A Super Synthesizer

Build the Speak 'N' Sing 2 speech synthesizer, which brings hi-fi speech and sound to computing.

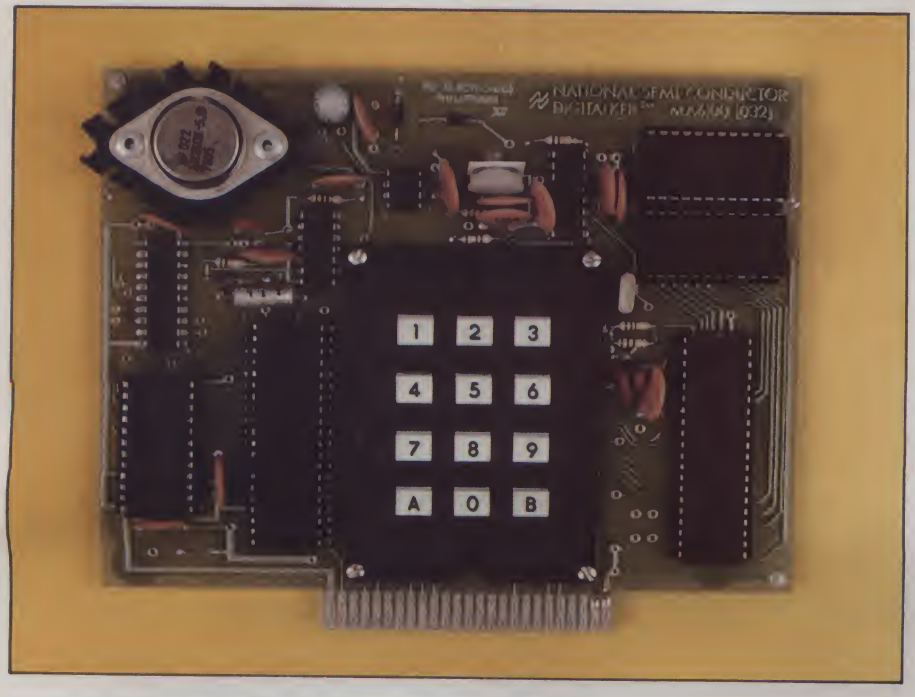
By Richard R. Parry

Hello, police? I want to report a burglary at 211 South Ave. Please send police officers to investigate. Thank you."

When I developed my home security system (see "Computerized Security and Status System," *Kilobaud Microcomputing*, November 1980, p. 30), I knew that a desirable feature of the system would be an automatic call for help. It could save both life and property if someone was breaking into the house or a fire broke out.

The call for help should be one that everyone can understand. Alarms, sirens or flashing lights have heretofore been the traditional means for a machine to communicate an emergency. However, such devices can't adequately describe the circumstances surrounding an event. Speech, on the other hand, can describe the situation.

Speech synthesis technology has advanced greatly since my article was published. With the help of LSI (large scale integration) techniques, the price of such speech synthesizers has decreased to the point where we are seeing speech being used to bridge the man-machine gap in such mundane devices as microwave ovens, vending machines and the family car.



National Semiconductor offers the evaluation board shown here. The user need only add a 9 V supply and a speaker. Commands are input via the keypad to the on-board microcomputer. The speech processor chip and the word ROMs can be seen on the right side of the board.

This article describes the Speak 'N' Sing 2 synthesizer. It can reproduce high-quality speech, music and sound effects. Speech is reproduced using the National Semiconductor Digitalker integrated circuit chip set. If you want to listen to a demonstration, call 408-737-3939.

The heart of the speech synthesizer is the speech processor chip (SPC), which is responsible for decoding the vocabulary stored in external read only memory (ROM). The board contains space for two such ROM sets. The standard ROM set contains 144 words and the optional ROM set contains 131 words.

An additional feature of the Speak 'N' Sing 2 is a 40-word FIFO buffer. This allows the processor to perform other operations while speech is being reproduced. Music and sound synthesis is done using an eight-bit digital-to-analog converter (D/A). The features include:

- High-quality speech
- 40-word FIFO buffer
- Interrupt driver capability
- Music synthesis
- Sound effects
- Audio power amplifier with volume control to drive external speaker
- Balance control to adjust volume of speech and music
- Simultaneous reproduction of music and speech
- Software available on five-inch or eight-inch disk for either 6800 or 6809 processors using Flex
- Hardware available as a board

Richard Parry (38 W 255 Deerpath Road, Batavia, IL 60510) is an electrical engineer and programmer at the Fermi National Accelerator Lab.

only, in kit form, or assembled and tested.

The Speak 'N' Sing 2 board described here was designed for the SS-50 bus. With a little ingenuity you could adapt the circuit or the board to other systems.

Speech Synthesis

There are several ways that you can program a computer to speak. Waveform digitization was one of the earliest methods developed. It's basically a brute force technique—the word or phrase to be synthesized is first recorded using an analog-to-digital converter (A/D). The digitized data can then be stored in memory, on a disk, magnetic tape or various other storing media.

Reproduction requires the opposite process. The digital information previously stored is played back through a digital-to-analog converter (D/A). Note that the Speak 'N' Sing 2 provides this method of reproducing speech or, for that matter, any other sound. But first you must digitize the audio signal using a separate A/D converter; once digitized, the D/A converter on the Speak 'N' Sing 2 board can be used to reproduce the original audio signal with exceptional quality.

This method of speech synthesis has the advantage of preserving the original speaker's voice and inflection with high fidelity. In fact, the music recording industry is increasingly using digital audio tape recorders because of their superiority over conventional analog tape recorders.

Unfortunately, the technique requires a great deal of memory. To understand why, you must understand how the speech is recorded. Digitizing the signal requires that you sample the original waveform. For good sound reproduction, the sampling rate must be at least twice the highest frequency that you want to reproduce accurately. This is referred to as the Nyquist rate. To record a male speaker, 10 kHz is an acceptable rate for high-quality recording. Recording a female's or child's voice typically requires a higher sampling rate.

Using a ten-bit A/D converter with a sampling rate of 10 kHz, a minute of speech would require nearly a megabyte of memory. This represents a bit rate of 100 Kbps (kilobits per second). Few, if any, low-cost computer systems can afford a megabyte to record a minute of speech. Obviously the sampling rate or the A/D resolution can be decreased to reduce the bit rate, but this results in lower-quality speech; the trick is to reduce the bit rate without significantly altering the quality.

> Speak 'N' Sing 2 was designed for the SS-50 bus. With a little ingenuity you could adapt it to other systems.

Other techniques such as pulse code modulation (PCM), differential PCM, delta modulation, continuous variable slope delta modulation and adaptive predictive coding can reduce the bit rate to approximately 30 Kbps. While certainly better, it is not quite good enough.

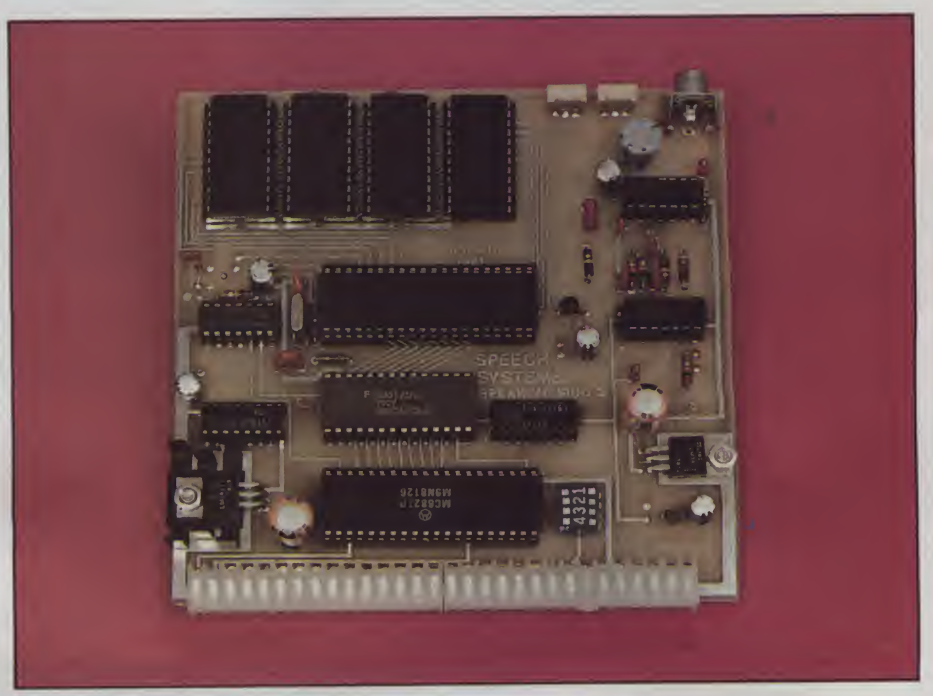
However, through computer analysis and compression, the bit rate can be reduced to approximately 1000 bps while still keeping speech fidelity. Bit rates as low as 60 to 90 bps are possible using phoneme synthesis, but the speech sounds more like Donald Duck than a human.

One of the more popular compression techniques is called linear predictive coding. If you've listened to the Texas Instruments Speak & Spell educational toy, you've heard this method in action. This technique makes it possible to encode speech with relatively little data. The name comes from the fact that LPC uses previous conditions to predict present values for filter coefficients.

The speech synthesis method used by the Speak 'N' Sing 2 is called time domain synthesis, a technique developed by Forrest Mozer. Mozer went into the field to develop a talking calculator for a blind student. The results of his efforts have benefited us all. The heart of the circuit is a speech processor chip which can address up to 128 kilobits of ROM directly. The SPC uses a speech compression technique that reduces the amount of memory needed to store speech by removing the redundant data from the speech signal. Four major methods to do this are:

1. Elimination of redundant pitch periods
2. Adaptive delta modulation coding to minimize bandwidth and memory requirements
3. Phase angle adjustments to create mirror image symmetry
4. Replacing the low-level portion of a pitch period with silence.

Figs. 1, 2 and 3 show the differences between the original waveform of the word "zone" and the digitized version after compression.



The Speak 'N' Sing 2. The four chips at the top of the board are the two ROM sets that contain a total of 275 words. Below these chips is the speech processor chip which decodes the words in the ROMs. Below the SPC is the 40-word FIFO buffer.

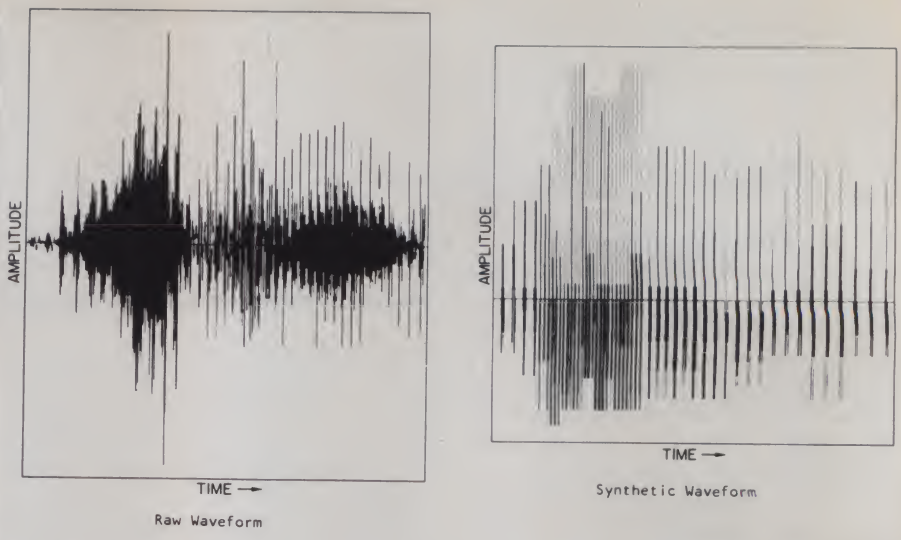


Fig. 1. The actual analog waveshape of the spoken word "zone." To the right is the digitized waveshape of the word using the compression technique called time domain synthesis. The salient point here is that the two waveshapes are very dissimilar. However, to the human ear, both waveshapes sound like the word "zone." (Courtesy of National Semiconductor, Inc.)

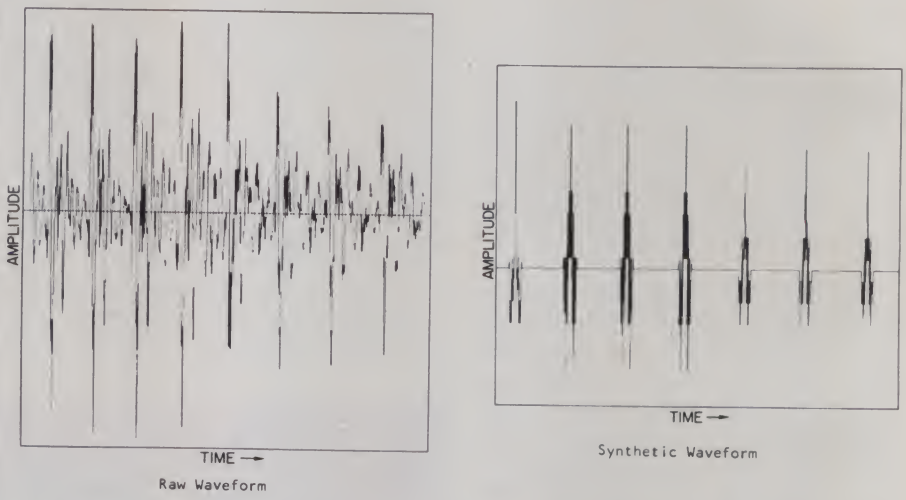


Fig. 2. An exploded view of a portion of the analog and digital waveshape of the word "zone." Once again there are vast differences between the waveshapes. Note the dead silence between the pulses of the digitized waveshape on the right, and the lack of prolonged silence in the analog representation on the left. (Courtesy of National Semiconductor, Inc.)

Fig. 1 vividly shows the differences between the two waveforms, and you might think that the two would sound quite different; in fact, they sound the same.

Fig. 2 shows the silence between pulses, which reduces the amount of information required to store speech. Fig. 3 shows the symmetry of the pulses, which further reduces the bit rate since only half of the speech information needs to be stored.

The advantage of speech compression techniques such as linear predictive coding and time domain synthesis is that both reproduce high-quality speech while drastically reducing the amount of memory required to store the speech. The memory required is sufficiently reduced to make storing words in high-density ROMs practical. A typical 8K-byte ROM can contain a full minute of high-quality speech. Prerecorded ROMs which contain many standard or commonly used words are available.

Speech Programming

Now I'll discuss some general rules and comments for programming speech using the Speak 'N' Sing 2. You program speech by transmitting a code for each word you want to reproduce. The code for each word is shown in Table 1. The standard ROM

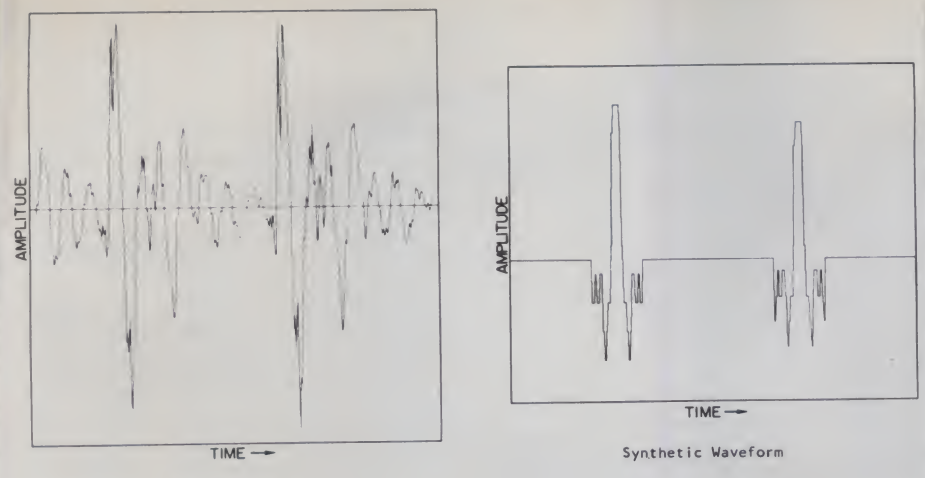


Fig. 3. A minute portion of the word "zone" in an even more exploded view. Again there are marked differences between the two waveshapes that to the human ear sound the same. The interesting point here is that the digital representation of the waveshape on the right is symmetrical. It is through this symmetry and other techniques that you can compress speech by several orders of magnitude. (Courtesy of National Semiconductor, Inc.)

Table 1. Shown here is the root vocabulary contained in the standard and optional ROM sets. Note that several of the words are actually suffixes which may be appended to other words.

ROM SET 1	HEX	DEC	ROM SET 2
THIS IS -	00	00	ABORT
ONE	01	01	ADD
TWO	02	02	ADJUST
THREE	03	03	ALARM
FOUR	04	04	ALERT
FIVE	05	05	ALL
SIX	06	06	ASK
SEVEN	07	07	ASSISTANCE
EIGHT	08	08	ATTENTION
NINE	09	09	BRAKE
TEN	0A	10	BUTTON
ELEVEN	0B	11	BUY
TWELVE	0C	12	CALL
THIRTEEN	0D	13	CAUTION
FOURTEEN	0E	14	CHANGE
FIFTEEN	0F	15	CIRCUIT
SIXTEEN	10	16	CLEAR
SEVENTEEN	11	17	CLOSE
EIGHTEEN	12	18	COMPLETE
NINETEEN	13	19	CONNECT
TWENTY	14	20	CONTINUE
THIRTY	15	21	COPY
FOURTY	16	22	CORRECT
FIFTY	17	23	DATE
SIXTY	18	24	DAY
SEVENTY	19	25	DECREASE
EIGHTY	1A	26	DEPOSIT
NINETY	1B	27	DIAL
HUNDRED	1C	28	DIVIDE
THOUSAND	1D	29	DOOR
MILLION	1E	30	EAST
ZERO	1F	31	(T)ED
A	20	32	(D)ED
B	21	33	(K)ED
C	22	34	(-)ED
D	23	35	EMERGENCY
E	24	36	END
F	25	37	ENTER
G	26	38	ENTRY
H	27	39	ER
I	28	40	EVACUATE
J	29	41	EXIT
K	2A	42	FAIL
L	2B	43	FAILURE
M	2C	44	FARAD
N	2D	45	FAST
O	2E	46	FASTER
P	2F	47	FIFTH
Q	30	48	FIRE
R	31	49	FIRST

ROM SET 1	HEX	DEC	ROM SET 2
CENTI	48	72	MORE
CHECK	49	73	MOVE
COMMA	4A	74	NANO
CONTROL	4B	75	NEED
DANGER	4C	76	NEXT
DEGREE	4D	77	NO
DOLLAR	4E	78	NORMAL
DOWN	4F	79	NORTH
EQUAL	50	80	NOT
ERROR	51	81	NOTICE
FEET	52	82	OHMS
FLOW	53	83	ONWARD
FUEL	54	84	OPEN
GALLON	55	85	OPERATOR
GO	56	86	OR
GRAM	57	87	PASS
GREAT	58	88	PER
GREATER	59	89	PICO
HAVE	5A	90	PLACE
HIGH	5B	91	PRESS
HIGHER	5C	92	PRESSURE
HOUR	5D	93	QUARTER
IN	5E	94	RANGE
INCHES	5F	95	REACH
IS	60	96	RECEIVE
IT	61	97	RECORD
KILO	62	98	REPLACE
LEFT	63	99	REVERSE
LESS	64	100	ROOM
LESSER	65	101	SAVE
LIMIT	66	102	SECURE
LOW	67	103	SELECT
LOWER	68	104	SEND
MARK	69	105	SERVICE
METER	6A	106	SIDE
MILE	6B	107	SLOW
MILI	6C	108	SLOWER
MINUS	6D	109	SMOKE
MINUTE	6E	110	SOUTH
NEAR	6F	111	STATION
NUMBER	70	112	SWITCH
OF	71	113	SYSTEM
OFF	72	114	TEST
ON	73	115	TH
OUT	74	116	THANK
OVER	75	117	THIRD
PARENTHESIS	76	118	THIS
PERCENT	77	119	TOTAL
PLEASE	78	120	TURN
PLUS	79	121	USE

More →

Table 1 continued.

S	32	50	FLOOR	POINT	7A	122	UTH
T	33	51	FORWARD	POUND	7B	123	WAITING
U	34	52	FROM	PULSES	7C	124	WARNING
V	35	53	GAS	RATE	7D	125	WATER
W	36	54	GET	RE	7E	126	WEST
X	37	55	GOING	READY	7F	127	SWITCH
Y	38	56	HALF	RIGHT	80	128	WINDOW
Z	39	57	HELLO	SS	81	129	YES
AGAIN	3A	58	HELP	SECOND	82	130	ZONE
AMPERE	3B	59	HERTZ	SET	83	131	
AND	3C	60	HOLD	SPACE	84	132	
AT	3D	61	INCORRECT	SPEED	85	133	
CANCEL	3E	62	INCREASE	STAR	86	134	
CASE	3F	63	INTRUDER	START	87	135	
CENT	40	64	JUST	STOP	88	136	
400 HZ	41	65	KEY	THAN	89	137	
80 HZ	42	66	LEVEL	THE	8A	138	
20 MS	43	67	LOAD	TIME	8B	139	
40 MS	44	68	LOCK	TRY	8C	140	
80 MS	45	69	MEG	UP	8D	141	
160 MS	46	70	MEGA	VOLT	8E	142	
320 MS	47	71	MICRO	WEIGHT	8F	143	

Listing 1. Have you ever wanted to play bingo but didn't want to be the caller? This program lets the Speak 'N' Sing 2 do the announcing.

```
100 REM BINGO
110 REM SPEECH PORT
120 P0=HEX("8008")
130 P1=P0+1
140 P2=P0+2
150 P3=P0+3
160 REM INITIALIZE SPEECH PORT
170 POKE P1,0
180 POKE P3,0
190 POKE P2,255 : REM B PORT ALL OUTPUTS
200 POKE P1,62  : REM $3E SELECT STANDARD ROM SET
210 POKE P3,46  : REM $2E AUTO STROBE
220 X=PEEK(P0)  : REM CLEAR HANDSHAKE FLAG BIT
230 POKE P2,71  : REM START TOGGLE "PAUSE"
240 REM ************************************************
250 REM * PORT INITIALIZATION COMPLETE, START MAIN PROGRAM *
260 REM ************************************************
270 INPUT "SECONDS PAUSE BETWEEN ANNOUNCEMENTS", K
280 INPUT "ANNOUNCE EACH CLUE TWICE (Y/N)", Q$
290 DIM N(75)
300 REM FILL ARRAY IN NUMBERS 1 TO 75
310 FOR I = 1 TO 75
320   N(I) = I
330 NEXT I
340 REM SCRABBLE ARRAY
350 FOR I = 1 TO 75
360   R = INT(74*RND(0)+1.5)
370   T = N(I)
380   N(I) = N(R)
390   N(R) = T
400 NEXT I
410 REM ************************************************
420 REM * SHUFFLING COMPLETE. ANNOUNCE LETTER ALONG WITH NUMBER *
430 REM ************************************************
440 FOR I = 1 TO 75
450   ON INT((N(I)+14)/15) GOTO 460,470,480,490,500
460   PRINT "B";:RESTORE 920:GOTO 510
470   PRINT "I";:RESTORE 930:GOTO 510
480   PRINT "N"; :RESTORE 940:GOTO 510
490   PRINT "G"; :RESTORE 950:GOTO 510
500   PRINT "O";:RESTORE 960:GOTO 510
510   PRINT N(I),
520   IF I/5 = INT(I/5) THEN PRINT
530   GOSUB 800
540   X9 = N(I)
550   GOSUB 1290
560   REM REPEAT IF REQUESTED
570   IF LEFT$(Q$,1) = "N" THEN 700
580   RESTORE 980
590   GOSUB 800
600   ON INT((N(I)+14)/15) GOTO 610,620,630,640,650
610   RESTORE 920:GOTO 660
620   RESTORE 930:GOTO 660
630   RESTORE 940:GOTO 660
640   RESTORE 950:GOTO 660
650   RESTORE 960:GOTO 660
660   GOSUB 800
670   X9 = N(I)
680   GOSUB 1290
```

More

set contains 144 words, each of which is coded from 0 to 143. The optional ROM set contains 131 words encoded from 0 to 130. This represents a total of 275 different words. However, considering that some words are actually prefixes and suffixes that modify root words, the actual number of different words is far greater.

When a human speaks, the exact way a word is pronounced is a function of many variables. For example, the overall quality of the speech is very much dependent on the pauses between words. So you should exercise care in selecting one of the five possible pause durations. As a rule of thumb, for words beginning with the letters k, t, p, b, d and g, insert an 80 ms pause prior to the words; for words ending in those letters, insert a 40 ms pause following the word.

The standard ROM contains the "ss" sound; this suffix can be used to pluralize many words. Therefore, abort can be easily changed to aborts by appending the "ss" suffix to the root word.

The optional ROM set contains several additional sounds that can be used to modify many words. For example, the "th" suffix can be used to change six, seven and eight to sixth, seventh and eighth. The "uth" sound can be added to words like twenty, thirty and forty to form the adjectives twentieth, thirtieth and fortieth.

Also available in the optional ROM set are four forms of the "ed" suffix, used to change a present-tense word to its past-tense form. Four forms are available because the way we say "ed" varies from word to word. Experimenting with each of the "ed" sounds will let you develop the best-quality results. As a guideline, address 31 "ed" or 32 "ed" should be used with words ending in T or D, such as bat or seed. Address 34 "ed" can be used with words ending with a soft sound, such as ask.

Software

This section will show you how to write a 6800 assembly-language program and a Basic program to reproduce speech. For the sake of our examples, the board memory address is 8008 (hexadecimal), which represents port #2 in most 6800 systems. Because of the FIFO (first in, first out) feature of the Speak 'N' Sing 2, there are two methods of determining if the synthesizer is ready for the next word.

Listing 1 continued.

```
690    REM PAUSE BETWEEN VALUES
700    FOR J = 1 TO K
710      RESTORE 1000
720      GOSUB 800
730    NEXT J
740 NEXT I
750 END
760 REM ************************************************
770 REM * THIS ROUTINE WILL OUTPUT A PHRASE UNTIL A 255 IS *
780 REM * WHICH DENOTES THE END OF THE PHRASE.           *
790 REM ************************************************
800 X4 = PEEK(P1)       : REM CHECK STATUS REGISTER FOR READY
810 IF X4 < 128 THEN 800
820 READ X4
830 IF X4=255 THEN 870     : REM END OF PHRASE
840 X8=PEEK(P0)     : REM CLEAR READY FLAG BIT
850 POKE P2,X4      : REM STORE WORD
860 GOTO 800
870 RETURN
880 REM ************************************************************
890 REM * BEGINNING OF LETTERS, PAUSES AND NUMBERS TO BE SPOKEN *
900 REM ************************************************************
910 REM FIVE LETTERS "R I N G O"
920 DATA 33,255
930 DATA 40,255
940 DATA 45,255
950 DATA 38,255
960 DATA 46,255
970 REM .2 SECOND PAUSE
980 DATA 67,69,67,69,255
990 REM 1 SECOND OF PAUSES
1000 DATA 71,71,71,68,255
1010 REM 1-9
1020 DATA 1,255
1030 DATA 2,255
1040 DATA 3,255
1050 DATA 4,255
1060 DATA 5,255
1070 DATA 6,255
1080 DATA 7,255
1090 DATA 8,255
1100 DATA 9,255
1110 REM TEN - NINETEEN
1120 DATA 10,255
1130 DATA 11,255
1140 DATA 12,255
1150 DATA 13,255
1160 DATA 14,255
1170 DATA 15,255
1180 DATA 16,255
1190 DATA 17,255
1200 DATA 18,255
1210 DATA 19,255
1220 REM TWENTY - SEVENTY
1230 DATA 20,255
1240 DATA 21,255
1250 DATA 22,255
1260 DATA 23,255
1270 DATA 24,255
1280 DATA 25,255
1290 REM SAY NUMBER IN X9
1300 IF X9<10 THEN 1620
1310 IF X9<20 THEN 1480
1320 X5=X9
1330 X9=INT(X9/10)
1340 REM X9 >= 20
1350 ON X9 GOTO 1350,1360,1370,1380,1390,1400,1410,1420,1430
1360 RESTORE 1230:GOTO 1440
1370 RESTORE 1240:GOTO 1440
1380 RESTORE 1250:GOTO 1440
1390 RESTORE 1260:GOTO 1440
1400 RESTORE 1270:GOTO 1440
1410 RESTORE 1280:GOTO 1440
1420 RESTORE 4960:GOTO 1440
1430 RESTORE 4980
1440 GOSUB 800
1450 X9=X5-INT(X5/10)*10
1460 GOTO 1620
1470 REM X9 < 20
1480 ON X9-9 GOTO 1490,1500,1510,1520,1530,1540,1550,1560,1570,1580
1490 RESTORE 1120:GOTO 1590
1500 RESTORE 1130:GOTO 1590
1510 RESTORE 1140:GOTO 1590
1520 RESTORE 1150:GOTO 1590
1530 RESTORE 1160:GOTO 1590
1540 RESTORE 1170:GOTO 1590
1550 RESTORE 1180:GOTO 1590
1560 RESTORE 1190:GOTO 1590
```

More

Listing 1 continued.

```
1570 RESTORE 1200:GOTO 1590
1580 RESTORE 1210
1590 GOSUB 800
1600 GOTO 1730
1610 REM X9 < 10
1620 ON X9+1 GOTO 1730,1630,1640,1650,1660,1670,1680,1690,1700,1710
1630 RESTORE 1020:GOTO 1720
1640 RESTORE 1030:GOTO 1720
1650 RESTORE 1040:GOTO 1720
1660 RESTORE 1050:GOTO 1720
1670 RESTORE 1060:GOTO 1720
1680 RESTORE 1070:GOTO 1720
1690 RESTORE 1080:GOTO 1720
1700 RESTORE 1090:GOTO 1720
1710 RESTORE 1100
1720 GOSUB 800
1730 RETURN
```

Listing 2. This 6800 machine-language program shows how you can reproduce a phrase that uses words in both the standard and optional ROM sets.

```
                        NAM     TALK2ROM
                *****************************************************
                *                                                   *
                * THIS PROGRAM WILL ALLOW THE SPEAK 'N' SING 2 BOARD TO  *
                * SPEAK A PHARSE FROM BOTH THE STAARD AND OPTIONAL WORD  *
                * ROMS. EACH WORD IN THE PHRASE TO BE SPOKEN IS      *
                * REPRESENTED BY 2 BYTES. THE FIRST BYTE DENOTES THE ROM *
                * SET (0 = STANDARD VOCABULARY, 1 = OPTIONAL VOCABULARY).*
                * THE SECOND BYTE INDICATES THE ACTUAL WORD.         *
                * THE END OF THE PHRASE IS NOTED BY A 255.           *
                *                                                   *
                *****************************************************
8008            PORT    EQU     $8008
A003            WARMS   EQU     $A003
                *
A100                    ORG     $A100
A100 CE 80 08   START   LDX     #PORT
A103 6F 01              CLR     1,X
A105 6F 03              CLR     3,X
A107 86 FF              LDA A   #$FF
A109 A7 00              STA A   0,X         A PORT ALL OUTPUTS
A10B A7 02              STA A   2,X         B PORT ALL OUTPUTS
A10D 86 3E              LDA A   #$3F        SELECT STANDARD VOCABULARY
A10F A7 01              STA A   1,X
A111 86 2F              LDA A   #$2F
A113 A7 03              STA A   3,X         SELECT CB2 AUTO STROBE
A115 6D 00              TST     0,X         CLR IRQ FLAGS
A117 6D 02              TST     2,X         CLR IRQ FLAGS
A119 86 43              LDA A   #$43        PAUSE WORD 20 MS.
A11B A7 02              STA A   2,X         SPEAK WORD TO START XSISTIONS
                * MAIN ROUTINE
A11D CE A1 4F           LDX     #PHRASE
A120 A6 00      MORE    LDA A   0,X
A122 81 FF              CMP A   #255        EOT
A124 27 18        -     BEQ     END         PHARSE COMPLETE, RETURN TO FLEX
A126 6D 00      NOTEOT  TST     0,X         1ST ROM SET?
A128 26 07              BNE     NOSTND
A12A C6 3E              LDA B   #$3E
A12C F7 80 09           STA B   PORT+1      SELECT STANDARD ROM SET
A12F 20 05              BRA     CONTIN
A131 C6 36      NOSTND  LDA B   #$36
A133 F7 80 09           STA B   PORT+1      SELECT OPTIONAL ROM SET
A136 08         CONTIN  INX
A137 A6 00              LDA A   0,X
A139 8D 06              BSR     SPEAK
A13B 08                 INX
A13C 20 E2              BRA     MORE
A13E 7F A0 03   END     JMP     WARMS       RETURN TO FLEX
                * ROUTINE TO SPEAK WORD
A141 36         SPEAK   PSH A
A142 B6 80 09   NOTRDY  LDA A   PORT+1      READY FOR NEXT WORD ?
A145 2A FB              BPL     NOTRDY
A147 7D 80 08           TST     PORT        CLEAR INTERRUPT BIT
A14A 32                 PUL A
A14B B7 80 0A           STA A   PORT+2      OUTPUT WORD (AUTO STROBE)
A14E 39                 RTS
                *************************************************
                * ATTENTION, ATTENTION                          *
                * FIRE ON THE FIRST, SECOND, AND THIRD FLOORS.  *
                * GO TO THE NORTH OR SOUTH EXITS.               *
                * THIS IS NOT A TEST!                           *
                *************************************************
A14F 01         PHRASE  FCB     1,8,0,70,1,8,0,71
A150 08 00
A152 46 01
A154 08 00
A156 47
A157 01                 FCB     1,48,0,115,1,49,0,130,0,60,1,117
A158 30 00
```

More

In some cases you'll want to synchronize the word being spoken with some event (such as a word or phrase being written at the computer's monitor). To do this you must bypass the FIFO buffer, by interrogating the CA1 line of the peripheral interface adapter (PIA U1). The schematic in Fig. 5 shows that this line of the PIA is connected directly to the SPC.

However, in most cases you'll want to take advantage of the FIFO by interrogating the status of the CB1 handshake line of the PIA. This line is connected to the SPC via the FIFO buffer. By using the FIFO you can store up to 40 words within a very short time and then proceed with processing other data while the words are spoken.

The buffer can store and reproduce approximately 20 seconds of speech without intervention from the host processor. If more than 40 words need to be stored and the processor must perform other tasks while words are being produced, a large buffer can be developed in the host computer's memory and driven by interrupts. However, in most cases you should find the 40-word speech buffer adequate.

> Synthesizing music is really not all that different from synthesizing sound effects—after all, music is just sound and silence.

An additional comment is worthy of note. The CA1 and CB1 lines of a PIA are triggered by transitions (i.e., high to low or low to high) rather than levels: This means that the PIA will not indicate that the circuit is ready for the first word, even though it actually is, because a transition has not occurred. For this reason a pause ($43) is sent to the speech synthesizer to, in essence, reboot the synthesizer.

A particular ROM set is selected by the CA2 output line of the PIA. If this line is high, the standard ROM set is selected. When CA2 is low, the optional ROM is selected. From a programming standpoint, the standard ROM set is selected by storing a 3E (hex) or 62 (dec) at the A port of the PIA. The optional ROM set is se-

Listing 2 continued.

```
     A15A 73 01
     A15C 31 00
     A15E 82 00
     A160 3C 01
     A162 75
64   A163 01                  FCB    1,50,0,129,0,71
     A164 32 00
     A166 81 00
     A168 47
65   A169 00                  FCB    0,86,0,2,0,138,1,79,1,86,1,110,1,41,0,129
     A16A 56 00
     A16C 02 00
     A16E 8A 01
     A170 4F 01
     A172 56 01
     A174 6E 01
     A176 29 00
     A178 81
66   A179 00                  FCB    0,71,1,118,0,96,1,80,0,32,1,114
     A17A 47 01
     A17C 76 00
     A17E 60 01
     A180 50 00
     A182 20 01
     A184 72
67   A185 FF                  FCB    255
68                  *
69                           END    START
```

Listing 3. This program simulates the sound of a plane as it passes overhead. The effect is accomplished by amplitude modulating noise, and the noise is synthesized by using a random number generator.

```
1                            NAM    PLANE
2                   ************************************************************
3                   *                                                          *
4                   * THIS SOUND EFFECTS PROGRAM SYNTHESIZES THE SOUND OF A    *
5                   * PLANE BY CONTROLLING THE AMPLITUDE OF WHITE NOISE.       *
```

More

lected by storing 36 (hex) or 54 (dec).

Listing 1 shows a Basic program that plays the game of bingo. The synthesizer announces each bingo tile that is picked at random. In this way, everyone can play the game and no one is stuck announcing. The program lets the players determine the length of time between announcements. You can also tell the synthesizer to speak each call twice.

As each call is spoken, it appears at the monitor to allow verification after the game is over. Note that this is an example of a case in which you don't want to use the buffer—you want the call to appear at the monitor as it is spoken. For this reason, the CA1 handshake line is used.

Listing 2 shows a 6800 machine-language program. Unlike the bingo program, which uses only the standard ROM set, this program produces a phrase that comprises words from both the standard as well as the optional ROM set. In order to accomplish this, each word is represented by two bytes. The first byte indicates which ROM contains the word, while the second represents the word code. If the first byte is a 0, the stan-

dard ROM set is selected; if it's a 1, the optional ROM set is selected. After the proper ROM is selected, the word code is stored.

Sound Effects and Music

Space does not allow room for an in-depth discussion into the fascinating field of music and sound effect synthesis. However, the example program and the following discussion should provide a good introduction.

The Plane program shown in Listing 3 produces nothing more than amplitude modulated white noise. The heart of the noise is a random number generator. The amplitude of this noise is determined by an envelope waveshape given in a table. As the plane approaches the listener, the noise amplitude is increased. As the plane passes overhead, the noise reaches its maximum loudness, and slowly decreases as the plane moves into the sunset. Noise is an important sound effect that can be used to simulate thunderstorms and gunshots. Bear in mind that creativity is the key to good sound effect programming.

Synthesizing music is really not all

Listing 3 continued.

```
                        * CONTROL RETURNS TO FLEX AT END OF PROGRAM.          *
                        ***********************************************************
8008                    PORT    EQU     $8008       SS-1 PORT
AD03                    WARMS   EQU     $AD03
002F                    FREQ    EQU     $2F         TIME BETWEEN NOISE PULSES
00FE                    DURA    EQU     $FE         NUMBER OF RANDOM PULSES
                        *
A100                            ORG     $A100
A100 7E A1 05           START   JMP     START1
A103                    HIB     RMB     1           HIB OF RNG
A104                    LOB     RMB     1           LOB OF RNG
A105 CE 80 08           START1  LDX     #PORT
A108 6F 01                      CLR     1,X
A10A 86 FF                      LDA A   #$FF
A10C A7 00                      STA A   0,X         A PORT ALL OUTPUTS
A10E 86 04                      LDA A   #4
A110 A7 01                      STA A   1,X
A112 CE A1 53                   LDX     #ENVELP     AMPLITUDE MOD. POINTER
A115 A6 00              AGN2    LDA A   0,X         END OF ENVELP ?
A117 81 04                      CMP A   #4
A119 26 03                      BNE     NOTEND
A11B 7E AD 03                   JMP     WARMS       RETURN TO FLEX
A11E 08                 NOTEND  INX
A11F C6 FE                      LDA B   #DURA       OUTPUT N RANDOM VALUES
A121 86 2F              AGN1    LDA A   #FREQ       DELAY
A123 4A                 AGN3    DEC A
A124 26 FD                      BNE     AGN3
A126 BD 0A                      BSR     RANDOM      GET RANDOM NUMBER IN ACC A
A128 A4 00                      AND A   0,X
A12A B7 80 08                   STA A   PORT        OUTPUT TO SS-1 D/A CONVERTER
A12D 5A                         DEC B
A12E 26 F1                      BNE     AGN1
A130 20 E3                      BRA     AGN2        GET NEXT AMPLITUDE VALUE
                        * RANDOM NUMBER GENERATOR.  RESULT IN ACC A
A132 B6 A1 03           RANDOM  LDA A   HIB
A135 46                         ROR A
A136 B8 A1 03                   EOR A   HIB
A139 46                         ROR A
A13A 46                         ROR A
A13B B8 A1 03                   EOR A   HIB
A13E 46                         ROR A
A13F B8 A1 04                   EOR A   LOB
A142 46                         ROR A
A143 46                         ROR A
```

More

Listing 3 continued.

```
 50  A144 84 02          AND A  #2
 51  A146 78 A1 04       ASL    LOR
 52  A149 79 A1 03       ROL    HIR
 53  A14C 8B A1 04       ADD A  LOR
 54  A14F 87 A1 04       STA A  LOR
 55  A152 39             RTS
 56                  * THE FOLLOWING TABLE REPRESENTS THE AMPLITUDE MODULATION
 57                  * ENVELOP OF THE RANDOM NOISE.
 58  A153 00         ENVELP  FCB    $0,$11,$18,$20,$28,$30,$38,$40,$48
     A154 11 18
     A156 20 28
     A158 30 38
     A15A 40 48
 59  A15C 50             FCB    $50,$58,$60,$68,$70,$78,$80,$88,$90,$98
     A15D 58 60
     A15F 68 70
     A161 78 80
     A163 88 90
     A165 98
 60  A166 A0             FCB    $A0,$A8,$B0,$B8,$C0,$C8,$D0,$D8,$E0,$E8
     A167 A8 B0
     A169 B8 C0
     A16B C8 D0
     A16D D8 E0
     A16F E8
 61  A170 F0             FCB    $F0,$F8,$FF,$FF,$FF,$FF
     A171 F8 FF
     A173 FF FF
     A175 FF
 62  A176 F8             FCB    $F8,$F0,$F8,$F0,$D8,$D0,$C8,$C0
     A177 F0 E8
     A179 E0 D8
     A17B D0 C8
     A17D C0
 63  A17E B8             FCB    $B8,$B0,$A8,$A0,$98,$90,$88,$80
     A17F B0 A8
     A181 A0 98
     A183 90 88
     A185 80
 64  A186 78             FCB    $78,$70,$68,$60,$58,$50,$48,$40
     A187 70 68
     A189 60 58
     A18B 50 48
     A18D 40
 65  A18E 38             FCB    $38,$38,$30,$30,$28,$28,$20,$20
     A18F 38 30
     A191 30 28
     A193 28 20
     A195 20
 66  A196 20             FCB    $20,$20,$20,$18,$18,$18,$18,$18
     A197 20 20
     A199 18 18
     A19B 18 18
     A19D 18
 67  A19E 10             FCB    $10,$10,$10,$10,$10,$10,$10,$10
     A19F 10 10
     A1A1 10 10
     A1A3 10 10
     A1A5 10
 68  A1A6 10             FCB    $10,$10,$10,$10,$10,$10,$10,$10
     A1A7 10 10
     A1A9 10 10
     A1AB 10 10
     A1AD 10
 69  A1AE 08             FCB    8,8,8,8,8,8,8,8
     A1AF 08 08
     A1B1 08 08
     A1B3 08 08
     A1B5 08
 70  A1B6 08             FCB    8,8,8,8,8,8,8,8
     A1B7 08 08
     A1B9 08 08
     A1BB 08 08
     A1BD 08
 71  A1BE 04             FCB    4          END OF TRANSMISSION
 72                  *
 73                      END    START
```

that different from synthesizing sound effects—after all, music is just sound and silence. A special program called a music interpreter allows the user to write music easily, using music notation. For example, if you wanted to reproduce the half-note C sharp in the first octave, you'd just write C#1H, rather than tediously write a machine-language routine specifying various parameters. If you are particularly interested in music, write to me for more information in this field.

Hardware

The Speak 'N' Sing 2 speech, music and sound effects synthesizer can be thought of as consisting of two separate circuits, a digital-to-analog converter driven by the A port of the PIA, U1, and a speech synthesizer

(U3) driven by the B port of the PIA. (See Fig. 4.)

The digital-to-analog converter is made up of a resistor ladder consisting of 16 resistors neatly packaged in a DIP and labelled RL1 in the schematic (Fig. 5). The output of the converter ranges from approximately 0 to 5 volts. Since the converter is connected to eight lines of the PIA, it is an eight-bit converter which allows 256 different voltages in the specified range. The D/A circuit is capable of reproducing virtually any sound, including music, sound effects and waveform speech.

The speech portion of the circuit consists of basically four parts: the

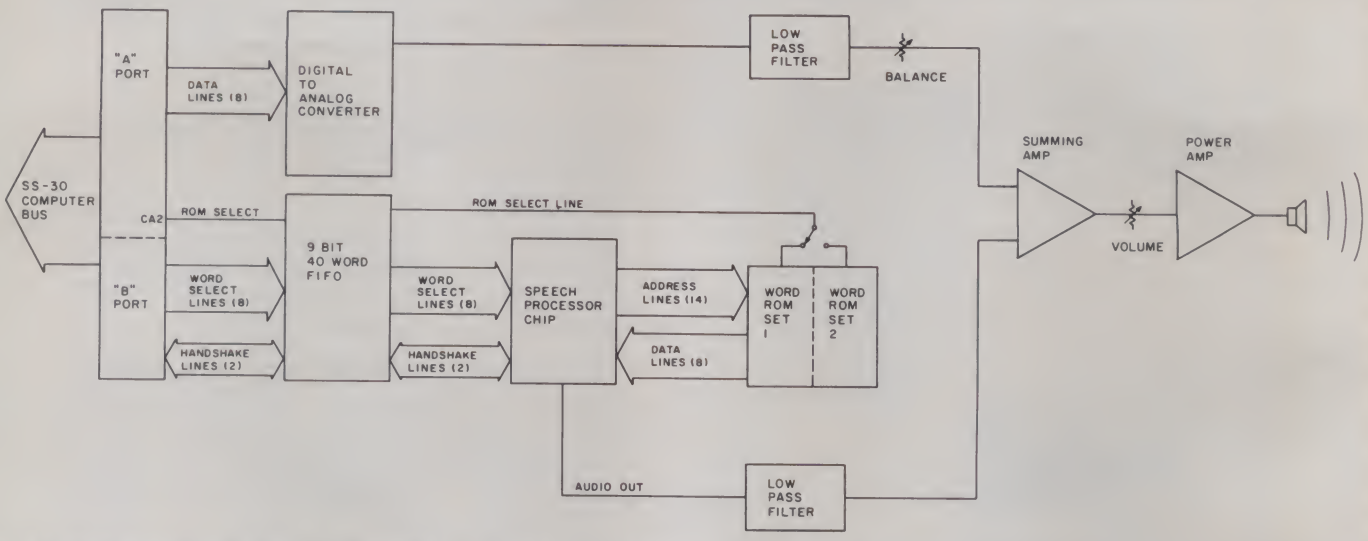


Fig. 4. Block diagram of the Speak 'N' Sing 2. The two major divisions of the synthesizer are the D/A converter, which allows music and sound effect synthesis, and the speech processor, which in conjunction with the word ROMs is responsible for speech synthesis.

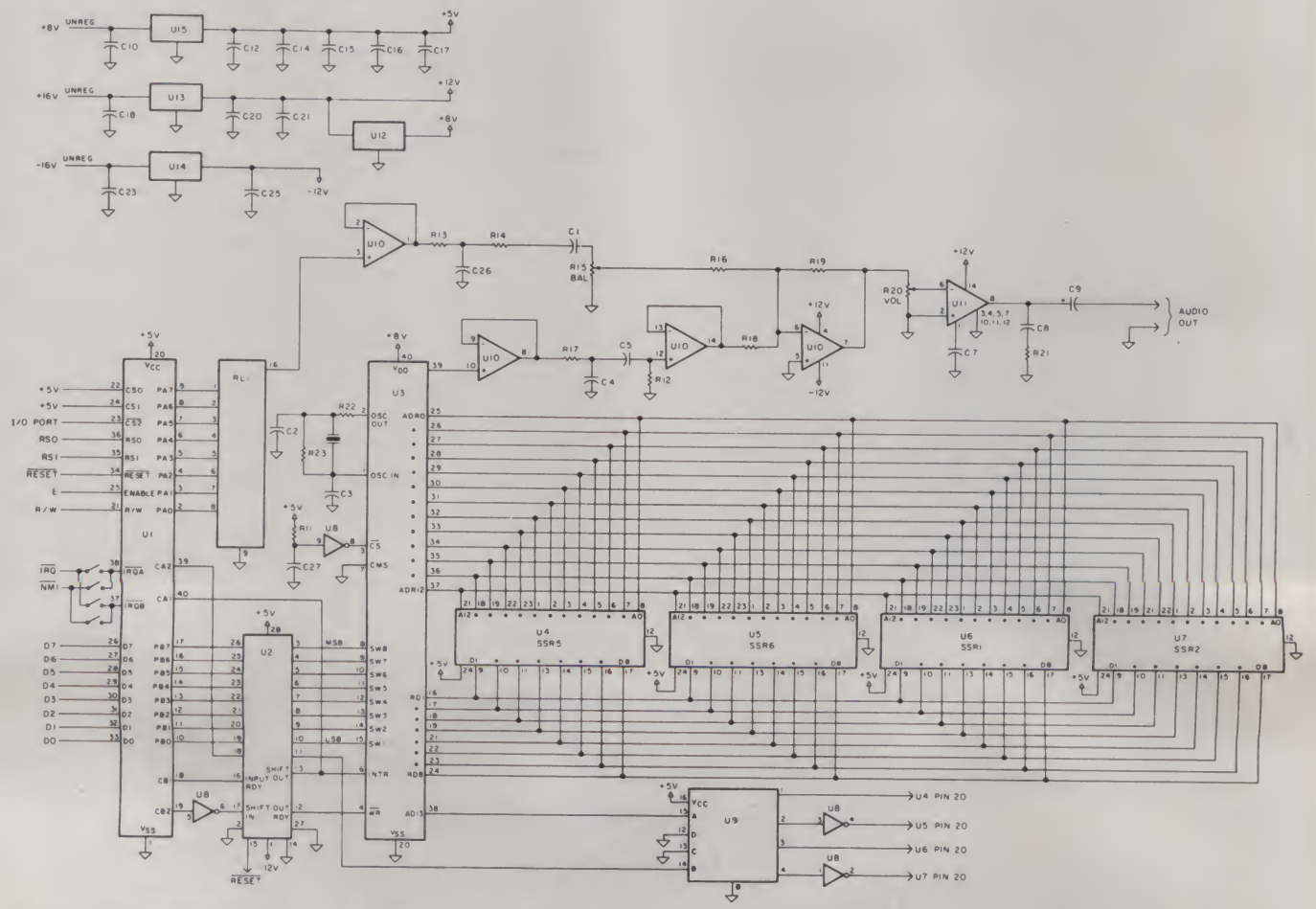


Fig. 5. Complete schematic of the Speak 'N' Sing 2 speech, music and sound effects synthesizer. Integrated circuit U3 is the heart of the speech synthesizer. Words are stored in ROMs U4,5,6,7. The eight-bit D/A converter, responsible for music and speech synthesis, comprises resistor ladder network RL1.

		INTEGRATED CIRCUITS	
1	1	U1	6821 PIA
2	1	U2	33512 40 WORD 9 BIT FIFO
3	1	U3 4 5	NATIONAL SEMI. DT-1050 3 CHIP SET
4	1	U6 7	NATIONAL SEMI. DT-1057 2 CHIP SET
5	1	U8	74LS04 HEX INVERTER
6	1	U9	74LS42 4 LINE TO 10 LINE DECODER
7	1	U10	LM 324 QUAD OP AMP
8	1	U11	LM 380 2 WATT AUDIO POWER AMP
9	1	U12	78L08 8 VOLT POS. REGULATOR
10	1	U13	7812 12 VOLT POS. REGULATOR
11	1	U14	79L12 12 NEG. REGULATOR
12	1	U15	7805 5 VOLT POS. REGULATOR
		RESISTORS	
13	1	RL1	RES. LADDER ALLEN BRADLEY 316L08253
14	4	R11 16 18 19	10K 1/4 WATT
15	1	R17	6.8K 1/4 WATT
16	2	R12 14	100K 1/4 WATT
17	2	R15 20	5K POT BAL & VOL
18	2	R13 22	1.5K 1/4 WATT
19	1	R21	2.7 1/4 WATT
20	1	R23	1 M 1/4 WATT
		CAPACITORS	
21	1	C1	1 MFD
22	1	C2	47 PFD
23	1	C3	15 PFD
24	10	C4 5 7 8 14 15 16 17 25 26	.1 MFD
25	1	C9	220 MFD
26	2	C10 18	100 MFD
27	5	C12 20 21 23 27	22 MFD
		MISCELLANEOUS	
28	1		4 MHZ CRYSTAL
29	1		P.C. BOARD
30	2		FEMALE MOLEX 15 SOCKET CONN.
31	1		POLARIZING KEY
32	1		RCA FEMALE PHONO CONN
33	1		RCA MALE PHONE CONN
34	3		14 PIN IC SOCKET
35	2		16 PIN IC SOCKET
36	4		24 PIN IC SOCKET
37	1		28 PIN IC SOCKET
38	2		40 PIN IC SOCKET
39	1		4 POLE DIP SW.
40	2		4 X 40 1/4" SCREW
41	2		4 X 40 NUT
42	1		HEAT SINK

Parts list. Complete parts list for the Speak 'N' Sing 2. The board is available in kit form as well as tested and assembled.

FIFO buffer (U2), the speech processor chip (U3), the word ROMs (U4 5 6 7) and the driver (PIA, U1). Note that the FIFO is a nine-bit device. Eight of the bits are used for the word code, while the remaining bit determines which ROM is selected.

The heart of the speech synthesizer is the speech processor chip (U3). It can reproduce any word in the ROMs. The input to the SPC consists of eight lines, which allows access to a maximum of 256 words. However, codes above 143 for U4 and 5, and codes above 130 for U6 and 7, will give unpredictable results. The CA2 output line of the PIA is used to select a particular pair of ROMs. If CA2 is high, U4 and 5 will be selected; ROMs U6 and 7 will be selected if CA2 is low.

The remaining circuitry, consisting mainly of U10 and U11, is responsible for audio summing and amplification. U10 is connected as a summing amplifier to sum the output signal of the D/A converter (i.e., music or sound effects) and the speech synthesizer. These signals are then amplified by U11, which is an audio power amplifier.

Two control potentiometers are conveniently placed at the edge of the pc board to allow the user to control the volume of music and speech. Note that the potentiometer marked BAL controls only the volume of the music.■

Price List

The following items are available postpaid in the U.S. Overseas orders add $12 postage.

All Speak 'N' Sing 2 prices include the standard ROM set only; the optional ROM set must be purchased separately.

Speech Systems
38 W 255 Deerpath Road
Batavia, IL 60510

Speak 'N' Sing 2, blank printed circuit board, $39.

Speak 'N' Sing 2, semi-kit (board, U3 U4 U5, crystal, RL1), $129.

Speak 'N' Sing 2, kit form (all components), $189.

Speak 'N' Sing 2, assembled and tested, $229.

Optional ROM set, 131 additional words, $49.

Software including speech-based games, music and sound effects in Flex 1.0, 2.0 or Flex 9.0 is available on five-inch or eight-inch disks, $29.

Illinois residents please add 5¼ percent sales tax. Visa and Master Card orders also accepted. Allow ten days for checks to clear.