AD-A051 543 INPUT OUTPUT COMPUTER SERVICES INC CAMBRIDGE MASS F/G 9/2
SINGLE-CHANNEL VOICE-RESPONSE-SYSTEM PROGRAM DOCUMENTATION. VOL--ETC(U)
DEC 77 DOT-TSC-1107-1

UNCLASSIFIED FAA-RD-77-177-1 NL

1 OF 1
AD
A051543

END
DATE
FILMED
4-78
DDC

REPORT NO. FAA-RD-77-177, I

SINGLE-CHANNEL VOICE-RESPONSE-SYSTEM PROGRAM DOCUMENTATION

13

Volume I: System Description

Input Output Computer Services, Inc.
689 Concord Avenue
Cambridge MA 02138

DECEMBER 1977

FINAL REPORT

DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research and Development Service
Washington DC 20591

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

19 77-177-1, FAA-77-24.1

18 FAA-RD, TSC

Technical Report Documentation Page

1. Report No. FAA-RD-77-177, I	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle 6 SINGLE-CHANNEL VOICE-RESPONSE-SYSTEM PROGRAM DOCUMENTATION. Volume I. System Description.		5. Report Date 11 Dec[redacted] 1977
		6. Performing Organization Code
7. Author(s)		8. Performing Organization Report No. DOT-TSC-FAA-77-24,I
9. Performing Organization Name and Address Input Output Computer Services, Inc.* 689 Concord Avenue Cambridge MA 02138		10. Work Unit No. (TRAIS) FA 831/R8109
		11. Contract or Grant No. 15 DOT-TSC-1107-1
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington DC 20591		13. Type of Report and Period Covered 9 Final Report. Sep. 1975-Jan. 1976.
		14. Sponsoring Agency Code
15. Supplementary Notes *Under contract to: U.S. Department of Transportation Transportation Systems Center Kendall Square Cambridge MA 02142		

1/2 AD51 745

16. Abstract

This report documents the design and implementation of a Voice Response System (VRS) using Adaptive Differential Pulse Code Modulation (ADPCM) voice coding. Implemented on a Digital Equipment Corporation PDP-11/20,^{R} this VRS system supports a single audio-output channel. Vocabulary size is limited to 900 words or phrases. Input to the system consists of text messages or sentences in ASCII format transmitted to the 11/20 through a 300-baud asynchronous interface. A preliminary design for a VRS for 10 channels is reported.

This is the first of three volumes. Volume II describes the program-design modules, and Volume III is a user's guide to the system.

17. Key Words Voice Response System VRS ADPCM Speech Coding	18. Distribution Statement DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 12 39p.	22. Price

Reproduction of completed page authorized

409553

TC

PREFACE

The developmental work summarized in this final report was carried out by Input/Output Computer Services, Inc., under contract to the United States Department of Transportation, Transportation Systems Center (DOT/TSC). The research was sponsored by the Federal Aviation Administration (FAA) and represents the first phase in the Flight Service Station (FSS) Automation Program to provide preflight weather briefings to the aviation community via computer generated Voice Response. The system described in this report provides for a single-channel Voice Response System (VRS) to be used in evaluating Adaptive Differential Pulse Code Modulation (ADPCM) speech compression techniques and the man/machine communications interface for a real-time pilot self-briefing system.

The work reported here was completed under the direction of the TSC Program Manager, Paul D. Abramson, the Technical Monitors, John Sigona and Bruce E. Ressler, and by John F. Canniff. Carey Weigel of the FAA provided overall program guidance.

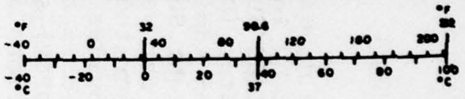
METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



CONTENTS

ILLUSTRATIONS

1. INTRODUCTION

This report covers the design and implementation of an experimental single-channel voice response system (VRS) to be used in evaluating Adaptive Differential Pulse Code Modulation speech compression techniques for a real-time pilot self-briefing system.* The report consists of separate volumes as follows:

Volume I:	Final Report, consisting of system design of a single-channel system, discussion of problem areas and their solution, and a preliminary design of a ten-channel VRS system using these concepts.
Volume II:	System Documentation, consisting of narrative on specific program modules, and flow charts.
Volume III:	User Manual for the single-channel system.

* A Multiline Computer Voice Response System Utilizing ADPCM Coded Speech. Rosenthal, et al. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-22, No. 5, October, 1974.

2. BACKGROUND

Flight Service Stations have evolved over the years in response to the increased demands of a rapidly growing fleet of aircraft. Recent analysis has shown that a four-fold increase in demand can be expected by the mid 1990's. The increase will require three times the present number of specialists resulting in a cost prohibitive labor-intensive system. An automated system using unattended pilot self-briefing terminals or pilot access to the system directly from the user's home has been conceived and initially tested. This automated system, when implemented, will be capable of providing improved services, accommodate the increased demands and be cost effective when implemented.

The initial testing of the automated concept was accomplished using a system built around the GE TimeShare network supplemented by a minicomputer using a similar data base and a leased single-channel computer generated Voice Response System (VRS). The technical feasibility of the concept was clearly demonstrated, however significant system design questions remain to be investigated before specifications for final system development implementation can be developed.

A system prototype is planned which will be used to investigate these system design questions. The prototype will consist of a dedicated computer for weather data formatting and retrieval. The data will be accessed from various types of unattended self-briefing terminals and remote terminals. A critical component of the system for mass weather dissemination and remote terminal access is the VRS. The VRS vocabulary requirements must be investigated further prior to complete user acceptance and the number of simultaneous users for the prototype system must be increased to a minimum of ten.

3. STATEMENT OF PROBLEM

Design and develop a Voice Response System (VRS) that will have a capability of one channel operation expandable to simultaneous ten channel operation. This VRS must have the capability for vocabulary development and experimentation i.e., the capability to digitize, store and recall with good fidelity the voice either as single words or in phrases.

The immediate requirement is to develop a single-channel VRS on a PDP-11/20 computer currently available. The PDP-11/20 will be driven by a H-516 computer which will act as a message handler and decoder. The PDP-11/20 will contain the voice response programs, and will access the vocabulary disk which will also contain the operating system and programs.

It is a goal that this interim system will be designed to be easily expanded and modified into a multi-channel system running under a multi-tasking monitor. This effort is to develop the single channel VRS and consists of the following tasks:

3.1 MODIFICATION OF OPERATING SYSTEM

a. DEC RT-11 Operating System, Single User.

b. Addition of drivers for A to D and D to A converters.

c. Interface to H-516.

3.2 DATA BASE GENERATION (1000 WORD OR PHRASE VOCABULARY)

a. Optimum file structure for vocabulary, Hierarchy, File Indexes, etc.

b. Development of word editing (Start and Stop codes).

c. File Updating (Word Addition, Deletion and change).

3.3 OUTPUT SPEECH GENERATION

a. Sentence builder.

b. Buffer loading strategy.

3.4 H-516 PROTOCOL

a. Message decoder.

b. Transmission.

3.5 DEVELOPMENT OF SOFTWARE

Development of ADPCM encoding and decoding software for a single channel system.

3.6 DEFINITION OF REQUIREMENTS

Define the requirements to implement the multi-channel VRS including the specifications of the necessary computer peripherals to expand the data base vocabulary from 1000 words or phrases to 4000 words or phrases.

4. DESIGN EFFORT

The design effort addressed the following tasks:

An efficient file structure for the digitized voice files and dictionary that relates identifiers to voice file addresses.

A flexible file editing scheme which permits easy user access to a given entry for modification or deletion.

Meeting a severe timing constraint which must be satisfied to prevent unwanted gaps in the output speech.

Implementation of the ADPCM approach in PDP-11 software, since a hardware speech decoder was not available.

The design constraints included the following:

Implementation on a PDP-11/20 computer.

Speech encoding and decoding using ADPCM software.

The first constraint is one of memory. Since the maximum usable memory address space is 28K words, the programs and dictionary (which need to be core-based for fast access) must fit within this space while yielding the maximum permissible vocabulary size. The second constraint creates timing problems since the entire processor bandwidth is used in decoding speech samples in real-time.

The following design goals were desirable:

As large a vocabulary as possible, given the constraints of a single, moving head disk.

Maximum quality speech within the constraints of 3 kHz cut-off. No gaps due to processor imposed delays.

Automatic editing. Close control of utterance end points without specific user manual intervention.

Parsing of text input strings to provide variable length pauses when identifying punctuation marks.

Proper speaking of numbers when digit strings are input.

These design goals and constraints highlighted problems in the design of the file structures and data base, capabilities of a single, moving head disk in servicing the requests for new speech buffers, and design of an efficient algorithm for decoding ADPCM compressed speech in the time allotted between speech samples. The discussion of these problems and their method of solution follows.

4.1 PROBLEM IDENTIFICATION

4.1.1 File Structures

The design of a suitable file structure having attributes of quick access, easy editing and minimum storage of auxiliary header information, was necessary.

A minimal file system is presented and some properties of the system are examined. This file system is then upgraded to what may be called a minimum "workable" system. The resultant problems involved are discussed. Finally the requirements not met by the minimum "workable" system are summarized and a final design is proposed to solve these problems.

A simple file system is shown in Figure 4-1. This contains a tag, or name for a file, a pointer to the file's location on disk and the actual files themselves. In this system, an entry is looked up by searching the entire file of tags. A comparison would begin at each point where a file name began and would continue until a match was found. Lookup could be speeded up by a partial sort which would permit better "guessing" of where in the directory to start looking.

The immediate drawback of this file system is the lack of size information in the file. A file, which is usually 4 blocks long, will seldom end on an even block boundary. In any event, to determine the endpoint of a file, each entry in the file must be examined until an end-of-record indicator is found. This is an unreasonable software requirement which is easily avoided at little expense.

Figure 4-2 presents an addition to a directory entry in order to simplify this access. The new entry contains two additional bytes of information. The first is a count of the length in 256 word blocks, and the second indicates the last block in the file.

The directory in this second file system must be sorted to expect real time response. This can be demonstrated quite easily by examining look-up times for a linear search. On the average, a linear search requires 2000 comparisons to find an entry in a 4000 word directory. An average entry, based on an examination of a sample vocabulary, is 5.8 characters long. Including looping, branching and returning, an average compare takes 50-100 usec., or 100-200

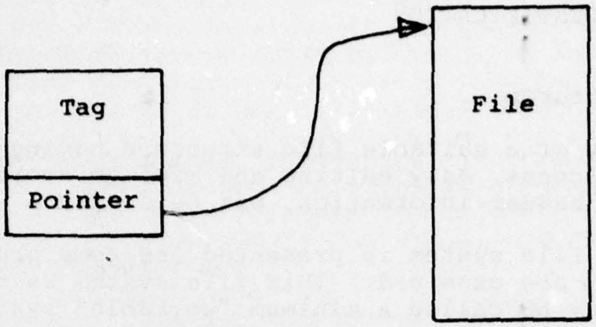


Figure 4-1. SIMPLE FILE SYSTEM

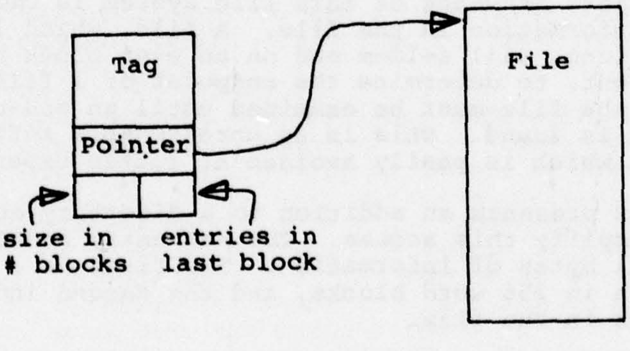


FIGURE 4-2. SIMPLE FILE WITH ADDITIONAL DESCRIPTORS

milliseconds for the entire file search. This 100-200 millisecond look-up time permits access to a file 1000 words or 4000 samples long which corresponds to 2/3 of a second of speech. This time is not prohibitive for the single channel system. Since simple extension to a 10-channel system is a design goal, we must examine the effects of a 100 millisecond lookup delay extended over each of the ten channels. It is then apparent that a one second penalty could be imposed for looking up two/three seconds of speech. Therefore, for the single channel, a more complicated but vastly more efficient binary search algorithm was developed. It can be directly applied to the 10-channel system.

A sorted dictionary permits the binary search algorithm to be used. The binary search requires only $\log_2$ comparisons or 12 for the case of the 4000 word dictionary. The comparisons would take somewhat longer (20% to include time to find the beginning of the entry nearest to the point which the binary search entered the directory). However, the total time for a look-up of 10 channels is on the order of 10-20 milliseconds for every 600 milliseconds of speech produced which provides more than enough time to perform many other functions.

The directory requires efficient organization in order to maintain core residency. For instance, using the very simple directory shown in Figure 4-2, a vocabulary of 4000 words requires a tag size of 3.4 words (4.8 characters + 1 character terminators in ASCII) average, and two descriptor words per entry or 21,600 words. Even assuming radix 50 text, this size only decreases to 18,800 words. Assuming roughly 1000 words of core buffers per channel, this brings the memory consumed to 28K words, giving 4K of memory for the entire operating software, assuming the full 32K address space available per user in a mapped PDP-11/70 system. In the 11/20 system, the directory consumes all available 28K of memory.

4.1.2 Disk Performance

For the single-channel design, the use of a moving head disk does not present a problem. For the expected 10-channel system, a guide analysis demonstrates the inadequacy of a single moving head disk. Briefly, the time available to fill 10 core buffers (one for each channel) is the time required to "speak" the contents of one buffer. One buffer is spoken in 170 ms. This figure is further reduced by the fact that files do not take an integral number of blocks; rather the last block is, on the average, half empty. Based on the average file length, every fourth block is half-empty requiring an extra buffer to be brought in for every four of the above mentioned 10 core buffers. This requires an average of 12.5 disk reads every 170 ms.

The 170 ms. corresponds to only four revolutions of the RK-11 disk. The file system, which would require up to four such disks, would increase the effective speed somewhat, but a disk scheduler would still be hard pressed to operate at this speed.

The single user experimental system can function reasonably with a single, moving head disk. It can be used to determine the performance to be expected from the above schemes. The above can be done by such things as keeping use counts on various voice files to determine frequency of use.

4.1.3 ADPCM Algorithms

The ADPCM algorithm used to encode and store the digitized and sampled speech was chosen because it permits a compact storage mode (2-fold improvement over comparable PCM technique). It also permits a simple auto-editing technique for identifying utterances from the intervening silence. In the single channel system, both the encode and decode algorithms are implemented in software. It should be noted that the encode portion operates quite slowly, about 10 times real-time, whereas the decode portion, which delivers speech samples to the D/A connector, must obviously operate in real time.

The ADPCM algorithm creates 4-bit code words representing the 10-bit PCM speech samples. The algorithms preserve resolution at low energy signal levels by decreasing the step size via an internally generated parameter, "L". During encoding (see Fig. 4.3), a predicted value (Y) is subtracted from the current PCM sample to form a difference (delta). A code word for this sample is generated by finding via iteration the index into a table of step sizes (ENCTAB) such that the actual step size is just greater than the predicted step size in ENCTAB. The initial approximation is based on the previous value of L, and is used as the initial table entry.

The code word, therefore, represents the "correction" required to match the current sample with the predicted value, based on the "old" value of L. A new value of L is derived from the new code word via a table look-up, and a new predicted speech sample, Y, is computed for the next encode cycle.

During decode (the "speaking" phase), the current code word is combined with the previous index value determined by L to form a new index to the step size table. The step size, delta, is extracted from the table and added to the previous output, Y, to form a new Y-value, which is sent to the D/A convertor. Finally, a new L-value is looked up using the current index value into the L-table. The process as implemented in the single-channel software is shown in Figure 4-4.

The initial encoding of utterance strings by the above algorithm requires an editing phase to determine end-point locations for the individual utterances. Only the code words corresponding to the utterances themselves are stored and catalogued with their corresponding dictionary entries. The ADPCM editing feature takes advantage of a sharp transition in code word energy (as opposed to speech energy) between silence and utterance (Ref. 1, p. 343). This imposes a severe (but tolerable) signal to noise ratio requirement in the input-utterance string, since excess background noise will make the end point detection process concatenate several utterances together, or drop-low energy syllables.*

At this point, some timings must be examined. A look at the PDP-11/20 cycle times indicates that the decode time for a 4-bit code word is 60 to 100 microseconds. Add to this the disk I/O cycle stealing, directory look-ups, etc. and it is clear that CPU bandwidth is just not available. The single-channel 11/20 system will be able to keep up, provided all operations are done serially; that is, directory look-up first, composition of sentences, followed by playback.

* Rosenthal, <u>loc. cit.</u>, p. 341-345.

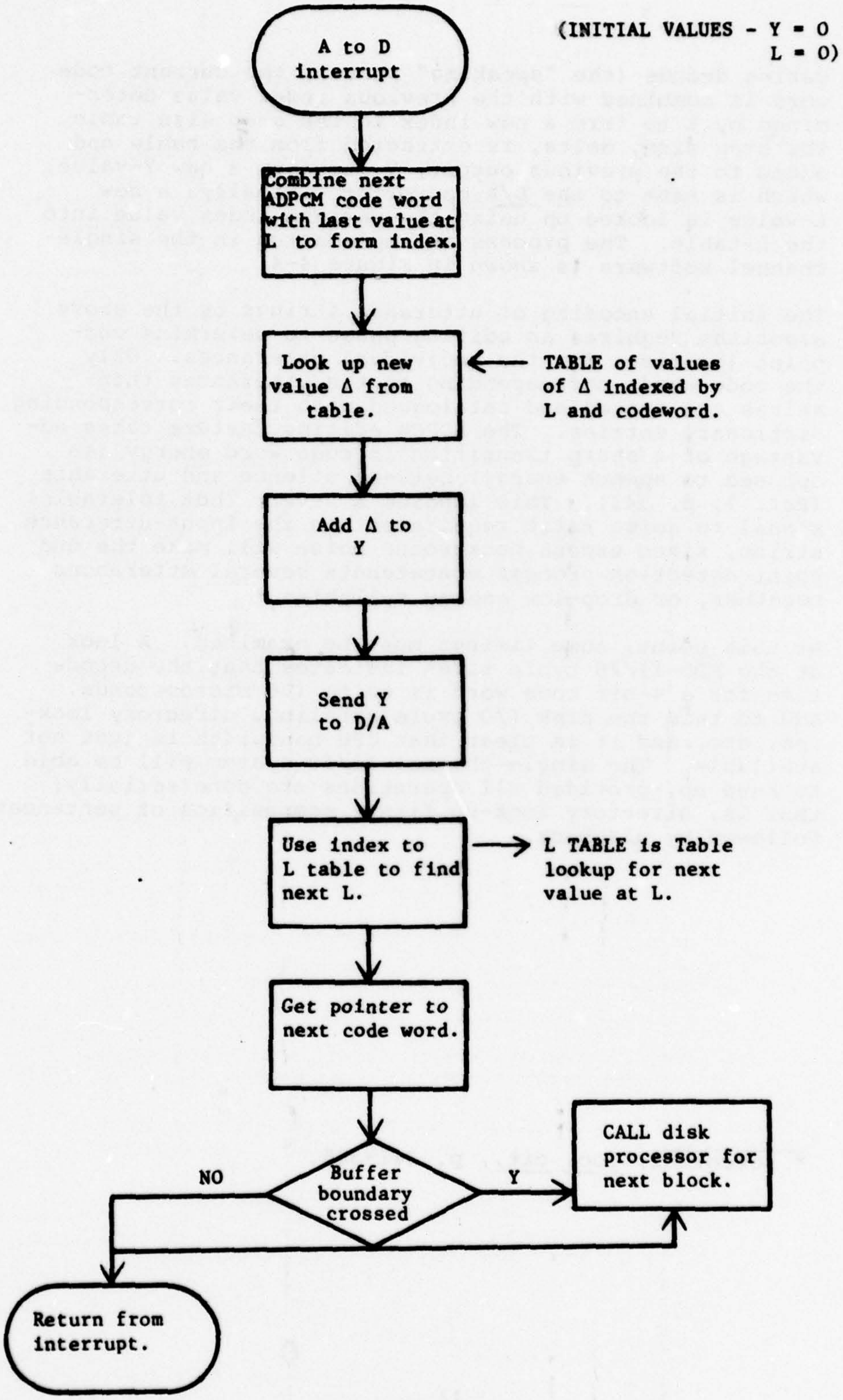
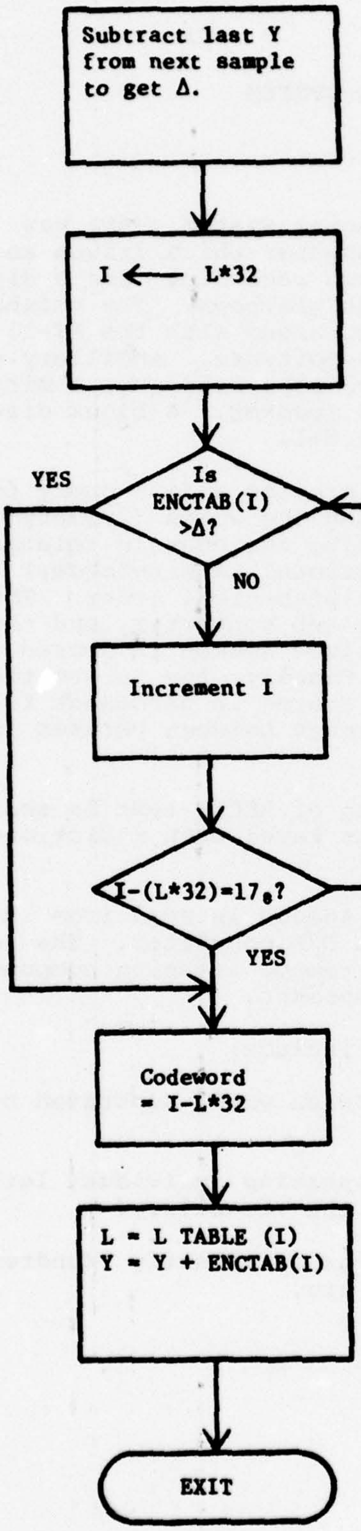

(INITIAL VALUES - Y = 0
L = 0)
A to D interrupt
Combine next ADPCM code word with last value at L to form index.
Look up new value Δ from table.
TABLE of values of Δ indexed by L and codeword.
Add Δ to Y
Send Y to D/A
Use index to L table to find next L.
L TABLE is Table lookup for next value at L.
Get pointer to next code word.
Buffer boundary crossed
NO
Y
CALL disk processor for next block.
Return from interrupt.

FIGURE 4-3 ADPCM ENCODE SERVICE ROUTINE



*(INITIAL VALUES - Y = 0
L = 0)*

ENCTAB is an array of values of Δ's indexed by L and code word.

L TABLE is table lookup for next value of L.

FIGURE 4-4 ADPCM DECODE

5. IMPLEMENTATION OF SINGLE-CHANNEL SYSTEM

5.1 GENERAL DESCRIPTION

The single channel voice response system (VRS) was implemented on a PDP-11/20 computer which drives an analog to digital convertor for speech recording, and a digital to analog convertor for speech playback. The vocabulary is stored on a RK03 disk pack along with the RT-11 operating system, and the VRS software. Ancillary equipment includes a cassette audio tape recorder, a microphone, an audio amplifier and a loud speaker. A block diagram of the system is shown in Figure 5-1.

In operation, the user first creates a dictionary for the required vocabulary by entering the words (primary definition and synonyms, if any) using the console teletype. Next, he records (on a conventional tape recorder) the required words or phrases, in alphabetical order. Then, the audio tape is fed through the A/D converter, and the resulting digitized and quantitized speech is stored in a temporary file on the disk. Finally, the information is packed into ADPCM format and stored in permanent files keyed to the dictionary. Silence between phrases is automatically edited out.

During voice response a string of ASCII text is scanned and each phrase in the text is keyed with a dictionary element.

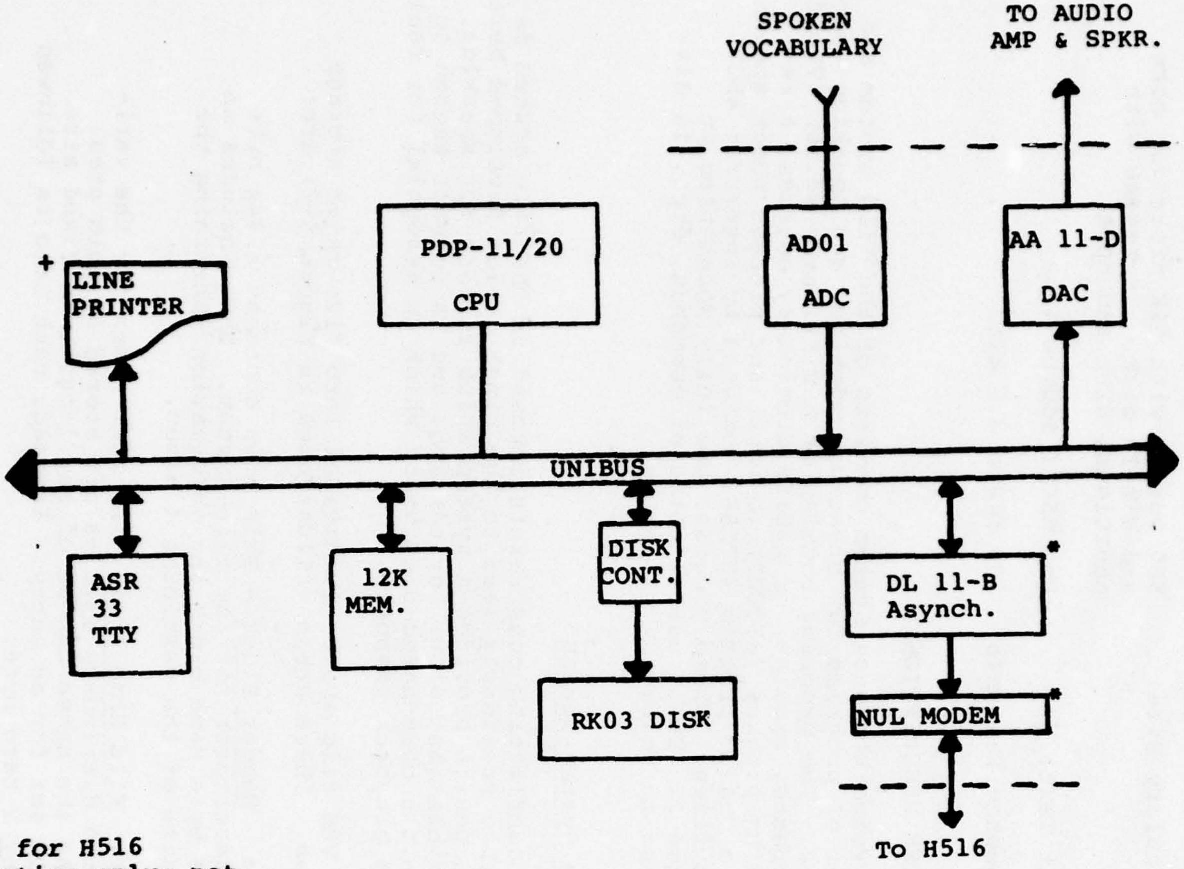
The associated ADPCM-encoded speech is read from the disk, decoded, and passed through a D/A converter. The resulting audio output is filtered (to remove aliasing components), amplified, and fed to a loudspeaker.

Additional playback features include:

Insertion of pauses between words separated by commas.

Automatic "spelling" (speaking individual letters) of text strings not in the vocabulary.

Parsing of number strings to identify "Hundred," "Thousand," "Million," etc.



* Required for H516 communication only; not in development system.

+ Required for program development only.

FIGURE 5-1. SINGLE-CHANNEL EXPERIMENTAL VRS

5.2 SPECIFICATIONS

Vocabulary Size:	500 words, with 12K processor core and single disk. Increases with additional disk and core.
Input Text:	Any ASCII source file.
Frequency Response:	100 Hz. to 3.0 kHz.

5.3 SYSTEM DESCRIPTION

The Voice Response System consists of the file system and a set of programs for creating, editing, and speaking voice files. The programs consist of a dictionary editor (VEDIT) to create, examine, or modify dictionary entries, a recording program (RECORD) to input and process human speech, and a "talk" program (SPEAK) designed to interpret ASCII text files. These programs are fully documented in Volume 2. Their use, listing of commands, etc., is discussed in Volume 3.

5.3.1 File System Design

The conflicting constraints imposed on the file system described previously lead to the final design developed here. This design provides a system which is fast for speaking, flexible (but slow) for editing, and is compact enough to permit a core-based dictionary which is essential for real-time playback response.

The VRS file system is divided into five major storage areas. These areas (illustrated in Figure 5-2) are:

a. Header Block - This area contains a 34_8 byte descriptor for the file system. This includes an 8 byte name block and information concerning the size of the remaining 4 areas.

b. File Name Area - The text names for the various dictionary entries are stored in this area. As the names can be of any length, no fixed size is set for an entry. Instead, each name is followed by a zero byte.

c. File Descriptor Area - The size and location of each dictionary entry is stored in this area.

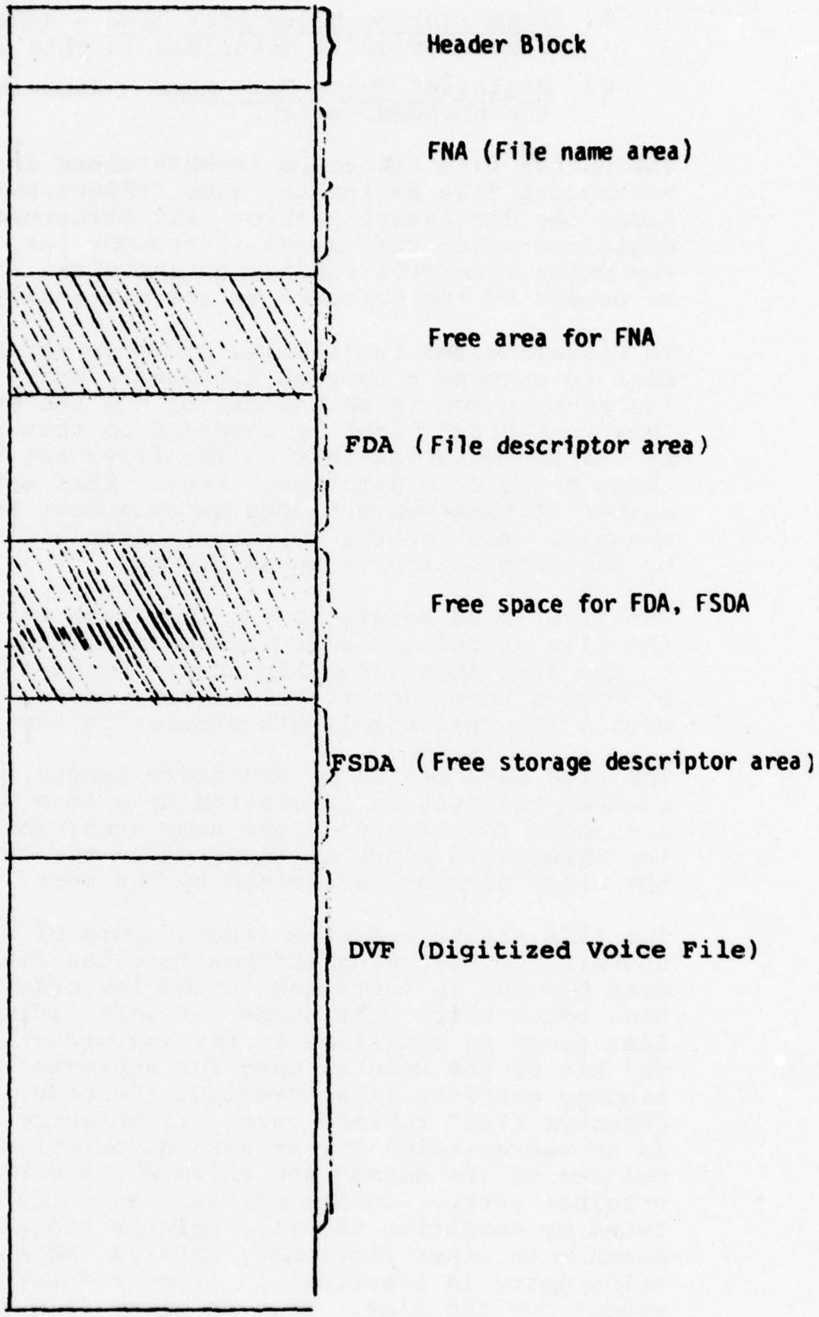

Header Block
FNA (File name area)
Free area for FNA
FDA (File descriptor area)
Free space for FDA, FSDA
FSDA (Free storage descriptor area)
DVF (Digitized Voice File)

FIGURE 5-2. SUMMARY OF THE PARTITION OF DISK FILE SYSTEM, "DIRECT. DVF"

d. Free Storage Descriptor Area - Unused disk storage is described in this area.

e. Digitized Voice File Area - Contains the encoded speech.

The entire file structure is maintained on the disk as a contiguous file having the name "DIRECT.DVF". In operation, the dictionary portion (all structures except the digitized voice file area) is brought into core memory. The voice file area remains on the disk, and is accessed as needed by the pointers in the dictionary.

To visualize the functioning of these various areas it is best to examine a typical dictionary entry (Figure 5-3). The entry shown is an example of how the encoded utterance "New York City" might be accessed by that name as well as by the abbreviation "NYC". The files are accessed first through the file descriptor area. This area consists of a number of three-word blocks which appear in detail in the example. Most of the important capabilities are provided by the information in these blocks.

The first word points to the file name the second contains the file status and size information, and the third points to the disk location. Alphabetic sorting is accomplished by moving these descriptor blocks, rather than directly moving the variable-length entries in the file name area.

The file name can be of arbitrary length, contain embedded blanks, and must be terminated by a zero byte. New names are added to the end of the name area, while the corresponding three-word block is inserted in the correct position in the block area as determined by the sort.

The file status and size (second word of descriptor block) contain the following information: The file size (in 246 word blocks) is contained in the low order 7 bits of the high order byte. The number of words actually used in the last block is contained in the low order byte. The high order bit of the word is used for abbreviations. When a dictionary entry is first created, the order bit, called a "synonym flag" is left zero. If an entry is being created as an abbreviation for an already existing entry, the remainder of the second and third words are copied from the original entry. During editing, an entry's synonyms are found by comparing the file pointer and size words for a match with other dictionary entries. Note that an abbreviation entry is identical in every respect to other entries except for the flag. When an empty file is created, a three

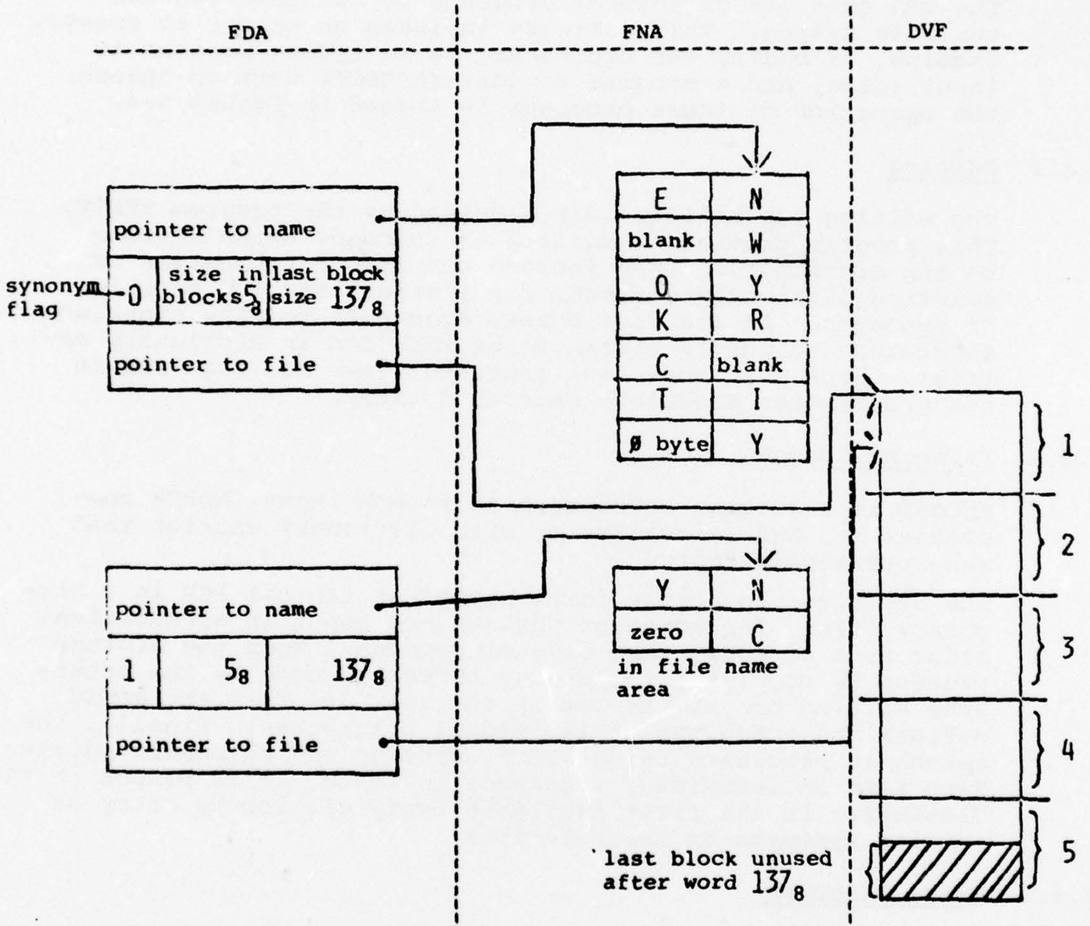


FIGURE 5-3. EXAMPLE OF FILE STRUCTURES CREATED FOR THE FILE NAME "NEW YORK CITY" AND ITS ABBREVIATION (SYNONYM) "NYC"

byte "unique identifier" maintained in the file header is copied into the file pointer and size of the last block portions of the dictionary entry. This provides a means of distinguishing synonyms before a file is created. When a file is deleted, the block size and the file pointer are copied into the free storage area to give information about available space in the file storage area.

5.3.2 Software

The VRS consists of several programs to maintain and use the file system. This software includes an editor to create, examine, or modify the dictionary, a recording program to input voice, and a program to convert ASCII text to speech. The operation of these programs is traced in Figure 5-4.

5.3.3 Editing

The editing capabilities are provided by the program VEDIT. This program provides a variety of commands which operate on the dictionary. They include commands for creating or deleting dictionary entries, for listing, and for handling of synonyms. In the edit phase, arguments require non-blank separators to permit blanks to be imbedded in dictionary entries. Note that non-blank separators are not required in the program for speaking, described later.

5.3.4 Entering Speech

RECORD is a program which accepts speech input, ADPCM compresses it, and associates it with dictionary entries that were previously empty.

The digitized speech is first stored as ten-bit PCM in a temporary file. The words or phrases are input in alphabetical order from an audio tape or a microphone. When the storage process is complete, the energy threshold used by the utterance editing program is set by the user.[1] This threshold defines the endpoints of individual utterances. Finally, the speech is processed using ADPCM encoding and automatic editing. Each time an individual utterance is found, it is placed in the dictionary in the first available empty dictionary entry as are its synonyms or abbreviations.

5.3.5 Voice Response

H516 is a program to "read" ASCII text. The text is input from the Honeywell H516 computer. The program uses a special lookup routine which combines a binary search with a tree search which is invoked in special cases for phrase look ahead. The result is speed approaching a straight binary search, but with the added ability to compare the input text stream with several matches in the dictionary to determine the best match.

[1] See User Guide, Vol. III, for threshold values.

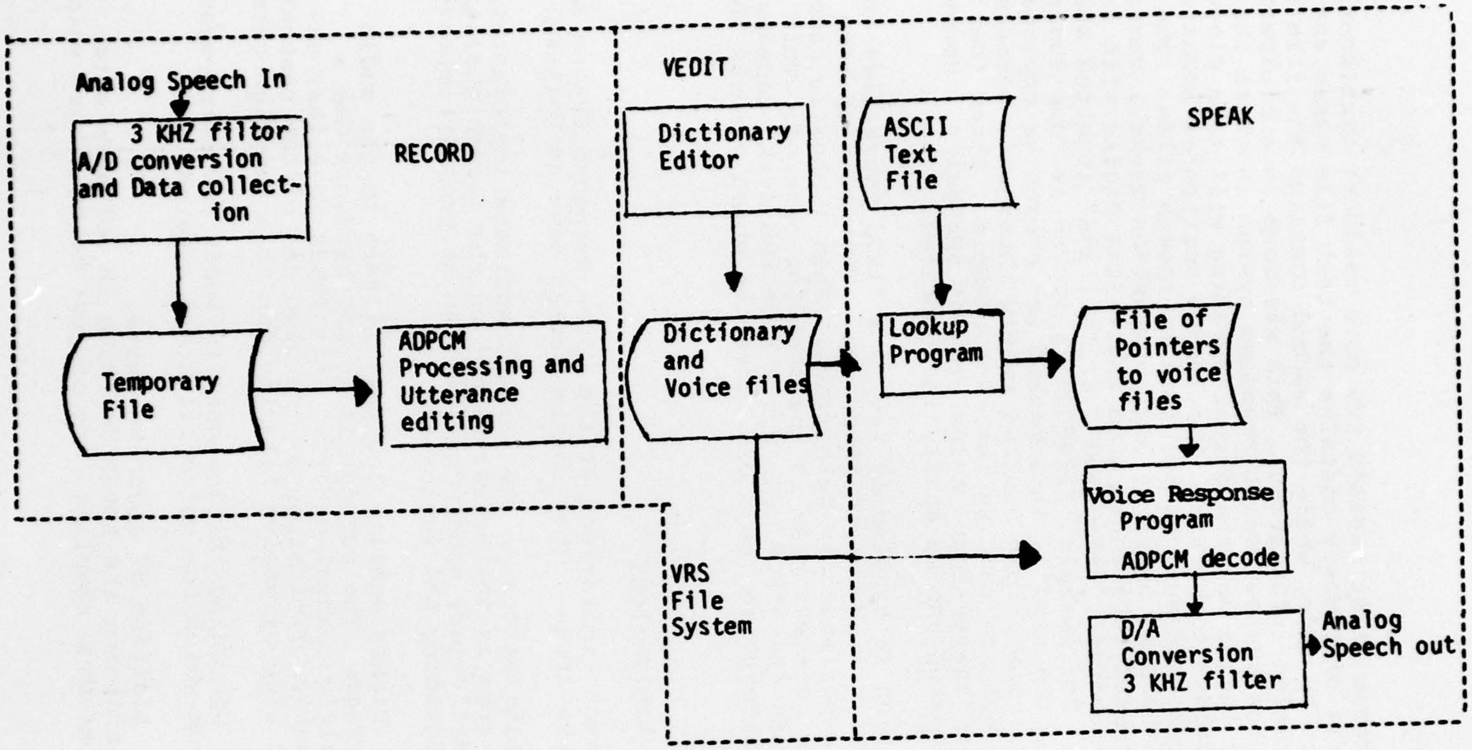


FIGURE 5-4. DATA FLOW BETWEEN PROGRAMS AND DATA BASES

The program first creates two core resident dictionaries. The first dictionary contains the text file names and pointers to them, while the second contains the file size and location information. This was done in anticipation of the ten channel voice response system in which the dictionary lookup and the actual speaking will take place in two separate computers. The lookup portion of SPEAK accepts ASCII text and outputs pointers to speech files. The lookup section handles numerals and punctuation marks. Numerals are currently broken up into individual digits with the words "hundred", "thousand", or "million" inserted where needed. For example, "120000" is spoken as "one hundred two zero thousand". This method was chosen as opposed to the more common "one hundred twenty thousand", because pilots indicate that saying, for example, "three four" instead of "thirty-four" helps prevent ambiguity. However, either method can be easily incorporated.

Punctuation is handled by having a fixed pause time associated with each punctuation mark. The value of this pause can be experimentally determined. When lookup is complete, the file pointers pass to the speaking section. This program reads the ADPCM code words from the disk, decodes them, and transfers them to a D/A converter which outputs the speech.

5.4 CURRENT LIMITATIONS

The current limitations of the Voice Response System fall into three areas. They can be broken down as follows:

a. Limitations in the current software implementation. This area is the primary limit on the speech quality. In particular, speech quality can be greatly improved by upgrading the following:

1) Higher quality low pass filters in the audio system. The current filtering hardware has a fairly gradual cutoff. This leads to either excessive limitation of the high end of the response, or else excessive aliasing due to a too-high cutoff.

2) Upgrading the low quality audio recording system from which the voice files are created.

3) Addition of ADPCM hardware.
Performing the ADPCM decoding in software sets a maximum sampling rate of just below 6 khz. which

is slightly below the response that is needed for speech reproduction. ADPCM hardware would increase the frequency response and would free a substantial amount of CPU Bandwidth for other processing.

b. Areas for further research.
These areas are summarized in the section concerning the test plan. They include experimentation with the automatic utterance editing algorithm, and inflection.

6. PRELIMINARY DESIGN OF TEN-CHANNEL SYSTEM

The initial design of the ten channel VRS is discussed in this section. An overview of the operation of the preliminary system is presented. The estimated software and hardware requirements are then discussed.

6.1 SYSTEM OVERVIEW

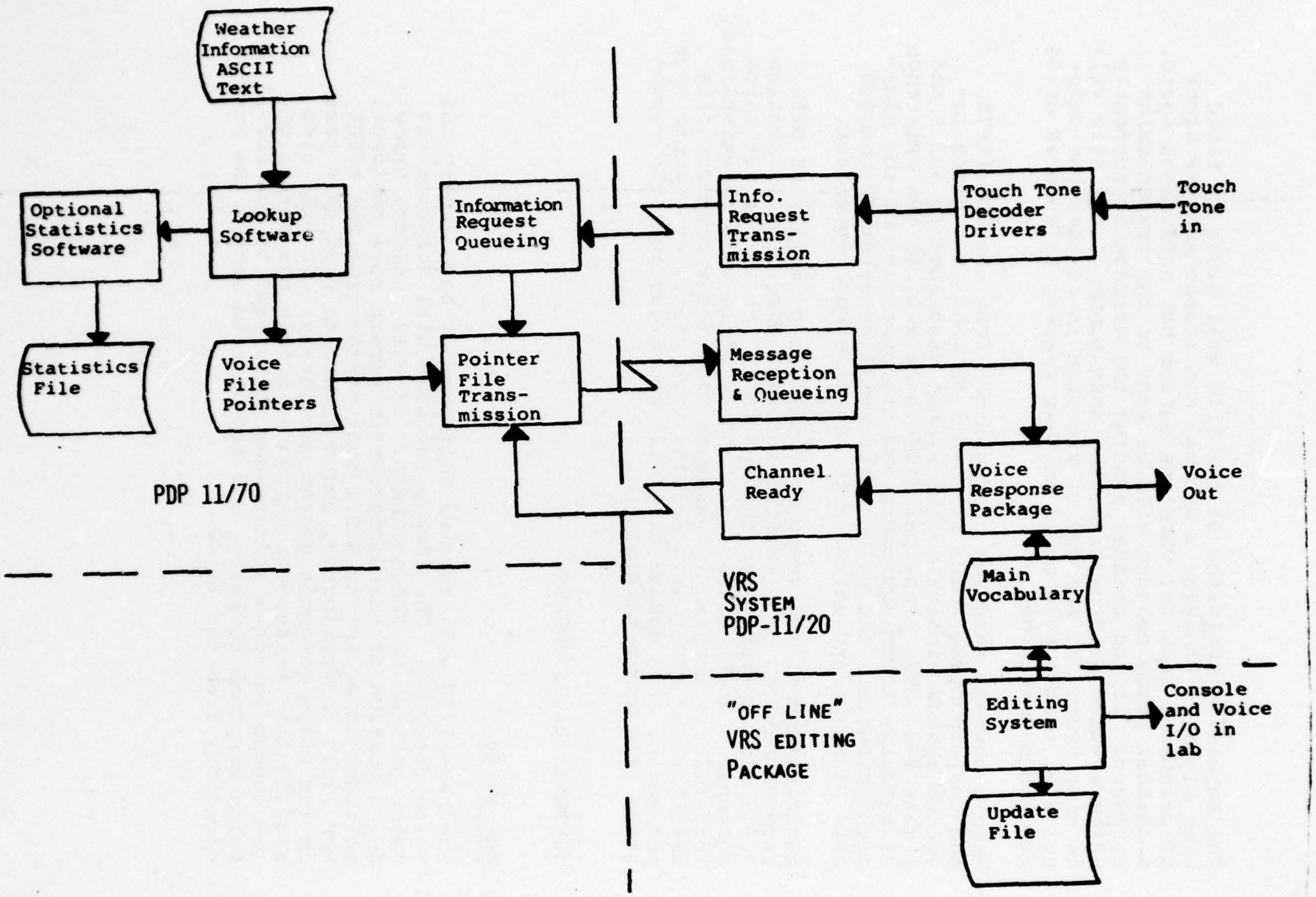
The ten channel VRS is to contain the software and hardware needed to provide voice output for up to ten users. This includes any software or hardware needed to convert ASCII text into spoken words, to handle user requests for such spoken words, and to modify the data bases involved for increased vocabulary and additional program development. A system flow diagram is shown in Figure 6-1.

The ASCII text to be spoken will exist in files residing in a PDP-11/70 system. These files will be constantly updated with current weather information. This is to be done externally to the voice response system.

When the ASCII text files are being updated, the VRS software will create a parallel voice pointer file. This file will be constructed by looking up the contents of the ASCII text file in the VRS dictionary. The dictionary will be sent to the PDP-11/70 from the VRS at start-up time. The voice pointer file's contents will be a series of pointers to digitized voice files which correspond to the ASCII text in the text file. An optional statistics package will monitor the lookup operation to record information such as frequency of use for vocabulary words, and words in the text files which are not in the vocabulary.

The requests for information will enter the VRS through a Telephone Company Line interface in the form of Touch-Tone signals. These signals will be decoded and converted to a message consisting of a channel number and an ASCII character corresponding to the touch-tone button depressed. The messages will then be translated into a channel number and a request to speak a specific voice pointer file.

The request will be transmitted to the PDP-11/70 via a 2400 baud serial interface and related driver software. The request will be queued for the program which will transmit the voice pointer file to the VRS system.



Weather Information ASCII Text
Lookup Software
Optional Statistics Software
Statistics File
Voice File Pointers
Information Request Queueing
Pointer File Trans-mission
PDP 11/70
Info. Request Trans-mission
Touch Tone Decoder Drivers
Touch Tone in
Message Reception & Queueing
Channel Ready
Voice Response Package
Voice Out
Main Vocabulary
VRS SYSTEM PDP-11/20
"OFF LINE" VRS EDITING PACKAGE
Editing System
Console and Voice I/O in lab
Update File

FIGURE 6-1. TEN-CHANNEL VRS SYSTEM FLOW DESCRIPTION

The actual transmission of the file will occur in fixed size blocks. Whenever a channel can accept another block of pointers, that is, when its buffer has sufficient space, a channel ready message will be sent to the transmission software. If the channel sending the message is currently using a voice pointer file, the next block of the file will be sent over. If the end of the file is reached the channel will remain idle until another request is entered using the Touch-Tone phone.

When a channel receives the pointers from the PDP-11/70, the corresponding voice file will be sent to a voice response system similar to the system described in 5.3.5 and Figure 5-4. The principal difference will be the existence of hardware ADPCM equipment and the operation of the playback software. The voice output will be returned through this hardware and software to the telephone interface.

Modification of the vocabulary will be primarily an off-line operation. The editing software described in Section 5.3.3 will operate when the system is not in use for voice response. When editing, an "update" file will be maintained. This file is completely separate from the vocabulary file used for the VRS. At scheduled intervals, for example once per month, this update file will be merged into the vocabulary file and reinitialized.

6.2 SOFTWARE REQUIREMENTS

6.2.1 PDP-11/70

The PDP-11/70 will require software for both lookup and voice response. The lookup software will comprise the major portion of the software. It will include: space for a portion of the directory to remain core resident, software to accept an ASCII text stream from the MITRE PDP-11/70 in Washington, and software to find this text in the voice response system dictionary. It will also include file management software to file the output of the lookup routines as messages to be sent to the PDP-11/20 voice response computer, and software to accept the current directory from the 11/20.

Software	Size	Comment
Core resident directory	12-16K	Contains text names and pointers to speed lookup.
Lookup software	2-3K	Searches dictionary for matches with weather text.
Directory input from PDP-11/20	1K	Can be overwritten by above software once directory is in core.
File handling	2K	
Communications	1K	

(Preceding three estimates are only approximations as the software is not yet written.)

During voice response the only software required will be that needed to communicate with the PDP-11/20.

Communications	2K	Approximation
Total	19-24K	(+1K overwriteable section)

6.2.2 PDP-11/20

The PDP-11/20 requires software for three major functions: voice response, dictionary and vocabulary editing, and program development. The voice response software will be a critical, real-time system requiring all software to be core resident, therefore it is treated in great detail. The editing and program development functions are not as critical and so are only treated in passing.

6.2.2.1 Voice Response

Voice response software is the collection of all programs needed to provide voice response over ten channels. This includes the following areas:

Software to transmit the directory to the PDP-11/70 computer.

Software to handle the Touch Tone requests, format them, and transmit them to the PDP-11/70.

Software to accept pointers to speech files to be spoken from the PDP-11/70 software to output the speech to the ten channels.

In addition the system will require buffering for each channel and a portion of the directory must be core resident.

Software	Size	Comment
Monitor	3.5K	
Directory Transmission	1K	Overwriteable
Touch tone decode and message formatting	1K	Approximate
Input from 11/70	1-2K	Approximate
Ten channel voice response software	2-3K	Approximate
Buffering for ten channels	10K	1K per channel
Core resident directory	8K	
TOTAL	25.5-27.5K	(+ 1K overwriteable)

6.2.2.2 Vocabulary Editing

The vocabulary editor will include all software for vocabulary creation and modification. This operation is not time critical so the software (currently 12K + Directory) can be overlayed and the Directory need not be core resident.

6.2.2.3 Program Development

Program development will require the standard DEC systems programs (MACRO-11, PDP, etc.).

6.3 HARDWARE REQUIREMENTS

The PDP-11/20 will require a variety of hardware and peripherals to perform the required functions. Except where indicated, the peripherals are assumed to be standard off-the-shelf components. (See Figure 6-2.)

6.3.1 PDP-11/70 to 11/20 Link

This will be a bidirectional serial interface capable of a minimum data rate of 2400 baud.

6.3.2 Vocabulary Store

The vocabulary store will consist of four megawords of storage capable of servicing an average of 75 requests for a 256 word sector every second.

6.3.3 Program Storage

Program storage will be a cartridge disk system to contain the program library and software development work space.

6.3.4 Program Backup

A magnetic tape unit will be used to backup vocabulary and developed programs. It will also be used in making recordings for vocabulary generation.

6.3.5 Master Console

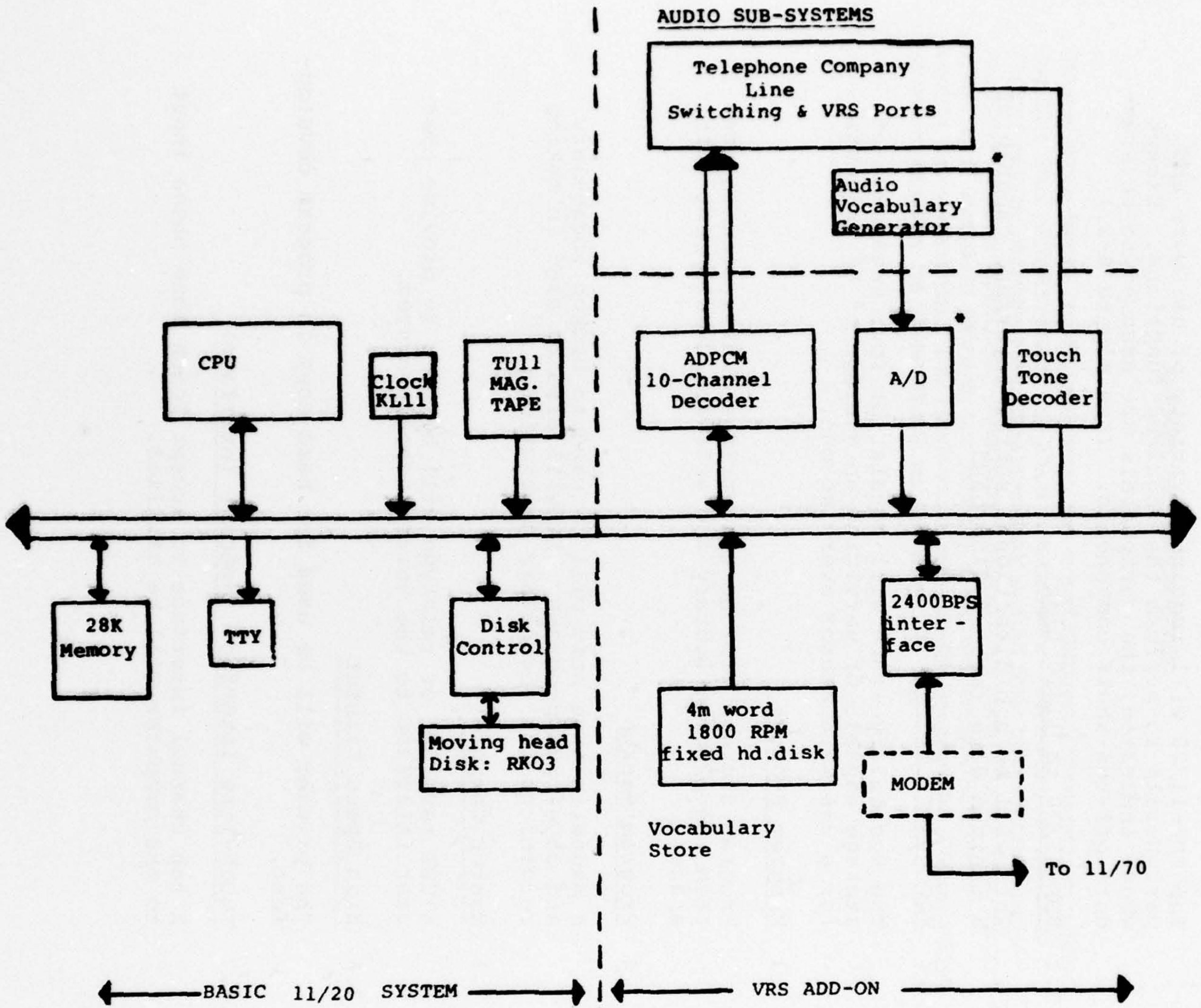
A CRT terminal or teletype will be used to provide operator interface to the voice response system.

6.3.6 High Speed Printer

The printer will be used for hard copy in program development.

6.3.7 Touch-Tone Interface (Special Interface)

A ten channel interface to accept Touch Tone phone input to the computer will be utilized.



* Required for vocabulary development

FIGURE 6-2. TEN-CHANNEL CONFIGURATION

6.3.8 ADPCM Decoder (Special Interface)

A ten channel ADPCM decoder to provide speech output will be utilized to provide output to the phone lines. Any of the ten lines may be monitored in the lab.

6.3.9 Voice Input

The voice input will consist of a microphone and preamp and either a ten bit A/D converter or an ADPCM encoder (special interface). These will be used in the lab for vocabulary input.

APPENDIX

REPORT OF INVENTIONS

There have been no inventions or significant discoveries made during the performance of this contract. However, a unique file system design was implemented which has the characteristics of being fast for speaking, flexible (but slow) for editing, and compact enough to permit a core-based dictionary, which is essential for real-time playback response. This file system design is described in Section 5.3.1.

180 copies