SOUND SYNTHESIS IN A TEACHING MACHINE
FOR TACTILE RECOGNITION

by

Karl D. Presser

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 6 8

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of them major department of the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Karl D. Presser

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

F. J. Hill
F. J. HILL
Professor of Electrical Engineering

20 July 1968
Date

ACKNOWLEDGMENTS

The author is most grateful for the encouragement and assistance given by Dr. Fredrick J. Hill in the preparation of this thesis.

Acknowledgment is also made of the use of spectral data obtained during a study of synthetic speech, which was conducted at the Applied Research Laboratory of The University of Arizona, and directed by Mr. Roland D. August.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

A small general-purpose digital computer was programmed to carry out the functions of a teaching machine for use in tactual and audio perception experiments. The machine generates synthetic vowels which are used as the standard training stimuli. A sequence of stimuli can be presented to a subject under the automatic control of various training and testing programs. A subject's responses are communicated to the machine via a Teletype console. The machine requires a minimum of operator intervention and records all test data in a permanent fashion. An operator is allowed to experiment with the generation of new stimuli through a set of flexible commands.

CHAPTER I

INTRODUCTION

There is at present extensive interest in the problem of speech perception for the deaf and hard-of-hearing. In 1966 a program to develop hardware which will enable the deaf to perceive speech through the tactile nerves was initiated by Dr. Fredrick J. Hill at The University of Arizona's Department of Electrical Engineering, and a prototype speech processor for use in this tactile sensing system was constructed (McClellan 1967). This processor accepts conventional speech as input, averages the amplitude distributions of eight frequency bands which span the audio range, and uses the average of each band or channel to modulate a sinusoid whose frequency is the center frequency of that particular band. The output of the processor was found to be quite intelligible to the human ear. The recognition rate for phonemes was approximately seventy percent.

Ultimately the eight-channel output of this processor will drive a bank of eight tactile transducers. The transducers have not been developed, and at this point in the project it is felt that a major effort should be extended toward determination of the stimulation properties of the skin and acquisition of data to aid in the design of adequate stimulators.

These investigations will require that a number of subjects be presented with tactual stimuli in a series of training sessions and tests, in order to measure their particular tactual learning curves. In addition, audio learning curves must be obtained by training another group of subjects to recognize audio stimuli. The intelligibilities of the two types of stimuli should be the same to facilitate direct comparison of the tactual and audio learning rates. If these experiments indicate that the tactile rate is close to the audio rate then, indeed, tactual speech recognition is possible.

Following this, another series of experiments emphasizing tactile stimuli will be required. The results of these tests will be used to determine the effectiveness of one transducer over that of another design. The test data can also be used to aid in modification of the speech processor itself since it is desirable to maximize the average information in each channel. Finally, once the tactual recognition system is completely designed, it is apparent that a hearing-impaired subject can effectively use this apparatus only after training.

The above considerations have resulted in the development of a teaching machine which can automatically administer training sessions, using either tactile or audio stimuli, to a subject, and then test the subject to measure his learning curve. The machine requires a minimum of operator intervention, provides tests which are repeatable at any time, and records all test data in a permanent fashion. It is the purpose of this report to describe the development and operation of this machine.

The teaching machine was implemented on a PDP-9 (Programmed Data Processor) digital computer. This machine is located in the Hybrid Computation Laboratory, and is operated under the direction of Dr. G. A. Korn and Dr. J. V. Wait. The computer was programmed to generate synthetic vowels which are used as the standard training and test signals. These vowel sounds are suitable for use as input to the tactual speech processor and for use as conventional audio stimuli.

The synthetic vowels are presented to a subject under program control of various training and testing routines through a nine-bit digital-to-analog converter. A Teletype console is used to communicate the subject's responses to the computer. After a training session or test, the console is also used to provide a complete list of stimuli presented, the subject's responses to the stimuli, and his total score.

Since the synthesis of vowels was of interest in its own right, much flexibility has been included in the programs so that an operator may easily change the characteristics of the generated sounds. Thus, in addition to testing, the machine provides a basic speech research facility.

Chapter II explains the generation of the synthetic vowels in detail while Chapter III defines the manner of training and testing a subject with these sounds. Chapter IV explains the system as it would appear to a typical operator, and Chapter V indicates some possible methods which could be employed to synthesize a wider variety of stimuli or sounds. The programs which were developed to implement the system are included in Appendices A through G.

CHAPTER II

SYNTHETIC VOWEL GENERATION

The prime consideration in the development of the teaching machine was the selection of the standard training stimuli. Initially these signals will be used in training sessions in which a subject will hopefully learn to associate a unique code tag with a certain stimulus. The amount of learning will then be measured by presenting stimuli at random to the subject and asking him to supply the associated code tag.

It is important to note that the recognition rates measured for hearing-impaired subjects have to be compared with the recognition rates of ordinary subjects receiving audio signals via the ear. If conventional recorded speech signals were to be used in the teaching machine, the recognition rates would be difficult to compare directly since the average audio rate would be several orders of magnitude greater than the average tactual recognition rate. Normalizing these two rates would be a difficult task since so little is known about perception processes. More insight into the problems of tactile speech perception would be obtained if ordinary subjects, including those engaged in building and modifying speech perception hardware, could be tested with audio signals which compared more directly with the tactile test signals. In other words, it was felt that it would be desirable to have test signals which could be used not only as input to a tactile processor, but also as input to a loudspeaker. The intelligibility of such signals should be approximately the same in either configuration.

In addition to this intelligibility requirement the test signals should be isolated to eliminate context completely, and the stimuli must be extremely repeatable since individual tests may be separated by long periods of time. Evaluation of the above considerations led to the selection of synthetic vowels as the standard test signals.

2.1 Pure Vowel Fourier Synthesis

The synthetic vowels generated by the teaching machine are close approximations of what are called pure vowels. A pure vowel such as the ē in "team" is characterized by a nearly periodic signal since the articulatory organs are held in fixed positions when the vowel is sounded.

A previous study conducted by the Engineering Research Laboratory (August 1965) had measured the frequency spectra of different vowel sounds. Using this data, the fundamental frequency of each vowel was determined. For each sound, the amplitude of the fundamental frequency and the amplitudes of the first thirty-one harmonics of this frequency were also measured.

A subroutine called "FORYEA", which is listed in Appendix C, uses these parameters to generate one cycle, or period, of the Fourier series associated with each pure vowel. Relative phase information, which has a great effect on speech quality but little effect on intelligibility (Manley 1963), is not used in the program. The length of time required to generate a typical cycle, or waveform, is approximately ten seconds. However, this operation is required only once since the resulting waveform is stored. A stored waveform may then be repetitively sent out through the digital-to-analog converter at a ten kilohertz rate to form a stimulus which lasts from one-hundredth of a second to thirteen seconds as desired.

The synthetic vowels generated as described above were found to resemble natural vowel sounds but were not intelligible enough to use as testing signals. Accordingly, a provision to enhance these sounds through the use of amplitude modulation was included in the system.

2.2 Amplitude Modulation

When a vowel is spoken in isolation the energy of the sound builds up and decays gradually. The teaching machine's modulator program, known as "TALKM" and listed in Appendix G, was designed primarily to include this important characteristic in the generation of the synthetic vowels.

Up to ten curves or envelopes can be resident in the computer at one time. Typically five of these are envelopes suitable for modulating the beginning portion of the synthetic sound while the others are used for the ending portion. Usually the middle portion of the vowel will be unmodulated, although this is not a necessary restriction.

An envelope is specified by designating points on its curve. The number of points can vary from six to fifty-one as explained in Chapter IV Section 4.3. The modulator program assumes the points represent amplitude values which are equally spaced along the horizontal or time axis and performs linear interpolations to fill in the envelope. Thus, it is not necessary to provide the time coordinates of the curve on a point-by-point basis. Only the total length of time during which the envelope is to be used to modulate the waveform need be specified. This provision allows rapid experimentation with modulation effects.

Through the use of very precise timing chains, the digital modulator varies the amplitude of the basic vowel sound according to the specified envelopes at the same time that it is outputting the waveform to the subject at a ten kilohertz rate. The obvious advantage here is that no additional storage is required for the modulated stimuli.

The intelligibility of the synthetic vowels with modulation showed a marked improvement over that of the unmodulated vowels. Perhaps the most interesting result of this project was that, if a synthetic vowel was modulated with a simple linear curve during the beginning and ending quarters of its total duration, intelligibility was vastly increased; but when more complex curves, or even envelopes which extended over the total duration of the vowel were used, the intelligibility increased very little over that of the simply modulated sound.

2.3 Filtering the Stimuli

Although the computer was capable of achieving a maximum output rate of one-hundred kilohertz, it was necessary to use only a ten kilohertz rate. This was done to allow the digital modulator time to perform the interpolations and multiplications required to generate each point of the modulated waveform. This was not a serious restriction since a study (Fletcher 1953) revealed that the frequencies from two hundred and fifty hertz to six kilohertz contribute over ninety-five percent to intelligibility. However, at this output rate, frequencies of five kilohertz and over were reproduced poorly and resembled square waves in appearance and sound.

To prevent these higher frequencies from adding noise and buzz to the synthetic vowel sounds, a low-pass active filter, with a cutoff frequency of five kilohertz, was added to the digital-to-analog converter's output stage. The second-order filter is shown in Figure 2.1 and is a maximal flatness or Butterworth type with a 40db/decade rolloff. While the filter suppressed any high frequency noise effects it was not detrimental to the quality of the synthetic vowels.

In conclusion, it was found that while the synthetic vowels generated by Fourier synthesis and enhanced by amplitude modulation were not representative of speech in general, they were reasonably intelligible, required little storage, and were quite adequate for use as the standard test stimuli in the teaching machine.

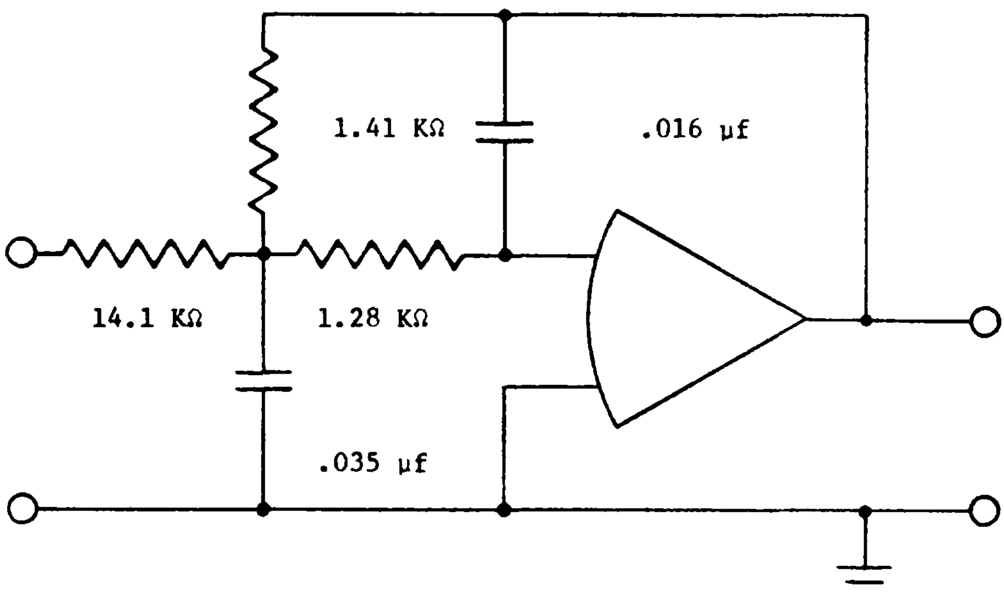

Figure 2.1 Low pass filter with component values

CHAPTER III

TRAINING AND TESTING

The teaching machine has two different modes of training a subject to recognize the stimuli; a prompting mode and a confirmation mode. The design of these training programs resulted from considerations of the relative efficiency of prompting versus confirmation in paired-associate learning (Sidley 1965) and the importance of the stimulus-label overlap.

3.1 The Test Table

The teaching machine performs all training and testing by following a test table which is specified by the operator at the beginning of a session. The test table consists of a sequence of special entries as explained in Chapter IV Section 4.3. Each entry designates a particular stimulus and the unique code tag which should be associated with it. The operator is free to arrange these stimuli in any order and use duplicate entries, as desired, to make up a test.

The test table must also contain two more parameters. The first is simply the total number of stimuli included in the table, and the second is the number of trials desired. Each of the training programs will present the stimuli one-by-one in the order of their appearance in the table. If the indicated number of trials exceeds the length of the table, the sequence is repeated by the program until the desired number of trials is obtained. The maximum number of trials is one hundred.

Once the test table is completed, the operator can use the operator mode as explained in Chapter IV Section 4.4 to select any entry in the table and output that particular stimulus upon command. This provision facilitates the examination of various test sequences as well as experimentation with new stimuli.

3.2 The Prompting Mode

In the prompting mode of training the subject is informed about each stimulus before it is presented. For each trial, the teaching machine prints the identification code associated with the stimulus on the Teletype console, and then outputs the signal to the subject. The program will not continue until the subject answers by retyping the given code on the console. Before answering, the subject can repeat the stimulus by typing the character "R".

After each answer the system prints the code once more and either the word "correct", if the subject has retyped the tag correctly, or the word "incorrect", if the subject was not paying attention and typed the wrong code.

An example of this training mode appears in Figure 3.1. Only five trials were used for this illustration, and the identification codes are numbered sequentially as would not be the case in a real training situation. The figure indicates the manner in which the stimulus is identified to the subject before presentation, and also illustrates the complete listing of the session provided by the program.

3.3 The Confirmation Mode

In the confirmation mode the subject is not given the identifying

```
LISTEN TO STIMULUS S001
YOUR ANSWER IS?  S001
S001     CORRECT


LISTEN TO STIMULUS S002
YOUR ANSWER IS?  S002
S002     CORRECT


LISTEN TO STIMULUS S003
YOUR ANSWER IS?  S003
S003     CORRECT


LISTEN TO STIMULUS S004
YOUR ANSWER IS?  S004
S004     CORRECT


LISTEN TO STIMULUS S005
YOUR ANSWER IS?  S005
S005     CORRECT
```

TRIAL	STIMULUS	RESPONSE	CORRECT/INCORRECT
001	001	001	CORRECT
002	002	002	CORRECT
003	003	003	CORRECT
004	004	004	CORRECT
005	005	005	CORRECT

Figure 3.1 An example of the prompting mode

code, but only told to listen before the presentation of each stimulus. After the stimulus, the subject may answer by using the console, or repeat the signal by typing the character "R". When the subject answers he is immediately given the correct identification code. If the subject's answer was correct, the teaching machine prints the word "correct" and proceeds to the next stimulus. If the subject was wrong, however, the word "incorrect" is printed, and the code belonging to the present stimulus is placed in a specific area in the computer's memory.

As long as there are any entries in this memory area, the teaching machine will interrupt the normal training sequence at every fourth trial and present one of the previously-missed stimuli to the subject. If the subject answers correctly, when presented with one of these repeated stimuli, the signal's code is removed from the memory area. Since these repeated stimuli are in a sense additions to the specified training sequence, they are listed with a "plus sign" when the final record of the session is printed.

A sample training session of this type appears in Figure 3.2 to illustrate the above points. In each trial, stimulus presentation occurs after the message "Listen to Stimulus". The subject's answer is confirmed as shown. The sequential numbering of the signal codes is used to illustrate the addition of missed stimuli to the training sequence. If during the second trial, the subject answers incorrectly, as is shown, stimulus S002 is presented again at trial four instead of stimulus S004. The normal sequence of signals is then resumed starting with trial five.

LISTEN TO STIMULUS
YOUR ANSWER IS? S001
S001 CORRECT

LISTEN TO STIMULUS
YOUR ANSWER IS? S001
S002 INCORRECT

LISTEN TO STIMULUS
YOUR ANSWER IS? S003
S003 CORRECT

LISTEN TO STIMULUS
YOUR ANSWER IS? S002
S002 CORRECT

LISTEN TO STIMULUS
YOUR ANSWER IS? S004
S004 CORRECT

TRIAL	STIMULUS	RESPONSE	CORRECT/INCORRECT	EXTRA
001	001	001	CORRECT	
002	002	002	INCORRECT	
003	003	003	CORRECT	
004	002	002	CORRECT	+
005	004	004	CORRECT	

SCORE 004/005

Figure 3.2 An example of the confirmation mode

The additional presentation of stimulus S002 is identified by a "plus sign" in the "Extra" column of the final listing. This feature allows the operator to quickly determine those stimuli which were difficult for the subject to identify. In this mode the subject's score is printed as indicated in the figure.

3.4 The Testing Mode

The testing mode is intended to provide researchers with an accurate measure of a subject's ability to recognize the different stimuli. During a test the subject is not informed about the stimulus, either before or after its presentation, and is not told if his answer is correct. The teaching machine presents the stimuli exactly as they appear in the test table without additions, and the subject is not allowed to repeat a stimulus before answering.

If the subject makes an error in typing his intended answer he may signal this to the system by typing the Teletype character "RUBOUT", and then retype his answer correctly. This provision is also included in the two training programs, but its necessity is more apparent in this mode.

The testing record provided by the system is similar to that of the two training modes, and of course the final score is always printed.

CHAPTER IV

SYSTEM OPERATION

The seven programs listed in Appendices A through G were developed with the intention of allowing the PDP-9 (Programmed Data Processor) digital computer to function as a teaching machine for tactile and audio perception experiments. Much effort was extended to ensure that the computer remained in the background, so to speak, and that the system placed no excessive demands on either the operator or the test subject. A person wishing to generate stimuli or set up a test needs only a minimal understanding of digital computers to perform these tasks. A test subject is only required to be able to type simple answers at his own rate of speed. This chapter is intended to be a guide to the operation of the system, and the material explained below applies to potential operators rather than to test subjects. Although this chapter must be, in essence, an operation manual, it is hoped that the material contained within will not appear to be excessively discontinuous.

A simple block diagram of the system appears in Figure 4.1. To set up a training session the operator does not press buttons, as in other teaching machines (Risberg and Spens 1967), but issues commands via the Teletype console. These commands are fully explained in this chapter, and a few preliminary conventions must be established. Two non-printing Teletype characters used in various commands are the "space" and the "carriage-return". The "space" will be represented by

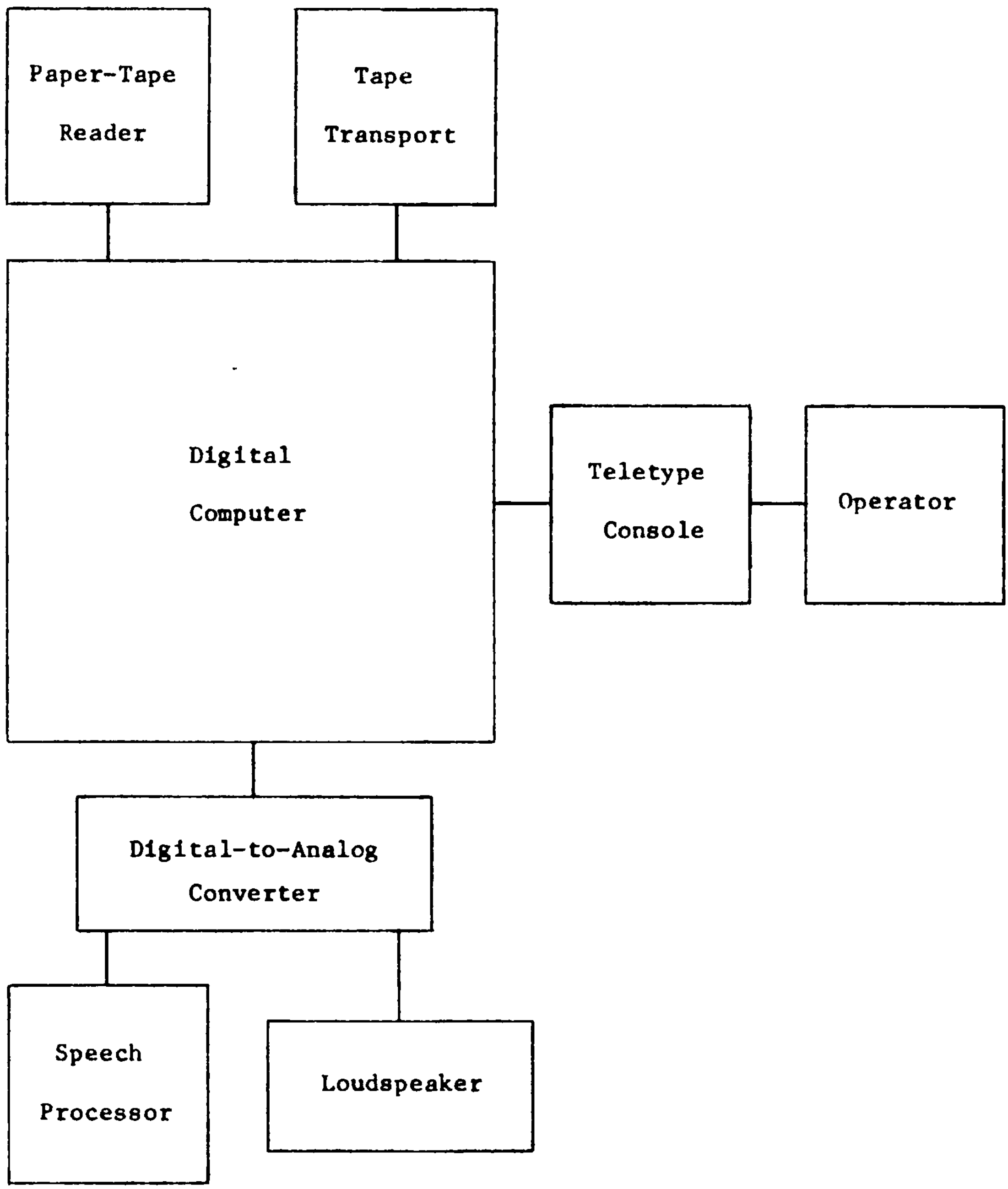

Figure 4.1 Block diagram of the teaching machine

underlining the positions where it occurs, and the "carriage-return" will be designated by the symbol "¢". One other character of special interest is the "CONTROL Z". To form a "CONTROL Z" the operator should hold down the "CONTROL" key on the Teletype console, and simultaneously type the character "Z".

4.1 Loading the System

Of course, the teaching system's programs must be loaded into the computer's memory before the computer can function as a training machine. This operation is performed by using the "LINKING LOADER", a program in the computer's system library. The teaching machine's programs are available in paper tape form or on a reel of magnetic tape, and a different sequence of operations is used for each format. If the paper tape version is used, the operator should type the command "GLOAD¢" to summon the "LINKING LOADER", place the tape in the paper-tape reader, and wait for the computer to print "LOADER". When the computer responds, the operator should type the letters "CNTROL", followed by six "commas" and then depress the "ALTMODE" key on the Teletype console. When the programs have been loaded, the teaching machine will signify that it is ready for use by printing the message "MODE".

To load the magnetic tape version, the operator should place the reel of tape on the tape transport, and rotate the transport's unit indicator until a "1" is in view. The operator should issue the command "ASSIGN_DTC1_-4¢", which indicates that magnetic tape is being used, and then type "GLOAD¢". When the computer prints "LOADER", the operator should type the letters "CNTROL" and depress the "ALTMODE" key. The "LINKING LOADER" will then load the programs. When this is accomplished the teaching machine will print the message "MODE" to indicate that it is ready for use.

4.2 Tabular Storage

In order to generate synthetic vowels or administer a training session, the teaching machine refers to various tables. These tables are actually certain groups of storage locations in the computer's memory. There are three different types of tables; the spectra tables, the modulation tables, and a test table.

The data in a spectra table is used to synthesize one cycle of a pure vowel. There are eighteen spectra tables, and the two-digit numbers from "01" to "18" are used to distinguish them. These numbers will be referred to as spectra table codes.

A modulation table contains a point-by-point representation of a curve, or envelope. The teaching machine will use a particular envelope to modulate a synthetic vowel. Typically, two envelopes are needed to form an intelligible synthetic vowel; one for the beginning portion of the vowel and one for the ending portion. In view of this requirement, the modulation tables are grouped into two classes. Five of the ten modulation tables are designated by the two-character codes "1N" through "5N", and in practice these tables will contain values which specify curves suitable for modulating the beginning of the vowel sound. The remaining five tables are identified by the two-character codes "1F" through "5F" and specify envelopes for modulating the ending portion of the vowel. Both types of the two-character codes will be referred to as modulation table codes.

The single test table specifies the sequence of stimuli to be used in a training session or test. The single character "T" is used to designate this table and is referred to as the test table code.

The teaching machine possesses a variety of commands designed to facilitate the entry of data into these tables. The same commands are used for all tables. The command "I_XX¢", where the symbol "XX" represents a particular table code as explained above, will cause the computer to locate the storage area associated with that table and print the table's title. After receiving this command, the computer assumes that all further commands refer to the previously designated table. This assumption is in effect until the operator types a "CONTROL Z".

Since each location in a table represents the value of a different parameter, a pointer method is used to enter data into a specific location. The pointer is initially positioned at the first location in the table. To move the pointer three commands are available. The command "N¢" will bring the pointer to the next location in the table. The command "A_XX¢" will move the pointer past a number of locations equal to the two-digit decimal number represented by the symbol "XX". If either of these two commands result in an attempt to move the pointer past the end of the table, an error message is printed, and the pointer is reset to the first location in the table. A third command, "T¢", allows the operator to move the pointer to the first location in the table at any time.

To enter data into any location of any table the operator should type "E_XX¢", where the symbol "XX" represents a particular value of a parameter. When this command is given the data is placed in the location at which the pointer is positioned. The data transfer can be verified by using the command "P¢". This command will cause the computer to print the value currently stored in the location, as well as a shortened form of the parameter name associated with the location.

To obtain a complete listing of a table, it is necessary to type "L¢". An illustration of the type of table listing provided by the computer appears in Figure 4.2. Each line of the listing shows the shortened form of the parameter name, as well as the current value of the parameter. Any location which contains only the number zero is not listed.

Of course, entering data into a table is a tedious job. In view of this, a command is available which records the contents of a table by punching the data values on a piece of paper tape. The data is coded in a fashion acceptable to the computer. Another command is provided to read this tape into the computer's memory at a later date. Thus it is only necessary to type in tabular data once. The command "P_XX¢", where the symbol "XX" represents the code of a particular table, causes the tape to be punched. The memory location of the table is also punched on this tape. Tables recorded in this fashion on paper tape can be read into the computer's memory in the following manner. The tape should be placed in the paper-tape reader, and the command

SPECTRA 06

FREQFO	000133
AMPFOO	000030
AMPF01	000295
AMPF02	000400
AMPF03	000810
AMPF04	001450
AMPF05	002620
AMPF06	002150
AMPF07	002950
AMPF08	000375
AMPF09	000350
AMPF10	000140
AMPF11	000075
AMPF12	000045
AMPF13	000045
AMPF14	000050
AMPF15	000250
AMPF16	000740
AMPF17	000150
AMPF18	000195
AMPF19	000095
AMPF20	000050
AMPF21	000040
AMPF22	000050

Figure 4.2 An example of a spectra table listing

"R_¢" should be issued. The computer uses the location previously punched on the tape to place the table in the proper memory area.

4.3 Tabular Data

Although the commands explained above are used for all tables, the data content of the three types of tables is not the same. The thirty-four parameters whose values are stored in a spectra table are assigned sequential locations in this manner. The value of the fundamental frequency is first, followed by the d-c level of the waveform. The third location contains the amplitude of the fundamental frequency, while the remaining locations contain the amplitudes of the first thirty-one harmonics of the fundamental frequency. All data in a spectra table must be in integer form. The lowest allowable value of the fundamental frequency is one-hundred hertz while all other parameters may vary from zero to one-hundred thousand.

The data in a modulation table is formatted in this manner. The first location indicates the total number of point values to be contained in the table. This number must be in integer form, and the only permissible values are 6, 11, 21, 25, and 51. The remaining entries are the values of the points used to define a particular curve, or envelope. These values must be either decimal fractions in the range from 0.0 to 0.9999, or 1.0. The values are assumed to represent sequential points of the curve which are equally spaced along the horizontal axis.

The test table specifies the particular stimuli to be used in a training session or test. The first entry gives the total number of stimuli listed in the table. This number must be in integer form and the maximum permissible value is fifty. The second entry specifies the desired number of trials. The maximum number of trials is one hundred. The remaining entries in the table specify the stimuli. Since the teaching machine uses the eighteen spectra tables and the ten modulation tables to generate a variety of stimuli, these entries are rather involved. Each entry of this type must specify seven items. The first item is the code of the particular spectra table which should be used to generate one cycle of the stimulus. The second item is the duration of the unmodulated portion of the stimulus and is given in seconds. An envelope suitable for modulating the beginning portion of the stimulus is specified by designating a particular modulation table code, and this code is the third item. The next item is the duration of time during which this envelope should be used. This is given in seconds. In a similar fashion the fifth and sixth items specify another envelope and its duration of use. This envelope should be suitable for modulating the ending portion of the vowel. The seventh item is simply the unique decimal number or code, which should be associated with the stimulus. These seven items are listed in Figure 4.3. As stated above, the second, fourth, and sixth items are durations of time, in seconds, and these can vary from zero to thirteen seconds in increments of one-hundredth of a second. The seven items are entered into a location in the test table by using a command which begins with the characters "E_". These characters should be followed by the seven items. The first six items should be terminated with a "comma" while the seventh item should be terminated with a "carriage return". Thus, except for the fact that seven items are being specified at once, this command is identical to the command for entering data as described in Section 4.2. An example of this type of command appears in Figure 4.4.

4.4 Other Commands

Before the teaching machine can synthesize a stimulus, one cycle of the basic waveform must be computed and stored. Of course, the data necessary to generate this waveform is contained in a spectra table as explained in Section 4.3. The command "C_XX¢", where the symbol "XX" represents a particular spectra table code, will cause the computer to generate the waveform specified by the table. Since the resulting waveform is stored, this operation is only required once for each spectra table.

When the operator has computed the basic waveforms and established a test table, he can initiate a training session or test by typing "T_X¢". The symbol "X" is used to specify which mode of operation is desired. The letters "P", "C", and "T" are used to designate the prompting, confirmation, and test modes respectively. These modes are explained in Chapter III.

The operator can listen to any stimulus specified in the test table through the use of one more mode of operation. This mode facilitates rapid experimentation with new stimuli, and is called the operator mode. The operator mode is entered by using the command "T_O¢". After giving this command the operator can move a pointer up or down through the test table locations by using two commands. The command

ITEM	SPECIFICATION
1	Spectra Table Code
2	Duration of Unmodulated Sound
3	Modulation Table Code
4	Duration of Modulation Using Envelope Specified by Item 3
5	Modulation Table Code
6	Duration of Modulation Using Envelope Specified by Item 5
7	Association Code

Figure 4.3 Items necessary to specify a stimulus

. . . . . .

E_15,0.3,1N,0.15,3F,0.15,32¢

Figure 4.4 Typical command to enter seven items

"I¢" will increment the pointer's position by one, and the command "D¢" will decrement the position by one.

The command "O¢" causes the stimulus specified by the current table location, the location at which the pointer is positioned, to be outputted through the digital-to-analog converter. The command "P¢" will cause the machine to print the associated code tag identified with the stimulus. A "CONTROL Z" is typed by the operator to exit from this mode.

The operator may cancel any of the commands explained in this chapter by typing the Teletype character "RUBOUT". Of course, this character must be typed before the particular command's terminating "carriage-return" is typed. This completes the discussion of the commands necessary to operate the teaching machine.

CHAPTER V

CONCLUSION

The teaching machine described in the preceding chapters solves at least some of the problems associated with tactual and audio perception investigations. The automation of the training periods and the precise bookkeeping performed by the computer greatly reduce the amount of manpower necessary to run experiments. The success of the machine in achieving this reduction indicates a possible extension of the system. It is certainly conceivable to expand the machine's capabilities to include the simultaneous training of several subjects. This would require a modest investment to cover the construction of several training terminals which would be used in place of the single Teletype console presently available. This extension is warranted only if the number of experiments would greatly exceed present expectations.

The synthetic vowels used in the machine are certainly adequate for many types of perception experiments; and, although the system was not intended to perform the function of a speech synthesizer, methods to generate a wider variety of stimuli would be of great use to researchers. The experience resulting from the development of the teaching system offers some indications of what would be required to extend the synthesis.

First, a method to string together basic sounds is needed, as well as the ability to specify periods of noise and silence. The prime consideration here is the development of an efficient language which can be used to build up a complex sound from many elementary ones. This simulation language should not only free the operator from tedious speech specification; but, in addition, should be able to display the resulting waveform on a cathode ray tube at any point in the building process. This display would not have to operate at the high rate required for actual output of the sound, and could exist in a time-sharing mode with the Teletype console.

Some form of analysis tool would also be desirable, and it is felt that the "Fast Fourier Transform", which is currently available in subroutine form, could be used for spectrum analysis of the synthetic sounds. Again, this information should be available in display form for rapid evaluation.

The most serious problem anticipated for an all-digital synthesizer is that of storage. To achieve good resolution of consonants a high output rate must be used, and this forces the waveform to be generated prior to output and stored. However, if the synthesizer was only required to generate speech sounds of limited duration, and the operator could tolerate a moderate delay between specification of the sound and actual output of the sound, it is felt that an appreciable amount of success in this area could be achieved with existing facilities at the Hybrid Computation Laboratory.

APPENDIX A

THE PROGRAM "CNTROL"

```
        .TITLE CNTROL
        /PROVIDES ENTRY TO ALL SUBROUTINES
        /PART OF SYNTHETIC VOWEL TESTER
        .GLOBL     VECADD,SOUADD,FLG
        .GLOBL     INPUT,FORYEA,COD,TESTO,TEST
ST      ISA+1Ø;    /INTERRUPT OFF
        IOF
        DZM FLG
        LAC (215
        7ØØ4Ø6;    /TLS
        7ØØ4Ø1;    /TSF
        JMP .-1
        LAC (212215;         JMS WPRT
        LAC (315317;         JMS WPRT
        LAC (3Ø43Ø5;         JMS WPRT
        LAC (212215;         JMS WPRT
        LAC (276;            JMS LPRT
        JMS BRRS;            /READ COMMAND
        DZM* XBR
        ISZ XBR;             ISZ TBR
        JMP .-3
        JMS BRRS
        7ØØ3Ø1
        JMP .-1
        7ØØ312
        SAD (232
        JMP ST
        SAD (215
        JMP .+7
        SAD (377
        JMP .-22
        DAC* XBR
        ISZ XBR
        ISZ TBR
        JMP .-14
        DAC* XBR
        LAC BR+1
        SAD (24Ø
        JMP .+2
        JMP W
        LAC BR
        SAD (311;            /I
```

CNTROL

```
JMP .+54
SAD (322;          /R
JMP R
SAD (320;          /P
JMP P
SAD (303;          /C
JMP C
SAD (324;          /T
JMP .+2
JMP W
LAC BR+2
SAD (317;          /O
JMP .+2
JMP .+11
JMS* TESTO
JMP .+6
.DSA DR
.DSA DRP
.DSA DRN
.DSA DRF
.DSA DRT
JMP ST
SAD (320;          /P
JMP .+6
SAD (303;          /C
JMP .+6
SAD (324;          /T
JMP .+6
JMP W;    /WHAT?
CLA
JMP .+4
LAC (1
JMP .+2
LAC (2
DAC .+10
JMS* TEST;         /TEST MODES P,C,T
JMP .+7
.DSA DR
.DSA DRP
.DSA DRN
.DSA DRF
.DSA DRT
0
JMP ST
CLQ
LAC BR+2
LLS+11
XOR BR+3
```

CNTROL

```
          DAC COD
          LAC BR+2
          SAD (324;           /T
          JMP T
          LAC BR+3
          SAD (316;           /N
          JMP .+12
          SAD (3Ø6;           /F
          JMP .+21
          LAC (DR
          DAC DRA
          JMS GA
          DAC VECADD
          DZM FLG
          JMS* INPUT
          JMP ST
          LAC (DRN
          DAC DRA
          LAC (1
          DAC FLG
          JMS GA
          DAC VECADD
          JMS* INPUT
          DZM FLG
          JMP ST
          LAC (DRF
          DAC DRA
          JMP .-11
T         LAC DRT
          DAC VECADD
          LAC (2
          DAC FLG
          JMS* INPUT
          DZM FLG
          JMP ST
C         LAC (DR;  /COMPUTE
          DAC DRA
          JMS GA
          DAC VECADD
          LAC (DRP
          DAC DRA
          JMS GA
          DAC SOUADD
          JMS* FORYEA
          JMP ST
R         LAC (212215;        /READ TAPE
          JMS WPRT
          JMS RED
```

CNTROL

```
          JMP ST
P         LAC BR+2;          /PUNCH TAPE
          SAD (324
          JMP .+37
          LAC BR+3
          SAD (316
          JMP .+15
          SAD (3Ø6
          JMP .+2Ø
          LAC (DR
          DAC DRA
          JMS GA
          DAC STADD
          LAC (-42
          DAC TALYP
          LAC (212215
          JMS WPRT
          JMS PUN
          JMP ST
          LAC  (DRN
          DAC DRA
          JMS GA
          DAC STADD
          JMP .+5
          LAC (DRF
          DAC DRA
          JMS GA
          DAC STADD
          LAC (-64
          DAC TALYP
          LAC (212215
          JMS WPRT
          JMS PUN
          JMP ST
          LAC DRT
          DAC STADD
          LAC (-374
          DAC TALYP
          LAC (212215
          JMS WPRT
          JMS PUN
          JMP ST
W         LAC (212215;       /WHAT?
          JMS WPRT
          LAC (32731Ø
          JMS WPRT
          LAC (3Ø1324
          JMS WPRT
```

CNTROL

```
        LAC (277
        JMS LPRT
        JMP ST
DBI     Ø
        JMS BTRS
        ISZ TBT
        CLA
        CLL
        TAD* XBT
        ISZ XBT
        DAC .+3
        LAC (12
        MUL
        HLT
        SZL!CLL
        JMP W
        LACQ
        ISZ TBT
        JMP .-13
        TAD* XBT
        JMP* DBI
BRRS    Ø
        LAC (-24
        DAC TBR
        LAC (BR
        DAC XBR
        JMP* BRRS
BTRS    Ø
        LAC (-6
        DAC TBT
        LAC (BT
        DAC XBT
        JMP* BTRS
ZBT     Ø
        JMS BTRS
        DZM* XBT
        ISZ XBT
        ISZ TBT
        JMP .-3
        JMS BTRS
        JMP* ZBT
LPRT    Ø
        7ØØ4Ø1
        JMP .-1
        7ØØ4Ø6
        JMP* LPRT
WPRT    Ø
        CLL
```

CNTROL

```
            LRS+11
            JMS LPRT
            CLA
            LLS+11
            JMS LPRT
            JMP* WPRT
SRI         Ø
            JMS ZBT
            LAC (BT+5
            DAC XSR
            LAC* TOP
            JMS DIG
            SPA!SNA
            JMP .+6
            TAD (-26Ø
            DAC* XSR
            LAC XSR
            TAD (-1
            DAC XSR
            LAC TOP
            TAD (-1
            SAD BOT
            JMP* SRI
            DAC TOP
            JMP .-16
DIG         Ø
            DAC SAV
            LAC (-12
            DAC TLA
            LAC (26Ø
            DAC TEMP
            LAC SAV
            SAD TEMP
            JMP .+5
            ISZ TEMP
            ISZ TLA
            JMP .-5
            CLA
            JMP* DIG
GA          Ø;          /GET ADDRESS
            JMS BRRS
            LAC XBR
            DAC BOT
            JMS COM
            LAC XBR
            DAC TOP
            JMS SRI
            JMS DBI
```

CNTROL

```
        TAD (-1
        TAD (200000
        TAD DRA
        DAC .+1
        HLT
        JMP* GA
COM     0
        LAC* XBR
        SAD (215
        JMP* COM
        ISZ XBR
        ISZ TBR
        JMP .-5
        JMP W
PUN     0;          /PUNCH
        JMS LT
        LAC STADD
        DAC BOT
        CLL
        CLQ
        LRS+14
        PSF
        JMP .-1
        PSB
        CLA
        LLS+6
        PSF
        JMP .-1
        PSB
        CLA
        LLS+6
        PSF
        JMP .-1
        PSB
        CLL
        LAC* BOT
        JMS LISP
        LRS+6
        JMS LISP
        LRS+6
        JMS LISP
        ISZ BOT
        ISZ TALYP
        JMP .-10
        LAC (300
        PSF
        JMP .-1
        PSA
```

CNTROL

```
        PSF
        JMP .-1
        JMS LT
        JMP* PUN
LT      Ø;          /LEADER-TRAILER
        LAC (-4ØØ
        DAC TALY
        PSA+1Ø
        PSF
        JMP .-1
        ISZ TALY
        JMP .-4
        JMP* LT
LISP    Ø
        CLL
        CLQ
        PSF
        JMP .-1
        PSB
        JMP* LISP
RED     Ø;          /READ
        CLL
        CLQ
        RSB
        RSF
        JMP .-1
        RRB
        DAC BOT
        JMS LRED
        AND (1
        SZA
        JMP .+7
        JMS LRED
        JMS LRED
        LACQ
        DAC* BOT
        ISZ BOT
        JMP .-11
        JMS LRED
        JMS LRED
        JMS LRED
        JMP* RED
LRED    Ø
        CLA
        RSA
        RSF
        JMP .-1
        RRB
```

CNTROL

```
            CLL
            LRS+6
            JMP* LRED
            .DEC;       /DIRECT FOR ALL TABLES
DR          1Ø1;        /SPECTRA TABLES
            135
            169
            2Ø3
            237
            271
            3Ø5
            339
            373
            4Ø7
            441
            475
            509
            543
            577
            611
            645
            679
DRN         713;        /ON MOD TABLES
            765
            817
            869
            921
DRF         973;        /OF MOD TABLES
            1Ø25
            1Ø77
            1129
            1181
DRP         1233;       /COMPUTED VOWELS
            1333
            1433
            1533
            1633
            1733
            1833
            1933
            2Ø33
            2133
            2233
            2333
            2433
            2533
            2633
            2733
```

CNTROL

```
            2833
            2933
DRT         3Ø33;        /TEST TABLE
            .OCT
PSF=7ØØ2Ø1
PSA=7ØØ2Ø4
RSF=7ØØ1Ø1
PSB=7ØØ244
RSA=7ØØ1Ø4
RSB=7ØØ144
RRB=7ØØ112
FLG         Ø
COD         Ø
XBR         Ø
TBR         Ø
XBT         Ø
TBT         Ø
TALY        Ø
TALYP       Ø
STADD       Ø
TOP         Ø
BOT         Ø
DRA         Ø
VECADD      Ø
SOUADD      Ø
TLA         Ø
XAV         Ø
XAS         Ø
SAV         Ø
TEMP        Ø
XSR         Ø
BR          .BLOCK 24
BT          .BLOCK 6
            Ø
            Ø
            Ø
            Ø
            Ø
            Ø
            Ø
            Ø
            .END ST
```

APPENDIX B

THE PROGRAM "INPUT"

```
        /PART OF A SYNTHETIC VOWEL
        /TRAINING SYSTEM
        /PERFORMS INPUT FOR ALL TABLES
        .GLOBL     INPUT,VECADD,COD,FLG
INPUT   Ø
        DZM FL;              DZM FLT
        JMS RS
START   LAC (212;            JMS LPRT
        LAC (215;            JMS LPRT
        JMS WTB;             JMS WTB
        LAC* FLG;            SNA
        JMP .+4
        SAD (1
        JMP .+7
        JMP .+13
        LAC* COD;            /HERE IF INPUT
        DAC TAB+4;           /TO SPECTRA
        LAC (TAB
        JMS PTB
        JMP .+1Ø
        LAC* COD;            /HERE IF INPUT
        DAC TAB+23;          /TO MODUL
        LAC (TAB+2Ø
        JMS PTB
        JMP .+3
        LAC (TAB+31;         /HERE IF INPUT
        JMS PTB;             /TO TEST TABLE
        LAC FL;   /CHECK TO SEE IF LISTING
        SZA
        JMP .+4
        JMP DN
RET     JMS BRRS;            /READ UNTIL CR,
        DZM* XBR;            /RB OR READ
        ISZ XBR;  /BUFFER OVERFLOWS
        ISZ TBR
        JMP .-3
        JMS BRRS
        7ØØ3Ø1;   /KSF
        JMP .-1
        7ØØ312;   /KRB
        SAD (232;            /CONTROL Z
        JMP* INPUT
```

INPUT

```
        SAD (215
        JMP .+7
        SAD (377;            /RUBOUT
        JMP DN
        DAC* XBR
        ISZ XBR
        ISZ TBR
        JMP .-14
        DAC* XBR
        LAC BR
        SAD (3Ø5;            /E
        JMP E
        SAD (32Ø;            /P
        JMP P
        SAD (316;            /N
        JMP N
        SAD (324;            /T
        JMP T
        SAD (314;            /L
        JMP L
        SAD (332;            /Z
        JMP Z
        SAD (3Ø1;            /A
        JMP A
        JMP W
P       LAC (212;            /CODING FROM P
        JMS LPRT;            /TO PD PRINTS
        LAC (215;            /ID TAGS FOR
        JMS LPRT;            /ALL PARAMETERS
        LAC* FLG
        SNA
        JMP .+4
        SAD (1
        JMP .+1Ø
        JMP .+13
        LAC XLIN;            /HERE IF PRINT
        SAD (Ø;   /FROM SPECTRA
        JMP .+16
        SAD (1
        JMP .+17
        JMP .+21
        LAC XLIN;            /HERE IF PRINT
        SAD (Ø;   /FROM MODUL
        JMP .+27
        JMP .+31
        LAC XLIN;            /HERE IF PRINT
        SAD (Ø;   /FROM TEST TABLE
        JMP .+34
```

INPUT

```
        SAD (1
        JMP .+35
        JMP .+37
        LAC (TAB+6;          /HERE IF PRINT
        JMS PTB;  /FREQFØ
        JMP PD
        LAC (TAB+12;         /HERE IF PRINT
        JMS PTB;  /AMPDC
        JMP PD
        LAC (TAB+15;         /HERE IF PRINT
        JMS PTB;             /AMPF
        LAC (2;   /PRINT NUMBER OF HARMONIC
        DAC FLT
        LAC XLIN
        TAD (-2
        JMS PI
        DZM FLT
        JMP PD
        LAC (TAB+25;         /HERE IF PRINT
        JMS PTB;  /POINTS
        JMP PD
        LAC (3;   /HERE IF PRINT POINT NUMBER
        DAC FLT
        LAC XLIN
        JMS PI
        DZM FLT
        JMP PD
        LAC (TAB+37;         /HERE IF PRINT
        JMS PTB;  /ENTRIES
        JMP PD
        LAC (TAB+43;         /HERE IF PRINT
        JMS PTB;  /TRIALS
        JMP PD
        LAC (3;   /HERE IF PRINT ENTRY NUMBER
        DAC FLT
        LAC XLIN
        TAD (-1
        JMS PI
        DZM FLT
        JMP PD
PD      JMS WTB
        LAC* FLG
        SNA
        JMP .+15
        SAD (1
        JMP .+7
        LAC XLIN;            /HERE IF PRINT
        SAD (Ø;   /FROM TEST TABLE
```

```
        JMP .+1Ø
        SAD (1
        JMP .+6
        JMP PTV
        LAC XLIN;          /HERE IF PRINT
        SZA;     /FROM MODUL TABLE
        JMP .+11
        JMP .+1
        DZM FLT; /HERE IF PRINT INTEGER
        LAC* XLOC
        JMS PI
        LAC FL
        SZA
        JMP L+7
        JMP DN
        LAC* XLOC;         /HERE IF PRINT
        TAD RU;  /FRACTION-ROUND UP
        JMS PF
        LAC FL
        SZA
        JMP L+7
        JMP DN
PTV     LAC (2;  /PRINT A 6-ENTRY TEST VECTOR
        DAC FLT
        LAC* XLOC
        CLL
        CLQ
        LRS+1Ø
        SPA!SNA
        JMP .+2
        JMS PI
        DZM FLT
        LAC (254
        JMS LPRT
        LAC (4
        DAC FLT
        ISZ XLOC
        LAC* XLOC
        SPA!SNA
        JMP .+2
        JMS PI
        DZM FLT
        LAC (254
        JMS LPRT
        LAC XLOC
        TAD (-1
        DAC XLOC
        LAC (2
```

INPUT

```
DAC FLT
LAC* XLOC
CLL
CLQ
LRS+4
AND (17
SPA!SNA
JMP .+4
JMS PI
LAC (316
JMS LPRT
DZM FLT
LAC (254
LMS LPRT
ISZ XLOC
ISZ XLOC
LAC (4
DAC FLT
LAC* XLOC
SPA!SNA
JMP .+2
JMS PI
DZM FLT
LAC (254
JMS LPRT
LAC (2
DAC FLT
LAC XLOC
TAD (-2
DAC XLOC
LAC* XLOC
AND (17
SPA!SNA
JMP .+4
JMS PI
LAC (3Ø6
JMS LPRT
DZM FLT
LAC (254
JMS LPRT
LAC (4
DAC FLT
ISZ XLOC
ISZ XLOC
ISZ XLOC
LAC* XLOC
SPA!SNA
JMP .+2
```

INPUT

```
          JMS PI
          DZM FLT
          LAC (254
          JMS LPRT
          ISZ XLOC
          LAC (3
          DAC FLT
          LAC* XLOC
          JMS PI
          DZM FLT
          LAC XLOC
          TAD (-4
          DAC XLOC
          LAC FL;     /CHECK TO SEE IF LISTING
          SZA
          JMP L+7
          JMP DN
PF        Ø;          /PRINT A FRAC
          JMS BDF
          LAC BT
          SAD (377777
          JMP .+15
          JMS BTA
          LAC (F
          JMS PTB
          JMS BTRS
          ISZ TBT
          ISZ TBT
          LAC* XBT
          JMS LPRT
          ISZ XBT
          ISZ TBT
          JMP .-4
          JMP* PF
          LAC (0
          JMS PTB
          JMP* PF
E         JMS BRRS;             /ENTER
          LAC* FLG
          SPA!RNA
          JMP .+11
          SAD (1
          JMP .+12
          LAC XLIN;             /TEST TABLE
          SAD (Ø
          JMP .+4
          SAD (1
          JMP .+2
```

INPUT

```
JMP ET
JMS GI;    /HERE IF ENTER INTEGER
DAC* XLOC
JMP DN
LAC XLIN;            /MODUL TABLE
SZA
JMP .+2
JMP .-6
LAC XBR;   /ENTERS FRACTION
DAC BOT
JMS COM
SZA
JMP W
LAC XBR
DAC MD
ISZ XBR
JMS COM
LAC XBR
DAC TOP
LAC MD
TAD (-1
TAD (2ØØØØØ
DAC .+1
HLT
SAD (261
JMP .+25
LAC (BT+5
DAC XSR
JMS ZBT
LAC* MD
JMS DIG
SPA!SNA
JMP .+6
TAD (-26Ø
DAC* XSR
LAC XSR
TAD (-1
DAC XSR
ISZ MD
LAC MD
SAD TOP
JMP .+2
JMP .-15
JMS DBF
DAC* XLOC
JMP DN
LAC (377777
DAC* XLOC
```

INPUT

```
        JMP DN
ET      JMS GI;     /ENTERS TEST VECTOR
        CLL
        CLQ
        LLS+1Ø
        DAC* XLOC
        ISZ XLOC
        JMS TI
        DAC* XLOC
        LAC XLOC
        TAD (-1
        DAC XLOC
        JMS GI
        CLL
        CLQ
        LLS+4
        XOR* XLOC
        DAC* XLOC
        ISZ XLOC
        ISZ XLOC
        JMS TI
        DAC* XLOC
        LAC XLOC
        TAD (-2
        DAC XLOC
        JMS GI
        XOR* XLOC
        DAC* XLOC
        ISZ XLOC
        ISZ XLOC
        ISZ XLOC
        JMS TI
        DAC* XLOC
        ISZ XLOC
        JMS GI
        DAC* XLOC
        LAC XLOC
        TAD (-4
        DAC XLOC
        JMP DN
RS      Ø;          /RESETS INDICES
        DZM XLIN
        LAC* VECADD
        DAC XLOC
        LAC* FLG
        SZA
        JMP .+4
        LAC (-42
```

INPUT

```
          DAC TLG
          JMP* RS
          SAD (2
          JMP .+4
          LAC (-64
          DAC TLG
          JMP* RS
          LAC (-64
          DAC TLG
          JMP* RS
BRRS      Ø;          /RESET THE READER BUFFER
          LAC (-7Ø
          DAC TBR
          LAC (BR
          DAC XBR
          JMP* BRRS
BTRS      Ø;          /RESET THE BCD BUFFER
          LAC (-6
          DAC TBT
          LAC (BT
          DAC XBT
          JMP* BTRS
ZBT       Ø;          /ZERO THE BCD BUFFER
          JMS BTRS
          DZM* XBT
          ISZ XBT
          ISZ TBT
          JMP .-3
          JMS BTRS
          JMP* ZBT
LPRT      Ø;          /WAIT AND PRINT A CHARACTER
          7ØØ4Ø1;     /TSF
          JMP .-1
          7ØØ4Ø6;     /TLS
          JMP* LPRT
PTB       Ø;          /PRINT UNTIL A BLANK
          DAC TL
          CLL
          CLQ
          LAC* TL
          LRS+11
          SNA
          JMP* PTB
          JMS LPRT
          LLS+11
          AND (777
          SNA
          JMP* PTB
```

INPUT

```
          JMS LPRT
          ISZ TL
          JMP .-15
WTB       Ø;          /PRINT A TAB
          LAC (211
          JMS LPRT
          LAC (-26ØØØØ
          DAC TL
          NOP
          ISZ TL
          JMP .-2
          JMP* WTB
COM       Ø;          /SEARCH FOR A COMMA, CR,
          LAC* XBR;             /OR PERIOD
          SAD (254
          JMP* COM
          SAD (215
          JMP* COM
          SAD (256
          JMP .+5
          ISZ XBR
          ISZ TBR
          JMP .-11
          JMP W
          CLA;        /IF PERIOD EXIT WITH Ø IN AC
          JMP* COM
SRP       Ø;          /SEARCH FOR A TIME INTEGER
          JMS ZBT
          LAC (BT+2
          DAC XSR
          LAC* MD
          JMS DIG
          SPA!SNA
          JMP .+4
          TAD (-26Ø
          DAC* XSR
          ISZ XSR
          DZM* MD
          ISZ MD
          LAC MD
          SAD TOP
          JMP .+2
          JMP .-14
          DZM* MD
          JMP* SRP
SRI       Ø;          /SEARCH FOR AN INTEGER
          LAC* TOP
          JMS DIG
```

INPUT

```
        SPA!SNA
        JMP .+6
        TAD (-260
        DAC* XSR
        LAC XSR
        TAD (-1
        DAC XSR
        DZM* TOP
        LAC TOP
        TAD (-1
        DAC TOP
        SAD BOT
        JMP* SRI
        JMP .-17
DIG     0;          /ENTER WITH CHAR. IN AC
        DAC SAV;    /IF DIGIT EXIT WITH DIGIT IN AC
        LAC (-12
        DAC TL
        LAC (260
        DAC TEMP
        LAC SAV
        SAD TEMP
        JMP .+5
        ISZ TEMP
        ISZ TL
        JMP .-5
        CLA;        /EXIT WITH 0 IN AC
        JMP* DIG
GI      0;          /GET AN INTEGER
        LAC XBR
        DAC BOT
        JMS COM
        SPA!SNA
        JMP W
        LAC XBR
        DAC TOP
        LAC (BT+5
        DAC XSR
        JMS ZBT
        JMS SRI
        JMS DBI
        JMP* GI
PI      0;          /PRINT A NUMBER
        JMS BDI
        JMS BTA
        LAC FLT
        SNA
        JMP .+7
```

INPUT

```
          SAD (1
          JMP .+3Ø
          SAD (2
          JMP .+24
          SAD (3
          JMP .+15
          LAC BT+5
          JMS LPRT
          LAC BT+4
          JMS LPRT
          LAC FLT
          SAD (4
          JMP .+2
          JMP .+3
          LAC (256
          JMS LPRT
          LAC BT+3
          JMS LPRT
          LAC BT+2
          JMS LPRT
          LAC FLT
          SAD (4
          JMP* PI
          LAC BT+1
          JMS LPRT
          LAC BT
          JMS LPRT
          JMP* PI
TI        Ø;         /GET A TIME INTEGER
          LAC XBR
          DAC BOT
          JMS COM
          SZA
          JMP .+13
          LAC XBR
          DAC MD
          ISZ XBR
          JMS COM
          SPA!SNA
          JMP W
          LAC XBR
          DAC TOP
          JMS SRP
          JMP .+4
          LAC XBR
          DAC TOP
          JMS ZBT
          LAC (BT+1
```

INPUT

```
        DAC XSR
        JMS SRI
        JMS DBI
        JMP* TI
DBI     Ø;          /ENT WITH BCD IN BT MS-LS
        JMS BTRS;             /(B*1Ø+(B-1)) 1Ø
        ISZ TBT;    /EXIT WITH BIN IN AC
        CLA
        CLL
        TAD* XBT
        ISZ XBT
        DAC .+3
        LAC (12
        MUL
        HLT
        SZL!CLL
        JMP W
        LACQ
        ISZ TBT
        JMP .-13
        TAD* XBT
        JMP* DBI
DBF     Ø
        JMS BTRS;             /ENTER WITH BCD
        LAC BT+5;             /IN BT LS-MS
        SAD (377777;          /((LS/1Ø)+LS-1)1Ø
        JMP* DBF;             /EXIT WITH BIN IN AC
        CLQ;        /MS IS 377777
        CLL
        LAC (12
        DAC .+6
        LACQ
        LRS+21
        TAD* XBT
        LRS+1
        DIV
        HLT
        SZL!CLL
        JMP W
        ISZ XBT
        ISZ TBT
        JMP .-15
        LACQ
        JMP* DBF
BDI     Ø;          /ENTER WITH BIN IN AC
        DAC SAV;    /PUT BCD IN BUFTEM LS-MS
        JMS BTRS;             /B/1Ø=Q1+R1  (LS)
        ISZ TBT;    /Q1/1Ø=Q2+R2
```

INPUT

```
        CLL
        LAC (12
        DAC .+3
        LAC SAV
        IDIV
        HLT
        SZL!CLL
        JMP W
        DAC* XBT
        LACQ
        DAC SAV
        LAC (12
        DAC .+3
        LAC SAV
        IDIV
        HLT
        SZL!CLL
        JMP W
        ISZ XBT
        ISZ TBT
        JMP .-14
        DAC* XBT
        JMP* BDI
BDF     Ø;          /ENTER WITH BIN IN AC
        DAC SAV
        JMS BTRS;            /BCD IN BT MS-LS
        ISZ TBT;  /.B*1Ø=I1.F1 (I1 IS MS)
        LAC SAV;  /.F1*1Ø=I2.F2 ETC
        SAD (377777
        JMP .+34
        CLL
        LAC (12
        DAC .+3
        LAC SAV
        MUL
        HLT
        SZL!CLL
        JMP W
        LLS+1
        DAC* XBT
        CLA!CLL
        LRS+1
        LACQ
        DAC SAV
        LAC (12
        DAC .+3
        LAC SAV
        MUL
```

INPUT

```
         HLT
         SZL!CLL
         JMP W
         ISZ XBT
         ISZ TBT
         JMP .-17
         LLS+1
         DAC* XBT
         JMP* BDF
         DAC BT
         JMP* BDF
BTA      Ø;          /ADD 26Ø TO THE BCD BUFFER
         JMS BTRS
         LAC* XBT
         TAD (26Ø
         DAC* XBT
         ISZ XBT
         ISZ TBT
         JMP .-5
         JMP* BTA
Z        JMS RS;     /ZERO THE TABLE
         LAC (1
         DAC FL
         LAC* FLG
         SAD (2
         JMP .+1Ø
         DZM* XLOC
         JMS NE
         LAC FL
         SZA
         JMP .-4
         JMS RS
         JMP DN
         DZM* XLOC;          /HERE IF TEST
         JMS NE
         DZM* XLOC
         JMS NE
         LAC (-5
         DAC TL
         DZM* XLOC
         ISZ XLOC
         ISZ TL
         JMP .-3
         LAC XLOC
         TAD (-5
         DAC XLOC
         JMS NE
         LAC FL
```

INPUT

```
          SZA
          JMP .-14
          JMS RS
          JMP DN
T         JMS RS;   /TOP POINTER
          JMP DN
W         LAC (212;           /WHAT?
          JMS LPRT
          LAC (215
          JMS LPRT
          LAC (WH
          JMS PTB
          JMP DN
L         JMS RS;   /LIST
          LAC (1
          DAC FL
          JMP START
          LAC* XLOC
          SZA
          JMP P
          JMS NE
          LAC FL
          SZA
          JMP .-6
          JMS RS
          JMP DN
N         JMS NE;   /NEXT
          JMP DN
DN        LAC (C;   /DONE
          JMS PTB
          JMP RET
NE        Ø;        /GET NEXT
          LAC* FLG
          SAD (2
          JMP .+2Ø
          ISZ XLOC
          ISZ XLIN
          ISZ TLG
          JMP* NE
          LAC FL
          SNA
          JMP .+3
          DZM FL
          JMP* NE
          LAC (212
          JMS LPRT
          LAC (B
          JMS PTB
```

INPUT

```
          JMS RS
          JMP DN
          LAC XLIN
          SAD (Ø
          JMP .-21
          SAD (1
          JMP .-23
          LAC XLOC
          TAD (5
          DAC XLOC
          JMP .-26
A         JMS BRRS;            /ADVANCE
          JMS GI
          SPA!SNA
          JMP DN
          CMA;      TAD (1
          DAC TL
          JMS NE
          ISZ TL
          JMP .-2
          JMP DN
TAB       32332Ø;   /SPECTRA
          3Ø53Ø3
          324322
          3Ø124Ø
          Ø
          Ø
          3Ø6322;   /FREQFØ
          3Ø5321
          3Ø626Ø
          ØØØØØØ
          3Ø1315;   /AMPDC
          32Ø3Ø4
          3Ø3ØØØ
          3Ø1315;   /AMPF
          32Ø3Ø6
          Ø
          315317;   /MODUL
          3Ø4325
          31424Ø
          Ø
          Ø
          32Ø317;   /POINTS
          311316
          324323
          Ø
          3243Ø5;   /TEST TABLE
          323324
```

INPUT

```
                240240
                324301
                302314
                305000
                305316;    /ENTRIES
                324322
                311305
                323000
                324322;    /TRIALS
                311301
                314323
                0
F               260256;    /0
                0
O               261256;    /1.0
                260000
C               212215
                276000
WH              327310
                301324
                277000
B               305316;    /END OF TABLE
                304240
                317306
                240324
                301302
                314305
                255255
                320317;    /POINTER HAS BEEN MOVED
                311316;    /TO TOP
                324305
                322240
                310301
                323240
                302305
                305316
                240315
                317326
                305304
                240324
                317240
                324317
                320000
XLIN            0
XLOC            0
XBR             0
TBR             0
XBT             0
```

INPUT

```
TBT     Ø
XSR     Ø
TEMP    Ø
TOP     Ø
BOT     Ø
MD      Ø
SAV     Ø
RU      2
TL      Ø
FL      Ø
FLT     Ø
TLG     Ø
BR      .BLOCK 7Ø
BT      .BLOCK 6
BTO     .BLOCK 6
        .END
```

APPENDIX C

THE PROGRAM "FORYEA"

```
        /SUBROUTINE TO COMPUTE
        /ONE CYCLE OF SYNTHETIC
        /VOWEL FOR OUTPUT AT
        /1ØKHZ RATE
        /CALLED BY CNTROL
FORYEA  Ø
        .GLOBL    VECADD,SOUSCL,STRPTS,FORYEA
        .GLOBL    ZSCALE,.AK,.AL,COS
        .GLOBL    .AG,.AH,.AX,.AW,.AI,.AJ
START   LAC* VECADD;        DAC FRØ
        LAC* FRØ
        SPA!SNA;            JMP* FORYEA
        DAC .+4
        CLL
        LAC (2342Ø
        IDIV
        HLT
        LACQ
        DAC STRPTS
        LAC (2342Ø
        JMS* .AW
        JMS .AH
        .DSA OUTRAT
        LAC* VECADD;        DAC INDEX1
        LAC (FREQFØ
        DAC INDEX2
        LAC (-42
        DAC TALY
LOOP1   LAC* INDEX1
        JMS* .AW
        JMS* .AH
        .DSA INDEX2+A
        ISZ INDEX2;         ISZ INDEX2
        ISZ INDEX1;         ISZ TALY
        JMP LOOP1
        LAC (1
        JMS* .AW
        JMS* .AH
        .DSA ONE
        JMS* .AG
        .DSA ONE
        JMS* .AL
```

FORYEA

```
          .DSA FREQFØ
          JMS* .AH
          .DSA PERIOD
          JMS* .AG
          .DSA ONE
          JMS* .AL
          .DSA OUTRAT
          JMS* .AH
          .DSA DELTAT
          JMS* .AG
          .DSA TWOPI
          JMS* .AK
          .DSA FREQFØ
          JMS* .AK
          .DSA DELTAT
          JMS* .AH
          .DSA DELTAW
          CLA
          JMS* .AW
          JMS* .AH
          .DSA ZERO
          JMS* .AG
          .DSA ZERO
          JMS* .AH
          .DSA DEXTIM
          LAC (SOUND
          DAC DEXLOC
          LAC STRPTS
          CMA;      TAD (1
          DAC INDEX2
MORPTS    LAC (-4Ø;           /GET ANOTHER POINT
          DAC INDEX1
          JMS* .AG
          .DSA ONE
          JMS* .AH
          .DSA DEXHAR
          LAC (AMPFØ
          DAC DXAMPF
          JMS* .AG
          .DSA AMPDC
          JMS* .AH
          .DSA DEXLOC+A
TEN       JMS* .AG;           /GET ALL HARMONICS
          .DSA DEXTIM
          JMS* .AK
          .DSA DEXHAR
          JMS* .AH
          .DSA TEMP
```

FORYEA

```
        JMS* COS
        JMP .+2
        .DSA TEMP
        JMS* .AK
        .DSA DXAMPF+A
        JMS* .AI
        .DSA DEXLOC+A
        JMS* .AH
        .DSA DEXLOC+A
        ISZ DXAMPF
        ISZ DXAMPF
        JMS* .AG
        .DSA DEXHAR
        JMS* .AI
        .DSA ONE
        JMS* .AH
        .DSA DEXHAR
        ISZ INDEX1
        JMP TEN
        ISZ DEXLOC
        ISZ DEXLOC
        JMS* .AG
        .DSA DEXTIM
        JMS* .AI
        .DSA DELTAW
        JMS* .AH
        .DSA DEXTIM
        ISZ INDEX2
        JMP MORPTS
        LAC (SOUND
        DAC SOUSCL
        JMS* ZSCALE
        JMP* FORYEA
        HLT
FREQFØ  Ø
        Ø
AMPDC   Ø
        Ø
AMPFØ   .BLOCK 1ØØ
OUTRAT  Ø
        Ø
TWOPI   331ØØ3
        311Ø37
ONE     Ø
        Ø
ZERO    Ø
        Ø
PERIOD  Ø
```

```
    FORYEA


            Ø
DELTAT      Ø
            Ø
DELTAW      Ø
            Ø
DEXHAR      Ø
            Ø
DEXTIM      Ø
            Ø
TEMP        Ø
            Ø
INDEX1      Ø
INDEX2      Ø
TALY        Ø
STRPTS      Ø
DXAMPF      Ø
DEXLOC      Ø
FRØ         Ø
SOUSCL      Ø
SOUND       .BLOCK 312
A=4ØØØØØ
            .END
```

APPENDIX D

THE PROGRAM "ZSCALE"

```
        /SUBROUTINE TO SCALE
        /THE OUTPUT OF
        /FORYEA FOR D-A
        /CALLED BY FORYEA
        .GLOBL    ZSCALE,SOUADD,STRPTS,SOUSCL
ZSCALE  Ø
        LAC* SOUSCL;        DAC RELADD
        LAC* SOUADD;        DAC FIXADD
        LAC* STRPTS
        CMA;      TAD (1
        DAC TALY1;          DAC TALY2
        DZM LARGEX
LOOP1   LAC* RELADD
        AND (777
        TAD (7774ØØ
        XOR (7774ØØ
        DAC TEMP
        CMA;      TAD (1
        TAD LARGEX
        SPA!CLL;            JMP REPLAC
CONTIN  ISZ RELADD;         ISZ RELADD
        ISZ TALY1
        JMP LOOP1
        JMP .+4
REPLAC  LAC TEMP
        DAC LARGEX
        JMP CONTIN
/SCALE AND FIX
        LAC* SOUSCL;        DAC RELADD
LOOP2   LAC* RELADD
        AND (777
        TAD (7774ØØ
        XOR (7774ØØ
        CMA;      TAD (-21
        TAD LARGEX
        SMA;      CLA
        TAD (LRSS+22
        DAC EXPON
        ISZ RELADD
        LAC* RELADD
        RCL;      SZL
        CMA;      RAR
```

ZSCALE

```
EXPON   HLT
        DAC* FIXADD
        ISZ FIXADD
        ISZ RELADD
        ISZ TALY2
        JMP LOOP2
        JMP* ZSCALE
RELADD  Ø
FIXADD  Ø
TALY1   Ø
TALY2   Ø
LARGEX  Ø
TEMP    Ø
        .END
```

APPENDIX E

THE PROGRAM "TESTO"

```
        /USED FOR OUTPUT IN
        /OPERATOR MODE
        /CALLED BY CNTROL
        .GLOBL     TESTO,TALKM,.DA
TESTO   CAL Ø
        JMS* .DA
        JMP .+6
AC      Ø;         /DR
        Ø;         /DRP
        Ø;         /DRN
        Ø;         /DRF
        Ø;         /DRT
        JMS TRS
        LAC (CR;            JMS PTB
        LAC (-16;           DAC SAV
        LAC (24Ø;           JMS LPRT
        ISZ SAV
        JMP .-3
        LAC (HD;            JMS PTB
        LAC (CR;            JMS PTB
RET     JMS BRS
        DZM* XBR
        ISZ XBR
        ISZ TBR
        JMP .-3
        JMS BRS
        7ØØ3Ø1;    /KSF
        JMP .-1
        7ØØ312;    /KRB
        SAD (232
        JMP* TESTO
        SAD (215
        JMP .+5
        DAC* XBR
        ISZ XBR
        ISZ TBR
        JMP .-12
        DAC* XBR
        LAC (377
        JMS MT
        SZA
```

TESTO

```
          JMP RET
          LAC (311
          JMS MT
          SZA
          JMP I
          LAC (3Ø4
          JMS MT
          SZA
          JMP D
          LAC (317
          JMS MT
          SZA
          JMP O
          LAC (32Ø
          JMS MT
          SZA
          JMP P
          JMP RET-2
I         ISZ XLIN;          /INCREMENT POINTER
          LAC TLC
          TAD (5
          DAC TLC
          ISZ TLT
          JMP .+2
          JMS TRS
          JMS SU
          JMP RET-2
D         LAC XLIN;          /DECREMENT POINTER
          TAD (-1
          SPA
          JMP RET-2
          LAC TLC
          TAD (-5
          DAC TLC
          LAC TLT
          TAD (-1
          DAC TLT
          JMS SU
          JMP RET-2
P         LAC (CR;  /PRINT CODE TAG
          JMS PTB
          LAC TLC
          TAD (4
          DAC TLC
          LAC (V
          JMS PTB
          LAC* TLC
          JMS PI
```

TESTO

```
            LAC TLC
            TAD (-4
            DAC TLC
            JMP RET-2
O           JMS SU;    /OUTPUT A SOUND
            LAC (CR
            JMS PTB
            LAC (-64
            DAC 7
            7ØØØ44;    /CLON
            7ØØØØ1;    /CLSF
            JMP .-1
            7ØØØØ4;    /CLOF
            JMS* TALKM
            JMP .+1Ø
            .DSA B+A
            .DSA B+1+A
            .DSA B+2+A
            .DSA B+3+A
            .DSA B+4
            .DSA B+5
            .DSA B+6
            JMP RET-2
BRS         Ø
            LAC (-1Ø
            DAC TBR
            LAC (BR
            DAC XBR
            JMP* BRS
BTS         Ø
            LAC (-6
            DAC TBT
            LAC (BT
            DAC XBT
            JMP* BTS
TRS         Ø
            LAC* AC+4
            DAC TLC
            LAC* TLC
            SPA!SNA
            JMP* TESTO
            CMA;       TAD (1
            DAC TLT
            ISZ TLC
            ISZ TLC
            DZM XLIN
            JMP* TRS
LPRT        Ø
```

TESTO

```
           7ØØ4Ø1;    /TSF
           JMP .-1
           7ØØ4Ø6
           JMP* LPRT
PTB        Ø
           DAC SAV
           CLL
           LAC* SAV
           LRS+11
           SNA
           JMP* PTB
           JMS LPRT
           LLS+11
           AND (777
           SNA
           JMP* PTB
           JMS LPRT
           ISZ SAV
           JMP .-14
PI         Ø
           JMS BDI
           LAC BT+2
           TAD (26Ø
           JMS LPRT
           LAC BT+1
           TAD (26Ø
           JMS LPRT
           LAC BT
           TAD (26Ø
           JMS LPRT
           JMP* PI
MT         Ø
           DAC SAV
           JMS BRS
           LAC* XBR
           SAD SAV
           JMP* MT
           ISZ XBR
           ISZ TBR
           JMP .-5
           JMS BRS
           CLA
           JMP* MT
SU         Ø;         /SET UP FOR TALKM
           CLL
           CLQ
           LAC* TLC
           LRS+1Ø
```

TESTO

```
SPA!SNA
JMP RET
TAD (-1
TAD AC
TAD (2ØØØØØ
DAC .+1
HLT
DAC B;    /VECADD
CLL
CLQ
LAC* TLC
LRS+1Ø
TAD (-1
TAD AC+1
TAD (2ØØØØØ
DAC .+1
HLT
DAC B+1;  /SOUADD
CLL!CLA
LLS+4
SPA!SNA
JMP .+1Ø
TAD (-1
TAD AC+2
TAD (2ØØØØØ
DAC .+1
HLT
DAC B+2;  /ONADD
JMP .+2
DZM B+2
CLL!CLA
LLS+4
SPA!SNA
JMP .+1Ø
TAD (-1
TAD AC+3
TAD (2ØØØØØ
DAC .+1
HLT
DAC B+3
JMP .+2
DZM B+3
ISZ TLC
LAC* TLC
DAC B+4;  /STRTIM
ISZ TLC
LAC* TLC
DAC B+5;  /ONTIM
```

TESTO

```
            ISZ TLC
            LAC* TLC
            DAC B+6;  /OPTIM
            LAC TLC
            TAD (-3
            DAC TLC
            JMP* SU
BDI         Ø
            DAC SAV
            JMS BTS
            ISZ TBT
            CLL
            LAC (12
            DAC .+3
            LAC SAV
            IDIV
            HLT
            SZL!CLL
            JMP RET
            DAC* XBT
            LACQ
            DAC SAV
            LAC (12
            DAC .+3
            LAC SAV
            IDIV
            HLT
            SZL!CLL
            JMP RET
            ISZ XBT
            ISZ TBT
            JMP .-14
            DAC* XBT
            JMP* BDI
CR          212215
            276ØØØ
HD          3243Ø5;   /TEST OPERATOR MODE
            323324
            24Ø255
            24Ø317
            32Ø3Ø5
            3223Ø1
            324317
            32224Ø
            315317
            3Ø43Ø5
            Ø
V           326ØØØ
```

TESTO

```
A=4ØØØØØ
XLIN     Ø
TLC      Ø
TLT      Ø
XBR      Ø
TBR      Ø
XBT      Ø
TBT      Ø
SAV      Ø
B        .BLOCK 7
BR       .BLOCK 1Ø
BT       .BLOCK 6
         .END
```

APPENDIX F

THE PROGRAM "TEST"

```
        /SUBROUTINE TO ADMINISTER
        /TESTS - 3 MODES
        /PROMPTING, CONFIRMATION,
        /AND TEST
        /CALLED BY CNTROL
        .GLOBL    TALKM,TEST,.DA
TEST    CAL Ø
        JMS* .DA
        JMP .+7
        Ø;        /DR
        Ø;        /DRP
        Ø;        /DRN
        Ø;        /DRF
        Ø;        /DRT
        Ø;        /FLAG
        JMS TRS
        JMS TSV
        LAC TLC
        TAD (-1
        DAC TLC
        LAC* TLC
        SPA!SNA
        JMP* TEST
        CMA;      TAD (1
        DAC TLL
        JMS RST
        LAC (RES
        DAC LLC;            DZM LLN
        JMS WRS
        DZM* WLC
        ISZ WLC
        ISZ TLW
        JMP .-3
        JMS WRS
        DZM WFL;            DZM XFL
        LAC (-3
        DAC TLR
        LAC (CR;            JMS PTB
        LAC (1Ø;            JMS SPC
        LAC (1Ø;            JMS SPC
        LAC (CR
        TAD .+2;  /TEST MODE
```

```
   TEST


         JMP .+2
         .DSA 26
         JMS PTB
         LAC (CR
         TAD .+2
         JMP .+2
         .DSA 34;  /P,C,T,
         TAD TEST+1Ø
         JMS PTB
         JMP O
         LAC (CR
         JMS PTB
RET      JMS BRS;  /READ
         DZM* XBR
         ISZ XBR
         ISZ TBR
         JMP .-3
         JMS BRS
         7ØØ3Ø1;   /KSF
         JMP .-1
         7ØØ312;   /KRB
         SAD (232
         JMP* TEST
         SAD (215
         JMP .+5
         DAC* XBR
         ISZ XBR
         ISZ TBR
         JMP .-12
         DAC* XBR
         JMS BRS
         LAC .+2
         JMP .+2
         377;      /RB
         JMS MT
         SZA
         JMP RET-2
         JMS BRS
         LAC .+2
         JMP .+2
         323;      /S
         JMS MT
         SZA
         JMP S
         JMS BRS
         LAC .+2
         JMP .+2
         322;      /R
```

```
TEST

        JMS MT
        SZA
        JMP .+11
        LAC (CR
        JMS PTB
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 37;  /WHAT?
        JMS PTB
        JMP RET-2
        LAC TEST+10
        SAD (2
        JMP .-12
        JMP O
O       CLL;      /OUTPUT A VOWEL
        CLQ
        LAC* TLC
        LRS+10
        SPA!SNA
        JMP* TEST
        TAD (-1
        TAD TEST+3;        /DR
        TAD .+2
        JMP .+2
        200000
        DAC .+1
        HLT
        DAC B;    /VECADD
        CLL
        CLQ
        LAC* TLC
        LRS+10
        TAD (-1
        TAD TEST+4
        TAD .+2
        JMP .+2
        200000
        DAC .+1
        HLT
        DAC B+1;  /SOUADD
        CLL!CLA
        LLS+4
        SPA!SNA
        JMP .+12
        TAD (-1
        TAD TEST+5
        TAD .+2
```

TEST

```
JMP .+2
200000
DAC .+1
HLT
DAC B+2;  /ONADD
JMP .+2
DZM B+2
CLL!CLA
LLS+4
SPA!SNA
JMP .+12
TAD (-1
TAD TEST+6
TAD .+2
JMP .+2
200000
DAC .+1
HLT
DAC B+3;  /OFADD
JMP .+2
DZM B+3
ISZ TLC
LAC* TLC
DAC B+4;  /STRTIM
ISZ TLC
LAC* TLC
DAC B+5;  /ONTIM
ISZ TLC
LAC* TLC
DAC B+6;  /OFTIM
ISZ TLC
CLL
CLQ
LAC* TLC
LLS+12
DAC* LLC
LAC TLC
TAD (-4
DAC TLC
LAC (CR
JMS PTB
LAC (CR+2;          /LISTEN
JMS PTB
LAC TEST+10
SPA!SNA
JMP .+2
JMP .+13
LAC (CR
```

TEST

```
         TAD .+2
         JMP .+2
         .DSA 14
         JMS PTB
         CLL
         CLQ
         LAC* LLC
         LRS+12
         JMS PI
         LAC (-64
         DAC 7
         7ØØØ44;    /CLON
         7ØØØØ1;    /CLSF
         JMP .-1
         7ØØØØ4;    /CLOF
         JMS* TALKM
         JMP .+1Ø
         .DSA B+A
         .DSA B+1+A
         .DSA B+2+A
         .DSA B+3+A
         .DSA B+4
         .DSA B+5
         .DSA B+6
         LAC (CR
         JMS PTB
         LAC (CR
         TAD .+2
         JMP .+2
         .DSA 15;   /ANSWER
         JMS PTB
         JMP RET
S        JMS BRS;   /GET RESPONSE
         LAC XBR
         DAC SV1
         LAC (215
         JMS MT
         SPA!SNA
         JMP RET
         LAC XBR
         DAC SV2
         LAC (BT+5
         DAC XSR
         JMS BTS
         DZM* XBT
         ISZ XBT
         ISZ TBT
         JMP .-3
```

TEST

```
JMS BTS
LAC (-12
DAC TL1
LAC (260
DAC TMP
LAC* SV2
SAD TMP
JMP .+5
ISZ TMP
ISZ TL1
JMP .-5
JMP .+6
TAD (-260
DAC* XSR
LAC XSR
TAD (-1
DAC XSR
LAC SV2
TAD (-1
SAD SV1
JMP .+3
DAC SV2
JMP .-25
JMS BTS
ISZ TBT
CLA
CLL
TAD* XBT
ISZ XBT
DAC .+3
LAC (12
MUL
HLT
SZL!CLL
JMP RET
LACQ
ISZ TBT
JMP .-13
TAD* XBT
DAC TMP
CLL
CLQ
LLS+12
SAD* LLC
JMP .+4
LAC (2
DAC WFL
JMP .+2
```

```
TEST

          DZM WFL
          LAC TMP
          CLL
          CLQ
          LLS+2
          XOR* LLC;            /STORE RESULTS OF TRIAL
          XOR WFL
          XOR XFL
          DAC* LLC
          LAC TEST+1Ø
          SAD (1
          JMP .+2
          JMP BK1+3
          LAC XFL
          SZA
          JMP XTR
          LAC WFL;  /HERE IF REG
          SZA
          JMP RWR
BK1       ISZ TLR;  /HERE IF REG AND RIGHT
          JMP .+2
          JMP TFX
          DZM XFL;  /BK1+3
          LAC TLC
          TAD (5
          DAC TLC
          ISZ TLT
          JMP .+2
          JMS TRS
          LAC TEST+1Ø;         /BK1+12
          SAD (2
          JMP .+32
          LAC (CR
          JMS PTB
          LAC (CR
          TAD .+2
          JMP .+2
          .DSA 14
          JMS PTB
          CLL
          CLQ
          LAC* LLC
          LRS+12
          JMS PI
          LAC (4
          JMS SPC
          LAC WFL
          CLL
```

TEST

```
        CLQ
        LRS+2
        CMQ
        LLS+1
        TAD (CR
        TAD .+2
        JMP .+2
        .DSA 57
        JMS PTB
        ISZ LLC
        ISZ TLL
        JMP .+2
        JMP L
        JMP 0
XTR     LAC WFL;  /HERE IF EXTRA
        SZA
        JMP .+13
        JMS WSV;  /HERE IF EXTRA AND RIGHT
        JMS WRS
        LAC* WLC
        SAD TMP
        JMP .+4
        ISZ WLC
        ISZ TLW
        JMP .-5
        DZM* WLC;           /FOUND IT
        JMS RSW
        LAC (-3;  /XTW
        DAC TLR
        JMS RST
        JMP BK1+3;          /BK2
RWR     JMS TSV;  /REG AND WRONG
        CLL
        CLQ
        LAC* LLC
        LRS+12
        DAC TMP
        JMS FNT
        SPA!SNA
        JMP .+6;  /JMP FUL
        JMS RST
        JMS WSV
        JMS WRS
        LAC* WLC
        SAD TMP
        JMP .+13
        ISZ WLC
        ISZ TLW
```

```
        JMP .-5
        JMS WRS
        LAC* WLC
        SNA
        JMP .+5
        ISZ WLC
        ISZ TLW
        JMP .-5
        JMP .+3
        LAC TMP
        DAC* WLC
        JMS RSW;  /FUL
        JMS RST
        JMP BK1
TFX     LAC* WLC;          /TIME FOR EXTRA
        SZA
        JMP .+7
        ISZ WLC
        ISZ TLW
        JMP .-5
        JMS WRS
        LAC* WLC
        SZA
        JMP .+6
        ISZ WLC
        ISZ TLW
        JMP .-5
        JMS WRS
        JMP BK1+3;         /NO EXTRA
        LAC (1
        DAC XFL
        JMS TSV
        LAC* WLC
        JMS FNT
        LAC TLC
        TAD (-4
        DAC TLC
        ISZ WLC
        ISZ TLW
        JMP .+2
        JMS WRS
        JMP BK1+12;        /BK3
L       JMS TRS;  /LIST TEST RESULTS
        LAC TLC
        TAD (-1
        DAC TLC
        LAC* TLC
        CMA;      TAD (1
```

TEST

```
        DAC TLL
        LAC (RES
        DAC LLC
        DZM LLN
        LAC (CR
        JMS PTB
        LAC (4
        JMS SPC
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 42;  /TRIAL
        JMS PTB
        LAC (4
        JMS SPC
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 45;  /STIMULUS
        JMS PTB
        LAC (4
        JMS SPC
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 52;  /RESPONSE
        JMS PTB
        LAC (4
        JMS SPC
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 6Ø;  /CORRECT
        JMS PTB
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 64;  //
        JMS PTB
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 57;  /INCORRECT
        JMS PTB
        LAC (4
        JMS SPC
        LAC (CR
        TAD .+2
```

```
   TEST

        JMP .+2
        .DSA 65;  /EXTRA
        JMS PTB
        DZM TMP
        LAC (CR-2
        JMS PTB
        JMS PL
        CLL
        CLQ
        LAC* LLC
        LRS+1
        AND (1
        TAD TMP
        DAC TMP
        ISZ LLC
        ISZ LLN
        ISZ TLL
        JMP .-15
        LAC (CR-2
        JMS PTB
        LAC (10
        JMS SPC
        LAC (10
        JMS SPC
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 70
        JMS PTB
        LAC (5
        JMS SPC
        LAC TMP
        CMA;      TAD (1
        TAD LLN
        JMS PI;   /NUMBER CORRECT
        LAC (CR
        TAD .+2
        JMP .+2
        .DSA 64
        JMS PTB
        LAC LLN
        JMS PI;   /TOTAL NUMBER
        LAC (CR-2
        JMS PTB
        JMP* TEST
BRS     0
        LAC (-24
        DAC TBR
```

TEST

```
        LAC (BR
        DAC XBR
        JMP* BRS
BTS     Ø
        LAC (-6
        DAC TBT
        LAC (BT
        DAC XBT
        JMP* BTS
TRS     Ø
        LAC* TEST+7
        DAC TLC
        LAC* TLC
        SPA!SNA
        JMP* TEST
        CMA;        TAD (1
        DAC TLT
        ISZ TLC
        ISZ TLC
        JMP* TRS
WRS     Ø
        LAC (WT
        DAC WLC
        LAC (-24
        DAC TLW
        JMP* WRS
TSV     Ø;          /SAVE TEST INDEX
        LAC TLC
        DAC STC
        LAC TLT
        DAC STT
        JMP* TSV
RST     Ø;          /RESTORE INDEX
        LAC STC
        DAC TLC
        LAC STT
        DAC TLT
        JMP* RST
WSV     Ø;          /SAVE WRONG INDEX
        LAC WLC
        DAC SWC
        LAC TLW
        DAC STW
        JMP* WSV
RSW     Ø;          /RESTORE INDEX
        LAC SWC
        DAC WLC
        LAC STW
```

TEST

```
            DAC TLW
            JMP* RSW
LPRT        Ø
            7ØØ4Ø1;     /TSF
            JMP .-1
            7ØØ4Ø6;     /TLS
            JMP* LPRT
PTB         Ø
            DAC SV1
            CLL
            LAC* SV1
            LRS+11
            SNA
            JMP* PTB
            JMS LPRT
            LLS+11
            AND (777
            SNA
            JMP* PTB
            JMS LPRT
            ISZ SV1
            JMP .-14
PI          Ø;          /PRINT INTEGER
            DAC SV1
            JMS BTS
            ISZ TBT
            CLL
            LAC (12
            DAC .+3
            LAC SV1
            IDIV
            HLT
            SZL!CLL
            JMP RET
            DAC* XBT
            LACQ
            DAC SV1
            LAC (12
            DAC .+3
            LAC SV1
            IDIV
            HLT
            SZL!CLL
            JMP RET
            ISZ XBT
            ISZ TBT
            JMP .-14
            DAC* XBT
```

TEST

```
        LAC BT+2
        TAD (26Ø
        JMS LPRT
        LAC BT+1
        TAD (26Ø
        JMS LPRT
        LAC BT
        TAD (26Ø
        JMS LPRT
        JMP* PI
SPC     Ø
        CMA;        TAD (1
        DAC SV1
        LAC (24Ø
        JMS LPRT
        ISZ SV1
        JMP .-3
        JMP* SPC
FNT     Ø;          /FIND TEST VECTOR
        DAC SV3
        JMS TRS
        LAC TLC
        TAD (4
        DAC TLC
        LAC* TLC
        SAD SV3
        JMP* FNT
        ISZ TLC
        ISZ TLT
        JMP .-1Ø
        CLA
        JMP* FNT
MT      Ø;          /MATCH
        DAC SV3
        LAC* XBR
        SAD SV3
        JMP* MT
        ISZ XBR
        ISZ TBR
        JMP .-5
        CLA
        JMP* MT
PL      Ø;          /PRINT RESULT LINE
        LAC (6
        JMS SPC
        LAC LLN
        TAD (1
        JMS PI
```

```
TEST

          .DSA 57;   /INCORRECT
          JMS PTB
          LAC (6
          JMS SPC
          LAC* LLC
          AND (1
          SZA
          JMP .+2
          JMP* PL
          LAC (CR
          TAD .+2
          JMP .+2
          .DSA 73
          JMS PTB
          JMP* PL
          215212
          215ØØØ
CR        212215
          276ØØØ
          314311;    /LISTEN TO STIMULUS
          323324
          3Ø5316
          24Ø324
          31724Ø
          32324;     /STIMULUS
          311315
          325314
          325323
          24ØØØØ
          323ØØØ;    /S
          331317;    /YOUR ANSWER IS?
          325322
          24Ø3Ø1
          316323
          3273Ø5
          32224Ø
          311323
          27724Ø
          24ØØØØ
          3243Ø5;    /TEST MODE
          323324
          24Ø315
          3173Ø4
          3Ø524Ø
          Ø
          32ØØØØ;    /P
          3Ø3ØØØ;    /C
          324ØØØ;    /T
```

TEST

```
          327310;    /WHAT?
          301324
          277000
          324322;    /TRIAL
          311301
          314000
          323324;    /STIMULUS
          311315
          325314
          325323
          0
          322305;    /RESPONSE
          323320
          317316
          323305
          0
          311316;    /IN
          303317;    /CORRECT
          322322
          305303
          324000
          257000;    //
          305330;    /EXTRA
          324322
          301000
          323303;    /SCORE
          317322
          305000
          253000;    /+
A=400000
TLC       0
TLT       0
STC       0
STT       0
LLC       0
TLL       0
LLN       0
WLC       0
TLW       0
SWC       0
STW       0
WFL       0
XFL       0
TLR       0
XBR       0
TBR       0
XBT       0
TBT       0
```

TEST

```
TMP     Ø
SV1     Ø
SV2     Ø
SV3     Ø
XSR     Ø
TL1     Ø
B       .BLOCK 7
BR      .BLOCK 24
BT      .BLOCK 6
        Ø
        Ø
        Ø
        Ø
        Ø
WT      .BLOCK 24
RES     .BLOCK 144
        .END
```

TEST

```
LAC (7
JMS SPC
CLL
CLQ
LAC* LLC
LRS+12
JMS PI
LAC (11
JMS SPC
CLL
CLQ
LAC* LLC
LRS+12
CLA
LLS+1Ø
JMS PI
LAC (7
JMS SPC
CLL
CLQ
LAC* LLC
LRS+1
AND (1
SZA
JMP .+16
LAC (CR
TAD .+2
JMP .+2
.DSA 6Ø;  /CORRECT
JMS PTB
LAC (CR
TAD .+2
JMP .+2
.DSA 64;  //
JMS PTB
LAC (17
JMS SPC
JMP .+17
LAC (7
JMS SPC
LAC (CR
TAD .+2
JMP .+2
.DSA 64;  //
JMS PTB
LAC (CR
TAD .+2
JMP .+2
```

APPENDIX G

THE PROGRAM "TALKM"

```
        /SUBROUTINE TO MODULATE
        /SYNTHETIC VOWEL AT
        /1ØKHZ RATE OF OUTPUT
        /CALLED BY TEST, TESTO
        .GLOBL    TALKM,.DA
TALKM   Ø
        JMS* .DA
        JMP .+1Ø
A       Ø;        /VECADD
        Ø;        /SOUADD
        Ø;        /ONADD
        Ø;        /OFADD
        Ø;        /STRTIM
        Ø;        /ONTIM
        Ø;        /OFTIM
        DZM ONAD1;         DZM OFAD1
        LAC* A
        DAC .+4
        CLL
        LAC (2342Ø
        IDIV
        HLT
        LACQ
        DAC STRPTS
START   LAC A+1
        DAC STRAD1;        DAC STRAD2
        LAC A+2
        SPA!SNA
        JMP .+2
        DAC ONAD1
        LAC A+3
        SPA!SNA
        JMP +2
        DAC OFAD1
        LAC* A+4;          DAC STRTM
        LAC* A+5;          DAC ONTM
        LAC* A+6;          DAC OFTM
        LAC ONAD1
        SPA!SNA
        JMP .+3
        LAC* ONAD1;        DAC ONPTS
        LAC OFAD1
```

TALKM

```
        SPA!SNA
        JMP .+3
        LAC* OFAD1
        DAC OFPTS
        LAC ONAD1
        TAD (1
        DAC INDEX1
        LAC ONTM
        SPA!SNA
        JMP A3
        LAC ONPTS
        SPA!SNA
        JMP A3
        TAD (-1
        CMA;      TAD (1
        DAC TALY
        LAC (FCTND
        DAC INDEX2
LOOP1   LAC* INDEX1;        /GET Y2-Y1 FOR
        CMA!CLL;  /ON MOD
        ISZ INDEX1
        ADD* INDEX1
        DAC* INDEX2
        ISZ INDEX2
        ISZ TALY
        JMP LOOP1
        LAC OFAD1
        TAD (1
        DAC INDEX1
A3      LAC OFTM
        SPA!SNA
        JMP A4
        LAC OFPTS
        SPA!SNA
        JMP A4
        TAD (-1
        CMA;      TAD (1
        DAC TALY
        LAC (FCTFD
        DAC INDEX2
LOOP2   LAC* INDEX1;        /GET Y2-Y1 FOR
        CMA!CLL;  /OFF MOD
        ISZ INDEX1
        ADD* INDEX1
        DAC* INDEX2
        ISZ INDEX2
        ISZ TALY
        JMP LOOP2
```

TALKM

```
A4      LAC ONTM;              /GET ONINC
        SPA!SNA
        JMP A1
        LAC ONPTS
        SPA!SNA
        JMP A1
        TAD (-1
        DAC .+4
        CLL
        LAC ONTM
        IDIV
        HLT
        LACQ
        DAC INCN
A1      LAC OFTM;              /GET OFINC
        SPA!SNA
        JMP A2
        LAC OFPTS
        SPA!SNA
        JMP A2
        TAD (-1
        DAC .+4
        CLL
        LAC OFTM
        IDIV
        HLT
        LACQ
        DAC INCF
A2      LAC STRPTS;            /GENERAL SETUP
        TAD STRAD1;            /TIMING FOR ALL
        DAC TOP;  /LOOPS IS FOR 1ØKHZ
        LAC STRTM;             /OUTPUT RATE WITH EAE
        TAD (1
        DAC INCS
        DZM TIMS
        DZM TIMN
        LAC ONAD1
        TAD (1
        DAC FCTN
        LAC (FCTND
        DAC FCTNDF
        LAC ONPTS
        TAD (-1
        CMA;      TAD (1
        DAC TALYN
        DZM TIMF
        LAC OFAD1
        TAD (1
```

TALKM

```
          DAC FCTF
          LAC (FCTFD
          DAC FCTFDF
          LAC OFPTS
          TAD (-1
          CMA;      TAD (1
          DAC TALYF
          LAC ONTM
          SPA!SNA
          JMP PTRSO
LOOPN     CLL;      /ONMOD LOOP
          LAC INCN
          DAC .+3
          LAC TIMN
          FRDIV
          HLT
          LACQ
          RCR
          CLL
          DAC .+3
          LAC* FCTNDF
          MULS
          HLT
          LLSS+1
          CLL
          CLL
BKIN      ADD* FCTN
          GSM
          DAC .+3
          LAC* STRAD1
          MULS
          HLT
          LLSS+1
          SNL!CLL
          JMP LG2 N
          TAD (1
BK2N      LDY;      /OUTPUT POINT
          ISZ TIMN
          LAC TIMN
          SAD INCN
          JMP TMRSN
          ISZ STRAD1
          LAC STRAD1
          SAD TOP
          JMP PTRSN
          NOP
          NOP
          NOP
```

TALKM

```
          NOP
          NOP
          NOP
          NOP
          JMP LOOPN
TMRSN     ISZ STRAD1;          /RESET MODUL
          LAC STRAD1
          SAD TOP
          JMP PTTRSN
          ISZ TALYN
          JMP .+2
          JMP TRSO
          DZM TIMN
          ISZ FCTN
          ISZ FCTNDF
          LAC (-14
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
          NOP
          NOP
          CLA!CLL
          JMP BKIN
PTRSN     LAC STRAD2;          /RESET CYCLE
          DAC STRAD1
          NOP
          NOP
          JMP LOOPN
PTTRSN    LAC STRAD2;          /RESET CYCLE AND MODUL
          DAC STRAD1
          ISZ TALYN
          JMP .+2
          JMP PTRSO
          DZM TIMN
          ISZ FCTN
          ISZ FCTNDF
          LAC (-13
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
          CLA!CLL
          JMP BKIN
TRSO      LAC STRTM
          SPA!SNA
          JMP TRSOF
          LAC (-25
```

TALKM

```
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
          NOP
          NOP
          JMP LOOPS
PTRSO     LAC STRTM
          SPA!SNA
          JMP PTRSOF
          LAC (-24
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
          JMP LOOPS
LOOPS     NOP;       /STRAIGHT LOOP
          CLL
          LAC* STRAD1
          SMA
          JMP LGIS
          TAD (1
BK1S      LDY;       /OUTPUT POINT
          ISZ STRAD1
          LAC STRAD1
          SAD TOP
          JMP PTRSS
          ISZ TIMS
          LAC TIMS
          SAD INCS
          JMP STRO
          LAC (-3Ø
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
          JMP LOOPS
PTRSS     LAC STRAD2;          /RESET CYCLE
          DAC STRAD1
          ISZ TIMS
          LAC TIMS
          SAD INCS
          JMP STRPO
          LAC (-26
          DAC WAIT
          ISZ WAIT
          JMP .-1
          NOP
```

TALKM

```
        NOP
        JMP LOOPS
STRO    NOP
        NOP
        NOP
        NOP
        NOP
STRPO   NOP
        NOP
TRSOF   NOP
        NOP
        NOP
        NOP
        NOP
PTRSOF  LAC OFTM
        SPA!SNA
        JMP* TALKM
        LAC (-12
        DAC WAIT
        ISZ WAIT
        JMP .-1
        NOP
        NOP
        NOP
        CLA!CLL
        JMP BK1F
LOOPF   CLL;        /OFMOD LOOP
        LAC INCF
        DAC .+3
        LAC TIMF
        FRDIV
        HLT
        LACQ
        RCR
        CLL
        DAC .+3
        LAC* FCTFDF
        MULS
        HLT
        LLSS+1
        CLL
        CLL
BK1F    ADD* FCTF
        GSM
        DAC .+3
        LAC* STRAD1
        MULS
        HLT
```

TALKM

```
        LLSS+1
        SNL!CLL
        JMP LG2F
        TAD (1
BK2F    LDY;        /OUTPUT POINT
        ISZ TIMF
        LAC TIMF
        SAD INCF
        JMP TMRSF
        ISZ STRAD1
        LAC STRAD1
        SAD TOP
        JMP PTRSF
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        JMP LOOPF
TMRSF   ISZ STRAD1;          /RESET MODUL
        LAC STRAD1
        SAD TOP
        JMP PTTRSF
        ISZ TALYF
        JMP .+2
        JMP* TALKM
        DZM TIMF
        ISZ FCTF
        ISZ FCTFDF
        LAC (-14
        DAC WAIT
        ISZ WAIT
        JMP .-1
        NOP
        NOP
        NOP
        CLA!CLL
        JMP BK1F
PTRSF   LAC STRAD2;          /RESET CYCLE
        DAC STRAD1
        NOP
        NOP
        JMP LOOPF
PTTRSF  LAC STRAD2;          /RESET CYCLE AND MODUL
        DAC STRAD1
        ISZ TALYF
```

TALKM

```
            JMP .+2
            JMP* TALKM
            DZM TIMF
            ISZ FCTF
            ISZ FCTFDF
            LAC (-13
            DAC WAIT
            ISZ WAIT
            JMP .-1
            NOP
            CLA!CLL
            JMP BK1F
LG1N        JMP BK1N
LG2N        JMP BK2N
LG1S        JMP BK1S
LG1F        JMP BK1F
LG2F        JMP BK2F
LDY=7ØØ6Ø4
STRPTS      Ø
FRØ         Ø
STRAD1      Ø
STRAD2      Ø
ONAD1       Ø
OFAD1       Ø
STRTM       Ø
ONTM        Ø
OFTM        Ø
ONPTS       Ø
OFPTS       Ø
INDEX1      Ø
INDEX2      Ø
TALY        Ø
INCN        Ø
INCS        Ø
INCF        Ø
TOP         Ø
TIMN        Ø
TIMS        Ø
TIMF        Ø
TALYN       Ø
TALYF       Ø
WAIT        Ø
FCTN        Ø
FCTF        Ø
FCTNDF      Ø
FCTFDF      Ø
FCTND       .BLOCK 146
FCTFD       .BLOCK 146
```

```
TALKM

        .END
```

LIST OF REFERENCES

August, Roland D. "Feasibility Study of a Synthetic Speech Signal Test Set", Final Report Signal Corps Contract No. DA-02-086-AMC-055(E), 1965.

Fletcher, Harvey D. Speech and Hearing in Communication, D. Van Nostrand, New York, 1953, p. 288.

Manley, Harold J. "Analysis-Synthesis of Connected Speech in Terms of Orthogonalized Exponentially Damped Sinusoids", Journal of Acoustical Society of America, Vol. 35, No. 4 April 1963, p. 473.

McClellan, Richard P. "An Upper Bound for Tactile Recognition of Speech", Master's Thesis, University of Arizona, 1967.

Risberg, A., and Spens, K. "Teaching Machine for Trained Experiments in Speech Perception", Speech Transmission Laboratory, Quarterly Progress and Status Report, Royal Institute of Technology, Stockholm, Sweden, October, 1967, pp. 72-75.

Sidley, Norman A. "Stimulus-Identification Overlap in Learning to Identify Complex Sounds", Journal of Acoustical Society of America, No. 38, 1965, pp. 11-13.