Johnny Woods
Johnny@johnnywoods.com

8,400 words

A Guide to the LZX Visionary

by Johnny Woods
(Incomplete and unillustrated)

Chapter One: Basic Principles

The Analog Video Signal

Voltage Principles

The fundamental concept behind all operations in an analog video system is simple: the signal is represented as a voltage, and the user creates images by modifying this voltage. Over the next few chapters we will look at various tools for manipulating this voltage, but first it’s important to understand how the signal works.

When one thinks of moving images, the usual representation is in “frames”. In both film and digital video, this is how images are stored: as a collection of discrete pictures, which when played back in succession, create movement. Analog video is a different animal, and the concept of frames needs to be rethought. Unlike film or digital video, we are never presented with an entire frame. Rather, a single beam draws to the screen very, very quickly. So quickly in fact, that we perceive it as a complete frame. However, this distinction is the defining characteristic of analog video. Instead of visualizing discrete frames, we must imagine a *continuous stream* of image information. This stream, of course, is represented by a voltage.

The easiest way to visualize this phenomenon is to think of the voltage stream as a piece of string. Beginning from the top left of the screen, the string travels from left to right, changing in color and intensity based on the values defined by the voltage. When it reaches the right edge, it is cut and returns to the left edge, one line down, and begins drawing again. It continues in this fashion until it reaches the bottom right hand corner of the screen, at which point it returns to the top-left, and begins its journey again. (Figure 1)[add image] This length of string (one trip from the top-left to bottom-right) represents one “frame” of an image. However, it is important to keep in mind that it is *not* representing an entire frame at one moment in time. This property defines a lot of what is and is not possible within the realm of analog image creation.

We will return many times to this image of the “voltage string”. For now, it is just important to understand two things:

first, that analog video images are represented over a continuous range of time, not at distinct intervals of frames per second. Second, that this constant stream of voltage is the final result of any analog video process.

Throughout your exploration of video synthesis, you will encounter many different types of voltages, and it can be easy to get confused at first. If this doesn't make sense at first, bear with us, and as we get deeper into the examples, things will begin to clear up. Like with any technology, your mastery of analog video synthesis benefits from a complete understanding of its underlying technical principles. At first, however, it can be more beneficial to simply play around and explore the effects of your system. While it is recommended to at least skim the rest of this chapter, the over-eager amongst you may feel free to jump ahead to the examples.

NTSC Signal

Interlacing

Understanding Sync

Composite Encoding

Control Voltage

Introduction to CV

In the last chapter, we looked at the way analog video is represented using voltages. That discussion was referring principally to the final output of any video device (that is, the idea applies equally to video cameras, playback devices, and your LZX Visionary). When using an analog video synthesizer, we employ voltages *within* patches to control all operations of the system. This can be confusing at first, but actually makes operation very simple to understand. The image of the voltage string still applies, and we can still imagine our voltages traveling down and across each frame of the image. Instead of just imagining one string, however, we will have dozens of voltage signals, which when combined, generate the final image.

Voltages can be defined by their amplitude, frequency, and waveform. This is the key concept to understand when creating even the simplest of effects. The amplitude is represented in volts (v), ranging from 0-1. If we were to create a steady voltage of 0v across each of the RGB channels, we would see a completely black screen, whereas a value of 1v would produce a completely white screen. In between, we would see different values of gray. If a value of 1v was applied to just the Red input on the Color Video Encoder, with 0v applied to the Green and Blue, we would see a pure red screen.

Of course, using static voltages would not result in very interesting images! Usually, we will want to modulate these voltages over a period of time. This period is the frequency of the voltage, and it can range from very slow (many seconds, even minutes) to very fast (each line of the analog raster). We measure the frequency in Hz, using kHz (1000 Hz) or mHz (1,000,000 Hz) as necessary. This value tells us how many times a voltage changes per second. Imagine we have a voltage with a frequency of 1 Hz, and an amplitude range of 0-1v. This would be displayed as an image which changes from black to white every second. If we take the same voltage and increase its frequency, we will get a faster rate of change from white to black. As we get into even higher ranges, the change from black to white will begin to happen many times *per frame.* This results in patterns within each frame of the image, a phenomenon which will be discussed at great length in the next chapter.

The way a voltage gets from one amplitude value to the next is called its waveform. This will again be described in more detail later, but to build off the previous examples, consider this: if a voltage with a frequency of 1 Hz, and an amplitude of 0-1v is measured halfway through its cycle, what will its amplitude be? If the waveform is linear (which is called a sawtooth wave), the value will be 0.5v. However, we can also use a square wave, which would produce a value of 0v (figure) [sawtooth, square comp]. So, in summary,we can think of voltages as a measure of change defined by its amplitude (how much change?), its frequency (how quickly does it change?), and its waveform (how does it change over time?).

Thus far, we have been using the example of a voltage producing an image or color on a screen. However, this is only half the story. While it is certainly essential to understand how images are made up of varying voltage signals, it is equally important to see how we can use voltage signals to control operations within our system. These are known, simply enough, as *control voltages*, or CV signals. If were to look at these voltages, they are identical in form to image-creating voltages, and can be of the same amplitude, frequency, and waveform. In fact, **there is no actual difference between the two types of signals other than their application.** Imagine again an image represented by a sawtooth voltage 0-1v at 1hZ. We would see the screen progress from black to white once per second. Now, imagine that another voltage (CV) was being used to adjust (or modulate) the frequency of the first voltage. Let’s say this CV is also a sawtooth 0-1v, at a frequency of 0.1hZ. The effect would be a a screen changing from back to white quite slowly at first, and increasing in rate over the course of 10 seconds, causing the image to flash or flicker with increased speed. Just like voltages used for image generation, these CVs can be of frequencies so fast that they cycle many times *per frame*. To extend our above example, we could use a CV to produce different rates of flashing within each frame. We can even use external video (say, from a camera) as a CV source to create very complex effects.

Each of these voltages are generated, modified, or combined by specific modules within the Visionary system. By design, the Visionary does not distinguish between different types of voltages, and all modules are capable of working with voltages of any frequency. This opens up a lot of flexibility within the system, and allows for open exploration of techniques. Nonetheless, it is useful to classify frequencies for the sake of practical explanation.

(INSERT TABLE OF FREQUENCY RANGES)[frequency table]

(INSERT EXPLANATIONS FROM LZX WEBSITE)[Johnny Woods, 2/25/12 5:24 PM]

Voltage Specifications

A few other notes on voltage specifications before we proceed. This section will be of particular interest to those working with other voltage-controlled equipment, such as an audio synthesizer. The LZX Visionary utilizes a system-wide, standardized voltage specification. This allows maximum flexibility since all signals, no matter what their intended use, can be used for any application. Some other analog systems make a distinction between different voltage signals (for example, image signals may have a different specification from those used for CV). If your system solely contains LZX modules, you can freely patch any output into any input with predictable results. However, if integrating other equipment, there are a few things you should be aware of.

Many LZX signals are *unipolar*, meaning their values do not go below 0. Many analog devices intended for audio use are capable of outputting *bipolar* voltages. This results in a voltage that swings from negative to positive voltages, say −1v to 1v. Many LZX modules will happily accept these bipolar voltages, and they can even be used for great effect. However, in other cases they can create unpredictable results, so it's important to understand which types of signals you are working with. In most cases, audio-rate oscillators, LFOs, and analog audio signals (eg: from a pre-recorded source) output bipolar signals, while envelopes, trigger/clock sources, and sequencers output unipolar signals.

There is also the question of maximum amplitude. Again, LZX modules are generally expecting 0-1v on their inputs. If you are working with other Eurorack audio modules, they are typically in a far greater range (commonly, −5v to 5v). Some voltage controlled equipment will go into even greater ranges. You should be able to check manufacturer's specifications on a specific device to find this information. Inputting a voltage over 1v (or below 0v) will not harm your LZX modules, but you will sometimes get undesirable results. The voltages above 1v or below 0v can be simply clipped off, leaving you with a very narrow range of control. To compensate, you need to *attenuate* or scale the signal into an appropriate range. This will be discussed in the next section.

Equally important, if you are attempting to take a Visionary output to the input of a device expecting a greater range of voltage, you will often have to amplify the signal. This is particularly important when trying to use your Visionary voltages to trigger events or open VCAs.

With that being said, do not be afraid to experiment! Just because your LZX system is not expecting a certain type of voltage, it might still produce an interesting result.

Attenuation, Bias, and VCAs

When applying a control voltage to any function of your synthesizer, it is important to have the ability to adjust how much of an effect the voltage should have. In other words, thinking back to our previous example, if we are using a CV to change the frequency of a flashing screen, we might want to increase that frequency *just a bit*. Instead of applying a CV that goes from 0-1v, we may want a CV to scale from 0-.1v. In most modular systems, this modification is performed on the input of the module receiving the CV by way of an *attenuator*. Attenuation is simply a fancy word for “turning something down”.

So, if we attenuate an incoming CV, we are basically just "turning down" its effect. Of course, this applies to image voltages as well; attenuating them will make them darker. Most modules in your LZX Visionary will have attenuators connected to their inputs.

As noted earlier, your Visionary system is often expecting 0-1v signals. So what if, in our earlier example, you wanted to slow down the frequency using a CV instead of speeding it up? Since the Visionary usually outputs 0-1v on its image generators, many CV inputs that can make good use of negative values feature *attenuverters*. In addition to attenuating the incoming voltage, they can also invert it (subtract it from 0) in order to produce an opposite effect. So, if we dialed our attenuator into the negative range, we could obtain a voltage of −0.1 – 0v. This is especially useful for color blending and frequency modulation effects.

Another effect you may need is the ability to add a *bias* to the CV. Say we have attenuated a CV so it goes from 0-0.1v, but what we need is a range of 0.3-0.4v. We can simply add a bias of 0.3 to the signal, and voila! We can also add a negative bias in many cases.

As you can see, using these 2 simple operations, we have the ability to precisely define the range of effect any CV will produce. Sometimes, you may want to subsequently use another voltage to CV the amount of attenuation. This requires the use of a voltage controlled amplifier, or VCA. Many have noted the rather confusing nomenclature of this function, since it typically does not amplify the signal, but rather, attenuates it. Your synthesizer uses many VCAs within its internal circuitry, and of course, has modules which offer direct access to this functionality, which will be discussed in a further chapter.

VI1 Diagram

Images of VI1, Voltage Bridge from LZX website

Voltage Interfaces

If you are working with other modular equipment, you will most likely want a Voltage Interface I (VI1) or Voltage Bridge (VB) in your system. VI1 is a basic implementation of attenuation and bias controls, with LEDs to provide visual feedback of your voltage range. Each channel can accept an incoming voltage (say, from an audio oscillator, lfo, etc…), and will *clip* voltages over 1v or below 0v. When a signal is clipped, it simply means it will not pass a certain value (figure[clipping diagram]). So, if we had a sawtooth wave with a maximum amplitude of 2v, when the amplitude reaches 1v, it will simply stay there, until the voltage drops back below the threshold. The top LED will flash red when the signal is being clipped at 1v, while the bottom LED will flash when the signal is being clipped at 0v (to remove negative voltages). While clipping can be used for expressive purposes, the module is mostly designed to produce clean voltage signals from external equipment. For cleanest results, the attenuation and bias controls should be adjusted until neither LED is lit. See the figure below for an appropriate use of this technique (figure) [LFO -> VI diagram].

Voltage Bridge provides a simpler, “preset” version of the VI1, plus the ability to amplify 0-1v signals for use in external equipment. There are no attenuation or bias controls, just inputs and outputs. The top four jacks will attenuate a signal by 20% (in other words, from 5v to 1v), while the bottom four will amplify a signal 500% (1v to 5v). This is a quick and easy way to interface signals between different systems, and although it lacks the control of the VI1, in many cases, it will be more than sufficient.

LZX Specific Information

(INFO FROM WEBSITE, INTERFACE STANDARDS)[http://www.lzxindustries.net/interface

]Basic Signal Flow

So now that we understand the basics of how CVs work, and how we can modify them, we can begin to talk about their actual use. *Signal flow* refers to the various connections between functions within your synthesizer. The great advantage of a modular system is that we are presented with a completely open signal flow.The user is free to make any connection she can imagine. While this presents us with an almost infinite number of options, it also requires a higher degree of understanding signal flow on the part of the operator. Although it can be daunting and confusing at first, with continued use, the process of defining a signal flow becomes the main nexus of expression for the synthesist. The particular signal flow is described as a *patch*, a term coined quite literally in a modular system since it is defined by the connections patched by cables between the modules. Within a certain patch, a great range of images are possible. An entire performance can, and often is, the result of just one patch. In many ways, the patch creates the instrument, and the possible outcomes are limited only by the complexity of the patch and the skill of the performer. We will examine many, many patches in this text, accompanied with signal flow diagrams to illustrate the exact connections.

Although, as previously discussed, the LZX Visionary makes no distinctions between CVs and voltages used primarily for image generation, it will be useful for us to define them separately at first. This distinction will become less relevant as we get into deeper examples, and once we have an understanding of basic module functions. For now, we will refer to them as IVs (image voltages, or sometimes just images) and CVs (also known as *modulators* or *modulation sources*). There is one other specific use for voltages, which is to carry sync signals throughout the system. These will be examined later.

For now, let's look at the example we have been using throughout this chapter as a theoretical exercise (without using any specific module information). To review the example: an IV with an amplitude of 0-1v at a frequency of 1hZ is created in order to make the screen flash from black to white once per second. In order to make this effect more interesting, we are applying a CV to the frequency of the first voltage to change the speed of the flashing. This effect would require only 2 patch connections: one from the CV signal to modulate the IV's frequency, and one from the IV to the video encoder (for output to the screen). This would graphically be represented as follows (figure[patch diagram: basic FM]).

And that’s about as simple as signal flow gets! Of course, this patch would give the performer a very limited palette of images, but we have to start somewhere. Speaking of which, it’s now time to examine our first real patch using the core modules of any LZX system, the Video Sync Generator and Color Video Encoder.

A QUICK NOTE ON INPUTS< OUTPUTS AND NORMALLED CONNECTIONS:

All jacks on the LZX Visionary are designed for 3.5 mm cables, and are buffered on the outputs. This means they are able to drive multiple inputs from one output. This effect is achieved using either a “multiple” module (such as the Doepfer A-180) or TipTop Audio’s Stackcables. It is also very simple to build your own multiple panels (see sidebar[HOW TO BUILD A MULT PANEL]). In any case, multiples should not be seen as optional in a video modular system. Many examples will make heavy use of this technique.

To make your life a bit easier, most LZX modules feature a built in multiple function, called *normalling*. If jacks are normalled, it means that the input signal to one is automatically multiplied to the other inputs. This is graphically represented by small arrows on the right hand side of the jack. If the normalling function is not desired, you can break the connection by inserting another signal, or plugging in a "dummy jack" (a cable with nothing plugged in on the other end!).

Visionary modules distinguish between input and output jacks by placing a black background around output jacks. Input jacks have a "clear" background. Nothing bad will happen to your equipment if you accidentally plug an output into an output, or vice versa. Of course, nothing good will happen either.

CVE/VSG diagrams

Images of CVE/VSG, from LZX website.

Inputting Video

One of the first things you may want to explore with your LZX Visionary is processing an external video source (such as a video camera, VHS tape, laser disc, or DVD). Every Visionary system must contain one Video Sync Generator (VSG). As mentioned previously, this module handles the creation and distribution of system-wide sync signals. As an added bonus, it is also capable of converting a common composite video signal into the LZX voltage standard. Simply plug the video output from your external device into the “Video In” jack, and switch the “Ext Sync” switch on. Flipping that switch tells the VSG to lock to the sync signal coming from the external source, instead of generating an internal sync. This ensures that the incoming image will display properly. If your image is scrolling across the screen, you will need to flip the switch to the other position.

Below the input jack, there is a Y out. This patch point allows you to take the video input image to any other point in the system. For now, just patch it into the R input on your Color Video Encoder, adjust the knobs so they match the diagram below (figure)[CVE “neutral” position], and make sure the composite or S-Video output from the CVE is connected to your screen. You should see a black and white version of the external source. The VSG is only able to input the black and white (or Y) portion of the external video (sidebar[LZX and Color Subcarrier explanation]), and in this case we are making use of the normalled connections on the CVE to distribute this evenly to all three color channels. Begin to turn the various knobs on the CVE, and you will see the image shift through different color ranges. Pretty cool, right?

So, what exactly is going on in this patch? It seems as simple as plugging your VCR into your television, but in fact, something quite different is taking place. The external video image is actually working as a CV! It is controlling the amplitude of the red, green, and blue channels of the image. When you adjust the attenuverters and bias controls on the CVE, you are scaling the modulation voltage, which in turn affects the amplitude of each particular color channel. Hopefully, this example gives you some insight on how CVs work within the LZX to define the output image. If you’re still a little lost, don’t worry. Theory is good, but once we start patching, things should become much clearer.

Oscillation

Sync, Range, and Oscillators

If you are not processing external video, most of your patches will begin with one or more oscillators. A video oscillator generates voltages which alternate (or oscillate!) between 0-1v at a certain frequency and with a specific waveform. At lower frequencies, oscillators make great modulation sources. Once you get into higher speeds, they are essential building blocks for any shape or pattern. The LZX Visionary currently boasts two models of oscillator, Video Ramps, and the Video Waveform Generator. Before looking at these in detail, lets examine two key attributes of any oscillator's operation: frequency range and synchronization.

As mentioned previously, video synthesizers employ a wide range of frequencies to produce different effects. The table below identifies how these frequency ranges relate to oscillators: (figure)

For all intents and purposes, we can break these into 3 broad categories: animation rate, horizontal rate, and vertical rate. As you can also see, the table indicates 3 different sync options: unsynced, field sync, and line sync. These concepts were touched on in a previous section, but how do they specifically relate to oscillators?

Let's think again about our voltage string. The final output from your Visionary is controlled by a master sync signal. Basically, this tells the system when it needs to "cut the string" to create a new line of the image. Now imagine that you have an oscillator of a fairly high frequency, so that it creates several cycles per frame of video. Imagine this voltage as its own string. Essentially, syncing the oscillator ensures that its string begins and is cut at the same place as the master output. This results in stable images. The two available sync signals for oscillators are *field* and *line*. If an oscillator is synced to field, it will reset itself every time it reaches the bottom right hand corner of the screen. This results in stable horizontal bars. If an oscillator is synced to line, it will reset itself every time it reaches the end of a line of video. (Sidebar)[audio oscillators and the LZX

] This results in stable vertical bars.

Unsynced oscillators are useful for creating movement within a frame. Animation rate signals will rarely be synced (since it usually has the effect of blanking the signal), and unsynced signals in the horizontal rate range are a very common source of modulation in patches.

For now, though, let's look at a very simple implementation of video oscillation. The humble, but mighty, Video Ramps.

Video Ramps diagram

Video Ramps Diagram and Controls

Video Ramps Patches

Video Ramps (or just Ramps, as we will affectionately call it) is easily the simplest of all LZX modules. However, most users swear by them as a must-have component of a system, and I strongly suggest all new users to have one in their setup. The basic concept is simple: a pair of fixed-frequency oscillators, each with 4 outputs: triangle, sawtooth, inverted triangle, and inverted sawtooth. Each of the oscillators is locked to the exact frequency which produces one cycle per frame, and synced to the master. One locks to horizontal, and one to vertical. There are no controls, and no inputs.

Once we begin working with comparators, the true greatness of Ramps will become apparent. Until then, it will serve to illustrate basic principles of oscillators. Connect the topmost output on your Ramps module to the R input on the CVE. (Figure [Ramps -> CVE]) You should see a smooth gradation from black to white running vertically along your screen. This is what we would refer to as a horizontal rate oscillator, synced to field rate. If the oscillator was not synced, the beginning of the ramp would not consistently start at the top of the screen. It would scroll up or down, simply because the signal does not know where the frame resets, and it will do so whenever it completes a cycle.

So now, switch the output of the Ramps to the fifth jack down. You will now see a similar picture, but with the gradation running horizontally across the screen. This is what we refer to as a vertical rate oscillator, synced to line rate. If this oscillator were not synced, we would see chaos. Vertical rate oscillators need to be reset at the end of *every line* of video. Horizontal rate oscillators just need to be reset so they know where the top of the frame is, but vertical rate requires a signal for each line (14,400 times per second!), so it knows where the right edge of the frame is, and can reset accordingly.

Both of these outputs represent sawtooth waveforms, which linearly progress from 0-1v. Ramps can also output a triangle waveform, which moves linearly from 0v, to 1v, and then back to 0v over one cycle. Inversions of each of these waveforms are also available. Try patching each output to the CVE to get familiar with the waveforms. You can also combine them by plugging into different inputs on your CVE. Try and make a beautiful sunset.

Okay, not terribly exciting, I admit. Let’s take a look at our full featured oscillator, and have some real fun.

VWG diagram

VWG DIAGRAM AND CONTROLS

Frequency and Sync Relationships

The Video Waveform Generator is a powerful, full-featured oscillator, and the core component of most complex shape or pattern synthesis patches. It is capable of a wide frequency range with 5 simultaneous waveform outputs. All voltages on these outputs are 0-1v. There are two inputs for frequency modulation, and two for specific waveform modulations. These will be discussed later in this chapter. At the top of the module, there is an input for sync. As previously mentioned, to achieve a stable image from the oscillator, the appropriate sync must be applied here. To make patching multiple VWGs easier, the Visionary system uses a distributed sync signal which travels along the same bussboard that powers your modules. This allows access to two global sync signals without the need for any patch cables. The switch next to the sync input will connect one of these global sync signals to the VWG sync input. By default, the field sync is connected to Bus 1, and the line sync to Bus 2. The VSG allows you to patch other signals to the sync busses, but for now, we will assume the default configuration.

Let’s begin exploring sync and frequency relationships using your VWG. To begin, patch the topmost output (the triangle wave) from your VWG into the R input on the CVE (figure[VWG tri -> R CVE]). Make sure the sync selection switch is set to the middle position, and adjust the frequency range control on the VWG to its lowest setting (fully counter clockwise). Play around with the large frequency knob at the top of the module. You will notice the entire screen flashing from black to white, with the rate of flashing increasing as you turn the knob up. Now try switching the frequency range knob up one click. You will now have access to a faster rate of flashing, and at the highest speeds, you should begin to see horizontal bars. When the oscillator is functioning in these “flashing” ranges, we refer to its operation as an *LFO*, or low frequency oscillator. You can certainly use an oscillator in this range to create a strobing image effect, but a more common application is to modulate some other parameter within the system. This is why we refer to this as the “animation rate”, because a typical use case is to create motion on another parameter.

Traveling further up the frequency range knob, we reach the “horizontal rate”. At these frequencies, we are able to create horizontal bars. Adjust the sync switch so it is set to Bus 1 (by default, *field* rate from the VSG). As you adjust the large frequency knob, you will notice more and more horizontal bars filling your screen. As we increase the frequency, we are increasing the number of times the waveform repeats within one frame of video. You will notice a soft edge on the bars. This is a visual manifestation of the signal waveform, in this case, a triangle wave. Over the course of one cycle, a triangle wave will linearly increase from 0v to 1v and back to 0v. Try switching to the different waveform outputs, and notice the resulting shapes. Take special notice of the last two outputs (pulse and sine) as they have adjustable waveform controls. The knobs directly to the left of the outputs will modify the width of the waveform. These provide a very powerful source of modulation, which we will discuss later. Try adjusting the frequency of your VWG so that you have exactly one cycle of the waveform on your screen. Notice that the output of the triangle and sawtooth waves will look identical to certain outputs on the Ramps module. That’s because Ramps is exactly that: an oscillator fixed at this frequency, with triangle and sawtooth outputs.

Return for now to the triangle output, and adjust the frequency knob so you have at least 5 or 6 horizontal bars on your screen. Now, switch the sync to the middle position. This will remove the system sync from the oscillator causing it to run freely. The bars will begin moving vertically across the screen. Using the course and fine frequency adjustments, try to get the bars to come to a complete stop. What’s going on here? Imagine again the voltage string. Our entire Visionary system is running on a master sync, which essentially tells the system where to reset the top of the frame. In our previous example using a synced VWG, this information was communicated to the oscillator, causing the “string” to be cut in the same location for each frame. This results in stable images, since there is no change from frame to frame. Once we remove the sync signal, the oscillator is free to cut its own string independent from the rest of the system. Thus, every frame produces slightly different image information, which we perceive as movement. When you tune the VWG to a frequency close to an exact multiple of the frame rate (29.97 fps NTSC), we get a more stable (slowly moving) image. This technique is a fundamental source of animation for video synthesis, and an important effect to understand. Take a few moments to become comfortable with this function. We will refer to this as a *scrolling* horizontal bar.

The last two positions on the frequency range knob will allow us to produce stable vertical bars when we select the Bus 2 sync (by default, *line* sync from the VSG). Adjust the settings on your VWG so that you see several vertical bars on your screen. You will probably be tempted now to unsync the oscillator, and may be expecting a similar result: scrolling vertical bars. Unfortunately, this will not be the case. Instead, you will see a noisy chaotic texture. At these frequencies, things are happening way too fast (almost 500 times per frame!) to create a stable perceivable image. (See sidebar [oscillators and sync] for a more complete description of oscillators and sync relationships)

So now, you understand the basic functions of the VWG. In all of our patch examples, we will refer to our oscillator settings as LFO, stable horizontal, scrolling horizontal, stable vertical, and unsynced vertical. Make sure you are familiar with each of these settings. We will later explore the use of sync signals other than the default *line* and *field*.

Frequency Modulation

As exciting as straight bars can be, sometimes you may want to create more complex shapes. The VWG has four modulation inputs to allow for this. The first two are frequency modulation, or *FM*, inputs. Basically, these inputs will allow you to use a CV to adjust the frequency of the oscillator. Like all LZX inputs, these can accept a voltage in any frequency range. Each input has an attenuverter, which allows you to scale the incoming voltage to produce the desired effect. At the center position, the incoming voltage will be ignored. Turning the knob clockwise will produce a greater positive effect, while turning it counterclockwise will produce a negative effect. When applying FM CV, the modulation is limited to the currently selected frequency range. This is important to note because in some cases the FM input will produce no effect. For example, if you have a VWG with its frequency knob turned all the way up, and apply a positive CV to the FM input, you will see no change. Even if you are in a lower frequency range, the VWG can not jump across ranges as a result of FM operations. This can become a source of frustration when first learning FM techniques, so to be safe, try to keep your frequency knob somewhere in the middle for these exercises.

We will assume for these examples that you have 2 VWGs. Traditionally, when speaking of FM techniques, the oscillator producing the final output is called the *carrier*, and the one providing the FM modulation, the *modulator*. If you only have one VWG, you can use another voltage source as the modulator for these examples (Ramps, any non-LZX oscillator or lfo should work for most cases). Begin by setting the carrier to an LFO setting, and patch the triangle output into your CVE. Set the modulator to a very slow LFO. The LED on the module will give you an indication of the LFO rate. Patch the modulator's triangle output into the FM1 input on the carrier, and play with the attenuverter on the input (figure[basic FM]). This is the same example we discussed theoretically in the previous chapter. You should now see a flashing screen that changes in speed based on the frequency of the modulator. Experiment with different ranges and frequencies, and you will begin to see the power of FM synthesis.

Let's look at a few specific use cases. Set the first VWG to a frequency in the horizontal range, select sync bus 1, and connect the square wave output to your CVE. You should see a pattern of black and white bars. Now, connect a second, un-synced (middle-position switch), LFO-rate VWG's triangle output to the FM1 input on the carrier. The bars should be getting fatter and thinner based on the speed of the modulator. Dialing different values in on the FM1 attenuator will create more subtle or drastic effects. Now, adjust the range switch on the modulator VWG to get into horizontal rates, and notice the difference. Your carrier oscillator is now being modulated by a separate set of horizontal bars, and if you get the settings correct, you should see a pleasant effect of undulating stripes. Adjusting the range even higher on the modulator VWG will produce a surprising result: nothing at all. The modulating oscillator must always be of a lower range than the carrier oscillator if we are to see any effect. A vertical range frequency can not effectively modulate a horizontal range frequency. A horizontal range frequency can not effectively modulate an LFO rate frequency. (Sidebar[FM and frequency])

However, the opposite can be a very effective, and often-employed technique. Switch your carrier VWG up to vertical range (sync to bus 2), set your modulator VWG to horizontal rate (sync to bus 1). Adjust your frequency ranges and FM amount until you see a design reminiscent of Charlie Brown's shirt. Switch through the different outputs on the modulator, and become familiar with the different effects. These are an important component of pattern synthesis, and form the basis for some of the most expressive and interesting patches.

Spend some time experimenting with different FM combinations. Try using different waveforms from the modulator into both of the FM inputs on the carrier. If you have a video ramps module, you can use that to modulate the modulating VWG! You will quickly find yourself making mesmerizing op-art designs. Don't forget to un-sync your horizontal rate oscillators for hypnotic effects. (Image examples)[examples of FM op-art]

For even more fun, use an external video source (such as a camera feed) as an FM source for a vertical rate oscillator.

Waveform Modulation

The LZX VWG modules have another important modulation feature, which functions in much the same way as FM. Pulse and sine outputs are capable of waveform modulation, which allows an external signal to adjust the shape of their wave at a specific frequency. These are referred to as pulse-width-modulation (*pwm*) and sine-shaping, respectively. Repeat the first patch above, this time using the pulse output and plugging the modulator into the pulse width input instead of FM1 (LFO rate, unsynced, triangle out modulator -> horizontal rate, bus 1 synced, pulse out carrier). Adjust the two pulse width control knobs and notice the different effects. Essentially, you are increasing or decreasing the "high" portion of the wave. Since a pulse wave simply switches from 0-1v (with no interpolation) you will see your bars getting thicker and thinner. However, as opposed to the FM version of this effect, the center of the bars here remains the same (instead of progressively moving further down the screen) and the the effect is uniform across all the bars.

Sine shaping works in the same fashion. Switch your carrier VWG connections down one jack each, so the modulation input is connected to "sine shape", and your output to sine. If you maintain similar settings, you should be seeing a very similar result, with the only difference being the soft edges of the sine wave.

There is one key operational difference between waveform modulation and FM: it is possible (and quite useful) to have a higher-rate oscillator modulate a lower-rate one. Switch your modulating VWG to vertical range, synced to bus 2. Adjust your frequencies until you see a field of diamonds. Interestingly, when dealing with waveform modulation, the carrier-modulator relationship is interchangeable (definitely not the case with FM!). With your field of diamonds still on screen, adjust the sine shape knobs on your modulator to match the carrier. Now, switch all outputs, so the two oscillators effectively switch places *without adjusting the frequency settings*. With a little fine tuning of the sine shape controls, you should see the exact same result. This is an important feature of waveform modulation, and opens up a lot of patching possibilities. The patch below represents a very complex example of cross-modulation. This uses all 4 waveform modulation inputs between your two VWGs, and allows for some very interesting patterns. Experiment with different settings, and try to create some of the patterns below. (Figure[4-way waveform modulation + example stills])

As with FM, it can be fun to use external video as a modulation source. Trying using a camera feed as a sine shape modulator for a high-frequency vertical rate oscillator and notice how you can “re-synthesize” the external image!

Operations Within the LZX

Addition and multiplication are very simple concepts to understand within a modular system. Remember that all images are internally represented as a voltage between 0-1v. If you add or multiply two signals, you will get an image that represents the sum or product of their voltages at a specific moment in time. If you are familiar with image processing software such as Photoshop or After Effects, these effects will be identical to using the "add" or "multiply" blending modes (as long as the two sources in question are horizontal rate or vertical rate. The result for LFO-rate signals is quite different). (Figure[add and multiply blending examples])

Most of your LZX modules employ addition and multiplication processes somewhere in their internal architecture. For example, when you use the attenuverter on your VWG FM inputs, you are effectively multiplying the incoming signal by a value between 0-1v. This is often referred to as adjusting the "gain" of the incoming signal. In fact, the terms gain and multiplication can be, and often are, used interchangeably. "Bias" is a similar term which refers to adding (or subtracting) a static value from an incoming signal.

There are several modules specifically designed for adding and multiplying multiple input signals. The Triple Video Processor (TVP) and Triple Video Fader and Key Generator (TVFKG) are two such modules we will examine in this section. (Note: the Video Blending Matrix can replicate many functions of the TVP, and is more powerful in many ways. It will be discussed more in a future chapter. Many of the TVP examples below can be patched with a VBM as well.)

TVP diagram

Summing Signals

The TVP is a simple utility module for summing 2 or more signals. It has three identical channels, each with 2 inputs (A and B) and 1 output. The voltage running to input A has two controls for modification: a bias and gain knob. The bias control simply adds a static value between -1v and 1v to the incoming signal. This is further adjusted by the gain control, which takes the sum of the input and bias and multiplies it by a voltage from -1.5v to 1.5v. The result of these two operations is then added to the signal in input B, and output to the sum jack. It may sound confusing, but in practice, it is a very simple module to understand.

For this example, use the top-most (sawtooth) output on a Ramps module and connect it to the A1 input on your TVP (you can also use a sawtooth output on the VWG, synced to bus 1, and tuned so only 1 wave repetition is seen on screen). Connect the Sum1 output to your CVE, and assure that all controls on the TVP are in the center position. If you did everything correctly, you should see nothing on your screen. Adjust the knob labeled 1A at the top. This is modifying the gain of the signal, effectively multiplying the ramp input by a fixed voltage. At around 3 o'clock, you should see the ramp evenly gradate from black to white across the screen. Keep the knob there for the moment, and begin adjusting the control to its left. Notice how the position of the ramp appears to change. We are now adding a bias to the incoming signal, adding (or subtracting) a fixed voltage to the original signal. Hence, darker parts of the signal get brighter (or vice versa), creating the illusion that the ramp is shifting along the vertical axis. If you turn the bias control all the way up, it adds a 1v signal across the input, and your whole screen goes white. Turn it all the way down, and it subtracts a 1v signal, causing the whole screen to go black.

Now, connect the vertical sawtooth ramp (5th jack down) to the B1 input on the TVP. You should now see a diagonal gradient fading from black in the top left to white in the lower right. Adjust the bias controls, and you will see a similar animation effect as before. With both inputs connected, the TVP acts as a summing mixer, that is, it combines the two signals by adding them together. The A input has a static bias applied (via the first knob) and a static gain adjustment (via the second knob) before being summed with the unaffected B input. Use different ramp shapes and experiment with different bias settings on the A input. Keep the gain at 3 o'clock for now so you can more clearly see the summing effect. If we were to express this mathematically:

[(A input + A Bias)*A gain] + B input = sum output

Don't let the math scare you! It's an important concept to understand, especially when trying to troubleshoot unexpected outputs or create specific designs and effects. The best way to get a feel for it is to experiment, and notice the various outputs. Try and predict what a certain setting will do before patching and adjusting settings. Once you are able to approximately predict the behavior of your synthesizer, it's use becomes much more enjoyable and expressive.

Keeping all three gain controls at 3 o'clock, patch the sum1 output into the B2 input, and the sum2 output into the B3 input. By connecting different ramps into A1, B1, A2, and A3, and adjusting bias controls, you should be able to make all of the following shapes. (Figure[ramps TVP patch: complex shapes]) Again, try and predict which ramps will go where before patching.

Now that you have a basic grasp on summing principles, let’s look at an example of image processing. Attach an external video source to your VSG, and place the Y output into the B1 input on a TVP. Ensure that nothing is plugged into A1 and view the output from Sum1. Again, keep your gain at 3 o’clock, and adjust the bias control, uniformly adding or subtracting a fixed voltage from the entire image. The image will get brighter and darker across all values. When summing a positive bias, the darkest parts of your image become lighter, and vice versa, which may or may not be an appropriate effect. Now, patch in a ramp or VWG waveform to the A1 input and continue adjusting the bias control. You are now adding three signals together (the video source, bias, and oscillator waveform). An important fact to recognize is that the lower voltage waveform segments (ie: the black parts) of the oscillator pattern have no effect on the output. This is characteristic of additive mixing (since 0v + any source = the original source), and allows for some powerful compositing and processing tricks. Continue to experiment with different combinations, and begin using multiple channels of the

TVP to create more advanced effects.

One last basic example is summing signals of different ranges. Create an LFO, using either a VWG or any other module you may have for this purpose (audio modules will work here as well). Place the output of the LFO into the A1 input on your TVP, and connect a triangle ramp (3rd input down on ramps, or use a VWG) into the B1 input. Again ensure that the gain is at 3 o'clock, and adjust the bias control. The effect is essentially the same as adjusting the bias control by hand at a regular rate. This technique is useful in creating complex modulation sources for other operations, and will be a common component of many patches. Try using this sum output to modulate FM, pwm, or sine shaping on a VWG. Experiment with different LFO speeds, different combinations of multiple TVP channels, and various bias settings. (Patch example[fm controlled by TVP sum of LFO and ramp patch])

TVFKG

TVFKG DIAGRAM AND CONTROLS

Multiplying Signals

Multiplication of signals (also known as "gain") happens throughout all of your video modules, at almost every input jack. When you attenuate an input signal, you are actually multiplying the signal by a static value between 0-1v. The shape and frequency of the incoming signal does not change, only the amplitude. Multiplication operations are invaluable for fine control over your patches. Although the vast majority of these operations will happen "behind the scenes", we can perform explicit multiplication operations using a "voltage controlled amplifier" or VCA. It may be more helpful (and accurate) to think of these as voltage-controlled attenuators, since they do not actually amplify the incoming signal. A VCA typically has two inputs and one output, and simply multiplies the two signals together, presenting their product at the output. This produces a very different effect than combining signals additively, as you can see from these figures (figure[additive vs multiplicative waveforms]).

Many of the examples we discussed in the last section can be equally useful with multiplication operations, although the results will be quite different. The LZX system does not have a standalone, dedicated VCA module (although there are free DIY schematics available, and other manufacturers are starting to offer modules based on these). However, the Triple Video Fader and Key Generator offers standard VCA capabilities (and more) when placed in “Fade” mode. Since this module is designed for cross-fading, it has an additional input. A mathematical representation of this is:

$$A*(1-CV) + B*(CV) = Output$$

Notice that if we do not have any signal going into input A, the module will act as a basic VCA, multiplying the B and CV inputs. For the next few examples, we will look at basic multiplication using just these two inputs on the module.

Take the topmost sawtooth output from the Video Ramps module, and plug it into the B1 input on a TVFKG.