity:
  While more data is available:
    In the foreground, prepare the next block to be transferred.
    Upon interrupt:
      Save all registers.
      Acknowledge the ESP's interrupt by reading I/O address 2xE.
      If next block is available:
        Program the DMA controller.
        Program the ESP to process the next prepared block.
    Else:
      Signal the foreground that the transfer is complete.
    Acknowledge the motherboard's interrupt by outputting a 20h to
    I/O address 20h.
    Restore all registers.
  Perform a return from interrupt instruction, IRET.
End of while.
```

```
End of Chain activity:
  Reset the interrupt mask that allowed interrupts on this IRQ.
  Unhook the interrupt vector.
  Disable DMA channel 1.
```

Using continuous DMA for playback

```
First in Chain activity:
  Hook the interrupt vector.
  Setup the interrupt mask to allow interrupts on this IRQ.
  Program the DMA controller using auto-init mode.
  Set the sample rate by programming the time constant.
  Send half the DMA buffer size to the ESP as the block length.
  If using compression:
    Program the ESP for the first transfer with reference byte.
  else:
    Program the ESP for the first transfer without reference byte.

Middle of Chain activity:
  While more data is available:
    In the foreground, Prepare the next block to be transferred.
    Upon interrupt:
      Save all registers.
      Acknowledge the ESP's interrupt by reading I/O address 2xE.
    If next block is available:
      Load the next prepared block into the consumed portion of the
      DMA buffer.
```

```
      Else:
        Send a Finish DMA command to the ESP.
      Acknowledge the motherboard's interrupt by outputting a 20h to
      I/O address 20h.
      Restore all registers.
    Perform a return from interrupt instruction, IRET.
End of while.


End of Chain activity:
  Reset the interrupt mask that allowed interrupts on this IRQ.
  Unhook the interrupt vector.
  Disable DMA channel 1.
```

Using non-continuous DMA for recording:

```
First in Chain activity:
  Hook the interrupt vector.
  Setup the interrupt mask to allow interrupts on this IRQ.
  Program the DMA controller using non auto-initialize mode.
  Set the sample rate by programming the time constant.
  If using compression:
    Program the ESP for the first transfer with reference byte.
  Else:
    Program the ESP for the first transfer without reference byte.
```

```
Middle of Chain activity:
  While more data is available:
    In the foreground, prepare the next block to be transferred.
    Upon interrupt:
      Save all registers.
      Acknowledge the ESP's interrupt by reading I/O address 2xE.
      If next block is available:
        Program the DMA controller.
        Program the ESP to process the next prepared block.
      Else:
        Signal the foreground that the transfer is complete.
      Acknowledge the motherboard's interrupt by outputting a 20h
      to I/O address 20h.
      Restore all registers.
    Perform a return from interrupt instruction, IRET.
  End of while.

End of Chain activity:
  Reset the interrupt mask that allowed interrupts on this IRQ.
  Unhook the interrupt vector.
  Disable DMA channel 1.
```

Using continuous DMA for recording:

```
First in Chain activity:
     Hook the interrupt vector.
     Setup the interrupt mask to allow interrupts on this IRQ.
     Program the DMA controller using auto-init mode.
     Set the sample rate by programming the time constant.
     Send half the DMA buffer size to the ESP as the block length.
     If using compression:
       Program the ESP for the first transfer with reference byte.
     else:
       Program the ESP for the first transfer without reference byte.
Middle of Chain activity:
     While recording:
       Upon interrupt:
         Save all registers.
         Acknowledge the ESP's interrupt by reading I/O address 2xE.
         If next block is available:
         Load the next prepared block into the consumed portion of the
         DMA buffer.
         Else:
           Send a Finish DMA command to the ESP.
         Acknowledge the motherboard's interrupt by outputting a 20h
         to I/O address 20h.
         Restore all registers.
       Perform a return from interrupt instruction, IRET.
       In the foreground, when the next block is recorded, move it from
       the DMA buffer to disk or memory.
     End of while.
```

```
End of Chain activity:
     Reset the interrupt mask that allowed interrupts on this IRQ.
     Unhook the interrupt vector.
     Disable DMA channel 1.
```

Chapter A *Examples of Programming the Thunder Board for DMA Transfers*

B FM Rate to Time Conversion Tables

Use the the following tables to compute attack, decay, release values for Attack/Decay Rate (60h to 75h) and Sustain Level/Release Rate Registers (80h to 95h).

These tables list time periods (in milliseconds) for the full spread of rate values (63 to 0) and show two durations for each rate value:

- 10% to 90% (or 90% to 10%)

  Indicates the time period over which the envelope makes 80% of the attack, decay, or release transition.

- 0% to 100% (or 100% to 0%)

  Indicates the total duration for the attack, decay, or release.

Note that a sustain column is not provided; the sustain bits specify a level, in decibels below the peak of envelope, for the sustain period. The sustain period is maintained as long as the note is played (Key On is set to 1).

Rate	Attack Time (in milliseconds) (10%-90%)	(0%-100%)	Decay/Release Time (in milliseconds) (90%-10%)	(100%-0%)
63	0.00	0.00	0.51	2.40
62	0.00	0.00	0.51	2.40
61	0.00	0.00	0.51	2.40
60	0.00	0.00	0.51	2.40
59	0.11	0.20	0.58	2.74
58	0.11	0.24	0.63	3.20
57	0.14	0.30	0.81	3.84
56	0.19	0.38	1.01	4.80
55	0.22	0.42	1.15	5.48
54	0.26	0.46	1.35	6.40
53	0.31	0.56	1.62	7.68
52	0.37	0.70	2.02	9.60
51	0.43	0.80	2.32	10.96
50	0.49	0.92	2.68	12.80
49	0.61	1.12	3.22	15.36
48	0.73	1.40	4.02	19.20
47	0.85	1.56	4.62	21.92
46	0.97	1.84	5.38	25.56
45	1.13	2.20	6.42	30.68

Table 7 Rate Table For Rates 63 to 45

*10%-90% is equivalent to -86.4 dB to -9.6 dB

0%-100% is equivalent to -96 dB to 0 dB

Rate	Attack Time (in milliseconds) (10%-90%)	(0%-100%)	Decay/Release Time (in milliseconds) (90%-10%)	(100%-0%)
44	1.45	2.76	8.02	38.36
43	1.70	3.12	9.24	43.84
42	1.94	3.68	10.76	51.12
41	2.26	4.40	12.84	61.36
40	2.90	5.52	16.04	76.72
39	3.39	6.24	18.48	87.68
38	3.87	7.36	21.52	102.24
37	4.51	8.80	25.68	122.72
36	5.79	11.04	32.08	153.44
35	6.78	12.48	36.96	175.36
34	7.74	14.72	43.04	204.48
33	9.02	17.60	51.36	245.44
32	11.58	22.08	64.16	306.88
31	13.57	24.96	73.92	350.72
30	15.49	29.44	86.08	408.96
29	18.05	35.20	102.72	490.88
28	23.17	44.16	128.32	613.76
27	27.14	49.92	147.84	701.44
26	30.98	58.88	172.16	817.92
25	36.10	70.40	205.44	981.76

Table 8 Rate Tables For Rates 44 to 25

*10%-90% is equivalent to -86.4 dB to -9.6 dB

0%-100% is equivalent to -96 dB to 0 dB

Rate	Attack Time (in milliseconds) (10%-90%)	(0%-100%)	Decay/Release Time (in milliseconds) (90%-10%)	(100%-0%)
24	46.34	88.32	256.64	1227.52
23	54.27	99.84	295.68	1402.88
22	61.95	117.76	344.32	1635.84
21	72.19	140.80	410.88	1963.52
20	92.67	176.84	513.28	2455.04
19	108.54	199.68	591.36	2805.76
18	123.90	235.52	688.64	3271.68
17	144.38	281.60	821.76	3927.04
16	185.34	353.28	1026.56	4910.08
15	217.09	399.36	1182.72	5611.52
14	247.81	471.04	1377.28	6543.36
13	288.77	563.20	1643.52	7854.08
12	370.69	706.56	2053.12	9820.16
11	434.18	798.72	2365.44	11223.04
10	495.62	942.08	2754.56	13086.72
9	577.54	1126.40	3287.04	15708.16
8	741.38	1413.12	4106.24	19640.32
7	868.35	1597.44	4730.88	22446.08
6	991.23	1884.16	5509.12	26173.44
5	1155.04	2252.80	6574.08	31416.32
4**	1482.75	2826.24	8212.48	39280.64

Table 9 Rate Table For Rates 24 To 4

*10%-90% is equivalent to -86.4 dB to -9.6 dB

0%-100% is equivalent to -96 dB to 0 dB

**There is no waveform for rates less than 4.

C FM Key Scaling Level Tables

The two tables that follow list the key scaling level attenuation for each octave, and for notes within the octave, for the 3 dB/octave KSL setting of the KSL/ Total Level registers (40h to 55h). The first table shows the attenuation (in dB) for octaves 0 through 3, while the second shows the same for octaves 4 through 7.

Note that you can calculate the 1.5 dB and 6 dB KSL attenuation by halving and doubling the values shown in this table.

F-Num High Nibble	Octave 0	Octave 1	Octave 2	Octave 3
0	0.000	0.000	0.000	0.000
1	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	1.875
4	0.000	0.000	0.000	3.000
5	0.000	0.000	1.125	4.125
6	0.000	0.000	1.875	4.875
7	0.000	0.000	2.625	5.625
8	0.000	0.000	3.000	6.000
9	0.000	0.750	3.750	6.750
10	0.000	1.125	4.125	7.125
11	0.000	1.500	4.500	7.500
12	0.000	1.875	4.875	7.875
13	0.000	2.250	5.250	8.250
14	0.000	2.625	5.625	8.625
15	0.000	3.000	6.000	9.000

Table 10 Key Scaling Levels for Octaves 0 Through 3

* The value of the F-Number four most significant bits

**1.5 dB values are one-half of these; 6 dB are twice

F-Num High Nibble	Octave4	Octave 5	Octave 6	Octave 7
0	0.000	0.000	0.000	0.000
1	0.000	3.000	6.000	9.000
2	3.000	6.000	9.000	12.000
3	4.875	7.875	10.875	13.875
4	6.000	9.000	12.000	15.000
5	7.125	10.125	13.125	16.125
6	7.875	10.875	13.875	16.125
7	8.625	11.625	14.625	17.625
8	9.000	12.000	15.000	18.000
9	9.750	12.750	15.750	18.750
10	10.125	13.125	16.125	19.125
11	10.500	13.500	16.500	19.500
12	10.875	13.875	16.875	19.875
13	11.250	14.250	17.250	20.250
14	11.625	14.625	17.625	20.625
15	12.000	15.000	18.000	21.000

Table 11 Key Scaling Levels for Octaves 4 Through 7

* The value of the F-Number four most significant bits

**1.5 dB values are one-half of these; 6 dB are twice

D FM Standard Pitch Values

The following tables show standard pitch values for all notes within the eight octave range of the FM synthesizer. Use them to set F-Numbers in Block and F-Number registers (A0h to A8h and B0h to B8h).

NOTE: OCTAVE	C	C#	D	D#
0	16.35	17.32	18.35	19.45
1	32.70	34.65	36.71	38.89
2	65.41	69.30	73.42	77.78
3	130.81	138.59	146.83	155.56
4	261.63	277.18	293.66	311.13
5	523.25	554.37	587.33	622.25
6	1046.50	1108.73	1174.66	1244.51
7	2093.00	2217.46	2349.32	2489.02
8	4186.01			

Table 12 Standard Pitch Values: C, C#, D, and D#

NOTE: OCTAVE	E	F	F#	G
0	20.60	21.83	23.12	24.50
1	41.20	43.65	46.25	49.00
2	82.41	87.31	92.50	98.00
3	164.81	174.61	185.00	196.00
4	329.63	349.23	369.99	392.00
5	659.26	698.46	739.99	783.99
6	1328.51	1396.91	1479.98	1567.98
7	2637.02	2793.83	2959.96	3135.96

Table 13 Standard Pitch Values: E, F, F#, and G

NOTE: OCTAVE	G#	A	A#	B
0	25.96	27.50	29.14	30.87
1	51.91	55.00	58.27	61.74
2	103.83	110.00	116.54	123.47
3	207.65	220.00	233.08	246.94
4	415.30	440.00	466.16	493.88
5	830.61	880.00	932.33	987.77
6	1661.22	1760.00	1864.66	1975.53
7	3322.44	3520.00	3729.31	3951.07

Table 14 Standard Pitch Values: G#, A, A#, and B