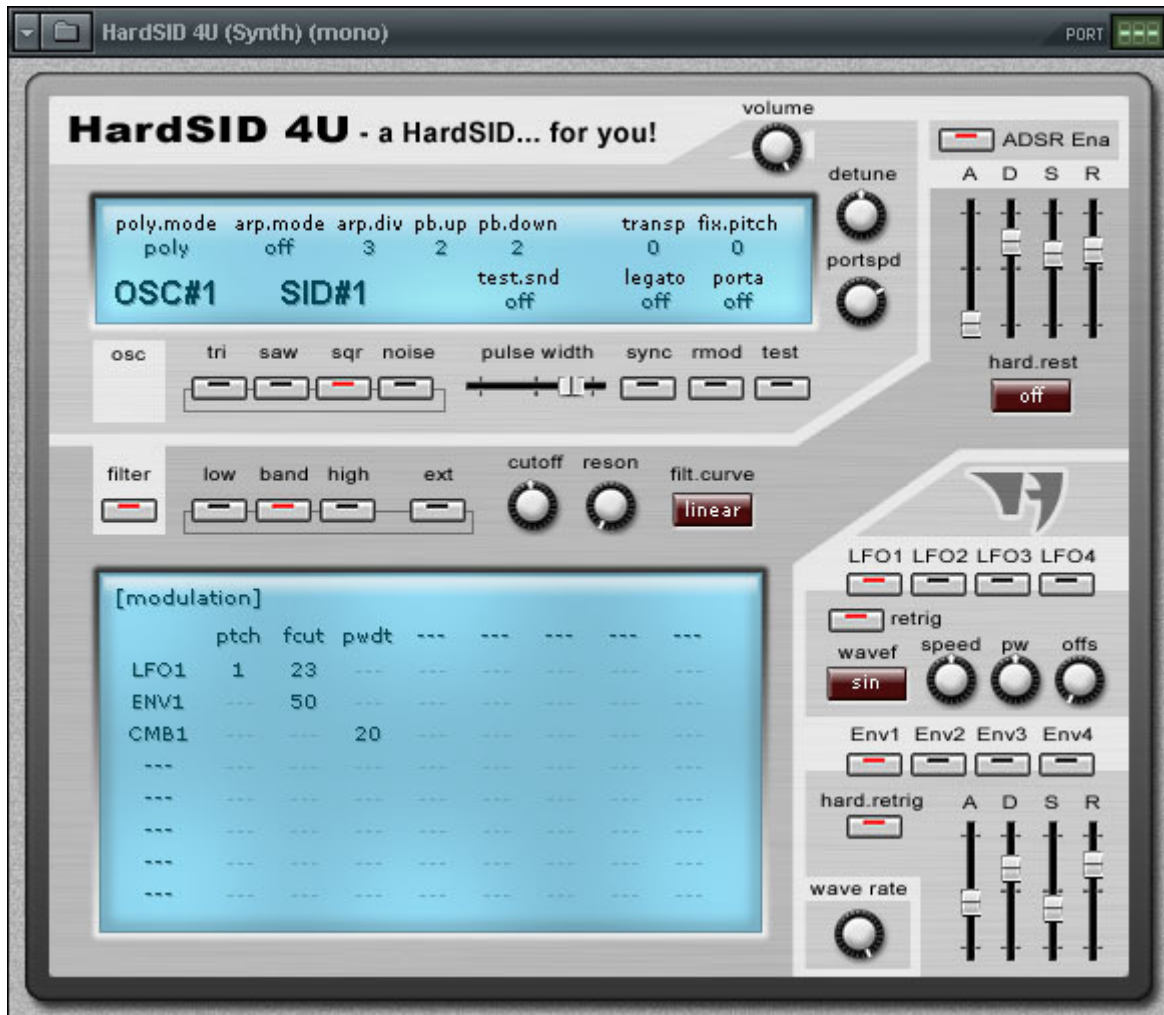


The HardSID 4U VSTi



About the HardSID 4U VSTi.....	4
Installation:	4
Using multiple SIDs:	4
The main LCD screen.....	4
The SID selector (SID#1, SID#2, SID#3, SID#4):	4
The Oscillator selector (OSC#1, OSC#2, OSC#3):	4
Poly mode:	5
Arp mode:	5
Arp div:.....	5
PB up, PB down:.....	5
Transp:	5
Fix pitch:.....	6
Porta:.....	6
Legato:	6
Test snd:.....	6
Other oscillator controllers.....	6
Detune knob:	6
Portspd knob:.....	6
Filter toggle:	6
tri / saw / sqr / noise toggles:	6
Pulse width slider:.....	7
Sync toggle:	7
Rmod toggle:.....	7
Test toggle:.....	7

ADSR sliders:	7
ADSR Ena toggle:.....	7
Hard rest:	7
Global SID controllers	8
Volume:.....	8
Filter:	8
Filter mode toggles:	8
Cutoff knob:	8
Reson knob:.....	8
Filt curve selector:.....	8
External audio filtering toggle:.....	8
Parameter modulation.....	8
LFOs (LFO1-4)	8
Envelopes (Env1-4).....	9
Wave rate knob.....	9
The modulation LCD screen	9
Modulation matrix editor:	10
Modulation sources:	10
Modulation targets:	11
Combiners editor:	12
Operations:	13
Table editor:.....	13
Basics.....	14
Inserting and removing a complete row.....	14
Optimizing your Windows for music production.....	15

About the HardSID 4U VSTi

The HardSID 4U VSTi is a VST plugin that can be used with many VST host applications, just like Cubase, FL Studio, etc... It is not a simple MIDI remote control application, but a fully featured replaceable/upgradeable synthesizer engine for the HardSID 4U. It renders SID register data and passes down to the HardSID 4U hardware at 8000Hz data rate via USB.

You can use the analog SID outputs on the HardSID 4U unit if you have to route the sound back to your soundcard for further processing (if you want to apply VST effects on it for example). The VSTi was designed to run in ASIO driven VST hosts with low buffer sizes (low latency).

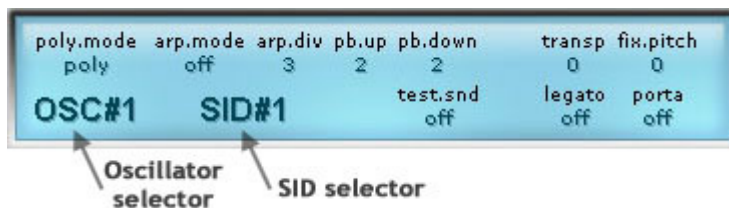
Installation:

Copy the VSTi's dll into your VST plugin directory where your VST host will look for it. You may also want to check out the 'Optimizing your Windows for music production' appendix.

Using multiple SIDs:

The HardSID 4U VSTi can run in up to four instances inside your VST host application. Each instance handles one SID (three oscillators, one filter). So, you can use up to four single-SID synthesizers at once if you have four installed SIDs in your HardSID 4U. This way you can use up to 12 oscillators and three filters. Each single-SID synthesizer listens to up to three MIDI channels depending on the selected polyphony mode.

The main LCD screen



The SID selector (SID#1, SID#2, SID#3, SID#4):

Each VSTi instance needs one SID to talk to. Each instance tries to start with SID#1 and displays '#1: NA' if some other instance is already using that SID.

In this case the first thing you have to do after starting the VSTi instance is to select the SID you want to use with it (just click on the SID selector on the LCD screen and select a SID from the popup menu). If the requested SID is currently used by another VSTi instance, the selector will display '#2: NA' (if you requested SID#2 for example).

If the SID has been successfully selected, the selector displays 'SID#2' (if you requested SID#2 for example).

The Oscillator selector (OSC#1, OSC#2, OSC#3):

Each VSTi handles one SID, so each instance has three oscillators.

The user interface of the VSTi lets you to adjust the parameters of one oscillator at once. You can select the oscillator you want to edit by clicking on the oscillator selector (OSC#1). All oscillator-specific controls will change according to the selected oscillator's current parameter values.

Global controls that does not belong to oscillators will not change. (Global controls are: the SID selector, poly mode, volume, filter mode selectors, filter cutoff/resonance/curve controls, test sound toggle.)

Poly mode:

Poly mode determines the polyphony mode of the VSTi instance.

poly:

The VSTi instance listens to MIDI channel 1 only and uses voice allocation for utilizing the three oscillators. This way you can create polyphonic patches to play accords for example.

unision:

The VSTi instance listens to MIDI channel 1 only and triggers all oscillators at once. So you can create rich multi-layer mono patches using up to three layers (oscillators).

multi:

The VSTi instance listens to MIDI channels 1,2,3. This way you can control each oscillator directly from separate MIDI channels.

Arp mode:

Arp mode determines the arpeggiator mode of the current oscillator.

The arpeggiator can be only activated in multi/unision modes. Does not work in poly mode, since its purpose is to "emulate" accords on mono channels.

normal:

Notes are added to the arpeggio sequence when you start a note on the oscillator and are removed real-time when you release the note.

additive:

Notes are added to the arpeggio sequence when you start a note on the oscillator and are removed only when you've released all notes.

Arp div:

Arp div determines the arpeggiator speed of the current oscillator.

It is a divider value that divides the current BPM speed, so the arpeggiator speed always depends on the actual song tempo.

The smaller divider value means faster arpeggiator speed.

PB up, PB down:

PB up/down determines the pitch bender range (in up/down direction) in seminotes for the current oscillator.

Transp:

Transp determines the transpose value of the current oscillator in seminotes. Set to zero to disable this feature.

Fix pitch:

Transp determines the fixed note value of the current oscillator in seminotes. Set to zero to disable this feature.

Porta:

Selects the portamento mode for the current oscillator. The portamento can be only activated in multi/unision modes.

on:

Portamento is always enabled, so the oscillator's pitch will always slide from the previous pitch to the new pitch when starting a note.

manual:

Portamento is only activated when you start a note while still holding the previous one.

Legato:

The legato function allows you to create smooth note slides by starting a new note while still holding the previous one. The new note will start without re-triggering the envelopes (ADSR for example) for the current oscillator. Note: Legato is fixed to be 'on' for the current VSTi release even it is turned off.

Test snd:

The test sound can be used to test your ASIO settings. When you enable it ('on' or 'heavy'), it plays a beep sound at 8000Hz data rate on the SID's volume register. You should hear a solid beep sound. If it is full of gaps, your ASIO settings and/or your OS may be configured incorrectly. See the 'Optimizing your Windows for music production' appendix.

Other oscillator controllers

Detune knob:

Lets you to detune the oscillator's pitch. The minimum value means -1 seminote, the maximum value means +1 seminote..

Portspd knob:

Portamento speed for the current oscillator.

Filter toggle:

Enables filtering on the current oscillator. If you switch it off, the SID's filter will have no effect on the oscillator.

tri / saw / sqr / noise toggles:

Each oscillator has four selectable waveforms (triangle, saw tooth, square (pulse), noise). It is possible to select more than one waveform at once, but not all combinations produce sound. The SID 8580 supports more valid waveform combinations than the SID 6581.

Pulse width slider:

Sets the pulse width of the current oscillator's square/pulse waveform. This setting has no effect on other waveforms than square/pulse.

Sync toggle:

Synchronizes the fundamental frequency of the current oscillator with the fundamental frequency or the previous oscillator (osc1-osc3; osc2-osc1; osc3-osc2) producing "Hard Sync" effects. Varying the frequency of the current oscillator with respect to the previous oscillator produces a wide range of complex harmonic structures from the voice of the actual oscillator at the frequency of the previous oscillator. In order for sync to occur, the previous oscillator must be set to some frequency other than zero but preferably lower than the frequency of the actual oscillator. No other parameters of the previous oscillator have any effect on sync.

Rmod toggle:

The Ring Mod replaces the triangle waveform output of the actual oscillator with a "ring modulated" combination of the actual and previous oscillators (osc1-osc3; osc2-osc1; osc3-osc2). Varying the frequency of the actual oscillator with respect to the previous oscillator produces a wide range of non-harmonic overtone structures for creating bell or gong sounds and for special effects. In order for ring modulation to be audible the triangle waveform of the actual oscillator must be selected and the previous oscillator must be set to some frequency other than zero. No other parameters of the previous oscillator have any effect on ring modulation.

Test toggle:

Freezes the oscillator, so it does not generate any waveform as long as the test bit is enabled. It is frequently used to re-initialize the white noise waveform generator when producing drum sounds. It can also be used to mute an oscillator.

The HardSID 4U VSTi uses the Test toggle to perform an automatic "hard restart" by initializing/resetting the ADSR SID registers when the Test bit gets disabled after enabled state. It provides a "hard restart" functionality for table based sounds by simply handing the test bit in the table. The oscillator's hard restart function should be set to "off" when using the automatic table-based hard restart handling.

ADSR sliders:

The ADSR sliders are setting the Attack, Decay, Sustain, Release properties of the actual oscillator's volume envelope.

ADSR Ena toggle:

Enables/disables the triggering of an ADSR volume envelope when a new note starts playing on the actual oscillator.

It is recommended to disable for oscillators that are used only for ring modulating or synchronizing another oscillator.

Table based sounds are often handling the gate bit (volume envelope trigger bit) directly from the table. (drum sounds for example) In these cases the ADSR Ena toggle should be disabled.

Hard rest:

The SID's ADSR (volume envelope) is buggy and not stable. If you're designing a short and sharp drum sound for example, the volume envelope will not be the same stable envelope for every time. This is why

hard restart was developed in the old days on C64. It initializes and holds the ADSR envelope in a stable state until a new note is triggered. The numeric setting of the Hard rest (2x for example) means setting*125usec (microseconds) ADSR reset time. Hard restart causes ADSR latency.

You will not need hard restart for most of normal synthesizer sounds. It is typically needed for very short, sharply starting sounds, like drums. The actual setting is not pre-determinable. You have to work it out yourself for every given special sound (oscillator).

Note: Drum sounds and other short/special effects are usually designed using tables. In this case you should disable both the ADSR Ena toggle and hard restart and handle the gate bit / test bit directly from the table. It is also a good idea to enable the Test toggle of the actual oscillator permanently and modulating the test bit from the table with a value of -100 when you want to make audible sounds. (the HardSID 4U VSTi generates an automatic hard restart when the test bit is released)

Global SID controllers

Volume:

The Volume knob lets you to set the SID's main volume.

Filter:

Filter mode toggles:

The SID has one filter with three modes. All modes can be activated and used at the same time. Use the low/band/high toggles to enable/disable the low pass/band pass/high pass filter modes.

Cutoff knob:

Sets the cutoff frequency of the filter.

Reson knob:

Sets the resonance of the filter.

Filt curve selector:

Selects the filter curve to be used with the current SID/VSTi instance. The logarithmic mode is ideal for the 8580, the linear mode is ideal for the 6581.

External audio filtering toggle:

The SID has a mono audio input. This audio input can be found on the back of the HardSID 4U unit (two stereo jacks are used as four mono inputs for the four SIDs). You can filter this incoming mono sound if you wish.

Use the ext toggle to enable external audio filtering.

Parameter modulation

The HardSID 4U VSTi allows you to modulate various SID parameters (just like pitch, filter cutoff, etc...) with signals coming from various modulation sources (just like LFOs, Envelopes, CCs, etc...).

LFOs (LFO1-4)

LFOs are low frequency oscillators that can be used as modulation sources in the modulation matrix. (see "The modulation LCD screen" topic below)

Each oscillator has four LFOs, so there are 12 LFOs in the HardSID 4U VSTi for one SID.

LFO properties:

retrig:

If enabled, the LFO will be re-triggered every time a new note starts playing on the actual oscillator.

wavf:

The output signal's waveform:

sin (sine), tri (triangle), saw (saw tooth), sqr (square/pulse), rnd (random)

speed:

The LFO's speed.

pw:

Pulse width setting for the sqr (square/pulse) waveform. The middle setting means 50% low, 50% high state.

offs:

Waveform offset. The waveform will start with an offset when the LFO is (re)triggered.

Offset does not apply for the rnd (random) waveform, as it does not have a predefined waveform shape.

Envelopes (Env1-4)

Envelopes can be used as modulation sources in the modulation matrix. (see "The modulation LCD screen" topic below). Env1-4 must not be confused with the actual oscillator's ADSR volume envelope. Each oscillator has four Envelopes, so there are 12 Envelopes in the HardSID 4U VSTi for one SID.

Envelope properties:

hard retrig:

If enabled, the Envelope will be re-triggered from its initial state (zero value) every time a new note starts playing on the actual oscillator.

If disabled, the Envelope will be re-triggered smoothly from the actual state of the Envelope.

ADSR sliders:

Attack, Decay, Sustain, Release settings for the Envelope.

Env1-4 must not be confused with the actual oscillator's ADSR volume envelope.

Wave rate knob

The HardSID 4U VSTi can be used as an effect to other VSTs. It means that you can modulate any SID parameters with another VSTi's output audio signal. The most well known example of this is playing digital audio on the SID's volume register. (see the "Modulation sources" section below)

The wave rate knob sets the resolution of the incoming audio signal between 0Hz and 8000Hz.

The modulation LCD screen

The bigger LCD screen has three editor modes:

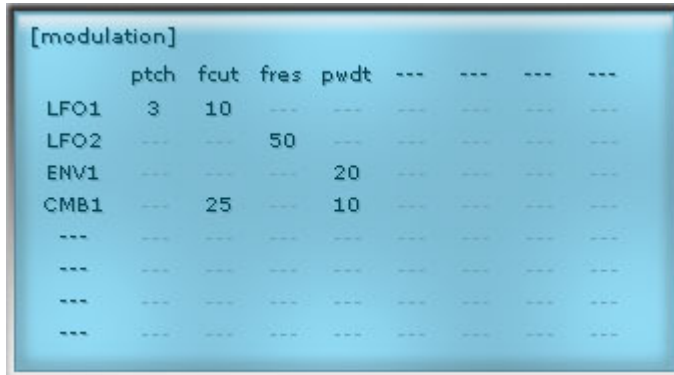
1. modulation
2. combiners
3. table

You can choose from the editor modes by clicking on the default [modulation] text on the upper-left corner of the LCD screen.

Modulation matrix editor:

There are three essential parts of the modulation matrix.

- Left column: Modulation sources
- Top row: Modulation targets
- The 8x8 matrix: Connects modulation sources to modulation targets (amount: 0-100%)



The screenshot shows a window titled "[modulation]" containing an 8x8 matrix. The columns are labeled 'ptch', 'fcut', 'fres', 'pwt', and four empty columns. The rows are labeled 'LFO1', 'LFO2', 'ENV1', 'CMB1', and four empty rows. The matrix contains numerical values representing modulation amounts.

	ptch	fcut	fres	pwt	---	---	---	---
LFO1	3	10	---	---	---	---	---	---
LFO2	---	---	50	---	---	---	---	---
ENV1	---	---	---	20	---	---	---	---
CMB1	---	25	---	10	---	---	---	---
---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---

Usage:

- Select the modulation sources in the left column (learn about modulation sources below)
- Select the modulation targets in the top row (learn about modulation targets below)
- Set the amount of modulation for each valid point in the modulation matrix
(click with your mouse to switch between 0 and 100% modulation; or drag the value with your mouse up-and-down for smooth settings between -100% and 100%)

Modulation sources:

Click on any row in the modulation source column (left column) and a popup menu will appear:

LFO1-4:

The modulation source will reflect the current value of LFO1-4. (bipolar)

ENV1-4:

The modulation source will reflect the current value of Envelope 1-4. (unipolar)

CMB1-8:

The modulation source will reflect the current value of Combiner 1-8. (bipolar)

See the Combiners section for further details.

note:

The modulation source will reflect the current note-on pitch MIDI value. (unipolar)

The simple definition would say: It is the number of the key you've pressed on your MIDI keyboard.

velo:

The modulation source will reflect the current note-on velocity MIDI value. (unipolar)

aftr:

The modulation source will reflect the current channel aftertouch MIDI value. (unipolar)

pbn:

The modulation source will reflect the current pitch bender MIDI value. (bipolar)

bpm:

The modulation source will reflect the current song BPM (queried from the VST host). (unipolar)

Example: BPM 140 = $0.14f$ = 14%; BPM 1000 = 1 = 100%

wave:

The modulation source will reflect the current value of the wave-in VSTi pin. (bipolar)

This modulation source will have value only if the HardSID 4U VSTi plugin is inserted into your project as an effect.

In this case the VST host will route the original sound data through the effect (HardSID 4U wave-in pin).

This can be used to modulate targets with digital sound data.

One well known usage is to play digitized sounds using the SID's volume register.

Example: Set the VSTi's main volume to 50%; modulate the main volume (mvol) with the wave-in pin (wave) by 50% (tweak the modulation matrix with your mouse). Now everything will play as digitized SID sound that comes in from the VSTi instrument you're listening to as an effect.

MIDI CC 0-127:

The modulation source will reflect the selected MIDI Controller's value. (unipolar)

Const:

The modulation source will reflect the selected constant value. (bipolar)

This is also useful when you simply want to divide something by two using combiners or simple things like that.

Modulation targets:

ptch:

Modulates the pitch of the current oscillator.

wtri:

Modulates the triangle waveform selector of the current oscillator.

wsaw:

Modulates the saw tooth waveform selector of the current oscillator.

wsqr:

Modulates the square waveform selector of the current oscillator.

wnoi:

Modulates the white noise waveform selector of the current oscillator.

pwdt:

Modulates the pulse width of the current oscillator (valid for the square waveform only).

fcut:

Modulates the filter cutoff value (there is one filter for one SID).

fres:

Modulates the filter resonance value (there is one filter for one SID).

fena:

Modulates the filter enable selector of the current oscillator.

fext:

Modulates the external filtering selector (there is one filter for one SID).

fmlp:

Modulates the low-pass filter mode selector (there is one filter for one SID).

fmbp:

Modulates the band-pass filter mode selector (there is one filter for one SID).

fmhp:

Modulates the high-pass filter mode selector (there is one filter for one SID).

gate:

Modulates the volume envelope's gate bit of the current oscillator.

sync:

Modulates the synchronization selector of the current oscillator.

rmod:

Modulates the ring modulation selector of the current oscillator.

test:

Modulates the test bit selector of the current oscillator. (freezes/mutes the oscillator's output when enabled)

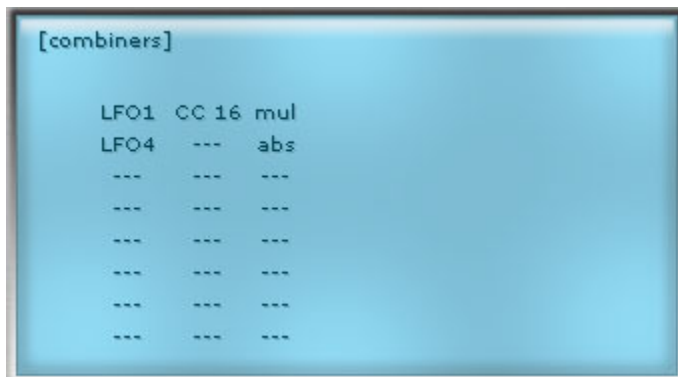
mvol:

Modulates the main volume (there is one main volume for one SID).
It can be used also for playing digitized sounds on the SID chip.

Combiners editor:

There are three columns in the combiners editor (from left to right):

1. Source 1
2. Source 2
3. Operation



Usage:

Select Source1 (and Source2 if you're planning to combine two sources). (learn about modulation sources at the Modulation matrix editor's section)

Select the operation you need to perform on Source1 (and Source2 if you've selected a source for Source2 too).

Now your combiner is ready to be used in the Modulation matrix editor as a modulation source.

Operations:

Two-source operations:

add:

The output will be $\text{Source1} + \text{Source2}$

Valid only when both Source1 and Source2 is assigned!

sub:

The output will be $\text{Source1} - \text{Source2}$

Valid only when both Source1 and Source2 is assigned!

mul:

The output will be $\text{Source1} * \text{Source2}$

Valid only when both Source1 and Source2 is assigned!

div:

The output will be $\text{Source1} / \text{Source2}$

Valid only when both Source1 and Source2 is assigned!

mix:

The output will be $\text{Source1} / 2 + \text{Source2} / 2$

Valid only when both Source1 and Source2 is assigned!

One source operations for Source1 only:

inv:

Inverts Source1 and assigns to the output.

abs:

The output will be the absolute value of Source1.

-abs:

The output will be the inverted absolute value of Source1.

upp:

The output will reflect only positive Source1 values. The negative values will be replaced by zero.

low:

The output will reflect only negative Source1 values. The positive values will be replaced by zero.

bip:

The output will be the bipolar conversion of the unipolar value of Source1. ($\text{Source1} * 2 - 1$)

unp:

The output will be the unipolar conversion of the bipolar value of Source1. ($\text{Source1} / 2 + 0.5$)

Table editor:

The table editor is also a modulation editor in general, but it is different, because:

- The modulation values are coming from values entered by the user row-by-row (value sequencing).
- When you assign a modulation value for oscillator pitch (ptch), it means MIDI note number modulation

instead of pitch modulation. For example +1 means one semitone higher, not +1% pitch.

	wnoi	wsqr	gate	test	pwdt	ptch	---	---
1.	---	---	100	---	---	---	---	---
2.	100	---	100	-100	---	21	---	---
3.	---	100	100	-100	---	2	---	---
4.	---	100	---	-100	---	1	---	---
5.	100	---	---	-100	---	21	---	---
6.	---	100	---	-100	---	---	---	---
7.	100	---	---	-100	---	21	---	---
8.	100	---	---	-100	---	21	---	---

(a typical snare sound table)

Basics

- Select modulation targets (the top row) (learn about modulation targets at the Modulation matrix editor's section)
- Enter the modulation values
- Set the value sequencing speed (little LCD knob in the top-right corner of the LCD screen)
- Set up looping if needed (around the middle-top of the LCD screen)

Hint: You can see on the illustration image (typical snare sound table) that both the gate and test bits are handled directly in the table. ADSR Ena should be disabled for this example to work (so, the drum sound won't get longer when you keep pressing the MIDI keyboard's key for a longer time). Since the test bit is also handled directly in the table, the Test toggle for the oscillator should be enabled for this example to work. The example table modulates the Test bit with negative values (-100%), so the permanently enabled Test bit will be released only for table rows with test: -100%. Since the HardSID 4U VSTi performs an automatic ADSR hard restart when the test bit is released, this method keeps the ADSR in a stable/frozen state as long as possible, then a hard restart is performed automatically when a new drum sound starts.

Inserting and removing a complete row

There are some situations when one needs to insert a new row at the top or into the middle of the table editor. ...or to delete a complete row from a table.

Press SHIFT and click on a row in order to duplicate it. The rest of the table will move one row down.

Press Ctrl (Control) and click on a row in order to delete it from the table. The rest of the table will move one row up. The very last row of the table will stay intact (row #32).

APPENDIX

Optimizing your Windows for music production

Professional musicians are building their PCs especially for music production. Even their motherboard is chosen carefully when building a PC for music. USB Audio + VST quality/stability/latency highly depends on the PC components. The next most important thing is the software environment. Professional musicians are not using the same OS install for everything. They have a clean Windows install dedicated especially for music production. These are only a few things mentioned here, but the ability of producing high quality music in a stable/low latency environment is a serious profession in itself that starts at choosing the right PC components and optimizing a dedicated Windows install for music production.

If you are not a pro and will not spend a lot of money on a new PC dedicated for music production, you should at least install a clean and fresh Windows instance that will be used only for music production and follow the next few advises.

A few hints to optimize your Windows XP for music production:

1. **Start with a clean install**

Some pros say that system restore, automatic updates, power management, themes, etc... should also be disabled. It is up to you how minimal system you can accept.

2. **Set processor scheduling to background processes**

Steinberg (the inventor of the VST and ASIO standards, creator of Cubase) says this:

http://knowledgebase.steinberg.net/91_1.html

PC users with Windows XP have to change an option in regard to the processor scheduling in order to work with low latency ASIO drivers without trouble.

- Go to Windows Start menu, select "Control Panels"

- There, start "System"

- In the window "System Properties" select the tab "Advanced"

- Click on "Settings" in the section "Performance"

- In the window "Performance Options" select the tab "Advanced"

- There you find the "Processor scheduling" options, set it to "Background services".

Your system is now ready to work with low latency ASIO drivers without drop outs.

3. **Disable your WiFi/Ethernet card in the device manager**

WiFi and Ethernet card drivers are well known sources of periodic timing problems on Windows. Poorly designed drivers (most of WiFi/Ethernet drivers) are periodically spending too much time in a privileged state called "interrupt level" when nothing can interrupt their thread. It means that your VST host / ASIO / USB Audio stream will be interrupted frequently. It can cause extra latency and lost USB packets (lost sound data).

4. **Use ASIO**

Steinberg invented a soundcard driver protocol for supporting low latency audio streams named ASIO (Audio Stream Input/Output). Professional soundcard makers are providing ASIO drivers with their soundcards. The quality of the ASIO driver highly determines the overall quality (CPU usage, latency, stability) of your virtual studio. If you have a professional soundcard with a professional ASIO driver, you should use it as the main timing source for your VST host. If you only have a SB Live, you can use the KX Driver project. If you don't even have a SB Live, you may try ASIO4ALL. It is better than nothing, but KX is better and a dedicated ASIO driver for a

professional soundcard is the best choice.

KX Project: <http://kxproject.lugosoft.com/>

ASIO4ALL: <http://www.asio4all.com/>

5. **Do not use this OS instance for anything else than music production**

Don't install anything else on it than the minimal required software for your music production process. You may never know which software registers a nasty service or alters some important Windows setting, etc... Be aware of anything else than well known software for your music production process.

If you are a professional VST musician, you already know about a lot more optimization steps than these few hints. However, we believe that these are the most important ones a newcomer musician should follow.