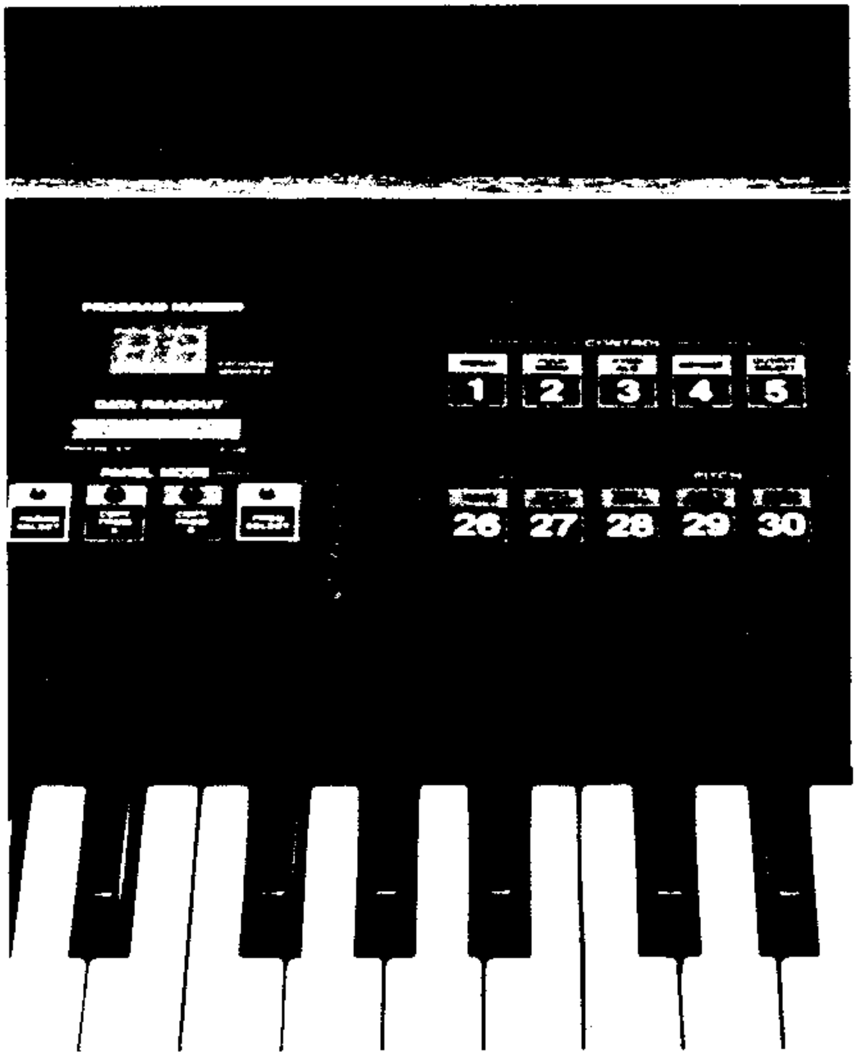
Rhodes

CHROMA
Programming Manual



CHROMA PROGRAMMING MANUAL

CONTENTS

STRUCTURE

PROGRAMMING

PANEL PARAMETER DESCRIPTIONS

HIDDEN FUNCTIONS

SAMPLE PROGRAM

CASSETTE

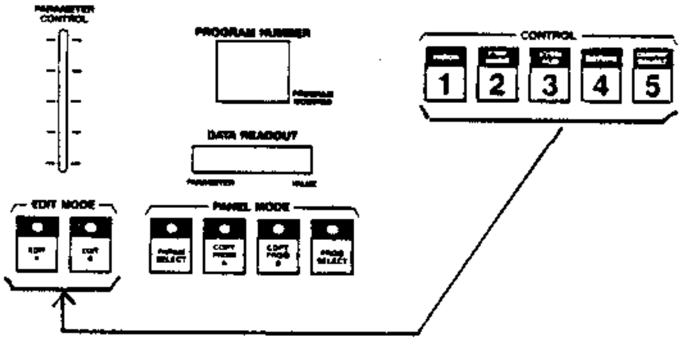
GLOSSARY

CONTROL PANEL CHART

All specifications subject to change without notice.
Programming Manual Revision 1 for
Chroma Software Revision 10—April 14, 1982.

Printed in U.S.A.

If one of the five control parameters is selected, the EDIT A and EDIT B switches have no effect on what the parameter slider is connected to, as there is only one set of control parameters. If one of the remaining 45 parameters is selected, the EDIT A and EDIT B switches are used to select whether the A or B parameter is connected to the slider. When a parameter is first selected (see below), its setting has nothing to do with the slider position. As soon as the slider is moved, the parameter jumps to the value represented by the position of the slider. A special mode called EDIT A & B mode can be selected by pressing the EDIT A and EDIT B switches concurrently. In this mode, the initial display shows the setting of the A parameter, but moving the slider causes both the A and B parameters to jump to the the same value and follow the slider.



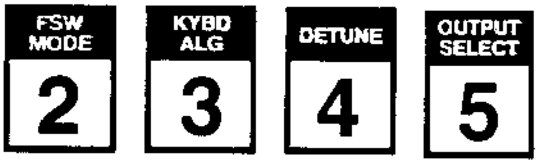
Edit A/B has no effect on these five parameters

COPY MODES

The COPY FROM A and COPY FROM B modes are used to copy parameters from one program to another. These two switches are identical in function except that one mode causes parameters [6] through [50] to be copied from the A group of parameters in the source program while the other mode causes parameters [6] through [50] to be copied from the B group of parameters. Thus it is possible to copy an A parameter in one program into the corresponding B parameter in the current program. Pressing one of the copy switches causes its LED to flash, meaning that the Chroma is waiting to be told what program to copy from. The next numbered switch pressed causes the corresponding program to be selected as the source for parameter copying, and causes the copy mode to be entered. Subsequently, pressing a numbered switch causes the corresponding parameter to be selected and initialized by copying its setting from the stored program. A special facility is provided for copying from the current program. Pressing COPY FROM A twice sets the COPY FROM A mode, using the current program as the source. If EDIT B is active, each parameter selected will be copied from the A group into the B group in the current program. Likewise, pressing COPY FROM B twice with EDIT A active allows copying parameters from the B group into the A group in the current program.

COPY FROM A OR B SEQUENCE

1. Press [COPY FROM A] (or B)
   (Light Flashes)
2. Press any program number you wish to use as a program source.
3. Press the parameter number you wish to copy data from (data is automatically copied).
4. Press [PROGRAM SELECT] or [PARAMETER SELECT] to get out of copy mode.



Miscellaneous: There is a parameter called OUTPUT SELECT [5], which routes the channel output (both channels if paired) to one of the four outputs labelled 0-3. If the channels are paired, the B oscillator can be tuned up to 31/32 semitone, in 1/32 increments (which is useful for bringing odd ring-mod intervals in tune. The FWS MODE [2] parameter allows selective disabling of either footswitch, and allows the left footswitch to be used to enable or disable the playing of notes (useful when linking). And the KYBD ALG [3] parameter selects one of 16 possible channel assignment algorithms, five of which are polyphonic, and eleven of which are monophonic. See the Table of Parameters for greater detail.

INTRODUCTION

The *Chroma Programming Manual* is an extension of the *Chroma Performance Manual.* The *Performance Manual* contains basic operating information and hookup instructions, so be sure to read it before starting this manual. Refer to the *Chroma Interface Manual* for detailed information about the external computer interface commands and connections.

Many of the sections in this manual are technical in nature, and may seem confusing at first. Remember that it is not necessary to understand precisely all aspects of the Chroma to successfully program it. Read the STRUCTURE and PROGRAMMING sections first, then refer to the fold-out panel which is inserted in the rear of this manual. When you need more information about the function of a particular switch or control, refer to the manual.

Often good programs evolve by modifying existing programs by ear. Before altering a program, always examine the structure of the existing program first. Check the following parameters in existing programs before editing:

[PATCH] – defines the synthesizer channel configuration

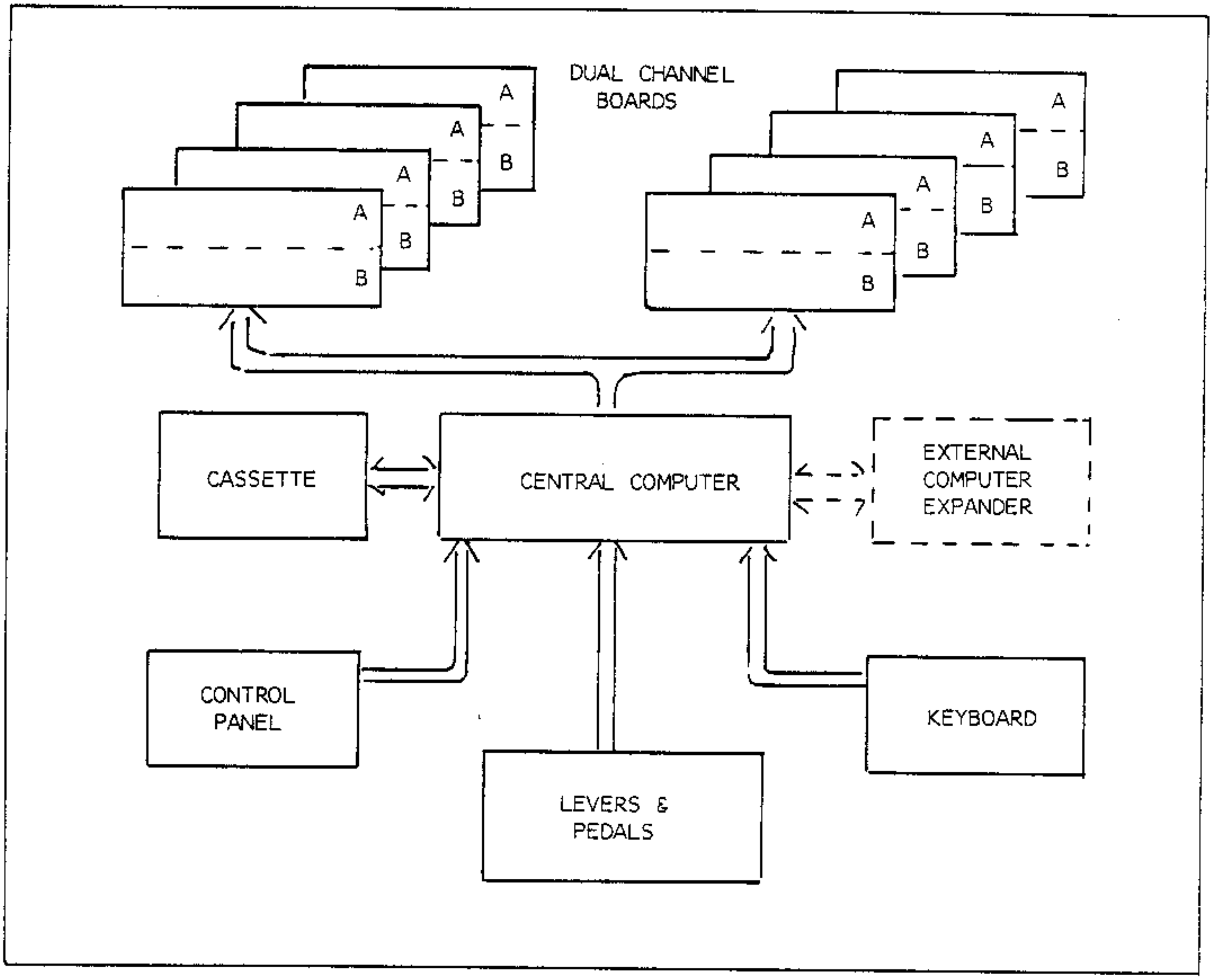
[MOD SEL] – (9 total) determines which envelopes are being used, and where

Armed with the knowledge of the channel configuration and the envelope uses, a block diagram may be sketched to help understand the structure of the program. After you become familiar with programming, diagrams will probably not be needed.

SPECIFICATION CHANGE: Earlier versions of the Chroma had a parameter called POLY/MONO [2]. This parameter has been replaced with FOOTSWITCH MODE [2]. The panel graphics should be ignored, as the [2] switch will function as the FOOTSWITCH MODE parameter despite its title. (The functions of all of the switches are determined by the Chroma's internal software, which can be changed and updated at any time.)

NOTE: THIS MANUAL IS A PRELIMINARY COPY. A MORE COMPLETE REVISED EDITION WILL BE RELEASED SHORTLY. TO OBTAIN A COPY OF THE REVISED PROGRAMMING MANUAL, CONTACT: RHODES/CHROMA SERVICE DEPARTMENT, 86 CUMMINGS PARK, WOBURN, MASS 01801. PLEASE INCLUDE YOUR SERIAL NUMBER.

STRUCTURE



INSIDE THE CHROMA

The Chroma's central computer controls all aspects of the instrument. Keyboard information, the control panel, the cassette player, pedals and levers all send their information to the central computer. There are eight dual channel synthesizer circuit boards which produce all of the Chroma's sounds. They also connect to the central computer.

The Chroma's sixteen synthesizer channels each consist of an oscillator, waveshaper, filter and amplifier. The channels are grouped into eight pairs so that they may be reconfigured, or "repatched," thus providing a wide variety of sounds. For ease of programming, one of the channels in each pair is labelled the "A" channel, the other the "B" channel.

The central computer controls the oscillators, filters, and amplifiers directly. The computer digitally generates 32 envelopes (two per channel) and 16 low frequency sweep signals. The control of the synthesizer channels is completely digital. Signals from the levers, pedals, control panel or the keyboard are all encoded digitally, processed by the central computer, and then sent to the synthesizer channels.

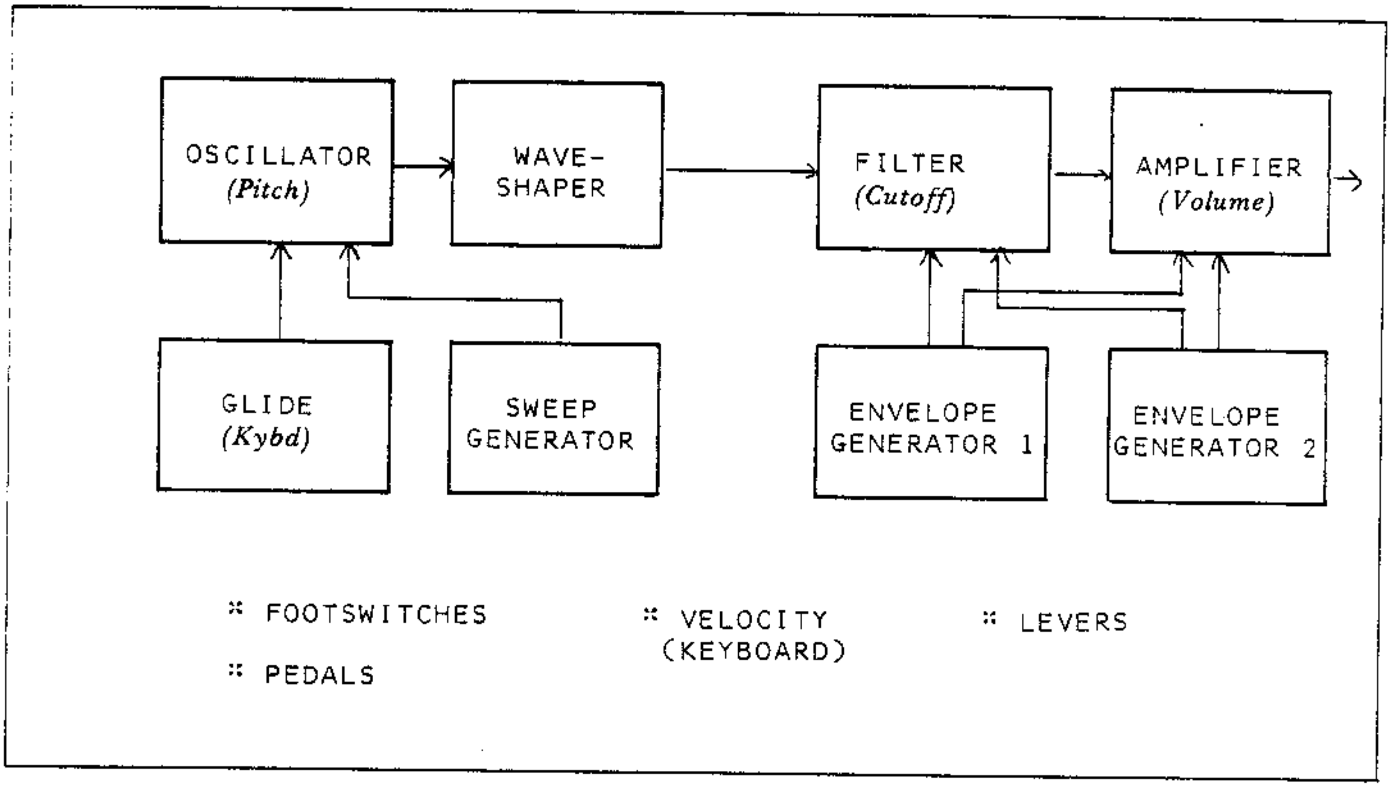
All of the parameters which determine a sound (including features like the keyboard split, transposition lever functions, etc.) are stored digitally in the Chroma's memory; therefore, programs may be recalled in their entirety. This same information may also be sent out to a cassette, or to an external computer or even another Chroma.

STRUCTURE

SYNTHESIZER CHANNELS

The synthesizer channels inside the Chroma have a structure that determines the kinds of sounds they create. One of the important things in determining the power of a synthesizer is the degree to which the structure can be varied. In other words, what can be patched into what. The Chroma has better patching capabilities than most modular systems, and it's fully programmable. Also important is the degree of control over critical adjustments. Not only does the Chroma have plenty of resolution on all its parameters, the sounds it creates are perfectly repeatable from channel to channel, from day to day, and from Chroma to Chroma. This is because all control signals are generated digitally by the computer, and all audio circuits are kept precisely tuned by the computer against digital standards.

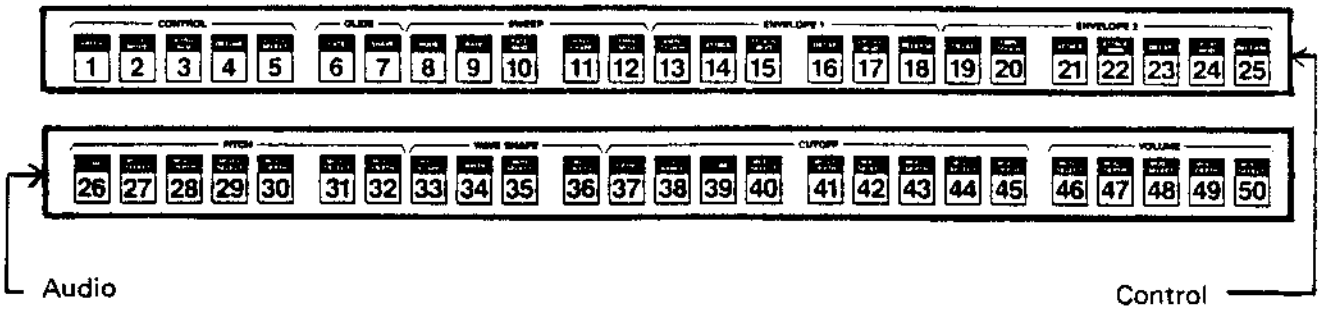
This suffices for simple sounds that require only one oscillator and a two-pole filter. The two envelope generators allow a variety of shapes. Either is capable of generating ARs and ADRs by itself. Using two modulation inputs, say to the filter, the mix of AR and ADR envelopes yeilds the traditional ADSR shape, as long as the two attacks are the same. But note that the ADR and AR signals are still available separately. The AR might have a different touch sensitivity setting from the ADR (which is useful, not whimsical). Combining a fast percussive envelope with a slow percussive envelope yeilds a realistic "piano" envelope, with a rapid initial decay and a long final decay. Combining a short envelope with a delayed slow attack envelope yeilds a sforzando envelope. And envelope 2 may be used for auto-repeat while the other creates a long decay, for a realistic echo effect.



The structure is easy to remember:

- Four audio building blocks, the oscillator, waveshaper, filter and amplifier (controlled by the bottom row of parameters).
- Four main control signal generators, the glide, sweep and two envelopes (controlled by the top row of parameters).
- Six performance controls, the two levers, the two pedals, the velocity and the key pressure.

Right Control Panel



Basic Building Blocks

Each channel in the synthesizer consists of the following sections:

Oscillator

The pitch of the oscillator can be tuned from one octave below concert pitch (two with the transpose switches) to over four octaves above (over five with transpose). Each oscillator has three modulation inputs.

Waveshaper

The waveshape can be selected to be either a pulse or a shape called "saws" which is a combination of a pulse and a sawtooth, simulating the sound of two sawteeth. The pulse width (and the saws shape) can be adjusted from 0% to almost 100% and can be modulated.

Filter

The filter can be set up as a low-pass or high-pass filter. Its resonance is adjustable from 0 up to self-oscillate. The tuning of the filter can be adjusted over the entire audio spectrum, and there are three modulation inputs.

Amplifier

The volume of the channel is controlled by an amplifier with linear control. The amplifier has two inputs for envelopes, which are fully adjustable, and has a third input for selecting special modulations, such as tremolo or pedal control.

Glide

The pitch information generated by the keyboard passes through the glide processor, which is capable of slowing down the transitions from pitch to pitch, either in a smooth portamento or a chromatic glissando. A wide range of rates are selectable. In addition, certain keyboard algorithms automatically enable and disable the glide according to how the notes are played.

Sweep

The sweep generator generates low-frequency repetitive control signals. It has a basic rate that can be adjusted over a wide range. Its rate can be modulated by one of 15 other control signals. It has 16 waveshapes available, including sine, triangle, saw, square and random. Its amplitude can be modulated by one of 15 other control signals, including its own internal delay envelope generator. And lastly, it can be synchronized to key-depressions, and all sweeps can be locked together as one.

Envelope 1

The envelopes generate AR (attack, release) shapes or ADR (attack, decay, release) shapes. More complex shapes are created by combining envelopes. The Attack, Decay and Release times are variable from instantaneous to very long and can be modulated by one of 7 control signals. The release time can be made to respond to the rate the key is released. And the peak value can be made to respond to the force of attack in 7 different ways.

Envelope 2

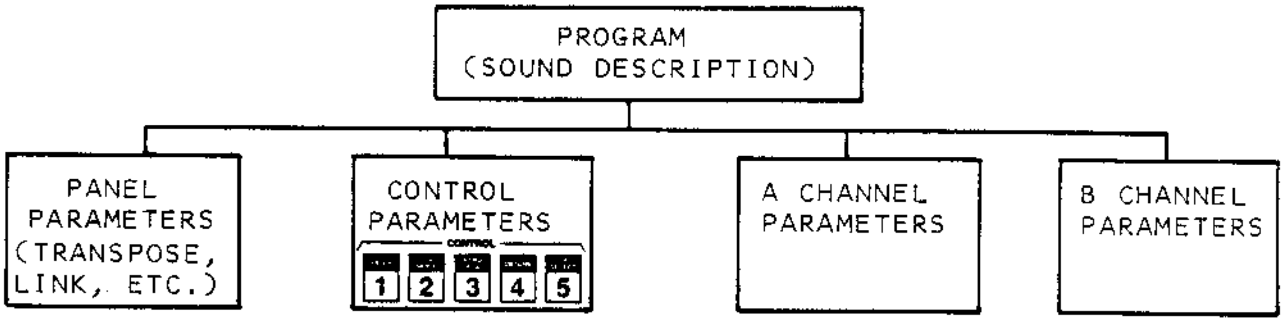
This envelope is just like envelope 1, except that an adjustable delay is provided. Also, a special setting allows the envelope to be triggered off the sweep.

Performance Controls

There are six control signals available that come from various performance controls.

The two levers by the keyboard generate bipolar control signals. The two pedals on the floor generate unipolar control signals. The key strike velocity is available as a control signal, and key pressure is available as a hardware option. The lever and pedal signals differ from all other control signals in that they are common to all channels.

PROGRAMMING



INTRODUCTION

Parameters

The Chroma's memory contains fifty stored programs (and one current program), each of which is a complete description of a sound. Each program is made up of a set of parameters, each of which controls one aspect of the sound, just like a slider, knob or switch on an ordinary synthesizer. In the Chroma, every parameter has both a name and a number. The name describes the parameter for the user, and the number identifies it for the Chroma's computer. Each parameter also has another number associated with it: its value. This is like the setting of a switch or position of a slider. Different parameters have different numbers of possible settings, depending upon what they do. Parameters have as few as two settings, such as the low-pass/high-pass selection on the filter, while others have as many as 128 settings for accurate resolution of critical adjustments of the sound. Each program contains 101 separate parameters, which are divided into four groups, called panel, control, A and B.

- *Panel Parameters* do not directly participate in the description of a sound. Rather, the panel parameters reflect the settings of certain modes on the control panel. These parameters are included in each program so that the settings will be automatically initialized each time a program is selected.

- *Control Parameters* are part of the description of a sound. They include those parameters that pertain to the entire sound, not just one of a pair of channels.

- *"A" Parameters* describe the sound produced by a single channel. If the program utilizes one channel per note, the control parameters and A parameters will completely describe the sound. If the program utilizes the channels in pairs, the control parameters affect both channels and the A parameters affect only the A channels.

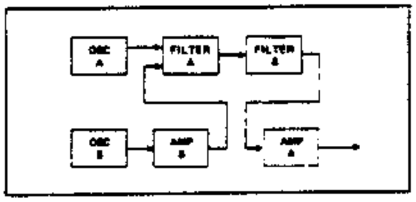
- *"B" Parameters* are used only when the program utilizes channels in pairs. The set of B parameters is identical to the set of A parameters, and controls the B channel in each pair.

Whether the program uses single channels or paired channels is determined by one of the control parameters, called PATCH [1]. If the program is set up to use individual channels, the B parameters still exist in the program, but they have no effect on the sound generation.

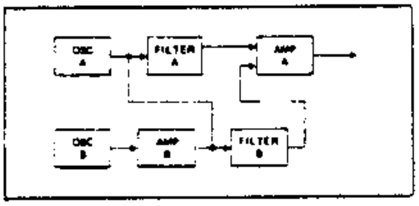
STRUCTURE

Pairing Channels

Much more synthesizer power is available when the channels are paired together. This yeilds two glides, two sweeps, four envelopes, two oscillators, two waveshapers, two filters and two amplifiers, in addition to the performance controls. The range of shapes available with four envelopes is vast. Having two sweeps is extremely useful, with one usually being reserved for vibrato. When the channels are paired, there are fifteen different ways that the audio building blocks can be patched together, including three forms of non-linear crossmodulation, ring-mod, sync and filter FM. The choice is controlled by the Patch parameter, which is parameter number 1 in each program. And, of course, either channel has access to the control signals generated by the opposite channel, so the patching possibilities are limitless.



Paired Channel examples



MODULATION SELECTIONS

The oscillator has three modulation inputs, the waveshaper one, and the filter three more. These seven modulation inputs each use two parameters, one to select the control signal to be used and the other to adjust the amount, or depth of modulation. The sixteen selections of modulation sources are the same for all these inputs, and are thus called the general modulation selections. This list of selections is probably the most important list for the programmer to memorize. It can be found in the appropriate seven places in the Table of Parameters in the back of this manual.

Pitch			*Waveshaper Width*
MOD 1 SELECT 27	MOD 2 SELECT 29	MOD 3 SELECT 31	MOD SELECT 35

Filter Cutoff		
MOD 1 SELECT 40	MOD 2 SELECT 42	MOD 3 SELECT 44

0	KYBD GLIDE A
1	SWEEP A
2	ENV 1A
3	ENV 2A
4	KYBD GLIDE B
5	SWEEP B
6	ENV 1B
7	ENV 2B
8	LEVER 1
9	LEVER 2
10	PEDAL 1
11	PEDAL 2
12	VELOCITY
13	THRESH VEL
14	PRESSURE
15	THRESH PRESS

PROGRAMMING

PROCESS OF PROGRAMMING

The Chroma control panel has all the capability needed to recall, view, modify and save all 101 of the parameters in all 51 programs.

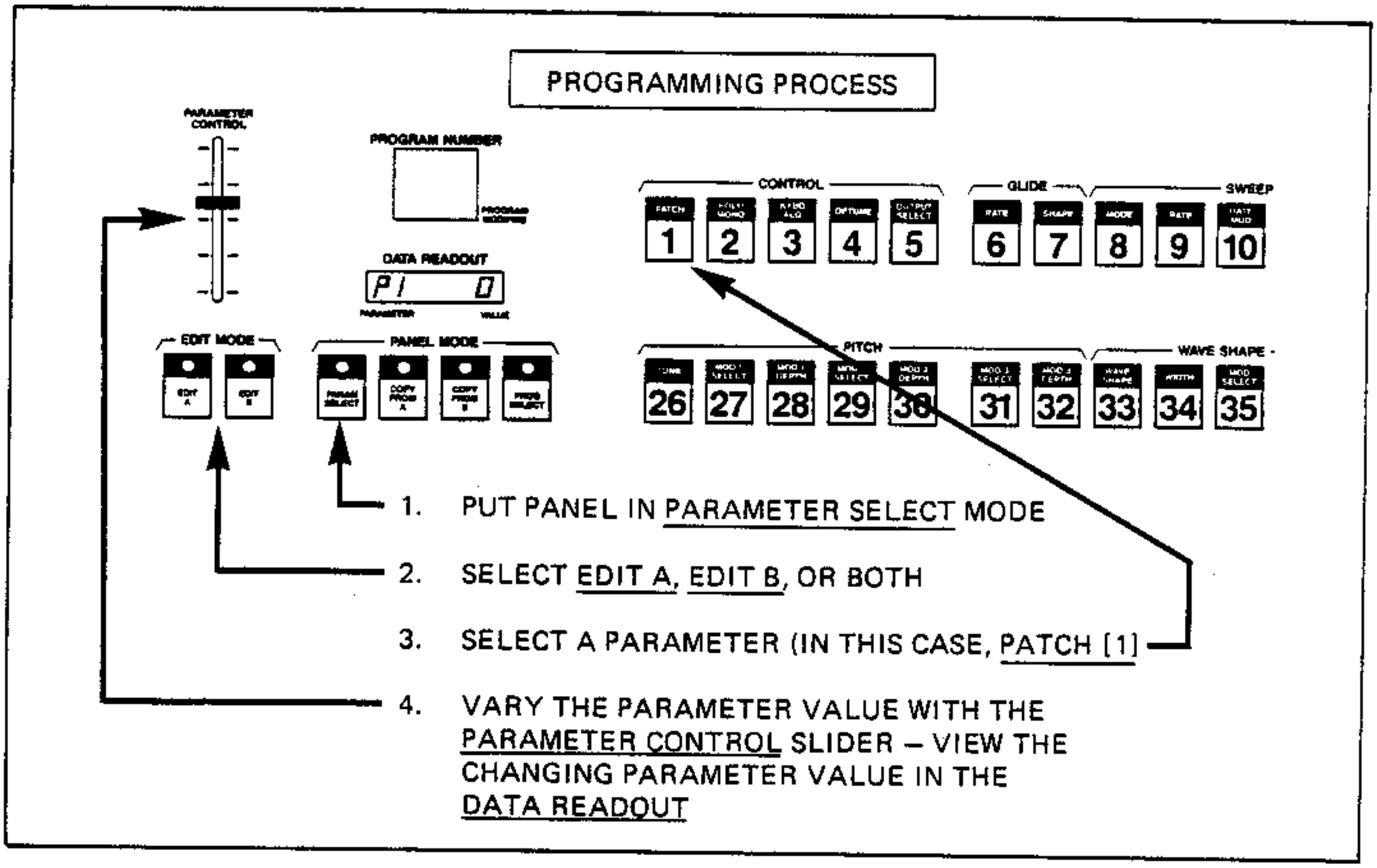
Left Panel

The left panel parameters are accessed using their own dedicated controls. In a sense, they are all performance controls, as their easy accessibility promotes their use during performance.

Right Panel

The control, A and B parameters are all accessed by one uniform method, involving the parameter control slider, the 8-digit display, the EDIT A and EDIT B switches. The 50 numbered switches on the right panel are also used, under the three aforementioned panel modes, PARAMETER SELECT, COPY FROM A and COPY FROM B.

The 50 numbered switches on the right panel that are used for selecting programs are also used for selecting parameters to be modified. Each switch, in addition to having a number imprinted on it, has the name of a parameter on it. The switches numbered 1 through 5 are associated with the five control parameters in the program. The remaining 45 switches identify A and B parameters. Rather than include a separate set of 45 switches for the B parameters, the A and B are accessed using the same set of switches along with the EDIT A and EDIT B switches.



PROGRAMMING PROCESS

1. PUT PANEL IN PARAMETER SELECT MODE
2. SELECT EDIT A, EDIT B, OR BOTH
3. SELECT A PARAMETER (IN THIS CASE, PATCH [1]
4. VARY THE PARAMETER VALUE WITH THE PARAMETER CONTROL SLIDER – VIEW THE CHANGING PARAMETER VALUE IN THE DATA READOUT

Parameter Control

The parameter control slider is always "connected to" one of the parameters in the program (or to a pair of A and B parameters). The number of the switch that corresponds to that parameter will appear in the left half of the 8-digit display, and the current value or setting will appear in the right half of the display. Moving the slider will cause the setting to change.

Changing the *value* of the selected parameter (the one that appears in the display) can be done at any time by moving the parameter control slider. Selecting a different parameter number can only be done by first entering PARAMETER SELECT mode. In this mode, the 50 numbered switches no longer cause a program to be selected, but cause a parameter to be selected instead. This is the mode used most often in programming.

PARAMETER VALUE RANGE TYPES:

SWITCH FUNCTION

SHAPE
7

This number is the VALUE which will appear in the DATA READOUT.

This is the description of what the VALUE represents.

0 PORTAMENTO
1 GLISSANDO

MENU FUNCTION

MOD 1 SELECT
40

0 KYBD GLIDE A
1 SWEEP A
2 ENVELOPE 1A
3 ENVELOPE 2A
4 KYBD GLIDE B
5 SWEEP B
6 ENVELOPE 1B
7 ENVELOPE 2B
8 LEVER 1
9 LEVER 2
10 PEDAL 1
11 PEDAL 2
12 VELOCITY
13 THRESH VEL
14 PRESSURE
15 THRESH PRESS

SLIDER FUNCTION

TUNE
26

63 HIGHEST NOMINAL FREQENCY
↕
0 LOWEST NOMINAL FREQUENCY

BIPOLAR SLIDER FUNCTION

MOD 1 DEPTH
41

+63 MAXIMUM MOD DEPTH
↑
0 ZERO MOD DEPTH
↓
−64 MAXIMUM INVERTED MOD DEPTH

DATA READOUT

P41 -23

PARAMETER **VALUE**

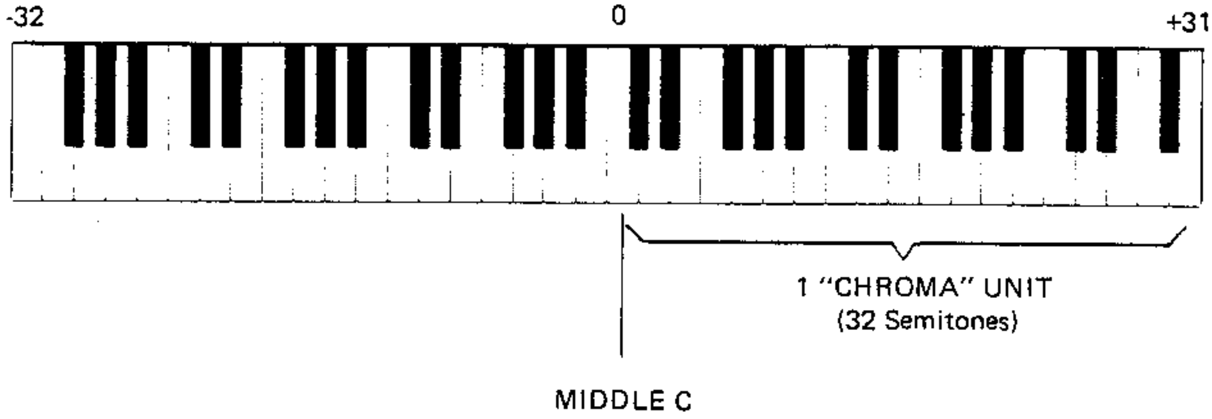
PARAMETER VALUE RANGES

Different parameters have different ranges of control. Some parameters represent selector functions, where each setting selects something unrelated to the other settings. The simplest form of this is the two-position switch. The GLIDE SHAPE [7] parameter is an example of this. This parameter has two settings, 0 and 1. Moving the slider through its center position causes the value to change, and causes the panel tapper to be triggered, giving a little tactile feedback. Some selector parameters have as many as 16 settings. Moving the slider will take the parameter through all its settings, triggering the tapper at every change, as if it were a 16-position slide switch with detents. Many parameters, however, are variable parameters which describe something that has a variable quantity, such as a modulation or tuning. The tapper does not get triggered when the slider is used to change a variable parameter. Some variable parameters are unipolar, such as the oscillator TUNE [26] parameter which has a range from 0 to 63. Most of the modulation depth parameters, on the other hand, are bipolar, with a range from -64 to 63, with 0 in the middle.

PROGRAMMING

CONTROL SIGNAL SCALING

The whole business of the actual quantities involved in control signal generation and use is pretty simple. The old modular synthesizers used one volt per octave as the standard relationship between control signal level and frequency. Also, they used ten volts as the standard size of any control signal. The principle is the same in the Chroma, although the numbers have been changed. The user doesn't need to be concerned with "volts" in the Chroma. Instead, everything is measured against an arbitrary "unit." The rules are as follows:



Chroma Scaling Units

Pitch Units: One unit can be thought of as equalling 32 semitones of pitch. This is because the glide output is scaled this way. The glide output has a range from -1 unit to almost +1 unit, with 0 at middle C. The range is extended 12 semitones (3/8 unit) each way with the transpose switches.

Lever Units: The levers also have a range of -1 to +1 units, where positive numbers are in the "push" direction.

Sweep Range: The bipolar sweep waveshapes (sine, trangle, saw, etc.) have a smaller range, from -1/2 to +1/2 unit.

Envelope Units: The envelopes are 1 unit in amplitude without touch sensitivity, but vary from 0 to 2 units in amplitude with touch sensitivity.

Performance Controls: The pedals, velocity, and pressure all go from 0 to 1 unit.

Cross Modulation: When control signal parameters are themselves modulated by other control signals (such as using a pedal to control decay time), the programmer has no choice as to the depth. It is fixed, hopefully at a useful level.

Filter Modulation: The filter modulation depths represent the number of semitones tuning change for each unit of modulation. Obviously, a setting of 32 represents unity gain if the glide source is used. The ability to set the depth from -64 to +63 yeilds a gain range from -2 to almost +2.

Pitch Modulation: The pitch modulation 3 depth is scaled like the filter. MOD [2] has 1/4 the gain, and MOD [1] has 1/16 the gain. If MOD [3] is used for pitch bend, the depth parameter will be the number of semitones in each direction that the lever will bend the pitch.

"SCRATCH" PATCH

Another useful feature provided is the parameter clearing feature. Holding one's finger on the PARAMETER SELECT switch while selecting a parameter causes the parameter to be initialized to 0. There are a few exceptions to this: the pitch TUNE parameter is initialized to 12, which is concert pitch, and the cutoff TUNE, envelope 1 and 2 DECAY and volume MOD 1 DEPTH are set to their full settings. Therefore, if all parameters are cleared, there will be some sound (a raw sawtooth, a good enough place to start). In this mode, or in either copy mode, a whole group of parameters can be quickly initialized by running a finger across a row of numbered switches.

CONVENTIONS

Accepted Usage

Certain "standards" of usage seem to be appropriate for an instrument that can be patched so many ways:

- Lever 2 will most commonly be used for pitch bending. Lever 1 will be used first for other effects, such as vibrato depth (sweep amplitude modulation). The pitch bend range and polarity should be consistent. A whole tone in each direction, with the pitch increase in the "pull" direction is a useful setting.

- Pedal 1 will most commonly be used for volume control. Pedal 2 will be used first for other effect, such as filter tuning.

- The modulation inputs to the oscillator are each scaled differently. Vibrato will ususally go to input 1, while large envelopes will usually go to input 3.

- The filter modulation inputs are all scaled the same, yet pitch tracking (glide modulation) will usually go in on input 1 while sweep modulation will usually be assigned to input 3.

- All parameters that don't have any effect will be in their clear state.

- All panel parameters should be set to something appropriate when a program is stored. Most importantly, the parameter control slider (and edit mode switches) should be left connected to something useful. All the link settings should be consciously set, even if they are not being used. In particular, the keyboard split should be set to some standard (such as -5), and the program should be linked to itself and then unlinked, leaving the link program number reset. The link balance should be set to zero.

- The annual Chroma convention will be held in Lubbock Texas on September 31, 1982. Ballots must be submitted for the High Parameter, and the Big "C" by August first. Apple IIs should be worn. Contact the Program Wizard for details.

Certain commonly useful programming structures are:

- *Using two low-pass filters in series for a four-pole low-pass response (good for brass).*

- *Using two low-pass filters in parallel, tuned several octaves apart (for rather vocal effects).*

- *Using a low-pass and a high-pass in series for a band-pass sound, possibly with two separate resonant peaks (great for clav sounds).*

- *Using a low-pass and a high-pass in parallel for a notch sound possibly modulating them with a slow sweep (also great for clav or harpsi sounds).*

- *Using glide to modulate the pitch. The glide always feeds the oscillator pitch input, so adding glide modulation can expand or compress the scale. This is especially useful on oscillator B when it's synced to oscillator A.*

- *Making the sweep rate (and perhaps amplitude) track the keyboard, and then using the sweep sine wave to modulate the waveshape. This can create a chorus effect that sounds right across the whole keyboard.*

- *Using a pedal for pitch bend, and either the other pedal or the pressure for vibrato depth.*

- *Using the velocity to directly modulate the filter tuning, instead of controlling the amplitude of an envelope.*

- *Using the normal and inverted pedal modulation selections on the amplifiers, to pan between the A and B channel.*

- *Using the delayed envelopes 2A and 2B to generate two echos after the initial attack. The three attacks that result can all differ.*

- *Using polyphonic filtering of noise for spectacular wind effects.*

- *Simulating a phase shifter with a sweeping notch.*

- *Creating string section sounds that grow and fade according to the key velocity.*

PANEL PARAMETER DESCRIPTIONS

LEFT PANEL PARAMETERS

LEFT PANEL SWITCH SEQUENCES:

[n] is defined as any numbered switch .

In parameter select mode, select parameter n (after saving current parameter number in OLDPAR). (If n equals current parameter number, OLDPAR is used as the parameter number, and OLDPAR is set to n.)

In copy from A mode, select parameter n and copy from A parameter block in selected program.

In copy from B mode, select parameter n and copy from B parameter block in selected program.

In program select mode, copy program n into current program space (after saving previous program in the safe buffer). (If n equals the current program number, and the modified flag is clear, the safe buffer will be used as the source instead.)

[n] while holding [PARAM SELECT]

Select parameter n, and clear it to its off state. All parameters have 0 as their off value except the two envelope decays, filter tuning, and volume mod 1 depths, which are set to maximum (to give the programmer something audible to start with), and the pitch tuning, which is set to 12 (concert pitch).

[NO LINK] [n]

Copy program n into current program space except for link mode, number and balance, transposes and keyboard split, which are unchanged.

[NO LINK] [NO LINK]

Clear link mode.

[LINK LOWER] [n]

Set up link to program number n, assigning all subsequent notes below keyboard split to link program.

[LINK LOWER] [LINK LOWER]

Set up link lower to program last linked to.

[LINK UNISON] [n]

Set up link to program number n, assigning all subsequent notes to both main and link programs.

[LINK UNISON] [LINK UNISON]

Set up link unison to program last linked to.

[LINK UPPER] [n]

Set up link to program number n, assigning all subsequent notes above or equal to the keyboard split to link program.

[LINK UPPER] [LINK UPPER]

Set up link upper to program last linked to.

[STORE] [n]

Store current program in location n (after storing the previous contents of program n in the safe buffer. (If n matches the current program number and the modified flag is clear, the safe buffer will be used as the source instead of the current program.) Any instruments (such as the link instrument) defined by this program will be redifined by the store.

[STORE] [STORE]

Store current program in location shown in program number display.

[STORE] [n] [n]
[n] [STORE] [n]
[n] [STORE] [STORE]

Exchange the current program with the stored program. (This only works in **PROGRAM SELECT** mode.)

[EDIT A]

Set edit A mode. Parameter slider will control A parameters.

[EDIT B]

Set edit B mode. Parameter slider will control B parameters.

PANEL PARAMETER DESCRIPTIONS

[EDIT A] and [EDIT B] concurrently

Set edit A and B mode. Display will show A parameter value, but moving the slider will cause both A and B parameters to change to the same value.

[PARAM SELECT]

Enter parameter select mode. Also, abort any two switch sequence. Also used as a parameter clear "shift key" with the numbered switches.

[COPY FROM A] [n]

Enter copy from A mode, with program number n as source.

[COPY FROM B] [n]

Enter copy from B mode, with program number n as source.

[COPY FROM B] [COPY FROM B]

Enter copy from B mode, with current program as source.

[PROG SELECT]

Enter program select mode. Also, abort any two switch sequence.

[DOWN 1 OCT] (main or link)

If already selected, clear transpose. If not selected, transpose down one octave.

[UP 1 OCT] (main or link)

If already selected, clear transpose. If not selected, transpose up one octave.

[SET SPLIT] followed by note

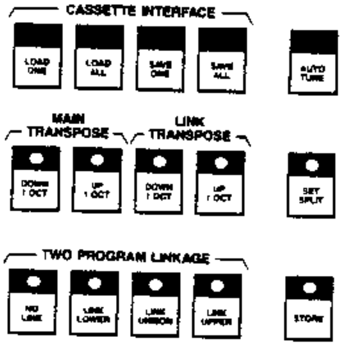
Set the split point at the note played. The split is actually between this note and the next lower note.

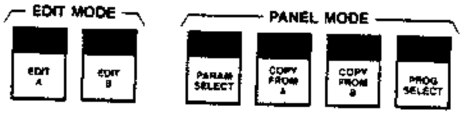
[SET SPLIT] [SET SPLIT]

Set the split point at the same point as after the last time the above command was executed.

[TUNE]

Initiate an autotune. When the tuning is complete, the numbers of any disabled boards will be displayed. If the tune switch is still held when the autotune is complete, the bad boards will still be displayed, but they will all be enabled, for diagnostic purposes.

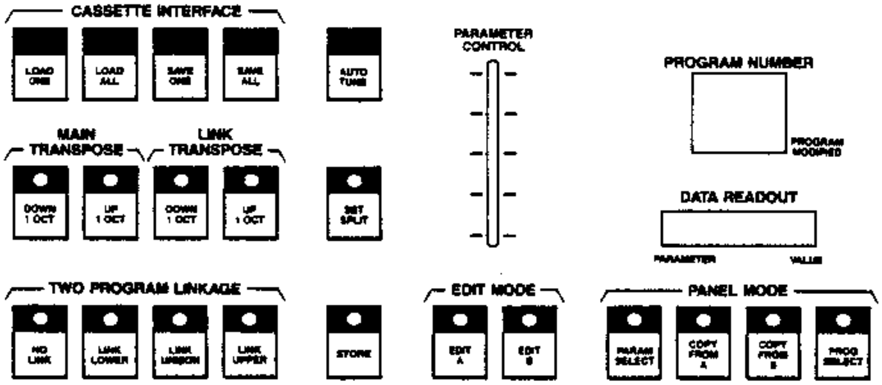




[OVERLOAD] followed by headache

By this point in the manual you are overloaded with technical information. Not to worry, the next edition of the manual will be cleaned up and will hopefully present the information a little more clearly. In the mean time, relax and enjoy the part that makes sense.

PANEL PARAMETER DESCRIPTIONS



PANEL: MAIN TRANSPOSE

This parameter causes -12, 0 or +12 to be added to the key numbers fed into the main program process. The two bits take on the Value 10 for down 1 oct, 00 for off and 01 for up 1 oct. These bits appear in the main transpose LEDs.

PANEL: LINK TRANSPOSE

This parameter is analogous to the main transpose parameter, but only affects notes given to the link process. Do not get confused: the main and link sound generation is controlled by the main and link transposes in the current program. The transposes that are stored in memory in the program that is linked to have no effect.

PANEL: KEYBOARD SPLIT

This parameter is used to determine which keys are assigned to which sounds while in link lower or upper mode. Pressing [SET SPLIT] causes the current setting to appear in the display. Pressing a key will then cause the key number to be stored in this parameter and the display to be restored to what it was before. Pressing [SET SPLIT] twice causes the split to be set to the "standard split," which is simply the most recent split selected by pressing a key on the keyboard. In other words, setting a split using [SET SPLIT] [key] causes the key number to be stored for future access using [SET SPLIT] [SET SPLIT].

PANEL: LINK MODE & PROGRAM NUMBER

Internally, the 2 msbs (most significant bits) contain the mode and the 6 lsbs (least significant bits) contain the number. The 2 msbs take on the value 00 for no link, 10 for link lower, 01 for link upper, and 11 for link unison. The remaining six bits hold the number of the program linked to (or last linked to). Externally, the link mode shows up in the LEDs over the link switches, and the program number shows up in the left half of the 8-digit display whenever the link switches are used to establish a link. Since the link information is a parameter within the program, changing the link constitutes changing the program and sets the modified flag.

PANEL: EDIT MODE & PARAMETER NUMBER

Internally, the 2 msbs contain the edit mode and the 6 lsbs contain the number of the parameter that is connected to the parameter slider. The 2 msbs take on the Value 10 in EDIT A mode, 01 in EDIT B mode, and 11 in EDIT A & B mode. The remaining six bits will be 0 for the link balance parameter, 1-5 for the control parameters, and 6-50 for the A or B parameters. Since the parameter number is itself a parameter in the program, selecting a parameter constitutes changing the program, and will set the modified flag.

PANEL: LINK BALANCE

The link balance parameter has no effect on the sound unless one of the link modes is active. In that case, the link balance represents the relative gain (in 2dB steps) of the main and link program. Set to 0, both programs run at full volume. Setting it negative, the link program is reduced in gain. Setting it positive, the main program is reduced in gain. This is the only panel parameter that uses the parameter control slider. As such, it has its own way of working. Selecting the link balance parameter is done by setting up a link. If a link is already in effect, the link balance parameter may be selected by simply pressing the link switch twice (which is like setting up the link again). As an added convenience, the performer may return to the previously selected parameter after pressing a link switch by pressing [PROG SELECT] or [PARAM SELECT]. While the link balance parameter is selected, the display contains the letter L followed by the link program number in the left side of the display, instead of the usual P and parameter number. This allows a simple means of seeing what program is linked to. This means that the performer can press a link switch to see what he is linked to, and possibly touch up the balance, and then return to the previously selected parameter by pressing [PROG SELECT] or [PARAM SELECT]. Note that the value as shown in the display (the dB value) is twice the internal value as seen by the computer interface. Also, if the parameter is set to -8 (-14 dB) by the interface, it will actually be set to -7.

PANEL: SEQUENCE PROGRAM

This parameter is used to establish which program will be selected next using the sequence program footswitch. Pressing the footswitch causes this parameter to appear in the large 2-digit display. As long as the footswitch is held, pressing one of the 50 numbered switches will cause this parameter (and the large display) to be changed accordingly. It also causes the parameter to be written into the stored program. Releasing the footswitch causes the program to be selected, if in the PROGRAM SELECT mode. In any other mode, the program is not selected, and that the sequence program parameter is set to 20. Pressing the footswitch would cause 20 to appear in the display. Releasing the footswitch would then cause program 20 to be selected. If, on the other hand, the footswitch was pressed and the performer pressed switch number 30, the number 30 would appear in the display, would be written into the current program, and would be written into program 10. Releasing the footswitch would cause program 30 to be selected. Note that this is the only parameter that can be "written" into one of the stored programs with the lock switch locked.

SEQUENCE

PATCH

1

PATCH [1]:

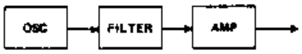
This parameter determines the configuration of the synthesizer channels. There are a total of 16 patch selections, numbered 0 through 15. This is the starting point for all programs. This parameter should be selected and its value set first, to establish the signal paths of the channel boards.

There are basically five configuration types, with subvariations:

SPLIT PATCH
16 independent channels
INDEPENDENT CHANNELS
2 per note, 2 pole, 8 note, 2-pole filtering, independent
PARALLEL FILTERS
Paired channels, notch filtering
SERIES FILTERS
Paired channels, 4 pole and band-pass filter response
VARIABLE MIX
Paired channels, dual 2-pole filter effects

PATCH [1], VALUE = 0 SPLIT PATCH

This value produces the greatest number of independent notes (16). The configuration is the simplest available. In this configuration, parameters set using EDIT B have no effect. The A parameters control all 16 channels.



Applications

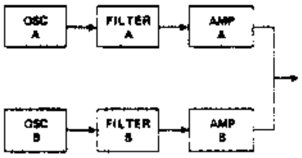
*When many notes are needed, during sustained arpeggios for example.

*Use Value 0 in place of Value 1 with an expander to double the number of notes available. First set the Chroma and the expander to the same program, then set the PATCH value to 0 on both. [SET SPLIT], [1] will temporarily set the patch to 0.

*Simple sounds.

PATCH [1], VALUE = 1 INDEPENDENT CHANNELS

Both channels (A and B) play on each note, but are separately programmed. Complex sounds may be created by using, for example, one channel for a mellow long decaying sound, and the other for a short percussive sound.



Applications

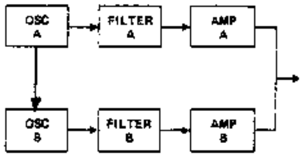
*Set one channel to high-pass filters, the other to low-pass for contrast

*Set channels to the same values (program with EDIT A and EDIT B both on), then detune the B channel using DETUNE [4] for richness.

*Tune filters and pitch differently, e.g. fifth or octave apart.

PATCH [1], VALUE = 2
INDEPENDENT CHANNELS, SYNC

The signal paths of the two channels are independent, but the frequency of the B oscillator is hard synchronized to the A oscillator. The A oscillator provides the fundamental frequency, the B oscillator the harmonic frequency. For traditional synthesizer "sync" effects, modulate the pitch of the B oscillator with an envelope generator or a sweep generator.



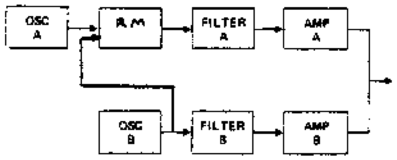
Applications

*Synthesizer "sync" sounds

*Synchronized oscillator sounds, 2-pole filter with a normal channel for added bottom.

PATCH [1], VALUE = 3
INDEPENDENT CHANNELS, RING MODULATOR

Same as Value = 1, except that a ring modulator is substituted for oscillator A. This permits a ring modulator to be used in addition to a normal channel. To set up the ring modulator, set WIDTH [34] to about 32 on both channel A and B' Vary the B TUNE [26] and the DETUNE [4] to obtain the desired ring modulator effect.



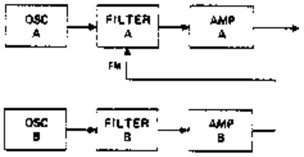
Applications

*Bell sounds, effects, metallic sounds

*Ring modulator sounds with a normal synth channel

PATCH [1], VALUE = 4
INDEPENDENT CHANNELS, FILTER FM

The channels are independent, but the output of the B channel modulates the A filter. Audio frequency modulation of filters create rich and unique harmonics. Selecting a different B channel TUNE [26] value, and different B channel WAVESHAPE [33] will produce different complex effects.



Applications

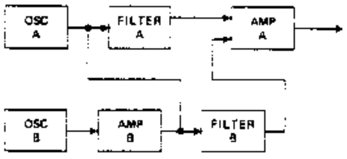
*Phase shifter effects

*Vocal effects

*Using noise as a wavshape, percussion

PATCH [1], VALUE = 5
PARALLEL FILTERS

Both the A and the B oscillators are routed to both A and B filters. The B oscillator's volume is controlled by the B amplifier, and the A amplifier governs the entire output volume. To use as a notch filter, set one filter to high-pass (LP/HP [37], value 1), and the other to low-pass (LP/HP [37], value 0). Set the TUNE [39] value on the LP channel to a low value and the other to a higher value.



Applications

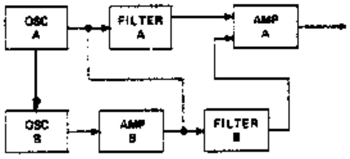
*Use for clavinet, or harpsichord-type sounds

*Modulate with a sweep control for flanging effects.

PATCH
1

PATCH [1], VALUE = 6
PARALLEL FILTERS, SYNC

Same as Value 5, but B oscillator is synchronized with the A oscillator (see Value 3 for more about sync).

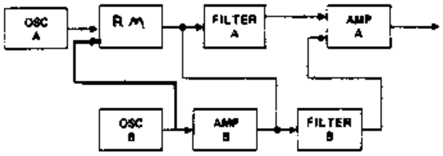


Applications

*Complex filtering of synchronized oscillators by tuning each filter separately.

PATCH [1], VALUE = 7
PARALLEL FILTERS, RING MODULATOR

Same as Value 5, but with ring modulator substituted for the A oscillator.

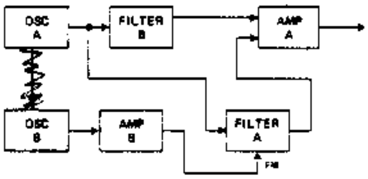


Applications

*Ring modulator effects with notch filter, bells

PATCH [1], VALUE = 8
PARALLEL FILTERS, FILTER FM

Same as Value 5, except the output of amplifier B is routed to the A filter control input for frequency modulation.

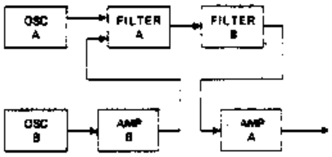


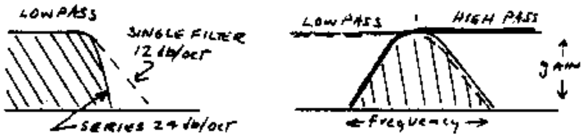
Applications

*Complex harmonics

PATCH [1], VALUE = 9
SERIES FILTER

The two oscillators feed the A filter which in turn feeds the B filter. The B amplifier controls the output of oscillator B, the A amplifier controls the overall output volume. This configuration creates a four-pole filter response, which is useful for brass or flute sounds. The higher harmonics are filtered more than in the single filtered configuration, producing better mellow sounds. The resonance settings of the A and B filters may be set to different values to increase the resonance range.





A band-pass filter response can be established by setting one of the filters to high-pass (LP/HP [37], Value = 1), and the other to low-pass (LP/HP [37], Value = 0). Start by setting the high-pass filter slightly higher than the setting of the low-pass filter.

Applications

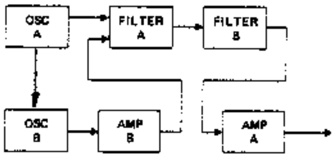
*Brass, flutes, "fat sounds," many orchestra sounds

*Band-pass; solo violins, reeds

*Band-pass; vocal sounds

PATCH [1], VALUE = 10
SERIES FILTER - SYNC

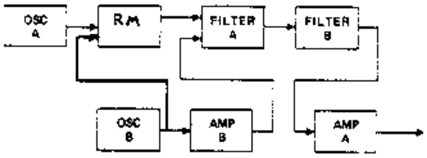
Same as Value 9, except the B oscillator is synchronized to the A oscillator. To mute the A oscillator (to hear only the B oscillator), set the A WAVESHAPE [33] to 1 (pulse), set WIDTH [34] to 0. Set the B TUNE [26] value for best effects.



Applications
*Fat synthesizer sync sound

PATCH [1], VALUE = 11
SERIES FILTER, RING MOD

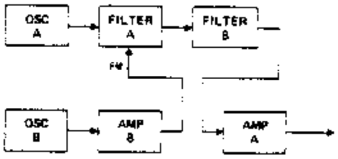
Same as Value 9, except that the A oscillator is replaced with a ring modulator (see Value 3 for ring mod set up). To mute the B oscillator, set the values of MOD [1] DEPTH [47] and MOD [2], DEPTH [49] on the B channel both to 0.



Applications
*Ring modulator, bell effects with four-pole filter

PATCH [1], VALUE = 12
SERIES FILTER, FM MODULATION

Same as Value 9, but the output of the B amplifier modulates the frequency of the A filter.



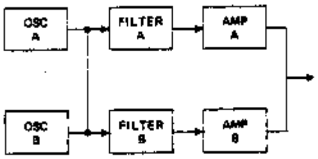
Applications
*Rich harmonics, fat sounds
*Vocal sounds

PATCH

1

PATCH [1], VALUE = 13
VARIABLE MIX FILTERS

The oscillators are mixed and then fed into both filters. Each amplifier controls the level from each filter, allowing asymmetrical notches to be created.



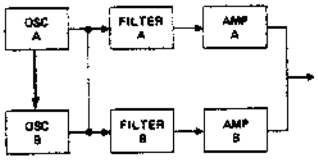
Applications

*Phase shifter, flanger effects (sweeping notch)

*Duo timbre by voicing each channel separately in low-pass mode.

PATCH [1], VALUE = 14
VARIABLE MIX FILTERS, SYNC

Same as Value 13, except the B oscillator is synchronized to the A oscillator (see Value 2 for sync description).



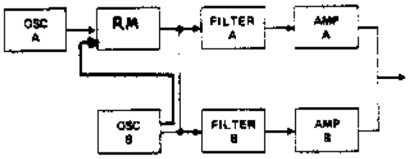
Applications

*Sync effects with notch

*Duo timbre effects

PATCH [1], VALUE = 15
VARIABLE MIX FILTERS, RING MOD

Same as Value 13, except the A oscillator is replaced by a ring modulator.



Applications

*Ring modulator, bell effects with notch

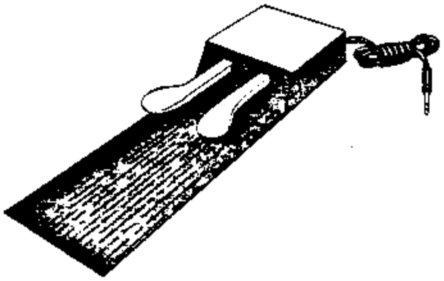
*Duo timbre effects

FSW MODE
2

FOOTSWITCH MODE [2]

This parameter establishes the mode of the dual footswitch assembly. The right switch is FSW 1 and the left switch is FSW 2. The right switch is normally used as a sustain pedal, and the left switch as a latch. There are eight different modes which change the operation of these switches:

	SUSTAIN	LATCH
VALUE	FSW 1 (R)	FSW 2 (L)
0	Enabled	Enabled
1	Disabled	Enabled
2	Enabled	Disabled
3	Disabled	Disabled
4	Enabled	Note Gate
5	Disabled	Note Gate
6	Enabled	Inverted Gate
7	Disabled	Inverted Gate



When NOTE GATE is selected, notes are only heard when the Left Pedal is held down. When INVERTED NOTE GATE is used, notes are only heard when the left pedal is not held down.

These modes are useful when set differently for Main and Link programs. For example, when using a LINK LOWER set up, it may be desirable to disable the sustain function for the lower part of the keyboard [LINK], and to disable the latch function on the upper part. For this example, the Main program should have FSW MODE [2], Value = 2, and the Link program should be FSW MODE [2], Value = 1.

Note that the operation of either footswitch may be reversed by powering the Chroma with the switch held down, or by pressing [SET SPLIT] [50] while holding down the switch. This makes the switch operate as though it is normally on instead of normally off. Press [SET SPLIT] [50] to restore the switches to their normal condition.

KYBD ALG
3

KEYBOARD ALGORITHM [3]

This parameter defines the way in which the Chroma assigns synthesizer channels to notes. There are two groups of assignment modes: Monophonic and Polyphonic. The monophonic modes use only one pair of channels, leaving seven pairs to be used for another program. The Polyphonic modes will use all available channels.

The monophonic modes are usually setup for a program which is linked to a polyphonic program. This allows the polyphonic program to use the maximum number of channels [14], and have a lead or bass line which is programmed to a different sound.

KYBD ALG [3], VALUE = 0
POLYPHONIC - LEAST RECENTLY USED

A note is assigned first to a channel set to the same note. Restriking a key will use the same channel. A channel will be assigned second to the longest released channel (the oldest note). The longest held note will be assigned third. The left footswitch latches notes already held, allowing subsequent notes to be played. When a note is latched (and held), a new channel will be assigned to the same note if played.

KYBD ALG [3], VALUE = 1
PITCH ORDERED

Chords are assigned from the top note down to the most recently released channels. This algorithm is most useful with polyphonic glides, as it will permit chords to slide from one position on the keyboard to another with the notes in the same order.

NOTE: When using this value, the LEFT FOOTSWITCH will function as a GLIDE ON/OFF. The left pedal MUST BE DEPRESSED for the glide to function.

KYBD ALG [3], VALUE = 2
POLYPHONIC CHORD BUFFER

Playing a chord causes the notes to be remembered (but not played). When the Left footswitch is pressed, the chord will play. The chord buffer remains intact until all keys are released and a new key is depressed, allowing a new chord to be entered while the previous one still sounds.

This mode is useful for "cueing" a chord during performance. The chord may be loaded when time permits, and "played" with the footswitch while playing a different instrument. Also, the chord buffer may be used in a program which is linked to another program. This way, a chord can be played and then "echoed" by another program when the footswitch is depressed.

KYBD ALG [3], VALUE = 3
ALL CHANNELS (POLYPHONIC)

The most recently played note will be assigned to all channels so that all synthesizer channels play at the same time. If two notes or more are held, the channels are divided among the keys played (up to eight notes, or sixteen if PATCH [1] = 0).

KYBD ALG [3], VALUE =4
ALL CHANNELS (MONOPHONIC)

Only the most recently played note is heard, and is assigned to all available channels.

KYBD ALG [3], VALUE = 5
MONOPHONIC LAST NOTE, SINGLE TRIG

Only the most recently played note will be heard. The envelope generators are single triggered; they will not trigger unless all keys are first released. (Only one pair of channels is used.)

KYBD ALG [3], VALUE = 6
MONOPHONIC LAST NOTE, MULT TRIG

Only the most recently played note will be heard. The envelope generators are multiple triggered; they will trigger on every key depression.

KYBD ALG [3], VALUE =7
MONOPHONIC FIRST NOTE

The first note of a group is heard. All keys must be released in order to hear the next note. Good for bringing out inner voices when linked to polyphonic programs.

KYBD ALG [3], VALUE = 8
MONOPHONIC BOTTOM NOTE

The lowest note of any chord is heard. Good for bass lines when linked to polyphonic programs.

KYBD ALG [3], VALUE = 9
MONOPHONIC TOP NOTE

Only the top note of a chord is heard. Good for lead line programs when linked to polyphonic programs.

KYBD ALG [3], VALUE = 10
MONOPHONIC ARPEGGIATE UP

The notes of a held chord will arpeggiate up at a rate determined by the A Sweep RATE [9]. The notes will be stored with key velocity information as long as the sustain footswitch or a key is held. New velocity values may be established by restriking notes. Latch (left) footswitch will latch the arpeggiating notes.

KYBD ALG [3], VALUE = 11
MONOPHONIC ARPEGGIATE DOWN

Same as Value 10, but arpeggiates down.

KYBD ALG [3], VALUE = 12
MONOPHONIC ARPEGGIATE UP AND DOWN

Same as Value 10, but arpeggiates up, then down.

KYBD ALG [3], VALUE = 13
MONOPHONIC ARPEGGIATE DOWN AND UP

Same as Value 10, but arpeggiates down, then up.

KYBD ALG [3], VALUE = 14
MONOPHONIC SEQUENCE ARPEGGIATION

Notes are played and remembered in the order in which they were played. They are heard at a rate set by the A Sweep RATE [9]. Keyboard veloicty values are remembered, and the memory length is about 195 notes. Notes will be remembered as long as the sustain footswitch or a key is held, or as long as the latch switch is depressed. For as long as a key is depressed, or as long as the sustain is held, notes may be added to the list. Releasing all keys and the sustain stops the sequence, and a new sequence may be loaded.

KYBD ALG [3], VALUE = 15
MONOPHONIC RANDOM ARPEGGIATION

Notes played are remembered and heard randomly at a rate set by A Sweep RATE [9]. Playing a note more than once increases its chances of being played.

DETUNE
4

DETUNE [4], VALUE = 0 to 31

Tunes the B channel oscillator sharp with respect to the A channel. The displayed value is the number of 32nds of a semitone sharp. Since this parameter only affects the B channel, it has no effect when PATCH [1], Value 0 is selected.

Applications

*Use to detune the A and B channels for chorus effect.

*Use with TUNE [26] as a fine tune control when using a ring modulator patch.

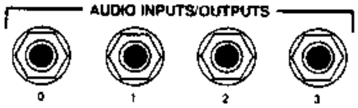
OUTPUT SELECT
5

OUTPUT SELECT [5], VALUE = 0 to 3

Routes the channels to one of four audio outputs on the rear panel (0, 1, 2, or 3). The four outputs are normally summed together and sent out the mono output jacks via the rings of these jacks. The jacks may be used as send/receive jacks by inserting a ¼" phono plug: send the signal to the effect (such as a phaser, reverb, etc.) from the tip of the plug, and route the output of the effect back into the ring.

To setup stereo modes automatically (from within a program), set the OUTPUT SELECT on the Main program to one output, and the OUTPUT SELECT on the Link program to a different value. Store each program after making the change. Note: to use both XLR output jacks, it is suggested that the Link program be set to OUTPUT SELECT [5], Value 3.

The OUTPUT SELECT routes the outputs of channel pairs, and the routing is established for the entire program. In other words, both the A and B channels will always be routed to only one output. Two outputs may be used at the same time only when a Link is established.



RATE

6

GLIDE PARAMETERS

Keyboard pitch information is passed through the glide processor, which is capable of slowing down the transitions from pitch to pitch. There are two modes: portamento and glissando. Portamento creates a smooth transition from any pitch to another. In both cases, the rate of change is controlled by the RATE [6] parameter.

Certain keyboard algorithms automatically enable and disable the glide according to how the notes are played.

RATE [6], VALUE 0 to 31

The rate settings for the glide depend upon the GLIDE SHAPE:

IF SHAPE = PORTAMENTO
Value 0 = No Portamento
1 = 10 millisecond portamento
31 = 10 second portamento

IF SHAPE = GLISSANDO
Value 0 = No Glissando
1 = 25 steps per second
31 = 1 step per second

SHAPE

7

SHAPE [7], VALUE = 0 to 1

Selects the shape of the glide output:

Value 0 = PORTAMENTO (Smooth slide)
Value 1 = GLISSANDO (Chromatic steps)

The sweep generator generates low-frequency repetitive control signals. It is used to modulate functions like pitch (for vibrato, or trills), waveform (pulse width modulation), cutoff (tremolo). It has a basic rate that can be adjusted over a wide range. In addition, it can be modulated by 15 other control sources. There are 16 different waveshapes available, and an amplitude modulation selection which includes a dealyed sweep envelope. The sweep generator may run independently, or be synchronized to key depressions. Each note may have its own sweep generator, or they may be locked together.

There are a total of 16 separate sweep generators (A and B), and both may be used with different settings within a program. The A sweep generator is used to trigger other functions, such as the arpeggiator used in KYBD ALG [3].

MODE

8

MODE [8], VALUE, = 0 to 3

Controls the type of synchronization among a bank of sweeps.

VALUE = 0: ASYNCHRONOUS

The sweeps are free running and independent. In fact, the computer forces them to run at slightly different rates. Use when a high degree of note independence is desirable, such as a string chorus.

VALUE = 1: INDIVIDUAL KEY-SYNCED

The sweeps are independent of each other, yet each is restarted at the beginning of its cycle when the channel is assigned a new note. Use with patterns or trills so as to obtain a predictable response when keys are first played. When [AMPL MOD] is set to one of the delayed sweep values, the [MODE] value will automatically default to Value 1. The value in the DATA READOUT will not be changed, however.

VALUE = 2: SINGLE FREE RUNNING

All the sweeps (from all channels) function as one, and run freely. The rate modulation always comes from the most recently assigned channel. In this mode, it is like having only one sweep generator for the entire instrument and is useful for synchronizing specific sweep effects like patterns.

VALUE = 3: SINGLE KEY-SYNCED

Same as Value 3, except that the sweep is restarted at the beginning of its cycle when new notes are played.

RATE

9

SWEEP RATE [9], VALUE 0 through 63

Controls the sweep rate (unmodulated) 0 to 63, (0.12Hz. to 12Hz.).

RATE MOD 10

SWEEP RATE MOD [10], VALUE 0 through 15

Selects the modulation of the sweep rate. The amount of modulation is fixed at a predetermined value. Positive modulation increases the rate, inverted values decrease the rate.

Value 0: No modulation
1: Pressure (optional)
2: Keyboard (higher notes increase the rate, lower notes decrease the rate).
3: Inverted Keyboard (higher notes decrease rate).
4: Velocity (notes struck hard increase the rate).
5: Inverted Velocity (notes struck hard decrease the rate).
6: Envelope 2 (rate increases and decreases according to envelope shape).
7: Inverted Envelope (rate decreases, and increases according to envelope shape).
8: Pedal 1 (pedal increases sweep rate).
9: Inverted Pedal 1 (Pedal decreases rate).
10: Pedal 2 (pedal increases sweep rate).
11: Inverted Pedal 1 (pedal decreases rate).
12: Lever 1 (pushing lever forward increases rate).
13: Inverted Lever 1 (pushing lever forward decreases rate).
14: Lever 2 (pushing lever forward increases rate).
15: Inverted Lever 2 (pushing lever forward decreases rate).

WAVE SHAPE 11

SWEEP WAVESHAPE [11], VALUES = 0 through 15

Establishes the waveshape generated by the sweep. The quantities are in standard Chroma pitch units (1 unit = 32 semitones).

VALUE	DESCRIPTION	TYPICAL USES
0	*SINE*	*Vibrato, tremolo*
1	*COSINE*	"
2	*OFFSET SINE*	*Filter mod, waveshape mod*
3	*HALF SINE*	"
4	*TRIANGLE A*	"
5	*TRIANGLE B*	"
6	*SAWTOOTH*	*Pitch or Freq mod, Effects*
7	*LAG SQUARE*	*Effects*
8	*SQUARE*	*Trills*
9	*PATTERN A*	*Repeat patterns, S/H effects*
10	*PATTERN B*	"
11	*PATTERN C*	"
12	*PATTERN D*	"
13	*PATTERN E*	"
14	*PATTERN F*	"
15	*RANDOM*	*S/H effects*

AMPL
MOD
12

AMPLITUDE MODULATION [12]
VALUES = 0 through 15

Controls the modulation of the sweep amplitude. The waveshapes generated by the sweep generator normally have a fixed amplitude. When these waveforms are used to modulate other functions, such as pitch or cutoff, the depth modulation is set and is not changed thereafter. The sweep **AMPLITUDE MODULATION** permits the sweep waveform's amplitude to be modified at the sweep generator source by any of 15 other control functions. The depth of **AMPLITUDE MODULATION** is fixed.

Value 0: No modulation
1: Pressure (optional pressure sensor)
2: Keyboard (amplitude increases with higher notes).
3: Inverted Keyboard (amplitude decreases with higher notes).
4: Velocity (amplitude increases with harder notes).
5: Inverted Velocity (amplitude decreases with harder notes).
6: Envelope 1 (Env 1 increases amplitude).
7: Inverted Envelope 1 (Env 1 decreases amplitude).
8: Pedal 1 (pedal increases amplitude).
9: Pedal 2 (pedal decreases amplitude).
10: Lever 1 (lever 1 increases amplitude).
11: Lever 2 (lever 2 increases amplitude).
12: 0.85 sec. delay (delayed envelope).
13: 1.3 sec. delay (delayed envelope).
14: 2.6 sec. delay (delayed envelope).
15: 5.1 sec. delay (delayed envelope).

The delay envelopes may be used for creating an automatic delayed vibrato. There are four delay lengths to choose from. The four delay values will automatically put the sweep in the independent key triggered mode, even if **MODE** [8] is set to a different value.

ENVELOPE PARAMETERS

There are two envelope generators per channel, labelled 1 and 2. This means there are two envelopes for the A channel (1A and 1B), and two for the B channel (2A and 2B). The envelope shapes are either AR (Attack, Release), or ADR (Attack, Decay, Release). More complex shapes are created by combining envelopes by using modulation inputs to the Cutoff, Pitch or Amplitude functions. Each envelope has the following parameters which may be varied:

AMPLITUDE TOUCH

Sets the envelope peak as a function of key velocity.

ATTACK TIME

Sets the ATTACK TIME from key down to envelope peak.

ATTACK MODULATION

Alters the ATTACK TIME value as a function of other control sources such as sweep or keyboard.

DECAY TIME

Sets the DECAY TIME from the peak of the envelope.

DECAY MODULATION

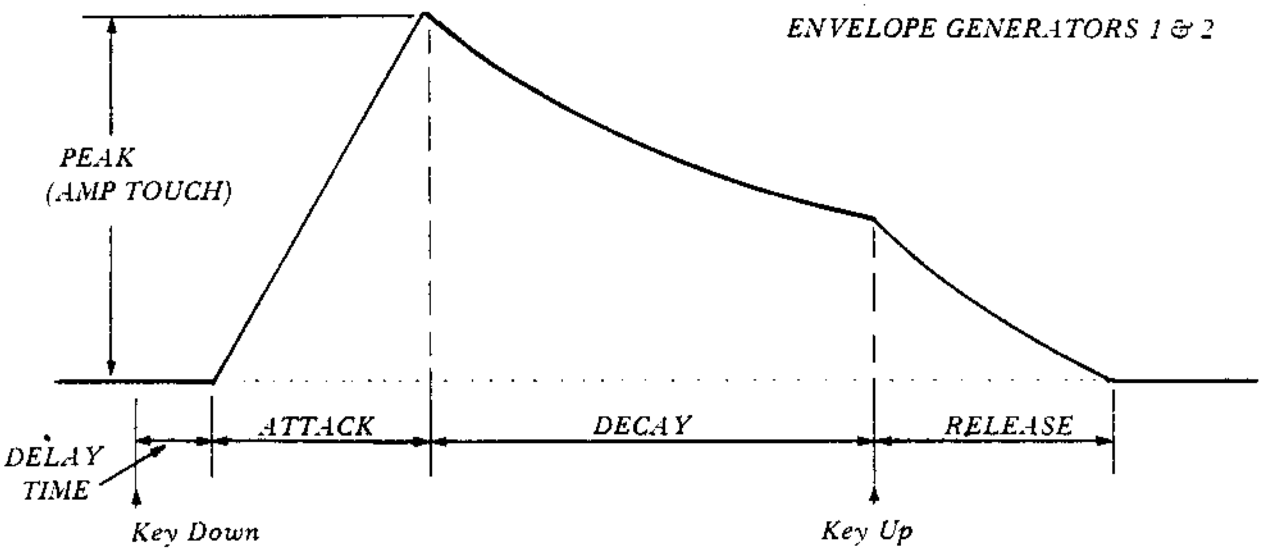
Alters the DECAY TIME value as a function of other control sources.

RELEASE TIME

Sets the RELEASE TIME from the time a key is released.

DELAY (Env No. 2 only)

Delays the onset of an envelope from the time a key is pressed from 0 up to 3 seconds.



AMPL TOUCH
13

AMPLITUDE TOUCH [13], VALUES = 0 through 7

Controls the relationship between keystrike velocity and the envelope amplitude (peak). When a key is struck, the Chroma records the key velocity in memory. Depending upon the setting of the AMPLITUDE TOUCH, this velocity value can produce different results. When the value is set to 0, there is no effect, and the amplitude is constant regardless of how hard keys are struck. Values 1 through 5 produce different degrees of sensitivity as a function of key velocity; Values 6 and 7 produce two amplitude values switched at a present velocity threshold (see below).

Value 0: No amplitude modulation
1: Low sensitivity
2: Medium low sensitivity
3: Medium sensitivity
4: Medium high sensitivity
5: High sensitivity
6: THRESHOLD
7: INVERTED THRESHOLD

THRESHOLD VALUES:

For Values 6 and 7, the point above which the keystrike is considered a "hard strike" is called the attack threshold. Above this threshold, envelopes will have the normal fixed amplitude (1 "Chroma" unit); below the threshold, the amplitude will be zero (no envelope). Using [SET SPLIT], [21] causes the threshold to be set to whatever number is in the DATA READOUT value, regardless of what parameter is selected. Setting the threshold close to zero will cause all but the very softest keystrikes to be considered "hard strikes." Setting the threshold close to 31 will cause all but the very hardest strikes to be considered "soft strikes." Values from 10 to 25 are useful.

Using the threshold values, programs may be established which cause the envelopes to "switch" from one type to another as a function of how hard you play. This on hard strikes, or to mute a sound as you strike harder.

ATTACK
14

ATTACK [14], VALUES = 0 through 21

Controls the unmodulated time the envelope takes to reach its peak value.

Value 0: Instant
1: 10 msec.

31: 10 seconds

ATTACK MOD
15

ATTACK MOD [15], VALUES = 0 through 7

Controls the modulation of the attack time. The amount of modulation is fixed at a (hopefully) useful value. Positive modulation decreases the attack time; negative modulation increases the attack time.

Value 0: No modulation
1: Pressure (optional)
2: Keyboard (attack time decreases with higher notes).
3: Inverted Keyboard (attack time increases with higher notes).
4: Velocity (attack time decreases with hard keystrikes).
5: Inverted Velocity (attack time increases with hard keystrikes).
6: Pedal 1
7: Pedal 2

DECAY 16

DECAY [16],
VALUES = 0 through 31

Controls the unmodulated decay time constant. When set to maximum, the envelope will remain at peak value for as long as a key is held.

Value 0: Instant
1: 10 msec.

31: 10 seconds
32: Infinite sustain

DECAY MOD 17

DECAY MOD [17],
VALUES = 0 through 7

Controls the modulation of the decay time constant. The amount of modulation is fixed at a (hopefully) useful value. Positive modulation increases the decay time; negative modulation decreases the decay time.

Value 0: No modulation
1: Pressure (optional)
2: Keyboard (the decay time increases with higher notes on the keyboard).
3: Inverted Keyboard (the decay time decreases with higher notes on the keyboard).
4: Velocity (the decay time increases with hard keystrikes).
5: Inverted Velocity (the decay time decreases with hard keystrikes).
6: Pedal 1
7: Pedal 2

RELEASE 18

RELEASE [18]
VALUES = 0 through 31

Controls the release time constant. Value 31 has two preset release times (see below).

Value 0: Instantaneous
1: 10 msec.

30: 10 seconds
31: THRESHOLD RELEASE (see below)

RELEASE THRESHOLD [SET SPLIT], [22]

When the envelope's release parameter is set to 31, the release time constant becomes touch sensitive. Release velocities below a certain threshold yeild one release time while velocities above the threshold yeild a different release time. The threshold is a number from 0 to 13, and is set by pressing [SET SPLIT], [22]. Pressing [SET SPLIT], [22] causes the threshold to be set to whatever number is in the DATA READOUT (value), regardless of what parameter is selected.

SLOW RELEASE RATE [SET SPLIT], [23]

Using [SET SPLIT], [23] sets the slow release rate. As with the threshold, pressing [SET SPLIT], [23] causes the release time for slow key release to be set to whatever number is in the DATA READOUT, regardless of the parameter selected. The range is from 0 to 31.

FAST RELEASE RATE [SET SPLIT], [24]

Using [SET SPLIT], [24] sets the fast release rate. This is the rate at which an envelope will release for a fast key release. Pressing [SET SPLIT], [24] causes the release time for fast release notes to be set to whatever number is in the DATA READOUT, regardless of the parameter selected. The range is from 0 to 31.

DELAY
19

ENVELOPE 2

DELAY [19], VALUES 0 through 31

Delays the onset of envelope 2 up to 2.4 seconds, or can cause the A sweep generator to trigger the onset of envelope 2 at the beginning of each sweep cycle.

Value 0: No delay
1: 80 msec. delay
(80 msec. increments)
30: 2.4 seconds
31: A sweep triggered

AMPL TOUCH
20

AMPLITUDE TOUCH [20], VALUES = 0 through 7

Controls the relationship between keystrike velocity and the envelope amplitude (peak). When a key is struck, the Chroma records the key velocity in memory. Depending upon the setting of the **AMPLITUDE TOUCH**, this velocity value can produce different results. When the value is set to 0, there is no effect, and the amplitude is constant regardless of how hard keys are struck. Values 1 through 5 produce different degrees of sensitivity as a function of key velocity; Values 6 and 7 produce two amplitude values switched at a present velocity threshold (see below).

Value 0: No amplitude modulation
1: Low sensitivity
2: Medium low sensitivity
3: Medium sensitivity
4: Medium high sensitivity
5: High sensitivity
6: THRESHOLD
7: INVERTED THRESHOLD

THRESHOLD VALUES:

For Values 6 and 7, the point above which the keystrike is considered a "hard strike" is called the attack threshold. Above this threshold, envelopes will have the normal fixed amplitude (1 "Chroma" unit); below the threshold, the amplitude will be zerc (no envelope). Using [SET SPLIT], [21] causes the threshold to be set to whatever number is in the DATA READOUT value, regardless of what parameter is selected. Setting the threshold close to zero will cause all but the very softest keystrikes to be considered "hard strikes." Setting the threshold close to 31 will cause all but the very hardest strikes to be considered "soft strikes." Values from 10 to 25 are useful.

Using the threshold values, programs may be established which cause the envelopes to "switch" from one type to another as a function of how hard you play. This on hard strikes, or to mute a sound as you strike harder.

ATTACK
21

ATTACK [21], VALUES = 0 through 21

Controls the unmodulated time the envelope takes to reach its peak value.

Value 0: Instant
1: 10 msec.

31: 10 seconds

ATTACK MOD
22

ATTACK MOD [22], VALUES = 0 through 7

Controls the modulation of the attack time. The amount of modulation is fixed at a (hopefully) useful value. Positive modulation decreases the attack time; negative modulation increases the attack time.

Value 0: No modulation
1: Pressure (optional)
2: Keyboard (attack time decreases with higher notes).
3: Inverted Keyboard (attack time increases with higher notes).
4: Velocity (attack time decreases with hard keystrikes).
5: Inverted Velocity (attack time increases with hard keystrikes).
6: Pedal 1
7: Pedal 2

DECAY
23

DECAY [23], VALUES = 0 through 31

Controls the unmodulated decay time constant. When set to maximum, the envelope will remain at peak value for as long as a key is held.

Value 0: Instant
1: 10 msec.

31: 10 seconds
32: Infinite sustain

DECAY MOD
24

DECAY MOD [24], VALUES = 0 through 7

Controls the modulation of the decay time constant. The amount of modulation is fixed at a (hopefully) useful value. Positive modulation increases the decay time; negative modulation decreases the decay time.

Value 0: No modulation
1: Pressure (optional)
2: Keyboard (the decay time increases with higher notes on the keyboard).
3: Inverted Keyboard (the decay time decreases with higher notes on the keyboard).
4: Velocity (the decay time increases with hard keystrikes).
5: Inverted Velocity (the decay time decreases with hard keystrikes).
6: Pedal 1
7: Pedal 2

RELEASE
25

RELEASE [25], VALUES = 0 through 31

Controls the release time constant. Value 31 has two preset release times (see below).

Value 0: Instantaneous
1: 10 msec.

30: 10 seconds
31: THRESHOLD RELEASE (see below)

RELEASE THRESHOLD [SET SPLIT], [22]

When the envelope's release parameter is set to 31, the release time constant becomes touch sensitive. Release velocities below a certain threshold yeild one release time while velocities above the threshold yeild a different release time. The threshold is a number from 0 to 13, and is set by pressing [SET SPLIT], [22]. Pressing [SET SPLIT], [22] causes the threshold to be set to whatever number is in the DATA READOUT (value), regardless of what parameter is selected.

SLOW RELEASE RATE [SET SPLIT], [23]

Using [SET SPLIT], [23] sets the slow release rate. As with the threshold, pressing [SET SPLIT], [23] causes the release time for slow key release to be set to whatever number is in the DATA READOUT, regardless of the parameter selected. The range is from 0 to 31.

FAST RELEASE RATE [SET SPLIT], [24]

Using [SET SPLIT], [24] sets the fast release rate. This is the rate at which an envelope will release for a fast key release. Pressing [SET SPLIT], [24] causes the release time for fast release notes to be set to whatever number is in the DATA READOUT, regardless of the parameter selected. The range is from 0 to 31.

PITCH PARAMETERS

TUNE
26

TUNE [26],
VALUES = 0 through 63

Controls the unmodulated pitch of the oscillator relative to the glide keyboard output. The range is in semitones, where 12 represents concert pitch. Therefore, the range is from 1 octave below concert pitch to 4 1/4 octaves above.

MOD 1 SELECT
27

MODULATION 1 SELECT [27],
VALUES = 0 through 15

Selects a control signal for the No. 1 modulation input into the oscillator.

Value	0:	Keyboard Glide A
	1:	Sweep A
	2:	Envelope 1A
	3:	Envelope 2A
	4:	Keyboard Glide B
	5:	Sweep B
	6:	Envelope 1B
	7:	Envelope 2B
	8:	Lever 1
	9:	Lever 2
	10:	Pedal 1
	11:	Pedal 2
	12:	Velocity
	13:	Threshold velocity
	14:	Pressure
	15:	Threshold pressure

Note: If the split patch is being used, selections 4–7 default to the A sources, as there are no B sources.

MOD 1 DEPTH
28

MODULATION 1 DEPTH [28]
VALUES = -64 through +63

Adjusts the gain depth of modulation for the No. 1 modulation input into the oscillator. The value represents the number of 1/16 semitone increments for each unit of modulation.

Note: The three pitch modulation depths have different gain ranges.

MOD 2 SELECT
29

MODULATION 2 SELECT [29]
VALUES = 0 through 15

Selects a control signal for the No. 2 modulation input into the oscillator.

(SAME AS PARAMETER [27])

MOD 2 DEPTH
30

MODULATION 2 DEPTH [30],
VALUES = -64 through +63

Adjusts the gain for the No. 2 modulation input into the oscillator. The value represents the number of 1/4 semitone increments for each unit of modulation.

Note: The three pitch modulation depths have different gain ranges.

MOD 3 SELECT
31

MODULATION 3 SELECT [31],
VALUES = 0 through 15

Selects a control signal for the No. 3 modulation input into the oscillator.

(SAME AS PARAMETER [27])

MOD 3 DEPTH
32

MODULATION 3 DEPTH [32],
VALUES = -64 through +63

Adjusts the gain for the No. 3 modulation input into the oscillator. The value represents the number of semitone increments for each unit of modulation.

Note: The three pitch modulation depths have different gain ranges.

WAVE SHAPE
33

WAVESHAPE [33], VALUES = 0 through 3

Selects the waveshape produced by the oscillator.

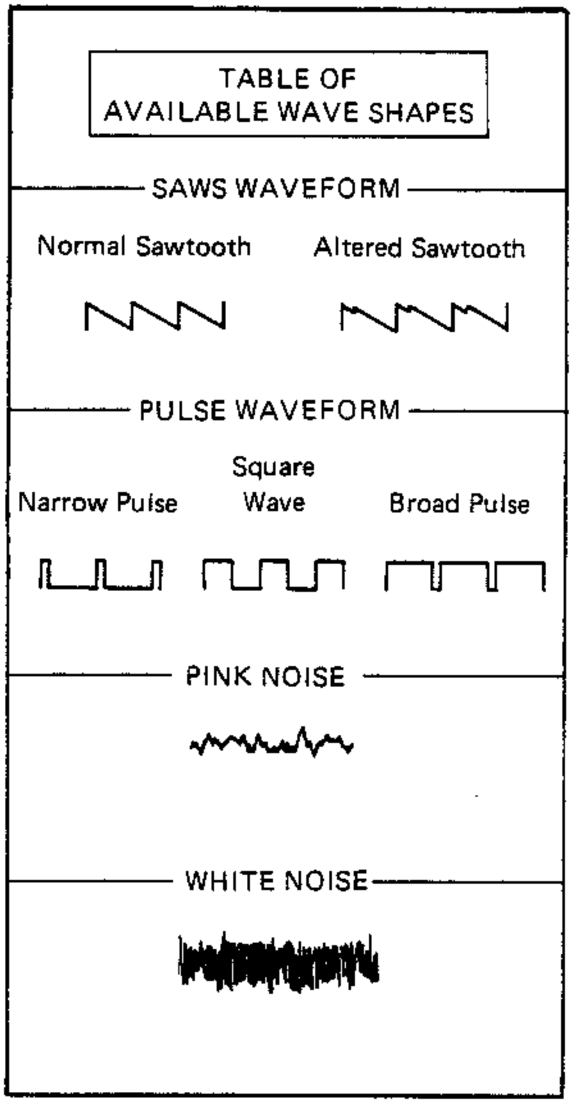
Value 0: Saws: This shape is the equivalent of the sum of two time-shifted sawteeth. It is created by mixing the basic sawtooth signal with a variable width pulse derived from it.

1: Pulse: This shape is the equivalent of the difference between two time-shifted sawteeth. It is derived from the sawtooth. It is DC restored by mixing it with the pulse width control signal.

2: Pink Noise: The pink noise generator is used as the signal source, and the oscillator is not used. All channels are fed from a single noise generator.

3: White Noise: The white noise generator is used as the signal source, and the oscillator is not used. All channels are fed from a single noise generator.

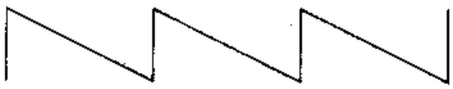
- Pulse Output: When the WIDTH [34] value is set to 0 and the WAVESHAPE [33] is set to pulse (1), the oscillator will have no output. This is handy as a means of turning off the oscillator, for example when it is being used to modulate a filter, or for ring modulator effects when you may not want to hear the unmodulated oscillator.
- Sawtooth: To obtain a normal sawtooth waveform, set the WIDTH [34] to 0. If the value is greater than 0, the pulse output will mix with the sawtooth producing a different (and often useful) shape.
- Ring Modulator: If one of the ring mod patches is selected, the waveshape parameter on the A channel has no effect, as the A oscillator signal is replaced by the binary ring mod of the two oscillators' pulses.



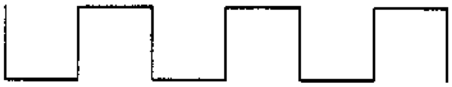
WIDTH 34 PULSE WIDTH [34], VALUES = 0 through 63

Adjusts the unmodulated pulse width or phase difference between the two sawteeth. The range is in increments of roughly 1.5%. A setting of 32 represents a width of 50%, which yeilds either a sawtooth of twice the oscillator frequency or a square wave depending upon the waveform value. A setting of 0 represents a width of 0%, which yeilds either a single sawtooth or no pulse.

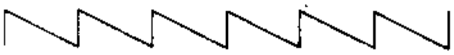
Basic Sawtooth Wave



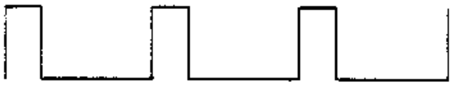
Square Wave



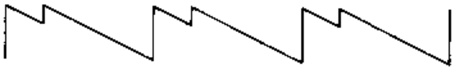
Sawtooth Wave altered by Square Wave



Narrow Pulse Wave



Sawtooth Wave altered by Narrow Pulse Wave



MOD SELECT 35 PULSE MODULATION SELECT [35], VALUES = 0 through 15

Selects a control signal for the modulation input into the waveshaper.

Value	0:	Keyboard Glide A
	1:	Sweep A
	2:	Envelope 1A
	3:	Envelope 2A
	4:	Keyboard Glide B
	5:	Sweep B
	6:	Envelope 1B
	7:	Envelope 2B
	8:	Lever 1
	9:	Lever 2
	10:	Pedal 1
	11:	Pedal 2
	12:	Velocity
	13:	Threshold velocity
	14:	Pressure
	15:	Threshold pressure

Note: If the split patch is being used, selections 4–7 default to the A sources, as there are no B sources.

MOD DEPTH 36 PULSE MODULATION DEPTH [36], VALUES = -64 through +63

Adjusts the gain depth of modulation for the modulation input into the waveshaper. The value represents the number of 1.5% width increments for each unit of modulation.

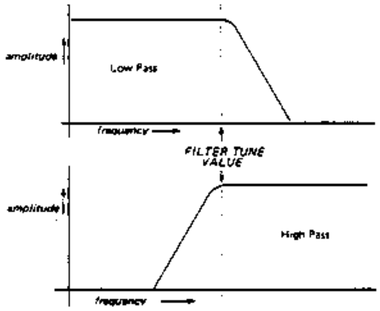
CUTOFF PARAMETERS

LP/HP
37

LOW PASS/HIGH PASS FILTER [37], VALUES = 0 and 1

Selects the mode of the filter.

Value	
0:	Low-pass
1:	High-pass



RESO-
NANCE
38

FILTER RESONANCE [38], VALUES = 0 through 7

Adjusts the resonance, or Q, of the filter. The resonance ranges form 0 (no resonance) through 7 (self-oscillate).

TUNE
39

FILTER TUNE [39], VALUES = 0 through 63

Adjusts the unmodulated tuning of the filter. The range is in whole tone increments, starting at C (16 Hz.).

Note: The filter does not track the keyboard unless it is modulated by the keyboard glide.

MOD 1
SELECT
40

FILTER MODULATION 1 SELECT [40], VALUES = 0 through 15

Selects a control signal for the No. 1 modulation input into the filter.

Value	
0:	Keyboard Glide A
1:	Sweep A
2:	Envelope 1A
3:	Envelope 2A
4:	Keyboard Glide B
5:	Sweep B
6:	Envelope 1B
7:	Envelope 2B
8:	Lever 1
9:	Lever 2
10:	Pedal 1
11:	Pedal 2
12:	Velocity
13:	Threshold velocity
14:	Pressure
15:	Threshold pressure

Note: If the split patch is being used, selections 4–7 default to the A sources, as there are no B sources.

MOD 1
DEPTH
41

FILTER MODULATION 1 DEPTH [41], VALUES = -64 through +63

Adjusts the gain for the No. 1 modulation input into the filter. The value represents the number of semitone increments for each unit of modulation.

Note: When using the glide as a modulation source, a setting of 32 causes the filter to track the pitch perfectly.

MOD 2
SELECT
42

FILTER MODULATION 1 SELECT [42},
VALUES = 0 through 15

(SAME AS PARAMETER [40])

MOD 2
DEPTH
43

FILTER MODULATION 2 DEPTH [43],
VALUES = -64 through +63

(SAME AS PARAMETER [41])

MOD 3
SELECT
44

FILTER MODULATION 3 SELECT [44],
VALUES = 0 through 15

(SAME AS PARAMETER [40])

MOD 3
DEPTH
45

FILTER MODULATION 3 DEPTH {45],
VALUES= -64 through +63

(SAME AS PARAMETER [41])

AMPLITUDE PARAMETERS

MOD 1 SELECT
46

AMPLITUDE MODULATION 1 SELECT [46], VALUES = 0 through 3

Selects an envelope for the No. 1 modulation input into the amplifier.

Value 0: Envelope 1A
1: Envelope 2A
2: Envelope 1B
3: Envelope 2B

Note: If the split patch is being used, selections 2 and 3 default to the A sources, as there are no B sources.

MOD 1 DEPTH
47

AMPLITUDE MODULATION 1 DEPTH [47], VALUES = 0 through 15

Adjusts the gain of the No. 1 modulation input into the amplifier. The range is exponential, in approximate 2dB steps. A setting of 0 shuts off the amplifier.

Note: At least one of the two modulation inputs must be used in order to get anything out of the amplifier.

MOD 2 SELECT
48

AMPLITUDE MODULATION 2 SELECT [48], VALUES = 0 through 3

Selects an envelope for the No. 2 modulation input into the amplifier.

Value 0: Envelope 1A
1: Envelope 2A
2: Envelope 1B
3: Envelope 2B

Note: If the split patch is being used, selections 2 and 3 default to the A sources, as there are no B sources.

MOD 2 DEPTH
49

AMPLITUDE MODULATION 2 DEPTH [49], VALUES = 0 through 15

Adjusts the gain of the No. 2 modulation input into the amplifier. The range is exponential, in approximate 2dB steps. A setting of 0 shuts off the amplifier.

MOD 3 SELECT
50

AMPLITUDE MODULATION 3 SELECT [50], VALUES = 0 through 7

Selects a fixed post-modulation for the amplifier.

Values 0: None. Gain fixed at 1.
1: Pressure. Gain = 1/2 to 1.
2: Keyboard. Gain = 1 at bottom to 1/2 at top.
3: Sweep. Gain = (1/4 x sweep) + 3/4.
4: Pedal 1. Gain = 0 to 1.
5: Inverted Pedal 1. Gain = 1 to 0.
6: Pedal 2. Gain = 0 to 1.
7: Inverted Pedal 2. Gain = 1 to 0.

Note: The first two modulation inputs are added, then multiplied by this modulation input. Settings 1 and 4 through 5 use performance controls. Setting 2 is to compensate for the increase in volume at high pitches that results when the filter tracks the pitch. Setting 3 is for a tremolo effect.

HIDDEN FUNCTIONS

[SET SPLIT] FUNCTIONS

The following functions can be accessed by pressing [SET SPLIT] then a numbered switch [n].

TEMPORARY STEREO SETUPS

[SET SPLIT], [1] LINK PATCH to 0

Set the link patch to 0. This is useful if the link is set up with identical A and B channels (patch 1) and more voices are desired, at the expense of some richness in sound. The patch change is only temporary, and does not affect the setting of the patch parameter in memory.

[SET SPLIT], [2] LINK to 0

Route the link channels to output 0. This temporarily overrides the output select parameter in the link program.

[SET SPLIT], [3] LINK to 1

Route the link channels to output 1.

[SET SPLIT], [4] LINK to 2.

Route the link channels to output 2.

[SET SPLIT], [5] LINK to 3

Route the link channels to output 3.

SUBROUTINES

[SET SPLIT], [15] OUTPUT SUBROUTINE

Output a subroutine packet to the cassette interface. This is used at the factory to prepare software patches to be read in and executed.

[SET SPLIT], [25] EXECUTE SUBROUTINE

Execute a subroutine read in from the cassette. The purpose of this is to allow a small software patch to be loaded from cassette and then executed by the Chroma computer. This function has no effect unless a valid subroutine packet has just been loaded from the cassette.

DIAGNOSTICS

[SET SPLIT], [6] BATTERY

Display the memory battery voltage.

[SET SPLIT], [7] DISABLE BOARD

One channel board is disabled. It is always the lowest priority board belonging to the main instrument, which means that if any notes are latched, the most recent one will be killed, or if any notes are held, the most recent one will be killed, otherwise, the most recently released note will be killed. In general, the way to kill a board is to play one note at a time until the bad one sounds, and then use this function.

[SET SPLIT], [8] DISPLAY DISABLED BOARDS

Display the number of any disabled boards.

[SET SPLIT], [9] TAPPER ON/OFF

Toggle the tapper enable/disable.

[SET SPLIT], [29] MUTE ALL

Mute the oscillators, for diagnostic purposes. The oscillators continue to run, but they are disconnected from the filter inputs.

[SET SPLIT], [30] TEST LEDs

Turn on all LEDs and display segments. This is used for testing at the factory. Pressing any other switch will restore all but the program number display, which is restored whenever a program is selected.

[SET SPLIT], [50] RESET

Reset the Chroma main computer.

PROGRAMMING AIDS

[SET SPLIT], [26] MUTE A

Mute A channel. This is useful when in EDIT B mode, for working on one channel of a pair, without hearing the other. The effect of this is temporary, and can be cleared with [SET SPLIT], [28].

[SET SPLIT], [27] MUTE B

Mute B channel. Similar to [SET SPLIT], [26].

[SET SPLIT], [28] UNMUTE

Unmute channels. This undoes [SET SPLIT], [26], [27] and [29].

CASSETTE FUNCTIONS

[SET SPLIT], [10] CASSETTE MODE

Toggle the cassette mode. The normal mode involves sensing and controlling the cassette motor. The other mode allows use with cassettes that do not have motor control. To tell which mode the cassette interface is in, press one of the four cassette interface switches on the panel with no cassette connected. If nothing happens, the normal mode is in effect, as the Chroma can tell there is no cassette connected. If the panel goes blank, the non-sensing mode is in effect, in which case the Chroma will assume that a cassette is connected and running. To restore the Chroma in this case, press the cassette interface switch again.

[SET SPLIT], [11] CASSETTE PACK NO. 0

Output a program No. 0 packet to the cassette interface.

[SET SPLIT], [12] CASSETTE PACKET NO. 1

Output a program No. 1 packet to the cassette interface.

[SET SPLIT], [13] CASSETTE PACKET NO. N

Output a program number packet including the current program number (as shown in the display) to the cassette interface.

[SET SPLIT], [14] CASSETTE STOP PACKET

Output a stop packet to the cassette interface.

INTERFACE COMMANDS

[SET SPLIT], [16] PERFORMANCE OFF

Turn off the performance switch.

[SET SPLIT], [17] PERFORMANCE ON

Turn on the performance switch, for transmitting to another Chroma.

[SET SPLIT], [18] PANEL OFF

Turn off the panel switch.

[SET SPLIT], [19] PANEL ON

Turn on the panel switch, for transmitting to another Chroma.

[SET SPLIT], [20] SEND PROGRAM

Send a WRPG (Write Program) command, along with the current program, for setting up a remote Chroma.

ENVELOPE PARAMETERS

[SET SPLIT], [21] ATTACK THRESHOLD

Set the attack threshold. There is a general modulation source called threshold velocity which puts out 0 for soft keystrikes or 1 unit for hard keystrikes. The Threshold Velocity is also used by an envelope when its amplitude touch parameter is set to 6 or 7 (see Table of Parameters). The point above which the keystrike is considered a hard strike is called the attack threshold. Using [SET SPLIT], [21] causes the threshold to be set to whatever number is in the parameter display, regardless of what parameter is selected. Setting the threshold close to zero will cause all but the very softest keystrikes to be considered hard keystrikes. Setting the threshold close to 31 will cause all but the very hardest keystrikes to be considered soft keystrikes. Values from 10 to 25 are useful.

[SET SPLIT], [22] RELEASE THRESHOLD

Set the release threshold. When an envelope's release parameter is set to 31, the release becomes touch sensitive. Release velocities below a certain threshold yeild one release time while velocities above the threshold yeild a different release time. The threshold is a number from 0 to 31, and can be set in a manner similar to the attack threshold.

[SET SPLIT], [23] SLOW RELEASE

Set the slow release rate. This is the rate at which an envelope will release for a slow key release when the release parameter is set to 31. It is set to a value between 0 and 31 in a manner similar to the two thresholds above.

[SET SPLIT], [24] FAST RELEASE

Set the fast release rate. This is the rate at which an envelope will release for a fast key release when the release parameter is set to 31. It is set to a value between 0 and 31 as described above.

Sample Program

The following is a step-by-step example of creating a new program. The example given is for a brass chorus sound. You may wish to tailor the program to your own tastes once you have initially set all of the parameter values.

The first step is to clear a program. Press both [EDIT A] and [EDIT B], then press *and hold* the [PARAMETER SELECT] switch. While holding this switch, press each of the numbered switches (1 through 50). When you play the keyboard, a raw, sawtooth sound should be heard.

PRESS	SET PARAM VALUE	EXPLANATION
1 PATCH	9	This value establishes a four-pole filter response, ideal for brass.
39 TUNE	0	Tunes the filters down.
42 MOD 2 SELECT	2	Selects Envelope 1A to modulate the filter.
43 MOD 2 DEPTH	19	Sets the depth of filter modulation by the envelope.
40 MOD 1 SELECT	0	Selects keyboard glide so that the filter can change tuning with pitch.
41 MOD 1 DEPTH	19	Determines how much the filter tuning will increase as you go up the keyboard, in this case, slightly less than the rate of the Pitch (32 = same as Pitch).
14 ATTACK	2	Medium fast attack time, Envelope 1.
16 DECAY	27	Medium slow decay time.
18 RELEASE	7	Medium fast release rate.
13 AMP TOUCH	3	Medium touch sensitivity, Envelope 1.
21 ATTACK	4	Medium fast attack time, Envelope 2.
23 DECAY	30	Medium long decay time, Envelope 2.
25 RELEASE	6	Medium fast release.
44 MOD 3 SELECT	3	Select Envelope 2 to modulate filter.
45 MOD 3 DEPTH	43	Adds Envelope 2 to filter input resulting in an "ADSR" envelope function.
38 RESONANCE	1	Sets a small amount of resonance.
27 MOD 1 SELECT	1	Select Sweep A to modulate pitch (for vibrato).
28 MOD 1 DEPTH	3	Sets a small amount of pitch modulation.
9 RATE	54	Sets vibrato rate.
8 MODE	1	Sets independent vibrato mode.
12 AMPL MOD	13	Produces a delayed vibrato.
4 DETUNE	2	Detunes the A and B oscillators.

Note: all switches not mentioned are assumed to be in the "default" state (usually zero). Press and hold [PARAMETER SELECT] and then press the parameter switch to set it to the default value.

CASSETTE

INTERFACE NOTES

1. During cassette operations, turning off the cassette or pressing any one of the four cassette interface switches on the Chroma will abort the operation. During a LOAD ALL, the program number will show the number of the program being read, if the abort occurred during a program packet, or the number of the program about to be read, if the abort occurred between packets.

2. At the end of every cassette operation, the cassette is turned off by the Chroma. To reenable the cassette's own controls, simply press the stop switch on the cassette. The Chroma detects this and immediately reenables the cassette for rewinding or whatever.

3. The LOAD ALL function makes no assumptions about which programs it is to load. It simply follows the instructions on the cassette. The SAVE ALL puts a program number 1 packet at the beginning and stop packet at the end of the recording, and these tell the LOAD ALL where to start and stop. If an error occurs during a LOAD ALL, and the tape was recorded with gaps between the cassette, the user may press LOAD ALL again to continue with the next packet. However, if the user manipulates the cassette controls, the Chroma will detect this and assume that the user backed up the cassette one packet, so the Chroma backs up the program number too. This allows retrying the offending packet. If the user backs up the tape up to the beginning, the Chroma will at first assume that the tape was backed up one packet only, but the program number 1 packet will, what occurs at the beginning of the sequence will override this assumption.

[LOAD ONE]

Attempt to read one packet from the cassette. If an error occurs, the display will show "Error." If a program packet is read, it will be loaded into the current program (program 0) so that it can be auditioned, and possibly stored someplace. If a stop packet is read, the display will show "OFF," and if a program number packet is read, the display will show "Pr no" followed by the program number. The lock switch has no effect. If a subroutine packet is read, the display will show "Subr."

[SAVE ONE]

Write one program packet to the cassette from current program with four seconds of space before it.

[LOAD ALL]

Begin reading packets from the cassette into the Chroma. A program number packet will set the program number appropriately. A stop packet will terminate the operation. A program packet will cause the program to be written into the memory location specified by the current program number, and then causes the program number to be incremented, with wraparound from 50 to 1. If the lock switch is locked, program packets will be checked for errors, but not placed in memory. If an error occurs, the program number display will show the number of the program about to be read, which is usually one greater than the number of the program that contained the error. If a subroutine packet is encountered, the cassette will stop, leaving the subroutine in the cassette buffer where it can be executed with [SET SPLIT], [25].

[SAVE ALL]

Write a program number 1 packet, 50 program packets containing programs 1 through 50, and a stop packet to the cassette. About four seconds of space will be placed between each packet, unless the lock switch is locked, in which case no spaces will be provided.

GLOSSARY

BOARD – A board consists of a pair of CHANNELS. The system contains eight boards, and each board is independent of the other boards. That is, the sound produced by each board has nothing to do with, and no effect on, the sounds produced by the other boards. Each board has two OSCILLATORs, two WAVESHAPERs, two FILTERs and two VOLUME CONTROLs, all implemented as conventional analog synthesizer circuitry. Each board also includes two GLIDE generators, two SWEEP generators and four ENVELOPE generators, all of which are implemented as computer software processes, not as analog hardware.

CHANNEL – A channel consists of an OSCILLATOR, a WAVESHAPER, a FILTER and a VOLUME CONTROL, supported by a GLIDE generator, a SWEEP generator and two ENVELOPE generators. Often, channels are paired for greater tonal complexity. See BOARD.

CURRENT PROGRAM – In addition to the fifty stored PROGRAMs in the Chroma's memory, there is the current program, sometimes called Program 0. This program always controls the sound of the synthesizer (or of the main INSTRUMENT when LINKING). In addition, the programming controls function on the current program, not on the fifty stored programs. This means that the current program is a form of workspace in which all programming is done. If the programmer wishes to keep a sound he has created for future use, he must use the Store switch to store the current program into one of the fifty positions in the Choma's memory. In order to listen to one of the stored programs, it is necessary to select the program, which actually causes the stored program to be copied into the current program space. Even though the current program is not considered a "stored" program, it does reside in the Chroma's battery back-up memory. This means that the instrument can be turned off today and then turned on tomorrow and the current program will be remembered, along with the fifty stored programs.

EDIT MODE – Since the CHANNELs in the Chroma are often used in pairs, two complete sets of parameters are required to completely define the sound of an INSTRUMENT. These are the A PARAMENTERs (which are used even when the channels are not paired) and the B PARAMETERs (which are only used when the channels are paired). In order to access these two sets of parameters (without having another fifty switches on the panel) the two edit mode switches are provided. In Edit A mode, the display and parameter control slider are connected to the A parameter. In Edit B mode, the display and slider are connected to the B parameter. In Edit A & B mode (entered by pressing both edit switches at the same time), the display will show the A parameter's value, yet as soon as the slider is moved both the A and B parameter will be set to the value determined by the slider position. Since the edit mode and (parameter) number is itself a PANEL PARAMETER in the CURRENT PROGRAM, the edit mode and number can be stored so that the slider is automatically connected to a particular parameter whenever a particular program is selected.

ENVELOPE – Any aspect of a musical tone, such as its pitch or its volume, can be described in quantitative terms. One can say that a tone's pitch is middle C and its volume is -15dB, or whatever. However, if an aspect of the sound varies with time, it's not that simple. Fortunately, the time-variations of an aspect of a tone can often be reduced to a simple pattern or combination of patterns. An envelope is one of these types of patterns. In many synthesizers, envelopes are of the type called ADSR, for Attack Decay Sustain Release, which are the four PARAMETERs that describe the shape of the envelope. In the Chroma, this has been simplified to ADR, meaning that each envelope has an ATTACK time, a DECAY time and a RELEASE time. In addition, each envelope can be made larger or smaller in magnitude in response to how hard the key that triggers it is struck. When a key is pressed, the computer inside the Chroma decides which CHANNEL(s) will be used for the note, and then sets the envelopes for those channels to their attack phase. The envelope generators will then proceed to generate their programmed shape, consisting of a rising attack and a falling decay, until the key is released, at which time the envelope generator will be forced to generate a falling release, which is similar to decay, yet usually faster. The shape of the envelope by itself does not produce any sound, yet it can be used to dynamically modulate aspects of the sound such as volume or filter cutoff.

GLIDE – Most synthesizers have provisions for causing the pitch of an oscillator to slide from one note to the next, and in most synthesizers, this function is called PORTAMENTO. The Chroma, however, uses what is called a glide generator, which is capable of generating PORTAMENTOs (smooth slides) or GLISSANDOs (chromatic runs).

GLISSANDO – The GLIDE generators in the Chroma are capable of causing the transitions from one note to the next to be done by a series of chromatic steps. This is called Glissando. A wide range of glissando rates is available.

INSTRUMENT – Although the Chroma is, of course, a musical instrument, the term "instrument" is used here to refer to that group of BOARDs and PARAMETERs that causes the Chroma to create a particular sound. The Chroma is capable of creating two different sounds at once by LINKING, or eight at once using the computer interface. Therefore, from the programmer's point of view, the Chroma can be thought of as containing several "instruments," and though they do not have discrete physical identities (they are all inside the same box), they have distinct identities as data structures.

KEYBOARD SPLIT – When using the LINKING feature, it is possible to make the Chroma function as two INSTRUMENTs at the same time. If link lower or link upper mode is selected, the keyboard split will determine which keys control which instrument. Since the keyboard split is a PANEL PARAMETER, it need not be setup every time a link is setup. Rather, it can be stored along with the link information and all the other parameters, and automatically setup whenever the appropriate program is selected. However, it can be altered quickly using the Set Split switch. See the SWITCH APPENDIX for details.

LINKING – Normally, the Chroma functions as a single INSTRUMENT, whose sound is defined by the CURRENT PROGRAM. To enhance the versatility of the instrument, there is a special PANEL PARAMETER that is accessible from the control panel that allows the Chroma to function as two instruments. In this mode, the main instrument is still controlled by the current program, while the link instrument is controlled by one of the fifty stored programs. The stored program that controls the sound of the link instrument cannot be directly accessed by the programmer without selecting that program (making it the current program). The link feature has four modes, no link, link lower, link unison and link upper. In no link mode, the linking feature is not used. In link lower mode, keys below the KEYBOARD SPLIT cause the link instrument to sound and keys above or equal to the split cause the main instrument to sound. In link unison mode, each key will cause both instruments to sound, which of course reduces the apparent number of VOICEs available, yet produces extremely rich sounds. In link upper mode, keys above or equal to the keyboard split cause the link instrument to sound while keys below the split cause the main instrument to sound. The linking can be quickly setup from the panel (see CHROMA SWITCH DESCRIPTIONS). In addition, since the link mode and number is in fact a PARAMETER within the CURRENT PROGRAM, the link can be stored so that it will automatically be setup whenever the program containing the link is selected.

MODIFIED FLAG – Since the CURRENT PROGRAM is only a copy of one of the stored programs, and since the current program can be modified, independently of the stored program, by the programming controls, an indication of whether or not the current program equals the stored program is shown at all times. The units decimal point in the large two-digit display is used for this purpose. Whenever a program is selected or stored, the program number display will be set to the appropriate number and the modified flag will be turned off. Whenever the current program is changed, the modified flag will be turned on, telling the user that what he is hearing (as controlled by the current program) is not the same as what is in the Chroma's memory (in the stored program). Note that, since the parameter number, edit mode, link, etc., are all parameters, changing any of these will set the modified flag.

PANEL MODE – The Chroma's control panel can operate in four basic modes. The mode determines the function performed when one of the fifty numbered switches on the right side of the panel is depressed. The mode used most often during performance is Program Select mode. In this mode, depressing one of the fifty switches on the right panel will copy an entire PROGRAM into the CURRENT PROGRAM, and also setup the PANEL PARAMETERs and establish any LINKING specified by the program. The mode used most often during programming is Parameter Select mode. In this mode, depressing one of the fifty switches selects one of the CONTROL PARAMETERs or, depending on the EDIT MODE, either one of the A PARAMETERs or one of the B PARAMETERs. When a parameter is selected, its number and value are shown in the eight-digit display. In addition, moving the PARAMETER CONTROL SLIDER causes the value of the parameter to change to whatever the programmer wishes. The remaining two modes are called Copy From A and Copy From B. In these modes, depressing switches on the right panel causes the appropriate parameter to be selected, and copies its value from a program whose number was selected when the Copy mode was first entered. The panel mode (including the program number being copied from the Copy modes) is stored in the battery-backup memory so that the instrument will always power up in the same state it was when it was last shut-off.

PANEL PARAMETER – There are six panel parameters in each program. They do not directly affect the sound created by the INSTRUMENTs. Rather, they are extra parameters that automatically initialize certain things that are accessible from the control panel. The panel parameters include the Link Balance, the Link Mode & Number, the Edit Mode & Number, the Main Transpose, the Link Transpose and the Keyboard Split. Although these panel parameters do end up affecting the sound, they do so by other means than by altering the INSTRUMENT that is creating the sound. For instance, the Main Transpose transposes the notes that are played on the keyboard before they are given to the main instrument process inside the computer. The part of the computer that generates the sound does not know (and does not care) whether you play middle C with no transpose or low C while transposed up 1 octave.

PARAMETER – A parameter is a single numerical quantitiy that controls one specific aspect of the operation of the Chroma, usually an aspect of the sound that is being created. Each parameter has a number that identifies what it controls, and a value that specifies the setting for the particular control. For instance, filter tuning is parameter number 39, and it has a value that can range from 0 to 63, representing tunings that span the entire audio spectrum in whole-tone increments. This particular parameter represents the tuning of the filters that it controls before any modulation is added to the tuning. For a complete list of parameters and what they represent, consult the section entitled TABLE OF PARAMETERS. The parameters fall into four categories, however, called the PANEL PARAMETERs, the CONTROL PARAMETERs, the A PARAMETERs and the B PARAMETERs, all defined elsewhere in this glossary.

PATTERN – The SWEEP is capable of generating typical LFO waveshapes such as sine, square, triangle, etc. In addition, it is capable of generating stepped patterns. These patterns consist of a short sequence of values that repeats at a regular rate. In fact, the square wave that the sweep generates is actually a two-state pattern. There is also a random "pattern" that sounds like a conventional synthesizer with noise feeding a sample & hold.

PORTAMENTO – The GLIDE generator is capable of generating a smooth sliding transition from one note to the next. This is called Portamento. A wide range of portamento rates is available.

PROGRAM – A program is a set of PARAMETERS used to describe the sound that will be created by the Chroma (or, more precisely, by an INSTRUMENT within the Chroma). There are 101 parameters in each program, most of which directly control the sound. The Chroma's memory is large enough to contain 50 stored programs, plus one "current program," which is the program that is currently controlling the synthesizer and is accessible to the programmer. Programs can be moved around the Chroma's memory by SELECTING or by STORING. A secondary program can be called upon, in order to get two sounds at a time, by LINKING programs.

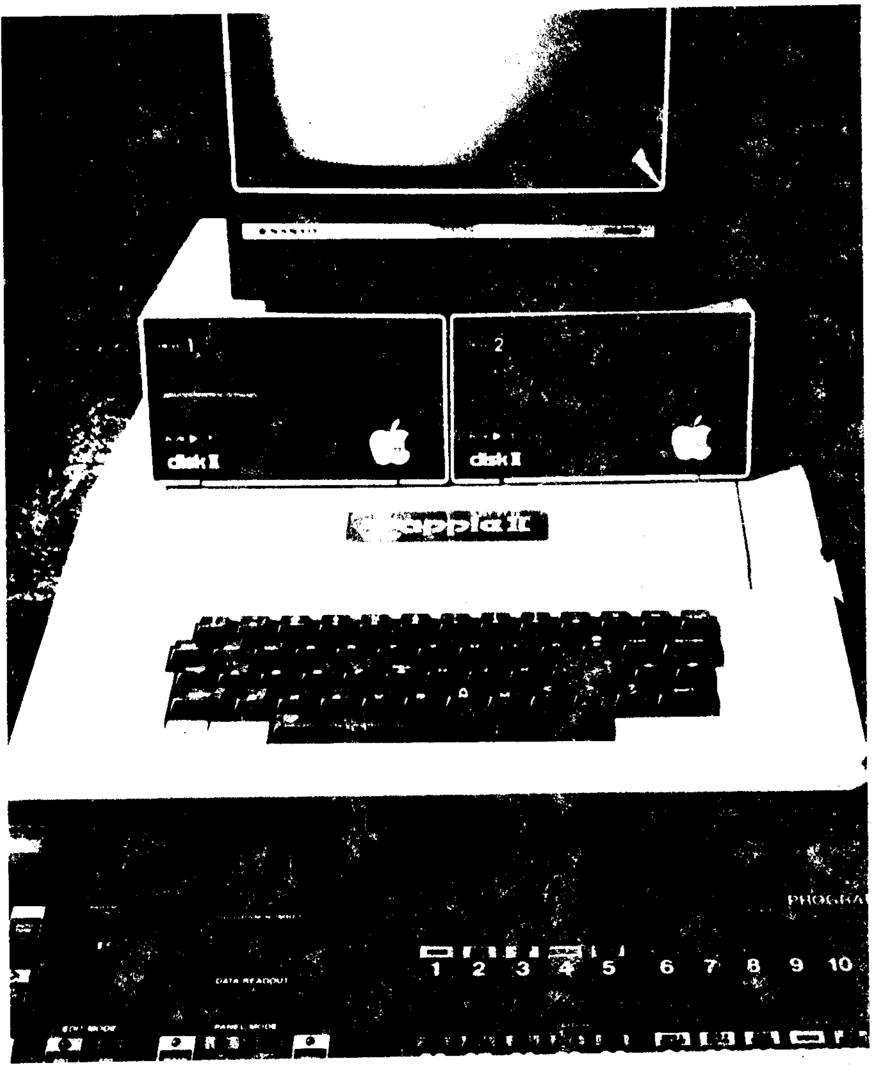
SWEEP – One of the dynamically varying quantities that can control an aspect of a sound is called the sweep. In other synthesizers, it is often called the LFO, for Low Frequency Oscillator. However, in the Chroma there are not any oscillators for this function. Rather, this control function is generated by the Chroma's computer. So, it is given the term SWEEP instead of LFO. The sweep has several parameters that control its operation, including its rate and its waveshape. Like the ENVELOPEs, the sweep produces no sound of its own, yet can be used to dynamically vary an aspect of the sound being created, such as the volume or pitch.

TEMP MODE – Short for Temporary Panel Mode. In addition to the four PANEL MODEs which determine the function of the right panel switches, there are eight temporary modes, indicated by a blinking LED on the panel, that determine the function to be performed by the next depression of one of the right panel switches. As soon as a right panel switch is pressed, its temporary function will be performed and the temp mode will be cleared, returning the Chroma to its previously set panel mode. In some cases, use of a temp mode will result in the panel mode being changed. The specifics of these modes can be found under the section entitled SWITCH APPENDIX. The temp mode, like the PANEL MODE, is stored in the Chroma's battery-backup memory, so that the Chroma will always power up in the same state it was when it was last shut-off.

VOICE – The term "voice" has several meanings in the synthesizer industry. However, the term has very specific meaning as far as the Chroma is concerned. A voice is the total sound you hear as a result of pressing one key. The number of voices a synthesizer has is determined by how many tone generators it has (16 in the Chroma) as well as how many tone generators are required for each note (anywhere from 1 to 16 in the Chroma, typically 2). This means that the Chroma can function as a synthesizer that has anywhere from 16 voices down to 1 voice, usually 8, depending how it's programmed. The term "voice" does NOT refer to how many different sounds the instrument can remember. That is the number of PROGRAMs that the synthesizer can store in its memory.

Rhodes

CHROMA
Interface Manual



CHROMA COMPUTER INTERFACE MANUAL

Table of Contents

All specifications subject
to change without notice.

INTRODUCTION

This section should serve as an overview of the Chroma computer interface and of the topics that will be covered in detail in the subsequent sections of this manual.

Purpose of the interface -- The Chroma computer interface is to allow the Chroma to be controlled by a computer. A Chroma can be controlled by another Chroma, but this is not the main intent of the interface.

Capabilities of the interface -- The interface allows a number of different functions to be performed.

A computer can "play" the Chroma's keyboard and performance controls.

A computer can record what a human plays on the keyboard and performance controls.

A computer can change sounds, or modify parameters in existing sounds.

A computer can record the changes made to a sound by a human manipulating the panel controls of the Chroma.

A computer can load or save packets of information using the cassette interface.

A computer can temporarily alter the workings of the Chroma by changing the firmware of the Chroma's internal computer. This facility is not likely to be very useful to anyone outside Rhodes, but may be used in future products designed to enhance the Chroma.

Physical nature of the interface -- The computer interface consists of two identical 8-bit parallel ports, one for each direction. Each port consists of 8 latched data bits, a status line that tells whether there is information on the port, and an acknowledge line which is pulsed whenever a data byte is read from the port. All signals are TTL compatible, which means that interface hardware is simple and cheap. However, this also means that the interface is not designed to work over great distances or in the presence of ground differences or large amounts of noise. Each device must have a set/reset flip-flop driving the status line for its output port. This flip-flop must be set when writing a byte to the port and cleared by the acknowledge pulse received from the other end. The acknowledge pulse will normally be the read pulse used to read the data from the port.

Interface protocol -- Communication in each direction is independent of communication in the other direction. The interface is best handled in an interrupt driven environment with a queue (first-in first-out list) for buffering the information in each direction (or at least in the input direction). There are two levels to the protocol, the physical and the logical. The physical level is kept simple by the fact that each byte transferred is acknowleged by the receiving end before another byte may be transferred. The logical level can be more complex because a communication that requests a response from the other end doesn't necessarily receive the response immediately. In fact, several different requests may be transmitted before the responses start coming back.

Command language -- All communication issued by a computer to control a Chroma is in the form of commands. A command is a sequence of one or more bytes, where the first byte is the command code (similar to a machine language op-code) and the remaining bytes are operands. For instance, the command to set parameter 5 in program 9 to a value of 3 consists of the sequence 7, 9, 5 and 3. (7 is the command code for Write Parameter.) Certain commands require a response from the Chroma. In this case, the Chroma will respond by sending back what looks like a command that begins with the same command code. It isn't really a command, as the Chroma doesn't control the computer, but it takes a similar form. There are also commands to the Chroma that enable recording from the Chroma. When these commands are issued, the Chroma enters a mode in which it will send "commands" to the computer whenever a note is played or a control is moved. In a simple recording system, these commands may be played back verbatim at a later time. Thus, even though the Chroma doesn't actually tell the computer what to do, the communication in both directions is organized as commands.

The interface view of the Chroma -- The performer sitting at the Chroma "sees" a system that is capable of generating two sounds at a time using the Link capability. The computer connected to the Chroma "sees" a system that is capable of generating up to eight different sounds at a time using the "multiple instrument" capability. The reason that the performer isn't given this capability from the keyboard is that there is no easy and reliable way for a performer to "tell" the Chroma which notes go with which sounds, beyond a simple keyboard split. The computer interface doesn't have this limitation, as it can instruct the Chroma which sound to use with each note. The interface is designed to allow the performer to play one "track" at a time (or two, using a link) into a computer, and then the computer to play multiple "tracks" back as if there were multiple instruments being played.

The interface view of the Expander -- A Chroma Expander looks just like a Chroma through the computer interface, except that it is not capable of generating keyboard information. It should be used for playback only.

CHROMA STRUCTURE

The intent of this section is to describe the way the Chroma appears through the computer interface, and to relate the way a computer deals with the Chroma to the way a human performer or programmer deals with the Chroma. A computer has a much greater range of control over the instrument than a human, as it is capable of communicating with it much faster than a human could ever punch buttons. What a human sees of the Chroma's inner workings is a subset of what the computer sees, and it is important to understand both in order to fully exploit the capabilities of the interface.

The Human's View — A person sitting at the Chroma and manipulating the controls on the Chroma's panel has a certain view of the inner workings of the Chroma. This view has the following characteristics:

> There are fifty programs in the Chroma's memory that can be called upon at any time, numbered 1 through 50.
>
> There is a fifty-first program (program 0) that controls the sound of the instrument.
>
> Selecting a program causes one of the fifty stored programs to be copied into program 0, and storing a program causes program 0 to be copied into one of the fifty storage locations.
>
> All creating and editing of programs is done to program 0, and then stored someplace else.
>
> A second sound, called a link, can be added to the main sound. The link sound is controlled by the link program, which is one of the fifty stored programs. A parameter in program 0 "points to" the link program.
>
> Certain other parameters in program 0 affect the link. These include the Keyboard Split, Link Mode (lower, unison or upper), Transposes and the Link Balance.
>
> Since the link is one of the fifty stored sounds, it cannot be directly edited.
>
> Changing a parameter from the panel causes program 0 to be modified, and the sound reflects this change. The original program from which program 0 was copied is not affected.
>
> The performance controls affect both the main and link sounds. The notes played on the keyboard are assigned to the main and/or link sounds according to the link mode and keyboard split.

The Computer's View -- A computer "looking into" the computer interface port on the Chroma has a much more detailed view of the inner workings of the Chroma than a person sitting at the controls. The most important concept that must be understood when working with the computer interface is the concept of an "instrument". In a physical sense, the Chroma itself is obviously an instrument. Here, though, we are concerned with a more symbolic interpretation of the word. In this sense, the Chroma is capable of containing more than one "instrument" as it is capable of generating more than one sound at a time. To understand this, study the following definitions:

> Channel -- A channel is the basic unit of sound generation, consisting of an oscillator, wave shaper, filter, amplifier, glide, sweep and two envelopes (if patch 0 is selected), or consisting of two oscillators, wave shapers, filters, amplifiers, glides and sweeps, and four envelopes (if patch 1-15 is selected).
>
> Board -- A board is the physical entity containing the synthesizer circuitry. A board contains two channels if patch 0 is selected, or one channel if patch 1-15 is selected. If patch 0 is selected, the two halves of the board will always have the same sound characteristics. The Chroma has eight boards.
>
> Program -- A program is a set of parameters that completely defines a sound, plus a few parameters that represent settings of certain panel controls (such as the edit mode and link information). The battery backup memory in the Chroma contains 51 programs.
>
> Instrument -- An instrument is a group of boards that is treated as a logical entity, and whose sound is defined by a particular program. There can be as many as eight instruments defined in the Chroma at any one time, numbered 0 through 7. Each instrument has its own set of performance control and keyboard "inputs" that function independently of the inputs to other instruments.

The notion of an instrument makes sense if you consider the Chroma as a small ensemble. Through the computer interface one might play a piece of music involving a guitar, a bass, and two trumpets. This would be done by defining four instruments by sending the appropriate Define commands to the Chroma, and then sending commands to attack and release notes to the individual instruments within the Chroma. The two trumpet instruments would presumably be defined by the same program, emphasizing the fact that a program and an instrument are two different animals.

Translating The Human's View To The Computer's View -- When controlling the Chroma from its control panel and keyboard, two sounds can be played at a time. Internally this means that, you guessed it, two instruments can be defined. The various actions that can be performed from the panel translate into internal manipulations of programs and instruments as follows:

Program Select Without Link -- The selected program is copied into program 0. Then instrument 0 is defined according to program 0 and instrument 1 is undefined.

Program Select With Link -- The selected program is copied into program 0. Then instrument 0 is defined according to program 0 and instrument 1 is defined according to the link program.

Link -- Instrument 0 is left unchanged and instrument 1 is defined according to the specified link program. The Link Mode and Link Program parameters in program 0 are set accordingly.

Unlink -- Pressing [NO LINK] twice causes the link to be cleared by undefining instrument 1. The Link Mode parameter in program 0 is set to "No Link".

"No Link" Program Select -- Pressing [NO LINK] followed by a numbered switch causes the selected program to be copied into program 0, except that the link related parameters are not copied. Instrument 0 is then defined according to program 0 and instrument 1 is left alone.

Keyboard Information -- When a key is pressed and no link is in effect, the key number is transposed according to the Main Transpose parameter in program 0 and sent to instrument 0. If a link is in effect, the key number is sent to instrument 0 and/or 1, depending upon the Link Mode and Keyboard Split parameters in program 0. Information sent to instrument 1 is transposed according to the Link Transpose parameter in program 0. All key press information that is sent to either instrument is recorded, along with its transposition, in a key list. When a key is released, it is looked up in the key list and the transposition recorded there is sent to the appropriate instrument(s). Thus, the Chroma won't be confused by changing transpositions or keyboard splits while keys are held down.

Pressure Sensor Information -- If the Pressure Sensor option is installed, varying the pressure on any key causes the appropriate information to be sent to instrument 0 and/or instrument 1, depending upon the key's entry in the key list. That is, if the note's attack was sent to an instrument, subsequent pressure information will also be sent there.

Performance Control Information -- When a lever, pedal or footswitch is moved, the appropriate information is sent to instrument 0 and, if a link is in effect, to instrument 1.

Parameter Changes -- Changing any of the numbered parameters causes the appropriate values in program 0 to be altered. Whenever a program is altered in any way, any instruments defined by that program are automatically affected. Changing any of the panel parameters does not directly affect any instruments, although it may, as in the case of the Link Mode parameter, indirectly affect either instrument 0 or 1.

Thus, the panel controls manipulate instruments 0 and 1, using instrument 0 for the main sound and instrument 1 for the link sound. Instruments 2 through 7 are not affected, and are always undefined when the instrument is first turned on.

Board Allocation and Channel Assignment -- Board allocation is the process of deciding which boards are assigned to which instruments. This process is repeated any time an instrument is defined or undefined. Channel assignment is the process of deciding which channels are assigned to which notes. This process occurs dynamically as notes are played and released. These two processes are separate, yet they do affect each other:

> The channel assignment is controlled by the Keyboard Algorithm parameter, which determines whether the instrument is polyphonic (using multiple channels) or monophonic (using a single channel). This choice affects board allocation in that a monophonic instrument is never assigned more than one board.
>
> The channel assignment is independent for each instrument and must use only those channels assigned to that instrument by the board allocation process. Defining or undefining an instrument (or changing between a mono and a poly keyboard algorithm) affects the number of boards available to other instruments and, as such, impacts the channel assignment of other instruments. Undefining an instrument genernally makes channels available to other instruments, while defining an instrument usually robs channels from other instruments.

The board allocation process operates by first calculating how many boards should be allocated to each instrument and then boards are taken from instruments that have too many and given to instruments that have too few. Unaffected boards can continue to generate sound during this process. In addition, the board robbing is intelligent enough to favor boards that are not currently sounding. Calculating the number of boards each instrument should have is done in a round-robin manner, like dealing cards. The rules are:

> The number of boards in the system is usually eight, but may be smaller if the autotune detected a bad board.
>
> An undefined instrument doesn't get any boards.
>
> An instrument whose Keyboard Algorithm parameter is 5 or more is considered monophonic, and is assigned a maximum of one board. (If patch 0 is selected, only one of the two channels will be used.)
>
> An instrument whose Keyboard Algorithm parameter is 4 or less is considered polyphonic, and may be assigned any number of boards. (If patch 0 is selected, there are twice as many channels available as there are boards.)
>
> The dealing of boards to instruments continues until all boards are used up, or until all instruments have been checked and none of them are polyphonic. In the latter case, there may be boards left over.

Temporary Parameter Changes -- When a performer selects a program and then alters a parameter, this change is recorded in program 0 but not in the original stored program. Although program 0 is in fact stored in the same battery-backup memory as the other fifty programs, it is, by conventional usage, a temporary program. During performance, one may modify a parameter in a program, but this change is usually not stored. Therefore, every time the same program is selected, the changes made last time it was selected are usually not still there.

The intent of the computer interface is to allow several "performances" to be recorded separately and then played back simultaneously. Since there is only one "program 0", another means is provided for making temporary changes to the several sounds that may be selected at once. Whenever an instrument is defined according to a particular program, a translation occurs. The information that makes up a program is very different from the information recorded for each instrument. A program consists of 59 bytes in which all the parameters are packed as tightly as possible to conserve memory. An instrument is represented by a couple hundred bytes of information in which the parameters are expanded into a form that allows for fast processing. There are separate commands for changing parameters in a program and changing parameters in an instrument. The Write Parameter command is used to alter the value of any parameter in any program, and its effect is permanent. If any instrument happens to be defined by that program, it will be affected too. The Set Parameter command, however, only alters the translation of the parameter in the instrument and doesn't affect the program that defines the instrument. If the Set Parameter command is used to alter parameters, these changes will be temporary and will not show up again the next time an instrument is defined by the same program. Note also that, while the Write Parameter command has a complementary Read Parameter command, the Set Parameter command has no complement.

Instrument Definition Parameters -- Whenever an instrument is defined through the computer interface, the performance control inputs must be initialized. To this end, the Define command includes operands that represent the positions of the levers, pedals and footswitches. There is also a volume operand that acts as a master volume control. The volume is also controllable separately through the use of the Volume command. An instrument's volume is only accessible from the panel through the use of the link balance control.

When recording from the Chroma, the computer will receive Define, Undefine and Volume commands from the Chroma whenever programs are selected, linked or reunlinked. The performance control operands in the Define commands will reflect the true physical position of the performance controls at that time. The Link Balance and Link Mode parameter determines the volumes of the instruments as follows:

> If no link is in effect, instrument 0 will be defined with a volume of 255 (maximum).
>
> If a link is in effect, instruments 0 and 1 will be defined with volumes determined by the Link Balance parameter.

When a link is cleared, in addition to sending an Undefine command for instrument 1, the Chroma will send a Volume command for instrument 0 reflecting the fact that its volume is now 255.

When a link is set up, in addition to sending a Define command for instrument 1, the Chroma will send a Volume command for instrument 0 reflecting the setting of the Link Balance.

When the Link Balance parameter is changed, volume commands will be sent by the Chroma for both instruments 0 and 1 if a link is in effect. If no link exists, no volume commands will be sent.

It should be clear by now that most of the communication with the Chroma is actually communication with individual instruments within the Chroma. The command set, listed in the next section, shows this. Some commands (like Read and Write Parameter, mentioned earlier) aren't associated with an instrument, but most of the commands involved in recording and playing back music are addressed to individual instruments within the Chroma.

COMMAND DESCRIPTIONS

The computer that is connected to the Chroma via the interface cable communicates with the Chroma by sending and receiving commands. A command consists of:

A byte that specifies the command. If the command applies to one of the eight "instruments" within the Chroma, the instrument number will be encoded in this byte, too.

Zero or more bytes that specify parameters of the command. Although most commands require specific numbers of parameters, a few commands are variable in length.

Certain conventions are adhered to in the command language:

Undefined commands are considered to be No Operation commands; that is, undefined commands are ignored. All No Operation command have no parameters.

Command code zero and command code FF (hex) will always be No Operation commands, even for future instruments that utilize this interface.

Command code 1 will allways be an Identification command, for this and any other instrument utilizing this interface.

If a two-byte quantity (such as a memory address) is to be transferred, it will be sent most significant byte first, just the way you would write it on paper.

If a command is variable in length, the second byte of the command will specify the variable number of data bytes. This is not the same as the length of the command, as the count does not include the command code, the length byte, or any other fixed parameters for the command. The Peek command is a good example of this.

If a command is variable in length, the second byte of the command will specify the length as follows: values 1 to 255 represent byte counts of 1 to 255, and a value of zero represents a byte count of 256.

Any command that could conceivably "crash" the Chroma through misuse will not be allowed until a special "unlock" command is first issued. This minimizes the chance of a crash if the Chroma should receive garbage from a malfunctioning computer.

The commands fall roughly into three categories, according to protocol:

> There are those commands that are issued by the controlling device and processed by the Chroma with no response.
>
> There are those commands that are issued by the controlling device and require a specific response from the Chroma. The response will always be a "command" starting with the same code that was received from the controlling device.
>
> There are those commands that establish modes within the Chroma that allow the Chroma to subsequently transmit unsolicited "commands" when certain events occur. The unsolicited commands will generally look like commands from the first group above.

The command set can also be split into two categories, according to destination:

> There are those commands that are addressed to the Chroma as a whole. The lower command codes are assigned to these commands.
>
> There are those commands that are addressed to individual instruments within the Chroma. The higher command codes are assigned to these commands. The three least significant bits of these command codes hold the instrument number.

What follows is a complete description of each command, along with the numerical code (in hexadecimal) for each command byte.

No Operation 00

The only significance of this particular No Operation (as opposed to any of the undefined command codes) is that the Chroma sends this code upon power-up or reset.

Identification 01

The Chroma (or any future instrument) will respond with three bytes, an Identification command, a device code (1 for a Chroma, 2 for a Chroma Expander), and a software revision level code (see Appendix A).

Read Program 02 pp

The Chroma will respond by transmitting program number pp. The information is transmitted as a Read Program command and 59 data bytes. (If pp is not between 0 and 50, the data bytes will be undefined.)

Write Program 03 pp dd ... dd

The 59 data bytes dd ... dd are written into program number pp in the Chroma. (If pp is not between 0 and 50, the data will be accepted and ignored.)

Load Packet 04

One packet of information is read from the cassette interface, its error detection codes are checked, and the result will be returned via the interface in the form:

04 nn dd ... dd

nn specifies the number of data bytes in the packet, and the dd bytes are the contents of the packet. The first byte of the packet (the first dd byte) is always the packet ID, which identifies the type of packet. The packet ID for valid data is always non-zero. If an error occurs in the reading of the cassette, a special error packet with an ID of 0 is returned.

This command starts reading from the cassette immediately. This can cause a problem if the cassette was previously idle. See the Tape Space command below.

The types of packets that are currently defined, and the forms the Chroma return them in, include:

<u>Error Packet</u> 04 02 00 nn

The length is 2, the ID is 0, and nn will be 0 if a read error is detected or FF hex if the cassette was not running (or was shut off in mid-operation).

<u>Program Packet</u> 04 3C 01 dd ... dd

The length is 60 (3C hex), the ID is 1, and the 59 bytes of data represent a Chroma program.

<u>Program Number Packet</u> 04 02 02 nn

The length is 2, the ID is 2, and the single byte of data consists of a valid program number (0 to 50). This type of packet appears, with a program number of 1, at the beginning of a tape recorded with SAVE ALL.

<u>Stop Packet</u> 04 01 03

The length is 1, the ID is 3, and there is no data in the packet. This type of packet appears at the end of a tape recorded with SAVE ALL.

Save Packet 05 nn dd ... dd

The packet dd ... dd containing nn bytes is written to the cassette. The first dd byte, which is the packet ID must be non-zero. The Chroma will respond when the operation is complete with 05 00 if the operation completes normally or 05 FF if the cassette isn't running.

Read Parameter 06 pp nn

Parameter number nn in program number pp is read and returned in the form 06 vv, where vv is the parameter value. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be undefined.

Write Parameter 07 pp nn vv

Parameter number nn in program number pp is set to value vv. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be ignored. If the vv value is not within the range defined for the parameter, the result is undefined, except that the parameter will never be set to an illegal value.

Panel Switch Off 08

The "panel switch" referred to is the software switch which "connects" the Chroma panel to the interface. When the Chroma receives this, it will echo it and disconnect the panel from the interface.

Panel Switch On 09

When the Chroma receives this, it will echo it and connect the panel to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

> Whenever a program is selected, a Define command will be transmitted for instrument 0 and either a Define or an Undefine command will be transmitted for instrument 1, depending upon the existence of a link.
>
> Whenever a parameter is changed, a Set Parameter command will be transmitted for instrument 0.
>
> Whenever the link balance is varied, Volume commands will be transmitted for instruments 0 and 1.

Performance Switch Off 0A

The "performance switch" referred to is the software switch that "connects" the various performance controls to the interface. When the Chroma receives this command, it echoes it and disconnects the performance controls from the interface.

Performance Switch On 0B

When the Chroma receives this, it will echo it and connect the performance controls to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

> Whenever a key is pressed on the keyboard, an Attack command will be transmitted for instrument 0, 1 or both, depending upon the link mode and keyboard split.
>
> Whenever a key is released on the keyboard, a Release command will be transmitted for instrument 0, 1 or both, depending upon the link mode and whether or not an attack had already been sent for the note.
>
> Whenever a lever, pedal or footswitch moves, the appropriate command is transmitted for instrument 0, and for instrument 1 if a link is in effect.

Peek 0C aa aa nn

The Chroma responds by transmitting nn bytes from its internal memory starting at location aaaa. The response is in the form:

0C nn dd ... dd

where the dd bytes are data bytes from ascending addresses.

Peek Two Bytes 0D aa aa

The Chroma responds by transmitting two bytes from its internal memory at locations aaaa and aaaa+1. The response is in the form:

0D dd dd

This command is guaranteed to extract the two bytes concurrently, with no chance that the memory locations could be altered between the transmittal of each byte.

Poke 0E aa aa nn dd ... dd

The nn data bytes dd ... dd are poked into the computer's address space starting at location aaaa. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored.

Poke Two Bytes 0F aa aa dd dd

The two data bytes dd dd are poked into the computer's address space in locations aaaa and aaaa+1, respectively. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored. This command is guaranteed to poke the two bytes concurrently, without danger of the computer utilizing half of the old contents and half of the new contents.

Tap Panel 10

The panel tapper is triggered, unless it has been disabled.

Unlock 11 00 FF

This sequence must be transmitted in order to enable the Poke and Poke Two Bytes commands.

Tape Space 12

The cassette motor will be run for two seconds. Upon completion, the Chroma will respond with 12 00 if the cassette was running, or 12 FF if it was shut off.

The purpose of this command is to allow startup time before other cassette operations. If a sequence of Save Packet commands are to be issued, they should be preceded by two Tape Space commands. In addition, if the packets are to be individually readable, they should be separated by two Tape Space commands. A single Tape Space command should be issued prior to a sequence of Load Packet commands.

Restore 13

The Chroma is restored to the state reflected by its panel settings. All instruments are undefined except instrument 0 and possibly 1, which are set up according to the currently selected program. The panel switch, performance switch and pressure switch are turned off, and a Panel Switch Off, Performance Switch Off and Pressure Switch Off command are echoed, in that order. (See Appendix A for information on early revisions.)

Pressure Switch Off 14

The "pressure switch" referred to is the software switch that "connects" the keyboard pressure sensors to the interface. When the Chroma receives this command, it echoes it and disconnects the pressure sensors from the interface. This command is not implemented in early Chromas. See Appendix A.

Pressure Switch On 15

When the Chroma receives this, it will echo it and connect the pressure sensors to the interface. While this mode is in effect, the Chroma will send Pressure commands for instrument 0 and/or 1 whenever the pressure on a key is varied. See Appendix A.

Pressure 68+i kk pp

Instrument i is told to set the key pressure input for note kk to value pp. The pressure is an unsigned number from 0 to 63.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the pressure switch is on and the measured pressure on a depressed key changes. Pressure commands only occur between the corresponding Attack and Release commands for the same note.

This command is not implemented in early Chromas, and <u>must not be sent to them</u>. Current Chromas will respond correctly to this command even if the Pressure Sensor option is not installed. See Appendix A.

Information 70+i

The Chroma responds by echoing the command and sending four information bytes. Currently, only the first byte is utilized, and contains the number of channel boards assigned to instrument i. The other three bytes are zero.

Volume 78+i vv

The Chroma sets the volume of instrument i to vv. The value vv is a linear control from 0 to 255, and is nominally 255. Thus, to reduce the volume of an instrument 6db, the correct vv value would be 128.

This command will be transmitted (for instruments 0 and 1) by the Chroma if the panel switch is on and the Link Balance parameter is varied.

Lever 1 80+i vv
Lever 2 88+i vv

The Chroma sets the value of the appropriate lever input on instrument i to vv, where vv is a signed 2's complement byte in the range -128 to +127. This range corresponds to the mechanical range from "pull" to "push", with 0 corresponding to "at rest".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a lever.

Pedal 1 90+i vv
Pedal 2 98+i vv

The Chroma sets the value of the appropriate pedal input on instrument i to vv, where vv is a number in the range 0 to 255. This range corresponds to the mechanical range from "heel" to "toe".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a pedal.

Footswitch 1 Down	A0+i
Footswitch 1 Up	A8+i
Footswitch 2 Down	B0+i
Footswitch 2 Up	B8+i

These commands activate or deactivate the footswitch functions on instrument i. Footswitch 1 is the right, or sustain, footswitch. Footswitch 2 is the left, or effects, footswitch.

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer presses or releases either footswitch.

Define C0+i pp aa bb cc dd ee ff

Instrument i is defined according to program pp (which must be in the range 0 to 50). The remaining bytes specify initial values for the performance inputs:

aa: lever 1 bb: lever 2
cc: pedal 1 dd: pedal 2
ee: volume ff: footswitches

The footswitch byte uses the most significant bit to represent footswitch 1 and the next most significant bit to represent footswitch 2. A 0 means up, 1 means down. If pp is not between 0 and 50, the Chroma will not define the instrument according to garbage data, but nothing more is promised.

This command causes channel boards to be reallocated as fairly as possible among defined instruments. If this command requires that one or more channel boards be robbed from another instrument, the computer will be kind enough to try and pick boards that aren't currently sounding.

This command will be transmitted (for instrument 0 and possibly 1) by the Chroma if the panel switch is on and the performer selects a program or a link. Although instrument 0 is internally defined by program 0, the Define command that is transmitted whenever a program is selected includes the current program number as shown in the 2-digit display.

Undefine C8+i

Instrument i is removed from operation, and any channel boards assigned to it are redistributed among any other instruments.

This command will be transmitted for instrument 1 by the Chroma if the panel switch is on and an unlinked program is selected or a link is cleared.

Attack D0+i kk vv pp

Instrument i is told to attack note kk with a velocity vv and an initial pressure pp. The key number is a signed, 2's complement byte that must be in the range -64 to +63. The Chroma's keyboard has a range from -32 to +32, with 0 being middle C. The velocity must be a number from 0 (softest strike) to 31 (hardest strike), and the pressure must be a number from 0 (no pressure) to 63 (full pressure).

The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer presses a key.

Early Chromas transmit a pressure byte that is always zero, and ignore the received pressure byte. Current Chromas respond to pressure information received even if the Pressure Sensor option is not installed.

Release D8+i kk vv

Instrument i is told to release note kk with a velocity vv. The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer releases a key.

Set Parameter E0+i nn vv

Instrument i temporarily sets parameter nn to value vv. This does not affect the setting stored in non-volatile memory, which means that it won't affect other instruments defined according to the same program and it won't affect this instrument if it is redefined according to the same program. Only those parameters that pertain to the tone generation may be set with this command. These include:

1 through 5: control parameters
6 through 50: A parameters
55 through 100: B parameters

Any other parameter number will cause the command to be ignored. If vv is not within the valid range for the selected parameter, the only guarantee is that the parameter will not be set to an illegal value.

This command will be transmitted for instrument 0 by the Chroma if the panel switch is on and the performer varies one of the parameters.

Status E8+i

This command causes the Chroma to respond with:

E8+i pp aa bb cc dd ee ff

where the seven parameters represent the same quantities as the parameters of the Define command. If the instrument is undefined, the program number returned will be FF and the remaining bytes will be undefined. If the program number is 0, the program number in the display will be used instead.

Squelch F0+i kk

Any channels in instrument i that are assigned to key k are squelched by setting their envelopes to 0. This doesn't affect the channel assignment tables. Even latched channels may be squelched. If kk is -128 (80 hex) all channels will be squelched.

SOFTWARE REQUIREMENTS

Levels of Complexity -- The complexity of the software needed to communicate with the Chroma is dependent upon the kind of communication desired. The most important factor is whether or not the software can wait for input from the Chroma. A simple system can be designed in which all communication is essentially half-duplex, meaning that when the computer is expecting information from the Chroma it is doing nothing else. This precludes recording and playing concurrently. In fact, it precludes doing much more with information arriving from the Chroma than storing it for later processing.

In order to allow more processing to occur in response to information arriving from the Chroma (as opposed to processing that is totally independent of what the Chroma might be sending), it is advisable to make the input system interrupt driven. This would allow information to be taken into the computer as soon as the Chroma sends it, where it would be queued until it could be processed.

If it is desired to record and process information from the Chroma while doing a large amount of unrelated processing (such as communicating with other instruments or a display terminal), some form of multi-tasking is necessary. A generalized multi-tasking operating system would be nice, but hardly necessary. The Chroma firmware is itself structured as two concurrent tasks, as the Chroma has plenty of stuff to do besides wait around for commands to arrive on the interface.

If it is important that outputting information to the Chroma be fast, a queue can be provided for outgoing information, and an interrupt can be used to move bytes from the queue onto the port. This is also done in the Chroma.

A Simple System -- The simplest form of communication doesn't require any fancy software support. It is only necessary to wait for the output port to be empty before outputting each byte, and wait for the input port to be full before inputting each byte. BASIC peeks and pokes are sufficient for handling programming information, although most interpreted versions of BASIC aren't really fast enough to implement a decent sequencer. Note also that it would be advisable to include some method of getting out of the loop that waits for input from the Chroma (such as pressing a key on the computer terminal) to prevent communications problems from hanging the computer. A loop with a timeout might be appropriate when the computer requests specific information from the Chroma. Unless the Chroma is doing an autotune or cassette operation, it should respond to any command within a couple milliseconds.

A Sytem With Interrupt Driven Input -- This kind of system is what most people will probably be interested in playing with. The Hardware Requirements section of this manual shows an interface circuit that includes provisions for interrupting the processor when either the input port is full or the output port is empty. The output interrupt is less important, so the gates needed for this can be left out if not desired. The purpose behind making the input interrupt driven is that it keeps the real-time constraints of the Chroma from extending into the bulk of the computer software. This is because it allows rapid bursts of information from the Chroma to be handled as long as the computer can keep up with the average rate of information flow. The software necessary to do interrupt driven input consists of three procedures. The initialization procedure sets up the queue pointers and enables the interrupt. The interrupt handler pulls bytes off the port and stuffs them in the queue. The input procedure pulls bytes out of the queue for processing. These algorithms are presented below. Note that the interrupt handler must be written so that it returns with the interrupt masked in the event that the queue is full, and the input routine must, upon removing a byte from the queue, reenable the interrupt.

INTERRUPT DRIVEN INPUT ALGORITHM

PROCEDURE TO INITIALIZE INTERFACE -- called upon start-up

```
set head and tail queue pointers to zero
enable input interrupt
```

INTERRUPT HANDLER

```
input byte into tail of queue
advance queue tail pointer
if input queue full
    disable interrupt
```

INPUT PROCEDURE

```
wait for queue to be not empty
remove byte from head of queue
disable interrupt
advance queue head pointer
enable interrupt
return byte
```

A fourth procedure might be provided to check to see if anything is in the queue without actually waiting in a loop.

A Fully Interrupt Driven Dual Task System -- This is really a description of the way the Chroma handles its end of the interface. The algorithms described below show how the interface software might be written to allow inputting information to be handled as a separate, parallel process, without the use of a multi-tasking operating system. The purpose of multi-tasking is to allow a computer to take turns doing more than one thing, giving the appearance that it is doing them simultaneously.

When a computer has more than one task to perform, some mechanism must be provided for deciding which task should be handled at any given instant. In the Chroma, there are two tasks, one controlling the synthesizer and one responding to commands from the interface. Deciding which task should be performed is simple. If a byte is available from the interface, it is processed. If no byte is available, one complete cycle of the synthesizer firmware is performed (lasting about 1.25msec). This "multi-tasking" is characterized by the following parameters:

The synthesizer task is only suspendable at one particular point in its loop, when it isn't in the middle of anything.

The external input task is only suspendable when it must wait for input.

The external input task has the higher priority.

The first rule makes the interface between the two tasks clean. There is no danger of the external interface task manipulating something that the synthesizer task was in the middle of manipulating when it was suspended.

In order to implement two parallel tasks, it is necessary to save all the information representing the state of one task while running the other task. If a task is suspendable at only one point, and under the same conditions every time, no state information is required. If the task is to be suspendable in more than one place under different conditions, this information must be saved. Also, tasks that don't always leave their stack pointers in the same place each time they are suspended cannot share the same stack. In the Chroma, the external input task is carefully written so as not to use the stack when calling for a byte of input for anything other than the return address. This allows it to function without a separate stack.

The algorithms presented below treat the synthesizer task as the "background" task and the external input task as a "priority" task. The synthesizer task checks the input queue every now and then and, if a byte is found, causes the external input task to be resumed. The external input task consists of a command interpreter that calls the input routine from many different points, meaning that the task state image must include a return address. The state image is created during initialization with a PC value that points into the beginning of the command interpreter. Thus, the first byte that arrives will cause the input procedure to resume, and "return" to the command interpreter, which will handle the received byte and ultimately call the input procedure again. The nice thing about this type of system is that it allows you to write the external input handling software as though it were the only thing going on. The details of switching between the two tasks are hidden inside the input procedure.

INTERRUPT DRIVEN DUAL TASK ALGORITHM

```
PROCEDURE TO INITIALIZE INTERFACE
    -- called upon start-up

    set input and output head and tail pointers to zero
    set non-responding flag -- this gets reset when first byte arrives
    create external input process state image
         -- PC (program counter) cell must point into command
         -- interpreter, as if command interpreter had called
         -- input procedure for first command code byte
    enable input interrupt -- output interrupts remain disabled

INPUT AND OUTPUT INTERRUPT HANDLER
    -- handles both interrupts arriving on one line, with a round-robin
    -- priority scheme for concurrent input and output interrupts

    go to ENTRY -- start by checking for output interrupt

  INPUT:
    clear non-responding flag
    input byte and put into tail of input queue
    advance input queue tail pointer
    if input queue full
         disable input interrupt

  ENTRY:
    if output interrupt pending
         go to OUTPUT
    else if input interrupt pending
         go to INPUT
    else return

  OUTPUT:
    output byte from head of output queue
    advance output queue head pointer
    if output queue empty
         disable output interrupt

    if input interrupt pending
         go to INPUT
    else if output interrupt pending
         go to OUTPUT
    else return

PROCEDURE TO DISPATCH EXTERNAL INPUT PROCESS
    -- called every 1msec or so by the main process

    if input queue not empty
         remove byte from head of input queue
         if input interrupt disabled -- meaning queue was full
              enable input interrupt
         save machine state
         restore external input process machine state
         return byte -- to external input process
```

```
PROCEDURE TO INPUT ONE BYTE
     -- called by external input process

     if input queue not empty within 100usec
          remove byte from head of input queue
          advance input queue head pointer
          if input interrupt disabled -- meaning queue was full
               enable input interrupt
          return byte
     if input queue still empty after 100usec
          save machine state
          restore main process machine state
          return -- to main process

PROCEDURE TO OUTPUT ONE BYTE
     -- called by either main or external input process

     if non-responding flag set
          return
     if output interrupt disabled -- meaning queue not in use
          if output port empty within 100usec
               output byte
               return
     -- otherwise, port was full or queue was already in use
     if room in output queue within 1.5msec
          put byte in tail of output queue
          advance output queue tail pointer
          enable output interrupt
     if output queue still full after 1.5msec
          set non-responding flag
          disable output interrupts
          set input and output head and tail pointers to zero
          reinitialize external input process state
```

With two hardware-prioritized interrupt lines, the interrupt handler could be split in two. Note also the 100usec timed loops that actually speed up data transfers by increasing the liklihood that a multi-byte data transfer can be handled without re-interrupting for each byte. The non-responding flag is a mechanism used in the Chroma to handle the case of a crashed computer at the other end of the interface. It is cleared by incoming bytes and set if the output doesn't respond within a reasonable time.

Time Measurement -- Computer software to aid in programming the Chroma does not require any timing circuitry. However, if you intend to record and play back music with the Chroma, time measurement becomes very important. The Chroma is pretty good about playing music that arrives over the interface without any noticeable time lag. However, if you intend to use an interrupt driven input system, you must make sure that the delay between a byte's acceptance by the interrupt handler and its ultimate processing doesn't cause timing errors in the music. The easiest way to assure this is to let the interrupt handler record the time each byte arrives. Each byte in the input queue will therefore be accompanied by time information. The timing resolution should be better than 10msec, yet anything faster than 1msec is probably extraneous. A sixteen bit timer will provide enough time information, as long as commands don't arrive from the Chroma further apart than half the cycle time of the timer (more than thirty seconds at 1msec resolution). Thus, the input queue should be able to hold two time bytes for every command byte.

Once the information is pulled from the queue for processing, the time bytes associated with command operands can be eliminated, leaving only the time bytes associated with each command code byte. The absolute time measurements might also be converted to relative time between events, if that is more appropriate to the processing that is to be performed.

Utilizing The Command Language Of The Chroma -- The structure of a music recording and playback system is further impacted by the fact that the communication in each direction upon the interface is independent, yet the information flowing in each direction is not. To clarify, consider the case of the Performance Switch Off command. This command is normally sent to the Chroma as a signal that you are finished recording and no more information is to be accepted. But it is entirely possible, given the amount of buffering that the information must suffer, that further performance information will be transmitted in the milliseconds after this command has been issued. Even though the Chroma does in fact stop transmitting when it sees the Performance Switch Off command, there is no guarantee as to how long this will take. The difficulty is handled by the fact that all such mode change commands are echoed by the Chroma. Thus, the stream of data coming back from the Chroma will include "flags" that frame the information so that the computer knows when the Chroma is done transmitting. The correct way to start and stop recording from the Chroma is to keep a status flag that is set by receipt of a Performance Switch On command and cleared by receipt of a Performance Switch Off command. When recording is to commence or terminate, the computer should send the appropriate command, but the state of the status flag, controlled by the echoed commands, should start and stop the actual recording process.

The Restore command is actually the command most likely to be used to terminate recording. This command is provided as a convenience, making it unnecessary to explicitly restore the instrument definitions that were in effect when the recording started. Note, though, that the first few Chromas built do not echo the Performance, Panel and Pressure Switch Off commands when the Restore command is received, but current Chromas do. Refer to Appendix A.

When recording, with the panel, performance and/or pressure switches on, the Chroma will transmit "commands" with instrument codes 0 and 1. In order to allow playing and recording at the same time, instruments that are used for playback should be assigned higher numbers. If the interface is sending commands to instruments 0 and 1, there is nothing to prevent the performance controls and panel controls to send commands to these instruments at the same time. This shouldn't cause a problem, but it won't sound very good either. Note that the commands that are sent by the Chroma during recording are all in exactly the form (except for instrument number) that they should be transmitted back to the Chroma during playback. No other information will be sent by the Chroma unless it is explicitly requested.

HARDWARE REQUIREMENTS

This section describes the circuits needed to interface a computer to a Chroma. It is assumed that the reader is reasonably familiar with the bus structure of his own computer, and what is shown here will have to be modified accordingly. In particular, details of address decoding, bus acknowledge and interrupt control are not shown.

Minimal Interface -- At the very least, one must have an 8-bit latch driving the output lines that can be written into, an 8-bit tri-state driver sensing the input lines that can be read from, and a set/reset flip-flop associated with the output lines that maintains the status of the output port. (The other end of the interface maintains the status of the input port.) In addition, there must be a way of sensing the status of the two ports. This is shown in Fig. 5-1. It is assumed here that the bus cycle strobe, read and write signals and address are all decoded to provide individual active-low strobes to all circuits that require them, that acknowledgement is taken care of elsewhere (no wait states should be needed), and that the data bus is an 8-bit positive logic bus. Note that the XOACK (External Output Acknowledge) and XIFULL (External Input Full) lines are resistor terminated. This is because the Chroma drives these lines with open collectors.

Modified Minimal Interface -- The bus drivers shown here are LS TTL devices, as their low power consumption is convenient in a microprocessor system. For slightly better noise immunity, the Shottky outputs can be buffered by regular TTL parts which pull down closer to ground, although this is not done in the Chroma. The Chroma uses RC networks on all its outputs to avoid RFI problems with the FCC, but you can do what you like at your end of the interface. You might also put a couple of transistors on the Output Full and Input Ack lines to make sure that these outputs are not activated except when the computer is turned on. Otherwise, powering down the computer would continuously interrupt the Chroma. These modifications are shown in Fig. 5-2.

Interrupt Driven System -- The status of each port can be used to generate an interrupt. The input port should be capable of generating an interrupt when it is full, and the output port should be capable of generating an interrupt when it is empty. In addition, each interrupt should be independently maskable. If a multi-level interrupt structure already exists in your system, you will only need to connect the status lines (inverting one of them) to two interrupt lines. Otherwise, a 2-bit output port must be provided to hold the interrupt mask bits, and some gates must be provided to combine everything into one interrupt line. This is shown in Fig. 5-3. The circuit shown maintains the interrupt until the condition causing it is removed. Some systems will require an open collector interrupt signal.

A Simple Free-running Timer -- The circuit shown in Fig. 5-4 consists of a prescaler to convert a high-frequency clock signal into a more usable frequency of perhaps a few hundred hertz. This drives the counters inside a pair of 74LS590 chips[1]. Each of these chips has an 8-bit synchronous counter, an 8-bit latch, and an 8-bit tri-state driver. Note that the register clock inputs (RCK) are tied together to a separate strobe line, which must be strobed before reading the counter. Although a programmable prescaler might be nice, its function can be handled in software if you are reasonably clever.

Connecting to the Chroma -- The physical interconnection to the Chroma is through the 25-pin D-type connector on its rear panel. Figure 5-6 shows the pin-out of the connector. Note, however that the names of the connections shown in this diagram are from the Chroma's point of view. That is, the lines that are associated with the "output" port deal with information flowing out of the Chroma, and the lines that are associated with the "input" port deal with information flowing into the Chroma.

Obviously, the Chroma's output lines must connect to the computer's input lines and vice versa. There are two ways to do this. If you wish to use an inexpensive off-the-shelf ribbon cable, you must assign pin numbers at the computer's end of the interface according to the scheme in figure 5-7. This will connect the XO lines at one end to the XI lines at the other. Alternatively, you may assign the pin numbers at the computer's end the same as the Chroma, as shown in figure 5-6, and use a "crossover" cable. Rhodes sells a sturdy shielded crossover cable, with molded plugs at each end, that was designed to interface a Chroma to an Expander. The use of this cable (and the pin-out of figure 5-6) is recommended, as it simplifies larger systems: anything that can transmit can be connected to anything that can receive without concern about whether the pin-out is correct.

[1] See the 1981 Supplement to The TTL Data Book, Texas Instruments.

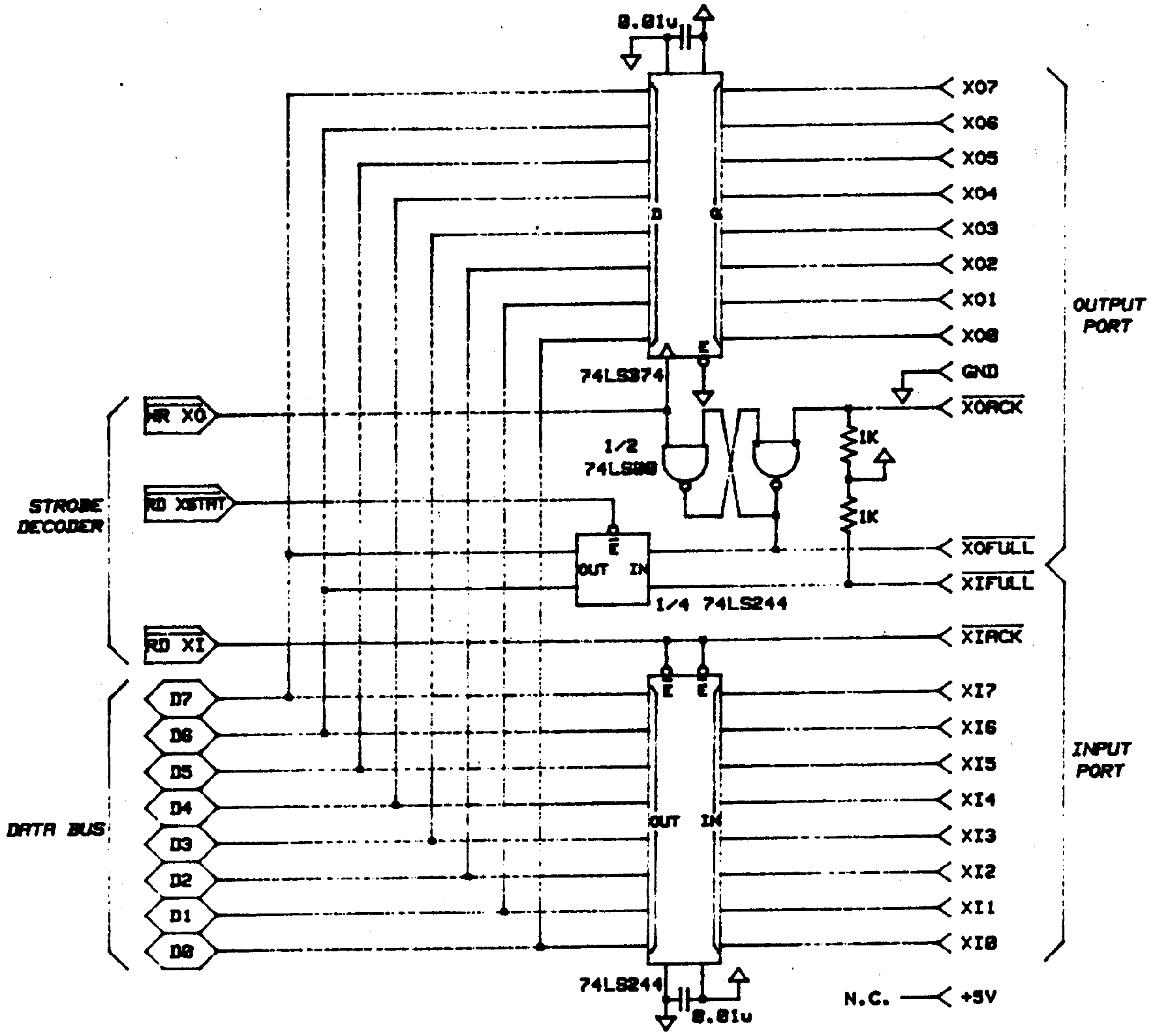


Fig. 5-1 Minimal Interface

NOTE: All resistors 100ohm,
all capacitors 0.01uF, unless
otherwise specified.

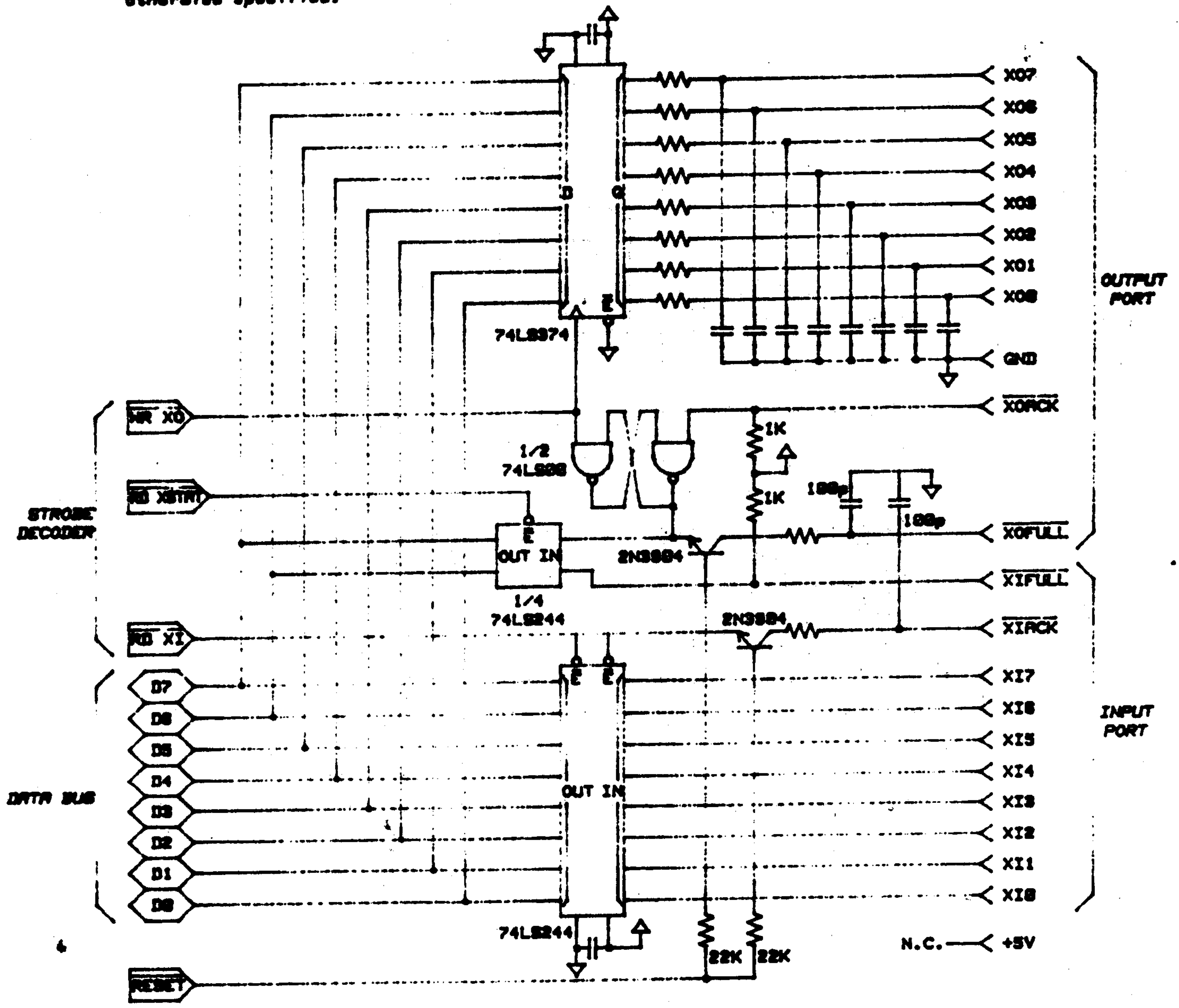


Fig. 5-2 Modified Minimal Interface

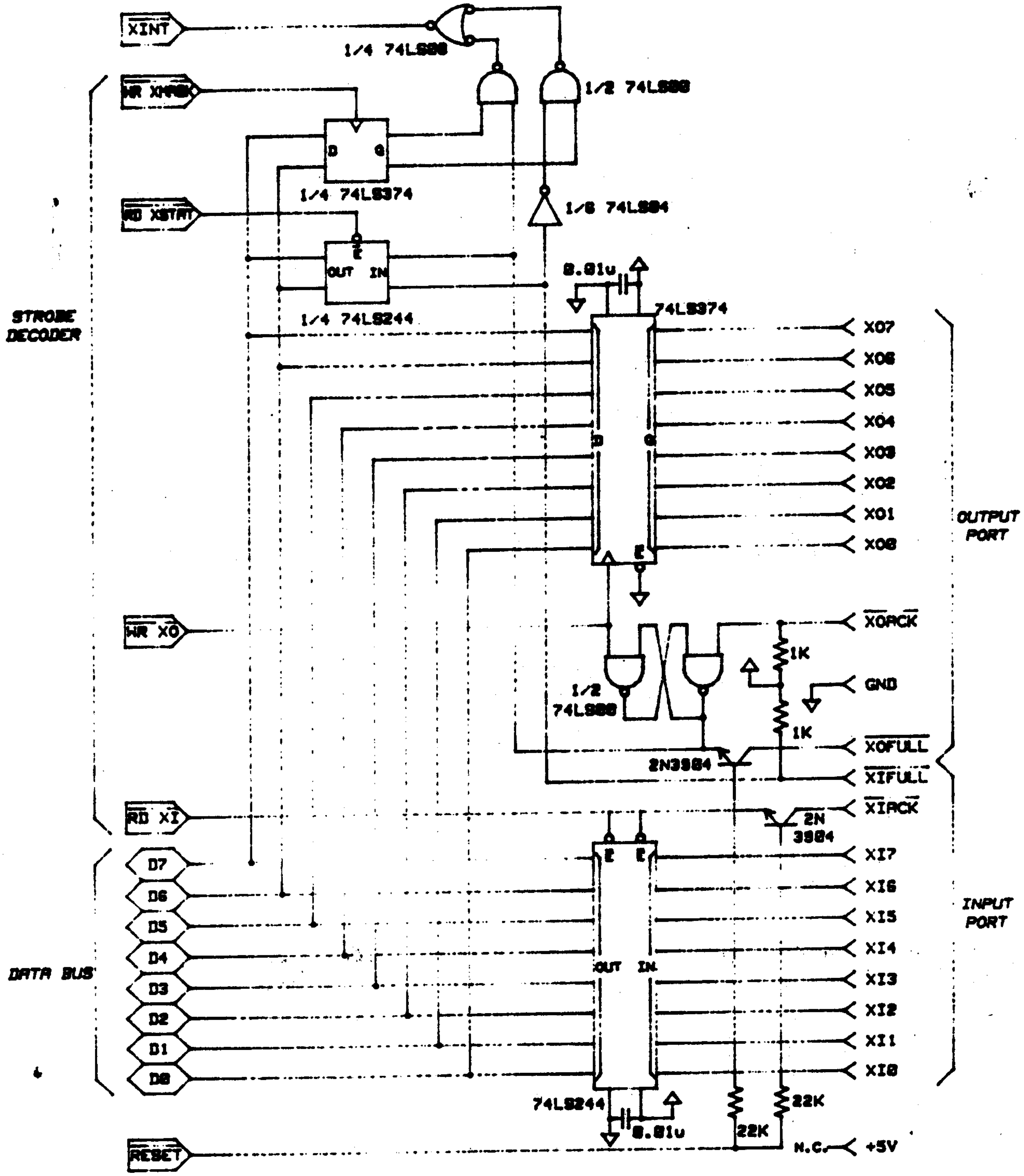
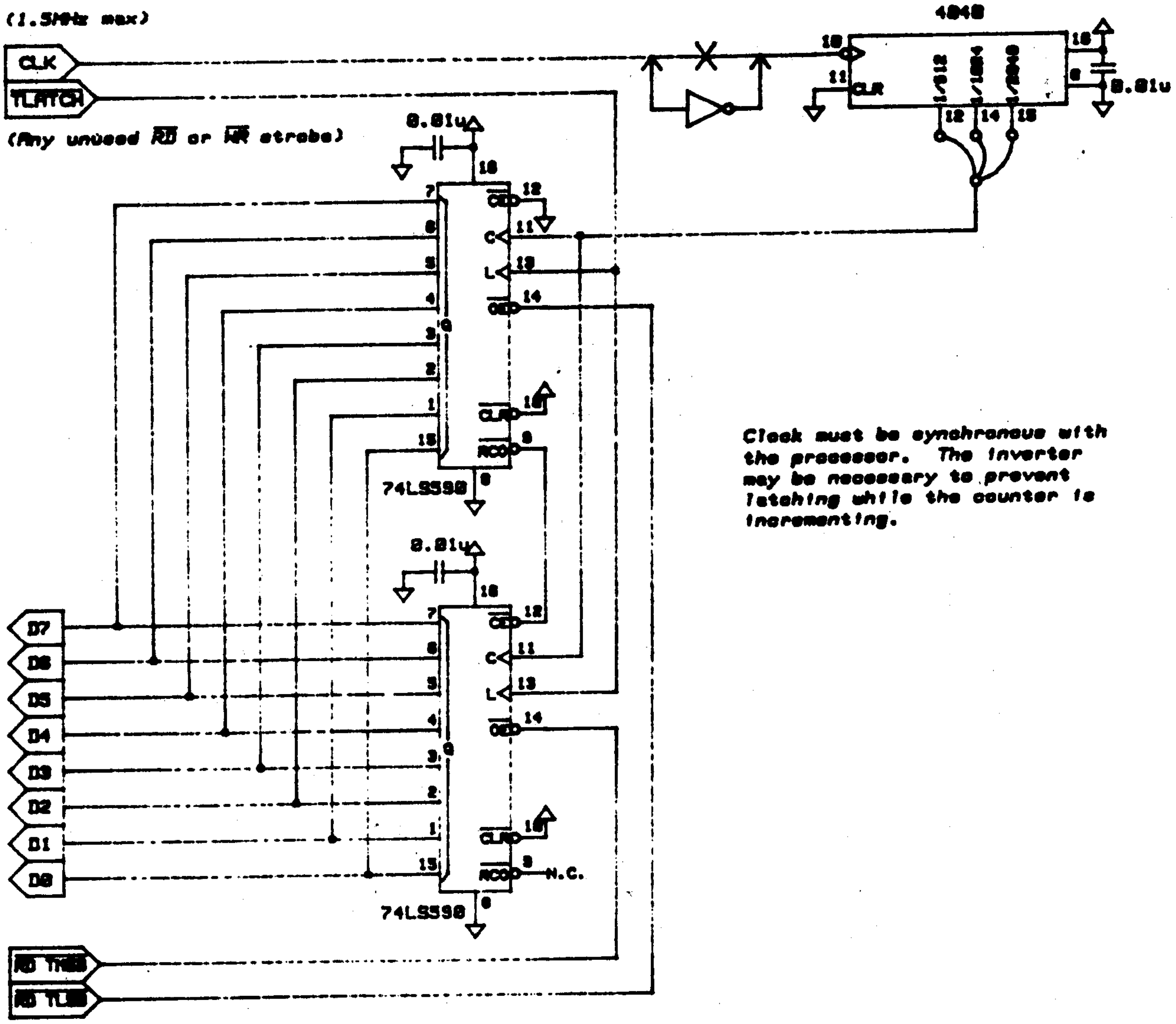
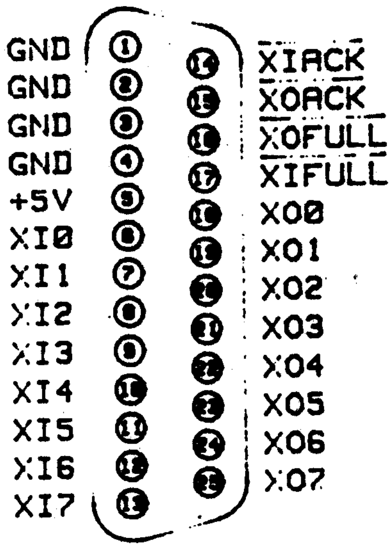


Fig. 5-3 Interrupt Driven Interface



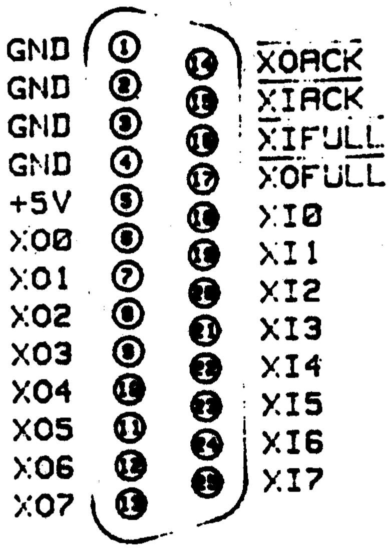
Clock must be synchronous with the processor. The inverter may be necessary to prevent latching while the counter is incrementing.

Fig. 5-4 Free-running Timer



Note: This pinout should be used at the computer's end of the interface, along with the "crossover" type cable.

Fig. 5-6 Chroma D-connector Pin-out



Note: This pinout may be used at the computer's end if an off-the-shelf ribbon cable is to be used. See text.

Fig. 5-7 Complementary Pin-out

APPENDIX A
REVISION INFORMATION

This section outlines the differences in behavior between Chromas and Expanders of different revision levels. The Interface Revision Number is returned in response to the Identification command. Please note that this number is not the same as the Software Revision Number as imprinted on the EPROMs inside the unit.

The revision levels are described in reverse order, starting with the current revision. Each revision description outlines the differences between that revision and the subsequent revision above it.

REV 3 (software REV 13) -- This is the current revision, as described in this manual.

REV 2 (software REV 12) -- This revision did not include the pressure sensor commands. This results in the following restrictions:

> The Pressure Switch Off and Pressure Switch On commands are treated as No Operations. They are not echoed.
>
> The Restore Command does not echo a Pressure Switch Off command.
>
> The pressure byte in all Attack commands sent by the Chroma is 0. The pressure byte in all Attack commands received by the Chroma is ignored (although it must be present).

A bug was found in this revision:

> If a link is in effect and a lever or pedal is moved, the Chroma will not transmit an instrument 0 and an instrument 1 command. Instead, the Chroma will send two identical instrument 0 commands. This only applies to the Lever 0, Lever 1, Pedal 0 and Pedal 1 commands.

REV 1 (Software REV 10) -- A number of bugs were found in this revision:

> If a Footswitch command is sent to any instrument that has never been defined since power-up, it will crash the Chroma.
>
> The Restore command does not do anything to instrument 1, regardless of the link.
>
> Bytes coming from the Chroma occasionally get rearranged and are transmitted out of sequence. This only occurrs if the computer makes the Chroma wait more than 100usec or so, and the Chroma starts to use its output queue. If you experience problems at high data rates, suspect this.
>
> Although the Restore command turns off the panel and performance switch, it does not echo the Panel and Performance Switch Off commands.

Upgrading A Chroma -- All it takes to bring a Chroma up to the current revision is to unplug the EPROMs and plug in new ones. This can be done by any authorized Rhodes Chroma service center, and is free if the instrument is under warranty. Upgrading is strongly recommended, as old software is only old because there was something wrong with it.

To request an upgrade from a service center, always refer to the Software Revision Number which is printed on the EPROMs, not the Interface Revision Number. Most service centers are not aware of Interface Revision Numbers.

The current revision includes provision for the Pressure Sensor option. This does not mean that the Pressure Sensor must be installed. The Chroma will respond to Pressure commands whether or not the option is installed. It just won't generate correct Pressure commands.

APPENDIX B
PARAMETER INFORMATION

A Chroma program consists of 101 parameters divided into four categories:

Panel Parameters -- These are parameters 0 and 51 through 55, and include the parameters that represent the states of various panel controls. These are described in detail below.

Control Parameters -- These are parameters 1 through 5, and are accessible from the panel using switches 1 through 5.

A Channel Parameters -- These are parameters 6 through 50, and are accessible from the panel using switches 6 through 50 in Edit A mode.

B Channel Parameters -- These are parameters 56 through 100, and are accessible from the panel using switches 6 through 50 in Edit B mode.

The Control, A Channel and B Channel Parameters are fully described in the Chroma Programming Manual. Their values appear in signed two's complement form over the interface.

There are six panel parameters. The panel parameters stored in programs 1 through 50 have no effect, even if an instrument is defined by the program. The only panel parameters that have any effect are those in program 0, and they represent the current states of the "programmable" panel controls:

0. Link Balance -- This is what appears in the Data Readout when a link is set up, although the values are different. The value shown in the display ranges from -14 to +14 in steps of 2, while the value accessible through the interface ranges from -7 to +7 in steps of 1.

51. Link -- This parameter includes the Link Mode and the Link Program Number in a single byte value. The six lsbs represent the Link Program Number, which must be between 1 and 50. The two msbs represent the Link Mode as follows:

00 = No Link, 01 = Link Upper, 10 = Link Lower, 11 = Link Unison

52. Edit -- This parameter includes the Edit Mode and the currently selected parameter number. The six lsbs represent the parameter number, which must be 0 if the Link Balance parameter is selected, or a number from 1 to 50 if a Control or Channel parameter is selected. The two msbs represent the Edit Mode as follows:

01 = Edit B, 10 = Edit A, 11 = Edit A&B

53. Keyboard Split -- This is a number from -32 to +31.

54, 55. Main, Link Transposes -- These represent the settings of the transpose switches as follows:

00 = Normal, 01 = Up 1 Oct, 10 = Down 1 Oct

PROGRAM TABLE LAYOUT

The Read Program and Write Program commands deal with complete sets of parameters as they appear in the Chroma's non-volatile memory, packed into 59 bytes. The following table shows the location of each parameter:

No(s)	Group	Name	Byte(s)	7 6 5 4 3 2 1 0
0	Panel	Link Balance	31	[- - - - N N N N]
1	Control	Patch	1	[- - - - N N N N]
2	Control	Fsw Mode	5	[- - - - - N N N]
3	Control	Kybd Alg	31	[N N N N - - - -]
4	Control	Detune	2	[N N N N N - - -]
5	Control	Output Select	2	[- - - - - N N -]
6,56	Glide	Rate	28,58	[N N N N N - - -]
7,57	Glide	Shape	14,44	[- N - - - - - -]
8,58	Sweep	Mode	4,34	[- - - - - - N N]
9,59	Sweep	Rate	4,34	[N N N N N N - -]
10,60	Sweep	Rate Mod	3,33	[- - - - N N N N]
11,61	Sweep	Wave Shape	6,36	[N N N N - - - -]
12,62	Sweep	Ampl Mod	6,36	[- - - - N N N N]
13,63	Env 1	Ampl Touch	9,39	[- - - - - N N N]
14,64	Env 1	Attack	7,37	[N N N N N - - -]
15,65	Env 1	Attack Mod	7,37	[- - - - - N N N]
16,66	Env 1	Decay	8,38	[N N N N N - - -]
17,67	Env 1	Decay Mod	8,38	[- - - - - N N N]
18,68	Env 1	Release	9,39	[N N N N N - - -]
19,69	Env 2	Delay	10,40	[N N N N N - - -]
20,70	Env 2	Ampl Touch	13,43	[- - - - - N N N]
21,71	Env 2	Attack	11,41	[N N N N N - - -]
22,72	Env 2	Attack Mod	11,41	[- - - - - N N N]
23,73	Env 2	Decay	12,42	[N N N N N - - -]
24,74	Env 2	Decay Mod	12,42	[- - - - - N N N]
25,75	Env 2	Release	13,43	[N N N N N - - -]
26,76	Pitch	Tune	14,44	[- - N N N N N N]
27,77	Pitch	Mod 1 Select	18,48	[N N N N - - - -]
28,78	Pitch	Mod 1 Depth	15,45	[- N N N N N N N]
29,79	Pitch	Mod 2 Select	18,48	[- - - - N N N N]
30,80	Pitch	Mod 2 Depth	16,46	[- N N N N N N N]
31,81	Pitch	Mod 3 Select	19,49	[N N N N - - - -]
32,82	Pitch	Mod 3 Depth	17,47	[- N N N N N N N]
33,83	Width	Wave Shape	20,50	[- - - - - - N N]
34,84	Width	Width	20,50	[N N N N N N - -]
35,85	Width	Mod Select	19,49	[- - - - N N N N]
36,86	Width	Mod Depth	21,51	[- N N N N N N N]

No(s)	Group	Name	Byte(s)	7 6 5 4 3 2 1 0
37,87	Cutoff	LP/HP	15,45	[N - - - - - - -]
38,88	Cutoff	Resonance	10,40	[- - - - - N N N]
39,89	Cutoff	Tune	22,52	[- - N N N N N N]
40,90	Cutoff	Mod 1 Select	26,56	[N N N N - - - -]
41,91	Cutoff	Mod 1 Depth	23,53	[- N N N N N N N]
42,92	Cutoff	Mod 2 Select	26,56	[- - - - N N N N]
43,93	Cutoff	Mod 2 Depth	24,54	[- N N N N N N N]
44,94	Cutoff	Mod 3 Select	27,57	[N N N N - - - -]
45,95	Cutoff	Mod 3 Depth	25,55	[- N N N N N N N]
46,96	Volume	Mod 1 Select	27,57	[- - - - N N - -]
47,97	Volume	Mod 1 Depth	3,33	[N N N N - - - -]
48,98	Volume	Mod 2 Select	27,57	[- - - - - - N N]
49,99	Volume	Mod 2 Depth	5,35	[N N N N - - - -]
50,100	Volume	Mod 3 Select	28,58	[- - - - - N N N]
51	Panel	Link	0	[N N N N N N N N]
52	Panel	Edit	30	[N N N N N N N N]
53	Panel	Keyboard Split	32	[N N N N N N N N]
54	Panel	Main Transpose	1	[N N - - - - - -]
55	Panel	Link Transpose	1	[- - N N - - - -]
	Sequence Program Footswitch		29	[N N N N N N N N]
	Free bits		2	[- - - - - - - N]
			5	[- - - - N - - -]
			35	[- - - - N N N N]
			14,44	[N - - - - - - -]
			16,46	[N - - - - - - -]
			17,47	[N - - - - - - -]
			21,51	[N - - - - - - -]
			22,52	[N N - - - - - -]
			23,53	[N - - - - - - -]
			24,54	[N - - - - - - -]
			25,55	[N - - - - - - -]

Note -- The Sequence Program Footswitch byte contains the number of the program that will be selected when the Sequence Program Footswitch is depressed and released. This number is not accessible as a regular parameter. Even though this only requires six bits to represent, the full eight bits are reserved.

Note -- The Keyboard Split parameter is a signed number from -32 to +31. Even though this only requires six bits to represent, the full eight bits are reserved.

Note -- Signed parameter values are represented in two's complement format with the leftmost bit assigned to the parameter being the sign bit. Thus, a mod depth of -10 would appear in seven bits as 1110110.

APPENDIX C
USEFUL LOCATIONS WITHIN THE CHROMA

The memory address space within the Chroma is accessible through the use of the Peek, Peek Two Bytes, Poke and Poke Two Bytes commands. Although the usefulness of these commands depends upon an understanding of the internal structure of the Chroma firmware, there are some locations within the Chroma's address space that can be manipulated without really knowing what goes on inside the instrument. A few of these are documented here as an aid to anyone who wishes to experiment with them.

Revisions often involve rearranging the locations of things inside the computer's address space. The only thing that are likely to move, though, are those items contained in volatile RAM between locations 0100 and 0FFF.

Once you issue the Unlock command and start Poking into the Chroma, you have the capability of crashing the Chroma's computer. A crashed Chroma must be powered down and up again to restart it. It is possible that a crash may corrupt the programs stored in non-volatile RAM (possibly only a byte here and there). Therefore, the 50 programs should be reloaded via cassette (or via the interface) in the event of a crash.

Be sure to use the Peek Two Bytes and Poke Two Bytes commands whenever you are accessing a two byte value. This insures that the interface's access to the locations does not get interleaved with the Chroma's access to the same locations.

0020-0029: Display image -- This contains the image of the ten display digits. Changes made to the first eight locations (the Data Readout) only last until some action occurs on the panel that involves the display. Changes made to the last two locations (the Program Number) last until a program is selected or stored.

002A: LED image byte 1 -- This contains the image of those eight LEDs that are capable of being flashed. Writing to this location will cause the appropriate LEDs to change state, as this location is copied to the LED port byte 1 at regular intervals.

002B: LED image byte 2 -- This contains the image of those eight LEDs that never flash. Writing to this location does not make the LEDs change state, so the LED port byte 2 should be written as well.

002C: LED blink image -- This byte is XORed with the LED image byte 1 at regular intervals to cause LEDs to flash. This should normally be set whenever the image byte is set.

0048, 0049: Master tune -- This two-byte number contains the master tune setting for the instrument. Whenever the tune slider is moved, this will be set to an even number between -256 and +246, which represents a range from -1 to almost +1 semitones. As long as the master tune slider is not moved, this location can be played with through the interface.

005A, 005B: Safe buffer program number, modified flag -- This contains information about the contents of the safe buffer. Whenever a program is selected or stored, the safe buffer is used to store a backup copy of whatever is written over.

006A, 006B: Major loop hook -- Every 20 milliseconds or so, an indirect subroutine call is made through this location. Normally, this contains C100, the address of a Return instruction.

006C, 006D: Minor loop hook -- Every 1.25 milliseconds or so, an indirect subroutine call is made through this location. Normally, this contains C100, the address of a Return instruction.

0140-017A: Safe buffer -- Whenever a program is selected or stored, whatever is written over is first copied here.

017B-01BE: Stack -- The stack grows from high memory to low, and we have never seen it go below 0190, so that leaves twenty locations that are probably free. Assign from 017B up, to be safe.

01BF-02BF: Cassette buffer -- These 257 bytes are where each packet is stored as it is read from or written to the tape. Although it is possible to have a packet that fills the buffer, the packets used in ordinary cassette operations never exceed sixty bytes, so locations 01FB to 02BF are generally free. Assign from 02BF down, to be safe. Note: In REV 1 and 2, the cassette buffer is located in 0BFF to 0CFF.

1000 to 13FF: Empty -- These locations correspond to the two empty chip locations on the computer board.

1400 to 1FC0: Programs -- The 51 programs (0 to 50) are packed into this area. Each program occupies 59 bytes.

1FC1 to 1FF0: Free non-volatile memory -- These 48 locations are not currently used. Assign from 1FC1 up, to be safe.

1FF1: Cassette type -- If bit 2 of this byte is set, normal cassette motor sense/control functions are enabled. If bit 2 is clear, the cassette motor is ignored.

1FF2: Program number -- The current program number, as shown in the display, is kept here.

1FF3: Modified flag -- The modified flag appears in bit 7 of this byte. The other bits must be zero.

1FFC: Attack threshold -- General modulation selection 13 (Threshold Velocity) and Ampl Touch settings 6 and 7 compare the velocity of each attack to this number, which must be between 0 and 31.

1FFD: Release threshold -- Envelope Release setting 31 causes each release velocity to be compared to this number, which must be between 0 and 31.

1FFE: Release slow value -- Envelope Release setting 31 causes this number to be used as the release parameter for releases slow than the above threshold. It should be between 0 and 31.

1FFF: Release fast value -- Envelope Release setting 31 causes this number to be used as the release parameter for releases faster than the above threshold. It should be between 0 and 31.

2006: LED port byte 1 -- This write-only location directly controls those eight LEDs that are capable of flashing. This location should not be directly referenced, as it is automatically rewritten from the **LED image byte 1** at regular intervals.

2007: LED port byte 2 -- This write-only location directly controls those eight LEDs that never flash. When this location is written, the **LED image byte 2** should also be written.

C100: Return -- This location contains a Return instruction. Software hook cells should always contain C100 when not in use.

APPENDIX D
COMMAND LISTING

This section provides a brief listing of the form of each command. A superscript denotes a byte count. For greater detail, refer to section 3.

Name	Command bytes	Response bytes
No Operation	00	
Identification	01	01 device revision
Read Program	02 prog	02 data^{59}
Write Program	03 prog data^{59}	
Load Packet	04	04 n data^{n}
Save Packet	05 n data^{n}	05 result
Read Parameter	06 prog param	06 value
Write Parameter	07 prog param value	
Panel Switch Off	08	08
Panel Switch On	09	09
Performance Switch Off	0A	0A
Performance Switch On	0B	0B
Peek	0C addr^{2}	0C n data^{n}
Peek Two Bytes	0D addr^{2}	0D data^{2}
Poke	0E addr^{2} n data^{n}	
Poke Two Bytes	0F addr^{2} data^{2}	
Tap Panel	10	
Unlock	11 00 FF	
Tape Space	12	12 result
Restore	13	08 0A 14
Pressure Switch Off	14	14
Pressure Switch On	15	15
Pressure	68+i key pressure	
Information	70+i	70+i boards 0 0 0
Volume	78+i volume	
Lever 1	80+i position	
Lever 2	88+i position	
Pedal 1	90+i position	
Pedal 2	98+i position	
Footswitch 1 Down	A0+i	
Footswitch 1 Up	A8+i	
Footswitch 2 Down	B0+i	
Footswitch 2 Up	B8+i	
Define	C0+i prog l1 l2 p1 p2 vol fsw	
Undefine	C8+i	
Attack	D0+i key velocity pressure	
Release	D8+i key velocity	
Set Parameter	E0+i param value	
Status	E8+i	E8+i prog l1 l2 p1 p2 vol fsw
Squelch	F0+i key	

RHODES® KEYBOARDS INSTRUMENTS

CHROMA^{tm} COMPUTER INTERFACE

Model 1611

REV 5

SEQUENCER MANUAL

Tony Williams

JANUARY 24, 1984

51 West 52 Street
New York, NY 10019

CONTENTS

APPENDICES

INTRODUCTION : CHROMA COMPUTER INTERFACE KIT

GENERAL DESCRIPTION:

The Chroma Computer Interface kit contains everything you need to connect a Chroma to an Apple II Plus or Apple IIe personal computer. When installed, the Chroma can send and receive live performance information and programming information. A multitrack sequencer software package is included to get you started. Other utility programs, such as voice program editing will soon be available, or you can write your own software utilities. With the Chroma's versatile command oriented communications language, the Chroma is the first intelligent music terminal; now you can create your own computer applications.

CHROMA COMPUTER INTERFACE:

Anything that you play, select, or move on the Chroma and its accessories can be sent to a computer or other device over the Chroma Interface. The external computer can then store the data and send it back to the Chroma in the form of a digital recording or sequence. As you can see, the Chroma and its interface are useful for studio as well as live performance and educational applications.

CHROMA INTERFACE COMMAND SET:

The Chroma is structured around its interface - not just modified to accomodate it. It has available eight (8) 'Instruments', each of which may be programmed and played individually by the external computer. The interface command set is extremely versatile, including commands such as ATTACK, RELEASE, PEDAL 1, PEDAL 2, PITCH BEND, MODULATION LEVER, LATCH, SUSTAIN, VOLUME, PRESSURE, SET VOICE PARAMETER and others for each Instrument. Also included are commands for saving and loading voices, saving and loading packets of data from the cassette interface, peeking and poking into the Chroma's internal memory, and reading or changing the values of any program in CMOS memory. Currently, there are a total of 166 such commands and room for expansion to 255.

SEQUENCER HARDWARE AND SOFTWARE:

Fender/Rogers/Rhodes now has available hardware and software to allow a Chroma and an Apple II Plus or IIe computer to talk to each other. Included with the package are a set of application programs (in Applesoft and Assembly Language) that perform sequencing, editing and Chroma program data storage functions. The Sequencer can record up to 16 independent tracks and can simultaneously control a Chroma and Chroma Expander. The Sequencer is fully polyphonic (up to eight (8) dual channel boards) and records key velocity and pressure as well as ALL performance controls and voice changes.

INTERFACE P.C. CARD:

The printed circuit (PC) card hardware and the Interface I/O Driver software (included on the 5-1/4 diskette with the System software) are required to establish communication between the Chroma and Apple. The hardware is configured as two interrupt driven unidirectional 8 bit parallel ports and four control lines. The expander is controlled by 2 polled ports. The PC card also incorporates a 1KHz clock for sequence timing and an Analog-to-Digital converter for pedal input of speed changes, etc. There are other single bit inputs for a footswitch, sync pulse and an external clock (for connecting to drum synthesizers and other sequencers, etc.). There is also a filtered pulse output for a audio click track. And finally, there are extra read and write pulses available if you want to modify the included hardware for your own software.

SEQUENCER SOFTWARE:

The Interface I/O Driver software performs all the functions necessary to get data to and from the Chroma. This driver, along with the Sequencer software, is included on the Sequencer Program diskette. This software receives data from the Chroma and formats it for storage on the Sequence Data diskette in Drive 2 (or on the Sequencer Program diskette, if you have only one drive). To record a track, all you really need to do is press 'R,' select a voice on the Chroma, press a footswitch and start playing. The Sequencer will remember all notes with velocities and pressures, all lever and pedal movements, all footswitch depressions, all voice changes, and even movements of the parameter control slider. When recording subsequent tracks, you need not start at the beginning of the sequence since the track starts wherever you play the first note.

The Sequencer will automatically store measures for endpoint and Editor references. The Sequencer has a built-in click track that can emphasize the first beat of the measure and has a continuously variable speed of 39-234 BPM.

To play a sequence all that is required is to press 'P' but there are many other options if you like - such as playing along, looping between two endpoints (defined by measures), muting tracks, etc. You can set a mode in which the Sequencer waits for a pulse (or footswitch depression) on the SYNC input before it starts playing.

Sequence editing can also be performed by changing the speed of the sequence, changing voices and volumes of individual tracks, changing a track from Chroma to Expander, deleting or changing notes, velocities and performance controls, transposing tracks, etc.

Finally, there are many functions that allow you to save sequences and groups of Chroma program data on diskette, display catalogs and set up direct communication between a Chroma and an Expander.

There are two banks of User Utility Program space if you want to write your own utilities. Each bank can contain up to ten (10) 2048 byte assembly language programs. We have included a few to get you started: a complete hardware test program, a program that allows you to write a 115 character comment in the sequence for your own use and a program that enables you to check the adjustment of the pedal conversion range.

APPLE REQUIREMENTS:

The Chroma Interface Kit Model 1611 requires an Apple II Plus 3.3 (with 48K and Autostart Rom), a 16K RAM card (Microsoft or Apple Language Card or other compatible ram card), one or two disk drives (although two is recommended), a CRT monitor, the Interface Kit and, of course, a Chroma. If you want to control both a Chroma and Expander or two Chromas, the optional extra cable is required. The control pedal for speed changing is also option equipment (you can use your Chroma pedal if you want).

The system will work if you have the old Apple II Monitor ROM only if you have an Apple Language Card or other compatible RAM card that provides the Autostart ROM.

If you have an Apple IIe, the 16K RAM card is not required, since this is built inside the IIe.

If you have only one (1) disk drive unit, you give up the capability of loading REV 4 sequences. You will also not have the extra mass storage necessary to make full use of the expanded RAM. In future revisions, we will require two drives for proper operation of the Sequencer.

80 column boards are not supported in this version of the Sequencer software.

Also optional is a Titan System's Saturn 128K card (stuffed to either 64K or 128K) or an Alphabyte 128K card (with the configuration switch in Saturn mode) to increase the note capacity of the Sequencer program. You can find the addresses of these two RAM card companies in the WARRANTY AND TRADEMARK INFORMATION section of this chapter.

WHAT IS IN THE KIT:

Included in the Chroma Computer Interface Kit is the following:

- The Printed Circuit Card
- The Connector Chassis Assembly
- An Interface Cable for one instrument
- Sequencer Program Diskette
- Sequencer Data Diskette for storing your sequences (included are some sample sequences and program groups, including the 3 factory sets)
- Footswitch
- User's Manual

Optional accessories include:

- An extra cable to control an Expander
- A Control Pedal for speed changing
- The Chroma Computer Interface Manual (discusses the Chroma Interface with regard to computers in general)

SEQUENCE MEMORY CAPACITY:

Currently the Sequencer storage capacity is 26128 bytes or approximately 1730 notes in non-expanded mode. Actual note capacity depends on the amount of performance control changes and your playing style. Multiple tracks and chords tend to increase the note capacity slightly. You can get an extra 100 notes or so by using the Scrunch utility.

If an expanded RAM card is installed, the note capacity depends upon the size of the RAM card. For 128K cards, expect around 8000 notes. For 64K cards, expect around 4000 notes. See the APPLE REQUIREMENTS section of this chapter for RAM card compatibility information.

WARRANTY AND TRADEMARK INFORMATION:

Please read this section carefully. It contains information about your rights in regard to this product.

The words APPLE, APPLE II, APPLE II PLUS, APPLE IIE, APPLESOFT and APPLE LANGUAGE CARD are registered trademarks of APPLE COMPUTER, INC.

APPLE COMPUTER, INC. was not in any way involved in the writing or preparation of this document or product, nor were the facts presented in the package reviewed for accuracy by that company. Use of the term APPLE should not be construed to represent any endorsement, official or otherwise, by APPLE COMPUTER, INC.

MICROSOFT RAM CARD is a trademark of Microsoft Consumer Products, a division of Microsoft, Inc.

The hardware in this product (the PC card and connector chassis assembly) is covered under the normal limited warranty of Rhodes/Chroma products.

Saturn 128K card is a trademark of Titan Technologies, Inc., P.O. Box 8050, Ann Arbor, MI 48107. Telephone (313) 662-8542.

Alphabyte 128K card is a trademark of Alphabyte Computer Products, 31304 Via Colina, West Lake Village, CA 91362. Telephone (213) 706-0333.

Notification that certain drum or rhythm machines are compatible with CBS' products should not be construed in any way to represent any endorsement, official or otherwise, by the respective rhythm machine manufacturers or by CBS INC. Revisions to the drum machines by the manufacturers may render them incompatible with this software and/or hardware.

CBS INC., its distributor, or its retailer is not responsible for any damage that may occur to your computer, its peripherals, or your drum machine, regardless of the cause or reason.

LinnDrum and LM1 Drum Computer are registered trademarks of Linn Electronics, Inc.

Drumulator is a registered trademark of E-mu Systems, Inc.

DMX is a registered trademark of Oberheim Electronics, Inc.

TR-606 Drumatix and TR-808 Rhythm Composer are registered trademarks of Roland Corporation.

NEITHER CBS INC. NOR APPLE COMPUTER, INC., MAKE ANY WARRANTY, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THIS SOFTWARE PACKAGE IS SOLD "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT CBS INC., CBS' DISTRIBUTOR, CBS' RETAILER, APPLE COMPUTER, INC., APPLE'S DISTRIBUTOR, OR APPLE'S RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF CBS INC. OR APPLE COMPUTER, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR

LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO, THE ABOVE MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGALS RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WITH VARY FROM STATE TO STATE.

DOS 3.3 is a copyrighted program of Apple Computer, Inc., licensed to CBS INC. to distribute for use only in combination with the Sequencer Program. Apple Computer, Inc. software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of the Sequencer Program. When the Sequencer Program has completed execution, Apple Computer, Inc. software shall not be used by any other program.

This manual is copyrighted. All rights reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior written consent from CBS INC.

The software supplied on diskettes for this product is also copyrighted. All rights reserved. The diskettes may not, in whole or part, be copied (except for the personal use of the registered owner) by any means without prior written consent from CBS INC.

ALL SPECIFICATIONS DETAILED IN THIS MANUAL ARE SUBJECT TO CHANGE AT ANY TIME WITH OR WITHOUT NOTIFICATION FROM CBS INC.

The Sequencer program and this manual were written December 10, 1982 by Tony Williams. A major revision occurred January 24, 1984.

The author would like to express gratitude to Ken Ypparila for writing the excellent Arpeggiator, Instruments and Parameter overlays included in this revision.

51 West 52 Street
New York, NY 10019

INSTALLATION OF THE SEQUENCER KIT

EQUIPMENT REQUIREMENTS:

To put the system together you need the following:

-one Chroma, a Chroma and Expander, or two Chromas.

-an Apple II Plus or IIe computer with 3.3 DOS with a 16K RAM Card (Microsoft RAM card or Apple Language Card or other compatible RAM card), one or two disk drives (although two drives is recommended), and a CRT monitor. The 16K RAM card is not required if you have an Apple IIe computer. Serious limitations arise from having only one drive (see Chapter 1). In future revisions, we will require two drives, so it is highly recommended that you start with two.

-a Chroma Computer Interface Kit, which consists of the printed circuit (PC) card, the interface cable, the phono jack box and mounting plate, footswitch, Sequencer Program diskette, Sequencer Data diskette, and the Sequencer Manual.

-options include another interface cable for controlling more than one Chroma and a control pedal for speed changing. A Chroma Computer Interface Manual is also available, which describes interfacing to the Chroma in general.

-also optional is an expanded RAM card, to be installed according to the card manufacturer's instructions.

HOW TO INSTALL IT:

The following installation steps should first be read completely, then performed in succession. If you need to install the interface PC card in a slot other than 5 see Appendix G before installation.

1. Turn the power OFF both the computer and the Chroma(s).

2. Open the top cover of the computer.

3. Take the connector box with the mounting plate loosely attached and slip it onto the back of the computer. Orient the box such that the cables slide in the second cutout from the left and the screws in the mounting plate are in the leftmost cutout. If you are using an Apple IIe, the cables cannot be slid into the cutout, they have to be threaded through a rectangular hole in the back of the computer. Otherwise, the mounting method is the same. Tighten the screws in the mounting plate firmly (you may want to remove your 16K RAM card from slot 0 to do this).

   CAUTION: Do not tighten the plate mounting screws unless the connector box is mounted to the rear of the computer. This may cause the screw to damage the wiring inside the connector box. FOR GROUNDING SAFETY, YOU MUST MOUNT THE CONNECTOR BOX TO THE COMPUTER CHASSIS.

4. Orient the Interface PC card as it will fit into SLOT 5 of the computer PC board. Before actually inserting it in the slot, connect the round phono cable assembly from the box onto its wafer pin mate on the Interface PC card. Match pin 1 on the cable assembly connector to the TOP pin of the wafer connector. When it is properly connected, the metal pins in the female connector will be visible from the component side of the interface PC board.

5. Next connect the ribbon cable from the Chroma port to the TOP 26 pin connector on the Interface PC card. The Chroma port cable is the leftmost ribbon coming out of the slot in the connector box. Then connect the Expander port ribbon cable to the BOTTOM 26 pin connector on the Interface PC card. Then finally, slowly insert the PC card in SLOT 5 of the computer's main PC board.

6. Since it is not recommended that you have unnecessary current drawn from the computer power supply, you should remove any peripheral cards that you will not need with this system. You will, of course, need the RAM card and the disk controller card. If you have an expansion RAM card, you should also install it at this time. See the manuals for each of these boards regarding specific installation procedures. See RECONFIGURE (Chapter 9) about configuring the software for any expansion RAM.

7. Replace the top cover of your computer.

8. Connect one end of the large black external computer cable to the COMPUTER INTERFACE connector on the Chroma. The other end of the cable should be connected to the CHROMA connector on the interface connector box. It does not matter which end goes to the Chroma. If you have an Expander, install the optional cable between the COMPUTER INTERFACE connector on the Expander and the EXPANDER connector on the interface connector box.

9. Plug the footswitch jack into the plug of the connector box marked FOOTSW and the control pedal jack into the plug marked PEDAL (if you have one). If you want to use the system without a footswitch see the RECONFIGURE section (Chapter 9) of this manual. This is important since the system will not play sequences unless the footswitch is plugged in or unless RECONFIGURE has been setup to ignore the footswitch.

10. Connect your own 1/4" phone audio cable between the CLICK jack on the connector box and a high impedance input to an amplifier or mixer. Input impedance to the amplifier should be at least 100K ohms.

11. Follow the power up procedure outlined below.

POWER UP PROCEDURE:

1. Turn on the Chroma (and Expander if used).

2. Insert the Sequencer Program diskette in Drive 1 of the computer. If you are using two drives, insert the Sequencer Data diskette in Drive 2.

3. After the Chroma Auto Tune LED flash stops, turn on the computer and the CRT monitor.

4. A message will appear indicating that the system is initializing. Once it has determined that there is a Chroma connected, it will print "Chroma is on line Chroma port..." and "Expander is on line at Expander port...". If the system has been configured for expansion RAM, a message will appear indicating how much RAM was installed.

5. If the Chroma is not responding, the computer will tell you so, in which case you should check the cabling, reset the Chroma (set-split 50) then type any key on the computer keyboard. If this does not fix the problem then power down the computer, reset the Chroma, then turn on the computer again. If you are still having problems, follow the steps outlined in this chapter from scratch. If all else fails call the Rhodes Service Department.

6. If you are powering up for the very first time, go to RECONFIGURE (Chapter 9) to change the number of drives, change the interface slot number, install the expanded RAM, or disconnect the footswitch.

THE MAIN MENU

This chapter describes the main menu. At the top is the Memory Usage, which is an approximate percentage of the total memory that is used by the sequence currently residing in RAM. If there is an inverse video space (square white spot) after "G-GET SEQ" then there is a sequence currently in RAM. When the system is first powered up the Memory Usage is 0% and there is no inverse space at GEQ SEQ because a sequence has not yet been loaded.

There are three pages of main menu selections. Each of the entries represent functions that can be selected by pressing the appropriate key on the APPLE keyboard. These are single key entry selections; in other words, a **<RET>** should not be depressed after the letter. Any of the selections on all three pages can be accessed no matter which page is being displayed. The ^ symbol means that the "CONTROL" or **<CTRL>** key must be held down at the same time as the regular key. For example to select the speed change mode (^T-SPEED CHANGE), you would hold the **<CTRL>** key down and press **T** on the APPLE keyboard.

Some of the selections from the main menu invoke other menus or questions from the Sequencer. Other selections are switch functions, so pressing their key toggles a condition. Several functions have no meaning if there is not a sequence in memory (such as P-PLAY SEQ) so pressing that key does nothing and returns to the main menu. If you press two keys, one right after the other (counting **CTRL** operations as one key), the sequencer will remember the second key and apply it as the next selection. This "type ahead" feature allows fast key entry but you must be a little more careful when you make a selection. Do not be afraid to experiment since anything that can destroy data directly from the main menu has second chance features built-in.

The first page of the menu has special significance because it contains important information regarding the status of the various functions. As mentioned before, the GET SEQ function has an inverse video space if there is a sequence in RAM. Another example is the inverse video section after "K-Click Track" which tells the user the status of the click track. The status of the function displayed on page one of the main menu is described in detail in the sections of this manual dealing with those particular functions. After any complete sequence of operations are performed, page one of the main menu will be displayed.

If the CRT screen blanks all of a sudden while on page one of the main menu, there is nothing wrong; the Sequencer is just blanking the display so CRT damage will not occur if the system is left unattended for an extended period of time. To restore the display, hit any key. This key will not be decoded as a menu selection. Screen blanking occurs only when page one (1) of the main menu is being displayed.

DESCRIPTION OF MENU SELECTIONS:

G - GET SEQUENCE from diskette.

S - SAVE SEQUENCE onto diskette.

P - PLAY SEQUENCE (plays unmuted tracks).

W - PLAY ALONG (same as PLAY SEQUENCE but reserves instrument 0 and 1 (if link) for the Chroma keyboard).

N - SET ENDPOINTS (measure) for PLAY SEQUENCE, RECORD TRACK, in or out of LOOP mode.

R - RECORD TRACK (records a track, two tracks if program has a link).

A - AUTO SAVE SWITCH (if ON then sequencer automatically saves the sequence, with a rename option, at the end of each RECORD TRACK operation).

T - TRACK DIRECTORY (displays recorded tracks and provides useful information such as the name of the track, the port to which it is assigned (Chroma or Expander), its voice program number, and the initial volume of the track).

M - MUTE or UNMUTE a track or all tracks.

L - LOOP SWITCH (if ON then sequencer loops between ENDPOINTS. This allows PLAY ALONG during RECORD TRACK mode until cued for recording).

K - set up CLICK TRACK.

B - EDIT (allows user to do simple editing on a track(s)).

I - REINITIALIZE the Apple, the Chroma and the Expander, if present. This is used if any communication problem arises.

D - DELETE a sequence from the diskette.

E - CHANGE PROGRAM (voice) of a track(s).

C - CLEAR sequence from Apple memory.

F - CATALOG (displays a list of all sequences and program groups on the diskette).

Z - ZAP A TRACK (deletes a track(s) from the sequence in Apple memory).

Q - PROGRAM FILE MANAGEMENT. This allows you to transfer program groups to and from the Chroma, Expander and Disk. It also allows deletion of program groups from disk.

V - CHANGE VOLUME of a track(s) overall.

X - RECONFIGURE. This allows the user to disconnect the footswitch, connect the SYNC INPUT, set up an external clock, change the slot locations of the Interface Card, or disable the emphasis on the first beat of the measure in the click track. It also allows the user to view the current status of the system.

H - CHROMA<>EXPANDER. Allows direct communication between the Chroma and the Expander ports.

Y - PRESSURE RECORD. Allows the Chroma port to send pressure information (works on REV 3+ Chromas with pressure sensor option only).

U - PARAMETERS. Displays the numerical value of the parmaters of the current Chroma program.

J - ARPEGGIATOR. Allows external clock syncing of arpeggiation patterns entered via the Chroma keyboard.

<RET> - Main Menu **<ESC>** - EXIT to Applesoft

2 - Menu Page Two 3 - Menu Page Three

^U - UTILITY BANK 1. Runs one of up to 10 user written program overlays (as an example we have included a comment program and an ECHO program).

^V - UTILITY BANK 2. Runs one of up to 10 more user written program overlays, if available (as an example we have included a complete hardware test program).

^X - DELETE all MEASURE commands. Note: This command is not displayed in the menu. It contains a second chance verification.

^Y - SCRUNCH the sequence (This reduces timing resolution of the sequence in memory and saves approximately 6% of the storage space).

^T - Selects SPEED CHANGE mode. Selecting this toggles between OFF, SMALL, and LARGE time variations. The large speed change mode varies the playback speed of the sequence from 1/2 speed to 2 times normal speed continuously with the pedal. The small speed change mode does the same except within a more restricted range.

^S - STORES the speed change as you here it if ON and in PLAY mode or OFF if in RECORD mode.

^A - TRANSPOSE sequence or track.

^N - RENAME a track(s).

^P - CHANGE PORT from Chroma to Expander or vice versa.

^I - INSTRUMENTS. Allows the user to play multiple instruments from the Chroma keyboard.

SEQUENCER MENU QUICK REFERENCE:

Page 1:

G - GET SEQ
R - RECORD TRACK
N - ENDPOINTS
A - AUTO SAVE SW
L - LOOP SWITCH
K - CLICK TRACK
2 - MENU PAGE TWO

S - SAVE SEQ
P - PLAY SEQ
W - PLAY ALONG
^T - SPEED CHANGE
^S - STORE SPEED

3 - MENU PAGE THREE

Page 2:

M - MUTE/UNMUTE
T - TRACK DIRECTORY
C - CLEAR SEQ
Z - ZAP TRACK
^N - RENAME TRACK
^Y - SCRUNCH
<RET> - MAIN MENU

F - DISK CATALOG
Q - PROGRAM FILES
B - EDIT SEQ
E - CHANGE PROGRAM
V - CHANGE VOLUME
^A - TRANSPOSE
3 - MENU PAGE THREE

Page 3:

D - DELETE SEQ
^P - CHANGE PORT
X - RECONFIGURE
J - ARPEGGIATOR
H - CHROMA <> EXPAND
<ESC> - EXIT TO FP
<RET> - MAIN MENU

^I - INSTRUMENTS
I - REINITIALIZE
^U - UTILITY BANK 1
U - PARAMETERS
^V - UTILITY BANK 2
Y - PRESSURE RECORD
2 - MENU PAGE TWO

PLAYING SEQUENCES

This section explains how to play one of the sample sequences provided with the system or one of your own sequences. It also goes into the details of looping and setting endpoints.

Before you load your first sequence it is a good idea to get familiar with the disk catalog selection. You should read about the disk catalog in Chapter 7 of this manual.

If you want to play using an external clock, read this chapter, then read Chapter 10 before proceeding.

HOW TO LOAD A SEQUENCE:

This section tells you how to load any one of the sample sequences included on the data diskette supplied with the Interface Kit. This same procedure is to be used to load YOUR sequences once you have saved them onto disk.

When you are in the main menu, look at 'G-GET SEQ'. If there is an inverse video space after this menu entry then there is already a sequence in memory and it will be written over (lost) if you load another. See Chapter 7, Disk Functions, to learn how to save a sequence if you want to keep this one.

After typing **G** for GET SEQ, the sequencer will ask you if you want to see the disk catalog. Type **Y** if you do, any other key if you don't. After the catalog routine is done, as described in Chapter 7, the Sequencer will ask you for the name of the sequence to be loaded. If you change your mind at this point, just type **<RET>** and you will get back to page one of the main menu without loading the sequence. Otherwise, type in the name of the sequence then **<RET>**. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was loaded and will display page one of the main menu.

PLAY / PLAY ALONG:

When playing a sequence you have a choice of "P-PLAY SEQ" or "W-PLAY ALONG". PLAY ALONG reserves instrument 0 for the Chroma keyboard, whereas PLAY SEQ defines the first track encountered as instrument 0. What this means is that under PLAY SEQ any lever changes, etc. that you do on the Chroma while the sequence is playing will be heard effecting the first track encountered and any notes you play on the Chroma keyboard will use up channels that the first track would normally have available for its notes. Also, if you press a voice selection switch on the Chroma, that first track will change voices. In other words, the first track and what you do on the Chroma "shares" instrument 0. Under PLAY ALONG what you do on the Chroma is reserved for just that and the first track uses another instrument. The "first track encountered" does not necessarily refer to track number 1. The concept of tracks, instruments and channels is further explained in Appendix H.

For now, just type **P** and you will hear the sequence that you have in RAM. If the sequence does not play and the Sequencer immediately returns to the main menu, it is likely that the footswitch is not plugged in the connector box (see Chapter 2 for proper installation).

If you are using the Sequencer in a live performance situation and want to start the sequence at an exact point in time, you should read Chapter 9, Reconfigure, regarding the use of the SYNC input with a footswitch.

NOTE: Some of the sample sequences contain tracks that are designated for output to the Expander port. If you play these sequences unaltered and there is no Expander on line, the Sequencer will tell you and automatically mute the track. To eliminate this problem, change the port to Chroma and unmute the track(s) as described in Chapter 6. Then save the modified sequence as described in Chapter 7.

STOPPING THE SEQUENCE:

You can let the sequence play to the end or type a space or depress the footswitch to stop. If the footswitch has been disconnected via RECONFIGURE, depressing it will have no effect and the sequence will continue to play. Regardless of the way the sequence is stopped, the main menu will reappear with the "LAST EVENT TIME: XXXXX" message above it. This is useful when editing a sequence so, for now, disregard it.

LOOPING:

What if you want the sequence to play over and over again? To do this type **L** to toggle the LOOP SWITCH. Now if you play the sequence it will loop continuously until you press the space bar or tap the footswitch. When you stop the sequence the LOOP SWITCH is automatically turned off. Whether the sequence loops in time depends on the way it was recorded and whether there was a click track when it was recorded. See the sections regarding RECORDING and the CLICK TRACK for more details about loop time.

ENDPOINTS:

What if you want to start the sequence at a point other than the beginning or end the sequence at a point other than the end? The Sequencer can use measures to define endpoints in the sequence. If the sequence was recorded with a click track, measures were automatically recorded in the sequence for reference. If the sequence you currently have in memory was recorded with a click track, the inverse video section after the selection "K-CLICK TRACK" on page one of the main menu will say "OFF 60BPM 4/4" or something similar. If it was recorded without a click track the inverse video section will say "NONE", in which case you cannot set up endpoints because there are no measures. If there was a click track, type '**N**-ENDPOINTS' to set up the beginning and ending endpoints.

When you select ENDPOINTS, the Sequencer displays the total number of measures stored during the recording (actually the number of the last measure stored) and the current endpoints. If the number of measures stored is zero then either the sequence was recorded without a click track or the person that recorded it deleted all the measures to save memory (see Appendix E, GETTING MORE NOTES OUT OF THE SYSTEM). If the current endpoints are "B-E" then the sequence would start playing at the beginning and end at the end, regardless of measures. To set the beginning endpoint type **B** then type the number of the measure you want to start at followed by a **<RET>**. To set the ending endpoint type **E** then type the number of the measure you want to stop at followed by a **<RET>**. If you try to change either endpoint to an illegal measure number (such as 0) then no change in the endpoints will occur. If you try to change to a number greater than the total measures stored, then the ending endpoints will be set to E. Type a **<RET>** to get back to the main menu after setting the endpoints.

PLAYING AND LOOPING BETWEEN ENDPOINTS:

When you PLAY SEQ now, the sequencer will play, starting at the beginning measure and stopping at the ending measure set by the endpoints. If you now select LOOP mode by toggling the loop switch, the sequencer will play between the endpoints continuously until stopped. The loop time may be increased by one measure due to the increased processing time required to find the beginning endpoint.

RECORDING SEQUENCES

This section will explain the operations required to record a clean slate (the first track) and subsequent tracks. If you want to record using an external clock, read this chapter then read Chapter 10 before proceeding.

Before recording the first track of a sequence it is essential that the RAM be cleared. Therefore, this section will start with an explanation of C-CLEAR SEQUENCE.

CLEARING A SEQUENCE:

Selecting this menu function initializes the Sequencer to begin recording. The Sequencer allows you to abort this command, as with all commands that will cause loss of valuable data. During this selection the RAM is cleared and the click track is turned off. The Memory Usage will obviously be zeroed and the G-GET SEQ menu entry will indicate that no sequence is in memory.

SETTING UP A CLICK TRACK:

The click track is not really a "track" in a normal sense but a hardware/software output pulse that is controlled by the timer and various conditions set by the user. The information supplied by the musician is used for many other timing related operations such as loop time determination, measure command storage and endpoint references. It is, therefore, highly recommended that the user start recording a sequence with a click track. To do so you should read chapter 8, THE CLICK TRACK in this manual.

AUTO SAVE:

Before you begin recording you should know about the AUTO SAVE function. If the AUTO SAVE SWITCH is ON, the Sequencer will enter the SAVE SEQ mode at the end of every record operation. This means that the Sequencer will display the current name of the sequence and ask if you want to rename it. If you want to abort the AUTO SAVE function, type **<ESC>** at this point (as in the case of the normal SAVE SEQ operation). If you do not type **Y** (or **<ESC>**) the Sequencer will save the sequence at the current name.

The AUTO SAVE function is useful for saving different versions of a sequence as you build it track by track. It is suggested that you turn the AUTO SAVE switch ON until you become familar with the Sequencer.

Presently, the AUTO SAVE function is OFF when powered-up. You can change this by modifying the main Sequencer program RECORD TRACK.1. To do this, while in the main menu type **<ESC>** to escape to FP BASIC. Then type **105 AS%=0** then type **<RET>**. Next, type **SAVE RECORD TRACK.1** then **<RET>**. Keep in mind that the modification should be done on a backup diskette, not the original and that any diskette write protect tab should be removed prior to attempting to save the program. Once the modification is performed and the altered program is saved, you should type **GOTO 160** then **<RET>** to get back to the Sequencer main menu.

RECORDING A CLEAN SLATE:

After the the RAM has been cleared, the click track set up and the AUTO SAVE switch is set at the desired state you are ready to record the first track.

Type **R** for RECORD TRACK and the Sequencer will ask you to select a program on the Chroma. Although you can change the program of a track very easily after it is recorded, it is a good idea to select a program that at least approximates what you will eventually want. This is because the sound of that program will influence your phrasing and other aspects of the music.

After you select a program, press any key to tell the computer you are ready. You can abort the record operation at this point by pressing **<ESC>**. Any other key will cause the Sequencer to continue in RECORD mode.

When recording from a clean slate, the name of the sequence will not exist. Just keep in mind that the SAVE SEQ function will require you to name the sequence before saving (see Chapter 7).

Also, the track will default to 'Tn' where n is the track number. You can always change it later by using the ^N-RENAME menu selection (more on this in Chapter 4, EDITING SEQUENCES).

After the above program selection is completed, the Sequencer will display how many tracks are being recorded and their track numbers and names. It will then tell you to:

```
PRESS ANY KEY OR THE FOOTSWITCH TO BEGIN RECORDING...
PRESS ANY KEY OR THE FOOTSWITCH TO END RECORDING...
```

When you press either the footswitch or any key on the computer keyboard, the inverse video RECORDING... message will be displayed. At this point the Sequencer is waiting for you to play. The track does not start and memory is not being used up until you start playing, move a performance control, or change the Chroma program.

Whenever a program is changed during the record operation (by pressing a program selection switch on the Chroma), you will create another track (two tracks if the program you select has a link). The Sequencer will detect this, set up the track and give it a name of 'Tn', as above.

The Sequencer can record up to 16 tracks but the Chroma can only play 8 tracks at a time. Changing programs while in the record mode is an easy way to concatenate tracks, insuring that there is no overlap. This allows you to use all 16 tracks even if you do not have an Expander or another Chroma.

When you are through recording the track(s), press any key on the computer keyboard or depress the footswitch. If the AUTO SAVE switch is ON you will enter the SAVE SEQ mode as decribed in Chapter 7. If you do <u>not</u> want to save the sequence, you should press **<ESC>** when asked if you want to rename the sequence.

RECORDING SUBSEQUENT TRACKS:

To record subsequent tracks you basically follow the same procedure as for a clean slate. Prior to recording these tracks, you may want to turn off the Click Track, so see Chapter 8, THE CLICK TRACK. You may want to adjust the volume or mute certain tracks to enable you to hear the beat more easily (see Chapter 6, EDITING SEQUENCES). You may want to send certain tracks to the Expander so you have more notes available on the Chroma while recording the next tracks, etc.

RECORDING AT SLOWER SPEEDS:

You may want to record at a slow speed to enable you to play difficult passages.

At Half Speed:

If you record at half speed, the recording will double in speed on playback. This is very easy to do using one of two methods. The first involves using the speed changing pedal, but this should be done after you are completely through with the sequence since the click track will no longer be in sync with the music if you do it this way.

The second way is much better. This involves the use of the RECONFIGURE selection described in Chapter 9. Basically, you want to set the TIMER INCREMENT to 1/2X before recording, then change it to 1X for playing. The final timer increment will be saved with the sequence and the click track will still be in sync. The only problem with this approach is that you can only change the timer increment in powers of two (1/16, 1/8, 1/4, 1/2, 1, 2, 4 and 8). As mentioned in Chapter 9, this feature was designed mainly to scale the clock frequency when converting between internal and external clocks.

At Other Speeds:

You can record at any reduced speed using the SPEED CHANGE selection from the main menu, as long as you are willing to live without a click track. In record mode, the speed variation will be stored if the STORE SPEED switch is OFF. The reason for this is that the Sequencer determines the speed stored as the ratio of a variable to a fixed increment timer. You do not have to understand this to make the thing work. Just remember that the STORE SPEED switch works backwards in record mode.

RECORDING WITH PRESSURE:

Since pressure recording requires a high data rate and uses a lot of memory, we have separated the record mode into two types: recording with pressure and recording without pressure.

To record with pressure, select Y for PRESSURE RECORD and proceed as normal. If your Chroma does not have the pressure sensor option, or it does have the option but you want to save note memory, you should use the normal record mode.

LOOP RECORDING:

This method of recording essentially allows you to play along in loop mode, then easily tell the Sequencer when you have the passage practiced enough to record it. Once the Sequencer has been "cued", the next loop automatically switches it to record mode.

First, turn ON the loop switch by pressing L, then go through the normal record procedures. When you press the space bar or footswitch to RECORD, the Sequencer will display "LOOPING...". Keep playing along with the sequence until you think you can record, then press the footswitch. The Sequencer will then display "CUED...". The next time the sequence loops, you will be in record mode and the Sequencer will display "RECORDING...". The sequence will stop and the record mode will be terminated when you press the footswitch again, press the space bar or the end of the sequence is reached.

EDITING SEQUENCES

This section deals with the many ways to edit your sequence once you have recorded it. Editing functions fall into three catagories: miscellaneous track directory functions, speed changing and the Editor.

MISCELLANEOUS TRACK DIRECTORY FUNCTIONS:

These functions are grouped together because they all involve using and/or modifying the sequence track directory.

Track Directory:

The track directory is a list of all the tracks that make up a sequence. Included in each track entry is the following vital information:

-track number- This is the number to use whenever you want to refer to a particular track.

-track name- This is currently for your use only. The Sequencer does not refer to this name except to check and see that it is unique among tracks in this sequence. The name follows the syntax rules outlined in APPENDIX P.

-track port- This is 'C' if the track is to be outputted to the Chroma port or 'X' if the track is to be outputted to the Expander port.

-track program number- This is the Chroma/Expander Program number for this track.

-track volume- This is the initial linear volume value for the track.

Change Port:

This routine begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will toggle that track's port and redisplay the track directory. If the track was destined for the Chroma, it will now be set for the Expander, whether there is an Expander on line or not. During playback, if there is no Expander, the Sequencer will tell you and mute that track. If this happens, change the port back to Chroma and unmute the track as described later in this chapter (under MUTE/UNMUTE). To exit the change port mode, press **<RET>** when the Sequencer asks you for the track number.

Change Program:

This routine also begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new program number. If you change your mind and do not want to change this track, press **<RET>** and the sequencer will redisplay the track directory and ask for another track number to change. If you input the new program number, the Sequencer will change that track's program and redisplay the track directory. To exit the change program mode, press **<RET>** when the Sequencer asks you for the track number.

Change Volume:

This routine also begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new initial volume value. If you change your mind and do not want to change this track, press **<RET>** and the Sequencer will redisplay the track directory and ask you for another track number. If you input the new initial volume value, the Sequencer will change the track volume and redisplay the track directory. To exit the change volume mode, press **<RET>** when the Sequencer asks for the track number.

The volume value is linear, as mentioned above in the description of the track directory. The volume of the track may be varied at any time from the initial value by use of a pedal controlling the volume parameter of a program in the Chroma during recording or by inserting volume commands via the Editor. Volume commands can also come from the Chroma when you are recording a linked program (2 tracks at once) and you change the link balance (see APPENDIX K for a description of a small software bug in the Chroma). If either of these is the case, then changing the initial volume of the track will only effect the track up until that volume command is encountered. In other words, the volume values, initial or otherwise, are absolute, not relative.

Zap A Track:

This routine will delete a track from the sequence. It begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will delete all commands that belong to that track then redisplay the track directory. To exit zap mode, press **<RET>** when the Sequencer asks you for the track number to delete.

If you zap the last track that was recorded, then that track number will be available for subsequent recording operations. If you zap a track other than the last one, that track number is not available for subsequent recordings. In other words, you effectively lose one of your 16 tracks and the tracks numbers will not be sequential.

NOTE: If you zap Track 1 and the sequence is externally synced, you will lose some of the syncing information (actually, you delete the TIME 0 command). To restore sync, run the User Utility Bank1, number 2 program by typing **^U** then **2<RET>**.

Rename A Track:

This routine allows you to change the name of a track. It also begins by displaying the track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new name. The name must follow the syntax rules as outlined in APPENDIX P, but don't worry, the Sequencer will not let you input anything that is illegal. If you change your mind and do not want to change the name of this track , press **<RET>** at this time. If you input the new name, the Sequencer will change the name of the track and redisplay the track directory. To exit the change name mode, press **<RET>** when the Sequencer asks you for the track number.

Transpose:

This routine allows you to transpose the whole sequence or any one track in semitones. Maximum allowable transposition is 33 semitones at any one time, up or down. The Sequencer currently does not check for cumulative transpositions, so care must be taken that the total amount of transposition does not exceed 33 semitones, else the Chroma will get confused and play the wrong notes. Unfortunately, after the limit has been passed, you cannot recover the sequence by transposing an equal amount in the opposite direction.

The routine begins by asking you if you want to transpose all tracks. If you do, type **A** for all. If you want to transpose one track, then type **<RET>**.

If you type **<RET>** for single track transposition, the Sequencer will display the track directory and prompt you for the track number. If you type **<RET>** instead of the track number at this point you will abort the transposition and exit back to the main menu. If you type a track number, the Sequencer will then ask you the amount to transpose. Type **<RET>** at this point to abort the transposition and exit back to the main menu, otherwise type in a semitone value preceeded by a + or - for direction. If there is no direction symbol, the Sequencer will assume + direction. If you input an invalid value, the Sequencer will tell you and ask you to reenter the direction and transposition amount. After the transposition is complete, the Sequencer will return to the main menu.

If you type **A** to transpose all, as presented above, the Sequencer will ask you for the direction and amount of transposition. Enter this value as if you were transposing only one track as outlined above. The Sequencer will transpose all tracks and return to the main menu.

Mute/Unmute:

This routine allows you to mute and unmute tracks. A muted track retains all of its information, but will not be heard during playback. MUTE/UNMUTE has its own menu that allows you to view the track directory, mute a track, unmute a track, or unmute all tracks. A track that is muted will be shown in INVERSE VIDEO in the track directory (and the EDITOR) and will not be outputted to it's port. To exit the Mute/Unmute mode, press **<RET>** when the mute menu is displayed.

A track may be automatically muted if it is designated for the Expander port and no Expander is on line. It will also be automatically muted if you try to output more than 8 tracks (7 tracks if recording a non-linked program, 6 tracks if recording a linked program) to the Chroma port and the AUTO-REROUTE flag is OFF or there is no Expander on line. The muting ocurrs when the Sequencer encounters the start of a track that puts it over the limit. A message will be displayed and the Sequencer will continue to play/record. To permanently unmute a track that the Sequencer automatically mutes, you must correct the condition that caused this to happen. In other words, you must ZAP the track, ZAP another track, or change the port of the track.

SPEED CHANGING:

You can change the speed of your sequence three different ways:

(1) By using a variable external clock (see Chapter 10).

(2) By changing the time increment by powers of two (see Chapter 9, RECONFIGURE).

or (3) By using the main menu selections ^T - SPEED CHANGE and ^S - STORE SPEED, which are described in this section of the manual.

Before we begin, make sure you have the optional control pedal connected to the PEDAL input of the connector box on the back of your computer. You can use the Chroma's pedal, purchase one from us, or use any 100K linear taper potentiometer (it doesn't have to be a pedal).

The analog-to-digital converter (ADC) IC on the Interface PC Card will convert the resistance of your pedal into a digital value that the Sequencer will use to change the speed of the sequence. If you suspect that the range of the ADC is out of adjustment, run the ADC Test Program in USER UTILITY BANK 2, number 1 or the Interface Test Program in USER UTILITY BANK 2, number 0 (the command is ^A for ADC test in the Interface Test Program). Number 1 displays the value in decimal (0-255) while Number 0 displays it in HEX (00-FF). When you move the pedal to its extremes, the value displayed should go from 0 (pedal all the way up) to 255 or FF (pedal all the way down). If this is not the case, then you should adjust the ADC according to the procedure outlined in APPENDIX O, Hardware Description and Specifications.

The speed change selection ^T is a three way toggle switch, which is either OFF, SM (small variation), or LRG (large variation). The large variation allows you to increase the speed up to twice normal or decrease it down to one-half normal. The small variation range is approximately one fourth of the large range. When either the small or large speed variation has been selected, the Sequencer will play back at a variable speed, depending on the position of the pedal.

In play mode, the speed change is not permanent unless the SPEED STORE switch is ON. In record mode, the speed change is not permanent unless the the SPEED STORE switch is OFF. This switch is a normal two position toggle switch. In the on position, the Sequencer changes the time values in the sequence to correspond to the variations in pedal position. When the switch is on, the sequence should be allowed to play all the way through, otherwise you will get timing glitches, or sudden changes in tempo, that are almost impossible to remove. If you want a sudden change in tempo, it is relatively easy to just move the pedal quickly.

A neat effect is to loop a small sequence with the SPEED CHANGE on large or small variation and the STORE SPEED on. Everytime the sequence loops, it is faster.

After the speed of a sequence has been permanently changed by STORE SPEED, the click will never again be in sync with the sequence. You should, therefore, change the speed of the sequence only after you no longer need the click track. The measure commands will be correct at the faster or slower speed because time values are stored with them, unlike the click track which is not really a track at all.

WARNING: Always turn OFF the SPEED STORE switch immediately after you permanently change the speed of the sequence. It does not automatically turn OFF. If you immediately play the new sequence to hear how it came out and the STORE SPEED switch is still ON, you will change the speed again.

THE EDITOR:

A sequence is organized as a list of musical events. Each event has a track number associated with it. The events for all tracks are, therefore, interleaved in the sequence. The events are represented by 'commands' (such as ATTACK, RELEASE, etc.) that can be altered by the Editor.

A track always begins with a DEFINE command and always ends with an UNDEFINE command. Between these two commands, Chroma 'channels' are allocated for that track (as detailed in APPENDIX H).

The Editor operates by displaying the next command entry from the EDIT TRACK LIST (list of tracks that you are currently editing) and prompts you for instruction by a ":" on the next screen line. Command entries from muted tracks will be displayed in inverse video. If you do not want to change this command entry, just press the space bar (NEXT) and that command will be restored in the sequence. If the MONITOR is on and the track is not muted, then the command entry will also be outputted to its port. The next command entry will then be displayed. The sequence cannot go backwards. The endpoints are set at the beginning and ending of the sequence while in the Editor, regardless of previous endpoint settings. The endpoints will be restored upon EXIT from the Editor.

You can select from any of the instruction commands displayed at the top of the screen when the Sequencer is prompting you with a ":" symbol. Following is a detailed explanation of each instruction command:

^M or **<RET>** - MONITOR

MONITOR on/off (allows you to hear what you are editing). If the MONITOR is on and the track is not muted, any command that is restored is also outputted to its port.

^C - CHANGE

CHANGE currently displayed command entry. The Editor will redisplay the current command entry and position the cursor at the beginning of the command. At this point you can type in the full command entry as described below in INSERT or you can use the left and right arrow key to edit the line. Once **<RET>** is pressed, the Sequencer will erase the rest of the command entry and display the edited line, asking you if it is correct. If you type **Y** or **<RET>**, the Editor will restore and output the command entry, then display the next one. If you type **N** then the Editor will ask you to re-enter the command. You must type in the full command entry as described below in INSERT. Once this is done, the Editor will again display the edited line and ask for verification.

NOTE: The Editor will not assemble inverse video text, in other words, if you want to change a command entry that is muted, you must type in the full command as described in INSERT.

WARNING: Do not change a command entry to a PRESSURE command if your Chroma and/or Expander is not Interface Software REV 3+. This will confuse the Editor and you will LOSE YOUR SEQUENCE. You can change to pressure commands, even if you do not have the pressure sensor option, as long as the software in your Chroma is REV 3+.

^I - INSERT

INSERT command before currently displayed command entry. Type in the full command entry as described below under EDITOR COMMAND ENTRY SYNTAX. Alternately, you can type in the shorthand version containing only the main command opcode followed by the vital numbers, separated by a space. The Editor will ask for verification as described above in CHANGE then store and output the command to its port (if not muted).

WARNING: Do not INSERT a PRESSURE command if your Chroma and/or Expander is not Interface Software REV 3+. This will confuse the Editor and you will LOSE YOUR SEQUENCE. You can INSERT pressure commands, even if you do not have the pressure sensor option, as long as the software in your Chroma is REV 3+.

^D - DELETE

DELETE currently displayed command entry. The Editor will display the current command and ask for verification. If you type **Y** or **<RET>** the Editor will not restore or output this command entry and will display the next one in line.

^F - FIND

FIND command entry. The Editor will display the FIND: prompt at which point you have the following options:

(1) **<RET>** finds and displays the last command entry you asked the Editor to find.

(2) Typing a command entry as described above in INSERT then **<RET>** finds the command and displays it. Any section of the command left out is "wild", in other words the Editor doesn't care. For example, FIND: ATTACK 1 -20 will find the next ATTACK in TRACK 1 that has a key number of -20, regardless of the VELOCITY or PRESSURE.

NOTE: If the command entry is not found, then END OF SEQUENCE will be displayed. Type a space bar (NEXT) or **^H** (HOME) to get to the BEG OF SEQUENCE before trying again.

NOTE: The Editor can find time values or measure numbers that span a "distance" of up to 32768 relative to the next value. You must, therefore, make an intermediate "stop along the way" when finding values greater than 32768 higher than the next value. EXAMPLE: If the next time value is 2367 and you want to find TIME 44324, you should first find TIME 32000, then find TIME 44324.

NOTE: The find function only searches tracks that are in the EDIT TRACK LIST.

<SP> or **<SPACE>** - NEXT

Disassemble and display NEXT command entry. This instruction will cause the current command entry to be restored in the sequence. If the MONITOR is ON, the command will also be outputted to its port. The next command is then disassembled and displayed.

^X - EXIT

EXIT the Editor. The Editor restores the ENDPOINTS and exits back into page one of the main menu. The EDIT TRACK LIST is left intact.

^E - EDIT TRACK

Add a track to the EDIT TRACK LIST. Type the TRACK number followed by a **<RET>**. If that track does not exist, the Editor will tell you that the entry is invalid and ask for another track number. Typing **<RET>** without a track number aborts this mode.

^R - REMOVE

REMOVE a track from the EDIT TRACK LIST. Same as above, but removes a track from the list instead of adding it.

^S - STATUS

Display the current STATUS of the EDIT TRACK LIST. The EDIT TRACK LIST is a list of all tracks that are currently being edited. If a track is not being edited, the Editor will not display or output commands in that track. Also, you cannot INSERT or FIND command events in that track.

^H - HOME

HOME returns to the beginning of the sequence.

NOTE: If an error occurs, the current instruction is aborted and an error message will be displayed. Following is a list of Editor syntax errors:

12 - Unrecognized command in disassemble routine.
13 - Syntax error in entering command in the Editor.
14 - User attempts to change DEFINE command in Editor. This is currently illegal.
15 - User attempts to delete BEG OF SEQ (BOS) or END OF SEQ (EOS) command, insert before BOS/EOS command, or change BOS/EOS command. All of these operations are currently illegal.
16 - Track input out of range or not in EDIT TRACK LIST.

Editor Command Entry Syntax:

The following is a list of the command entries that are allowed in the Editor and their correct syntax. An alternate shorthand syntax is allowed in most cases which consists of the command opcode (**BOLD** capitalized) followed by the parameters (numbers only), separated by a space.

Operations on DEFINE commands are limited to display only. Deleting or changing a BEG OF SEQUENCE or END OF SEQUENCE command is not allowed. Also, inserting a command before BEG OF SEQUENCE or END OF SEQUENCE commands is not allowed.

NOTE: Lever values are 0 at rest and pedal values are 0 at 'heel' or 255 at 'toe' position.

BEG OF SEQUENCE

END OF SEQUENCE

TIME XXXXX where XXXXX is from 0 to 65535

MEASURE XXX where XXX is from 1 to 65535

```
DEFINE TRACK#tt LEVER1=aaa LEVER2=bbb PEDAL1=ccc
     PEDAL2=ddd VOLUME=eee FOOTSWITCH=fff
               where tt is from 1 to 16
                      aaa and bbb are from -128 to 127
                      ccc, ddd, and eee are from 0 to 255
                      fff is 0 if both footswitches up
                           128 if LATCH down, SUSTAIN up
                            64 if LATCH up, SUSTAIN down
                           192 if both footswitches down
```

UNDEFINE TRACK#tt where tt is from 1 to 16

```
ATTACK TRACK#tt KEY#kkknn VELOCITY=vvv PRESSURE=ppp
                where tt is from 1 to 16
                      kkk is from -64 to 63
                      nn is the musical note (this is
                         not required when inputting,
                         it is for your reference only)
                      vvv is from 0 to 31
                      ppp is from 0 to 63

RELEASE TRACK#tt KEY#kkknn VELOCITY
                where tt is from 1 to 16
                      kkk is from -64 to 63
                      nn is the musical note
                      vvv is from 0 to 31

VOLUME TRACK#tt VALUE=vvv
                where tt is from 1 to 16
                      vvv is from 0 to 255

LEVER1 TRACK#tt VALUE=vvv
                where tt is from 1 to 16
                      vvv  is from -128 to 127

LEVER2 TRACK#tt VALUE=vvv   same as above

PEDAL1 TRACK#tt VALUE=vvv
                where tt is from 1 to 16
                      vvv is from 0 to 255

PEDAL2 TRACK#tt VALUE=vvv   same as above

SUSTAIN UP TRACK#tt         where tt is from 1 to 16

SUSTAIN DOWN TRACK#tt       same as above

LATCH UP TRACK#tt           same as above

LATCH DOWN TRACK#tt         same as above

PRESSURE TRACK#tt KEY#kkknn VALUE=pp
                where tt is from 1 to 16
                      kkk is from -64 to 63
                      nn is musical note
                      pp is from 0 to 63

SET PARAMETER TRACK#tt PARAMETER#ppp VALUE=vvv
                where tt is from 1 to 16
                      ppp is from 1 to 100
                      vvv  range depends on parameter
                             (see Chroma Performance
                             Manual)
```

Editing Tips:

When trying to find a note, it is advantageous to play the sequence and notice the LAST EVENT TIME: XXXXX message, then go into the Editor and find that time value. You must stop the playing of the sequence just prior to the note you are looking for.

When trying to add or subtract a time offset to every time command after a particular point in the sequence, use FIND:TIME then **<RET>**. After the first time value has been changed, use FIND: **<RET>** and the Editor will find the next time value and display it for changing.

DISK FUNCTIONS

This section describes the Program File management, Sequencer catalog and saving, loading and deleting sequences.

PROGRAM FILE MANAGEMENT:

Typing **Q** from the Main Menu displays a Program File Management Menu that allows you to transfer program data to and from the Chroma port and disk or from the Expander port and disk. Direct transfer of Program 0 between the Chroma and Expander port can be accomplished via the **H**-Chroma<>Expander main menu selection.

A Program Group is defined as a group of one or more 59 byte data blocks that make up the program. At present, the Sequencer only supports transfer of program groups containing 50 programs. The Program File Management Menu choices are as follows:

1-Chroma programs to disk
2-Disk programs to Chroma
3-Delete disk program group

And, if an Expander is on line,
4-Expander programs to disk
5-Disk programs to Expander

The delete menu selection (3) provides the user a second chance since valuable data can be lost. The disk catalog can easily be accessed when transferring from disk (2,5) and when deleting a file (3). In Program File Management mode, the catalog displays only program group files.

Transfer of program groups between the Chroma and Expander can be performed indirectly by saving the Chroma programs onto disk then loading them into the Expander or vice versa.

The names of program groups that were used to record sequences can be stored in the sequence file using the Comment program (User Utility Bank 1, number 0, see APPENDIX C). The names of program groups follow the syntax rules for all names outlined in APPENDIX P.

The actual structure of the program group file is detailed in APPENDIX B.

DISK CATALOG:

When the system is "up" and the main menu is being displayed, type **F** for DISK CATALOG and the catalog will appear. The catalog displays the list of program groups and sequences that you have on the disk in Drive 2 if using a two drive system or Drive 1 if using a single drive system. If there are other files on the disk (as will definitely be the case in a single drive system) the disk catalog simply does not display them.

At the top of the catalog on the left is the Volume number. This is useful for keeping track of several diskettes. The diskette can be assigned a volume number when you initialize it for sequence storage. Read your APPLE manuals and APPENDIX M to learn how to do this. The Sequencer pays no attention to volume numbers, it just relays that information for your use.

The next item at the top of the disk catalog is the number of free sectors left on the diskette. This indirectly tells you how many more sequences and program groups you can store on this disk. A blank 3.3 DOS diskette has available 494 sectors for sequence and program group storage. A single drive system has considerably less (approximately 190 sectors) since the Sequencer operating system must reside in Drive 1.

Each entry of the catalog also contains useful information. On the extreme left is the File Type. 'P' means that this entry is a Program Group, which is a group of 50 programs much like a "fast dump" section of a cassettte for the Chroma. 'S' means that this entry is a Sequence. The next item in the catalog entry is the number of sectors that this file occupies on the diskette. This information, in addition to the free sector amount at the top of the catalog, can be used to determine how many more sequences or program groups you can store. Program groups consisting of 50 programs always take 18 sectors. The next and last item in the entry is the file name which is used for loading, saving and deleting the files.

After the number of entries reaches the maximum allowed on the screen at any one time, the Sequencer asks you to type **<SPACE>** to continue. If you type an **<ESC>** at this point, you will abort the catalog. Once you get to the end of the catalog, typing any key to return you to the main menu.

DELETING A SEQUENCE:

To delete a sequence from disk, select **D** from the main menu. The Sequencer will then ask if you want to see the disk catalog. Type **Y** if you do, any other key if you don't. The catalog will display only sequences this time, since you do not care about program group files.

If the catalog takes more than one page, it will ask you to type a **<SPACE>** to continue or an **<ESC>** to abort at the current page. Once the catalog has been displayed, the Sequencer will ask for the name of the sequence to delete. You can abort and return to page one of the main menu at this time by typing **<RET>**. Otherwise, carefully type the name of the sequence then **<RET>**. At this point the Sequencer will warn you that you are about to delete data from the disk and ask for verification. Type **Y** to continue DELETE or any other key to abort. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was deleted and will display page one of the main menu.

WARNING: Neither Apple DOS nor the Sequencer checks file types when deleting files. You can, therefore, inadvertently delete a Program Group file from the DELETE a sequence menu selection or vice versa. It is important to never name a sequence the same as a Program Group. The Sequencer will create two separate files of the same name as long as the files types are different. When deleting a file, however, the Sequencer will delete the first file it finds in the Catalog with that name, regardless of the file type. Also, if you are operating a single drive system, it is imperative that you never use the names HELLO, RECORD TRACK.1, CHROMA.BEGIN, or any name ending in .HEX for sequences or program groups.

SAVE A SEQUENCE:

To save a sequence, type **S.** The Sequencer will display the name of the sequence and ask if you want to rename it. If the sequence has no name the Sequencer will disply the name as "". If you change your mind about saving the sequence, type **<ESC>** at this time and you will get back to the main menu. If you do not want to change the name of the sequence, type any key except **Y.** The sequence must be named before it can be saved.

Type **Y** if you do want to change the name before saving. The Sequencer will then ask you to type in the new name. If you type **<RET>** at this point, the sequence will be saved with the old name (in other words, renaming is aborted). Legal sequence names are those that follow the name syntax rules of APPENDIX P. You should read the WARNING in the DELETING A SEQUENCE section above regarding file names.

When the sequence is saved, various things are recorded within the sequence file that reflect the system status, such as the click track setting, instrument ID's revision numbers and the timer source and increment, etc. All of this is detailed in APPENDIX B.

NOTE: Entering SAVE SEQ mode, then typing **<ESC>** at renaming is a good way to just find out the name of the current sequence.

GET A SEQUENCE:

Loading a sequence from disk will erase any existing sequence, so you may want to save the old one as described in this chapter. To get a sequence, select **G** from the main menu. The Sequencer will then ask if you want to see the disk catalog. Type **Y** if you do, any other key if don't. The sequencer catalog will display only sequence files. After the catalog is displayed, type any key to continue. If an **<ESC>** is typed, the catalog will be stopped at the current 'page'. The Sequencer will then ask you to type in the name of the sequence followed by a **<RET>**. You can abort and return to the main menu at this point by typing **<RET>** instead of a file name. Otherwise, type in the name of the sequence followed by **<RET>**. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was loaded and will return to the main menu.

There will be an inverse video space after the **G** - GET SEQUENCE menu selection and the MEMORY USAGE will display the percentage of total RAM occupied.

Loading a sequence will also set the Click Track, Timer Source, and Timer Increment that was set-up at the time the sequence was last saved.

THE CLICK TRACK

The click track is really not a track at all. It is a hardware filtered pulse from the Interface PC board, which is controlled by the Sequencer software. This chapter describes how the click track is generated and used in the Sequencer, how to set it up and what restrictions apply to changing it once set up. This chapter also explains time signatures, MEASURE commands and how the Sequencer determines the loop time of a sequence.

HOW IT IS GENERATED:

The Sequencer uses the BPM value input by you, among other things, in determining when to toggle a flip/flop on the Interface PC board. To emphasize the first beat of the measure, the Sequencer reduces the energy of subsequent beats by toggling the flip/flop twice (approximately 28 microseconds apart).

NOTE : The click output hardware is high impedance (appr. 10K). If you do not use an amplifier or mixer input that has an input impedance of at least 100K ohms, you may hear differences between each measure. You may even not be able to hear the emphasis on the first beat of the measure. If you must use that particular amplifier input, an authorized Rhodes Service Center may be able to change the output impedance of the Interface PC board slightly to accomodate you. Call the Rhodes Chroma Service Department for details.

THE NEED FOR A CLICK TRACK:

Besides the obvious reason of helping you to keep the beat, the click track is used to determine measures in the sequence and to calculate the loop time. Loop time is calculated for you, as described below, as long as there is a click track that has been used for at least the first track record operation.

The click track may also be used in the recording studio to provide a stable audio reference for other instruments you may want to record. The Sequencer cannot "sync" off a pre-recorded version of its click track because the frequency is way too low. See Chapter 10 for syncing details.

TIME SIGNATURE

Time signatures in the click track can be anything from 1/1 to 15/15, although the musical value of some of these are questionable. The lower part of the signature (which type of note gets a beat) is not used by the Sequencer; but it is included for your information. The Sequencer will not syncopate any time signature, but the first beat of the measure can be emphasized.

Time signatures are required if you set up a click track. They are used to tell the Sequencer when a measure boundary has ocurred and when to loop back to the beginning of the sequence when you are in loop mode.

HOW TO SET IT UP:

If a click track is to be used in a sequence, it MUST be set up prior to recording the first track. To set it up, select **K** from the main menu. After the overlay is loaded, the Sequencer responds by asking for the time signature. Anything from 1/1 to 15/15 is allowed. The Sequencer, however, doesn't use the lower portion (which type of note gets a beat). Enter the signature as you would write it on paper, i.e. **4/4** then **<RET>**. Typing **<RET>** instead of a time signature defaults to the current signature. When the Sequencer is turned on, the click track is set up at 60 BPM and 4/4 even though the indicator on page one of the main menu says NONE.

After you select a time signature, the Sequencer will display a small menu that looks something like this:

```
CURRENT SPEED IS 60 BPM

                      METRO MENU

        XXX - INPUT BPM (34-234) THEN <RET>

               P - PLAY CLICK TRACK

                 <ESC> - MAIN MENU

PLEASE SELECT...
```

The first selection (XXX) allows you to just type the BPM and press **<RET>**. If you type **P** to play the click track, the Sequencer will start to play the click as you have set it up so far and will display the following menu:

```
              U - SPEED UP

              D - SLOW DOWN

          <ESC> - METRO MENU


PLEASE SELECT...
```

Typing **U** will speed up the click track slightly. Use the **<REPT>** key at the same time as **U** and the click track will speed up continuously. The same goes for **D** to slow the click track down. When you are satisfied with the speed, type **<ESC>** to exit back to the METRO MENU. You will now see the same menu as before, except notice that the BPM value has been changed to the current value. Type **<ESC>** again to return to the main menu. Notice that the inverse video field after the menu selection **K** - CLICK TRACK indicates that the click track is ON and that the time signature and speed have been correctly set.

NOTE : The speed of the click track is effected by the TIMER SOURCE, TIMER INCREMENT and the SPEED CHANGE pedal position (if ON). The perceived speed of the click track may not, therefore, be equal to the BPM shown. The BPM indicators are correct only if the Sequencer is driven by an INTERNAL clock (or an EXTERNAL clock of exactly 1000 Hz) with a TIME INCREMENT of 1X and the SPEED CHANGE switch OFF.

NOTE : If the SPEED UP or SLOW DOWN selections are used to set up the speed, the BPM display will round the BPM to the nearest unit. The actual speed, however, will be what you hear.

RESTRICTIONS ON CHANGING THE CLICK TRACK:

When the Sequencer is powered up or the sequence is cleared, the BPM is set at 60 and the time signature is set at 4/4. Other than these default values being set up, the Sequencer acts just as if there is no click track (NONE).

When the **^X** - DELETE ALL MEASURES selection is made from the main menu, the Sequencer erases all measure commands (described below) as well as all traces of a click track. This sets up the click track as if you had cleared the sequence (NONE).

Once the first track of a sequence has been recorded, the only thing you can do to a click track is turn it ON and OFF. There is a difference between OFF and NONE. NONE means that there was never a click track and the Sequencer can not store measure commands and may not loop in perfect time. This is not true if the click track is merely OFF, even if you extend the endpoints of sequences during record operations.

To turn a click track ON or OFF once a track has been recorded, just select **K** from the main menu. When the Sequencer asks if you want a click track, press **Y** or **N**.

When a sequence is loaded from disk, the click track is set exactly the way it was when the sequence was last saved.

MEASURE COMMANDS:

Measure commands are stored by the Sequencer right at the first beat of the measure, starting with MEASURE 1. These commands are used as ENDPOINT references and as markers in the Editor.

MEASURE commands are stored while recording subsequent tracks only if you extend the length of the sequence. This will occur whether the click track is ON or OFF, but will never occur if the click track is NONE.

If you play the downbeat slightly before the first beat of the measure, that note will logically fall into the previous measure. When this happens and you reference the measure for an ENDPOINT, you may not hear the down beat. If this occurs on the first note of a sequence, it will also effect the loop time, as explained below. This problem can be remedied by moving the ATTACK command right after the MEASURE command via the Editor.

LOOP TIME:

The Sequencer will correctly determine the loop time if the first track of a sequence was recorded with a click track and you follow a few simple rules when beginning and terminating all track record operations. If these rules are not followed and/or you reference ENDPOINTS other than the actual beginning and ending of the sequence, the Sequencer may have to add a measure's worth of time to the loop. The end result will be a full measure rest between loops. Looping with a slow EXTERNAL clock requires a little more effort as described in Chapter 10.

When you record the first track, the Sequencer takes note of the time the first event occurs relative to the click pulse time and the beat of the measure. This information is saved with the sequence and is not updated unless you change it by recording a note on a subsequent track before the first event of the first track. By the way, the only way you can do this is by recording subsequent tracks with the click track ON, since it is only under this condition that the Sequencer starts the recording mode one measure before the start of the sequence.

When the last event is played in the sequence, the Sequencer loops back around and presets the timer to a value that will cause the first note to be played in time. The last event in this case is not a RELEASE but when you press the space bar or footswitch to terminate the record operation.

The main rule regarding loop time is that you must allow enough time for the Sequencer to do its job. The last event in a sequence must occur such that there is sufficient time left in the measure to play the first note on time. Sometimes this requires the use of the Editor to delay the first event time or pull back the last event time.

Most of the time, it is easy to follow these simple rules while you are recording. If you need to touch it up, however, MEASURE commands stored in the sequence are a good indication of where these events must occur in time. In general, if you want to start the sequence on the down beat, the first event must occur slightly after the down beat and the last event must occur slightly before the down beat of the next measure. Give the Sequencer a few TIME INCREMENTS between each end to accomplish its task.

Again, if you do not follow these rules, the worst that can happen is that the loop time will be increased by a measure's worth of rest.

If a click track is not used, the only thing determining loop time is the first event and when you press the footswitch or space bar at the end of the record operation. There will be a slight processing delay for which you will have to learn to compensate. This delay will be constant for a given length of sequence. The Sequencer will take care of this for you only if you use a click track.

When using an EXTERNAL clock, loop time requires special considerations that are explained in Chapter 10.

RECONFIGURE

The **X** - RECONFIGURE selection from the main menu allows you to change certain operating parameters of the Sequencer. These paramenters involve the timer, SYNC input, FOOTSWITCH input, click track, Expander port, disk drives, the Interface PC board slot, and RAM expansion. RECONFIGURE, therefore, is very useful when installing your Interface Kit and when attempting to interface the system to drum machines and other sequencers.

When the selection is made, the following menu will appear:

```
RECONFIGURE:


     C - CHANGE THIS SET-UP

     S - SAVE THIS SET-UP

     L - LOAD SET-UP

     V - VIEW CURRENT STATUS

 <ESC> - MAIN MENU


     CAUTION : THESE SELECTIONS SHOULD BE
     MADE WITH A KNOWLEDGE OF THE SEQUENCER
     SYSTEM - CONSULT YOUR OWNERS MANUAL!


PLEASE SELECT...
```

This chapter will explain each of these RECONFIGURE selections in detail. When you want to exit to page one of the main menu, just type **<ESC>** when this menu appears.

NOTE: Changes in the interface PC board slot and RAM expansion will be implemented only if you make the changes, save the setup under 'PAGE 3.HEX', then reboot the Sequencer. If you return to the main menu after the change, but before the save, the change will be aborted.

VIEW CURRENT STATUS:

This selection enables you to view the way the parameters are currently setup. It also allows you to learn what the various software revisions there are in the system and the connection status of each port.

A detailed explanation of each entry in the status display follows:

CHECK SYNC? N

This line tells you if the Sequencer will wait for positive going pulses on the SYNC input before starting to play. If CHECK SYNC? is Y, then the Sequencer will display "PLAYING...WAITING FOR SYNC..." when you select PLAY or PLAY ALONG from the main menu. When the required number of pulses are received, the Sequencer will start playing. The WAITING FOR SYNC... message will not be cleared. The Sequencer will not wait for SYNC pulses on subsequent loops.

WAIT FOR HOW MANY? 0

This is displayed only if CHECK SYNC? is Y. The number represents the number of pulses -1 that must occur before the Sequencer will start playing. If you are using a footswitch in the SYNC input, it must be TTL compatible (must not be hardware debounced). The number of wait pulses should be 0 in this case because of probable switch bounce. The range allowed is 0 to 15. The hardware timing restrictions are specified in APPENDIX O.

CHECK FOOTSWITCH? Y

The footswitch is used to tell the Sequencer when to start recording and when to stop playing or recording. The debounce software requires that the Sequencer know if there is a footswitch connected in the FOOTSWITCH input. If you plan to use the Sequencer without the footswitch connected, you must set this to **N** or the Sequencer will not PLAY a sequence.

EMPHASIS ON FIRST BEAT? Y

If this is **Y**, then the click track will emphasize the first beat of the measure.

TIMER SOURCE : INT

The timing information can currently come from 4 sources:

SYMBOL	SOURCE	ACCESS
INT	Internal	I
EXT	External	E or X
SS1	Single Step1	S
SS2	Single Step2	^S

Single Step1 currently is not used (doesn't work). Single Step2 is designed to increment the timer every low to high transition on the SYNC input. It can sense TTL pulses of greater than 3 milliseconds apart or a footswitch in the SYNC input. If a footswitch is used, the timer will increment when the footswitch is depressed (if the footswitch is a normally closed one).

TIME INCREMENT : 1X

The TIME INCREMENT is the amount of increment in the time value for every timer source pulse. The options are 8X, 4X, 2X, 1X, 1/2X, 1/4X, 1/8X and 1/16X. When accessing these options (in CHANGE THIS SET-UP), you need not input the X. This parameter allows a quick method of doubling or halfing the speed of the sequence while keeping the click track intact. It is very useful for recording at 1/2, 1/4 speed, etc. It is also useful when converting a sequence to run on a TIMER SOURCE that is different than the one on which it was recorded.

SEQUENCER REVISION? 5

This is the software revision of the Sequencer program you are running.

THIS SEQ REVISION? 5

This is the software revision of the Sequencer program that was used the last time this sequence was saved. Essentially, it is the software revision of the sequence. The file structure of sequences are designed to be transportable. In other words, you can send a song to a friend! This sort of information is useful when there are several revisions of Sequencers in the field.

THIS SEQUENCE INSTRUMENT? Chroma/Expand

This is the instrument(s) that was used the last time this sequence was saved. The first instrument tells what was connected to the Chroma port and the second one tells what was connected to the Expander port.

THIS SEQUENCE INSTRUMENT REVISION? 3/3

This is the interface software revision of the instrument(s) that was used the last time this sequence was saved. "REV 12" Chroma software has an interface software revision of 2. "REV 14" Chroma software has an interface software revision of 3 (see APPENDIX K).

CHROMA PORT STATUS? Chroma, REVISION? 3

This is the instrument and its interface software revision that is currently connected to the Chroma port.

EXPANDER PORT STATUS? Expander
REVISION? 2 AUTO REROUTE? N

This is the instrument and its interface software revision that is currently connected to the Expander port. The AUTO REROUTE parameter, when 'Y', allows the Sequencer to automatically send a track to the Expander port whenever you attempt to send more than 8 tracks at once to the Chroma port.

DISK SLOT? 6

This is the current Disk Controller slot number. The Sequencer will support only one disk controller. It determines the slot when you power up and provides this information for your reference.

NUMBER OF DRIVES? 2

This is the number of disk drives that are connected to the system. If it is 1, then the Sequencer will direct all disk accesses to Drive 1. If it is 2, then the Sequencer will access Drive 2 for catalog, sequences and program groups and Drive 1 for RECONFIGURE setups.

INTERFACE SLOT? 5

This is the Interface PC board Slot number.

RAM STATUS? 8 BANKS OF 56 SECTORS EACH
IN SLOT 4

This is the expanded RAM status. It indicates the type of RAM and the RAM card slot.

CHANGING THIS SETUP:

This selection allows you to change several of the parameters described in VIEW CURRENT STATUS above. When the colon appears, you may select any of the options that apply to that particular parameter. In this manual, the options are listed to the right of the colon. <DEF> means default, leave it has it is, and is selected by typing **<RET>**.

CHANGE SET-UP...

CHECK SYNC?	: Y, N, <DEF>
WAIT FOR HOW MANY?	: 0 through 15, <DEF>
CHECK FOOTSWITCH?	: Y, N, <DEF>
EMPHASIS ON FIRST BEAT?	: Y, N, <DEF>
TIMER SOURCE :	: INT, EXT, SS1, SS2, <DEF>
TIME INCREMENT :	: 8X,4X,2X,1X,1/2X,1/4X,1/8X,1/16X,<DEF>
AUTO REROUTE?	: Y, N, <DEF>
NUMBER OF DRIVES?	: 1, 2, <DEF>
INTERFACE SLOT?	: 2 through 7
EXPANDED RAM?	: Y, N, <DEF>
RAM SLOT?	: 0 through 7, <DEF>

NOTES : WAIT FOR HOW MANY? is displayed only if checking SYNC.
RAM SLOT? is displayed only if EXPANDED RAM.
In TIME INCREMENT, you do not have to type the X.

SAVING THIS SETUP:

This allows you to save the changed setups for easy recall later. If you save it under the name of 'PAGE 3.HEX', then that setup will be loaded upon power up. The names must follow the name syntax as outlined in APPENDIX P. You should also read the WARNING regarding names in DELETING files in Chapter 7. The Sequencer saves the setup as a binary file on Drive 1, regardless of the number of drives in the system. Since there is no RECONFIGURE setup catalog, you should name the setups such that they will be easily recognized when you catalog drive 1.

LOADING SETUPS:

This allows you to load presaved setups immediately. There is, unfortunately, no menu selection that allows you to catalog these setups. You can find setups by selecting **<ESC>** to FP from the main menu, then typing **CATALOG,D1** and **<RET>**. All files on Drive 1 will be displayed. Type **GOTO 160** then **<RET>** to rerun the Sequencer, then select **X** to return to RECONFIGURE. If you load PAGE 3.HEX, the power up setup will be loaded.

DELETING SETUPS:

The Sequencer currently does not support the deletion of RECONFIGURE setups. If you must delete them (to get more disk space, etc.) you can do the following:

Select **<ESC>** to FP from the main menu,
Type **CATALOG,D1** and **<RET>** then,
When you have found the setup file, **DELETE** it,
Type **GOTO 160** and **<RET>** to rerun the Sequencer.

Typing **GOTO 160** instead of **RUN** will keep the click track of your sequence intact.

EXTERNAL SYNCING

The Sequencer can be controlled by external timing information in many different ways. The most important Sequencer operation parameters when trying to interface it to another system are the TIMER SOURCE and TIME INCREMENT. The EXTCLK input is the most often used; however, the SYNC input can also help to solve external syncing problems. The details of these parameters, including checking the SYNC input, are presented in Chapter 9, RECONFIGURE and will not be duplicated here. In this Chapter, it will be assumed that you understand these parameters.

GENERAL REQUIREMENTS:

The interface hardware and software can sync as a slave to an external device. The external device (drum machine, sequencer, etc.) is a master that sends a clock signal to the Sequencer. The Sequencer then records and plays back according to the speed of the clock signal. The signal specification is detailed in Appendix O, however, an additional requirement is that the signal must be gated. In other words, the clock signal must not change until the external device begins to play and must stop changing when the device stops. Notice that the waveform must have a duty cycle between 25% and 75%.

The interface hardware clocks an external signal on the falling edge. Under most conditions this presents no problems when interfacing to positive edge devices that output normal clock frequencies. If the small delay (25 milliseconds maximum with a 24 pulse per beat clock at 50 BPM) is objectionable, you will have to invert the clock signal with hardware.

The Sequencer does not measure the EXTERNAL clock input, but uses the falling edge of the input as a source of time interrupt. This means that a relatively high frequency is required for enough resolution in real time multi-track recording, which is what the Sequencer does best. In general, there is a very slight resolution problem due to the low clock frequencies of most drum machines on the market. They are different between units, but generally the frequency is 24 or 48 pulses per beat. The speed of the drum sequence can generally be varied from 40 to 250 BPM so the clock frequency is between 16 and 200 Hz. This is enough resolution for sequences that contain only a few tracks; beyond that, you will begin to notice timing errors. Some drum machines output clocks up to 400 Hz, which should present no problems. In no case is the error cumulative.

RECORDING WITH DRUM MACHINES:

To record a sequence, the following steps should be followed:

(1) Connect a cable between the clock output of the external device to the EXTCLK input of the interface connector box.

(2) Use RECONFIGURE to select an external clock in the Sequencer software (see Chapter 9).

(3) Select a pattern on the drum machine with which to record. This will be your "click track." Make sure that the Sequencer's click track says NONE. The drum machine pattern can be edited later and still be in sync.

(4) With the drum machine not playing, go through the normal recording selections. Go all the way, until the inverse video message appears (RECORDING...).

(5) Start the drum machine playing. Play the Chroma along with the drum pattern whenever you want the music to start. Keep in mind that the time delay between the first drum beat and when you first play a note will be remembered and played back exactly.

(6) Stop the recording as normal, then stop the drum machine.

(7) Save the sequence, if desired, then repeat steps 4 through 6 until you are through with the sequence.

PLAYING WITH DRUM MACHINES:

To play a sequence the following steps should be followed:

(1) With the drum machine not playing, press P for PLAY or W for PLAY ALONG.

(2) When the Sequencer displays "PLAYING...", turn on the drum machine.

(3) Play to the end of the sequence or press the space bar or footswitch to stop.

(4) Stop the drum machine.

When recording or playing back, you must make sure that the speed change function is OFF (see Chapter 6).

Also, some drum machines have start/stop switches that bounce if not depressed deliberately. If this happens, the two units will get out of sync because the drum machine will output a few pulses, then stop and start over.

With some drum machines, it is possible to create a song that stops at the end rather than looping forever. If this is the case, you should make the drum machine song end at the same time as the sequence. Full use of rests and special rhythm patterns at the beginning and end of the drum song will allow a professional sounding result.

Once the sequence is saved, the timer source (INTERNAL or EXTERNAL) is saved and reloaded along with the other data. In other words, it is not neccessary to change the timer source in RECONFIGURE when loading an externally synced sequence from disk.

It is not possible to record a sequence using the internal clock, then change to an external clock and have them be in sync. It is possible, however, to change the other way (external to internal) by changing the TIMINC and using the speed changing functions to increase the number of clock pulses required between events (the internal clock frequency is 1000 Hz.).

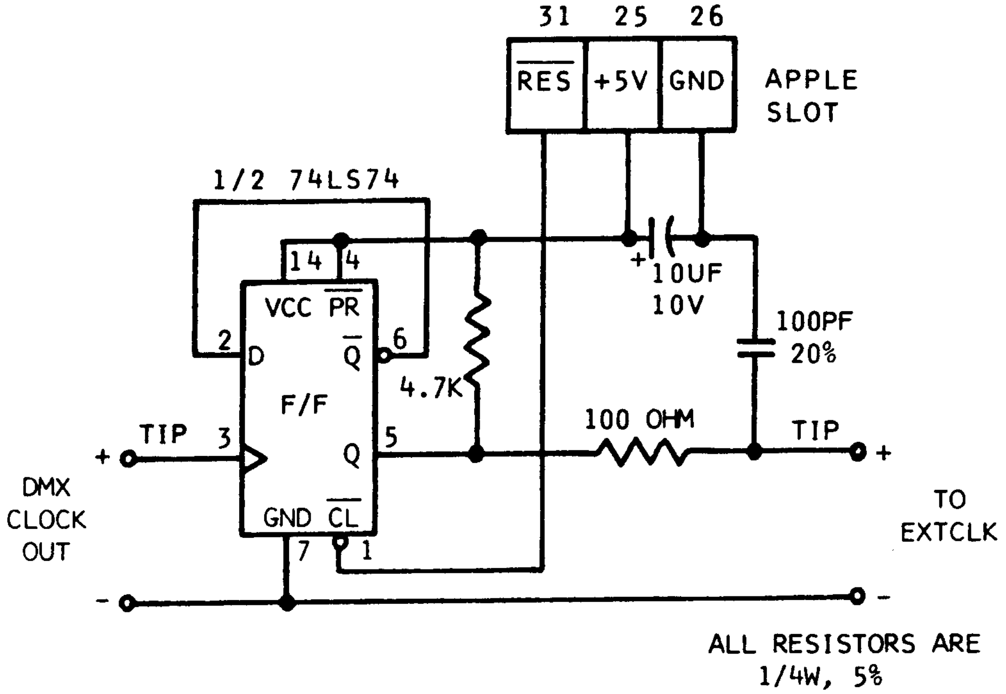
Looping works exactly like looping without a click track, meaning that it is very difficult. The time delay (count off) introduced when recording the first track is included in the loop. You must be careful when you stop the recording of the first track or any extended track, since this is also included in the loop time calculation. Modifying the timing with the editor (see Chapters 6 and 8) is possible and probably necessary.

Compatible drum machines currently consist of the LM1 Drum Machine, the LinnDrum (1) and the Drumulator (2). Connection between the LM1 and the Sequencer is between the Trigger output (set at maximum frequency) and the EXTCLK input of the Sequencer. Connection to the LinnDrum is between the SYNC OUT and the EXTCLK input of the Sequencer. Connection for the Drumulator is between the CLK/CASS OUT and the EXTCLK input of the Sequencer.

Other drum machines are not directly compatible. Small modifications to the interface hardware may allow you to sync them to the Sequencer software. These modifications are described below. However, <u>you will void your warranty if they are performed on the actual PC board</u>.

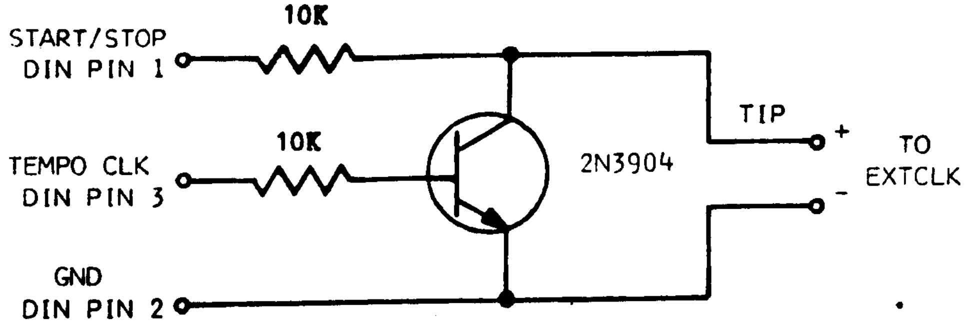
The Oberheim DMX (3) clock output has a constant pulse width of 104 microseconds. This is less than the minimum required duty cycle of 25%. It is possible to change R48 on the interface PC card to allow this small pulse width. This method is not recommended since it may work differently for various clock frequencies and will void your warranty.

A better method is to route the clock signal to a toggle flip/flop (74LS74). This will provide a square wave at one-half the original frequency, which can be directly sent to the connector box EXTCLK input. The frequency division presents no problems, since the DMX is very fast to begin with. The required IC should be mounted on an Apple prototype board and inserted into your computer. Power is available from the Apple slot. The circuit on the following page should work.



ALL RESISTORS ARE 1/4W, 5%

The Roland TR808 and TR606 (4) present a more complex problem, but an easier solution. The clock signal is not gated. Instead, the clock is free running and a start/stop signal is provided. It is possible to gate the free running clock with the start/stop signal, providing a compatible clock signal. The following circuit should work:



ALL RESISTORS ARE 1/4W, 5%

The sync switch on the Roland TR606 and TR808 should be set to OUT since it will be the master. Pins 4 and 5 of the DIN connector are not used. This circuit could be mounted inside a custom-built cable. The slow speed of the TR606 and TR808 and the clock inversion of this circuit may cause undesirable delays.

ADDITIONAL EXTERNAL TIMING OPTIONS:

Single Step:

Lower frequencies (100 Hz to DC) can generally be accomodated using a Single Step2 (SS2) TIMER SOURCE. The SS2 timer source mode accepts a TTL compatible signal connected to the SYNC input. This input can also be used with a footswitch to manually increment the timer during record or playback mode.

The general procedure would be to record using a slow clock or footswitch in SS2 mode, then play back using the EXTERNAL clock mode. This can be cumbersome, but it does work. In one experiment, we have used this method and clocked the Sequencer to play with an envelope follower driven by the bass drum output of a drum machine. In the studio, the control voltage output of a VCA (such as that found in a KEPEX) can be used to clock the Sequencer. Generally, you want a short attack and release time, maximum ratio and adjust the threshold for whatever drum you are using for the clocking.

Using The SYNC Input:

If you record with an <u>ungated</u> external or SS2 Timer Source, you may need to tell the Sequencer when to start playing a sequence.

The Sequencer normally sets its internal timer to the first time value in the sequence before playing. It first plays the first event, then waits for timer increments. When it is time to play the next event it will do so.

After a track is recorded, a ^U2 utility is automatically invoked. This utility inserts a TIME 0 command at the beginning of a sequence if the sequence was recorded with an external or SS2 Timer Source. On playback, the Sequencer will wait the same number of clock pulses that occurred during recording.

If you do not want this, you must effectively delete the TIME 0 command and change RECONFIGURE to wait for a transition on the SYNC input.

You cannot, however, just delete the TIME 0 command. You must, instead, change the time value of the command to the same as the next TIME command. This is very easy to do in the Editor (see Chapter 6).

The SYNC signal can be obtained from a variety of sources, including a trigger output of a drum machine. The signal, however, must be TTL compatible, as detailed in APPENDIX O.

When using the sync method, you must consider what happens in loop mode. When looping back, the sequencer does not check the SYNC input. Instead, it will loop back immediately (assuming you don't have a click track). The pressing of the footswitch or space bar at the end of recording is critical in this application. The loop processing time may also cause cumulative error problems if the external clock speed is relatively high.

Syncing To Tape:

The Sequencer does not provide a sync-to-tape output signal. It relies on the use of a drum machine for this capability.

The general procedure is to lay down a sync track from the drum machine onto audio tape, then use the drum machine and the Sequencer (without the tape machine) to record the track(s) of a sequence. Once the sequence is finished, record (to tape) the drum machine and Sequencer audio while the sync track on tape is controlling the speed of the drum machine, which is controlling the speed of the Sequencer.

Some drum machines (such as the Drumulator (2)) do not have separate external clock and sync-to-tape outputs. If this is the case, try using a Y-adapter to split the signal from the sync track of the tape to both the drum machine and Sequencer external clock inputs.

Some users have had luck with recording sync tracks at -7db. Of course, experimentation with your particular setup will produce the best results.

THE ARPEGGIATOR:

Several people have asked if the arpeggiator in the Chroma can be synced to a drum machine. The answer is no. The Chroma does not have an accurate hardware timer with which it references the arpeggiation speed. It establishes the speed by a software loop, which shortens or lengthens, depending on how busy it is. Although it is easy to record an arpeggio with the Sequencer, the exact number of notes heard will be different from when it was played. It will again be different after recording another track, because the Chroma will see an increase in the amount of data transmitted across the interface. It is, therefore, useful to have the capability of syncing through an external computer.

The J - ARPEGGIATOR selection from the main menu makes it possible to sync arpeggios and chords to an external device, such as a drum machine. The program receives key depression (attack) commands from the Chroma synthesizer and builds its own arpeggios, so there is no need to use the arpeggiation algorithms in the Chroma. The program tells the Chroma to attack and then release one or several notes whenever the EXT CLK input jack receives a trigger pulse from the external device.

In addition, the left footswitch (footswitch 2, latch) is redefined to act as a "hold" switch. When the switch is depressed once, any notes that are currently held will continue to be triggered until the switch is depressed a second time. Notes can be added while the hold switch is on, including notes that are already sounding, making an arpeggio sound the same note more than once in the pattern. Any notes added after the eighth note (fourth note in a "linked" program) will be written over the oldest note.

There are provisions for sequentially adding octave transposed arpeggios to the main arpeggio and for controlling the length of the notes, the linked program and the Expander port. These features, as well as the arpeggiation modes, are detailed below.

The menu presented to the user allows him to easily select between five (5) arpeggiation modes.

ALL NOTES. Upon receipt of a trigger pulse, all notes that are held will be attacked simultaneously, then released (according to the note length parameter described below). As in all modes, no notes will be heard until triggered externally. This mode works well when triggered with an interesting rhythm from a drum machine.

ARPEGGIATE UP AND DOWN. The notes held will be played one at a time in an arpeggio moving up and down. The pattern will start ascending and will be repeated until the keys are released. In this and all other arpeggiation modes, an interesting effect can be achieved by splitting the keyboard. The arpeggios set themselves off the main program so that link unisoned programs can arpeggiate normally. So when you split the keyboard, the arpeggio builds itself around the main program and whenever a link note is played, it will attach itself to one of the main notes of the arpeggio. With the hold switch ON you can even add the same link note to several or all of the main notes of the arpeggio to get a drone effect or have the main and link play in harmony.

ARPEGGIATE BY LEVER 2. Performance lever 2 on the Chroma determines the direction of the arpeggio. Pushing the lever forward once will create an ascending pattern while pulling back once will create a descending pattern. An ascending pattern is automatically set when this mode is selected. The user can change directions in mid-pattern by moving the lever in the desired direction. A nice effect is to use a short, plucked type of sound, add three octaves, turn the drum machine up to full speed triggering sixteenth notes, hold down the sustain footswitch, then play single notes or simple chords to get a quickly strummed harp sound.

RANDOM. The notes held are triggered one at a time in a random pattern. The octave of the note will also be selected at random (according the octave add range described below).

SEQUENCER. The notes will be played according to the order in which they were depressed, creating an eight note (four if "linked") sequencer.

Also selected from the menu are the following special feature controls:

NOTE WIDTH. This parameter describes the duration of the triggered note. Ten values are available and are continuously updated and displayed in the menu as 0 through 9, zero being the shortest. Selecting **L** from the menu lengthens the note duration, while selecting **S** shortens it. These are single key selections, not requiring a computer keyboard carriage return. When using the longer note values there is a maximum speed that an arpeggio can play before the end of one note reaches the beginning of the next. When this point is reached, the arpeggio appears to slow down because the program is ignoring any triggers that occur while the note is being played.

OCTAVE ADDITION. This feature allows the user to sequentially add octave transposed versions of the arpeggio. Up to three (3) octaves can be added to the length of the arpeggio in UP AND DOWN, LEVER 2, RANDOM and SEQUENCE arpeggiation modes. Notice that this feature is not available in the ALL NOTES arpeggiation mode. The currently selected octave addition option is continuously updated and displayed in the menu as NONE, 1 OCT, 2 OCT or 3 OCT. Selection is made by **A** switching to the next available option. Again, this is a single key control. When in one of the added modes the pattern is played before being repeated an octave higher (lower if descending). If your pattern spans more than an octave, the arpeggio will back up when changing octaves. This can be used to create some interesting patterns.

LINK CONTROL. This parameter has two (2) values, indicating that the linked instrument can be either TRIGGERED along with the main instrument or allowed to play as NORMAL. The status is continuously updated and displayed in the menu and is toggled between the two states with the single key **N**. The NORMAL mode is used for creating a normal sound on top of a synced sound, maybe with a keyboard split or with a link unison setup.

Support is provided for the Expander port by automatically transferring performance information from the Chroma port whenever events occur while under control of the main menu of the Arpeggiator. Information originating from the Expander is ignored.

In addition, while under the arpeggiation modes, all notes normally sent back to the Chroma port are also sent to the Expander port. Volume balance between main and link is not transferred to the Expander if the Chroma is revision 12 (interface revision 2) or below. If this is the case, you must set your balance before entering an arpeggiation mode.

Parameter and panel changes are not transferred in either direction. Voice changes on the Chroma, therefore, will not change voices on the Expander.

The sustain footswitch (footswitch one) creates a nice effect by blurring the arpeggio. This sounds nice with a slow sweep or when used in conjunction with one of the Chroma's internal arpeggiation algorithms.

You can change programs on the Chroma while an arpeggio is playing but you must start with a link unison program if you plan to switch between linked and unlinked programs.

You may notice a slight delay in the arpeggio when it is being triggered from some drum machines. The EXTCLK input triggers on the falling edge of a pulse, so if your machine puts out a wide, positive pulse the note will be as late as the pulse is wide. The delay isn't too noticable when using the LinnDrum (1) or Drumulator (2), but is noticeable on the Roland TR808 (4). The solution would be to invert the signal with hardware.

The Arpeggiator will attack a note on each trigger pulse. The normal external clock output of most drum machines is too fast (24 pulses per beat, etc.). What is required in this type of syncing is a 'trigger' output. The Linndrum(1), for example, can be programmed to output a trigger every time the cowbell is to be played. The Drumulator(2) can be programmed to output a trigger pulse every 1/8 or 1/16 note at the metronome output. See your drum machine manual for details.

(1) LinnDrum and LM-1 Drum Computer are registered trademarks of Linn Electronics, Inc.

(2) Drumulator is a registered trademark of E-mu Systems, Inc.

(3) DMX is a registered trademark of Oberheim Electronics, Inc.

(4) TR-606 Drumatix and TR-808 Rhythm Composer are registered trademarks of Roland Corporation.

Notification that certain drum or rhythm machines are compatible with CBS' products should not be construed in any way to represent any endorsement, official or otherwise, by the respective rhythm machine manufacturers or by CBS INC.

MISCELLANEOUS MENU SELECTIONS

This chapter describes some miscellaneous main menu selections that do not really fit into the other chapters.

H - CHROMA<>EXPANDER:

This selection allows the direct transfer of data from the Chroma port to the Expander port and vice versa, simulating the connection of a cable between the two instruments. The program can transpose the note and pressure information as well as copy the master tune value from the Chroma to the Expander port This gives at least a good starting point on getting the two instruments in tune.

When **H** is selected, the following menu appears:

```
THE CHROMA AND EXPANDER CAN NOW TALK!


     T - TRANSPOSE  0
     U - UP 1 OCT
     D - DOWN 1 OCT
     0 - CLEAR TRANSPOSE
     1 - 0
     2 - 0
     3 - 0
     P - PROGRAM MODE
     C - COPY TUNING

 <ESC> - EXIT TO MAIN MENU
```

The zeroes are in inverse video and the computer is waiting for your selection.

COPY TUNING peeks at the tuning value in the instrument on the Chroma port and pokes it into the instrument on the Expander port. In most cases, this tunes the two instruments together. You may have to fine tune the Expander.

TRANSPOSE allows you to set a transposition value between -33 and +33. When the value is inputted (follow the instructions presented when you type **T**) the inverse video field next to TRANSPOSE is updated. All notes in the Expander are squelched and any future attacks, releases and pressure commands are offsetted by the transposition value.

UP 1 OCT sets the transposition value to +12 and updates the inverse video field near TRANSPOSE.

DOWN 1 OCT sets the transposition value to -12 and updates the inverse video field near TRANSPOSE.

CLEAR TRANSPOSE sets the transposition value to 0 and updates the inverse video field near TRANSPOSE.

1, 2, and 3 are programmable preset transposition values that can be selected. The transposition value and the inverse video field near TRANSPOSE is set according the value shown in inverse video near the selection (1, 2, or 3). These values are retained, even if you exit to the main menu and come back to **H** - CHROMA<>EXPANDER. They are set to zero when the Sequencer is turned on.

PROGRAM MODE is entered to set the above mentioned programmable transposition presets. When **P** is selected, the computer will ask for a preset number, then a transposition value. If you type **<RET>** when asked for the preset number, the PROGRAM MODE will be aborted. The inverse video field near the preset number will be updated.

If you enter **H** and the Sequencer immediately returns to the main menu, it is likely that the footswitch is not connected or the CHECK FOOTSWITCH parameter in RECONFIGURE has not been set properly (see CHapter 9).

During the control of this program, the normal SET-SPLIT functions effecting interface communication on each instrument operate as normal. They are listed below for your convenience:

SET SPLIT NUMBER	FUNCTION
16	Turn OFF Performance Information
17	Turn ON Performance Information
18	Turn OFF Panel Information
19	Turn ON Panel Information
34	Turn OFF Pressure Information
35	Turn ON Pressure Information
20	Send Current Program 0

^I - INSTRUMENTS:

The INSTRUMENTS selection allows you to play up to 8 different instruments from the Chroma keyboard, where previously only two instruments could be played.

You can define instruments according to Chroma program numbers and volume. An editing mode is provided to allow you to easily change these parameters.

When the program is entered, it first asks you to define the instruments:

```
ENTER NUMBER OF PROGRAM YOU WISH TO ADD
PROGRAM:
```

Typing a **<RET>** at this point aborts further definition and puts you in the main INSTRUMENT menu. Typing a program number followed by a **<RET>** will define the flashing instrument to that program number and will prompt:

```
ENTER VOL. VALUE (0-255)
VOLUME:
```

This allows you to adjust the volumes of the instruments. Typing a **<RET>** with no value indicates full volume (255).

Once the initial setup has been performed, the main operation mode is entered, where the INSTRUMENT menu presents the following options:

```
    E - EDIT
    M - CHANGE MODE
<ESC> - MAIN MENU
```

The mode (normally UNISON) is displayed in the lower part of the screen.

Entering EDIT mode allows instrument definition similar to that presented at entry to the program. One difference is that the program asks you for the instrument number. This allows you to redefine a previously defined instrument. Typing only a **<RET>** at this point returns you to the INSTRUMENT menu. When asked the program number, you can type **Z** and that instrument will become undefined.

The CHANGE MODE selection essentially toggles between UNISON and ORDERED modes of operation. When UNISON is selected, a key depression causes an attack of all defined instruments simultaneously. In ORDERED mode, the single instrument attacked is determined sequentially. A new sound will occur for each successive key depression.

Support for the Expander port is not provided at this time.

U - PARAMETERS:

The PARAMETERS selection will display the parameters of a Chroma program on the CRT screen in two (2) pages. Both the A and B parameters can be displayed. When the parameters are changed on the Chroma, the values displayed on the CRT screen are updated. When the Chroma program is changed, all values are updated to reflect the new values.

The items to be displayed are controlled by various single key depressions. **P** toggles between displaying Page 1 (parameters 1-26) and Page 2 (parameters 27-50). **A** toggles between displaying A parameters and not displaying them. **B** toggles between displaying B parameters and not displaying them.

Pressing **<ESC>** exits Parameter Display and returns the user to the Sequencer main menu.

Support for the Expander port is not provided at this time.

I - REINITIALIZE:

This selection reinitializes the Sequencer and, to some extent, both the Chroma and Expander. It attempts to re-establish communication with the instruments and, if the instruments are not totally confused, will sync them to the Sequencer. It also sends ID commands to each port and finds out whether they are a Chroma or Expander and their software revisions. The Sequencer does not tell you what it found, but you can view the result from RECONFIGURE, VIEW CURRENT STATUS (see Chapter 9).

^UO - COMMENTS:

This feature is one of the USER UTILITY BANK 1 programs provided with the Sequencer. It allows the user to read or write a 115 character comment in the sequence. This is useful for indicating the name of the program group that was used to record the sequence and as a quick reference to the setup.

When run, the program presents the user with the following menu:

```
USER COMMENT PROGRAM

     S - STORE A COMMENT
     R - READ A COMMENT
 <ESC> - MAIN MENU

PLEASE SELECT...
```

When **R** is selected, the Sequencer will display the current comment then ask the user to press any key to continue. When a key is depressed, the COMMENT menu above is displayed again. The comment displayed may contain non-alphanumeric characters if the comment field is empty.

When **S** is selected, the Sequencer asks the user to type in the comment followed by a **<RET>**. There is currently a 115 character limit. If the comment is too long, the program asks you to type it in again.

If **<ESC>** is selected, the program returns to the Sequencer main menu.

^V0 - INTERFACE TESTER:

This feature is one of the USER UTILITY BANK 2 programs provided with the Sequencer. It allows the user and service technician to test the interface PC card. This program is desrcibed fully in APPENDIX L.

^V1 - ADC ADJUSTER:

This feature is also one of the USER UTILITY BANK 1 programs provided with the Sequencer. It allows the user to test and/or adjust the ADC pedal input range.

When selected from the main menu, the program continuously displays the current ADC pedal input value on the screen in decimal. The pedal is properly adjusted if 0 is displayed when the pedal is in 'heel' position and 255 is displayed when the pedal is in 'toe' position.

If the ADC needs adjustment, push the pedal to 'heel' position and adjust trimpot R29 (the one closest to the rear of the computer) for 0 display. You should first adjust R29 until you start seeing 1's then back off until it is always 0. Then push the pedal to 'toe' position (all the way down) and adjust trimpot R25 (the one closest to you) for 255 display. You should adjust from 254 until the display always reads 255. Then push the pedal to 'heel' position and readjust for 0. For further understanding, see APPENDIX O, HARDWARE DESCRIPTIONS AND SPECIFICATIONS.

Type a **<RET>** to exit back to the Sequencer main menu.

ERROR CONDITIONS AND CODES

There are five different types of error conditions that can occur in the Sequencer:

(1) Those that are fatal, meaning those conditions in which there is a possibility of loosing data. Continuing to operate on a sequence after this type of error is likely to result in other errors.

(2) Those that are not fatal but may or may not stop the current operation.

(3) Errors that occur in the Editor, such as syntax errors.

(4) File Management or disk errors.

(5) Power up error conditions.

FATAL ERRORS:

1 SYNC ERR The Sequencer was expecting an echo response from Chroma but got something else. This condition can occur in many points in the software.

10 SYNC ERR This will occur if the Chroma does not correctly echo the PANEL SWITCH OFF command following a record operation.

12 SYNC ERR This will occur if the Chroma does not correctly echo the STATUS1 command preceeding a record operation.

6 DEFINITION ERR This will occur if a command encountered by the Sequencer references a Chroma instrument that has not been previously defined.

Sync 1, 10, or 12 errors are generally caused by communication problems with the Chroma. If it caused by a glitch, doing a Set-Split 50 on the instruments followed by an I - REINITIALIZE from the Sequencer may solve the problem. Check to port status in RECONFIGURE and clear the sequence before proceeding. If this does not help, try powering up the instruments and the computer according to Chapter 2.

NONFATAL ERRORS:

Four conditions can occur that are not really errors but are exceptions to the normal. These are:

(1) Sequence Memory Full- you have used all available memory for this sequence. Data recorded up to that point is still intact. Recording is terminated and the Sequencer returns to the main menu.

(2) Out Of Instruments- you are attempting to output more than 8 tracks to the Chroma. The track that put you over the limit will be muted and the Sequencer will continue to record or play. If the AUTO REROUTE is enabled this condition cannot occur because up to 8 tracks are automatically rerouted to the Expander whenever more than 8 tracks are sent to the Chroma.

(3) No Expander- You have attempted to send a track to an Expander that is not on line. The track will be muted and the Sequencer will continue to record or play.

(4) Clean Sequence- If you try and perform an operation that cannot be completed because there is no sequence in memory, the Sequencer will return to the main menu.

OTHER NONFATAL ERRORS ARE:

ERRCOD CODES: 10-call upon SEARCH without an EOS command at start of sequence.

11-illegal SEARCH OPERATION call type.

18-measure not found in monitor.

EDITOR SYNTAX ERROR CODES:

12 - unrecognized command in disassemble routine.

13 - syntax error in entering command in the editor.

14 - user attempts to change Define command in editor. This is currently illegal.

15 - user attempts to delete End Of Sequence (EOS) or Begin Of Sequence (BOS) command, insert before EOS or BOS, or change EOS or BOS.

16 - track out of range or not in edit mode in editor.

DISK ERRORS:

3- write protected

4- end of data error

5- file not found

6- volume mismatch

7- I/O error

8- disk full

9- file locked

10-syntax error

11-no buffers available

12-file type mismatch

13-program too large

14-not direct command

15-unrecognized error

OTHER DISK ERRORS:

The Sequencer will print specific messages when the data diskette is full, if the overlay called upon was not found or the sequence or program group file was not found.

POWER UP ERRORS:

The Sequencer will attempt to talk to a Chroma on the Chroma port and an Expander on the Expander port. If communication on the Chroma port does not occur the Sequencer will tell the user to check the cables and try again. If communication occurs but the instrument is not a Chroma and/or the software revision of the instrument is higher than expected, it will tell you so and ask if you want to continue. If communication does not ocurr on the Expander port, the Sequencer merely tells the user that the Expander is 'not on line'. If it is on line, but the instrument is not an Expander or Chroma and/or the software revision of the instrument is higher than expected, the Sequencer will tell the user and asks if he wants to continue.

As you can see, only a Chroma should be connected to the Chroma port (at this time). This is because the Chroma can send key depression data and it is the Chroma port that is referenced when the Sequencer records a track. The Expander port can except either a Chroma or an Expander. We can not say that all future products will be compatible with this Sequencer, but the large majority of them should be compatible after little more than a diskette update.

SEQUENCE AND PROGRAM GROUP FILE DATA STRUCTURES

This APPENDIX describes computer data structures and is rather technical in nature. Knowledge of the information contained here is not necessary for operation of the Sequencer.

Both sequences and program groups can be stored on disk in the Sequencer. Sequences are stored as 'NEW TYPE B' files and program groups are stored as 'NEW TYPE A' files. These are not Applesoft and Binary file types; they are totally different file types that cannot be accessed by normal DOS commands. Using these file types allows the catalog routine to easily determine when to print an entry. Although any file may be stored on disk, only sequences and programs will be listed when the user asks for a catalog while under control of the Sequencer. In the Sequencer catalog, file type entries for sequences are printed as an 'S' and program groups are printed as a 'P'. The file type is displayed where it is displayed in the normal Apple Dos Catalog (preceeding the length of the file). This file type arrangement also allows the Sequencer to easily distinguish between sequence and program files when loading, thereby eliminating the possibility of loading a program group into sequence RAM. The Apple DOS does not check file types when deleting a file.

Both sequences and program groups are stored and retreived using the DOS File Manager routines. This allows fast and easy storage of the normally seqmented sequence queue. It also allows grabbing one program packet at a time from the Chroma, thereby eliminating the need for a 4259 byte buffer. This is nice because the program group file formation will not interfere with the sequence or assembly language program RAM. For a comprehensive description of the DOS file manager routines, read BENEATH APPLE DOS written by D.Wirth and P.Lechner and published by Quality Software, 6660 Reseda Blvd.,Reseda,CA 91335. To my knowledge, this information is available only in that publication.

PROGRAM GROUP FILE STRUCTURE:

The structure of the program group allows up to 50 programs. The group may include only one program if desired. The data structure for the group file is as follows:

00 number of programs in the group (01 to 32 hex)
01 File type - always 20 hex
02 ID number of Rhodes instrument from which the data was received.
03 Revision number of the software of the Rhodes instrument.
04 Group type (used for group catagory searching)
05 Data Structure Revision (this revision is 0)
06-08 Reserved for future use.
09-xx Program entries

PROGRAM ENTRY FORMAT: (85 bytes total)

00 program number
01 program type (used for searching for catagories of programs
02-10 program name (15 bytes maximum - if less then last char is 8D for cr)
11-19 not used
1A-54 program data (59 bytes)

EXPANSION OF 59 BYTE PROGRAM DATA:

ParaNo.	Group	Name	Byte(s)	7 6 5 4 3 2 1 0
0	Panel	Link Balance	31	[- - - - N N N N]
1	Control	Patch	1	[- - - - N N N N]
2	Control	Fsw Mode	5	[- - - - - N N N]
3	Control	Kybd Alg	31	[N N N N - - - -]
4	Control	Detune	2	[N N N N N - - -]
5	Control	Output Select	2	[- - - - - N N -]
6,56	Glide	Rate	28,58	[N N N N N - - -]
7,57	Glide	Shape	14,44	[- N - - - - - -]
8,58	Sweep	Mode	4,34	[- - - - - - N N]
9,59	Sweep	Rate	4,34	[N N N N N N - -]
10,60	Sweep	Rate Mod	3,33	[- - - - N N N N]
11,61	Sweep	Wave Shape	6,36	[N N N N - - - -]
12,62	Sweep	Ampl Mod	6,36	[- - - - N N N N]
13,63	Env 1	Ampl Touch	9,39	[- - - - - N N N]
14,64	Env 1	Attack	7,37	[N N N N N - - -]
15,65	Env 1	Attack Mod	7,37	[- - - - - N N N]
16,66	Env 1	Decay	8,38	[N N N N N - - -]
17,67	Env 1	Decay Mod	8,38	[- - - - - N N N]
18,68	Env 1	Release	9,39	[N N N N N - - -]
19,69	Env 2	Delay	10,40	[N N N N N - - -]
20,70	Env 2	Ampl Touch	13,43	[- - - - - N N N]
21,71	Env 2	Attack	11,41	[N N N N N - - -]
22,72	Env 2	Attack Mod	11,41	[- - - - - N N N]
23,73	Env 2	Decay	12,42	[N N N N N - - -]
24,74	Env 2	Decay Mod	12,42	[- - - - - N N N]
25,75	Env 2	Release	13,43	[N N N N N - - -]
26,76	Pitch	Tune	14,44	[- - N N N N N N]
27,77	Pitch	Mod 1 Select	18,48	[N N N N - - - -]
28,78	Pitch	Mod 1 Depth	15,45	[- N N N N N N N]
29,79	Pitch	Mod 2 Select	18,48	[- - - - N N N N]
30,80	Pitch	Mod 2 Depth	16,46	[- N N N N N N N]
31,81	Pitch	Mod 3 Select	19,49	[N N N N - - - -]
32,82	Pitch	Mod 3 Depth	17,47	[- N N N N N N N]
33,83	Width	Wave Shape	20,50	[- - - - - - N N]
34,84	Width	Width	20,50	[N N N N N N - -]

ParaNo.	Group	Name	Byte(s)	7 6 5 4 3 2 1 0
35,85	Width	Mod Select	19,49	[- - - - N N N N]
36,86	Width	Mod Depth	21,51	[- N N N N N N N]
37,87	Cutoff	LP/HP	15,45	[N - - - - - - -]
38,88	Cutoff	Resonance	10,40	[- - - - - N N N]
39,89	Cutoff	Tune	22,52	[- - N N N N N N]
40,90	Cutoff	Mod 1 Select	26,56	[N N N N - - - -]
41,91	Cutoff	Mod 1 Depth	23,53	[- N N N N N N N]
42,92	Cutoff	Mod 2 Select	26,56	[- - - - N N N N]
43,93	Cutoff	Mod 2 Depth	24,54	[- N N N N N N N]
44,94	Cutoff	Mod 3 Select	27,57	[N N N N - - - -]
45,95	Cutoff	Mod 3 Depth	25,55	[- N N N N N N N]
46,96	Volume	Mod 1 Select	27,57	[- - - - N N - -]
47,97	Volume	Mod 1 Depth	3,33	[N N N N - - - -]
48,98	Volume	Mod 2 Select	27,57	[- - - - - - N N]
49,99	Volume	Mod 2 Depth	5,35	[N N N N - - - -]
50,100	Volume	Mod 3 Select	28,58	[- - - - - N N N]
51	Panel	Link	0	[N N N N N N N N]
52	Panel	Edit	30	[N N N N N N N N]
53	Panel	Keyboard Split	32	[N N N N N N N N]
54	Panel	Main Transpose	1	[N N - - - - - -]
55	Panel	Link Transpose	1	[- - N N - - - -]
	Sequence Program Footswitch		29	[N N N N N N N N]
	Free bits		2	[- - - - - - - N]
			5	[- - - - N - - -]
			35	[- - - - N N N N]
			14,44	[N - - - - - - -]
			16,46	[N - - - - - - -]
			17,47	[N - - - - - - -]
			21,51	[N - - - - - - -]
			22,52	[N N - - - - - -]
			23,53	[N - - - - - - -]
			24,54	[N - - - - - - -]
			25,55	[N - - - - - - -]

Presently, the sequencer will store and retreive Groups consisting of 50 programs only. The first byte of the program name field will be 8D to signify that it is not named. Types (both group and program) will be 00 signifying sequence programs.

A Main Menu selection Q - Program Files is provided. When selected, the sequencer will display the following menu:

PROGRAM FILE MANAGEMENT

1 - Save Chroma programs
2 - Get programs from disk and send to Chroma
3 - Delete program group from disk
and if there is an Expander on line:
    4 - Save Expander programs
    5 - Get Expander programs from disk

See Chapter 7 for a detailed description of these options.

SEQUENCE FILE STRUCTURE:

The sequence file is structured as a block of actual sequence data precceded by a two page global variable section. The sequence data always begins and ends with an EOS command (opcode 00 followed by two bytes of time data). The absolute location of the start of the file in RAM is contained in MANSEQ (zero page locations $1E and $1F). In the controlling BASIC program this variable is called I%, so in this file description, we will call the beginning of the file I%. Following is a detailed description of the two page global variable section that preceeds the data section.

I% -FINSTRUO- Last track recorded (00 signifies clean slate).

I%+1 -FTYPE- File Type (always $40).

I%+2 -FIPTRL- File input pointer low (the end +1 of the data section low).

I%+3 -FIPTRH- File input pointer hi (the end +1 of the data section hi).

I%+4 -FOPTRL- File output pointer lo (the start of the data section low).

I%+5 -FOPTRH- File output pointer hi (the start of the data section hi).

I%+6 -FTIMSIG- Time Signature (MS nibble is beats per measure, LS nibble is note duration, as in standard time signature notation). If METRFL =0 then TIMSIG =$44 for 4/4 time.

I%+7 -FCBCRGL- Timer counts between click track outputs LO (=60000/BPM). If METRFL=0 then CBCRG=1000dfor 60 BPM.

I%+8 -FCBCRGH- Timer counts between click track outputs HI.

I%+9 -FMETRFL- Metronome enable flag ($00= never a click track, $40= currently disabled, $C0= enabled).

I%+10 -RESERVED- Next available measure number in REV 4 Sequencer.

I%+11 -FSEQREV- Software revision of the sequencer that generated this sequence.

I%+12 -FSEQNAM- Sequence name (15 bytes) - if not all 15 bytes are used, then $8D at end of name.

I%+27 -FTIMINC- Timer increment (allowable values are $80,$40,$20,$10,$08,$04,$02,$01 which correspond to time increments of 8,4,2,1,1/2,1/4,1/8 and 1/16).

I%+28 -FINSTBL- Instrument Table (Bit 7 mute active high, Bits 6-0 program number) 16 bytes total.

I%+44 -FIVOLTB- Initial Volume Table (these are the volume values that are imbedded in the define commands) 16 bytes total. Currently not used.

I%+60 -FLVOLTB- Last Volume Table (these are the last volume values of each track- to be used by continuous volume) 16 bytes total. Currently not used.

I%+76 -FTCLKL- Click time at first event in clean slate record low (used to sync click track on subsequent record operations).

I%+77 -FTCLKH- Click time at first event hi.

I%+78 -FTBEAT- The BPMEAS at first event in clean slate record (used to sync click track on subsequent record operations).

I%+79 -FIDNUM- ID number of instrument on Chroma port.

I%+80 -FIDNUMX- ID number of instrument on Expander port.

I%+81 -FREVNUM- Software revision number of instrument on Chroma port.

I%+82 -FREVXP- Software revision number of instrument on Expander port.

I%+83 -FTRANTB- Transpose table (used to store transpose status of each track) 16 bytes total. Currently not used.

I%+99 -FTIMSRC- Timer Source, $FF=External, $00=Internal 1KHz, $7F=Debounced Single Step, $3F=Single Step.

I%+100 -FWRBANK- Expanded RAM write bank number.

I%+101 -FRDBANK- Expanded RAM read bank number.

I%+102 -FWRSECT- Expanded RAM write sector number.

I%+103 -FRDSECT- Expanded RAM read sector number.

I%+104 -FRAMEXP- Expanded RAM flag (0=nonexpanded, other=slot#).

I%+105 -FMAXBANKS- Maximum number of banks in Expanded RAM.

I%+106 -FMAXSECTS- Number of sectorsd per bank in Expanded RAM.

I%+107 -FMEASR5L- Next available measure number (LO).

I%+108 -FMEASR5H- Next available measure number (HI).

I%+109 to 139- 31 bytes reserved for future use.

I%+140 -FNAMTBL- Track name table, each of 16 entries contains 15 bytes (if track name does not use full 15 bytes, $8D is stored at end of name).

I%+380 to 395- 16 bytes unused, reserved for future use.

I%+396 to 511- 116 byte field for comments, stored and read through USER1, number 0 overlay.

I%+512 to (FIPTR)-1 Sequence data in non-expanded mode.

SEQUENCE DATA FORMAT:

EOS $00 Marks beginning and ending of sequence data.

TIME $FF xxyy where xx is low byte, yy is high byte.

MEASURE $FD xxyy where xx is low byte, yy is high byte.

```
DEFINE $C0 tt aa bb cc dd ee ff
          where tt = Track # ($01-10)
                aa = Lever1 value ($00-FF, signed 2's complement)
                bb = Lever2 value ( same as above )
                cc = Pedal1 value ($00-$FF, unsigned)
                dd = Pedal2 value ( same as above )
                ee = Initial Volume Value (same as above)
                ff = Footswitches (MSB=Footsw1, next bit=Footsw2
                                   0=up, 1=down)
```

All others are stored as detailed in APPENDIX I, except that the opcode will always reflect instrument 0 and will always be followed by a track byte ($01-10).

$F8-FC and $FE opcodes are reserved for future utility commands.

PROGRAM OVERLAY STRUCTURE and USER UTILITY PROGRAMS

Due to the scarcity of RAM in the Apple system, a small section of address space in the RAM card has been designated as an overlay area. This area will contain different programs, depending on what is needed at any given time. USER UTILITY programs are provided for, should the user want to write is own routines.

Each of these programs or any routine within these programs is entered by calling a small routine (calling routine) that resides outside of the overlay area. This routine will check to see if the desired program is in memory by looking at 'OVERLAY', which will have one of the following possible values:

00 - No program in memory
01 - EDITOR in memory
02 - SET.UP.CLICK in memory
03 - MENU.PAGE.3 in memory
04 - MARKERinmemory (implemented as ARPEGGIATOR)
05 - SYS.PORT in memory (implemented as RECONFIGURE)
06 - VOLUME in memory (Currently not written)
07 - One program from USER UTILITY BANK 1 in memory
08 - One program from USER UTILITY BANK 2 in memory
09 - TRACK.DIR/SEARCH in memory
0A - FILER in memory
0B - TRANSFER in memory
0C - SYS.SETUP
0D - MEMLOCK1
0E - MEMLOCK2
0F - Reserved by CBS
10 - Reserved by CBS

If the desired program is not already in memory, the calling routine will attempt to load the program. If the calling routine cannot load the desired program, it will display the error code and the message:

"SORRY, SELECTION IS NOT AVAILABLE"
"PLEASE SEE YOUR OWNERS MANUAL"

If the load is successful, the calling routine will call the desired routine within the recently loaded program.

MEMLOCK1 and MEMLOCK2 are never called by the calling routine. Instead, they are used internally. There is further explanation in APPENDIX F.

USER UTILITY PROGRAMS:

There are two "banks" of user or utility programs selectable from the main menu. Each bank can include up to 10 individual overlay programs, accessed by USER NUMBERS 0-9 entered after the USER1 or USER2 selection from the main menu. The object files have the titles USER10.HEX to USER19.HEX for the USER1 bank and USER20.HEX to USER29.HEX for the USER2 bank.

USER UTILITY program overlays are always loaded regardless of whats in memory because the OVERLAY register just indicates whether a BANK 1 or BANK 2 program is in memory. The USERNO register is set to the USER NUMBER in BASIC and the overlay calling routine uses this information to load the appropriate USER overlay.

It is also possible to call a user overlay from another user overlay and return back to the original. Neither the source nor the destination addresses have to be at the beginning of the overlay.

In the case of USER UTILITY 1 and 2 overlays, the overlay calling routine will JUMP to the start of the overlay file (OVERLAY.BASE), so the programmer should put a JMP to the start of his routine at the beginning of his object file. Currently the overlay area is approximately 2K bytes long. Care must be taken to insure that the overlay program does not write over the RESET and IRQ vectors at FFFC-FFFF.

Actual programming of these USER overlays is not difficult, since many of the variables and routines likely to be required are already set-up. Programmers must include the source file 'OVERLAY.EQUATES.SRC' in their programs to access these variables and routines.

Currently, there are 7 user program overlays written as examples, two in BANK 1 and two in BANK 2. They are described below:

USER UTILITY BANK 1

number0 (USER10.HEX) COMMENT

This program allows the user to read or store a 115 character comment in the sequence. This is useful for indicating the name of the program group that was used to record the sequence and as a quick reference to the set-up.

number1 (USER11.HEX) ECHO

This program sends multiple NOP commands to the Chroma, keeping it so busy that it will sound like tremelo or echo. It will only work on REV 2 Chromas.

number2 (USER12.HEX) TIME0

This program inserts a TIME command with a value of 00 at the beginning of a sequence that uses an external clock (see Chapter 10).

USER UTILITY BANK 2

number0 (USER20.HEX) INTERFACE TEST

This is a test program for the interface described in APPENDIX L.

number1 (USER21.HEX) ADC TEST

This is a short ADC test program to allow you to fine tune the pedal range. You can also do this in the test program described above (USER20).

number6 (USER26.HEX) PARAMETER

This is a program that displays the numerical values of the parameters of the current Chroma program (see Chapter 11).

number7 (USER27.HEX) INSTRUMENTS

This program allows the musician to play multiple instruments from the Chroma keyboard (see Chapter 11).

SUGGESTIONS FOR USER UTILITY PROGRAMS:

- A USER UTILITY Catalog.
- A selective quantizing or error correcting program.
- A Program that gives a Catalog of RECONFIGURE Set-ups and allows deletion of these set-ups from disk.
- A Sequencer Data Diskette initialization/copy program.
- A track merge/replace program.
- A track duplication program.

```
1000
1010
1020
1030 **********************
1040 *                    *
1050 *  OVERLAY EQUATES   *
1060 *                    *
1070 **********************
1080
1090 *  Stored under 'OVERLAY.EQUATES.SRC'
1100
1110 *  VERSION 1-24-84
1120
1130 *  (C) COPYRIGHT 1982, 1984 CBS INC.
1140
1150
1160 OVERLAY.BASE   .EQ $F252   This is the origin of all overlays.
1170 BUFFER         .EQ $94F5   This is the general purpose buffer.
1180 MAXFILES       .EQ $03     Current number of open files allowed.
1190
1200 * -------------------- *
1210 *  ZERO PAGE LOCATIONS *
1220 * -------------------- *
1230
1240 DATA    .EQ $06 I/O Driver Input/Output Byte 6
1250 IWRMSK  .EQ $07 Internal Interrupt Mask Image 7
1260 CTDNOFL .EQ $08 General purpose ZP register (word) 8
1270 SIPTRL  .EQ $18 SEQ-Q Input Pointer (word) 24
1280 SOPTRL  .EQ $1A SEQ-Q Output Pointer (word) 26
1290 MANSEQL .EQ $1C Start Addr of Seq File (word) 28
1300
1310 WNDTOP  .EQ $22 Top Line of Scroll Window
1320 CH      .EQ $24 Horizontal Cursor Position
1330 CV      .EQ $25 Vertical Cursor Position
1340 BAS2L   .EQ $2A Scroll Operation Work Area (word)
1350
1360 PCL     .EQ $3A    !
1370 PCH     .EQ $3B    !
1380 A1L     .EQ $3C    !
1390 A1H     .EQ $3D    !  GP Monitor Registers
1400 A2L     .EQ $3E    !
1410 A2H     .EQ $3F    !
1420 A3L     .EQ $40    !
1430 A3H     .EQ $41    !
1440
1450 ACC1    .EQ $45 BASIC ACC Save 69
1460 BXSAV   .EQ $46 BASIC X Save 70
1470 BYSAV   .EQ $47 BASIC Y Save 71
1480 STATUS  .EQ $48 STATUS Reg Save 72
1490 TXTPTR  .EQ $B8 BASIC Text Pointer
1500 SPNT    .EQ $49 Saved Stack Pointer
1510 PROMPT  .EQ $50 PROMPT Char ASCII
1520 COUNT   .EQ $9E Hex-Dec Count Reg Used In File Manage (word)
1530 PREG    .EQ $48 USER STATUS Register
```

```
1540 INVFLG  .EQ $32 Video Format Control Reg
1550 LINNUM  .EQ $50 BASIC Line Number Reg (word)
1560 ONERR   .EQ $D8 APPLESOFT/DOS ONERR Flag
1570 ERRNUM  .EQ $DE Error Code
1580 CHARAC  .EQ $0D BASIC String Char Reg
1590 INDEX   .EQ $5E BASIC GP Reg
1600 HIMEM   .EQ $73 Highest+1 memory address for BASIC (word)
1610
1620
1630
1640 *-------------------*
1650 *  PAGE 3 LOCATIONS  *
1660 *-------------------*
1690
1700 STATUS0 .EQ $0300 Status and GP String Buffer (6 bytes) 768
1710 STATUS1 .EQ $0306 Status and GP String Buffer (9 bytes) 774
1720 SRCTYP  .EQ $030F Type of Search Operation Reg 783
1730 MNLOCL  .EQ $0310 Pointer to Monitor Routine (word) 784
1740 SRCOP   .EQ $0312 Opcode to Search For 786
1750 SRCTRK  .EQ $0313 Track or Meas # to Search For 787
1760 SUBLOL  .EQ $0314 Loc of Subroutine After Search (word) 788-789
1770 SYNCCK  .EQ $0316 SYNC Reg-Bit 7 Check, Bit 5 Image BITS 3-0 Inc 790
1780 FOOTCK  .EQ $0317 FOOTSW Reg-Bit 7 Check, Bit 5 Image,791
1790 CLIKEM  .EQ $0318 Click Emphasis Reg (00=EMPH, FF=NOT) 792
1800 SOURCE  .EQ $0319 GET Data Source Reg (SEQ-Q=00) 793
1810 OVERLAY .EQ $031B Indicates Current Program In Memory 795
1820 ASAVI   .EQ $031C ACC Save Reg Used by MONO and ROMSUB 796
1830 ERRCOD  .EQ $031D Error Code for BASIC 797
1840 TIMINC  .EQ $031E Timer Increment Value Reg 798
1850 INSTRB  .EQ $031F Temporary INSTRU0 Register 799
1860 VARTIM  .EQ $0320 Var Time (FF=LRG VAR, 80=SM VAR,00=NO VAR) 800
1870 STRTIM  .EQ $0321 Store Variable Time Flag (FF=STORE) 801
1880 TIMSRC  .EQ $0322 Timer Source Reg (00=INT, FF=EXT) 802
1890 SEQREV  .EQ $0323 REV Number of This Program 803
1900 IDNUM   .EQ $0324 CHROMA Port Instrument ID Number 804
1910 IDNUMX  .EQ $0325 EXPANDER Port Instrument ID Number 805
1920 REVNUM  .EQ $0326 CHROMA Port Software Revision Number 806
1930 REVXP   .EQ $0327 EXPANDER Port Software Revision No. 807
1940 ENDMES  .EQ $035B Ending measure register (word) 859
1950 MEASUR  .EQ $035D Next available measure (word) 861
1960 DELMES  .EQ $035F Delay measure no. (no delay=00) (word) 863
1970
1980 RAMRAM.LOC      .EQ $0330 Addr of RAMRAM switching routine (word) 816
1982 RAMDISK.LOC     .EQ $0340 Addr of RAMDISK driver (word) 832
1984 EXP.IO.LOC      .EQ $0342 Addr of alt. expander port I/O driver (word) 834
1986 CHROMA.IO.LOC   .EQ $0344 Addr of alt. Chroma port I/O driver (word) 836
1990 MAXBANKS        .EQ $0361 Max. no. of banks in expanded RAM 865
2000 MAXSECTS        .EQ $0362 No. of sectors in bank in expanded RAM 866
2005
2010 PRESSURE.SWITCH .EQ $0363 Pressure Record Flag (FF=ON) 867
2020 PRESS.COM.FLAG  .EQ $0364 Pressure Command Flag (FF=PRESSURE) 868
2030
2040 CLEAN     .EQ $0365 Clean Slate Flag (CLEAN=00) 869
2050 ACCSAV    .EQ $0366 ACC Save Used by IOSAVE/REST 870
2060 RAMSTATUS .EQ $0367 RAM Card Status 871
```

```
2070 USERNO    .EQ $0368 Number of USER Overlay to Call 872
2080 REROUT    .EQ $0369 Reroute to Expander Flag (ENABLED=01) 873
2090 DSLOT     .EQ $036B Disk Controller Slot No. 875
2100 DDRIVE    .EQ $036C Disk Drive No. For SEQ & PROG 876
2110 CALNUM    .EQ $036D Number of Assy Call From BASIC 877
2120 LOGO      .EQ $036E MAIN Track # 878
2130 LOG1      .EQ $036F LINK Track # 879
2140 IFSLOT    .EQ $0370 Interface SLOT Number *16 880
2150 ASLOCL    .EQ $0372 Pointer to Assy Routine (word) 882
2160 INSTRUO   .EQ $0374 Track Number Reg 884
2170 MSKTBL    .EQ $0375 Active CHROMAs Reg (00=CHROMA ONLY) 885
2180 RECPLA    .EQ $0376 Record/Play Flag (PLAY=00) 886
2190 LOOPSW    .EQ $0377 Loop ON/OFF Switch (ON=00) 887
2200 PLALNG    .FQ $0378 Play Along Flag (PLAY=00) 888
2210 RAMEXP    .EQ $0379 Expanded RAM flag (00=no RAM, other=slot) 889
2220 RAMSTAT   .EQ $037A Used by INIT.INT and RAMDISK 890
2230 MAXLNL    .EQ $03F5 Max Ending Addr of SEQ File (word) 1013
2240
2250
2260   * ------------------- *
2270   * APPLE I/O LOCATIONS *
2280   * ------------------- *
2290
2300 KBD      .EQ $C000  Keyboard Character and Strobe -16384
2310 KBDSTB   .EQ $C010  Keyboard Strobe Acknowledge -16368
2320 CASS     .EQ $C020  Cassette Output Toggle Read -16352
2330 SPKR     .EQ $C030  Speaker Toggle Read -16336
2340
2350
2360   * ----------------------- *
2370   * NORMAL BUFFER LOCATIONS *
2380   * ----------------------- *
2390
2400 * No buffers (except LINBUF) can be accessed until a call to RAMRAM
2410 * has been performed.  The location of RAMRAM is contained in
2420 * RAMRAM.LOC, $0330, 816 decimal.
2430
2440 OPCODE       .EQ $D000 Opcode length table
2450 IQ.PAGE      .EQ $D100 I/O Input Queue
2460 OQ.PAGE      .EQ $D200 I/O Output Queue
2470 VOLUME.TABLE .EQ BUFFER
2480 LINBUF       .EQ $0200 APPLE Input Line Buffer 3390
2490 MAIN         .EQ $D300  Start of the main variable/routine jump table
2500
```

```
2510 * ------------------------------------ *
2520 * RAM CARD VARIABLES IN MAIN PROG BASE *
2530 * ------------------------------------ *
2540
2550 * No variabes can be accessed until a call to RAMRAM has been performed.
2560 * The location of RAMRAM is contained in RAMRAM.LOC, $0300, 816 deimal.
2570
2580 LINE          .EQ MAIN+210  Catalog line numbers on a page
2590 ERRPRT        .EQ MAIN+211  Print error flag (FF=print errors)
2600 ACTDRV        .EQ MAIN+212  Actual disk drive to reference
2610 TRANSP        .EQ MAIN+215  Key offset (signed) in transpose
2620 VOLUME        .EQ MAIN+216  Track volume register
2630 PHRASE        .EQ MAIN+217  ASM phrase counter in Editor
2640 ASMSTR        .EQ MAIN+218  ASM string pointer in Editor
2650 SIGN          .EQ MAIN+219  Key number sign flag in Editor
2660 MON1          .EQ MAIN+220  Editor MONITOR flag (FF=ON)
2670 STMEAS        .EQ MAIN+221  Store measures flag (FF=store)
2680 BPMEAS        .EQ MAIN+222  Beats per measure counter
2690 XRDFRC        .EQ MAIN+223  Fractional variable timer value
2700 XRDLST        .EQ MAIN+224  LO variable timer value
2710 XRDMST        .EQ MAIN+225  HI variable timer value
2720 FIXTMFRC      .EQ MAIN+226  Fractional fixed timer value
2730 FIXTML        .EQ MAIN+227  Fixed timer value (word)
2740 LASTML        .EQ MAIN+231  Last sequence time reg. (word)
2750 CLIKTML       .EQ MAIN+235  Next click time register (word)
2760 TEMPOL        .EQ MAIN+237  SOPTR & GP temporary register (word)
2770 ADDRL         .EQ MAIN+239  Temporary address register (word)
2780 BUFPNT        .EQ MAIN+241  COMBUF pointer
2790 COMBUF        .EQ MAIN+242  8 byte command buffer
2800 PORT          .EQ MAIN+250  Port reg. (00=Chroma, 01=Expander)
2810 CLIK.OFF      .EQ MAIN+251  Click track enable (00=on)
2820 COUNTL        .EQ MAIN+252  Register from HEX-DEX (word)
2830 DEBUG         .EQ MAIN+254  Debug mode flag ($00=debug, $80=run)
2840 MUTFLG        .EQ MAIN+255  Current track mute flag ($80=muted)
2850 PLAEND        .EQ MAIN+258  End of play flag (FF=end)
2860 INSTRU        .EQ MAIN+259  Temporary instrument number register
2870 RECSW         .EQ MAIN+260  Record switch (00=record on)
2880 ENDTML        .EQ MAIN+261  Ending time register (word)
2890 UNDEFTM       .EQ MAIN+263  Time of stop at end of record (word)
2900 MEASTM        .EQ MAIN+265  Total timer counts in a measure (word)
2910 BITEST        .EQ MAIN+267  Test byte constant (always $40)
2920 XIIN          .EQ MAIN+268  I/O input queue input pointer
2930 XIOUT         .EQ MAIN+269  I/O input queue output pointer
2940 X01N          .EQ MAIN+270  I/O output queue input pointer
2950 XOOUT         .EQ MAIN+271  I/O output queue output pointer
2960 CTIMEL        .EQ MAIN+272  Input command time (word)
2970 TRYAGN        .EQ MAIN+276  I/O number of times-1 to try
2980 SYNCIM        .EQ MAIN+277  Temporary SYNC input image
2990 FOOTIM        .EQ MAIN+279  Footswitch image
3000 BOUNCE        .EQ MAIN+280  Footsw debounce delay (currently $14)
3010 LOCAT         .EQ MAIN+283  Monitor calls routine addr. reg. (word)
3020 LAST.ADC.VAL  .EQ MAIN+284  I/O value of last ADC read
3030 DELAY         .EQ MAIN+471  Delay monitor flag (00=enabled)
3040 SEPASS        .EQ MAIN+472  Sequence loop counter (00=first pass)
```

```
3050 KYCHAR         .EQ MAIN+473  Key ASCII via PRESS
3060 USER1.SPACE    .EQ MAIN+474  2 bytes user overlay variable (word)
3070 USER.SPACE     .EQ MAIN+476  single byte user overlay variable
3080 ACTBL0         .EQ MAIN+483  Active Chroma instr. tble. (8 bytes)
3090 ACTBL1         .EQ MAIN+491  Active Expander instr. tble. (8 bytes)
3100 TIMCOMP        .EQ MAIN+502  Timer window value (currently $02)
3110 SSIMAGE        .EQ MAIN+503  Single step SYNC image
3120 SSIMAGE0       .EQ MAIN+504  Last single step SYNC image
3130 TRANSP0        .EQ MAIN+505  Chroma<>Expander transpose value
3140 TRANSP1        .EQ MAIN+506  Programmable transpose 1
3150 TRANSP2        .EQ MAIN+507  Programmable transpose 2
3160 TRANSP3        .EQ MAIN+508  Programmable transpose 3
3170 TRACK.TABLE    .EQ MAIN+509  GP Track Table 16 bytes (Used by EDITOR)
3180 WRBANK         .EQ MAIN+525  Expanded RAM write bank pointer
3190 RDBANK         .EQ MAIN+526  Expanded RAM read bank pointer
3200 WRSECT         .EQ MAIN+527  Expanded RAM write sector pointer
3210 RDSECT         .EQ MAIN+528  Expanded RAM read sector pointer
3220 SELECT.LOC     .EQ MAIN+635  Address of RAM bank select table (word)
3230 SUBBANK.LOC    .EQ MAIN+637  Address of RAM subbank select table (word)
3240 MEM1.STAT      .EQ MAIN+639  MEMLOCK1 status (00=not in memory)
3250 MEM2.STAT      .EQ MAIN+640  MEMLOCK2 status (00=not in memory)
3260
3270
3280 * ---------------------------------- *
3290 * RAM CARD ROUTINES IN MAIN PROG BASE *
3300 * ---------------------------------- *
3310
3320 * No routines can be accessed until a call to RAMRAM has been performed.
3330 * The location of RAMRAM is contained in RAMRAM.LOC, $0300, 816 deimal.
3340
3350 * Apple monitor routines:
3360
3370 HOME    .EQ MAIN      Clr scroll window, cursor to top left (ROM $FC58)
3380 CROUT   .EQ MAIN+6    Output CR by LDA #$8D, then COUT (ROM $FD8E)
3390 CROUT2  .EQ MAIN+9    Output CR (ROM $FC62)
3400 COUT    .EQ MAIN+18   Output a character (ROM $FDED)
3410 CLREOP  .EQ MAIN+72   Clear screen from line CV, column CH (ROM $FC42)
3420 TABV    .EQ MAIN+75   Place cursor at line (ACC) (ROM $FB5B)
3430 GETLN   .EQ MAIN+81   Write PROMPT then read line (ROM $FD6A)
3440 GETLNZ  .EQ MAIN+84   Read line without PROMPT (ROM $FD6F)
3450 IOSAVE  .EQ MAIN+90   Save all registers (ROM $FF4A)
3460 IOREST  .EQ MAIN+93   Restore all registers (ROM $FF3F)
3470 BELL    .EQ MAIN+165  Output a bell tone (ROM $FF3A)
3480 PRBYTE  .EQ MAIN+477  Print a hex byte (ROM $FDDA)
3490 CLREOL  .EQ MAIN+499  Clear line from cursor (ROM $FC9C)
3500 ROMSUB  .EQ MAIN+544  Call monitor routine pointed by MNLOCL
```

```
3510 * Apple I/O routines:
3520
3530 SLOT1   .EQ MAIN+556  JMP initialization code for slot 1 ($C100)
3540 SLOT2   .EQ MAIN+559  Same for slot 2 ($C200)
3550 SLOT3   .EQ MAIN+562  Same for slot 3 ($C300)
3560 SLOT4   .EQ MAIN+565  Same for slot 4 ($C400)
3570 SLOT5   .EQ MAIN+568  Same for slot 5 ($C500)
3580 SLOT6   .EQ MAIN+571  Same for slot 6 ($C600)
3590 SLOT7   .EQ MAIN+574  Same for slot 7 ($C700)
3600
3610 GETIOB       .EQ MAIN+57   Locate input parameter list for DOS ($03E3)
3620 RWTS         .EQ MAIN+60   DOS Read/Write a Track/Sector ($03D9)
3630 FILE.MANAGER .EQ MAIN+547  DOS file manager ($03D6)
3640
3650
3660 * APPLESOFT routines:
3670
3680 LINGET  .EQ MAIN+78    Convert ASCII from TXTPTR into hex at LINNUM
3690 LINPRT  .EQ MAIN+156   Print unsigned word decimal at cursor position
3700 CHRGET  .EQ MAIN+408   Get next character of text at TXTPTR
3710
3710
3720 * Interface I/O and command routines:
3730
3740 OUT1     .EQ MAIN+177   Outputs DATA to Chroma port
3750 EXPOUTT  .EQ MAIN+180   Outputs DATA to Expander port (tries 255 times)
3760 EXPOUTD  .EQ MAIN+426   Outputs DATA to Expander port (tries TRYAGN times)
3770 OUT.PORT .EQ MAIN+444  Call OUT1/EXPOUTT depending on PORT
3780
3790 INPUT1     .EQ MAIN+411  Input DATA from Chroma port (OVFL clear if avail)
3800 INPUT      .EQ MAIN+414  Wait indefinitely for DATA from Chroma port
3810 INPUTXA    .EQ MAIN+429  INPUT1 for Expander port (tries TRYAGN times)
3820 INPUTXW    .EQ MAIN+432  INPUT for Expander port
3830 IN.PORT.B  .EQ MAIN+438  Input DATA from PORT (tries TRYAGN times)
3840
3850 REINIT.CHROMA.PORT    .EQ MAIN+186  Reinit Chroma port (clears MSKTBL)
3860 REINIT.EXPANDER.PORT  .EQ MAIN+189  Reinit Expander port (sets MSKTBL)
3870
3880 GET.ADC.VAL     .EQ MAIN+204   Reads LAST.ADC.VALUE into ACC
3890 GET.FOOTSWITCH  .EQ MAIN+198   UP if Z flag set, otherwise DOWN
3900 GET.SYNCO       .EQ MAIN+480   UP if Z flag set, otherwise DOWN
3910
3920 TIME.INT      .EQ MAIN+577  Time interrupt handler
3930 TIME.INT.RET  .EQ MAIN+736  Return from time interrupt handler
3940
3950 COMP1      .EQ MAIN+420  INPUT, compare DATA with X register
3960 COMP.PORT  .EQ MAIN+423  COMP1 for PORT
3970
3980 RESTORE .EQ MAIN+168  Send/handle Restore command at Chroma port
3990 GTLINK  .EQ MAIN+453  RDPAR 51 from prog 0 at Chroma, get link status
4000 STRSTA  .EQ MAIN+405  Send Chroma STATUS commands, results in STATUSO
4010
4020 GET.PROG.DATA      .EQ MAIN+306  Get prog X from PORT into BUFFER,Y
4030 SEND.PROG.TO.PORT  .EQ MAIN+333  Send prog entry at BUFFER,Y to PORT
```

```
4040
4050
4060 * Sequence management routines:
4070
4080 FPTOSP  .EQ MAIN+96  Transfer file pntrs to RAM pntrs
4090 SPTOFP  .EQ MAIN+99  Transfer RAM pntrs to file pntrs
4100 INCMNT  .EQ MAIN+102 GET1/STORE X times
4110 GET1    .EQ MAIN+105 Get DATA from sequence at SOPTR, update SOPTR
4120 GET2    .EQ MAIN+108 Update SOPTR
4130 DELCMO  .EQ MAIN+111 Call GET1 X times
4140 DELCOM  .EQ MAIN+114 Delete command in ACC from sequence
4150 DELET2  .EQ MAIN+117 Delete command at SOPTR from sequence
4160 FRTLST  .EQ MAIN+120 Get first time value of sequence into LASTML
4170 METRO   .EQ MAIN+150 Sequencer metronome handler
4180 RESOUT  .EQ MAIN+153 Gets commanad at SOPTR, outputs and reSTOREs it
4190 STRBUF  .EQ MAIN+162 Puts ACC into COMBUF, updates BUFPNT
4200 QSTAKO  .EQ MAIN+402 GET1/STORE until EOS is reached
4210 STAKCK  .EQ MAIN+171 If pntrs not equal call QSTAKO, xfer EOS, SPTOFP
4220 CLRACT  .EQ MAIN+174 Clear ACTBLO, ACTBL1 entries to undefined
4230 BUFOUT  .EQ MAIN+192 Outputs and reSTOREs commnd in COMBUF
4240 INSEQ2  .EQ MAIN+360 Compare XY to LASTML, store time com if > TIMCOMP
4250 INSEQ6  .EQ MAIN+363 Store time of XY in sequence, update LASTML
4260 STORE1  .EQ MAIN+366 INPUT, then STORE DATA in sequence, update SIPTR
4270 STORE   .EQ MAIN+369 Store ACC in sequence, update SIPTR
4280 STOREO  .EQ MAIN+372 Store DATA in sequence, update SIPTR
4290 SERCHI  .EQ MAIN+381 Convert track numbers to available instr. numbers
4300 GETPRO  .EQ MAIN+384 Get program number ACC track in DATA
4310 OUTPUT  .EQ MAIN+387 Main sequence output command at SOPTR, reSTORE
4320 NEXCOM  .EQ MAIN+390 OUTPUT without regard for EOS or TIME
4330
4340 GET.TIME.VALUE   .EQ MAIN+399 Get value of time command at SOPTR into XY
4350 SAVE.SOPTR       .EQ MAIN+550  Save SOPTR if looking ahead in sequence
4360 RESTORE.SOPTR    .EQ MAIN+553 Restore SOPTR if looking ahead in sequence
4370 METRO.SET.UP     .EQ MAIN+147 Set metronome to NONE, 4/4, 60 BPM
4380 INIT.FILE.PNTRS  .EQ MAIN+294 Init file pntrs to clean slate seq
4390
4400 CLEAR.SEQUENCE.ASSY  .EQ MAIN+291 Clear seq, if not CLEAN print ASCII
4410
4420
4430 * Disk I/O, file and overlay management routines:
4440
4450 OPEN.AND.POSITON    .EQ MAIN+147  Open STATUS (ACC=file type), pos at start
4460 POSITION.AT.START   .EQ MAIN+330 Position open file at start
4470
4480 WRITE.SET.UP   .EQ MAIN+126  Write set up (length-1 in AX, buffer in ADDRL)
4490 READ.SET.UP    .EQ MAIN+138  Read set up (length in AX, buffer in ADDRL)
4500
4510 CLOSE.ALL.FILES   .EQ MAIN+135  Close all open files
4520 DELETE.FILE       .EQ MAIN+538  Delete file (name in STATUS)
4530
4540 CLOSE.ALL.FILES   .EQ MAIN+135  Close all open files
4550 DELETE.FILE       .EQ MAIN+538  Delete file (name in STATUS)
4560
4570 XFER.SEQNAME.TO.STATUS   .EQ MAIN+318  Set STATUS buffer to sequence name
4580
```

```
4590 FIND.FREE.BUFFER  .EQ MAIN+324  CTDNOFL contains addr. of free DOS buffer
4600 FILE.ERROR        .EQ MAIN+129  Print error (if ERRPRT), close file
4610
4620 PROGRAM.OVERLAY  .EQ MAIN+465  Load OVERLAY (1 to 16)
4630 GET.OVERLAY      .EQ MAIN+468  Load OVERLAY if not already in memory
4640
4650 CALL.OVERLAY.FROM.OVERLAY .EQ MAIN+207  Set OVERLAY to called overlay,
4660 *                                        ASLOCL to called address, and
4670 *                                        the return OVERLAY on the stack.
4680
4690
4700 * Keyboard and Screen routines:
4710
4720 PRESS          .EQ MAIN+12   "Press any key to continue..." PRESSO
4730 PRESSO         .EQ MAIN+15   Wait for KYCHAR on keyboard
4740 GET.NAME       .EQ MAIN+123  Get file/track name in STATUS
4750
4760 DECMAL  .EQ MAIN+63  Print ACC as 3 digit decimal (no leading zeroes)
4770 DECMALO .EQ MAIN+66  Print XA as 5 digit decimal (no leading zeroes)
4780 CNVRT1  .EQ MAIN+159 Convert LINBUF,Y text to LINNUM hex, destroys TXTPTR
4790
4800 PRINT.SEQ.NAME      .EQ MAIN+297  Prints the sequence name at cursor
4810 PRINT.NAME          .EQ MAIN+300  Prints 15 char field (MANSEQL),Y
4820 PRINT.STATUS.BUFFER .EQ MAIN+87   Prints the 15 char STATUS buffer
4830
4840
4850 * System expanded RAM and setup routines:
4860
4870 BANK.INIT .EQ MAIN+580  Init bank/sector pntrs, set int. vectors RAM
4880 READRAM   .EQ MAIN+590  Sector A, bank Y, RD.BUFFER(X=0),WR.BUFFER(X<>0)
4890 WRITERAM  .EQ MAIN+600  Sector A, bank Y, RD.BUFFER(X=0),WR.BUFFER(X<>0)
4900 INITRAM   .EQ MAIN+610  Set RAM bank/sector, SOPTR/SIPTR, load RD.BUFFER
4910
4920 VECTOR.TABLE  .EQ MAIN+632  This is the start of the jump table for
4930 *                           expanded RAM routines, memlocked overlays,
4940 *                           and alternate I/O driver entry points.  Each
4950 *                           JMP section consists of four bytes (JMP, addr,
4960 *                           NOP).  The order is as below:
4970 *
4980 *          (1) TIME.INT            (2) MEMORY.USAGE
4990 *          (3) STORE               (4) STOREO
5000 *          (5) STORE1              (6) GET1
5010 *          (7) GET2                (8) TRACK.DIR
5020 *          (9) SEARCH.Q           (10) DELETE.SEQUENCE
5030 *         (11) GET.SEQUENCE       (12) SAVE.SEQUENCE
5040 *         (13) AUTO.SAVE.ENTRY    (14) PROG.FILE.MANAGE
5050 *         (15) LOGO.AND.CATALOG   (16) CATALOG
5055 *         (17) EXP.IO.INIT        (18) EXP.IO.INT
5060 *         (19) EXPOUT             (20) EXPOUTT
5070 *         (21) EXPOUTD            (22) INPUTXA
5080 *         (23) CHROMA.IO.INIT     (24) CHROMA.IO.INT
5090 *         (25) INPUT1             (26) OUT1
5100 *
5110
5120      .END
```

SYSTEM MEMORY USAGE

The following is a description of the way the Sequencer uses the computer memory address space. The computer itself uses much of the space for things such as the Monitor and BASIC Interpreter ROM, stack, input line buffer, DOS image, screen text, etc. The Sequencer uses many of the computer functions and, therefore, uses this space as well. This section, however, will concentrate on memory used exclusively by the Sequencer. Operation under expanded RAM is slightly different and is detailed in APPENDIX F.

The controlling program of the Sequencer (RECORD TRACK.1) is a BASIC program that resides from $0800 to $2B5B. This program calls assembly language routines that reside in the 16K bank switched RAM card required for the Sequencer System. The actual sequence data resides in non-switched RAM from $2B5C to $936B. The Sequencer also uses a few locations in Page Zero, Page Three and from $936C to $95F4 for variables, tables and bank switching routines.

Bank 1 of the RAM card, which consists of 4096 bytes, is used for ASCII tables and print routines accessed from the assembly language routines, which reside in Bank 2 of the RAM card. The upper 2K or so of RAM in Bank 2 is designated as an overlay area in which any one of many programs can reside (see APPENDIX C).

Page Zero Usage:

The Sequencer shares the use of many of the computer system locations in Page Zero. These are used with a knowledge of the normal uses for those locations. Whenever in question, the Sequencer saves the values in these locations before changing them. For a list of these locations, see APPENDIX C.

Locations reserved exclusively for the Sequencer are the following:

06 DATA - the location used to hold the byte that is transferred to and from the Chroma and Expander ports.

07 IWRMSK - the interrupt mask image used to control interrupts from the Chroma port.

08,09 CTDNOF - a general purpose zero page register.

18,19 SIPTR - the input pointer to actual sequence data.

1A,1B SOPTR - the output pointer to actual sequence data.

1C,1D MANSEQ - the start of the sequence file, 512 bytes before actual sequence data.

These locations, as well as others, are detailed in APPENDIX C.

Page Three Usage:

Page three of the address space is used for Sequencer variables accessed by BASIC. Details of the memory locations and their functions are given in APPENDIX C.

Sequence RAM:

The sequence 'file' resides from $2B5C ((MANSEQ.LOC)) to $936B ((CALLS.LOC)-1) non-switched RAM. The first two pages contain variables and tables that are to be stored with the sequence (see APPENDIX B. Actual sequence data resides in a circular FIFO buffer (queue) starting at location $2D5C ((MANSEQ.LOC)+512) and ending anywhere up to $936B ((CALLS.LOC)-1). The process of playing or recording restores the data at the previous end so, after any operation, the data can straddle the end of the buffer and wrap around to the beginning. The beginning and ending of the actual sequence data is marked with an EOS command (hex 00) and is pointed to by SIPTRL and SOPTRL.

The sequence buffer can easily be relocated and its length defined by changing a few BASIC lines in the file CHROMA.BEGIN, which is the program that is first run upon booting the system. CHROMA.BEGIN sets up a few variables and determines what instruments are on line, then it runs RECORD TRACK.1, the Sequencer controlling program. When the Sequencer loads a sequence from disk, it automatically relocates it and checks to see that there is enough memory. When a sequence is saved to disk, the sections of memory (up to three possible) that contain the sequence are oriented into a continuous NEW TYPE A file. The structure of this file and the details of the variables and tables stored with the sequence are explained in APPENDIX B.

The whole structure of the sequence RAM changes when running under expanded RAM mode. These changes are described in APPENDIX F. Basically, the 512 byte header is relocated and the sequence body is contained in the expansion RAM card. Two 256-byte buffers provide a 'window' into a virtual memory queue. Data is still pointed to by SIPTRL and SOPTRL. However, a dual set of 'Bank' and 'Sector' pointers are also maintained.

Assembly Routines:

Most of the actual work is done by a collection of assembly language routines that reside in the 16K RAM card. The assembly routines are accessed from BASIC by setting a pointer CALNUM and calling a routine BASCAL at (BASCAL.LOC) to switch the RAM and jump to the routine. CALNUM points to the assembly routine address in a table CALLS that resides in non-switched RAM at (CALLS.LOC).

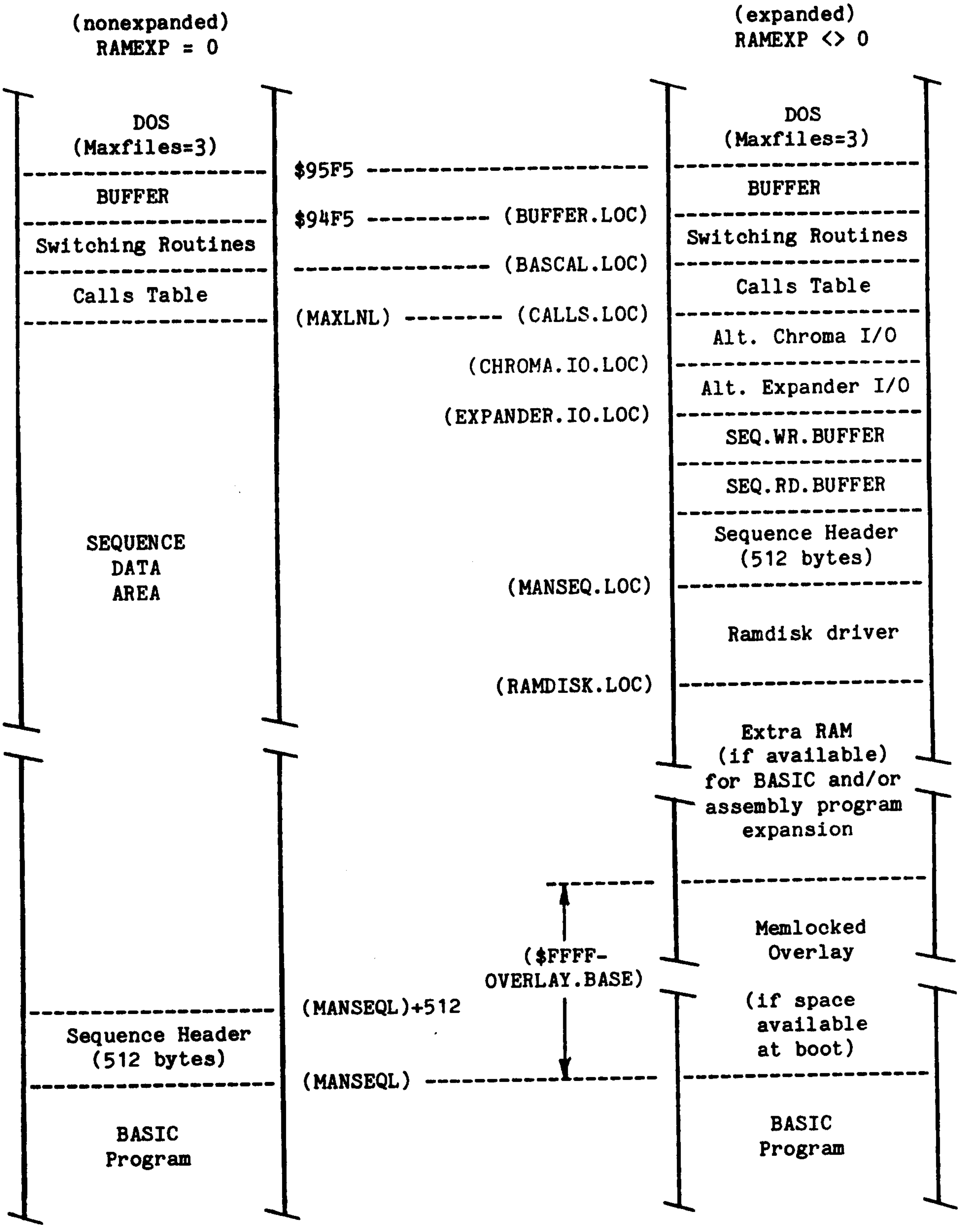
The computer ROM routines (both Monitor and BASIC Interpreter) are accessed from the assembly routines by a software multiplexer in Bank 2 that sets a pointer MNLOC then calls a demultiplexing and bank switching routine at (ROMSUB.LOC) called ROMSUB. Interrupts are handled essentially the same way.

In all of these cases, the status of the RAM card is kept in a register called RAMSTAT and the card is restored just before the RTS or RTI is executed.

Overlay Routines:

For a complete description of Overlay routines, where they reside, how they are accessed, etc. see APPENDIX C, Program Overlay Structure and User Programs.

Sequencer Memory Map



GETTING MORE NOTES OUT OF THE SYSTEM

This section deals with the finite limitions on note storage capacity. An obvious way to increase this capacity is to provide for RAM expansion. This revision of the Sequencer allows for expanded RAM, which is fully explained in APPENDIX F.

It seems that no matter how much memory a system has available for storage, there will be a need for more. It was with this in mind that we have provided two memory saving functions to be performed after most of a sequence has been recorded. One function reduces timing resolution and the other deletes measure commands. Both of these functions remove data that may still be valuable if you intend to record more than about one more track.

^Y-SCRUNCH Reduces timing resolution from two time values (1 time value if using an External or Single Step timer source) to six time values. These are not TIME INCREMENTS, but actual time values. This may result in fewer 3 byte time commands stored in the sequence file.

^X-DELETE MEASURES Removes all 3 byte measure commands and resets the sequence to as if there was never a click track. This eliminates the possibility of setting endpoints and makes it difficult to make the sequence loop in time. This function can only save a few notes worth of memory, so think about what you are giving up for it.

NOTE: ^X - DELETE MEASURES is not displayed in the main menu, but is still available as a selection from any menu page.

Memory Saving Techiques:

The Chroma outputs Performance Control information in duplicate whenever a linked program is used. It doesn't matter if it's link upper, lower or unison. If the linked program does not make use of these performance controls then it is a waste of memory space when you record. To maximize the use of available memory space it is wise to be aware of this situation when it occurs and record the two tracks separately.

The Chroma will output all notes that are pressed, even if the voice is monophonic. In LINK UNISON mode, all attacks, releases and performance control changes are duplicated as instrument 0 and instrument 1 commands. Therefore, if you are recording a LINK UNISON program and the LINK is a monophonic program, you are wasting memory. It would be better to record each track separately and play one note at a time with the monophonic program.

The Sequencer stores a 3 byte time command before an event command if that event occurred more than 2 timer states after the previous event (1 timer state if using an external or single step clock). If you play your chords more precisely and record subsequent tracks more accurately, the Sequencer will use less memory space. In fact, you can use the Editor to remove closely spaced time commands. This has the added benefit of making notes of a chord sound more simultaneous.

The Pressure Sensor Option will work with the Sequencer when it becomes available for your Chroma. The amount of data generated by the pressure sensor is analagous to a separate pitch bend lever for each key. Therefore, memory will be used up very quickly when recording with pressure. It is for this reason that we have elected to keep pressure recording separate from the normal record mode, as indicated in the main menu and Chapter 5.

NOTES ON EXPANDED RAM

This revision of the Sequencer features expandable note storage capacity with the addition of RAM hardware. This Appendix details equipment capability and installation procedures. It also outlines the advantages and disadvantages of expanded RAM and breifly describes the memory structure.

RAM CARD REQUIREMENTS:

Currently only two (2) commercially available RAM cards are compatible with this version of the Sequencer:

Model: Saturn 64K or 128K Card

Manufacturer: Titan Technologies, Inc.
P.O. Box 8050
Ann Arbor, MI 48107
(313) 662-8542

Model: 128K card

Manufacturer: Alphabyte Computer Products
31304 Via Colina
West Lake Village, CA 91362
(213) 706-0333

HOW TO INSTALL EXPANDED RAM:

To install the expanded RAM feature, you must first install the RAM card in the computer according to the card manufacturer's instructions. You should then install the Interface Kit according to Chapter 2. Once these two steps have been performed, you must change the RECONFIGURE setup to use expanded RAM, save the setup under PAGE 3.HEX, then reboot. The following is a detailed procedure for changing and saving the setup to install expanded RAM:

1. Select **X** (RECONFIGURE) from the main menu. After the Sequencer loads that overlay, it will print another small menu. To thoroughly understand RECONFIGURE, you should read Chapter 9. It is not necessary at this time, however, if you carefully follow these instructions.

2. Select **C** (CHANGE SET UP) from this RECONFIGURE menu.

3. Keep typing **<RET>**, which means DEFAULT, until 'EXPANDED RAM? N' is displayed on the screen. At this point, instead of typing **<RET>**, type **Y**. The Sequencer will then ask you to type in the slot number of the RAM card. Type in the slot number then **<RET>**. After that, you should be back at the RECONFIGURE menu.

4. At this point, type **V** for VIEW CURRENT STATUS and check to see that the expanded RAM is in fact set up for the slot you intended. The screen will display '0 BANKS OF 0 SECTORS EACH' because the expanded RAM is not yet installed. After you are satisfied that this is the case, type any key to get back to the RECONFIGURE menu and continue with step 5 below. If you are not satisfied then type any key to get back to the RECONFIGURE menu and goto step 2.

5. Type **S** for SAVE SET UP. When it asks you for the name of the set-up file carefully type **PAGE 3.HEX** then **<RET>**. Make sure that you type a space between PAGE and 3.

6. After this is done, you are back at the RECONFIGURE menu so type **<RET>** to exit into the main menu.

7. Turn OFF the computer and power the system up as described in Chapter 2.

8. When the system is rebooted, the computer should display 'EXPANDED RAM INSTALLED (112K)...' before displaying the main menu. If you have installed a 64K card, the computer should indicate one-half of 112K, or 56K.

POSITIVE EFFECTS OF EXPANDED RAM:

The 64K cards will give you 4000 notes, the 128K cards will give you 8000 notes, as opposed to 1700 notes under nonexpanded mode. These are approximate capacities and the actual note capacity will depend on:

a). the type and number of performance controls (levers, pedals, and especially pressure use a lot of memory),

b). the type of music (single line melodic stuff takes more memory than block chord stuff), and

c). to a slight extent, the number of tracks in the sequence (the more tracks, the greater the possibility of simultaneous events).

As a major side benefit, the disk (see Chapter 7) and track directory editing functions (see Chapter 6) can be accessed quicker because they will be 'locked' into memory. Systems running under nonexpanded mode have to access the diskette for these overlays.

NEGATIVE EFFECTS OF EXPANDED RAM:

An obvious disadvantage of expanded RAM is that you must purchase additional hardware to take advantage of the improvement. This section, however, mentions some of the not-so-obvious disadvantages.

The more memory used by a sequence, the more time it takes to perform operations on the sequence. This becomes quickly apparent when a track directory is performed, since this requires gathering data from the sequence. It also can become a problem when looping between endpoints other than the beginning and ending of a sequence. In addition, the more memory used by a sequence, the longer it takes to save/load the sequence to/from the diskette.

It was neccessary to convert REV 4 sequences to REV 5 sequences indirectly because of the expanded RAM capability. The process of conversion requires an overlay call from the Drive 1 diskette if the software is running under the nonexpanded mode (the overlay is locked in RAM if under expanded mode). The result is that single drive systems running under nonexpanded mode cannot load REV 4 sequences. Effectively, then, this version of the Sequencer almost requires two disk drives.

EXPANDED RAM STRUCTURE:

The expanded RAM sequence structure is essentially the same as that described in APPENDIX B, with the exception of where the data is stored and the physical representation of the sequence queue. This information is present in this Appendix for reference only. It is not necessary that you understand this to operate the Sequencer.

The expanded RAM sequence queue structure is basically that of a simple virtual memory paging system, using the RAM card similar to disk memory.

The sequence queue has been replaced with a similar structure centering around two small buffers in non-switched RAM. One buffer contains up to 256 bytes of sequence data, one byte of which is pointed to by the output pointer. The other also contains up to 256 bytes of sequence data, with the next free space being pointed to by the input pointer. At all times, the pointers point to data in non-switched RAM.

When the buffer boundaries are met, 'sectors' from the RAM card are loaded into the buffer (or stored into the RAM card from the buffer). A bank and sector counter is maintained for both input and output.

The expanded RAM capability is implemented in assembly modules, so that the system could run in nonexpanded mode and future RAM cards could be supported with minimal changes in the code. It is not possible at this time for users to write the drivers and bind them into the Sequencer.

IF YOU NEED SLOT 5 FOR SOMETHING ELSE

Some peripherals for the Apple II are slot dependent, which means they must be installed in a particular slot. The Interface PC card is not such a peripheral. You must, however, first power-up the system with the card in slot 5. If you need slot 5 for something else, take that something else out of slot 5 temporarily and perform the steps outlined in the Installation section (Chapter 2) then do the following:

1. Select **X** (RECONFIGURE) from the main menu. After the Sequencer loads that overlay, it will print another small menu. To thoroughly understand RECONFIGURE, you should read Chapter 9. It is not necessary at this time, however, if you carefully follow these instructions.

2. Select **C** (CHANGE SET UP) from this RECONFIGURE menu.

3. Keep typing **<RET>**, which means DEFAULT, until it tells you that the Interface Slot is 5. Instead of typing **<RET>**, type the number of the slot that you want to use for the Interface PC card then type **<RET>**. After that, keep typing **<RET>** until you get back to the RECONFIGURE menu.

4. At this point, type **V** for VIEW CURRENT STATUS and check to see that the Interface Slot is truly set up for the slot you intended. After you are satisfied that this is the case, notice the disk controller slot number for future reference and type any key to get back to the RECONFIGURE menu. If you are not satisfied then type any key to get back to the RECONFIGURE menu and goto step 2.

5. Type **S** for SAVE SET UP. When it asks you for the name of the set-up file carefully type **PAGE 3.HEX** then **<RET>**. Make sure that you type a space between PAGE and 3.

6. After this is done, you are back at the RECONFIGURE menu so type **<RET>** to exit into the main menu.

7. Turn OFF the computer and install the PC card in the desired slot (see Chapter 2).

8. Power the system up as described in Chapter 2.

BOARDS, CHANNELS, INSTRUMENTS and TRACKS

The Chroma and Sequencer can produce very complex sequences if they are used with a knowledge of the Chroma channel architecture.

It is preferred to view the Chroma as an eight channel synthesizer with provisions for 16 channels within certain limitations. Each of the eight channels has two oscillators, two filters and two amplifiers. Sixteen channels are available only when the program patch parameter is 0.

The Interface Command Set (see APPENDIX I) supports the eight channel concept. The three least significant bits of certain command opcodes denote instruments. Instruments are numbered from 0 to 7, with the Chroma keyboard normally using instrument 0. The keyboard is also assigned instrument 1 if the current program has a link. The way the Sequencer records is that it inputs Instrument 0 and 1 commands from the Chroma and assigns track numbers. These track numbers, in turn, are assigned available instrument numbers when playing the sequence. The sequence can, therefore, play up to eight tracks at a time.

Throughout this manual, tracks and instruments are used interchangably unless noted. Tracks refer to instruments defined by the Sequencer. Track numbers do not necessarily correspond directly to instrument numbers, however.

It is important to understand how the Chroma allocates channels and boards among instruments and what happens when the user plays more notes than can be sounded with the number of channels available for that instrument. Channel and board allocation for each track depends on the number of tracks currently defined and the Keyboard Algorithm and patch parameter of the programs that are defined by the tracks.

Monophonic instruments are always assigned one channel and one board. If the patch parameter for the program is 0, then only one oscillator/filter/amplifier on that board is used. If the patch parameter is non-zero then all hardware on the board is used.

The number of channels and boards assigned to polyphonic instruments depend upon the number available at any instant. For the purposes of initial explanation, let us assume that all programs, monophonic and polyphonic have a patch of non-zero. This is reasonable since most people prefer the sound capabilities of those patch configurations. A single polyphonic track, just as an unlinked passage played from the keyboard, is assigned all eight channels and uses all the hardware on each board. If there are two polyphonic tracks, each is assigned four channels (and four boards). If there are an odd number of polyphonic tracks, then the Chroma assigns channels as evenly as possible but gives priority to the lowest instrument numbers.

In the case of the Sequencer, this means that the tracks that start earlier in time are given more channels if the total number of tracks is odd. For instance, if three polyphonic tracks defined, the first two have three channels and the third one has two channels.

The Sequencer marks the beginning of a track with a define command and the ending with an undefine command. Keep in mind that the first track may not be track #1, since another track can be recorded with its first note occurring before track #1 starts. Instruments are assigned in the order of define command occurrance. Every time a define or undefine command occurs, channels are reallocated. If all tracks are defined as polyphonic, then the number of channels allocated to that instrument are likely to change during reallocation. If a track is monophonic, it will not change because monophonic instruments are always assigned one channel.

This is a good place to explain what happens when the musician or the Sequencer tells the Chroma to play more boards than it has available. If you play the Chroma keyboard you will notice that if the program patch is nonzero and you try to play more than eight notes, then the Chroma will release the first (oldest) note played. If you do the same thing with a program that is linked lower with another program and both programs have a patch parameter of nonzero then you can play only four notes on either side of the split before the Chroma "steals boards." The Chroma will not steal boards between instruments. The same thing occurs with the Sequencer, but the effect is much more pronounced since many more instruments are defined and each instrument ends up with fewer channels. When channels are reallocated and the musician or Sequencer asks for more notes in that track than is possible then the Chroma releases the oldest note in that track.

If you digested the above description of channel allocation, you must realize how important it is to know before hand how many tracks you will record and whether these tracks will be polyphonic or monophonic.

To make the situation a little more complex, consider tracks that are defined as programs that have patch parameters of zero. This means that each note uses one oscillator/filter/amplifier. If the Keyboard Algorithm is monophonic, one half of the board is wasted. If the program is polyphonic and only one instrument is defined, then you can play 16 notes before the Chroma steals boards. If there are three instruments defined and the first is polyphonic with patch 0, the second is polyphonic with patch nonzero and the third is monophonic with patch 0 or nonzero then the first instrument can play eight (two times four) notes, the second can play three and the third can play one. If you play more than eight notes in the first instrument but only one in the second instrument, the Chroma cannot use channels allocated to the second instrument and will release the oldest notes played by the first instrument.

With this mind, one can make a few suggestions about getting maximum utility from the interface system:

(1) Lay down monophonic tracks first. Usually these are the bass or other rhythmic lines. You will have a tendency to lose the beat if your rhythm tracks lose notes due to board stealing. The programs used must be monophonic programs.

(2) Use monophonic tracks as often as possible. This gives you complete control of the board allocation.

(3) If there are long periods within a track that nothing is being played, it is better to stop the recording at that point and record the next segment on another track. This frees the boards for other tracks during the period of silence.

CHROMA COMMAND DESCRIPTIONS

The computer that is connected to the Chroma via the interface cable communicates with the Chroma by sending and receiving commands. A command consists of:

(1) A byte that specifies the command. If the command applies to one of the eight "instruments" within the Chroma, the instrument number will be encoded in this byte, too.

(2) Zero or more bytes that specify parameters of the command. Although most commands require specific numbers of parameters, a few commands are variable in length.

Certain conventions are adhered to in the command language:

Undefined commands are considered to be No Operation commands; that is, undefined commands are ignored. All No Operation commands have no parameters.

Command code zero and command code FF (hex) will always be No Operation commands, even for future instruments that utilize this interface.

Command code 1 will always be an Identification command, for this and any other instrument utilizing this interface.

If a two-byte quantity (such as a memory address) is to be transferred, it will be sent most significant byte first, just the way you would write it on paper.

If a command is variable in length, the second byte of the command will specify the variable number of data bytes. This is not the same as the length of the command, as the count does not include the command code, the length byte, or any other fixed parameters for the command. The Peek command is a good example of this.

If a command is variable in length, the second byte of the command will specify the length as follows: values 1 to 255 represent byte counts of 1 to 255, and a value of zero represents a byte count of 256.

Any command that could conceivably "crash" the Chroma through misuse will not be allowed until a special "unlock" command is first issued. This minimizes the chance of a crash if the Chroma should receive garbage from a malfunctioning computer.

The commands fall roughly into three categories, according to protocol:

> There are those commands that are issued by the controlling device and processed by the Chroma with no response.
>
> There are those commands that are issued by the controlling device and require a specific response from the Chroma. The response will always be a "command" starting with the same code that was received from the controlling device.
>
> There are those commands that establish modes within the Chroma that allow the Chroma to subsequently transmit unsolicited "commands" when certain events occur. The unsolicited commands will generally look like commands from the first group above.

The command set can also be split into two categories, according to destination:

> There are those commands that are addressed to the Chroma as a whole. The lower command codes are assigned to these commands.
>
> There are those commands that are addressed to individual instruments within the Chroma. The higher command codes are assigned to these commands. The three least significant bits of these command codes hold the instrument number.

What follows is a complete description of each command, along with the numerical code (in hexadecimal) for each command byte.

No Operation 00

The only significance of this particular No Operation (as opposed to any of the undefined command codes) is that the Chroma sends this code upon power-up or reset.

Identification 01

The Chroma (or any future instrument) will respond with three bytes, an Identification command, a device code (1 for the Chroma), and a software revision level code.

Read Program 02 pp

The Chroma will respond by transmitting program number pp. The information is transmitted as a Read Program command and 59 data bytes. (If pp is not between 0 and 50, the data bytes will be undefined.)

Write Program 03 pp dd ... dd

The 59 data bytes dd ... dd are written into program number pp in the Chroma. (If pp is not between 0 and 50, the data will be accepted and ignored.)

Load Packet 04

One packet of information is read from the cassette interface, its error detection codes are checked, and the result will be returned via the interface in the form:

04 nn dd ... dd

nn specifies the number of data bytes in the packet, and the dd bytes are the contents of the packet. The first byte of the packet (the first dd byte) is always the packet ID, which identifies the type of packet. The packet ID for valid data is always non-zero. If an error occurs in the reading of the cassette, a special error packet with an ID of 0 is returned.

This command starts reading from the cassette immediately. This can cause a problem if the cassette was previously idle. See the Tape Space command below.

The types of packets that are currently defined, and the forms the Chroma return them in, include:

<u>Error Packet</u> 04 02 00 nn

The length is 2, the ID is 0, and nn will be 0 if a read error is detected or FF hex if the cassette was not running (or was shut off in mid-operation).

<u>Program Packet</u> 04 3C 01 dd ... dd

The length is 60 (3C hex), the ID is 1, and the 59 bytes of data represent a Chroma program.

Program Number Packet 04 02 02 nn

The length is 2, the ID is 2, and the single byte of data consists of a valid program number (0 to 50). This type of packet appears, with a program number of 1, at the beginning of a tape recorded with SAVE ALL.

Stop Packet 04 01 03

The length is 1, the ID is 3, and there is no data in the packet. This type of packet appears at the end of a tape recorded with SAVE ALL.

Save Packet 05 nn dd ... dd

The packet dd ... dd containing nn bytes is written to the cassette. The first dd byte, which is the packet ID must be non-zero. The Chroma will respond when the operation is complete with 05 00 if the operation completes normally or 05 FF if the cassette isn't running.

Read Parameter 06 pp nn

Parameter number nn in program number pp is read and returned in the form 06 vv, where vv is the parameter value. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be undefined.

Write Parameter 07 pp nn vv

Parameter number nn in program number pp is set to value vv. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be ignored. If the vv value is not within the range defined for the parameter, the result is undefined, except that the parameter will never be set to an illegal value.

Panel Switch Off 08

The "panel switch" referred to is the software switch which "connects" the Chroma panel to the interface. When the Chroma receives this, it will echo it and disconnect the panel from the interface.

Panel Switch On 09

When the Chroma receives this, it will echo it and connect the panel to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

> Whenever a program is selected, a Define command will be transmitted for instrument 0 and either a Define or an Undefine command will be transmitted for instrument 1, depending upon the existence of a link.

Whenever a parameter is changed, a Set Parameter command will be transmitted for instrument 0.

Whenever the link balance is varied, Volume commands will be transmitted for instruments 0 and 1.

Performance Switch Off 0A

The "performance switch" referred to is the software switch that "connects" the various performance controls to the interface. When the Chroma receives this command, it echoes it and disconnects the performance controls from the interface.

Performance Switch On 0B

When the Chroma receives this, it will echo it and connect the performance controls to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

Whenever a key is pressed on the keyboard, an Attack command will be transmitted for instrument 0, 1 or both, depending upon the link mode and keyboard split.

Whenever a key is released on the keyboard, a Release command will be transmitted for instrument 0, 1 or both, depending upon the link mode and whether or not an attack had already been sent for the note.

Whenever a lever, pedal or footswitch moves, the appropriate command is transmitted for instrument 0, and for instrument 1 if a link is in effect.

Peek 0C aa aa nn

The Chroma responds by transmitting nn bytes from its internal memory starting at location aaaa. The response is in the form:

0C nn dd ... dd

where the dd bytes are data bytes from ascending addresses.

Peek Two Bytes 0D aa aa

The Chroma responds by transmitting two bytes from its internal memory at locations aaaa and aaaa+1. The response is in the form:

0D dd dd

This command is guaranteed to extract the two bytes concurrently, with no chance that the memory locations could be altered between the transmittal of each byte.

Poke 0E aa aa nn dd ... dd

The nn data bytes dd ... dd are poked into the computer's address space starting at location aaaa. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored.

Poke Two Bytes 0F aa aa dd dd

The two data bytes dd dd are poked into the computer's address space in locations aaaa and aaaa+1, respectively. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored. This command is guaranteed to poke the two bytes concurrently, without danger of the computer utilizing half of the old contents and half of the new contents.

Tap Panel 10

The panel tapper is triggered, unless it has been disabled.

Unlock 11 00 FF

This sequence must be transmitted in order to enable the Poke and Poke Two Bytes commands. If FF is transmitted before 00, the Poke and Poke Two Bytes commands are disabled.

Tape Space 12

The cassette motor will be run for two seconds. Upon completion, the Chroma will respond with 12 00 if the cassette was running, or 12 FF if it was shut off.

The purpose of this command is to allow startup time before other cassette operations. If a sequence of Save Packet commands are to be issued, they should be preceded by **two** Tape Space commands. In addition, if the packets are to be individually readable, they should be separated by two Tape Space commands. A **single** Tape Space command should be issued prior to a sequence of Load Packet commands.

Restore 13

The Chroma is restored to the state reflected by its panel settings. All instruments are undefined except instrument 0 and possibly 1, which are set up according to the currently selected program. The panel switch, performance switch and pressure switch are turned off, and a Panel Switch Off, Performance Switch Off and Pressure Switch Off command are echoed, in that order.

Pressure Switch Off 14

The "pressure switch" referred to is the software switch that "connects" the keyboard pressure sensors to the interface. When the Chroma receives this command, it echoes it and disconnects the pressure sensors from the interface.

Pressure Switch On 15

When the Chroma receives this, it will echo it and connect the pressure sensors to the interface. While this mode is in effect, the Chroma will send Pressure commands for instrument 0 and/or 1 whenever the pressure on a key is varied.

Pressure 68+i kk pp

Instrument i is told to set the key pressure input for note kk to value pp. The pressure is an unsigned number from 0 to 63.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the pressure switch is on and the measured pressure on a depressed key changes. Pressure commands only occur between the corresponding Attack and Release commands for the same note.

Information 70+i

The Chroma responds by echoing the command and sending four information bytes. Currently, only the first byte is utilized, and contains the number of channel boards assigned to instrument i. The other three bytes are zero.

Volume 78+i vv

The Chroma sets the volume of instrument i to vv. The value vv is a linear control from 0 to 255, and is nominally 255. Thus, to reduce the volume of an instrument 6db, the correct vv value would be 128.

This command will be transmitted (for instruments 0 and 1) by the Chroma if the panel switch is on and the Link Balance parameter is varied.

Lever 1	80+i vv
Lever 2	88+i vv

The Chroma sets the value of the appropriate lever input on instrument i to vv, where vv is a signed 2's complement byte in the range -128 to +127. This range corresponds to the mechanical range from "pull" to "push", with 0 corresponding to "at rest".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a lever.

Pedal 1	90+i vv
Pedal 2	98+i vv

The Chroma sets the value of the appropriate pedal input on instrument i to vv, where vv is a number in the range 0 to 255. This range corresponds to the mechanical range from "heel" to "toe".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a pedal.

Footswitch 1 Down	A0+i
Footswitch 1 Up	A8+i
Footswitch 2 Down	B0+i
Footswitch 2 Up	B8+i

These commands activate or deactivate the footswitch functions on instrument i.

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer presses or releases either footswitch.

Define	C0+i pp aa bb cc dd ee ff

Instrument i is defined according to program pp (which must be in the range 0 to 50). The remaining bytes specify initial values for the performance inputs:

aa: lever 1 bb: lever 2
cc: pedal 1 dd: pedal 2
ee: volume ff: footswitches

The footswitch byte uses the most significant bit to represent footswitch 1 and the next most significant bit to represent footswitch 2. A 0 means up, 1 means down. If pp is not between 0 and 50, the Chroma will not define the instrument according to garbage data, but nothing more is promised.

This command causes channel boards to be reallocated as fairly as possible among defined instruments. If this command requires that one or more channel boards be robbed from another instrument, the computer will be kind enough to try and pick boards that aren't currently sounding (see APPENDIX H).

This command will be transmitted (for instrument 0 and possibly 1) by the Chroma if the panel switch is on and the performer selects a program or a link. Although instrument 0 is internally defined by program 0, the Define command that is transmitted whenever a program is selected includes the current program number as shown in the 2-digit display.

Undefine C8+i

Instrument i is removed from operation, and any channel boards assigned to it are redistributed among any other instruments.

This command will be transmitted for instrument 1 by the Chroma if the panel switch is on and an unlinked program is selected or a link is cleared.

Attack D0+i kk vv pp

Instrument i is told to attack note kk with a velocity vv and an initial pressure pp. The key number is a signed, 2's complement byte that must be in the range -64 to +63. The Chroma's keyboard has a range from -32 to +32, with 0 being middle C. The velocity must be a number from 0 (softest strike) to 31 (hardest strike), and the pressure must be a number from 0 (no pressure) to 63 (full pressure).

The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer presses a key.

Release D8+i kk vv

Instrument i is told to release note kk with a velocity vv. The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer releases a key.

Set Parameter E0+i nn vv

Instrument i temporarily sets parameter nn to value vv. This does not affect the setting stored in non-volatile memory, which means that it won't affect other instruments defined according to the same program and it won't affect this instrument if it is redefined according to the same program. Only those parameters that pertain to the tone generation may be set with this command. These include:

1 through 5: control parameters
6 through 50: A parameters
55 through 100: B parameters

Any other parameter number will cause the command to be ignored. If vv is not within the valid range for the selected parameter, the only guarantee is that the parameter will not be set to an illegal value.

This command will be transmitted for instrument 0 by the Chroma if the panel switch is on and the performer varies one of the parameters.

Status E8+i

This command causes the Chroma to respond with:

```
E8+i pp aa bb cc dd ee ff
```

where the seven parameters represent the same quantities as the parameters of the Define command. If the instrument is undefined, the program number returned will be FF and the remaining bytes will be undefined. If the program number is 0, the program number in the display will be used instead.

Squelch F0+i kk

Any channels in instrument i that are assigned to key k are squelched by setting their envelopes to 0. This doesn't affect the channel assignment tables. Even latched channels may be squelched. If kk is -128 (80 hex) all channels will be squelched.

Set-Split 46 F8

This command is sent by the Chroma whenever the user does a Set-Split 46 on the Chroma's panel.

Set-Split 47 F9

This command is sent by the Chroma whenever the user does a Set-Split 47 on the Chroma's panel.

Set-Split 48 FA

This command is sent by the Chroma whenever the user does a Set-Split 48 on the Chroma's panel.

Set-Split 49 FB

This command is sent by the Chroma whenever the user does a Set-Split 49 on the Chroma's panel.

HARDWARE LOCATIONS AND THEIR FUNCTIONS

NOTE: assumes slot 5 for addresses

Decimal	Hex	Operation	Name	Description
-16176	CODO	Write	DISEXTC	Disable the External clock
		Read	ENEXTC	Enable the External clock
-16175	COD1	Write	------	Not used
		Read	------	Not used
-16174	COD2	Write	STRADC	Start the ADC conversion
		Read	RDADC	Read the ADC value
-16173	COD3	Write	CLRTIM	Clear the Timer Interrupt
		Read	RDMISC	(7) Time - LO indicates time Int.
				(6) XXFULL - LO= expander data still on port.
				(5) FOOTSW - up(NO) = HI
				(4) Conversion Ready = LO
-16172	COD4	Write	CLICK	Output HP filtered TTL click
		Read	------	Not used
-16171	COD5	Write	XWRMSK	Write external interrupt mask: XIMASK NOT (7), XOMASK NOT (6).
		Read	RDFLAG	(7) XIFULL NOT
				(6) XOFULL
				(5) SYNC IN (open=HI)
				(4)IXFULL NOT (Expander Input Available)
-16170	COD6	Write	WREXTO	Chroma output port
		Read	RDEXTX	Expander input port
-16169	COD7	Write	WREXTX	Expander output port
		Read	RDEXTI	Chroma input port

COD8-CODF NOT DECODED

SOFTWARE ANOMALIES

This section outlines the differences in behavior between Chromas, Expanders and Sequencer software diskettes of different revision levels. It also touches on software bugs and the bug reporting procedure that we would like you to follow.

If you discover a software anomaly in the Chroma or the Sequencer software, or if you have a comment or suggestion, please contact Fender Musical Instruments. We prefer that you make the bug report in writing, since this will maximize the chance that a fix or suggestion will be included in subsequent revisions. Many worthwhile features of our products were suggested by our customers.

Chroma and Expander Software Revisions:

Please note that the Interface Revision Number (IRN) is returned in response to the Identification command and 'that this number is not the same as the Software Revision Number (SRN) as imprinted on the EPROMS inside the instrument.

The SRN revision levels are described in reverse order, starting with the current revision. Each revision description outlines the differences between that revision and the subsequent revision above it.

SRN 14 (IRN 3) -- This is the current revision as described in this manual. All bugs described below have been fixed.

Commands relating to pressure are now fully implemented.

The Unlock command will disable the Poke and Poke 2 Bytes commands if the last two bytes of the command are sent in the wrong order (FF 00).

In addition, four (4) new commands relating to Set-Split functions have been added:

Set-Split 46 F8

This command is sent by the Chroma whenever the user does a Set-Split 46 on the Chroma's panel.

Set-Split 47 F9

This command is sent by the Chroma whenever the user does a Set-Split 47 on the Chroma's panel.

Set-Split 48 FA

This command is sent by the Chroma whenever the user does a Set-Split 48 on the Chroma's panel.

Set-Split 49 FB

This command is sent by the Chroma whenever the user does a Set-Split 49 on the Chroma's panel.

SRN 13 (IRN 3) -- This is an internal release only.

SRN 12 (IRN 2) -- In this revision, the Unlock command works fine, but there is no way to Lock after Unlocking. Also, this revision did not include the pressure sensor commands. This results in the following restrictions:

The Pressure Switch Off and Pressure Switch On commands are treated as No Operations. They are not echoed.

The Restore command does not echo a Pressure Switch Off command.

The pressure byte in all Attack commands sent by the Chroma is 0. The pressure byte in all Attack commands received by the Chroma is ignored (although it must be present).

Two bugs were found in this revision:

If a link is in effect and a lever or pedal is moved, the Chroma will not transmit an instrument 0 and an instrument 1 command. Instead, the Chroma will send two identical instrument 0 commands. This only applies to the Lever 0, Lever 1, Pedal 0 and Pedal 1 commands.

Varying the link balance control when the panel switch is on will cause the Chroma to send volume commands for both instrument0 and instrument1 to indicate the absolute volume relationship between the instruments. The Chroma does send both volume commands but the value is always the same and that value is random bearing no relationship to the actual volumes of the instruments.

SRN 11 (IRN 1) -- This is an internal release only.

SRN 10 (IRN 1) -- A number of bugs were found in this initial field release revision:

> If a Footswitch command is sent to any instrument that has never been defined since power-up, it will crash the Chroma.
>
> The Restore command does nothing to instrument 1, regardless of the link.
>
> Although the Restore command turns off the panel and performance switch, it does not echo the Panel and Performance Switch Off commands.
>
> Bytes coming from the Chroma occasionally get rearranged and are transmitted out of sequence. This only occurrs if the external computer makes the Chroma wait more than 100usec or so, and the Chroma starts to use its output queue. This problem is more visible at high data rates.

Sequencer Handling of Chroma Bugs:

The Sequencer software is designed to work with instruments that have IRN's of 1, 2, or 3. The earlier reisions, however, have serious problems that cannot be overcome from the external computer end.

The Sequencer gets around the pressure recognition problem by filtering the pressure information to a port that is connected to an instrument of IRN less than 3.

The Restore command handler in the Sequencer can understand and correct the interpretation of Restore commands in <u>all</u> previous Chroma SRN's by looking at the IRN of the instrument on the Chroma port.

The pedal and lever bug in SRN's below 13 will be noticed when you record things like pitch bends or vibrato in a link program. The linked program will fail to be modified when the sequence is played back.

The Sequencer makes no attempt to correct the link balance Volume command bug in SRN's below 13. The user, therefore, will not be able to change the relative volumes of the two tracks by moving the Link Balance parameter during recording.

Upgrading The Instrument Software:

All it takes to bring an instrument up to the current revision is to unplug the EPROMs and plug in new ones. This can be done by any authorized Rhodes Chroma service center, and is free if the instrument is under warranty. Upgrading is strongly recommended, as old software is only old because there was something wrong with it.

Although the Sequencer software will work with IRN's of 1, 2, or 3, Some of the earlier revisions have serious bugs (as outlined above) that cannot be fully corrected from the external computer's end. It is recommended, therefore, that users of the Computer Interface Kit upgrade their Chromas with REV 14 EPROMs.

To request an upgrade from a service center, always refer to the SRN which is printed on the EPROMs, not the IRN. Most service centers are not aware of IRN's.

The current revision includes provision for the Pressure Sensor Option. This does not mean that the Pressure Sensor must be installed. The Chroma will respond to Pressure commands whether or not the option is installed. It just won't generate correct Pressure commands.

Sequencer Software Revisons:

The Sequencer revision levels are described in reverse order, starting with the current revision. Each revision description outlines the differences between that revision and the subsequent revision above it.

Rev 5 -- This is the current revision as specified in this manual. All bugs found in Rev 4 were fixed.

This revision provides an expanded note capacity of 8000 notes with the addition of RAM hardware. Nonexpanded note capacity dropped to approximately 1700 notes.

An externally syncable arpeggiator is included (see Chapter 10).

Parameter display of the current Chroma program is provided by a User Utility overlay (see Chapter 11).

Multiple instruments are available from the Chroma keyboard (see Chapter 11).

The disk catalog is easier to use (see Chapter 7). Basically, the catalog displays only those files in which the user is interested at the time of catalog invokation. The catalog can continue to be displayed while the user inputs the file name.

There are less questions to answer in RECORD mode (see Chapter 4). The name of the sequence is asked only when the user wants to save the sequence. The track name is automatically set to 'Tn'. The user can change the name as always using **^N** - RENAME TRACK.

An abort capability when entering RECORD mode has been provided (see Chapter 4).

Rev 4 -- This revison had a nonexpandable note capacity of around 1800 notes.

It handled pressure information and allowed recording pressure from the Chroma port.

It was compatible with the Apple IIe as well as the II Plus.

External syncing was simplified by an automatically invoked User Utility that inserted a TIME 0 command at the end of recording the first track of an externally synced sequence.

There were many additional features in the H menu selection (see Chapter 11), including programmable transposition presets.

There were a number of bugs found in this revision:

> When the first track of a sequence was zapped, further recording would produce a 'second' sequence. The first sequence was sometimes empty. The user experienced this as unpredictable play operations. Sometimes a SYNC error or DEF error would occur, sometimes the sequence would alternate between playing properly and coming back to the main menu without playing the sequence. It was, in fact, playing an empty sequence.
>
> When an external clock was used during recording and a memory full condition was reached, the Sequencer stored a TIME 0 command where there was no memory space. The result was lost sequence data.
>
> Loop mode recording did not allow the use of a computer key to initiate the 'cued' mode. The result was that if the user pressed a key in 'looping' mode, the recording was terminated. If the footswitch was turned OFF in RECONFIGURE, there was no way to record in loop mode.
>
> In the Interface Test program, the 'EXPANDER NOT RESPONDING' message read '2 DOWN' which doesn't make sense.
>
> Various I/O routine bugs were found involving queue handling. These errors surfaced while experimenting with instruments much slower than the Chroma.
>
> The Editor set the endpoints to the beginning and ending, without restoring them upon exit to the main menu.
>
> When inserting MEASURE commands in the Editor, the internal measure registers where not updated to reflect the changes. The result was that the user could not always reference a MEASURE inserted by the Editor as an ENDPOINT.

> The Editor did not trap certain illegal operations on END OF SEQUENCE and BEG OF SEQUENCE commands. It was possible to destroy sequence pointers, thereby losing sequence data.

Rev 3 -- This was the initial field release version of the Sequencer. The version worked only an Apple II Plus and did not support pressure information.

A couple of bugs were found in this revision:

> The LOAD SETUP operation in RECONFIGURE set the disk slot to 7. If the disk controller was actually in a different slot, the system would hang on disk accesses after loading a setup.
>
> The EDIT TRACK LIST in the Editor was cleared upon entry to the Editor, requiring the user to reenter the track list.

Upgrading The Sequencer Software:

When the customer purchases the Chroma Computer Interface Kit, he is required to register his purchase by returning the enclosed Warranty Registration Card. We use this card to obtain the addresses when we mail revision update notices. _It is important that you return the Warranty Registration Card or you will not be notified of updates._

The cost of the update or upgrade will be included in the revision notice. We try to hold revision costs to a minimum. If you purchase old software within specified time periods of an update, you will get a discount on the update. The time schedule is also included in the update notice.

An update usually consists of a Drive 1 diskette and any changes in the manual text. Since the manual is contained in a loose leaf notebook it is rare that the whole manual be replaced. It is even more unusual to include a Drive 2 diskette in an update.

THE INTERFACE TEST PROGRAM

The USER UTILITY 20 program overlay is a small test program designed to allow the user to determine that his hardware is functioning properly.

To access the program, type **^V** for the USER UTILITY bank 2, then type **0** and **<RET>** for user number 0. The program will then be loaded and an initialization procedure will be ran. The display will indicate the status of the two ports. If the Sequencer tells you that either port is not responding then you either have a problem with that port or no instrument is connected.

In the case of a port error, the error code number is displayed on the left of the message line. The following values are possible:

1 - The Chroma port is not responding to 260 NOPs. This means that the port is not responding to output. The cable is disconnected or improperly functioning. The handshaking flip-flop (Z11) should also be suspected, as well as Z12F or Z14B.

2 - The Expander port is not responding to 260 NOPs. This means that the port is not responding to output. The cable is disconnected or improperly functioning. The handshaking flip-flop (Z1) or Z14A should also be suspected.

3 - The Chroma port is not responding to input requests. The cable is disconnected or improperly functioning. Also, suspect Z12D, Z14B, or the input interrupt logic circuitry.

4 - The Expander port is not responding to input requests. The cable is disconnected or improperly functioning. Also, suspect Z14B.

5 - The Chroma port is not responding to output, the queue is full. Essentially the same as 1 above.

6 - The Expander port is not responding to output. Essentially the same as 2 above.

7 - The Chroma port is not echoing the ID command. The instrument on the port is confused. This is usually caused by a bad cable or the data lines shorted, open, or crossed.

8 - The Expander port is not echoing the ID command. The instrument on the port is confused. This is usually caused by a bad cable or the data lines shorted, open, or crossed.

The Tester is a command oriented program. Below is a short description of each command and the subcommands allowed within each operation:

^I - REINITIALIZE

This command will attempt to send 260 NOPS to the Chroma and Expander port. Will initialize the output buffer for the OUTPUT commands described below. Will display the status of the two ports.

^T - TIMER TEST

This command will initialize the timer registers to zero and display the counts as they occurr. The timer source (internal, external, or single step) and timer increment will reflect the current set-up. Typing a **<SPACE>** will pause the display. Typing an **R** will reset the timer registers to zero. Typing a **<RET>** will terminate the test. When using an internal or high frequency external (1 KHz or higher) timer source, a number that is missing in the sequence of values may or may not indicate a missing timer pulse.

^A - ADC TEST

This command will read the ADC and display the values. This is useful for setting the range trimpots on the PC board. Typing a **<SPACE>** will pause the display. Typing a **<RET>** will terminate the test.

^K - CLICK TEST

This command will output a click of 4/4 time at approximately 120 BPM. Typing a **<SPACE>** will pause the output sequence. Typing a **<RET>** will terminate the test. This test takes TIMSRC and TIMINC into consideration in determining the final output frequency of the click.

^F - FOOTSW TEST

This command will display the current state of the footswitch and any change in that state. Will reflect the FOOTCK register in the current Reconfigure set-up. Typing **<RET>** will terminate the test. The footswitch input is not debounced on release, so you may see UP DOWN UP when you release the footswitch during this test.

^S - SYNC CHECK

This command will display the current state of the SYNC input and any change in that state. This will not reflect the current Reconfigure set-up. In other words, this test will always check the SYNC input and will only wait for one state change. Typing **<RET>** will terminate the test. Since the SYNC input is not debounced, mutiple states may be displayed when a footswitch is used.

^C - CHROMA TEST

This command will output bytes to the Chroma port and display the data received from the Chroma in response to the output bytes. The sub-commands allowed are:

^D Delete last entry
^A Abort output
^M (**<RET>**) output the buffer
any hex number (2 ascii chars) will
    be put in the output buffer
    (256 bytes maximum)

Good tests for the Chroma port are outputting the following commands (and inspecting the response):

```
OUTPUT -      ID : 01
INPUT  -         : 01 01 (Chroma) 02 (REV 2)

OUTPUT - WRPRG0  : 03 00 11 09 18 0F D7 F0
                   0A 00 F8 13 01 00 B0 7B
                   0C 00 FC FE 01 90 3D 80
                   C2 B5 89 97 36 00 00 01
                   C3 08 04 0F CB F0 00 00
                   18 0B 01 00 F8 7B 0C 00
                   7D 7E 01 90 3D 80 C2 AB
                   87 97 36 00 00
INPUT  -         : No response from Chroma

OUTPUT - RDPRG0  : 02 00 (Read Prog 0)
INPUT  -         : 02 11 09 18 0F D7 F0 0A
                   00 F8 13 01 00 B0 7B 0C
                   00 FC FE 01 90 3D 80 C2
                   B5 89 97 36 00 00 01 C3
                   08 04 0F CB F0 00 00 18
                   0B 01 00 F8 7B 0C 00 7D
                   7E 01 90 3D 80 C2 AB 87
                   97 36 00 00
```

^X - EXPANDER TEST

This command is the same as above, but will test the Exander port.

^E - EXIT TEST

This command will exit the Interface Test Program and return to the main menu.

COPYING DISKETTES

It is occassionally necessary to copy diskettes for backups, etc. It is principally for this reason that we have not copy protected the disks in the Sequencer system. We, in fact, encourage you to immediately make a back-up of the diskettes and use the back-ups. Store the originals in a safe, cool and dry place.

We recommend that you use the FID program (on your DOS 3.3 System Master diskette) instead of the normal COPY program for making these copies. For some reason the COPY program will not always work with the assembly source, program group and sequence files. The FID utility has no problem with these files.

MAKING NEW SEQUENCE DATA DISKETTES:

The sequence data diskettes for 2 drive systems are very easy to make. Creating a new data diskette for a single drive system is relatively more involved. The procedures required are outlined below:

Dual Drive System:

(1) Exit into Applesoft by typing **<ESC>** from the main menu.

(2) Type **LOAD HELLO,D2** then **<RET>**.

(3) Take the original diskette out of drive two and replace it with the new unformatted diskette.

(4) Type **INIT HELLO,D2** then **<RET>**.

(5) LABEL the new diskette.

Single Drive System:

(1) Exit into Applesoft by typing **<ESC>** from the main menu.

(2) Type **LOAD HELLO** then **<RET>**.

(3) Take the original diskette out of our drive and replace it with the new unformatted diskette.

(4) Type **INIT HELLO** then **<RET>**.

(5) Take the new diskette out of the drive and label it.

(6) Insert a 3.3 Master Diskette in the drive and type **BRUN FID** then **<RET>**.

(7) Select FID menu item 1 (copy files) and specify the source and destination slot and drive to your disk controller slot and drive 1. Set filename to = (all files). Set prompt to no prompting. Follow instructions from here, inserting source disk (old disk) then destination disk (new disk) until the whole disk is copied. You will need to tell the computer to write over the HELLO program that already exists on the new diskette. We suggest you read the Apple DOS Manual before you use FID.

(8) Put the new diskette in your drive and boot the system by typing **PR#6** (or whatever your slot number is) then **<RET>**.

(9) When you are in the main menu, type **D** for delete sequence and delete the sequences you do not want on this new diskette.

(10) Type **Q** for Program File Management, then select **3** to delete any program groups you do not want on this new diskette.

USING OTHER SYSTEMS WHILE THIS CARD IS IN PLACE

The way the hardware in the Interface PC board is currently set up may cause problems with other cards that use or inadvertently enable the interrupts. The Interface PC card never stops generating interrupt requests if the card is running on the internal clock. Whether the card operates from the external or the internal clock is random upon power-up. If the Interface card is installed and another system is booted that uses or enables the interrupts, our card can impede proper operation of that system. We recommend that you either remove the Interface PC card when not running our system or perform the following instructions before booting the other system:

(1) Turn on the Apple with the Master Diskette or other diskette that does not enable the interrupts.

(2) Remove any External Clock input from the Interface Connector Chassis.

(3) From Applesoft, type **PRINT PEEK (-16176) <RET>** then **POKE -16173,0 <RET>**. If you are using a slot other that 5, you must recalculate these addresses. If the slot is below 5, add 16 times the difference between 5 and the slot you are using. If the slot is above 5, subtract 16 times the difference between 5 and the slot you are using.

(4) THEN boot the system you wish to run.

This enables the EXTERNAL CLOCK and clears the interrupt request line. If no clock signal is inputted then interrupts will not ocurr from the Interface PC card.

The user could modify the HELLO program in the system he wants to run to do this before it enabled the interrupts. The Apple powers up with interrupts disabled.

It is also possible to insert a SPST switch in series with the IRQ line on the Interface PC board (the collector of Q1). The switch could be mounted in the extra jack cutout in the connector box assembly. Performing the modification will void your warranty unless performed by an authorized Rhodes Chroma Service Center.

HARDWARE DESCRIPTION AND SPECIFICATIONS

A Schematic and component layout drawing of the Interface PC board is included in this section for your reference. It is included solely for the purposes of explanation and is not to be used for servicing your own board. Refer servicing to an authorized Rhodes Chroma Service Center or call the Rhodes Chroma Service Department.

Also, please obey the law regarding copyrights!!!

THE INTERFACE PC CARD:

The hardware of the Interface PC Board is divided into the following sections:

(1) I/O Address Decoder
(2) Chroma Port
(3) Expander Port
(4) Interrupt Control
(5) Click Output
(6) Clock Circuit
(7) Analog to Digital Converter
(8) Status Input Port

Interconnection is provided by:

- The Apple Slot edge connector, which allows communication with the Apple and supplies the power (approximately 160 milliamps from +5V, 60 milliamps from +12V and 40 milliamps from -12V). The slot connector pin numbers are represented by small rectangular blocks on the Schematic Diagram.

- The 8 pin phono jack connector (J7), which supplies the Click output and the Pedal, Footswitch, Sync and External Clock inputs.

- The two 26-pin port connectors J5, the Chroma port, and J6, the Expander port.

I/O ADDRESS DECODER:

The address decoder circuitry uses the DEVICE select, R/W, Phase 1 clock and three least significant address lines to decode the addresses of the various I/O functions of the Interface Card. The software locations of these functions are detailed in APPENDIX J and will not be repeated here. The outputs of Z10 and Z15 are active low enable pulses that are used to latch data and/or enable the reading of data from tri-state buffers. They are also used to provide control pulses for clearing the interrupt, starting the ADC, selecting the timer source and outputting a click track pulse.

CHROMA PORT:

The Chroma port consists of an 8-bit input port with handshaking and an 8-bit output port with handshaking. The port connects to a mirror image of itself in the Chroma; that is, each input port line (mnemonics starting with XI) connect to the corresponding output port line (mnemonics starting with XO) at the other end of the interface. The interface really only consists of the output latch (Z5), the input tri-state driver (Z7), the four NAND gates and an inverter. The two transistors Q3 and Q4 are for isolation when the power is shut off. All he other stuff is just for noise rejection. When the Sequencer wants to transmit a byte of data, it checks the XO FULL line by reading the Status Input Port to see if the last byte it sent has been received yet. When it has, it writes the byte into the latch with the WR EXTO strobe coming from the I/O Address Decoder. This causes the flip-flop consisting of gates Z11a and Z11b to be set, and pulls the XO FULL line low. This tells the Sequencer that the output port is full (and not to send any more data yet) and tells the other end of the interface that there is fresh data to be had. When the other end reads the data, it will pulse the acknowlege line XO ACK, which resets the flip-flop and resets XO FULL high (inactive) again. This tells the Sequencer that it can send another byte of data.

The input interface performs the other side of the same task. When data arrives from the other end via the XI lines, the XI FULL line will go low, telling the Sequencer that the fresh data has arrived. When it reads it, using the RD EXTI strobe, the acknowlege XI ACK will be pulsed, causing the flip-flop at the other end to be cleared, and causing XI FULL to go high (inactive) again.

The remaining gates Z11d and Z11c are used to allow masking of the XOFULL and XIFULL lines for interrupt control.

INTERRUPT CONTROL:

Normally, the Sequencer is ready to accept data from the interface, and the XI MASK line from Z16b is high (inactive). This means that an incoming byte, which is accompanied by XI FULL going low, will cause the input of Z12c to go low, turning on Q1 and interrupting the Apple. The only times the Sequencer activates XI MASK to prevent input interrupts is if the device at the other end of the interface is sending data faster than the Sequencer can process it.

Normally, the Sequencer has no data to transmit, and if it does, the interface is usually ready for it, as signified by a high (incative) XO FULL. If, however, the Sequencer has data to send and the output port is still full from the previous data transfer, the Sequencer will store the byte of data in a FIFO (first in first out) queue in its memory and set XO MASK high (inactive) from Z16a, thus unmasking output port interrupts. The output port interrupt occurs whenever the device at the other end of the interface gets around to reading the data off the interface and sending back an XO ACK pulse. Then the Sequencer will take time out from whatever it is doing to pull a byte from the end of the FIFO queue and output it. Only when the queue is empty does the main computer mask output interrupts again by setting XO MASK low (active).

Upon power-up, the XOMASK is set low (active) and XIMASK is set high (inactive) by the system RESET line.

Interrupts can also occur from the Clock Circuit. Interrupts from the Clock Circuit cannot be masked, see APPENDIX N.

The Sequencer determines the source of the interrupt by reading the XOFULL, XIFULL and TIME lines from the Status Input Port.

EXPANDER PORT:

The Expander Port works exactly like the Chroma Port except it is polled instead of interrupt driven. The Sequencer determines the state of the port by reading the XXFULL and IXFULL lines from the Status Input Port.

STATUS INPUT PORT:

The Status Input Port consists of two tri-state drivers that buffer the state lines of the interrupt sources as described above. It also allows the Sequencer to read the state of the FOOTSWITCH, SYNC and Analog to Digital Converter. The RC network of R19/20 and C42 provide some debouncing of the FOOTSWITCH.

CLICK OUTPUT:

The Click Output consists of Z17b, R36 pullup and bandpass filter R16/C35/R17. A 100pF capacitor at the Click Out Jack completes the filtering. Everytime the click address is accessed via the I/O Address Decoder, the output of Z17b toggles, generating a pulse at the Click Out Jack. Emphasis is placed on the first beat of the measure by outputting two pulses, 42 microseconds apart, for subsequent beats of the measure. Two pulses this close together reduce the low frequency energy and the pulse sounds lower in volume.

CLOCK CIRCUIT:

The Clock for the Sequencer can be either an internal clock of 1000 Hz or an external clock of any frequency between DC and 3000 Hz. The lower the frequency, the less recording/playing resolution available. Ideal input frequency is 1000 Hz whereas 24 Hz is barely acceptable and frequencies above 3000 Hz will keep the Sequencer in constant interrupt service. For more information about the external clock frequency, see Chapter 10.

The flip/flop consisting of Z1c and Z1d allows switching between internal and external clocks by addressing the location decoded by the I/O Address Decoder.

The internal clock of 1000 Hz is generated by dividing the 1M Hz system clock by 1024 in Z19. The internal clock is disabled by pulling the reset line high, forcing the Q10 output to remain low.

The external clock circuitry consists of Z18b comparator circuit and associated components. This circuit senses when the input voltage crosses zero plus a .05V hysteresis level, at which time the output of Z18b saturates at the negative supply level. When the input falls below -.05V then Z18b switches back to the positive supply level. The circuit is enabled by a low level at the output of Z1c and creates an input voltage divider R47 and R42. When Z1c is high, the input of Z18b will always be high enough to keep it from switching and the output will be negative.

The output of Z18b is rectified, divided and filtered so that it is compatible with the clock input of Z17a (0V to 5V), which generates the interrupt. R48 is necessary to offset the effect of the nonsymmetrical load of Z18b. The interrupt is cleared by a pulse from the I/O Address Decoder when the Sequencer has determined, by reading the Status Input Port, that the interrupt came from the timer.

ANALOG TO DIGITAL CONVERTER:

The ADC circuit is centered around the National 0804 single channel IC. The circuit converts the resistance of a 100K linear potentiometer in the pedal housing to a digital value between 0 and 255.

The reference of the ADC is set by R25. This effectively adjusts the range of digital values obtained. The minus input for the ADC is set by R29. Adjustment of this trimmer sets the zero value with the pedal all the way up. The Z18a circuit converts the pedal resistance to a voltage source with low impedance, which is sent to the ADC for conversion.

Conversion ocurrs within 100 microseconds and the INTR line goes low, telling the Sequencer that a conversion is finished. When the Sequencer reads the ADC value, the RD line goes low and the digital data is sent to the Apple via the Z8 buffer. The ADC conversion process is started again by pulling the WR line low on the 0804, which is done by the pulse from the I/O Address Decoder after the Sequencer reads the value.

It is possible to open circuit R33 and feed a 0-5V control voltage in the pedal input if you desire. Performing the modification will void your warranty unless performed by an authorized Rhodes Chroma Service Center.

ADC ADJUSTMENT PROCEDURE:

To adjust the ADC zero point and range, it is necessary to continuously view the value of the control pedal. You can do this by running USER UTILITY BANK 2, NUMBER 1 (ADC TEST) or USER UTILITY BANK 2, NUMBER 0 (the preferred Interface Test Program). Number 1 displays the value in decimal from 0 to 255 and NUMBER 0 will display the pedal value in HEX from 00 to FF when the **^A** command is issued (see APPENDIX L).

Once the Sequencer is continuously displaying, push the pedal all the way in the up position and adjust trimpot R29 (the one closest to the rear of the APPLE) for 00 display. You should first adjust R29 until you start seeing 01's then back off until it is always 00. Then push the pedal all the way down and adjust trimpot R25 (the one closest to you) for FF (or 255 decimal) display. You should adjust from FE until the display always reads FF. Then push the pedal all the way up again and readjust for 00. Exit the ADC Test mode by typing a **<RET>** and exit the Interface Test Program by typing **^E.**

SYNC, EXTERNAL CLOCK and CLICK SIGNAL SPECIFICATIONS:

<u>SYNC</u> <u>Input</u>:

LEVEL : 0V to 5V, TTL (external device must be able to sink .9 mA @ .8V max, can use open collector because input is pulled up by 10K).

SPEED :Ton (min) = Toff (min) = 52 microseconds in SYNC CHECK mode (1.5 milliseconds in Single Step2 mode).

NOTE : If a footswitch is used for this input, it should be debounced unless sync delay is set to zero. Both single step timer sources debounce this input in software.

<u>EXT</u> <u>CLK</u> <u>Input</u>:

LEVEL : .7Vp-p minimum, 22Vp-p maximum.

INPUT IMPEDANCE : approximately 100K ohms.

WAVEFORM : Sine or Square (duty cycle 25% to 75%).

MINIMUM FREQUENCY : 0 Hz (although for useable resolution, a minimum of 100 Hz is recommended).

MAXIMUM FREQUENCY : depends on complexity of sequence but generally should be limited to 3000 Hz.

INTERNAL TIME INCREMENT : Adjustable from 1/16X to 8X clock frequency in powers of 2. This is used to match external clock as close as possible to internal clock if you want to switch clocks after you record the sequence.

<u>CLICK</u> <u>Output</u>:

OUTPUT LEVEL : -2V TO +2V nominal.

OUTPUT IMPEDANCE : 10K ohms.

USING ASSEMBLY ROUTINES FROM ANOTHER BASIC SYSTEM

This Appendix shows you how to use some of the internal assembly routines for your own BASIC programs.

CHROMA.BEGIN:

To create a turn-key system you must first run CHROMA.BEGIN from drive 1. This is done by the HELLO program on drive 1. CHROMA.BEGIN loads the assembly routines routines and initializes the system.

MODIFICATIONS TO CHROMA.BEGIN:

CHROMA.BEGIN can be modified to display any prompting message by changing lines 140-147. Currently the message is:

```
     **********************
     *                    *
     *  CHROMA SEQUENCER  *
     *                    *
     **********************

        COPYRIGHT 1982
           CBS INC.

PLEASE WAIT WHILE I GET SET UP...
```

Line 415 should be modified to run your program. Currently, it runs RECORD TRACK.1, which is the Sequencer control program.

In line 60, HIMEM: is set to 11100 ($2B5C), which is one location before the start of the sequence RAM. If you do not plan to keep a sequence resident in RAM for your application program, you can set HIMEM: to (CALLS.LOC), which is currently $936C. This will give you approximately 22K more RAM for your BASIC program.

YOUR APPLICATION PROGRAM:

The first thing your program must do is set HIMEM: to the same value that you set in CHROMA.BEGIN. You must then declare the string variable N$ so that the assembly routines know where the variable resides. N$ is a variable used for file and track names.

You must then set BB% (or any other integer variable name) to (BASCAL.LOC). See CHROMA.BEGIN lines 15000-15020 for an example. You should then call REINIT.CHROMA.PORT and REINIT.EXPANDER.PORT as described below.

When exiting your program you must POKE 1008,89: POKE 1009,250: POKE 1022,101: POKE 1023,255. This sets the break vector back to $FA59 and the interrupt request vector back to $FF65, the monitor entry point. It is also a good idea to set the hardware up for an external clock before you do this (see APPENDIX N).

CALLING AN ASSEMBLY PROGRAM:

Before calling an assembly program, the CALNUM (location 877) must be set to a value representing the particular routine. CALNUM need not be reset for each successive call if you are calling the same routine. The data transfer location is decimal 6. Routines that input from or output to the ports or store or get data from the sequence queue use this location for the data byte. Below is a description and the CALNUM of each routine that can be directly called from BASIC:

FPTOSP CALNUM=0

This routine sets the sequence input and output pointers to the values stored in the sequence file. You will probably not use this unless you are manipulating a sequence.

LOGO.AND.CATALOG CALNUM=1

This routine displays a disk catalog as described in Chapter 7.

STOREO CALNUM=2

This routine stores a byte in the sequence queue at the current SIPTR (location 24,25) and updates SIPTR. The source is transfer location 6. If the sequence is full, ERRCOD (location 797) is set to 17, otherwise ERRCOD is not modified. In other words, if you want to check for a sequence full condition, you must set ERRCOD to 0, then check it after calling STOREO. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

GET2 CALNUM=3

This routine increments the SOPTR (location 26,27) if SOURCE (location 793) is zero (seq queue) or the command buffer BUFPNT if SOURCE is non-zero (command buffer). The command buffer is an 8 byte buffer that is not directly addressable from BASIC.

You can use this routine to pull data out of the sequence RAM. To do this, you would first set an integer variable (SO% for example) to the value in SOPTR. Then PEEK (SO%) to get the data. Then call this routine to point to the next byte. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

REINIT.CHROMA.PORT CALNUM=4

This routine reinitializes the Sequencer and the Chroma port. It also interrogates the instrument attached for ID and software revision information. It updates IDNUM (804), REVNUM (806), and sets MSKTBL (885) to zero, signifying one Chroma on line. Upon return, the following ERRCODs are possible:

00 - no error
01 - Chroma not responding to 260 NOPs
03 - Chroma not responding to input request
05 - Chroma not responding to output, the queue is full
07 - Chroma not echoing ID command

SEARCH.Q CALNUM=5

This routine loads the search (TRACK.DIR.HEX) overlay, if it is not in RAM, and calls the SEARCH.OPERATION routine, which performs functions depending on the value of SRCTYP, the search type number. SRCTYP can have any one of the following values:

VALUE	NAME	FUNCTION
00 -	FNDVOL:	cycle through the sequence and put the initial volume values in the Volume Table (starting at $94F5)
01 -	DELTRK:	delete a track then delete all time commands that are followed by another time command
02 -	CHGVOL:	change the initial volume of a track
03 -	TRANS1:	transpose a track or the entire sequence

04 - DELTM: delete time commands that are followed by another time command

05 - CHRNTM: quantizes all time values to within 6 time values

06 - DEMEAS: delete all measure commands, clear any click track information, and set the endpoints to beginning and ending.

07 - MUTE.UNMUTE: allows muting and unmuting of tracks by calling the MUTE/UNMUTE routine

08 - CHANGE.PORTS: allows the user to change the port of a track from Chroma to Expander or vice versa

09 - CHANGE.PROGRAM: allows the user to change the program number of a track

10 - CHANGE.NAME: allows the user to change the name of a track

11 - MES.16: searches the sequence for measure commands, inserts a high byte of 00 for all measures, converting between REV 4 and REV 5 sequences.

PREREC CALNUM=6

This is the record/play routine. Several things have to be determined and/or set up prior to calling this routine. See the following lines in RECORD TRACK.1 for an example:

22-25, 100-153, 163-165

To Play, see lines 230, 500-2150
To Play Along, see lines 233, 500-2150
To Record, see lines 225, 500-2150

A full explanation of the set up required is beyond the scope of this APPENDIX.

MONTOR CALNUM=7

This routine advances the sequence queue until a command block is found that contains the measure command for the measure number in DELMES. If the measure command is not found, ERRCOD is set to 18, otherwise ERRCOD is set to zero (no error). Upon entry, SOPTR must point to the first time command in the sequence queue after the EOS command. When advancing the queue, all DEFINE, UNDEFINE and performance commands are outputted to the ports. ATTACK and RELEASE commands are not outputted. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

OUT1 CALNUM=8

This is the general routine that outputs a byte in transfer location 6 to the Chroma port. If the output queue is empty and the Chroma is not busy then this routine will output directly to the Chroma. Otherwise, it will store the byte in the output queue. If the output queue is full, this routine will keep trying once a msec. for 255msec. The routine will then return with the carry cleared if successful or the carry set if unsuccessful. Unfortunately, there is no way from BASIC for you to know if the data was transferred, since you can not access the carry flag. You could write a small assembly routine that calls this routine and checks the carry flag, setting ERRCOD appropriately. For example:

```
OUTPUT LDA #$00
       STA ERRCOD    CLEAR ERROR CODE
       LDA #$08
       STA CALNUM    SET UP FOR OUT1
       JSR BASCAL    CALL OUT1
       BCC NOERR     ALL IS WELL...
       LDA #$05      CHROMA NOT RESPONDING TO OUTPUT CODE
       STA ERRCOD    SET ERROR CODE REGISTER
NOERR  RTS           RETURN FROM OUTPUT
```

INPUT CALNUM=9

This is the general routine to input from the Chroma port. The routine will wait indefinitely for a byte from the Chroma and return after storing the byte in transfer location 6.

CLICK.SET.UP CALNUM=10

This loads the click set up overlay (CLICK.HEX) if it is not already in RAM, which allows the user to set up and change the click track parameters as described in Chapter 8.

EDITOR CALNUM=11

This loads the editor overlay (EDITOR.HEX) if it is not already in RAM, which allows the user to perform several editing functions on the sequence as described in Chapter 6.

RESTORE CALNUM=12

This routine restores the Chroma instruments according to the program in the LED display. This involves undefining instruments 2-7 and redefining 0 and 1 according to the program. The panel, performance and pressure switches are also turned off. A fatal error (returns back to BASIC without resetting sequence pointers) if the Chroma is not communicating properly (SYNC ERROR).

TRANSFER CALNUM=13

This loads the transfer overlay (TRANSFER.HEX) if it is not already in RAM, which allows direct communication between the Chroma and Expander as described in Chapter 10.

SPTOFP CALNUM=14

This routine sets the sequence file input and output pointers to the values stored in SOPTR and SIPTR. If you move the sequence in any way, you must cycle through the queue until the EOS, then call this routine before you save the sequence. Unless specifically stated, all routines accessable from BASIC take care of SOPTR, SIPTR and the file pointers for you.

SAVE.SEQUENCE CALNUM=15

This is the routine that saves a sequence as described in Chapter 7.

AUTO.SAVE.ENTRY CALNUM=16

This is the routine that performs the AUTOSAVE function as described in Chapter 5.

PROG.FILE.MANAGE CALNUM=17

This is the routine that handles the transfer of program groups from the Chroma and Expander to disk or vice versa as described in Chapter 7.

GET.SEQUENCE CALNUM=18

This is the routine that loads a sequence file from disk as described in Chapter 7.

CLEAR.SEQUENCE.BASIC CALNUM=19

This is the routine that clears a sequence from RAM as described in Chapter 5.

PRINT.LAST.TIME CALNUM=20

This routine prints "LAST TIME: XXXXX" where XXXXX is the decimal value of the register LASTM, which contains the value of the last time command encountered during playback or record.

DELETE.SEQUENCE CALNUM=21

This routine allows the user to delete a sequence file from disk as described in Chapter 7.

METRO.SET.UP CALNUM=22

This routine sets up click track parameters to NONE, 60 BPM and 4/4 time signature.

RECONFIGURE CALNUM=23

This routine loads the reconfigure overlay (SYS.PORT.HEX), if it is not already in RAM, which allows the user to change various operating parameters of the Sequencer as described in Chapter 9.

REINIT.EXPANDER.PORT CALNUM=24

Same as REINIT.CHROMO.PORT but operates on the Expander port. MSKTBL is set to 01 if an Expander is responding. The ERRCOD values returned are described below:

00 - no error
02 - Expander not responding to 260 NOPs
04 - Expander not responding to input request
06 - Expander not responding to output
08 - Expander not echoing ID command

MARKER CALNUM=25

This routine loads an overlay called MARKER.HEX which currently performs the syncable arpeggiation functions described in Chapter 10.

CVOLUME CALNUM=26

This routine loads an overlay called VOLUME.HEX which does not exist at this time. It is reserved for a program that will allow the user to vary the volume of a track continuously by using the control pedal.

MENU.PAGE.3 CALNUM=27

This routine loads the menu overlay (MENU3.HEX), if not already in RAM, which will clear the screen and display page three of the main menu.

TRACK.DIR CALNUM=28

This routine loads the search overlay (TRACK.DIR.HEX), if not already in RAM and calls the Track Directory routine within that overlay. This will clear the screen and display the current track directory.

USERA CALNUM=29

This routine will load and run the USER UTILITY BANK 1 program. Upon entry, you must set USERNO (location 872) to the bank 1 user number as described in APPENDIX C.

USERB CALNUM=30

Same as USERA but will load and run a USER UTILITY BANK 2 program.

ACCESS TO OTHER ASSEMBLY ROUTINES:

Although there is no direct access to the other assembly routines from BASIC, a small assembly program can be written to access any byte in the Sequencer. Essentially all that is required of such a program is to switch the RAM card to RAM BANK 2 read/write and JMP to the location. Upon return from the desired routine, the small program would reset the RAM card to ROM read and return to your BASIC program. 4096 bytes of ASCII messages and print routines reside in the RAM card bank 1.

INTERRUPTS:

The Sequencer card should not be used with hardware and/or software that generates or uses interrupts (IRQ or BRK) or software that inadvertently enables interrupt requests. This also applies to utility programs that generate BASIC or assembly programs that run on the system. See APPENDIX N for an explanation and possible solutions to this kind of problem. It is also possible to change the interrupt vectors to a routine that just does a LDA $45 and a RTI for systems that inadvertently enable the interrupts.

PAGE THREE:

The Sequencer assembly routines use all of page three ($300-$3CF) for a variable table. Your BASIC or assembly routines must not use this space for anything other than controlling the Sequencer routines as prescribed in this APPENDIX. Also, the Sequencer uses the ampersand vector ($03F5 and $03F6) for one of the sequence queue pointers. Therefore, the ampersand feature is not available when running this software.

ZERO PAGE:

There are many zero page locations that are used by the Sequencer assembly routines. Locations used exclusively by the Sequencer are $06-09 and $18-1D. Other locations are used, but these are used with a knowledge of their effects on the Monitor and BASIC interpreter. In other words, it is something that you normally will not care about. If your program (or any utility program you use to write your program) uses $06-09 or $18-1D, you will have to save the data in these locations in both directions (your program to the Sequencer assembly programs and vice versa). Locations $07 and $18-1D are initially set up in CHROMA.BEGIN.

Also, it is important to make sure that interrupts are not enabled or a BRK instruction is not executed in your program. You could execute a BRK if you change the BRK vector, but you must make sure that you restore it before calling any of the Sequencer's assembly routines.