To the Graduate Council:

I am submitting herewith a thesis written by Keith Marlin Anspach entitled "An Application of Voice Input to a Microcomputer for Environmental Control." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Ronald A. Belz

Ronald A. Belz, Major Professor

We have read this thesis and recommend its acceptance:

R L Young

K.R. Kimble

Accepted for the Council:

CW Minkel

The Graduate School

AN APPLICATION OF VOICE INPUT TO A MICROCOMPUTER FOR ENVIRONMENTAL CONTROL

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Keith Marlin Anspach

August 1984

ACKNOWLEDGMENTS

I thank Dr. Ron Belz for the many hours he spent proofreading this manuscript and making helpful suggestions, for the assistance he gave in the development of a hardware device built for this study and for being the chairman of my committee. I also thank committee members, Dr. Kenneth R. Kimble and Dr. Robert L. Young, for their time spent reviewing and critiquing this manuscript. I thank the R. L. Young family for their financial support of the B. H. Goethert scholarship, and particularly for their treasured friendship. I wish to thank the UTSI library staff, Marge Joseph, Mary Lo and Helen Mason for their help in finding and acquiring literature. I thank my parents, Roberta and Gene, for their assistance in drafting the figures contained in this manuscript and for "putting up with me." I also thank Roberta for the excellent job of typing the manuscript.

ABSTRACT

An application of voice input to a microcomputer for environmental control is described. This application is a useful aid to the handicapped for routine home operations, such as, turning on or off lights and dialing or answering the telephone. Software for the microcomputer was written to provide environmental control in a logical, easy-to-use fashion. Hardware was designed to interface the microcomputer with environmental control devices which provide control of, among other things, a speaker telephone, a lamp and two televisions. Some commercially available hardware was implemented to provide speech recognition and control of lights and appliances through the existing house wiring.

PREFACE

In the beginning, this study was initiated to provide an aid to the handicapped. The author himself, after suffering a severed spinal cord resulting in permanent paralysis of all four extremities, is a member of the handicapped population and has had to deal with some of the frustrations associated with tasks which this study provides aid. It is hoped, by the author, that not only the handicapped will find this study to be an aid, but also the "able-bodied" population will find aids in it which can be used in their applications.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	ENVIRONMENTAL CONTROL	3
	A. Introduction	3
	B. Environmental Control Unit	4
	C. Existing Wiring Home Control Systems	6
III.	SPEECH RECOGNITION	9
IV.	METHODS AND PROCEDURES	13
	A. Introduction	13
	B. Hardware Implementation	15
	Microcomputer System	15
	Voice Input Module	15
	Annunciator Interfacing Device	16
	C. Software Development	21
	Introduction	21
	Home Control	25
	Telephone Control	32
	Environmental Control Unit Control	48
	D. Utility Subroutines	57
	Vocabulary Switch	57
	Voicein Subroutine	58
V.	RESULTS AND RECOMMENDATIONS FOR FURTHER STUDY	63
	A. Results	63
	B. Recommendations for Further Study	68

CHAPTER PAGE

BIBLIOGRAPHY 70

VITA 72

LIST OF FIGURES

FIGURE PAGE

1. Microcomputer Based, Voice Input Controlled Environmental Control System 14
2. Schematic of Annunciator Interfacing Device 18
3. Flowchart of System Initialization 22
4. Flowchart of Home Control Subroutine 27
5. Flowchart of Telephone Control Subroutine 33
6. Flowchart of ECU Control Subroutine 49
7. Flowchart of Vocabulary Switch Subroutine 59
8. Flowchart of VOICEIN Subroutine 60

CHAPTER I

INTRODUCTION

Tasks which are deemed routine to most people are major undertakings for the severely physically handicapped. For example, routine tasks such as turning on or off a television, stereo or lamp, changing television channels, answering the telephone and dialing a number are denied to the disabled. A way for the handicapped to perform some of these routine tasks through the application of a microcomputer, voice input technology and home control devices will be described.

The severely disabled can be aided through voice input to a microcomputer. The voice input module provides, through an analog to digital converter and bandpass filters, the capability of digitizing voice input. A microcomputer programmed with a speech recognition algorithm responds to an input word or phrase spoken by the handicapped individual. The response of the microcomputer to this input is an output such as a switch closure that can be used for home environmental control. Home environmental control can be achieved through several methods--two are discussed in this study. One method interfaces the microcomputer through Transistor-Transistor Logic (TTL) and reed relays, which is designated as the Annunciator Interfacing Device (AID) and is used to control an environmental control unit that provides switching of a lamp, a radio, a tape player, an intercom and two televisions. AID is also used to answer, hang up and dial a speaker telephone. The second method takes advantage of the voice input module's capability to transmit coded signals through existing house wiring for remote control of switches which control lights and appliances.

The software needed to bring the voice input and environmental control technologies together is the major thrust of this study. It was written to provide home environmental control in a logical and easy-to-use fashion. The programs are menu driven so that voice input phrases can take on multiple meanings, depending on which menu is active at the time voice input is made. The menu technique allows the limited vocabulary of recognizable voice input phrases to be used more extensively.

Environmental control of a house for a handicapped person is presented in Chapter II. In Chapter III general concepts for speech recognition are discussed. The hardware chosen, the hardware designed and the software written for this study are presented in Chapter IV. In Chapter V the results of the development of the microcomputer based home environmental control concept based on voice input are discussed, and recommendations are made for further improvement of the concept.

CHAPTER II

ENVIRONMENTAL CONTROL

A. Introduction

Home environmental control, such as turning on or off the lights, or a television, and answering, or initiating, a telephone call, is considered routine to most people. Control of these devices in the traditional ways is not feasible for the severely disabled. For example, a handicapped individual may not be able to walk across a room and do a simple task such as turn on a lamp. Some means is therefore required to provide this capability through the facilities the handicapped person has. Some kind of centralized control device is needed to control the entire home environment. The device used for control needs to be remote from the individual but easily accessible. This feature is especially urgent should an emergency arise, and a call for help must be made.

Two commercially available home environmental control methods were used in this study. One unit provides control of latching and momentary action switches through dedicated wiring. The second provides remote control, via existing 120 volt electrical power wiring, of wall switch modules and modules which plug into the wall sockets.

B. Environmental Control Unit

The Environmental Control Unit (ECU) (Environmental Control Unit, Prentke Romich Co., Shreve, Ohio) provides the control of six latching switches and six momentary action switches. Four of the latching switches output 120 VAC. These switches are all contained in a single chassis, which limits the ECU to a single room, such as a bedroom or a living room.

The ECU is controlled by two sets of switches. One set of switch inputs is pneumatically controlled. This pneumatic activated switch is composed of a straw connected to a cylinder with a rubber lid. By puffing into the straw the pressure inside the cylinder increases expanding the lid outwardly which closes a microswitch. Conversely, by sipping on the straw the pressure inside the cylinder decreases expanding the lid inwardly activating another microswitch.

The other set of switches is controlled remotely using an infrared transmitter attached to a wheelchair which communicates with an infrared receiver attached to the ECU. The infrared transmitter produces two infrared signal outputs. The infrared receiver detects the transmitted infrared signals and relays them to the ECU.

These signals are used to control the ECU's scanner; one positions the scanner and the other activates the desired device. The infrared signals are transmitted from the wheelchair by positioning a joystick. One direction of motion activates one of the signals while motion in another direction position activates the other.

The scanner of the ECU indicates, by means of two columns of seven light-emitting diodes (LEDs), which device is being activated or deactivated. Each scanner position, designated by a specific letter or word, controls a different device. Table 1 lists the function associated with each scanner position. Since each LED is associated with a single device only one LED is lit at any specific time. Two switches control the scanner operation. One switch closure starts the scanner, and when it is released the scanner is stopped on the desired device. The other switch closure is used to turn the device "on" or "off."

Table 1. Scanner Positions and Associated Functions.

Position	Function	Position	Function
SPACE	Dialing	T	Telephone
A	120 VAC switch	N	120 VAC switch
E	120 VAC switch	D	120 VAC switch
I	Latching switch	M	Latching switch
O	Momentary action	Y	Momentary action
U	Momentary action	B	Momentary action
SHIFT	Momentary action	X	Momentary action

The latching switches can be used to control devices such as lamps, radios and televisions. The momentary action switches can be used to change channels on a hospital-type television, raise or lower a hospital bed or control a cassette tape player. The range of applications, for the latching and momentary action switches, is very broad.

This system is applicable to controlling the environment of a single room. The two-switch approach to control allows the severely handicapped to control the ECU by a pneumatic switch (i.e., sip for one switch action and puff for the other), or micro switches could be set up to detect slight body movements such as an eyebrow motion or the blinking of an eye.

C. Existing Wiring Home Control System

When using a microcomputer for home control the job of running wires to each and every device that is to be controlled becomes a tedious task. An Existing Wiring System (EWS) (BSR System X-10, BSR (USA) LTD., Blauvelt, NY), however, uses the AC wiring in the house to carry transmitted coded messages. The codes are received by modules which act as remotely activated switches and can be used to control lights and appliances.

The EWS consists of five components (1) a command console, (2) a wall switch module, (3) an appliance module, (4) a lamp module and (5) a cordless controller. The command console produces coded signals which are transmitted via the existing house wiring to the individual modules. The appliance and lamp modules are used as remotely controlled switches. The specific appliance or lamp that is to be controlled is turned on, plugged into the appropriate module and the module is plugged into the wall socket. The wall switch module on the other hand replaces an existing wall switch. The cordless controller is used to remotely control the command console.

The command console can produce 256 unique codes. The codes are set by two switches, one is the "house code" and the second is the "unit code." There are 16 house codes, designated by the letters A through P, and 16 unit codes, designated by the numbers 1 through 16. There is actually a 16 coded message limit since the command console can generate only one house code (at the user's choice). However, there is a commercially available interface which plugs directly into a microcomputer and the AC powerline and allows generation of all 256 codes.

The appliance, lamp and wall switch modules can be uniquely addressed by setting the house code and unit code on each module. The house code must be the same as the house code setting on the command console. For example if the house code is A for the command console, then the unit codes for the modules can be A1, A2, A3, ..., A16.

The voice input module can simulate the cordless controller of the EWS by producing the appropriate signals for remote control, via a transducer, which becomes the interface between the microcomputer and the EWS. This interface method isolates the microcomputer from the command console's 120 VAC power source.

A disadvantage to the existing wiring system is the cost of the separate modules, which is approximately 20 dollars each. However, considering the inconvenience of running individual wires throughout a house to each device and using relays which isolate the microcomputer from the AC power source, this disadvantage is trivial in most cases.

CHAPTER III

SPEECH RECOGNITION

In the past the disabled have had to turn to various methods for input to a computer, such as a mouthstick, a light pen or a specially designed keyboard when use of their hands was limited. Voice input is helpful because it provides a certain amount of independence which is of great importance to the individual. Although the mouthstick and light pen also provide independence, if the mouthstick or light pen happens to be accidentally dropped, then the individual is again dependent upon someone to pick it up. The keyboard provides independence as well, but for someone who has very limited strength the keyboard approach can be very tiresome. A solution to these potential problems is voice input. The microphone used for input can be strategically placed so it will not be dropped.

Voice input control is based on a device which decodes speech patterns and applies a speech recognition algorithm to identify and discriminate between spoken words or phrases. Speech recognition is basically straightforward: the computer digitizes the voice input waveform and compares it to voice signatures previously stored [1][1]. Voice signatures can be achieved by many methods including bandpass filtering and zero-crossing detection [2].

---

[1] Numbers in brackets refer to similarly numbered references listed in the Bibliography.

Speech recognition is still in its early years, but one way that it can be used effectively is in a limited vocabulary, isolated word speech recognition system. A vocabulary is defined as a set of voice patterns previously stored to which the input voice patterns are compared in hopes of finding a match. A match occurs when the incoming voice pattern is determined to be similar to one of the voice patterns in the vocabulary. In isolated word recognition the words or phrases are separated by definite pauses. A limited vocabulary is used so that the recognition reliability is increased [3]. Limiting the vocabulary also limits the amount of memory used. This becomes important when using a microcomputer.

One method of determining a match is to consider the entire incoming voice signature and compare it to the vocabulary. Some kind of matching algorithm which determines the similarity of the input signature to each of the vocabulary signatures can be used for the comparison of the voice signatures. This method requires that a vocabulary of voice signatures be created before recognition can begin.

Another method for finding a match is the segmental approach which involves breaking an utterance down into subunits like phonemes (i.e., a basic subunit sound of speech) and syllables. The segmental approach compares the utterance to known subunits. A match is calculated by stringing the subunits together in the order of occurrence and searching the vocabulary for that combination of subunits. A decision on what size subunits to use must be made, and there are several good arguments about size [3, 4]. There are also certain acoustic features which are useful for speech recognition, e.g., vowels and consonants [5].

It is common practice in isolated word recognition to consider each voice signature separately and select the one with the best match. In most algorithms a match is determined by the voice pattern which is the least dissimilar. A score is associated with the dissimilarity where the lower the score the lower the amount of dissimilarity. If the best score is acceptable, then the input is recognized as the appropriate signature [6].

Recognition of connected speech is much more complex than isolated word recognition. People tend to run their words together when speaking in a normal fashion, thus making the boundaries of the words difficult to determine. Contextual information about connected words helps in the recognition process. Details on how to teach the computer to recognize different contexts have not yet been sufficiently developed [7]. Dynamic programming techniques to some extent can be used efficiently in connected speech algorithms [8].

In some connected speech systems the segmental approach is used, that is, the voice input is preprocessed and divided into phonemes, syllables or words [9]. However, even though much work has been done using these segmental approaches, more research is needed before these systems are viable [8].

There are two basic types of speech recognition; speaker-dependent and speaker-independent. In a speaker-dependent system the operator must train the system to recognize their voice. This requires each operator to speak certain words or phrases which are digitized and stored in a vocabulary. One advantage to speaker-dependent systems is that they can be used with any language. A disadvantage is that only one person can control the system, and if their voice intonation changes, the speech pattern may not be recognized.

In a speaker-independent system the voice inputs are compared to voice signatures which are not created by the speaker. The voice signatures are created by ". . . . a set of statistically sampled pronouncements. . . ." of the word or phrases [7]. This makes the system independent of the speaker. There are problems associated with speaker-indepdent systems because of the wide variations of speech such as speed, accents, and inflections. An advantage of the speaker-independent system is its ability to recognize multiple speakers.

The first complete speech recognizer was developed in 1952 by Davis, Biddulph and Balashek of Bell Telephone Laboratories. Their system divided the frequency spectrum into signal levels which were above 900 Hz and below 900Hz and counted the zero-crossings (i.e., the number of times the speech signal crosses zero in a specific interval). The input signature was compared to ten previously stored signatures for the digits zero through nine, and the most similar match was determined by cross-correlation. The previously stored signature with the closest match was taken to be the recognized digit [3].

CHAPTER IV

METHODS AND PROCEDURES

A. Introduction

The system requirements for environmental control are split between two categories, hardware and software. Hardware is needed which can take an analog voice input, convert it to a digital representation, and perform predefined tasks associated with the voice input. Environmental control devices, which can be interfaced to this hardware are used to control the lights, the appliances and the telephone. Software is needed to program the hardware so that specific home control tasks, through the environmental control devices, can be performed in a logical, easy-to-use fashion.

As shown in the block diagram in Figure 1, a microcomputer is the central device by which all the other devices are controlled. The universal symbol of the handicapped shows an individual speaking into a microphone which is connected to the Voice Input Module (VIM). The VIM is interfaced to the microcomputer through one of its I/O expansion slots. Voice input is recognized by software in the microcomputer, and the appropriate actions are taken. These actions may cause another program to be executed or cause any of the devices connected to the computer controlled environmental control systems, to be activated (e.g., turning on or off a lamp, dialing the telephone or changing channels on the television). The ECU and the speaker

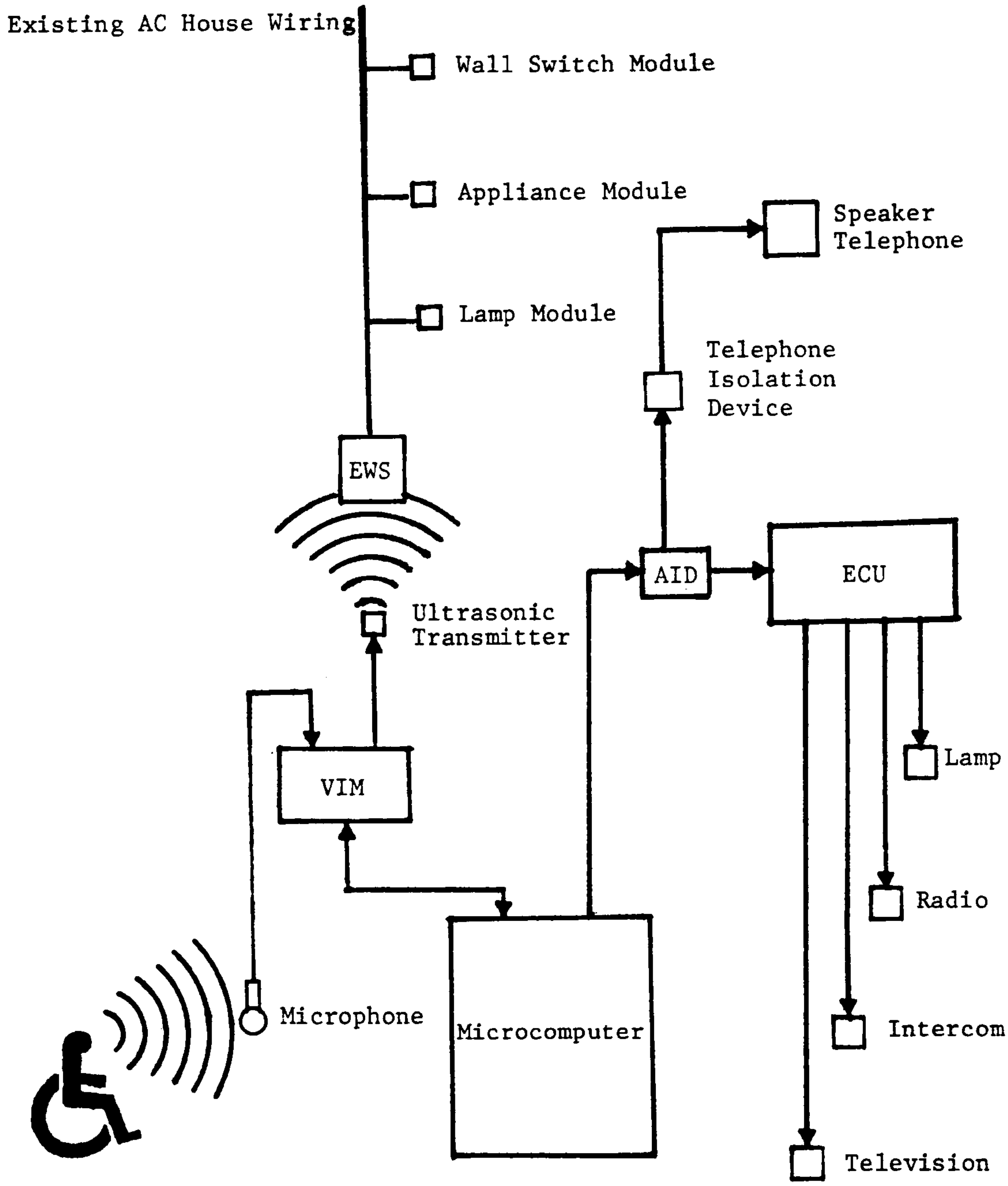

Figure 1. Microcomputer Based, Voice Input Controlled Environmental Control System.

telephone are interfaced to the microcomputer through the Annunciator Interfacing Device (AID), which was designed specifically for this study, while the EWS is interfaced through the VIM.

B. Hardware Implementation

Microcomputer System

A microcomputer (Apple IIe, Apple Computer, Inc., Cupertino, CA) was used to recognize the voice input and perform predefined tasks. It was easily adapted for voice input by simply plugging an interface card into one of the I/O expansion slots. An interface to an EWS can be achieved using an interface card. In addition an ECU can be controlled through existing annunciator outputs provided in the microcomputer to drive reed relays which parallel the ECU's inputs.

Voice Input Module

The Voice Input Module (VIM) (Waldo, ARTRA, Inc., Arlington, VA) used in this study was designed for the chosen microcomputer system, and it solves two of the hardware requirements, voice input and an interface to the existing wiring system. It also has speech recognition software, a real-time clock which could possibly be used for turning on devices at specified times and sound generation capabilities which could be used for audible feedback.

This VIM operates by digitizing the analog voice input which is passed by two bandpass filters. One filter passes sounds which are above 1000 Hz, and the other filter passes sounds below 1000 Hz. The speech recognition software allows the user to store a vocabulary of 24 voice signatures. The program samples the microcphone for voice input; when a sound is detected it then collects data. The data collected is the sum of positive amplitude peaks, from the two bandpass filters, during a twelve-millisecond period. There are 160 twelve-millisecond periods, thus making the total input period about two seconds long. The digitzied voice input data is then compressed and stored into a 64-byte buffer [10].

When the program is in the recognition mode the voice input data is collected and processed by the same procedure as described above, but stored in a temporary buffer. The voice signature is then compared to the vocabulary using a "linear fit" method. The signature in the vocabulary which is most similar to the voice input signature is determined. A score is associated with the similarity, and if the score is below a predetermined value (the lower the score the better the fit), then the voice input is said to be recognized. A recognition reliability of 85% to 95% for multiple word phrases is claimed [10]. However, the setting of the predetermined value for acceptable scores greatly influences the recognition reliability. For example when the value is too small then recognition reliability is very low, and when the value is too large, the chance of accepting a phrase which is erroneous is increased.

Annunciator Interfacing Device

Interface of the microcomputer to an ECU can be achieved through standard annunciator outputs provided with the microcomputer. Four annunciator outputs are available on a 16-pin Dual In-line Package (DIP) socket provided for GAME I/O [11].

The ECU was controlled by paralleling its two switch inputs with two relay contact closures activated by the outputs from the annunciators. One of the switches controls the scanner which sequentially steps through each of the ECU's control relays. The second switch controls the ON/OFF status of the addressed control relay.

Each of the four annunciator outputs was connected to dual input NAND relay drivers, as shown in Figure 2. They were, in turn, connected to reed relays, thereby providing control for four switching circuits. The relay driver circuit is a two-input AND gate driving an open collector transistor. The two inputs of the AND gate are tied together so that when the input is brought to a high state (+5v) the open collector transistor is turned on. When the input is low (Gnd) the open collector transistor is turned off. The collector of the transistor is connected to the coil of the reed relay circuit. Therefore, when the transistor is on, current flows through the reed relay coil closing the relay. When the transistor is off, the reed relay switch is open. Each reed relay coil has a diode and a capacitor in parallel for protection from the collapsing magnetic field which occurs when the transistor turns off.

Two of the annunciator outputs are initially set high, and two are initially set low by the micrccomputer. Therefore, inverters were inserted at the high annunciator outputs so that the two relays are not closed when the computer is turned on. One of the

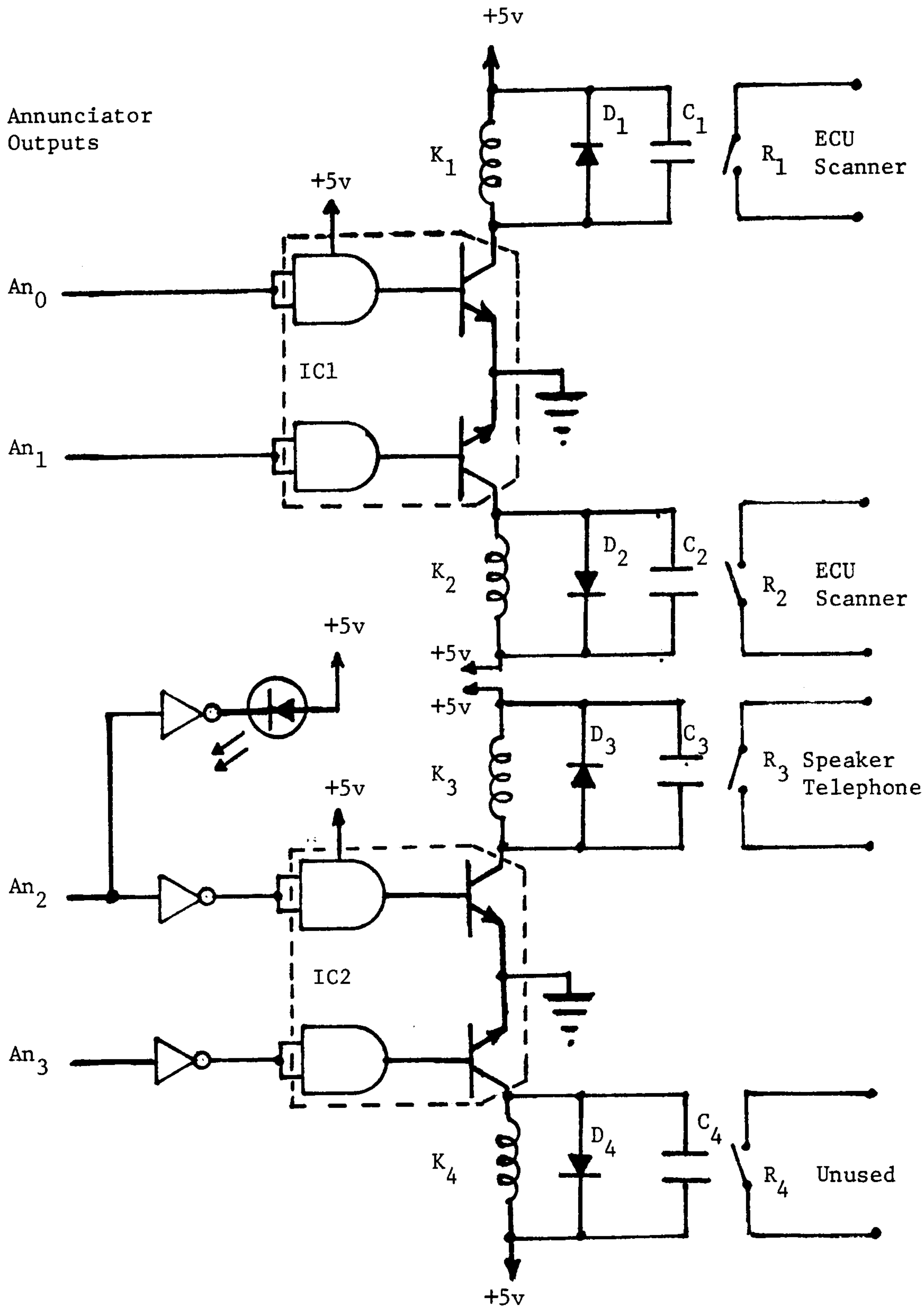

Figure 2. Schematic of Annunciator Interfacing Device.

annunciator outputs is connected to a LED which gives a visual indication of the relay's state (i.e., on for open and off for closed).

The state of each annunciator is controlled by addressing a pair of memory locations shown in Table 2. The operation performed when addressing these memory locations is irrelevant because only a reference to that address is required. When the first location of the pair is addressed the annunciator goes low. When the second location is addressed the annunciator goes high.

Table 2. Addresses for Annunciator Control

ANNUNCIATOR		ADDRESS	
NUMBER	STATE	DECIMAL	HEX
0	OFF	49240	$C058
	ON	49241	$C059
1	OFF	49242	$C05A
	ON	49243	$C05B
2	OFF	49244	$C05C
	ON	49245	$C05D
3	OFF	49246	$C05E
	ON	49247	$C05F

The two reed relays of annunciator Numbers 0 and 1 are in parallel with the switch inputs to the ECU providing an interface between the microcomputer and the ECU. The reed relay which is connected to the output of annunciator Number 2 (with the LED visual indicator) is used to control a speaker telephone. When the LED is on or off, then the phone is on or off the hook, respectively. The LED blinks during the pulse dialing process indicating a phone number is being dialed. The pulse dialing process is discussed in more detail later in this chapter.

C. Software Development

Introduction

The software in this study is divided into two types. One type is the software which is provided with the VIM, and the other type is the software written for this application. As shown by the flowchart in Figure 3, the VIM software is used to initialize the system. The user must enter some keyboard input to place the VIM software into a voice input mode, which requires stepping through menus. When the VIM software is in the voice input mode, there are three voice inputs which cause the execution of the software written for this study.

The voice input are phrases which are associated with the three kinds of home control provided (1) "HOME CONTROL", (2) "TELEPHONE" control and (3) "ENVIRONMENTAL CONTROL UNIT" control, and they are used to invoke the appropriate subroutines (these subroutines are discussed in more detail in subsequent sections). Any voice input recognized which is not one of these three phrases is ignored until after these subroutines are executed.

The subroutines written utilize portions of the VIM software. These portions include the speech recognition algorithm, clock display subroutine and time schedule subroutine. A utility subroutine, VOICEIN (which is discussed later), was written so that the three control programs could take advantage of the VIM software. In this subroutine the VIM's speech recognition algorithm was used to collect and analyze the voice input data. The utility program was written so that the number of the recognized phrase was in the

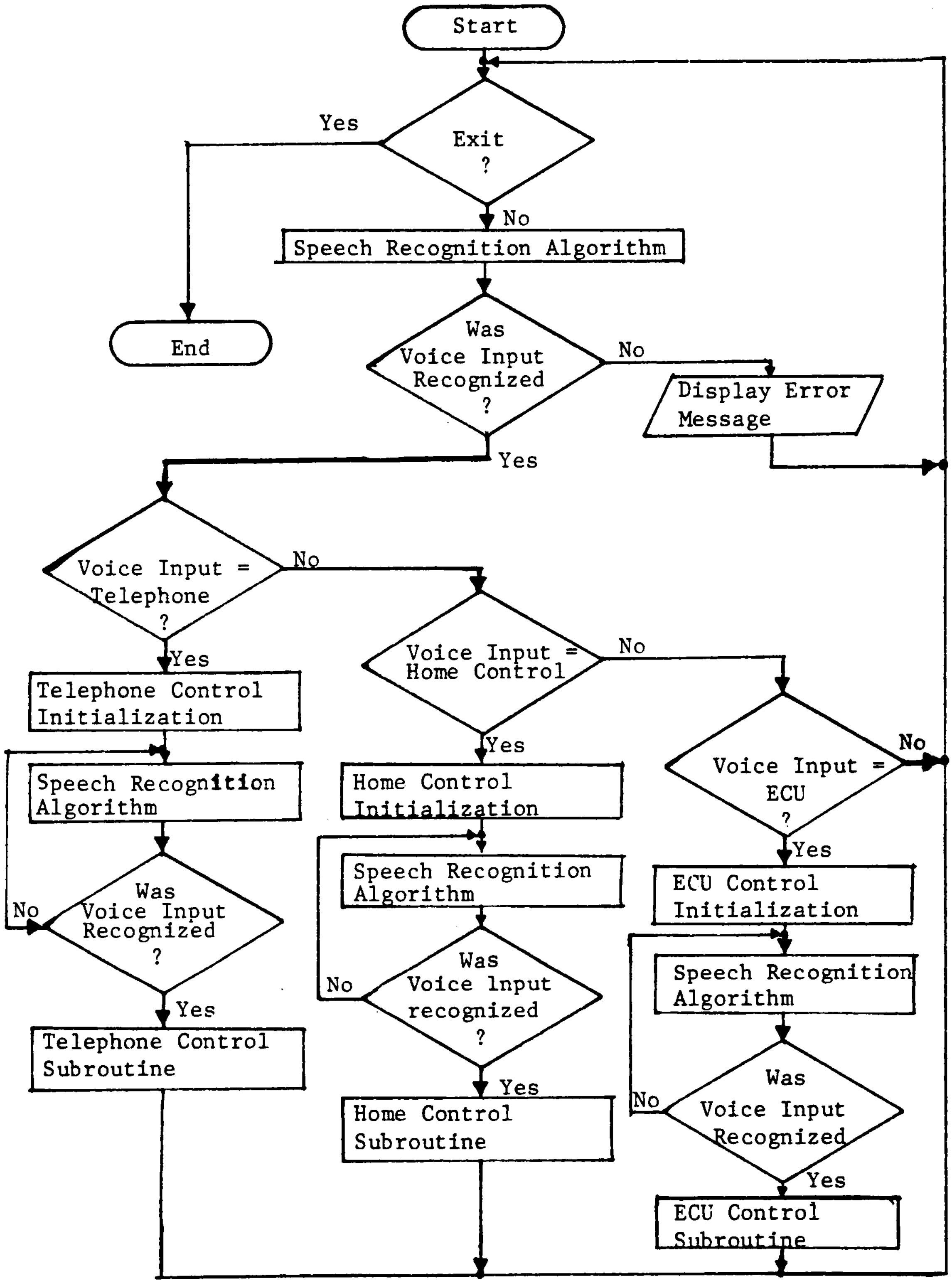

Figure 3. Flowchart of System Initialization.

accumulator when the subroutine transferred control to the program which called it. The utility program returns with 99 (hexidecimal) in the accumulator if the voice input is invalid. Examples of invalid voice input would be if the phrase was too long or if the phrase was not recognized as a phrase in the vocabulary.

The Home Control program was written to control the overhead lights, floor lamps and appliances through an EWS. The Telephone Control program was written to control a speaker telephone and allowed for the dialing of 48 previously stored phone numbers or a phone number entered by voice. The Telephone Control program also provided the ability to answer and hang up the speaker telephone by voice input. The ECU control program was written to control an environmental control unit and provided for the control of an intercom, a lamp, a tape player, a radio and the changing of channels on two television sets.

The VIM speech recognition algorithm allows the VIM to be used with a limited vocabulary for isolated word recognition. The vocabulary is limited to 24 words, and each word must be separated by definite pauses. Since the software provided makes this VIM into a limited vocabulary, isolated word recognition system some of the software written for this study was tailored so that one word or phrase takes on multiple meanings depending on the context in which it is used.

The VIM software provides the ability to create and store a vocabulary of 24 cues. The vocabulary consists of three basic parts (1) the cues, (2) the actions and (3) the voice signatures. The cues are messages describing the voice input words or phrases. The actions specify what is to be performed when the associated voice input is recognized. The voice signatures are digitized representations of the voice input associated with the cues. The vocabulary is stored in three buffers associated with each of the three basic parts.

The VIM software allows two alternate actions to be performed (1) a command to the EWS and (2) a call to a subroutine which has been appended to the VIM software. The appended subroutine feature was used in this study to invoke programs for home, telephone and ECU control. The action which controls the EWS was used in the home control program to activate lights and appliances.

The limited vocabulary of 24 cues proved to be an insufficient amount of phrases, so a utility program was written to give the ability to change vocabularies when special activities were to be performed. The program was written so that it stores a secondary vocabulary, of 24 phrases, in a buffer and switches the two vocabularies byte for byte, therefore, adding 24 separate phrases and increasing the vocabulary of available voice input.

The utility programs are used by the appended subroutines for two reasons (1) to switch to new vocabularies and (2) to get voice input. The appended subroutines are used to provide Home, Telephone and ECU control.

Home Control

The EWS allows for the control of up to 16 devices. Sixteen words of the vocabulary could designate the 16 devices and two more words could be used for their control (i.e., on or off) resulting in the use of 18 of the 24 available words in the limited vocabulary. Since many of the devices to be controlled are the same (e.g., overhead lights and floor lamps), but located in different parts of the home (e.g., bedroom and living room), the same vocabulary word can used to designate these devices. An initial room identifier directs in which room the device to be controlled is located. Likewise, by keeping up with the on/off status of the devices with the computer, the need for using two words for the purpose of turning each device on or off can be eliminated. Therefore the number of vocabulary words or phrases required can be decreased, leaving more words or phrases for other tasks.

The Home Control program was arbitrarily subdivided to control four rooms (1) bedroom, (2) kitchen, (3) living room and (4) playroom. Each room was allowed four devices to be controlled by the EWS (1) the overhead lighting, (2) a floor lamp, (3) an appliance and (4) a miscellaneous device. When a command is executed, if a device is on, it is turned off, and vice versa. If the status of the device is unknown, it is assumed to be off.

By the use of a menu driven program this approach trims the number of vocabulary words to control 16 devices to eight. The first menu allows a choice to be made of the four separate rooms. When a room is specified the appropriate actions are loaded into the action buffer, and then a menu displaying the four device choices appears. When a device is specified it is turned on or off, depending on its status.

When the Home Control subroutine is executed as shown by the flowchart in Figure 4, it goes through an initialization process and then displays a menu of the names of the four rooms to be controlled with voice input. There is a fifth item on this menu, "RETURN", which is a phrase used to exit the subroutine.

The program then waits for voice input. If the exit phrase is recognized the EWS actions are disabled, and the subroutine call actions are enabled, and control is transferred to the location where the Home Control subroutine was called. The EWS actions are disabled by storing an ASCII space, for each action, into the first location of the action buffer of the speech recognition software. The subroutine actions are enabled by storing an ASCII dollar sign in the first location of the action buffer for each subroutine call action. If one of the other four items is recognized the subroutine transfers control to a subroutine associated with that item.

The four room subroutines control the bedroom, kitchen, living room and playroom. These room subroutines display the name of the room that is currently being controlled, to verify to the operator that the correct input speech pattern was recognized, and then the subroutine, SWTCHRM, that moves the appropriate EWS device numbers into the action buffer of the speech recognition is executed. This assures that the device that is designated is the correct device activated. These device numbers are the unit codes (from EWS origin)

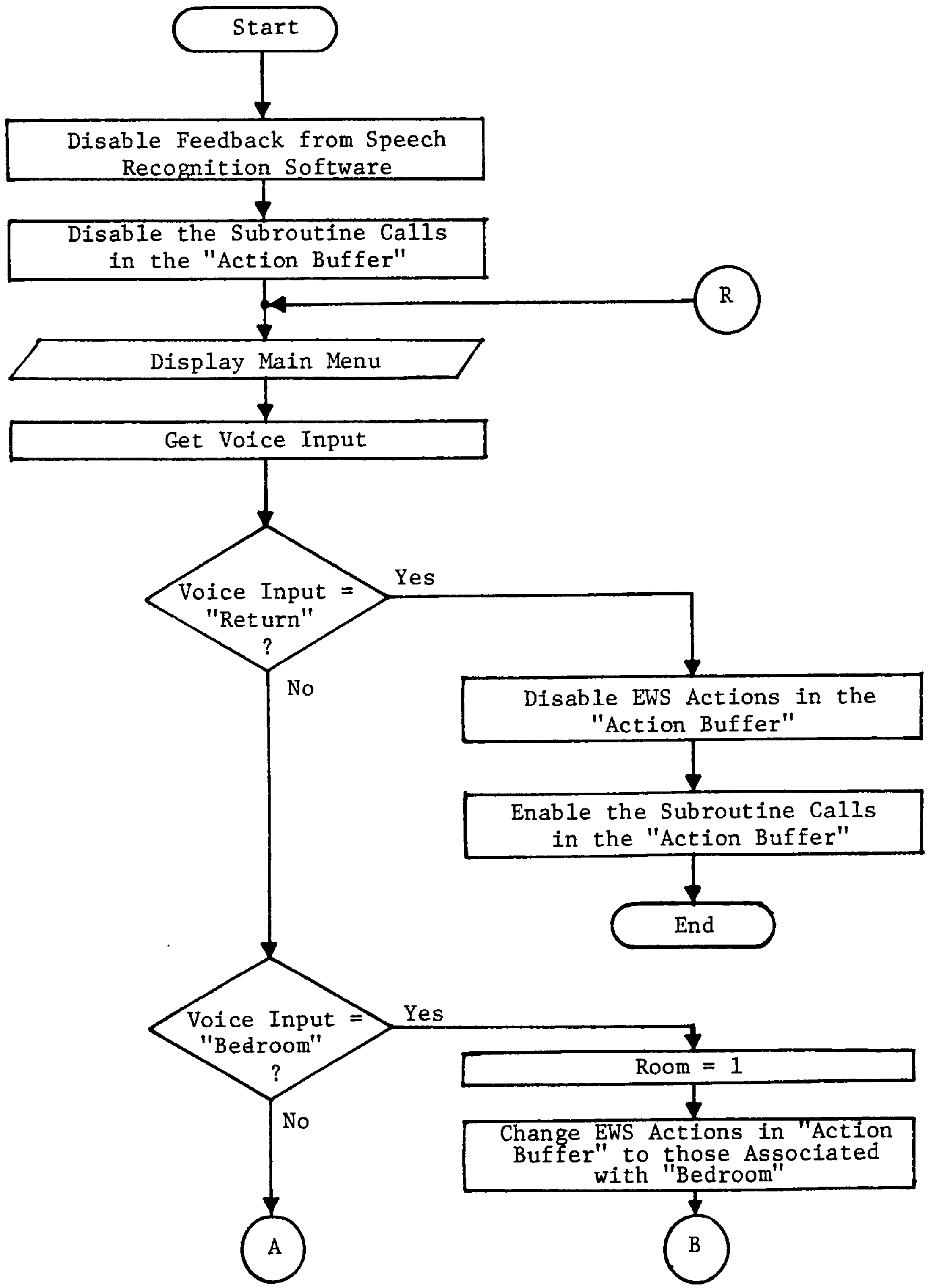

Figure 4. Flowchart of Home Control Subroutine.

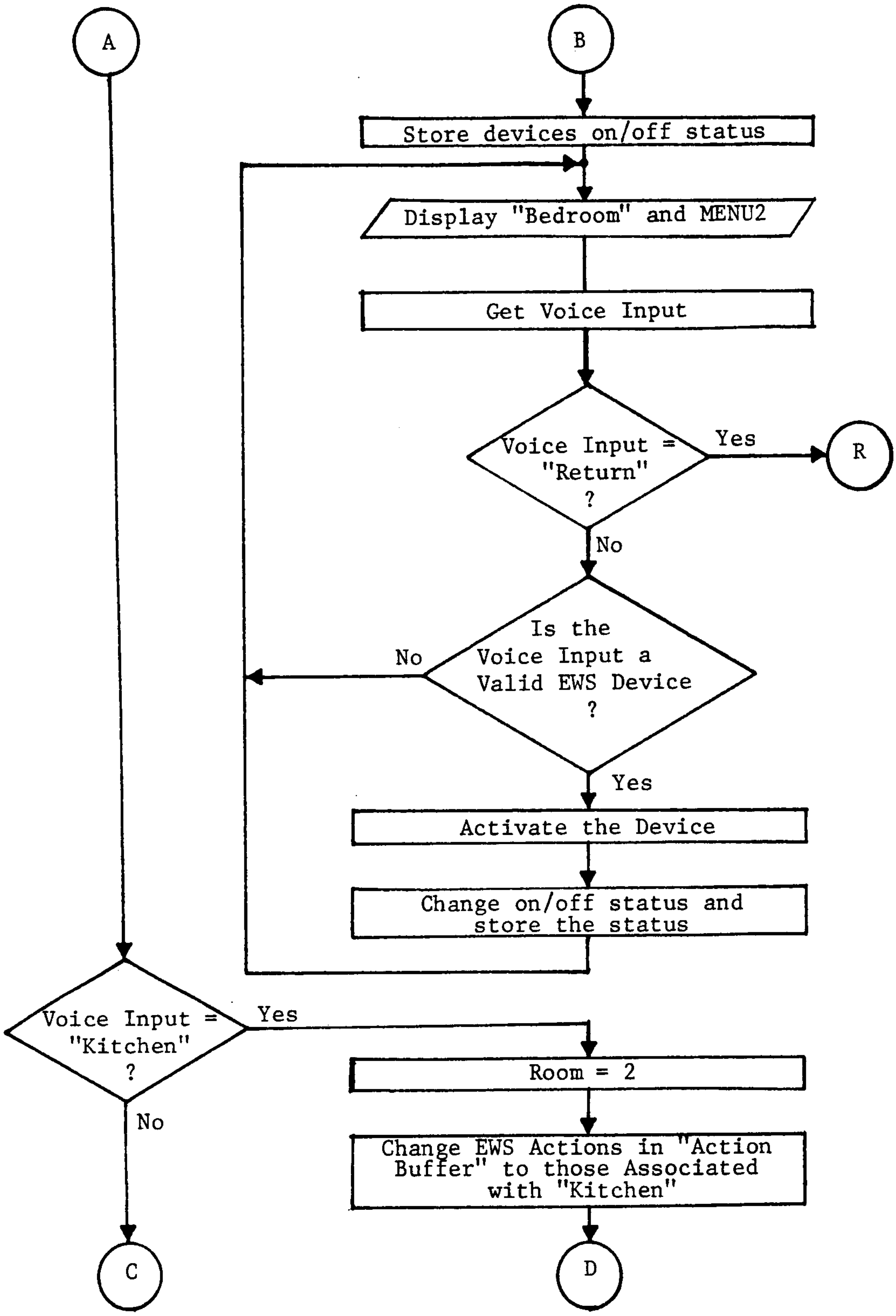

Figure 4. (Continued)

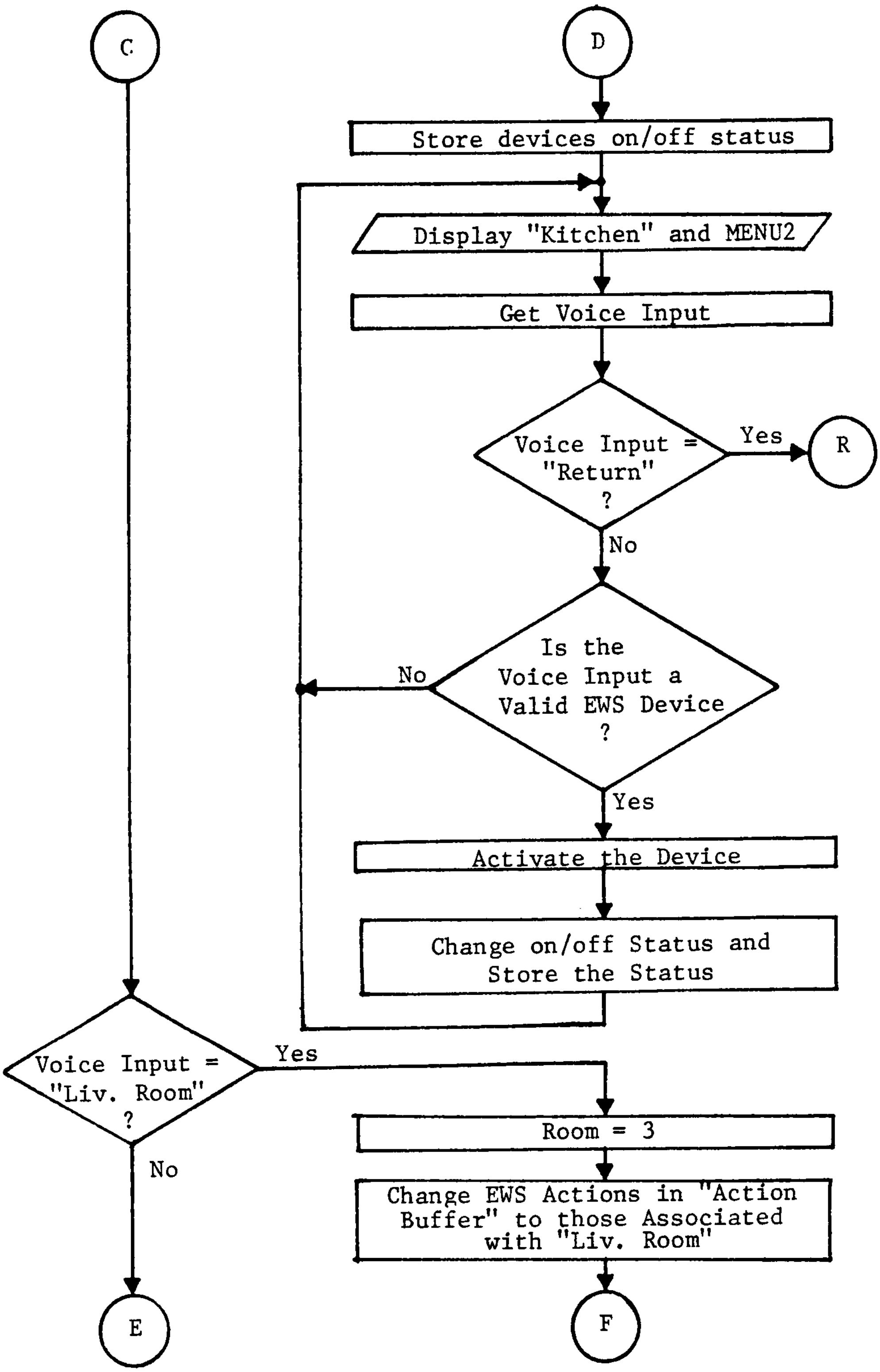

Figure 4. (Continued)

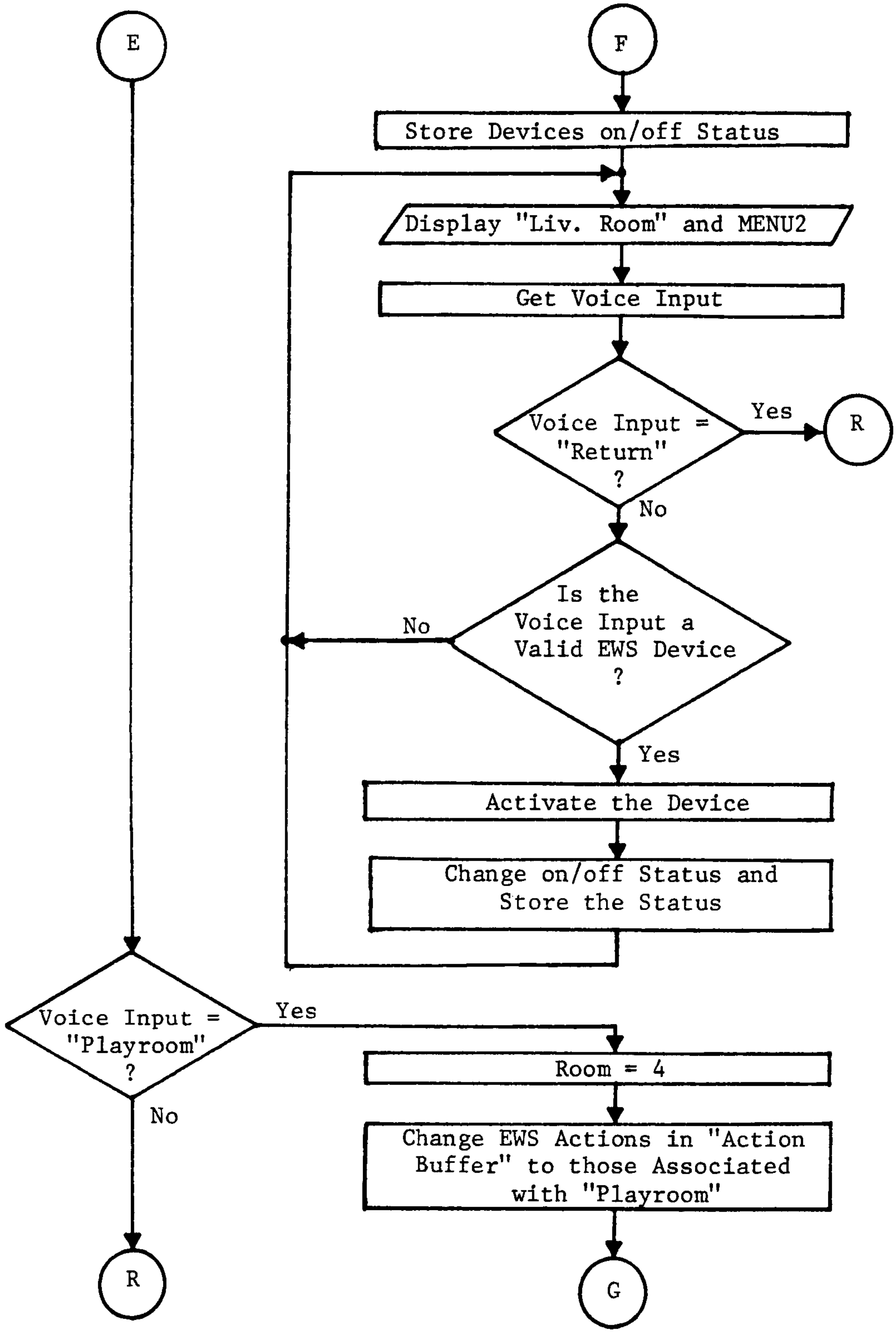

Figure 4. (Continued)

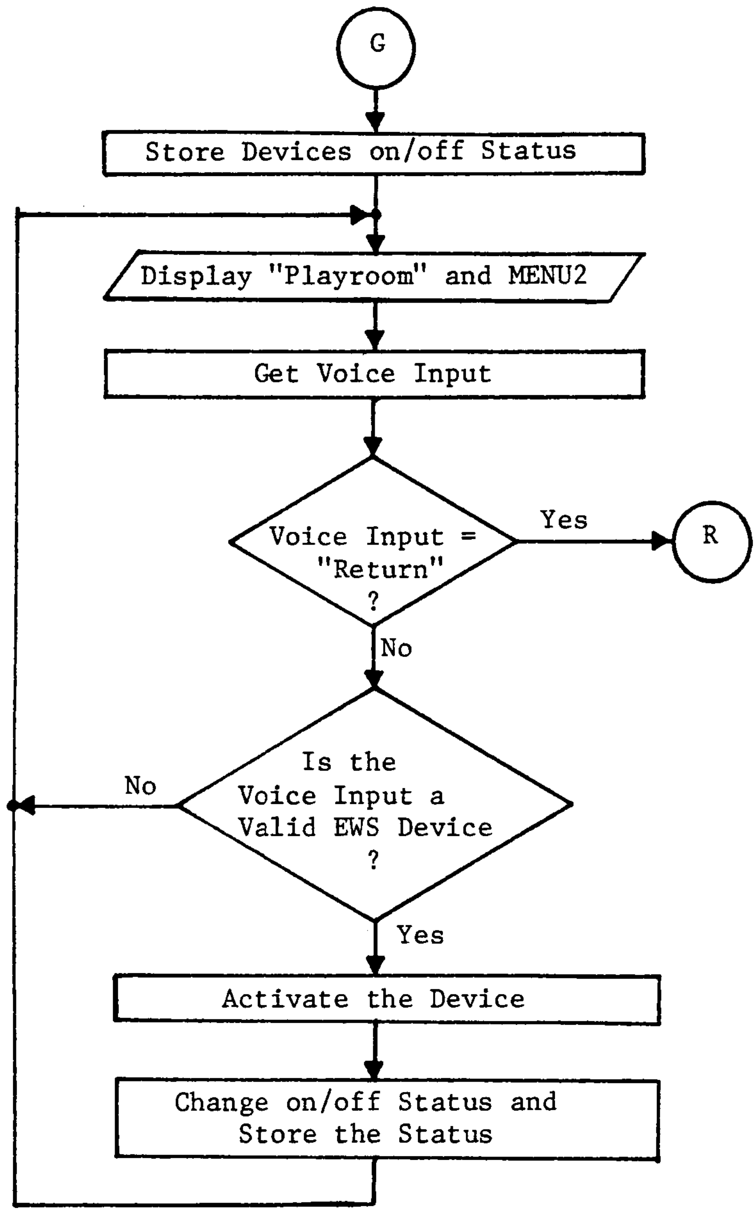

Figure 4. (Continued)

for the devices which are associated with the next menu. Control is then transferred to a subroutine, MENU2, which displays the menu containing the device choices and waits for voice input. If the voice input is one of the four choices then that device is activated (by the action feature in the speech recognition software), and control is returned to the room subroutine. The room subroutine then transfers control to subroutine SWITCH, which determines and stores the on/off status of the device actions. Exit from the room subroutines is by the recognition of the same phrase used to exit the main subroutine.

A visual display of which cue has been recognized is blocked by the subroutine FB so that a quick flash of the recognized cue on the screen does not interfere with the menu display process. The recognized cues which are associated with an EWS action are displayed. The Home Control subroutine handles the necessary output of recognized cues which are not associated with an EWS device.

Telephone Control

The telephone control subroutine is executed upon the recognition of the word "TELEPHONE" during the execution of the VIM's speech recognition software. As shown in the flowchart in Figure 5, an initialization process occurs first, and then the primary menu is displayed. During the initialization the primary vocabulary is switched with a secondary vocabulary. The secondary vocabulary contains the voice signatures associated with the telephone control.

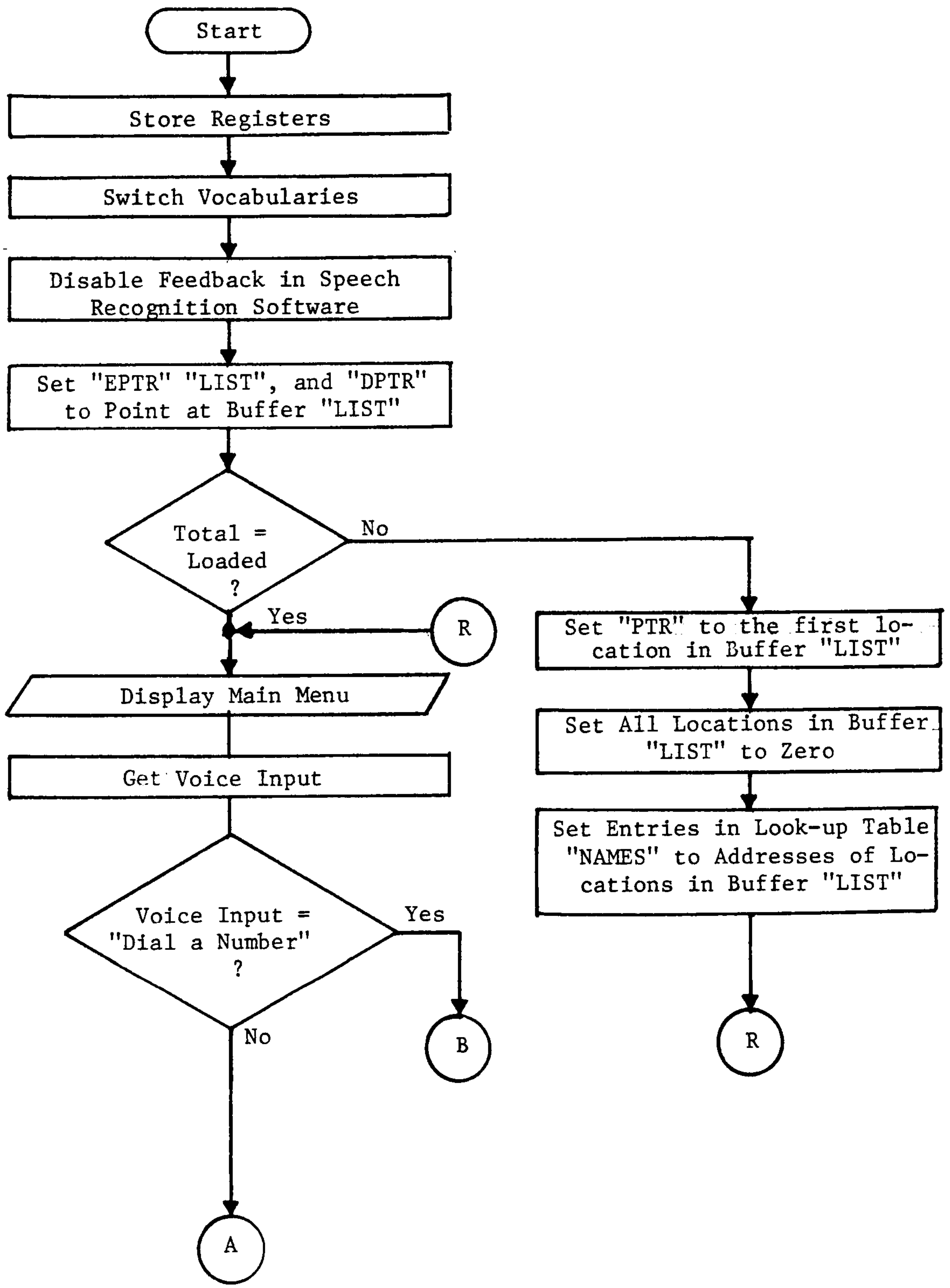

Figure 5. Flowchart of Telephone Control Subroutine.

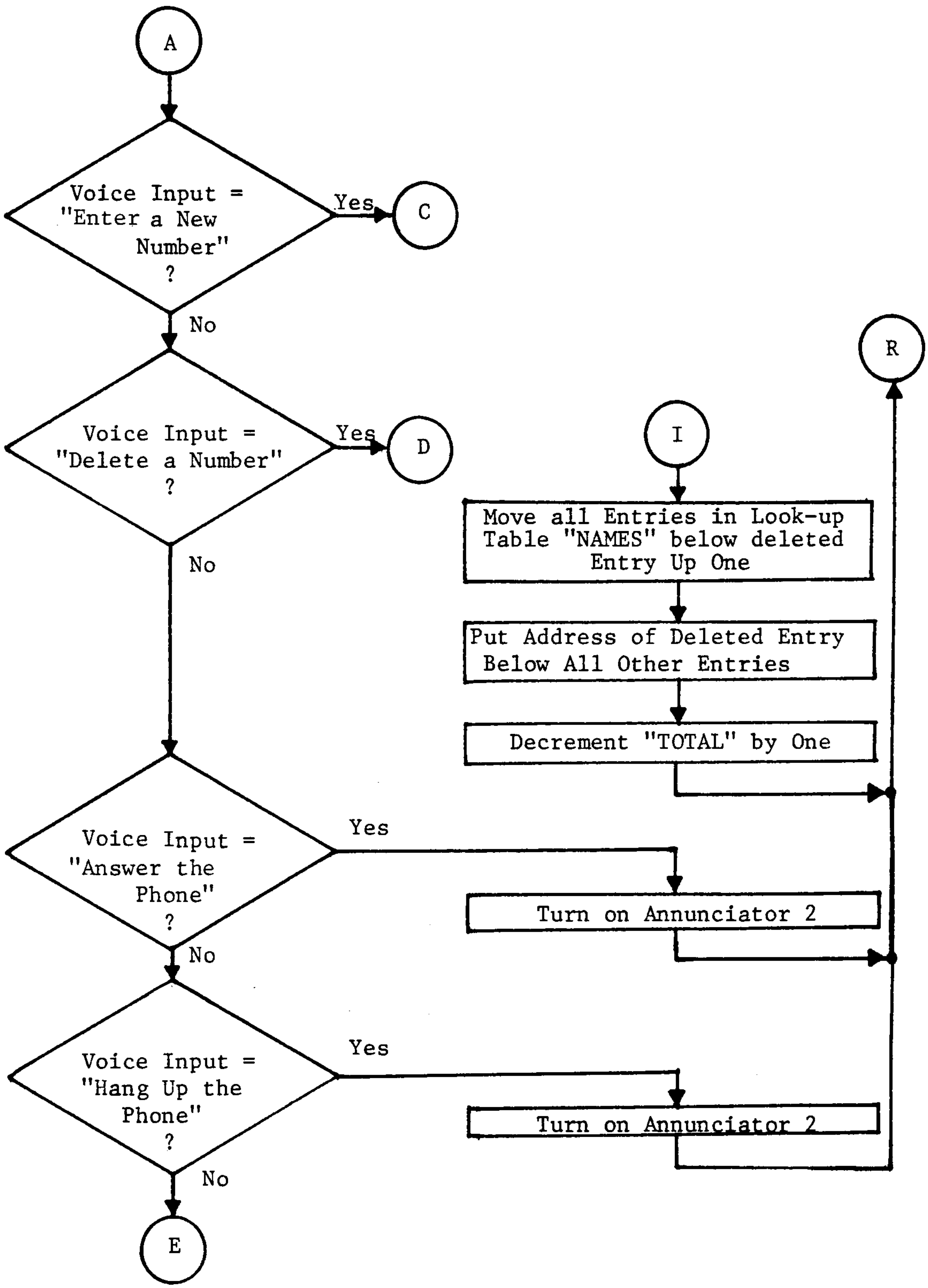

Figure 5. (Continued)

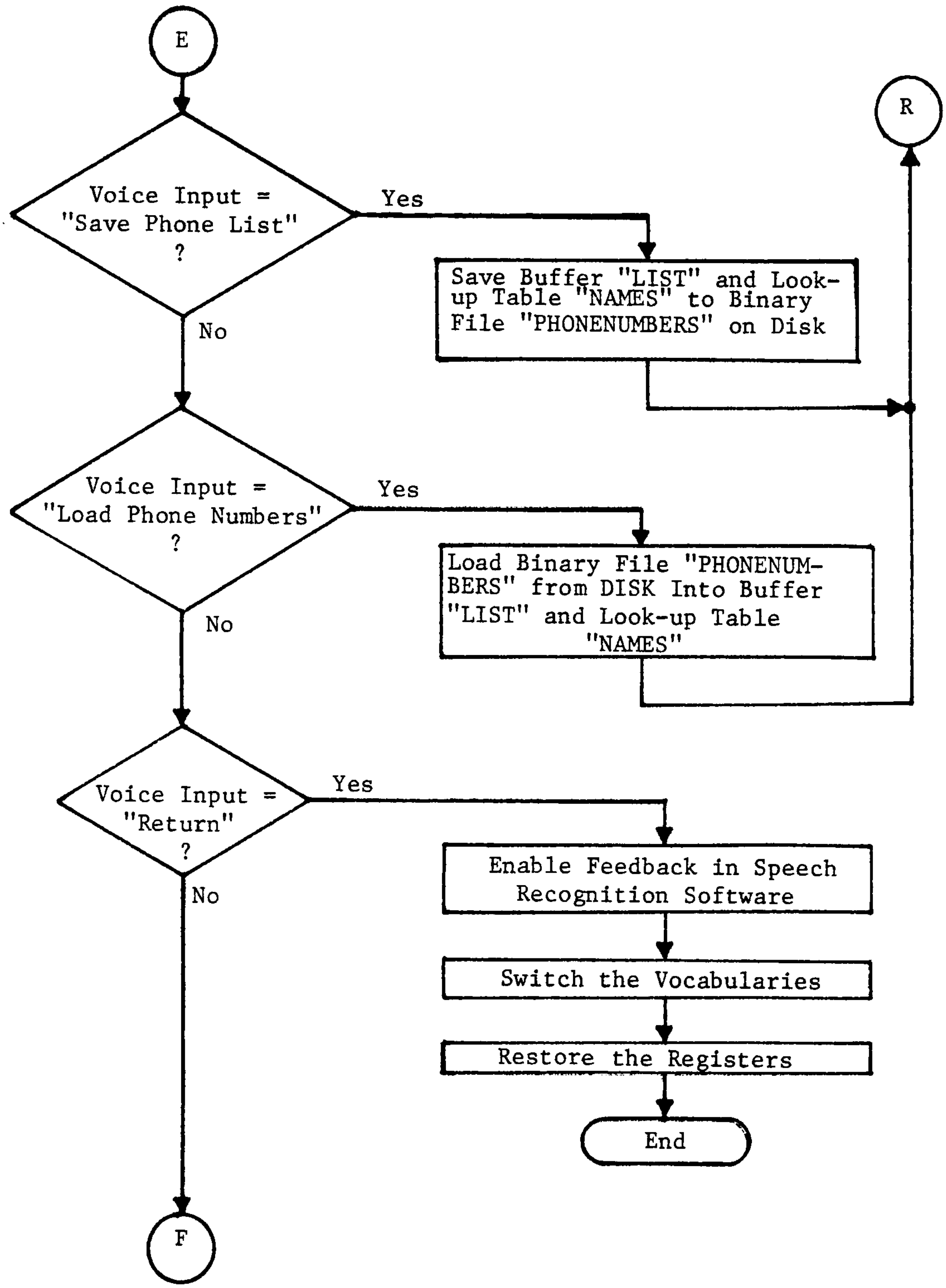

Figure 5. (Continued)

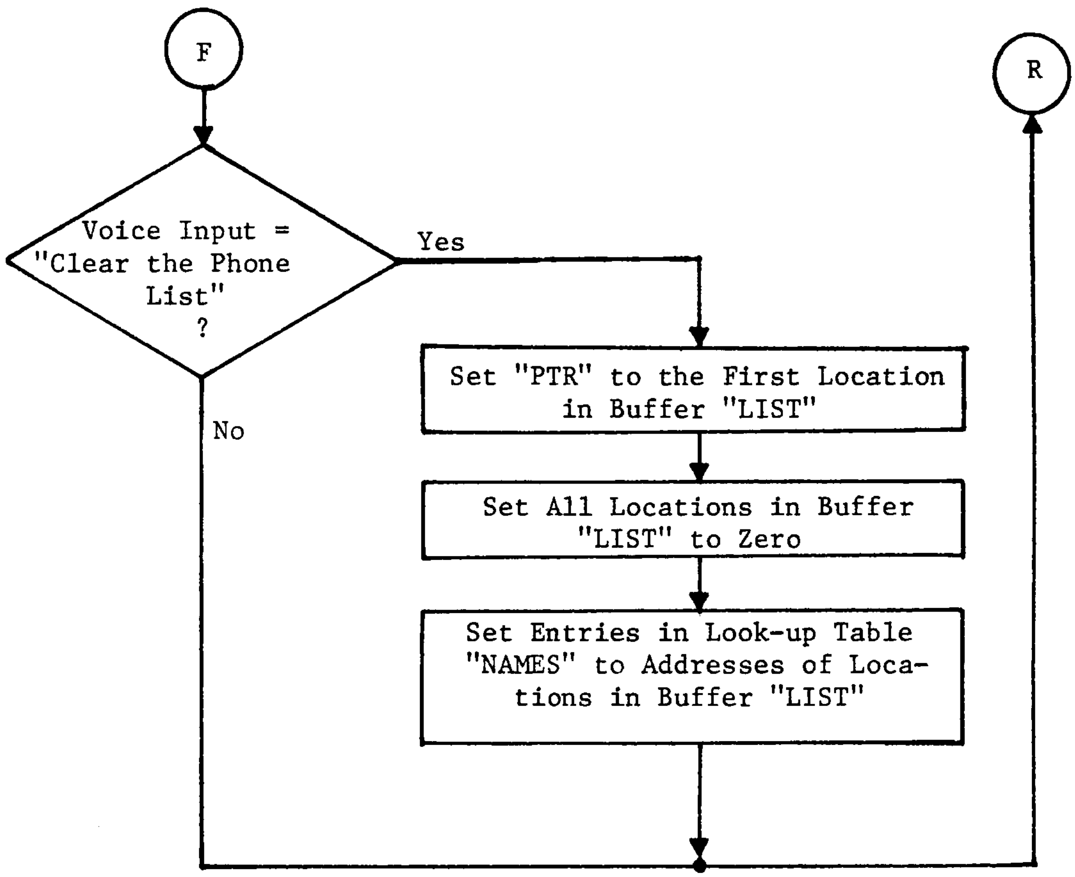

Figure 5. (Continued)

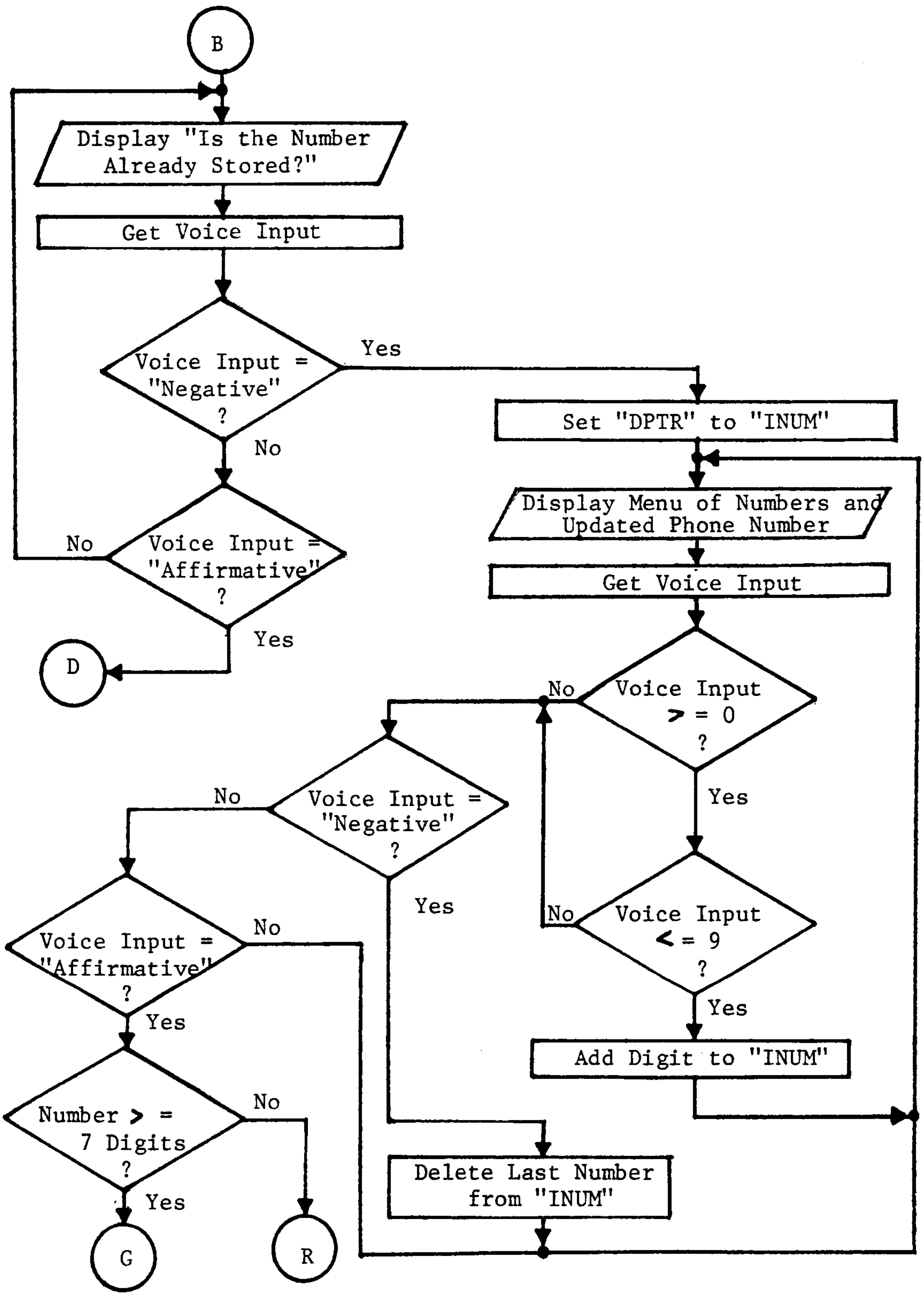

Figure 5. (Continued)

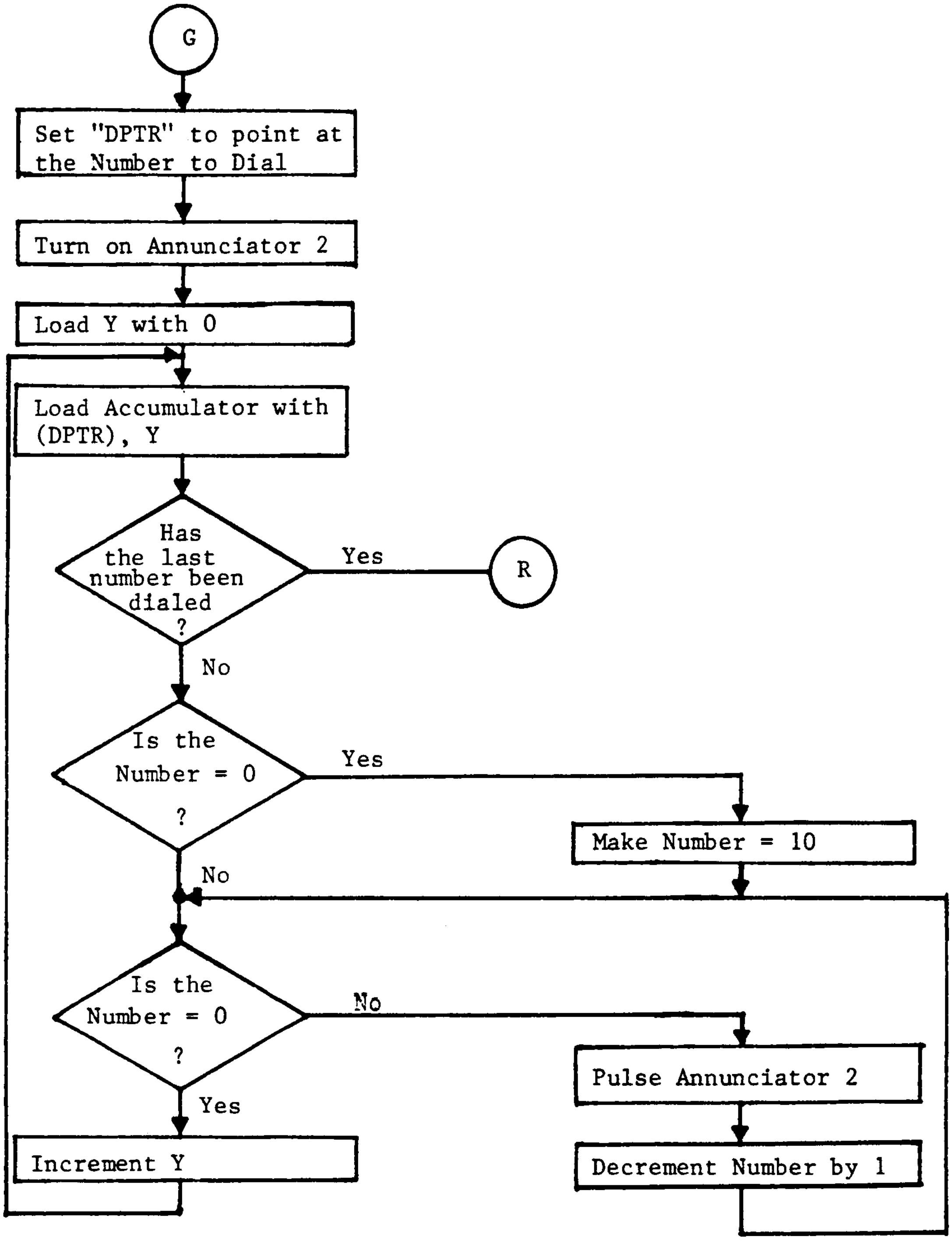

Figure 5. (Continued)

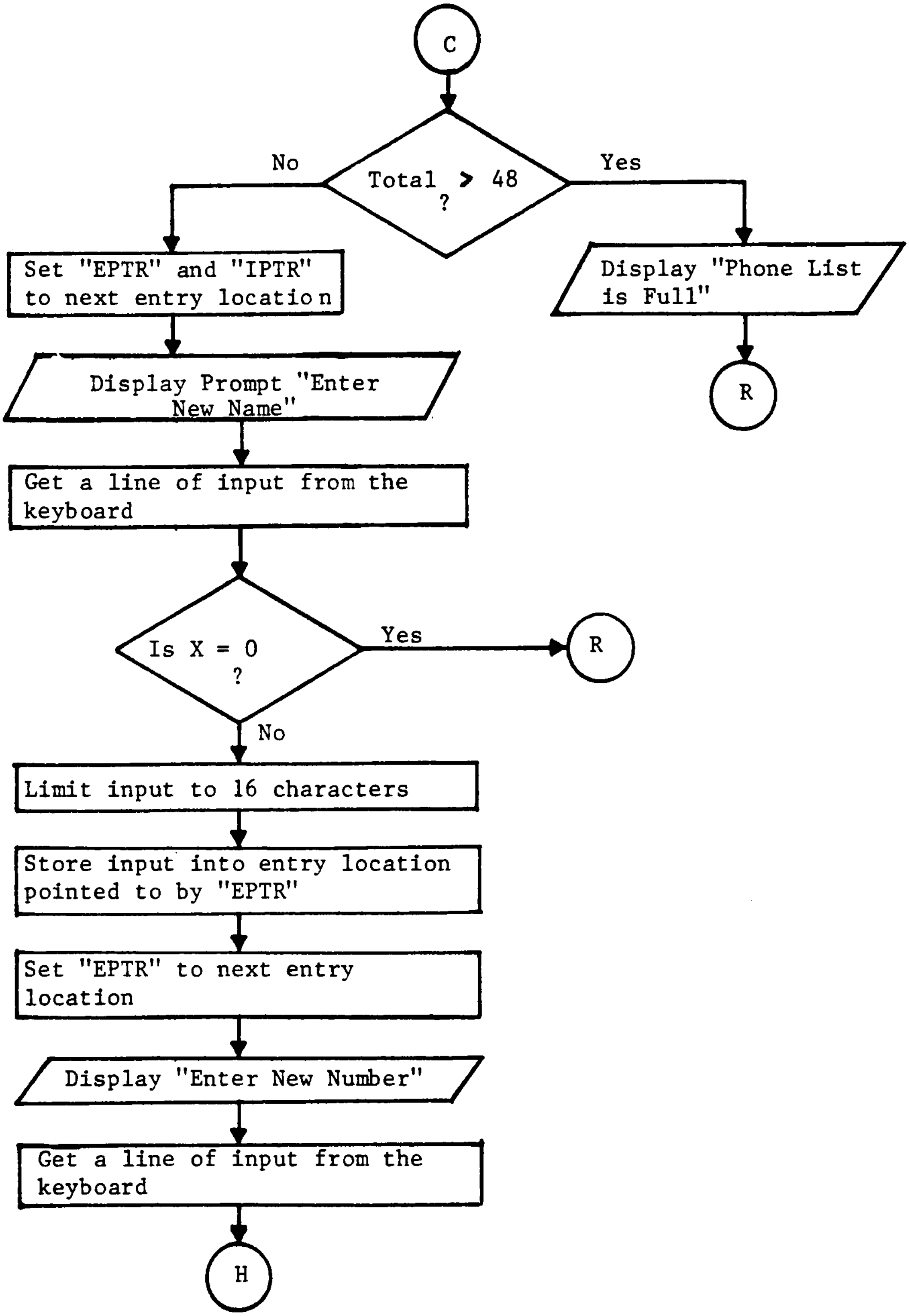

Figure 5. (Continued)

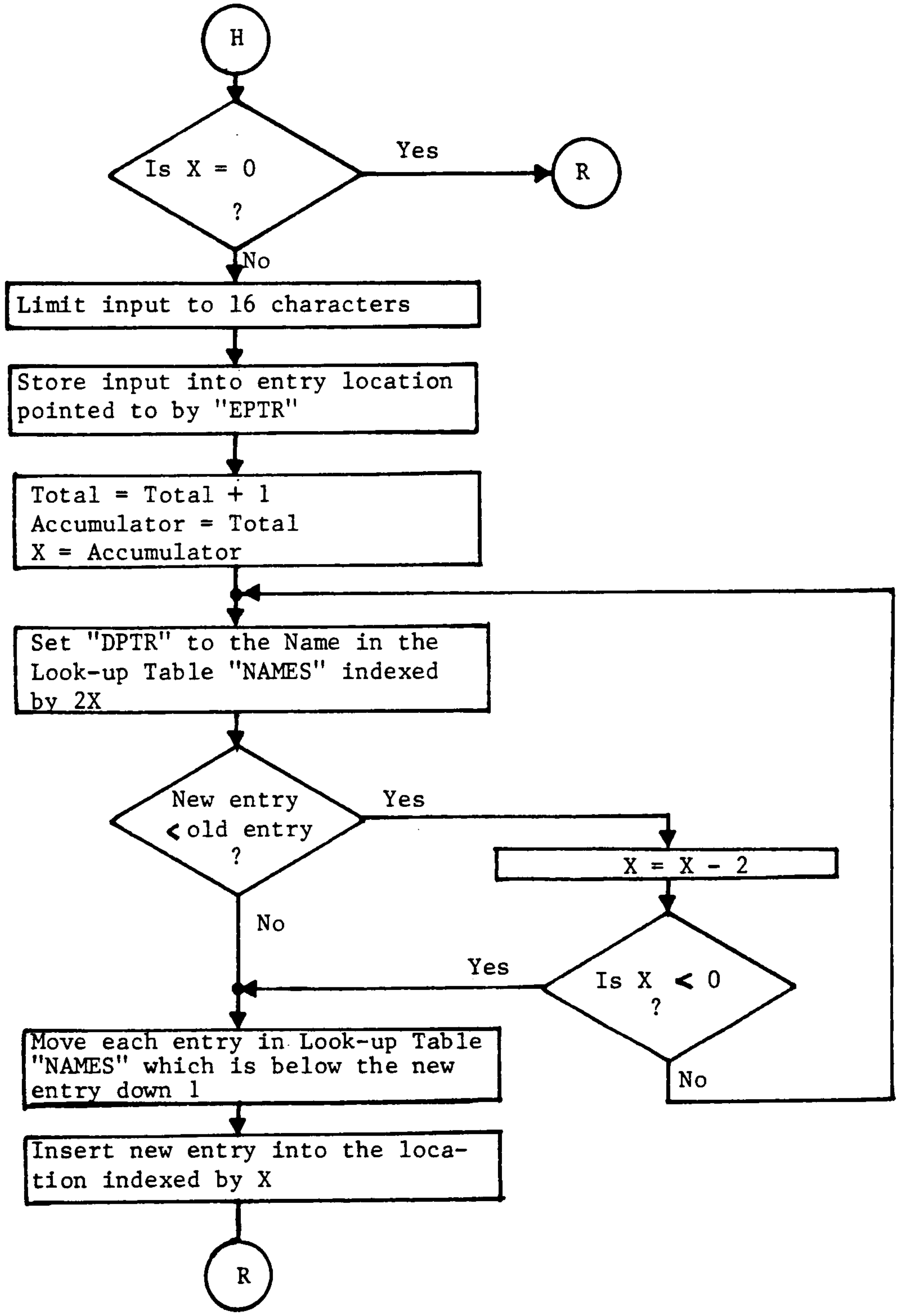

Figure 5. (Continued)

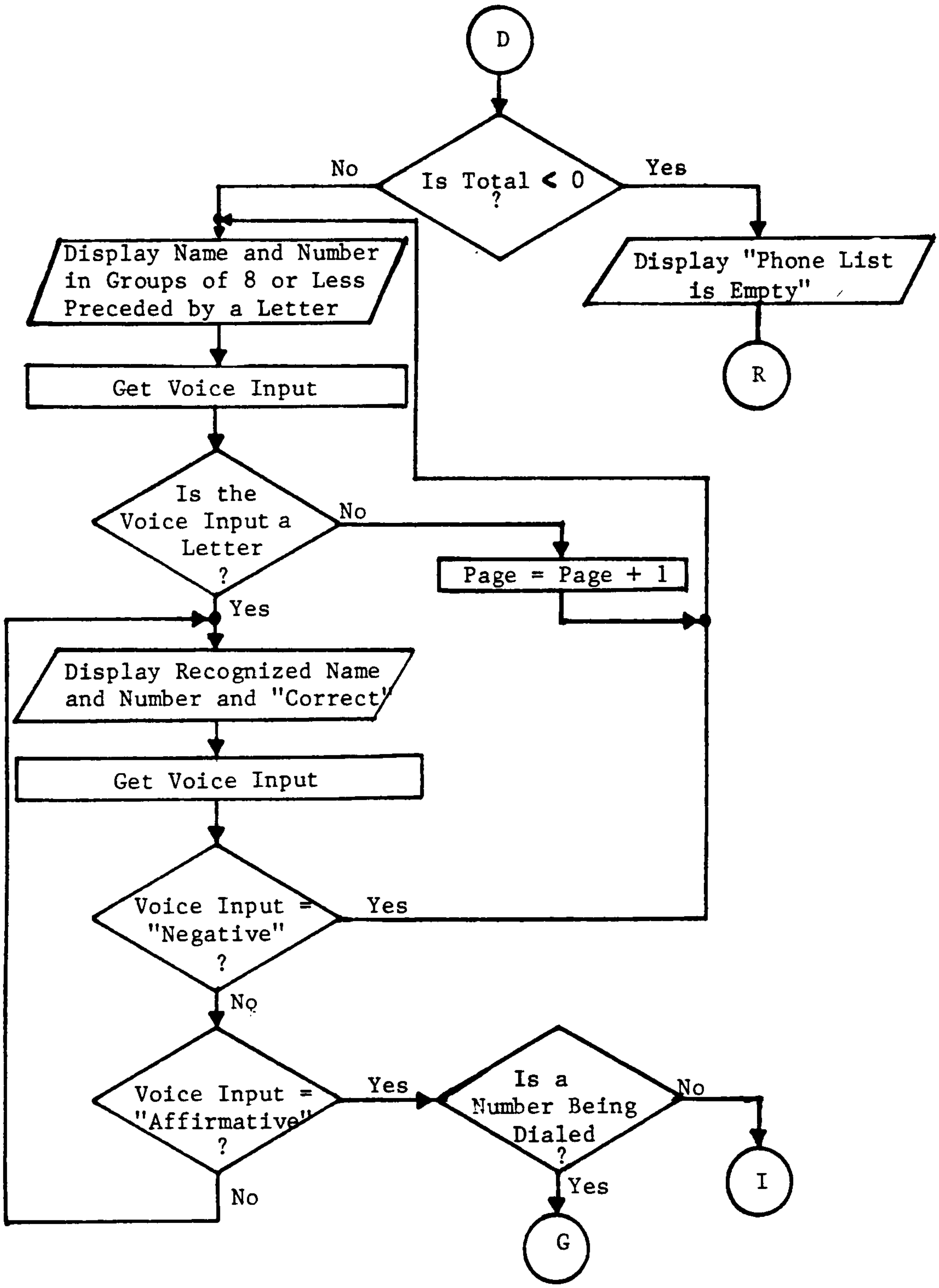

Figure 5. (Continued)

All feedback from the speech recognition software is disabled. The program then waits for voice input. If the voice input is recognized as one of nine choices then the appropriate subroutine is called. These choices are listed in Table 3.

Table 3. Main Menu for Telephone Control

A.	DIAL A NUMBER
B.	ENTER A NEW NUMBER
C.	DELETE A NUMBER
D.	ANSWER THE PHONE
E.	HANG UP THE PHONE
F.	SAVE PHONE LIST
G.	LOAD PHONE NUMBERS
H.	RETURN
I.	CLEAR THE PHONE LIST

Subroutine DIAL is called when the voice input is "dial a number." After displaying "IS THE NUMBER ALREADY STORED?", this program waits for voice input. Upon a "NEGATIVE" input to the question "IS THE NUMBER ALREADY STORED?", section DINUM is called and begins by setting a pointer, DPTR, to the location of buffer INUM. A menu of the ten digits, zero through nine, and the associated voice input is displayed. Due to the lack of reliability of the VIM to recognize the monosyllabic digits, phrases which found by experiment to give good recognition results were used to designate the digits. The phrases chosen are listed in Table 4. A loop is executed to get, by voice input, the phrases associated with the digits in the phone number to be dialed. If the voice input is a

Table 4. Letters/Digits and Associated Voice Input

LETTER/DIGIT	VOICE INPUT
A/1	ABLE
B/2	BAKER
C/3	CHARLIE
D/4	DOG
E/5	ECHO
F/6	FOXTROT
G/7	GRUMPY
H/8	HOTEL
I/9	IDAHO
J/0	JULIET

phrase associated with one of the ten digits, the associated digit is added to the phone number and then the updated phone number is displayed. If the phrase "NEGATIVE" is recognized then the last digit of the phone number is deleted. The loop continues until the phrase "AFFIRMATIVE" is recognized or more than eleven digits have been entered. When the phrase "AFFIRMATIVE" is recognized the program checks to see if the phone number contains at least seven digits. If it does, then the phone number is dialed. Upon the "AFFIRMATIVE" input, if the phone number has less than seven digits or more than eleven digits, control is transferred to the main menu.

Upon an "AFFIRMATIVE" input to the question "IS THE NUMBER ALREADY STORED?" subroutine DSPLY is called which allows the user to choose the previously stored phone numbers they desire to dial. Subroutine DIAL then sets a pointer to the phone number to be dialed. A time delay loop is first executed which allows enough time for the dial tone to be established. The phone number is dialed using the pulse dialing technique. Dial pulses are created by momentarily opening the loop. The following standards for dial pulses were met (1) pulsing rate of 10 pulses per second (+/- 10%), (2) each pulse was 58% to 64% open and (3) interdigit time was greater than 600 milliseconds. By using the AID, described earlier in the Hardware section, the above standards were achieved.

If the voice input is "ENTER A NEW NUMBER", then the subroutine ENTER is executed. This subroutine first checks to see if the phone list is full. A total of 48 names and associated phone numbers can be stored in the phone list. If the phone list is full a message "Phone List is Full" is displayed on the screen, and control is returned to the primary menu. If the list is not full the subroutine sets two pointers to the next entry location in the buffer LIST. The subroutine then allows keyboard entry of the new name and phone number. Both the name and phone number are limited to 16 characters. Control is then transferred to SORT.

SORT is used to put the names into alphabetical order. This section contains a double loop. The outer loop sets the pointer, DPTR, to the name to be compared to the newly entered name, starting at the end of the list and progressing toward the beginning until the correct alphabetical location for the new name is determined. The inner loop compares the new name entry, character by character, with the previously stored names, if any, and when the appropriate location is determined, control is transferred to INSERT. INSERT places the new entry into its correct alphabetical location by moving the entries in the look-up table NAMES, which contains the addresses of the locations where the entries are stored.

If the voice input is "CLEAR THE PHONE LIST", then the subroutine CLRLST is executed. This subroutine sets the variable TOTAL to a negative one, sets the variable NPGE (i.e., number of pages) to zero, sets pointer, PTR, to the first location in the buffer LIST and stores zeroes into all locations of the buffer LIST. Control is then transferred to section LUINIT.

LUINIT sets the entries in the look-up table NAMES to the addresses of locations to be used in the buffer LIST. This creates a list of available locations which is used by the subroutine ENTER to determine where to store new entries.

The subroutine DSPLY, which is invoked by both subroutine DIAL (discussed earlier) and DELETE (which will be discussed later), first checks to see if the list is empty. If the list is empty a message "Phone List is Empty" is displayed, and control is transferred back to the primary menu. If the list is not empty the contents of the list are displayed in pages. Each page contains up to eight entries. Each entry on a page is designated by a letter of the alphabet, from A to H. The first entry is preceded by an A, the second entry by a B, and the eighth entry by an H. The names listed in Table 4 which begin with the letters, and found experimentally to give good recognition results, are used to represent the letters. If the voice input to DISPLY is one of the names in the list, then DPTR is set to point to the name and phone number associated with that letter. The validity of the recognized name is confirmed by subroutine CDEL.

If the name is confirmed in CDEL then control is returned to the subroutine which called DISPLY. If the name is not confirmed in CDEL, then the program displays the page of entries which was last displayed, giving the user another opportunity to get the correct entry in case of poor recognition or accidental input. If the voice input recognized is not a name representing a letter between A and H, then the next page of entries is displayed. If no entry is designated and there is not another page to display, then control is transferred to the primary menu.

If the voice input is "DELETE A NUMBER" then control is transferred to subroutine DELETE. This subroutine first transfers control to subroutine DSPLY to get, by voice input, the entry to delete. The subroutine DSPLY returns with the value of the entry to be deleted, which was input and confirmed by voice. The deletion process takes place by moving up one location the entries in the look-up table NAMES which are below the entry being deleted. The address of the entry which was deleted is then placed at the end of the stored entries in the look-up table NAMES, making it the next available location. The variable TOTAL, which counts the number of entries, is decreased by one.

If the voice input is "ANSWER THE PHONE" then control is transferred to subroutine ANSWER which turns on the phone by activating annunciator number two. If the voice input is "HANG UP THE PHONE", then control is transferred to subroutine HANGUP which turns off the phone by deactivating annunciator number two.

If the voice input is "LOAD PHONE NUMBERS" then control is transferred to subroutine LOAD which loads the binary file "PHONENUMBERS" from disk into the buffer LIST. The binary file also contains the entries for the look-up table NAMES which are also loaded.

If the voice input is "SAVE PHONE LIST" then control is transferred to subroutine SAVE. This subroutine saves the buffer LIST and the look-up table NAMES to the binary file "PHONENUMBERS" on the disk.

If the voice input is "RETURN" then the feedback from the speech recognition software is enabled. The secondary vocabulary is switched with the primary vocabulary, and control is returned to the speech recognition software.

Subroutine CDEL, which is used by DSPLY to insure the validity of the voice input, sets the pointer, IPTR, equal to the contents of the pointer, DPTR, and then transfers control to section CORR1. CORR1 displays the name and phone number followed by the question "IS THIS CORRECT?", then waits for voice input. If the voice input is "AFFIRMATIVE" then the accumulator is loaded with a one indicating the input is correct. If the voice input is "NEGATIVE" then a zero is loaded into the accumulator indicating the input is incorrect. Control is then transferred back to DSPLY. If the voice input is neither "AFFIRMATIVE" or "NEGATIVE" then CORR1 is repeated.

Environmental Control Unit Control

The ECU control subroutine is executed upon the recognition of the phrase "ENVIRONMENTAL CONTROL UNIT" during execution of the speech recognition software. The flowchart of the ECU control subroutine in Figure 6 shows in the program initialization the variables for the present scanner position, and the present television channels are set, through voice input, during the first execution of this subroutine. Thereafter they are assumed to be known; however, further in this subroutine, if the phrase "ENVIRONMENTAL CONTROL UNIT" is recognized, their values can be changed.

When inputting the present scanner position and present television channels, the program displays the current recognized input, then waits for the voice input of either of the phrases "AFFIRMATIVE" or "NEGATIVE." If the phrase "AFFIRMATIVE" is recognized, the accumulator is set to 0, indicating correct input. If the phrase "NEGATIVE" is recognized, the accumulator is set to 1, indicating incorrect input. This process takes place to insure the validity of the voice input.

The scanner position and the television channels are initialized so that the program can calculate the number of positions or channels to move upon entry of a given phrase. There are 14 positions on the scanner, 13 channels on one television and 18 channels on the other television. The positions of these three devices are changed by sequentially stepping forward through the possible positions.

After the initialization process a menu is displayed. The menu lists the devices controlled by the environmental control unit.

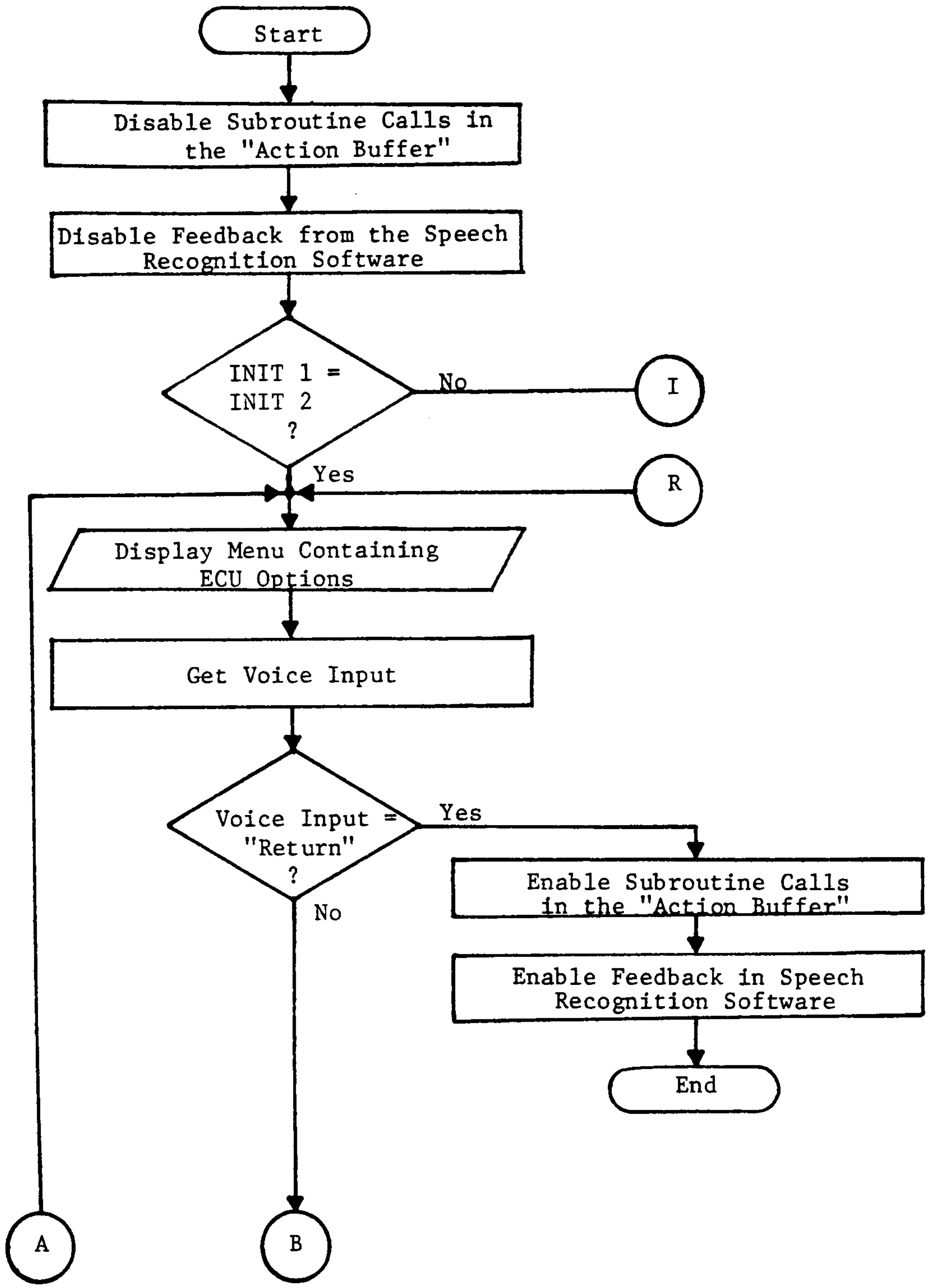

Figure 6. Flowchart of ECU Control Subroutine.

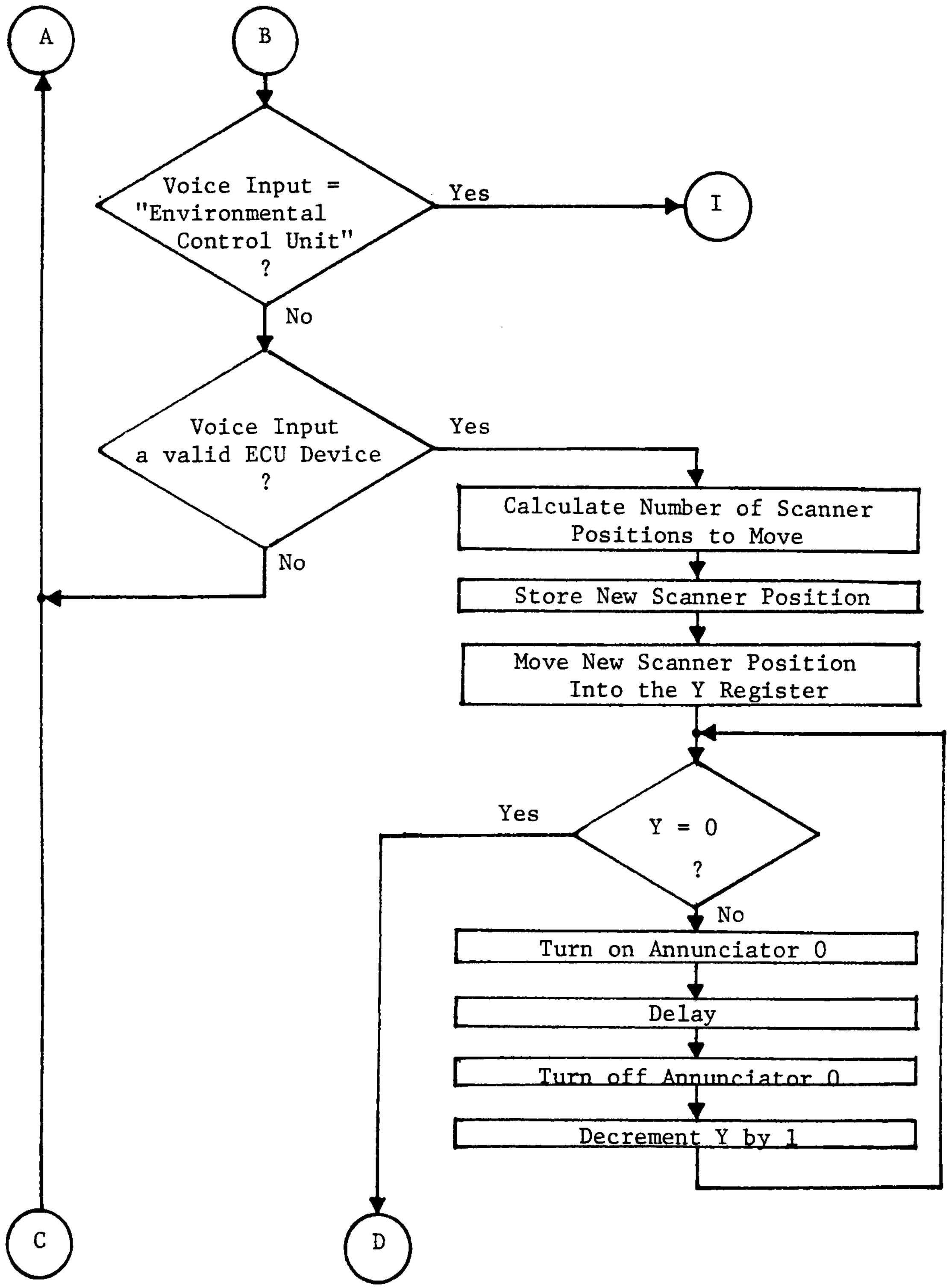

Figure 6. (Continued)

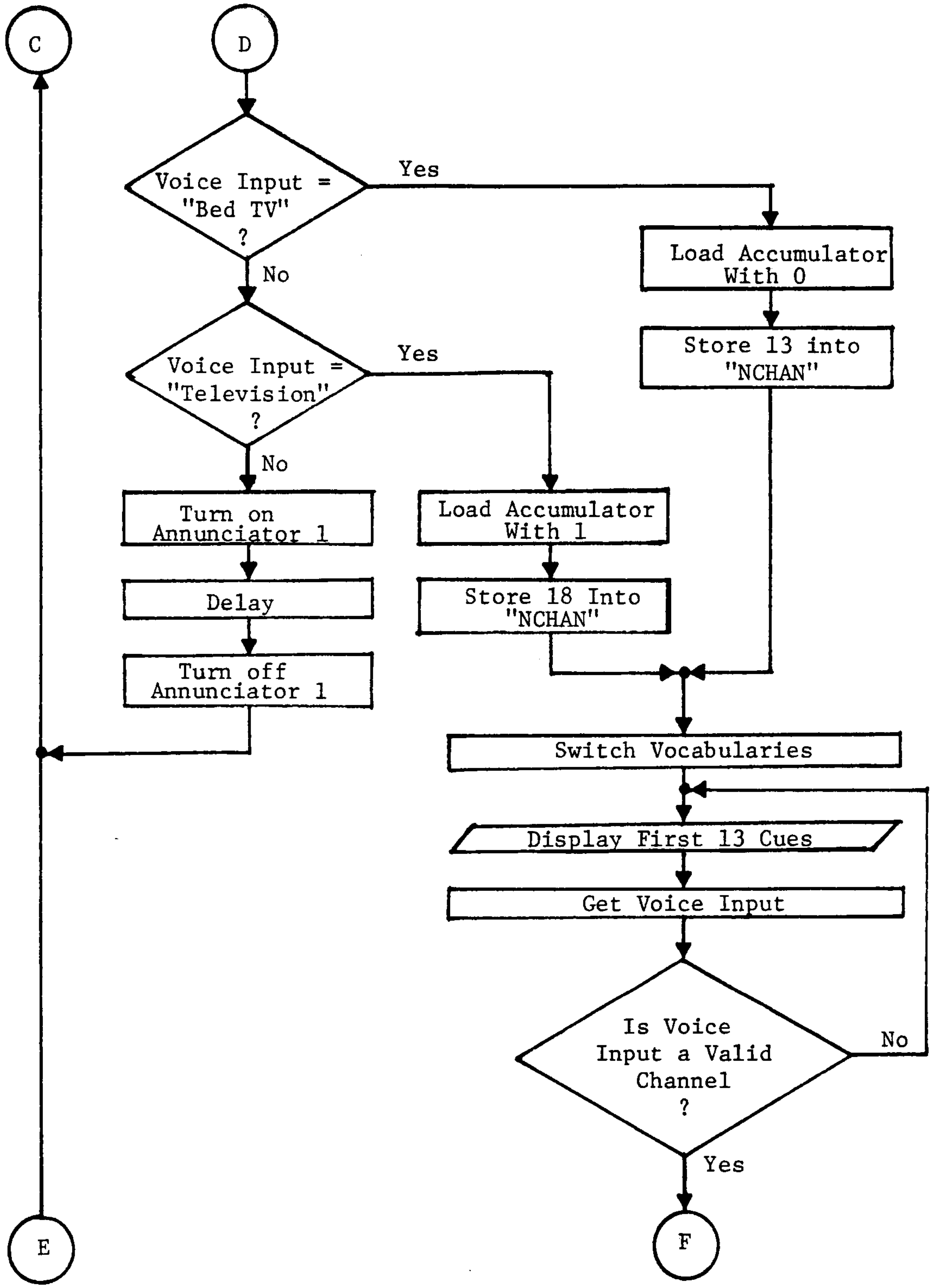

Figure 6. (Continued)

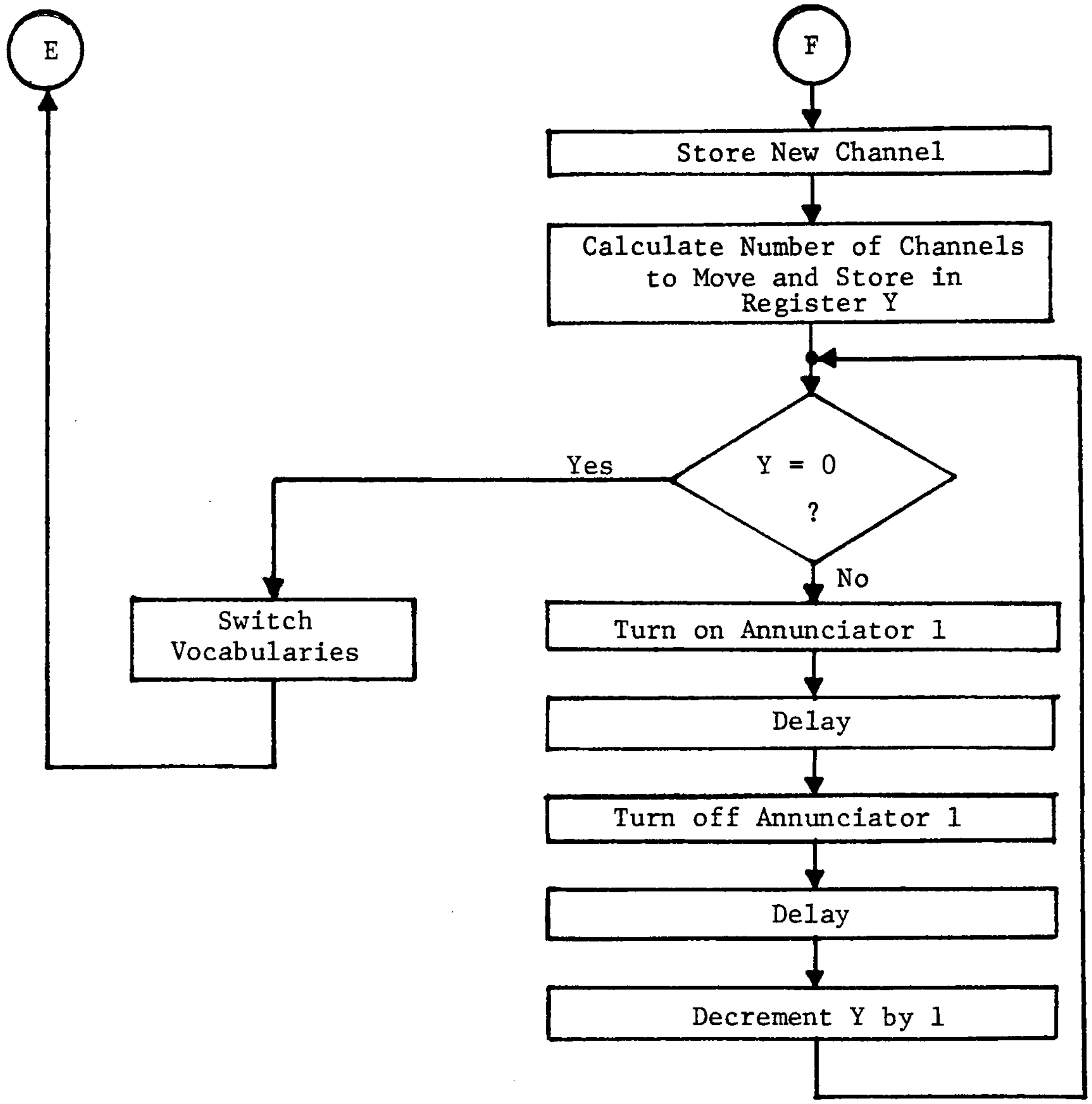

Figure 6. (Continued)

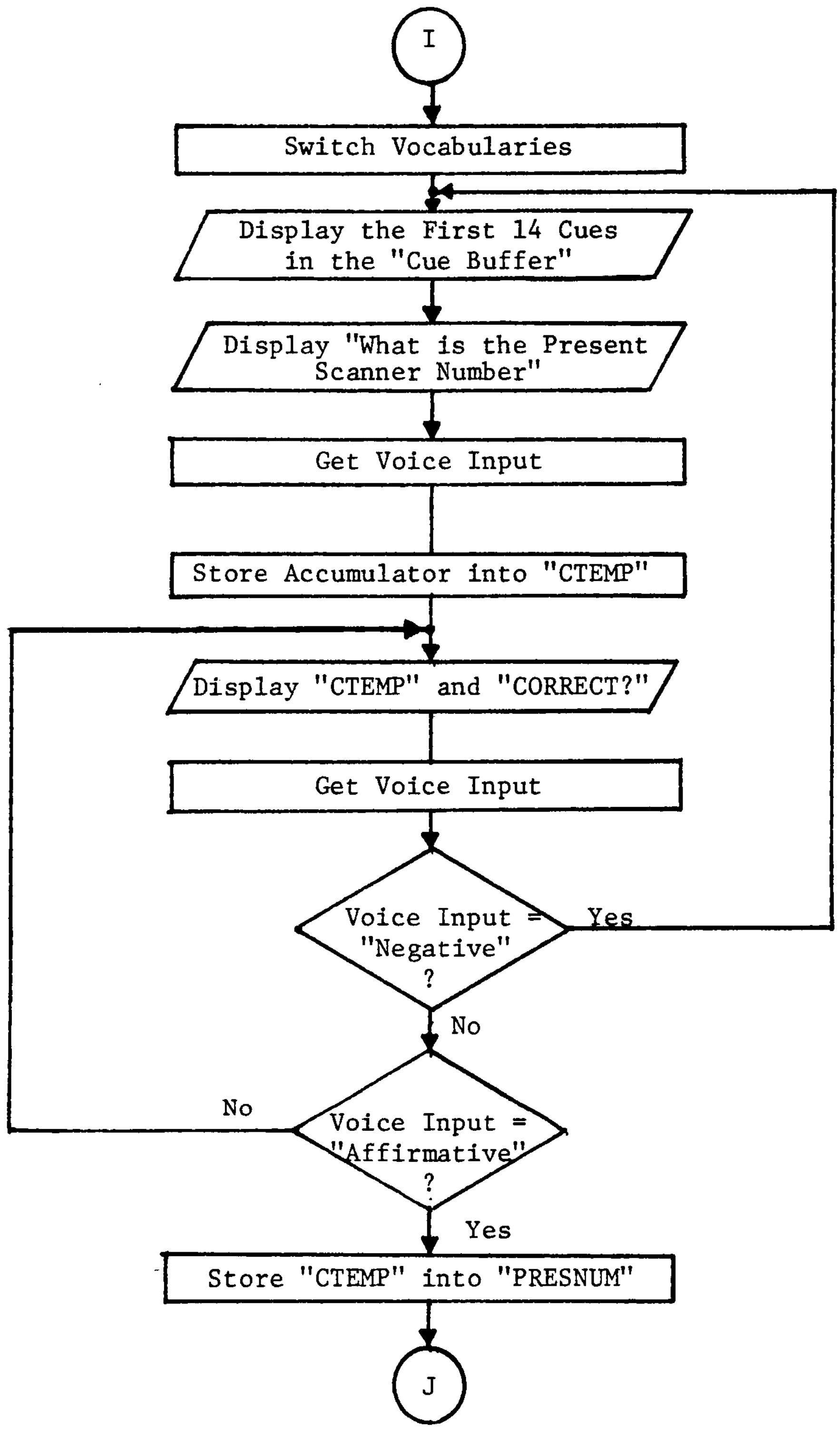

Figure 6. (Continued)

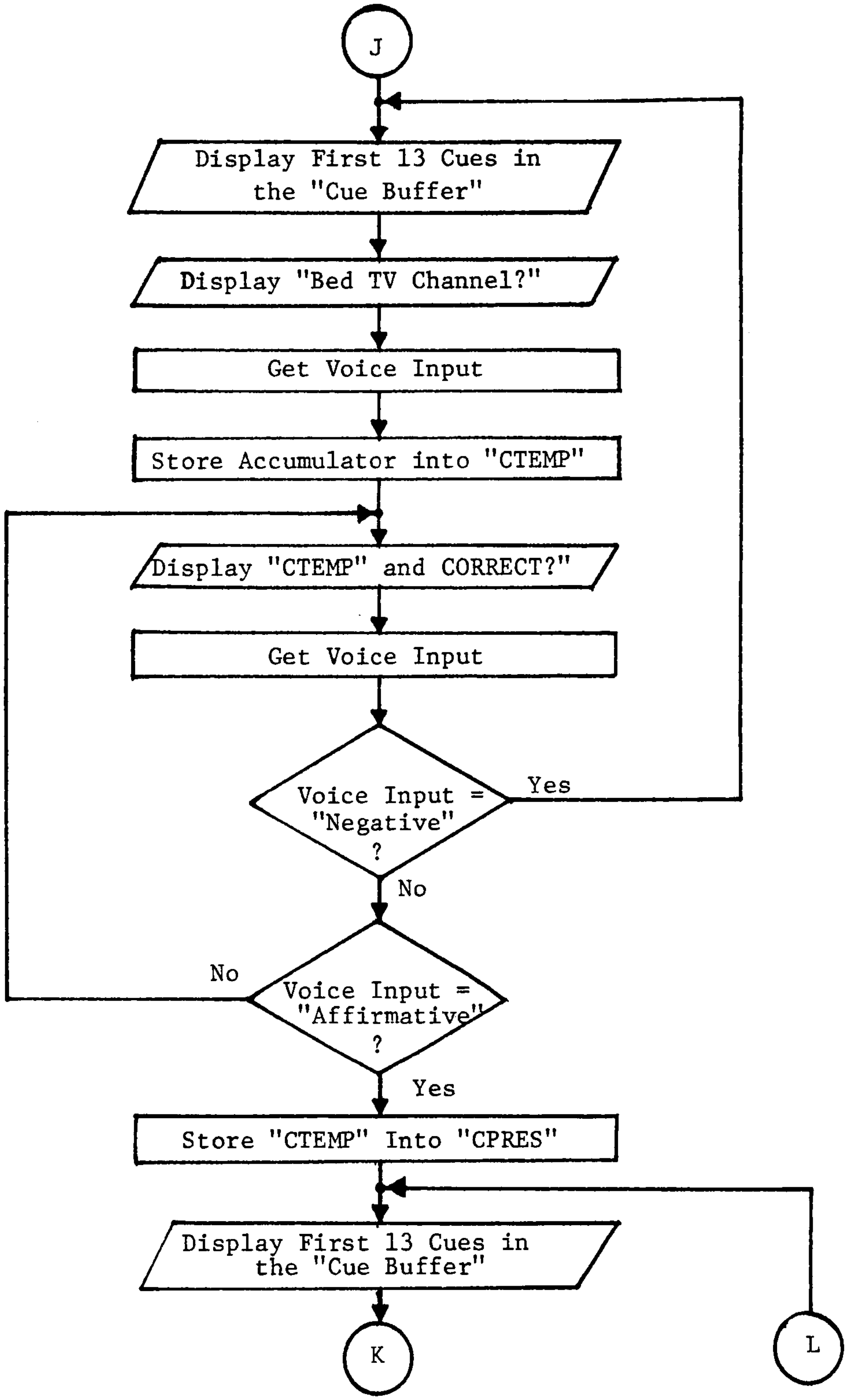

Figure 6. (Continued)

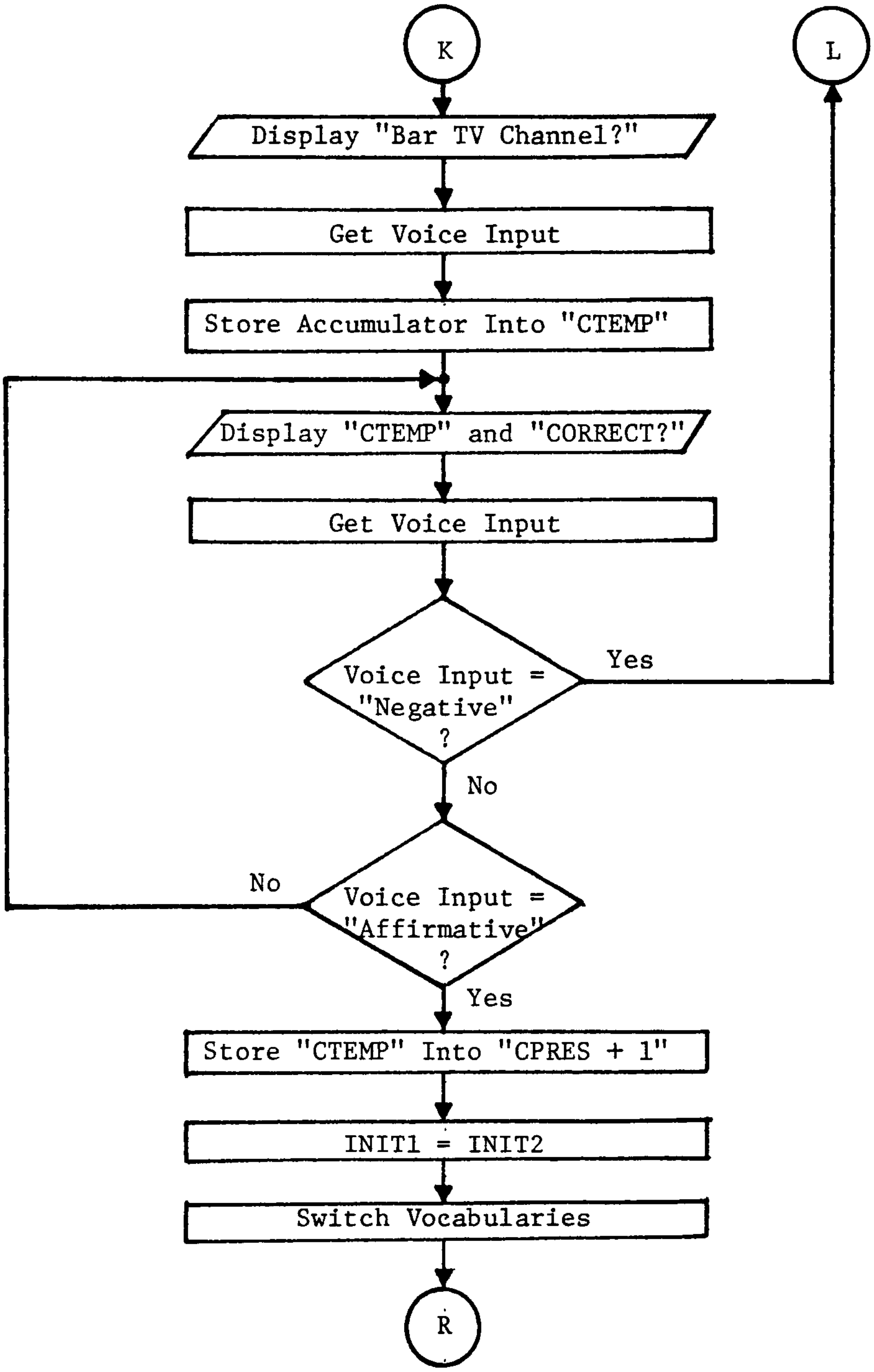

Figure 6. (Continued)

Then the subroutine waits for voice input. If the voice input is one of the menu options, control is transferred to subroutine MOVE; otherwise, the menu display process is repeated.

Subroutine MOVE first calculates the number of scanner positions to move the environmental control unit's scanner by subtracting the current scanner position from the desired scanner position. If the result is negative the program adds 14 to the number since scanning is cyclic. The new scanner position is stored as the variable PRESNUM, and control transferred to subroutine STEP which moves the scanner the correct number of positions. Delays are put into this routine to give the ECU's scanner time to respond accurately. Upon return from subroutine STEP the program checks to see if the device designated is a television. If a television is designated, control is transferred to subroutine TV; otherwise, section ACT is executed. Section ACT activates the designated device by turning annunciator 1 on, then off, which causes the switch in the ECU to change state (i.e., from on to off or vice versa).

Subroutine TV stores the contents of the accumulator, which indicates which television is being controlled, into the variable YTV. The ECU is used to control two televisions. Each television has a different number of channels. The correct value associated with the designated television tuner is loaded into the variable NCHAN. This value is 13 for one television and 18 for the other. The subroutine next loads a secondary vocabulary, by calling the subroutine VOCSW2, which contains the voice signatures for the television channels. Subroutine WHCHAN is called to change the channel on the television by voice input. Subroutine WHCHAN begins by calling the subroutine MENU2. Subroutine MENU2 displays the entries in the cue buffer of the speech recognition software. These entries are the phrases associated with the television channels. The program then displays the question "WHICH CHANNEL?" and waits for voice input. If the voice input is a channel then control is returned to the TV subroutine with the value of the recognized channel stored in the accumulator. If the voice input is not a channel then the subroutine WHCHAN is repeated. If the voice input is not a valid input the program loops back to call WHCHAN again. If the voice input is valid, the number of channels to be changed is calculated by subtracting the current television channel from the desired television channel. If the result is negative, the program adds NCHAN to the result because the tuners are cyclic, then the new channel is stored. The channel is changed by turning annunciator 1 on and off the calculated number of times. The primary vocabulary is then switched with the secondary vocabulary via VOCSW2, and control is transferred to the primary menu.

D. Utility Subroutines

Vocabulary Switch

This subroutine switches the primary vocabulary pointed to by VOCAB, in the speech recognition software, with a secondary vocabulary pointed to by BUFF. The subroutine was written in relocatable code so that another portion of memory could be set aside for a different secondary vocabulary by assembling the code at another address. Figure 7 is a flowchart of this subroutine.

Voicein Subroutine

This subroutine, which is summarized by the flowchart in Figure 8, begins by calling the clock display subroutine in the speech reocgnition software, which places the correct time in the lower right hand corner of the display. The program then checks to see if a key has been pressed. If a key is pressed then the value of the key is loaded into the accumulator, and control is returned to the program that called this subroutine.

If a key has not been pressed, then the subroutine checks to see if the time scheduled portion of the speech recognition software is enabled. The time schedule portion of the speech recognition software allows the user to activate EWS devices at specified times. If the time schedule portion is enabled then the subroutine to handle the time schedule is called. Upon return from the time schedule portion the program checks to see if any action was performed. If an action was performed then control is transferred to the program that called this subroutine.

If the time schedule portion was not enabled, or if no action was performed in the time schedule portion, the program checks to see if the voice input portion is enabled. If the voice input is enabled then the program calls the voice input subroutine in the speech recognition software; otherwise, the VOICEIN subroutine repeats. If there is no voice input upon return from the voice

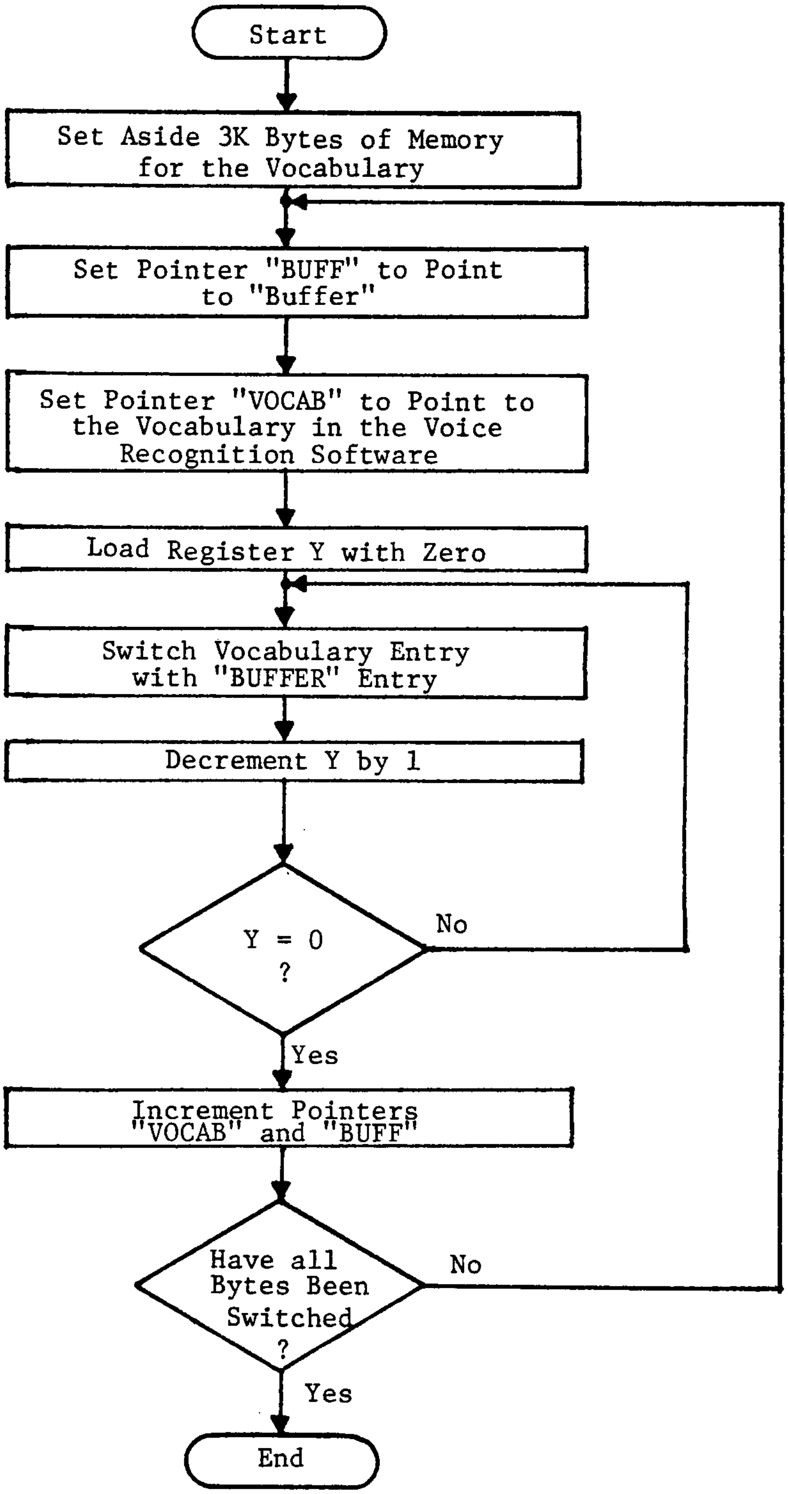

Figure 7. Flowchart of Vocabulary Switch Subroutine.

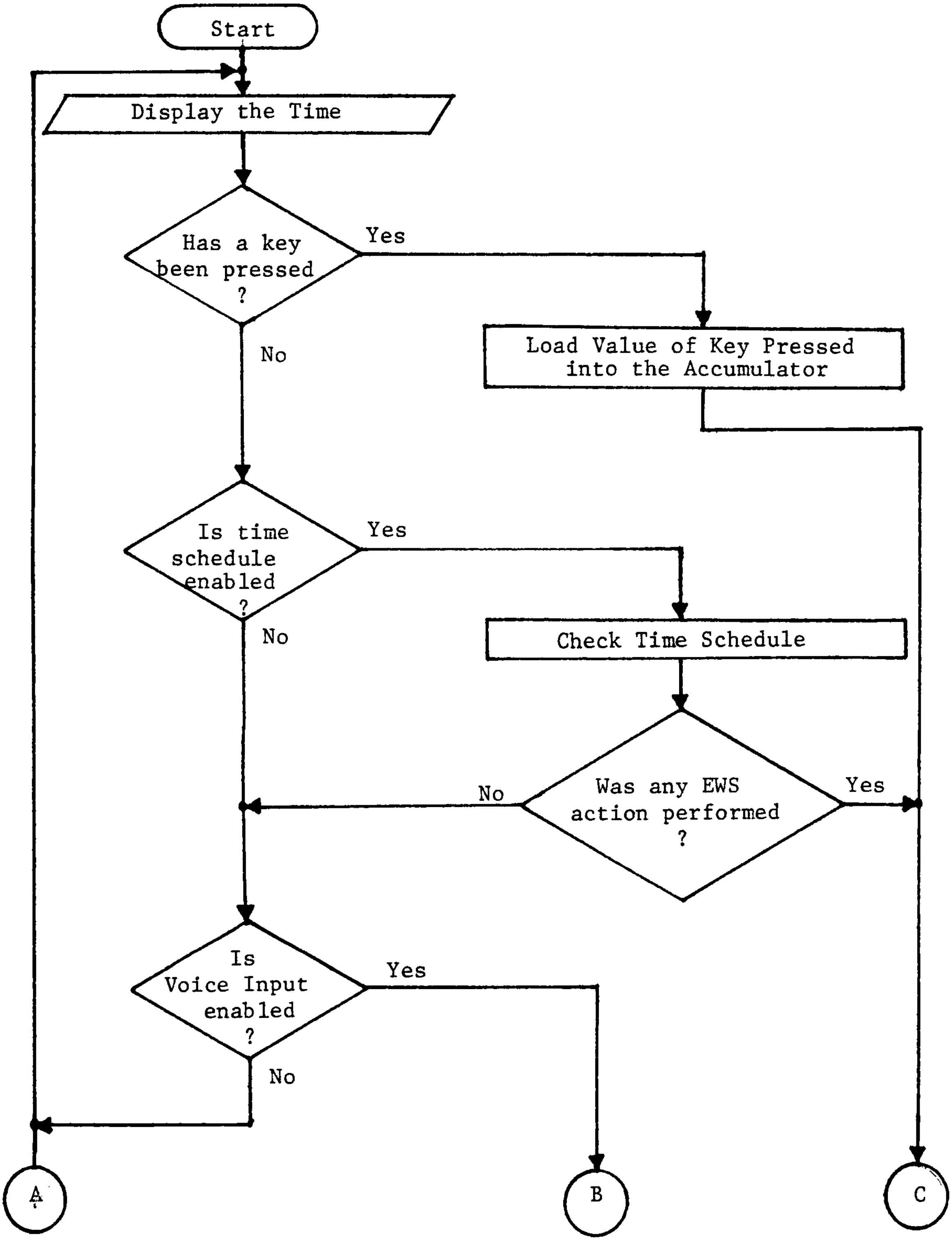

Figure 8. Flowchart of VOICEIN Subroutine.

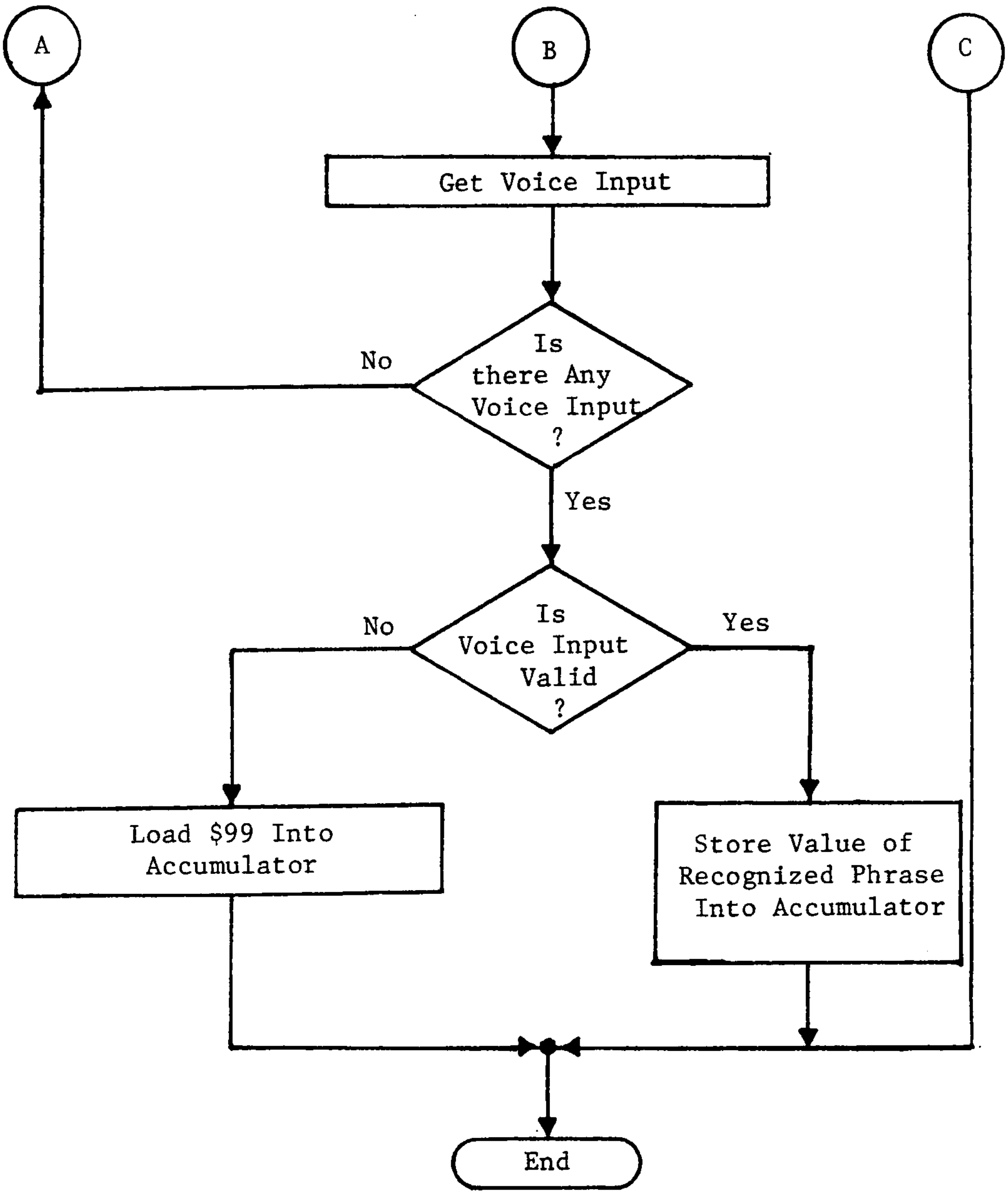

Figure 8. (Continued)

input subroutine, then the VOICEIN subroutine also repeats. If there is voice input upon return from the voice input subroutine the validity of the input is checked. If the voice input is valid as determined by the VIM software (i.e., not erroneous, or the score was acceptable) then the value of the recognized phrase is loaded into the accumulator, and control is transferred back to the calling program. If the voice input is invalid, then the value of 99 (hexidecimal) is loaded into the accumulator, and control is returned to the program which called this subroutine.

CHAPTER V

RESULTS AND RECOMMENDATIONS FOR FURTHER STUDY

A. Results

This study was conducted to ascertain whether or not voice input to a microcomputer as an aid to the handicapped for environmental control was viable. The goals were to provide (1) voice input, (2) environmental control and (3) telephone control. These goals were set so that the handicapped population would profit from the investigation.

Voice input was accomplished by a commercially available voice input module which was designed to interface with a popular microcomputer. Environmental control was effectively achieved through two different types of environmental control systems, one was an existing wiring system, and the other was an environmental control unit. An interface between the microcomputer and a commercially available speaker telephone provided Telephone control. The microcomputer was used to control the VIM, the environmental control systems and the telephone interface.

The ultimate goal of home environmental control by voice input to a microcomputer was found to be a viable and useful aid to the handicapped. By developing software, designing hardware and using some commercially available equipment this study indicates home environmental control can be developed to control, among other things (1) the lighting, (2) the telephone, (3) a television, (4) an intercom, (5) a radio and (6) an eight-track tape player. These items are only a few of the devices that can be controlled with this system. In general, this system can control most devices which are plugged into a standard wall socket.

By designing the Annunciator Interfacing Device (AID), the microcomputer was able to control the Environmental Control Unit (ECU) and a speaker telephone. The AID provided control of the ECU and telephone by parallelling switch closures. Software in the microcomputer produced the pulse dialing technique for the telephone.

The VIM obtained for this study had speech recognition software provided with it. The VIM software was used to initialize the VIM for use with the software written in this study for envrionmental control. When the system is "booted up" the VIM software is executed. The VIM software allows a vocabulary of desired voice input to be created and provides speech recognition of the voice input and an interface to the existing wiring control system. When the VIM software is put into the speech recognition mode it waits for voice input. When the voice input recognized is a word or phrase indicating home, environmental or telephone control, the software written for this study is executed.

The software which was written consists of the Home Control, Telephone Control and ECU Control subroutines. The Home Control subroutine utilizes the VIM's ability to interface with the EWS, while the Telephone Control and ECU Control subroutines utilize the AID's ability to interface with the speaker telephone and ECU.

A utility subroutine was written so that the speech recognition ability of the VIM's software could be used in the environmental and telephone control subroutines. Another utility subroutine was written which increased the number of voice input phrases the VIM could recognize.

The ECU is controlled by a scanner which is controlled by two switch inputs that allows the user to select and activate a device. One switch closure selects the device, and the other switch closure activates the device. The AID was used to parallel these two switch closures. In the ECU Control program, the method of determining the number of positions to move the ECU's scanner resulted from the consideration of several approaches. One method was based on voice input of the number of positions to move the scanner. This method had its merits. For example, devices could be interchanged, and the voice input phrases describing them would not have to be changed. However, this was unsatisfactory by virtue of the fact that the individual had to determine the number of scanner positions to move and then input the phrase associated with this number.

The chosen method had the computer calculate the number of scanner positions to move. This was accomplished through input of the present scanner position during the initialization of the program. The program calculates the number of positions to move by subtracting the desired position from the current position. If the result is negative the program adds 14 to the number since scanning is cyclic. One problem arises with this method, i.e., the scanner position can be changed by input independent of the computer. When this occurs the position stored in the computer is in error. This problem is solved by letting the individual reset the present scanner position by inputting, through voice input, the phrase "ENVIRONMENTAL CONTROL UNIT" which causes the program to repeat the portion of the program where the scanner position and television channels are initialized. The same method used to change the scanner positions was also used to change television channels. The hospital-type television channel selectors are also cyclic.

In the Telephone Control program the AID was used to parallel a switch closure on a speaker telephone. This switch closure controls the on-or-off hook status and dialing process of the telephone. The program allows a directory of 48 names and associated phone numbers, entered by the keyboard, to be stored, and it also allows selections from this directory to be made by voice input. The program also provides the ability to input a phone number by voice in case the desired phone number is not contained in the directory. However, the voice input phone number is not stored in the directory. A speaker telephone was used because it eliminates the requirement of the handicapped person to negotiate a standard telephone handle.

The Home Control subroutine controls lights and appliances through the interface provided with the VIM. This program is menu driven so as to take full advantage of the limited vocabulary aspect of the VIM software.

Voice input is an aid to the handicapped, however, when using voice input there are a couple of assumptions made. One assumption is that there is no, or very little, background noise. Background noise interferes with the voice input and can cause frustration to the user by making him wait until there is only an acceptable amount of it. Another assumption is that the handicapped individual can speak and can do so in a repeatable fashion. However, distinctively uttered sounds could be sufficient for input.

There are a few disadvantages to the system presented in this study, but most can be handled by investing a little more time and money. One of the disadvantages of this system is its lack of recognition of monosyllabic words. Of the digits zero through nine only 7 (sev-en) and 0 (ze-ro) are polysyllabic. Therefore, single digit number entry for selecting television channels and inputting phone numbers to be dialed is unavailable. This would require a different algorithm for speech recognition. To overcome this deficiency phrases were used in place of the numbers and alphabet.

Another disadvantage of this sytem is the microphone which is used for voice input. The microphone restricts mobility. The system could be enhanced by using a remote microphone which could be attached, in a comfortable fashion, to the disabled individual.

The cost of most of the VIMs which can be interfaced to a microcomputer starts at about $1,000.00, and does not include EWS interfacing. The computer interfaced EWS device costs begin at about $250.00. Therefore, a system with both voice input and EWS control would have a minimum cost of about $1,250.00. The cost of the VIM used in this study was less than half this value and includes the EWS interface. This system is considered to be effective in spite of some disadvantages. The price/performance ratio of the VIM outweighs its disadvantages. The fact that this module provides voice input and EWS control, is very attractive for this system.

B. Recommendations for Further Study

One area which should be considered is programming the microprocessor for interrupt based control. With interrupt processing the microcomputer could be used for other applications or entertainment, and when a specific phrase or sound is detected the programs for environmental control would be loaded. Also, the telephone could be connected to a coupler which detects the presence of an incoming call and then issues an interrupt so that the telephone control program is loaded.

Another area of application would use some of the same hardware presented in this study, but a different VIM. Some of the VIMs have their own read/write memory which takes some of the load off the microcomputer's memory. Many of the VIMs also can store larger vocabularies which means a secondary vocabulary would not have to be set aside for special functions.

Still another improvement could be made in using a device which produces all 256 of the codes for EWS control. This device is directly connected between the microcomputer and the AC powerline and transmits all the codes for the control of the separate modules.

Any of the systems studied can be enhanced by the use of a remote, cordless microphone. The system could be used to drive a motor which opens and closes a door and would allow the handicapped person to come and go as necessary. They could also operate lights and appliances from anywhere inside and outside their home within the transmission radius of the remote microphone.

Another application, aside from aid to the handicapped, concerns the use of a computerized phone directory in conjunction with the telephone control program. For example, in fields which require extensive phone calling such as in sales, software could be developed to provide the ability to dial a number, wait for an answer and output a message, possibly by voice synthesis. By using a sophisticated coupler the program could also detect a busy signal or no answer, make a note of it, and then call back at a later time.

All in all, the fields of speech recognition and environmental control present potential aids to the handicapped which can greatly enhance their day-to-day lives by providing methods to perform so-called routine tasks.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Ciarcia, Steve, "Use Voiceprints to Analyze Speech," BYTE, Vol. 7, No. 3, p. 50.

2. Bridges, Alan, "Speech Recognition Systems Emerge for Untapped Market," Computer Technology Review, Winter 1983, p. 91.

3. Lea, Wayne A., ed., Trends in Speech Recognition, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1980.

4. Shoup, June E., "Phonological Aspects of Speech Recognition," Trends in Speech Recognition, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1980.

5. Zue, Victor W., and Richard M. Schwartz, "Acoustic Processing and Phonetic Analysis," Trends in Speech Recognition, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1980.

6. Conrad, Ann E., "Toss Your Keyboards and Just Tell Your Computer What to Do," Research and Development, January 1984, p. 86.

7. Kahdeman, Jan E., "Application of Automatic Speech Recognition to X-ray Microanalysis," American Laboratory, September 1983, p. 110.

8. Bridle, J. S., M. D. Brown, and R. M. Chamberlain, "Continuous Connected Word Recognition Using Whole Word Templates," Joint Speech Research Unit, Research Report No. 1013, March 1983.

9. Klugman, Dietrich, "Talking to the Computer: The Principles and Applications of Automatic Speech Recognition," Siemens Review, XLVII (1980) No. 4, p. 16.

10. Waldo Operating Manual, Arlington, VA, ARTRA, Inc., 1983.

11. Apple IIe Reference Manual, Cupertino, CA, Apple Computer, Inc., 1982.

12. Faflick, Philip, "Power to the Disabled: Micros Help the Handicapped Help Themselves," TIME, December 13, 1982, p. 76.

13. Ciarcia, Steve, "Computerize a Home," BYTE, Vol. 5, No. 1, p. 28.

14. Kuecken, John A., Talking Computers and Telecommunications, New York, NY, Van Nostrand Reinhold Company, Inc., 1983.

VITA

Keith Marlin Anspach was born in Manchester, Tennessee on November 22, 1959. He attended elementary schools in both Tullahoma, Tennessee and New Orleans, Louisiana; Junior High Schools in Gallatin and Tullahoma, Tennessee, and he graduated from Tullahoma High School on June 6, 1977. In August of 1977 he was injured in an automobile accident in which he suffered a broken neck and severed spinal cord leaving him a total quadriplegic, thus postponing his entrance to college until September 1978.

He graduated from Motlow State Community College in June 1980 with an Associate of Science degree in Mathematics. In August 1980 he entered Middle Tennessee State University and graduated in May 1982 with a Bachelor of Science degree with a major in Computer Science and minors in Mathematics and Physics.

In September 1982 he enrolled in the Graduate School at the University of Tennessee Space Institute and studied for a Master's degree. During his studies at the Institute, he received the B. H. Goethert Scholarship, and in August 1984, he received the Master of Science degree with a major in Computer Science.