FM Tone Generator Seminar by Noritaka Ubukata

The producer of the "Shofuku" voice ROM for the DX7, as well as many other synthesizer sound data collections. Mr. Ubukata was also involved in producing the factory preset sounds of the SY series and other models. He has also been involved in many recordings as a synthesizer programmer. Currently he is active in producing games, commercials, television programs, composition for events, and music production.

His home page is
http://www.asahi-net.or.jp/~bq5n-ubkt/

Session 1
The world's first synthesizer

Let me start by asking you --- What do you think was the first synthesizer in the world?

The answer is the Pipe Organ. That gigantic instrument that can be seen in large concert halls or churches is actually the first synthesizer ever made. I realize that this is a strange and perhaps confusing way to begin, so let's back up and start by asking just what sort of instrument we mean by a "synthesizer."

Looking in a dictionary tells us that a "synthesizer" is a person or device that combines things to form a whole, or an electronic device that produces sound in this manner. Today of course, "synthesizer" refers to the latter destination. However in actuality, most synthesizers of our time are keyboards or rackmounted modules that play back sounds that have been sampled from other instruments. In other words, two devices that originally were completely different (i.e., a sound-synthesizing instrument and a sample-playback instrument) are today treated as a single category. But if we trace back the roots of the synthesizer as a device that "combines sounds to create a whole," the pipe organ is where we end up.

So --- it is true to say that the synthesizer has a really long history. You are probably asking just why a pipe organ can be considered a synthesizer. After all, a sampled pipe organ sound produced by a synthesizer of today sounds just like a pipe organ, and not at all like a synthesizer(!)

In order for you to appreciate how the pipe organ is the ancestor of the synthesizer, you need to understand something about instrumental sounds (including human voices and the cries of animals and birds).

In the same way that air or water are compounds or mixtures of various elements, instrumental sounds (which we will hereafter refer to as "sounds") are also "mixtures" of various units called "overtones" or "partials." The term "partial" is actually more precise, but the term "overtone" is more widely used, so that's the term that I'm going to use in this series of lectures.

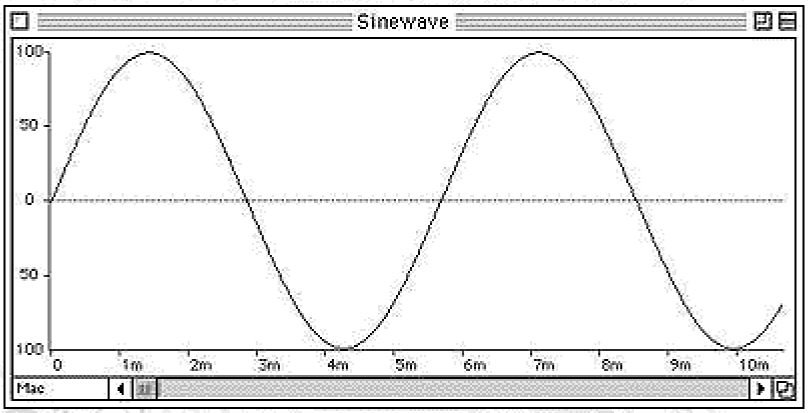
Let's take the example of water. Everyone knows that water --- H2O --- is a compound of hydrogen and oxygen. This is pure water. Only once in my life have I ever drunk this "pure water," and I can tell you that it didn't taste very good!

Naturally occurring water (from a spring or well) contains minerals such as calcium or sodium in addition to this "pure water," and if (for example) the water is from a mountain stream, it may also contain "impurities" such as sap from trees. However, the components that make water taste good or bad are actually a minute fraction of the whole, and the fluid that we are discussing is still essentially a compound of hydrogen and oxygen.

The same goes for air. Air is a mixture of nitrogen, oxygen, and carbon dioxide, but in the city it also contains automobile exhaust, emissions from factories, and lots of other impurities that are probably bad for our bodies, making it smell like "bad air." Meanwhile, the air in a natural environment contains scents of the forest and earth, and various ions, making it smell like "good air." But it is these "impurities" that make the air smell bad or good --- the majority of what we call "air" remains a mixture of nitrogen and oxygen etc.

Now let's get back to sounds. Just as water or air are made up of elements such as oxygen and hydrogen, sound is also made up of "elements" called "overtones."

If we look at the waveform, each of these overtones is something called a "sine wave" (figure A).



[figure,`]

If you listen to a sine wave all by itself, it is simply a featureless "tone."

You've probably heard sine waves used to check audio equipment. (Listening to a sine wave for extended periods of time may be harmful to your hearing.) There are no objects in the natural world that produce a pure sine wave. Even sounds that are very close to a sine wave (such as a harmonic on a guitar) always contain some noise or other components. To get a pure sine wave, you must use some device to filter a sound to isolate the sine wave, or you can generate a sine wave electrically or electronically. Using a filter to isolate a sine wave is similar to the process of refining iron ore or gold ore to isolate pure iron or gold. Of course, iron or gold cannot be synthesized artificially. (Although this is what the alchemists of the past attempted to do!)

But it is possible to create a sine wave artificially. And when modern electronic synthesizers wer invented, this became very important. Let's leave this aside for a time, and return to pipe organs.

Even in the days when there was yet no way to electrically produce a sine wave, people knew that sound consisted of overtones (sine waves) of differing pitches.

For example, suppose that the C3 note was played on a bowed string instrument such as a violin. The C3 note contains a C3 sine wave (this is called the "fundamental"), but also contains additional, higher-pitched sine waves in an orderly sequence of C4 (one octave higher), G4 (a fifth above that), C5 (two octaves above the fundamental), E5 (a third above that), and G5, C6, ... etc., and all of these sine waves are heard together by our ear as the sound of the violin.

When you play a note on a pipe organ, air is sent to a pipe, and the pipe (with reed) produces a sound that is fairly close to a sine wave. Some pipes produce sound that is similar to a flute, and more muted. Each note on the keyboard is associated to more than one pipe (on a large organ, this can be dozens), and these pipes are pitched in the regular sequence that we noted earlier when we talked about the overtones that make up the sound of a violin. In other words, each pipe produces one of the overtones that make up the overall sound. For some reason, a pipe organ also has pipes that produce pitches of an octave below the fundamental and a fifth above the fundamental.

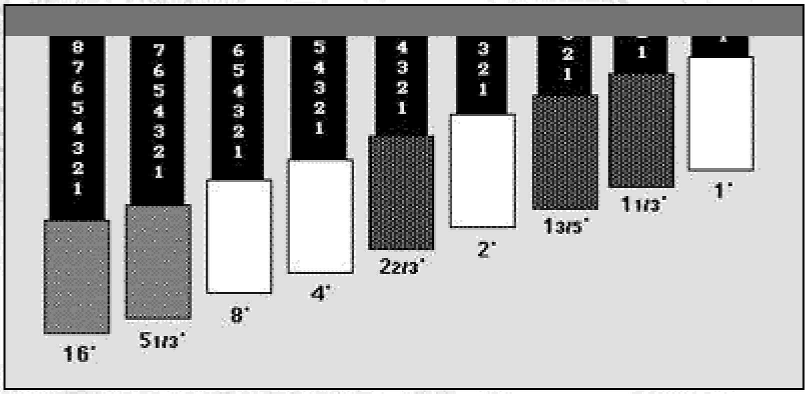
I have no idea how many pipes are found on the largest pipe organs in the world. Since the reverberation of the large hall or cathedral in which these instruments are placed is an important part of their sound, these instruments are really cases in which the entire building "is" the instrument. (And to think that in the past, they had to rely on human power to pump air to all these pipes!)

So why am I saying that these pipe organs were the first synthesizers? The reason is that levers called "stops" could be operated by the organist to specify whether or not air would be sent to each of the pipes that correspond to the overtones. The selected combination of stops allowed the organist to synthesize a wide variety of sounds, such as clarinet, oboe, or flute.

When I actually had the opportunity to play a real pipe organ, I played around with combinations of stops that would not normally be used. Finding that I was able to create sounds reminiscent of synth strings and even buzzers, I began to realize that this instrument really was a type of synthesizer.

Historically, pipe organs have been used mainly in churches, but it is not the case that all of them are necessarily examples of the "building as instrument." Building a pipe organ is a very expensive proposition. Smaller instruments with fewer pipes, and even portable organs ("portative" organs) were also developed. When we get into this category of pipe organ, there are far fewer resources for synthesizing sound, but these smaller instruments still have a beautiful character of their own.

Let's move onward in history. Once people learned how to use electricity and produce artificial sine waves, a certain American wondered if it would be possible to electrically synthesize the sound of a pipe organ. He thought that if this were possible, even small rural churches would be able to accompany their hymns with pipe organ sounds (obviously, assuming they had electricity). The result of his inventive work was the now-famous "Hammond organ." The style of Hammond organ that became most wide-spread had nine sinewave generators for each key (originally the sine waves were produced by geared wheels and electrical coils), and the volume of each sine wave was controlled by levers called "drawbars." (Figure B)



[Figure B]

Some of you may be wondering about the meaning of the numbers and fractions such as 16' or 8' that are located beside each drawbar. These numbers indicate the length of the pipes of a pipe organ. 8' means an eight-foot pipe, and 4' means a four-foot pipe --- half the length of an eight-foot pipe, and therefore one octave higher. A two-foot pipe would be yet another octave higher. The fractions indicate pitches of a fifth or a third, and are color-coded for each recognition.

Even today, some synthesizers use values such as 16' or 8' or 4' to indicate pitch in octaves. This is an example of how synthesizers have pipe organs as their ancestors --- it's part of their "pre-history."

In the next session I'm going to explain more details of how this organ worked, and will also talk about what led up to the FM tone generation system used in the DX synthesizers, and about the predecessors of sampling tone generators. See you then!

Seminar Top

Session 2
From the organ to the analog synthesizer

In the previous session I talked about an ancestor of the synthesizer --- the Hammond organ, which was an attempt to make an electrical version of the pipe organ. Pipe organs use levers called "stops" to specify the pipes to which air is sent, thus determining the resulting sound. In a similar way, the Hammond organ used levers called "drawbars" to electrically control the volume of each sine wave (each overtone), in this way allowing the musician to create a variety of tones. At the time that the Hammond organ was introduced, it is said that blindfolded listening tests were conducted, and that a majority of people were unable to distinguish between the Hammond and a real pipe organ. If this is true, I am sure that it is because no one at that time had ever heard a Hammond organ! It is highly unlikely that anyone today would mistake a Hammond organ for a pipe organ.

Pipe organs are often inseparable from the building in which they exist. However a Hammond organ used a rotating speaker --- the "Leslie" speaker --- to distribute the sound throughout the room and produce a more spacious impression. The Leslie speaker used motors to actually rotate horn-type speakers for the mid- and high-frequency ranges, and a separate rotor speaker for the low-frequency range. This had the effect of making the sound reflect off various surfaces of the room. It was a real innovation, and I would be curious to know just what type of person invented it! It was an amazing feat of manufacturing, and I believe that these are still being produced to this day.

In any case, when compared to a real pipe organ there were quite a few differences. The amplifier added a significant amount of distortion, the speakers were noisy and in addition had to rotate (which produced a swishing sound), the sound from the horns tended to crack --- all of these things must have been real headaches for the designer. In addition, the Leslie speaker could be speeded up to a very rapid rotation, which produced a sound that was quite unlike any pipe organ. To top it off, the Hammond organ itself had functions such as vibrato and percussion which are not found on any pipe organ. There was even a "click" sound when you pressed a key. Later generations of musicians found this quite a funky feature, and happily used the Hammond for jazz and rock. (Initially it was intended for use in churches, so the musical link may actually be through Gospel to R&B, jazz, and rock, but I'm no music historian, so don't quote me on this.)

Although the Hammond start it, there were actually several other organs that used drawbars to combine sine waves. In any case, different combinations of drawbars would supposedly produce flute sounds, string sounds, or reed sounds etc., but to our ears today they all sound simply like organ sounds. Later I will talk about why this is the case, but for now let's just say that simply combining overtones to approximate a given tone is not enough to simulate the sound of a particular instrument.

At this point, let's summarize a few points about Hammond organs. I've listed the main elements that determine the sound of these organs.

1. The volumes of several overtones can be adjusted to modify the tone.

2. The sound is played through a rotating speaker.
3. There is a function called "percussion" .(details will be given later)B
4. There is a "key click" noise when a note is played.B

These characteristics are going to be important when we later attempt to use the DX board to simulate the sound of an organ, so make sure that you understand them.

Regardless of the fact that the Hammond organ was created as an imitation of the pipe organ, its strong character gave it its own identity --- the "Hammond organ" sound. Separately from the Hammond organ, it is an amazing fact that sine wave additive sound synthesis devices were actually invented back when not even transistors (much less analog synthesizers or LSIs) existed. These mammoth systems were as big as a house. I don't know how many overtones these systems were able to produce. (From my own experience I have found that 128 overtones will give a pretty faithful simulation.)

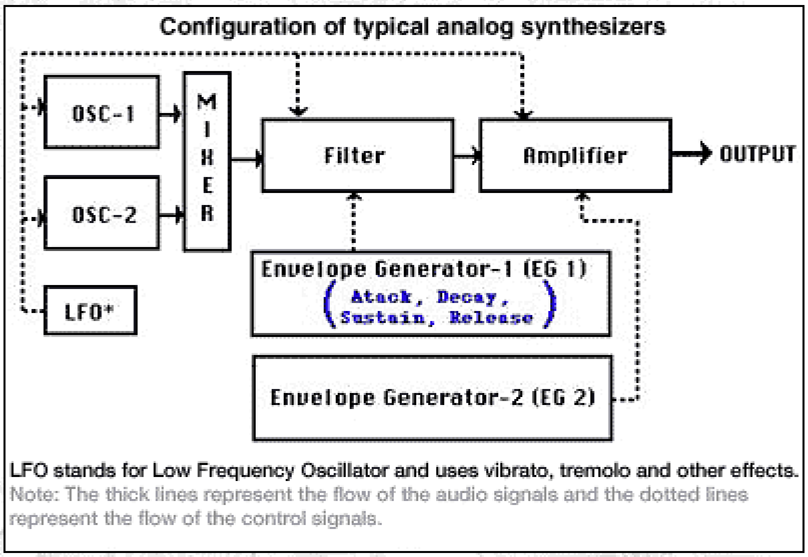
It is said that piano sounds generated by this sinewave additive synthesis device fooled the majority of people in a blindfolded listening test, but as in the case of the Hammond organ, it seems to me that we shouldn't rely very strongly on the ears of people in the past (!)

Many years passed after the invention of the sinewave additive synthesizer, and in time, its conceptual descendent appeared --- the Yamaha DX7, representing the first generation of sinewave multiplicative synthesis instruments. This technology produced sound by multiplying sine waves with each other, and is usually referred to as FM (frequency modulation) synthesis. I remember how I used to go to recording sessions struggling under the weight of a DX7 in a soft case with a shoulder strap and my effect units in another case. Today, it continues to amaze me that all this can fit in such a tiny circuit board.

I'm going to discuss the DX7 in depth later. Let's put it aside for a while, and take a step backward to look at the filter-style analog synthesizers (subtractive synthesizers) that were so widespread, and at the very special case of the tape-playback instruments. The reason is that unless you clearly grasp the differences between FM, filter-style, and tape-playback methods of producing sound, you are going to be progressively more confused as we continue. Some of you may think that this is already confusing enough! But the fact is that "musical instrument sound" is a highly complex phenomenon, and that if your task is to precisely capture the sound of a particular musical instrument, you are going to be dealing with a lot of complexity no matter what type of synthesizer you are using.

What are today called analog synthesizers can more accurately be called "filter-type" analog synthesizers. We parenthetically called these "subtractive" synthesizers because they use a filter to process a previously prepared waveform (an electrically generated "buzzer-like" waveform, not a sampled waveform). Of course, some such synthesizers contained two or more tone-generating units, and added these together to produce the final sound, but the basis for this method is still that a filter is used to process a waveform.

After Dr. Robert Moog first developed these filter-type synthesizers (which we will refer to as "analog synthesizers"), the word "synthesizer" came to be synonymous with synthesizers of this type. Today, there are synthesizers (such as the Yamaha AN1x) that use DSP technology to simulate the sound-producing mechanism of these analog synthesizers, but as far as I know, analog synthesizers in the strict sense of the word are no longer being manufactured to a significant extent. Analog synthesizers were structured as shown in figure A.



[figureA]

The section marked OSC stands for the oscillator. The waveform produced by this oscillator was processed by being sent through a filter. The oscillator was usually able to produce sawtooth, square (pulse), and triangle waveforms, and some models were able to mix sawtooth and triangle, or even to mix all of the waveforms they had.

There is actually more to these synthesizers than the diagram shows. For example, control signals from the keyboard or pitch bender etc. were also involved, but I've omitted this to keep the diagram from becoming cluttered. But still, the system is structured rather simply, isn't it?

Some analog synthesizers were even simpler, and had only a single oscillator and EG. Conversely, some allowed you to use an EG to modulate an oscillator (today we would call this a pitch EG), and others had three oscillators and used one as an LFO. One driving force behind the development of these instruments was the need for portability and the need to make the circuitry simpler so that the price could be kept down. The price was a particularly important factor. When I first purchased an imported Mini Moog (a classic analog synthesizer that still has loyal devotees today) sometime in 1973 or '74, I remember paying over two thousand dollars for it. Ouch! And this for a monophonic synthesizer that could not play chords.

And this for a monophonic synthesizer that could not play chords.

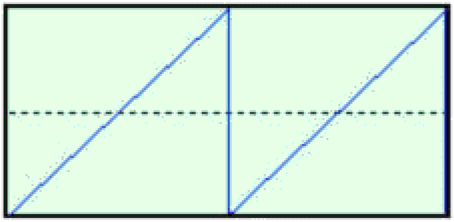
Yes, you read correctly. Today, most synthesizers are at least 32-note or 64-note polyphonic, or even 128-note polyphonic, but in the beginning, monophonic was the rule.

In order to play chords, there was the technical problem of key assignment, and in addition, the need for as many sound-producing circuits as notes: if you wanted to make a eight-note polyphonic synthesizer you simply had to provide eight synthesizers worth of circuitry. Even if the price did not end up eight times higher, it was certain to be several times more expensive, and the size was correspondingly larger. Not many people would buy such a monster.

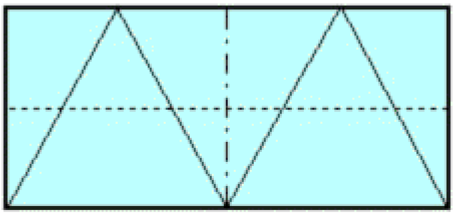
Then miniaturized components such as the integrated circuit (IC) and large-scale integrated circuit (LSI) appeared, and synthesizers that could play chords became available in the bargain price range of ten or twenty thousand dollars (!), and in conveniently transportable packages that weighed only (!) thirty pounds or so (over a hundred pounds for some). A friend of mine owns one of these early polyphonic synthesizers (unfortunately not in operating condition), the circuitry of which was so laughably complex that it was not only impossible to fix the instrument, but even to determine where the malfunction had occurred! However, one tended to develop an affection for these synthesizers, and it's hard to let them go even when they are obviously unusable.

I used to have a duophonic (two-note polyphonic) synthesizer called an ARP 2600, and since it was unusable and simply gathering dust, finally brought myself to sell it several years ago, on the condition that the buyer promise to actually use it. It was a sad occasion.
In our next session I'm going to talk about the ancestor of the sampling tone generator (the tape-playback keyboard), and then get into some hands-on work with the DX board. (To be precise, I should say hands-on work with the editing program.)

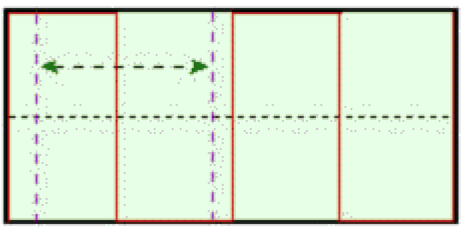
Typical waveforms for analog synthesizers



Sawtooth waves



Triangular waves



quare (pulse) waves

With regard to the pulse waves, as indicated by the dotted lines and arrows, there are units where the width of the amplitude can be modified and units where the style is fixed in two or three patterns.
Fifty percent of the pulse waves displayed

Note: The waveforms for actual analog synthesizers cannot be drawn out in the perfect patterns shown in the diagrams. Depending on the manufacturer, certain lines will be warped, the edges will be distorted, noise will be mixed in and the entire waveform will be distorted. This distortion and noise provide the characteristics for each unit and create the so-called 'tone flavor.'

[Seminar Top]

Session 3.
From tape-playback to PCM

The phrase "tape-playback tone generator" probably gives you an impression of something highly complex, but basically these were just analog sample players.

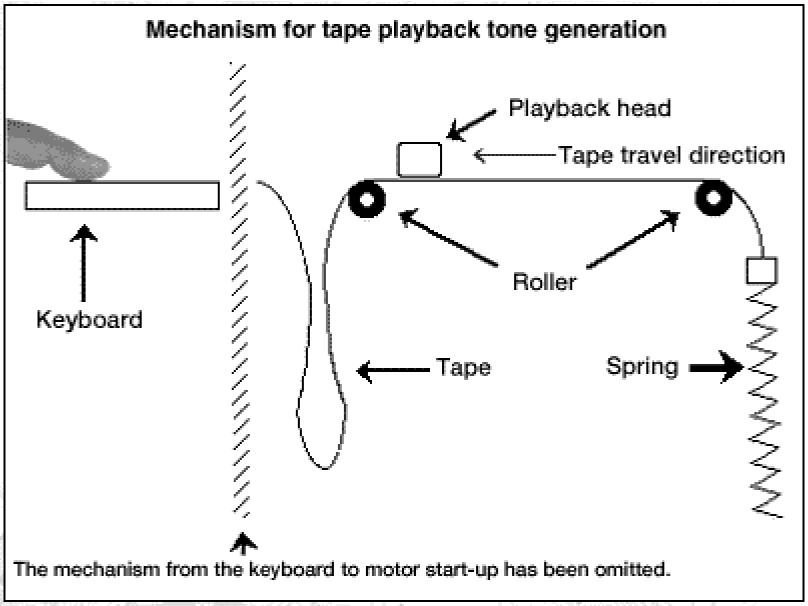
This category of instrument was represented by the Mellotron and the Chamberlin. My dim recollection is that the Mellotron was made in Germany, and the Chamberlin was made in the United States.

Both of these produced sound by playing back tape recordings of acoustic instruments. Basically, the keyboard functioned as the start button of the tape recorder.

There was one tape for each note, and on each tape was recorded a single pitch of an instrument. If we use the analogy of today's digital samplers, this would mean an independent sample for each semitone --- truly extravagant!

The model which became most popular was the Mellotron 400. This instrument had 36 keys, and as controllers had a master volume, a sound select switch that let you select one of three sounds, and a tuning knob. However in reality, tuning was basically impossible, since the pitch when recorded was already wrong in some places. This meant that if you tuned to make the C pitch correct E would be wrong, and if you tuned to A then F# would be wrong. By the way, I once used a tuning meter to measure the pitch of a Mellotron owned by an acquaintance, and it was truly frightful. Still, a "C-E-G" chord played by the string section of a real orchestra is rarely precisely the same as a "C-E-G" chord played by the brass section, and it is fair to say that the slightly out-of-tune nature of this instrument was part of its charm. Another problem was that all tapes were driven by a single motor, meaning that the more notes you played, the greater the load placed on the motor, and the more the overall pitch would drop --- somewhat like a bus engine that has to work harder and harder as more passengers get aboard.

Take a look at figure A.

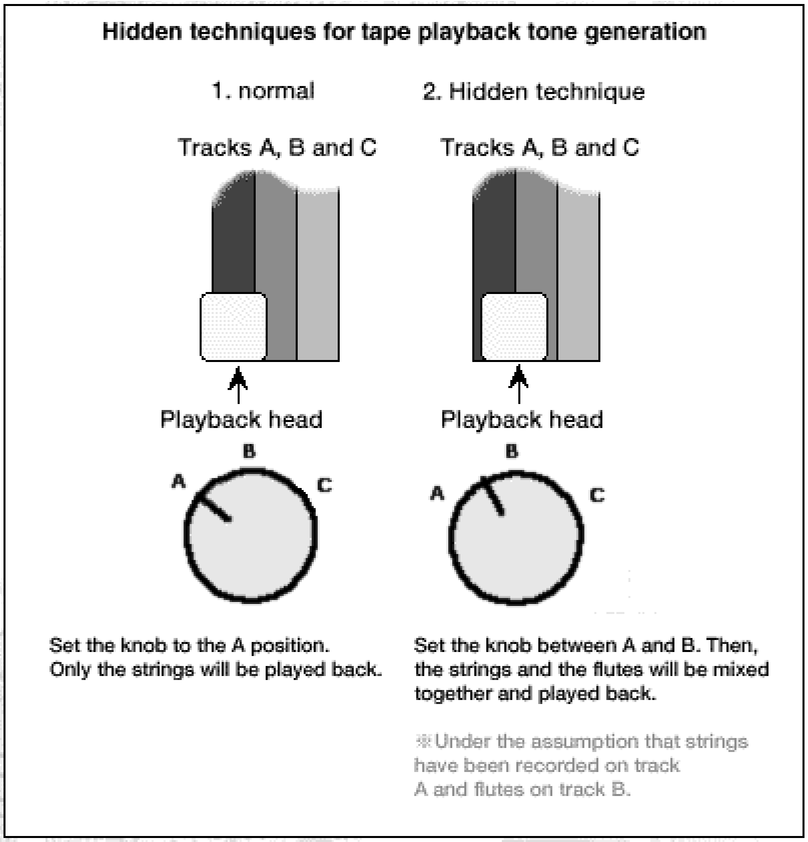


[figureA]

This shows the basic structure of the Mellotron. The clever part is what happens when the keyboard is released. The playback head would lift up, and simultaneously a spring that was tensioning the tape would rewind the tape in an instant. When you again pressed the note, the roller would begin rotating, the playback head would contact the tape, and the recorded sound would playback. With this structure, it was obviously impossible to play rapid passages, and if you attempted to do so, the tape would be in danger of breaking. Furthermore, the length of the tape was limited, and the sound could be sustained for only seven seconds or so. (True, seven seconds per note is long enough for most cases.) If you listened closely to each tape, you could hear artifacts such as the flute player taking a breath and blowing again, or the string player's bow contacting the string. This gave the sound more life and personality.

The Mellotron was an extremely delicate instrument, and experienced continual mechanical problems. When being transported to a concert, the tapes would sometimes get tangled, or the metal plate that maintained tape tension would get bent.
Let's talk about the sounds of this instrument. The tapes of the Mellotron had three tracks, and these tracks usually contained strings, brass, and chorus recordings. Some contained flute instead of brass, and there were several varieties of strings and chorus.

I have heard that you could even send your own recordings to the company, and they would create custom Mellotron tapes for you. For us today looking back, all of this sounds like a mind-boggling hassle. Not many Japanese bands used this instrument, but I have heard of some who recorded sound effects on the Mellotron tapes, or recorded tapes of monophonic synthesizers (in the days before polyphonic synthesizers) so that they could play chords. An undocumented trick on these instruments was to place the sound selector switch between the A and B positions so that the two sounds would be mixed (see figure B).



[figureB]

My vague recollection (it's been a long time!) is that you could regulate the balance between the two sounds by the location at which the switch was placed, but I'm not sure about this. This technique is similar to how guitarists used to place the pickup selector switch of a Stratocaster between two pickups to get a mixed sound. Most Stratocasters today have switches that let you place them mid-way between two pickups, but it used to be a bit of a trick to get the switch to stay between the two settings.

Anyhow, the Mellotron was difficult to transport, delicate, and required an enormous amount of maintenance, so various improvements were attempted. There was the Virotron which used cassette-type tape loops (the playback had no attack, so I can't say that it was a very good idea), the Orchestron and the Novatron (I think one of these actually used a laser disc), and others. Some even appeared after the inventions of sampling keyboards and modules, but the progress of digital audio technology meant that these mechanical playback devices were soon replaced by sampling tone generators (hereafter referred to as PCM tone generators). One way to look at this is that the sound-recording media changed from tape to memory chips or hard disks. The basic idea is the same as in the days of tape playback instruments.

PCM tone generators are probably the most widespread type of instrument today. Strictly speaking, however, they are not synthesizers in that they do not "synthesize" sound, but simply use a filters and EGs to process a recorded sound.

In other words, the progress of digital technology brought together the "pre-recorded playback" tone generator represented by the Mellotron, and the "sound processing by filter" type of tone generator used by analog synthesizers.

And finally we are ready to begin our long-awaited (I hope!) discussion of FM tone generators. FM tone generation is a method that has fundamentally different roots than either analog or PCM, but we've run out of space in this session. In our next session we will begin by actually creating some sounds.

I would actually like to write much more about analog synthesizers and about the Mellotron, but that would be a whole series in itself!

I'll close with a bit of trivia. In the United States, there was a fear that these tape playback instruments would damage the livelihood of studio string and wind musicians, and sufficient pressure was applied from the musician's union that the Mellotron and the Chamberlin were actually taken off the market. Hearing these instruments today it is hard for us to imagine how they could ever replace the real thing, but to people hearing them for the first time, they must have been quite a shock.

I myself heard the Mellotron for the first time at a concert by the Dutch band "Focus," and have never forgotten the impact it made on me.

Seminar Top

Session 4.
FM tone generators at last

Thank you all for your patience. From this session we begin the discussion that you have been waiting for --- FM tone generators.

In our first session, I wrote that that FM tone generators produce sound by multiplying overtones. Actually, the "FM" (Frequency Modulation) in FM tone generators is the same FM as in FM broadcasting. An electrical wave called the carrier is modulated by another electrical wave called the modulator, and this combined signal faithfully conveys the content of the broadcast. The only difference is that the FM in synthesizers has lowered these waves to an audible frequency.

Huh?

Ok, I confess --- I'm just trying to make it sound difficult. You see, I'm trying to make you realize that just as you can enjoy an FM broadcast without understanding the theory of FM broadcasting, and just as you can record movies on your video deck and even edit your home videos without understanding just how a video recorder operates, it's not impossible for you to take full advantage of an FM tone generator without understanding the theory behind it.

Frequency modulation is actually a very familiar thing. The vibrato that you use in a vocal or instrumental performance is a type of FM, and the way in which one sometimes sees children amuse themselves by pounding rapidly on their chest or throat while intoning in a strange voice "Hello earthlings, we have come from ..." --- is also, broadly speaking, FM. But while these familiar examples of FM all consist of modulation at a frequency much lower than the sound that is being modulated, most of the FM used in synthesizers to produce new sounds takes place at the same frequency as the original sound (or even far higher).

Don't feel that you need to understand the foregoing material. There is actually nothing that you absolutely have to memorize in order to use an FM synthesizer to create sounds.

The most well-known synthesizer based on FM tone generation was the DX7. It was a classic that earned an undying place in the world of music. The DX7 was the first 16-note polyphonic synthesizer that you could buy without being rich, and its chordal capability was shocking for its time. In those days, the "low priced" analog synthesizers topped out at six note polyphony. Even models that cost in the range of ten thousand dollars could produce only five notes or at the most ten notes. The DX7 was far cheaper, could produce sixteen notes, and was digital!

Ah, in those days, the phrase "digital synthesizer" was magic to so many people. Today it's just taken for granted.

First came the DX7, then the slightly downscaled DX9, the high-end DX1 (this was heavy and expensive, but had a truly gorgeous display), the DX5, DX21, the cute little DX100, the DX7II ... new models kept being added to the DX series just like sequels to a successful movie. But finally with advances in PCM-based instruments (I'm not going to call them synthesizers) and declines in the price of memory chips, PCM began sounding better and getting cheaper, and finally overtook the DX. After all, those were the strange years when most people thought that the quality of a synthesizer could be measured by the quality of its piano sound.

In the early days, PCM keyboards cost tens of thousands of dollars, but they used eight-bit sampling with low sampling rates, and they sounded cheap.

Those of you who have ever used a 68K Mac are familiar with that noisy and rough sound that's a little hard on the ears.

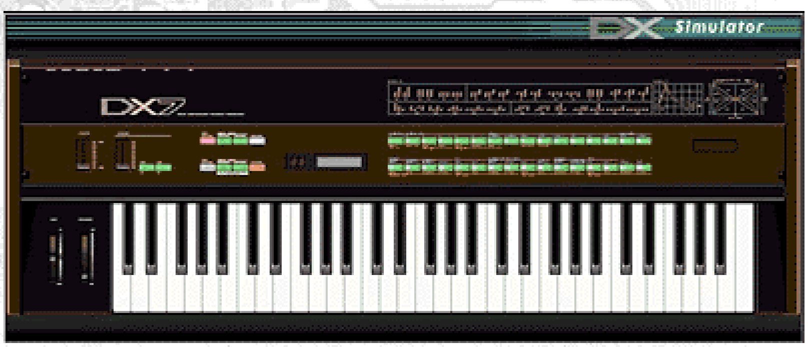
Although this lecture has taken numerous side trips, please be assured that its purpose really is to discuss the DX. They say that history repeats itself, but just like analog synthesizers are now fashionable once again, I have the feeling that FM synthesizers are also going to have a second period of popularity. I myself, who spent uncounted hours tweaking DX sounds, am often still taken by surprise at how good FM synthesis sounds.

The original DX is by now seen only seldom in used stores, and is not so easy to obtain. Not to worry. Now there's a DX on an XG plug-in board --- the PLUG100-DX. And it's even more advanced than the original DX, with low pass and high pass filters, and even EQ. (When I used to edit sounds on the DX, I don't know how often I wished that filters and EQ were built in.)

Details and pictures of the PLUG100-DX itself can be seen in this Yamaha website, so I'm not going to discuss them. Let's jump straight into the interior workings! Don't misunderstand --- I'm not going to ask you to take the board apart.

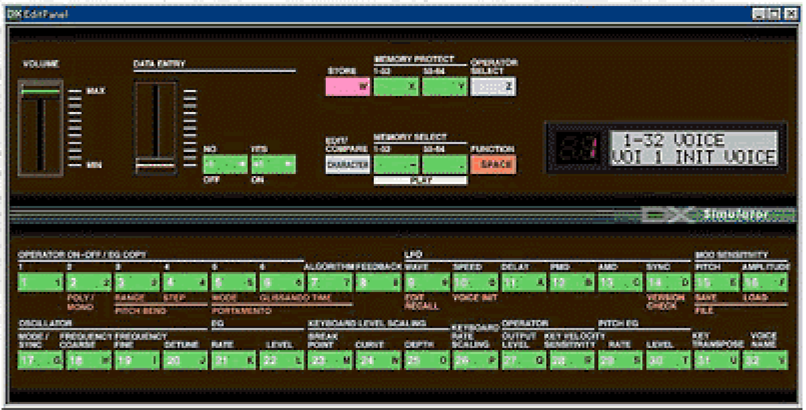
This plug-in board comes with an editor program called DX Simulator, that runs on Windows computers. This program functions as a plug-in for an XG editor such as XGworks or XGworks Lite.

Thankfully, this simulator can be used not only with the PLUG100-DX, but also with the original DX7 or DX7II, and with the TX802 rackmount module. (I myself am wishing that a Mac version would join the Windows version, ...)
Let's take a look at the screen of the DX simulator.



Brings back memories, doesn't it? The design almost looks fresher than the latest synthesizers of today. The DX7 normally had 32 sounds in internal memory, and let you add another 32 by inserting a RAM cartridge, for a total of 64 sound memories. (What a tiny number!) My own DX was expanded to 128 memories, but I've let it go. (No use regretting now.) However I did hold onto my DX7II.

So, in the screen of the simulator, click the keyboard area, and the corresponding note will be sounded. Click the edit panel, and the editor window will open. Pretty neat, huh? Now take a look at the editor window. Except for very minor differences, this is basically the same as the original DX7.



I mentioned earlier that internal memory contained 32 sounds, but you can see from this screen that MEMORY SELECT 1-32 and 33-64 let you store a total of 64 memories. (It might be a bit difficult to see.) You can think of this DX as having a RAM cartridge already plugged in.

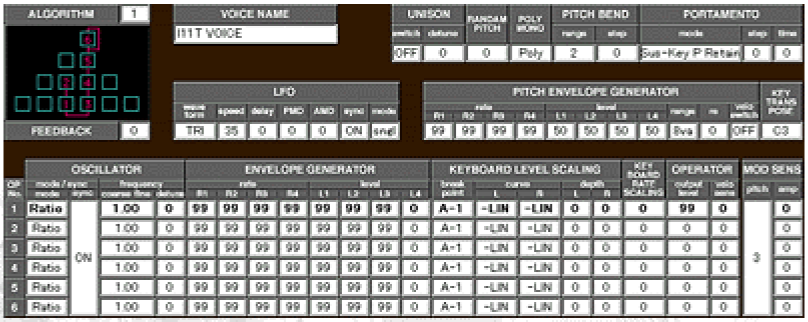
With the default settings, try clicking the lower two rows of light green buttons numbered from 1 to 32. These buttons select sounds from the PLUG100-DX memory that correspond to the preset sounds of the original DX7. However, in what would be the LCD of an actual DX, the display reads ""VOI 1 INIT VOICE." This is finally what we are getting at in this series of lectures. (Yea!)

You can select and play the preset sounds, and even use the DX Easy Editor plug-in software to perform simple editing of the sounds (although the available parameters are limited), but we will skip over all of this. It is the ultimate purpose of this lecture to create a sound from scratch! And everything starts from the "INIT VOICE" that you see here.

Let's go right ahead and make this "INIT VOICE." It's very easy. Click the yellow button (the one marked SPACE) that is located at the right edge of the middle of the second row of buttons from the top. Next, click button number 10 (the first button in the lower row, marked VOICE INIT). The (simulated) LCD will rudely ask "ARE YOU SURE," so just click the YES button (of the two buttons located beside Data Entry. This completes the "INIT VOICE." Easy, wasn't it? This sound is a totally unprocessed sine wave.

Unless you are a true fanatic who insists on using the same operations as on the original DX to create the sound, I suggest that you click the left-most button in the menu bar of the window that appears when you start up the simulator, and do your sound editing in the "DX Edit List" that appears. I will be using this "Edit List" in my explanation. The button marked "DX bulk transmission" is also very important. When you wish to store a completed sound or to compare sounds, you will return to the edit panel window, but the "Edit List" will be the center of operations while you are editing a sound.

By the way, the "INIT VOICE" that you see in the "Edit List" has this parameter structure.



It is a bit overwhelming to see all the parameters lined up like this, but oscillator 1 is the only one that is actually sounding, and the sound is like this.

MIDI Data
dx_init.zip
(4.72kbyte)

SoundVQ Data
dx_init.vqf
(38.4kbyte)

Load 64DxVoice4.MID first as cartridge file.

It's a sound with neither character nor interest. (Oh, and please turn off the XG reverb.) This sound is going to undergo an amazing transformation. Next will be the semi-final session of this series, in which we will create an organ using sine wave additive synthesis.

Seminar Top

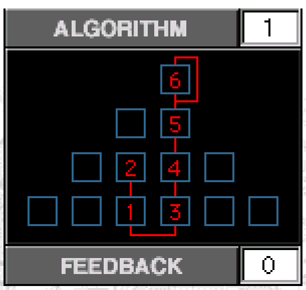
Session 5.
Using additive sine wave synthesis to create an organ sound

First we're going to try using the DX (which is how I'm going to refer to any FM synthesizer from now on, including the PLUG100-DX) to do some sine wave (additive) synthesis.

Since I was talking about FM in the previous session, some of you may be flustered by my taking a side trip into sine wave additive synthesis. But remember the saying, "The journey of a thousand miles begins with one step."

The word "synthesize" means "to create by bringing together things of a different character," and it's important to remember this point. That uninteresting sine wave that you heard toward the end of the previous session will, when combined with two or three other sine waves, change in a mysterious way.

Now let's recall the "INIT VOICE" that you created (?) in the last session. If you didn't store it, read the previous session once again and re-create it.
Notice the box located in the upper left of the screen, the one marked ALGORITHM.



If you want to know the general meaning of the word "algorithm," please consult a dictionary. Just remember that on the DX, the word is used to mean "a way in which the operators are connected." I almost forgot --- the sine wave oscillators of the DX are called "operators." (Unlike the oscillators of an analog synthesizer, these are able to produce only sine waves.) The reason that they are called operators rather than oscillators is that each operator can change its role depending on the algorithm, and may be either modulated or be a modulator. After all, doesn't a telephone operator have more than one role, sometimes answering a call and sometimes placing a call? DX operators are actually like those recirculating pond fountains that never run out of water and just keep flowing forever --- they keep continuously producing a sine wave. (That's how they actually do function.)

On the DX, the role assigned to each operator will determine the sound that results. Each operator is controlled directly by its amplitude EG, and mixing is also done by adjusting the output level of each operator.

Since the oscillators of the DX play more than one role, we're going to refer to them as operators.

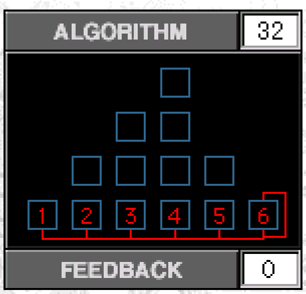
The current status is clearly shown on the screen. We have two stacks: operator 2 is sitting on top of operator 1 (child turtle on top of parent turtle), and operator 6 is on top of 5 which is on top of 4 which is on top of 3 (great-grandchild turtle on grandchild turtle on child turtle on parent turtle). On the DX, the operators in the bottom are always the output operators. **Each operator stacked above modulates the operator below it.** It's sort of like a team of acrobats. Suppose that there were four people stacked up as in the right-hand stack of the algorithm, and the top person thought of something funny and began shaking with laughter. The person below him shifts his weight in an attempt to keep his balance. The person below that one reels under the shifting weight of the person above. And the poor chap on the bottom, bearing all that weight and movement, lets out a yell! If the shaking and shifting is minor, the bottom person might let out only a small yell. But as the movement increases, his complaint would escalate --- "Arrgh!!!" --- and finally he would collapse with a scream into chaos (noise).

If you think of the algorithm in this way, the operator stack on the left is much easier to understand. There is only one person above, and even if he wobbles a fair amount, there is a limit to how greatly the person below can be affected. It is unlikely that this stack of operators would ever be able to produce complete noise.

However in the operator stack on the right, the movements of the person on the very top are magnified and transformed as they are transmitted downward, so there's no telling what effect they will have on the bottom operator.

Now let's move from explanation to actually tweaking the data.

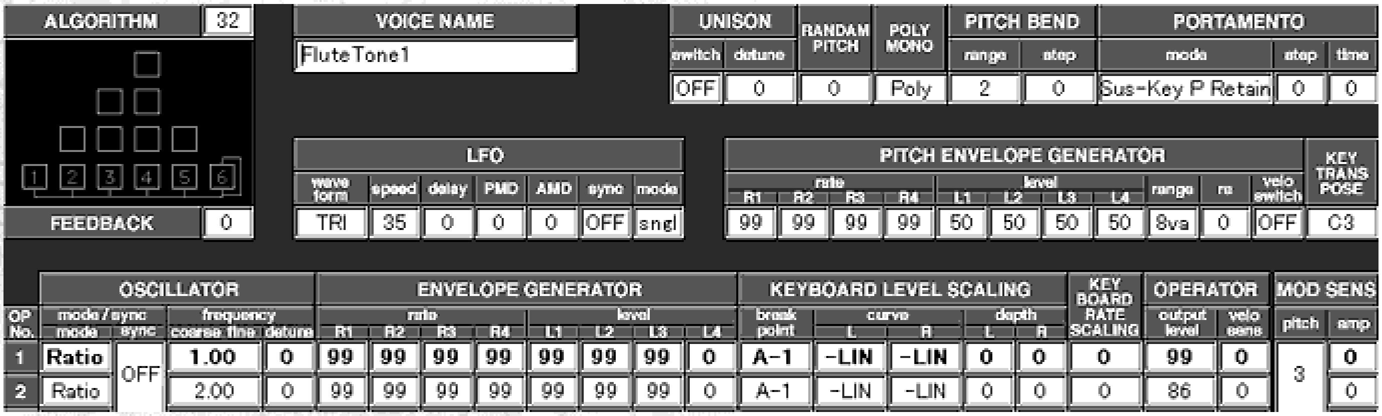
The box located at the right of ALGORITHM currently has a value of "1." Using either the mouse or the keyboard, input a value of 32 into this box. The arrangement of operators will change to the configuration shown in the following diagram.



All of the operators are now arranged side by side. In DX algorithms, the bottom row is for output only. This means that this algorithm has six sine wave oscillators placed side by side. The sixth one has something a little different, but I'll explain this later --- ignore it for now.
With these six sine wave oscillators (operators) arranged side by side, set the level of each operator to the maximum, and set the Frequency of each in steps of one, starting with 1.00, 2.00 ... etc. This will produce the same state as when the six drawbars 8' through 5-1/3' of a Hammond organ are pulled out all the way. (See diagram B of session 1.)

By adjusting the output level of each operator (which corresponds to the drawbars of a Hammond organ), you can use your DX as a sort of organ. The pitch of each drawbar is fixed on an organ, but can be freely changed on a DX (!).

As I mentioned when I discussed organs earlier, different combinations of drawbars allow you to create different types of sounds such as reeds or flutes, so let's try this out. First we'll attempt a flute sound. We will use just two drawbars --- I mean operators. Refer to the following diagram and make the corresponding settings.



(Author's note: To conserve screen space, unused operators are not shown. Similarly, only the operators that are used will be shown in the edit lists that follow.)

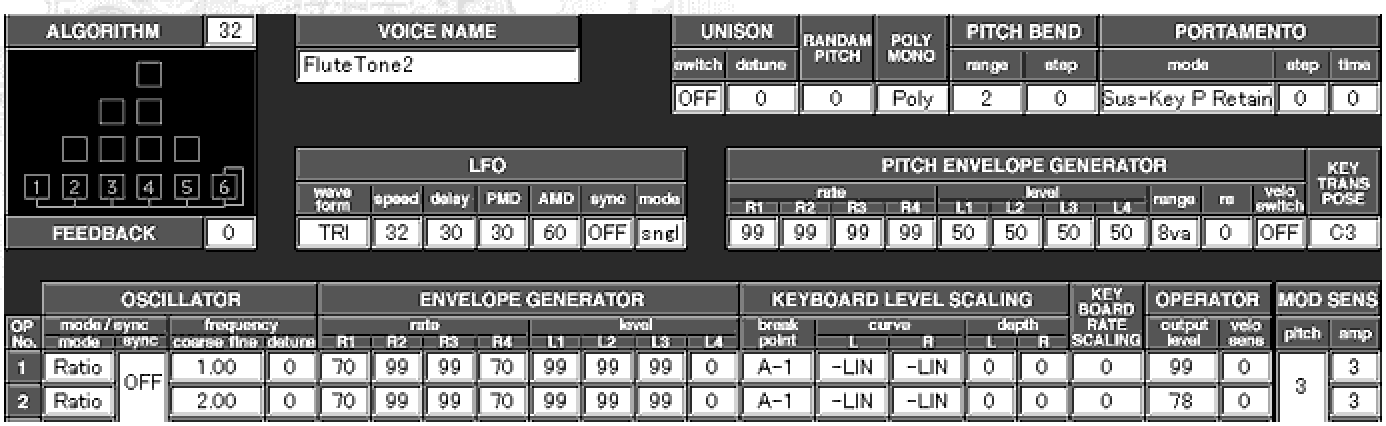
The operator output level display has the potential of being misleading. The volume of the second operator is actually about half of the first operator, but if you think that a value of 50 must be in the middle of the range 0--99 you are mistaken. The sound will be so faint you won't even hear it. Trust your ears, and set the value that sounds to you like a half. (As you can probably tell, I don't sweat the details.) If you happen to be an obsessive-compulsive personality, you may input the sound to an audio device that contains a level meter, and adjust the levels precisely.

Play a few notes using this sound. Does it sound like a flute?

SAMPLE DATA DOWNLOAD **MIDIData** flutone.zip (4.77kbyte) SAMPLE DATA DOWNLOAD **SoundVQData** flutone.vqf (46.3kbyte)

Load 64DxVoice4.MID first as cartridge file.

I guess it just sounds like an organ anyway. After all, the sound appears abruptly and remains at the very same level all the while you continue pressing the note, and then disappears with a blip the moment you release the note. So, now try changing the EG as shown in the following diagram. While you're at it, add some vibrato. Since this is a wind instrument, you can add a bit of variation to the amplitude as well as to the pitch.



How does it sound now?

Well, rather than being flute-like, it sounds more like a synthesized human voice, or something that you would expect to hear in the background music of an old science fiction movie. Flute sounds actually consist mainly of the fundamental and the second partial, but also contain lots of subtle and minute elements which play an important role in the character of the sound. A good flute sound is difficult for a synthesizer to produce. Even for sampled flutes, I have never come across a sound that made me sit up and say "That's it!"

SAMPLE DATA DOWNLOAD **MIDI Data** fluedit.zip (4.78kbyte) SAMPLE DATA DOWNLOAD **SoundVQ Data** fluedit.vqf (46.7kbyte)

Load 64DxVoice4.MID first as cartridge file.

However for some reason, it's possible to get a fairly decent clarinet or oboe. It's a mystery.

Let's try a clarinet.

ALGORITHM	FEEDBACK	VOICE NAME
32	0	Clar.tone1

UNISON switch	UNISON detune	RANDAM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0

LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	35	0	0	0	ON	sngl

PITCH ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANSPOSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

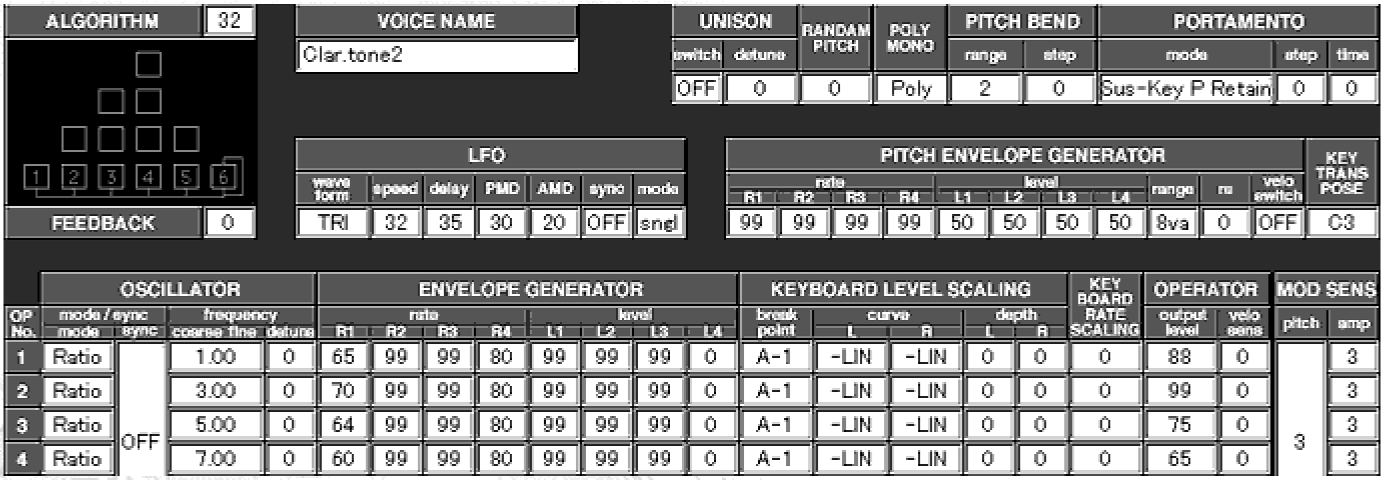
OP No.	OSCILLATOR mode	sync	frequency coarse fine	detune	EG rate R1	R2	R3	R4	EG level L1	L2	L3	L4	KLS break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio	OFF	1.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	88	0	3	0
2	Ratio		3.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
3	Ratio		5.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	80	0		0
4	Ratio		7.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	70	0		0

Although it's just an organ, the character of a clarinet is brought out quite well.

SAMPLE DATA DOWNLOAD **MIDI Data** clartone.zip (4.74kbyte) SAMPLE DATA DOWNLOAD **SoundVQ Data** clartone.vqf (28.8kbyte)

Load 64DxVoice4.MID first as cartridge file.

Now let's try changing the EG. Of course we'll also apply vibrato. On the EG we'll try a little trick to change the attack speed for each overtone. This will simulate the attack characteristics of a woodwind instrument.



ALGORITHM	FEEDBACK	VOICE NAME	UNISON switch	UNISON detune	RANDAM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
32	0	Clar.tone2	OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0

LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	32	35	30	20	OFF	sngl

PITCH EG rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANSPOSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

OP No.	mode	sync	coarse fine	detune	R1	R2	R3	R4	L1	L2	L3	L4	break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	output level	velo sens	MOD SENS pitch	MOD SENS amp
1	Ratio	OFF	1.00	0	65	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	88	0	3	3
2	Ratio		3.00	0	70	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		3
3	Ratio		5.00	0	64	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	75	0		3
4	Ratio		7.00	0	60	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	65	0		3

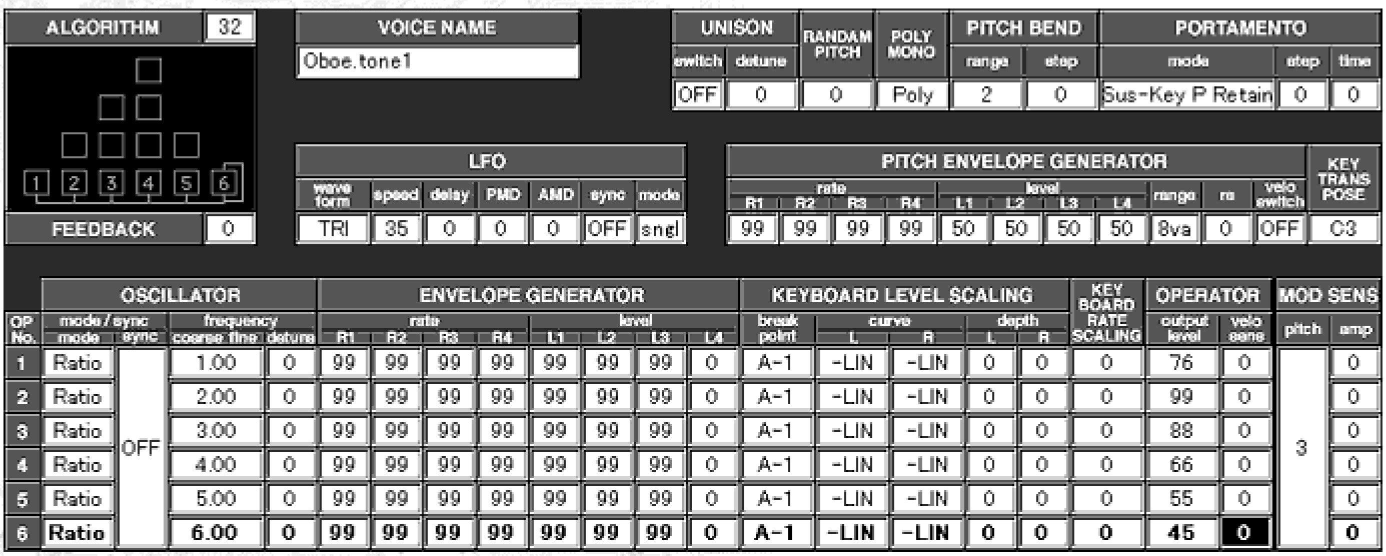
It's getting better and better.

MIDIData
claredit.zip
(4.74kbyte)

SoundVQData
claredit.vqf
(28.8kbyte)

Load 64DxVoice4.MID first as cartridge file.

Since we're on a roll, let's try an oboe as well



ALGORITHM	FEEDBACK	VOICE NAME	UNISON switch	UNISON detune	RANDAM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
32	0	Oboe.tone1	OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0

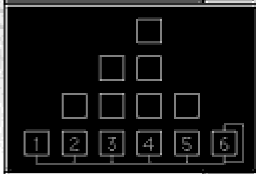
LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	35	0	0	0	OFF	sngl

PITCH EG rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANSPOSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

OP No.	mode	sync	coarse fine	detune	R1	R2	R3	R4	L1	L2	L3	L4	break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	output level	velo sens	MOD SENS pitch	MOD SENS amp
1	Ratio	OFF	1.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	76	0	3	0
2	Ratio		2.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
3	Ratio		3.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	88	0		0
4	Ratio		4.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	66	0		0
5	Ratio		5.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	55	0		0
6	Ratio		6.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	45	0		0

@
You already know how to do the next thing --- EG and vibrato.

ALGORITHM	VOICE NAME	UNISON switch	UNISON detune	RANDOM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
32	Oboe.tone2	OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0



FEEDBACK
0

LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	32	39	30	25	OFF	sngl

PITCH ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANSPOSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

OP No.	OSCILLATOR mode / sync mode	sync	frequency coarse fine	detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio	OFF	1.00	0	68	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	76	0	3	3
2	Ratio		2.00	0	66	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		3
3	Ratio		3.00	0	64	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	88	0		3
4	Ratio		4.00	0	62	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	66	0		3
5	Ratio		5.00	0	60	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	55	0		3
6	Ratio		6.00	0	58	99	99	80	99	99	99	0	A-1	-LIN	-LIN	0	0	0	45	0		3

SAMPLE DATA DOWNLOAD **MIDI Data**
oboedit.zip
(4.83kbyte)

SAMPLE DATA DOWNLOAD **SoundVQ Data**
oboedit.vqf
(40.6kbyte)

Load 64DxVoice4.MID first as cartridge file.

I've not done it here, but you can greatly increase the expressiveness of the sound by making appropriate velocity and rate scaling settings for each operator to increase the EG speeds as the pitch rises.

It's amazing that you can create sounds with such dynamic differences simply by lining up six overtones (or even fewer). If you had one or two hundred sine wave oscillators, you would be able to create some really wonderful sounds. On the other hand, having to make EG and other settings for all sixty or two hundred operators would be a daunting task. Instead of arranging large numbers of sine waves in parallel, the DX allows six operators to be stacked vertically in order to create sounds of comparable complexity.

We have finally come to our last session, in which we will create sounds using FM. And this is where we will (I hope) use the DX to re-create the ancestor of the synthesizer --- the pipe organ.

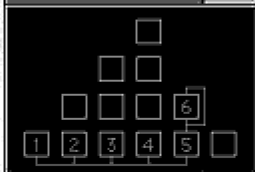
Seminar Top

Session 6: Final session:
Using FM to create a pipe organ

Here we are at our final session. This time I'm going to dispense with all the useless chatter, and step up the pace. Otherwise we won't have enough space to finish. Heh heh! (Although I suppose this also qualifies as useless chatter.)

Take a look at the following edit list.

ALGORITHM	FEEDBACK	VOICE NAME
31	7	FamousOrg.



UNISON switch	UNISON detune	RANDAM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0

LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	35	0	0	0	ON	sngl

PITCH EG rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANSPOSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

OP No.	OSCILLATOR mode	sync	frequency coarse fine	detune	EG rate R1	R2	R3	R4	level L1	L2	L3	L4	KLS break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio		0.500	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
2	Ratio		1.50	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
3	Ratio	ON	1.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0	3	0
4	**Ratio**		**3.00**	**0**	**99**	**40**	**99**	**99**	**99**	**0**	**0**	**0**	**A-1**	**-LIN**	**-LIN**	**0**	**0**	**0**	**99**	**0**		**0**
5	Fixed		1.000Hz	0	99	60	99	99	99	96	0	0	A-1	-LIN	-LIN	0	0	0	70	0		0
6	Fixed		2716Hz	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0

This sound simulates one of the most popular Hammond organ sounds: "lower three drawbars wide open plus 3rd harmonic percussion." This is a setting loved by Jimmy Smith, Keith Emerson, and many other jazz and rock organists.

While our previous example used algorithm 32, this one uses algorithm 31. Operator 6 is used not to directly produce sound, but to modulate operator 5. In other words, operator 6 in this algorithm is acting as a modulator. I will talk more about this later.

Now take a look at operator number 1. The list indicates OP No. The display is a bit hard to read, but you can see that it shows a Frequency of 0.50. This corresponds to an organ drawbar with a pitch of 16' --- one octave below the fundamental (8'). Next take a look at operator (hereafter written as "OP") two. This has a frequency of 1.5, which is a fifth above the fundamental. If the fundamental is C, this would be G. The display of the DX rises in integers as 1.00, 2.00, 3.00 etc., and these correspond to the natural harmonic series. I think that this is easier to understand than expressing the length of pipes in feet ('). When we say that integers correspond to the natural harmonic series, this means that numbers with a decimal point value would be overtones that are not found in the natural harmonic series. In order to give the sound greater depth, these two pitches were added to pipe organs as a sort of special option. The same is true for the organ drawbars: the second frequency of 1.50 (a pitch of a fifth above the fundamental) is actually located below (to the left of) the fundamental drawbar, rather than above it (to the right). I seem to recall that harmoniums (pump organs) and accordion also allow a frequency below the fundamental to be mixed in.

Next is OP3. This has a frequency of 1.00, and is the fundamental. OP4 is more interesting. It has a frequency of 3.0, indicating that it produces the third harmonic overtone, but this has been given a special role. You can switch an operator on/off by clicking the OP No., so try turning off all the other operators. Yes, it's a decay-type sound. This is called "percussion," and is unique to electric organs. It probably appeared first on the Hammond. I have no idea why percussion was added to an organ whose design intention was to simulate a pipe organ, but you can add two percussion tones: the 2nd and the 3rd harmonics. For some reason on actual Hammond organs, the percussion is monophonic, with single-triggered first-note priority. If you have two DX plug-in boards, you can set one to Mono mode for greater realism. (Obsessive personalities only!) Anyhow, the sound of the "lower three drawbars wide open plus 3rd harmonic percussion" is not an attempt to imitate any existing instrument. It is a sound that only the Hammond (and perhaps other electric organs) can produce. It's a simple sound, but has many loyal devotees.

Now let's get back to OP5 and 6 that we bypassed earlier.

Take a good look at the list. Notice that the MODE is Fixed. This means that regardless of the note that was played on the keyboard, the operator will produce a fixed frequency. Next, look at the FEED BACK item in the algorithm box. This shows "7." Feedback is a function that lets an operator modulate itself continually.

This means that the operators own signal is sent back to the same operator, and that this modulated waveform is then sent back to itself, ad infinitum. The connection is similar to the feedback circuit found in delay or flanger devices. In these devices, raising the feedback level excessively will turn the sound into total noise. Likewise, raising the FM feedback to the maximum will produce a hissing sound known as "white noise."

The fact that operators which are able to produce only a sine wave can be used to create white noise (the polar opposite of a sine wave) makes me wonder whether the process of using a FM tone generator to create sounds is essentially the search for musically expressive tones that lie between pure sound (sine waves) and noise. (Excuse my moment of poetic self-indulgence!)

Getting back to our subject, you can see from the output level of OP6 that this feedback is being used to create white noise. The EG for the operator (carrier) that produces this white noise a sound is cut extremely short. You may wonder why on earth anyone would bother to make such a short sound. Recall that in our discussion of Hammond organs, we mentioned the fact that a brief "click" noise occurs when a note is played. In our example, this is being simulated by OP5 and OP6. To think that the first sound you created using FM was a click noise --- what an extravagance!

Speaking of extravagance, remember that we mentioned rotary speakers? The XG effects include a rotary speaker as well as an amp simulator. You've simply got to take advantage of these. Use the amp to slightly distort the sound and use the Variation effect to send the sound through the rotary speaker, and I think you'll be impressed by the result.

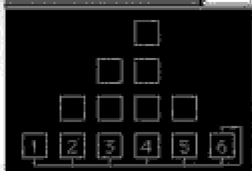
SAMPLE DATA DOWNLOAD **MIDI Data** jazorg.zip (1.5kbyte)

SAMPLE DATA DOWNLOAD **SoundVQ Data** jazorg.vqf (100kbyte)

Load 64DxVoice4.MID first as cartridge file.

Next is also an organ, but a cheap, small, and portable organ. This is a type which in the past was used mainly by bands that didn't have a lot of money. The edit list is given below. Raise or lower the output levels to modify the sound. Vibrato is applied, but you can turn this off if desired. Taking off the vibrato will enhance the cheapness that is such an important part of the character of this organ(!).

ALGORITHM	FEEDBACK	VOICE NAME
32	0	PortaOrgan



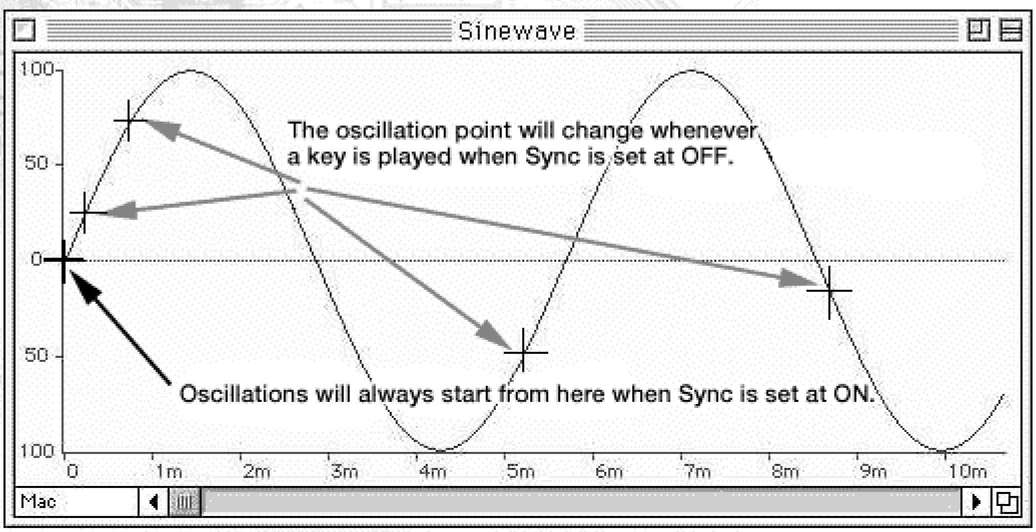
UNISON switch	UNISON detune	RANDAM PITCH	POLY MONO	PITCH BEND range	PITCH BEND step	PORTAMENTO mode	PORTAMENTO step	PORTAMENTO time
OFF	0	0	Poly	2	0	Sus-Key P Retain	0	0

LFO wave form	speed	delay	PMD	AMD	sync	mode
TRI	35	0	16	0	OFF	sngl

PITCH ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	range	rs	velo switch	KEY TRANS POSE
99	99	99	99	50	50	50	50	8va	0	OFF	C3

OP No.	OSCILLATOR mode/sync mode	sync	frequency coarse fine	detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio	ON	0.500	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0	3	0
2	Ratio		1.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
3	Ratio		2.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	99	0		0
4	Ratio		4.00	0	99	99	99	99	99	99	99	0	A-1	-LIN	-LIN	0	0	0	90	0		0
5	**Ratio**		**8.00**	**0**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**0**	**A-1**	**-LIN**	**-LIN**	**0**	**0**	**0**	**90**	**0**		**0**

Oh, yes. In the oscillator box of the list, you will see that the MODE/SYNC item is set to SYNC On by default. This has nothing whatever to do with the OSC Sync function found on analog synthesizers, and specifies whether or note the waveform will start from the beginning at key-on. In the case of our example, this setting makes little difference, but for some sounds it can be a very important parameter. I suppose that for an electric organ this would logically be Off. The following diagram should give you an image of what this parameter does --- or perhaps it will just confuse you more(!)



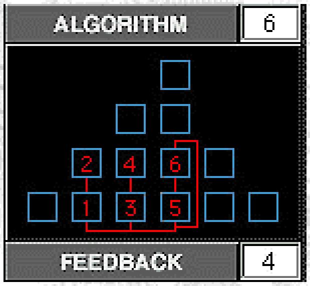
[MIDI Data](portorg.zip)
portorg.zip
[(978byte)](portorg.zip)

[SoundVQ Data](portorg.vqf)
portorg.vqf
[(72kbyte)](portorg.vqf)

Load 64DxVoice4.MID first as cartridge file.

And now we come to the main event for today --- creating a pipe organ sound.

First, either recall INIT VOICE or create it. We are going to use algorithm 6. (See the following diagram.)



Along with algorithm 5, this is one of the standard algorithms for FM editing, and is ideal for beginners. It provides three groups each of which has a single carrier and single modulator. In other words, this algorithm provides both FM and additive synthesis.

I've listed all of the parameters below, but when you create the sound, do so one operator at a time, working horizontally in the chart. Don't just input an entire column at a time. You can if you want to, but you won't understand the role of each operator.

OP No.	OSCILLATOR mode / sync: mode	sync	frequency: coarse fine	detune	ENVELOPE GENERATOR rate: R1	R2	R3	R4	level: L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve: L	R	depth: L	R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio	ON	1.00	-1	55	99	99	54	99	99	99	0	A-1	+LIN	-LIN	0	0	2	99	0	3	0
2	Ratio		0.500	0	99	99	99	58	99	99	99	0	B2	+LIN	-LIN	23	0	2	87	0		0
3	Ratio		2.00	0	60	99	99	54	99	99	99	0	A-1	+LIN	-LIN	0	0	3	99	0		0
4	**Ratio**		**3.00**	**0**	**93**	**50**	**97**	**99**	**99**	**93**	**93**	**89**	**B3**	**+LIN**	**-LIN**	**20**	**0**	**3**	**70**	**0**		**0**
5	Ratio		4.00	2	64	99	99	73	99	95	95	0	C3	+LIN	-LIN	6	0	2	94	0		0
6	Ratio		8.00	2	99	99	99	65	99	90	90	85	C3	+LIN	-LIN	0	10	2	76	0		0

@
Set the output level of OP2, and this will complete the sound of the first group. This group corresponds to the low-range stops. The oscillator detune parameter is used to produce a bit of modulation. This is because on a real pipe organ, it is rarely the case that all of the pipes are producing a precisely accurate pitch. I suppose it's a matter of taste.

Notice that there is a "chiff" sound when you release a note. This simulates the sound that occurs when the air being sent to the pipe is cut off. On large pipes of a real pipe organ, a similar sound is heard when a note is released. This is very important in conveying the character of a pipe organ.

In this example we finally see the appearance of something called Keyboard Level Scaling.

This is a function that adjusts the level of an operator according to the keyboard location in which the note was played. On this sound, I have created an upward curve starting at B2 in order to slightly brighten the lower range of the sound.

The next group is the mid-range. Try muting operator 1 so that you hear only this group, and you will hear a somewhat metallic sound that you might not associate with pipe organs. However in the same way that combining two or more sine waves results in a new sound, this group will play an important role in the finished sound when it is combined with the other groups.

The final group of course produces the high range. The EG settings are slightly different than those of the other parts. This is to produce the "squeak" that is heard when air is sent to short pipes. This too, is a matter of providing details to build up the desired image.

Finally, we will use level scaling to lower the modulator level in the high register. This will prevent aliasing noise from occurring when high notes are played. A bit of feedback is applied to add some brilliance across a wide range.

It's finished! If we had tried to create the same sound using sine wave additive synthesis, I have no idea how many oscillators would have been required. I suppose that twenty or thirty ought to have done the job. In particular, it would have taken a lot of resources to produce the "details" of the sound. However on the DX, all of this was taken care of by just six operators.

Now it's up to you to try modifying the frequency or level of the operators. At first it may be difficult for you to make the sound change in the way you wish, but as you become accustomed to it, you may become addicted to FM sound editing. Once upon a time I myself was immersed in the DX day and night, to the extent that my girlfriend nearly left me on account of it.

SAMPLE DATA DOWNLOAD **MIDIData**
pipeorg.zip
(5.41kbyte)

SAMPLE DATA DOWNLOAD **SoundVQData**
pipeorg.vqf
(129kbyte)

Load 64DxVoice4.MID first as cartridge file.

I'll leave you with some MIDI data that includes effect settings, and draw this series to a close.
Bye for now!

--- The end ---

1th | 2th | 3th | 4th | 5th

--> Seminar Top

Session 7
The Great DX sound --- Electric Piano! (part 1)

The best DX sound of all time simply has to be the electric piano.
The DX's electric piano sound was what placed it firmly in its place as the world's most popular synth. After all, all previous synthesizers had struggled futilely to make a good electric piano sound.

While the DX series was still being sold, other manufacturers even included the DX electric piano sound as a preset waveform in their sampling keyboards --- it may not have been alive, but it was a living legend.
There are several reasons why this sound became so popular.

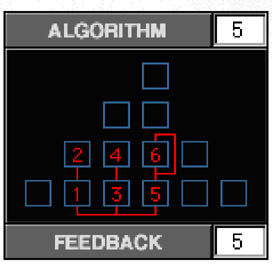
1. It was recognizable by anyone as an electric piano sound.
2. It was more brilliant-sounding than a real electric piano. (At the time, the Fender Rhodes was the standard electric piano. Details will follow.)
3. It was highly responsive to touch (velocity).
4. The DX was cheaper and lighter than a real electric piano (important!).
5. It could be edited to create numerous variations.

I think this summarizes its popularity.
First let me explain a little about the Fender Rhodes electric piano (that's "electric," not "electronic"!).
This electric piano did not use electronic circuitry to synthesize sound. Instead, playing a note on the keyboard caused a hammer to strike a metal bar, and the vibrations of this bar were sensed by a pickup.
Although it was called an electric piano, it produced a completely original tone that was totally different than that of an acoustic piano. The phrase "strike a metal bar" might give you the impression of a toy piano, but it's actually a completely different sound.
There was another electric piano called a Wurlitzer, which produced sound by striking a thin metal reed instead of a bar. The Carpenters were one of the bands who used this. Both of these sounds are probably quite familiar to you on sampling keyboards, so I won't go into detail about their sounds.
In their stock condition, these electric pianos produced a fairly mild tone. In particular for the Fender Rhodes, there were even companies that would "hot-rod" a Rhodes to make the sound brighter, and many owners of these pianos put money into their instruments to adjust the tone to their own taste.
Into this scene that the DX7 electric piano sound made its appearance. The sound was actually not totally identical with the Fender Rhodes or Wurlitzer electric pianos. Although FM tone generation is able to produce more realistic sounds than the analog oscillator synthesizers that came before it, it cannot compete with today's sampling. (In those days, samplers with audio quality and functionality that would be laughable today sold for eye-popping prices. Still, it's funny that there are some aficionados who insist that they like this cheap sound.)
Anyway, the DX electric piano sound took the world by

storm, and in particular played a big role in American music scenes such as AOR and fusion.
When I was in LA once in the 80's at a certain recording session, I had intended to use a Fender Rhodes, but when I started playing I decided that the DX was more to my liking, and switched instruments at the session. Once you're used to the sound of the DX, you become annoyed with the various mechanical noises that occur when you play the Rhodes (such as the sound of the damper pedal being pressed or released, or the noise of the hammers). Of course, those were my feelings at that time. Today, I think these noises are something special. In fact for recently-created FM sounds (well, already a few years ago), it is the fashion to include as many of these noise elements as possible.

Let's get on with our sound editing.
Back in those days, I was quite involved with creating the preset sounds for instruments such as the DX7 and DX1, and spend a lot of time creating simulations of existing instruments or sounds of nature. Probably 80 percent of my work was of this type.
Still, it wasn't that we had those instruments in front of us so that we could compare the sound. The person doing the voicing simply re-created the sound according to his own impression of it.
So instead of setting up a Fender Rhodes beside the DX and trying to simulate its sound, I would be listening only to the sound of the DX, and keep working until I got a result that I liked. Then it was on to attempt the next variation..

I'd like to return to the feeling of those days, and use the DX simulator to once again create an electric piano. (Though we sometimes talk about "returning to the beginner's mind," it's true that the principles of sound editing have long since soaked into every corner of my thinking.)
We'll use algorithm number 5, which we used to create the pipe organ in the previous series. This algorithm is one that's frequently used at the beginning levels of sound editing.B



At the bottom of the operator diagram is a field marked FEEDBACK with a value of 5 displayed. This indicates a feedback level of 5 (in a range of 0 to 7).
I explained feedback in the previous series. In many ways, it's an extremely important parameter. Without feedback, I think that the DX would be only half as interesting.

After you have selected the algorithm, our next step is to create the basic sound that will be the core of the electric piano tone.
To do this, we will use operators 1 and 2.

OP No.	OSCILLATOR mode / sync: mode	sync	frequency: coarse fine	detune	ENVELOPE GENERATOR rate: R1	R2	R3	R4	level: L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve: L	R	depth: L	R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio		1.00	2	99	65	28	58	99	92	0	0	A-1	-LIN	-LIN	0	0	3	99	2		0
2	**Ratio**		**1.00**	**0**	**99**	**99**	**24**	**70**	**99**	**99**	**0**	**99**	**C2**	**-LIN**	**-LIN**	**0**	**23**	**0**	**91**	**4**		**0**

The carrier and modulator both have a pitch of 1.00; i.e., modulation is going to occur between the same frequencies. This will produce a strong fundamental and second overtone one octave above it. As modulation increases, even higher overtones will appear.
In addition to creating electric pianos, this combination is also used for flutes. When combined with feedback, it can be the starting point for brass or strings. It really is the most basic of the basic combinations.
When a note is played softly on an electric piano, you get a soft sound that is close to a sine wave. When a note is played strongly, you get a more metallic and nasal sound. To create this characteristic, we adjust the velocity sensitivity.
Take a look at Detune. For the carrier, it has a value of 2. This slightly skews the pitch of the carrier and modulator to create a soft modulation. Ruler-straight sounds are not natural, and don't feel very good either.

Next is the envelope. This is very important for decay-type sounds such as the electric piano. The decay speed of the volume and tone are particularly important. Carefully compare operators 1 and 2. As the volume decreases, in other words as the decay progresses, the modulation will lessen, causing the sound to become closer to a sine wave. This is a very typical setting for decay-type sounds such as electric piano.
Operator 1 has a decay that is initially rapid, and then causes the sound to decay gradually. This is really up to your own taste.
Envelopes on the DX are created by adjusting four rates and four levels. Level 1 (L1) is the attack level, and L2 is the decay level. However if L1 is lower than L2, L2 will be the second attack level. L3 is the sustain level, and if this is higher than zero, the sound will continue to be heard as long as you continue pressing the note or the sustain pedal. L4 is the release level (initial level). This means that if L4 is set to 99 on a carrier operator, the sound will persist until you switch voices or turn off the power, so be careful. The reason that I added "initial level" in parentheses is that if, for example, this is set to 60, the level will remain at 60, so that at the next key-on, the level will move to the L1 value starting from 60 without first going to zero.
Trying to explain this in writing tends to make matters more confusing, but you don't need to worry about this too much, since you won't be inputting an L4 value for a carrier operator unless you are creating a very special type of sound. In nearly all cases, an L4 value will be for a modulator operator.
The rate settings indicate the speed at which movement occurs between levels. A higher value means faster movement, and a lower value means slower movement.

Take a look at the envelope for operator 2. Hey, there's a value for L4, and it's 99!

Notice the rate settings. R4 is the rate of movement from L3 to L4. This is 70. Fairly fast. The reason for this setting is that on an electric piano, releasing a key causes the metal sound-producing bar to be muted. This too, is a matter of taste, and when the sound is in a context of other instruments, it's pretty hard to tell the difference. Still, this sort of subtlety does affect the playing feel, so don't ignore it.

We're finished with the envelope, and next is Keyboard Level Scaling. This regulates the output of the operator across the keyboard. Depending on the instrument, volume and tone will change as you play across the instrument's range of pitches. The keyboard level scaling parameters control this type of change. In the case of an electric piano, the tone gets more mild as you play higher; i.e., it becomes more like a sine wave. (This is true of most instruments.) That's why the level scaling has been set to decrease the modulation as the key range rises. If we didn't do this, the tone would be too harsh in the upper register.

This completes our basic electric piano sound.
On FM keyboards prior to the DX7, or on the inexpensive FM synthesizers that appeared after the DX7, the sound was sometimes left at this point. But since we have four more operators to use, let's spend our next session taking advantage of these to get more deeply into the sound-creating process.

Seminar Top

Session 8

The Great DX sound --- Electric Piano! (part 2)

In our previous session we used two operators to create a basic electric piano sound.
In this session I'm going to explain the process of adding things to make the sound more like an electric piano, and how to create different variations.

First we will use the pair of operators 5 and 6. The reason that we aren't going to start with the pair 3 and 4 is that the sound we're going to create requires feedback on the modulator. (This is the parameter shown below the algorithm display box.)
Take a look at the following edit list. It's not too different from the basic sound that we created in our last session.

	OSCILLATOR				ENVELOPE GENERATOR								KEYBOARD LEVEL SCALING					KEY BOARD RATE SCALING	OPERATOR		MOD SENS	
	mode / sync		frequency		rate				level				break point	curve		depth			output level	velo sens	pitch	amp
OP No.	mode	sync	coarse fine	detune	R1	R2	R3	R4	L1	L2	L3	L4		L	R	L	R					
5	Ratio		1.00	0	99	65	28	58	99	92	0	0	A-1	-LIN	-LIN	0	0	3	99	2		0
6	Ratio		1.00	-2	99	99	24	70	99	99	0	99	C2	-LIN	-LIN	0	22	0	89	3		0

There are only slight differences in the modulator detune, level key scaling, output level, and velocity sensitivity.
Still, try playing a low note strongly. (If you're not using a keyboard, try transmitting a note with a strong velocity.) You'll get a "sprongy" tone that sounds a little bit "cracked."
As a test, try switching operators 1 and 5 on/off as you continue playing the same key, and listen to the difference.
Actually even on a Fender Rhodes, the low range sounded like the tone we just created, with a cracked character. ("Cracked" might not be a very precise adjective, but I can't think of a more appropriate word.)
In the low range of an electric piano, the metal bars (plates) that substitute for strings are longer, so that when you strike a key strongly, the metal bar (plate) vibrates quite strongly, and this vibration will affect the tone. I used feedback to simulate this phenomenon, producing a similar sound.
Of course the purpose of this is merely to produce the impression of an electric piano, and not the actual sound of an electric piano, so don't go trying to sample this sound and compare it with the real thing. It's not a useful thing to be doing anyway(!)

Since I've put the feedback level at 5, I expect you may be wondering why I didn't use a setting of 6 or 7? Well, as an experiment, let's try raising the feedback level. At a setting of 6, the tone is somehow more fine-grained. Not quite like an electric piano. With a setting of 7, the modulation is excessive, and we would have to lower the output level to prevent the sound from becoming unpleasant. Now let's try lowering it to 4. At this setting, we've lost the "sprongy" character.
Why was the setting of 5 just right? I don't know of any logic here. I just know from experience that in such cases, a feedback setting of 5 works well. It's like a cook knowing how much salt or spices to add to the food.

You may have another question at this point.
"If we can simulate the character of an electric piano using only

operators 5 and 6, why do we need the pair 1 and 2? Alternatively, couldn't we use the pair 1 and 2 to produce some other impression?" You are absolutely right. There are examples of using the 1/2 pair or the 5/6 pair for some other purpose. (Our next session will introduce such an example.) However, that does not say that the other pair is meaningless. Using two pairs (in this case, pairs 1/2 and 5/6) to produce a similar sound will give the sound more body. The sound will be thicker and more powerful. It's a simple thing, but often this is more important than adding detail to the sound .
And now we come to the last pair, operators 3 and 4. This pair actually provides that special "something" that made the DX electric piano a world standard
Look at the edit list.

OP No.	OSCILLATOR mode	sync	frequency coarse fine	detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
3	Ratio	ON	1.00	0	99	99	35	99	99	99	0	0	A-1	-LIN	-LIN	0	0	0	99	1	3	0
4	Ratio		9.00	0	99	52	44	70	99	83	0	0	A1	-LIN	-LIN	0	29	0	82	3		0

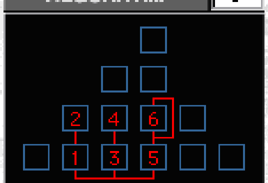
[MIDI Data](#)
[ep_sample1.zip (6kbyte)](#)

[SoundVQ Data](#)
[ep_sample1.vqf (100kbyte)](#)

Load 64DxVoice2.MID first as cartridge file.

Next let's create a variation based on this sound. This is going to be very easy.
We begin with our customary edit list.

ALGORITHM 5



FEEDBACK 4

OP No.	OSCILLATOR mode	sync	frequency coarse fine	detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens
1	Ratio	ON	1.00	7	99	65	28	58	99	92	0	0	A-1	-LIN	-LIN	0	0	3	99	3
2	**Ratio**		**1.00**	**4**	**99**	**99**	**24**	**70**	**99**	**99**	**0**	**99**	**C2**	**-LIN**	**-LIN**	**0**	**22**	**0**	**90**	**4**
3	Ratio		1.00	0	99	99	35	99	99	99	0	0	A-1	-LIN	-LIN	0	0	3	99	2
4	Ratio		13.40	0	99	47	33	70	99	85	0	0	A1	-LIN	-LIN	0	29	2	79	4
5	Ratio		1.00	-4	99	65	28	58	99	92	0	0	A-1	-LIN	-LIN	0	0	3	99	3
6	Ratio		1.00	-7	99	99	24	70	99	99	0	99	C2	-LIN	-LIN	0	22	0	94	4

The colored areas in the chart are the portions that were changed from the sound we created earlier.
The light green areas are not very important. It would probably make little difference if we had carried these over from the previous sound, although the effect of velocity on expression is a bit different.
Setting the feedback to 4 is just an example of how a different feedback level produces a different impression. Because the feedback level is low, I've raised the output level, but you may make your own choice.

There's actually something that I skipped in the previous explanation due to lack of space, so I'd like

to give some details in this session. It's about the areas colored orange. Keyboard Rate Scaling controls the rate (speed) of the EG according to keyboard range. On recent instruments, this can be set rather precisely, but on the DX you are limited to specifying how the decay of decay-type sounds is speeded up. There's no break point to set. (Some of you may feel let down!)
However as is the case for other parameters, the DX does let you set this independently for each operator, so that you can actually get a lot of potential out of this apparently simple method. In particular when creating delicate metallic decay-type sound that have a lot of high-frequency components, skillful use of level scaling and rate scaling is the core of the sound-editing process.

Now let's talk about the parameters that have the most important role in creating variations of the sound.
These are the areas colored pink in the chart.
I'll begin my explanation from the left. We'll start with the oscillator frequency. This time, the oscillator frequency is 13.40. The decimal portion .40 is the key here. If it were 13.00, it would fit right into the overtone series, and we would get a smooth sound. However .40 is a deep detuning that just doesn't fit in. If we attempted this at a closer frequency ratio, the sound would just be a mess. But with a ratio that's this distant, we get a realistic metallic character. I'd like you to try changing this to find your favorite settings, but you will find it more convenient to go back to the panel window and use the data entry slider instead of using the edit list window.
Moving to the right, let's look at the EG. By moving R2, 3 and L2, you can change the way in which the metallic sound decays. Try various possibilities.
Finally we have the output level. By now this should need no explanation.
Compared with other methods of tone generation, it is popularly though that FM uses a lot of parameters and is complicated, but the elements that determine the character of the sound are actually quite few.
And so next time we're going to try some even bolder variations.

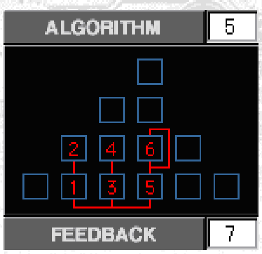
Seminar Top

Session 9
Leaving the bounds of the Electric Piano!

This time we're going to start with the electric piano, but create some sounds that break out of the usual bounds of the electric piano. Even if we stayed with electric pianos, there are many other ways to create sounds and plenty of things that I'd like to explain. But this would mean that we would be getting in deeper and deeper, and might not be able to get back out again. Plus, it's boring to be always making the same type of sounds.
We're going to create two different sounds. One is an electric piano layered with another sound. The other is an example of how I was intending to create an electric piano, but other elements got added so that it no longer fit into the category of an electric piano.
The first is interesting, and has a good feel to it. In particular, it's quite effective when combined with chorus or reverb.
The second represents a step up in sound-editing technique. This might be one step before advancing to intermediate level.

Without further delay, here's the first sound; electric piano + velocity brass.
In recent years, it is no longer unusual for a synthesizer to be able to create two different sounds within a single voice. (Some synthesizers have a layering structure within the voice.) But when the DX appeared, this was something special.
Well, there's no point in being nostalgic, so let's press on and create some good sounds.

Here's our familiar edit list. For now, we'll mute operator 1, and focus on the pair of operators 5 and 6.



OP No.	OSCILLATOR mode / sync: mode	sync	frequency: coarse	fine	detune	ENVELOPE GENERATOR rate: R1	R2	R3	R4	level: L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve: L	curve: R	depth: L	depth: R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
5	Ratio		1.00		0	99	65	28	70	99	99	99	0	A-1	-LIN	-LIN	0	0	3	99	7		0
6	Ratio		1.00		-2	99	65	29	69	70	99	93	0	C2	-LIN	-LIN	0	0	2	82	2		0

This is algorithm number 5, and at first glance it doesn't look much different than the electric piano. But take a good look at the EG settings, and you'll see that L3 is set to 99 for operator 5, and L3 is set to 93 for operator 6. This means that it's not a decaying sound, but a sustaining sound. (I suppose that no explanation would be necessary if you just input the data and played the sound.)

Now look at the algorithm box. The feedback level is 7. When creating the click noise of an organ in the previous session, we had set the feedback level to 7. This time, however, it's not for the purpose of creating noise. When the feedback level is 7, the carrier and modulator are at a frequency ratio of 1 to 1, and the output level is an appropriate amount, the result will be a sound quite similar to the sawtooth wave of an analog synth. This is the pattern that is almost always used when creating synth brass or strings, so it's worth remembering.

Let's get back to the EG. The modulator detune is -2, but this could just as well be 0, or would be little different if it were +2.
Notice L1 and L2 of the modulator. L1 is a lower value than L2.
Now look at the rate. R1 is 99, the maximum, and R2 is a medium speed. This means that the movement from L1 to L2 will give us that "bwah" attack that is so typical of brass. So why didn't we do this for R1 and L1? The reason is that if we used R1 and L1 to create the brass attack, this would mean that the level would start out from a level of zero. A modulator level of zero means that the output sound would begin as a sine wave. This would be OK for a brass sound that begins gently, like a horn. However since we are creating a sound that needs to go well with the electric piano, we want a sharp attack. That's why we set the rate to the maximum, and set L1 to a medium value in order to simulate a situation in which we are starting with a moderately filtered sawtooth wave.

Next look at velocity sensitivity, located at the far right. This has a value of 7 (maximum) for operator 5. The reason for this is that we want the brass to be added to the electric piano for strongly-played notes in order to add dynamic emphasis.

Now we're going to use the remaining four operators to create the electric piano sound. Let's look at all of the parameters together.

	OSCILLATOR				ENVELOPE GENERATOR								KEYBOARD LEVEL SCALING					KEY BOARD RATE SCALING	OPERATOR		MOD SENS	
OP No.	mode / sync mode	sync	frequency coarse fine	detune	rate R1	R2	R3	R4	level L1	L2	L3	L4	break point	curve L	R	depth L	R		output level	velo sens	pitch	amp
1	Ratio	ON	1.00	2	99	65	28	58	99	92	0	0	A-1	-LIN	-LIN	0	0	3	99	2	3	0
2	Ratio		1.00	0	99	99	24	70	99	99	0	99	C2	-LIN	-LIN	0	23	0	91	4		0
3	Ratio		1.00	0	99	99	35	70	99	99	0	0	A-1	-LIN	-LIN	0	0	0	98	1		0
4	Ratio		11.00	0	99	52	44	58	99	83	0	83	A1	-LIN	-LIN	0	29	0	82	3		0

No particular explanation is needed here.
This is a sound that should be played in realtime, rather than being used for step input.

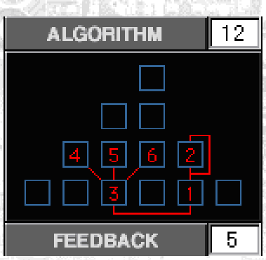
MIDI Data
ep_sample2.zip
(1.5kbyte)

SoundVQ Data
ep_sample2.vqf
(100kbyte)

Load 64DxVoice.MID first as cartridge file.

Let's move on to the last sound in our series of electric piano sounds. As I wrote at the beginning, there have always been numerous variations of DX electric piano sound. Some of these are far removed from the sounds of existing electric pianos, yet are still attractive sounds that are undeniably electric pianos. If readers of this lecture series request it, I may be given the opportunity to discuss such sounds, so please send in your opinions and requests.

Returning to our topic, let's look at all of the parameters at once.



OP No.	OSCILLATOR mode / sync: mode	sync	frequency coarse fine	detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens
1	Ratio	ON	1.00	0	99	99	28	60	99	99	0	0	A-1	-LIN	-LIN	0	0	2	99	2
2	Ratio		1.00	0	99	24	99	65	99	0	0	0	G1	-LIN	-LIN	0	26	0	88	3
3	**Ratio**		**2.00**	**0**	**99**	**99**	**28**	**60**	**99**	**99**	**0**	**0**	**A-1**	**-LIN**	**-LIN**	**0**	**0**	**2**	**99**	**3**
4	Ratio		5.00	0	99	70	30	66	99	60	0	90	G#2	-LIN	-LIN	0	22	0	86	6
5	Ratio		19.57	0	99	42	34	97	99	92	0	72	F2	-LIN	-LIN	0	99	0	69	4
6	Ratio		7.44	0	99	42	34	97	99	92	0	63	F3	-LIN	-LIN	0	0	0	70	4

Ah, finally! We get to use an algorithm other than number 5.
Maybe this is a sign of your progress. (Still, if we move at this pace, it will take quite a while to advance to the most complicated algorithms.)
I'm going to explain the structure of this sound.
First comes the operator pair1 and 2.
This is clearly the basic sound of the electric piano. No explanation necessary.
Now look at operator 3. Hey, the carrier pitch is 2.00. And operator 4 is 5.00. Just what was this guy (OK, so it's me) trying to do?
Before I explain, notice operators 4 and 5. Both are set to produce metallic sound, but why do we use two operators for this?
In order to understand this, let's leave this sound aside for a while, and listen once again to the two electric pianos we created in the preceding sessions (cartridge voices 1 and 2).
In particular in the high range, the metallic component does not sound like a "clang" or "cling," but produces a more delicate impression. This feeling is stronger for the second sound, in which the modulator frequency is higher. This is not because level scaling is used to restrain the output level, but because the metallic component ascends in pitch proportionally to the overall pitch.
Although this is natural for the DX, it's something that could not occur on an electric piano, since the metallic component obviously does not ascend in parallel with the keyboard pitch. Since the length of the metal bars does change, there is a certain amount of change in comparison to the lower range of the keyboard, but the change is not proportional. As the pitch rises, the frequency difference between the pitch and the metallic portion becomes narrower. This may be easier to understand if we compare the keyboard pitch to ascending a stairs at a 45 degree angle while the metallic component is rising at a 20 degree angle.

The person who created this sound (myself) evidently wanted to use operators 5 and 6 to create this. In other words, the idea was to use operator 5 for the metallic component of the lower range, and operator 6 for the metallic component of the higher range. As evidence of this, look at the level scaling for operator 5. It drastically cuts off the region above F2.
At this point, I was still trying to create an electric piano.
That's right. In the beginning when I was creating an electric piano, the frequency of operator 3 was still 1.00, and the mode of operator 2 was not Ratio but Fixed in order to simulate the "thunk" of the hammer action.
But as I was using operators 5 and 6 to play around with the metallic component, an idea or an image of a different sound occurred to me, and forgetting the electric piano, I slipped into a different direction.
It was on a whim that I set operator 2 to 2.00. This produced a feeling that I liked, so I just decided to use it. There was no logic behind this.
But with operator 3 still in the Fixed mode to produce the "thunk,"

it still had the flavor of an electric piano.
So in order to distance the sound from an electric piano, I changed from Fixed back to Ratio, and tried various frequency settings before settling on 5.00.

In addition, the metallic component of this sound lingers in a unique way after the key is released, and this is due to the EG settings of operators 4, 5, and 6. Look at L4. It's set to a fairly high value. R4 is also quite high. I also use this technique in order to create a realistic electric piano, but not to this extreme. I suppose that as my intention drifted from an electric piano into some other direction, the settings also drifted.

If we had to categorize this sound, it would not be impossible to call it an electric piano, but it's become really quite distant as a sound.
Not limited to the DX, but when creating a sound on any synthesizer, it frequently occurs that a different image occurs to you during the process, or you end up going in a different direction than you intended. And without this, it's hard to come up with new sounds. Having shown you an example in this session of how such trial and error can be a way to create fresh new sounds, I'd like to conclude this session.

SAMPLE DATA DOWNLOAD **MIDI Data** ep_sample3.zip (1.5kbyte)

VQ SAMPLE DATA DOWNLOAD **SoundVQ Data** ep_sample3.vqf (100kbyte)

Load 64DxVoice.MID first as cartridge file.

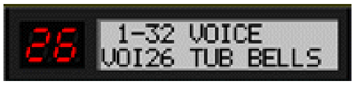
Seminar Top

Part 10
The classic DX sound --- The Bell! (Part 1)

It's a bell. It's metal. Things were getting pretty metallic toward the end of our sessions on the electric piano, but now we'll get into the sounds of some real metallic objects.

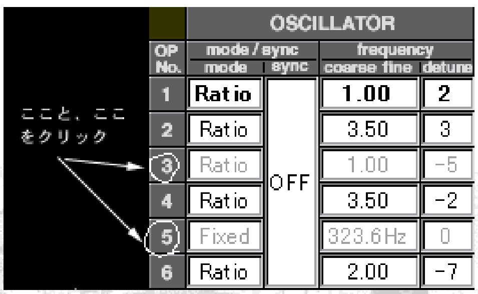
If we were to list the of sounds that are most typical of the DX, or that brought the DX its world-wide acclaim and were used in numerous genres of music, we would begin with the electric piano, but would also include bell, electric bass, clavinets, organ, and harpsichord. As far as strings or brass, those were the days in which classic analog synthesizers like the Prophet and Oberheim were still filling these needs. The DX also played important roles for realistic brass instrument sounds such as horn and trumpet.
And so we can see a common thread that links the greatest sounds of the DX, and that is "metal." FM tone generation is able to produce extremely realistic simulations of metal bars, metal plates, or metal strings. That's why metallic sounds such as vibraphone, kalimba, toy piano, steel drum, or gamelan were virtually the monopoly of the DX. In other areas, hard woody sounds (although I don't know of any soft instruments made of wood) were another strong point, and these included xylophone, wood block, slit drum, and log drum. But with this sort of sound, close listening did betray a slightly plastic-y feel.
Anyhow, the DX was superbly good at producing the sounds of metal being struck or plucked. So for this session and the next, we're going to concentrate on metallic sounds, and in particular, on bell sounds.

You may have noticed that we have not discussed any of the preset sounds of the DX7 in this series. There are various reasons for this, but one reason is that in order for you to understand FM tone generation, using existing presets would take us on an unnecessarily circuitous route. Another reason is that most of the presets were arrived at by a process of trial and error, and that here and there are meaningless parameter values. We would have to explain each of these, which would lead us far afield. Furthermore, FM voicing technology made great advances after the presets were created, resulting in a significant gap. In spite of this, the presets do contain one completed form of the bell sound, so we're going to use it as an example here.



The parameters are clear enough.
Some of you reading this may not have the XG Works program, so I've included a thumbnail below. Click this if you want to take a look, and all parameters will be displayed in a larger form.

"TUB BELLS" of course stands for tubular bells. This instrument consists of a frame from which hang numerous metal pipes, which are played by striking them with a hammer. I'm sure you've heard this instrument as some point in your lives. Real tubular bells do not sound as bright as this DX sound. Still, this completely transcended any category of sounds that previous synthesizers could produce, and was so realistic a "bell" sound that nobody complained.
As shown in the figure below, click on the operator number to mute operators 3 and 5 so that you can hear the sound that is produced only by operators 1 and 2.



OP No.	OSCILLATOR mode / sync: mode	sync	frequency: coarse fine	detune
1	Ratio	OFF	1.00	2
2	Ratio		3.50	3
3	Ratio		1.00	-5
4	Ratio		3.50	-2
5	Fixed		323.6Hz	0
6	Ratio		2.00	-7

What do you think? This is enough to complete 90% of the sound.
Operators 3 and 4 create the same sound as operators 1 and 2, but merely skew the pitch to create spaciousness, and do not fundamentally affect the tone. The operator stack 5 and 6 simulates the noise of the hammer strike, and is sort of a decoration.

Look at the OSC frequency ratio. The carrier is 1.00 and the modulator is 3.50. This is like a formula, and if you apply modulation at this ratio, you will automatically get the sound of a bell. 3.50 means the 3.5th overtone relative to the fundamental as 1. Of course, this overtone is not found in the natural harmonic series. In spite of this, bells have the strange property of possessing a strong tone that is a fifth above the fundamental. As a child when I used to hear bells strike the hour, I used to wonder about this. Although the bells were sounding C - E - D - G, they seemed at the same time to also be sounding G - B - A - D, almost as though they were a chord.
Since the frequency ratio 1:3.5 creates this state, we hear it as the sound of a bell.
Still, listening closely makes it obvious that the overtones are mixed far more cleanly than in a real bell. A real bell does not sound this clean. It contains numerous components that tend to destroy its sense of pitch (non-integer partials or irregular harmonics), and the pitch a fifth above is more clearly differentiated. The reasons that it is more differentiated are first that it is extremely loud, and second that it is not precisely a fifth, but slightly (actually quite a bit) skewed.

So, let's start with this preset sound, and try to make a more realistic bell sound. You've probably noticed that as our series continues, the content is becoming more difficult. For metallic sounds in particular, there's more logic involved. After all, we're now talking about "equations" (something I personally dislike). We'll start by a complete unveiling of the parameters.

	OSCILLATOR				ENVELOPE GENERATOR								KEYBOARD LEVEL SCALING					KEY BOARD	OPERATOR		MOD SENS	
OP	mode / sync		frequency		rate				level				break	curve		depth		RATE	output	velo		
No.	mode	sync	coarse fine	detune	R1	R2	R3	R4	L1	L2	L3	L4	point	L	R	L	R	SCALING	level	sens	pitch	amp
1	Ratio		1.00	0	95	33	71	25	99	0	0	0	A-1	-LIN	-LIN	0	0	4	99	3		0
2	Ratio		3.50	2	98	12	71	20	99	0	0	0	C1	-LIN	-EXP	0	75	4	80	2		0
3	Ratio		1.51	0	95	33	71	26	99	0	0	0	A-1	-LIN	-LIN	0	0	4	99	4		0
4	**Ratio**	OFF	**5.00**	**0**	**98**	**12**	**71**	**29**	**99**	**0**	**0**	**0**	**G#1**	**-LIN**	**-LIN**	**0**	**31**	**2**	**79**	**5**	1	**0**
5	Ratio		1.50	0	95	70	33	24	99	95	0	0	C2	-LIN	-LIN	0	21	4	99	4		0
6	Fixed		794.3Hz	-7	98	60	74	60	99	97	0	0	A-1	-LIN	-LIN	0	0	2	73	5		0

There are three key points. The first is that by setting the frequency of the remaining two carriers to 1.50 we have caused the overtone a fifth above to be emphasized more than the fundamental. And by skewing one of them slightly to 1.51, we create a sense of separation from the fundamental, and an overall sense of modulation. If this is changed to 1.52 or 1.54, or conversely to 1.49, other interesting effects can be created.
The second key point is the level scaling. In the high range, we enhance the realism by lightening the modulation. Not just on bells, but on guitar or piano as well, playing upward into the higher register causes the number of overtones in the sound to diminish, producing a more mellow sound that is closer to a sine wave. It's a somewhat mysterious feeling.
The third is the velocity setting. For some reason, the preset has a velocity setting only for operator 5. I think that by using playing dynamics to apply a bit more change to the volume and tone, we can produce a better simulation of a struck object.
Let's get into some more details. First is operator 4. The frequency is 5.00, but there is no reason for this. I tried 3.00, 4.00, 5.00, and 6.00, and liked this setting the best. This sort of vagueness may not seem to match with my talk about complicated things like "non-integer overtones." Still, results and not logic are what is important when it comes to sound. In the case of FM tone generation, some types of sound do allow you set the frequency ratios according to logic, but sounds created in this way are not always necessarily the best. The same goes for music. It's my own opinion that if the results are good, you can do anything you like even if it doesn't follow the theory.

Let's move on to operator 6. The mode is Fixed, and the frequency is 794.3 Hz. Of course, this too is a number based solely on its result. This modulator has two purposes; 1) to weaken the sense of pitch (or strengthen the sense of inharmonicity), and 2) to give individuality to each pitch. Since bells produce only one pitch from each sounding object, each one has a unique character. It is somehow unnatural for the pitches to change too evenly.
By adding a modulator whose pitch is Fixed, i.e., whose frequency is not changed by the pitch that you play, we can apply modulation at a differing ratio to each note. In other words, the pitch and the parameter value will not move in parallel. This will be easier to understand if you temporarily set the frequency in the 1,000 Hz range or the 2,000 Hz range.
This completes our sound. It started as TUB BELLS (tubular

bells), but now has the additional character of ordinary hanging bells. (Of course, they were related to begin with.) I've renamed it DX BELL.
Listen to this sample using either the MIDI file or the audio file. In particular, the octave centered on C3 is quite realistic.

MIDI Data
bell_sample1.zip (1.5kbyte)

SoundVQ Data
jazorg.vqf (100kbyte)

Load 64DxVoice.MID first as cartridge file.

Now look at the edit list shown below.

	OSCILLATOR				ENVELOPE GENERATOR								KEYBOARD LEVEL SCALING					KEY BOARD RATE SCALING	OPERATOR		MOD SENS	
	mode / sync		frequency		rate				level				break point	curve		depth			output level	velo sens	pitch	amp
OP No.	mode	sync	coarse fine	detune	R1	R2	R3	R4	L1	L2	L3	L4		L	R	L	R					
1	Ratio	OFF	1.00	0	95	33	71	25	99	0	0	0	A-1	-LIN	-LIN	0	0	4	99	3	1	0
2	Ratio		7.00	2	98	12	71	20	99	0	0	0	C1	-LIN	-EXP	0	75	4	80	2		0
3	Ratio		3.00	0	95	33	71	26	99	0	0	0	A-1	-LIN	-LIN	0	0	4	99	4		0
4	Ratio		9.00	0	98	12	71	29	99	0	0	0	G#1	-LIN	-LIN	0	31	2	79	5		0
5	**Ratio**		**1.50**	**0**	**95**	**70**	**33**	**24**	**99**	**95**	**0**	**0**	**C2**	**-LIN**	**-LIN**	**0**	**21**	**4**	**99**	**4**		**0**
6	Fixed		794.3Hz	-7	98	51	56	48	99	97	0	0	A-1	-LIN	-LIN	0	0	2	80	5		0

All I did was to change DX BELL to a different frequency setting for the operators, but this has substantially altered the character of the sound. The biggest factor is that the ratio of 1.00 : 3.50 has disappeared. Since 7.00 is twice 3.50 (i.e., one octave higher than 3.50), this tonal character is not so very different than for a setting of 2.00 or 3.00. Still, it's quite brilliant. Most of the other operators are also set to a higher frequency setting, producing a fairly bright bell sound.
Try changing the frequency ratios in various ways and see how the character of the sound is affected. I'm sure you will be able to create any number of variations.

MIDI DataBR>
bell_sample2.zip (1.5kbyte)

SoundVQ Data
bell_sample2.vqf (100kbyte)

Load 64DxVoice.MID first as cartridge file.

Seminar Top

Session 11
The classic DX sound --- The Bell! (Part 2)

In our previous session, we edited a preset sound to create a bell. This time we're going to create a bell starting from the INIT VOICE. You already know how to create the INIT VOICE, don't you?
We're going to use algorithm 20. Why? Just because I thought it might be suitable. You may consider that an inadequate reason, but it's the truth.
It is true that certain algorithms are particularly suitable for strings, and that other algorithms are good for electric piano. But that is not the whole story. Many times I have selected a certain algorithm and attempted without success to create a particular sound, but was able to succeed after I changed to a different algorithm.

In voicing (creating sounds) on the DX, numerous sounds have been the result of chance or intuition. Of source, there are certain principles. For example, a frequency ratio of 1:1 with feedback at 7 produces a sound suitable for synth strings or synth brass, and modulation at a ratio of 1:3.5 (as in our previous session) produces a bell sound. Still, the most important thing is the "inspiration" or "intuition" of the programmer. So for those of you who are reading this series and would like to create original sounds on the DX, my recommendation is that even more than understanding the theory of FM tone generation, you develop your sense of "inspiration" and "intuition." And in order to develop your sense of inspiration and intuition, I suggest that you

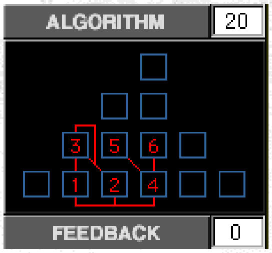
*** Go for the numbers.**

More than understanding with your head, learn with your body. This might sound like sports, but I think it's an idea that's closer to cooking.
Take for example, adding salt or spices. A master chef can be amazingly accurate without actually measuring amounts. The next important thing is

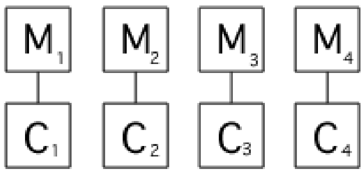
*** If you feel a good sound coming on, even if it's not the one you were intending, leave your intention for a time, and take the sound where it wants to go.**

I touched on this a bit in our 9th session, but let me explain here more fully.
Suppose that you are attempting to create a violin sound, but the sound is becoming more like a brass sound. What should you do? In such times, leave off trying to create the violin, and go ahead with creating the brass instrument. Authors often say that the characters in their books start acting on their own. In a sense, this is a similar phenomenon. You might be trying to create a violin, but feel that the sound wants to be a brass instrument.
The path that I chose intending to create a violin was actually the path by which I arrived at a brass instrument, and this sort of cumulative chance can be very important. Each instance of trial and error will be compiled into the database in your brain.

Let's leave the generalities at this point, and start creating our bell. We're going to create a "slightly ethnic bell."
Check the algorithm.



This is similar to our old familiar 5 and 6. It appears that the difference is just that the modulator connection is different. However depending on the settings, this algorithm provides the functionality of a total of eight operators; four carriers and four modulators (see the figure below).



Let me explain. In the above diagram, if modulators 1 and 2 (M1, M2) have the very same settings, they can be combined into one. And if carriers 3 and 4 (C3, C4) have the very same settings, they can be combined into one. The result is the same as algorithm number 20. Turning this around, it means that we can use six operators to perform (almost) the same functionality as eight operators, allowing us to use a wider variety of effects then algorithms that are simply connected one to one.

Now on to the editing.

OP No.	OSCILLATOR mode / sync: mode	OSCILLATOR mode / sync: sync	OSCILLATOR frequency coarse fine	OSCILLATOR detune	ENVELOPE GENERATOR rate R1	R2	R3	R4	ENVELOPE GENERATOR level L1	L2	L3	L4	KEYBOARD LEVEL SCALING break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	OPERATOR velo sens	MOD SENS pitch	MOD SENS amp
1	Ratio	OFF	1.00	0	56	99	28	28	99	99	0	0	A-1	-LIN	-LIN	0	0	3	90	3	3	0
2	Ratio		4.11	0	96	75	28	28	99	94	0	0	A-1	-LIN	-LIN	0	0	3	99	3		0
3	Ratio		5.05	0	50	99	20	20	99	99	0	0	C3	-LIN	-LIN	0	57	3	73	2		0
4	Ratio		2.00	0	96	70	28	28	99	95	0	0	A-1	-LIN	-LIN	0	0	3	99	3		0
5	**Ratio**		**5.07**	**0**	**95**	**60**	**20**	**20**	**99**	**76**	**0**	**0**	**C3**	**-LIN**	**-LIN**	**0**	**10**	**3**	**76**	**2**		**0**
6	Fixed		1259Hz	0	55	50	20	20	99	95	0	0	C3	-LIN	-LIN	0	20	3	64	3		0

Some of you may already have noticed, but for bell-type metallic sounds, the carrier and modulator frequencies are distant, and there are many odd-numbered overtones. Ratios such as 1:1 or 2:1 that we used when creating electric pianos do not usually appear. Conversely when creating strings, brass, or woodwinds, the classic pattern is to use a ratio of 1:1 (in most cases) or 1:2 (mainly for clarinets), and to also use feedback.
What I want you to notice is that there are many operators with a decimal place in their frequency setting.
This is because we need inharmonic or irregular overtones that are outside the regular overtone series.
Metal plates or metal rods that are solid and have a fixed shape, such as the tone bars of an electric piano or a vibraphone, have a fairly regular arrangement of harmonics. But in the case of a hollow pipe

such as a tubular bell, or a curved metal plate as in a gamelan, the inharmonic overtones are prominent. There are reasons for this, but no space here to explain. Maybe another time.

In any case, the sound produced by the parameters in the above edit list was not created with any particular instrument of any particular country in mind, but if I was pinned down, I might say that I was thinking of a feeling somewhere between Thailand and Bali.
Before I explain the structure of the sound, please listen to the sound.

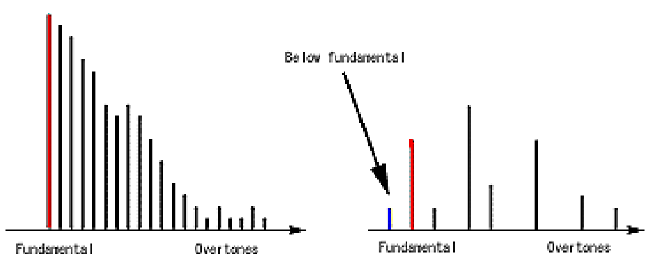
SAMPLE DATA DOWNLOAD **MIDI Data** bell_sample3.zip (1.5kbyte)

SAMPLE DATA DOWNLOAD **SoundVQ Data** bell_sample3.vqf (100kbyte)

Load 64DxVoice.MID first as cartridge file.

Since the scale I used was Balinese, it creates a fairly strong Indonesian impression
The core of this sound of course lies in the operator frequency, but is also in the EG (envelope generator) settings.
Look at Rate 1 of operators 1, 3, and 6. This is set fairly slow, in order to produce a slightly delayed "fwhan" or "dwhan" sound similar to that of a steel drum immediately after it is struck. This is just a slight delay of several dozen milliseconds.
If we attempt to produce this together with the attack (impact), we get a sound that's more like "kwhan" or "twhan." (What bizarre expressions!)

Let's go backward one step to the operator frequency. Unlike a violin, in which many overtones are packed together, this type of bell has just a few characteristic overtones scattered around. (Refer to the figure below. On the left is shown the packed overtones, and on the right is shown the overtones of a metallic object. *Note: These diagrams are not intended to be precise.)



In particular, struck objects often produce a pitch that is lower than the fundamental.

For this sound as well, you can try changing the operator frequency and output level in various ways to create variations to your taste. By now, this should be easy for you.
If you are not satisfied with simply changing the operator frequency and output level, it's time to move from beginning to intermediate level. Go ahead and try editing the EG (Envelope Generator). The modulator EG produces a big change in the tone. The important points of the EG are R1 (attack time), R3 (decay time), and R4 (release time).

For your reference, here is a variation that I created from this sound. Here are the parameters.

OP No.	OSCILLATOR mode	OSCILLATOR sync	frequency coarse fine	detune	EG rate R1	R2	R3	R4	EG level L1	L2	L3	L4	KLS break point	curve L	curve R	depth L	depth R	KEY BOARD RATE SCALING	OPERATOR output level	velo sens	MOD SENS pitch	amp
1	Ratio		1.00	0	95	99	28	28	99	99	0	0	A-1	-LIN	-LIN	0	0	3	90	3		0
2	Ratio		4.00	4	96	75	28	28	99	94	0	0	A-1	-LIN	-LIN	0	0	3	99	3		0
3	Ratio	OFF	5.00	0	95	99	20	20	99	99	0	0	C3	-LIN	-LIN	0	57	3	73	2	3	0
4	Ratio		2.00	7	96	70	28	28	99	95	0	0	A-1	-LIN	-LIN	0	0	3	99	3		0
5	**Ratio**		**5.00**	**3**	**95**	**60**	**20**	**20**	**99**	**76**	**0**	**0**	**C3**	**-LIN**	**-LIN**	**0**	**10**	**3**	**76**	**2**		**0**
6	Ratio		12.06	0	90	50	20	20	99	95	0	0	C3	-LIN	-LIN	0	20	3	67	3		0

I've made few changes other than the operator frequencies and R1 of the EG, but the sound has changed from an ethnic bell to a high-class European music box. I think that the impression it makes is completely different. (If you don't think that it's that different, please accept my apologies.)

@

SAMPLE DATA DOWNLOAD **MIDI Data** bell_sample4.zip (1.5kbyte)

SAMPLE DATA DOWNLOAD **SoundVQ Data** bell_sample4.vqf (100kbyte)

Load 64DxVoice.MID first as cartridge file.

The sample performance uses XG parameters to apply chorus deeply. The original DX did not have such effects or filters, so there was no way to edit the sound other than by using the FM parameters. However the plug-in board DX does provide various XG parameters, and this alone is enough to create quite a lot of variations. However if you really want to attain the summit of FM tone generator editing, you should avoid using the XG parameters until the very last stage. Certainly you should not apply chorus or reverb while you are creating the sound.
Turn the effects off as far as possible (I myself would say completely off), set all filters to the completely open position, and use just the FM parameters to create the sound until you are sure that no further progress is possible. Then store this in memory, and use the filter and effects. Unless you do this, you will loose track of which elements of the sound are due to the FM parameters and which elements are due to the XG parameters. Conversely if you wait to apply effects until the sound is completed, you can be amazed at what a great sound you created.
In actuality, I was rather impressed when I applied chorus and reverb, and played the music box sound that I had created by editing the first sound.

We have come to the end of our series. Even on the one topic of metallic sounds, there is much more to be said. It feels as though there is virtually an infinity of sounds that we have not touched. Someday if I have the opportunity, I might be able to continue this series on the web. If and when that happens, I'll see you again!

- The End -

Seminar Top