Texas Instruments

APPLICATIONS REPORT

by Ted Mahler Applications Engineer

A GUIDE TO USING THE TEXAS INSTRUMENTS SN76489A SOUND GENERATOR

INTRODUCTION

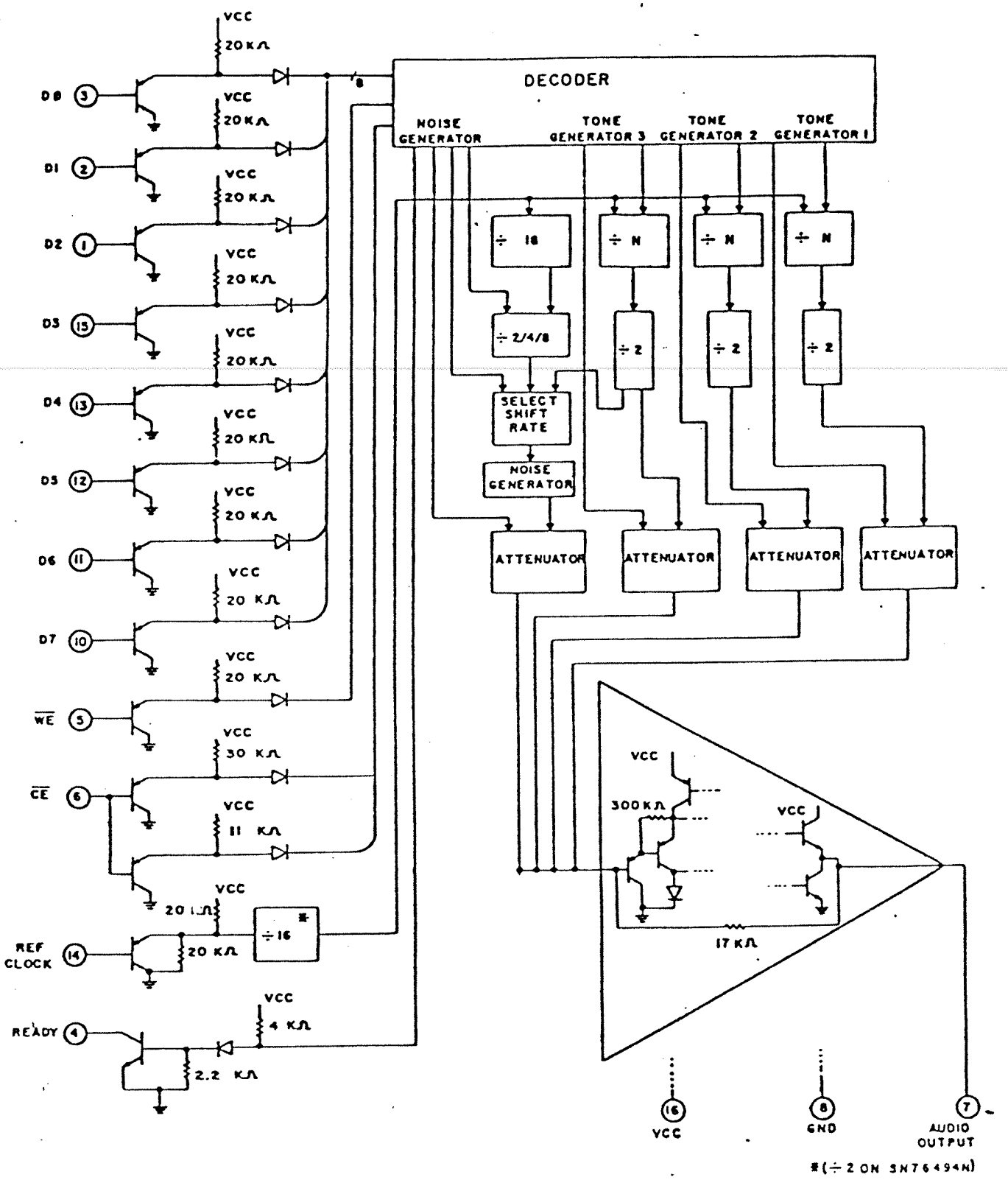
The TEXAS INSTRUMENTS SN76489A sound generator is designed to provide a low cost means of adding sound generation capabilities to a microprocessor system. These sounds can include sound effects for video games, alarms for home or industrial use, or any application requiring audio feedback.

Internally, the SN76489A contains 3 programmable tone generators, each with its own programmable attenuator, and a noise source with its own attenuator. By using tone generators or combinations of tone generators and noise, an extremely wide variety of sounds can be easly created.

A complete description of the functions available in the SN76489A along with electrical characteristics is presented in the TEXAS INSTRUMENTS data sheet for this device. This application report contains examples of the methods needed to generate control bytes for the SN76489A, along with interfacing data and examples.

TEXAS INSTRUMENTS also manufactures the SN76489 which is pin compatable with the SN76489A with one exception. Audio out on the SN76489 is 100ma, which is capable of driving a small capacitively coupled speaker. The SN76489A has a 10ma output which will not drive a speaker but should be capacitively coupled into an amplifer. Software generation for both devices is identical.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222



SN76489A / SN76494 SCHELOGIC

The SCHELOGIC of the SN76489A illustrates both its logic arrangement and the actual components on each on the pins of this device. This information is presented to aid in understanding and interfacing the SN76489A.

This example will show how to program a 400HZ tone on tone generator 2 with a clock frequency of 4MHZ.

A) First find n;

a) n = clock/(32(tone))
b) n = 4Mz/(32(400HZ))
c) n = 312.5

Note that since n can only be an integer, you must change it to be either 312 or 313. This decesion depends on the particular sound you are trying to obtain. For this example, 312 will be used. By substituting this number back into the equation you can find the actual frequency that will be produced.

a) n = clock/(32(tone))
b) tone = clock/(32(n))
c) tone = 4MHZ/(32(312))
d) tone = 400.6HZ

B) Now convert n to a 10 bit binary number.

Frequency bit#	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
	0	1	0	0	1	1	1	0	0	0

C) Using the frequency format;

*								*							
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	R0	R1	R2	F6	F7	F8	F9	0	X	F0	F1	F2	F3	F4	F5

* MSB

the two bytes needed will be:

BYTE 1 1 0 1 0 1 0 0 0

BYTE 2 0 0 0 1 0 0 1 1

White noise can easly be produced by selecting two options and using them to construct a one byte control word for the SN76489A.

A) The first option is the noise feedback control (FB). A FB bit of 0 selects periodic noise which sounds like a low frequency tone. White noise, which sounds like a hiss, is generated by a FB bit of 1.

B) The noise source in the SN76489A is a shift register with an exclusive OR feedback. The rate at which it shifts is dependent on the clock and two noise frequency (NF) bits which make up the second option. A description of the two bits is given in the following table:

BITS NF0	NF1	
0	0	HIGHER PITCH (LESS COARSE HISS)
0	1	
1	0	LOWER PITCH (MORE COARSE)

C) The options are now placed in the proper format for a noise control byte.

*							
B	B	B	B	B	B	B	B
I	I	I	I	I	I	I	I
T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7
1	R0	R1	R2	X	FB	NF0	NF1

* MSB

Bits 1, 2, and 3 tell the SN76489A that this is a noise byte and should be as follows, R0 = 1, R1 = 1, and R2 = 0. Therefore a noise byte to select white noise with the less coarse hissing sound will be:

1 1 1 0 0 1 0 0

The rate at which the noise clock shifts can also be controlled by the frequency of tone generator 3. To transfer control to tone 3 set both the NF0 and NF1 bits to 1. Now vary the noise generator, either periodic or white, by changing the frequecy of tone generator 3.

The third major function available in the SN76489A is attenuation. Each of the three tone generators and the noise source has its own attenuator. The procedure for controlling each of these is the same.

A) First select the device, within the SN76489A, which is to be attenuated. Each device and its associated code is :

ATTENUATION REGISTER CODE			DEVICE
R0	R1	R2	
0	0	1	TONE 1
0	1	1	TONE 2
1	0	1	TONE 3
1	1	1	NOISE

B) Next select the desired level of attenuation as shown in this table.

ATTENUATION	ATTENUATION CODE				ATTENUATION	ATTENUATION CODE			
	A0	A1	A2	A3		A0	A1	A2	A3
0 db	0	0	0	0	16 db	1	0	0	0
2 db	0	0	0	1	18 db	1	0	0	1
4 db	0	0	1	0	20 db	1	0	1	0
6 db	0	0	1	1	22 db	1	0	1	1
8 db	0	1	0	0	24 db	1	1	0	0
10 db	0	1	0	1	26 db	1	1	0	1
12 db	0	1	1	0	28 db	1	1	1	0
14 db	0	1	1	1	OFF	1	1	1	1

C) Now construct the attenuation byte using this format:

*							
B	B	B	B	B	B	B	B
I	I	I	I	I	I	I	I
T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7
1	R0	R1	R2	A0	A1	A2	A3

* MSB

Therefore to attenuate Tone generator 3 by 4db the byte would be:

1 1 0 1 0 0 1 0

An 18db attenuation of the noise source would be:

1 1 1 1 1 0 0 1

The SN76489A is interfaced through 3 control lines and 8 data lines. For the following definitions a high logic level is a minimum of 2 volts and a low is a maximum of .8 volts.

CHIP ENABLE ($\overline{CE}$)	When low, this pin enables the SN76489A and drops the READY line to a low state. This pin must remain low until the completion of the data transfer. In most applications the CE pin is connected to address decoding logic.
WRITE ENABLE ($\overline{WE}$)	When active, low, this pin signals the SN76489A that data is available on the data bus. As with CE this pin must remain low during the data transfer. A memory write line is typically connected here.
READY	When the SN76489A is ready to accept data this pin will be high. It will drop to a low level when CE goes low. From the falling edge of WE, the READY line will remain low for 32 clock cycles.
DATA LINES (D0-D7)	These pins accept data for the SN76489A. Since the SN76489A requires 32 clock cycles for a data transfer, data must be available on these pins for that period of time.

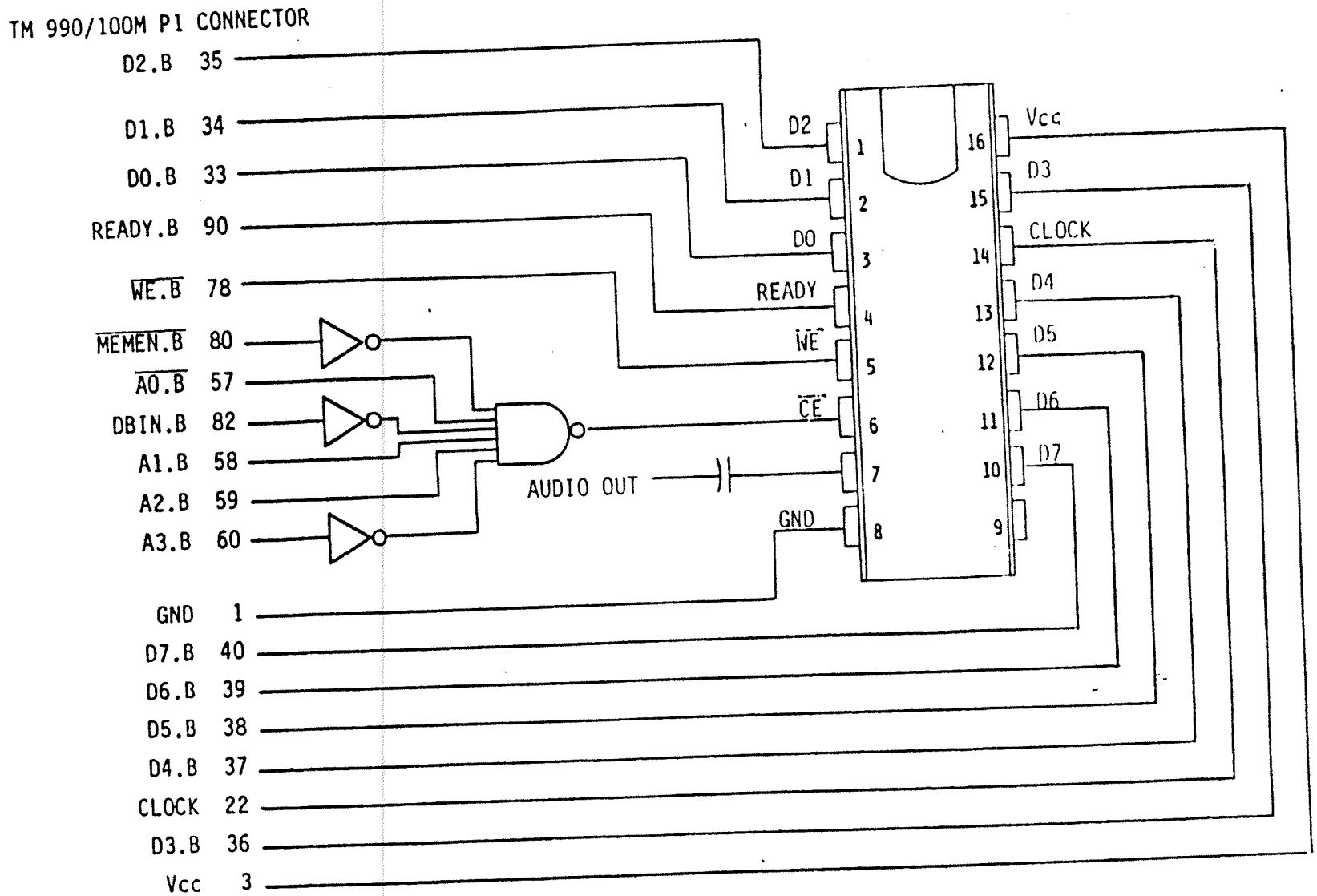
INTERFACING EXAMPLES

CIRCUIT 1. TMS990/100M COMPUTER TO SN76489A INTERFACE

This interface illustrates the use of the READY line to synchronize the SN76489A and the TMS990/100M. With this circuit the CPU will halt, allowing the slower SN76489A to accept the data without necessitating any sort of software timing loops. Note that not all processors will function using this type of interface. A processor must be able to complete the write cycle (place data on the data bus and enable its WRITE ENABLE line) after its READY line is disabled. If the processor's architecture is not designed this way or if the processor is not capable of being halted, an interface such as circuit 2 should be used.

Addressing used in CIRCUIT 1 will select the SN76489A when writing to any address from E000 to EFFF. This 4K block of memory can be reduced by further decoding of the address lines if necessary.

TM 990/100M TO SN76489A INTERFACE



APPLICATIONS ENGINEERING

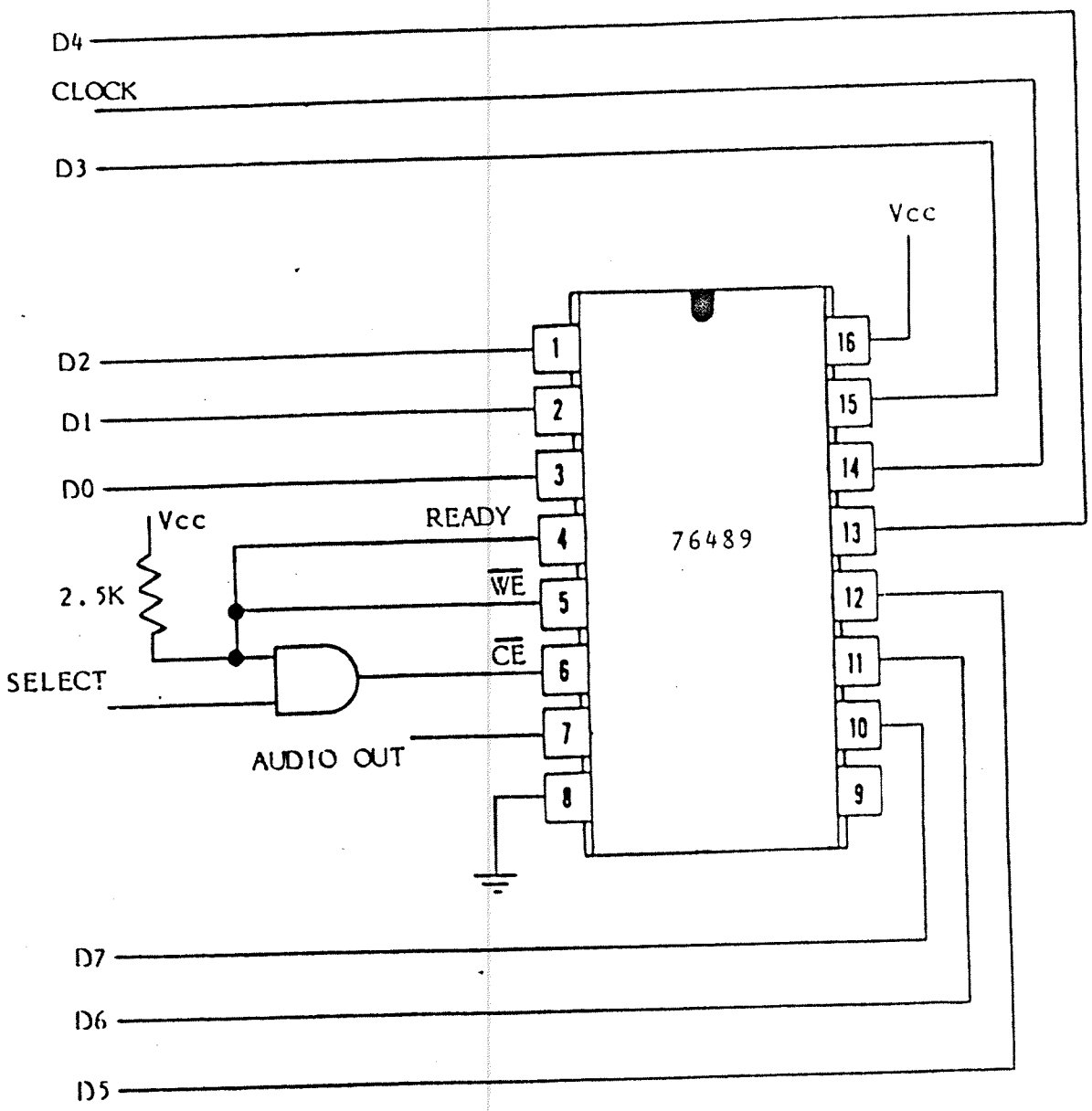
CIRCUIT 2. PARALLEL PORT INTERFACE

By using the READY line to control the duration of the WRITE ENABLE and CHIP ENABLE pulse, the SN76489A is assured that these pulses will be in their proper state for the proper amount of time. Two points must be considered when using this interface:

1. The data lines must maintain the data for the entire transfer period (32 clock cycles). This can be accomplished by latching the data lines.

2. Software using this interface must contain timing loops to avoid transfering a data byte to the SN76489A before it has had time to load the previous one. If a machine language program is being used, the number of machine cycles in the instructions can be counted and used to construct a 32 cycle loop. If a higher level language is being used, generally timing loops for data transfers are not necessary.

PARALLEL PORT INTERFACE



NOTES

1. The data lines must be latched so that the data remains on them for at least 32 clock cycles after the select line goes low.
2. The select pulse should be a negative going pulse a minimum of 150ns wide.
3. Audio out should be capacitively coupled to either an 8 OHM speaker (SN76489) or an audio amplifer (SN76489A).
4. Pin 9 must be left open.

APPLICATIONS ENGINEERING

CLOCK CHIME

The clock chime sound can be generated by using two of the tone generators and their attenuators. Tone generator 1 is set to a frequency of 679HZ at 0 dB down, while tone 2 is set to a frequency of 694 HZ at 24 dB down. While holding tone 2 steady, tone 1 attenuation in ramped up until completely off. A time delay is used to hold tone 1 for a period of time at each attenuation level. The two frequencies, being slightly different, beat to create the chime sound. The chime is repeated by ramping tone 1 attenuation again.

The following is a list of bytes which when presented to the SN76489A, with the suggested time intervals, will produce this sound. All frequencies were calculated using a 1MHZ clock.

HEX BYTE TO SN76489A	TIME INTERVAL FROM LAST BYTE	COMMENTS
9F		TURN OFF TONE 1
BF	*	TURN OFF TONE 2
DF	*	TURN OFF TONE 3
FF	*	TURN OFF NOISE
84	*	STE TONE 1 AT 679HZ
0A		
90	*	SET TONE 1 ATTENUATION AT 0 dB DOWN.
A0	*	SET TONE 2 AT 694HZ
0A		
BB	*	SET TONE 2 ATTENUATION AT 24 dB DOWN.
91	*	TONE 1 ATTENUATED 2dB
92	114 ms	" " " 4dB
93	"	" " " 6dB
94	"	" " " 8dB
95	"	" " " 10dB
96	"	" " " 12dB
97	"	" " " 14dB
.	.	
.	.	
.	.	
9F	"	TONE 1 OFF
BF	"	TONE 2 OFF

* This period of time does not affect the chime sound. It should be long enough to allow a data transfer from the controlling system to the SN76489A.

MISSILE

The sound of a missile or jet can be produced using the noise generator and tone generator number 3. Generation of the sound is done by clocking the noise generator from tone generator 3 and then sweeping the frequency of tone 3 from a high frequency to a low frequency. During the sweep, the noise generator is attenuated to give the effect of distance to the sound.

HEX BYTE TO SN76489A	TIME INTERVAL FROM LAST BYTE	COMMENTS
9F	*	TURN OFF TONE 1
BF	*	" " " 2
DF	*	" " " 3
FF	*	" " NOISE
E7	*	CONTROL NOISE BY TONE 3
F0	*	TURN ON NOISE
C0	*	TONE 3 FREQUENCY (HIGH)
00		
C1	18 ms	TONE 3 FREQUENCY UPDATE
00	145 us	
C2	18 ms	TONE 3 FREQUENCY UPDATE
00	145 us	
C3	18 ms	TONE 3 FREQUENCY UPDATE
00	145 us	
.		
.		
.		
CF	18 ms	TONE 3 FREQUENCY UPDATE
00	145 us	
F1	18 ms	NOISE ATTENUATION UPDATE
C0	18 ms	TONE 3 FREQUENCY UPDATE
01	145 us	
C1	18 ms	TONE 3 FREQUENCY UPDATE
01	145 us	
C2	18 ms	TONE 3 FREQUENCY UPDATE
01	145 us	
.		
.		
.		
CF	18 ms	LAST TONE 3 UPDATE WORD
0E	145 us	
FF		HIGHEST ATTENUATION LEVEL

* This period of time does not affect the sound. It should be long enough to allow the data transfer from the controlling system to the SN76489A.

Notice that a complete frequency sweep of tone three is done by incrementing the first byte of the two byte frequency word from C0 to CF while keeping the second byte at 00. After the last iteration of the first byte (CF) , the second byte is incremented by 1. Now the process repeats incrementing byte 1 from C0 to CF but this time holding byte 2 at 01. This process continues until byte 2 is at 0E. The noise attenuation can also be ramped down, during this sweep, from 0dB down to off (hex bytes F0 to FF). For the attenuation to reach maximum attenuation at the end of the sweep, it should be updated every other cycle of byte 1.

A jet sound can be produced using the same basic program. For this sound the frequency sweep of tone 3 is run from a low frequency to a high frequency then back to low frequency with the time interval between the frequency words increased.

EXPLOSIONS

Explosions are generated by setting the noise generator to white noise and ramping the noise attenuator from 0dB down (F0) to off (FF). Different time periods can be used between each attenuation update to simulate different types of explosions.

BASIC Language program listings

To aid in sound development a higher level language can be used to control the SN76489A. Listings given here were used on a system where the sound chip was interfaced using a parallel port interface. The SN76489A will then appear as a memory location which can be written to using a FILL command (on some systems this is a POKE command). These programs, of course, will not be compatable with every system, but they should present the basic method used to generate some given sounds.

```
10 REM ** BELL OR CHIME **
20 REM SYSTEM CLOCK FREQUENCY OF 2 MHZ
30 N=59392 : REM LOCATION OF SN76489A
40 FILL N,159 : REM TURN OFF TONE 1
50 FILL N,191 : REM TURN OFF TONE 2
60 FILL N,223 : REM TURN OFF TONE 3
70 FILL N,255 : REM TURN OFF NOISE
80 INPUT A$ : REM INPUT ANYTHING
90 FILL N,140 : TONE 1 AT 679 HZ
100 FILL N,5
110 FILL N,170 : REM TONE 2 AT 694 HZ
120 FILL N,5
130 FOR B=0 TO 11 : REM B=NUMBER OF BELLS
140 FOR I=145 TO 159 : REM LOOP TO GENERATE ATTENUATION STEPS
150 FILL N,I : FILL N,(I+32) : REM OUTPUT ATTENUATION TO 76489
160 FOR D=0 TO 75 : NEXT D : REM DELAY LOOP
170 NEXT I
180 NEXT B
190 PRINT "END OF SOUND"
```

```
10 REM ** BASIC BIRD SOUND **
20 REM SYSTEM CLOCK FREQUENCY OF 2 MHZ
30 N=59392 : REM LOCATION OF SN76489A
40 FILL N,159 : REM TURN OFF TONE 1
50 FILL N,191 : REM TURN OFF TONE 2
60 FILL N,223 : REM TURN OFF TONE 3
70 FILL N,255 : REM TURN OFF NOISE
80 INPUT A$ : S=0 : REM  INPUT ANYTHING
90 Z=INT(10 *(RND(0)) : REM RANDOM CHIRP LENGTH
100 FILL N,144 : REM SET TONE ATTENUATION
110 FOR I=0 TO 15 : REM START CHIRP LOOP
120 FILL N,(128+I) : REM STEP FREQUENCY FROM 3906 HZ TO 2016 HZ
130 FILL N,1
140 FOR D=0 TO Z : NEXT D : REM DELAY BY RANDOM AMOUNT
150 NEXT I
160 S=S+Z : REM COUNT TO STOP PROGRAM
170 IF S)200 THEN 190 : REM BRANCH TO END
180 GOTO 90
190 FILL N,159 : REM TURN OFF TONE 1
```

```
10 REM ** MISSILE SOUND **
20 REM SYSTEM CLOCK FREQUENCY OF 2MHZ
30 N=59392 :  REM LOCATION OF SN76489A
40 FILL N,159 : REM TURN OFF TONE 1
50 FILL N,191 : REM TURN OFF TONE 2
60 FILL N,223 : REM TURN OFF TONE 3
70 FILL N,255 : REM TURN OFF NOISE
80 INPUT A$ : REM INPUT ANYTHING
90 FILL N,231 : REM CONTROL NOISE BY TONE 3
100 FILL N,240 : REM SET NOISE ATTENUATION
110 FOR B=0 TO 15 : REM LOOP TO GENERATE SECOND BYTE
120 FOR A=192 TO 207 : REM LOOP TO GENERATE FIRST BYTE
130 FILL N,A : FILL N,B : REM OUTPUT BYTE TO 76489
140 NEXT A
150 FILL N,(240+B) : REM ATTENUATE NOISE
160 NEXT B
170 GOTO 40
```

```
10 REM ** BOMB DROP AND EXPLOSION **
20 REM SYSTEM CLOCK FREQUENCY OF 2MHZ
30 N=59392 : REM LOCATION OF SN76489A
40 FILL N,159 : REM TURN OFF TONE 1
50 FILL N,191 : REM TURN OFF TONE 2
60 FILL N,223 : REM TURN OFF TONE 3
70 FILL N,255 : REM TURN OFF NOISE
80 INPUT A$ : REM INPUT ANYTHING
90 FILL N,144 : REM ATTENUATE TONE 1 0 DB
100 FOR J=5 TO 17 : REM LOOP TO GENERATE BYTE 2 FOR SWEEP
110 FOR I=128 TO 143 : REM LOOP TO GENERATE BYTE 1 FOR SWEEP
120 FILL N,I : FILL N,J : REM OUTPUT BYTE TO SN76489A
130 FOR D=0 TO 10 : NEXT D : REM DELAY LOOP
140 NEXT I
150 NEXT J
160 FILL N,159 : REM TURN OFF TONE 1
170 FILL N,228 : REM SET NOISE TO HIGH PITCH WHITE NOISE
180 FOR I=240 TO 255 : REM LOOP TO GENERATE ATTENUATION BYTES
190 FILL N,I : REM OUTPUT BYTE TO SN76489A
200 FOR D=0 TO 75 : NEXT D : REM DELAY LOOP
210 NEXT I
```

PLEASE NOTE:

TEXAS INSTRUMENTS cannot assume responsibility for any circuits or systems shown, or represent that they are free from patent infringement.