FROM MICRO POWER

The Micro Power ADD ON Board gives Spectrum users the following advantages:

i) 3 Channnel sound effects
ii) Amplification of both standard sound and ADD ON sound
iii) Allows a connection of one or two potentiometer joy-sticks.
iv) Allows connection of one switch type joystick

If you are anxious to sample the facilities of the ADD ON turn to the back page .

Introduction to the sound generator chip

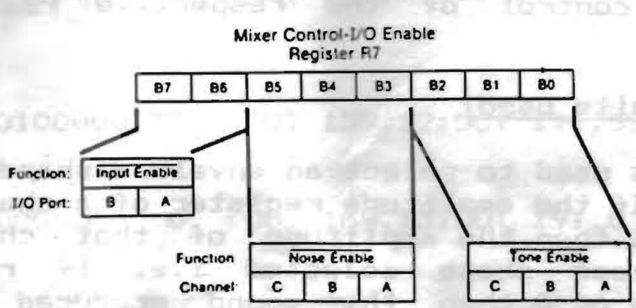
The I.C. used on the board for sound generation is the popular AY-3-8910 chip by General Instruments. This chip is capable of producing sound and or noise from 3 independent channels. The user has full control of pitch, amplitude and envelope shape. There are also two 8 bit Input/Output ports, in this implementation, they are used to control the joysticks.

Sound chip registers

The sound generator chip contains 14 registers for the production of programmed sound. There follows a brief description of the usage of the registers, for more detail see Appendix A.

Register 7 (8 bits used)

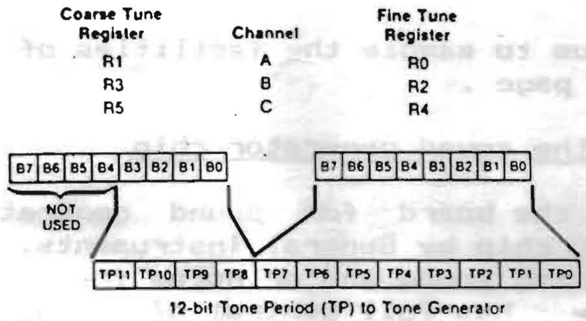
Register 7 controls the mixing of tone and noise on the three independent sound channels. To understand how this register works, it is necessary to appreciate that the registers are binary and in register 7 six of the eight bits are used to control the mixing. Setting a bit to zero enables tone or noise on the appropriate channel.




Setting register 7 as in the diagram will enable tone on channel A and noise on channel B. To set that value from BASIC, send a decimal value equivalent to 11101110 i.e. 238 to register 7. If you don't understand how these figures are arrived at then it might be a good idea if you read up on decimal to binary conversion.

Registers 0-5

This group of registers are used in pairs to control the frequency of the three tone generators. Register 0 & 1 for channel A, 2 & 3 for channel B and 4 & 5 for channel C. The even number register is used for fine tuning of the frequency and the odd register for coarse tuning. The sound generator chip uses each pair of registers to form a 12 bit number thus:




It can be seen from the diagram that the top four bits of the odd registers are not used and so those registers will only use values between 0 and 15 decimal. The 12 bit number is then used by the chip to divide down the master frequency, and so lower numbers produce higher notes.

Register 6 (5 bits used)

Register six takes a value between 0 and 31 decimal and is used to change the pitch of the noise generator.

Registers 8,9 & 10 (5 bits used)

These three registers are used to control the amplitude (volume) of the three channels. A value between 0 and 15 decimal will give volumes of between off and maximum. A value of 16 (or any other with bit four set to one) will enable envelope control of the respective register (see register 13)

Register 13 (4 bits used)

This register is used to select an envelope shape (see fig. 3 Appendix A). If the amplitude register of a channel is set to a value of 16 then the amplitude of that channel will follow the envelope shape selected i.e. if register 13 contains a value of 0 then the sound produced will rise sharply and die away gradually. The time period over which the envelope operates is controlled by the values in registers 11 & 12. If the envelope selected is a non-repeating shape , then re-inputting the same value will re-trigger the sound.

Register 11 & 12

Registers 11 & 12 are used together to make a single 16 bit value which the chip uses to time the envelope period. A large value produces a long envelope, a small value produces a short envelope. Register 11 may be regarded as fine tuning and register 12 as coarse.

Programming the chip

The registers on the sound chip may be accessed in BASIC using IN and OUT key words. Use of the sound chip from BASIC is simple. Before sending a value to a particular register the number of the register must be set up, by sending it to port 159. e.g. to prepare register 7 for data, use the following BASIC statement:

```
OUT 159,7
```

Once the register is prepared a value sent to port 191 will over-write the current contents of the register with new data e.g. OUT 191,56. The whole operation may be contained in one line of BASIC.

```
e.g. 10 OUT 159,0: OUT 191,50: OUT 191,100
```

will set up register 0, put the value 50 in register 0 and then replace it with the value 100. To write a value to a different register it is necessary to send the register number to 159 again.

Here is an example of the programming of the sound chip.

| REGISTER | VALUE Binary | Dec | BASIC LINE | REMARKS |
|---|---|---|---|---|
| 7 | 00110111 | 55 | OUT 159,7:OUT 191,55 | Sets channel A to noise |
| 8 | 00010000 | 16 | OUT 159,8:OUT 191,16 | Puts channel under envelope control |
| 12 | 00100000 | 32 | OUT 159,12:OUT 191,32 | Sets envelope time period |
| 13 | 00000000 | 0 | OUT 159,13:OUT 191,0 | Set envelope shape 0 and triggers the sound effect |

The sample cassette enclosed with the ADD-ON contains 3 programs.

No.1 SOUND EDIT, allows changes to register values to be made easily using on screen editing. Try running the program and entering the following values into the registers.

Gun-shot

Register 7=7
Register 8=16
Register 12=20
Register 13=0
Every time you enter 0 onto register 13 you will hear a further gun-shot

The program is entirely in BASIC and you may find it instructive to examine the routines

No.2 JSTICK DEM, sample game showing use of joystick and sounds. Joystick No. 1 controls the position of your sights on the screen (a single dot) whilst objects appear at random. Position your sights and use the fire button to shoot the objects. Once again the program is written in BASIC.

No.3 JOYCODE, machine code program for reading the joy-sticks and buttons. This block of machine code consists of two small machine code programs, the first at 23300 is used to read the joy-sticks. To read a joy-stick from BASIC use the following BASIC line

```
PAUSE 1:POKE 23301,2:LET a=USR 23300
```

On return from the routine a value in the range 0-255 will be placed in the variable 'a'. The number 2 in POKE 23301,2 tells the routine which joy-stick to read:
Use a value of 1 to read stick 1 horizontal, use 2 to read stick 1 vertical, 3 to read stick 2 horizontal and 4 to read stick 2 vertical. Any other value may give very strange results.

The other routine, starting at 23353 is called to read the fire buttons on both joy-sticks, it doesn't matter if you haven't got both fitted. Use the following BASIC:

```
LET a=USR 23353:LET a=PEEK 23296
```

The variable 'a' will then contain a value of either 0 or 1. If the value is 0 the button on stick one was held down at the time the routine was called. PEEKing 23297 would give similar results for the button on stick two.

JSTICK DEM loads and uses JOYCODE. Programs 1 and 2 contain instructions for use, simply load as normal.

To load JOYCODE alone type LOAD ""CODE and start the tape.If you need the routine at an alternative address type LOAD ""CODE followed by the address in decimal at which you need the routine.To save the routine from its normal address type SAVE "JOYCODE"CODE 23300,100

Resetting the sound generator

The reset pin on the AY 5 8910 has been wired to the least significant bit of I/O port B and so is capable of resetting itself

OUT 159,15:OUT 191,0:OUT 159,7:OUT 191,255

The sound registers will be set to 0 and the I/O ports set to input. Please bear this feature in mind when programming sounds, inadvertantly resetting the chip may be a difficult bug to track down.

## Connecting the board to your Spectrum

ENSURE THAT THE POWER IS REMOVED BEFORE CONNECTING OR DISCONNECTING

Holding the Spectrum keyboard in one hand and the ADD-ON board in the the other, offer up the edge connector on the board to the Spectrum expansion socket to be found at the rear right of the case. Take extra care to ensure that the key in the edge connector mates correctly with the keyway on the Spectrum.

The board contains a 2w amplifier and loud-speaker which may be used to amplify either the output from AY 5 8910 or the standard BEEP built into the Spectrum and is connected in one of two modes:

1) To use the amplifier on BEEP plug the cord from ADD ON into the MIC socket at the rear of the Spectrum. Adjust the sound level using the volume control. Any BEEPs or sound produced by the Spectrum will now be amplified.

   N.B. when using in this mode ensure the tape recorder lead does not come into contact with the ADD ON.

2) To amplify the output of the sound generator, plug the cord into the earphone socket on the ADD ON. (Adjust sound level as in 1) It is possible to take the ADD ON sound o/p from the socket on board to an external amplifier and speaker - this should only be attempted if you are confident of your skills and knowledge in electronics. (Direct connection of the ADD-ON to a HiFi system is not recommended )

## Connecting potentiometer joy-sticks to the ADD-ON board

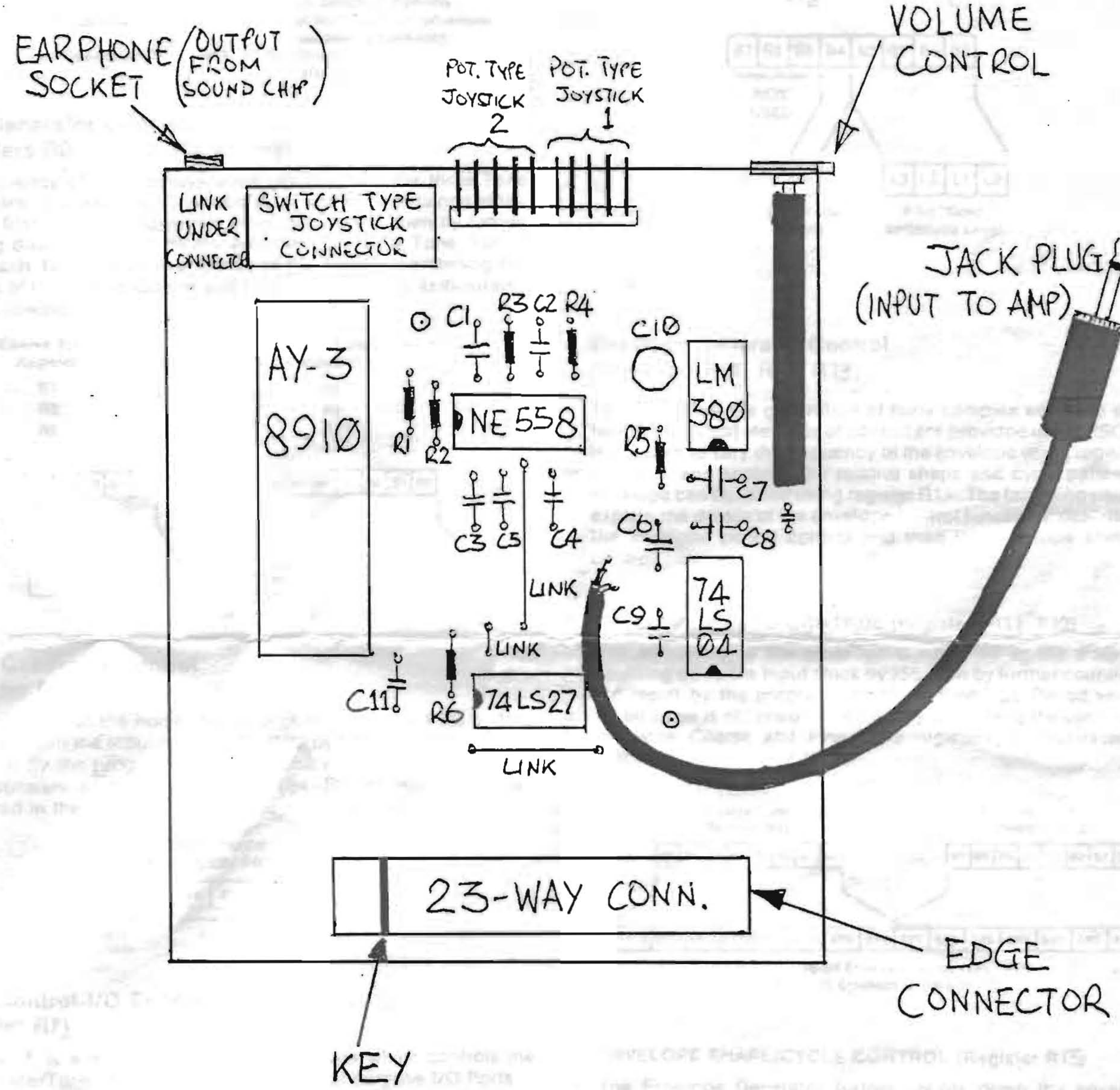
Provision is made for the connection of two independent joy-sticks to the board. If you only have one stick, connect it to port 1, the right hand connector looking at the component side of the board. If you accidentally connect the socket to the plug the wrong way round, no damage will be caused but of course the joystick will not work (see notes with the joy-stick kits).

## Connecting a switch type joystick

Simply plug the lead from the joystick into the 9 way socket at the top left hand corner of the ADD-ON board.

# Diagram showing connections to Spectrum ADD-ON




| |
|---|
| Disconnect power to Spectrum before attaching or removing the ADD-ON board |

## OPERATION

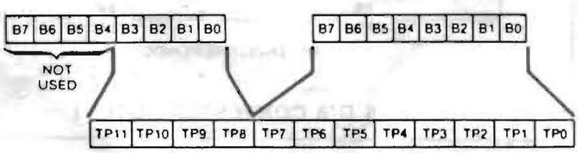
Since all functions of the PSG are controlled by the host processor via a series of register loads, a detailed description of the PSG operation can best be accomplished by relating each PSG function to the control of its corresponding register. The function of creating or programming a specific sound or sound effect logically follows the control sequence listed:

| Operation | Registers | Function |
|---|---|---|
| Tone Generator Control | R0--R5 | Program tone periods |
| Noise Generator Control | R6 | Program noise period |
| Mixer Control | R7 | Enable tone and/or noise on selected channels. |
| Amplitude Control | R8--R10 | Select "fixed" or "envelope-variable" amplitudes. |
| Envelope Generator Control | R11--R13 | Program envelope period and select envelope pattern |

### Tone Generator Control
(Registers R0, R1, R2, R3, R4, R5)

The frequency of each square wave generated by the three Tone Generators (one each for Channels A, B, and C) is obtained in the PSG by first counting down the input clock by 16, then by further counting down the result by the programmed 12-bit Tone Period value. Each 12-bit value is obtained in the PSG by combining the contents of the relative Coarse and Fine Tune registers, as illustrated in the following:

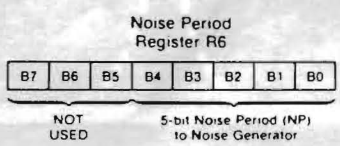
| Coarse Tune Register | Channel | Fine Tune Register |
|---|---|---|
| R1 | A | R0 |
| R3 | B | R2 |
| R5 | C | R4 |




12-bit Tone Period (TP) to Tone Generator

### Noise Generator Control
(Register R6)

The frequency of the noise source is obtained in the PSG by first counting down the input clock by 16, then by further counting down the result by the programmed 5-bit Noise Period value. This 5-bit value consists of the lower 5 bits (B4--B0) of register R6, as illustrated in the following:



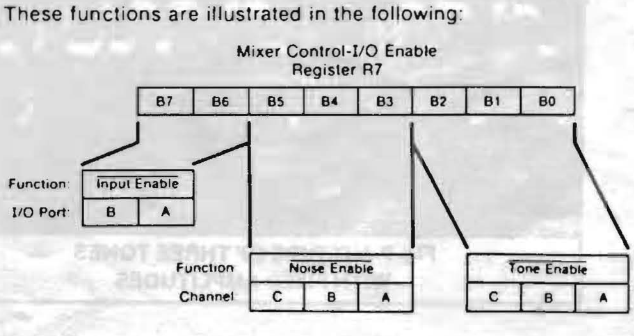

### Mixer Control-I/O Enable
(Register R7)

Register 7 is a multi-function Enable register which controls the three Noise/Tone Mixers and the two general purpose I/O Ports.

The Mixers, as previously described, combine the noise and tone frequencies for each of the three channels. The determination of combining neither/either/both noise and tone frequencies on each channel is made by the state of bits B5--B0 of R7.

The direction (input or output) of the two general purpose I/O Ports (IOA and IOB) is determined by the state of bits B7 and B6 of R7.

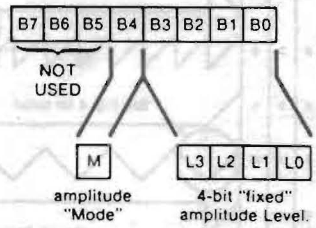
These functions are illustrated in the following:




### Amplitude Control
(Registers R8, R9, R10)

The amplitudes of the signals generated by each of the three D/A Converters (one each for Channels A, B, and C) is determined by the contents of the lower 5 bits (B4--B0) of registers R8, R9, and R10 as illustrated in the following:

| Amplitude Control Register # | Channel |
|---|---|
| R8 | A |
| R9 | B |
| R10 | C |

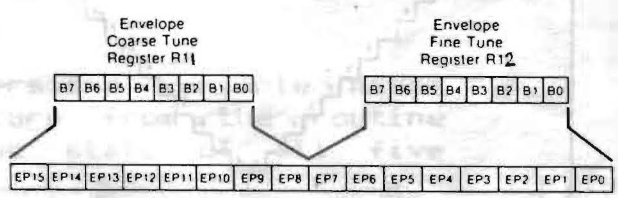



### Envelope Generator Control
(Registers R11, R12, R13)

To accomplish the generation of fairly complex envelope patterns, two independent methods of control are provided in the PSG: first, it is possible to vary the frequency of the envelope using registers R11 and R12; and second, the relative shape and cycle pattern of the envelope can be varied using register R13. The following paragraphs explain the details of the envelope control functions, describing first the envelope period control and then the envelope shape/cycle control.

#### ENVELOPE PERIOD CONTROL (Registers R11, R12)

The frequency of the envelope is obtained in the PSG by first counting down the input clock by 256, then by further counting down the result by the programmed 16-bit Envelope Period value. This 16-bit value is obtained in the PSG by combining the contents of the Envelope Coarse and Fine Tune registers, as illustrated in the following:

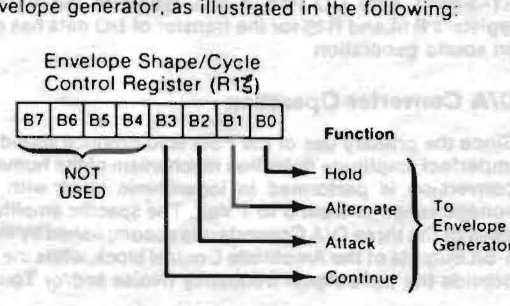



16-bit Envelope Period (EP) to Envelope Generator

#### ENVELOPE SHAPE/CYCLE CONTROL (Register R13)

The Envelope Generator further counts down the envelope frequency by 16, producing a 16-state per cycle envelope pattern as defined by its 4-bit counter output, E3 E2 E1 E0. The particular shape and cycle pattern of any desired envelope is accomplished by controlling the count pattern (count up/count down) of the 4-bit counter and by defining a single-cycle or repeat-cycle pattern.

This envelope shape/cycle control is contained in the lower 4 bits (B3--B0) of register R13. Each of these 4 bits controls a function in the envelope generator, as illustrated in the following:



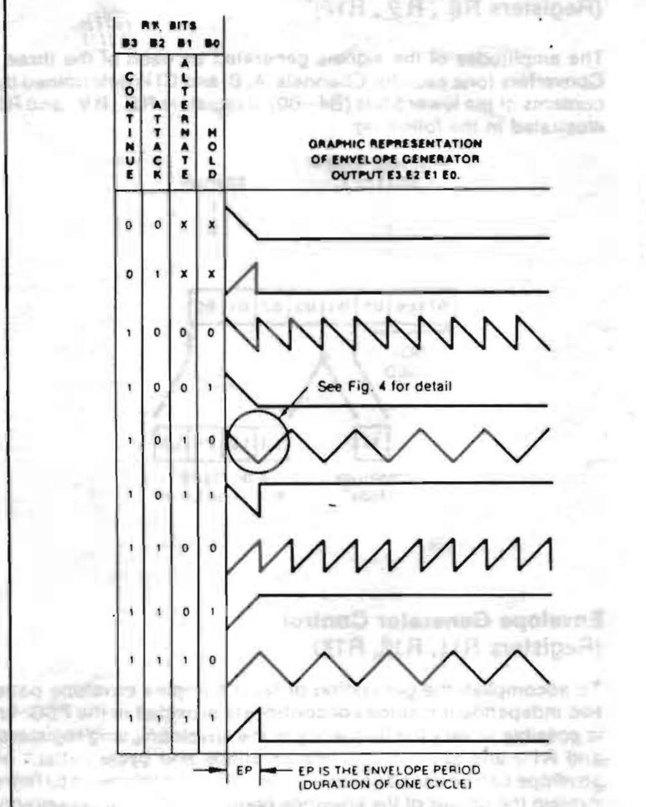



**Fig. 3 ENVELOPE SHAPE/CYCLE OPERATION**

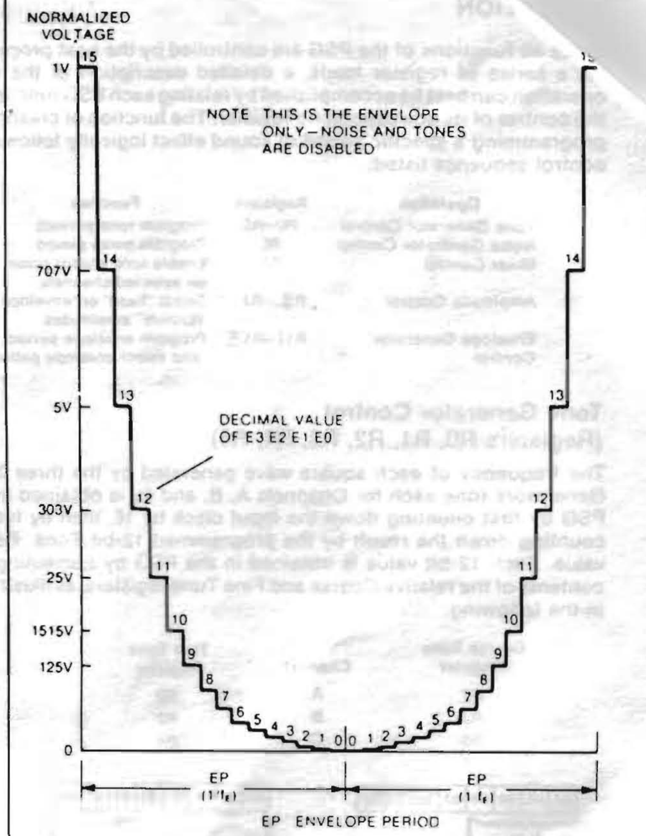



**Fig. 5 D/A CONVERTER OUTPUT**

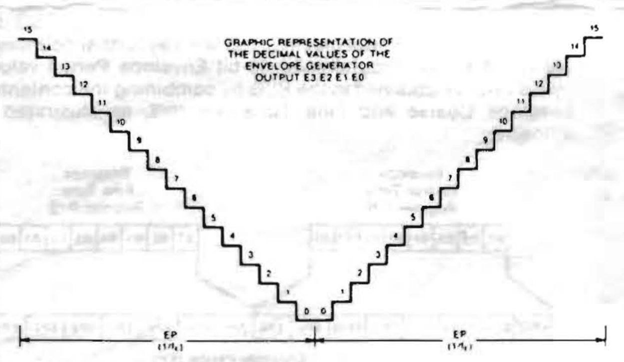



**Fig. 4 DETAIL OF TWO CYCLES OF Fig. 3 (ref. waveform "1010" in Fig. 3)**

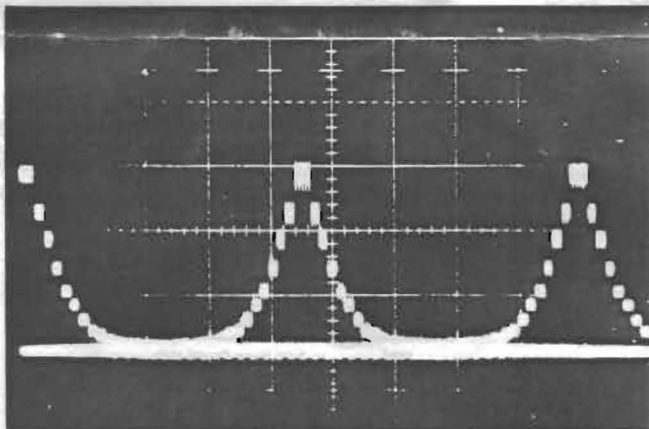


**Fig. 6 SINGLE TONE WITH ENVELOPE SHAPE/CYCLE PATTERN 1010**

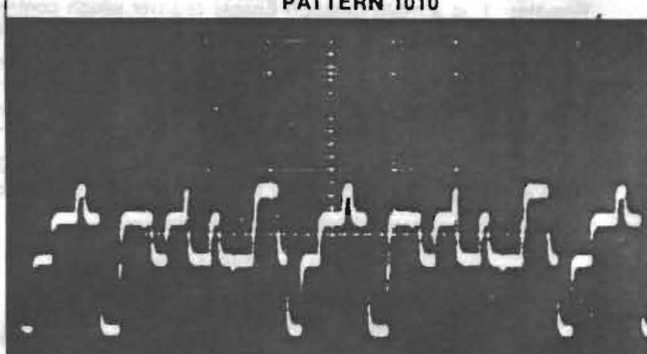


**Fig. 7 MIXTURE OF THREE TONES WITH FIXED AMPLITUDES**

## I/O Port Data Store
### (Registers R14, R15)

Registers R14 and R15 function as intermediate data storage registers between the PSG/CPU data bus (DA0--DA7) and the two I/O ports (IOA7--IOA0 and IOB7--IOB0). Both ports are available in the AY-3-8910; only I/O Port A is available in the AY-3-8912. Using registers R14 and R15 for the transfer of I/O data has no effect at all on sound generation.

## D/A Converter Operation

Since the primary use of the PSG is to produce sound for the highly imperfect amplitude detection mechanism of the human ear, the D/A conversion is performed in logarithmic steps with a normalized voltage range of from 0 to 1 Volt. The specific amplitude control of each of the three D/A Converters is accomplished by the three sets of 4-bit outputs of the Amplitude Control block, while the Mixer outputs provide the base signal frequency (Noise and/or Tone).

## Reading the switch type joystick

The switch type joystick is connected to 5 bits of the second I/O port on the sound chip and is mapped thus:

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | bØ |
|---|---|---|---|---|---|---|---|
| / | / | F | R | L | U | D | / |

Where F=fire R=right L=left U=up and D=down

The following BASIC subroutine may be used to read the joystick:

```
10 DIM c(5)
.
.
.
.
5000 OUT 159,7:OUT 191,255:OUT 159,15:OUT
191,255::LET a=IN 223
5005 LET p=32
5010 FOR t=5 TO 1 STEP -1:LET c(t)=(a<p)
5020 IF a>p THEN LET a=a-p
5030 LET p=p/2
5040 NEXT t:RETURN
```

You may of course use line numbers and variable names to suit your own program. On return from the routine the array c will contain the state of all five switches.

c(1)=Down c(2)=Up c(3)=Right c(4)=Left and c(5)=Fire

e.g. If c(1)=1 then the joystick is thrown down
If c(2)=1 and c(3)=1 then the stick is thrown to the top left.