

SoundprocessTM

Digital Sound Synthesizer

For the Ensoniq Mirage

By Mark Cecys



Triton

Contents:

Introduction

1 - Getting Started

- Starting Soundprocess
- The Front Panel
- Selecting New Programs
- Editing Parameters
- Loading and Saving Sound Banks
- Making Data Disks
- Disk Error Messages

2 - Configuration

- System Configuration
- MIDI Interface
- Program Parameters

3 - Patch Programming

- Patch Architecture
- Oscillator Pair Parameters
- Filter Parameters
- Misc. Patch Parameters

4 - Waves and Wavesamples

- Wave Parameters
- Wavesample Parameters
- Disk Functions

Appendix A - Parameter Chart

Appendix B - MIDI Specifications

INTRODUCTION

One reason why personal computers have become so popular is because of their flexibility. If you want to change a sound inside one of your keyboards, you load your patch editor program into your computer. When you are creating a new arrangement for a song, you run your sequencer. And finally when you need to calculate your band's expenses, you can load and run your spreadsheet program. Fortunately you didn't have to buy a new computer for each different task. The program that was loaded into your computer actually defined what your computer does.

Your Mirage is also a computer. When you load in an operating system, instructions are loaded that tell your Mirage exactly what it's supposed to do. If you could load another operating system, a whole new set of instructions could be loaded, telling your Mirage to do something entirely different.

Soundprocess is a new set of instructions. By defining a completely new voice architecture, your Mirage can now play 48 memory resident multi-timbral sounds.

SYSTEM OVERVIEW

In order to understand how **Soundprocess** works, several terms must first be explained.

To create sound, **Soundprocess** uses two types of Waveforms. **WAVES** are single cycle waveforms that continuously repeat to produce a static pitch and timbre. They can be of any form, from the classic sawtooth and square waves to a sampled cycle of orchestral strings.

You should already be familiar with **WAVESAMPLES**. They are digital recordings that contain many cycles of a sound. Unlike **WAVES**, the pitch and timbre usually vary throughout the waveform.

Both types of waveforms are controlled by **PATCHES**. Selecting which Waveforms will be played by the oscillators, controlling the amplitude and filter envelopes and defining the oscillator configurations are all controlled by the **PATCHES**. They define all the characteristics of a particular sound.

A **PROGRAM** is simply a group of **PATCHES** organized into keyboard split points. Every split point has an associated **PATCH** that will play when an incoming note falls inside its range.

Referring to the Block Diagram, note information comes from either the built-in keyboard or from an external source through the MIDI port. When the Mirage keyboard is played, **Soundprocess** will use the front panel selected **PROGRAM** to determine which **PATCH** to play. If the note comes in through the MIDI port, the **PROGRAM** that will be used is decided by the state of the Multi-timbral Switch. When the switch is off, **Soundprocess** treats the external controller like the built-in keyboard by using the front panel selected **PROGRAM**. If the switch is on, the Channel To Program list is used to select the source **PROGRAM**. Each of the 16 channels can have any one of the 32 **PROGRAMS** assigned to it.

In either case the note is compared to the six splits in the selected **PROGRAM** and depending on the position of the note on the keyboard, a corresponding **PATCH** is selected. The **PATCH** selects the appropriate waveforms and turns its oscillators on to produce an output sound.

DIFFERENCES

It is important to understand that **Soundprocess** is not an enhancement of an existing Mirage operating system. A totally new voice architecture was created, transforming your Mirage into a completely different musical instrument. It is not intended to replace the sampling operating systems. **Soundprocess**

adds a powerful new dimension that greatly increases the flexibility designed into your Mirage.

Even though the sampling operating systems and **Soundprocess** have much in common, several basic differences should be pointed out.

First, **Soundprocess** does not have any user sampling capability. Ensoniq already has a sampling operating system (MASOS) that is well supported with many visual editors. Rather than duplicating these efforts with a new version of user sampling, **Soundprocess** can read **WAVESAMPLES** and **WAVES** directly from your Ensoniq disks.

Next, **Soundprocess** does not include an on-board sequencer. Every available resource inside the Mirage is used to generate the sounds. To include a sequencer would result in sacrificing some of the important features in **Soundprocess**.

Finally, **Soundprocess** cannot play a **WAVESAMPLE** that is over 64 pages long. Playing long samples would quickly use up all the available memory in your Mirage, defeating most of the features in **Soundprocess**. You must remember that the key to **Soundprocess**' power is that it combines sampling with synthesis. By combining the subtle complexities of sampled sounds with the efficiency of synthesis, that same limited memory can now be used to generate an entire spectrum of interesting sounds.

One important final comment should be made before we get started. There is nothing that is inherently difficult when you create your own sounds with **Soundprocess**. As in learning any skill, it takes some patience and some trial and error to develop the techniques that are required. If you spend some time with it, the results will be rewarding.

INTRODUCTION

SECTION 1 - GETTING STARTED

This section describes the basic things you need to know to use Soundprocess. You'll find out how to start the program, how the Front Panel works, how to change **PROGRAMS** and **PARAMETERS** and finally how to Save and Load **SOUND BANKS** with **DATA DISKS**.

STARTING SOUNDPROCESS

Turn your Mirage on and insert the Soundprocess System Disk with the label side up. It will take about 20 seconds for Soundprocess to load. (Since Soundprocess "boots" differently from other operating systems, you will hear different sounds from your disk drive. This is normal, no damage is occurring to the disk or the drive). When the program is loaded the display will show a "01". Play a few notes on your keyboard to check your setup. If you don't hear any sound, something is probably set or wired incorrectly in your setup. Check your *Mirage Musician's Manual* for setup instructions.

THE FRONT PANEL

Because Soundprocess lets you do many new things with your Mirage, some of the front panel keys have been redefined with new functions. Also a few changes have been made to the way data is displayed on the two digit display.

You can select and edit 4 types of data with Soundprocess:

	Display Format	Function Key
1 - PROGRAM	X X	<PROG>
2 - PARAMETER NUMBER	.X X	<PARM>
3 - PARAMETER VALUE	X.X	<VALUE>
4 - SOUND BANK	.X.X	<SAVE> or <LOAD>

Decimal points next to the displayed digits are used to let you know which type of data is currently being displayed.

When selecting **PROGRAM** numbers, the display will show a 2 digit number without any decimal points. To view the currently selected **PROGRAM** or to select a new **PROGRAM**, press the new <PROG> key. (The Soundprocess <PROG> key is the labeled SEQ REC key. The 0 / PROG functions only as a number "0" entry.)

PARAMETER NUMBERS are displayed with a decimal point before the first digit on the left side. Pressing the <PARM> key selects this mode.

A decimal point between the two digits means that a **PARAMETER VALUE** is being displayed. This mode can be selected only while in the **PARAMETER NUMBER** mode and pressing the <VALUE> key.

Finally, when Loading or Saving a **SOUND BANK**, decimal points before both digits are displayed. The Bank <LOAD> or <SAVE> keys are pressed to enter this mode.

SOUNDPROCESS KEY SUMMARY

- <PROG> Displays the selected **PROGRAM**. To select another **PROGRAM**:
- Enter a 2 digit number between 01-32 using the <0>-<9> keys.
 - Or press the <ON>/<OFF> keys to scroll through all 32 **PROGRAMS**.
- <PARM> Displays the selected **PARAMETER NUMBER**. To select another **PARAMETER NUMBER**:
- Enter a 2 digit number between 00-99 using the <0>-<9> keys.

- Or press the <ON>/<OFF> keys to scroll through all 100 **PARAMETER NUMBERS**.

- <VALUE> Displays the value of the selected Parameter. This key is recognized only while in the **PARAMETER NUMBER** mode. To modify this value:
- Press the <ON>/<OFF> keys to scroll through the Parameter's value range.
 - Or press any of the **PARAMETER EDIT** keys.
(<UPPER>, <LOWER>, <MID> or <RESTORE>)
- <LOAD>
<SAVE> Load or Save a **SOUND BANK** to disk. Enter a Bank number 1-3 and press <ENTER>.
- <CANCEL> Abort a command or clear an error message.
- <ENTER> Execute a command.
- <STOP> "Panic button" used to clear hanging notes. Does the same as a MIDI "All Notes Off"

Since the Mirage does not have a Data Entry Slider, the following **PARAMETER EDIT** keys have been added to speed things up while editing Parameter Values.

- <UPPER> Sets the **PARAMETER VALUE** to it's maximum limit.
- <LOWER> Sets the **PARAMETER VALUE** to it's minimum limit.
- <MID> Sets the **PARAMETER VALUE** to halfway between the upper and lower limit.
- <RESTORE> Restores the **PARAMETER** to the **VALUE** it had when it was selected.

SELECTING NEW PROGRAMS

As mentioned before, **PROGRAMS** are nothing more than **PATCH** split-points assigned to various zones across the keyboard. **PATCHES** define the sounds while **PROGRAMS** define the keyboard areas in which they will be played. When you play your Mirage, the selected **PROGRAM** is used to find which **PATCH** should be played. When a number is displayed without any decimal points, this indicates that **Soundprocess** is displaying the current **PROGRAM** number. To enter this display mode from another display mode, press the <PROG> key from the front panel.

A new **PROGRAM** can be selected any one of three ways:

- 1 - Entering a 2 digit number between 01-32 from the front panel. After the first digit is entered, the display will flash a cursor ("L") in the second digit position. **Soundprocess** will wait until the second digit is entered before changing to the new **PROGRAM**. Note that if a number is entered that will be out of the range of 01-32, that key entry is ignored.
- 2 - Pressing the <ON/OFF> keys to scroll through the 32 **PROGRAMS**. If the <ON/OFF> keys are pressed for longer than 1/2 of a second, the **PROGRAM** number will automatically start scrolling until the key is released or it reaches it's top or bottom limit at 01 or 32.
- 3 - Sending a "Program Change" status message through MIDI. When this message is received, **Soundprocess** will automatically enter the **PROGRAM** display mode if it is in another display mode. Note that **Soundprocess PROGRAM** numbers

SECTION 1 - GETTING STARTED

are offset by +1 from MIDI Program numbers. Soundprocess PROGRAM "01" is really a MIDI Program "00". Many MIDI controllers make this offset, some do not. An out of range PROGRAM number will be ignored.

EDITING PARAMETERS

All of the variables used in Soundprocess are defined by the PARAMETERS. If you wish to change PATCH assignments to PROGRAMS, change your MIDI configuration or create new PATCH sounds, then you must modify the PARAMETERS associated with that function.

Modifying a PARAMETER is usually a two step process. First, press the <PARM> key and select the PARAMETER NUMBER that you will edit. Then press the <VALUE> key to display and edit the actual VALUE of that PARAMETER.

Like selecting PROGRAMS, PARAMETER NUMBERS can be selected by either entering a number between 00-99 or by scrolling through all the PARAMETER NUMBERS by pressing the <ON> or <OFF> keys.

PARAMETER VALUES can be edited by either scrolling with the <ON>/<OFF> keys or by using the special PARAMETER EDIT keys. Pressing any of the number keys, <0> - <9>, while editing VALUES will return you to selecting PARAMETER NUMBERS. If you have finished editing a PARAMETER, you can directly select a new PARAMETER NUMBER by entering a two digit number, without pressing the <PARM> key.

This should give you an overall view on how to edit PARAMETERS. Detailed information about the PARAMETERS, is given throughout this manual.

LOADING AND SAVING SOUND BANKS

When you modify or create sounds with Soundprocess, you can Save these changes to a diskette in your disk drive. Later when you wish to hear these new sounds, you can Load the information back into your Mirage from the diskette.

All WAVES and WAVESAMPLES, PATCHES, CONFIGURATION data and PROGRAMS are transferred to the diskette as a SOUND BANK when Saved. When a SOUND BANK is then Loaded from the diskette, your Mirage will sound and behave exactly as it did when you Saved the SOUND BANK.

To avoid reading incompatible data from other Mirage operating systems, you must first create a Soundprocess DATA DISK. This is easily done with any Mirage formatted diskette in the drive and activating PARAMETER 17. Each DATA DISK can hold 3 SOUND BANKS.

EXAMPLE : Creating a Soundprocess DATA DISK.

- 1 - Format a blank 3.5" disk with any of the available disk formatters.
- 2 - Boot the Soundprocess System Disk. Insert the formatted disk into the disk drive
- 3 - Turn on PARAMETER 17:
 - Press the <PARM> key to display PARAMETER NUMBERS.
 - Press the <1> key followed by the <7> key to select PARAMETER 17
 - Press the <VALUE> key. The display shows PARAMETER 17 is "OFF".
 - Press the <ON> key. PARAMETER 17 is now "ON".
- 4 - Command Soundprocess to create a DATA DISK by pressing the <ENTER> key.
- 5 - The disk drive will turn on for only a couple of seconds then stop.

SECTION 1 - GETTING STARTED

You can now Save **Soundprocess SOUND BANKS** to this disk.

NOTE: Creating a **DATA DISK** does NOT transfer a copy of the **Soundprocess** operating system to the disk. If you "boot" a **DATA DISK**, your Mirage will run whatever operating system your Formatter copied on to the disk. You can only "boot" **Soundprocess** from a **System Disk**.

EXAMPLE: Saving a **SOUND BANK** to disk.

- 1 - Press the <SAVE> key on the front panel. An ".S.b" with the ".b" flashing is displayed, waiting for you to enter a Bank number.
- 2 - Enter a 1,2 or 3 depending on which bank you wish to Save.
- 3 - Press the <ENTER> key to command **Soundprocess** to Save the **SOUND BANK** to disk.

Load a **SOUND BANK** using the same procedure as above except press the <LOAD> key instead. The display will be prefixed by an "L" for Load.

If you want to **QUIT Soundprocess** and boot to another Mirage operating system, Load **SOUND BANK "0"** instead of the usual 1,2 or 3. This will cause your Mirage to load and run the operating system that is on the diskette in your drive.

DISK ERROR MESSAGES

Disk **ERROR** messages are identified by flashing both display digits without any decimal points. They can be cleared by pressing the <CANCEL> key.

- | | |
|-----------|--|
| nS | - Disk is not a Soundprocess DATA DISK . Probably an Ensoniq disk. |
| Sd | - "System Disk" in drive. You can only Load or Save to DATA DISK . |
| nF | - "NOT FOUND" - A SOUND BANK cannot be Loaded because one was never Saved. |
| Pd | - "PROTECTED DISK" - Cannot write to disk because the write-protect notch on the diskette is open. |
| nd | - "NO DISK" in drive. |
| ud | - "UN-FORMATTED" in drive. |
| dE | - Unidentified "DISK ERROR". |

SUMMARY

An overall view of the **Soundprocess** user interface was described in this section. There are three basic things you can do from the front panel:

- 1 - Use the <PROG> key to select the 32 **PROGRAMS** in **Soundprocess**.
- 2 - Use the <PARM>, <VALUE> and **DATA EDIT** keys to modify **PARAMETERS**.
- 3 - Use the <SAVE> and <LOAD> keys to access **SOUND BANKS** on disks.

SECTION 2 - CONFIGURATION

This section describes the **SYSTEM CONFIGURATION PARAMETERS**, the **MIDI INTERFACE PARAMETERS** and the **PROGRAM PARAMETERS**. You will be shown how to use the powerful multi-timbral capabilities of **Soundprocess** and how to set the **PATCH SPLITS** in **PROGRAMS**.

SYSTEM CONFIGURATION

The System Configuration **PARAMETERS** are global variables that affect all **PROGRAMS** and **PATCHES**.

[49] MASTER TUNE 00-99

Adjusts the frequency of all the oscillators up or down in 1/20 semitone increments. A value of "49" is set for the "A" above Middle "C" at 440 hz. Decreasing this value towards zero will lower the pitch, while increasing this value towards 99 will raise the pitch. This **PARAMETER** is used if you have to tune your Mirage to other instruments that are not easily tuned such as a piano.

[51] PITCH BEND 00-12

Adjusts the sensitivity of the pitch bend wheel in half-step increments. A value of "00" disables the pitch wheel, while a value of "12" gives the pitch wheel a range of a full octave up or down. This **PARAMETER** also affects MIDI pitch bend values.

MIDI INTERFACE

[48] OMNI MODE ON-OFF

Controls how **Soundprocess** responds to MIDI channel numbers on messages that it receives. When "ON", **Soundprocess** acts on all channel messages that enter the MIDI port. It will respond to all 16 MIDI channels. When "OFF", **Soundprocess** enters the **POLY** mode, responding to only those MIDI messages whose channel number matches [47] **POLY CHANNEL**. Information sent to your Mirage on the 15 remaining channels will be ignored.

[47] POLY CHANNEL 01-16

Assigns which one of the 16 MIDI channels **Soundprocess** will respond to when it is in the **POLY** mode ([48] **OMNI** mode "OFF"). It also assigns the MIDI channel number that **Soundprocess** will **SEND** it's messages on. All information (Notes, Controllers, Program Changes, etc.) will be sent through this channel number. This affects all MIDI modes (**OMNI**, **POLY**, **LOCAL**, **REMOTE** and **MULTI**).

[52] MULTI MODE ON-OFF

Turning **MULTI** mode "ON" enables the unique multi-timbral feature of **Soundprocess**. Unlike **OMNI** / **POLY** modes where only one **PROGRAM** can be selected to play at any one time, **MULTI** mode allows up to 16 separate **PROGRAMS** to be active at once. You can assign a different **PROGRAM** to each of the 16 MIDI

channels. So when a MIDI Note message is received, it's channel number determines which **PROGRAM** will be played.

Each MIDI channel is now independent of all the other channels. Controller (MOD and PITCH wheel, SUSTAIN pedal) information will only affect those Notes that match the channel number of the received controller. In other words, the only Notes that will be affected by a Channel 16 PITCH BEND are the ones that were sent by your external MIDI controller on Channel 16. Active Notes sent to your Mirage on Channels 1-15 will not be affected by that particular PITCH BEND message.

MULTI mode is a powerful feature to use with your multi-track sequencer. You can assign to each track a different MIDI Channel number. Then using those Channel numbers you can assign each one of them to play a different **PROGRAM** with **Soundprocess**. Track 1, for example, can play a Bass instrument, Track 2 plays a Sax, and Track 3 plays a Drum Kit. Like having 16 separate synthesizers inside your Mirage!

If your external MIDI keyboard can send more than one Channel for each Note message, you can now do **PROGRAM** layering. Each note pressed on the MIDI keyboard will play 2 or more **PROGRAMS** at once. If you need more than the 6 splits for each **PROGRAM** in **Soundprocess**, you can increase this with your MIDI controller and **Soundprocess** in **MULTI** mode. Suppose the MIDI keyboard can be programmed for 2 splits. Together with the 6 splits per **PROGRAM**, this will give you 12 split points across the keyboard.

When **MULTI** is "OFF", **Soundprocess**' response to MIDI information is controlled by the setting of the [48] **OMNI** switch. **Soundprocess** will either accept information on all Channels (**OMNI** = "ON") or accept information on only one MIDI Channel (**OMNI** = "OFF" = **POLY**). In either case, **Soundprocess** will only play the **PROGRAM** which is selected on the front panel.

When **MULTI** is "ON", the state of the **OMNI** switch is overridden and is ignored. **PARAMETERS** [31]-[46] determine which **PROGRAM** to play based on the MIDI Channel number of the incoming message. MIDI out information still behaves as before, controlled by [47] **POLY CHANNEL**, [50] **OVERFLOW** and [53] **LOCAL**.

[31]-[46] MULTI CHANNEL TO PROGRAM LIST 00-32

These **PARAMETERS** are active only when the [52] **MULTI** switch is "ON". They correspond to the 16 MIDI Channels. **PARAMETER** [31] sets the **PROGRAM** that will be played on Channel 1 while **PARAMETER** [46] sets the same for Channel 16.

Any one of the 32 **PROGRAMS** can be set for each Channel. A setting of "00" will cause that channel to be ignored.

EXAMPLE: Setting up **MULTI** mode

- 1 - Assign **PROGRAM** numbers to MIDI Channel numbers.
 [31] = "08" - MIDI Channel 1 plays **PROGRAM** 8
 [32] = "31" - MIDI Channel 2 plays **PROGRAM** 31
 .
 .
 .
 [45] = "01" - MIDI Channel 15 plays **PROGRAM** 1
 [46] = "00" - Ignore all information on Channel 16
- 2 - Set [52] **MULTI MODE** to "ON".

[53] **LOCAL MODE** **ON-OFF**

SECTION 2 - CONFIGURATION

Determines whether or not the local Mirage keyboard will activate sound output. When "ON", Mirage keyboard will play the selected **PROGRAM** and it's Note information will be sent out the MIDI port. When **LOCAL** is "OFF" (**REMOTE** mode), **Soundprocess** sends keyboard information out through MIDI, but does not play the selected **PROGRAM**. This also affects the MOD wheel, PITCH BEND and SUSTAIN pedal in the same way as the local Mirage keyboard. **Soundprocess PROGRAMS** will respond to MIDI IN data only.

CAUTION: WHEN LOCAL MODE IS "OFF", PLAYING THE KEYBOARD WILL *NOT* PRODUCE ANY SOUND OUTPUT. IT CAN APPEAR AS THOUGH SOMETHING IS WORKING INCORRECTLY IN YOUR SETUP. ALWAYS MAKE SURE THAT LOCAL IS "ON" IF YOU ARE NOT USING THIS FEATURE!

[50] MIDI OVERFLOW ON-OFF

Because of the way the hardware was designed, **Soundprocess** can play a maximum of only eight output voices at once. If all eight voices are busy and you play another note on the keyboard, **Soundprocess** has to "steal" a voice from one of the busy voices so that it can sound the new note. It must try to do this in such a way that you won't notice this has happened.

Most of the time, **Soundprocess** will optimally "steal" these notes without notice. You won't be aware that this is even going on. However, there can be situations when **Soundprocess** can no longer hide this "stealing" process and it will be noticeable.

To get around this, instead of "stealing" additional notes from the internal voices, you can **OVERFLOW** these additional voices to a second slave Mirage giving 16 output voices.

When **OVERFLOW** is "ON", your Mirage will not send any MIDI note data until all eight internal voices are busy. When an additional note is played, instead of "stealing" an internal voice, **Soundprocess** sends this note out the MIDI port. It will continue to do this as long as all eight internal voices are busy.

The MIDI OUT port of the master Mirage must be wired to the MIDI IN of the slave Mirage. So when the master can no longer play any more notes, the slave unit receives these additional notes and plays them. Make sure that the slave unit is Loaded with the same **SOUND BANK** and set to the same **PROGRAM** as the master Mirage. Also, both units should be set to the same **MIDI CHANNEL**.

CAUTION: IF OVERFLOW IS "ON" WITHOUT A SLAVE UNIT LINKED IN, IT WILL APPEAR AS THOUGH YOUR MIRAGE IS MISSING NOTES AT TIMES WHEN YOU PLAY THEM. THIS IS BECAUSE THE EXCESS NOTES ARE BEING SENT OUT THE MIDI PORT WITHOUT SOUNDING THEM. ALWAYS BE SURE THAT OVERFLOW IS "OFF" IF YOU ARE NOT USING THIS FEATURE!

PROGRAM PARAMETERS

When you play **Soundprocess**, what you hear is a result of the **PATCHES**. Most of the many elements that characterize the sounds you hear are defined in the various **PROGRAMS** grouped in what is called a **PATCH**. The term is borrowed from the old analog synth days where sounds were defined by wiring the many Modules together with Patch Cords.

Which of the 48 **PATCHES** will sound when you press a note on the keyboard is decided in the

SECTION 2 • CONFIGURATION

PROGRAMS.

Once you've created a new **PATCH**, you must now decide on a keyboard layout strategy. If the new **PATCH** is a Bass sound, you may want it played on only the bottom two octaves of the keyboard, with the upper three octaves playing your Sax **PATCH**. Or if your new **PATCH** is the high end of a Voice Choir, you may want to play it on the top octave and have five other "multi-sampled" Voice Choir **PATCHES** handling the lower octaves giving the effect of one Choir through a five octave range.

These multi-**PATCH** keyboard layouts are called **PROGRAMS**. They consist of **TOP KEY** and **PATCH NUMBER** pairs called **SPLITS**. In the above example, you would enter the **PATCH NUMBER** of the Bass sound and enter middle C as the **TOP KEY** for **SPLIT #1**. The Sax **PATCH NUMBER** and three octaves above middle C for **TOP KEY** as **SPLIT #2**. You can define up to 6 **SPLITS** for each of the 32 **PROGRAMS** in **Soundprocess**.

When a key is played, **Soundprocess** compares this key to the **TOP KEYS** of all six **SPLITS** starting with **SPLIT #1** on up to **SPLIT #6** until a range match is found. If the key is outside the range of all six splits, then the key will be ignored and no sound will output. The range of **SPLIT #1** starts at MIDI key number 00 to it's **TOP KEY**. **SPLIT #2**'s range starts at **SPLIT #1**'s **TOP KEY** to **SPLIT #2**'s **TOP KEY**. And so on. Because of the two digit display, **TOP KEYS** have a range of 0-127 but must be entered in hex format of 00-7F. The Mirage keyboard has a hex range of 24-60 (See your *Advanced Sampler's Guide* for information on hex formats).

The values of the **PROGRAM PARAMETERS** [00]-[11] reflect those of the **PROGRAM** that is currently selected from the front panel. To edit a different **PROGRAM**, enter the **PROGRAM** display mode by pressing the <PROG> key, enter the desired **PROGRAM NUMBER** then press the <PARM> key to edit the **PROGRAM PARAMETERS**.

[00]-[11] PROGRAM PARAMETERS

<u>SPLIT #1</u>		<u>SPLIT #2</u>		<u>SPLIT #3</u>	
[00]	TOP KEY 00-7F	[02]	TOP KEY 00-7F	[04]	TOP KEY 00-7F
[01]	PATCH 01-48	[03]	PATCH 01-48	[05]	PATCH 01-48
<u>SPLIT #4</u>		<u>SPLIT #5</u>		<u>SPLIT #6</u>	
[06]	TOP KEY 00-7F	[08]	TOP KEY 00-7F	[10]	TOP KEY 00-7F
[07]	PATCH 01-48	[09]	PATCH 01-48	[11]	PATCH 01-48

EXAMPLE: Defining PROGRAM 26

Select the PROGRAM

- 1 - Press the <PROG> key to enter the **PROGRAM** select mode.
- 2 - Press the <2> then the <6> key to select **PROGRAM 26**.

Edit the PARAMETERS

- 3 - Press the <PARM> key to select **PARAMETER NUMBERS**.
- 4 - Press the <0> key twice to select **PARAMETER 00**.
- 5 - Press the <VALUE> key to edit the value of **PARAMETER 00**.
- 6 - Press the <MID> key for a value of "3F".
- 7 - Press the <OFF> key 3 times for a **TOP KEY** value of "3C".
- 8 - Press the <0> then the <1> key to select **PARAMETER 01**.
- 9 - Press the <VALUE> key to edit.
- 10 - Press the <LOWER> key then press the <ON> key 6 times for a **PATCH** value of "07".

SECTION 2 - CONFIGURATION

11- Continue editing all of the **PROGRAM PARAMETERS** with the following values:

	TOP KEY	PATCH
SPLIT #2	43	36
SPLIT #3	27	48
SPLIT #4	4C	14
SPLIT #5	53	43
SPLIT #6	53	19

Try this layout on your keyboard. You will notice that although you entered 6 different **SPLITS** for this **PROGRAM**, only 4 **PATCHES** are active across the keyboard. The top octave of the keyboard produces no sound at all! Here are the **TOP KEYS** you entered for the 6 **SPLITS**:

Key number:	24	30	3C	48	54	60
Mirage KBD	-----	-----	-----	-----	-----	-----
SPLIT #1	*****					
SPLIT #2	-----*****					
SPLIT #3	-----					
SPLIT #4	-----*****					
SPLIT #5	-----*****					
SPLIT #6	-----					

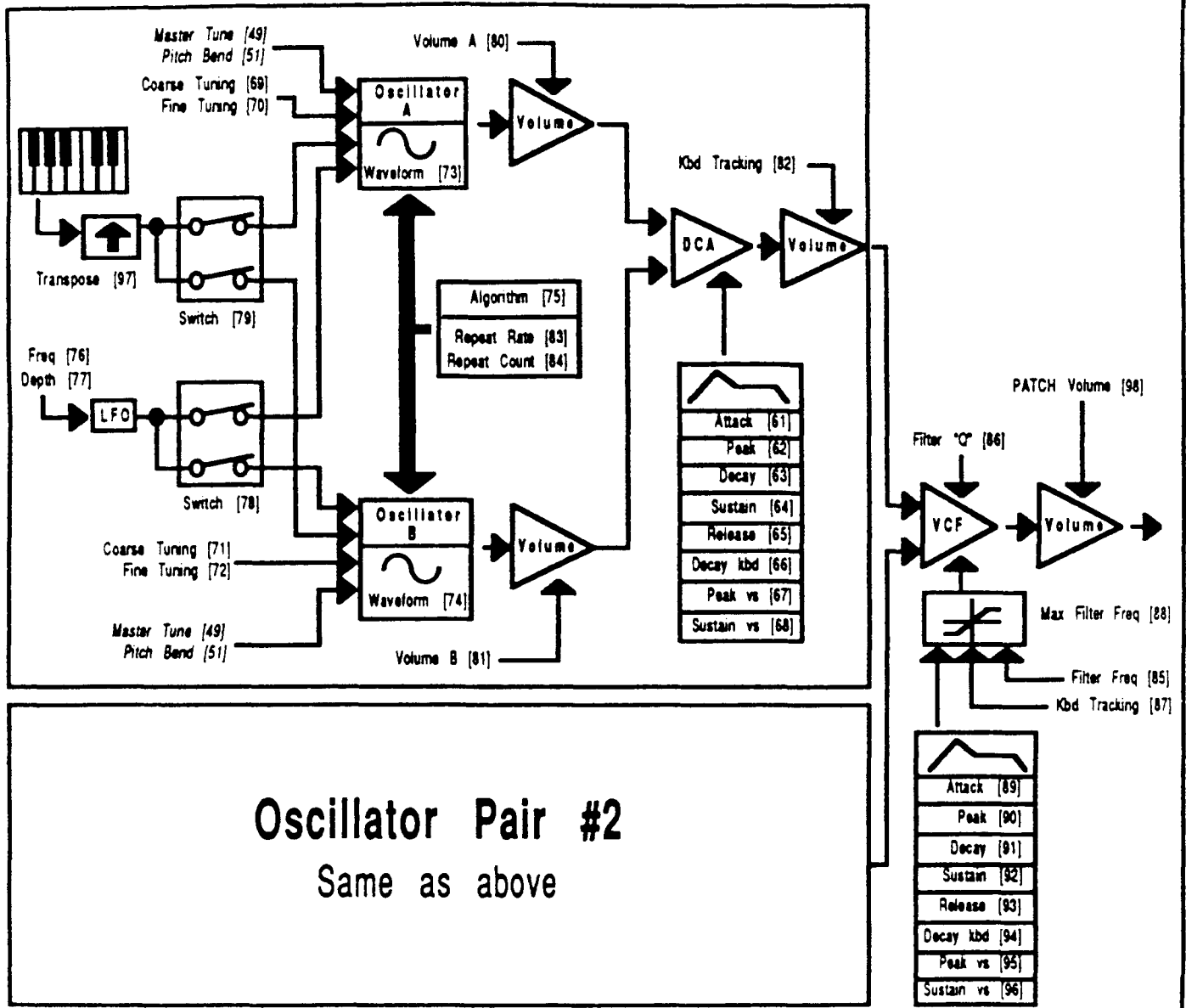
You cannot hear **SPLIT #3** because it has a **TOP KEY** less than both **SPLIT #1** and **#2**. **SPLIT #6** gets masked out because it has the same **TOP KEY** value as **SPLIT #5**. Since **Soundprocess** compares the entered key number to the active range of **SPLIT #5** before **SPLIT #6**, **SPLIT #5** gets the priority.

You cannot hear anything on the top octave because all 6 **SPLITS** have **TOP KEY** values below number 54. If you set the **TOP KEY** of **SPLIT #6** to a value of "60" or higher, you can now play 5 different **PATCHES** across the keyboard with **SPLIT #6** filling in the top octave.

Raising the **TOP KEY** of **SPLIT #1** to "60" will mask out all 5 remaining **SPLITS** giving only one **PATCH** across the keyboard.

SECTION 2 - CONFIGURATION

Oscillator Pair #1



PATCH Select [54]
OSCILLATOR PAIR Select [60]

Master Tune and Pitch Bend are global Parameters that affect all 48 PATCHES

Soundprocess

PATCH ARCHITECTURE

SECTION 3 - PATCH PROGRAMMING

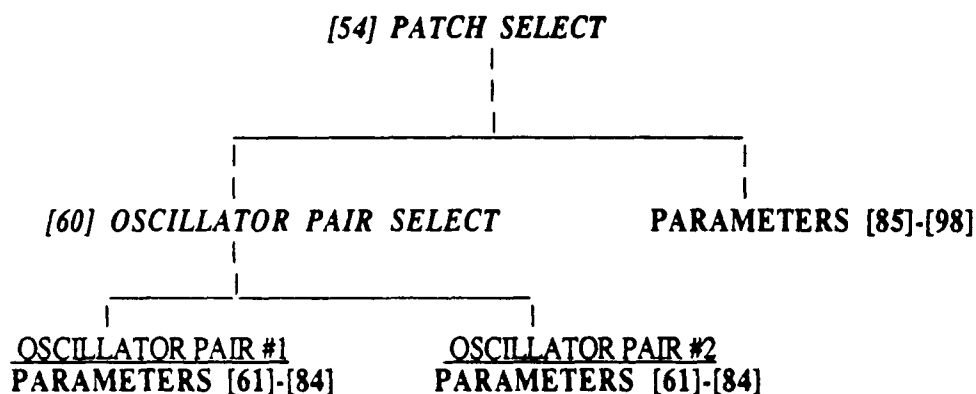
This section describes all the **PARAMETERS** that make up a Soundprocess **PATCH**. A general overview of the **PATCH Architecture** is given, followed by a detailed look at the **PATCH PARAMETERS**.

PATCH ARCHITECTURE

Take a look at the Soundprocess Patch Architecture diagram on the following page. You will notice two blocks called **OSCILLATOR PAIR #1** and **OSCILLATOR PAIR #2**. Since there are two volume Envelope Generators for four Oscillators, the Oscillators are grouped into two **PAIRS** of two Oscillators. In addition, each of the two identical **PAIRS** have one LFO, one DCA with a 5 stage Envelope Generator and a Keyboard Tracking Volume control. Depending on the selected **ALGORITHM**, each one of the four Oscillators can sound any one of the 72 available Waveforms in your Mirage memory. The output signal of each **PAIR** is then mixed and shaped by the Filter Envelope before being sent to the Output.

When editing a **PATCH**, you first select one of the 48 **PATCHES** with **PARAMETER [54] PATCH SELECT**. Then you can select whether you want to work with **PAIR #1** or **PAIR #2**. Each **OSCILLATOR PAIR** shares the block of **PARAMETERS** from [60] to [84]. Which **PAIR** will be accessed is determined by [60] **OSCILLATOR PAIR SELECT**. The Filter and Output **PARAMETERS** [85]-[98] are not affected by this selection since they are common to both **PAIRS**.

Shown below is how the **PATCH PARAMETERS** are organized and selected.



OSCILLATOR PAIR PARAMETERS

VOLUME ENVELOPE

[61] ATTACK RATE	00-31
[62] PEAK LEVEL	00-31
[63] DECAY RATE	00-31
[64] SUSTAIN LEVEL	00-31
[65] RELEASE RATE	00-31
[66] DECAY KBD	00-31
[67] PEAK vs	00-31
[68] SUSTAIN vs	00-31

Pressing a note on the keyboard will start the Oscillators and trigger the Filter and two Amplitude Envelopes in each **PATCH**. The first stage of the Envelope, the **ATTACK**, ramps up from the "00" level at the set rate until it

reaches the **PEAK** level. It then ramps down at the **DECAY** rate to the **SUSTAIN** level. The envelope remains at the **SUSTAIN** level as long as the key remains pressed. When you let go of the key, the Envelope ramps down to the "00" level at the **RELEASE** rate for the final stage of the Envelope.

Decreasing the **RATE** value will make the Envelope ramp faster. A value of "00" will give instantaneous jumps. Note that you are setting a **RATE** and not **TIME**. If you decrease the **PEAK** level, the Envelope will reach the **PEAK** stage faster because the **ATTACK** ramp has a shorter distance to go until it reaches the new lower **PEAK** level.

PARAMETERS [66], [67] and [68] modify the Envelope depending on what you are doing on your keyboard.

[66] **DECAY KBD** will lower the **DECAY** value for higher note values on the keyboard giving slow **DECAYS** on the low end of the keyboard and fast **DECAYS** on the top end of the keyboard. This is typically used on plucked or hammered instruments where the sound dies out quicker on the higher notes and frequencies (The **SUSTAIN** is set at "00"). With the **DECAY KBD** set to zero, set the **DECAY** value you desire on the lowest note then adjust the **DECAY KBD** value for the highest note. The middle notes will then be of varying degrees in between the two values.

[67] **PEAK VS** and [68] **SUSTAIN VS** add velocity sensitivity to the **PEAK** and **SUSTAIN** **PARAMETERS**. The larger the value, the more sensitivity is added.

[69] **OSC A TUNING - OCTAVE** 00-07
[70] **OSC A TUNING - FINE** 00-FF*
[71] **OSC B TUNING - OCTAVE** 00-07
[72] **OSC B TUNING - FINE** 00-FF*

The two Oscillators in each **PAIR** are labeled as A and B. These four **PARAMETERS** will tune the two Oscillators anywhere in an 8 octave range to within 1/256th of an octave. The **FINE TUNING** values are in hex notation to give the full 256 per octave step resolution.

[73] **OSC A WAVEFORM** 01-72
[74] **OSC B WAVEFORM** 01-72

The source Waveform for each of the two Oscillators is selected by these two **PARAMETERS**. Waveforms 01-16 are **WAVESAMPLES** while Waveforms 17-72 are single-cycle **WAVES**. Some **ALGORITHMS** allow the full 72 Waveforms to be selected, while other **ALGORITHMS** allow only the first 16 **WAVESAMPLES** as their source for one or both Waveforms. The value which is entered for those that only allow **WAVESAMPLES** folds over when entering in values over 16. Waveform 17 selects **WAVESAMPLE 1**, Waveform 18 selects **WAVESAMPLE 2** and so on to Waveform 72.

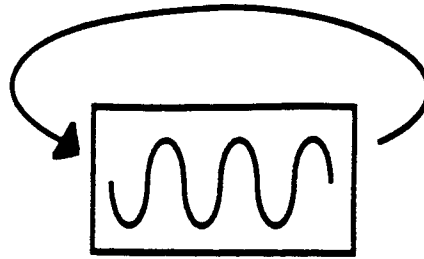
[75] **OSC ALGORITHM** 01-11

Whereas the two [73], [74] **WAVEFORM** **PARAMETERS** tell the Oscillators what Waveforms to play, the **ALGORITHMS** tell the Oscillators how to play those Waveforms. Since the selected **ALGORITHM** affects both of the Oscillators in the **PAIR**, only one **ALGORITHM** is selected for each Oscillator **PAIR**.

Each of the Soundprocess Oscillators can play in one of four modes of operation. These four modes are configured in various configurations between the two Oscillators in each **PAIR** to form the 11 **ALGORITHMS**.

SECTION 3 - PATCH PROGRAMMING

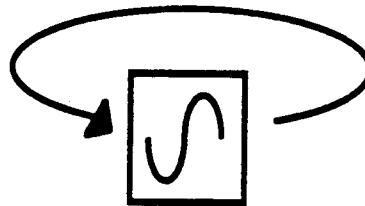
CONTINUOUS MODE



The Oscillator will continuously repeat it's selected Waveform in this mode. A single-cycle WAVE (Waveforms 17-72) will typically produce a static pitch while the a WAVESAMPLE (Waveforms 1-16) will loop around to the beginning of the WAVESAMPLE whenever it reaches it's end. This causes low notes to repeat slowly and high notes to repeat very quickly since the WAVESAMPLE will be played at a faster rate. One thing to watch out for when selecting WAVESAMPLES in this mode, is that the Oscillator will stop whenever it encounters a Loop Marker (a sample value of "00") in the WAVESAMPLE. This will cause the WAVESAMPLE to sound only once, stopping at the Loop Marker point. Loop Markers are inserted by Soundprocess whenever the [57] LOOP END PAGE or [58] LOOP END FINE are set to a value less than their maximum limit. In other words the Loop End is not the last byte in the WAVESAMPLE table.

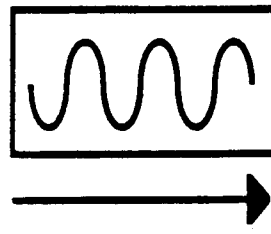
This is the only mode that can play WAVES. The remaining three modes exclusively use the WAVESAMPLES.

LOOP MODE



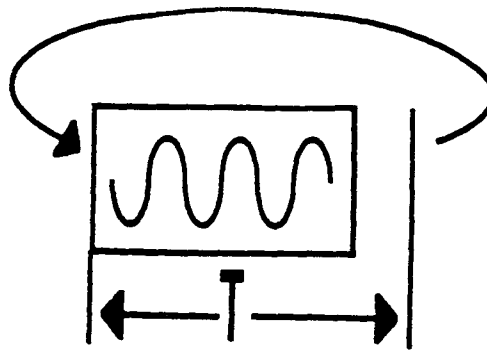
This mode is similar to the CONTINUOUS mode. However, instead of repeating the entire WAVESAMPLE, only the section that is marked off with the WAVESAMPLE LOOPING PARAMETERS [56]-[58] is repeated. The LOOP mode is used only in ALGORITHM 4.

ONE-SHOT MODE



In this mode the selected WAVESAMPLE is triggered only once. The Oscillator will play the WAVESAMPLE from the beginning of the table and stop at either the end of the WAVESAMPLE or at an earlier point if there is a Loop Marker in inside the WAVESAMPLE. Low notes will sound longer when they are played because the Oscillators will scan the WAVESAMPLE at a slower rate to produce a lower pitch.

TIMED RE-TRIGGER MODE



In this mode the selected WAVESAMPLE is started from the beginning of its table. At the same time an internal clock is started with a time-out value entered by [83] RE-TRIGGER TIME. When this internal clock times out, the WAVESAMPLE is re-triggered again from the beginning and the clock is reset to start timing again. This cycle is repeated either indefinitely or a fixed number of times by the value set in [84] REPEAT COUNTER.

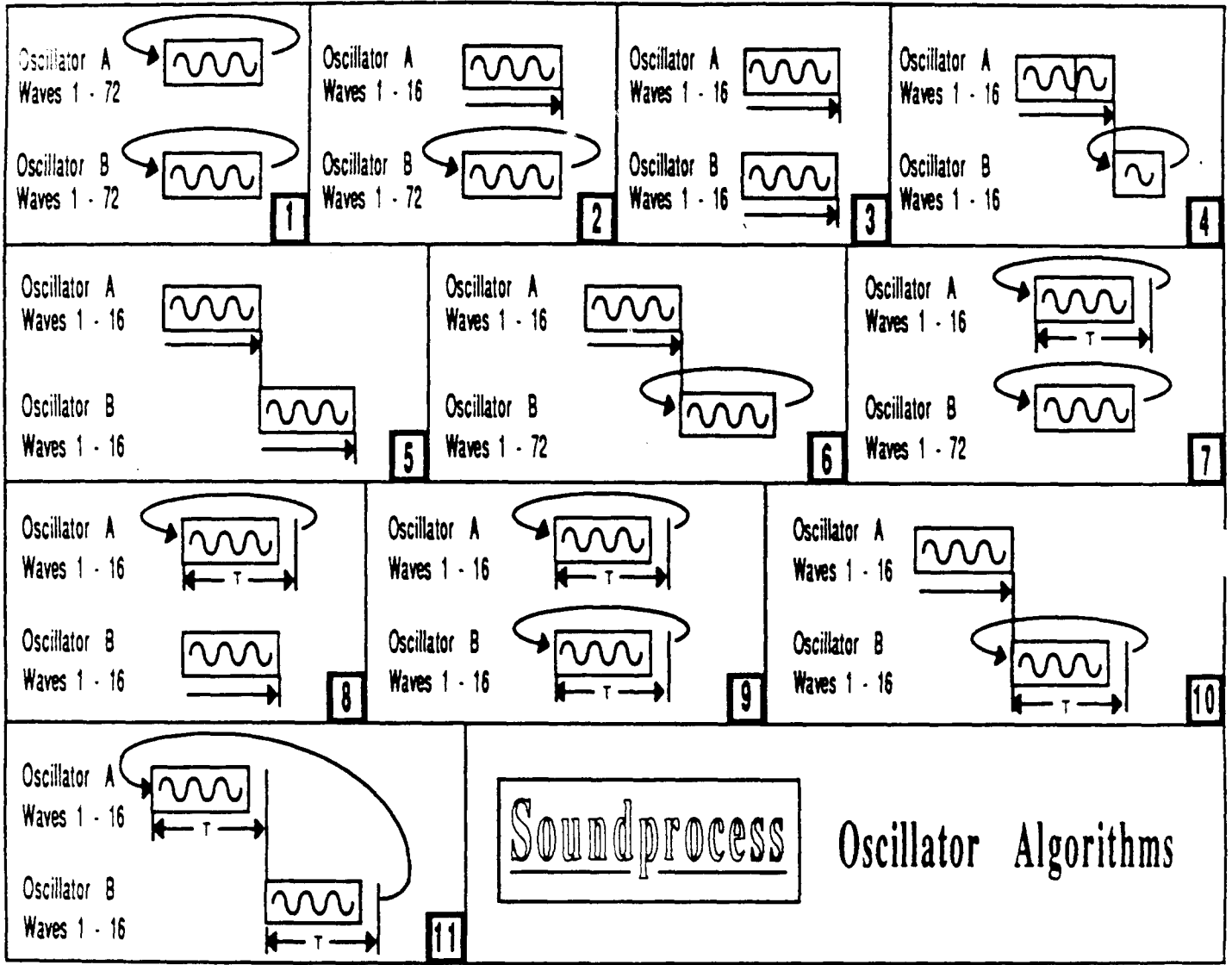
If the WAVESAMPLE has ended before the clock has timed out, the Oscillator will turn off and wait for a time-out. Once re-triggered by the time-out, the Oscillator will turn on again and replay the WAVESAMPLE.

As mentioned before, the 11 ALGORITHMS are various combinations of the four modes described above. The selected ALGORITHM applies to both the Oscillators in the PAIR. Refer to the Soundprocess Oscillator Algorithms diagram on the following page.

- ALGORITHM #1 - Both Oscillators are in the CONTINUOUS mode.
- ALGORITHM #2 - OSC A in ONE-SHOT with OSC B in CONTINUOUS mode.
- ALGORITHM #3 - Both OSC A and OSC B are in ONE-SHOT.
- ALGORITHM #4 - OSC A plays once in ONE-SHOT, then OSC B starts in LOOP mode.
- ALGORITHM #5 - OSC A plays in ONE-SHOT, followed by OSC B in ONE-SHOT mode.
- ALGORITHM #6 - OSC A plays in ONE-SHOT, followed by OSC B in CONTINUOUS mode.
- ALGORITHM #7 - OSC A plays in RE-TRIGGER with OSC B in CONTINUOUS mode.
- ALGORITHM #8 - OSC A plays in RE-TRIGGER with OSC B in ONE-SHOT mode.
- ALGORITHM #9 - Both OSC A and OSC B play in RE-TRIGGER mode.
- ALGORITHM #10 - OSC A plays in ONE-SHOT, followed by OSC B in RE-TRIGGER mode.
- ALGORITHM #11 - OSC A and OSC B alternate in RE-TRIGGER mode.

[83] RE-TRIGGER TIME 01-FF*

Sets the time-out value for the TIMED RE-TRIGGER Oscillator mode. The value is entered in 10 ms increments. Note that because of the two digit Mirage display, this value is in hex notation to give a full 10 ms - 2.5 second range.



Soundprocess

Oscillator Algorithms

[84] REPEAT COUNTER 00-63

Sets the number of times the RE-TRIGGER mode repeats it's cycle. When set to "00", the Oscillator will repeat indefinitely.

[76] LFO FREQUENCY 00-63

Sets the speed of the LFO oscillator. Adjustable from .3Hz at a value of "01" to a maximum of 25Hz at a value of "63". Entering a value of "00" will modulate the LFO oscillator with random values.

The LFO is used to vary the pitch of one or both of the Oscillators in the PAIR. It's effect is similar to vibrato. Although the LFO can be any arbitrary waveform once down-loaded through MIDI, the System Disk LFO is a sine wave.

[77] LFO DEPTH 00-63

Scales how much of an effect the LFO will have on varying the pitch of the Oscillators. A value of "00" lets you control DEPTH with the Mod Wheel.

[78] LFO SWITCH 00-03

Enables or disables LFO modulation on the two Oscillators in the PAIR.

"00" = LFO has no effect on OSC A and OSC B.

"01" = LFO modulates OSC A only.

"02" = LFO modulates OSC B only.

"03" = LFO modulates both OSC A and OSC B.

[79] KEYBOARD FREQ TRACKING SWITCH 00-03

Disables keyboard note values from setting the pitch of the Oscillators. They will sound only at their TUNING frequency when a note is played with FREQ TRACKING set to off.

"00" = FREQ TRACKING is off on both OSC A and OSC B. The pitch will be the same for all notes played on the keyboard.

"01" = FREQ TRACKING enabled for OSC A only.

"02" = FREQ TRACKING enabled for OSC B only.

"03" = FREQ TRACKING is on for both OSC A and OSC B. This is the setting used most often since both Oscillators will track the pitch of the note played on the keyboard.

[80] VOLUME A 00-63

[81] VOLUME B 00-63

Adjusts the output amplitude of the two Oscillators in each PAIR. These PARAMETERS are often used to balance the levels of the 4 Oscillators in a PATCH. They can also be used to turn off the output of an unused Oscillator by setting it's VOLUME value to "00".

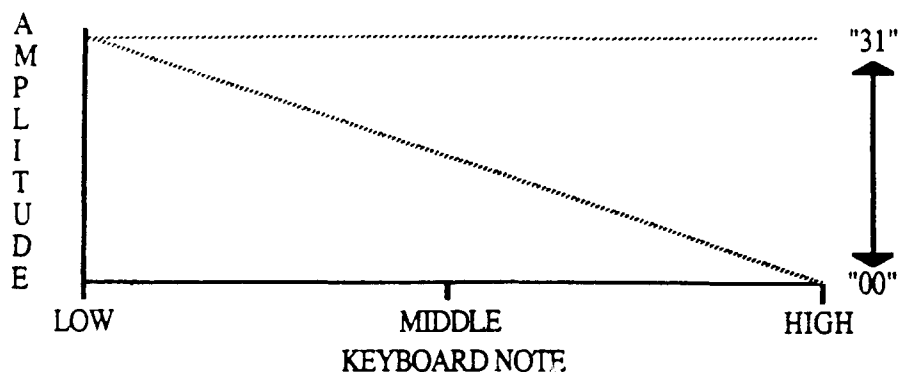
[82] KEYBOARD VOLUME TRACKING 00-63

SECTION 3 - PATCH PROGRAMMING

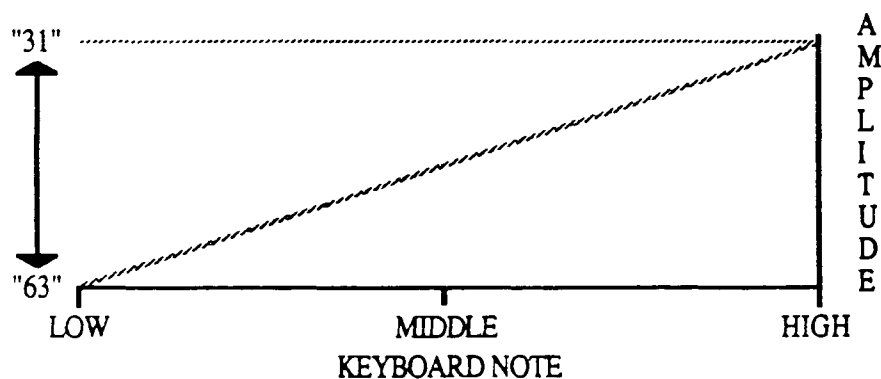
Adjusts the output level of a **PAIR** depending on where on the keyboard the note is pressed.

A value of "31" will disable **VOLUME TRACKING** and Soundprocess will not vary the **PAIR** amplitude across the keyboard.

Entering values below "31" will start decreasing the **PAIR** volume on higher notes. Entering in smaller numbers towards "00" will increase the effect that higher notes will have on the **PAIR** volume. A value of "00" will give full attenuation on the top key and no attenuation on the bottom key. This is shown in the diagram below.



Increasing the **TRACKING** value to numbers above "31" will have the complete opposite effect. Now the lower notes will decrease the **PAIR** output volume. A value of "63" will completely attenuate the bottom key while having no effect on the volume of the top key on the keyboard. The remaining keys in between will be at levels linearly varying between the two extremes.



VOLUME TRACKING can be used to balance the volume of a **PAIR** across the keyboard so that it sounds at the same level on low notes as it does on high notes. It can also be used to vary the mix of the two **PAIRS** in a **PATCH**. If **PAIR #1** has a **VOLUME TRACKING** of "00" and **PAIR #2** has a **VOLUME TRACKING** of "63", then playing the **PATCH** across the keyboard will pan between **PAIR #1** on the low end and **PAIR #2** on the top end of the keyboard.

SECTION 3 - PATCH PROGRAMMING

FILTER PARAMETERS

All of the **PARAMETERS** discussed so far are part of the **OSCILLATOR PAIR**. Since there are two **PAIRS** for each **PATCH**, there are two sets of all the above **PARAMETERS** in each **PATCH**.

Looking at the Patch Architecture Diagram, you can see that the outputs of both **PAIRS** are summed at the Filter input. Both **PAIRS** share one Filter.

[85] FILTER CENTER FREQUENCY 00-99

This sets the base cut-off frequency of the low pass Filter. The Filter envelope and Filter **KBD TRACKING** will "ride" on top of this setting.

[86] FILTER RESONANCE 00-63

Adjusts the amount of Filter resonance or "Q". At the maximum value of "63" the Filter will oscillate at the center frequency.

[87] FILTER KBD TRACKING 00-63

On many sounds that you will be creating, after adjusting the Filter to get the sound that you want on the high notes, you will notice a lot of unwanted noise when this same sound is played on the low notes. If you readjust the Filter to clean up the low notes, the high notes will lose their previous brightness because the Filter is now set at a lower frequency. What you need is a way for the Filter to track the note played, increasing the Filter's center frequency as you play higher notes on the keyboard.

This is exactly what **FILTER KBD TRACKING** does. As you increase its value, the Filter's center frequency will increase in greater amounts as you play up the scales on the keyboard. A value of "00" will give no tracking at all.

Typically, you adjust the Filter **PARAMETERS** for the sound you want on the low notes with **FILTER TRACKING** set to "00". Then increase this **PARAMETER** for the sound on the high notes.

[88] MAX FILTER FREQUENCY 00-99

Sets the absolute maximum limit on the Filter's center frequency. It will have no effect on the Filter as long as the Filter's frequency is below the **MAX** value. If the Filter Envelope or some other source tries to raise the Filter frequency above this **MAX** value, it will be "clipped" to the **MAX** frequency.

FILTER ENVELOPE

[89] ATTACK RATE	00-31
[90] PEAK LEVEL	00-31
[91] DECAY RATE	00-31
[92] SUSTAIN LEVEL	00-31

SECTION 3 - PATCH PROGRAMMING

[93] RELEASE RATE	00-31
[94] DECAY KBD	00-31
[95] PEAK vs	00-31
[96] SUSTAIN vs	00-31

These are similar to the **VOLUME ENVELOPE** except that the Filter frequency is controlled by the envelope rather than volume. Refer to the paragraph on **VOLUME ENVELOPE** for details on these **PARAMETERS**.

It should be pointed out that the Filter Envelope is offset by [85] **FILTER FREQ** and by [87] **FILTER KBD TRACKING**. This means that the **SUSTAIN** level will be the sum of all three values (**FREQ**, **TRACKING** and **SUSTAIN**). The exception is when the **SUSTAIN** value is set to "00". In this case the **DECAY** will ignore this summed value and instead it will ramp down to the "00" frequency level. The **RELEASE** ramp always ramps down to the "00" level.

MISC. PATCH PARAMETERS

[97] PATCH TRANSPOSE 00-60

This will transpose the incoming keyboard Note in half step increments. A value of "12" will cause no effect on the Note, while "00" will transpose the Note up one octave and "60" will transpose the Note down 4 octaves.

[98] PATCH VOLUME 00-63

Adjusts the overall level of the **PATCH** output. This is used to balance the **PATCH** volume to other **PATCH** volumes.

[12] SET PATCH TO DEFAULT ON-OFF

Sets all the **PATCH PARAMETERS** of the selected **PATCH** to their default values. Affects only the **PATCH** which is selected by [54] **PATCH SELECT**. This can be used as a starting point when creating new sounds. **PARAMETERS** [60] through [98] are modified. After turning this **PARAMETER** "ON", you must press the <ENTER> key to initiate this function.

EXAMPLE: Setting **PATCH 21** to default values.

SELECT THE PATCH NUMBER

- 1 - Press the <PARM> key to select **PARAMETERS**.
- 2 - Select **PARAMETER** [54] by pressing the <5> and the <4> key.
- 3 - Press the <VALUE> to display **PARAMETER VALUES**.
- 4 - Select **PATCH 21** by scrolling the display to "21" using the <ON>/<OFF> keys.

SET THE SELECTED PATCH TO DEFAULT VALUES

- 5 - Press the <1> key followed by the <2> key to select **PARAMETER** [12].

SECTION 3 - PATCH PROGRAMMING

- 6 - Press the <VALUE> key to display **PARAMETER VALUES** again.
- 7 - Press the <ON> key to enable the **DEFAULT** function.
- 8 - Press the <ENTER> key to command **Soundprocess** to set the **PATCH** to it's default values.
- 9 - Since you already have **PATCH 21** selected, you can now start adjusting the various **PATCH PARAMETERS** to create the sound you want.

SECTION 3 - PATCH PROGRAMMING

SECTION 4 - WAVEFORMS AND DISK FUNCTIONS

This section describes the **PARAMETERS** associated with **WAVES** and **WAVESAMPLES**. A description is also given on Loading single Waveforms and **PATCHES** from disks with the Disk Function **PARAMETERS**.

WAVE PARAMETERS

[18] WAVE SELECT 17-72

Before you adjust any of the **WAVE PARAMETERS**, you must first select which one of the 56 **WAVES** you wish to work on. Once the **WAVE** is selected with **WAVE SELECT**, the remaining **WAVE PARAMETERS** [19]-[30] will now reflect the **VALUES** of the selected **WAVE**.

[20] WAVE TYPE / SIZE 01-04

This defines the **WAVE** to be one of two **TYPES** and one of three **SIZES**. A **WAVE** can be either a Waveform which was loaded into **Soundprocess** from a **MASOS** disk or a Waveform which was user defined. In this second **TYPE**, you can enter the levels of the first ten harmonics and let **Soundprocess** calculate the resultant Waveform. This lets you define almost any possible single cycle Waveform easily and logically.

WAVES that **Soundprocess** calculates are always one page in length. **WAVES** that you load in from your **MASOS** disks can be either one, two or four pages long. You must remember that there are 56 single page **WAVES** in **Soundprocess**. If a **WAVE** is longer than one page, it will use up additional **WAVES** to cover it's size. For example, if **WAVE 41** is four pages long then **WAVES 41,42,43** and **44** will be used to hold this 4 page **WAVE**. The **TYPE/SIZE PARAMETER** for **WAVE 41** must be set to "04". It does not matter what values **WAVES 42-44** are set to.

Because of the way the **Mirage** was designed, there are restrictions on assigning the **SIZES** to the **WAVES**. All **WAVES** can be at least one page long. All odd **WAVES** can also be 2 pages long. Finally, every 4th **WAVE** can be 4 pages long. Note that even numbered **WAVES** can only be one page in length. This configuration is shown in the chart below.

	<u>1,2 or 4 pages</u>	<u>1 page only</u>	<u>1 or 2 pages</u>	<u>1 page only</u>
	17	18	19	20
W	21	22	23	24
A	25	26	27	28
V	29	30	31	32
E	33	34	35	36
	37	38	39	40
N	41	42	43	44
U	45	46	47	48
M	49	50	51	52
B	53	54	55	56
E	57	58	59	60
R	61	62	63	64
S	65	66	67	68
	69	70	71	72

[20] WAVE TYPE/SIZE values have the following meanings:

- "01" - WAVE SIZE is one page. WAVE is harmonically defined.
- "02" - WAVE SIZE is one page. WAVE is loaded from MASOS disk.
- "03" - WAVE SIZE is two pages. WAVE is loaded from MASOS disk.
- "04" - WAVE SIZE is four pages. WAVE is loaded from MASOS disk.

[19] WAVE AMPLITUDE 00-63

Adjusts the amplitude of the resultant WAVE calculated from the harmonic levels.

[21] HARMONIC #1 LEVEL	00-63
[22] HARMONIC #2 LEVEL	00-63
[23] HARMONIC #3 LEVEL	00-63
[24] HARMONIC #4 LEVEL	00-63
[25] HARMONIC #5 LEVEL	00-63
[26] HARMONIC #6 LEVEL	00-63
[27] HARMONIC #7 LEVEL	00-63
[28] HARMONIC #8 LEVEL	00-63
[29] HARMONIC #9 LEVEL	00-63
[30] HARMONIC #10 LEVEL	00-63

Adjusts the levels of the first 10 harmonics in the selected WAVE. This enables you to actually define any arbitrary WAVE by defining it's harmonic spectrum. Soundprocess then takes these harmonic levels with the [19] WAVE AMPLITUDE value and calculates a resultant Waveform.

You must realize that Soundprocess only calculates the WAVE. It does NOT analyze the selected WAVE. You cannot load in a WAVE from a MASOS disk and modify it by adjusting the harmonic levels. Sound process knows nothing about the harmonic levels of that loaded WAVE. You can only modify those WAVES which were previously created by defining their harmonic spectrum.

Here is the procedure for defining your own WAVE:

- 1 - Select the WAVE you want to create or modify with [18] WAVE SELECT.
- 2 - Set [20] TYPE/SIZE to "01".
- 3 - Adjust the HARMONIC LEVELS with [21]-[30] and the [19] AMPLITUDE.
- 4 - With one of the PARAMETERS still selected from Step 3, press the <ENTER> key to calculate the resultant WAVE.
- 5 - Play the WAVE on your keyboard. Repeat Steps 3 and 4 until you get the desired WAVE.

WAVESAMPLE PARAMETERS

The WAVESAMPLE PARAMETERS are used to set the LOOP points for ALGORITHM #4.

WAVEFORMS AND DISK FUNCTIONS

[55] WAVESAMPLE SELECT 01-16

Selects which one of the 16 **WAVESAMPLES** you wish to work with. When adjusting the **LOOP** points, keep in mind that **WAVESAMPLES 1-8** are 16 pages long and **WAVESAMPLES 9-16** are 32 pages in length.

[56] LOOP START PAGE	WAVESAMPLES 1-8	00-0F*
	WAVESAMPLES 9-16	00-1F*

Selects the starting point at which the **LOOP** will begin.

[57] LOOP END PAGE	WAVESAMPLES 1-8	00-0F*
	WAVESAMPLES 9-16	00-1F*

[58] LOOP END FINE 00-FF*

Selects the ending point for the **LOOP**. When this point is reached, the **WAVESAMPLE** is restarted from the **[56] LOOP START PAGE**. Consult your *Advanced Sampler's Guide* that came with your Mirage for details on **LOOPING**.

DISK FUNCTIONS

As discussed in Section 1, you can use the **<LOAD>** key from the front panel to Load a **SOUND BANK** into **Soundprocess**. A **BANK** will contain a complete set of **PROGRAMS**, **PATCHES**, **WAVES** and **WAVESAMPLES**. Everything that made your Mirage sound like it did when you initiated a Save is Loaded back inside. The Disk Functions allow you to Load only part of the information that was Saved in the **BANK**. You can Load any single **PATCH** or Waveform directly from a **SOUND BANK** on disk. In addition, the Disk Functions allow you to read your **MASOS** disks in order to load in your sampled **WAVES** and **WAVESAMPLES**.

[13] DISK FUNCTION TYPE 01-03

There are three **TYPES** of information which you can Load into **Soundprocess** with the Disk Functions.

"01"	- Soundprocess Waveforms (WAVES and WAVESAMPLES)
"02"	- Soundprocess PATCHES
"03"	- MASOS Waveforms

Before you set any of the other Disk Function **PARAMETERS**, you must first select one of the three **TYPES** shown above. When a **TYPE** is selected, the remaining Disk Function **PARAMETERS** will be reset to their starting values, so always make the **TYPE** selection first.

WAVEFORMS AND DISK FUNCTIONS

[14] DISK BANK

Selects the **BANK** from which you will Load the information. On **TYPES** "01" and "02" you select 01-03 since **Soundprocess** has only three **BANKS** per disk. When Loading **TYPE** "03" **MASOS** Waveforms you must enter a value between 01-06 since **Soundprocess** must know whether you want the "Upper" or "Lower" part of the three **MASOS** Bank.

"01" = Lower Bank #1
"02" = Upper Bank #1
"03" = Lower Bank #2
"04" = Upper Bank #2
"05" = Lower Bank #3
"06" = Upper Bank #3

[15] SOURCE

Specifies the location from where the information will be read. **Soundprocess** Waveforms have a **SOURCE** of 01-72 while **PATCHES** have a **SOURCE** of any number between 01-48. When Loading **MASOS** Waveforms, select the starting page number of where in memory the Waveform was located while running **MASOS** (00-FF*).

[16] DESTINATION

Specifies the location to where the disk information will be Loaded. **Soundprocess** Waveforms and **PATCHES** have the same range as their **SOURCE**. When Loading a **MASOS** Waveform you must now enter which one of the 72 **Soundprocess** Waveforms as a **DESTINATION**. **Soundprocess** will automatically Load the correct number of pages to the specified Waveform. Loading **MASOS** Waveforms to **WAVESAMPLES** will reset the **LOOP** of that **WAVESAMPLE** to a single page **LOOP** on the last page.

The Disk Function will only Load **Soundprocess** **WAVES** one page at a time in a **TYPE** "01" **WAVE** to **WAVE** Load. If the **WAVE** is 4 pages long, then you must do 4 consecutive **TYPE** "01" Loads to read in the full Waveform. (A **TYPE** "03", **MASOS** to **WAVE** Load will read 4 pages in if the **DESTINATION** **WAVE** had a **SIZE** of 4 pages.)

Remember to reset the **PATCH** **WAVEFORM** **PARAMETERS** to their new values after Loading in a **PATCH**. Typically you would have Loaded in the Waveforms to a different location from where they were in the disk **BANK**. The new Loaded **PATCH** may still think it's Waveforms are at the old location.

After setting up the **TYPE**, **BANK**, **SOURCE** and **DESTINATION**, you must press the <ENTER> key to initiate the Disk Function. One of the four Disk Function **PARAMETERS** [13]-[16] must still be selected so that **Soundprocess** will know that it has to do a Disk Function and not something else.

WAVEFORMS AND DISK FUNCTIONS

SOUNDPROCESS PARAMETER LIST

PROGRAM

Program Select	01-32
[00] Top Key #1	00-7F*
[01] Patch #1	01-48
[02] Top Key #2	00-7F*
[03] Patch #2	01-48
[04] Top Key #3	00-7F*
[05] Patch #3	01-48
[06] Top Key #4	00-7F*
[07] Patch #4	01-48
[08] Top Key #5	00-7F*
[09] Patch #5	01-48
[10] Top Key #6	00-7F*
[11] Patch #6	01-48

CONFIGURATION

[31] Ch. #1 Program	00-32
[32] Ch. #2 Program	00-32
[33] Ch. #3 Program	00-32
[34] Ch. #4 Program	00-32
[35] Ch. #5 Program	00-32
[36] Ch. #6 Program	00-32
[37] Ch. #7 Program	00-32
[38] Ch. #8 Program	00-32
[39] Ch. #9 Program	00-32
[40] Ch. #10 Program	00-32
[41] Ch. #11 Program	00-32
[42] Ch. #12 Program	00-32
[43] Ch. #13 Program	00-32
[44] Ch. #14 Program	00-32
[45] Ch. #15 Program	00-32
[46] Ch. #16 Program	00-32
[47] Poly CHANNEL	01-16
[48] OMNI Mode	ON-OFF
[49] MASTER TUNE	00-99
[50] OVERFLOW	ON-OFF
[51] PITCH BEND	00-12
[52] MULTI Mode	ON-OFF
[53] LOCAL Mode	ON-OFF

PATCH

[54] Patch Select	01-48	
[60] Oscillator Pair Select	01-02	
[61] Volume ATTACK Rate	00-31	[89] Filter ATTACK Rate 00-31
[62] Volume PEAK Level	00-31	[90] Filter PEAK Level 00-31
[63] Volume DECAY Rate	00-31	[91] Filter DECAY Rate 00-31
[64] Volume SUSTAIN Level	00-31	[92] Filter SUSTAIN Level 00-31
[65] Volume RELEASE Rate	00-31	[93] Filter RELEASE Rate 00-31
[66] Volume DECAY KBD	00-31	[94] Filter DECAY KBD 00-31
[67] Volume PEAK VS	00-31	[95] Filter PEAK VS 00-31
[68] Volume SUSTAIN VS	00-31	[96] Filter SUSTAIN VS 00-31
[69] Osc A TUNING - OCTAVE	00-07	[85] Filter CENTER FREQ 00-99
[70] Osc A TUNING - FINE	00-FF*	[86] Filter RESONANCE 00-63
[71] Osc B TUNING - OCTAVE	00-07	[87] Filter KBD TRACKING 00-63
[72] Osc B TUNING - FINE	00-FF*	[88] Filter MAX FREQUENCY 00-99
[73] Osc A WAVEFORM	01-72	
[74] Osc B WAVEFORM	01-72	[97] Patch TRANSPOSE 00-60
[75] Osc ALGORITHM	01-11	[98] Patch VOLUME 00-63
[76] LFO FREQ	00-63	
[77] LFO DEPTH	00-63	[12] Set Patch DEFAULT ON-OFF
[78] LFO SWITCH	00-03	
[79] KBD TRACK SWITCH	00-03	
[80] VOLUME A	00-63	
[81] VOLUME B	00-63	
[82] VOLUME TRACKING	00-63	
[83] RE-TRIGGER Time	01-FF*	
[84] REPEAT COUNTER	00-63	

WAVESAMPLE

[55] WAVESAMPLE Select	01-16
[56] LOOP START Page	
WAVES 1-8	00-0F*
WAVES 9-16	00-1F*
[57] LOOP END Page	
WAVES 1-8	00-0F*
WAVES 9-16	00-1F*
[58] LOOP END FINE	00-FF*
[59] LOOP END BYTE	—

WAVE

[18] WAVE Select	17-72
[19] Wave AMPLITUDE	00-63
[20] Wave TYPE / SIZE	01-04
[21] HARMONIC #1 Level	00-63
[22] HARMONIC #2 Level	00-63
[23] HARMONIC #3 Level	00-63
[24] HARMONIC #4 Level	00-63
[25] HARMONIC #5 Level	00-63
[26] HARMONIC #6 Level	00-63
[27] HARMONIC #7 Level	00-63
[28] HARMONIC #8 Level	00-63
[29] HARMONIC #9 Level	00-63
[30] HARMONIC #10 Level	00-63

DISK FUNCTIONS

[13] Type	01-03		
[14] Disk BANK		[15] SOURCE	[16] DESTINATION
1 - WAVEFORM	01-03	1 - WAVEFORM	01-72
2 - PATCH	01-03	2 - PATCH	01-48
3 - MASOS	01-06	3 - MASOS	00-FF*
[17] Make DATA DISK	ON - OFF		

SOUNDPROCESS MIDI SPECIFICATIONS

CHANNEL MESSAGES

Status Byte	Second Byte	Third Byte	Description
\$8x	Note 36 - 96	Velocity 0 - 127	Note off
\$9x	Note 36 - 96	Velocity 0 - 127	Note on
\$Bx	Control 01	Range 0 - 127	Mod Wheel
\$Bx	Control 64	State 0,127	Sustain Pedal off / on
\$Bx	Control 123 - 127	0	All Notes Off
\$Cx	Prog # 0 - 31	----	Program change*
\$Ex	00	Range 0 - 127	Pitch Wheel

x = channel number

Numbers preceded by a "\$" are in hex notation

MIDI Pitch Wheel and Mod Wheel messages will only affect notes which are sounded as a result of MIDI key messages. Notes played on the local keyboard will not be affected. Local Pitch and Mod wheels will not affect these MIDI key messages.

* MIDI Program numbers 0-31 correspond to Soundprocess Programs 1-32. In other words, MIDI Program 31 is the same as Soundprocess Program 32.

SYSTEM EXCLUSIVE MESSAGES

In order to gain access to the System Exclusive messages, a *Computer Control* command must be transmitted by the MIDI controller. Once in this mode, all Channel messages and System Exclusive messages will be recognized by Soundprocess. A *Front Panel* command will return Soundprocess to it's normal mode, disabling the System Exclusive messages. This command can also be used to abort any System Exclusive command in progress.

Front Panel Control

All front panel keys are active.

Normal data messages are displayed.

All Channel messages are transmitted and recognized when received.

Only the *Computer Control* System Exclusive command is recognized. All others will cause a *Not Computer Control* status to be sent out.

Computer Control

Only the <STOP> front panel key is active. When this key is pressed, Front Panel mode is enabled, Computer Control is disabled and a *Not Computer Control* status message is transmitted.

Display always shows "CC"

Channel messages are recognized, but not transmitted.

All System Exclusive messages are recognized.

System Exclusive Control Commands

"Front Panel Control"

CODE: \$01

Returns Soundprocess to it's normal operating mode with the front panel enabled. Any further System Exclusive messages will be ignored.

\$F0	System Exclusive
\$01	Ensoniq code
\$7F	Soundprocess code
\$01	"Front Panel Control" command
\$F7	EOX

Soundprocess returns a "STATUS" message.

"Computer Control"

CODE: \$02

Enables recognition of all System Exclusive messages. Display shows "CC". All front panel keys are disabled except for the <STOP> key. All Channel messages are still recognized.

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$02	"Computer Control" command
\$F7	EOX

Soundprocess returns a "STATUS" message

MIDI SPECIFICATIONS

"Load Sound Bank"

Loads a Sound Bank from disk.

CODE: \$0F

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$0F LOAD SOUND BANK
1-3 BANK number
\$F7 EOX

"Save Sound Bank"

Saves a Sound Bank to disk.

CODE: \$10

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$10 SAVE SOUND BANK
1-3 BANK number
\$F7 EOX

"Execute Disk Function"

Executes Disk Function specified by Parameters [13]-[16]

CODE: \$11

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$11 EXECUTE DISK FUNCTION
\$F7 EOX

"Calculate Wave"

Calculates Wave specified by Parameters [18]-[30]

CODE: \$12

MIDI SPECIFICATIONS

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$12	CALCULATE WAVE
\$F7	EOX

"Status"

STATUS message sent by Soundprocess

CODE: \$00

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$00	STATUS
\$00-\$12	STATUS code
\$F7	EOX

STATUS codes:

\$00	System OK
\$01	Not Computer Control
\$02	Undefined error
\$03	Qualifier over range
\$04	Qualifier under range
\$05	DATA over range
\$06	DATA under range
\$07	Buffer overflow
\$08	Undefined disk error
\$09	Bank not found
\$0A	Unformatted disk
\$0B	No disk in drive
\$0C	Write protected disk
\$0D	Not a Soundprocess disk
\$0E	Can't load or save to System Disk
\$10	Can't calculate on MASOS loaded Wave
\$11	Source - Destination type match error
\$10	Can't load MASOS Wave on harmonic defined Wave

System Exclusive Dump Messages

The Dump messages are used to send and receive Soundprocess Parameters, Waves Wavesamples and various internal tables.

Code Bit Format:

00xx xxxx READ data from Soundprocess to external computer.
01xx xxxx WRITE data to Soundprocess from external computer.

x = Dump message code.

Example: F0 0F 7F 03 21 F7 A request is sent to Soundprocess for the value of Parameter 33.
F0 0F 7F 43 21 01 00 F7 Parameter 33 is set to a value of "01".

DATA format: Each byte is split into 2 nybbles and sent separately as 2 bytes with the high nybble set to zero. The low nybble of the original byte is sent first.
For example, a "D3" is sent as "03" followed by a "0D".

All parameters that are sent or received in Dump messages are in binary with the same data ranges as they are defined in this manual. As an example, Parameter [75] can have any decimal value between 1 and 11. In a Dump message, the value sent is between 01 and 0B.

"Parameter"

READ or WRITE any one of the 100 Soundprocess Parameters.

CODE: \$03 / \$43

READ format:

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$03 READ "Parameter"
0-99 Parameter number
\$F7 EOF

Soundprocess returns either a WRITE "Parameter" message or a "STATUS" message if there is an error.

WRITE format:

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$43 WRITE "Parameter"

MIDI SPECIFICATIONS

0-99	Parameter number
DATA	Parameter value byte in DATA format
\$F7	EOX

Soundprocess returns a "STATUS" message

"Patch Parameters"

CODE: \$04 / \$44

READ or WRITE any of the 48 PATCH PARAMETER BLOCKS.
Parameters [61]-[84] of both Oscillator
Pairs and Parameters [85]-[98] form the
DATA block. Note that each Patch can be given
an 8 character PATCH NAME.

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$04	Read PATCH PARAMETER BLOCK
1-48	Patch number
\$F7	EOX

Soundprocess returns either a WRITE "Patch Parameters" message or a "STATUS" message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$44	Write PATCH PARAMETER BLOCK
1-48	Patch number
DATA	PATCH PARAMETER BLOCK in DATA format
\$F7	EOX

Soundprocess returns a "STATUS" message.

PATCH PARAMETER BLOCK - 70 bytes

<u>Byte offset</u>	<u>Parameters</u>
0 - 23	[61] - [84] of Oscillator Pair 1
24 - 47	[61] - [84] of Oscillator Pair 2
48 - 61	[85] - [98]
62 - 69	8 ASCII characters for PATCH NAME

"Configuration Parameters"

READ or WRITE the CONFIGURATION
PARAMETER BLOCK. Parameters [31] - [53]
form the DATA block.

CODE: \$05 / \$45

READ format:

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$05 Read CONFIGURATION PARAMETER BLOCK
\$00
\$F7 EOX

Soundprocess returns either a WRITE ***"Configuration Parameters"*** message or a ***"Status"*** message if there is an error.

WRITE format:

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$45 Write CONFIGURATION PARAMETER BLOCK
\$00
DATA CONFIGURATION PARAMETER BLOCK in DATA format
\$F7 EOX

Soundprocess returns a ***"STATUS"*** message

CONFIGURATION PARAMETER BLOCK - 23 bytes

<u>Byte offset</u>	<u>Parameters</u>
00 - 22	[31] - [53]

"Program Parameters"

READ or WRITE any of the 32 PROGRAM
PARAMETER BLOCKS. Parameters [00] - [11]
of each Program form the DATA block.

CODE: \$06 / \$46

READ format:

\$F0 System Exclusive
\$0F Ensoniq code
\$7F Soundprocess code
\$06 Read PROGRAM PARAMETER BLOCK
1-32 Program number
\$F7 EOX

Soundprocess returns either a WRITE ***"Program Parameters"*** message or a ***"Status"*** message if there

is an error.

Write format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$46	Write PROGRAM PARAMETER BLOCK
1-32	Program number
DATA	PROGRAM PARAMETER BLOCK in DATA format
\$F7	EOX

Soundprocess returns with a *"STATUS"* message.

PROGRAM PARAMETER BLOCK - 17 bytes

<u>Byte offset</u>	<u>Parameters</u>
00 - 11	[00] - [11]

"Wavesample Parameters"

READ or WRITE any of the 16 WAVESAMPLE PARAMETER BLOCKS. Parameters [56] - [58] of each Wavesample form the DATA block.

CODE: \$07 / \$47

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$07	Read WAVESAMPLE PARAMETER BLOCK
1-16	Wavesample number
\$F7	EOX

Soundprocess returns either a WRITE *"Wavesample Parameters"* message or a *"Status"* message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$47	Write WAVESAMPLE PARAMETER BLOCK
1-16	Wavesample number
DATA	WAVESAMPLE PARAMETER BLOCK in DATA format
\$F7	EOX

Soundprocess returns a *"STATUS"* message.

WAVESAMPLE PARAMETER BLOCK - 3 bytes

<u>Byte offset</u>	<u>Parameters</u>
00 - 02	[56] - [58]

"Wave Parameter"

READ or WRITE any of the 56 WAVE
PARAMETER BLOCKS. Parameters [18] - [30]
of each Wave form the DATA block.

CODE: \$08 / \$48

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$08	Read WAVE PARAMETER BLOCK
17-72	Wave number
\$F7	EOX

Soundprocess returns either a WRITE *"Wave Parameters"* message or a *"Status"* message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$48	Write WAVE PARAMETER BLOCK
17-72	Wave number
DATA	WAVE PARAMETER BLOCK in DATA format
\$F7	EOX

Soundprocess returns a *"STATUS"* message.

WAVE PARAMETER BLOCK - 12 bytes

<u>Byte offset</u>	<u>Parameters</u>
00 - 11	[18] - [30]

"Wavesample"

READ or WRITE any of the 16 WAVESAMPLES.

CODE: \$09 / \$49

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code

\$09 Read WAVESAMPLE
 1-16 Wavesample number
 \$F7 EOX

Soundprocess returns either a WRITE "*Wavesample*" message or a "*Status*" message if there is an error.

WRITE format:

\$F0 System Exclusive
 \$0F Ensoniq code
 \$7F Soundprocess code
 \$49 Write WAVESAMPLE
 1-16 Wavesample number
 DATA WAVESAMPLE in DATA format
 \$F7 EOX

Soundprocess returns a "*STATUS*" message.

WAVESAMPLE

Byte offset	Description
00 - 4095	Wavesamples 1-8
00 - 8191	Wavesamples 9-16

Valid values for the data in Wavesamples are \$01 - \$FF. A value of \$80 will produce zero volts at the output. Values of \$00 are normally avoided since they used as loop markers in the Mirage hardware which cause the oscillators to stop.

"Wave"

READ or WRITE any of the 56 WAVES.

CODE: \$0A / \$4A

READ format:

\$F0 System Exclusive
 \$0F Ensoniq code
 \$7F Soundprocess code
 \$0A Read WAVE
 17-72 Wave number
 \$F7 EOX

Soundprocess returns either a WRITE "*Wave*" message or a "*Status*" message if there is an error.

WRITE format:

\$F0 System Exclusive
 \$0F Ensoniq code
 \$7F Soundprocess code
 \$4A Write WAVE
 17-72 Wave number
 DATA WAVE in DATA format
 \$F7 EOX

MIDI SPECIFICATIONS

Soundprocess returns a "STATUS" message.

WAVE - 256 bytes

Valid values for waves are \$01 - \$FF. A value of \$80 will produce zero volts at the output. Values of \$00 are used as loop markers in the Mirage hardware which cause the oscillators to stop.

This message handles only single pages at a time. If a Wave is larger than one page, multiple "*Wave*" Dump messages should be made. For example, if Wave 21 is four pages long, the external computer should dump Waves 21,22,23,and 24 using four consecutive "*Wave*" Dump messages.

"Keyboard Tuning Table"

READ or WRITE the KEYBOARD TUNING TABLE.

CODE: \$0B / \$4B

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$0B	Read KEYBOARD TUNING TABLE
\$00	
\$F7	EOX

Soundprocess returns either a WRITE "*Keyboard Tuning Table*" message or a "*Status*" message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$4B	Write KEYBOARD TUNING TABLE
\$00	
DATA	KEYBOARD TUNING TABLE in DATA format
\$F7	EOX

Soundprocess returns a "*STATUS*" message.

KEYBOARD TUNING TABLE

264 bytes of coarse (00-09) and fine (\$00-\$FF) note tuning pairs.

"Envelope Time and Slope Table"

CODE: \$0C / \$4C

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$0C	Read ENVELOPE TIME AND SLOPE TABLE
\$00	
\$F7	EOX

Soundprocess returns either a WRITE *"Envelope Time and Slope Table"* message or a *"Status"* message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$4C	Write ENVELOPE TIME AND SLOPE TABLE
\$00	
DATA	ENVELOPE TIME AND SLOPE TABLE in DATA format (128 bytes)
\$F7	EOX

Soundprocess returns a *"STATUS"* message.

"Default Patch Parameters"

CODE: \$0D / \$4D

READ or WRITE the DEFAULT PATCH PARAMETER BLOCK.
Parameters [61]-[84] of both Oscillator
Pairs and Parameters [85]-[98] form the
DATA block.

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$0D	Read DEFAULT PATCH PARAMETER BLOCK
1-48	Patch number
\$F7	EOX

Soundprocess returns either a WRITE *"Default Patch Parameters"* message or a *"STATUS"* message if there is an error.

WRITE format:

\$F0	System Exclusive
------	------------------

\$0F	Ensoniq code
\$7F	Soundprocess code
\$4D	Write DEFAULT PATCH PARAMETER BLOCK
1-48	Patch number
DATA	DEFAULT PATCH PARAMETER BLOCK in DATA format
\$F7	EOX

Soundprocess returns a "STATUS" message.

DEFAULT PATCH PARAMETER BLOCK - 62 bytes

<u>Byte offset</u>	<u>Parameters</u>
0 - 23	[61] - [84] of Oscillator Pair 1
24 - 47	[61] - [84] of Oscillator Pair 2
48 - 61	[85] - [98]

"LFO Table"

CODE: \$0E/\$4E

READ format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$0E	Read LFO TABLE
\$00	
\$F7	EOX

Soundprocess returns either a WRITE "LFO TABLE" message or a "Status" message if there is an error.

WRITE format:

\$F0	System Exclusive
\$0F	Ensoniq code
\$7F	Soundprocess code
\$4E	Write LFO TABLE
\$00	
DATA	LFO TABLE in DATA format (256 bytes)
\$F7	EOX

Soundprocess returns a "STATUS" message.