MusicParts.Com

Technical Document Distribution

Brand:	**E-mu Systems**	
Model:	**Emulator III (Three)**	
Product:	**Digital Sound Production System**	
Description:	**Service Manual**	**Dated: 1988**
Musicparts order number: 38490	TechTips: 0	Pages: 187

MusicParts.Com Special Notes

Blank pages have been eliminated to save paper.

E-mu Systems, Inc.
applied magic for the arts

EMULATOR THREE

DIGITAL SOUND PRODUCTION SYSTEM

TECHNICAL MANUAL

BY RILEY B. SMITH

Emulator Three
Digital Sound Production System
Technical Manual

■ AG 200 Rev A

E-mu Systems, Inc.
1600 Green Hills Road
Scotts Valley, California 95066
(408) 438-1921

Important Notice: In order to obtain warranty service on an Emulator III, the serial number sticker on the back panel must be intact, and the customer must have a sales receipt or other proof of purchase. If there is no serial number sticker on an Emulator III, please contact E-mu Systems at once.

TABLE OF CONTENTS

INTRODUCTION

The Emulator Three Digital Sampler is a powerful and complex instrument. E-mu Systems Inc. intends this manual to be an aid to the experienced service technician only.

To service the EIII you should be familiar with the 7400 series of digital logic, floppy disk drives, hard disk drives, ADC's and DAC's, op-amps as well as microprocessor troubleshooting techniques, and current music synthesizer technology. The minimum equipment required to service and repair the EIII is: a digital multimeter, a 100MHz dual trace oscilloscope and basic technician hand tools. Since the circuit boards in the Emulator Three are mounted in a card cage, a board extender (available direct from the factory) would also be extremely helpful.

Because the EIII has so many complex functions, we have included brief Guided Tour type of operating instructions in this manual. This will help to familiarize yourself with the instrument. In addition, we highly recommend that you have a complete EIII reference manual on hand to help you understand all the functions. This is also available directly from the factory.

The information contained in this manual is proprietary to E-mu Systems Inc. The entire manual is protected under copyright and none of it may be reproduced by any means without written permission from E-mu. Please consider all of the data in this manual secret and use it only to service the EIII.

We feel obliged to remind you that any modification of an EIII other than as specified by a factory authorized E-mu Change Order (ECO) will void the warranty of the instrument.

Please read this manual thoroughly before attempting to service the EIII. If you feel unsure about working on the instrument, please feel free to contact our service department.

EIII GUIDED TOURS

The inclusion of the EIII Guided Tours in this service manual is intended to quickly familiarize the technician with the basic operation of the Emulator Three. It is not a substitute for a complete EIII Reference manual. The EIII Reference manual can be purchased directly from E-mu Systems Inc. for a nominal fee.

■ THE EIII MODULAR SYSTEM

The left side of the EIII front panel includes the volume slider, sample input slider, data slider, On/Yes and Off/No (Inc and Dec) buttons, the Liquid Crystal Display, four cursor control buttons, and a numeric keypad. Once you select a module and function, this area is where you will specify parameters.

The modules start towards the right of the keypad. Each module will be discussed in detail later on; the following is intended mostly as background information. Each module affects a certain area of EIII's operation.

Function buttons. These are the buttons that get you going. Load Bank and Enter load disk data into EIII, Save Bank lets you save your work to disk, and Transpose, as you probably suspect, transposes the keyboard.

Sequencer. This module is a 16-track solid state control data recorder. The EIII sequencer is a full function, Segment/Song type sequencer with sophisticated auto-correct functions and the ability to control other instruments via MIDI. It also has some very useful features such as Mono operation when driven via MIDI, ability to re-assign presets, track bounce, and so on. It consists of the Sequencer Setup module (which sets up a sequence for recording), the Sequencer Management module (which lets you load, erase, copy, and do other sequence "housekeeping"), and the Sequencer Edit module (for extremely flexible sequence editing using "Cut and Paste" techniques).

Master. This module contains functions that affect the overall keyboard or bank (memory available, master tune, audio trigger, erase bank, MIDI analyzer, and disk utilities). This also contains the Special commands.

Sample. This "recording studio" module records sounds from the outside world into the bank. Features include adjustable preamp gain, variable threshold setting, and adjustable sample rate and length.

Digital Processing. With this powerful module, you may edit a samples length, loop (i.e. infinitely sustain) any portion of the sample (with several different looping options such as Crossfade Looping), have EIII automatically find the best loop points ("Autocorrelate"), splice two different samples together, mix two samples, amplify or attentuate the sample, and more.

Preset Management. This module handles the preset "housekeeping"—load presets from disk; save presets to disk; create, copy, rename, or erase presets; and check on how much memory space a preset uses.

Preset Definition. This module lets you change parameters within a given preset. Set up the arpeggiator or MIDI options, assign the pitch bend and modulation wheels to control various parameters in real time, copy or erase zones, edit the sample assignment, and set some keyboard parameters (dynamics and crossfade between overlapping samples).

Analog Processing. This module sets parameters that should be familiar to those who have worked with analog synthesizers. Set the dynamics by adjusting the VCA envelope; control timbre using the VCF (filter) and its associated AHDSR envelope generator; modulate the signal with the LFO; tie the keyboard velocity to various parameters; set the tuning, attenuation, and delay for each zone; and more.

Remember—if you want to save modified presets, save the altered bank to disk. Otherwise, any changes will be lost as soon as power to EIII is interrupted or a new bank is loaded.

■ ACTIVATING A MODULE, SELECTING FUNCTIONS, DE-ACTIVATING A MODULE

Here's important background information on how to access the various module functions. As the tours progress, we'll relate this information to practical examples.

Activating: Each module has an associated switch. Most of these are in the right-most section indicated by the blue bars next to the buttons, however, the Sequencer Setup, Sequencer Edit, and Sequencer Manage module buttons are found in the left section indicated by the yellow bars. Pushing the associated switch "activates" the module, as indicated by an LED next to the switch. Upon activation, the display's top line will show the Module Identifier (such as "Master," "Preset Management," etc.). In some cases, upon activation the display will ask you to specify the current zone or current sample (as described in the next guided tour). Once the module is active, it's time to start...

Selecting functions: Each module includes a printed list of functions on the front panel; these functions are available when the module is active. Selecting a module function requires keying in its associated number with the keypad. We will shortly give an example as to how this works.

De-activating: When you're finished with the module, either press its button again to de-activate, or simply activate a new module.
Hint: Any time you make a mistake, get confused, or otherwise get "lost in the module" and need to bail out, simply de-activate the module. Then, re-activate and try again.

GUIDED TOUR #1: MEET EIII

Welcome to the Guided Tours! If you have just met EIII for the first time, follow these tours until you complete the Guided Tours section. This will get you "up and running" on EIII in the fastest possible time. Also, you'll learn some tricks in this section that will come in handy as you play some more with EIII.

This tour covers how to...

- ■ Load a bank from the hard disk
- ■ Select different presets within the bank
- ■ Load and save to floppy disks
- ■ Tune EIII to other instruments
- ■ Transpose the keyboard

...and also discusses EIII's "modular" design philosophy.

■ LOADING A BANK FROM THE HARD DISK

Press the Load Bank button; the display says: Load Bank, and shows the name and number of the current bank. Use the data slider to scroll through the available hard disk banks. Stop when you find the bank you want, then press ENTER.
An alternate method of loading a hard disk bank is to press Load Bank, then simply type in the number of the bank using the numeric keypad.
The display will show the current preset number and name (the cursor will flash underneath the first digit). Start playing the keyboard and adjust the Master Volume slider for a comfortable

■ SELECTING DIFFERENT PRESETS

1. The bank you just loaded contains several presets. To call up a new current preset, use the up/down cursor buttons to place the cursor under the preset number, then use the keypad underneath the display. Note that "leading zeroes" must be entered for preset numbers (i.e. type 0 and 2, not just 2, to call up preset 02). Now type 0 then 2 on the keypad; these will replace the numbers indicated by the flashing cursor.

2. Now call up more presets. If you enter a number for which there is no preset, the display will list the entered preset number and say "Empty Preset"; try again.

3. To scroll through the presets available in the bank, move the data slider. The various preset names will scroll on the lower display line. When this line shows the desired preset, press ENTER to make that the current preset. This is an alternative preset selection method.

4. Yet another method is to use the left and right (<->) cursor buttons to increment or decrement through the presets. This method allows you to arrange your presets in the proper order, then access them sequentially with a single press of a button.

When you're ready to check out some more sounds, proceed.

■ SAVING DATA TO FLOPPY DISK

The floppy disk drive is used to make permanent backups of your work. In preparation for this tour you must have at least five, double-sided, double-density, 3.5" disks on hand. Before a floppy disk can be used by the EIII, it must be formatted using the Format Disk function.

To format a floppy disk:

1. Activate the Master Module; its LED will light. Select Master 7 (Disk Utilities), and select Disk Utilities 7 (Format Disk).

2. Position the cursor under the drive number in line two. The EIII will normally default to the floppy drive. If another drive is listed, select the floppy drive using the data slider and press

3. Insert a floppy disk into the drive and press Yes to format. Keep in mind that formatting a disk will erase all information on the disk.

4. After formatting is complete, the display will ask, "Format Another?". Remove the first disk, insert the next, and press Yes. Continue to format at least four more disks.

To save a bank to floppy disk:

1. Press the button labeled Save Bank, and position the cursor under the drive number in line two. Select the floppy drive using the data slider. Now insert the first floppy disk and press ENTER. The display will say "Saving Bank". After a few seconds, the display will either ask you to insert

the next disk or will revert to the preset/sequence select screen. If the save required multiple disks, make sure to label them numerically.

■ LOADING FLOPPY DISKS

So far, so good...now it's time to move on.

1. Press the button labeled Load Bank, and position the cursor under the drive number in line two. Select the floppy drive using the data slider. Now insert the first floppy disk of the selected bank and press ENTER. The display will say: Loading Bank. Remember that loading in a new bank will erase the currently loaded Bank, so always think twice before pressing Enter to load the bank. Since EIII banks can be quite large, it may require several floppy disks to hold an entire bank. Multiple floppy disks must be inserted in the proper order (that's why it's a good idea to label them

2. Play the sounds from the newly-loaded disks. Feel free to check out the various presets, or load new disks. When you've had enough, move on to the next section.

■ TUNING EIII TO OTHER INSTRUMENTS

Refer to Master 1. This function demonstrates how EIII uses the data slider to adjust a parameter. Play the keyboard while adjusting the data slider to change the overall tuning.

■ TRANSPOSING THE KEYBOARD

Refer to Function, Transpose. While holding the transpose button, play a key on the keyboard in the lower two octaves. The second C from the bottom (C2) corresponds to normal or no transposition. All transpositions are based from this C2 key. For example, pressing the G key above C2 will transpose the keyboard up a perfect fifth. If the keyboard has been transposed, the transpose LED will stay lit. Press and hold the transpose button while pressing C2 to return to normal transposition.

GUIDED TOUR #2: SPECIFYING THE "CURRENT SAMPLE" AND THE "CURRENT ZONE"

■ BACKGROUND

EIII has two modules dedicated exclusively to processing samples within a bank: Digital Processing and Sample. Each sample stored in a bank can be processed independently by the Digital Processing module. Therefore, we need a way to specify the current sample, which is the individual sample to be processed.

The concept of the current sample is important. To process one sample out of a bank, you must first select the sample to be processed using function 0, (Select Sample) in the Digital Processing or Sample modules.

Zones are sections of the keyboard which can be selected to apply Analog Processing parameters or to be copied, erased or loaded from another preset. A copied zone contains samples as well as analog parameters. A zone can be one key or the entire keyboard range. Zones can exist within other zones. For example, if a zone is selected which includes previously defined zones, only the parameters that are altered in the newly selected zone will be altered in the previously defined zones. All other parameters will remain as previously specified.

Identifying which keyboard keys belong to which Sample

1. Activate the Preset Definition module; its LED will light. Select Preset Definition 2 (Edit Assignment) and play a key on the keyboard. The display will look something like this:

```
 EDIT  ASSIGNMENT
P00  pri  D2             <—Preset Number and last key pressed
p01  Sample  Name        <—Primary Sample assigned to the key
No  Secondary  Sample    <—Secondary Sample assigned to the key
```

As you run your fingers up and down the keyboard, the primary and or secondary sample numbers will change indicating the keyboard range of those samples. Moving the data slider will also show you the sample boundaries. Thus, Edit Assignment is good for checking out which samples are assigned to the keyboard. Changes do not necessarily have to be made.

2. Choose a sample and press ENTER. The display now shows the range of the current zone on the upper line of the display. Don't play any keys but press ENTER again. Now the display shows something like this:

```
EDIT  ASSIGNMENT

Zone:  XX  to  XX        <—Upper and lower boundaries of the Zone
Select  High  Key
```

Where XX is the name of the key (such as D2).

Specifying the Zone of the Analog Processing Module

Now that we know how many samples there are in the preset, and the range covered by each sample, let's specify a range of keys to be altered by the Analog Processing Module, starting with the lowest key.

1. Activate the Analog Processing module. Press 0 on the keypad to Select Zone.

2. The display instructs you to "Select Low Key". Press the lowest key on the keyboard, then press the ENTER button (whose LED is now flashing). The display now instructs you to "Select High Key". Press a key on the keyboard near the top of the keyboard, then press ENTER. The display now shows the current zone and asks you to "Select a Submodule".

3. You're now ready to start modifying the sounds in the bank with the Analog Processing Module the subject of the next tour.

4. The assigned current zone will remain as is until you either change the current zone assignment, change presets, or load another bank. If you switch between modules, the current zone remains as assigned (unless you select the Digital Processing Module, which will be a subject of a later tour).

5. Before proceeding, look over Analog Processing 0 to help reinforce what you've learned. Now that you know what a zone is and how to specify it, we've reached the end of this tour (although

you may need to come back from time to time to refresh your memory). In the next tour, we'll see how to modify samples with the Analog Processing section.

GUIDED TOUR #3: ANALOG PROCESSING OF A ZONE

■ BACKGROUND

The Analog Processing module consists of several sound processing functions. Let's start with the Filter and VCA sections, as they are among the most important.

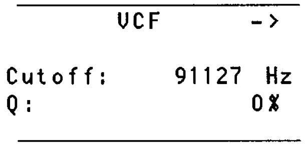
The VCA function contains sixteen VCAs that control a sound's amplitude envelope. The filter function contains sixteen VCFs that control a sound's timbre.

■ WORKING WITH THE FILTER

1. You could activate the Filter Setup function directly by keying in 3. However, let's investigate another way to select the filter function. Move the data slider to catalog the various Analog Processing functions. When the display shows function 3, VCF, press ENTER.

Changing Filter Cutoff Frequency

1. Refer to Analog Processing 3. The display should look something like this:



```
          VCF      ->

Cutoff:     91127  Hz
Q:                 0%
```

2. Play with the up and down cursor buttons (the arrow buttons directly to the right of the display). Note how you can move the cursor under the various parameters to be adjusted. For now, move the cursor under the Cutoff frequency.

3. Vary the data slider. Note how the numbers to the right of cutoff change. Lower numbers mean a lower filter cutoff frequency (less high frequencies). Higher numbers mean a higher filter cutoff frequency (more high frequencies). Observe how only the notes within the current zone are affected by the slider setting. Note: You must re-trigger a note to hear any changes. Holding down a note and playing the slider will not change the sound; you must play a note after changing the slider to hear the results of changing the slider. This is true when making any changes to the sound, not just while you're in the filter function or the Analog Processing Module.

By the way, now might be a good time to mention that although we are changing the sounds in the bank, the sounds on the disk remain unchanged. This is because we haven't saved the bank to disk. Thus, you can fool around with the bank sounds as much as you want without having to worry about altering the original sounds on the disk.

Changing Filter Q

1. Move the cursor under the Q% on the display. Vary the slider to change the sharpness of the sound; higher numbers give a sharper sound. Again, this affects only the range of notes covered by the current zone. Leave the Q at about 60 and proceed.

2. Move the cursor back to Cutoff and vary the data slider. Note how this produces a sort of wa-wa effect (remember, you have to re-trigger the key to hear the results of changing the Q).

3. Now set Cutoff to 200 Hz and Q to about 50. The range of notes covered by the current zone should sound muted.

Changing the Filter Cutoff Envelope

1. Let's investigate the effects of envelope control over the filtered sound. Select page two by pressing the right cursor arrow (>). Move the cursor under the Envelope Amount and use the slider to set a value of +40%. This allows the envelope to control the filter cutoff frequency.

```
<-     VCF       ->

Tracking:     +0.00
Envelope Amt: +  0%
```

2. Now we need to change over to the third page. Press the right cursor arrow (>) again and the display will change to:

```
<- VCF Attack: 0.00s
Hold:          0.00s
Decay:         0.17s
Sustain:0%  Rel:0.93s
```

3. Move the cursor under attack time and vary the slider. With larger values, it will take more attack time for the filter frequency to go from lowest to highest cutoff frequency. Vary the various envelope parameters, and observe the effect these changes have on the sound.

4. Now let's check out inverted envelopes. Set the envelope parameters as follows:

```
<- VCF Attack: 0.00s
Hold:          0.00s
Decay:         0.43s
Sustain:0% Rel:0.00s
```

...and play and hold a chord. This is a non-inverting envelope in the sense that the envelope increases the filter cutoff frequency above the initial cutoff. To select an inverting envelope, where the cutoff decreases below the initial cutoff, press the left cursor arrow (<).

5. We're back at the familiar Tracking/ Envelope Amt. screen. Move the cursor under Envelope Amount and select -40% to invert the envelope. Note that the envelope effect is not all that noticeable when you play a chord. This is because the envelope forces the cutoff frequency in a negative direction, and since the cutoff frequency is already fairly low, it can't go that much lower.

6. Now go back to the cutoff frequency and increase it. The effect will be far more noticeable since there will be more range available for the negative going envelope excursion.

7. If you feel like experimenting, play with the Tracking control to affect the way the filter frequency tracks the keyboard pitch.

8. Before proceeding with the tour, set Cutoff Frequency=91127, Q=00, Envelope Amount= +00, and Tracking=1.00. Set the envelope to Attack=0.00s, Hold=0.00s, Decay=0.00s, Sustain=100%, and Release=0.49s. After entering these values, press ENTER to return to the Module Identifier.

■ FUN WITH VCAS

1. In preparation for the following experiments, let's change the Current Zone to include the entire keyboard. Enter 0, move the data slider to the bottom of its travel and press ENTER. Now move the data slider to the top of its travel and press ENTER again. You have now selected the entire keyboard.

2. Now key in 2 to select the VCA function. The display shows:

```
       VCA     ->
Level:       100%
Pan:         + 0%
L..............|.................R
```

Use the right cursor/page arrow (>) to move to the next page of the VCA controls. The display will show something like this:

```
<- VCA Attack:   0.00s
Hold:            0.00s
Decay:           0.00s
Sustain:100%  Rel:1.65s
```

Move the cursor under the various envelope parameters and see how different settings affect the sound.

3. Before moving on, make sure you have a sound that is fairly sustained with little or no envelope attack time.

■ OTHER ANALOG PROCESSING OPTIONS

1. Key in 1, and note how the controls affect the sound.

2. Let's add some LFO effects. Key in 4 and the display shows:

```
        LFO        ->
Rate:        4.25Hz
Shape:       triangle
Delay:        0.00s
```

If the LFO rate is different, change the rate so that it is about 4.25Hz. Next, use the right cursor/page arrow (>) to move to the next page of the LFO controls. The display will show something like this:

```
<-       LFO        ->
Variation:          0%
LFO  -> Pitch       0%
LFO  -> Cutoff      0%
```

Position the cursor under each display option. Vary the data slider and observe how this affects the sound. Note that adding LFO to Cutoff might not sound all that noticeable, especially if the Cutoff Frequency is set above the audible frequency range. If you want a more obvious effect, bounce back to function 3 and set the Cutoff to about 200 Hz and Q to about 50. This should make the LFO's effect more noticeable.

Use the right cursor/page arrow (>) to move to the third page of the LFO controls. The display will show:

```
<-       LFO        ->

LFO  -> VCA         0%
LFO  -> Pan         0%
```

3. If the LFO settings aren't to your liking, then key the left cursor/page arrow (<) to change the LFO rate, delay, and variation.

■ THE AUXILIARY ENVELOPE

1. Key in 5 and the display shows the first page of the auxiliary envelope.

```
AUXILIARY  ENVELOPE ->

Dest:              off
Envelope  Amt:      0%
```

2. Move the cursor under destination and use the data slider to scroll through the various auxiliary envelope destinations. Select Pitch as the destination, then set the envelope amount to -50%.

3. Use the right cursor/page arrow (>) to move to the next page of the auxiliary envelope. Set the parameters so that the display looks like this:

```
<- AUX Attack: 0.00s
Hold:          0.00s
Decay:         0.40s
Sustain:0% Rel:0.00s
```

4. Now play the keyboard. Notes will bend up (since we are using an inverted envelope) to pitch and hold there. This is an effect common in many natural sounds.

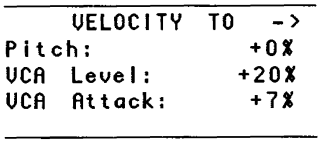
Vary the various envelope parameters, and observe the effect these changes have on the sound.

■ UNDERSTANDING VELOCITY

1. By now you might have a pretty messy sound due to all those exercises. Let's start with a clean slate. Press function button Load Bank and then ENTER to re-load the bank. Press the Analog Processing button.

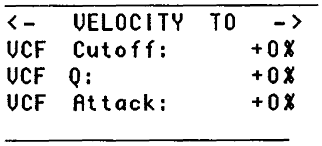
2. Let's make the entire keyboard the current zone. The zone defaults to the entire keyboard so we don't have to do anything!

3. Key in 6. The display will look something like this:



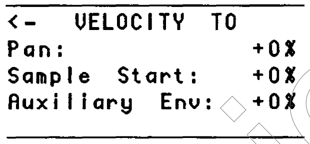
```
    VELOCITY TO   ->
Pitch:          +0%
VCA Level:     +20%
VCA Attack:     +7%
```

Moving on to the next velocity page we see:



```
<-  VELOCITY TO   ->
VCF Cutoff:     +0%
VCF Q:          +0%
VCF Attack:     +0%
```

Moving on to the last velocity page we see:



```
<-  VELOCITY TO
Pan:            +0%
Sample Start:   +0%
Auxiliary Env:  +0%
```

Move the slider to select different values, and note the effects. With + velocity sent to the VCA, EIII plays softer as you play softer. In other words, EIII equates harder play with the nominal volume setting and goes down from there as you play softer.

4. Move the cursor under the other available parameters and vary the slider. Notice that the filter cutoff frequency lowers as you play softer . The amount downward change is dependent on the value in the display.

Filter Q is affected differently than level or filter cutoff. It raises from the initial setting as you play harder Also, note that velocity can be set to affect Q inversely—in other words, if the filter is set to a high Q setting, playing harder on the keyboard will lower the Q.

Remember that the velocity-to-envelope attack setting interacts with the initial envelope attack settings; if you don't take our word for it, then by all means call up the envelope attack parameters for the filter and VCA and see how different values interact with different velocity values.

GUIDED TOUR #4: REAL TIME CONTROL PROGRAMMING

■ BACKGROUND

Ever wanted to add vibrato to a grand piano? Or bend its pitch? The Real Time Control Module can do this, and lots more.

■ PITCH-BENDING

First, let's check out pitch-bending. pitch bend can be enabled or disabled for any zone within a preset. Let's have pitch bend affect only the lower half of the keyboard. Key in Analog Processing 0 and select the upper half of the keyboard as the current zone (surely you know how to do this by now, so we'll spare you the details). Next key in Analog Processing 8 (Realtime Control Enable). Press the right and left (<>) cursor/page buttons, and you'll see a list of modulation destinations. Pitch will be set to On. Press Off, and like magic, you can now pitch bend only the lower half of the keyboard.

The important point of all this is that if modulation does not seem to affect a zone, make sure that modulation is enabled.

Want to change the pitch bend range? De-activate the Analog Processing module, activate Preset Definition, and refer to Preset Definition 6.

■ CHANGING MODULATION WHEEL DESTINATIONS

Note: EIII offers two modulation options: Pre-programmed (which adds a constant, selectable amount of modulation), and real time (where the player adds in modulation by using one of the wheels or other controller).

Each EIII wheel can be assigned to a particular destination. For example, if the left wheel is assigned to pitch, then rotating the wheel bends pitch. If assigned to the filter, rotating the wheel varies the cutoff frequency.

Let's set up for the next part of the tour. Choose preset 01 as the current preset if it is not already. Activate Analog Processing, assign the entire keyboard as the current zone if necessary, then key in 8. Press the cursor/page buttons and press On for all the enable options. This will make it easier to hear the results of the next series of experiments.

Now activate Preset Definition and key in 0. To make "live" playing as simple as possible, the display works somewhat differently for this module. The display shows:

```
 REALTIME  CONTROLS
1 Left  Wheel
1 Pitch
 Select  a  Controller
```

Use the data slider to scroll through the possible realtime control sources and their currently assigned destinations.

EIII Control Sources

1: Left wheel
2: Right wheel
3: Pressure
4: Control pedal (plugs into rear panel Pedal jack
5: MIDI control A (can be assigned to any MIDI controller #)
6: MIDI control B (can be assigned to any MIDI controller #)
7: Footswitch 1 (plugs into rear panel Footswitch 1 jack)
8: Footswitch 2 (plugs into rear panel Footswitch 2 jack)

EIII Modulation Destinations

0: Off
1: Pitch
2: VCF Cutoff
3: VCA Level
4: LFO -> Pitch
5: LFO -> Cutoff
6: LFO -> VCA
7: Pan
8: Attack
9: Crossfade

EIII Footswitch Destinations

0: Off
1: Sustain
2: Cross/Switch
3: Sequencer Play/Stop
4: Sequencer Continue/Stop
5: Sequencer Punch-in/out
6: Arpeggiator On/Off
7: Arpeggiator Latch
8: Preset Increment
9: Preset Decrement

Each of the destinations printed in the upper right-hand column (0-9) can be controlled by control source 1, 2, 3, or 4, or via data sent over MIDI. Each of the destinations printed in the lower right-hand column (0-9) can be controlled by sources 7 and 8.

You'll be happy to know that real time control settings are memorized for each individual preset. Thus, if desired each preset can react to the real time controls and MIDI controllers in different

■ SELECTING A CONTROL SOURCE and CONTROL DESTINATION

1. With Preset Definition 0 activated, the left wheel will be selected as a control source. Next move the cursor down to the next line and the display asks you to "Select a Destination".

2. The left wheel should be assigned to 1 (pitch). Vary the wheel and check that the keyboard pitch is indeed affected. If sections of the keyboard are not affected, check that pitch control is enabled (Analog Processing 8). Now press 2; this assigns the left wheel to filter cutoff. Rotating the wheel towards you should produce a more muted sound. Key in 3, and the left wheel will affect overall volume. If you feel adventurous, check out the other control destinations. Note that if you select a destination that is already specified for one of the other control sources, the old assignment will be de-selected and that control source will be turned off (0).

3. Think about it for a bit...the left wheel can control a destination, the right wheel can control a different one (as can the pedal), and there are MIDI control possibilities too. These assignments can be different for each preset, and particular controller destinations can be disabled for different presets and samples. We're talking versatile here, so if you feel like taking out the next couple of hours and checking out all the possibilities, be our guest! Note: When assigning the footpedal, make sure it's plugged in to prevent unpredictable results.

4. Oh yes, and there are footswitches too. But before experimenting with the footswitches (sources 7 and 8), we need to understand the difference between looped and unlooped sounds. But basically, a "sustain looped" sound is one where a portion of the sound is put into an "infinite repeat" loop for as long as you hold down the key. This is similar to the infinite repeat function on digital delay lines. Looping allows for sustaining a normally non-sustaining sound for as long as you like. An unlooped sound is not artificially sustained, and therefore lasts its normal length.

Keeping this in mind, refer to Preset Definition 0, footswitch destinations 0-9. Assign various functions to the footswitches; note that some sounds lend themselves to the sustain function better than other sounds, so feel free to experiment.

GUIDED TOUR #5: BASIC SAMPLING

■ BACKGROUND

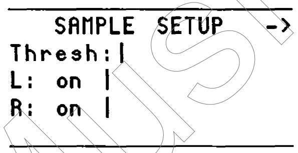
Sampling does not just involve sticking a microphone in front of something—sampling is an art. This guided tour gives you the basics, and also lays the groundwork for the guided tour of the Digital Processing Module.

■ SETUP

1. Plug a microphone into either the left or right (or both) rear panel Sample Input jacks (an instrument can also be used, but a mic is easier to work with for now).

2. Press Master 3 and then Yes. This clears out the memory, which gives us maximum sampling time.

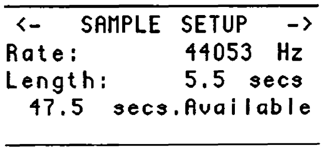
3. Activate the Sample Module and select 5 (Setup). The display shows:



```
   SAMPLE SETUP   ->
Thresh:|
L: on  |
R: on  |
```

4. If you are sampling in mono, move the cursor to the input that you are not using and turn it off with the On/Off buttons. Otherwise, leave both channels on.

5. Next, set the level using the Sample Input slider below the display. The level should be set so that the peak of the VU meter does not quite reach the top of its range with a full level signal. Use the right cursor/page button to view the next page of the Setup section which looks like this:



```
<-  SAMPLE SETUP   ->
Rate:       44053 Hz
Length:     5.5 secs
 47.5 secs.Available
```

Also set the sampling rate. This should be at 44.053 kHz; might as well leave it there for now. Check the available sampling time, which should be 47.5 seconds if you are sampling in mono or 23.6 seconds if you are sampling in stereo.

6. Go on to the next page of the Setup by pressing the right cursor/page button again. The display shows:

```
<-SAMPLE  SETUP(Auto)
Truncate:          off
Normalize:         off
Placement:     12 keys
```

7. Set the automatic parameters as shown above. The Auto-Placement parameter determines where the samples we are going to take will be automatically placed. In this case with Auto-Placement set at 12 keys, the first sample will be placed on the lowest octave (C1 to B1), the next sample will be automatically placed on the next octave up (C2 to B2), and so on.

8. Go back to the first page and set the threshold using the data slider. This should be set up a few bars from the bottom so that the EIII does not trigger on the ambient room noise. Press ENTER to return to the module identifier, press 7 to arm the sampling process, and you're ready to sample! Speak into the mic; as soon as the level exceeds the threshold, the display will say "sampling." Feel free to talk away for 47.5 seconds, but if you lose patience, press ENTER to stop sampling.

9. Now experiment with more sampling: Try setting a particular sample length, using forced sampling instead of threshold-sensitive sampling, assigning the sampled sound to other portions of the keyboard, and also, practice terminating the sampling process.

10. To save the sample as part of a preset, de-activate the sample module. Since memory was cleared prior to sampling, EIII created a preset (00) called "Untitled Preset," and this is the preset that holds your new sample. Had you sampled into a bank with existing presets, the sample would have been stored in a specially created preset.

11. Let's try another sample. Notice that EIII, a very polite instrument, doesn't overwrite the existing sample without your express permission; thus, the new sample is automatically assigned to the next higher octave. This assignment can always be modified with Sample 6. Also note that if you're ever in a position where you're going to overwrite an existing sample, EIII will notify you via

12. Remember, this guided tour is intended simply as an overview to give you a "feel" for the sampling process. Good sampling requires skill and patience; practice!!

GUIDED TOUR #6: DIGITALLY PROCESSING SAMPLES

■ BACKGROUND and SETUP

Digital processing allows for radically altering samples stored in the bank. In this guided tour, we'll learn — among other topics — how to truncate, loop, reverse, splice, and combine samples. First, though, we need to take a couple of samples with which we can practice.

1. Clear the Bank of memory (Master 3).

2. Follow the directions in Guided Tour #5 and make a sample of yourself speaking. However, select a 4 second sample length (Sample 5). After setting the threshold and such, arm sampling and start jabbering.

3. De-activate the Sample module, then re-activate and take another 4 second sample. You will not have to do any setup — just hit Sample 6 and talk away. You now have two samples suitable for experimentation.

4. Assign your samples to a keyboard zone in preparation for the next steps.

■ CHOOSING THE CURRENT SAMPLE

1. Activate Digital Processing. The module acts upon the current sample. To select another sample press Digital Processing 0 (Select Sample).

■ TRUNCATING A SAMPLE

1. Choose Digital Processing 4 to truncate the ends off samples.

2. Vary the data slider as you play a key in the range assigned to the current sample. Notice the start of the sample will disappear. Use this technique to get rid of silence at the start of a sample

3. Move the cursor using the up/down arrow keys to the End display. Use the data slider to truncate any undesired part off the end of the sample.

4. When you get tired of truncating, press ENTER. The EIII makes a backup of the sample (on the hard disk) in case you decide that you don't like the truncation after all.

5. If desired, change the current sample (Digital Processing 0) and experiment with truncating the other sample you took.

■ LOOPING A SAMPLE

1. If you haven't yet done so, activate the Digital Processing Module and assign the current sample. Select function 1 and turn the loop mode On.

2. Refer to Digital Processing 2 to graphically see how looping affects a sample. If you play and hold a key in the current sample, it will play indefinitely since the loop function is on.

3. Now try modifying the loop. Activate Digital Processing 2 and adjust the Start and Length values. Note that if the length equals the full length of the sample, you will not be able to set a new start point. Reduce the length, and you should be able to adjust the start point. For practice, try looping individual words or sentence fragments.

■ A PRACTICE SAMPLING SESSION

1. Plug a microphone into the rear panel right SAMPLE INPUT jack.

2. Erase all memory (Use Master 3).

3. Set the level (Sample 5). Sing ahhhh into the microphone (This is an easy sound to loop). The VU level should not quite reach the top of it's range. Adjust the gain (use the Sample Input slider) until a good level is attained.

4. Set the threshold (Sample 5). This should be set about 5 or 6 bars from the left. When the input sound exceeds this level, recording will start.

5. OK, now get ready to sing ahhhhh and press (Sample 7) to arm the sampling process.

6. Now sing ahhhhh! When you are out of breath, press (ENTER) to stop sampling.

7. Play the keyboard in the assigned range (the sound should default to the lower keyboard range, C1-B1) to hear the results.

8. Now we can truncate the silence off of the beginning and end of the sound. Press (Digital Processing), the EIII will default to the only sample in the unit, which is the one you just made. Now press 3 to select truncation.

9. Adjust the start and end points by moving the slider to remove any silence or unwanted portion of the sound (You must play the keyboard after you move the data slider in order to hear the results of the slider change). Press ENTER then No to exit truncation.

10. Now we are ready to loop the sound. Press 2 to select looping. To loop, first move the data slider to make the loop length about half a second long.

11. Next move the start point into the sustained portion of the ahhh sound so that the ticking sound becomes softer. The idea is to loop the "hhh" portion of the "ahhh" but not the "a " part. The loop length can be adjusted so that it matches the rhythm or wavering of the sound. When you get a fairly good loop press ENTER. The display says: Auto Correlate? Y/N. Press Yes. You now should have a fairly good loop. If not, adjust the start point and again Auto Correlate. It takes practice,

That's it! Feel free to play around with other processors such as the filter, chorus and LFO.

12. In summary; these are the steps you will go through every time you have a sampling session (of course you will develop your own order and style of doing it).

A) Take the sample
B) Truncate the sample
C) Loop the sample (if desired)
D) Assign the sample to a keyboard position
E) Continue to refine and make presets

Please remember that looping is a very complex process which requires a great deal of practice for proficiency. The only way to fully understand looping is to experiment with a variety of signals. The purpose of this guided tour is to acquaint you with the concept of looping, but there is much more to the subject than simply fooling around with looping words from a spoken sample.

■ TIME TO SAVE?

Maybe you haven't created any masterpieces during these experiments...but maybe you have. If you want to save a preset to disk, by all means do so. Also since hard disks are not immune to crashes, take some blank formatted disks prepared earlier and Save Bank.

■ MORE DIGITAL MAGIC

Let's play with some more digital processors. In preparation, erase all memory (Master 3), then take a sample of your voice speaking your own name.

1. Activate the Digital Processing Module. Select function 1, Playback Setup. The display shows:

```
   PLAYBACK  SETUP
Loop  Type:   forward
Loop  in  Release:  on
Reverse  Playback:off
```

2. Move the cursor down to Reverse Playback and select on. Now play the sample back until you feel that you have learned to say your name backwards. Think you've got it, do you? OK, now sample yourself saying the reversed version of your name, then of course..... reverse it!

■ CUT AND PASTE

1. Erase all memory, then take yet another sample of your voice. Set the sample length to about 1.2 seconds. This time say, "Cut and Paste". Pronounce the words clearly and distinctly.

2. We're going to rearrange this phrase to say: Paste and Cut. Activate the Digital Processing Module, select function 5 (Cut Section) and press ENTER. The display shows:

```
CUT    secs   samples
Start:0.00    00000
End:   1.00   44096
Size:  1.00   44096
```

The original pitch of the sample will appear on key C4(the 4th C up from the bottom).

3. Place the cursor under End and adjust the data slider so that only the word "cut" is heard.
Hint: Use the left and right cursor keys to adjust the point once you get close. They will only select zero crossing points.

Press ENTER and do not Auto Correlate. The display will say backing up sample and return you to the module identifier.

4. Now select Digital Processing 6 (Paste), and then ENTER. The display shows:

```
PASTE    secs  samples
Offset:  0.00  00000

  Select  Location
```

5. Adjust the offset past that "tt" of the word paste. Again use the left and right (<>) cursor keys once you get close. Next press ENTER and don't Auto Correlate. The sample should now say,"and paste, cut".

6. Go back to the Cut submodule and cut out the word "and" using the same procedure you used for "cut".

7. Adjust the offset past the "tt" of the word paste. Use the left and right (<>) cursor keys once you get close. Next press ENTER and don't Auto Correlate. The sample should now say,"paste and cut", albeit probably sounding a bit strange. Anyway, that's the basic technique. Practice makes perfect.

8. Now that you have gotten a feel for cut and paste, make up your own experiments using the various other options such as mixing and crossfading.

■ DIGITAL EFFECTS

Let's try using a couple of the Digital Effects; Gain Change and Taper. In preparation, sample your voice saying "ahhh" for 1 second. This time when you sample, set the Sample Input slider so that the signal only reaches about halfway up the VU meter at it's peak. Also make sure that the sample time "runs out" before you finish singing so that the sample is "cut off" prematurely. We're intentionally creating problems so that we can "fix" them using Taper and Gain Change.

1. First, let's boost the gain to full level using the Gain Change function. This is called normalization. Select Digital Effects 3, Gain Change (8/3). The display shows:

```
GAIN    secs  samples
Start:  0.00  00000
End:    1.00  44096
Size:   1.00  44096
```

2. We want to normalize the entire sample (which is already selected), so simply press ENTER. The display now shows:

```
   GAIN  CHANGE

Amount:     +00dB
+XXdB  =  Normalize
```

where XX is the amount of gain needed to achieve normalization or full level.

3. Set the amount of gain so that it matches the normalization reading and press ENTER. The display shows:

```
FADE  secs   samples
Size: 0.00   00000
Type:        Linear
```

4. We don't want a fade in this example, so simply press ENTER to affect the gain change. The sample should now be at a higher volume. Note that Digital Processing 9 (Undo) will "undo" the effects of Gain Change. You can use this feature to further experiment with different Gain Change settings.

5. Now let's taper the end of the sample so that it smoothly fades out instead of ending in a thump. Select Digital Effects 2, Taper (8/2). The display shows:

```
TAPER   secs  samples
Start:  0.00  00000
End:    1.00  44096
Size:   1.00  44096
```

6. Move the start point to about 0.70 seconds and press ENTER. the display changes to:

```
        TAPER
Start  Amount:0.00dB
End  Amount:    -96dB
Type:          Linear
```

7. The Start and End amounts are already at the correct settings. There will be no attenuation at the Start point tapering to full attenuation at the End point of the sample. Move the cursor under the curve type, select Exponential Curve 2, then press ENTER.

8. The sample should now smoothly fade out instead of ending abruptly. Note that Digital Processing 9 (Undo) will "undo" the effects of Taper. You can use this feature to further experiment with different Taper settings.

GUIDED TOUR #7: MANAGING THE BANK

■ BACKGROUND AND SETUP

The following functions don't necessarily do glamorous things, but they are very useful. This guided tour acquaints you with these utilities.
Load any bank from the hard disk.

■ ERASING A PRESET

1. Refer to Preset Management 3 and erase preset 01. Don't worry, this is just gone from the bank, not the disk. And we can get it back anyway by...

■ LOADING A PRESET

See Preset Management 1 and load preset 01. Now it's back in the bank again.
This is the basis method for creating a bank with presets from other banks. The bank can only be saved as a whole. Therefore, individual presets are loaded into the bank and when the bank is arranged to your specifications, it is saved to disk.

■ OTHER BANK MANAGEMENT FUNCTIONS

1. The other functions—copy, rename, create, and preset size— are pretty much self-explanatory. Refer to Preset Management 4, 2, 5, and 6 respectively. Try these various functions to get a feel for how they work.

GUIDED TOUR #8: MAKE YOUR OWN

These guided tours cover only the basics; to cover every possibility of how to use the instrument would drown you in words. It's better that you just start playing! The best way to learn EIII is to dive right in. Remember, you can experiment as much as you want on stuff in the bank; you have to actually save the bank to disk in order to alter the contents of the disk, and in case you're nervous, you can even lock the disk.

Try to spend some time not playing with, but rather practicing with, the instrument. After you're a little more familiar with EIII, read through the Reference Manual and delve deeply into a particular function or module. EIII is like an audio construction set, where sounds can be captured, held, processed, mutated, spliced, sped up, slowed down...and lots more, so take advantage of what

The more you practice with EIII, the more you'll be able to put your personal stamp on the music you play. And when you do play, you'll know the instrument well enough so that you are free to concentrate on the music.

INTERFACE SPECIFICATION

MIDI INTERFACE

■ **Hardware:** MIDI is a serial, 5 milliamp current loop interface between two or more computer controlled musical instruments. It uses optoisolators and non-grounded inputs to prevent ground loops. MIDI operates at a 31.25kHz Baud rate and does not use any handshaking. Each MIDI connection requires two wires to complete the loop. The hardware used is very simple. The output is usually an open collector driver, typically a 7407 and a series resistor. On the EIII, the driver is not an open collector driver but a leftover line driver (74HCT240) and a 1N914 diode (see schematic). This is not a typical MIDI driver circuit, but it works fine. The return line is a series resistor connected to 5 volts. The MIDI input current runs through a 150 Ω resistor and the directly into a PC900 optoisolator. EIII uses the standard 5 pin DIN connectors for MIDI IN and MIDI OUT and THRU.

■ **MIDI Thru:** MIDI Thru on EIII is a hardwire function which simply routes incoming MIDI data to the MIDI THRU jack.

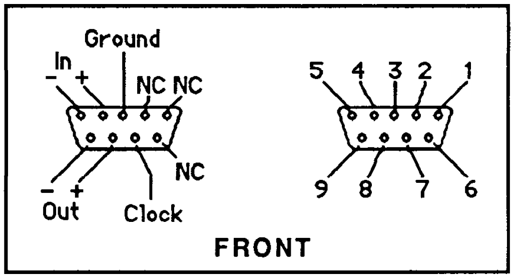
■ **Continuous Controllers:** Note that EIII's continuous controller channels are <u>programmable</u> on any of the 32 channels for each preset. This allows for much flexibility and sometimes confusion for many users. When MIDI problems are encountered, make sure that the problem is not due to incorrect preset programming (or outside MIDI connections) before assuming hardware or software faults. Note also that the preset's MIDI parameters can be overridden by the MIDI Globals located in the Master module.

■ **System Exclusives:** EIII will contain a few MIDI system exclusive commands. These will include the MMA sample dump commands and hard disk bank load commands. These commands can be transmitted and received via MIDI or by the faster RS-422 port. Most of the functions normally associated with MIDI system exclusives will be implemented via the faster SCSI port. The MIDI commands and SCSI links should be implemented by Oct.. 1988. For more information on the system exclusive specifications or SCSI links, contact our Customer Service department.

RS-422 CONNECTION

The RS-422 connector is a high-speed serial communication port which allows data to be transferred to and from an external computer at a very high rate (500K baud). The 422 port uses a differential mode of operation in order to reduce noise and the chance of transmission error. The connection cable to the external computer will generally be supplied with a software package (such as Sound Designer by Digidesign) and does not come with EIII. The EIII contains built in diagnostics which will test the RS-422 interface. Refer to the Diagnostics section for information on these tests.

■ **RS422 CONNECTOR PINOUTS**



SCSI CONNECTION

The SCSI (pronounced skuzzy) connection on the rear panel of the EIII is the standard high speed interface of the computer industry. SCSI stands for Small Computer System Interface and allows external devices such as CD-ROM drives, WORM (Write Once, Read Many) drives, or external Hard Disk drives to communicate and transfer data to and from the EIII at very high speed. In addition, external computers such as the Macintosh II can be interfaced via SCSI in order to implement additional DSP (Digital Signal Processing) functions. SCSI can support up to 8 devices (including the EIII and its hard drive). Each SCSI device has a unique ID number so that it can distinguish its data from data meant for another device. When adding SCSI devices to the system, a device may be inadvertently added which has the same SCSI ID number as an existing device in the system. In this case a SCSI error would occur. To remedy this situation, the ID number of one of the SCSI devices would have to be changed. Many SCSI devices have *hardware* defined ID numbers, which means that there is a little switch somewhere on the device to change the ID number. Consult the operation manual of the external device for information on changing the SCSI ID number.

If a SCSI device has been powered up after the Emulator Three, it will not appear in the list of available devices. The Mount Drive utility (MASTER, Disk Utilities, 1) tells the EIII to check then SCSI bus and add any new SCSI devices that it finds to its list of available SCSI devices. Normally, if the external SCSI devices are powered up before the EIII, this will be done automatically.

The SCSI ID number of the Macintosh is fixed at ID #7 and cannot be changed. The SCSI ID numbers of most other devices *can* be changed.

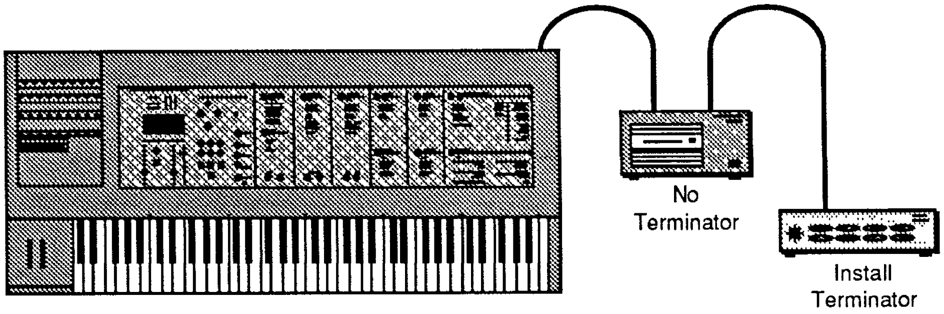
Default SCSI ID numbers

Macintosh	ID# 7 (fixed)
__________	ID# 6
__________	ID# 5
__________	ID# 4
__________	ID# 3
HD300	ID# 2
EIII Internal HD	ID# 1
Emulator III	ID# 0

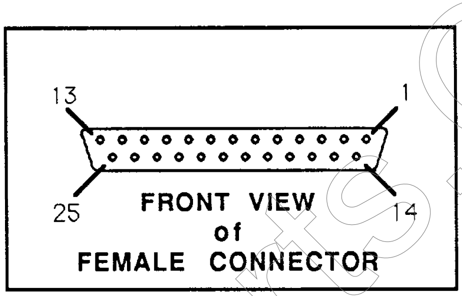
Terminating Resistors

Terminating resistors or terminators are special resistor packs that are installed on the first and last device in the SCSI chain. Terminators are used to reduce line echos, or standing waves on the SCSI bus. You can think of termination resistors as noise reducers.

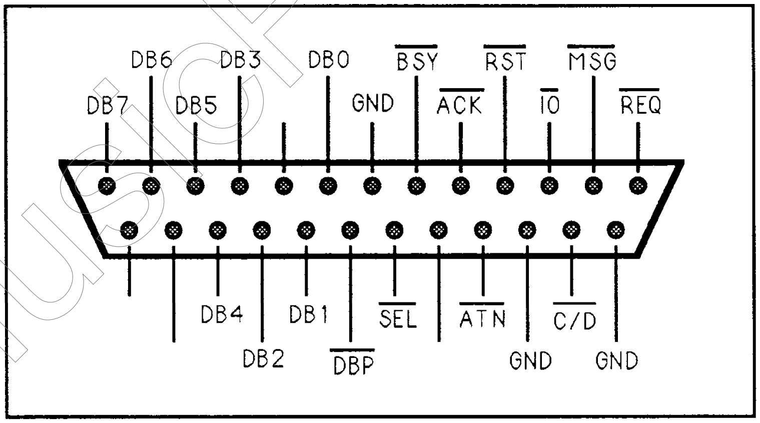
Most manufacturers ship their external SCSI devices with terminators installed. In theory, the termination resistors on devices in the middle of the SCSI chain should be removed. In practice, however, we have found that the SCSI bus has fewer errors when all terminators are left in place. The old adage,"Don't try to fix something that ain't broke." certainly seems to apply here. If you have SCSI problems with all terminators in place, try removing the centermost resistors to eliminate the problem. The longer your SCSI cables are, the more likely it is that problems will occur. The EIII does not seem to like SCSI cables of greater than 12 feet.



This is the theoretical way to attach SCSI devices to the Emulator III. In actual practice, the terminators should not be removed unless necessary.

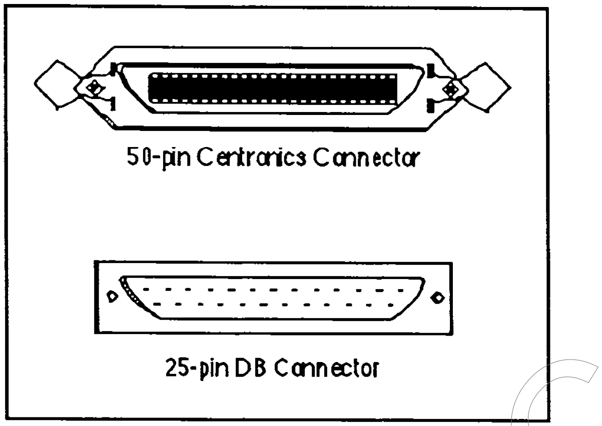


The SCSI Connector



Types of SCSI Cables

There are two basic types of SCSI cables in use: DB25 and 50-pin Centronics. The two types of cable can be identified by their connectors, which are very different. In buying SCSI extension cables, it is important to get ones with the right type of connectors, since the two types will not interconnect without a special adapter. The type we use are called DB25 connectors.These are the type found most often on the back of personal computers. When ordering, ask for a DB25 to DB25 type cable with male pins on both ends. Make sure to get a cable that has all the pins connected. Many cables only have some of the pins wired.



50-pin Centronics Connector

25-pin DB Connector

SMPTE IN and SMPTE OUT

■ The SMPTE IN jack allows the EIII to receive SMPTE time code at any of the four standard frame rates of 24, 25, 30, and 30df frames-per-second. SMPTE sync can be used to synchronize the internal sequencer clock or locate sequencer events to actual SMPTE times using the Cue List mode. Note that to keep synchronization with a SMPTE track, the playback tempo of a sequence must match the tempo at which it was recorded since sequence tempo is independent of SMPTE time.
The SMPTE IN jack also allows an external device such as a drum machine or sequencer to control the tempo of the sequencer or arpeggiator. EIII can receive input clock rates of 24, 48, or 96 pulses-per-quarter note. The pulses should be **at least 1 millisecond wide and have a level of 1 to 5 volts.**

■ The SMPTE OUT jack generates any of the four standard frame rates to be recorded on tape. SMPTE is generated out of the EIII at a **level of 5 Volts.**

METRONOME/CLOCK OUT

The metronome out jack in addition to providing a metronome click, allows the EIII to be the master clock and drive sequencers and drum machines at a rate of 96 pulses per quarter note. This is a **5 volt pulse which is about 6 milliseconds wide.**

FOOTSWITCH AND PEDAL CONNECTION

■ **Footswitch:** Two control footswitch jacks are provided for performance control of sequencer, arpeggiator, sustain, release, Xswitch, and preset advance. The footswitches should be of the **momentary contact type** but can be **either normally-open or normally-closed.** The type of footswitch can be selected, using the Footswitch Polarity function, under SPECIAL, in the **MASTER** module.

■ **Pedal:** The control pedal is used as a realtime control in the same manner as the wheels. It can be programmably routed to a destination by using the realtime control section of the preset definition module. The pedal can either be a **voltage type**, (which outputs a DC voltage from 0-9 volts at the tip of the jack) **or a resistance type** (which varies a 100K ohm resistance from the tip to ground of the jack). Any 0-10 volt control voltage can be input to the pedal input jack.

AUDIO CONNECTIONS

■ **Stereo:** EIII has provisions for a variety of output connection schemes. The most common hookup will probably be for stereo operation. Stereo operation allows the use of EIII's panning features and adds another dimension to the sound. Connect the left and right audio out phone jacks to the inputs of your mixer or stereo preamp. **EIII will also drive stereo headphones,** which can be plugged directly into the left/stereo output.

■ **Mono:** If a monophonic amplifier is used, simply connect the right/mono output to the input of the amplifier. Guitar amps are not recommended. The output level of EIII is somewhere in between instrument and line level. Care should be taken when connecting to an instrument amplifier so that the delicate nerve cells in your ears are not damaged or destroyed.

■ **Individual Outputs:** There may be times when different equalization or reverb settings are desired on the various instruments that have been sampled. EIII has 16 monophonic channel outputs that can be used when individual processing on specific instruments is desired. Zones are assigned to output channels using the output channel assignment function in the Analog Processing module. Dynamic Allocation defeats the channel assignments. Note that the stereo outputs are summed from the individual outputs. Output impedance is approximately 1 K Ω.

■ **Sample Input:** The sample in jack can accept any signal level from microphone level to line level. The gain is adjusted with the sample input slider while in VU/gain mode in the sample module. The level can be read in the liquid crystal display while in this mode. Input impedance is 10K Ω.

AC POWER CONNECTION and FUSE

The AC power connector is how EIII gets power (obviously). The fuse receptacle is also fairly obvious. It is located directly next to the power receptacle. Before changing or checking a fuse, UNPLUG the power cord. To remove the fuse holder, squeeze the two tabs located on either side of the fuse holder together. The fuse holder will now pop out with its two fuses in the fuse sockets. **EIII uses two, 2 amp, 250 volt fast-blo mini-fuses.** EIII should not normally blow fuses. If a fuse that has been replaced blows again, suspect the switching power supply.

EIII MIDI Implementation Chart

X = item implemented
- = item not implemented

MIDI command	Transmitted?	Received?	Comments
Note off	X	X	keys#21 thru 108 (A1 thru C7)
Note on	X	X	keys#21 thru 108 (A1 thru C7)
Poly key pressure	-	-	
Control change	X	X	controllers # 0 thru 31
Program change	X	X	program # (preset#) 0 thru 99
Channel pressure	X	X	
Pitch wheel	X	X	
Sustain footswitch	X	X	assignable 64 thru 79
Local control on/off	-	X	
All notes off	-	X	
Omni mode off/on	-	-	ignores mode, turn all notes off
Mono mode Select	-	-	ignores mode, turn all notes off
Poly mode Select	-	-	ignores mode, turn all notes off
Song position pointer	X	X	
Song select	X	X	
Tune request	-	-	
Timing clock	X	X	
Start sequence	X	X	
Continue sequence	X	X	
Stop sequence	X	X	
Active sensing	-	-	
System reset	-	-	
MIDI Time Code	X	-	
System Exclusives	X	X	Available Oct. '88. Call for info.

Supermode

"Supermode" is an EIII MIDI mode designed to enhance the Sequencer/MIDI interface. It is basically equivalent to "Poly Mode On" for all 16 MIDI channels simultaneously.
When Supermode is on, the OMNI/POLY setting is ignored, but the MIDI Notes/Wheels flag of the current preset is still checked by EIII to determine whether to accept or ignore incoming MIDI performance data.

The EIII sequencer has 16 tracks and each track has an independent preset assigned to it. These presets can be referred to as"Sequencer Presets".

Realtime Controllers

EIII has 6 realtime sources: left wheel, right wheel, pressure, pedal, MIDI A, and MIDI B. These sources may be the on board controls, or coming in over MIDI, or both. Each realtime control source can be assigned to any MIDI continuous controller channel from 0 to 31.

EIII has 9 realtime destinations: pitch, filter, level, LFO to pitch, LFO to filter cutoff, LFO to VCA level, attack, pan, and crossfade.

SYNCHRONIZATION

The Emulator III contains provisions for most types of synchronization in use today and thus, in most cases eliminates the need for an additional sync box . The two types of sync that the EIII does not acknowledge are FSK Sync and Roland Din sync. Fortunately, most units that use FSK sync (Oberheim) also have provision for a square wave clock input which the EIII can easily provide. To use Roland Din sync you must use a converter box such as the Doctor Click from Garfield Electronics.

Common Forms of Synchronization

The various forms of synchronization can be placed into two basic groups: Click type syncs and Clock type syncs. Most modern equipment will probably have one of the clock type syncs incorporated into it, however, you may have occasion to interface with an older piece of gear. The following is a listing of the most common forms of synchronization.

■ = Emulator III compatible
▲ = Interface Instructions

Click Type Syncs which advance the sequencer 1 step.

■ **Click Sync**-- Square wave or pulse. A carry-over from the days of modular analog synthesizers and sequencers. The sequencer starts when it receives pulses and stops when the pulses stop. Usually recorded at a rate of 24, 48 or 96 pulses-per-quarter note. Can be recorded on tape.
▲ Connect the click sync to the SMPTE IN jack of the Emulator Three and set the clock divisor (located under "Clock" in the Sequencer Management module) to the appropriate rate.

FSK-- Frequency Shift Key. An audio signal made up of two frequencies, a carrier and another that modulates against the carrier. Records to tape well. Rates of 24, 48 and 96 pulses-per-quarter note.

DIN Sync-- Roland Sync 24. Called DIN because of the DIN connector used. Uses a continuously running 24 pulse-per-quarter note square wave clock and another line to signal start/stop. Cannot be recorded on tape.

Clock Type Syncs which convey absolute location information.

■ **SMPTE**-- Stands for Society of Motion Picture and Television Engineers who standardized the time code in 1969. SMPTE is an audio signal based upon hours, minutes, seconds (and frames). Each frame of SMPTE time code has its own unique identity number and can be located precisely by a SMPTE reader. SMPTE can be written at frame rates of 24, 25, and 30 frames-per-second. The Emulator III reads and writes (LTC) Longitudinal Time Code which is designed to be recorded on audio tape.
▲ Use the SMPTE IN and SMPTE OUT jacks to write and read SMPTE. Select SMPTE clock (located under Clock in the Sequencer Management module). Make sure that the correct SMPTE rate has been selected when reading SMPTE code.

■ **MIDI Clock**-- MIDI, Musical Instrument Digital Interface, is a serial, computer language which can transmit synchronizing "MIDI Clocks" at a 24 pulse-per-quarter note rate. (MIDI clock can actually be placed in the click sync category as well since the individual MIDI clock pulses do not convey time information, however, if the MIDI Song Position Pointer is added it becomes a powerful clock type sync.) A "MIDI Song Position Pointer" is used to locate a precise position within a song. MIDI information is not designed to be recorded on tape.

▲ Set the clock type to MIDI clock (located under Clock in the Sequencer Management module) and make sure that the clock enables are set correctly (located under MIDI in the Preset Definition module). If you are using Song Position Pointer, make sure that this is set correctly for your application. Song Position Pointer is located under MIDI Options in the Sequencer Management module. Oh yes, the MIDI cables must also be properly connected.

■ **MTC**-- MIDI Time Code is the MIDI equivalent of SMPTE time code and uses the same standard time format (hours, minutes, seconds, frames). MTC is a serial computer language and is not designed to be recorded on tape but can be converted to SMPTE if tape sync is required.
▲ MIDI Time Code must be selected. This option is located under MIDI Options in the Sequencer Management module. The Emulator Three only transmits MTC. It does not receive it.

More about MIDI Sync--

The MIDI interface is actually a specialized type of computer interface and the MIDI clock is probably the simplest type of MIDI word. A MIDI clock is simply the digital word 11111000 or F8 in hexidecimal, which is sent out over the MIDI cable 24 times per quarter note. By design or by coincidence, the MIDI clock looks exactly like a pulse wave clock when viewed on an oscilloscope. There are also other Real Time Messages that pertain to synchronization such as Start, Stop, Continue, Song Select, Song Position Pointer, and MIDI Time Code. The digital codes for these messages are listed below.

Status Byte	(Hex)	Data Byte	(Hex)	Description
11111000	F8	none	-	Timing Clock
11111010	FA	none	-	Start (from beginning)
11111011	FB	none	-	Continue
11111100	FC	none	-	Stop
11110011	F4	0sssssss	-	Song Select (sssssss= Song number)
11110010	F2			Song Position Pointer
		0lllllll	-	lllllll=Least Significant
		0hhhhhhh	-	hhhhhhh=Most Significant

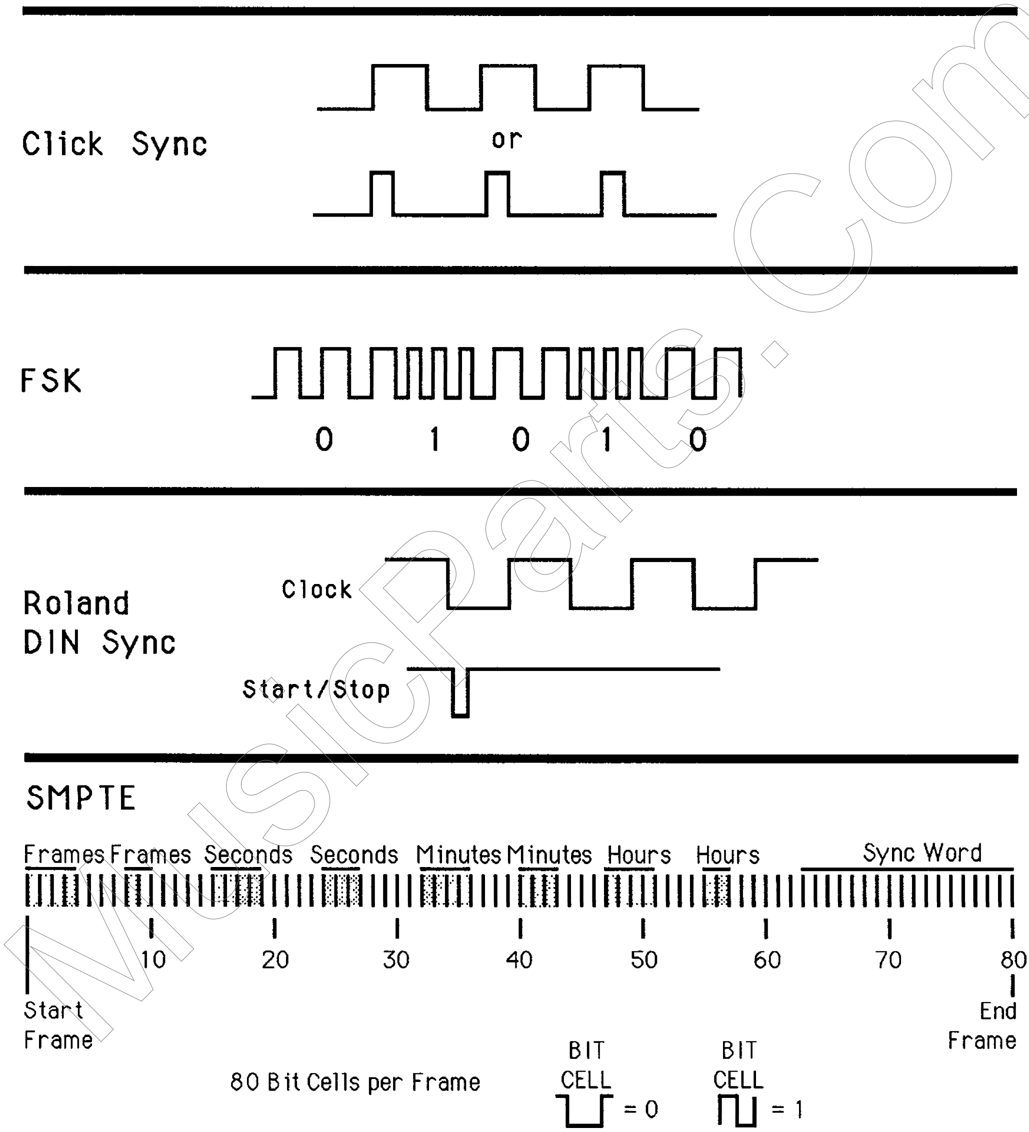
You will notice that the messages are divided up into Status Bytes and Data Bytes. What's the difference between a status byte and a data byte? Status Bytes always have a "1" as their first bit and identify the message type. Sometimes there is more information to be communicated and so Data Bytes, which follow immediately after the status byte, contain that extra information. Data bytes always have a "0" as their first bit. Note that the first four messages do not have data bytes. These are simple commands which can deliver all of their message in a single byte. The Song Select message contains a status byte and a single following data byte. The status byte tells the computer that the next message will contain the song select information, which can be a number from 0 to 127.

Song Position Pointer

Song Position Pointer contains two data bytes following the status byte. The song position pointer is the number of MIDI beats (1 beat = 6 MIDI clocks) since the beginning of the song. Because there are 14 bits of information for the song position, a song can be 16,384 MIDI beats long without overflowing the song pointer.
As an example of how song position pointer works in the EIII, let's say we are adding parts to the end of a song, and an external sequencer which is controlling other synthesizers is synchronized via MIDI clock to the EIII. Using the Locate buttons on the EIII we can fast-forward through the song to

just before the point where we want to come in. Here's what happens in terms of MIDI sync: When we release the locate button, the EIII immediately sends a song position pointer status byte followed by the two data bytes to the external sequencer. The EIII now begins sending MIDI clocks (at 24 ppqn). The external sequencer locates to the proper position in the song using the song position pointer and then continues to keep time using the MIDI clocks. The two units now can run together in perfect sync.

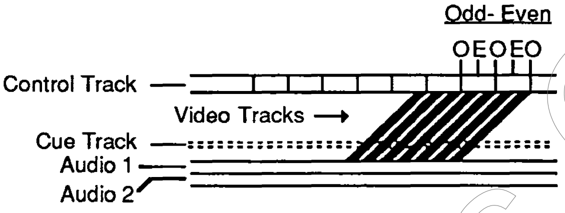
COMMON SYNCHRONIZATION WAVEFORMS



SMPTE TIME CODE

SMPTE time code was standardized in 1969 by the Society of Motion Picture and Television Engineers as a way to mark frame numbers on video tape. Before SMPTE standardization, there were various systems in use, none of which would work together. The SMPTE standard was also adopted by the European Broadcasting Union which is why you will often see it referred to as SMPTE/EBU time code.

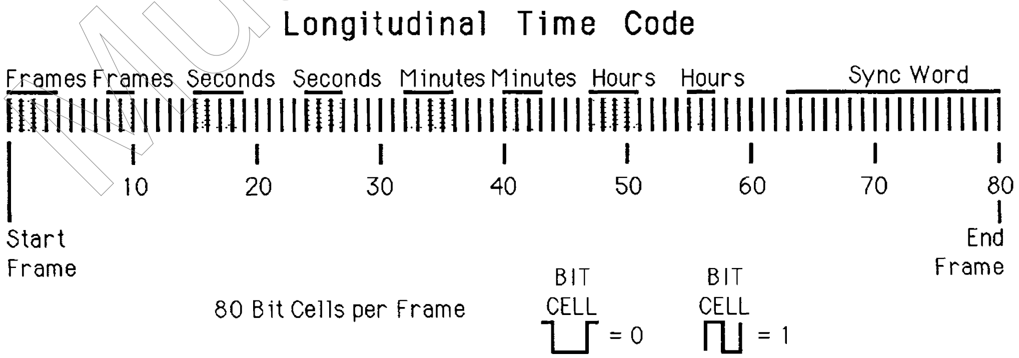
Using SMPTE, a particular location can be precisely located by simply entering the appropriate time code number which is expressed in Hours, Minutes, Seconds, Frames and often Subframes. This is possible because each frame of SMPTE time code contains absolute location information expressed in digital form. There are two types of SMPTE time code **VITC** (Vertical Interval Time Code) which is used on video tape and **Longitudinal** or audio, time code. VITC is strictly used for video and has the advantage of being able to be read while the video deck is paused. Longitudinal time code can be recorded on the audio or sync tracks of video tape and can thus be used in audio or video work.



VITC is recorded on the video tracks themselves during the vertical blanking interval.

Longitudinal time code is recorded on the cue track or one of the audio tracks.

Longitudinal time code is the type used on the Emulator III and contains 80 bits of information per frame. Each frame can be further divided into subframes which run from 00-79. The SMPTE start point on the EIII sequencer can be specified to the subframe level. An audio SMPTE frame is divided into 80 "bit cells". A voltage change during a bit cell period constitutes a digital "1" and no change during a bit cell period constitutes a digital "0". The transition system is used because it is easily recorded on audio tape. In addition to the location bits, there are user bits that may contain information about tape reel numbers, bits dealing with video information, and a 16-bit sync word at the end of the frame.



There are four types of SMPTE time code in use: 24, 25, and 30 frame-per-second and 30 drop-frame.

Type	Use	Hours	Minutes	Seconds	Frames
24 frame	US Film	00-23	00-59	00-59	00-23
25 frame	Euro. Film + Video	00-23	00-59	00-59	00-24
30 frame	US B/W Video	00-23	00-59	00-59	00-29
30 drop-frame	US Color Video	00-23	00-59	00-59	00-29

The four frame rates are all straightforward except 30 drop-frame. The 30 df rate came about because the US color video frame rate is actually 29.97 frames/sec instead of 30 frames/sec. The .03 frames/sec difference adds up to an error of 108 frames each hour! So drop frame was invented to correct this time difference. To achieve this, every minute except 00-10-20-30-40-50 have the first two frames, 00 and 01, "dropped", hence the name drop-frame.

Why use SMPTE?

Older types of syncs, namely click tracks and FSK, rely on counting electronic pulses from a known starting point (the start of the song). A main problem with these older types of sync is that the count can be upset by data dropout or tape slippage, and these errors are cumulative. Click type syncs have the added disadvantage of always having to be started from the beginning of the song, which can be frustrating when doing edits at the end of long songs. To make matters worse, there are at least five different types of click syncs in use which creates additional interfacing headaches. SMPTE sync conveys "absolute" location information. Since each frame of SMPTE code provides its own unique identification, it provides the ability for a receiving device to recover from data dropout or tape slippage. In addition, edits can be performed in the middle of a song with just a few seconds of pre-roll before the punch-in point. SMPTE is also standardized, which means that code generated on different makes of equipment will compatible with each other. SMPTE also has fairly good resolution, especially at the subframe level.

	at 24 fps = .521 mS
Each Subframe	at 25 fps = .500 mS
	at 30 fps = .417 mS

"Striping" SMPTE

Printing SMPTE on tape is called striping (as in stripe). SMPTE is usually placed at an edge track on a multitrack deck to avoid crosstalk problems. SMPTE is usually recorded at about -3 VU on semi-pro gear, -10 VU on professional gear and 0 VU on video gear. Hotter levels have a tendency to "bleed" over to adjacent tracks while lower levels may themselves be corrupted by crosstalk from an adjacent track. Experiment to find the optimum levels. When printing to a time code track of a video deck, be careful. The time code playback head locations on video decks are not standardized and can cause gross timing errors. Time code which is striped on an audio track will always be in sync with the picture. SMPTE code is normally recorded on the right channel of a video recorder.

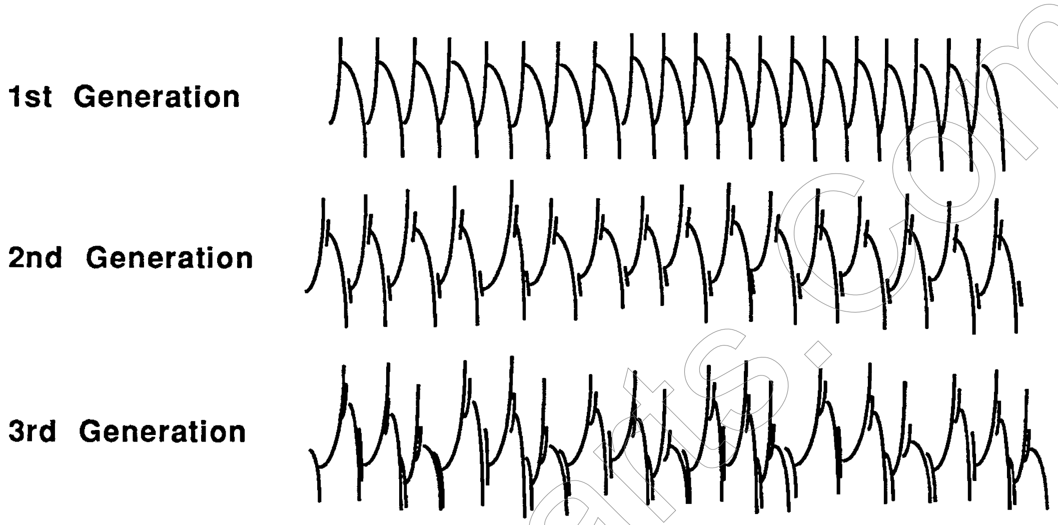
Avoiding SMPTE problems

Problems in reading SMPTE time code can often be related to poor quality code on the tape. Poor quality code can be caused by a number of problems, the most common being dirty or misaligned heads, amplifier clipping, or too many generations of audio dubbing. Other problems can be caused by running the SMPTE signal through signal processing devices such as Limiters, Reverbs, Harmonizers, etc. You may laugh, but it *has* been done. In fact, many video decks have built in AGCs (Automatic Gain Controls) which will ruin the SMPTE signal if the input level is too high. Always check playback to insure that the time code is usable. In general, no signal processing should be used on the SMPTE signal. SMPTE code (and all time code) is delicate and should be treated as such.

Duplicating SMPTE time code

The best way to duplicate a tape with a SMPTE track on it is to re-generate the time code through a master/slave clock or slave generator that produces fresh SMPTE code in sync with the source tape. Straight dubbing from tape to tape will produce deterioration of the SMPTE signal with each generation. One generation of dubbing will probably be OK. The diagram shows the effect of duplicating SMPTE code.

SMPTE Distortion Caused By Tape Duplication



Other Tips

■ Use ascending time code. Jumps in the code are OK as long as the SMPTE code jumps forward in time as the tape moves forward in time. A good way to avoid any problems with this is to simply stripe the entire tape with SMPTE before you record any other tracks.

■ Allow enough leader. A 20 second leader before the first recorded material is recommended so that the tape will not come off the take-up reel of the recorder on a fast rewind. Also, leave a few seconds between each song to allow the EIII to sync up before the song starts.

■ Keep written logs. Keeping written records of song start points and edit cues can save time and avoid wasteful searching through a tape that was recorded earlier.

MECHANICAL PROCEDURES

PRECAUTIONS

Observe the following precautions when working on EIII:

Switch power off and check 110/220 switch before connecting EIII to power outlet or amplifier.

NEVER toggle the 110/220 switch with power on.

Do not bend or strain the PCBs or you may cause tiny breaks in the printed circuit traces which will be very difficult to find.

Switch power off before disconnecting or connecting any circuitry, or removing or installing PCBs.

To replace soldered components, switch power off, remove the PCB completely from the instrument, and desolder from both sides. Use a vacuum desoldering tool. The traces on the EIII circuit boards are very thin. Use extreme care and work carefully. Remember: Heat the pin, not the pad.

The Microcontroller and CPU boards on the EIII are multi-layer boards. Use extreme care when desoldering from a multi-layer board. If you are not an expert desolderer, it may be best to actually "clip out" the suspect part, then desolder the leads, rather than risk damage to the inner board traces. If inner traces of a multi-layer board are damaged, the entire board may need to be replaced.

KEYBOARD EIII

Before taking EIII apart we recommend providing a soft work surface. A carpeted or rubber covered workbench is ideal.

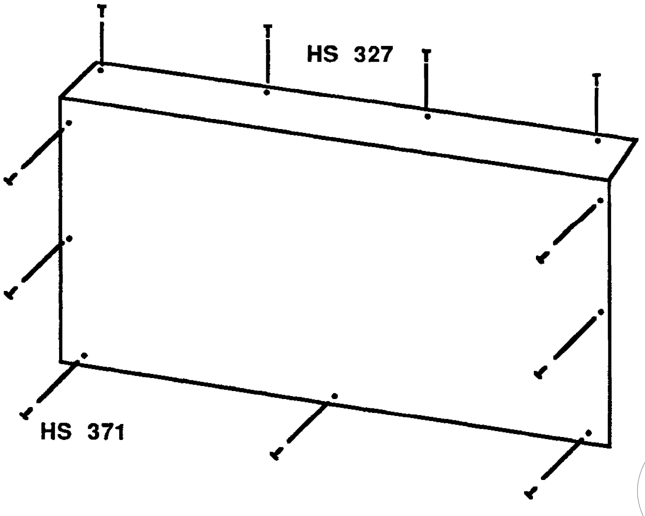
NOTES ON EIII SCREWS

Particular care must be exercised when removing and replacing the chassis screws on the EIII. These screws will strip out the brass Pemserts (the little brass threaded pieces which are embedded in the plastic housing) if they are tightened down too hard or crossthreaded. Make sure that the Pemserted screws go in smoothly. If a screw is crossthreaded, simply replace it with a new screw. If the Pemsert is pulled out from the plastic housing, don't worry. It can be "welded" back in place by heating it up with the tip of your trusty soldering iron or glued back with epoxy.

OPENING THE EIII

Turn the Emulator Three upside down on the padded work surface so that the back panel is facing you. This will allow easy access to all of the chassis screws. Remove the (8) bottom panel/chassis screws (E-mu P/N HS371) from around the perimeter of the chassis, then remove the (4) screws from the edge of the back panel (E-mu P/N HS327). Set these screws aside in a safe place.

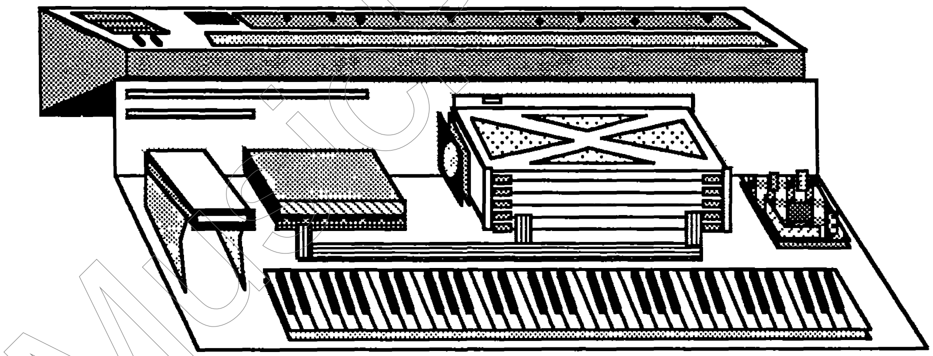
While holding the plastic housing to the chassis, carefully flip the instrument over back into a normal playing position with the keyboard facing you. The plastic housing can now be separated from the chassis by carefully lifting the plastic housing up in the back, then sliding it toward you slightly to clear the keyboard, and finally up and around to be placed behind the chassis so that you are looking through the keyboard cutout. Be careful not to pull or strain any of the fragile ribbon cables. When replacing the chassis screws be very careful not to cross-thread the screws into the brass Pemserts.



EIII Chassis Screw Locations

THE SERVICE POSITION

Place the EIII rightside-up on the padded workbench so that the keyboard is toward you. This will allow you to play the keyboard and have the jacks toward the rear of your bench. After the chassis screws are removed the plastic housing can be lifted up and slightly forward so that it clears the edge of the keyboard. It can now be set down resting on the back of the chassis. This position allows access to the front panel and LCD.



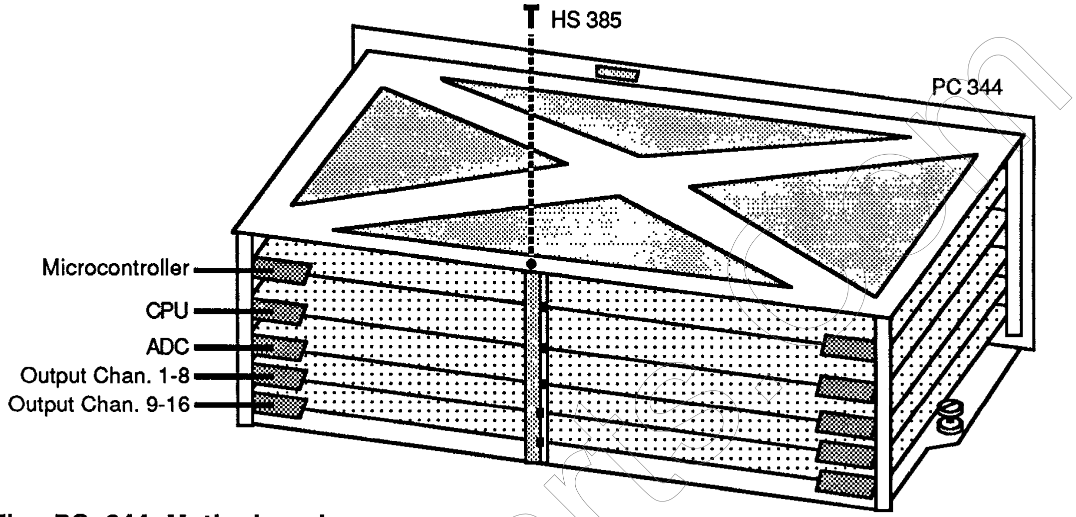
THE SERVICE POSITION

ACCESSING THE CARD CAGE

To access the card cage and circuit boards contained therein, first loosen the large screw on either side of the card cage which fastens the cage assembly to the chassis. The card cage can now be tilted up in order to access the circuit boards. To allow easier access to the circuit boards, a small object such as block of wood can be placed underneath the front of the card cage to tilt it upwards. This is almost a necessity when working on the lower boards in the cage. Be careful, however, that the back of the motherboard does not short out to the rear of the chassis.

Before any boards can be removed the white nylon circuit board support must be removed. To do this, first remove the phillips screw (HS 385) fastening the support to the card cage and put it in a safe place. Next, press down slightly next to the other end of the card support. This action should allow the little pin on the lower end of the support to pop out of its hole. Set the support aside.

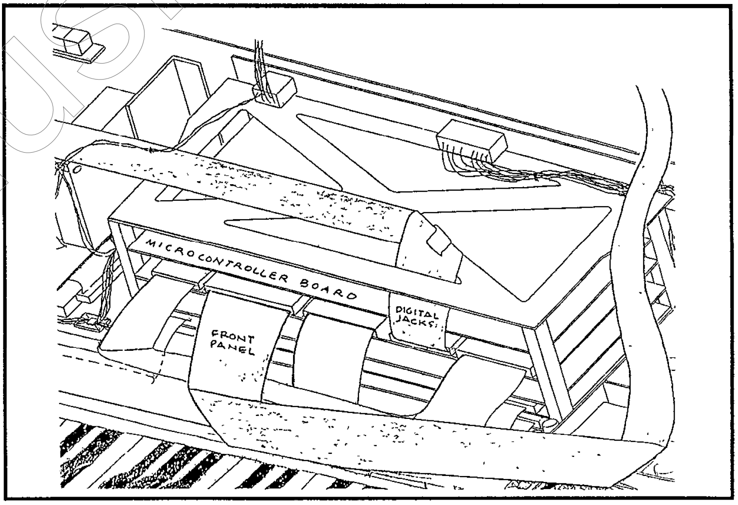
Use the white nylon board ejectors to eject the boards and carefully slide out the board to be worked on. Use the EIII extender/debug board to gain access to any of the circuit boards.



The PC 344 Motherboard

NOTE: On the sealed-box type of card cage, the CPU and Microcontroller boards are reversed.

Since the ribbon cables that emerge from the front of the card cage tend to get in the way while working on the unit, you may have to rearrange them. Make a note of how they are routed so they can be replaced properly later. The digital jack cable should be routed (if it isn't already) in the configuration shown in the following diagram. This prevents the front panel board from punching holes in the cable.



REMOVING THE POWER SUPPLY

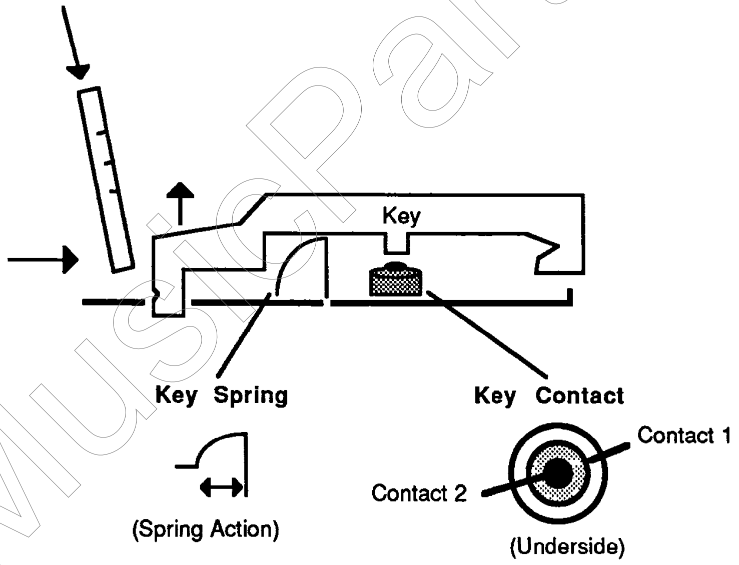
The switching power supply is mounted directly to the steel chassis by 4 screws (E-mu P/N HS353). Before removing the screws, make sure that the AC power is disconnected from the unit. Next, remove the DC connector to the supply, then the 2-pin AC connector and the 110/220 switch connection. The 110/220 switch is connected to the supply by an in-line type connector and carries 110 V. When reinstalling the supply, make sure that there is no exposed metal on this connector. If there is, heatshrink it!

REMOVING THE KEYBOARD

Removal of EIII's keyboard will not be necessary for most repair jobs. To remove the keyboard: First, disconnect the keyboard ribbon cable from the CPU board to the keyboard circuit board. Next, disconnect the pressure connection on the right side of the keyboard. The keyboard is fastened to the chassis by 6 screws (HS384) and 6 washers (HW336). Set these parts aside in a safe place. The keyboard assembly can now be carefully lifted out from the housing and set aside.

To remove a key from the keyboard, press the rear of the key with a plastic object such as the cardcage card support (EP340)as shown in the diagram. The rear of the key will pop up and can now be removed by sliding it forward. Be careful not to lose the key spring. If a black key is to be removed, the adjacent white keys will have to be removed first. If a key is mistriggering, carefully lift up the rubber key contact and gently clean the contacts with a high quality contact cleaner.

Use a piece of plastic to pry out the keyboard keys.
The cardcage card support (EP340) works well.



REMOVING THE FLOPPY DISK DRIVE

To remove the floppy disk drive, the drive bracket should be removed first (it's easier). The drive bracket is mounted to the chassis by (4) kepf nuts (HN 308). The disk drive itself is secured to the bracket by (4) 4-40 x 1/4" machine screws (HS 353). These should be removed after first removing the power and data cables from the floppy drive. Take particular care when handling the fragile DC power connector.

REMOVING THE HARD DISK DRIVE

FIRST...BACK IT UP! Before removing or doing anything to the hard disk drive, make sure that *all* the data contained on the drive is completely backed-up. Hopefully the customer has already backed it up. It would be a good policy to insist that the customer back-up the HD, if possible, before you begin any work. This sort of policy will absolve you from blame if some unforeseen accident were to occur and the HD's data dumped. Keep in mind that the minor torsional stresses imposed on the HD by simply being bolted into the unit may cause read or format errors on the HD.

Early EIII's have the hard disk mounted directly to the chassis bottom via two L-brackets. Unfortunately, this mounting arrangement subjected the HD to torsional stress. A newer design utilizing a mounting plate (EM392) which mounts to the back panel, as well as the bottom of the chassis, eliminates problems caused by torsion. Units with the HD mounted to the bottom panel by means of an L-bracket should be updated after the contents of the HD have been safely backed-up (the HD may have to be re-formatted after this update). When ordering this mounting plate, be sure to order the two screws (HS119) which attach the plate to the back panel. An even newer HD mounting design utilizes a standard hard disk mounting plate (EM395) which is also used on the EIII rack mount. This plate requires only a single screw (HS336). Refer to the EIII rack drawing AG131 for an illustration.

RACK MOUNT EIII

THE SERVICE POSITION

Before taking EIII apart we recommend providing a soft work surface. A carpeted or rubber covered workbench is ideal. Place the EIII rack rightside-up on the padded workbench so that the front panel is toward you. You should also have a MIDI keyboard within reach so that EIII can be played. The rack mount EIII is easier to work on than the keyboard model. The cards can be easily accessed when mounted on the extender card.

REMOVING THE TOP PANEL

Remove the (6) top panel screws (E-mu P/N HS 368) from the perimeter of the top panel, then lift off the metal panel and place next to the main unit. With the EIII in this position, most of the circuitry can be easily accessed.

REMOVING THE FRONT PANEL BOARD

The front panel may need to be removed in order to replace the slider pots or to clean the front panel buttons. The first step in removing the front panel board is to disconnect the ribbon cable to the front panel board. Next, is to disconnect the AC power switch. The AC power switch is held in place by two small machine screws (HS352). Next remove the (4) 11/32 size (HN 308) nuts from the rear of the front panel with a nut driver, then one more 11/32 nut (HN 308) which is located in the notch in the lower center of the front panel board. The control panel board can now be completely separated from the rack chassis. The front panel board is secured to the rack chassis by means of 11 phillips head machine screws (E-mu P/N HS353).

REMOVING THE FLOPPY DISK DRIVE

To remove the floppy disk drive, first remove the power and data cables. Take particular care when handling the fragile DC power connector. The floppy drive bracket (EM 394) is mounted to the chassis by a single screw (HS 353) and two tabs. The disk drive itself is secured to the bracket by (4) 4-40 x 1/4" machine screws (HS 353).

REMOVING THE HARD DISK DRIVE

FIRST...BACK IT UP! Before removing or doing anything to the hard disk drive, make sure that *all* the data contained on the drive is completely backed-up. Hopefully the customer has already backed it up. It would be a good policy to *insist* that the customer back-up the HD themselves, if possible, before you begin any work.

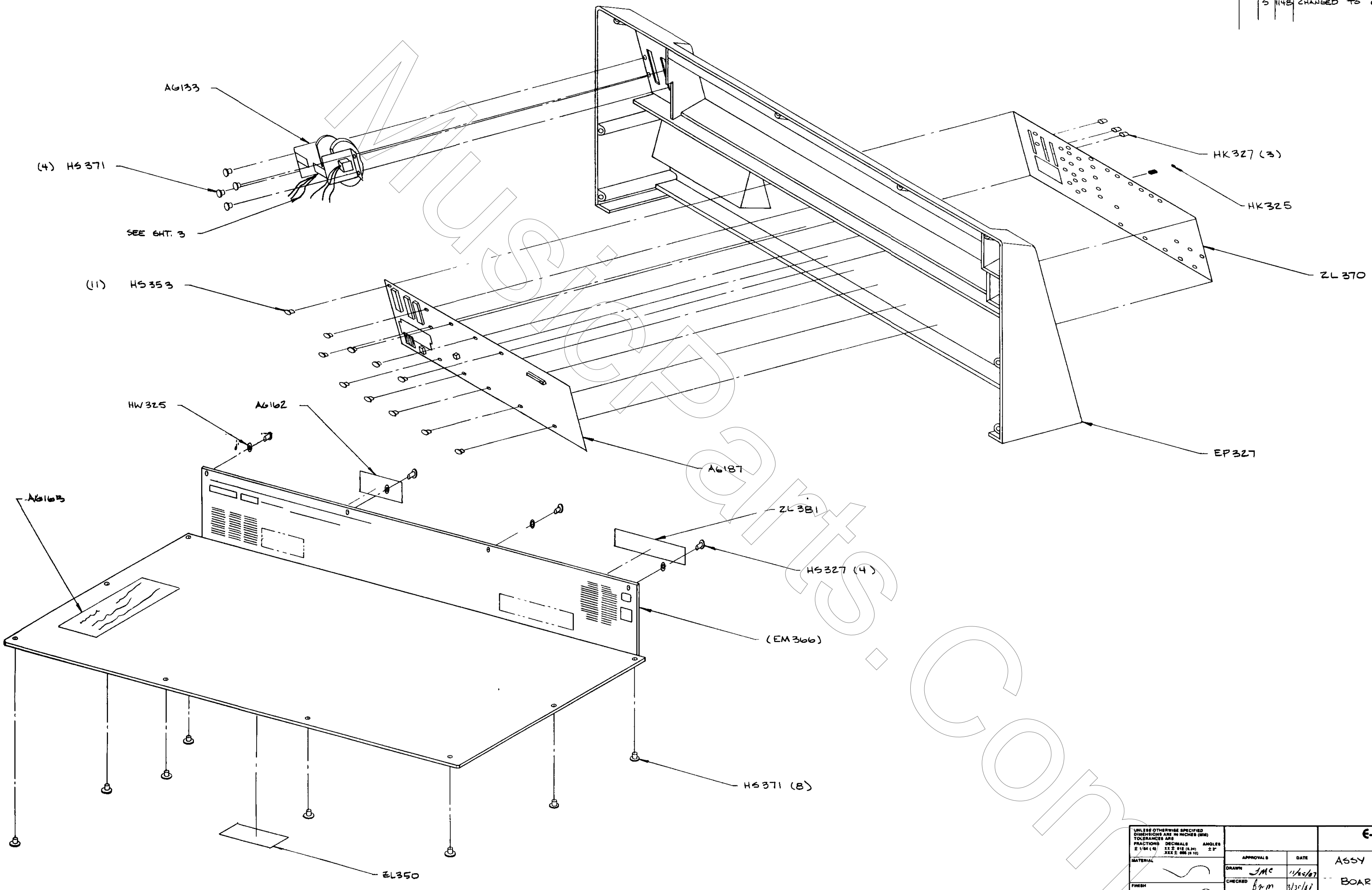
The HD mounting utilizes a standard hard disk mounting plate (EM395). This plate requires only a single screw (HS336). The hared disk drive attaches to the mounting plate with (4) 4-40 x 1/4" machine screws (HS 353). Refer to drawing AG131 for an illustration.

REMOVING THE POWER SUPPLY

The switching power supply is mounted directly to the steel chassis by 4 screws (E-mu P/N HS353). Before removing the screws, make sure that the AC power is disconnected from the unit. Next, remove the DC connector to the supply, then the 2-pin AC connector and the 110/220 switch connection. The 110/220 switch is connected to the supply by an in-line type connector and carries 110 V. When reinstalling the supply, make sure that there is no exposed metal on this connector. If there is, heatshrink it!

NOTES UNLESS OTHERWISE SPECIFIED

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR'D/DATE
	N	1055	INITIAL RELEASE	3/29/88
	O	1069	EM392 WAS EM396, REMOVED HS368	4/7/88
	P	1085	AG187 WAS AG129, ADDED AG185	5/22/88
	Q	1127	BOM CHANGE ONLY	6/10/88
	R	1149	RELEASE 1.21 OPER SOFTWARE	BZ 6/28/88
	S	1148	CHANGED TO CONNER H.D.	7/15/88

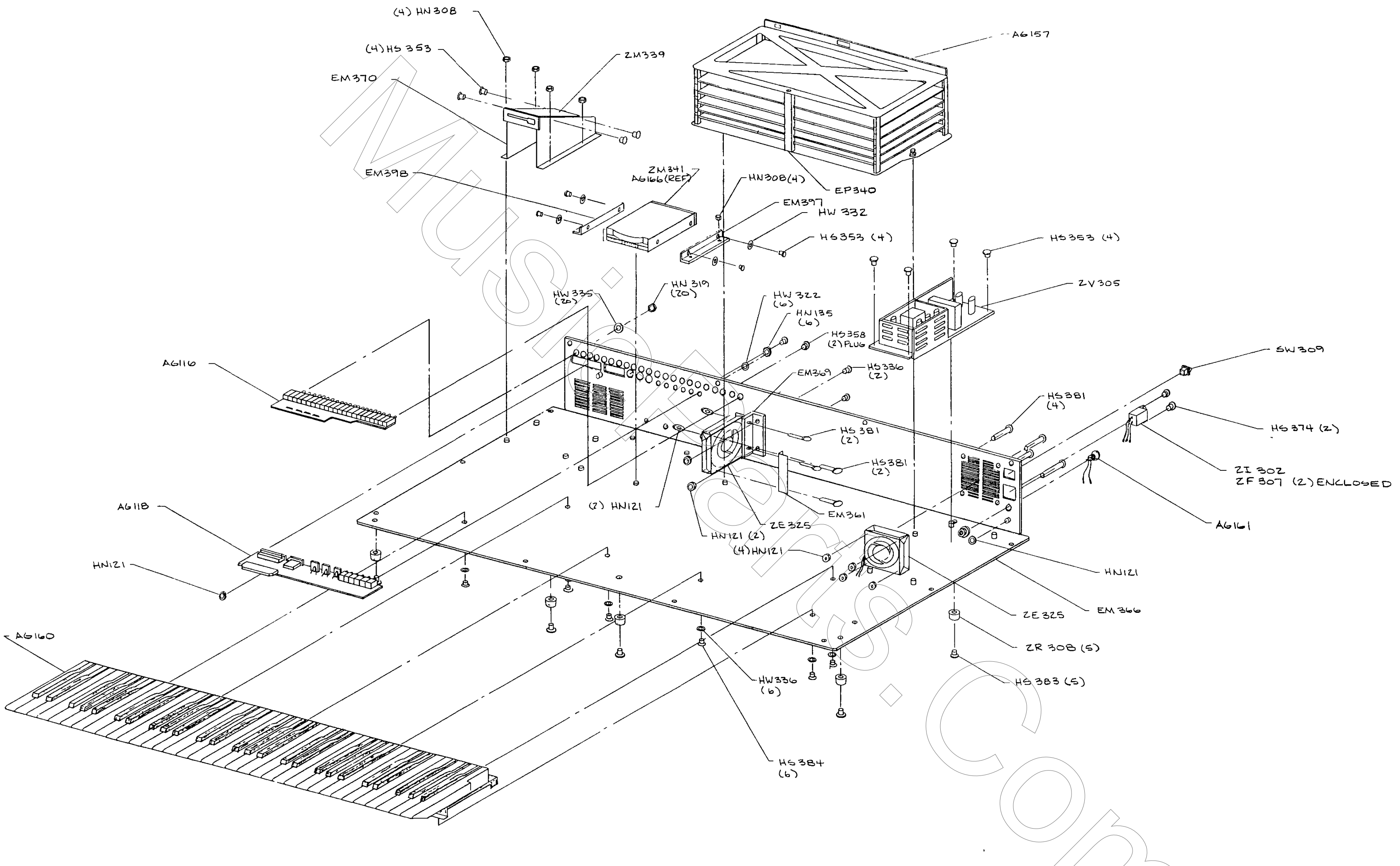


UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES (MM) TOLERANCES ARE FRACTIONS ± 1/64 (.4) DECIMALS .XX ± .012 (0.30) .XXX ± .005 (0.12) ANGLES ± 2°			**E-mu Systems, Inc.** Scotts Valley, CA
MATERIAL	APPROVALS	DATE	ASSY FINAL KEY- BOARD, 4 MEG E-3
	DRAWN JMC	11/25/87	
FINISH	CHECKED Brm	3/30/88	
	APPROVED	3-30-88	SIZE D · DWG NO. AG132(A) · REV. S
DO NOT SCALE DRAWING			SCALE 1/4 · SHEET 1 OF 3

8 7 6 5 4 3 2 1

NOTES UNLESS OTHERWISE SPECIFIED

REVISIONS				
REV	REV	ECO	DESCRIPTION	APPR'D/DATE



UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES (MM)
TOLERANCES ARE
FRACTIONS ± 1/64 (4)
DECIMALS .XX ± .015 (0.34)
.XXX ± .005 (0.12)
ANGLES ± 2°

MATERIAL

FINISH

DO NOT SCALE DRAWING

APPROVALS	DATE
DRAWN	
CHECKED	
APPROVED	

E-mu Systems, Inc.
Scotts Valley, CA

SIZE D

DWG NO AG132 (B)

REV. 5

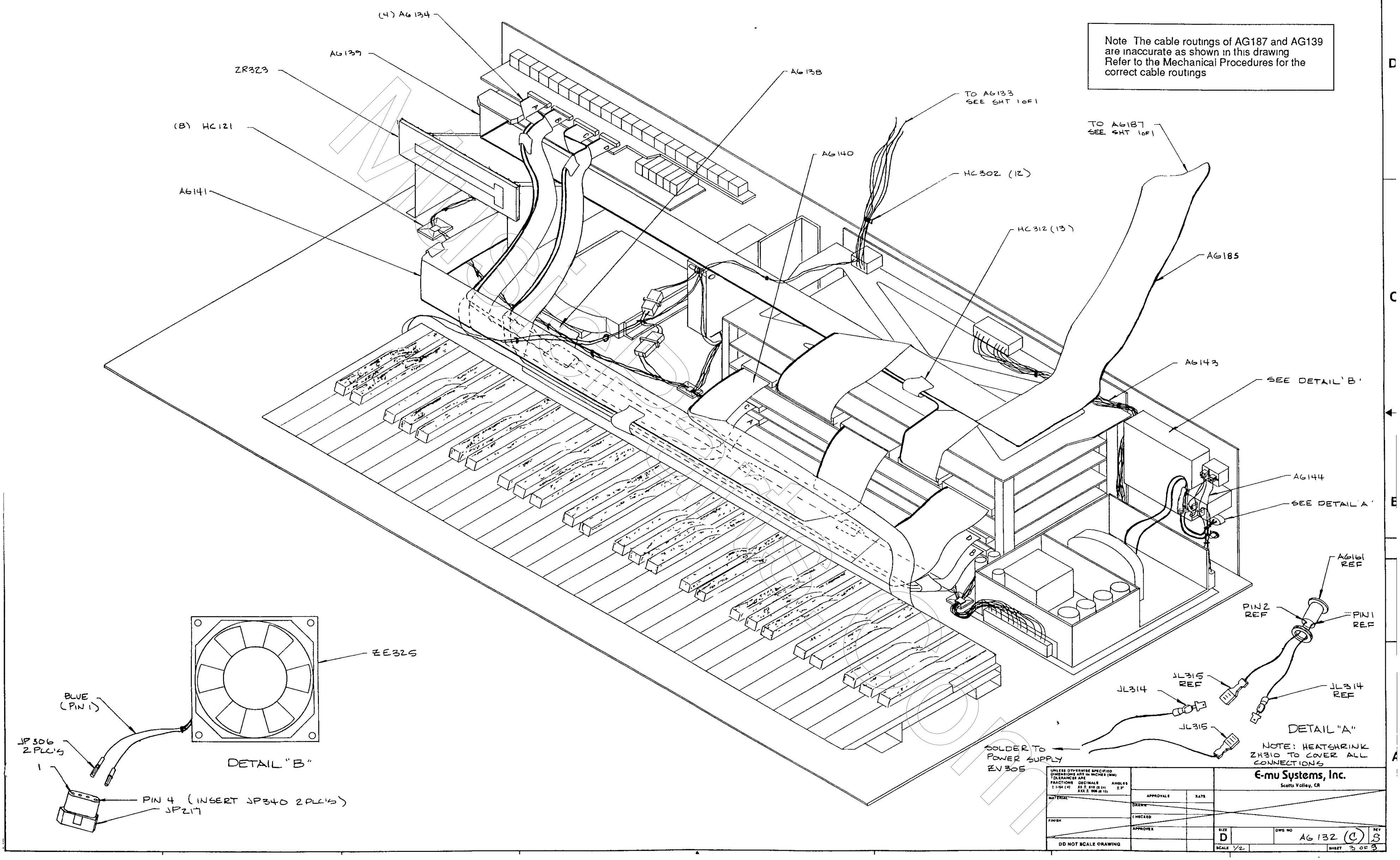
SCALE 1/4

SHEET 2 OF 3

NOTES UNLESS OTHERWISE SPECIFIED

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR D/DATE
/	/	/	SEE SHT 1 OF 3	

Note The cable routings of AG187 and AG139 are inaccurate as shown in this drawing Refer to the Mechanical Procedures for the correct cable routings



UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES (MM)
TOLERANCES ARE
FRACTIONS ± 1/64 (.4)
DECIMALS XX ± .010 (0.24) XXX ± .005 (0.13)
ANGLES ± 2°

MATERIAL

FINISH

DO NOT SCALE DRAWING

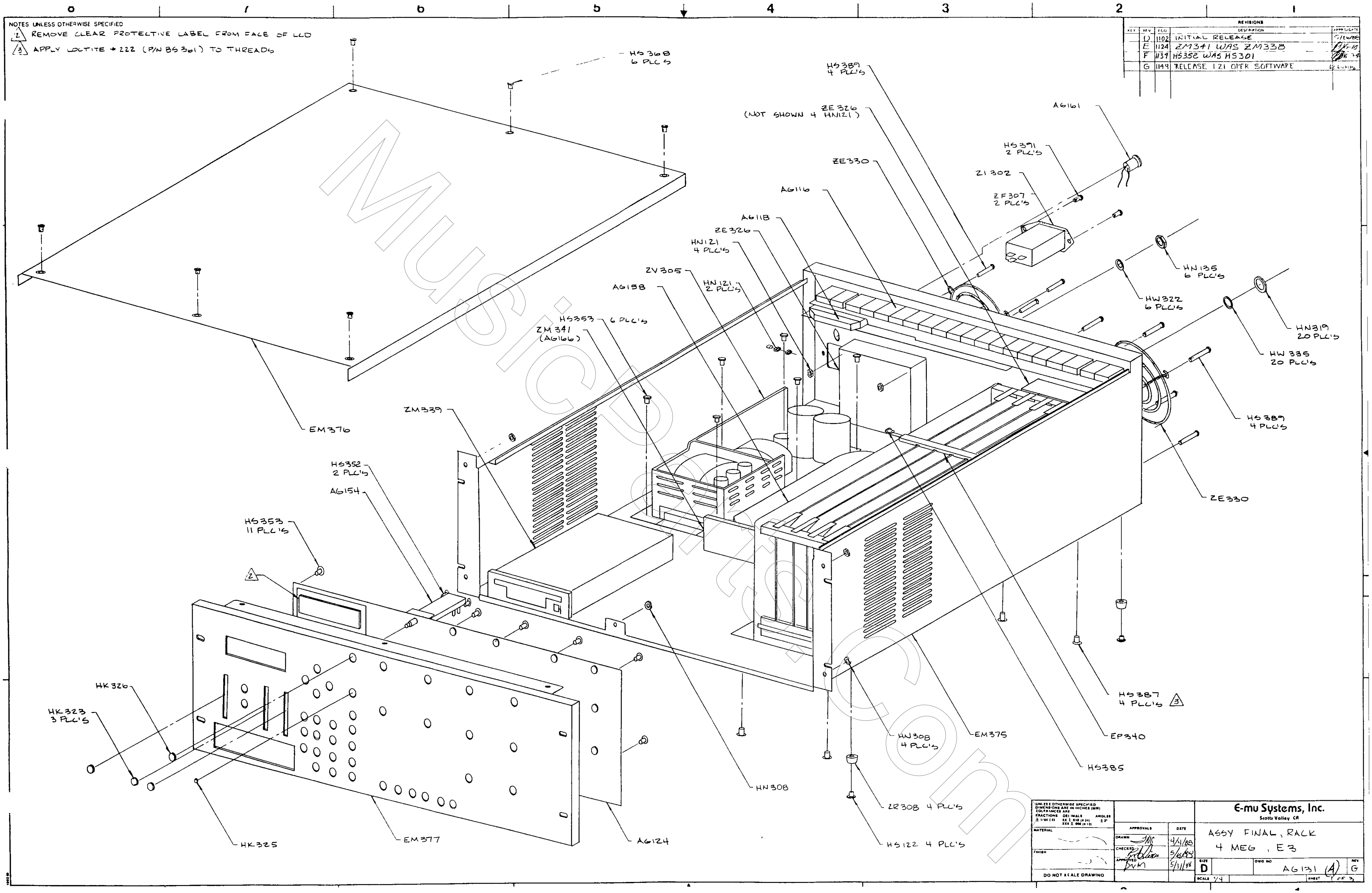
APPROVALS	DATE
DRAWN	
CHECKED	
APPROVED	

E-mu Systems, Inc.
Scotts Valley, CA

SIZE D
DWG NO AG 132 (C)
REV S
SCALE 1/2
SHEET 3 OF 3

NOTES UNLESS OTHERWISE SPECIFIED
2. REMOVE CLEAR PROTECTIVE LABEL FROM FACE OF LCD
3. APPLY LOCTITE #222 (P/N BS361) TO THREADS

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR/DATE
	D	1102	INITIAL RELEASE	5/12WEE
	E	1124	ZM341 WAS ZM338	5-10
	F	1139	HS352 WAS HS301	24
	G	1149	RELEASE 1.21 OPER SOFTWARE	[illegible]

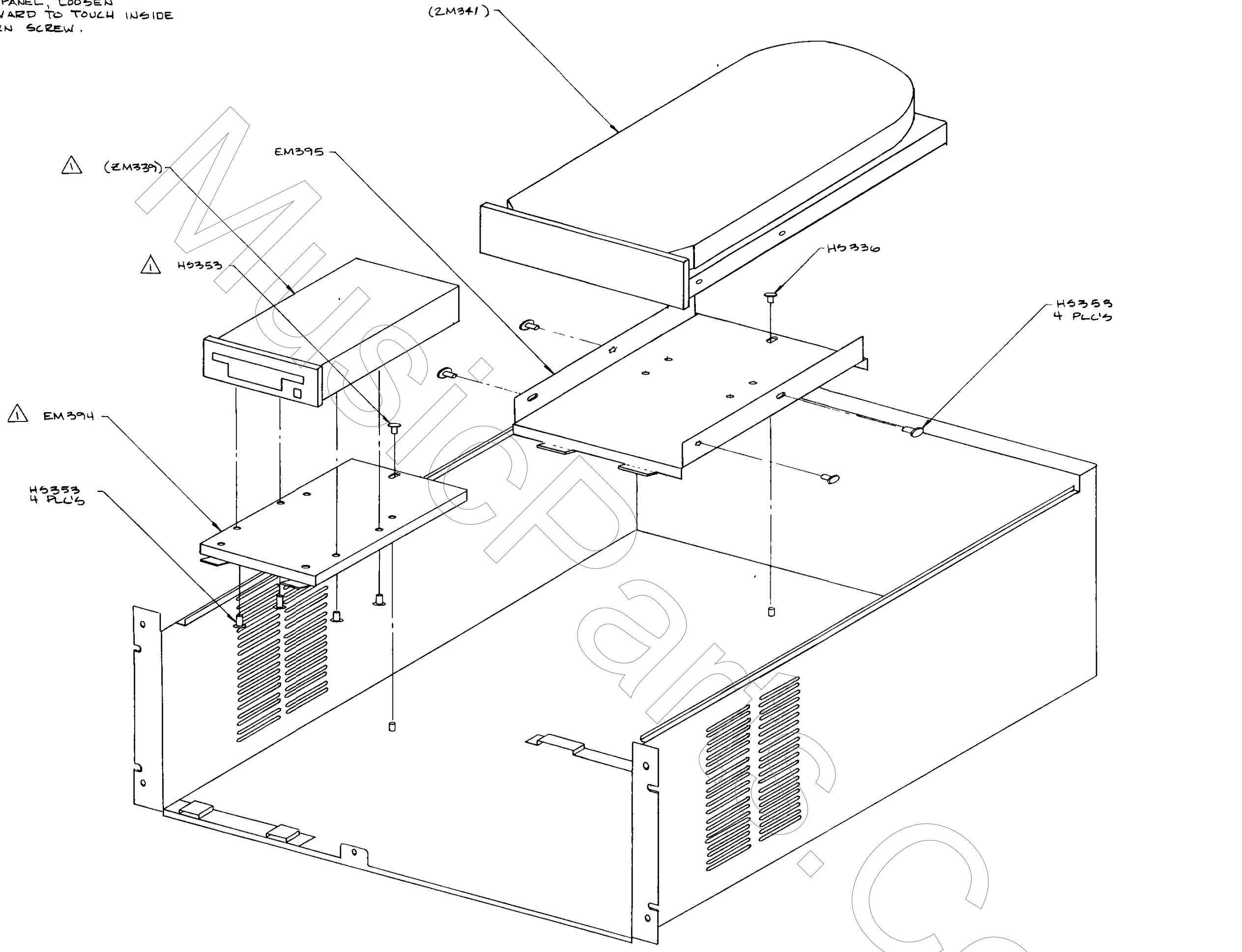


UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES (MM) TOLERANCES ARE			APPROVALS		DATE	E-mu Systems, Inc. Scotts Valley CA
FRACTIONS ± 1/64 (.4)	DECIMALS XX ± .010 (0.24) XXX ± .005 (0.13)	ANGLES ± 2°	DRAWN	JMC	4/4/89	ASSY FINAL, RACK 4 MEG, E3
MATERIAL			CHECKED		5/16/89	
FINISH			APPROVED	DVM	5/31/98	SIZE D · DWG NO AG131 (A) · REV G
DO NOT SCALE DRAWING						SCALE 1/4 · SHEET 1 OF 3

NOTES UNLESS OTHERWISE SPECIFIED

⚠ INSTALL DRIVE TO MAX REARWARD POSITION, USE SLOT. THERE AFTER MOUNTING FRONT PANEL, LOOSEN SCREW AND SLIDE DRIVE FORWARD TO TOUCH INSIDE OF FRONT PANEL RETIGHTEN SCREW.

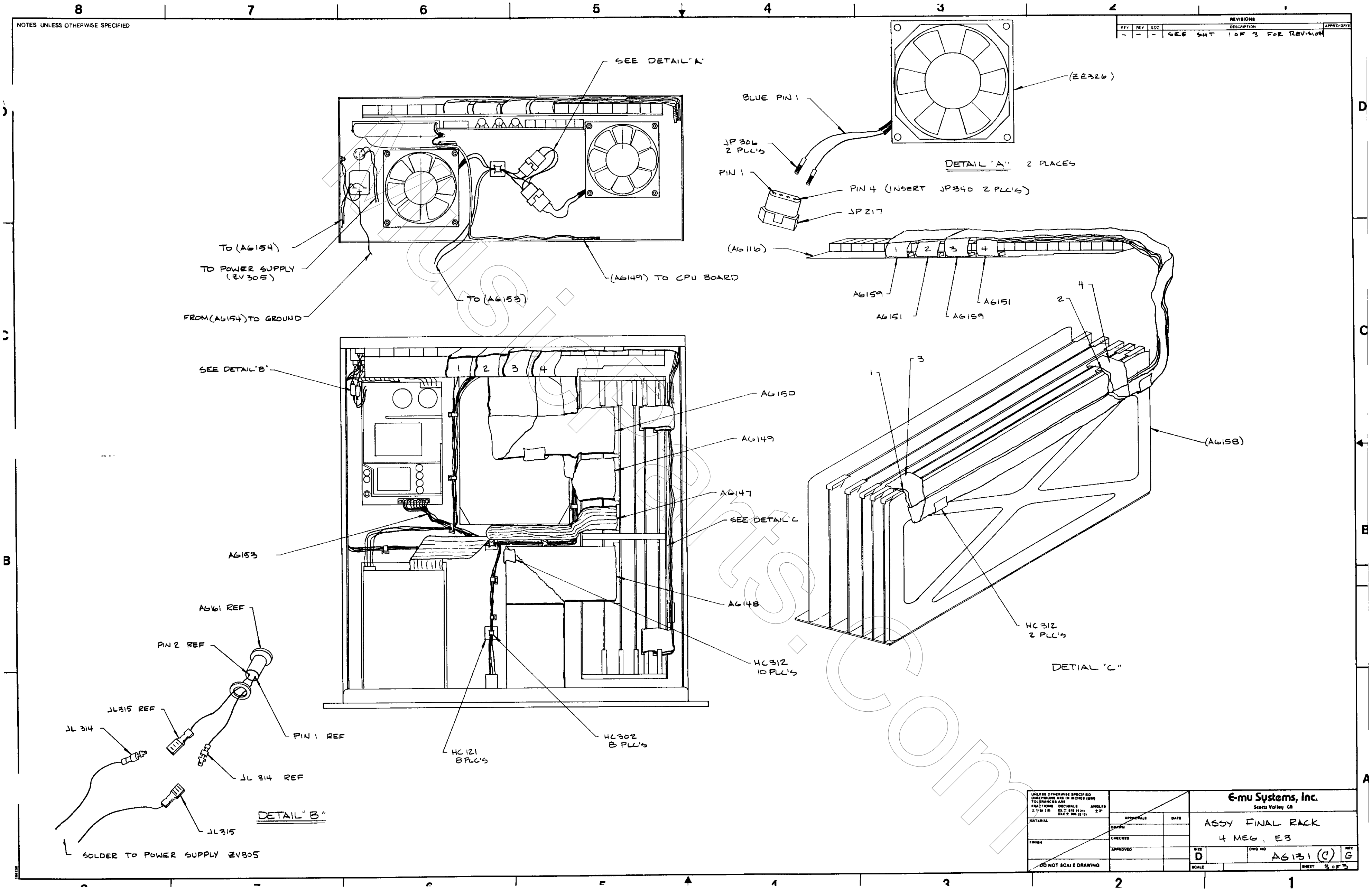
REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPROV/DATE



UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES (MM) TOLERANCES ARE FRACTIONS ± 1/64 (±.4) DECIMALS XX ± .010 (±.24) XXX ± .005 (±.12) ANGLES ± 2°	APPROVALS	DATE	E-mu Systems, Inc. Scotts Valley CA
MATERIAL	DRAWN		ASSY FINAL, RACK 4 MEG, E3
FINISH	CHECKED		SIZE D
	APPROVED		DWG NO A6131 (B)
DO NOT SCALE DRAWING			REV. G
			SCALE
			SHEET 2 OF 3

NOTES UNLESS OTHERWISE SPECIFIED

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR'D/DATE
-	-	-	SEE SHT 1 OF 3 FOR REVISION	



UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES (MM)
TOLERANCES ARE
FRACTIONS ± 1/64 (.8)
DECIMALS XX ± .010 (±.24) XXX ± .005 (±.12)
ANGLES ± 2°

MATERIAL

FINISH

DO NOT SCALE DRAWING

APPROVALS	DATE
DRAWN	
CHECKED	
APPROVED	

E-mu Systems, Inc.
Scotts Valley CA

ASSY FINAL RACK
4 MEG, E3

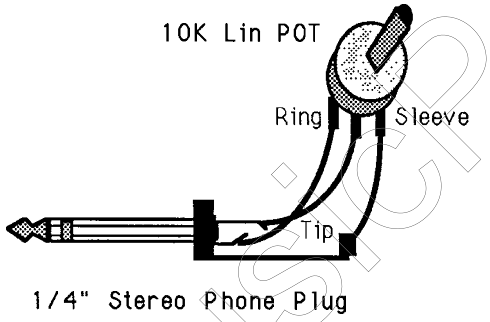
SIZE		DWG NO	REV
D		AG131 (C)	G
SCALE		SHEET 3 OF 3	

FUNCTIONAL TEST PROCEDURE

This functional test procedure is very similar to the E-mu factory final test for a new EIII. A functional test can be performed to verify that a problem exists or as a final test to verify that the unit is working perfectly before you return it to your customer. Instruments to be serviced should be thoroughly tested beforehand. This will verify that a malfunction has indeed occurred, perhaps reveal related or unrelated malfunctions and provide a basis for troubleshooting. When performing these tests, remember to play all 16 channels. Most of the tests are performed by ear, so you will need at least one of these in addition to the equipment listed below.

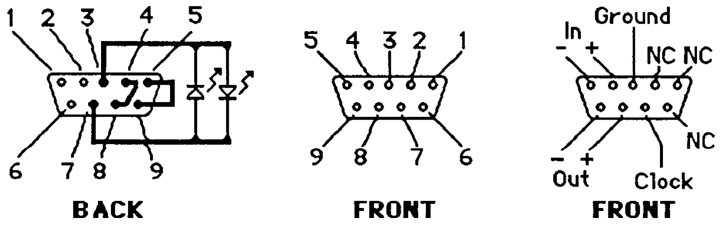
Equipment needed: Amp and Headphones
2 mono audio cables
MIDI cable
Sample source
Sync source
Footpedal Simulator
Footswitch
MUST disk
Scratch diskette

Construction of the Footpedal Simulator:
To test and calibrate EIII's footpedal input, either a footpedal (voltage or resistance type; see Interface Specification), or a footpedal simulator is required. The construction of a footpedal simulator is described below.



CONSTRUCTION OF THE FOOTPEDAL SIMULATOR

Construction of the RS-422 Test Plug: This will come in handy if you plan to repair many EIII's or Emax's. It allows you to test the USART and Clock in one operation.
To build the test plug, you will need: a 9 pin D-type female RS-422 connector, 2 LEDs, and 2 small pieces of wire. Wire the test plug as per the diagram below.



FUNCTIONAL TEST PROCEDURE

□ VISUAL INSPECTION

A. Scratches, blemishes, etc. on housing, label?

B. Scratched keys?

C. Wheels rub?

D. Check for keyboard rattle.

□ POWER UP/DOWN OUTPUT RELAY, GLITCH TEST

(Checks audio output relay)

A. Listen to stereo outputs.

B. Power up/down.

C. Confirm no output.

D. Repeat 3 times.

E. Glitch spec is 250mV (measure w/scope if glitch seems unusually loud).

□ POWER-UP INSPECTION

A. Re-boot from hard drive.

B. Look at LCD carefully for boot PROM rev. (firmware).

C. Watch for proper RAM software revision.

D. Check for diagnostic error.

E. Listen to pattern of sound of HD spin-up.

F. Look for smoke (!).

G. Look and listen carefully!

ENTER DIAGNOSTICS SUBMODULE (Master, Special, #8, Magic code: 1358)

□ PANEL, FOOTSWITCH, KEYBOARD

A. Press all buttons ten times, check for sticking on label, proper incrementation and correct function.

B. Check all LED's.

C. Check footswitches - use shorted 1/4" phone plug, watch display.

D. Check keyboard - all notes - each for low and high velocity.

□ SERIAL PORTS

A. Connect MIDI loop, MIDI thru LED, RS422 tester.

B. Watch LED while pressing "ENTER" (LED should flash once).

C. Verify tests pass.

■ LOAD UST DISK

□ SOUND CHECK AND CONTROLLER CHECK

A. Select "Sound Check" preset.

B. Play keyboard and verify good sounds.

C. Check Controllers: Left wheel-> Pitch, Right wheel-> LFO pitch, Pressure-> VCF cutoff, Pedal-> Attack, Footswitch 1-> Sustain, Footswitch 2-> Preset Increment.

□ PANNING

A. Select "Panner" preset.

B. The first 16 white keys correspond to the 16 output channels.

C. Play all 16 channels while monitoring stereo output.

D. Verify sound pans fully to each side smoothly (opposite side should have no output).

□ STEREO, MONO, CHANNEL OUTPUT JACKS

A. Select "Panner" preset.

B. Monitor mono output.

C. Play sound, verify steady output.

D. Monitor stereo output with headphones.

E. Verify stereo output.

F. "Channel jacks" preset.

G. Play first 16 white keys. Monitor channel output jacks. Verify output in each channel out.

❑ CHORUS

A. Select "Chorus" preset.

B. The first 8 white keys correspond to 2 channels each. (First key is channels 1 & 2, second key is 3 & 4, etc.)

C. Play each key and verify proper function.

❑ FILTER CHECKS

A note on the Filter presets: The first 16 white keys are channel-assigned white noise corresponding to the 16 output channels. The 17th key (E3) is white noise on all 16 channels for quick comparison purposes. The next 16 keys (F3 -G5) are channel-assigned ramp waves, with key A5 on all 16 channels.

A. Listen to the filter presets on all channels and listen for inconsistency, strangeness, etc.

B. If necessary, go into trim module and adjust, remembering to save new table. The Master Software Disk for the machine will need to be updated.

❑ SINE WAVES

A note on sine wave presets: The first 16 white keys are channel assigned to the corresponding 16 output channels.
Listen to all sine wave presets and listen for noise, amplitude variations, etc.

❑ KEYBOARD TRIGGER

A. Select "Vibes" preset.

B. Press each key lightly, checking that each key triggers before hitting key stop.

❑ METRONOME

A. Select a "musical" preset.

B. Monitor metronome output jack.

C. Press sequence "record", then "play".

D. Verify metronome clicks.

E. Play notes.

F. Verify they are recorded.

❑ AUDIO TRIGGER

A. Enter audio trigger module.

B. Turn on left and right sample inputs.

C. Select a note within preset range for both channels.

D. Insert an audio signal into sample input L/R.

E. Adjust threshold accordingly.

F. Verify proper function.

❑ SAMPLE TESTS

A. Erase bank (Master, #3).

B. Silent sample - full bank, approx. 0db.

C. Unusually noisy?

D. 50Hz,1KHz,15KHz, sweep samples - sound good? Any noise?

❑ HARD DISK TESTS

Load and Save bank to and from HD if space permits, verifying good sounds on each bank.

❑ RAM CHECK

Perform the sound RAM test in diagnostics at this point.

❑ SHAKE TEST

Listen for loose or rattling parts.

U.S.T. DISK

The U.S.T. disk (Universal Sound Test) disk is provided to facilitate easier and more thorough functional tests and troubleshooting. The disk contains various presets which have EIII parameter set-ups programmed into them. When used in conjunction with the Functional Test Procedure, the U.S.T. disk provides a very complete test of the EIII functions. We recommend that you play other banks (such as the Piano) in addition to the U.S.T. disk after performing your functional test in order to make certain that the unit is problem free.

Note: It is a good idea to keep another floppy disk updated with the most recent version of EIII software so that you can update a customer's unit when necessary.

U.S.T. PRESETS

PRESET	TITLE	FUNCTION
P01	Sound Check	First 16 white keys = 16 channels, 17th = all
P02	Panner	Tests panning function
P03	Chorus	Tests chorus function: First 8 white keys
P04	Sine 1378 Hz	Troubleshooting and channel comparison
P05	315 Hz Sine	Troubleshooting and channel comparison
P06	100 Hz Sine	Troubleshooting and channel comparison
P07	21.5 Hz Sine	Troubleshooting and channel comparison
P08	9999 Hz Sine	Troubleshooting and channel comparison
P09	Channel Jack	First 16 white keys = 16 channels
P10	Filter Cutoff	Filter trims and channel comparison
P11	Filter Mid Q	Filter trims and channel comparison
P12	Filter Max Q	Filter trims and channel comparison
P13	Filter Resonance	Filter trims and channel comparison
P14	Filter Sweep	Channel comparison
P15	Filter Sweep Q	Channel comparison
P16	500 Hz Sine Full	Computer generated full 16-bit wave
P17	Ramp Wave Synth	Listening and comparison
P18	Square Wave Synth	Listening and comparison
P19	Vibes (Full)	Playing, full keyboard
P20	Pink Filter	Filter trims and channel comparison
P21	Pink Filter Mid Q	Filter trims and channel comparison
P22	Pink Filter Max Q	Filter trims and channel comparison
P23	Pressure Check	Pressure calibration
P24	Ambient Vibe	Playing
P25	Wheel Check	Verify wheel operation
P26	Pedal Check	Verify pedal operation
P27	Channel Check	Output channel check
P28	500 Hz Sine 60%	Trim Procedure
P99	Drum Set	Sequencing, playing, etc.

DIAGNOSTICS and TRIMS

The Emulator Three is made up of literally thousands of individual components. Because of its complexity, a good understanding of the operation of the EIII will help immensely when troubleshooting. We suggest you first perform a functional test on the unit. Many times a complete functional test will reveal important clues to the problem that might have otherwise been overlooked. Try to isolate the problem as much as possible through the operational controls, then go in with your instruments to nail it down. The Channel Disable function is an invaluable tool to determine which channel is at fault. A high quality amplifier and speaker system is also necessary to pick out subtle problems that would otherwise be missed.

Certain boards on the Emulator Three can be more difficult to troubleshoot than others. The microcontroller board because of its complexity can be a nightmare to diagnose and repair. In many cases, if you have isolated the board but cannot seem to locate the exact problem, a board swap is in order. Simply call the E-mu Customer Service department to arrange for a board swap.

The EIII contains a number of on-board diagnostic tests and trim procedures. These can be accessed or read from the front panel without even opening the unit!

SPECIAL FUNCTION DIAGNOSTICS

The on-board tests are hidden in the Special functions and can be accessed in the following manner:

1. Activate **MASTER 8** (Special). The display will say:

SPECIAL

Select a Submodule

2. Press **8** on the keypad (Software Version 1.21 and above).

Hint: A good way to remember this is: **1 number past the last choice.**

The display will now say:

DIAGNOSTICS

Enter MagicCode:

NOW LISTEN CLOSELY!

Pssst, the secret code is: **1-3-5-8** or ... the intervals of a major chord.

The display will now say:

DIAGNOSTICS

Select a Submodule

Moving the data slider will show the following tests and calibrations:

1. Panel/Keyboard Test
2. Sound RAM Test
3. Serial Test
4. Freq. Response Test
5. Analog Trims
6. ADC Trims
7. Sample Gain Check
8. Disk Diagnostics
9. Disable Channels

1 PANEL/KEYBOARD TEST

1. Flashes the LCD with all pixels on to test the LCD.

2. All button names are displayed to the LCD as they are pressed, and all buttons toggle their corresponding LED. In addition, keyboard note on and note off as well as key velocity are displayed.

2 SOUND RAM TEST

1. Tests Sound RAM for shorted address lines, shorted data lines, crossed bits, and address to data problems.

2. Displays the amount of RAM installed and does a 3 word pattern Write/Read comb of the entire installed RAM. Flags all errors.

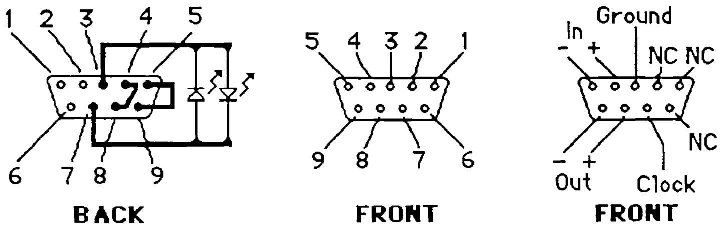
Note: Reboot the EIII after running a RAM test or sounds may distort. There is garbage in memory.

3 SERIAL TEST (loop through test)

1. MIDI - tests the signal path and the SCC in its intended operating mode.

2. RS422 - This test writes and reads an AA and then a 55 to the RS-422 port. The test waits a reasonable length of time, for each response. If there is no response, it records a timeout error. In order for the test to work, a jumper plug must be inserted into the RS-422 jack on the back panel of EIII. The LEDs show that the 500 kHz clock is running. If you do not wish to make the test plug, you can simply connect pins 4 and 8, and pins 5 and 9 together with clip leads. To test the 500 kHz clock, monitor pin 7 with your oscilloscope.

Construction of the RS-422 Test Plug: This will come in handy if you plan to repair many EIII's or Emax's. It allows you to test the USART and Clock in one operation.
To build the test plug, you will need: a 9 pin D-type female RS-422 connector, 2 LEDs, and 2 small pieces of wire. Wire the test plug as per the diagram below.



3. SMPTE- Test still N/A as the SCC must be in two different modes to read and write SMPTE time code.

To Test SMPTE: First, stripe a multitrack tape with SMPTE, then record a simple sequence in the EIII sequencer. Extend the sequence using song mode so that it is several minutes long. Next record the sequence to another track of the tape with the sequencer synced to SMPTE. Lastly, rewind the tape and sync the EIII up again, this time monitoring both the recorded track and the EIII. Slight flanging is OK but the two "tracks" must remain in sync.

4 FREQUENCY RESPONSE TEST

The frequency response test is a quick way to test all the EIII channels at once.
Simply follow the instructions given in the display.

```
 FREQUENCY RESPONSE
Hook MonoMix->R.Smpl
Set SmplInputLevel=0
Press ENTER to start
```

When ENTER has been pressed, the EIII will measure the frequency response of all channels, then the display will show either:

```
 FREQUENCY RESPONSE

 All Channels:   OK
```

or

```
 FREQUENCY RESPONSE
0 = Dead Channel(s)
X =  Bad Channel(s)
0000 0000 0000 0000
```

1. Loop-through test

2. Checks peak-peak amplitude response of all 16 channels at 32Hz, 1kHz, and 16kHz.

3. Compares to make sure p-p voltages at the 3 test frequencies are within 6dB (2X p-p) of each other.

4. Also checks for dead channels. Compares signal amplitude to 2X the amplitude of the noise floor (no signal).

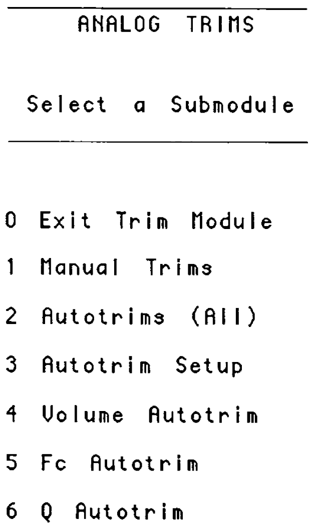
5. Flags good, bad and dead channels.

5 ANALOG TRIMS

The analog trims on the EIII are stored with the software on the hard disk. After trimming a customer's machine, it would be a good idea to save the system software on a 3.5" floppy disk and give this to the customer so that they will have a backup in case of hard disk failure (Disk Utilities, 4).

Rules relating to saving and loading trims:
Rule #1: Saving software from HD to floppy saves the trims *and* software to the floppy.
Rule #2: Loading software from floppy to a HD with software resident *does not* load trims.
Rule #3: Loading software from floppy to a HD without software resident *does* load trims.
So, to load the trims from a floppy disk to the HD, you must erase software from the HD first.

The Analog Trim menu shows:



```
      ANALOG TRIMS

   Select a Submodule


0 Exit Trim Module
1 Manual Trims
2 Autotrims (All)
3 Autotrim Setup
4 Volume Autotrim
5 Fc Autotrim
6 Q Autotrim
```

ANALOG TRIM PROCEDURE

Tools Needed:

1. 12/30 UST (Universal Sound Test) disk
2. Short 1/4" phone cable
3. Oscilloscope

Information you need to know:

The lowest 16 keys correspond to channels 1-16 respectively.
The 17th key (E3) cycles through all 16 channels.

Before you Start:

1. Make sure that the sliders are calibrated. Master, Special, 3.
2. Save all trims after each step.
3. Warm up the EIII (closed) for at least 1 hour before you begin this procedure.

1. Load the UST disk.

2. Go to preset 16, "500Hz sine wave, 100% VCA level".

3. Go to "Analog Trims" in this way.

A. Choose Master module.
B. Choose Special (8) submodule.
C. Choose Diagnostics (8) (It's not on the menu. Remember it's 1 number past the last choice.)
D. The display will respond, "Enter Magic Code".
E. Enter 1-3-5-8 (Major Chord).
F. Choose Analog Trims (5).
G. Choose Manual Trims (1).

4. You will now see a screen of many numbers.

```
Analog  Trims  Ch:01
Fc:96909393  85818590
  Q:80808585  86808179
Vo:81837896  88839980
```

The top row represents "Fc".
The middle row represents the "Q" of the filter.
The bottom row represents the "Amplitude or Volume" of the channel.
There are two pages of these numbers representing all 16 channels.

5. With the volume slider at maximum, measure the output at the Mono out jack, using your oscilloscope. All channels should measure at least 4V p-p (If E.C.O.#1036 has been installed, otherwise levels should be 1Vp-p).

6. Amplitude Trim.

A. Press Master again to exit the module.
B. Choose preset 28 (500Hz sine, 60% VCA level).
C. Enter the Diagnostics submodule again and choose Analog Trims.
D. Locate the channel with the lowest output and leave its value at 99.
E. Trim the other channels to match it.
F. Hit "Enter". Save New Trims? YES.

7. Resonance.

A. Select preset 13 (Filter Resonance).
B. Enter the Diagnostics submodule again and choose Analog Trims.
C. Press the first 16 keys and locate the dullest sounding channel. This will be our reference channel.
D. Lower the dullest channel's Fc trim to 95. (This gives you some leeway which you may need later on).
E. Match the other channels to this one.
F. Hit "Enter". Save New Trims? YES.

8. Q trim.

A. Without exiting this module, and using the first 16 keys to listen, lower the Q values until the filter just drops out of oscillation. You should hear a hollow sound instead of a squeal.
B. Hit "Enter". Save New Trims? YES.

9. Fine tune Fc.

A. Exit Master module and select preset 10 (Filter Cutoff).
B. Tune all channels to the reference channel (as closely as possible).
C. Hit "Enter". Save New Trims? YES.

Manual trims

1. Allows the technician to trim (by ear or test gear) the Volume, Fc, and Q of the CEM VCA/VCF for each channel of the EIII. Trim range is 0 to 99. Volume and Q trims should be as high as possible (least effect). Fc trims will vary as to trim each machine to the displayed frequency in the Analog Processing module.

Autotrims

Autotrims may be useful on a completely uncalibrated machine in order to get the trims "in the ballpark". Unfortunately, the Autotrim procedure is not as accurate as trimming by ear and is not recommended except in cases where saving time is a factor. To Autotrim, simply follow the instructions given in the display.

1. A loop-thru test to automatically trim the CEM chips. The Sample Input Gain is first set by using a level calibrated test signal. With this known input reference each channel is then trimmed in order Volume, Fc, and Q.

2. Channels can be disabled if necessary to allow autotrimming a machine with known bad channels, which might otherwise cause the test to abort.

6 ADC TRIM PROCEDURE

Tools Needed:

1. Signal Generator (1kHz sine wave at 2.82 Volts peak-peak)
3. Long Trimpot Tweeker
4. Several mono phone cables

Information you need to know:

1. Make sure ECO 1036 has been implemented (220pf cap at C2 and C3; 10K resistor at R4 and R5 on analog jack board). This ECO increases the mix output level.
2. A trimpot location diagram is shown at the end of this section.

Before you Start:

1. Warm up the EIII for at least 1 hour. It is important to trim the EIII ADC board when it is at its normal operating temperature.
2. Ensure that the Analog Trims are O.K.

1. Set "Sample Input" slider to -10dB by the following steps:

A. Enter "Sample Gain Check" mode:

1. Choose Master module.
2. Choose Special (8) submodule.
3. Choose Diagnostics (8) (It's not on the menu. Remember it's 1 number past the last choice.)
4. The display will respond, "Enter Magic Code".
5. Enter 1-3-5-8 (Major Chord).
6. Choose Sample Gain Check (7).

B. Inject a 1kHz 2.82Vpp sine wave into the left Sample Input jack. Measure at the right side of R8 (which is not stuffed) on the analog jack board or measure with a standard Y-splitter cord.
C. Set the Sample Input slider so that the EIII's VU reads -10dB. (To avoid having to repeat this procedure, index the slider position with a light pencil mark on the front panel label.

2. Go to ADC Trims in Diagnostics.

A. Follow the screen instructions.

```
        DC TRIM
Remove cables from
Set SmplInputLevel=0
Press ENTER to start
```

B. Adjust the DC offset trimpots for minimum offset.

```
       ZERO TRIM
Left:+065  Right:+003
L:||||||||
R:||||
```

C. Press "Enter".

D. Follow the screen instructions.

```
ADC TRIM       Ch:Left
Connect Mono Mix Out
to Left Sample Input
Press ENTER to start
```

E. While looking at the bar graph, adjust the left channel slope trimpot (RT4) for the most stable settings.

```
2ndHD TRIM    Ch:Left
Bar=.04%  Avg2hd=.08%
```

F. Adjust left 2nd Harmonic trimpot (RT1) for minimum overall distortion.

G. Alternate between steps E and F for an optimum reading. All channels must be below the midpoint of the bargraph range (typically below 0.12%).

H. Follow the screen instructions and repeat for the right input.
Right input.
3 Right DC Offset Trimpot (RT6).
1 Right Slope Trimpot (RT5).
2 Right 2nd Harmonic Trimpot (RT2).

I. Start over at step 2 and repeat at least once. This is important as these adjustments are interactive.

On early EIII units, the trimpots are not mounted vertically on the ADC board and are very hard to reach. Technicians with very thin hands may be able to reach the trimpots when the ADC board is in the cage. If you are working on an early EIII with the hard to reach trimpots you may wish to change the trimmers to the upright variety (This is covered under warranty). If this is not practical, the ADC board may be trimmed while out on the extender board. Obviously, when the ADC board is out on the extender, it is going to cool down. One technique to overcome this is, to feel the temperature of the ADC board ICs after it has been on a while, then try to simulate this temperature with a heat gun when it is out on the extender board. Be extremely careful not to overheat and damage the ICs when using a heat gun. After trimming, close the machine up and watch for any change in the diagnostic displays.

DC offset (zero) trim
Sets the center point of the ADC. The technician should be aware that this will drift considerably with temperature and the sampling routines will compensate for this. It is trimmed here for nominal center and to ensure a proper slope trim.

Slope trim
Sets the slope of the ADC current generator at low signal levels.

2nd harmonic distortion (symmetry) trim
Adjusts the symmetry of the dbx 2155 input amp. Unevenness in the positive and negative half-cycles will cause even harmonic distortion, hence we can trim this by monitoring the 2nd harmonic. As the CEM 3387 VCF/VCA adds its own 2nd harmonic distortion, we use a loop-thru type of test to minimize 2nd harmonic distortion system throughput. The test uses all 16 channels to let the technician average the distortion, that is, get the best compromise for the system. Pressing the MASTER button (this is not labeled on the display) demultiplexes the test and displays only channel 1, allowing a quick comparison with outboard test equipment.

7 SAMPLE GAIN CHECK

Displays left and right calibrated VU meters to check the gain of the sample input path. 0 VU = 16 bits = 2^16 = full scale (consider a digital audio system as having no headroom). Using a 0dBV = 2.82Vpp 1kHz sinewave input signal as a reference, verify that the sample input path has approximately 40dB gain at maximum slider setting to approximately -15 to -20dB attenuation at the lowest slider position.

8 DISK DIAGNOSTICS

0 Exit Disk Module- Returns you to the Diagnostics submodule.

1 HD Select Drive- Allows you to select any currently mounted drive.

2 HD Read Only- Non-Destructive. Exercises (reads) the entire HD media for data read errors. Runs continuously. Press and hold ENTER to quit. Exiting the drive in this manner sets the drive Error Correction to ON.

3 HD Read/Reassign- Potentially Destructive. Same as Read Only, but first turns error correction OFF and reassigns bad blocks using the drive block reassignment.

4 HD Write/Read/Reass- DESTROYS ALL DATA. Exercises the entire HD media by writing a test pattern, reading it back, and comparing. Runs continuously. Disk must currently be reformatted after this test with the standard EIII disk utility. This installs the file system to allow the EIII to recognize the drive. Press and hold ENTER to quit.

5 HD Error Correct - Allows investigation and change of the drive error correction state.

6 HD Result - SCSI Sense Key and Sense Code, Sector number and Status of the last HD operation.

7 HD Media Defects - Displays HD's defect list in Hex.

8 Floppy Read Only - Continuously reads the entire floppy surface and verifies to a known test pattern. Must use a disk previously written using the Floppy Write/Verify test. Logs all soft errors according to sector number, byte number, and data compared. Useful for read exercising and for checking drive to drive compatibility (alignment, etc.).

9 Floppy Write/Verify - Runs continuously. Writes a test pattern and reads it back, comparing the data. Logs errors as in the read test. The test pattern is an ascending pattern (i.e. 104, 105, 106, ...) The first byte in the sector is the sector number mod 256. Each sector has 512 data bytes. There are 10 sectors * 80 tracks * 2 sides = 1600 sectors per disk.

IMPORTANT- Powering down the EIII while it is running a HD continuous test will leave the drive Error Correction OFF! Make sure the Error Correction is turned ON when finished.

9 DISABLE CHANNELS

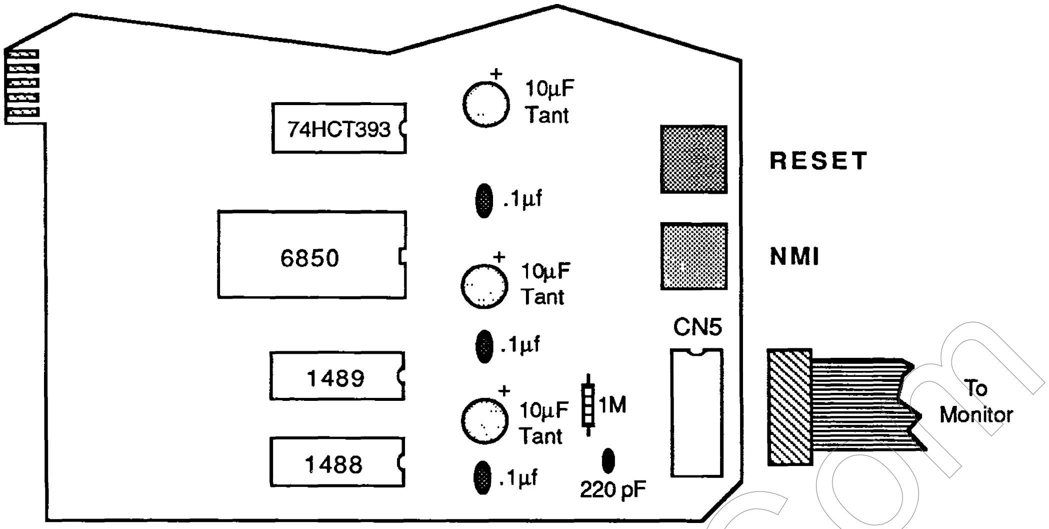
Allows the technician to disable specific output channels without having to leave the Diagnostics submodule. Channels can also be disabled by the user in the MASTER, Special, submodule.

THE DEBUG/EXTENDER BOARD

The EIII Extender board serves as both an extender board and as a powerful debugging tool when coupled with a "smart" computer monitor. Many E-mu Service Centers will already have a monitor which was used in debugging the Emulator II. Most EIII problems can be easily traced without getting down to the "nuts and bolts" of the machine, but every once in a while a problem will arise which defies normal troubleshooting techniques. In these cases, the debugger can be an invaluable tool.

The Debug board

The Debug board can be used as an interface from your monitor to the Dbmon program which is resident in the EIII boot PROMs. Since the Debug boards do not have the interface parts installed by E-mu, you will have to stuff them yourself if you wish to use the monitor function. The parts layout is shown in the following diagram. The schematic diagram is shown in the schematic diagram section of this manual. All parts for the Debugger are commonly available but can also be obtained from E-mu Systems. In addition to these parts you will need a 16-pin ribbon cable which interfaces with your monitor. The pinouts for construction of this cable are also diagramed.
There are two buttons on the Debug board; NMI and RESET. NMI stops the program that is currently running and returns control to the user. RESET will give the same result as turning off the machine then back on again. Any programs that you enter using the Dbmon will be lost is you push RESET.



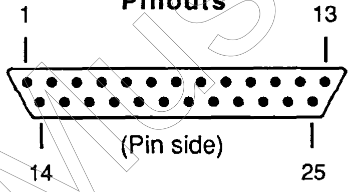
The Monitor

You will need a monitor with a keyboard to use the Debug board. The communication protocol parameters should be set as follows:

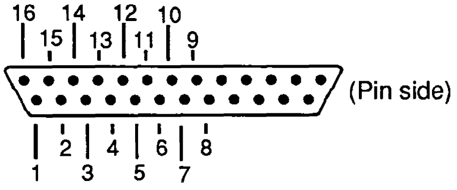
Baud Rate: 9600
Parity: Even
Half/Full Duplex: Full
Hardware/Software Handshake: Hardware

You will also need an interface cable between the Debug board and the monitor. The Debug board interfaces with a 16 pin IC socket. DB 25 to 16 pin is not a very common interface so you will have to build the interface cable yourself. A male DB25 connector with a ribbon cable connection on the back is readily available at Radio Shack and other parts supply houses. The wiring diagram for the cable is shown below.

Male DB 25 Connector Pinouts



Connections to 16 pin Socket



If you do not have a monitor and wish to purchase one, may we recommend the Radio Shack TRS-80 Model 100. This is a battery operated portable computer which has built-in BASIC, a built-in modem and can be used for word processing and other applications (such as a monitor). The screen size is a bit small, but otherwise the computer serves well. This computer is the model that our test and service departments use and so we have included the setup procedures for the terminal mode on the TRS-80 Model 100.

TRS-80 Model 100 Terminal Mode

1. Connect the interface cable to the computer and turn it on. You should get a menu with BASIC highlighted. If BASIC is not highlighted, push the reset button on the back of the computer.
2. With BASIC highlighted, type the right arrow key twice to select "telecom". Press ENTER and the display will show something like this. These are the terminal settings.

```
88N1E,10 pps
Telecom:
```

(Actually, it won't look like this if you are first setting it up, but this is how we want it to look .) Press F4 to enter the terminal mode.

3. If the settings were not as shown, press F3 to change the terminal settings. The display will say "STAT". Now type in the new settings.

```
88N1E,10 [ENTER] [F4]
```

Now you are in terminal mode.

GETTING STARTED

A few examples of how to read and write to the EIII memory locations are given below. Try writing to the LED latch first in order to test the system. See the Dbmon documents at the end of this section for detailed information on using the Dbmon program.

To Invoke dBmon:

Set up your video monitor and press the NMI switch on the Debug board. The display will say NMI and show an asterisk *.

[sp]= space   [CR]= Return

To Test Memory:

T2 [sp] [start addr] [# of bytes] [CR]

To test from 400000 to 4000FF you would type: T2 [sp] [400000] [FF] [CR]

To Fill Memory:

F [sp] [start addr] [stop addr] [CR]

To fill from 400000 to 480000 with FF, you would type: F [sp] [400000] [480000] [FF] [CR]

To Change a Memory Byte: (Write to a memory location)

CMB [sp] [addr]=[new byte] [CR]

Use **CMW** to Change a Memory *Word*

To turn the panel LEDs on, you would type: CMW [sp] DB0000=FFFF [CR]

To Display the Contents of Memory:

D [sp] [start addr] [# of bytes] [CR]

EIII MEMORY LOCATIONS

These Emulator Three memory locations are provided to assist the advanced troubleshooter. The RAM locations will probably be the most frequently used. Writing, then reading from the µController can give confusing results because the state of a given register may be affected by a µController flag setting.

Sound RAM Addresses (4 Megabytes)

IC 64 SIMM Location	400000
IC 58 SIMM Location	480000
IC 57 SIMM Location	47FFFF
IC 56 SIMM Location	4FFFFF
IC 55 SIMM Location	500000
IC 54 SIMM Location	580000
IC 32 SIMM Location	57FFFF
IC 31 SIMM Location	5FFFFF
IC 30 SIMM Location	600000
IC 29 SIMM Location	680000
IC 28 SIMM Location	67FFFF
IC 27 SIMM Location	6FFFFF
IC 04 SIMM Location	700000
IC 03 SIMM Location	780000
IC 02 SIMM Location	77FFFF
IC 01 SIMM Location	7FFFFF

] = SAME 16 BIT WORD

Sound RAM Addresses (8 Megabytes)

IC 64 SIMM Location	400000
IC 58 SIMM Location	600000
IC 57 SIMM Location	5FFFFF
IC 56 SIMM Location	7FFFFF
IC 55 SIMM Location	800000
IC 54 SIMM Location	A00000
IC 32 SIMM Location	9FFFFF
IC 31 SIMM Location	BFFFFF

] = SAME 16 BIT WORD

Floppy Controller

Floppy Status Register	50008
Floppy Track Write	50002
Floppy Track Read	5000A
Floppy Sector Write	50004
Floppy Sector Read	5000C
Floppy Data Write	50006
Floppy Data Read	5000E

LEDs and Misc.

LED Latch	DB0000
Misc Latch	D70000
Metronome	CC0000

Microcontroller

Current Size (Chan 0)	3E8000
Current Size (Chan 1)	3E8010
Current Size (Chan 15)	3E80F0

Initial Size (Chan 0)	3E80**0**4
Current Address (Chan 0)	3E80**0**8
Initial Address (Chan 0)	3E80**0**C

Bold = Channel number

F-Chip Output Freq. (write only)

F-Chip Channel 0	D30010
F-Chip Channel 1	D30012
F-Chip Channel 2	D30014
F-Chip Channel 3	D30016
F-Chip Channel 4	D30018
F-Chip Channel 5	D3001A
F-Chip Channel 6	D3001C
F-Chip Channel 7	D3001E
F-Chip Channel 8	D30020
F-Chip Channel 9	D30022
F-Chip Channel 10	D30024
F-Chip Channel 11	D30026
F-Chip Channel 12	D30028
F-Chip Channel 13	D3002A
F-Chip Channel 14	D3002C
F-Chip Channel 15	D3002E

F-Chip Pitch Divisor Range = 0FFF - 00A4

Control Voltage DAC

Tracking Filter Channel 0	DE0000
Tracking Filter Channel 1	DE0002
Tracking Filter Channel 2	DE0004
Tracking Filter Channel 3	DE0006
Tracking Filter Channel 4	DE0008
Tracking Filter Channel 5	DE000A
Tracking Filter Channel 6	DE000C
Tracking Filter Channel 7	DE000E
Tracking Filter Channel 8	DE0010
Tracking Filter Channel 9	DE0012
Tracking Filter Channel 10	DE0014
Tracking Filter Channel 11	DE0016
Tracking Filter Channel 12	DE0018
Tracking Filter Channel 13	DE001A
Tracking Filter Channel 14	DE001C
Tracking Filter Channel 15	DE001E

4-pole Filter Channel 0	DE0020
4-pole Filter Channel 1	DE0022
4-pole Filter Channel 2	DE0024
4-pole Filter Channel 3	DE0026
4-pole Filter Channel 4	DE0028
4-pole Filter Channel 5	DE002A
4-pole Filter Channel 6	DE002C
4-pole Filter Channel 7	DE002E
4-pole Filter Channel 8	DE0030
4-pole Filter Channel 9	DE0032
4-pole Filter Channel 10	DE0034
4-pole Filter Channel 11	DE0036
4-pole Filter Channel 12	DE0038
4-pole Filter Channel 13	DE003A

4-pole Filter Channel 14	DE003C
4-pole Filter Channel 15	DE003E

Filter Q Channel 0	DE0040
Filter Q Channel 1	DE0042
Filter Q Channel 2	DE0044
Filter Q Channel 3	DE0046
Filter Q Channel 4	DE0048
Filter Q Channel 5	DE004A
Filter Q Channel 6	DE004C
Filter Q Channel 7	DE004E
Filter Q Channel 8	DE0050
Filter Q Channel 9	DE0052
Filter Q Channel 10	DE0054
Filter Q Channel 11	DE0056
Filter Q Channel 12	DE0058
Filter Q Channel 13	DE005A
Filter Q Channel 14	DE005C
Filter Q Channel 15	DE005E

VCA Channel 0	DE0060
VCA Channel 1	DE0062
VCA Channel 2	DE0064
VCA Channel 3	DE0066
VCA Channel 4	DE0068
VCA Channel 5	DE006A
VCA Channel 6	DE006C
VCA Channel 7	DE006E
VCA Channel 8	DE0070
VCA Channel 9	DE0072
VCA Channel 10	DE0074
VCA Channel 11	DE0076
VCA Channel 12	DE0078
VCA Channel 13	DE007A
VCA Channel 14	DE007C
VCA Channel 15	DE007E

Pan Channel 0	DE0080
Pan Channel 1	DE0082
Pan Channel 2	DE0084
Pan Channel 3	DE0086
Pan Channel 4	DE0088
Pan Channel 5	DE008A
Pan Channel 6	DE008C
Pan Channel 7	DE008E
Pan Channel 8	DE0090
Pan Channel 9	DE0092
Pan Channel 10	DE0094
Pan Channel 11	DE0096
Pan Channel 12	DE0098
Pan Channel 13	DE009A
Pan Channel 14	DE009C
Pan Channel 15	DE009E

Scanner

Read Keyboard Scanner Port	F10000
Read Extra Switch Port	FA0000

LCD

Write to LCD Instruction Register	EB0000
Read from LCD Instruction Register	EB0002
Write to LCD Data Register	EB0004
Read from LCD Data Register	EB0006

Interrupts

Clear Sequencer/Sync Interrupt (write only)	C00000
Clear Transient Generator Interrupt (write only)	C40000
Interrupt Vector Port (read only)	FFFE00

POWER SUPPLY SPECIFICATIONS

POWER SUPPLY TEST

EIII uses a switching power supply. We do not repair these supplies at the factory and do not even have schematics for them. If a supply is defective, contact E-mu customer service at (408) 438-1921, to obtain a swap supply.
Note: Switching supplies will not operate without a load.

Green wire is **ground**.

Verify **+12V** on **orange** wire.

Verify **+5V** on **yellow** wire.

Verify **-14.25V to - 15.75V** on **black** wire.

Verify **+14.25V to +15.75V** on **red** wire.

TROUBLESHOOTING GUIDE

When troubleshooting EIII, common sources of problems are connectors, sockets, and broken solder joints. The traces on all EIII boards are very thin. Be extremely careful when desoldering parts from the multi-layer CPU and µController boards. If you are having problems desoldering a component, we suggest that you clip the part out rather than damaging the board. You should also make sure that you are using the latest version of software to avoid chasing phantoms. If you suspect a software problem and you are using the latest version, contact the factory. Most analog problems seem to center around the SSM and Curtis chips (especially SSM 2300). If you get stuck on a problem, please feel free to contact our Customer Service department at (408) 438-1921. They will be happy to assist you. Telephone support hours are between 8:30 am and 5:30 pm PST Monday through Friday.

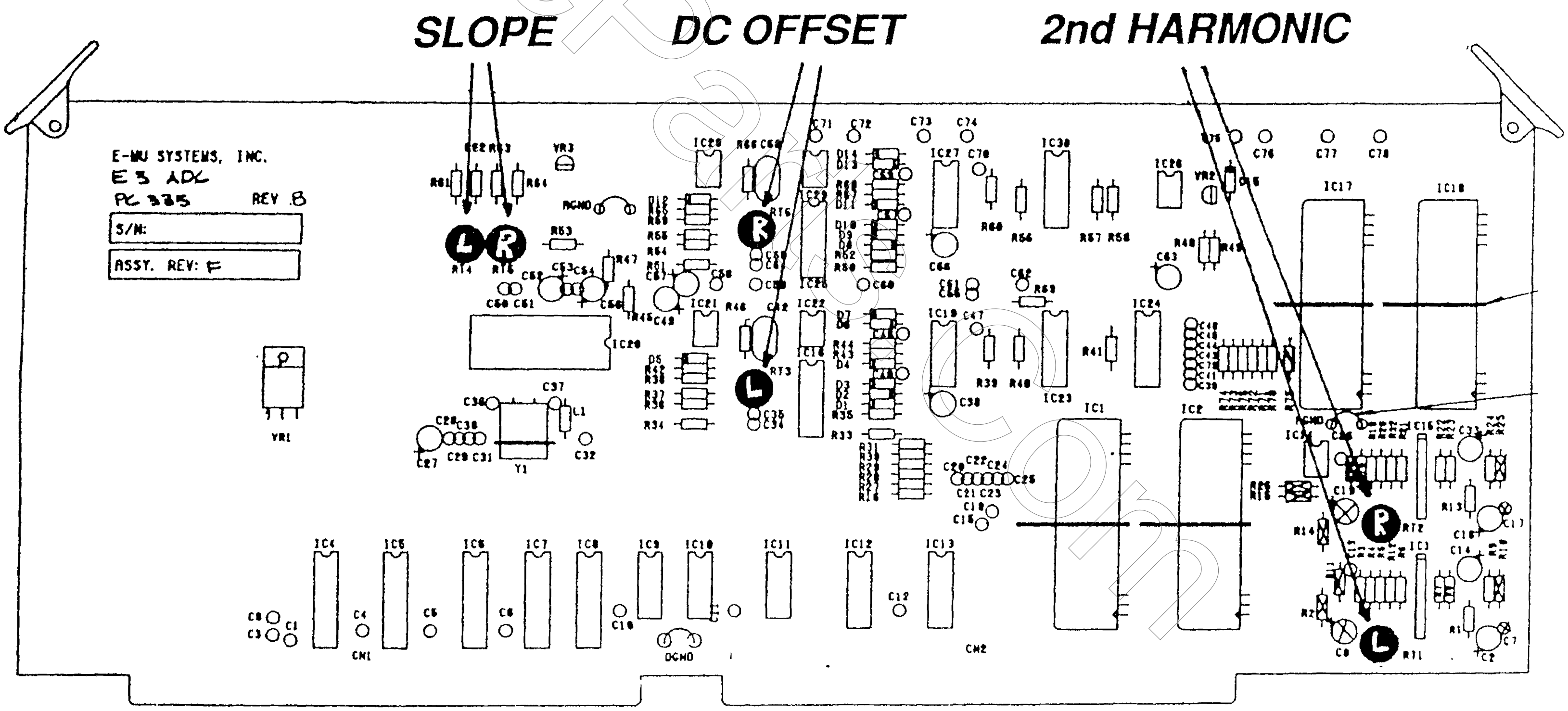
PROBLEM	CAUSE	SOLUTION
Computer		
No lights, no power.	Power supply bad or bad power connector crimp.	Replace supply or fix connector.
No Boot	HD problem.	Try floppy software boot.
HD doesn't power up.	Bad connection at power supply connector.	Clean power connector.
Panel lights flash on/off.	110/220 selector at 220.	Set to correct voltage.
Won't load floppy disks.	Drive out of alignment or bad disk.	Try different disks or swap drive.
Loads software then display goes blank.	Bad Scanner MPU.	Replace bad chip.
Intermittent lock-up, usually during HD access.	Bi-metallic contamination on main processor pins.	Clean µprocessor pins.
Sample LED,Enter LED, and Enter button don't work.	Broken ribbon cable wires on CPU side of cable.	Repair or replace cable.
Digital Distortion		
Bad distortion on 1 channel	Bad F-Chip, S/H or DAC.	Replace bad chip.
Single channel plays wrong pitch	Bad F-Chip.	Replace F-Chip.
Bad distortion, crackling	Memory problem, loose SIMM.	Perform memory check, clean and re-seat SIMM.
No output, single channel.	Possible DAC problem.	Check pin 24 on DAC 6.0->6.5 = OK Low = Bad DAC

PROBLEM	CAUSE	SOLUTION
Microcontroller		
Parts of multiple samples play.	Microcontroller problem.	Locate and correct fault.
Random DSP type effects.	Microcontroller problem.	Locate and correct fault.
Clank, Clunk, Clink sounds.	Microcontroller problem.	Locate and correct fault.
Wrong sounds play.	Microcontroller problem.	Locate and correct fault.
Analog		
Clicks during loops on some channels.	Bad SSM 2300 S/H	Swap to determine bad chip and replace.
Channel distortion or pops.	Bad Curtis chip.	Replace bad chip.
Pop on output with low Fc.	Bad Curtis chip.	Replace bad chip.
Buzzing sample, drifting Fc, no Q.	Bad 15pF Cap. on Curtis chip.	Replace bad cap.
Whistling or hissing samples at 42 kHz sample rate.	Bad SSM 2300 S/H in Gain location.	Replace bad chip.
No Fc, Gain, Pan, Q control	Bad SSM 2300 S/H.	Replace bad chip.
Warbling Fc	Bad SSM 2300 S/H.	Replace bad chip.
Sound plays during disk loading.	Bad SSM2300: VCA location.	Replace bad chip.
Amplitude variations between channels	Levels not calibrated or Curtis Chip bad.	Check level calibrations or replace Curtis Chip.
Can't trim ADC, sampling noise.	Bad C42 or C68 on ADC.	Replace bad Cap.
Keyboard		
Single key doesn't work	Dirty contact.	Carefully clean contact.
Every 8th key doesn't work.	Bad diode in KYBD matrix.	Replace diode.
Pressure doesn't work.	Calibration or bad connection.	Check cal. and wiring.
Hard Disk		
EIII reads "Disk Not Formatted".	The HD may have "crashed".	Try reformatting the HD.

PROBLEM	CAUSE	SOLUTION
Hard Disk		
EIII reads "Disk Not Formatted".	The HD may have "crashed".	Try reformatting the HD.
EIII reads "SCSI Error" when there. is an external drive connected.	Two SCSI devices have the same ID # or the external device doesn't have power.	Check Power. Check SCSI ID #s. Try different device.
Other		
Slider doesn't work	Broken solder joint at pot.	Resolder or replace pot.
Slider or wheels don't work	EIII needs recalibration	Recalibrate wheels and sliders.
Squealing power supply.	Bad decoupling Capacitor.	Insert one board at a time to determine board, then find and replace bad Cap.
Intermittent power.	Bad connection at power supply.	Clean contacts and reseat DC power connector.
Operator Error		
Certain sounds do not play polyphonically.	The zone may be in solo mode, or only assigned to one output channel.	Disable solo mode and check channel assignment.
Stereo samples heard out of only one side.	Disable side fountain is turned on.	Turn disable side function off (Analog Processing, 1).
Looping only works while in Digital Processing module.	Disable Loop function is turned on.	Turn off Disable Loop function (Analog Proc. 1).
EIII drift out of tune or the pitch wavers.	Controls need to be recalibrated.	Recalibrate controls (Master, Special, 3).
Footpedal or LFO has little or no effect when assigned to the VCA.	The initial VCA level is set to 100%.	Lower the initial VCA setting as desired.
Channels "rip off" or don't sound when assigning stereo samples to specific output channels.	Incorrect output channel assignment.	Assign output channels in pairs, 1-2, 3-4, 5-6, etc.
EIII doesn't respond to MIDI as programmed in the MIDI submodule.	MIDI Globals are turned on.	Turn off MIDI Globals (Master, Special, 7).

ADC BOARD TRIM

POT LOCATIONS



1. Introduction

This document describes National Semiconductor's monitor, *dbmon. Dbmon* works on a *Series 32000*™ microprocessor based target system, such as a *Series 32000* Development Board. It is tailored for use in remote debugging of a *Series 32000* microprocessor with *ddt*(1) running on a host system and is included as part of the GENIX™ Cross-Support package for A VAX™ running UNIX™, or with the GENIX operating system.

In addition to running with *ddt*, *dbmon* can also be used directly either from a terminal connected to the target system or from a terminal connected to the host, with the program *cu32*(1) managing the serial line between the host and the target system. Because the monitor commands are tailored for use with *ddt*, some of the commands may be inconvenient when entered directly by the user. Except for loading and executing programs, *dbmon* does not provide operating system functions. (See also: *GENIX Cross-Support Software for a Series 32000-Based System.*)

A typical application of the monitor would be in a *Series 32000* Development Board connected to a VAX by a serial line. Information specific to this application is included in this document.

Dbmon replaces the TDS PROMs supplied with the *Series 32000* Development Board.

2. Notation Conventions

Certain characters are specified symbolically in this document:

$	= escape
<CR>	= return
<lf>	= line feed (ctrl/j)
<bs>	= backspace

3. Running Mode

The monitor runs in supervisor mode. By default, a user program loaded with *dbmon* starts in supervisor mode, though with the stack pointer set to SP1 (user stack pointer). However, the program may be started in user mode by first changing the setting of the PSR with the command "cpsr=0b00".

4. Command Input Protocol

The monitor reads lines up to a <CR>. Commands are not executed until the <CR> are typed. <lf> characters are ignored.

The backspace character may be used to delete previously typed input. The escape key will abort commands at any time. It will also discard any input and terminate a ctrl/q wait.

All numeric quantities are entered in hex. Spaces may be inserted before any number.

Commands and hex numbers may be typed in upper or lower case.

The monitor's output may be suspended with ctrl/s and resumed with ctrl/q.

GENIX and *Series 32000* are trademarks of National Semiconductor Corporation.

GENIX was derived from 4.1bsd which was developed at the University of California at Berkeley.

GENIX 4.2 was derived from 4.2bsd which was developed at the University of California at Berkeley.

VAX is a trademark of Digital Equipment Corporation

UNIX is a trademark of AT&T Bell Laboratories.

The information in this document is for reference only and is subject to change without notice.

5. Memory Organization

Dbmon requires no RAM except for space on the interrupt stack. The interrupt stack begins at 64k.

A program which moves the interrupt stack may still reenter *dbmon*; it does not matter where the stack is in RAM when the monitor is entered.

INITIAL MEMORY ORGANIZATION			
SYSTEM	**AREA**	**LOW**	**HIGH**
Series 32000 Development Board	Interrupt Stack	~ 63.8k	64k
Series 32000 Development Board	User Stack	N.A.	60k

The monitor runs in supervisor mode and always uses the interrupt stack. A program can run in either mode with either stack. There are two reasonable stack configurations for the monitor and program:

(1) Program uses SP1 and separate memory locations are reserved for each stack.

(2) Program uses SP0 (program and monitor run on the same stack).

In the first case, the program must avoid the memory used by the interrupt stack, because those locations will be overwritten when the monitor is entered.

In the second case, the program can allocate memory anywhere; however, if the program fails because its stack is not in RAM, the monitor will also fail.

The first case is *dbmon's* initial organization. Specify "cps=0" to change to case two.

6. *Dbmon* Commands

In the following command descriptions, lower-case letters represent literal character input and upper-case letters are used to indicate parameters, while either case is actually allowed in the command.

The commands groups are as follows:

- Memory and register print commands
- Memory and register change commands
- Read and write MMU register commands
- Set configuration register command
- Group data commands, *e.g.*, move, fill and load blocks of memory
- Program execution control commands
- Additional commands
- Test commands
- Reserved commands

NOTE: The print and change virtual memory commands require an NS32082 MMU chip.

6.1. Print Commands

The contents of memory and the registers may be printed with the following commands:

all	- Print all the registers.
ppsr	- Print the processor status register.
psb	- Print the static base register.
pis	- Print the interrupt stack pointer (SP0).
pmod	- Print the mod register.
pfp	- Print the frame pointer.
pus	- Print the user stack pointer (SP1).
psp	- Print the user stack pointer (SP1). NOTE: **pus** and **psp** are functionally identical.
pintbase	- Print the interrupt base register.
prN	- Print general register N.
ppc	- Print the program counter.
pmbA	- Print one byte of memory at address A.
pmwA	- Print one word of memory at address A.
pmdA	- Print one double-word of memory at address A.
pvbA	- Print one byte of memory at virtual address A.
pvwA	- Print one word of memory at virtual address A.
pvdA	- Print one double-word of memory at virtual address A.

Note that the **p** commands are constructed in the following way: the letter "p" followed by a register name, *e.g.*, **fp** or **r1** or **m** for physical memory, or **v** for virtual memory. The "m" or "v" is then followed by a "b", "w", or "d" for byte, word or double-word and then the address of interest.

All register names may be abbreviated to two characters, *e.g.*, "in" for **intbase.**

6.2. Change Commands

Memory and registers may be changed with the following commands:

cpsr=V	- Change the processor status register to value V.
cmod=V	- Change the mod register to value V.
cfp=V	- Change the frame pointer to value V.
cus=V	- Change the user stack pointer (SP1) to value V.
csp=V	- Change the user stack pointer (SP1) to value V. NOTE: **cus** and **csp** are functionally identical.
cintbase=V	- Change the interrupt base register to value V.
crN=V	- Change general register N to value V.
cpc=V	- Change the program counter to value V.
cmbA=V	- Change one byte of memory at address A to value V.
cmwA=V	- Change one word of memory at address A to value V.
cmdA=V	- Change one double-word of memory at address A to value V.
cvbA=V	- Change one byte of memory at virtual address A to value V.
cvwA=V	- Change one word of memory at virtual address A to value V.
cvdA=V	- Change one double-word of memory at virtual address A to value V.

Note that the **c** commands are constructed in the same manner as the **p** commands.

All register names may be abbreviated to two characters, *e.g.*, "mo" for **mod.**

There is no change register command for the interrupt stack or static base registers. The interrupt stack pointer can be changed with a supervisor mode program; the static base register can be changed by altering the module entry selected by the mod register.

6.3. MMU Commands

The MMU registers are read from, and written to, with:

wN=V - Write value V to MMU register N.

rN - Read (and print) MMU register N
(the registers and the corresponding value of N:
bpr0=0 bpr1=1 pf0=4 pf1=5 sc=8 msr=a bcnt=b
ptb0=c ptb1=d eia=f)

6.4. Configuration Register Command

The configuration can be changed with the **x** command:

xV - Execute a setcfg instruction for configuration V
(icu=1 fpu=2 mmu=4)

6.5. Data Commands

m A1 A2 N - Move N bytes from A1 to A2.

m A1 A2 N [bwd] - Move N bytes, words, or double-words from A1 to A2.

f A1 A2 DD - Fill memory from A1 to A2 with data byte DD.

f A1 A2 V [bwd] - Fill memory from A1 to A2 with byte, word, or double-word value V.

d A1 N - Dump (print) **N** bytes of memory starting at **A1**. The format for the output of this command is the same as the format for input to the load command.

NOTE: Although the data is printed in 8-digit groups, these groups are not double-words, as the order of the bytes is from the least significant to the most significant which is the reverse of the order for double-words.

l A DDDDDDCC - Load bytes DD... starting at address A. Bytes are represented by pairs of hex digits, up to 16 data bytes (32 hex characters) per line. The line is terminated by a checksum byte, which is the 2-digit hex representation of the 8-bit sum of the data bytes on the line. If the transmitted checksum does not match the computed checksum, the message **E CRC** is printed with the value of the computed checksum.

image - Start the binary image loader. The image loader reads a header, data bytes, and then a 1-byte checksum from the serial line. The header consists of a 32-bit starting address and a 32-bit length value. These parameters are sent as 8-bit bytes, least significant byte first. The length parameter specifies the number of data bytes, to be transferred as 8-bit binary bytes. The checksum is also transferred as an 8-bit byte. <Esc>, <bs>, and ctrl/s have no special meaning in this mode.

If "i" is typed from the keyboard, the *Series 32000* Development Board will probably need to be reset. Entering ten null bytes for an address, length, and checksum of zero may prevent the need for resetting the board, but only if the terminal uses even parity.

6.6. Program Execution Commands

s - Single-step: execute the instruction at the current pc.

g - Start the program (with the current pc).

q - Do a nonsequential fetch step, using NS32082 MMU.

6.7. Additional Commands

v A1 L - Compute and print the 32-bit checksum of the **L** bytes beginning at address **A1**

! - Ignore the remainder of a line. Useful for comments when *dbmon* commands are stored in a file.

6.8. Test Commands

t1 - Run test 1. This test probes the RAM configuration at every 4k interval, up to 2^{20} bytes. The test reports the first and last address of available memory and possibly the last address checked.

t2 A N - Run test 2 starting at address A for N bytes. This test runs the memory diagnostics described in Section 8.

7. NS32082 MMU Dependencies

There are two versions of *dbmon* supplied in PROM. The first version, dbmon1, is for use in systems with an NS32082 MMU; the second version, dbmon2, is for use in systems without an MMU.

The monitor program source file, *rom.S*, contains the source for the version which uses the MMU. The *rom.S* file can be modified to produce a custom monitor. If the custom monitor is not to be dependent on an MMU, a search for the name **NOMMU** in the source file will find comments indicating lines which need to be deleted for non-MMU systems. Do not use the commands **pv[b,w,d]**, **cv[b,v,d]**, **q**, **r#**, and **w#** when running a monitor built without those source lines. These commands use the MMU slave instructions and will cause an undefined trap in a program modified for no MMU.

8. Memory Testing

The **t2** command starts test two, the memory test sequence. Eight subtests are run in sequence; the sequence is restarted after the last subtest. The tests destroy the original contents of the addresses checked. The memory test itself uses no RAM; however, if a failure is found, RAM locations on the stack are used temporarily by the formatted output routines. If an error occurs and the stack is within the memory range being checked, spurious errors may be reported in addition to any actual errors. The first error reported will always be a real error.

8.1. Test Patterns

Each of the subtests cycles through seven patterns (see Table 1).

TABLE 1. MEMORY TEST PATTERNS IN HEXADECIMAL

NO.	PATTERN	FUNCTION
1.	00000000	all zeros
2.	ffffffff	all ones
3.	aaaaaaaa	adjacent bits different
4.	55555555	adjacent bits different
5.	11224488	a different high in each byte
6.	eeddbb77	a different low in each byte
7.	39a7c736	miscellaneous bits

8.2. Test Sequence

Each subtest prints its name as it starts (see Table 2).

TABLE 2. MEMORY TESTS

NO.	NAME	CONTEXT	TEST
1.	parity	words	parity(address) xor pattern
2.	addr ->	double-words	address xor pattern, low to high
3.	addr <-	double-words	address xor pattern, high to low
4.	doubles	double-words	pattern
5.	e. words	even words	pattern
6.	e. bytes	even bytes	pattern
7.	load	doubles	high transfer rate

8.3. Test Failures

When a test fails, a message is printed with the following format:

Failure(=addr=xor=test=mem)=AAAA=XXXX=TTTT=MMMM

AAAA
- address containing incorrect data.

XXXX
- pattern used to provide the current test variation.

TTTT
- expected contents of memory.

MMMM
- actual contents of memory.

8.4. Memory Test 1: Address Parity

This test computes even parity for each word address. The parity bit is then extended to form 16 bits of zeros or ones; and the exclusive-or of this value and the current pattern is written to the word.

The words are checked on a second pass.

8.5. Memory Test 2: Increasing Addresses

For each double-word address, this test writes the exclusive-or of the pattern and the address to the double-word. The addresses are written to in increasing order from the base address. They are checked on a second pass.

8.6. Memory Test 3: Decreasing Addresses

This test is similar to Memory Test 2, but the addresses are written to in decreasing order of address, starting from the base address. The addresses are checked, in increasing order, on a second pass.

8.7. Memory Test 4: Double Words

Test 4 writes each pattern into each double-word.

8.8. Memory Test 5: Even Words

The lower word of each pattern is written to each even word.

8.9. Memory Test 6: Even Bytes

The lower byte of each pattern is written to each even byte.

8.10. Memory Test 7: Load

A sequence of double-word string moves are generated in order to test the memory system with a constant high transfer rate. This test uses memory parity hardware to detect errors. If the target board does not have memory parity hardware, the test is not useful.

THEORY OF OPERATION

OVERVIEW

The Emulator Three is the latest heir to the Emulator lineage that began with the Emulator I. Unlike either the Emulator I or Emulator II, the Emulator Three does not use any form of data compression and takes full advantage of current technology to deliver true 16-bit stereo sampling. The EIII also utilizes an internal 40 MByte hard disk, holds up to 8 MBytes of RAM and has 16 output channels. The design concept of the microcontroller is very similar to the Emulator II and the computer sections are very similar to the design of the Emax. Those of you familiar with the EII and Emax will have a definite advantage in servicing the EIII. Like the EII, most of the logic is implemented in TTL (74LS and 74HCT series) except for a few custom chips such as the F-Chip (a custom timer, oversampling chip), the CMOS memory and a few PALs (programmable logic arrays).

The simplified block diagram shows the entire EIII. Starting at the top of the page we can see that the keyboard, footswitches and front panel buttons are read by the scanner CPU. The scanner CPU interrupts the main CPU when it has user data to transfer.

The LEDs are treated as a write-only port. They are connected to a data latch that is written to by the main CPU when an LED is to be turned on or off.

The liquid crystal display is connected to the data bus and treated as a read/write port. All user information is written to this port in ASCII except for a few graphic symbols.

The National 32016 serves as the main CPU. It is coupled with a 32201 timing control unit (TCU) which generates clocks, reset, read/write signals and allows bus timing control. There is also a 32081 floating point co-processor coupled to the main CPU. The main operating system is loaded into four 4464 dynamic RAMs on boot up from the hard disk. Since the EIIIs operating system is larger than 128K Bytes, an "overlay" system is used. As different processing modules on the front panel of the EIII are accessed, different sections of the program are loaded into RAM from the hard disk. This accounts for the hard disk activity as different modules are entered. Fortunately, the hard disk access time is so short that these delays are almost imperceptible. Two 27128 EPROMs hold the bootstrap routines as well as various lookup tables. The main CPU generates the digital envelopes that are sent to the output channels and control the analog filters and amplifiers.

The floppy disk interface is handled by a Western Digital 1772 single chip floppy controller which interfaces to the main CPU via the data bus.

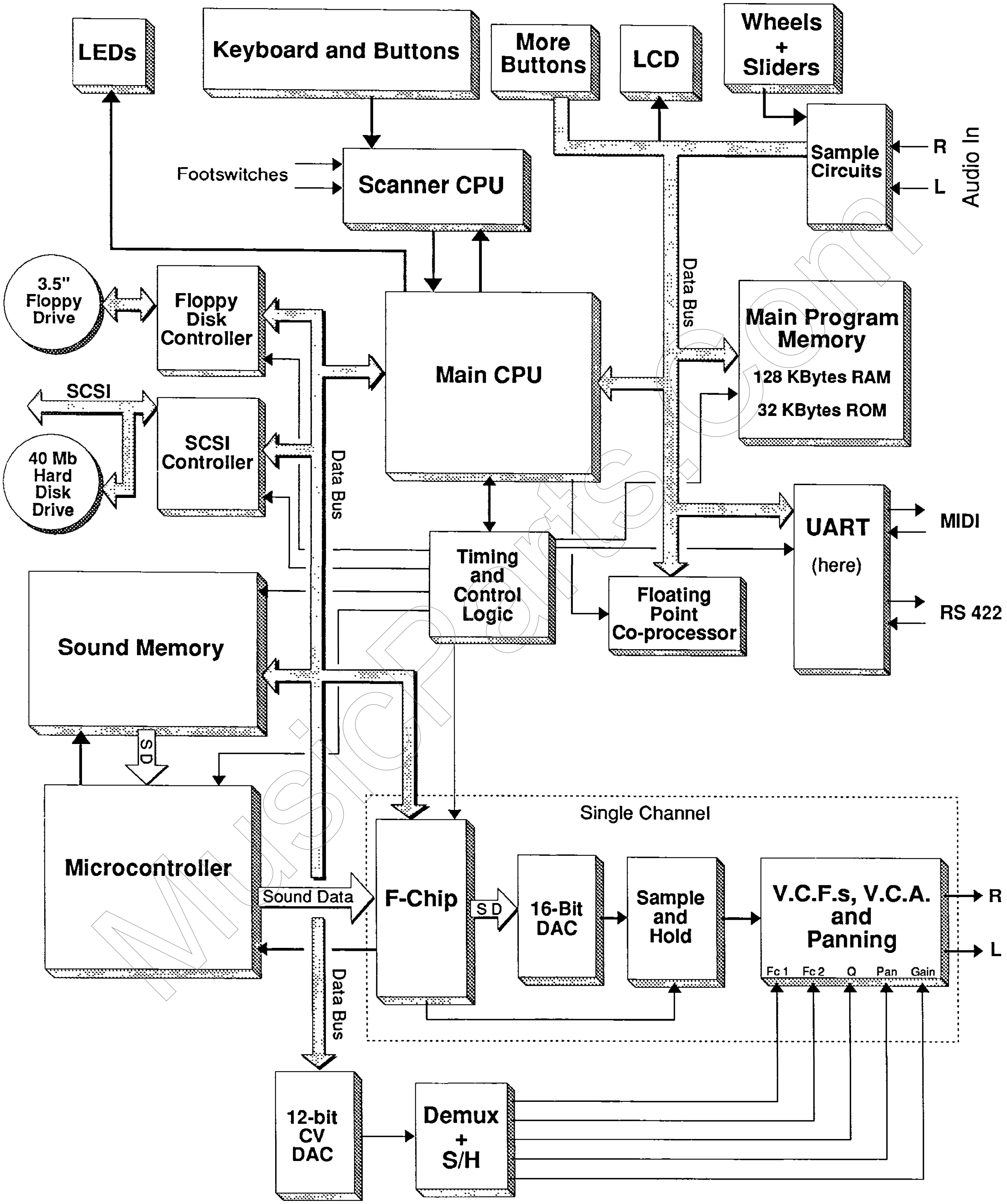
The SCSI interface is handled by a 5380 SCSI interface chip which interfaces to the main CPU via the data bus.

MIDI and the RS422 serial computer interfaces are performed by an 82530 serial communications controller which is also attached to the main data bus.

Looking now at the lower left hand corner, we can see that the microcontroller controls the flow of sound data to the output channels. The microcontroller is a special purpose processor very similar to the EII microcontroller. It has its own program which resides in various PALs which are scattered around the microcontroller, and has registers, adders and clocks. The microcontroller directs all the output sound addressing for each of the 16 channels and thus frees the main CPU from this formidable task.

The dotted box in the lower right hand corner of the simplified block diagram shows the sound generation circuitry that is duplicated for each of the 16 output channels. Sound data is passed from

EIII SIMPLIFIED BLOCK DIAGRAM



the microcontroller to the F-chip. The F-chip is a custom designed chip that controls the channel timing or frequency, and performs a 2x oversampling function which doubles the sample rate. The F-chip also contains circuitry to request data from the microcontroller when needed and to control the operation of the sample and hold. The data is passed from the F-chip to the 16-bit DAC where it is converted back into audio information. The DAC is followed by a sample and hold which is controlled by the F-chip. The signal is now passed through a Curtis 3387 analog signal processor chip which contains a three-pole elliptical lowpass filter (for signal reconstruction), a four-pole lowpass filter (for synthesizer effects), a voltage controlled amplifier (for amplitude contouring, and a stereo panning network. The control voltages for the Curtis chips originate at the main CPU and are supplied through a 12-bit DAC (on each output board) and distributed by SSM 2300s, which are 8 channel sample and holds.

NOTICE

The EIII was designed to use 74HCT series logic instead of 74LS. We did this to lower the power requirements and use a smaller power supply. We do NOT recommend using 74LS chips as replacements for HCT as they overtax the power supply. There are a few locations where LS is used because of circuit loading, however the majority of the TTL ICs are HCT.

CPU AND CLOCKS

Now refer to the CPU board block diagram. The main processor is a 10MHz 32016 CPU, driven by a 32201 Timing Control Unit. The CPU produces a 24-bit address, latching the low order 16 bits of the address from the multiplexed address/data bus with two 74HCT373's. The data bus is buffered to the other circuit boards by two 74HCT245's. The CPU obtains INT and -CWAIT from the interrupt/wait control PAL, SPC from the floating point unit, WAIT1, WAIT2, and PR from the chip select PAL. The TCU acts as a crystal oscillator producing 10 and 20MHz clocks and additional CPU signals: -RD, -WR, and TSO (an additional timing signal). It also generates a clean -RESET signal for the rest of the system. There is an additional 16MHz crystal oscillator which supplies 1, 2, 4, and 8MHz clocks for other devices within the system such as the scanner CPU.

The "wait state" timing of the 32016 is determined by the 32201, which in turn is controlled by the address bus of the 32016 via a PAL (IC27). Hence, the port address of memory and I/O will declare their timing through the states of -WAIT1 and -WAIT2. Wait states can be invoked by the ADC (-ADCWAIT), the microcontroller (-RFWAIT, -SDWAIT) or the main processor DRAM (-UPRWAIT) which communicate with the main processor via the interrupt/wait PAL.

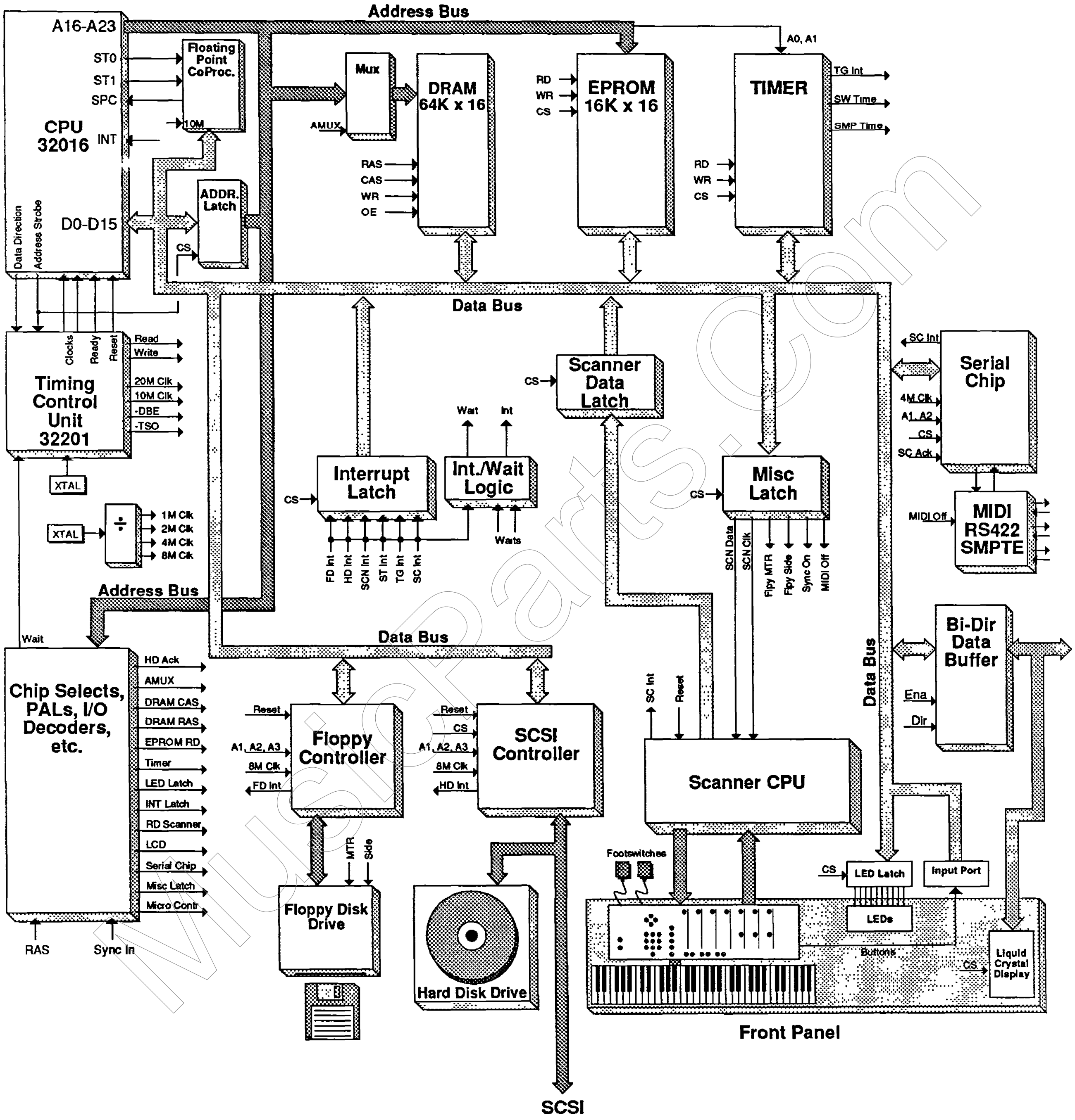
PROCESSOR DRAM

The processor DRAM consists of 64K x 16 bits of DRAM, controlled by a PAL (IC11) with an address mux and an additional 74S74 flip-flop. The PAL selects the start of a cycle from the address strobe (ADS), and initiates a RAS-AMUX-CAS cycle as a consequence. If the 74HCT393 refresh counter has reached a terminal count (which occurs every 8 µsec), AMUX and CAS will be extended, and a CAS-before-RAS cycle will occur. If the processor requires DRAM before this cycle is complete, it will have to wait. The WAIT line from the PAL will always go low during refresh; however, the INT/WAIT PAL will only cause a CPU CWAIT when a new bus cycle is begun.

BOOT EPROM, 8254 TIMER

The boot EPROMS are two 27128's. These supply 32K bytes, configured as 16K by 16 of boot area, ending just below the start of DRAM.

EIII CPU BOARD



The 8254 timer produces three signals. The Transient Generator (TG interrupt) is a periodic interrupt produced from a divisor on the 10MHz clock, latched from the rising edge of TGTIME, and reset by the RSTTGI chip select. The TG interrupt is so named because it signals the CPU that it is time for a TG update cycle. The SWTIME signal (divided from the 1MHz clock), the SMPTIME signal (divided from the SMPTE phase locked loop clock), and the SYNC signal (from the SMPTE/SYNC input comparator) all proceed to the serial interface PAL for selection into the STINT (software timer interrupt) signal.

CHIP SELECTS AND INTERRUPTS

The chip selects are produced by a PAL and three 74HCT 138's. The PAL decodes the addresses and the bus control signals to produce enables for the 138's. The PAL generates the WAIT1 and WAIT2 signals as well as the buffer enable signals. As mentioned earlier, the WAIT1 and WAIT2 signals are address dependent, hence the programmer can generally adjust device timing and which side of the buffer is being used.

Three 138's are used for the three different types of devices. One supplies "address decoded" chip selects for devices which sense RD and WR directly. The second provides chip selects pre-ANDed with WR for latches and other write-only devices. The third provides chip selects ANDed with (RD+WR) for devices that require a set-up of the read and write lines (which are supplied as address lines), and for read only devices.

The interrupt vector is supplied to the CPU by directly reading the various interrupt request lines. The programmer will assemble a prioritized vector table for all possibilities. The interrupt/wait PAL combines the various sources of WAIT and INT signals to produce CWAIT and INT. This PAL also contains the logic associated with 5380 SCSI controller handshake.

FLOPPY DISK AND SCSI CONTROLLERS

The floppy disk controller is based on the Western Digital 1772 single chip floppy controller. This chip supplies most of the floppy disk control signals and directly interfaces to the main CPU. The input signals are terminated by R-pack networks, and the output signals are driven by inverting drivers. Two additional signals, -MTR and -SIDE, are supplied by latch IC26 because they are not correctly supplied by the 1772.

The SCSI interface is the 5380 SCSI interface chip. The random logic associated with the "pseudo-DMA" interface is contained in the INT/WAIT PAL.

SCANNER MPU

The scanner processor is the Rockwell 6500/11, which is the same scanner used in the Emax. This is a single chip microcomputer containing a 6502 microprocessor, 3K of ROM, 192 bytes of RAM, 2 counter/timers and built-in I/O ports and clock.

The scanner MPU handles the velocity-sensing keyboard and most of the front panel pushbutton interfacing with associated 4028 decoders and HCT244 bus driver. The keyboard and buttons are arranged in a standard diode matrix. The EIII contains more buttons than the scanner was designed to handle, therefore eight of the front panel buttons must be read directly by the main CPU.

The communication between the 6500 and the 32016 microprocessors is handled in the following manner: When the 6500 has data for the 32016, it requests a SCNINT interrupt. When the 32016 honors this interrupt by reading the CSSCAN port with SCNDATA and SCNCLK bits low, the 6500 receives an NMI, allowing it to pass additional data if required. When the 32016 wishes to send data to the 6500, it issues a CSSCAN with either (or both, depending on data to be transferred) of the SCNDATA and SCNCLK lines high. The 6500 reacts to this NMI by either performing the requested operation, or entering a serial transfer mode whereby the 32016 sends data using the SCNDATA or SCNCLK lines to the 6500.

SERIAL INTERFACES AND MISCELLANEOUS

The serial interfaces for the EIII are performed by an 82530 SCC, with associated analog interfaces and the serial interface PAL. MIDI is received and sent by familiar interface electronics. Similarly, the RS422 drivers and receivers are standard and familiar. The PAL selects between MIDI and RS422 operation depending on the state of the MIDIOF bit. MIDI OFF causes the serial RXD to come from the RS422 receiver and the serial TXD to go to the RS422 driver while MIDI remains inactive. MIDI ON deactivates the RS422 drivers and instead gets RXD from the MIDI iso-optolator and sends TXD to the MIDI driver.

The LM311 comparator provides a 100mV threshold interface for SMPTE and SYNC inputs.

The serial PAL also combines the SWTIME and SWPTIME signals (which are mutually exclusive), and selects between this and the SYNC signal from the SMPTE comparator depending on the state of SYNCON. This interrupt is then latched, and supplied to the INT/WAIT PAL. The interrupt is cleared by the RSTSTI chip select signal.

Three 74HCT273 data latches hold the miscellaneous bits for the system. Two of the latches and one bit of the third hold LED data for the front panel. Additional bits include:

SYNCON	-Determines if STINT is from SYNC or SWTIME/SMPTIME
MTR	-Floppy Disk Motor
SIDE	-Floppy Disk Side Select
SCNCLK	-Serial Clock to Scanner Processor
SCNDATA	-Serial Data to Scanner Processor
MIDIOF	-Determines if serial interface is MIDI or RS422
POTSON	-Informs ADC that front panel pots are to be converted

A 74HCT221 one-shot provides a fixed width pulse for the metronome/clock out.

Eight additional front panel pushbuttons are handled by a 74HCT244. These buttons are scanned directly by the 32016 on a polled basis.

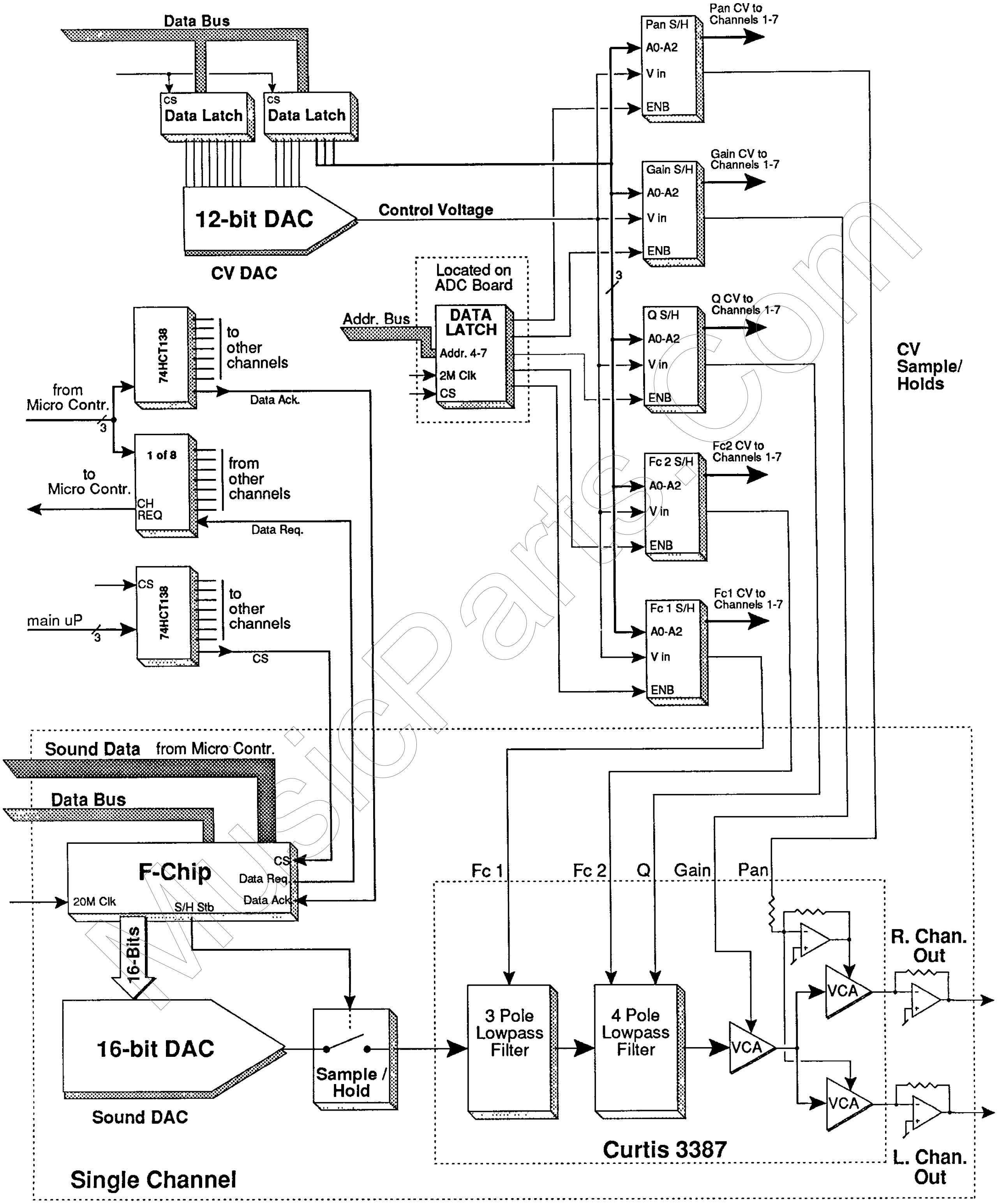
OUTPUT BOARDS

The EIII contains two identical output boards. Each output board consists of eight identical channels and common control circuitry. The common logic includes: a 74HCT138 which decodes the FCHIP chip select signals from the 32016 address bus and the common CSFCHIP signal, a 74HCT151 which multiplexes the eight channel request signals into a common CHREQ signal for the microcontroller, and a 74HCT138 which decodes the the ACK signal from the microcontroller and accesses the sound memory in the cycle following the request cycle. The ACK signal is decoded from the CHSL value one higher than the channel number.

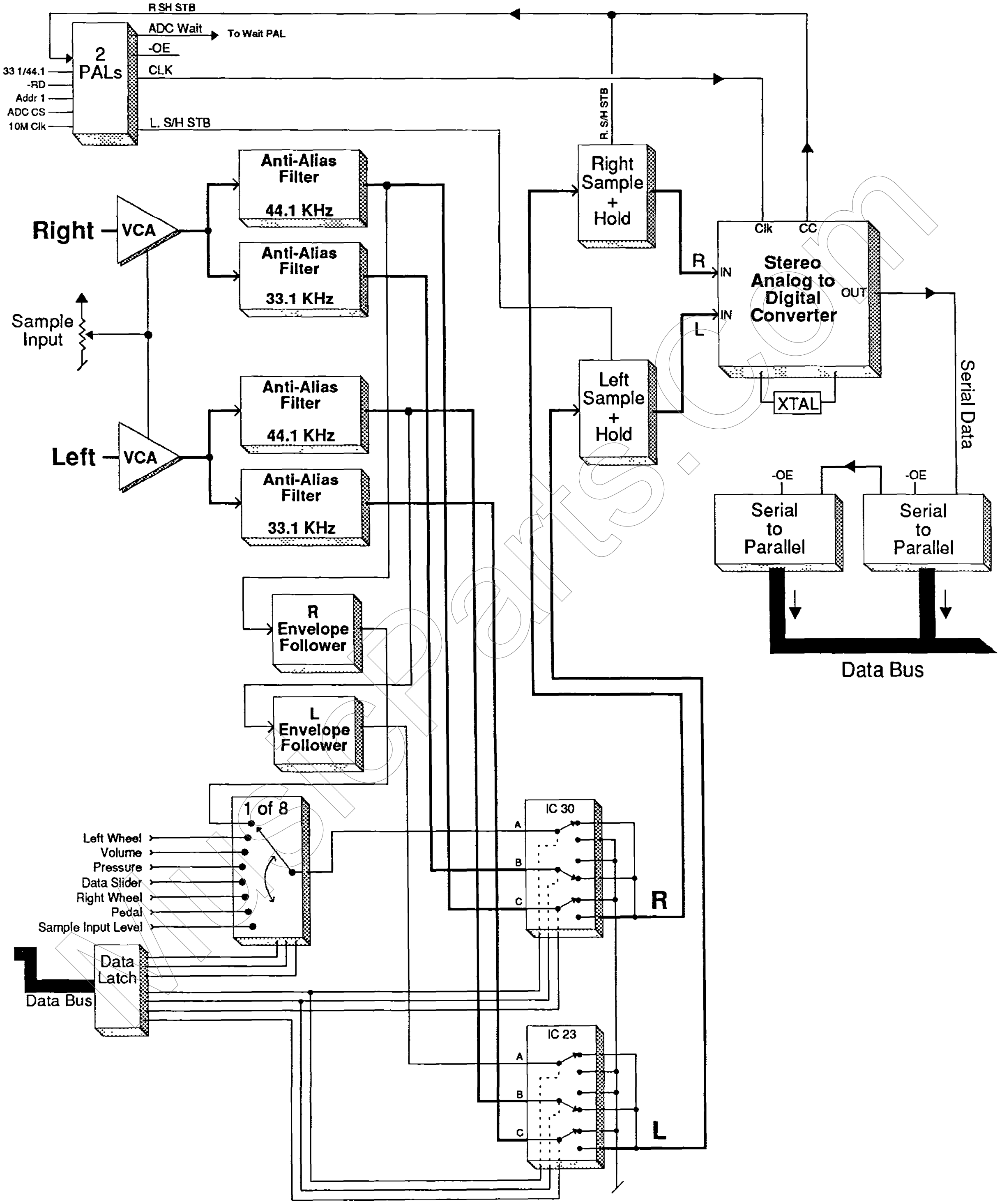
Additional common circuitry includes the 74HCT273 latches and the 12-bit DAC for the control voltages. Note that the DAC uses monopolar data in bits 13-2 of the data bus. The channels are coded as consecutive words in the address space. The control voltages range from 0 to +5 volts.

The final common circuitry is five SSM2300 control voltage multiplexer/sample-hold IC's. Each of these chips outputs one of the control parameters to each of the eight channels. The control parameters are Extra 3-pole filter Fc, 4-pole filter Fc, Q, Amplitude and Pan. The amplitude CV is processed to provide a fast attack, but slow decay, and all other parameters except for the Q are filtered.

EIII OUTPUT BOARD



EIII ADC BOARD



Each channel consists of an F-Chip, which handles the channel timing, data requesting, data latching, and performs a 2:1 upsample and digital FIR filtering of the signal. The resulting 16-bit digital signal is converted to analog with a 16-bit DAC and sampled by a discrete sample/hold circuit built from a 74HCT4053 and a LF356 opamp.
The resulting analog signal is dynamically filtered, amplified and panned by the CEM3387. The left and right current outputs are converted into voltages by two halves of a 5532 dual opamp.

ADC BOARD

The left and right input signals are treated identically. The signal is first applied to a low distortion dbx 2155 VCA for amplitude scaling, then converted back to a voltage by a LF353 opamp. The resulting signal is applied to both a 20KHz and a 14.2KHz hybrid anti-aliasing filter, producing two band-limited signals.

The 44KHz signals are envelope followed to produce signals appropriate for VU metering. These signals, along with the wheel, footpedal, and slider voltages, are multiplexed into the left channel when POTSON is active. The left and right channel anti-alias filter outputs are selected by the signal KHZ44.

After being selected, the signal is sampled by a discrete sample/hold circuit built from a 74HCT4053 and a LF356 opamp. The ADC is next and is accomplished by a Sony 20018 stereo ADC chip. The ADC has a serial output and so the serial data is next sent to a pair of 74HCT299 serial to parallel converters.

The overall control of the ADC process is performed by two PAL's which direct two basic functions.

First, they decode the CPU control signals (RD, A1, CSADC) into the left and right output enable signals for even and odd word ADC reads, respectively, and into the clock for a 74HCT174 which holds the the slider multiplexer addresses (MXA), the pot conversion enable (POTSON), and the sample rate selection (KHZ44) signals.

Second, the PAL's divide down the 10MHz clock to produce the BCLK, and the left and right SHSTB. These signals are devised to vary the conversion rate depending on the significance of the bit under conversion and the sample rate selection. The ADC Wait signal to the CPU is also generated here.

The ADC card also contains some common circuitry for the output function. This includes an inverter to produce -CHSL3, allowing a upper and lower output board. Also included is the circuitry to decode the address of the CVDAC port into an enable for the appropriate SSM2300 sample/hold. The 74LS293 counter is reset to on the write to the CVDAC, and then waits 2 µsec before enabling the selected 2300. This enable lasts 2 µsec, then the circuit reaches a terminal count and awaits the next CSCVDAC. The 2300's are enabled such that there are 16 words per parameter (one for each channel), for parameters (in increasing order above base address) Extra filter, Filter, Q, Amplitude, and Pan.

MICROCONTROLLER BOARD

The microcontroller is in control of the EIII's sound RAM. It is a further extension of the microcontroller designs found on other E-mu products from the Drumulator to the EII. The microcontroller can be thought of as a separate computer running at a higher speed to which other devices must request access. For example, when an output channel needs data, it requests it from the microcontroller. The microcontroller is running so fast that the wait time is negligible. Refer to the block and schematic diagrams while reading the theory of operation and signal descriptions. The EIII µController is an extremely complicated piece of hardware. To thoroughly understand its operation will take serious study.

The microcontroller is clocked by a 74LS161 based 6-bit counter. Each channel is serviced by four counts (UCNT0 and UCNT1), and there are 16 channels (CHLS0-CHLS3).
The two sound PALs take in a variety of flags and signals to produce the pipeline control signals. The signals are:

INPUTS:
UCNT0 - Defines the current microcode step (0-3)
UCNT1 - Defines the current microcode step (0-3)
FORBKFLG - Valid on count 0. Set during a "forward-backward" loop.
INITFLG - Valid on count 3 and 0. Set for first sample of a loop.
DECFLG - Valid on count 0. Set when address is decreasing (bkwds).
LOOPFLG.1 - Valid on count 1. Set when sound is looping.
CARRY - Valid on count 1 and 2. Indicates adder carry.
CHREQ - Valid throughout cycle. Indicates FCHIP needs data.
CSUPRF - Chip select indicates 32016 requests register file access.
WR - Indicates 32016 register file request is a write.
RD - Indicates 32016 register file request is a read.
A1 - Indicates 32016 register file bus cycle is MS word.

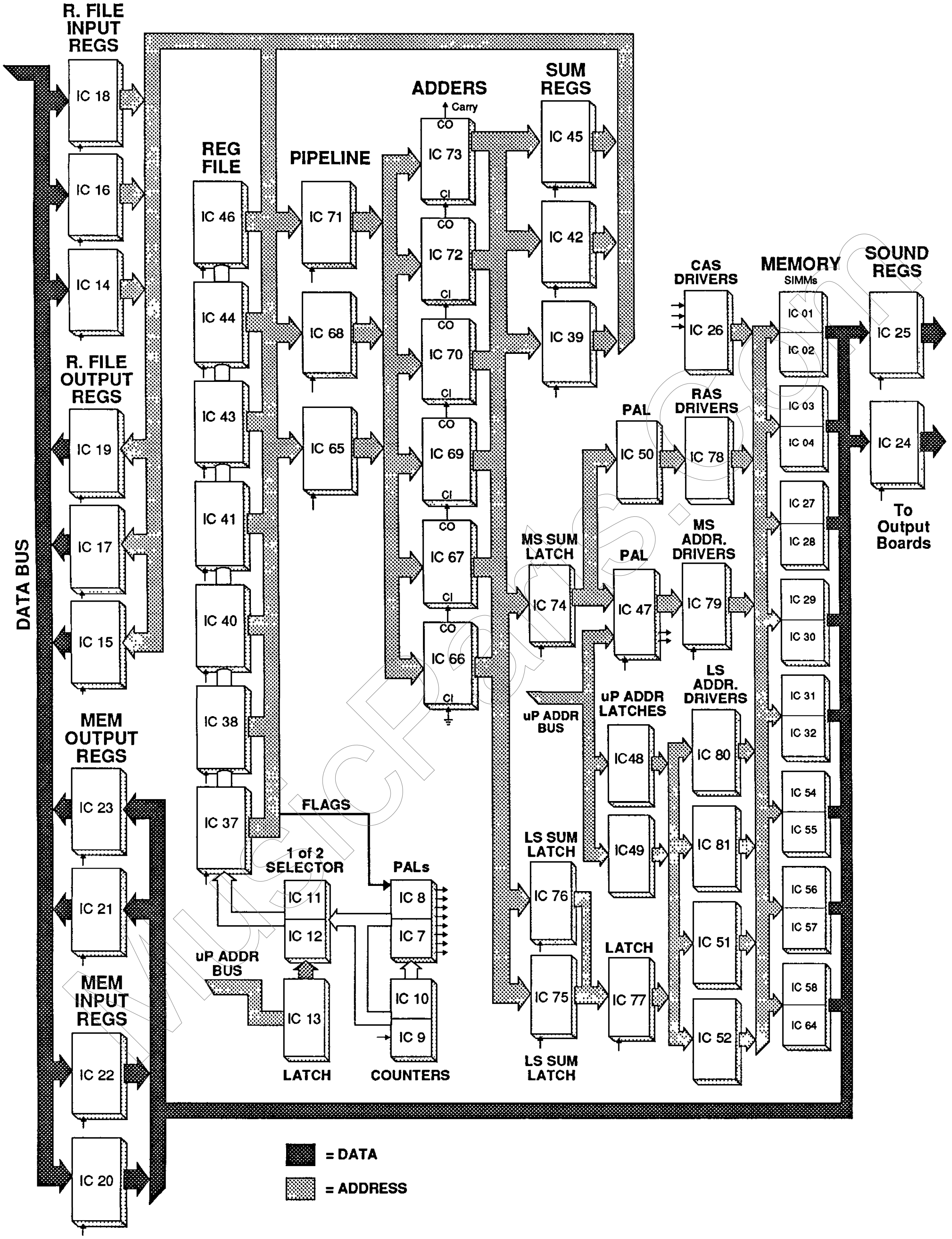
OUTPUTS:
DECOUT - Pipelines output causes adder to decrement when true.
INITOUT - Input to shift register which gives INITFLG next cycle.
INIT - Address to register file. Indicates "initial" data when true.
SUMWE - Write enable of adder output into register file.
MLE - Latch enable to memory address latch from adder. Also SR clk.
SDLE - Latch enable for DRAM sound latch data. Also creates ACK.
UPWRPND - Internal PAL signal indicates a 32016 RF write cycle pending.
UPRFSEL - Signal to register file address mux selects 32016 address.
RFWAIT - WAIT signal to 32016 indicating RF access requires more time.
UPRFWE - Write enable of UP data latch into register file.
UPRFDLE - Latch Enable of 32016 data bus into RF UP address latch.
ALVALID - Internal PAL signal indicating RF UP address latch valid.
ENBXFR - Signal indicating RF data transfer occurring. Decoded by WR,A1
ENUPMSRD - Enable MS word of RF UP data latch onto 32016 bus.
ENUPLSRD - Enable LS word of RF UP data latch onto 32016 bus.
UPRFMSWR - Write 32016 data bus into MS work of RF UP write register.
UPRFLSWR - Write 32016 data bus into LS work of RF UP write register.

The Register File UP Write Register consists of 25 bits of data in three 74HCT374's and a 74HCT74. This data is written by the 32016 and placed in the Register File during a 32016 Register File Request.

The RF UP Address Latch is a 74HCT373 holding 6-bits of address for UP Register File requests. The 4 MS bits are the channel number (0-15), the LS bit is 1 for initial data and 0 for current data. The next least significant bit (bit 1) is 0 for size data and 1 for address data. The "flags" are stored in the current size.

The RF UP Address Latch is multiplexed with the channel select, UCNT0, and INIT signals by a 74AS158 to form the register file address.

EIII MICROCONTROLLER



The RF UP Data Latch consists of 3 74ALS373 latches and a 74S74. This latch holds data read from the register file on a UP RF Read for the 32016. It's output is enabled by the ENUPLSRD and ENUPMSRD signals.

The Register File Write Enable (RFWE) signal is simply the OR combination of SUMWE and UPRFWE qualified by the second half of the clock period.
he INIT shift register consists of two 74HCT164's. These delay the INIT signal 16 channel times to produce the INITFLG for the next channel cycle.

CAS to the DRAM and ACK to the FCHIPs are delayed 1/2 cycle for timing reasons.

The register file is a 64 word by 25 bit RAM built of seven 2149H's.

The output of the register file is latched by the 74LS374's and a 74S74. This produces the " .1 " pipeline delayed register file data.

The Sound Data Memory is contained by a 22V10 PAL. See the PAL source for details. The signals involved are:

INPUTS:

UCNT0	- Microcode State Counter (0-3)
UCNT1	- Microcode State Counter (0-3)
PREVREQ	- Delayed CHREQ signal indicates is FCHIP needs data
RFSHTIM	- Periodic Signal indicated time for a refresh cycle
A23, A22	- 32016 Addr. Bus Signals. If 01 or 10 indicate SDRAM access
RD, WR	- 32016 control signals indicate SDRAM access is read or write
HBE, A0	- 32016 control signals indicate which byte of SDRAM active

OUTPUTS:

UPASEL	- Selects 32016 address for SDRAM access
RASEN	- Signal to the RS PAL enables RAS to the DRAMs
WEL	- Write Enable Signal to the DRAM low bytes
WEH	- Write Enable Signal to the DRAM high bytes
CASEN	- Enables CAS to the DRAMs
SDWAIT	- WAIT signal to the 32016 during SDRAM access
UPLE	- Latch Enable to the 32016 Sound Data Latches
UPOE	- Output Enable signal puts 32016 Sound Data Latches on bus
RFSHREQ	- Internal PAL signal indicates refresh operation pending

The DRAM RAS signals are created by a PAL from the most significant portions of the latched adder output (LSUM) or the 32016 Address Bus, depending on UPASEL. The signals RASEN, CASEN, and UCNT1 are used to time the RAS signals and to determine if all RAS's are active for a refresh operation.

The adder consists of seven 74F283 adders. The inputs are the DEC signals and the RFD.1 pipelined register file data. The output of the adder is latched in the memory address latch (LSUM/RA), and also in a temporary latch for re-writing the register file.

The output of the LSUM latches is multiplexed, along with the 32016 address, to form the DRAM addresses. The multiplexing is determined by UCNT1, which selects row or column address, and UPASEL, which selects 32016 or LSUM data. The multiplexing is done on a tri-state RA bus, which is then latched and buffered to form the buffered RAM address (BRA). The one or two additional BRA signals for 256K or 1M DRAMS are created using the RA PAL, a 16R4.

The 16-bit sound data for the FCHIP is stored in two 74HCT373's, as is the data for the 32016. The 32016 data bus is also buffered onto the DRAM data bus using two 74CT244's.
The DRAM array consists of up to 16 256K x 8 DRAM SIPs, or 8 1M x 8 SIPs.

Oversampling in the Emulator III

by Riley Smith

Digital Audio has become quite familiar since the advent of the Compact Disc. Compact Disc players are being billed as the "ultimate in sound reproduction" with "unmeasurable noise and distortion" and that the lowest priced CD player sounds as good as the most expensive. Now that 16-bit sampling instruments have finally become affordable we hear comparisons being made between CD players and Sampling Instruments. "CD quality sampling" and the ubiquitous "96 db S/N" read the promotions of the new 16-bit machines.

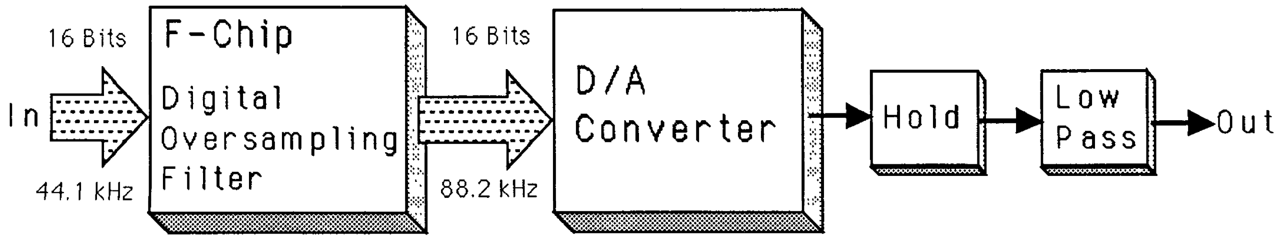
Is all 16-bit digital audio created equal? If the differences are not audible, why spend more money than you have to? There may be more to digital audio equipment then the spec sheets reveal.

I had an interesting and educational experience about a year ago when I decided to purchase my first CD player. I had shopped around, compared the spec sheets of the various brands, and settled on a model that seemed to have all the bells and whistles that I wanted. One of the first things I did was to buy the CD version of some of my favorite records so that I could hear all the new detail and sonic clarity. Well, the CDs did sound cleaner and I sure didn't miss those clicks and pops, but there seemed to be something else missing. The music from the CD player seemed to be coming from behind a thick curtain, while the music from the record seemed to be alive in the room. What was wrong with this audio picture? I asked an audiophile friend of mine and was told that the problem was the output filtering on my new CD player and that I should try another brand. He recommended I look for one that used a technique called "oversampling" which effectively doubled or quadrupled the output sample rate and required less output filtering. I took the CD player back to my dealer to compare it with the same model and with other brands. No, the CD player wasn't defective but when I A/B'd it with another brand (one that used oversampling), the difference was like night and day. I returned home with the oversampling unit and have been happy ever since. Since my ears could easily tell the difference between the two units and since the noise and distortion specs were virtually identical, I concluded that there must be types of distortion or coloration that do not appear in standard spec sheets.

Sampling instruments are not CD players although they do have similarities. One of the differences between a CD player and a sampling musical instrument is that on a CD player the sample rate stays at a constant 44.1 kHz while on a sampling instrument, the sample rate changes as a function of the pitch of the sound.[1] This at once creates problems in that it is much easier to design a good fixed frequency reconstruction (lowpass) filter than one that can track a variable sample rate. The reconstruction filter must have a slope which is steep enough to remove the "staircase voltage" harmonics created by the D/A process without seriously cutting into the audio band and at the same time be "quiet" enough to be considered for use in a "no compromise" 16-bit system. We have found that when a four-pole reconstruction filter is used (as is common on 12-bit machines), most users will use less filtering than needed to remove the "staircase voltage" or "clock" noise in an attempt to preserve the high frequency content of the sound. This results in a harsh, grainy sound as well as excess background noise.

The Emulator III uses custom designed oversampling filters and custom low-noise analog tracking filters to deal with the problem of output filtering. The oversampling filter chip actually doubles the sample rate by examining a number of samples on either side of the current sample and inserting a new sample at the appropriate level. This means that a sound sampled at 44.1 kHz played back at its original pitch will actually be output at 88.2 kHz and if transposed up one octave will appear at the output at a 176.4 kHz rate!

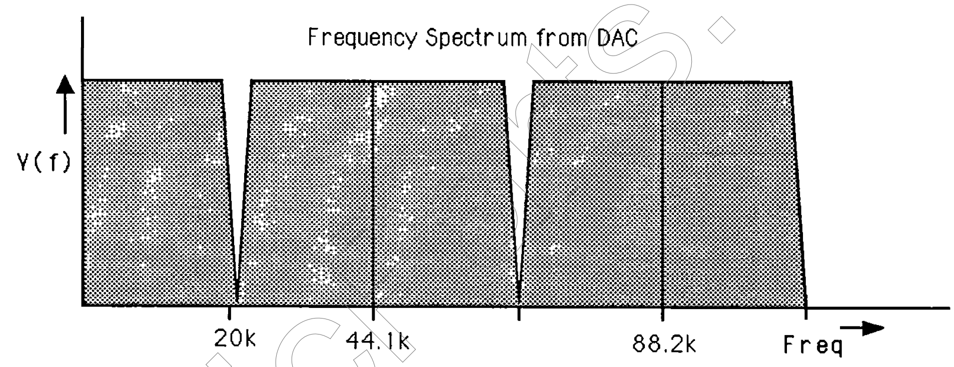
[1] It should be noted that some sampling instruments shift pitch using a constant sample rate, but since most of these methods cause serious distortion we will not consider them for use in a high quality system.



The use of oversampling has several important advantages in a sampling instrument.
First, because the quantization noise is distributed over an area which is twice as great, the residual noise within the audio bandwidth is only one half of the original figure. This adds 3 dB to the signal-to-noise specification.
Second, when using sampling instruments, the pitch of a sound is often shifted down an octave or more to create an unusual sound effect. Normally when this is done, the sample rate begins to enter the audio spectrum and it becomes audible. By use of oversampling the sample rate does not begin to enter the audio range until it has been shifted down two octaves.
Third, since the sample rate is now an octave higher, so are the "staircase voltage" harmonics. They can now be filtered out with a much simpler filter (ie. one with much less phase distortion and ripple).
The oversampling filter can thus be thought of as a zero phase-shift digital filter. Phase distortion is accountable for a lifelessness or lack of presence for which digital audio is often criticized.
The analog filter used in the Emulator III is a custom designed, 7-pole, voltage controlled filter with a signal-to-noise ratio of greater than 96 dB. Three poles are used for reconstruction and the remaining four poles are used for traditional synthesizer type signal processing. The reconstruction filtering is automatic and eliminates the audible artifacts of the A/D, D/A process.

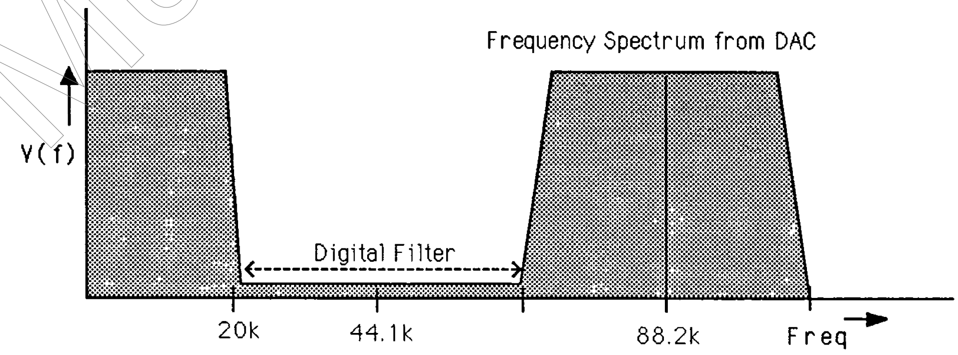
The combination of a phase-linear digital filter with a high performance D/A converter and simple phase-linear analog output filter results in a nearly ideal sampling system for digital audio, the Emulator III.

The frequency plot below shows the frequency spectrum generated from a normal D/A converter system. Note the harmonic frequencies above the audio range that must be filtered out to avoid production of audible intermodulation products.



Normal D/A Converter System

The frequency plot below shows the frequency spectrum generated from the Emulator III D/A converter system. The first harmonics appear far above the audio range at 66 kHz, which requires much less filtering to remove.



EIII Oversampled D/A System

SIGNAL NAME DEFINITIONS

The signal names are comprised of:
(a) + or - specifying active high or low.
(b) An abbreviation of the signal function.
(c) An abbreviation of the signal type.
(d) The signal's destination and source page numbers. Source page is indicated by an + following the page number. If there is no + after a page number then the present page is the source page. If the present page is listed with the destination page then the signal appears again on the same page. There are three types of signals; **D, V** and **I. D** stands for digital, **V** stands for voltage or an analog signal and **I** stands for a current.

Note: The prefix UP on a signal name abbreviation means that its source is the CPU board.

EIII Signal Name List

Name	Source	Destination	Type	Description
+A0.D	CPU 2	CPU 3,5, µC 5	TTL	Main CPU Address Bus 0
+A1.D	CPU 2	CPU 3,5,6,8 ADC 2 µC 1,8	TTL	Main CPU Address Bus 1
+A2.D	CPU 2	CPU 3,5,6,8 µC 2,8	TTL	Main CPU Address Bus 2
+A3.D	CPU 2	CPU 3,5,6 µC 2,8	TTL	Main CPU Address Bus 3
+A4.D	CPU 2	CPU 3,5 µC 2,8 OUT 1	TTL	Main CPU Address Bus 4
-A4.D		OUT 1	TTL	- Main CPU Address Bus 4
+A5.D	CPU 2	CPU 3,5 µC 2,8 OUT 1	TTL	Main CPU Address Bus 5
-A5.D		OUT 1	TTL	- Main CPU Address Bus 5
+A6.D	CPU 2	CPU 3,5 µC 2,8	TTL	Main CPU Address Bus 6
+A7.D	CPU 2	CPU 3,5 µC 2,8	TTL	Main CPU Address Bus 7
+A8.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 8
+A9.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 9
+A10.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 10
+A11.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 11
+A12.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 12

Name	Source	Destination	Type	Description
+A13.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 13
+A14.D	CPU 2	CPU 3,5 µC 8	TTL	Main CPU Address Bus 14
+A15.D	CPU 2	CPU 4,5 µC 8	TTL	Main CPU Address Bus 15
+A16.D	CPU 1	CPU 4,5 µC 8	TTL	Main CPU Address Bus 16
+A17.D	CPU 1	CPU 4 µC 8	TTL	Main CPU Address Bus 17
+A18.D	CPU 1	µC 5,8	TTL	Main CPU Address Bus 18
+A19.D	CPU 1	µC 5,8	TTL	Main CPU Address Bus 19
+A20.D	CPU 1	µC 5,8	TTL	Main CPU Address Bus 20
+A21.D	CPU 1	CPU 4 µC 5	TTL	Main CPU Address Bus 21
+A22.D	CPU 1	CPU 4 µC 5	TTL	Main CPU Address Bus 22
+A23.D	CPU 1	CPU 4 µC 5	TTL	Main CPU Address Bus 23
ACK.D	µC 3	OUT 1	TTL	Channel Acknowledge
-ACK0.D	OUT 1	OUT 4	TTL	Channel Acknowledge Ch.0
-ACK1.D	OUT 1	OUT 5	TTL	Channel Acknowledge Ch.1
-ACK2.D	OUT 1	OUT 6	TTL	Channel Acknowledge Ch.2
-ACK3.D	OUT 1	OUT 7	TTL	Channel Acknowledge Ch.3
-ACK4.D	OUT 1	OUT 8	TTL	Channel Acknowledge Ch.4
-ACK5.D	OUT 1	OUT 9	TTL	Channel Acknowledge Ch.5
-ACK6.D	OUT 1	OUT 10	TTL	Channel Acknowledge Ch.6
-ACK7.D	OUT 1	OUT 11	TTL	Channel Acknowledge Ch.7
+ACV0.V	OUT 3	OUT 4	Voltage	Gain Control Voltage Ch. 0
+ACV1.V	OUT 3	OUT 5	Voltage	Gain Control Voltage Ch. 1
+ACV2.V	OUT 3	OUT 6	Voltage	Gain Control Voltage Ch. 2
+ACV3.V	OUT 3	OUT 7	Voltage	Gain Control Voltage Ch. 3
+ACV4.V	OUT 3	OUT 8	Voltage	Gain Control Voltage Ch. 4
+ACV5.V	OUT 3	OUT 9	Voltage	Gain Control Voltage Ch. 5
+ACV6.V	OUT 3	OUT 10	Voltage	Gain Control Voltage Ch. 6
+ACV7.V	OUT 3	OUT 11	Voltage	Gain Control Voltage Ch. 7
+ADCDTA.D	ADC 4	ADC 1	TTL	Serial Data from ADC
-ADCWAIT.D	ADC 2	CPU 4	TTL	A/D Converter Wait
-ADS.D	CPU 1	CPU 5	TTL	Address Strobe
AGND			GND	Analog Ground
+BCLK.D	ADC 2	ADC 1,4	TTL	Serial Bit Clock
+BD0.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 0
+BD1.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 1
+BD2.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 2
+BD3.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 3
+BD4.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 4
+BD5.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 5
+BD6.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 6
+BD7.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 8
+BD9.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 9
+BD10.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 10
+BD11.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 11

Name	Source	Destination	Type	Description
+BD12.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 12
+BD13.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 13
+BD14.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 14
+BD15.D	CPU 2	CPU 9	TTL	Bi-Directional Data Bus 15
-BDSEL.D	µC 1	OUT 1	TTL	Board Select
(+CHSL3.D)				
+BRA0.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 0 Bank A
+BRA0.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 0 Bank B
+BRA0.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 0 Bank C
+BRA0.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 0 Bank D
+BRA1.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 1 Bank A
+BRA1.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 1 Bank B
+BRA1.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 1 Bank C
+BRA1.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 1 Bank D
+BRA2.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 2 Bank A
+BRA2.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 2Bank B
+BRA2.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 2 Bank C
+BRA2.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 2 Bank D
+BRA3.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 3 Bank A
+BRA3.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 3 Bank B
+BRA3.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 3 Bank C
+BRA3.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 3 Bank D
+BRA4.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 4 Bank A
+BRA4.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 4 Bank B
+BRA4.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 4 Bank C
+BRA4.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 4 Bank D
+BRA5.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 5 Bank A
+BRA5.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 5 Bank B
+BRA5.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 5 Bank C
+BRA5.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 5 Bank D
+BRA6.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 6 Bank A
+BRA6.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 6 Bank B
+BRA6.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 6 Bank C
+BRA6.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 6 Bank D
+BRA7.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 7 Bank A
+BRA7.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 7 Bank B
+BRA7.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 7 Bank C
+BRA7.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 7 Bank D
+BRA8.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 8 Bank A
+BRA8.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 8 Bank B
+BRA8.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 8 Bank C
+BRA8.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 8 Bank D
+BRA9.A.D	µC 8	µC 9	TTL	Buff. RAM Addr. 9 Bank A
+BRA9.B.D	µC 8	µC 9	TTL	Buff. RAM Addr. 9 Bank B
+BRA9.C.D	µC 8	µC 10	TTL	Buff. RAM Addr. 9 Bank C
+BRA9.D.D	µC 8	µC 10	TTL	Buff. RAM Addr. 9 Bank D
-BUFFEN.D	CPU 4	CPU 2	TTL	Enable Off Board Buffer
+C1M.D	CPU 1	CPU 3,5	TTL	1 MHz Clock
+C2M.D	CPU 1	ADC 2	TTL	2 MHz Clock
+C4M.D	CPU 1	CPU 7,8	TTL	4 MHz Clock
+C8M.D	CPU 1	CPU 6	TTL	8 MHz Clock

Name	Source	Destination	Type	Description
+C10M.D	CPU 1	CPU 3,5,8 ADC 2 µC 1,8	TTL	10 MHz Clock
-C10M.D	CPU 1	µC 3,5,6	TTL	Inverted 10 MHz Clock
+C10MD.D	µC 3	µC 5,7	TTL	10 MHz Clock
+C16M.D	CPU 1	CPU 1	TTL	16 MHz Clock
+C20M.D	CPU 1	CPU 5 OUT 4	TTL	20 MHz Clock
+CARRY.D	µC 7	µC 1	TTL	µController Carry
-CAS0.D	µC 3	µC 9	TTL	CAS Row 0
-CAS1.D	µC 3	µC 9	TTL	CAS Row 1
-CAS2.D	µC 3	µC 9	TTL	CAS Row 2
-CAS3.D	µC 3	µC 9	TTL	CAS Row 3
-CAS4.D	µC 3	µC 10	TTL	CAS Row 4
-CAS5.D	µC 3	µC 10	TTL	CAS Row 5
-CAS6.D	µC 3	µC 10	TTL	CAS Row 6
-CAS7.D	µC 3	µC 10	TTL	CAS Row 7
-CASEN.D	µC 5	µC 3	TTL	CAS Enable
+CHCV0.D	OUT 2	OUT 3	TTL	CV Select 0
+CHCV1.D	OUT 2	OUT 3	TTL	CV Select 1
+CHCV2.D	OUT 2	OUT 3	TTL	CV Select 2
+CHREQ.D	OUT 1	µC 1	TTL	Channel Request
-CHREQA.D	OUT 1	µC 1	TTL	Channel Request A
-CHREQB.D	OUT 1	µC 1	TTL	Channel Request B
+CHSL0.D	µC 1	µC 2 OUT 1	TTL	Channel Select 0
+CHSL1.D	µC 1	µC 2 OUT 1	TTL	Channel Select 1
+CHSL2.D	µC 1	µC 2 OUT 1	TTL	Channel Select 2
+CHSL3.D	µC 1	µC 2	TTL	Channel Select 3
-CLEAR.D	ADC 2	ADC 2	TTL	Sample Rate Counter Clear
+CO4.D	µC 6	µC 7	TTL	Carry Out
-CSADC.D	CPU 4	ADC 2	TTL	A/D Converter Chip Select
-CSCVDAC.D	CPU 4	OUT 2	TTL	CV DAC Chip Select
-CSCVDAV.D	CPU 4	ADC 2	TTL	CV DAC Chip Select
-CSDDT.D	CPU 4		TTL	Debugger Chip Select
-CSDRAM	CPU 4	CPU 5	TTL	Dynamic RAM Chip Select
-CSEWE.D	CPU 4	CPU 9	TTL	Misc Bit Latch Chip Select
-CSFCHIP.D	CPU 4	OUT1	TTL	F-Chip Chip Select
-CSFDC.D	CPU 4	CPU 6	TTL	Floppy Controller Chip Select
-CSFP.D	CPU 4	CPU 9	TTL	Front Panel Port Chip Select
-CSHDC.D	CPU 4	CPU 6	TTL	HD Controller Chip Select
-CSHDD.D	CPU 4	CPU 4,6	TTL	Hard Disk Drive Chip Select
-CSLED.D	CPU 4	CPU 9	TTL	LED Chip Select
-CSMET.D	CPU 4	CPU 9	TTL	Metronome Chip Select
-CSPROM.D	CPU 4	CPU 3	TTL	PROM Chip Select
-CSRF.D	CPU 4		TTL	Register File Chip Select
-CSSCAN.D	CPU 4	CPU 7	TTL	Scanner Chip Select
-CSSCC.D	CPU 4	CPU 8	TTL	Serial Controller Chip Select
-CSSCIACK.D	CPU 8	CPU 4	TTL	Serial Cont. Int. Ack. Chip Sel.
-CSTIMR.D	CPU 4	CPU 3	TTL	Timer Chip Select

Name	Source	Destination	Type	Description
-CSUPRF.D (-CSRF.D)	CPU 4	µC 1	TTL	µProc Reg File Chip Select
-CSWCVDAC.D	CPU 4	OUT 2	TTL	Write CV DAC Chip Select
+CV.V	OUT 2	OUT 3	Voltage	CV DAC Output
-CWAIT.D	CPU 4		TTL	Continuous Wait
+CYCREQ.D	CPU 5	CPU 4	TTL	RAM Cycle Request
+D0.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 0
+D1.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 1
+D2.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 2
+D3.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 3
+D4.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 4
+D5.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 5
+D6.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 6
+D7.D	CPU 1	CPU 2,3,5,6,7,8	Tri-State	Main CPU Data Bus 7
+D8.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 8
+D9.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 9
+D10.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 10
+D11.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 11
+D12.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 12
+D13.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 13
+D14.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 14
+D15.D	CPU 1	CPU 2,5	Tri-State	Main CPU Data Bus 15
-DBE.D	CPU 1	CPU 4	TTL	Data Buffer Enable
-DDIN.D	CPU 1	CPU 2	TTL	Data Direction In
+DEC1.D	µC 1	µC 6	TTL	Buffered Decrement Flag
+DEC2.D	µC 1	µC 6	TTL	Buffered Decrement Flag
+DEC3.D	µC 1	µC 6	TTL	Buffered Decrement Flag
+DEC4D	µC1	µC 6	TTL	Buffered Decrement Flag
+DEC5.D	µC 1	µC 7	TTL	Buffered Decrement Flag
+DEC6.D	µC 1	µC 7	TTL	Buffered Decrement Flag
+DECFLG.D	µC 2	µC 1,4,5,7	TTL	Decrement Flag
+DECFLGWT.D	UC 1	µC 7	TTL	Decrement Flag Wait
DGND			GND	Digital Ground
-DIR.D	CPU 6	Floppy Disk	TTL	Floppy Disk Head Direction
+ELCD.D	CPU 4	Front Panel	TTL	LCD Chip Enable
-ENAH.D	ADC 2		TTL	VCA Demux. Enable High
-ENAL.D	ADC 2	OUT 3	TTL	VCA Demux. Enable Low
-ENBXFR.D	µC 1	µC 1	TTL	Enable Transfer
-ENFH.D	ADC 2	OUT 3	TTL	Filter Demux. Enable High
-ENFL.D	ADC 2	OUT 3	TTL	Filter Demux. Enable Low
-ENPH.D	ADC 2	OUT 3	TTL	Pan Demux. Enable High
-ENPL.D	ADC 2	OUT 3	TTL	Pan Demux. Enable Low
-ENQH.D	ADC 2	OUT 3	TTL	Q Demux. Enable High
-ENQL.D	ADC 2	OUT 3	TTL	Q Demux. Enable Low
-ENSCBUF.D	CPU 4	CPU 3	TTL	SCSI Buffer Enable
-ENUPLSRD.D	µC 1	µC 3	TTL	µP- Reg File LS Read
-ENUPMSRD.D	µC 1	µC 3	TTL	µP- Reg File MS Read
-ENXH.D	ADC 2	OUT 3	TTL	3-Pole Filter Demux. Ena. High
-ENXL.D	ADC 2	OUT 3	TTL	3-Pole Filter Demux. Ena. Low
-FCS0.D	OUT 1	OUT 4	TTL	F-Chip Chip Select Ch. 0
-FCS1.D	OUT 1	OUT 5	TTL	F-Chip Chip Select Ch. 1
-FCS2.D	OUT 1	OUT 6	TTL	F-Chip Chip Select Ch. 2

Name	Source	Destination	Type	Description
-FCS3.D	OUT 1	OUT 7	TTL	F-Chip Chip Select Ch. 3
-FCS4.D	OUT 1	OUT 8	TTL	F-Chip Chip Select Ch. 4
-FCS5.D	OUT 1	OUT 9	TTL	F-Chip Chip Select Ch. 5
-FCS6.D	OUT 1	OUT 10	TTL	F-Chip Chip Select Ch. 6
-FCS7.D	OUT 1	OUT 11	TTL	F-Chip Chip Select Ch. 7
+FCV0.V	OUT 3	OUT 4	Voltage	Filter Control Voltage Ch.0
+FCV1.V	OUT 3	OUT 5	Voltage	Filter Control Voltage Ch.1
+FCV2.V	OUT 3	OUT 6	Voltage	Filter Control Voltage Ch.2
+FCV3.V	OUT 3	OUT 7	Voltage	Filter Control Voltage Ch.3
+FCV4.V	OUT 3	OUT 8	Voltage	Filter Control Voltage Ch.4
+FCV5.V	OUT 3	OUT 9	Voltage	Filter Control Voltage Ch.5
+FCV6.V	OUT 3	OUT 10	Voltage	Filter Control Voltage Ch.6
+FCV7.V	OUT 3	OUT 11	Voltage	Filter Control Voltage Ch.7
+FDCINT.D	CPU 6	CPU 4	TTL	Floppy Disk Contr. Interrupt
+FORBKFLG.1.D	µC 5	µC 7	TTL	Latched Forw./Backw. Flag
+FORBKFLG.D	µC 2	µC 1,3,4,5,7	TTL	Forward/Backward Flag
+FPSWX1.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 1
+FPSWX2.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 2
+FPSWX3.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 3
+FPSWX4.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 4
+FPSWX5.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 5
+FPSWX6.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 6
+FPSWX7.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 7
+FPSWX8.D	Frnt Pnl	CPU 9	TTL	Front Panel Switch 8
-HBE.D	CPU 1	CPU 5 µC 5	TTL	High Byte Enable
-HDACK.D	CPU 6	CPU 4	TTL	Hard Disk Acknowledge
+HDINT.D	CPU 6	CPU 4	TTL	Hard Disk Interrupt
+HDREQ.D	CPU 4	CPU 6	TTL	Hard Disk Request
+HVSUP.V	Frnt Pnl	Frnt Pnl	Voltage	LCD High Voltage Supply
+INIT.D	µC 1	µC 2	TTL	Initialize Loop
+INITFLG.D	µC 3	µC 1	TTL	Carry Shift Reg Output
+INITOUT.D	µC 1	µC 3	TTL	Carry Shift Reg Input
-INT.D	CPU 4	CPU 1	TTL	Main CPU Maskable Interrupt
+KHZ44.D	ADC 2	ADC 2,3	TTL	44.1 kHz Sample Clock Select
-KHZ30.D	ADC 2	ADC 2,3	TTL	30 kHz Sample Clock Select
+MIDINTHRU.D	Dig JACK	CPU 8	Current	MIDI Thru
-MIDINTHRU.D	Dig JACK	CPU 8	Current	MIDI Thru
+LCDBIAS.V	CPU 10	F. Panel	Voltage	LCD Angle Voltage
-LCDBIAS.V	CPU 10	F. Panel	Voltage	LCD Angle Voltage
+LCH.I	ADC 3	ADC 4	Current	Left Ch.Current (from Mux)
+LCH30.V	ADC 1	ADC 3	Voltage	Left Ch. 30K Signal (to Mux)
+LCH44.V	ADC 1	ADC 3	Voltage	Left Ch. 44K Signal (to Mux)
+LCHIN.V	ANJK	ADC 1	Voltage	Left Ch. Sample Input
+LINRNG.V	ADC 2	ADC 1	Voltage	Left Input Range (Unused)
+LOOPFLG.1.D	µC 5	µC 1,7	TTL	Latched Loop Flag
+LOOPFLG.D	µC 2	µC 3,4,5,7	TTL	Loop Flag
+LOUT0.V	OUT 4	JACK 1	Voltage	Left Audio Out Ch. 0
+LOUT1.V	OUT 5	JACK 1	Voltage	Left Audio Out Ch. 1
+LOUT2.V	OUT 6	JACK 1	Voltage	Left Audio Out Ch. 2
+LOUT3.V	OUT 7	JACK 1	Voltage	Left Audio Out Ch. 3
+LOUT4.V	OUT 8	JACK 1	Voltage	Left Audio Out Ch. 4

Name	Source	Destination	Type	Description
+LOUT5.V	OUT 9	JACK 1	Voltage	Left Audio Out Ch. 5
+LOUT6.V	OUT 10	JACK 1	Voltage	Left Audio Out Ch. 6
+LOUT7.V	OUT 11	JACK 1	Voltage	Left Audio Out Ch. 7
+LOUT8.V	OUT 4	JACK 2	Voltage	Left Audio Out Ch. 8
+LOUT9.V	OUT 5	JACK 2	Voltage	Left Audio Out Ch. 9
+LOUT10.V	OUT 6	JACK 2	Voltage	Left Audio Out Ch. 10
+LOUT11.V	OUT 7	JACK 2	Voltage	Left Audio Out Ch. 11
+LOUT12.V	OUT 8	JACK 2	Voltage	Left Audio Out Ch. 12
+LOUT13.V	OUT 9	JACK 2	Voltage	Left Audio Out Ch. 13
+LOUT14.V	OUT 10	JACK 2	Voltage	Left Audio Out Ch. 14
+LOUT15.V	OUT 11	JACK 2	Voltage	Left Audio Out Ch. 15
+LSHSTB.D	ADC 2	ADC 4	TTL	Left Ch. Samp/Hold Strobe
-LSSUMEN.D	µC 6	µC 6	TTL	LS Sum Enable
+LSUM08.D	µC 7	µC 8	TTL	Low Sum Bit 8
+LSUM17.D	µC 7	µC 8	TTL	Low Sum Bit 17
+LSUM18.D	µC 7	µC 5,8	TTL	Low Sum Bit 18
+LSUM19.D	µC 7	µC 5,8	TTL	Low Sum Bit 19
+LSUM20.D	µC 7	µC 5	TTL	Low Sum Bit 20
+LSUM21.D	µC 7	µC 5	TTL	Low Sum Bit 21
-LSUPAEN.D	µC 6	µC 8	TTL	LS µProc Addr Enable
+LWHL.V	Wheel	ADC 3	Voltage	Left Wheel CV
+MDB0.D	µC 11	OUT 4-11	TTL	Sound Data Bus 0
+MDB1.D	µC 11	OUT 4-11	TTL	Sound Data Bus 1
+MDB2.D	µC 11	OUT 4-11	TTL	Sound Data Bus 2
+MDB3.D	µC 11	OUT 4-11	TTL	Sound Data Bus 3
+MDB4.D	µC 11	OUT 4-11	TTL	Sound Data Bus 4
+MDB5.D	µC 11	OUT 4-11	TTL	Sound Data Bus 5
+MDB6.D	µC 11	OUT 4-11	TTL	Sound Data Bus 6
+MDB7.D	µC 11	OUT 4-11	TTL	Sound Data Bus 7
+MDB8.D	µC 11	OUT 4-11	TTL	Sound Data Bus 8
+MDB9.D	µC 11	OUT 4-11	TTL	Sound Data Bus 9
+MDB10.D	µC 11	OUT 4-11	TTL	Sound Data Bus 10
+MDB11.D	µC 11	OUT 4-11	TTL	Sound Data Bus 11
+MDB12.D	µC 11	OUT 4-11	TTL	Sound Data Bus 12
+MDB13.D	µC 11	OUT 4-11	TTL	Sound Data Bus 13
+MDB14.D	µC 11	OUT 4-11	TTL	Sound Data Bus 14
+MDB15.D	µC 11	OUT 4-11	TTL	Sound Data Bus 15

Note: The MDB and the SDB are the same bus.

Name	Source	Destination	Type	Description
-MIDIOF.D	CPU 9	CPU 8	TTL	MIDI Off Control
-MLE.D	µC 1	µC 3,6	TTL	Memory Latch Enable
-MSSUMEN.D	µC 6	µC 6	TTL	MS Sum Enable
-MSUPAEN.D	µC 6	µC 8	TTL	MS µProc Addr Enable
-MTR.D	CPU 9	Floppy Disk	TTL	Floppy Disk Motor Control
+MXA0.D	ADC 2	ADC 3	TTL	Sample Mux. Addr. 0
+MXA1.D	ADC 2	ADC 3	TTL	Sample Mux. Addr. 1
+MXA2.D	ADC 2	ADC 3	TTL	Sample Mux. Addr. 2
-NMI.D	Debug	CPU 1	TTL	Non-Maskable Interrupt
-OE.D	ADC 2	ADC 1	TTL	Sample Data Output Enable
+PEDAL.V	Dig JACK	ADC 3	Voltage	Pedal Control Voltage
-PER.D	CPU 4	CPU 1	TTL	Peripherial Timing

Name	Source	Destination	Type	Description
+PCV0.V	OUT3	OUT 4	Voltage	Pan Control Voltage Ch.0
+PCV1.V	OUT3	OUT 5	Voltage	Pan Control Voltage Ch.1
+PCV2.V	OUT3	OUT 6	Voltage	Pan Control Voltage Ch.2
+PCV3.V	OUT3	OUT 7	Voltage	Pan Control Voltage Ch.3
+PCV4.V	OUT3	OUT 8	Voltage	Pan Control Voltage Ch.4
+PCV5.V	OUT3	OUT 9	Voltage	Pan Control Voltage Ch.5
+PCV6.V	OUT3	OUT 10	Voltage	Pan Control Voltage Ch.6
+PCV7.V	OUT3	OUT 11	Voltage	Pan Control Voltage Ch.7
-POTSON.D	ADC 2	ADC 3	TTL	Panel Pots to ADC Enable
+PRESS.V	Mother	ADC 3	Voltage	Pressure CV
+PREVREQ.D	µC 1	µC 5	TTL	Previous Request Pending
PULLUP.V	OUT 1	OUT 2	Voltage	Pullup Voltage
+QCV0.V	OUT 3	OUT 4	Voltage	Q Control Voltage Ch. 0
+QCV1.V	OUT 3	OUT 5	Voltage	Q Control Voltage Ch. 1
+QCV2.V	OUT 3	OUT 6	Voltage	Q Control Voltage Ch. 2
+QCV3.V	OUT 3	OUT 7	Voltage	Q Control Voltage Ch. 3
+QCV4.V	OUT 3	OUT 8	Voltage	Q Control Voltage Ch. 4
+QCV5.V	OUT 3	OUT 9	Voltage	Q Control Voltage Ch. 5
+QCV6.V	OUT 3	OUT 10	Voltage	Q Control Voltage Ch. 6
+QCV7.V	OUT 3	OUT 11	Voltage	Q Control Voltage Ch. 7
+RA0.D	µC 6	µC 8	TTL	Sound RAM Addr. 0
+RA1.D	µC 6	µC 8	TTL	Sound RAM Addr. 1
+RA2.D	µC 6	µC 8	TTL	Sound RAM Addr. 2
+RA3.D	µC 6	µC 8	TTL	Sound RAM Addr. 3
+RA4.D	µC 6	µC 8	TTL	Sound RAM Addr. 4
+RA5.D	µC 6	µC 8	TTL	Sound RAM Addr. 5
+RA6.D	µC 6	µC 8	TTL	Sound RAM Addr. 6
+RA7.D	µC 6	µC 8	TTL	Sound RAM Addr. 7
+RAD0.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 8
+RAD1.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 9
+RAD2.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 10
+RAD3.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 11
+RAD4.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 12
+RAD5.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 13
+RAD6.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 14
+RAD7.D	µC 11	µC 9,11	TTL	Sound RAM Addr. 15
+RAD8.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 16
+RAD9.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 17
+RAD10.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 18
+RAD11.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 19
+RAD12.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 20
+RAD13.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 21
+RAD14.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 22
+RAD15.D	µC 11	µC 10,11	TTL	Sound RAM Addr. 23
-RAS0.D	µC 5	µC 9	TTL	Row Address Select 0
-RAS1.D	µC 5	µC 9	TTL	Row Address Select 1
-RAS2.D	µC 5	µC 9	TTL	Row Address Select 2
-RAS3.D	µC 5	µC 9	TTL	Row Address Select 3
-RAS4.D	µC 5	µC 10	TTL	Row Address Select 4
-RAS5.D	µC 5	µC 10	TTL	Row Address Select 5
-RAS6.D	µC 5	µC 10	TTL	Row Address Select 6
-RAS7.D	µC 5	µC 10	TTL	Row Address Select 7

Name	Source	Destination	Type	Description
+RCH.I	ADC 3	ADC 4	Current	Right Ch. Current
+RCH30.V	ADC 1		Voltage	Right Ch. 30K Signal (to Mux)
+RCH44.V	ADC 1	ADC 3	Voltage	Right Ch. 44K Signal (to Mux)
+RCHIN.V	ANJK	ADC 1	Voltage	Right Ch. Sample Input
-RD.D	CPU 1	CPU 3,4,6,8 ADC 2 µC 1,5	TTL	Main CPU Read Line
+REFREQ.D (+RFSHTIM.D)	CPU 5	µC 5	TTL	Refresh Request Timer
+REQ0.D	OUT 4	OUT 1	TTL	Channel Request 0
+REQ1.D	OUT 5	OUT 1	TTL	Channel Request 1
+REQ2.D	OUT 6	OUT 1	TTL	Channel Request 2
+REQ3.D	OUT 7	OUT 1	TTL	Channel Request 3
+REQ4.D	OUT 8	OUT 1	TTL	Channel Request 4
+REQ5.D	OUT 9	OUT 1	TTL	Channel Request 5
+REQ6.D	OUT 10	OUT 1	TTL	Channel Request 6
+REQ7.D	OUT 11	OUT 1	TTL	Channel Request 7
-RESET.D	CPU 1	CPU 6,7	TTL	System Reset
+RFA0.D	µC 2	µC 4	TTL	Register File Addr. 0
+RFA1.D	µC 2	µC 4	TTL	Register File Addr. 1
+RFA2.D	µC 2	µC 4	TTL	Register File Addr. 2
+RFA3.D	µC 2	µC 4	TTL	Register File Addr. 3
+RFA4.D	µC 2	µC 4	TTL	Register File Addr. 4
+RFA5.D	µC 2	µC 4	TTL	Register File Addr. 5
+RFD00.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 0
+RFD00.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 0
+RFD01.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 1
+RFD01.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 1
+RFD02.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 2
+RFD02.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 2
+RFD03.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 3
+RFD03.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 3
+RFD04.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 4
+RFD04.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 4
+RFD05.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 5
+RFD05.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 5
+RFD06.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 6
+RFD06.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 6
+RFD07.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 7
+RFD07.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 7
+RFD08.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 8
+RFD08.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 8 +RFD09.1.D
	µC 5	µC 6	TTL	Reg.File Latch Data Bus 9
+RFD09.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 9
+RFD10.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 10
+RFD10.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 10
+RFD11.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 11
+RFD11.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 11
+RFD12.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 12
+RFD12.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 12
+RFD13.1.D	µC 5	µC 6	TTL	Reg.File Latch Data Bus 13
+RFD13.D	µC 2	µC 3,4,5,7	TTL	Reg.File Data Bus 13

Name	Source	Destination	Type	Description
+RFD14.1.D	μC 5	μC 6	TTL	Reg.File Latch Data Bus 14
+RFD14.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 14
+RFD15.1.D	μC 5	μC 6	TTL	Reg.File Latch Data Bus 15
+RFD15.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 15
+RFD16.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 16
+RFD16.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 16
+RFD17.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 17
+RFD17.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 17
+RFD18.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 18
+RFD18.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 18
+RFD19.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 19
+RFD19.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 19
+RFD20.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 20
+RFD20.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 20
+RFD21.1.D	μC 5	μC 6,7	TTL	Reg.File Latch Data Bus 21
+RFD21.D	μC 2	μC 3,4,5,7	TTL	Reg.File Data Bus 21
+RFSHTIM.D (+REFREQ.D)	CPU 5	μC 5	TTL	Refresh Request Timer
-RFWAIT.D	μC 1	CPU 4	TTL	Register File Wait
-RFWE.D	μC 3	μC 4	TTL	Register File Write Enable
+RINRNG.V	ADC 2	ADC 1	Voltage	Right Input Range
+ROUT0.V	OUT 4	JACK 1	Voltage	Right Audio Out Ch. 0
+ROUT1.V	OUT 5	JACK 1	Voltage	Right Audio Out Ch. 1
+ROUT2.V	OUT 6	JACK 1	Voltage	Right Audio Out Ch. 2
+ROUT3.V	OUT 7	JACK 1	Voltage	Right Audio Out Ch. 3
+ROUT4.V	OUT 8	JACK 1	Voltage	Right Audio Out Ch. 4
+ROUT5.V	OUT 9	JACK 1	Voltage	Right Audio Out Ch. 5
+ROUT6.V	OUT 10	JACK 1	Voltage	Right Audio Out Ch. 6
+ROUT7.V	OUT 11	JACK 1	Voltage	Right Audio Out Ch. 7
+ROUT8.V	OUT 4	JACK 2	Voltage	Right Audio Out Ch. 8
+ROUT9.V	OUT 5	JACK 2	Voltage	Right Audio Out Ch. 9
+ROUT10.V	OUT 6	JACK 2	Voltage	Right Audio Out Ch. 10
+ROUT11.V	OUT 7	JACK 2	Voltage	Right Audio Out Ch. 11
+ROUT12.V	OUT8	JACK 2	Voltage	Right Audio Out Ch. 12
+ROUT13.V	OUT 9	JACK 2	Voltage	Right Audio Out Ch. 13
+ROUT14.V	OUT 10	JACK 2	Voltage	Right Audio Out Ch. 14
+ROUT15.V	OUT 11	JACK 2	Voltage	Right Audio Out Ch. 15
+RSHSTB.D	ADC 4	ADC 2,4	TTL	Right Ch. Samp/Hold Strobe
-RST.D			TTL	Reset
-RSTSTI.D	CPU 4	CPU 8	TTL	Reset Software Timer Interrupt
-RSTTGI.D	CPU 4	CPU 3	TTL	Reset Trans. Gen. Interrupt
+RWHL.V	Wheel	ADC 3	Voltage	Right Wheel CV
+SCNCLK.D	CPU 9	CPU 7	TTL	Scanner CPU Data Clock
+SCNDATA.D	CPU 9	CPU 7	TTL	Scanner CPU Data Line
-SCCINT.D	CPU 8	CPU 4	TTL	Scanner CPU Interrupt
-SCNINT.D	CPU 7	CPU 4	TTL	Scanner Interrupt
+SD00.D	μP 11	OUT 4-11	TTL	Sound Data Bus 0
+SD01.D	μP 11	OUT 4-11	TTL	Sound Data Bus 1
+SD02.D	μP 11	OUT 4-11	TTL	Sound Data Bus 2
+SD03.D	μP 11	OUT 4-11	TTL	Sound Data Bus 3
+SD04.D	μP 11	OUT 4-11	TTL	Sound Data Bus 4
+SD05.D	μP 11	OUT 4-11	TTL	Sound Data Bus 5

Name	Source	Destination	Type	Description
+SD06.D	µP 11	OUT 4-11	TTL	Sound Data Bus 6
+SD07.D	µP 11	OUT 4-11	TTL	Sound Data Bus 7
+SD08.D	µP 11	OUT 4-11	TTL	Sound Data Bus 8
+SD09.D	µP 11	OUT 4-11	TTL	Sound Data Bus 9
+SD10.D	µP 11	OUT 4-11	TTL	Sound Data Bus 10
+SD11.D	µP 11	OUT 4-11	TTL	Sound Data Bus 11
+SD12.D	µP 11	OUT 4-11	TTL	Sound Data Bus 12
+SD13.D	µP 11	OUT 4-11	TTL	Sound Data Bus 13
+SD14.D	µP 11	OUT 4-11	TTL	Sound Data Bus 14
+SD15.D	µP 11	OUT 4-11	TTL	Sound Data Bus 15

Note: The MDB and the SDB are the same bus.

Name	Source	Destination	Type	Description
+SDLE.D	µC 1	µC 3,11	TTL	Sound Data Latch Enable
-SDWAIT.D	µC 5	CPU 4	TTL	Sound Data Wait
+SHSTB.D	OUT 4-11	OUT 4-11	TTL	Sample/Hold Strobe
+SLDR1.V	FNTPNL	ADC 3	Voltage	Slider 1 Voltage
+SLDR2.V	FNTPNL	ADC 3	Voltage	Slider 2 Voltage
+SMPLVOL.V	FNTPNL	ADC 1	Voltage	Sample Input Control Voltage
+SMPTCLK.D	CPU 8	CPU 3	TTL	SMPTE Clock
-SIDE.D	CPU 9	Floppy Disk	TTL	Floppy Disk Side Select
-SMPTIME.D	CPU 3	CPU 8	TTL	SMPTE Time Out
-SPC.D	CPU 2	CPU 1	TTL	Slave Processor Control
+SPLVOL.V	FNTPNL	ADC 1	Voltage	Sample In Volume
-STEP.D	CPU 6	Floppy Disk	TTL	Floppy Disk Head Step
+STINT.D	CPU 8	CPU 4	TTL	Software Timer Interrupt
+ST0.D	CPU 1	CPU 2	TTL	µProc Status Bit 0
+ST1.D	CPU 1	CPU 2	TTL	µProc Status Bit 1
+SUM00.D	µC 6	µC 7	TTL	µC Sum Bit 0
+SUM01.D	µC 6	µC 7	TTL	µC Sum Bit 1
+SUM02.D	µC 6	µC 7	TTL	µC Sum Bit 2
+SUM03.D	µC 6	µC 7	TTL	µC Sum Bit 3
+SUM04.D	µC 6	µC 7	TTL	µC Sum Bit 4
+SUM05.D	µC 6	µC 7	TTL	µC Sum Bit 5
+SUM06.D	µC 6	µC 7	TTL	µC Sum Bit 6
+SUM07.D	µC 6	µC 7	TTL	µC Sum Bit 7
+SUM08.D	µC 6	µC 7	TTL	µC Sum Bit 8
+SUM09.D	µC 6	µC 7	TTL	µC Sum Bit 9
+SUM10.D	µC 6	µC 7	TTL	µC Sum Bit 10
+SUM11.D	µC 6	µC 7	TTL	µC Sum Bit 11
+SUM12.D	µC 6	µC 7	TTL	µC Sum Bit 12
+SUM13.D	µC 6	µC 7	TTL	µC Sum Bit 13
+SUM14.D	µC 6	µC 7	TTL	µC Sum Bit 14
+SUM15.D	µC 6	µC 7	TTL	µC Sum Bit 15
+SUM16.D	µC 7	µC 6	TTL	µC Sum Bit 16
+SUMLE.D	µC 6	µC 7	TTL	Sum Latch Enable
-SUMWE.D	µC 1	µC 3,7	TTL	µC Sum Write Enable
-SWTIME.D	CPU 3	CPU 8	TTL	Software Timer Clock
+SYNCON.D	CPU 9	CPU 8	TTL	Ext. Sync On
-TGINT.D	CPU 3	CPU 4	TTL	Transient Generator Interrupt
-TSO.D	CPU 1	CPU 4,	TTL	Fast Read or Write
+UCNT0.D	µC 1	µC 2,5	TTL	µC Clock 0

Name	Source	Destination	Type	Description
+UCNT1.D	µC 1	µC 5,6,8	TTL	µC Clock 1
+UPA1.D	CPU 2	OUT1,2	TTL	µProc Address 1
+UPA2.D	CPU 2	OUT1,2	TTL	µProc Address 2
+UPA3.D	CPU 2	OUT1,2	TTL	µProc Address 3
+UPA4.D	CPU 2	ADC 2	TTL	µProc Address 4
+UPA5.D	CPU 2	ADC 2	TTL	µProc Address 5
+UPA6.D	CPU 2	ADC 2	TTL	µProc Address 6
+UPA7.D	CPU 2	ADC 2	TTL	µProc Address 7

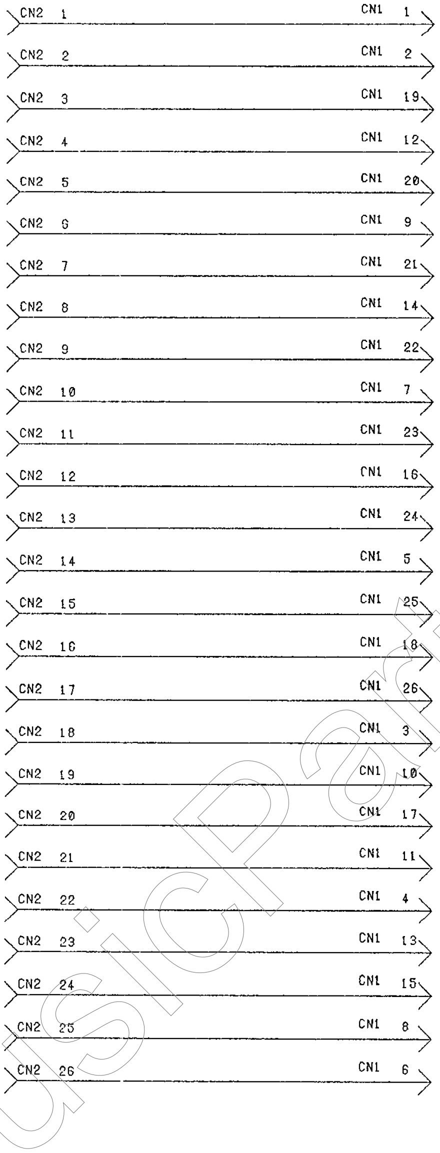
Note: The UPA and A are the same bus.

Name	Source	Destination	Type	Description
+UPADS.D	CPU 1	CPU 2	TTL	µProc Address Select
+UPASEL.D	µC 5	µC 6,8	TTL	µProc Address -> RAM
+UPD00.D	CPU 2	µC 2,3,11	TTL	µProc Data Bus 0
+UPD01.D	CPU 2	µC 2,3,11	TTL	µProc Data Bus 1
+UPD02.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 2
+UPD03.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 3
+UPD04.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 4
+UPD05.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 5
+UPD06.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 6
+UPD07.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 7
+UPD08.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 8
+UPD09.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 9
+UPD10.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 10
+UPD11.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 11
+UPD12.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 12
+UPD13.D	CPU 2	µC 2,3,11 OUT2	TTL	µProc Data Bus 13
+UPD14.D	CPU 2	µC 2,3,11	TTL	µProc Data Bus 14
+UPD15.D	CPU 2	µC 2,3,11	TTL	µProc Data Bus 15

Note: The UPD and BD are the same bus.

Name	Source	Destination	Type	Description
+UPLE.D	µC 5	µC 11	TTL	Snd RAM -> µProc Latch Ena
-UPOE.D	µC 5	µC 11	TTL	µProcessor Output Enable
+UPRFALE.D	µC 1	µC 2	TTL	µProc Reg File Addr Latch Ena
-UPRFLSWR.D	µC 1	µC 2	TTL	µProc Reg File LS Write
-UPRFMSWR.D	µC 1	µC 2	TTL	µProc Reg File MS Write
+UPRFDLE.D	µC 1	µC 3	TTL	µProc Reg File Data Latch Ena
+UPRFSEL.D	µC 1	µC 2	TTL	Reg File Addr Select
-UPRFWE.D	µC 1	µC 2,3	TTL	µProc Reg File Write Enable

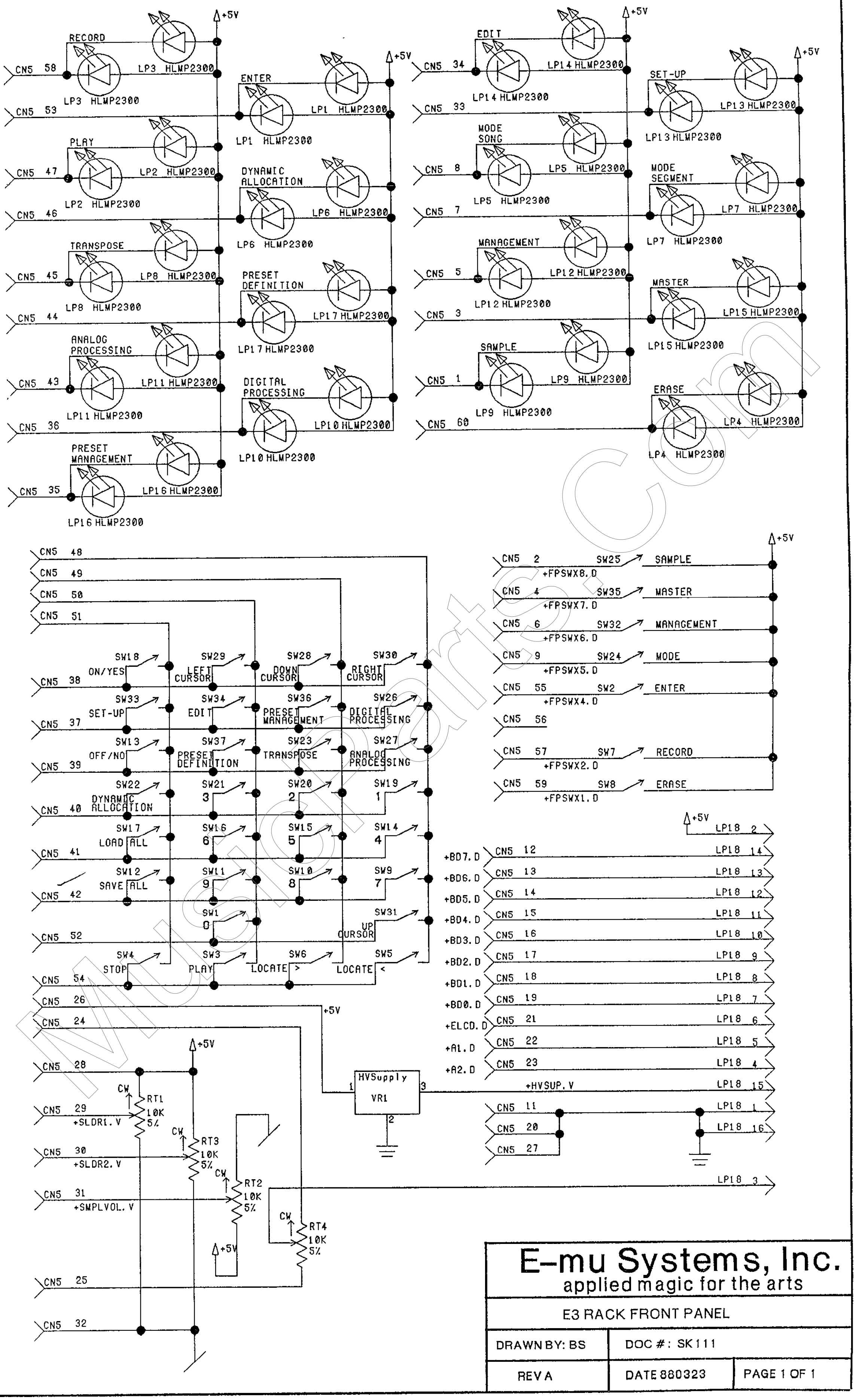
Name	Source	Destination	Type	Description
-UPRWAIT.D	CPU 5	CPU 4	TTL	µProcessor Wait
-WAIT1.D	CPU 4	CPU 1	TTL	Add 1 Wait State
-WAIT2.D	CPU 4	CPU 1	TTL	Add 2 Wait State
-WDTA.D	CPU 6	Floppy Disk	TTL	Floppy Disk Write Data
+WEH.D	µC 5	µC 6,11	TTL	RAM Write Enable High
-WEH0.D	µC 5	µC 9	TTL	RAM Write Enable High 0
-WEH1.D	µC 5	µC 9	TTL	RAM Write Enable High 1
-WEH2.D	µC6	µC 10	TTL	RAM Write Enable High 2
-WEH3.D	µC 6	µC 10	TTL	RAM Write Enable High 3
+WEL.D	µC 5	µC 6,11	TTL	RAM Write Enable Low
-WEL0.D	µC 5	µC 9	TTL	RAM Write Enable Low 0
-WEL1.D	µC 5	µC 9	TTL	RAM Write Enable Low 1
-WEL2.D	µC 5	µC 10	TTL	RAM Write Enable Low 2
-WEL3.D	µC 5	µC 10	TTL	RAM Write Enable Low 3
-WG.D	CPU 6	Floppy Disk	TTL	Floppy Disk Write Gate
-WP.D	CPU 6	Floppy Disk	TTL	Floppy Disk Write Protect
-WR.D	CPU 1	CPU 4,5,6,8 ADC 2 µC 5	TTL	Main CPU Write Line
+XCV0.V	OUT 3	OUT 4	Voltage	Reconstr. Filter CV Ch. 0
+XCV1.V	OUT 3	OUT 5	Voltage	Reconstr. Filter CV Ch. 1
+XCV2.V	OUT 3	OUT 6	Voltage	Reconstr. Filter CV Ch. 2
+XCV3.V	OUT 3	OUT 7	Voltage	Reconstr. Filter CV Ch. 3
+XCV4.V	OUT 3	OUT 8	Voltage	Reconstr. Filter CV Ch. 4
+XCV5.V	OUT 3	OUT 9	Voltage	Reconstr. Filter CV Ch. 5
+XCV6.V	OUT 3	OUT 10	Voltage	Reconstr. Filter CV Ch. 6
+XCV7.V	OUT 3	OUT 11	Voltage	Reconstr. Filter CV Ch. 7



E–mu Systems, Inc.
applied magic for the arts

E3 KEYBOARD MOD PCB

DRAWN BY: BFS	DOC # · SK 114	
REV A	DATE 871022	PAGE 1 OF 1

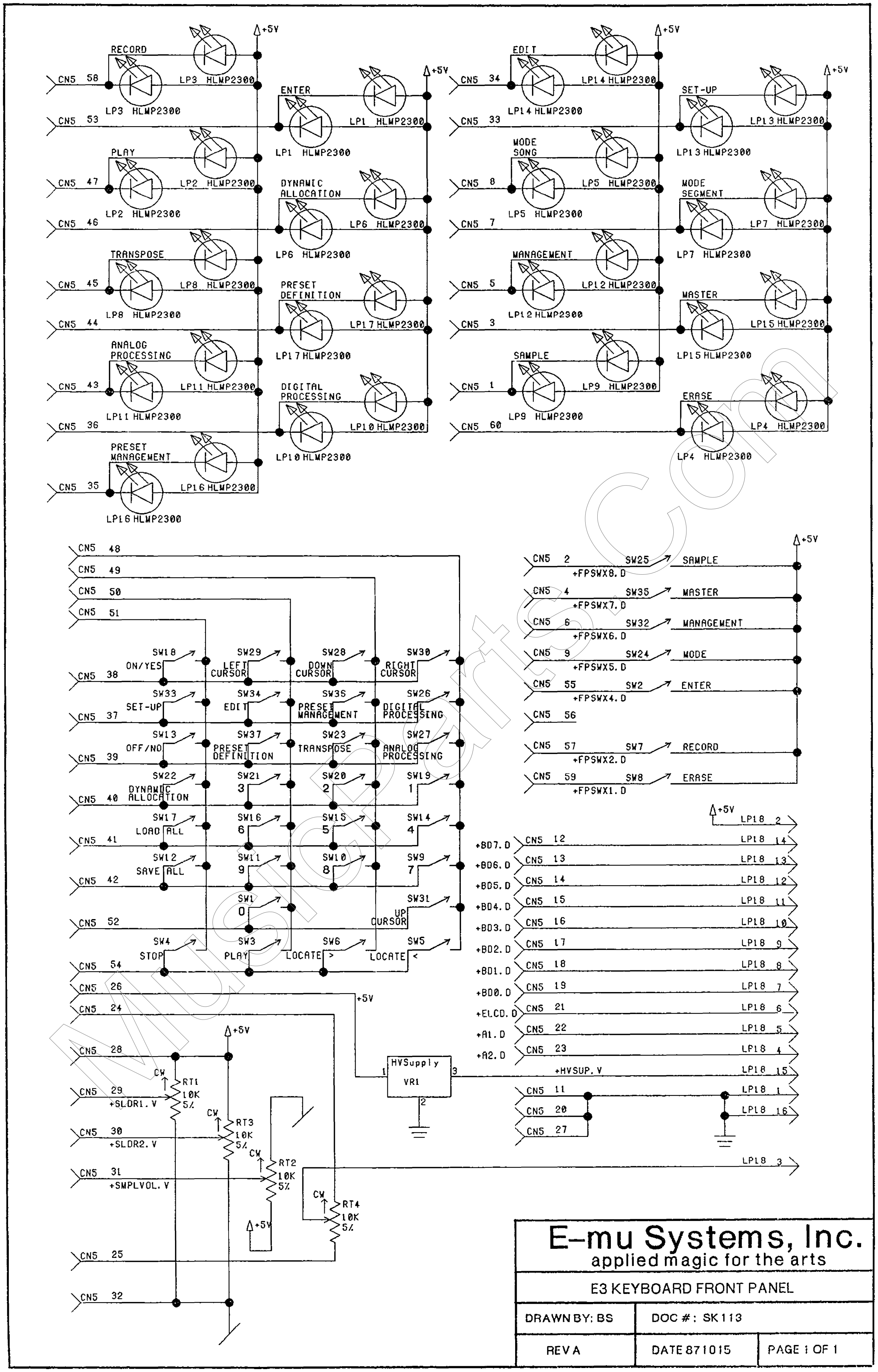


E–mu Systems, Inc.
applied magic for the arts

E3 RACK FRONT PANEL

DRAWN BY: BS	DOC #: SK111

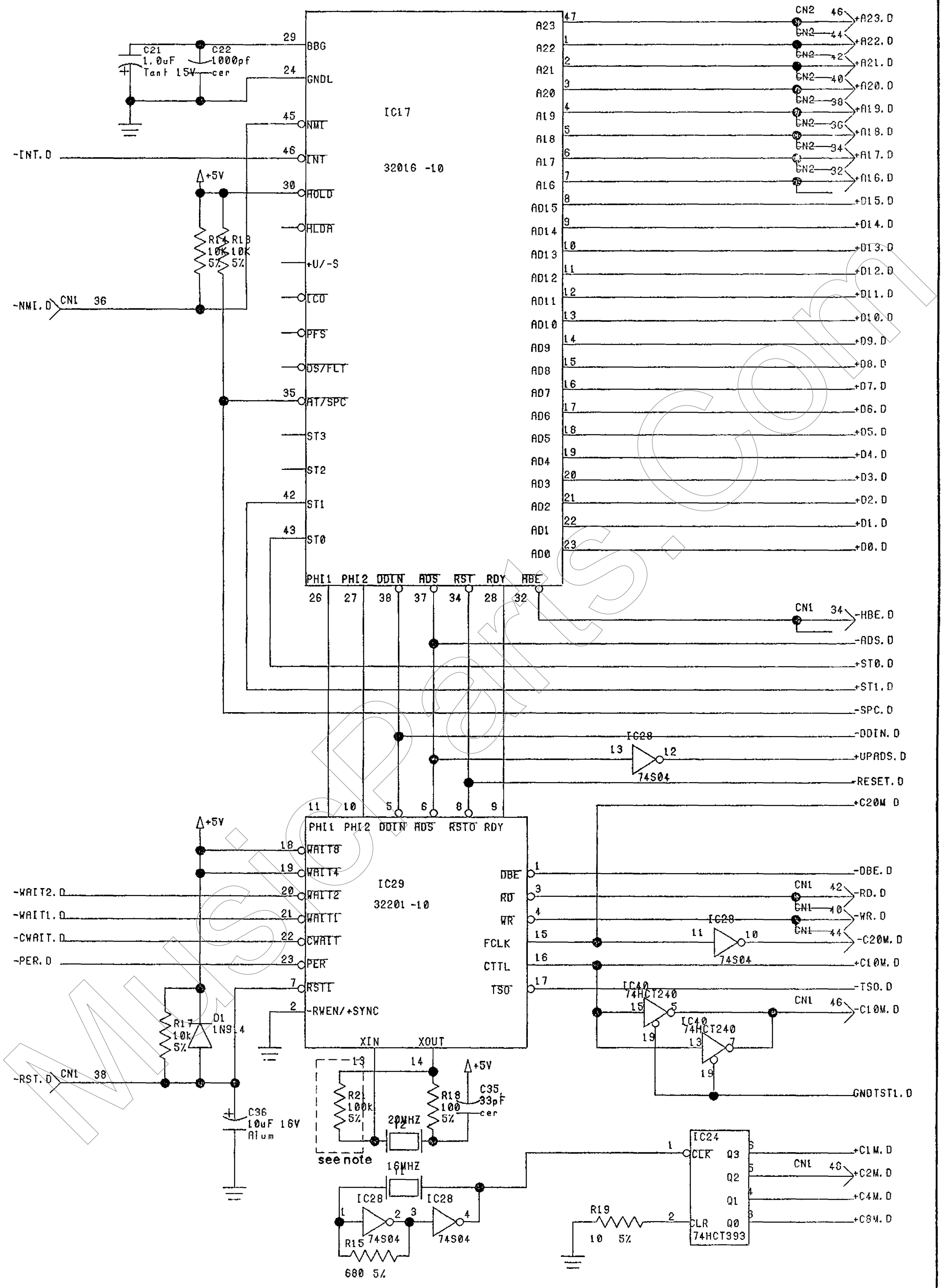
REV A	DATE 880323	PAGE 1 OF 1



E-mu Systems, Inc.
applied magic for the arts

E3 KEYBOARD FRONT PANEL

DRAWN BY: BS	DOC #: SK113
REV A	DATE 871015
	PAGE 1 OF 1



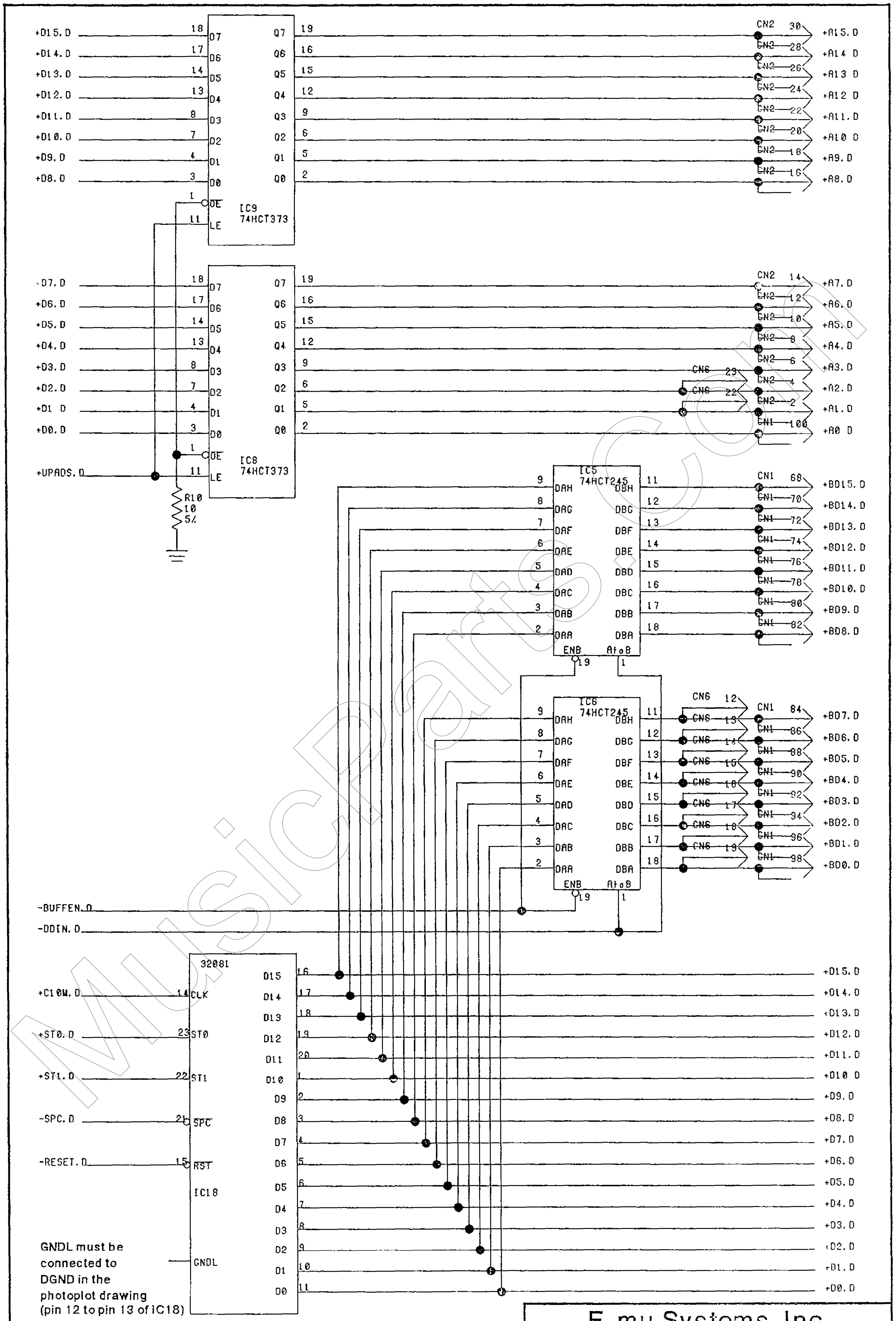
Note: R21 should only be loaded in board
if IC29 is a CMOS device (32C201).

E-mu Systems, Inc.

applied magic for the arts

E3 CPU2: MAIN PROCESSOR AND TCU

DRAWN BY: BFS	DOC.#: SK115
REV A	DATE 871208
	Page 1 of 10



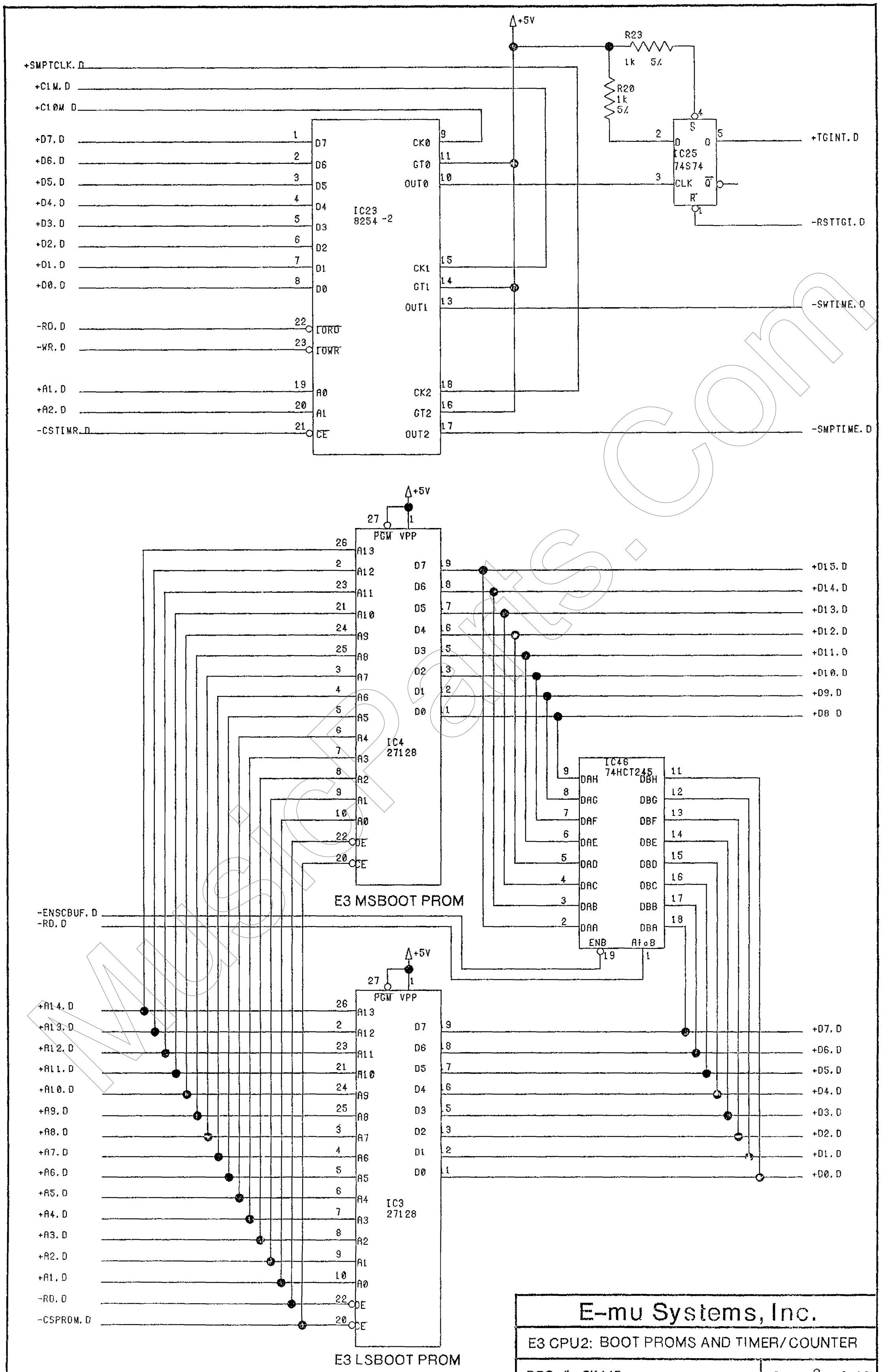
GNDL must be connected to DGND in the photoplot drawing (pin 12 to pin 13 of IC18)

E-mu Systems, Inc.

E3 CPU2ADDR REG;COPROCESSOR;BI-DIR BUFF

DOC. #: SK115

Page 2 of 10

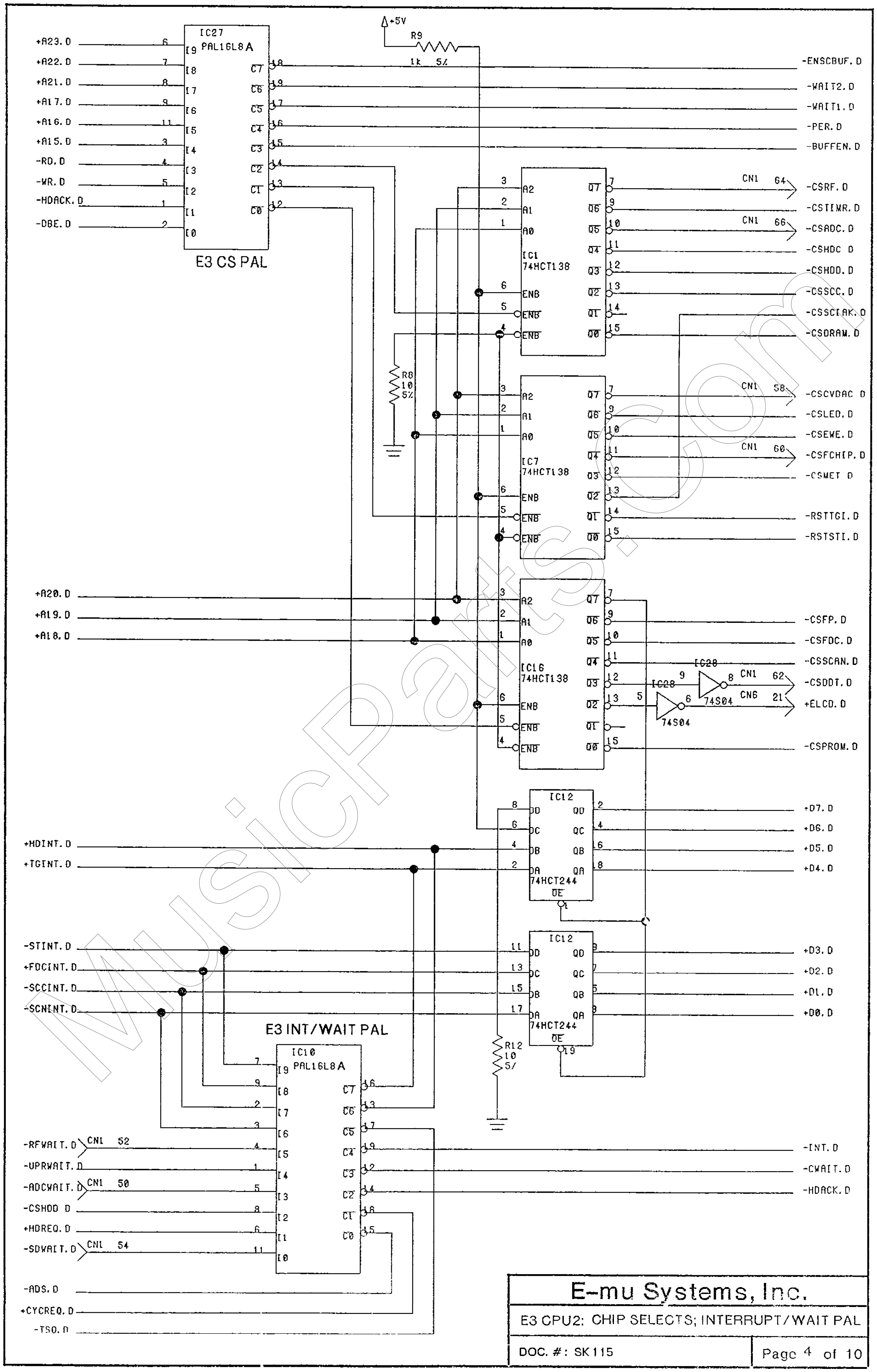


E-mu Systems, Inc.

E3 CPU2: BOOT PROMS AND TIMER/COUNTER

DOC. #: SK115

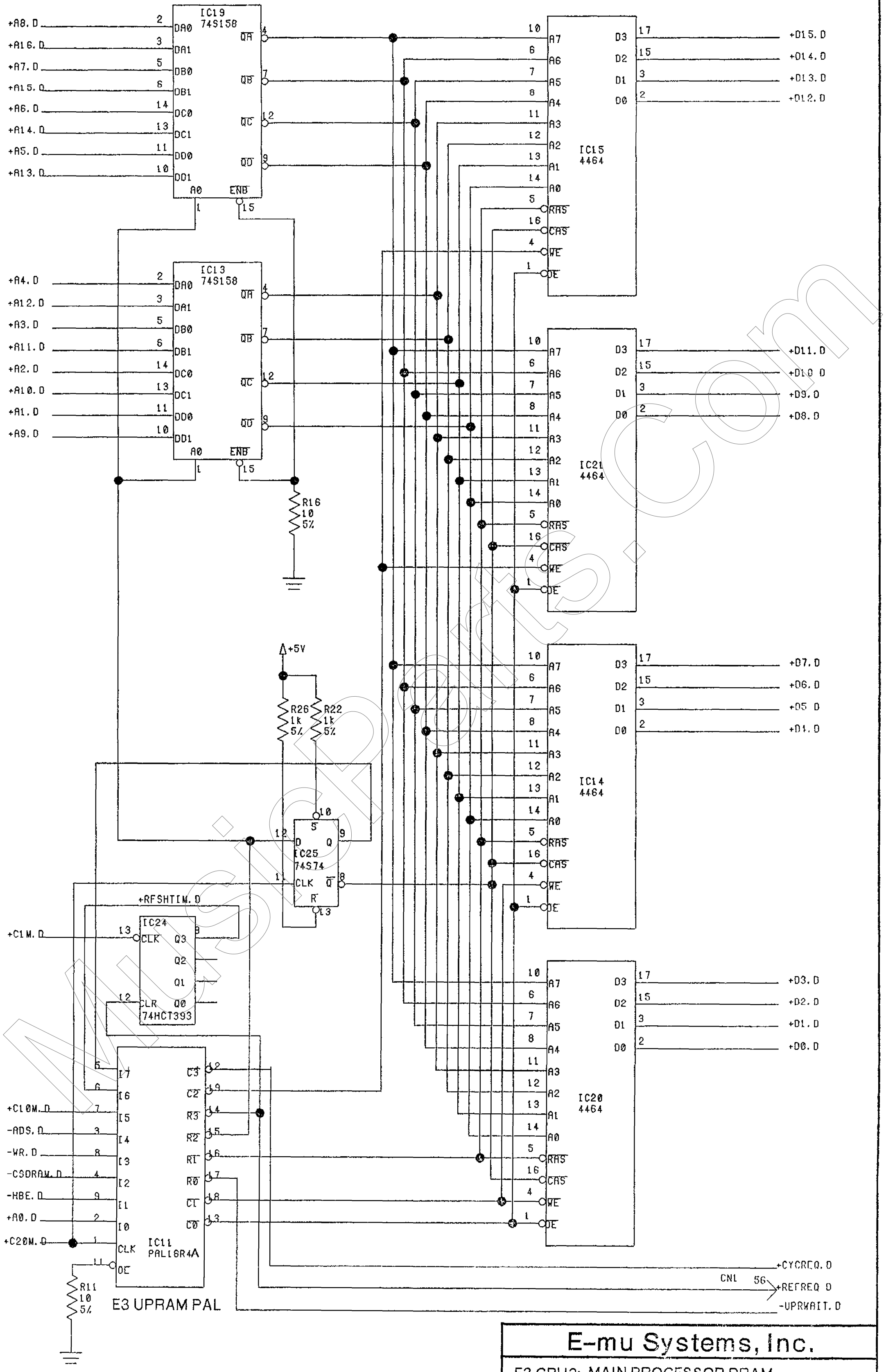
Page 3 of 10



E-mu Systems, Inc.

E3 CPU2: CHIP SELECTS; INTERRUPT/WAIT PAL

DOC. #: SK115

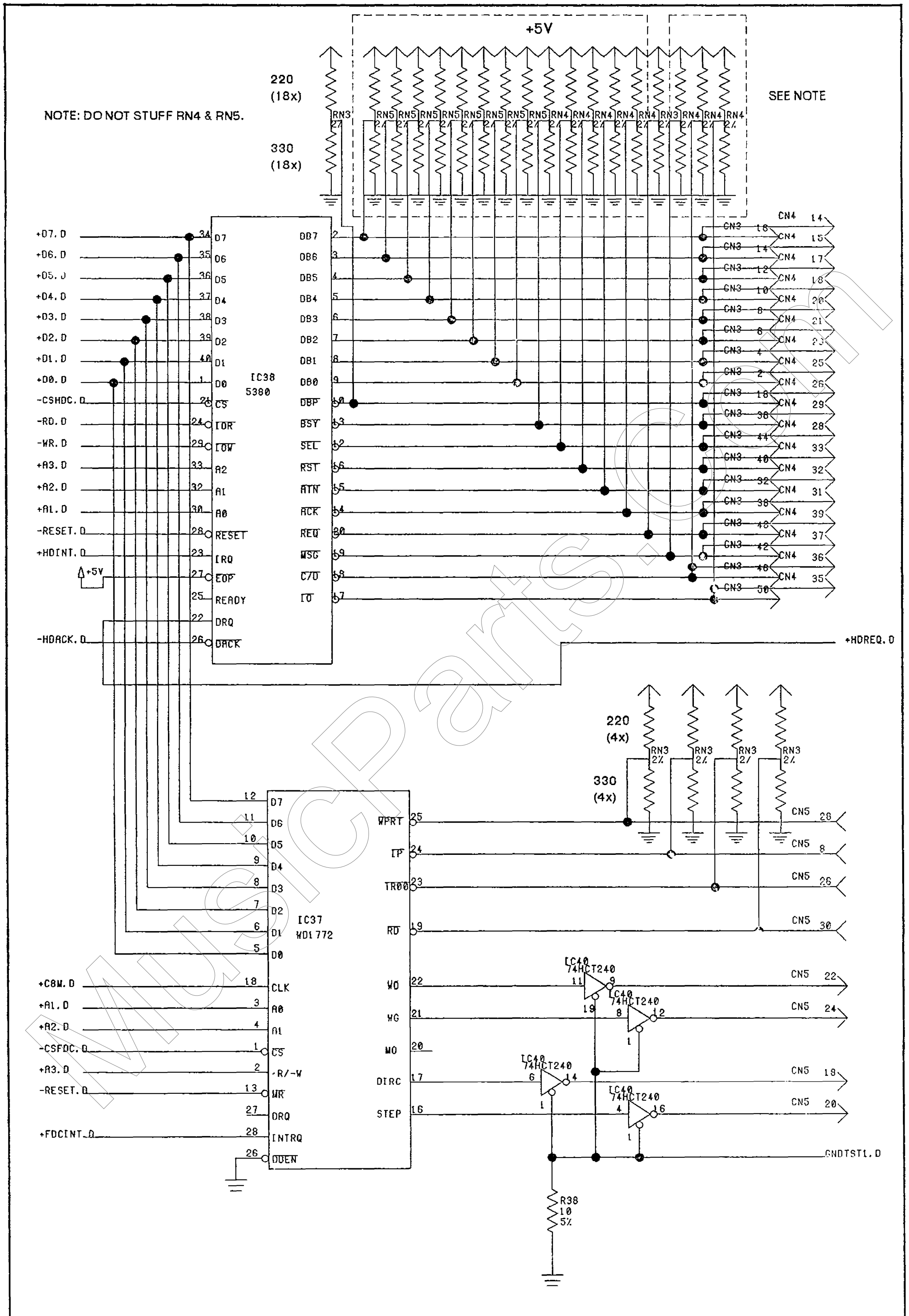


E-mu Systems, Inc.

E3 CPU2: MAIN PROCESSOR DRAM

DOC. #: SK115

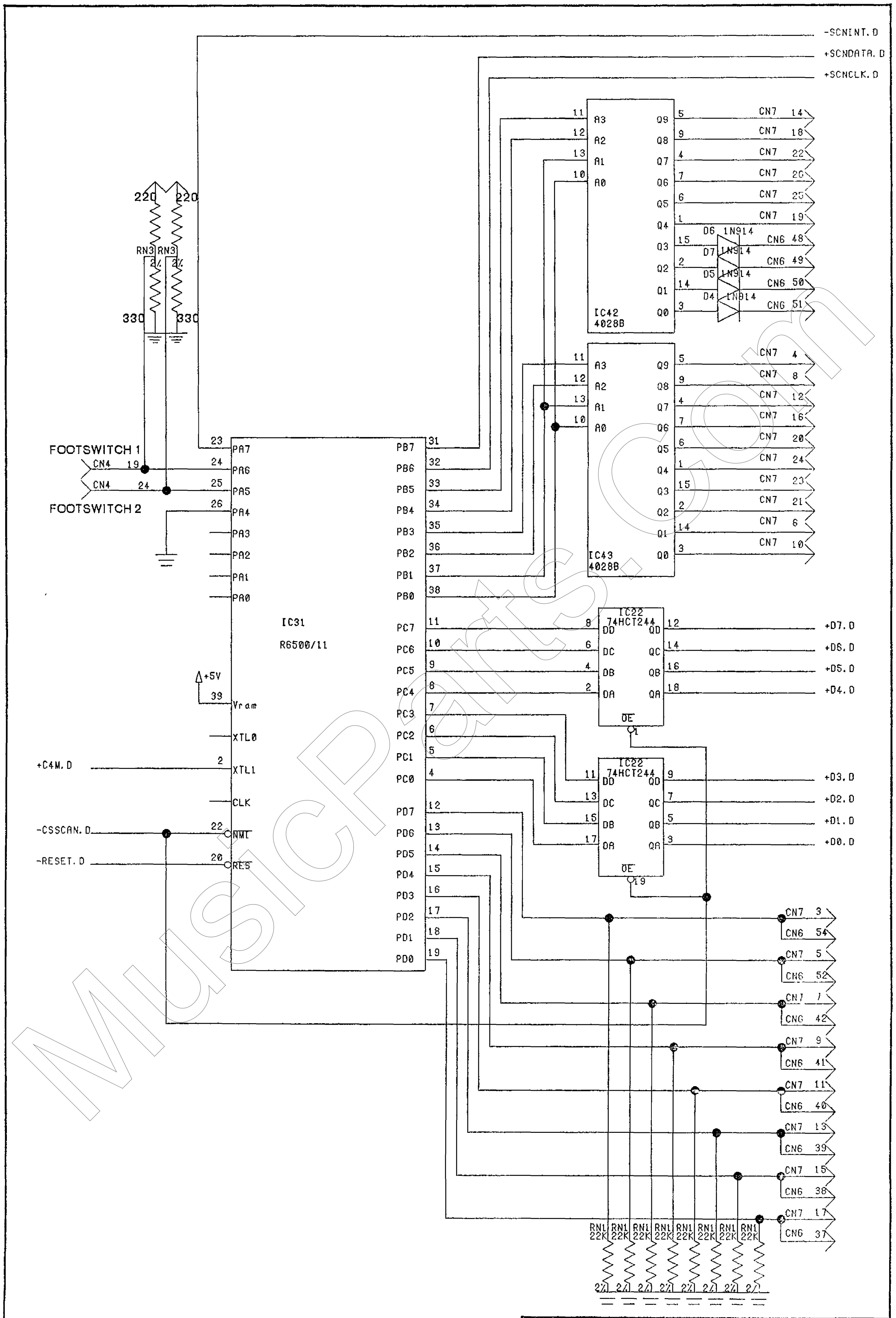
Page 5 of 10



E-mu Systems, Inc.

E3 CPU2: FLOPPY/HARD DISK/SCSI CHIPS

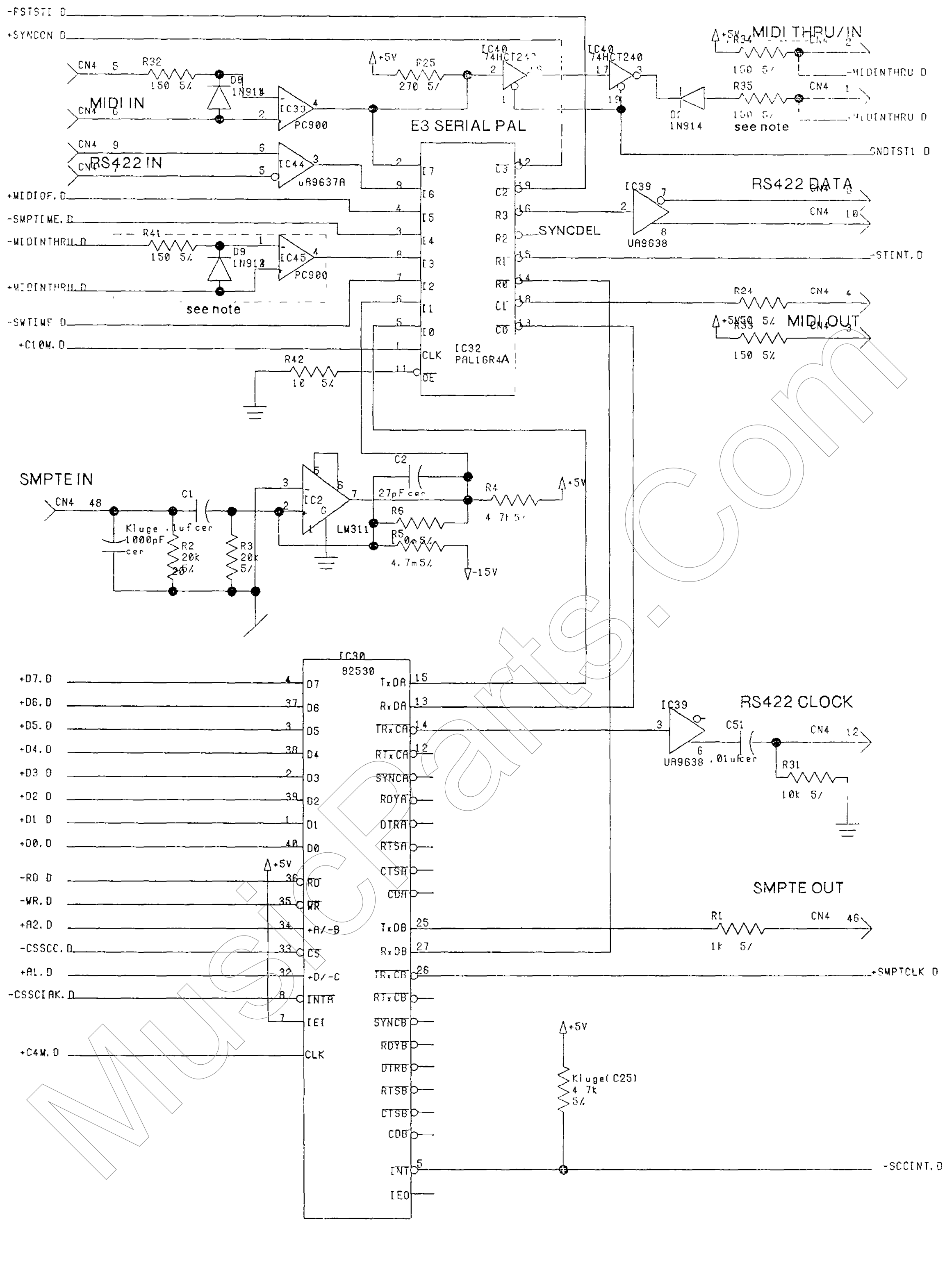
DOC. #: SK115 | Page 6 of 10



E-mu Systems, Inc.

E3 CPU2 Scanner and Keyboard Interface

DOC. #: SK115



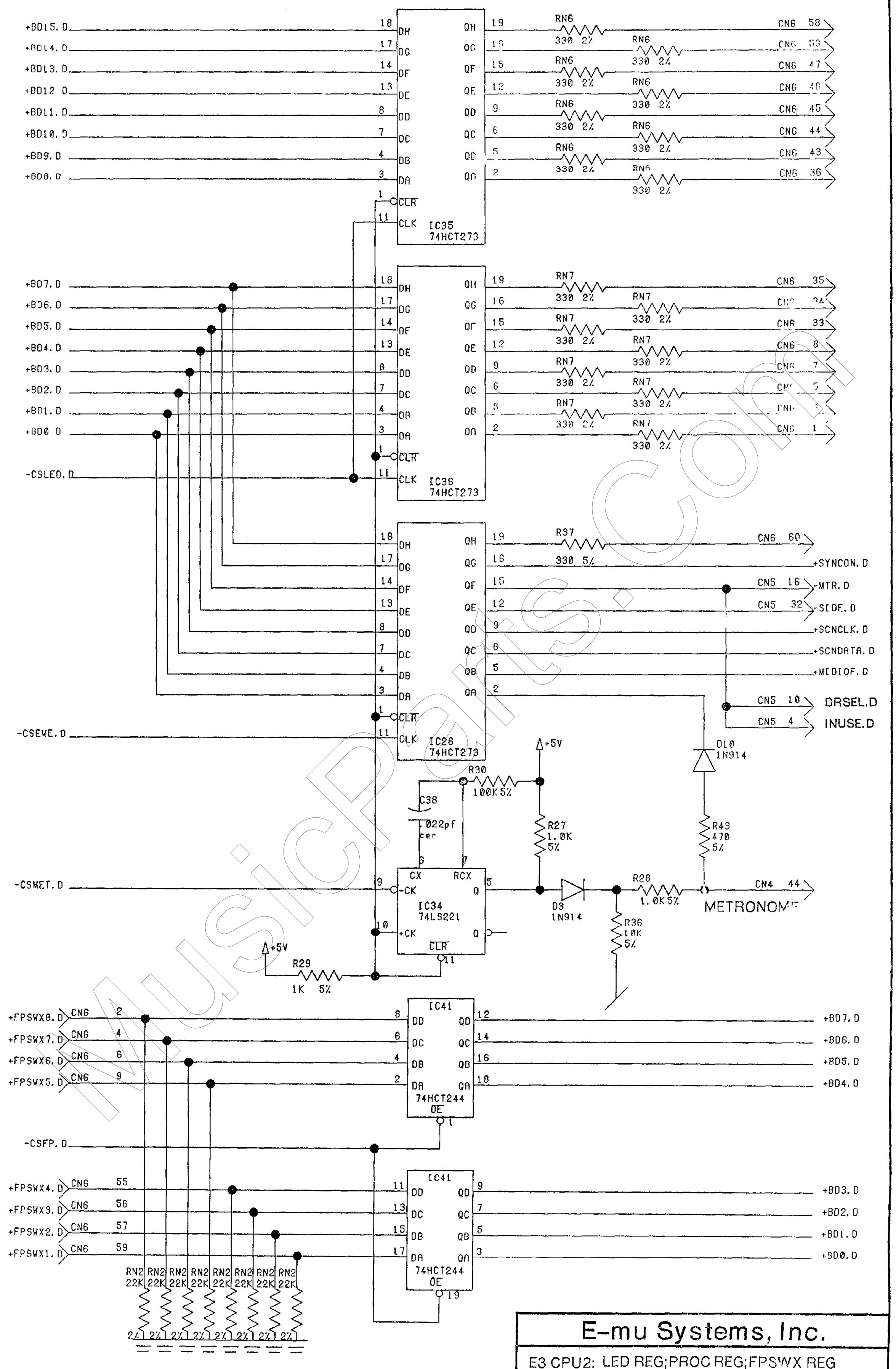
NOTE: Do NOT stuff locations R41, D9, and IC45 in non-Expander units.
For Expander units, stuff R41, D9, and IC45 and do NOT stuff R34 and R35.

E-mu Systems, Inc.

E3 CPU2 SMPTE,MIDI, and RS422 Interface

DOC #: SK115

Page 8 of 10

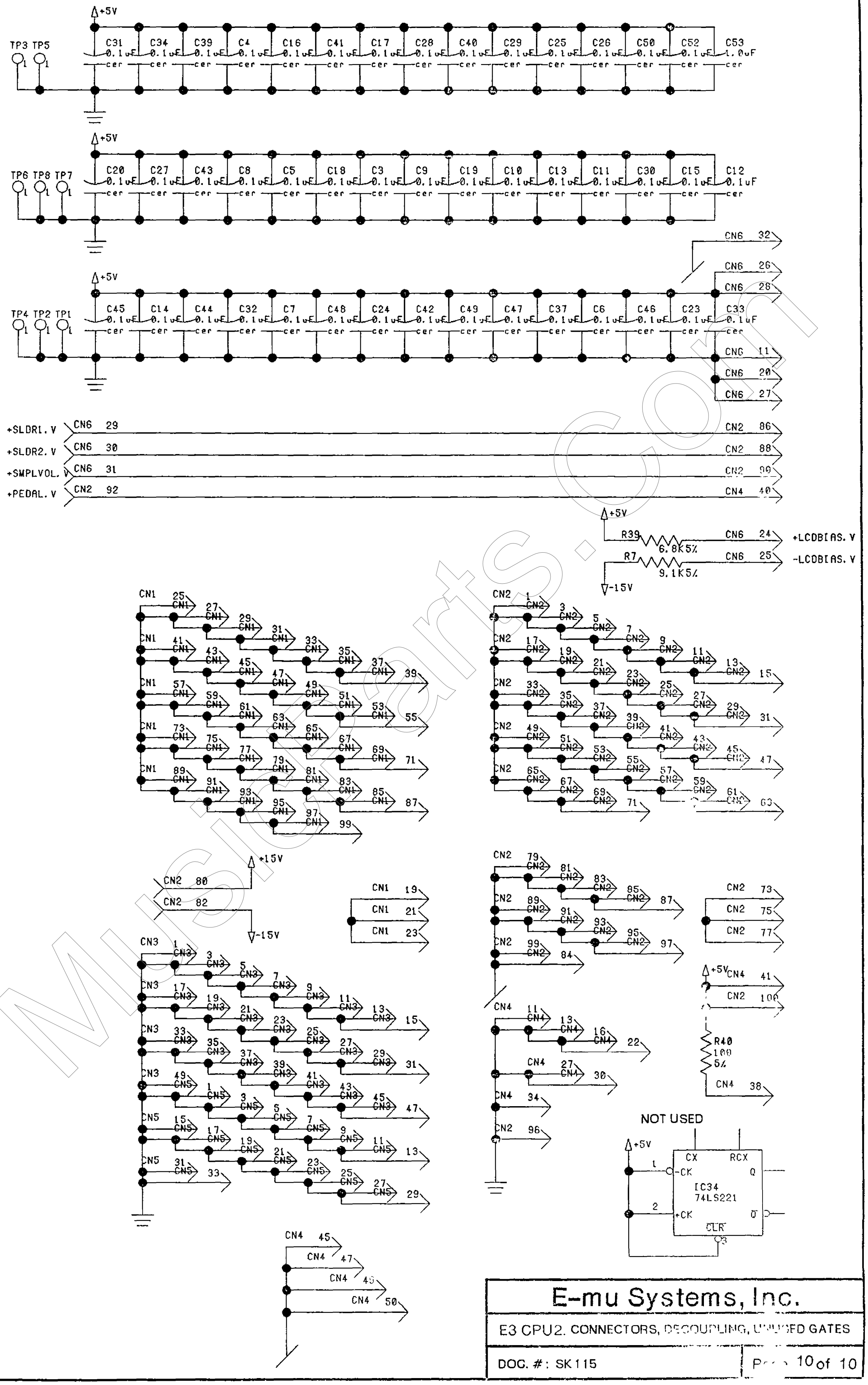


E-mu Systems, Inc.

E3 CPU2: LED REG;PROC REG;FPSWX REG

DOC. #: SK115

Page 9 of 10

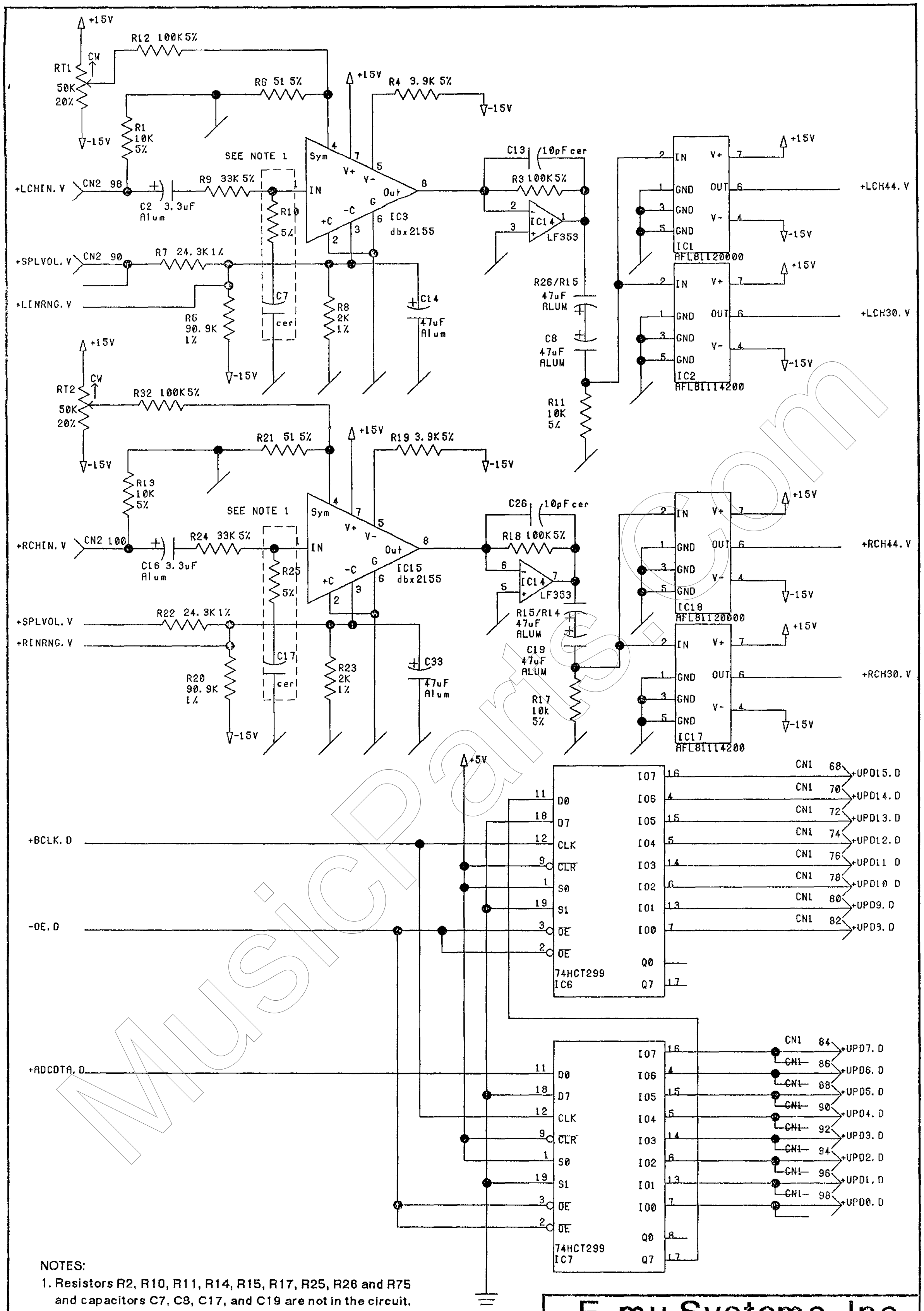


E-mu Systems, Inc.

E3 CPU2. CONNECTORS, DECOUPLING, UNUSED GATES

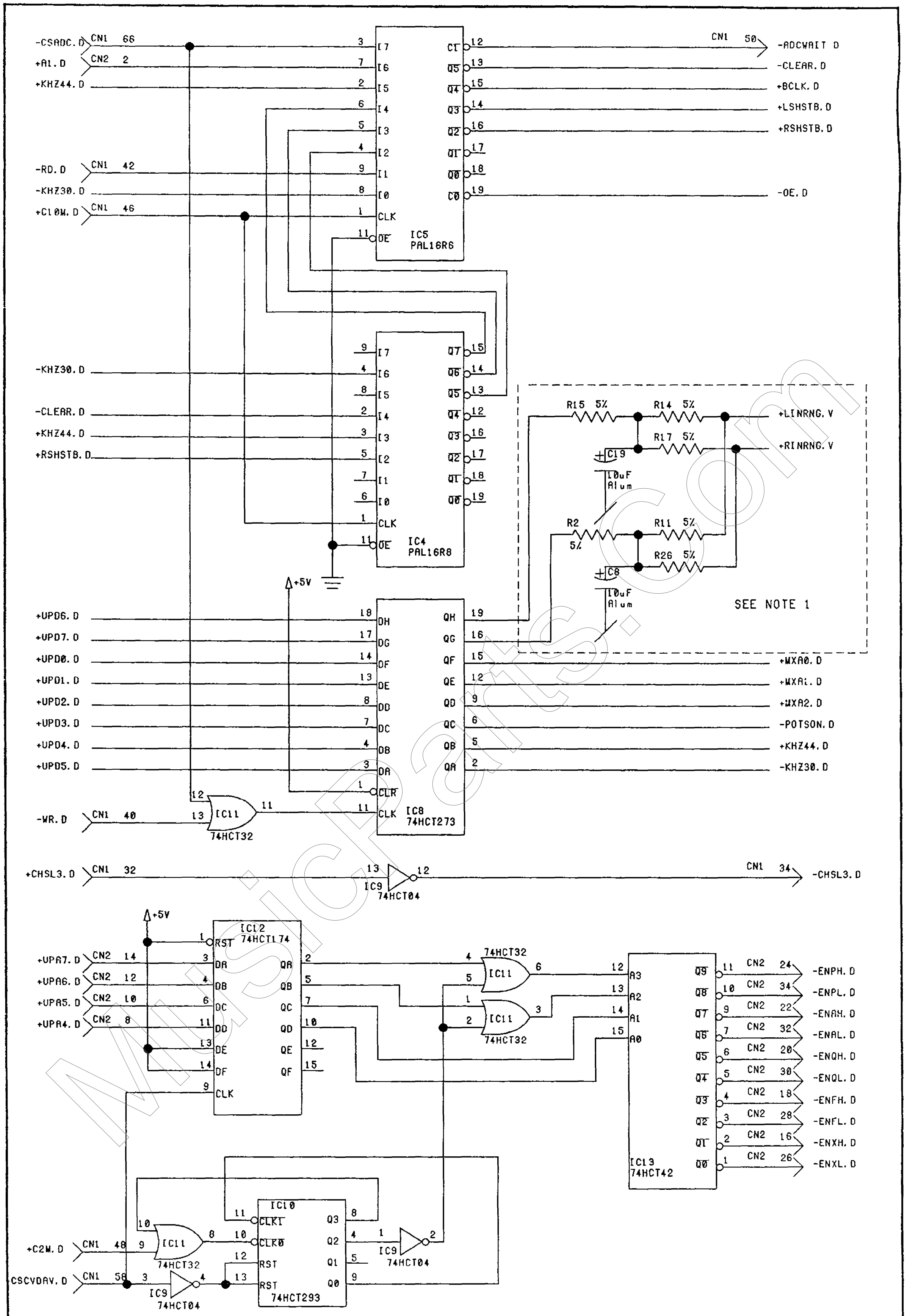
DOC. #: SK115

Page 10 of 10



NOTES:
1. Resistors R2, R10, R11, R14, R15, R17, R25, R26 and R75 and capacitors C7, C8, C17, and C19 are not in the circuit.

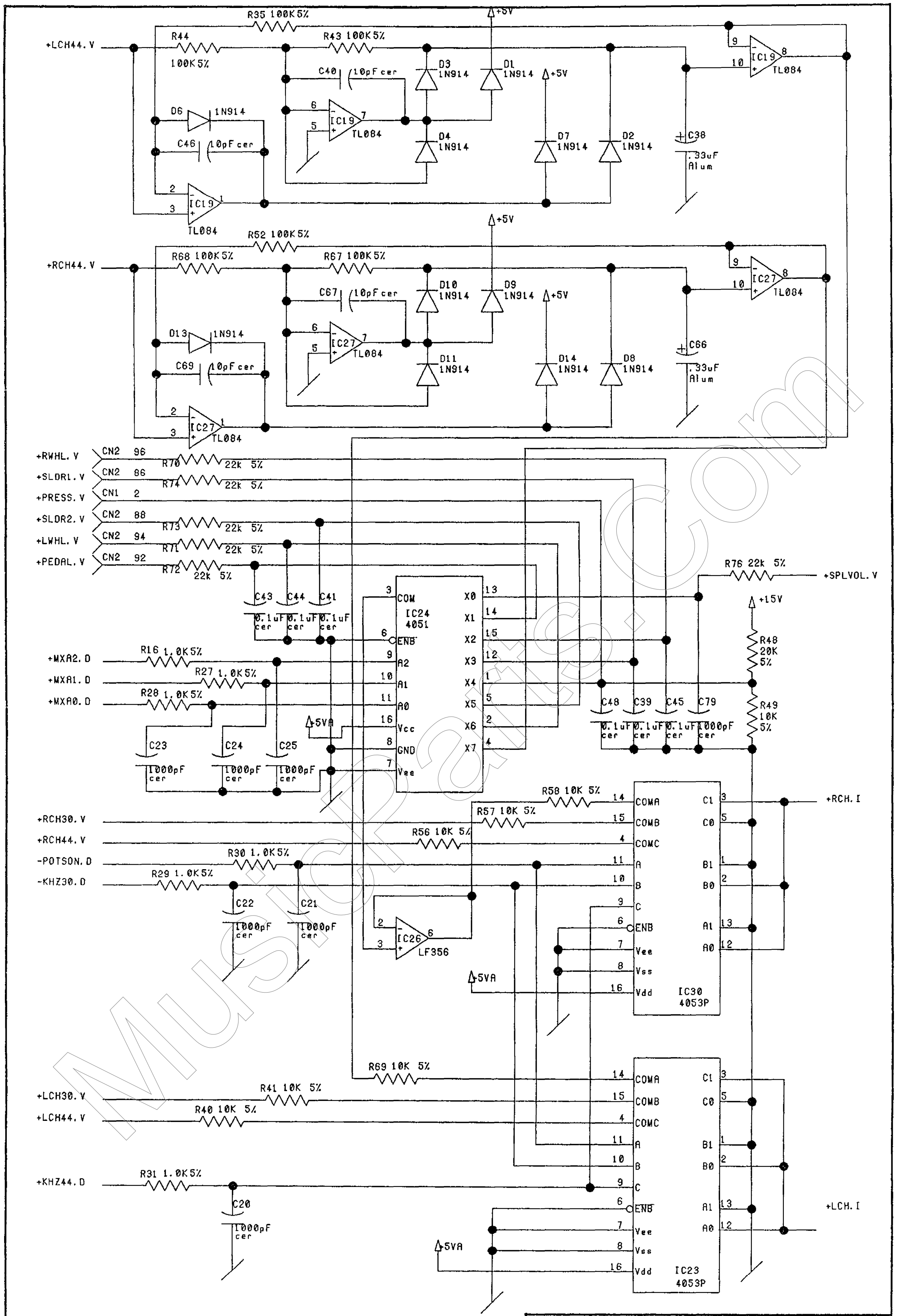
E–mu Systems, Inc. applied magic for the arts	
E3 ADC: Input Amps and Shift Register	
DRAWN BY: BFS	DOC.#: SK110
REV C	DATE 871111
	Page 1 of 5



E-mu Systems, Inc.

E3 ADC:Control Logic and Misc Channel Logic

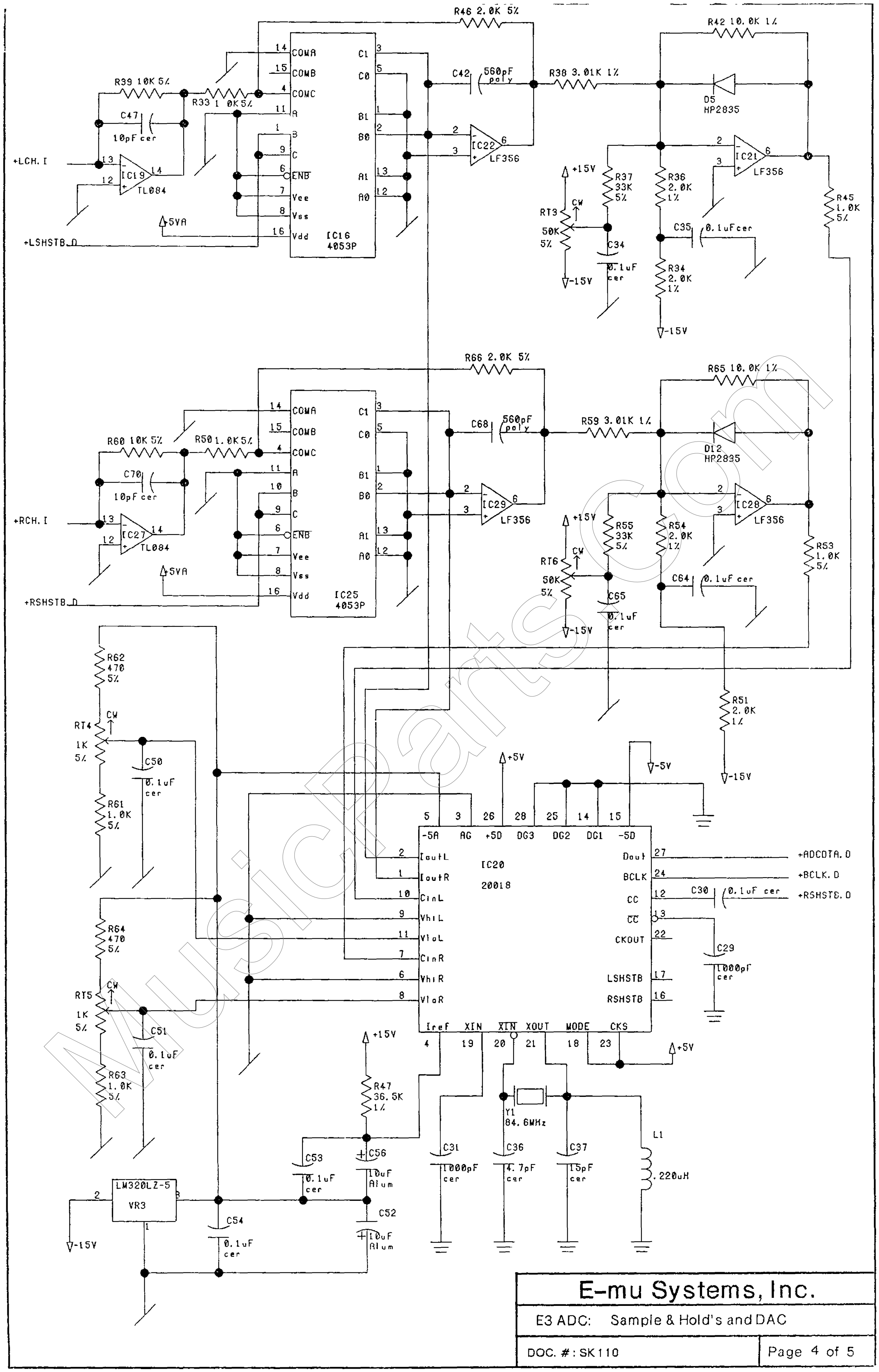
DOC. #: SK110



E-mu Systems, Inc.

E3 ADC: Envelope Followers and Signal Muxes

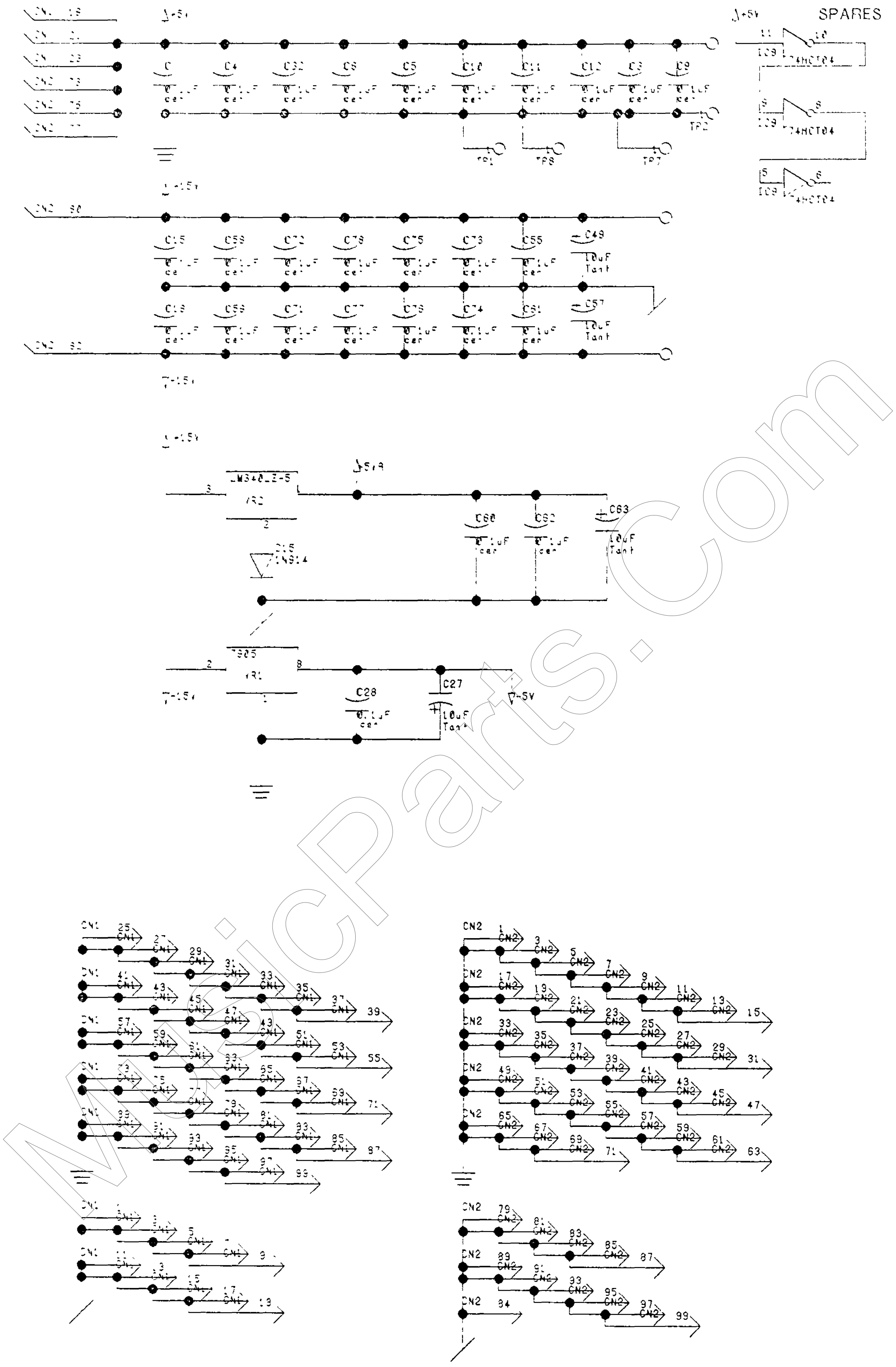
DOC. #: SK110



E-mu Systems, Inc.

E3 ADC: Sample & Hold's and DAC

DOC. #: SK110

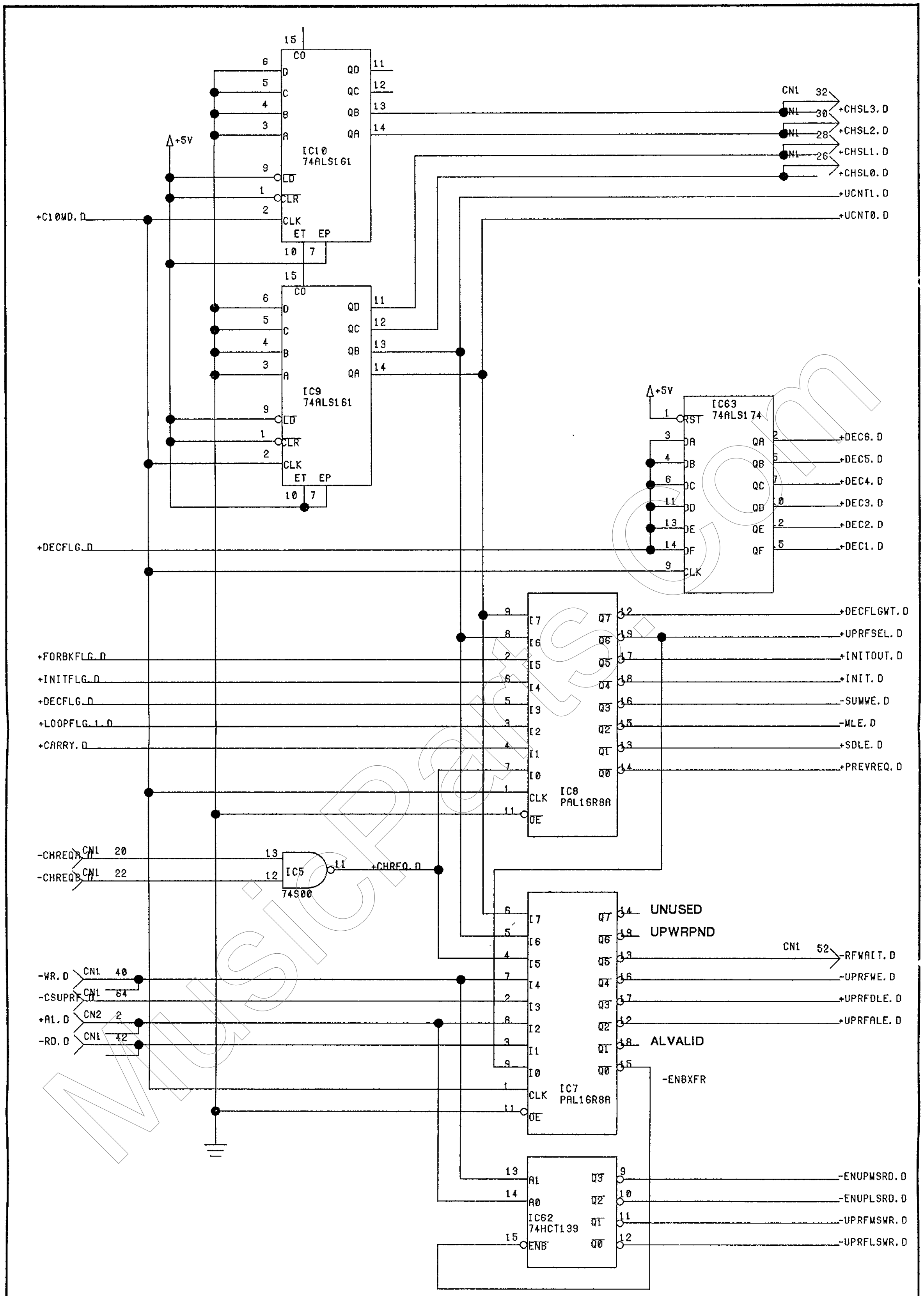


E-mu Systems, Inc.

E3 ADC: Power and Decouplers

DOC. #: SK110

Page 5 of 5

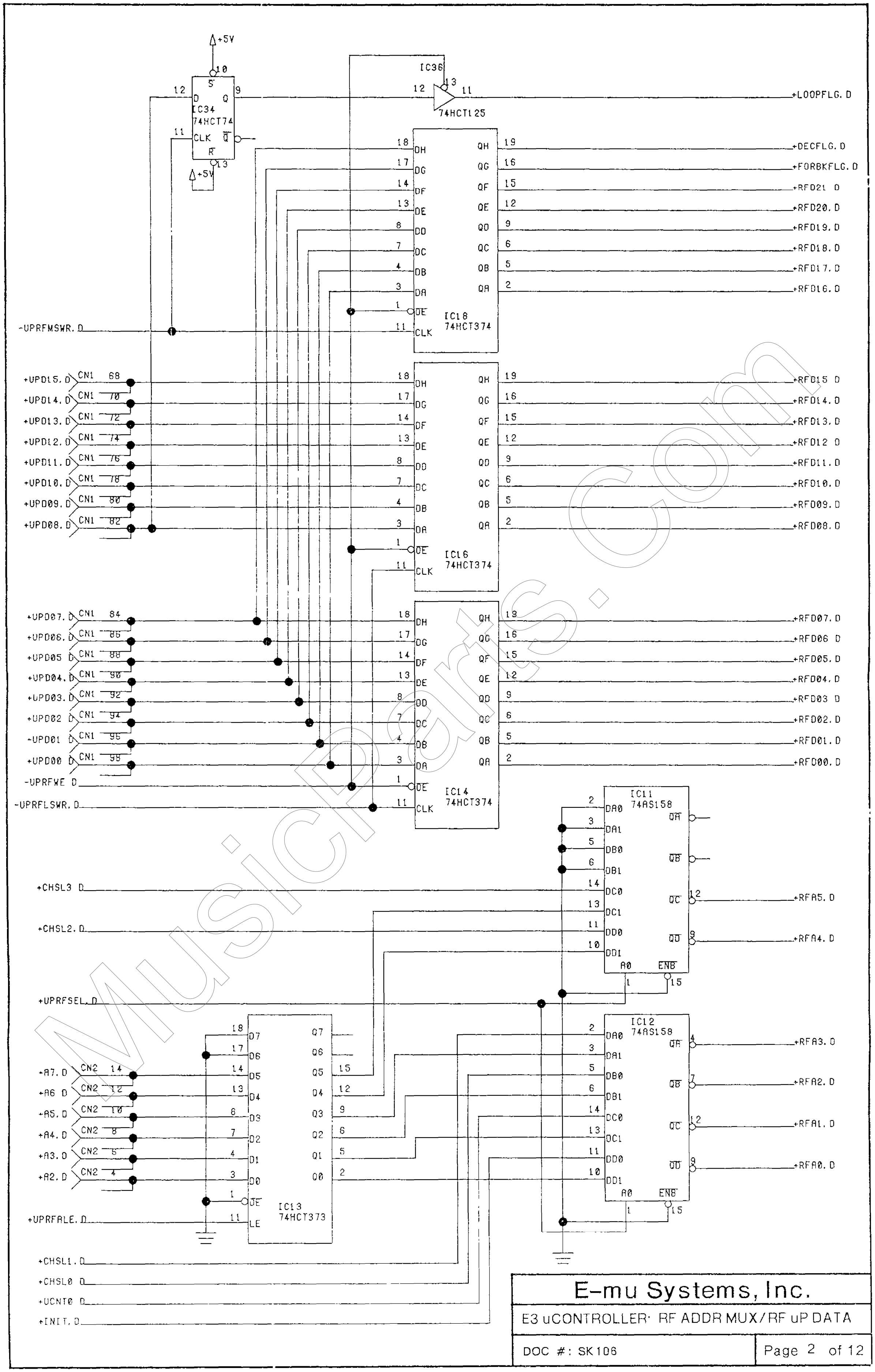


E–mu Systems, Inc.
applied magic for the arts

E3 uCONTROLLER: uCOUNTER AND PALS

DRAWN BY: BFS	DOC.#: SK108

REV A	DATE 871014	Page 1 of 12

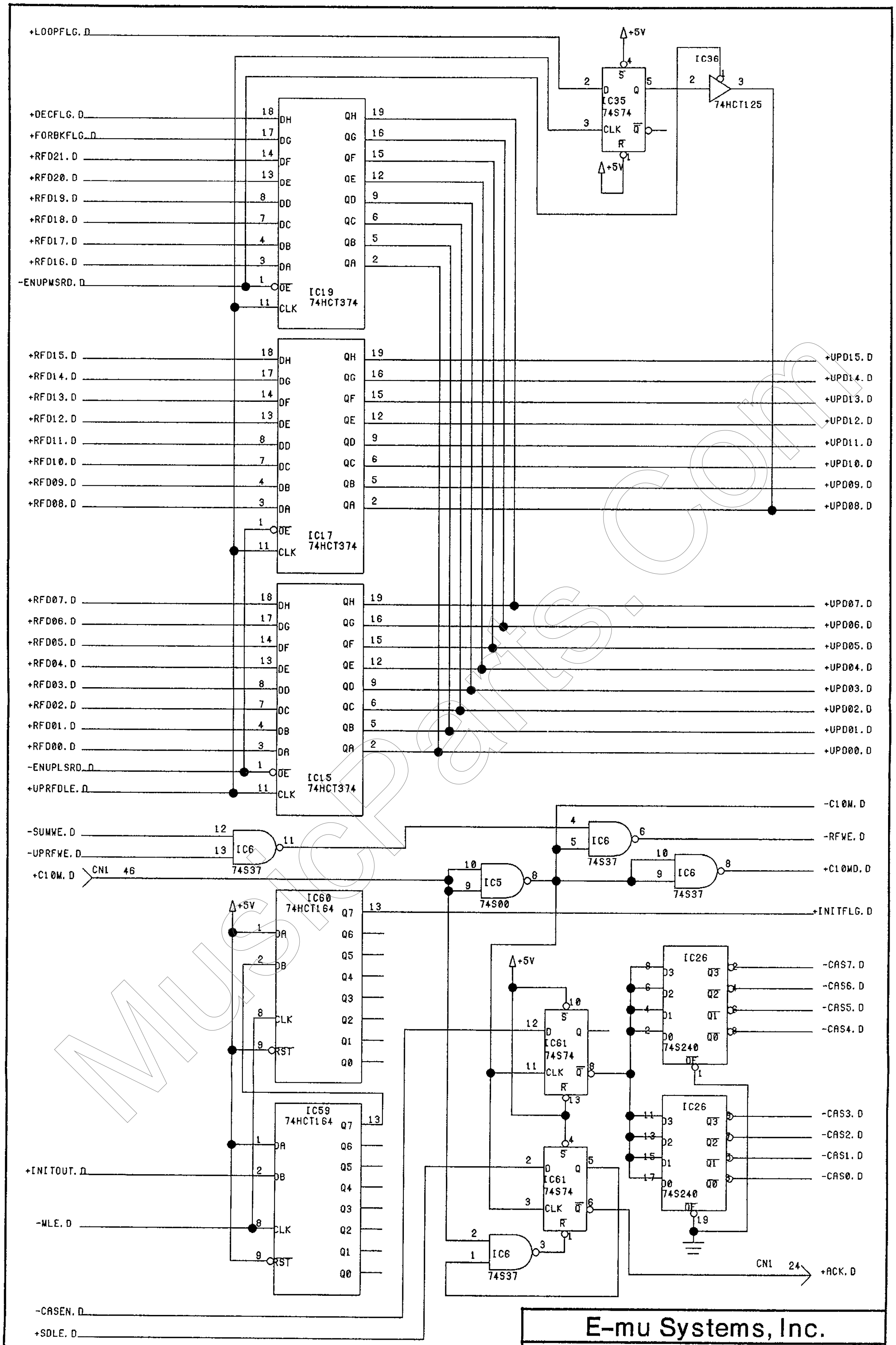


E–mu Systems, Inc.

E3 uCONTROLLER· RF ADDR MUX/RF uP DATA

DOC #: SK106

Page 2 of 12

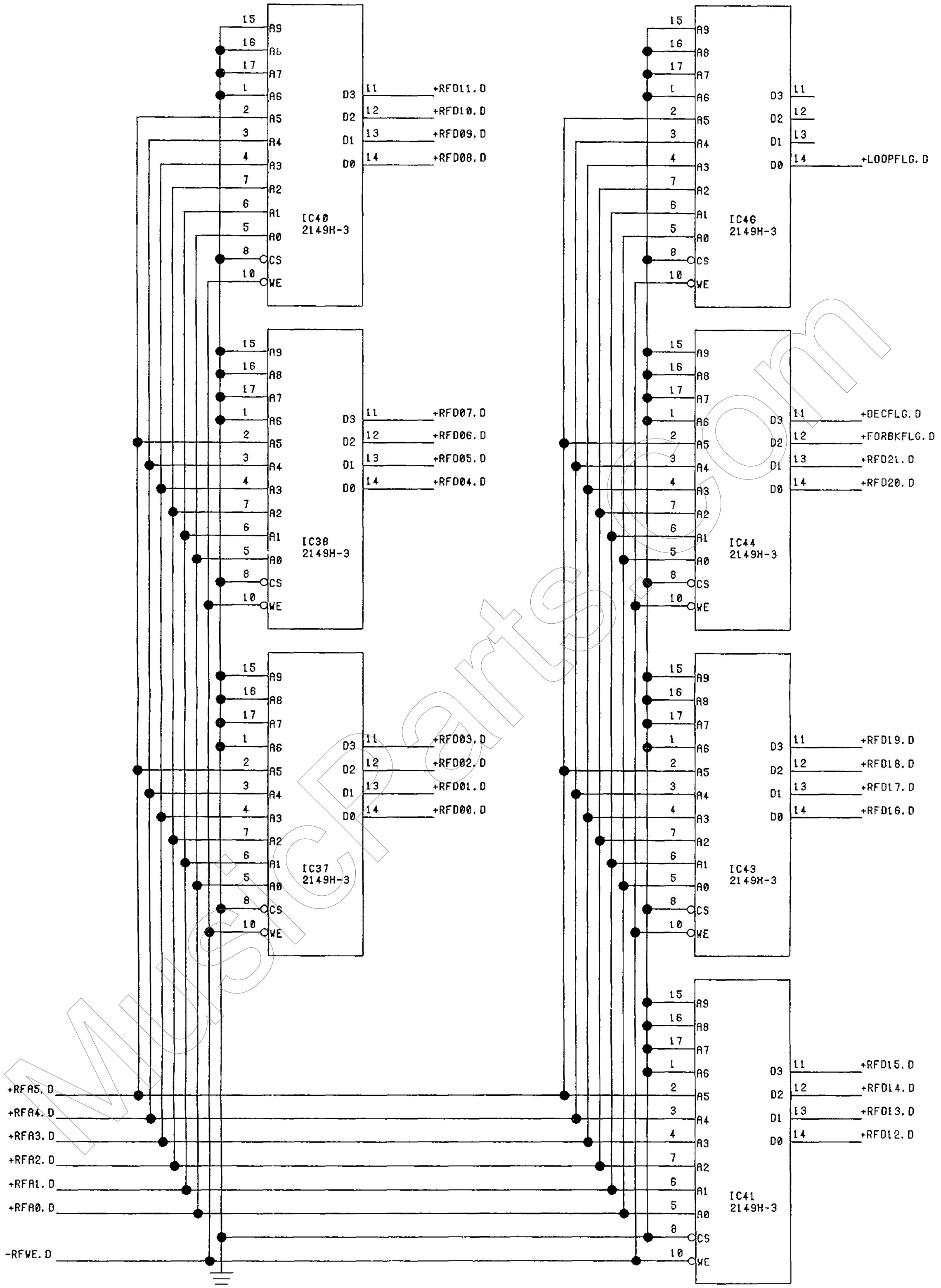


E–mu Systems, Inc.

E3 uCONTROLLER: CARRY REG/uP←RF DATA

DOC. #: SK106

Page 3 of 12

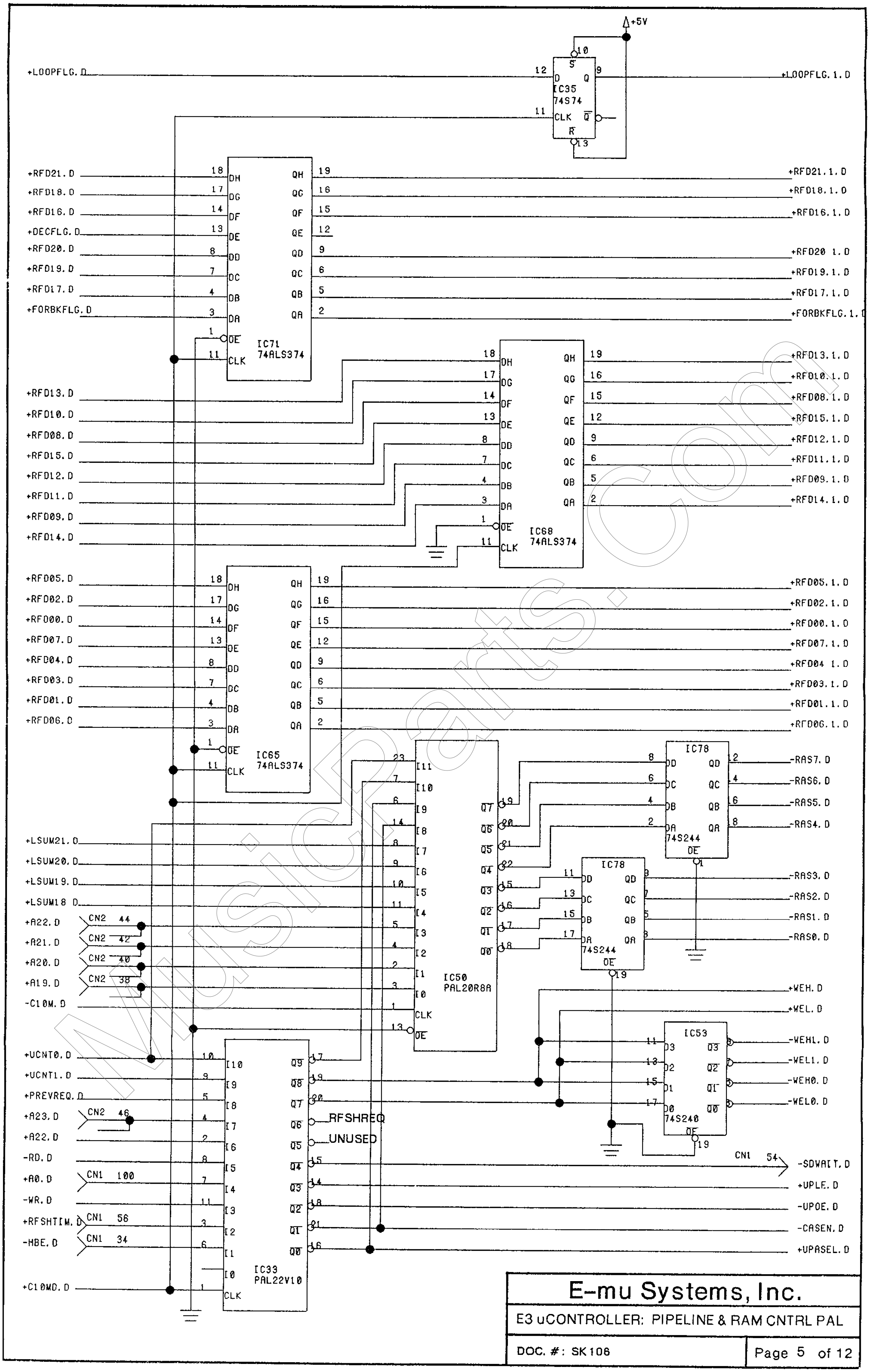


E-mu Systems, Inc.

E3 uCONTROLLER: REGISTER FILE

DOC. #: SK106

Page 4 of 12

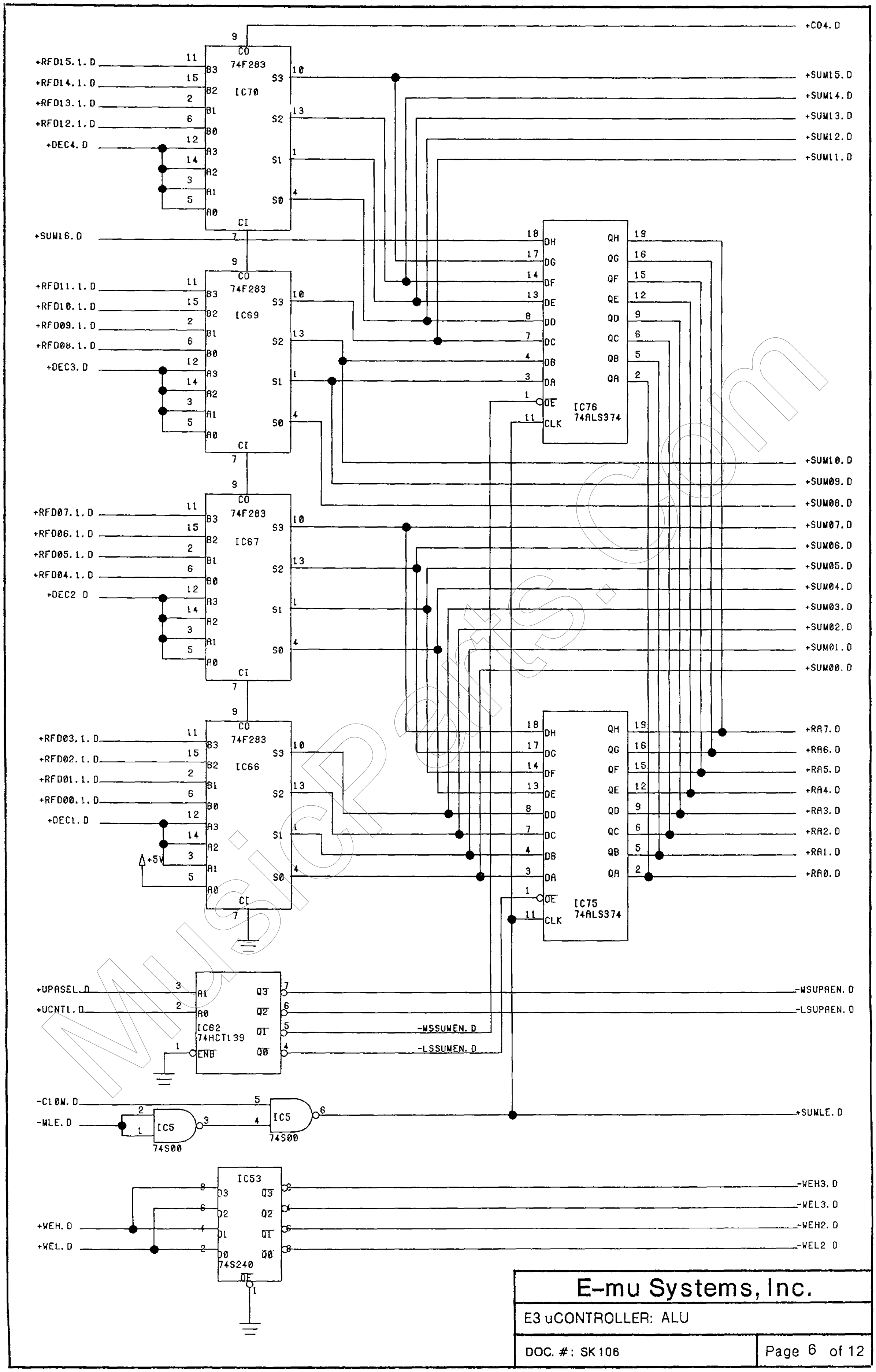


E-mu Systems, Inc.

E3 uCONTROLLER: PIPELINE & RAM CNTRL PAL

DOC. #: SK106

Page 5 of 12

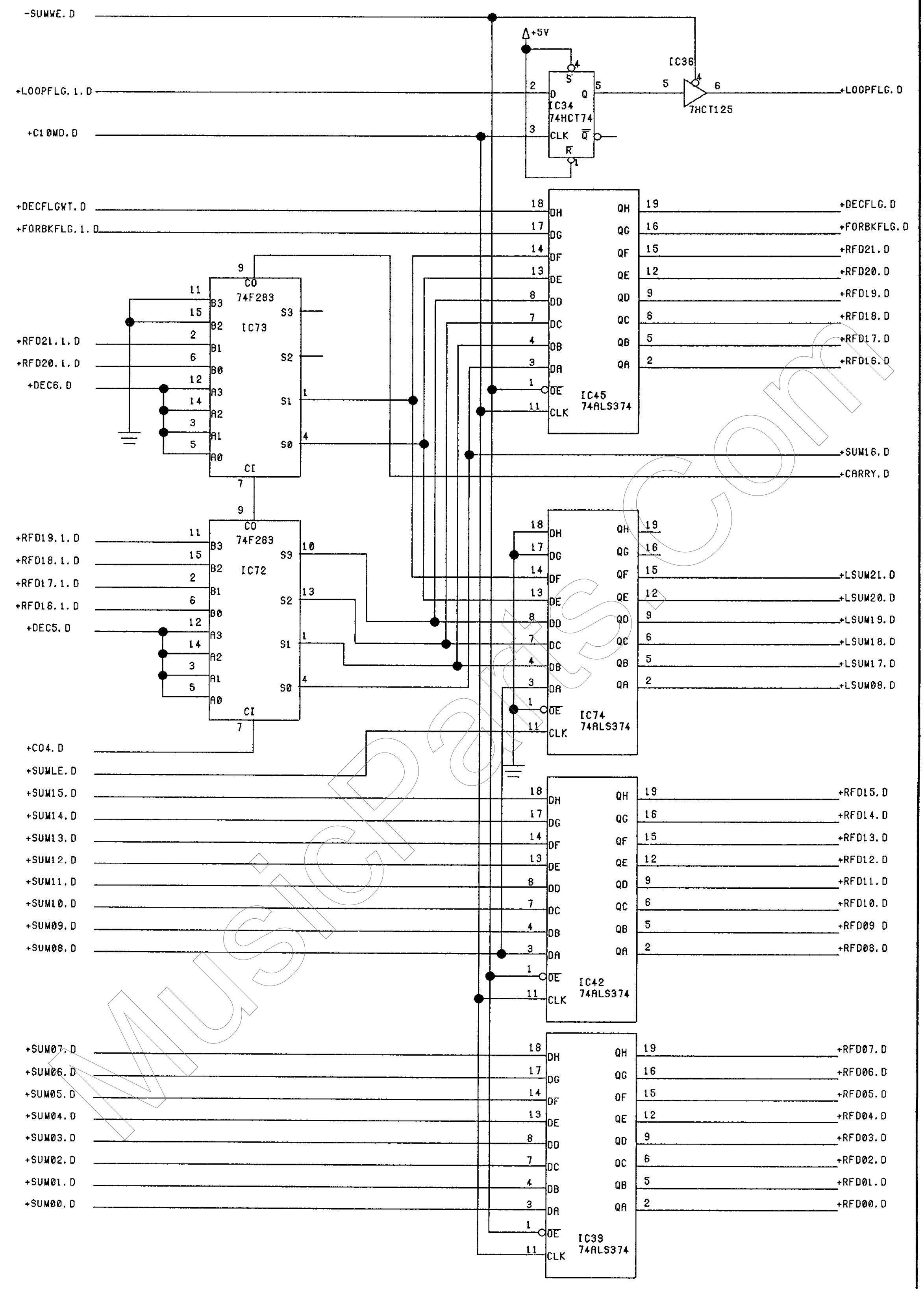


E–mu Systems, Inc.

E3 uCONTROLLER: ALU

DOC. #: SK106

Page 6 of 12

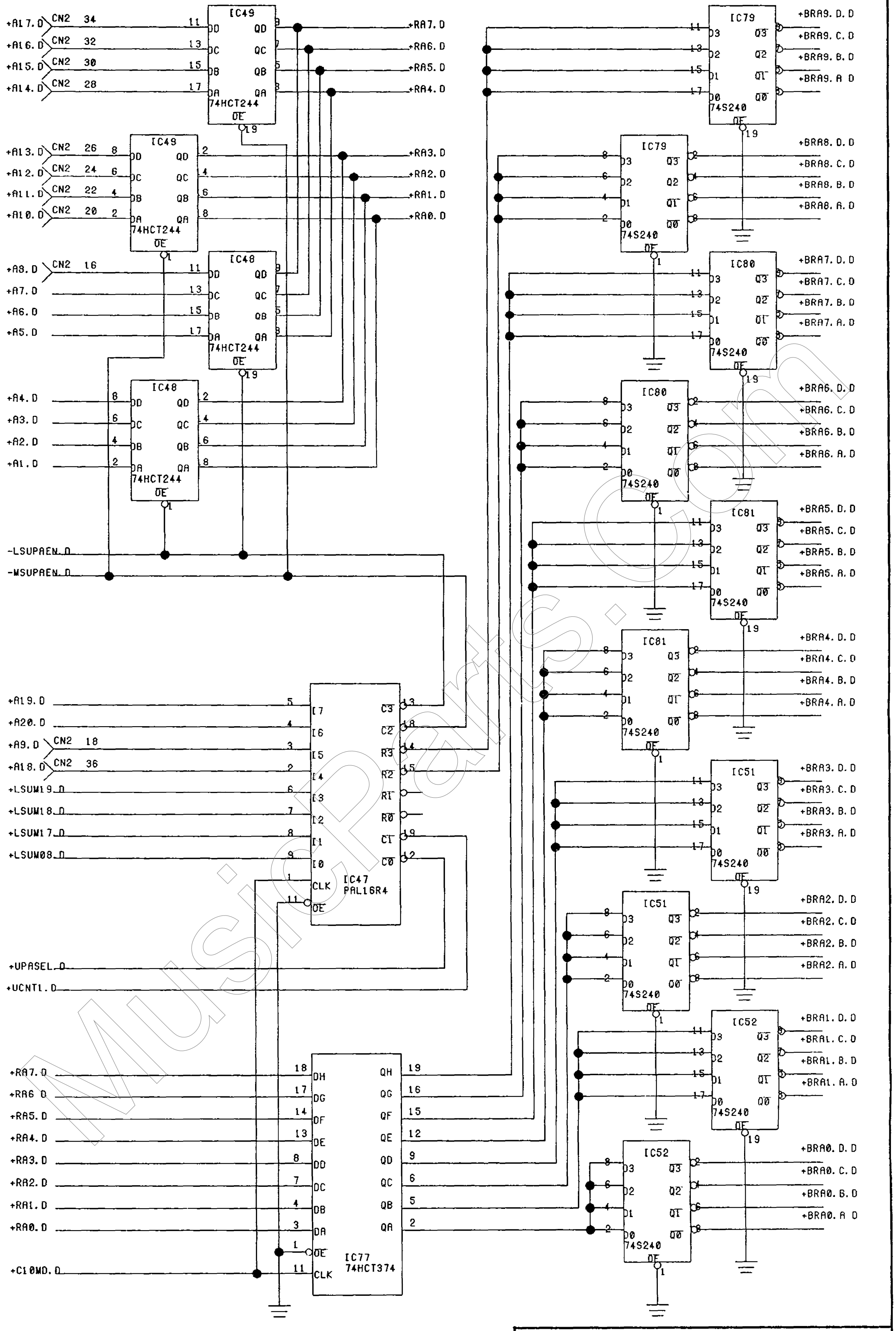


E-mu Systems, Inc.

E3 uCONTROLLER: ALU & SUM REGISTER

DOC. #: SK106

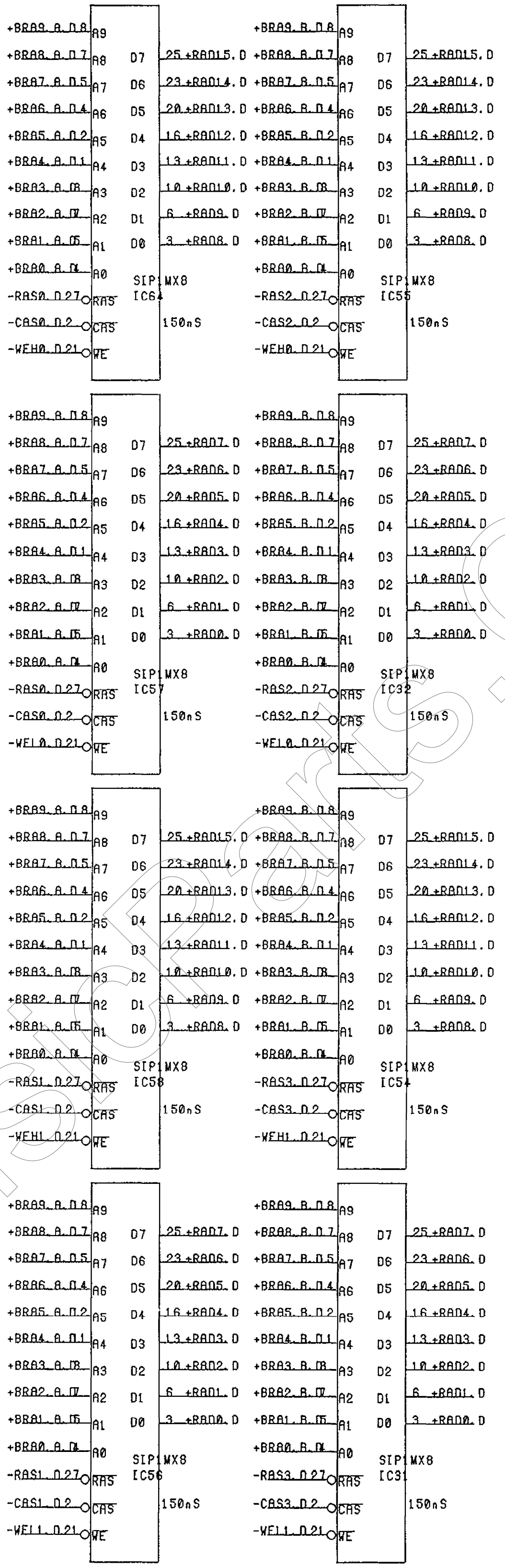
Page 7 of 12



E-mu Systems, Inc.

E3 uCONTROLLER: ADDRESS MUX AND DRIVERS

DOC. #: SK106



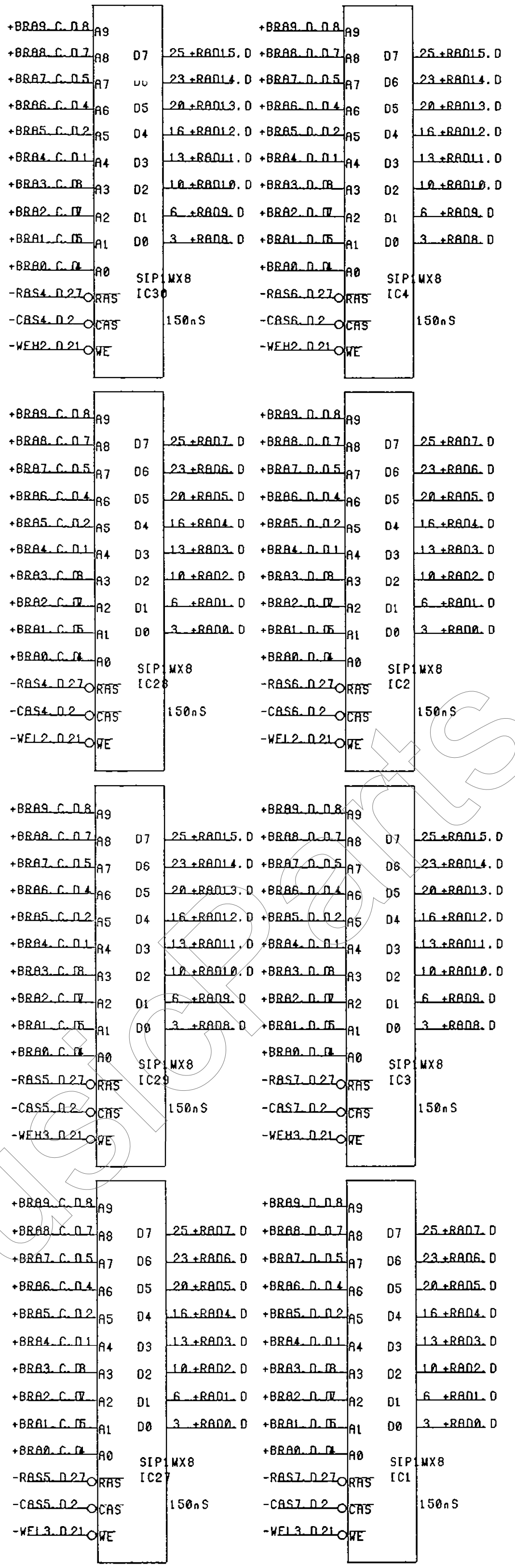
+BRA9.A.D8 A9
+BRA8.A.D7 A8 D7 25 +RAD15.D
+BRA7.A.D5 A7 D6 23 +RAD14.D
+BRA6.A.D4 A6 D5 20 +RAD13.D
+BRA5.A.D2 A5 D4 16 +RAD12.D
+BRA4.A.D1 A4 D3 13 +RAD11.D
+BRA3.A.D8 A3 D2 10 +RAD10.D
+BRA2.A.D7 A2 D1 6 +RAD9.D
+BRA1.A.D5 A1 D0 3 +RAD8.D
+BRA0.A.D4 A0
SIP1MX8
IC64
-RAS0.D27 RAS
-CAS0.D2 CAS
150nS
-WEH0.D21 WE
+BRA9.B.D8 A9
+BRA8.B.D7 A8 D7 25 +RAD15.D
+BRA7.B.D5 A7 D6 23 +RAD14.D
+BRA6.B.D4 A6 D5 20 +RAD13.D
+BRA5.B.D2 A5 D4 16 +RAD12.D
+BRA4.B.D1 A4 D3 13 +RAD11.D
+BRA3.B.D8 A3 D2 10 +RAD10.D
+BRA2.B.D7 A2 D1 6 +RAD9.D
+BRA1.B.D5 A1 D0 3 +RAD8.D
+BRA0.B.D4 A0
SIP1MX8
IC55
-RAS2.D27 RAS
-CAS2.D2 CAS
150nS
-WEH0.D21 WE
+BRA9.A.D8 A9
+BRA8.A.D7 A8 D7 25 +RAD7.D
+BRA7.A.D5 A7 D6 23 +RAD6.D
+BRA6.A.D4 A6 D5 20 +RAD5.D
+BRA5.A.D2 A5 D4 16 +RAD4.D
+BRA4.A.D1 A4 D3 13 +RAD3.D
+BRA3.A.D8 A3 D2 10 +RAD2.D
+BRA2.A.D7 A2 D1 6 +RAD1.D
+BRA1.A.D5 A1 D0 3 +RAD0.D
+BRA0.A.D4 A0
SIP1MX8
IC57
-RAS0.D27 RAS
-CAS0.D2 CAS
150nS
-WEL0.D21 WE
+BRA9.B.D8 A9
+BRA8.B.D7 A8 D7 25 +RAD7.D
+BRA7.B.D5 A7 D6 23 +RAD6.D
+BRA6.B.D4 A6 D5 20 +RAD5.D
+BRA5.B.D2 A5 D4 16 +RAD4.D
+BRA4.B.D1 A4 D3 13 +RAD3.D
+BRA3.B.D8 A3 D2 10 +RAD2.D
+BRA2.B.D7 A2 D1 6 +RAD1.D
+BRA1.B.D5 A1 D0 3 +RAD0.D
+BRA0.B.D4 A0
SIP1MX8
IC32
-RAS2.D27 RAS
-CAS2.D2 CAS
150nS
-WEL0.D21 WE
+BRA9.A.D8 A9
+BRA8.A.D7 A8 D7 25 +RAD15.D
+BRA7.A.D5 A7 D6 23 +RAD14.D
+BRA6.A.D4 A6 D5 20 +RAD13.D
+BRA5.A.D2 A5 D4 16 +RAD12.D
+BRA4.A.D1 A4 D3 13 +RAD11.D
+BRA3.A.D8 A3 D2 10 +RAD10.D
+BRA2.A.D7 A2 D1 6 +RAD9.D
+BRA1.A.D5 A1 D0 3 +RAD8.D
+BRA0.A.D4 A0
SIP1MX8
IC58
-RAS1.D27 RAS
-CAS1.D2 CAS
150nS
-WEH1.D21 WE
+BRA9.B.D8 A9
+BRA8.B.D7 A8 D7 25 +RAD15.D
+BRA7.B.D5 A7 D6 23 +RAD14.D
+BRA6.B.D4 A6 D5 20 +RAD13.D
+BRA5.B.D2 A5 D4 16 +RAD12.D
+BRA4.B.D1 A4 D3 13 +RAD11.D
+BRA3.B.D8 A3 D2 10 +RAD10.D
+BRA2.B.D7 A2 D1 6 +RAD9.D
+BRA1.B.D5 A1 D0 3 +RAD8.D
+BRA0.B.D4 A0
SIP1MX8
IC54
-RAS3.D27 RAS
-CAS3.D2 CAS
150nS
-WEH1.D21 WE
+BRA9.A.D8 A9
+BRA8.A.D7 A8 D7 25 +RAD7.D
+BRA7.A.D5 A7 D6 23 +RAD6.D
+BRA6.A.D4 A6 D5 20 +RAD5.D
+BRA5.A.D2 A5 D4 16 +RAD4.D
+BRA4.A.D1 A4 D3 13 +RAD3.D
+BRA3.A.D8 A3 D2 10 +RAD2.D
+BRA2.A.D7 A2 D1 6 +RAD1.D
+BRA1.A.D5 A1 D0 3 +RAD0.D
+BRA0.A.D4 A0
SIP1MX8
IC56
-RAS1.D27 RAS
-CAS1.D2 CAS
150nS
-WEL1.D21 WE
+BRA9.B.D8 A9
+BRA8.B.D7 A8 D7 25 +RAD7.D
+BRA7.B.D5 A7 D6 23 +RAD6.D
+BRA6.B.D4 A6 D5 20 +RAD5.D
+BRA5.B.D2 A5 D4 16 +RAD4.D
+BRA4.B.D1 A4 D3 13 +RAD3.D
+BRA3.B.D8 A3 D2 10 +RAD2.D
+BRA2.B.D7 A2 D1 6 +RAD1.D
+BRA1.B.D5 A1 D0 3 +RAD0.D
+BRA0.B.D4 A0
SIP1MX8
IC31
-RAS3.D27 RAS
-CAS3.D2 CAS
150nS
-WEL1.D21 WE

E–mu Systems, Inc.

E3 uCONTROLLER: SOUND RAM

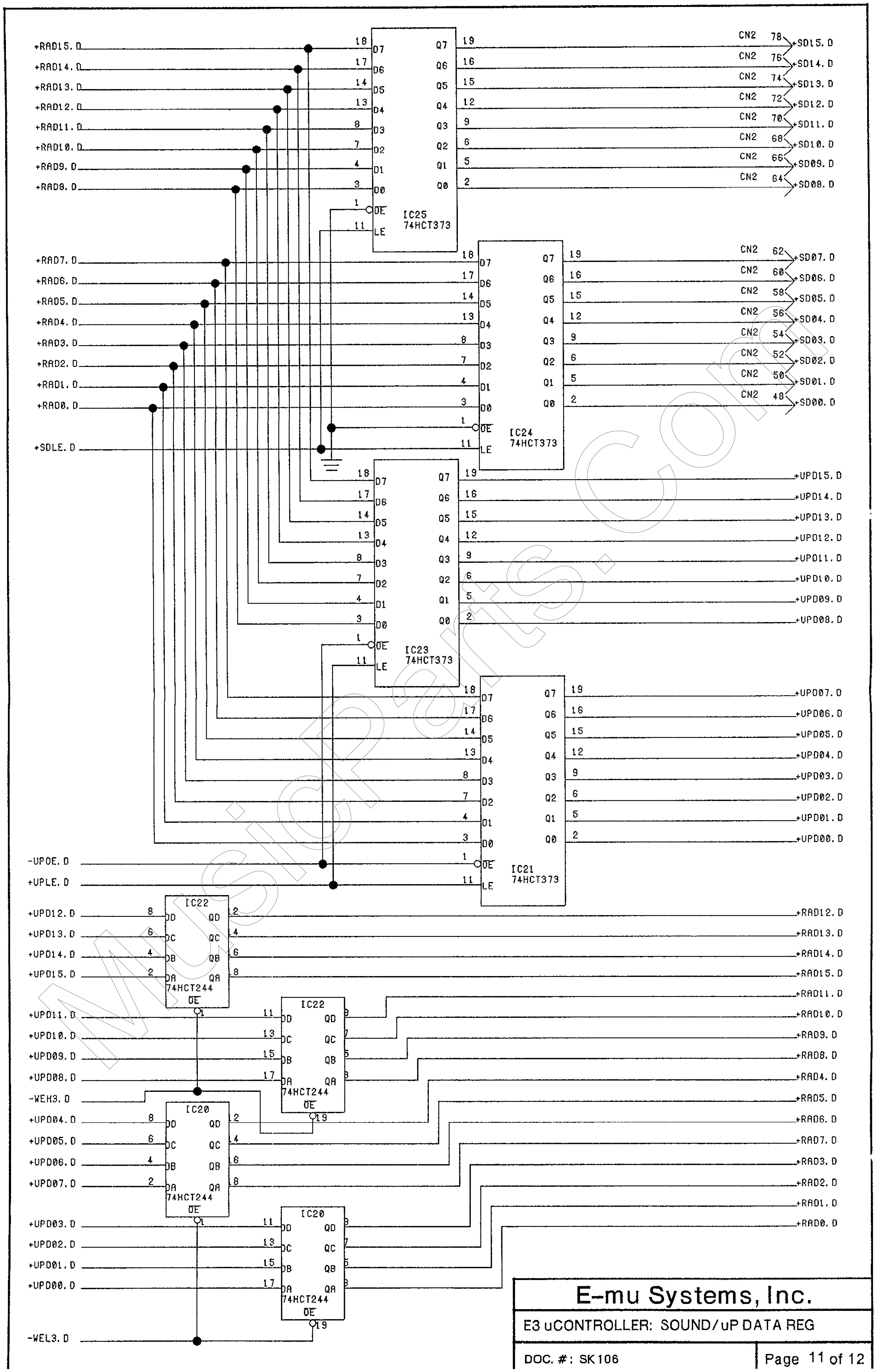
DOC. #: SK106

Page 9 of 12



+BRA9.C.D8 A9
+BRA8.C.D7 A8 D7 25 +RAD15.D
+BRA7.C.D5 A7 D6 23 +RAD14.D
+BRA6.C.D4 A6 D5 20 +RAD13.D
+BRA5.C.D2 A5 D4 16 +RAD12.D
+BRA4.C.D1 A4 D3 13 +RAD11.D
+BRA3.C.D18 A3 D2 10 +RAD10.D
+BRA2.C.D17 A2 D1 6 +RAD9.D
+BRA1.C.D15 A1 D0 3 +RAD8.D
+BRA0.C.D14 A0
-RAS4.D27 RAS
-CAS4.D2 CAS
-WEH2.D21 WE
SIP1MX8
IC30
150nS
+BRA9.D.D8 A9
+BRA8.D.D7 A8 D7 25 +RAD15.D
+BRA7.D.D5 A7 D6 23 +RAD14.D
+BRA6.D.D4 A6 D5 20 +RAD13.D
+BRA5.D.D2 A5 D4 16 +RAD12.D
+BRA4.D.D1 A4 D3 13 +RAD11.D
+BRA3.D.D18 A3 D2 10 +RAD10.D
+BRA2.D.D17 A2 D1 6 +RAD9.D
+BRA1.D.D15 A1 D0 3 +RAD8.D
+BRA0.D.D14 A0
-RAS6.D27 RAS
-CAS6.D2 CAS
-WEH2.D21 WE
SIP1MX8
IC4
150nS
+BRA9.C.D8 A9
+BRA8.C.D7 A8 D7 25 +RAD7.D
+BRA7.C.D5 A7 D6 23 +RAD6.D
+BRA6.C.D4 A6 D5 20 +RAD5.D
+BRA5.C.D2 A5 D4 16 +RAD4.D
+BRA4.C.D1 A4 D3 13 +RAD3.D
+BRA3.C.D18 A3 D2 10 +RAD2.D
+BRA2.C.D17 A2 D1 6 +RAD1.D
+BRA1.C.D15 A1 D0 3 +RAD0.D
+BRA0.C.D14 A0
-RAS4.D27 RAS
-CAS4.D2 CAS
-WEL2.D21 WE
SIP1MX8
IC28
150nS
+BRA9.D.D8 A9
+BRA8.D.D7 A8 D7 25 +RAD7.D
+BRA7.D.D5 A7 D6 23 +RAD6.D
+BRA6.D.D4 A6 D5 20 +RAD5.D
+BRA5.D.D2 A5 D4 16 +RAD4.D
+BRA4.D.D1 A4 D3 13 +RAD3.D
+BRA3.D.D18 A3 D2 10 +RAD2.D
+BRA2.D.D17 A2 D1 6 +RAD1.D
+BRA1.D.D15 A1 D0 3 +RAD0.D
+BRA0.D.D14 A0
-RAS6.D27 RAS
-CAS6.D2 CAS
-WEL2.D21 WE
SIP1MX8
IC2
150nS
+BRA9.C.D8 A9
+BRA8.C.D7 A8 D7 25 +RAD15.D
+BRA7.C.D5 A7 D6 23 +RAD14.D
+BRA6.C.D4 A6 D5 20 +RAD13.D
+BRA5.C.D2 A5 D4 16 +RAD12.D
+BRA4.C.D1 A4 D3 13 +RAD11.D
+BRA3.C.D18 A3 D2 10 +RAD10.D
+BRA2.C.D17 A2 D1 6 +RAD9.D
+BRA1.C.D15 A1 D0 3 +RAD8.D
+BRA0.C.D14 A0
-RAS5.D27 RAS
-CAS5.D2 CAS
-WEH3.D21 WE
SIP1MX8
IC29
150nS
+BRA9.D.D8 A9
+BRA8.D.D7 A8 D7 25 +RAD15.D
+BRA7.D.D5 A7 D6 23 +RAD14.D
+BRA6.D.D4 A6 D5 20 +RAD13.D
+BRA5.D.D2 A5 D4 16 +RAD12.D
+BRA4.D.D1 A4 D3 13 +RAD11.D
+BRA3.D.D18 A3 D2 10 +RAD10.D
+BRA2.D.D17 A2 D1 6 +RAD9.D
+BRA1.D.D15 A1 D0 3 +RAD8.D
+BRA0.D.D14 A0
-RAS7.D27 RAS
-CAS7.D2 CAS
-WEH3.D21 WE
SIP1MX8
IC3
150nS
+BRA9.C.D8 A9
+BRA8.C.D7 A8 D7 25 +RAD7.D
+BRA7.C.D5 A7 D6 23 +RAD6.D
+BRA6.C.D4 A6 D5 20 +RAD5.D
+BRA5.C.D2 A5 D4 16 +RAD4.D
+BRA4.C.D1 A4 D3 13 +RAD3.D
+BRA3.C.D18 A3 D2 10 +RAD2.D
+BRA2.C.D17 A2 D1 6 +RAD1.D
+BRA1.C.D15 A1 D0 3 +RAD0.D
+BRA0.C.D14 A0
-RAS5.D27 RAS
-CAS5.D2 CAS
-WEL3.D21 WE
SIP1MX8
IC27
150nS
+BRA9.D.D8 A9
+BRA8.D.D7 A8 D7 25 +RAD7.D
+BRA7.D.D5 A7 D6 23 +RAD6.D
+BRA6.D.D4 A6 D5 20 +RAD5.D
+BRA5.D.D2 A5 D4 16 +RAD4.D
+BRA4.D.D1 A4 D3 13 +RAD3.D
+BRA3.D.D18 A3 D2 10 +RAD2.D
+BRA2.D.D17 A2 D1 6 +RAD1.D
+BRA1.D.D15 A1 D0 3 +RAD0.D
+BRA0.D.D14 A0
-RAS7.D27 RAS
-CAS7.D2 CAS
-WEL3.D21 WE
SIP1MX8
IC1
150nS

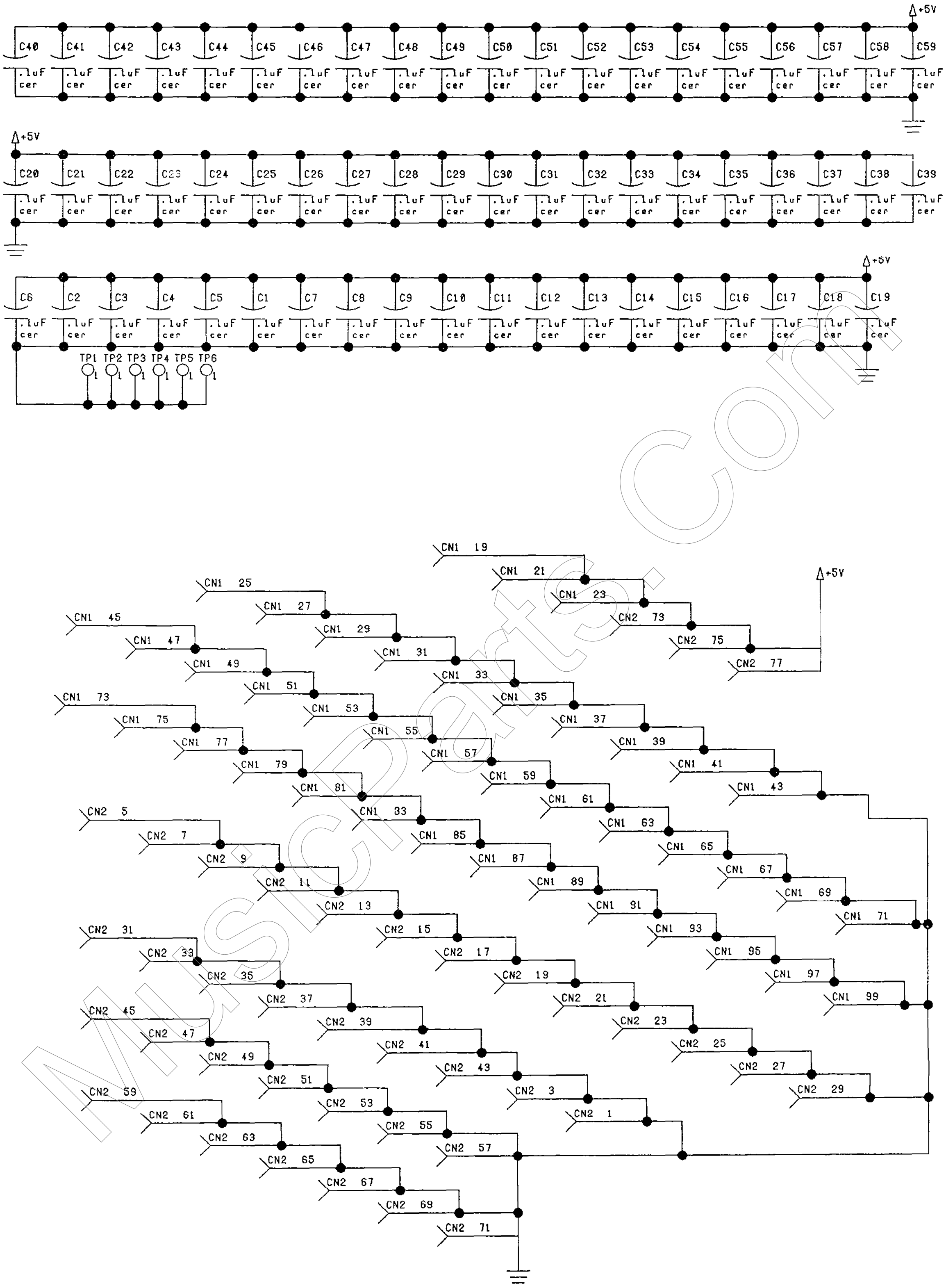
E-mu Systems, Inc.	
E3 uCONTROLLER: SOUND RAM (CONT.)	
DOC. #: SK106	Page 10 of 12



E-mu Systems, Inc.

E3 uCONTROLLER: SOUND/uP DATA REG

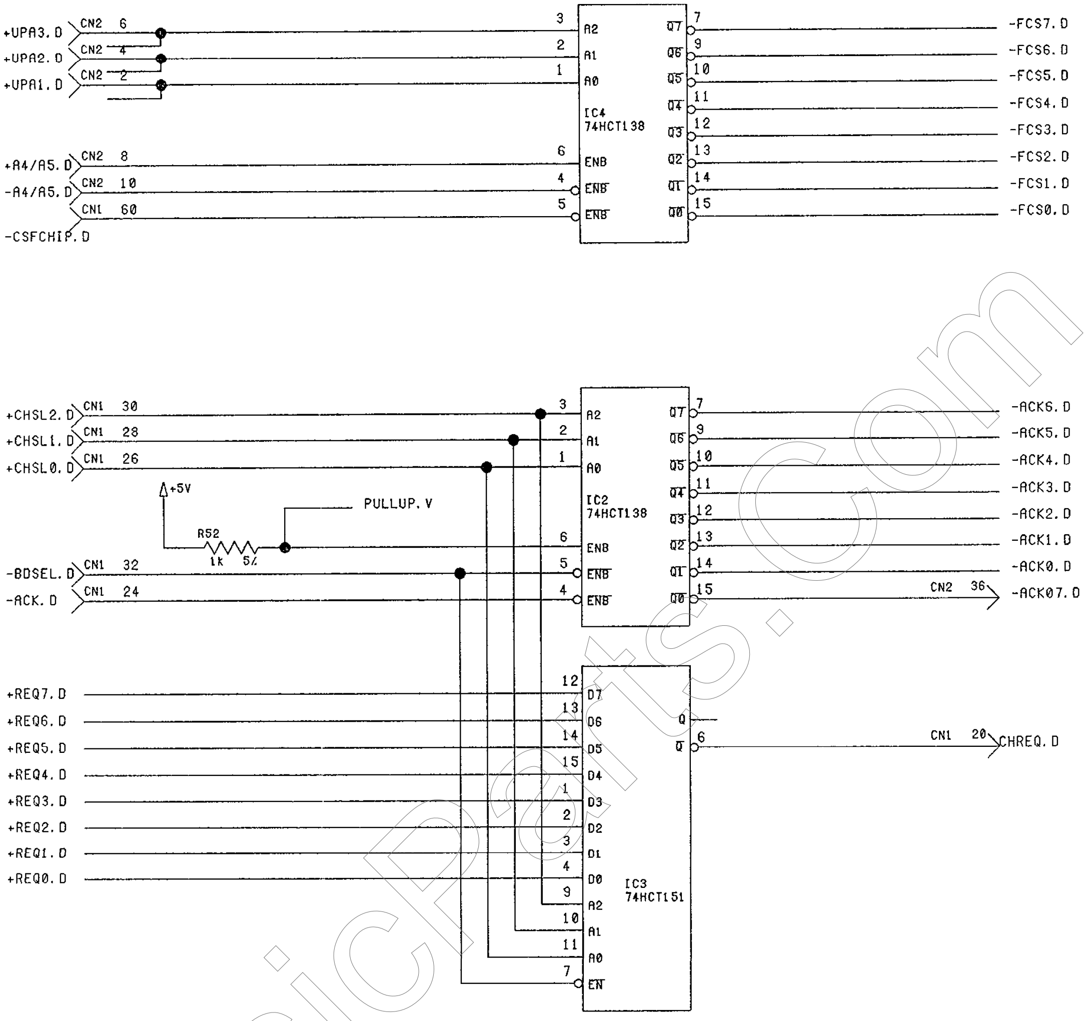
DOC. #: SK106



E–mu Systems, Inc.

E3 uCONTROLLER: CONNECTIONS, DECOUPLING

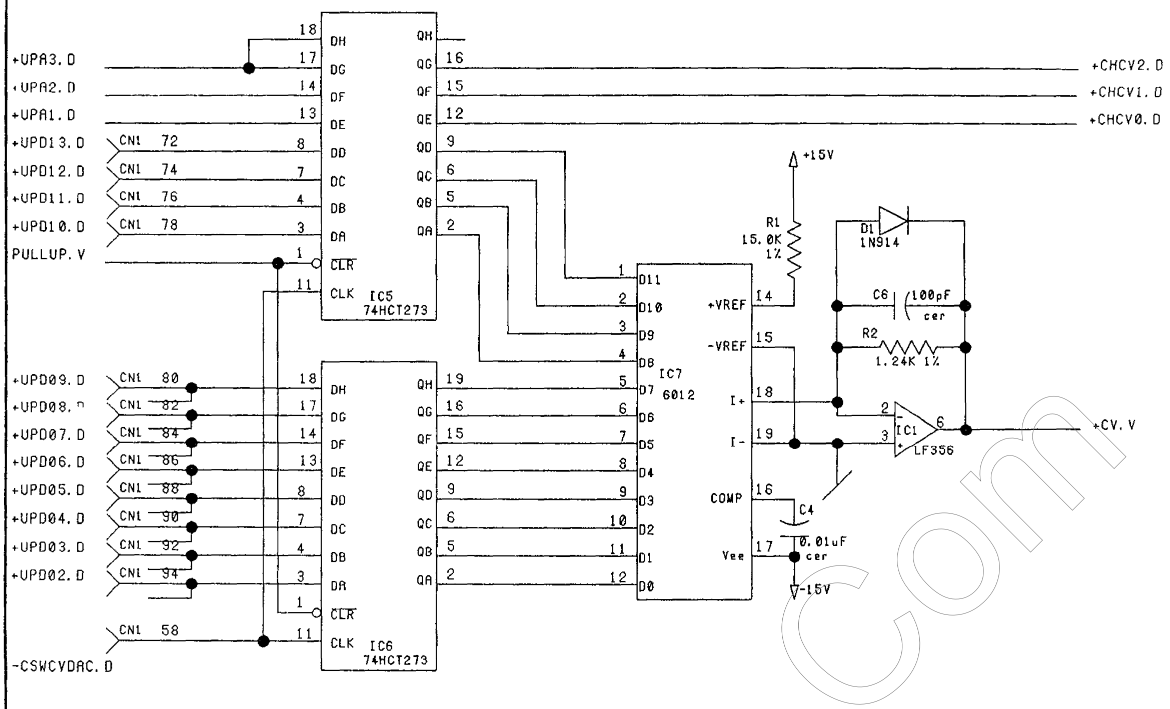
DOC. #: SK106	Page 12 of 12



E–mu Systems, Inc.
applied magic for the arts

E3 OUTPUT: Board and Channel Selects

DRAWN BY: BFS	DOC.#: SK105
REV C	DATE 871014

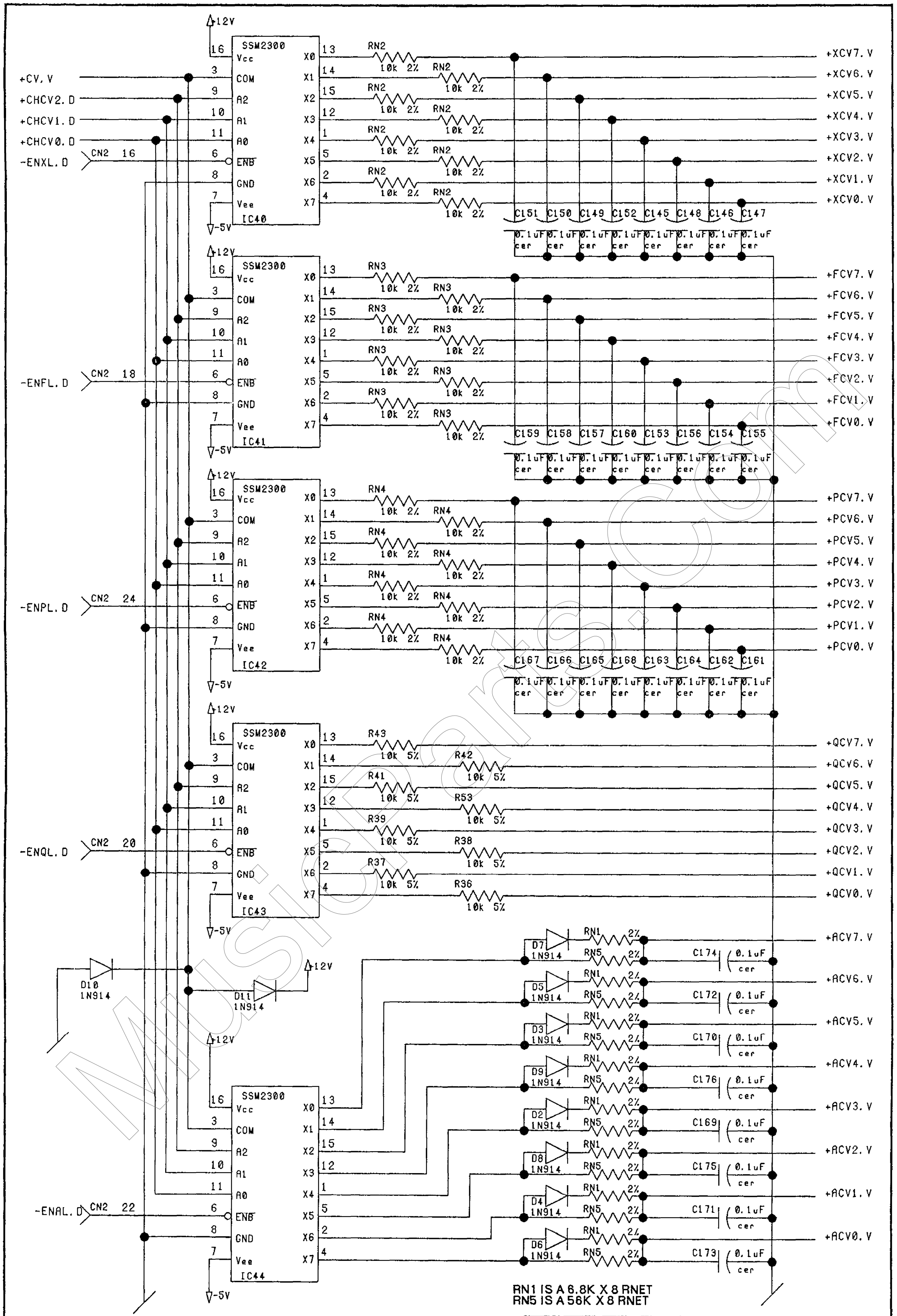


E-mu Systems, Inc.

E3 OUTPUT: Control Voltage DAC

DOC. #: SK105

Page 2 of 12

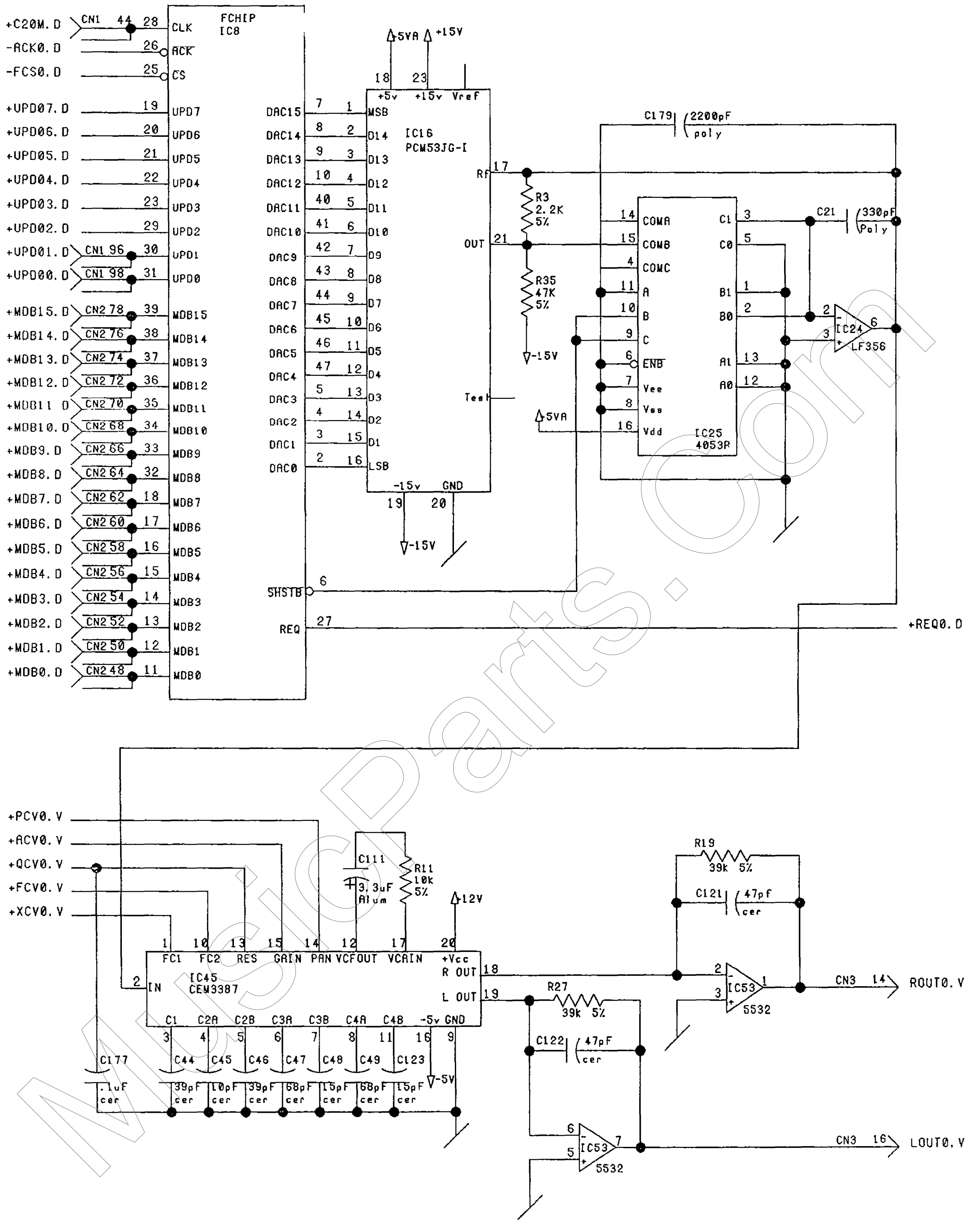


E-mu Systems, Inc.

E3 OUTPUT: CV MUX's

DOC. #: SK105

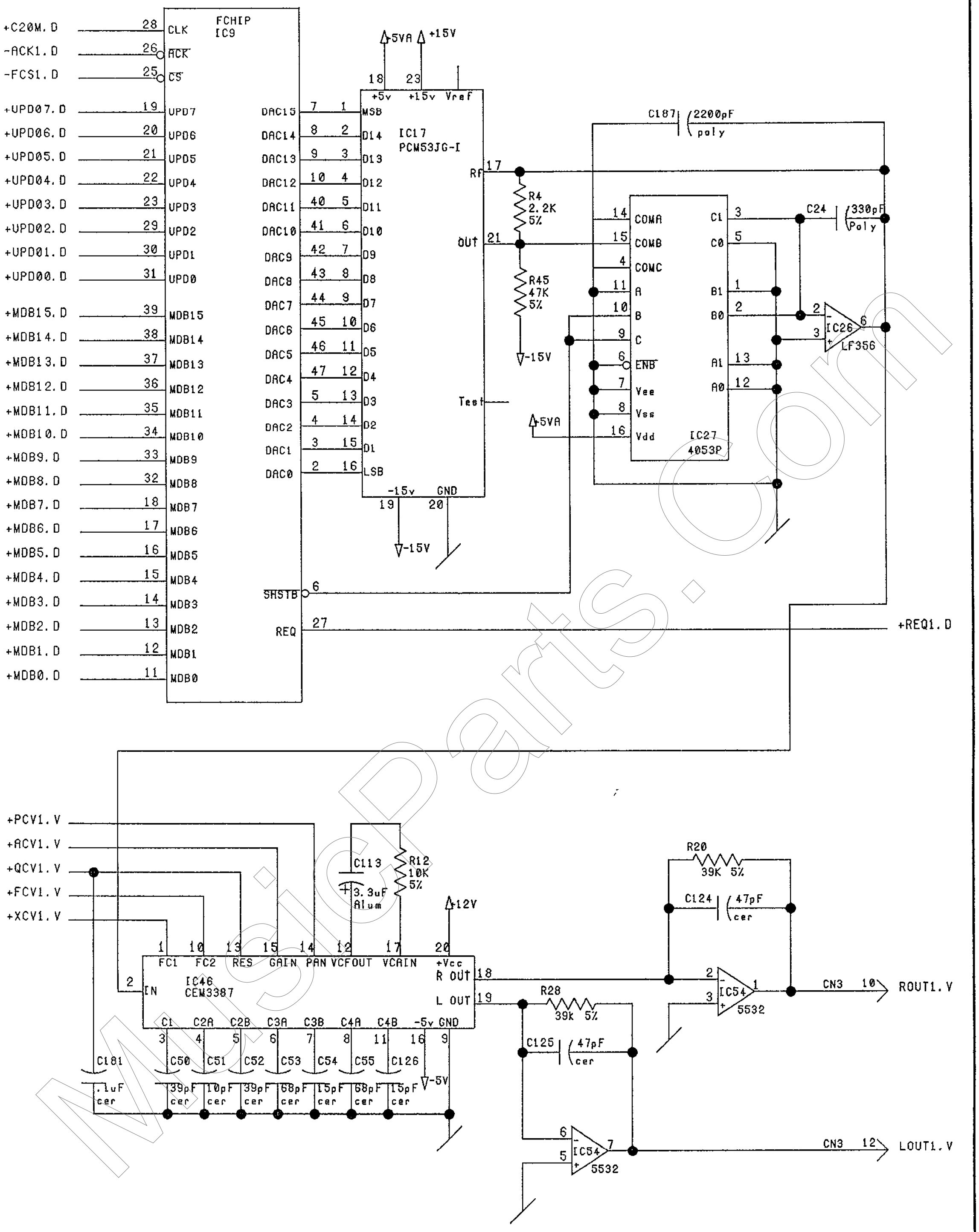
Page 3 of 12



E-mu Systems, Inc.

E3 OUTPUT: Output Channel 0

DOC. #: SK105

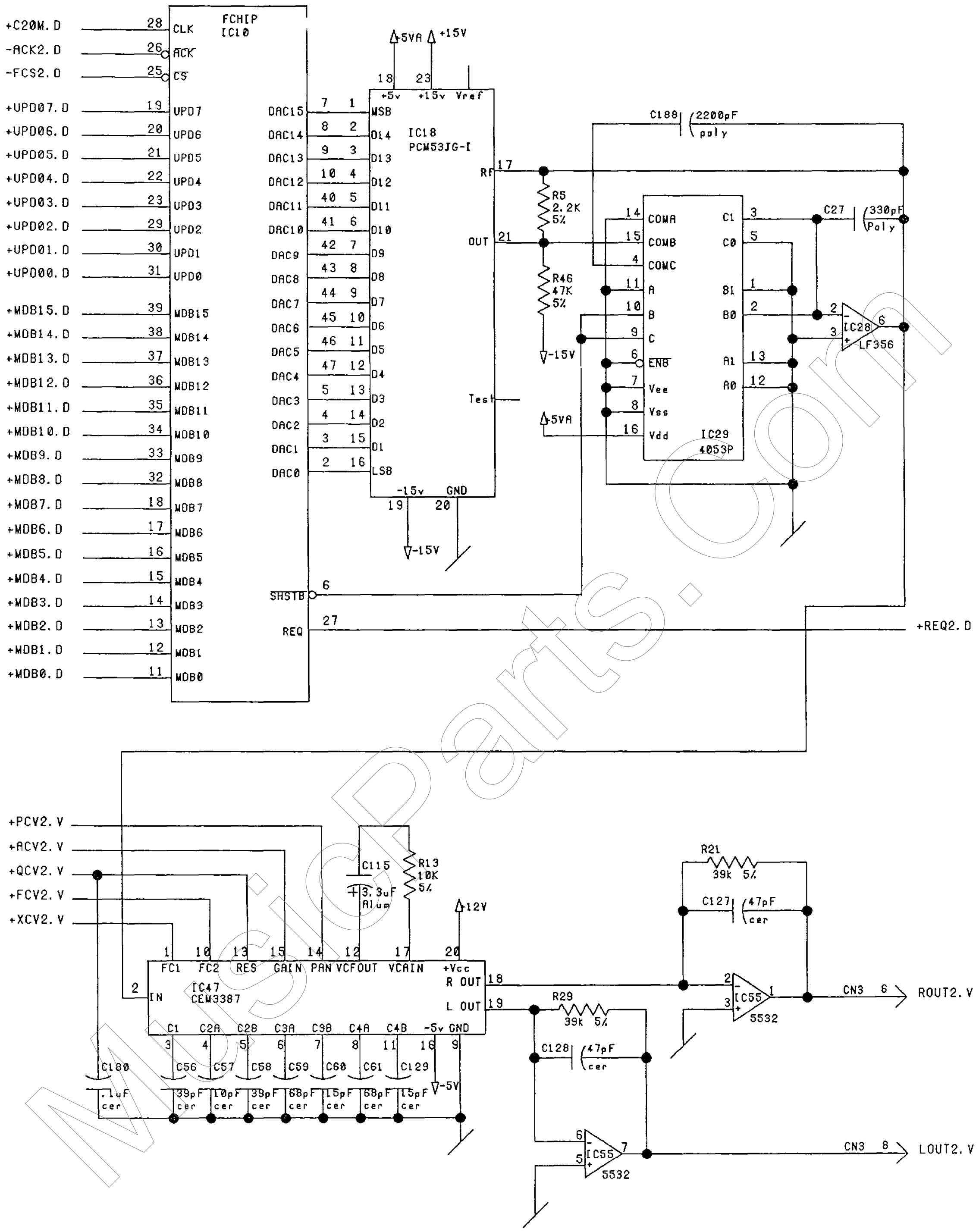


E–mu Systems, Inc.

E3 OUTPUT: Output Channel 1

DOC. #: SK105

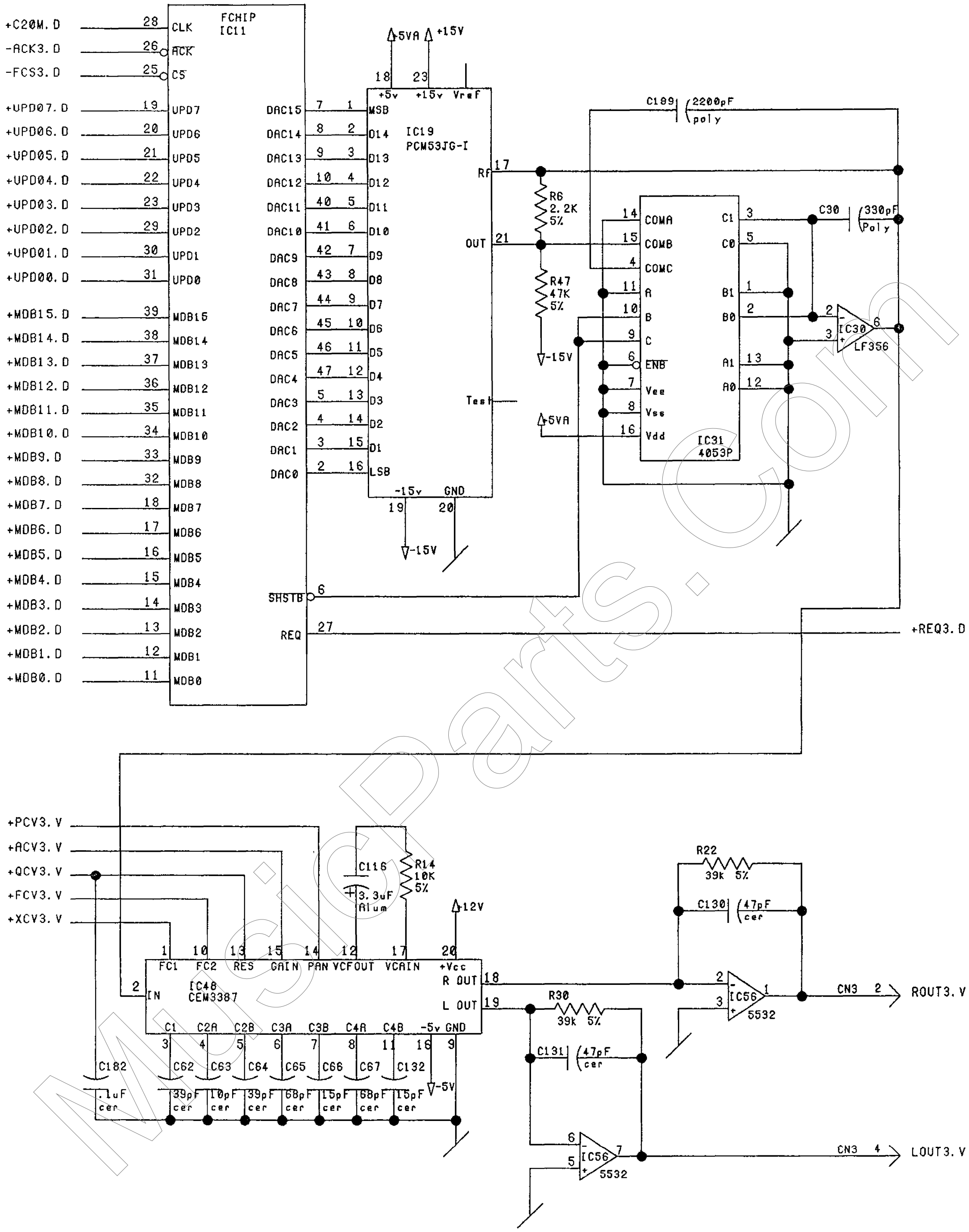
Page 5 of 12



E-mu Systems, Inc.

E3 OUTPUT: Output Channel 2

DOC. #: SK105

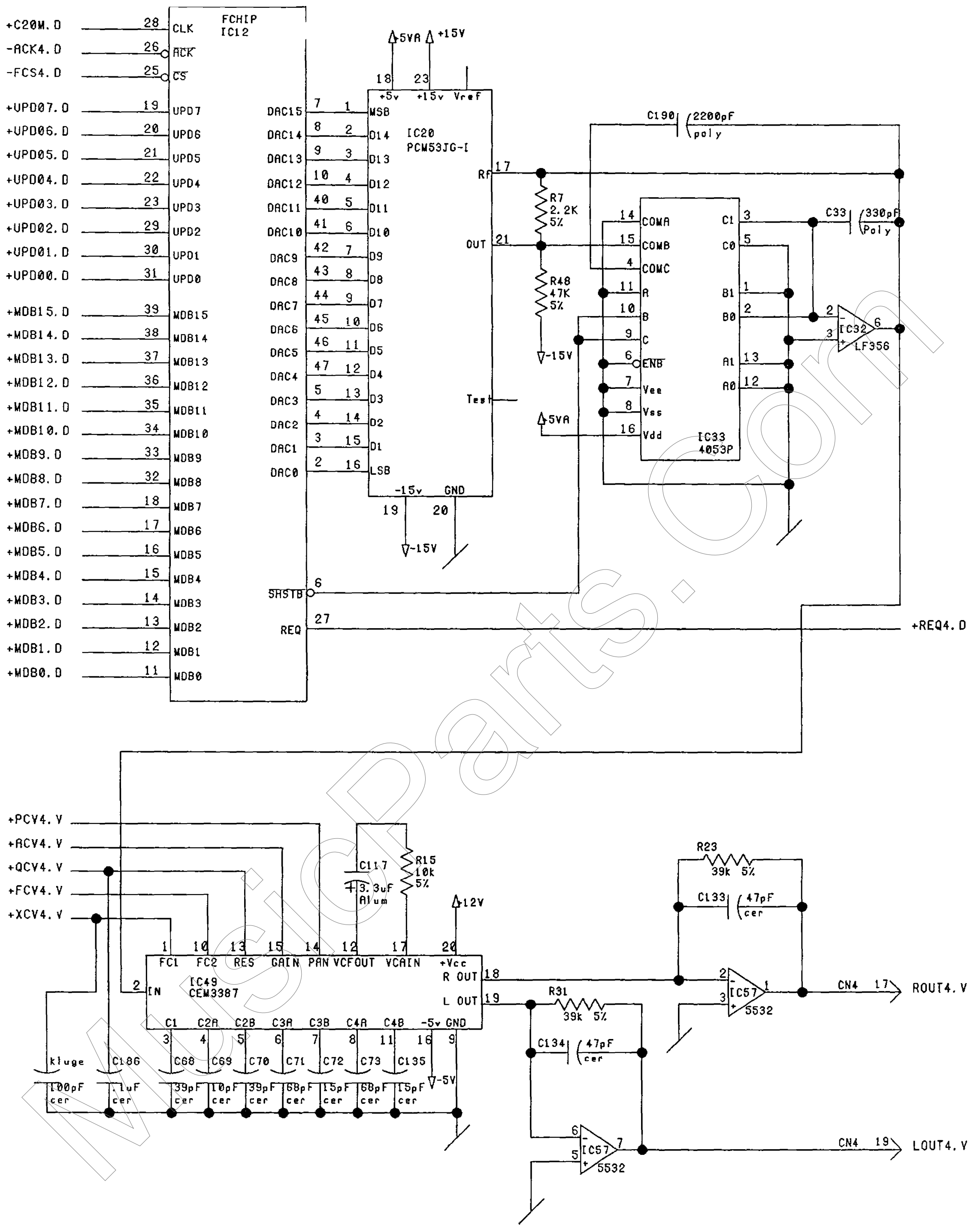


E-mu Systems, Inc.

E3 OUTPUT: Output Channel 3

DOC. #: SK105

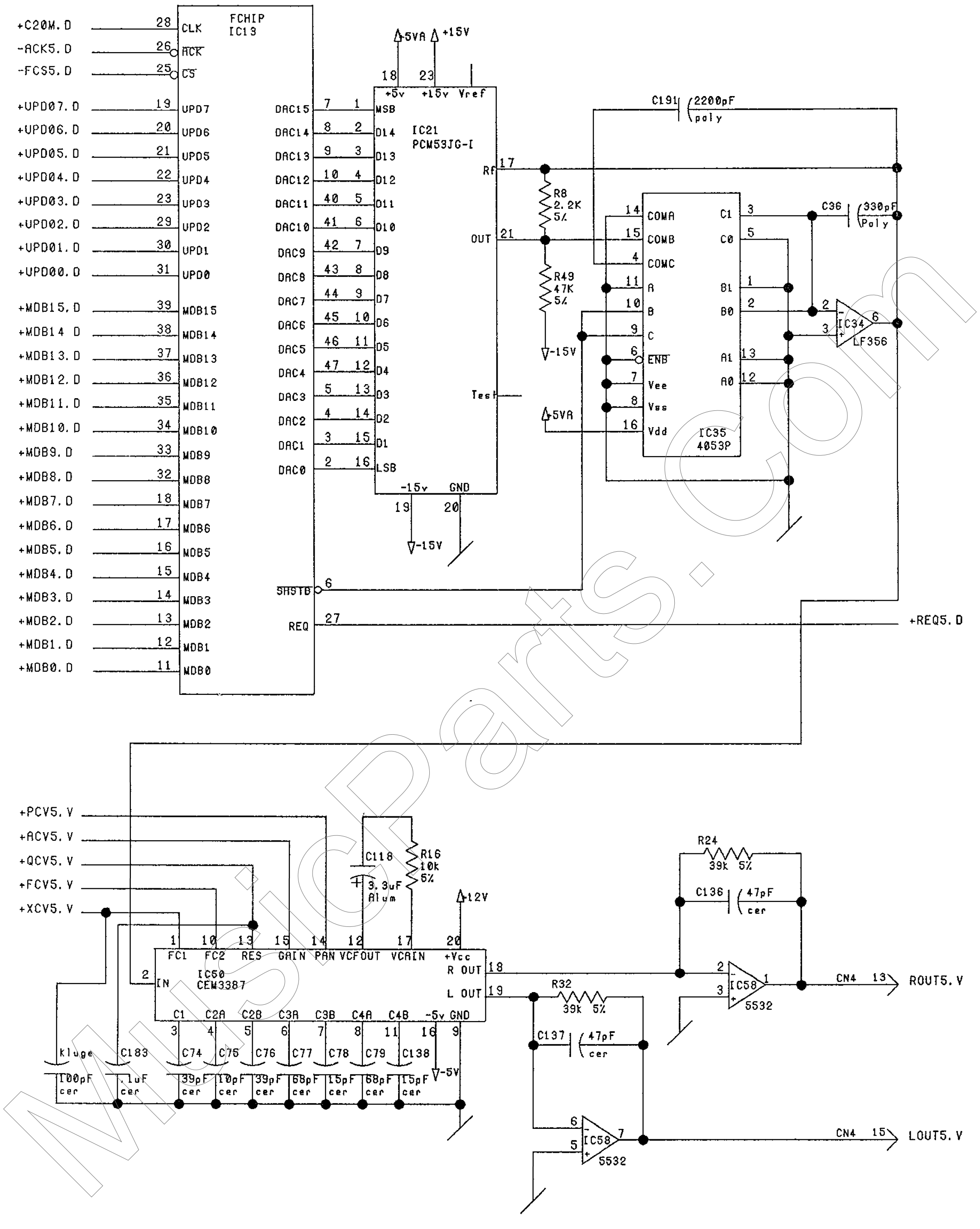
Page 7 of 12



E-mu Systems, Inc.

E3 OUTPUT: Output Channel 4

DOC. #: SK105

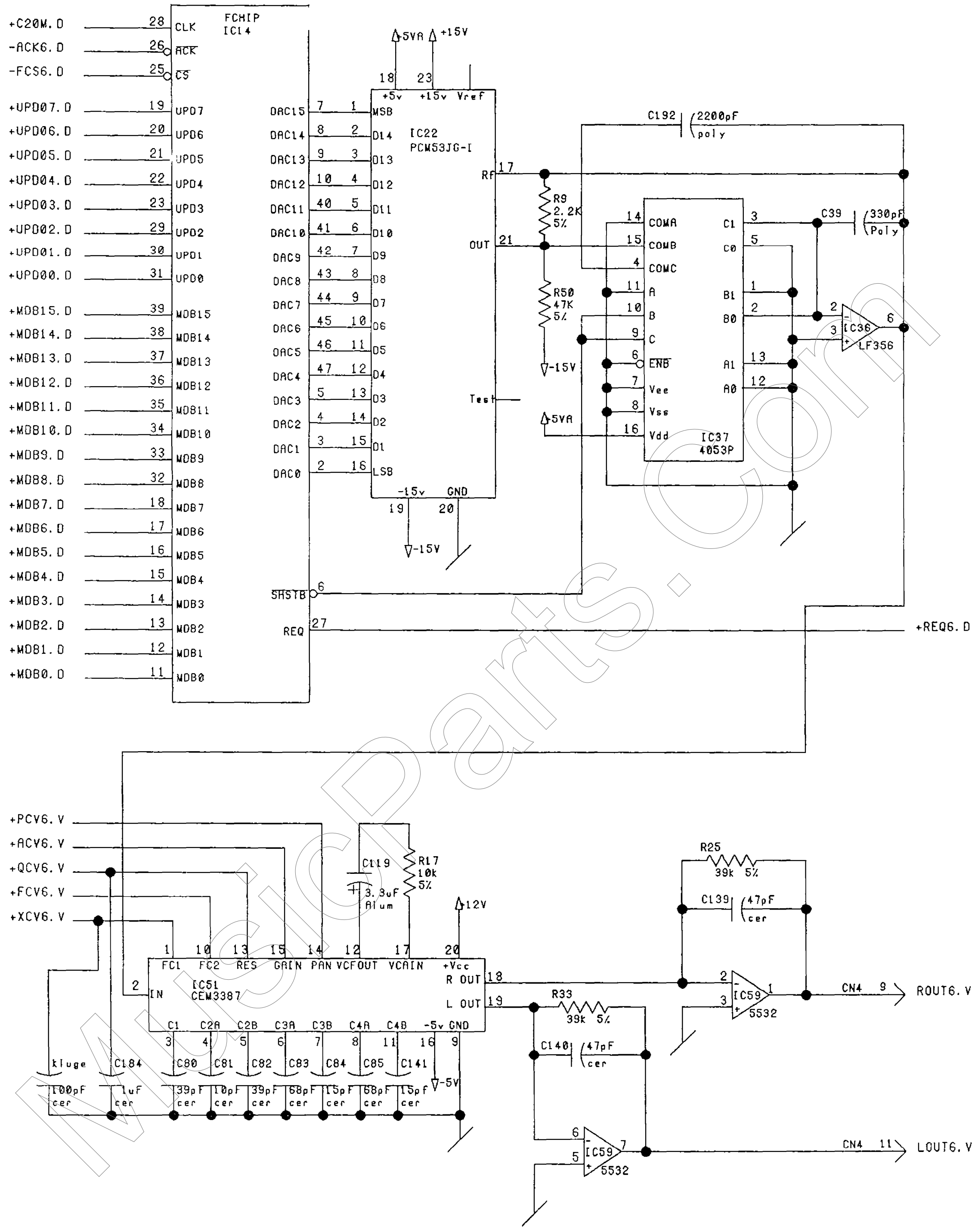


E-mu Systems, Inc.

E3 OUTPUT: Output Channel 5

DOC. #: SK105

Page 9 of 12

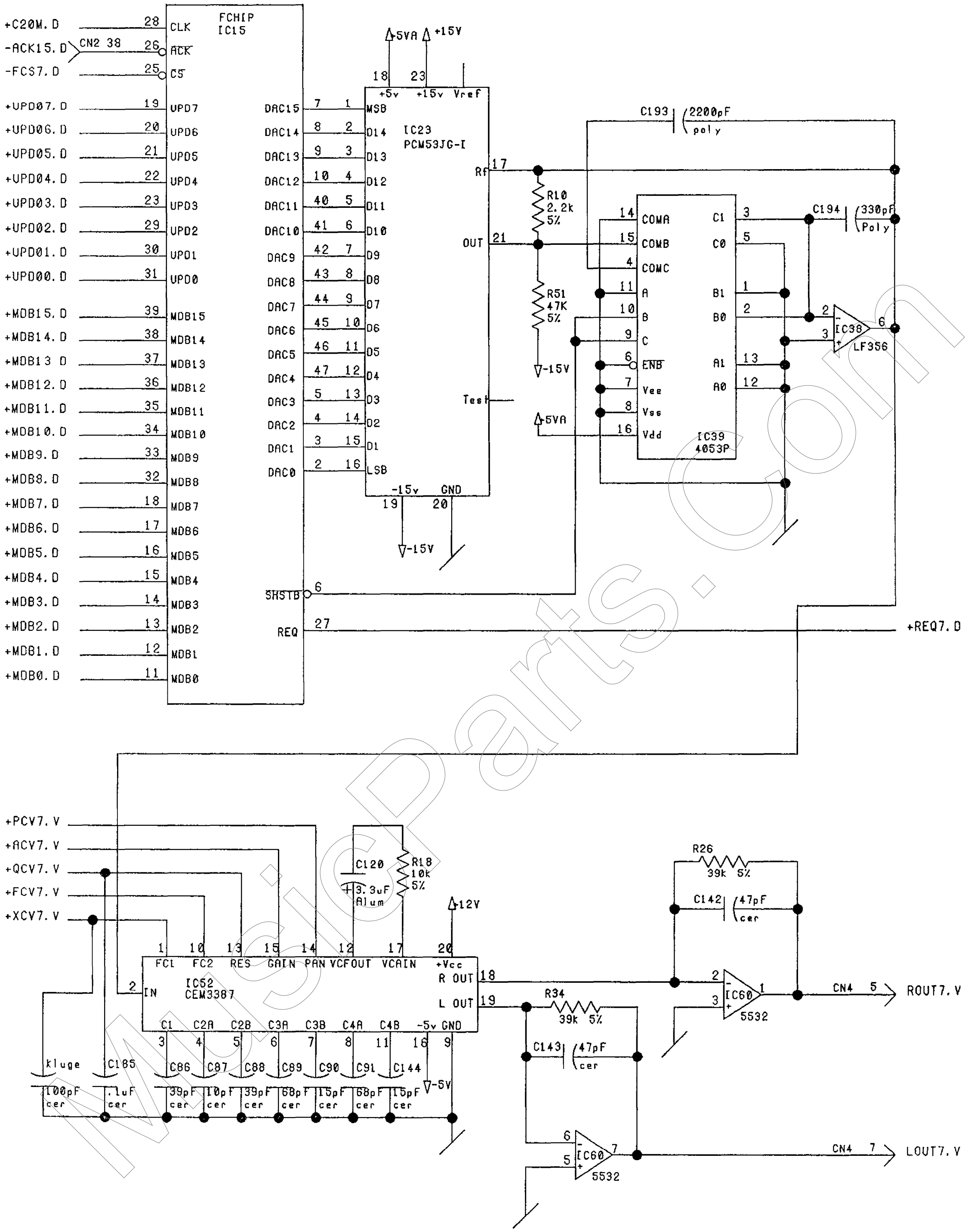


E-mu Systems, Inc.

E3 OUTPUT: Output Channel 6

DOC. #: SK105

Page 10of 12

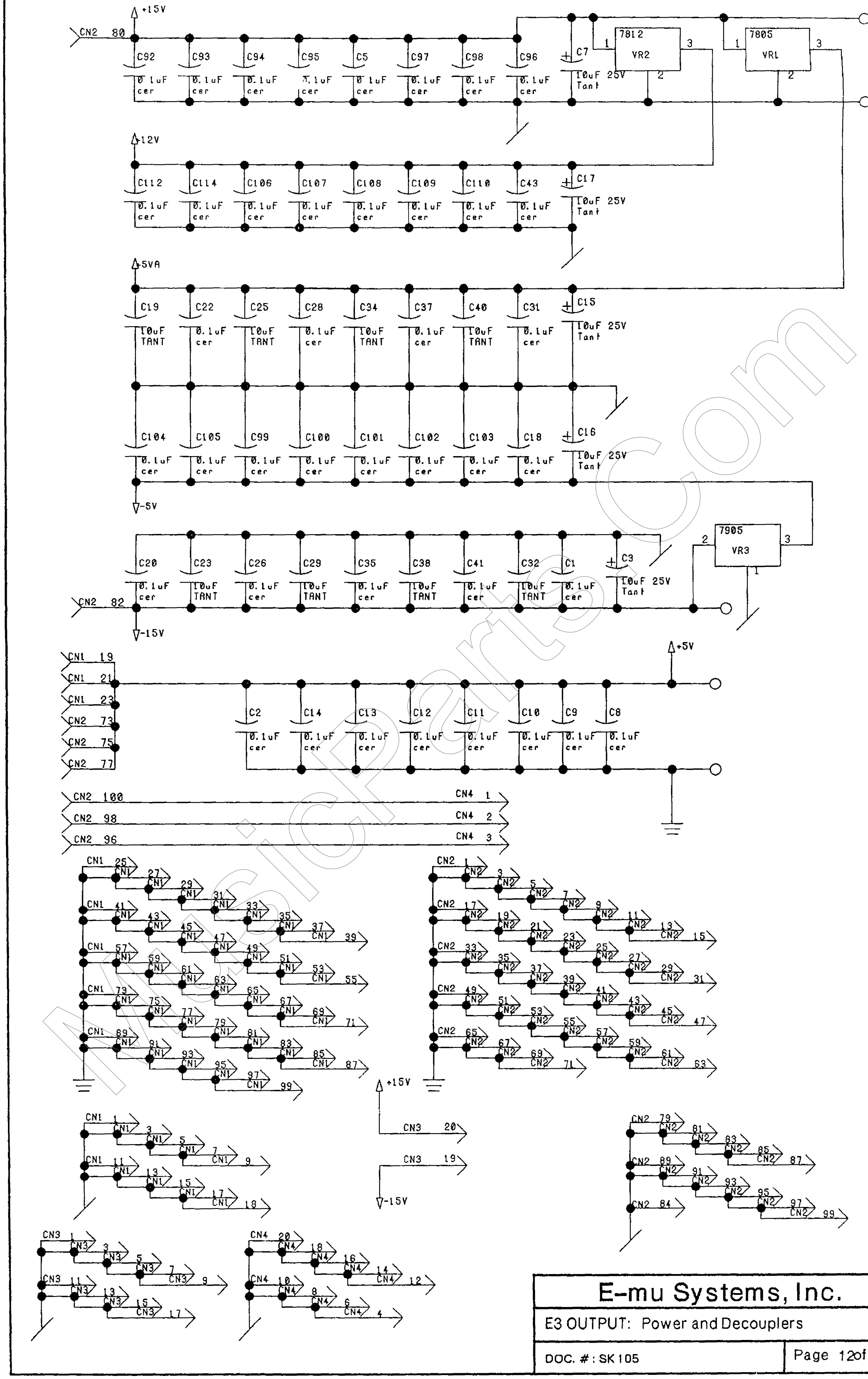


E–mu Systems, Inc.

E3 OUTPUT: Output Channel 7

DOC. #: SK105

Page 11of 12

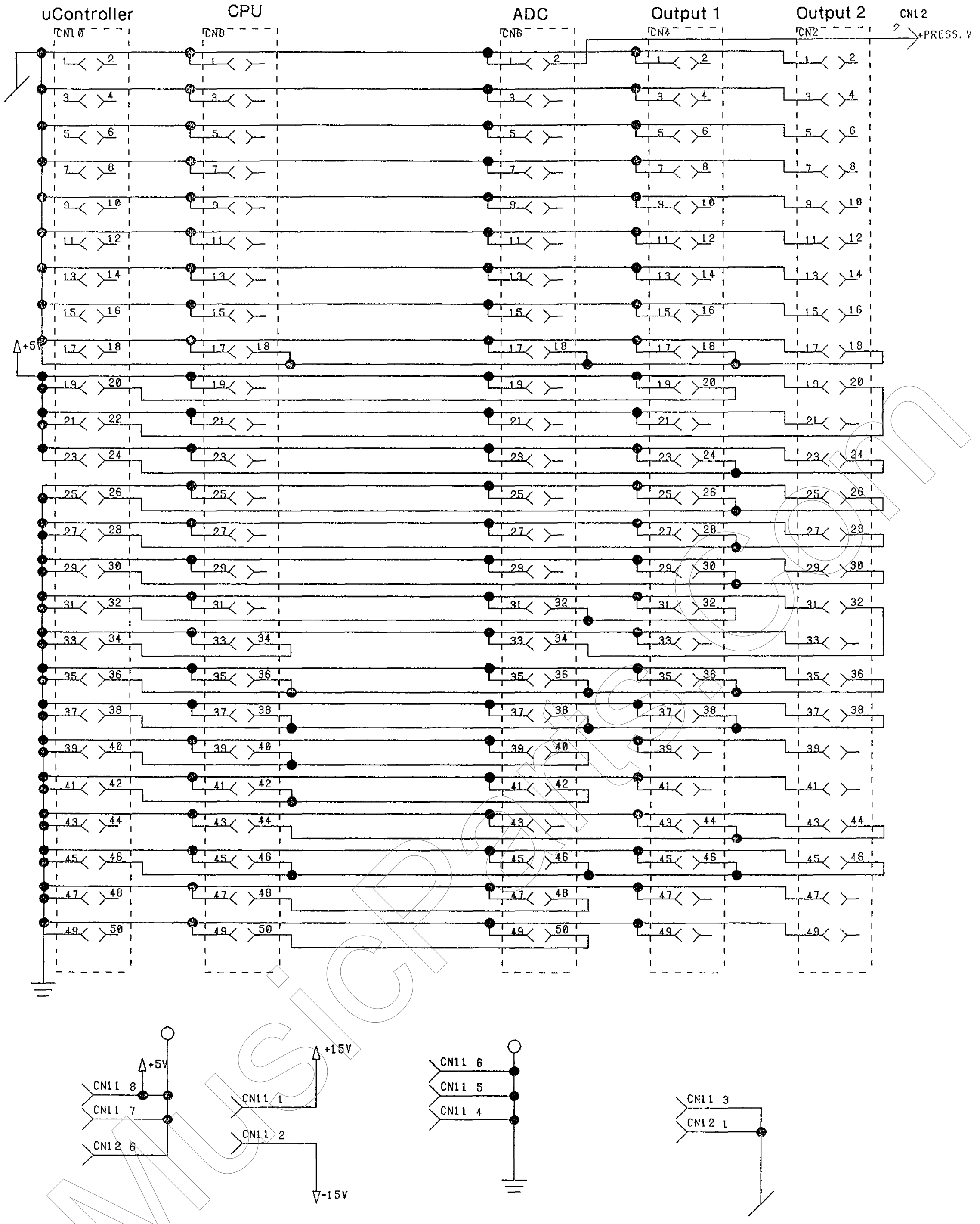


E-mu Systems, Inc.

E3 OUTPUT: Power and Decouplers

DOC. #: SK105

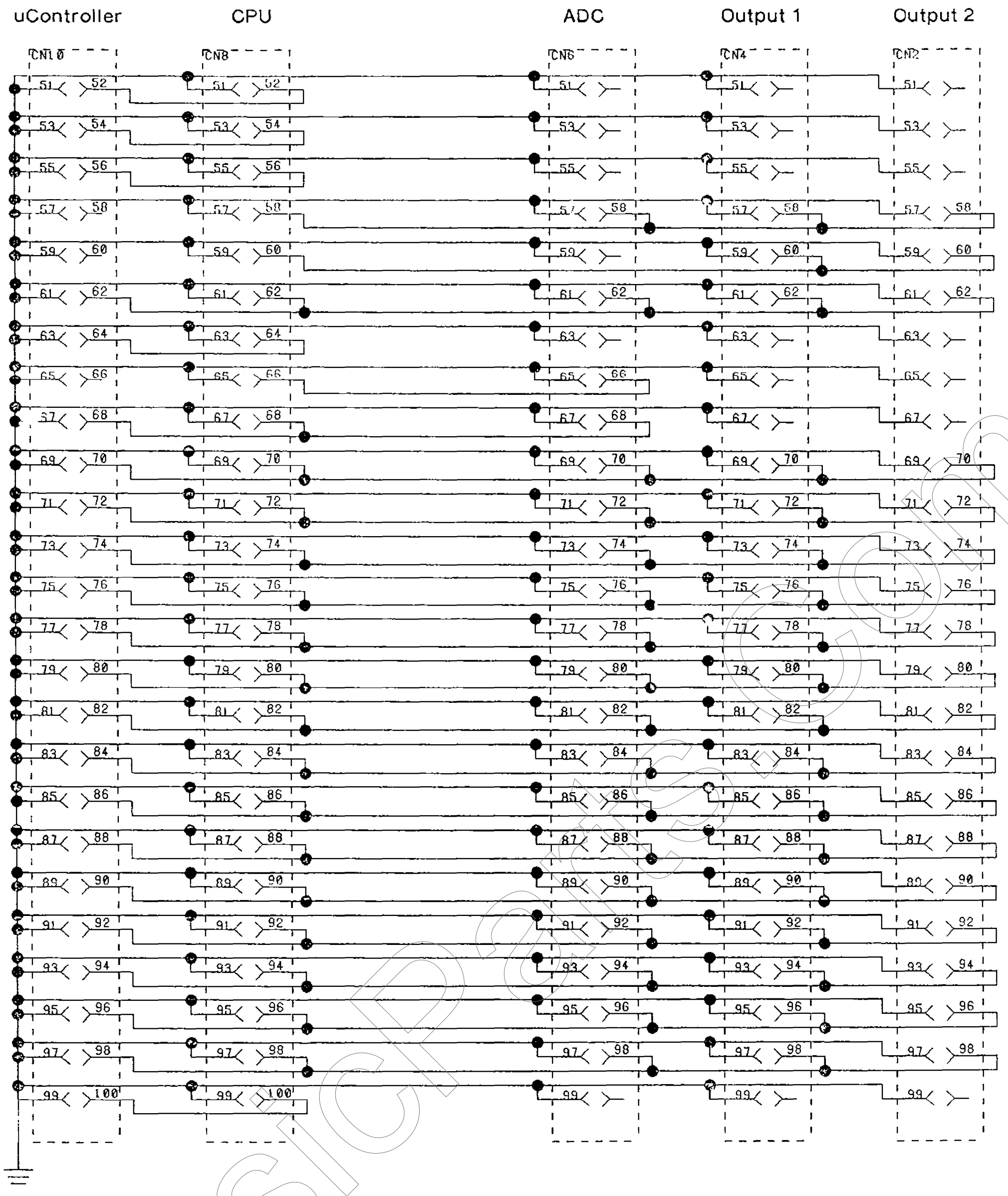
Page 12of 12



E–mu Systems, Inc.
applied magic for the arts

SCHEMATIC PCB MOTHERBOARD E3

DRAWN BY: BFS	DOC #: SK109	
REV B	DATE 880522	PAGE 1 OF 4

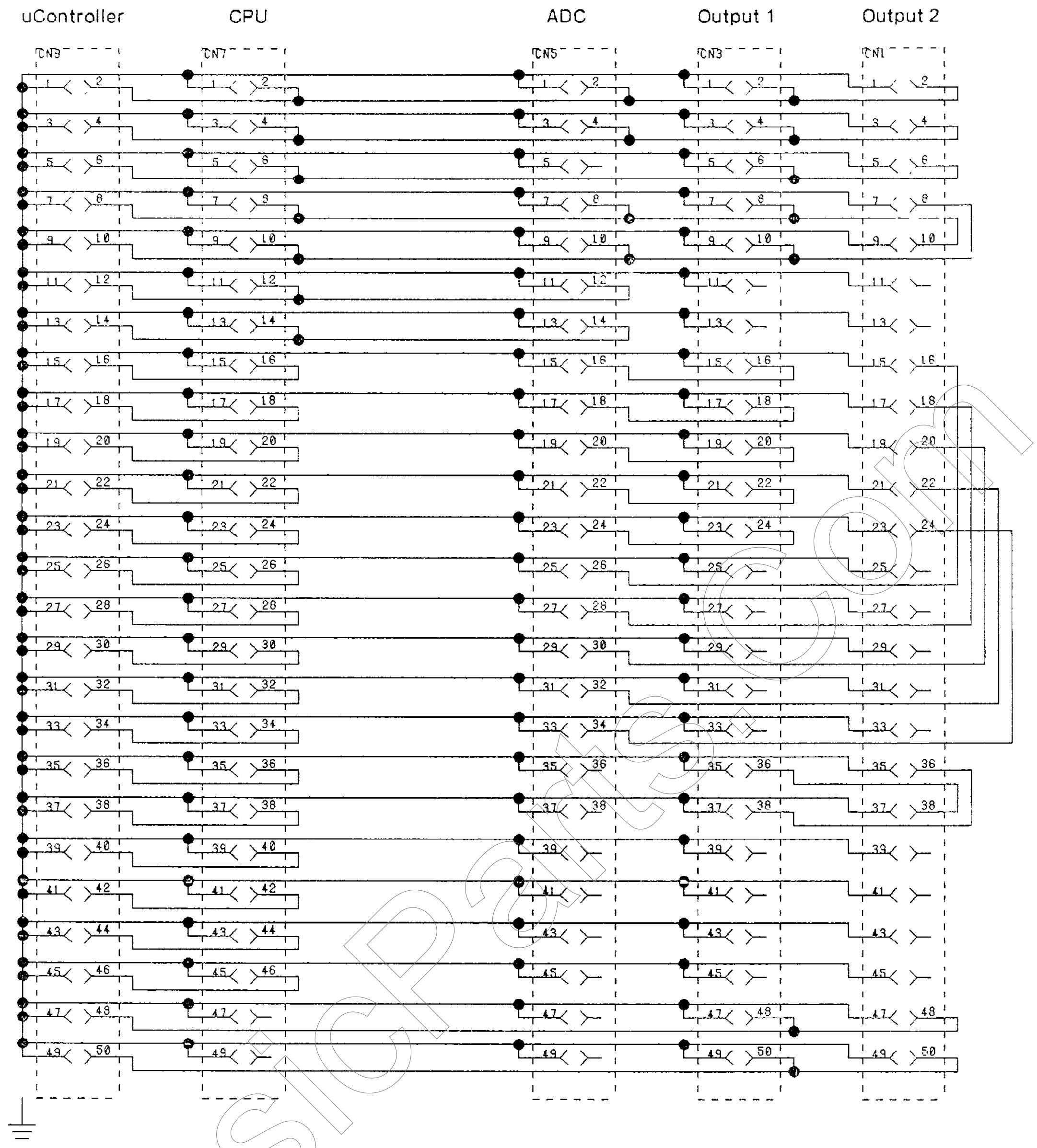


E-mu Systems, Inc.

SCHEMATIC PCB MOTHERBOARD E3

DOC #: SK109

PAGE 2 OF 4

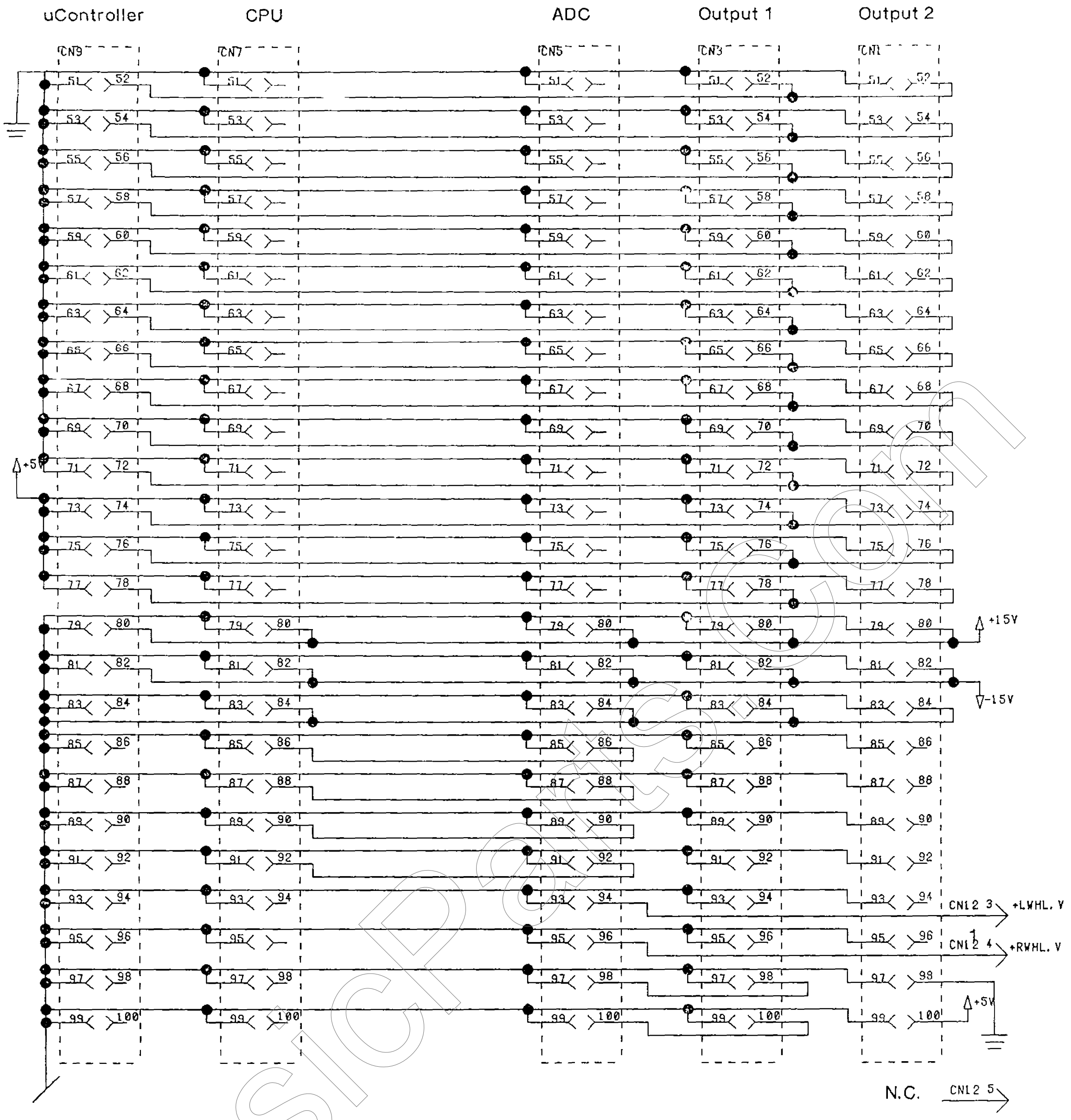


E-mu Systems, Inc.

SCHEMATIC PCB MOTHERBOARD E3

DOC #: SK109

PAGE 3 OF 4

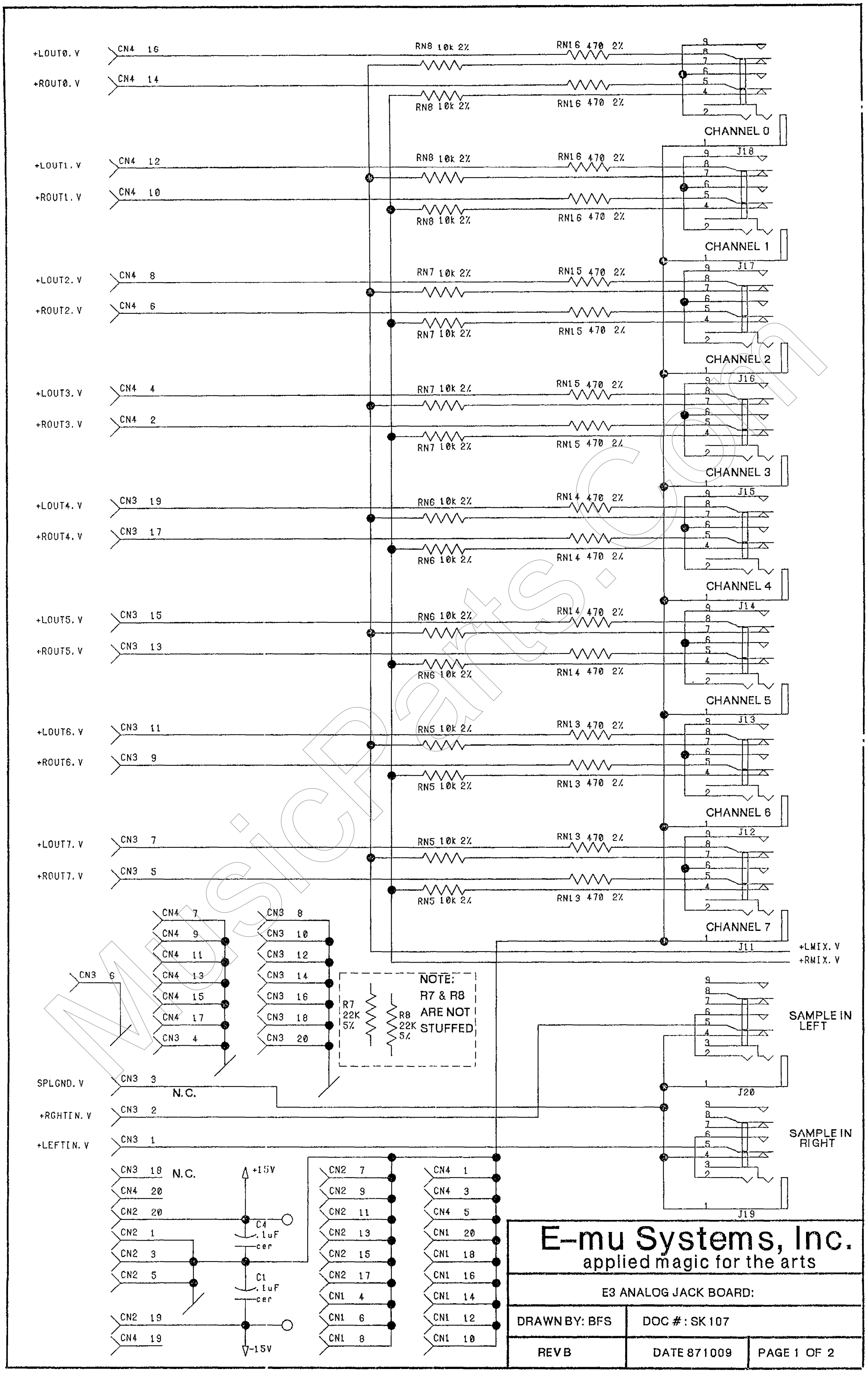


E-mu Systems, Inc.

SCHEMATIC PCB MOTHERBOARD E3

DOC #: SK 109

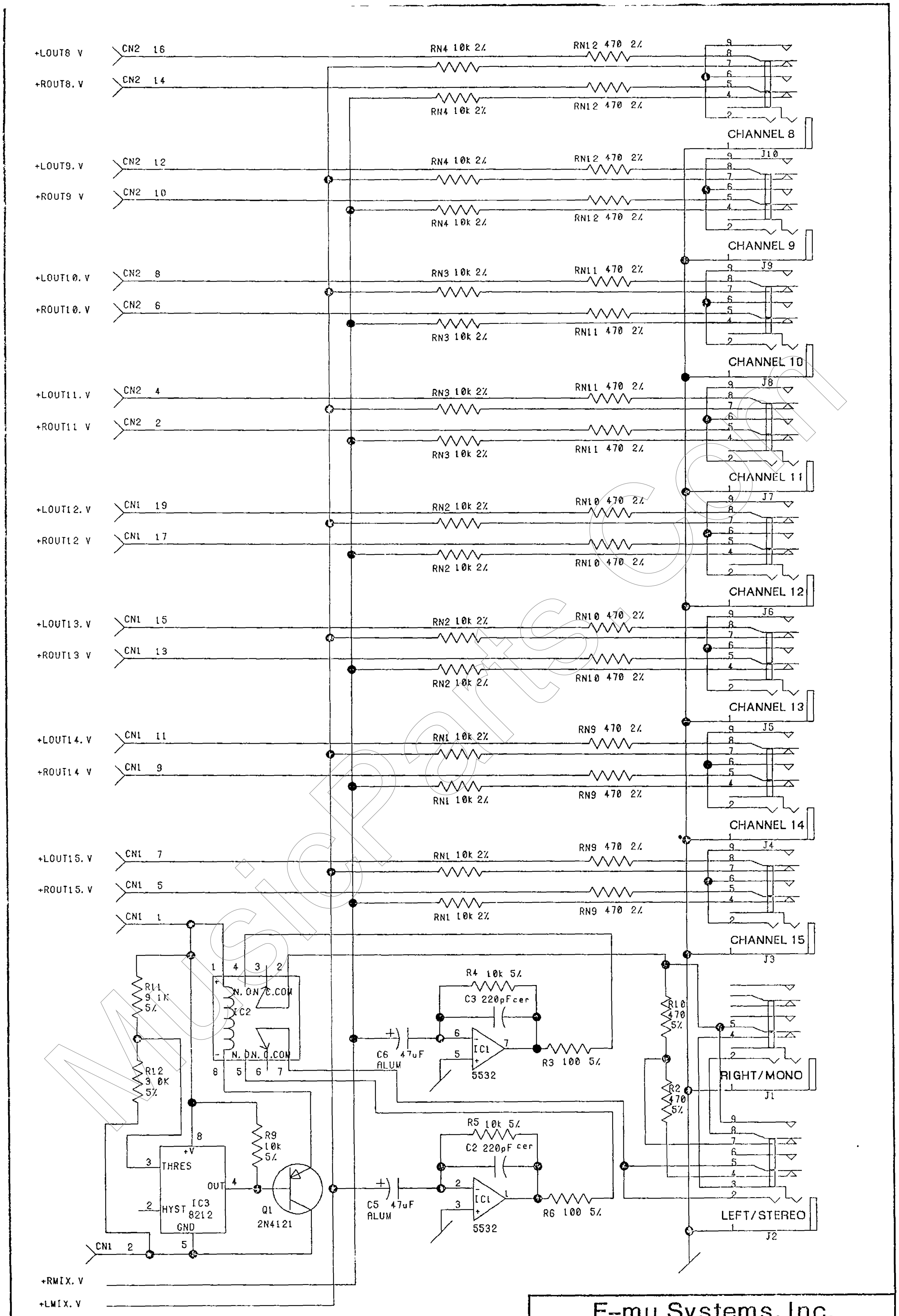
PAGE 4 OF 4



E-mu Systems, Inc.
applied magic for the arts

E3 ANALOG JACK BOARD:

DRAWN BY: BFS	DOC #: SK107	
REV B	DATE 871009	PAGE 1 OF 2

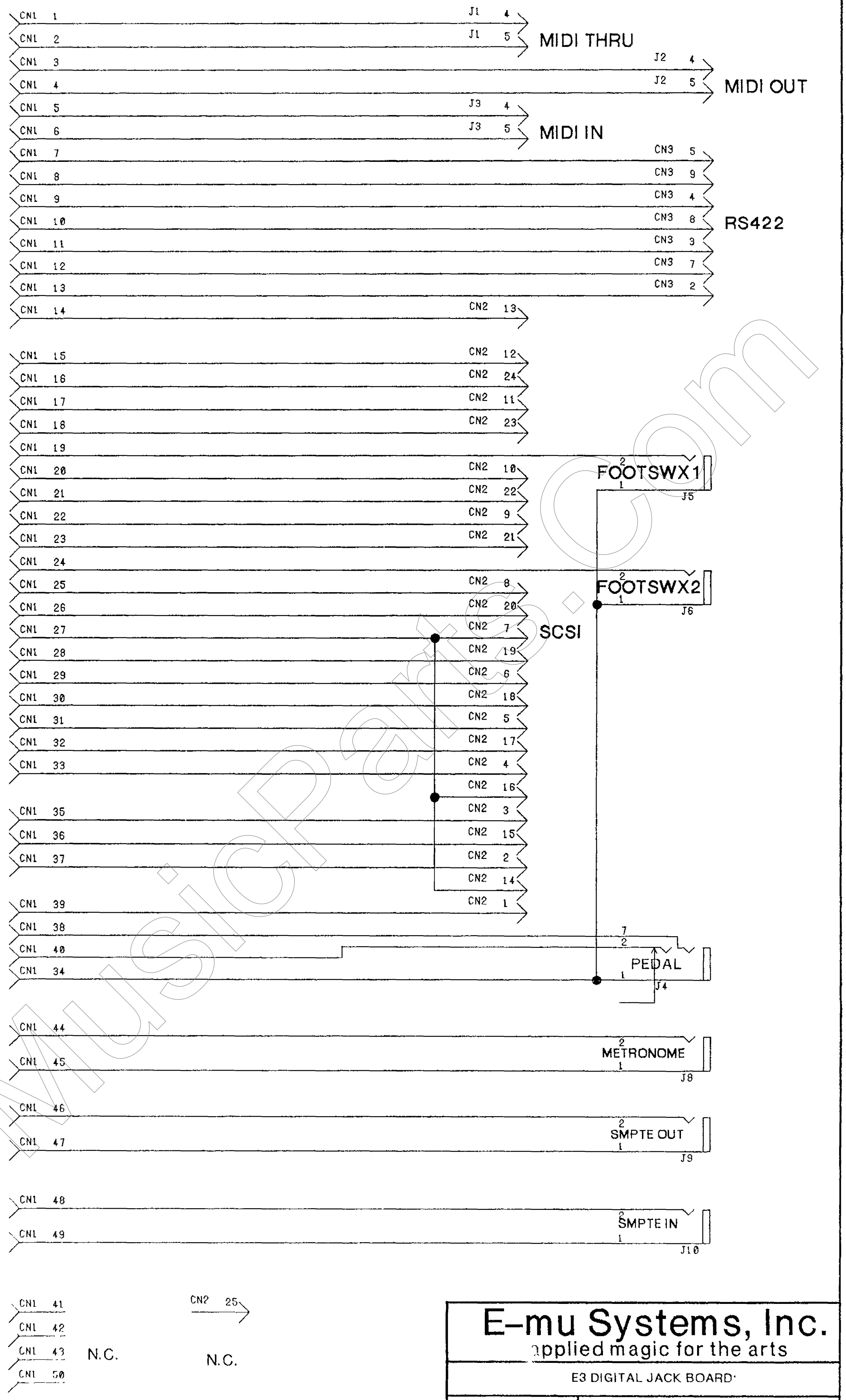


E-mu Systems, Inc.

E3 ANALOG JACK BOARD:

DOC #: SK107

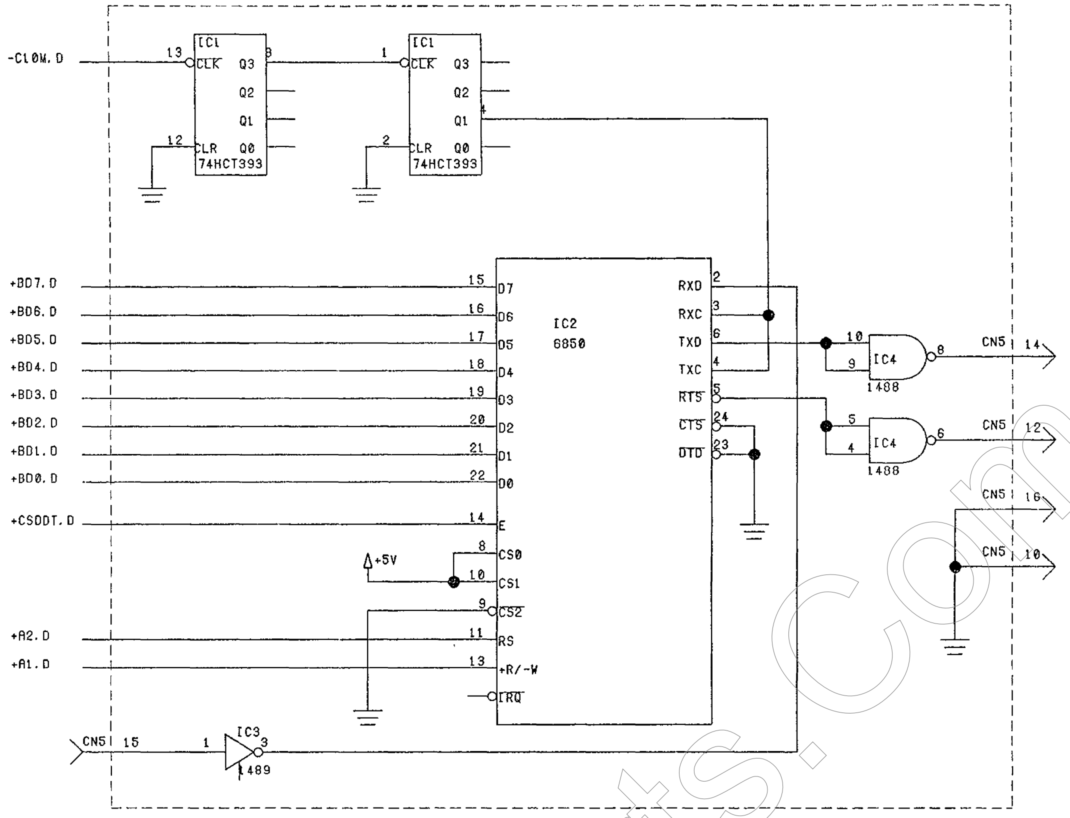
PAGE 2 OF 2



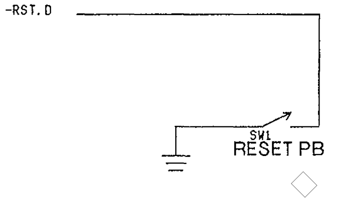
E–mu Systems, Inc.
ıpplied magic for the arts

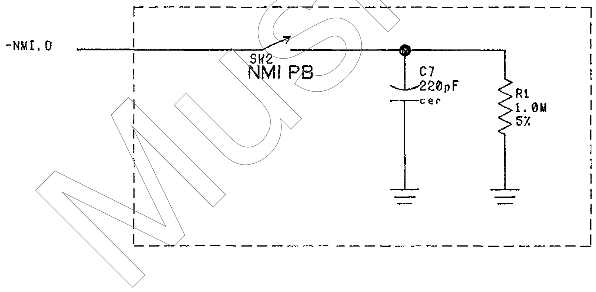
E3 DIGITAL JACK BOARD·

DRAWN BY BFS	DOC #. SK 108	
REV C	DATE 880509	PAGE 1 OF 1



COMPONENTS WITHIN DASHED BOX
ARE NOT INSTALLED ON THE EXTENDER
BOARDS. THESE COMPONENTS ARE USED
FOR IN HOUSE DIAGNOSTICS ONLY.



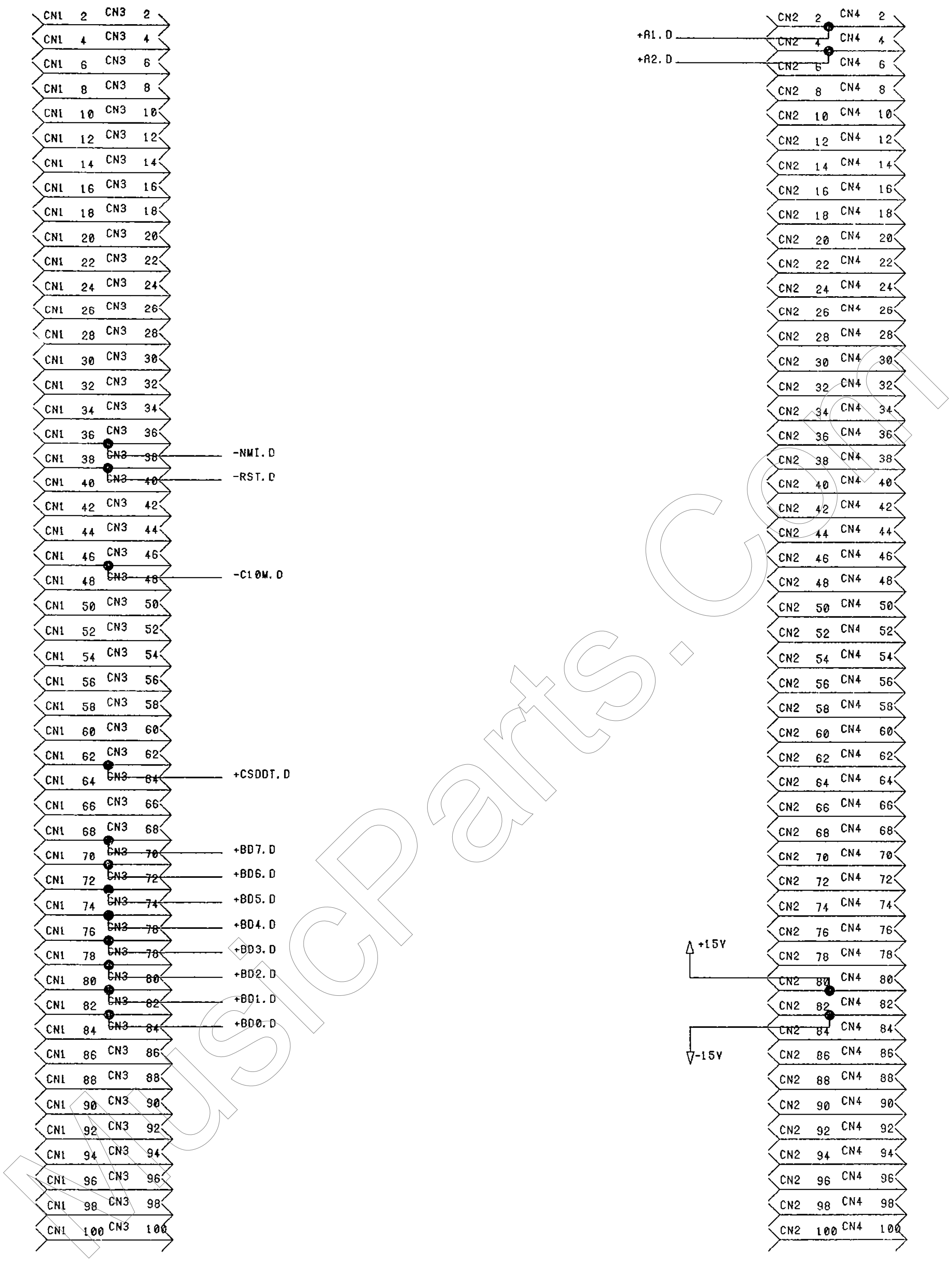


COMPONENTS WITHIN DASHED BOX
ARE NOT INSTALLED ON THE EXTENDER
BOARDS. THESE COMPONENTS ARE USED
FOR IN HOUSE DIAGNOSTICS ONLY.

E–mu Systems, Inc.
applied magic for the arts

E3 DEBUG/EXTENDER: SERIAL PORT & PB'S

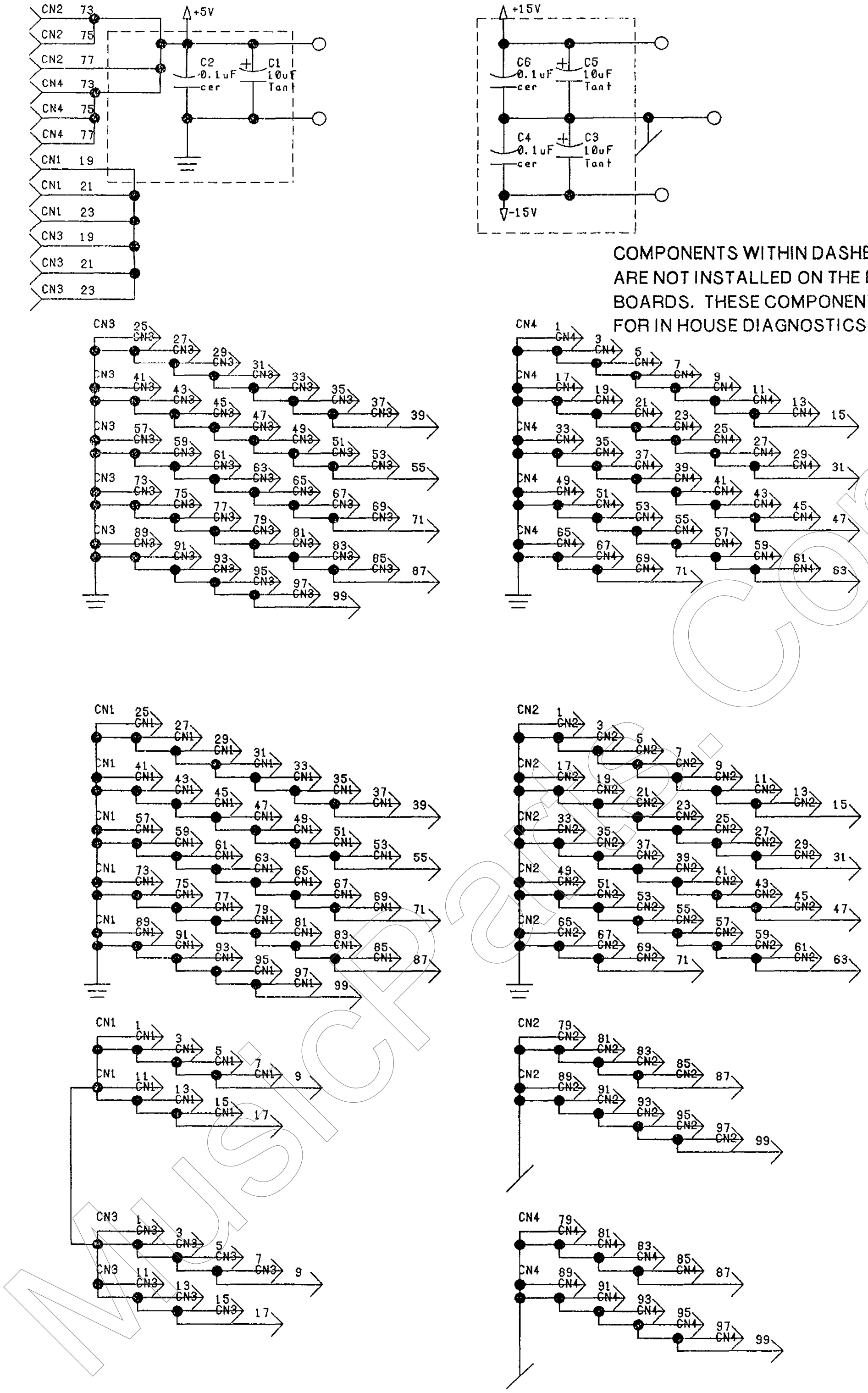
DRAWN BY: DPR	DOC.#: SK112
REV A	DATE 871130



E-mu Systems, Inc.

E3 DEBUG/EXTENDER: EXTDR CONNECTIONS

DOC. #: SK112	Page 2 of 3



COMPONENTS WITHIN DASHED BOX ARE NOT INSTALLED ON THE EXTENDER BOARDS. THESE COMPONENTS ARE USED FOR IN HOUSE DIAGNOSTICS ONLY.

E-mu Systems, Inc.

E3 DEBUG/EXTENDER: POWER CONNECTIONS

DOC. #: SK112	Page 3 of 3

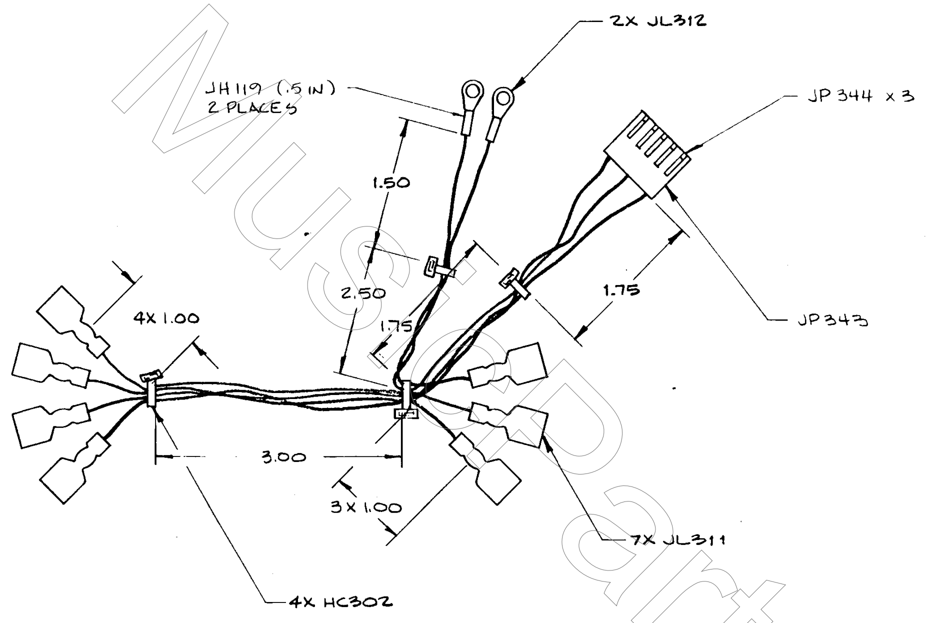
103844

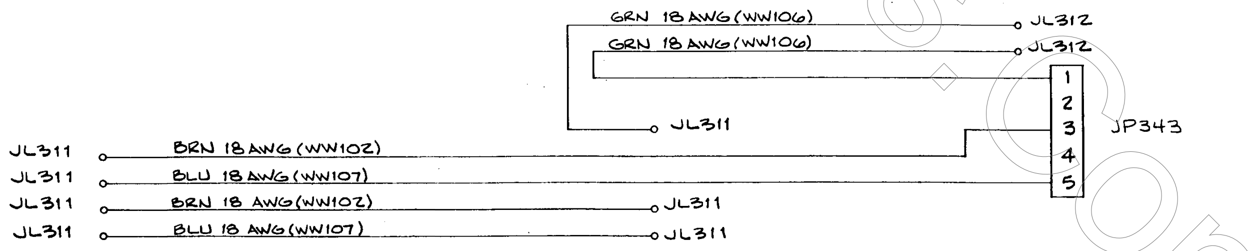
NOTES: UNLESS OTHERWISE SPECIFIED.

REVISIONS

KEY	REV	ECO	DESCRIPTION	APPR'D/DATE
	A	756	INITIAL RELEASE	10/26/87
	B	896	3,00 WAS 1.50	12/31/87

AUG 04 1988





UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES (MM) TOLERANCES ARE:
FRACTIONS ± 1/64 (.4) DECIMALS ±.2 ANGLES ±2°

MATERIAL: NOTED

FINISH: NONE

DO NOT SCALE DRAWING

APPROVALS		DATE
DRAWN	J.D.B.	8/5/87
CHECKED		10/2
APPROVED		10/ /87

E-mu Systems, Inc.
Scotts Valley, CA

ASSEMBLY, HARNESS, CABLE, AC, KEYBOARD, E3

SIZE C | DWG. NO. AG144 | REV. B

SCALE — | SHEET 1 OF 1

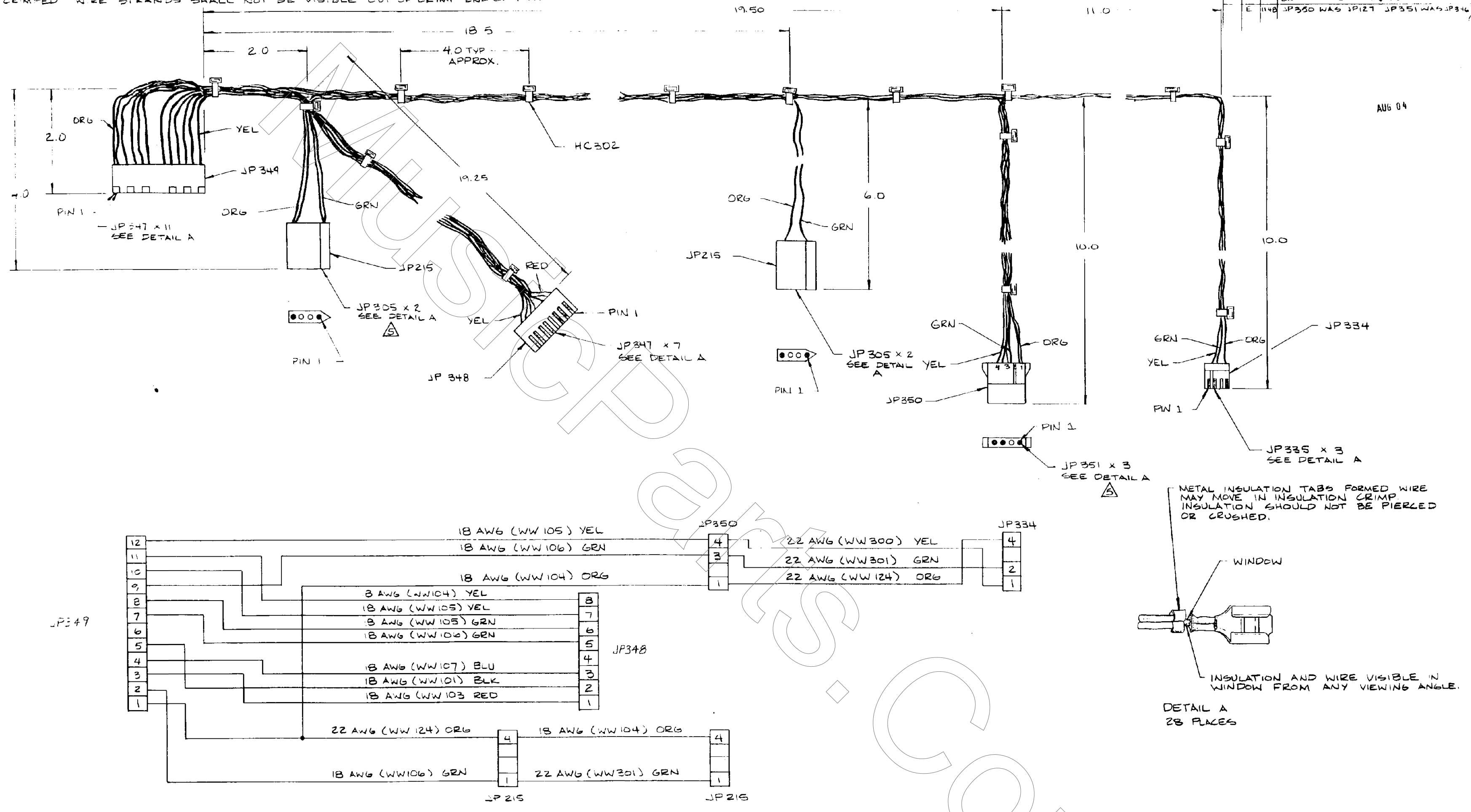
NOTES UNLESS OTHERWISE SPECIFIED

1. FOR CRIMPING OF E-MU PART NUMBER JP335, USE CRIMP TOOL AMP-722569-1.
2. FOR CRIMPING OF E-MU PART NUMBER JP346, USE CRIMP TOOL AMP-90124-2.
3. FOR CRIMPING OF E-MU PART NUMBER JP305 USE CRIMP TOOL MOLEX-HTR 3316 A CHUCK.
4 FOR CRIMPING OF E-MU PART NUMBER JP347 USE CRIMP TOOL MOLEX-HTR-2445-A
5 WHEN CRIMPING A PIN WHERE 2 OR MORE WIRES ARE INSERTED WIRE INSULATION MUST BE CRIMPED WIRE STRANDS SHALL NOT BE VISIBLE OUT OF CRIMP END OF PIN.

REVISIONS

KEY	REV	ECO	DESCRIPTION	APPR / DATE
	A	756	INITIAL RELEASE	7/23/87
	B	1019	JP346 WAS JP128	3/17/88
	C	1109	JP348 WAS JP341; JP347 WAS JP344; JP349 WAS JP342.	
	D	1134	ADDED NOTES SPECIFYING PROPER CRIMPING METHOD, & DETAIL A	6.17
	E	1148	JP350 WAS JP127 JP351 WAS JP346	5.88

AUG 04



UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES (MM)
TOLERANCES ARE:
FRACTIONS ± 1/64 (.4) DECIMALS ±.2 ANGLES ± 2°
MATERIAL
FINISH
DO NOT SCALE DRAWING

APPROVALS		DATE
DRAWN	JMc	7/22/87
CHECKED		7-24 87
APPROVED		7/24/87

E-mu Systems, Inc.
Scotts Valley, CA

ASSY, CABLE HARNESS
DC KEYBD, E3

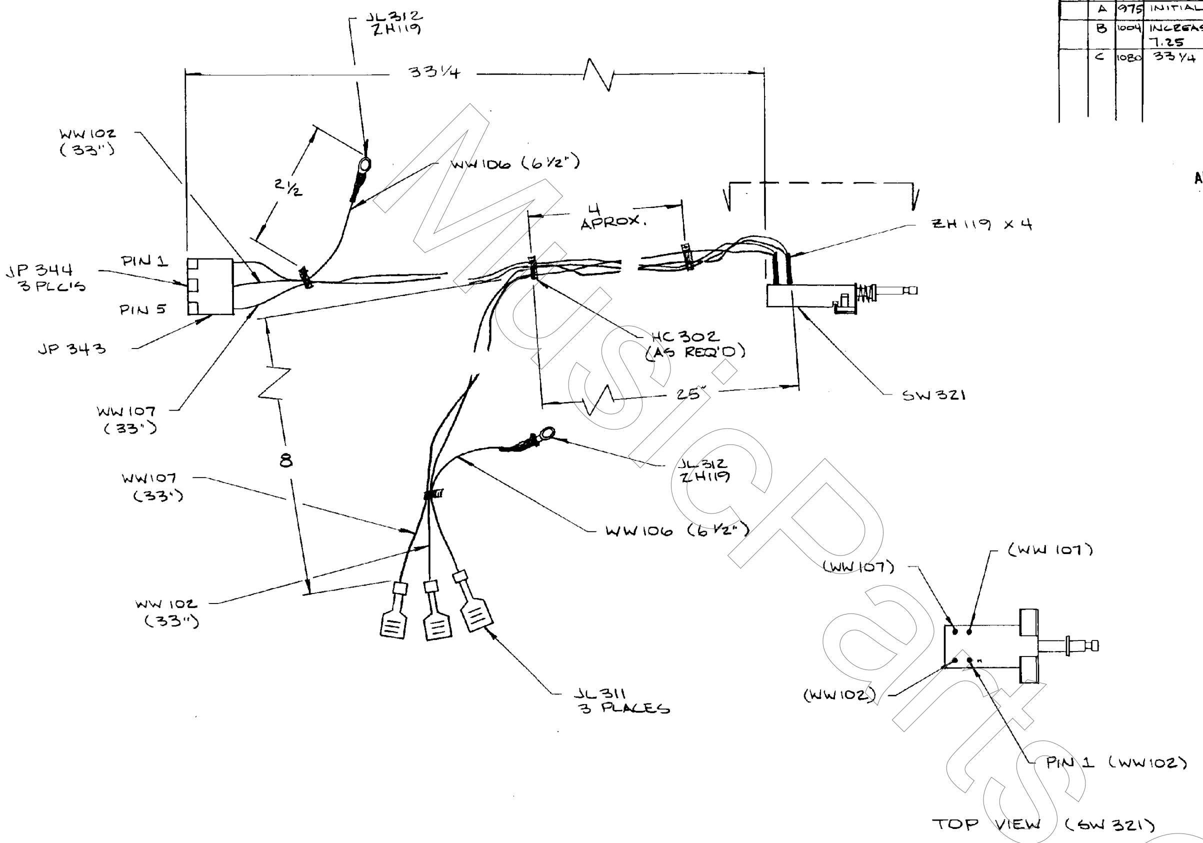
SIZE D DWG. NO A6143 REV E

SCALE NONE SHEET 1 OF 1

NOTES: UNLESS OTHERWISE SPECIFIED.

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR'D/DATE
	A	975	INITIAL RELEASE	2/2/88
	B	1004	INCREASED WIRE LENGTH BY 7.25	2-17
	C	1080	33 1/4 WAS 24 1/2, ADDED 25" DIM	4/21/88

AUG 04 1988



TOP VIEW (SW 321)

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES (MM)
TOLERANCES ARE:

FRACTIONS	DECIMALS	ANGLES
± 1/64 (.4)	± .2	±2°

MATERIAL ———

FINISH ———

DO NOT SCALE DRAWING

APPROVALS		DATE
DRAWN	JMc	1/20/88
CHECKED		1/22/88
APPROVED		2/2/88

E-mu Systems, Inc.
Scotts Valley, CA

ASSY, CABLE HARNESS, AC, RACK, EIII

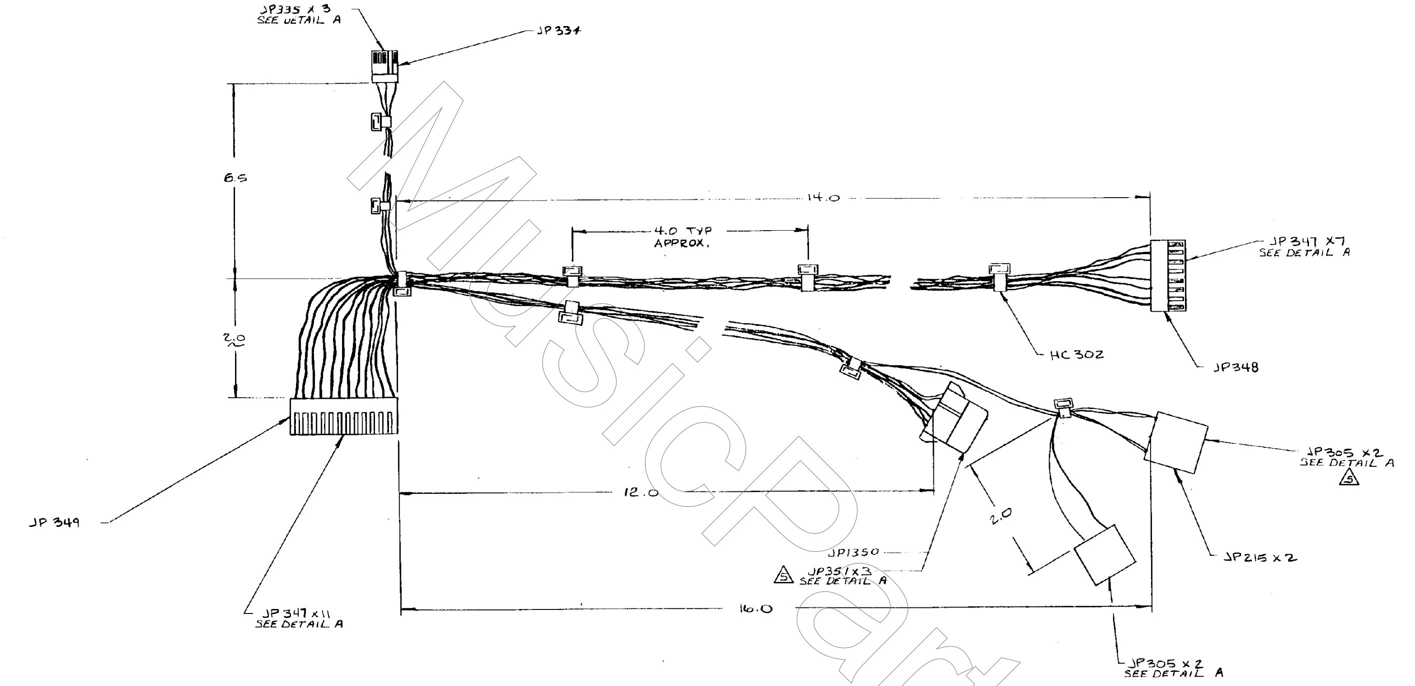
SIZE	DWG. NO.	REV.
C	A6154	C

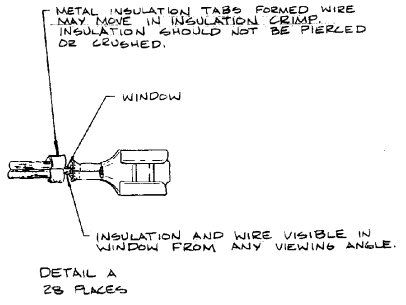
SCALE 1/1 SHEET 1 OF 1

NOTES: UNLESS OTHERWISE SPECIFIED.
1. FOR CRIMPING OF E-MU PART NUMBER JP335 USE CRIMP TOOL AMP-722560-1.
2. FOR CRIMPING OF PART NUMBER JP351, USE CRIMP TOOL MOLEX-HTR1031E.
3. FOR CRIMPING OF E-MU PART NUMBER JP305 USE CRIMP TOOL MOLEX-HTR1031E A CHUCK.

REVISIONS				
KEY	REV	ECO	DESCRIPTION	APPR'D/DATE
	A	962	INITIAL RELEASE	2/2/83
	B	1019	JP346 WAS JP128	3/22/88
	C	1073	REDESIGN, REVISED PER NEW RACK DESIGN	4/7/88
	D	1080	CORRECTED JP342 CONFIGURATION	4/21/88
	E	1109	JP348 WAS JP341; JP344 WAS JP347; JP349 WAS JP342.	6-1
	F	1124	JP350 WAS JP334; JP351 WAS JP335. DIM. 12.0 WAS 10.0	6-10
	G	1134	ADDED NOTES SPECIFYING PROPER CRIMPING METHOD, & DETAIL A. JP334 WAS JP350, JP335 WAS JP351, JP350 WAS JP127, JP351 WAS JP128 (CORRECTED ECO 1124)	8/15

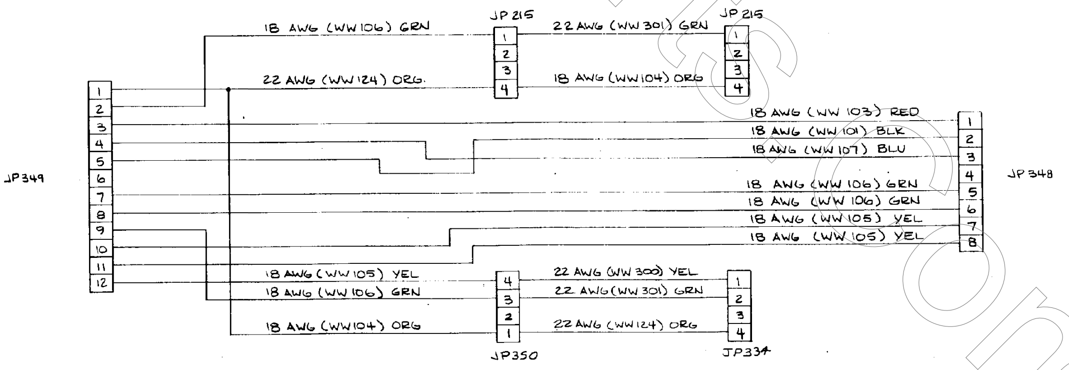
AUG 04 1988





METAL INSULATION TABS FORMED WIRE MAY MOVE IN INSULATION CRIMP. INSULATION SHOULD NOT BE PIERCED OR CRUSHED.
WINDOW
INSULATION AND WIRE VISIBLE IN WINDOW FROM ANY VIEWING ANGLE.

DETAIL A
28 PLACES



5 WHEN CRIMPING A PIN WHERE 2 OR MORE WIRES ARE INSERTED WIRE MUST BE CRIMPED. WIRE STRANDS SHALL NOT BE VISIBLE OUT OF CRIMP END OF PIN.
4. FOR CRIMPING OF E-MU PART NUMBER JP347 USE CRIMP TOOL MOLEX-HTR-2445-A.
NOTES CONTINUED:

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES (MM) TOLERANCES ARE: FRACTIONS ± 1/64 (.4) DECIMALS ± .20 ANGLES			E-mu Systems, Inc. Scotts Valley, CA
MATERIAL	APPROVALS	DATE	ASSY CABLE HARNESS DC RACK E3
	DRAWN JM	9.23.87	
FINISH	CHECKED	7-28-88	SIZE D DWG. NO. AG153 REV 6
DO NOT SCALE DRAWING	APPROVED	9-21-88	SCALE — SHEET 1 OF 1

EIII PARTS LIST

ORDERING PARTS: Parts can be ordered by written order or by phone. When ordering parts, you MUST order by E-mu part number. The minimum parts order is $15.00. Emergency rush orders can usually be sent out the same day if the order is received by 11:00 PST. Parts orders can be placed between the hours of 8:30 am and 5:30 pm PST Monday through Friday. E-mu Customer Service Department (408) 438-1921.

CPU BOARD (AG100)

CAPACITORS	E-MU P/N	QUANTITY
10 µF 25V Alum.	CA325	1
1000 pF 1kV Cer.	CC106	1
.1 µF 50V Cer.	CC335	46
33 pF 50V Cer.	CC336	1
1000 pF 50V Cer.	CC337	1
.01 µF 50V Cer.	CC338	1
.022 µF 50V Cer.	CC339	1
27 pF 50V Cer.	CC350	1
1 µF 25V Tant.	CT315	1

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
4028 CMOS Decoder	IC129	2
LM311 Comparator	II102	1
WD1772 Floppy Controller	II341	1
9637A Differential Recvr.	II348	1
5380 SCSI Interface	II349	1
82530 Serial Controller	II351	1
8254-2 Interval Timer	IM341	1
R6500/11 µProcessor	IM368	1
4464 64k x 4 DRAM	IM373	4
32016-10 CPU	IM378	1
32201-10 Timing Control Unit	IM379	1
32081-10 Floating Point Unit	IM380	1
LS Boot EPROM	IP363	1
MS Boot EPROM	IP364	1
µP RAM PAL	IP365	1
EIII Serial PAL	IP366	1
Interrupt/Wait PAL	IP373	1
Chip Select PAL	IP374	1
74LS138 Decoder	IT311	1
74S74 Dual D Flip-Flop	IT337	1
74HCT138 3 to 8 Decoder	IT357	2
74HCT244 Octal Buffer	IT360	3
74HCT273 Octal Flip-Flop	IT361	3
74HCT393 4-bit Dual Counter	IT364	1
74HCT221 Dual One-Shot	IT365	1
74HCT240 Octal Buffer	IT374	1
74S04 Hex Inverter	IT375	1
74HCT245 Triple Octal Buffer	IT383	3
74HCT373 Octal Latch	IT391	2

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
UA9638 Dual Line Driver	IT400	1
74S158 2 Input Quad Multiplexer	IT401	2
PC900 Opto-Isolator	OE302	1

CONNECTORS	E-MU P/N	QUANTITY
18-pin LP DIP socket	JC105	4
20-pin LP DIP socket	JC106	4
24-pin LP DIP socket	JC107	3
28-pin LP DIP socket	JC308	3
40-pin LP DIP socket	JC309	3
48-pin LP DIP socket	JC329	1
Straight Ribbon Header	JR307	1
Straight Ribbon Header	JR308	1
50-pin Dual Row Rib. Header	JR311	2
60-pin Dual Row Rib. Header	JR316	1

RESISTORS	E-MU P/N	QUANTITY
330x8 1/8W Res. Network	RN320	2
220/330x8 1/8W Res. Network	RN322	1
22k x 9 1/8W Res. Network	RN323	2
150Ω 1/4 W	RR102	5
470Ω 1/4 W	RR106	1
4.7MΩ 1/4 W	RR132	1
100Ω 1/4 W	RR301	2
330Ω 1/4 W	RR303	1
1KΩ 1/4 W	RR305	9
4.7KΩ 1/4 W	RR307	2
6.8KΩ 1/4 W	RR308	1
10KΩ 1/4 W	RR309	5
100KΩ 1/4 W	RR318	1
9.1KΩ 1/4 W	RR321	1
20KΩ 1/4 W	RR323	2
1MΩ 1/4 W	RR325	1
680Ω 1/4 W	RR343	1
270Ω 1/4 W	RR347	1
10Ω 1/4W Conformal	RR364	8

MISCELLANEOUS	E-MU P/N	QUANTITY
EIII CPU PCB	PC338	1
Nylon Card Ejector	HM007	2
1N914 Signal Diode	DD301	9
20MHz Crystal	ZX304	1
16MHz Crystal	ZX305	1

OUTPUT BOARD (AG101)

CAPACITORS	E-MU P/N	QUANTITY
3.3 µF 50V Alum. Elec.	CA327	16
.1 µF 50V Cer.	CC335	82
15 pF 50V Cer.	CC340	32
47 pF 50V Cer.	CC342	32

CAPACITORS	E-MU P/N	QUANTITY
68 pF 50V Cer.	CC343	32
39 pF 50V Cer.	CC344	32
10 pF 50V Cer.	CC345	16
100 pF 50V Cer.	CC346	10
.01 µF 50V Cer.	CC348	2
.1 µF 50V Cer.	CC349	80
2200 pF 33V Poly.	CP317	16
330 pF 33V Poly.	CP318	16
10 µF 25V Tant.	CT317	26

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
SSM2300 Sample/Hold Demux	IC347	10
F-Chip FIR	IC350	16
74HCT4053 Analog Mux	IC353	16
PCM53JGI 16-bit DAC	II347	16
AD6012 12-bit DAC	II350	2
7812 +12V 1A Regulator	IL110	2
7905 -5V 1A Regulator	IL112	2
7805 +5V 1A Regulator	IL307	2
LF356 BiFET Op Amp	IL301	18
5532 Dual Op Amp	IL326	16
CEM3387 VC Signal Processor	IL337	16
74HCT138 3 to 8 Decoder	IT357	4
74HCT151 8-bit Mux.	IT358	2
74HCT273 Octal Flip-Flop	IT361	4

CONNECTORS	E-MU P/N	QUANTITY
20-pin LP DIP Socket	JC106	18
24-pin LP DIP Socket	JC107	16
16-pin LP DIP Socket	JC311	10
48-pin LP DIP Socket	JC329	16
20-pin Dual Row Header	JR317	4
20-pin Dual Row Rt Angle Header	JR320	4

RESISTORS	E-MU P/N	QUANTITY
10KΩ x 8 Res. Net.	RN312	6
56KΩ x 8 1/8W Res. Net.DIP	RN341	2
6.8KΩ x 8 1/4W Res. Net. DIP	RN342	2
15KΩ Precision Res. (1%) 1/4W	RP107	2
1.24KΩ Precision Res. (1%) 1/4W	RP357	2
47 KΩ 1/4W	RR122	16
1 KΩ 1/4W	RR305	2
10 KΩ 1/4W	RR309	32
39 KΩ 1/4W	RR317	32
2.2 KΩ 1/4W	RR337	16
2.7 KΩ 1/4W	RR360	16

MISCELLANEOUS	E-MU P/N	QUANTITY
EIII Output PCB	PC336	1
1N914 Signal Diode	DD301	22
Card Ejector	HM007	4
4-40 Kepf Nut	HN304	6
4-40 3/8" Screw	HS302	6

MISCELLANEOUS	E-MU P/N	QUANTITY
Black Alum. Heatsink	ZE324	6
Output Board Fisch Paper	ZP330	1

MICROCONTROLLER BOARD (AG113)

CAPACITORS	E-MU P/N	QUANTITY
.1µF 50V Cer.	CC335	59

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
74HCT125 Quad Buffer	IC351	1
1M x 8 DRAM 120nS SIMM	IC356	4 or 8
256K x 8 DRAM 150nS SIMM	IM377	16
2149 1K x 4 Static RAM	IM339	7
256K Address PAL	IP367	1
Sound 1 PAL	IP369	1
Sound 2 PAL	IP370	1
256K RAS PAL	IP371	1
µController Memory PAL	IP372	1
1M RAS PAL	IP376	1
1M Address PAL	IP377	1
74S244 Octal Buffer	IT332	1
74S240 Octal Buffer	IT333	7
74S74 Dual D Flip-Flop	IT337	2
74S37 Quad NAND Buffer	IT339	1
74S00 Quad 2 Input NAND	IT342	1
74HCT74 Dual D Flip-Flop	IT355	1
74HCT244 Octal Buffer	IT360	4
74HCT374 Octal Flip-Flop	IT362	7
74HCT164 Ser./Paral. Shift Reg.	IT370	2
74HCT139 Dual Decoder	IT387	1
74HCT373 Octal Latch	IT391	5
74S158 2-Input Quad Mux	IT401	2
74ALS161 4-bit Binary Counter	IT405	2
74ALS174 Hex D Flip-Flop	IT406	1
74F283 4-bit Full Adder	IT407	6
74ALS374 Octal D Flip-Flop	IT409	9

CONNECTORS	E-MU P/N	QUANTITY
18-pin LP DIP Socket	JC105	7
20-pin LP DIP Socket	JC106	3
SIMM Socket (RAM)	JC331	16
24-pin DIP Socket	JC333	2

MISCELLANEOUS	E-MU P/N	QUANTITY
Microcontroller PCB	PC337	1
Card Ejector	HM007	2

ADC BOARD (AG106)

CAPACITORS	E-MU P/N	QUANTITY
3.3µF 50V Alum. Elect.	CA327	2
47µF 16V Alum. Elect.	CA329	6
.1µF 50V Cer.	CC335	42
1000pF 50V Cer.	CC337	9
15pF 50V Cer.	CC340	1
4.7pF 50V Cer.	CC341	1
10pF 50V Cer.	CC345	8
.33pF 50V Cer.	CC351	2
22pF 50V Cer.	CC353	1
560pF 33V Poly.	CP316	2
10µF 16V Tant.	CT316	6

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
4051 Analog Mux.	IC208	1
74HCT4053 Analog Mux/Demux.	IC353	4
74HCT299 8-bit Shift Reg.	IC355	2
CX20018 Stereo 16-bit A/D Conv.	II354	1
LF353 Dual BIFET Op-Amp	IL101	1
7905 -5V 1A Regulator	IL112	1
LF356 BIFET Op-Amp	IL301	5
TL084 Quad BIFET Op-Amp	IL302	2
dbx2155 VCA	IL330	2
15K Anti-Alias Filter	IL335	2
20K Anti-Alias Filter	IL336	2
-5V 100mA Regulator	IL338	1
+5V 100mA Regulator	IL339	1
Sony 1 ADC PAL	IP368	1
Sony 2 ADC PAL	IP375	1
74HCT04 Hex Inverter	IT352	1
74HCT32 Quad 2-Input AND	IT354	1
74HCT174 Hex D Flip-Flop	IT359	1
74HCT273 Octal Flip-Flop	IT361	1
74HCT42 BCD->Decimal Decoder	IT366	1
74LS293 4-bit Counter	IT399	1

RESISTORS	E-MU P/N	QUANTITY
10KΩ Precision Res. (1%)	RP106	2
36.5KΩ Precision Res. (1%)	RP354	1
3.01KΩ Precision Res. (1%)	RP355	2
2KΩ Precision Res. (1%)	RP356	6
2.43KΩ Precision Res. (1%)	RP358	2
9.09KΩ Precision Res. (1%)	RP359	2
24.3KΩ Precision Res. (1%)	RP361	2
90.9KΩ Precision Res. (1%)	RP362	2
470Ω 1/4W	RR106	2
1KΩ 1/4W	RR305	12
2KΩ 1/4W	RR306	2
10KΩ 1/4W	RR309	13
22KΩ 1/4W	RR313	6
33KΩ 1/4W	RR316	4

RESISTORS	E-MU P/N	QUANTITY
100KΩ 1/4W	RR318	10
20KΩ 1/4W	RR323	1
3.9KΩ 1/4W	RR355	2
51Ω 1/4W	RR361	2
5KΩ Trim Pot 1-Turn	RT306	2
50KΩ Trim Pot 1-Turn	RT308	4

MISCELLANEOUS	E-MU P/N	QUANTITY
ADC PCB	PC335	1
1N914 Signal Diode	DD301	13
2835 Schottky Barrier Diode	DD315	2
Card Ejector	HM007	2
4-40 Kepf Nut	HN304	1
4-40 1/4" Screw	HS301	1
20-pin LP DIP Socket	JC106	2
3-Hole Thermal Insulator	ZE329	6
220µH Inductor, Axial	ZI303	1
.270µH Inductor, Axial	ZI304	1
84.6MHz Crystal	ZX306	1

ANALOG JACK BOARD (AG103)

CAPACITORS	E-MU P/N	QUANTITY
47 µF 16V Alum.	CA329	2
220 pF Cer.	CC104	2
.1 µF 50V Cer.	CC335	2
100 pF 50V Cer.	CC346	2
470 pF 100V Cer.	CC352	2

INTEGRATED CIRCUITS	E-MU P/N	QUANTITY
5532 Op Amp	IL326	1
8212 Prog. Volt. Ref.	IL334	1

CONNECTORS	E-MU P/N	QUANTITY
1/4" Stereo Phone Jack	JA311	20
8-pin LP DIP Socket	JC102	1
20-pin Ribbon Header	JR320	4

RESISTORS	E-MU P/N	QUANTITY
10k x 4 Res.Net. SIP	RN334	8
470 x 4 Res.Net. SIP	RN339	8
470Ω 1/4 W 5%	RR106	2
100Ω 1/4 W 5%	RR301	2
10KΩ 1/4 W 5%	RR309	3
22KΩ 1/4 W 5%	RR313	4
9.1KΩ 1/4 W 5%	RR321	1
3KΩ 1/4 W 5%	RR327	1
2.7KΩ 1/4 W 5%	RR360	2

MISCELLANEOUS	E-MU P/N	QUANTITY
Analog Jack PCB	PC339	1
2N4121 PNP Trans.	QQ101	1

MISCELLANEOUS	E-MU P/N	QUANTITY
5V Relay	SW315	1

DIGITAL JACK BOARD (AG104)

CONNECTORS	E-MU P/N	QUANTITY
1/4" Mono Phone Jack	JA301	5
1/4" Jack TP Closed, Ring Open	JA309	1
5-pin DIN Connector	JI302	3
9-pin D-Conn. Male	JI309	1
25-pin D-Conn. Female	JI312	1
50-pin Ribbon Header	JR301	1
50-pin Dual Row Header	JR311	1

MISCELLANEOUS	E-MU P/N	QUANTITY
Digital Jack PCB	PC340	1
Support Bracket	EM368	1
4-40 Kepf Nut	HN304	4
4-40 1/2" Screw	HS118	2
4-40 3/8" Screw	HS302	2
#4 Insul. Nylon Washer	HW210	1

FRONT PANEL BOARD (AG129)

PART	E-MU P/N	QUANTITY
Front Panel PCB	PC341	1
CPU -> Front Panel Cable	AG136	1
Black Round Pushbutton Cap	EP343	37
1/8" Shaft Knob	HK324	1
Knob Cap	HK325	1
2-56 Hex Nut	HN139	4
3 x 10MM Phill. Screw	HS325	6
2-56 x 3/8 Phil. Screw	HS355	4
.125 #6 Spacer	HS369	6
#4 1/6 x 3/8 Nyl. Washer	HW323	6
#2 Lock Washer	HW328	4
LED Socket	JC332	17
Red Rect. LED	LP312	17
20 x 4 Char. LCD	LP313	1
Linear 10kΩ Slider	RC311	3
Linear 10kΩ Pot	RC315	1
SPST Button	SW313	37
14 Conductor Flexstrip	WW313	1
LCD Power Supply	ZV300	1

PITCH WHEEL ASSEMBLY (AG146)

PART	E-MU P/N	QUANTITY
6-hole Molex Conn.	JP131	1
4-hole Molex Conn.	JP217	1
Female Pin 18-22 AWG	JP305	3

PART	E-MU P/N	QUANTITY
Male Pin 14-20 AWG	JP306	3
Female Pin 14-20 AWG	JP345	1
Linear 10kΩ Pot	RC101	2
White 22G StrandedWire	WW112	-
Orange 22G StrandedWire	WW124	-
Violet 22G StrandedWire	WW125	-
Green 22G StrandedWire	WW301	-
Blue 22G StrandedWire	WW304	-
1/8 Heatshrink	ZH119	-
Wheel Bracket	EM373	2
EIII Plastic Wheel	EP341	2
Torsion Spring	HB300	1
3/8 -32 Nut	HN311	4
3/8" Lock Washer	HW312	4
3/8 Retaining Ring	HW333	1

EIII RIBBON CABLES

CABLE	E-MU P/N	QUANTITY
Output to Analog Jack Board	AG134	4
CPU to Hard Drive	AG138	1
CPU to Digital Jack Board	AG139	1
CPU to Keyboard	AG140	1
CPU to Floppy Drive	AG141	1
DC Power Harness	AG143	1
AC Power Harness	AG144	1

EIII CARD CAGE (AG157)

PART	E-MU P/N	QUANTITY
Upper Card Cage Bracket	EM374	1
Lower Card Cage Bracket	EM380	1
Card Cage Spacer .750	HS375	4
Card Cage Spacer .375	HS376	4
Card Cage Spacer .250	HS377	4
Card Cage Spacer .125	HS378	8
6-32 x 3/8 Phil Screw	HS382	8
8-32 x 3/8 Phil Screw	HS383	6
6-32 x 3/8 UC Phil Screw	HS392	8
6-32 x 1/4 Self-Tap	HS393	1
100-pin Edge Conn.	JI311	10
Molex Conn.	JP111	1
8-pin Male Power Conn.	JP325	1
EIII Mother Board	PC344	1
Mother Board Fisch Paper	ZP329	1
Rubber Bumber	ZP330	4

CHASSIS (Keyboard)

PARTS	E-MU P/N	QUANTITY
KYBD Scrambler PCB	AG164	-
Fan Baffle	EM361	1
EIII Keyboard Chassis	EM366	1
Jackboard Support Bracket	EM368	1
Fan Mounting Bracket	EM369	1
Floppy Drive Mounting Bracket	EM370	1
HD Mounting Bracket Type 1	EM392	1
HD Mounting Bracket Type 2	EM396	1
Emulator III Plastic Housing	EP327	1
Cardcage Card Support	EP340	1
Cable Stickdown	HC121	8
Mini Tie-wrap	HC302	12
Ribbon Cable Clamp	HC312	13
LCD Angle Knob Cap	HK325	1
Slider Knob	HK327	3
Fan Mount Washers	HN121	6
Floppy Mounting Bracket Nuts	HN308	4
Audio Jack Nuts	HN319	20
Rear Chassis Screws	HS325	4
Fan Mount Screws	HS336	2
Front Panel Board Screws	HS353	23
Bottom Chassis Screws	HS371	12
Line Filter Mounting Screws	HS374	2
6-32 x 1 1/4" Screw	HS381	6
Rubber Foot Screws	HS383	5
KYBD Mounting Screws	HS384	6
Card Support Screw	HS385	1
Rear Chassis Lock Washers	HW325	4
Audio Jack Washers	HW335	20
KYBD Mounting Washers	HW336	6
Male Connector Lug .187 x .020	JL314	1
Female Connector Lug.187 x .020	JL315	1
Ribbon Header (Scrambler)	JR308	-
110/230V Selector Switch	SW308	1
Power Switch DPST	SW309	1
12V DC Fan	ZE325	1 or 2
5 x 20mm 2A Slo-Blo Fuse	ZF307	2
AC Line Filter	ZI302	1
5 Octave Keyboard w/Pressure	ZK324	1
Front Panel Label	ZL370	1
Caution Label	ZL350	1
Caution Label (Electric Shock)	ZL381	1
3.5" 1/2 Height Floppy Disk Drive	ZM336	1
40 MByte Hard Disk Drive	ZM338	1
Rubber Feet	ZR308	5
Switching Power Supply	ZV305	1

PARTS	E-MU P/N	QUANTITY
110/220 Selector Assembly	AG161	1
Rack Enclosure	EM375	1
Rack Top Panel	EM376	1
Rack Front Panel	EM377	1
Floppy Drive Mounting Plate	EM394	1
HD Drive Mounting Plate	EM395	1
Cardcage Card Support	EP340	1
Tie Wrap Stickdown	HC121	8
Tie Wrap	HC302	8
Ribbon Cable Clamp	HC312	10
Fan Mounting Nuts	HN121	8
Audio Jack Nuts (Digital)	HN135	6
Front Panel Mounting Nut	HN308	5
Audio Jack Nuts (Analog)	HN319	20
Rubber Foot Screws	HS122	4
HD Plate Mounting Screw	HS336	1
Mounting Screws	HS353	26
Top Panel Screws	HS368	6
Card Support Screw	HS385	1
Fan Guard Mounting Screws	HS389	8
AC Filter Mounting Screws	HS391	2
Audio Jack Washers (Digital)	HW322	6
Audio Jack Washers (Analog)	HW335	20
Male Connector Lug	JL314	2
Female Connector Lug	JL315	2
Molex Housing	JP217	2
Molex Pins	JP306	4
Cooling Fan	ZE326	2
Fan Guard	ZE330	2
Floppy Disk Drive	ZM339	1
40MByte HD Drive	ZM341	1
Rubber Feet	ZR308	4
AC Filter/Receptacle	ZI302	1
MANUALS, PEDAL, FOOTSWITCHES		
EIII Reference Manual	AG165	1
EIII Tutorial Manual	AG188	1
EIII Foot Switch	SW318	1
EIII Foot Pedal	AG175	1
US Power Cord	WC307	1
Western Europe Power Cord	WC308	1
10" MIDI Cable	WC309	1
SHIPPING BOX (KYBD)		
KYBD Shipping Box	ZS377	1
KYBD Foam Tray Bottom	ZS378	1
KYBD Foam Tray Top	ZS379	1
KYBD Shipping Insrt	ZS380	1
SHIPPING BOX (Rack)		
Rack Shipping Box	ZS381	1
Rack Shipping Insert	ZS382	1
Rack Shipping Tray	ZS391	1

EMU CHANGE ORDERS

E-mu Change Orders or ECOs document hardware or software design changes to an E-mu product in order to enhance or improve its functionality. Some ECOs are installed on an "as needed" basis, perhaps to compensate for the timing of a particular part or some other abnormality. Other ECOs are mandatory updates and should be installed on all units. We have listed the important ECOs to date. The E-mu Service department will keep you updated of any further ECOs if they become necessary. All mandatory ECOs are, of course, covered under warranty.

When checking a board for ECOs, keep in mind that the modification may have been installed in a different configuration than the one shown. For instance, the ECO may have been performed on the other side of the circuit board. We have seen units come back with double ECOs! These ECOs apply only to the Keyboard Emulator Three and most units will have all the ECOs performed at the factory.

ECO CHECKLIST

PROMS and PALs

Boot PROMS (IC 3,4)..	Rev B.
CS PAL (IC 27)...	Rev B.
Serial PAL (IC 32)..	Rev B.

ECOs	REASON	TO IDENTIFY
ECO #921 Add rubber bumpers to motherboard.	Prevent shorts to chassis	Presence of rubber bumpers
ECO #1025 Capacitively couple input to ADC.	Prevent DC offsets from affecting sampling performance.	Kludged caps and resistors on ADC board.
ECO #1069 Change mounting of HD	Eliminate torsion on HD	HD mounting plate.
ECO # 666 Change routing of panel ribbon cable.	Prevent damage from front panel board.	Correct cable routing per mechanical procedures.
ECO #1093 Remove +5V from SCSI connector.	Prevent conflicts with certain HD units.	Absence of wire to SCSI connector on CPU board.
ECO #1109 Change DC power harness to double-wipe contacts.	Decrease resistance at connectors.	Bifurcated contacts at power supply connector.
ECO #1131 CPU board mod prevents MIDI crash bug.	CS from IC1 not qualified R/W resulting in glitches in addr lines.	See CPU board ECO diagram for details.

EIII Version 1.13 Software

March 17, 1988

Newly implemented:

DIGITAL:	gain change including normalization taper reverse section stereo <-> mono sample calculator--calculates loop size, etc.
MIDI:	overflow mode midi time code song select echo thru all notes off reset all controllers
SEQUENCER:	transpose segment scale velocity erase all sequences

Functional enhancements:

Save Bank:
"Overwrite?" message appears when saving over an existing bank.

Sample module:
Newly recorded samples are now placed on the keyboard automatically (if not already placed). The default range can be adjusted from 1 to 24 semitones, or white keys only. Empty samples can also be placed before sampling.

Sampling can be initiated from the "setup" screen (#5) by pressing 7 or 8.

Digital module:
The left/right cursors now locate zero crossings when selecting start and end points for loop, cut, taper, etc.

Sequencer:
Step edit now displays & edits MIDI controllers, and also displays notes as they are being recorded

Step edit allows an erased event to be pasted anywhere in the segment by pressing the 'erase' switch twice.

Track mix is now saved with each segment. Also, mix now affects notes currently playing.

Misc:

Clock is displayed in lower right corner during long functions.

Left and right arrows are displayed on screens with multiple pages.

Recalibration screen shows pot values prior to calibrating.

Filter CV's and microcontroller are turned off during disk operations.

Major bug fixes:

An HD file system bug in 1.01 caused problems in the load single preset / zone / sample / segment / song routines for certain banks. The data in these banks should now load properly with 1.13.

A bug in the sequencer's memory manager caused certain sequences to either "run forever" or hang the machine. Banks which contain sequences created with 1.01 may continue to have similar problems. The "Erase All" function will clear all sequences and restore a bank's sequencer variables to a known state. Sequences may then be loaded individually from the hard disk.

A sequencer bug that didn't let you copy and then paste or mix to another segment.

JJ/BA

EIII Version 1.21 Software

June 15, 1988

This software version contains numerous enhancements not found in version 1.13. It contains many new features as well as several bug fixes. Following is a comprehensive list of these exciting new features. To install 1.21 on your EIII, do not perform an "Erase Software". Simply "Copy Software" from floppy to Hard Disk (follow the instructions on the disk label). Once you have copied version 1.21 to your hard disk and rebooted, be sure to copy your software and "system" from your EIII's Hard Disk back to a floppy disk. This new floppy which will have all of your EIII's trims and calibrations stored on it should be labeled your "Master System Disk". Use this disk in case you need to reformat your hard disk or restore calibrations.

New features

Stack Mode - (Preset Management 7).
Stack those presets!

Velocity Switch Level - (Preset Manage 8).
Switch level is now programmable per preset.

Disable Loop - (Analog Process 1).
Loops defined in Digital Process may be turned "Off" in user definable zones per preset.

Disable Stereo Side - (Analog Process 1).
The user can now define zones in which stereo samples will play either only the left side or the right side of the sample. Mono samples will remain unaffected.

Change Sample #- (Analog Process 1).
The sample(s) which are assigned to a zone can now be quickly reassigned to any other sample number (including an empty sample #) without affecting VCA, VCF, and other Analog Process parameters. **Select your zones carefully!** If a zone is selected which contains more than one sample, all samples in the zone will be reassigned to the new sample number once a change is made.

Change Original Key - (Analog Process 1).
The original key assignment of all samples within a zone may now be quickly changed. The same precautions which apply to "Change Sample #" apply here also.

Scrub Wheel - (Range of Scrub is set in Digital Process 1).
While in Digital Process, the left wheel now functions much like the varispeed on a tape machine. This feature has been added to enhance the operation of all functions in the Digital Process module which require the location of specific points within samples.

Zero Crossing Threshold - (Digital Process 1).
The sensitivity of Auto Truncation and Dig. Process Zero Crossing are now user definable from a

range of -96db up to -30db, with -96db being the most sensitive setting.

Left <-> Right- (Digital Process 8, Submodule 7).
This digital process function can either swap the left and right sides of a stereo sample, or move a mono sample from its current position (from the left or right side) to the other side.

Ping Pong- (Digital Process 8, Submodule 8).
This Digital Process function can be best conceptualized when viewed as a "Symmetrical Pan" operation. It pans each side of a stereo sample towards and back away from its complimentary side. "Volley" specifies the number of times the signal pans. "Swing" determines the intensity of the pan, and the waveform chosen for "Curve" designates the shape of the pan.

Digital Delay- (Digital Process 8, Submodule 9).
A very basic digital delay has been implemented. The user can specify the portion of the sample file that goes through the delay line, the delay time, mix amount, and feedback "On" or "Off". The delayed signal gets summed in with the original signal. **Caution:** When using this function clipping may occur if the original signal is already normalized to full amplitude. It is recommended that digital attenuation be used prior to using the digital delay.

Mount Drives - (Master 7, Submodule 1).
If the EIII is not responding to a drive, the drive needs to be "Mounted." This function replaces the "Install Drive" function used in 1.13 software.

MIDI Globals - (Master 8, Submodule 7).
There is now a global override for the MIDI parameters which often vary from preset to preset.

Also New In 1.21:

SAMPLING

New sample rates of 44,053 Hz and 33,333 Hz replace the former sampling rates of 44,100 Hz and 33,100 Hz. Old samples will play back the same as in former versions of software.

MIDI

Song Select (transmit) has been implemented.

Song Position Pointer (transmit and receive) has been implemented.

MIDI note-on response is now faster and more consistent.

Local Control On/Off has been implemented.

SEQUENCER

Copy/paste now defaults to the first empty Song/Segment.

Erase track resets track volume to max.

Fast-forward/rewind speed now increases after the switch is held down.

Frames per beat fraction is now expressed in 1/8th frames.

Sequencer is now more reliable and fun to use.

SMPTE

Chase performance has been greatly improved; most sequences should now lock in under 3.5 seconds. Dropout tolerance has also been improved. The last minute of SMPTE time (>=23:59:00:00) is now interpreted as "negative time" to accommodate those who are using zero as a start time.

Misc.

Improved autocorrelation.

You can now paste to one side of a stereo sample.

Increased volume trim range.

Cleaned up ADC trims.

Note :

Although our records show that no production EIIIs should be affected, those made in December '87 or January '88 potentially have a compatibility problem with version 1.21. If you think your machine was manufactured around that time, backup your hard disk before installing new software. If you see "Drive Not Formatted" or "SCSI Hardware Error" messages after rebooting from the hard disk, you will have to reformat the hard disk and reinstall version 1.13 to make your EIII operational. You should then get in touch with the E-mu service department and make arrangements to have your instrument updated to current hardware revisions.

If you are using an 8 Meg EIII, do not use the "Interactive Backup/Restore" function (Master 7, Submodule 8). Instead, use the "Automatic Backup/Restore" (same Submodule) or simply load and save banks individually.

EMULATOR III 8 MEGABYTE MEMORY UPGRADE

Materials needed:
EIII 8 Megabyte update kit (E-mu PN 6084 or 6085)
6084 Contains:
(1) Memory expansion instructions (you're reading them)
(8) 1 Megabyte SIMMs (IC356)
(1) RAS PAL (IP376)
(1) Addr. PAL (IP377)

6085 Contains:
(1) Memory expansion instructions
(4) 1 Megabyte SIMMs (IC356)

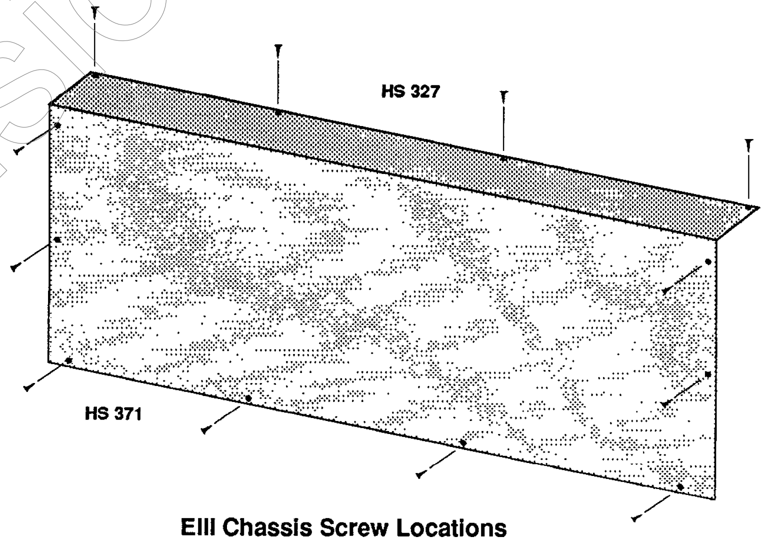
Tools needed:
Phillips screwdriver

▲ **Before you Begin**
Check the unit out thoroughly to make sure the unit is completely functional.

Opening the EIII (Keyboard)

1. Remove the (8) screws (HS 371) from around the perimeter of the bottom panel of the EIII and set aside in a safe place.

2. Remove the (4) screws (HS 327) from the top edge of the back panel. Set these screws aside in the safe place.



EIII Chassis Screw Locations

EIII MEMORY UPGRADE
(continued)

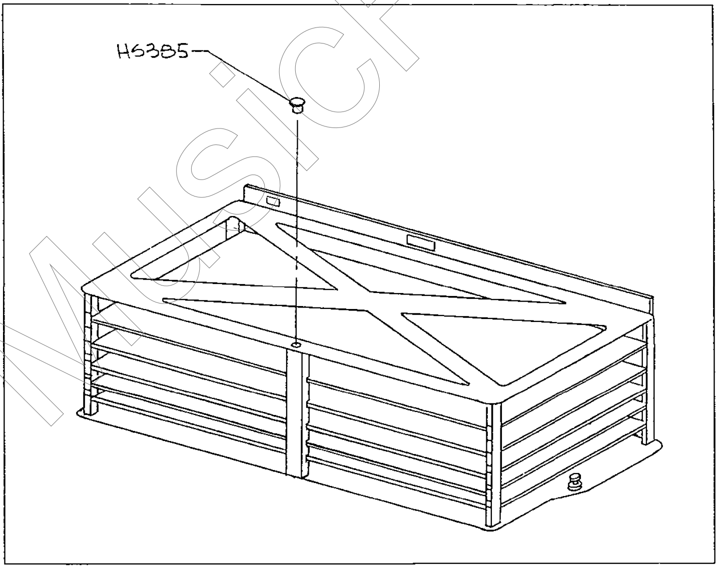
3. With the keyboard facing you, carefully lift the plastic housing up from the rear about an inch. Next, slide the housing toward you so that it clears the key of the keyboard. Gently lift up the housing and set it down behind the chassis so that you are looking through the keyboard cutout.

Opening the EIII (Rack Mount)

1. Remove the (6) phillips screws from the top panel of the rack and set aside in a safe place, then remove the top panel. The card cage can now be easily seen.

Removal of the Microcontroller Board

1. Before any boards can be removed the white nylon circuit board support must be removed. To do this, first remove the phillips screw (HS 385) fastening the support to the card cage and put it in your safe place. Next, press down slightly next to the other end of the card support. This action should allow the little pin on the lower end of the support to pop out of its hole. Set the support aside.



▲ Make sure not to stress the front panel ribbon cable when removing the EIII housing.

■ Use the same procedures for keyboard and rack versions.

EIII MEMORY UPGRADE
(continued)

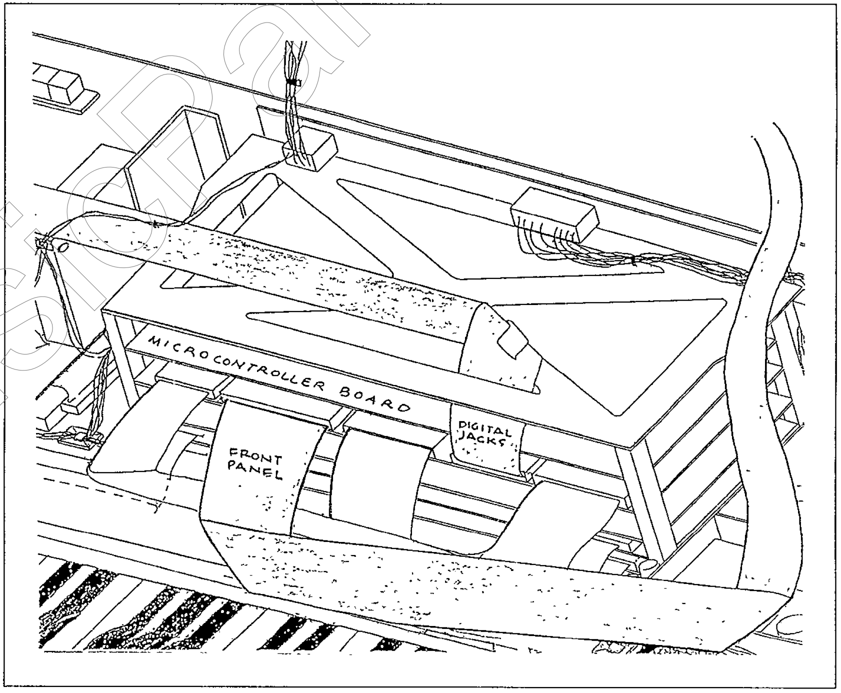
▲ On the rack version EIII ,the microcontroller board is blocked by all of the ribbon cables. Make a note of the cable routings before removing them.

2. The microcontroller board which contains the EIII's memory is the top board in the card cage. Its exit from the cage is blocked by the digital jack and front panel cables. These must first be removed.

3. Use the white nylon board ejectors to eject the second board from the top (the CPU board). Slide the board out only slightly.

4. Use your fingers to remove the ribbon cables that are blocking the exit of the top board. These ribbon cables are very fragile so work carefully.

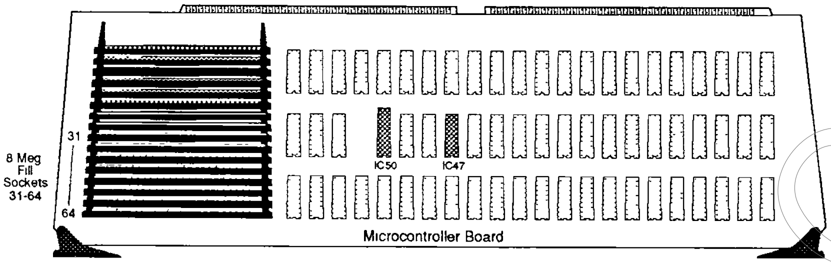
5. Now the microcontroller board can be removed. Use the board ejectors to eject the board and carefully slide it completely out of the card cage. Take the microcontroller board to a padded work area to perform the upgrade.



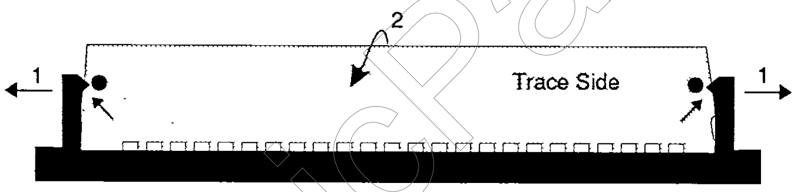
EIII MEMORY UPGRADE
(continued)

Update the Memory

1. Place the EIII microcontroller board in front of you as shown in the diagram.



2. Remove all of the 256 Kbyte SIMM boards. Start at the IC 64 location and work toward the other end. The SIMMs are easily removed by pushing the two end tabs outward (1) and tilting the SIMM toward you (2). Set the 256 Kbyte SIMMs aside so that they are not confused with the 1 Mbyte SIMMs.

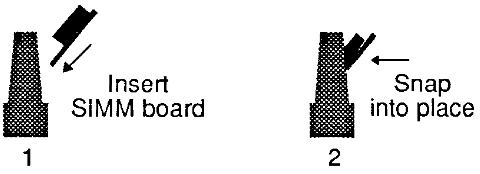


Front View

▲ Caution: the SIMM modules are static sensitive. Use static protection when handling these expensive parts.

3. Insert the 1 Mbyte SIMMs starting at location IC 31 and ending at location IC 64. With the microcontroller board in the position shown in the diagram, the chip side of the SIMM will face away from you.

Side View



EIII MEMORY UPGRADE
(continued)

▲ The most common mistake when installing memory upgrades is not firmly seating the SIMM modules.

4.Make sure that all the SIMM boards are firmly seated in their sockets. They will "snap" into place when inserted correctly.

5. Locate the RAS PAL (IP376). Replace the chip in location IC 50 with this PAL.

6. Locate the Addr. PAL (IP377). Replace the chip in location IC 47 with this PAL.

Replacement of the Circuit Boards

1. Re-insert the microcontroller board. Lift up the center of the board before pressing it in all the way. It should now slide easily into its sockets on the motherboard. Exercise caution but make sure that the board is securely mated.

■ The ribbon cables may not be routed as shown in the diagram. You should re-route the cables as shown.

2. Carefully plug in the digital jack and front panel cables. Route the cables as shown in the diagram.

3. Re-insert the CPU board. Again lift the center of the board and use caution when mating it with the motherboard.

4. Line up the notches in the nylon circuit board support with the edges of the circuit boards. The support should be positioned next to the pin mating hole. Now press down on the lower side of the card cage and slide the card support over so that the pin slips into its mating hole. Secure the support to the top side of the cage with the phillips screw.

Re-assembly

1. Lift up the plastic housing and slide the keyboard cutout under the front edge of the keyboard.

2. While continuing to hold the rear of the housing up, tuck the front panel ribbon cable down near the rear of the keyboard. This is to make sure that the cable does not get pinched between the housing and the card cage.

3. The plastic housing should seat snugly to the bottom chassis. Again, make sure that no cables are being pinched.

EIII MEMORY UPGRADE

(continued)

4. Replace the (4) back panel screws, then replace the (8) bottom screws. Be careful not to cross-thread the screws.

Power on

1. The moment of truth. Plug in audio cables and apply power. The EIII should power up normally.

NO!? → First check the ribbon cables to the card cage. Next, check the seating of the SIMMs. After all, it was OK before you worked on it.

YES → Great! Let's do a memory test to make sure that all is indeed well.

EIII RAM Test

1. Activate the Master module.

2. Select Submodule 8 (Special).

3. Move the data slider up to show the last choice.

4. Use the keypad and type in one number past the last choice. For example: if the last numbered choice was 7, you would type in 8.

5. The display will say: DIAGNOSTICS, Enter Magic Code.

6. The magic code for EIII diagnostics is 1-3-5-8 . A good way to remember this number is to recall that 1-3-5-8 = Major chord.

7. Now select 2, RAM Test from the diagnostics menu. The EIII should verify that it is now an 8 MB machine and begin testing RAM. If all is well, the display will indicate RAM Test Complete!! If an error is detected, the display will indicate which SIMM failed. A failure might indicate that you have not seated a SIMM correctly, so check them before assuming a bad SIMM.

8. Done.