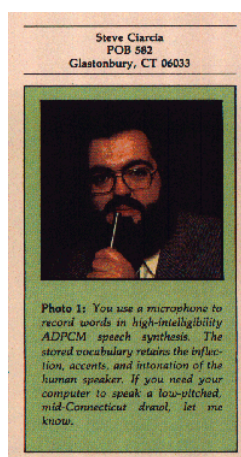


---

## ADPCM for Highly Intelligible Speech Synthesis



Copyright © 1983 Steven A. Ciarcia. All rights reserved, Juni 1983 @ BYTE Publication Inc

---

posted 1 February 2002

Steve Ciarcia  
POB 582  
Glastonbury1 CT 06033

Special thanks to Bill Curlew for his software expertise.

Some new integrated circuits from Oki Semiconductor compress digitized speech data efficiently.

### Use ADFCM for Highly Intelligible Speech Synthesis

- During the past few years I have presented four different computer speech-synthesizer projects (see references). With each article I have tried to present the latest technology and describe successively more cost-effective synthesis methods. This month I'd like to describe a new variation on digitized speech that uses adaptive differential pulsecode modulation.

### What Is Digitized Speech?

- Computers communicate in a digital language, but the language of humans is analog. If computers are to speak as we do, this obvious barrier must be overcome. Fortunately for us, a number of techniques have been devised to allow a computer to synthesize a human voice, some of them quite effective. Some synthesized voices employ electronic circuitry to simulate the throat and vocal tract, but the purest form of machine-generated speech is simply a digital recording of an actual human voice, using digital circuitry to mimic the action of a tape recorder. For example, in most parts of the United States you can dial a telephone number and hear a recorded voice saying something like, "The number you have reached has been changed. The new number is 924-9281." The voice is distinctly human in quality, highly intelligible, and machine-generated an excellent example of digitized speech. Although it uses a lot of memory, digitized speech is the most intelligible machine-generated speech currently possible. The basic concepts of producing stored digital speech are fairly simple. The process begins with data acquisition. A voice waveform can be treated like any other fluctuating voltage input; the computer can record the waveform by periodically taking a sample of the signal's voltage through an analog-to-digital (A/D) converter and storing it as a binary value. (The number of samples needed per second depends upon the frequency of the input signal.) Once the samples have been stored, the computer can recreate the original waveform by sequentially sending the stored values to a digital-to-analog (D/A) converter at the same rate as the original sampling.

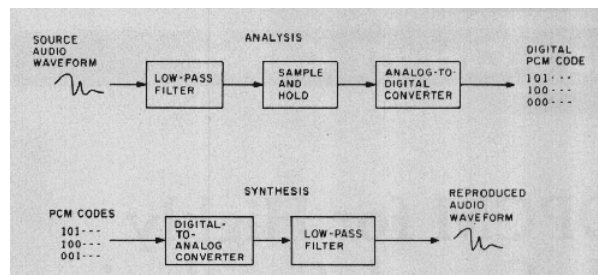
### Pulse-Code Modulation

- A common method of representing continuous analog values in digital form is pulse-code modulation, or PCM. In PCM, distinct binary representations (pulse codes) are chosen for a finite number of points along the continuum of possible states. When-ever the value is being measured and it falls between two encoded points, the code for the closer point is used.

(This process is called quantization: the dividing of the range of values of a wave into subranges, each of which is represented by an assigned value.) A series of these pulse codes can be transmitted in a pulse train, resulting in a pulse-code modulated signal. Because the samples of digitized speech referred to above are stored in the form of digital pulses, the stored speech waveform can be thought of as an example of pulse-code modulation. Figure 1 shows a block diagram of a speech synthesizer that reproduces speech stored in pulse-code-modulated form.

Fig.1

Functional Block Diagram of a digitized speech-reproduction system that employs pulse-code modulation



### Sampling Rates and Other Messy Stuff

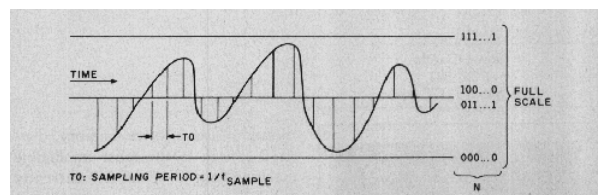
- The sampling rate you use in recording any signal must be chosen with awareness of a theoretical limit called the Nyquist interval. At the very minimum, the sampling rate must be at least twice the highest frequency found in the input signal. With an input bandwidth of 2 kHz (kilohertz), adequate for intelligible speech, the sampling frequency would have to be at least 4 kHz. This rule holds strictly true only when an ideal lowpass filter is used on the output of the D/A converter.

The ear is sensitive, and too coarse a reproduction will sound unnatural or even unintelligible.

- In real equipment, sampling rates of 3 or 4 times the input bandwidth are sometimes necessary. So for speech reproduction, a sampling rate around 6 or 8 kHz is good. (Optical digitized-music recordings, which are just now coming to market, use 16-bit A/D conversion at a 50-kHz sample rate to achieve high fidelity. The resulting data rate is 800,000 bps-bits per second.) Other technical limitations crop up. Once you have determined the sampling rate, you must consider the resolution of the analog-to-digital converter. A/D converters operate in discrete steps (quanta) rather than continuous levels, as shown in figure 2.

Fig. 2

Waveform sampling by pulse-code modulation (PCM). The interval between samples is  $T_0$ ; the sampling frequency is the reciprocal of the interval. Each sample of PCM data consists of  $N$  bits; the leftmost is the most significant bit and the rightmost is the least significant bit.



- If a 4-bit A/D converter is used, then only 16 values are available to define the signal. Any reading could potentially be in error by ca.  $1/16$ , or about 6 percent. A 12-bit converter, which has 4096 potential levels, would have a possible quantization error of only 0.02 percent.

### Achieving Fidelity

- In dealing with analog voice signals, we must accurately reproduce a the input signal for it to be understood. The ear is sensitive, and too coarse a reproduction will sound unnatural or even unintelligible. A direct relationship exists between the PCM data rate and reproduced speech quality. Let's consider a case in which we have an 8-kHz sampling rate. If we use 12-bit A/D conversion, then the data rate (in bits per second) is found using the following equation:

- bit rate = sample rate X conversion bits  
 - 8000 Hz X 12 bits  
 - 96,000 bits/second

- Using standard PCM on a voice signal with a 4-kHz bandwidth would require a 96,000-bps data rate. The average personal computer could store only about 8 seconds of speech in its 64K-byte memory. The data rate can be reduced somewhat by using an 8-bit A/D converter rather than a 12-bit unit. The raw data rate now becomes 8000 X 8 or 64,000 bps. (This reduces the signal-to-noise ratio from 66 to 42 dB (decibels), but the sound quality is more than adequate for experimentation. For commercial applications, however, I recommend a 12-bit converter.)

### Delta Modulation

- The pulse-code modulation we have been examining uses no data compression. In playback, the data bits representing the absolute values of each successive signal sample are sent to a full-resolution D/A converter and reproduced at the same rate at which they were recorded: 96,000 bps in, 96,000 bps out. The circuit can operate with no assumptions made about the signal it is to process. On the other hand, voice waveforms contain much redundant data. Long periods of silence are interspersed with sounds that vary in pitch slowly. If you take some time to analyze the A/D samples, you will notice that the changes are, for the most part, gradual and that the variations in the signal between adjacent samples are a limited portion of the full dynamic range. One method of reducing the data rate used in PCM voice reproduction is called delta modulation. This process assumes that the input signal's waveform has a fairly uniform and predictable slope (rate of rising and falling). Rather than storing an 8- or 12-bit quantity for each sample, a delta modulator stores only a single bit. When the computer samples the input signal from the A/D converter, it compares

the current reading to the preceding sample. If amplitude of the new sample is greater, then the computer stores a bit value of 1. Conversely, if the new sample is less, then a 0 will be stored. Figure3 shows how this works.

Fig.3a

Waveform sampling by delta modulation. Each sample of the source waveform is tested to see if its amplitude is higher or lower (within the resolution of a fixed quantization value  $\Delta r$ ) than that of the previous sample. If the amplitude is higher, the single-bit delta-modulated encoding value is set to 1; if lower, the encoding value is set to 0.

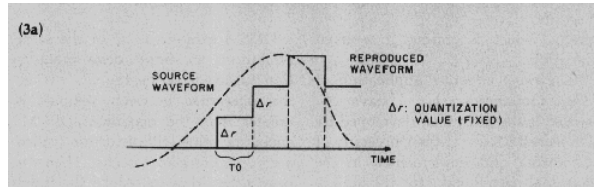
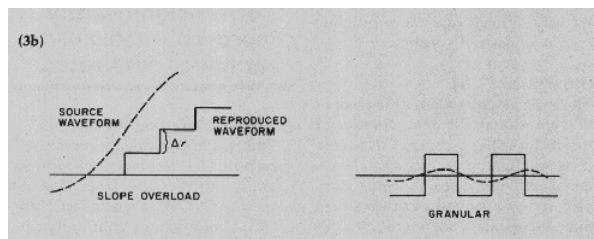


Fig. 3b

Two potential problems occurring in delta modulation. When the source waveform changes too rapidly, the fixed quantization value may be too small to express the full change in the input; this slope overload causes a compliance error. Or when there is little change in the input waveform (at the extreme, a DC signal), vertical deflection in the quantization value results in granular noise in the output.



- Reproduction of the waveform is accomplished by sending the stored bits in sequence to the output, where their values are integrated. But, like other techniques, delta modulation has limitations, one of them the familiar sampling-rate restriction. Because only a single bit changes between samples, the rate at which samples are taken must be sufficiently fast that no significant information is lost from the input signal. Furthermore, if the slope of the input waveform varies a lot, the reproduced waveform may be audibly distorted. So using delta modulation may not reduce the data rate much, although there are many different variant schemes, and it's difficult to predict which is optimal in a given situation.

### Competing Digitizing Methods

**Competing Digitizing Methods**

*By the time you read this, you may have heard the results of a standardization proceeding that put two of the speech-digitizing methods discussed here in competition. (See reference 7.)*

*American Telephone and Telegraph (AT & T), in the throes of beginning its divestiture of its 22 local Bell operating companies (BOCs), submitted a proposal to the CCITT (Comité Consultatif International Téléphonique et Télégraphique) to standardize a certain form of adaptive differential pulse-code modulation as the worldwide preferred method of digitizing voice telephone signals for long-distance transmission. The proposed CCITT scheme uses a 4-bit sampling size with an 8-kHz sampling rate, for a 32,000-bps overall data rate. This would be a change from the two different 64,000-bps digitizing systems now in use.*

*A competing digitizing scheme, developed by Satellite Business Systems (SBS—a company jointly owned by IBM, Comsat, and Aetna Life & Casualty), employs delta modulation: 1-bit samples taken at a rate of 32 kHz, arriving at the same 32,000-bps data rate by a completely different route.*

*Standards published by the CCITT are called "recommendations," but they are quite strictly followed in most regions of the globe.*

*Telecommunication experts outside of AT & T have expressed concerns that the local BOCs and the independent long-distance common carriers will face both financial and technical problems in upgrading equipment to interface with the proposed CCITT 32,000-bps ADPCM circuits. At this writing, it is not known when AT & T will begin installing the new system. . . . R. S. S.*

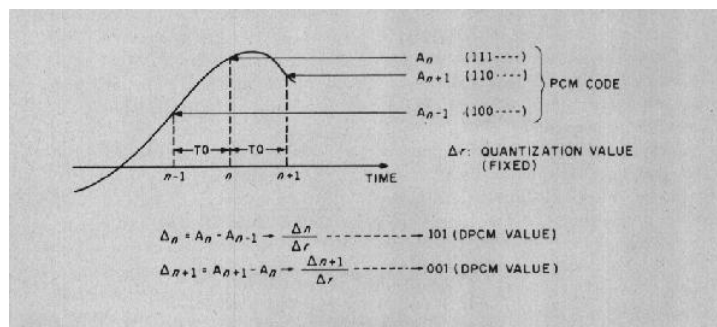
- The most effective application of delta modulation that I have observed is the technique developed by Dr. Forest Moser at the University of California and implemented in the National Semiconductor Digitaler voice-synthesis chip set (see reference 1). However, while the Digitaler's process is definitely a variant of delta modulation, the data-compression and zero-phase-encoding algorithms that produce the stored bit patterns take hours of processing per word; it's very difficult for you to program your own custom vocabulary.

#### Differential PCM

- We can actually reduce the amount of data stored for reproduction of speech by using a concept related to delta modulation as follows. When the speech waveform is being sampled, for each sample a value is stored that represents the amplitude difference between samples. This scheme, called differential pulse-code modulation, or DPCM, allows more that a single bit of difference between stored samples, accommodating more variation in the input waveform before severe distortion sets in. The DPCM value can be expressed as a fraction of the allowed input range or the absolute difference between samples (see figure 4).

Fig.4

Differential pulse-code modulation (DPCM) is an attempt to reduce the amount of data stored or transmitted, as compared with regular PCM. For each sample, the difference between the previous PCM code and the current code is expressed in terms of a fixed quantization value  $\Delta r$  (delta-r), which must be chosen with attention to the characteristics of the source waveform. If too large or small a quantization value is used, compliance errors occur.



- DPCM exhibits some of the same limitations as simple delta modulation but to a lesser degree. Only when the difference between samples is greater than the maximum DPCM encoding value will distortion (called a compliance error) occur. Then the only solution is to reduce the input bandwidth or raise the sampling frequency.

ADPCM Is a specialized form of PCM that offers significantly Improved intelligibility at lower data rates.

#### Adaptive Differential PCM

- The real breakthrough in digitized speech is the technique known as adaptive differential pulse-code modulation (ADPCM), a specialized form of PCM that offers significantly improved intelligibility at lower data rates. This system was devised to overcome the defects of the delta-modulation techniques described thus far while still reducing the overall data rate and improving the output's compliance with the source waveform. ADPCM improves upon DPCM by dynamically varying the quantization between samples depending upon their rate of change while maintaining a low bit rate, condensing 12-bit PCM samples into only 3 or 4 bits. (The variations in the quantization value are regulated with regard to the characteristic complex sine waves that occur in voice. The technique is therefore not applicable to other kinds of signals, such as square waves.)
- In ADPCM, each sample's encoding is derived by a complicated procedure that includes the following steps: a PCM-value differential  $\Delta n$  is obtained by subtracting the previous PCM-code value from the current value; the quantization value  $\Delta n$  (delta-n) is obtained by multiplying the previous quantization value times a coefficient times the absolute value of the previous PCM-code value; the PCM-value differential is then expressed in terms of the quantization value and encoded in four bits, as shown in figure 5.

#### Build an ADPCM Speech Analyzer/Synthesizer

- The Oki Semiconductor Corporation produces a number of integrated circuits (ICs) that perform ADPCM encoding and decoding. Of these, the MSM5218RS and the MSM5205RS are worthy of attention. The 5218 is designed to perform both storing and reproducing of digitized speech, while the 5205 provides only the reproducing function. Using these CMOS (complementary metal-oxide semiconductor) components, we can put together a cost-effective speech-synthesis system that produces highly intelligible output and yet makes efficient use of memory.
- Figure 6 on page 40 is the block diagram of the MSM5218RS IC. It is designed to work with 12-bit analog-to-digital converters and contains both an ADPCM analyzer and synthesizer. An internal 10-bit D/A converter is provided to reconstruct the waveform where direct analog output is wanted, or the decoded PCM data may be routed to an external D/A converter. The schematic in figure 7 on pages 42 and 43 diagrams a speech-synthesis circuit built around this chip (see photo 2 on page 41). In the

circuit, a low-cost 8-bit A/D converter is used in place of a higher-resolution, more costly 12-bit converter. The Oki MSM5204RS 8-bit CMOS A/D converter, employed here, uses a successive capacitor ladder conversion system. It also incorporates a sample-and-hold stage that enables direct input of rapidly changing analog signals. An external clock signal provides timing for the chip; the clock's frequency is not critical and can be anywhere from 450 to 500 kHz.

Fig.5

Adaptive differential pulse-code modulation (ADPCM) improves upon DPCM by dynamically varying the quantization between samples, depending upon their rate of change, while maintaining a low bit rate, condensing 12-bit PCM samples into only 3 or 4 bits. In ADPCM, each sample's encoding is derived by a procedure that includes the following steps. A PCM-value differential  $d_n$  is obtained by subtracting the previous PCM-code value from the current value. The quantization value  $\Delta_n$  (delta-n) is obtained by multiplying the previous quantization value times a coefficient times the absolute value of the previous PCM-code value. The PCM-value differential is then expressed in terms of the quantization value and encoded in four bits. The mathematical relations are shown here in figure 5a, whereas figure 5b shows a typical encoded waveform.

Fig.5a

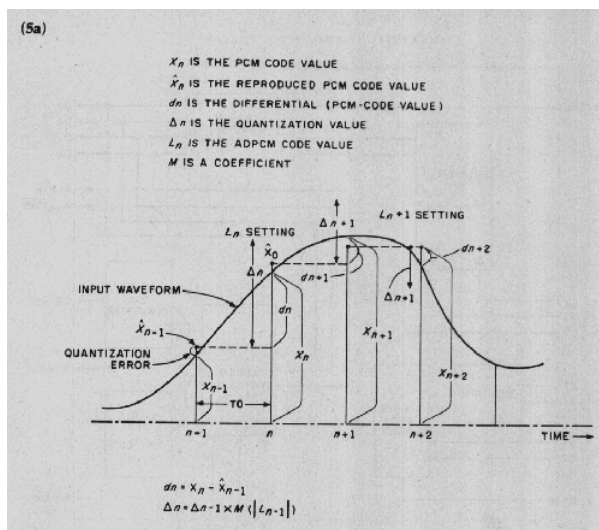
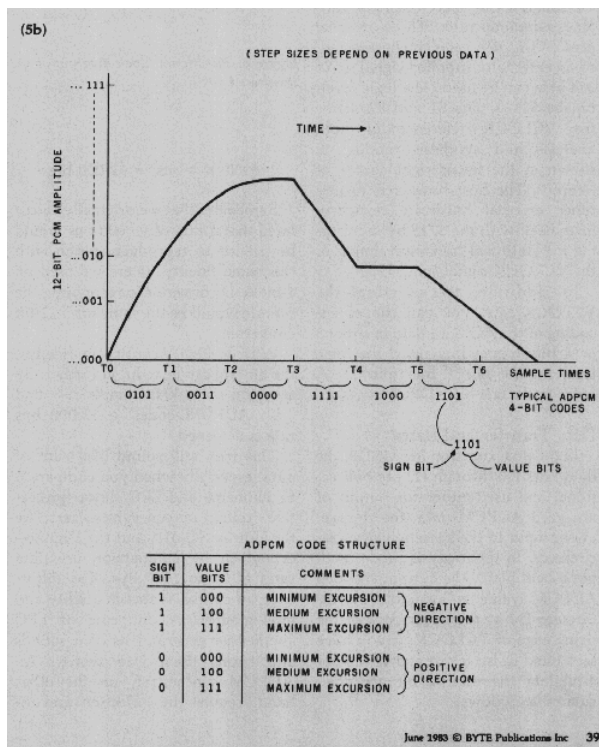


Fig.5b



- The frequency bandwidth of the signal input to the A/D converter is limited by an active low-pass filter, IC2, an Oki ALP-2 filter with a 1.7-kHz cutoff frequency. Attenuation is 18 dB per octave above the cut-off frequency. (Although frequencies up to 4 kHz can theoretically be captured with an 8-kHz sample rate, in this application the lower cutoff frequency gives better-sounding reproduction.)

#### A/D Conversion in Operation

- Data conversion is started when the S CON (start conversion) line (pin 13) of the MSM5218 forces the write line (WR), pin 15) on the 5204 A/D converter into a lower state. After conversion is complete, the A/D read line (RD), pin 14) is brought low to latch the data onto the 5204's output lines. At a clock rate of 450 kHz, the 5204 completes the 8-bit conversion in approximately 73 microseconds. The digital representation of the input data from the 5204 is fed into a CD4014 serial-to-parallel converter (IC9) for transposition into the serial format required by the MSM5218's input. Because we are using an 8-bit converter and the MSM5218 expects 12-bit input, the four remaining low-order bits are clocked in as zeros by the CD4024 counter (IC7) and sections of the quad NAND gate (IC6). These components provide four extra SI-CK (serial input OR clock) pulses with zero-logic-level data.

#### Selectable Parameters

- The MSM5218 can analyze or synthesize ADPCM speech using a variable sampling rate. Three internal preset VCLOCK rates can be selected, or an externally supplied signal up to 384 kHz can be used. The logic levels on the 5218's pins S1 and S2 define the VCLOCK reference in both analysis and synthesis modes, as shown in the lower right corner of figure 7. The host computer, or any other external hardware, synchronizes itself with the 5218 by monitoring the state and transition timing of the VCLOCK signal (pin 1). In addition to selecting the VCLOCK rate, you can choose encoding of the ADPCM data in either 3 or 4 bits, depending upon the logic level on the 4B/3B line (pin 7). A logic 1 selects 4-bit ADPCM values.

#### Data Transfer and Rates

- In the dual-function MSM5218, the data lines D0 through D3 are bidirectional and used either for output of analyzed ADPCM data (for storage) or for input to the speech-synthesizer circuitry. In the analysis mode (with pin 6 held high), the current encoded ADPCM value is available on D0 through D3 at the occurrence of the rising edge of VCLOCK. If you have set S1 and S2 for 8 kHz and 4B/3B for 4-bit data, the resulting bit rate is calculated as follows:  $8000 \times 4 \text{ bits} = 32,000 \text{ bps}$

### MSM5218 INTERNAL BLOCK DIAGRAM

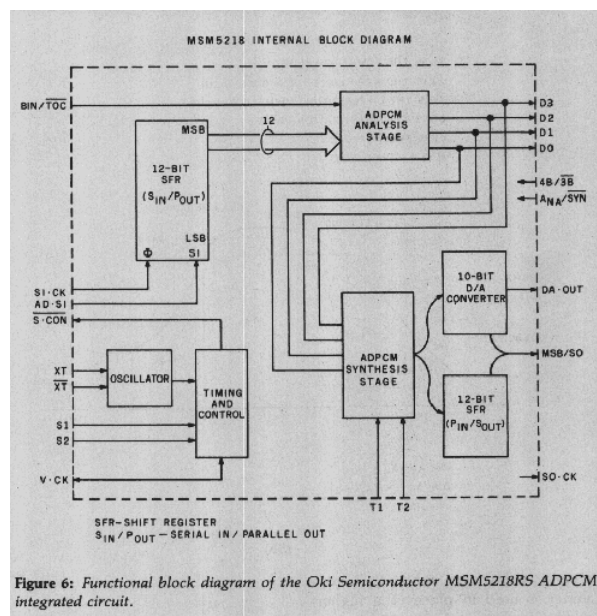
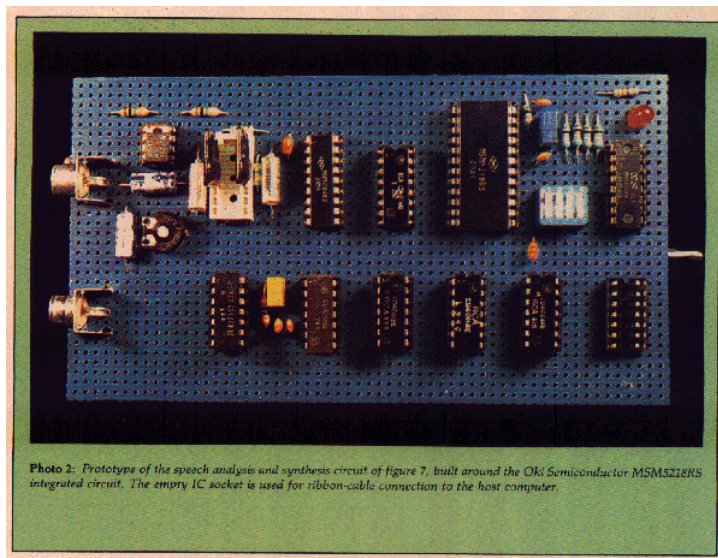


Figure 6: Functional block diagram of the Oki Semiconductor MSMS218RS ADPCM integrated circuit.

- Remember that we originally calculated that a rate of 96,000 bps would be needed to reproduce speech with this same fidelity. (Here we used an 8-bit A/D converter for economy; the bit rate would be the same for a 12-bit converter.) With a slight sacrifice in fidelity, the bit rate can be reduced further. By selecting the 4-kHz sample rate and 3-bit ADPCM codes, a 12,000 bps rate is achieved. This may still sound like a lot of data, especially when you compare it to phoneme and LPC (linear-predictive coding) speech synthesizers like the Votrax SC-1A and the Digitalker, which by comparison use data rates of 70 to 1000 bps. The difference, of course, is speech quality and intelligibility. A phoneme or LPC synthesizer generates its own sounds and forms them into words. An ADPCM synthesizer, on the other hand, retains the inflection and intonation of the original human voice. With ADPCM, as with an analog recording, it's possible to have a voice output that reproduces the regional accents of the human speaker.



- The circuit of figure 7 can be used as both an analyzer and a synthesizer. Both subsystems function concurrently when the MSM5218 is in the analysis mode; the results, the reconstructed waveform, can be heard in real time (delayed by 3 VCLOCK periods). In figure 7, this output is smoothed by a low-pass filter and externally amplified to drive a speaker.

Fig. 7

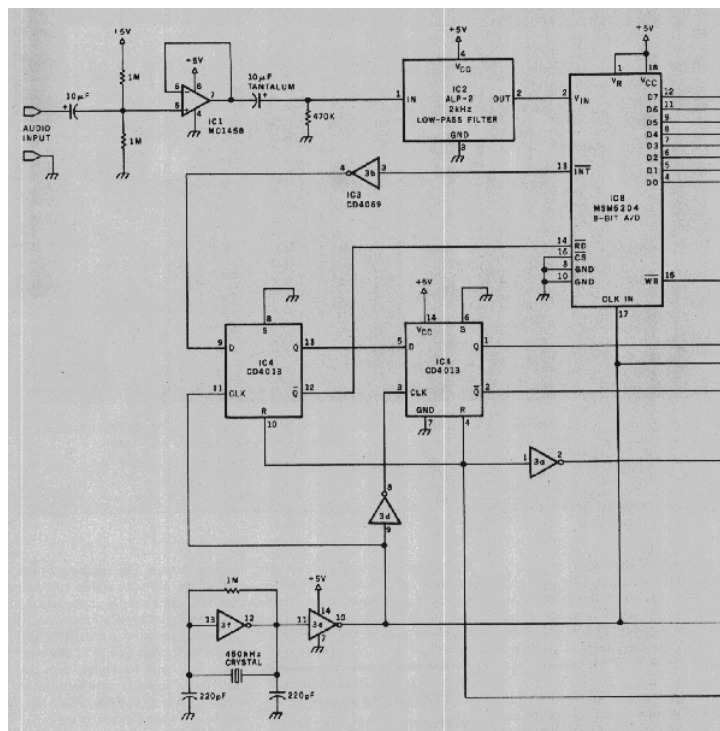


Fig. 7

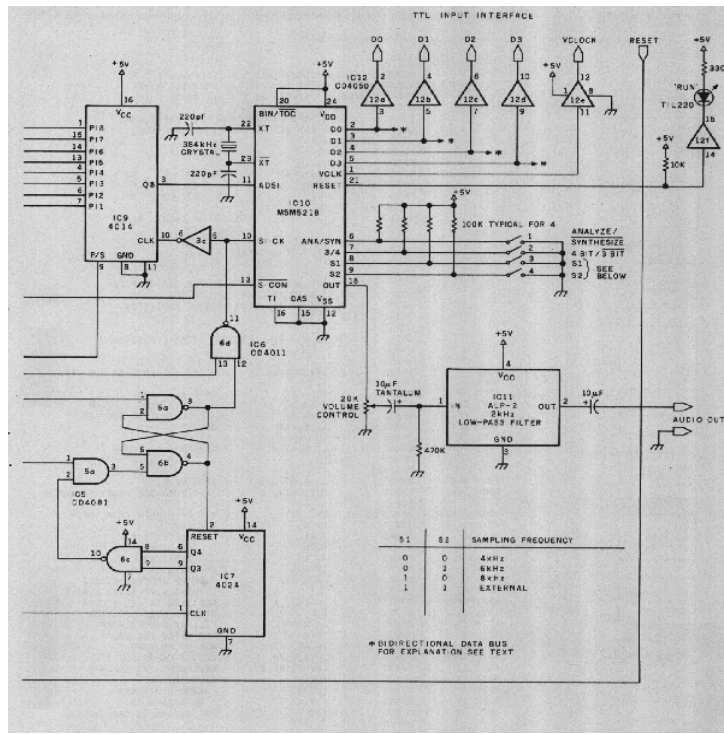


Fig. 7

An ADPCM speech analysis and synthesis (storage and reproduction) circuit built around the Oki MSM5218RS chip. A low-cost 8-bit A/D converter is used in place of a higher-resolution and more costly 12-bit converter. The Oki MSM5204RS 8-bit CMOS A/D converter, used in this circuit, contains a successive-capacitor-ladder conversion system. It also incorporates a sample and hold stage that enables direct input of rapidly changing analog signals. An external clock-signal provides timing for the chip; the clock's frequency is not critical and can be anywhere from 450 to 500 kHz. The frequency bandwidth of the signal input to the A/D converter is limited by an active low-pass filter, IC2, an Oki ALP-2 filter with a 1.7-kHz cutoff frequency and attenuation of 18 dB per octave above the cutoff frequency.

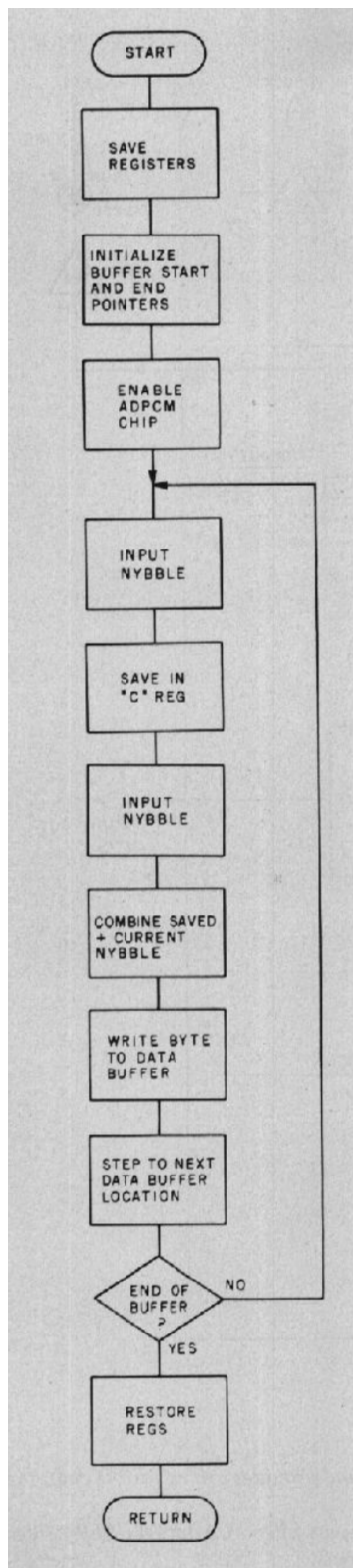
#### Use of the ADPCM Circuit

- As I said at the beginning, the purpose of this project is to create intelligible machine-generated speech. With the circuit of figure 7 connected to a Z80-based computer, and using the LOAD routine in the program of listing 1 (the algorithm shown in the flowchart of figure 8), you can analyze and store 10 seconds of speech (or 20 seconds at the 4-kHz sample rate).

#### Flowchart Fig. 8

Algorithm of the LOAD routine in the program of listing 1. Used with the circuit of figure 7, LOAD takes analog voice signals from a microphone or other source and stores them in ADPCM-encoded form in user memory.





- The program simply turns on the synthesizer by lowering the reset line and then observing VCLOCK. At every negative-going transition it reads a 4-bit ADPCM nybble (there are 2 nybbles per byte) and stores it in a memory-resident table. For experimental purposes I set this table to occupy a rather large region of user memory (40K bytes). For most practical applications, you might prefer to store segments of speech on disk and load them into memory in smaller increments. Once some speech has been stored, you can play it back using the DUMP routine from listing 1 (whose algorithm appears in figure 9).

Listing 1

Listing 1: Z80 assembler program to control the speech-synthesis circuit of figure 7.

```

00010 ;
00020 ; THESE ROUTINES ARE USED TO ALLOW A Z-80
00030 ; BASED MICRO TO LOAD AND DUMP THE 4 BIT
00040 ; NYBBLES CREATED BY THE OKI ADPCM CHIP.
00050 ;
00060 ; ONE I/O PORT IS USED FOR BOTH INPUT AND
00070 ; OUTPUT. THE BIT MAP FOR THE PORT IS AS
00080 ; FOLLOWS :
00090 ;
00100 ;
00110 ;      B7  B6  B5  B4  B3  B2  B1  B0
00120 ;      *      *      *      *
00130 ;                      D3  D2  D1  D0
00140 ;
00150 ;      * CLOCK ON INPUT, RESET
00160 ;      ON OUTPUT
00170 ;
00180 ;      *      *      *      UNUSED
00190 ;
00200 ;
00210 ; SINCE THE MASTER CLOCK OF THE OKI CHIP IS
00220 ; USED BY THIS PROGRAM, ANY OF THE CLOCK
00230 ; RATES MAY BE SELECTED WITHOUT MODIFYING
00240 ; THIS CODE.
00250 ;
00260 ; AT 8KHZ, YOU WILL FILL 4K BYTES PER SECOND.
00270 ;
00280 ;
4000      00290      ORG 4000H      ; START AT 4000H ON D/G CPU.
4000      00300      LOAD      EQU $
4000 F5      00310      PUSH AF      ; SAVE ANY REGS
4001 C5      00320      PUSH BC      ; THAT I WILL CLOBBER
4002 E5      00330      PUSH HL      ; DURING THE ROUTINE
4003 2641      00340      LD H,41H      ; BUFFER STARTS AT
4005 2E00      00350      LD L,00H      ; 4100H
4007 06DF      00360      LD B,0DFH      ; AND ENDS AT DF00H
          00370 ;
          00380 ;
4009 3E00      00390      LD A,00H      ; PICK UP 0.
400B D311      00400      OUT (11H),A      ; AND TURN OFF RESET TO ADPCM
          00410 ;
400D CD2740      00420      LOOP1      CALL NIN      ; BIT UP NYBBLE FROM ADPCM
4010 07      00430      R/CA      ; PUT THE
4011 07      00440      R/CA      ; NYBBLE AT THE
4012 07      00450      R/CA      ; TOP OF THE
4013 07      00460      R/CA      ; REGISTER.
4014 4F      00470      LD C,A      ; AND SAVE IN THE C REG
4015 CD2740      00480      CALL NIN      ; PICK UP ANOTHER NYBBLE
4018 81      00490      OR C      ; COMBINE WITH SAVED NYBBLE
4019 77      00500      LD (HL),A      ; AND SAVE IN THE BUFFER
401A 23      00510      INC HL      ; STEP TO NEXT LOCATION
4018 73      00520      LD A,B      ; PICK UP END POINTER
401C BC      00530      CP H      ; ARE WE AT THE END ?
401D 20EE      00540      JR NZ,LOOP1      ; NO, DO ANOTHER
401F 3E2F      00550      LD A,OFFH      ; YES, PICK UP ALL 1'S
4021 D311      00560      OUT (11H),A      ; AND RESET THE ADPCM
4023 E1      00570      POP HL      ; RECOVER THE
4024 C1      00580      POP BC      ; SAVED REGISTERS
4025 F1      00590      POP AF      ; WE SCRAMBLED
4026 C9      00600      RET      ; AND RETURN TO THE CALLER.
          00610 ;
          00620 ;
          00630 ;
4027      00640      NIN      EQU $
4027 DB11      00650      INH1W      IN A,(11H)      ; PICK UP INPUT FROM ADPCM
4029 CB67      00660      BIT 4,A      ; CHECK FOR A 1 ON THE CLOCK
402B 28FA      00670      JR Z,INH1W      ; WAIT FOR ONE TO COME
402D DB11      00680      INLOW      IN A,(11H)      ; PICK UP INPUT FROM ADPCM
402F CB67      00690      BIT 4,A      ; CHECK FOR A 0 ON THE CLOCK
4031 28FA      00700      JR Z,INLOW      ; WAIT FOR ONE TO COME
          00710 ;
00720 ; WHEN A 1 TO 0 TRANSITION HAS OCCURRED, THE DATA
00730 ; IN THE A REG IS THE NYBBLE WE WANT TO STORE.
00740 ;
4033 860F      00750      AND 0FH      ; MASK OFF THE HI BITS

```

Listing 1 continued on page 46

Listing 1 continued

Listing 1 continued:

```

4035 C9      00760      RET          ; AND RETURN TO CALLER
              00770 ;
              00780 ;
              00790 ;
              00800 ;
              00810 ;
              00820 ;
              00830 ;
4036        00840 DUMP  EQU $
4036 F5      00850      PUSH AF      ; SAVE THE REGS
4037 C5      00860      PUSH BC      ; JUST LIKE
4038 F5      00870      PUSH HL      ; BEFORE
4039 2641    00880      LD H,41H    ; SET UP START
403B 2E00    00890      LD L,08H    ; ADDRESS OF BUFFER
403D 06DF    00900      LD B,0DFH  ; SET UP END TOO
403F 3E00    00910      LD A,00H    ; PICK UP ALL 0'S
4041 D311    00920      OUT (11H),A  ; AND TURN OFF RESET TO ADPCM
              00930 ;
              00940 ;
4043 7E      00950 LOOP2  LD A,(HL)  ; PICK UP STORED DATA BYTE
4044 E6F0    00950      AND 0F0H  ; ISOLATE THE UPPER NYBBLE
4046 07      00970      RLC A      ; AND SWING
4047 07      00930      RLC A      ; IT TO
4048 07      00990      RLC A      ; THE LOWER
4049 07      01000      RLC A      ; NYBBLE
404A CD6040  01010      CALL NOUT ; WRITE THE NYBBLE TO THE ADPCM
404D 7E      01020      LD A,(HL)  ; PICK UP THE BYTE AGAIN
404E E60F    01030      AND 0FH  ; LOWER NYBBLE THIS TIME
4050 CD6040  01040      CALL NOUT ; WRITE IT TO THE ADPCM
4053 23      01050      INC HL   ; STEP TO NEXT BUFFER POSITION
4054 78      01060      LD A,B   ; PICK UP END OF BUFFER POINTER
4055 BC      01070      CF H    ; ARE WE THERE YET ?
4056 20EB    01050      JR NZ,LOOP2 ; NO, DO SOME MORE
4059 36FF    01090      LD A,0FFH ; PICK UP ALL 1'S
405A D311    01100      OUT (11H),A ; TURN ON RESET TO ADPCM
405C E1      01110      POP HL  ; RECOVER THE
405D C1      01120      POP BC  ; REGS THAT WE
405E F1      01130      POP AF  ; HAVE CLUTTERED
405F C9      01140      RET          ; AND RETURN TO CALLER
              01150 ;
              01160 ;
              01170 ;
4060        01180 NOUT  EQU $
4060 47      01190      LD C,A   ; SAVE A FOR LATER
4061 DA11    01200 OL0W  IN A,(11H) ; PICK UP ADPCM BYTE
4063 C857    01210      BIT 4,A  ; TEST FOR 0 BIT
4065 20FA    01220      JR NZ,OL0W ; WAIT FOR ONE TO COME
4067 DB11    01230 OH1W  IN A,(11H) ; PICK UP ADPCM BYTE
4069 C857    01240      BIT 4,A  ; TEST FOR 1 BIT
406B 28FA    01250      JR Z,OH1W ; WAIT FOR ONE TO COME
              01260 ;
              01270 ; WHEN 0 TO 1 TRANSITION IS DETECTED, WE PUT OUR DATA
              01280 ; OUT TO THE ADPCM.
              01290 ;
406D 79      01300      LD A,C   ; RECOVER THE SAVED DATA
406E D311    01310      OUT (11H),A ; WRITE TO THE ADPCM
4070 C9      01320      RET          ; AND GO BACK TO CALLER
              01330 ;
              01340 ;
              01350 ;
              01360 ;
              01370 ; END OF LOAD/DUMP ROUTINES
              01380 ;
              01390 ;
0000        01400      END
00000 TOTAL ERRORS

```

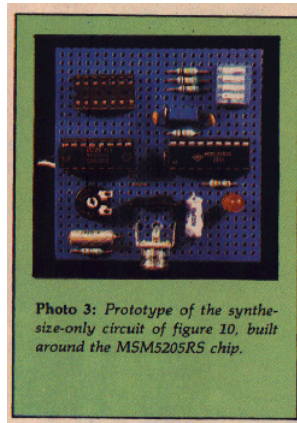
46 June 1983 © BYTE Publications Inc

- With the MSM5218 set to the synthesis mode, the ADPCM codes are sequentially loaded on each rising edge of VCLOCK. If you want to store some speech permanently and then play it back in a dedicated application (as an annunciator, for instance), you won't need the analysis part of the circuit after the ADPCM codes have been stored.

For such cases you may wish to use the synthesize-only circuit of figure 10, which uses the 18-pin MSM5205RS ADPCM-synthesis chip instead of the dual-function 24-pin 5218 (see photo 3 on page 48). The 5205's synthesis capabilities are equal in every way to those of the 5218, but the 5205 saves the expense and complication of the analysis section. The resulting 2-chip circuit, the parts of which cost less than \$15, can be easily manufactured for a variety of applications.

One significant aspect of ADPCM speech synthesis is the ease of producing a custom vocabulary.

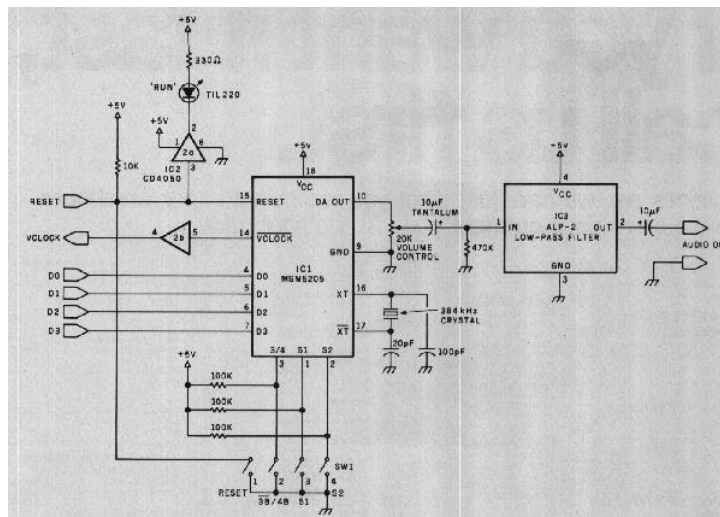
Photo 3



- I was pleasantly surprised at the fidelity using ADPCM at 32,000 bps. It was still more intelligible than the majority of current synthesis techniques even at 12,000 bps. While testing the software I attached the input of the analysis unit to an FM radio. Even when using the 1.7-kHz filters, I was surprised how good even music sounded.

Fig. 10

A voice-reproduction circuit built around the Oki MSM5205RS speech synthesis chip. This circuit is useful in applications where you need a fairly inexpensive means of reproducing a custom vocabulary. You can store your vocabulary with the circuit of figure 7., and load the encoded speech into this simple circuit for output



#### Summary of ADPCM Synthesis

- Probably the most significant aspects of ADPCM speech synthesis are the simplicity of the hardware and the ease of producing a custom vocabulary. You don't have to send a word list and recording tape to a manufacturer and wait for the company to spend days doing a Fourier analysis of the tape. To produce a ROM (read-only memory) containing your custom vocabulary, you can use merely a microphone and a simple LOAD/DUMP routine. It may require 4 to 5 times more memory space than other high-intelligibility speech-synthesis schemes, but the price of that memory is minuscule compared to the cost of producing vocabularies for the other schemes.

Fig. 9 Algorithm of the DUMP routine from listing 1.

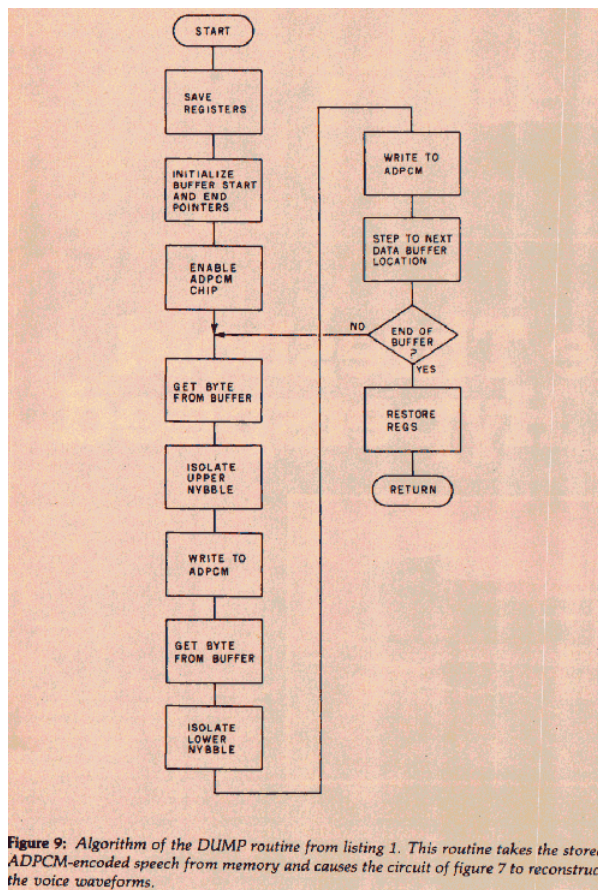


Figure 9: Algorithm of the DUMP routine from listing 1. This routine takes the stored ADPCM-encoded speech from memory and causes the circuit of figure 7 to reconstruct the voice waveforms.

#### Future Applications of ADPCM

- We've looked at ADPCM here only as it relates to voice synthesis, but in actuality, the possible applications of ADPCM to speech recognition prompted my initial interest. The first phase of any speech-recognition technique is digitizing the waveforms and getting them into the computer for analysis, compression, and comparison. My previous article on voiceprints (reference 5) demonstrated the large quantity of hardware necessary to merely condition the waveform for traditional speech-recognition methods. With ADPCM and these Oki chips, we have an inexpensive (under \$30) circuit for digitizing voice waveforms and presenting them to a computer in a form that it can digest. Even though 1500 to 4000 bytes of raw data per second of speech stream into the computer, the data thus recorded should be unique for each individual word. Speech recognition could be accomplished by brute-force comparison of all the data, or perhaps there exists some applicable compression algorithm that might reduce one second of data to 200 bytes or so. The final compacted data would not be for reconstruction of the original waveform but rather stored as a signature of the input word (derived from an ADPCM code table) for use in comparison.

We have accomplished the first step and now have means to place the ADPCM codes in memory. In the course of the next few months I will be experimenting with various compression and comparison techniques in hope of developing a practical speech-recognition project. But if by chance you happen upon the solution to the problem overnight, let me know.

#### References

<p><b>References</b></p> <ol style="list-style-type: none"> <li>1. Ciarcia, Steve. "Build a Low-Cost Speech-Synthesizer Interface." June 1981 BYTE, page 46.</li> <li>2. Ciarcia, Steve. "Build the Microvox Text-to-Speech Synthesizer." Part 1, September 1982 BYTE, page 64; Part 2, October 1982 BYTE, page 40.</li> <li>3. Ciarcia, Steve. "Build an Unlimited-Vocabulary Speech Synthesizer." September 1981 BYTE, page 38.</li> <li>4. Ciarcia, Steve. "Talk to Me: Add a Voice to Your Computer for \$35." June 1978 BYTE, page 142.</li> <li>5. Ciarcia, Steve. "Use Voiceprints to Analyze Speech." March 1982 BYTE, page 50.</li> <li>6. General Explanation of Oki ADPCM Speech Synthesis LSI (MSM5205RS). Tokyo: Oki Electric Ltd., 1982.</li> <li>7. Mier, Edwin E. "Competitors Fear AT &amp; T's New \$1 Billion Scheme." Data Communications, December 1982, page 39.</li> </ol>	<p><b>Editor's Note:</b> Steve often refers to previous Circuit Cellar articles as reference material for each month's current article. Most of these past articles are available in reprint books from BYTE Books, McGraw-Hill Book Company, POB 400, Hightstown, NJ 08250.</p> <p>Ciarcia's Circuit Cellar, Volume I covers articles that appeared in BYTE from September 1977 through November 1978. Ciarcia's Circuit Cellar, Volume II contains articles from December 1978 through June 1980. Ciarcia's Circuit Cellar, Volume III contains the articles that were published from July 1980 through December 1981.</p> <p>Digitalker is a trademark of National Semiconductor Corporation.</p>
---	--

To receive a complete list of Ciarcia's Circuit Cellar project kits available from the Micromart, circle 100 on the reader service inquiry card at the back of the magazine.

## Parts

*For those of you who would like to construct an ADPCM analyzer/synthesizer, the chips are readily available from Oki Semiconductor Corporation as part of an experimenter's kit.*

*Included are:*

- *Oki MSM5218 analyzer/synthesizer*
- *Oki MSM5205 synthesizer*
- *Oki MSM5204 8-bit A/D converter*
- *384-kHz ceramic resonator*
- *Oki MSM5128-15 CMOS 2K by 8-bit static RAMs (eight)*
- *ALP-2 hybrid 1.7-kHz filter*
- *ALP-4 hybrid 3.4-kHz filter*

*The package is called the Real-Voice Application Kit. It sells for \$89 post-paid (check or money order only—no CODs or company purchase orders) and can be ordered from:*

*Oki Semiconductor, Inc.  
1333 Lawrence Expressway, Suite 401  
Santa Clara, CA 95051  
telephone: (408) 984-4842  
TWX: (910) 338-0508*