# An Introduction to the SSSP Digital Synthesizer

William Buxton, E. A. Fogels, Guy Fedorkow, Lawrence Sasaki, and K. C. Smith

The Structured Sound Synthesis Project
Computer Systems Research Group
University of Toronto
Tóronto, Ontario
Canada
M5S 1A4

## 1. Introduction

One of the main interests of the Structured Sound Synthesis Project (SSSP) is to develop a highly interactive environment to serve as an aid in the composition of music. We would like to do this within the context of a small, inexpensive, accessible system. In order to resolve the conflict between the high computational demands of sound synthesis and these design objectives, we have developed a special-purpose digital sound synthesizer. Work on this project began in January 1977, and the device was functioning in a limited state by December of that year.

The design owes much to the Dartmouth synthesizer (Alonso *et al.*, 1976) as well as the VOSIM oscillator (Kaegi and Tempelaars, 1978). Generally described, the synthesizer consists of one "real" oscillator which is time-division-multiplexed so as to function as sixteen oscillators. There are two aspects of the device which we see as particularly significant. First it was designed so as to incorporate in hardware five important techniques of sound synthesis: fixed waveform, frequency modulation, additive synthesis, waveshaping, and VOSIM. Secondly, the device can be easily interfaced to most digital computers.

The generators are essentially fixed sampling rate, accumulator-type digital oscillators. The sampling rate of each oscillator is 50 kHz, with a system bandwidth of 20 kHz. Dynamic range is well over 60 dB (and should improve with further adjustment) while frequency resolution is approximately .7 Hz (linear scale) over the entire bandwidth. This will shortly be improved to enable resolution of less than one "cent" at even extremely low frequencies[1]. The output signal of each oscillator can be fed to one of four analogue output busses which may then either be fed directly to an amplifier, or to a *channel distributor* (Fedorkow, 1978; Fedorkow, Buxton and Smith, 1978). The waveform output by each generator may not only be user defined—up to eight waveforms available at one time—but one may switch waveforms in mid-cycle. This is possible since the sixteen oscillators share a 16k buffer of 12-bit words to store waveforms. This 16k of RAM is partitioned into eight 2k blocks, one for each of the eight possible waveforms defined by the user or system. Apart from this memory configuration, this synthesizer is particularly interesting because the oscillators may be used to generate sounds according to five different synthesis modes: fixed waveform, frequency modulation, VOSIM, additive synthesis, and waveshaping[2]. This goes a long way towards a "universal module"—that is, all modules of a uniform type (with the resulting ease of conceptualization and communication). While this is in direct contrast with analogue synthesizers, a very wide repertoire of sounds is possible (including all phonemes in Indo-European languages, for example). We shall now present in greater detail the actual design of this device.

## 2. Technical Details

In this section, we shall give the design details of the digital synthesizer. Details will not, however, be taken to the logic level. Rather, the purpose is to illustrate and discuss the design approach to a level that enables the reader to evaluate the appropriateness of this design as compared to the alternatives. We shall begin by presenting an overview of the general architecture. This is followed by a discussion of the method of frequency control. Finally, a presentation of the various acoustic models embodied in the design is made.

### 2.1 General Architecture

The general layout of the device is shown in Figure 1. The synthesizer itself is made up of four main modules: the controller, memory, oscillator, and digital-to-analogue converter modules, respectively. Communication among the modules is via a single high-speed data bus which is under the supervision by the controller module. Communication with the host computer is achieved *via* a parallel bus connecting the controller with an address decoder residing on the host computer's input/output bus.

Before proceeding to present details of the synthesizer's functional organization, a few points should be made concerning the method of interfacing. By isolating the synthesizer from the host, we are able to power down, test, repair, and/or modify the synthesizer *without* having to also power down the host. Furthermore, the host is protected from damage due to faults or damage in the synthesizer. These were important considerations given that the synthesizer was hosted by an expensive time-sharing system. Other users could not be allowed to suffer due to work being undertaken on the synthesizer. All of this was even more important since we take the same iterative approach to implementing hardware as we do for software. Namely, the device was up and running in the simple fixed waveform mode well before—one by one—the other modes were added.

### 2.2 Frequency Control

The basis of the digital oscillator is the sampling of a stored function. In this case, the function stored is one cycle of a selected waveform. The waveform is stored as 2K 12-bit

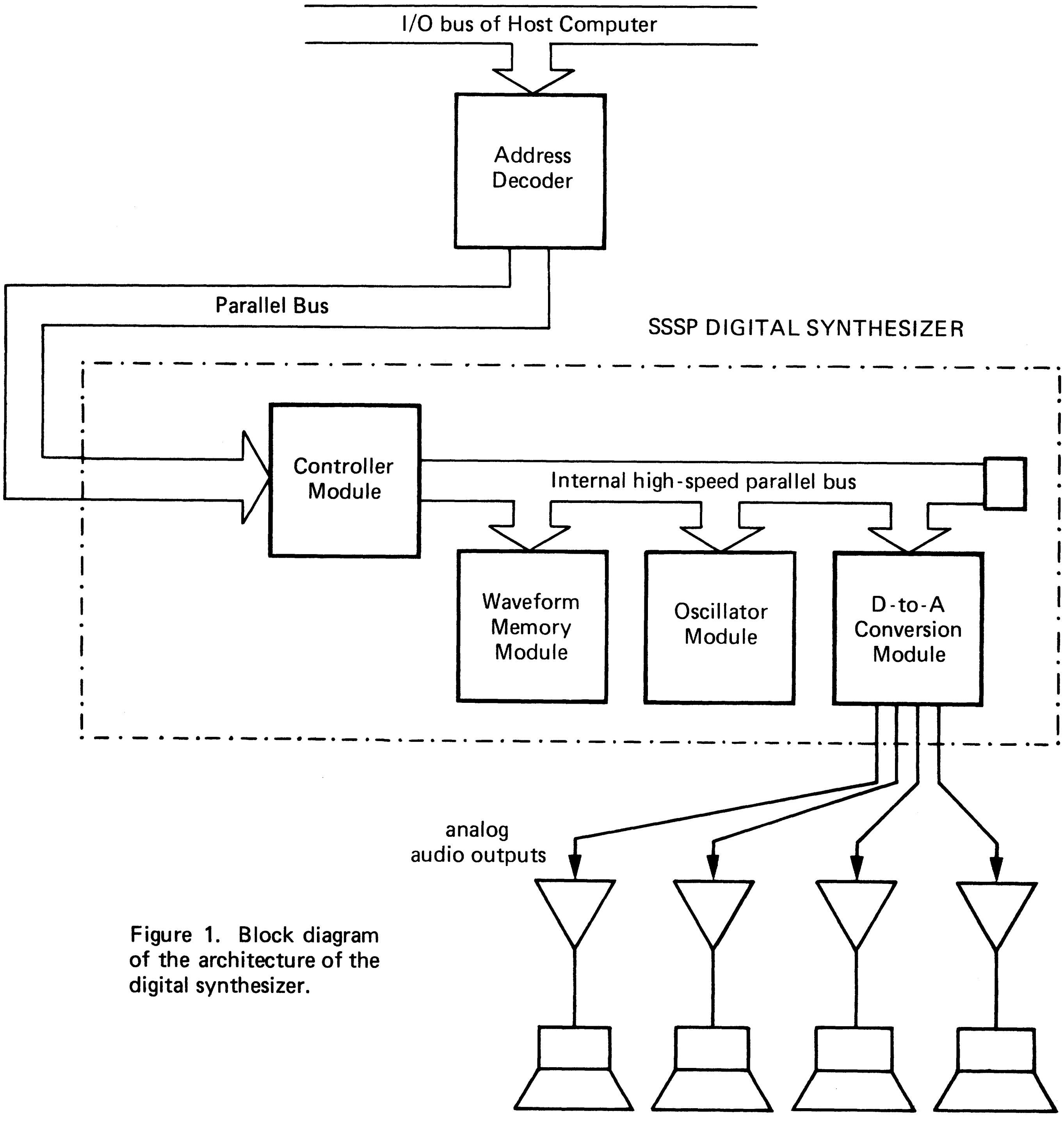



Figure 1. Block diagram of the architecture of the digital synthesizer.

samples in a random access memory (RAM) *internal to the synthesizer.* Sound is generated by outputting (scaled) samples from this table through a digital-to-analogue converter (DAC) which is connected to some transducer such as a loudspeaker.

Since the waveform buffer (WFB) contains only one cycle of the waveform, frequency (cycles per second) is controlled by the number of times we cycle through the samples of the buffer each second. This can be controlled two ways. One is to sequentially output each sample of the buffer every cycle. Since the number of samples output per cycle is constant—as is the WFB size—the *rate* at which samples are output must vary for each frequency. The second alternative is to keep the sampling rate constant and vary the number of samples output each cycle; for example, if at frequency $f$ every sample is output, then to output the frequency $2f$ we would output every second sample in the buffer. (Outputting half as many samples—equally spaced throughout the buffer—at the same rate doubles the frequency.) In the first case, frequency is specified in terms of the sampling rate; in the second, in terms of the offset between subsequent samples in the buffer.

Both approaches have been used with success. The variable sampling rate oscillator is used, for example, in the Dartmouth Synthesizer (Alonso, et al., 1976) and the VOSIM oscillator (Kaegi and Tempelaars, 1978). However, two main problems must be overcome in taking this approach. First, when a bank of such oscillators is used—each at a different frequency—their samples are output asynchronously, making it difficult to mix or process them digitally. Hence, each oscillator in the Dartmouth synthesizer has its own DAC. The second main problem is that at low frequencies the sampling rate may enter into the audio band (below about 16 kHz). This requires special consideration in terms of the filters following the DACs. Consequently, the cut-off frequency of the low-pass filters may be required to vary with the sampling rate.

In our system, we have chosen to take the more straightforward fixed sampling rate approach. The technique is well understood (it is the basis of the MUSIC V software oscillators), the sampling of different oscillators is synchronous (making the time-division multiplexing of several oscillators rather straightforward), and the final stage filters have a fixed cut-off frequency. As stated above, frequency in this type of oscillator is controlled by specifying the offset in the WFB between subsequent samples. This offset—or increment—we call F_INC ("frequency increment"). Given the address of any sample $n(A_n)$, then:

$$A_{n+1} = (A_n + \text{F\_INC}) \text{ modulo WFB size}$$

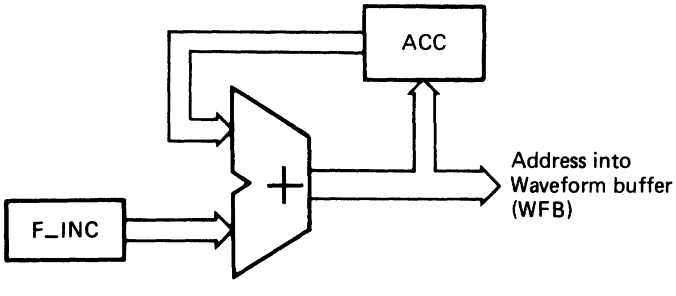



Figure 2. Simple ramp generator.

A generalized view of this mechanism is shown in the simple ramp generator shown in Figure 2. Here it is seen that the sum of the addition (and hence the address of the current sample) is accumulated in the register ACC—to be used in calculating the address of the next sample. The modulo arithmetic is accomplished by simply ignoring the carry bits. Converting frequency from Hz to F_INCs is straightforward, given the sampling rate and WFB size. This is effected using the following formula:

F_INC = FREQ * WFBS/SR

where FREQ is the frequency in Hertz, WFBS is the waveform buffer size, and SR is the sampling rate.

### 2.3. Fixed Waveform

We obtain the fixed waveform mode of operation through a slight extension of Figure 2. A simplified presentation of this mode is made in Figure 3. Here, the principal components added concern the WFB table lookup and the DAC mechanism. In addition to F_INC, four new addressable registers appear. These are: WF_SEL, OP_SEL, ENV, and AMP.

As was stated in the Introduction, there are eight buffers in which waveforms can be stored. Therefore, besides calculating the address within any particular WFB (the process illustrated in Figure 2), one must also specify from which of the eight WFBs the samples are to be taken. This is the purpose of the register WF_SEL ("waveform select"), which is simply a three-bit value concatenated onto the address calculated by the ramp-generator. One benefit of being able to easily change waveforms is the potential for minimizing distortion due to aliasing, or fold-over. This can be accomplished simply by substituting simple sine tones for complex signals whose fundamental frequency is above a certain threshold. Since the harmonics of such signals would fall outside of the bandwidth of human pitch perception, the substitution would not be perceived and the generation of partials above the Nyquist frequency would be avoided.

There are four audio output channels in the synthesizer. The output of each of the sixteen oscillators may be routed to any one (and only one) of these four channels. The purpose of the register OP_SEL is simply to specify to which of these output channels the oscillator's output is to be routed.

Once a sample is obtained from the WFB, it is generally scaled in amplitude so as to be able to produce sounds of different loudness. One technique of doing so is to digitally multiply the sample by a scaling factor and then output the product through a normal DAC. This is the technique used by Alles and di Giugno (1978), for example. At the time of design, however, it appeared more economical to take an approach similar to that of the Dartmouth synthesizer. Here we carry out the scaling through the use of multiplying DACs, which were less expensive and complex than digital scaling. The waveform sample is placed in a 12-bit multiplying DAC and the output is scaled according to a reference voltage input. Thus, even at low amplitudes there are 12-bits of resolution of the waveform, resulting in better dynamic range than would be possible with an ordinary "fixed-point" 12-bit converter.

One interesting idea of Alonso's which is incorporated into our design deals with the derivation of this reference, or

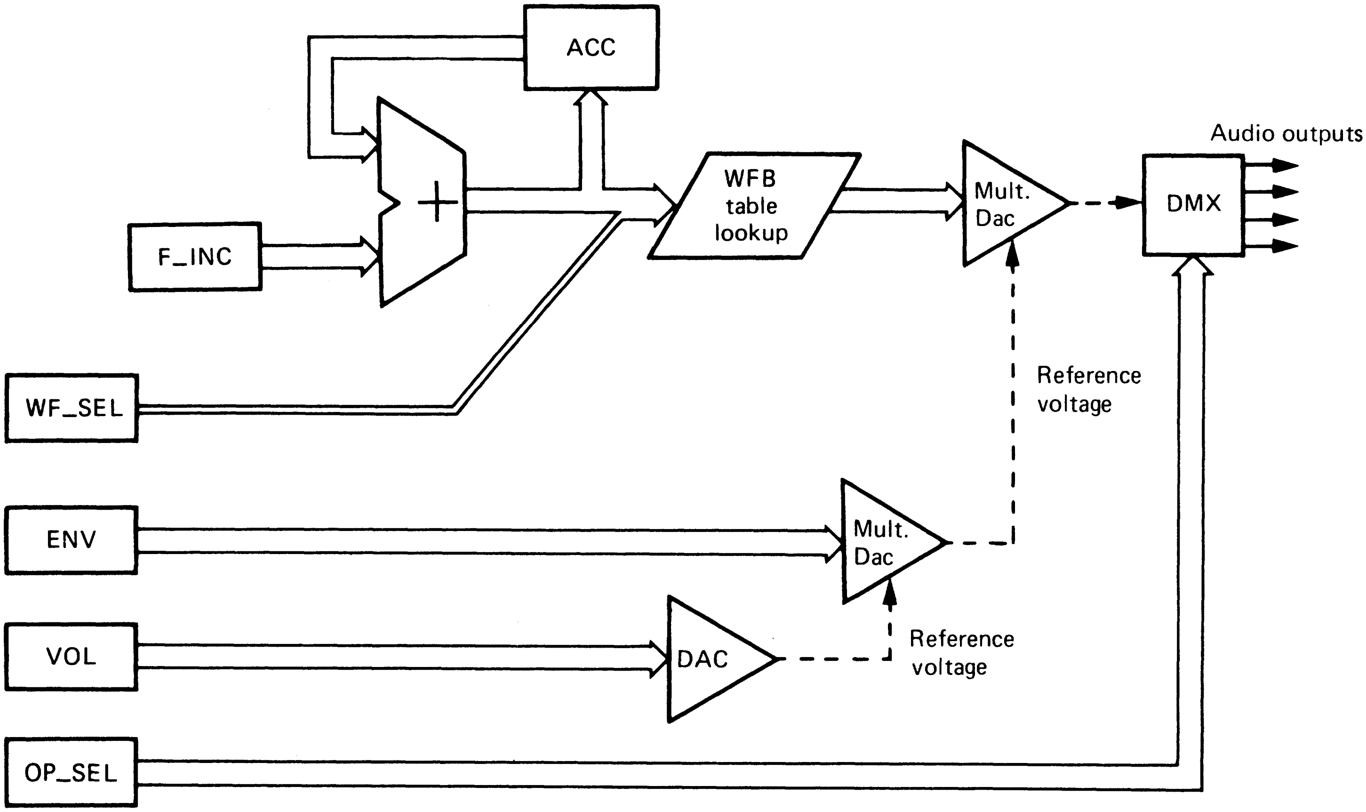



Figure 3. General view of simple fixed-waveform digital oscillator.

"scaling" voltage. Rather than coming from a single amplitude value, separate notions of "envelope" and "volume" are carried over into hardware. That is, the envelope of a sound is scaled in hardware rather than software, thereby saving valuable CPU time. Amplitude scaling is, therefore, accomplished by three DACs in series. The first is the volume DAC. The second and third—both multiplying DACs—are the envelope and waveform DACs, respectively. The output of the volume DAC (as determined by the contents of register VOL) is used to scale the output of the envelope DAC (whose unscaled output is determined by the contents of register ENV). The scaled output of the envelope DAC is then used as the scaling voltage for the waveform DAC. One point worth noting in the current implementation concerns the volume DAC. Since loudness varies more logarithmically than linearly, a logarithmic DAC is used. Thus we have an example where psychoacoustic research has affected hardware design.

### 2.4. Additive Synthesis: Bank Mode

In the preceding section we saw how we can generate a sound having a particular fixed waveform and a varying amplitude contour. Given the principles of additive synthesis (Risset and Mathews, 1969; Grey, 1975), we see how a group of fixed waveform oscillators can be used to generate complex sounds having time-varying spectra. In this case, we have one oscillator corresponding to each partial to be synthesized. Since the technique simply involves the use of a group of fixed waveform oscillators, we refer to it as *bank mode.* For the same reason, we see that the technique requires no special-purpose hardware beyond that already described.

### 2.5. VOSIM Mode

The basis of VOSIM (Kaegi and Tempelaars, 1978) is the ability to output one cycle of a particular function (such as a cosine pulse[3]) followed by a controlled delay before the next cycle is output. Let us assume that the function is stored in one of the WFBs. Our method of implementation, then, incorporates a mechanism which outputs one cycle of the function stored in the WFB (the period controlled by F_INC), and then steps into a time-out mode for a specific delay period. A simplified illustration of our implementation of this mechanism is shown in Figure 4. Here it is seen that the period of delay is controlled by the contents of the register DEL. The oscillator functions in two modes: cycle and timeout. Cycle mode ends and timeout is triggered when there is a carry-bit out of register ACC (i.e., at the peak of the ramp, or when the WFB addressing "wraps around"). At this time—timeout—the contents of DEL are loaded into the count/compare (CNT/COMP) register which is decremented every 1/50,000th sec. When the contents of this register equals zero (0), cycle mode is re-triggered and the CNT/COMP register disabled. We see, then, that when DEL equals zero we are continually in cycle mode and therefore effectively in fixed waveform mode of operation.

The VOSIM mechanism as described thus far is an over-simplification to facilitate the presentation of material. What the description omits is the method for controlling random deviation, or noise in the sound. The mechanism employed is illustrated in Figure 5. Again, the register CNT/COMP is loaded at the start of each timeout cycle. Similarly, CNT/COMP containing the value zero still triggers cycle mode, while an overflow from ACC still triggers timeout. The difference, however, is in the value which is loaded into CNT/COMP. Instead of simply loading the value contained in register DEL, as diagrammed in Figure 3, the value loaded is the value contained in DEL *plus* a random value. The range of this random value is plus or minus some specified percentage of the contents of DEL. This percentage value is determined by the contents of the register DEV (% deviation). The actual origin of the random value is the random number generator labelled RNG. It is clear from Figure 5 how the actual delay—the value loaded into the register CNT/COMP—is arrived at. A few points are worth noting, however. First, when the contents of DEV equals zero, we have effectively the situation diagrammed in Figure 4. Second, when the contents of DEL equal zero, we still effectively have fixed-waveform mode. Finally, the effective (average) fundamental frequency in VOSIM mode is determined by a combination of the contents of *both* the F_INC and DEL registers.

### 2.6. Frequency Modulation

The synthesizer has sixteen digital oscillators. Frequency modulation (Chowning, 1973) is implemented such as to allow any oscillator *n* to frequency modulate oscillator *n + 1* (modulo 16). While this format does not allow the use of multiple modulators of a single carrier wave (such as described in Schottstoedt, 1978), this deficiency is largely made up for by our ability to use modes other than FM. At the time the device was designed, the tradeoff was weighted towards economy and accessibility rather than generality.

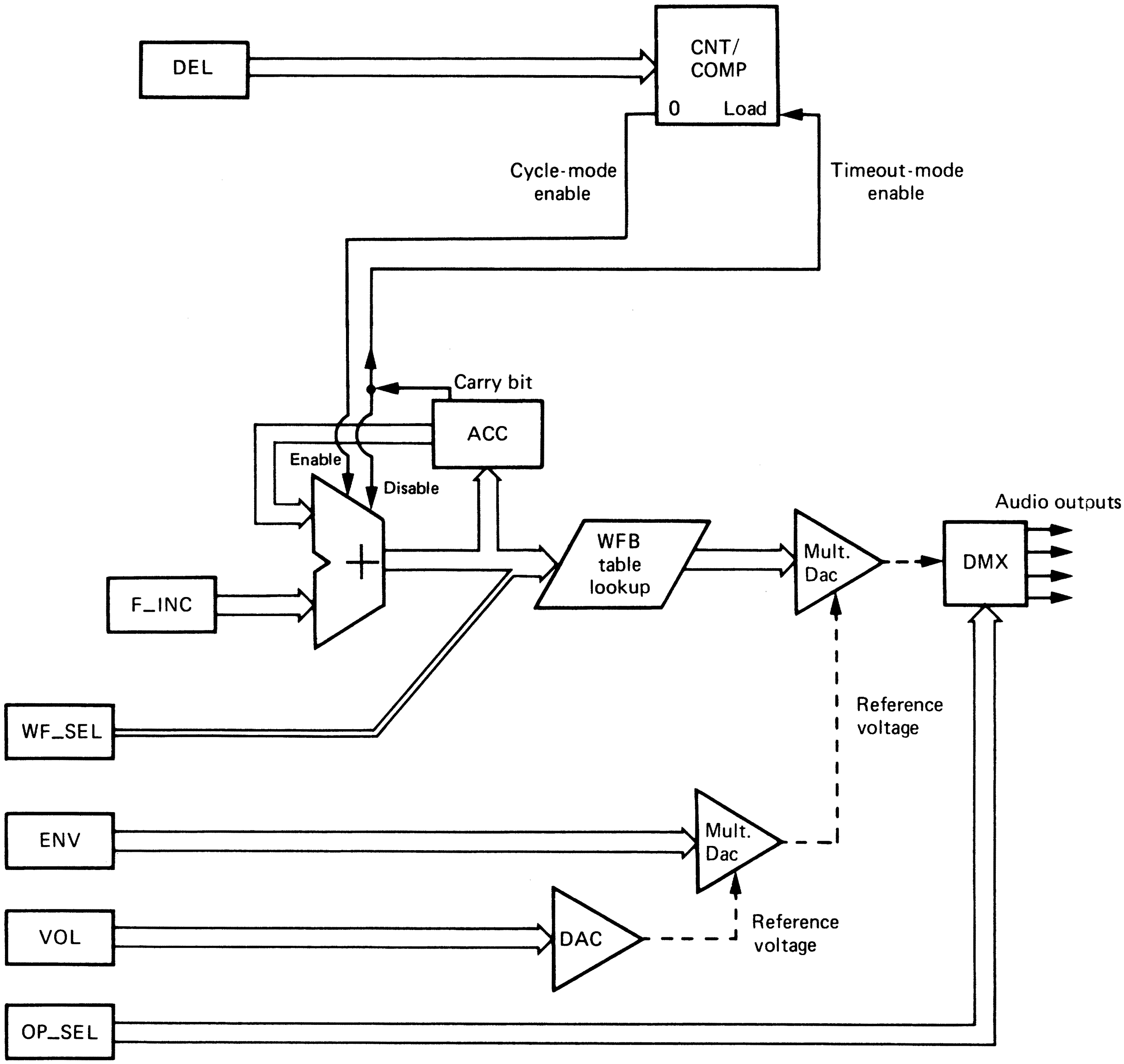



Figure 4. Simplified version of VOSIM oscillator.

In implementing FM certain extensions had to be made upon the basic oscillator as described thus far. These fall into two categories: those which enable the oscillator to be modulated, and those which enable it to modulate. The method of implementation is shown in Figure 6. Here a pair of oscillators are shown. For simplicity's sake the VOSIM components have been omitted (as in fixed waveform mode, register DEL would be set to zero). Similarly, since the DACs of the modulating oscillator are not used (i.e., are set to zero), they are not shown. Finally the diagram is made such that the first oscillator shows only the modulating mechanism, while the second shows only the additions to allow it to be modulated. It should be remembered, however, that both oscillators are in fact identical. This is seen in Figure 8, which shows a single oscillator which includes the mechanisms for all oscillator modes.

Returning to Figure 6, we see several points of interest. First, the maximum deviation of the frequency of the carrier oscillator is determined by the product obtained by multiplying the contents of the registers F_INC and MOD_INDEX

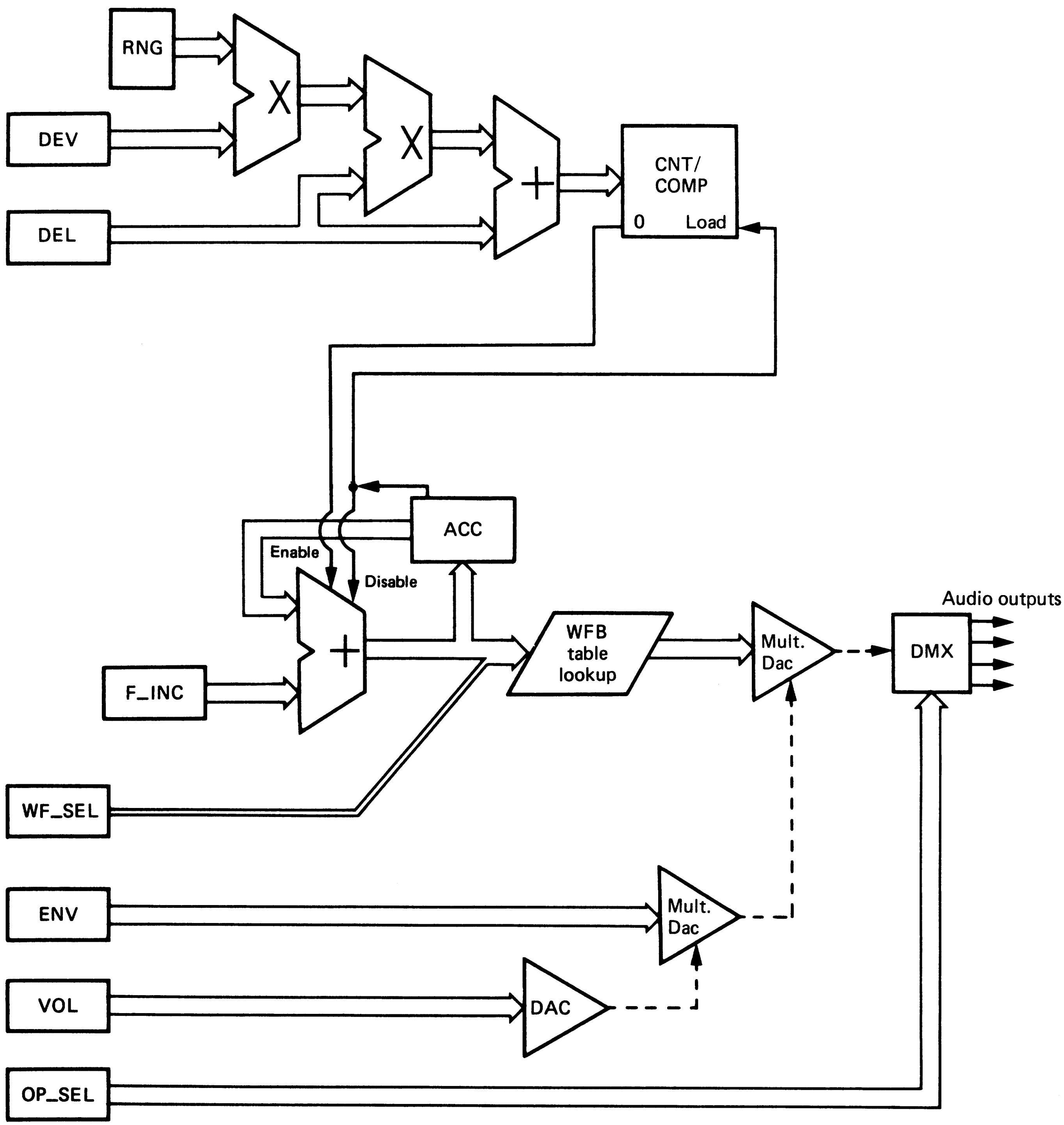



Figure 5. The VOSIM mechanism.

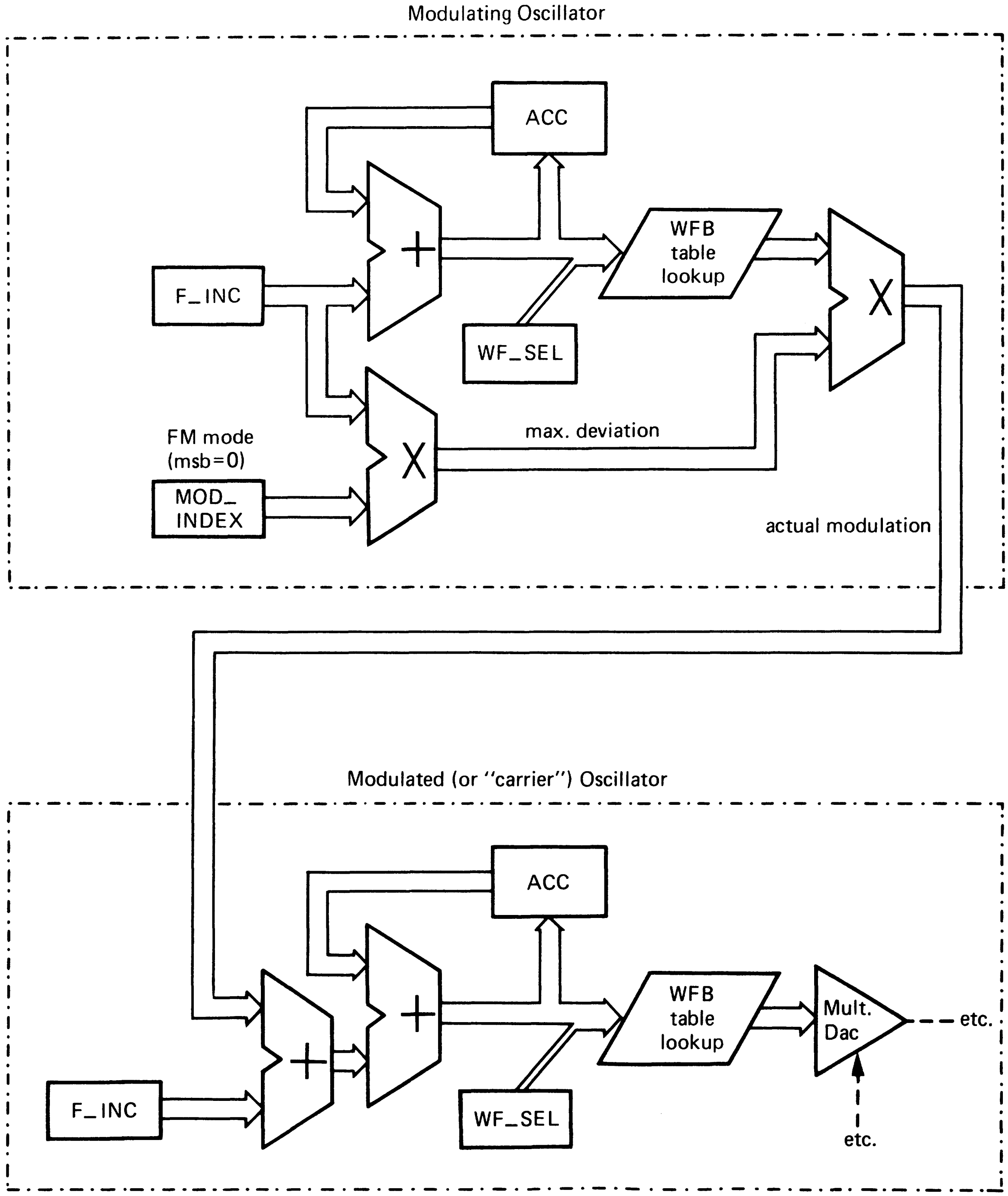



Figure 6. Simplified view of frequency modulation.

of the modulating oscillator. Second, the actual instantaneous amount of deviation (MODULATION) is derived by multiplying the maximum deviation by the current sample taken from the WFB (again, of the modulating oscillator). Finally, the actual modulation is effected by adding the MODULATION to the contents of the register F_INC of the carrier oscillator. The sum of this addition is then used as input into the ramp generator.

There are a few additional points to note in the above. First, remember that both modulator and carrier may address any one of the eight WFBs through the use of their WF_SEL registers. Thus, we are not restricted to FM with sine waves only. Second—and not so obvious—every oscillator is *always* modulating its neighbour. However, in fixed-waveform mode, for example, these MOD_INDEX registers are set to zero—effectively switching off the modulation. Finally, note that the richness of sound obtainable through FM is gained at the expense of two oscillators per sound. In complex structures this makes the low number of oscillators (sixteen) felt rather strongly. (This is equally true in waveshaping mode, and even more true in additive synthesis.) The user must, therefore, often resign himself to resources comparable to a quartet or octet rather than that of a symphony. Given the other benefits of the system, however, these limitations seem not so serious. There has, after all, been a great deal of "acceptable" chamber music written over the years.

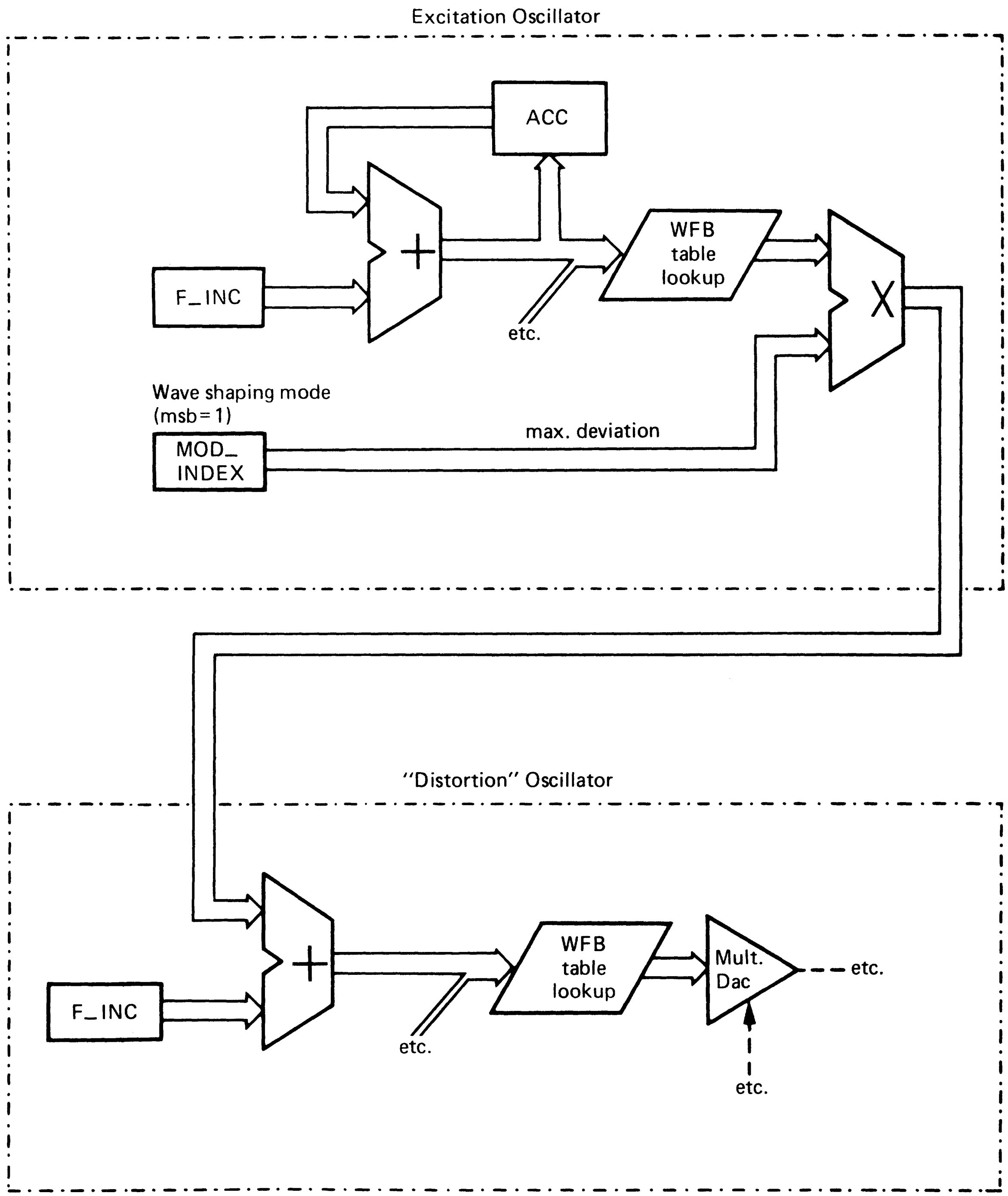



Figure 7. Functional view of waveshaping mechanism.

## 2.7. Waveshaping

Inspired by LeBrun (1977), we decided to determine if we could incorporate waveshaping into the hardware structure. This turned out to be rather easier than expected and the result is a slight variation on the FM mode already described. A functional representation of the waveshaping implementation is shown in Figure 7. Like Figure 6, we see only the critical components of two oscillators. However, instead of referring to the oscillators as "modulator" and "carrier" as in FM, we will refer to them as "excitation" and "distortion", respectively.

Starting with the excitation oscillator, note that the configuration is just as in FM except that the multiplication of the contents of the F_INC and MOD_INDEX registers is missing. The contents of MOD_INDEX, therefore, function as a simple scaling factor for the samples taken from the WFB. Secondly--concerning the distortion oscillator--notice that the entire ramp generating mechanism for WFB address calculation (including ACC) is missing. In this mode, the contents of F_INC is set to a value such that (by itself) it addresses the sample midway into the WFB. The output of the excitation is then simply added onto this constant and the sum is used as the address into the distortion WFB. The sample thus addressed is then output to a DAC, thereby completing the basic waveshaping process.

It still remains, however, to present how the mechanism shown in Figure 7 is obtained using that shown in Figure 6. The key to doing so lies in having the MOD_INDEX register function in two different modes. The current mode is determined by the most significant bit (msb) of the MOD_INDEX register. When this bit is zero, we have regular FM as described in the preceding section. When, however, the msb is equal to one for a particular oscillator, the following happens:

1. In the multiplication of the contents of the F_INC and MOD_INDEX registers, the F_INC factor is replaced by the constant "one". This effectively nulls the effect of the multiply. F_INC still controls the frequency of the excitation oscillator.

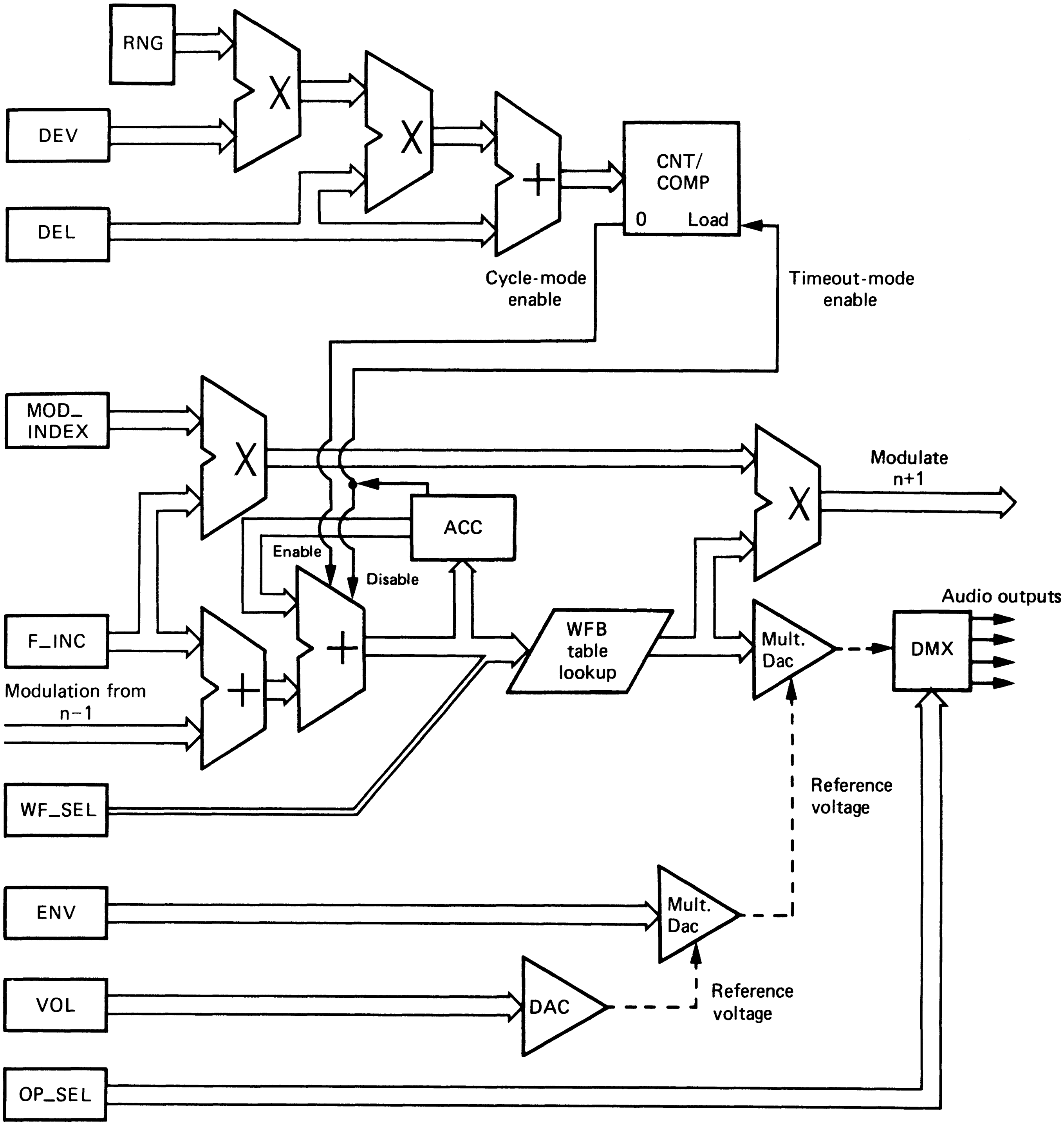



Figure 8. Composite view of digital oscillator.

2. The ACC register of the next oscillator (i.e., that of the distortion oscillator in the pair) is cleared *after every sample.* This effectively disables the ramp generator in the way diagrammed in Figure 7.

Thus, the requirements of waveshaping are seen to be straightforward, and the implementation not difficult.

## 3. Summary

An outline of the implementation of the various modes used in the digital synthesizer has been presented. Perhaps the most significant point about the implementation is that *all of the oscillators are, in effect, functioning in all modes at all times. The effects are simply made invisible by "nulling" appropriate registers.* The effect of this is that the synthesizer's resources can be easily distributed so as to permit sounds utilizing different acoustic models to be synthesized simultaneously with very little overhead on the CPU. In addition, while FM mode does not allow the use of multiple modulators for a single carrier, the hardware does allow for obtaining similar effects by using the output of waveshaping mode as the modulator in FM, or waveshaping the output of FM. The device is designed so as to produce good audio quality, and is able to be easily interfaced to a large number of different computers.

One aspect of the design which is now somewhat questionable is the method of implementing the digital-to-analogue conversion process. While we would retain the control mechanism – that is, the method of scaling the waveform using the ENV and VOL registers – the actual scaling would be done digitally rather than using multiplying DACs. Digital scaling would then enable us to use only four DACs in total. The current design was made in the interest of economy, and has served well in terms of fidelity and reliability. With changes made in the future, it is worth noting that the modular nature of the architecture enables us to change one module with only minimal trauma to the others.

Other drawbacks of the device are the limited number of oscillators and the lack of generality – when compared, for example, to MUSIC V. These are limitations which we are willing to accept for the time-being, given that our prime interest is in the *relationships among sounds,* rather than the intrinsic value of the sounds themselves. That is, we see our main priority in developing the software to manipulate the sounds currently in our repertoire (e.g., Reeves *et al,* 1978), rather than concentrating on expanding that repertoire. The next iteration of the synthesizer – currently in the final stages of design – will, however, offer sixty-four oscillators similar to those currently available.

The reader is redirected to Figure 8 where the synthesizer registers are collectively illustrated.

## 4. Acknowledgments

The work described in this paper has been undertaken as part of the research of the Structured Sound Synthesis Project (SSSP) of the University of Toronto. The SSSP is an interdisciplinary project whose aim is to conduct research into problems and benefits arising from the use of computers in music composition. This research can be considered in terms of two main areas: the investigation of new representations of musical data and processes, and the study of human-machine interaction as it relates to music.

The research of the SSSP is funded by the Humanities and Social Sciences Research Council of Canada (grant S76-0099), while logistic support comes from the Computer Systems Research Group (CSRG) of the University of Toronto. This support is gratefully acknowledged.

## 5. Notes

1. One "cent" is an interval equal to 1/100 of a tempered semi-tone. One cent is slightly below the limit of human pitch discrimination and therefore represents the preferred pitch resolution of the oscillators. See, for example Backus (1969).
2. An important point to note is that the use of the various modes is not mutually exclusive. That is, it is perfectly possible to be synthesizing an eight-partial tone using additive synthesis, while we have VOSIM, FM, and fixed waveform modes being utilized at the same time. The flexibility of the arrangement is obvious, as is its benefit.
3. The actual form of the pulse is arbitrary. However, since one is trying to control the partial content, one typically wants to generate a pulse with few spurious components (Kaegi and Tempelaars, 1978).

## 6. APPENDIX 1: Synthesizer Register Summary

DEL: Non-zero value implies VOSIM mode. DEL is the average delay, or time-out period, between instances of cycle-mode.

DEV: The % of random deviation in the value DEL. Is used to control degree of non-periodicity in VOSIM sounds.

ENV: The current value of the (as yet unscaled) amplitude envelope. ENV is scaled by VOL, and the product is used to scale the waveform sample to being output.

F_INC: Used for frequency control. F_INC is the increment added to the address of the last waveform sample output, in order to derive that of the next sample.

MOD_INDEX: Controls the amount of modulation that the current oscillator $n$ effects on oscillator $n + 1$. When the msb is zero, mode is FM. When the msb is one, mode is waveshaping.

OP_SEL: Used to select to which of the four audio output busses the oscillator's output should be fed.

VOL: The maximum volume to which the envelope (ENV), and consequently waveform, is to be scaled. VOL is logarithmic scale.

WF_SEL: A value to select from which of the eight waveform buffers the waveform sample is to be selected. Used to change waveforms.

## 7. References and Bibliography

Alles, H. (1978). "A Modular Approach to Building Large Digital Systems." *Computer Music Journal,* Vol. 1, No. 4, pp. 10-13.

Alles, H., & P. di Giugno. (1978). "A One-Card 64-Channel Digital Synthesizer." *Computer Music Journal,* Vol. 1, No. 4, pp. 7-9.

Alonso, S., J. Appleton, & C. Jones. (1976). "A Special Purpose Digital System for Musical Instruction, Composition and Performance." *Computers and the Humanities,* Vol. 10, pp. 209-215.

Backus, J. (1969). *The Acoustical Foundations of Music.* New York: W. W. Norton & Co.

Buxton, W. (1978). *Design Issues in the Foundation of a Computer-Based Tool for Music Composition.* Technical Report CSRG-97. Toronto: University of Toronto.

Buxton, W., & G. Fedorkow. (1978). *The Structured Sound Synthesis Project (SSSP): an Introduction.* Technical Report CSRG-92. Toronto: University of Toronto.

Chowning, J. (1973). "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *Journal of the Audio Engineering Society,* Vol. 21, No. 7, pp. 526-34. Reprinted in *Computer Music Journal,* Vol. 1, No. 2, pp. 46-54.

Fedorkow, G. (1978). *Audio Network Control.* Toronto: M.Sc. Thesis, Dept. of Electrical Engineering, University of Toronto.

Fedorkow, G., W. Buxton, & K. C. Smith. (1977). "A Computer-Controlled Sound Distribution System for the Performance of Electroacoustic Music." *Computer Music Journal,* Vol. 2, No. 3, pp. 33-42.

Grey, J. (1975). *An Exploration of Musical Timbre.* Ph.D. Thesis, Department of Music, Stanford University. Distributed as Dept. of Music Report No. Stan-M-2.

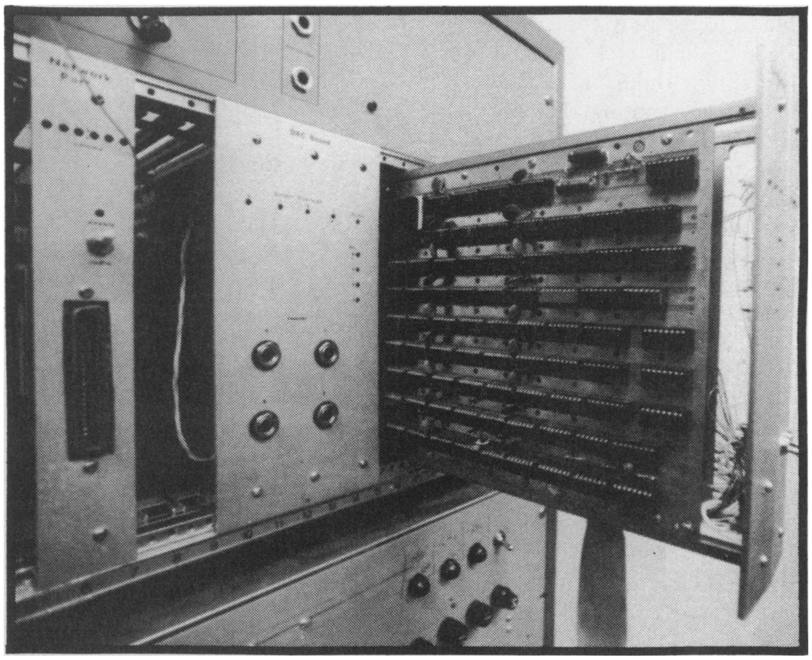
Kaegi, W., & S. Tempelaars. (1978). "VOSIM—A New Sound Synthesis System." *Journal of the Audio Engineering Society,* Vol. 26, No. 6, pp. 418-25.

Le Brun, M. (1977). "Waveshaping Synthesis." Unpublished Manuscript. CCRMA, Stanford University.

Reeves, W., W. Buxton, R. Pike, & R. Baecker. (1978). "Ludwig: an Example of Interactive Computer Graphics in a Score Editor." Presented at the 1978 International Computer Music Conference, School of Music, Northwestern University, Evanston, Illinois.

Risset, J., & M. Mathews. (1969). "Analysis of Musical-Instrument Tones." *Physics Today,* Vol. 22, No. 2, pp. 23-30.

Schottstaedt, B. (1977). "The Simulation of Natural Instrument Tones using Frequency Modulation with a Complex Modulating Wave." *Computer Music Journal,* Vol. 1, No. 4, pp. 46-50.



The SSSP digital synthesizer, detail of the oscillator board.