A PRELIMINARY INVESTIGATION OF AURAL INPUT/OUTPUT SYSTEMS FOR IN-FLIGHT INFORMATION RETRIEVAL

Albert George Mertz

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PRELIMINARY INVESTIGATION OF
AURAL INPUT/OUTPUT SYSTEMS FOR
IN-FLIGHT INFORMATION RETRIEVAL

by

Albert George Mertz

September 1978

Thesis Advisor: D. M. Layton

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

A Preliminary Investigation of Aural Input/Output Systems for In-Flight Information Retrieval

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis
September 1978

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Albert George Mertz

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

September 1978

13. NUMBER OF PAGES

163

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech Understanding System (SUS)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An analysis of two commercially available Speech Understanding Systems (SUS) was conducted. Each system was tested against various background noise conditions. Results obtained were compared with current criteria for SUS application in aircraft. Additionally, since the P-3 Orion aircraft is being considered as a SUS test aircraft, a survey of Fleet P-3 pilots was conducted. Their opinion was sought on what a SUS equipped microprocessor should be capable of accomplishing as an aid to the flight crew.

Approved for public release; distribution unlimited

A Preliminary Investigation of Aural Input/Output Systems for In-Flight Information Retrieval

by

Albert George Mertz
Lieutenant, United States Navy
B.S., United States Naval Academy, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1978

ABSTRACT

An analysis of two commercially available Speech Understanding Systems (SUS) was conducted. Each system was tested against various background noise conditions. Results obtained were compared with current criteria for SUS application in aircraft. Additionally, since the P-3 Orion aircraft is being considered as a SUS test aircraft, a survey of Fleet P-3 pilots was conducted. Their opinion was sought on what a SUS equipped microprocessor should be capable of accomplishing as an aid to the flight crew.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author would like to thank the following Officers who not only acted as test subjects for these experiments but also contributed many helpful suggestions throughout the course of the testing:

LCDR Richard M. Fifer

LT Gary G. Maxwell

LT Jon A. Buresh

Special thanks must also go to Mr. Paul Sparks, the Technician at the Human Factors Laboratory, Naval Postgraduate School. Mr. Sparks hand built from kits the IMSAI 8080 and all controlling Input/Output boards required for operation of the microprocessor. He also contributed freely of his time and technical expertise in solving problems encountered during interfacing system components. His personal involvement was a significant factor in the success of this experiment.

I. INTRODUCTION

Recent years have seen a remarkable advance in aircraft technology, along with a corresponding increase in aircraft complexity and pilot workload. Since a high pilot workload has always been associated with a high aircraft accident rate, it is most desirable to find some means of reducing pilot workload as much as possible.

One possible answer lies in the area of Speech Understanding microprocessor systems. These would allow the pilot of an aircraft so equipped to verbally command a microprocessor system which in turn would cause the aircraft to follow the pilot's instructions, such as a climb or descent to a specified altitude. The pilot might also request information from the computer, such as a Maximum range profile or to display the appropriate emergency checklist as required. Voice communication with the computer would free the pilot of the necessity of keying in instructions as current technology requires.

Speech Understanding Systems (SUS) have experienced rapid advancement in recent years also, with both government and private industry sponsored programs being undertaken. Reference 1 focuses on the operational military applications of Speech Understanding Systems and provides an overview of the nontechnical factors in the military environment which are likely to affect the introduction of SUS capabilities in military systems.

Microprocessor hobbyists have also developed systems not only for speech understanding but speech synthesis as well. Two such systems were available at the Naval Postgraduate School through the Human Factors Engineering Lab. Heuristics, Inc. had developed a system called SPEECHLAB for speech understanding, and Computalker Consultants had developed COMPUTALKER for speech synthesis.

Both systems have advantages which make them particularly applicable to aircraft cockpit design. They are both small in physical size and require less than 6K of RAM for their controlling programs. Vocabulary size dictates the actual amount of RAM required by each system, with 5K being adequate for a 16 word vocabulary in the SPEECHLAB system. The COMPUTALKER system, as currently structured, allows the user to encode words or phrases, according to phonetic spelling rules, in up to 76 characters at a time. Since the average word requires less than 15 such characters, phrases of approximately four words could easily be constructed. The main advantage of the COMPUTALKER system is that many such phrases could be stored in available memory, loaded as required into the COMPUTALKER input buffer area, and quickly output as recognizable speech. This eliminates the requirement for a complicated algorithm to assemble prestored words into phrases, thus saving valuable memory area. The current COMPUTALKER input buffer requires approximately 2½K of memory for encoding the 76 characters. Both systems are relatively inexpensive, with current models available for under $300 in kit form. [Ref. 2 and 3]

This study was undertaken to determine the performance of each system as well as the possibility of incorporation in an aircraft cockpit control system. Specific goals were:

1) Determine response time and accuracy of SPEECHLAB under low background noise level conditions.
2) Determine the recognizability of COMPUTALKER under low background noise conditions.
3) Repeating goals 1) and 2) with varying levels of background noise intensity to determine the limiting tolerance of each system to background interference.

Current studies being carried on by the Navy indicate the possible use of the P-3 Orion aircraft as a test bed for voice controlled cockpit systems [Ref. 4]. With this in mind, a survey of Fleet P-3 pilots was conducted to determine their reaction of the possible incorporation of such a voice controlled system in the P-3. The survey questions used are listed in Appendix C. The answers to the questions are discussed in the Results and Conclusions section of this report.

II. DESCRIPTION OF APPARATUS

A. SPEECHLAB

The SPEECHLAB hardware, manufactured by Hueristics, Inc., Los Altos, California, is compatible with the 8080 microprocessor developed by Intel and employs the S100 bus [Fig. 1].

Audio input is amplified and passed through three band pass filters encompassing the range from 150 to 900 Hz, 900 Hz to 2.2 KHz, and 2.2 to 5 KHz. These ranges roughly correspond to the first three resonances of the human vocal tract. A zero-crossing detector generates a voltage proportional to the number of times the raw waveform crosses the rest level in a given period of time. The signal is then passed to a six-bit A/D converter. The output of this A/D converter is fed directly onto the computer data bus.

The controlling software programs utilized in this experiment were taken from the manual supplied with the SPEECHLAB board and modified for use on the Human Factor Engineering Labs' 8080 microprocessor. Investigations conducted prior to commencement of the experiment showed the algorithm which used Euclidian distance measure to be the most accurate of the programs in the manual. Of the most importance in this particular program was the variable S, which represented the number of divisions into which the speech utterance would be divided in order to be analyzed by the program. This parameter, set at 64 as provided, was shown to have the greatest effect on

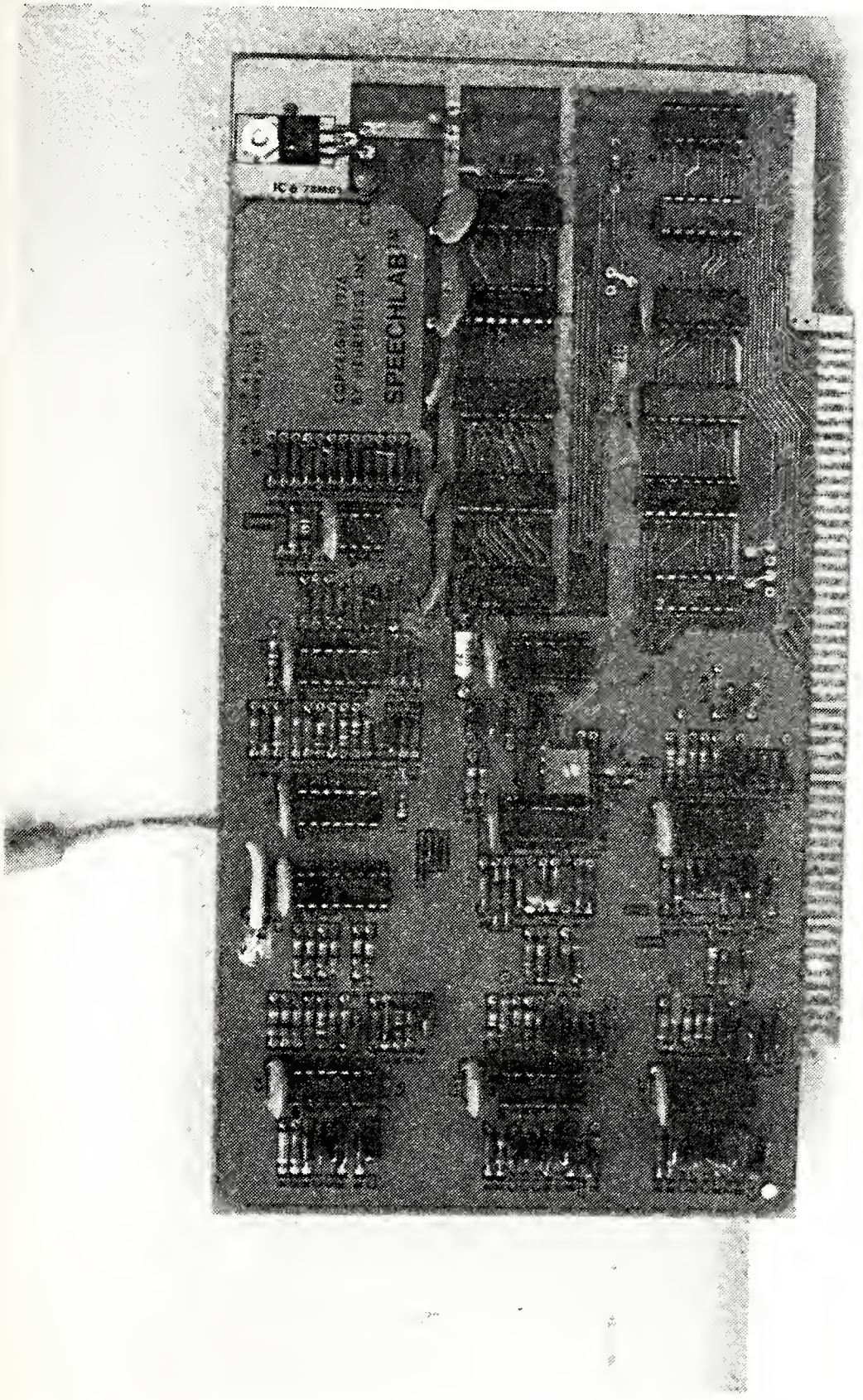

Figure 1. SPEECHLAB Hardware Board

response time. Response time was measured from the closing of the speech window to the time the machine began to type the recognition message on the teletype. The speech window width as set by the algorithm was approximately 2.5 sec in duration. This is a variable and could be changed by the operator. The beginning of the speech window was signaled by a beep at the microphone in use. A similar beep signaled the close of the window [Ref. 5]. The Basic and controlling programs used by speech lab are included as Appendix A.

B. COMPUTALKER

The COMPUTALKER hardware was developed by Computalker Consultants, of Santa Monica, California [Fig 2].

The COMPUTALKER board is controlled by CSR1, a software program provided with the system. CSR1 is a phoneme-input speech synthesizer program. It contains definitions in the form of various table entries for 55 different input symbols which represent the phonemes of the English language along with several punctuation marks. CSR1 accepts a phonetic input string consisting of these phonemes and punctuation symbols and decimal digits which specify vowel stress levels. This input string is parsed to fill out a phonetic feature matrix for the desired synthetic phrase. A set of phonetic structure rules scans the matrix to look for feature patterns and combinations which must be adjusted to conform to English pronunciation. After these adjustments, a parameter generator algorithm scans the matrix and produces the control parameters needed to operate

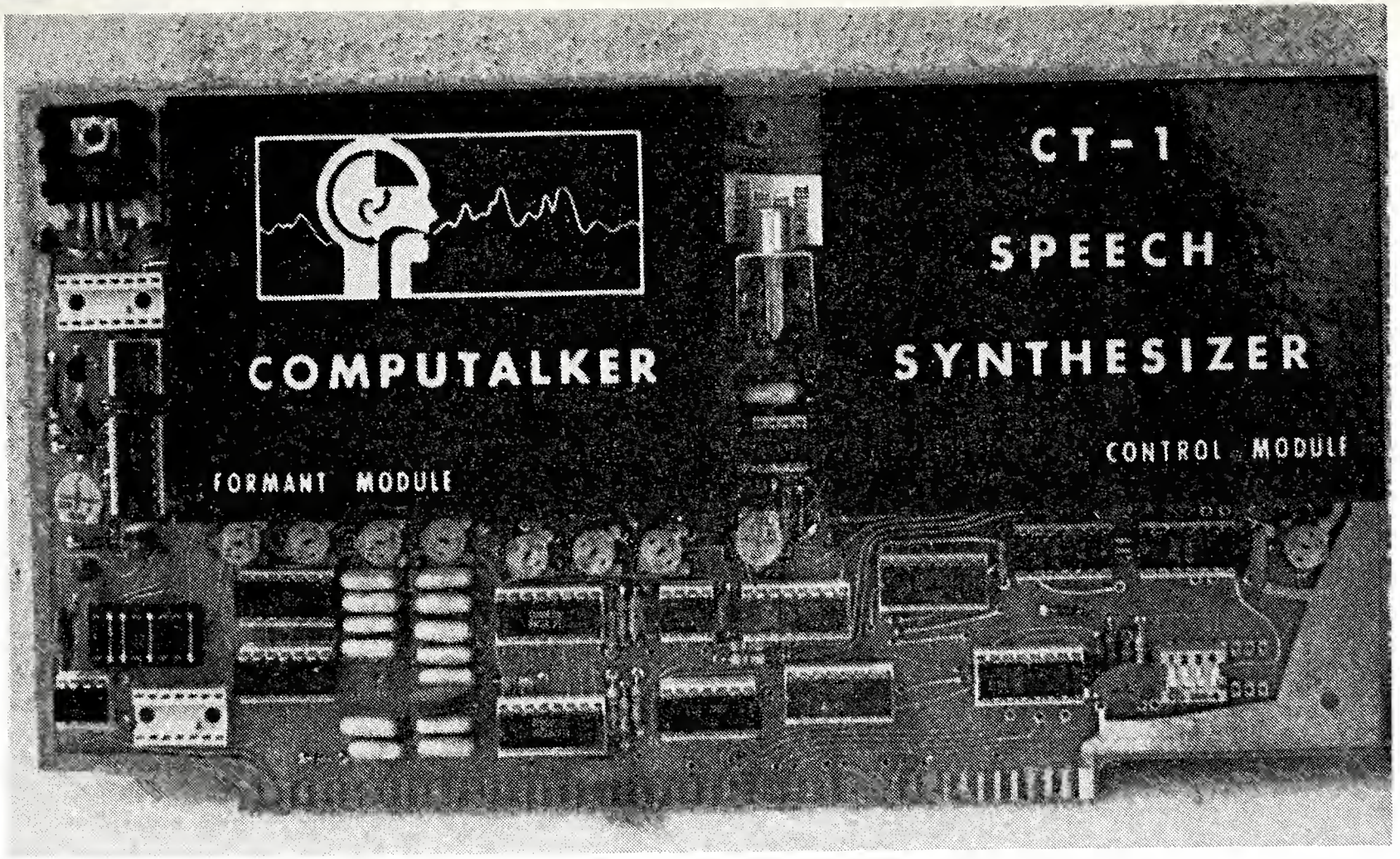

Figure 2. COMPUTALKER Hardware Board

the COMPUTALKER Synthesizer [Ref 6]. The COMPUTALKER board is shown in Figure 2. The CSR1 program is included as Appendix B.

C. SOUND LEVEL METER

The sound level meter used during the experiment was the General Radio Corporation, Type 1565-B. The range of the instrument was from 40 dB to 140 dB, referenced to 20 micro Newtons per square meter. Unless otherwise noted, all sound pressure level measurements in this experiment were made with this sound level meter. The meter also provided for the use of three different weighting levels, A, B, or C, which conform to ANSI S1.4-1971 Type 2 and IEC 123,1961 specifications. Weighting C was chosen for use in this experiment as it provided the flatter response curve over a wider range of frequencies than either of the other weightings [Fig. 3].

D. MAICO AUDIOMETER

The MAICO Dual Channel Research and Diagnostic Audiometer, Model MA-24B, was used to provide the background noise and pure tones used during the course of this investigation. The Hearing Threshold Level in each channel could be controlled accurately through the range from -3 dB to 112 dB. This equated to a range of from 47 dB to 162 dB when referenced to 20 micro Newtons per square meter [Fig. 4].

E. MICROPROCESSOR

The microprocessor used for the experiments was built from a kit manufactured by IMSAI Corporation. The basic kit utilizes

Figure 3. Sound Level Meter

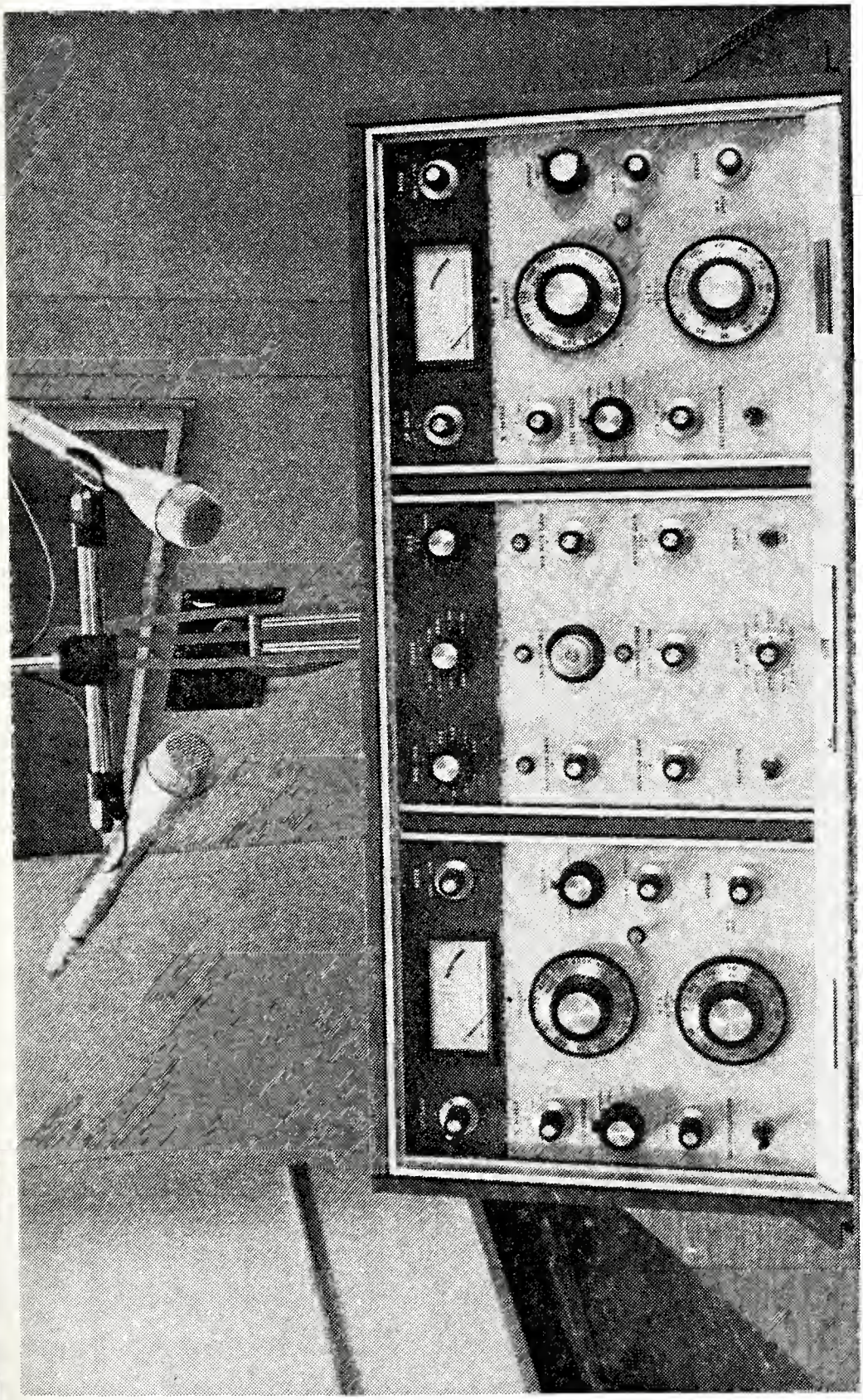

Figure 4. Audiometer

the INTEL 8080 microprocessor chip and standard INTEL 8080 instructions. An IMSAI Multiple Input/Output board was used for input/output. Memory available included 4K of PROM and 32K of RAM. Sufficient space was available for up to 32K additional memory, assuming the same type of memory packaging. Figure 5 shows the 8080 microprocessor with the SPEECHLAB, COMPUTALKER, memory and I/O boards installed.

Figure 6 shows the Audiometer and Microprocessor set up outside the test booth. The microprocessor was controlled with a standard teletype keyboard. Figure 7 shows the interior of the test booth. All inputs to the booth were channeled through the patch box on the right of the table. The microphone stand had full freedom of motion which allowed the subjects to position the microphone within one inch of their lips while speaking. The speakers used to introduce noise, tones and the COMPUTALKER speech are located in the rear of the booth.

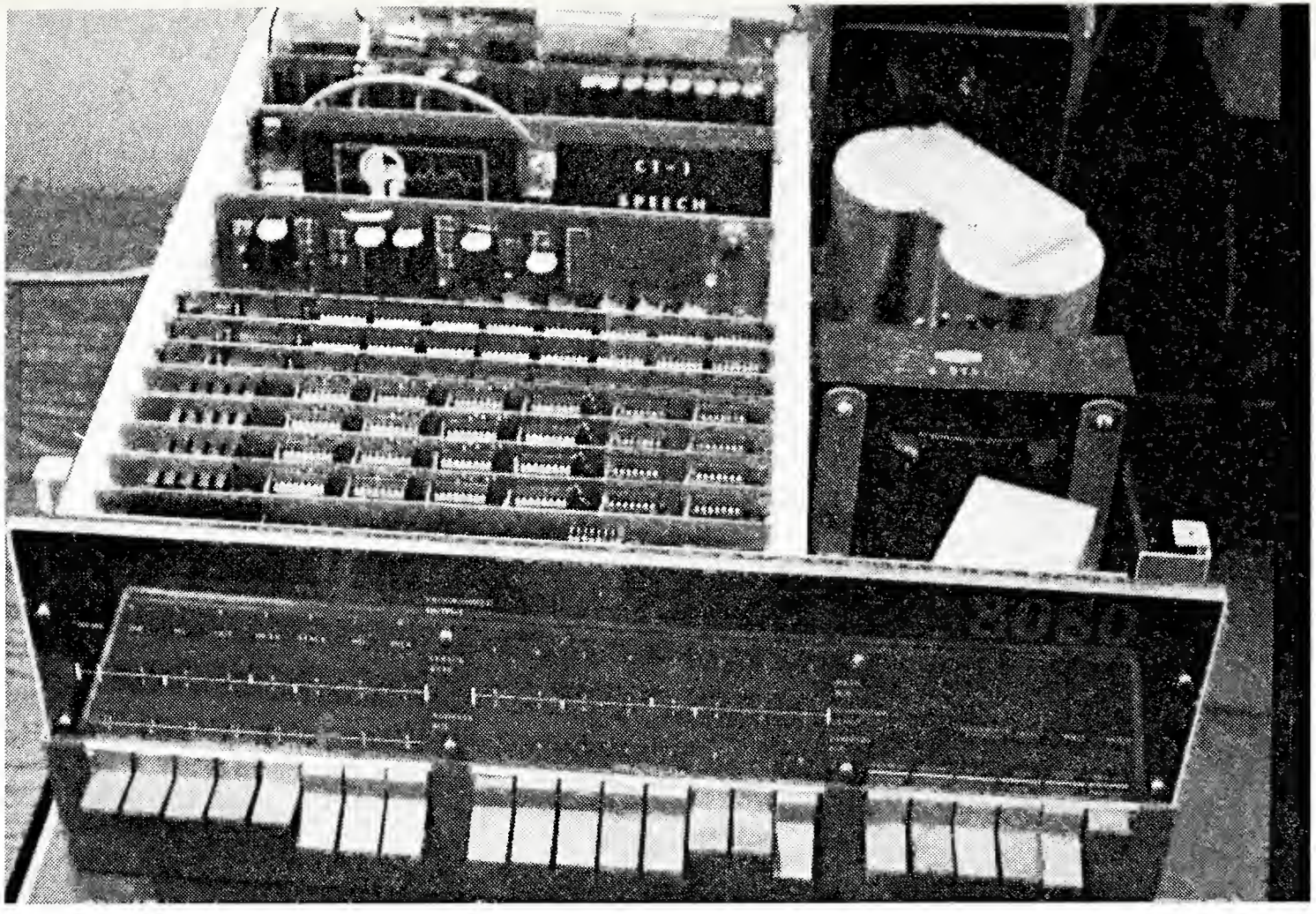

Figure 5. 8080 Microprocessor with SPEECHLAB and COMPUTALKER installed

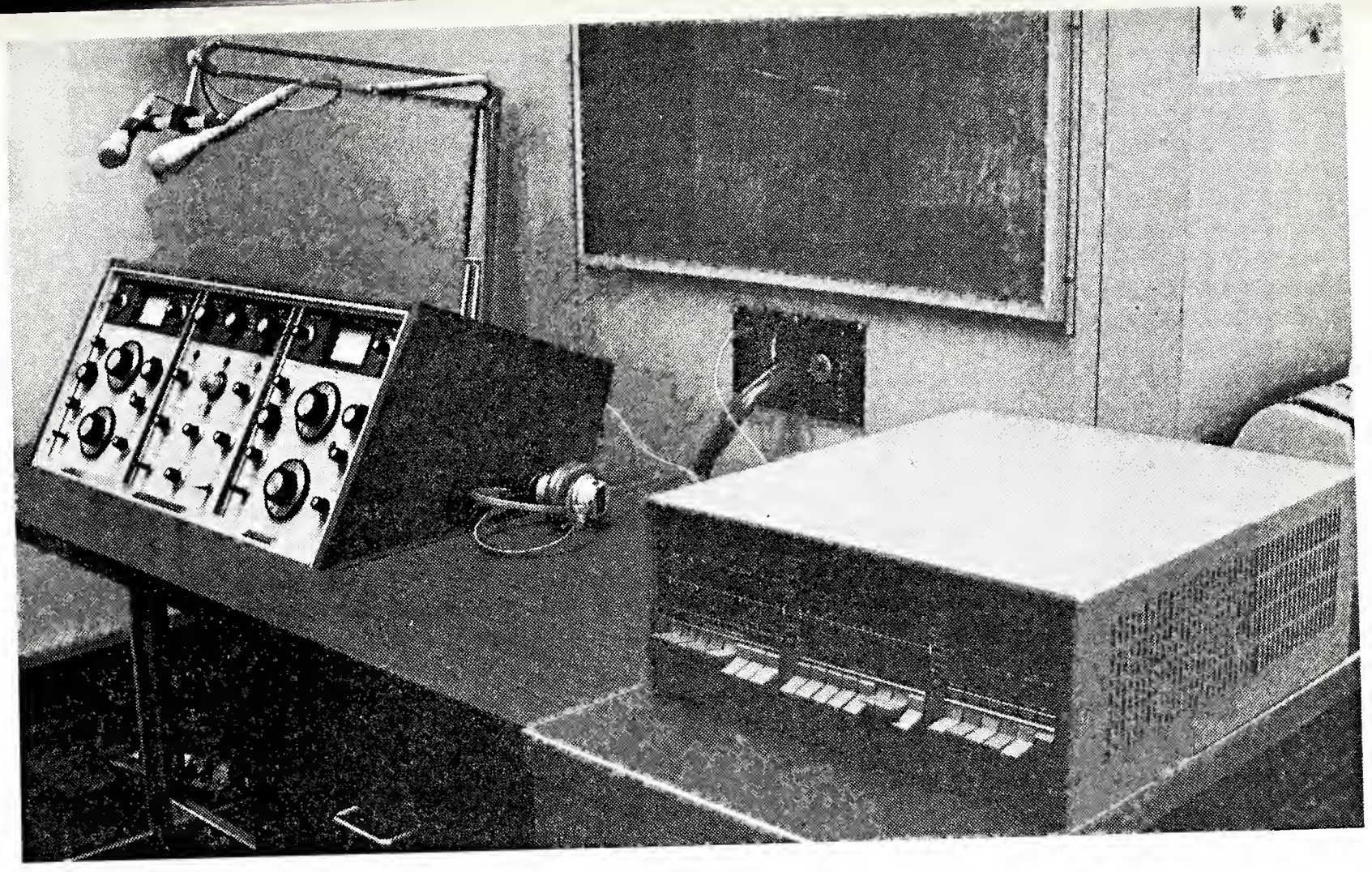

Figure 6. Audiometer and Microprocessor Outside Test Booth

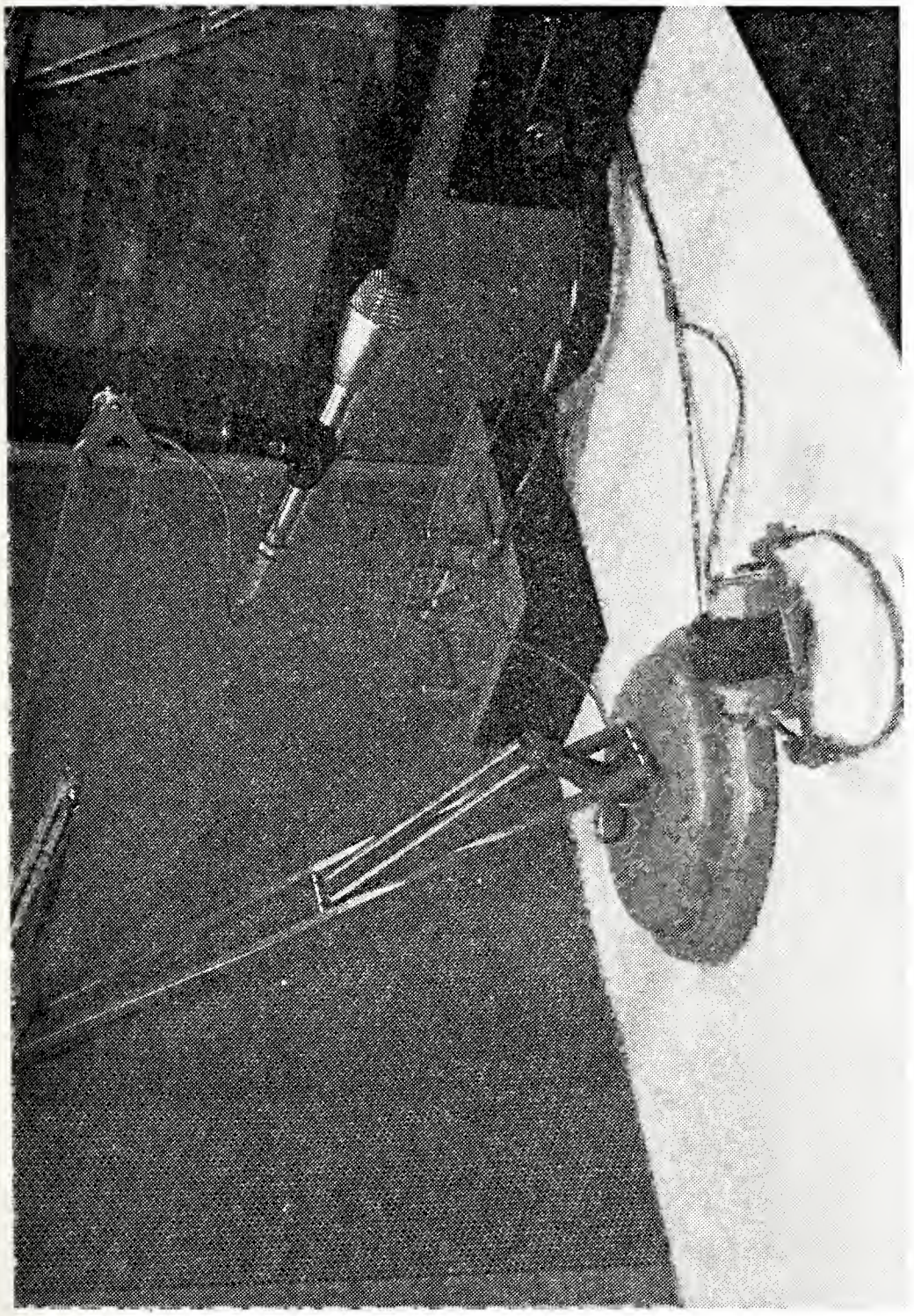

Figure 7. Microphone Setup Inside Test Booth

III. EXPERIMENTAL PROCEDURE

All experiments were conducted at the Human Factors Engineering Laboratory at the Naval Postgraduate School.

A. SPEECHLAB TESTING

For the voice recognition tests, each subject trained the SPEECHLAB unit on the ten word vocabulary selected from Ref. 1. No background noise was introduced into the booth in which the subjects operated from. Noise level in the booth under these conditions was measured at 40 dB using the sound level meter. After training, the performance mode of the system was entered. Background noise level remained at 40 dB for the tests. The subject was required to read through the word list, in order, five times, thus obtaining fifty samples of speech utterances from each subject per test. Each test was repeated three times, with the S parameter of the SPEECHLAB algorithm set at 64, 32, and 16. This was done to test the performance of the algorithm with varying sample sizes. Typical results are shown in Figure 8.

Next, various background noise levels were introduced in the booth to determine system response to noise. The subjects followed the same procedures as in the no-noise tests. The S parameter remained set at 64 throughout the tests as this was determined to be the most accurate mode for SPEECHLAB. Background noise level was raised in 5 dB increments from 60 dB to

DATE 17 Aug 78

SUBJECT Buresh

SPEECHLAB X S 64

COMPUTALKER ___

PROGRAM 1

BACKGROUND NOISE 40 dB

FREQUENCY --

TIME (AVERAGE) 27 sec

S \ R	1	2	3	4	5	6	7	8	9	0
1	5									
2		5								
3		2							3	
4				5						
5					5					
6						4	1			
7							4		1	
8								5		
9									5	
0										5

Total $\frac{43}{50} = 0.86$

WORD LIST

1) ALTITUDE
2) SPEED
3) HEADING
4) POSITION
5) TACAN
6) VOR
7) ILS
8) GLIDESLOPE
9) CLIMB
0) DESCEND

Figure 8. Typical Results of SPEECHLAB Tests (with low background noise)

80 dB. However, in no test was the SPEECHLAB system able to accomodate more than 74 dB. Typical results for this series of tests are shown in Figure 9.

To test the effect of frequency on the SPEECHLAB system, pure tones ranging from 4000 Hz to 8000 Hz were induced and the tests repeated by the subjects. Incremental tone steps were 1000 Hz. Typical test results are shown in Fig. 10. A summary of results from the SPEECHLAB tests is presented in Fig. 11.

B. COMPUTALKER TESTING

To test the accuracy with which COMPUTALKER produced speech, the same word list used for the SPEECHLAB tests was prepared for use with COMPUTALKER. Each word on the list was rearranged according to the phonetic spelling rules used by COMPUTALKER. Various combinations of phonetic spellings were tried until an acceptable utterance was produced. This list of words, which were now optimized to the principal researcher, was then presented to each of the subjects. Re-optimization, where necessary, was done until each subject readily recognized each word on the list. The word list and final phonetic spelling was as follows:

(1) ALTITUDE -- AH2LTTIYTUX3D

(2) SPEED -- SHPQIY3DX

(3) HEADING -- HHEHDIYNGX

(4) POSITION -- POHSIH1SSIYOHN

(5) TACAN -- TAE3QKAAN

DATE 11 Aug 70

SUBJECT Fifer

SPEECHLAB X S 32

COMPUTALKER ___

PROGRAM _____

BACKGROUND NOISE 65 dB

FREQUENCY --

TIME (AVERAGE) 26 sec

S \ R	1	2	3	4	5	6	7	8	9	0
1	3		1							
2		2	3							
3			3		1		1			
4		4		1						
5	3		1						1	
6		2	3							
7	1		3			1				
8	1		1					3		
9		1	4							
0				3			2			

Total $\frac{12}{50} = 0.24$

WORD LIST

1) ALTITUDE
2) SPEED
3) HEADING
4) POSITION
5) TACAN
6) VOR
7) ILS
8) GLIDESLOPE
9) CLIMB
0) DESCEND

Figure 9. Typical Results of SPEECHLAB Tests (with various noise levels)

DATE 22 Aug 78

SUBJECT Buresh

SPEECHLAB X S 64

COMPUTALKER ___

PROGRAM 1

BACKGROUND NOISE 75 dB

FREQUENCY 4000 Hz

TIME (AVERAGE) 28 sec

S \ R	1	2	3	4	5	6	7	8	9	0
1	3		1						1	
2	1	4								
3			4						1	
4			2	3						
5				1	4					
6		1				4				
7							5			
8								5		
9			1						4	
0										5

Total $\frac{41}{50} = 0.82$

WORD LIST

1) ALTITUDE
2) SPEED
3) HEADING
4) POSITION
5) TACAN
6) VOR
7) ILS
8) GLIDESLOPE
9) CLIMB
0) DESCEND

Figure 10. Typical Results of SPEECHLAB Tests (with various tone levels)

	NO BACKGROUND NOISE (1)			BACKGROUND NOISE (2)				BACKGROUND TONE (3)				
	S			dB				Hz				
SUBJECT	64	32	16	60	65	70	74	4K	5K	6K	7K	8K
1	100	84	58	86	70	42	37	100	98	82	92	90
2	76	46	32	32	24	33	20	76	72	72	72	70
3	86	78	52	78	60	37	27	82	86	86	80	80
4	90	82	60	82	60	30	24	80	82	75	75	78
AVERAGE	88	72.5	50.5	69.5	53.5	35.5	27	84.5	84.5	78.8	79.8	79.5

(1) With no noise input to the test booth, the noise level inside the booth was measured to be 40 dB.

(2) Background noise was broadband white noise.

(3) Pure tones were input at 75 dB.

Figure 11. Summary of Results from SPEECHLAB Testing

(6) VOR -- VVIY1, OH1UW, REH1R

(7) ILS -- AY1, ELL, SS

(8) GLIDESLOPE -- GXLAY3DSLUHP

(9) CLIMB -- KLAY1MB

(0) DESCEND -- DIYSEHND

The phonetic spelling rules used by COMPUTALKER are shown in Fig. 12.

Consonants

P	p	pie
T	t	tie
K	k	key
B	b	by
D	d	die
G	g	guy
M	m	my
N	n	nigh
NX	ŋ	hang
F	f	fie
V	v	vie
TH	θ	thigh
DH	ð	thy
S	s	sigh
Z	z	zoo
SH	ʃ	shy
ZH	ʒ	vision
L	l	lie
W	w	we
R	r	rye
Y	j	you
HH	h	high
CH	tʃ	chime
JH	dʒ	jive
WH	ʍ	why
EL	l̩	battle
EM	m̩	bottom
EN	n̩	button
Q	ʔ	(glottal stop)

Vowels

IY	i	heed
IH	ɪ	hid
EY	eɪ	hayed
EH	ɛ	head
AE	æ	had
AA	ɑ	hod
AO	ɔ	hawed
OW	oʷ	hoed
UH	ʊ	hood
UW	uʷ	who'd
ER	ɚ	herd
AH	ʌ	Hudd
AY	ɑɪ	hide
AW	aʷ	how
OY	ɔɪ	boy
AX	ə	about
IX	ɨ	David
OH	o	core
UX	ʉ	too

Punctuation

space	word boundary
,	pause/silence
.	falling pitch
?	rising pitch
return	end of input

Stress Marks

Ø	no stress
1	primary (max) stress
2	secondary stress
3	tertiary stress
4	etc.
5	
> 5	no stress

Several other symbols are used internally by certain rules. These may also be used in the input string.

KX	coo	(K before back vowel)
GX	goo	(G before back vowel)
RX	card	(R after a vowel)
LX	kill	(L after a vowel)
DX	pity	(T between vowels)
YX		diphthong ending
WX		diphthong ending

Table 1 CSR1 Phonetic Input Symbols

Stress marks may be placed on any vowel in the form of a following digit from 5 (weakest stated stress) to 1 (maximum stress). Any stress digit greater than 5 will be ignored. Ending an input string with a period or question mark has the effect of changing the ending pitch, falling or rising, respectively. A comma has no effect on the pitch, but only introduces a pause in the pronunciation.

Figure 12. COMPUTALKER Phonetic Spelling Rules

IV. RESULTS AND CONCLUSIONS

A. SPEECHLAB

Results from the SPEECHLAB tests showed that while the performance approximated what the manufacturers claim, i.e., acceptability percentage in the high nineties, any significant amount of background noise causes a marked deterioration in its performance. Above 74 dB, the algorithm would not accept speech utterances at all, rather signaling "SPEECH OUT OF WINDOW", indicating that it thought the noise was the speech utterance.

An attempt was made to improve the performance of the algorithm by training it with the background noise set at 62 dB. While this did improve the recognition rate of the program with the background noise level at 60 and 65 dB, performance again fell off rapidly at 70 dB and at 74 dB the algorithm again could not distinguish the speech from the background noise.

A noise cancelling boom mike would almost certainly have improved the performance of the system, as would the wearing of an oxygen mask. However, as noted by previous researchers [Ref. 7], the background noises induced by heavy breathing under moments of stress and high G interfere greatly with a speech recognition system's performance.

The response time for the most accurate algorithm, i.e., with S set at 64, was judged to be unacceptable, averaging out to 27 sec in all tests. However, with the S parameter set at

16, the average response time was 9 sec, a slightly high but possibly acceptable time [Ref. 8]. The system accuracy in this case dropped to just slightly more than 50%, clearly unacceptable.

Obviously, the algorithm provided with the SPEECHLAB hardware requires a great deal of improvement. If this is not possible, and none of the improvements suggested by the SPEECHLAB manual [Ref. 9] produce noticeable improvement, a completely new algorithm must be found before the SPEECHLAB system could be considered acceptable for military use. Further testing should also include vibration levels of various intensities to simulate aircraft motion.

Two observations made during the course of the experiment suggest, however, that the low recognition rates obtained may not be due entirely to the algorithms used.

First, it was noted that as a subject became more used to "talking" to the machine, a slight improvement in the algorithm's recognition rate was seen. It appears that the subject began to pronounce the words on the list in a more consistent manner after numerous repetitions. This was especially true when the subject was told that a successful recognition had been achieved. This appeared to reinforce in the subject's mind the particular pronunciation which produces the successful recognition. If the subjects had been trained in this manner prior to commencement of the testing, the initially lower recognition rates might have been avoided. It would seem therefore that operator training would play a significant role in any successful voice recognition system.

The second observation concerned the day-to-day changes that occur in the human voice. On one occasion, a subject reported for testing with a cold which had changed the tone of his voice considerably. Although this particular subject had enjoyed one of the highest recognition rates of the group (in the middle nineties), on this particular day his recognition rate fell to just below 80%. No other subject was tested with any other ailment, but it is felt that such things as a sore throat, asthma or other allergic reactions which changed the tone of the voice would cause a similar drop in the system performance.

The obvious conclusion to be drawn here is that some means must be developed, whether through software programming or hardware devices, to account for possible daily changes in an operator's voice characteristics.

Several attempts were made to train the SPEECHLAB algorithm with one subject and test it with another person speaking. The results were so poor that, in general, it appears impractical with the present system to attempt to use it in such a manner. As presently set up, the best method to use the system with more than one person would be to train the algorithm with each person's voice. This would, of course, require more storage space and greatly increase recognition time.

B. COMPUTALKER

The results from the COMPUTALKER testing were unique in that after initial presentation of the word list, all subjects scored 100% on all tests. Although the initial recognition

rate during the optimization testing was very low (20%), the subjects rapidly related the machine speech with the proper word. Changing background noises and tone had almost no effect on the tests. The only area of difficulty encountered was due to the nature of the white noise introduced to the testing chamber. The sound produced by the noise is best described as a hiss, which is very similar to the letter "S" as pronounced by COMPUTALKER. This had the effect of masking the letter "S" when the machine spoke the word "ILS". All subjects, however, were able to hear the "IL" part of the word with no trouble and correctly deduced the word to be "ILS".

At this point in the experiment, the testing procedure was modified since the vocabulary under consideration was obviously too small, allowing the subjects to easily determine the correct choice by process of elimination. Another ten words were added to the vocabulary and optimized as before. The additional words and their phonetic spelling were as follows:

(11) ABOVE -- AHBAHOHV

(12) BELOW -- BIY3LOW2

(13) CHECK -- CHEHK

(14) DOWN -- DAOWHN

(15) ENGINE -- EHNJHIHN

(16) FIRE -- FQAY1R

(17) GEAR -- GEHER

(18) NAVIGATION -- NAE1VEHGEY1SOHN

(19) PRESSURE -- PREHSUHR

(20) SOME -- SQEMN

Various combinations of ten words from the twenty word vocabulary were chosen and presented to the subjects. Although results were expected to be poor, the opposite was true. Only one subject missed one word. However, all expressed the opinion that they had had a much more difficult time in determining the correct word. This would infer that they thought about their choice much longer than required before. This is obviously undesirable, as a pilot can usually ill afford to have his attention drawn away from the task of flying his aircraft. One solution to this problem would be to limit the machine's vocabulary to twenty words or less, but this might not be feasible in all situations.

One method of increasing the vocabulary size that could be used with the COMPUTALKER system would be to improve the quality of the sounds produced by the governing phonetic spelling rules. It was noted throughout the course of the experiments that the sounds produced by COMPUTALKER were not quite what was expected from the description in the phonetic rules [Fig. 12]. Most sounds were close enough that the subjects readily adjusted to them with little or no difficulty. Some letters of the alphabet were difficult to recognize, as was noted previously about the letter "S", the sound in this case being more of a hiss than the "es" sound normally associated with the letter. Also, COMPUTALKER produced soft sounds much more readily than hard ones, most notably at the end of a word. For example, the word "CHECK" is pronounced more like "cheg" by the system. Although this was a minor

problem in this instance, incorporation of the word "CHECK" in certain phrases may lead to ambiguous meanings due to its mispronunciation.

Correcting the sounds produced by the phonetic rules used by the system would involve changing the computer code which controls the COMPUTALKER hardware. These rules are contained in Sections 2 through 6 of the CSR1 Source listing. Due to time constraints, no such improvements were attempted, but are considered to be necessary to upgrade the quality of the COMPUTALKER system.

C. SURVEY

The general results of the survey, while not conclusive, indicate that most of the pilots would prefer the computer be used to calculate such things as Weight and Balance, Proven Limit of Endurance profiles (PLE), Maximum Range profiles, etc. Many felt it would be advantageous to have the computer warn the pilots of impending failures of systems or emergencies, such as slowly falling oil pressure, increasing temperatures in engine sections, etc. Several pilots favored a verbal warning from the computer, in addition to the normal indications, in the case of an approach to landing with the gear still up.

Few pilots indicated a willingness to allow the computer to actually take physical control of the aircraft under any circumstance, although a few would have allowed the computer to do so with the pilot's explicit approval. The latter would allow the computer to suggest an appropriate course of action

and, if it met with the pilot's approval, give permission to carry out the procedure. Interestingly, many pilots expressed concern that a computer system of this type would be detrimental to the morale of the Flight Engineer and cause complacency among the pilots themselves. A few pilots were even of the opinion that the autopilot, while of great benefit on long transit flights, was causing complacency and reduced flying skills among some pilots.

It would seem from the general reaction of most pilots that the initial flight testing and operational experience of a SUS be conducted at the TACCO and Navigator stations aboard the aircraft, as indicated by Ref. 4.

APPENDIX A

SPEECHLAB COMPUTER PROGRAMS

The controlling program for SPEECHLAB is written in BASIC. The SPEECH Driver requires modifications be made to the BASIC system being used. In order to save space, Hueristics, Inc., decided to use the Palo Alto TINY BASIC system, developed by Dr. Li-Chen Wang. TINY BASIC handles only interger numbers, but this is not a major problem in the SPEECHLAB program. TINY BASIC, the required modifications for the SPEECHLAB Driver, and the SPEECHLAB controlling programs are reproduced on the following pages with the permission of Hueristics, Inc.

```
*****************************************************************
*
*                  TINY BASIC FOR INTEL 8080
*                       VERSION 1.0
*                    BY LI-CHEN WANG
*                      10 JUNE, 1976
*                         @COPYLEFT
*                    ALL WRONGS RESERVED
*
*****************************************************************
*
*   *** ZERO PAGE SUBROUTINES ***
*
*   THE 8080 INSTRUCTION SET LETS YOU HAVE 8 ROUTINES IN LOW
*   MEMORY THAT MAY BE CALLED BY RST N, N BEING 0 THROUGH 7.
*   THIS IS A ONE BYTE INSTRUCTION AND HAS THE SAME POWER AS
*   THE THREE BYTE INSTRUCTION CALL LLHH.  TINY BASIC WILL
*   USE RST 0 AS START OR RESTART AND RST 1 THROUGH RST 7 FOR
*   THE SEVEN MOST FREQUENTLY USED SUBROUTINES.
*   TWO OTHER SUBROUTINES (CRLF AND TSTNUM) ARE ALSO IN THIS
*   SECTION.  THEY CAN BE REACHED ONLY BY 3-BYTE CALLS.
*
                        ORG  X'0000'
0000 F3           START DI   ,          *** START/RESTART ***
0001 310020 @@@@        LODI SP,STACK   INITIALIZE THE STACK
0004 C3BA00             JMP  ST1        GO TO THE MAIN SECTION
0007 4C                 CHAR 'L'
*
0008 E3                 XCH  HL,(SP)    *** TSTC OR RST 1 ***
0009 EF                 IGNBLK          IGNORE BLANKS AND
000A BE                 CMP  M          TEST CHARACTER
000B C36800             JMP  TC1        REST OF THIS IS AT TC1
*
000E 3E0D         CRLF  LODI A,@CR      *** CRLF ***
*
0010 F5                 PUSH AF         *** OUTC OR RST 2 ***
0011 3A0008             LD   A,OCSW     PRINT CHARACTER ONLY
0014 B7                 IOR  A          IF OCSW SWITCH IS ON
0015 C31A07             JMP  OC2        REST OF THIS IS AT OC2
*
0018 CD5504             CALL EXPR2      *** EXPR OR RST 3 ***
001B E5                 PUSH HL         EVALUATE AN EXPRESION
001C C31104             JMP  EXPR1      REST OF IT IS AT EXPR1
001F 57                 CHAR 'W'
*
0020 7C                 LOD  A,H        *** COMP OR RST 4 ***
0021 BA                 CMP  D          COMPARE HL WITH DE
0022 C0                 RET  NZ         RETURN CORRECT C AND
0023 7D                 LOD  A,L        Z FLAGS
0024 BB                 CMP  E          BUT OLD A IS LOST
0025 C9                 RET  U
0026 414E               CHAR 'AN'
*
0028 1A           SS1   LD   A,(DE)     *** IGNBLK/RST 5 ***
0029 FE20               CMPI ' '        IGNORE BLANKS
002B C0                 RET  NZ         IN TEXT (WHERE DE->)
002C 13                 INC  DE         AND RETURN THE FIRST
002D C32800             JMP  SS1        NON-BLANK CHAR. IN A
*
0030 F1                 POP  AF         *** FINISH/RST 6 ***
```

```
0031 CD91Ø5              CALL PIN         CHECK END OF COMMAND
0034 C3A4Ø5              JMP  QWHAT       PRINT "WHAT?" IF WRONG
0037 47                  CHAR 'G'
*
0038 EF                  IGNBLK           *** TSTV OR RST 7 ***
0039 D64Ø                SUBI '@'         TEST VARIABLES
003B D8                  RET  C           C:NOT A VARIABLE
003C C258ØØ              JMP  NZ,TV1      NOT "@" ARRAY
003F 13                  INC  DE          IT IS THE "@" ARRAY
0040 CDFB04              CALL PARN        @ SHOULD BE FOLLOWED
0043 29                  ADD  HL,HL       BY (EXPR) AS ITS INDEX
0044 DA9FØØ              JMP  C,QHOW      IS INDEX TOO BIG?
0047 D5                  PUSH DE          WILL IT OVERWRITE
0048 EB                  XCH  HL,DE       TEXT?
0049 CD3DØ5              CALL SIZE        FIND SIZE OF FREE
004C E7                  COMP             AND CHECK THAT
004D DADØØ5              JMP  C,ASORRY    IF SO, SAY "SORRY"
0050 21ØØ1F @@@@         LODI HL,VARBGN   IF NOT, GET ADDRESS
0053 CD6ØØ5              CALL SUBDE       OF @(EXPR) AND PUT IT
0056 D1                  POP  DE          IN HL
0057 C9                  RET  U           C FLAG IS CLEARED
0058 FE1B          TV1   CMPI 27          NOT @, IS IT A TO Z?
005A 3F                  CMC  ,           IF NOT RETURN C FLAG
005B D8                  RET  C
005C 13                  INC  DE          IF A THROUGH Z
005D 21ØØ1F @@@@         LODI HL,VARBGN   COMPUTE ADDRESS OF
0060 Ø7                  ROT  L           THAT VARIABLE
0061 85                  ADD  L           AND RETURN IT IN HL
0062 6F                  LOD  L,A         WITH C FLAG CLEARED
0063 3EØØ                LODI A,Ø
0065 8C                  ADC  H
0066 67                  LOD  H,A
0067 C9                  RET  U
*
*                  TSTC  XCH  HL,(SP)     *** TSTC OR RST 1 ***
*                        IGNBLK           THIS IS AT LOC. 8
*                        CMP  M           AND THEN JMP HERE
0068 23            TC1   INC  HL          COMPARE THE BYTE THAT
0069 CA73ØØ              JMP  Z,TC2       FOLLOWS THE RST INST.
006C C5                  PUSH BC          WITH THE TEXT (DE->)
006D 4E                  LOD  C,M         IF NOT =, ADD THE 2ND
006E Ø6ØØ                LODI B,Ø         BYTE THAT FOLLOWS THE
0070 Ø9                  ADD  HL,BC       RST TO THE OLD PC
0071 C1                  POP  BC          I.E., DO A RELATIVE
0072 1B                  DEC  DE          JUMP IF NOT =
0073 13            TC2   INC  DE          IF =, SKIP THOSE BYTES
0074 23                  INC  HL          AND CONTINUE
0075 E3                  XCH  HL,(SP)
0076 C9                  RET  U
*
0077 21ØØØØ       TSTNUM LODI HL,Ø        *** TSTNUM ***
007A 44                  LOD  B,H         TEST IF THE TEXT IS
007B EF                  IGNBLK           A NUMBER
007C FE3Ø          TN1   CMPI 'Ø'         IF NOT, RETURN Ø IN
007E D8                  RET  C           B AND HL
007F FE3A                CMPI X'3A'       IF NUMBERS, CONVERT
0081 DØ                  RET  NC          TO BINARY IN HL AND
0082 3EFØ                LODI A,X'FØ'     SET B TO # OF DIGITS
0084 A4                  AND  H           IF H>255, THERE IS NO
0085 C29FØØ              JMP  NZ,QHOW     ROOM FOR NEXT DIGIT
```

```
0088 04                    INC  B          B COUNTS # OF DIGITS
0089 C5                    PUSH BC
008A 444D                  LOD  BC,HL      HL=10*HL+(NEW DIGIT)
008C 29                    ADD  HL,HL      WHERE 10* IS DONE BY
008D 29                    ADD  HL,HL      SHIFT AND ADD
008E 09                    ADD  HL,BC
008F 29                    ADD  HL,HL
0090 1A                    LD   A,(DE)     AND (DIGIT) IS FROM
0091 13                    INC  DE         STRIPPING THE ASCII
0092 E60F                  ANDI X'0F'      CODE
0094 85                    ADD  L
0095 6F                    LOD  L,A
0096 3E00                  LODI A,0
0098 8C                    ADC  H
0099 67                    LOD  H,A
009A C1                    POP  BC
009B 1A                    LD   A,(DE)     DO THIS DIGIT AFTER
009C F27C00                JMP  NS,TN1     DIGIT. S SAYS OVERFLOW
009F D5            QHOW    PUSH DE         *** ERROR: "HOW?" ***
00A0 11A600        AHOW    LODI DE,HOW
00A3 C3A805                JMP  ERROR
00A6 484F573F      HOW     CHAR 'HOW?',@CR
00AA 0D
00AB 4F4B          OK      CHAR 'OK',@CR
00AD 0D
00AE 574841543F    WHAT    CHAR 'WHAT?',@CR
00B3 0D
00B4 534F525259    SORRY   CHAR 'SORRY',@CR
00B9 0D
*
*****************************************************************
*
* *** MAIN ***
*
* THIS IS THE MAIN LOOP THAT COLLECTS THE TINY BASIC PROGRAM
* AND STORES IT IN THE MEMORY.
*
* AT START, IT PRINTS OUT "(CR)OK(CR)", AND INITIALIZES THE
* STACK AND SOME OTHER INTERNAL VARIABLES.  THEN IT PROMPTS
* ">" AND READS A LINE.  IF THE LINE STARTS WITH A NON-ZERO
* NUMBER, THIS NUMBER IS THE LINE NUMBER.  THE LINE NUMBER
* (IN 16 BIT BINARY) AND THE REST OF THE LINE (INCLUDING CR)
* IS STORED IN THE MEMORY.  IF A LINE WITH THE SAME LINE
* NUMBER IS ALREDY THERE, IT IS REPLACED BY THE NEW ONE.  IF
* THE REST OF THE LINE CONSISTS OF A CR ONLY, IT IS NOT STORED
* AND ANY EXISTING LINE WITH THE SAME LINE NUMBER IS DELETED.
*
* AFTER A LINE IS INSERTED, REPLACED, OR DELETED, THE PROGRAM
* LOOPS BACK AND ASK FOR ANOTHER LINE.  THIS LOOP WILL BE
* TERMINATED WHEN IT READS A LINE WITH ZERO OR NO LINE
* NUMBER; AND CONTROL IS TRANSFERED TO "DIRECT".
*
* TINY BASIC PROGRAM SAVE AREA STARTS AT THE MEMORY LOCATION
* LABELED "TXTBGN" AND ENDED AT "TXTEND".  WE ALWAYS FILL THIS
* AREA STARTING AT "TXTBGN", THE UNFILLED PORTION IS POINTED
* BY THE CONTENT OF A MEMORY LOCATION LABELED "TXTUNF".
*
* THE MEMORY LOCATION "CURRNT" POINTS TO THE LINE NUMBER
* THAT IS CURRENTLY BEING INTERPRETED.  WHILE WE ARE IN
* THIS LOOP OR WHILE WE ARE INTERPRETING A DIRECT COMMAND
```

```
* (SEE NEXT SECTION), "CURRNT" SHOULD POINT TO A Ø.
*
*                 START   LODI SP,STACK   THIS IS AT LOC. Ø
ØØBA CDØEØØ       ST1     CALL CRLF       AND JUMP TO HERE
ØØBD 11ABØØ               LODI DE,OK      DE->STRING
ØØCØ 97                   SUB  A          A=Ø
ØØC1 CD3CØ6               CALL PRTSTG     PRINT STRING UNTIL CR
ØØC4 21CBØØ               LODI HL,ST2+1   LITERAL Ø
ØØC7 22Ø1Ø8               ST   HL,CURRNT  CURRNT->LINE # = Ø
ØØCA 21ØØØØ       ST2     LODI HL,Ø
ØØCD 22Ø7Ø8               ST   HL,LOPVAR
ØØDØ 22Ø3Ø8               ST   HL,STKGOS
ØØD3 3E3E          ST3     LODI A,'>'      PROMPT '>' AND
ØØD5 CDD6Ø5               CALL GETLN      READ A LINE
ØØD8 D5                   PUSH DE         DE->END OF LINE
ØØD9 11371F @@@@          LODI DE,BUFFER  DE->BEGINNING OF LINE
ØØDC CD77ØØ               CALL TSTNUM     TEST IF IT IS A NUMBER
ØØDF EF                   IGNBLK
ØØEØ 7C                   LOD  A,H        HL=VALUE OF THE # OR
ØØE1 B5                   IOR  L          Ø IF NO # WAS FOUND
ØØE2 C1                   POP  BC         BC->END OF LINE
ØØE3 CAF5Ø1               JMP  Z,DIRECT
ØØE6 1B                   DEC  DE         BACKUP DE AND SAVE
ØØE7 7C                   LOD  A,H        VALUE OF LINE # THERE
ØØE8 12                   ST   A,(DE)
ØØE9 1B                   DEC  DE
ØØEA 7D                   LOD  A,L
ØØEB 12                   ST   A,(DE)
ØØEC C5D5                 PUSH BC,DE      BC,DE->BEGIN, END
ØØEE 79                   LOD  A,C
ØØEF 93                   SUB  E
ØØFØ F5                   PUSH AF         A=# OF BYTES IN LINE
ØØF1 CD14Ø6               CALL FNDLN      FIND THIS LINE IN SAVE
ØØF4 D5                   PUSH DE         AREA, DE->SAVE AREA
ØØF5 C2Ø8Ø1               JMP  NZ,ST4     NZ:NOT FOUND, INSERT
ØØF8 D5                   PUSH DE         Z:FOUND, DELETE IT
ØØF9 CD3ØØ6               CALL FNDNXT     FIND NEXT LINE
*                                         DE->NEXT LINE
ØØFC C1                   POP  BC         BC->LINE TO BE DELETED
ØØFD 2A13Ø8               LD   HL,TXTUNF  HL->UNFILLED SAVE AREA
Ø1ØØ CDBD06               CALL MVUP       MOVE UP TO DELETE
Ø1Ø3 6Ø69                 LOD  HL,BC      TXTUNF->UNFILLED AREA
Ø1Ø5 2213Ø8               ST   HL,TXTUNF  UPDATE
Ø1Ø8 C1            ST4    POP  BC         GET READY TO INSERT
Ø1Ø9 2A13Ø8               LD   HL,TXTUNF  BUT FIRST CHECK IF
Ø1ØC F1                   POP  AF         THE LENGTH OF NEW LINE
Ø1ØD E5                   PUSH HL         IS 3 (LINE # AND CR)
Ø1ØE FE03                 CMPI 3          THEN DO NOT INSERT
Ø11Ø CAØØØØ               JMP  Z,START    MUST CLEAR THE STACK
Ø113 85                   ADD  L          COMPUTE NEW TXTUNF
Ø114 6F                   LOD  L,A
Ø115 3EØØ                 LODI A,Ø
Ø117 8C                   ADC  H
Ø118 67                   LOD  H,A        HL->NEW UNFILLED AREA
Ø119 11ØØ1F @@@@          LODI DE,TXTEND  CHECK TO SEE IF THERE
Ø11C E7                   COMP            IS ENOUGH SPACE
Ø11D D2CF05               JMP  NC,QSORRY  SORRY, NO ROOM FOR IT
Ø12Ø 221303               ST   HL,TXTUNF  OK, UPDATE TXTUNF
Ø123 D1                   POP  DE         DE->OLD UNFILLED AREA
Ø124 CDC606               CALL MVDOWN
```

```
Ø127 D1E1                POP  DE,HL      DE->BEGIN, HL->END
Ø129 CDBDØ6              CALL MVUP       MOVE NEW LINE TO SAVE
Ø12C C3D3ØØ              JMP  ST3        AREA
*
*****************************************************************
*
* *** TABLES *** DIRECT *** & EXEC ***
*
* THIS SECTION OF THE CODE TESTS A STRING AGAINST A TABLE.
* WHEN A MATCH IS FOUND, CONTROL IS TRANSFERED TO THE SECTION
* OF CODE ACCORDING TO THE TABLE.
*
* AT 'EXEC', DE SHOULD POINT TO THE STRING AND HL SHOULD POINT
* TO THE TABLE-1.  AT 'DIRECT', DE SHOULD POINT TO THE STRING,
* HL WILL BE SET UP TO POINT TO TAB1-1, WHICH IS THE TABLE OF
* ALL DIRECT AND STATEMENT COMMANDS.
*
* A '.' IN THE STRING WILL TERMINATE THE TEST AND THE PARTIAL
* MATCH WILL BE CONSIDERED AS A MATCH.  E.G., 'P.', 'PR.',
* 'PRI.', 'PRIN.', OR 'PRINT' WILL ALL MATCH 'PRINT'.
*
* THE TABLE CONSISTS OF ANY NUMBER OF ITEMS.  EACH ITEM
* IS A STRING OF CHARACTERS WITH BIT 7 SET TO Ø AND
* A JUMP ADDRESS STORED HI-LOW WITH BIT 7 OF THE HIGH
* BYTE SET TO 1.
*
* END OF TABLE IS AN ITEM WITH A JUMP ADDRESS ONLY.  IF THE
* STRING DOES NOT MATCH ANY OF THE OTHER ITEMS, IT WILL
* MATCH THIS NULL ITEM AS DEFAULT.
*
Ø12F                TAB1   EQU  *          DIRECT COMMANDS
Ø12F 4C495354               ITEM 'LIST',LIST
Ø133 8261
Ø135 52554E                 ITEM 'RUN',RUN
Ø138 8233                     .
Ø13A 4E4557                 ITEM 'NEW',NEW
Ø13D 8226
Ø13F                TAB2   EQU  *          DIRECT/STATEMENT
Ø13F 4E455854               ITEM 'NEXT',NEXT
Ø143 8349
Ø145 4C4554                 ITEM 'LET',LET
Ø148 84Ø7
Ø14A 4946                   ITEM 'IF',IF
Ø14C 839A
Ø14E 474F544F               ITEM 'GOTO',GOTO
Ø152 8252
Ø154 474F535542             ITEM 'GOSUB',GOSUB
Ø159 82B1
Ø15B 52455455524E           ITEM 'RETURN',RETURN
Ø161 82D1
Ø163 52454D                 ITEM 'REM',REM
Ø166 8396
Ø168 464F52                 ITEM 'FOR',FOR
Ø16B 82EA
Ø16D 494E5Ø5554             ITEM 'INPUT',INPUT
Ø172 83B1
Ø174 5Ø52494E54             ITEM 'PRINT',PRINT
Ø179 8279
Ø17B 53544F5Ø               ITEM 'STOP',STOP
Ø17F 822F
```

```
0181 84Ø1                  ITEM ,DEFLT
0183 594F552Ø4D415920      CHAR 'YOU MAY INSERT  MORE COMMANDS.'
018B 494E5345525420Ø2Ø
0193 4D4F52452Ø434F4D
019B 4D414E44532E
01A1                 TAB4  EQU  *          FUNCTIONS
01A1 524E44                ITEM 'RND',RND
01A4 85Ø6
01A6 414253                ITEM 'ABS',ABS
01A9 8531
01AB 53495A45              ITEM 'SIZE',SIZE
01AF 853D
01B1 84EC                  ITEM ,XP4Ø
01B3 594F552Ø4D415920      CHAR 'YOU MAY INSERT  MORE FUNCTIONS'
01BB 494E5345525420Ø2Ø
01C3 4D4F52452Ø46554E
01CB 4354494F4E53
01D1                 TAB5  EQU  *          "TO" IN "FOR"
01D1 544F                  ITEM 'TO',FR1
01D3 82FA
01D5 85A4                  ITEM ,QWHAT
01D7                 TAB6  EQU  *          "STEP" IN "FOR"
01D7 53544550              ITEM 'STEP',FR2
01DB 83Ø4
01DD 83Ø8                  ITEM ,FR3
01DF                 TAB8  EQU  *          RELATION OPERATORS
01DF 3E3D                  ITEM '>=',XP11
01E1 8417
01E3 23                    ITEM '#',XP12
01E4 841D
01E6 3E                    ITEM '>',XP13
01E7 8423
01E9 3D                    ITEM '=',XP15
01EA 8432
01EC 3C3D                  ITEM '<=',XP14
01EE 842A
01FØ 3C                    ITEM '<',XP16
01F1 8438
01F3 843E                  ITEM ,XP17
*
01F5 212E01          DIRECT LODI HL,TAB1-1 *** DIRECT ***
*
01F8                 EXEC  EQU  *          *** EXEC ***
01F8 EF              EXØ   IGNBLK          IGNORE LEADING BLANKS
01F9 D5                    PUSH DE         SAVE POINTER
01FA 1A              EX1   LD   A,(DE)     IF FOUND '.' IN STRING
01FB 13                    INC  DE         BEFORE ANY MISMATCH
01FC FE2E                  CMPI '.'        WE DECLARE A MATCH
01FE CA17Ø2                JMP  Z,EX3
Ø2Ø1 23                    INC  HL         HL->TABLE
Ø2Ø2 BE                    CMP  M          IF MATCH, TEST NEXT
Ø2Ø3 CAFAØ1                JMP  Z,EX1
Ø2Ø6 3E7F                  LODI A,X'7F'    ELSE, SEE IF BIT 7
Ø2Ø8 1B                    DEC  DE         OF TABLE IS SET, WHICH
Ø2Ø9 BE                    CMP  M          IS THE JUMP ADDR. (HI)
Ø2ØA DA1EØ2                JMP  C,EX5      C:YES, MATCHED
Ø2ØD 23              EX2   INC  HL         NC:NO, FIND JUMP ADDR.
Ø2ØE BE                    CMP  M
Ø2ØF D2ØDØ2                JMP  NC,EX2
Ø212 23                    INC  HL         P'TR TO NEXT TAB. ITEM
```

```
0213 D1                  POP   DE          RESTORE STRING POINTER
0214 C3F801              JMP   EX0         TEST AGAINST NEXT ITEM
0217 3E7F          EX3   LODI  A,X'7F'     PARTIAL MATCH, FIND
0219 23            EX4   INC   HL          JUMP ADDR., WHICH IS
021A BE                  CMP   M           FLAGGED BY BIT 7
021B D21902              JMP   NC,EX4
021E 7E            EX5   LOD   A,M         LOAD HL WITH THE JUMP
021F 23                  INC   HL          ADDRESS FROM THE TABLE
0220 6E                  LOD   L,M
0221 E67F                ANDI  X'7F'       MASK OFF BIT 7
0223 67                  LOD   H,A
0224 F1                  POP   AF          CLEAN UP THE GABAGE
0225 E9                  JMP   (HL)        AND WE GO DO IT
*
*****************************************************************
*
* WHAT FOLLOWS IS THE CODE TO EXECUTE DIRECT AND STATEMENT
* COMMANDS.  CONTROL IS TRANSFERED TO THESE POINTS VIA THE
* COMMAND TABLE LOOKUP CODE OF 'DIRECT' AND 'EXEC' IN LAST
* SECTION.  AFTER THE COMMAND IS EXECUTED, CONTROL IS
* TRANSFERED TO OTHER SECTIONS AS FOLLOWS:
*
* FOR 'LIST', 'NEW', AND 'STOP': GO BACK TO 'START'
* FOR 'RUN': GO EXECUTE THE FIRST STORED LINE IF ANY; ELSE
* GO BACK TO 'START'.
* FOR 'GOTO' AND 'GOSUB': GO EXECUTE THE TARGET LINE.
* FOR 'RETURN' AND 'NEXT': GO BACK TO SAVED RETURN LINE.
* FOR ALL OTHERS: IF 'CURRNT' -> 0, GO TO 'START', ELSE
* GO EXECUTE NEXT COMMAND.  (THIS IS DONE IN 'FINISH'.)
*
*****************************************************************
*
* *** NEW *** STOP *** RUN (& FRIENDS) *** & GOTO ***
*
* 'NEW(CR)' SETS 'TXTUNF' TO POINT TO 'TXTBGN'
*
* 'STOP(CR)' GOES BACK TO 'START'
*
* 'RUN(CR)' FINDS THE FIRST STORED LINE, STORE ITS ADDRESS (IN
* 'CURRNT'), AND START EXECUTE IT.  NOTE THAT ONLY THOSE
* COMMANDS IN TAB2 ARE LEGAL FOR STORED PROGRAM.
*
* THERE ARE 3 MORE ENTRIES IN 'RUN':
* 'RUNNXL' FINDS NEXT LINE, STORES ITS ADDR. AND EXECUTES IT.
* 'RUNTSL' STORES THE ADDRESS OF THIS LINE AND EXECUTES IT.
* 'RUNSML' CONTINUES THE EXECUTION ON SAME LINE.
*
* 'GOTO EXPR(CR)' EVALUATES THE EXPRESSION, FIND THE TARGET
* LINE, AND JUMP TO 'RUNTSL' TO DO IT.
*
0226 CDA005        NEW   CALL  ENDCHK      *** NEW(CR) ***
0229 211503              LODI  HL,TXTBGN
022C 221308              ST    HL,TXTUNF
*
022F CDA005        STOP  CALL  ENDCHK      *** STOP(CR) ***
0232 C7                  RSTART
*
0233 CDA005        RUN   CALL  ENDCHK      *** RUN(CR) ***
0236 111503              LODI  DE,TXTBGN   FIRST SAVED LINE
*
```

```
0239 210000       RUNNXL LODI HL,0       *** RUNNXL ***
023C CD1C06              CALL FNDLNP     FIND WHATEVER LINE #
023F DA0000              JMP  C,START    C:PASSED TXTUNF, QUIT
*
0242 EB           RUNTSL XCH  HL,DE      *** RUNTSL ***
0243 220108              ST   HL,CURRNT SET 'CURRNT'->LINE #
0246 EB                  XCH  HL,DE
0247 1313                INC  DE,2      BUMP PASS LINE #
*
0249 CD3207       RUNSML CALL CHKIO      *** RUNSML ***
024C 213E01              LODI HL,TAB2-1 FIND COMMAND IN TAB2
024F C3F801              JMP      EXEC  AND EXECUTE IT
*
0252 DF           GOTO   EXPR            *** GOTO EXPR ***
0253 D5                  PUSH DE        SAVE FOR ERROR ROUTINE
0254 CDA005              CALL ENDCHK    MUST FIND A CR
0257 CD1406              CALL FNDLN     FIND THE TARGET LINE
025A C2A000              JMP  NZ,AHOW   NO SUCH LINE #
025D F1                  POP  AF        CLEAR THE "PUSH DE"
025E C34202              JMP  RUNTSL    GO DO IT
*
*****************************************************************
*
* *** LIST *** & PRINT ***
*
* LIST HAS TWO FORMS:
* 'LIST(CR)' LISTS ALL SAVED LINES
* 'LIST #(CR)' START LIST AT THIS LINE #
* YOU CAN STOP THE LISTING BY CONTROL C KEY
*
* PRINT COMMAND IS 'PRINT ....;' OR 'PRINT ....(CR)'
* WHERE '....' IS A LIST OF EXPRESIONS, FORMATS, BACK-
* ARROWS, AND STRINGS.  THESE ITEMS ARE SEPERATED BY COMMAS.
*
* A FORMAT IS A POUND SIGN FOLLOWED BY A NUMBER.  IT CONTROLS
* THE NUMBER OF SPACES THE VALUE OF A EXPRESION IS GOING TO
* BE PRINTED.  IT STAYS EFFECTIVE FOR THE REST OF THE PRINT
* COMMAND UNLESS CHANGED BY ANOTHER FORMAT.  IF NO FORMAT IS
* SPECIFIED, 6 POSITIONS WILL BE USED.
*
* A STRING IS QUOTED IN A PAIR OF SINGLE QUOTES OR A PAIR OF
* DOUBLE QUOTES.
*
* A BACK-ARROW MEANS GENERATE A (CR) WITHOUT (LF)
*
* A (CRLF) IS GENERATED AFTER THE ENTIRE LIST HAS BEEN
* PRINTED OR IF THE LIST IS A NULL LIST.  HOWEVER IF THE LIST
* ENDED WITH A COMMA, NO (CRLF) IS GENERATED.
*
0261 CD7700       LIST   CALL TSTNUM     TEST IF THERE IS A #
0264 CDA005              CALL ENDCHK     IF NO # WE GET A 0
0267 CD1406              CALL FNDLN      FIND THIS OR NEXT LINE
026A DA0000       LS1    JMP  C,START    C:PASSED TXTUNF
026D CDAA06              CALL PRTLN      PRINT THE LINE
0270 CD3207              CALL CHKIO      STOP IF HIT CONTROL-C
0273 CD1C06              CALL FNDLNP     FIND NEXT LINE
0276 C36A02              JMP  LS1        AND LOOP BACK
*
0279 0E06         PRINT  LODI C,6        C = # OF SPACES
027B CF                  TSTC ';',PR2    IF NULL LIST & ";"
```

```
027C 3B
027D 06
027E CD0E00             CALL CRLF       GIVE CR-LF AND
0231 C34902             JMP  RUNSML     CONTINUE SAME LINE
0284 CF          PR2    TSTC @CR,PR0    IF NULL LIST (CR)
0285 0D
0286 06
0287 CD0E00             CALL CRLF       ALSO GIVE CR-LF AND
028A C33902             JMP  RUNNXL     GO TO NEXT LINE
028D CF          PR0    TSTC '#',PR1    ELSE IS IT FORMAT?
028E 23
028F 05
0290 DF                 EXPR            YES, EVALUATE EXPR.
0291 4D                 LOD  C,L        AND SAVE IT IN C
0292 C39B02             JMP  PR3        LOOK FOR MORE TO PRINT
0295 CD4806      PR1    CALL QTSTG      OR IS IT A STRING?
0298 C3A802             JMP  PR8        IF NOT, MUST BE EXPR.
029B CF          PR3    TSTC ',',PR6    IF ",", GO FIND NEXT
029C 2C
029D 06
029E CD9105             CALL FIN        IN THE LIST.
02A1 C38D02             JMP  PR0        LIST CONTINUES
02A4 CD0E00      PR6    CALL CRLF       LIST ENDS
02A7 F7                 FINISH
02A8 DF          PR8    EXPR            EVALUATE THE EXPR
02A9 C5                 PUSH BC
02AA CD6E06             CALL PRTNUM     PRINT THE VALUE
02AD C1                 POP  BC
02AE C39B02             JMP  PR3        MORE TO PRINT?
*
*****************************************************************
*
* *** GOSUB *** & RETURN ***
*
* 'GOSUB EXPR;' OR 'GOSUB EXPR (CR)' IS LIKE THE 'GOTO'
* COMMAND, EXCEPT THAT THE CURRENT TEXT POINTER, STACK POINTER
* ETC. ARE SAVE SO THAT EXECUTION CAN BE CONTINUED AFTER THE
* SUBROUTINE 'RETURN'.  IN ORDER THAT 'GOSUB' CAN BE NESTED
* (AND EVEN RECURSIVE), THE SAVE AREA MUST BE STACKED.
* THE STACK POINTER IS SAVED IN 'STKGOS'. THE OLD 'STKGOS' IS
* SAVED IN THE STACK.  IF WE ARE IN THE MAIN ROUTINE, 'STKGOS'
* IS ZERO (THIS WAS DONE BY THE "MAIN" SECTION OF THE CODE),
* BUT WE STILL SAVE IT AS A FLAG FOR NO FURTHER 'RETURN'S.
*
* 'RETURN(CR)' UNDOS EVERYTHING THAT 'GOSUB' DID, AND THUS
* RETURN THE EXCUTION TO THE COMMAND AFTER THE MOST RECENT
* 'GOSUB'.  IF 'STKGOS' IS ZERO, IT INDICATES THAT WE
* NEVER HAD A 'GOSUB' AND IS THUS AN ERROR.
*
02B1 CDF106      GOSUB  CALL PUSHA      SAVE THE CURRENT "FOR"
02B4 DF                 EXPR            PARAMETERS
02B5 D5                 PUSH DE         AND TEXT POINTER
02B6 CD1406             CALL FNDLN      FIND THE TARGET LINE
02B9 C2A000             JMP  NZ,AHOW    NOT THERE. SAY "HOW?"
02BC 2A0108             LD   HL,CURRNT  FOUND IT, SAVE OLD
02BF E5                 PUSH HL         'CURRNT' OLD 'STKGOS'
02C0 2A0308             LD   HL,STKGOS
02C3 E5                 PUSH HL
02C4 210000             LODI HL,0       AND LOAD NEW ONES
02C7 220708             ST   HL,LOPVAR
```

```
Ø2CA 39                 ADD   HL,SP
Ø2CB 22Ø3Ø8             ST    HL,STKGOS
Ø2CE C342Ø2             JMP   RUNTSL      THEN RUN THAT LINE
Ø2D1 CDAØØ5       RETURN CALL   ENDCHK     THERE MUST BE A CR
Ø2D4 2AØ3Ø8             LD    HL,STKGOS   OLD STACK POINTER
Ø2D7 7C                 LOD   A,H         Ø MEANS NOT EXIST
Ø2D8 B5                 IOR   L
Ø2D9 CAA4Ø5             JMP   Z,QWHAT     SO, WE SAY: "WHAT?"
Ø2DC F9                 LOD   SP,HL       ELSE, RESTORE IT
Ø2DD E1                 POP   HL
Ø2DE 22Ø3Ø8             ST    HL,STKGOS   AND THE OLD 'STKGOS'
Ø2E1 E1                 POP   HL
Ø2E2 22Ø1Ø8             ST    HL,CURRNT   AND THE OLD 'CURRNT'
Ø2E5 D1                 POP   DE          OLD TEXT POINTER
Ø2E6 CDD5Ø6             CALL  POPA        OLD "FOR" PARAMETERS
Ø2E9 F7                 FINISH            AND WE ARE BACK HOME
*
*****************************************************************
*
* *** FOR *** & NEXT ***
*
* 'FOR' HAS TWO FORMS:
* 'FOR VAR=EXP1 TO EXP2 STEP EXP1' AND 'FOR VAR=EXP1 TO EXP2'
* THE SECOND PORM MEANS THE SAME THING AS THE FIRST FORM WITH
* EXP1=1.  (I.E., WITH A STEP OF +1.)
* TBI WILL FIND THE VARIABLE VAR. AND SET ITS VALUE TO THE
* CURRENT VALUE OF EXP1.  IT ALSO EVALUATES EXPR2 AND EXP1
* AND SAVE ALL THESE TOGETHER WITH THE TEXT POINTER ETC. IN
* THE 'FOR' SAVE AREA, WHICH CONSISTS OF 'LOPVAR', 'LOPINC',
* 'LOPLMT', 'LOPLN', AND 'LOPPT'.  IF THERE IS ALREADY SOME-
* THING IN THE SAVE AREA (THIS IS INDICATED BY A NON-ZERO
* 'LOPVAR'), THEN THE OLD SAVE AREA IS SAVED IN THE STACK
* BEFORE THE NEW ONE OVERWRITES IT.
* TBI WILL THEN DIG IN THE STACK AND FIND OUT IF THIS SAME
* VARIABLE WAS USED IN ANOTHER CURRENTLY ACTIVE 'FOR' LOOP.
* IF THAT IS THE CASE, THEN THE OLD 'FOR' LOOP IS DEACTIVATED.
* (PURGED FROM THE STACK..)
*
* 'NEXT VAR' SERVES AS THE LOGICAL (NOT NECESSARILLY PHYSICAL)
* END OF THE 'FOR' LOOP.  THE CONTROL VARIABLE VAR. IS CHECKED
* WITH THE 'LOPVAR'.  IF THEY ARE NOT THE SAME, TBI DIGS IN
* THE STACK TO FIND THE RIGHT ONE AND PURGES ALL THOSE THAT
* DID NOT MATCH.  EITHER WAY, TBI THEN ADDS THE 'STEP' TO
* THAT VARIABLE AND CHECK THE RESULT WITH THE LIMIT.  IF IT
* IS WITHIN THE LIMIT, CONTROL LOOPS BACK TO THE COMMAND
* FOLLOWING THE 'FOR'.  IF OUTSIDE THE LIMIT, THE SAVE ARER
* IS PURGED AND EXECUTION CONTINUES.
*
Ø2EA CDF1Ø6       FOR   CALL  PUSHA       SAVE THE OLD SAVE AREA
Ø2ED CD7EØ5             CALL  SETVAL      SET THE CONTROL VAR.
Ø2FØ 2B                 DEC   HL          HL IS ITS ADDRESS
Ø2F1 22Ø7Ø8              ST   HL,LOPVAR   SAVE THAT
Ø2F4 21DØØ1             LODI  HL,TAB5-1   USE 'EXEC' TO LOOK
Ø2F7 C3F8Ø1             JMP   EXEC        FOR THE WORD 'TO'
Ø2FA DF           FR1   EXPR              EVALUATE THE LIMIT
Ø2FB 22ØBØ8             ST    HL,LOPLMT   SAVE THAT
Ø2FE 21D6Ø1             LODI  HL,TAB6-1   USE 'EXEC' TO LOOK
Ø3Ø1 C3F8Ø1             JMP   EXEC        FOR THE WORD 'STEP'
Ø3Ø4 DF           FR2   EXPR              FOUND IT, GET STEP
Ø3Ø5 C3ØBØ3             JMP   FR4
```

```
0308 2101ØØ        FR3     LODI HL,1       NOT FOUND, SET TO 1
030B 2209Ø8        FR4     ST   HL,LOPINC  SAVE THAT TOO
030E 2A0108        FR5     LD   HL,CURRNT  SAVE CURRENT LINE #
0311 220D08                ST   HL,LOPLN
0314 EB                    XCH  HL,DE      AND TEXT POINTER
0315 220F08                ST   HL,LOPPT
0318 010A00                LODI BC,10      DIG INTO STACK TO
031B 2A0708                LD   HL,LOPVAR  FIND 'LOPVAR'
031E EB                    XCH  HL,DE
031F 60                    LOD  H,B
0320 68                    LOD  L,B        HL=0 NOW
0321 39                    ADD  HL,SP      HERE IS THE STACK
0322 3E                    SKIP
0323 09            FR7     ADD  HL,BC      EACH LEVEL IS 10 DEEP
0324 7E                    LOD  A,M        GET THAT OLD 'LOPVAR'
0325 23                    INC  HL
0326 B6                    IOR  M
0327 CA4403                JMP  Z,FR8      0 SAYS NO MORE IN IT
032A 7E                    LOD  A,M
032B 2B                    DEC  HL
032C BA                    CMP  D          SAME AS THIS ONE?
032D C22303                JMP  NZ,FR7
0330 7E                    LOD  A,M        THE OTHER HALF?
0331 BB                    CMP  E
0332 C22303                JMP  NZ,FR7
0335 EB                    XCH  HL,DE      YES, FOUND ONE
0336 210000                LODI HL,0
0339 39                    ADD  HL,SP      TRY TO MOVE SP
033A 444D                  LOD  BC,HL
033C 210A00                LODI HL,10
033F 19                    ADD  HL,DE
0340 CDC606                CALL MVDOWN     AND PURGE 10 WORDS
0343 F9                    LOD  SP,HL      IN THE STACK
0344 2A0F08        FR8     LD   HL,LOPPT   JOB DONE, RESTORE DE
0347 EB                    XCH  HL,DE
0348 F7                    FINISH          AND CONTINUE
*
0349 FF            NEXT    TSTV            GET ADDRESS OF VAR.
034A DAA405                JMP  C,QWHAT    NO VARIABLE, "WHAT?"
034D 220508                ST   HL,VARNXT  YES, SAVE IT
0350 D5            NX0     PUSH DE         SAVE TEXT POINTER
0351 EB                    XCH  HL,DE
0352 2A0708                LD   HL,LOPVAR  GET VAR. IN 'FOR'
0355 7C                    LOD  A,H
0356 B5                    IOR  L          0 SAYS NEVER HAD ONE
0357 CAA505                JMP  Z,AWHAT    SO WE ASK: "WHAT?"
035A E7                    COMP            ELSE WE CHECK THEM
035B CA6803                JMP  Z,NX3      OK, THEY AGREE
035E D1                    POP  DE         NO, LET'S SEE
035F CDD506                CALL POPA       PURGE CURRENT LOOP
0362 2A0508                LD   HL,VARNXT  AND POP ONE LEVEL
0365 C35003                JMP  NX0        GO CHECK AGAIN
0368 5E            NX3     LOD  E,M        COME HERE WHEN AGREED
0369 23                    INC  HL
036A 56                    LOD  D,M        DE=VALUE OF VAR.
036B 2A0908                LD   HL,LOPINC
036E E5                    PUSH HL
036F 19                    ADD  HL,DE      ADD ONE STEP
0370 EB                    XCH  HL,DE
0371 2A0708                LD   HL,LOPVAR  PUT IT BACK
```

```
Ø374 73                  LOD  M,E
Ø375 23                  INC  HL
Ø376 72                  LOD  M,D
Ø377 2AØBØ8              LD   HL,LOPLMT HL->LIMIT
Ø37A F1                  POP  AF        OLD HL
Ø37B B7                  IOR  A
Ø37C F28ØØ3              JMP  NS,NX1    STEP > Ø
Ø37F EB                  XCH  HL,DE     STEP < Ø
Ø38Ø CD76Ø5       NX1    CALL CKHLDE    COMPARE WITH LIMIT
Ø383 D1                  POP  DE        RESTORE TEXT POINTER
Ø384 DA92Ø3              JMP  C,NX2     OUTSIDE LIMIT
*Ø387 2AØDØ8)            LD   HL,LOPLN  WITHIN LIMIT, GO
Ø38A 22Ø1Ø8              ST   HL,CURRNT BACK TO THE SAVED
Ø38D 2AØFØ8              LD   HL,LOPPT  'CURRNT' AND TEXT
Ø39Ø EB                  XCH  HL,DE     POINTER
Ø391 F7                  FINISH
Ø392 CDD5Ø6       NX2    CALL POPA      PURGE THIS LOOP
Ø395 F7                  FINISH
*
**********************************************************************
*
* *** REM *** IF *** INPUT *** & LET (& DEFLT) ***
*
* 'REM' CAN BE FOLLOWED BY ANYTHING AND IS IGNORED BY TBI.
* TBI TREATS IT LIKE AN 'IF' WITH A FALSE CONDITION.
*
* 'IF' IS FOLLOWED BY AN EXPR. AS A CONDITION AND ONE OR MORE
* COMMANDS (INCLUDING OUTHER 'IF'S) SEPERATED BY SEMI-COLONS.
* NOTE THAT THE WORD 'THEN' IS NOT USED.  TBI EVALUATES THE
* EXPR. IF IT IS NON-ZERO, EXECUTION CONTINUES.  IF THE
* EXPR. IS ZERO, THE COMMANDS THAT FOLLOWS ARE IGNORED AND
* EXECUTION CONTINUES AT THE NEXT LINE.
*
* 'INPUT' COMMAND IS LIKE THE 'PRINT' COMMAND, AND IS FOLLOWED
* BY A LIST OF ITEMS.  IF THE ITEM IS A STRING IN SINGLE OR
* DOUBLE QUOTES, OR IS A BACK-ARROW, IT HAS THE SAME EFFECT AS
* IN 'PRINT'.  IF AN ITEM IS A VARIABLE, THIS VARIABLE NAME IS
* PRINTED OUT FOLLOWED BY A COLON.  THEN TBI WAITS FOR AN
* EXPR. TO BE TYPED IN.  THE VARIABLE IS THEN SET TO THE
* VALUE OF THIS EXPR.  IF THE VARIABLE IS PROCEDED BY A STRING
* (AGAIN IN SINGLE OR DOUBLE QUOTES), THE STRING WILL BE
* PRINTED FOLLOWED BY A COLON.  TBI THEN WAITS FOR INPUT EXPR.
* AND SET THE VARIABLE TO THE VALUE OF THE EXPR.
*
* IF THE INPUT EXPR. IS INVALID, TBI WILL PRINT "WHAT?",
* "HOW?" OR "SORRY" AND REPRINT THE PROMPT AND REDO THE INPUT.
* THE EXECUTION WILL NOT TERMINATE UNLESS YOU TYPE CONTROL-C.
* THIS IS HANDLED IN 'INPERR'.
*
* 'LET' IS FOLLOWED BY A LIST OF ITEMS SEPERATED BY COMMAS.
* EACH ITEM CONSISTS OF A VARIABLE, AN EQUAL SIGN, AND AN EXPR.
* TBI EVALUATES THE EXPR. AND SET THE VARIBLE TO THAT VALUE.
* TBI WILL ALSO HANDLE 'LET' COMMAND WITHOUT THE WORD 'LET'.
* THIS IS DONE BY 'DEFLT'.
*
Ø396 21ØØØØ       REM    LODI HL,Ø      *** REM ***
Ø399 3E                  SKIP           THIS IS LIKE 'IF Ø'
*
Ø39A DF           IF     EXPR           *** IF ***
Ø39B 7C                  LOD  A,H       IS THE EXPR.=Ø?
```

```
039C B5                    IOR  L
039D C24902                JMP  NZ,RUNSML NO, CONTINUE
03A0 CD3206                CALL FNDSKP    YES, SKIP REST OF LINE
03A3 D24202                JMP  NC,RUNTSL AND RUN THE NEXT LINE
03A6 C7                    RSTART         IF NO NEXT, RE-START
*
03A7 2A0508        INPERR  LD   HL,STKINP *** INPERR ***
03AA F9                    LOD  SP,HL     RESTORE OLD SP
03AB E1                    POP  HL        AND OLD 'CURRNT'
03AC 220108                ST   HL,CURRNT
03AF D1                    POP  DE        AND OLD TEXT POINTER
03B0 D1                    POP  DE        REDO INPUT
*
03B1               INPUT   EQU  *         *** INPUT ***
03B1 D5            IP1     PUSH DE        SAVE IN CASE OF ERROR
03B2 CD4806                CALL QTSTG     IS NEXT ITEM A STRING?
03B5 C3BF03                JMP  IP2       NO
03B8 FF                    TSTV           YES. BUT FOLLOWED BY A
03B9 DAF903                JMP  C,IP4     VARIABLE?   NO.
03BC C3CF03                JMP  IP3       YES.  INPUT VARIABLE
03BF D5            IP2     PUSH DE        SAVE FOR 'PRTSTG'
03C0 FF                    TSTV           MUST BE VARIABLE NOW
03C1 DAA405                JMP  C,QWHAT   "WHAT?" IT IS NOT?
03C4 1A                    LD   A,(DE)    GET READY FOR 'PRTSTG'
03C5 4F                    LOD  C,A
03C6 97                    SUB  A
03C7 12                    ST   A,(DE)
03C8 D1                    POP  DE
03C9 CD3C06                CALL PRTSTG    PRINT STRING AS PROMPT
03CC 79                    LOD  A,C       RESTORE TEXT
03CD 1B                    DEC  DE
03CE 12                    ST   A,(DE)
03CF D5            IP3     PUSH DE        SAVE IN CASE OF ERROR
03D0 EB                    XCH  HL,DE
03D1 2A0108                LD   HL,CURRNT ALSO SAVE 'CURRNT'
03D4 E5                    PUSH HL
03D5 21B103                LODI HL,IP1    A NEGATIVE NUMBER
03D8 220108                ST   HL,CURRNT AS A FLAG
03DB 210000                LODI HL,0      SAVE SP TOO
03DE 39                    ADD  HL,SP
03DF 220508                ST   HL,STKINP
03E2 D5                    PUSH DE        OLD HL
03E3 3E3A                  LODI A,':'     PRINT THIS TOO
03E5 CDD605                CALL GETLN     AND GET A LINE
03E8 11371F @@@@           LODI DE,BUFFER POINTS TO BUFFER
03EB DF                    EXPR           EVALUATE INPUT
03EC 000000                NOP  3         CAN BE 'CALL ENDCHK'
03EF D1                    POP  DE        OK, GET OLD HL
03F0 EB                    XCH  HL,DE
03F1 73                    LOD  M,E       SAVE VALUE IN VAR.
03F2 23                    INC  HL
03F3 72                    LOD  M,D
03F4 E1                    POP  HL        GET OLD 'CURRNT'
03F5 220108                ST   HL,CURRNT
03F8 D1                    POP  DE        AND OLD TEXT POINTER
03F9 F1            IP4     POP  AF        PURGE JUNK IN STACK
03FA CF                    TSTC ',',IP5   IS NEXT CH. ','?
03FB 2C
03FC 03
03FD C3B103                JMP  IP1       YES, MORE ITEMS.
```

```
0400 F7            IP5    FINISH
*
0401 1A            DEFLT  LD   A,(DE)     *** DEFLT ***
0402 FE0D                 CMPI @CR        EMPTY LINE IS OK
0404 CA1004               JMP  Z,LT1      ELSE IT IS 'LET'
*
0407 CD7E05        LET    CALL SETVAL     *** LET ***
040A CF                   TSTC ',',LT1    SET VALUE TO VAR.
040B 2C
040C 03
040D C30704               JMP  LET        ITEM BY ITEM
0410 F7            LT1    FINISH          UNTIL FINISH
*
*****************************************************************
*
* *** EXPR ***
*
* 'EXPR' EVALUATES ARITHMETICAL OR LOGICAL EXPRESSIONS.
* <EXPR>::=<EXPR2>
*          <EXPR2><REL.OP.><EXPR2>
* WHERE <REL.OP.> IS ONE OF THE OPERATORS IN TAB8 AND THE
* RESULT OF THESE OPERATIONS IS 1 IF TRUE AND 0 IF FALSE.
* <EXPR2>::=(+ OR -)<EXPR3>(+ OR -<EXPR3>)(....)
* WHERE () ARE OPTIONAL AND (....) ARE OPTIONAL REPEATS.
* <EXPR3>::=<EXPR4>(<* OR /><EXPR4>)(....)
* <EXPR4>::=<VARIABLE>
*           <FUNCTION>
*           (<EXPR>)
* <EXPR> IS RECURSIVE SO THAT VARIABLE '@' CAN HAVE AN <EXPR>
* AS INDEX, FUNCTIONS CAN HAVE AN <EXPR> AS ARGUMENTS, AND
* <EXPR4> CAN BE AN <EXPR> IN PARANTHESE.
*
*                  EXPR   CALL EXPR2      THIS IS AT LOC. 18
*                         PUSH HL         SAVE <EXPR2> VALUE
0411 21DE01        EXPR1  LODI HL,TAB8-1  LOOKUP REL.OP.
0414 C3F801               JMP  EXEC       GO DO IT
0417 CD4004        XP11   CALL XP18       REL.OP.">="
041A D8                   RET  C          NO, RETURN HL=0
041B 6F                   LOD  L,A        YES, RETURN HL=1
041C C9                   RET  U
041D CD4004        XP12   CALL XP18       REL.OP."#"
0420 C8                   RET  Z          FALSE, RETURN HL=0
0421 6F                   LOD  L,A        TRUE, RETURN HL=1
0422 C9                   RET  U
0423 CD4004        XP13   CALL XP18       REL.OP.">"
0426 C8                   RET  Z          FALSE
0427 D8                   RET  C          ALSO FALSE, HL=0
0428 6F                   LOD  L,A        TRUE, HL=1
0429 C9                   RET  U
042A CD4004        XP14   CALL XP18       REL.OP."<="
042D 6F                   LOD  L,A        SET HL=1
042E C8                   RET  Z          REL. TRUE, RETURN
042F D8                   RET  C
0430 6C                   LOD  L,H        ELSE SET HL=0
0431 C9                   RET  U
0432 CD4004        XP15   CALL XP18       REL.OP."="
0435 C0                   RET  NZ         FALSE, RETRUN HL=0
0436 6F                   LOD  L,A        ELSE SET HL=1
0437 C9                   RET  U
0438 CD4004        XP16   CALL XP18       REL.OP."<"
```

```
043B D0                 RET  NC          FALSE, RETURN HL-0
043C 6F                 LOD  L,A         ELSE SET HL-1
043D C9                 RET  U
043E E1           XP17  POP  HL          NOT REL.OP.
043F C9                 RET  U           RETURN HL=<EXPR2>
0440 79           XP18  LOD  A,C         SUBROUTINE FOR ALL
0441 E1C1               POP  HL,BC       REL.OP.'S
0443 E5C5               PUSH HL,BC       REVERSE TOP OF STACK
0445 4F                 LOD  C,A
0446 CD5504             CALL EXPR2       GET 2ND <EXPR2>
0449 EB                 XCH  HL,DE       VALUE IN DE NOW
044A E3                 XCH  HL,(SP)     1ST <EXPR2> IN HL
044B CD7605             CALL CKHLDE      COMPARE 1ST WITH 2ND
044E D1                 POP  DE          RESTORE TEXT POINTER
044F 210000             LODI HL,0        SET HL-0, A-1
0452 3E01               LODI A,1
0454 C9                 RET  U
*
0455 CF           EXPR2 TSTC '-',XP21    NAGATIVE SIGN?
0456 2D
0457 06
0458 210000             LODI HL,0        YES, FAKE '0-'
045B C37F04             JMP  XP26        TREAT LIKE SUBTRACT
045E CF           XP21  TSTC '+',XP22    POSITIVE SIGN? IGNORE
045F 2B
0460 00
0461 CD8904       XP22  CALL EXPR3       1ST <EXPR3>
0464 CF           XP23  TSTC '+',XP25    ADD?
0465 2B
0466 15
0467 E5                 PUSH HL          YES, SAVE VALUE
0468 CD8904             CALL EXPR3       GET 2ND <EXPR3>
046B EB           XP24  XCH  HL,DE       2ND IN DE
046C E3                 XCH  HL,(SP)     1ST IN HL
046D 7C                 LOD  A,H         COMPARE SIGN
046E AA                 XOR  D
046F 7A                 LOD  A,D
0470 19                 ADD  HL,DE
0471 D1                 POP  DE          RESTORE TEXT POINTER
0472 FA6404             JMP  S,XP23      1ST 2ND SIGN DIFFER
0475 AC                 XOR  H           1ST 2ND SIGN EQUAL
0476 F26404             JMP  NS,XP23     SO IS RESULT
0479 C39F00             JMP  OHOW        ELSE WE HAVE OVERFLOW
047C CF           XP25  TSTC '-',XP42    SUBTRACT?
047D 2D
047E 83
047F E5           XP26  PUSH HL          YES, SAVE 1ST <EXPR3>
0480 CD8904             CALL EXPR3       GET 2ND <EXPR3>
0483 CD6A05             CALL CHGSGN      NEGATE
0486 C36B04             JMP  XP24        AND ADD THEM
*
0489 CDE604       EXPR3 CALL EXPR4       GET 1ST <EXPR4>
048C CF           XP31  TSTC '*',XP34    MULTIPLY?
048D 2A
048E 2C
048F E5                 PUSH HL          YES, SAVE 1ST
0490 CDE604             CALL EXPR4       AND GET 2ND <EXPR4>
0493 0600               LODI B,0         CLEAR B FOR SIGN
0495 CD6705             CALL CHKSGN      CHECK SIGN
0498 EB                 XCH  HL,DE       2ND IN DE NOW
```

```
0499 E3                 XCH  HL,(SP)    1ST IN HL
049A CD67Ø5             CALL CHKSGN     CHECK SIGN OF 1ST
049D 7C                 LOD  A,H        IS HL > 255 ?
049E B7                 IOR  A
049F CAA8Ø4             JMP  Z,XP32     NO
04A2 7A                 LOD  A,D        YES, HOW ABOUT DE
04A3 B2                 IOR  D
04A4 EB                 XCH  HL,DE      PUT SMALLER IN HL
04A5 C2AØØØ             JMP  NZ,AHOW    ALSO >, WILL OVERFLOW
04A8 7D           XP32  LOD  A,L        THIS IS DUMB
04A9 21ØØØØ             LODI HL,Ø       CLEAR RESULT
04AC B7                 IOR  A          ADD AND COUNT
04AD CAD8Ø4             JMP  Z,XP35
04BØ 19           XP33  ADD  HL,DE
04B1 DAAØØØ             JMP  C,AHOW     OVERFLOW
04B4 3D                 DEC  A
04B5 C2BØØ4             JMP  NZ,XP33
04B8 C3D8Ø4             JMP  XP35       FINISHED
04BB CF           XP34  TSTC '/',XP42   DIVIDE?
04BC 2F
04BD 44
04BE E5                 PUSH HL         YES, SAVE 1ST <EXPR4>
04BF CDE6Ø4             CALL EXPR4      AND GET 2ND ONE
04C2 Ø6ØØ               LODI B,Ø        CLEAR B FOR SIGN
04C4 CD67Ø5             CALL CHKSGN     CHECK SIGN OF 2ND
04C7 EB                 XCH  HL,DE      PUT 2ND IN DE
04C8 E3                 XCH  HL,(SP)    GET 1ST IN HL
04C9 CD67Ø5             CALL CHKSGN     CHECK SIGN OF 1ST
04CC 7A                 LOD  A,D        DIVIDE BY Ø?
04CD B3                 IOR  E
04CE CAAØØØ             JMP  Z,AHOW     SAY "HOW?"
04D1 C5                 PUSH BC         ELSE SAVE SIGN
04D2 CD4AØ5             CALL DIVIDE     USE SUBROUTINE
04D5 6Ø69               LOD  HL,BC      RESULT IN HL NOW
04D7 C1                 POP  BC         GET SIGN BACK
04D8 D1           XP35  POP  DE         AND TEXT POINTER
04D9 7C                 LOD  A,H        HL MUST BE +
04DA B7                 IOR  A
04DB FA9FØØ             JMP  S,QHOW     ELSE IT IS OVERFLOW
04DE 78                 LOD  A,B
04DF B7                 IOR  A
04EØ FC6AØ5             CALL S,CHGSGN   CHANGE SIGN IF NEEDED
04E3 C38CØ4             JMP  XP31       LOOK FOR MORE TERMS
*
04E6 21AØØ1       EXPR4 LODI HL,TAB4-1  FIND FUNCTION IN TAB4
04E9 C3F8Ø1             JMP  EXEC       AND GO DO IT
04EC FF           XP4Ø  TSTV            NO, NOT A FUNCTION
04ED DAF5Ø4             JMP  C,XP41     NOR A VARIABLE
04FØ 7E                 LOD  A,M        VARIABLE
04F1 23                 INC  HL
04F2 66                 LOD  H,M        VALUE IN HL
04F3 6F                 LOD  L,A
04F4 C9                 RET  U
04F5 CD77ØØ       XP41  CALL TSTNUM     OR IS IT A NUMBER
04F8 78                 LOD  A,B        # OF DIGIT
04F9 B7                 IOR  A
04FA CØ                 RET  NZ         OK
04FB CF           PARN  TSTC '(',XP43   NO DIGIT, MUST BE
04FC 28
04FD Ø5
```

```
04FE DF                     EXPR            "(EXPR)"
04FF CF                     TSTC ')',XP43
0500 29
0501 01
0502 C9             XP42    RET  U
0503 C3A405         XP43    JMP  QWHAT      ELSE SAY: "WHAT?"
*
0506 CDFB04         RND     CALL PARN       *** RND(EXPR) ***
0509 7C                     LOD  A,H        EXPR MUST BE +
050A B7                     IOR  A
050B FA9F00                 JMP  S,QHOW
050E B5                     IOR  L          AND NON-ZERO
050F CA9F00                 JMP  Z,QHOW
0512 D5E5                   PUSH DE,HL      SAVE BOTH
0514 2A1108                 LD   HL,RANPNT  GET MEMORY AS RANDOM
0517 11FF07                 LODI DE,LSTROM  NUMBER
051A E7                     COMP
051B DA2105                 JMP  C,RA1      WRAP AROUND IF LAST
051E 210000                 LODI HL,START
0521 5E             RA1     LOD  E,M
0522 23                     INC  HL
0523 56                     LOD  D,M
0524 221108                 ST   HL,RANPNT
0527 E1                     POP  HL
0528 EB                     XCH  HL,DE
0529 C5                     PUSH BC
052A CD4A05                 CALL DIVIDE     RND(N)=MOD(M,N)+1
052D C1D1                   POP  BC,DE
052F 23                     INC  HL
0530 C9                     RET  U
*
0531 CDFB04         ABS     CALL PARN       *** ABS(EXPR) ***
0534 CD6705                 CALL CHKSGN     CHECK SIGN
0537 7C                     LOD  A,H        NOTE THAT -32768
0538 B4                     IOR  H          CANNOT CHANGE SIGN
0539 FA9F00                 JMP  S,QHOW     SO SAY: "HOW?"
053C C9                     RET  U
053D 2A1308         SIZE    LD   HL,TXTUNF  *** SIZE ***
0540 D5                     PUSH DE         GET THE NUMBER OF FREE
0541 EB                     XCH  HL,DE      BYTES BETWEEN 'TXTUNF'
0542 21001F @@@@            LODI HL,VARBGN  AND 'VARBGN'
0545 CD6005                 CALL SUBDE
0548 D1                     POP  DE
0549 C9                     RET  U
*
*****************************************************************
*
* *** DIVIDE *** SUBDE *** CHKSGN *** CHGSGN *** & CKHLDE ***
*
* 'DIVIDE' DIVIDES HL BY DE, RESULT IN BC, REMAINDER IN HL
*
* 'SUBDE' SUBTRACTS DE FROM HL
*
* 'CHKSGN' CHECKS SIGN OF HL.  IF +, NO CHANGE.  IF -, CHANGE
* SIGN AND FLIP SIGN OF B.
*
* 'CHGSGN' CHNGES SIGN OF HL AND B UNCONDITIONALLY.
*
* 'CKHLDE' CHECKS SIGN OF HL AND DE.  IF DIFFERENT, HL AND DE
* ARE INTERCHANGED.  IF SAME SIGN, NOT INTERCHANGED.  EITHER
```

```
* CASE, HL DE ARE THEN COMPARED TO SET THE FLAGS.
*
054A E5            DIVIDE PUSH HL          *** DIVIDE ***
054B 6C                   LOD  L,H         DIVIDE H BY DE
054C 2600                 LODI H,0
054E CD5505               CALL DV1
0551 41                   LOD  B,C         SAVE RESULT IN B
0552 7D                   LOD  A,L         (REMAINDER+L)/DE
0553 E1                   POP  HL
0554 67                   LOD  H,A
0555 0EFF          DV1    LODI C,-1        RESULT IN C
0557 0C            DV2    INC  C           DUMB ROUTINE
0558 CD6005               CALL SUBDE       DIVIDE BY SUBTRACT
055B D25705               JMP  NC,DV2      AND COUNT
055E 19                   ADD  HL,DE
055F C9                   RET  U
*
0560 7D            SUBDE  LOD  A,L         *** SUBDE ***
0561 93                   SUB  E           SUBTRACT DE FROM
0562 6F                   LOD  L,A         HL
0563 7C                   LOD  A,H
0564 9A                   SBB  D
0565 67                   LOD  H,A
0566 C9                   RET  U
*
0567 7C            CHKSGN LOD  A,H         *** CHKSGN ***
0568 B7                   IOR  A           CHECK SIGN OF HL
0569 F0                   RET  NS          IF -, CHANGE SIGN
*
056A 7C            CHGSGN LOD  A,H         *** CHGSGN ***
056B 2F                   CMA  ,           CHANGE SIGN OF HL
056C 67                   LOD  H,A
056D 7D                   LOD  A,L
056E 2F                   CMA  ,
056F 6F                   LOD  L,A
0570 23                   INC  HL
0571 78                   LOD  A,B         AND ALSO FLIP B
0572 EE80                 XORI X'80'
0574 47                   LOD  B,A
0575 C9                   RET  U
*
0576 7C            CKHLDE LOD  A,H
0577 AA                   XOR  D           SAME SIGN?
0578 F27C05               JMP  NS,CK1      YES, COMPARE
057B EB                   XCH  HL,DE       NO, XCH AND COMP
057C E7            CK1    COMP
057D C9                   RET  U
*
*****************************************************************
*
* *** SETVAL *** FIN *** ENDCHK *** & ERROR (& FRIENDS) ***
*
* "SETVAL" EXPECTS A VARIABLE, FOLLOWED BY AN EQUAL SIGN AND
* TEEN AN EXPR.  IT EVALUATES THE EXPR. AND SET THE VARIABLE
* TO THAT VALUE.
*
* "FIN" CHECKS THE END OF A COMMAND.  IF IT ENDED WITH ";",
* EXECUTION CONTINUES.  IF IT ENDED WITH A CR, IT FINDS THE
* NEXT LINE AND CONTINUE FROM THERE.
*
```

```
* "ENDCHK" CHECKS IF A COMMAND IS ENDED WITH CR.  THIS IS
* REQUIRED IN CERTAIN COMMANDS. (GOTO, RETURN, AND STOP ETC.)
*
* "ERROR" PRINTS THE STRING POINTED BY DE (AND ENDS WITH CR).
* IT THEN PRINTS THE LINE POINTED BY 'CURRNT' WITH A "?"
* INSERTED AT WHERE THE OLD TEXT POINTER (SHOULD BE ON TOP
* OF THE STACK) POINTS TO.  EXECUTION OF TB IS STOPPED
* AND TBI IS RESTARTED.  HOWEVER, IF 'CURRNT' -> ZERO
* (INDICATING A DIRECT COMMAND), THE DIRECT COMMAND IS NOT
* PRINTED.  AND IF 'CURRNT' -> NEGATIVE # (INDICATING 'INPUT'
* COMMAND, THE INPUT LINE IS NOT PRINTED AND EXECUTION IS
* NOT TERMINATED BUT CONTINUED AT 'INPERR'.
*
* RELATED TO 'ERROR' ARE THE FOLLOWING:
* 'QWHAT' SAVES TEXT POINTER IN STACK AND GET MESSAGE "WHAT?"
* 'AWHAT' JUST GET MESSAGE "WHAT?" AND JUMP TO 'ERROR'.
* 'QSORRY' AND 'ASORRY' DO SAME KIND OF THING.
* 'QHOW' AND 'AHOW' IN THE ZERO PAGE SECTION ALSO DO THIS
*
057E CF            SETVAL TSTV              *** SETVAL ***
057F DAA405                JMP  C,QWHAT     "WHAT?" NO VARIABLE
0582 E5                    PUSH HL          SAVE ADDRESS OF VAR.
0583 CF                    TSTC '=',SV1     PASS "=" SIGN
0584 3D
0585 03
0586 DF                    EXPR             EVALUATE EXPR.
0587 444D                  LOD  BC,HL       VALUE IN BC NOW
0589 E1                    POP  HL          GET ADDRESS
058A 71                    LOD  M,C         SAVE VALUE
058B 23                    INC  HL
058C 70                    LOD  M,B
058D C9                    RET  U
058E C3A405        SV1     JMP  QWHAT       NO "=" SIGN
*
0591 CF            FIN     TSTC ';',FI1     *** FIN ***
0592 3B
0593 04
0594 F1                    POP  AF          ";", PURGE RET. ADDR.
0595 C34902                JMP  RUNSML      CONTINUE SAME LINE
0598 CF            FI1     TSTC @CR,FI2     NOT ";", IS IT CR?
0599 0D
059A 04
059B F1                    POP  AF          YES, PURGE RET. ADDR.
059C C33902                JMP  RUNNXL      RUN NEXT LINE
059F C9            FI2     RET  U           ELSE RETURN TO CALLER
*
05A0 EF            ENDCHK  IGNBLK           *** ENDCHK ***
05A1 FE0D                  CMPI @CR         END WITH CR?
05A3 C8                    RET  Z           OK, ELSE SAY: "WHAT?"
*
05A4 D5            QWHAT   PUSH DE          *** QWHAT ***
05A5 11AE00        AWHAT   LODI DE,WHAT     *** AWHAT ***
05A8 97            ERROR   SUB  A           *** ERROR ***
05A9 CD3C06                CALL PRTSTG      PRINT 'WHAT?', 'HOW?'
05AC D1                    POP  DE          OR 'SORRY'
05AD 1A                    LD   A,(DE)      SAVE THE CHARACTER
05AE F5                    PUSH AF          AT WHERE OLD DE ->
05AF 97                    SUB  A           AND PUT A 0 THERE
05B0 12                    ST   A,(DE)
05B1 2A0108                LD   HL,CURRNT   GET CURRENT LINE #
```

```
Ø5B4 E5                  PUSH HL
Ø5B5 7E                  LOD  A,M        CHECK THE VALUE
Ø5B6 23                  INC  HL
Ø5B7 B6                  IOR  M
Ø5B8 D1                  POP  DE
Ø5B9 CAØØØØ              JMP  Z,START    IF ZERO, JUST RESTART
Ø5BC 7E                  LOD  A,M        IF NEGATIVE,
Ø5BD B7                  IOR  A
Ø5BE FAA7Ø3              JMP  S,INPERR   REDO INPUT
Ø5C1 CDAAØ6              CALL PRTLN      ELSE PRINT THE LINE
Ø5C4 1B                  DEC  DE         UPTO WHERE THE Ø IS
Ø5C5 F1                  POP  AF         RESTORE THE CHARACTER
Ø5C6 12                  ST   A,(DE)
Ø5C7 3E3F                LODI A,'?'      PRINT A "?"
Ø5C9 D7                  OUTC
Ø5CA 97                  SUB  A          AND THE REST OF THE
Ø5CB CD3CØ6              CALL PRTSTG     LINE
Ø5CE C7                  RSTART          THEN RESTART
Ø5CF D5          QSORRY  PUSH DE         *** QSORRY ***
Ø5DØ 11B4ØØ      ASORRY  LODI DE,SORRY   *** ASORRY ***
Ø5D3 C3A8Ø5              JMP  ERROR
*
*****************************************************************
*
* *** GETLN *** FNDLN (& FRIENDS) ***
*
* 'GETLN' READS A INPUT LINE INTO 'BUFFER'.  IT FIRST PROMPT
* THE CHARACTER IN A (GIVEN BY THE CALLER), THEN IT FILLS THE
* THE BUFFER AND ECHOS.  IT IGNORES LF'S AND NULLS, BUT STILL
* ECHOS THEM BACK.  RUB-OUT IS USED TO CAUSE IT TO DELETE
* THE LAST CHARATER (IF THERE IS ONE), AND ALT-MOD IS USED TO
* CAUSE IT TO DELETE THE WHOLE LINE AND START IT ALL OVER.
* CR SIGNALS THE END OF A LINE, AND CAUSE 'GETLN' TO RETURN.
*
* 'FNDLN' FINDS A LINE WITH A GIVEN LINE # (IN HL) IN THE
* TEXT SAVE AREA.  DE IS USED AS THE TEXT POINTER.  IF THE
* LINE IS FOUND, DE WILL POINT TO THE BEGINNING OF THAT LINE
* (I.E., THE LOW BYTE OF THE LINE #), AND FLAGS ARE NC & Z.
* IF THAT LINE IS NOT THERE AND A LINE WITH A HIGHER LINE #
* IS FOUND, DE POINTS TO THERE AND FLAGS ARE NC & NZ.  IF
* WE REACHED THE END OF TEXT SAVE ARE AND CANNOT FIND THE
* LINE, FLAGS ARE C & NZ.
* 'FNDLN' WILL INITIALIZE DE TO THE BEGINNING OF THE TEXT SAVE
* AREA TO START THE SEARCH.  SOME OTHER ENTRIES OF THIS
* ROUTINE WILL NOT INITIALIZE DE AND DO THE SEARCH.
* 'FNDLNP' WILL START WITH DE AND SEARCH FOR THE LINE #.
* 'FNDNXT' WILL BUMP DE BY 2, FIND A CR AND THEN START SEARCH.
* 'FNDSKP' USE DE TO FIND A CR, AND THEN STRART SEARCH.
*
Ø5D6 D7          GETLN   OUTC            *** GETLN ***
Ø5D7 11371F @@@@         LODI DE,BUFFER PROMPT AND INIT.
Ø5DA CD32Ø7      GL1     CALL CHKIO      CHECK KEYBOARD
Ø5DD CADAØ5              JMP  Z,GL1      NO INPUT, WAIT
Ø5EØ D7                  OUTC            INPUT, ECHO BACK
Ø5E1 FEØA                CMPI @LF        IGNORE LF
Ø5E3 CADAØ5              JMP  Z,GL1
Ø5E6 B7                  IOR  A          IGNORE NULL
Ø5E7 CADAØ5              JMP  Z,GL1
Ø5EA FE7F                CMPI @DLCH      DELETE LAST CHARACTER?
Ø5EC CAFFØ5              JMP  Z,GL3      YES
```

```
05E? FE7D               CMPI @DLLN      DELETE THE WHOLE LINE?
05F1 CA0C05             JMP  Z,GL4      YES
05F4 12                 ST   A,(DE)     ELSE, SAVE INPUT
05F5 13                 INC  DE         AND BUMP POINTER
05F6 FE0D               CMPI @CR        WAS IT CR?
05F8 C8                 RET  Z          YES, END OF LINE
05F9 7B                 LOD  A,E        ELSE, MORE FREE ROOM?
05FA FE7F               CMPI BUFEND,>
05FC C2DA05             JMP  NZ,GL1     YES, GET NEXT INPUT
05FF 7B           GL3   LOD  A,E        DELETE LAST CHARACTER
0600 FE37               CMPI BUFFER,>   BUT DO WE HAVE ANY?
0602 CA0C06             JMP  Z,GL4      NO, REDO WHOLE LINE
0605 1B                 DEC  DE         YES, BACKUP POINTER
0606 3E5C               LODI A,@BKS     AND ECHO A BACK-SLASH
0608 D7                 OUTC
0609 C3DA05             JMP  GL1        GO GET NEXT INPUT
060C CD0E00       GL4   CALL CRLF       REDO ENTIRE LINE
060F 3E5E               LODI A,@UPA     CR, LF AND UP-ARROW
0611 C3D605             JMP  GETLN
*
0614 7C           FNDLN LOD  A,H        *** FNDLN ***
0615 B7                 IOR  A          CHECK SIGN OF HL
0616 FA9F00             JMP  S,QHOW     IT CANNOT BE -
0619 111508             LODI DE,TXTBGN  INIT. TEXT POINTER
*
061C              FNDLNP EQU  *         *** FNDLNP ***
061C E5           FL1   PUSH HL         SAVE LINE #
061D 2A1308             LD   HL,TXTUNF  CHECK IF WE PASSED END
0620 2B                 DEC  HL
0621 E7                 COMP
0622 E1                 POP  HL         GET LINE # BACK
0623 D8                 RET  C          C,NZ PASSED END
0624 1A                 LD   A,(DE)     WE DID NOT, GET BYTE 1
0625 95                 SUB  L          IS THIS THE LINE?
0626 47                 LOD  B,A        COMPARE LOW ORDER
0627 13                 INC  DE
0628 1A                 LD   A,(DE)     GET BYTE 2
0629 9C                 SBB  H          COMPARE HIGH ORDER
062A DA3106             JMP  C,FL2      NO, NOT THERE YET
062D 1B                 DEC  DE         ELSE WE EITHER FOUND
062E B0                 IOR  B          IT, OR IT IS NOT THERE
062F C9                 RET  U          NC,Z:FOUND; NC,NZ:NO
*
0630              FNDNXT EQU  *         *** FNDNXT ***
0630 13                 INC  DE         FIND NEXT LINE
0631 13           FL2   INC  DE         JUST PASSED BYTE 1 & 2
*
0632 1A           FNDSKP LD    A,(DE)   *** FNDSKP ***
0633 FE0D               CMPI @CR        TRY TO FIND CR
0635 C23106             JMP  NZ,FL2     KEEP LOOKING
0638 13                 INC  DE         FOUND CR, SKIP OVER
0639 C31C06             JMP  FL1        CHECK IF END OF TEXT
*
*****************************************************************
*
* *** PRTSTG *** QTSTG *** PRTNUM *** & PRTLN ***
*
* 'PRTSTG' PRINTS A STRING POINTED BY DE.  IT STOPS PRINTING
* AND RETURNS TO CALLER WHEN EITHER A CR IS PRINTED OR WHEN
* THE NEXT BYTE IS THE SAME AS WHAT WAS IN A (GIVEN BY THE
```

```
* CALLER).  OLD A IS STORED IN B, OLD B IS LOST.
*
* 'QTSTG' LOOKS FOR A BACK-ARROW, SINGLE QUOTE, OR DOUBLE
* QUOTE.  IF NONE OF THESE, RETURN TO CALLER.  IF BACK-ARROW,
* OUTPUT A CR WITHOUT A LF.  IF SINGLE OR DOUBLE QUOTE, PRINT
* THE STRING IN THE QUOTE AND DEMANDS A MATCHING UNQUOTE.
* AFTER THE PRINTING THE NEXT 3 BYTES OF THE CALLER IS SKIPPED
* OVER (USUALLY A JUMP INSTRUCTION).
*
* 'PRTNUM' PRINTS THE NUMBER IN HL.  LEADING BLANKS ARE ADDED
* IF NEEDED TO PAD THE NUMBER OF SPACES TO THE NUMBER IN C.
* HOWEVER, IF THE NUMBER OF DIGITS IS LARGER THAN THE # IN
* C, ALL DIGITS ARE PRINTED ANYWAY.  NEGATIVE SIGN IS ALSO
* PRINTED AND COUNTED IN, POSITIVE SIGN IS NOT.
*
* 'PRTLN' PRINTS A SAVED TEXT LINE WITH LINE # AND ALL.
*
Ø63C 47          PRTSTG LOD  B,A        *** PRTSTG ***
Ø63D 1A          PS1    LD   A,(DE)     GET A CHARACTER
Ø63E 13                 INC  DE         BUMP POINTER
Ø63F B8                 CMP  B          SAME AS OLD A?
Ø64Ø C8                 RET  Z          YES, RETURN
Ø641 D7                 OUTC            ELSE PRINT IT
Ø642 FEØD               CMPI @CR        WAS IT A CR?
Ø644 C23DØ6             JMP  NZ,PS1     NO, NEXT
Ø647 C9                 RET  U          YES, RETURN
*
Ø648 CF          QTSTG  TSTC '"',QT3    *** QTSTG ***
Ø649 22
Ø64A ØF
Ø64B 3E22               LODI A,'"'      IT IS A "
Ø64D CD3CØ6      QT1    CALL PRTSTG     PRINT UNTIL ANOTHER
Ø65Ø FEØD               CMPI @CR        WAS LAST ONE A CR?
Ø652 E1                 POP  HL         RETURN ADDRESS
Ø653 CA39Ø2             JMP  Z,RUNNXL   WAS CR, RUN NEXT LINE
Ø656 232323      QT2    INC  HL,3       SKIP 3 BYTES ON RETURN
Ø659 E9                 JMP  (HL)       RETURN
Ø65A CF          QT3    TSTC @QT,QT4    IS IT A ' ?
Ø65B 27
Ø65C Ø5
Ø65D 3E27               LODI A,@QT      YES, DO SAME
Ø65F C34DØ6             JMP  QT1        AS IN "
Ø662 CF          QT4    TSTC @BKA,QT5   IS IT BACK-ARROW?
Ø663 5F
Ø664 Ø8
Ø665 3E8D               LODI A,X'8D'    YES, CR WITHOUT LF
Ø667 D7                 OUTC            DO IT TWICE TO GIVE
Ø668 D7                 OUTC            TTY ENOUGH TIME
Ø669 E1                 POP  HL         RETURN ADDRESS
Ø66A C356Ø6             JMP  QT2
Ø66D C9          QT5    RET  U          NONE OF ABOVE
*
Ø66E D5          PRTNUM PUSH DE         *** PRTNUM ***
Ø66F 11ØAØØ             LODI DE,1Ø      DECIMAL
Ø672 D5                 PUSH DE         SAVE AS A FLAG
Ø673 42                 LOD  B,D        B=SIGN
Ø674 ØD                 DEC  C          C=SPACES
Ø675 CD67Ø5             CALL CHKSGN     CHECK SIGN
Ø678 F27EØ6             JMP  NS,PN1     NO SIGN
Ø67B Ø62D               LODI B,'-'      B=SIGN
```

```
067D 0D                 DEC  C          '-' TAKES SPACE
067E C5           PN1   PUSH BC         SAVE SIGN & SPACE
067F CD4A05       PN2   CALL DIVIDE     DEVIDE HL BY 10
0682 78                 LOD  A,B        RESULT 0?
0683 B1                 IOR  C
0684 CA8F06             JMP  Z,PN3      YES, WE GOT ALL
0687 E3                 XCH  HL,(SP)    NO, SAVE REMAINDER
0688 2D                 DEC  L          AND COUNT SPACE
0689 E5                 PUSH HL         HL IS OLD BC
068A 6069               LOD  HL,BC      MOVE RESULT TO BC
068C C37F06             JMP  PN2        AND DIVIDE BY 10
068F C1           PN3   POP  BC         WE GOT ALL DIGITS IN
0690 0D           PN4   DEC  C          THE STACK
0691 79                 LOD  A,C        LOOK AT SPACE COUNT
0692 B7                 IOR  A
0693 FA9C06             JMP  S,PN5      NO LEADING BLANKS
0696 3E20               LODI A,' '      LEADING BLANKS
0698 D7                 OUTC
0699 C39006             JMP  PN4        MORE?
069C 78           PN5   LOD  A,B        PRINT SIGN
069D D7                 OUTC            MAYBE - OR NULL
069E 5D                 LOD  E,L        LAST REMAINDER IN E
069F 7B           PN6   LOD  A,E        CHECK DIGIT IN E
06A0 FE0A               CMPI 10         10 IS FLAG FOR NO MORE
06A2 D1                 POP  DE
06A3 C8                 RET  Z          IF SO, RETURN
06A4 C630               ADDI '0'        ELSE COVERT TO ASCII
06A6 D7                 OUTC            AND PRINT THE DIGIT
06A7 C39F06             JMP  PN6        GO BACK FOR MORE
*
06AA 1A           PRTLN LD   A,(DE)     *** PRTLN ***
06AB 6F                 LOD  L,A        LOW ORDER LINE #
06AC 13                 INC  DE
06AD 1A                 LD   A,(DE)     HIGH ORDER
06AE 67                 LOD  H,A
06AF 13                 INC  DE
06B0 0E04               LODI C,4        PRINT 4 DIGIT LINE #
06B2 CD6E06             CALL PRTNUM
06B5 3E20               LODI A,' '      FOLLOWED BY A BLANK
05B7 D7                 OUTC
05B8 97                 SUB  A          AND THEN THE TEXT
05B9 CD3C06             CALL PRTSTG
06BC C9                 RET  U
*
**************************************************************
*
* *** MVUP *** MVDOWN *** POPA *** & PUSHA ***
*
* 'MVUP' MOVES A BLOCK UP FROM WHERE DE-> TO WHERE BC-> UNTIL
* DE = HL
*
* 'MVDOWN' MOVES A BLOCK DOWN FROM WHERE DE-> TO WHERE HL->
* UNTIL DE = BC
*
* 'POPA' RESTORES THE 'FOR' LOOP VARIABLE SAVE AREA FROM THE
* STACK
*
* 'PUSHA' STACKS THE 'FOR' LOOP VARIABLE SAVE AREA INTO THE
* STACK
*
```

```
Ø6BD E7            MVUP    COMP             *** MVUP ***
Ø6BE C8                    RET   Z           DE = HL, RETURN
Ø6BF 1A                    LD    A,(DE)      GET ONE BYTE
Ø6CØ Ø2                    ST    A,(BC)      MOVE IT
Ø6C1 13                    INC   DE          INCREASE BOTH POINTERS
Ø6C2 Ø3                    INC   BC
Ø6C3 C3BDØ6                JMP   MVUP        UNTIL DONE
*
Ø6C6 78            MVDOWN  LOD   A,B         *** MVDOWN ***
Ø6C7 92                    SUB   D           TEST IF DE = BC
Ø6C8 C2CEØ6                JMP   NZ,MD1      NO, GO MOVE
Ø6CB 79                    LOD   A,C         MAYBE, OTHER BYTE?
Ø6CC 93                    SUB   E
Ø6CD C8                    RET   Z           YES, RETURN
Ø6CE 1B            MD1     DEC   DE          ELSE MOVE A BYTE
Ø6CF 2B                    DEC   HL          BUT FIRST DECREASE
Ø6DØ 1A                    LD    A,(DE)      BOTH POINTERS AND
Ø6D1 77                    LOD   M,A         THEN DO IT
Ø6D2 C3C6Ø6                JMP   MVDOWN      LOOP BACK
*
Ø6D5 C1            POPA    POP   BC          BC = RETURN ADDR.
Ø6D6 E1                    POP   HL          RESTORE LOPVAR, BUT
Ø6D7 22Ø7Ø8                ST    HL,LOPVAR   -Ø MEANS NO MORE
Ø6DA 7C                    LOD   A,H
Ø6DB B5                    IOR   L
Ø6DC CAEFØ6                JMP   Z,PP1       YEP, GO RETURN
Ø6DF E1                    POP   HL          NOP, RESTORE OTHERS
Ø6EØ 22Ø9Ø8                ST    HL,LOPINC
Ø6E3 E1                    POP   HL
Ø6E4 22ØBØ8                ST    HL,LOPLMT
Ø6E7 E1                    POP   HL
Ø6E8 22ØDØ8                ST    HL,LOPLN
Ø6EB E1                    POP   HL
Ø6EC 22ØFØ8                ST    HL,LOPPT
Ø6EF C5            PP1     PUSH  BC          BC = RETURN ADDR.
Ø6FØ C9                    RET   U
*
Ø6F1 21A71F @@@@   PUSHA   LODI  HL,STKLMT *** PUSHA ***
Ø6F4 CD6AØ5                CALL  CHGSGN
Ø6F7 C1                    POP   BC          BC=RETURN ADDRESS
Ø6F8 39                    ADD   HL,SP       IS STACK NEAR THE TOP?
Ø6F9 D2CFØ5                JMP   NC,QSORRY   YES, SORRY FOR THAT.
Ø6FC 2AØ7Ø8                LD    HL,LOPVAR   ELSE SAVE LOOP VAR.S
Ø6FF 7C                    LOD   A,H         BUT IF LOPVAR IS Ø
Ø7ØØ B5                    IOR   L           THAT WILL BE ALL
Ø7Ø1 CA17Ø7                JMP   Z,PU1
Ø7Ø4 2AØF Ø8               LD    HL,LOPPT    ELSE, MORE TO SAVE
Ø7Ø7 E5                    PUSH  HL
Ø7Ø8 2AØD Ø8               LD    HL,LOPLN
Ø7ØB E5                    PUSH  HL
Ø7ØC 2AØBØ8                LD    HL,LOPLMT
Ø7ØF E5                    PUSH  HL
Ø71Ø 2AØ9Ø8                LD    HL,LOPINC
Ø713 E5                    PUSH  HL
Ø714 2AØ7Ø8                LD    HL,LOPVAR
Ø717 E5            PU1     PUSH  HL
Ø718 C5                    PUSH  BC          BC = RETURN ADDR.
Ø719 C9                    RET   U
*
*****************************************************************
```

```
*
* *** OUTC *** & CHKIO ***
*
* THESE ARE THE ONLY I/O ROUTINES IN TBI.
* 'OUTC' IS CONTROLLED BY A SOFTWARE SWITCH 'OCSW'.  IF OCSW=Ø
* 'OUTC' WILL JUST RETURN TO THE CALLER.  IF OCSW IS NOT Ø,
* IT WILL OUTPUT THE BYTE IN A.  IF THAT IS A CR, A LF IS ALSO
* SEND OUT.  ONLY THE FLAGS MAY BE CHANGED AT RETURN, ALL REG.
* ARE RESTORED.
*
* 'CHKIO' CHECKS THE INPUT.  IF NO INPUT, IT WILL RETURN TO
* THE CALLER WITH THE Z FLAG SET.  IF THERE IS INPUT, Z FLAG
* IS CLEARED AND THE INPUT BYTE IS IN A.  HOWERER, IF THE
* INPUT IS A CONTROL-O, THE 'OCSW' SWITCH IS COMPLIMENTED, AND
* Z FLAG IS RETURNED.  IF A CONTROL-C IS READ, 'CHKIO' WILL
* RESTART TBI AND DO NOT RETURN TO THE CALLER.
*
*                   OUTC.   PUSH AF         THIS IS AT LOC. 1Ø
*                           LD   A,OCSW     CHECK SOFTWARE SWITCH
*                           IOR  A
Ø71A C21FØ7         OC2     JMP  NZ,OC3     IT IS ON
Ø71D F1                     POP  AF         IT IS OFF
Ø71E C9                     RET  U          RESTORE AF AND RETURN
Ø71F DBØØ           OC3     INP  Ø          COME HERE TO DO OUTPUT
Ø721 E6Ø2                   ANDI X'Ø2'      STATUS BIT
Ø723 CA1FØ7                 JMP  Z,OC3      NOT READY, WAIT
Ø726 F1                     POP  AF         READY, GET OLD A BACK
Ø727 D3Ø1                   OUT  1          AND SEND IT OUT
Ø729 FEØD                   CMPI @CR        WAS IT CR?
Ø72B CØ                     RET  NZ         NO, FINISHED
Ø72C 3EØA                   LODI A,@LF      YES, WE SEND LF TOO
Ø72E D7                     OUTC            THIS IS RECURSIVE
Ø72F 3EØD                   LODI A,@CR      GET CR BACK IN A
Ø731 C9                     RET  U
Ø732 DBØØ           CHKIO   INP  Ø          *** CHKIO ***
Ø734 ØØ                     NOP  ,          STATUS BIT FLIPPED?
Ø735 E62Ø                   ANDI X'2Ø'      MASK STATUS BIT
Ø737 C8                     RET  Z          NOT READY, RETURN "Z"
Ø738 DBØ1                   INP  1          READY, READ DATA
Ø73A E67F                   ANDI X'7F'      MASK BIT 7 OFF
Ø73C FEØF                   CMPI @CO        IS IT CONTROL-O?
Ø73E C24BØ7                 JMP  NZ,CI1     NO, MORE CHECKING
Ø741 3AØØØ8                 LD   A,OCSW     CONTROL-O FLIPS OCSW
Ø744 2F                     CMA  ,          ON TO OFF, OFF TO ON
Ø745 32ØØØ8                 ST   A,OCSW
Ø748 C332Ø7                 JMP  CHKIO      GET ANOTHER INPUT
Ø74B FE03           CI1     CMPI @CC        IS IT CONTROL-C?
Ø74D CØ                     RET  NZ         NO, RETURN "NZ"
Ø74E C7                     RSTART          YES, RESTART TBI
Ø74F 594F552Ø4D41592Ø       CHAR 'YOU MAY NEED THIS SPACE TO'
Ø757 4E4545442Ø544849
Ø75F 532Ø535Ø41434528
Ø767 544F
Ø769 5Ø4154434820555Ø       CHAR 'PATCH UP THE I/O ROUTINES,'
Ø771 2Ø544845Ø492F4F
Ø779 2Ø524F5554494E45
Ø781 532C
Ø783 544F2Ø4649582Ø55       CHAR 'TO FIX UP BUGS, OR TO ADD'
Ø78B 5Ø2Ø425547532C2Ø
Ø793 4F5228544F2Ø4144
```

```
079B 44
079C 4D4F52452043 4F4D      CHAR 'MORE COMMANDS AND FUNCTIONS.'
07A4 4D414E445320 414E
07AC 442046554E43 5449
07B4 4F4E532E
07B8 534B592028 53504 1     CHAR 'SKY (SPACE) IS THE LIMIT.'
07C0 43452920 49532054
07C8 48452049 4C494D49 54
07D0 2E
07D1 474F4F442049 5543      CHAR 'GOOD LUCK AND GOOD BYE.'
07D9 4B204 14E442047 4F
07E1 4F442042 59452E
07E8 4C4943484 54E2057      CHAR 'LICHEN WANG, 10 JUNE 76'
07F0 414E472C 20313020
07F8 4A554E45 203736
07FF                  LSTROM EQU  *          ALL ABOVE CAN BE ROM
                             ORG  X'0800'    HERE DOWN MUST BE RAM
0800 FF               OCSW   DB   X'FF'      SWITCH FOR OUTPUT
0801 0000             CURRNT DW   0          POINTS TO CURRENT LINE
0803 0000             STKGOS DW   0          SAVES SP IN 'GOSUB'
0805                  VARNXT EQU  *          TEMP STORAGE
0805 0000             STKINP DW   0          SAVES SP IN 'INPUT'
0807 0000             LOPVAR DW   0          'FOR' LOOP SAVE AREA
0809 0000             LOPINC DW   0          INCREMENT
080B 0000             LOPLMT DW   0          LIMIT
080D 0000             LOPLN  DW   0          LINE NUMBER
080F 0000             LOPPT  DW   0          TEXT POINTER
0811 0000             RANPNT DW   START      RANDOM NUMBER POINTER
0813 1508             TXTUNF DW   TXTBGN     ->UNFILLED TEXT AREA
0815 00               TXTBGN DS   1          TEXT SAVE AREA BEGINS
1F00                  TXTEND EQU  VARBGN     TEXT SAVE AREA ENDS
                             ORG  X'1F00'
1F00                  VARBGN DS   2*27       VARIABLE @(0)
1F36                         DS   XL1        EXTRA BYTE FOR BUFFER
1F37                  BUFFER DS   XL72       INPUT BUFFER
1F7F                  BUFEND EQU  *          BUFFER ENDS
1F7F                         DS   XL40       EXTRA BYTES FOR STACK
1FA7                  STKLMT EQU  *          TOP LIMIT FOR STACK
                             ORG  X'2000'
2000                  STACK  EQU  *          STACK STARTS HERE
                             END
```

```
                        ;
                        ;       MODIFY TINY BASIC TABLE TO INCLUDE THIS
                        ;       COMMAND
                        ;
0181                            ORG 0181H
0181 5350454543                 DB 'SPEECH'         ;SPEECH COMMAND
0187 874F                       DB 087H,04FH        ;ENTRY POINT TO PATCH
0189 8401                       DB 084H,01H         ;DEFAULT ENTRY POINT
                        ;
074F DF         START:  RST 3               ;EXPR
0750 22D607             SHLD SAMPS          ;NO. OF SAMPLES
0753 CF                 RST 1               ;LOOK FOR COMMA
0754 2C                 DB ','
0755 0B                 DCX B
0756 DF                 RST 3               ;EXPR
0757 22D807             SHLD PERIOD         ;SAMPLE PERIOD
075A CDA005             CALL ENDCHK         ;SHOULD BE END OF STATEMENT
075D CD6407             CALL SPCH           ;GET SPEECH DATA
0760 F7                 RST 6               ;FINISH
0761 C3A405             JMP QWHAT           ;DIDN'T FINISH COMMAND PROPERLY
0764 F5         SPCH:   PUSH PSW            ;SAVE REGISTERS
0765 C5                 PUSH B
0766 D5                 PUSH D
0767 3EFA               MVI A,250           ;PAUSE FOR TTY NOISE
0769 CDA107             CALL DELAY
076C CDC607             CALL BEEP           ;SIGNAL BEGINNING OF WINDOW
076F 3E96               MVI A,150           ;DELAY FOR BEEPER NOISE
0771 CDA107             CALL DELAY
0774 21001F             LXI H,VARBGN        ; @@@@ BASE OF @(0) CHANGE FOR LARGER
                        ;                    SYSTEMS
0777 3AD607             LDA SAMPS           ;NO. OF SAMPLES TO TAKE
077A 5F                 MOV E,A
077B 1600       SP00:   MVI D,0
077D 7A         SP0:    MOV A,D
077E FE04               CPI 4
0780 CA9007             JZ SP1              ;FINISHED 1 SAMPLE, 4 BANDS
```

```
0783 CDB507            CALL GET1       ;NO, GET ANOTHER BAND
0786 77                MOV M,A         ;STORE SAMPLE
0787 2B                DCX H           ;ARRAY STORED BACKWARDS
0788 3E00              MVI A,0         ;0 HI ORDER HALF OF WORD
078A 77                MOV M,A
078B 2B                DCX H
078C 14                INR D           ;UP BAND NO.
078D C37D07            JMP SP0
0790 3AD807    SP1:    LDA PERIOD
0793 CDA107            CALL DELAY      ;PAUSE BETWEEN SAMPLES
0796 1D                DCR E           ;SAMPLES TO GO
0797 C27B07            JNZ SP00        ;MORE?
079A CDC607            CALL BEEP       ;NO--- END WINDOW
079D D1                POP D           ;RESTORE REGISTERS
079E C1                POP B
079F F1                POP PSW
07A0 C9                RET
07A1 C5        DELAY:  PUSH B          ;SAVE B
07A2 FE00      DEL0:   CPI 0           ;FINISHED?
07A4 CAB307            JZ RETDEL
07A7 0669              MVI B,105       ;MAGIC CONSTANT
07A9 00        DEL1:   NOP             ;KILL TIME
07AA 00                NOP
07AB 05                DCR B
07AC C2A907            JNZ DEL1
07AF 3D                DCR A           ;1 MS GONE
07B0 C3A207            JMP DEL0
07B3 C1        RETDEL: POP B           ;RESTORE REGISTER B
07B4 C9                RET
               ;
```

```
07B5 F620       GET1:   ORI 020H        ;DISABLE SWITCH
07B7 D3AF               OUT 0AFH        ;DO IT
07B9 E6DF               ANI 0DFH        ;CLEAR OUT DISABLE BITS
07BB D3AF               OUT 0AFH        ;REENABLE SWITCH
07BD DBAF       GET2:   IN 0AFH         ;GET STATUS
07BF 17                 RAL
07C0 DABD07             JC GET2         ;CARRY MEANS NOT READY
07C3 DBAF               IN 0AFH         ;GET DATA
07C5 C9                 RET

07C6 F5         BEEP:   PUSH PSW
07C7 3E10               MVI A,010H      ;TURN ON BEEPER
07C9 D3AF               OUT 0AFH
07CB 3E64               MVI A,100       ;LEAVE IT ON 100MS
07CD CDA107             CALL DELAY
07D0 3E00               MVI A,0         ;TURN BEEPER OFF
07D2 D3AF               OUT 0AFH
07D4 F1                 POP PSW
07D5 C9                 RET
                ;

07D6            SAMPS:  DS 2            ;NO. OF SAMPLES
07D8            PERIOD: DS 2            ;DELAY BETWEEN SAMPLES
                ;
```

```
 10 REM ENHANCED SPEECH RECOGNITION PROGRAM
 20 REM --- PROGRAM 1 ---
 25 REM --- MOD ALLOWS USER TO KEEP WORD TABLES
 26 REM --- AS DESIRED.
 30 V=10
 31 INPUT "CLEAR TABLES?;1 FOR YES,2 FOR NO"X
 32 IF X = 2 GOTO 100
 35 REM CLEAR TABLES
 40 FOR I=0 TO 64*V-1
 50 @(700+I)=0
 60 NEXT I
100 T=6
120 INPUT "1=T,2=P"X
140 IF X # 1 GOTO 170
150 GOSUB 1000
160 GOTO 120
170 IF X # 2 GOTO 200
180 GOSUB 2000
190 PRINT "RECOGNIZED:",N
200 GOTO 120
210 PRINT "?"
220 GOTO 120
1000 REM TRAINING ROUTINE
1010 INPUT "NUMBER"N
1015 IF N>V PRINT "?" GOTO 1010
1020 REM GET SPEECH
1030 GOSUB 3000
1040 REM GET 64 PARAMETERS
1050 S=64
1060 FOR J=0 TO S-1
1070 @(700+N*S+J)=@(600+J)
1080 NEXT J
1090 RETURN
2000 REM PERFORMANCE ROUTINE
2010 GOSUB 3000
2020 FOR I=664 TO 663+V
2040 @(I)=0
2050 NEXT I
2060 FOR I=0 TO V-1
2070 FOR J=0 TO S-1
2080 X=ABS(@(600+J)-@(700+S*I+J))
2090 IF @(664+I)<27500 @(664+I)=@(664+I)+X*X
2100 NEXT J
2120 NEXT I
2130 REM FIND SMALLEST DIFFERENCE
2140 A=32000,N=0
2150 FOR I=0 TO 9
2160 IF @(664+I)<A A=@(664+I);N=I
2170 NEXT I
2180 RETURN
3000 REM SPEECH COLLECTION ROUTINE
3010 C=0
3020 SPEECH 150,10
3030 T=6
3040 FOR I=C TO 599 STEP 4
3050 IF @(I)+@(I+1)+@(I+2)>T GOTO 3090
3060 NEXT I
```

```
3070 PRINT "NO SPEECH"
3080 RETURN
3090 B=I
3100 D=B
3110 FOR I=D TO 599 STEP 4
3120 IF @(I)+@(I+1)+@(I+2)<T GOTO 3160
3130 NEXT I
3140 PRINT "SPEECH OUT OF WINDOW
3150 RETURN
3160 E=I
3180 L=(E-B)/4+1
3190 IF L<10 C=E:GOTO 3040
3200 FOR J=E TO E+39 STEP 4
3210 IF @(J)+@(J+1)+@(J+2)>T D=J:GOTO 3110
3220 NEXT J
3230 REM COMPUTE AVERAGE AMPLITUDE
3240 P=0
3250 FOR I=B TO E STEP 4
3260 P=P+@(I)+@(I+1)+@(I+2)
3270 NEXT I
3280 A=P/L/3
3290 Y=10*L/16
3300 D=B/4*10
3310 FOR K=0 TO S-1 STEP 4
3320 Q=D/10:R=D-Q*10
3330 Q=Q*4
3340 FOR J=0 TO 3
3350 @(600+K+J)=((@(Q+J+4)-@(Q+J))*R/10+@(Q+J))
3360 NEXT J
3370 D=D+Y
3380 NEXT K
3390 RETURN

OK
>
```

The following changes are required to convert Program 1 to Program 2, which utilizes Polynomial Regression along with the Euclidian Distance measure of Program 1:

(1) Add line 105 105 INPUT F

(2) Change line 2090 to read:

2090 IF @(664+I)<27500 @(664+I) = @(664+I)+X*X-F*X

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3700            0010 *  CSRMON
3700            0020 *
3700            0030 *  KEYBOARD INPUT CONTROLLER TO CALL CSR1
3700            0040 *
3700            0050 *  LLOYD RICE, COMPUTALKER CONSULTANTS
3700            0060 *  VERSION 1.06   MAY 30, 1977
3700            0070 *
3700            0080 *  DEFINE INPUT STRING BUFFER LENGTH
3700            0090 INBFLN     EQU   76
3700            0100 *
3700 31 00 20   0110 CSRMON     LXI   SP,2000H * THIS IS THE ONLY STACK INSTR
3703 21 94 37   0120            LXI   H,MSG1
3706 CD 7D 37   0130 DSP0       CALL  DISPLA  * DISPLAY HEADING AND CUE
3709 3E 0D      0140            MVI   A,0DH   * THEN ANOTHER CR
370B CD D8 38   0150 DSP1       CALL  CHROUT
370E 21 B4 37   0160            LXI   H,BUFF
3711 0E 4A      0170            MVI   C,INBFLN-2
3713 CD 6F 37   0180 CHLOOP     CALL  CHRIN   * READ A CHAR FROM KBD
3716 E6 7F      0190            ANI   7FH
3718 FE 20      0200            CPI   ' '
371A DA 32 37   0210            JC    CTRL
371D FE 5B      0220            CPI   'Z'+1
371F D2 63 37   0230            JNC   GTZ
3722 0D         0240            DCR   C
3723 C2 2A 37   0250            JNZ   DSP2-2
3726 0C         0260            INR   C
3727 C3 13 37   0270            JMP   CHLOOP
372A 77         0280            MOV   M,A
372B 23         0290            INX   H
372C CD D8 38   0300 DSP2       CALL  CHROUT  * DISPLAY THE CHAR
372F C3 13 37   0310            JMP   CHLOOP  * GET ANOTHER
3732            0320 *
3732 FE 18      0330 CTRL       CPI   18H     * CTRL X
3734 CA 0B 37   0340            JZ    DSP1    * ECHO IT TO CLEAR THE SCREEN
3737 FE 10      0350            CPI   10H     * CTRL P
3739 CA 59 37   0360            JZ    REPLAY  * PLAY THE LAST ONE AGAIN
373C FE 02      0370            CPI   2       * CTRL B
373E CA 4B 38   0380            JZ    BUFDIS  * DUMP THE BUFFER
3741 FE 0D      0390            CPI   0DH
3743 C2 13 37   0400            JNZ   CHLOOP  * IGNORE ALL ELSE BUT CR
3746 77         0410            MOV   M,A
3747 CD D8 38   0420            CALL  CHROUT  * ECHO THE RETURN
374A 21 B4 37   0430            LXI   H,BUFF
374D CD 00 20   0440            CALL  CSR1    * COMPUTE AND SAY IT
3750 C4 88 37   0450            CNZ   ERROUT  * SOMPIN' HAPND, WRITE ERR MSG
3753 21 A6 37   0460 CUE1       LXI   H,MSG2
3756 C3 06 37   0470            JMP   DSP0
3759            0480 *
3759 E5         0490 REPLAY     PUSH  H
375A C5         0500            PUSH  B
375B CD 03 20   0510            CALL  PLAY    * PLAY THE LAST ONE AGAIN
375E C1         0520            POP   B
375F E1         0530            POP   H
3760 C3 13 37   0540            JMP   CHLOOP
3763            0550 *
3763 FE 7F      0560 GTZ        CPI   7FH     * CHAR > 'Z', CHECK FOR RUBOUT
3765 C2 13 37   0570            JNZ   CHLOOP
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3768 0D         0580            DCR   C
3769 2B         0590            DCX   H
376A 00         0600            NOP   *       * MVI A,XX  WITH CHAR TO ECHO
376B 00         0610            NOP   *       *    ON RUBOUT (IF NOT 7FH)
376C C3 2C 37   0620            JMP   DSP2
376F            0630 *
376F            0640 *  CHAR INPUT (CALLED FOR CONSOLE KBD INPUT)
376F CD 73 37   0650 CHRIN      CALL  TTYIN   * REDEFINE AS NEEDED
3772 C9         0660            RET
3773            0670 *
3773            0680 *  TTY INPUT HANDLER
3773 DB 00      0690 TTYIN      IN    0
3775 E6 80      0700            ANI   80H
3777 CA 73 37   0710            JZ    $-4
377A DB 01      0720            IN    1
377C C9         0730            RET
377D            0740 *
377D            0750 *  MESSAGE OUTPUT LOOP, STOP ON CHAR=04
377D            0760 *
377D 7E         0770 DISPLA     MOV   A,M     * GET CHAR
377E FE 04      0780            CPI   4
3780 C8         0790            RZ    *       * RETURN IF EOT
3781 CD D8 38   0800            CALL  CHROUT  * NO, OUTPUT IT
3784 23         0810            INX   H
3785 C3 7D 37   0820            JMP   DISPLA
3788            0830 *
3788            0840 *  ERROR MSG OUTPUT, STOP ON CR (0DH)
3788 7E         0850 ERROUT     MOV   A,M
3789 CD D8 38   0860            CALL  CHROUT
378C 7E         0870            MOV   A,M
378D FE 0D      0880            CPI   0DH
378F C8         0890            RZ
3790 23         0900            INX   H
3791 C3 88 37   0910            JMP   ERROUT
3794            0920 *
3794 0C         0930 MSG1       DB    0CH     * FIRST CLEAR THE SCREEN
3795 53 59 4E   0940            DT    'SYNTHESIS BY RULE'
3798 54 48 45
379B 53 49 53
379E 20 42 59
37A1 20 52 55
37A4 4C 45
37A6 0D         0950 MSG2       DB    0DH
37A7 45 4E 54   0960            DT    'ENTER TEXT:'
37AA 45 52 20
37AD 54 45 58
37B0 54 3A
37B2 0D         0970            DB    0DH
37B3 04         0980            DB    4
37B4            0990 *
37B4            1000 BUFF       DS    INBFLN  * PHONEME STRING INPUT BUFFER
3800            1010 *
3800            1020 *  END OF CSRMON KEYBOARD INPUT HANDLER
3800            1030 *
3800            1040 **********
3800            1050 *
3800            1060 *  DIAGNOSTIC DUMP ROUTINES FOR INFO & DEBUGGING
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3800            1070 *
3800            1080            ORG   CSRMON+100H
3800            1090 *
3800            1100 *  DISPLAY CURRENT MATRIX CONTENTS
3800            1110 *  (CALLED BY INSTR INSERTED IN CSR1 CODE)
3800 21 EF 38   1120 MATDIS     LXI   H,MDTX
3803 CD 7D 37   1130            CALL  DISPLA  * DISPLAY "FEATURE MATRIX"
3806 2A 02 35   1140            LHLD  NEGEND
3809 11 06 35   1150            LXI   D,MATRIX+2
380C 19         1160            DAD   D
380D 7D         1170            MOV   A,L
380E 2F         1180            CMA
380F 4F         1190            MOV   C,A
3810 06 14      1200            MVI   B,20    * NUMBER OF COLS TO DISPLAY
3812 B8         1210            CMP   B
3813 DA 17 38   1220            JC    $+4
3816 48         1230            MOV   C,B
3817 C5         1240            PUSH  B
3818 21 05 35   1250            LXI   H,MATRIX+1
381B 23         1260 DL1        INX   H
381C CD 9D 38   1270            CALL  CODOUT
381F 0D         1280            DCR   C
3820 C2 1B 38   1290            JNZ   DL1
3823 CD D6 38   1300            CALL  CROUT
3826 C1         1310            POP   B
3827 06 04      1320            MVI   B,4
3829 21 05 35   1330            LXI   H,MATRIX+1
382C C5         1340 DL2        PUSH  B
382D 11 5F 00   1350            LXI   D,MATLEN
3830 19         1360            DAD   D
3831 E5         1370            PUSH  H
3832 23         1380 DL3        INX   H
3833 7E         1390            MOV   A,M
3834 CD 8D 38   1400            CALL  BYTE
3837 CD D1 38   1410            CALL  BLANK
383A 0D         1420            DCR   C
383B C2 32 38   1430            JNZ   DL3
383E CD D6 38   1440            CALL  CROUT
3841 E1         1450            POP   H
3842 C1         1460            POP   B
3843 05         1470            DCR   B
3844 C2 2C 38   1480            JNZ   DL2
3847 00         1490            NOP   *       * 'CALL CHRIN' TO PAUSE HERE
3848 00         1500            NOP
3849 00         1510            NOP
384A C9         1520            RET
384B            1530 *
384B            1540 *  BUFDIS
384B            1550 *  DUMP THE CURRENT BUFFER CONTENTS
384B            1560 *  (CALLED BY CTRL B DURING INPUT)
384B 2A 06 20   1570 BUFDIS     LHLD  BUFADR
384E 11 0B 00   1580            LXI   D,11
3851 19         1590            DAD   D
3852 22 DF 36   1600            SHLD  BUFPTR
3855 21 05 35   1610            LXI   H,MATRIX+1
3858 7E         1620 PHLOOP     MOV   A,M
3859 FE 04      1630            CPI   4
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

385B CA 53 37   1640            JZ    CUE1     * GO BACK TO INPUT LOOP
385E E5         1650            PUSH  H
385F 11 7C 01   1660            LXI   D,MATLEN*4
3862 EB         1670            XCHG
3863 19         1680            DAD   D
3864 7E         1690            MOV   A,M
3865 47         1700            MOV   B,A
3866 B7         1710            ORA   A
3867 CA 98 38   1720            JZ    ENDLP
386A EB         1730            XCHG
386B CD 9D 38   1740            CALL  CODOUT
386E 2A DF 36   1750            LHLD  BUFPTR
3871 C3 7D 38   1760            JMP   PHLP3
3874 CD D1 38   1770 PHLP2      CALL  BLANK
3877 CD D1 38   1780            CALL  BLANK
387A CD D1 38   1790            CALL  BLANK
387D 0E 09      1800 PHLP3      MVI   C,9
387F CD D1 38   1810 FRLOOP     CALL  BLANK
3882 7E         1820            MOV   A,M
3883 CD BD 38   1830            CALL  BYTE
3886 23         1840            INX   H
3887 0D         1850            DCR   C
3888 C2 7F 38   1860            JNZ   FRLOOP
388B CD D6 38   1870            CALL  CROUT
388E 05         1880            DCR   B
388F C2 74 38   1890            JNZ   PHLP2
3892 22 DF 36   1900            SHLD  BUFPTR
3895 00         1910            NOP
3896 00         1920            NOP
3897 00         1930            NOP
3898 E1         1940 ENDLP      POP   H
3899 23         1950            INX   H
389A C3 58 38   1960            JMP   PHLOOP
389D            1970 *
389D            1980 *  OUTPUT PHONEME CODE OF CURRENT MATRIX COLUMN
389D 7E         1990 CODOUT     MOV   A,M
389E 87         2000            ADD   A
389F EB         2010            XCHG
38A0 2A 0A 20   2020            LHLD  PVTAB
38A3 85         2030            ADD   L
38A4 6F         2040            MOV   L,A
38A5 7C         2050            MOV   A,H
38A6 CE 00      2060            ACI   0
38A8 67         2070            MOV   H,A
38A9 7E         2080            MOV   A,M
38AA CD D8 38   2090            CALL  CHROUT
38AD 23         2100            INX   H
38AE 7E         2110            MOV   A,M
38AF B7         2120            ORA   A
38B0 C2 B5 38   2130            JNZ   $+5
38B3 3E 20      2140            MVI   A,' '
38B5 CD D8 38   2150            CALL  CHROUT
38B8 CD D1 38   2160            CALL  BLANK
38BB EB         2170            XCHG
38BC C9         2180            RET
38BD            2190 *
38BD            2200 *  THE FOLLOWING ROUTINES ARE USED ONLY BY
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

38BD            2210 *  LOCAL CALLS, NOT BY CSR1
38BD F5         2220 BYTE       PUSH  PSW      * OUTPUT (A) AS 2 HEX DIGITS
38BE 0F         2230            RRC
38BF 0F         2240            RRC
38C0 0F         2250            RRC
38C1 0F         2260            RRC
38C2 CD C6 38   2270            CALL  HEXO
38C5 F1         2280            POP   PSW
38C6 E6 0F      2290 HEXO       ANI   0FH      * MASK OFF UPPER HALF
38C8 C6 90      2300            ADI   90H
38CA 27         2310            DAA
38CB CE 40      2320            ACI   40H
38CD 27         2330            DAA
38CE C3 D8 38   2340            JMP   CHROUT   * OUTPUT HEX DIGIT & TAKE LAST RET
38D1            2350 *
38D1 3E 20      2360 BLANK      MVI   A,' '
38D3 C3 D8 38   2370            JMP   CHROUT   * OUTPUT A SPACE
38D6            2380 *
38D6 3E 0D      2390 CROUT      MVI   A,0DH    * OUTPUT CR,(LF)
38D8            2400 *
38D8            2410 *  CHAR OUTPUT (CALLED TO OUTPUT (A))
38D8            2420 *   REDEFINE AS NEEDED
38D8 CD DC 38   2430 CHROUT     CALL  TTYOUT   * REDEFINE AS NEEDED
38DB C9         2440            RET
38DC            2450 *
38DC            2460 *  TTY OUTPUT HANDLER
38DC F5         2470 TTYOUT     PUSH  PSW
38DD DB 00      2480            IN    0
38DF E6 01      2490            ANI   1
38E1 CA DD 38   2500            JZ    $-4
38E4 F1         2510            POP   PSW
38E5 D3 01      2520            OUT   1
38E7 FE 0D      2530            CPI   0DH
38E9 C0         2540            RNZ
38EA 3E 0A      2550            MVI   A,0AH    * LAST CHAR WAS CR, ...
38EC C3 DC 38   2560            JMP   TTYOUT   * ADD A LINE FEED
38EF            2570 *
38EF            2580 *
38EF 0D         2590 MDTX       DB    0DH
38F0 46 45 41   2600            DT    'FEATURE MATRIX'
38F3 54 55 52
38F6 45 20 4D
38F9 41 54 52
38FC 49 58
38FE 0D         2610            DB    0DH
38FF 04         2620            DB    4
3900            2630 *
3900            2640 **********
3900            2650 *
3900            2660 *  A PORTION OF THE CSR1 JUMP TABLE IS DEFINED
3900            2670 *  TO ALLOW ACCESS TO MISC. CSR1 ADDRESSES
3900            2680 *
3900            2690            ORG   2000H
2000            2700 *
2000            2710 CSR1       DS    3
2003            2720 PLAY       DS    3
2006            2730 BUFADR     DS    2
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2008            2740 BUFEND     DS    2
200A            2750 PVTAB      DS    2
200C            2760 *
200C            2770 *  OTHER CSR1 LOCATIONS DEPENDENT ON LOC OF COMRAM
200C            2780 COMRAM     EQU   3500H
200C            2790 NEGEND     EQU   COMRAM+2
200C            2800 MATRIX     EQU   COMRAM+4
200C            2810 MATLEN     EQU   95
200C            2820 BUFPTR     EQU   MATLEN*5+MATRIX
200C            2830 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  PARSE
2000            0020 *
2000            0030 *  SECTION 1 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.07   MAY 30, 1977
2000            0070 *
2000            0080 *  THE ENTIRE CSR1 SOFTWARE SYSTEM IS PROVIDED
2000            0090 *  BY COMPUTALKER CONSULTANTS WITH THE UNDERSTANDING
2000            0100 *  THAT IT MAY BE REPRODUCED FOR NON-COMMERCIAL
2000            0110 *  PURPOSES, PROVIDED THAT THE SOURCE IS QUOTED.
2000            0120 *  OTHERWISE, ALL COPYRIGHTS ARE RETAINED.
2000            0130 *     (C)  1977,  COMPUTALKER CONSULTANTS
2000            0140 *
2000            0150 **********
2000            0160 *
2000            0170 *  ASSEMBLE ALL SECTIONS TO BEGIN AT THE SAME
2000            0180 *  STARTING ADDRESS.  THIS WILL BE THE ADDRESS
2000            0190 *  AT WHICH TO CALL THE CSR1 SYSTEM
2000            0200 *
2000            0210 **********
2000            0220 *
2000            0230 *  COMMON (JUMP) ADDRESS REFERENCE TABLE
2000            0240 *  THIS TABLE ALLOWS REASSEMBLING ANY SECTION
2000            0250 *  WITHOUT CHANGING REFERENCES IN OTHER SECTIONS
2000            0260 *
2000            0270 *  ANY SECTION MAY BE MOVED BY CHANGING THE ORG
2000            0280 *  AT SECTAD (SECTION 1 IS SLIGHTLY DIFFERENT).
2000            0290 *  ALL REFERENCES WILL BE HANDLED CORRECTLY.
2000            0300 *  THE COMMON RAM WORKSPACE DOES NOT SHARE THIS
2000            0310 *  NICE PROPERTY.  IT MUST BE DEFINED THE SAME
2000            0320 *  IN ALL SECTIONS.
2000            0330 *
2000            0340 **********
2000            0350 *
2000            0360 *  THE PARAMETER BUFFER IS DEFINED BY CONSTANTS IN
2000            0370 *  LOCATIONS BUFADR AND BUFEND.  BUFADR CONTAINS THE
2000            0380 *  ADDRESS OF THE 1ST BYTE OF THE FRAME COUNT, AND
2000            0390 *  BUFEND CONTAINS THE LAST ADDRESS AVAILABLE AS
2000            0400 *  BUFFER SPACE.
2000            0410 *  THESE LOCATIONS ARE DEFINED ONLY IN SECT1 SOURCE
2000            0420 *
2000            0430 **********
2000            0440 *
2000            0450 *  COMMON JUMP ADDRESS TABLE
2000            0460 *
2000            0470 COMJMP     EQU   $
2000            0480 *
2000 C3 3A 20   0490            JMP   CSR1
2003            0500 PLAY       DS    3
2006 00 39      0510 BUFADR     DW    COMJMP+1900H * 1ST BUFFER LOC AVAIL
2008 FF 5F      0520 BUFEND     DW    COMJMP+3FFFH * LAST LOC AVAILABLE
200A C5 22      0530            DW    PVTAB
200C C3 AC 22   0540            JMP   MATPAK
200F C3 9A 22   0550            JMP   MATERR
2012            0560 RULES      DS    3
2015            0570 SETDUR     DS    3
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2018            0580 RULES3     DS    3
201B            0590 GENF0      DS    3
201E            0600 CLRBUF     DS    3
2021            0610 GENPRM     DS    3
2024            0620 DUMMY      DS    22
203A            0630 *
203A            0640 SECTAD     EQU   $
203A            0650 *
203A            0660 **********
203A            0670 *
203A            0680 *  COMRAM ORIGEN DEFINITION
203A            0690 *
203A            0700            ORG   COMJMP+1500H
3500            0710 COMRAM     EQU   $
3500            0720 *
3500            0730 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0740 *
3500            0750 MATPTR     DS    2
3502            0760 NEGEND     DS    2
3504            0770 MATRIX     EQU   $
3504            0780 MATLEN     EQU   95
3504            0790 PHCODE     DS    MATLEN
3563            0800 FEATA      DS    MATLEN
35C2            0810 FEATB      DS    MATLEN
3621            0820 STRES      DS    MATLEN
3680            0830 DUR        DS    MATLEN
36DF            0840 MATEND     EQU   $
36DF            0850 *
36DF            0860 *  PARSE (LOCAL) RAM WORKSPACE
36DF            0870 *
36DF            0880 CHAR       DS    1
36E0            0890 PHON       DS    2
36E2            0900 NUM        DS    1
36E3            0910 INPTR      DS    2
36E5            0920 *
36E5            0930 **********
36E5            0940 *
36E5            0950 *  CSR1 SYSTEM MAIN ROUTINE
36E5            0960 *
36E5            0970            ORG   SECTAD
203A            0980 *
203A            0990 *  DEFINE PHONEME CODES FOR PARSE
203A            1000 *
203A            1010 CPAUSE     EQU   2
203A            1020 CTERM      EQU   4
203A            1030 CAX        EQU   15
203A            1040 CEL        EQU   29
203A            1050 CEM        EQU   30
203A            1060 CEN        EQU   31
203A            1070 CL         EQU   33
203A            1080 CM         EQU   36
203A            1090 CN         EQU   37
203A            1100 CT         EQU   40
203A            1110 CD         EQU   44
203A            1120 CSH        EQU   51
203A            1130 CZH        EQU   55
203A            1140 CCH        EQU   56
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

203A            1150 CJH       EQU   57
203A            1160 *
203A            1170 *  DEFINE FEATURE LABELS
203A            1180 *
203A            1190 VOWEL     EQU   80H
203A            1200 CONS      EQU   40H
203A            1210 FRONT     EQU   20H
203A            1220 DIPHTH    EQU   10H
203A            1230 WDBND     EQU   4
203A            1240 PHBND     EQU   2
203A            1250 IGNORE    EQU   1
203A            1260 STOP      EQU   80H
203A            1270 VOICE     EQU   40H
203A            1280 PLOS      EQU   20H
203A            1290 PLOSA     EQU   10H
203A            1300 FRIC      EQU   8
203A            1310 LIQUID    EQU   4
203A            1320 NASAL     EQU   2
203A            1330 DENTAL    EQU   1
203A            1340 *
203A            1350 *  DEFINE INPUT STRING TERMINATOR
203A            1360 *
203A            1370 TERM      EQU   0DH
203A            1380 *
203A            1390 **********
203A            1400 *
203A            1410 *  CSR1 MAIN LOOP
203A            1420 *
203A C5         1430 CSR1      PUSH  B       * SAVE BC,DE
203B D5         1440           PUSH  D
203C CD 63 20   1450           CALL  PARSE   * PARSE INPUT & SET UP MATRIX
203F C2 60 20   1460           JNZ   CSERR   * RETURN IF ERROR
2042 00         1470           NOP
2043 00         1480           NOP
2044 00         1490           NOP
2045 CD 12 20   1500           CALL  RULES   * APPLY RULES TO MATRIX
2048 C2 60 20   1510           JNZ   CSERR   * RETURN IF ERROR
204B 00         1520           NOP
204C 00         1530           NOP
204D 00         1540           NOP
204E CD 1B 20   1550           CALL  GENF0   * GENERATE F0 PARAMETER
2051 C2 60 20   1560           JNZ   CSERR   * RETURN IF ERROR
2054 00         1570           NOP
2055 00         1580           NOP
2056 00         1590           NOP
2057 CD 21 20   1600           CALL  GENPRM  * GENERATE OTHER PARAMETERS
205A 00         1610           NOP
205B 00         1620           NOP
205C 00         1630           NOP
205D CD 03 20   1640           CALL  PLAY    * PLAYBACK TO SYNTHESIZER
2060 D1         1650 CSERR     POP   D
2061 C1         1660           POP   B
2062 C9         1670           RET
2063            1680 *
2063            1690 *
2063            1700 **********
2063            1710 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2063            1720 *  PARSE CODE
2063            1730 *
2063            1740 **********
2063            1750 *
2063 22 E3 36   1760 PARSE      SHLD  INPTR     * SAVE INPUT TEXT POINTER
2066 21 04 35   1770            LXI   H,MATRIX
2069 22 00 35   1780            SHLD  MATPTR    * INITIALIZE PHONEME MATRIX
206C AF         1790            XRA   A
206D 32 E2 36   1800            STA   NUM
2070 3E 04      1810            MVI   A,CTERM   * FRONT MATRIX TERMINATOR
2072 CD 7A 22   1820            CALL  MATSET    * NO ERROR TEST NEEDED
2075 3E 02      1830            MVI   A,CPAUSE
2077 CD 7A 22   1840            CALL  MATSET    * SET UP INITIAL PAUSE
207A CD BD 21   1850 PARSA      CALL  GET
207D FE 0D      1860 PARSB      CPI   TERM      * END OF INPUT?
207F 3E 00      1870            MVI   A,0       * DON'T WIPE THE Z FLAG
2081 32 E2 36   1880            STA   NUM
2084 CA 3F 21   1890            JZ    FINUP     * END, CLOSE THE MATRIX
2087 67         1900            MOV   H,A
2088 6F         1910            MOV   L,A
2089 22 E0 36   1920            SHLD  PHON      * CLEAR PHON,
208C 4F         1930            MOV   C,A       * & C FOR HH COUNTER
208D 7A         1940            MOV   A,D       * GET PARSER FLAGS INTO A
208E 17         1950            RAL
208F DA A6 20   1960            JC    VOWELA    * VOWEL CLASS A
2092 17         1970            RAL
2093 DA CD 20   1980            JC    CONSB     * CONSONANT CLASS B
2096 17         1990            RAL
2097 DA 18 21   2000            JC    CONSA     * CONSONANT CLASS A
209A 17         2010            RAL
209B DA 33 21   2020            JC    COMMT     * COMMENT DELIMITER
209E E6 C0      2030            ANI   0C0H
20A0 C2 20 21   2040            JNZ   PLKPA     * SINGLE CHAR SYMBOL
20A3 C3 5F 21   2050            JMP   ERROR     * BAD INPUT
20A6            2060 *
20A6 CD B1 21   2070 VOWELA     CALL  PUSHP     * PUSH VOWEL CHAR A, GET CHAR B
20A9 7A         2080            MOV   A,D
20AA 1F         2090            RAR
20AB D2 5F 21   2100            JNC   ERROR     * IF NOT VOWEL B, IT'S AN ERROR
20AE CD B1 21   2110            CALL  PUSHP     * PUSH CHAR B, GET NEXT
20B1 7A         2120 GETDIG     MOV   A,D       * SEE IF VOWEL IS FOLLOWED BY INT
20B2 1F         2130            RAR
20B3 1F         2140            RAR
20B4 D2 DA 20   2150            JNC   LKPNG     * NOT A DIGIT, LOOKUP THE VOWEL
20B7 3A DF 36   2160            LDA   CHAR      * ACCUMULATE STRESS VALUE IN C
20BA D6 30      2170            SUI   '0'
20BC 47         2180            MOV   B,A       * SAVE NEW DIGIT IN B
20BD 79         2190            MOV   A,C
20BE 87         2200            ADD   A
20BF 87         2210            ADD   A
20C0 81         2220            ADD   C
20C1 87         2230            ADD   A         * 10 TIMES PREVIOUS VALUE
20C2 80         2240            ADD   B         * PLUS NEW DIGIT
20C3 4F         2250            MOV   C,A       * INTO C
20C4 32 E2 36   2260            STA   NUM       * AND INTO NUM
20C7 CD BD 21   2270            CALL  GET
20CA C3 B1 20   2280            JMP   GETDIG    * & SEE IF THERE'S ANOTHER DIGIT
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

20CD            2290 *
20CD CD B1 21   2300 CONSB      CALL  PUSHP     * PUSH CONS CHAR, GET NEXT
20D0 FE 48      2310            CPI   'H'       * IS IT FOLLOWED BY AN "H"?
20D2 CA F0 20   2320            JZ    GETH      * YES
20D5 FE 58      2330            CPI   'X'       * IS IT FOLLOWED BY AN 'X'
20D7 CA 20 21   2340            JZ    PLKPA     * YES, PUSH & LOOK IT UP
20DA CD 0B 22   2350 LKPNG      CALL  LOOKUP    * NO, LOOKUP, DON'T GET AGAIN
20DD C0         2360            RNZ   *         * RETURN IF ERROR
20DE 3A DF 36   2370 PARSBG     LDA   CHAR
20E1 CD CA 21   2380            CALL  GETFLG    * GET CHAR FLAGS INTO D AGAIN
20E4 C3 7D 20   2390            JMP   PARSB
20E7            2400 *
20E7 0C         2410 INCRH      INR   C         * INCREMENT THE "H" COUNTER
20E8 CD BD 21   2420            CALL  GET
20EB FE 48      2430            CPI   'H'
20ED C2 03 21   2440            JNZ   LKPH      * CHAR IS NOT H, LOOKUP LAST HH
20F0 CD BD 21   2450 GETH       CALL  GET
20F3 FE 48      2460            CPI   'H'
20F5 CA E7 20   2470            JZ    INCRH     * WE HAVE ANOTHER 'H'
20F8 3A E1 36   2480            LDA   PHON+1    * END OF THE STRING OF 'H'S
20FB 32 E0 36   2490            STA   PHON
20FE 3E 48      2500            MVI   A,'H'
2100 32 E1 36   2510            STA   PHON+1    * PUSH AN 'H' INTO PHON
2103 CD 0B 22   2520 LKPH       CALL  LOOKUP    * LOOK IT UP
2106 C0         2530            RNZ
2107 79         2540            MOV   A,C
2108 B7         2550            ORA   A
2109 CA DE 20   2560            JZ    PARSBG    * ALL HH'S DONE, CHAR IS NEXT
210C 3E 48      2570            MVI   A,'H'
210E 32 E0 36   2580            STA   PHON
2111 32 E1 36   2590            STA   PHON+1    * FORCE PHON TO 'HH'
2114 0D         2600            DCR   C
2115 C3 03 21   2610            JMP   LKPH      * AND LOOK IT UP
2118            2620 *
2118 CD B1 21   2630 CONSA      CALL  PUSHP     * PUSH CONS CHAR, GET NEXT
211B FE 48      2640            CPI   'H'
211D C2 5F 21   2650            JNZ   ERROR     * CLASS A CONS MUST HAVE 'H' NEXT
2120 3A E1 36   2660 PLKPA      LDA   PHON+1    * PUSH IT INTO PHON
2123 32 E0 36   2670            STA   PHON
2126 3A DF 36   2680            LDA   CHAR
2129 32 E1 36   2690            STA   PHON+1
212C CD 0B 22   2700            CALL  LOOKUP    * LOOK IT UP
212F C0         2710            RNZ
2130 C3 7A 20   2720            JMP   PARSA     * GET NEXT INPUT
2133            2730 *
2133 CD BD 21   2740 COMMT      CALL  GET       * GET ANOTHER CHAR
2136 7A         2750            MOV   A,D
2137 E6 10      2760            ANI   10H
2139 CA 33 21   2770            JZ    COMMT     * IGNORE ALL UNTIL NEXT CMMT DELI
213C C3 7A 20   2780            JMP   PARSA
213F            2790 *
213F AF         2800 FINUP      XRA   A
2140 32 E2 36   2810            STA   NUM
2143 3E 02      2820            MVI   A,CPAUSE
2145 CD 7A 22   2830            CALL  MATSET    * PUT FINAL PAUSE IN MATRIX
2148 C0         2840            RNZ
2149 3E 04      2850            MVI   A,CTERM
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

214B CD 7A 22    2860           CALL  MATSET   * TERMINATE MATRIX
214E C0          2870           RNZ
214F 2A 00 35    2880           LHLD  MATPTR
2152 7C          2890           MOV   A,H
2153 2F          2900           CMA
2154 67          2910           MOV   H,A
2155 7D          2920           MOV   A,L
2156 2F          2930           CMA
2157 6F          2940           MOV   L,A
2158 23          2950           INX   H
2159 23          2960           INX   H
215A 22 02 35    2970           SHLD  NEGEND  * -ADDRESS OF TERM PHON
215D AF          2980           XRA   A
215E C9          2990           RET
215F             3000 *
215F             3010 *  SYNTAX ERROR MESSAGE RETURN
215F             3020 *
215F 21 80 36    3030 ERROR     LXI   H,DUR    * USE DURATION ARRAY FOR MSG
2162 11 9F 21    3040           LXI   D,ERTX1
2165 CD 98 21    3050           CALL  MOVCH    * MOVE 1ST PART OF MSG
2168 C2 65 21    3060           JNZ   $-3
216B 36 0D       3070           MVI   M,0DH    * TERMINATE IT
216D 11 E1 36    3080           LXI   D,PHON+1 * SEE IF ANYTHING IN PHON
2170 1A          3090           LDAX  D
2171 47          3100           MOV   B,A
2172 1B          3110           DCX   D
2173 1A          3120           LDAX  D
2174 B0          3130           ORA   B
2175 CA 92 21    3140           JZ    NZRET    * NO, SET NON-0 & RETURN
2178 11 AB 21    3150           LXI   D,ERTX2  * YES, ADD MORE TO MSG
217B CD 98 21    3160           CALL  MOVCH
217E C2 7B 21    3170           JNZ   $-3
2181 11 E0 36    3180           LXI   D,PHON
2184 CD 98 21    3190           CALL  MOVCH    * MOVE NON-0 PARTS OF PHON & CHA
2187 CD 98 21    3200           CALL  MOVCH
218A CD 98 21    3210           CALL  MOVCH
218D 36 22       3220           MVI   M,'"'
218F 23          3230           INX   H
2190 36 0D       3240           MVI   M,0DH    * TERMINATE THE MSG
2192 21 80 36    3250 NZRET     LXI   H,DUR
2195 F6 FF       3260           ORI   255
2197 C9          3270           RET
2198             3280 *
2198 1A          3290 MOVCH     LDAX  D
2199 13          3300           INX   D
219A B7          3310           ORA   A
219B C8          3320           RZ
219C 77          3330           MOV   M,A
219D 23          3340           INX   H
219E C9          3350           RET
219F             3360 *
219F 49 4E 50    3370 ERTX1     DT    'INPUT ERROR'
21A2 55 54 20
21A5 45 52 52
21A8 4F 52
21AA 00          3380           DB    0
21AB 20 41 54    3390 ERTX2     DT    ' AT "'
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

21AE 20 22              3400            DB    0
21B0 00                 3410 *
21B1                    3420 **********
21B1                    3430 *
21B1                    3440 *  PARSE SUBROUTINES
21B1                    3450 *
21B1 3A E1 36           3460 PUSHP      LDA   PHON+1
21B4 32 E0 36           3470            STA   PHON
21B7 3A DF 36           3480            LDA   CHAR
21BA 32 E1 36           3490            STA   PHON+1
21BD                    3500 *
21BD 2A E3 36           3510 GET        LHLD  INPTR
21C0 7E                 3520            MOV   A,M       * GET NEXT INPUT CHAR
21C1 E6 7F              3530            ANI   7FH
21C3 32 DF 36           3540            STA   CHAR      * SAVE IT
21C6 23                 3550            INX   H         * AND BUMP INPUT POINTER
21C7 22 E3 36           3560            SHLD  INPTR
21CA C5                 3570 GETFLG     PUSH  B
21CB 4F                 3580            MOV   C,A       * KEEP CHAR IN C FOR NOW
21CC E6 1F              3590            ANI   1FH       * KEEP 5 LS BITS
21CE 5F                 3600            MOV   E,A
21CF 21 E7 21           3610            LXI   H,PARFLG  * ADDR OF FLAGS TABLE
21D2 16 00              3620            MVI   D,0
21D4 19                 3630            DAD   D
21D5 46                 3640            MOV   B,M       * GET FLAGS WORD IN B
21D6 79                 3650            MOV   A,C       * GET WHOLE CHAR BACK
21D7 07                 3660            RLC
21D8 07                 3670            RLC
21D9 07                 3680            RLC
21DA E6 03              3690            ANI   3         * KEEP CHAR BITS 5,6 IN A 0,1
21DC 5F                 3700            MOV   E,A       * OFFSET TO MASK TABLE
21DD 21 07 22           3710            LXI   H,CHMASK
21E0 19                 3720            DAD   D
21E1 7E                 3730            MOV   A,M       * GET MASK BITS IN A
21E2 A0                 3740            ANA   B
21E3 57                 3750            MOV   D,A       * RETURN FLAGS IN D
21E4 79                 3760            MOV   A,C       * RETURN CHAR IN A
21E5 C1                 3770            POP   B
21E6 C9                 3780            RET
21E7                    3790 *
21E7                    3800 *  PARSER FLAGS TABLE
21E7                    3810 *
21E7 04                 3820 PARFLG     DB    04H   * @,SPACE
21E8 81                 3830            DB    81H   * A,!
21E9 08                 3840            DB    08H   * B,"
21EA 20                 3850            DB    20H   * C,#
21EB 40                 3860            DB    40H   * D,$
21EC 81                 3870            DB    81H   * E,%
21ED 08                 3880            DB    08H   * F,&
21EE 40                 3890            DB    40H   * G,'
21EF 21                 3900            DB    21H   * H,(
21F0 81                 3910            DB    81H   * I,)
21F1 30                 3920            DB    30H   * J,*
21F2 40                 3930            DB    40H   * K,+
21F3 45                 3940            DB    45H   * L,,
21F4 0D                 3950            DB    0DH   * M,-
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

21F5 45          3960           DB    45H   * N,.
21F6 85          3970           DB    85H   * O,/
21F7 0A          3980           DB    0AH   * P,0
21F8 0A          3990           DB    0AH   * Q,1
21F9 43          4000           DB    43H   * R,2
21FA 42          4010           DB    42H   * S,3
21FB 42          4020           DB    42H   * T,4
21FC 83          4030           DB    83H   * U,5
21FD 0A          4040           DB    0AH   * V,6
21FE 43          4050           DB    43H   * W,7
21FF 03          4060           DB    03H   * X,8
2200 43          4070           DB    43H   * Y,9
2201 44          4080           DB    44H   * Z,:
2202 04          4090           DB    04H   * [,;
2203 00          4100           DB    0     * BACK SLASH,<
2204 00          4110           DB    0     * ],=
2205 00          4120           DB    0     * UP ARROW,>
2206 04          4130           DB    04H   * LEFT ARROW,
2207             4140 *
2207             4150 *  PARSER FLAGS BIT MASKS
2207             4160 *
2207 00          4170 CHMASK    DB    0       * CONTROL CHAR, CLEAR FLAGS
2208 16          4180           DB    16H     * PUNCTUATION, USE BITS 4,2,1
2209 E9          4190           DB    0E9H    * UPPER CASE, USE BITS 7,6,5,3,0
220A 00          4200           DB    0       * LOWER CASE, CLEAR FLAGS
220B             4210 *
220B             4220 *
220B             4230 *  LOOKUP SUBROUTINE
220B             4240 *
220B C5          4250 LOOKUP    PUSH  B
220C 2A E0 06    4260           LHLD  PHON      * 2ND HALF OF PHON INTO E
220F 7D          4270           MOV   A,L       * & 1ST HALF IN A & L
2210 B7          4280           ORA   A
2211 CA 16 22    4290           JZ    NCHP      * IF 1ST 1/2 =0, MAKE IT THE 2ND
2214 6C          4300           MOV   L,H       * INTERCHANGE H & L
2215 67          4310           MOV   H,A
2216 EB          4320 NCBP      XCHG  *         * NON 0 PART IN D, 2ND 1/2 IN E
2217 0E 40       4330           MVI   C,64
2219 21 C5 22    4340           LXI   H,PVTAB
221C 7A          4350 CHECK     MOV   A,D       * SEARCH VALID. TABLE FOR (PHON)
221D BE          4360           CMP   M
221E 23          4370           INX   H
221F C2 27 22    4380           JNZ   PiBAD
2222 7E          4390           MOV   A,E       * 1ST 1/2 FOUND. TRY 2ND 1/2
2223 BE          4400           CMP   M
2224 CA 30 22    4410           JZ    MATCH
2227 23          4420 PiBAD     INX   H         * NOT THIS ONE
2228 0D          4430           DCR   C
2229 C2 1C 22    4440           JNZ   CHECK
222C C1          4450           POP   B
222D C3 5F 21    4460           JMP   ERROR     * (PHON) NOT IN VALID. TABLE
2230             4470 *
2230 3E 3A       4480 MATCH     MVI   A,-PVTAB-1&256  * (-PVTAB-1)MOD 256
2232 85          4490           ADD   L
2233 0F          4500           RRC
2234 FE 1D       4510           CPI   CEL       * CHECK NOW FOR DUMMY CODES
2236 CA 50 22    4520           JZ    INSAXL    * EL BECOMES AX,L
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2239 FE 1E      4530            CPI   CEM
223B CA 55 22   4540            JZ    INSAXM    * EM BECOMES AX,M
223E FE 1F      4550            CPI   CEN
2240 CA 5A 22   4560            JZ    INSAXN    * EN BECOMES AX,N
2243 FE 38      4570            CPI   CCH
2245 CA 61 22   4580            JZ    INSTSH    * CH BECOMES T,SH
2248 FE 39      4590            CPI   CJH
224A CA 68 22   4600            JZ    INSDZH    * JH BECOMES D,ZH
224D C3 75 22   4610            JMP   PUTMAT    * ELSE PUT IT IN THE MATRIX
2250            4620 *
2250 06 21      4630 INSAXL     MVI   B,CL
2252 C3 5C 22   4640            JMP   INS1
2255 06 24      4650 INSAXM     MVI   B,CM
2257 C3 5C 22   4660            JMP   INS1
225A 06 25      4670 INSAXN     MVI   B,CN
225C 3E 0F      4680 INS1       MVI   A,CAX
225E C3 6C 22   4690            JMP   INS2
2261 3E 28      4700 INSTSH     MVI   A,CT
2263 06 33      4710            MVI   B,CSH
2265 C3 6C 22   4720            JMP   INS2
2268 3E 2C      4730 INSDZH     MVI   A,CD
226A 06 37      4740            MVI   B,CZH
226C CD 7A 22   4750 INS2       CALL  MATSET    * PUT 1ST PHON IN MATRIX
226F 3E 00      4760            MVI   A,0
2271 32 E2 36   4770            STA   NUM       * CLEAR STRESS VALUE FOR 2ND PHO
2274 78         4780            MOV   A,B
2275 CD 7A 22   4790 PUTMAT     CALL  MATSET    * AND PUT IT IN MATRIX
2278 C1         4800            POP   B
2279 C9         4810            RET
227A            4820 *
227A            4830 *
227A 2A 00 35   4840 MATSET     LHLD  MATPTR    * TEST COLUMN POINTER
227D 11 9C CA   4850            LXI   D,-MATRIX-MATLEN-1
2280 19         4860            DAD   D
2281 DA 9A 22   4870            JC    MATERR    * MATRIX OVERFLOW
2284 C5         4880            PUSH  B         * SAVE B
2285 CD AC 22   4890            CALL  MATPAK    * PUT NEW CODE INTO MATRIX
2288 09         4900            DAD   B
2289 3A E2 36   4910            LDA   NUM
228C 77         4920            MOV   M,A       * GET STRESS VALUE FROM NUM
228D 09         4930            DAD   B
228E 36 00      4940            MVI   M,0       * DURATION VALUES
2290 C1         4950            POP   B         * RESTORE B
2291 2A 00 35   4960            LHLD  MATPTR
2294 23         4970            INX   H
2295 22 00 35   4980            SHLD  MATPTR    * BUMP THE COLUMN PTR
2298 AF         4990            XRA   A         * SET ZERO CONDITION
2299 C9         5000            RET
229A            5010 *
229A 21 A0 22   5020 MATERR     LXI   H,MERTX
229D F6 FF      5030            ORI   255
229F C9         5040            RET
22A0            5050 *
22A0 4D 41 54   5060 MERTX      DT    'MATRIX FULL'
22A3 52 49 58
22A6 20 46 55
22A9 4C 4C
```

```
  ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

  22AB 0D         5070            DB    0DH
  22AC            5080 *
  22AC            5090 *
  22AC            5100 *  MATPAK SUBROUTINE
  22AC            5110 *
  22AC 2A 00 35   5120 MATPAK     LHLD  MATPTR
  22AF 77         5130            MOV   M,A       * PUT CODE IN ROW 1
  22B0 EB         5140            XCHG
  22B1 07         5150            RLC
  22B2 4F         5160            MOV   C,A
  22B3 06 00      5170            MVI   B,0       * CODE*2 INTO BC
  22B5 21 3D 23   5180            LXI   H,FEATAB
  22B8 09         5190            DAD   B
  22B9 EB         5200            XCHG  *         * FEATURE ADDR IN DE
  22BA 01 5F 00   5210            LXI   B,MATLEN
  22BD 09         5220            DAD   B         * BUMP HL TO FEATA ROW
  22BE 1A         5230            LDAX  D
  22BF 77         5240            MOV   M,A       * MOVE 1ST FEATURE WORD T MATRIX
  22C0 13         5250            INX   D
  22C1 1A         5260            LDAX  D
  22C2 09         5270            DAD   B
  22C3 77         5280            MOV   M,A       * MOVE 2ND FEATURE WORD TO MATRIX
  22C4 C9         5290            RET
  22C5            5300 *
N 22C5            5310 *******
  22C5            5320 *
  22C5            5330 *  PHONEME VALIDATION TABLE
  22C5            5340 *
  22C5 20         5350 PVTAB      DB    ' '
  22C6 00         5360            DB    0
  22C7 2E         5370            DB    '.'
  22C8 00         5380            DB    0
  22C9 2C         5390            DB    ','
  22CA 00         5400            DB    0
  22CB 3F         5410            DB    '?'
  22CC 00         5420            DB    0
  22CD 23         5430            DB    '#'
  22CE 00         5440            DB    0
  22CF 49 59      5450            DT    'IY'
  22D1 49 48      5460            DT    'IH'
  22D3 45 48      5470            DT    'EH'
  22D5 41 45      5480            DT    'AE'
  22D7 41 41      5490            DT    'AA'
  22D9 41 48      5500            DT    'AH'
  22DB 41 4F      5510            DT    'AO'
  22DD 4F 57      5520            DT    'OW'
  22DF 55 48      5530            DT    'UH'
  22E1 55 57      5540            DT    'UW'
  22E3 41 58      5550            DT    'AX'
  22E5 49 58      5560            DT    'IX'
  22E7 45 52      5570            DT    'ER'
  22E9 55 58      5580            DT    'UX'
  22EB 4F 48      5590            DT    'OH'
  22ED 41 57      5600            DT    'AW'
  22EF 41 59      5610            DT    'AY'
  22F1 4F 59      5620            DT    'OY'
  22F3 45 59      5630            DT    'EY'
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

22F5 52 58       5640           DT    'RX'
22F7 4C 58       5650           DT    'LX'
22F9 57 58       5660           DT    'WX'
22FB 59 58       5670           DT    'YX'
22FD 57 48       5680           DT    'WH'
22FF 45 4C       5690           DT    'EL'
2301 45 4D       5700           DT    'EM'
2303 45 4E       5710           DT    'EN'
2305 52          5720           DB    'R'
2306 00          5730           DB    0
2307 4C          5740           DB    'L'
2308 00          5750           DB    0
2309 57          5760           DB    'W'
230A 00          5770           DB    0
230B 59          5780           DB    'Y'
230C 00          5790           DB    0
230D 4D          5800           DB    'M'
230E 00          5810           DB    0
230F 4E          5820           DB    'N'
2310 00          5830           DB    0
2311 4E 58       5840           DT    'NX'
2313 50          5850           DB    'P'
2314 00          5860           DB    0
2315 54          5870           DB    'T'
2316 00          5880           DB    0
2317 4B          5890           DB    'K'
2318 00          5900           DB    0
2319 4B 58       5910           DT    'KX'
231B 42          5920           DB    'B'
231C 00          5930           DB    0
231D 44          5940           DB    'D'
231E 00          5950           DB    0
231F 47          5960           DB    'G'
2320 00          5970           DB    0
2321 47 58       5980           DT    'GX'
2323 44 58       5990           DT    'DX'
2325 46          6000           DB    'F'
2326 00          6010           DB    0
2327 54 48       6020           DT    'TH'
2329 53          6030           DB    'S'
232A 00          6040           DB    0
232B 53 48       6050           DT    'SH'
232D 56          6060           DB    'V'
232E 00          6070           DB    0
232F 44 48       6080           DT    'DH'
2331 5A          6090           DB    'Z'
2332 00          6100           DB    0
2333 5A 48       6110           DT    'ZH'
2335 43 48       6120           DT    'CH'
2337 4A 48       6130           DT    'JH'
2339 48 48       6140           DT    'HH'
233B 51          6150           DB    'Q'
233C 00          6160           DB    0
233D             6170 *
233D             6180 *
233D             6190 *  FEATURE DEFINITION TABLE
233D             6200 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

233D 05         6210 FEATAB     DB    WDBND+IGNORE   * SPACE
233E 00         6220            DB    0
233F 03         6230            DB    PHBND+IGNORE   * PERIOD
2340 00         6240            DB    0
2341 02         6250            DB    PHBND    * COMMA
2342 00         6260            DB    0
2343 03         6270            DB    PHBND+IGNORE   * QUESTION
2344 00         6280            DB    0
2345 03         6290            DB    PHBND+IGNORE   * TERMINATOR
2346 00         6300            DB    0
2347 A0         6310            DB    VOWEL+FRONT    * IY
2348 40         6320            DB    VOICE
2349 A0         6330            DB    VOWEL+FRONT    * IH
234A 40         6340            DB    VOICE
234B A0         6350            DB    VOWEL+FRONT    * EH
234C 40         6360            DB    VOICE
234D A0         6370            DB    VOWEL+FRONT    * AE
234E 40         6380            DB    VOICE
234F A0         6390            DB    VOWEL+FRONT    * AA
2350 40         6400            DB    VOICE
2351 A0         6410            DB    VOWEL+FRONT    * AH
2352 40         6420            DB    VOICE
2353 80         6430            DB    VOWEL    * AO
2354 40         6440            DB    VOICE
2355 90         6450            DB    VOWEL+DIPHTH   * OW
2356 40         6460            DB    VOICE
2357 80         6470            DB    VOWEL          * UH
2358 40         6480            DB    VOICE
2359 90         6490            DB    VOWEL+DIPHTH   * UW
235A 40         6500            DB    VOICE
235B 80         6510            DB    VOWEL          * AX
235C 40         6520            DB    VOICE
235D 80         6530            DB    VOWEL          * IX
235E 40         6540            DB    VOICE            .
235F 80         6550            DB    VOWEL          * ER
2360 40         6560            DB    VOICE
2361 80         6570            DB    VOWEL          * UX
2362 40         6580            DB    VOICE
2363 80         6590            DB    VOWEL          * OH
2364 40         6600            DB    VOICE
2365 90         6610            DB    VOWEL+DIPHTH   * AW
2366 40         6620            DB    VOICE
2367 B0         6630            DB    VOWEL+FRONT+DIPHTH * AY
2368 40         6640            DB    VOICE
2369 B0         6650            DB    VOWEL+FRONT+DIPHTH * OY
236A 40         6660            DB    VOICE
236B B0         6670            DB    VOWEL+FRONT+DIPHTH * EY
236C 40         6680            DB    VOICE
236D 80         6690            DB    VOWEL          * RX
236E 40         6700            DB    VOICE
236F 80         6710            DB    VOWEL          * LX
2370 40         6720            DB    VOICE
2371 80         6730            DB    VOWEL          * WX
2372 40         6740            DB    VOICE
2373 80         6750            DB    VOWEL          * YX
2374 40         6760            DB    VOICE
2375 40         6770            DB    CONS           * WH
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2376 40         6780            DB    VOICE
2377 00         6790            DB    0       * EL
2378 00         6800            DB    0
2379 00         6810            DB    0       * EM
237A 00         6820            DB    0
237B 00         6830            DB    0       * EN
237C 00         6840            DB    0
237D 40         6850            DB    CONS            * R
237E 44         6860            DB    VOICE+LIQUID
237F 40         6870            DB    CONS            * L
2380 44         6880            DB    VOICE+LIQUID
2381 40         6890            DB    CONS            * W
2382 44         6900            DB    VOICE+LIQUID
2383 40         6910            DB    CONS            * Y
2384 40         6920            DB    VOICE
2385 40         6930            DB    CONS            * M
2386 C2         6940            DB    STOP+VOICE+NASAL
2387 40         6950            DB    CONS            * N
2388 C3         6960            DB    STOP+VOICE+NASAL+DENTAL
2389 40         6970            DB    CONS            * NX
238A C2         6980            DB    STOP+VOICE+NASAL
238B 40         6990            DB    CONS            * P
238C B0         7000            DB    STOP+PLOS+PLOSA
238D 40         7010            DB    CONS            * T
238E B1         7020            DB    STOP+PLOS+PLOSA+DENTAL
238F 40         7030            DB    CONS            * K
2390 B0         7040            DB    STOP+PLOS+PLOSA
2391 40         7050            DB    CONS            * KX
2392 B0         7060            DB    STOP+PLOS+PLOSA
2393 40         7070            DB    CONS            * B
2394 E0         7080            DB    STOP+PLOS+VOICE
2395 40         7090            DB    CONS            * D
2396 E1         7100            DB    STOP+PLOS+VOICE+DENTAL
2397 40         7110            DB    CONS            * G
2398 E0         7120            DB    STOP+PLOS+VOICE
2399 40         7130            DB    CONS            * GX
239A E0         7140            DB    STOP+PLOS+VOICE
239B 40         7150            DB    CONS            * DX
239C 81         7160            DB    STOP+DENTAL
239D 40         7170            DB    CONS            * F
239E 08         7180            DB    FRIC
239F 40         7190            DB    CONS            * TH
23A0 09         7200            DB    FRIC+DENTAL
23A1 40         7210            DB    CONS            * S
23A2 09         7220            DB    FRIC+DENTAL
23A3 40         7230            DB    CONS            * SH
23A4 08         7240            DB    FRIC
23A5 40         7250            DB    CONS            * V
23A6 48         7260            DB    FRIC+VOICE
23A7 40         7270            DB    CONS            * DH
23A8 49         7280            DB    FRIC+VOICE+DENTAL
23A9 40         7290            DB    CONS            * Z
23AA 49         7300            DB    FRIC+VOICE+DENTAL
23AB 40         7310            DB    CONS            * ZH
23AC 48         7320            DB    FRIC+VOICE
23AD 00         7330            DB    0           * CH
23AE 00         7340            DB    0
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

23AF 00         7350            DB    0          * JH
23B0 00         7360            DB    0
23B1 40         7370            DB    CONS       * HH
23B2 00         7380            DB    0
23B3 40         7390            DB    CONS       * Q
23B4 C0         7400            DB    STOP+VOICE
23B5            7410 *
23B5            7420 *  END OF SECT1
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  RULES, GROUPS 1 AND 2
2000            0020 *
2000            0030 *  SECTION 2 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.07   MAY 30, 1977
2000            0070 *
2000            0080 **********
2000            0090 *
2000            0100 *  COMMON JUMP ADDRESS TABLE
2000            0110 *  THIS TABLE ALLOWS REASSEMBLING ANY SECTION
2000            0120 *  WITHOUT CHANGING REFERENCES IN OTHER SECTIONS
2000            0130 *
2000            0140 **********
2000            0150 *
2000            0160 COMJMP     EQU   $
2000            0170 *
2000            0180 CSR1       DS    3
2003            0190 PLAY       DS    3
2006            0200 BUFADR     DS    2
2008            0210 BUFEND     DS    2
200A            0220 PVTAB      DS    2
200C            0230 MATPAK     DS    3
200F            0240 MATERR     DS    3
2012 C3 C0 23   0250             JMP   RULES
2015            0260 SETDUR     DS    3
2018            0270 RULES3     DS    3
201B            0280 GENFO      DS    3
201E            0290 CLRBUF     DS    3
2021            0300 GENPRM     DS    3
2024            0310 DUMMY      DS    22
203A            0320 *
203A            0330 **********
203A            0340 *
203A            0350 *  COMRAM ORIGEN DEFINITION
203A            0360 *
203A            0370            ORG   COMJMP+1500H
3500            0380 COMRAM     EQU   $
3500            0390 *
3500            0400 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0410 *
3500            0420 MATPTR     DS    2
3502            0430 NEGEND     DS    2
3504            0440 MATRIX     EQU   $
3504            0450 MATLEN     EQU   95
3504            0460 PHCODE     DS    MATLEN
3563            0470 FEATA      DS    MATLEN
35C2            0480 FEATB      DS    MATLEN
3621            0490 STRES      DS    MATLEN
3680            0500 DUR        DS    MATLEN
36DF            0510 MATEND     EQU   $
36DF            0520 *
36DF            0530 *  RULES (LOCAL) RAM WORKSPACE
36DF            0540 *
36DF            0550 MOVAD      DS    2
36E1            0560 COUNT      DS    1
36E2            0570 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

36E2            0580 **********
36E2            0590 *
36E2            0600 *  RULES   GROUPS 1 AND 2
36E2            0610 *
36E2            0620            ORG   COMJMP+3C0H
23C0            0630 *
23C0            0640 *  PHONEME CODE DEFINITIONS FOR RULES
23C0            0650 *
23C0            0660 CSPACE     EQU   0
23C0            0670 CTERM      EQU   4
23C0            0680 CUW        EQU   14
23C0            0690 CUX        EQU   18
23C0            0700 CRX        EQU   24
23C0            0710 CLX        EQU   25
23C0            0720 CWX        EQU   26
23C0            0730 CYX        EQU   27
23C0            0740 CWH        EQU   28
23C0            0750 CR         EQU   32
23C0            0760 CW         EQU   34
23C0            0770 CL         EQU   33
23C0            0780 CM         EQU   36
23C0            0790 CN         EQU   37
23C0            0800 CNX        EQU   38
23C0            0810 CP         EQU   39
23C0            0820 CT         EQU   40
23C0            0830 CK         EQU   41
23C0            0840 CKX        EQU   42
23C0            0850 CB         EQU   43
23C0            0860 CD         EQU   44
23C0            0870 CG         EQU   45
23C0            0880 CGX        EQU   46
23C0            0890 CDX        EQU   47
23C0            0900 CS         EQU   50
23C0            0910 CSH        EQU   51
23C0            0920 CZ         EQU   54
23C0            0930 CZH        EQU   55
23C0            0940 CHH        EQU   58
23C0            0950 CQ         EQU   59
23C0            0960 *
23C0            0970 *  DEFINE FEATURE LABELS
23C0            0980 *
23C0            0990 VOWEL      EQU   80H
23C0            1000 CONS       EQU   40H
23C0            1010 FRONT      EQU   20H
23C0            1020 DIPHTH     EQU   10H
23C0            1030 WDBND      EQU   4
23C0            1040 PHBND      EQU   2
23C0            1050 IGNORE     EQU   1
23C0            1060 STOP       EQU   80H
23C0            1070 VOICE      EQU   40H
23C0            1080 PLOS       EQU   20H
23C0            1090 PLOSA      EQU   10H
23C0            1100 FRIC       EQU   8
23C0            1110 LIQUID     EQU   4
23C0            1120 NASAL      EQU   2
23C0            1130 DENTAL     EQU   1
23C0            1140 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

23C0            1150 *
23C0            1160 ************
23C0            1170 *
23C0            1180 *  RULES CODE
23C0            1190 *
23C0            1200 ************
23C0            1210 *
23C0            1220 *  RULES MAIN
23C0            1230 *
23C0 CD D8 23   1240 RULES      CALL  RULES1  * APPLY RULE GROUP 1
23C3 C0         1250            RNZ   *       * RETURN IF ERROR
23C4 00         1260            NOP
23C5 00         1270            NOP
23C6 00         1280            NOP
23C7 CD 78 24   1290            CALL  RULES2  * APPLY RULE GROUP 2
23CA 00         1300            NOP
23CB 00         1310            NOP
23CC 00         1320            NOP
23CD CD 15 20   1330            CALL  SETDUR  * ASSIGN INITIAL DURATIONS
23D0 00         1340            NOP
23D1 00         1350            NOP
23D2 00         1360            NOP
23D3 CD 18 20   1370            CALL  RULES3  * APPLY RULE GROUP 3
23D6 AF         1380            XRA   A       * CLEAR ERROR CONDITION
23D7 C9         1390            RET
23D8            1400 *
23D8            1410 *
23D8            1420 **********
23D8            1430 *
23D8            1440 *  RULE GROUP 1
23D8            1450 *
23D8 21 05 35   1460 RULES1     LXI   H,MATRIX+1
23DB 22 00 35   1470            SHLD  MATPTR
23DE 01 5F 00   1480            LXI   B,MATLEN
23E1            1490 R1LOOP     EQU   $
23E1            1500 *
23E1            1510 *  Q <= VOWEL STRESS.GT.0,SPACE//VOWEL STRESS.GT.0
23E1            1520 *
23E1 09         1530 R1A        DAD   B
23E2 7E         1540            MOV   A,M     * GET COL X FEATA
23E3 E6 80      1550            ANI   VOWEL
23E5 CA 1C 24   1560            JZ    R1B     * NOT A VOWEL
23E8 09         1570            DAD   B
23E9 09         1580            DAD   B
23EA 7E         1590            MOV   A,M     * GET COL X STRESS
23EB B7         1600            ORA   A
23EC CA 1C 24   1610            JZ    R1B     * VOWEL NOT STRESSED
23EF 2A 00 35   1620            LHLD  MATPTR
23F2 2B         1630            DCX   H
23F3 7E         1640            MOV   A,M     * GET COL X-1 CODE
23F4 FE 04      1650            CPI   CTERM
23F6 CA 65 24   1660            JZ    ENDR1   * X IS 1ST COLUMN
23F9 FE 00      1670            CPI   CSPACE
23FB C2 1C 24   1680            JNZ   R1B     * NOT A WORD BOUNDARY
23FE 2B         1690            DCX   H
23FF 7E         1700            MOV   A,M     * GET COL X-2 CODE
2400 FE 04      1710            CPI   CTERM
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2402 CA 1C 24   1720            JZ    R1B
2405 09         1730            DAD   B
2406 7E         1740            MOV   A,M       * GET COL X-2 FEATA
2407 E6 80      1750            ANI   VOWEL
2409 CA 1C 24   1760            JZ    R1B       * NOT A VOWEL
240C 09         1770            DAD   B
240D 09         1780            DAD   B
240E 7E         1790            MOV   A,M       * GET COL X-2 STRESS
240F B7         1800            ORA   A
2410 CA 1C 24   1810            JZ    R1B       * STRESS MUST BE NON 0
2413 3E 3B      1820            MVI   A,CQ
2415 CD 0A 27   1830            CALL  MATNSR    * OK, INSERT A GLOTTAL STOP
2418 C0         1840            RNZ
2419 C3 65 24   1850            JMP   ENDR1
241C            1860 *
241C            1870 *  RX <= VOWEL/R/
241C            1880 *
241C 2A 00 35   1890 R1B        LHLD  MATPTR
241F 7E         1900            MOV   A,M
2420 FE 20      1910            CPI   CR        * TEST FOR R
2422 C2 2A 24   1920            JNZ   R1C
2425 16 18      1930            MVI   D,CRX     * CHANGE? TO RX
2427 C3 31 24   1940            JMP   R1BC
242A            1950 *
242A            1960 *  LX <= VOWEL/L/
242A            1970 *
242A FE 21      1980 R1C        CPI   CL        * TEST FOR L
242C C2 46 24   1990            JNZ   R1D
242F 16 19      2000            MVI   D,CLX     * CHANGE? TO LX
2431 2B         2010 R1BC       DCX   H
2432 7E         2020            MOV   A,M       * GET COL X-1 CODE
2433 FE 04      2030            CPI   CTERM
2435 CA 65 24   2040            JZ    ENDR1
2438 09         2050            DAD   B
2439 7E         2060            MOV   A,M       * GET COL X-1 FEATA
243A E6 80      2070            ANI   VOWEL
243C CA 65 24   2080            JZ    ENDR1     * NOT A VOWEL
243F 7A         2090            MOV   A,D
2440 CD 0C 20   2100            CALL  MATPAK    * OK, CHANGE THE CONSONANT
2443 2A 00 35   2110            LHLD  MATPTR
2446            2120 *
2446            2130 *  YX <= DIPHTH FRONT//
2446            2140 *  WX <= DIPHTH -FRONT//
2446            2150 *
2446 2B         2160 R1D        DCX   H
2447 7E         2170            MOV   A,M
2448 FE 04      2180            CPI   CTERM
244A CA 65 24   2190            JZ    ENDR1
244D 09         2200            DAD   B
244E 7E         2210            MOV   A,M       * GET COL X-1 FEATA
244F E6 10      2220            ANI   DIPHTH
2451 CA 65 24   2230            JZ    ENDR1     * NOT A DIPHTHONG
2454 7E         2240            MOV   A,M
2455 E6 20      2250            ANI   FRONT
2457 CA 5F 24   2260            JZ    BACK
245A 3E 1B      2270            MVI   A,CYX
245C C3 61 24   2280            JMP   GLIDE
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

245F 3E 1A      2290 BACK        MVI   A,CWX
2461 CD 0A 27   2300 GLIDE       CALL  MATNSR  * INSERT GLIDE AFTER DIPHTH
2464 C0         2310             RNZ
2465            2320 *
2465            2330 *  END OF RULE GROUP 1
2465            2340 *
2465 00         2350 ENDR1       NOP   *       * BREAKPOINT LOC
2466 2A 00 35   2360             LHLD  MATPTR
2469 23         2370             INX   H
246A 22 00 35   2380             SHLD  MATPTR  * STEP POINTER
246D AF         2390             XRA   A       * CLEAR ERROR CONDITION
246E EB         2400             XCHG
246F 2A 02 35   2410             LHLD  NEGEND
2472 19         2420             DAD   D
2473 D8         2430             RC    *       * RETURN IF END OF MATRIX
2474 EB         2440             XCHG
2475 C3 E1 23   2450             JMP   R1LOOP
2478            2460 *
2478            2470 **********
2478            2480 *
2478            2490 *  RULE GROUP 2
2478            2500 *
2478 21 05 35   2510 RULES2      LXI   H,MATRIX+1
247B 22 00 35   2520             SHLD  MATPTR
247E 01 9F 00   2530             LXI   B,MATLEN
2481            2540 R2LOOP      EQU   $
2481            2550 *
2481            2560 *  STRESSX=1 <= /CONS STRESS.GE.0/VOWEL STRESS.NE.0
2481            2570 *
2481 09         2580 R2A         DAD   B
2482 7E         2590             MOV   A,M     * GET COL X FEATA
2483 E6 40      2600             ANI   CONS
2485 CA FD 25   2610             JZ    R2G     * NOT A CONSONANT
2488 09         2620             DAD   B
2489 09         2630             DAD   B
248A 7E         2640             MOV   A,M     * GET COL X STRESS
248B B7         2650             ORA   A
248C FA AA 24   2660             JM    R2B     * STRESS IS -, DON'T CHANGE IT
248F 2A 00 35   2670             LHLD  MATPTR
2492 23         2680             INX   H
2493 7E         2690             MOV   A,M     * GET COL X+1 CODE
2494 FE 04      2700             CPI   CTERM
2496 CA FD 25   2710             JZ    R2G
2499 09         2720             DAD   B
249A 7E         2730             MOV   A,M     * GET COL X+1 FEATA
249B E6 80      2740             ANI   VOWEL
249D CA AA 24   2750             JZ    R2B     * NOT A VOWEL
24A0 09         2760             DAD   B
24A1 09         2770             DAD   B
24A2 7E         2780             MOV   A,M     * GET COL X+1 STRESS
24A3 B7         2790             ORA   A
24A4 CA 7D 25   2800             JZ    R2E     * VOWEL NOT STRESSED
24A7 2B         2810             DCX   H
24A8 36 01      2820             MVI   M,1     * OK, SET CONSONANT STRESS = 1
24AA            2830 *
24AA            2840 *  STRESSX,X+1=-1 <= /S,PLOS -VOICE/VOWEL STRESS.NE:0
24AA            2850 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

24AA 2A 00 35   2860 R2B        LHLD  MATPTR
24AD 7E         2870            MOV   A,M
24AE FE 32      2880            CPI   CS
24B0 C2 E3 24   2890            JNZ   R2C      * COL X NOT AN S
24B3 23         2900            INX   H
24B4 7E         2910            MOV   A,M      * GET COL X+1 CODE
24B5 FE 04      2920            CPI   CTERM
24B7 CA F8 26   2930            JZ    ENDR2    * S IN LAST COL, END GROUP
24BA 09         2940            DAD   B
24BB 09         2950            DAD   B
24BC 7E         2960            MOV   A,M      * GET COL X+1 FEATB
24BD EE 20      2970            XRI   PLOS
24BF E6 60      2980            ANI   PLOS+VOICE
24C1 C2 E3 24   2990            JNZ   R2C      * X+1 IS NOT UNVOICED PLOSIVE
24C4 2A 00 35   3000            LHLD  MATPTR
24C7 23         3010            INX   H
24C8 23         3020            INX   H
24C9 7E         3030            MOV   A,M      * GET COL X+2 CODE
24CA FE 04      3040            CPI   CTERM
24CC CA F8 26   3050            JZ    ENDR2
24CF 09         3060            DAD   B
24D0 7E         3070            MOV   A,M      * GET COL X+2 FEATA
24D1 E6 80      3080            ANI   VOWEL
24D3 CA E3 24   3090            JZ    R2C      * NOT A VOWEL
24D6 09         3100            DAD   B
24D7 09         3110            DAD   B
24D8 7E         3120            MOV   A,M
24D9 B7         3130            ORA   A
24DA CA E3 24   3140            JZ    R2C      * X+2 STRESS MUST BE NON 0
24DD 2B         3150            DCX   H
24DE 36 FF      3160            MVI   M,-1     * SET STRESS X+1 =-1
24E0 2B         3170            DCX   H
24E1 36 FF      3180            MVI   M,-1     * SET STRESS X =-1
24E3            3190 *
24E3            3200 *  STRESSX,X+1=-1 <= /PLOS OR (FRIC -VOICE),LIQUID
24E3            3210 *                       OR NASAL/VOWEL STRESS.NE.0
24E3            3220 *
24E3 2A 00 35   3230 R2C        LHLD  MATPTR
24E6 09         3240            DAD   B
24E7 09         3250            DAD   B
24E8 7E         3260            MOV   A,M      * GET COL X FEATB
24E9 E6 20      3270            ANI   PLOS
24EB C2 F6 24   3280            JNZ   R2C1     * COL X IS PLOSIVE
24EE 7E         3290            MOV   A,M
24EF EE 08      3300            XRI   FRIC
24F1 E6 48      3310            ANI   FRIC+VOICE
24F3 C2 2A 25   3320            JNZ   R2D      * NEITHER PLOS NOR FRIC -VOICE
24F6 2A 00 35   3330 R2C1       LHLD  MATPTR
24F9 23         3340            INX   H
24FA 7E         3350            MOV   A,M      * GET COL X+1 CODE
24FB FE 04      3360            CPI   CTERM
24FD CA 42 26   3370            JZ    R2JK
2500 09         3380            DAD   B
2501 09         3390            DAD   B
2502 7E         3400            MOV   A,M      * GET COL X+1 FEATB
2503 E6 06      3410            ANI   LIQUID+NASAL
2505 CA 2A 25   3420            JZ    R2D      * NEITHER LIQUID NOR NASAL
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2508 2A 00 35   3430 R2C2      LHLD  MATPTR
250B 23         3440           INX   H
250C 23         3450           INX   H
250D 7E         3460           MOV   A,M      * GET COL X+2 CODE
250E FE 04      3470           CPI   CTERM
2510 CA 42 26   3480           JZ    R2JK
2513 09         3490           DAD   B
2514 7E         3500           MOV   A,M      * GET COL X+2 FEATA
2515 E6 80      3510           ANI   VOWEL
2517 CA 42 26   3520           JZ    R2JK     * NOT A VOWEL
251A 09         3530           DAD   B
251B 09         3540           DAD   B
251C 7E         3550           MOV   A,M
251D B7         3560           ORA   A
251E CA 42 26   3570           JZ    R2JK     * VOWEL X+2 NOT STRESSED
2521 2B         3580           DCX   H
2522 36 FF      3590           MVI   M,-1     * PUT -1 STRESS IN COL X+1
2524 2B         3600           DCX   H
2525 36 FF      3610           MVI   M,-1     * PUT -1 STRESS IN COL X
2527 C3 42 26   3620           JMP   R2JK
252A            3630 *
252A            3640 *  STRESSX,X+1,X+2=-1 <= /S,PLOS -VOICE,LIQUID/
252A            3650 *                        VOWEL STRESS.NE.0
252A            3660 *
252A 2A 00 35   3670 R2D       LHLD  MATPTR
252D 7E         3680           MOV   A,M
252E FE 32      3690           CPI   CS
2530 C2 7D 25   3700           JNZ   R2E      * THIS COL NOT AN S
2533 23         3710           INX   H
2534 7E         3720           MOV   A,M
2535 FE 04      3730           CPI   CTERM
2537 CA F8 26   3740           JZ    ENDR2    * S IN LAST COL, END GROUP 2
253A 09         3750           DAD   B
253B 09         3760           DAD   B
253C 7E         3770           MOV   A,M      * GET COL X+1 FEATB
253D EE 20      3780           XRI   PLOS
253F E6 60      3790           ANI   PLOS+VOICE
2541 C2 F8 26   3800           JNZ   ENDR2    * COL X+1 NOT UNVOICED PLOSIVE
2544 2A 00 35   3810           LHLD  MATPTR
2547 23         3820           INX   H
2548 23         3830           INX   H
2549 7E         3840           MOV   A,M      * GET COL X+2 CODE
254A FE 04      3850           CPI   CTERM
254C CA F8 26   3860           JZ    ENDR2    * LAST COL, END OF GROUP
254F 09         3870           DAD   B
2550 09         3880           DAD   B
2551 7E         3890           MOV   A,M      * GET COL X+2 FEATB
2552 E6 04      3900           ANI   LIQUID
2554 CA F8 26   3910           JZ    ENDR2    * NOT A LIQUID
2557 2A 00 35   3920           LHLD  MATPTR
255A 23         3930           INX   H
255B 23         3940           INX   H
255C 23         3950           INX   H
255D 7E         3960           MOV   A,M      * GET COL X+3 CODE
255E FE 04      3970           CPI   CTERM
2560 CA F8 26   3980           JZ    ENDR2
2563 09         3990           DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2564 7E         4000            MOV   A,M      * GET COL X+3 FEATA
2565 E6 80      4010            ANI   VOWEL
2567 CA F8 26   4020            JZ    ENDR2    * NOT A VOWEL
256A 09         4030            DAD   B
256B 09         4040            DAD   B
256C 7E         4050            MOV   A,M      * GET COL X+3 STRESS
256D B7         4060            ORA   A
256E CA F8 26   4070            JZ    ENDR2    * VOWEL MUST BE STRESSED
2571 2B         4080            DCX   H
2572 36 FF      4090            MVI   M,-1     * SET STRESS X+2 = -1
2574 2B         4100            DCX   H
2575 36 FF      4110            MVI   M,-1     * SET STRESS X+1 = -1
2577 2B         4120            DCX   H
2578 36 FF      4130            MVI   M,-1     * SET STRESS X = -1
257A C3 F8 26   4140            JMP   ENDR2
257D            4150 *
257D            4160 *  STRESSX,X+1=-1 <= /T OR D,SH OR ZH/VOWEL STRESS.NE.0
257D            4170 *
257D 2A 00 35   4180 R2E        LHLD  MATPTR
2580 7E         4190            MOV   A,M      * GET COL X CODE
2581 FE 28      4200            CPI   CT
2583 CA 8B 25   4210            JZ    R2E1     * IT'S A T
2586 FE 2C      4220            CPI   CD
2588 C2 1D 26   4230            JNZ   R2H      * IT'S NEITHER T NOR D
258B 23         4240 R2E1       INX   H
258C 7E         4250            MOV   A,M      * GET COL X+1 CODE
258D FE 04      4260            CPI   CTERM
258F CA 42 26   4270            JZ    R2JK     * LAST COL
2592 FE 33      4280            CPI   CSH
2594 CA 9C 25   4290            JZ    R2E2     * COL X+1 IS SH
2597 FE 37      4300            CPI   CZH
2599 C2 BA 25   4310            JNZ   R2F      * X+1 IS NEITHER SH OR ZH
259C 23         4320 R2E2       INX   H
259D 7E         4330            MOV   A,M      * GET COL X+2 CODE
259E FE 04      4340            CPI   CTERM
25A0 CA 42 26   4350            JZ    R2JK     * LAST COL
25A3 09         4360            DAD   B
25A4 7E         4370            MOV   A,M      * GET COL X+2 FEATA
25A5 E6 80      4380            ANI   VOWEL
25A7 CA 42 26   4390            JZ    R2JK     * X+2 NOT A VOWEL
25AA 09         4400            DAD   B
25AB 09         4410            DAD   B
25AC 7E         4420            MOV   A,M      * GET COL X+2 STRESS
25AD B7         4430            ORA   A
25AE CA 42 26   4440            JZ    R2JK     * VOWEL X+2 NOT STRESSED
25B1 2B         4450            DCX   H
25B2 36 FF      4460            MVI   M,-1     * SET STRESS X+1 = -1
25B4 2B         4470            DCX   H
25B5 36 FF      4480            MVI   M,-1     * SET STRESS X = -1
25B7 C3 42 26   4490            JMP   R2JK
25BA            4500 *
25BA            4510 *  DX <= VOWEL/T OR D/(WDBND,VOWEL) OR VOWEL STRESS.EQ.0
25BA            4520 *
25BA 2A 00 35   4530 R2F        LHLD  MATPTR
25BD 2B         4540            DCX   H
25BE 7E         4550            MOV   A,M      * GET COL X-1 CODE
25BF FE 04      4560            CPI   CTERM
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

25C1 CA 42 26    4570           JZ    R2JK     * COL X IS 1ST COL
25C4 09          4580           DAD   B
25C5 7E          4590           MOV   A,M      * GET COL X-1 FEATA
25C6 E6 80       4600           ANI   VOWEL
25C8 CA 42 26    4610           JZ    R2JK     * NOT A VOWEL
25CB 23          4620           INX   H
25CC 23          4630           INX   H
25CD 7E          4640           MOV   A,M      * GET COL X+1 FEATA
25CE E6 04       4650           ANI   WDBND
25D0 C2 E3 25    4660           JNZ   R2F2     * IT'S A WORD BOUNDARY
25D3 7E          4670           MOV   A,M      * GET FEAT A AGAIN
25D4 E6 80       4680           ANI   VOWEL
25D6 CA 42 26    4690           JZ    R2JK     * IT'S NOT A VOWEL
25D9 09          4700           DAD   B
25DA 09          4710           DAD   B
25DB 7E          4720           MOV   A,M      * GET COL X+1 STRESS
25DC B7          4730           ORA   A
25DD CA F5 25    4740           JZ    R2F3     * VOWEL IS UNSTRESSED, DO THE RULE
25E0 C3 42 26    4750           JMP   R2JK
25E3 2A 00 35    4760 R2F2      LHLD  MATPTR
25E6 23          4770           INX   H
25E7 23          4780           INX   H
25E8 7E          4790           MOV   A,M      * GET COL X+2 CODE
25E9 FE 04       4800           CPI   CTERM
25EB CA 42 26    4810           JZ    R2JK
25EE 09          4820           DAD   B
25EF 7E          4830           MOV   A,M
25F0 E6 80       4840           ANI   VOWEL
25F2 CA 42 26    4850           JZ    R2JK     * IT'S NOT A VOWEL
25F5 3E 2F       4860 R2F3      MVI   A,CDX
25F7 CD 0C 20    4870           CALL  MATPAK
25FA C3 F8 26    4880           JMP   ENDR2
25FD             4890 *
25FD             4900 *  UX <= DENTAL/UW/
25FD             4910 *
25FD 2A 00 35    4920 R2G       LHLD  MATPTR
2600 7E          4930           MOV   A,M      * GET COL X CODE
2601 FE 0E       4940           CPI   CUW
2603 C2 1D 26    4950           JNZ   R2H
2606 2B          4960           DCX   H
2607 7E          4970           MOV   A,M      * GET COL X-1 CODE
2608 FE 04       4980           CPI   CTERM
260A CA F8 26    4990           JZ    ENDR2    * UW IS IN COL 1, END OF GROUP
260D 09          5000           DAD   B
260E 09          5010           DAD   B
260F 7E          5020           MOV   A,M      * GET COL X-1 FEATB
2610 E6 01       5030           ANI   DENTAL
2612 CA F8 26    5040           JZ    ENDR2    * NOT A DENTAL CONSONANT
2615 3E 12       5050           MVI   A,CUX
2617 CD 0C 20    5060           CALL  MATPAK   * CHANGE COL TO UX
261A C3 F8 26    5070           JMP   ENDR2
261D             5080 *
261D             5090 *  KX <= /K/VOWEL -FRONT
261D             5100 *
261D FE 29       5110 R2H       CPI   CK
261F C2 27 26    5120           JNZ   R2I      * COL X MUST BE EITHER K, ....
2622 1E 2A       5130           MVI   E,CKX
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2624 C3 2E 26    5140           JMP   R2HI
2627             5150 *
2627             5160 *  GX <= /G/VOWEL -FRONT
2627             5170 *
2627 FE 2D       5180 R2I       CPI   CG
2629 C2 42 26    5190           JNZ   R2JK     * ... OR G
262C 1E 2E       5200           MVI   E,CGX
262E 23          5210 R2HI      INX   H
262F 7E          5220           MOV   A,M      * GET COL X+1 CODE
2630 FE 04       5230           CPI   CTERM
2632 CA 42 26    5240           JZ    R2JK     * PLOSIVE IS IN LAST COL
2635 09          5250           DAD   B
2636 7E          5260           MOV   A,M      * GET COL X+1 FEATA
2637 EE 80       5270           XRI   VOWEL
2639 E6 A0       5280           ANI   VOWEL+FRONT
263B C2 42 26    5290           JNZ   R2JK     * NOT A VOWEL -FRONT
263E 7B          5300           MOV   A,E
263F CD 0C 20    5310           CALL  MATPAK   * CHANGE COL X TO BACK CONSONANT
2642             5320 *
2642             5330 *  ADD 4 TO CODE <= S/PLOS -VOICE/
2642             5340 *  ADD 4 TO CODE <= /PLOS -VOICE/WORDBOUND
2642             5350 *
2642 2A 00 35    5360 R2JK      LHLD  MATPTR
2645 09          5370           DAD   B
2646 09          5380           DAD   B
2647 7E          5390           MOV   A,M      * GET COL X FEATB
2648 EE 20       5400           XRI   PLOS
264A E6 60       5410           ANI   PLOS+VOICE
264C C2 76 26    5420           JNZ   R2L      * NOT AN UNVOICED PLOSIVE
264F 2A 00 35    5430           LHLD  MATPTR
2652 2B          5440           DCX   H
2653 7E          5450           MOV   A,M      * GET COL X-1 CODE
2654 FE 04       5460           CPI   CTERM
2656 CA 5E 26    5470           JZ    R2JK1    * PLOSIVE IS IN 1ST COL
2659 FE 32       5480           CPI   CS
265B CA 6D 26    5490           JZ    R2JK2    * COL X-1 IS S, DO THE RULE
265E 23          5500 R2JK1     INX   H
265F 23          5510           INX   H
2660 7E          5520           MOV   A,M      * GET COL X+1 CODE
2661 FE 04       5530           CPI   CTERM
2663 CA F8 26    5540           JZ    ENDR2    * PLOSIVE IS IN LAST COL
2666 09          5550           DAD   B
2667 7E          5560           MOV   A,M      * GET COL X+1 FEATA
2668 E6 06       5570           ANI   WDBND+PHBND
266A CA 76 26    5580           JZ    R2L      * NOT A BOUNDARY
266D 2A 00 35    5590 R2JK2     LHLD  MATPTR
2670 7E          5600           MOV   A,M
2671 C6 04       5610           ADI   4
2673 CD 0C 20    5620           CALL  MATPAK   * ADD 4 TO CODE IN COL X
2676             5630 *
2676             5640 *  -PLOS -PLOSA <= /PLOS/OPTIONAL WDBOUND,STOP
2676             5650 *
2676 2A 00 35    5660 R2L       LHLD  MATPTR
2679 54          5670           MOV   D,H
267A 5D          5680           MOV   E,L
267B 09          5690           DAD   B
267C 09          5700           DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

267D 7E         5710            MOV   A,M       * GET COL X FEATB
267E E6 20      5720            ANI   PLOS
2680 CA F8 26   5730            JZ    ENDR2     * NOT PLOSIVE, END GROUP 2
2683 EB         5740            XCHG  *         * LEAVE DE AT COL X FEATB
2684 23         5750            INX   H
2685 7E         5760            MOV   A,M       * GET COL X+1 CODE
2686 FE 04      5770            CPI   CTERM
2688 CA F8 26   5780            JZ    ENDR2     * PLOS IN LAST COL, END GP 2
268B FE 00      5790            CPI   CSPACE
268D C2 97 26   5800            JNZ   R2L2      * X+1 IS NOT WDBOUND, Y=X+1
2690 23         5810            INX   H         * X+1 IS WDBOUND, Y=X+2
2691 7E         5820            MOV   A,M
2692 FE 04      5830            CPI   CTERM
2694 CA F8 26   5840            JZ    ENDR2     * WDBOUND IS LAST COL
2697 09         5850 R2L2       DAD   B
2698 09         5860            DAD   B
2699 7E         5870            MOV   A,M       * GET COL Y FEATB
269A E6 80      5880            ANI   STOP
269C CA A6 26   5890            JZ    R2M
269F 1A         5900            LDAX  D         * GET COL X FEATB
26A0 E6 CF      5910            ANI   255-PLOS-PLOSA
26A2 12         5920            STAX  D         * DELETE FEATURES PLOS & PLOSA
26A3 C3 F8 26   5930            JMP   ENDR2
26A6            5940 *
26A6            5950 *  -PLOSA <= /PLOSA/OPTIONAL WDBOUND,WH OR HH
26A6            5960 *
26A6 2A 00 35   5970 R2M        LHLD  MATPTR
26A9 54         5980            MOV   D,H
26AA 5D         5990            MOV   E,L
26AB 09         6000            DAD   B
26AC 09         6010            DAD   B
26AD 7E         6020            MOV   A,M       * GET COL X FEATB
26AE E6 10      6030            ANI   PLOSA
26B0 CA F8 26   6040            JZ    ENDR2     * NOT PLOSIVE ASPIRATE, END 2
26B3 EB         6050            XCHG  *         * LEAVE DE AT COL X FEATB
26B4 23         6060            INX   H
26B5 7E         6070            MOV   A,M       * GET COL X+1 CODE
26B6 FE 04      6080            CPI   CTERM
26B8 CA F8 26   6090            JZ    ENDR2     * PLOS ASP IS IN LAST COL
26BB FE 00      6100            CPI   CSPACE
26BD C2 C7 26   6110            JNZ   R2M2      * X+1 IS NOT WDBOUND, Y=X+1
26C0 23         6120            INX   H         * X+1 IS WDBOUND, Y=X+2
26C1 7E         6130            MOV   A,M       * GET COL Y CODE
26C2 FE 04      6140            CPI   CTERM
26C4 CA F8 26   6150            JZ    ENDR2     * WDBOUND IS LAST COL
26C7 FE 1C      6160 R2M2       CPI   CWH
26C9 CA F4 26   6170            JZ    R2MN      * COL Y IS WH, DO THE RULE
26CC FE 3A      6180            CPI   CHH
26CE CA F4 26   6190            JZ    R2MN      * COL Y IS HH, DO THE RULE
26D1            6200 *
26D1            6210 *  -PLOSA <= -BOUNDARY/PLOSA/VOWEL STRESS.EQ.0
26D1            6220 *
26D1 2A 00 35   6230 R2N        LHLD  MATPTR
26D4 23         6240            INX   H
26D5 09         6250            DAD   B
26D6 7E         6260            MOV   A,M       * GET COL X+1 FEATA
26D7 E6 80      6270            ANI   VOWEL
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

26D9 CA F8 26   6280            JZ    ENDR2    * NOT A VOWEL
26DC 09         6290            DAD   B
26DD 09         6300            DAD   B
26DE 7E         6310            MOV   A,M      * GET COL X+1 STRESS
26DF B7         6320            ORA   A
26E0 C2 F8 26   6330            JNZ   ENDR2    * VOWEL IS STRESSED, END GP 2
26E3 2A 00 35   6340            LHLD  MATPTR
26E6 2B         6350            DCX   H
26E7 7E         6360            MOV   A,M      * GET COL X-1 CODE
26E8 FE 04      6370            CPI   CTERM
26EA CA F8 26   6380            JZ    ENDR2
26ED 09         6390            DAD   B
26EE 7E         6400            MOV   A,M      * GET COL X-1 FEATA
26EF E6 06      6410            ANI   WDBND+PHBND
26F1 C2 F8 26   6420            JNZ   ENDR2    * EITHER BOUNDARY STOPS THE RULE
26F4 1A         6430 R2MN       LDAX  D        * GET COL X FEATB
26F5 E6 EF      6440            ANI   255-PLOSA
26F7 12         6450            STAX  D        * DELETE COL X PLOSA FEATURE
26F8            6460 *
26F8            6470 *  END OF RULE GROUP 2
26F8            6480 *
26F8 00         6490 ENDR2      NOP   *        * BREAKPOINT LOC
26F9 2A 00 35   6500            LHLD  MATPTR
26FC 23         6510            INX   H
26FD 22 00 35   6520            SHLD  MATPTR
2700 EB         6530            XCHG
2701 2A 02 35   6540            LHLD  NEGEND
2704 19         6550            DAD   D
2705 D8         6560            RC    *        * RETURN IF AT END OF MATRIX
2706 EB         6570            XCHG
2707 C3 81 24   6580            JMP   R2LOOP
270A            6590 *
270A            6600 **********
270A            6610 *
270A            6620 *  SUBROUTINES
270A            6630 *
270A            6640 *  MOVE DATA IN MATRIX AND INSERT A COLUMN
270A            6650 *
270A 11 61 35   6660 MATNSR     LXI   D,MATRIX+MATLEN-2
270D 2A 02 35   6670            LHLD  NEGEND   * IS THERE ROOM TO INSERT
2710 19         6680            DAD   D
2711 D2 0F 20   6690            JNC   MATERR   * NO
2714 F5         6700            PUSH  PSW      * YES, SAVE THE NEW CODE
2715 2A 00 35   6710            LHLD  MATPTR   * COMPUTE # OF COLS TO MOVE
2718 EB         6720            XCHG
2719 2A 02 35   6730            LHLD  NEGEND
271C 19         6740            DAD   D
271D 7D         6750            MOV   A,L
271E 2F         6760            CMA
271F 3C         6770            INR   A
2720 3C         6780            INR   A
2721 32 E1 36   6790            STA   COUNT
2724 6F         6800            MOV   L,A      * COMPUTE NEW ADDR OF LAST COL
2725 26 00      6810            MVI   H,0
2727 19         6820            DAD   D
2728 22 DF 36   6830            SHLD  MOVAD
272B 2A DF 36   6840 INSR1      LHLD  MOVAD
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

272E 54         6850            MOV   D,H
272F 5D         6860            MOV   E,L
2730 1B         6870            DCX   D
2731 3A E1 36   6880            LDA   COUNT
2734 4F         6890            MOV   C,A
2735 1A         6900 INSR2      LDAX  D       * MOVE A ROW RIGHT 1 LOC
2736 77         6910            MOV   M,A
2737 1B         6920            DCX   D
2738 2B         6930            DCX   H
2739 0D         6940            DCR   C
273A C2 35 27   6950            JNZ   INSR2
273D 2A DF 36   6960            LHLD  MOVAD   * UPDATE NEW POINTER
2740 11 5F 00   6970            LXI   D,MATLEN
2743 19         6980            DAD   D
2744 22 DF 36   6990            SHLD  MOVAD
2747 11 21 C9   7000            LXI   D,-MATEND * ANY MORE ROWS?
274A 19         7010            DAD   D
274B D2 2B 27   7020            JNC   INSR1   * YES
274E F1         7030            POP   PSW     * GET THE NEW CODE
274F CD 0C 20   7040            CALL  MATPAK  * SET CODE AND FEATURES
2752 09         7050            DAD   B
2753 36 00      7060            MVI   M,0     * PUT 0 STRESS IN NEW COL
2755 09         7070            DAD   B
2756 36 00      7080            MVI   M,0     * PUT 0 DURATION IN NEW COL
2758 2A 02 35   7090            LHLD  NEGEND  * DONE, DECREMENT NEGEND
275B 2B         7100            DCX   H
275C 22 02 35   7110            SHLD  NEGEND
275F 2A 00 35   7120            LHLD  MATPTR  * AND INCREMENT MATPTR
2762 23         7130            INX   H
2763 22 00 35   7140            SHLD  MATPTR
2766 AF         7150            XRA   A       * CLEAR ERROR CONDITION
2767 C9         7160            RET
2768            7170 *
2768            7180 *  END OF SECT2
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  RULES GROUP 3
2000            0020 *
2000            0030 *  SECTION 3 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.08   MAY 30, 1977
2000            0070 *
2000            0080 **********
2000            0090 *
2000            0100 *  COMMON JUMP ADDRESS TABLE
2000            0110 *
2000            0120 COMJMP     EQU   $
2000            0130 *
2000            0140 CSR1       DS    3
2003            0150 PLAY       DS    3
2006            0160 BUFADR     DS    2
2008            0170 BUFEND     DS    2
200A            0180 PVTAB      DS    2
200C            0190 MATPAK     DS    3
200F            0200 MATERR     DS    3
2012            0210 RULES      DS    3
2015 C3 70 27   0220            JMP   SETDUR
2018 C3 1F 28   0230            JMP   RULES3
201B            0240 GENF0      DS    3
201E            0250 CLRBUF     DS    3
2021            0260 GENPRM     DS    3
2024            0270 MUL        DS    3
2027            0280 DIV        DS    3
202A            0290 DUMMY      DS    16
203A            0300 *
203A            0310 **********
203A            0320 *
203A            0330 *  COMRAM ORIGEN DEFINITION
203A            0340 *
203A            0350            ORG   COMJMP+1500H
3500            0360 COMRAM     EQU   $
3500            0370 *
3500            0380 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0390 *
3500            0400 MATPTR     DS    2
3502            0410 NEGEND     DS    2
3504            0420 MATRIX     EQU   $
3504            0430 MATLEN     EQU   95
3504            0440 PHCODE     DS    MATLEN
3563            0450 FEATA      DS    MATLEN
35C2            0460 FEATB      DS    MATLEN
3621            0470 STRES      DS    MATLEN
3680            0480 DUR        DS    MATLEN
36DF            0490 MATEND     EQU   $
36DF            0500 *
36DF            0510 *  RULES GROUP 3 (LOCAL) RAM WORKSPACE
36DF            0520 *
36DF            0530 PREVBD     DS    2
36E1            0540 NV         DS    1
36E2            0550 NEXT       DS    1
36E3            0560 *
36E3            0570 **********
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

36E3            0580 *
36E3            0590 *  CSR1 SECTION 3 CODE
36E3            0600 *
36E3            0610            ORG   COMJMP+770H
2770            0620 SECTAD     EQU   $
2770            0630 *
2770            0640 **********
2770            0650 *
2770            0660 *  PHONEME CODE DEFINITIONS FOR RULES
2770            0670 *
2770            0680 CSPACE     EQU   0
2770            0690 CTERM      EQU   4
2770            0700 CUW        EQU   14
2770            0710 CUX        EQU   18
2770            0720 CRX        EQU   24
2770            0730 CLX        EQU   25
2770            0740 CWX        EQU   26
2770            0750 CYX        EQU   27
2770            0760 CR         EQU   32
2770            0770 CW         EQU   34
2770            0780 CL         EQU   33
2770            0790 CM         EQU   36
2770            0800 CN         EQU   37
2770            0810 CNX        EQU   38
2770            0820 CP         EQU   39
2770            0830 CT         EQU   40
2770            0840 CK         EQU   41
2770            0850 CKX        EQU   42
2770            0860 CE         EQU   43
2770            0870 CD         EQU   44
2770            0880 CG         EQU   45
2770            0890 CGX        EQU   46
2770            0900 CDX        EQU   47
2770            0910 CS         EQU   50
2770            0920 CSH        EQU   51
2770            0930 CZ         EQU   54
2770            0940 CZH        EQU   55
2770            0950 CQ         EQU   59
2770            0960 *
2770            0970 *  DEFINE FEATURE LABELS
2770            0980 *
2770            0990 VOWEL      EQU   80H
2770            1000 CONS       EQU   40H
2770            1010 FRONT      EQU   20H
2770            1020 DIPHTH     EQU   10H
2770            1030 WDBND      EQU   4
2770            1040 PHBND      EQU   2
2770            1050 IGNORE     EQU   1
2770            1060 STOP       EQU   80H
2770            1070 VOICE      EQU   40H
2770            1080 PLOS       EQU   20H
2770            1090 PLOSA      EQU   10H
2770            1100 FRIC       EQU   8
2770            1110 LIQUID     EQU   4
2770            1120 NASAL      EQU   2
2770            1130 DENTAL     EQU   1
2770            1140 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2770            1150 **********
2770            1160 *
2770            1170 *  ALL DURATIONS ARE ASSIGNED AND MANIPULATED
2770            1180 *  IN UNITS OF 2.5 MSEC IN THIS SECTION.
2770            1190 *  FINALLY, AT THE END OF RULES3, THE DURATION
2770            1200 *  VALUES ARE DIVIDED BY 4 TO GIVE UNITS OF
2770            1210 *  10 MSEC, CORRESPONDING TO THE ACTUAL FRAME
2770            1220 *  COUNT FOR EACH PHON (EACH MATRIX COLUMN).
2770            1230 *
2770            1240 *  SET DURATION ROW OF MATRIX FROM TABLE
2770            1250 *
2770 21 05 35   1260 SETDUR      LXI   H,MATRIX+1
2773 22 00 35   1270             SHLD  MATPTR
2776 01 5F 00   1280             LXI   B,MATLEN
2779 5E         1290 SDLOOP      MOV   E,M       * GET CODE X INTO E
277A 09         1300             DAD   B
277B 09         1310             DAD   B
277C 09         1320             DAD   B
277D 7E         1330             MOV   A,M       * GET STRESS X INTO A
277E FE 06      1340             CPI   6
2780 FA 86 27   1350             JM    SD1
2783 3E 00      1360             MVI   A,0       * STRESS.GT.5, SET IT TO 0
2785 77         1370             MOV   M,A
2786 E5         1380 SD1         PUSH  H
2787 C6 FF      1390             ADI   255       * SET CARRY IF A.NE.0
2789 7B         1400             MOV   A,E
278A 17         1410             RAL
278B 5F         1420             MOV   E,A
278C 16 00      1430             MVI   D,0
278E 21 A7 27   1440             LXI   H,DURTAB
2791 19         1450             DAD   D         * COMPUTE LOC IN DURATION TABLE
2792 7E         1460             MOV   A,M       * GET DURATION (IN 2 MSEC UNITS)
2793 E1         1470             POP   H
2794 09         1480             DAD   B
2795 77         1490             MOV   M,A       * PUT DUR VALUE IN THE MATRIX
2796 2A 00 35   1500             LHLD  MATPTR
2799 23         1510             INX   H
279A 22 00 35   1520             SHLD  MATPTR    * INCREMENT POINTER
279D EB         1530             XCHG
279E 2A 02 35   1540             LHLD  NEGEND
27A1 19         1550             DAD   D
27A2 D8         1560             RC    *         * RETURN IF END OF MATRIX
27A3 EB         1570             XCHG
27A4 C3 79 27   1580             JMP   SDLOOP
27A7            1590 *
27A7            1600 *  TABLE OF DURATION VALUES (IN 2.5 MSEC UNITS)
27A7            1610 *  EACH PHON HAS A STRESSED DURATION VALUE
27A7            1620 *  AND AN UNSTRESSED DURATION VALUE.
27A7            1630 *
27A7 00         1640 DURTAB      DB    0    * SPACE (WDBND
27A8 00         1650             DB    0
27A9 00         1660             DB    0    * PERIOD
27AA 00         1670             DB    0
27AB 48         1680             DB    72   * COMMA (PAUSE)
27AC 48         1690             DB    72
27AD 00         1700             DB    0    * QUEST
27AE 00         1710             DB    0
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

27AF 00         1720            DB    0    * TERM
27B0 00         1730            DB    0
27B1 22         1740            DB    34   * IY, UNSTRESSED
27B2 2B         1750            DB    43   * IY, STRESSED
27B3 1D         1760            DB    29   * IH, UNSTRESSED
27B4 22         1770            DB    34   * IH,   ETC.
27B5 22         1780            DB    34   * EH
27B6 2B         1790            DB    43
27B7 1F         1800            DB    31   * AE
27B8 3A         1810            DB    58
27B9 2B         1820            DB    43   * AA
27BA 3C         1830            DB    60
27BB 18         1840            DB    24   * AH
27BC 2A         1850            DB    42
27BD 30         1860            DB    48   * AO
27BE 40         1870            DB    64
27BF 3A         1880            DB    58   * OW
27C0 3A         1890            DB    58
27C1 29         1900            DB    41   * UH
27C2 2E         1910            DB    46
27C3 24         1920            DB    36   * UW
27C4 38         1930            DB    56
27C5 16         1940            DB    22   * AX
27C6 18         1950            DB    24
27C7 16         1960            DB    22   * IX
27C8 18         1970            DB    24
27C9 2B         1980            DB    43   * ER
27CA 38         1990            DB    56
27CB 24         2000            DB    36   * UX
27CC 38         2010            DB    56
27CD 26         2020            DB    38   * OH
27CE 38         2030            DB    56
27CF 30         2040            DB    48   * AW
27D0 3A         2050            DB    58
27D1 30         2060            DB    48   * AY
27D2 3A         2070            DB    58
27D3 30         2080            DB    48   * OY
27D4 3A         2090            DB    58
27D5 35         2100            DB    53   * EY
27D6 37         2110            DB    55
27D7 26         2120            DB    38   * RX
27D8 30         2130            DB    48
27D9 26         2140            DB    38   * LX
27DA 2B         2150            DB    43
27DB 20         2160            DB    32   * WX
27DC 20         2170            DB    32
27DD 1D         2180            DB    29   * YX
27DE 20         2190            DB    32
27DF 22         2200            DB    34   * WH
27E0 2A         2210            DB    42
27E1 00         2220            DB    0    * EL
27E2 00         2230            DB    0
27E3 00         2240            DB    0    * EM
27E4 00         2250            DB    0
27E5 00         2260            DB    0    * EN
27E6 00         2270            DB    0
27E7 1C         2280            DB    28   * R
```

```
ADDR B1 B2 B3 E LINE  LABEL      OPCD  OPERAND

27E8 28          2290             DB    40
27E9 18          2300             DB    24    * L
27EA 22          2310             DB    34
27EB 20          2320             DB    32    * W
27EC 20          2330             DB    32
27ED 18          2340             DB    24    * Y
27EE 20          2350             DB    32
27EF 1C          2360             DB    28    * M
27F0 1E          2370             DB    30
27F1 1C          2380             DB    28    * N
27F2 1E          2390             DB    30
27F3 1C          2400             DB    28    * NX
27F4 1E          2410             DB    30
27F5 20          2420             DB    32    * P
27F6 20          2430             DB    32
27F7 10          2440             DB    16    * T
27F8 18          2450             DB    24
27F9 18          2460             DB    24    * K
27FA 18          2470             DB    24
27FB 18          2480             DB    24    * KX
27FC 18          2490             DB    24
27FD 18          2500             DB    24    * B
27FE 1A          2510             DB    26
27FF 12          2520             DB    18    * D
2800 14          2530             DB    20
2801 18          2540             DB    24    * G
2802 18          2550             DB    24
2803 18          2560             DB    24    * GX
2804 18          2570             DB    24
2805 0A          2580             DB    10    * DX
2806 0C          2590             DB    12
2807 2A          2600             DB    42    * F
2808 2C          2610             DB    44
2809 26          2620             DB    38    * TH
280A 28          2630             DB    40
280B 30          2640             DB    48    * S
280C 30          2650             DB    48
280D 2C          2660             DB    44    * SH
280E 2C          2670             DB    44
280F 1A          2680             DB    26    * V
2810 1E          2690             DB    30
2811 18          2700             DB    24    * DH
2812 18          2710             DB    24
2813 18          2720             DB    24    * Z
2814 18          2730             DB    24
2815 18          2740             DB    24    * ZH
2816 18          2750             DB    24
2817 00          2760             DB    0     * CH
2818 00          2770             DB    0
2819 00          2780             DB    0     * JH
281A 00          2790             DB    0
281B 18          2800             DB    24    * HH
281C 1C          2810             DB    28
281D 0A          2820             DB    10    * Q
281E 0A          2830             DB    10
281F             2840 *
281F             2850 **********
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

281F            2860 *
281F            2870 *  RULES GROUP 3
281F            2880 *
281F            2890 **********
281F            2900 *
281F 00         2910 RULES3     NOP
2820 AF         2920            XRA   A
2821 32 E1 36   2930            STA   NV        * CLEAR VOWEL COUNTER (RULE 3A)
2824 32 E2 36   2940            STA   NEXT      * CLEAR NEXT (RULE 3B)
2827 01 5F 00   2950            LXI   B,MATLEN
282A 21 05 35   2960            LXI   H,MATRIX+1
282D 22 00 35   2970            SHLD  MATPTR
2830            2980 R3LOOP     EQU   $
2830            2990 *
2830            3000 *
2830            3010 *  IN THE SPAN BETWEEN TWO WORD OR PHRASE BOUNDARIES
2830            3020 *  CHANGE THE DURATION OF EACH STRESSED VOWEL
2830            3030 *  BY (NV+1)/(2*NV), WHERE NV IS THE NUMBER OF
2830            3040 *  VOWELS IN THE SPAN.
2830            3050 *
2830 09         3060 R3A        DAD   B
2831 01 E1 36   3070            LXI   B,NV
2834 7E         3080            MOV   A,M       * GET COL X FEATA
2835 E6 06      3090            ANI   WDBND+PHBND
2837 C2 46 28   3100            JNZ   R3A2      * EITHER BOUND, COMPUTE THE SPAN
283A 7E         3110            MOV   A,M       * NOT A BOUNDARY, IS IT A VOWEL?
283B E6 80      3120            ANI   VOWEL
283D CA AB 28   3130            JZ    R3B       * NO
2840 0A         3140            LDAX  B         * YES, INCREMENT NV
2841 3C         3150            INR   A
2842 02         3160            STAX  B
2843 C3 AB 28   3170            JMP   R3B
2846 0A         3180 R3A2       LDAX  B         * GET NV
2847 FE 02      3190            CPI   2
2849 DA A1 28   3200            JC    R3A5      * 0 OR 1 VOWEL, NO DUR CHANGE
284C 3C         3210            INR   A
284D 6F         3220            MOV   L,A       * PUT (N+1)*64 IN HL
284E 26 00      3230            MVI   H,0
2850 29         3240            DAD   H
2851 29         3250            DAD   H
2852 29         3260            DAD   H
2853 29         3270            DAD   H
2854 29         3280            DAD   H
2855 29         3290            DAD   H
2856 EB         3300            XCHG  *         * THEN INTO DE
2857 0A         3310            LDAX  B         * GET NV AGAIN
2858 6F         3320            MOV   L,A
2859 26 00      3330            MVI   H,0
285B 23         3340            INX   H
285C CD 27 20   3350            CALL  DIV       * RESULT IS DUR RATIO IN 64THS
285F 7D         3360            MOV   A,L
2860 02         3370            STAX  B         * SAVE IT IN NV
2861 2A DF 36   3380            LHLD  PREVBD
2864 23         3390            INX   H
2865 7C         3400            MOV   A,H       * -(PREVBD)-2 INTO DE
2866 2F         3410            CMA
2867 57         3420            MOV   D,A
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2868 7D         3430            MOV   A,L
2869 2F         3440            CMA
286A 5F         3450            MOV   E,A
286B 2A 00 35   3460            LHLD  MATPTR
286E 19         3470            DAD   D       * (MATPTR)-(PREVBD)-2 INTO HL
286F 5D         3480            MOV   E,L     * E IS COL COUNTER FOR THE SPAN
2870 01 5F 00   3490 R3A3       LXI   B,MATLEN
2873 2A DF 36   3500            LHLD  PREVBD
2876 23         3510            INX   H       * SCAN THE SPAN
2877 22 DF 36   3520            SHLD  PREVBD
287A 09         3530            DAD   B
287B 7E         3540            MOV   A,M     * GET FEATA
287C E6 80      3550            ANI   VOWEL
287E CA 9D 28   3560            JZ    R3A4    * NOT A VOWEL
2881 09         3570            DAD   B
2882 09         3580            DAD   B
2883 7E         3590            MOV   A,M     * GET THE STRESS VALUE
2884 B7         3600            ORA   A
2885 CA 9D 28   3610            JZ    R3A4    * VOWEL NOT STRESSED
2888 09         3620            DAD   B
2889 3A E1 36   3630            LDA   NV
288C 47         3640            MOV   B,A
288D D5         3650.           PUSH  D
288E 5E         3660            MOV   E,M     * PUT OLD DURATION IN DE
288F CD 24 20   3670            CALL  MUL     * NEW DUR *64 IN HL
2892 EB         3680            XCHG
2893 29         3690            DAD   H
2894 29         3700            DAD   H
2895 EB         3710            XCHG
2896 7B         3720            MOV   A,E     * NEW DUR INTO A
2897 17         3730            RAL
2898 7A         3740            MOV   A,D
2899 CE 00      3750            ACI   0
289B 77         3760            MOV   M,A
289C D1         3770            POP   D
289D 1D         3780 R3A4       DCR   E
289E C2 70 28   3790            JNZ   R3A3
28A1 AF         3800 R3A5       XRA   A
28A2 32 E1 36   3810            STA   NV      * CLEAR THE VOWEL COUNTER
28A5 2A 00 35   3820            LHLD  MATPTR
28A8 22 DF 36   3830            SHLD  PREVBD  * UPDATE PREVBD
28AB            3840 *
28AB            3850 *  DUR*1.38 <= /LAST VOWEL OF A PHRASE, .../PHBND
28AB            3860 *
28AB 2A 00 35   3870 R3B        LHLD  MATPTR
28AE 01 5F 00   3880            LXI   B,MATLEN * RESTORE BC
28B1 09         3890            DAD   B
28B2 7E         3900            MOV   A,M     * GET COL X FEATA
28B3 E6 02      3910            ANI   PHBND
28B5 CA D8 28   3920            JZ    R3C
28B8 2B         3930 R3B2       DCX   H
28B9 7E         3940            MOV   A,M     * GET PREV COL FEATA
28BA E6 02      3950            ANI   PHBND
28BC C2 D8 28   3960            JNZ   R3C     * STOP SCAN IF PHBOUND
28BF E5         3970            PUSH  H
28C0 09         3980            DAD   B
28C1 09         3990            DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

28C2 09          4000           DAD   B
28C3 46          4010           MOV   B,M      * GET PRESENT DUR INTO B
28C4 1E 58       4020           MVI   E,58H    * 1.38 (SHIFTED) INTO E
28C6 CD 24 20    4030           CALL  MUL      * (B)*(E) INTO DE
28C9 EB          4040           XCHG
28CA 29          4050           DAD   H        * SHIFT BIN PT TO BYTE BOUNDARY
28CB 29          4060           DAD   H
28CC EB          4070           XCHG
28CD 72          4080           MOV   M,D      * UPDATE DURATION
28CE E1          4090           POP   H
28CF 01 5F 00    4100           LXI   B,MATLEN * RESTORE B,C,H,L
28D2 7E          4110           MOV   A,M      * GET FEATA AGAIN
28D3 E6 80       4120           ANI   VOWEL
28D5 CA B8 28    4130           JZ    R3B2     * NOT A VOWEL, CONTINUE
28D8             4140 *
28D8             4150 *  DUR*0.6 <= /VOWEL/PLOS -VOICE
28D8             4160 *
28D8 00          4170 R3C       NOP
28D9 2A 00 35    4180           LHLD  MATPTR
28DC 09          4190           DAD   B
28DD 7E          4200           MOV   A,M      * GET COL X FEATA
28DE E6 80       4210           ANI   VOWEL
28E0 CA 39 29    4220           JZ    R3F      * NOT A VOWEL
28E3 2A 00 35    4230           LHLD  MATPTR
28E6 23          4240           INX   H
28E7 7E          4250           MOV   A,M      * GET COL X+1 CODE
28E8 FE 04       4260           CPI   CTERM
28EA CA DD 2A    4270           JZ    ENDR3
28ED 09          4280           DAD   B
28EE 09          4290           DAD   B
28EF 7E          4300           MOV   A,M      * GET COL X+1 FEATB
28F0 EE 20       4310           XRI   PLOS
28F2 E6 60       4320           ANI   PLOS+VOICE
28F4 C2 FC 28    4330           JNZ   R3D      * NOT AN UNVOICED PLOSIVE
28F7 1E 26       4340           MVI   E,26H    * OK, MULTIPLY BY 0.6 (SHIFTED)
28F9 C3 27 29    4350           JMP   R3CDE
28FC             4360 *
28FC             4370 *  DUR*1.25 <= /VOWEL/FRIC VOICE
28FC             4380 *
28FC 7E          4390 R3D       MOV   A,M      * GET COL X+1 FEATB AGAIN
28FD 2F          4400           CMA
28FE E6 48       4410           ANI   FRIC+VOICE
2900 C2 08 29    4420           JNZ   R3E      * NOT A VOICED FRICATIVE
2903 1E 50       4430           MVI   E,50H    * SET MULTIPLIER TO 1.25 (*64)
2905 C3 27 29    4440           JMP   R3CDE
2908             4450 *
2908             4460 *  DUR*0.5 <= /VOWEL/RX OR LX,CONS
2908             4470 *
2908 2A 00 35    4480 R3E       LHLD  MATPTR
290B 23          4490           INX   H
290C 7E          4500           MOV   A,M      * GET COL X+1 CODE
290D FE 18       4510           CPI   CRX
290F CA 17 29    4520           JZ    R3E2
2912 FE 19       4530           CPI   CLX
2914 C2 DD 2A    4540           JNZ   ENDR3    * NO GOOD, END GROUP 3
2917 23          4550 R3E2      INX   H
2918 7E          4560           MOV   A,M      * GET COL X+2 CODE
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2919 FE 04      4570            CPI   CTERM
291B CA DD 2A   4580            JZ    ENDR3    * X+1 IS LAST COL
291E 09         4590            DAD   B
291F 7E         4600            MOV   A,M      * GET COL X+2 FEATA
2920 E6 40      4610            ANI   CONS
2922 CA DD 2A   4620            JZ    ENDR3    * NOT A CONSONANT
2925 1E 20      4630            MVI   E,20H    * SET MULTIPLIER TO 0.5 (*64)
2927 2A 00 35   4640 R3CDE     LHLD  MATPTR
292A 09         4650            DAD   B
292B 09         4660            DAD   B
292C 09         4670            DAD   B
292D 46         4680            MOV   B,M      * GET COL X DURATION
292E CD 24 20   4690            CALL  MUL      * MULTIPLY IT BY (E)
2931 EB         4700            XCHG
2932 29         4710            DAD   H
2933 29         4720            DAD   H        * MOVE THE BIN POINT
2934 EB         4730            XCHG
2935 72         4740            MOV   M,D      * & SET NEW DURATION
2936 C3 DD 2A   4750            JMP   ENDR3    * GOTO END OF GROUP
2939            4760 *
2939            4770 *  DUR=90 MSEC <= S/(W OR R OR L) STRESS.LT.0//VOWEEL
2939            4780 *
2939 2A 00 35   4790 R3F        LHLD  MATPTR
293C 7E         4800            MOV   A,M      * GET COL X CODE
293D FE 22      4810            CPI   CW
293F CA 4C 29   4820            JZ    R3F2
2942 FE 20      4830            CPI   CR
2944 CA 4C 29   4840            JZ    R3F2
2947 FE 21      4850            CPI   CL
2949 C2 8D 29   4860            JNZ   R3H      * NOT W OR R OR L
294C 09         4870 R3F2       DAD   B
294D 09         4880            DAD   B
294E 09         4890            DAD   B
294F 7E         4900            MOV   A,M      * GET COL X STRESS
2950 B7         4910            ORA   A
2951 F2 AD 29   4920            JP    R3I      * STRESS.GE.0
2954 2A 00 35   4930            LHLD  MATPTR
2957 23         4940            INX   H
2958 7E         4950            MOV   A,M      * GET COL X+1 CODE
2959 FE 04      4960            CPI   CTERM
295B CA 8D 29   4970            JZ    R3H      * W,R, OR L IS IN LAST COL
295E 09         4980            DAD   B
295F 7E         4990            MOV   A,M      * GET COL X+1 FEATA
2960 E6 80      5000            ANI   VOWEL
2962 CA 8D 29   5010            JZ    R3H      * NOT A VOWEL
2965 2A 00 35   5020            LHLD  MATPTR
2968 2B         5030            DCX   H
2969 7E         5040            MOV   A,M      * GET COL X-1 CODE
296A FE 04      5050            CPI   CTERM
296C CA 8D 29   5060            JZ    R3H      * W, R, OR L IS IN 1ST COL
296F FE 32      5070            CPI   CS
2971 09         5080            DAD   B
2972 09         5090            DAD   B        * MOVE HL TO COL X-1 FEATB
2973 C2 7E 29   5100            JNZ   R3G      * NO S ON COL X-1
2976 23         5110            INX   H
2977 09         5120            DAD   B
2978 09         5130            DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2979 36 24      5140            MVI   M,36     * SET DURATION TO 90 MSEC
297B C3 8D 29   5150            JMP   R3H
297E            5160 *
297E            5170 *  DUR+20 MSEC <= PLOS -VOICE/(W OR R OR L)
297E            5180 *                    STRESS.LT.0/VOWEL
297E            5190 *
297E 7E         5200 R3G        MOV   A,M      * GET COL X-1 FEATB
297F EE 20      5210            XRI   PLOS
2981 E6 60      5220            ANI   PLOS+VOICE
2983 C2 8D 29   5230            JNZ   R3H      * NOT AN UNVOICED PLOSIVE
2986 23         5240            INX   H
2987 09         5250            DAD   B
2988 09         5260            DAD   B
2989 7E         5270            MOV   A,M      * GET COL X STRESS
298A C6 08      5280            ADI   8        * ADD 20 MSEC
298C 77         5290            MOV   M,A
298D            5300 *
298D            5310 *  DUR*0.8 <= /CONS STRESS.LT.0/
298D            5320 *
298D 2A 00 35   5330 R3H        LHLD  MATPTR
2990 09         5340            DAD   B
2991 09         5350            DAD   B
2992 7E         5360            MOV   A,M      * GET COL X FEATB
2993 E6 40      5370            ANI   CONS
2995 CA DD 2A   5380            JZ    ENDR3    * NOT A CONS, END OF GROUP
2998 09         5390            DAD   B
2999 7E         5400            MOV   A,M      * GET COL X STRESS
299A B7         5410            ORA   A
299B F2 AD 29   5420            JP    R3I      * STRESS IS NOT NEGATIVE
299E 09         5430            DAD   B
299F 46         5440            MOV   B,M      * GET COL X DURATION
29A0 1E 33      5450            MVI   E,33H    * MULTIPLY BY 0.8
29A2 CD 24 20   5460            CALL  MUL
29A5 EB         5470            XCHG
29A6 29         5480            DAD   H        * SHIFT IT OVER
29A7 29         5490            DAD   H
29A8 EB         5500            XCHG
29A9 72         5510            MOV   M,D      * SET NEW DUR
29AA 01 5F 00   5520            LXI   B,MATLEN
29AD            5530 *
29AD            5540 *  DUR=70,60 MSEC <= /T STRESS.GE.0,SH STRESS.GE.0/
29AD            5550 *
29AD 2A 00 35   5560 R3I        LHLD  MATPTR
29B0 54         5570            MOV   D,H
29B1 5D         5580            MOV   E,L
29B2 13         5590            INX   D
29B3 7E         5600            MOV   A,M      * GET COL X CODE
29B4 FE 28      5610            CPI   CT
29B6 C2 DD 29   5620            JNZ   R3J      * NOT T, TRY RULE 3J
29B9 1A         5630            LDAX  D        * GET COL X+1 CODE
29BA FE 04      5640            CPI   CTERM
29BC CA 78 2A   5650            JZ    R3N      * T IS IN LAST COL
29BF FE 33      5660            CPI   CSH
29C1 C2 78 2A   5670            JNZ   R3N      * T BUT NO SH, TRY RULE 3N
29C4 09         5680            DAD   B
29C5 09         5690            DAD   B
29C6 09         5700            DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

29C7 7E         5710            MOV   A,M      * GET COL X STRESS
29C8 FE 01      5720            CPI   1
29CA FA 06 2A   5730            JM    R3K      * T UNSTRESSED, TRY RULE 3K
29CD 23         5740            INX   H
29CE 7E         5750            MOV   A,M      * GET COL X+1 STRESS
29CF FE 01      5760            CPI   1
29D1 FA 78 2A   5770            JM    R3N      * SH UNSTRESSED, TRY RULE 3N
29D4 09         5780            DAD   B
29D5 36 18      5790            MVI   M,24     * SET COL X+1 DUR TO 60 MSEC
29D7 2B         5800            DCX   H
29D8 36 1C      5810            MVI   M,28     * SEL COL X DUR TO 70 MSEC
29DA C3 78 2A   5820            JMP   R3N
29DD            5830 *
29DD            5840 *  DUR=70,50 MSEC <= /D,STRESS.GT.0,ZH STRESS.GT.0/
29DD            5850 *
29DD FE 2C      5860 R3J        CPI   CD
29DF C2 26 2A   5870            JNZ   R3M      * NOT T OR D, TRY RULE 3M
29E2 1A         5880            LDAX  D
29E3 FE 04      5890            CPI   CTERM
29E5 CA 78 2A   5900            JZ    R3N      * D IS IN LAST COL
29E8 FE 37      5910            CPI   CZH
29EA C2 78 2A   5920            JNZ   R3N      * D BUT NO ZH, TRY RULE 3N
29ED 09         5930            DAD   B
29EE 09         5940            DAD   B
29EF 09         5950            DAD   B
29F0 7E         5960            MOV   A,M      * GET  COL X STRESS
29F1 FE 01      5970            CPI   1
29F3 FA 16 2A   5980            JM    R3L      * D UNSTRESSED, TRY RULE R3L
29F6 23         5990            INX   H
29F7 7E         6000            MOV   A,M      * GET COL X+1 STRESS
29F8 FE 01      6010            CPI   1
29FA FA 78 2A   6020            JM    R3N      * ZH UNSTRESSED, TRY RULE 3N
29FD 09         6030            DAD   B
29FE 36 14      6040            MVI   M,20     * SET X+1 DUR TO 50 MSEC
2A00 2B         6050            DCX   H
2A01 36 1C      6060            MVI   M,28     * SET X DUR TO 70 MSEC
2A03 C3 78 2A   6070            JMP   R3N
2A06            6080 *
2A06            6090 *  DUR=60,40 MSEC <= /T STRESS.EQ.0,SH STRESS.EQ.0/
2A06            6100 *
2A06 23         6110 R3K        INX   H
2A07 7E         6120            MOV   A,M      * GET COL X+1 STRESS
2A08 FE 01      6130            CPI   1
2A0A F2 78 2A   6140            JP    R3N      * SH IS STRESSED, TRY RULE 3N
2A0D 09         6150            DAD   B
2A0E 36 10      6160            MVI   M,16     * SET COL X+1 DUR TO 40 MSEC
2A10 2B         6170            DCX   H
2A11 36 18      6180            MVI   M,24     * SET COL X DUR TO 60 MSEC
2A13 C3 78 2A   6190            JMP   R3N
2A16            6200 *
2A16            6210 *  DUR=40,30 MSEC <= /D STRESS.EQ.0,ZH STRESS.EQ.0/
2A16            6220 *
2A16 23         6230 R3L        INX   H
2A17 7E         6240            MOV   A,M      * GET COL X+1 STRESS
2A18 FE 01      6250            CPI   1
2A1A FA 78 2A   6260            JM    R3N
2A1D 09         6270            DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2A1E 36 0C      6280            MVI   M,12     * SET COL X+1 DUR TO 30 MSEC
2A20 2B         6290            DCX   H
2A21 36 10      6300            MVI   M,16     * SET COL X DUR TO 40 MSEC
2A23 C3 78 2A   6310            JMP   R3N
2A26            6320 *
2A26            6330 *  DUR=30,30 MSEC <= /(M,P OR B) OR (N,T OR D) OR
2A26            6340 *                    (NX,K OR KX OR G OR GX)/
2A26            6350 *
2A26 FE 24      6360 R3M        CPI   CM       * TEST COL X CODE
2A28 CA 38 2A   6370            JZ    R3M2     * IT'S M
2A2B FE 25      6380            CPI   CN
2A2D CA 46 2A   6390            JZ    R3M3     * IT'S N
2A30 FE 26      6400            CPI   CNX
2A32 CA 54 2A   6410            JZ    R3M4     * IT'S NX
2A35 C3 78 2A   6420            JMP   R3N
2A38 1A         6430 R3M2       LDAX  D
2A39 FE 27      6440            CPI   CP
2A3B CA 6C 2A   6450            JZ    R3M5     * IT'S M,P
2A3E FE 2B      6460            CPI   CB
2A40 CA 6C 2A   6470            JZ    R3M5     * IT'S M,B
2A43 C3 DD 2A   6480            JMP   ENDR3    * IT'S SOMETHING ELSE
2A46 1A         6490 R3M3       LDAX  D
2A47 FE 28      6500            CPI   CT
2A49 CA 6C 2A   6510            JZ    R3M5     * IT'S N,T
2A4C FE 2C      6520            CPI   CD
2A4E CA 6C 2A   6530            JZ    R3M5     * IT'S N,D
2A51 C3 DD 2A   6540            JMP   ENDR3    * IT'S SOMETHING ELSE
2A54 1A         6550 R3M4       LDAX  D
2A55 FE 29      6560            CPI   CK
2A57 CA 6C 2A   6570            JZ    R3M5     * IT'S NX,K
2A5A FE 2A      6580            CPI   CKX
2A5C CA 6C 2A   6590            JZ    R3M5     * IT'S NX,KX
2A5F FE 2D      6600            CPI   CG
2A61 CA 6C 2A   6610            JZ    R3M5     * IT'S NX,G
2A64 FE 2E      6620            CPI   CGX
2A66 CA 6C 2A   6630            JZ    R3M5     * IT'S NX,GX
2A69 C3 DD 2A   6640            JMP   ENDR3    * IT'S SOMETHING ELSE
2A6C 09         6650 R3M5       DAD   B
2A6D 09         6660            DAD   B
2A6E 09         6670            DAD   B
2A6F 09         6680            DAD   B
2A70 36 0C      6690            MVI   M,12     * SET COL X DUR TO 30 MSEC
2A72 23         6700            INX   H
2A73 36 0C      6710            MVI   M,12     * SET COL X+1 DUR TO 30 MSEC
2A75 C3 DD 2A   6720            JMP   ENDR3
2A78            6730 *
2A78            6740 *  DUR*0.5 <= PLOS,OPTIONAL WDBOUND/PLOS/
2A78            6750 *
2A78 2A 00 35   6760 R3N        LHLD  MATPTR
2A7B 09         6770            DAD   B
2A7C 09         6780            DAD   B
2A7D 7E         6790            MOV   A,M      * GET COL X FEATB
2A7E E6 20      6800            ANI   PLOS
2A80 CA DD 2A   6810            JZ    ENDR3    * NOT A PLOSIVE, END OF GROUP
2A83 2A 00 35   6820            LHLD  MATPTR
2A86 2B         6830            DCX   H
2A87 7E         6840            MOV   A,M      * GET COL X-1 CODE
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2A88 FE 04      6850            CPI   CTERM
2A8A CA A6 2A   6860            JZ    R3N3     * X IS 1ST COL, TRY PLOS AFTER
2A8D 54         6870            MOV   D,H
2A8E 5D         6880            MOV   E,L
2A8F 09         6890            DAD   B
2A90 7E         6900            MOV   A,M      * GET COL X-1 FEATA
2A91 E6 04      6910            ANI   WDBND
2A93 CA 9F 2A   6920            JZ    R3N2     * NOT A WDBND, TEST FOR PLOSIVE
2A96 EB         6930            XCHG
2A97 2B         6940            DCX   H
2A98 7E         6950            MOV   A,M      * GET COL X-2 CODE
2A99 FE 04      6960            CPI   CTERM
2A9B CA A6 2A   6970            JZ    R3N3
2A9E 09         6980            DAD   B
2A9F 09         6990 R3N2       DAD   B        * MOVE TO FEATB
2AA0 7E         7000            MOV   A,M
2AA1 E6 20      7010            ANI   PLOS
2AA3 C2 C9 2A   7020            JNZ   R3N5     * OK, DO RULE 3N
2AA6            7030 *
2AA6            7040 *  DUR*0.5 <= /PLOS/OPTIONAL WDBOUND,PLOS
2AA6            7050 *
2AA6 2A 00 35   7060 R3N3       LHLD  MATPTR
2AA9 23         7070            INX   H
2AAA 7E         7080            MOV   A,M      * GET COL X+1 CODE
2AAB FE 04      7090            CPI   CTERM
2AAD CA DD 2A   7100            JZ    ENDR3    * X IS LAST COL, END GROUP
2AB0 54         7110            MOV   D,H
2AB1 5D         7120            MOV   E,L
2AB2 09         7130            DAD   B
2AB3 7E         7140            MOV   A,M      * GET COL X+1 FEATA
2AB4 E6 04      7150            ANI   WDBND
2AB6 CA C2 2A   7160            JZ    R3N4     * NOT A WDBND, TEST FOR PLOS
2AB9 EB         7170            XCHG
2ABA 23         7180            INX   H
2ABB 7E         7190            MOV   A,M      * GET COL X+2 CODE
2ABC FE 04      7200            CPI   CTERM
2ABE CA DD 2A   7210            JZ    ENDR3
2AC1 09         7220            DAD   B
2AC2 09         7230 R3N4       DAD   B        * MOVE TO FEATB
2AC3 7E         7240            MOV   A,M
2AC4 E6 20      7250            ANI   PLOS
2AC6 CA DD 2A   7260            JZ    ENDR3    * NO, END GROUP 3
2AC9 2A 00 35   7270 R3N5       LHLD  MATPTR
2ACC 09         7280            DAD   B
2ACD 09         7290            DAD   B
2ACE 09         7300            DAD   B
2ACF 09         7310            DAD   B
2AD0 C5         7320            PUSH  B
2AD1 46         7330            MOV   B,M      * GET COL X DURATION
2AD2 1E 20      7340            MVI   E,20H    * MULTIPLY BY 0.5
2AD4 CD 24 20   7350            CALL  MUL
2AD7 EB         7360            XCHG
2AD8 29         7370            DAD   H
2AD9 29         7380            DAD   H
2ADA EB         7390            XCHG
2ADB 72         7400            MOV   M,D      * STORE NEW DURATION
2ADC C1         7410            POP   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2ADD            7420 *
2ADD            7430 *  END OF RULE GROUP 3
2ADD            7440 *
2ADD 2A 00 35   7450 ENDR3      LHLD  MATPTR
2AE0 23         7460            INX   H
2AE1 22 00 35   7470            SHLD  MATPTR
2AE4 EB         7480            XCHG
2AE5 2A 02 35   7490            LHLD  NEGEND
2AE8 19         7500            DAD   D
2AE9 EB         7510            XCHG
2AEA D2 30 28   7520            JNC   R3LOOP  * LOOP AGAIN IF NOT DONE
2AED            7530 *
2AED            7540 *  READJUST DURATIONS FROM 2.5 MSEC TO 10 MSEC
2AED            7550 *    (SO DUR UNIT = FRAME TIME)
2AED            7560 *
2AED 11 FB CA   7570            LXI   D,-MATRIX-1
2AF0 19         7580            DAD   D
2AF1 4D         7590            MOV   C,L
2AF2 21 80 36   7600            LXI   H,DUR
2AF5 23         7610 ADJDUR     INX   H
2AF6 7E         7620            MOV   A,M
2AF7 B7         7630            ORA   A       * CLEAR CARRY, DIV BY SHIFTING
2AF8 1F         7640            RAR
2AF9 B7         7650            ORA   A
2AFA 1F         7660            RAR
2AFB CE 00      7670            ACI   0       * ROUND UP 1 IF CARRY SET
2AFD 77         7680            MOV   M,A
2AFE 0D         7690            DCR   C
2AFF C2 F5 2A   7700            JNZ   ADJDUR
2B02 C9         7710            RET
2B03            7720 *
2B03            7730 *  END OF SECT3
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  GENF0
2000            0020 *
2000            0030 *  SECTION 4 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.11   MAAY 30, 1977
2000            0070 *
2000            0080 **********
2000            0090 *
2000            0100 *  COMMON JUMP ADDRESS TABLE
2000            0110 *
2000            0120 COMJMP     EQU   $
2000            0130 *
2000            0140 CSR1       DS    3
2003            0150 PLAY       DS    3
2006            0160 BUFADR     DS    2
2008            0170 BUFEND     DS    2
200A            0180 PVTAB      DS    2
200C            0190 MATPAK     DS    3
200F            0200 MATERR     DS    3
2012            0210 RULES      DS    3
2015            0220 SETDUR     DS    3
2018            0230 RULES3     DS    3
201B C3 10 2B   0240            JMP   GENF0
201E C3 D9 2D   0250            JMP   CLRBUF
2021            0260 GENPRM     DS    3
2024            0270 MUL        DS    3
2027            0280 DIV        DS    3
202A            0290 DUMMY      DS    16
203A            0300 *
203A            0310 **********
203A            0320 *
203A            0330 *  COMRAM ORIGEN DEFINITION
203A            0340 *
203A            0350            ORG   COMJMP+1500H
3500            0360 COMRAM     EQU   $
3500            0370 *
3500            0380 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0390 *
3500            0400 MATPTR     DS    2
3502            0410 NEGEND     DS    2
3504            0420 MATRIX     EQU   $
3504            0430 MATLEN     EQU   95
3504            0440 PHCODE     DS    MATLEN
3563            0450 FEATA      DS    MATLEN
35C2            0460 FEATB      DS    MATLEN
3621            0470 STRES.     DS    MATLEN
3680            0480 DUR        DS    MATLEN
36DF            0490 MATEND     EQU   $
36DF            0500 BUFPTR     DS    2
36E1            0510 *
36E1            0520 *  GENF0 (LOCAL) RAM WORKSPACE
36E1            0530 *
36E1            0540 STRSX      DS    1
36E2            0550 DURX       DS    1
36E3            0560 DURX4      DS    1
36E4            0570 HF0        DS    2
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

36E6            0580 FTERM     DS    1
36E7            0590 DELTA     DS    2
36E9            0600 SD        DS    2
36EB            0610 NEGBUF    DS    2
36ED            0620 NEGPSE    DS    2
36EF            0630 NEGBMD    DS    2
36F1            0640 *
36F1            0650 **********  .
36F1            0660 *
36F1            0670 *  GENF0
36F1            0680 *
36F1            0690           ORG   COMJMP+0B10H
2B10            0700 SECTAD    EQU   $
2B10            0710 *
2B10            0720 **********
2B10            0730 *
2B10            0740 *  PHONEME CODE DEFINITIONS FOR GENF0
2B10            0750 *
2B10            0760 CPER      EQU   1
2B10            0770 CQUEST    EQU   3
2B10            0780 CTERM     EQU   4
2B10            0790 *
2B10            0800 *  FEATURE LABEL DEFS FOR GENF0
2B10            0810 *
2B10            0820 VOWEL     EQU   80H
2B10            0830 WDBND     EQU   4
2B10            0840 VOICE     EQU   40H
2B10            0850 *
2B10            0860 **********
2B10            0870 *
2B10 2A 06 20   0880 GENF0     LHLD  BUFADR  * GET (BUFADR) IN DE
2B13 EB         0890           XCHG
2B14 21 0B 00   0900           LXI   H,11
2B17 19         0910           DAD   D
2B18 0E 09      0920           MVI   C,9
2B1A 2B         0930           DCX   H
2B1B 36 00      0940           MVI   M,0     * CLEAR FRAME 1
2B1D 0D         0950           DCR   C
2B1E C2 1A 2B   0960           JNZ   $-4
2B21 23         0970           INX   H
2B22 22 DF 36   0980           SHLD  BUFPTR  * POINT TO F0 BYTE IN FRAME 1
2B25 7A         0990           MOV   A,D
2B26 2F         1000           CMA
2B27 67         1010           MOV   H,A
2B28 47         1020           MOV   B,A
2B29 7B         1030           MOV   A,E
2B2A 2F         1040           CMA
2B2B 6F         1050           MOV   L,A
2B2C 4F         1060           MOV   C,A
2B2D 2B         1070           DCX   H
2B2E 2B         1080           DCX   H
2B2F 22 EB 36   1090           SHLD  NEGBUF  * NEGBUF=-(BUFADR)-3
2B32 21 5E FF   1100           LXI   H,-9*18
2B35 09         1110           DAD   B       * NEGPSE = -ADDR OF F0 PARAM
2B36 22 ED 36   1120           SHLD  NEGPSE  *  IN LAST FRAME OF INITIAL PAUSE
2B39 2A 08 20   1130           LHLD  BUFEND  * GET LAST AVAILABLE BUFFER LOC
2B3C 09         1140           DAD   B
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2B3D EB         1150            XCHG
2B3E 21 09 00   1160            LXI   H,9
2B41 CD 27 20   1170            CALL  DIV       * DIVIDE AVAILABLE SPACE BY 9
2B44 54         1180            MOV   D,H       * IGNORE REMAINDER
2B45 5D         1190            MOV   E,L       * MOVE MAX FRAME COUNT TO DE
2B46 29         1200            DAD   H
2B47 29         1210            DAD   H
2B48 29         1220            DAD   H
2B49 19         1230            DAD   D         * (HL) = 9*FRAME COUNT
2B4A 0B         1240            DCX   B         * (BC) = -(BUFADR)-2
2B4B 79         1250            MOV   A,C
2B4C 95         1260            SUB   L
2B4D 6F         1270            MOV   L,A
2B4E 78         1280            MOV   A,B
2B4F 9C         1290            SBB   H
2B50 67         1300            MOV   H,A
2B51 22 EF 36   1310            SHLD  NEGBND    * -(BUFADR)-2-9*MAX FRAME CNT
2B54 21 05 35   1320            LXI   H,MATRIX+1
2B57 22 00 35   1330            SHLD  MATPTR    * POINT TO MAT COL 2 (AFTER #)
2B5A            1340 *
2B5A            1350 ******
2B5A            1360 *
2B5A            1370 *  EACH PASS THRU F0LOOP GENERATES F0 DATA INTO
2B5A            1380 *  FRAMES I+1 THRU I+DURATION X, CORRESPONDING TO
2B5A            1390 *  THE PHONEME IN MATRIX COLUMN X.  BUFPTR ALWAYS
2B5A            1400 *  POINTS TO F0(I) AT THE BEGINNING OF THE LOOP.
2B5A            1410 *
2B5A 00         1420 F0LOOP     NOP   *         * BREAKPOINT LOC
2B5B 2A 00 35   1430            LHLD  MATPTR
2B5E 01 5F 00   1440            LXI   B,MATLEN
2B61 5E         1450            MOV   E,M       * GET COL X CODE IN E
2B62 09         1460            DAD   B
2B63 56         1470            MOV   D,M       * GET COL X FEATA IN D
2B64 09         1480            DAD   B
2B65 7E         1490            MOV   A,M       * GET COL X FEATB IN A
2B66 09         1500            DAD   B         * MOVE PTR TO STRESS X
2B67 E6 40      1510            ANI   VOICE
2B69 C2 22 2C   1520            JNZ   VOICED    * IT'S A VOICED PHONEME
2B6C 09         1530            DAD   B
2B6D 7E         1540            MOV   A,M
2B6E 32 E2 36   1550            STA   DURX      * SAVE DURATION X
2B71 7B         1560            MOV   A,E
2B72 1E D8      1570            MVI   E,-40     * TERMINAL DROP IF "."
2B74 FE 01      1580            CPI   CPER
2B76 CA 80 2B   1590            JZ    ENDS      .
2B79 1E 28      1600            MVI   E,40      * TERMINAL RISE IF "?"
2B7B FE 03      1610            CPI   CQUEST
2B7D C2 11 2C   1620            JNZ   NOF0
2B80            1630 *
2B80            1640 *  COMPUTE PHRASE-TERMINAL F0 CHANGE
2B80            1650 *  ENTER WITH F0 OFFSET IN E
2B80            1660 *
2B80 7B         1670 ENDS       MOV   A,E
2B81 32 E6 36   1680            STA   FTERM     * SAVE F0 CHANGE VALUE
2B84 AF         1690            XRA   A
2B85 01 F7 FF   1700            LXI   B,-9
2B88 2A DF 36   1710            LHLD  BUFPTR    * SCAN BACK FOR LAST NON 0 F0
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2B8B B6           1720 ENDS1    ORA   M
2B8C C2 9C 2B     1730          JNZ   RAMP     * FOUND A NON 0 VALUE
2B8F 09           1740          DAD   B
2B90 EB           1750          XCHG
2B91 2A EB 36     1760          LHLD  NEGBUF
2B94 19           1770          DAD   D
2B95 D2 13 2C     1780          JNC   NOF0     * HIT FRONT OF BUFFER, EXIT
2B98 EB           1790          XCHG
2B99 C3 8B 2B     1800          JMP   ENDS1
2B9C 3A E6 36     1810 RAMP     LDA   FTERM    * NON 0 F0 + CHANGE
2B9F 86           1820          ADD   M
2BA0 FE 0F        1830          CPI   15
2BA2 D2 AA 2B     1840          JNC   RAMP1
2BA5 3E 0F        1850          MVI   A,15     * FINAL F0 < 15, MAKE IT 15
2BA7 C3 B1 2B     1860          JMP   RAMP2
2BAA FE 7D        1870 RAMP1    CPI   125
2BAC DA B1 2B     1880          JC    RAMP2
2BAF 3E 7D        1890          MVI   A,125    * FINAL F0 > 125, MAKE IT 125
2BB1 96           1900 RAMP2    SUB   M
2BB2 32 E6 36     1910          STA   FTERM    * THE REAL FINAL F0 OFFSET
2BB5 EB           1920          XCHG  *        * END-OF-RAMP ADDR TO DE
2BB6 06 1D        1930          MVI   B,29     * TRY TO MAKE A 290 MSEC RAMP
2BB8 05           1940 RAMP3    DCR   B
2BB9 78           1950          MOV   A,B      * TENTATIVE RAMP LENGTH
2BBA 87           1960          ADD   A
2BBB 87           1970          ADD   A
2BBC 87           1980          ADD   A
2BBD 80           1990          ADD   B
2BBE 2F           2000          CMA
2BBF 3C           2010          INR   A
2BC0 4F           2020          MOV   C,A      * -9*TRL INTO A AND C
2BC1 2A ED 36     2030          LHLD  NEGPSE
2BC4 19           2040          DAD   D        * AVAILABLE SPACE IN HL
2BC5 85           2050          ADD   L
2BC6 3E FF        2060          MVI   A,0FFH
2BC8 8C           2070          ADC   H
2BC9 D2 B8 2B     2080          JNC   RAMP3    * NO ROOM, SHORTEN RAMP
2BCC 69           2090          MOV   L,C
2BCD 26 FF        2100          MVI   H,0FFH
2BCF 19           2110          DAD   D
2BD0 E5           2120          PUSH  H        * BEGIN-RAMP ADDR, SAVE IT
2BD1 3A E6 36     2130          LDA   FTERM
2BD4 4F           2140          MOV   C,A
2BD5 B7           2150          ORA   A
2BD6 F2 DB 2B     2160          JP    $+5      * OFFSET IS +
2BD9 2F           2170          CMA
2BDA 3C           2180          INR   A        * OFFSET IS -, MAKE IT +
2BDB 57           2190          MOV   D,A
2BDC 1E 00        2200          MVI   E,0      * 256*OFFSET INTO DE
2BDE 68           2210          MOV   L,B
2BDF 26 00        2220          MVI   H,0      * RAMP FRAME COUNT IN HL
2BE1 CD 27 20     2230          CALL  DIV
2BE4 79           2240          MOV   A,C
2BE5 B7           2250          ORA   A
2BE6 F2 F0 2B     2260          JP    RAMP4    * OFFSET IS +, OK
2BE9 7C           2270          MOV   A,H      * OFFSET IS -, MAKE DELTA -
2BEA 2F           2280          CMA
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2BEB 67         2290            MOV   H,A
2BEC 7D         2300            MOV   A,L
2BED 2F         2310            CMA
2BEE 6F         2320            MOV   L,A
2BEF 23         2330            INX   H
2BF0 22 E7 36   2340 RAMP4      SHLD  DELTA    * DELTA=256*OFFSET/RAMP LENGTH
2BF3 11 00 00   2350            LXI   D,0
2BF6 C3 FD 2B   2360            JMP   RAMP5+4
2BF9 E5         2370 RAMP5      PUSH  H        * SAVE BUFFER POINTER
2BFA 2A E7 36   2380            LHLD  DELTA
2BFD 19         2390            DAD   D        * DIF=DIF+DELTA
2BFE E3         2400            XTHL  *        * DIF TO STACK
2BFF EB         2410            XCHG  *        * BUFFER PTR TO DE
2C00 21 09 00   2420            LXI   H,9
2C03 19         2430            DAD   D        * UPDATE BUFFER PTR
2C04 D1         2440            POP   D        * DIF TO DE
2C05 7E         2450            MOV   A,M      * F0=OLDF0+DIF/256
2C06 B7         2460            ORA   A
2C07 CA 0C 2C   2470            JZ    $+5      * IT' 0, DON'T CHANGE IT
2C0A 82         2480            ADD   D
2C0B 77         2490            MOV   M,A
2C0C 05         2500            DCR   B
2C0D C2 F9 2B   2510            JNZ   RAMP5
2C10 C3 E7 2C   2520            JMP   ENDF0
2C13            2530 *
2C13            2540 *  UNVOICED PHON, GENERATE ARC TO STRESS 2 LEVEL
2C13            2550 *
2C13 3A E2 36   2560 NOF0       LDA   DURX
2C16 4F         2570            MOV   C,A
2C17 06 05      2580            MVI   B,5
2C19 2A DF 36   2590            LHLD  BUFPTR
2C1C CD 12 2D   2600            CALL  FPARAB   * PARABOLA PEAKS AT END OF PHON
2C1F C3 E7 2C   2610            JMP   ENDF0
2C22            2620 *
2C22            2630 *  GENERATE F0 FOR VOICED PHONS
2C22            2640 *  ON ENTRY: (D)=FEATA X, (B)=MATLEN
2C22            2650 *            HL POINTS TO STRESS X
2C22            2660 *
2C22 5E         2670 VOICED     MOV   E,M      * GET STRESS X IN E
2C23 09         2680            DAD   B
2C24 7E         2690            MOV   A,M
2C25 32 E2 36   2700            STA   DURX     * GET DUR X & SAVE IT
2C28 7A         2710            MOV   A,D      * GET FEATA X
2C29 E6 80      2720            ANI   VOWEL
2C2B CA 63 2C   2730            JZ    VNEXT    * X NOT A VOWEL, TRY X+1
2C2E 7B         2740            MOV   A,E      * X IS VOWEL, SEE IF STRESSED
2C2F B7         2750            ORA   A
2C30 CA 63 2C   2760            JZ    VNEXT    * NO, TRY X+1
2C33 32 E1 36   2770            STA   STRSX    * SAVE STRESS VALUE
2C36 06 00      2780            MVI   B,0
2C38 3A E2 36   2790            LDA   DURX     * GET DURX AGAIN
2C3B B7         2800 VSTRS      ORA   A        * DIVIDE IT BY 4
2C3C 1F         2810            RAR
2C3D B7         2820            ORA   A
2C3E 1F         2830            RAR
2C3F 80         2840            ADD   B        * ADD EITHER 0 OR DURX
2C40 32 E3 36   2850            STA   DURX4    * EITHER DURX/4 OR DURX+DURY/4
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2C43 4F         2860            MOV   C,A
2C44 3A E1 36   2870            LDA   STRSX    * MOVE STRESS INTO B
2C47 47         2880            MOV   B,A
2C48 2A DF 36   2890            LHLD  BUFPTR
2C4B CD 12 2D   2900            CALL  FPARAB   * COMPUTE FRONT OF PARABOLA
2C4E C0         2910            RNZ    *       * ERROR RETURN
2C4F 3A E3 36   2920            LDA   DURX4
2C52 47         2930            MOV   B,A
2C53 3A E2 36   2940            LDA   DURX
2C56 90         2950            SUB   B
2C57 4F         2960            MOV   C,A      * FULL DUR - FRONT DUR INTO C
2C58 3A E1 36   2970            LDA   STRSX
2C5B 47         2980            MOV   B,A
2C5C CD 7F 2D   2990            CALL  BPARAB   * COMPUTE BACK OF PARABOLA
2C5F C0         3000            RNZ    *       * ERROR RETURN
2C60 C3 E7 2C   3010            JMP   ENDF0
2C63            3020 *
2C63            3030 *  CURRENT VOICED PHON IS NOT A STRESSED VOWEL
2C63            3040 *  IF NEXT ONE IS, GENERATE PARABOLA ACROSS BOTH
2C63            3050 *
2C63 2A DF 36   3060 VNEXT      LHLD  BUFPTR   * GET LAST F0 OF PREV PHON
2C66 7E         3070            MOV   A,M
2C67 B7         3080            ORA   A
2C68 C2 6D 2C   3090            JNZ   $+5
2C6B 36 45      3100            MVI   M,69     * LAST F0 WAS 0, MAKE IT 69
2C6D 2A 00 35   3110            LHLD  MATPTR
2C70 23         3120 VNXT1      INX   H
2C71 54         3130            MOV   D,H
2C72 5D         3140            MOV   E,L
2C73 7E         3150            MOV   A,M      * GET NEXT COL CODE
2C74 FE 04      3160            CPI   CTERM
2C76 CA A6 2C   3170            JZ    DOWNDR   * END OF MATRIX, FINISB X
2C79 09         3180            DAD   B
2C7A 7E         3190            MOV   A,M
2C7B E6 04      3200            ANI   WDBND
2C7D CA 84 2C   3210            JZ    VNXT2    * NOT A WORD BOUNDARY
2C80 EB         3220            XCHG
2C81 C3 70 2C   3230            JMP   VNXT1    * X+1 IS WDBOUND, BUMP AGAIN
2C84 7E         3240 VNXT2      MOV   A,M      * GET COL Y FEATA
2C85 E6 80      3250            ANI   VOWEL
2C87 CA A6 2C   3260            JZ    DOWNDR   * Y NOT A VOWEL, DOWNDRIFT X
2C8A 09         3270            DAD   B
2C8B 09         3280            DAD   B
2C8C 7E         3290            MOV   A,M      * GET COL Y STRESS
2C8D B7         3300            ORA   A
2C8E CA A6 2C   3310            JZ    DOWNDR   * Y IS VOWEL BUT NOT STRESSED
2C91 32 E1 36   3320            STA   STRSX
2C94 09         3330            DAD   B
2C95 4E         3340            MOV   C,M      * GET COL Y DURATION
2C96 3A E2 36   3350            LDA   DURX
2C99 47         3360            MOV   B,A
2C9A 81         3370            ADD   C
2C9B 32 E2 36   3380            STA   DURX     * DURX=DUR X + DUR Y
2C9E EB         3390            XCHG
2C9F 22 00 35   3400            SHLD  MATPTR   * MOVE MATPTR UP TO COL Y
2CA2 79         3410            MOV   A,C      * GET COL Y DURATION
2CA3 C3 3B 2C   3420            JMP   VSTRS
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2CA6            3430 *
2CA6            3440 *  LET F0 DRIFT DOWNWARD TOWARD 43 (100 HZ)
2CA6            3450 *
2CA6 2A 00 35   3460 DOWNDR     LHLD  MATPTR
2CA9 01 7C 01   3470            LXI   B,MATLEN*4
2CAC 09         3480            DAD   B
2CAD 4E         3490            MOV   C,M       * GET DUR X IN C
2CAE 2A DF 36   3500            LHLD  BUFPTR
2CB1 56         3510            MOV   D,M       * GET PREVIOUS F0 IN D
2CB2 1E 00      3520            MVI   E,0
2CB4 D5         3530            PUSH  D
2CB5 E3         3540 DWN1       XTHL  *         * BUFPTR TO STACK, GET F0
2CB6 7C         3550            MOV   A,H       * SHIFT HL RIGHT 5
2CB7 29         3560            DAD   H
2CB8 29         3570            DAD   H
2CB9 29         3580            DAD   H
2CBA 6C         3590            MOV   L,H
2CBB 07         3600            RLC
2CBC 07         3610            RLC
2CBD 07         3620            RLC
2CBE E6 07      3630            ANI   7
2CC0 67         3640            MOV   H,A
2CC1 2F         3650            CMA
2CC2 57         3660            MOV   D,A       * MULTIPLY (HL) BY 31
2CC3 7D         3670            MOV   A,L       *  IE. HL=(HL)*32-(HL)
2CC4 2F         3680            CMA
2CC5 5F         3690            MOV   E,A
2CC6 13         3700            INX   D
2CC7 29         3710            DAD   H
2CC8 29         3720            DAD   H
2CC9 29         3730            DAD   H
2CCA 29         3740            DAD   H
2CCB 29         3750            DAD   H
2CCC 19         3760            DAD   D
2CCD 11 58 01   3770            LXI   D,8*43    * ADD 43 SHIFTED 3 LEFT
2CD0 19         3780            DAD   D
2CD1 7C         3790            MOV   A,H       * NEW F0 VALUE INTO A
2CD2 D1         3800            POP   D         * GET BUFFER PTR
2CD3 E5         3810            PUSH  H
2CD4 21 09 00   3820            LXI   H,9
2CD7 19         3830            DAD   D         * UPDATE TO NEXT FRAME
2CD8 EB         3840            XCHG
2CD9 2A EF 36   3850            LHLD  NEGBND
2CDC 19         3860            DAD   D
2CDD DA AC 2D   3870            JC    BFERR-1   * SORRY, BUFFER IS FULL
2CE0 EB         3880            XCHG
2CE1 77         3890            MOV   M,A       * PUT F0 IN BUFFER
2CE2 0D         3900            DCR   C
2CE3 C2 B5 2C   3910            JNZ   DWN1
2CE6 D1         3920            POP   D
2CE7            3930 *
2CE7            3940 *  END F0LOOP, STEP MATRIX TO NEXT PHON
2CE7            3950 *
2CE7 22 DF 36   3960 ENDF0      SHLD  BUFPTR    * SAVE BUFFER POINTER
2CEA 2A 00 35   3970            LHLD  MATPTR
2CED 23         3980            INX   H
2CEE 22 00 35   3990            SHLD  MATPTR    * INCREMENT MATRIX POINTER
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2CF1 EB         4000            XCHG
2CF2 2A 02 35   4010            LHLD  NEGEND
2CF5 19         4020            DAD   D
2CF6 D2 5A 2B   4030            JNC   FOLOOP  * THERE'S MORE TO THIS MATRIX
2CF9 2A EB 36   4040            LHLD  NEGBUF
2CFC EB         4050            XCHG
2CFD 2A DF 36   4060            LHLD  BUFPTR
2D00 19         4070            DAD   D
2D01 EB         4080            XCHG
2D02 21 09 00   4090            LXI   H,9       * BUFFER SPACE USED /9
2D05 CD 27 20   4100            CALL  DIV       *  = OVERALL FRAME COUNT
2D08 23         4110            INX   H         * BUT THAT WAS 1 FRAME SHORT
2D09 EB         4120            XCHG
2D0A 2A 06 20   4130            LHLD  BUFADR    * PUT IT IN 1ST 2 BYTES OF BUFFER
2D0D 73         4140            MOV   M,E
2D0E 23         4150            INX   H
2D0F 72         4160            MOV   M,D
2D10 AF         4170            XRA   A
2D11 C9         4180            RET
2D12            4190 *
2D12            4200 **********
2D12            4210 *
2D12            4220 *  PARABOLA GENERATOR SUBROUTINES
2D12            4230 *
2D12            4240 *  FPARAB
2D12            4250 *  CONSTRUCT FRONT PARABOLA CURVE UP TO PEAK
2D12            4260 *  ON ENTRY: (B)=STRESS X, (C)=FRAME COUNT TO PEAK
2D12            4270 *            HL POINTS TO F0 IN LAST FRAME OF PREV PHON
2D12            4280 *
2D12 AF         4290 FPARAB     XRA   A
2D13 B9         4300            CMP   C         * RETURN IF N (FRAME COUNT) =0
2D14 C8         4310            RZ
2D15 56         4320            MOV   D,M       * GET PREV F0 INTO D
2D16 E5         4330            PUSH  H
2D17 BA         4340            CMP   D
2D18 C2 1F 2D   4350            JNZ   FPRB1     * PREV F0 WAS NON-0, USE IT
2D1B CD BF 2D   4360            CALL  GETF0     * PREV WAS 0, GET ONE FROM TABLE
2D1E 56         4370            MOV   D,M
2D1F 3E 05      4380 FPRB1      MVI   A,5       * INDEX TO 2ND COL OF TABLE
2D21 CD BF 2D   4390            CALL  GETF0     * GET F0 LEVEL AT PARAB PEAK
2D24 7E         4400            MOV   A,M
2D25 92         4410            SUB   D
2D26 47         4420            MOV   B,A
2D27 21 E5 36   4430            LXI   H,HF0+1 * SET HF0 TO STARTING PT
2D2A 72         4440            MOV   M,D
2D2B 57         4450            MOV   D,A
2D2C 69         4460            MOV   L,C
2D2D 2C         4470            INR   L
2D2E AF         4480            XRA   A
2D2F 5F         4490            MOV   E,A
2D30 67         4500            MOV   H,A
2D31 CD 27 20   4510            CALL  DIV
2D34 29         4520            DAD   H         * DELTA=2*((256*DIF)/(N+1))
2D35 59         4530            MOV   E,C       * MOVE FRAME COUNT TO E
2D36 22 E7 36   4540 PARAB2     SHLD  DELTA
2D39 60         4550            MOV   H,B       * MOVE B & C OUT OF MUL'S WAY
2D3A 69         4560            MOV   L,C
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2D3B 43         4570            MOV   B,E      * GET EITHER N OR N-1 FROM E
2D3C 04         4580            INR   B
2D3D CD 24 20   4590            CALL  MUL      * DENOM=(B)*((B)-1)
2D40 4D         4600            MOV   C,L
2D41 EB         4610            XCHG
2D42 1E 00      4620            MVI   E,0
2D44 CD 27 20   4630            CALL  DIV      * S=2*((256*DIF)/DENOM)
2D47 29         4640            DAD   H
2D48 7C         4650            MOV   A,H
2D49 2F         4660            CMA
2D4A 67         4670            MOV   H,A
2D4B 7D         4680            MOV   A,L
2D4C 2F         4690            CMA
2D4D 6F         4700            MOV   L,A
2D4E 23         4710            INX   H
2D4F 22 E9 36   4720            SHLD  SD
2D52 AF         4730            XRA   A
2D53 32 E4 36   4740            STA   HF0      * CLEAR LS PART OF HF0
2D56 E1         4750            POP   H
2D57 EB         4760 PARAB3     XCHG
2D58 21 09 00   4770            LXI   H,9      * MOVE BUFFER PTR TO NEXT FRAME
2D5B 19         4780            DAD   D
2D5C EB         4790            XCHG
2D5D 2A EF 36   4800            LHLD  NEGBND
2D60 19         4810            DAD   D
2D61 DA AD 2D   4820            JC    BFERR    * RAN OUT OF ROOM!
2D64 D5         4830            PUSH  D
2D65 2A E7 36   4840            LHLD  DELTA
2D68 EB         4850            XCHG
2D69 2A E4 36   4860            LHLD  HF0
2D6C 19         4870            DAD   D
2D6D 22 E4 36   4880            SHLD  HF0      * HF0=HF0+DELTA
2D70 44         4890            MOV   B,H      * H CONTAINS NEW F0 VALUE
2D71 2A E9 36   4900            LHLD  SD
2D74 19         4910            DAD   D
2D75 22 E7 36   4920            SHLD  DELTA    * DELTA=DELTA+S
2D78 E1         4930            POP   H        * GET BUF PTR BACK
2D79 70         4940            MOV   M,B
2D7A 0D         4950            DCR   C
2D7B C2 57 2D   4960            JNZ   PARAB3
2D7E C9         4970            RET
2D7F            4980 *
2D7F            4990 *  BPARAB
2D7F            5000 *  CONSTRUCT BACK PARABOLIC CURVE DOWN FROM PEAK
2D7F            5010 *  ON ENTRY: (B)=STRESS X, (C)=FRAME CNT DOWN FROM PEAK
2D7F            5020 *            HL POINTS TO F0 PARAM AT PEAK
2D7F            5030 *'
2D7F AF         5040 BPARAB     XRA   A
2D80 B9         5050            CMP   C        * RETURN IF FRAME COUNT =0
2D81 C8         5060            RZ
2D82 56         5070            MOV   D,M      * GET PREV F0 VALUE
2D83 E5         5080            PUSH  H
2D84 3E 0A      5090            MVI   A,10
2D86 CD BF 2D   5100            CALL  GETF0    * SET HL TO END-PARAB LEVEL
2D89 7A         5110            MOV   A,D
2D8A 32 E5 36   5120            STA   HF0+1    * SAVE OLD F0
2D8D 96         5130            SUB   M
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2D8E 47         5140            MOV   B,A      * DIFF TO NEW F0 INTO B
2D8F 21 00 00   5150            LXI   H,0
2D92 59         5160            MOV   E,C
2D93 1D         5170            DCR   E
2D94 C2 36 2D   5180            JNZ   PARAB2   * N>1, COMPUTE PARABOLA
2D97 3A E5 36   5190            LDA   HF0+1    * N=1, STORE 1 POINT
2D9A 80         5200            ADD   B
2D9B D1         5210            POP   D
2D9C 21 09 00   5220            LXI   H,9      * INCR PTR TO NEXT FRAME
2D9F 19         5230            DAD   D
2DA0 EB         5240            XCHG
2DA1 2A EF 36   5250            LHLD  NEGBND
2DA4 19         5260            DAD   D
2DA5 DA AD 2D   5270            JC    BFERR    * NO MORE ROOM
2DA8 EB         5280            XCHG
2DA9 77         5290            MOV   M,A
2DAA AF         5300            XRA   A
2DAB C9         5310            RET
2DAC            5320 *
2DAC            5330 *  ERROR SUBROUTINE
2DAC            5340 *
2DAC D1         5350            POP   D        * ENTRY HERE FROM DOWNDR
2DAD 21 B3 2D   5360 BFERR      LXI   H,BFTX
2DB0 F6 FF      5370            ORI   255
2DB2 C9         5380            RET
2DB3            5390 *
2DB3 42 55 46   5400 BFTX       DT    'BUFFER FULL'
2DB6 46 45 52
2DB9 20 46 55
2DBC 4C 4C
2DBE 0D         5410            DB    0DH
2DBF            5420 *
2DBF            5430 *  GET AN F0 LEVEL FROM STRSF0 TABLE (SET HL TO IT)
2DBF            5440 *  ON ENTRY:
2DBF            5450 *    A TELLS THE COLUMN, 0=SP, 5=MP, 10=EP
2DBF            5460 *    B CONTAINS THE STRESS LEVEL 1-5
2DBF            5470 *  ON RETURN, HL POINTS TO THE DESIRED ENTRY
2DBF            5480 *
2DBF 21 C9 2D   5490 GETF0      LXI   H,STRSF0-1
2DC2 80         5500            ADD   B
2DC3 85         5510            ADD   L
2DC4 6F         5520            MOV   L,A
2DC5 3E 00      5530            MVI   A,0
2DC7 8C         5540            ADC   H
2DC8 67         5550            MOV   H,A
2DC9 C9         5560            RET
2DCA            5570 *
2DCA            5580 *  F0 TABLE FOR DIFFERENT STRESS LEVELS
2DCA            5590 *
2DCB 63         5600 STRSF0     DB    99    * PARAB STARTING PT., STRESS 1
2DCB 59         5610            DB    89    * S.P., STRESS 2
2DCC 54         5620            DB    84    *  ETC.
2DCD 52         5630            DB    82
2DCE 51         5640            DB    81
2DCF 6C         5650            DB    108   * PARAB MID POINT (PEAK)
2DD0 63         5660            DB    99
2DD1 5E         5670            DB    94
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2DD2 5C         5680            DB    92
2DD3 5B         5690            DB    91
2DD4 59         5700            DB    89    * PARAB END POINT
2DD5 4F         5710            DB    79
2DD6 4A         5720            DB    74
2DD7 47         5730            DB    71
2DD8 46         5740            DB    70
2DD9            5750 *
2DD9            5760 **********
2DD9            5770 *
2DD9            5780 *  CLRBUF IS CALLED BY GENPRM TO INITIALIZE
2DD9            5790 *  THE PARAMETER BUFFER (OR PART OF IT)
2DD9            5800 *  VALUES IN THE F0 PARAMETER ARE NOT DISTURBED
2DD9            5810 *
2DD9            5820 *  ON ENTRY: (BC)= THE # OF FRAMES TO INITIALIZE
2DD9            5830 *            STARTING WITH THE 2ND FRAME OF THE BUFFER
2DD9            5840 *
2DD9 2A 06 20   5850 CLRBUF     LHLD  BUFADR
2DDC 11 0B 00   5860            LXI   D,11
2DDF 19         5870            DAD   D
2DE0 16 80      5880            MVI   D,128
2DE2 1E AC      5890            MVI   E,172
2DE4 AF         5900 CLOOP      XRA   A
2DE5 77         5910            MOV   M,A     * AV=0
2DE6 23         5920            INX   H
2DE7 23         5930            INX   H
2DE8 72         5940            MOV   M,D     * F1=128
2DE9 23         5950            INX   H
2DEA 72         5960            MOV   M,D     * F2=128
2DEB 23         5970            INX   H
2DEC 73         5980            MOV   M,E     * F3=172
2DED 23         5990            INX   H
2DEE 77         6000            MOV   M,A     * AH=0
2DEF 23         6010            INX   H
2DF0 77         6020            MOV   M,A     * AF=0
2DF1 23         6030            INX   H
2DF2 72         6040            MOV   M,D     * FF=128
2DF3 23         6050            INX   H
2DF4 77         6060            MOV   M,A     * AN=0
2DF5 23         6070            INX   H
2DF6 0B         6080            DCX   B
2DF7 B0         6090            ORA   B
2DF8 B1         6100            ORA   C
2DF9 C2 E4 2D   6110            JNZ   CLOOP
2DFC C9         6120            RET
2DFD            6130 *
2DFD            6140 *  END OF SECT4
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  GENPRM
2000            0020 *
2000            0030 *  SECTION 5 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.0d   MAY 30, 1977
2000            0070 *
2000            0080 **********
2000            0090 *
2000            0100 *  COMMON JUMP ADDRESS TABLE
2000            0110 *
2000            0120 COMJMP     EQU   $
2000            0130 *
2000            0140 CSR1       DS    1
2003            0150 PLAY       DS    3
2006            0160 BUFADR     DS    2
2008            0170 BUFLEN     DS    2
200A            0180 PVTAB      DS    2
200C            0190 MATPAR     DS    3
200F            0200 MATERR     DS    3
2012            0210 RULES      DS    3
2015            0220 SETDUR     DS    3
2018            0230 RULES3     DS    3
201B            0240 GENPO      DS    3
201E            0250 CLRBUF     DS    3
2021 C3 00 2E   0260            JMP   GENPRM
2024 C3 00 32   0270            JMP   MUL
2027 C3 13 32   0280            JMP   DIV
202A            0290 GETONS     DS    3
202D            0300 PCBF1      DS    3
2030            0310 F1TAR      DS    2
2032            0320 F2TAR      DS    2
2034            0330 F3TAR      DS    2
2036            0340 AVGTAR     DS    2
2038            0350 FRNTAR     DS    2
203A            0360 *
203A            0370 **********
203A            0380 *
203A            0390 *  COMRAM ORIGEN DEFINITION
203A            0400 *
203A            0410            ORG   COMJMP+1500H
3500            0420 COMRAM     EQU   $
3500            0430 *
3500            0440 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0450 *
3500            0460 MATPTR     DS    2
3502            0470 NEGEND     DS    2
3504            0480 MATRIX     EQU   $
3504            0490 MATLEN     EQU   95
3504            0500 PHCODE     DS    MATLEN
3563            0510 FEATA      DS    MATLEN
35C2            0520 FEATB      DS    MATLEN
3621            0530 STRES      DS    MATLEN
3680            0540 DUR        DS    MATLEN
36DF            0550 MATEND     EQU   $
36DF            0560 BUFPTR     DS    2
36E1            0570 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

36E1            0580 *  THE NEXT 19 LOCATIONS ARE ACCESSED BY ROUTINES
36E1            0590 *  IN SECTION 6 AND MUST NOT BE MOVED
36E1            0600 *
36E1            0610 OCODE      DS    1
36E2            0620 OFEATA     DS    1
36E3            0630 OFEATB     DS    1
36E4            0640 ODUR       DS    1
36E5            0650 OCID       DS    1
36E6            0660 ORANK      DS    1
36E7            0670 OPCT       DS    1
36E8            0680 OTIMES     DS    1
36E9            0690 *
36E9            0700 CODEX      DS    1
36EA            0710 FEATAX     DS    1
36EB            0720 FEATBX     DS    1
36EC            0730 DURX       DS    1
36ED            0740 CIDX       DS    1
36EE            0750 RANKX      DS    1
36EF            0760 PCTX       DS    1
36F0            0770 TIMESX     DS    1
36F1            0780 *
36F1            0790 BT         DS    1
36F2            0800 FT         DS    1
36F3            0810 FC         DS    1
36F4            0820 *
36F4            0830 *  GENPRM (LOCAL) RAM WORKSPACE
36F4            0840 *
36F4            0850 TARG       DS    1
36F5            0860 BVAL       DS    1
36F6            0870 SD         DS    2
36F8            0880 DELTA      DS    2
36FA            0890 PCOUNT     DS    1
36FB            0900 NTARGS     DS    1
36FC            0910 ASPT       DS    1
36FD            0920 *
36FD            0930 **********
36FD            0940 *
36FD            0950 *  GENPRM
36FD            0960 *
36FD            0970            ORG   COMJMP+0E00H
2E00            0980 SECTAD     EQU   $
2E00            0990 *
2E00            1000 **********
2E00            1010 *
2E00            1020 *  PHONEME CODE DEFINITIONS
2E00            1030 *
2E00            1040 STERM      EQU   4
2E00            1050 CP         EQU   39
2E00            1060 *
2E00            1070 *  FEATURE LABEL DEFINITIONS
2E00            1080 *
2E00            1090 IGNORE     EQU   1
2E00            1100 STOP       EQU   30H
2E00            1110 PLOS       EQU   20H
2E00            1120 PLOSA      EQU   10H
2E00            1130 *
2E00            1140 *  OUTPUT CHANNEL DEFINITIONS
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2E00            1150 *
2E00            1160 CHANAV     EQU   0
2E00            1170 CHANF1     EQU   2
2E00            1180 CHANF2     EQU   3
2E00            1190 CHANF3     EQU   4
2E00            1200 CHANAH     EQU   5
2E00            1210 CHANAF     EQU   6
2E00            1220 CHANFF     EQU   7
2E00            1230 CHANAN     EQU   8
2E00            1240 *
2E00            1250 **********
2E00            1260 *
2E00            1270 *  GENPRM SUBROUTINE
2E00            1280 *
2E00 21 05 35   1290 GENPRM     LXI   H,MATRIX+1
2E03 22 00 35   1300            SHLD  MATPTR  * SET MATPTR TO COL 2 (SKIP #)
2E06 11 E1 36   1310            LXI   D,OCODE
2E09 CD 2A 20   1320            CALL  GETCNS  * GET CONSTANTS FOR COL 2 PAUSE
2E0C 3A E4 36   1330            LDA   ODUR
2E0F 4F         1340            MOV   C,A
2E10 06 00      1350            MVI   B,0
2E12 CD 1E 20   1360            CALL  CLRBUF  * INITIALIZE THE 1ST FRAME
2E15 11 F7 FF   1370            LXI   D,-9
2E18 19         1380            DAD   D
2E19 22 DF 36   1390            SHLD  BUFPTR  * SET BUFPTR TO LAST FRAME AV
2E1C 21 06 35   1400            LXI   H,MATRIX+2
2E1F 22 00 35   1410            SHLD  MATPTR  * MATPTR PTS TO 1ST ACTUAL PHON
2E22            1420 *
2E22            1430 *
2E22            1440 *  DURING PROCESSING FOR EACH PHON, MATPTR INDICATES
2E22            1450 *  THE CURRENT MATRIX COLUMN.  BUFPTR IS THE PARAMETER
2E22            1460 *  BUFFER INDEX.  IT CONTAINS THE ADDRESS OF THE AV
2E22            1470 *  PARAMETER IN THE LAST FRAME OF THE PREVIOUSLY
2E22            1480 *  PROCESSED PHON.
2E22            1490 *
2E22 11 E9 36   1500 GPLOOP     LXI   D,CODEX
2E25 CD 2A 20   1510            CALL  GETCNS  * GET CONSTANTS FOR NEXT PHON
2E28 3A EA 36   1520            LDA   FEATAX
2E2B E6 01      1530            ANI   IGNORE
2E2D C2 7C 30   1540            JNZ   ENDPRM  * IGNORE THIS MATRIX COLUMN
2E30 3A E3 36   1550            LDA   CFEATB
2E33 E6 20      1560            ANI   PLOS
2E35 CA 40 2E   1570            JZ    NOPLOS  * THE PRECEEDING WAS NOT A PLOSIV
2E38 3A EB 36   1580            LDA   FEATBX  * PREVIOUS WAS PLOSIVE
2E3B E6 80      1590            ANI   STOP
2E3D CA 30 2F   1600            JZ    SETASP  * CURR IS -STOP, SET ASPIRATION
2E40            1610 *
2E40            1620 *  NOPLOS IS THE MAIN GENERATOR LOOP FOR MOST
2E40            1630 *  PHON SEQUENCES.
2E40            1640 *  FORMANT TRANSITIONS ARE GOVERNED BY RELATIVE RANKS
2E40            1650 *
2E40 3A EE 36   1660 NOPLOS     LDA   RANKX   * COMPARE RANKS OF CURRENT
2E43 47         1670            MOV   B,A
2E44 3A E6 36   1680            LDA   ORANK   *    AND PREVIOUS PHONS
2E47 B8         1690            CMP   B
2E48 CA 6D 2E   1700            JZ    EQRANK  * RANKS ARE EQUAL
2E4B DA 85 2E   1710            JC    OLTCUR  * OLD LESS THAN CURRENT
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2E4E 3A E8 36   1720            LDA   OTIMES  * OLD GREATER THAN CURRENT
2E51 6F         1730            MOV   L,A
2E52 E6 0F      1740            ANI   0FH
2E54 32 F1 36   1750            STA   BT      * BACK TIME = OLD TPRI
2E57 26 00      1760            MVI   H,0
2E59 29         1770            DAD   H
2E5A 29         1780            DAD   H
2E5B 29         1790            DAD   H
2E5C 29         1800            DAD   H
2E5D 7C         1810            MOV   A,H
2E5E 32 F2 36   1820            STA   FT      * FORWARD TIME = OLD TSEC
2E61 3A E7 36   1830            LDA   OPCT
2E64 2F         1840            CMA
2E65 C6 05      1850            ADI   5
2E67 32 F3 36   1860            STA   PC      * PC=1.0-OPCT(BN PT RT OF BIT 2)
2E6A C3 9E 2E   1870            JMP   SETFOR
2E6D            1880 *
2E6D 3A E8 36   1890 EQRANK     LDA   OTIMES
2E70 E6 0F      1900            ANI   0FH
2E72 32 F1 36   1910            STA   BT      * BACK TIME = OLD TPRI
2E75 3A FC 36   1920            LDA   TIMESX
2E78 E6 0F      1930            ANI   0FH
2E7A 32 F2 36   1940            STA   FT      * FORWARD TIME = CURR TPRI
2E7D 3E 02      1950            MVI   A,2
2E7F 32 F3 36   1960            STA   PC      * PC=.5 (BIN PT RIGHT OF BIT 2)
2E82 C3 9E 2E   1970            JMP   SETFOR
2E85            1980 *
2E85 3A FC 36   1990 OLTCUR     LDA   TIMESX
2E88 6F         2000            MOV   L,A
2E89 E6 0F      2010            ANI   0FH
2E8B 32 F2 36   2020            STA   FT      * FORWARD TIME = CURR TPRI
2E8E 26 00      2030            MVI   H,0
2E90 29         2040            DAD   H
2E91 29         2050            DAD   H
2E92 29         2060            DAD   H
2E93 29         2070            DAD   H
2E94 7C         2080            MOV   A,H
2E95 32 F1 36   2090            STA   BT      * BACK TIME = CURR TSEC
2E98 3A EF 36   2100            LDA   PCTX
2E9B 32 F3 36   2110            STA   PC      * PC = CURRENT PCT
2E9E            2120 *
2E9E            2130 *  COMPUTE & STORE 3 FORMANT PARAMETERS
2E9E            2140 *
2E9E 2A 30 20   2150 SETFOR     LHLD  F1TAR
2EA1 EB         2160            XCHG
2EA2 01 02 00   2170            LXI   B,CHANF1
2EA5 CD AE 30   2180            CALL  GENFX   * SET FORMANT 1
2EA8 2A 32 20   2190            LHLD  F2TAR
2EAB EB         2200            XCHG
2EAC 01 03 00   2210            LXI   B,CHANF2
2EAF CD AE 30   2220            CALL  GENFX   * SET FORMANT 2
2EB2 2A 34 20   2230            LHLD  F3TAR
2EB5 EB         2240            XCHG
2EB6 01 04 00   2250            LXI   B,CHANF3
2EB9 CD AE 30   2260            CALL  GENFX   * SET FORMANT 3
2EBC            2270 *
2EBC            2280 *  COMPUTE & STORE AV
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2EBC            2290 *
2EBC C6 00      2300 SETAMP     MVI   B,0
2EBE CD 2D 2C   2310            CALL  PCBFT    * SETUP PC,BT & FT WITH AV INFO
2EC1 2A 36 2D   2320            LHLD  AVHTAR
2EC4 3A E9 36   2330            LDA   CODEX
2EC7 16 00      2340            MVI   D,0
2EC9 5F         2350            MOV   E,A
2ECA 19         2360            DAD   D
2ECB 7E         2370            MOV   A,M      * GET AV/AH TARGET BYTE
2ECC F5         2380            PUSH  PSW
2ECD E6 F0      2390            ANI   0F0H     * MASK TO KEEP AV
2ECF 01 00 00   2400            LXI   B,CHANAV
2ED2 CD C6 3C   2410            CALL  GENAX    * SET AMPLITUDE OF VOICING
2ED5            2420 *
2ED5            2430 *  COMPUTE & STORE AH
2ED5            2440 *
2ED5 F1         2450            POP   PSW
2ED6 E6 0F      2460            ANI   0FH      * NOW MASK TO KEEP AH
2ED8 87         2470            ADD   A
2ED9 87         2480            ADD   A        * MULT BY 4
2EDA 01 05 00   2490            LXI   B,CHANAH
2EDD CD C6 3C   2500            CALL  GENAX    * SET AMPLITUDE OF HISS
2EE0            2510 *
2EE0            2520 *  COMPUTE & STORE AF
2EE0            2530 *
2EE0 06 01      2540            MVI   B,1
2EE2 CD 2D 2C   2550            CALL  PCBFT    * SET PC, BT & FT WITH AF INFO
2EE5 2A 38 2D   2560            LHLD  FRNTAR
2EE8 3A E9 36   2570            LDA   CODEX
2EEB 5F         2580            MOV   E,A
2EEC 16 00      2590            MVI   D,0
2EEE 19         2600            DAD   D
2EEF 7E         2610            MOV   A,M      * GET FRIC/NASAL TARGET BYTE
2EF0 F5         2620            PUSH  PSW
2EF1 E6 1C      2630            ANI   1CH      * MASK TO KEEP AF TARGET
2EF3 87         2640            ADD   A
2EF4 01 C6 00   2650            LXI   B,CHANAF
2EF7 CD C6 3C   2660            CALL  GENAX    * SET AMPLITUDE OF FRICATION
2EFA            2670 *
2EFA            2680 *  SET FF TO TARGET VALUE FOR TIME DURX
2EFA            2690 *
2EFA F1         2700            POP   PSW      * GET FRIC TARG AGAIN
2EFB 3B         2710            DCX   SP
2EFC 3B         2720            DCX   SP       * BUT LEAVE IT IN THE STACK, TOO
2EFD E6 E0      2730            ANI   0E0H     * KEEP FF TARGET
2EFF 57         2740            MOV   D,A
2F00 01 07 00   2750            LXI   B,CHANFF
2F03 2A DF 36   2760            LHLD  BUFPTR
2F06 09         2770            DAD   B
2F07 3A EC 36   2780            LDA   DURX
2F0A 01 09 00   2790            LXI   B,9
2F0D 09         2800 SETFF      DAD   B
2F0E 72         2810            MOV   M,D      * SET FF DIRECTLY FROM TARGET
2F0F 3D         2820            DCR   A
2F10 C2 0D 2F   2830            JNZ   SETFF
2F13            2840 *
2F13            2850 *  COMPUTE & STORE AN
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2F13            2d60 *
2F13 3E 03      2870            MVI   A,3      * PC=.5 FOR NASAL
2F15 32 F3 36   2880            STA   PC
2F18 3D         2890            DCR   A
2F19 32 F1 36   2900            STA   BT       * BACK & FORW TIMES = 2 FRAMES
2F1C 32 F2 36   2910            STA   FT
2F1F F1         2920            POP   PSW      * GET NASAL TARGET
2F20 E6 03      2930            ANI   3
2F22 87         2940            ADD   A
2F23 87         2950            ADD   A
2F24 87         2960            ADD   A
2F25 87         2970            ADD   A
2F26 87         2980            ADD   A        * MULT BY 32
2F27 01 08 00   2990            LXI   B,CHANAN
2F2A CD C8 30   3000            CALL  GENAX    * SET AMPLITUDE OF NASAL
2F2D C3 7C 30   3010            JMP   ENDPRM
2F30            3020 *
2F30            3030 *
2F30            3040 *  HANDLE -STOP AFTER PLOSIVE
2F30            3050 *  SETASP SETS THE PLOSIVE BURST PULSE
2F30            3060 *  AND IF NEEDED, ASPIRATION IN THE FOLLOWING PHON.
2F30            3070 *
2F30 3A E1 36   3080 SETASP     LDA   OCODE
2F33 D6 27      3090            SUI   CP
2F35 4F         3100            MOV   C,A      * PLOSIVE CODE - CODE OF "P"
2F36 06 00      3110            MVI   B,0
2F38 21 36 32   3120            LXI   H,PLOSC  * MAKE POINTER TO PLOSC TABLE
2F3B 09         3130            DAD   B
2F3C 7E         3140            MOV   A,M      * GET LENGTH OF BURST PULSE
2F3D 47         3150            MOV   B,A
2F3E 87         3160            ADD   A
2F3F 87         3170            ADD   A
2F40 87         3180            ADD   A
2F41 80         3190            ADD   B
2F42 2F         3200            CMA
2F43 3C         3210            INR   A
2F44 4F         3220            MOV   C,A
2F45 78         3230            MOV   A,B      * A = PULSE LENGTH
2F46 06 FF      3240            MVI   B,255    * BC = -9*(PULSE LENGTH)
2F48 2A DF 36   3250            LHLD  BUFPTR
2F4B 09         3260            DAD   B
2F4C 01 C6 00   3270            LXI   B,CHANAF
2F4F 09         3280            DAD   B
2F50 01 09 00   3290            LXI   B,9
2F53 09         3300 BURST      DAD   B
2F54 36 24      3310            MVI   M,36     * PLOSIVE BURST AF VALUE
2F56 3D         3320            DCR   A
2F57 C2 53 2F   3330            JNZ   BURST
2F5A            3340 *
2F5A            3350 *  COMPUTE FORMANTS FOR PHON AFTER PLOSIVE
2F5A            3360 *
2F5A 3A EE 36   3370            LDA   RANKX
2F5D FE 03      3380            CPI   3
2F5F D2 71 2F   3390            JNC   GTPLOS   * CURR RANK > PLOS RANK (=2)
2F62 3E 02      3400            MVI   A,2
2F64 32 F3 36   3410            STA   PC       * CURR RANK <= 2, PC=0.5
2F67 3A E8 36   3420            LDA   OTIMES   * FT = OLD TSEC
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2F6A 0F          3430           RRC
2F6B 0F          3440           RRC
2F6C 0F          3450           RRC
2F6D 0F          3460           RRC
2F6E C3 7A 2F    3470           JMP   SETIMS
2F71             3480 *
2F71 3A EF 36    3490 GTPLOS    LDA   PCTX
2F74 32 F3 36    3500           STA   PC       * PC = PCT OF CURRENT PHON
2F77 3A F0 36    3510           LDA   TIMESX   * FT = CURR. TPRI
2F7A E6 0F       3520 SETIMS    ANI   0FH
2F7C 32 F2 36    3530           STA   FT
2F7F 3E 01       3540           MVI   A,1
2F81 32 F1 36    3550           STA   ET       * BACK TIME = 1 (TO SET BVAL)
2F84             3560 *
2F84             3570 *  GENERATE FORMANTS DURING ENTIRE POST-PLOSIVE
2F84             3580 *
2F84 2A 30 20    3590           LHLD  F1TAR
2F87 EB          3600           XCHG
2F88 01 02 00    3610           LXI   B,CHANF1
2F8B CD AE 30    3620           CALL  GENFX    * SET FORMANT 1
2F8E 2A 32 20    3630           LHLD  F2TAR
2F91 EB          3640           XCHG
2F92 01 03 00    3650           LXI   B,CHANF2
2F95 CD AE 30    3660           CALL  GENFX    * SET FORMANT 2
2F98 2A 34 20    3670           LHLD  F3TAR
2F9B EB          3680           XCHG
2F9C 01 04 00    3690           LXI   B,CHANF3
2F9F CD AE 30    3700           CALL  GENFX    * SET FORMANT 3
2FA2             3710 *
2FA2             3720 *  SET FF = TARGET VALUE FOR TIME DURX
2FA2             3730 *
2FA2 2A 38 20    3740           LHLD  FRNTAR
2FA5 3A E9 36    3750           LDA   CODEX
2FA8 5F          3760           MOV   E,A
2FA9 16 00       3770           MVI   D,0
2FAB 19          3780           DAD   D
2FAC 7E          3790           MOV   A,M      * GET PHON X FRICATIVE TARGET
2FAD E6 E0       3800           ANI   0E0H     * MASK TO KEEP FF
2FAF 57          3810           MOV   D,A
2FB0 2A DF 3o    3820           LHLD  BUFPTR
2FB3 01 07 00    3830           LXI   B,CHANFF
2FB6 09          3840           DAD   B
2FB7 3A EC 36    3850           LDA   DURX
2FBA 01 09 00    3860           LXI   B,9
2FBD 09          3870 SETFF2    DAD   B
2FBE 72          3880           MOV   M,D      * SET FF DIRECTLY FROM TARGET
2FBF 3D          3890           DCR   A
2FC0 C2 BD 2F    3900           JNZ   SETFF2
2FC3             3910 *
2FC3             3920 *  SET AN=0 FOR TIME DURX
2FC3             3930 *
2FC3 2A DF 36    3940           LHLD  BUFPTR
2FC6 11 08 00    3950           LXI   D,CHANAN
2FC9 19          3960           DAD   D
2FCA 3A EC 36    3970           LDA   DURX
2FCD 09          3980 SETAN     DAD   B
2FCE 36 00       3990           MVI   M,0      * SET AN = 0
```

```
ADDR B1 B2 B3 S LINE  LABEL     OPCD  OPERAND

2FD0 3D         4000            DCR   A
2FD1 C2 CD 2F   4010            JNZ   SETAN
2FD4            4020 *
2FD4            4030 *  SEE IF PLOSIVE SHOULD BE ASPIRATED
2FD4            4040 *
2FD4 3A E3 36   4050            LDA   OFEATB
2FD7 E6 10      4060            ANI   PLOSA
2FD9 CA 3A 30   4070            JZ    FINASP  * NO, FINISH CURR. PHON
2FDC 3A EC 36   4080            LDA   DURX
2FDF 16 05      4090            MVI   D,5     * BASE ASP TIME = 50 MSEC
2FE1 BA         4100            CMP   D
2FE2 DA E6 2F   4110            JC    $+4     * DURX < 50 MSEC, SET ASPT=DURX
2FE5 7A         4120            MOV   A,D     * THERE'S ROOM, SET ASPT=50 MSEC
2FE6 32 FC 36   4130            STA   ASPT
2FE9            4140 *
2FE9            4150 *  SET AV = 0 FOR DURATION OF ASPIRATION (ASPT)
2FE9            4160 *
2FE9 2A DF 36   4170            LHLD  BUFPTR
2FEC 11 00 00   4180            LXI   D,CHANAV
2FEF 19         4190            DAD   D
2FF0 09         4200 SETAV      DAD   B
2FF1 36 00      4210            MVI   M,0     * SET AV = 0
2FF3 3D         4220            DCR   A
2FF4 C2 F0 2F   4230            JNZ   SETAV
2FF7            4240 *
2FF7            4250 *  SET AF = 0 DURING ASPIRATION
2FF7            4260 *
2FF7 3A FC 36   4270            LDA   ASPT
2FFA 2A DF 36   4280            LHLD  BUFPTR
2FFD 11 06 00   4290            LXI   D,CHANAF
3000 19         4300            DAD   D
3001 09         4310 SETAF      DAD   B
3002 36 00      4320            MVI   M,0     * SET AF = 0
3004 3D         4330            DCR   A
3005 C2 01 30   4340            JNZ   SETAF
3008            4350 *
3008            4360 *  CALL FRWRD TO PRODUCE TRANSIENT PULSE ON AH
3008            4370 *
3008 AF         4380            XRA   A
3009 32 F4 36   4390            STA   TARG
300C 3A FC 36   4400            LDA   ASPT
300F 47         4410            MOV   B,A     * DECAY TO 0 BY END OF ASP TIME
3010 4F         4420            MOV   C,A
3011 2A DF 36   4430            LHLD  BUFPTR
3014 11 05 00   4440            LXI   D,CHANAH
3017 19         4450            DAD   D
3018 36 32      4460            MVI   M,50    * INITIAL AH PULSE AMPLITUDE
301A CD E7 30   4470            CALL  FRWRD
301D            4480 *
301D            4490 *  SEE IF ASPIRATION TIME TOOK THE ENTIRE CURR. PHON
301D            4500 *
301D 3A FC 36   4510            LDA   ASPT
3020 47         4520            MOV   B,A
3021 3A EC 36   4530            LDA   DURX
3024 90         4540            SUB   B
3025 CA 7C 30   4550            JZ    ENDPRM  * ASP TIME TOOK IT ALL, DONE
3028 32 EC 36   4560            STA   DURX    * SET DURX=DURX-ASPT
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

302B 78         4570            MOV   A,E
302C 87         4580            ADD   A
302D 87         4590            ADD   A
302E 87         4600            ADD   A
302F 80         4610            ADD   B
3030 4F         4620            MOV   C,A       * ASPT*9 INTO BC
3031 06 00      4630            MVI   B,0
3033 2A DF 36   4640            LHLD  BUFPTR
3036 09         4650            DAD   B
3037 22 DF 36   4660            SHLD  BUFPTR    * BUFPTR=BUFPTR+9*ASPT
303A            4670 *
303A            4680 *  COMPUTE AV FOR REMAINDER OF PHON
303A            4690 *
303A 06 00      4700 FINASP     MVI   B,0
303C CD 2D 20   4710            CALL  PCBFT     * GET AV/AH VALUES FOR PC & FT
303F 3E 01      4720            MVI   A,1
3041 32 F1 36   4730            STA   BT        * BT=1 SETS BVAL FROM TARG
3044 2A 36 20   4740            LHLD  AVHTAR
3047 3A E9 36   4750            LDA   CODEX
304A 16 00      4760            MVI   D,0
304C 5F         4770            MOV   E,A
304D 19         4780            DAD   D
304E D5         4790            PUSH  D         * SAVE CODEX TABLE OFFSET
304F 7E         4800            MOV   A,M       * GET AV/AH TARGET BYTE
3050 F5         4810            PUSH  PSW       * SAVE IT
3051 E6 F0      4820            ANI   0F0H      * MASK TO KEEP AV
3053 01 00 00   4830            LXI   B,CHANAV
3056 CD C8 30   4840            CALL  GENAX     * FINISH AV CURVE
3059            4850 *
3059            4860 *  COMPUTE AH FOR REMAINDER OF PHON
3059            4870 *
3059 F1         4880            POP   PSW
305A E6 0F      4890            ANI   0FH       * MASK TARGET BYTE TO KEEP AH
305C 87         4900            ADD   A
305D 87         4910            ADD   A         * MULT BY 4
305E 01 05 00   4920            LXI   B,CHANAH
3061 CD C8 30   4930            CALL  GENAX     * FINISH AH CURVE
3064            4940 *
3064            4950 *  COMPUTE AF FOR REMAINDER OF PHON
3064            4960 *
3064 3E 03      4970            MVI   A,3
3066 32 F2 36   4980            STA   FT        * FT = 3
3069 3C         4990            INR   A
306A 32 F3 36   5000            STA   PC        * PC = 1.00
306D 2A 38 20   5010            LHLD  FRNTAR
3070 D1         5020            POP   D
3071 19         5030            DAD   D
3072 7E         5040            MOV   A,M       * GET FRIC/NASAL TARGET BYTE
3073 E6 1C      5050            ANI   1CH       * MASK TO KEEP AF
3075 87         5060            ADD   A
3076 01 06 00   5070            LXI   B,CHANAF
3079 CD C8 30   5080            CALL  GENAX     * FINISH AF CURVE
307C            5090 *
307C            5100 *
307C 00         5110 ENDPRM     NOP   *         * BREAKPOINT LOC
307D 2A EC 36   5120            LHLD  DURX
3080 26 00      5130            MVI   H,0
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3082 54         5140            MOV   D,H
3083 5D         5150            MOV   E,L      * MULTIPLY DURX BY 9
3084 29         5160            DAD   H
3085 29         5170            DAD   H
3086 29         5180            DAD   H
3087 19         5190            DAD   D
3088 EB         5200            XCHG
3089 2A DF 36   5210            LHLD  BUFPTR
308C 19         5220            DAD   D
308D 22 DF 36   5230            SHLD  BUFPTR   * BUFPTR=BUFPTR+9*DURX
3090 21 E9 36   5240            LXI   H,CODEX
3093 11 E1 36   5250            LXI   D,OCODE
3096 0E 08      5260            MVI   C,8
3098 7E         5270 CNSLP      MOV   A,M      * MOVE CURR CONSTANT VALUES
3099 12         5280            STAX  D        * INTO OLD CONSTANTS
309A 23         5290            INX   H
309B 13         5300            INX   D
309C 0D         5310            DCR   C
309D C2 98 30   5320            JNZ   CNSLP
30A0 2A 00 35   5330            LHLD  MATPTR
30A3 23         5340            INX   H
30A4 22 00 35   5350            SHLD  MATPTR
30A7 7E         5360            MOV   A,M
30A8 FE 04      5370            CPI   CTERM    * SEE IF NEW CODE IS TERMINATOR
30AA C2 22 2E   5380            JNZ   GPLOOP   * NO, DO ANOTHER COLUMN
30AD C9         5390            RET   *        * THAT'S IT, LETS GO LISTEN
30AE            5400 *
30AE            5410 *
30AE            5420 **********
30AE            5430 *
30AE            5440 *  GENPRM SUBROUTINES AND SOME TABLES
30AE            5450 *
30AE            5460 **********
30AE            5470 *
30AE            5480 *  GENERATE FORMANT TRANSITIONS BETWEEN PREVIOUS
30AE            5490 *  AND CURRENT PHONS
30AE            5500 *  ON ENTRY:  DE POINTS TO FORMANT TARGET TABLE
30AE            5510 *             BC CONTAINS THE OUTPUT CHAN NO OF THAT
30AE            5520 *                FORMANT (TELLS WHICH BYTE OF FRAME)
30AE            5530 *
30AE 3A E9 36   5540 GENFX      LDA   CODEX
30B1 6F         5550            MOV   L,A
30B2 26 00      5560            MVI   H,0
30B4 19         5570            DAD   D
30B5 7E         5580            MOV   A,M
30B6 32 F4 36   5590            STA   TARG     * SET TARGET VALUE
30B9 2A DF 36   5600            LHLD  BUFPTR
30BC 09         5610            DAD   B        * SET HL TO FORMANT, FRAME I
30BD 96         5620            SUB   M
30BE CD D2 31   5630            CALL  MULPC    * MULT TARG-FX(I) BY PC
30C1 86         5640            ADD   M
30C2 32 F5 36   5650            STA   BVAL     * BVAL=FX(I)+PC*(TARG-FX(I))
30C5 C3 D8 30   5660            JMP   GENBF
30C8            5670 *
30C8            5680 *  GENAX SUBROUTINE
30C8            5690 *  GENERATE AMPLITUDE TRANSITIONS BETWEEN PREV
30C8            5700 *  AND CURRENT PHONS
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

30C8            5710 *  ON ENTRY:  A CONTAINS THE PARAM TARGET VALUE
30C8            5720 *             BC CONTAINS THE OUTPUT CHAN NO.
30C8            5730 *
30C8 32 F4 36   5740 GENAX      STA   TARG     * SET TARGET VALUE
30CB 2A DF 36   5750            LHLD  BUFPTR
30CE 09         5760            DAD   B        * SET HL TO PARAM IN FRAME 1
30CF 86         5770            ADD   M
30D0 1F         5780            RAR
30D1 B7         5790            ORA   A        * DIVIDE BY 2, CLEAR CARRY
30D2 CD D2 31   5800            CALL  MULPC
30D5 32 F5 36   5810            STA   BVAL     * BVAL=PC*(TARG+Y)/2
30D8 3A F1 36   5820 GENBF      LDA   BT
30DB 47         5830            MOV   B,A
30DC CD 0B 31   5840            CALL  BCWRD    * MODIFY CURVE BACK FROM BOUNDARY
30DF 3A EC 36   5850            LDA   DURX
30E2 4F         5860            MOV   C,A
30E3 3A F2 36   5870            LDA   FT
30E6 47         5880            MOV   B,A
30E7            5890 *
30E7            5900 *  FRWRD
30E7            5910 *  STORES THE NEXT N FRAMES BEYOND THE PARAM VALUE
30E7            5920 *  CURRENTLY INDICATED BY HL.  B CONTAINS THE NUMBER
30E7            5930 *  OF FRAMES NEEDED TO REACH THE VALUE IN LOC TARG.
30E7            5940 *  C CONTAINS THE TOTAL NO. OF NEW FRAMES TO BE
30E7            5950 *  STORED.  B MAY BE ANY SIZE RELATIVE TO C
30E7            5960 *
30E7 AF         5970 FRWRD      XRA   A
30E8 B9         5980            CMP   C        * RETURN IF NOTHING TO STORE
30E9 C8         5990            RZ
30EA 3C         6000            INR   A
30EB B8         6010            CMP   B
30EC D2 5E 31   6020            JNC   SETARG   * B<=1, SET PARAM=TARG
30EF 79         6030            MOV   A,C
30F0 90         6040            SUB   B
30F1 F2 F8 30   6050            JP    FINPB    * POS # OF TARGS AFTER PARABOLA
30F4 AF         6060            XRA   A
30F5 C3 FB 30   6070            JMP   DOPB
30F8 48         6080 FINPB      MOV   C,B
30F9 0D         6090            DCR   C
30FA 3C         6100            INR   A
30FB 32 FB 36   6110 DOPB       STA   NTARGS   * SET FOR N TARGS AFTER PARAB.
30FE 79         6120            MOV   A,C
30FF 32 FA 36   6130            STA   PCOUNT   * SET # OF PARABOLA POINTS
3102 4E         6140            MOV   C,M      * GET PREV PARAM INTO C
3103 E5         6150            PUSH  H
3104 3A F4 36   6160            LDA   TARG
3107 91         6170            SUB   C
3108 F5         6180            PUSH  PSW
3109 D2 0E 31   6190            JNC   $+5
310C 2F         6200            CMA
310D 3C         6210            INR   A
310E 57         6220            MOV   D,A      * ABS(TARG-PREV) INTO D
310F 68         6230            MOV   L,B
3110 2C         6240            INR   L
3111 AF         6250            XRA   A
3112 67         6260            MOV   H,A      * HL=N+1 (N IS DIST TO PEAK)
3113 5F         6270            MOV   E,A
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND
3114 CD 13 32    6260          CALL  DIV     * ABS(256*(TARG-PREV)/(N+1))
3117 29          6290          DAD   H
3118 54          6300          MOV   D,H     * SAVE ASB(DELTA)=2*QUOTIENT
3119 5D          6310          MOV   E,L
311A F1          6320          POP   PSW
311B D2 26 31    6330          JNC   FRW2
311E 7C          6340          MOV   A,H
311F 2F          6350          CMA
3120 67          6360          MOV   H,A
3121 7D          6370          MOV   A,L
3122 2F          6380          CMA
3123 6F          6390          MOV   L,A
3124 23          6400          INX   H       * FIX SIGN OF DELTA
3125 37          6410          STC
3126 E5          6420 FRW2     PUSH  H       * AND SAVE IT
3127 F5          6430          PUSH  PSW     * AND ITS SIGN
3128 68          6440          MOV   L,B
3129 26 00       6450          MVI   H,0
312B CD 13 32    6460          CALL  DIV     * COMPUTE ABS(DELTA/N)
312E F1          6470          POP   PSW
312F DA 39 31    6480          JC    FRW3
3132 7C          6490          MOV   A,H
3133 2F          6500          CMA
3134 67          6510          MOV   H,A
3135 7D          6520          MOV   A,L
3136 2F          6530          CMA
3137 6F          6540          MOV   L,A
3138 23          6550          INX   H
3139 22 F6 36    6560 FRW3     SHLD  SD      * SECOND DIFF = -DELTA/N
313C E1          6570          POP   H       * GET DELTA INTO HL
313D 51          6580          MOV   D,C     * PREV PARAM TO DE
313E 1E 00       6590          MVI   E,0
3140 3A FA 36    6600          LDA   PCOUNT  * AND PARAB COUNT TO A
3143 44          6610 FWDLP    MOV   B,H
3144 4D          6620          MOV   C,L
3145 19          6630          DAD   D
3146 D1          6640          POP   D
3147 E5          6650          PUSH  H       * Y=Y+DELTA
3148 21 09 00    6660          LXI   H,9
314B 19          6670          DAD   D
314C D1          6680          POP   D
314D E5          6690          PUSH  H       * ADDR=ADDR+9
314E 72          6700          MOV   M,D     * STORE NEW Y VALUE
314F 2A F6 36    6710          LHLD  SD
3152 09          6720          DAD   B       * DELTA=DELTA+SD
3153 3D          6730          DCR   A
3154 C2 43 31    6740          JNZ   FWDLP
3157 E1          6750          POP   H       * GET ADDR POINTER
3158 3A FB 36    6760          LDA   NTARGS  * NO. OF TARG VALUES TO WRITE
315B B7          6770          ORA   A
315C C8          6780          RZ
315D 4F          6790          MOV   C,A
315E 3A F4 36    6800 SETARG   LDA   TARG
3161 11 09 00    6810          LXI   D,9
3164 19          6820 SETAL    DAD   D
3165 77          6830          MOV   M,A     * SET PARAM=TARG
3166 0D          6840          DCR   C
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3167 C2 64 31   6850            JNZ   SETAL
316A C9         6860            RET
316B            6870 *
316B            6880 *  BCWRD
316B            6890 *  MODIFIES N FRAMES BACKWARD FROM AND INCLUDING
316B            6900 *  FRAME I, CURRENTLY INDICATED BY HL.
316B            6910 *  B CONTAINS THE NUMBER N
316B            6920 *  LOCATION BVAL CONTAINS THE FINAL DESIRED VALUE
316B            6930 *  FOR FRAME I (THE LAST ONE MODIFIED).
316B            6940 *  ON RETURN, HL IS LEFT POINTING TO THE PARAM, FRAME I
316B            6950 *
316B 78         6960 BCWRD      MOV   A,B       * CHECK N
316C FE 02      6970            CPI   2
316E D8         6980            RC
316F CA CD 31   6990            JZ    SETIB     * MODIFY ONLY THE LAST FRAME
3172 87         7000            ADD   A
3173 87         7010            ADD   A
3174 87         7020            ADD   A
3175 80         7030            ADD   B
3176 2F         7040            CMA
3177 5F         7050            MOV   E,A
3178 16 FF      7060            MVI   D,255
317A 13         7070            INX   D         * DE = -9*N
317B 19         7080            DAD   D
317C 56         7090            MOV   D,M       * GET YC, VALUE AT FRAME I-N
317D E5         7100            PUSH  H
317E D5         7110            PUSH  D
317F 3A F5 36   7120            LDA   BVAL
3182 92         7130            SUB   D
3183 F5         7140            PUSH  PSW       * PUT SIGN(BVAL-YC) IN STACK
3184 D2 89 31   7150            JNC   $+5
3187 2F         7160            CMA
3188 3C         7170            INR   A
3189 6F         7180            MOV   L,A
318A 26 00      7190            MVI   H,0
318C 29         7200            DAD   H         * 2*ABS(BVAL-YC), SIGN IN STK
318D 29         7210            DAD   H         *   ... *16
318E 29         7220            DAD   H
318F 29         7230            DAD   H
3190 29         7240            DAD   H
3191 58         7250            MOV   E,B
3192 1C         7260            INR   E
3193 CD 00 32   7270            CALL  MUL       * COMPUTE (DE)=N*(N+1)
3196 EB         7280            XCHG
3197 CD 13 32   7290            CALL  DIV       * COMPUTE 16*SECOND DIFF
319A F1         7300            POP   PSW
319B D2 A5 31   7310            JNC   BCW3
319E 7C         7320            MOV   A,H
319F 2F         7330            CMA
31A0 67         7340            MOV   H,A
31A1 7D         7350            MOV   A,L
31A2 2F         7360            CMA
31A3 6F         7370            MOV   L,A
31A4 23         7380            INX   H
31A5 29         7390 BCW3       DAD   H
31A6 29         7400            DAD   H
31A7 29         7410            DAD   H
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND

31A8 29         7420           DAD   H
31A9 22 F6 36   7430           SHLD  SD       * SD=256*SECOND DIFF
31AC D1         7440           POP   D        * GET YC VALUE BACK
31AD 1E 00      7450           MVI   E,0      * (DE)=256*YC
31AF 78         7460           MOV   A,B      * LOOP COUNT IS N-1
31B0 3D         7470           DCR   A
31B1 01 00 00   7480           LXI   B,0      * INIT DELTA=0
31B4 2A F6 36   7490 BCWLP     LHLD  SD
31B7 09         7500           DAD   B
31B8 44         7510           MOV   B,H      * DELTA=DELTA+SD INTO BC
31B9 4D         7520           MOV   C,L
31BA 19         7530           DAD   D        * Y=Y+DELTA
31BB D1         7540           POP   D
31BC E5         7550           PUSH  H
31BD 21 09 00   7560           LXI   H,9
31C0 19         7570           DAD   D        * NEW ADDR=ADDR+9
31C1 D1         7580           POP   D
31C2 E5         7590           PUSH  H
31C3 72         7600           MOV   M,D
31C4 3D         7610           DCR   A
31C5 C2 B4 31   7620           JNZ   BCWLP
31C8 D1         7630           POP   D        * CLEAN UP THE STACK
31C9 11 09 00   7640           LXI   D,9
31CC 19         7650           DAD   D
31CD 3A F5 36   7660 SETIB     LDA   BVAL     * PUT BVAL IN LAST FRAME
31D0 77         7670           MOV   M,A
31D1 C9         7680           RET
31D2            7690 *
31D2            7700 *  MULPC
31D2            7710 *  MULTIPLY ABS(A) BY PC, SIGN(A) IS IN CARRY
31D2            7720 *  POSSIBLE VALUES OF PC ARE 0,1/4,1/2,3/4, & 1
31D2            7730 *  STORED WITH THE BIN PT TO THE RIGHT OF BIT 2
31D2            7740 *  RETURN RESULT IN A
31D2            7750 *    (DE) LOST, (BC) & (HL) RESTORED
31D2            7760 *
31D2 E5         7770 MULPC     PUSH  H
31D3 1E 00      7780           MVI   E,0
31D5 D2 DB 31   7790           JNC   MPC1     * IF CARRY SET, ...
31D8 2F         7800           CMA
31D9 3C         7810           INR   A        * COMPL A, SAVE CARRY IN E
31DA 1D         7820           DCR   E
31DB 6F         7830 MPC1      MOV   L,A
31DC 3A F3 36   7840           LDA   PC
31DF 1F         7850           RAR
31E0 67         7860           MOV   H,A
31E1 3E 00      7870           MVI   A,0
31E3 D2 E8 31   7880           JNC   MPC2
31E6 85         7890           ADD   L
31E7 1F         7900           RAR
31E8 57         7910 MPC2      MOV   D,A
31E9 7C         7920           MOV   A,H
31EA 1F         7930           RAR
31EB 67         7940           MOV   H,A
31EC 7A         7950           MOV   A,D
31ED D2 F2 31   7960           JNC   MPC3
31F0 85         7970           ADD   L
31F1 1F         7980           RAR
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

31F2 57          7990 MPC3      MOV   D,A
31F3 7C          8000           MOV   A,H
31F4 1F          8010           RAR
31F5 7A          8020           MOV   A,D
31F6 D2 FA 31    8030           JNC   MPC4
31F9 85          8040           ADD   L
31FA E1          8050 MPC4      POP   H
31FB 1C          8060           INR   E
31FC C0          8070           RNZ
31FD 2F          8080           CMA
31FE 3C          8090           INR   A        * RESTORE SIGN OF A
31FF C9          8100           RET
3200             8110 *
3200             8120 *  MUL SUBROUTINE
3200             8130 *  MULTIPLY (B) BY (E), 16 BIT RESULT IN DE
3200             8140 *    SEE INTEL ASSY LANG PROG MAN., PAGE 54
3200             8150 *    HL NOT DISTURBED
3200             8160 *
3200 16 00       8170 MUL       MVI   D,0
3202 0E 09       8180           MVI   C,9
3204 7B          8190 MUL2      MOV   A,E
3205 1F          8200           RAR
3206 5F          8210           MOV   E,A
3207 0D          8220           DCR   C
3208 C8          8230           RZ
3209 7A          8240           MOV   A,D
320A D2 0E 32    8250           JNC   MUL3
320D 82          8260           ADD   B
320E 1F          8270 MUL3      RAR
320F 57          8280           MOV   D,A
3210 C3 04 32    8290           JMP   MUL2
3213             8300 *
3213             8310 *  DIV SUBROUTINE
3213             8320 *  DIVIDE (DE)/(HL), RRESULT IN HL, REM IN DE
3213             8330 *    BC RESTORED, A DESTROYED
3213             8340 *
3213 C5          8350 DIV       PUSH  B
3214 44          8360           MOV   B,H
3215 4D          8370           MOV   C,L
3216 21 00 00    8380           LXI   H,0
3219 3E 10       8390           MVI   A,16
321B F5          8400 DIV1      PUSH  PSW
321C 29          8410           DAD   H
321D EB          8420           XCHG
321E 29          8430           DAD   H
321F EB          8440           XCHG
3220 D2 24 32    8450           JNC   DIV2
3223 23          8460           INX   H
3224 7D          8470 DIV2      MOV   A,L
3225 91          8480           SUB   C
3226 6F          8490           MOV   L,A
3227 7C          8500           MOV   A,H
3228 98          8510           SBB   B
3229 67          8520           MOV   H,A
322A 13          8530           INX   D
322B D2 30 32    8540           JNC   DIV3
322E 09          8550           DAD   B
```

```
ADDR B1 B2 B3 F LINE  LABEL     OPCD  OPERAND

322F 1B          8560           DCX   D
3230 F1          8570 DIV3      POP   PSW
3231 3D          8580           DCR   A
3232 C2 1B 32    8590           JNZ   DIV1
3235 C1          8600           POP   B
3236 EB          8610           XCHG
3237 C9          8620           RET
3238             8630 *
3238             8640 *  PLOSC TABLE
3238             8650 *  PLOSIVE BURST PULSE DURATIONS
3238             8660 *
3238 01          8670 PLOSC     DB    1    * P
3239 01          8680           DB    1    * T
323A 02          8690           DB    2    * K
323B 02          8700           DB    2    * KX
323C 01          8710           DB    1    * B
323D 01          8720           DB    1    * D
323E 02          8730           DB    2    * G
323F 02          8740           DB    2    * GX
3240             8750 *
3240             8760 *  END OF SECT 5
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

2000            0010 *  PLAY ROUTINE, TARGET TABLES, & MISC
2000            0020 *
2000            0030 *  SECTION 6 OF THE CSR1 SYNTHESIS BY RULE SYSTEM
2000            0040 *
2000            0050 *  LLOYD RICE,  COMPUTALKER CONSULTANTS
2000            0060 *  VERSION 1.08   MAY 30, 1977
2000            0070 *
2000            0080 **********
2000            0090 *
2000            0100 *  COMMON JUMP ADDRESS TABLE
2000            0110 *
2000            0120 COMJMP     EQU   $
2000            0130 *
2000            0140 CSR1       DS    3
2003 C3 50 32   0150            JMP   PLAY
2006            0160 BUFADR     DS    2
2008            0170 BUFEND     DS    2
200A            0180 PVTAB      DS    2
200C            0190 MATPAK     DS    3
200F            0200 MATERR     DS    3
2012            0210 RULES      DS    3
2015            0220 SETDUR     DS    3
2018            0230 RULES3     DS    3
201B            0240 GENF0      DS    3
201E            0250 CLRBUF     DS    3
2021            0260 GENPRM     DS    3
2024            0270 MUL        DS    3
2027            0280 DIV        DS    3
202A C3 96 32   0290            JMP   GETCNS
202D C3 D0 32   0300            JMP   PCBFT
2030 1C 33      0310            DW    F1TAR
2032 58 33      0320            DW    F2TAR
2034 94 33      0330            DW    F3TAR
2036 D0 33      0340            DW    AVHTAR
2038 0C 34      0350            DW    FRNTAR
203A            0360 *
203A            0370 **********
203A            0380 *
203A            0390 *  COMRAM ORIGEN DEFINITION
203A            0400 *
203A            0410            ORG   COMJMP+1500H
3500            0420 COMRAM     EQU   $
3500            0430 *
3500            0440 *  CSR1 SYSTEM RAM SPACE DEFINITION
3500            0450 *
3500            0460 MATPTR     DS    2
3502            0470 NEGEND     DS    2
3504            0480 MATRIX     EQU   $
3504            0490 MATLEN     EQU   95
3504            0500 PHCODE     DS    MATLEN
3563            0510 FEATA      DS    MATLEN
35C2            0520 FEATB      DS    MATLEN
3621            0530 STRES      DS    MATLEN
3680            0540 DUR        DS    MATLEN
36DF            0550 MATEND     EQU   $
36DF            0560 BUFPTR     DS    2
36E1            0570 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

36E1            0580 *  THE NEXT 19 LOCATIONS ARE USED IN COMMON
36E1            0590 *  WITH SECTION 5.  THEY MUST NOT BE MOVED
36E1            0600 *
36E1            0610 OCODE      DS    1
36E2            0620 OFEATA     DS    1
36E3            0630 OFEATB     DS    1
36E4            0640 ODUR       DS.   1
36E5            0650 OCID       DS    1
36E6            0660 ORANK      DS    1
36E7            0670 OPCT       DS    1
36E8            0680 OTIMES     DS    1
36E9            0690 *
36E9            0700 CODEX      DS    1
36EA            0710 FEATAX     DS    1
36EB            0720 FEATBX     DS    1
36EC            0730 DURX       DS    1
36ED            0740 CIDX       DS    1
36EE            0750 RANKX      DS    1
36EF            0760 PCTX       DS    1
36F0            0770 TIMESX     DS    1
36F1            0780 *
36F1            0790 BT         DS    1
36F2            0800 FT         DS    1
36F3            0810 PC         DS    1
36F4            0820 *
36F4            0830 **********
36F4            0840 *
36F4            0850 *  PLAY (LOCAL) RAM WORKSPACE
36F4            0860 *
36F4            0870 CTO        DS    3       * SPACE FOR CT-1 OUTPUT CODE
36F7            0880 *
36F7            0890 **********
36F7            0900 *
36F7            0910            ORG   COMJMP+1250H
3250            0920 SECTAD     EQU   $
3250            0930 *
3250            0940 *
3250            0950 *  DEFINITIONS FOR PLAY SUBROUTINE
3250            0960 *
3250            0970 CTBASE     EQU   0E0H
3250            0980 CHANSW     EQU   15
3250            0990 *
3250            1000 **********
3250            1010 *
3250            1020 *  PLAYBACK SUBROUTINE
3250            1030 *
3250 21 F4 36   1040 PLAY       LXI   H,CTO    * SETUP CT-1 OUTPUT SUBROUTINE
3253 36 D3      1050            MVI   M,0D3H
3255 23         1060            INX   H
3256 23         1070            INX   H
3257 36 C9      1080            MVI   M,0C9H   * SETUP RETURN
3259 2A 06 20   1090            LHLD  BUFADR
325C 5E         1100            MOV   E,M      * GET FRAME COUNT (LOW BYTE)
325D 23         1110            INX   H
325E 56         1120            MOV   D,M      *  (AND HIGH BYTE)
325F 23         1130            INX   H
3260 CD 77 32   1140            CALL  CTOUT    * SET CT-1 PARAMS FROM FRAME 1
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3263 1B         1150            DCX   D        * COUNT THAT FRAME
3264 3E FF      1160            MVI   A,255
3266 D3 EF      1170            OUT   CTBASE+CHANSW * TURN ON CT-1
3268 CD 77 32   1180 PLALP      CALL  CTOUT    * PLAY THE BUFFER
326B CD 8A 32   1190            CALL  DLY10    * WAIT 10 MSEC
326E 1B         1200            DCX   D
326F 7A         1210            MOV   A,D
3270 B3         1220            ORA   E
3271 C2 68 32   1230            JNZ   PLALP    * LOOP UNTIL BUFFER DONE
3274 D3 EF      1240            OUT   CTBASE+CHANSW * TURN OFF CT-1
3276 C9         1250            RET
3277            1260 *
3277            1270 *  CTOUT PLAYS ONE DATA FRAME FROM THE BUFFER
3277            1280 *  ON ENTRY: HL POINTS TO AV OF THE FRAME TO PLAY
3277            1290 *              SUBR CTO HAS BEEN SET UP AS:
3277            1300 *                     CTO  OUT CTBASE
3277            1310 *                          RET
3277            1320 *  (DE) ARE UNCHANGED
3277            1330 *
3277 06 E0      1340 CTOUT      MVI   B,CTBASE * RE-INITIALIZE CTO ROUTINE
3279 0E 09      1350            MVI   C,9
327B 78         1360 CTLP       MOV   A,B
327C 32 F5 36   1370            STA   CTO+1
327F 7E         1380            MOV   A,M
3280 CD F4 36   1390            CALL  CTO      * OUTPUT THE PARAMETER
3283 23         1400            INX   H
3284 04         1410            INR   B
3285 0D         1420            DCR   C
3286 C2 7B 32   1430            JNZ   CTLP     * GO AROUND 9 TIMES
3289 C9         1440            RET
328A            1450 *
328A            1460 *  DELAY 10 MILLISECONDS (ASSUMES 2MHZ CLOCK)
328A            1470 *  (A) CHANGED, ALL ELSE RESTORED
328A            1480 *
328A E5         1490 DLY10      PUSH  H
328B 21 20 03   1500            LXI   H,800
328E 2B         1510            DCX   H
328F 7C         1520            MOV   A,H
3290 B5         1530            ORA   L
3291 C2 8E 32   1540            JNZ   $-3
3294 E1         1550            POP   H
3295 C9         1560            RET
3296            1570 *
3296            1580 **********
3296            1590 *
3296            1600 *  GETCNS SUBROUTINE
3296            1610 *  GET MISC CONSTANTS PERTAINING TO THE CURRENT PHON
3296            1620 *  PUT DATA INTO TABLE BEGINNING AT (DE)
3296            1630 *     ALL REGISTERS MODIFIED
3296            1640 *
3296 2A 00 35   1650 GETCNS      LHLD  MATPTR
3299 01 5F 00   1660            LXI   B,MATLEN
329C 7E         1670            MOV   A,M      * GET CURRENT PHON CODE
329D 12         1680            STAX  D
329E 09         1690            DAD   B
329F 13         1700            INX   D
32A0 7E         1710            MOV   A,M      * GET FEATA
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

32A1 12          1720           STAX  D
32A2 09          1730           DAD   B
32A3 13          1740           INX   D
32A4 7E          1750           MOV   A,M       * GET FEATB
32A5 12          1760           STAX  D
32A6 09          1770           DAD   B
32A7 09          1780           DAD   B
32A8 13          1790           INX   D
32A9 7E          1800           MOV   A,M       * GET DURATION
32AA 12          1810           STAX  D
32AB 13          1820           INX   D
32AC 2A 00 35    1830           LHLD  MATP1R    * GET CODE AGAIN
32AF 4E          1840           MOV   C,M
32B0 06 00       1850           MVI   B,0
32B2 21 4D 34    1860           LXI   H,RAPCID
32B5 09          1870           DAD   B
32B6 7E          1880           MOV   A,M       * GET RANK/PC/ID BYTE
32B7 6F          1890           MOV   L,A
32B8 E6 07       1900           ANI   7
32BA 12          1910           STAX  D         * MASK & STORE CID
32BB 13          1920           INX   D
32BC 60          1930           MOV   H,B
32BD 29          1940           DAD   H
32BE 29          1950           DAD   H
32BF 29          1960           DAD   H
32C0 7C          1970           MOV   A,H
32C1 12          1980           STAX  D         * STORE RANK
32C2 13          1990           INX   D
32C3 60          2000           MOV   H,B
32C4 29          2010           DAD   H
32C5 29          2020           DAD   H
32C6 7C          2030           MOV   A,H
32C7 12          2040           STAX  D         * SHIFT & STORE PERCENT
32C8 13          2050           INX   D
32C9 21 84 34    2060           LXI   H,TTIMES
32CC 09          2070           DAD   B
32CD 7E          2080           MOV   A,M       * GET TRANSITION TIMES BYTE
32CE 12          2090           STAX  D
32CF C9          2100           RET
32D0             2110 *
32D0             2120 *  PCBFT SUBROUTINE
32D0             2130 *  GET PERCENT AND TIME VALUES FROM MATRIX ARRAYS
32D0             2140 *    SET PC=PCVF(CIDX,OCID)
32D0             2150 *    SET BT=TAVF(OCID)
32D0             2160 *    SET PT=TAVF(CIDX)
32D0             2170 *  ON ENTRY: B=0 TO GET AV/AH INFO
32D0             2180 *              B#0 TO GET AF INFO
32D0             2190 *
32D0 3A E5 36    2200 PCBFT      LDA   OCID
32D3 4F          2210           MOV   C,A
32D4 87          2220           ADD   A
32D5 81          2230           ADD   C
32D6 87          2240           ADD   A
32D7 81          2250           ADD   C
32D8 57          2260           MOV   D,A
32D9 3A ED 36    2270           LDA   CIDX
32DC 82          2280           ADD   D
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

32DD 5F         2290            MOV   E,A
32DE 16 00      2300            MVI   D,0       * PCVF INDEX IS 7*OCID+CIDX
32E0 21 C0 34   2310            LXI   H,PCVF
32E3 19         2320            DAD   D
32E4 AF         2330            XRA   A
32E5 B0         2340            ORA   B
32E6 7E         2350            MOV   A,M
32E7 CA ED 32   2360            JZ    $+6       * KEEP AV/AH PC VALUE
32EA 1F         2370            RAR
32EB 1F         2380            RAR
32EC 1F         2390            RAR   *         * SHIFT OVER TO AF PC VALUE
32ED E6 07      2400            ANI   7
32EF 32 F3 36   2410            STA   PC        * STORE PERCENT VALUE
32F2 59         2420            MOV   E,C       * USE OLD CID AS TIME TABLE INDEX
32F3 21 F1 34   2430            LXI   H,TAVF
32F6 EB         2440            XCHG
32F7 19         2450            DAD   D
32F8 AF         2460            XRA   A
32F9 B0         2470            ORA   B
32FA 7E         2480            MOV   A,M
32FB CA 01 33   2490            JZ    $+6       * KEEP AV TIME CONSTANTS
32FE 1F         2500            RAR
32FF 1F         2510            RAR
3300 1F         2520            RAR   *         * SHIFT OVER TO AF CONSTANTS
3301 E6 07      2530            ANI   7
3303 32 F1 36   2540            STA   BT        * STORE BACKWARD TIME CONSTANT
3306 3A ED 36   2550            LDA   CIDX
3309 6F         2560            MOV   L,A
330A 26 00      2570            MVI   H,0       * USE NEW CID TO INDEX TAVF
330C 19         2580            DAD   D
330D AF         2590            XRA   A
330E B0         2600            ORA   B
330F 7E         2610            MOV   A,M
3310 CA 16 33   2620            JZ    $+6
3313 1F         2630            RAR
3314 1F         2640            RAR
3315 1F         2650            RAR
3316 E6 07      2660            ANI   7
3318 32 F2 36   2670            STA   FT        * STORE FORWARD TIME CONST.
331B C9         2680            RET
331C            2690 *
331C            2700 *  F1 TARGET TABLE
331C            2710 *
331C 00         2720 F1TAR      DB    0         * SPACE
331D 00         2730            DB    0         * PERIOD
331E 80         2740            DB    128       * COMMA (PAUSE)
331F 00         2750            DB    0         * QUEST
3320 00         2760            DB    0         * # (TERMINATOR)
3321 B6         2770            DB    182       * IY
3322 93         2780            DB    147       * IH
3323 71         2790            DB    113       * EH
3324 58         2800            DB    88        * AE
3325 55         2810            DB    85        * AA
3326 67         2820            DB    103       * AH
3327 6D         2830            DB    109       * AO
3328 77         2840            DB    119       * OW
3329 95         2850            DB    149       * UH
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

332A A8         2860            DB    168   * UW
332B 48         2870            DB    72    * AX
332C 86         2880            DB    134   * IX
332D 8D         2890            DB    141   * ER
332E AF         2900            DB    175   * UX
332F 77         2910            DB    119   * OH
3330 55         2920            DB    85    * AW
3331 55         2930            DB    85    * AY
3332 6D         2940            DB    109   * OY
3333 71         2950            DB    113   * EY
3334 8D         2960            DB    141   * RX
3335 95         2970            DB    149   * LX
3336 AB         2980            DB    171   * WX
3337 98         2990            DB    152   * YX
3338 A5         3000            DB    165   * WH
3339 00         3010            DB    0     * EL
333A 00         3020            DB    0     * EM
333B 00         3030            DB    0     * EN
333C A8         3040            DB    168   * R
333D A2         3050            DB    162   * L
333E B3         3060            DB    179   * W
333F C2         3070            DB    194   * Y
3340 EF         3080            DB    239   * M
3341 EF         3090            DB    239   * N
3342 EF         3100            DB    239   * NX
3343 BE         3110            DB    190   * P
3344 BE         3120            DB    190   * T
3345 BE         3130            DB    190   * K
3346 BE         3140            DB    190   * KX
3347 BE         3150            DB    190   * B
3348 BE         3160            DB    190   * D
3349 BE         3170            DB    190   * G
334A BE         3180            DB    190   * GX
334B BE         3190            DB    190   * DX
334C BE         3200            DB    190   * F
334D BE         3210            DB    190   * TH
334E BE         3220            DB    190   * S
334F BE         3230            DB    190   * SH
3350 BE         3240            DB    190   * V
3351 BE         3250            DB    190   * DH
3352 BE         3260            DB    190   * Z
3353 BE         3270            DB    190   * ZH
3354 00         3280            DB    0     * CH
3355 00         3290            DB    0     * JH
3356 80         3300            DB    128   * HH
3357 80         3310            DB    128   * Q
3358            3320 *
3358            3330 *  F2 TARGET TABLE
3358            3340 *
3358 00         3350 F2TAR      DB    0     * SPACE
3359 00         3360            DB    0     * PERIOD
335A 80         3370            DB    128   * COMMA (PAUSE)
335B 00         3380            DB    0     * QUEST
335C 00         3390            DB    0     * # (TERMINATOR)
335D 58         3400            DB    88    * IY
335E 6A         3410            DB    106   * IH
335F 75         3420            DB    117   * EH
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3360 71         3430            DB    113   * AE
3361 56         3440            DB    150   * AA
3362 A0         3450            DB    160   * AH
3363 B3         3460            DB    179   * AO
3364 C2         3470            DB    194   * OW
3365 B6         3480            DB    182   * UH
3366 9B         3490            DB    155   * UW
3367 A0         3500            DB    160   * AX
3368 79         3510            DB    121   * IX
3369 92         3520            DB    146   * ER
336A 8D         3530            DB    141   * UX
336B C2         3540            DB    194   * OH
336C 96         3550            DB    150   * AW
336D 96         3560            DB    150   * AY
336E B3         3570            DB    179   * OY
336F 6A         3580            DB    106   * EY
3370 8D         3590            DB    141   * RX
3371 B1         3600            DB    177   * LX
3372 CC         3610            DB    204   * WX
3373 64         3620            DB    100   * YX
3374 CC         3630            DB    204   * WH
3375 00         3640            DB    0     * EL
3376 00         3650            DB    0     * EM
3377 00         3660            DB    0     * EN
3378 B1         3670            DB    177   * R
3379 B8         3680            DB    184   * L
337A DC         3690            DB    220   * W
337B 64         3700            DB    100   * Y
337C BE         3710            DB    190   * M
337D 79         3720            DB    121   * N
337E 6A         3730            DB    106   * NX
337F BE         3740            DB    190   * P
3380 79         3750            DB    121   * T
3381 4D         3760            DB    77    * K
3382 8D         3770            DB    141   * KX
3383 BE         3780            DB    190   * P
3384 79         3790            DB    121   * D
3385 4D         3800            DB    77    * G
3386 8D         3810            DB    141   * GX
3387 79         3820            DB    121   * DX
3388 BE         3830            DB    190   * F
3389 80         3840            DB    128   * TH
338A 92         3850            DB    146   * S
338B 5E         3860            DB    94    * SH
338C BE         3870            DB    190   * V
338D 80         3880            DB    128   * DH
338E 92         3890            DB    146   * Z
338F 5E         3900            DB    94    * ZH
3390 00         3910            DB    0     * CH
3391 00         3920            DB    0     * JH
3392 80         3930            DB    128   * HH
3393 60         3940            DB    128   * Q
3394            3950 *
3394            3960 *  F3 TARGET TABLE
3394            3970 *
3394 00         3980 F3TAR      DB    0     * SPACE
3395 00         3990            DB    0     * PERIOD
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3396 AC          4000           DB    172   * COMMA (PAUSE)
3397 00          4010           DB    0     * QUEST
3398 00          4020           DB    0     * # (TERMINATOR)
3399 84          4030           DB    132   * IY
339A A7          4040           DB    167   * IH
339B B0          4050           DB    176   * EH
339C B5          4060           DB    181   * AE
339D 9B          4070           DB    155   * AA
339E 9F          4080           DB    159   * AH
339F AC          4090           DB    172   * AO
33A0 C8          4100           DB    200   * OW
33A1 BE          4110           DB    190   * UH
33A2 BE          4120           DB    190   * UW
33A3 A3          4130           DB    163   * AX
33A4 B5          4140           DB    181   * IX
33A5 FF          4150           DB    255   * ER
33A6 BE          4160           DB    190   * UX
33A7 C8          4170           DB    200   * OH
33A8 9B          4180           DB    155   * AW
33A9 9B          4190           DB    155   * AY
33AA AC          4200           DB    172   * OY
33AB B0          4210           DB    176   * EY
33AC FF          4220           DB    255   * RX
33AD 8B          4230           DB    139   * LX
33AE C8          4240           DB    200   * WX
33AF AC          4250           DB    172   * YX
33B0 B9          4260           DB    185   * WH
33B1 00          4270           DB    0     * EL
33B2 00          4280           DB    0     * EM
33B3 00          4290           DB    0     * EN
33B4 FF          4300           DB    255   * R
33B5 88          4310           DB    136   * L
33B6 B9          4320           DB    185   * W
33B7 84          4330           DB    132   * Y
33B8 DC          4340           DB    220   * M
33B9 A3          4350           DB    163   * N
33BA F3          4360           DB    243   * NX
33BB DC          4370           DB    220   * P
33BC A3          4380           DB    163   * T
33BD AC          4390           DB    172   * K
33BE FF          4400           DB    255   * KX
33BF DC          4410           DB    220   * B
33C0 A3          4420           DB    163   * D
33C1 AC          4430           DB    172   * G
33C2 FF          4440           DB    255   * GX
33C3 A3          4450           DB    163   * DX
33C4 D2          4460           DB    210   * F
33C5 9B          4470           DB    155   * TH
33C6 A7          4480           DB    167   * S
33C7 9B          4490           DB    155   * SH
33C8 D2          4500           DB    210   * V
33C9 9B          4510           DB    155   * DH
33CA A7          4520           DB    167   * Z
33CB 9B          4530           DB    155   * ZH
33CC 00          4540           DB    0     * CH
33CD 00          4550           DB    0     * JH
33CE AC          4560           DB    172   * HH
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

33CF AC          4570           DB    172   * Q
33D0             4580 *
33D0             4590 *  AV/AH TARGET TABLE
33D0             4600 *
33D0 00          4610 AVHTAR    DB    0     * SPACE
33D1 00          4620           DB    0     * PERIOD
33D2 00          4630           DB    0     * COMMA (PAUSE)
33D3 00          4640           DB    0     * QUEST
33D4 00          4650           DB    0     * # (TERMINATOR)
33D5 E0          4660           DB    224   * IY
33D6 E0          4670           DB    224   * IH
33D7 E0          4680           DB    224   * EH
33D8 E0          4690           DB    224   * AE
33D9 E0          4700           DB    224   * AA
33DA E0          4710           DB    224   * AH
33DB E0          4720           DB    224   * AO
33DC E0          4730           DB    224   * OW
33DD E0          4740           DB    224   * UH
33DE E0          4750           DB    224   * UW
33DF E0          4760           DB    224   * AX
33E0 E0          4770           DB    224   * IX
33E1 E0          4780           DB    224   * ER
33E2 E0          4790           DB    224   * UX
33E3 E0          4800           DB    224   * OH
33E4 E0          4810           DB    224   * AW
33E5 E0          4820           DB    224   * AY
33E6 E0          4830           DB    224   * OY
33E7 E0          4840           DB    224   * EY
33E8 E0          4850           DB    224   * RX
33E9 E0          4860           DB    224   * LX
33EA E0          4870           DB    224   * WX
33EB E0          4880           DB    224   * YX
33EC EA          4890           DB    234   * WH
33ED 00          4900           DB    0     * EL
33EE 00          4910           DB    0     * EM
33EF 00          4920           DB    0     * EN
33F0 A0          4930           DB    160   * R
33F1 A0          4940           DB    160   * L
33F2 A0          4950           DB    160   * W
33F3 A0          4960           DB    160   * Y
33F4 A0          4970           DB    160   * M
33F5 A0          4980           DB    160   * N
33F6 A0          4990           DB    160   * NX
33F7 00          5000           DB    0     * P
33F8 00          5010           DB    0     * T
33F9 00          5020           DB    0     * K
33FA 00          5030           DB    0     * KX
33FB 40          5040           DB    64    * B
33FC 40          5050           DB    64    * D
33FD 40          5060           DB    64    * G
33FE 40          5070           DB    64    * GX
33FF 00          5080           DB    0     * DX
3400 00          5090           DB    0     * F
3401 00          5100           DB    0     * TH
3402 05          5110           DB    5     * S
3403 08          5120           DB    8     * SH
3404 E0          5130           DB    224   * V
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3405 E0         5140            DB    224   * DH
3406 E5         5150            DB    229   * Z
3407 E8         5160            DB    232   * ZH
3408 00         5170            DB    0     * CH
3409 00         5180            DB    0     * JH
340A 0A         5190            DB    10    * HH
340B 80         5200            DB    128   * Q
340C            5210 *
340C            5220 *  FF/AF/AN TARGET TABLE
340C            5230 *
340C 80         5240 FRNTAR     DB    128   * SPACE
340D 80         5250            DB    128   * PERIOD
340E 80         5260            DB    128   * COMMA (PAUSE)
340F 80         5270            DB    128   * QUESTT
3410 80         5280            DB    128   * $ (TERMINATOR)
3411 80         5290            DB    128   * IY
3412 80         5300            DB    128   * IH
3413 80         5310            DB    128   * EH
3414 80         5320            DB    128   * AE
3415 80         5330            DB    128   * AA
3416 80         5340            DB    128   * AH
3417 80         5350            DB    128   * AO
3418 80         5360            DB    128   * OW
3419 80         5370            DB    128   * UH
341A 80         5380            DB    128   * UW
341B 80         5390            DB    128   * AX
341C 80         5400            DB    128   * IX
341D 80         5410            DB    128   * ER
341E 80         5420            DB    128   * UX
341F 80         5430            DB    128   * OH
3420 80         5440            DB    128   * AW
3421 80         5450            DB    128   * AY
3422 80         5460            DB    128   * OY
3423 80         5470            DB    128   * EY
3424 80         5480            DB    128   * RX
3425 80         5490            DB    128   * LX
3426 80         5500            DB    128   * WX
3427 80         5510            DB    128   * YX
3428 80         5520            DB    128   * WH
3429 80         5530            DB    128   * EL
342A 80         5540            DB    128   * EM
342B 80         5550            DB    128   * EN
342C 80         5560            DB    128   * R
342D 80         5570            DB    128   * L
342E 80         5580            DB    128   * W
342F 80         5590            DB    128   * Y
3430 82         5600            DB    130   * M
3431 82         5610            DB    130   * N
3432 82         5620            DB    130   * NX
3433 80         5630            DB    128   * P
3434 80         5640            DB    128   * T
3435 80         5650            DB    128   * K
3436 80         5660            DB    128   * KX
3437 80         5670            DB    128   * B
3438 80         5680            DB    128   * D
3439 80         5690            DB    128   * G
343A 80         5700            DB    128   * GX
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

343B 80          5710           DB    128   * DX
343C B0          5720           DB    176   * F
343D 70          5730           DB    112   * TH
343E 98          5740           DB    152   * S
343F 74          5750           DB    116   * SH
3440 B0          5760           DB    176   * V
3441 74          5770           DB    116   * DH
3442 98          5780           DB    152   * Z
3443 74          5790           DB    116   * ZH
3444 60          5800           DB    128   * CH
3445 80          5810           DB    128   * JH
3446 80          5820           DB    128   * HH
3447 80          5830           DB    128   * Q
3448             5840 *
3448             5850 *  RANK/PC/ID TABLE
3448             5860 *
3448 00          5870 RAPCID     DB    0     * SPACE
3449 00          5880           DB    0     * PERIOD
344A 80          5890           DB    128   * COMMA (PAUSE)
344B 00          5900           DB    0     * QUEST
344C 00          5910           DB    0     * # (TERMINATOR)
344D 11          5920           DB    17    * IY
344E 11          5930           DB    17    * IH
344F 11          5940           DB    17    * EH
3450 11          5950           DB    17    * AE
3451 11          5960           DB    17    * AA
3452 11          5970           DB    17    * AH
3453 11          5980           DB    17    * AO
3454 11          5990           DB    17    * OW
3455 11          6000           DB    17    * UH
3456 11          6010           DB    17    * UW
3457 11          6020           DB    17    * AX
3458 11          6030           DB    17    * IX
3459 11          6040           DB    17    * ER
345A 11          6050           DB    17    * UX
345B 11          6060           DB    17    * OH
345C 11          6070           DB    17    * AW
345D 11          6080           DB    17    * AY
345E 11          6090           DB    17    * OY
345F 11          6100           DB    17    * EY
3460 11          6110           DB    17    * RX
3461 3E          6120           DB    62    * LX
3462 31          6130           DB    49    * WX
3463 31          6140           DB    49    * YX
3464 3E          6150           DB    62    * WH
3465 00          6160           DB    0     * EL
3466 00          6170           DB    0     * EM
3467 00          6180           DB    0     * EN
3468 3E          6190           DB    62    * R
3469 3E          6200           DB    62    * L
346A 3E          6210           DB    62    * W
346B 3E          6220           DB    62    * Y
346C 53          6230           DB    83    * M
346D 53          6240           DB    83    * N
346E 53          6250           DB    83    * NX
346F 52          6260           DB    82    * P
3470 52          6270           DB    82    * T
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

3471 52         6280            DB    82   * K
3472 52         6290            DB    82   * KX
3473 53         6300            DB    83   * P
3474 53         6310            DB    83   * D
3475 53         6320            DB    83   * G
3476 53         6330            DB    83   * GX
3477 53         6340            DB    83   * DX
3478 7C         6350            DB    124  * F
3479 7C         6360            DB    124  * TH
347A 7C         6370            DB    124  * S
347B 7C         6380            DB    124  * SH
347C 7D         6390            DB    125  * V
347D 7D         6400            DB    125  * DH
347E 7D         6410            DB    125  * Z
347F 7D         6420            DB    125  * ZH
3480 00         6430            DB    0    * CH
3481 00         6440            DB    0    * JH
3482 86         6450            DB    134  * NH
3483 83         6460            DB    131  * Q
3484            6470 *
3484            6480 *  TRANSITION TIMES TABLE
3484            6490 *
3484 00         6500 TTIMES     DB    0    * SPACE
3485 00         6510            DB    0    * PERIOD
3486 0A         6520            DB    10   * COMMA (PAUSE)
3487 00         6530            DB    0    * QUEST
3488 00         6540            DB    0    * ; (TERMINATOR)
3489 AA         6550            DB    170  * IY
348A AA         6560            DB    170  * IH
348B AA         6570            DB    170  * EH
348C AA         6580            DB    170  * AE
348D AA         6590            DB    170  * AA
348E AA         6600            DB    170  * AH
348F AA         6610            DB    170  * AO
3490 AA         6620            DB    170  * OW
3491 AA         6630            DB    170  * UH
3492 AA         6640            DB    170  * UW
3493 AA         6650            DB    170  * AX
3494 AA         6660            DB    170  * IX
3495 AA         6670            DB    170  * ER
3496 AA         6680            DB    170  * UX
3497 AA         6690            DB    170  * OH
3498 AA         6700            DB    170  * AW
3499 AA         6710            DB    170  * AY
349A AA         6720            DB    170  * OY
349B AA         6730            DB    170  * EY
349C 77         6740            DB    119  * RX
349D 74         6750            DB    116  * LX
349E AA         6760            DB    170  * WX
349F AA         6770            DB    170  * YX
34A0 74         6780            DB    116  * HH
34A1 00         6790            DB    0    * BL
34A2 00         6800            DB    0    * EM
34A3 00         6810            DB    0    * EN
34A4 74         6820            DB    116  * R
34A5 74         6830            DB    116  * L
34A6 74         6840            DB    116  * W
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

34A7 74         6850            DB    116 * Y
34A8 60         6860            DB    96  * M
34A9 70         6870            DB    112 * N
34AA 90         6880            DB    144 * NX
34AB 50         6890            DB    80  * P
34AC 70         6900            DB    112 * T
34AD 90         6910            DB    144 * K
34AE 90         6920            DB    144 * KX
34AF 60         6930            DB    96  * B
34B0 70         6940            DB    112 * D
34B1 90         6950            DB    144 * G
34B2 90         6960            DB    144 * GX
34B3 70         6970            DB    112 * DX
34B4 54         6980            DB    84  * F
34B5 54         6990            DB    84  * TH
34B6 54         7000            DB    84  * S
34B7 54         7010            DB    84  * SH
34B8 54         7020            DB    84  * V
34B9 54         7030            DB    84  * DH
34BA 54         7040            DB    84  * Z
34BB 54         7050            DB    84  * ZH
34BC 00         7060            DB    0   * CH
34BD 00         7070            DB    0   * JH
34BE 0A         7080            DB    10  * HH
34BF 0A         7090            DB    10  * Q
34C0            7100 *
34C0            7110 *  AF PC/AV PC ARRAY, PERCENT CROSSING VALUES
34C0            7120 *  (BITS 3-5)/32 = AF PERCENT BOUNDARY XING VALUE
34C0            7130 *  (BITS 0-2)/4 = AV PERCENT BOUNDARY XING VALUE
34C0            7140 *
34C0 24         7150 PCVP       DB    32+4  * OLD CID=0, NEW CID=0
34C1 09         7160            DB    8+1   * OLD CID=0, NEW CID=1
34C2 00         7170            DB    0     *     "          "   =2
34C3 00         7180            DB    0     *    ETC.
34C4 09         7190            DB    8+1
34C5 09         7200            DB    8+1
34C6 09         7210            DB    8+1
34C7 09         7220            DB    8+1   * OLD CID=1, NEW CID=0
34C8 24         7230            DB    32+4  *         1
34C9 12         7240            DB    16+2                      1
34CA 12         7250            DB    16+2
34CB 12         7260            DB    16+2
34CC 14         7270            DB    16+4
34CD 24         7280            DB    32+4
34CE 00         7290            DB    0     *  OLD =2,  NEW =0
34CF 12         7300            DB    16+2
34D0 24         7310            DB    32+4
34D1 00         7320            DB    0
34D2 12         7330            DB    16+2
34D3 12         7340            DB    16+2
34D4 12         7350            DB    16+2
34D5 00         7360            DB    0     *   3,  0
34D6 12         7370            DB    16+2
34D7 00         7380            DB    0
34D8 24         7390            DB    32+4
34D9 12         7400            DB    16+2
34DA 12         7410            DB    16+2
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

34DB 12          7420           DB    16+2
34DC 09          7430           DB    8+1     *   ETC.
34DD 12          7440           DB    16+2
34DE 12          7450           DB    16+2
34DF 12          7460           DB    16+2
34E0 24          7470           DB    32+4
34E1 24          7480           DB    32+4
34E2 12          7490           DB    16+2
34E3 09          7500           DB    8+1
34E4 14          7510           DB    16+4
34E5 12          7520           DB    16+2
34E6 12          7530           DB    16+2
34E7 22          7540           DB    32+2
34E8 24          7550           DB    32+4
34E9 14          7560           DB    16+4
34EA 09          7570           DB    8+1
34EB 24          7580           DB    32+4
34EC 12          7590           DB    16+2
34ED 12          7600           DB    16+2
34EE 12          7610           DB    16+2
34EF 14          7620           DB    16+4
34F0 24          7630           DB    32+4
34F1             7640 *
34F1             7650 *  AF/AV TRANSITION TIME TABLE
34F1             7660 *  (BITS 3-5)/8 = AF TRANSITION TIME IN FRAMES
34F1             7670 *  (BITS 0-2) = AV TRANSITION TIME IN FRAMES
34F1 00          7680 TAVF       DB    0       * CID=0
34F2 0B          7690           DB    8+3     *   =1
34F3 00          7700           DB    0       *   =2
34F4 00          7710           DB    0       *   =3
34F5 18          7720           DB    24+0    *   =4
34F6 1B          7730           DB    24+3    *   =5
34F7 0B          7740           DB    8+3     *   =6
34F8             7750 *
34F8             7760 *  END OF SECTION 6
```

APPENDIX C

P-3 FLEET PILOT QUESTIONNAIRE

This questionnaire has been prepared by a Thesis student in the Aeronautical Engineering Curriculum at the Naval Postgraduate School, Monterey, California. The general topic of the Thesis deals with aircraft warning systems; those in the current state of the art and those proposed for future aircraft. This questionnaire specifically deals with the P-3 Orion aircraft. Part One deals with the current system, its strengths and weaknesses. Part Two is concerned with proposals for future P-3 cockpit designs. Some questions require brief written answers. Please do not spend too much time on these. Incomplete sentence structure is acceptable, just ensure your idea is there. You are also asked to complete the following items that we may have a good idea of your experience level. This does not mean we will ignore "low time JO's", however. A cockpit must be designed to adequately accomodate all experience levels of pilots, and we want to know what each type of pilot feels about the P-3. The name is optional, of course, but would help us to recontact you in the event one of your ideas merits further consideration and/or requires clarification. Also we would like to give credit where it is due. If you have any comments or additional ideas in this area not covered by this questionnaire, please feel free to use any free space for your comments or attach an additional sheet if necessary.

NAME _____

AIRCRAFT QUALIFICATIONS _____

P-3 QUALIFICATIONS _____

PREVIOUS SQUADRON TOURS _____

FLIGHT TIME IN P-3 _____

TOTAL FLIGHT TIME _____

PART 1

The questions in this part are concerned with the adequacy of the current instruments/warning systems in the P-3 aircraft.

1. Do you feel that the current instrument/warning system in the P-3 is:

   a) more than adequate?
   b) adequate?
   c) could be improved upon?

2. What, if any, instrument/warning devices would you replace and with what?

3. What additional instruments do you feel would significantly improve the safety and/or instrument flight capabilities of the P-3?

4. How do you feel about the placement of the instruments in the P-3 cockpit?

   a) excellent - no problems with position
   b) acceptable - only a few instruments are inconveniently placed
   c) marginal - many instruments are inconvenient
   d) unacceptable - critical instruments are not readily accessible in certain circumstances

5. If your answer to question 4 was c) or d), please briefly explain your answer. List the instruments in question, and the particular instance when their placement would make them difficult to read.

6. Do you have any other comments concerning the current instrument/warning system in the P-3?

PART 2

The second part of this questionnaire is concerned with possible devices which may be included in future P-3 models.

1. Recognizing the fact that a pilot or flight engineer may be able to gain valuable information from an instrument due to its relative needle position, how do you feel about replacing all dial guages with digital readouts?

   a) don't like the idea - requires each individual guage be read and takes too much time
   b) think its a good idea - like the accuracy of the digital readouts
   c) no opinion either way
   d) other (please explain)

2. Suppose all dial guages were replaced with digital readouts, the numerals of which changed color according to the following code:

   GREEN - normal range
   YELLOW - outside normal range, but still safe
   RED - danger zone

   Now how would you feel about these instruments?

   a) still don't like the idea
   b) didn't like it before, but do now
   c) like the digital readouts, but don't think the color code is necessary
   d) like digital readouts more now because of the color code

3. If a Heads Up Display were proposed for the P-3, what information would you desire it present to: a) the pilot?, b) the copilot?

4. Do you feel the HUD should be projected on the side as well as the forward windshield? Why?

5. Do you feel it would be advantageous to have the HUD incorporate the Master Caution lights (i.e., when a Master Caution light illuminates, it is also repeated on the HUD)?

Assuming that a computer could be designed which would be capable of monitoring all flight systems and parameters (engine performance, pressurization, electrical system, etc.) and be linked to the navigation system so that it would "know" at all times the position of the aircraft, consider the following questions:

6. If the computer could perform certain emergency procedures for the pilot, which would you prefer it be allowed to handle?

7. Which would you not allow the computer to handle under any circumstances?

8. Which would you allow it to perform only with pilot approval (such approval to be given by alphanumeric code or spoken into a microphone)?

9. How would you prefer the computer warn the pilot of impending failures or abnormal readings?

   a) by a printed message on a CRT
   b) by speaking the message (assume the computer can be made to "talk" in an easily recognizable "voice")
   c) both of the above methods
   d) other (please explain)

10. How would you feel about a system where the computer had primary control of the landing gear and flap system? Assume that the functions were voice controlled and the computer would not allow gear or flap extension above limit airspeeds. Also assume that the current system remained as a mechanical backup.

   a) strongly oppose the concept - too dangerous
   b) like the idea - the computer would lower the geat and/or flaps when called for and not miss the call
   c) no opinion
   d) other (please explain)

11. Would you prefer that the computer be programmed to lower the gear and flaps as the aircraft approached the ground? Assume the system were armed only below 500 feet and when combinations of low airspeed/power settings indicated to the computer that the aircraft was being landed. (Obviously, this is designed to prevent gear up landings. Comments, please.)

   a) don't like the idea (Why?)
   b) like the idea because it would prevent gear up landings
   c) like the idea because it would reduce pilot workload
   d) other (please explain)

12. Briefly describe in the following space any ideas you might have concerning cockpit design. Suggestions are specifically requested for improvements in cockpit instrumentation/warning devices, items which you would like to see delegated to a computer (with or without pilot approval), and anything which would reduce pilot workload in an emergency situation. Also use this space if you did not have enough room to answer any of the previous questions. Be sure to number your answers appropriately.

LIST OF REFERENCES

1. Defense Advanced Research Agency Report, R-1434-ARPA, Military Applications of Speech Understanding Systems, by Hoffman, A., Lippiatt, T., and Turn, R., p. 1-44, June 1974.

2. Hueristics, Inc., SPEECHLAB Hardware Assembly and Test Manual, p. 2-5.

3. Computalker Consultants, CSR1 Theory and Operation Manual, p. 1-4, May 1977.

4. National Aeronautics and Space Administration, Proceedings from the symposium on Voice Technology for Interactive Real-Time Command/Control Systems Application, Ames Research Center, p. 143-165, 6-8 December 1977.

5. Hueristics, Inc., SPEECHLAB Hardware Assembly and Test Manual, p. 3-2 - 3-10.

6. Computalker Consultants, CSR1 Theory and Operation Manual, p. 1, May 1977.

7. SCOPE Electronics, Inc., TECHNICAL REPORT AFAL - TR - 72 - 290, Voice Activated Cockpit Control, Kleiner, Norbert, and Miller, Kenneth, p. 2, p. 7-9, October 1972.

8. Defense Advanced Research Agency Report, R-1434-ARPA, Military Applications of Speech Understanding Systems, by Hoffman, A., Lippiatt, T., and Turn, R., p. 67, June 1974.

9. Hueristics, Inc., SPEECHLAB Hardware Assembly and Test Manual, Laboratory Manual, p. 3-75 - 3-95.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Assoc. Professor D. M. Layton, Code 67Ln Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Mr. Paul Sparks, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	LT Albert George Mertz, USN AIRTEVRON ONE (VX-1) Naval Air Station Patuxent River, Maryland 20653	1

Thesis
M544
c.1

Mertz

A preliminary investigation of aural input/output systems for in-flight information retieval.