*K*2600 v2.0

Musician's Guide Supplement

KURZWEIL

Music Systems

Part Number: 910337 Rev. A

CAUTION

RISK OF ELECTRIC SHOCK

DO NOT OPEN

CAUTION: TO REDUCE THE RISK OF ELECTRIC SHOCK,
DO NOT REMOVE THE COVER
NO USER SERVICEABLE PARTS INSIDE
REFER SERVICING TO QUALIFIED SERVICE PERSONNEL

The lightning flash with the arrowhead symbol, within an equilateral triangle, is intended to alert the user to the presence of uninsulated "dangerous voltage" within the product's enclosure that may be of sufficient magnitude to constitute a risk of electric shock to persons.

The exclamation point within an equilateral triangle is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.

IMPORTANT SAFETY & INSTALLATION INSTRUCTIONS

INSTRUCTIONS PERTAINING TO THE RISK OF FIRE, ELECTRIC SHOCK, OR INJURY TO PERSONS

WARNING: *When using electric products, basic precautions should always be followed, including the following:*

1. *Read all of the Safety and Installation Instructions and Explanation of Graphic Symbols before using the product.*
2. *This product must be grounded. If it should malfunction or break down, grounding provides a path of least resistance for electric current to reduce the risk of electric shock. This product is equipped with a power supply cord having an equipment-grounding conductor and a grounding plug. The plug must be plugged into an appropriate outlet which is properly installed and grounded in accordance with all local codes and ordinances.*

   DANGER: *Improper connection of the equipment-grounding conductor can result in a risk of electric shock. Do not modify the plug provided with the product - if it will not fit the outlet, have a proper outlet installed by a qualified electrician. Do not use an adaptor which defeats the function of the equipment-grounding conductor. If you are in doubt as to whether the product is properly grounded, check with a qualified serviceman or electrician.*
3. ***WARNING:*** *This product is equipped with an AC input voltage selector. The voltage selector has been factory set for the mains supply voltage in the country where this unit was sold. Changing the voltage selector may require the use of a different power supply cord or attachment plug, or both. To reduce the risk of fire or electric shock, refer servicing to qualified maintenance personnel.*
4. *Do not use this product near water - for example, near a bathtub, washbowl, kitchen sink, in a wet basement, or near a swimming pool, or the like.*
5. *This product should only be used with a stand or cart that is recommended by the manufacturer.*
6. *This product, either alone or in combination with an amplifier and speakers or headphones, may be capable of producing sound levels that could cause permanent hearing loss. Do not operate for a long period of time at a high volume level or at a level that is uncomfortable. If you experience any hearing loss or ringing in the ears, you should consult an audiologist.*
7. *The product should be located so that its location or position does not interfere with its proper ventilation.*
8. *The product should be located away from heat sources such as radiators, heat registers, or other products that produce heat.*
9. *The product should be connected to a power supply only of the type described in the operating instructions or as marked on the product.*
10. *This product may be equipped with a polarized line plug (one blade wider than the other). This is a safety feature. If you are unable to insert the plug into the outlet, contact an electrician to replace your obsolete outlet. Do not defeat the safety purpose of the plug.*
11. *The power supply cord of the product should be unplugged from the outlet when left unused for a long period of time. When unplugging the power supply cord, do not pull on the cord, but grasp it by the plug.*
12. *Care should be taken so that objects do not fall and liquids are not spilled into the enclosure through openings.*
13. *The product should be serviced by qualified service personnel when:*
    - *A. The power supply cord or the plug has been damaged;*
    - *B. Objects have fallen, or liquid has been spilled into the product;*
    - *C. The product has been exposed to rain;*
    - *D. The product does not appear to be operating normally or exhibits a marked change in performance;*
    - *E. The product has been dropped, or the enclosure damaged.*
14. *Do not attempt to service the product beyond that described in the user maintenance instructions. All other servicing should be referred to qualified service personnel.*
15. ***WARNING:*** *Do not place objects on the product's power supply cord, or place the product in a position where anyone could trip over, walk on, or roll anything over cords of any type. Do not allow the product to rest on or be installed over cords of any type. Improper installations of this type create the possibility of a fire hazard and/or personal injury.*

RADIO AND TELEVISION INTERFERENCE

WARNING: *Changes or modifications to this instrument not expressly approved by Young Chang could void your authority to operate the instrument.*

IMPORTANT: *When connecting this product to accessories and/or other equipment use only high quality shielded cables.*

NOTE: *This instrument has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This instrument generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this instrument does cause harmful interference to radio or television reception, which can be determined by turning the instrument off and on, the user is encouraged to try to correct the interference by one or more of the following measures:*

- *Reorient or relocate the receiving antenna.*
- *Increase the separation between the instrument and the receiver.*
- *Connect the instrument into an outlet on a circuit other than the one to which the receiver is connected.*
- *If necessary consult your dealer or an experienced radio/television technician for additional suggestions.*

NOTICE

This apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

AVIS

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class B prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

SAVE THESE INSTRUCTIONS

Young Chang Distributors

Contact the nearest Young Chang office listed below to locate your local Young Chang/ Kurzweil representative.

Young Chang America, Inc.
P.O. Box 99995
Lakewood, WA 98499-0995
Tel: (253) 589-3200
Fax: (253) 984-0245

Young Chang Co.
178-55 Gajwa-Dong
Seo-Ku, Inchon, Korea 404-714
Tel: 011-82-32-570-1380
Fax: 011-82-32-570-1218

Young Chang Akki Europe GmbH
Industriering 45
D-41751 Viersen
Germany
Tel: 011-49-2162-4491
Fax: 011-49-2162-41744

Young Chang America, Inc. (Canadian Division)
3650 Victoria Park Ave. Suite 105
Toronto, Ontario Canada M2H 3P7
Tel: (416) 492-9899
Fax: (416) 492-9299

Chapter 1 Introduction

Chapter 2 Triple Modular Processing

Chapter 3 Disk Partitioning

Chapter 1
Introduction

Version 2 operating system software for the K2600 works in any K2600 model. It provides two important new features:

- Triple-modular processing
- Hard-disk partitioning

Triple-modular processing takes VAST to the third power, you can create a super-layer (called a triple) that combines the sound-generation and DSP capabilities of three layers into one.

Hard-disk partitioning enables you to use up to 8 gigabytes of partitioned hard-disk space to organize your samples, programs, setups, and songs. You can use a partitioned hard disk exclusively with your K2600, or you can share it between the K2600 and a personal computer.

What's In This Supplement

This supplement to the K2600 Musician's Guide provides information about the features available with Version 2 software for the K2600. There are two chapters:

- Chapter 2, *Triple Modular Processing*, describes how to create and program triple layers for astounding new possibilities in digital synthesis
- Chapter 3, *Disk Partitioning*, explains how to format, partition, and use your hard disk for convenient storage of all K2600-related files

This is Revision A of the Version 2 supplement to the K2600 Musician's Guide. Check our website for more recent revisions:

www.kurzweilmusicsystems.com/html/support.html

Chapter 2
Triple Modular Processing

Overview

Triple-modular processing is an enhancement to Kurzweil's VAST synthesis model (Variable-Architecture Synthesis Technology). The VAST model incorporates multi-layer programs; each layer in a VAST program uses an algorithm that includes pitch control, the renowned Kurzweil VAST DSP (digital signal-processing) functions, and amplitude control. This provides for a huge number of combinations.

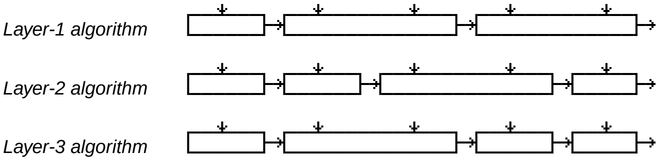


Figure 2-1 Algorithms of three possible normal layers, with three separate signal paths

Triple-modular processing makes the VAST model even more flexible, enabling you to create *triple layers* (or just *triples*)—specialized groups of three layers with one continuous signal path. The audio signal from the first layer of a triple gets routed to the second layer, where it's mixed with whatever audio signal the second layer may be generating. Similarly, the output of the second layer is routed to the third layer. The third layer's output goes to the input buses of KDFX, then to the physical audio outputs.

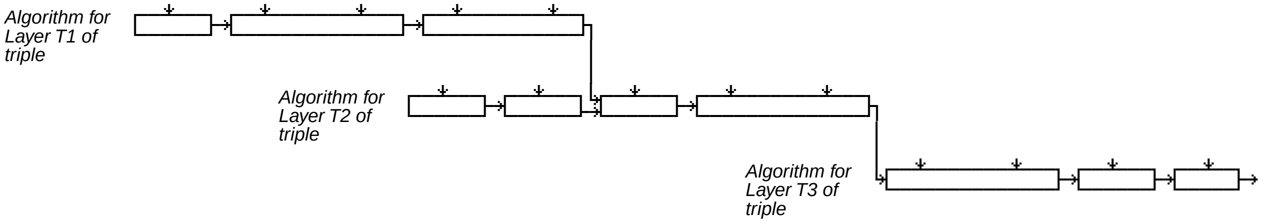

Algorithm for Layer T1 of triple
Algorithm for Layer T2 of triple
Algorithm for Layer T3 of triple

Figure 2-2 Algorithm of a possible triple, with a single signal path

This enables you to create sophisticated signal paths and chains of DSP functions. For example, you can mix the audio signal generated by one layer with the audio signal generated by the following layer, or you can apply three layers of DSP to the audio signal generated by Layer 1. There's an almost limitless number of possible combinations (something like 30 *billion*, depending on how you count).

In Figure 2-2, for example, the processed audio signal from Layer 1 (sample playback and/or one or more DSP-generated waveforms) gets mixed with the processed audio signal from Layer 2. The Layer-1 signal gets mixed with the Layer-2 signal just before the third DSP block. If there were a different algorithm in Layer 2, the Layer-1 signal might get mixed with the Layer-2 signal at a different location, or in multiple locations (both Layer 2 and Layer 3 can have multiple inputs from the previous layer).

Note that any processing in Layer 2 that's "downstream" (to the right) of the input(s) from Layer 1 gets applied to the signals from both Layer 1 and Layer 2. After the processing in Layer 2, the signal passes to Layer 3 for still more processing before going to KDFX.

Any non-KB3 program can contain a triple, alone or in combination with normal layers and/or other triples (KB3 programs don't use layers, so they can't contain triples). Each triple is a contiguous set of three otherwise normal layers, with all the usual layer parameters. Because their audio paths are linked, the layers of a triple are always numbered consecutively. You can't reorder the layers within a triple, but you can locate the triple anywhere within a program. For example, a ten-layer program might contain two triples: one triple spanning Layers 2–4 and the other spanning Layers 7–9 (with Layers 1, 5, 6, and 10 being normal layers).

Triples and Polyphony

Each triple uses three voices, the equivalent of three normal layers. By using a triple instead of three normal layers, you add millions of sound-editing possibilities without using any more voices. If you play enough notes to exceed the K2600's polyphonic limit, the three voices used by each triple get stolen as a unit, according to the priority set by the parameters on the AMPENV page of Layer 3 of the triple.

Soloing and Muting

You can solo and mute triples much as you can normal layers. Soloing and muting always affects the entire triple, however; you can't solo or mute individual layers of a triple.

KB3 Programs

You can't use triples in KB3 programs, since KB3 programs don't use the VAST model. That is, they don't consist of layers like non-KB3 programs (instead, every KB3 program plays notes that are generated by tone wheels that start running as soon as you select the program). Consequently, KB3 programs can't contain triples, because triples are specialized layers.

Even so, KB3 programs can coexist in setups with programs that use triples. Keep in mind though, that since a typical KB3 program uses about 40 voices, any setup that uses one of these programs has only eight voices left over. This means that you might not always hear every note of a setup that contains both a KB3 program and one or more programs containing a triple (or multiple triples). Whenever a KB3 note is playing, the K2600 can play a maximum of two notes from triples.

***Note**: You can edit a KB3 program to reduce the number of tone wheels it uses, which frees up additional voices (one voice for every two tone wheels). See Number of Tone Wheels on page 6-49 of your Musician's Guide.*

Live Mode

Live Mode enables you to take an external audio signal and route it through the K2600's VAST sound engine. (This works only if you have the Sampling Option installed in your K2600.) Triple-modular processing makes for an even greater number of possibilities in Live mode. For example, a triple can route a single Live-mode input through three layers of DSP, or it can mix a Live-mode input with an internally-generated K2600 sound. A triple can also mix two Live-mode inputs with each other and/or with an internally-generated K2600 sound.

KDS Input

If you have the digital I/O option for the K2600, you can use inputs KDS1–8 to patch external signals into layers, enabling you to use VAST processing on virtually any signal. You can't patch an external signal into a triple, however; the KDS inputs work only on normal VAST programs.

Algorithms for Triple Modular Processing

There are 94 new VAST algorithms, incorporating familiar Kurzweil DSP functions. The algorithms are divided into three sets, each of which correspond to a specific layer within the triple. In other words, certain algorithms are available only for certain layers.

Layer-1 Algorithms: 33–62

There are 30 algorithms available for Layer 1 of any triple. The output of each of these algorithms is permanently configured to go to one or more inputs in a Layer-2 algorithm (63–100). All of these algorithms begin with pitch blocks, like the normal VAST algorithms. This ensures that every triple has an audio signal.

Layer-2 Algorithms: 63–100

There are 38 algorithms available for Layer 2 of any triple. The output of each of these algorithms is permanently configured to go to one or more inputs in a Layer-3 algorithm (101–127).

Some of the Layer-2 algorithms (63–80) begin with pitch blocks, and some (81–100) do not. Layers that use algorithms 81–100 don't make use of the parameters on the PITCH page in the Program Editor.

Layer-3 Algorithms: 101–126

There are 26 algorithms available for Layer 3 of any triple. The output of each of these algorithms is permanently configured to go to the input of KDFX. None of these algorithms has a pitch block, so layers that use them don't make use of the parameters on the PITCH page in the Program Editor.

Algorithm Reference

Turn to page 2-12 for diagrams of each algorithm, including the DSP functions available for each block in each algorithm.

Compatibility with Other Kurzweil Instruments

K2000s and K2500s (as well as all earlier models) can't load programs that contain triples, since they don't have the software required to support triples. Similarly, you can't load programs containing triples into K2600s with operating system versions earlier than v2.0.

Creating Triples

There are several ways to add a triple to a program. The most convenient method depends on what you want to do.

In case you're wondering, there's no way to convert a normal layer to a triple. Whenever you want to create a program containing a triple, or add a triple to a program, use one of the following methods.

Creating a New Triple

Creating a new triple inserts a copy of Layers 1–3 of Program **739** into the current program. This enables you to use Program **739** as a template for triples.

1. While in Program mode, select the program in which you want to create a triple.
2. Press **Edit** to enter the Program Editor.
3. Press **<more**, then press **NewLyr**. This displays a prompt asking you whether to create a normal layer or a triple.
4. Press **Triple**. The K2600 creates the triple, adding three linked layers after the highest-numbered existing layer (for example, if the program contained three layers, the new triple would occupy Layers 4–6). Adding the triple takes a few seconds; the display shows you the K2600's progress).
5. Rename and/or save the program as desired.

If you want to use one of your own triples as the template, edit the triple in Program **739**, then save the program with ID **739** (this creates a RAM version of the program, temporarily overwriting the ROM version—you can restore the ROM version by deleting Program **739**, which erases the RAM version).

Copying a Program Containing a Triple

This method works well when you want to make a few relatively simple adjustments to an existing program (without changing the original): changing keymaps, using a different KDFX studio, tweaking an amplitude envelope, or adjusting one or more DSP blocks in the triple itself.

1. While in Program mode, select the program you want to copy.
2. Press **Edit** to enter the Program Editor.
3. Press **<more** until you see the **Name** soft button.
4. Press **Name**, rename the program, and press **OK**.
5. Press **Save**, and save the program to a new ID (if you're editing a RAM program, change the ID to avoid replacing an existing program). See page 5-3 of your *Musician's Guide* if you need help naming or saving.
6. Edit the new program as desired.

As an alternative, you can use the Copy function in Master mode, as described on page 11-16 of your *Musician's Guide*.

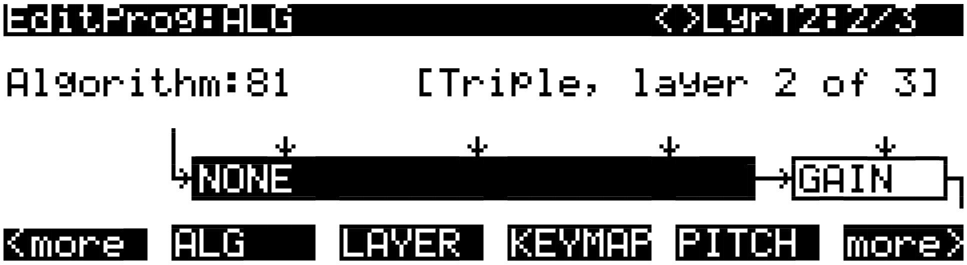
Importing a Triple into a Program

Use this method when there’s an existing triple that you want to insert into a particular program.

1. While in Program mode, select the program into which you want to import the triple.
2. Press **Edit** to enter the Program Editor.
3. Press **<more**, then press **ImpLyr** (Import Layer).
4. When the Import Layer dialog appears, select a program using the **Left** and **Right** cursor buttons.
5. Select a triple using the **Up** and **Down** cursor buttons. As you scroll through the layers, the display indicates which layers are triples. Notice that you can’t select individual layers within a triple.
6. Press **Import**. The K2600 imports the specified triple, adding three linked layers after the highest-numbered existing layer.
7. Rename and/or save the program as desired.

Editing Triples

The ALG page contains all the parameters for editing triples. The following diagram shows the ALG page for Layer 2 of Program **739 Default Triple**:



```
EditProg:ALG                    <>LyrT2:2/3
Algorithm:81      [Triple, layer 2 of 3]
```

Notice the arrow pointing to the left side of the first DSP block (not the arrows pointing to the top of the block; these are the DSP block’s control inputs). The arrow pointing to the left side of the DSP block represents the output from the first layer of the triple, which is the input for the second layer of the triple.

The third line of the display clearly identifies the layer as part of a triple. The top line also indicates that you’re looking at a triple: `T2` indicates that the current layer is the first layer of a triple, while `2/3` indicates that the current layer is Layer 2 of three layers in the program. Similarly, `LyrT3:6/8` would indicate that the current layer is Layer 3 of a triple, and is the sixth layer in a program containing a total of eight layers.

There are four basic operations in editing a triple:

- Selecting a layer with the **Chan/Bank** buttons
- Selecting an algorithm
- Changing DSP functions within the DSP blocks of an algorithm
- Editing the control page for each DSP block in each layer

The parameters on the ALG page consist of the algorithm itself, and the DSP blocks within the algorithm (In the diagram, the second DSP block in the algorithm is selected). Use the cursor buttons to select the parameters, and a data entry method to change the value of the selected parameter.

Amplitude Envelopes

A layer's amplitude envelope is a critical element of the layer's sound; it determines the layer's attack, decay, and release times—among other things. These features are controlled in one of two ways: by default settings stored in a layer's keymap, or by the settings for the parameters on the layer's AMPENV page.

The amplitude envelope gets applied to a layer's audio signal as it passes through the algorithm's AMP block (if it contains one). All the normal VAST algorithms (1–31) end with an AMP block, so normal VAST layers always use an amplitude envelope. This isn't true for all triple algorithms, however.

All the Layer-3 algorithms (101–126) end with AMP blocks, which ensures that there's at least one amplitude envelope for every triple. For Layers 1 and 2 of a triple, the AMP block is optional. To include an AMP block in an algorithm, assign a value of **AMP** in any one-stage DSP block from F1 to F4 (the PITCH block, if the layer contains one, has a fixed value).

It's important to keep in mind that whatever happens in the amplitude envelops in Layers 1 and 2 of a triple, the amplitude envelope in layer 3 affects the sound of the entire triple. For example, if you have a long release in the amplitude envelope in Layer 1, but a short release in the amplitude envelope in Layer 3, Layer 1's release will continue only as long as Layer 3's release

AMP Blocks in Layers 1 and 2 of a Triple

When you put an AMP block in Layer 1 or 2 of a triple, that layer uses its own amplitude envelope, as specified on the layer's AMPENV page. This envelope affects only its own layer, and not the subsequent layers in the triple. Also, amplitude envelopes (if any) applied to Layers 1 and 2 don't apply MIDI volume data; only the amplitude envelope in Layer 3 controls overall volume.

A layer that doesn't contain an AMP block ignores its own amplitude envelope. So, for example, if Layer 1 of a triple has no AMP block, Layer 1's output has no amplitude envelope. As the signal from Layer 1 mixes with the signal from Layer 2, the amplitude envelope from Layer 2, if any, may affect the sound from Layer 1 (this depends on Layer 2's inputs—see *Input Locations* on page 2-7.

When to Use AMP Blocks in Layers 1 and 2

Put an AMP block in Layer 1 or Layer 2 whenever you want that layer to use its own amplitude envelope. This affects the layer's sound to varying degrees, depending on two things: the keymap assigned to the layer, and the mode (Natural or User) of the layer's AMPENV.

AMP blocks can be particularly important in Layers 1 and 2 when those layers use keymaps that contain ROM samples or edited versions of ROM samples (especially samples of acoustic sounds like piano or strings). That's because the factory-default (natural) amplitude envelopes of many of these keymaps also control the decompression of the keymap's samples (to maximize audio quality, many samples are stored in compressed form, and are decompressed just before playback). If there's no AMP block, there's no amplitude envelope, and therefore no decompression information. This can alter the sound of a ROM sample drastically.

When Layer 1 or 2 of a triple uses a keymap containing ROM samples (or edited ROM samples), and you want the samples to sound relatively normal, you'll generally need to do two things:

- Include an AMP block in the layer
- Use the layer's default (natural) amplitude envelope

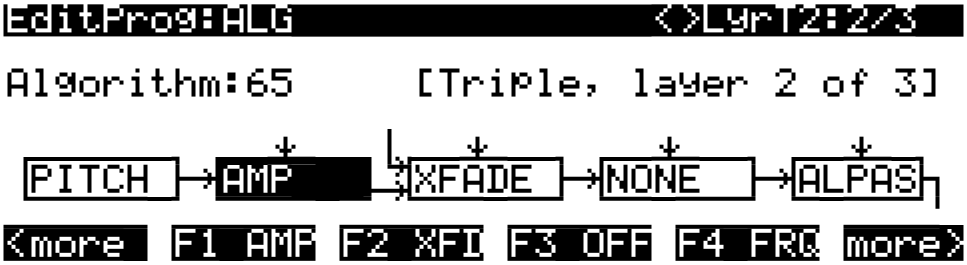
The AMP block ensures that the layer's amplitude envelope gets applied to the layer. The natural amplitude envelope ensures that the decompression information gets applied to the layer (user-defined envelopes contain only Attack-Decay-Release information, and no decompression information). Of course, you might like the way a layer sounds *without* the AMP block, or with a user-defined amplitude envelope.

To check whether a layer's keymap uses its default (natural) amplitude envelope or a user-defined amplitude envelope, go to the AMPENV page for that layer. If the only visible parameter is Mode, with a value of **Natural**, the layer is using its natural amplitude envelope. If you see a lot of parameters like Att1, Dec1, and Rel1, the layer is using a user-defined amplitude envelope. To revert to the natural amplitude envelope, select the word **User**, and change it to **Natural** by turning the Alpha Wheel one click in either direction.

Input Locations

We mentioned that Layer 2's amplitude envelope may or may not affect the sound of Layer 1—it depends on the location of Layer 2's AMP block relative to the input(s) from Layer 1.

Figure 2-3 shows the algorithm for Layer 2 of a triple. The input point from Layer 1 is in the third DSP block (F2, a crossfade mixer in this case), as indicated by the arrow entering the F2 block from above the algorithm. Layer 2's AMP block (F1, which is highlighted) is *before* the input point, which means that the amplitude envelope for Layer 2 gets applied *before* the signal from Layer 1 joins the signal from Layer 2. Consequently the amplitude envelope of Layer 2 affects only Layer 2, and doesn't affect the sound of Layer 1.



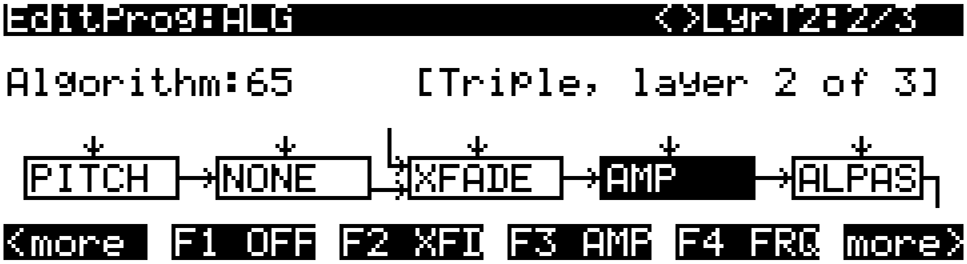
EditProg:ALG <>Lyr 2:2/3

Algorithm:65 [Triple, layer 2 of 3]

Figure 2-3 AMP block before input point

Figure 2-4 shows the same algorithm, with one change: the AMP block is now in F3 (highlighted), *after* the input point of Layer 2. This means that the amplitude envelope for

Layer 2 gets applied *after* the signal from Layer 1 joins the signal from Layer 2. Consequently, Layer 2's amplitude envelope affects Layer 1's sound as well as Layer 2's sound.



```
EditProg:ALG                <>LyrT2:2/3
Algorithm:65     [Triple, layer 2 of 3]
```

Figure 2-4 AMP block after input point

Amplitude Envelopes and the Keymap in Layer 3

Since Layer 3 of a triple always ends with an AMP block, triples always apply Layer 3's amplitude envelope, which may override the effects of any amplitude envelopes in Layers 1 or 2. The effect of Layer 3's envelope on the sound of the triple depends on whether Layer 3 uses a natural amplitude envelope or a user-defined one. When Layer 3 of a triple uses a natural amplitude envelope, the keymap used in Layer 3 determines the final amplitude envelope for the triple (because the natural amplitude envelope is stored as part of the keymap). In other words, changing the keymap for Layer 3 changes the amplitude envelope. This is an easy way to use alternative amplitude envelopes—for example, applying the amplitude envelope of a piano to a guitar sound generated in layer 1 or Layer 2. If you use a user envelope in Layer 3, it doesn't matter what the keymap is; changing the keymap doesn't change the amplitude envelope.

The Natural Amplitude Envelope

Look at the AMPENV page for Layer 3 of the triple. If you see the default (natural) amplitude envelope (that is, if you see only the Mode parameter with a value of **Natural**), the natural amplitude envelope of the keymap used in Layer 3 defines the final amplitude envelope for the triple. So if you assign a piano keymap to Layer 3, the piano keymap's amplitude envelope affects the sound of the entire triple—possibly overriding the amplitude envelopes of Layers 1 and 2. On the other hand, if you use a keymap that doesn't produce a sound (like **0 None** or **168 Silence**), the triple produces no sound at all.

***Note**: When you're using the Layer-3 keymap's default (natural) amplitude envelope, changing the value of the Xpose parameter on the KEYMAP page for Layer 3 changes the scaling of the amplitude envelope, and consequently, the sound of the triple. In general, raising the value of Xpose decreases the duration of each amplitude-envelope segment, while lowering the value of Xpose increases the duration of each segment.*

The User Envelope

You can use one of your own amplitude envelopes instead of the layer's natural amplitude envelope. On the AMPENV page for Layer 3, change the value of the Mode parameter from **Natural** to **User**. This removes the default amplitude envelope, and applies a user-defined envelope according to the values of the parameters on the AMPENV page (Att1, Dec1, Rel1, etc.). In this case, it doesn't matter which keymap you assign to Layer 3.

Summary of Amplitude Envelopes in Triples

Because of the special nature of triples and their interactions with amplitude envelopes, you'll need to give some thought to the amplitude.in each layer of the triples you program. There are three primary points to remember, each of which significantly affects the sound of a triple:

- The optional use of AMP blocks in Layers 1 and 2 (which determines whether Layers 1 and 2 use their own amplitude envelopes)
- The keymaps (if any) in each layer of a triple (which determine the sound produced by each layer)
- Choosing natural or user amplitude envelopes, especially in Layer 3

Other Considerations

Triples are a lot like normal layers, but there are a few important differences.

Processing-Only Layers

Layers that use Algorithms 81–127 can be processing-only layers—that is, they can provide additional processing for the audio output of other layers, without generating their own sounds. (They can provide additional processing *and* produce sound, as you'll see.)

Layer 1 of a triple usually generates sound via sample playback (unless it uses Keymap **0 None** or **168 Silence**). In other words, the sample playback of Layer 1 is routed to the input of Layer 1's algorithm. Consequently Layer 1 doesn't work as a processing-only layer. In addition to processing sample playback, most of the Layer-1 algorithms (33–62) can also generate waveforms within the algorithms themselves, using DSP functions like **SINE** and **SAW**.

For some of the Layer-2 algorithms (63–80) the input is the sample playback from Layer 2, mixed at one or more points with the output from Layer 1. For the remainder of the Layer-2 algorithms (81–100), the only input is the output from Layer 1; the sample playback from layer 2 (if any) is not routed to the inputs of these algorithms. Most of the Layer-2 algorithms can also generate waveforms within the algorithms themselves.

The input for all of the Layer-3 algorithms is the output from Layer 2. These algorithms never process the sample playback (if any) from Layer 3. Like the other algorithms, however, they can generate their own waveforms.

Since layers that use Algorithms 81–127 don't process their own sample playback data, they have no pitch parameters, and their algorithm diagrams contain no pitch blocks. When you're editing a layer that uses one of these algorithms, if you press the **PITCH** soft button, you'll see a blank page.

Layer Parameters

Since a triple is essentially a single layer (except that it uses three voices per note), it has a single set of layer parameters. You can view and edit these parameters by pressing the **LAYER** soft button when Layer 1 of the triple is current. If you press **LAYER** when Layer 2 or 3 is current, you see a blank page.

Output Parameters

Triples have a single set of output parameters controlling all three layers (again because triples are essentially single layers). You can view and edit the output parameters by pressing the **OUTPUT** soft button when Layer 3 of the triple is current. If you press **OUTPUT** when Layer 1 or 2 is current, you see a blank page.

New Combinations of DSP Functions

Seven new two-stage DSP functions combine a filter or double shaper with a gain function (the gain occurs *after* the filtering/shaping). One or more of these functions is available in most of the two-stage DSP blocks. They're listed below.

LOPAS2 GAIN

HIPAS2 GAIN

BAND2 GAIN

NOTCH2 GAIN

LP2RES GAIN

SHAPE2 GAIN

LPGATE GAIN

These functions are equivalent to two single-stage blocks in a v1.0 algorithm—a block using a filter or shaper followed by a block using the GAIN function.

Stereo Keymaps

Because of sound-processing requirements, triples can't use stereo keymaps. Consequently, there's no Stereo parameter on the Keymap page when any layer of a triple is the current layer. Programs that use both triples and normal layers can use stereo keymaps for the normal layers.

Note Stealing in Triples

When you exceed the 48-voice polyphonic limit, the K2600 uses the AMPENV-page settings for Layer 3 of a triple to determine how the voices from that triple get stolen. Consequently, if you want to change how the triple's notes get stolen, you should edit the parameters on the AMPENV page for Layer 3 of the triple (as opposed to editing the AMPENV parameters for Layers 1 or 2). For example, decreasing the duration of the decay and/or release segments reduces your polyphony requirements.

Using DSP Waveforms

Keep in mind that the DSP waveforms (like SINE, SQR, SAW, and NOISE are typically 5 to 6 dB hotter than ROM samples. This can cause clipping in some of the filters used in triple algorithms. Consequently, you may want to reduce the input levels for filters that are processing DSP waveforms.

Using PWM

DSP blocks that use the PWM function are meant to be followed by blocks using the DIST function (this is true for normal VAST layers, as well). See Chapter 16 of your *Musician's Guide* for more information.

Using NOISE+

The DSP function called NOISE+ is available for many one-stage DSP blocks in Layer-2 and Layer-3 algorithms. This provides a convenient way to add noise at one or more points in a triple. When you use this function in a DSP block, it adds nearly-white noise—that is, an audio signal with nearly equal amplitude at all audible frequencies—to the existing signal (unless the existing signal has zero amplitude, in which case no noise gets added).

In order to function as designed, DSP blocks that use NOISE+ must be followed immediately by one of the DSP functions listed below. Otherwise, you won't be able to attenuate the level of the added noise. If you use NOISE+ in the *last* DSP block of a Layer-2 algorithm, the *first* DSP block of Layer 3's algorithm must be one of these functions.

NONE
AMP
PANNER
BAL AMP
+AMP
+GAIN
XFADE

Exploring the Possibilities

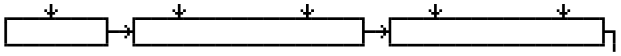
You may never exhaust the possible combinations of DSP functions available with triple-modular processing, but keep experimenting. We've found that using simple waveforms when trying different DSP functions makes it easier to hear the effects of each function. Try using SINE, SQR, and SAW in various DSP blocks. These block all incoming audio signals and generate a simple waveform as output (as opposed to SINE+, SQR+, and SAW+, which add waveforms to the existing signal).

Algorithm Reference

This section contains a diagram for each triple algorithm, as you see it in the K2600's display. Below each diagram is a list of the DSP functions available in each block of the algorithm.

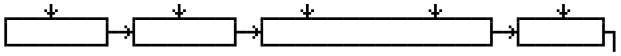
Layer-1 Algorithms (33–62)

Algorithm:33



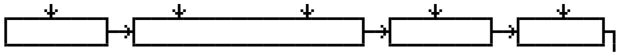
PITCH	*NONE*	*NONE*
	2PARAM SHAPER	*2PARAM SHAPER*
	LOPAS2 GAIN	*LOPAS2 GAIN*
	2POLE LOWPASS	*2POLE LOWPASS*
	BANDPASS FILT	*BANDPASS FILT*
	NOTCH FILTER	*NOTCH FILTER*
	2POLE ALLPASS	*2POLE ALLPASS*
	PARA BASS	*PARA BASS*
	PARA TREBLE	*PARA TREBLE*
	PARA MID	*PARA MID*

Algorithm:34



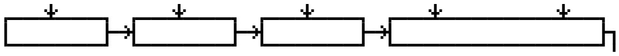
PITCH	*NONE*	*NONE*	*NONE*
	AMP	*2PARAM SHAPER*	*AMP*
	LOPASS	*LOPAS2 GAIN*	*LOPASS*
	HIPASS	*2POLE LOWPASS*	*HIPASS*
	ALPASS	*BANDPASS FILT*	*ALPASS*
	GAIN	*NOTCH FILTER*	*GAIN*
	SHAPER	*2POLE ALLPASS*	*SHAPER*
	DIST	*PARA BASS*	*DIST*
	PWM	*PARA TREBLE*	*PWM*
	SINE	*PARA MID*	*SINE*
	LF SIN		*LF SIN*
	SW+SHP		*SW+SHP*
	SAW+		*SAW+*
	SAW		*SAW*
	LF SAW		*LF SAW*
	SQUARE		*SQUARE*
	LF SQR		*LF SQR*
	WRAP		*WRAP*

Algorithm:35



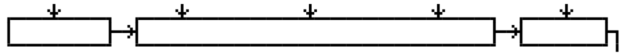
PITCH	*NONE*	*NONE*	*NONE*
	2PARAM SHAPER	*AMP*	*AMP*
	LOPAS2 GAIN	*LOPASS*	*LOPASS*
	2POLE LOWPASS	*HIPASS*	*HIPASS*
	BANDPASS FILT	*ALPASS*	*ALPASS*
	NOTCH FILTER	*GAIN*	*GAIN*
	2POLE ALLPASS	*SHAPER*	*SHAPER*
	PARA BASS	*DIST*	*DIST*
	PARA TREBLE	*PWM*	*PWM*
	PARA MID	*SINE*	*SINE*
		LF SIN	*LF SIN*
		SW+SHP	*SW+SHP*
		SAW+	*SAW+*
		SAW	*SAW*
		LF SAW	*LF SAW*
		SQUARE	*SQUARE*
		LF SQR	*LF SQR*
		WRAP	*WRAP*

Algorithm:36



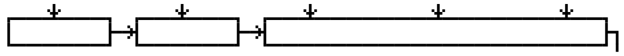
PITCH	*NONE*	*NONE*	*NONE*
	AMP	*AMP*	*2PARAM SHAPER*
	LOPASS	*LOPASS*	*LOPAS2 GAIN*
	HIPASS	*HIPASS*	*2POLE LOWPASS*
	ALPASS	*ALPASS*	*BANDPASS FILT*
	GAIN	*GAIN*	*NOTCH FILTER*
	SHAPER	*SHAPER*	*2POLE ALLPASS*
	DIST	*DIST*	*PARA BASS*
	PWM	*PWM*	*PARA TREBLE*
	SINE	*SINE*	*PARA MID*
	LF SIN	*LF SIN*	
	SW+SHP	*SW+SHP*	
	SAW+	*SAW+*	
	SAW	*SAW*	
	LF SAW	*LF SAW*	
	SQUARE	*SQUARE*	
	LF SQR	*LF SQR*	
	WRAP	*WRAP*	

Algorithm:37



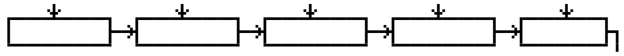
PITCH	NONE	NONE
	PARAMETRIC EQ	AMP
	STEEP RESONANT BASS	LOPASS
		HIPASS
		ALPASS
		GAIN
		SHAPER
		DIST
		PWM
		SINE
		LF SIN
		SW+SHP
		SAW+
		SAW
		LF SAW
		SQUARE
		LF SQR
		WRAP

Algorithm:38



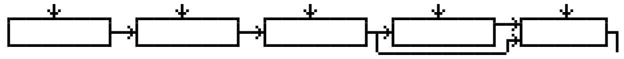
PITCH	NONE	NONE
	AMP	PARAMETRIC EQ
	LOPASS	STEEP RESONANT BASS
	HIPASS	
	ALPASS	
	GAIN	
	SHAPER	
	DIST	
	PWM	
	SINE	
	LF SIN	
	SW+SHP	
	SAW+	
	SAW	
	LF SAW	
	SQUARE	
	LF SQR	
	WRAP	

Algorithm:39



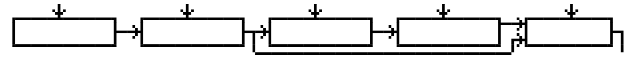
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	AMP
	LOPASS	LOPASS	LOPASS	LOPASS
	HIPASS	HIPASS	HIPASS	HIPASS
	ALPASS	ALPASS	ALPASS	ALPASS
	GAIN	GAIN	GAIN	GAIN
	SHAPER	SHAPER	SHAPER	SHAPER
	DIST	DIST	DIST	DIST
	PWM	PWM	PWM	PWM
	SINE	SINE	SINE	SINE
	LF SIN	LF SIN	LF SIN	LF SIN
	SW+SHP	SW+SHP	SW+SHP	SW+SHP
	SAW+	SAW+	SAW+	SAW+
	SAW	SAW	SAW	SAW
	LF SAW	LF SAW	LF SAW	LF SAW
	SQUARE	SQUARE	SQUARE	SQUARE
	LF SQR	LF SQR	LF SQR	LF SQR
	WRAP	WRAP	WRAP	WRAP

Algorithm:40



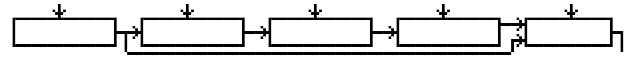
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	xGAIN
	LOPASS	LOPASS	LOPASS	+GAIN
	HIPASS	HIPASS	HIPASS	XFADE
	ALPASS	ALPASS	ALPASS	AMPMOD
	GAIN	GAIN	GAIN	
	SHAPER	SHAPER	SHAPER	
	DIST	DIST	DIST	
	PWM	PWM	PWM	
	SINE	SINE	SINE	
	LF SIN	LF SIN	LF SIN	
	SW+SHP	SW+SHP	SW+SHP	
	SAW+	SAW+	SAW+	
	SAW	SAW	SAW	
	LF SAW	LF SAW	LF SAW	
	SQUARE	SQUARE	SQUARE	
	LF SQR	LF SQR	LF SQR	
	WRAP	WRAP	WRAP	

Algorithm:41



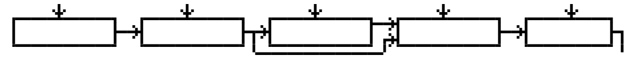
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	xGAIN
	LOPASS	LOPASS	LOPASS	+GAIN
	HIPASS	HIPASS	HIPASS	XFADE
	ALPASS	ALPASS	ALPASS	AMPMOD
	GAIN	GAIN	GAIN	
	SHAPER	SHAPER	SHAPER	
	DIST	DIST	DIST	
	PWM	PWM	PWM	
	SINE	SINE	SINE	
	LF SIN	LF SIN	LF SIN	
	SW+SHP	SW+SHP	SW+SHP	
	SAW+	SAW+	SAW+	
	SAW	SAW	SAW	
	LF SAW	LF SAW	LF SAW	
	SQUARE	SQUARE	SQUARE	
	LF SQR	LF SQR	LF SQR	
	WRAP	WRAP	WRAP	

Algorithm:42



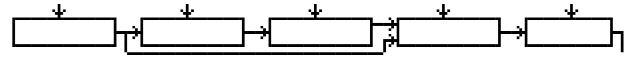
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	xGAIN
	LOPASS	LOPASS	LOPASS	+GAIN
	HIPASS	HIPASS	HIPASS	XFADE
	ALPASS	ALPASS	ALPASS	AMPMOD
	GAIN	GAIN	GAIN	
	SHAPER	SHAPER	SHAPER	
	DIST	DIST	DIST	
	PWM	PWM	PWM	
	SINE	SINE	SINE	
	LF SIN	LF SIN	LF SIN	
	SW+SHP	SW+SHP	SW+SHP	
	SAW+	SAW+	SAW+	
	SAW	SAW	SAW	
	LF SAW	LF SAW	LF SAW	
	SQUARE	SQUARE	SQUARE	
	LF SQR	LF SQR	LF SQR	
	WRAP	WRAP	WRAP	

Algorithm:43



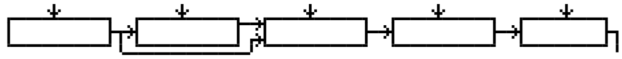
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	xGAIN	AMP
	LOPASS	LOPASS	+GAIN	LOPASS
	HIPASS	HIPASS	XFADE	HIPASS
	ALPASS	ALPASS	AMPMOD	ALPASS
	GAIN	GAIN		GAIN
	SHAPER	SHAPER		SHAPER
	DIST	DIST		DIST
	PWM	PWM		PWM
	SINE	SINE		SINE
	LF SIN	LF SIN		LF SIN
	SW+SHP	SW+SHP		SW+SHP
	SAW+	SAW+		SAW+
	SAW	SAW		SAW
	LF SAW	LF SAW		LF SAW
	SQUARE	SQUARE		SQUARE
	LF SQR	LF SQR		LF SQR
	WRAP	WRAP		WRAP

Algorithm:44



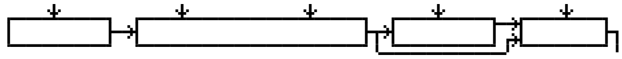
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	xGAIN	AMP
	LOPASS	LOPASS	+GAIN	LOPASS
	HIPASS	HIPASS	XFADE	HIPASS
	ALPASS	ALPASS	AMPMOD	ALPASS
	GAIN	GAIN		GAIN
	SHAPER	SHAPER		SHAPER
	DIST	DIST		DIST
	PWM	PWM		PWM
	SINE	SINE		SINE
	LF SIN	LF SIN		LF SIN
	SW+SHP	SW+SHP		SW+SHP
	SAW+	SAW+		SAW+
	SAW	SAW		SAW
	LF SAW	LF SAW		LF SAW
	SQUARE	SQUARE		SQUARE
	LF SQR	LF SQR		LF SQR
	WRAP	WRAP		WRAP

Algorithm:45



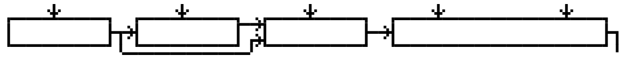
PITCH	NONE	NONE	NONE	NONE
	AMP	xGAIN	AMP	AMP
	LOPASS	+GAIN	LOPASS	LOPASS
	HIPASS	XFADE	HIPASS	HIPASS
	ALPASS	AMPMOD	ALPASS	ALPASS
	GAIN		GAIN	GAIN
	SHAPER		SHAPER	SHAPER
	DIST		DIST	DIST
	PWM		PWM	PWM
	SINE		SINE	SINE
	LF SIN		LF SIN	LF SIN
	SW+SHP		SW+SHP	SW+SHP
	SAW+		SAW+	SAW+
	SAW		SAW	SAW
	LF SAW		LF SAW	LF SAW
	SQUARE		SQUARE	SQUARE
	LF SQR		LF SQR	LF SQR
	WRAP		WRAP	WRAP

Algorithm:47



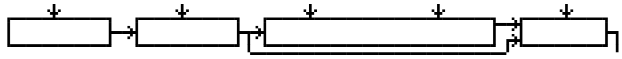
PITCH	NONE	NONE	NONE
	2PARAM SHAPER	AMP	xGAIN
	LOPAS2 GAIN	LOPASS	+GAIN
	2POLE LOWPASS	HIPASS	XFADE
	BANDPASS FILT	ALPASS	AMPMOD
	NOTCH FILTER	GAIN	
	2POLE ALLPASS	SHAPER	
	PARA BASS	DIST	
	PARA TREBLE	PWM	
	PARA MID	SINE	
		LF SIN	
		SW+SHP	
		SAW+	
		SAW	
		LF SAW	
		SQUARE	
		LF SQR	
		WRAP	

Algorithm:46



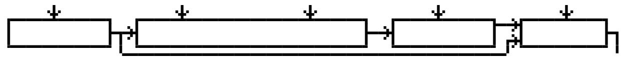
PITCH	NONE	NONE	NONE
	AMP	xGAIN	2PARAM SHAPER
	LOPASS	+GAIN	LOPAS2 GAIN
	HIPASS	XFADE	2POLE LOWPASS
	ALPASS	AMPMOD	BANDPASS FILT
	GAIN		NOTCH FILTER
	SHAPER		2POLE ALLPASS
	DIST		PARA BASS
	PWM		PARA TREBLE
	SINE		PARA MID
	LF SIN		
	SW+SHP		
	SAW+		
	SAW		
	LF SAW		
	SQUARE		
	LF SQR		
	WRAP		

Algorithm:48



PITCH	NONE	NONE	NONE
	AMP	2PARAM SHAPER	xGAIN
	LOPASS	LOPAS2 GAIN	+GAIN
	HIPASS	2POLE LOWPASS	XFADE
	ALPASS	BANDPASS FILT	AMPMOD
	GAIN	NOTCH FILTER	
	SHAPER	2POLE ALLPASS	
	DIST	PARA BASS	
	PWM	PARA TREBLE	
	SINE	PARA MID	
	LF SIN		
	SW+SHP		
	SAW+		
	SAW		
	LF SAW		
	SQUARE		
	LF SQR		
	WRAP		

Algorithm:49



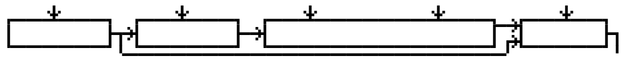
- PITCH

- NONE
- 2PARAM SHAPER
- LOPAS2   GAIN
- 2POLE LOWPASS
- BANDPASS FILT
- NOTCH FILTER
- 2POLE ALLPASS
- PARA BASS
- PARA TREBLE
- PARA MID

- NONE
- AMP
- LOPASS
- HIPASS
- ALPASS
- GAIN
- SHAPER
- DIST
- PWM
- SINE
- LF SIN
- SW+SHP
- SAW+
- SAW
- LF SAW
- SQUARE
- LF SQR
- WRAP

- NONE
- xGAIN
- +GAIN
- XFADE
- AMPMOD

Algorithm:50



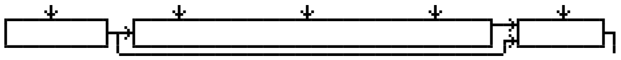
- PITCH

- NONE
- AMP
- LOPASS
- HIPASS
- ALPASS
- GAIN
- SHAPER
- DIST
- PWM
- SINE
- LF SIN
- SW+SHP
- SAW+
- SAW
- LF SAW
- SQUARE
- LF SQR
- WRAP

- NONE
- 2PARAM SHAPER
- LOPAS2   GAIN
- 2POLE LOWPASS
- BANDPASS FILT
- NOTCH FILTER
- 2POLE ALLPASS
- PARA BASS
- PARA TREBLE
- PARA MID

- NONE
- xGAIN
- +GAIN
- XFADE
- AMPMOD

Algorithm:51

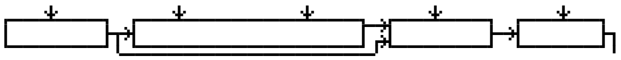


- PITCH

- NONE
- PARAMETRIC EQ
- STEEP RESONANT BASS

- NONE
- xGAIN
- +GAIN
- XFADE
- AMPMOD

Algorithm:52



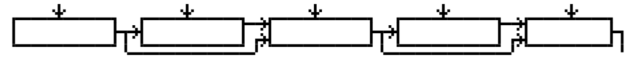
- PITCH

- NONE
- 2PARAM SHAPER
- LOPAS2   GAIN
- 2POLE LOWPASS
- BANDPASS FILT
- NOTCH FILTER
- 2POLE ALLPASS
- PARA BASS
- PARA TREBLE
- PARA MID

- NONE
- xGAIN
- +GAIN
- XFADE
- AMPMOD

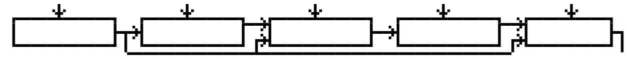
- NONE
- AMP
- LOPASS
- HIPASS
- ALPASS
- GAIN
- SHAPER
- DIST
- PWM
- SINE
- LF SIN
- SW+SHP
- SAW+
- SAW
- LF SAW
- SQUARE
- LF SQR
- WRAP

Algorithm:53



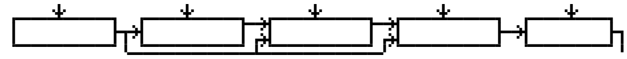
PITCH	NONE	NONE	NONE	NONE
	AMP	xGAIN	AMP	xGAIN
	LOPASS	+GAIN	LOPASS	+GAIN
	HIPASS	XFADE	HIPASS	XFADE
	ALPASS	AMPMOD	ALPASS	AMPMOD
	GAIN		GAIN	
	SHAPER		SHAPER	
	DIST		DIST	
	PWM		PWM	
	SINE		SINE	
	LF SIN		LF SIN	
	SW+SHP		SW+SHP	
	SAW+		SAW+	
	SAW		SAW	
	LF SAW		LF SAW	
	SQUARE		SQUARE	
	LF SQR		LF SQR	
	WRAP		WRAP	

Algorithm:54



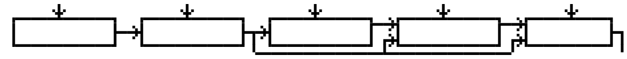
PITCH	NONE	NONE	NONE	NONE
	AMP	xGAIN	AMP	xGAIN
	LOPASS	+GAIN	LOPASS	+GAIN
	HIPASS	XFADE	HIPASS	XFADE
	ALPASS	AMPMOD	ALPASS	AMPMOD
	GAIN		GAIN	
	SHAPER		SHAPER	
	DIST		DIST	
	PWM		PWM	
	SINE		SINE	
	LF SIN		LF SIN	
	SW+SHP		SW+SHP	
	SAW+		SAW+	
	SAW		SAW	
	LF SAW		LF SAW	
	SQUARE		SQUARE	
	LF SQR		LF SQR	
	WRAP		WRAP	

Algorithm:55



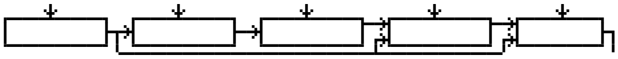
PITCH	NONE	NONE	NONE	NONE
	AMP	xGAIN	xGAIN	AMP
	LOPASS	+GAIN	+GAIN	LOPASS
	HIPASS	XFADE	XFADE	HIPASS
	ALPASS	AMPMOD	AMPMOD	ALPASS
	GAIN			GAIN
	SHAPER			SHAPER
	DIST			DIST
	PWM			PWM
	SINE			SINE
	LF SIN			LF SIN
	SW+SHP			SW+SHP
	SAW+			SAW+
	SAW			SAW
	LF SAW			LF SAW
	SQUARE			SQUARE
	LF SQR			LF SQR
	WRAP			WRAP

Algorithm:56



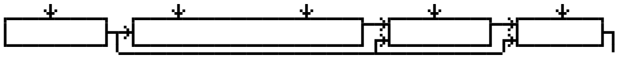
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	xGAIN	xGAIN
	LOPASS	LOPASS	+GAIN	+GAIN
	HIPASS	HIPASS	XFADE	XFADE
	ALPASS	ALPASS	AMPMOD	AMPMOD
	GAIN	GAIN		
	SHAPER	SHAPER		
	DIST	DIST		
	PWM	PWM		
	SINE	SINE		
	LF SIN	LF SIN		
	SW+SHP	SW+SHP		
	SAW+	SAW+		
	SAW	SAW		
	LF SAW	LF SAW		
	SQUARE	SQUARE		
	LF SQR	LF SQR		
	WRAP	WRAP		

Algorithm:57



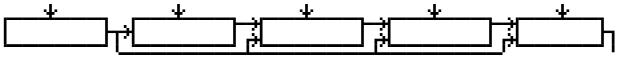
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	xGAIN	xGAIN
	LOPASS	LOPASS	+GAIN	+GAIN
	HIPASS	HIPASS	XFADE	XFADE
	ALPASS	ALPASS	AMPMOD	AMPMOD
	GAIN	GAIN		
	SHAPER	SHAPER		
	DIST	DIST		
	PWM	PWM		
	SINE	SINE		
	LF SIN	LF SIN		
	SW+SHP	SW+SHP		
	SAW+	SAW+		
	SAW	SAW		
	LF SAW	LF SAW		
	SQUARE	SQUARE		
	LF SQR	LF SQR		
	WRAP	WRAP		

Algorithm:58



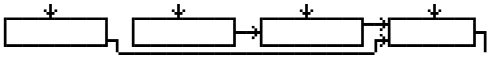
PITCH	NONE	NONE	NONE
	2PARAM SHAPER	xGAIN	xGAIN
	LOPAS2 GAIN	+GAIN	+GAIN
	2POLE LOWPASS	XFADE	XFADE
	BANDPASS FILT	AMPMOD	AMPMOD
	NOTCH FILTER		
	2POLE ALLPASS		
	PARA BASS		
	PARA TREBLE		
	PARA MID		

Algorithm:59



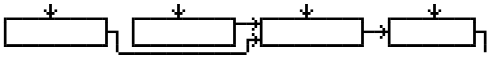
PITCH	NONE	NONE	NONE	NONE
	AMP	xGAIN	xGAIN	xGAIN
	LOPASS	+GAIN	+GAIN	+GAIN
	HIPASS	XFADE	XFADE	XFADE
	ALPASS	AMPMOD	AMPMOD	AMPMOD
	GAIN			
	SHAPER			
	DIST			
	PWM			
	SINE			
	LF SIN			
	SW+SHP			
	SAW+			
	SAW			
	LF SAW			
	SQUARE			
	LF SQR			
	WRAP			

Algorithm:60



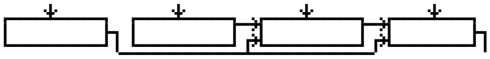
SYNCM	SYNCS	NONE	NONE
		AMP	xGAIN
		LOPASS	+GAIN
		HIPASS	XFADE
		ALPASS	AMPMOD
		GAIN	
		SHAPER	
		DIST	
		PWM	
		SINE	
		LF SIN	
		SW+SHP	
		SAW+	
		SAW	
		LF SAW	
		SQUARE	
		LF SQR	
		WRAP	

Algorithm:61



SYNCM	SYNCS	NONE	NONE
		xGAIN	AMP
		+GAIN	LOPASS
		XFADE	HIPASS
		AMPMOD	ALPASS
			GAIN
			SHAPER
			DIST
			PWM
			SINE
			LF SIN
			SW+SHP
			SAW+
			SAW
			LF SAW
			SQUARE
			LF SQR
			WRAP

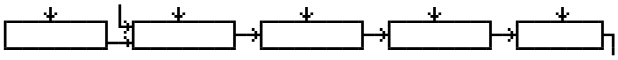
Algorithm:62



SYNCM	SYNCS	NONE	NONE
		xGAIN	xGAIN
		+GAIN	+GAIN
		XFADE	XFADE
		AMPMOD	AMPMOD

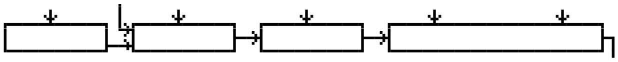
Layer-2 Algorithms (63–100)

Algorithm:63



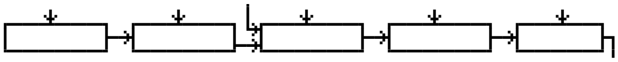
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	AMP
	+GAIN	LOPASS	LOPASS	LOPASS
	XFADE	HIPASS	HIPASS	HIPASS
	AMPMOD	ALPASS	ALPASS	ALPASS
		GAIN	GAIN	GAIN
		SHAPER	SHAPER	SHAPER
		DIST	DIST	DIST
		PWM	SINE	SINE
		SINE	LFSIN	LFSIN
		LF SIN	SW+SHP	SW+SHP
		SW+SHP	SAW+	SAW+
		SAW+	SW+DIST	SW+DIST
		SAW	LPCLIP	LPCLIP
		LF SAW	SINE+	SINE+
		SQUARE	NOISE+	NOISE+
		LF SQR		
		WRAP		

Algorithm:64



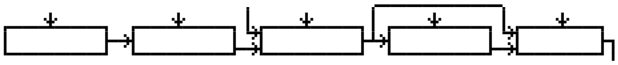
PITCH	NONE	NONE	NONE
	xGAIN	AMP	LOPAS2 GAIN
	+GAIN	LOPASS	HIPAS2 GAIN
	XFADE	HIPASS	BAND2 GAIN
	AMPMOD	ALPASS	NOTCH2 GAIN
		GAIN	LP2RES GAIN
		SHAPER	SHAPE2 GAIN
		DIST	LPGATE GAIN
		PWM	PARA MID
		SINE	
		LF SIN	
		SW+SHP	
		SAW+	
		SAW	
		LF SAW	
		SQUARE	
		LF SQR	
		WRAP	

Algorithm:65



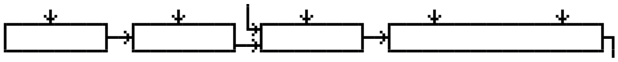
PITCH	*NONE*	*NONE*	*NONE*	*NONE*
	AMP	*xGAIN*	*AMP*	*AMP*
	LOPASS	*+GAIN*	*LOPASS*	*LOPASS*
	HIPASS	*XFADE*	*HIPASS*	*HIPASS*
	ALPASS	*AMPMOD*	*ALPASS*	*ALPASS*
	GAIN		*GAIN*	*GAIN*
	SHAPER		*SHAPER*	*SHAPER*
	DIST		*DIST*	*DIST*
	PWM		*SINE*	*SINE*
	SINE		*LFSIN*	*LFSIN*
	LF SIN		*SW+SHP*	*SW+SHP*
	SW+SHP		*SAW+*	*SAW+*
	SAW+		*SW+DIST*	*SW+DIST*
	SAW		*LPCLIP*	*LPCLIP*
	LF SAW		*SINE+*	*SINE+*
	SQUARE		*NOISE+*	*NOISE+*
	LF SQR			
	WRAP			

Algorithm:66



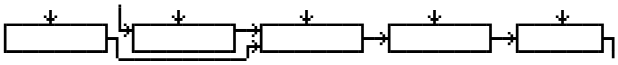
PITCH	*NONE*	*NONE*	*NONE*	*NONE*
	AMP	*xGAIN*	*AMP*	*xGAIN*
	LOPASS	*+GAIN*	*LOPASS*	*+GAIN*
	HIPASS	*XFADE*	*HIPASS*	*!GAIN*
	ALPASS	*AMPMOD*	*ALPASS*	
	GAIN		*GAIN*	
	SHAPER		*SHAPER*	
	DIST		*DIST*	
	PWM		*SINE*	
	SINE		*LFSIN*	
	LF SIN		*SW+SHP*	
	SW+SHP		*SAW+*	
	SAW+		*SW+DIST*	
	SAW		*LPCLIP*	
	LF SAW		*SINE+*	
	SQUARE		*NOISE+*	
	LF SQR			
	WRAP			

Algorithm:67



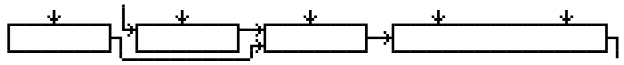
PITCH	*NONE*	*NONE*	*NONE*
	AMP	*xGAIN*	*LOPAS2 GAIN*
	LOPASS	*+GAIN*	*HIPAS2 GAIN*
	HIPASS	*XFADE*	*BAND2 GAIN*
	ALPASS	*AMPMOD*	*NOTCH2 GAIN*
	GAIN		*LP2RES GAIN*
	SHAPER		*SHAPE2 GAIN*
	DIST		*LPGATE GAIN*
	PWM		*PARA MID*
	SINE		
	LF SIN		
	SW+SHP		
	SAW+		
	SAW		
	LF SAW		
	SQUARE		
	LF SQR		
	WRAP		

Algorithm:68



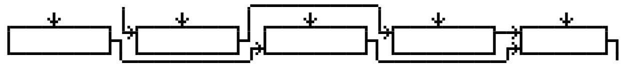
PITCH	*NONE*	*NONE*	*NONE*	*NONE*
	AMP	*xGAIN*	*AMP*	*AMP*
	LOPASS	*+GAIN*	*LOPASS*	*LOPASS*
	HIPASS	*XFADE*	*HIPASS*	*HIPASS*
	ALPASS	*AMPMOD*	*ALPASS*	*ALPASS*
	GAIN		*GAIN*	*GAIN*
	SHAPER		*SHAPER*	*SHAPER*
	DIST		*DIST*	*DIST*
	PWM		*SINE*	*SINE*
	SINE		*LFSIN*	*LFSIN*
	LF SIN		*SW+SHP*	*SW+SHP*
	SW+SHP		*SAW+*	*SAW+*
	SAW+		*SW+DIST*	*SW+DIST*
	SAW		*LPCLIP*	*LPCLIP*
	LF SAW		*SINE+*	*SINE+*
	SQUARE		*NOISE+*	*NOISE+*
	LF SQR			
	WRAP			

Algorithm:69



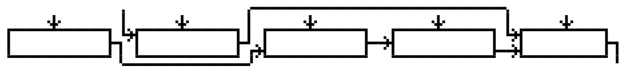
PITCH	NONE	NONE	NONE
	AMP	xGAIN	LOPAS2 GAIN
	LOPASS	+GAIN	HIPAS2 GAIN
	HIPASS	XFADE	BAND2 GAIN
	ALPASS	AMPMOD	NOTCH2 GAIN
	GAIN		LP2RES GAIN
	SHAPER		SHAPE2 GAIN
	DIST		LPGATE GAIN
	PWM		PARA MID
	SINE		
	LF SIN		
	SW+SHP		
	SAW+		
	SAW		
	LF SAW		
	SQUARE		
	LF SQR		
	WRAP		

Algorithm:70



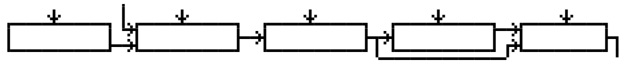
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	xGAIN
	LOPASS	LOPASS	LOPASS	+GAIN
	HIPASS	HIPASS	HIPASS	!GAIN
	ALPASS	ALPASS	ALPASS	
	GAIN	GAIN	GAIN	
	SHAPER	SHAPER	SHAPER	
	DIST	DIST	DIST	
	PWM	PWM	SINE	
	SINE	SINE	LFSIN	
	LF SIN	LF SIN	SW+SHP	
	SW+SHP	SW+SHP	SAW+	
	SAW+	SAW+	SW+DIST	
	SAW	SAW	LPCLIP	
	LF SAW	LF SAW	SINE+	
	SQUARE	SQUARE	NOISE+	
	LF SQR	LF SQR		
	WRAP	WRAP		

Algorithm:71



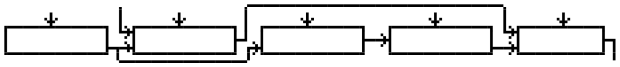
PITCH	NONE	NONE	NONE	NONE
	AMP	AMP	AMP	xGAIN
	LOPASS	LOPASS	LOPASS	+GAIN
	HIPASS	HIPASS	HIPASS	!GAIN
	ALPASS	ALPASS	ALPASS	
	GAIN	GAIN	GAIN	
	SHAPER	SHAPER	SHAPER	
	DIST	DIST	DIST	
	PWM	PWM	SINE	
	SINE	SINE	LFSIN	
	LF SIN	LF SIN	SW+SHP	
	SW+SHP	SW+SHP	SAW+	
	SAW+	SAW+	SW+DIST	
	SAW	SAW	LPCLIP	
	LF SAW	LF SAW	SINE+	
	SQUARE	SQUARE	NOISE+	
	LF SQR	LF SQR		
	WRAP	WRAP		

Algorithm:72



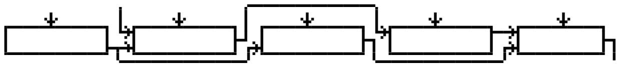
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:73



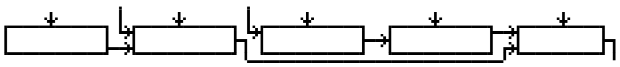
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:75



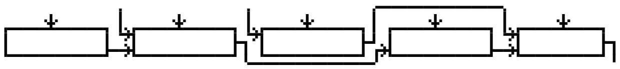
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:74



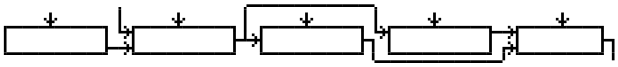
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:76



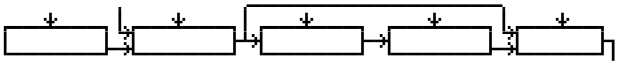
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:77



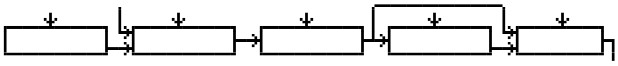
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:78



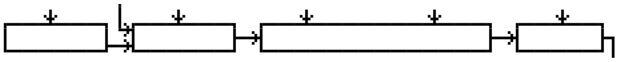
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:79



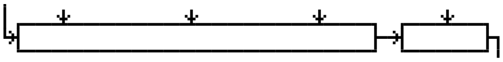
PITCH	NONE	NONE	NONE	NONE
	xGAIN	AMP	AMP	xGAIN
	+GAIN	LOPASS	LOPASS	+GAIN
	XFADE	HIPASS	HIPASS	!GAIN
	AMPMOD	ALPASS	ALPASS	
		GAIN	GAIN	
		SHAPER	SHAPER	
		DIST	DIST	
		PWM	SINE	
		SINE	LFSIN	
		LF SIN	SW+SHP	
		SW+SHP	SAW+	
		SAW+	SW+DIST	
		SAW	LPCLIP	
		LF SAW	SINE+	
		SQUARE	NOISE+	
		LF SQR		
		WRAP		

Algorithm:80



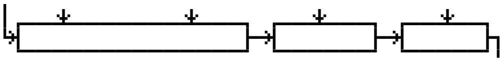
PITCH	NONE	NONE	GAIN
	xGAIN	PARA BASS	
	+GAIN	PARA TREBLE	
	XFADE		
	AMPMOD		

Algorithm:81



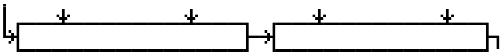
NONE	*GAIN*
HIFREQ STIMULATOR	
PARAMETRIC EQ	
STEEP RESONANT BASS	
4POLE LOPASS W/SEP	
4POLE HIPASS W/SEP	
TWIN PEAKS BANDPASS	
DOUBLE NOTCH W/SEP	

Algorithm:82



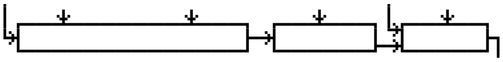
NONE	*NONE*	*NONE*
2PARAM SHAPER	*AMP*	*AMP*
LOPAS2 GAIN	*LOPASS*	*LOPASS*
2POLE LOWPASS	*HIPASS*	*HIPASS*
BANDPASS FILT	*ALPASS*	*ALPASS*
NOTCH FILTER	*GAIN*	*GAIN*
2POLE ALLPASS	*SHAPER*	*SHAPER*
PARA BASS	*DIST*	*DIST*
PARA TREBLE	*SINE*	*SINE*
PARA MID	*LFSIN*	*LFSIN*
	SW+SHP	*SW+SHP*
	SAW+	*SAW+*
	SW+DIST	*SW+DIST*
	LPCLIP	*LPCLIP*
	SINE+	*SINE+*
	NOISE+	*NOISE+*

Algorithm:83



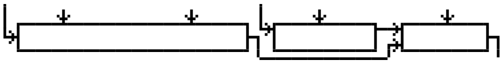
NONE	*NONE*
2PARAM SHAPER	*LOPAS2 GAIN*
LOPAS2 GAIN	*HIPAS2 GAIN*
2POLE LOWPASS	*BAND2 GAIN*
BANDPASS FILT	*NOTCH2 GAIN*
NOTCH FILTER	*LP2RES GAIN*
2POLE ALLPASS	*SHAPE2 GAIN*
PARA BASS	*LPGATE GAIN*
PARA TREBLE	*PARA MID*
PARA MID	

Algorithm:84



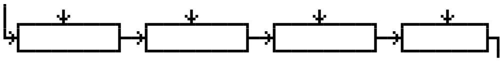
NONE	*NONE*	*NONE*
2PARAM SHAPER	*AMP*	*xGAIN*
LOPAS2 GAIN	*LOPASS*	*+GAIN*
2POLE LOWPASS	*HIPASS*	*!GAIN*
BANDPASS FILT	*ALPASS*	
NOTCH FILTER	*GAIN*	
2POLE ALLPASS	*SHAPER*	
PARA BASS	*DIST*	
PARA TREBLE	*SINE*	
PARA MID	*LFSIN*	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

Algorithm:85



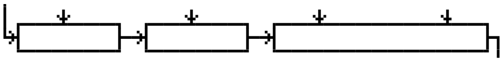
NONE	NONE	NONE
2PARAM SHAPER	AMP	xGAIN
LOPAS2 GAIN	LOPASS	+GAIN
2POLE LOWPASS	HIPASS	!GAIN
BANDPASS FILT	ALPASS	
NOTCH FILTER	GAIN	
2POLE ALLPASS	SHAPER	
PARA BASS	DIST	
PARA TREBLE	SINE	
PARA MID	LFSIN	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

Algorithm:86



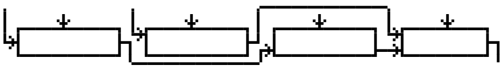
NONE	NONE	NONE	NONE
AMP	AMP	AMP	AMP
LOPASS	LOPASS	LOPASS	LOPASS
HIPASS	HIPASS	HIPASS	HIPASS
ALPASS	ALPASS	ALPASS	ALPASS
GAIN	GAIN	GAIN	GAIN
SHAPER	SHAPER	SHAPER	SHAPER
DIST	DIST	DIST	DIST
PWM	PWM	SINE	SINE
SINE	SINE	LFSIN	LFSIN
LF SIN	LF SIN	SW+SHP	SW+SHP
SW+SHP	SW+SHP	SAW+	SAW+
SAW+	SAW+	SW+DIST	SW+DIST
SAW	SAW	LPCLIP	LPCLIP
LF SAW	LF SAW	SINE+	SINE+
SQUARE	SQUARE	NOISE+	NOISE+
LF SQR	LF SQR		
WRAP	WRAP		

Algorithm:87



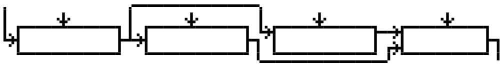
NONE	NONE	NONE
AMP	AMP	LOPAS2 GAIN
LOPASS	LOPASS	HIPAS2 GAIN
HIPASS	HIPASS	BAND2 GAIN
ALPASS	ALPASS	NOTCH2 GAIN
GAIN	GAIN	LP2RES GAIN
SHAPER	SHAPER	SHAPE2 GAIN
DIST	DIST	LPGATE GAIN
PWM	PWM	PARA MID
SINE	SINE	
LF SIN	LF SIN	
SW+SHP	SW+SHP	
SAW+	SAW+	
SAW	SAW	
LF SAW	LF SAW	
SQUARE	SQUARE	
LF SQR	LF SQR	
WRAP	WRAP	

Algorithm:88



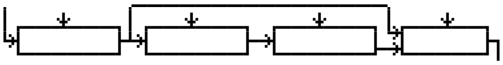
NONE	NONE	NONE	NONE
AMP	AMP	AMP	xGAIN
LOPASS	LOPASS	LOPASS	+GAIN
HIPASS	HIPASS	HIPASS	!GAIN
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
PWM	PWM	SINE	
SINE	SINE	LFSIN	
LF SIN	LF SIN	SW+SHP	
SW+SHP	SW+SHP	SAW+	
SAW+	SAW+	SW+DIST	
SAW	SAW	LPCLIP	
LF SAW	LF SAW	SINE+	
SQUARE	SQUARE	NOISE+	
LF SQR	LF SQR		
WRAP	WRAP		

Algorithm:89



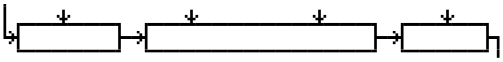
NONE	NONE	NONE	NONE
AMP	AMP	AMP	xGAIN
LOPASS	LOPASS	LOPASS	+GAIN
HIPASS	HIPASS	HIPASS	!GAIN
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
PWM	PWM	SINE	
SINE	SINE	LFSIN	
LF SIN	LF SIN	SW+SHP	
SW+SHP	SW+SHP	SAW+	
SAW+	SAW+	SW+DIST	
SAW	SAW	LPCLIP	
LF SAW	LF SAW	SINE+	
SQUARE	SQUARE	NOISE+	
LF SQR	LF SQR		
WRAP	WRAP		

Algorithm:90



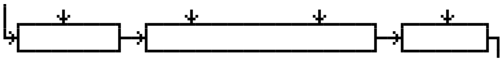
NONE	NONE	NONE	NONE
AMP	AMP	AMP	xGAIN
LOPASS	LOPASS	LOPASS	+GAIN
HIPASS	HIPASS	HIPASS	!GAIN
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
PWM	PWM	SINE	
SINE	SINE	LFSIN	
LF SIN	LF SIN	SW+SHP	
SW+SHP	SW+SHP	SAW+	
SAW+	SAW+	SW+DIST	
SAW	SAW	LPCLIP	
LF SAW	LF SAW	SINE+	
SQUARE	SQUARE	NOISE+	
LF SQR	LF SQR		
WRAP	WRAP		

Algorithm:91



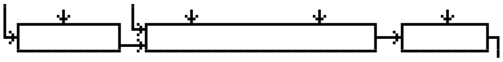
NONE	NONE	GAIN
AMP	PARA BASS	
LOPASS	PARA TREBLE	
HIPASS		
ALPASS		
GAIN		
SHAPER		
DIST		
PWM		
SINE		
LF SIN		
SW+SHP		
SAW+		
SAW		
LF SAW		
SQUARE		
LF SQR		
WRAP		

Algorithm:92



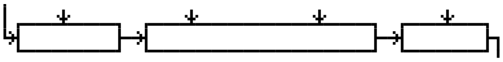
NONE	NONE	GAIN
AMP	SHAPE MOD OSC	
LOPASS	AMP MOD OSC	
HIPASS		
ALPASS		
GAIN		
SHAPER		
DIST		
PWM		
SINE		
LF SIN		
SW+SHP		
SAW+		
SAW		
LF SAW		
SQUARE		
LF SQR		
WRAP		

Algorithm:93



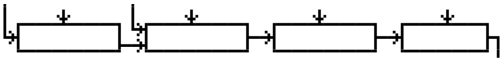
NONE	NONE	GAIN
AMP	x SHAPEMOD OSC	
LOPASS	+ SHAPEMOD OSC	
HIPASS		
ALPASS		
GAIN		
SHAPER		
DIST		
PWM		
SINE		
LF SIN		
SW+SHP		
SAW+		
SAW		
LF SAW		
SQUARE		
LF SQR		
WRAP		

Algorithm:94



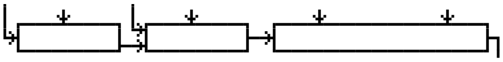
LOPAS2	SHAPEMOD OSC	GAIN

Algorithm:95



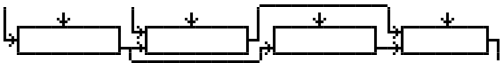
NONE	NONE	NONE	NONE
AMP	xGAIN	AMP	AMP
LOPASS	+GAIN	LOPASS	LOPASS
HIPASS	XFADE	HIPASS	HIPASS
ALPASS	AMPMOD	ALPASS	ALPASS
GAIN		GAIN	GAIN
SHAPER		SHAPER	SHAPER
DIST		DIST	DIST
PWM		SINE	SINE
SINE		LFSIN	LFSIN
LF SIN		SW+SHP	SW+SHP
SW+SHP		SAW+	SAW+
SAW+		SW+DIST	SW+DIST
SAW		LPCLIP	LPCLIP
LF SAW		SINE+	SINE+
SQUARE		NOISE+	NOISE+
LF SQR			
WRAP			

Algorithm:96



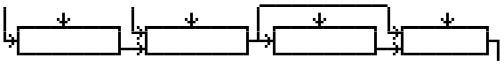
NONE	NONE	NONE
AMP	xGAIN	LOPAS2 GAIN
LOPASS	+GAIN	HIPAS2 GAIN
HIPASS	XFADE	BAND2 GAIN
ALPASS	AMPMOD	NOTCH2 GAIN
GAIN		LP2RES GAIN
SHAPER		SHAPE2 GAIN
DIST		LPGATE GAIN
PWM		PARA MID
SINE		
LF SIN		
SW+SHP		
SAW+		
SAW		
LF SAW		
SQUARE		
LF SQR		
WRAP		

Algorithm:97



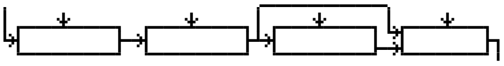
NONE	NONE	NONE	NONE
AMP	xGAIN	AMP	xGAIN
LOPASS	+GAIN	LOPASS	+GAIN
HIPASS	XFADE	HIPASS	!GAIN
ALPASS	AMPMOD	ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
PWM		SINE	
SINE		LFSIN	
LF SIN		SW+SHP	
SW+SHP		SAW+	
SAW+		SW+DIST	
SAW		LPCLIP	
LF SAW		SINE+	
SQUARE		NOISE+	
LF SQR			
WRAP			

Algorithm:98



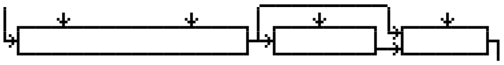
NONE	NONE	NONE	NONE
AMP	xGAIN	AMP	xGAIN
LOPASS	+GAIN	LOPASS	+GAIN
HIPASS	XFADE	HIPASS	!GAIN
ALPASS	AMPMOD	ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
PWM		SINE	
SINE		LFSIN	
LF SIN		SW+SHP	
SW+SHP		SAW+	
SAW+		SW+DIST	
SAW		LPCLIP	
LF SAW		SINE+	
SQUARE		NOISE+	
LF SQR			
WRAP			

Algorithm:99



NONE	NONE	NONE	NONE
AMP	AMP	AMP	xGAIN
LOPASS	LOPASS	LOPASS	+GAIN
HIPASS	HIPASS	HIPASS	XFADE
ALPASS	ALPASS	ALPASS	AMPMOD
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
PWM	PWM	PWM	
SINE	SINE	SINE	
LF SIN	LF SIN	LF SIN	
SW+SHP	SW+SHP	SW+SHP	
SAW+	SAW+	SAW+	
SAW	SAW	SAW	
LF SAW	LF SAW	LF SAW	
SQUARE	SQUARE	SQUARE	
LF SQR	LF SQR	LF SQR	
WRAP	WRAP	WRAP	

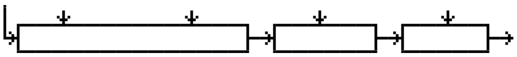
Algorithm:100



NONE	NONE	NONE
2PARAM SHAPER	AMP	xGAIN
LOPAS2 GAIN	LOPASS	+GAIN
2POLE LOWPASS	HIPASS	!GAIN
BANDPASS FILT	ALPASS	
NOTCH FILTER	GAIN	
2POLE ALLPASS	SHAPER	
PARA BASS	DIST	
PARA TREBLE	SINE	
PARA MID	LFSIN	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

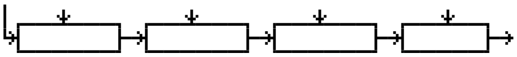
Layer-3 Algorithms (101–126)

Algorithm:101



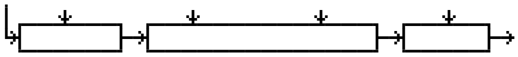
NONE	NONE	AMP
LOPAS2 GAIN	AMP	
HIPAS2 GAIN	LOPASS	
BAND2 GAIN	HIPASS	
NOTCH2 GAIN	ALPASS	
LP2RES GAIN	GAIN	
SHAPE2 GAIN	SHAPER	
LPGATE GAIN	DIST	
PARA MID	SINE	
	LFSIN	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

Algorithm:102



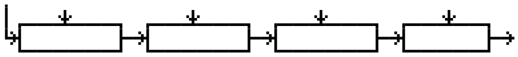
NONE	NONE	NONE	AMP
AMP	AMP	AMP	
LOPASS	LOPASS	LOPASS	
HIPASS	HIPASS	HIPASS	
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
SINE	SINE	SINE	
LFSIN	LFSIN	LFSIN	
SW+SHP	SW+SHP	SW+SHP	
SAW+	SAW+	SAW+	
SW+DIST	SW+DIST	SW+DIST	
LPCLIP	LPCLIP	LPCLIP	
SINE+	SINE+	SINE+	
NOISE+	NOISE+	NOISE+	

Algorithm:103



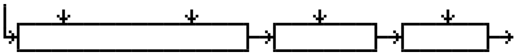
NONE	NONE	AMP
AMP	LOPAS2 GAIN	
LOPASS	HIPAS2 GAIN	
HIPASS	BAND2 GAIN	
ALPASS	NOTCH2 GAIN	
GAIN	LP2RES GAIN	
SHAPER	SHAPE2 GAIN	
DIST	LPGATE GAIN	
SINE	PARA MID	
LFSIN		
SW+SHP		
SAW+		
SW+DIST		
LPCLIP		
SINE+		
NOISE+		

Algorithm:104



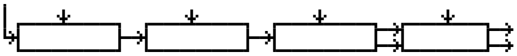
NONE	NONE	NONE	AMP
AMP	AMP	LOPAS2	
LOPASS	LOPASS	HIPAS2	
HIPASS	HIPASS	LPGATE	
ALPASS	ALPASS	LP2RES	
GAIN	GAIN	SHAPE2	
SHAPER	SHAPER		
DIST	DIST		
SINE	SINE		
LFSIN	LFSIN		
SW+SHP	SW+SHP		
SAW+	SAW+		
SW+DIST	SW+DIST		
LPCLIP	LPCLIP		
SINE+	SINE+		
NOISE+	NOISE+		

Algorithm:105



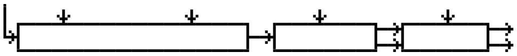
NONE	NONE	AMP
LOPAS2 GAIN	LOPAS2	
HIPAS2 GAIN	HIPAS2	
BAND2 GAIN	LPGATE	
NOTCH2 GAIN	LP2RES	
LP2RES GAIN	SHAPE2	
SHAPE2 GAIN		
LPGATE GAIN		
PARA MID		

Algorithm:106



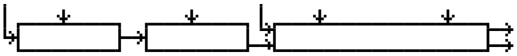
NONE	NONE	PANNER	AMP
AMP	AMP		
LOPASS	LOPASS		
HIPASS	HIPASS		
ALPASS	ALPASS		
GAIN	GAIN		
SHAPER	SHAPER		
DIST	DIST		
SINE	SINE		
LFSIN	LFSIN		
SW+SHP	SW+SHP		
SAW+	SAW+		
SW+DIST	SW+DIST		
LPCLIP	LPCLIP		
SINE+	SINE+		
NOISE+	NOISE+		

Algorithm:107



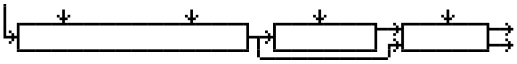
NONE	PANNER	AMP
LOPAS2 GAIN		
HIPAS2 GAIN		
BAND2 GAIN		
NOTCH2 GAIN		
LP2RES GAIN		
SHAPE2 GAIN		
LPGATE GAIN		
PARA MID		

Algorithm:108



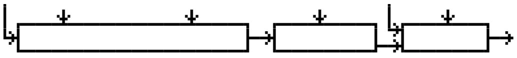
NONE	NONE	BAL AMP
AMP	AMP	AMPU AMPL
LOPASS	LOPASS	
HIPASS	HIPASS	
ALPASS	ALPASS	
GAIN	GAIN	
SHAPER	SHAPER	
DIST	DIST	
SINE	SINE	
LFSIN	LFSIN	
SW+SHP	SW+SHP	
SAW+	SAW+	
SW+DIST	SW+DIST	
LPCLIP	LPCLIP	
SINE+	SINE+	
NOISE+	NOISE+	

Algorithm:109



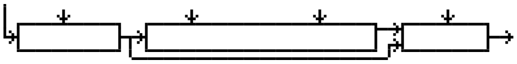
NONE	NONE	xAMP
LOPAS2 GAIN	AMP	+AMP
HIPAS2 GAIN	LOPASS	!AMP
BAND2 GAIN	HIPASS	
NOTCH2 GAIN	ALPASS	
LP2RES GAIN	GAIN	
SHAPE2 GAIN	SHAPER	
LPGATE GAIN	DIST	
PARA MID	SINE	
	LFSIN	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

Algorithm:110



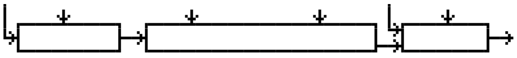
NONE	NONE	xAMP
LOPAS2 GAIN	AMP	+AMP
HIPAS2 GAIN	LOPASS	!AMP
BAND2 GAIN	HIPASS	
NOTCH2 GAIN	ALPASS	
LP2RES GAIN	GAIN	
SHAPE2 GAIN	SHAPER	
LPGATE GAIN	DIST	
PARA MID	SINE	
	LFSIN	
	SW+SHP	
	SAW+	
	SW+DIST	
	LPCLIP	
	SINE+	
	NOISE+	

Algorithm:111



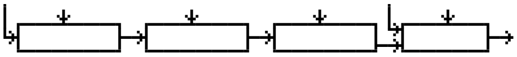
NONE	NONE	xAMP
AMP	LOPAS2 GAIN	+AMP
LOPASS	HIPAS2 GAIN	!AMP
HIPASS	BAND2 GAIN	
ALPASS	NOTCH2 GAIN	
GAIN	LP2RES GAIN	
SHAPER	SHAPE2 GAIN	
DIST	LPGATE GAIN	
SINE	PARA MID	
LFSIN		
SW+SHP		
SAW+		
SW+DIST		
LPCLIP		
SINE+		
NOISE+		

Algorithm:112



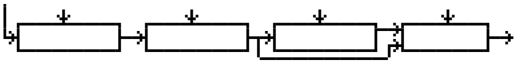
NONE	NONE	xAMP
AMP	LOPAS2 GAIN	+AMP
LOPASS	LP2RES GAIN	!AMP
HIPASS	SHAPE2 GAIN	
ALPASS	LPGATE GAIN	
GAIN	NOISE+ GAIN	
SHAPER		
DIST		
SINE		
LFSIN		
SW+SHP		
SAW+		
SW+DIST		
LPCLIP		
SINE+		
NOISE+		

Algorithm:113



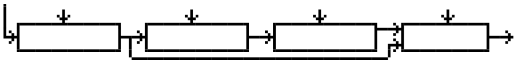
NONE	NONE	NONE	xAMP
AMP	AMP	AMP	+AMP
LOPASS	LOPASS	LOPASS	!AMP
HIPASS	HIPASS	HIPASS	
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
SINE	SINE	SINE	
LFSIN	LFSIN	LFSIN	
SW+SHP	SW+SHP	SW+SHP	
SAW+	SAW+	SAW+	
SW+DIST	SW+DIST	SW+DIST	
LPCLIP	LPCLIP	LPCLIP	
SINE+	SINE+	SINE+	
NOISE+	NOISE+	NOISE+	

Algorithm:114



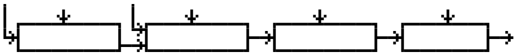
NONE	NONE	NONE	xAMP
AMP	AMP	AMP	+AMP
LOPASS	LOPASS	LOPASS	!AMP
HIPASS	HIPASS	HIPASS	
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
SINE	SINE	SINE	
LFSIN	LFSIN	LFSIN	
SW+SHP	SW+SHP	SW+SHP	
SAW+	SAW+	SAW+	
SW+DIST	SW+DIST	SW+DIST	
LPCLIP	LPCLIP	LPCLIP	
SINE+	SINE+	SINE+	
NOISE+	NOISE+	NOISE+	

Algorithm:115



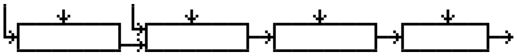
NONE	NONE	NONE	xAMP
AMP	AMP	AMP	+AMP
LOPASS	LOPASS	LOPASS	!AMP
HIPASS	HIPASS	HIPASS	
ALPASS	ALPASS	ALPASS	
GAIN	GAIN	GAIN	
SHAPER	SHAPER	SHAPER	
DIST	DIST	DIST	
SINE	SINE	SINE	
LFSIN	LFSIN	LFSIN	
SW+SHP	SW+SHP	SW+SHP	
SAW+	SAW+	SAW+	
SW+DIST	SW+DIST	SW+DIST	
LPCLIP	LPCLIP	LPCLIP	
SINE+	SINE+	SINE+	
NOISE+	NOISE+	NOISE+	

Algorithm:116



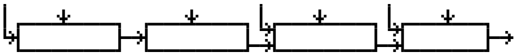
NONE	NONE	NONE	AMP
AMP	xGAIN	AMP	
LOPASS	+GAIN	LOPASS	
HIPASS	!GAIN	HIPASS	
ALPASS		ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
SINE		SINE	
LFSIN		LFSIN	
SW+SHP		SW+SHP	
SAW+		SAW+	
SW+DIST		SW+DIST	
LPCLIP		LPCLIP	
SINE+		SINE+	
NOISE+		NOISE+	

Algorithm:117



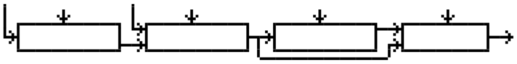
NONE	NONE	NONE	AMP
AMP	xGAIN	LOPAS2	
LOPASS	+GAIN	HIPAS2	
HIPASS	!GAIN	LPGATE	
ALPASS		LP2RES	
GAIN		SHAPE2	
SHAPER			
DIST			
SINE			
LFSIN			
SW+SHP			
SAW+			
SW+DIST			
LPCLIP			
SINE+			
NOISE+			

Algorithm:118



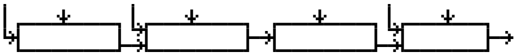
NONE	NONE	NONE	xAMP
AMP	AMP	xGAIN	+AMP
LOPASS	LOPASS	+GAIN	!AMP
HIPASS	HIPASS	!GAIN	
ALPASS	ALPASS		
GAIN	GAIN		
SHAPER	SHAPER		
DIST	DIST		
SINE	SINE		
LFSIN	LFSIN		
SW+SHP	SW+SHP		
SAW+	SAW+		
SW+DIST	SW+DIST		
LPCLIP	LPCLIP		
SINE+	SINE+		
NOISE+	NOISE+		

Algorithm:119



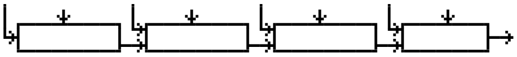
NONE	NONE	NONE	xAMP
AMP	xGAIN	AMP	+AMP
LOPASS	+GAIN	LOPASS	!AMP
HIPASS	!GAIN	HIPASS	
ALPASS		ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
PWM		SINE	
SINE		LFSIN	
LF SIN		SW+SHP	
SW+SHP		SAW+	
SAW+		SW+DIST	
SAW		LPCLIP	
LF SAW		SINE+	
SQUARE		NOISE+	
LF SQR			
WRAP			

Algorithm:120



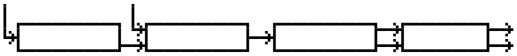
NONE	NONE	NONE	xAMP
AMP	xGAIN	AMP	+AMP
LOPASS	+GAIN	LOPASS	!AMP
HIPASS	!GAIN	HIPASS	
ALPASS		ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
SINE		SINE	
LFSIN		LFSIN	
SW+SHP		SW+SHP	
SAW+		SAW+	
SW+DIST		SW+DIST	
LPCLIP		LPCLIP	
SINE+		SINE+	
NOISE+		NOISE+	

Algorithm:121



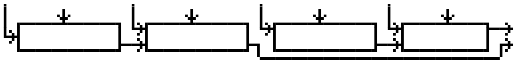
NONE	NONE	NONE	xAMP
AMP	xGAIN	xGAIN	+AMP
LOPASS	+GAIN	+GAIN	!AMP
HIPASS	!GAIN	!GAIN	
ALPASS			
GAIN			
SHAPER			
DIST			
SINE			
LFSIN			
SW+SHP			
SAW+			
SW+DIST			
LPCLIP			
SINE+			
NOISE+			

Algorithm:122



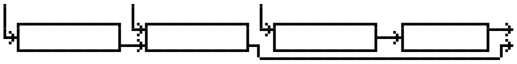
NONE	NONE	PANNER	AMP
AMP	xGAIN		
LOPASS	+GAIN		
HIPASS	!GAIN		
ALPASS			
GAIN			
SHAPER			
DIST			
SINE			
LFSIN			
SW+SHP			
SAW+			
SW+DIST			
LPCLIP			
SINE+			
NOISE+			

Algorithm:123



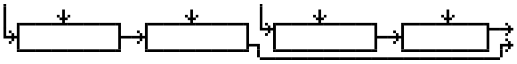
NONE	xAMP	NONE	xAMP
AMP	+AMP	AMP	+AMP
LOPASS	!AMP	LOPASS	!AMP
HIPASS		HIPASS	
ALPASS		ALPASS	
GAIN		GAIN	
SHAPER		SHAPER	
DIST		DIST	
SINE		PWM	
LFSIN		SINE	
SW+SHP		LFSIN	
SAW+		SW+SHP	
SW+DIST		SAW+	
LPCLIP		SW+DIST	
SINE+		LPCLIP	
NOISE+		SINE+	
		NOISE+	

Algorithm:124



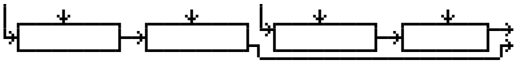
NONE	xAMP	NONE	AMP
AMP	+AMP	LOPAS2	
LOPASS	!AMP	HIPAS2	
HIPASS		LPGATE	
ALPASS		LP2RES	
GAIN		SHAPE2	
SHAPER			
DIST			
SINE			
LFSIN			
SW+SHP			
SAW+			
SW+DIST			
LPCLIP			
SINE+			
NOISE+			

Algorithm:125



NONE	*AMP*	*NONE*	*AMP*
AMP		*AMP*	
LOPASS		*LOPASS*	
HIPASS		*HIPASS*	
ALPASS		*ALPASS*	
GAIN		*GAIN*	
SHAPER		*SHAPER*	
DIST		*DIST*	
SINE		*SINE*	
LFSIN		*LFSIN*	
SW+SHP		*SW+SHP*	
SAW+		*SAW+*	
SW+DIST		*SW+DIST*	
LPCLIP		*LPCLIP*	
SINE+		*SINE+*	
NOISE+		*NOISE+*	

Algorithm:126



NONE	*AMP*	*NONE*	*AMP*
LOPAS2		*LOPAS2*	
HIPAS2		*HIPAS2*	
LPGATE		*LPGATE*	
LP2RES		*LP2RES*	
SHAPE2		*SHAPE2*	

K2600 Triple Programs: Controller Assignments

The following tables describe the controller assignments for the 10 ROM-base triple programs provided with v2. Table 2-1 lists a series of MIDI Controller numbers and their default and/or generic functions within triples. Table 2-2 lists each triple program and the specific assignments for each MIDI controller number in Table 2-1.

MIDI Controller Number	Default Function
MWheel MIDI01)	Pitch Modulation
Data (MIDI06)	Filter Attack Time
MIDI22	Filter Decay Time
MIDI23	Filter Frequency
MIDI24	Amplitude Release Time
MIDI25	Filter Resonance
MIDI26	LFO Depth (filter layers1–3, PWM layers 4–6)
MIDI27	Delay Wet/Dry
MIDI28	Reverb Wet/Dry
MIDI29	Switches Saws to PWM oscillator

Table 2-1 Default controller functions for triple programs

Program ID	Program Name	MIDI Controller	Assignment
730	Shores of Tripoli	MWheel	Sawtooth wave pitch
		Data	Lowpass filter frequency
		MIDI22	LP filter resonance (max --> min)
		MIDI23	Crossfade: partials / sawtooth
		MIDI24	Lowpass filter frequency (post-Saw)
		MIDI25	Shaper amount
		MIDI26	Resonant lowpass filter frequency
		MIDI27	Bass rolloff, treble boost (via FX)
		MIDI28	Reverb wet/dry & reverb time
		MIDI29	Attack envelope control

Table 2-2 Controller assignments for v2 factory triple programs

Program ID	Program Name	MIDI Controller	Assignment
731	Mono Triple Lead	MWheel	Pitch modulation
		Data	Shaper amount
		MIDI22	Lowpass filter frequency
		MIDI23	X gain amount
		MIDI24	Impact
		MIDI25	Chorus-delay-reverb wet/dry amount
		MIDI26	Chorus amount
		MIDI27	Delay amount
		MIDI28	Reverb time
		MIDI29	Reverb switch (Aux /insert reverb)
		MPress	Pitch modulation
732	Triple Play	MWheel	Pitch modulation
		Data	LFO on/off, filter frequency
		MIDI22	Filter resonance
		MIDI23	Pitch (Saw 2)
		MIDI24	AmpMod amount
		MIDI25	Notch2 frequency
		MIDI26	Notch2 amplitude
		MIDI27	Reverb wet/dry
		MIDI28	Pitch detune
		MIDI29	Delay mix
		MPress	Pitch modulation
733	ABCDE = ADSR+Res	MWheel	Pitch modulation
		Data	Filter attack time
		MIDI22	Filter decay time
		MIDI23	Filter frequency
		MIDI24	Amplitude release time
		MIDI25	Filter resonance
		MIDI26	LFO depth (filter layers1-3, PWM layers 4-6)
		MIDI27	Delay wet/dry
		MIDI28	Reverb wet/dry
		MIDI29	Switches saws to PWM oscillator

Table 2-2 Controller assignments for v2 factory triple programs

Program ID	Program Name	MIDI Controller	Assignment
734	*StringMod Pad*	*MWheel*	*Modulating sawtooth, flange rate (FX)*
		Data	*Filter frequency*
		MIDI22	*Amplitude modulation depth*
		MIDI23	*Attack envelope control*
		MIDI24	*Frizzle distortion*
		MIDI25	*Aux reverb send & aux reverb time*
		MIDI26	*Bus2 reverb wet/dry*
		MIDI27	*Flange feedback*
		MIDI28	*Aux Reverb wet/dry & attenuation*
		MIDI29	*Flange rate shift*
735	*Plucky Emu*	*MWheel*	*Vibrato*
		Data	*Hi-freq stimulation / lowpass filter frequency*
		MIDI22	*Shaper & filter frequency / resonance*
		MIDI23	*Decay envelope*
		MIDI24	*Release envelope*
		MIDI25	*Aux reverb send*
		MIDI26	*Reverb decay time*
		MIDI27	*Bus2 Reverb wet/dry, stereo delay level*
		MIDI28	*FX2 input width, treble attenuation*
736	*Mad Three-Oh*	*MWheel*	*Pitch modulation*
		Data	*Filter frequency*
		MIDI22	*Filter resonance, separation, distortion, hipass gain*
		MIDI23	*Envelope filter frequency*
		MIDI24	*Filter envelope decay*
		MIDI25	*XFade saw/square*
		MIDI26	*LFO rate (for pitch modulation)*
		MIDI27	*(Aux) Plate reverb amount*
		MIDI28	*Delay mix*
		MIDI29	*Flange mix, insert reverb wet/dry*
		MPress	*Pitch modulation*
737	*Dastardly Drums*	*Data*	*Filters, crossfades*
		MIDI22	*Lowpass filters*
		MIDI23	*Various boosts, cuts, filter control*
		MIDI24	*Drum-piano intermodulation*
		MIDI25	*Aux reverb sends*
		MIDI26	*Hollow Laserverb effect*
		MIDI27	*Aux reverb time & HF damping*
		MIDI28	*FX distortion Level*
		MIDI29	*Switch hi-hats to shaped noise waveform*

Table 2-2 Controller assignments for v2 factory triple programs

Program ID	Program Name	MIDI Controller	Assignment
738	H.Sync Rhythm	MWheel	Clocked pitch effect depth
		Data	Lowpass frequency
		MIDI22	Lowpass2 freq / Parametric EQ frequency
		MIDI23	Lowpass2 Gain / Shaper Amount
		MIDI24	Crossfade master / slave
		MIDI25	Stereo delay level
		MIDI26	Stereo delay feedback
		MIDI27	Chorus depth
		MIDI28	Chorus feedback
		MIDI29	Switch to alternative program

Table 2-2 Controller assignments for v2 factory triple programs

Alphanumeric Buttonpad Entries for DSP Functions

The tables on the following pages list the entries that you can use on the K2600's alphanumeric buttonpad to select DSP functions without having to scroll through a long list of values. Table 2-3 is sorted alphabetically by the name of the DSP function, and

When you're on the ALG page, use the cursor buttons to select the DSP block whose function you want to change, then type the numeric value corresponding to the function you want to select. Press **Enter** after typing the value. For example, to select a DSP function of **None** in a single-block function, type **6**, **0**, **Enter**.

Special Cases

In some algorithms, there are nonstandard DSP blocks containing specialized functions. These functions have different numeric entries, as listed below. In all other cases, numeric entries select DSP functions as listed in Table 2-3 and Table 2-4.

Algorithm and DSP Block	Numeric Entry	DSP Function
F1 in Algorithm 126	105	xAMP
	106	+AMP
	107	!AMP
F2 in Algorithms 123 and 124	108	LP2RES
	109	SHAPE2
	110	BAND2
	111	NOTCH2
	112	LOPAS2
	113	HIPAS2
	114	LPGATE

Function Block Size	DSP Function	Numeric Entry
Single-block	!AMP	75
	+AMP	49
	+GAIN	42
	ALPASS	17
	AMP	110
	AMPMOD	44
	BAND2	35
	DIST	20
	GAIN	18
	HIPAS2	52
	HIPASS	16
	LF SAW	28
	LF SINE	24
	LF SQR	30
	LOPAS2	37
	LOPASS	15
	LP2RES	73
	LPCLIP	70
	LPGATE	57
	MASTER	77
	NOISE+	76
	NONE	60
	NOTCH2	36
	PWM	22
	SAW	27
	SAW+	26
	SHAPE2	74
	SHAPER	19
	SINE	23
	SINE+	71
	SLAVE	78
	SQUARE	29
	SW+DST	53
	SW+SHP	25
	WRAP	31
	XFADE	43
	xAMP	48
	xGAIN	41

Table 2-3 Buttonpad entries for DSP functions (sorted alphabetically)

Function Block Size	DSP Function	Numeric Entry
Two-block	+ SHAPE MOD OSC	67
	2PARAM SHAPER	64
	2POLE ALLPASS	5
	2POLE LOPAS	2
	AMP MOD OSC	72
	AMP U AMP L	38
	BAL AMP	39
	BAND2 GAIN	85
	BANDPASS FILT	3
	HIPAS2 GAIN	84
	LOPAS2 GAIN	83
	LPGATE GAIN	89
	LPRES2 GAIN	87
	NONE	61
	NOTCH FILT	4
	NOTCH2 GAIN	86
	PARA BASS	8
	PARA MID	51
	PARA TREBLE	9
	SHAPE MOD OSC	68
	SHAPE2 GAIN	88
	x SHAPE MOD OSC	66
Three-block	4 POLE HIPASS W/SEP	54
	4 POLE LOPASS W/SEP	50
	DOUBLE NOTCH W/SEP	56
	HIFREQ STIMULATOR	44
	NONE	62
	PARAMETRIC EQ	13
	STEEP RESONANT BASS	14
	TWIN PEAKS BANDPASS	55

Table 2-3 Buttonpad entries for DSP functions (sorted alphabetically)

Function Block Size	*Numeric Entry*	*DSP Function*
Single-block	*15*	*LOPASS*
	16	*HIPASS*
	17	*ALPASS*
	18	*GAIN*
	19	*SHAPER*
	20	*DIST*
	22	*PWM*
	23	*SINE*
	24	*LF SINE*
	25	*SW+SHP*
	26	*SAW+*
	27	*SAW*
	28	*LF SAW*
	29	*SQUARE*
	30	*LF SQR*
	31	*WRAP*
	35	*BAND2*
	36	*NOTCH2*
	37	*LOPAS2*
	41	*xGAIN*
	42	*+GAIN*
	43	*XFADE*
	44	*AMPMOD*
	48	*xAMP*
	49	*+AMP*
	52	*HIPAS2*
	53	*SW+DST*
	57	*LPGATE*
	60	*NONE*
	70	*LPCLIP*
	71	*SINE+*
	73	*LP2RES*
	74	*SHAPE2*
	75	*!AMP*
	76	*NOISE+*
	77	*MASTER*
	78	*SLAVE*
	110	*AMP*

Table 2-4 Buttonpad entries for DSP functions (sorted numerically)

Function Block Size	*Numeric Entry*	*DSP Function*
Two-block	*2*	*2POLE LOPAS*
	3	*BANDPASS FILT*
	4	*NOTCH FILT*
	5	*2POLE ALLPASS*
	8	*PARA BASS*
	9	*PARA TREBLE*
	38	*AMP U AMP L*
	39	*BAL AMP*
	51	*PARA MID*
	61	*NONE*
	64	*2PARAM SHAPER*
	66	*x SHAPE MOD OSC*
	67	*+ SHAPE MOD OSC*
	68	*SHAPE MOD OSC*
	72	*AMP MOD OSC*
	83	*LOPAS2 GAIN*
	84	*HIPAS2 GAIN*
	85	*BAND2 GAIN*
	86	*NOTCH2 GAIN*
	87	*LPRES2 GAIN*
	88	*SHAPE2 GAIN*
	89	*LPGATE GAIN*
Three-block	*13*	*PARAMETRIC EQ*
	14	*STEEP RESONANT BASS*
	44	*HIFREQ STIMULATOR*
	50	*4 POLE LOPASS W/SEP*
	54	*4 POLE HIPASS W/SEP*
	55	*TWIN PEAKS BANDPASS*
	56	*DOUBLE NOTCH W/SEP*
	62	*NONE*

Table 2-4 Buttonpad entries for DSP functions (sorted numerically)

Chapter 3
Disk Partitioning

Overview

Prior to v2.0, the K2600 could address a maximum 2 G (gigabytes) of space on a SCSI hard disk. Although you could format a disk of any size for use with the K2600, only the first partition—up to 2 G—would be available to the K2600.

The K2600 can now address up to 8 G of hard-disk space, in 2-G partitions. This is true for any hard disk formatted with the DOS-compatible FAT-16 format. Hard disks larger than 8 G can be formatted to make 8 G (in four partitions) accessible to the K2600.

When properly partitioned, the K2600 files on your hard disk should be accessible to both the K2600 and your personal computer, if you're using one. Using a personal computer makes it much easier to manage your K2600 files. It also enables you to use other disk management tools on your K2600 partitions—for example, disk defragmenters and system integrity checkers.

We strongly recommend that you partition your hard disk from the K2600, although it's possible to partition it from a PC (see page 3-4).

***Note**: Keep in mind that if you partition a hard disk, K2600s without v2.0 (as well as K2000s and K2500s) will be able to use only the first partition of that disk.*

The remainder of this chapter consists of the following sections:

Partitioning Your Hard Disk

Once you've connected a hard disk drive to one of the K2600's SCSI ports, you're ready to begin partitioning. If you need information about using your K2600 and a PC together in a SCSI chain, see the following sections in your *Musician's Guide* (Rev A):

- Connecting a SCSI Device 13-3
- SCSI Termination 13-3
- Using your K2600 in a SCSI System 13-4

Partitioning is part of the formatting procedure. When you format a disk from the 2600, you can choose whether to add partitions. When partitioning a disk, the K2600 creates as many 2-G partitions as possible, to a maximum of 8 G (even if your disk is larger than 8 G, the K2600 formats only the first 8 G).

***Note**: If you happen to have a 2-G hard disk, you might wonder whether it's worth the effort to partition it. If you want the disk to be fully DOS/Windows-compatible, you should partition it. If DOS/Windows compatibility isn't important, partitioning isn't necessary (for example, when you know you'll never use the hard disk with a computer).*

The last partition includes whatever disk space (up to the 8-G limit) remains after the last full 2-G partition is formatted. The last partition is usually slightly less than 2 G, because each partition contains some data for disk management. For example, if you format a new 10-G hard disk from the K2600, you end up with three 2-G partitions, and a fourth partition of slightly under 2 G. The K2600 doesn't format the remaining 2 G. You may be able to use a personal computer to format the remaining space for use with that computer. See *Partitioning Large Disks* on page 3-5.

***Note**: There are two schools of thought about describing the size of storage devices. Some people think of a kilobyte as 1,000 bytes, and others think of it as 1,024 bytes (which is the actual number). If you consider a kilobyte to be 1,000 bytes, then a gigabyte is $1,000^3$, or a billion bytes. If you think of a kilobyte as 1,024 bytes (as we do), a gigabyte is $1,024^3$, or 1,073,741,824 bytes. So a 2-G partition stores about 2.15 billion bytes, which you might prefer to think of as 2.1 G.*

Doing a Hard Format

There are two kinds of formatting: we'll call them normal formatting and "hard" formatting. Normal formatting is what usually happens when you format a disk. Although normal formatting makes an entire hard disk accessible for data storage, it doesn't actually *delete* files; instead, normal formatting deletes underlying system-level data about how and where the files are stored. Although this makes the files on the disk unusable, much of the files' contents remain on the disk (that's why you can often recover a file even after you've deleted it).

Hard formatting physically erases every bit of data on the disk. Once a disk is hard formatted, there's no data to recover; it's completely blank, and must be formatted normally before it can store data again. Hard formatting can sometimes make a corrupted disk usable, or make it easier to use a disk on a different operating system.

The *first* time you format a disk for use with the K2600—whether it's new or previously formatted with a computer—you should begin with a hard format, then format it normally with the K2600. (You should do this because any residual file data left by normal formatting could cause the K2600 to miscalculate partition boundaries.)

Any time you're using a PC or a Mac to format or reformat a disk for use with the K2600, you should start with a hard format to remove residual file data.

When using the K2600 to reformat a disk or partition that's been formatted *with the K2600*, you can probably skip the hard format, unless you're having problems saving and loading files.

The time required for hard formatting varies widely depending on the hard disk. Newer drives in the 10-G range will generally take from 30 to 60 minutes, although your results may differ.

1. Press the **Disk** mode button to enter Disk mode.
2. Select the Current Disk parameter, if necessary, then select the SCSI ID of the disk you want to format.
3. Press the **Chan/Bank** buttons simultaneously. You'll see several prompts asking you whether you want to do a hard format.
4. Answer **Yes** in each case. The display informs you when formatting begins. When it's finished, the display returns to Disk mode.

The Partitioning Procedure

⚠ ***Caution**: This procedure destroys all data files on the disk. Before formatting, make sure to back up any data that you want to preserve.*

1. If you haven't already done a hard format, do so now, as described in the preceding procedure. Doing a hard format is important for ensuring data integrity. When the hard format is complete, the K2600 returns to Disk mode, and you can proceed with Step 2.
2. Press one of the **more** soft buttons until you see the **Format** soft button. Press it. The display prompts you to decide whether to format the disk. Press **Yes**.
3. The display prompts you to choose whether to *partition* the disk. Press **Yes**.
4. The display prompts you with a reminder that formatting erases everything on the hard disk, and asks whether you want to continue. Press **Yes**.
5. The display shows `Formatting Drive X with N Partitions`. `X` is the SCSI ID of the disk, and `N` is the number of partitions that the disk will contain. This is the last chance to abort the formatting process. If you do a soft reset while this message is visible, you'll cancel the format. Each partition takes about 30 seconds to format.

When Disks Are Already Partitioned

If the hard disk you've selected for formatting is already partitioned, the display you see at Step 2 prompts you to decide whether to reformat the entire disk. Press **Yes** if you want to reformat the whole disk, then proceed with Step 3.

If you don't want to reformat the entire disk, press **No** at Step 2. The display prompts you to decide whether to format a single partition. Press **Yes**, and the display prompts you to select a partition to format. By the way, this is a quick method for deleting all the files in a single partition.

Partitioning With a Computer

You can partition disks for the K2600 using a DOS-capable computer that's compatible with the FAT-16 file format. This works well on DOS and Windows machines (consequently that's the focus of this description). We don't have any reliable recommendations for partitioning disks with a Macintosh.® Nevertheless, there's some Mac-specific information beginning on page 3-7.

Partitioning with a PC gives you greater flexibility in defining the number and size of partitions on your hard disk. Unless this flexibility is extremely important to you, however, we strongly recommend that you use the K2600 to partition your hard disk for use with either a PC or a Mac. Our tests show that this provides the greatest cross-platform compatibility.

For example, partitions created by the K2600 are visible to any PC that's compatible with the FAT-16 file format. These partitions are also visible to any Mac running System 8.5 (or later), and version 3.0.2 (or later) of the File Exchange extension. On the other hand, partitions created on a PC are less likely to be accessible on older Macs.

Formatting from the PC requires **FDISK**, a disk-partitioning utility that's built into Windows 95 and Windows 98.

1. If you haven't already done a hard format, do so now, as described on page 3-3. Doing a hard format is important for ensuring data integrity. When the hard format is complete, the K2600 returns to Disk mode, and you can proceed with Step 2.
2. On the PC, start **FDISK** by typing **FDISK** at a DOS command prompt (or in the Run menu). **FDISK** prompts you to decide whether to enable support for large hard disks.
3. Respond with **N** (No), because you don't want support for large hard disks (although **FDISK** doesn't mention it by name, large-hard-disk format is *FAT-32*—answering **No** specifies *FAT-16* format, which is compatible with the K2600).
4. Use Menu Option 5 to select the drive to format.
5. Use Menu Option 1 to create a primary partition. Specify a size up to 2 G.
6. When the primary partition is complete, choose Option 2 to create an extended partition. At the prompt, select any size (up to the 8 G limit; including the primary partition), or select the default size to use the remainder of the drive for the extended partition. Press **Escape** to begin partitioning. When the partition is complete, you'll have the opportunity to create one or more logical drives within the extended partition (as described in Step 7). The K2600 recognizes each of these logical drives as a separate partition).
7. When the extended partition is complete, you'll see a prompt asking you to specify the size for a logical drive. Specify a size up to 2 G, then press **Escape**. When the logical drive is complete, the prompt returns. Repeat this step until you've used the remaining space in the extended partition. Each of these partitions (logical drives) gets a drive letter as you create it. Keep track of these, because you'll need to format each partition. At any time in this process, you can use FDISK Menu 3 to display the current list of partitions.
8. When you're finished partitioning, press escape several times to exit **FDISK**.
9. Restart the computer.
10. Format each of the new partitions (using the drive letters you created with Steps 5 through 7). You can do this from the DOS prompt, or from Windows Explorer.

Partitioning from a PC is more time-consuming, but gives you more control over the number and size of your partitions.

Partitioning Large Disks

The K2600 can address a maximum of 8 G of disk space, so there's not much point in connecting a larger disk to the K2600—unless you like to do a bit of hacking, in which case you may be able to use a personal computer to format the extra space for use with that computer.

There are several disk-formatting utilities available for the Mac and PC. You may be able to use one of these (including **FDISK** for the PC) to format a large hard disk with 8 G of space for the K2600, and additional space that's accessible to your computer.

Working With Partitions

Once your hard disk is formatted and partitioned, you should verify each partition before attempting to store files. The quickest way to do this is to execute a Load command for each partition. If the K2600 can read the disk (even though there are no files), the partition is OK. The display informs you if there's a problem addressing the partition.

One you've verified each partition, you can start using the disk with your K2600, and optionally with a personal computer. If your disk is attached to the K2600 and nothing else, you'll interact with the hard disk exclusively through Disk mode. If you're in Disk mode and you can't get to a partition, try reformatting the partition (see *When Disks Are Already Partitioned* on page 3-3). If that doesn't work, do a hard format on the disk, then format and partition it again.

If your hard disk is also attached to a computer, the computer should be able to address each partition. PCs can address each partition as an individual drive (at a DOS prompt or via icons in Windows), while Macs display a file icon for each partition.

Having a computer hooked up to the hard disk gives you another way to verify the formatting and partitioning of the disk. Look at the disk from the K2600, then from the computer. If they don't have the same partition data (number of partitions, size of each partition), you could have a formatting problem. In this case, you should start over by doing a hard format, then proceeding to formatting and partitioning.

Managing a SCSI Chain that Includes a Computer

If you create a SCSI chain that contains both a K2600 and a computer, you'll need to use caution to prevent file corruption. We'll explain how this risk of file corruption occurs, then describe how to avoid it.

When you boot up your computer, it addresses the SCSI devices to which it's connected, and attempts to mount every SCSI device whose format it recognizes. While mounting a disk, the computer examines the disk's file allocation table (FAT). The FAT contains the locations of all the data on the disk. Consequently, the FAT gets updated every time you write data to the disk. Here's where the risk of file corruption occurs.

When you write to your hard disk from the K2600, the K2600 updates the disk's FAT. But there's no way to instruct the computer that the FAT has been changed, so the K2600 and the computer have different versions of the disk's FAT. If you write to the disk from the computer at this time, you run the risk of writing over the data you just stored from the K2600.

You can eliminate this risk by ensuring that the K2600 and the computer have the same version of the disk's FAT before writing from the computer. The most foolproof way to ensure this is to adopt the following rule: *Whenever* you write to the hard disk from the K2600, unmount and remount that disk before writing to it from your computer. If it's a removable-media drive, ejecting the disk and reinserting it will update the computer's version of the FAT.

You might be able to unmount and remount the hard disk from the desktop, or using a software utility—or you might have to restart the computer. If you're using Windows, you can probably designate the hard disk as a removable-media disk (as described below), which is convenient because Windows machines always read the FATs of removable-media disks before writing to them.

Designating a Disk as a Removable-Media Disk (Windows only)

1. On your PC, run the Device Manager.
2. Expand the Disk Drives category.
3. Double click icon for the K2600's internal hard disk to bring up the Properties dialog.
4. Select the **Settings** tab.
5. Check the **Removable** checkbox.

Configuring Device Drivers (Windows 95 and 98 Only)

PCs running Windows 95 or Windows 98 require a SCSI driver for every device in a SCSI chain. There is no SCSI driver for the K2600, because it's not required for normal SCSI operations. Consequently, the first time you boot up your PC after connecting it to the K2600 via SCSI, you'll need to do some configuring to enable the PC to recognize the K2600. This configuration isn't required, but it enables the PC and the K2600 to interact more smoothly.

A Bit of Background

When you start up your PC, Windows scans the PC's SCSI port(s) for connected devices. For each SCSI device it finds, Windows checks a set of entries in the Device Manager, to verify that the appropriate SCSI drivers are installed. Windows 95 and 98 require that the Device Manager contains an entry for each SCSI Logical Unit of each SCSI device. The K2600 has eight possible SCSI IDs (0–7), so your Device Manager should contain eight entries corresponding to the K2600. Each of these entries must specify a valid SCSI driver—otherwise Windows will report one or more unknown devices as it boots up.

The K2600 requires no SCSI drivers for disk operations, but Windows 95 and 98 require them nonetheless. You can satisfy this requirement by specifying a generic "Unsupported Device" Windows driver for each of the K2600's entries in the Device Manager. The following procedure describes how.

The Configuration Procedure

We'll assume you've already connected your K2600, your PC, and your hard disk in a SCSI chain (with whatever SCSI termination is required), and that everything is powered down.

1. Power up the hard disk. Wait until it's fully up to speed.
2. Power up the K2600. Wait until it reaches Program mode.
3. Power up your PC. During the boot-up process, Windows detects the K2600 as an unknown SCSI device. The Update Device Driver Wizard appears, prompting you to specify a SCSI device driver to install.
4. Make sure that the PC's floppy and CD-ROM drives are empty and click **Next—**that is, proceed without specifying a SCSI driver.

5. The PC displays a dialog informing you that Windows did not locate a driver for the SCSI device. Click **Finish** to proceed without installing a driver.
6. Repeat Steps 4 and 5 seven times—once for each of the K2600's remaining SCSI Logical Units. When you finish the last dialog, Windows finishes booting up.
7. Start the Device Manager (in the Start menu, select **Settings**, then **Control Panels**).
8. Select the **System** icon in the Control Panels window.
9. Select the **Device Manager** tab in the System Properties window.
10. Click the icon for **Other Devices**. You'll see the eight entries for the K2600. Note the exclamation marks, indicating problems with the corresponding device drivers.
11. Double-click on one of the entries. This opens the Properties window for that entry.
12. Click the **Drivers** tab. The status for this entry indicates that the driver isn't working properly, and that the driver files either are not required or haven't been loaded.
13. Click **Update Driver**. This opens the Update Device Driver Wizard, which asks whether Windows should search for the driver.
14. Select the option **No, select driver from list**, then click **Next**. A list of driver types appears. The default is **Other devices**.
15. Click **Next** to accept the default. A list of devices appears. The default is **Unsupported device**.
16. Click **Finish**. This "installs" the driver and returns to the Properties window for the current entry.
17. Click **Close**, which returns you to the Device Manager.
18. Repeat Steps 11 through 17 for the remaining entries.
19. Close the Device Manager and Control Panels window.

There's a page on our website where you can find more information about including a K2600 and a PC in a SCSI chain:

http://www.kurzweilmusicsystems.com/html/scsi_help.html

If a Partition is Inaccessible

On the PC

You're not likely to have problems addressing K2600 partitions from your PC, especially if you created those partitions on the K2600 itself. If you can't mount a partition on your PC, try reformatting the partition, using the K2600. If that doesn't work, do a hard format, then format and partition the disk again.

On the Mac

System 8.5 or later—when used with the File Exchange extension—supports DOS partitions, so you should be able to view partitions as file icons. If they aren't visible, try reformatting the partitions using the K2600

If you're running an operating system older than 8.5, your Mac will probably not be able to address the last partition on the disk. If you have the Formatter 5 utility from Software Architects, you may be able to make the last partition visible (we can't guarantee that this will work):

1. Start with a hard format (see the procedure on page 3-3).
2. Format the disk with Formatter 5 (see the Formatter 5 instructions).
3. Use Mac/PC manager (a utility provided with Formatter 5) or DOS Mounter 95 (also available from Software Architects) to view the partitions.

Other Macintosh Issues

There are several things to keep mind when attempting to use a Mac with disks formatted for the K2600.

First, we recommend using MacOS 8.5 or later, and version 3.0.2 (or later) of the File Exchange system extension. This combination has given the best results, especially for disks formatted with the K2600. With this configuration you can read most partitioned DOS disks, but you can't format them.

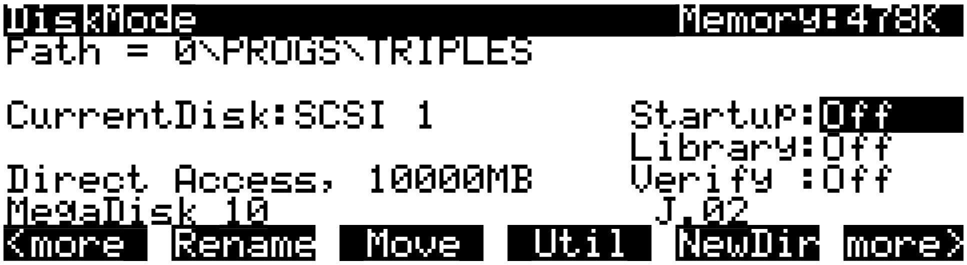
On Macs running operating systems prior to 8.5, you can use Formatter 5 (and its accompanying extension, Mac/PC manager). these work reasonably well for formatting and reading small disks. Results are unreliable for disks larger than 2 G, however. Formatter 5 often has trouble reading any partition but the first, even on disks formatted with the K2600. You might also try DOS Mounter 95, also available from Software Architects.

Disk Mode

Disk mode has changed slightly to accommodate support for partitions. The changes affect several operations to varying degrees. We've already discussed the changes to the formatting/partitioning procedure (see page 3-1); this section describes the changes to other Disk-mode operations.

Basic Operations

This is how the Disk mode page looks when you're viewing a hard disk. Note the partition ID (**0**, which indicates the first partition), which is included in the path information. The partition ID for unpartitioned disks is always **0**. There's no real limit to the number of partitions your disk can contain; the K2600 should be able to read all of them. Nevertheless, 20 or so partitions is a reasonable maximum.



```
DiskMode                                  Memory:478K
Path = 0\PROGS\TRIPLES

CurrentDisk:SCSI 1         Startup:Off
                           Library:Off
Direct Access, 10000MB     Verify :Off
MegaDisk 10                  J.02
<more   Rename   Move   Util   NewDir   more>
```

The K2600 records path information up to the disk level. In other words, the K2600 tracks your movement within directories and partitions (but not from disk to disk). As long as you're addressing the same disk, the K2600 has all the information it needs to load or save files (or any other Disk-mode command).

When you select a different disk, the first command you execute causes the K2600 to prompt you to select a partition. Once you do so, you won't see the prompt again until the next time you change disks and execute another command.

The following table summarizes the changes (if any) in the twelve disk operations:

Load	*Support for selecting different partitions; see Navigation on page 3-9.*
Save	*Support for selecting different partitions; see Navigation on page 3-9.*
Macro	*Macro IDs include partition IDs; supports multi-disk macro files. See page 3-11.*
Delete	*Support for selecting different partitions; see Navigation on page 3-9.*
Backup	*Support for backups across partitions, and between directories within partitions. See page 3-10.*
Copy	*Support for copies across partitions, and between directories within partitions. See page 3-10.*
Sleep	*Unchanged (puts current disk to sleep, or ejects removable media).*
Format	*Support for multiple partitions; see Partitioning Your Hard Disk on page 3-1.*
Rename	*Support for selecting different partitions; see Navigation on page 3-9.*
Move	*Unchanged (you can't move files across partitions; see Backups and Copies)*
Util	*Several changes; see Utilities on page 3-10*
NewDir	*Support for selecting different partitions; see Navigation on page 3-9*

Navigation

Many disk-operation dialogs (the Save dialog, for example) include file lists, where you select the file to save, delete, copy, or whatever. Any time you're navigating through the directories in a file list, you can select a different partition.

1. Press the **Root** soft button to select the root directory for the current partition. The path information in the top line of the display will look something like Dir:0\. The 0 (or some other numeral) is the partition ID, and the backslash (\) denotes the root directory. (You can also press **Parent** one or more times to get to the root directory.)
2. Press **Root** or **Parent** to display the partition selector, which prompts you to select a partition to mount.
3. Select a partition ID, then press **OK**.

The quickest way to get to the partition selector is to press **Root** twice in succession. This brings you to the partition selector from anywhere in the file list.

Backups and Copies

You can use the Backup and Copy operations to copy files to different partitions on the same disk, or even to some directories within the same partition (prior to version 2, Backup and Copy worked only from one SCSI device to another).

You can always back up or copy files to a different partition, regardless of location. If you're backing up or copying to the same partition, there's a restriction: you can't back up or copy to the same directory, or to a subdirectory of the current directory (you *can* back up and copy to parent directories, however).

For example, suppose you had several program files in the root directory of a partition, and you wanted to copy some of them to a directory called PROGS, also in the root directory. You couldn't use Backup or Copy to make copies of the programs in the PROGS directory, but you could use Move to *move* them into PROGS, then use Backup or Copy to make copies of them in the root directory.

Utilities

Pressing the **Util** soft button displays the DiskUtil page, which now includes a soft button labeled **Part'n** (if your hard disk is 2 G or smaller, you won't see the **Part'n** soft button). Pressing this button displays a prompt asking you to select a partition (like pressing **Root** or **Parent** when you're in the root directory).

Info

The disk report that you see when you press the **Info** soft button includes the total number of K2600 partitions on the disk, and which partition you're currently viewing. This information is on the last line of the second page of information.

Free

Pressing the **Free** soft button displays the free space on the current partition (not on the entire disk, as with earlier K2600 operating systems). There are three components to the free-space information: percentage used, kilobytes free, and total kilobytes.

If you change partitions while the free-space information is visible, the K2600 automatically calculates and displays the free-space information for the new partition. Exiting the DiskUtil page disables this automatic updating until the next time you use the Free utility.

List

The List utility is a convenient way to view the directories and files on your hard disk. It's also a time saver for disk operations. When you press **List**, the display shows a hierarchical listing of all the files and directories in the current partition (this can take a while, depending on the size and complexity of your directories). Directory names are followed by a backslash (\).

Use the **Up** and **Down** cursor buttons to select different objects in the list. The index number on the far right of the display's top line shows where you are in the list (each object is numbered according to its position in the directory hierarchy). Using the cursor buttons, you can quickly find files on the hard disk without having to change directories.

Changing partitions while using the List utility resets the index number to 1. In other words, no matter where you are in the list for the current partition, if you change partitions you'll start at the first entry in that partition's list.

You can use the alphanumeric buttonpad to navigate through the list. Press the index number you want to view, then press **Enter**.

Use the **Up** and **Down** *soft* buttons (not the cursor buttons) to navigate through the partition. Pressing **Down** when a directory is highlighted takes you into that directory (pressing **Down** when a file is highlighted doesn't do anything). Pressing **Up** always takes you to the parent directory of the current directory (if you're at the partition's root directory, the K2600 prompts you to select a partition.). As you've probably guessed, pressing **Root** takes to the root directory of the current partition.

Press the **Go To** button to change the current path to the highlighted directory (or to the directory that contains the highlighted file). You'll save a lot of time if you use the List utility to navigate to the desired directory before executing a command (instead of executing the command first, then navigating to the desired directory in the resulting dialog).

Macros

To support partitions, each macro entry now has a partition ID, which is recorded when you create the macro. The partition ID is part of the macro entry's name; it appears after the SCSI ID. You can view macro entries (if there's a current macro table) by pressing the **Macro** soft button while in Disk mode.

When editing a macro entry (using the Modify option on the MACRO function page in Disk mode), you can edit the partition ID along with other properties of the entry. If you change the partition ID of a macro entry, make sure it's a valid partition ID.

Support for Multiple Removable Disks

This convenient enhancement enables you to use a single macro to load files from different removable disks.

1. In Disk mode, press **Load** to begin the Load dialog. You'll see a prompt asking you to select a file to load.
2. Select the macro file to load, and press **OK**. The K2600 displays a prompt asking you to select a bank for loading the macro.
3. Select a bank (or select **specified** or **Everything**), and press **OK**.
4. The K2600 begins loading files. When it tries to load a file that's not on the disk, the display shows you the name of the file that it couldn't find. Press **OK**.
5. Another prompt asks you if you want to retry loading after changing disks. Press **Yes**.
6. Change the removable disk, and press **OK**, as instructed by the prompt in the display. The K2600 resumes loading files according to the macro entries.
7. Repeat Steps 4 through 6 as often as necessary to load all the files specified by the macro. The display returns to the Disk-mode page when loading is complete.

Index