SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS SONUS

USING THE LATEST TECHNOLOGY, THE SONUS IS A VERY VERSATILE TOOL FOR PRODUCING SOUNDS OF ALL KINDS. IT CAN BE PROGRAMMED TO PRODUCE ANYTHING FROM AMAZING SOUND EFFECTS TO HARMONIOUS 3 PART MELODIES.THE SONUS HAS THREE TONE GENERATORS, 1 NOISE GENERATOR AND HAS A RANGE OF 4 OCTAVES. THE STEREO MODEL HAS 6 TONE GENERATORS AND 2 NOISE GENERATORS.

INSTRUCTIONS

Look at diagram 1, as you can see, there is a number next to each note.This number would tell the Sonus which note to play but the Sonus must also be told WHICH VOICE and HOW LOUD.

NOTE	VALUE(F)	NOTE	VALUE (F)
C	239	C	120
C#	225	C#	113
D	213	D	107
D#	201	D#	101
E	190	E	95
F	179	F	89
F#	169	F#	86
G	159	G	80
G#	150	G#	75
A	142	A	70
A#	134	A#	67
B	127	B	63

Diagram 1. FREQUENCY VALUES

CONTROLING THE VOLUME

Each Tone Generator/Voice has individual Attenuation, (The volume of each can be set seperately).To set the volume, a number must be POKED into the Sonus Base Address (S).

SONUS Base Add.
From Diag.1

Volume level
From Diag.4

POKE S, T + 16 + V

Gen Select
From Diag.3

Tells Sonus this is a volume command.(Always 16)

ZX 81 - 47616
SPECTRUM 55808

Diagram 2
BASE ADDRESS

TONE 1 - 128
TONE 2 - 160
TONE 3 - 192
NOISE - 224

Diagram 3.GENERATOR SELECT CODE (T)

15 - OFF
14 -
13 -
12 -
11 -
10 -
9 -
8 - HALF
7 -
6 -
5 -
4 -
3 -
2 -
1 -
0 - FULL

Diagram 4

So, for example to set the volume of Tone 1 to full:-

POKE S, (128 + 16 + 0)

(144 + 0)

= POKE 47616,144

```
5 rem demonstration
10 let s=47616
15 dim t(3)
20 let t(1)=128
30 let t(2)=160
40 let t(3)=192
50 gosub 200
55 let v=int (rnd*15+1)
56 let d=int (rnd*20)
60 let r=int (rnd*3+1)
70 let f=int (rnd*255+1)
80 let f1=int (f/16)
90 let f2=f-f1*16
100 poke s,t(r)+f2
110 poke s,f1
115 if v>8 then let v=15
120 poke s,t(r)+16+v
130 for i=1 to d
140 next i
160 goto 55
200 rem vol off
210 poke s,t(1)+16+15
220 poke s,t(2)+16+15
230 poke s,t(3)+16+15
240 poke s,224+16+15
250 return
```

For fun, try running this program for some interesting effects. Reading the next section (How to Control the Pitch) will help you to understand how it works.

CONTROLING THE PITCH

To change a frequency the Sonus must have the following information:-

1. Which Generator to change? (T) See diagram 3.
2. What Frequency Value? (F) See diagram 1.

(T) + (F) must be put in the Sonus Base Address to effect a frequency change to the right generator.

T + F is the information the Sonus needs to effect a frequency change.However, because the Sonus cannot accept a number larger than 255 we split the frequency value (F) into 2 smaller numbers by dividing F by 16 and calling it F(1). The remainder is called F(2).

F1 = INT (F/16) F2 = F - (F1 x 16)
FREQUENCY VALUE FORMULA

Diagram 5.

We can now send the information via a two Byte transfer:-

POKE S, F2 + T POKE S, F1
FORMAT OF FREQUENCY CHANGE DATA TO SONUS

Where T =128 for TONE GEN.1 160 for TONE GEN.2 192 for TONE GEN.3

Diagram.6

For example to play a note of C on Generator 1 (+1).

T1 = 128 F = 239 F1 = INT (239/16) = 14
F2 = 239 - (14 x 16) = 15

POKE 47616, 15 + 128
POKE 47616, 14

NOTE: It is a good idea to tur off a Tone Generator before changing the Freq Value. This ensures a clean transition dur the 2 Byte Data Transfer.

THE NOISE GENERATOR

As you can see from Diagram 3 the Noise Generator Select Code (T) is 224. This is added to 2 other Special Codes (Z & Z1) which control the type and frequency of the noise produced.

POKE S , 224 + Z + Z1

Z = Ø or 4 Z = Ø - Lower frequency periodic noise Z = 4 - White Noise
NOISE TYPE SELECT (Z)

Z1 = Ø - C/512 Z1 = 1 - C/1024 Z1 = 2 - C/2048 Z1 = 3 - Frequency of noise control by output of Tone Gen.3
NOISE FREQUENCY SELECT (Z1)

C = Clock Frequency of Computer

FINAL NOTES

You can work out your own frequency values from the following formula where C is the clock rate (F) = C / (32*8)

Try Running this program .....

```
1 dim b$(80)
10 rem phasers
20 let s=47616
30 gosub 200
35 rem set noise and attenuation
40 poke s,231
50 poke s,240
60 for v=0 to 15
70 for a=192 to 220
75 if inkey$<>"" then goto 40
80 poke s,a
85 poke s,v
90 next a
100 poke s,240+v
110 next v
120 stop
200 poke s,144+15
210 poke s,176+15
220 poke s,208+15
230 poke s,240+15
240 return
```

OPERATION

1. TONE GENERATORS

Each tone generator consists of a frequency synthesis section and an attenuation section. The frequency synthesis section requires 10 bits of information (F0-F9) to define half the period of the desired frequency (n). F0 is the most significant bit and F9 is the least significant bit. This information is loaded into a 10 stage tone counter, which is decremented at a N/16 rate where N is the input clock frequency. When the tone counter decrements to zero, a borrow signal is produced. This borrow signal toggles the frequency flip-flop and also reloads the tone counter. Thus, the period of the desired frequency is twice the value of the period register.

The frequency can be calculated by the following:

$$f = \frac{N}{32n}$$

where N = ref clock in Hz
n = 10 bit binary number

The output of the frequency flip-flop feeds into a four stage attenuator. The attenuator values, along with their bit position in the data word, are shown in Table 1. Multiple attenuation control bits may be true simultaneously. Thus, the maximum attenuation is 28 db.

Table 1 ATTENUATION CONTROL

BIT POSITION				
A0	A1	A2	A3	WEIGHT
0	0	0	1	2 db
0	0	1	0	4 db
0	1	0	0	8 db
1	0	0	0	16 db
1	1	1	1	OFF

2. NOISE GENERATOR

The Noise Generator consists of a noise source and an attenuator. The noise source is a shift register with an exclusive OR feedback network. The feedback network has provisions to protect the shift register from being locked in the zero state.

TABLE 2 NOISE FEEDBACK CONTROL

FB	CONFIGURATION
0	"Periodic" Noise
1	"White" Noise

Whenever the noise control register is changed, the shift register is cleared. The shift register will shift at one of four rates as determined by the two NF bits. The fixed shift rates are derived from the input clock.

TABLE 3 NOISE GENERATOR FREQUENCY CONTROL

BITS		
NFO	NFI	SHIFT RATE
0	0	N/512
0	1	N/1024
1	0	N/2048
1	1	Tone Generator #3 Output

The output of the noise source is connected to a programmable attenuator as shown in Figure 4.

3. AUDIO SUMMER/OUTPUT BUFFER

The summer is a conventional operational amplifier summing circuit. It will sum the three tone generator outputs, noise generator output, and an external audio source. The output buffer will generate up to 10mA (see figure 2).

Each tone generator requires 10 bits of information to select the frequency and 4 bits of information to select the attenuation. A frequency update requires a double byte transfer, while an attenuator update requires a single byte transfer.

If no other control registers on the chip are accessed, a tone generator may be rapidly updated by initially sending both bytes of frequency and register data, followed by just the second byte of data for succeeding values. The register address is latched on the chip, so the data will continue going into the same register. This allows the 6 most significant bits to be quickly modified for frequency sweeps.

5. CONTROL REGISTERS

The SONUS has 8 internal registers which are used to control the 3 tone generators and the noise source. During all data transfers to the SONUS the first byte contains a three bit field which determines the destination control register. The register address codes are shown in Table 4.

TABLE 4 REGISTER ADDRESS FIELD

R0	R1	R2	DESTINATION CONTROL REGISTER
0	0	0	Tone 1 Frequency
0	0	1	Tone 1 Attenuation
0	1	0	Tone 2 Frequency
0	1	1	Tone 2 Attenuation
1	0	0	Tone 3 Frequency
1	0	1	Tone 3 Attenuation
1	1	0	Noise Control
1	1	1	Noise Attenuation

6. DATA FORMATS

The formats required to transfer data are shown below.

	REG ADDR			DATA			
1	R0	R1	R2	F6	F7	F8	F9

BIT 0 FIRST BYTE BIT 7

		DATA					
0	x	F0	F1	F2	F3	F4	F5

BIT 0 SECOND BYTE BIT 7

UPDATE NOISE SOURCE (SINGLE BYTE TRANSFER)

	REG ADDR					SHIFT	
1	R0	R1	R2	x	FB	NF0	NF1

BIT 0 BIT 7

UPDATE ATTENUATOR (SINGLE BYTE TRANSFER)

	REG ADDR			DATA			
1	R0	R1	R2	A0	A1	A2	A3

BIT 0 BIT 7