DIGITAL RESEARCH COMPUTERS
P.O.BOX 401565
GARLAND,TEXAS 75040
214-494-1505

INTRODUCTION:

This sound effects board utilizing two of the General Instruments AY-3-8910 programmable sound generators allows your computer to generate an infinite number of real time complex sounds. Each stereo chip's sounds may vary in complexity from simple tones all the way upto triple tones with frequency sweeps mixed with noise and modulated by an envelope control. The types and number of sounds possible is limited only by your imagination and your computer's memory size.

FEATURES:

S-100 Compatible

Two GI AY-3-8910 Sound ICs on board.

Two on board audio amps for stereo effects.

Two low level outputs available for external amps.

Prototyping area on board.

Four I/O ports available.

PC board is solder masked and silkscreened.

Gold plated contact fingers for long life.

This complete kit includes all parts,sockets,ICs,etc.

Board is buffered and liberally bypassed for trouble free operation.

Designed for quick and easy assembly.

Addressable as any 4 of the 256 available ports.

Compatible with Sound Command Language (SCL tm) programs.

Works with either 8080 or Z-80 CPU.

CPU is free to do other tasks once a sound is started.

Programmed I/O; does not conflict with system memory.

PARTS LIST

2	40 pin IC sockets
8	16 pin IC sockets
10	14 pin IC sockets
11	2.2K resistors
2	1K resistors
2	2.7 ohm resistors
13	.01 mfd Bypass Disc Caps (value not critical)
2	.1 mfd. Caps
3	6.8 mfd >15 V Tantalum (value not critical)
1	.33 mfd >25V Tantalum (value not critical)
4	100 mfd 16V Electrolytic
2	10K to 50K Trim pots
2	TO-220 Heatsinks (with hardware) for voltage regulators
1	7805 +5VDC Voltage Regulator
1	7812 +12VDC Voltage Regulator
3	7400 (or LS) TTL Gate
2	7404 (or LS) TTL Gate
1	7410 (or LS) TTL Gate
2	7474 (or LS) TTL Flip flop
2	8242 (or 74LS266) TTL Gate
3	74367 (or LS or 8T97) Tri-State Buffers
2	LM380 (612162-1) Audio Amps
2	GI AY-3-8910 Sound Chips
1	PC Board

General Construction Hints

For soldering we recommend a 32 watt soldering pencil. DO NOT use a soldering gun!!! Use small diameter (such as 22 Ga.) rosin core 60/40 alloy solder.

Keep the soldering pencil CLEAN with a wet sponge or cloth.

After components such as resistors or caps have been soldered, use a small pair of diagonal cutters to remove the excess lead length.

Observe polarities of all tantalum and electrolytic caps.

For a professional appearance insure that all sockets are mounted flush against the PC board.

LIMITED WARRANTY

Digital Research Computers warrants all components in this kit to be free of defects in material and workmanship for a period of 90 days. Any defective parts must be returned to us and will be replaced at no charge. Any board purchased as a kit which malfunctions during the warranty period and has not been subjected to abuse and has been assembled with reasonable care will be repaired or replaced (our option) at no charge.

Any unassembled kit purchased from us may be returned within 14 days of receipt for a FULL "no questions asked" refund. NO reason is necessary. The above limited warranties also apply to kits assembled by DR:Computers.

Any board which is not covered by the above warranties will be repaired at a cost commensurate with the work required. This charge will not exceed $20 without prior approval.

This warranty is made in lieu of any other warranty expressed or implied and is limited in all cases to the repair or replacement of the kit involved. Absolutely no claims are made for the suitability or reliability of this product in any application or system, and we are in no way responsible for any case of consequential damage. If the terms of this warranty are not acceptable, the product must be returned for a refund. Retention of the product by the customer shall constitute a agreement that he has read and accepts the terms of this limited warranty.

ASSEMBLY INSTRUCTIONS

( ) Give the PC board a good visual inspection for any obvious defects. A few minutes spent here could save hours of work later.

( ) Using an ohm meter, insure that there are no shorts between Buss Pins 1,2,and 50.

(✓) Install and solder 14 pin sockets in IC locations 1,2,3,10,11,12,13,14,17,and 18. Note that there is a notch or indentation on each of the IC sockets. These should be oriented in the same direction as the small dot on the PC board. Each dot signifies Pin #1 of each IC.NOTE: Do Not install sockets in locations 15 and 22. To do so would eliminate the heatsinking for the LM380 audio amps.

(✓) Install and solder 16 pin sockets in locations 4,5,8,9,19,20,21, and S1.

( ) Install and solder 40 Pin sockets in locations 6 and 7.

( ) Install and solder the 2 LM380 audio amps in locations 15 and 22. Insure that Pin one of the devices is adjacent to the dots on the board.

( ) Install and solder the 13 bypass caps in locations C1-C4,C6-C9,C12-C16. Note: capacitor C5 is not used.

( ) Install and solder .1 mfd caps in locations C10 and C17.

( ) Install and solder the four 100 mfd caps in locations C11,C18,C22 and C23. Observe polarity.

( ) Install and solder the 6.8 mfd >15V tantalums in locations C19,C20,and C24. Observe polarity

( ) Install and solder the .33 mfd >25V tantalum cap at location C21. Observe the polarity.

( ) Install and solder the 11 2.2K resistors in locations R1,4-6,9-15.

( ) Install and solder the 2 1K resistors in locations 2 and 3.

( ) Install and solder the 2 2.7 ohm resistors in locations R8 and 17.

( ) Install and solder the 2 trim pots at locations R7 and R16.

( ) Mount and solder the 7805 voltage regulator with its heatsink in location Z16 using the hardware supplied.

( ) Mount and solder the 7812 voltage regulator with its heatsink in location Z23 using the hardware supplied.

( ) Plug the board in your system (preferably using an extender board). Using either regulator mounting tab as ground, measure the output of each regulator. The 7805 should measure +5VDC + or - 5%. The 7812 should measure +12VDC + or - 5%. If either regulator is defective, replacement is necessary before completing the board.

( ) Insert the 7400s in locations 10,12,and 13. Be extremely careful to match pin #1 of each IC (denoted by a small notch or indentation) with the small dot on the board.

( ) Insert the 7404s in locations 11 and 14.

( ) Insert the 7410 in location 2.

( ) Insert the 7474s in locations 1 and 3.

( ) Insert the 8242s (or 74LS266) in locations 17 and 18.

( ) Insert the 74367s in locations 19,20, and 21.

( ) Insert the GI AY-3-8910s in locations 6 and 7. Note these are MOS devices and should be handled using static electricity precaution.

( ) It is a good idea to repeat the above procedures again, double checking your work for mistakes. Also, recheck the voltage regulators for proper operation.

( ) Before connecting the speakers, set the trim pots to approximately the center of their rotation.

INITIAL SET UP AND USE

The first thing that you should do before trying to use the Sound Board is to read and study the GI AY-3-8910 Data Manual included with the kit. This book will give you an understanding of how the sound chip works. Try to memorize pages 8 and 9 of the manual, this will make programming easier. To understand the sound chip is to understand the sound board.

Hidden Gotchas in the Data Manual:
The internal registers are referred to by their octal number in the manual. Remember Basic talks in decimal, and most system monitors talk in hex. It is recommended that you obtain a good 8080 or Z-80 programming card which contains hex, octal, decimal, and ASCII equivalents.
The table shown on page 47 on the data manual is for 1.79 mhz clock compared to the 2.00 mhz clock used from the S-100 buss.
Resister 7 (Enable Resister) is all important, and is active low. Nothing will work unless this resister is treated with utmost care. Also R7 controls the direction of the I/O ports A and B.

All of our software presumes that the board is addressed at ports 80H thru 83H (128 thru 131 decimal). To select this port address, location S1 (address select), position A2 thru A6 should be jumpered or closed. Leave position A7 open. If you are using a Z-80 CPU at 4 Mhz you must jumper or close position labeled "4 Mhz". This inserts two Wait states when the sound board is accessed.

If port 80H conflicts with any of your software you may use S1 to select any other group of 4 I/O ports. S1 works like this: the binary representation of the desired block of ports is represented by A2-A7. A logical zero is a closed switch and a logical one is an open switch, i.e. ports 0 thru 3 would be selected when switch positions A2 thru A7 are closed; ports B0 thru B3 are selected when positions A2,A3,and A6 are closed.

Two types of audio outputs are provided, low and high level. The low level outputs are desigred to drive an external amplifier such as a home stereo. These low level outputs are marked "A and B Input" since they are inputs into your stereo. The high level outputs are designed to drive speakers directly. Full output is approx. 2 to 5 watts per side. Speakers are connected at E3-E4 and E7-E8 for chips A and B respectively. A speaker impedance of 8 ohms or more is recommended. Sheilded audio cable is necessary to connect the low level inputs (E1-E2 and E5-E6) to your external amp. It is very important to observe proper grounding practices. All even E numbers are ground, and all odd are signals.

The most important thing to remember is that the GI chips are subsystems of their own. These subsystems are totally controlled by the contents of each sound chip's 16 resisters. The main function of your computer is to initialize or modify the contents of these resisters. As the resisters change the sounds produced change. The main function of the S-100 Sound Effects Board is to provide the hardware interface between your computer and the sound chips.

The internal resisters are not directly addressable. Resister addressing is accomplished by Outputing the Resister number as data to the EVEN port numbers. The resister number is latched internally by the sound chip and anticipates that the data to be loaded to that resister will be Output to the ODD port on the next board access. So remember, it takes two Outputs to load data into the sound chips resisters. Exception: since the Resister number is latched, it is possible to continually modify the SAME resister's contents once the Resister number is specified. This is nice for frequency sweeps, etc.

For most systems the quickest way to talk to the S-100 Sound Effects Board is via BASIC. If your BASIC has INPUT and OUTPUT commands, then you are in business! If you cannot use BASIC then it will take the machine code instructions INP (DBH) and OUT (D3H) to talk to the board. Why do you need any input instructions? Inputing from the ODD ports reads the contents of the last latched Resister. This is a programming aid, allowing you to read the registers' data. This is useful when you come

up with a good sound and want to know exactly how you did it. It is also nice when a sound does not work.

The following is actually two BASIC programs. Run at 10 is a program to write into Sound Chip A. It asks first for which register you want to modify, then it requests the data. Run at 100 is a program to dump all 16 register.

```
10 PRINT "ALL VALUES ARE DECIMAL"
15 PRINT "IT TAKES TWO OUTPUTS TO WRITE TO A REGISTER"
20 PRINT "ONE TO SPECIFY THE REGISTER"
22 PRINT "THE OTHER TO SPECIFY THE DATA"
24 PRINT "THE REGISTER # IS SENT TO THE EVEN PORT"
25 PRINT "THE DATA IS SENT TO THE ODD PORT"
26 INPUT "REGISTER #";R
30 OUT 128,R
40 INPUT "DATA";D
50 OUT 129,D
60 GOTO 26
100 J=0
110 FOR I=1 TO 16
120 OUT 128,J
130 PRINT J,INP(129)
135 J=J+1
140 NEXT I
150 END
```

By loading the following registers with the accompaning data a nice chime sound is produced.

Register #	Data
2 (decimal)	75
4	107
7	248
8	16
9	16
10	16
11	255
12	43
13	8

Set all other registers to 0. Register 14 and 15 are I/O ports and not used in sound generation.
Refering to page 8 of the Data Manual, we are loading two differant tones into channels B and C (registers 2 and 4). Register 7 (Enable reg.) is loaded with 248 which is F8H leaving bits 0, 1, and 2 LOW. This enables all three tone channels, disables Noise on all three channels, and sets I/O ports A and B for input only. The values loaded in registers 8, 9, and 10 set the amplitude. The values loaded in registers 11 and 12 set the modulation Envelope Period. The value loaded in register 13 determines the Envelope shape as shown on page 26 (third waveform).

To create the same sound via assembly language you would use the following program.

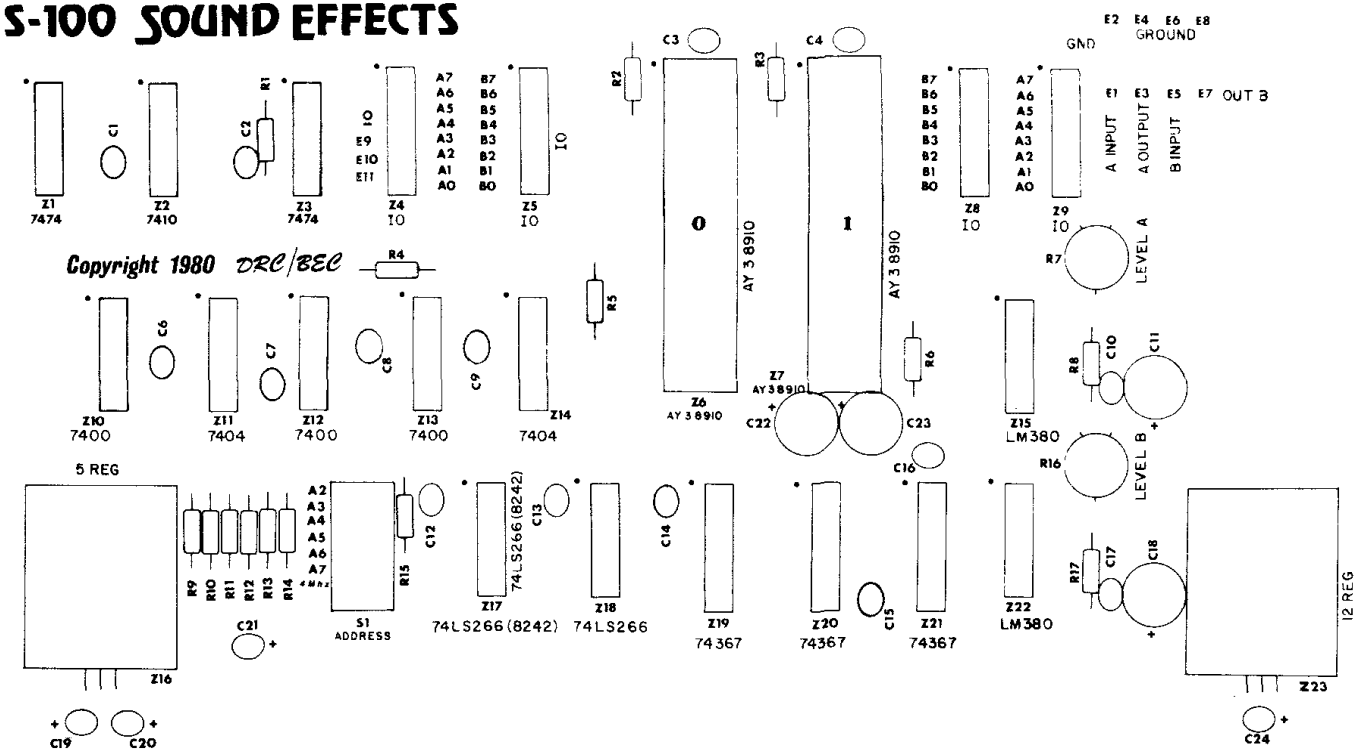
```
0000    3E 02    MVI A,02H        ;LOAD A WITH REG.#
0002    D3 80    OUT 80H          ;LATCH REG.#
0004    3E 4B    MVI A,4BH        ;LOAD A WITH DATA
0006    D3 81    OUT 81H          ;PUT DATA IN REG.
0008    3E 04    MVI A,04H        ;MORE OF THE SAME
000A    D3 80
000C    3E 6B
000E    D3 81
0010    3E 07
0012    D3 80
0014    3E F8
0016    D3 81
0018    3E 08
001A    D3 80
001C    3E 10
001E    D3 81
0020    3E 09
0022    D3 80
0024    3E 10
0026    D3 81
0028    3E 0A
002A    D3 80
002C    3E 10
002E    D3 81
0030    3E 0B
0032    D3 80
0034    3E FF
0036    D3 81
0038    3E 0C
003A    D3 80
003C    3E 2B
003E    D3 81
0040    3E 0D
0042    D3 80
0044    3E 08
0046    D3 81                     ;PFM
0048    C9           RET          ;GO BACK. ALL DONE
```

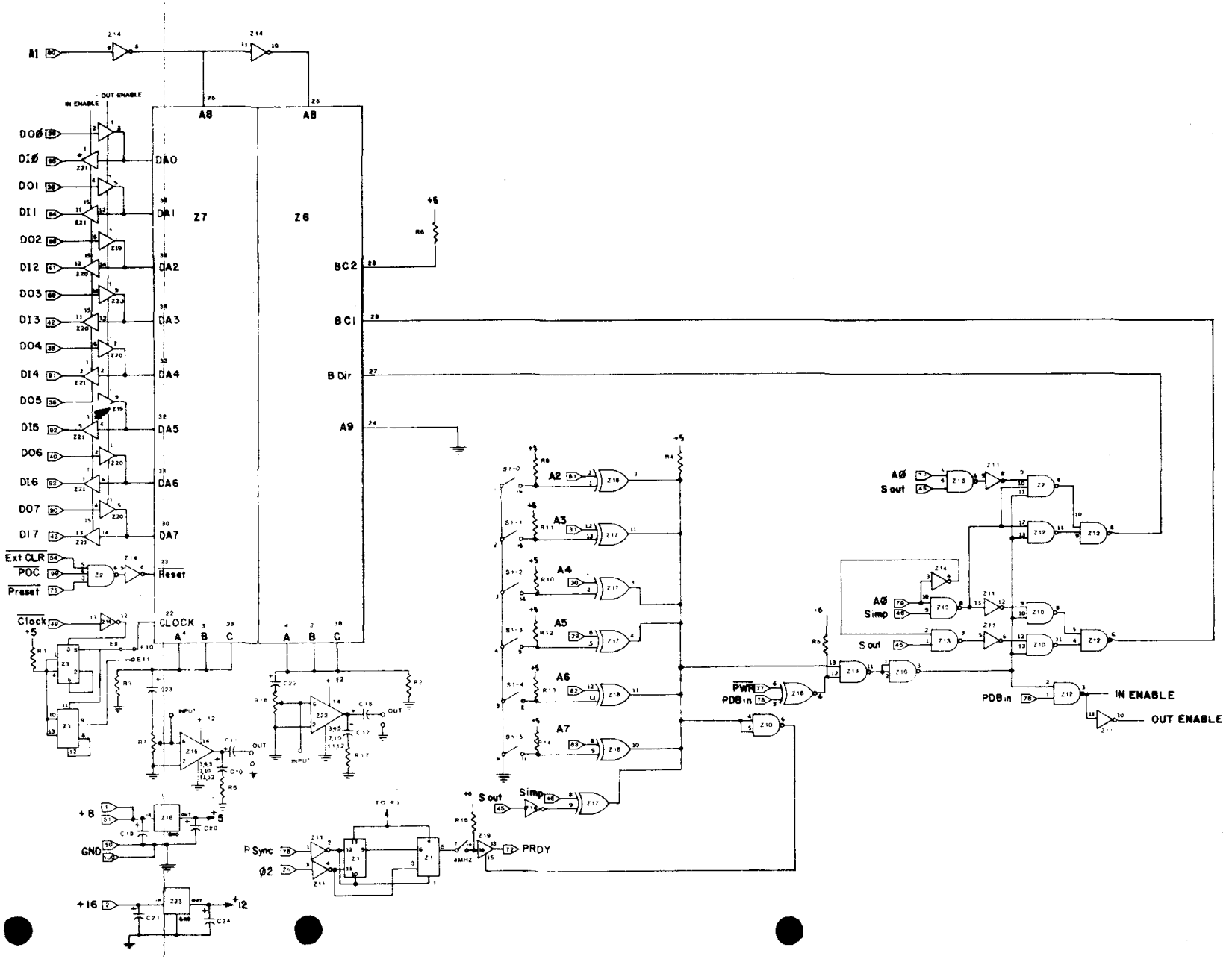
THEORY OF OPERATION

The S-100 Sound Effects Board is basically a parallel I/O board. The basic functions are Port Decode, Data Buffering and Enabling, Buss Control, Power Supplies, Wait States, and Clock conditioning. This board varies from the above by including audio amps. Port decode is accomplished by Z17 and 18 and switch S1. Data buffering is handled by Z19, 20, and 21. Buss control is developed by several gates of Z2 and Z10 thru 14. These gates convert control signals from the buss into signals the sound chip interprets into read and write commands. Z16 supplies regulated +5VDC to all digital devices on the board. Z23 supplies regulated +12VDC for the audio amps. Z1 provides Wait states necessary for processors faster than 500 ns clock periods. Z3 divides Buss Pin 49 (Clock) by two and optionally by four. This requires a cut of the trace from E9 to E10, and a jumper installed from E10 to E11. This will lower all

frequencies and noises, it is not recommended unless your application requires this effect. Z15 and Z22 provide high level outputs capable of driving speakers with "room filling sound". Low level outputs are also brought out. The sound chips are Reset by Buss pin 75 (Preset), Buss pin 99 (POC), or Buss pin 54 (Ext Clr). If your computer does not provide Ext Clr, we highly recommend you provide this funtion.

S-100 SOUND EFFECTS




A1
IN ENABLE
OUT ENABLE
A8
A8
DOØ
DIØ
DO1
DI1
DO2
DI2
DO3
DI3
DO4
DI4
DO5
DI5
DO6
DI6
DO7
DI7
DA0
DA1
DA2
DA3
DA4
DA5
DA6
DA7
Z7
Z6
BC2
BC1
B Dir
A9
+5
Ext CLR
POC
Preset
Reset
Clock
CLOCK
A
B
C
A2
A3
A4
A5
A6
A7
S out
Simp
AØ
PWR
PDBin
IN ENABLE
OUT ENABLE
+8
GND
+16
+5
+12
P Sync
Ø2
PRDY
4MHZ
INPUT
OUT

INTRODUCTION

SCL is a machine language interpreture for use with the S-100 Sound Effects Board. It consists of two functional modules: the Register Examine/modify routine and the Sound Command Language Interpreter.

EXAMINE/MODIFY ROUTINE

The internal architecture of the AY-3-8910 sound IC is composed of 16 registers that control the period, amplitude and envelope of three seperate oscillators. The chart in FIG. #1 lists these registers. Included also are noise frequency and mixer registers that allow a selected band of noise frequencies to appear on any or all of the three channels. Note that the Enable Register is negative logic (low true), so that FF turns all channels off while 00 enables tones and noise on all three channels. The Examine/Modify Routine allows you to individually display the contents of each register and modify it with any two digit Hex number. The AY-3-8910 DATA MANUAL provided outlines the detailed operation of the Programmable Sound Generator (PSG) IC's and explains each register. The manual uses octal notations while all references in this manual and SCL are hexadecimal. Thus Register 10 in octal 8 is in Hex, etc. up to register 17 octal which is Hex "F".

Examine/Modify allows you to work with either PSG chip on the board. The operation of the Examine/Modify routine is explained in greater detail later.

SCL INTERPRETER

The SCL interpreter allows the generation of complex sound effects with minimal programming effort by providing simple access to the PSG register set for loading values, incrementing, decrementing, looping and time delay. A program can be written to any open section of RAM memory in your machine and recalled, modified or played upon command. A more detailed analysis is included in "Using SCL".

INTERFACING SCL TO YOUR SYSTEM

The SCL software accesses the users I/O drivers via vectors passed to it in the DE and HL register pairs at startup time. The user must provide a short subroutine to fetch a character from a console device (keyboard) and pass it in register A. In addition, a subroutine to print the ASCII character on the console (printer, video, etc.) is required. The pointer to the "fetch a character" subroutine is placed in the HL refister pair while the pointer to the "Print a character" subroutine is placed in the DE register pair before entering the SCL program. SCL is then

ROM. The following is an example of a typical program.

```
                ;THIS ROUTINE SETS UP THE REGISTERS FOR SCL
                ;AND PERFORMS INPUT AND OUTPUT FUNCTIONS.
                ;THIS ROUTINE EXPECTS YOUR DATA PORT AT 01
                ;AND YOUR STATUS PORT AT 00.  RECEIVER FLAG
                ;AT BIT 0 AND TRANSMITTER FLAG AT BIT 7.
                ;WE ASSUME YOUR SYSTEM MONITOR TO BE AT
                ;LOCATION 0F000H.  REPLACE WITH YOUR MONITOR
                ;ADDRESS.

0000 210C00     START   LXI     H,000CH ;FETCH A CHAR.
0003 111800             LXI     D,0018H ;PRINT A CHAR.
0006 CD00E0             CALL    SCL     ;SCL ADDRESS
0009 CD00F0             CALL    MON     ;SYSTEM MONITOR ADD.(TYP)
000C DB00       INKS    IN      KS      ;STATUS PORT
000E E601               ANI     RDA     ;CHECK DATA AVAIL.
0010 C20C00             JNZ     INKS    ;LOOP
0013 DB01               IN      DATA    ;INPUT DATA
0015 E67F               ANI     PARITY  ;STRIP PARITY
0017 C9                 RET             ;GO BACK
0018 F5                 PUSH    PSW     ;SAVE A AND FLAGS
0019 DB00       OTPS    IN      PS      ;STATUS PORT
001B E680               ANI     TBE     ;CHECK BUFFER EMPTY
001D C21900             JNZ     OTPS    ;LOOP
0020 F1                 POP     PSW     ;GET A AND FLAGS
0021 D301               OUT     DATA    ;OUTPUT DATA
0023 C9                 RET
0024                    END
```

USING SCL

The following pages explain how to use the Examine/Modify and Sound Command Language Interpreter. The mnemonics of SCL were assigned to as closely approximate the actual operations as possible and allow for quick association to minimize learning time.

time.

INITIAL COMMANDS

When SCL is first executed at startup, the Examine/Modify Program (EMP) will prompt the user with "REG?" and wait for the user to enter a register number in Hex to examine. The register number desired is entered followed by a space whereupon upon EMP displays the contents of that register in Hex format followed by a question mark. If the contents are to be changed the new two digit Hex code is entered followed by a "space" or "return" character. If no change is required the "space" or "return" key leaves the register as is and once again prompts the user with "REG?". The following examples examine two of the PSG registers in chip "0" and modify one of them.

```
REG?          Enter "00" (space)
REG 00=00?    Enter AA (space)
REG?          Enter "01" (space)
REG 01=00?    Enter (space) or (return)
REG?
```

In the above example register 00 (Channel A Fine Tune) was changed from 00 to AA while the contents of register 01 were

unchanged at 00. Register numbers 00 through 0F apply to the registers in PSG chip "0" while register numbers 10 through 1F apply to PSG chip "1".

The user has the option of entering one of five special characters to the "REG?" prompt: Exclamation Point (!) to exit SCL (or REBOOT CPM); a capital "L" to access the sound effects library (explained later); capital "M" to write into memory; capital "P" to play the sound from memory and capital "R" to display chip registers. A period (.) will return the user from any of te modes to EMP mode. In the case of "M" or "P" the user will be prompted with "ADDR?" which is a request for the address in RAM where the desired function is to be exercised. This address loaction in the "M" mode, SCL will assume FF00. A period (.) will return the user to EMP mode.

"M" MODE

"M" mode allows you to write in SCL format into any available RAM area by entering the first location of that area following the "ADDR?" prompt. SCL will examine that location and display the contents of that Hex address. Subsequent strokes of the "Return" or "Space" keys will step sequentially through the RAM starting at the desired location. Any location can be changed by typing a two digit Hex number followed by a "Space" or "Return". The minus key (-) allows the user to step backwards through memory. A

Period (.) will cause SCL to leave the memory area and return to the EMP mode. SCL code is relocatable. A program can be moved from one RAM area to another and still be played with the "P" command.

P" MODE

"P" mode allows you to play the program you have entered into a RAM area one time. The play mode starts at the Hex address entered directly following the "ADDR" prompt and continues until it encounters a return command in the program at which time it re-enters the EMP mode with the "REG?" propmt.

"L" MODE

The "L" mode is a library retrieve mode and allows the user to play pre-programmed sounds in RAM or external ROM. If an "L" is entered the user is prompted with "INDEX" requesting the index number of the desired sound. Once the index number is entered the sound is played and the user is repromted for another index number. The method for calculating the index numbers is covered in "Library Call Section". A period (.) will return the user to the EMP mode.

"R" MODE

"R" mode dumps the contents of all registers of both PSG's on the

console device to allow review.

SCL LANGUAGE

The SCL interpreter allows the user to process a chain of SCL commands placed in either RAM or ROM. The chain defines the control of the two PSG chips on the S-100 Sound Effects Board. Commands fall into the following categories:

- Audio Channel Setup (Period and amplitude)
- Audio Channel Update (arithmetic)
- Enables
- Noise Setup
- Noise Update (arithmetic)
- Envelope Control
- PSG Chip Select
- Delays and Loops
- Individual Register Update (arithmetic)
- Library Call

SCL does not allow for GOTO/Jump control statements or Conditional Test statements. The interpreter does not allow the user to base the sound effect on the results of conditions external to the program. Note the commands deal with the PERIOD

of a tone, the inverse of its frequency (P=I/F).

AUDIO CHANNEL SETUP

Set Channel A (format 1M PP VV)

This command sets Channel A to a PERIOD of MPP Hex and and amplitude VV Hex. The Hex values are entered into registers 0,1 (for period) and 8 (amplitde) of the PSG chosen by the PSG Chip Select command discussed later.

Example: 13 26 05 would set register 0 to
26 Hex register 1 to 03 Hex and register 8 to 05.

A very important part of this command is the value of the most significant bit of the amplitude Hex value, VV. Values from 00 to 0F give a 16 level amplitude control that can be changed only by a revision of the hex value. If amplitude control by the envelope generator is desired, then the most significant bit must be high (Logic 1). Thus to control the amplitude of Channel A with the Envelope Control would then become 13 26 10. Since the amplitude value is assigned by the Envelope Control the value of the least significant amplitude bit can be ignored.

Set Channel B (format 2M PP VV)

This command performs the same functions on Channel B of the selected PSG chip and loads Hex values in Registers 2, 3 and 9.

Set Channel C (format 3M PP VV)

This command performs the smae functions on Channel C of the selected PSG chip. It loads registers 4, 5 and A.

AUDIO CHANNEL UPDATE

After the values of Period and Amplitude have been assigned to the three audio channels certain arithmatic operations can be performed on them using the following commands:

ADD TO PERIOD AND/OR AMPLITUDE of Channel A (format 40 XX YY).

XX is a Hex value that is added to MPP which because of the period-frequency inversion relationship, decreases the frequency. Adding a number greater than 80 Hex changes the sign of the operation and subtracts that value from MPP (2's compliment) thereby increasing the tone frequency. The same is true of the Amplitude registers where YY is a Hex value added/subtracted to VV. These commands are usually used in a loop and repeated several times to obtain upward or downward frequency and amplitude sweeps. It is not usually desirable to make the values of XX or YY too large because the tone registers will quickly be

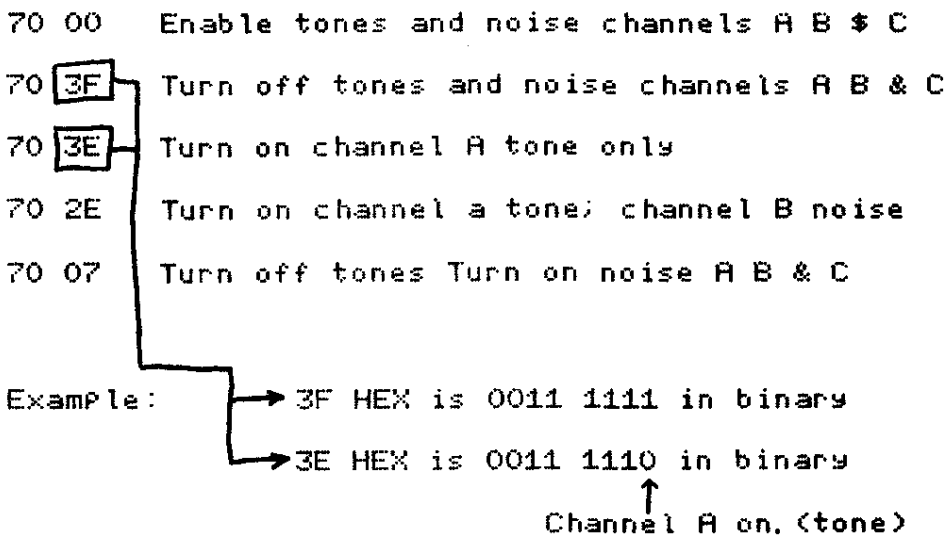
stepped over the top and wrap back around.

ADD TO PERIOD AND/OR AMPLITUDE of CHANNEL B (format 50 XX YY) performs the same operation on Channel B of the selected PSG chip.

ADD TO PERIOD AND/OR AMPLITUDE of Channel C (format 60 XX YY) performs the same operation on Channel C of the selected PSG chip.

ENABLES (format 70 NN)

An inspection of the PSG registers chart will disclose that register 7 allows for turning the noise and tones on or off on the three channels and selecting In/Out commands of the two I/O ports. SCL addresses the Enables Register with a 70 NN format where NN is a Hex value placed in Register 7. It should be noted that the Enables Register is Low True so that all inputs low (0) would turn on all three channels with noise and tone mixed and allow both the I/O ports to be written to. Tone A is the least significant bit of a dual Hex word while IO B is the most significant bit. [These examples will help explain the 70 command in greater detail:]. Since SCL does not concern itself directly with the I/O ports and they have nothing to do with the production of sound we will disregard the effect of 70 codes on I/O commands.



Actually since the two highest bits (B6 &B7) do not matter the values could just have easily been assigned as 70 C0 ; 70 FF; 70 FE; 70 EE and 70 C7. The term "enable" might be better described as "mixer" since it is advised that to turn off a channel completely the amplitude for that channel be assigned a value of 0.

NOISE SETUP (format 80 NN)

The PSG generates a noise spectrum whose frequency is his somewhat akin to the center frequency of a notch frequency whose value is fn= f clock/16NN 10 or the input clock frequency (usually 1 mhz) divided by the product of 16 times the DECIMAL

equivalent to the Hex value NN. Since register 6 to which the "80" code addresses itself, is only 5 bits the highest value for NN would be 1F Hex or 31 decimal. With a 1 mhz input frequency the noise could be as low as 2 khz or as high as 60 Khz.

NOSIE UPDATE (format 90 XX)

Once the noise frequency has been dstablished by the 80 command a Hex value can be added/subtracted to this frequency by the format 90 XX where XX is a Hex value added to the Hex frequency designator NN. As in the case of the Period and Amplitude commands values greater than 80 Hex change the sign and subtract from NN. Since NN represents an inverse function, being the period of the noise frequency, adding XX to NN will actually decrease the noise frequency. The 90 command is most often used in loops to sweep the noise frequency.

Example: 90 02 would add 2 to the noise period.
90 FE subtracts 2 from the noise period.
(2's compliment).

ENVELOPE CONTROL

Since most sounds have a defined attack, sustain and decay pattern the ability of a sound generator would be severly limited

by a lack of Envelope Control. The AY-3-8910 PSG provides for Envelope Control with PSG registers B, C & D. The SCL format is AX HH LL where HH LL is the Hex Period of X type. A study of the chart #4 will disclose that there is in essence eight different types of envelopes. It should be noted that while the chart shows ten, there are duplications. It should also be noted that a value for X of 0,1,2,3, or 9 Hex will all give the same basic envelope: a sharp attack (instant) with a decay whose value is determined by HH LL. The higher the value of HH LL the longer the decay time. To calulate the length, use the formula fe= f clock/(256) HH LL (base 10) where the clock frequency is divided by the product of 256 times the HH LL value converted to decimal.

It is the envelope that gives different characteristics to different types of sounds. Careful manipulation of this section will yield more realistic sounds.

PSG CHIP SELECT (format B0, B1, B2, B3)

Up to this point all of the SCL commands have been written in general. With the PSG chip select command the program is told which of the two PSG chips the data is to be written to. The commands are as follows.

1.) B0 selects chip #0 (left stereo channel) with random feature disabled

2.) B1 selects chip #1 (right stereo channel) with random feature disabled

3.) B2 Selects chip #0 random feature enabled.

4.) B3 selects chip #1 random feature enabled.

It is appropiate to explain the Random Feature part of SCL at this time. It allows a pseudo-random number to be placed into designated registers each time the program (or loop) is stepped through. With B2 as a PSG select command any 00 Hex value placed in any of the following SCL command with "randomize" that value:

1,2 or 3 series Period or Amplitude values.

40,50 or 60 series arithmatic operations.

80 series noise frequency

90 series noise arithmatic operations.

HH and LL Hex pairs of the envelope control (especially effective).

In addition "DO" loop values, delay values, and "E" type commands cam be randomized. (More about those operations in their respective sections).

Either chip can be chosen in either Random Feature or Non-Random Feature mode but both chips cannot be addressed or played at the same time. Remember, however, that values can be loaded into a

PSG's registers and they will remain until the value is changed or the reset activated. The tones, amplitude noise and envelope can be set in one PSG and activated, left by the host processor and will play merrily along until it is told to do something else. Thus both PSG's can be made to produce different sounds at the same time if the program is properly written. If all of this sounds confusing, it is... but a little time with the program and the Sound Board will help clear up the mystery.

DELAYS AND LOOPS (DO,DF,DE)

Ah yes! Now the crust of the SCL program. As may already be expected the program runs much faster than real time sounds take so all of the commands written up to this point satisfy the PSG chip but not your ears. Some method must be used to load a value and let it occur in real time. Delays and Loops consist of a DF Hex command (Delay For) and a DO command (Do Loop). Each of these commands is followed by a 2 digit Hex code to assign a value. DO loops can be nested but for each DO there MUST be a DE (loop end) command.

For Example:	DO 10	Do 16 times.
	40 06 00	Add 6 to channel A
	DF 0A	Delay each time 1 millisec.
	DE	End of loop.

The DF command caused SCL to pause for NN * 100 microsounds with a 2 mhz system clock (note this is not necessarily the PSG clock frequency). As can be calculated, the total delay with a single DF command is 256 * 100 microseconds or .256 seconds. Although the DF can be repeated, and easier way is to nest it in a DO loop which multiplies the total time by the value of DO NN where NN is a Hex number. Thus the following loop would delay for 2.56 seconds.

```
DO 0A      DO 10 Times
DF FF      Delay for FF * 100 Microsec.
DE         End of loop.
```

As stated previously DO loops can be nested up to the extent of stack RAM provided by the program calling the SCL (or 63 levels whichever is smaller). Each nested loop uses 4 bytes of stack.

For Example:

```
DO 05      Do 5 times.
40 06 00   Add 06 to channel A Period.
DO 0A      DO 10 times.
DF FF      Delay for .256 seconds.
DE         Inner Loop End.
DE         Outer Loop End.
```

This program would add 06 to channel A; delay for 2.56 seconds then loop four more times for a total of 20.48 seconds. Obviously very long time intervals can be quickly developed. The location of a DE greatly effects the programs' actions.

INDIVIDUAL REGISTER UPDATE

This SCL command formated ER XX YY where XX is added to PSG register and masked with YY. Since it addresses any single PSG register it allows manipulation of that single register only. It can also be used to reset any register by using the format ER OO OO where R is the desired register.

LIBRARY CALL

(Format CA XX) The Library Call command allows pre-programmed routines or sounds to be called from a library ROM and integrated into the current program. At the CA command SCL leaves the current program and jumps to a location in the Library ROM that is four times the Hex value of XX. SCL assumes that the Library ROM lives in the next 1K byte of memory above the location of the SCL ROM. Thus if the system SCL ROM is located at E000 the Library ROM would have to be located at E400. If there was a routine at E440 the command to call that routine would be CA 10. Routines written in the Library ROM must end with a DE command in order for the SCL to re-enter the original program at the spot it

left and continue. The CA command allows access of up to 1024 bytes of library effects in 4 byte increments. Since most SCL programs are more that 4 bytes long this offest presents no problems. The user must remember to locate the beginning of a library program at a ROM location that is a factor of 4. The access to external library sounds greatly increases the power of SCL to immulate complex and realistic sounds. An example would be a library ROM with all of the major and minor chords programmed in and to simply do a CA command to play that chord. One channel could play the melody while the other plays the chords.

MISCELLANEOUS CONTROL GROUPS

The final SCL control command is the EXIT command that consists of an open 00 or FF. It should be remembered that while the EXIT command causes SCL to leave the program and return to this Register Examine/ Modify mode, it does not reset any of the chip registers. If the oscillators, noise and envelope registers are not disabled, the chip will continue to execute the last command it received until it is updated. As you will see from the sample programs certain registers should be disabled/reset before the program is exited.

The following are a few sound effects along with description.

DUAL PHASOR

```
B0      Select left channel random disabled
D0      Do six times
10
05
0F      Set Channel A to period 005 at full volume
20
IF
0F      Set Channel B to period 01F at full volume
70
F8      Enable all Tone Channels
D0
90      D0 90 hex times
40
02
00      Add 02 to Channel A leave volume alone (sweep)
50
02
00      Add 02 to Channel B leave volume alone (sweep)
DF
03      Delay for 300 microseconds each time
DE      End inside loop
DE      End outside loop
70
FF      Disable all channels
00      End Program
```

SMALL WAR

```
B2      Select left channel ramdom enabled
70
F7      Enable Noise on all chanel A only
10
01
10      Put Channel A under Envelope Control
80
55      Set Noise period to 55 hex
D0
08      Do eight times:
90
00      Random noise period
D0
20      Do 20 hex times:
DF      Delay For
FF      FF hex times 100 microseconds
DE      End second D0 loop
A8      Set envelope to type 8
00      With random period in MSB
FF
DF
00      Delay for random period
D0
05      Do 5 times:
DF
```

```
00      Random delay period
DE      End third DO loop
DE      End first (outside) DO loop
70
FF      Disable all channels
10
00
00      Turn Channel A off
00      End of program
```

CALL (EXP1 {,EXP2}) X = CALL (decimal address) 22

IN DE
OUT HL

TYPE READ.ME

NOTES ON USING CP/M VERSION OF SCL

SCLX IS THE COM FILE THAT CONTAINS THE SCL INTERPERTER.
DEMO.COM IS NOT A TRUE COM FILE. IT IS A DEMO FILE CONTAINING
SEVERAL SCL PROGRAMS TO CHECK OUT YOUR SOUND EFFECTS BOARD.

DEMO.COM IS LOADED VIA DDT. IN ORDER TO LOAD , TYPE THE COMMANDS AS
FOLLOWS. ASSUMING THE SCL DISC IS IN THE SELECTED DRIVE.

```
B>A:DDT
DDT VERS 1.4
-IDEMO.COM
-R
NEXT PC
1500 0100
-GO
B>SCLX

REG? P
ADDR? 1000
```

AN "!" EXITS SCL AND REBOOTS CP/M.
DEMO IS LOADED BY DDT AT 1000H. DEMO IS PLAYED BY THE "P" COMMAND
OF SCL.

ANY SCL PROGRAM CAN BE STORED ON A DISC BY USING STANDARD CP/M "SAVE"
COMMAND AND RELOADED USING DDT. THE DEMO.HEX FILE IS A GOOD EXAMPLE.

CP/M IS A TM OF DIGITAL RESEARCH (CALIFORNIA.)

A>

CUTS I/O LINK FOR SCL ORG: ØØØØ

ØØØØ: 21 Øc ØØ 11 15ØØ CD ØØ EØ CD ØØ CØ CD 1F CØ CA
ØØ1Ø: ØC ØØ E6 7F C9 C5 47 CD 19 CØ 78 C1 C9