

Oberlin

Hybrid Synthesizer



THE OBERLIN HYBRID SYNTHESIZER

INTRODUCTION

The Hybrid Synthesizer is a four voice electronic synthesizer designed to be controlled from a computer. It is currently interfaced to the RSC FORTH Microcomputer, a 6502 based microcomputer with a ROM based Forth language.

The synthesizer was designed and built here at Oberlin in 1973-1974 by Sergio Franco. Before coming to Oberlin Sergio Franco was at the University of Illinois completing a PhD program in Computer Science. While there he worked closely with the composer Salvatore Martirano to put together an elaborate performer-oriented electronic music synthesizer which became known as the **Sal-Mar Construction**. At the time it was one of the few electronic music instruments to allow real-time performer interaction with the Instrument. The **Oberlin Hybrid Synthesizer** is an offspring of the **Sal-Mar**. In fact, Sergio Franco's PhD dissertation, *Hardware Design of a Real-Time Musical System* serves well as documentation for both the **Sal-Mar** and the **Hybrid Synthesizers** circuit modules.

The main difference between the two machines lies in their performer-to-machine interface. The Sal-Mar Construction used a large panel of multitudinous switches and knobs at which the performer sat like a grand wizard. The Hybrid Synthesizer has hardly any knobs and its only switch is an On/Off switch. The only way to play the instrument is through a computer program. This gives the performer a malleable interface. The Hybrid synthesizer, like the original Sal-Mar, can be played in real-time by a performer but it can also be automated by the computer or controlled by a combination of the two. Thus the Hybrid has the advantage of being adaptable to many different playing situations.

THE HYBRID SYNTHESIZER VOICE

The Hybrid has four identical voices. Figure One shows a block diagram of a voice. It consists of a voltage-controlled oscillator which cycles through two waveshape memories. The two waveshapes are mixed together and sent through a low-pass filter with a controllable cutoff frequency. After the filter the signal goes through a voltage-controlled amplifier to set the overall volume. Finally, pan controls place the signal within a quad space by distributing it among the four synthesizer outputs. Any voice can be patched to modulate any other voice. The modulation can be frequency modulation or waveshape mixer modulation. Notice that if one of the two waveshapes is silence, then the mixer modulation becomes simple amplitude modulation. A 16-bit timer is included with each voice to facilitate computer controlled timing of the sound events.

Each voice has ten computer controlled parameters: pitch, waveshape memory, waveshape mixing, lowpass filter cutoff frequency, volume, left/right pan, front/back pan, transposition or frequency modulation depth, modulation patch, and timer delay. All but four of the parameters (pitch, waveshape, modulation patch, and timer) have ramp type parameters. You can think of a ramp as a single segment envelope generator. The parameter value slides from its present value to some end value at a programmable speed. The programmer specifies the end value and the speed of the ramp. Both the endpoint and the speed for all the ramp parameters are limited to a range of 0 to 15 (four bit numbers in digital terms). Of the non-ramping parameters, the pitch and mixer have a ranges of 0 to 255 (8 bit numbers) and the timer has a range of 0 to 65,535 (16 bit number).

THE RAMP MODULES

Most of the Hybrid's programmable parameters are ramps; these include the waveform mix, the lowpass filter's cutoff frequency, transposition or FM depth control, the X-pan, Y-pan, and the volume. For these parameters the programmer must specify both a final value and a ramp speed. Both values are limited to a range of 0 to 15. The following table gives the 16 possible ramp speeds:

VALUE	RAMP RATE	TIME FOR FULL RAMP
0	1.5 levels/sec	0 seconds (from level 0 to 15)
1	2.4	6.3
2	3.75	4.0
3	6.0	2.5
4	9.45	1.6
5	15.0	1.0
6	24	0.63
7	37.5	0.40
8	60	0.25
9	94.5	0.16
10	150	0.10
11	240	63 milliseconds
12	375	40
13	600	25
14	945	16
15	1500	10

There is one exception to the ramp values given above: for volume decays the rates are half the values shown and the times are twice the value.

The ramp control voltages are available from 24 jacks on the Hybrid's front panel. Each jack is marked as belonging to a single hybrid parameter and voice number. The control voltages at these jacks range from zero to 10 volts. They can be patched to the control voltage inputs of any of the analog synthesizers in the studio thus giving you some computer control over the analog synthesizers.

THE PITCH MODULE (P)

The pitch module outputs a control voltage to a voltage-controlled oscillator which is used in cycling through the waveform memories. This parameter controls the pitch of the waveform. It is a non-ramping control with a range of 0 to 255; thus a total of 256 pitches can be set by the pitch parameter. The pitch of the output waveform changes in quarter tone increments with the value zero corresponding to approximately 5.15 hertz, and the value 255 corresponding to approximately 8.1Khz. The accompanying table shows the note and frequencies corresponding to each of the 256 pitch values. A tuning adjustment for each voice can be found on the front panel of the Hybrid. Be sure the transposition module is set to zero before attempting any tuning.

THE TRANSPOSITION / MODULATION DEPTH MODULE (T)

This is a double function module controlling either the pitch or the modulation depth. It is a ramping module. When used as a transposition control it adds an additional control voltage to the voltage controlled oscillator. Thus you can use it to transpose the pitch of a voice or to add frequency glissandos to a note using the slower ramp speeds.

When other voices are patched to frequency modulate the voice, this module changes its function to that of modulation depth control. Since the control is still a ramp function, the FM depth can be dynamically changed by using different ramp rates.

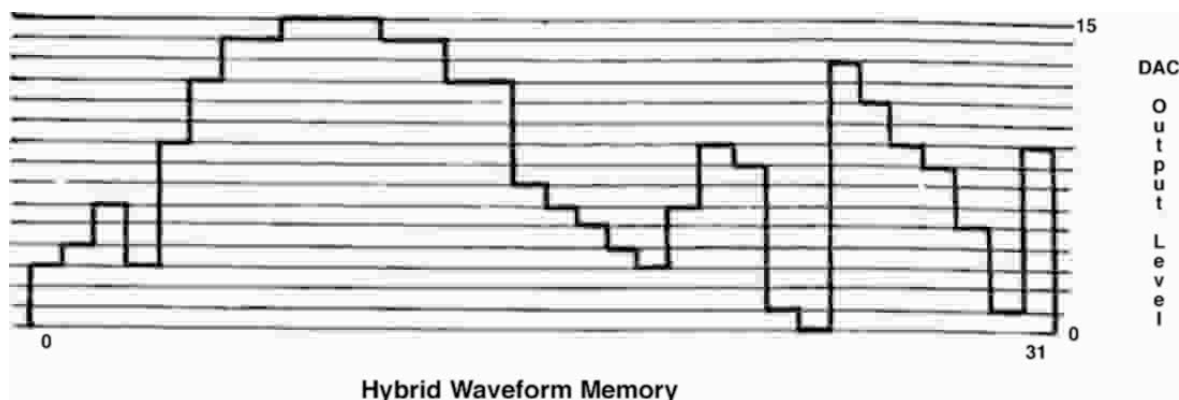
THE WAVEFORM GENERATOR (W)

The Hybrid voice is centered around a digital waveform generator. There are two waveform memories, each holding 32 four-bit samples. A figure shows an example of one cycle of a possible waveform. Each memory holds one complete cycle of a waveform. The waveform cycles are repeated at a rate set by the voltage controlled oscillator.

The user can load any of 64 different waveforms into waveform Memory A and/or Memory B at any time. Two of the waveforms are internally set and always available to the user: Waveform #63 is whitenoise, and Waveform # 62 is a random complex waveform. All other waveforms must be loaded by the user.

To load a waveform into the voice's waveform memories you must specify the Waveform number, and the memory (A and/or B) you want to load it into. The change will take place immediately.

To load a user-defined waveform into the general waveform memory first load the wave number times 64, then load the 32 waveshape samples one at a time (each sample is limited to a range of 0 to 15). Please use only waveshape numbers 0 through 31.



THE WAVEFORM MIXER MODULE (M)

This mixer has the rather specialized function of mixing the outputs of the two waveform Memories A and B. This is a ramp function so that the voice output can be dynamically panned between two different waveforms. A ramp endpoint value of zero sends only Memory A's waveform to the output. A ramp endpoint value of 15 sends only Memory B's waveform to the output. Any value inbetween 0 and 15 sends a mixture of the two waveforms.

If another voice is patched to waveform modulate the voice then the mixer parameter changes its function (as did the Transposition/FM depth parameter) to modulation depth control. Waveform modulation is a rapid panning between the two waveforms. Modulation depth then sets how far the panning varies around a central mix value. Note that if one of the waveforms is a constant zero then the waveform panning becomes simple amplitude modulation (AM).

LOW-PASS FILTER MODULE (L)

This is a voltage controlled filter with a variable cutoff frequency. The cutoff frequency is varied through a ramp function. The filter is setup to track the pitch value, so that the cutoff frequency is always a specified distant from the pitch frequency of the note. It is that distance from the pitch frequency which is changed through the ramping filter parameter. By making the filter track the pitch value, the waveforms timbre remains constant over the entire pitch range changing only when the cutoff frequency parameter is changed.

There are also manual filter response (filter 0) controls for each voice located on the Hybrid's front panel.

THE VOLUME CONTROL MODULE (V)

After the filter, the voice signal goes through a voltage controlled amplifier which sets the overall volume of the signal. This is a ramp parameter. A zero endpoint value completely shuts off the output signal and a value of 15 results in maximum signal strength. Since the volume control is a ramp, this module can act as an envelope generator by simply stringing several volume ramp commands together.

THE X-Y LOCATION MODULES (X. Y)

Each voice of the Hybrid can be dynamically panned to any location in a quadraphonic space. The two ramp modules X and Y control this function. The X-Location module controls right/left signal placement and the Y- Location module controls front/back placement (depending, of course, on how the Hybrid's 4 outputs are connected to the speaker system and in what direction the listener is facing).

To localize in the center of the space send endpoint values of 7 to both the X and Y modules. To send the voice signal to only one of the four speakers in the room send one of the following endpoint values: (X,Y) = (0,0), (0,15), (15,0), or (15,15). To dynamically move the sound around the space, use slow ramping speeds.

MODULATION PATCHING

The Hybrid Synthesizer has two different modulation capabilities: Frequency Modulation and Waveform Mix Modulation. Any voice or voices can be patched to modulate any other voice or voices. When an FM modulation patch is made, the voice's Transposition parameter becomes a Modulation Depth control. When a Waveshape Mix Modulation patch is made, the voice's Mixer parameter becomes a Modulation Depth control. Both depth controls are ramping controls.

PROGRAMABLE TIMERS (T)

The Hybrid contains four 16-bit timers to aid the computer in timing sound events. Each timer can be programmed for delays ranging from 10 milliseconds to 15 minutes in 10 millisecond increments. To use the timers you first load the timer module with a time delay value and then use the Hybrid's module polling system to check for completion of the delay. To get a delay of 't' seconds load a value of $100 * t$

MODULE POLLING

If the user is to program sound events using the Hybrid timers and ramping controls they must have some way of finding out when the ramps have finished ramping and when the timers have completed their time delays. A module polling system has been setup to facilitate this. The hybrid can be set up to poll all the ramp modules and the timer modules at an extremely fast rate (4 million modules per second). It steps through the modules testing each one for completion of a ramp or time delay. When it finds a completed ramp or time delay the polling stops and a computer interrupt signal is generated. It is then up to the programmer to react to this Interrupt signal. The number of the module at which the polling stopped can be read from a specific computer address. The programmer reads that location, performs whatever actions are desired and then enables the polling again.

If, for some reason, the programmer does not want a certain module to stop the polling action when it completes its ramp or time delay, they can set a flag called a MODMASK (Module Masking bit). Each module has a MODMASK flag. When it is set the polling will pass over that module without checking for a completed ramp or time delay.

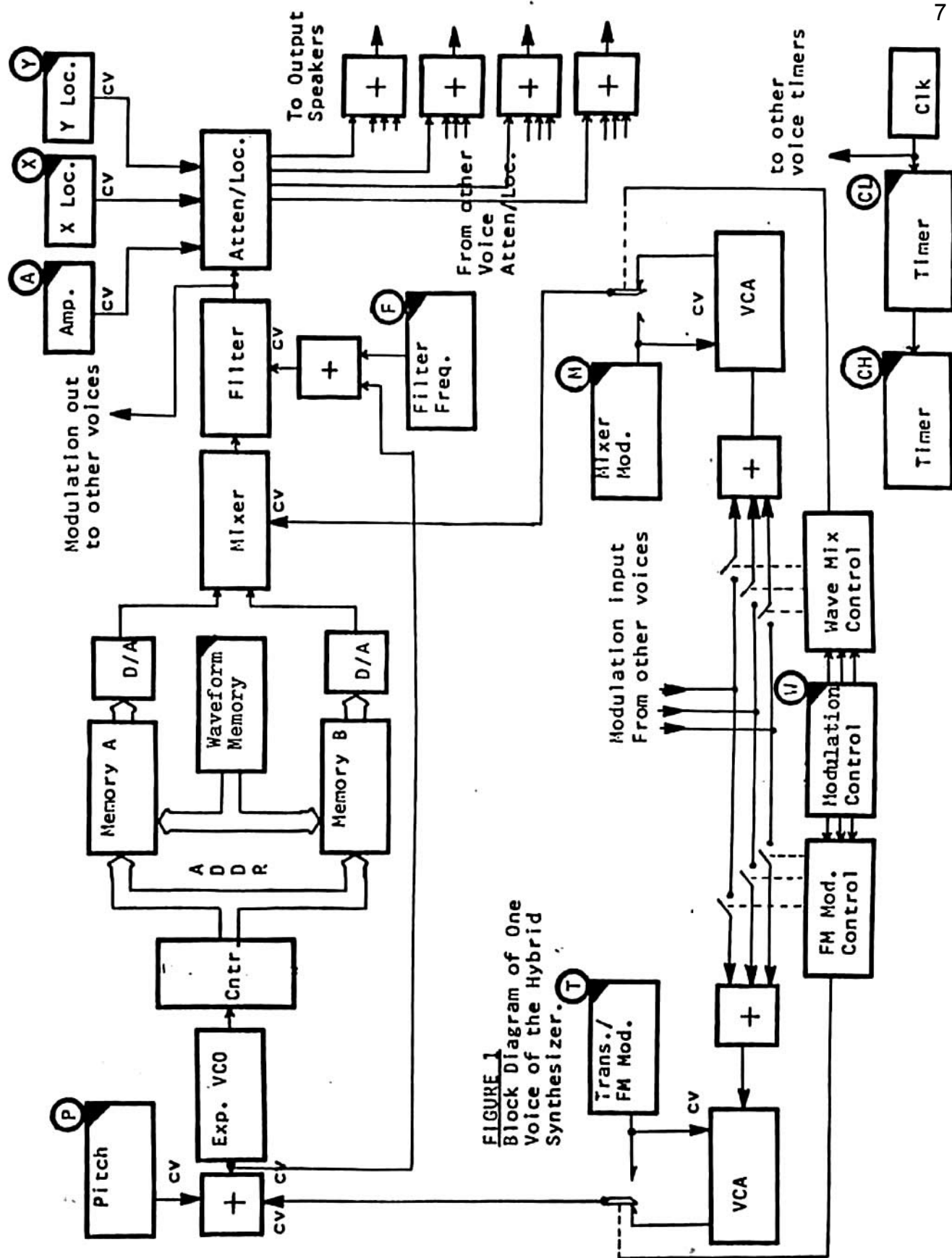
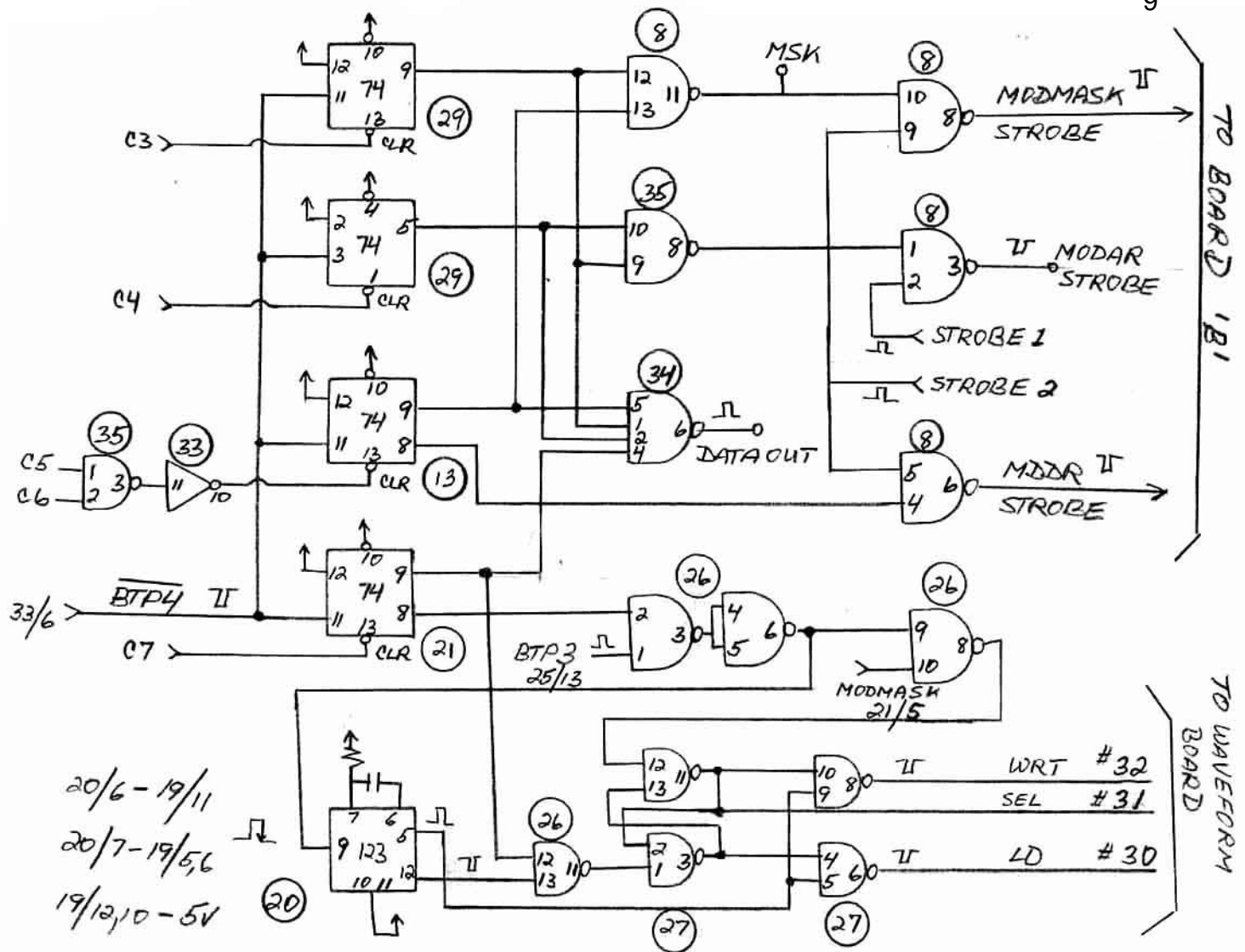
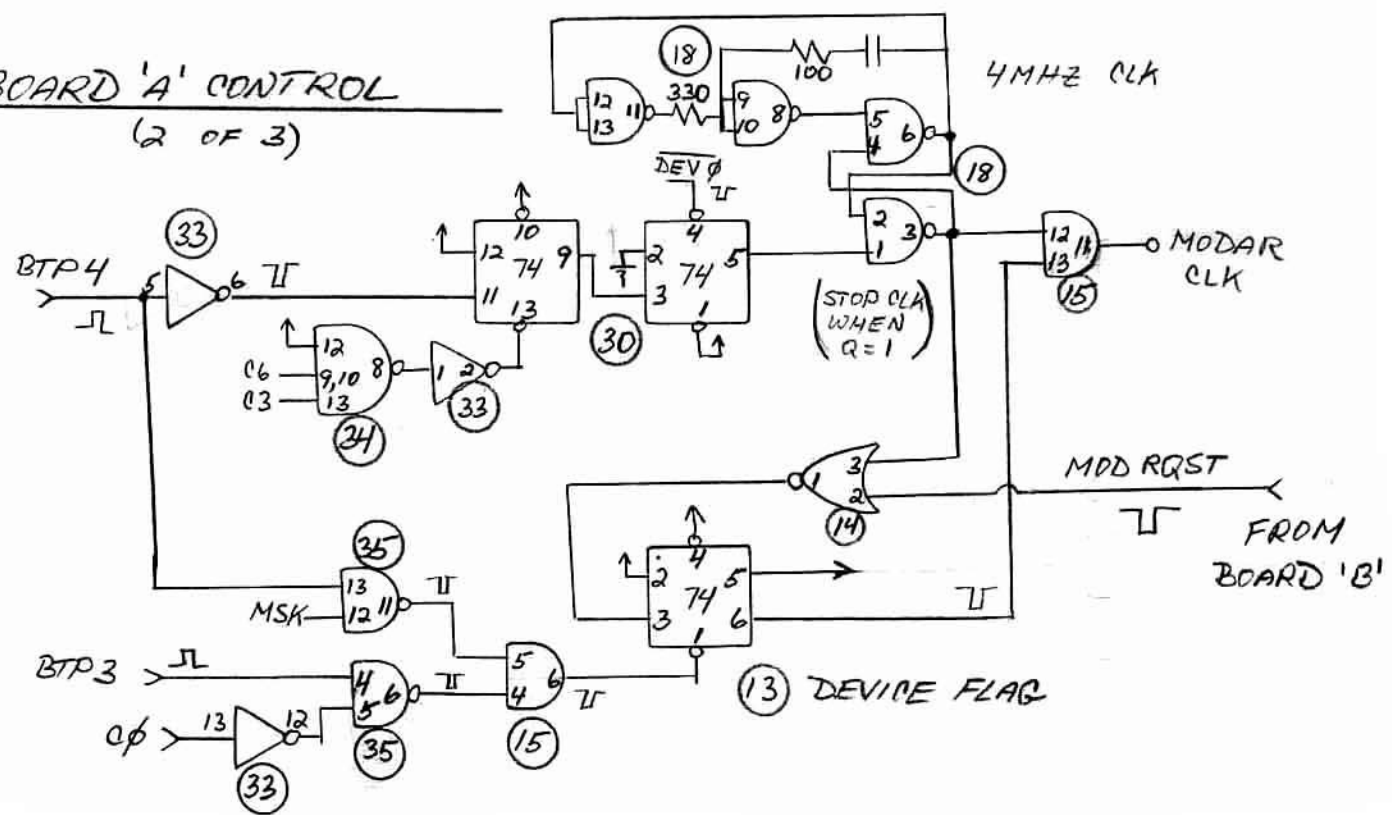


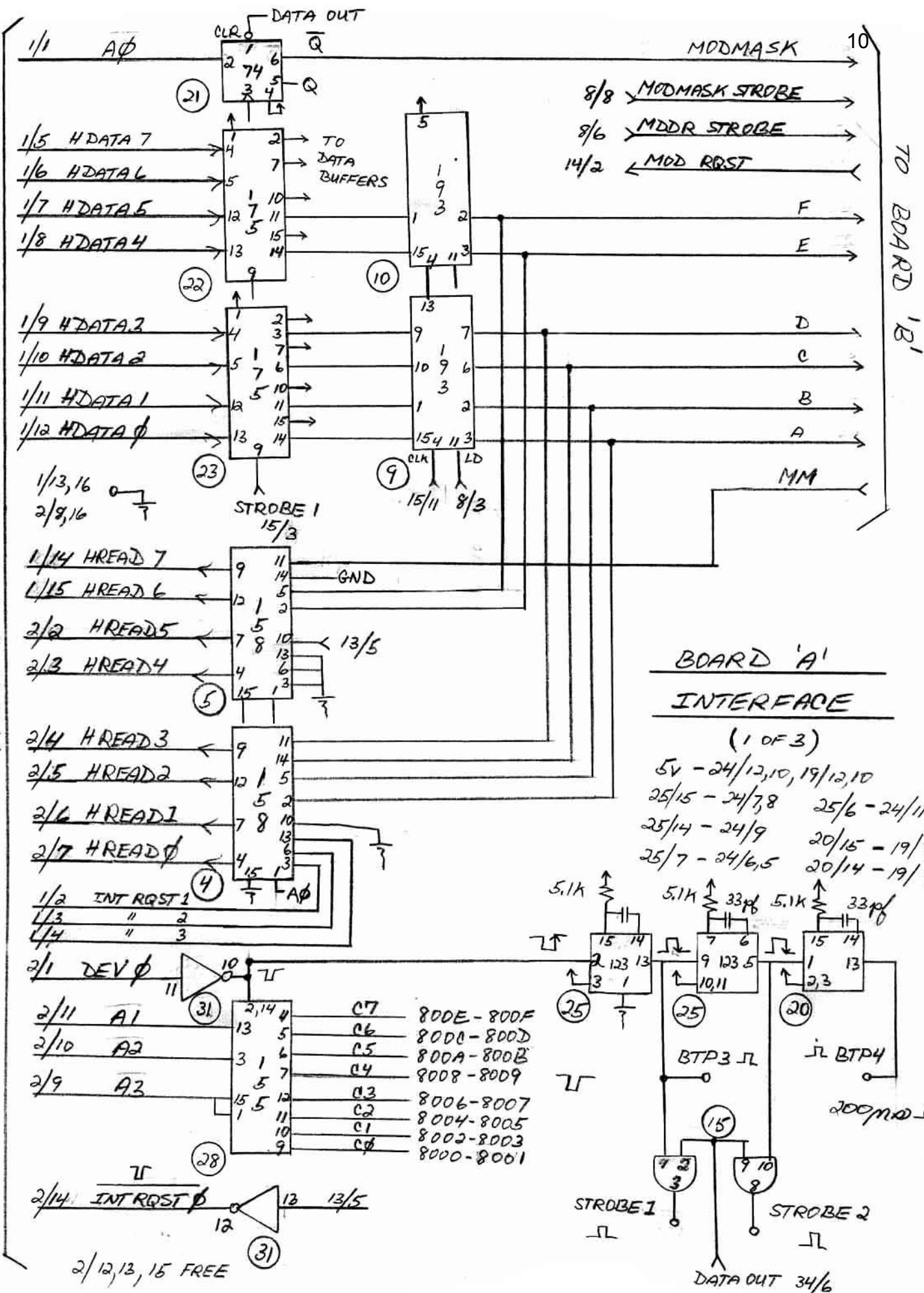
FIGURE 1
Block Diagram of One
Voice of the Hybrid
Synthesizer.



BOARD 'A' CONTROL
(2 OF 3)

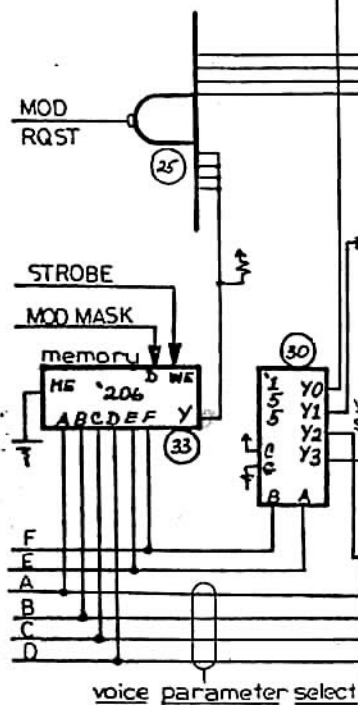
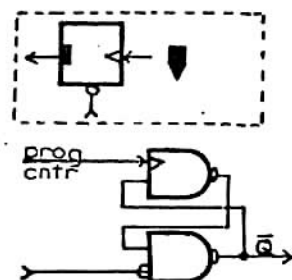
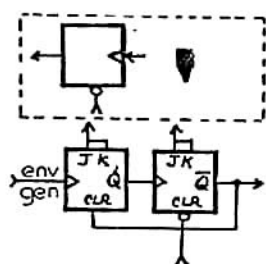


MICROCOMPUTER INTERFACE



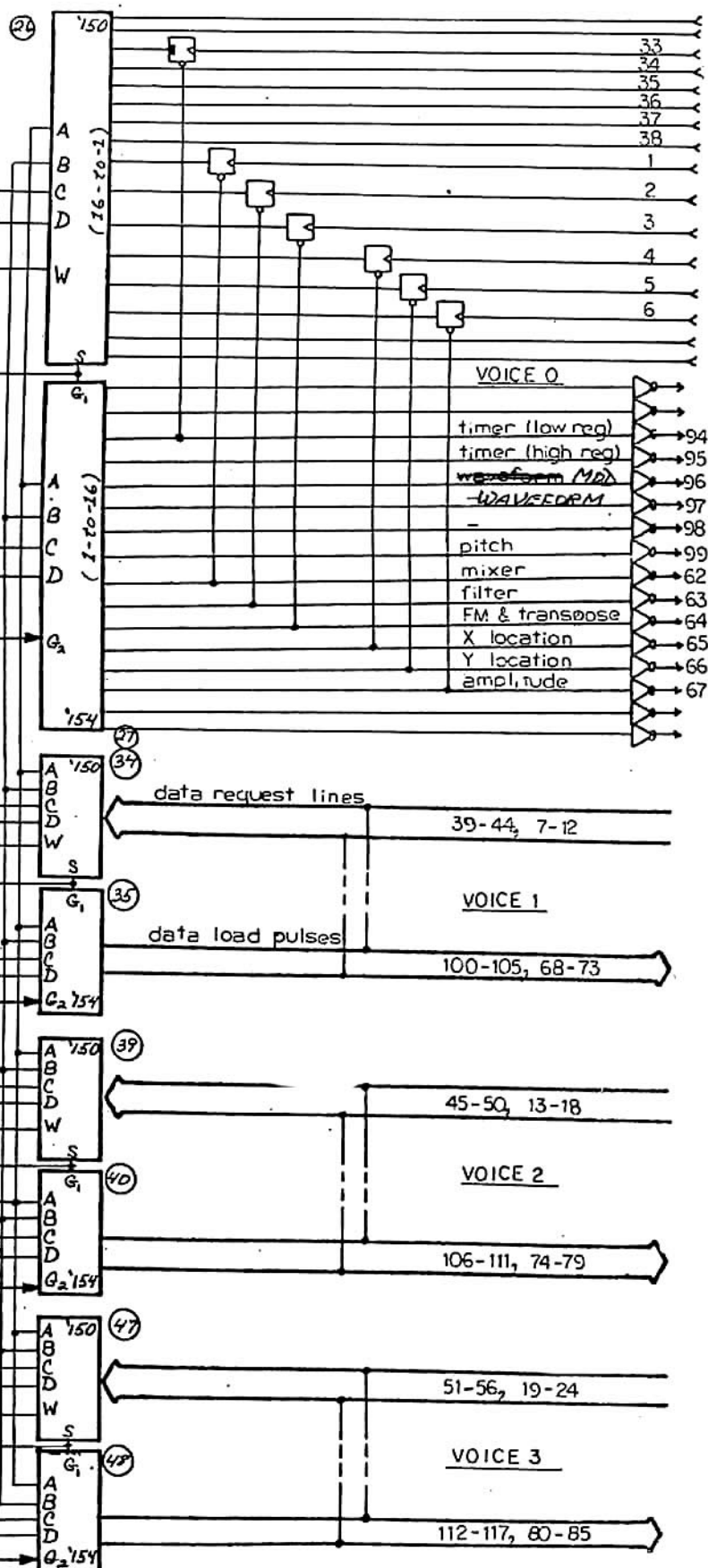
FROM BOARD 'A'

HYBRID MODULES



BOARD B

STROBE MODDR



HYBRID WAVEFORM GENERATOR

Each of the four Hybrid voices has two waveform memories. These 32x4 bit memories are to be called MEMA and MEMB. The pitch clock of each voice is used to count through these memories and the 4-bit outputs are converted to analog form and then mixed in the voice waveform mixer module.

The waveform contained in the MEMA and MEMB. of each voice is loaded from a central waveform generator card. The following is an explanation of how this waveform generator card operates.

MEMA and MEMB UPDATING

A voice Waveform Module # has been created for requesting an updating of the waveform contained in MEMA and/or MEMB of a voice. To request an update the user loads Waveform Module# into the microcomputer's address 32,776 10 and then loads the following data into address 32780:

BITS 0 to 4	WAVESHAPE NUMBER
BIT 5	0 -RAM, 1-ROM
BIT 6,7	00 - DO NOTHING
	01 - Load MEMA Only
	10 - Load MEMB Only
	11 - Load Both MEMA and MEMB

The user can select from a possible 64 waveshapes: 32 in RAM, and 32 in ROM. In the ROM Waveshape #11111 is White Noise and Waveshape #11110 is a complex random waveform.

On the Waveform Generator card this action generates a pulse on one of the voice module pulse lines MO, M1, M2, or M3 and the 8 bits of user loaded data appears on the data lines DDO to DD7. The data byte is loaded into a four by eight bit register (chip # 5 and 6). The loading pulse, derived from MO, M1, M2, M3, is translated by gates and flip-flops (#32,42) to form the load address. The address generated corresponds to the voice number. The M pulse also sets one of four flip-flops (23,24) if the data bits 6 and 7 are not in the "do nothing" state of 00.

The action according to voice number is as follows,

Voice 1 Waveshape update.

User loads 05Hex into 32776
 User loads Waveshape data in 32780
 M0 pulse occurs,
 Data loaded into register #0.
 Flip-flop #23-5 set.

Voice 2 Waveshape update.

User loads 15Hex into 32776.
 User loads Waveshape data into 32780.
 M1 pulse occurs.
 Data loaded into register #1.
 Flip-flop #23-9 set.

Voice 3 Waveshape update.

User loads 25Hex into 32776.
 User loads Waveshape data into 32780,
 M2 pulse occurs
 Data loaded into register #2.
 Flip-flop #24-5 set.

Voice 4 Waveshape update.

User loads 35Hex into 32776.
 User loads waveshape data into 32780.
 M3 pulse occurs,
 data loaded into register #3.
 Flip-flop #24-9 set.

POLLING OF THE UPDATE REQUESTS

The 4 voice flip-flop flags (23,24) are polled by the 555 clock (43) and the data selector (34). When one is found to be set, the polling clock is halted and the waveshape data is read from the 4x8 bit storage register as HA 5 to 12. HA5 to 9 (same as DDO to 5) is sent through '157 data selectors to form Memory address lines A3 to A9 for the RAM (11) and Prom (1) chips. HA10 (DD5) is used to enable the RAM output when 0, or the PROM output when 1. HA11 and 12 are used to turn on the MEMA write enable WA and/or the MEMB write enable WB for the specified voice (see chip #13).

The voice pitch clock is used to generated the W pulses. The voice counter is used to select one of the 4 voice pitch clocks and the (pitch clk)/32 in chip #7. The clk/32 marks the time to start loading. In this way the phase of the loaded waveshape is not random. The pitch clk is then used to initiate the W pulse HWRT and to increment the 5-bit counter (26) which supplies the memory address lines A0 to 4.

Thus the individual voice memories, MEMA, MEMB are loaded from the waveform generator card's memory over the 4-bit output lines (12,2) and the WA and WB write pulses. The speed of the loading depends on the current voice pitch.

When the 5-bit counter (26) is finished it sends a pulse to clear the flag flip-flop (through 33) and the polling is resumed.

WAVESHAPE RAM LOADING

The Waveform Generator board contains a 2114 Ram (11) which can be loaded by the user. The loading occurs through a two port system. The user first loads starting address into micro memory location 32782. The 8 bits loaded are translated as follows:

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
D7	D6	D5	D4	D3	D2	D1	D0	00	00

D7 through D3 determine the waveshape number, one of 32; D0 through D2, where in the 32 word waveform to start (4 word groups). This action causes the formation of a LD pulse on the Hybrid Board A which is sent to the Waveform Generator card to load the counters (27,37,47) supplying the RAM addresses.

The user then loads the 4-bit waveform data words into micro location 32783. This action causes the WRT pulse to be generated on Hybrid Board A which is sent to the Waveform Generator Card to load the RAM with Data bits 0 through 3. The Address counter is then automatically incremented in preparation for the next 4-bit word to be loaded by the user,

NOISE GENERATION

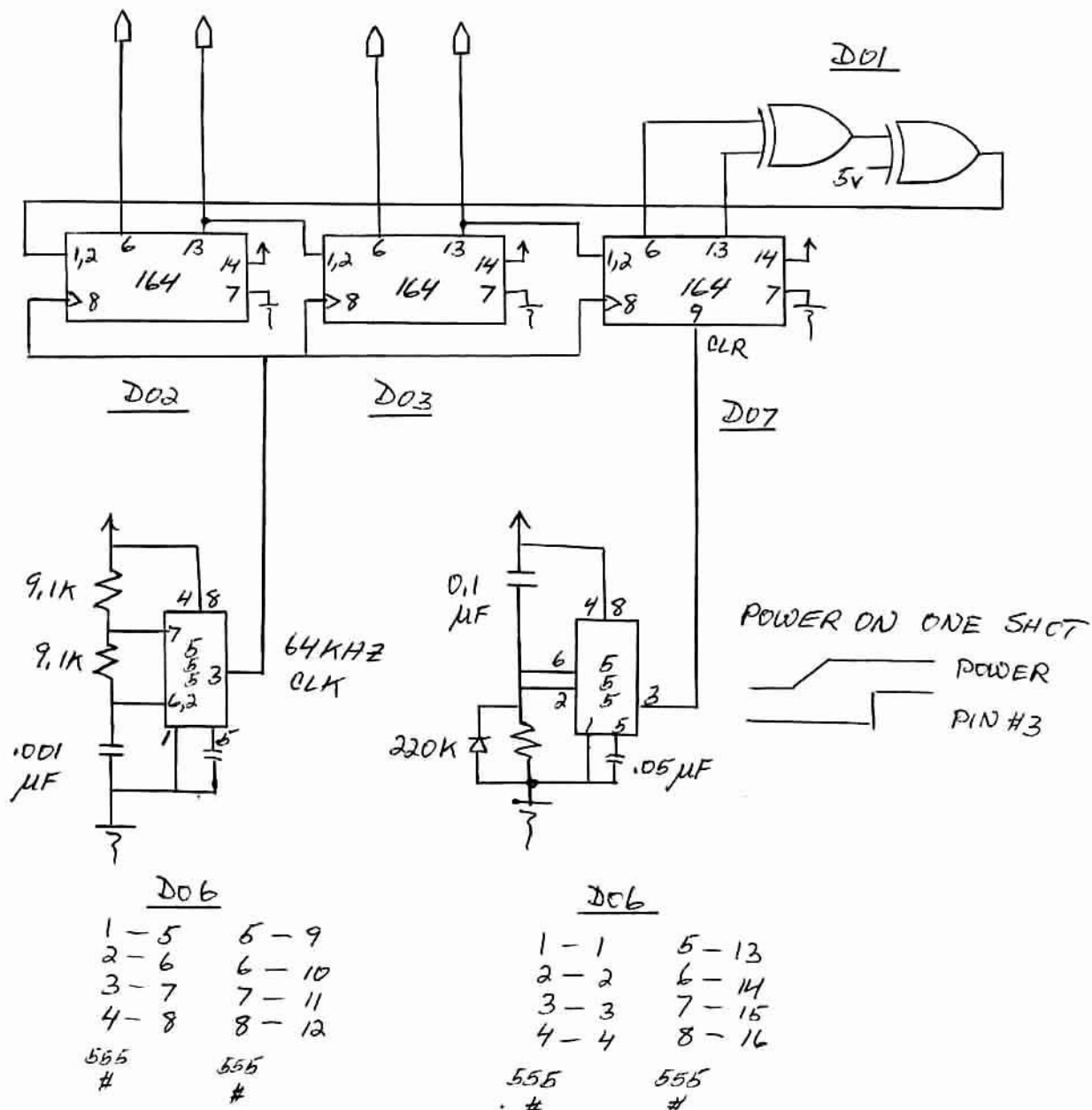
The user may obtain white noise by selecting prom waveshape #11111. This waveshape number is decoded in chips 25 and 36 to set the SELECT line (45-11) which puts the Noise Generator's outputs (31) onto the output lines (12,2). The Noise Generator is selected actually whenever the polling is in process. Chip number 14 (259) is an 8-bit register whose outputs are used to turn on and leave on the W lines for which noise was desired.

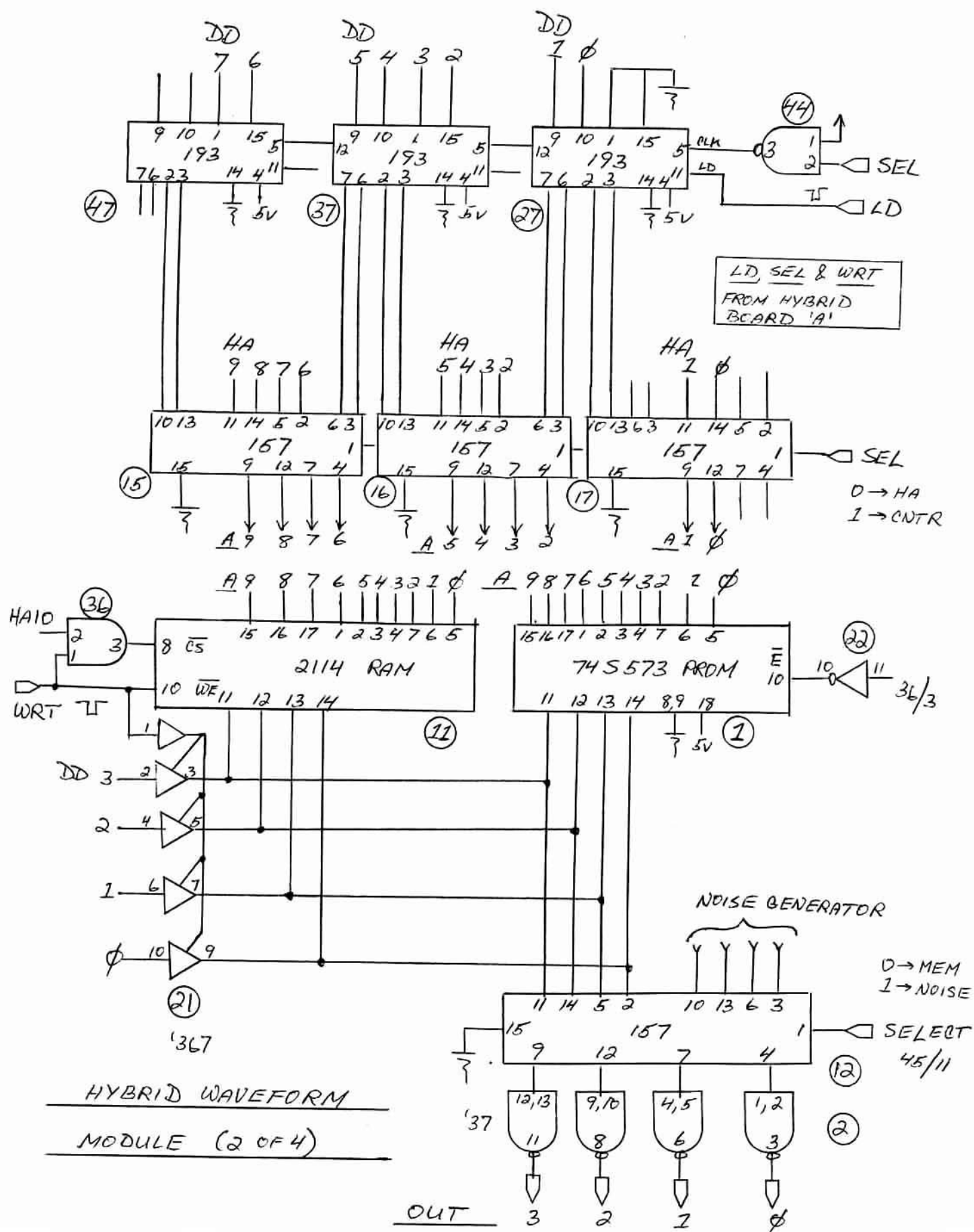
The user may obtain a random waveform by selecting prom waveshape #11110. In this case the normal updating procedure occurs except that the Noise Generator is selected for an output, In this case the W lines don't remain on after the updating is completed as happens with the white noise selection.

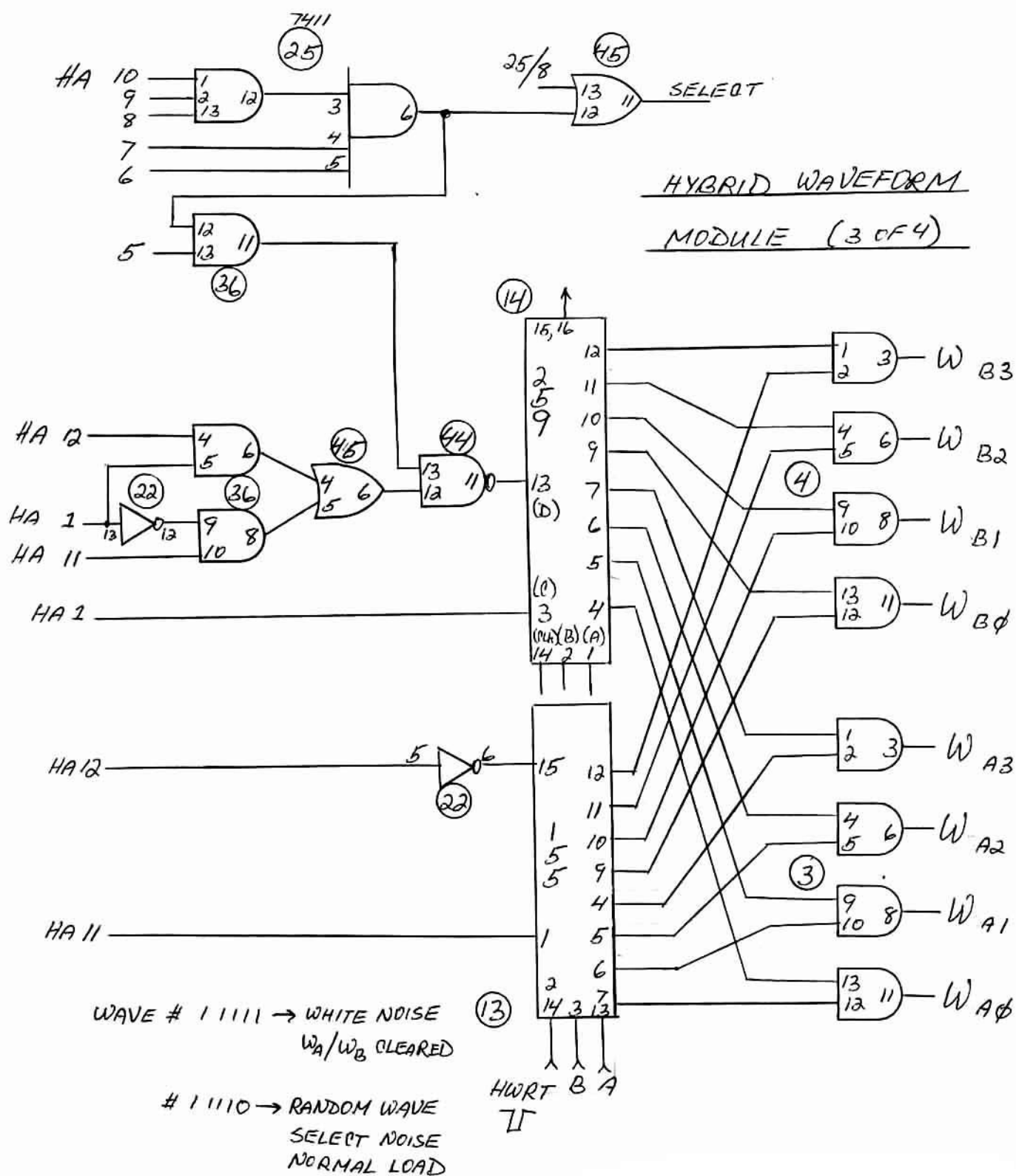
HYBRID
WAVEFORM MODULE
(4 OF 4)

NOISE OUTPUTS
ON RIBBON CABLE

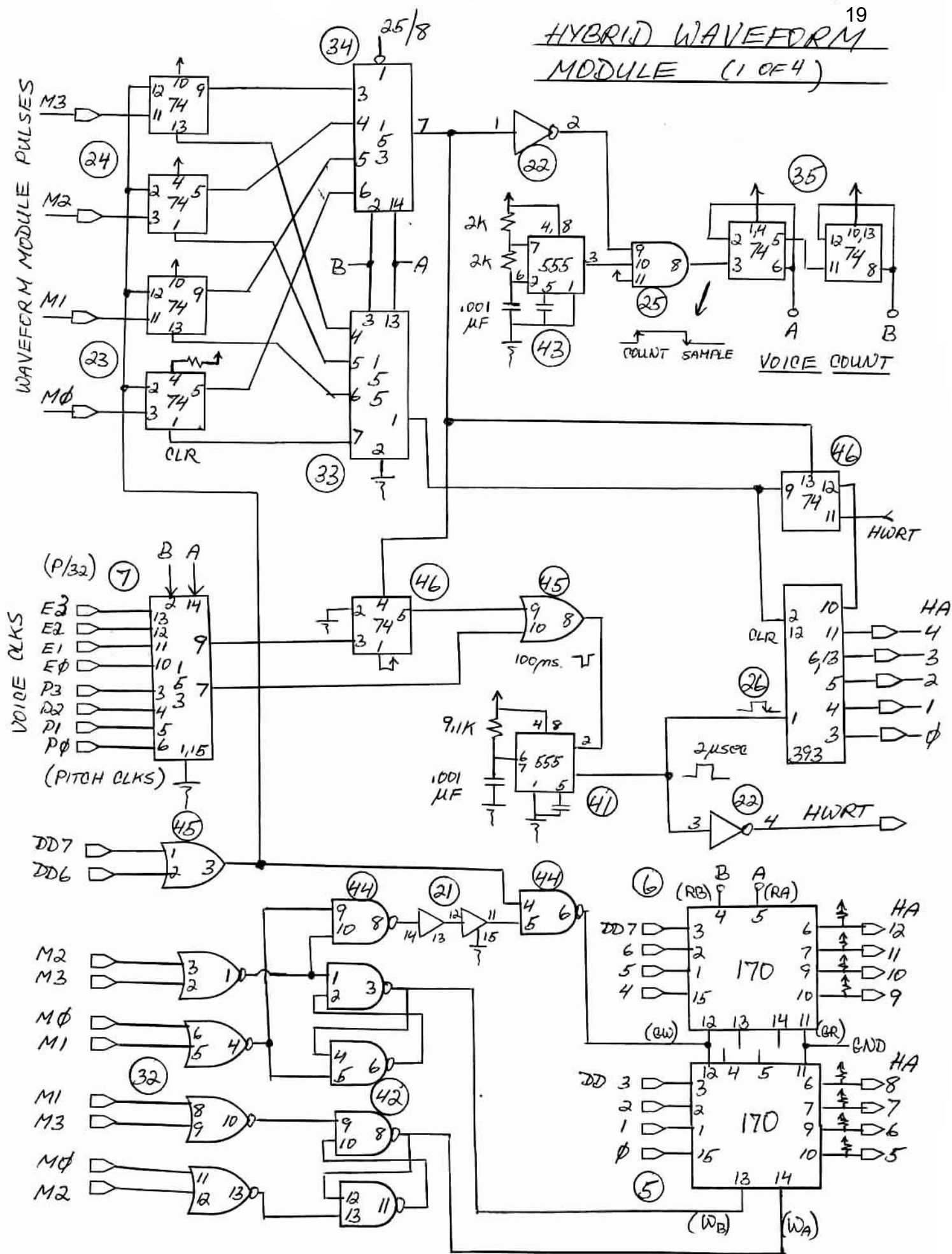
NOISE GENERATOR
ON BOARD 'D'







WAVEFORM MODULE PULSES



SAMPLE HYBRID PROGRAMS USING 8K BASIC

```

1 REM THE HYBRID TIMER MODULES ARE TESTED WITH THIS PROGRAM.
2 REM VOICE 0 INTERVAL TIMER IS LOADED, LOW AND HIGH
3 REM REGISTERS, WITH INPUT VARIABLES L AND H RESPECTIVELY.
10 INPUT L,H
29 POKE 32776,3 POKE 32779,1
30 POKE 32776,2 POKE 32779,11
40 PRINT PEEK(32772), PEEK(32771)
50 WAIT 32771,128
60 PRINT "DONE" PEEK(32772), PEEK(32771)
70 END

```

```

1 REM THIS PROGRAM WILL IMPLEMENT ANY NUMBER OF HYBRID MODULE
2 REM PARAMETER CHANGES AT THE START OF EACH TIME INTERVAL. THE
3 REM TIME INTERVAL BETWEEN EVENTS IS SET BY VOICE 0 TIMER-
4 REM DATA IS TO BE GIVEN BY <MODULE# DATA> PAIRS WITH THE
5 REM VOICE 0 TIMER MODULE ENDING EACH SOUND EVENT LIST.

```

```

10 DATA 10,15, 13,255, 4,0, 7,99, 3,90, 2,0
20 DATA 13,0, 20,0, 26,0, 29,240, 25,0, 23,97, 3,99, 2,0
50 READ M,D
40 POKE 32776,M POKE 32779,D
50 IF M=2 THEN WAIT 32771,128
60 GOTO 30
70 END

```

```

1 REM THIS PROGRAM WILL IMPLEMENT ANY NUMBER OF HYBRID MODULE
2 REM PARAMETER CHANGES AT THE START OF EACH TIME INTERVAL. IN
3 REM ADDITION, IT CAN IMPLEMENT ONE MULTI RAMP MODULE CHANGE
4 REM ILL EACH TIME INTERVAL (FOR ENVELOPES, FOR EXAMPLE).
5 REM DATA IS TO BE GIVEN IN (MODULE# DATA> PAIRS. VOICE 0
6 REM TIMER MODULE BEGINS EACH EVENT LIST AND THE MULTIRAMP
7 REM MODULE ENDS THE EVENT LIST.
10 DATA 5,99, 2,0, 7,45, 13,255, 13,10,
20 DATA 3,99, 2,0, 7,30, 13,255, 13,0
50 READ M,D
40 POKE 32776,M : POKE 32779,D
50 READ M1,D1
60 IF M1=3 THEN POKE 32776,2 : WAIT 32771,128
70 IF M1=M THEN WAIT 32771,128
80 M=M1 D=D1 GOTO 40
90 END

```

```

1 REM THIS PROGRAM WILL IMPLEMENT AS MANY AS 10 MULTI-RAMP
2 REM MODULE PARAMETER CHANGES AT THE START OF EACH TIME
3 REM INTERVAL. DATA IS TO BE GIVEN IN <MODULE#, NUMBER
4 REM OF RAMPS, 1st RAMP, DATA, 2nd RAMP DATA, AND SO ON>
5 REM VECTORS. THE VOICE 0 TIMER DATA, WHICH IS TO END EACH
6 REM EVENT LIST, IS TO BE GIVEN IN THE FORM, <3, LOW REGISTER
7 REM VALUE, HIGH REGISTER VALUE>.

```

```

8 DATA 7,1,140, 15,2,250,7, 9,2,250,7, 29,1,2140, 3,99,0
9 DATA 7,1,30, 13,2,250,8, 9,2,250,7, 29,1,0, 3,99,0

```

```

10 D=0
11 REM ONE SOUND EVENT DATA SET READ IN TO THE MATRIX X.
20 READ M,N
30 IF M=3 THEN 200
40 D=D+1 : X(D,1)=M : X(D,2)=N+1
50 FOR A=N+2 TO 3 STEP -1
60 READ Z : X(D,A)=Z
65 NEXT
70 GOTO 20

```

```

75 REM STARTING RAMPS ARE LOADED.
80 FOR A=1 TO D
90 POKE 32776,X(A,1) : P=X(A,2) : POKE 32779,X(A,P+3)
100 NEXT

```

```

105 REM MODULES SCANNED FOR PROPER TIME TO LOAD RAMP DATA.
110 F=0
120 FOR A=1 TO D
125 IF X(A,2)=0 THEN F=F+1 : GOTO 150
130 POKE 32776, X(A,1)
135 IF PEEK(32771)<128 THEN 150
140 P=X(A,2) : POKE 32779,X(A,P+2) : X(A,2)=X(A,2)-1
150 NEXT
155 IF F<0 THEN 110

```

```

160 REM FINAL RAMPS HAVE BEEN LOADED, NOW WAIT OUT THE TIME INT.
165 POKE 2776,2 : WAIT 32771,128 : GOTO 10

```

```

190 REM SUBROUTINE TO LOAD INTERVAL TIMERS (FROM STATEMENT 30)
200 POKE 32776,3 : POKE 32779,N : READ N
210 POKE 32776,2 : POKE 32779,N : GOTO 80
220 END

```

Note, the maximum allowed number of nodules updated per sound event and the maximum allowed ramps per nodule is set by the Z, Y dimensions of the matrix X(z,y) respectively. These can be specified in a DIM statement if necessary.

