Fall 1978

Speech Synthesis Utilizing Microcomputer Control

Joseph N. Uzel
University of Central Florida, ju84123@aol.com

SPEECH SYNTHESIS UTILIZING
MICROCOMPUTER CONTROL

BY

JOSEPH N. UZEL
B.S.A.E., Virginia Polytechnic Institute, 1972
B.S.E., Florida Technological University, 1977

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science: Engineering
in the graduate studies program of
the College of Engineering
of Florida Technological University at Orlando, Florida

Fall Quarter
1978

SPEECH SYNTHESIS UTILIZING
MICROCOMPUTER CONTROL

JOSEPH N. UZEL

ABSTRACT

This report explores the subject of speech synthesis. Information given includes a brief explanation of speech production in man, an historical view of speech synthesis, and four types of electronic synthesizers in use today.

Also included is a brief presentation on phonetics, the study of speech sounds. An understanding of this subject is necessary to see how a synthesizer must produce certain sounds, and how these sounds are put together to create words.

Finally a description of a limited text speech synthesizer is presented. This system allows the user to enter English text via a keyboard and have it output in spoken form.

The future of speech synthesis appears to be very bright. This report also gives some possible applications of verbal computer communication.

Approved by: BW Patz
Director of Research Report

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

INTRODUCTION

For centuries man has been fascinated with the concept of artificially produced speech. Mechanical analogs of the vocal system produced speech sounds 200 years ago; however, it was the recent invention of the digital computer which made practical speech synthesis feasible.

The purpose of this report is to explore the subject of speech synthesis. Questions to be answered are: (1) How is speech produced in man? (2) What approaches have been taken in the past to synthesize speech? (3) What methods are presently used to electronically produce artificial speech?

In addition to answering the above questions this report also presents a description of a limited text speech synthesizer which utilizes a Votrax speech module and is controlled by a 6800 microcomputer. The control software allows the user to enter English text via a keyboard or to store coded text in a memory table and then have the text output in spoken form by the machine.

Some applications of speech synthesis include talking typewriters, verbal response learning systems, aircraft flight control, and telephone information retrieval systems.

CHAPTER 1

HUMAN SPEECH PRODUCTION

Human speech production is a physiological process which is understood fairly well. Figure 1 shows the human vocal tract and those components which make speech possible. The vocal tract is a non uniform acoustic tube, 16 to 18 centimeters in length, which extends from the glottis to the lips, and varies in shape as a function of time. The major anatomical components causing this time-varying change are the lips, tongue, jaw, and the velum. The velum is a flap which couples the nasal tract to the vocal tract through a trap-door type of action. The nasal cavity is about 12 centimeters long and has an approximate volume of 60 cubic centimeters(Flanagan 1972a).

The vocal system can produce three basic types of sound: voiced, fricative, and plosive. Voiced sounds, such as the vowels, are produced by the vibration of the vocal chords due to air released from the lungs. These vibrations interrupt the airflow and generate a series of sharp pulses that excite the vocal tract.

Fricative sounds(s,sh,f,th) are generated by forcing air through a constricted vocal tract at a high velocity which causes turbulence. Plosives(p,t,k) are produced by closing the vocal tract completely with the lips or tongue, allowing a pressure buildup, and then abruptly opening the closure.

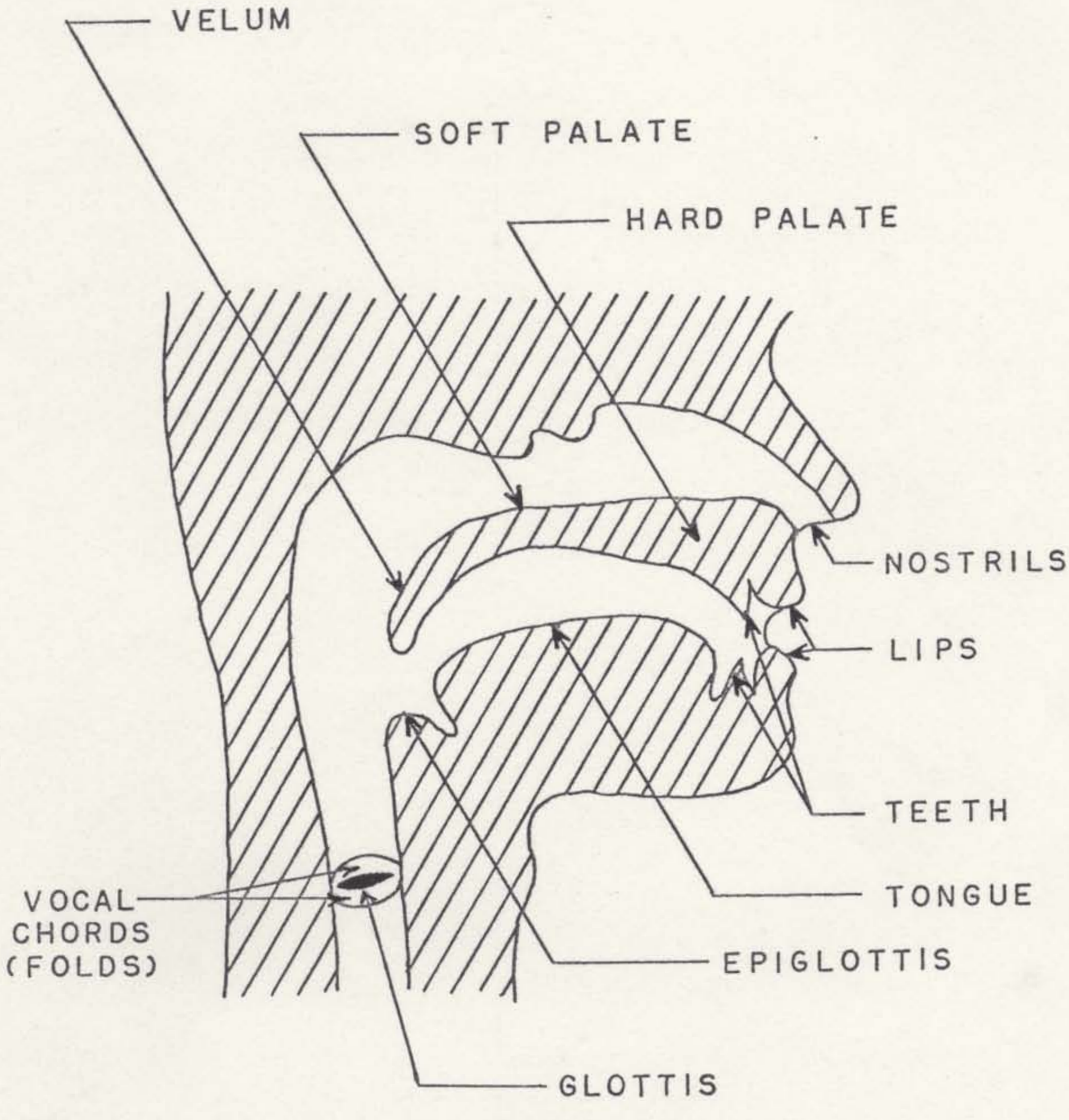


Fig. 1. The Human Vocal System

These speech sounds have a fairly broad spectrum of frequencies ranging from about 100 Hz to more than 3000 Hz. The vocal system acts as a time-varying filter to produce resonance characteristics.

For voiced sounds, the excitation source is typically pulsive and periodic, with a spectrum whose harmonics decrease in amplitude approximately 12 dB per octave. The vocal tract acts as a filter with poles corresponding to the acoustic resonances known as formants. The resonant frequencies roughly equal the odd quarter-wave resonances of a pipe 17 centimeters long. For a straight pipe of this length, the first three resonant frequencies are 500, 1500, and 2500 Hz. The vocal tract, however, is not of constant diameter, and the formants will not be located exactly at the 1000 Hz intervals described.

When the nasal tract is made part of the transmission system, another resonance pole and an antiresonance zero are introduced at around 1400 Hz. The various voiced sounds of speech are produced by changing the shape of the vocal tract and hence its resonances.

Unvoiced sounds are excited by a noise source that has a fairly broad, uniform spectrum. The source such as the tongue constricting flow in the back of the mouth, effectively alters the length of the vocal tract. Thus the resonances and antiresonances fall at different frequencies(Flanagan 1972b).

CHAPTER 2

A HISTORY OF SPEECH SYNTHESIS

Man has been fascinated with the production of speech for many centuries; however, the first successful artificial speech sounds were produced 200 years ago. In 1779 Christian Gottlieb Kratzenstein constructed a set of acoustic resonators which produced the vowel sounds a,e,i,o,u. The resonators were activated by a vibrating reed which, like the human vocal chords, interrupted a stream of air.

In 1791 Wolfgang Von Kempelen of Vienna, demonstrated an elaborate machine for generating connected speech. However, his machine was not taken seriously by the scientific community because of an earlier chess playing "machine;" a hoax, which involved a legless master chess player as the principle "mechanism." His speech machine was legitimate. A bellows was used to supply air to a reed. This in turn excited a hand controlled resonator that produced voiced sounds. Consonants and nasals were simulated by four constricted passages (Flanagan 1972a).

In 1820 a machine was constructed which could carry on a normal conversation when manipulated by a skilled operator. The machine, built by Joseph Faber, a Viennese professor, was demonstrated in London where it sang "God Save the Queen." Like the Von Kempelen synthesizer, Faber's also incorporated bellows, reeds, and resonant cavities to simulate the human vocal tract(Atmar 1976).

The first electrical synthesizer was demonstrated at the 1939 World's Fair. Built by Bell Telephone Laboratories, the VODER (Voice Operated Demonstrator) consisted of a periodic buzz oscillator to simulate the vocal chords, and a wide-band noise source to simulate constricted air flow for fricative production. These sounds were modified by passing them through a resonance control box containing 10 continuous bandpass filters that spanned the frequency range of normal speech. Ten finger keys activated gain control potentiometers which modulated the outputs of the filters. Three additional keys provided a transient excitation of selected filters to simulate three types of plosive sound: t-d, p-b, k-g. A wrist bar selected noise or buzz source and a foot pedal controlled the pitch of the buzz oscillator. Figure 2 shows a schematic of the VODER.

The VODER was limited only to demonstrational use due to its complex controls and bulky size. The development of the digital computer provided a boost to the production of practical speech synthesizers. The computer took over the control functions as well as provided greater freedom in modeling the vocal system because of its high speed computational abilities.

Technical advances in integrated circuit design also contributed to the feasibility of practical speech synthesis. Higher density integrated circuits have enabled designers to include more functions in a smaller amount of space.

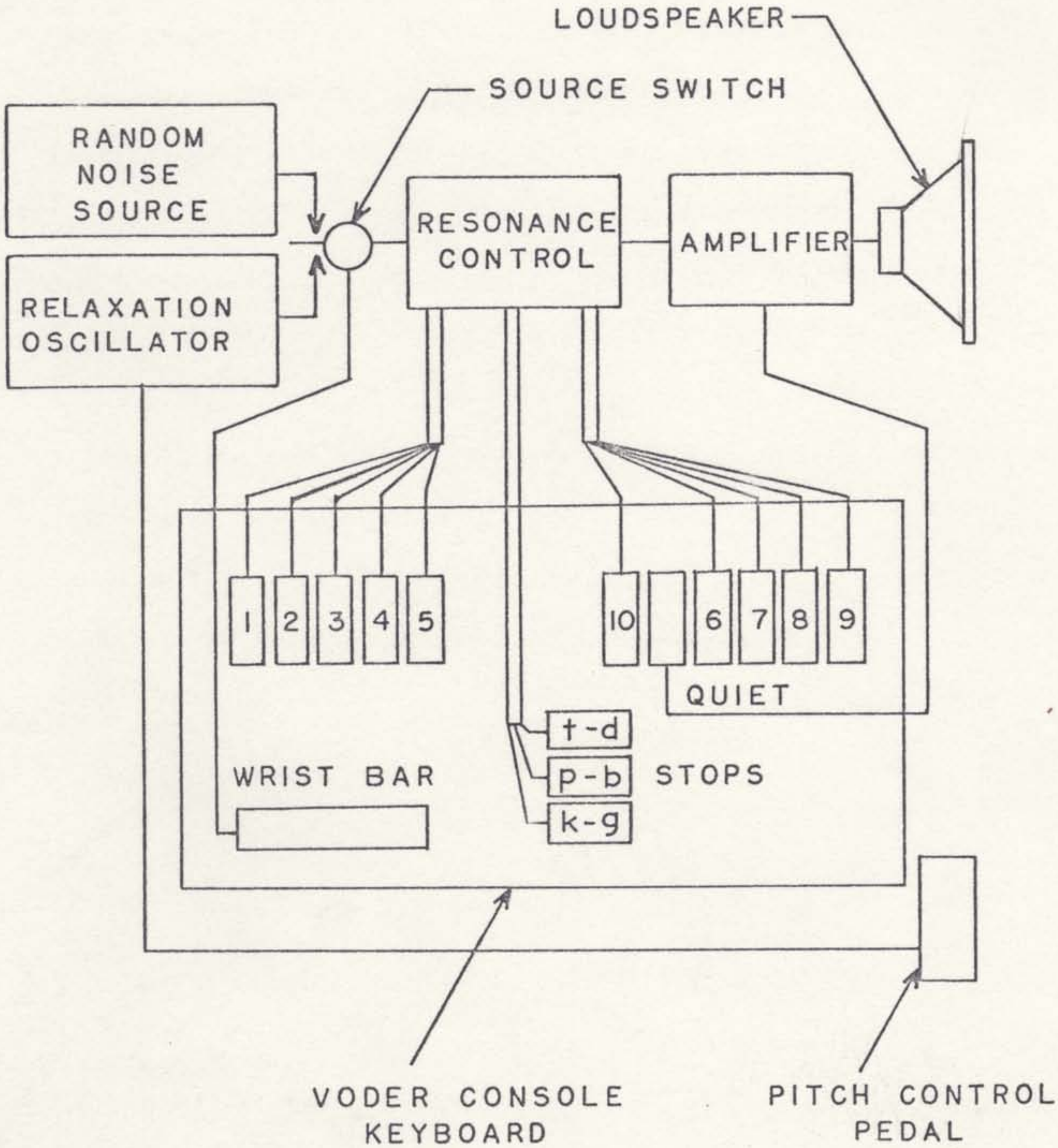


Fig. 2. Block Diagram of the VODER (Flanagan 1972a)

A speech synthesizer on a single LSI chip is presently a reality. Texas Instruments has produced a P-channel MOS chip, called the TMC0280, which synthesizes speech through a technique called linear predictive coding(LPC). This is a method of analyzing and storing human speech by extracting information from an input speech waveform and storing it in memory. Speech reproduction is accomplished by reconstructing speech from the information stored. This system uses a TMS1000 microprocessor along with two 16 K-byte ROMs, each with the ability to store over 100 seconds of speech. The fact that this circuit has been fabricated on a single chip indicates that very small and inexpensive speech synthesizers will shortly become a reality(Wiggins and Brantingham 1978).

CHAPTER 3

PHONETICS

In order to design or program an electronic speech synthesizer, it is essential to have a knowledge of phonetics.

Phonetics is the scientific study of speech sounds from the standpoint of their production, reception, and symbolization. A word is phonetically described by breaking it into distinctive sound units called phonemes. There are 38 phonemes in the English language (Atmar 1976). However, subtle differences in a phoneme call for another classification, the allophone. These are the slightly varying sounds of a single phoneme usually determined by the position in a word and phonemes preceeding and following it. Table 1 gives a list of International Phonetic Alphabet(IPA) symbols and key words demonstrating their use.

In a sense, phonetics is a language in itself. For example, the following are some common American English words with their phonetic transcriptions:

the	ðᵊi	shall	ʃæl
and	ænd	should	ʃUd
but	bʌt	was	wəz
to	tu	can	kæn
by	baI	as	æz

Thus, "ʃi wəz æt houm" is a phonetic transcription of "she was at home."

TABLE 1

IPA PHONETIC SYMBOLS(Leutenegger 1963)

Phonetic Symbol	Key Words	Additional Spellings of Sound
[p]	part	rope, stopped, hiccough
[b]	be	rubbed, cupboard
[t]	to	butt, debt, raked, indict, yacht, Thomas, receipt, might
[d]	do	add, rubbed, could
[k]	keep	cue, sick, account, lake, ache, walk, khaki
[g]	give	egg, ghost, guest
[f]	fame	off, phrase, laugh
[v]	vest	of, have, Stephen
[θ]	thin	
[ð]	the	
[s]	see	miss, scent, schism, cinder, psalm, sword, waltz
[z]	zoo	fuzz, raise, scissors, xylophone
[ʃ]	ship	issue, sugar, pension, gracious, ration, champagne, anxious, schottische, conscious
[ʒ]	lesion	leisure, azure, negligee
[h]	he	whole
[m]	milk	summer, comb
[n]	no	inn, pneumonia, Wednesday, mnemonic, knife, gnash
[l]	lake	tell
[w]	wig	language
[r]	red	merry, rhetoric, wrist
[j]	yes	onion
[tʃ]	chew	cello, witch, feature
[dʒ]	just	rage, gem, dodge, soldier
[ŋ]	sing	anchor, handkerchief, tongue
[i]	see	eat, people, chief, perceive, be, key, phoenix, ravine, Caesar
[I]	sit	here, hear, sieve, hymn, business, women, guild
[e]	ache	aim, beige, great, play, grey, gauge
[ε]	end	said, pear, says, heir, leopard, friend, any
[æ]	can't	laugh, half, as
[ɝ]	earn	worst, fir, fur, purr, germ, myrtle, journey, colonel
[ɚ]	ladder	surprise, sailor, liar
[ʌ]	up	son, tough, gull, does, blood
[ə]	sofa	succeed, famous, bargain, specimen

TABLE 1--Continued

Phonetic Symbol	Key Words	Additional Spellings of Sound
[u]	food	rude, whose, through, threw, shoes, group, blue
[U]	book	could, fully, wolf
[O]	rope	oak, though, sown, sew, goes, yeomen, shoulder, beau
[ɔ]	aught	raw, cough, abroad, gone, thought, all
[α]	farm	hot, honest
[aI]	sky	write, height, aisle, buy, lye, eye, aye, pie, sigh
[aU]	out	bough, crowd, hour, sauerkraut
[ɔI]	boy	broil

Phonemes can be divided into five classes as shown in Table 2. Some definitions are required for a better understanding of the table. A voiced sound was defined earlier as a sound in which air from the lungs causes the vocal chords to vibrate. The pulsating air stream is then resonated in the throat, mouth, and/or nasal cavity.

A diphthong is a voiced speech sound which changes smoothly from one vowel to another in the same syllable. This smooth transition is also called gliding. An example of a diphthong is the "oy" in boy.

A stop is caused by a complete momentary closure of the vocal tract, and then an explosive release of the built up pressure. Thus a voiced stop is a sound in which the released airstream of a stop has a vibratory element.

Nasal sounds result from resonance in the nasal cavity due to a complete closure of the mouth.

Fricatives result from a constriction of the air passage of the vocal tract, resulting in turbulence in the air stream. Fricatives can be voiced or unvoiced, and can include stops.

The h sound is an aspirant, also called a glottal fricative. It is produced by the airstream in the glottis, the opening between the vocal folds.

Affricates are consonant sounds produced by beginning with the vocal organs in the position of a stop. The corresponding pressure build up is released through a fricative constriction. Affricates can be either voiced(dʒ) or unvoiced(tʃ)(Wise 1958).

A speech synthesizer must be able to produce these sounds and connect them in such a way that the transition from one phoneme to another is accomplished as smoothly and naturally as possible.

TABLE 2

PHONEME CLASSES

1. Voiced

    a. vowels and diphthongs of vowels
    e, I, O, u, ɔ, ɛ, ɜ, ʌ, ə, æ, a, U, aI, au, ɔI, i, ɚ

    b. liquid consonants
    r, l, w, j

2. Stops(Plosives)

    a. voiced
    b, d, g

    b. unvoiced
    t, p, k

3. Nasal closures
    m, n, ŋ

4. Fricatives

    a. voiced
    z, ʒ, v, ð

    b. unvoiced
    s, ʃ, f, θ

    c. glottal(aspirant)
    h

5. Affricates

    a. voiced
    dʒ

    b. unvoiced
    tʃ

CHAPTER 4

ELECTRONIC SPEECH SYNTHESIS

The main components of a speech synthesis system are shown in Figure 3. The machine is required to speak a message, typically an English text. A synthesis program may access from a vocabulary store, or a set of vocabulary rules, to obtain a description of the required sequence of words. This description is then transferred to a synthesis device which transforms the information into speech.

Many different approaches to speech synthesis exist; however, the techniques are distinguished mainly by the memory storage requirements for the vocabulary and by the complexity of the control rules for generating the speech. Four different techniques illustrate the range of complexity. They are: adaptive differential pulse-code modulation(ADPCM), linear predictive coding(LPC), formant synthesis, and text synthesis. Typical bit-rate comparisons are shown in Table 3.

Adaptive Differential Pulse Code Modulation

The first and simplest technique utilizes a vocabulary composed of human-spoken words whose waveforms are digitally coded. The speech signal is band-limited to f_N Hz(the lowest frequency from which intelligable speech can be reconstructed). The signal is sampled at a rate of at least $2f_N$ samples per second(the Nyquist rate). In communication systems the most familiar example of this

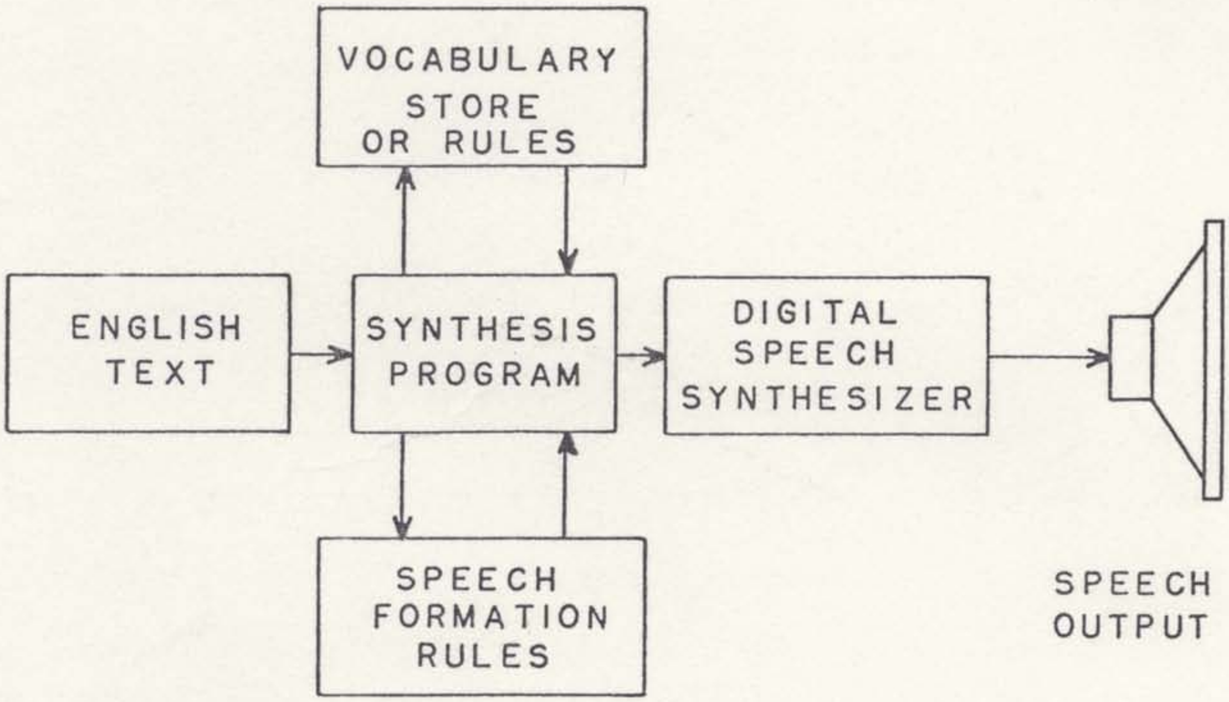


Fig. 3. Block Diagram of a Computer Voice Response System

TABLE 3

COMPARISON OF DATA RATES FOR
STORAGE OF SPEECH VOCABULARIES(Flannagan 1976)

CODING	BIT RATE	DURATION OF SPEECH IN 10^6 BITS OF STORAGE
ADPCM	20 k-bits/sec	1 min
LPC	2.4 k-bits/sec	7 min
FORMANT	500 bits/sec	30 min
TEXT	75 bits/sec	240 min

process is called pulse-code modulation(PCM). In most PCM representations each sample can take on only 2^B possible values, where B is the length of the binary code words that represent a quantized sample. The bit rate or information rate equals $2f_N B$, since B bits are required for each sample. The quantization step size, Δ, is normally set so that $\Delta \cdot 2^B$ spans the expected peak-to-peak amplitude range of the input(Rabiner and Schafer 1976).

A better representation of the signal can be obtained using differential rather than uniform quantization. The simplest case is when the quantizer has only two levels. Since a single-bit word can be used to represent these levels, the bit rate equals the sample rate. Such systems are called delta modulators(DM). If the step size of the one-bit quantizer varies to match the amplitude of the difference signal, the system is called an adaptive DM(ADM).

For a sampling rate at just the Nyquist rate, a multibit quantizer must be used. This is differential pulse-code modulation(DPCM). If the quantizer step size varies with speech signal level, we have an adaptive DPCM system(ADPCM)(Rabiner and Schafer 1976). Figure 4 shows a block diagram of an ADPCM system. Logic "L" observes the bit stream produced by the coder and adjusts the quantizer step size to minimize slope overload and distortion. By comparison of Figures 5(a) and 5(b) it is seen that ADPCM provides a better estimate of the signal than does DPCM(Flanagan 1976).

For message assembly, the synthesis program merely pulls the digitally coded words from storage, and supplies them to an ADPCM decoder to produce an analog output. The disadvantage of this system is the large amount of data storage required, due to the high bit rate, as shown previously in Table 3.

Linear Predictive Coding

Linear Predictive Coding(LPC) represents a substantial storage saving over ADPCM. In LPC speech synthesis, the bit rate is reduced by a factor of 8 or more.

This is done by using coefficients, calculated from an input voice waveform, to control a time-varying digital filter. These coefficients, called reflection coefficients, are found from a set of predictor coefficients which minimize error. This minimization of error is simply a method of obtaining a parametric representation of the speech signal. The reflection coefficients, along with pitch

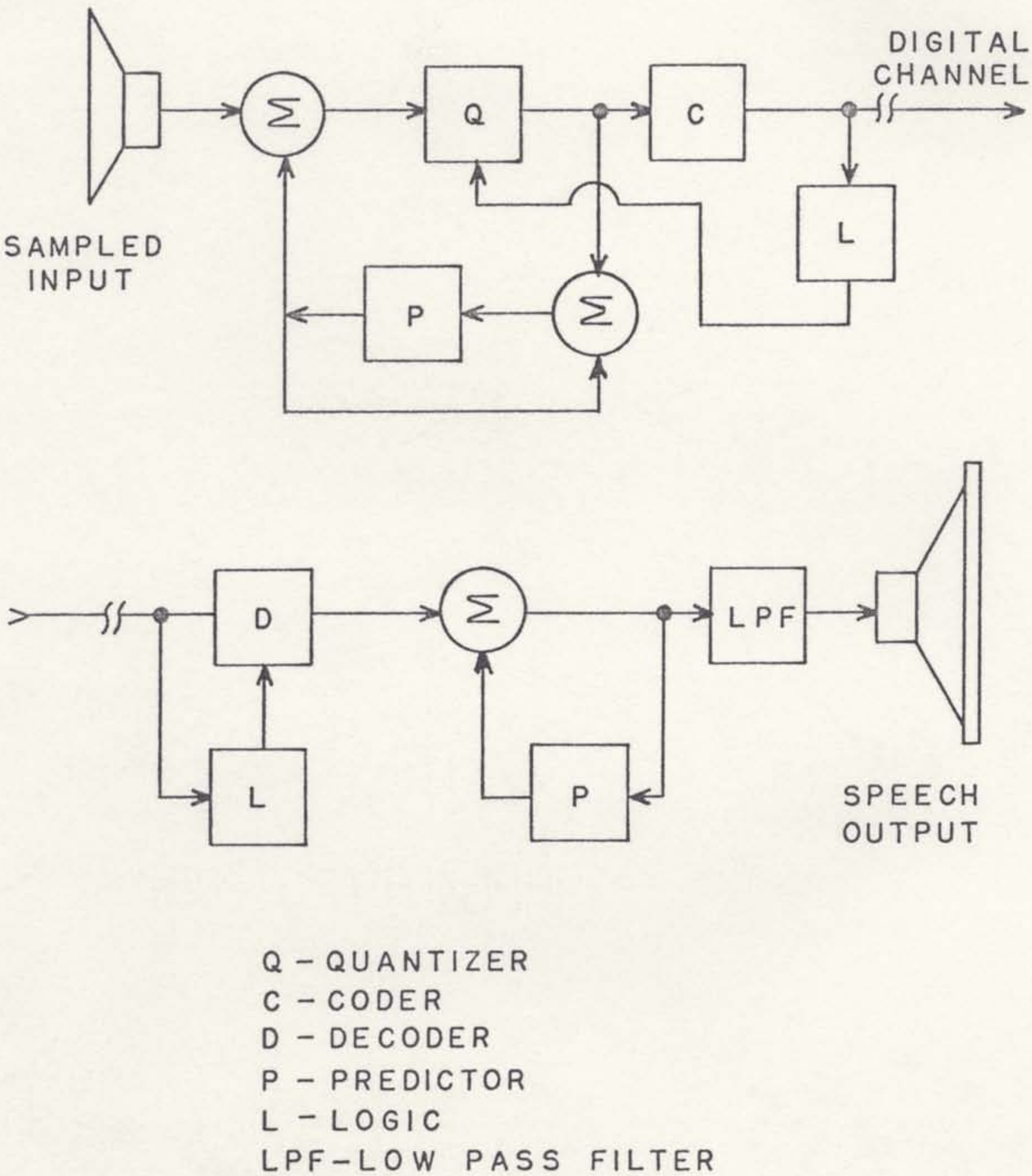


Fig. 4. Block Diagram of Adaptive Differential Coding of Speech (Flanagan 1976)

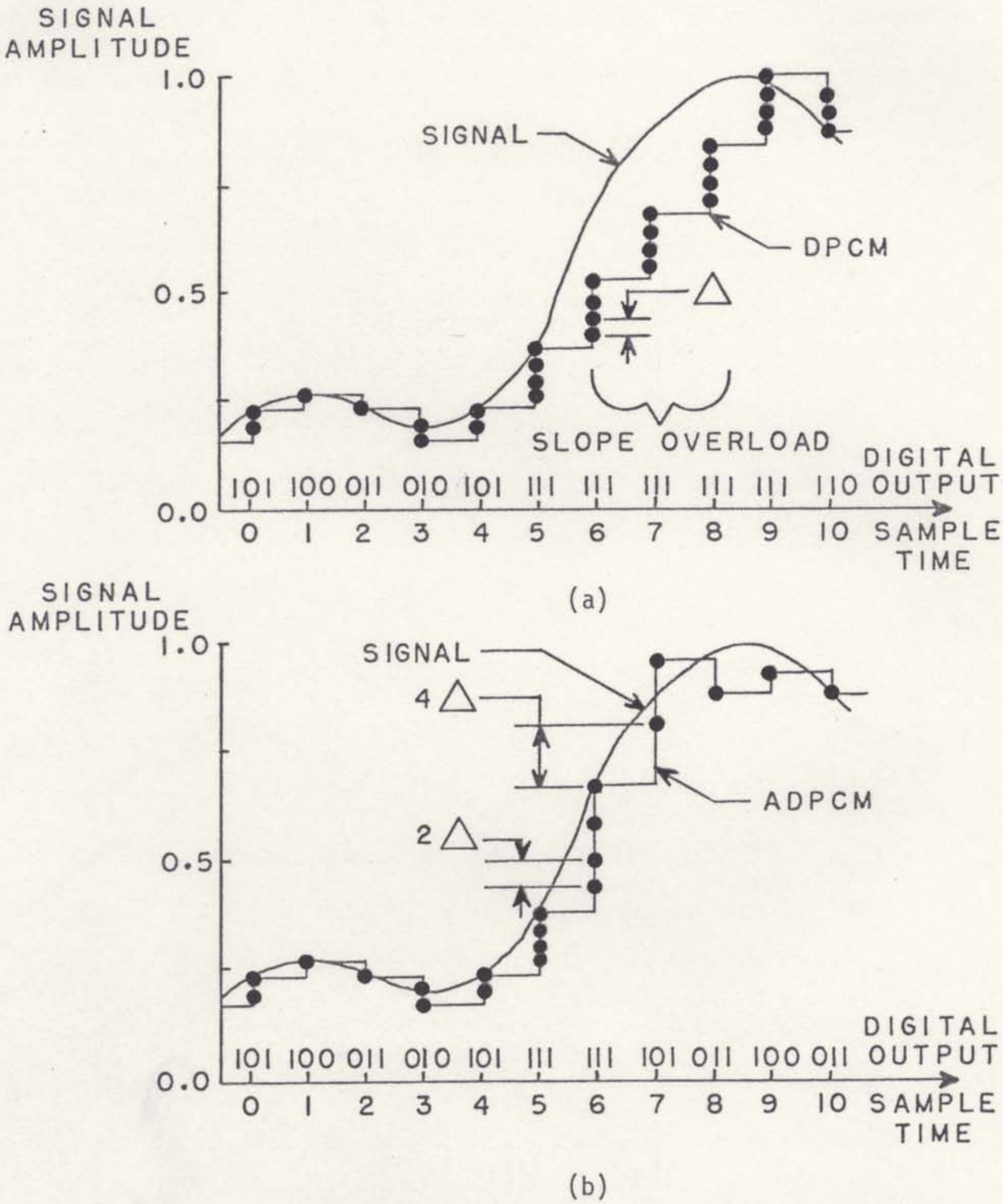


Fig. 5. Waveforms Coded by 3-Bit
(a) DPCM and (b) ADPCM(Flanagan 1976)

period and amplitude information, are transmitted to the synthesizer. Because of the physical constraints of the speech production process, these parameters need to be updated only every 10 to 20 milliseconds, thus resulting in a much lower bit-rate than necessary for ADPCM synthesis.

In LPC synthesis, the digital filter requires two sound sources. A white noise generator is used for unvoiced sound production, and a periodic source is used for voiced sounds. The output of the digital filter is converted from a digital signal to speech by a digital-to-analog converter. Figure 6 shows a diagram of how LPC is used in the TMC0280 synthesizer chip. Markel and Gray(1976), and Morgan and Craig(1976) present more detailed discussions of LPC and include computer programs for its use.

Formant Synthesis

The formant synthesizer, unlike the ADPCM and LPC types, does not require a voice input to be digitized and stored. Instead, a circuit is designed to reproduce sounds(usually phonemes and allophones) by manipulating filters and applying rules to produce the proper sound when used in the context of a word or sentence. Figure 7 shows block diagrams of serial and parallel formant synthesizers. Serial and parallel synthesizers both have their relative advantages and disadvantages; thus many different designs have been produced. See Atmar(1976), Holmes(1972), Klatt(1972), and Rice(1976).

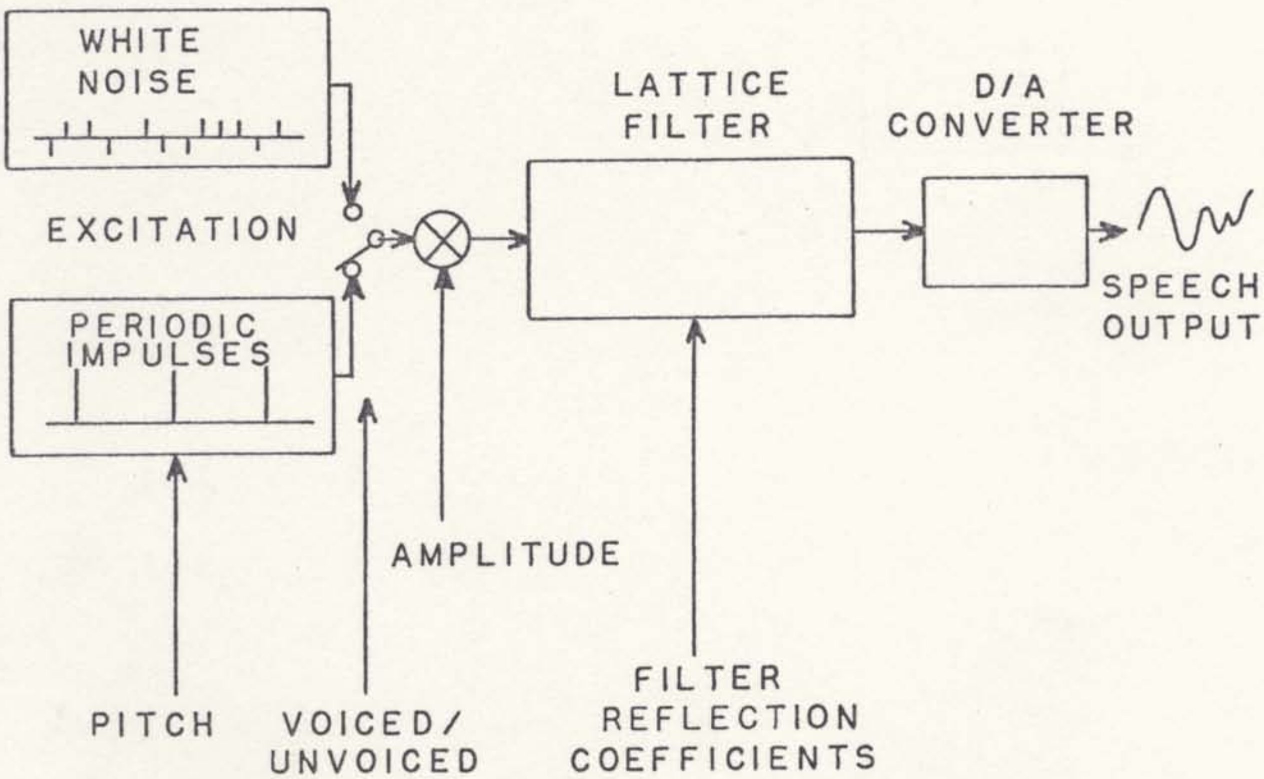
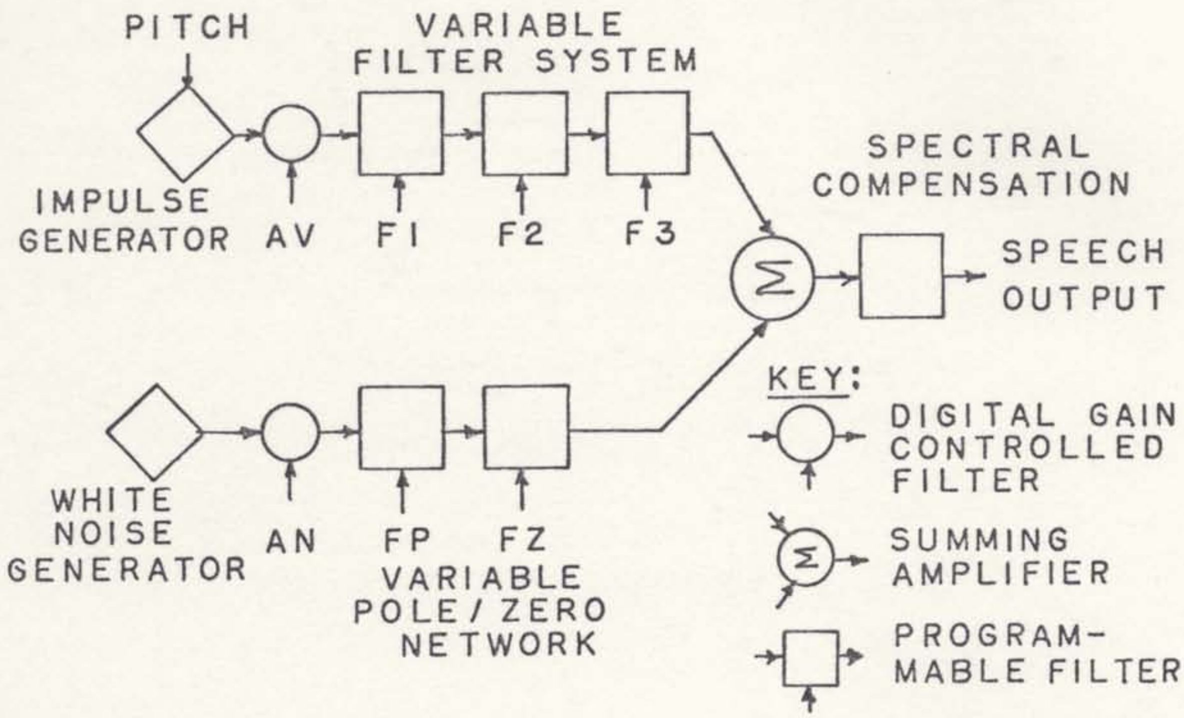
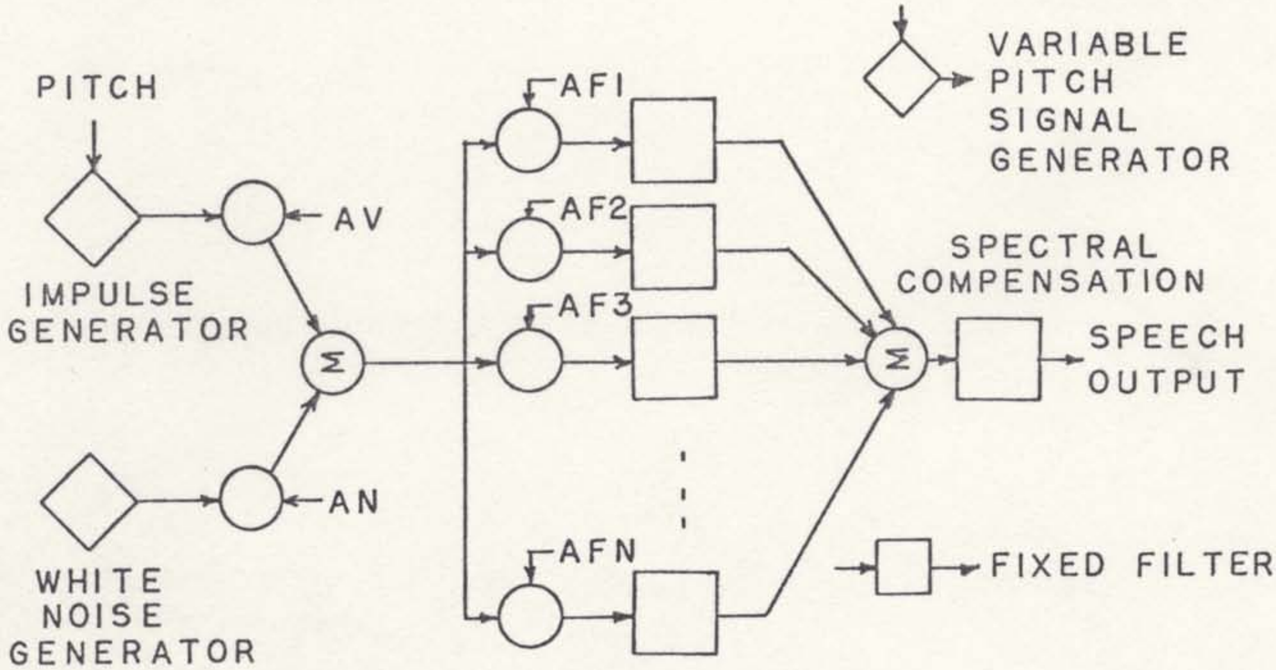


Fig. 6. Linear Predictive Coding Used in the TMC0280 Single Chip Synthesizer (Wiggins and Brantingham 1978)



(a)



(b)

Fig. 7. Formant Synthesizer Block Diagrams (a) Series (b) Parallel(Atmar 1976)

To model the vocal tract, a formant synthesizer typically has two excitation sources: an impulse generator to simulate a voicing source, and a white noise generator as a fricative source.

For vowel production the output of the impulse generator is fed into the formant filters. For minimal quality speech synthesis, the first three formants, F1, F2, and F3 are required. These are produced by variable digitally controlled resonator circuits. Higher formants are usually produced using fixed resonant circuits, since there is little change in the fourth and higher formants.

Most of the vowels of American English can be produced by the steady state formant frequencies shown in Table 4. These frequencies are approximate, and actual formant resonances vary from individual to individual.

TABLE 4

STEADY STATE ENGLISH VOWEL FORMANTS
(FREQUENCIES IN Hz)

	F1	F2	F3
heed	250	2300	3000
hid	375	2150	2800
head	550	1950	2600
had	700	1800	2550
hod	775	1100	2500
paw	575	900	2450
hood	425	1000	2400
who	275	850	2400

The distinctions between various vowel sounds can be seen more clearly by plotting them on a two dimensional graph as in Figure 8. F3 is not shown here because it varies only slightly for all vowels (except those with very high F2, where it is somewhat higher).

Some English vowels, the diphthongs, are characterized by rapid sweeps across the formant frequency space rather than relatively stable positions. Approximate traces of formants F1 and F2 are shown in Figure 9 for the vowels in bay, boy, buy, hoe, and how. These sweeps occur in 150 to 250 ms, depending on the speaking rate. The controlling computer must change the codes to the formant filters in such a way as to provide the desired transitions.

Stop consonant sounds "p," "t," "k," "b," "d," and "g" are formed by the white noise source. The amplified breakdown noise of a zener diode can be used as a simple fricative source. Figure 10 gives the formant frequency patterns necessary to produce the stop consonants when followed by the vowel "ah." The release of the stop closure(start of the noise pulse) is marked by "REL," and the beginning of the voicing sounds is marked by "VO."

A second group of consonant sounds consists of the liquids "w", "j," "r," and "l." These sounds are actually more like vowels. Thus, "w" and "j" can be associated with vowels "u" and "i" respectively. Timing makes the difference; for example, if the vowel "u" is immediately followed by vowel "a" and the rate of F1 and F2 transitions is increased, the result will sound like "wa." For the other liquids, "l" is marked by a brief increase of F3, while "r" is indicated by a

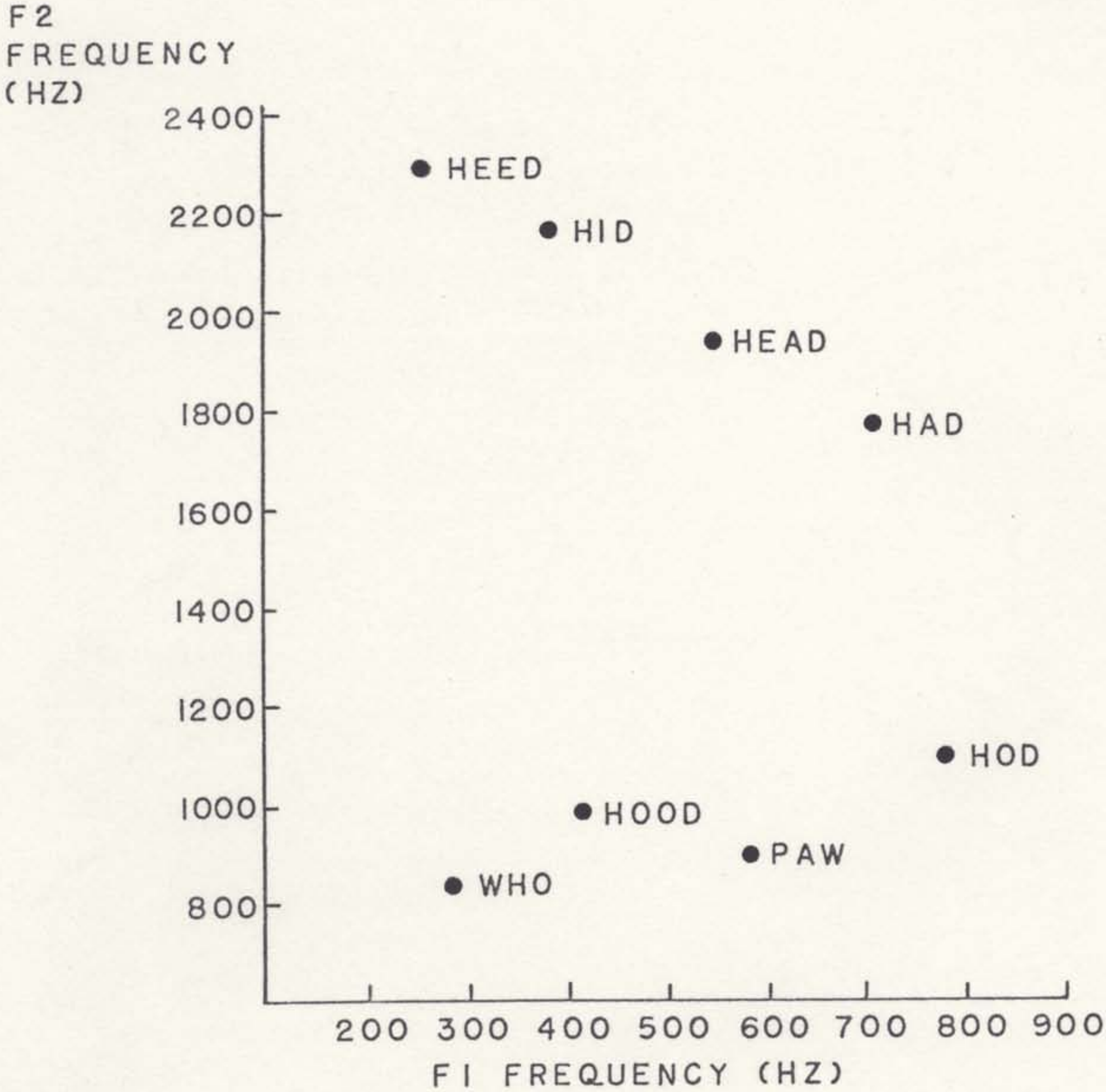


Fig. 8. Relationships Between the First Two Formants of Steady State English Vowels (Rice 1976)

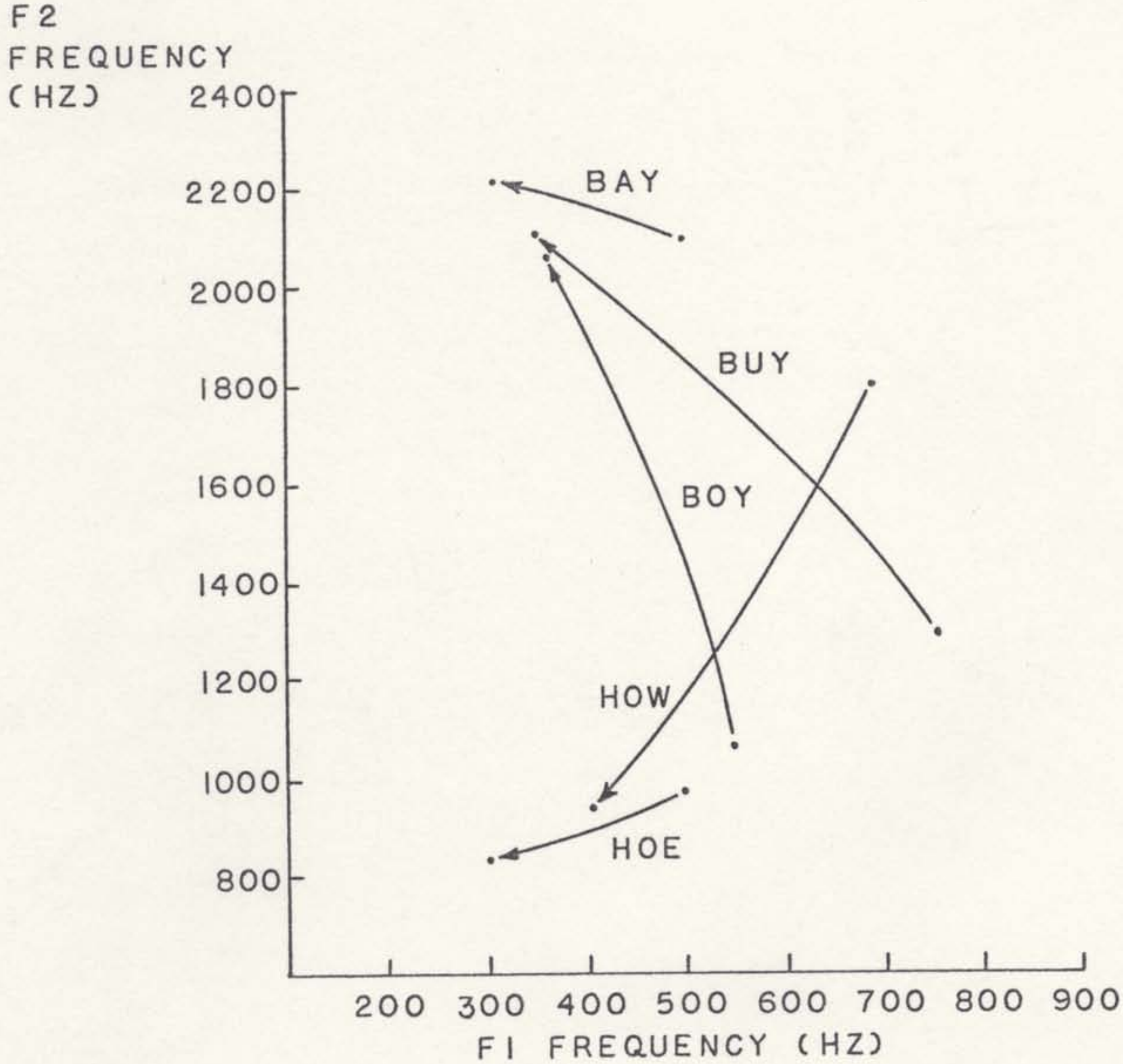


Fig. 9. Formant Sweeps for the English Diphthongs
(Rice 1976)

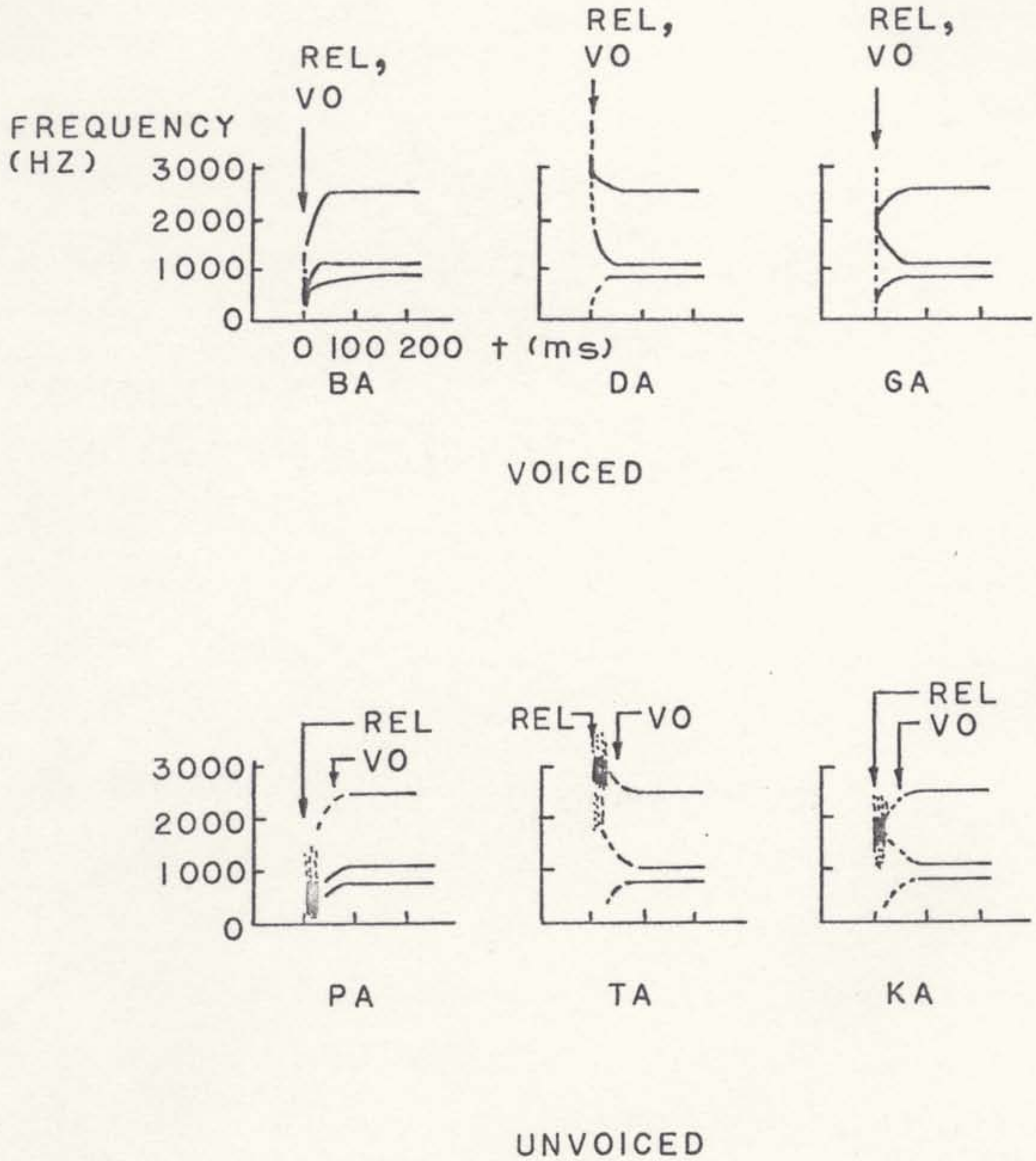


Fig. 10. Stop Consonant Patterns(Rice 1976)

sharp drop in F3 almost to the level of F2.

The nasal consonants "m," "n," and "ŋ" are very similar to the voiced stops "b," "d," and "g" respectively, except for the addition of a fixed nasal formant. This extra formant is generated by an additional resonator tuned to approximately 1400 Hz and having a fairly wide bandwidth. It is only necessary to control the amplitude of this extra resonator during the "closure" period to achieve the nasal quality in the synthesizer output.

Affricates "tʃ" and "dʒ" consist of the patterns for "t" and "d" followed immediately by the fricative "ʃ" or "ʒ" respectively, that is, "tʃ" = "t + ʃ" and "dʒ" = "d + ʒ." The fricatives "s," "ʃ," "z," "ʒ," "f," "v," "ð," and "θ" are characterized by a pulse of high frequency noise lasting from 50 to 150 milliseconds. Fricatives and affricates are classed as voiced or non-voiced. The first classification is according to voicing amplitude during the noise pulse, as seen in Figure 10 for voiced stops. Thus "s," "ʃ," "f," "tʃ," and the "θ" in "thin" have no voicing during the noise pulse, while "z," "ʒ," "v," "dʒ," and the "ð" in "then" have high voice amplitude. Thus different fricatives and affricates within a group are distinguished by the spectral characteristics of the noise pulse. Table 5 gives the fricative resonator settings needed to produce the various fricative and affricate consonants. Fricative noise amplitude settings are shown on a scale of 0 to 1(Rice 1976).

Figure 11 shows a partial circuit for a parallel formant synthesizer. This circuit is used in the Ai Cybernetic Systems Model

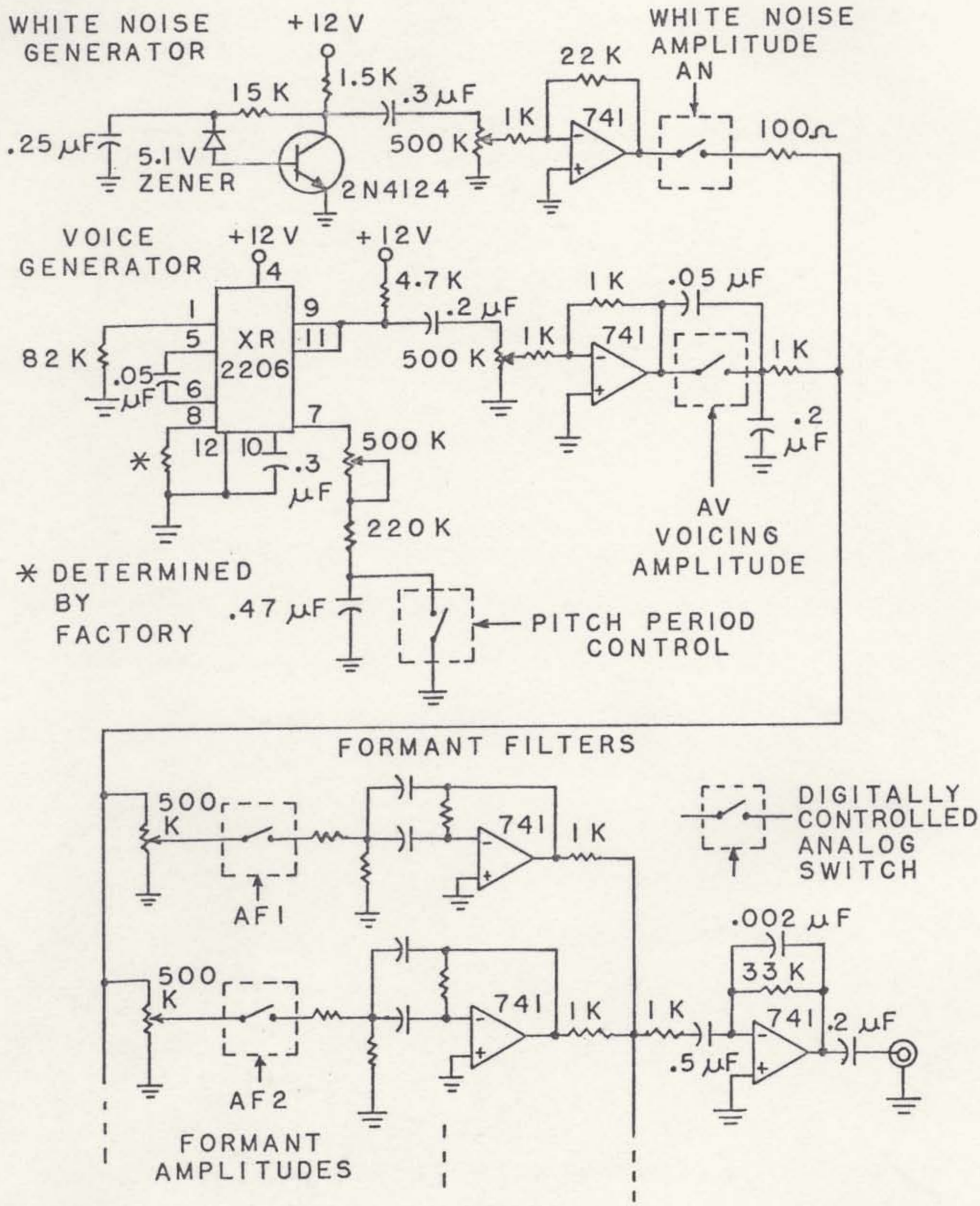


Fig. 11. Parallel Formant Circuit Used in the Ai Cybernetics Systems Model 1000 Speech Synthesizer (Atmar 1976)

TABLE 5

FRICATIVE SPECTRA

	RESONATOR FREQUENCY	FRICATIVE AMPLITUDE AN
ʃ, ʒ	2500 Hz	.9
s, z	5000 Hz	.7
f, v	6500 Hz	.4
θ ð	8000 Hz	.2

1000 Speech Synthesizer(Atmar 1976).

In order to build a quality formant synthesizer, the designer must consider much more than has been presented in the above discussion. The computer interface as well as the controlling software requirements must also be considered. Finally, detailed rules for phoneme and allophone production, pitch, duration, etc. must be programmed into the computer. A discussion of these rules is beyond the scope of this report; however, the interested reader can find this information given by the following authors: Klatt(1976), House(1961), Friedman(1975), Rao and Thosar(1974), and O'Shaugnessy (1974).

Text Synthesis

The final class of synthesizer to be considered here is a text synthesizer, shown in block diagram form in Figure 12. In its simplest form this type of synthesizer consists of a vocabulary stored

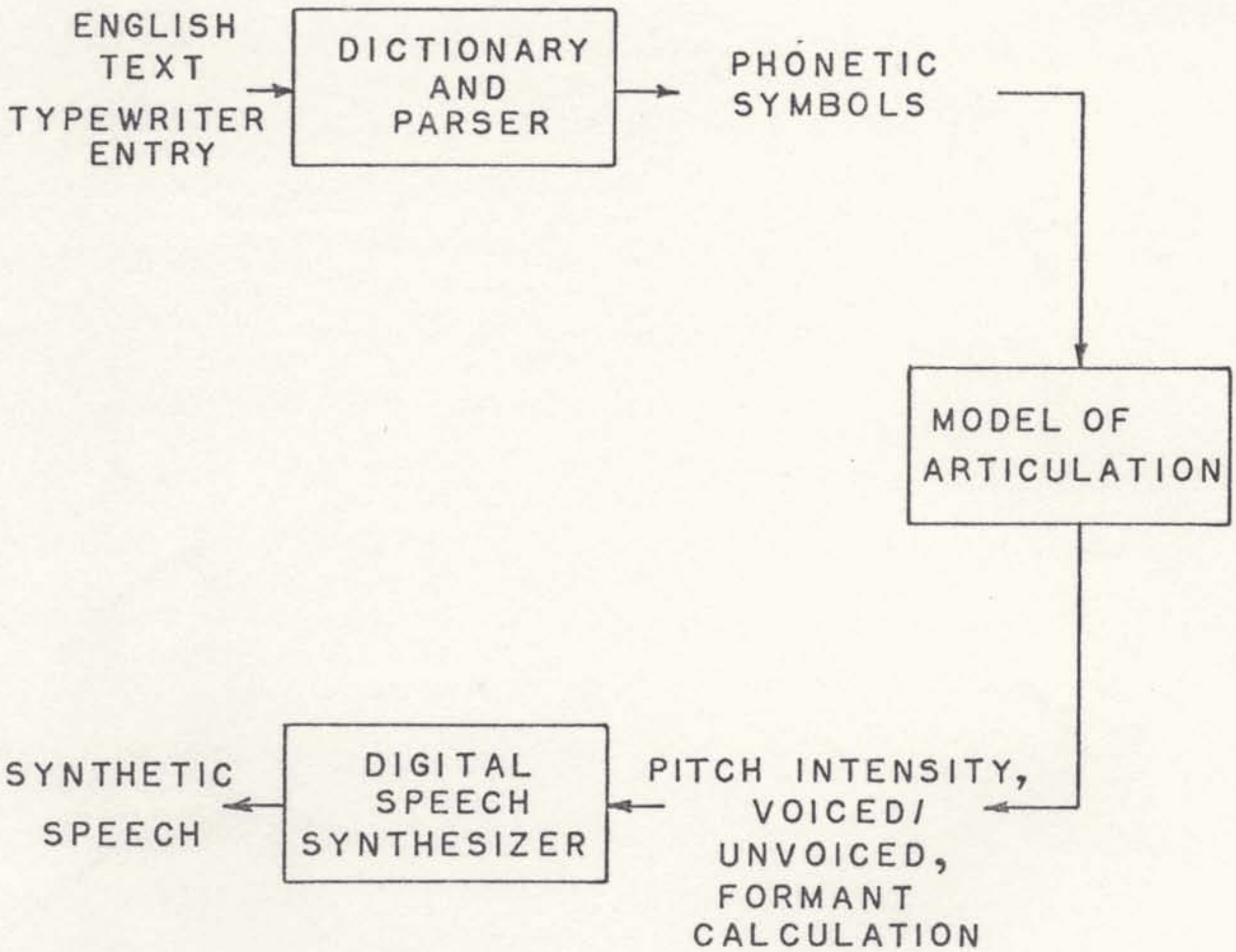


Fig. 12. Block Diagram of a Text Synthesizer

in a pronouncing dictionary, each entry of which has the English word and a phonetic transcription of the word.

When a word is to be output or "spoken," the controlling program outputs the sequence of codes to a synthesizer(such as a formant synthesizer) to be converted to an analog signal(Allen 1976). This type of text synthesizer has a limited vocabulary; however, it has the desired feature of converting English text directly into synthesized speech. Although the vocabulary is limited, over 700 words can be stored in 6.6 kilobytes of memory. This is the type of synthesizer under study in the next chapter.

In a more sophistocated form, a text synthesizer can have an unlimited vocabulary. It is able to take any English text, and through an appropriate set of rules, convert the text to speech. Such a synthesizer must therefore use similar rules that a human subconsciously applies when reading. An unrestricted text synthesizer has been built which operates with an accuracy of 90%(Elowitz, et al. 1976).

CHAPTER 5

MICROCOMPUTER CONTROLLED TEXT SYNTHESIZER

This chapter describes a microcomputer controlled, limited vocabulary text synthesizer. A Southwest Technical Products Corporation(SWTPC) 6800 microcomputer is used to control a Votrax(Federal Screw Works) speech synthesizer. The system has the capability of synthesizing speech using one of two methods: (1) via text supplied through a keyboard, or (2) by a table of codes stored in memory.

The keyboard method involves typing in a sentence or series of sentences. The computer looks up each word in a vocabulary dictionary and sends a string of codes to the Votrax unit which outputs the synthesized speech.

In the other method a table of codes must be stored in memory. To produce speech, the computer simply sends these codes to the synthesizer as before. This method enables the programmer to store messages which can be used in any program for prompting or responses by the computer.

Hardware

The heart of the system is the Votrax synthesizer module. This module is a sealed unit which contains circuitry to convert a 6-bit code into one of 64 phonemes or allophones. Thus by proper coding it has the capability of speaking English, Spanish, French, and many other languages, although the module has been optimized for English speech(The Digital Group 1978). Table 6 lists the phonemes in IPA and Votrax notation and gives the hexidecimal code for each. Note that in the Votrax notation some phonemes end in a number, for example A1. Phonemes with increasing number values decrease in duration. Thus A2 is of shorter duration than A1.

The Votrax module is mounted on a circuit board, the VOX-1, built by The Digital Group. This board was specifically designed to plug directly into a Z80 microcomputer I/O slot, however it can be used with any computer system which has an 8-bit parallel output port and an 8-bit parallel input port. The SWTPC 6800 has parallel I/O ports via a Motorola MC6820 peripheral interface adaptor(PIA); thus the board required no modifications.

In addition to the Votrax module, the board has two FIFO(first in-first out) memory buffer IC's(3351-2) which reduce processor overhead, a single chip audio amplifier(LM384) to drive a speaker and a dual retriggerable one shot(74L123) to indicate the buffer status to the microprocessor. The board also includes a limited speech recognition circuit. Figure 13 shows a schematic of the speech synthesis portion of the VOX-1 board.

It works as follows: hex phoneme codes(V-codes) are sent to the memory buffer IC's via a parallel 8-bit output port. Since the hex codes are 6 bits, the two extra bits are used for buffer control. Bit 7 is used to strobe codes into the buffer, and bit 6 is a master reset. The buffers hold up to 80 codes. These codes propagate

TABLE 6

VOTRAX PHONETIC CODES

VOTRAX	IPA	HEX CODE	VOTRAX	IPA	HEX CODE
PA0	Pause	83	IU		B6
PA1		BE	J	dζ	9A
A	e	A0	K	k	99
A1		86	L	l	98
A2		85	M	m	8c
AE	æ	AE	N	n	8D
AE1		AF	NG	ŋ	94
AH	a	A4	O	o	A6
AH1		95	O1		B5
AH2		88	O2		B4
AW	ɔ	BD	OO	U	97
AW1		93	OO1		96
AW2		B0	P	p	A5
AY		A1	R	r	AB
B	b	8E	S	s	9F
CH	tʃ	90	SH	ʃ	91
D	d	9E	T	t	AA
DT		84	TH	θ	B9
E	i	AC	THV	ð	B8
E1		BC	U	u	A8
EH	ε	BB	U1		B7
EH1		82	UH	ʌ or ə	B3
EH2		81	UH1		B2
EH3		80	UH2		B1
ER	ɝ or ɚ	BA	UH3		A3
F	f	9D	V	v	8F
G	g	9C	W	w	AD
H	h	9B	Y	j	A9
I	I	AY	Y1		A2
I1		8B	Z	z	92
I2		8A	ZH	ζ	87
I3		89	SILENCE/EOM*		BF

*Hex BF is used in software to denote an end of spoken message.

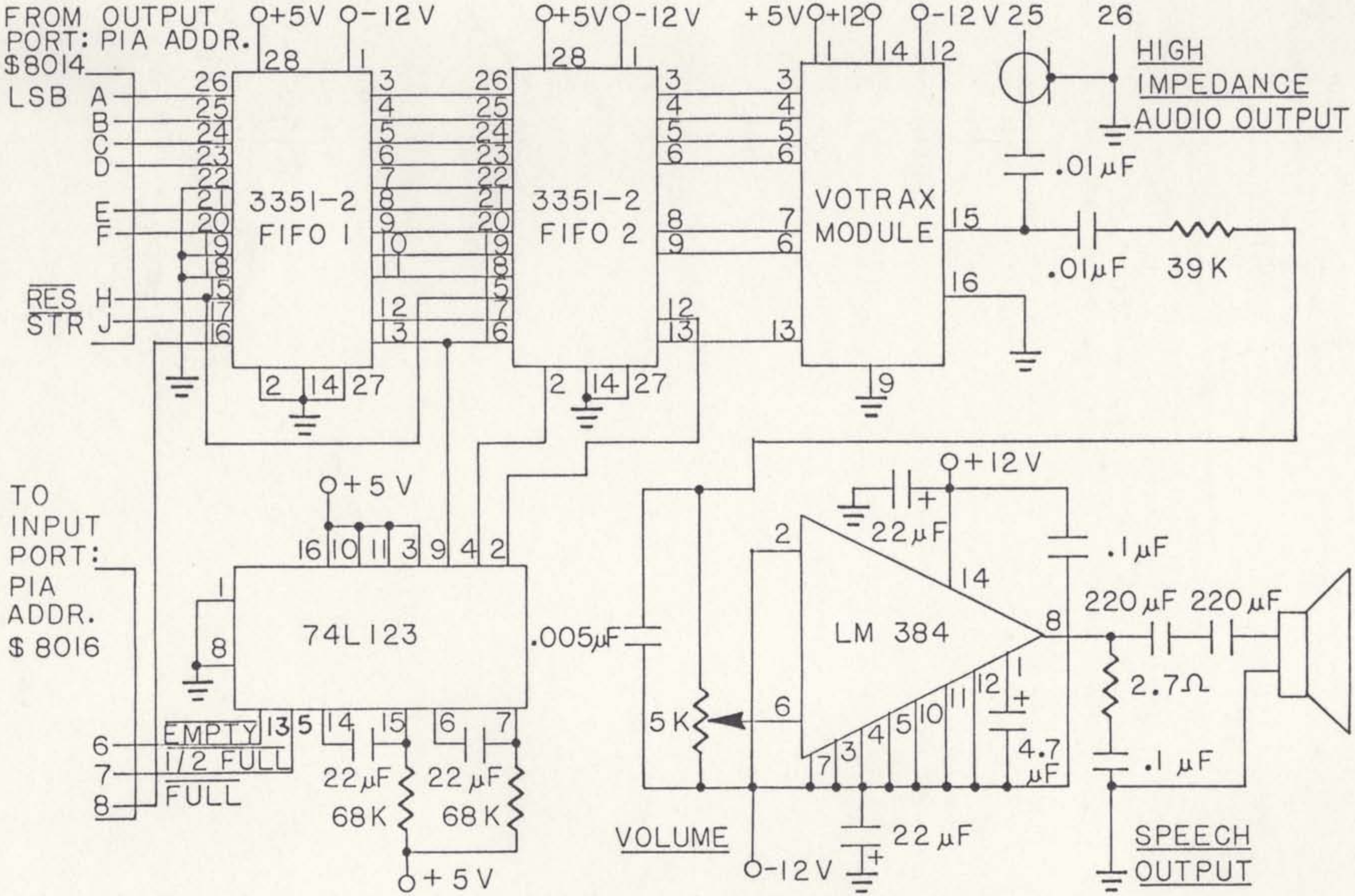


Fig. 13. Synthesizer Circuit of the VOX-1 Board
(Property of The Digital Group 1978)

through the buffers and are input to the Votrax module. The module then produces an audio output tone corresponding to the phoneme code and signals the buffers when it is ready for another phoneme.

The audio output of the Votrax module can either be used as an auxiliary input to an external audio amplifier or to drive the on-board audio amplifier. A 5 k ohm potentiometer is used to control the volume.

The dual retriggerable one-shot is used to indicate to the microprocessor, through the input port, whether or not the memory buffers are empty of half full. In addition, a line from FIFO 1 indicates whether or not the buffers are completely full. The microprocessor must monitor the input port and feed phoneme codes to the buffers whenever necessary(The Digital Group 1978).

The SWTPC 6800 computer system has a microcomputer with 32 k-bytes of memory and a dual mini floppy disk system. For input and output a Beehive Medical Electronics Video terminal and a Digital Equipment Corporation Decwriter II were used. The Decwriter enabled hard copies of assembly listings and program output to be obtained at a reasonable rate of 300 baud.

Software

The synthesizer software was designed by the author with two objectives in mind. The first was to find a method of checking out the VOX-1 board and Votrax speech synthesizer. The second objective was to produce a program which could be incorporated in the SWTPC 6800 system as a prompting and verbal message package.

The software can be divided into several parts. The main program is a monitor program which enables the user to choose between several options. Typing a "T" causes a previously stored text(in V-codes) to be spoken. If "L" is typed a list of words in the vocabulary, their memory addresses, and their V-codes are output. An "E" enables the user to input, via the keyboard, sentences or groups of sentences composed of words in the vocabulary list and immediately have them output as synthesized speech. Typing a "D" causes control to be returned to the disk operating system.

The voice message subroutine, VMSG, is responsible for the actual control of the Votrax synthesizer. This subroutine uses an address supplied via the index register, as the starting address of a table of hex V-codes. It sends these codes to the VOX-1 buffers and polls the buffer control lines in order to stop transmission should the buffers become full. VMSG can be used to output synthesized speech of a message stored in hex V-codes in a table anywhere in memory. This is done by simply loading the index register with the address of the first entry and then jumping to the subroutine. The table of V-codes must end with a $BF(a $ is used to denote a hex number) to signify the end of message.

A dictionary of words is located in memory starting at $100. Appendix A gives a 6800 assembly listing of the dictionary. Note that each letter of the alphabet has a 256 byte block of memory allocated to it. This can be called a letter block. The dictionary can be expanded easily by inserting words and codes until a letter block is filled. The starting address of each letter block increases by $100 which allows quick processor access. For example in the keyboard subroutine when a word is typed in, the first letter(ASCII) of the word is taken and the two most significant bits are masked out. Then two zeros are catenated to the end to form the letter block starting address. Thus if the word "at" is entered, the first 2 bits of the ASCII code for "a", $41(binary 0100 0001), are masked off leaving $01. When the two zeros are catenated an address of $0100 is given as the start of the "a" block. In this way, each letter block is accessed without going through preceding letter blocks. Each letter block ends with a $BF to signify its end. This method of access provides efficient use of processor time, especially if the dictionary is full.

Table 7 presents an ASCII code conversion chart for ASCII codes with the parity bit(bit 7) equal to zero. By referring back to Table 6 it is seen that all V-codes are greater than $7F. ASCII codes are less than or equal to $7F, thus the processor can distinguish between the ASCII codes and V-codes for words in the dictionary.

The "L" monitor command causes the address, words, and V-codes of all the dictionary words to be printed. A portion of such a listing is shown in Figure 14.

The most complicated subroutine is subroutine ENGLSH which accepts text input from a keyboard and converts it to speech. This subroutine stores the input sentence in a table in ASCII form. When

```
L
0100  A 86  85  A1  A9
0105  ABLE 85  85  A9  8E  98
010E  ABOUT B1  8E  95  96  AA
0118  ADDRESS AE  80  9E  AB  82  9F
0125  ADVANCE 81  9E  8F  AF  81  8D  9F
0133  AFTER AE  9D  AA  BA
013C  AGAIN B2  9C  BB  8D
0145  ALL BD  A3  98
014B  ALPHABET AF  B1  98  9D  B2  8E  BB  AA
015B  AM 81  82  80  8C  8C
0162  AN AF  80  89  8D
0168  AND AF  80  89  8D  9E
0170  ANY 82  82  8D  8D  BC  BC
0179  ARE 95  A3  BA
017F  AS AE  AF  92
0184  AT AF  80  AA
0189  AVAILABLE B1  8F  86  A1  98  A3  8E  A3  98
0200  B 8E  BC  AC
0204  BAD 8E  AE  9E
020A  BE 8E  AC  AC
  |     |      |
  |     |      |
  |     |      |
1727  WHEN AD  82  8D
172E  WHERE AD  81  85  AB
1737  WHICH AD  8B  AA  90
1740  WILL AD  8A  A3  98
1748  WILLING AD  8A  A3  98  8B  94
1755  WITH AD  8A  B9
175C  WORKING AD  BA  AB  99  AC  94
1769  WOULD AD  B7  B6  B6  9E
1800  X 81  82  89  99  83  9F
1807  XRAY 80  99  9F  AB  A1  A1
1900  Y AD  95  80  89  A9
1906  YALL A9  BD  A3  98
190E  YES A9  82  80  9F
1915  YOU A9  B6  A8
1A00  Z 92  BC  AC
1A04  ZERO 92  A1  89  AB  A3  B4  B7

      NUMBERS 0 THROUGH 9 ARE ALSO AVAILABLE
```

Fig. 14. Partial Results of the "L" Command

TABLE 7

ASCII CODE CONVERSION CHART
(PARITY BIT ZERO)

BITS 4 thru 6	-	0	1	2	3	4	5	6	7
	0	NUL	DLE	SP	0	@	P		p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
BITS 0 thru 3	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	'	7	G	W	g	w
	8	BS	CAN	(	8	H	X	h	x
	9	HT	EM	)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[	k	{
	C	FF	FS	,	<	L	/	l	/
	D	CR	GS	-	=	M	]	m	}
	E	SO	RS	.	>	N	^	n	≈
	F	SI	US	/	?	O	—	o	DEL

input, each word must be separated by a space or punctuation mark (, . ?). These denote word boundaries. Each word is looked at by the processor in the sequence entered. As shown above, the first letter determines the letter block. The processor compares the words within this block with the entered word. If the input word matches a word in the dictionary, the V-codes are stored in a table. If the input word is not in the vocabulary an error message is output on the monitor.

The user denotes the end of an entry by hitting the ESC key. If all words entered are in the dictionary the speech is synthesized by outputting the table of V-codes via subroutine VMSG.

Some of the features of subroutine ENGLSH are:

1. More than one line can be entered as long as each line ends with a space or punctuation, and a carriage return(CR) and line feed(LF)

2. Mistakes in text entry can be deleted via the "RUBOUT" key. The computer will respond with a "#" for each rubout, and the ASCII characters will not be stored in memory

3. A punctuation mark must precede ESC

4. Numbers are treated as words, thus there must be a space or punctuation mark between each single number

5. More than one space or punctuation mark, or adjacent space and punctuation mark between words are not allowed

Appendix B gives a complete 6800 assembly listing for the synthesizer software. Note that the word dictionary and operating program are separated by a large block of memory. This is because the disk operating system software resides in this space. Also observe that there are three tables; ATABLE, VTABLE, and TTABLE, each consisting of a block of 500 bytes. This is to facilitate large text entries. ASCII keyboard entries are stored in ATABLE, V-codes are stored in VTABLE prior to output to the Votrax module, and text is stored in TTABLE. These tables could be reduced considerably if large text entries are not anticipated.

Results

The speech synthesizer system tested works very well and gives good quality speech output. Figure 15 shows a sequence which demonstrates the features of the system software. In (a) the disk operating system is initiated when in the SWTPC monitor mode(denoted by $) by typing a "D." When the disk operating system(FDOS) is ready, the word dictionary, WORDS, is loaded and the synthesis program(VOTRAX) is run. The computer speaks a prompting message, "Enter T, L, D, or E."

When an "E" is entered, the computer responds with, "ENTER SENTENCE. HIT ESCAPE." on the monitor. In (d) a sentence is entered. Note that words as well as numbers are separated by a space or a punctuation mark which denotes a word boundary. When the "ESC" key is pressed the sentence is synthesized by the computer.

Entry (e) shows how a rubout is used to correct a typing error in the word "the." When a word is not in the dictionary of words the computer will respond with an error message as in (f) where the word address is spelled incorrectly. This is also demonstrated in (g).

The software automatically places a pause between words, however extra pauses may be desired. These are denoted as P1 and P2, and their use is shown in (h) and (i). P1 is equivalent to a PA1

```
a)      $D SWTPC V1.0 (C) 1977

        FDOS READY
b)      LOAD WORDS

        FDOS READY
c)      RUN VOTRAX

        E
d)      ENTER SENTENCE. HIT ESCAPE .
        THE ADDRESS IS 1 2 A F.
        E
e)      ENTER SENTENCE. HIT ESCAPE .
        TEHE ADDRESS IS 1 2 A F.
        E
f)      ENTER SENTENCE. HIT ESCAPE .
        THE ADRESS IS 1 2 A F.
         ADRESS IS NOT IN VOCABULARY.

        E
g)      ENTER SENTENCE. HIT ESCAPE .
        THE ADDRESS ID 1 2 A F.
         ID IS NOT IN VOCABULARY.

        E
h)      ENTER SENTENCE. HIT ESCAPE .
        THE ADDRESS IS 1 P1 2 P1 A P1 F.
        E
i)      ENTER SENTENCE. HIT ESCAPE .
        THE ADDRESS IS 1 P2 2 P2 A P2 F.
j)      I
        I
        E
k)      ENTER SENTENCE. HIT ESCAPE .
        I AM A SIXTY 8 HUNDRED MICRO COMPUTER.
        E
l)      ENTER SENTENCE. HIT ESCAPE .
        I AM A VOTRAX VOICE SYNTHESIZER.
        I AM TALKING TO YOU NOW.
        DO YOU UNDERSTAND?
        E
m)      ENTER SENTENCE. HIT ESCAPE .
        HA HA THAT IS A GOOD ONE.
        E
n)      ENTER SENTENCE. HIT ESCAPE .
        HA HA THAT IS A GOOD 1.
o)      DSWTPC V1.0 (C) 1977

        FDOS READY
```

Fig. 15. Illustration of Program Features

shown in TABLE 6 and a P2 is twice the duration of P1.

If a letter other than T, L, D, or E is entered the computer responds with a verbal message, "Error! You must enter T, L, D, or E." This occurs in (j) where "I" is entered.

More than one line can be entered by using a carriage return (CR) and line feed (LF) at the end of a line. The CR and LF are ignored, and to the computer the entry looks like one continuous line. The multiple line entry feature is shown in (l).

Entries (m) and (n) demonstrate that a word can be given different symbols, but will still sound the same since the V-codes are the same.

In (o), the "D" key has been pressed. This returns control to the disk operating system. The listing command, "L," was shown previously in Figure 14. The text "T" command cannot be shown since all output is verbal.

The synthesized speech can be understood very well. It is basically monotonic since the Votrax module used did not have pitch or intensity controls.

There were some problems with the "p" and "q" phoneme sounds, which tended to degrade the speech quality. The "p" sound is synthesized as a "t." Since "p" and "b" are in the same class and are very similar except for duration, it was felt that substitution of the "b" phoneme might be an improvement over the "t" sound. This, however, was not the case.

The "q" sound was also of poor quality. A "q" is made up of "k + w." When used in a word the synthesized sound is that of a "tw." Thus the word "quite" sounds like "twite." This is apparent in every case where a "q" sound is desired. A "g" sound was substituted for the "k," but did not improve the "q" sound.

Since the "p" and "q" sounds are characteristic of the Votrax module, no immediate solution to the problem was found. The Votrax circuit itself would have to be modified, and since the circuit is in a sealed unit, this is not a feasible solution.

CONCLUSIONS

Many different kinds of speech synthesizers are available today. This report has described several different types, and a hardware and software realization of a limited text synthesizer has been presented. This system has shown that very good quality speech synthesis can be achieved on a small scale computer. The number of words in a vocabulary is constrained only by memory size. The system described has the capability of over 700 words in 6.6 k-bytes of memory. In many applications only certain words or phrases are required, and very little memory is utilized, since these can be stored in V-code form in memory tables. The routine to output a message is very short (62 bytes) and can easily be stored in a ROM to be incorporated as a permanent feature of a small scale computer system.

The application of speech synthesis in many different areas is immediately apparent. The speed of the computer in accessing data coupled with a voice output makes telephone information systems a possibility. A Touchtone keyboard can be used as an input device, and the computer can provide requested data by speaking over the phone. There are many applications in the aviation industry, including air traffic control and on-board aircraft warning devices. Likewise, such verbal warnings might be used in automobiles to signal faults in vital systems such as brakes. Another speech synthesizer use includes training and learning devices which incorporate a processor to evaluate performance and give verbal prompting, encouragement, or correction to the trainee.

Essentially, any area which calls for verbal information output coupled with the processing capabilities of a computer, is a prime candidate for speech synthesis. The reduction in the physical size and increased performance of computers as well as synthesizers points to an exciting era of speaking machines in the near future.

APPENDIX A

TEXT SYNTHESIZER VOCABULARY
ASSEMBLY LISTING

```
                 NAM    WORDS
                 OPT    NOG
                 OPT    NOP
0100             ORG    $100
0100 41   WTABLE FCC    /A/

0101 86          FCB    $86,$85,$A1,$A9
0105 41          FCC    /ABLE/

0109 85          FCB    $85,$85,$A9,$8E,$98
010E 41          FCC    /ABOUT/

0113 B1          FCB    $B1,$8E,$95,$96,$AA
0118 41          FCC    /ADDRESS/

011F AE          FCB    $AE,$80,$9E,$AB,$82,$9F
0125 41          FCC    /ADVANCE/

012C 81          FCB    $81,$9E,$8F,$AF,$81,$8D,$9F
0133 41          FCC    /AFTER/

0138 AE          FCB    $AE,$9D,$AA,$BA
013C 41          FCC    /AGAIN/

0141 B2          FCB    $B2,$9C,$BB,$8D
0145 41          FCC    /ALL/

0148 BD          FCB    $BD,$A3,$98
014B 41          FCC    /ALPHABET/

0153 AF          FCB    $AF,$B1,$98,$9D,$B2,$8E,$BB,$AA
015B 41          FCC    /AM/

015D 81          FCB    $81,$82,$80,$8C,$8C
0162 41          FCC    /AN/

0164 AF          FCB    $AF,$80,$89,$8D
0168 41          FCC    /AND/

016B AF          FCB    $AF,$80,$89,$8D,$9E
0170 41          FCC    /ANY/

0173 82          FCB    $82,$82,$8D,$8D,$BC,$BC
0179 41          FCC    /ARE/

017C 95          FCB    $95,$A3,$BA
017F 41          FCC    /AS/

0181 AE          FCB    $AE,$AF,$92
0184 41          FCC    /AT/

0186 AF          FCB    $AF,$80,$AA
0189 41          FCC    /AVAILABLE/

0192 B1          FCB    $B1,$8F,$86,$A1,$98,$A3,$8E,$A3,$98
019B BF          FCB    $BF
0200             ORG    $200
0200 42          FCC    /B/

0201 8E          FCB    $8E,$BC,$AC
0204 42          FCC    /BAD/

0207 8E          FCB    $8E,$AE,$9E
020A 42          FCC    /BE/

020C 8E          FCB    $8E,$AC,$AC
020F 42          FCC    /BEGIN/

0214 8E          FCB    $8E,$BC,$9C,$89,$8B,$8D
021A 42          FCC    /BIEN/

021E 8E          FCB    $8E,$BC,$80,$82,$8D
0223 42          FCC    /BUT/

0226 8E          FCB    $8E,$B2,$A3,$AA
022A 42          FCC    /BY/

022C 8E          FCB    $8E,$95,$BC
022F BF          FCB    $BF
0300             ORG    $300
0300 43          FCC    /C/

0301 9F          FCB    $9F,$BC,$AC
0304 43          FCC    /CAN/

0307 99          FCB    $99,$AE,$80,$80,$8D
030C 43          FCC    /CODE/

0310 99          FCB    $99,$B4,$B5,$B7,$9E
0315 43          FCC    /COME/

0319 99          FCB    $99,$A3,$A3,$A3,$8C,$8C
031F 43          FCC    /COMPUTE/

0326 99          FCB    $99,$A3,$8C,$A5,$A9,$84,$AA
032D 43          FCC    /COMPUTER/

0335 99          FCB    $99,$A3,$8C,$A5,$A9,$84,$AA,$BA
033D 43          FCC    /COUNT/

0342 99          FCB    $99,$AF,$B2,$8D,$AA
0347 BF          FCB    $BF
0400             ORG    $400
```

```
0400 44       FCC    /D/
0401 9E       FCB    $9E,$BC,$AC
0404 44       FCC    /DATA/
0408 9E       FCB    $9E,$86,$A1,$84,$B1
040D 44       FCC    /DID/
0410 9E       FCB    $9E,$A7,$9E
0413 44       FCC    /DIGITAL/
041A 9E       FCB    $9E,$BB,$9A,$86,$9E,$A3,$98
0421 44       FCC    /DO/
0423 9E       FCB    $9E,$B6,$A8
0426 44       FCC    /DOES/
042A 9E       FCB    $9E,$A3,$92
042D BF       FCB    $BF
0500          ORG    $500
0500 45       FCC    /E/
0501 BC       FCB    $BC,$BC,$BC
0504 45       FCC    /ENTER/
0509 82       FCB    $82,$8D,$AA,$BA
050D 45       FCC    /ERROR/
0512 BB       FCB    $BB,$BA,$A6,$B4,$BA
0517 45       FCC    /EXCEPT/
051D 81       FCB    $81,$99,$9F,$82,$A5,$83,$AA
0524 45       FCC    /EXPECT/
052A 82       FCB    $82,$89,$99,$83,$9F,$A5,$BB,$99,$AA
0533 BF       FCB    $BF
0600          ORG    $600
0600 46       FCC    /F/
0601 81       FCB    $81,$82,$B1,$9D
0605 46       FCC    /FIND/
0609 9D       FCB    $9D,$95,$89,$BC,$8D,$9E
060F 46       FCC    /FOR/
0612 9D       FCB    $9D,$A6,$AB
0615 46       FCC    /FRANCAIS/
061D 9D       FCB    $9D,$A4,$8D,$9F,$80,$86,$A1
0624 46       FCC    /FRENCH/
062A 9D       FCB    $9D,$AB,$BB,$8D,$90
062F 46       FCC    /FROM/
0633 9D       FCB    $9D,$AB,$B2,$8C
0637 BF       FCB    $BF
0700          ORG    $700
0700 47       FCC    /G/
0701 9E       FCB    $9E,$9A,$BC,$AC
0705 47       FCC    /GET/
0708 9C       FCB    $9C,$82,$81,$AA
070C 47       FCC    /GIVE/
0710 9C       FCB    $9C,$A7,$8F,$8F
0714 47       FCC    /GO/
0716 9C       FCB    $9C,$B4,$B4
0719 47       FCC    /GOOD/
071D 9C       FCB    $9C,$A6,$B6,$9E
0721 BF       FCB    $BF
0800          ORG    $800
0800 48       FCC    /H/
0801 85       FCB    $85,$85,$85,$A1,$A9,$AA,$90
0808 48       FCC    /HA/
080A 9B       FCB    $9B,$95,$88
080D 48       FCC    /HAD/
0810 9B       FCB    $9B,$AE,$80,$9E
0814 48       FCC    /HAS/
0817 9B       FCB    $9B,$AE,$89,$92
081B 48       FCC    /HAVE/
081F 9B       FCB    $9B,$AE,$A3,$8F
0823 48       FCC    /HELLO/
0828 9B       FCB    $9B,$82,$A3,$98,$A3,$B4,$B7
082F 48       FCC    /HERE/
0833 9B       FCB    $9B,$AC,$AB
0836 48       FCC    /HERO/
083A 9B       FCB    $9B,$AC,$AB,$A3,$B4,$B7
0840 48       FCC    /HM/
0842 9B       FCB    $9B,$A3,$8C,$8C,$8C
0847 48       FCC    /HOW/
084A 9B       FCB    $9B,$95,$B4,$B7
```

```
0B4E 4B          FCC    /HUNDRED/

0855 9B          FCB    $9B,$B2,$8D,$9E,$AB,$81,$9E
085C BF          FCB    $BF
0900             ORG    $900
0900 49          FCC    /I/

0901 95          FCB    $95,$88,$80,$89,$BC
0906 49          FCC    /IF/

0908 A7          FCB    $A7,$9D,$BE
090B 49          FCC    /IN/

090D A7          FCB    $A7,$8D,$BE
0910 49          FCC    /INFORMATION/

091B 8A          FCB    $8A,$8D,$9D,$AB,$8C,$86,$A1,$91
0923 A3          FCB    $A3,$8D
0925 49          FCC    /INITIAL/

092C 8A          FCB    $8A,$8D,$8B,$91,$A3,$98
0932 49          FCC    /IS/

0934 A7          FCB    $A7,$92,$BE
0937 49          FCC    /IT/

0939 8B          FCB    $8B,$AA,$BE
093C BF          FCB    $BF
0A00             ORG    $A00
0A00 4A          FCC    /J/

0A01 9E          FCB    $9E,$9A,$80,$A0,$A1,$A1
0A07 4A          FCC    /JE/

0A09 87          FCB    $87,$B3
0A0B 4A          FCC    /JUST/

0A0F 9E          FCB    $9E,$9A,$B2,$9F,$AA
0A14 BF          FCB    $BF
0B00             ORG    $B00
0B00 4B          FCC    /K/

0B01 99          FCB    $99,$80,$A0,$A1
0B05 4B          FCC    /KEEP/

0B09 99          FCB    $99,$BC,$A9,$A5
0B0D 4B          FCC    /KIND/

0B11 99          FCB    $99,$95,$BC,$8D,$9E
0B16 4B          FCC    /KNOW/

0B1A 8D          FCB    $8D,$A3,$B5,$B7
```

```
0B1E BF          FCB    $BF
0C00             ORG    $C00
0C00 4C          FCC    /L/

0C01 81          FCB    $81,$82,$80,$98,$98
0C06 4C          FCC    /LEAVE/

0C0B 98          FCB    $98,$BC,$A2,$8F
0C0F 4C          FCC    /LIKE/

0C13 98          FCB    $98,$88,$BC,$99
0C17 4C          FCC    /LIST/

0C1B 98          FCB    $98,$8B,$9F,$AA
0C1F 4C          FCC    /LISTEN/

0C25 98          FCB    $98,$8B,$9F,$82,$8D
0C2A 4C          FCC    /LONG/

0C2E 98          FCB    $98,$A4,$A3,$94
0C32 BF          FCB    $BF
0D00             ORG    $D00
0D00 4D          FCC    /M/

0D01 81          FCB    $81,$82,$80,$8C,$8C
0D06 4D          FCC    /MADE/

0D0A 8C          FCB    $8C,$86,$A1,$A2,$9E
0D0F 4D          FCC    /MACHINE/

0D16 8C          FCB    $8C,$B1,$91,$AC,$8D
0D1B 4D          FCC    /MAY/

0D1E 8C          FCB    $8C,$80,$86,$A1
0D22 4D          FCC    /ME/

0D24 8C          FCB    $8C,$89,$BC,$BC
0D28 4D          FCC    /MICRO/

0D2D 8C          FCB    $8C,$95,$89,$99,$AB,$B2,$B5
0D34 4D          FCC    /MISTAKE/

0D3B 8C          FCB    $8C,$89,$9F,$AA,$86,$A1,$A2,$99
0D43 4D          FCC    /MORE/

0D47 8C          FCB    $8C,$B5,$B5,$AB
0D4B 4D          FCC    /MUST/

0D4F 8C          FCB    $8C,$B5,$9F,$AA
0D53 4D          FCC    /MY/

0D55 8C          FCB    $8C,$95,$89,$BC
```

```
0D59 BF          FCB     $BF
0E00             ORG     $E00
0E00 4E          FCC     /N/

0E01 82          FCB     $82,$82,$8D,$8D
0E05 4E          FCC     /NAME/

0E09 8D          FCB     $8D,$86,$A1,$A1,$8C
0E0E 4E          FCC     /NO/

0E10 8D          FCB     $8D,$B2,$B5
0E13 4E          FCC     /NOT/

0E16 8D          FCB     $8D,$95,$AA
0E19 4E          FCC     /NOW/

0E1C 8D          FCB     $8D,$A3,$95,$B4,$B7
0E21 4E          FCC     /NUMBER/

0E27 8D          FCB     $8D,$B2,$8C,$8E,$BA
0E2C BF          FCB     $BF
0F00             ORG     $F00
0F00 4F          FCC     /O/

0F01 96          FCB     $96,$B4,$B5,$A8
0F05 4F          FCC     /OF/

0F07 B2          FCB     $B2,$8F,$BE
0F0A 4F          FCC     /ON/

0F0C A4          FCB     $A4,$A3,$8D,$8D
0F10 4F          FCC     /ONE/

0F13 AD          FCB     $AD,$B3,$8D
0F16 4F          FCC     /ORDER/

0F1B A6          FCB     $A6,$B4,$BA,$9E,$BA
0F20 4F          FCC     /OVER/

0F24 B5          FCB     $B5,$B7,$8F,$BA
0F28 BF          FCB     $BF
1000             ORG     $1000
1000 50          FCC     /P/

1001 A5          FCB     $A5,$AC,$A1,$A9
1005 50          FCC     /PARLAIS/

100C A5          FCB     $A5,$95,$AB,$98,$80,$86,$A1
1013 50          FCC     /PARLE/

1018 A5          FCB     $A5,$95,$AB,$98
101C 50          FCC     /PART/

1020 A5          FCB     $A5,$95,$AB,$AA
1024 50          FCC     /PLACE/

1029 A5          FCB     $A5,$98,$86,$A1,$A2,$9F
102F 50          FCC     /PLEASE/

1035 A5          FCB     $A5,$98,$BC,$BC,$92
103A 50          FCC     /PLUS/

103E A5          FCB     $A5,$9B,$98,$B2,$9F
1043 50          FCC     /PRETTY/

1049 A5          FCB     $A5,$AB,$8B,$AA,$A9
104E 50          FCC     /PROGRAM/

1055 A5          FCB     $A5,$AB,$B5,$9C,$AB,$AF,$80,$8C
105D 50          FCC     /P1/

105F BE          FCB     $BE
1060 50          FCC     /P2/

1062 BE          FCB     $BE,$BE
1064 BF          FCB     $BF
1100             ORG     $1100
1100 51          FCC     /Q/

1101 99          FCB     $99,$A9,$B6,$B6,$B7
1106 51          FCC     /QUANTITY/

110E 99          FCB     $99,$AD,$88,$80,$8D,$84,$8A,$84,$A9
1117 51          FCC     /QUESTION/

111F 99          FCB     $99,$AD,$82,$9F,$AA,$90,$A3,$8D
1127 BF          FCB     $BF
1200             ORG     $1200
1200 52          FCC     /R/

1201 A4          FCB     $A4,$A3,$BA
1204 52          FCC     /RAISE/

1209 AB          FCB     $AB,$86,$86,$A1,$92
120E 52          FCC     /READ/

1212 AB          FCB     $AB,$80,$89,$9E
1216 52          FCC     /RECITE/

121C AB          FCB     $AB,$BC,$9F,$A4,$BC,$AA
1222 52          FCC     /REPEAT/

1228 AB          FCB     $AB,$BC,$A5,$BC,$AA
```

```
122D 52         FCC     /RETURN/

1233 AB         FCB     $AB,$BC,$AA,$BA,$AB,$8D
1239 52         FCC     /REVERSE/

1240 AB         FCB     $AB,$BC,$8F,$BA,$BA,$9F
1246 BF         FCB     $BF
1300            ORG     $1300
1300 53         FCC     /S/

1301 81         FCB     $81,$82,$89,$9F,$9F
1306 53         FCC     /SAME/

130A 9F         FCB     $9F,$80,$86,$A1,$A2,$8C
1310 53         FCC     /SAVE/

1314 9F         FCB     $9F,$80,$86,$A1,$A2,$8F
131A 53         FCC     /SAY/

131D 9F         FCB     $9F,$80,$86,$A1
1321 53         FCC     /SINCE/

1326 9F         FCB     $9F,$A7,$8D,$9F
132A 53         FCC     /SIXTY/

132F 9F         FCB     $9F,$8B,$99,$9F,$AA,$A9
1335 53         FCC     /SO/

1337 9F         FCB     $9F,$B5,$B5
133A 53         FCC     /SOME/

133E 9F         FCB     $9F,$B3,$8C
1341 53         FCC     /SOMEONE/

1348 9F         FCB     $9F,$B3,$8C,$AD,$B3,$8D
134E 53         FCC     /SORRY/

1353 9F         FCB     $9F,$95,$A3,$BA,$A9,$A2
1359 53         FCC     /SOUND/

135E 9F         FCB     $9F,$95,$A3,$B7,$8D,$9D
1364 53         FCC     /SPEAK/

1369 9F         FCB     $9F,$A5,$AC,$99
136D 53         FCC     /SPEECH/

1373 9F         FCB     $9F,$A5,$AC,$90
1377 53         FCC     /SYNTHESIZER/

1382 9F         FCB     $9F,$A7,$8D,$B8,$82,$9F,$95,$89
138A BC         FCB     $BC,$92,$BA
138D 53         FCC     /SYSTEM/

1393 9F         FCB     $9F,$8A,$9F,$AA,$B1,$8C
1399 BC         FCB     $BC,$92,$BA
139C BF         FCB     $BF
1400            ORG     $1400
1400 54         FCC     /T/

1401 AA         FCB     $AA,$BC,$AC
1404 54         FCC     /TALK/

1408 AA         FCB     $AA,$93,$99
140B 54         FCC     /TALKING/

1412 AA         FCB     $AA,$93,$99,$8B,$94
1417 54         FCC     /TEEN/

141B AA         FCB     $AA,$BC,$BC,$8D
141F 54         FCC     /THAT/

1423 B8         FCB     $B8,$AF,$80,$AA
1427 54         FCC     /THE/

142A B8         FCB     $B8,$A3,$A3,$A3
142E 54         FCC     /THEN/

1432 B8         FCB     $B8,$82,$80,$8D
1436 54         FCC     /THERE/

143B B8         FCB     $B8,$81,$85,$AB
143F 54         FCC     /THIS/

1443 B8         FCB     $B8,$A7,$9F
1446 54         FCC     /THROUGH/

144D B9         FCB     $B9,$AB,$A8,$AD
1451 54         FCC     /TO/

1453 AA         FCB     $AA,$B6,$A8
1456 54         FCC     /TRES/

145A AA         FCB     $AA,$AB,$80,$86,$A1
145F 54         FCC     /TYPE/

1463 AA         FCB     $AA,$B1,$88,$BC,$A5
1468 BF         FCB     $BF
1500            ORG     $1500
1500 55         FCC     /U/

1501 A9         FCB     $A9,$B6,$B6,$B7,$B7
```

```
1506 55            FCC    /UNDERSTAND/

1510 B1            FCB    $B1,$BD,$9E,$AB,$9F,$AA,$AF,$80
1518 BD            FCB    $BD,$9E
151A 55            FCC    /UNLIMITED/

1523 B1            FCB    $B1,$BD,$98,$8B,$8C,$8A,$AA,$89,$9E
152C 55            FCC    /USE/

152F A2            FCB    $A2,$B6,$A8,$9F
1533 BF            FCB    $BF
1600               ORG    $1600
1600 56            FCC    /V/

1601 8F            FCB    $8F,$BC,$AC
1604 56            FCC    /VALID/

1609 8F            FCB    $8F,$AF,$A3,$98,$A3,$9E
160F 56            FCC    /VERY/

1613 8F            FCB    $8F,$BB,$AB,$A9
1617 56            FCC    /VOCABULARY/

1621 8F            FCB    $8F,$96,$A8,$99,$AE,$8E,$B6,$98,$81
162C 56            FCC    /VOICE/

1631 8F            FCB    $8F,$B5,$A3,$BC,$9F
1636 56            FCC    /VOTRAX/

163C 8F            FCB    $8F,$A6,$AA,$AB,$AE,$99,$83,$9F
1644 56            FCC    /VOUS/

1648 8F            FCB    $8F,$B6,$A8
164B BF            FCB    $BF
1700               ORG    $1700
1700 57            FCC    /W/

1701 9E            FCB    $9E,$B2,$8E,$98,$A2,$B6,$B7
1708 57            FCC    /WANT/

170C AD            FCB    $AD,$95,$A3,$8D,$AA
1711 57            FCC    /WELL/

1715 AD            FCB    $AD,$82,$A3,$98
1719 57            FCC    /WERE/

171D AD            FCB    $AD,$BA,$AB
1720 57            FCC    /WHAT/

1724 AD            FCB    $AD,$B2,$AA
1727 57            FCC    /WHEN/

172B AD            FCB    $AD,$82,$8D
172E 57            FCC    /WHERE/

1733 AD            FCB    $AD,$81,$85,$AB
1737 57            FCC    /WHICH/

173C AD            FCB    $AD,$8B,$AA,$90
1740 57            FCC    /WILL/

1744 AD            FCB    $AD,$8A,$A3,$98
1748 57            FCC    /WILLING/

174F AD            FCB    $AD,$8A,$A3,$98,$8B,$94
1755 57            FCC    /WITH/

1759 AD            FCB    $AD,$8A,$B9
175C 57            FCC    /WORKING/

1763 AD            FCB    $AD,$BA,$AB,$99,$AC,$94
1769 57            FCC    /WOULD/

176E AD            FCB    $AD,$B7,$B6,$B6,$9E
1773 BF            FCB    $BF
1800               ORG    $1800
1800 58            FCC    /X/

1801 81            FCB    $81,$82,$89,$99,$83,$9F
1807 58            FCC    /XRAY/

180B 80            FCB    $80,$99,$9F,$AB,$A1,$A1
1811 BF            FCB    $BF
1900               ORG    $1900
1900 59            FCC    /Y/

1901 AD            FCB    $AD,$95,$80,$89,$A9
1906 59            FCC    /YALL/

190A A9            FCB    $A9,$BD,$A3,$98
190E 59            FCC    /YES/

1911 A9            FCB    $A9,$82,$80,$9F
1915 59            FCC    /YOU/

1918 A9            FCB    $A9,$B6,$A8
191B BF            FCB    $BF
1A00               ORG    $1A00
1A00 5A            FCC    /Z/

1A01 92            FCB    $92,$BC,$AC
1A04 5A            FCC    /ZERO/

1A08 92            FCB    $92,$A1,$89,$AB,$A3,$B4,$B7
1A0F BF            FCB    $BF
                   END
```

APPENDIX B

TEXT SYNTHESIZER PROGRAM
ASSEMBLY LISTING

```
                    NAM     VOTRAX
                    OPT     NOG
3B00                ORG     $3B00
3B00 00      ATABLE FCB     0
3D00                ORG     $3D00
3D00 00      VTABLE FCB     0
3F00                ORG     $3F00
3F00 00      TTABLE FCB     0
4100                ORG     $4100
4100 82      MONSSG FCB     $82,$8D,$AA,$BA,$83  ENTER
4105 AA             FCB     $AA,$BC,$AC,$83  T
4109 81             FCB     $81,$82,$80,$98,$98,$83  L
410F 9E             FCB     $9E,$BC,$AC,$83  D
4113 A6             FCB     $A6,$B4,$BA,$83  OR
4117 BC             FCB     $BC,$BC,$BC,$BF  E
411B BB      MSG1   FCB     $BB,$BA,$A6,$B4,$BA,$83,$83  ERROR
4122 A9             FCB     $A9,$B6,$AB,$83  YOU
4126 8C             FCB     $8C,$B2,$9F,$AA,$BF  MUST
412B 0014    TEMP   RMB     20
413F 00      XADR   FCB     0,0
4141 00      SAVA   FCB     0,0
4143 0002    SAVX   RMB     2
4145 00      TCODE  FCB     0
4146 0002    AX     RMB     2
4148 0002    TX     RMB     2
414A 0002    CX     RMB     2
414C 0002    BX     RMB     2
414E 0002    VX     RMB     2
4150 92      ZERO   FCB     $92,$A1,$89,$AB,$A3,$B4,$B7,$BF
4158 AD      ONE    FCB     $AD,$A3,$B1,$B1,$8D,$8D,$BE,$BF
4160 AA      TWO    FCB     $AA,$B6,$B6,$B7,$B7,$BE,$BE,$BF
4168 B9      THREE  FCB     $B9,$AB,$AC,$A9,$BE,$BE,$BE,$BF
4170 9D      FOUR   FCB     $9D,$B5,$B5,$AB,$BE,$BE,$BE,$BF
4178 9D      FIVE   FCB     $9D,$88,$95,$89,$A9,$8F,$8F,$BF
4180 9F      SIX    FCB     $9F,$8B,$89,$99,$83,$9F,$BE,$BF
4188 9F      SEVEN  FCB     $9F,$80,$81,$8F,$81,$8D,$8D,$BF
4190 85      EIGHT  FCB     $85,$85,$A1,$A9,$AA,$BE,$BE,$BF
4198 8D      NINE   FCB     $8D,$95,$81,$A9,$8D,$BE,$BE,$BF
41A0 0D      MSG2   FCB     $D,$A
41A2 45             FCC     /ENTER SENTENCE. HIT ESCAPE./

41BD 0D             FCB     $D,$A,4
41C0 20      MSG3   FCC     / IS NOT IN VOCABULARY./

41D6 0D             FCB     $D,$A,$4
41D9 0D      MSG4   FCB     $D,$A
41DB 20             FCC     /       NUMBERS 0 THROUGH 9/

41F4 20             FCC     / ARE ALSO AVAILABLE./

4208 0D             FCB     $D,$A,4

                    *MONITOR PROGRAM
                    *OPTIONS:
                    *T OUTPUTS TEXT LOCATED IN TTABLE
                    *L OUTPUTS LIST OF WORDS AND CODES
                    *E ENGLISH SENTENCE INPUT IS SPOKEN.
                    *D RETURNS CONTROL TO DOS.
4300                ORG     $4300       00000230
4300 CE FF04 BEGIN  LDX     #$FF04      00000234
4303 FF 8014        STX     $8014       INITIALIZE OUT PORT
4306 CE 0004        LDX     #$0004      00000242
4309 FF 8016        STX     $8016       INITIALIZE IN PORT
430C 86 00          LDA A   #0          00000250
430E B7 8014        STA A   $8014       00000254
4311 8E 3A00        LDS     #$3A00      INITIALIZE STACK
4314 86 0D          LDA A   #$D         00000262
4316 BD E1D1        JSR     $E1D1       OUTPUT CR
4319 86 0A          LDA A   #$A         00000270
431B BD E1D1        JSR     $E1D1       OUTPUT LF
431E CE 4100        LDX     #MONMSG      00000278
4321 BD 435E        JSR     VMSG        SPEAK MSG
4324 BD E1AC        JSR     $E1AC       GET CHAR
4327 81 44   DTEST  CMP A   #$44        D?
4329 26 03          BNE     TTEST       00000294
432B 7E 2400        JMP     $2400       00000298
432E 81 54   TTEST  CMP A   #$54        COMPARE INPUT WITH T CODE
4330 26 0C          BNE     LTEST       00000306
4332 CE 3F00        LDX     #TTABLE      INDEX GETS TAB ADDR.
4335 BD 435E        JSR     VMSG        SPEAK MSG
4338 BD 43B6        JSR     ENDCHK      00000316
433B 7E 4300        JMP     BEGIN       00000318
433E 81 4C   LTEST  CMP A   #$4C        'INPUT=L?
4340 26 06          BNE     ETEST       00000326
4342 BD 4548        JSR     WLIST       00000330
4345 7E 4300        JMP     BEGIN       00000334
4348 81 45   ETEST  CMP A   #$45        INPUT=E?
434A 26 06          BNE     EMSG        00000342
434C BD 43C0        JSR     ENGLSH      00000346
434F 7E 4300        JMP     BEGIN       00000350
4352 CE 411B EMSG   LDX     #MSG1       00000354
4355 BD 435E        JSR     VMSG        SPOKEN MSG
4358 BD 43B6        JSR     ENDCHK      00000362
435B 7E 4300        JMP     BEGIN       00000366


                    *ROUTINE TO OUTPUT SPOKEN MSG OR TEXT
435E 86 00   VMSG   LDA A   #0          00000378
4360 B7 8014        STA A   $8014       RESET FIFOS
4363 86 40          LDA A   #$40        00000386
4365 B7 8014        STA A   $8014       PREPARE FOR DATA ENTRY
4368 B6 8016 PHON   LDA A   $8016       00000394
436B 84 80          AND A   #$80        MASK FOR BIT 7
436D 81 00          CMP A   #0          CHECK FOR FIFO FULL
```

```
436F 27 F7            BEQ    PHON      LOOP IF FULL
4371 B6 8016          LDA A  $8016     00000410
4374 84 20            AND A  #$20      MASK FOR BIT 5
4376 81 00            CMP A  #0        CHECK FOR FIFO EMPTY
4378 26 0A            BNE    PHON2     00000422
437A 86 C3            LDA A  #$C3      DUMMY PAUSE
437C B7 8014          STA A  $8014     00000430
437F 86 43            LDA A  #$43      RESET BIT 7
4381 B7 8014          STA A  $8014     00000438
4384 A6 00     PHON2  LDA A  0,X       GET CHAR
4386 8A C0            ORA A  #$C0      MASK OFF RESET AND STROBE
4388 B7 8014          STA A  $8014     SEND CODE
438B 84 7F            AND A  #$7F      TURN OFF BIT 7 STROBE
438D B7 8014          STA A  $8014     00000458
4390 84 3F            AND A  #$3F      MASK BITS 6&7
4392 81 3F            CMP A  #$3F      END OF TABLE?
4394 26 01            BNE    CONTIN    00000470
4396 39               RTS              00000474
4397 08        CONTIN INX              INCR TABLE ADDR
4398 7E 4368          JMP    PHON      CONTINUE
439B 39               RTS              00000486

               *ROUTINE TO ENCODE & STORE V CODES IN TABLE
439C A6 00     ENCODE LDA A  0,X       A GETS VCODE
439E 08               INX              00000502
439F 81 BF            CMP A  #$BF      END OF CODES?
43A1 27 12            BEQ    CODEND    00000510
43A3 FF 414A          STX    CX        00000514
43A6 FE 414E          LDX    VX        00000518
43A9 A7 00            STA A  0,X       STORE IN V TABLE
43AB 08               INX              00000526
43AC FF 414E          STX    VX        00000530
43AF FE 414A          LDX    CX        00000534
43B2 7E 439C          JMP    ENCODE    00000538
43B5 39        CODEND RTS              00000542

               *SUBROUTINE ENDCHK
               *CHECKS FOR END OF SPOKEN MSG

43B6 B6 8016   ENDCHK LDA A  $8016     00000562
43B9 84 20            AND A  #$20      MASK FOR BIT 5
43BB 81 00            CMP A  #0        CHECK FOR FIFO EMPTY
43BD 26 F7            BNE    ENDCHK    00000574
43BF 39               RTS              00000578

               *SUBROUTINE ENGLISH
               *CONVERTS ENGLISH INPUT VIA KEYBOARD
               *TO SPEECH
               *USE CR AND LF FOR A NEW LINE
```

```
               *USE RUBOUT TO CANCEL ERROR
               *USE SPACE OR PUNCTUATION BETWEEN WORDS
               *ALLOWED PUNCTUATION MARKS ARE  ? , AND .
               *PUNCTUATION MUST PRECEDE ESCAPE
               *MUST HAVE SPACE BETWEEN NUMBERS. EX: 2 A 1 0
               *
               *NOT ALLOWED:
               *1) MORE THAN ONE SPACE BETWEEN WORDS
               *2) MORE THAN ONE PUNCTUATION MARK BETWEEN WORDS
               *3) ADJACENT PUNCTUATION MARK AND SPACE

43C0 CE 41A0   ENGLSH LDX    #MSG2     00000646
43C3 BD E07E          JSR    $E07E     PROMPTING MSG
43C6 CE 3B00          LDX    #ATABLE    00000654
43C9 BD E1AC   GETIN  JSR    $E1AC     KEYBOARD INPUT
43CC 81 0D            CMP A  #$D       CR?
43CE 27 F9            BEQ    GETIN     YES
43D0 81 0A            CMP A  #$A       LF?
43D2 27 F5            BEQ    GETIN     YES
43D4 81 7F            CMP A  #$7F      RUBOUT?
43D6 26 13            BNE    SKIP2     00000682
43D8 8C 3B00          CPX    #ATABLE    FIRST ENTRY?
43DB 27 EC            BEQ    GETIN     00000690
43DD 09               DEX              DECR X TO WRITE OVER ERROR
43DE 86 08            LDA A  #8        00000698
43E0 BD E1D1          JSR    $E1D1     00000702
43E3 86 23            LDA A  #$23      00000706
43E5 BD E1D1          JSR    $E1D1     00000710
43E8 7E 43C9          JMP    GETIN     00000714
43EB 81 1B     SKIP2  CMP A  #$1B      ESCAPE?
43ED 27 06            BEQ    INEND2    00000722
43EF A7 00            STA A  0,X       STORE ASCII CHAR IN TABLE
43F1 08               INX              INCR. ADDR
43F2 7E 43C9          JMP    GETIN     00000734
43F5 86 04     INEND2 LDA A  #4        END OF TABLE MARK
43F7 A7 00            STA A  0,X       00000742
43F9 CE 3D00          LDX    #VTABLE    X GETS BEGIN OF VOTRAX TABLE
43FC FF 414E          STX    VX        00000750
43FF CE 412B          LDX    #TEMP     X GETS TEMPORARY STORE ADDR
4402 FF 4148          STX    TX        00000758
4405 CE 3B00          LDX    #ATABLE    X GETS ASCII TABLE ADDR
4408 A6 00     EOUT   LDA A  0,X       00000766
440A 08               INX              00000770
440B 81 04            CMP A  #4        END?
440D 26 11            BNE    SPCHK2    00000778
440F FE 414E          LDX    VX        00000782
4412 86 BF            LDA A  #$BF      END OF V TABLE MARK
4414 A7 00            STA A  0,X       00000790
4416 CE 3D00          LDX    #VTABLE    START OF TEXT
4419 BD 435E          JSR    VMSG      OUTPUT SPOKEN TEXT
441C BD 43B6          JSR    ENDCHK    00000802
441F 39               RTS              00000806
```

```
4420 FF 4146 SPCHK2 STX     AX        00000810
4423 81 20          CMP A   #$20      SPACE?
4425 27 1B          BEQ     STNULL    00000818
4427 81 2C          CMP A   #$2C      , ?
4429 27 17          BEQ     STNULL    00000826
442B 81 2E          CMP A   #$2E      . ?
442D 27 13          BEQ     STNULL    00000834
442F 81 3F          CMP A   #$3F      ? ?
4431 27 0F          BEQ     STNULL    00000842
4433 FE 4148        LDX     TX        00000846
4436 A7 00          STA A   0,X       STORE LETTER
4438 08             INX               00000854
4439 FF 4148        STX     TX        00000858
443C FE 4146        LDX     AX        00000862
443F 7E 4408        JMP     EOUT      00000866
4442 86 04   STNULL LDA A   #4        END OF TABLE MARK
4444 FE 4148        LDX     TX        00000874
4447 A7 00          STA A   0,X       STORE AT END OF TABLE
4449 CE 412B        LDX     #TEMP     00000882
444C A6 00          LDA A   0,X       00000886
444E 08             INX               00000890
444F FF 4148        STX     TX        00000894
4452 81 2F          CMP A   #$2F      ZERO CODE
4454 22 03          BHI     NUMCH     A > OR = 0?
4456 7E 4480        JMP     NOTNUM    NOT A NUMBER
4459 81 39   NUMCH  CMP A   #$39      A < OR = 9?
445B 22 23          BHI     NOTNUM    00000914
445D CE 4150        LDX     #ZERO     START OF V NUMBER CODES
4460 C6 30          LDA B   #$30      00000922
4462 11      CPNO   CBA               A=B?
4463 26 0F          BNE     CHK10     00000930
4465 BD 439C        JSR     ENCODE    STORE V CODES IN TABLE
4468 CE 412B        LDX     #TEMP     00000938
446B FF 4148        STX     TX        00000942
446E FE 4146        LDX     AX        00000946
4471 7E 4408        JMP     EOUT      00000950
4474 5C      CHK10  INC B             INCR B
4475 08             INX               INCR ADDR
4476 08             INX               00000962
4477 08             INX               00000966
4478 08             INX               00000970
4479 08             INX               00000974
447A 08             INX               00000978
447B 08             INX               00000982
447C 08             INX               00000986
447D 7E 4462        JMP     CPNO      00000990
4480 81 40   NOTNUM CMP A   #$40      LETTER A OR GREATER?
4482 22 03          BHI     CHKLT     00000998
4484 7E 448B        JMP     NOTWD     NOT A WORD
4487 81 5A   CHKLT  CMP A   #$5A      LETTER Z OR LESS?
4489 23 1F          BLS     AWRD      BRANCH IF WORD
448B 86 0D   NOTWD  LDA A   #$D       00001014
448D BD E1D1        JSR     $E1D1     OUTPUT CR
4490 86 0A          LDA A   #$A       00001022
4492 BD E1D1        JSR     $E1D1     OUTPUT LF
4495 86 20          LDA A   #$20      00001030
4497 BD E1D1        JSR     $E1D1     OUTPUT 2 SPACES
449A BD E1D1        JSR     $E1D1     00001038
449D CE 412B        LDX     #TEMP     00001042
44A0 BD E07E        JSR     $E07E     OUTPUT INVALID WORD
44A3 CE 41C1        LDX     #MSG3     00001050
44A6 BD E07E        JSR     $E07E     OUTPUT ERROR MSG
44A9 39             RTS               RETURN
44AA B7 4141 AWRD   STA A   SAVA      00001062
44AD 84 3F          AND A   #$3F      MASK BITS 6&7
44AF B7 413F        STA A   XADR      00001070
44B2 B6 4141        LDA A   SAVA      00001074
44B5 FE 413F        LDX     XADR      00001078
44B8 FF 414C        STX     BX        00001082
44BB 81 04   CPNULL CMP A   #4        END OF WORD?
44BD 27 18          BEQ     WDEND     YES
44BF FE 414C        LDX     BX        00001094
44C2 E6 00          LDA B   0,X       00001098
44C4 08             INX               00001102
44C5 FF 414C        STX     BX        00001106
44C8 11             CBA               B=A?
44C9 26 51          BNE     RSTR      00001114
44CB FE 4148        LDX     TX        00001118
44CE A6 00          LDA A   0,X       00001122
44D0 08             INX               00001126
44D1 FF 4148        STX     TX        00001130
44D4 7E 44BB        JMP     CPNULL    00001134
44D7 FE 414C WDEND  LDX     BX        00001138
44DA E6 00          LDA B   0,X       00001142
44DC 08             INX               00001146
44DD FF 414C        STX     BX        00001150
44E0 C1 7F          CMP B   #$7F      V CODE?
44E2 23 38          BLS     RSTR      BRANCH IF ASCII
44E4 C1 BF          CMP B   #$BF      END OF TABLE?
44E6 27 A3          BEQ     NOTWD     00001166
44E8 FE 414E STCODE LDX     VX        00001170
44EB E7 00          STA B   0,X       STORE CODE IN VTABLE
44ED 08             INX               00001186
44EE FF 414E        STX     VX        00001190
44F1 FE 414C        LDX     BX        00001194
44F4 E6 00          LDA B   0,X       GET CHAR FROM WORD TABLE
44F6 08             INX               00001202
44F7 FF 414C        STX     BX        00001206
44FA C1 BF          CMP B   #$BF      00001207
44FC 27 04          BEQ     SKIP1     00001208
44FE C1 7F          CMP B   #$7F      V CODE?
4500 22 E6          BHI     STCODE    BRANCH IF V CODE
4502 C6 83   SKIP1  LDA B   #$83      PAUSE CODE
4504 FE 414E        LDX     VX        00001222
```

```
4507 E7 00          STA B   0,X        STORE PAUSE IN V TABLE
4509 08             INX                00001230
450A FF 414E        STX     VX         00001234
450D CE 412B        LDX     #TEMP      00001238
4510 FF 4148        STX     TX         00001242
4513 FE 4146        LDX     AX         00001246
4516 7E 4408        JMP     EOUT       00001250
4519 7E 440B PASJMP JMP     NOTWD      00001254
451C CE 412B RSTR   LDX     #TEMP      00001258
451F FF 4148        STX     TX         00001262
4522 FE 414C        LDX     BX         00001266
4525 E6 00   GETB1  LDA B   0,X        00001270
4527 08             INX                00001274
4528 C1 7F          CMP B   #$7F       VCODE?
452A 23 F9          BLS     GETB1      BRANCH IF ASCII
452C E6 00   GETB2  LDA B   0,X        00001286
452E 08             INX                00001290
452F C1 BF          CMP B   #$BF       00001294
4531 27 E6          BEQ     PASJMP     00001298
4533 C1 7F          CMP B   #$7F       VCODE?
4535 22 F5          BHI     GETB2      BRANCH IF VCODE
4537 09             DEX                DECR INDEX
4538 FF 414C        STX     BX         00001314
453B FE 4148        LDX     TX         00001318
453E A6 00          LDA A   0,X        00001322
4540 08             INX                00001326
4541 FF 4148        STX     TX         00001330
4544 7E 44BB        JMP     CPNULL     00001334
4547 39             RTS                RETURN

             *WLIST: SUBROUTINE TO OUTPUT A LIST OF WORDS
             *IN VOCABULARY.
             *WORD ADDR, WORD, AND HEX CODES ARE PRINTED.

4548 86 0D   WLIST  LDA A   #$D        CR CODE
454A BD E1D1        JSR     $E1D1      OUTPUT CR
454D 86 0A          LDA A   #$A        LF CODE
454F BD E1D1        JSR     $E1D1      OUTPUT LF
4552 4F             CLR A              A=0
4553 4C      ADD1   INC A              A=A+1
```

```
4554 B7 4141        STA A   SAVA       00001386
4557 FE 4141        LDX     SAVA       00001390
455A FF 4143        STX     SAVX       00001394
455D CE 4143 LOOPX  LDX     #SAVX      00001398
4560 BD E0C8        JSR     $E0C8      OUT4HS (OUTPUT ADDR)
4563 BD E0CC        JSR     $E0CC      OUTPUT SPACE
4566 FE 4143        LDX     SAVX       00001410
4569 A6 00   NCHAR  LDA A   0,X        00001414
456B 81 7F          CMP A   #$7F       ASCII?
456D 22 07          BHI     CODES      NO
456F BD E1D1        JSR     $E1D1      OUTPUT CHAR
4572 08             INX                INCR INDEX
4573 7E 4569        JMP     NCHAR      00001434
4576 B7 4145 CODES  STA A   TCODE      00001438
4579 08             INX                00001442
457A FF 4143        STX     SAVX       00001446
457D BD E0CC        JSR     $E0CC      OUTPUT SPACE
4580 CE 4145        LDX     #TCODE     00001454
4583 BD E0CA        JSR     $E0CA      OUT2HS (OUT HEX VCODE)
4586 FE 4143        LDX     SAVX       00001462
4589 A6 00          LDA A   0,X        GET CHAR
458B 81 BF          CMP A   #$BF       END OF TABLE?
458D 27 11          BEQ     CKSAVA     00001474
458F 81 7F          CMP A   #$7F       ASCII?
4591 22 E3          BHI     CODES      00001482
4593 86 0D          LDA A   #$D        CR CODE
4595 BD E1D1        JSR     $E1D1      OUTPUT CR
4598 86 0A          LDA A   #$A        LF CODE
459A BD E1D1        JSR     $E1D1      OUTPUT LF
459D 7E 455D        JMP     LOOPX      00001502
45A0 86 0D   CKSAVA LDA A   #$D        00001506
45A2 BD E1D1        JSR     $E1D1      CR
45A5 86 0A          LDA A   #$A        00001514
45A7 BD E1D1        JSR     $E1D1      LF
45AA B6 4141        LDA A   SAVA       00001522
45AD 81 1A          CMP A   #$1A       00001526
45AF 26 A2          BNE     ADD1       NOT END OF LIST
45B1 CE 41DA        LDX     #MSG4      00001534
45B4 BD E07E        JSR     $E07E      PDATA1 (OUTPUT MSG)
45B7 39             RTS                00001542
                    END                00001546
```

REFERENCES CITED

Allen, Jonathan. "Synthesis of Speech from Unrestricted Text." Proceedings of the IEEE 64 (April 1976): 433-442.

Atmar, Wirt. "The Time Has Come to Talk." Byte, August 1976, pp. 26-33.

Digital Group/Votrax Voice Synthesizer. Denver, Colorado: The Digital Group, [1978].

Elowitz, H.S. "Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics." IEEE Transactions on Acoustics, Speech, and Signal Processing 24 (December 1976): 446-459.

Flanagan, James L. "The Synthesis of Speech." Scientific American 226 (February 1972): 48-57.

Flanagan, James L. "Voices of Men and Machines." Journal of the Acoustical Society of America 51 (May 1972): 1375-1387.

Flanagan, James L. "Computers that Talk and Listen: Man-Machine Communication by Voice." Proceedings of the IEEE 64 (April 1976): 405-415.

Friedman, Joyce. "Computer Exploration of Fast-Speech Rules." IEEE Transactions on Acoustics, Speech, and Signal Processing 23 (February 1975): 100-103.

Holmes, J.N. "The Influence of Glottal Waveform on the Naturalness of Speech from a Parallel-Formant Synthesizer." In Record of IEEE 1972 Conference on Speech Communication and Processing Newton, Mass., April 24-26, 1972, pp. 148-151. Ed. by Caldwell P. Smith. Bedford, Mass.: Data Sciences Laboratories, Air Force Cambridge Research Laboratories, 1972.

House, Arthur. "On Vowel Duration in English." Journal of the Acoustical Society of America 33 (September 1961): 1174-1178.

Klatt, Dennis. "Acoustic Theory of Terminal Analog Speech Synthesis." In Record of the IEEE 1972 Conference on Speech Communication and Processing, Newton, Mass., April 24-26, 1972, pp. 131-135. Ed. by Caldwell P. Smith. Bedford, Mass.: Data Sciences Laboratories, Air Force Cambridge Research Laboratories, 1972.

Klatt, Dennis. "Structure of a Phonological Rule Component for a Synthesis-by-Rule Program." IEEE Transactions on Acoustics, Speech, and Signal Processing 24 (October 1976): 391-398.

Leutenegger, Ralph R. The Sounds of American English. Chicago: Scott, Foresman and Company, 1963.

Markel, J.D., and Gray, A.H., Jr., Linear Prediction of Speech. New York: Springer-Verlag, 1976.

Morgan, D.R., and Craig, S.C., Real-Time Linear Prediction Using the LMS Gradient Algorithm. Syracuse, N.Y.: General Electric, [1976].

O'Shaugnessy, Douglas. "Consonant Durations in Clusters." IEEE Transactions on Acoustics, Speech, and Signal Processing 22 (August 1972): 282-295.

Rabiner, Lawrence R., and Schafer, Ronald W. "Digital Techniques for Computer Voice Response: Implementations and Applications." Proceedings of the IEEE 64 (April 1976): 416-433.

Rao, P.V.S., and Thosar, R.B. "A Programming System for Studies in Speech Synthesis." IEEE Transactions on Acoustics, Speech, and Signal Processing 22 (June 1977): 217-225.

Rice, Lloyd D. "Friends, Humans, and Countryrobots: Lend Me Your Ears." Byte, August 1976, pp. 16-24.

Wiggins, Richard, and Brantingham, Larry. "Three-Chip System Synthesizes Human Speech." Electronics 51 (August 31, 1978): 109-116.

Wise, Claude Merton, Introduction to Phonetics. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1958.