THE POLYPHONIC SYNTHESIZER

By
John S. Simonton, Jr.

LAB NOTES

We've come a long way over the last year in terms of developing a series of digitally interfaced modules that will allow computer control of music synthesizers. I suppose that the time has come to look at tieing them all together, with the computer, and begin doing interesting things.

I had wanted to start with "the ultimate sequencer programs" but am not completely happy with them yet. They still need a little polishing.

Instead, we'll start with what should be another popular system:

THE POLYPHONIC SYNTHESIZER
Which is a much simpler job than the ultimate sequencer.

I would like to go through the system showing specific ways to do things for a variety of manufacturers equipment but that just isn't practical. Instead, we'll look at a completely PAIA based system and assume that if you are using different equipment you are familiar enough with it to make whatever changes are necessary.

Oh, one more thing before we begin, be sure that you understand that there are a wide variety of ways to do poly-phonic synthesizers. This is only one of them. I hope that the algorhythm used here works for you. It's one of many, some with sort of special quirks that make them useful in certain situations but difficult to work with generally - This seems to be good general purpose way. Ready? We have lots to do and little space and time; here we go.

THE HARDWARE

Most of the hardware that we'll be using has been described here over the last year (or so). For the controller portion of this system we'll need:

1) AN ENCODED KEYBOARD
8782 or EK-3 retro-fitted equivalent
2) A COMPUTER
An 8700 in it's minimum configuration will run the programs that we'll list. A cassette interface system is useful to the point of being almost mandatory. We'll show some new panels and stuff to make it all pretty.
3) DIGITAL/ANALOG CONVERTER AND SAMPLE AND HOLDS
the 8780/8781 system.

And, of course, we'll also need as much synthesizer as we think is necessary.

With all of the items listed, various wiring schedules have been mentioned for doing various non-computer things. We now need to establish some standards for this new use, a computer based polyphonic system.

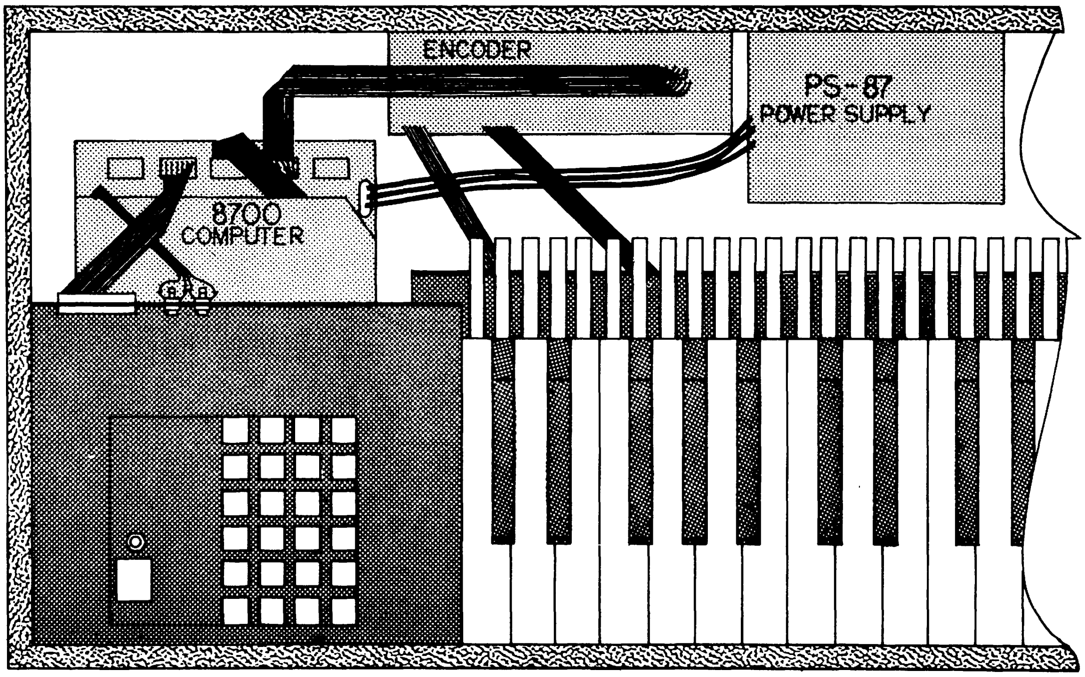
If we choose wisely, we should come up with a standard that has plenty of room for future growth. Some consideration has gone into the system which follows and I believe that it will serve our needs for some time to come.

Many of you will already have much of this wiring done, as much of it is simply an extension of what we've done before. Check carefully to be sure your wiring is to this new standard.

THE KEYBOARD

Let's go ahead and configure this system from the beginning so that the computer fits in the synthesizer cases that we've been using. All of the parts will fit in the case like this:

figure 1. computer/synthesizer sub-module placement.



PAIA 8700 COMPUTER, POWER SUPPLY AND KEYBOARD ENCODER
RETRO-FIT TO 4700 OR 8700 SERIES KEYBOARD.

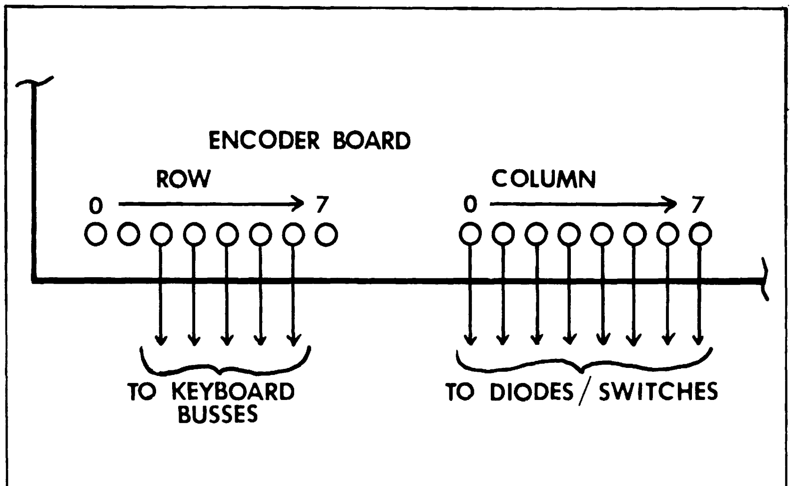


FIG. 2

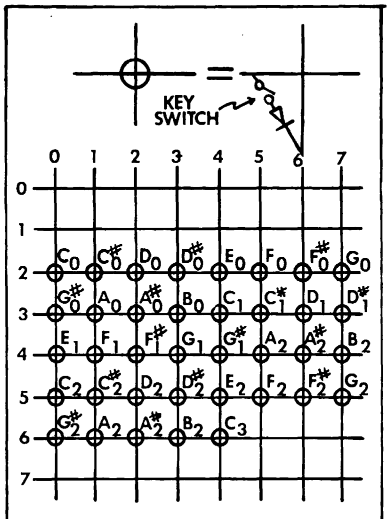

KEY SWITCH
0 1 2 3 4 5 6 7
C0 C#0 D0 D#0 E0 F0 F#0 G0
G#0 A0 A#0 B0 C1 C#1 D1 D#1
E1 F1 F#1 G1 G#1 A2 A#2 B2
C2 C#2 D2 D#2 E2 F2 F#2 G2
G#2 A2 A#2 B2 C3

FIG.3: The Keyboard as Matrixed

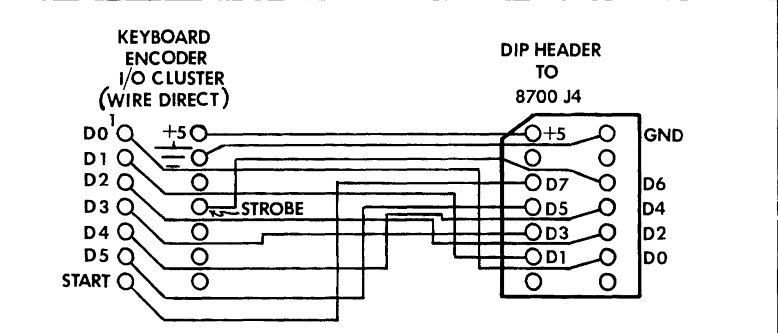


FIG. 4

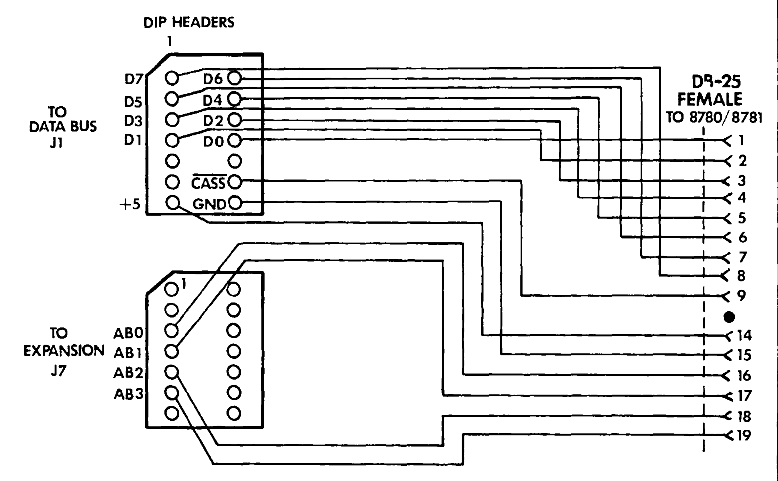


FIG.5: COMPUTER TO 8780/8781

At this stage you may have more dis-assembly to do than assembly. Particularly, the old control panel of the keyboard is removed to make room for the computer and any unregulated supply that was powering your keyboard encoder is replaced with a PS-87 which supplies all digital power for the entire system. This is going to give you a few parts for your "bench stock", the old power supply components and a couple of push-buttons, but some of the parts we will be re-using. Don't throw anything away.

KEYBOARD TO ENCODER CONNECTIONS

Maximum useability of the system would seem at first to depend on where the AGO keyboard switches appear in the key matrix. We want them in the middle so that we have as much room to transpose down in pitch as we do for up-scale transpositions. Some 8782 instructions had the keyboard placed 8 switch positions below where it should be for this ideal. The "column" connections are fine, but the "row" connections on these keyboards will need to be "slid up one" so that they conform to the configuration as shown in figure 2.

This will place the keyboard more or less in the middle of the matrix as shown in figure 3. This is really a fine point, and the system will work OK in most applications almost no matter where in the matrix the keys are, but go ahead and change now so that you won't be limited in the future.

ENCODER MODIFICATIONS

We don't need any of the "trick" things that we used when we didn't have a computer (the orgasmatronic glide circuit, etc.), just the bare-bones encoder. You may remove all push-buttons slide switches, pots etc.; most of these will come out when you remove the old front panel.

ENCODER TO COMPUTER

If your system previously had a DB-25 female connector tied to the output of the encoder, desolder it (carefully - whistling may make the job seem easier). In place of the DB-25 connector, we now need to terminate the output of the encoder in a DIP header that will mate with the INPUT PORT #1 (J4) connector on the rear edge of the 8700 computer board.

These connections should be made as in figure 4.

These connections should also be made carefully and the DIP header pins well heat-sinked to prevent melting the plastic header. NOTE that while many of the non-computer applications used the $\overline{\text{STROBE}}$ line to trigger the D/A, here we ignore this line and instead use the STROBE as the seventh data bit (D6) of the interface.

Similarly, the encoder's START line becomes the 8th data bit (D7).

Also, you will notice that power to

the encoder is picked up through this connection from the 8700 itself.

COMPUTER TO SYNTHESIZER HEAD

So that our resulting system can be easily broken down into two separate units (computer/keyboard and synthesizer head), this is the place to use the DB-25 connector that was salvaged from the old keyboard front panel.

Connections should be made between the female DB-25 connector and a pair of DIP headers like those in figure 5.

NOTE that the first header (P2) provides data lines and the $\overline{\text{CASS}}$ select signal (our 8780/8781 shares this output structure) while the second connector (P3) provides the address lines required by the QuASH.

8780/8781 WIRING

The male DB-25 connector that terminates the cable to the 8781 is wired in what is essentially an expanded version of our previous standard so that here you are faced more with adding wires than re-arranging them.

Connect these elements together as in figure 6.

This wiring schedule is examined in detail in the 8781 QuASH assembly manual. An important thing to notice here is the way the grounds are handled. Note that the ⏚ (ground) pin on the rear of the 8780 board serves as the central ground for both analog (synthesizer) and digital power distribution. This grounding scheme is important to prevent ground loop problems and should be followed exactly. This entire 8780/8781 assembly should be mounted in the synthesizer head cabinet.

FINAL ASSEMBLY

Finally, make arrangements for physically mounting the computer in the keyboard case by first mounting the computer to a suitable front panel as shown. (See figure 7)*

And don't forget to provide a socket at the 8700's expansion connector (J7) or to mate P3 with this socket before assembling the computer/front panel. If the cassette interface is being used, terminate the input and output lines in miniature phone jacks as shown in figure 8.

Plug all the connectors together and you should be ready to load a program.

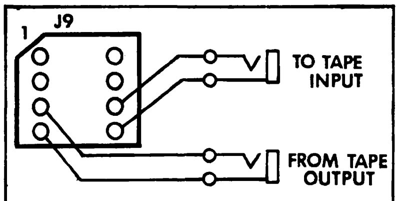


FIG.8: CASSETTE CONNECTIONS

THE PROGRAM

The polyphonic program that we'll be using is called simply:

POLY 1.0

This program supports up to 8 output channels the way that it is written and can be easily modified to provide for more.

POLY 1.0 allocates synthesizer resources to keyboard requirements using this algorhthm:

1) Output all notes appearing in the output buffer area (NTABLE) after adding the corresponding transposing figure from TTABLE. Go to 2.

2) Wait for keyboard scan to start and place a list of all keys currently being held down in the input buffer area (KTABLE). When buffer full or scan complete go to 3.

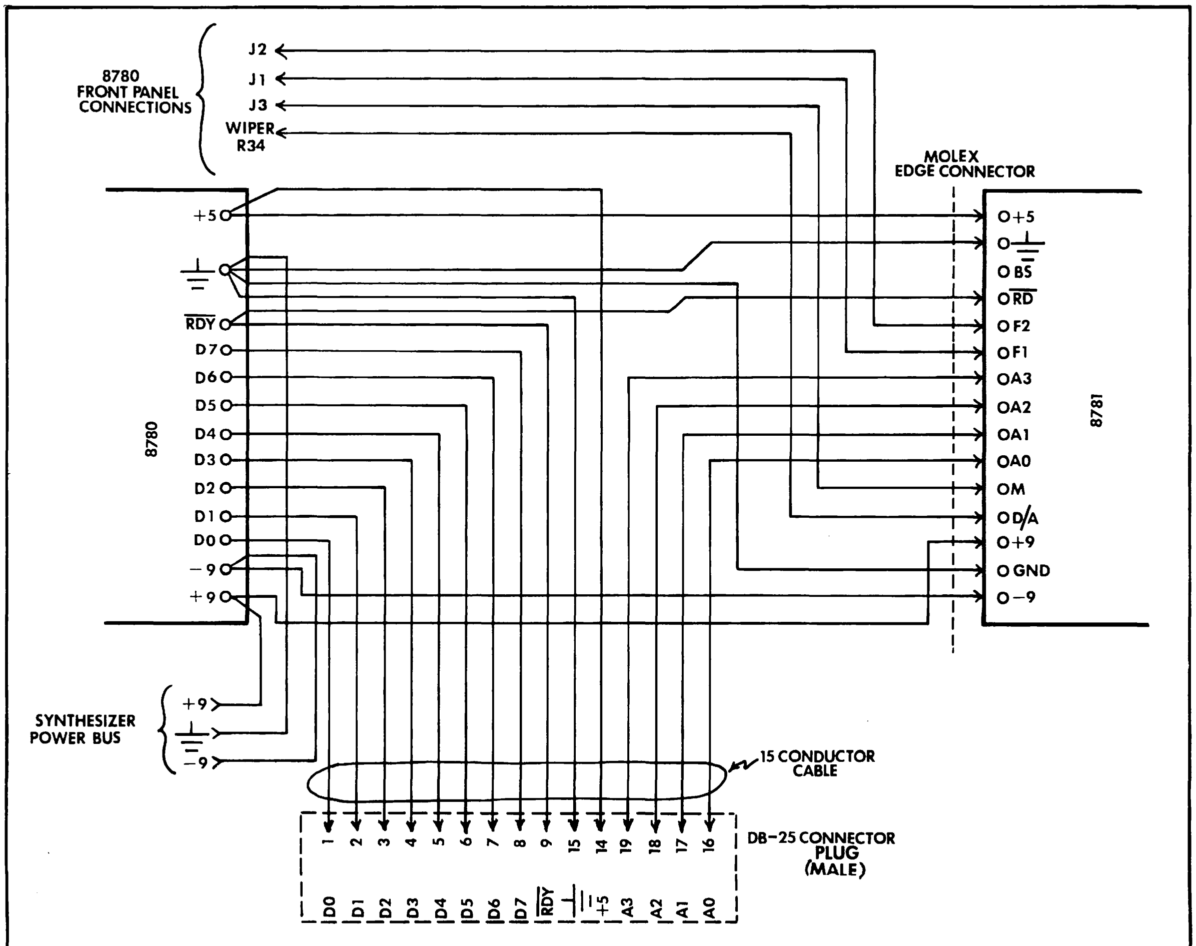
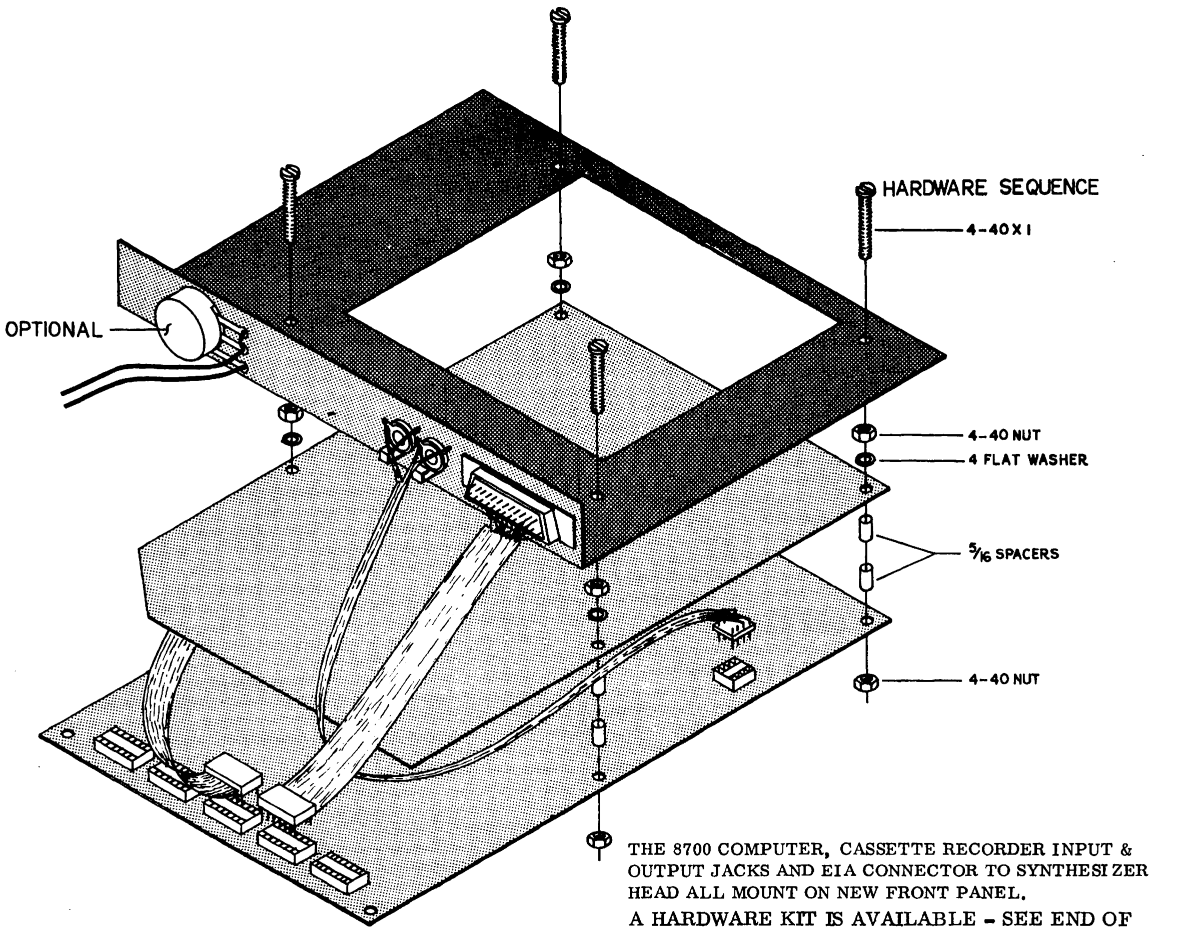


FIG.6



THE 8700 COMPUTER, CASSETTE RECORDER INPUT & OUTPUT JACKS AND EIA CONNECTOR TO SYNTHESIZER HEAD ALL MOUNT ON NEW FRONT PANEL.

A HARDWARE KIT IS AVAILABLE - SEE END OF ARTICLE

FIG. 7

3) Clear the trigger flags (D6) of all notes in NTABLE (the output buffer).
4) Compare each entry in the input buffer (KTABLE) to each entry in the output buffer (NTABLE). If they are the same, set the trigger bit of the NTABLE entry and eliminate (zero) the entry from KTABLE. If all available outputs are used, or if all keys down find a home go to 1.
5) Place the remaining input buffer entries in output buffer locations which do not currently correspond to a down key (those in which D6 is cleared). When all input data has been placed or all channels available have been used go to 1.

There are a number of subtle implications here and unfortunately not enough space to cover them all.

A couple of really important ones are that if we think of "new" notes as ones corresponding to keys that were just pressed, this method tries to place those new notes in output channels which at some point in the past were already producing those notes.

This prevents a string of identical eighth notes (for example) from being assigned to different outputs each time they're used. Notes, once assigned, tend to stay assigned regardless of other keyboard activity - they don't move around in a totally unpredictable fashion as with some analog multi-note keyboards.

It also means that once the number of output channels available is "used up" by down keys that need to be placed, all other keys that are down are simply ignored (this is exactly what you want).

One important aspect of the above is that the program must "know" how many output channels are available to it, otherwise there is the possibility that notes may be assigned to non-existant channels (ones that have no corresponding hardware, not too bad in itself) and further (the really bad part) future activations of the note will be assigned <u>again</u> to these non-existant outputs - producing "dead" synthesizer keys that seem not to be doing anything.

Memory location $00EA contains the number of synthesizer channels available, more on this shortly.

THE PROGRAM

Shown on the next page is a dis-assembled listing of POLY 1.0.

Because, again, of space limitation we cannot re-print a fully documented version of POLY 1.0. It is supplied with the assembly and using manuals for the 8781 QuASH.

POLY 1.0 is also available in 8700 compatible cassette-tape form for $4.00.

LOADING AND INITIALIZING POLY 1.0

If you have a cassette interface on your 8700 and the POLY 1.0 tape, loading is simply a matter of connecting your tape recorder to the cassette input connectors on the 8700 and loading the tape using the following entry sequence:

0-0-0-0-0-0-F-F-0-0-1-1-TAPE

If you don't have the CS-87 option, you must enter the code manually from the 8700 keyboard.

POLY 1.0

By John S. Simonton, Jr.

```
06-   A9 00       LDA   #$00          69-   A6 E9       LDX   $E9
08-   A2 18       LDX   #$18          6B-   B4 CF       LDY   $CF,X
0A-   95 CF       STA   $CF,X         6D-   F0 1D       BEQ   $088C
0C-   CA          DEX                 6F-   A2 09       LDX   #$09
0D-   D0 FB       BNE   $080A         71-   CA          DEX
0F-   A2 08       LDX   #$08          72-   F0 F1       BEQ   $0865
11-   B5 D7       LDA   $D7,X         74-   98          TYA
13-   18          CLC                 75-   55 D7       EOR   $D7,X
14-   75 DF       ADC   $DF,X         77-   0A          ASL
16-   8D 00 09    STA   $0900         78-   0A          ASL
19-   9D F7 09    STA   $09F7,X       79-   D0 F6       BNE   $0871
1C-   A0 04       LDY   #$04          7B-   98          TYA
1E-   88          DEY                 7C-   15 D7       ORA   $D7,X
1F-   D0 FD       BNE   $081E         7E-   95 D7       STA   $D7,X
21-   CA          DEX                 80-   C6 EB       DEC   $EB
22-   D0 ED       BNE   $0811         82-   F0 31       BEQ   $08B5
24-   A2 08       LDX   #$08          84-   A6 E9       LDX   $E9
26-   A9 00       LDA   #$00          86-   A9 00       LDA   #$00
28-   95 CF       STA   $CF,X         88-   95 CF       STA   $CF,X
2A-   CA          DEX                 8A-   F0 D9       BEQ   $0865
2B-   D0 FB       BNE   $0828         8C-   A9 00       LDA   #$00
2D-   A2 08       LDX   #$08          8E-   A2 09       LDX   #$09
2F-   2C 10 08    BIT   $0810         90-   CA          DEX
32-   30 FB       BMI   $082F         91-   F0 22       BEQ   $08B5
34-   2C 10 08    BIT   $0810         93-   B4 CF       LDY   $CF,X
37-   30 0F       BMI   $0848         95-   F0 F9       BEQ   $0890
39-   50 F9       BVC   $0834         97-   95 CF       STA   $CF,X
3B-   AD 10 08    LDA   $0810         99-   A2 09       LDX   #$09
3E-   95 CF       STA   $CF,X         9B-   CA          DEX
40-   CD 10 08    CMP   $0810         9C-   F0 17       BEQ   $08B5
43-   F0 FB       BEQ   $0840         9E-   A9 40       LDA   #$40
45-   CA          DEX                 A0-   35 D7       AND   $D7,X
46-   D0 EC       BNE   $0834         A2-   D0 F7       BNE   $089B
48-   E6 E8       INC   $E8           A4-   A9 80       LDA   #$80
4A-   A5 E8       LDA   $E8           A6-   35 D7       AND   $D7,X
4C-   8D 20 08    STA   $0820         A8-   95 D7       STA   $D7,X
4F-   EA          NOP                 AA-   98          TYA
50-   EA          NOP                 AB-   15 D7       ORA   $D7,X
51-   EA          NOP                 AD-   95 D7       STA   $D7,X
52-   A5 EA       LDA   $EA           AF-   C6 EB       DEC   $EB
54-   85 EB       STA   $EB           B1-   F0 02       BEQ   $08B5
56-   A2 08       LDX   #$08          B3-   D0 D7       BNE   $088C
58-   A9 BF       LDA   #$BF          B5-   20 00 FF    JSR   $FF00
5A-   35 D7       AND   $D7,X         B8-   C9 04       CMP   #$04
5C-   95 D7       STA   $D7,X         BA-   B0 03       BCS   $08BF
5E-   CA          DEX                 BC-   4C 06 00    JMP   $0006
5F-   D0 F7       BNE   $0858         BF-   C9 08       CMP   #$08
61-   A9 09       LDA   #$09          C1-   B0 05       BCS   $08C8
63-   85 E9       STA   $E9           C3-   A9 2E       LDA   #$2E
65-   C6 E9       DEC   $E9           C5-   4C 08 00    JMP   $0008
67-   F0 23       BEQ   $088C         C8-   4C 0F 00    JMP   $000F
```

The cassette version of this program loads all of page zero of memory (its total requirement) and in the process initializes a couple of things that you will need to care for manually if the cassette is not available. When entering manually, be sure to set the number of outputs to correspond to the number you have available. For example, assuming that you have a system with a single QuASH, the number of channels available should be set to 4 using the following computer keyboard sequence:

RESET-0-0-E-A-DISP
0-4-ENTER

The tape version initializes the number of outputs at the most likely number of 4. If you want to use less channels (because of lack of modules, say) or have a system with more, do it as was shown above.

When entering the program manually, make sure the decimal mode flag in the status register is cleared by using this sequence:

RESET-0-0-F-F-DISP
0-0-ENTER

This is automatically taken care of when the tape version is loaded.

USING POLY 1.0

With everything connected, loaded and initialized, we're ready to begin making music. Go to the beginning of the program and begin running it.

RESET-0-0-0-6-RUN

If everything is working properly, we will see the 8700 displays counting quickly, incrementing by one for each scan of the keyboard. All of the QuASH outputs should be at a very low output voltage (the program initializes them as zero) and the trigger flags for each channel should be cleared.

As we press synthesizer keys, QuASH channels should "come alive" and produce control voltages corresponding to the keys that POLY 1.0 has assigned to them. The trigger flags should be set if the key corresponding to the channel is currently down and clear when the key is released.

TWO MORE FEATURES OF POLY 1.0

While POLY is running, touching any of the keys from 0-3 on the 8700 keyboard (the first row of keys) causes the system to clear all QuASH channels to zero and wait for new data to be assigned. You'll figure out what this is good for as you become familiar with the system.

Maybe more importantly, touching any of the keys 4-7 (the second row on the 8700) provides a tuning function and causes all QuASH channels to produce the same note with the trigger flags set, allowing all oscillators to be set to the same pitch. The note produced corresponds to the 2nd C on a standard configuration 3 octave keyboard. THE CHANNELS MUST BE CLEARED AFTER TUNING by touching the first row of 8700 keys.

THE SYNTHESIZER

There are an almost unlimited number of ways to use the multiple control voltage produced by the QuASH and POLY 1.0. You may want to use multiple VCO's mixed into a single voicing circuit, (See figure 9), or what amounts to a complete synthesizer for each control channel or anything in between, (See figure 10).

A word of advice: in your beginning stages of learning to use this system, you should try to stick to configuration in which all of the channels are producing the same "type" of sound - as close to identical as possible. As your skills progress and you develope a feel for how POLY 1.0 is going to massage data you can work up to using some output channels to set VCO pitches while others control filter parameters (just an example - the number of possible combinations is extraordinarily large).

POLY 2.0 is under developement and features the use of some QuASH channels as software controlled envelope generators, reducing the need for lots of these hardware modules.

POLY 3.0 provides for computer storage of sequences of chordes or notes.

ONLY POLY 1.0 IS AVAILABLE NOW. The others are still a couple of months away. I mention them only because I want to make sure that we all understand that the nature of this new musical tool is a function of the program that is running and not so much of the hardware that it uses.

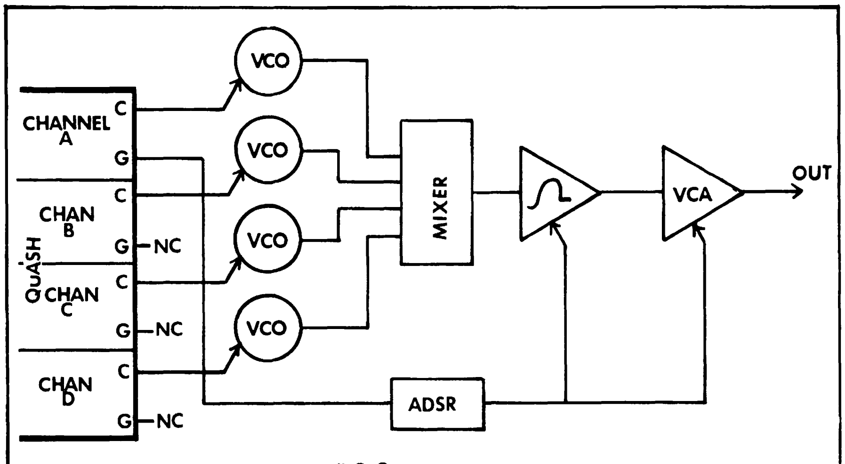


FIG.9

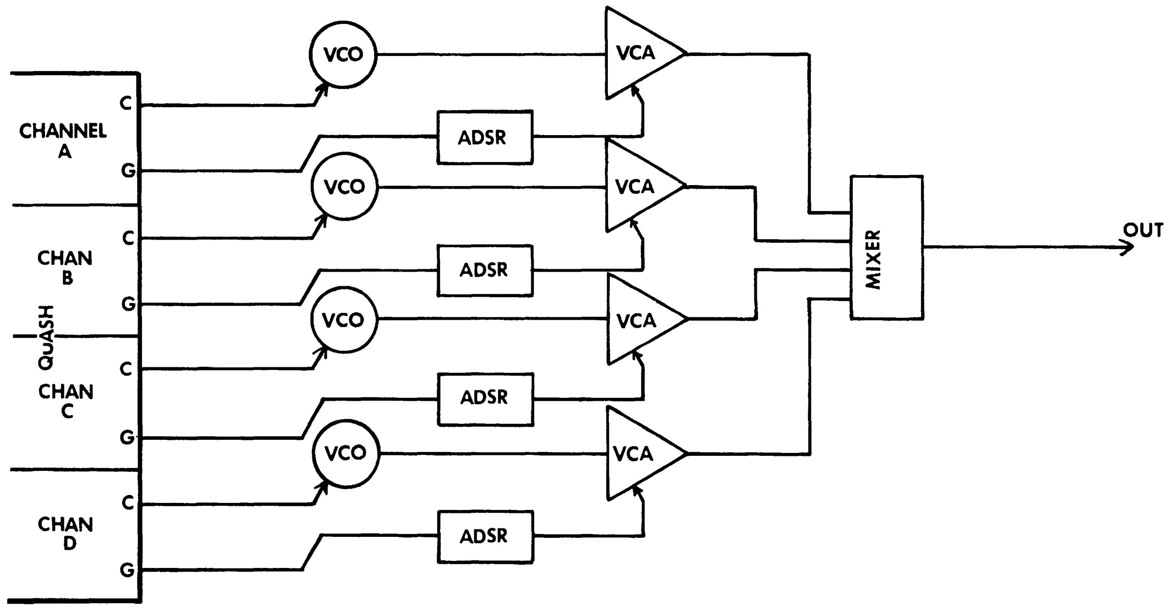


FIG. 10