MrX Sound Board

for the ZX81 from Sinclair

“Manual for users and programmers”

www.eightbits.de

CONTENT

Disclaimer

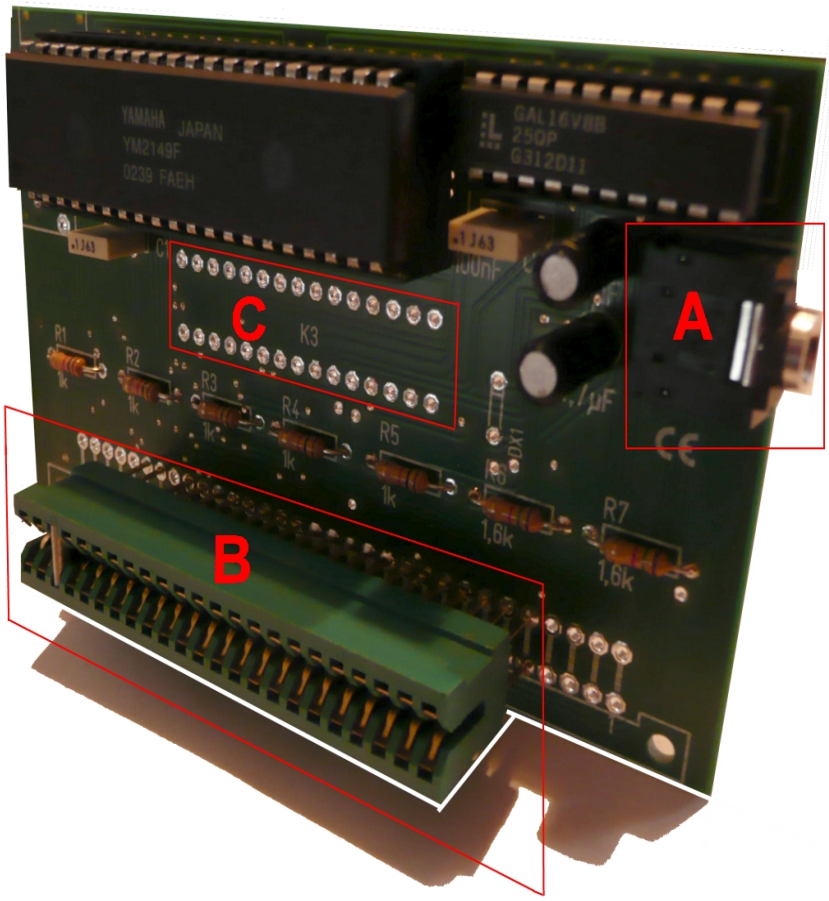
This book is presented solely for educational and entertainment purposes. The author and publisher are not offering it as legal, accounting, or other professional services advice. While best efforts have been used in preparing this book, the author and publisher make no representations or warranties of any kind and assume no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaim any implied warranties of merchantability or fitness of use for a particular purpose. Neither the author nor the publisher shall be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information or programs contained herein. No warranty may be created or extended by sales representatives or written sales materials. Every company is different and the advice and strategies contained herein may not be suitable for your situation. You should seek the services of a competent professional before beginning any actions described in this manual.

1. System Requirements

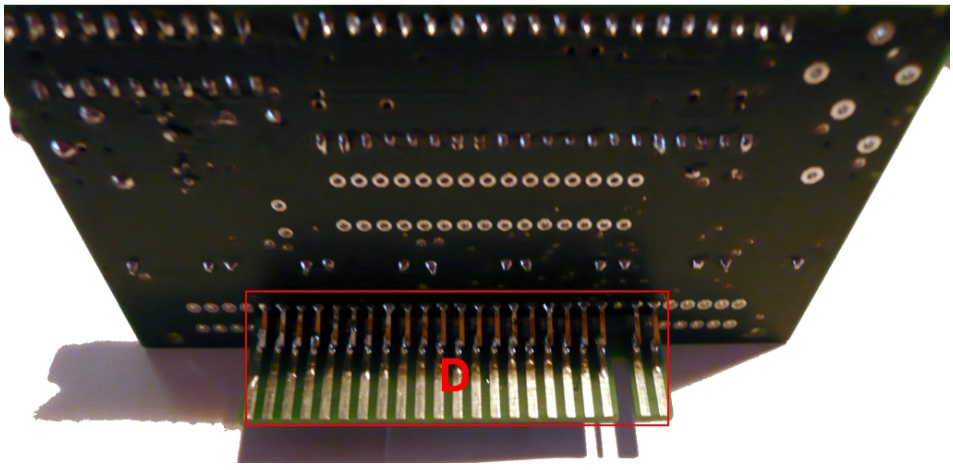
Computer: ZX81
Manufacturer: Sinclair, UK
Amplifier: active amplifier (PC-amplifier) with 3,5mm jack

Recommended: 16k Ram

2. MrX Sound Card



A) 3,5mm jack. Connect the amplifier here.
B) Port connector.
C) 30 pin expansion bus K3 (see next chapter)

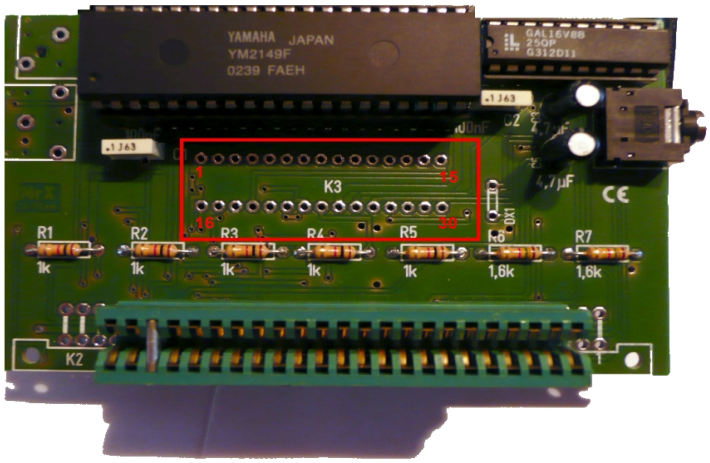


D) Through port connector.

3. MrX Expansion Bus K3

The MrX Interface is supplied with a built in expansion bus (K3) which allows direct access to the signals provided by the YM2149 sound chip, should you wish to develop an add-on daughterboard.

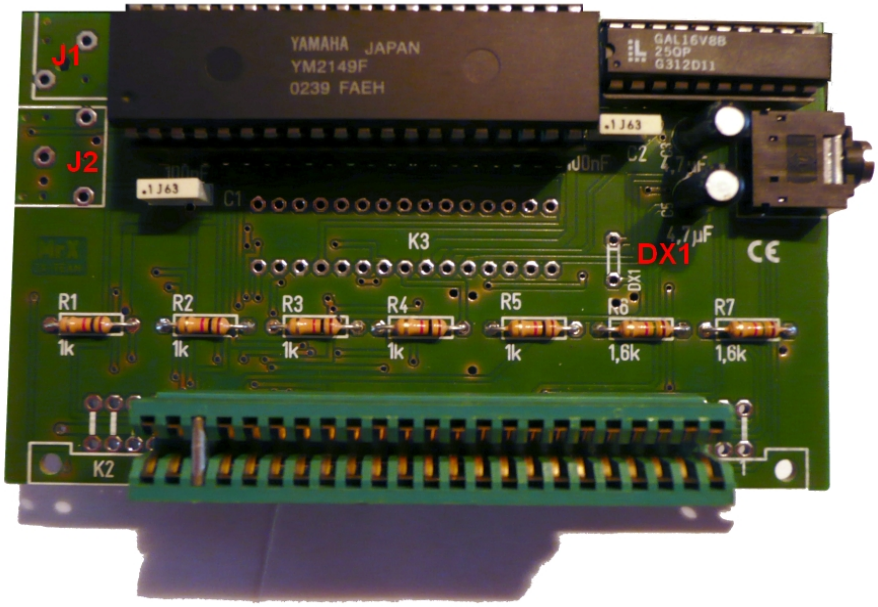
Pinout K3:



1)	IOB7	Port B from YM2149
2)	IOB6	Port B from YM2149
3)	IOB5	Port B from YM2149
4)	IOB4	Port B from YM2149
5)	IOB3	Port B from YM2149
6)	IOB2	Port B from YM2149
7)	IOB1	Port B from YM2149
8)	IOB0	Port B from YM2149
9)	IOA7	Port A from YM2149
10)	IOA6	Port A from YM2149
11)	IOA5	Port A from YM2149
12)	IOA4	Port A from YM2149
13)	IOA3	Port A from YM2149
14)	IOA2	Port A from YM2149
15)	IOA1	Port A from YM2149
16)	CHL	Left channel of 3,5mm jack, behind capacitor
17)	CHR	Right channel of 3,5mm jack, behind capacitor
18)	GND	Ground
19)	/CLK	ZX81 clock signal 3,25 MHz
20)	GND	Ground
21)	ANALOG_CH_C	Analog Channel C directly connected to YM2149
22)	ANALOG_CH_B	Analog Channel B directly connected to YM2149
23)	ANALOG_CH_A	Analog Channel A directly connected to YM2149
24)	VCC	Supply current +5V
25)	GAL1	Pin 15 of GAL 16V8
26)	GAL2	Pin 14 of GAL 16V8
27)	GAL3	Pin 13 of GAL 16V8
28)	GAL4	Pin 12 of GAL 16V8
29)	clock/2	1,625 MHz
30)	IOA0	Port A from YM2149

4. Optional 3,5mm jacks and ZX96 bus diodes

If needed additional 3,5mm jacks can be soldered on the MrX sound card on J1 and J2.



For the ZX96 bus (http://www.fischerkai.de/zxteam/treib_e.htm) a diode DX1 for the /BUSCS signal has to be soldered and the port connector has to be exchanged by a VG64 connector. The VG64 connector uses all pins of K2.

Note: The ZX96 bus is only used by some freaks, if your ZX81 is equipped with the same the original MrX won't fit mechanically.

5. Connection

- Switch off the computer before connecting or removing any interfaces. Disconnect the power lead to be certain! Otherwise, severe damage may occur to the computer and the sound card.

- The ZX81 computer is connected to the sound card via the ZX expansion port on the back of the computer.

- Make sure that the pins of the ZX81 PCB are exactly aligned with the connector of the MrX sound card.

- Don't use brute force to connect the MrX with the ZX81.

- The PC speaker (with integrated amplifier) has to be connected with the 3,5mm stereo jack.

- If needed, connect further equipment. NOTE: Make sure that the pins of the MrX PCB are exactly aligned with the connector of the equipment. Otherwise severe damage may occur to the computer, the equipment and the sound card.

6. Software

This manual and the software for the MrX sound card is provided at http://www.eightbits.de/ in the download section.

manual.pdf	→ This manual
AY-Demo	→ Sound-Demo
Basic-Demo	→ Original Basic Demos from the ZON-X Manual
Demon-Demo	→ Dancing Demon Demo
Games	→ 2 Games from Brasilian TK85
PT3-Player	→ Player, plays PT3-files
Pink-Panther	→ Music demo
ZON-X-Manual	→ HTML document original zonx manual

Concerning the PT3-Player, PT3-files have to be converted to wav-files according the instructions in the ZX81 forum „http://www.rwapservices.co.uk/ZX80_ZX81/forums/aye-aye-t528s170.html#p4919 “.

7. Programming

The MrX is compatible with the original ZON-X sound card.

Addresses:

The MrX Interface responds to data placed in the following addresses:

Latch	Data	Comment
0xDF	0x0F	modified ZON-X
0xCF	0x1F	original ZON-X
0xCF	0x0F	from ZON-X user manual
0xDF	0x1F	additional combination

See chapter “8. The Yamaha YM2149” for further explanation about “Latch” (register address latch) and “Data” (write mode).

Examples in assembler

Simple Sound:

```
;       LATCH:    equ $DF
        LATCH:    equ $CF
        DTAX:     equ $0F
;       DTAX:     equ $1F

xxx:

    LD   A,7
    out  (LATCH),A
    LD   A,$C0
    out  (DTAX),A

    LD   A,$08
    out  (LATCH),A
    LD   A,15
    out  (DTAX),A


looop:
    LD   A,0
    out  (LATCH),A
    LD   A,70
    out  (DTAX),A
    ret
```

Output on port A:

```
   DTAX:  equ $0F
;  DTAX:  equ $1F
   LATCH:  equ $CF
;  LATCH:  equ $DF

xxx:

  LD A,7
  out (LATCH),A
  LD A,$C0      ;---> set port A and B as output
  out (DTAX),A

looop:

 LD A,14
 out (LATCH),A
 LD A,$00
 out (DTAX),A  ; set port A to 0

 LD HL,(DFILE)
 INC HL
 LD (HL),_O
 INC HL
 LD (HL),_F
 INC HL
 LD (HL),_F

 LD  BC,100
 CALL $0F35

 LD A,14
 out (LATCH),A
 LD A,$FF
 out (DTAX),A   ; set all bits of port A to 1 (high)
 LD HL,(DFILE)
 INC HL
 LD (HL),_O
 INC HL
 LD (HL),_N
 INC HL
 LD (HL),__

 LD  BC,100
 CALL $0F35

 jp  looop
```

Reading from port A:

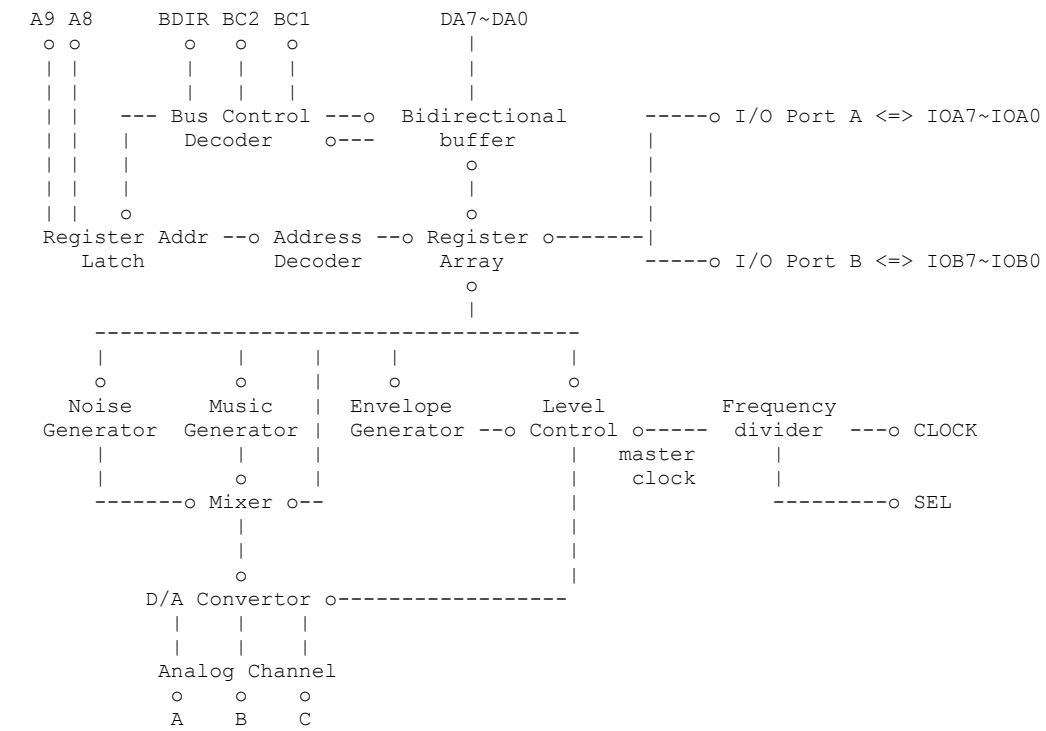
```
   DTAX:  equ $0F
;  DTAX:  equ $1F
   LATCH:  equ $CF
;  LATCH:  equ $DF

xxx:

  LD A,7
  out (LATCH),A
  LD A,$00       ;--> set port A & B to Input
  out (DTAX),A

looop:

 LD A,14
 out (LATCH),A

 in a,(LATCH)  ;Port A (register No. 14) is read
 and a,$3F

 LD HL,(DFILE)
 INC HL
 LD (HL),A    ;Print the port content to the screen


 jp  looop
```

8. The Yamaha YM2149

This is a copy from “http://www.atari-forum.com/wiki/ “.

```
Software-Controlled Sound Generator (SSG)
-----------------------------------------


Overview

The SSG (Software-Controlled Sound Generator) is an NMOS-LSI device designed
to be capable of music generation. It only requires the microprocessor or
microcomputer (CPU) to initialize its register array, thus reducing the load
on the CPU. Music generation is carried out by the three sequence square wave
generator, noise generator, and envelope generator according to the set
parameters. This allows for the generation of music, special effects,
warnings, and various other types of sounds.


Features

5V single power supply
Easy connection to 8 bit or 16 bit CPU
Simple connection to external system through 2 sequence 8 bit I/O port
Wide voicing range of 8 octaves
Smooth attenuation by 5 bit envelope generator
Built-in 5 bit D/A convertor
Input of double frequency clock can be handled by built-in clock frequency
 divider
TTL compatible level
Low power consumption (typical 125mW)
40 pin plastic DIL package
Pin compatible with AY-3-8910 manufactured by GI


Pin Layout

        Vss(GND) 1          40 Vcc(+5V)
             N.C 2          39 Test1
Analog Channel B 3          38 Analog Channel C
Analog Channel A 4          37 DA0
             N.C 5          36 DA1
            IOB7 6          35 DA2
            IOB6 7          34 DA3
            IOB5 8          33 DA4
            IOB4 9          32 DA5
           IOB3 10          31 DA6
           IOB2 11          30 DA7
           IOB1 12          29 BC1
           IOB0 13          28 BC2
           IOA7 14          27 BDIR
           IOA6 15          26 SEL
           IOA5 16          25 A8
           IOA4 17          24 A9
           IOA3 18          23 RESET
           IOA2 19          22 CLOCK
           IOA1 20          21 IOA0
```

Block diagram



Description of pins

1. DA7 ~ DA0
   This is an 8 bit bidirectional data bus which is used for moving data and addresses between the SSG and CPU. In the read and write modes, DA7 ~ DA0 corresponds to B7 ~ B0 of the register array. In the address mode, DA3 ~ DA0 is used for the register address, and DA7 ~ DA4 is used together with A9 and A8 for the upper address.
2. A8 and A9
   These are the upper address input pins. A8 has pullup resistance while A9 has pulldown resistance. When the voltage level at A8 while the level at A9 and DA7 ~ DA4 is low, the address mode is selected allowing for the fetching of a register address. Connect A8 and A9 to +5V and ground respectively when not in use.
3. RESET
   Reset is effective when the voltage level is low, and the contents of all registers in the array are reset to '0'. This pin has pullup resistance.
4. CLOCK
   Supplies the master clock to the sound generator and envelope generator. This is equipped with a 1/2 frequency divider which allows for the use of a frequency which is 1/2 of the input clock, as the master clock.
5. SEL
   When SEL is driven to the high level, the input clock is taken as the master clock. When the voltage level of SEL is low, the input clock is divided by 2 to obtain the master clock. This pin has pullup resistance, allowing for full pin compatibility with the AY-3-8910 manufactured by AI, when this pin is not connected to anything.

6. BDIR,BC1 and BC2
  Controls the external bus (DA7 ~ DA0) and internal bus of the SSG. The
  following four modes can be set by the bus control decoder. The bus control
  is redundant, control is possible even when BC5 is connected to +5V.

BDIR	BC2	BC1	Mode
0	0	0	Inactive
0	0	1	Address
0	1	0	Inactive
0	1	1	Read
1	0	0	Address
1	0	1	Inactive
1	1	0	Write
1	1	1	Address

Inactive mode: DA7 ~ DA0 has high impedance.
Address mode: DA7 ~ DA0 set to input mode, and address is fetched from
                register array.
Write mode: DA7 ~ DA0 set to input mode, and data is written to register
                currently being addressed.
Read mode: DA7 ~ DA0 set to output mode, and contents of register
                currently being addressed are output.

7. ANALOG CHANNEL A,B,C
  Each of the three channels is equipped with a D/A convertor which converts
  the calculated digital values to analog signals for output.
8. IOA7 ~ IOA0, IOB7 ~ IOB0
  These are two 8 bit I/O ports. These ports allow the SSG to be placed
  between an external system and the CPU for the transfer of data. These pins
  have pullup resistance.
9. TEST1
  Output pin for testing the device. Do not connect to anything.
10. Vcc
  +5V power pin.
11. Vss
  Ground pin.

Description of funtions

All functions of the SSG are controlled by the 16 internal registers. The CPU need only write data to the internal registers of the SSG. The SSG itself generates the sound. Sound is generated by the following blocks:

Music generator:	Square waves having a different frequency are generated for each channel (A,B and C)
Noise generator:	Pseudo-random waveforms are generated (variable frequency)
Mixer:	Music and noise output are mixed for the three channels (A,B and C)
Level control:	Constant level or variable level is given for each of the three channels (A,B and C). Constant levels are controlled by the CPU, and variable levels by the envelope generator.
Envelope generator:	Generates various types of attenuation (single burst attenuated and repeated attenuation)
D/A convertor:	Sound is output on each of the three channels (A,B and C) at the level determined by the level control.

The CPU can read the contents of the internal registers with no effect on sound.

Register Array

A9	A8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
_____						/_____			
Upper addresses (chip select)						Lower addresses (register select)			

Of the ten bit address, the lower addresses DA3 ~ DA0 are used to select the 16 internal registers(register array). The upper addresses are used for chip selection. A9 and A8 is programmed to 01 while DA7 through DA4 are set to 0000. When the upper addresses match this program in the address mode, a register address (lower four bits DA3 through DA0) is fetched from the register address latch. When the value set is in the upper addresses is different from the program value, the bidirectional bus formed from DA7 through DA0 is driven to high impedance. A register address which has been fetched is retained until the next address is fetched, and is not affected by the read, write, or inactive mode.

Register Array

	B7....B0	
R0 Frequency of Channel A	00000000	8 bit fine tone adjustment
R1	----0000	4 bit rough tone adjustment
R2 Frequency of Channel B	00000000	8 bit fine tone adjustment
R3	----0000	4 bit rough tone adjustment
R4 Frequency of Channel C	00000000	8 bit fine tone adjustment
R5	----0000	4 bit rough tone adjustment
R6 Frequency of Noise	---00000	5 bit noise frequency
R7 I/O port and mixer settings	iinnnttt bacbacba	i-I/O, n-Noise, t-Tone
R8 Level of channel A	---mllll	m-Mode, l-Level
R9 Level of channel B	---mllll	m-Mode, l-Level
RA Level of channel C	---mllll	m-Mode, l-Level
RB Frequency of envelope	00000000	8 bit fine adjustment
RC	00000000	8 bit rough adjustment
RD Shape of envelope	----cath	c-Cont, a-Att, t-Alt, h-Hold
RE Data of I/O port A	00000000	8 bit data
RF Data of I/O port B	00000000	8 bit data

(1) Setting of music frequency (controlled by registers R0 ~ R5)

The frequencies of the square wave generated by the music generators for the three channels (A,B and C) are controlled by registers R0 through R5. R0 and R1 control channel A, R2 and R3 are used for channel B, and R4 and R5 control channel C. The oscillation frequency fT is obtained in the following manner from the value of the register TP(decimal).

$$fT = \frac{fMaster}{16TP}$$

fMaster is the frequency of the master clock (this is the input click frequency when

SEL is high, and 1/2 of this frequency when low).

Rough tone adjustment register	Channel	Fine tone adjustment register
R1	A	R0
R3	B	R2
R5	C	R4

B7 B6 B5 B4 B3 B2 B1 B0 B7 B6 B5 B4 B3 B2 B1 B0

Not used | TP11 TP10 TP9 TP8 TP7 TP6 TP5 TP4 TP3 TP2 TP1 TP0

12 bit oscillation frequency setting value (TP)

(2) Setting of noise generator (controlled by register R6)

The noise frequency fN is obtained from the register value NP(decimal) in the following manner.

$$fN = \frac{fMaster}{16NP}$$ (fMaster if the frequency of the master clock)

Noise frequency register R6

B7 B6 B5 B4 B3 B2 B1 B0

Not used | NP4 NP3 NP2 NP1 NP0

5 bit noise frequency setting value (NP)

(3) Settings of mixer and I/O ports (controlled by register R7)

The mixer is used to combine music and noise components. The combination is determined by bits B5 ~ B0 of register R7. Sound is output when a '0' is written to the register. Thus, when both the noise and tone are '0', the output is combined by the mixer. When the noise is '0' and the tone is '1', only the noise signal is output. When the noise is '1' and the tone is '0', music (square wave) is output. Nothing is output when both the noise and tone are '1'. Selection of input/output for the I/O ports is determined by bits B7 and B6 of register R7. Input is selected when '0' is written to the register bits.

I/O port and mixer setting register R7

B7 B6 B5 B4 B3 B2 B1 B0

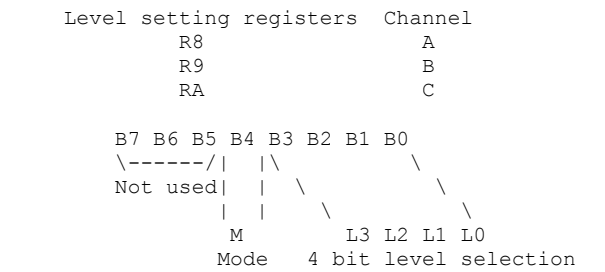
I/O Noise Tone

B A C B A C B A

(Input is selected for I/O port when '0', and noise or tone can be output when '0')

(4) Level control (controlled by R8 ~ RA)

The audio level output from the D/A convertors for the three channels (A,B and C) is adjusted by registers R8, R9 and RA.

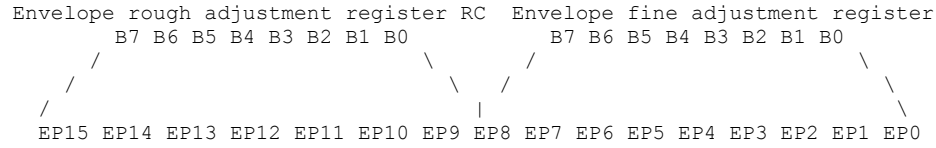


Mode M selects whether the level is fixed (when M=0) or variable (M=1). When M=0, level is determined from one of 16 by level selection signals L3,L2,L1 and L0 which compromise the lower four bits. When M=1, the level is determined by the 5 bit output of E4,E3,E2,E1 and E0 of the envelope generator of the SSG. (This level is variable as E4 ~ E0 change over time)

(5) Setting of envelope frequency (controlled by R8 and RC)

Thus, the envelope repetition frequency fE is obtained as follows from the envelope setting period value EP (decimal):

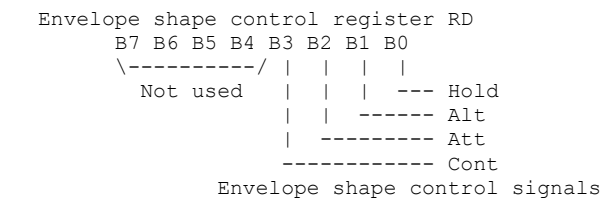
$$fE = \frac{fMaster}{256EP}$$ (fMaster if the frequency of the master clock)



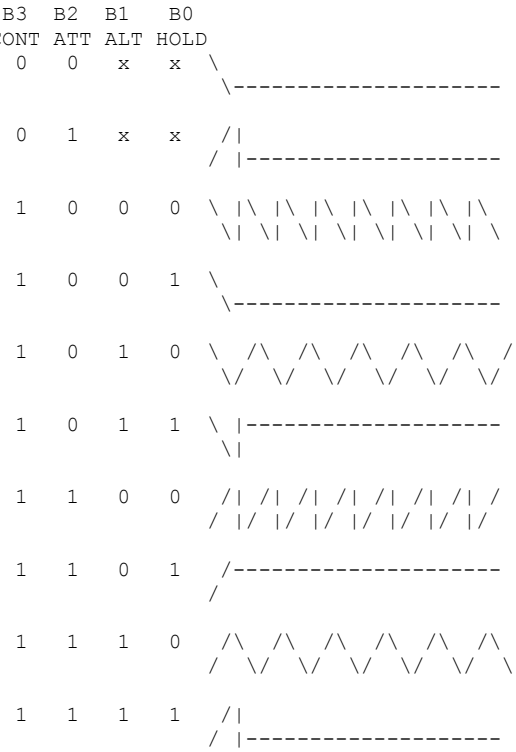
The period of the actual frequency fEA used for the envelope generated is 1/32 of the envelope repetition period (1/fE).

(6) Envelope shape control (controlled by RD)

The envelope generator counts the envelope clock fEA 32 times for each envelope pattern cycle. The envelope level is determined by the 5 bit output (E4 ~ E0) of the counter. The shape of the envelope is created by increasing, decreasing, stopping, or repeating this counter. The shape is controlled by bits B3 ~ B0 of the register RD.



The envelope can take the shapes shown below according to combinations of the CONT, ATT, ALT and HOLD signals.



NOTE - The writing to register RD will reset the envelope frequency clock