LAB NOTES

OG93: AN INTERPRETIVE ARPEGGIATION PROGRAMMER & EDITOR

BY: JOHN S. SIMONTON, JR.

One of the major advantages that our hybrid computer/synthesizer system offers is the ability to realize a class of new tricks which for lack of a better term we'll call "keyboard effects". I have in mind new sounds which arise not so much from the timbre of each note, but from the timing and sequence in which the keys are converted to notes and how they're allocated to available output channels.

Using this definition, I suppose that POLY-SPLIT from last time would qualify as a keyboard effect because it affects the way that keys held down are allocated to note-producing output channels. But, ECHO (January-March 1979 Polyphony, page 29) is more specifically what I feel the term should mean because with that program new effects (and at short delay settings, new timbres) arise that would be extremely difficult to accomplish without some means of juggling key activations and how they're assigned to outputs.

Another good example would be the ORGASMATRONIC GLIDE arpeggiation trick that the keyboard encoder and D/A did by themselves (remember?). Hold down a bunch of keys, while scanning, stopped momentarily when it reached one of the down keys and played the note briefly before continuing the scan. When another key was found down, it stopped again to play that one, and so on. Altogether an alright thing that allowed arpeggiations to be played much more rapidly than they could be without electronic assistance.

When we installed the computer in the loop, we lost Orgasmatronic Glide (OG), which maybe was not such a huge sacrifice when considering the power that was gained in the process; but still, I know several folks who mourned the loss because it was an effect that they were using to good purpose in their music.

Here's a terrific replacement. This new program does the same thing that the old OG did, hold down a bunch of keys and it plays them in sequence, but it also gives control that wasn't possible with the old "state machine" version. For instance, it can arpeggiate down-scale as well as up. And it plays staccato or legato. It also allows touch pad control of glide and similar control of the tempo of the arpeggiation.

Great. But not the greatest part, we'll get to that soon.

Enter the program as outlined at the end of the column and start it running, then press down a group of keys. If you've done everything correctly, you should hear a relatively slow down-scale arpeggiation of the notes that you're holding down. When the lowest note has played, the sequence should start again from the highest.

Now let's play with the control some. Here's what the keys mean with OG93 running:
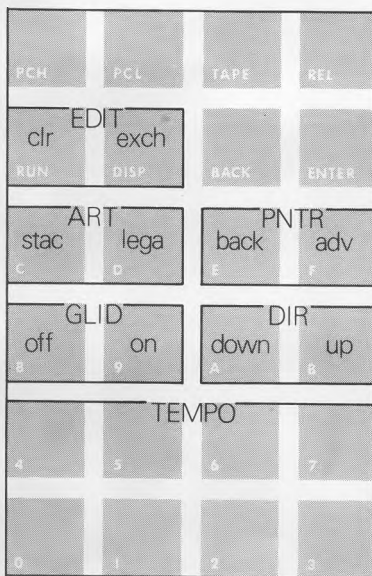
Touching the DIR:UP pad will cause the arpeggiation to change direction from down-scale up to up. GLID:ON turns the glide for the arpeggiation channel on and (you guessed it) GLID:OFF turns it off.

The LEGATO ARTICULATION pad causes the trigger signal to remain high as long as any keys are down so that there will be no re-articulation as one note finishes playing and the next begins. STACCATO ARTICULATION triggers the note the first instant that it plays then releases the trigger.

The TEMPO keys cause the rate of arpeggiation to change from slow (7) to fast (0) over a range from so slow that almost anyone could play the run manually to a rate that's so fast that the sequence begins to take on the texture of a chord (which should give you a clue to one interesting application of OG93 in a piece of music).

If you were an Orgamatronic Glide fan in the first place, we could probably stop here and you'd be completely happy — the program is a lot better than the old manual version. We'd also be stopping before we really got started, because by far the most interesting feature of OG93 is that it's an interpreter that allows us to program a series of arpeggiations and an editor that makes the entry and manipulation of those programs easier.

Each program step contains all of the information that we controlled earlier (glide on and off, up-scale or down, staccato or legato, and one of 8 tempos) and when the program is run, each step will be taken in turn and an arpeggiation of the keys held down performed using the status of the parameters specified by that step. At the end of the program it jumps back to its beginning and the sequence of arpeggiations repeats.

Each step of the program is "written" in exactly the same way that we set the parameters earlier; in fact, as you'll soon realize, you were in effect writing the first step then. The key to forming these steps into programs is the PNTR : BACK/ADV block of pads on the command keyboard. The pointer (PNTR) refers to the program step that you're writing.

One quick example should get the idea across. We'll write a program that sweeps up the keyboard at a moderate tempo, re-articulating each note, followed by a quick legato run down-scale. Program the first step by touching these keys - TEMPO:4, DIR:UP, GLID:OFF, ART:STAC. That takes care of the up part.

Now for the down part, begin by touching PNTR:ADV so the commands that we enter next are "pointed" at the second program step (which is step #1 as shown in the displays, the first step is #0) and touch TEMPO:2, DIR:DOWN, GLID:OFF, ART:STAC. Now hold down a big chord structure to hear the full effect of this dual arpeggiation.

Editing an existing program is simply a matter of pointing to the program step that you want to change and entering the changed parameter. To change the first step (#0) in the example above to a slower tempo, for example, touch PNTR:BACK so the display shows 00 and then touch TEMPO:7 (or whatever).

OG93 can handle programs up to 8 arpeggiations deep and, when you begin stacking that many steps, it's easy to get lost. The EDIT:EXCH key helps here by allowing us to remove the step pointed to from the program and replacing it with an instruction for repeat. By backspacing the pointer to step #1 and touching the EDIT:EXCH pad, we cut the program to just the first step, EDIT:EXCH again and the original program step is back in place, so that the entire program runs again. By stepping through the program and causing it to repeat after the 2nd, 3rd, etc. steps, it's fairly easy to locate where in the program a specific sound is coming from and then make changes there.

As you may surmise from the name, the EDIT:EXCH key causes the program step pointed to to be exchanged with a memory buffer location which is initialized to contain the interpreter's repeat code (00). This implies that this key can also be used to exchange two program steps by pointing first to one and touching EDIT:EXCH and then to the next and again EDIT:EXCH. In fact, this is the case; with one exception. The first step of the program may not be the repeat code 00. If it is, the interpreter will lock up as it reads the first step, finds that it's a repeat, so it reads the first step, and so on. OG93 protects against this by checking to see if you are pointing to the first program step and, if you are, checking to see if you're getting ready to make it 00. If you are, it doesn't complete the exchange. You can get around this if you want to exchange the first step with another by pointing first to the other step first, EXCHanging and then going to step #0 and EXCHanging again.

The final editing key is CLR (clear). Touching this pad clears the program, with the exception of the first step, which remains unchanged for the reasons given above.

For detailed operational information there is no substitute for the liberally commented assembler listing at the end, but let's talk in general terms about how the program works.

We use the MUS1 firmware NOTE to take care of the dynamics of maintaining analog outputs that must be refreshed and to read the AGO keyboard. The list of keys which the firmware returns as being held down is the "arpeggiation list", or the notes to be played. In simplest terms, OG93 does nothing but delay for a period of time which determines tempo and, when the time is up, pulls the next key from the list and plays it as a note. The bulk of the rest of the program checks that we're not yet to the ends of tables (keys down, program steps, etc.) and, if we are, takes care of starting from the beginning again, and controls things like re-setting the tempo timer when it expires and re-articulating between notes when playing staccato.

Most of the arpeggiation program's control information is contained in the single word of memory labeled SCTL (sequence control) in the listing. The 8 bits of that word have these uses:


```
D D D D           D
7 6 5 4           0
│ │ │ │  TEMPO
│ │ └── NOT USED
│ └──── 1=LEGATO
└────── 1=DIN: UP
```
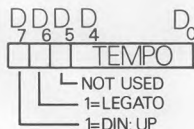
When the interpreter needs the tempo, it extracts it from this word with an AND operation (statement 1440). The status of the control bits D6 and D7 are determined with BIT operations at lines 0790 and 0900.

When one run has finished and the next is due to begin, the interpreter pulls the next program step from memory (the program buffer CSEQ) and after a manipulation which immediately isolates the glide controlling bit, places the program step in the control word SCTL, where the rest of the program accesses it as outlined above. The isolated glide bit is immediately rotated into the most significant bit of the transpose buffer (TTBL) word corresponding to the output channel being used.

```
0010  :********************************
0020  :*                              *
0030  :*      ORGASMATRONIC GLIDE      *
0040  :*                              *
0050  :*  ARPEGIATION PROGRAMMER AND   *
0060  :*            EDITOR             *
0070  :*              BY               *
0080  :*       JOHN S. SIMONTON, JR    *
0090  :*                              *
0100  :*(C) 1979 PAIA ELECTRONICS, INC*
0110  :*                              *
0120  :********************************
0130  :
0140  :****** MONITOR SUBROUTINES ******
0540  :THIS IS THE MAIN PROGRAM LOOP. START BY INITIALIZING THE SYNTHESIZER
0550  :AND CALLING THE QUASH DRIVERS AND AGO KBD READING ROUTINES FROM MUS1
0560  :CHECK TO SEE IF A COMMAND KEY HAS BEEN TOUCHED; AND IF SO, JUMP TO
0570  :SUBROUTINE TO DETERMINE THE COMMAND AND EXECUTE IT. DETERMINE THE
0580  :POINTER FOR THE OUTPUT CHANNEL AND JUMP TO SUBROUTINE FOR ORG. GLIDE
0590  :PROCESSING. ON RETURN, LOOP.
*
*1000LLL                     0600  :
1000-  20 21 1D   JSR        0610  :       JSR INIT    :MUS1 SYNTH INIT ROUTINE
1003-  20 2B 1D   JSR        0620  :LOOP JSR NOTE      :QUASH DRIVERS AND READ AGO
1006-  20 00 1F   JSR        0630  :      JSR DECD     :PIEBUG READ COMMAND KBD
1009-  B0 03      BCS        0640  :      BCS HERE     :IF NO NEW KEY TOUCHED, SKIP NEXT
100B-  20 00 11   JSR        0650  :      JSR CMND     :CALL COMMAND DECODER
100E-  A0 0F      LDY        0660  :HERE LDY 0F        :POINTER TO ORG. GLIDE OUTPUT CHANNEL
1010-  20 16 10   JSR        0670  :      JSR STAR     :CALL ORG. GLIDE PROGRAM
1013-  4C 03 10   JMP        0680  :      JMP LOOP     :LOOP TO CONTINUE
                             0690  :
                             0700  :FIRST THE TIMER IS TESTED AND IF NOT TIME FOR THE NEXT NOTE TO BE
                             0710  :PROCESSED THE STACCATO CONTROL BIT IS CHECKED AND IF CLEAR
                             0720  :(STACCATO) BRANCH IS TAKEN TO DE-TRIGGER NOTE IN OUTPUT
                             0730  :BUFFER. IF LEGATO MODE, EXIT IS IMMEDIATE
                             0740  :
1016-  24 E7      BIT        0750  :STAR BIT *KTBL+07  :ARE THERE ANY AGO KEYS DOWN?
1018-  50 1E      BVC        0760  :      BVC SINT     :NO KEYS, BRANCH TO RE-INIT ARP. POINTER
101A-  C6 72      DEC        0770  :      DEC *TIMR    :OTHERWISE, DECREMENT THE TIMER
101C-  30 05      BMI        0780  :      BMI ADYA     :IF EVENT TIME, BRANCH
101E-  24 74      BIT        0790  :      BIT *SCTL    :OTHERWISE CHECK FOR STACCATO AND IF TRUE....
1020-  50 46      BVC        0800  :      BVC CLRN     :BRANCH TO CLEAR TRIGGER FROM OUTPUT NOTE
1022-  60         RTS        0810  :      RTS          :OTHERWISE, RETURN WITHOUT CLEARING TRIGGER
```
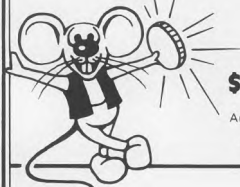
REVIEWS
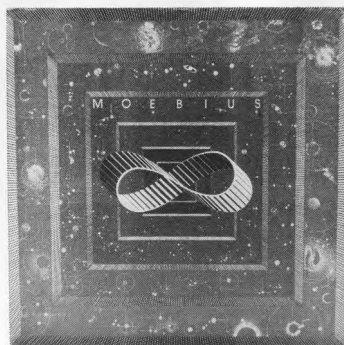
Moebius
Moonwind Records
MW 33801

This album represents an original foray into an area which has been dealt with so often in cliche that its real possibilities have been largely ignored: pop synthesis. Taking up where Brian Eno and Tangerine Dream left off, "Moebius" combines some of the sonorities of these groups with the more minimalistic production values of groups like The Normal and Suicide. Add a dash of biting lyrics and a mock serious approach, and you have what amounts to the missing link between the more traditional schools of pop synthesis and the outer fringes of new wave. The result is a refreshing LP whose bold approach makes you wonder why no one has spotted this fertile turf before.

If "Moebius" straddles such broad musical boundaries, it is in no small part due to its personnel, all of whom are well grounded in synthesis. They are: Douglas Lynner, editor of Synapse and former member of LEM, Bryce Robbley, also of LEM, and Synapse correspondent Steve Roach. Their knowledge of the territory makes for an album which is not only broad in scope, but sonically inventive using synthesizers, drums and violin as well as effects such as crinkling paper, and processed voices to augment its aural images.

For my money, the best cuts are those which bear the closest ties with the new wave: tunes like "Light My Fire", where crazed vocals combine with a bassline gone bonkers to create an atmosphere which bears only a cosmic resemblance to the original; and "Clone Zone", a cut whose cryptic lyrics, naked sequencer and tongue in cheek production conjure up a time when mindless clones stalk the Earth like giant Xerox copies in search of their originals. Of the rest of the material, the most accessible cuts are "Urth", and the title cut which uses sequencers to good rhythmic and harmonic effect. For more traditional tastes, "Prophecy" is a rolling instrumental reminiscent of Eno's dreamier moods.

There is a tendency towards overproduction which mars some of the tunes. It's as if, in the process of rejecting the production cliches of pop music, Moebius has not yet zeroed in on what its own values are. And a couple of the tunes are simply too long. But for all its flaws, this is an interesting album whose innovative approach proves that the synthesizer has far from realized its potential in pop.

—Melodie Bryant

```
0820  :
0830  :IF IT'S TIME FOR A NOTE TO BE PROCESSED, THE POINTER TO THE INPUT
0840  :BUFFER IS ADVANCED (EITHER FORWARD OR BACKWARD) AND IF THERE IS NO
0850  :MORE BUFFER LEFT WE DROP THROUGH TO ADVANCE THE POINTER TO THE SEQUENCE
0860  :BUFFER TO GET THE NEXT SET OF GLIDE PARAMETERS.  IF WE ARE NOT YET
0870  :TO THE END OF THE IN BUFFER, WE BRANCH OUT TO RESET THE TIMER, ETC.
0880  :
1023-  A6 73      LDX   0890  ADVA LDX *PNTR   :GET POINTER TO INPUT BUFFER
1025-  24 74      BIT   0900       BIT *SCTL   :CURRENTLY ARPEGGIATING UP?
1027-  10 05      BPL   0910       BPL DOWN    :NO, BRANCH TO DO DOWN
1029-  CA         DEX   0920       DEX         :TO GO UP-SCALE, DECREMENT POINTER
102A-  30 07      BMI   0930       BMI SADV    :IF POINTER NOW <0, BRANCH
102C-  10 26      BPL   0940       BPL STIM    :STILL IN RANGE, BRANCH ALWAYS
102E-  E8         INX   0950  DOWN INX         :DOWN-SCALE, INCREMENT POINTER
102F-  E0 08      CPX   0960       CPX 08      :OUT OF RANGE?
1031-  D0 21      BNE   0970       BNE STIM    :STILL IN RANGE, BRANCH
                        0980  :
                        0990  :IF WE GET HERE (SADV) IT MEANS THAT WE HAVE PLAYED ALL OF THE KEYS
                        1000  :THAT WERE DOWN AND HAVE REACHED THE END OF THE INPUT BUFFER
                        1010  :NOW IT'S TIME TO GET THE NEXT ENTRY FROM THE CONTROL SEQUENCE.
                        1020  :WE TEST TO SEE IF WE ARE AT THE END OF THE SEQUENCE AND IF SO THE
                        1030  :POINTER IS RE-INITIALIZED. OTHERWISE, THE COMMAND IS FETCHED AND IF
                        1040  :ZERO IT MEANS THAT IT IS THE END OF THE SEQUENCE AND THE POINTER
                        1050  :IS ALSO REINITIALIZED
                        1060  :
1033-  A6 76      LDX   1070  SADV LDX *SPNT   :GET CONTROL SEQUENCE POINTER
1035-  CA         DEX   1080       DEX         :POINT TO NEXT SEQUENCE ENTRY
1036-  10 02      BPL   1090       BPL GSEQ    :IF NOT TO END, BRANCH
1038-  A2 07      LDX   1100  SINT LDX 07      :RE-INIT SEQUENCE POINTER
103A-  86 76      STX   1110  GSEQ STX *SPNT   :SAVE SEQUENCE POINTER
103C-  B5 77      LDA   1120       LDA *CSEQ,X  :GET COMMAND FROM CONTROL SEQ.
103E-  F0 F8      BEQ   1130       BEQ SINT    :ZERO ENDS THE SEQUENCE, BRANCH
                        1140  :
                        1150  :A NEW COMMAND FROM THE SEQUENCE. FIRST USE IT TO SET OR CLEAR THE
                        1160  :THE GLIDE CONTROL BIT FROM THE TRANSPOSE BUFFER.  IN THE PROCESS,
                        1170  :THE NEW COMMAND IS SHIFTED ONE BIT TO THE LEFT; WHICH MULTIPLIES
                        1180  :THE TEMPO VARIABLE BY 2 AND SHIFTS THE UP/DOWN AND LEGA/STACC BITS
                        1190  :INTO MORE EASILY TESTED POSITIONS.
                        1200  :
1040-  85 74      STA   1210  GLID STA *SCTL   :SAVE SEQUENCE ENTRY IN CONTROL BUFFER
1042-  B9 C0 00   LDA   1220       LDA TTBL,Y  :GET THE CURRENT TRANSOSE BUFFER ENTRY
1045-  2A         ROL   1230       ROL         :ROTATE GLIDE BIT TO CARRY
1046-  06 74      ASL   1240       ASL *SCTL   :ROTATE CONTROL WORD GLIDE TO CARRY
1048-  6A         ROR   1250       ROR         :ROTATE CARRY TO GLIDE BIT
1049-  99 C0 00   STA   1260       STA TTBL,Y  :THEN RETURN TO TRANSPOSE BUFFER
                        1270  :
                        1280  :THIS LITTLE ROUTINE DETERMINES WHETHER SCAN IS UP OR DOWN AND
                        1290  :INITIALIZES THE POINTER TO THE PROPER VALUE
                        1300  :SKYP-SET KEY POINTER
                        1310  :
104C-  A2 07      LDX   1320  SKYP LDX 07      :PREPARE FOR ARP. UP INITIAL POINTER
104E-  24 74      BIT   1330       BIT *SCTL   :CHECK COMMAND BUFFER - ARP. UP?
1050-  30 02      BMI   1340       BMI STIM    :YES, BRANCH
1052-  A2 00      LDX   1350       LDX 00      :NO, ARP. DOWN INITIAL POINTER
                        1360  :
                        1370  :NOW THE ROUTINE TO RESET THE TIMER. SINCE ALL KEY POINTER MANIPULATIONS
                        1380  :WIND UP AT THIS POINT, THE FIRST INSTRUCTION IS TO SAVE THIS POINTER
                        1390  :THE TIMER VALUE IS EXTRACTED FROM THE CONTROL WORD SCTL
                        1400  :STIM-SET TIMER
                        1410  :
1054-  86 73      STX   1420  STIM STX *PNTR   :SAVE INPUT BUFFER POINTER
1056-  A9 1F      LDA   1430       LDA 1F      :PREPARE MASK AND
1058-  25 74      AND   1440       AND *SCTL   :GET THE TIMER (TEMPO) VALUE
105A-  85 72      STA   1450       STA *TIMR   :AND SAVE IN THE TIMER VARIABLE
                        1460  :
                        1470  :NOW WE GET THE CURRENT NOTE OF INTEREST FROM THE INPUT BUFFER
                        1480  :AND IF THE KEY IS NOT DOWN, A CHECK IS MADE TO SEE IF ANY KEYS
                        1490  :ARE DOWN. IF NONE ARE, THE TIMER IS TRICKED INTO TIMING OUT THE
                        1500  :NEXT TIME THROUGH WHICH WILL THEN RESULT IN THE WHOLE COMMAND
                        1510  :SEQUENCE FOLLOWING SYSTEM RESET
                        1520  :
105C-  B5 E0      LDA   1530       LDA *KTBL,X  :GET THE CURRENT KEY FROM INPUT BUFFER
105E-  D0 0D      BNE   1540       BNE BOUT    :IF ZERO, NO KEY - BRANCH
1060-  24 E7      BIT   1550       BIT *KTBL+07 :ARE ANY KEYS DOWN?
1062-  70 BF      BVS   1560       BVS ADVA    :YES, BRANCH
1064-  A9 01      LDA   1570       LDA 01      :NO, PREPARE TO MAKE TIMER RUN OUT
1066-  85 72      STA   1580       STA *TIMR   :NEXT PASS THROUGH
1068-  B9 D0 00   LDA   1590  CLRN LDA NTBL,Y  :GET THE CURRENT OUTPUT NOTE
106B-  29 BF      AND   1600       AND 0BF     :CLEAR THE TRIGGER FLAG
106D-  99 D0 00   STA   1610  BOUT STA NTBL,Y  :AND REPLACE IN OUTPUT BUFFER
1070-  60         RTS   1620       RTS         :RETURN
                        1630  :
```

```
                              1640  :NOW SOME TEMPORARY LOCATIONS AND THEIR INITIAL STATES
                              1650  :
1071  00                      1660  TEMP  .HS 00
1072  01                      1670  TIMR  .HS 01
1073  08                      1680  PNTR  .HS 08
1074  C4                      1690  SCTL  .HS C4
1075  07                      1700  PPNT  .HS 07
1076  07                      1710  SPNT  .HS 07
1077  000000                  1720  CSEQ  .HS 000000000000E404
                              1740        .OR 10E8
                              1750  STUP  .HS 402004
                              1770        .OR 1100
                              1790  :THIS IS THE COMMAND KEY DECODING AND SEQUENCE EDITING SUBROUTINE
                              1800  :# OF COMMAND KEY IS IN Y REGISTER
                              1810  :
1100-  A6 75      LDX         1820  CMND LDX *PPNT    :GET THE EDITORS POINTER TO COMMAND SEQ
1102-  B5 77      LDA         1830       LDA *CSEQ,X  :GET THE COMMAND POINTED TO (IN ACC, DON'T FORGET)
1104-  C0 10      CPY         1840       CPY 10       :IS KEY 10 - CLEAR COMMAND SEQUENCE
1106-  F0 0F      BEQ         1850       BEQ CLR      :YES, BRANCH
1108-  90 1F      BCC         1860       BCC CNXT     :NO, IT'S LESS THAN "F", BRANCH
                              1870  :
                              1880  :THE KEY IS 11 OR GREATER. EXCHANGE THE COMMAND POINTED TO WITH
                              1890  :TEMPORARY STORAGE LOCATION TEMP. NOTE THAT THIS CAN BE USED TO
                              1900  :EXCHANGE TWO OR MORE COMMANDS IN THE SEQUENCE
                              1910  :
110A-  A4 71      LDY         1920       LDY *TEMP    :GET THE COMMAND IN THE TEMPORARY BUFFER
110C-  D0 04      BNE         1930       BNE ELP0     :IS THE COMMAND FROM TEMP A 0? NO, BRANCH
110E-  E0 07      CPX         1940       CPX 07       :POINTING TO FIRST COMMAND?
1110-  F0 2B      BEQ         1950       BEQ RTN      :YES, BRANCH. DON'T WRITE ZERO AS FIRST COMMAND
1112-  94 77      STY         1960  ELP0 STY *CSEQ,X  :PUT COMMAND IN THE SEQUENCE SLOT POINTED TO
1114-  85 71      STA         1970       STA *TEMP    :AND THEN SAVE OLD COMMAND IN THE TEMP LOCATION
1116-  60         RTS         1980       RTS          :THEN RETURN
                              1990  :
                              2000  :THE KEY IS "10", CLEAR THE COMMAND SEQUENCE. NOTE THAT THE FIRST
                              2010  :ENTRY IN THE SEQUENCE IS NOT CHANGED.
                              2020  :
1117-  A2 07      LDX         2030  CLR  LDX 07       :SET POINT TO FIRST SEQUENCE ENTRY
1119-  86 75      STX         2040       STX *PPNT    :AND SAVE IT
111B-  CA         DEX         2050       DEX          :DECREMENT THE POINTER(SKIP FIRST ENTRY)
111C-  A9 00      LDA         2060       LDA 00       :AND GET READY
111E-  8D 20 18   STA         2070       STA DISP     :ZERO THE DISPLAYED EDITOR POINTER
1121-  85 71      STA         2080       STA *TEMP    :AND THE EXCHANGE REGISTER
1123-  95 77      STA         2090  CLLP STA *CSEQ,X  :ZERO THE SEQUENCE ENTRY
1125-  CA         DEX         2100       DEX          :AND POINT TO NEXT ENTRY
1126-  10 FB      BPL         2110       BPL CLLP     :SOME LEFT, LOOP
1128-  60         RTS         2120       RTS          :RETURN
                              2130  :
                              2140  :NOW WE TEST FOR "E" OR "F", BACKSPACE OR ADVANCE THE EDITOR'S
                              2150  :EDITOR'S POINTER TO THE COMMAND SEQUENCE. NOTE THAT INCREMENTING THE
                              2160  :POINTER PRODUCES A BACKSPACE.
                              2170  :
1129-  C0 0E      CPY         2180  CNXT CPY 0E       :IS KEY "E" OR "F"?
112B-  90 18      BCC         2190       BCC STMP     :NEITHER AND LESS THAN "E", BRANCH FOR NEXT TEST
112D-  F0 0F      BEQ         2200       BEQ BACK     :IT'S "E", BRANCH TO BACKSPACE
112F-  CA         DEX         2210       DEX          :IT'S "F", ADVANCE THE POINTER
1130-  30 0B      BMI         2220       BMI RTN      :AND IF OUT OF RANGE, BRANCH TO LEAVE IMMEDIATELY
1132-  86 75      STX         2230  COUT STX *PPNT    :SAVE NEW POINTER
                              2240  :
                              2250  :IN THIS SECTION THE POINTER (WHICH IS 07 FOR THE START OF THE SEQUENCE
                              2260  :AND 00 AT THE END) IS CONVERTED TO AN INCREASING NUMBER FROM 0-7 FOR
                              2270  :DISPLAY PURPOSES.
                              2280  :
1134-  8A         TXA         2290       TXA          :POINTER TO THE ACCUM. FOR A CALCULATION
1135-  38         SEC         2300       SEC          :PREPARE FOR A SUBTRACTION
1136-  E9 08      SBC         2310       SBC 08       :TWO'SD COMPLEMENT
1138-  49 FF      EOR         2320       EOR 0FF      :COMPLEMENT OF THAT
113A-  8D 20 18   STA         2330       STA DISP     :SHOW VALUE IN THE DISPLAYS
113D-  60         RTS         2340  RTN  RTS          :RETURN
                              2350  :
                              2360  :BACKSPACE POINTER AND MAKE SURE IT IS STILL IN RANGE, THEN BRANCH
                              2370  :
113E-  E8         INX         2380  BACK INX          :BACKSPACE THE POINTER
113F-  E0 08      CPX         2390       CPX 08       :OUT OF RANGE?
1141-  F0 FA      BEQ         2400       BEQ RTN      :YES, BRANCH TO LEAVE IMMEDIATELY
1143-  D0 ED      BNE         2410       BNE COUT     :NO, BRANCH ALWAYS TO SAVE POINTER, ETC.
                              2420  :
                              2430  :IF THE KEY IS ONE OF THE TEMPOS, ADD 1 (0 TEMPO NOT ALLOWED) AND
                              2440  :FIT IT INTO THE CONTROL SEQUENCE ENTRY POINTED TO
                              2450  :
1145-  C0 08      CPY         2460  STMP CPY 08       :TEMPO KEY?
1147-  B0 0A      BCS         2470       BCS SGLD     :NO, BRANCH
1149-  C8         INY         2480       INY          :YES, ADD 1 TO KEY #
114A-  29 F0      AND         2490       AND 0F0      :MASK PRESENT TEMPO IN COMMAND TO ZERO
114C-  95 77      STA         2500       STA *CSEQ,X  :SAVE CONTROL FLAGS IN CSEQ TEMPORARILY
114E-  98         TYA         2510       TYA          :BRING NEW TEMPO TO ACC
```
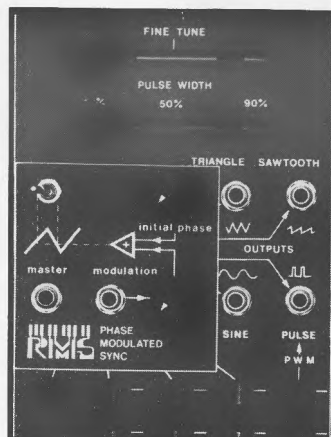

What can phase-modulated sync do for your 2600 ?

For a detailed explanation of phase-modulated sync and our other modifications (like voltage-controlled Q and a comparator) for your 2600, write to Rivera Music Services. 48 Brighton Ave. #11. Boston, Mass. 02134. or call us at 617-782-6554.

Rivera Music Services

Advertiser Index

When you write or order from our advertisers, be sure to tell them that you saw their products in

POLYPHONY

```
114F-  15 77     ORA      2520        ORA *CSEQ,X   :COMBINE WITH OLD CONTROL FLAGS
1151-  D0 1A     BNE      2530        BNE SAVA      :BRANCH ALWAYS
                          2540     :
                          2550     :NOW A SERIES OF TESTS WHICH RESULT IN THE CARRY BIT BEING SET OR
                          2560     :CLEAR. A SERIES OF ROTATES BRINGS THE CARRY TO THE APPROPRIATE BIT
                          2570     :IN THE COMMAND WORD
                          2580     :
1153-  2A        ROL      2590 SGLD ROL            :ROTATE THE GLIDE COMMAND BIT TO CARRY
1154-  08        PHP      2600        PHP           :AND SAVE THE CARRY ON THE STACK
1155-  C0 09     CPY      2610        CPY 09        :IS KEY GLIDE ON OR OFF?
1157-  F0 12     BEQ      2620        BEQ ROT1      :9-GLIDE ON, BRANCH
1159-  90 10     BCC      2630        BCC ROT1      :8-GLIDE OFF, BRANCH
                          2640     :
                          2650     :THE KEY WAS NEITHER GLIDE ON NOR OFF, TEST FOR DIRECTION UP OR DOWN
                          2660     :
115B-  28        PLP      2670 SMOD PLP            :GET THE OLD GLIDE BIT FROM THE STACK
115C-  2A        ROL      2680        ROL           :ROTATE DIRECTION BIT TO CARRY
115D-  08        PHP      2690        PHP           :SAVE IT ON STACK
115E-  C0 0B     CPY      2700        CPY 0B        :IS KEY UP OR DOWN?
1160-  F0 08     BEQ      2710        BEQ ROT2      :B-UP, BRANCH
1162-  90 06     BCC      2720        BCC ROT2      :A-DOWN, BRANCH
                          2730     :
                          2740     :THE KEY HAS TO BE C OR D (STACCATO OR LEGATO)
                          2750     :
1164-  28        PLP      2760 SDIR PLP            :GET THE OLD DIRECTION BIT
1165-  2A        ROL      2770        ROL           :STAC/LEGA BIT TO CARRY
1166-  08        PHP      2780        PHP           :SAVE IT
1167-  C0 0D     CPY      2790        CPY 0D        :CARRY SETS IF KEY IS "D" - LEGATO
1169-  6A        ROR      2800        ROR           :ROTATE COMMAND WORD BACK INTO PLACE
116A-  6A        ROR      2810 ROT2 ROR
116B-  6A        ROR      2820 ROT1 ROR
116C-  28        PLP      2830        PLP           :WASTE A STACK SLOT TO COMPENSATE
116D-  95 77     STA      2840 SAVA STA *CSEQ,X    :SAVE THE COMMAND WORD IN THE SEQUENCE
116F-  60        RTS      2850        RTS           :RETURN
                          2860     :
                          2870 END  .EN
```