FEBRUARY 1979

AY-3-8910 AY-3-8912

Programmable Sound Generator

FEATURES

- □ Full software control of sound generation.
- □ Interfaces to most 8-bit and 16-bit microprocessors.
- □ Three independently programmed analog outputs.
- □ Two 8-bit general purpose I/O ports (AY-3-8910).
- □ One 8-bit general purpose I/O port (AY-3-8912).
- □ Single +5 Volt Supply.

DESCRIPTION

The AY-3-8910/8912 Programmable Sound Generator (PSG) is a Large Scale Integrated Circuit which can produce a wide variety of complex sounds under software control. The AY-3-8910/8912 is manufactured in GI's N-Channel Ion Implant Process. Operation requires a single 5V power supply, a TTL compatible clock, and a microprocessor controller such as the GI 16-bit CP1600/1610 or one of GI's PIC 1650 series of 8-bit microcomputers.

The PSG is easily interfaced to any bus oriented system. Its flexibility makes it useful in applications such as music synthesis, sound effects generation, audible alarms, tone signalling and FSK modems. The analog sound outputs can each provide 4 bits of logarithmic digital to analog conversion, greatly enhancing the dynamic range of the sounds produced.

In order to perform sound effects while allowing the processor to continue its other tasks, the PSG can continue to produce sound after the initial commands have been given by the control processor. The fact that realistic sound production often involves more than one effect is satisfied by the three independently controllable channels available in the PSG.

All of the circuit control signals are digital in nature and intended to be provided directly by a microprocessor/microcomputer. This means that one PSG can produce the full range of required sounds with no change in external circuitry. Since the frequency response of the PSG ranges from sub-audible at its lowest frequency to post-audible at its highest frequency, there are few sounds which are beyond reproduction with only the simplest electrical connections. Since most applications of a microprocessor/PSG system would also require interfacing between the outside world and the microprocessor, this facility has been designed into the PSG. The AY-3-8910 has two general purpose 8-bit I/O ports and is supplied in a 40 lead package; the AY-3-8912 has one port and 28 leads.

PIN FUNCTIONS

DA7--DA0 (input/output/high impedance): pins 30--37 (AY-3-8910)
Data/Address 7--0: pins 21--28 (AY-3-8912)

These 8 lines comprise the 8-bit bidirectional bus used by the microprocessor to send both data and addresses to the PSG and to receive data from the PSG. In the data mode, DA7--DA0 correspond to Register Array bits B7--B0. In the address mode, DA3--DA0 select the register # (0--17$_8$) and DA7--DA4 in conjunction with address inputs $\overline{\text{A9}}$ and A8 form the high order address (chip select).

A8 (input): pin 25 (AY-3-8910)
pin 17 (AY-3-8912)

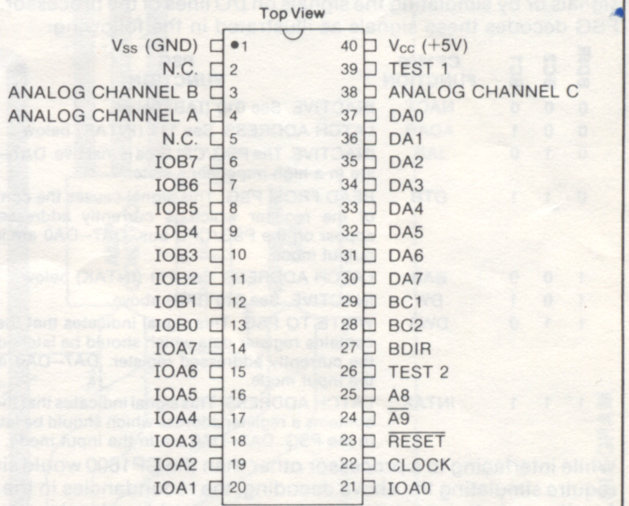
$\overline{\text{A9}}$ (input): pin 24 (AY-3-8910)
(not provided on AY-3-8912)

Address 9, Address 8

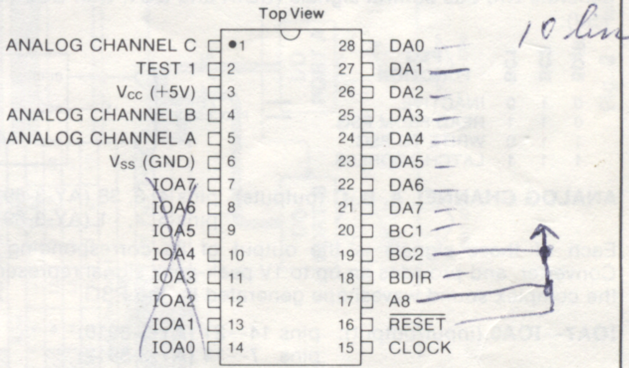
These "extra" address bits are made available to enable the positioning of the PSG (assigning a 16 word memory space) in a total 1,024 word memory area rather than in a 256 word memory area as defined by address bits DA7--DA0 alone. If the memory size does not require the use of these extra address lines they may be left unconnected as each is provided with either an on-chip pull down ($\overline{\text{A9}}$) or pull-up (A8) resistor. In "noisy" environments, however, it is recommended that $\overline{\text{A9}}$ and A8 be tied to an external ground and +5V, respectively, if they are not to be used.

PIN CONFIGURATIONS

40 LEAD DUAL IN LINE
AY-3-8910



28 LEAD DUAL IN LINE
AY-3-8912



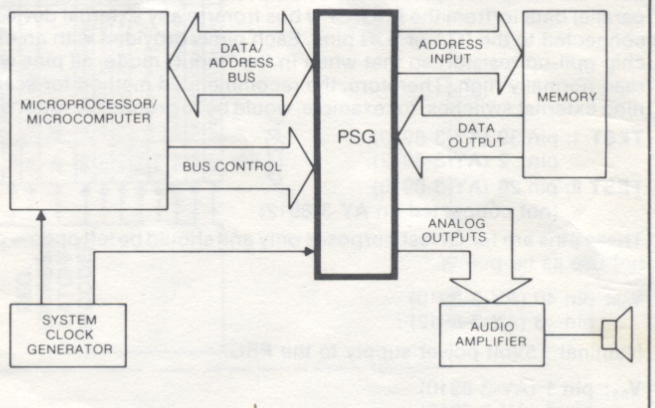


Fig. 1 SYSTEM BLOCK DIAGRAM

GENERAL INSTRUMENT CORPORATION • MICROELECTRONICS

RESET (input): pin 23 (AY-3-8910)
pin 16 (AY-3-8912)

For initialization/power-on purposes, applying a logic "0" (ground) to the $\overline{\text{Reset}}$ pin will reset all registers to "0". The $\overline{\text{Reset}}$ pin is provided with an on-chip pull-up resistor.

CLOCK (input): pin 22 (AY-3-8910)
pin 15 (AY-3-8912)

This TTL-compatible input supplies the timing reference for the Tone, Noise and Envelope Generators.

BDIR, BC2, BC1 (inputs): pins 27,28,29 (AY-3-8910)
pins 18,19,20 (AY-3-8912)

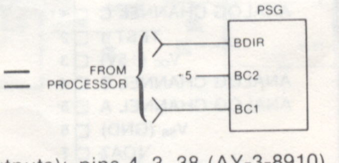
Bus DIRection, Bus Control 2,1

These bus control signals are generated directly by GI's CP1600 series of microprocessors to control all external and internal bus operations in the PSG. When using a processor other than the CP1600, these signals can be provided either by comparable bus signals or by simulating the signals on I/O lines of the processor. The PSG decodes these signals as illustrated in the following:

BDIR	BC2	BC1	CP1600 FUNCTION	PSG FUNCTION
0	0	0	NACT	INACTIVE. See 010 (IAB) below.
0	0	1	ADAR	LATCH ADDRESS. See 111 (INTAK) below.
0	1	0	IAB	INACTIVE. The PSG/CPU bus is inactive. DA7--DA0 are in a high impedance state.
0	1	1	DTB	READ FROM PSG. This signal causes the contents of the register which is currently addressed to appear on the PSG/CPU bus. DA7--DA0 are in the output mode.
1	0	0	BAR	LATCH ADDRESS. See 111 (INTAK) below.
1	0	1	DW	INACTIVE. See 010 (IAB) above.
1	1	0	DWS	WRITE TO PSG. This signal indicates that the bus contains register data which should be latched into the currently addressed register. DA7--DA0 are in the input mode.
1	1	1	INTAK	LATCH ADDRESS. This signal indicates that the bus contains a register address which should be latched in the PSG. DA7--DA0 are in the input mode.

While interfacing to a processor other than the CP1600 would simply require simulating the above decoding, the redundancies in the PSG functions vs. bus control signals can be used to advantage in that only four of the eight possible decoded bus functions are required by the PSG. This could simplify the programming of the bus control signals to the following, which would only require that the processor generate two bus control signals (BDIR and BC1, with BC2 tied to +5V):

BDIR	BC2	BC1	PSG FUNCTION
0	1	0	INACTIVE.
0	1	1	READ FROM PSG.
1	1	0	WRITE TO PSG.
1	1	1	LATCH ADDRESS.



ANALOG CHANNEL A, B, C (outputs): pins 4, 3, 38 (AY-3-8910)
pins 5, 4, 1 (AY-3-8912)

Each of these signals is the output of its corresponding D/A Converter, and provides an up to 1V peak-peak signal representing the complex sound waveshape generated by the PSG.

IOA7--IOA0 (input/output): pins 14--21 (AY-3-8910)
pins 7--14 (AY-3-8912)

IOB7--IOB0 (input/output): pins 6--13 (AY-3-8910)
(not provided on AY-3-8912)

Input/Output A7--A0, B7--B0

Each of these two parallel input/output ports provides 8 bits of parallel data to/from the PSG/CPU bus from/to any external devices connected to the IOA or IOB pins. Each pin is provided with an on-chip pull-up resistor, so that when in the "input" mode, all pins will read normally high. Therefore, the recommended method for scanning external switches, for example, would be to ground the input bit.

TEST 1: pin 39 (AY-3-8910)
pin 2 (AY-3-8912)

TEST 2: pin 26 (AY-3-8910)
(not connected on AY-3-8912)

These pins are for GI test purposes only and should be left open—do not use as tie-points.

V_{CC}: pin 40 (AY-3-8910)
pin 3 (AY-3-8912)

Nominal +5Volt power supply to the PSG.

V_{SS}: pin 1 (AY-3-8910)
pin 6 (AY-3-8912)

Ground reference for the PSG.

ARCHITECTURE

The AY-3-8910/8912 is a register oriented Programmable Sound Generator (PSG). Communication between the processor and the PSG is based on the concept of memory-mapped I/O. Control commands are issued to the PSG by writing to 16 memory-mapped registers. Each of the 16 registers within the PSG is also readable so that the microprocessor can determine, as necessary, present states or stored data values.

All functions of the PSG are controlled through its 16 registers which once programmed, generate and sustain the sounds, thus freeing the system processor for other tasks.

REGISTER ARRAY

The principal element of the PSG is the array of 16 read/write control registers. These 16 registers look to the CPU as a block of memory and as such occupy a 16 word block out of 1,024 possible addresses. The 10 address bits (8 bits on the common data/address bus, and 2 separate address bits A8 and $\overline{A9}$) are decoded as follows:

*$\overline{A9}$	A8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
0	1	0	0	0	0	0	0	0	0
THRU									
0	1	0	0	0	0	1	1	1	1
HIGH ORDER (Chip Select)						LOW ORDER (Register #)			

* $\overline{A9}$ is not provided on the AY-3-8912.

The four low order address bits select one of the 16 registers (R0--$R17_8$). The six high order address bits function as "chip selects" to control the tri-state bidirectional buffers (when the high order address bits are "incorrect", the bidirectional buffers are forced to a high impedance state). High order address bits $\overline{A9}$ A8 are fixed in the PSG design to recognize a 01 code; high order address bits DA7--DA4 may be mask-programmed to any 4-bit code by a special order factory mask modification. Unless otherwise specified, address bits DA7--DA4 are programmed to recognize only a 0000 code. A valid high order address latches the register address (the low order 4 bits) in the Register Address Latch/Decoder block. A latched address will remain valid until the receipt of a new address, enabling multiple reads and writes of the same register contents without the need for redundant re-addressing.

Conditioning of the Register Address Latch/Decoder and the Bidirectional Buffers to recognize the bus function required (inactive, latch address, write data, or read data) is accomplished by the Bus Control Decode block.

SOUND GENERATING BLOCKS

The basic blocks in the PSG which produce the programmed sounds include:

Tone Generators — produce the basic square wave tone frequencies for each channel (A,B,C)

Noise Generator — produces a frequency modulated pseudo random pulse width square wave output.

Mixers — combine the outputs of the Tone Generators and the Noise Generator. One for each channel (A,B,C).

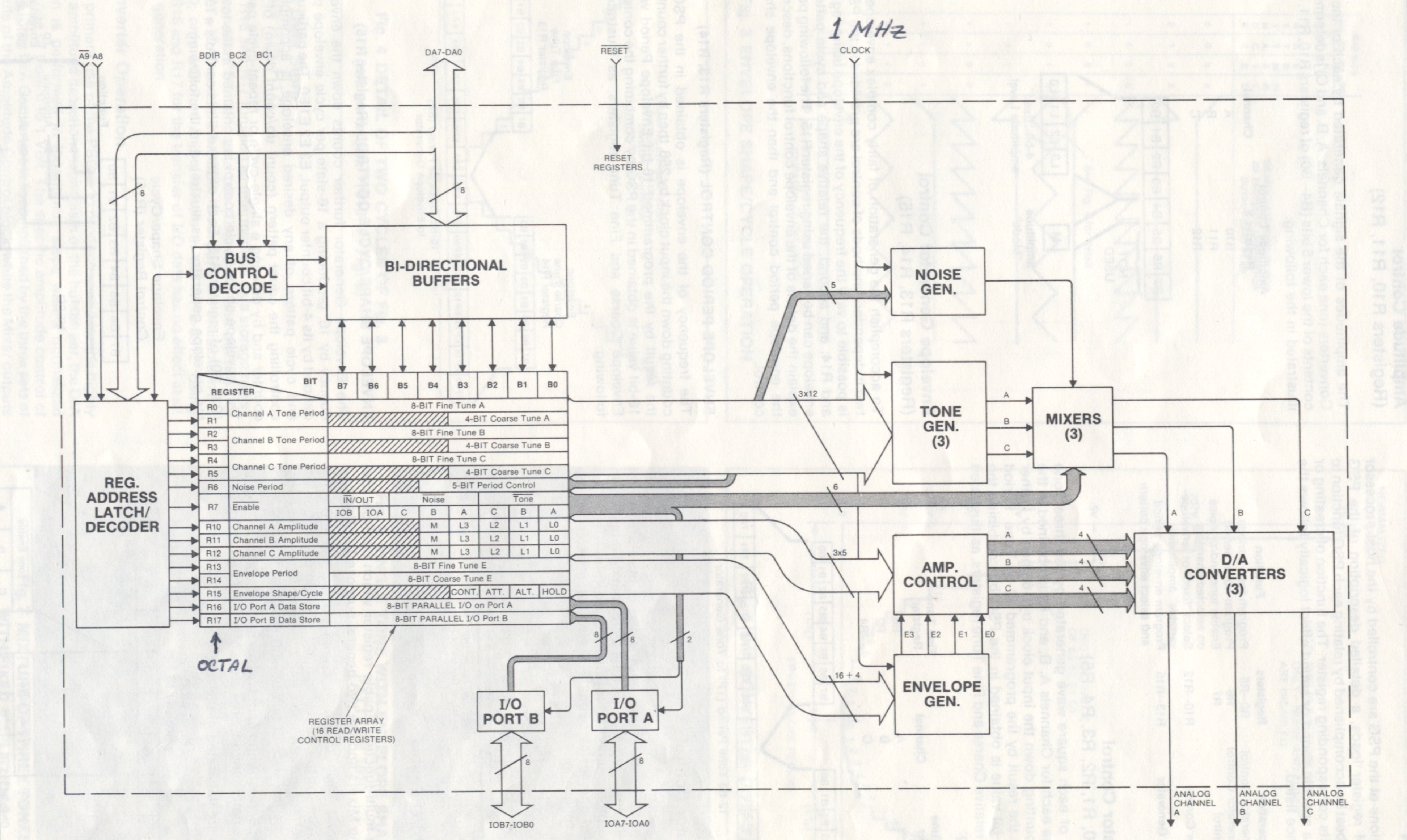
Amplitude Control — provides the D/A Converters with either a fixed or variable amplitude pattern. The fixed amplitude is under direct CPU control; the variable amplitude is accomplished by using the output of the Envelope Generator.

Envelope Generator — produces an envelope pattern which can be used to amplitude modulate the output of each Mixer.

D/A Converters — the three D/A Converters each produce up to a 16 level output signal as determined by the Amplitude Control.

I/O PORTS

Two additional blocks are shown in the PSG Block Diagram which have nothing directly to do with the production of sound—these are the two I/O Ports (A and B). Since virtually all uses of microprocessor-based sound would require interfacing between the outside world and the processor, this facility has been included in the PSG. Data to/from the CPU bus may be read/written to either of two 8-bit I/O Ports without affecting any other function of the PSG. The I/O Ports are TTL-compatible and are provided with internal pull-ups on each pin. Both Ports are available on the AY-3-8910; only I/O Port A is available on the AY-3-8912.



REGISTER		B7	B6	B5	B4	B3	B2	B1	B0
R0	Channel A Tone Period	8-BIT Fine Tune A							
R1						4-BIT Coarse Tune A			
R2	Channel B Tone Period	8-BIT Fine Tune B							
R3						4-BIT Coarse Tune B			
R4	Channel C Tone Period	8-BIT Fine Tune C							
R5						4-BIT Coarse Tune C			
R6	Noise Period				5-BIT Period Control				
R7	$\overline{\text{Enable}}$	$\overline{\text{IN}}$/OUT		$\overline{\text{Noise}}$			$\overline{\text{Tone}}$		
		IOB	IOA	C	B	A	C	B	A
R10	Channel A Amplitude				M	L3	L2	L1	L0
R11	Channel B Amplitude				M	L3	L2	L1	L0
R12	Channel C Amplitude				M	L3	L2	L1	L0
R13	Envelope Period	8-BIT Fine Tune E							
R14		8-BIT Coarse Tune E							
R15	Envelope Shape/Cycle					CONT.	ATT.	ALT.	HOLD
R16	I/O Port A Data Store	8-BIT PARALLEL I/O on Port A							
R17	I/O Port B Data Store	8-BIT PARALLEL I/O Port B							

Fig. 2 PSG BLOCK DIAGRAM

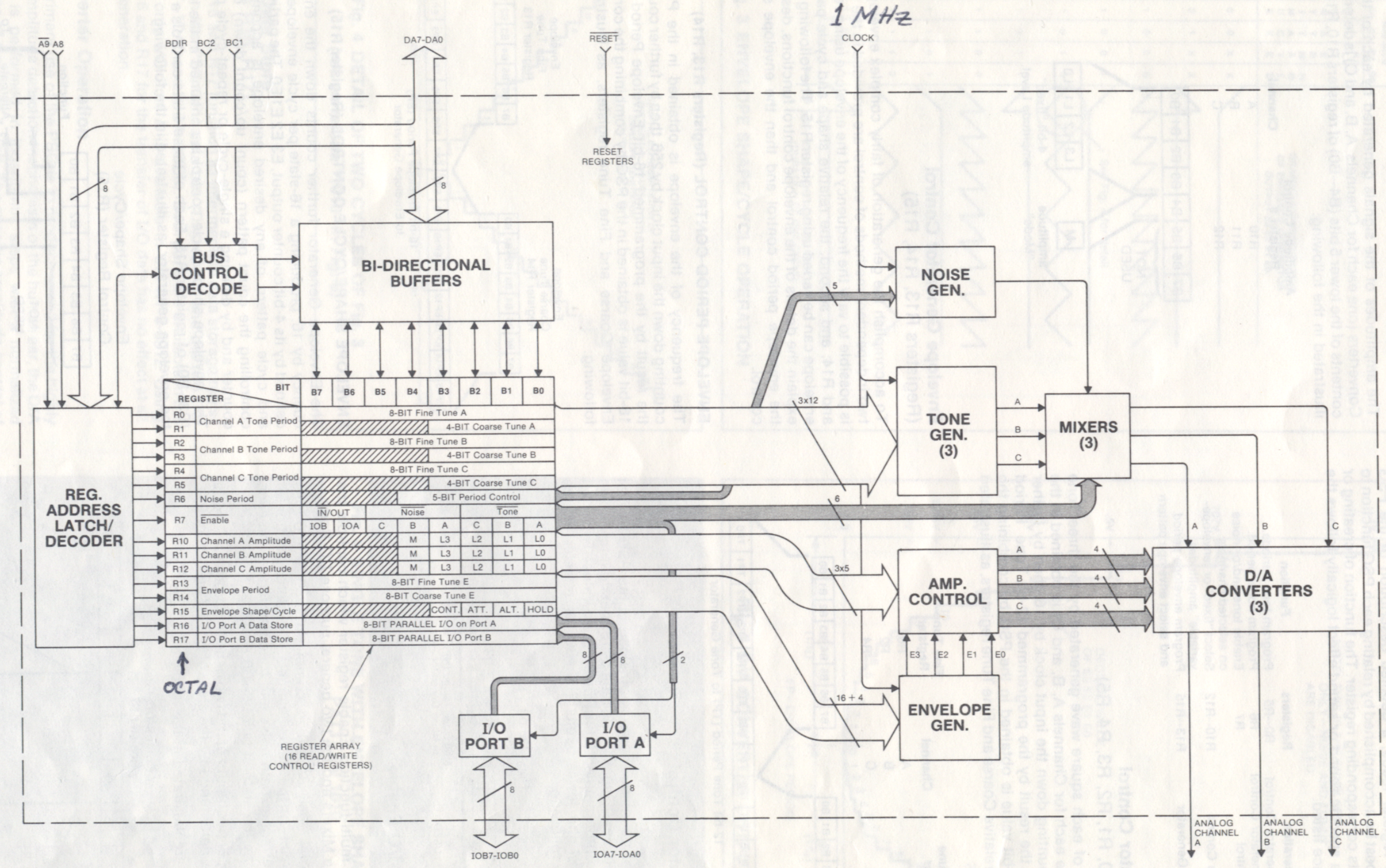

1 MHz
CLOCK
RESET
RESET REGISTERS
A9 A8
BDIR BC2 BC1
DA7-DA0
8
BUS CONTROL DECODE
BI-DIRECTIONAL BUFFERS
REG. ADDRESS LATCH/ DECODER
REGISTER
BIT
B7 B6 B5 B4 B3 B2 B1 B0
R0 Channel A Tone Period 8-BIT Fine Tune A
R1 4-BIT Coarse Tune A
R2 Channel B Tone Period 8-BIT Fine Tune B
R3 4-BIT Coarse Tune B
R4 Channel C Tone Period 8-BIT Fine Tune C
R5 4-BIT Coarse Tune C
R6 Noise Period 5-BIT Period Control
R7 Enable
IN/OUT
Noise
Tone
IOB IOA C B A C B A
R10 Channel A Amplitude M L3 L2 L1 L0
R11 Channel B Amplitude M L3 L2 L1 L0
R12 Channel C Amplitude M L3 L2 L1 L0
R13 Envelope Period 8-BIT Fine Tune E
R14 8-BIT Coarse Tune E
R15 Envelope Shape/Cycle CONT. ATT. ALT. HOLD
R16 I/O Port A Data Store 8-BIT PARALLEL I/O on Port A
R17 I/O Port B Data Store 8-BIT PARALLEL I/O Port B
OCTAL
REGISTER ARRAY (16 READ/WRITE CONTROL REGISTERS)
5
3x12
6
3x5
16 + 4
2
8
NOISE GEN.
TONE GEN. (3)
MIXERS (3)
AMP. CONTROL
ENVELOPE GEN.
E3 E2 E1 E0
A
B
C
4
D/A CONVERTERS (3)
ANALOG CHANNEL A
ANALOG CHANNEL B
ANALOG CHANNEL C
I/O PORT B
I/O PORT A
IOB7-IOB0
IOA7-IOA0

Fig. 2 PSG BLOCK DIAGRAM

OPERATION

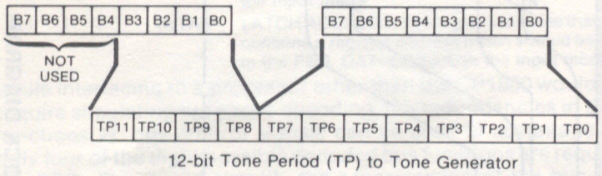
Since all functions of the PSG are controlled by the host processor via a series of register loads, a detailed description of the PSG operation can best be accomplished by relating each PSG function to the control of its corresponding register. The function of creating or programming a specific sound 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	R10--R12	Select "fixed" or "envelope-variable" amplitudes.
Envelope Generator Control	R13--R15	Program envelope period and select envelope pattern

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

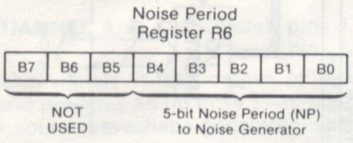
The frequency of each square wave generated by the three Tone Generators (one each for Channels A, B, and C) is obtained in the PSG by first counting down the input clock by 16, then by further counting down the result by the programmed 12-bit Tone Period value. Each 12-bit value is obtained in the PSG 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


B7 B6 B5 B4 B3 B2 B1 B0
NOT USED
B7 B6 B5 B4 B3 B2 B1 B0
TP11 TP10 TP9 TP8 TP7 TP6 TP5 TP4 TP3 TP2 TP1 TP0
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:



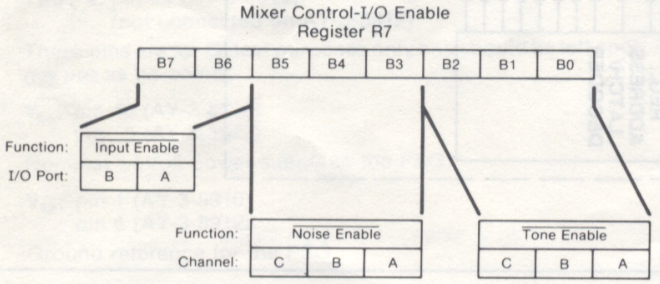
Mixer Control-I/O Enable
(Register R7)

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.

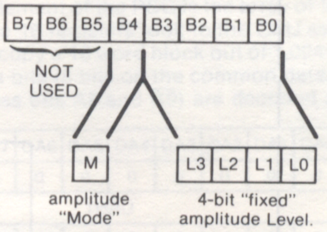
These functions are illustrated in the following:



Amplitude Control
(Registers R10, R11, R12)

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 R10, R11, and R12 as illustrated in the following:

Amplitude Control Register #	Channel
R10	A
R11	B
R12	C

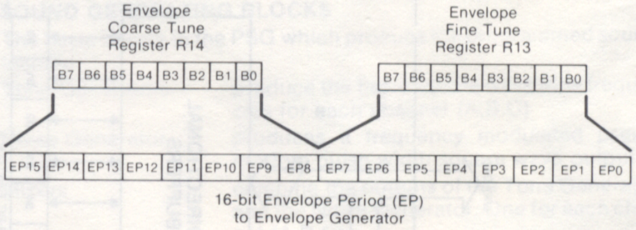


Envelope Generator Control
(Registers R13, R14, R15)

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 R13 and R14; and second, the relative shape and cycle pattern of the envelope can be varied using register R15. 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 R13, R14)

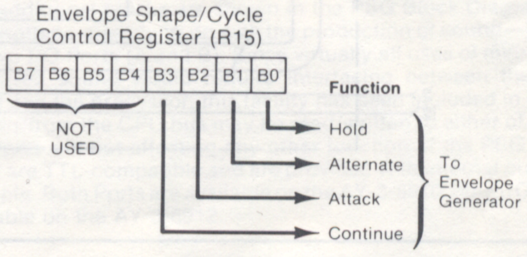
The frequency of the envelope is obtained in the PSG by first counting down the input clock by 256, then by further counting down the result by the programmed 16-bit Envelope Period value. This 16-bit value is obtained in the PSG by combining the contents of the Envelope Coarse and Fine Tune registers,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


Envelope Coarse Tune Register R14
B7 B6 B5 B4 B3 B2 B1 B0
Envelope Fine Tune Register R13
B7 B6 B5 B4 B3 B2 B1 B0
EP15 EP14 EP13 EP12 EP11 EP10 EP9 EP8 EP7 EP6 EP5 EP4 EP3 EP2 EP1 EP0
16-bit Envelope Period (EP) to Envelope Generator

ENVELOPE SHAPE/CYCLE CONTROL (Register R15)

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 R15. Each of these 4 bits controls a function in the envelope generator, as illustrated in the following:



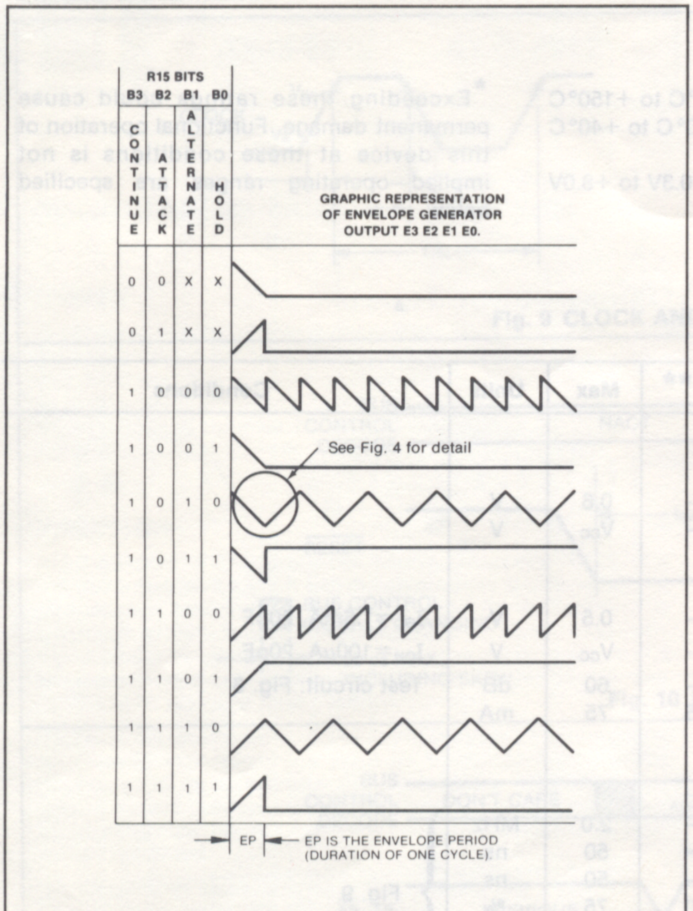


Fig. 3 ENVELOPE SHAPE/CYCLE OPERATION

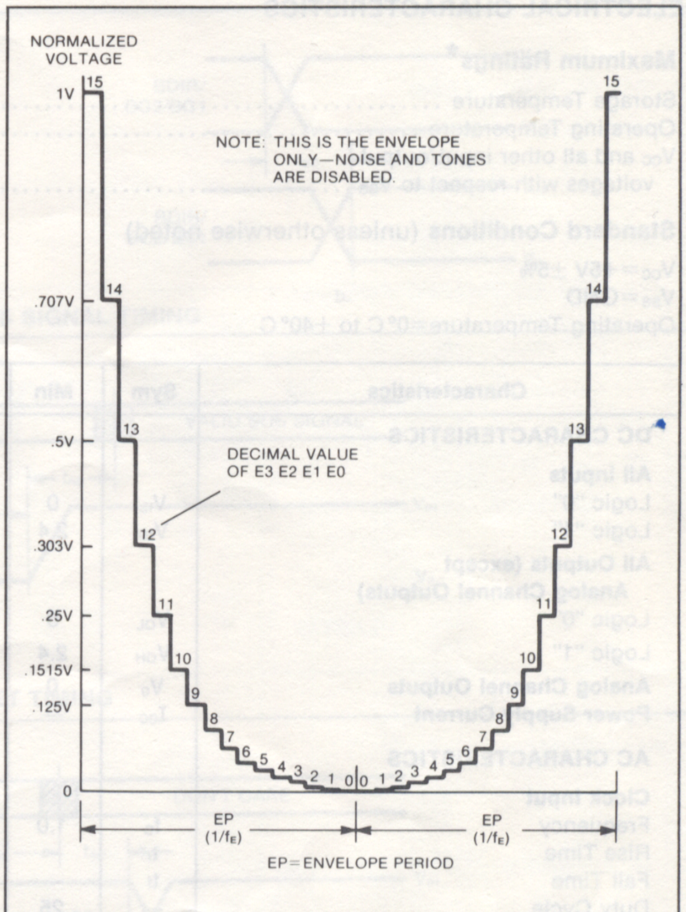


Fig. 5 D/A CONVERTER OUTPUT

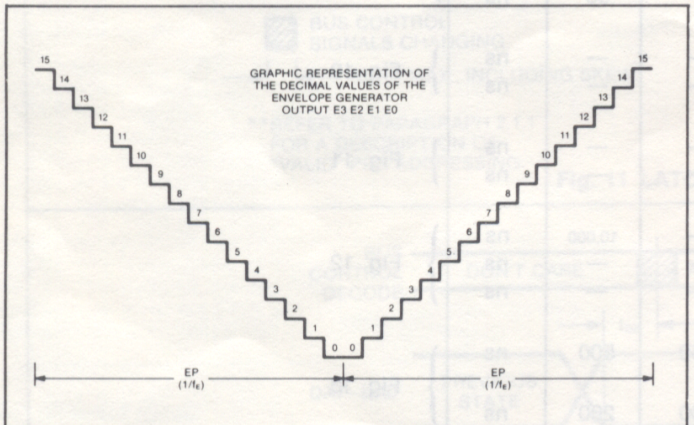


Fig. 4 DETAIL OF TWO CYCLES OF Fig. 3 (ref. waveform "1010" in Fig. 3)

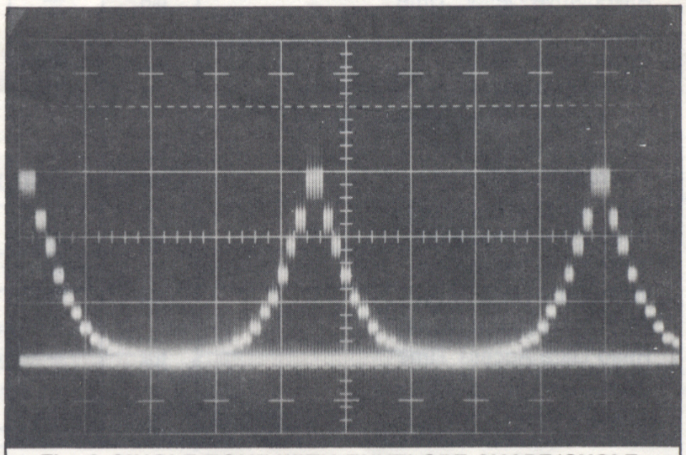


Fig. 6 SINGLE TONE WITH ENVELOPE SHAPE/CYCLE PATTERN 1010

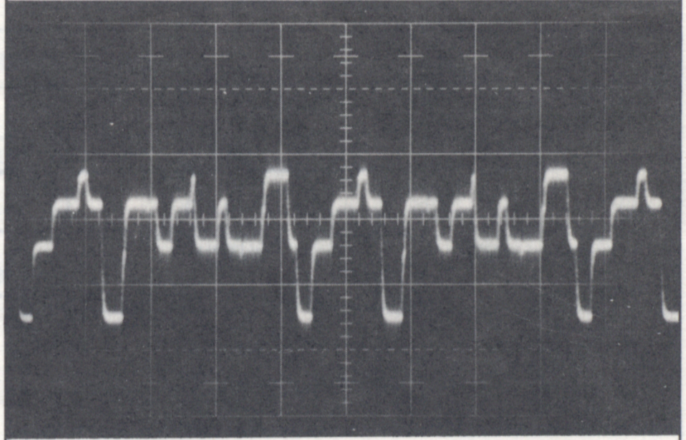


Fig. 7 MIXTURE OF THREE TONES WITH FIXED AMPLITUDES

I/O Port Data Store
(Registers R16, R17)

Registers R16 and R17 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 R16 and R17 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).

ELECTRICAL CHARACTERISTICS

Maximum Ratings*

Storage Temperature .... −55°C to +150°C
Operating Temperature .... 0°C to +40°C
V_{CC} and all other input/output
voltages with respect to V_{SS} .... −0.3V to +8.0V

*Exceeding these ratings could cause permanent damage. Functional operation of this device at these conditions is not implied—operating ranges are specified below.

Standard Conditions (unless otherwise noted)

V_{CC}=+5V ±5%
V_{SS}=GND
Operating Temperature=0°C to +40°C

Characteristics	Sym	Min	Typ**	Max	Units	Conditions
DC CHARACTERISTICS						
All Inputs						
Logic "0"	V_{IL}	0	—	0.6	V	
Logic "1"	V_{IH}	2.4	—	V_{CC}	V	
All Outputs (except Analog Channel Outputs)						
Logic "0"	V_{OL}	0	—	0.5	V	I_{OL}=1.6mA, 20pF
Logic "1"	V_{OH}	2.4	—	V_{CC}	V	I_{OH}=100μA, 20pF
Analog Channel Outputs	V_O	0	—	60	dB	Test circuit: Fig. 8
Power Supply Current	I_{CC}	—	45	75	mA	
AC CHARACTERISTICS						
Clock Input						
Frequency	f_C	1.0	—	2.0	MHz	Fig. 9
Rise Time	t_r	—	—	50	ns	
Fall Time	t_f	—	—	50	ns	
Duty Cycle	—	25	50	75	%	
Bus Signals (BDIR, BC2, BC1)						
Associative Delay Time	t_{BD}	—	—	50	ns	
Reset						
Reset Pulse Width	t_{RW}	500	—	—	ns	Fig. 10
Reset to Bus Control Delay Time	t_{RB}	100	—		ns	
A9, A8, DA7--DA0 (Address Mode)						
Address Setup Time	t_{AS}	400	—		ns	Fig. 11
Address Hold Time	t_{AH}	100	—	—	ns	
DA7--DA0 (Write Mode)						
Write Data Pulse Width	t_{DW}	500	—	10,000	ns	Fig. 12
Write Data Setup Time	t_{DS}	50	—		ns	
Write Data Hold Time	t_{DH}	100	—	—	ns	
DA7--DA0 (Read Mode)						
Read Data Access Time	t_{DA}	—	250	500	ns	Fig. 13
DA7--DA0 (Inactive Mode)						
Tristate Delay Time	t_{TS}	—	100	200	ns	

**Typical values are at +25°C and nominal voltages.

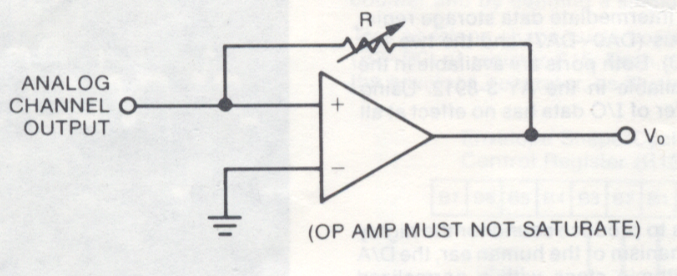


Fig. 8 ANALOG CHANNEL OUTPUT TEST CIRCUIT

TIMING DIAGRAMS

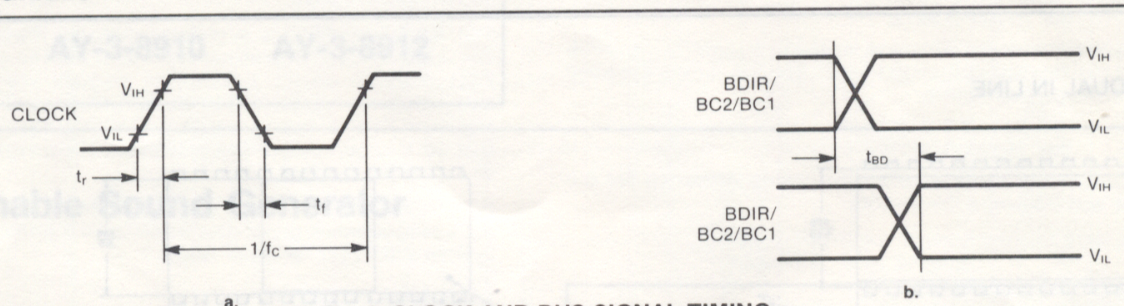


Fig. 9 CLOCK AND BUS SIGNAL TIMING

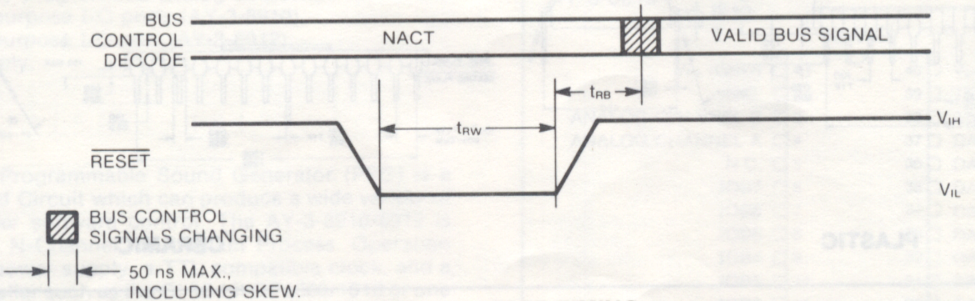
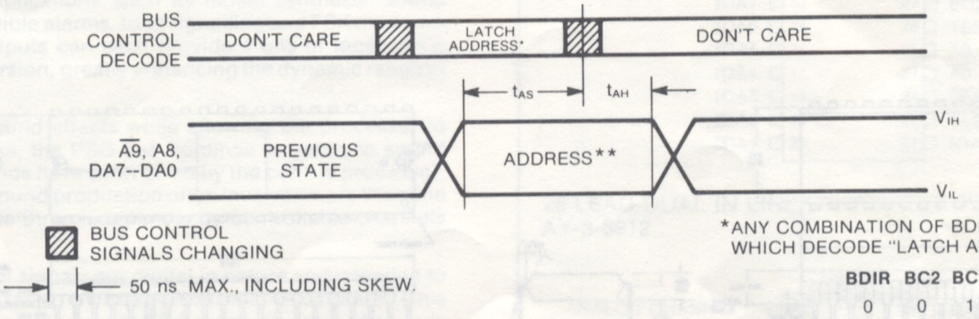


Fig. 10 RESET TIMING

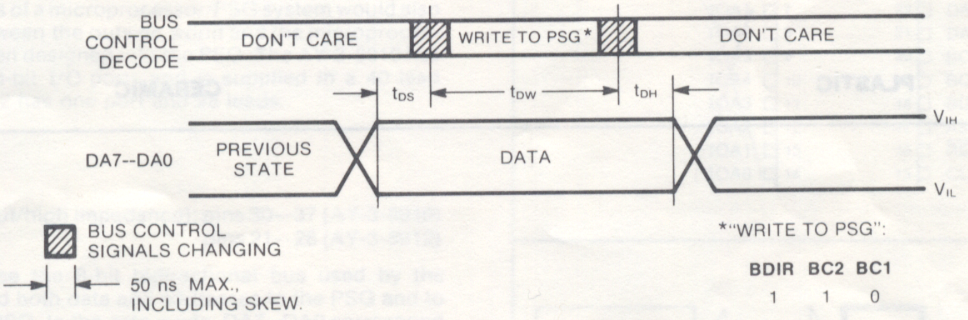


*ANY COMBINATION OF BDIR, BC2, BC1 WHICH DECODE "LATCH ADDRESS":

	BDIR	BC2	BC1
	0	0	1
or	1	0	0
or	1	1	1

**REFER TO PARAGRAPH 2.1.1
FOR A DESCRIPTION OF
"VALID" PSG ADDRESSING.

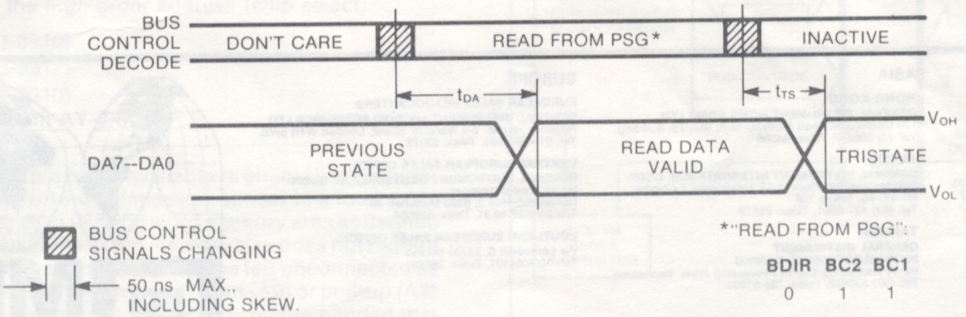
Fig. 11 LATCH ADDRESS TIMING



*"WRITE TO PSG":

BDIR	BC2	BC1
1	1	0

Fig. 12 WRITE DATA TIMING



*"READ FROM PSG":

BDIR	BC2	BC1
0	1	1

Fig. 13 READ DATA TIMING

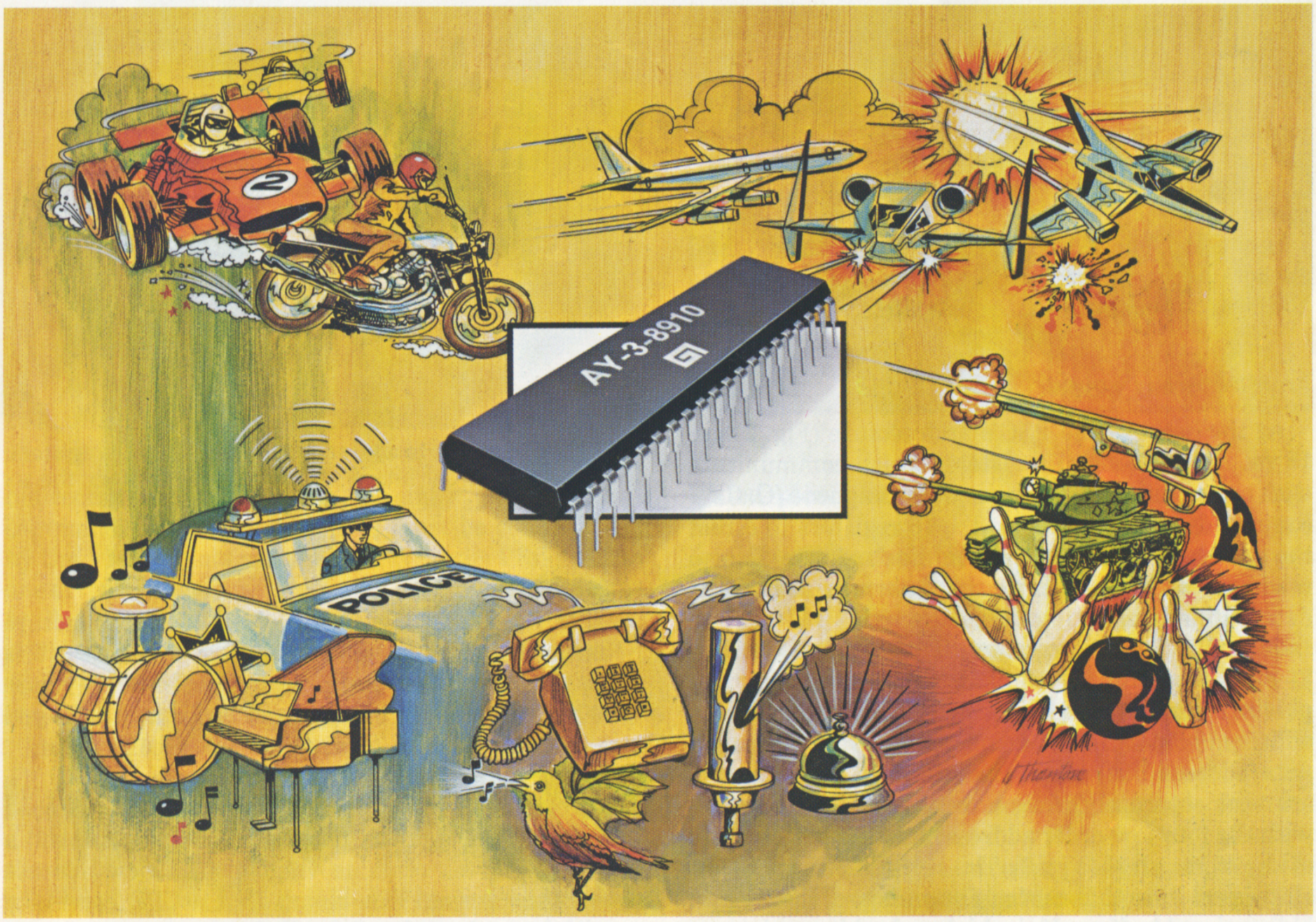
Software-controlled sound IC gives 'command performance'

Music and microprocessors usually don't mix—too much valuable time is wasted when a μP has to produce the music or sound effects for electronic games and musical instruments. But turn the job over to an LSI programmable sound generator, and you'll get software-produced sounds without constant attention from the processor.

Steven Burstein, Senior MOS Design Engineer, **Tom Mariner,** Software Consultant, and **John Wunner,** Group Director, Communications, General Instrument Corp., Microelectronics, 600 West John St., Hicksville, NY 11802.

You do it by letting μP commands direct the sound-creating circuits of a General Instrument AY-3-8910/8912 programmable sound generator (PSG), shown in Fig. 1. Not only can the IC play complete songs and generate complex sounds totally under software control, it can also sustain sound while the processor goes off to refresh, scan and service other system peripherals. What's more, the N-channel, ion-implant chip interfaces easily to any bus-oriented system. All you'll need to operate the PSG are a single 5-V supply, a TTL-compatible clock and either an 8 or 16-bit μP controller.

Since the PSG is controlled entirely by digital



1. **Sound effects, music, noise**—virtually no sound is beyond the capability of General Instrument's programmable sound generator ICs, the AY-3-8910 and 8912. Under microprocessor control, the devices generate sounds entirely from digital input commands. And microprocessor servicing of the chips is minimal.

GIMINI Cricket

The chip that chirps...

or cheeps, bleeps, hums, peeps, buzzes, dings, rings, roars, beeps, toots, blips, bongs, honks, hoots, hics, pings, booms, whistles, [illegible]ats, or...

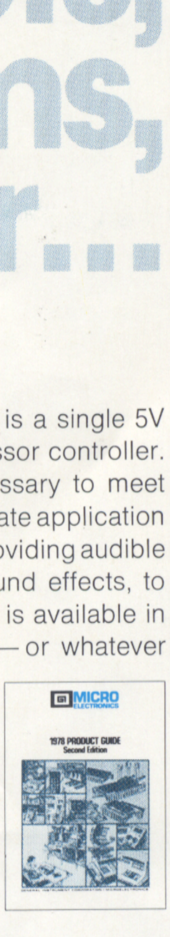
The point is, if you need a sound, GI's new "GIMINI Cricket" programmable sound generator, the AY-3-8910, can produce it. This gives a designer practically unlimited possibilities because, under full software control, the chip can generate complex sounds or combinations of sounds — music to soothe, rings and buzzes to alarm, and just about anything in between.

The AY-3-8910 is a natural for any products using microprocessors, interfacing easily to most 8- and 16-bit MPU's. In addition, it readily connects to most single-chip microcomputers — our PIC series, for example. The low-cost, AY-3-8910 has three independently programmed sound channels, an analog envelope generator, and two general purpose 8-bit I/O ports.

And all that's required for operation is a single 5V supply, a 2 MHz clock and a microprocessor controller.

With virtually all the hardware necessary to meet most audio needs, the AY-3-8910 has immediate application for a wide variety of systems and products providing audible signals, synthesized music and unique sound effects, to name but a few. The "GIMINI Cricket" chip is available in quantity and ready to chirp, cheep, beep — or whatever you decide to program into your product.

For more information on the AY-3-8910 and a free copy of our 1978 Product Guide, write to General Instrument Microelectronics, 600 West John Street, Hicksville, New York 11802, or call (516) 733-3379.

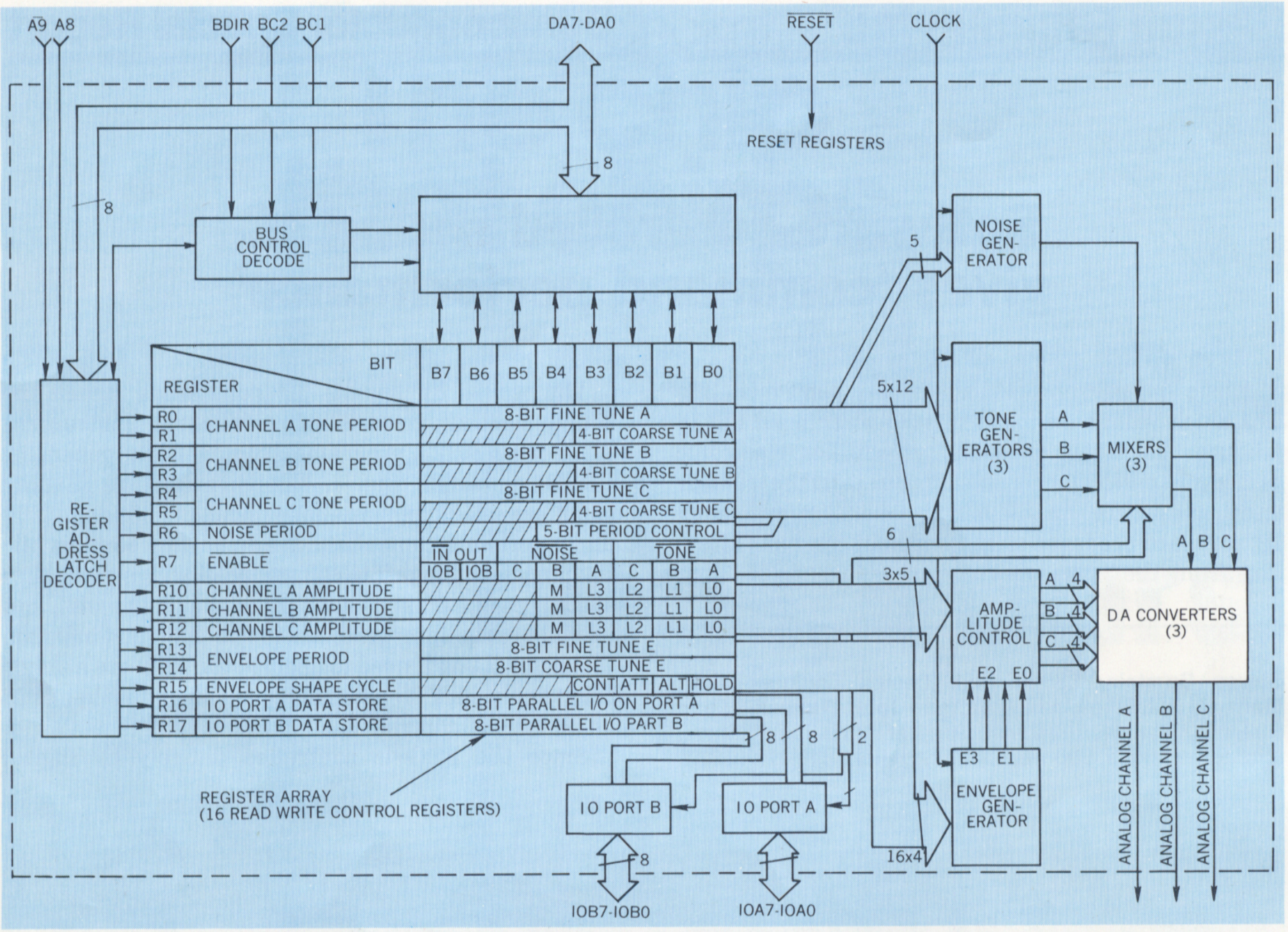


GI MICRO ELECTRONICS

1978 PRODUCT GUIDE
Second Edition

We help you compete.

GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS



REGISTER		B7	B6	B5	B4	B3	B2	B1	B0
R0	CHANNEL A TONE PERIOD	8-BIT FINE TUNE A							
R1						4-BIT COARSE TUNE A			
R2	CHANNEL B TONE PERIOD	8-BIT FINE TUNE B							
R3						4-BIT COARSE TUNE B			
R4	CHANNEL C TONE PERIOD	8-BIT FINE TUNE C							
R5						4-BIT COARSE TUNE C			
R6	NOISE PERIOD				5-BIT PERIOD CONTROL				
		$\overline{\text{IN}}$ OUT		$\overline{\text{NOISE}}$			$\overline{\text{TONE}}$		
R7	$\overline{\text{ENABLE}}$	IOB	IOB	C	B	A	C	B	A
R10	CHANNEL A AMPLITUDE				M	L3	L2	L1	L0
R11	CHANNEL B AMPLITUDE				M	L3	L2	L1	L0
R12	CHANNEL C AMPLITUDE				M	L3	L2	L1	L0
R13	ENVELOPE PERIOD	8-BIT FINE TUNE E							
R14		8-BIT COARSE TUNE E							
R15	ENVELOPE SHAPE CYCLE					CONT	ATT	ALT	HOLD
R16	IO PORT A DATA STORE	8-BIT PARALLEL I/O ON PORT A							
R17	IO PORT B DATA STORE	8-BIT PARALLEL I/O PART B							

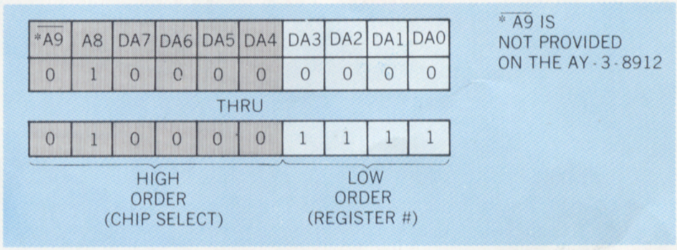
2. **Sixteen memory-mapped registers, designated R_0 through R_{17}, direct operations** inside the PSG. Once programmed, sounds are generated and sustained without further processor intervention. At the output, three independently programmed analog channels can create realistic sound effects or music.

commands, it generates a wide variety of sounds without any changes to its external circuitry. And chances are good that just a single device can create all the music or sound that your system will ever need.

Architecturally sound

At the heart of the PSG's architecture (Fig. 2) are 16 read/write-control registers, which communicate with a processor through a memory-mapped I/O. CPU commands are written into the 16 registers to direct the sound generator's operation. Each register is also readable, so the processor can determine its stored data value at any time.

To a CPU, the 16 registers (R_0 through R_{17} in Fig. 2) appear as a 16-word block of memory out of a possible 1024 address locations. Each register has ten address bits—8 bits come in on the common data/address bus, while two other bits (A_8 and $\overline{A_9}$) are sent separately. Addresses are decoded on-chip, with the four low-order address bits used to select one of the 16 registers (Fig. 3). The six high-order bits are chip-select lines that control the on-board, three-state bidirectional buffers. If the high-order bits contain an incorrect code, the buffers are forced into their high-impedance states.



$\overline{\text{A9}}$*	A8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
0	1	0	0	0	0	0	0	0	0
THRU									
0	1	0	0	0	0	1	1	1	1
HIGH ORDER (CHIP SELECT)						LOW ORDER (REGISTER #)			

* $\overline{\text{A9}}$ IS NOT PROVIDED ON THE AY-3-8912

3. **It takes ten bits to address** one of the PSG's 16 registers. The four low-order bits select one of the registers; the six high-order bits are chip select lines.

Bits A_8 and $\overline{A_9}$ are fixed in the PSG architecture—only code 01 is recognized. High-order address bits DA_4 through DA_7 are also normally fixed (only code 0000 is acceptable), but you can order any code you want by requesting a factory-mask modification.

When a valid high-order address enters the PSG, the register address (four low-order bits) is latched in the register-address/decoder block (Fig. 2). And a latched address remains valid until a new address enters the device. This means that a CPU can make multiple reads and writes of the same register contents without redundant addressing.

Before either the register-address/decoder or the

Music and sound effects—a combo on a chip

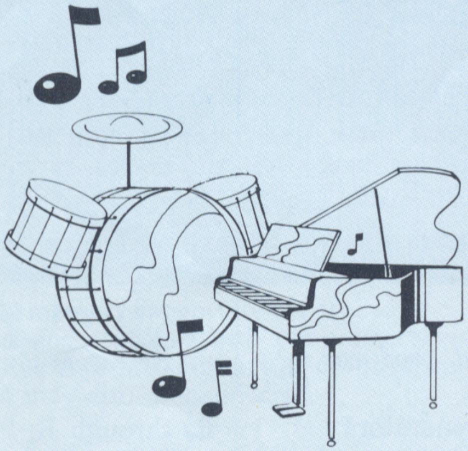
What your mind recognizes as music is essentially a set of mathematical relationships to a digital system like General Instrument's PSG. For example, shifting up or down in octaves can be represented by multiplying or dividing by a power of two, a simple shift operation for most microprocessors. Because the PSG operates digitally, musical notes are produced by simple register loads. Register load values create the first octave note which is shifted to the correct octave at playtime.

As good a music player as the PSG is, equally important is its sound effects capability. In fact, the chip is powerful enough to produce many sounds with just the tone generators. A tone generator sequence for a European siren effect, which is two distinct frequencies, sequentially produced, is illustrated.

Other commonly used sounds require only the noise and envelope generators. And if another channel needs the envelope generator, the system processor can control the channel envelope. An example of a sound produced with the noise and envelope generators is an explosion—it's pure noise with a decaying envelope. You can also produce a gunshot with this sequence by modifying the length of the envelope.

Siren sequence

Register #	Octal Load Value	Explanation
Any not specified	000	—
R0	376	Set channel A tone period to 2.27 ms (440 Hz).
R1	000	
R7	076	Enable tone only on channel A only.
R10	017	Select maximum amplitude on channel A.
(Wait approximately 350 ms before continuing).		
R0	126	Set channel A tone period to 5.346 ms (187).
R1	001	
(Wait approximately 350 ms before continuing).		
R10	000	Turn off channel A to end sound effect.



Explosion sequence

Register #	Octal Load Value	Explanation
Any not specified	000	—
R6	000	Set noise period to max. value.
R7	007	Enable noise only, on channels A, B, C.
R10	020	Select full amplitude range under direct control of envelope generator.
R11	020	
R12	020	
R14	070	Set envelope period 2.05 seconds.
R15	000	Select envelope "decay", one cycle only.

bidirectional buffer blocks recognize the function commanded from the bus, they must be conditioned by the bus-control decode block. Then the PSG enters one of four states: Inactive, Latch Address, Write Data or Read Data.

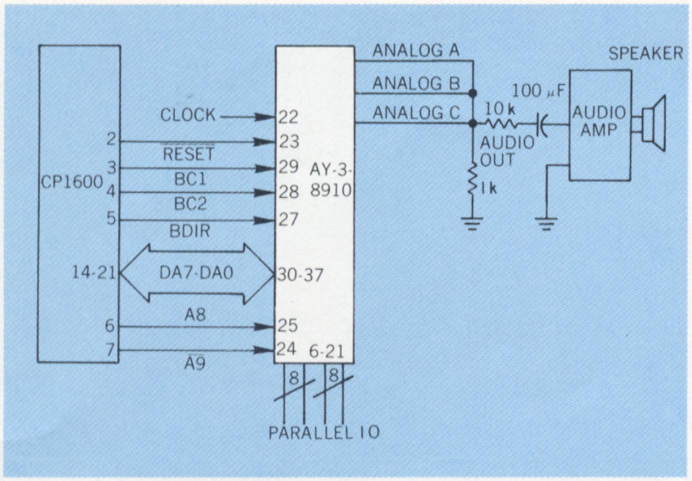
That's just the input side of the PSG. The output, which generates the programmed sounds, has its own series of functional blocks.

Sounding it out

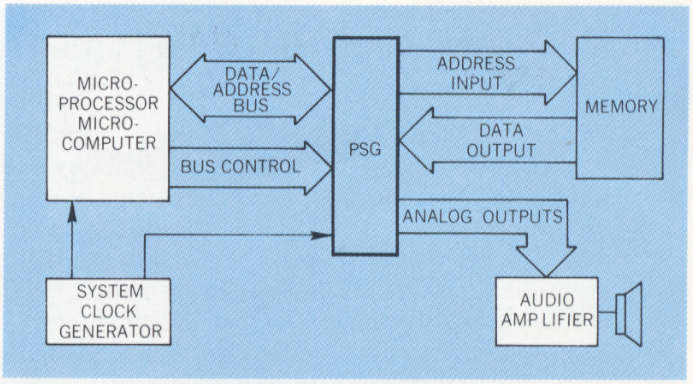
The basic blocks producing sound in the PSG are shown in Fig. 2 as:

- Tone generators to develop the basic square-wave tone frequencies for each of the three analog-output channels.
- A noise generator to produce a frequency-modulated pseudorandom-pulse-width square-wave output.
- Mixers that combine the tone-generator outputs with the noise-generator output. There's one mixer for each of the three analog-output channels.
- An envelope generator to produce a pattern that amplitude-modulates the output of each mixer.
- An amplitude control to provide internal d/a converters with either a fixed or variable-amplitude pattern. Fixed amplitude is under CPU control; variable amplitude uses the output of the envelope generator.
- Three d/a converters, each producing up to a 16-level output signal as determined by the amplitude control. 4-bit.

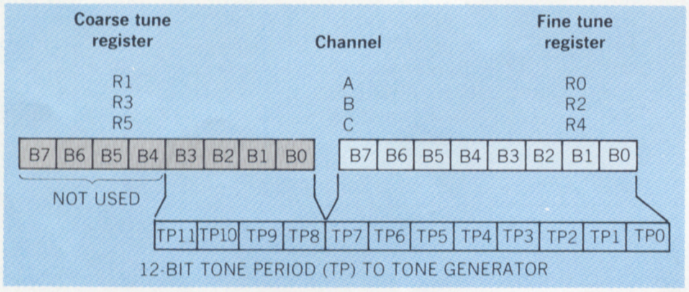
Actual sound signals for driving a speaker appear at three analog-channel outputs (channels A, B and



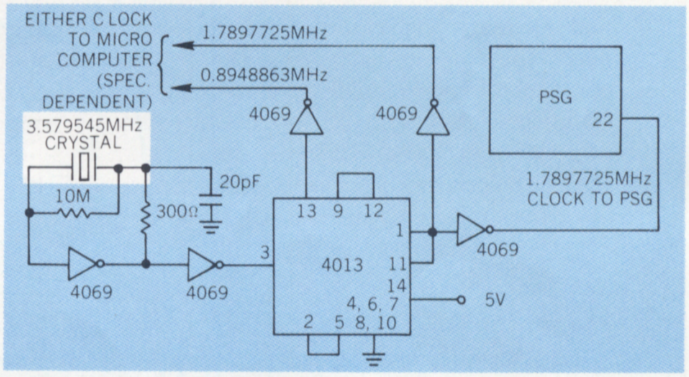
4. **A typical μP-PSG interface** requires a microprocessor, a TTL-compatible clock and just one 5-V supply. In this system, the analog-output channels are paralleled, but each channel can be driven separately.



6. **The key elements of a PSG interface** are indicated by the white blocks in this diagram. A system clock, an audio amplifier and a microprocessor are mandatory. However, the memory block, which can be a ROM or RAM, is optional —it's only needed when a system requires additional data for processor support.



5. **Two registers, one for coarse tuning and one for fine tuning,** are responsible for generating the PSG's 12-bit tone period. The tone period is sent to the tone-generator circuits to produce square-wave frequencies.



7. **Generate the system clock easily** with this simple circuit, which uses a standard color-burst crystal.

C in Fig. 2). A typical system, showing the PSG interfaced to a microprocessor, with its paralleled output channels driving a speaker is shown in Fig. 4. Each output channel provides four bits of logarithmic d/a conversion, enhancing the dynamic range of the sound. And since the analog channels can be independently programmed and driven through separate speakers, you'll be able to generate realistic sound effects.

Although the PSG's I/O ports have nothing to do with sound production, they are important to the μP-PSG communications interface. Two 8-bit ports are provided on the 40-pin AY-3-8910. The smaller AY-3-8912 (28-pins) has only one 8-bit port. All I/O port inputs are TTL-compatible, and contain internal pull-ups on each pin. The significant feature is that data going to or from a CPU can be written or read to the ports without affecting any other functions or operations of the PSG.

Now you have an over-all picture of what's inside a PSG. How it produces varied and complex sounds is another matter.

Registers call the tune

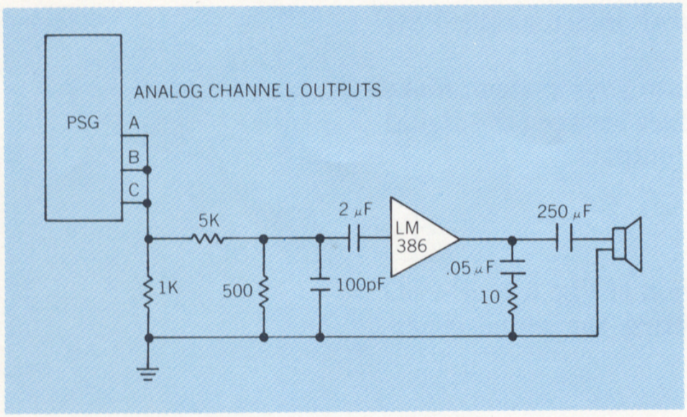
Since all PSG sound production is dictated by a host processor through commands to the 16 registers, the registers themselves are assigned to control the five actual sound producing blocks:

Sound producer	*Registers*
Tone generators	R_0 through R_5
Noise generator	R_6
Mixer control	R_7
Amplitude control	R_{10} through R_{12}
Envelope generator	R_{13} through R_{15}

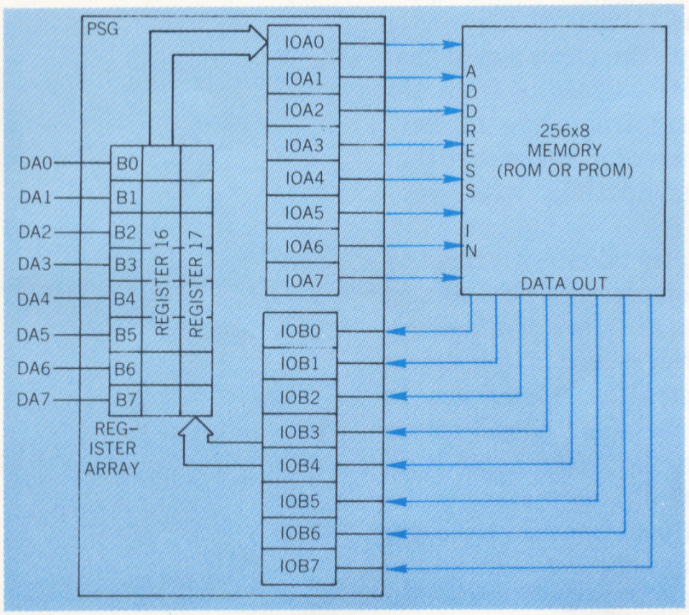
Registers R_{16} and R_{17} are responsible for controlling the two I/O ports (on the 8910 only).

By examining the tone generators' operation, you'll get an idea of how to program the other sound blocks. Each square wave produced by the three tone generators comes from dividing the input clock by 16, then counting that result down by the programmed, 12-bit tone-period value. The 12-bit value is determined by combining the contents of two registers—a coarse tune and a fine-tune register—as shown in Fig. 5.

The 12-bit result in the combined register represents a period value—the higher the value in the register, the lower the tone frequency (T = 1/f). Because of the PSG's internal count-down technique, the lowest period value is 000000000001 (divide by 1), the highest



8. **To connect an audio-interface to the sound generator,** put an LM-386 amplifier between speaker and PSG. Here the PSG outputs are summed together.



9. **External memory support** for the processor controlling the PSG is accomplished with the connections shown. Although this interface is for a ROM or PROM only, a RAM or EAROM may also be connected.

is 000000000000 (divide by 4096_{10}).

Two equations describe the relationship between the desired output tone frequency and the input clock frequency and tone period value:

$$f_t = \frac{f_{clock}}{16\ TP_{10}} \qquad (1)$$

$$TP_{10} = 256\ CT_{10} + FT_{10} \qquad (2)$$

where f_t is the desired tone frequency,

f_{clock} is the input clock frequency,

TP_{10} is the decimal equivalent of the tone period bits TP_{11} through TP_0,

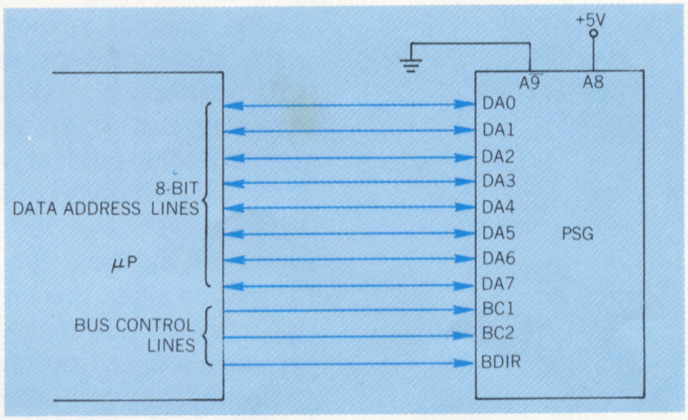
CT_{10} is the decimal equivalent of the coarse-tune-register bits B_0 through B_3 (TP_{11} through TP_8), and

FT_{10} is the decimal equivalent of the ~~coarse~~-tune-register bits B_0 through B_7 (TP_0 through TP_7).

Bus control	Bus codes BDIR	 BC2	 BC1	Explanation of bus data (DA7--DA0)
Latch address	1	1	1	00000111: Latch R7 to program I/O ports
Write to PSG	1	1	0	01000000: Set B7, B6 to 0,1 respectively
Latch address	1	1	1	00001110: Latch R16 to address memory
Write to PSG	1	1	0	00000001: Address data to memory
Latch address	1	1	1	00001111: Latch R17 to read memory
Read from PSG	0	1	1	XXXXXXXX: Memory data contained in R17

Note: BC2 in the above Bus Codes may be permanently tied to +5V thus requiring only two bus control lines for all control operations.

10. **To address and read an external memory,** a bus-control sequence like this operates into the PSG's two 8-bit I/O ports. Note that bus code BC2 can be permanently tied to 5 V. Therefore, just two bus control lines are needed for all control operations.



11. **An 8-bit bus passes all data and address** information between PSG and processor in the μP-PSG interface. And BC1, BC2 and BDIR are the only control signals needed to direct the operations.

Equations 1 and 2 tell you that f_t has a range whose low side is $f_{clock}/65{,}536$ ($TP_{10}=4096_{10}$) and whose high side is $f_{clock}/16$ ($TP_{10} = 1$). With a 2-MHz input clock, the range of tone frequencies that you can get would be 30.5 Hz to 125 kHz.

Say you want to calculate values for the contents of the tone period's coarse and fine-tune registers. Assume you know the input clock frequency and the tone period you want to generate. First, rearrange Eqs. 1 and 2.

$$TP_{10} = \frac{f_{clock}}{16 f_t} \qquad (3)$$

$$CT_{10} + FT_{10}/256 = TP_{10}/256 \qquad (4)$$

If f_{clock} is 2 MHz and you want to generate an f_t of 100 Hz,

$$TP_{10} = \frac{2 \times 10^6}{16\ (1 \times 10^2)} = 1250$$

Substitute the value for TP_{10} into Eq. 4:

$$\begin{aligned} CT_{10} + FT_{10}/256 &= 1250/256 \\ &= 4 + 226/256 \end{aligned}$$

Therefore, $CT_{10} = 4_{10} = 0100$ (B_3 through B_0)

and $FT_{10} = 226_{10} = 11100010$ (B_7 through B_0).

This calculation for tone-period register programming is similar to that required for the noise generator and envelope generator. For complete details of these blocks, plus an in-depth explanation of the amplitude control and mixer registers, d/a converter and I/O

6

port operation, consult General Instrument's PSG Data Manual.

Before designing the PSG into your system, make sure you understand the interface between the digital inputs and the analog sound outputs.

From input to output

As shown in the block diagram of Fig. 6, a μP-PSG interface involves interconnecting the sound generator to four subsystems:

- A clock generator.
- An audio-output interface.
- An external memory (optional).
- A microprocessor/microcomputer.

An economical way to provide a system clock is shown in Fig. 7. All you need are a 3.579545-MHz standard color-burst crystal, a CD4069 CMOS inverter and a CD4013 dual-D flip-flop. The clock signal that drives the PSG—1.7897725 MHz—results from the flip-flop's dividing the crystal frequency in half. Before you adopt this circuit however, make sure that the crystal frequency is compatible with your processor's specifications.

The audio-output interface also takes few parts and is simple to build. Fig. 8 illustrates the PSG analog channel outputs added together to drive a commercial-version LM386 IC audio amplifier, which in turn drives a speaker. The summing operation allows you to generate complex waveforms for amplification through a single speaker. On the other hand, each channel can be individually amplified for more exotic sound effects. In the PSG, output channels are separately controlled by amplitude registers R_{10}, R_{11} and R_{12} and enable register R_7 (see Fig. 2)

If you find you have to provide additional data information for processor support, you can connect an external memory to the PSG (see Fig. 9). Two on-chip registers communicate with the memory—I/O port A (8-bit) addresses it and I/O port B (also 8-bit) reads data from it.

Fig. 10 shows an example of the bus-control sequence needed to address and read an external memory. The memory is connected to the I/O ports—port A addresses and port B reads.

Depending on your design, a RAM or EAROM can replace the ROM or PROM in Fig. 9. In that case, Port B becomes the I/O and the sequence in Fig. 10 must be altered. Port B is then able to write data as an output and read data as an input. In another application, these two I/O ports could be used as switch inputs or as display outputs.

One more interface remains—the microprocessor interface. Signal lines DA_7 through DA_0 in Fig. 11 are I/O bus bits 7 through 0. All data and address information passes between the PSG and processor across this bus. Lines BC_1, BC_2 and BDIR are generated by the processor to direct all bus operations, which are identified as Latch Address, Write To PSG, Read From PSG and Inactive.■■