DREAM SOUND EFFECTS GENERATOR

by Michael Bauer
Deakin University, 1979

Here is a way of giving your computer a diverse range of sounds. It will add a new dimension to games and open up many possibilities for use in its own right; e.g. "music" generation. Many readers will be curious as to why I chose the T.I. 76477 rather than the General Instrument AY-8910chip, or some other scheme. The simplest way to generate sounds, from the hardware point of view, is to tack a digital-to-analog (D/A) converter onto an output port. This requires virtually continuous service from the microprocessor unit (MPU) to produce audio waveforms. Since many of our applications will require a lot of MPU power and speed (e.g: DREAM Invaders) we can't use this method. To take the load off the MPU, it becomes necessary to utilize external hardware for sound generation. Fortunately, there are a few ICs around which perform this function; the main two being the 76477 and the AY-8910.

The 76477 is like a micro-miniature 'Moog' synthesizer on a chip, but it is very very primitive; (see block digram enclosed within dotted lines in circuit). Since it is essentially an analog device (i.e. voltage controlled), it has a lot of external Rs and Cs to define time-constants, etc, and it is not readily able to be interfaced to the MPU bus. Conversely, the AY-8910 is entirely digital and is directly interfaced to the MPU bus. The chip contains a bank of registers which define the frequency of up to 3 oscillators and the noise and envelope generator characteristics. Since the AY-8910 is purely digital, it is highly accurate and stable.

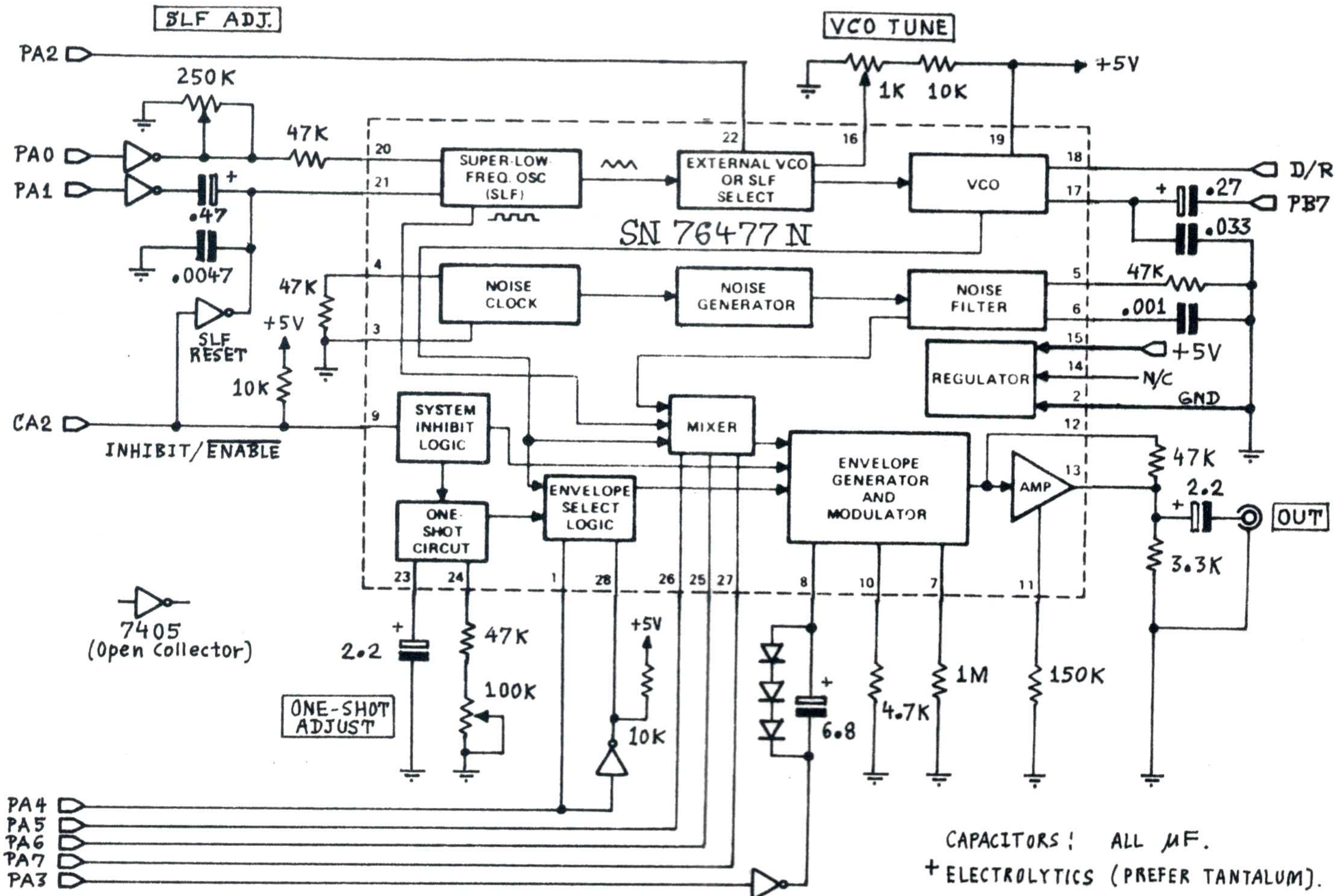
So why choose the 76477 ? Does the author have rocks in his head? Read on for the explanation! The AY-8910 has a severe limitation for use as an effects generator. It is impossible to program the chip so that one of the 3 oscillators can modulate another (AM or FM). A modulation capability is essential to the production of many desired noises. The AY-8910 requires rapid periodic MPU intervention (e.g: using RTC interrupt) in order to produce frequency modulation (FM) effects. The DREAM-6800 already sacrifices a large time-slice (40%) of its processor power to the video display generator (VDG), and its 50Hz relative-time-clock (RTC) is too slow for use with the AY-8910, and so it must, regretably, be ruled out. For applications where the VDG can be turned off and 100% of the MPU power devoted to controlling the AY-8910, however, some fantastic things can be done, especially in the field of music generation where it leaves the T.I. 76477 for dead!

Therefore, like it or not, I was stuck with the task of interfacing the 76477 to the MPU somehow. Only one PIA could be justified, so to cut a long story short, the range of resistance (/capacitance) selectable on each analog-programmed terminal had to be severely compromised. Referring to the circuit, it can be seen that most pins have a fixed R or C, while a few offer a switched choice. Readers who are familiar with the 76477 might jump to the conclusion that I have oversimplified things and that the configuration finally chosen couldn't offer a sufficient diversity of programmable effects. (Wrong!)

Most of the things that you'll ever want to do with the 76477 can be done under MPU control using this configuration. It is quite unnecessary to have a wide choice of such things as: one-shot duration, attack and decay times, output amplitude or noise filter roll-off; so these are fixed. We <u>do</u> need a selection of 'SLF' osc. frequencies and a wide range of VCO freq's. The voltage-controlled osc. (VCO) is controlled by a 'digital-to-resistance' (D/R) converter, utilising half the PIA (port B) and a binary resistor network. Bit-7 selects the VCO timing capacitor, giving one of two ranges. There are 256 possible frequencies for the VCO. Referring to the graph of VCO freq. vs PIA value, we see that accuracy increases towards the high end of each range. The high range covers most of the audio spectrum (100Hz to 10kHz) while the low range gives better resolution in the middle band (250 to 1000 Hz) and goes right down to about 10Hz. In addition to being controlled by the MPU, via the D/R converter, the VCO may be simultaneously frequency modulated or amplitude-modulated by the 'SLF' oscillator.

'DREAM' SOUND EFFECTS GENERATOR (M.J. Bauer 1980)

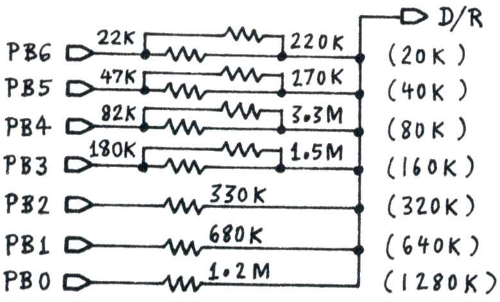
Circuit Diagram



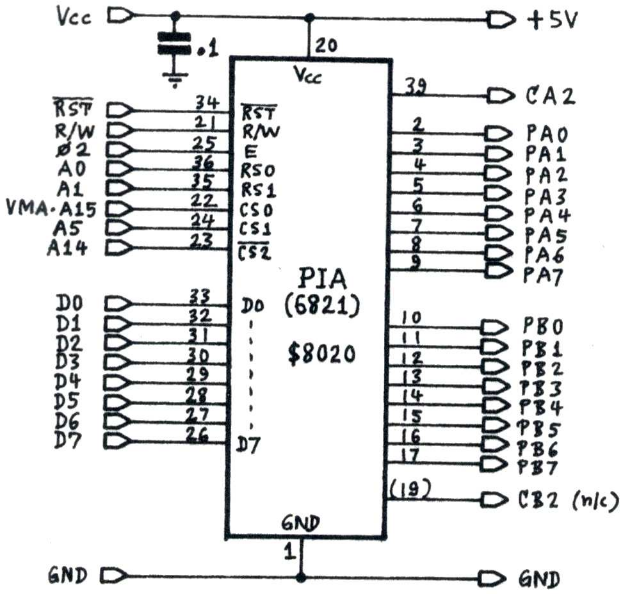
CAPACITORS: ALL μF.

+ ELECTROLYTICS (PREFER TANTALUM).

DIODES: 1N4148 (or similar).



'DIGITAL-TO-RESISTANCE' CONVERTER (D/R)



A couple of refinements have been made to the 76477, externally. Firstly, 3 diodes have been strapped across the envelope-generator capacitor to improve its otherwise abysmal performance. Secondly, an open-collector gate (SLF RESET) discharges the SLF osc. timing capacitor whenever the device is disabled. This forces the VCO to commence oscillating at the same freq. each time the device is triggered (assuming SLF controlling VCO). This feature is essential for many 'one-shot' sound effects.

Finally, this circuit will cost less than the AY-8910, including PIA and Rs and Cs. The 76477 is about $3 to $4, the AY-8910 would have been about $15 (if you can find one at all) and the going rate for a PIA is about $6.

PRACTICAL CONSIDERATIONS

By the time this issue is released, we hope there will be a PCB design available. If not, the circuit is simple enough to whip up on Vero DIP board, especially if you have an expansion board with a PIA already to go. This PIA should be located at $8020, else you will need to alter the software accordingly.

The circuit as it stands does not include an audio amplifier, because the output is designed to be fed into an external amplifier, for example the one in your T.V. set. The 2 transistor output stage given in the T.I. 76477 data sheet does not have enough guts! If you need a separate amplifier, an LM380 should do nicely. The use of 1% tolerance metal film resistors is recommended in the D/R converter, especially if you anticipate playing tunes with it. Set the trimpots initially to the half-way position; these can be tweaked for optimum effect later. By the way, try to get the data sheet and application notes with your 76477, for the useful info therein.

SOFTWARE

The PIA registers are programmed as indicated in the 'Programming Chart'. The PATCH byte is the A-side output register; the data direction register (DDR) being maintained at $FF (all outputs). The VCO-FREQ control byte is the B-side <u>data-direction</u> register (DDR); the output register being kept at 00. Writing a '1' into a bit position in the DDR makes the corresponding I/O line an output, thus grounding an external resistor (or C, if bit-7). Writing a '0' bit in the DDR makes the line an INPUT which is high impedance (floating), thereby effectively removing the resistor from circuit. Neat, huh?

To save you the bother of figuring out how to initialize the PIA, I've written 3 simple subroutines called 'low-level drivers' to handle this task. From the Programming Chart and VCO graph, you can work out the PATCH and VCO-FREQ data desired. Your program must call the subroutines DISAFX and INIZFX at the start, to initialize the PIA. To set up a new patch and enable the device, your program simply needs to load acc-A with the PATCH byte and call the subroutine ENABFX; thus:-

```
LDA A   PATCH
JSR     ENABFX
```

Thereafter, the patch may be altered without disabling the device, if desired, by writing to location $8020 the new patch data. The VCO frequency (and range) may be set or altered at any time simply by writing to location $8022. The low-level driver subroutines are relocatable.

Some unreal zany effects can be produced by sweeping the VCO under program control with any of the following patches (at least): 00, 60, C0, C1, 04, 07. Sweeping is accomplished simply by incrementing/decrementing the VCO freq. at periodic intervals.

TEST & DEMO PROGRAM

The test program generates one of 16 pre-defined effects stored in a look-up table. The listing shows the PATCH and VCO values and the corresponding key to press to get each sound. You can easily replace any or all of the table entries with your own contrived effects. Note that the test program merely sets up the sound effects generator and enables it; the PATCH and VCO-FREQ remain <u>constant</u> until a different key is pressed. All of these effects are being produced by the 76477 on its own, without any MPU assistance whatsoever.

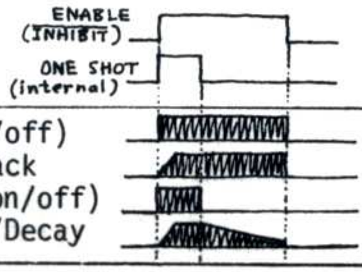
DREAM SOUND EFFECTS GENERATOR -------- PROGRAMMING CHART

'PATCH' Control Byte format (PIA port A, O/P reg.):

Bit:	7	6	5	4	3	2	1	0
	MC	MB	MA	ES	EE	VS	SC	SR

MIXER SELECT BITS			SIGNALS 'MIXED'
MC bit-7	MB bit-6	MA bit-5	
0	0	0	VCO
0	0	1	SLF
0	1	0	Noise
0	1	1	VCO & Noise
1	0	0	SLF & Noise
1	0	1	SLF & VCO & Noise
1	1	0	SLF & VCO
1	1	1	inhibit

ENVELOPE SEL.		ENV. FUNCTION
ES bit-4	EE bit-3	
0	0	Enable only (on/off)
0	1	Enable with Attack
1	0	One-shot only (on/off)
1	1	One-shot w. Att/Decay



VCO Control Voltage Select (VS, bit-2)	
0	EXTERNAL preset adjustment
1	INTERNAL by 'SLF' oscillator

SLF Oscillator Frequency Select			
SC bit-1	SR bit-0	SLF Osc. Freq.	
0	0	AUDIO-1	Preset
0	1	AUDIO-2	Fixed
1	0	LOW-1	Preset
1	1	LOW-2	Fixed

VCO FREQUENCY Control Byte format (PIA port B, DDReg.):

7	6	5	4	3	2	1	0
C	R6	R5	R4	R3	R2	R1	R0

RANGE
1 = 'low' (8Hz - 1kHz)
0 = 'wide' (80Hz - 10kHz)

FREQUENCY
1 = Rx in circuit
0 = Rx not in cct

```
                  *
                  ***************************************************************
                  *   DREAM-6800 SOUND-EFFECTS GENERATOR                        *
                  *   TEST & DEMO PROGRAM +                                     *
                  *   LOW-LEVEL DRIVER SUBROUTINES.                             *
                  *          M.J. BAUER,  1981.                                 *
                  ***************************************************************
                  *
                          TITLE   76477 SOUND FX GEN DRIVERS
                  *
                  FXPIA   EQU     $8020
                  GETKEY  EQU     $C2C4
                  KEYINP  EQU     $C297
                  BADRED  EQU     $0018
                  PAINZ   EQU     $C287              INIZ KEYPAD
                  ADDAI   EQU     $C189              16 BIT ADD A TO PTR (I)
                  I       EQU     $26                16 BIT POINTER
                  *
0200                      ORG     $0200
0200 8D 54        TESTFX  BSR     INIZFX
0202 8D 4C                BSR     DISAFX
0204 BD C287              JSR     PAINZ
0207 B6 8011      WAIT1   LDA A   $8011              WAIT FOR KEYDOWN
020A 2A FB                BPL     WAIT1
020C BD C297              JSR     KEYINP             FETCH KEYCODE ==> A
020F 48                   ASL A                      MULT BY 2
0210 CE 0230              LDX     #TABLE             USE TO LOOK UP TABLE
0213 DF 26                STX     I
0215 BD C189              JSR     ADDAI
0218 DE 26                LDX     I
021A A6 01                LDA A   1,X                SET VCO FREQ
021C B7 8022              STA A   FXPIA+2
021F A6 00                LDA A   0,X                SET PATCH & ENABLE FX
0221 8D 42                BSR     ENABFX
0223 BD C297      WAIT2   JSR     KEYINP             WAIT FOR KEY RELEASE
0226 7D 0018              TST     BADRED
0229 27 F8                BEQ     WAIT2
022B 20 D3                BRA     TESTFX             AGAIN...
                  *
0230                      ORG     $0230
                  * LOOK-UP TABLE:   DATA FOR PATCH & FREQ (16 x 2):
                  *                  ***  KEY  ***  CONTINUOUS ENABLE:-
0230 00FF         TABLE   FDB     $00FF   0  VCO, 1kHz
0232 0040                 FDB     $0040   1  VCO, 5kHz
0234 2000                 FDB     $2000   2  SLF, AUDIO-1 (Preset)
0236 4000                 FDB     $4000   3  NOISE
0238 0440                 FDB     $0440   4  FM, SLF AUDIO, VCO 5KHZ
023A 06FF                 FDB     $06FF   5  FM, SLF LOW-1, VCO 1KHZ
023C C0FF                 FDB     $C0FF   6  AM, SLF AUDIO-1, VCO 1KHZ
023E C183                 FDB     $C183   7  AM, SLF AUDIO-2, VCO 23HZ
                  *               ***  KEY  ***  ONE-SHOT ENVELOPE:-
0240 161A                 FDB     $161A   8  FM, SLF LOW-1, VCO 2KHZ
0242 1714                 FDB     $1714   9  FM, SLF LOW-2, VCO 1.5K
0244 7084                 FDB     $7084   A  NOISE & VCO (30HZ)
0246 D10D                 FDB     $D10D   B  AM, SLF AUDIO-2, VCO 1K
                  *               ***  KEY  ***  ATTACK/DECAY ENVELOPE:-
0248 780D                 FDB     $780D   C  NOISE & VCO (1KHZ)
024A 1C70                 FDB     $1C70   D  FM, SLF AUDIO-1, VCO 9KHz
024C D987                 FDB     $D987   E  AM, SLF AUDIO-2, VCO 50HZ
024E DF20                 FDB     $DF20   F  EVERYTHING! (well, almost!)
```

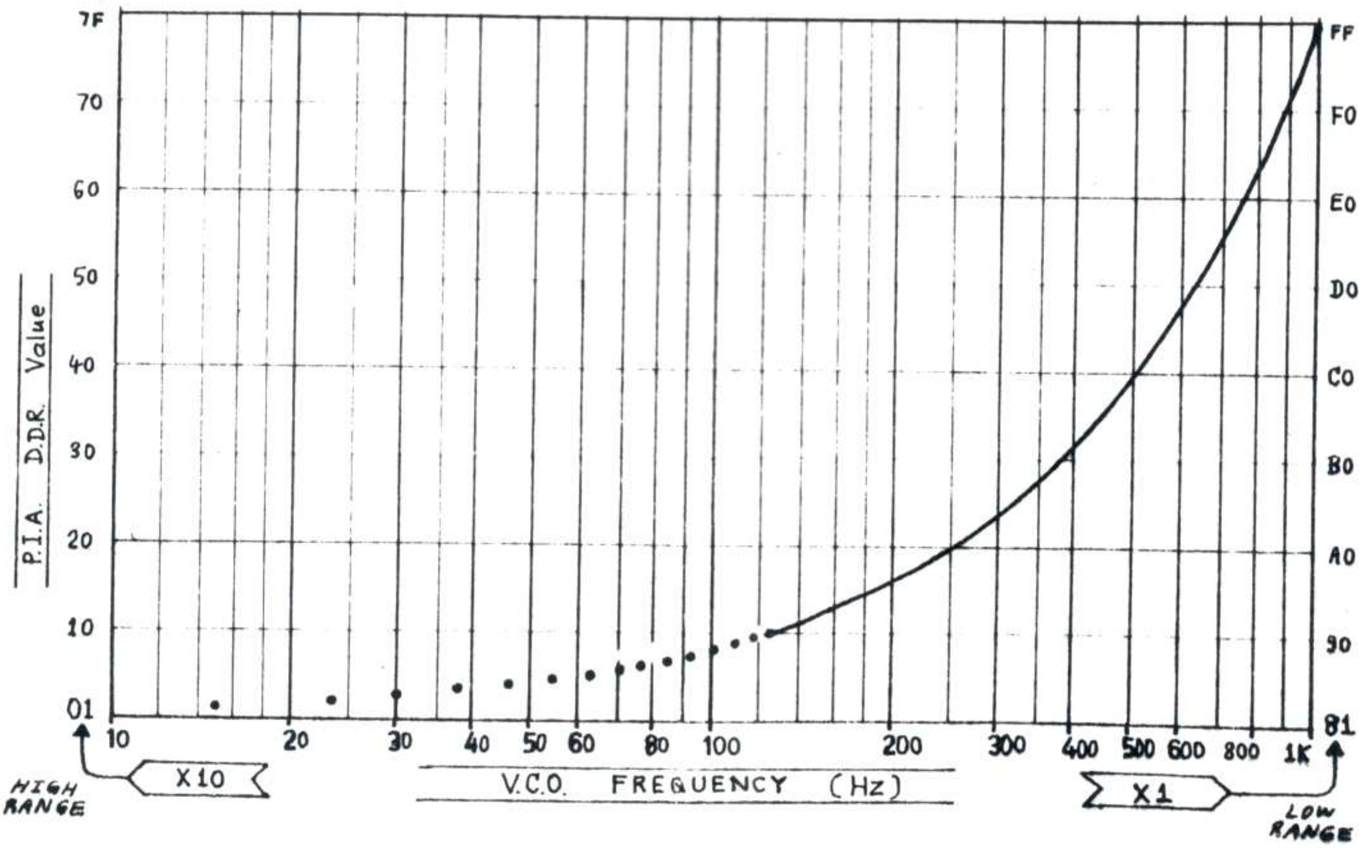
```
                   * LOW-LEVEL DRIVER SUBROUTINES:
0250                        ORG      $0250

                   * DISABLE SOUND EFFECTS GENERATOR:
0250 C6 3C         DISAFX   LDA B    #$3C
0252 F7 8021                STA B    FXPIA+1
0255 39                     RTS

                   * INITIALIZE SOUND-FX GEN (PIA-B DDR = VCO-FREQ.)
0256 C6 04         INIZFX   LDA B    #4              ACCESS O/P REG.
0258 F7 8023                STA B    FXPIA+3
025B 7F 8022                CLR      FXPIA+2         O/P LINES LOW
025E 7F 8023                CLR      FXPIA+3         ACCESS DDR
                   * SET VCO FREQ & RANGE (ACC-A ==) DDR)
0261 B7 8022                STA A    FXPIA+2
0264 39                     RTS

                   * ENABLE SOUND-FX GEN. (ACC-A ==) O/P REG)
0265 CE 8020       ENABFX   LDX      #FXPIA          GET PORT ADRS
0268 C6 38                  LDA B    #$38            INHIBIT & SEL DDR
026A E7 01                  STA B    1,X
026C C6 FF                  LDA B    #$FF            WRITE DDR (ALL OUTPUTS)
026E E7 00                  STA B    0,X
0270 C6 3C                  LDA B    #$3C            SEL O/P REG
0272 E7 01                  STA B    1,X
0274 A7 00                  STA A    0,X             WRITE O/P REG
0276 C6 34                  LDA B    #$34            ENABLE FX
0278 E7 01                  STA B    1,X
027A 39                     RTS

                            END
```



MELODY MAKING

The D/R converter should give adequate resolution over the top 2 octaves of the LOW range (250Hz to 1kHz, nominally) for musical purposes. Due to variations in resistor values and unknown 76477 anomalies, the VCO values for the musical scale cannot be accurately calculated. So far, I have not been able to investigate the musical potential of the sound generator due to lack of time, but here are some guidelines for experimenters who have a little music theory and like to dabble in machine language programming.

I suggest you write a program which accepts from the keyboard and displays 2-digit hex numbers, and 'plays' the corresponding VCO frequency. Then, with the aid of a piano, organ, guitar, bag-pipes, stylophone (or whatever), determine by trial-and-error those values which lie on the musical scale. These values can be used to construct a look-up table for use with your melody-making programs. Lets hear from you if you succeed in developing any useful software for the sound generator.

The sound generator described in this article is a very inexpensive enhancement to the DREAM-6800, which opens up many possibilities for experimentation. It should increase your motivation to learn about machine-code programming, and interfacing the digital and analog worlds.

SOUND EFFECTS for 'DREAM INVADERS'


```
Turret hit / Alien landed sound:
0700   86 FF BD 07 D6 86 07 BD 07 E5 C6 40 7F 00 20 7D
0710   00 20 27 FB 7A 80 22 7A 80 22 5A 26 EF 7E 07 D0

Initialization of I/O ports:
0720   BD C2 87 7E 07 D6

Alien descending sound:
0726   7C 00 9A CE DF 20 7E 07 51

Fire missile sound:
073D   7C 00 B3 CE 78 0E 7E 07 51

Alien hit sound:
0746   CE 16 14 BD 07 51 C6 03 7E 05 E4

Make sound specified in X-reg.:
0751   BD 07 D0 DF 26 96 27 B7 80 22 96 26 7E 07 E5

Essential modification :
0760   96 B3 91 B4 2D 09 96 21 84 78 26 03 7E 03 06 7E
0770   03 0C

Low-level drivers:  (see also 'Test & Demo' listing)
07D0   C6 3C F7 80 21 39 C6 04 F7 80 23 7F 80 22 7F 80
07E0   23 B7 80 22 39 CE 80 20 C6 38 E7 01 C6 FF E7 00
07F0   C6 3C E7 01 A7 00 C6 34 E7 01 39 -- -- -- -- --

Dream Invaders program changes:
0200   BD 07 20
0300   7E 07 60
0387   BD 07 26
0398   BD 07 00
03FB   BD 07 3D
045E   01 01 01 01 01
0468   BD 07 46
050D   01 01 01 01 BD 07 00
```

N.B: Sound generator I/O port addresses are marked thus: $\boxed{80\ XX}$.

MJB

PRINTED CIRCUIT BOARD for the DREAM SOUND GENERATOR

Thanks to Ian Combridge (of the Division of Electrical Engineering, Deakin University) we have a PCB layout for the sound generator. The board is designed to suit either those who need the extra PIA, or those who have a PIA on their existing expander board, in which case the PIA end of the sound board may be sawn off. A small proto area is included for experimenters to add extra circuitry (e.g; an LM380).
Observe that the I/O sockets are not configured the same as the J.R. expander board. The buss connectors are identical to the DREAM board, so that it is possible to "daisy chain" the mother board, a RAM expander board, and the sound board. Make sure you get the Vcc and GND connections right! Sockets are recommended for all IC's.

Note: There is as yet no commercial outlet for this PCB. Please do not write to the User Group or 'Dreamware' requesting PC boards. The layout is given here for those who are able to fabricate their own boards or who have access to facilities via friends in industry, etc. Certain PCB suppliers (e.g. 'RCS Radio') will produce "one-off" boards from your layout (at a price!). There is nothing printed on the back side of this page in the vicinity of the PCB pattern, so that it should be possible to produce a negative using the contact-print method.