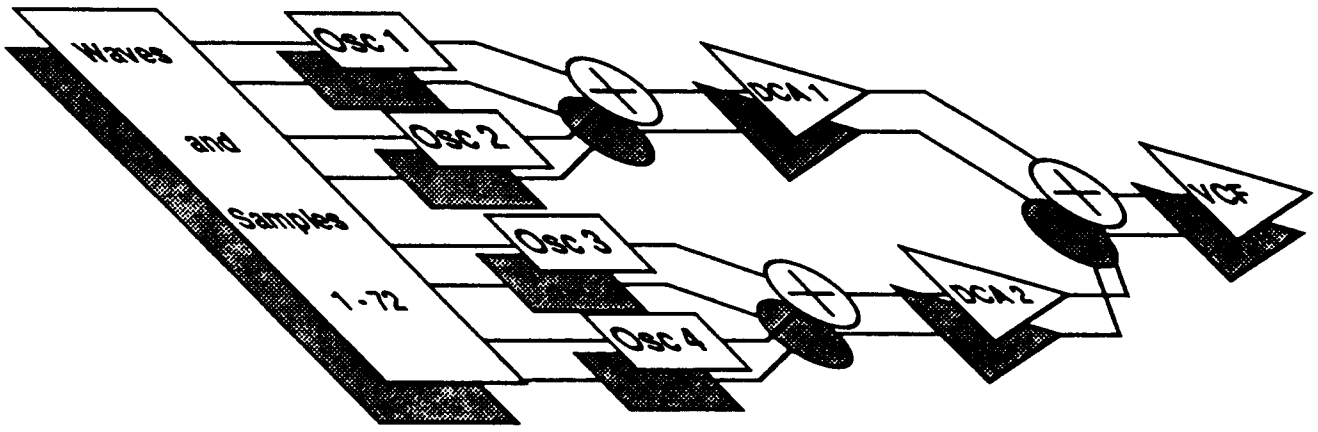
Soundprocess

Tutorial



By Alan Hamer

Triton

Soundprocess Tutorial

by Alan Hamer

INTRODUCTION

Welcome. You have joined the increasing number of Mirage owners who are utilizing the advantage provided by their software based instrument. Although most contemporary keyboards are microprocessor based, the ability to introduce a new set of internal instructions produces features that make this instrument improve with age.

The purpose of this volume is to speed and facilitate the use of SOUNDPROCESS. Most software includes a tutorial, but in this case the need is heightened by two factors; the dramatic increase of features and the limited degree of visual feedback from the Mirage's LED display.

If your head nods in agreement, then with similar conviction follow this guide's step-by-step instructions as the different functions of SOUNDPROCESS are illustrated. The content of each section builds upon the previous, so please progress sequentially through the text, as you experience getting the most out of SOUNDPROCESS.

Generally, it will be assumed that you are acquainted with the original Ensoniq Operating System (version 3.2), using a version of SOUNDPROCESS 1.2 or higher, and that you are seeking new sonic frontiers.

So before we begin the trip, plug in, boot up SOUNDPROCESS, and be sure to have a few diskettes formatted with the Ensoniq 3.2 OS (Operating System) and a MIDI cable handy.

MULTI-TIMBRAL OPERATION - LAYERING

SOUNDPROCESS transforms the Mirage into a multi-timbral instrument. As such, the role of MIDI to a Mirage owner changes, especially if it is your sole instrument. This first section is not about how to use SOUNDPROCESS, as much as how it can change the way the Mirage is used. Particularly important to MIDI neophytes will be the demonstration of MIDI concepts, but for all there should be a pleasing

adventure in sound.

The manual, by which is meant the book included in the SOUNDPROCESS package, provides clear instructions on "Getting Started". If you have not already done so, please read and try to verify the information in the following paragraphs; "Starting Soundprocess", "The Front Panel", "Soundprocess Key Summary", and "Selecting New Programs".

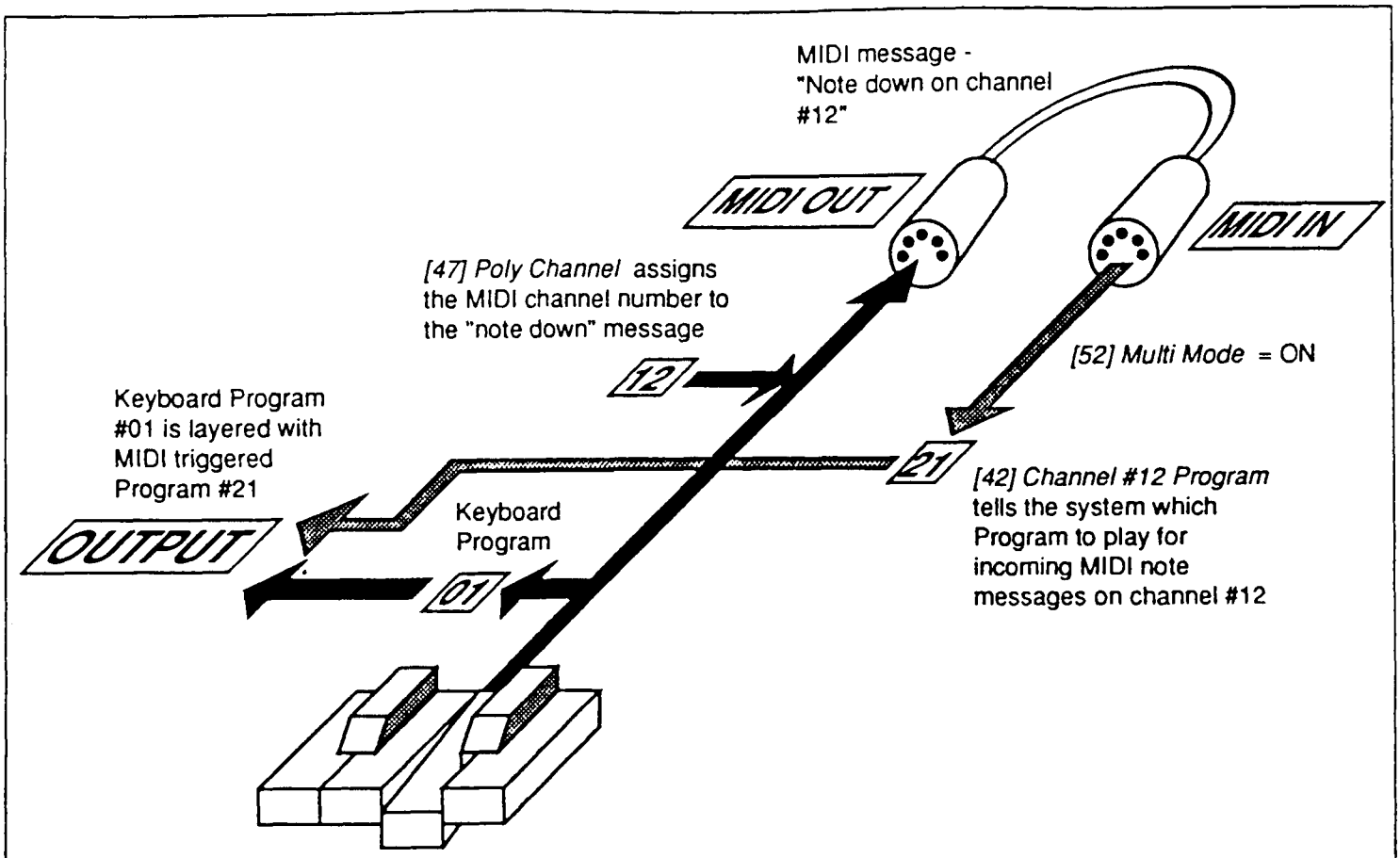
You should listen to all 32 keyboard programs on the system disk. Imagine these sounds layered on others, a total of eight oscillators for each key with four note polyphony. When multi mode is entered, the instrument will be able to produce both the sound indicated on the display and sounds triggered by MIDI messages, simultaneously.

If you have a keyboard Mirage, then take a MIDI cable and plug one end into the MIDI IN and the other end into the MIDI OUT. If you are using a rack mounted Mirage then the proper connections are less universally stated, but follow along and see if you can attain similar results (rack owners be not dismayed, though this discrimination occurs early, it does not apply in other parts of this tutorial and you are probably more accustomed to MIDI functions).

The numbers 01 should be showing in the LED indicating that program 01 is the sound heard. Press the parameter button. It will be 47. Increase the parameter value to 52 either by using the increment button or by entering 5 then 2. Now press the value button and change the setting to ON by pressing the increment button. We have just turned the machine into a multi- timbral instrument. From now on, the preceding operation will be notated as "set (52)=on". Multi-timbral, but to no advantage until we press the parameter button and set (31) (channel #1 program) to a value other than 01, so set parameter (31) to any value from 02 to 32.

Information, created as the keyboard is depressed, takes two routes. One part leaves the Mirage as a MIDI message traveling on the channel designated by Poly Channel, Parameter (47). The MIDI messages leave from the out port, go through the MIDI cable and come back to the MIDI in port. Then the Mirage responds as if the messages were coming from any external sorce. Simultaniously though, the Mirage also respondes to the keyboard via its usual internal routes, playing the sound designated by the program shown on the display. The sound triggered by MIDI is choosen by setting the configuration parameters. Open the manual to the Parameter Chart - Appendix A, this chart will become your faithful companion.

Let's test this MIDI arrangement. Set (53) (local mode)=off. The keyboard program has been eliminated and only the MIDI triggered sounds, once again in eight note polyphony, are heard. That's no fun, set (53)=on.



Layering Programs with Multi Mode

There is just one more article of faith to test and that is the role of parameter (47) (poly channel), the value which dictates the MIDI channel transmitted. Parameters (31) to (46) are used to establish a one to one relationship between channels and programs. Scroll through the values of parameter (47) and listen to the results.

Most of the time multi mode will be employed in conjunction with a sequencer. Although we have been using one MIDI channel, a sequencer can send MIDI information on all 16 channels. SOUNDPROCESS contains the provision to assign different programs to each channel which unlocks the Mirage's ability to play up to eight different sounds at the same time, a significant compositional advantage. There is one limitation to the system which is not documented in the manual. **Avoid assigning sequenced material to channel #1.** Internally, the microprocessor calls keyboard messages channel #1 and if it receives MIDI messages on channel #1 at nearly the same instant as you are playing, it interprets one of the messages to be erroneous.

Note that in multi mode ((52)=on), omni mode (48) is not in effect. When (52)=off, then (48)=on would cause the Mirage to respond to messages on any channel and if off, it will play notes sent on the channel selected by parameter (47) (poly channel), but in either case only the program selected on the display.

Please tend to get swept up in the sound combinations you come across, recording particularly striking settings. The layering which was done in this section is not intended to be regarded as the usual mode of operation, but in the course of gaining familiarity with the configuration parameters, one can not help being impressed at the ease with which SOUNDPROCESS produces interesting sound.

KEYSTROKE REVIEW:

Program SOUNDPROCESS starts in the program mode. The LED's decimal pointer is not lit. Programs can be changed, once in the program mode (if in another mode than press the "Seq Rec" button first), either by entering the number of the desired program with the keypad, or by scrolling through with the up and down buttons.

Parameter When in the parameter mode, the LED displays numbers in two digit decimal fashion (.XX). To enter parameter mode from program mode, first press the parameter button. To enter parameter mode from value mode, either use the parameter button or enter parameter numbers on the keypad. Like program mode, parameters are entered numerically or scrolled sequentially.

Value Values always refer to the parameter number just previously displayed. Values are displayed as they are in the original operating system (X.X). It is usually only appropriate to enter value mode from parameter mode, which is done by first pressing the value button. Values are changed both with the up and down buttons, and with the Quick Value buttons which instantly provide maximum, minimum, middle and original settings. The position of these helpful keys (referred to by the manual as Parameter Edit Keys) differs by model, so consult the manual. DO NOT CHANGE VALUES BY USING THE KEYPAD NUMBERS, this is a common source of error.

Stop This is the button which is labeled "Seq Play", now assigned to sending an all notes off message on all MIDI channels. Press this button to stop stuck notes from playing.

The preceding material centered upon the MIDI aspects of SOUNDPROCESS. MIDI provides an interface between the Mirage and the outside. For the rest of the tutorial the subject matter will turn inward. Coming sections will cover sound design, wave characteristics, additive synthesis, and moving data, among other subjects. But before we get too far, the procedure for keeping what we make will be demonstrated.

MAKING A DATA DISK - THE SAVE COMMAND

To *save* is to take all or part of the information in the machines memory and transfer it to a disk. SOUNDPROCESS data disks allow three banks of data to be assigned to a single diskette. The data includes all program parameters, all patch parameters, all the waveforms, and most of the configuration parameters. To save, please be forewarned, is not always a blessing, especially when the disk territory upon which new sounds were just deposited was once home to yesterdays killer sounds. Regard the save button as if it were blinking red. Keep the write protect feature of the diskette in the "open window" position until just before doing a save.

Now let's try an example. We will transfer the sounds which load with the system disk to another disk.

Reboot the systems disk by placing it in the drive (the assumption is that we will do a warm reboot from SOUNDPROCESS), press load (which means the transfer of data from the disk to machine memory, in this case the data is both operating system and sounds) then 0 on the keypad and enter. When the machine stops remove the systems disk and insert an empty but formatted diskette with the write protect slider in the "closed window" position, into the drive. Draw your finger steadily closer to the save button, press it. Chose bank 1 by pressing 1. Hit enter.

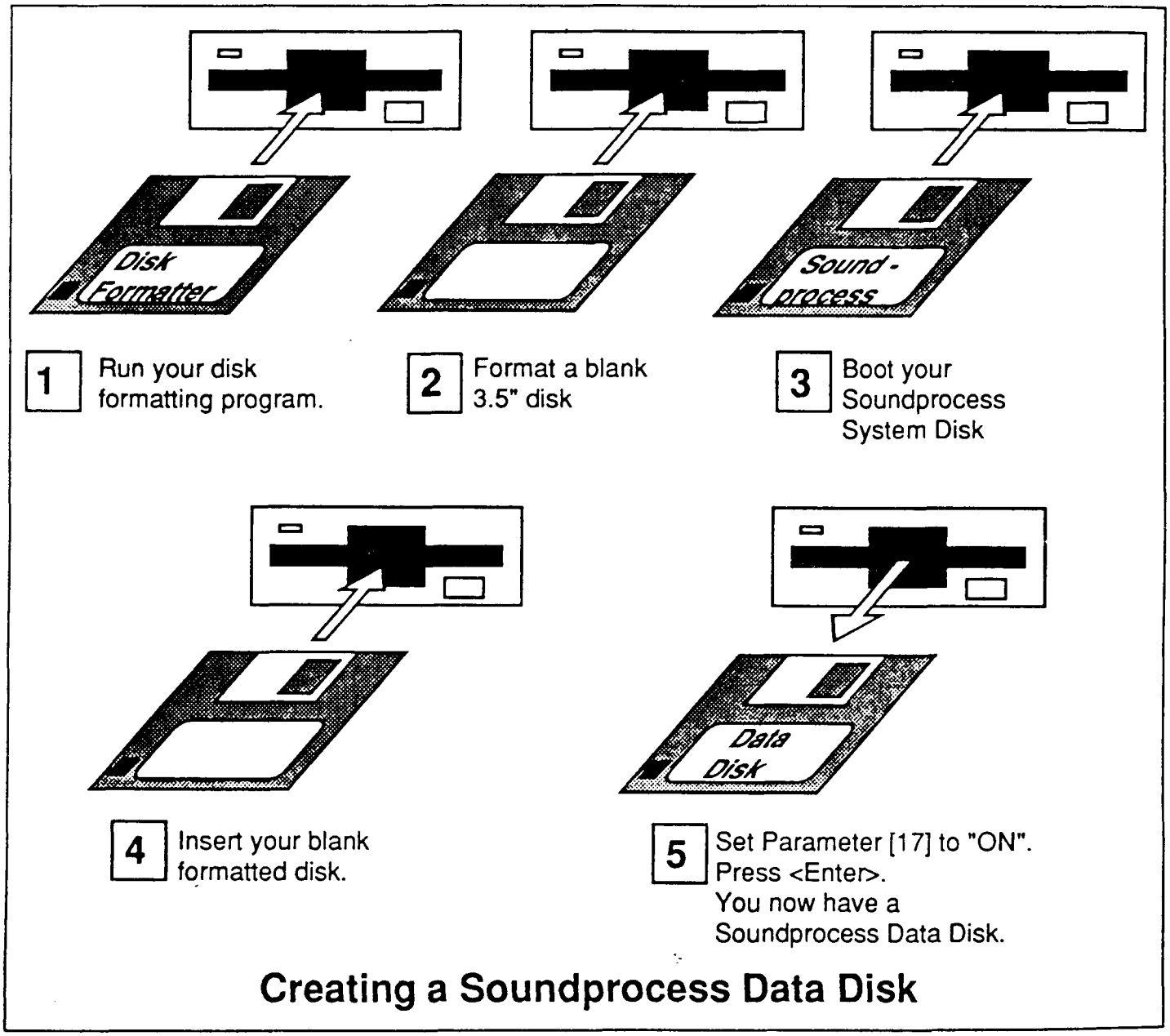
Pick the error message which best describes your display:

Sd- the system disk is in there

Pd- protected disk, check the hole in the disk's corner and pry the button into the window

ud- unformatted disk
dE- disk error, try a different disk
nS- this was the our aim, not a SOUNDPROCESS data disk

The empty though formatted disk must be further customized for it's role as a data disk. This step could have occurred prior to attempting a save. Press the parameter button and set (17) to the on position. Then press enter. After a brief spin, the disk is ready. With a purposeful gesture, now save the system disks sounds to bank #1. We will use this bank in the next section. Open the write protect window, safeguarding our data from accidental erasure. Label the disk, writing "SOUNDPROCESS TUTORIAL" at the top as a heading and below "Bank #1- sounds from system disk".



Creating a Soundprocess Data Disk

PROGRAM PARAMETERS - PATCHES

The manual thoroughly covers the subject of the arrangement of patches into programs through the use of the familiar "top key" system. If you have not already done so, please refer to those paragraphs in section 2 of the manual under the heading of "Program Parameters".

We will take a tour of the patches on bank #1 of our newly made disk of sounds from the systems disk. If it is not loaded then please do so. Program 01 shows on the display. Hit the parameter button and select parameter (00). This parameter is the top key setting for split #1 and the value reading of 7F tells us that what ever sound it is, it covers the entire keyboard range. Find the value of parameter (01), patch #1. The number in the display should be 24, the bottom line is that when we call up program 01 of the system disk sounds, we are listening to patch 24. If program 01 had more than one patch assigned to the keyboard, ie. splits, then top key #1 would have some lower value and there would be values for (02) and (03) as well.

Let's listen to all the patches on the disk. Set (01)=01 by pressing the appropriate Quick Value button. Do you recognize this sound? Increment the value of parameter (01) from 01 to 48 by one. Each time the patch will fill the entire keyboard. After you have gone through all the patches, press the appropriate Quick Value button to restore parameter (01) to it's initial value. This exercise was intended to illustrate that patches are the raw material which can be assembled into programs according to your needs, and that patches, not programs, are the basic unit of sound.

PATCH SELECT

Ideally, when a new program is chosen, the parameters which modify the sound would be there at the touch of the parameter button. But a program can be composed of more than one patch. So an ambiguous situation would exist; to which portion of the keyboard would the parameters apply. SOUNDPROCESS has a definite procedure for parameter selection. The patch number of the patch to be edited must first be entered as the value of parameter (54), patch select. For example, to modify the sound of program 01, we have already found that patch 24 is assigned to the entire keyboard for program 01, parameter (54) (patch select) should be set to 24.

Re-load the system disk sounds from bank #1 of your "tutorial" disk, set the program to 01 and open the manual to Appendix A - The Parameter Chart. Set (54)=24. We are now prepared to find out what makes this patch sound as it

does.

As we look down the parameter list, the amplitude envelope parameters come first, then tuning, and then oscillators A and B. Parameters (73) and (74) form two oscillator pairs, either pair 1 or 2 depending on the value of parameter (60) (Oscillator Pair Select). Set (60)=1, if it is not. Then we will enter the more fundamental level of sound formation, the waveforms. Verify that the present value of parameter (73) (waveform A) is 15. Now change this value and listen to the results, returning back to 15.

The part of the sound that stays the same must be oscillator B and the second pair of oscillators, A and B. The fact that there are four oscillators and 72 waveforms to chose from will give you an increased sense of the power of SOUNDPROCESS now that you have experienced the effect which varying only one variable achieves.

There is one more aspect to the way in which patch select works which must be clarified. Patch selection through the value of parameter (54), provides access to the criteria for editing a sound, but it does not change the sound assignment to the keyboard; that is the job of the program parameters. In practical terms then, when modifying a sound while listening to the changes, it is necessary to both set parameter (54) and adjust the program number to the desired sound.

SUMMARY: To Edit Patch Parameters

1. Find Patch Value
   a. When the sound to be edited is presently available on the keyboard, calculate the top key of the target sound (the 61 notes on the Mirage run from 24 to 60 in hex notation necessary because SOUNDPROCESS uses 128 notes), match this value to the values of parameters (00), (02), (04), (06), (08), (10), and find the value of the corresponding patch #.

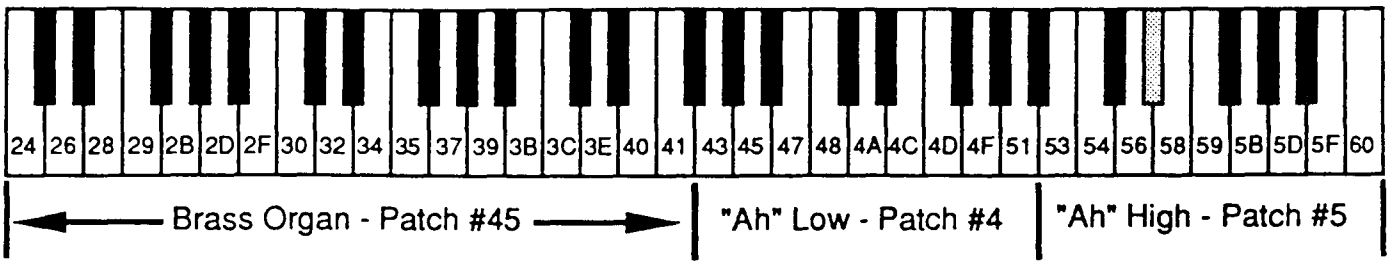
   b. Exclusive of program settings, set (00) to 7F, and scroll through parameter (01) until target sound is found. The value of parameter (01) is the patch value.

2. Enter patch value as value of parameter (54), patch select. This causes all patch parameters to reflect values which shape the target sound.

3. Before extensive editing occurs, it is prudent to test the patch assignment by changing a parameter with an obvious effect, such as

parameter (98), patch volume. This parameter is used to balance sounds in multi mode.

ATTEMPTING MODIFICATION OF A SOUND WITHOUT FIRST ENTERING THE PATCH NUMBER AS THE VALUE OF PARAMETER (54) (PATCH SELECT) IS A COMMON ERROR.



Editing Patches

Three Patches are assigned to play Program #4 on the System Disk: a brass organ in the low octaves and a multi-sampled voice on the high end. Looking at the Program Parameters of Program #4 (remember that parameters [00] - [11] reflect the current active Program) shows the following configuration:

	Top Key	Patch Assignment	
Split #1	[00] = 41	[01] = 45	Brass Organ
Split #2	[02] = 52	[03] = 04	"Ah" Low
Split #3	[04] = 7F	[05] = 05	"Ah" High
Split #4	[06] = --	[07] = --	
Split #5	[08] = --	[09] = --	
Split #6	[10] = --	[11] = --	

Before you can edit a sound, you must first find the Patch which defines the desired sound. In this example we want to edit the sound that plays E flat on the top octave (key #57). Since 57 falls inside the range of Split #3 (53-7F), Patch #5 must define that sound. After setting [54] Patch Select = 05, Patch Parameters [61]-[98] now contain the values which define the "Ah" high sound.

PATCH CONSTRUCTION

The subject of how to put together a sound is within the realms of both art and science. This section will cover those SOUNDPROCESS parameters which

most influence sound output. That is the "science" part. Please remember that the oscillators will interact in a synergetic manner, by which is meant; the whole is greater than the sum of it's parts. That is the "art".

Now, back to our dissection of the sound known alternately as program 01 and patch 24. Set (54)=24. We know that oscillator A of pair 1 uses wave 15. Find the value of parameter (74) and then change the value of parameter (60) to 2 (chose the second pair), and find the values of that set of (73) and (74). The results are as follows:

(73)	(pair 1)=15	(pair 2)=16
(74)	(pair 1)=15	(pair 2)=16

This means that the two oscillators of pair 1 use wave 15 and the two oscillators of pair 2 use wave 16 to produce the sound of patch 24. Each oscillator has an independent volume control. Before we use these controls, check that we are in the second oscillator pair, (60)=2. Set the volume of oscillator B, (81), to 00. There goes the sustain. Then set the volume of oscillator A, (80), to 00, and now only pair 1 can be heard. Further, change parameter (60) to 01, bringing us back to pair 1, and set (81)=00, leaving only oscillator A audible. With the opportunity to audition a waveform individually, we can listen to each of the 16 wavesamples in the wavetable. Change the value of parameter (73) from 01 to 16, and you can listen to the various wavesamples which were loaded with this bank.

Return the value of parameter (73) to 15. Return the volume to oscillator B by setting (81) to 63. The sustain portion of the tone is restored. That is because waveform 15 which is the value of parameter (74) (oscillator B) is assigned the role of being the loop.

The term "assigned" was used because the function of the oscillator algorithm, parameter (75), is to structure the relationship between waveforms A and B. Check the value of parameter (75) and then consult the chart in section 3 of the manual which graphically illustrates these differing relationships. Also note that some wave structures indicate the wave size hence wave values which are suitable as oscillator A or B. Algorithm 4 should have a familiar appearance for it is the usual scheme for attaining sustain beyond the duration of the sample. The sustain, it should be noted does not last forever due to the intervention of the amplitude envelope, parameters (61) to (68).

If you look at the diagram while changing parameter (75) from 1 through 6, the differences in the use and roles of the waveforms with each algorithm becomes apparent.

ALGORITHMS 7-11, TIME RE-TRIGGERED WAVESAMPLES

Here is the present status of the patch parameters, identical to the way it was left from above, but recapped just to be sure. Program 01, a.k.a. patch 24, is the sound in surgery. Patch 24 is selected ((54)=24), with the oscillators of pair 2 blanked out and the volume all the way up (not on your amp, but (80) and (81)=63) on pair one. We will continue to explore the subject of algorithms. A unique feature is incorporated into algorithms 7 through 11.

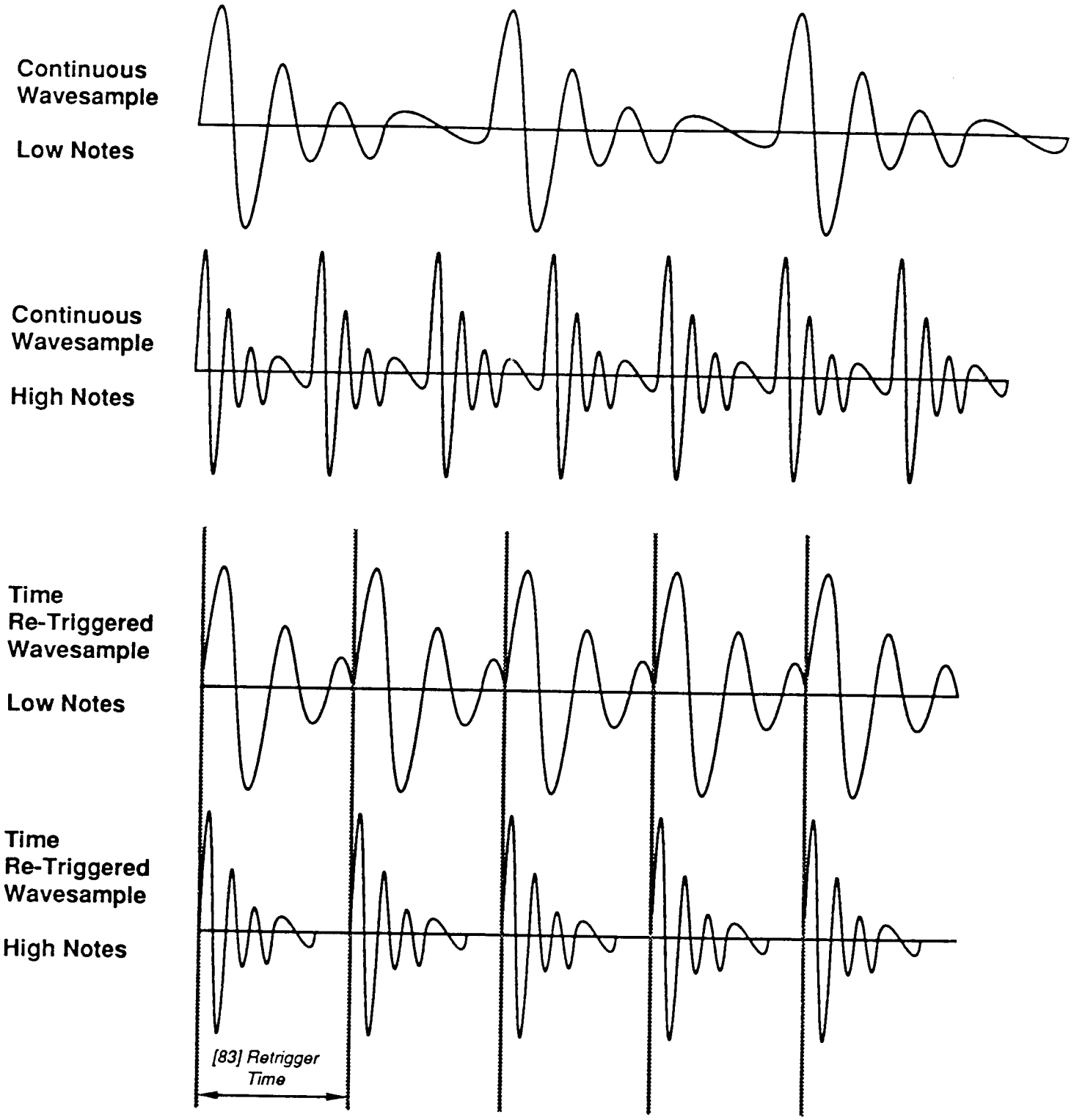
Set parameter (75) (Oscillator Algorithm) to algorithm 1. Diagrammatically observe that both oscillators continually repeat themselves. Listen to the effect of this structure on both high and low notes. This is a very common signature which sampling instruments impart to their sounds. Now change parameter (75) to a value of 9. This structure is the same as #1, except that now the repetition rate of the waves will be controlled rather than a function of frequency ie. high notes quickly and low notes slowly. Because the default setting of the time parameter (83) is 0, the waves buzz. As we increment the value of parameter (83), the promise declared above unfolds. When the value of 20 has been reached, again compare high and low notes, try playing them at the same time as well. Parameter (84) (Repeat Counter) allows control over the number of times the waveform repeats. A value of 00 means that no count is taken. Try a value of 3. Then explore the variations which the other time re-triggered algorithms provide. The differences between algorithms will be more strikingly heard if oscillators A and B can be differentiated, say by changing the value of parameter (74) to 16.

Applications of the time re-triggering feature ranges from mandolin and guitar strums to poly-rhythms setting pair 1 against pair 2. More subtle effects can also be produced when one pair has a traditional sound and the other pair beats softly in the background. Please note that the waves which can be assigned to a re-triggered algorithm are 1 through 16, the wavesamples.

OF WAVES AND WAVESAMPLES

It is now time to get more intimately acquainted with the workhorses of SOUNDPROCESS, the wavesamples and waves. The advantage of instant access to 48 unique patches is afforded by the preloaded, user definable selection of up to 72 waves. The techniques involved in "loading your own" will be covered later in this volume. When the load button is used to put a bank into the machine's memory, the array of up to 72 waves, known as the wavetable, is among the information which is transferred from the disk.

As a point of perspective, recall that each keyboard half of the original Mirage is divided into 256 pages. SOUNDPROCESS partitions it's memory into pieces of specific sizes. Wavesamples are the larger, more complex waves, num-



Comparing Continuous and Re-triggered Algorithms

Continuous wavesamples play their full length before repeating so that low notes repeat the wavesample slowly and high notes repeat at increasingly higher rates. Re-triggered wavesamples repeat at a fixed rate across the keyboard. Notice that with low notes the wavesample is re-triggered before it's end and with high high notes a pause is inserted at the end of the wavesample untill the next trigger.

bered 1 - 16. Waves 1 through 8 are 16 pages long and waves 9 through 16 are 32 pages long. Waves 17 to 72 can be either 1, 2, or 4 pages long as dictated by the rules found in the manual's section 4. Owners of samplers are naturally biased in the belief that more is better, particularly in regard to memory. Small waves could tend to be regarded as informational light weights. But small waves can offer some significant relief from two of the issues common to samplers; looping and aliasing, both of which rise in importance during the sustain portion of a sound.

The familiar single page loop is a wave based upon a sample. Also familiar are synth waves (square, sine, sawtooth etc.), the output of analog and some digital synthesizers. So there is certainly precedent for the musical use of single cycle waves. SOUNDPROCESS provides a choice of waves to be the sustained tone for samples which are themselves unsuitable for cycling due to either tonal or frequency fluctuation, undesirable characteristics for a loop. And SOUNDPROCESS usés the Mirage's microprocessor to generate waves mathematically (details in the section on additive synthesis) which are free from the digital noise inherent in sampled sounds so they require less filtering, yielding a clearer sound.

Wavesamples, like the samples played on the Mirage are harmonically rich and can vary in pitch and tone during their duration. Wavesamples can be assigned loop start (56) and end (57) points. Loop settings, including loop end fine (58), are incorporated in the code of the wavesamples themselves. And although a premature consideration which will have more meaning in the section on loading MASOS waves, the samples which are transferred to SOUNDPROCESS should not include their loop points as these are difficult to remove from the wavesample data and might preclude their usefulness. Loops, just as in the original operating system, must be at least one page long ((56)=(57),(58)=FF), and preferably a length which is a power of 2 (1, 2, 4, 8, etc.).

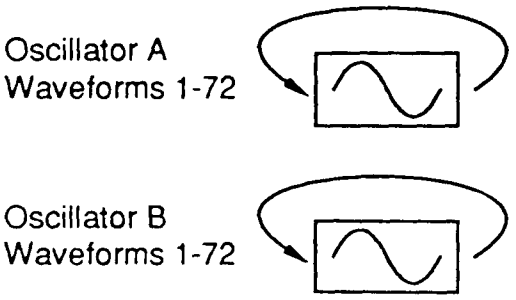
To inspect the loop settings for wavesample 15, one of the two looped waves from patch 24 of the systems disk, set parameter (55) (wavesample select) to 15 and then check the values of parameters (56), (57), and (58). Most changes to the settings will not be beneficial. Wavesamples 1 through 16 can be access in a similar manner. Looping is greatly enhanced through the use of visual editing software, in lieu of which luck and patience are valuable commodities.

Wavesamples in re-triggered and one shot modes, as defined by the manual in section 3, interpret loop end as the place wave playback ceases, effectively trimming the wave. In the continuous mode, a loop end marker placed anywhere except for the very end of the wave causes playback to stop. Looping is supported in algorithm 4, but it should be noted, one of the advantages which SOUNDPROCESS provides is an alternative to loops.

ALGORITHM STRATEGIES

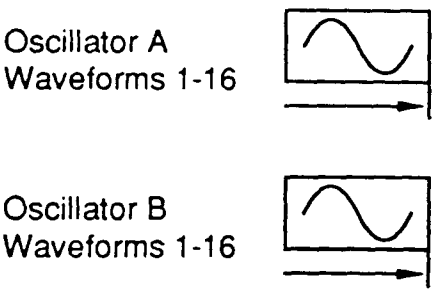
SOUNDPROCESS wavesamples are defined as 16 or 32 page waves. But waves of even larger size can be incorporated into this operating system by using algorithm 5. This structure allows two pieces of a longer sample to be played consecutively.

Algorithm 1 has two different uses. As it supports the continuous play of two waves of any size it can be used to create thick synth textures. The synth string sound of program 21 on the systems disk employs algorithm 1 for both pairs. Further, the same wave, 63, is assigned to all four oscillators, with slightly unique tunings for each.



Algorithm #1

The other use of algorithm 1 is to make one shot sounds with waves. One-shot mode (offered in algorithms 2, 3, and 5) is specifically for wavesamples as opposed to waves, so algorithm 1 contains a provision for non-continuous play of waves. An example of this technique can be found on program 05, the lowest split, kick drum. Examination of the program parameters reveals that this sound is patch 39. Enter this value in parameter (54), patch select. By looking at the values of oscillator A and B of pairs 1 and 2, we find that both oscillators of pair 1 use wave 65 and pair 2 is turned off ((60)=2,(80)=(81)=00). It seems strange though that while parameter (75) indicates algorithm 1, a mode of continuous wave play, the sound of the drum stops after one repetition. As loop markers cause continuous mode to end this must be the answer, but not so nearly obvious is the source of the marker as loops are set only for wavesamples according to the manual. The answer is that the wave was edited outside of SOUNDPROCESS and then inserted. The advantage of this method, though not technically possible for all, is provision of a drum sound in just four pages, usable by all.

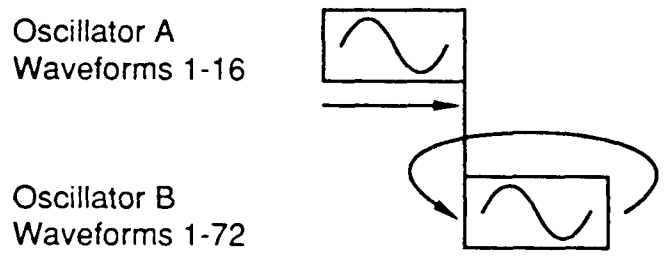


Algorithm #3

Algorithm 2 is a combination of algorithms 1 and 3, so we will focus on algorithm 3 next. Algorithm 3 is made of two one- shot wavesamples. This structure is particularly well suited to percussion and special effects. Most of the other drum sounds on the systems disk program 05 employ this algorithm. Another use of this structure is to provide a sharp attack when coupled with a sustain sound in the other oscillator pair. An example of this technique can be heard on

program 30 which is patch 30. Pair 1 uses algorithm 3 to play the bottle blower attack of wavesample 9 with both oscillators. Meanwhile, over in pair 2, the sustain portion of the patch is produced by using the wooden flute sound of wave 29 in the continuous mode of algorithm 1. Please recall the claim that SOUNDPROCESS provides an alternative to looping, as now one technique has been illustrated.

Algorithm 4 supports traditional looping to produce sustain based upon the tail end of the wavesample. For looping to be successful, the wavesamples should completely fill the space allocated by the wave table. We have previously examined one example of a sound which uses looping, patch 24 from the systems disk. In another case, algorithm 4 is used on only half of the sound in patch 46, a flute, the upper split of program 15 from the system disk. Wavesample 12 is looped in pair 1 while over in pair 2, wave 23 plays continuously with algorithm 1. The background sound provided by pair 2 can be used to mask looping problems though in this case that was not the intent. There are structural reasons to avoid the looping algorithm. SOUNDPROCESS utilizes relatively short samples. One of the assumptions of traditional looping is that the sound should stabilize in tone and pitch prior to use as a sustain. Check that wavesamples are of sufficient duration to attain equilibrium.



Algorithm #6

Algorithm 6 provides the capacity to make sound in a very additive way (not to be confused with additive synthesis of waves, a later topic). This format allows us to simply paste together sections of sound. Attacks are limited to wavesamples and sustains can be drawn from any wave. Nearly all the sounds on bank #1 of Sound Disk 1 use algorithm 6 in tandem with algorithm 1. Most of the waveforms on Sound Disk 1 have been enhanced using wave shaping software. Wavesamples used in oscillator A have been compressed to communicate detail during the attack, and waves used by oscillator B are of exactly one cycle and further processed to eliminate digital noise.

SUMMARY: Sound Construction

Several subjects are covered in either the SOUNDPROCESS or Mirage manuals and are not mentioned in detail here for the sake of brevity though their importance is not discounted. Those areas include: amplitude and filter envelopes, keyboard tracking, velocity sensitivity, and modulation. A suggested approach

to sound creation is to pick an existing patch with the right envelope settings and modify the choices of waves and algorithm. Another way to proceed involves beginning with a clean slate and putting all the pieces together individually. Parameter (12), set patch default, creates a "vanilla" environment which acts as a starting point and can be modified at will. To use this feature, select a patch value, enter it in parameter (54) (patch select), set (12)= on and then press enter.

Often issued, but rarely heeded, advise to programmers is to frequently save your work to disk ("I just had it, and now it's gone").

MOVING WAVES AND PATCHES TO CUSTOM DISKS

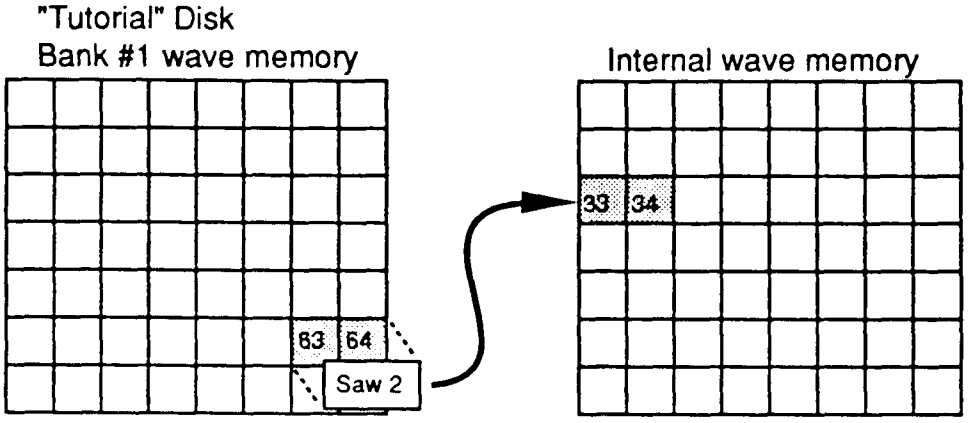
In a short time, you will be in the position of wanting some sounds from different disks to all be on the same bank. We will step through the transfer of both wave and patch data (for what good is one without the other) from the copy of the systems disk, previously made, to a blank bank on the same disk. These procedures are differentiated from the "save" command in that we will transfer specific data into the machines memory as opposed to saving a whole bank onto disk. Until we keep our work though, keep the disks protect slider in the open window position.

We will transfer the synth-string sound of program 21 of bank #1- The Systems Disk Sounds, to bank #3 on the same disk. Inspection of program 21 reveals that the program is comprised of only one patch, patch 21. Patch 21 uses wave 63, a two page wave, in all four oscillators. The destination in the wavetable must be a space which allows a two page wave. After consulting the wave type/size table in section 4 of the manual, wave 33 is selected as the new home for wave 63 as there is enough room for a two page wave. If an even numbered location were selected, the wave would not perform reliably. Section 4 of the manual will be the reference for the disk function codes, necessary to communicate which data is transferred. Look at the paragraphs under the heading, Disk Functions, found in Section 4.

Our first step is to make a silent bank. A bank on a sound disk without any information cannot be loaded and is meet with an "nF" (not found) error message. The following steps create a silent disk, a tedious process which only needs to be done once, but can be used like a blank canvas repeatedly. First load the system disk sounds from bank #1. Then beginning with program 01 set the top key for all patches to 00. Check the value of (00), and use the Quick Value key to set it to the minimum value of 00. If the original value of (00) was not 7F then go se-

quentially to (02), (04), (06), (08) and (10) setting each to 00. When that program is silent press the program button (labeled "Seq Rec"), increase the program number by one and repeat the process. Program 13 is tricky, both (00) and (02)=7F. When you are done, scroll through all the programs, listening for all to be silent. Save this to bank #2, and label the diskette "Bank #2-Loadable Blank".

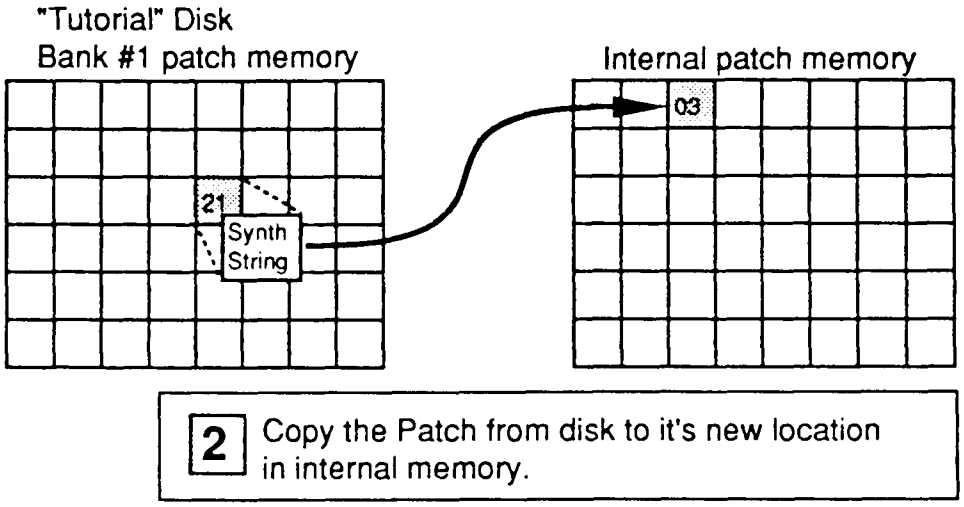
The disk with our copy of the system disk sounds should be in the drive and the blank bank should be in the machine's memory. Select the disk function type for a wave load by setting parameter (13) to 01. The sequence of steps is important because changes to parameter (13) initiates default settings to other parameters. Parameter (14) identifies the bank in which the source wave resides. Set (14) (disk bank) to 01. Set (15) (source) to 63, specifying where on bank #1 the wave can be found. If we were loading a wavesample, the number of the wave would indicate the size (number of pages) to transfer, but in the land of waves 17 -72 much variation is allowed. Consequently two steps are needed to transfer this two page wave, first just page 63 will be assigned to 33 (bank is decided when we save to disk). Set (16) (destination) to 33. Press enter. To complete the transfer, follow the same procedure, (13) and (14) keep the same values but change (15) to 64 and (16) to 34 and press enter. The sawtooth wave is now known as 33 and 34.



You might question how it is known that wave 63 is a two page wave. The SOUNDPROCESS Waveform Map shows us that "saw 2" occupies two pages, but there is an other method. Set (18) to 63. Parameter (18) is called wave select and is used to obtain information about specified waves. Find the value of parameter (20), wave type/size, and confirm using the table from the manual that the value of 03 indicates a two page wave loaded from MASOS. The term MASOS is used in

a generic manner meaning any Ensoniq OS 3.2 sound disk.

The wave does not do us any good until we supply the patch parameters. Return to parameter (13), disk function type, and enter a value of 02, indicating a patch transfer. Again the bank and number of the source need to be specified. Set (14)=01 and (15)=21. The new patch number is 3, so set (16)=03, then press enter. The original patch 03, a brass sweep of program 29 has been replaced though still intact on bank #1 of our disk.

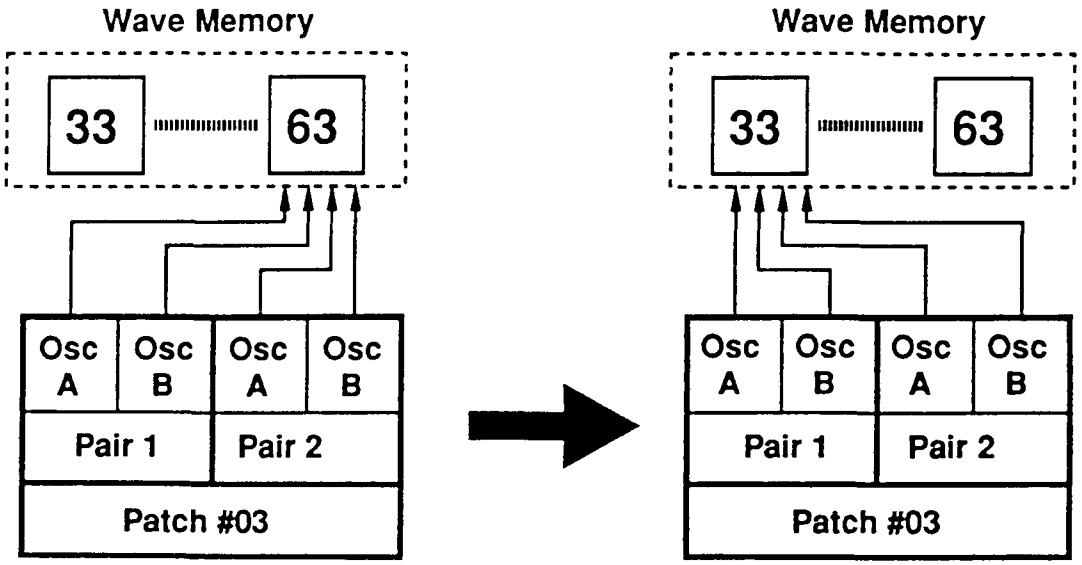


The patch data needs to be edited to reflect the changed location of the sawtooth wave. The word "edit" should trigger the strong desire to set patch select to 03 (set (54)=03). Let's begin with pair 1, so set parameter (60) to 01. Give both parameter (73) and (74) a value of 33, telling oscillators A and B where to find the wave. Then increment (60) to 02, to gain access to the other oscillator pair and again set parameters (73) and (74) to 33.

We will assign patch 3 to program 01. Enter program mode by pressing the button labeled "Seq Rec", and change the setting to 01. Set (00) to 7F and (01) to 03. Play a couple of notes, this patch probably never sounded so good. This could be the beginning of your first custom SOUNDPROCESS disk, so now save it to bank #3 and label the diskette, "Bank #3- Custom Sounds".

Organization is the key to successful disk arrangement. Use worksheets to keep track of waves, parameters, patches, and programs; sources and destinations.

During the transfer process, it is possible to stray. If you happened to see these error messages, here is what they mean:



3 Since the transfered waveforms will usually have a different location than the original on disk, change the oscillator waveform assignments ([73] and [74] of each pair) to their new locations.

Er- Source/Destination match error. A wave can not be sent to locations 01 to 16 in the wavetable.

hp- Can't load MASOS wavesample onto a harmonicallydefined wave.

SUMMARY: PATCH/WAVE TRANSFER

The target bank is loaded into the machine. The disk with the source patch/wave is inserted into the drive. Parameters (13)-(16) are given values which tell the Mirage:

(13) wave/patch/sample transfer
(14) bank of the source
(15) numerical location of the source
(16) destination on target bank

A change in (13) resets the rest of the parameters so start with (13). For the desired transfer to occur, parameters (13) through (16) must be on the display and then the enter button must be pressed. Save the resulting bank to disk.

LOADING A WAVESAMPLE FROM A 3.2 OS DISK (MASOS)

Most Mirage owners have disks and disks of samples done by Ensoniq, third party developers, or themselves. This section demonstrates the way that 16 or 32 pages of these samples are loaded into the multi-timbral, instantly accessible, highly creative environment of SOUNDPROCESS.

Load bank #3 of the disk made above, containing the synth- string sound. Place Ensoniq sound disk 102, the one with marimba/electric piano/organ/clav on upper bank 3, into the disk drive (check that the write protect window on the disk is open). We will add the electric piano to our custom disk.

The process, much like before will be to load the wave information, assign the patch to a program, and then adjust the sound. We start by setting (13)=03, the target sound is a MASOS wavesample, (14)=06, located in the upper part of the third bank, (15)=80, the electric piano begins on page 80 (what, Ensoniq didn't provide a waveform map?), and (16)=13 previously a bell wavesample. Press enter to make it count.

At this point a structure is needed to use the waveform. One could be designed from the ground up, or, as will be done, an existing set of parameters can be used, just like the crab who makes it's home in an unused shell. Patch 36 uses wave 13 on all four oscillators with algorithm 4 on both sides. To hear the sample of the electric piano, press the program button (labeled "Seq Rec") and set it to 02. Still nothing will happen until a patch and top key are assigned so set (00)=7F and (01)=36.

Do you recognize the sound? Electric piano calls for a bit of chorus which is done by tuning each of the oscillators to a slightly different value. Set (54)=36, we must indicate the patch to be edited. Enter the following values, starting with (60)=1, pair one.

Pair 1	(60)=01		**Pair 2**	(60)=02
	(69)=01	(osc A course)		(69)=02
	(70)=FF	(osc A fine)		(70)=02
	(71)=02	(osc B course)		(71)=02
	(72)=01	(osc B fine)		(72)=00
	(80)=63	(volume A)		(80)=63
	(81)=63	(volume B)		(81)=63

A slight amount of "static" can be detected in the higher notes indicating the need for filter adjustment. Bring the filter down by setting the maximum filter frequency, parameter (88), to 80. If the value of parameter (87) is set to 25, then the filters action will vary with frequency, especially targeting the high notes.

If you are satisfied with the patch, then take the Ensoniq sound disk out of the drive and put the disk with our tutorial sounds back in. Save the information in the Mirages memory to bank #3. This writes over the data previously stored, but recall that this was the bank which we first loaded. So now the custom bank has both a string and an electric piano patch.

BUILDING A SINGLE CYCLE WAVE - ADDITIVE SYNTHESIS

There is nothing like a sample to capture the nuances of an acoustic instrument's attack. The sound is complex and synthetic reconstruction often sounds false. But the sustain portion of an acoustic sound diminishes in complexity. The end of the sound produces the greatest challenge to digital samplers both due to the limitations in available memory and the decreased signal to noise ratio. To combat digital noise, filtering is employed, with a tendency to dull sound. SOUNDPROCESS includes a feature which allows a wave to be defined by the amplitudes of its first ten harmonics. The resulting single page, single cycle wave simply does not have any aliasing as it never was sampled. The plan is to make a wave, insert it into an existing patch and then put it onto program 03 of the custom bank.

Load bank #3, which has the sounds we have made. With program 03 assigned to the keyboard, set (00)=7F and (01)=14. We have not altered anything yet, so listen to this horn patch (known on the system disk sound bank as program 23) as a reference. Set (54)=14 so that you can look at the values of the wave and algorithm parameters to get a feeling for how the sound is constructed. The attack comes from wave 01 and all of the other oscillators, numbered either 25 or 47 depending upon your version, provide background or sustain. The waveform map indicates that wave position 17 is empty, so we will begin by setting wave select (18) to 17. Make the value of parameter (19) (wave amplitude), the maximum so that we will be able to hear it. When (20) is set to 01, we declare our intention to harmonically define a wave. Enter the following values corresponding to amplitude levels for the first ten harmonics:

(21)=35	(22)=57	(23)=63	(24)=34	(25)=15	(26)=10
(27)=08	(28)=04	(29)=03	(30)=02	then press enter.	

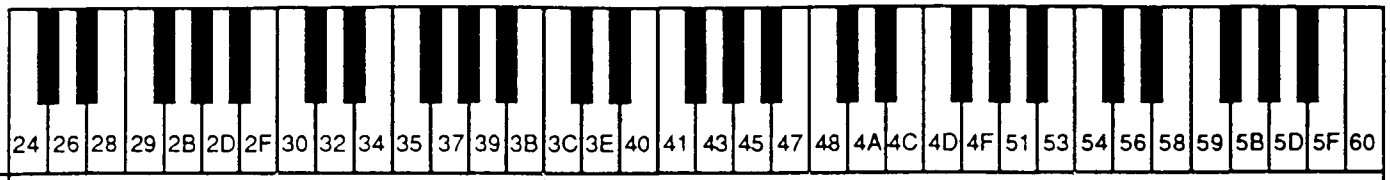
Now substitute wave 17 for the waves presently there. It would be good procedure to verify that the patch select setting is appropriate before doing extensive modifications ((54)=14). Set (60)=1 and (74)=17. For a more subtle attack, set (80)=36. Then switching over to the other pair, set (60)=2 and plant the new wave by setting (73) and (74) to 17. This patch can become a horn section by using different course tunings. Try changing (69) from 03 to 02 while you are on side 2. After this patch has been fine tuned to your satisfaction, save your work to the custom sounds bank. Please make sure the disk is again write protected.

If the sounds which are built using this method buzz or fuzz then decrease the setting for the wave amplitude, parameter (19). The entries to parameters (21) to (30) are amplitudes and if their total is too high, the output clips.

Where did the values for parameters (21) to (30) come from? They were generated by a computer program which does waveform analysis. Various reference books are also available which describe different acoustic instruments in terms of their characteristic harmonic composition. Then there is also trial and error, leading often to organ-like sounds.

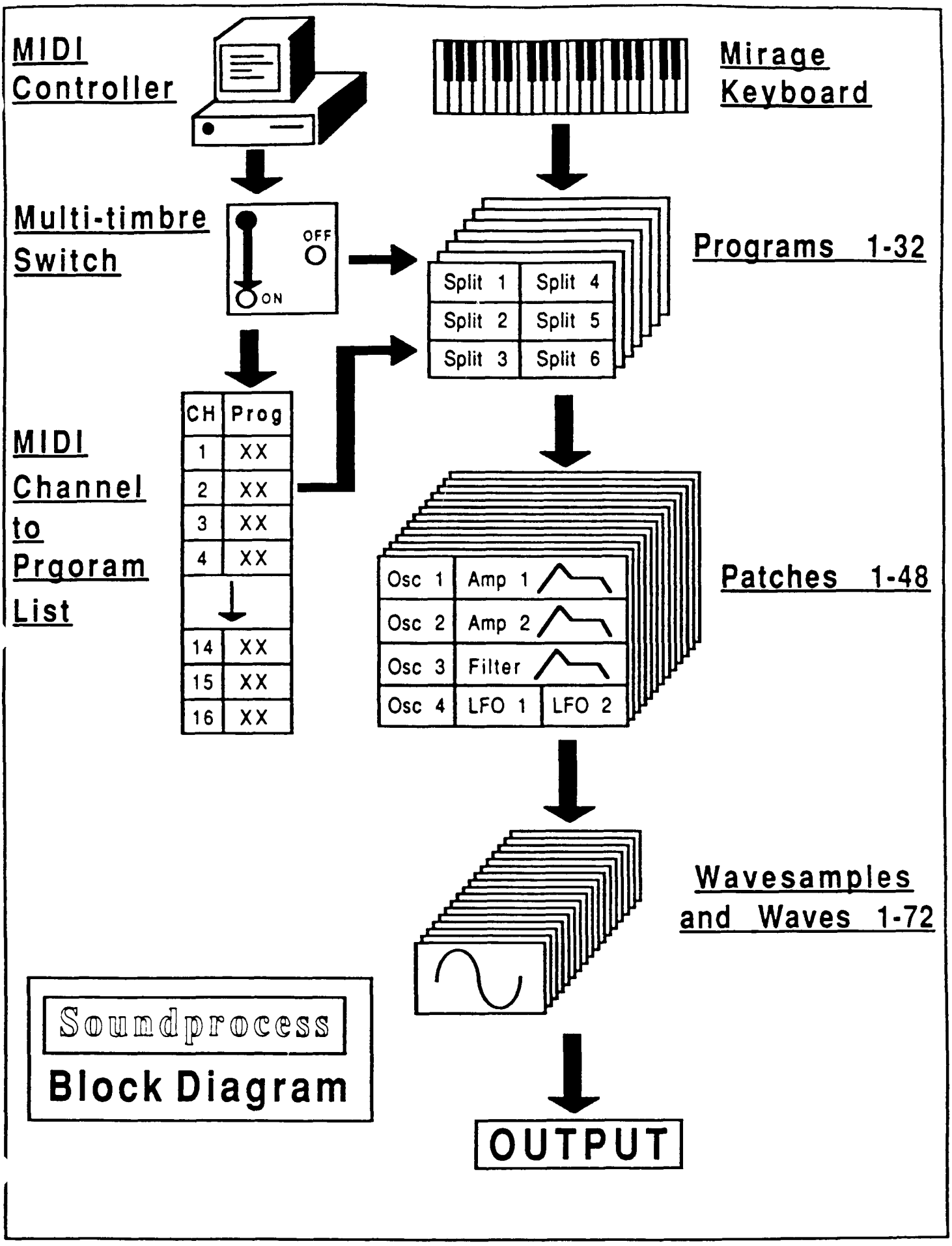
CONCLUSION

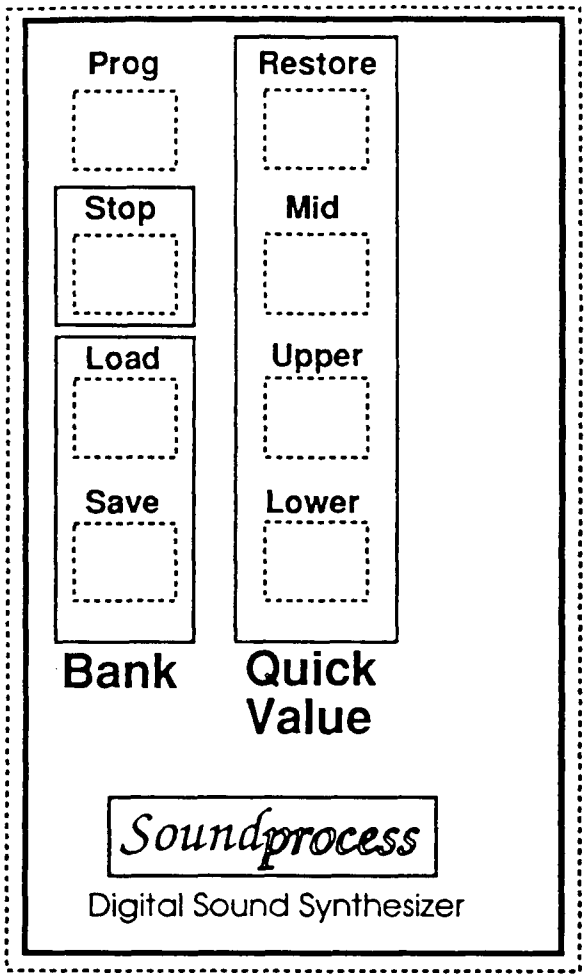
SOUNDPROCESS offers so many enhancements to the Mirage that there tends to be something for everyone. This tutorial was written in the hope that with some encouragement people would explore features which they otherwise would have avoided. Take your time as you learn, because due to this operating system, the Mirage should remain state of the art for awhile.



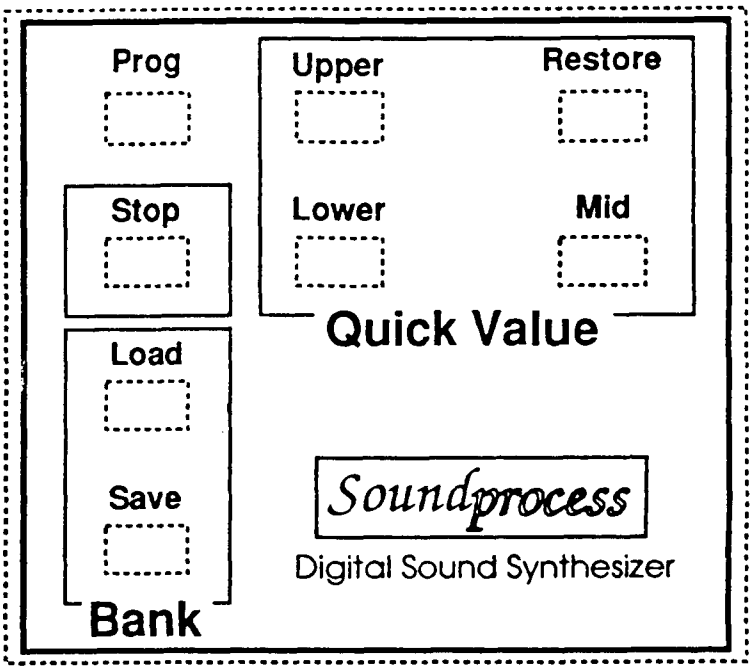
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	

Program Layout Chart


MIDI Controller
Mirage Keyboard
Multi-timbre Switch
OFF
ON
Programs 1-32
Split 1
Split 4
Split 2
Split 5
Split 3
Split 6
MIDI Channel to Prgoram List
CH
Prog
1
XX
2
XX
3
XX
4
XX
14
XX
15
XX
16
XX
Patches 1-48
Osc 1
Amp 1
Osc 2
Amp 2
Osc 3
Filter
Osc 4
LFO 1
LFO 2
Wavesamples and Waves 1-72
Soundprocess
Block Diagram
OUTPUT



DSK-8 Front Panel

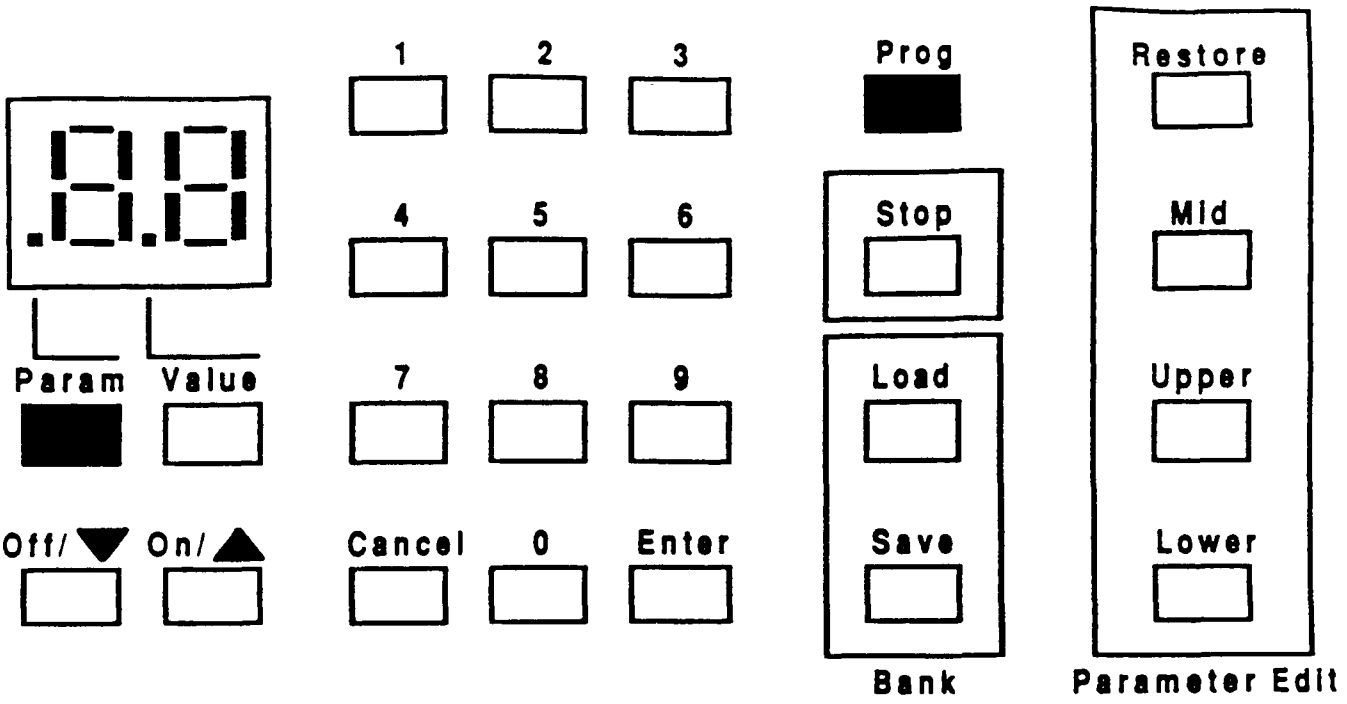


DSK and DMS Front Panel

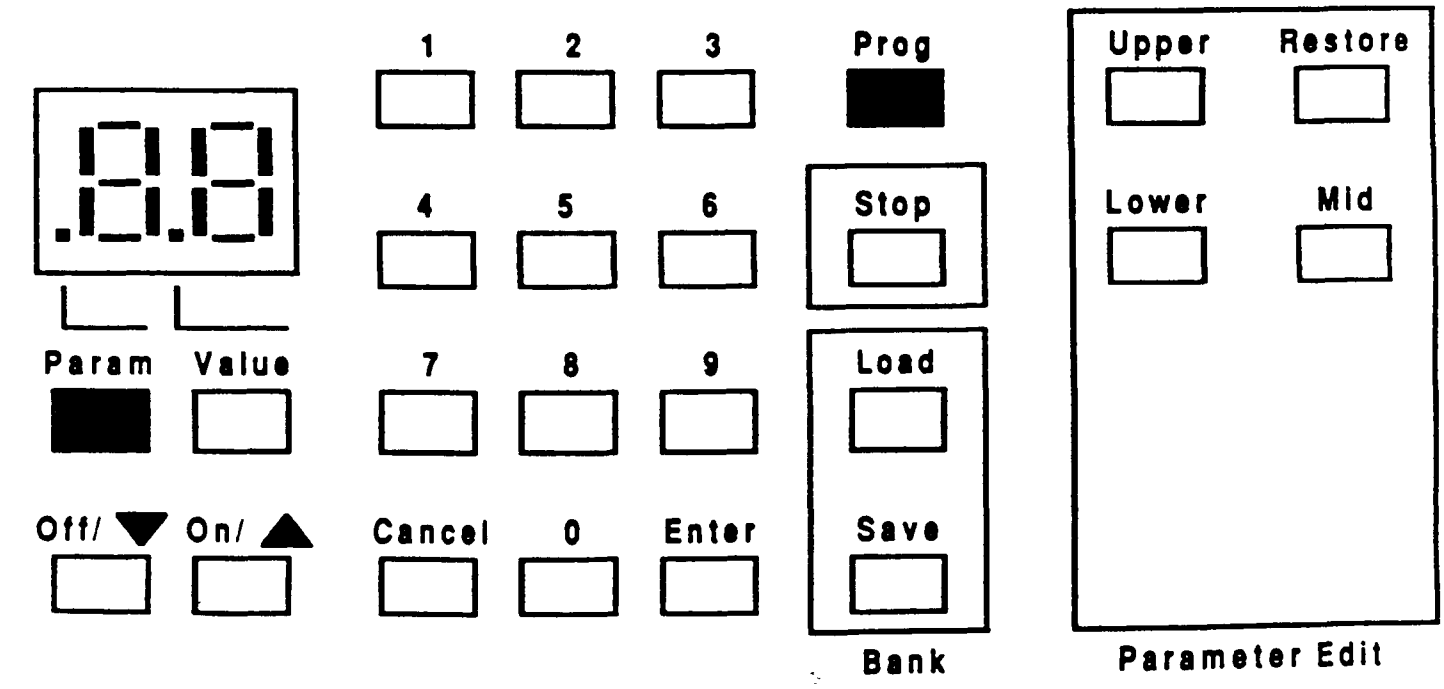
Instructions:

1. Make several copies of this page in case you make a mistake.
2. Apply a sheet of clear contact paper over one of the copies.
3. Cut along the outside dotted line to remove the overlay.
4. With a razor or sharp knife, cut the key holes out.
5. Set the overlay on your Mirage front panel so that the <Prog> key covers the original "Seq Rec" key.

Soundprocess Front Panel Overlay



Mirage DSK-8



Mirage DSK and DMS