

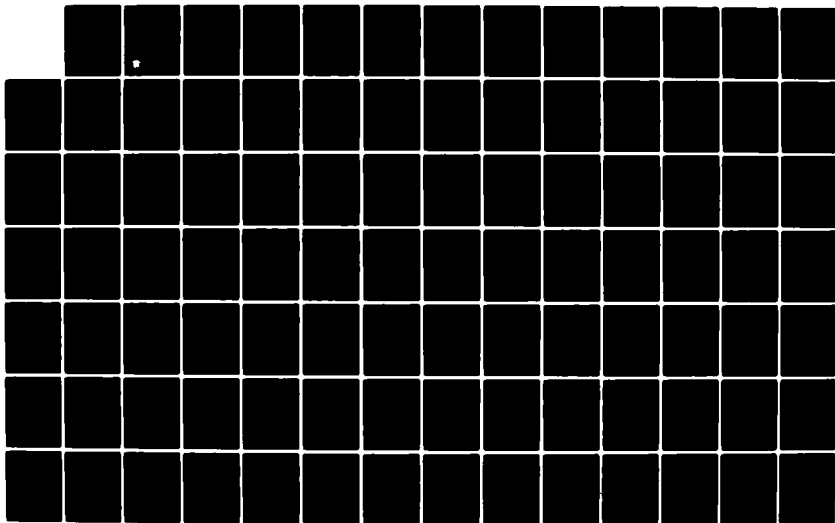
AD-A135 233

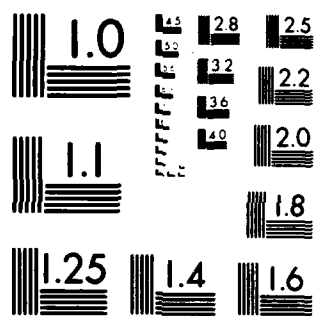
REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION
ALGORITHM COMBINING TIM... (U) NOTRE DAME UNIV IN DEPT OF
ELECTRICAL ENGINEERING J L MELSA ET AL. JUN 83
DCA100-82-C-0026

1/2

F/G 9/2

NL





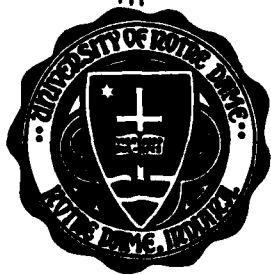
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(15)

AD-A135233

DTIC
S
DEC 2 1983
A

DTIC FILE COPY



Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

This document has been approved
for public release and sale; its
distribution is unlimited.

83 12 02 027

FINAL REPORT - VOLUME 2
REAL-TIME IMPLEMENTATION
OF A
SPEECH DIGITIZATION ALGORITHM
COMBINING
TIME-DOMAIN HARMONIC SCALING
AND
ADAPTIVE RESIDUAL CODING

STIC
ELECT
DEC 2 1983
A

Prepared for
Defense Communications Agency
Defense Communications Engineering Center
1860 Wichle Avenue
Reston, Virginia 22090

Contract No. DCA-100-82-C-0026

1 June 1983

This document has been approved
for public release and sale; its
distribution is unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION ALGORITHM COMBINING TIME-DOMAIN HARMONIC SCALING AND ADAPTIVE RESIDUAL CODING		5. TYPE OF REPORT & PERIOD COVERED Final Feb. 1982 to May 1983
7. AUTHOR(s) James L. Melsa James D. Mills Arvind A. Arora		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering University of Notre Dame Notre Dame, IN 46556		8. CONTRACT OR GRANT NUMBER(s) DCA-100-82-C-0026
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency Contract Management Division, Code 260 Washington, DC 20305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1983
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited. If may be released to the Clearinghouse, Department of Commerce, for sale to the public.		
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 coding, 16000 bps speech transmission, time-domain harmonic scaling, adaptive residual coding, hardware realization.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the results of a fifteen-month study, supported by DCA Contract 100-82-C-0026, of the real-time implementation of algorithm com- bining time-domain harmonic scaling (TDHS) and Adaptive Residual Coding (ARC) at a transmission bit rate of 16 kb/s. The modifications of this encoding algorithm as originally presented by Melsa and Pande (1981) to allow real-time implementation are described in detail. A non real-time FORTRAN simulation (continued)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

using a sixteen-bit word length was developed and tested to establish feasibility. The hardware implementation of a full-duplex, real-time system has demonstrated that this algorithm is capable of producing toll quality speech digitization.

This report has been divided into two volumes. The first volume discusses the algorithm modifications and FORTRAN simulation. The details of the hardware implementation, schematics for the system and operating instructions are included in Volume 2 of this final report.

ABSTRACT

This report describes the results of a fifteen-month study, supported by DCA Contract 100-82-C-0026, of the real-time implementation of an algorithm combining time-domain harmonic scaling (TDHS) and Adaptive Residual Coding (ARC) at a transmission bit rate of 16 kb/s. The modifications of this encoding algorithm as originally presented by Melsa and Pande (1981) to allow real-time implementation are described in detail. A non real-time FORTRAN simulation using a sixteen-bit word length was developed and tested to establish feasibility. The hardware implementation of a full-duplex, real-time system has demonstrated that this algorithm is capable of producing toll quality speech digitization.

This report has been divided into two volumes. The ^{Secret} first volume discusses the algorithm modifications and FORTRAN simulation. The details of the hardware implementation, schematics for the system and operating instructions, are included in Volume 2 of this final report.



Project Title	
100-82-C-0026	
Contract Number	
100-82-C-0026	
Project Description	
Implementation of a full-duplex, real-time system	
Distribution	
Availability Codes	
1.4 - UNCLASSIFIED	
Post Date	
1982	
AI	

PROJECT PERSONNEL

James L. Melsa, principal investigator

James D. Mills, research assistant

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	ii
PROJECT PERSONNEL	iii
LIST OF FIGURES AND TABLES	v
LIST OF ABBREVIATIONS	vii
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: HARDWARE IMPLEMENTATION	2
2.1 Analog Interface	2
2.2 Digital Interface	2
2.3 Clock and Timing	2
2.4 Control Center	4
2.5 Computation Center	4
2.6 Pitch Extraction	4
CHAPTER 3: INSTALLATION AND INTERFACE	7
3.1 Installation	7
3.2 Interface	7
CHAPTER 4: OPERATING INSTRUCTIONS	12
4.1 Introduction	12
4.2 Switches and Operations	12
CHAPTER 5: CONCLUSIONS	8
APPENDIX A: MAINTENANCE: SCHEMATICS AND COMPONENTS	19
APPENDIX B: REAL TIME SOFTWARE	35
B.1 Control Center Software: TARC	35
B.2 Pitch Extraction Software: PEXTR	52
B.3 Processing Software: TDHSI4, TDHSD4	67
APPENDIX C: PRODUCTION COST ESTIMATE	119
C.1 Off-the-Shelf Component Design	119
C.2 Custom LSI TDHS-ARC Transceiver Design	119

LIST OF FIGURES

		<u>Page</u>
Figure 2.1	Layout of Circuit Segments in TDHS-ARC ...	3
Figure 3.1	Null Modem Connections for TDHS-ARC	11
Figure 4.1	Front Panel Layout for the TDHS-ARC System	13
Figure 4.2	Back Panel Layout for the TDHS-ARC System	14
Figure 4.3	Functional Block Diagram of TDHS-ARC	15
Figure A.1	Layout of Circuit Segments in TDHS-ARC ...	20
Figure A.2	Schematic for the Analog Interface	21
Figure A.3	Schematic for the Digital Interface	23
Figure A.4	Schematic for the Clock and Timing	25
Figure A.5	Schematic for the Control Center	27
Figure A.6	Schematic for the Computation Center	29
Figure A.7	Schematic for the Pitch Extraction	31
Figure A.8	Schematic for the Back Plane	33
Figure A.9	Edge Connector	34
Figure B.1	Flow Chart for the Control Program, TARC .	37
Figure B.2	Flow Chart for the Pitch Extraction Program, PEXTR	54
Figure B.3	Flow Chart for Initialization	68
Figure B.4	Main Interrupt Service Routine	69
Figure B.5	Center Clip Routine #1 for Transmitter ...	73
Figure B.6	TDHC Routine	74
Figure B.7	ARC Transmitter Routine	75
Figure B.8	ARC Receiver Routine	77
Figure B.9	Center Clip Routine for Receiver	79
Figure B.10	Center Clip Routine #2 for Transmitter ...	80
Figure B.11	TDHE Routine #1	81

LIST OF FIGURES (cont'd)

		<u>Page</u>
Figure B.12	TDHE Routine #2	82
Figure C.1	Die Segmentation of a Custom TDHS-ARC Transceiver	120
Table 2.1	Data Flow in TDHS-ARC Supervised by the Control Center	5
Table 3.1	Handset Jack Connections	8
Table 3.2	Tape Input/Output Connections	8
Table 3.3	RS-232 Signal Connections	10
Table A.1	Parts List for the Analog Interface	22
Table A.2	Parts List for the Digital Interface	24
Table A.3	Parts List for the Clock and Timing	26
Table A.4	Parts List for the Control Center	28
Table A.5	Parts List for the Computation Center	30
Table A.6	Parts List for the Pitch Extraction	32

LIST OF ABBREVIATIONS

DCA	Defense Communications Agency
TDHS	time domain harmonic scaling
ARC	adaptive residual coder
FWL	finite word length
TDHC	time domain harmonic compression
TDHE	time domain harmonic expansion
CVSD	continuously variable slope delta modulator
$s(k)$	input sampled speech signal
$y(k)$	compressed speech signal
$q(k)$	quantizer level sequence
$\hat{q}(k)$	system estimate of the original $q(k)$
$\hat{y}(k)$	system estimate of the original $y(k)$
$\hat{s}(k)$	system estimate of the original $s(k)$
LPF	low pass filter
FIR	finite impulse response
Hz	Hertz
kb/s	kilobits per second
b/s	bits per second
BER	bit error rate
Δ	triangular or change in
NQ	number of quantization level
MSF	maximum scaling factor
SF_q	q^{th} scaling factor
TH_q	q^{th} threshold

$e(k)$	error sequence
$\hat{e}(k)$	system estimate of original $e(k)$
$\sigma(k)$	approximation to the standard deviation of input sequence
dB	decibels
C_k	clipping level
E_k	approximation of the energy

CHAPTER 1

INTRODUCTION

This volume describes various aspects of the TDHS-ARC hardware system. The TDHS-ARC algorithm and its development is described in Volume 1 of this report.

Chapter 2 discusses the algorithm from the point of view of the hardware partitioning. Chapter 3 describes the installation and interface procedures. Chapter 4 details the operating instructions. The final chapter comments about the TDHS-ARC hardware and its further potential. Appendix A has the schematics and parts lists for maintenance purposes. Appendix B has the listings for the real-time software in the three processors on the TDHS-ARC board. Finally, Appendix C presents a cost estimate for the TDHS-ARC system using off-the-shelf LSI components and a custom LSI TDHS-ARC transceiver.

CHAPTER 2

HARDWARE IMPLEMENTATION

The following section describes the TDHS-ARC system from the hardware point of view. The complete system, including the interface circuitry, is implemented on a single board. The hardware is divided into six functional segments. Figure 2.1 shows the hardware block diagram and the layout of the different segments in the hardware. There are two interface sections, a clock and timing section, a control section, and two sections that actually implement the TDHS algorithm.

2.1 Analog Interface (Ref. A*)

The analog interface, set to one side of the board, has its ground plane isolated from the rest of the system to minimize noise pickup. The section consists of the filters and codec to bandlimit the speech signal (50 Hz -3400 Hz) and to encode it in 8-bit μ -law format. It also has amplifiers for gain control to interface signals properly to the telephone and tape input/output.

2.2 Digital Interface (Ref. D)

The next segment is the digital interface and it performs two functions. It does the level conversion between TTL signals (clock and data) and the bipolar RS-232 signals required to connect two TDHS-ARC systems. It also does the code-word synchronization on the incoming data stream. The synchronization is achieved by monitoring the alternation of code words 0000 and 1111 both of which are assigned to quantizer level 0 which has a very high probability of occurrence, especially during silence.

2.3 Clock and Timing (Ref. T)

The clock and timing section has a crystal stabilized voltage controlled oscillator at 8.192 MHz to generate the system clock. It can be free-run when the system is operated stand alone, or it can be locked to the clock of

*Refers to schematic diagram labels.

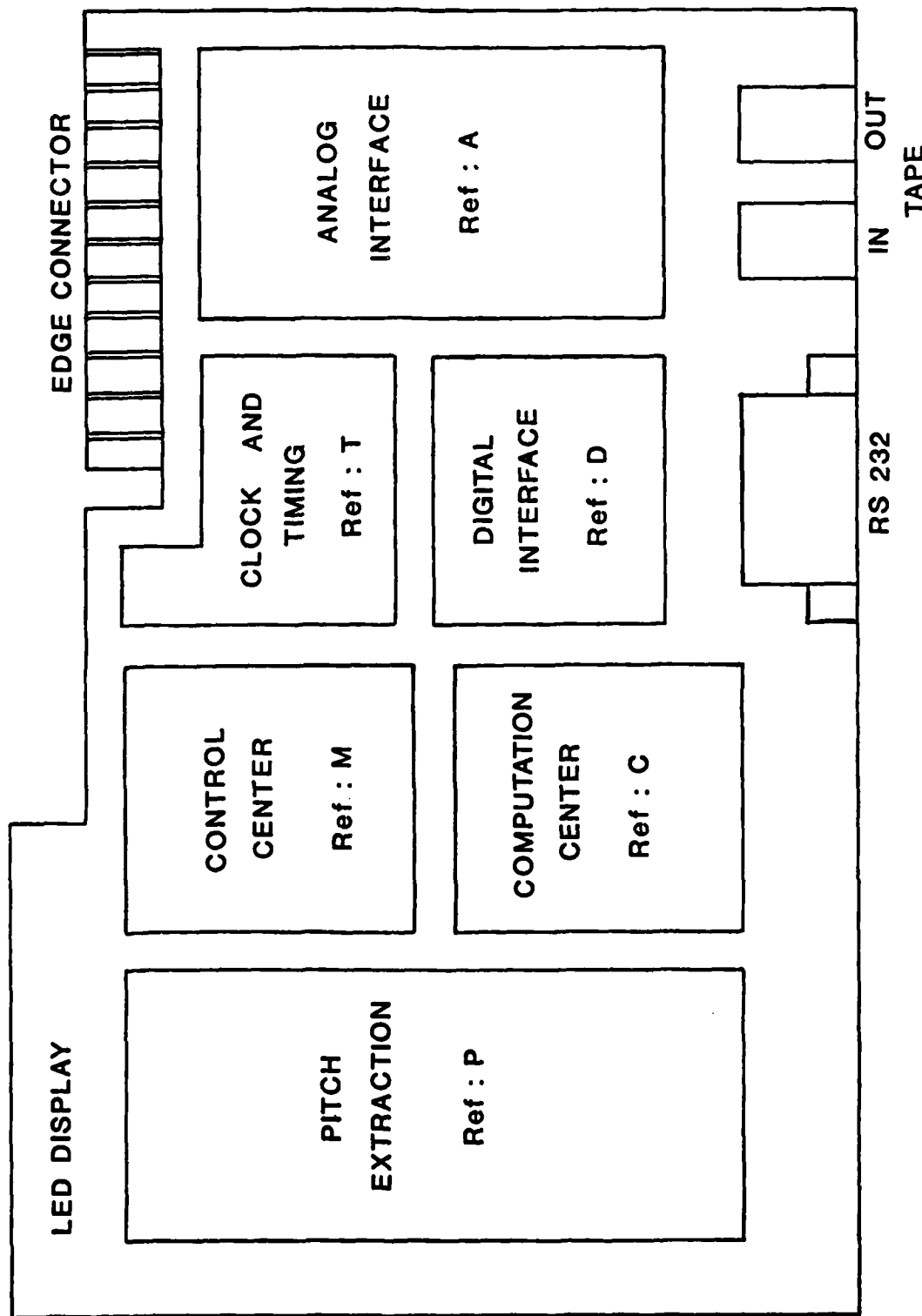


Fig. 2.1: Layout of Circuit Segments in TDHS-ARC

another TDHS-ARC system to enable perfectly synchronous operation between the two systems. The system clock is divided down to provide the different timing signals in the rest of the system.

2.4 Control Center (Ref. M)

The control center is based on an 8085 microprocessor. It contains the buffers to store speech data for the transmitter and receiver sections of the algorithm, and it is responsible for supervising all the data transfer from one section of the system to another. Table 2.1 shows the data flow in the system. In addition, the control center keeps track of the pitch period. Whenever necessary, it directs the pitch extraction section to find a new pitch period for the transmitter or the receiver.

The TDHS-ARC algorithm is actually implemented in the next two sections of the hardware, the computation center and the pitch extraction section.

2.5 Computation Center (Ref. C)

The arithmetic operations in the TDHS-ARC algorithm is done in the computation center. This section consists of a signal processing chip (NEC7720) where all the computation is performed, and the various interface circuitry to interface the signal processing chip to the control center. All data is transferred through 16 word FIFO's. The NEC chip also has a serial interface to the codec in the analog interface section.

2.6 Pitch Extraction (Ref. P)

The pitch extraction section estimates a new pitch period for the transmitter and receiver whenever directed by the control center. Pitch extraction is done using a three-level, center-clipped autocorrelation technique. This section consists of a microprocessor, dual memory for parallel access of the clipped speech, and hardware for addressing and correlation. The microprocessor initiates a block correlation operation for each value of the pitch period. The hardware does a series of correlations, incrementing the address

Table 2.1: Data Flow in TDHS-ARC Supervised by the Control Center

SOURCE	INPUT	FUNCTION/OPERATION	OUTPUT	DESTINATION
XMT				
A/D	S_n, S_{n+1}	Save Pass for clipping	S_n, S_{n+1} S_n, S_{n+1}	Xmt. Mem. NEC 7720
NEC 7720	$[S_n], [S_{n+1}]$	Save	$[S_n], [S_{n+1}]$	Clip Mem.
Xmt. Mem.	S_k, S_{k-t}, T	Pass for TDHS, ARC	S_k, S_{k-t}, T	NEC 7720
NEC 7720	Q_k	Pass	Q_k	Digital Interface
RCV				
Digital Interface	Q_k	Pass for ARC ⁻¹ , Clipping	Q_k	NEC 7720
NEC 7720	$\hat{S}_k$	Save	$\hat{S}_k$	Rev. Mem.
	$[\hat{S}_k]$	Save	$[\hat{S}_k]$	Clip. Mem.
Rev. Mem.	$\hat{S}_n, \hat{S}_{n-t}, T$ $\hat{S}_{n+1}, \hat{S}_{n+1-t}, T$	Pass for TDHE	S_n, S_{n-t}, T S_{n+1}, S_{n+1-t}, T	NEC 7720
NEC 7720	S_n S_{n+1}	Pass	S_n S_{n+1}	D/A

Table 2.1: Data Flow in TDHS-ARC Supervised by the Control Center

over the desired block of clipped speech and generates a histogram value. The microprocessor reads this histogram value to update its estimate of the pitch period and re-initiates another block correlation operation. When the pitch period is finally determined, the value is passed to the control center. The pitch extraction section then waits to be instructed to find another pitch period.

CHAPTER 3

INSTALLATION AND INTERFACE

The TDHS-ARC hardware system consists of a single wire-wrapped board in a self-contained chassis. It can operate stand-alone in the digital loopback configuration; or in full duplex with another TDHS-ARC system (or a real-time TDHS-ARC simulator) over a synchronous RS-232 interface at 16 kbs. The TDHS-ARC system requires no setup or calibration prior to operation.

3.1 Installation

The unit should be checked for physical damage before it is plugged in. The electrical plug is a standard 3-prong appliance plug and uses 120V at 60 Hz. The system consumes about 15 watts of power. When the power is turned on, the "power-on" led displays that proper voltages are being supplied to system. Refer to the Operating Instructions and the Maintenance sections for further details.

3.2 Interface

The TDHS-ARC has three external interfaces: a telephone interface (analog), a tape interface (analog), and an RS-232 interface (digital).

1. Telephone Interface: The interface connector is a 4 prong Switchcraft A4M jack located on the front panel. It interfaces to a dynamic microphone and a conventional telephone receiver. The connections are shown in Table 3.1.
2. Tape Interface: The tape interface located on the back panel has two stereo coaxial connectors for input and output. The interface has a dynamic range of ± 5 volts. The input impedance is $2K\Omega$ and the output load impedance should be greater than $2K\Omega$. The connections are shown in Table 3.2.

Table 3.1: Handset Jack Connections

Pin #	Signal
1	Microphone In
2	Microphone Return
3	Speaker Out
4	Speaker Return

Table 3.2: Tape Input/Output Connections

Connector	Signal
Tip	Signal
Ring	Return
Sleeve	Shield

3. RS-232 Interface: The digital interface is a standard synchronous RS-232 interface. It is located on the back panel. The five signals used in the interface are shown in Table 3.3. To connect two TDHS-ARC systems, the cable should be configured as a null modem, Fig. 3.1.

Table 3.3: RS-232 Signal Connections

Pin #	Signal
2	Xmt Data
3	Rcv Data
7	Signal Ground
15	Xmt Clock
17	Rcv Clock

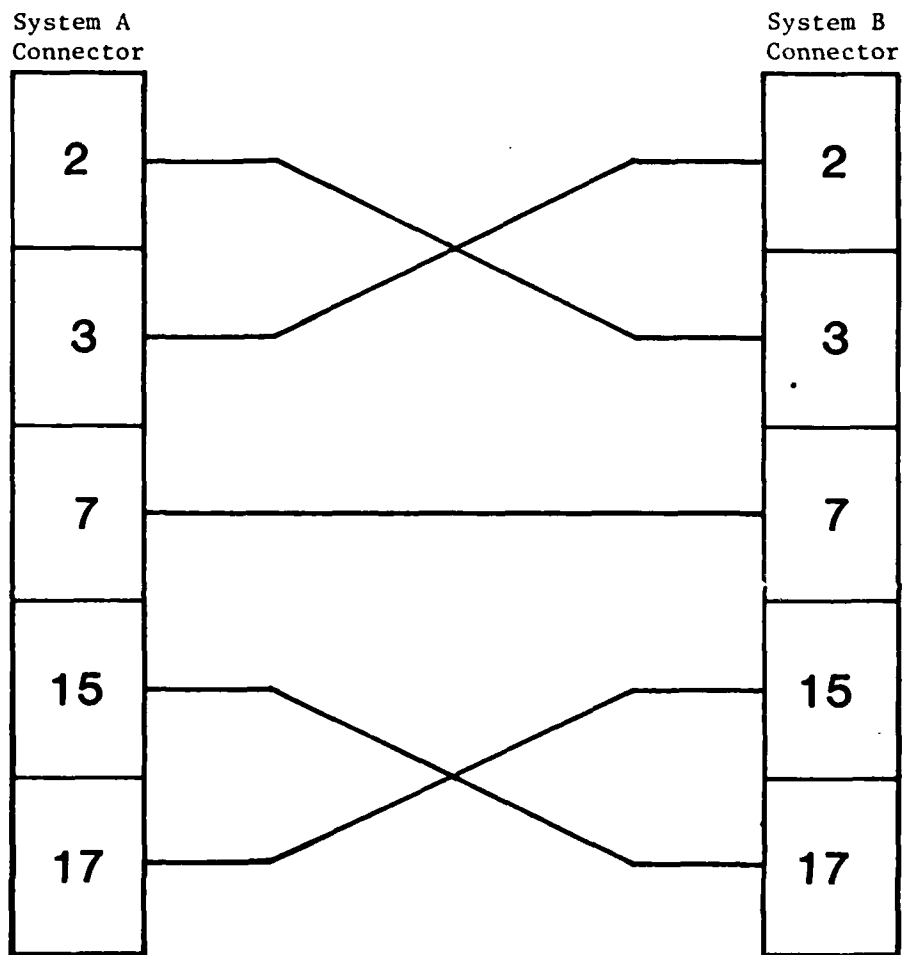


Fig. 3.1: Null Modem Connection for TDHS-ARC

CHAPTER 4

OPERATING INSTRUCTIONS

4.1 Introduction

This chapter describes how to operate the TDHS-ARC hardware system. The system encodes an analog signal into a 16 kbs data stream using a combination of Time Domain Harmonic Scaling and Adaptive Residual Coding. The hardware can operate in a single system configuration by looping the digital data stream from the transmitter back to the receiver in the same system; or it can operate in full duplex by interfacing with another TDHS-ARC system over a synchronous RS-232 channel.

The TDHS-ARC system requires no set-up or calibration prior to operation. However, it has several modes of operation which can be selected via front panel switches. In addition, the state of the system is displayed on the front panel LED's. Figure 4.1 shows the configuration of the front panel and Figure 4.2 shows the back panel. Figure 4.3 shows the system layout.

4.2 Switches & Operation

Power-On: The power switch turns the system on. The LED marked "power on" displays that all the power supplies in the system are functional. If the LED stays off, one or more of the voltages are inoperational.

Reset: The system resets when the power is turned on. In addition, it can be reset whenever necessary by pressing the "reset" switch.

Input Select: The system interfaces with two analog inputs: the telephone handset located on the front panel, and a tape i/o located on the back panel. The handset input is a dynamic

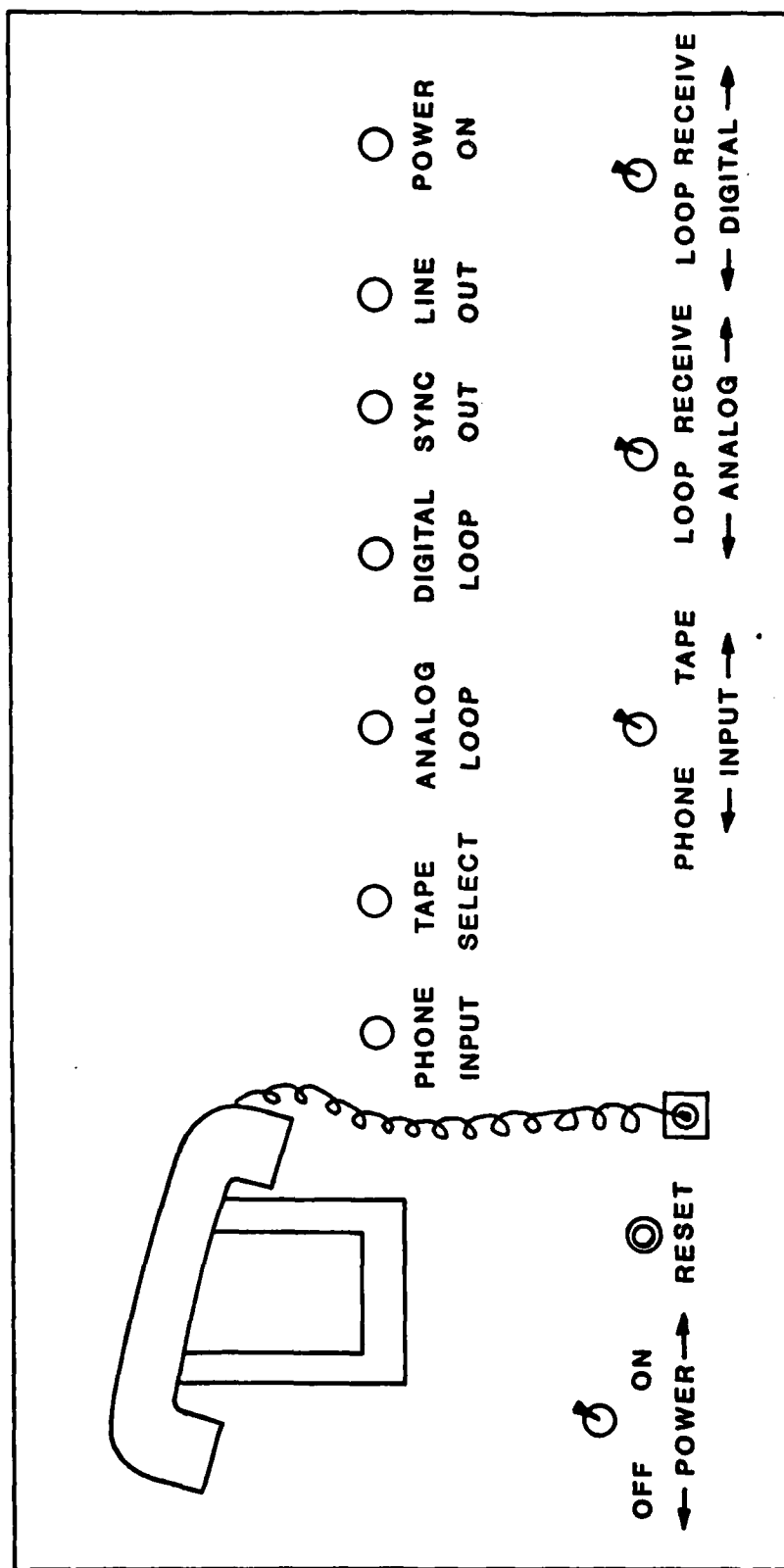


Fig. 4.1: Front Panel Layout for TDHS-ARC System

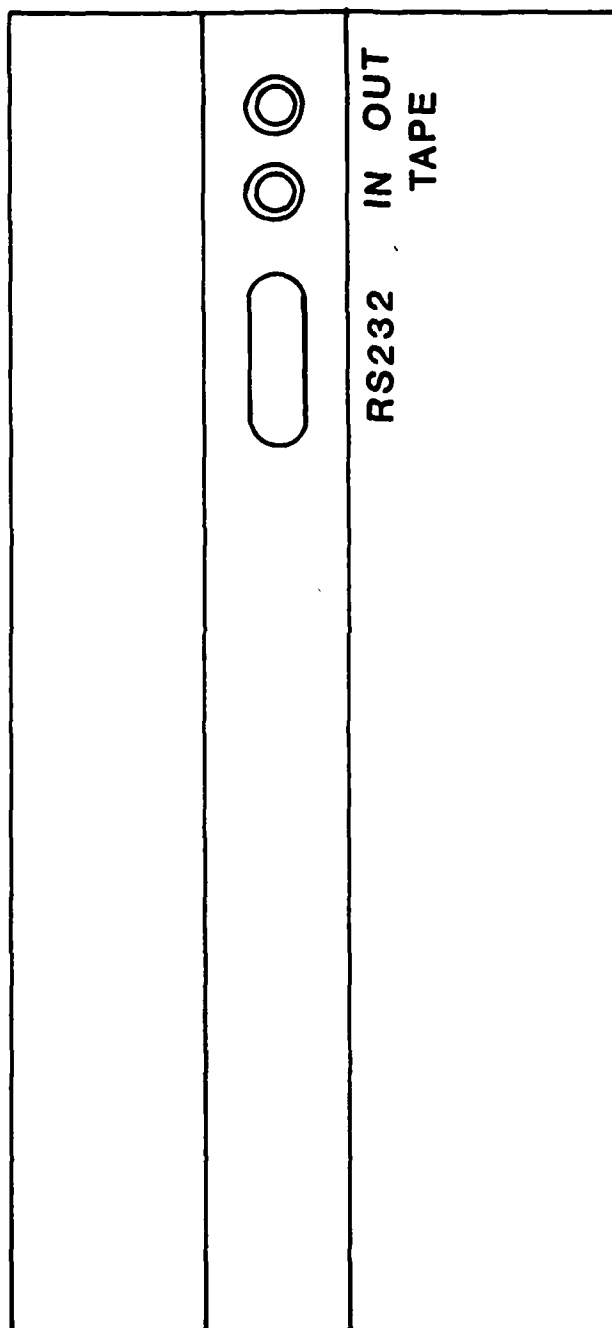


Fig. 4.2: Back Panel Layout for TDHS-ARC System

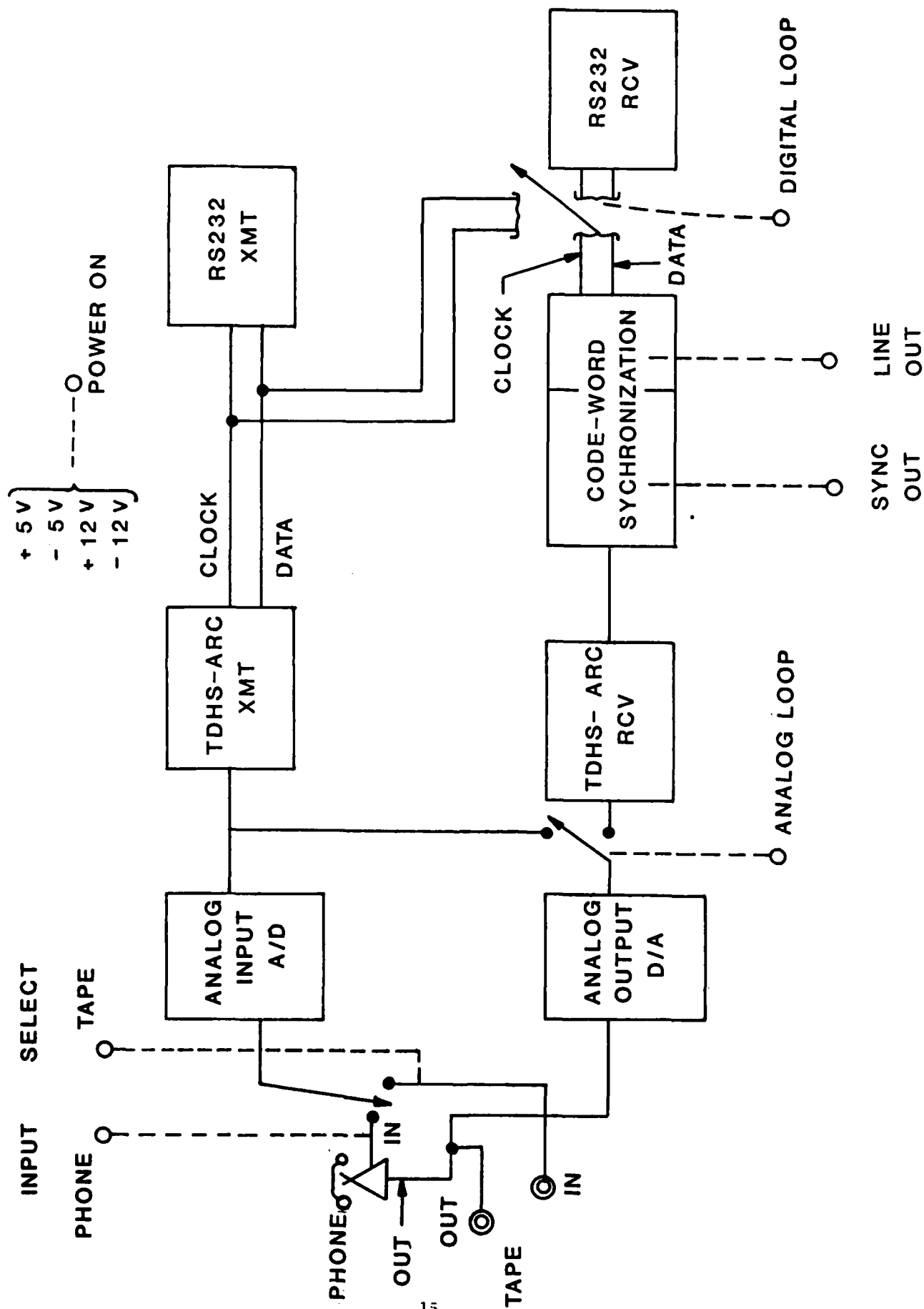


Fig. 4.3: Functional Block Diagram of TDHS-ARC

microphone at approximately 24 TLP, and the output is a conventional receiver at 10 TLP. The tape input and output are set at approximately -6 TLP, ± 5 v peak.

The "input select" switch on the front panel selects one of the two inputs and the LED's display which input is selected. Both analog outputs are always active, immaterial of which input is selected as shown in Fig. 4.3.

Analog Loop-Receive: The TDHS-ARC system uses a standard telephone 8-bit μ -law codec for analog-to-digital conversion. The performance of the analog-to-digital and the digital-to-analog conversion system can be monitored, for reference comparison with processed signals, in the "loop" position of the switch marked "analog". The LED marked "analog loop" displays this state. Even though the analog signal is looped back to the analog output stage, the TDHS-ARC transmitter and receiver continue to operate normally, Fig. 4.3.

In the "receive" position, the output of the receiver section of TDHS-ARC is enabled to the analog output stage, Fig. 4.3.

Digital Loop-Receive: The transmitter in the TDHS-ARC system encodes an analog signal into a 16 kbs data stream. The receiver in the system takes a 16 kbs data stream and decodes it into an analog signal.

In the "loop" position of the "digital" switch, the clock and data stream out of the transmitter are looped back to the receiver, Fig. 4.3 for a single system operation of TDHS-ARC. The LED marked "digital loop" displays the loopback state.

In the "receive" position, the TDHS-ARC receiver gets the data and clock from the RS-232 interface, Fig. 4.3, enabling two systems to interact with each other in full duplex. In either position, the output of the transmitter is always enabled to the RS-232 port.

Sync-Out Display: The receiver must acquire code-word synchronization on the serial data it receives before it can start processing it properly. The "sync-out" display indicates that the receiver is out of code-word synchronization. Once synchronization is achieved, the LED turns off.

If synchronization is lost due to any reason, in general, the receiver recognizes the loss of sync and tries to re-acquire synchronization. Also, resetting the system forces the receiver to re-synchronize.

Line-Out Display: In the full-duplex mode, the receiver gets the clock and data from the other system. The receiver monitors the received clock, to ensure it can clock in the serial data. A break in the received clock causes the "line-out" display to illuminate. This automatically re-initiates code-word synchronization.

CHAPTER 5

CONCLUSIONS

It has been demonstrated that a reasonable size single-board full-duplex implementation of the TDHS-ARC algorithm at 16 kbs is possible and produces toll quality speech. The design of the prototype system delivered as part of this contract emphasized flexibility rather than hardware minimization. It is believed that considerable hardware optimization can be achieved (30%) in a redesign of this system for production using off-the-shelf LSI components. If a custom LSI TDHS-ARC transceiver chip is developed for production, the savings in terms of power, size and cost can be dramatic (50 to 80%) without loss in performance.

APPENDIX A

MAINTENANCE: SCHEMATICS AND COMPONENTS

This section presents the block diagram and the schematics for the TDHS-ARC hardware system. Figure A.1 shows the physical layout of the board. The TDHS-ARC circuit is segmented into 6 functional sections: analog interface, digital interface, clock and timing, control center, computation center, and pitch extraction. The following figures and tables show the detailed schematics for each of the segments and their parts lists. Figure A.8 is the schematic for the circuit on the back plane. Figure A.9 shows the edge connector.

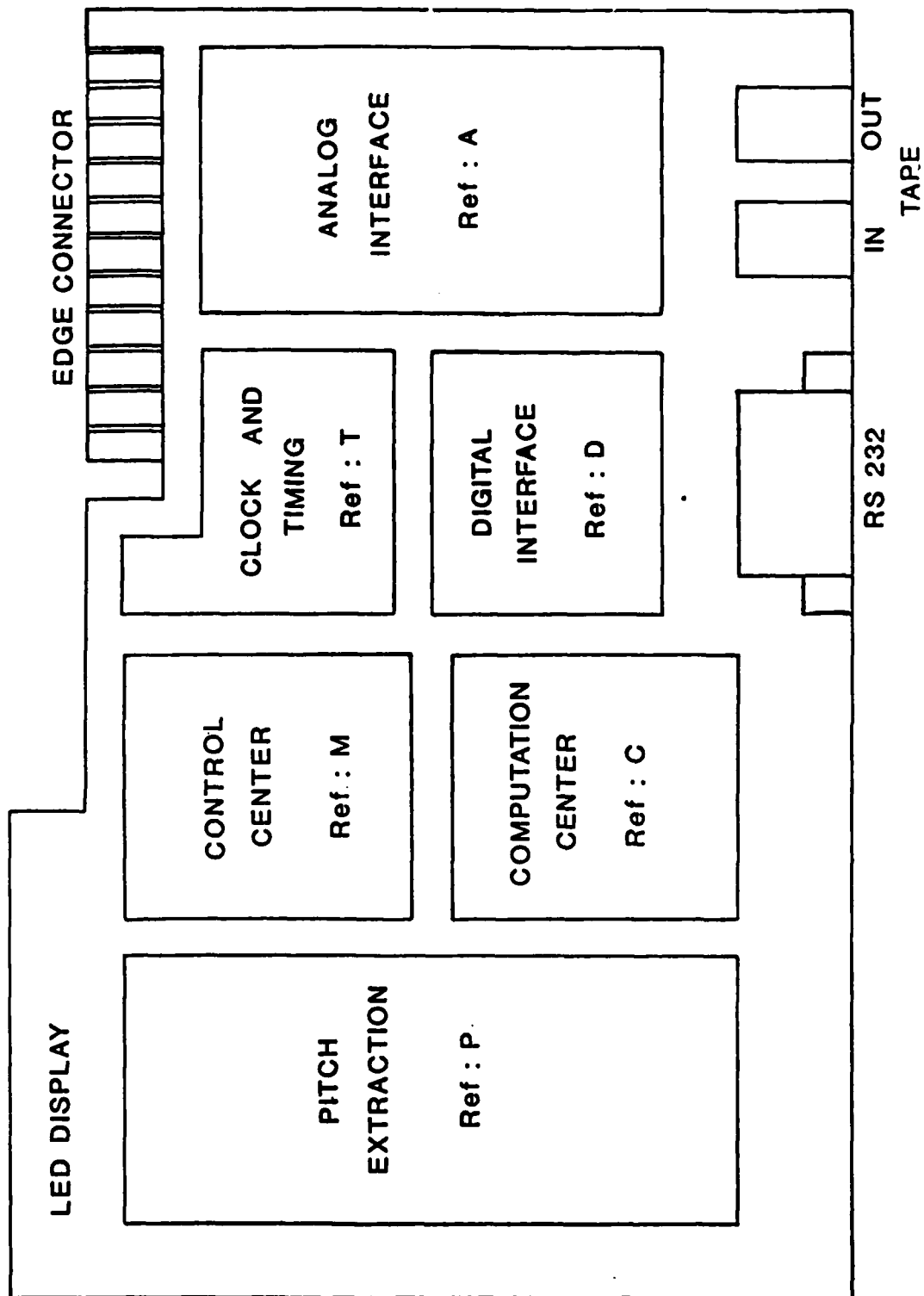
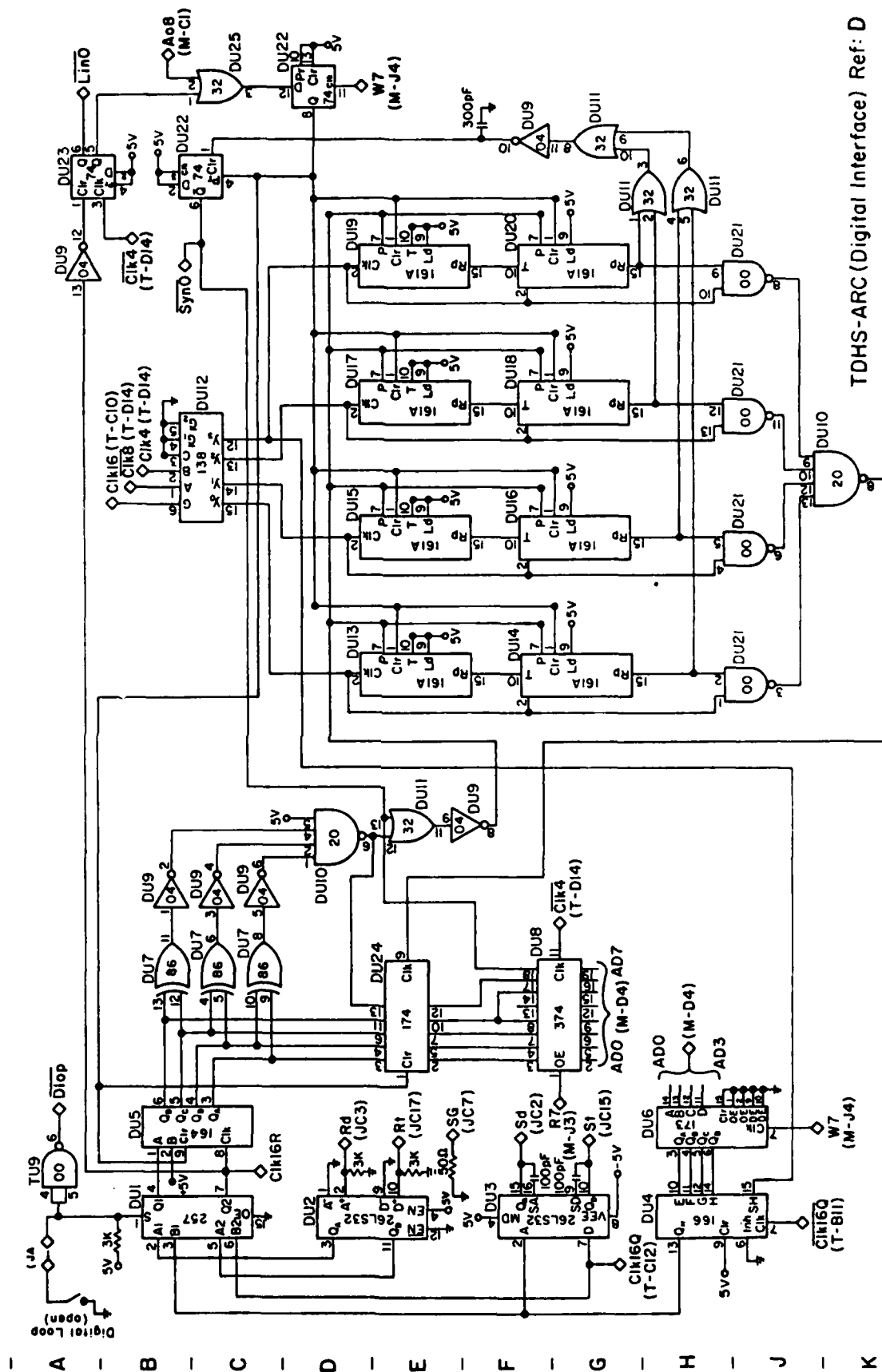


Fig. A.1: Layout of Circuit Segments in TDHS-ARC

Table A.1: Parts List for the Analog Interface

[illegible]



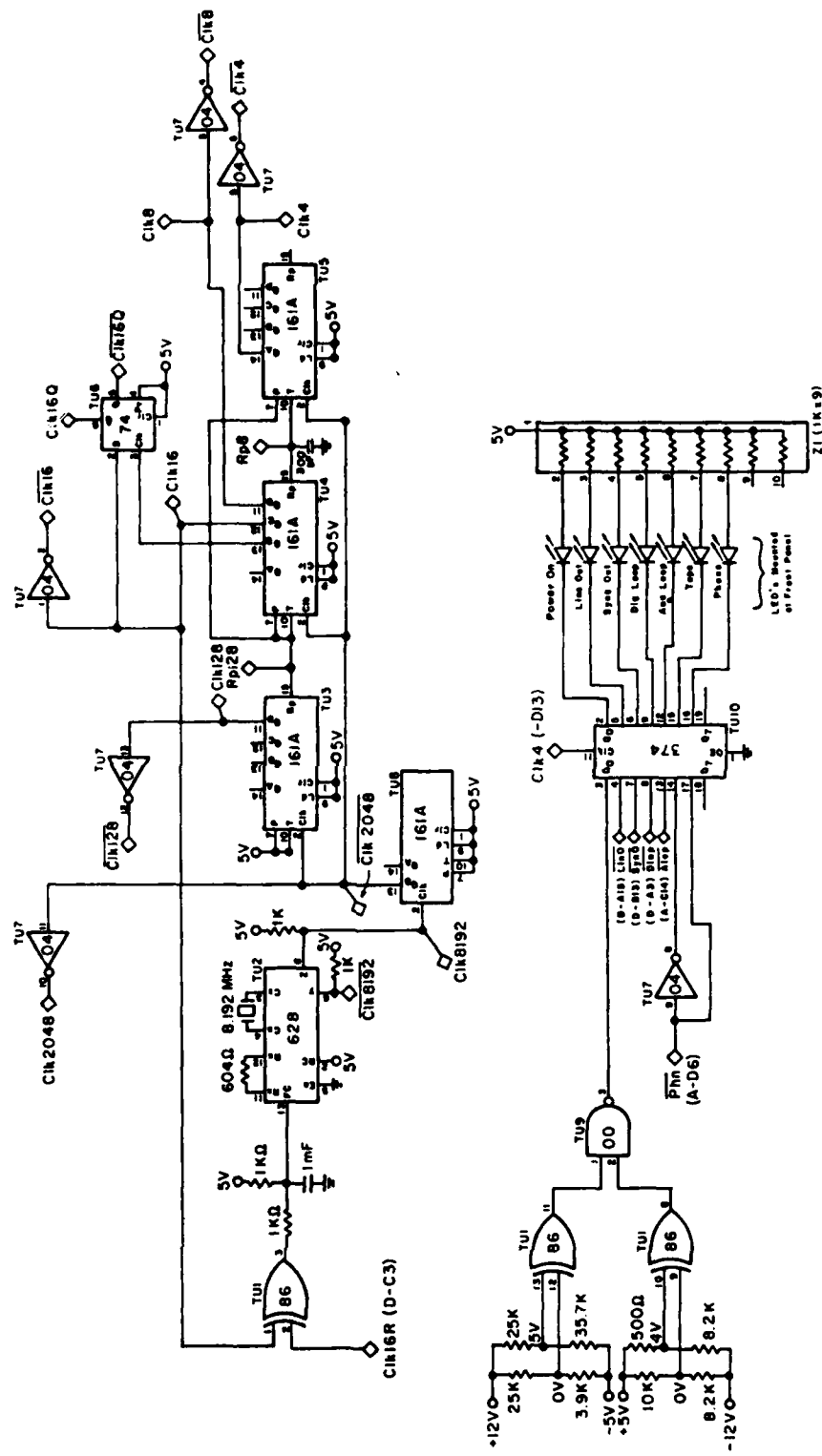
TDHS-ARC (Digital Interface) Ref: D

Fi 1.3 cher c f he [ta] - ter

Table A.2: Parts List for the Digital Interface

PART #	DESCRIPTION	USAGE	+12v	-12v	+5v	GND
26LS30	RS423 Driver	DU2		8(-5v)	1	5
26LS32	RS423 Receiver	DU3			16	8
74LS00	Quad Nand	DU21			14	7
74LS04	Hex Inverter	DU9			14	7
74LS20	Dual 4 Input Nand	DU10			14	7
74LS32	Quad Or	DU11,DU25			14	7
74LS74	Dual Flip Flop	DU22,DU23			14	7
74LS86	Quad Ex Or	DU7			14	7
74LS138	3-8 Decoder	DU12			16	8
74LS161A	4 bit Counter	DU13,DU14,DU15,DU16, DU17,DU18,DU19,DU20			16	8
74LS164	S/P Converter	DU5			14	7
74LS166	P/S Converter	DU4			16	8
74LS173	Quad Flip Flop	DU6			16	8
74LS174	Hex Flip Flop	DU24			16	8
74LS257	2-1 Data Select	DU1			16	8
74LS374	Octal Bus Driver	DU8			20	10

A B C D E F G H J K



TDHS-ARC (Clock and Timing) Ref: T

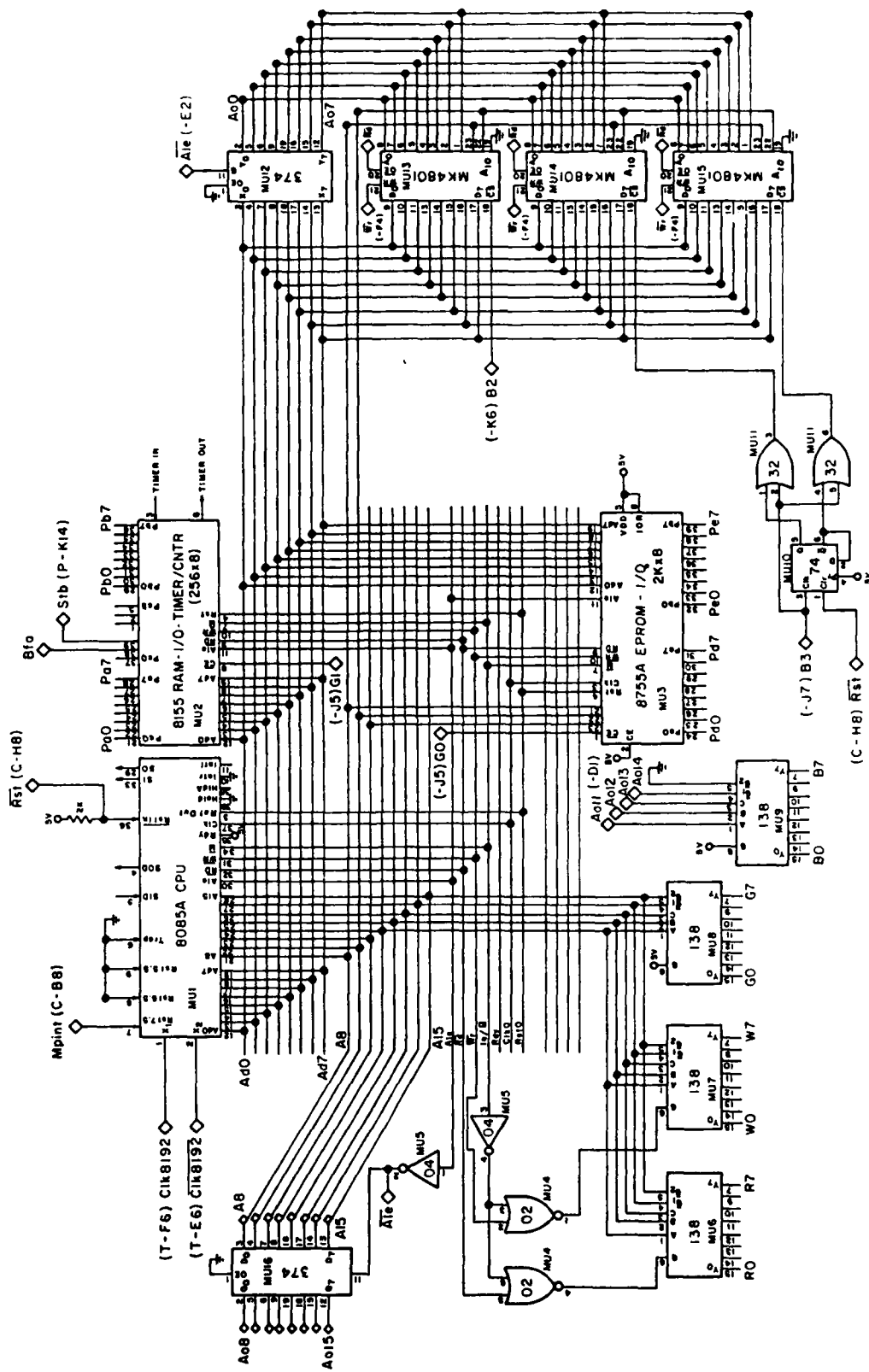
Fig. A.4: Schematic for the Clock and Timing

+ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Table A.3: Parts List for the Clock and Timing

[illegible]

A - B - C - D - E - F - G - H - J - K



TDHS-ARC (Control Center) Ref: M

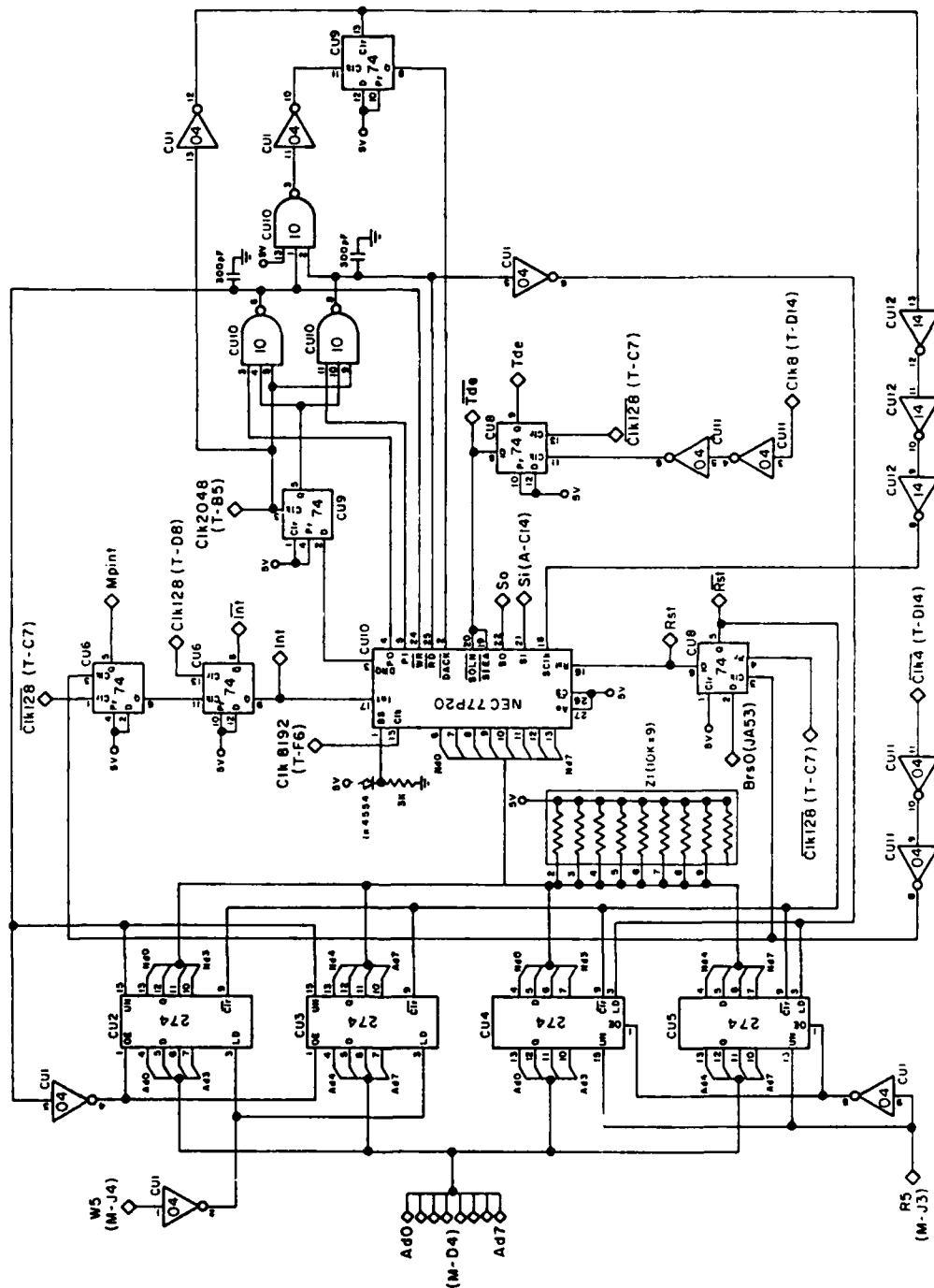
Fig A.5- Schematic for the Control Center

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Table A.4: Parts List for the Control Center

[illegible]

A B C D E F G H J K



TDHS-ARC (Computation Center) Ref: C

Fig. A.6. Schematic for the Computation Center

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Table A.5: Parts List for the Computation Center

[illegible]



13

Table A.6: Parts List for the Pitch Extraction

[illegible]

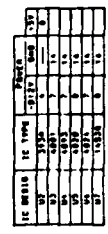
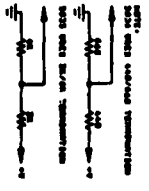
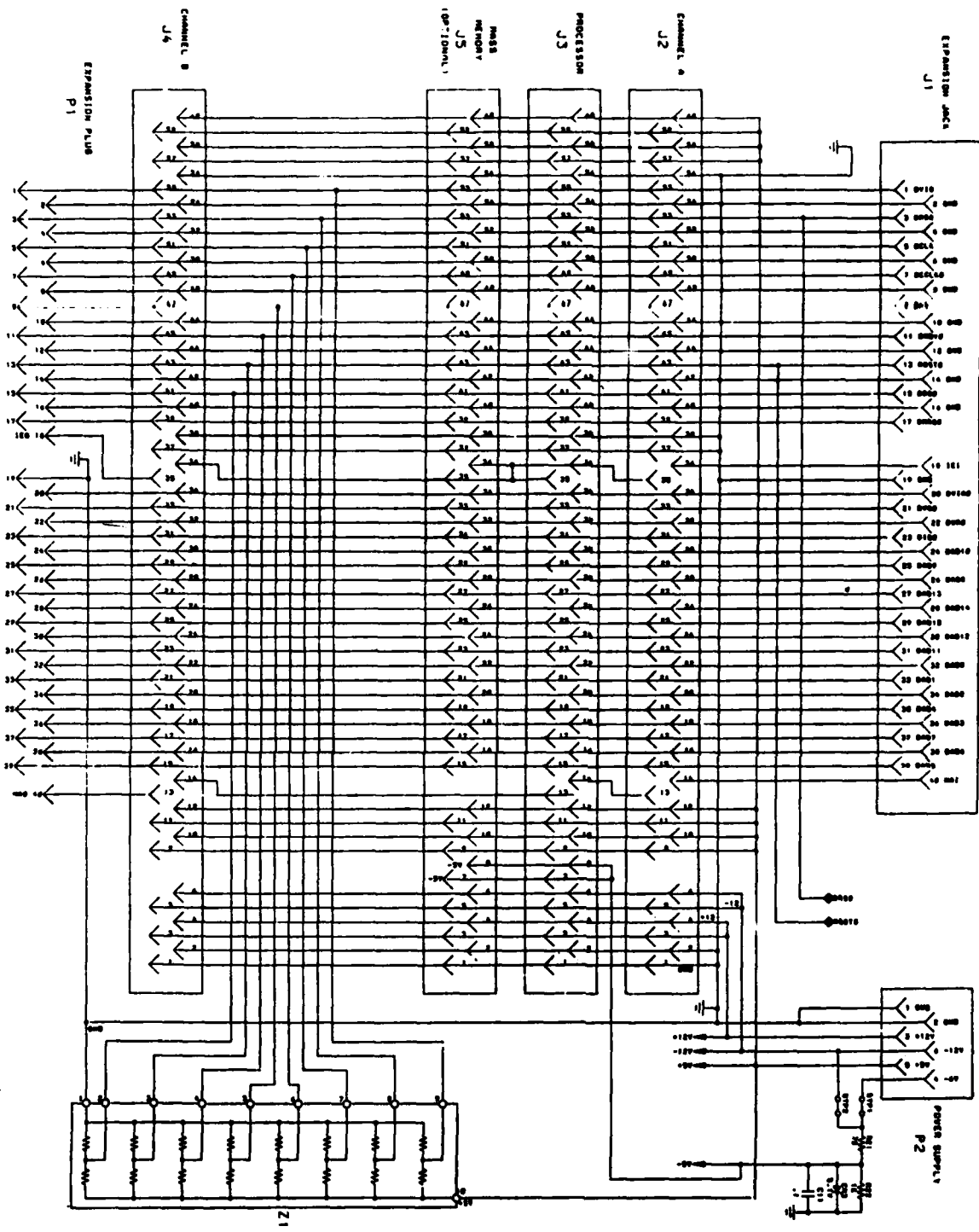
[illegible]

Fig. A.8: Schematic for the Back Plane



ITEM	DESCRIPTION	QTY	UNIT	PRICE	TOTAL
1	EXPANSION JACK	1	PCB	1.00	1.00
2	CHANNEL A	1	PCB	1.00	1.00
3	CHANNEL B	1	PCB	1.00	1.00
4	EXPANSION PLUG	1	PCB	1.00	1.00
5	POWER SUPPLY	1	PCB	1.00	1.00
6	Z1	1	PCB	1.00	1.00
7	EXPANSION PLUG	1	PCB	1.00	1.00
8	EXPANSION PLUG	1	PCB	1.00	1.00
9	EXPANSION PLUG	1	PCB	1.00	1.00
10	EXPANSION PLUG	1	PCB	1.00	1.00
11	EXPANSION PLUG	1	PCB	1.00	1.00
12	EXPANSION PLUG	1	PCB	1.00	1.00
13	EXPANSION PLUG	1	PCB	1.00	1.00
14	EXPANSION PLUG	1	PCB	1.00	1.00
15	EXPANSION PLUG	1	PCB	1.00	1.00
16	EXPANSION PLUG	1	PCB	1.00	1.00
17	EXPANSION PLUG	1	PCB	1.00	1.00
18	EXPANSION PLUG	1	PCB	1.00	1.00
19	EXPANSION PLUG	1	PCB	1.00	1.00
20	EXPANSION PLUG	1	PCB	1.00	1.00
21	EXPANSION PLUG	1	PCB	1.00	1.00
22	EXPANSION PLUG	1	PCB	1.00	1.00
23	EXPANSION PLUG	1	PCB	1.00	1.00
24	EXPANSION PLUG	1	PCB	1.00	1.00
25	EXPANSION PLUG	1	PCB	1.00	1.00
26	EXPANSION PLUG	1	PCB	1.00	1.00
27	EXPANSION PLUG	1	PCB	1.00	1.00
28	EXPANSION PLUG	1	PCB	1.00	1.00
29	EXPANSION PLUG	1	PCB	1.00	1.00
30	EXPANSION PLUG	1	PCB	1.00	1.00
31	EXPANSION PLUG	1	PCB	1.00	1.00
32	EXPANSION PLUG	1	PCB	1.00	1.00

Fig. 4: J4 Co for

APPENDIX B

REAL-TIME SOFTWARE

The TDHS-ARC hardware has 3 processors on board. This section provides the listings of the programs in each of the processors. The first program, TARC, is contained in the 8085 microprocessor in the control center. The second program, PEXTR, is contained in the 8085 microprocessor in the pitch extraction segment. The third and fourth programs are contained in the NEC 7720 signal processing chip in the computation center. TDHSI4 is the instruction software and TDH5D4 is the data software for the signal processing chip. Each of the programs has comments in the right margin to make it comprehensible.

B.1 Control Center Software: TARC

The microprocessor in the control center performs three tasks. It keeps track of the addresses in the transmitter and receiver buffers and manages data transfers through the TDHS-ARC system, asks for pitch period extraction whenever necessary, and monitors the synchronization on the incoming data stream. The program is interrupt driven and the process cycle is repeated every 4 KHz. This corresponds to 2 samples of the input speech.

The following flowchart describes the functions being performed in the control processor. In the first section, the program does all the necessary data transfers and address management for the transmitter. It then monitors the synchronization on the received quantizer levels. Finally, it does the data transfers and address management for the receiver. The initialization section sets up the microprocessor, address, data, etc., for subsequent processing loops.

The flowchart uses the following symbols:

S_n - Incoming transmitter sample

$[S_n]$ - Clipped value of transmitter sample
 S_k - Transmitter sample used in TDHC
 T_x - Transmitter pitch period
 C_x - Count of the samples in the transmitter pitch period
 $P(S_n)$ - Address pointer for S_n
 R_n - Incoming receiver sample
 $[R_n]$ - Clipped value of receiver sample
 R_k - Receiver sample used in TDHE
 T_r - Receiver pitch period
 C_r - Count of the samples in the receiver pitch period
 $P(R_n)$ - Address pointer for R_n
 $Pzcod$ - Previous zero code
 $Zerr$ - # of consecutive errors in the received zero code
 $Zcod$ - Zero code
 $Xcod$ - Transmitted quantizer code
 $Rcod$ - Received quantizer code
 $Tocnt$ - Time Out Count
 $Xmem$ - Transmitter memory, contains 8 bit μ -law speech samples
 $Rmem$ - Receiver memory, contains 16 bit linear samples of time domain compressed speech
 $Pmem$ - Pitch extraction memory, contains center clipped speech, 2 bits per sample, for the transmitter and receiver
 Spc - I/O port on the Computation Center
 Dio - Digital Interface

The address and data variables that are saved in the microprocessor are listed below. All the other variables are derived from these.

$P(S_k)$, $P([R_n])$, T_x , T_r , C_x , C_r , $Pzcod$, $Zerr$, $Tocnt$

Transmitter:

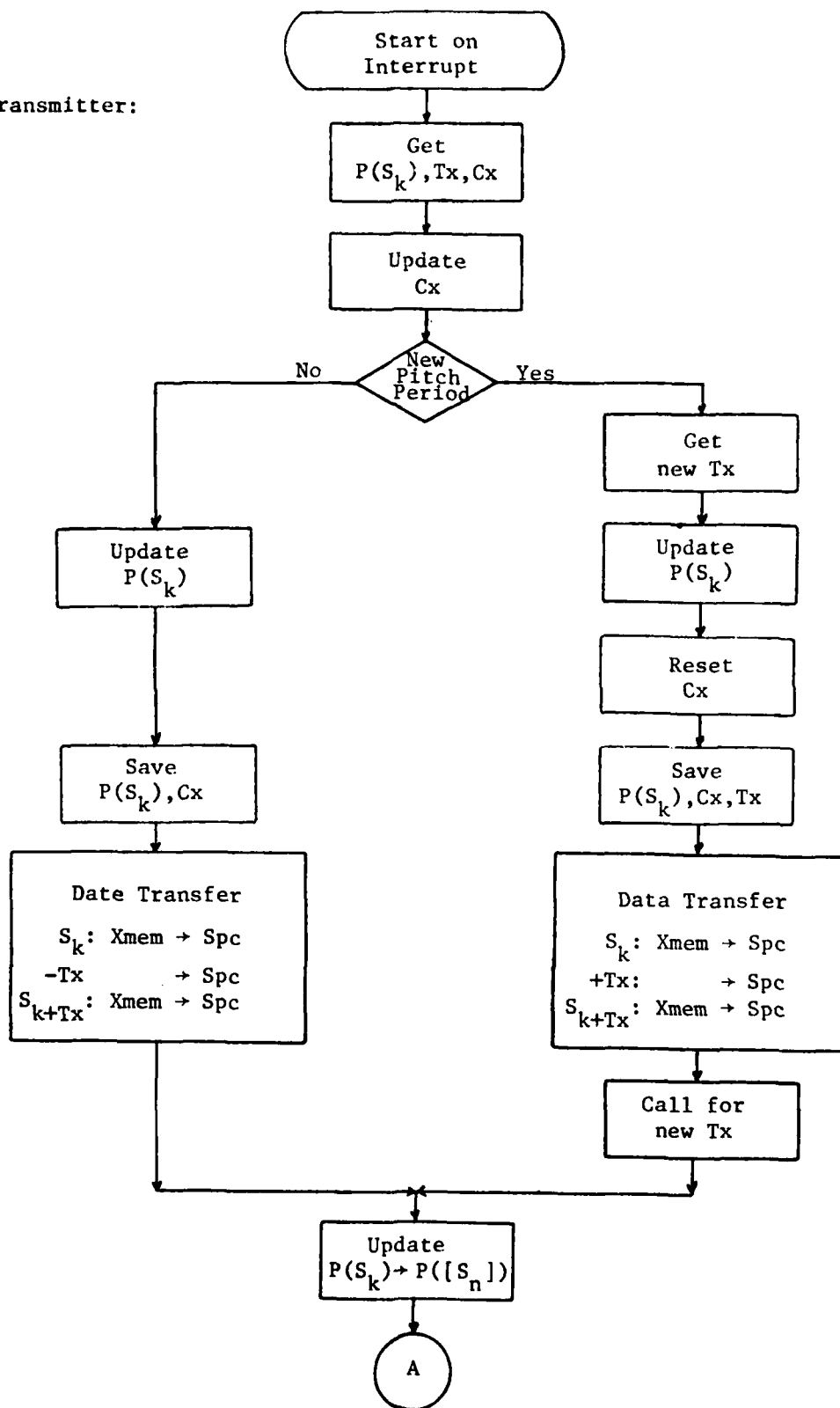


Fig. B.1: Flow Chart for the Control Program, TARC

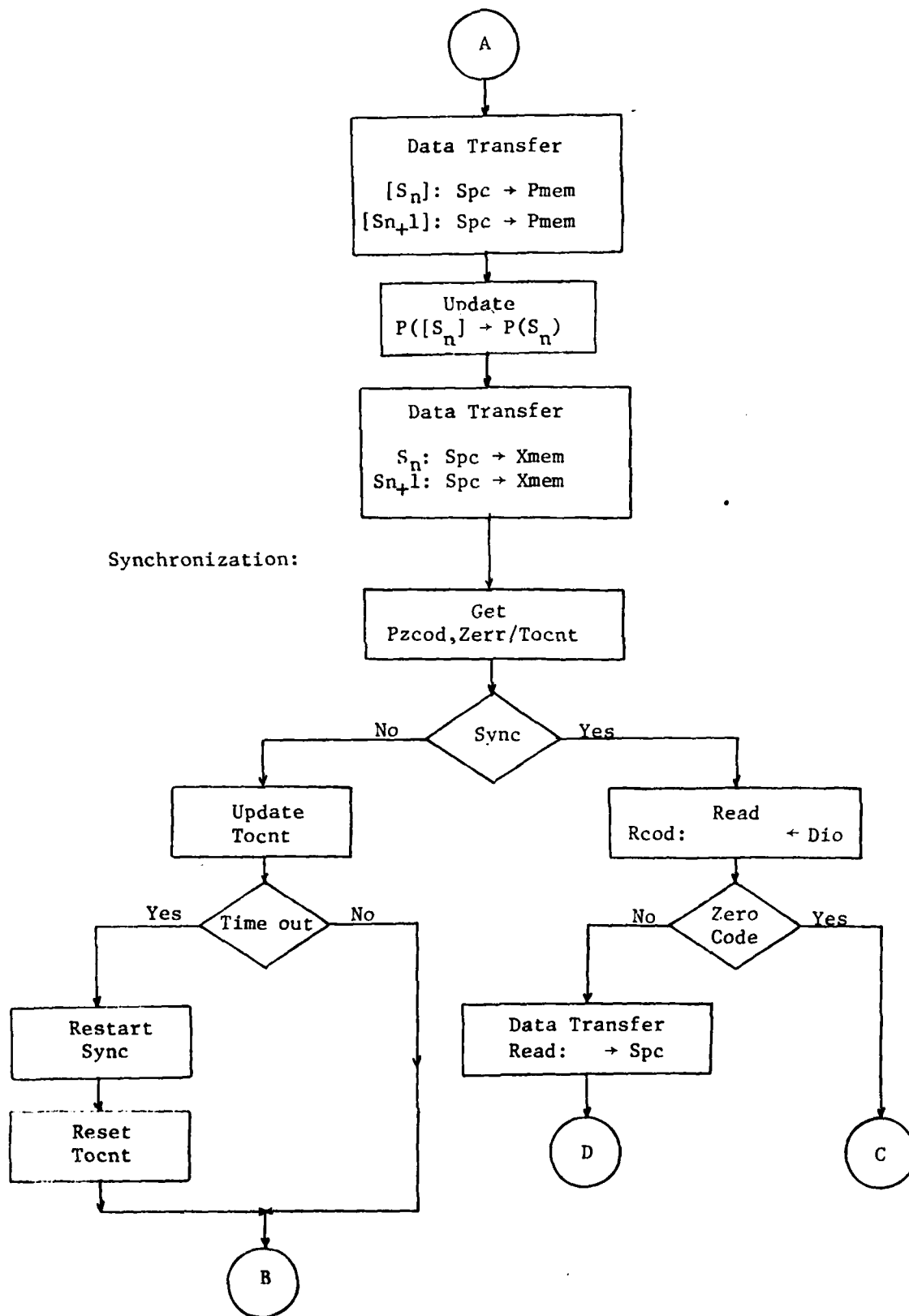


Fig. B.1 (contd): Flow Chart for the Control Program, TARC

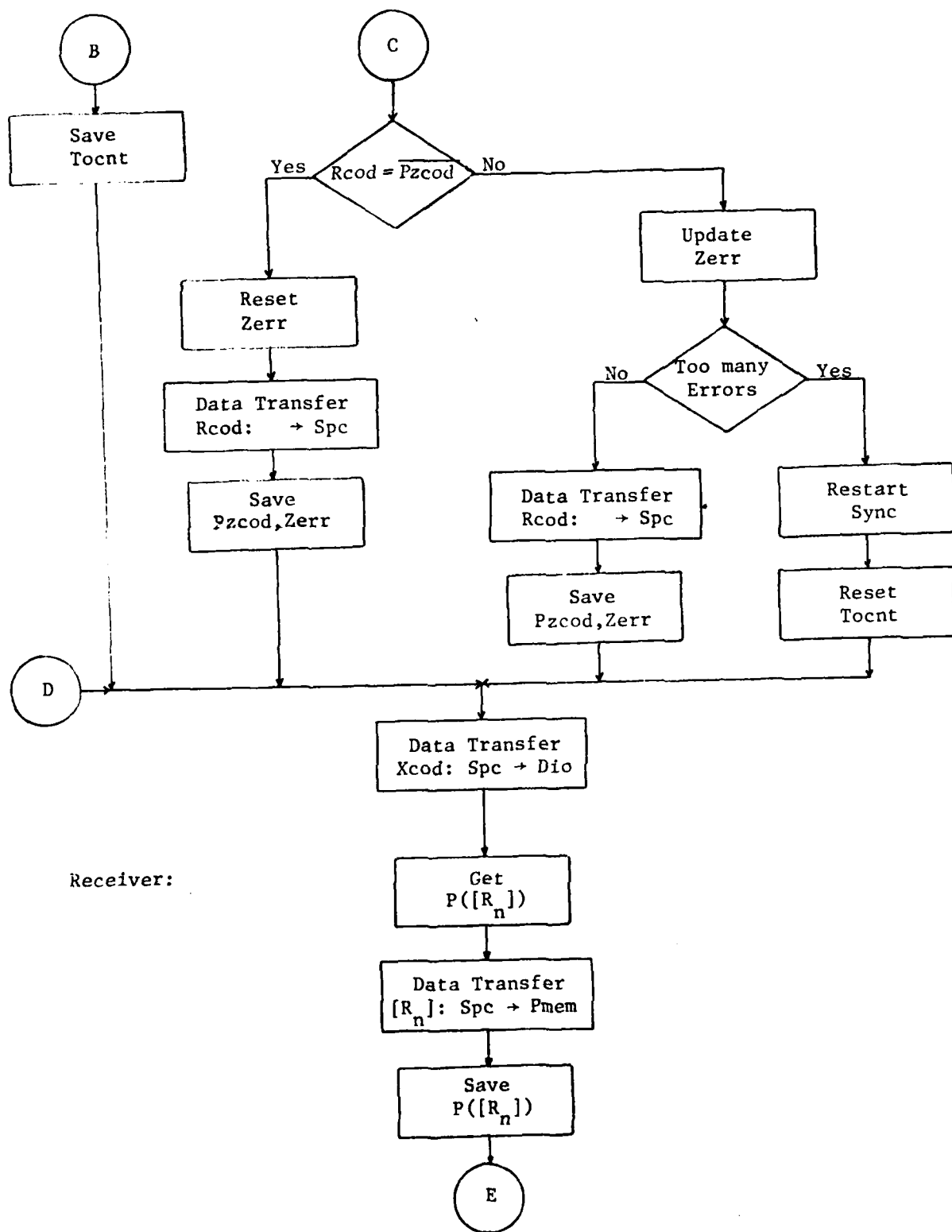


Fig. B.1 (contd): Flow Chart for the Control Program, TARC

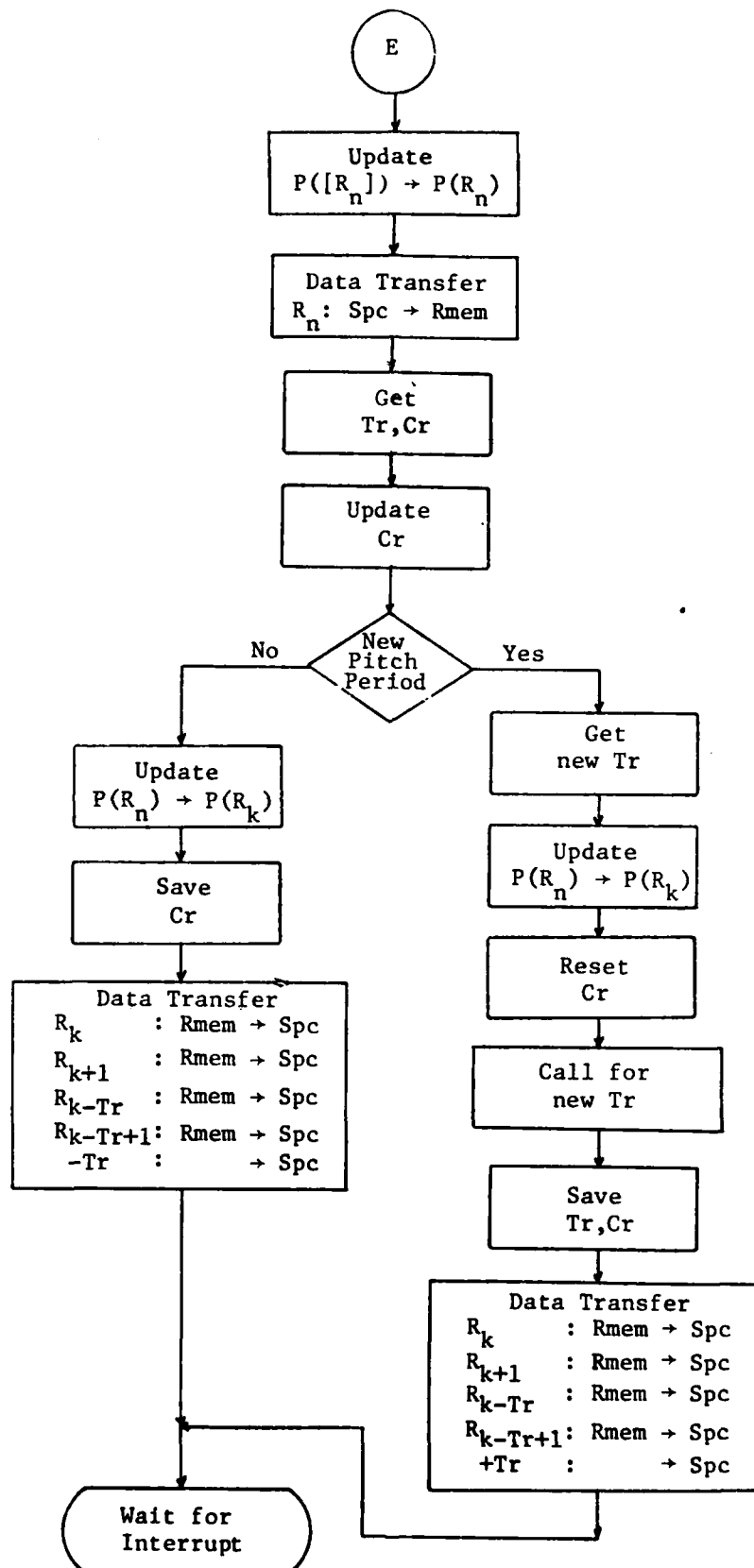


Fig. B.1 (contd): Flow Chart for the Control Program, TARC

LOC	OBJ	LINE	SOURCE STATEMENT
		1 ;	This program for control and data management for
		2 ;	TDHS-ARC. Goes in micro 1.
		3 ;	
		4	NAME MAIN
0028		5 ADNEC EQU 28H	;Port for NEC i/o
0039		6 RSYNC EQU 039H	;Port to reset sync
0038		7 QCODE EQU 038H	;Port for code i/o
0030		8 PTEN EQU 030H	;Temp port, for status
000B		9 ENINT EQU 00BH	;Enable interrupt 7.5
0080		10 RNINT EQU 080H	;Reset interrupt 7.5
08FF		11 SPTR EQU 08FFH	;Stack pointer
0800		12 MMST EQU 0800H	;beginning of memory
1000		13 XMST EQU 1000H	;top of xmt mem
1800		14 RMST EQU 1800H	;top of rcv mem
2000		15 PMST EQU 2000H	;top of pch mem
2000		16 PMSTX EQU 2000H	;top of pch mem, xmt
A000		17 PMSTR EQU 0A000H	;top of pch mem, rcv
0037		18 STP81 EQU 037H	;status for 8155
0008		19 ADS81 EQU 08H	;8155 status port
0009		20 AD81A EQU 09H	;8155 port A
000A		21 AD81B EQU 0AH	;8155 port B
000B		22 AD81C EQU 0BH	;8155 port C
0000		23 STP87A EQU 00H	;status for 8755 A, inp
0000		24 STP87B EQU 00H	;status for 8755 B, inp
0002		25 ADS87A EQU 02H	;8755 A status port
0003		26 ADS87B EQU 03H	;8755 B status port
0000		27 AD87A EQU 00H	;8755 port A
0001		28 AD87B EQU 01H	;8755 port B
0804		29 PTXSN EQU 0804H	;Sn pointer
0808		30 TXMT EQU 0808H	;pointer Txmt
080C		31 CTXMT EQU 080CH	;pointer xmt counter
0810		32 PTCWS EQU 0810H	;pointer code word sync
0814		33 PTRRN EQU 0814H	;pointer Rn
0818		34 TRCV EQU 0818H	;pointer Trcv
081C		35 CTRCV EQU 081CH	;pointer rcv counter
00FB		36 MSKXA EQU 0FBH	;msk out b10
00FB		37 MSKRA EQU 0FBH	;msk out b10
0002		38 MSKRB EQU 02H	;msk invert b9
00B2		39 XXP EQU 0B2H	;msk xmt -> pch(rcv bits), invert b9
00B0		40 XPX EQU 0B0H	;msk pch -> xmt
0038		41 XPR EQU 038H	;msk pch(xmt bits) -> rcv
0104		42 ZICOD0 EQU 0104H	;cod +1(0000). err cnt = 4

```

FF04          43 ZICOD1- EQU    0FF04H ;cod -1(1111), err cnt = 4
2000          44 SYNCNT EQU    2000H  ;time out cnt for sync acquisition
0008          45 ZCODE1 EQU    08H   ;zero code: 1000
0007          46 ZCODE0 EQU    07H   ;zero code: 0111
          47 ;
0000          48 ORG      0000H
          49 START:
0000 C3D801   50 RST00: JMP     INIT   ;Reset begins at INIT
          51 ;
          52 ;Main loop begins at reset 7.5
003C          53 ORG      003CH
003C 310408   54 RST75: LXI     SP,PTXSN ;P(P(Sn))

```

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 2

LOC	OBJ	LINE	SOURCE STATEMENT
		55 ;	
003F	2A0808	56	LHLD TXMT ;xmt T
0042	3A0C08	57	LDA CTXMT ;xmt cntr old
0045	44	58	MOV B,H ;msb = 00
0046	3C	59	INR A
0047	BD	60	CMP L ;compare with T
0048	CA6600	61	JZ XPNEW ; new pitch period
		62 ;	
		63 ;	
		64 ;	
		65	XPOLD: STA CTXMT ;save current counter
004B	320C08	66	MOV C,A ;save in C
004E	4F	67 ;	
004F	EB	68	XCHG ;DE=xmt T
0050	E1	69	POP H ;old P(Sn)
0051	23	70	INX H
0052	3EF8	71	MVI A,MSKXA ;msk out b10
0054	A4	72	ANA H
0055	67	73	MOV H,A ;HL=P(Sn)
0056	E5	74	PUSH H
		75 ;	
0057	7E	76	MOV A,M
0058	D328	77	OUT ADNEC ;Sn xmem -> NEC
		78 ;	
005A	3EFF	79	MVI A,OFFH
005C	D328	80	OUT ADNEC ;T(-ve) -> NEC
005E	EB	81	XCHG
005F	19	82	DAD D ;HL=P(Sn+T) ,DE=P(Sn)
0060	7E	83	MOV A,M
0061	D328	84	OUT ADNEC ;Sn+T xmem -> NEC
0063	C38B00	85	JMP XCON1
		86 ;	

0066 3E00	87	XPNEW:	MVI	A,00H	;new counter value
0068 320C08	88		STA	CTXMT	;save
006B 4F	89		MOV	C,A	; and in C
	90				
006C EB	91		XCHG		
006D E1	92		POP	H	;HL=old P(Sn)
006E 23	93		INX	H	
006F 19	94		DAD	D	
0070 3EFB	95		MVI	A,MSKXA	;msk out b10
0072 A4	96		ANA	H	
0073 67	97		MOV	H,A	;HL=P(Sn) new pitch
0074 E5	98		PUSH	H	
	99				
0075 7E	100		MOV	A,M	
0076 D328	101		OUT	ADNEC	;Sn xmem -> NEC
	102				
0078 DB00	103		IN	ADB7A	;new Txmt
007A D328	104		OUT	ADNEC	;T (+ve) -> NEC
007C 5F	105		MOV	E,A	
007D EB	106		XCHG		
007E 220808	107		SHLD	TXMT	;save new T .
0081 19	108		DAD	D	;HL=P(Sn+T)
0082 7E	109		MOV	A,M	

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 3

LOC	OBJ	LINE	SOURCE STATEMENT
0083	D328	110	OUT ADNEC ;Sn+T xmem -> NEC
		111	;
		112	; Ask for new pitch
		113	;
0085	7C	114	MOV A,H ;MSB
0086	D30A	115	OUT ADB1B
0088	7D	116	MOV A,L ;LSB
0089	D309	117	OUT ADB1A
		118	;
008B	EB	119	XCON1: XCHG ;HL=P(Sn)
008C	09	120	DAD B ;HL=P(Sn+cntr)
008D	3EB2	121	MVI A,XXP ;msk xmem->prmem, inv b9
008F	AC	122	XRA H
0090	67	123	MOV H,A ;HL=P([Sk])
		124	;
0091	DB28	125	IN ADNEC ;get xmt bits
0093	B6	126	ORA M ;tag on rcv bits
0094	77	127	MOV M,A ;[Sk] -> prmem
		128	;
0095	23	129	INX H ;HL=P([Sk]+1)
0096	DB28	130	IN ADNEC

0098 B6	131	ORA	M	
0099 77	132	MOV	M,A	;[Sk+1] -> pmen
	133 ;			
009A 3EB0	134	MVI	A,XPX	;ask pmen->xmen
009C AC	135	XRA	H	
009D 67	136	MOV	H,A	;HL=P(Sk)
	137 ;			
009E DB28	138	IN	ADNEC	
00A0 77	139	MOV	M,A	;Sk NEC -> xmen
	140 ;			
00A1 23	141	INX	H	;HL=P(Sk+1)
00A2 DB28	142	IN	ADNEC	
00A4 77	143	MOV	M,A	;Sk+1 NEC -> xmen
	144 ;			
	145 ;			Code word synchronization
	146 ;			
	147 ;	DUSYNC:	IN	ADNEC ;*** DIAGNOSTIC CODE ***
	148 ;	OUT	ADNEC	;loop q code to rcv
	149 ;	JMP	CODCON	;***
00A5 311008	150	CUSYNC:	LXI	SP,PTCWS
00AB E1	151	POP	H	;contents: H- prev 0 code, +1(0000)
	152 ;			-1(1111)
	153 ;			L- err cnt, max = 3
00A9 DB38	154	IN	QCODE	
00AB 4F	155	MOV	C,A	;Qcode in C
00AC 07	156	RLC		; 1 - sync
	157 ;			0 - no sync
00AD DACC00	158	JC	SYNQL	;Sync, continue
	159 ;			
	160 ;			No sync
00B0 2B	161	DCX	H	;HL = time out for acq - 1
00B1 3E08	162	MVI	A,ZCODE1	
00B3 D328	163	OUT	ADNEC	;0 code to NEC
	164 ;			

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 4

LOC	OBJ	LINE	SOURCE STATEMENT
00B5 BC		165	CMP H
00B6 CAC100		166	JZ SRSTR ;too long, restart sync
		167 ;	
00B9 DB28		168	IN ADNEC
00BB D338		169	OUT QCODE ;Qcode NEC -> DIO
00BD E5		170	PUSH H
00BE C34401		171	JMP CODCON ;--- end of code sync
00C1 DB28		172	SRSTR: IN ADNEC
00C3 D339		173	OUT RSYNC ;Qcode NEC -> DIO, restart sync
		174 ;	

00C5 210020	175	LXI	H,SYNCRNT	
00C8 E5	176	PUSH	H	
00C9 C34401	177	JMP	CODCON	;--- end of code sync
	178 ;			
	179 ;	---	Sync, continue	
	180 ;			
00CC 07	181 SYNQL:	RLC		; 1 - non zero code
00CD DA3C01	182	JC	NZCOD	; 0 - zero code
	183 ;			
00D0 07	184 ZCOD:	RLC		;type: 1 - 1111 ZCOD1
00D1 DA0801	185	JC	ZCOD1	; 0 - 0000 ZCOD0
	186 ;			
00D4 24	187 ZCOD0:	INR	H	
00D5 CAF900	188	JZ	RZCOD0	;prev code -1 for 1111
	189 ;			
00DB 2601	190 EZCOD0:	MVI	H,01H	;error
00DA 2D	191	DCR	L	;update err cnt
00DB CAEA00	192	JZ	OVCOD0	;--- too many errors
	193 ;			
00DE E5	194	PUSH	H	
00DF 3E08	195	MVI	A,ZCODE1	
00E1 D328	196	OUT	ADNEC	;0 code to NEC
	197 ;			
00E3 DB28	198	IN	ADNEC	
00E5 D338	199	OUT	QCODE	;Qcode NEC -> DIO
00E7 C34401	200	JMP	CODCON	;--- end of code sync
	201 ;			
00EA 210020	202 OVCOD0:	LXI	H,SYNCRNT	
00ED E5	203	PUSH	H	
00EE 3E08	204	MVI	A,ZCODE1	
00F0 D328	205	OUT	ADNEC	;0 code to NEC
00F2 DB28	206	IN	ADNEC	
00F4 D339	207	OUT	RSYNC	;Qcode NEC -> DIO, restart acq
00F6 C34401	208	JMP	CODCON	;--- end of code sync
	209 ;			
00F9 210401	210 RZCOD0:	LXI	H,ZICOD0	;code = 1, err cnt = 3
00FC E5	211	PUSH	H	
00FD 3E08	212	MVI	A,ZCODE1	
00FF D328	213	OUT	ADNEC	;0 code to NEC
0101 DB28	214	IN	ADNEC	
0103 D338	215	OUT	QCODE	;Qcode NEC -> DIO
0105 C34401	216	JMP	CODCON	;--- end of code sync
	217 ;			
	218 ;	---	Zero code 1111	
	219 ;			

LOC	OBJ	LINE	SOURCE STATEMENT
-----	-----	------	------------------

```

0108 25      220 ZCOD1: DCR      H
0109 CA2D01  221      JZ      RZCOD1 ;prev code +1 for 0000
                222 ;
010C 26FF    223 EZCOD1: MVI      H,0FFH
010E 2D      224      DCR      L
010F CA1E01  225      JZ      OVCOD1 ;--- too many errors
                226 ;
0112 E5      227      PUSH     H
0113 3E08    228      MVI      A,ZCODE1
0115 D328    229      OUT      ADNEC ;0 code to NEC
0117 DB28    230      IN       ADNEC
0119 D338    231      OUT      QCODE ;Qcode NEC -> DIO
011B C34401  232      JMP      CODCON ;--- end of code sync
                233 ;
011E 210020  234 OVCOD1: LXI      H,SYNCR
0121 E5      235      PUSH     H
0122 3E08    236      MVI      A,ZCODE1
0124 D328    237      OUT      ADNEC ;0 code to NEC
0126 DB28    238      IN       ADNEC
0128 D339    239      OUT      RSYNC ;Qcode NEC -> DIO, restart sync
012A C34401  240      JMP      CODCON ;--- end of code sync
                241 ;
012D 2104FF  242 RZCOD1: LXI      H,ZICOD1 ;code = -1(1111), err cnt = 3
0130 E5      243      PUSH     H
0131 3E08    244      MVI      A,ZCODE1
0133 D328    245      OUT      ADNEC ;0 code to NEC
0135 DB28    246      IN       ADNEC
0137 D338    247      OUT      QCODE ;Qcode NEC -> DIO
0139 C34401  248      JMP      CODCON ;--- end of code sync
                249 ;
                250 ;--- Non zero code
                251 ;
013C E5      252 NZCOD:  PUSH     H
013D 79      253      MOV      A,C
013E D328    254      OUT      ADNEC ;Qcode DIO -> NEC
0140 DB28    255      IN       ADNEC
0142 D338    256      OUT      QCODE ;Qcode NEC -> DIO
                257 CODCON:
                258 ;
                259 ; Receiver Section
                260 ;
0144 311408  261      LXI      SP,PTRRN ;P for pxmen
0147 E1      262      POP      H
0148 23      263      INX      H
                264 ;
0149 DB28    265      IN       ADNEC ;[Rn] from NEC
014B B6      266      ORA      H ;tag on xmt bits
014C 77      267      MOV      H,A ;save in pxmen
                268 ;
014D 3EFD    269      MVI      A,MSKRA ;msk out.b10

```

014F A4	270	ANA	H	
0150 67	271	MOV	H,A	
0151 E5	272	PUSH	H	;save P(Rn)
0152 EE38	273	XRI	XPR	;ask pxmen->rmen
0154 67	274	MOV	H,A	;HL=P(Rk)

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 6

LDC OBJ	LINE	SOURCE STATEMENT	
	275 ;		
0155 DB28	276	IN	ADNEC
0157 77	277	MOV	M,A
0158 DB28	278	IN	ADNEC
015A 77	279	MOV	M,A ;double word Rk
	280 ;		
015B EB	281	XCHG	;DE=P(Rk)
015C 2A1808	282	LHLD	TRCV
015F EB	283	XCHG	;HL=P(Rk), DE=Trcv
	284 ;		
0160 3A1C08	285	LDA	CTRCV ;rcv pitch block cntr -ve
0163 42	286	MOV	B,D ;msb = 03H
0164 3C	287	INR	A
0165 CA9C01	288	JZ	RPNEU ;--- new pitch period
	289 ;		
0168 321C08	290	RPOLD: STA	CTRCV ;save pitch block cntr
016B 4F	291	MOV	C,A ; and in C
016C 09	292	DAD	B ;HL=P(Rk-cntr)
016D 3EFB	293	MVI	A,MSKRA ;ask out b10
016F A4	294	ANA	H
0170 EE02	295	XRI	MSKRB ;ask invert b9
0172 67	296	MOV	H,A ;HL=P(Rn)
	297 ;		
0173 7E	298	MOV	A,M ;double word Rn
0174 D328	299	OUT	ADNEC
0176 7E	300	MOV	A,M
0177 D328	301	OUT	ADNEC
	302 ;		
0179 23	303	INX	H ;HL=P(Rn+1)
017A 7E	304	MOV	A,M ;double word Rn+1
017B D328	305	OUT	ADNEC
017D 7E	306	MOV	A,M
017E D328	307	OUT	ADNEC
0180 2B	308	DCX	H ;HL=P(Rn)
	309 ;		
0181 19	310	DAD	D ;HL=P(Rn-T)
0182 7E	311	MOV	A,M ;double word Rn-T
0183 D328	312	OUT	ADNEC
0185 7E	313	MOV	A,M

0186 D328	314	OUT	ADNEC	
	315 ;			
0188 23	316	INX	H	;HL=P(Rn+1-T)
0189 7E	317	MOV	A,M	;double word Rn+1-T
018A D328	318	OUT	ADNEC	
018C 7E	319	MOV	A,M	
018D D328	320	OUT	ADNEC	
	321 ;			
018F 3EFF	322	MVI	A,OFFH	;Trvc (-ve) to NEC
0191 D328	323	OUT	ADNEC	
	324 ;			
0193 FB	325	TWAIT: EI		
0194 00	326	NOP		
0195 00	327	NOP		
0196 00	328	NOP		
0197 00	329	NOP		

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 7

LOC	OBJ	LINE	SOURCE STATEMENT
0198 00		330	NOP
0199 C39301		331	JMP TWAIT
		332 ;	
019C DB01		333	RPNEU: IN AD87B ;get new Trcv
019E 4F		334	MOV C,A ;+ve T in C
019F 2F		335	CHA
01A0 3C		336	INR A ; -ve T
01A1 5F		337	MOV E,A ; in DE
01A2 321C08		338	STA CTRCV ;save pitch block cntr, -ve
		339 ;	
01A5 19		340	DAD D ;HL=P(Rk-cntr)
01A6 3EFB		341	MVI A,MSKRA ;msk out b10
01A8 A4		342	ANA H
01A9 EE02		343	XRI MSKRB ;msk invert b9
01AB 67		344	MOV H,A ;HL=P(Rn)
		345 ;	
		346 ;	Ask for new rcv pitch
		347 ;	
01AC 7C		348	MOV A,H ;MSB
01AD D30A		349	OUT AD81B
01AF 7D		350	MOV A,L ;LSB
01B0 D309		351	OUT AD81A
		352 ;	
01B2 7E		353	MOV A,M ;double word Rn
01B3 D328		354	OUT ADNEC
01B5 7E		355	MOV A,M
01B6 D328		356	OUT ADNEC
		357 ;	

01B8 23	358	INX	H	;HL=P(Rn+1)
01B9 7E	359	MOV	A,M	;double word Rn+1
01BA D328	360	OUT	ADNEC	
01BC 7E	361	MOV	A,M	
01BD D328	362	OUT	ADNEC	
01BF 2B	363	DCX	H	;HL=P(Rn)
	364 ;			
01C0 19	365	DAD	D	;HL=P(Rn-T)
01C1 7E	366	MOV	A,M	;double word Rn-T
01C2 D328	367	OUT	ADNEC	
01C4 7E	368	MOV	A,M	
01C5 D328	369	OUT	ADNEC	
	370 ;			
01C7 23	371	INX	H	;HL=P(Rn+1-T)
01C8 7E	372	MOV	A,M	;double word Rn+1-T
01C9 D328	373	OUT	ADNEC	
01CB 7E	374	MOV	A,M	
01CC D328	375	OUT	ADNEC	
	376 ;			
01CE 79	377	MOV	A,C	;new Trcv (+ve) to NEC
01CF D328	378	OUT	ADNEC	
	379 ;			
01D1 EB	380	XCHG		
01D2 221808	381	SHLD	TRCV	;save new Trcv -ve
	382 ;			
01D5 C39301	383	JMP	TWAIT	
	384 ;			

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN PAGE 8

LOC	OBJ	LINE	SOURCE STATEMENT
		385 ;	
		386 ;	Initialization
		387 ;	
		388 INIT:	
		389 ;	
01D8	3E37	390	MVI A,STP81
01DA	D308	391	OUT ADS81 ;8155 ports:A-strobed out
		392 ;	B-out
01DC	3E00	393	MVI A,STP87A ;8755 ports:A-inp
01DE	D302	394	OUT ADS87A ; B-inp
01E0	D303	395	OUT ADS87B
		396 ;	
		397 ;	Initialize each of the variables
		398 ;	
01E2	310408	399	LXI SP,PTXSN ;xmem pointer = 1000H
01E5	E1	400	POP H
01E6	210010	401	LXI H,1000H

01E9 E5	402	PUSH	H	
	403 ;			
01EA 310808	404	LXI	SP, TXMT	;Txmt = 40 (0028H)
01ED E1	405	POP	H	
01EE 212800	406	LXI	H, 0028H	
01F1 E5	407	PUSH	H	
	408 ;			
01F2 310C08	409	LXI	SP, CTXMT	;xmt cntr = 0
01F5 E1	410	POP	H	
01F6 210000	411	LXI	H, 0000H	
01F9 E5	412	PUSH	H	
	413 ;			
01FA 311408	414	LXI	SP, PTRRN	;rmen pointer = 2000H
01FD E1	415	POP	H	
01FE 210020	416	LXI	H, 2000H	; pxmen for rcv section
0201 E5	417	PUSH	H	
	418 ;			
0202 311808	419	LXI	SP, TRCV	;Trcv = -40 (03D8H) 10 bits
0205 E1	420	POP	H	
0206 21D803	421	LXI	H, 03D8H	
0209 E5	422	PUSH	H	
	423 ;			
020A 311C08	424	LXI	SP, CTRCV	;rcv cntr = -40
020D E1	425	POP	H	
020E 21D803	426	LXI	H, 03D8H	
0211 E5	427	PUSH	H	
	428 ;			
0212 311008	429	LXI	SP, PTCWS	;8000 for code sync time out
0215 E1	430	POP	H	
0216 210020	431	LXI	H, SYNCNT	
0219 E5	432	PUSH	H	
	433 ;			
	434 ;			Initialize FIFO, 13 values
	435 ;			
021A 3EFF	436	MVI	A, OFFH	
021C D328	437	OUT	ADNEC ;Sn	
	438 ;			
021E 3E28	439	MVI	A, 28H	

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MAIN

PAGE

9

LOC	OBJ	LINE	SOURCE STATEMENT
0220	D328	440	OUT ADNEC ;T
		441 ;	
0222	3EFF	442	MVI A, OFFH
0224	D328	443	OUT ADNEC ;Sn+T
0226	3E00	444	MVI A, 00H
0228	D328	445	OUT ADNEC ;Qk

	446 ;		
022A D328	447	OUT	ADNEC
022C D328	448	OUT	ADNEC ;Rn, double word
	449 ;		
022E D328	450	OUT	ADNEC
0230 D328	451	OUT	ADNEC ;Rn+1
	452 ;		
0232 D328	453	OUT	ADNEC
0234 D328	454	OUT	ADNEC ;Rn-T
	455 ;		
0236 D328	456	OUT	ADNEC
0238 D328	457	OUT	ADNEC ;Rn+1-T
	458 ;		
023A 3E28	459	MVI	A,28H
023C D328	460	OUT	ADNEC ;Trcv
	461 ;		
023E 3E00	462	MVI	A,00H ;Init sync
0240 D339	463	OUT	RSYNC
	464 ;		
0242 3E80	465	MVI	A,RNINT ;Reset interrupt
0 0244 30	466	SIM	
()			
0245 3E0B	467	MVI	A,ENINT ;Enable interrupt
0 0247 30	468	SIM	
(466)			
0248 FB	469	EI	
0249 C39301	470	JMP	TWAIT
0000	471	END	START

B.2 Pitch Extraction Software: PEXTR

The microprocessor in this section does a pitch extraction whenever requested by the control processor. The software is divided into two interrupt driven routines. The first part acknowledges a request for pitch extraction, determines whether it is for the transmitter or receiver, saves the starting address for pitch extraction and sets the appropriate pitch pending flag. The second routine normally sits in a waiting loop which polls the status of the pitch pending flags. If a flag is set, it sends out the appropriate correlation addresses to the hardware (which performs the actual correlation) and directs the hardware through the range of possible pitch periods. The point of highest correlation is selected to be the pitch period. It outputs the new pitch period (separate ports for the transmitter and receiver), resets the pending flag and goes back to the polling loop.

The flowchart uses the following symbols:

- Addr - Starting address for pitch extraction
- Mxr - Transmitter/receiver mode
- Fx - Flag to indicate request for transmitter pitch
- Fr - Flag to indicate request for receiver pitch
- Faddr - Fixed address for correlation
- Vaddr - Variable address for correlation
- Xtag - Transmitter indicator to hardware
- Rtag - Receiver indicator to hardware
- Lblk - Correlation block length
- H - Current Histogram value
- Hmax - Histogram maximum
- T - Current correlation delay
- Tmax - (pitch period) correlation delay corresponding to histogram maximum

Phdw - Pitch extraction hardware

Xport - Port for transmitter pitch

Rport - Port for receiver pitch

Pitch Request Acknowledgements:

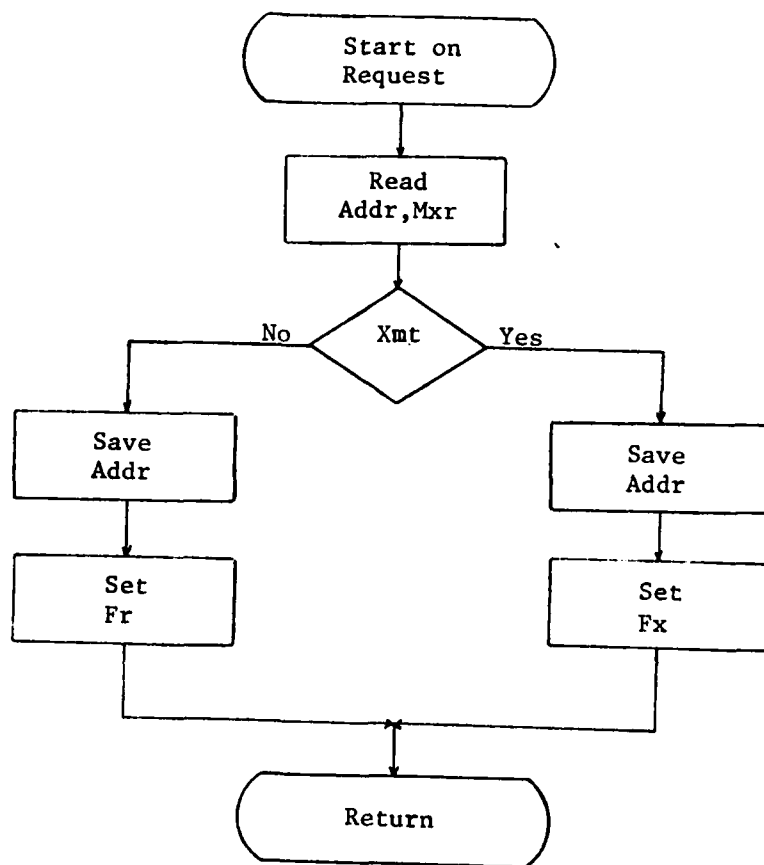


Fig. B.2: Flow Chart for the Pitch Extraction Program, PEXTR

Pitch Extraction Wait Loop:

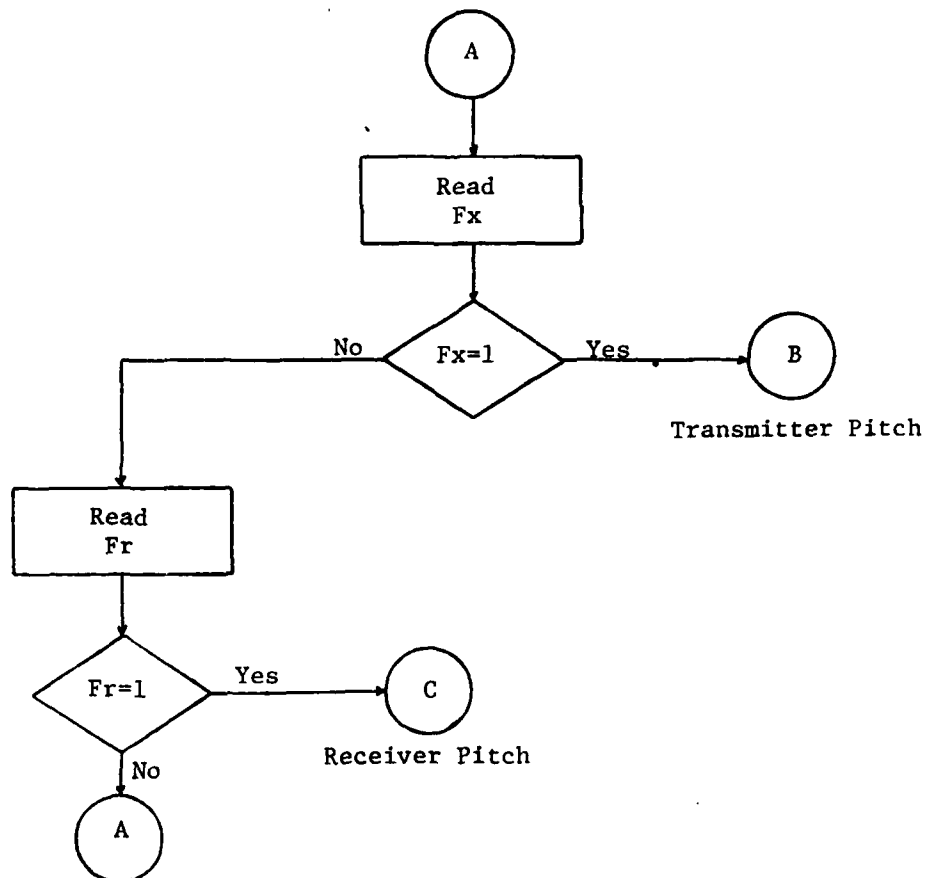


Fig. B.2 (contd): Flow Chart for the Pitch Extraction Program, PEXTR

Transmitter Pitch:

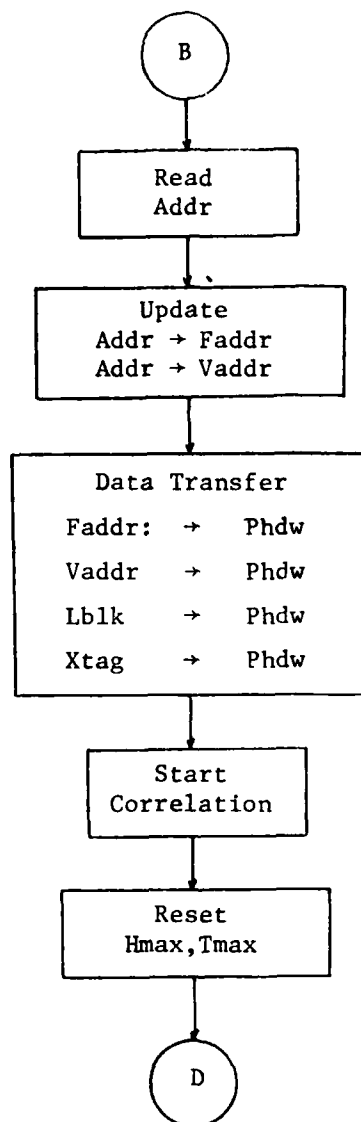


Fig. B.2 (contd): Flow Chart for the Pitch Extraction Program, PEXTR

Receiver Pitch:

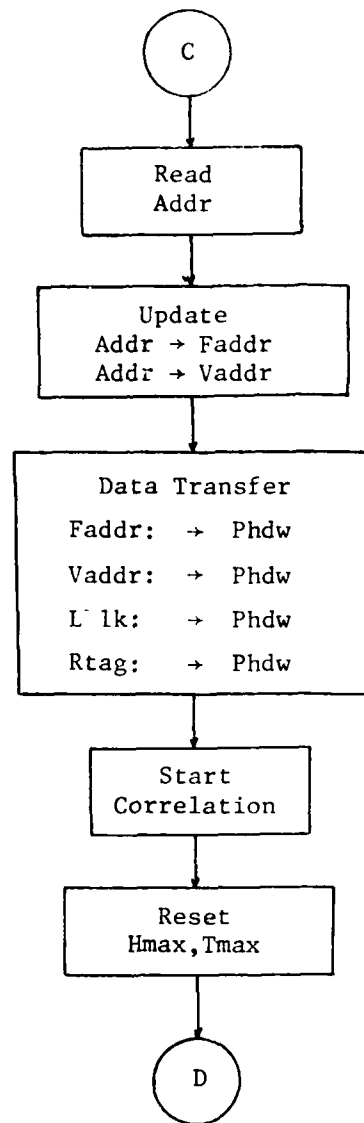


Fig. B.2 (contd): Flow Chart for the Pitch Extraction Program, PEXTR

Correlation Loop:

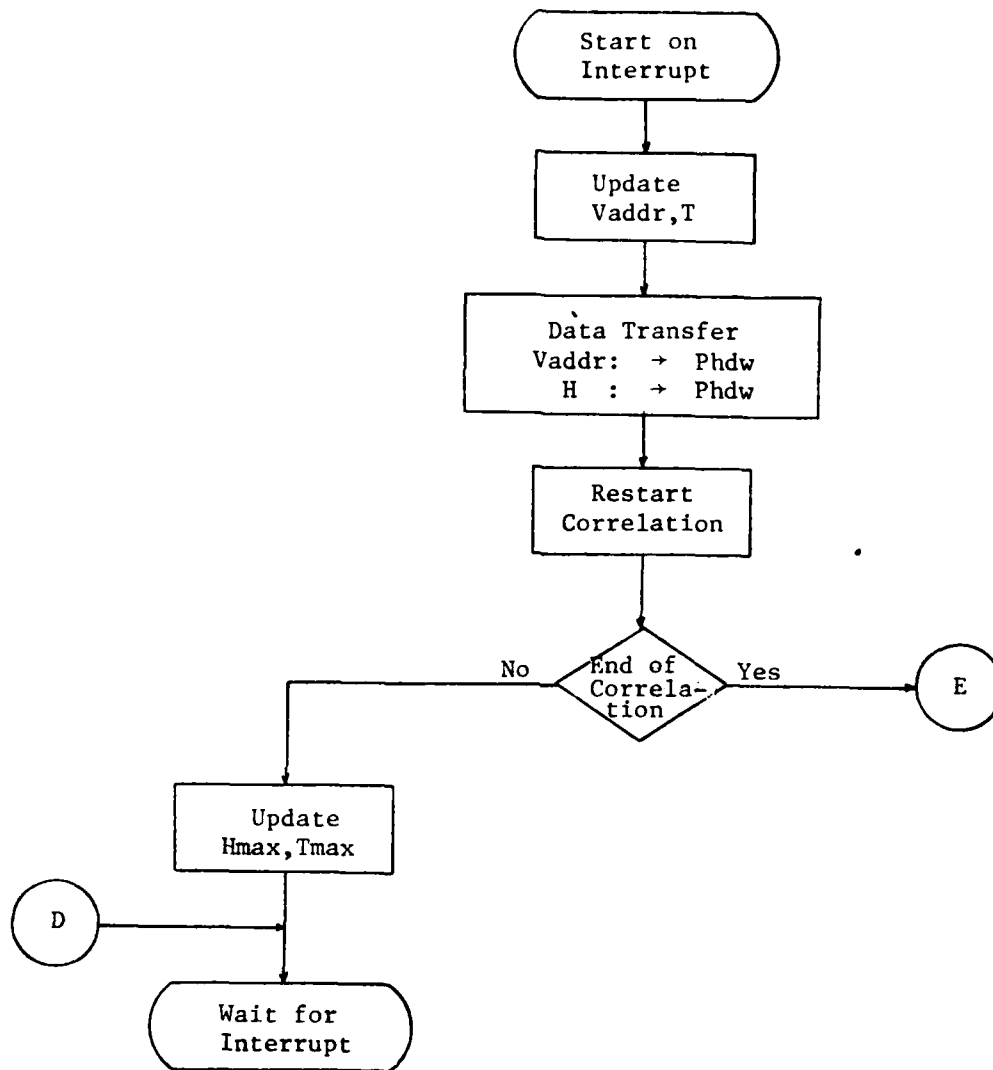


Fig. B.2 (contd): Flow Chart for the Pitch Extraction Program, PEXTR

End of Pitch Determination:

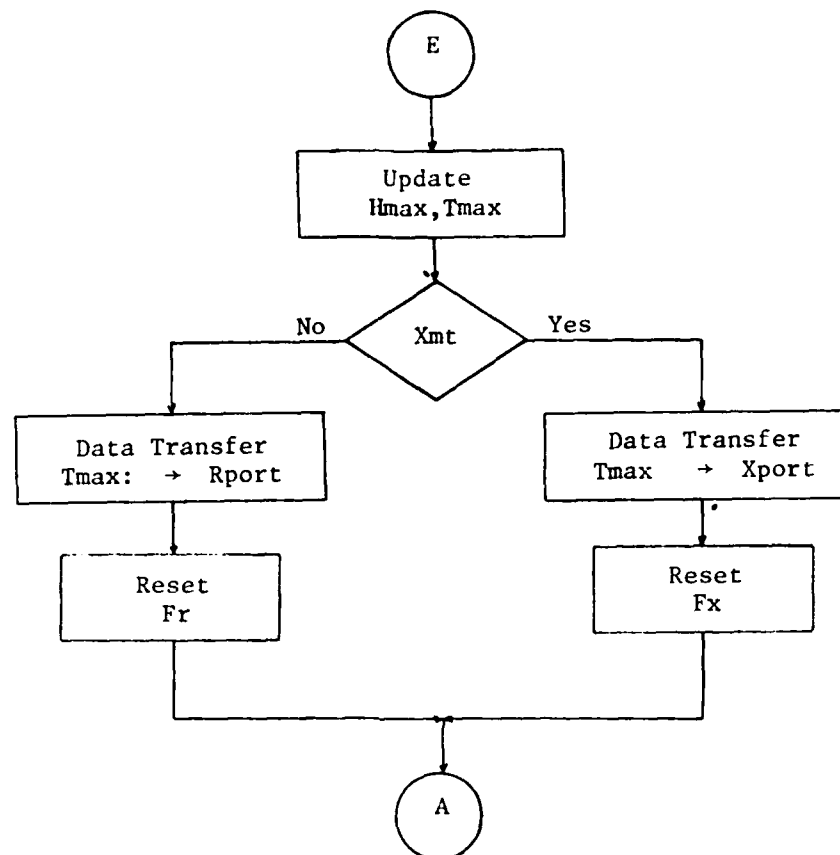


Fig. B.2 (contd): Flow Chart for the Pitch Extraction Program, PEXTR

LOC	OBJ	LINE	SOURCE STATEMENT
		1 ;	
		2 ;	This program does the pitch extraction.
		3 ;	It goes into micro 2.
		4 ;	
		5 ;	
C0FF		6 PTSTAK EQU	0C0FFH ;Stack pointer
C000		7 PTMTP EQU	0C000H ;Memory top
C000		8 HISADR EQU	0C000H ;Max hist., T
C002		9 XMTADR EQU	0C002H ;Addr for xmt pitch
C004		10 RCVADR EQU	0C004H ;Addr for rcv pitch
C006		11 XTOLD EQU	0C006H ;Old T xmt
C008		12 RTOLD EQU	0C008H ;Old T rcv
000D		13 ENINT EQU	0DH ;Int mask 65
00FC		14 STP81 EQU	0FCH ;Status for 8155 ports
007C		15 STI81 EQU	07CH ;Status - stop timer
00C0		16 AD81ST EQU	0C0H ;Port - 8155 status
00C1		17 AD81A EQU	0C1H ;Port - 8155 A
00C2		18 AD81B EQU	0C2H ;Port - 8155 B
00C3		19 AD81C EQU	0C3H ;Port - 8155 C
00C4		20 ADLCNT EQU	0C4H ;Port - timer LSB
00C5		21 ADHCNT EQU	0C5H ;Port - timer MSB
00FF		22 STP87A EQU	0FFH ;Status for 8755 A
00FF		23 STP87B EQU	0FFH ;Status for 8755 B
0002		24 ADS87A EQU	02H ;Port - status 8755 A
0003		25 ADS87B EQU	03H ;Port - status 8755 B
0000		26 AD87A EQU	00H ;Port - 8755 A
0001		27 AD87B EQU	01H ;Port - 8755 B
0099		28 PCHRST EQU	99H ;Port - pitch reset
0098		29 PCHSTX EQU	98H ;Port - pitch strt, Hist out
0090		30 ADPLF EQU	90H ;Port - LSB fixed addr
0088		31 ADPLV EQU	88H ;Port - LSB variable addr
0080		32 ADPHS EQU	80H ;Port - MSB addr, xmt/rcv
0064		33 XLCNT EQU	64H ;100, LSB, xmt pitch blk length
00C0		34 XMCNT EQU	0C0H ;MSB, xmt pitch blk, mode
0064		35 RLCNT EQU	64H ;100, LSB, rcv pitch blk length
00C0		36 RMCNT EQU	0C0H ;MSB, rcv pitch blk, mode
0000		37 RCVDSP EQU	00H ;Displacement for rcv addr,-ve
0028		38 TADLAG EQU	0028H ;Starting T
00B9		39 TCNT EQU	0B9H ;-71, value for pitch counter
006E		40 TNORM EQU	6EH ;110, add to cntr for T value
0010		41 XMTMSK EQU	10H ;Addr mask for xmt

```

0018          42 RCVMSK EQU    18H    ;Addr mask for rcv
          43 ;
0000          44 ORG      0000H
0000 C34301   45 RST00: JMP     INIT    ;Reset at INIT
0024          46 ORG      0024H
0024 C31C01   47 RST1R: JMP     TRAP    ;Trap at TRAP
0034          48 ORG      0034H
          49 ;
          50 ;    Here starts the process for each histogram.
          51 ;
0034 31FFC0   52 RST65: LXI     SP,PTSTAK
          53 ;
0037 13       54          INX     'D      ;update var. addr

```

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MODULE PAGE 2

LOC	OBJ	LINE	SOURCE STATEMENT
0038	7B	55	MOV A,E ;addr LSB
0039	D388	56	OUT ADPLV
003B	7A	57	MOV A,D ;addr MSB
003C	D380	58	OUT ADPMS
		59 ;	
003E	DB98	60	IN PCHSTX ;read histogram
0040	D398	61	OUT PCHSTX ;restart next histogram
		62 ;	
0042	04	63	INR B ;check if end of pitch
0043	CA5A00	64	JZ PCHEND
		65 ;	
0046	FE00	66	CPI 00H ;check +ve
0048	FA5600	67	JM PWAIT
004B	BE	68	CMP M
004C	FA5600	69	JM PWAIT ;< ,continue
004F	CA5600	70	JZ PWAIT ;= ,continue
		71 ;	
0052	77	72	HSTUP1: MOV M,A ;save new hist.
0053	2C	73	INR L
0054	70	74	MOV M,B ;save new Tmax
0055	2D	75	DCR L
		76 ;	
0056	FB	77	PWAIT: EI
0057	C35600	78	JMP PWAIT
		79 ;	
		80 ;	End of current pitch extraction
		81 ;	
005A	FE00	82	PCHEND: CPI 00H ;check +ve
005C	FA6A00	83	JM PEND1
		84 ;	
005F	BE	85	CMP M

0060 FA6A00	86	JM	PEND1	
0063 CA6A00	87	JZ	PEND1	;old max,continue
	88			;
0066 77	89	HSTUP2: MOV	M,A	;save new hist max
0067 2C	90	INR	L	
0068 70	91	MOV	M,B	;save Tmax
0069 2D	92	DCR	L	
	93			;
006A 7A	94	PEND1: MOV	A,D	;cont end of pitch
006B 07	95	RLC		;check xmt/rcv
006C DA7B00	96	JC	PCHRCV	;b7=1, rcv pitch
	97			;
006F 2C	98	PCHXNT: INR	L	;point to Tmax
0070 7E	99	MOV	A,M	
0071 C66E	100	ADI	TNORM	;add 110 to cntr
0073 D300	101	OUT	AD87A	;xmt pitch
0075 3206C0	102	STA	XTOLD	;save Told
0078 C3D100	103	JMP	PRNXT	
	104			;
007B 2C	105	PCHRCV: INR	L	;point to Tmax
007C 7E	106	MOV	A,M	
007D C66E	107	ADI	TNORM	;add 110 to cntr
007F D301	108	OUT	AD87B	;rcv pitch
0081 3208C0	109	STA	RTOLD	;save Told rcv

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MODULE PAGE 3

LOC	OBJ	LINE	SOURCE STATEMENT
0084	C38700	110	JMP PXNXT
		111	;
		112	; Next pitch extraction operation
		113	;
0087	DB00	114	PXNXT: IN AD87A ;
0089	07	115	RLC
008A	D2D100	116	JNC PRNXT ;b7=1, new xmt pitch needed
008D	2A02C0	117	PXNXT1: LHLD XMTADR
0090	2B	118	DCX H
0091	EB	119	XCHG
0092	2A06C0	120	LHLD XTOLD ;addr+Told-1
0095	19	121	DAD D
0096	112800	122	LXI D,TADLAG
0099	7D	123	MOV A,L
009A	D390	124	OUT ADPLF ;LSB, fixed addr
009C	7C	125	MOV A,H
009D	E603	126	ANI 03H ;2 LSB
009F	67	127	MOV H,A
00A0	07	128	RLC
00A1	07	129	RLC

00A2 07	130	RLC		;to b3,4 and xnt mode b7=0
00A3 47	131	MOV	B,A	
00A4 19	132	DAD	D	;var addr in HL
00A5 7D	133	MOV	A,L	
00A6 D388	134	OUT	ADPLV	;LSB, var. addr
00A8 7C	135	MOV	A,H	;get MSB var addr
00A9 E603	136	ANI	03H	;mask b0,1
00AB B0	137	ORA	B	;combine 2 addr, mode
00AC D380	138	OUT	ADPMS	;out MSB addr
00AE 5D	139	MOV	E,L	;LSB in E
00AF 57	140	MOV	D,A	;MSB in D
	141 ;			
00B0 3E64	142	MVI	A,XLCNT	;set up xnt block length
00B2 D3C4	143	OUT	ADLCNT	
00B4 3EC0	144	MVI	A,XMCNT	
00B6 D3C5	145	OUT	ADMCNT	
	146 ;			
00B8 3E7C	147	MVI	A,STI81	;stop timer
00BA D3C0	148	OUT	ADB1ST	
00BC D398	149	OUT	PCHSTX	
00BE 3EFC	150	MVI	A,STP81	;block for xnt
00C0 D3C0	151	OUT	ADB1ST	
00C2 D398	152	OUT	PCHSTX	;start pitch
	153 ;			
00C4 06B9	154	MVI	B,TCNT	;T counter
00C6 2100C0	155	LXI	H,HISADR	;addr for hist max
00C9 36FF	156	MVI	M,OFFH	;hmax=-1
00CB 2C	157	INR	L	
00CC 70	158	MOV	M,B	;Tmax=-70
00CD 2D	159	DCR	L	
	160 ;			
00CE C35600	161	JMP	PWAIT	
	162 ;			
00D1 DB01	163	PRNXT: IN	ADB7B	
00D3 07	164	RLC		

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MODULE PAGE 4

LOC	OBJ	LINE	SOURCE STATEMENT
00D4	D28700	165	JNC PXNXT
		166 ;	
00D7	2A04C0	167	PRNXT1: LHLD RCVADR
00DA	EB	168	XCHG
00DB	2A08C0	169	LHLD RTOLD ;addr+Told
00DE	19	170	DAD D
00DF	112800	171	LXI D,TADLAG
00E2	7D	172	MOV A,L
00E3	D390	173	OUT ADPLF ;LSB fixed addr

```

174 ;
00E5 7C      175      MOV    A,H
00E6 E603    176      ANI    03H
00E8 67      177      MOV    H,A
00E9 07      178      RLC
00EA 07      179      RLC
00EB 07      180      RLC
00EC F680    181      ORI     80H      ;to b3,4 and rcv node b7=1
00EE 47      182      MOV    B,A
00EF 19      183      DAD    D
00F0 7D      184      MOV    A,L
00F1 D388    185      OUT    ADPLV    ;LSB var addr
186 ;
00F3 7C      187      MOV    A,H
00F4 E603    188      ANI    03H
00F6 80      189      ORA    B
00F7 D380    190      OUT    ADPMS    ;MSB addr,node
00F9 57      191      MOV    D,A
00FA 5D      192      MOV    E,L
193 ;
00FB 3E64    194      MVI    A,RLCNT ;set up pitch blocj length
00FD D3C4    195      OUT    ADLCNT
00FF 3EC0    196      MVI    A,RMCNT
0101 D3C5    197      OUT    ADMCNT
198 ;
0103 3E7C    199      MVI    A,STI81 ;stop timer
0105 D3C0    200      OUT    AD81ST
0107 D398    201      OUT    PCHSTX
0109 3EFC    202      MVI    A,STP81 ;block for rcv
010B D3C0    203      OUT    AD81ST
010D D398    204      OUT    PCHSTX ;start pitch extr
205 ;
010F 06B9    206      MVI    B,TCNT  ;init val for p cntr
0111 2100C0  207      LXI    H,HISADR
0114 36FF    208      MVI    M,OFFH  ;Hmax=-1
0116 2C      209      INR    L
0117 70      210      MOV    M,B
0118 2D      211      DCR    L
212 ;
0119 C35600  213      JMP     PWAIT
214 ;
215 ;      New pitch address is received here
216 ;
011C F5      217 TRAP:  PUSH    PSU
011D E5      218      PUSH    H
011E DBC1    219      IN      ADB1A    ;LSB first

```

LOC	OBJ	LINE	SOURCE STATEMENT
0120	6F	220	MOV L,A
0121	DBC2	221	IN AD81B ;MSB second
0123	67	222	MOV H,A
0124	E618	223	ANI RCVMSK ;mask b3,4
0126	FE10	224	CPI XMTMSK
0128	CA3701	225	JZ XMTPOCH ;go to xmt section
		226	;
012B	2204C0	227	RCVPOCH: SHLD RCVADR ;rcv section
012E	DB01	228	IN AD87B
0130	F680	229	ORI 80H ;B7=1
0132	D301	230	OUT AD87B
0134	E1	231	POP H
0135	F1	232	POP PSW
0136	C9	233	RET
		234	;
0137	2202C0	235	XMTPOCH: SHLD XMTADR ;xmt section
013A	DB00	236	IN AD87A
013C	F680	237	ORI 80H ;b7=1
013E	D300	238	OUT AD87A
0140	E1	239	POP H
0141	F1	240	POP PSW
0142	C9	241	RET
		242	;
		243	; Initialization
		244	;
0143	31FFC0	245	INIT: LXI SP,PTSTAK
		246	;
0146	3E0A	247	MVI A,0AH ;set up pitch block =10
0148	D3C4	248	OUT ADLCNT
014A	3EC0	249	MVI A,XMCNT
014C	D3C5	250	OUT ADMCNT
014E	3E7C	251	MVI A,ST181 ;stop counter
0150	D3C0	252	OUT AD81ST
0152	3EFC	253	MVI A,STP81 ;set 8155 ports - inp
0154	D3C0	254	OUT AD81ST
		255	;
0156	3EFF	256	MVI A,STP87A ;set 8755 ports - out
0158	D302	257	OUT ADS87A
015A	D303	258	OUT ADS87B
015C	3E28	259	MVI A,TADLAG ;initial T value
015E	D300	260	OUT AD87A
0160	D301	261	OUT AD87B
0162	6F	262	MOV L,A
0163	2600	263	MVI H,00H
0165	2206C0	264	SHLD XTOLD
0168	2208C0	265	SHLD RTOLD ;save old T
		266	;
016B	2100C0	267	LXI H,HISADR
016E	06FF	268	MVI B,OFFH ;Tcount=-1

0170 1600	269	MVI	D,00H	
0172 1E00	270	MVI	E,00H	;addr=0
	271 ;			
0174 D399	272	OUT	PCHRST	
	273 ;			
0176 3E0D	274	MVI	A,ENINT	;setup interrupts

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MODULE PAGE 6

LOC	OBJ	LINE	SOURCE STATEMENT
0 0178	30	275	SIM
	()		
0179	F3	276	DI
		277 ;	
017A	C38700	278	JMP PXNXT
		279 ;	
0000		280 END	RST00

B.3 Processing Software: TDHSI4, TDHSD4

The processing software resides at NEC 7720 Digital Signal Processor. The software starts with the initialization steps followed by the interrupt service routine which includes Center-clip, TDHC, ARC-Transmitter, ARC-Receiver and TDHE routines. The symbols used in the flow charts are described as follows:

- e_m - Transmitter prediction error,
- $\hat{e}_m$ - Reconstructed transmitter prediction error
- $\hat{e}_t$ - Reconstructed receiver prediction error
- $\langle E_T \rangle$ - Threshold for transmitter center-clip
- $\langle E_R \rangle$ - Threshold for receiver center-clip
- $\hat{P}_m$ - Transmitter prediction value
- $\hat{P}_t$ - Receiver prediction value
- Q_m - Transmitted quantizer code
- $\hat{Q}_t$ - Received quantizer code
- R_m - Transmitter compressed sample
- $\hat{R}_t$ - Receiver reconstructed compressed sample
- $\hat{e}_x$ - Receiver reconstructed compressed sample for TDHE
- RMS_m - Quantizer adaptive coefficient for transmitter
- RMS_t - Quantizer adaptive coefficient for receiver
- S_n - Incoming transmitter sample
- S_k - Transmitter sample used in TDHC
- $[S_n]$ - 2-Bit center-clipped value for S_n
- $\hat{S}_i$ - Reconstructed sample
- T_x - Transmitter pitch period
- T_R - Receiver pitch period
- THR_j - Quantizer thresholds
- THS_j - Quantizer output factors

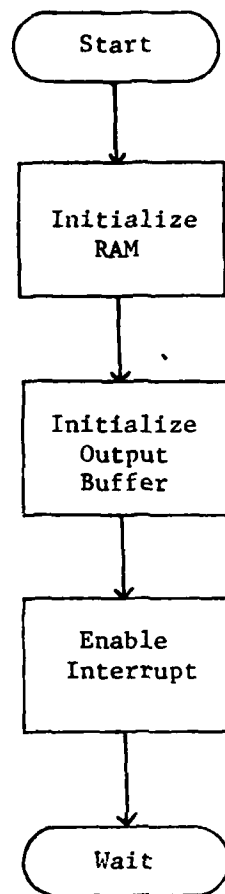


Fig. B.3 Flowchart for the Initialization

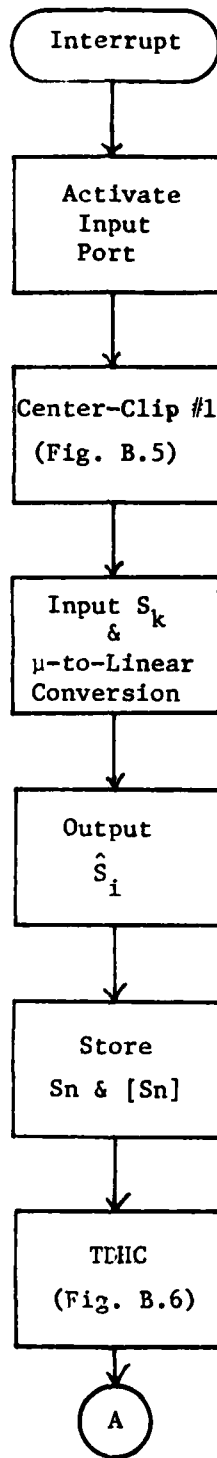


Fig. B4 Main Interrupt Service Routine

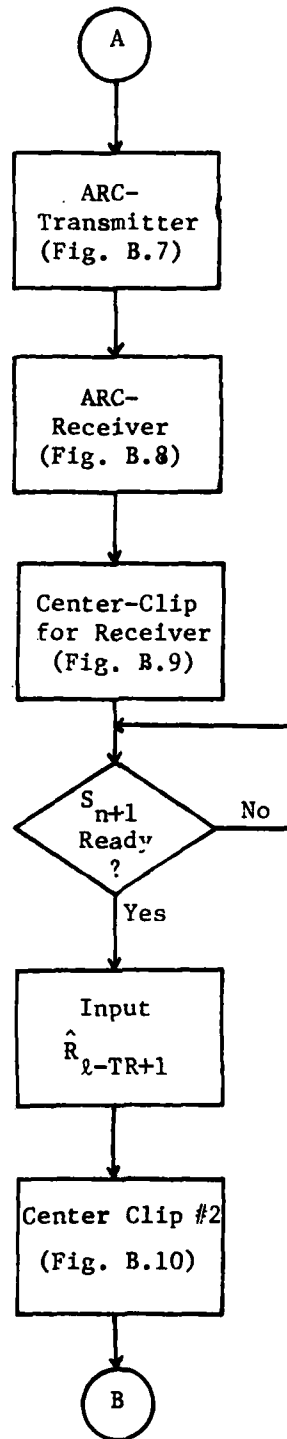


Fig. B.4 (contd.) Main Interrupt Service Routine

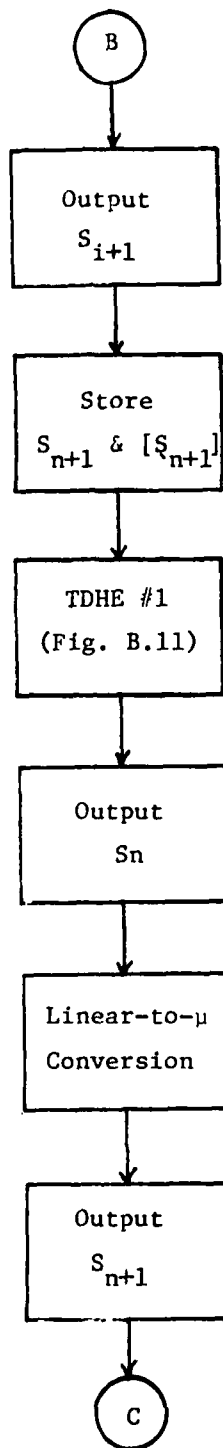


Fig. B.4 (contd.) Main Interrupt Service Rotuine

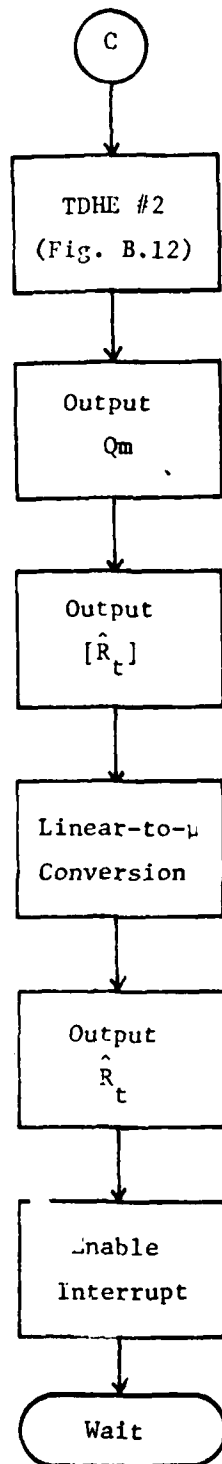


Fig. B.4 (contd.) Main Interrupt Service Routine

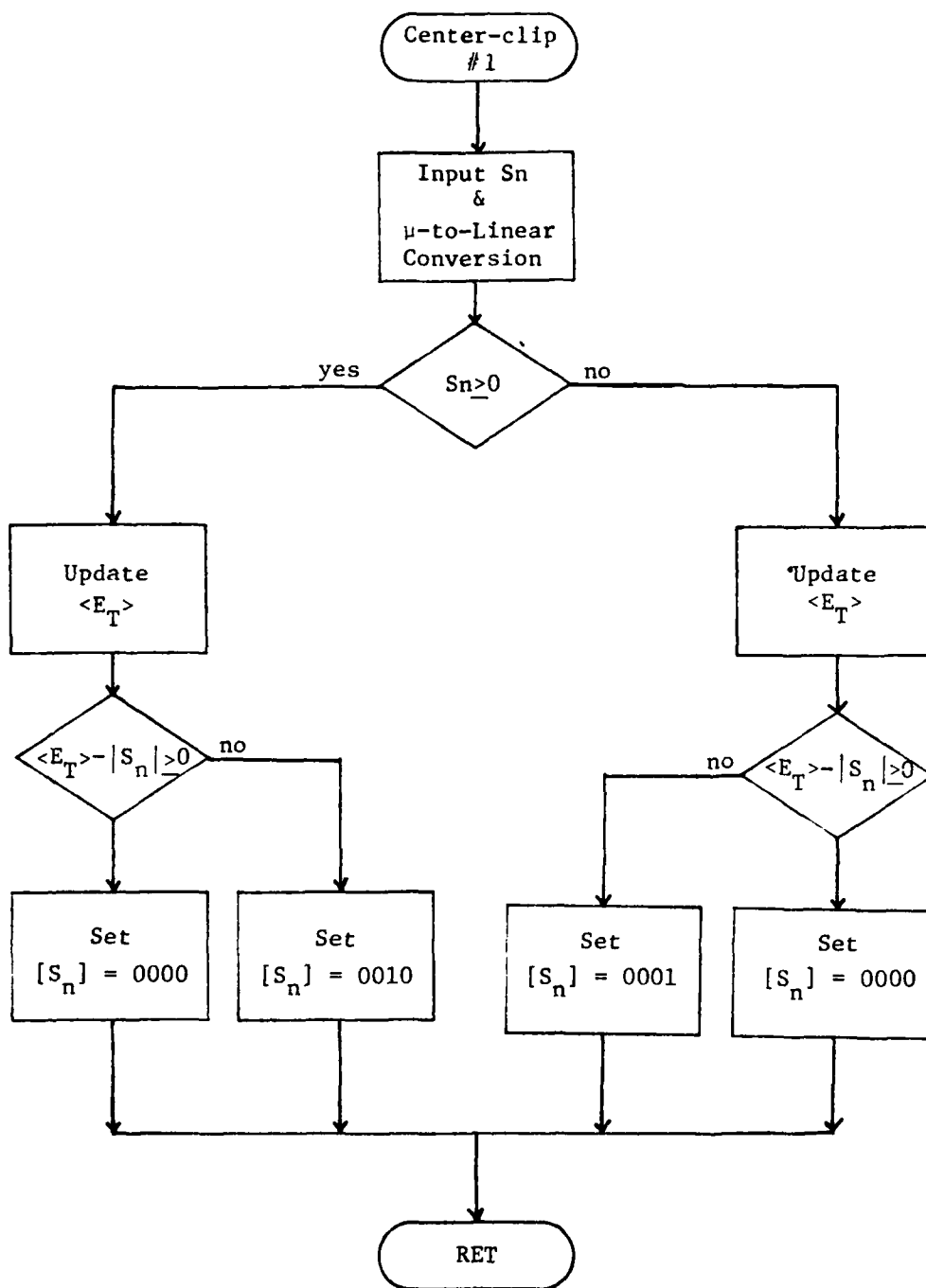


Fig. B.5 Center-Clip #1 for Transmitter

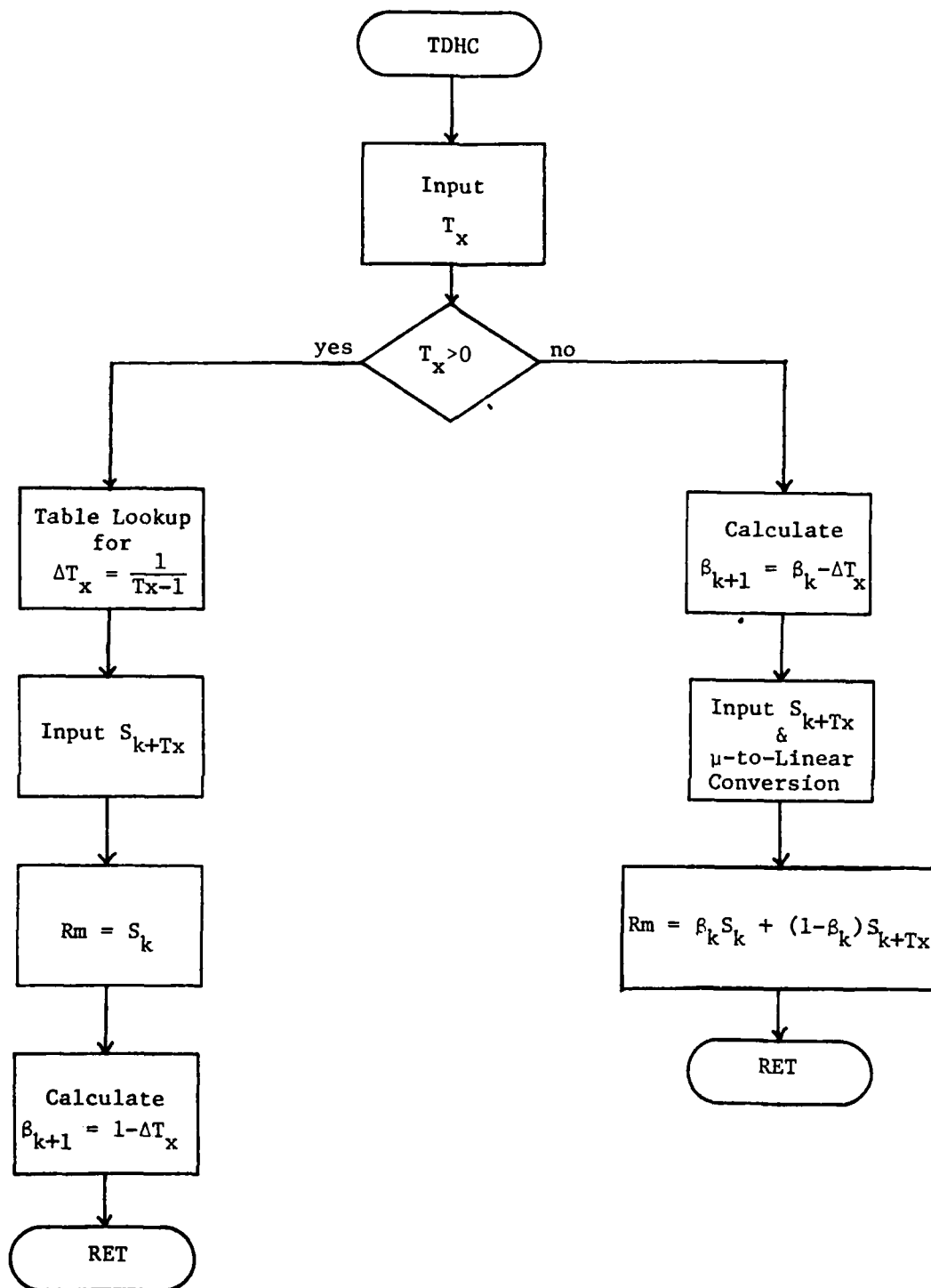


Fig. B.6 TDHC Routine

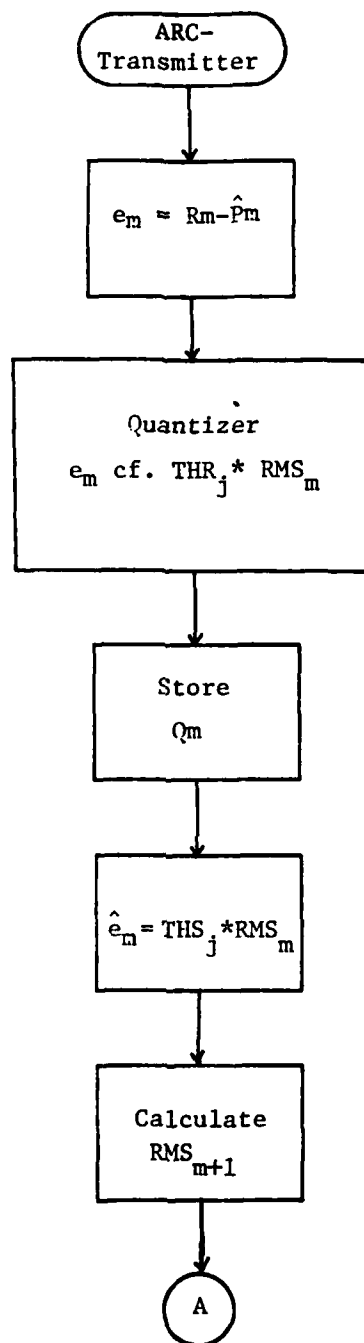


Fig. B.7 ARC-Transmitter Routine

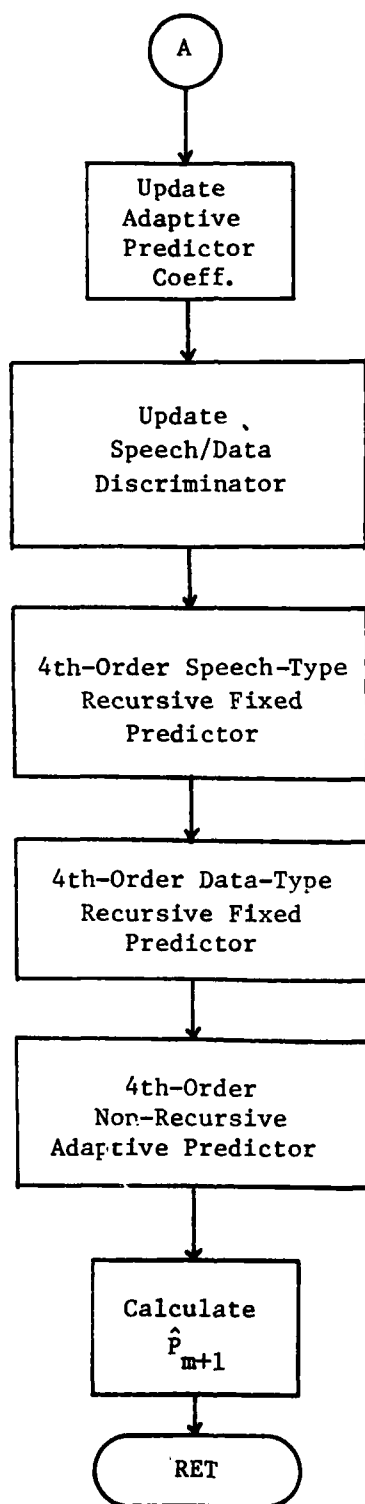


Fig. B.7 (contd) ARC-Transmitter Routine

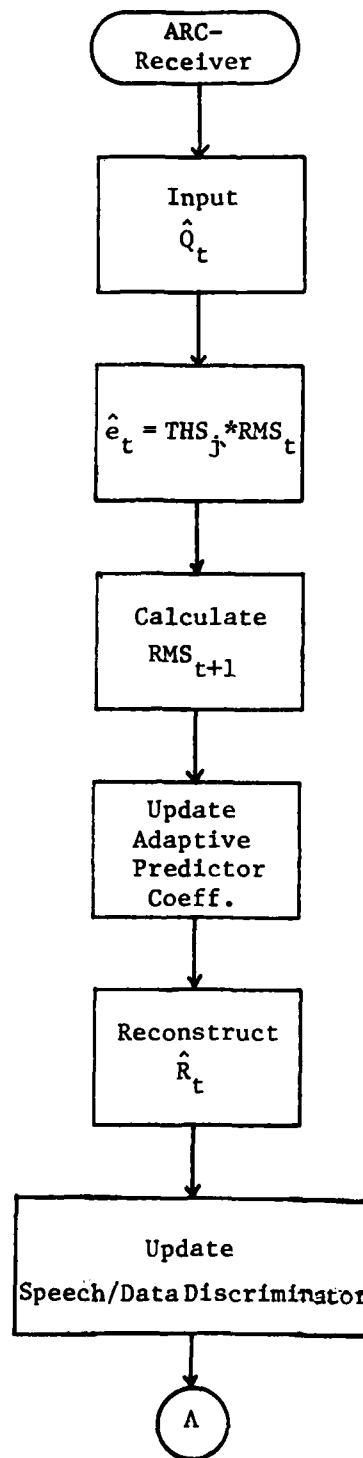


Fig. B.8 ARC-Receiver Routine

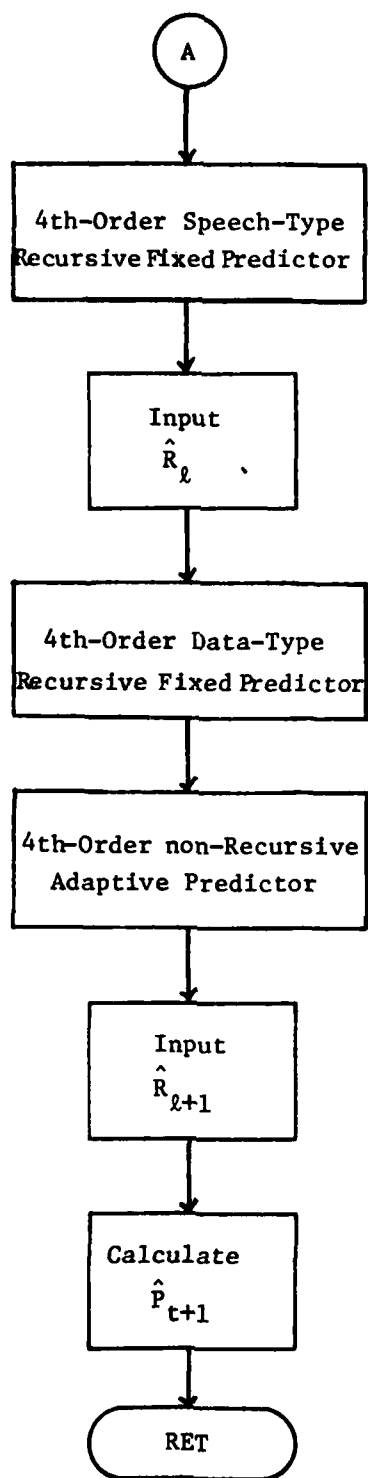


Fig. B.8 (contd) ARC-Receiver Routine

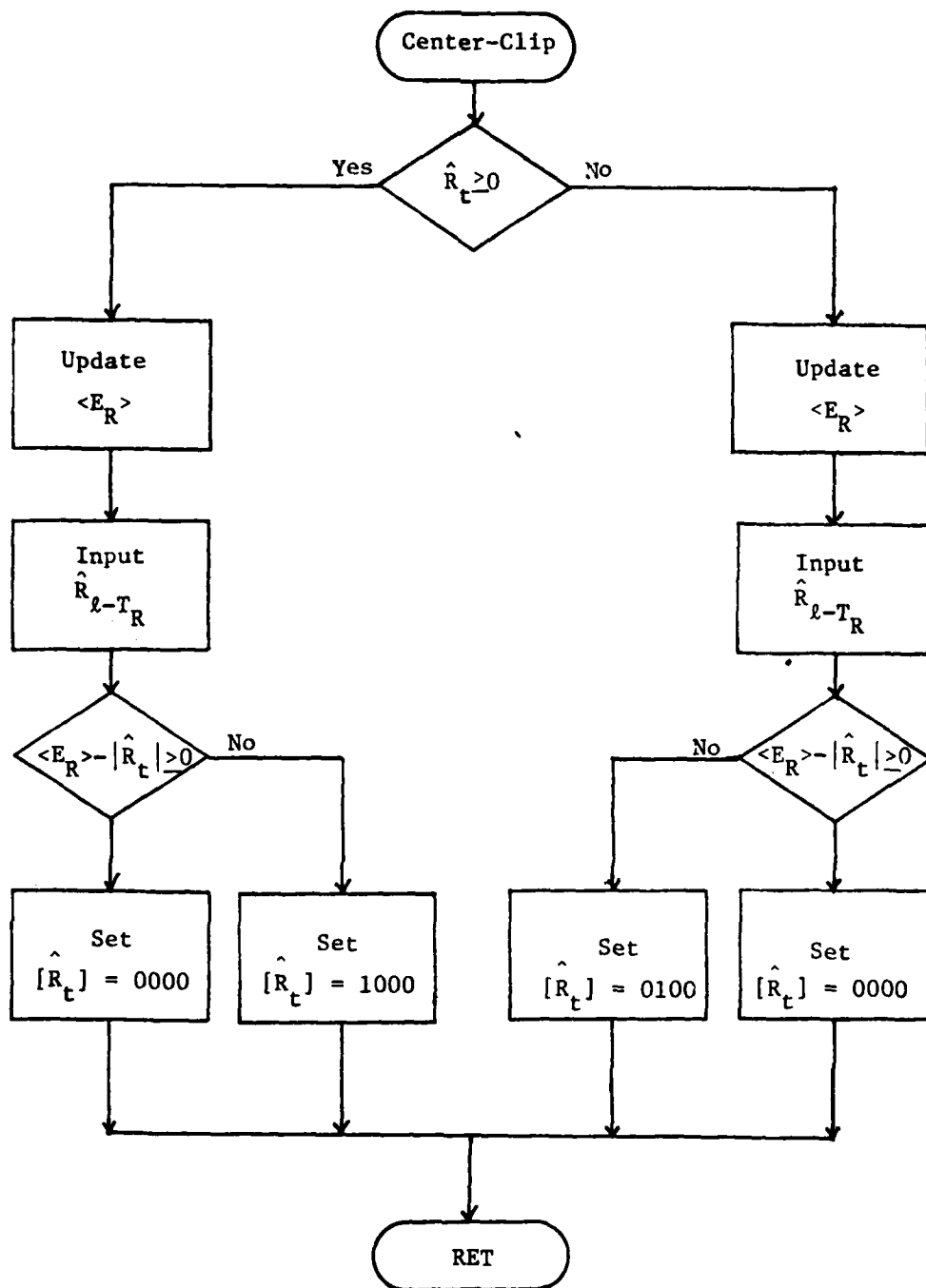


Fig. B.9 Center-Clip Routine for Receiver

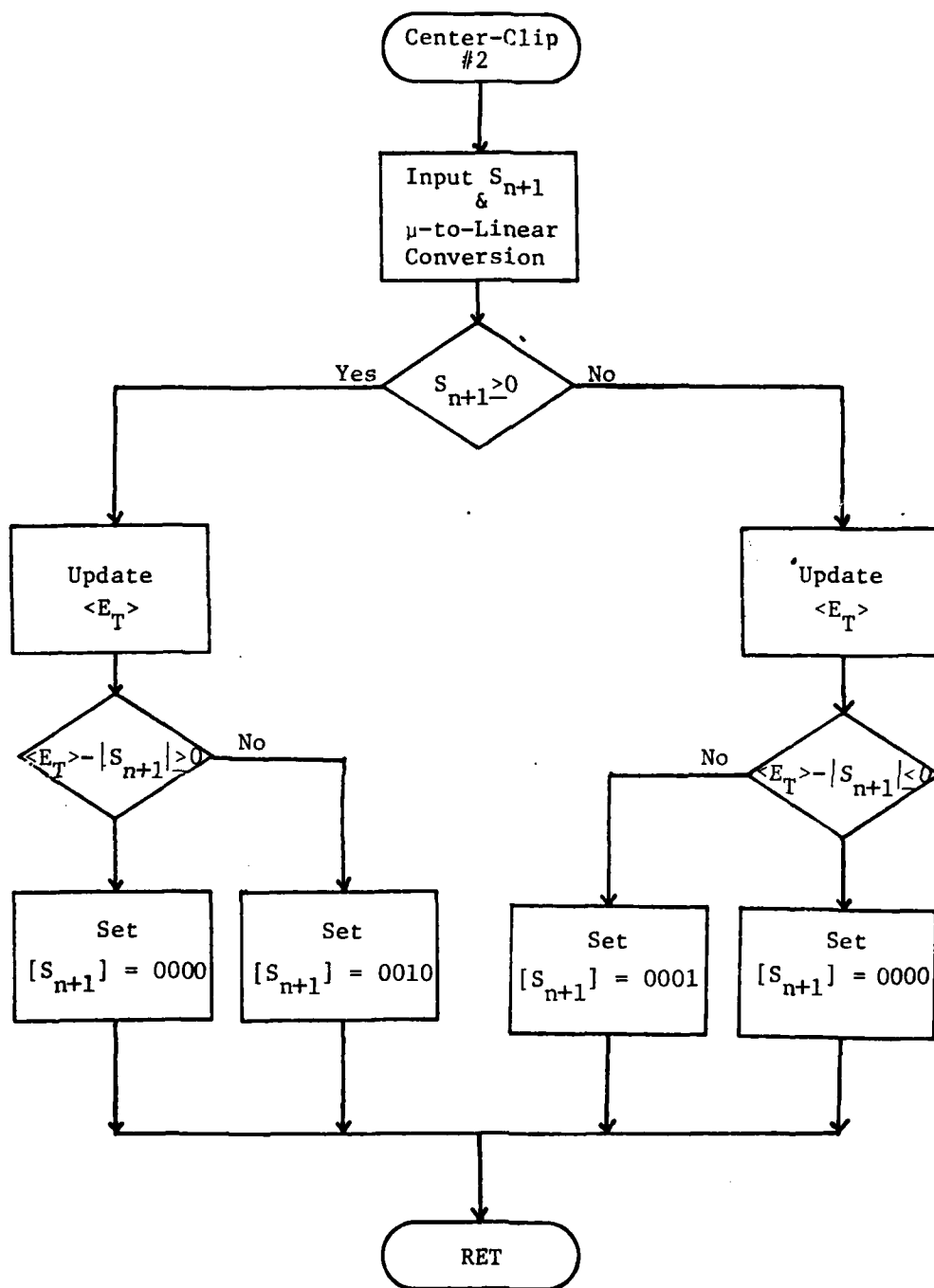


Fig. B.10 Center-Clip #2 for Transmitter

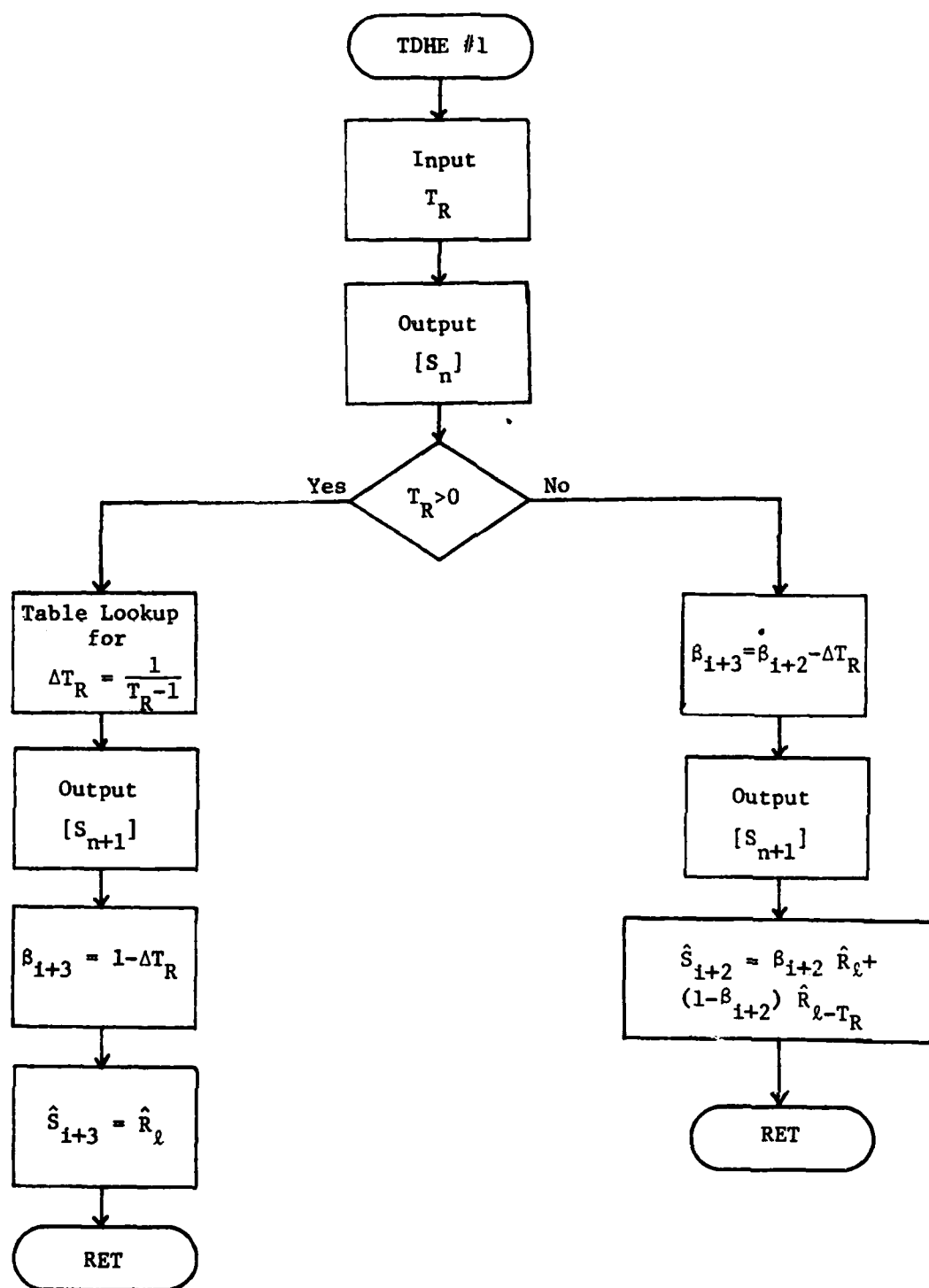


Fig. B.11 TDHE #1 Routine

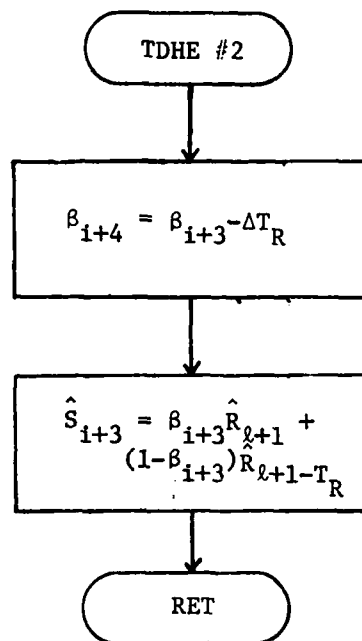


Fig. B.12 TDHE #2 Routine

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0001 0000      MPROG                      ;
0002 0000 /* *****
0002 0000      *
0002 0000      *                      TDHS - ARC
0002 0000      *
0002 0000 *****
0002 0000
0002 0000      AUTHOR: MAULIN YEH
0002 0000      DATE: 9/8/83
0002 0000
0002 0000      PROGRAM: TDHS14.SRC
0002 0000
0002 0000      VERSION #1: 12/13/82
0002 0000                      ARC TRANSMITTER & RECEIVER
0002 0000                      TDHS TRANSMITTER
0002 0000                      CENTER-CLIP TRANSMITTER & RECEIVER
0002 0000
0002 0000      VERSION #2: 3/11/83
0002 0000                      ARC TRANSMITTER & RECEIVER
0002 0000                      INCLUDE TDHS RECEIVER
0002 0000                      INCLUDE LINEAR-TO-MU CONVERTER
0002 0000
0002 0000      VERSION #3: 6/29/83
0002 0000                      NEW CODE WORDS
0002 0000                      4TH-ORDER FIXED PREDICTOR
0002 0000
0002 0000      VERSION #4: 7/1/83
0002 0000                      ARCJ ALGORITHM
0002 0000
0002 0000      ABSTRACT:
0002 0000          TDHS: A CENTER-CLIP ROUTINE IS USED TO PROVIDE CENTER-CLIP
0002 0000          VALUES TO THE PITCH EXTRACTION HARDWARE. A FIRST-ORDER
0002 0000          FILTER IS EMPLOYED TO CALCULATE THE ADAPTIVE THRESHOLD
0002 0000          WITH A TIME CONSTANT ABOUT 12ms.
0002 0000          ARC: THE ADAPTIVE QUANTIZER USES A FIRST-ORDER FILTER TO
0002 0000          ESTIMATE THE ADAPTIVE QUANTIZER STEP SIZE WITH A TIME
0002 0000          CONSTANT ABOUT 10ms. IT CAN BE VERIFIED THAT THE
0002 0000          MAXIMUM DISCREPANCY ON THE ADAPTIVE QUANTIZER STEP SIZE
0002 0000          BETWEEN THE TRANSMITTER AND THE RECEIVER DUE TO A SINGLE
0002 0000          CHANNEL ERROR IS 7.4%. ALSO, ONCE A DISCREPANCY OCCURS,
0002 0000          IT WILL REMAIN THERE UNTILL REACHING THE MINIMUM QUANTIZER
0002 0000          STEP SIZE WHICH IS SET TO BE 32. NOTE THAT THERE IS NO
0002 0000          OVERFLOW CONTROL ON THIS PARAMETER.
0002 0000          THE LINEAR PREDICTOR INVOLVES THREE PREDICTORS. A 4th-ORDER
0002 0000          RECURSIVE VOICE-TYPE PREDICTOR EMPHASIZES THE BANDWIDTH

```


(TDHS W 4)

STNO LOC. SOURCE STATEMENT

```

0002 0000          BETWEEN 700Hz TO 1100Hz. A 4th-ORDER RECURSIVE MODER-TYPE
0002 0000          PREDICTOR EMPHASIZES THE BANDWIDTH BETWEEN 600Hz TO 3400Hz.
0002 0000          A 4th-ORDER ADAPTIVE NON-RECURSIVE PREDICTOR EMPHASIZES
0002 0000          THE REGION WHERE PREDICTOR ERROR SIGNALS HAS DOMINANT ENERGY.
0002 0000          IT CAN BE VARIFIED THAT THE ADAPTIVE PREDICTOR COEFFICIENTS
0002 0000          ARE ABSOLUTELY TRACKING BETWEEN TRANSMITTER AND RECEIVER.
0002 0000          THE SYSTEM IS ABSOLUTELY STABLE.
0002 0000          */
0002 0000          ORG          00H
0003 0000          DRABS      EQUAL      0180H          ; /* ABS. AND TABLE-LOOK-UP */
0004 0000          RPINI      EQUAL      0A3H          ;
0005 0000          INTERP      EQUAL      000011110000000B      ; /* ENABLE INTERRUPT */
0006 0000          SR16I      EQUAL      0000101100000001B      ; /* 16-BIT TRANSFER */
0007 0000          SR8I       EQUAL      0000111100000001B      ;
0008 0000          SR16O      EQUAL      00001011000000010B      ;
0009 0000          SR8O       EQUAL      00001111000000010B      ;
0010 0000          RPLTU      EQUAL      0B2H          ;
0011 0000          ZERO      EQUAL      0000H          ;
0012 0000          DRINI      EQUAL      0100H          ; /* TOP OF A-TO-LINEAR TABLE */
0013 0000          RMSMIN     EQUAL      080          ; /* RMSMIN */
0014 0000          TRP        EQUAL      079H          ; /* T-TRANSMITTER POINTER+BASE */
0015 0000          RRP        EQUAL      0C9H          ; /* T-RECEIVER POINTER+BASE */
0016 0000          RPTH       EQUAL      0FDH          ;
0017 0000          RP2        EQUAL      0C8H          ;
0018 0000          RPPRE      EQUAL      0BDH          ;
0019 0000          TABLE     EQUAL      0E7H          ;
0020 0000          RPE        EQUAL      0F6H          ;
0021 0000          RPQS       EQUAL      0D7H          ;
0022 0000          ZEROC      EQUAL      0E008H          ;
0023 0000          DPQ1       EQUAL      7FH          ;
0024 0000          DPQ2       EQUAL      71H          ;
0025 0000          DPQ3       EQUAL      2FH          ;
0026 0000          START:     LDI          @DP,ZERO          ; /* DP=0 */
0027 0001                      LDI          @SR,SR80          ;
0028 0002                      LDI          @SON,0FFFFH          ;
0029 0003                      LDI          @A,ZERO          ; /* A=0 */
0030 0004          LOOP:      OP          MOV          @MEM,A          ; /* MEM(0X)=0 */
0030 0004                      M1
0031 0005                      OP          MOV          @MEM,A          ;
0031 0005                      M3          ; /* MEM(DP=1X)=0 */
0032 0006                      OP          MOV          @MEM,A          ; /* MEM(2X)=0 */
0032 0006                      M1
0033 0007                      OP          MOV          @MEM,A          ; /* MEM(DP=3X)=0 */
0033 0007                      M7

```

(TDHS W 4)

SYNO LOC. SOURCE STATEMENT

```

0034 0008      OP      MOV      @MEM,A      /* MEM(4X)=0 */
0034 0008              M1                      ;
0035 0009      OP      MOV      @MEM,A      /* MEM(5X)=0 */
0035 0009              M3                      ;
0036 000A      OP      MOV      @MEM,A      /* MEM(6X)=0 */
0036 000A              M1                      ;
0037 000B      OP      MOV      @MEM,A      /* MEM(7X)=0 */
0037 000B              M7                      ;
0038 000C      JDPLF    CONT                      ;
0039 000D      OP      DPINC                      ;
0040 000E      JMP      LOOP                      ;
0041 000F      CONT:   LDI      @DP,42H          ;
0042 0010              LDI      @MEM,RMSMIN      ;
0043 0011              LDI      @DP,4DH          ;
0044 0012              LDI      @MEM,RMSMIN      ;
0045 0013              LDI      @DP,52H          ;
0046 0014              LDI      @MEM,7FFFH       ;
0047 0015              LDI      @DP,5DH          ;
0048 0016              LDI      @MEM,7FFFH       ;
0049 0017              LDI      @DP,071H         ;
0050 0018              LDI      @MEM,ZEROC       ;
0051 0019              LDI      @DP,7FH          ;
0052 001A              LDI      @MEM,ZEROC       ;
0053 001B              LDI      @DP,00H          ;
0054 001C              LDI      @MEM,0FFFFH      ;
0055 001D              LDI      @DP,21H          ;
0056 001E              LDI      @MEM,0FFFFH      ;
0057 001F              LDI      @DP,41H          ;
0058 0020      OP      MOV      @DR,MEM          ;
0058 0020              DPDEC                      ;
0058 0020              M6                      ;
0059 0021      CALL    DUMY                      ;
0060 0022      OP      MOV      @DR,MEM          ;
0060 0022              DPINC                      ;
0060 0022              M1                      ;
0061 0023      CALL    DUMY                      ;
0062 0024      OP      MOV      @DR,MEM          ;
0062 0024              DPDEC                      ;
0062 0024              M2                      ;
0063 0025      CALL    DUMY                      ;
0064 0026      OP      MOV      @DR,MEM          ;
0064 0026              DPINC                      ;
0064 0026              M6                      ;
0065 0027      CALL    DUMY                      ;
0066 0028      OP      MOV      @DR,MEM          ;

```

AD-A135 233

REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION
ALGORITHM COMBINING TIM..(U) NOTRE DAME UNIV IN DEPT OF
ELECTRICAL ENGINEERING J L MELSA ET AL. JUN 83

2/2

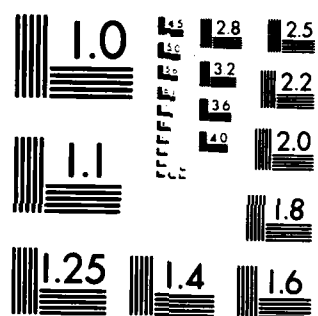
UNCLASSIFIED

DCA100-82-C-0026

F/G 9/2

NL

END
DATE
FILMED
12 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0066 0028                DPDEC                ;
0067 0029                OP                DPDEC
0067 0029                M5                ;
0068 002A                CALL             DUMY                ;
0069 002B                OP                MOV                QDR,MEM
0069 002B                M1                ;
0070 002C                CALL             DUMY                ;
0071 002D                LDI              @SR,SR160           ;
0072 002E                OP                MOV                QDR,MEM
0073 002F  WAIT2:        CALL             DUMY                ;
0074 0030                CALL             DUMY                ;
0075 0031  WAIT:         LDI              @SR,INTERP          ;
0076 0032  WAIT1:        JMP              WAIT1               ;
0077 0033  DUMY:         OP                NOP                ;
0078 0034                OP                NOP                ;
0079 0035                OP                NOP                ;
0080 0036                OP                NOP                ;
0081 0037                OP                NOP                ;
0082 0038                OP                NOP                ;
0082 0038                RET                ;
0083 0039  LTU:         LDI              @TR,ZERO             ;
0084 003A                LDI              @RP,RPLTU           ;
0085 003B                OP                MOV                @NON,B
0085 003B                XOR                ACCB,IDB           ;
0086 003C                JNSAO            SPOS                ;
0087 003D                OP                SUB                ACCB,IDB
0087 003D                MOV                @NON,A            ;
0088 003E                OP                MOV                @A,B
0088 003E                XOR                ACCB,IDB           ;
0089 003F                LDI              @TR,0080H           ;
0090 0040  SPOS:        LDI              @B,-6                ;
0091 0041                OP                MOV                @K,A
0092 0042                OP                SUB                ACCA,IDB
0092 0042                MOV                @NON,RO           ;
0092 0042                RPDEC                ;
0093 0043                OP                MOV                QDR,MEM      /* S(N)/[C^R(N)] */
0093 0043                M1                ;
0094 0044                JSAO             C7                ;
0095 0045                LDI              @A,007FH           ;
0096 0046                OP                ADD                ACCA,IDB
0096 0046                MOV                @NON,TR           ; /* ADD SIGN BIT */
0097 0047                JMP              BIT                ;
0098 0048  C7:         OP                MOV                @A,K
0099 0049                OP                SUB                ACCA,IDB      /* FIND CORD */
0099 0049                MOV                @NON,RO

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0099 0049                RPDEC                ;
0100 004A                C6                    ;
0101 004B                JSA0                  ;
0101 004B                OP                    ;
0101 004B                ADD    ACCB,IDB        ;
0101 004B                MOV    @NON,RP        ;
0102 004C                OP                    ;
0102 004C                XCHG   ACCA            /* STEP BITS */
0102 004C                MOV    @RP,B          ;
0103 004D                OP                    ;
0103 004D                ADD    ACCA,IDB        ;
0103 004D                MOV    @NON,RO        /* ADD CORD */
0104 004E                OP                    ;
0104 004E                ADD    ACCA,IDB        ;
0104 004E                MOV    @NON,TR        /* ADD SIGN BIT */
0105 004F                JMP     BIT            /* JMP TO INVERT BIT */
0106 0050    C6:         OP                    ;
0107 0051                OP                    ;
0107 0051                MOV    @A,K            ;
0107 0051                SUB    ACCA,IDB        /* FIND CORD */
0107 0051                MOV    @NON,RO        ;
0107 0051                RPDEC                ;
0108 0052                JSA0                  ;
0108 0052                C5                    ;
0109 0053                OP                    ;
0109 0053                SHL1   ACCA            ;
0110 0054                OP                    ;
0110 0054                ADD    ACCB,IDB        ;
0110 0054                MOV    @NON,RP        ;
0111 0055                OP                    ;
0111 0055                XCHG   ACCA            /* STEP BITS */
0111 0055                MOV    @RP,B          ;
0112 0056                OP                    ;
0112 0056                ADD    ACCA,IDB        ;
0112 0056                MOV    @NON,RO        /* ADD CORD */
0113 0057                OP                    ;
0113 0057                ADD    ACCA,IDB        ;
0113 0057                MOV    @NON,TR        /* ADD SIGN BIT */
0114 0058                JMP     BIT            /* JMP TO INVERT BIT */
0115 0059    C5:         OP                    ;
0116 005A                OP                    ;
0116 005A                MOV    @A,K            ;
0116 005A                SUB    ACCA,IDB        /* FIND CORD */
0116 005A                MOV    @NON,RO        ;
0116 005A                RPDEC                ;
0117 005B                JSA0                  ;
0117 005B                C4                    ;
0118 005C                OP                    ;
0118 005C                SHL2   ACCA            ;
0119 005D                OP                    ;
0119 005D                ADD    ACCB,IDB        ;
0119 005D                MOV    @NON,RP        ;
0120 005E                OP                    ;
0120 005E                XCHG   ACCA            /* STEP BITS */
0120 005E                MOV    @RP,B          ;
0121 005F                OP                    ;
0121 005F                ADD    ACCA,IDB        ;
0121 005F                MOV    @NON,RO        /* ADD CORD */
0122 0060                OP                    ;
0122 0060                ADD    ACCA,IDB        ;
0122 0060                MOV    @NON,TR        /* ADD SIGN BIT */
0123 0061                JMP     BIT            /* JMP TO INVERT BIT */
0124 0062    C4:         OP                    ;
0125 0063                OP                    ;
0125 0063                MOV    @A,K            ;
0125 0063                SUB    ACCA,IDB        /* FIND CORD */
0125 0063                MOV    @NON,RO        ;
0125 0063                RPDEC                ;

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0126 0064      JSA0    C3
0127 0065      OP      SHL1    ACCA
0128 0066      OP      SHL2    ACCA
0129 0067      OP      ADD     ACCB,IDB
0129 0067      OP      MOV     @NON,RP
0130 0068      OP      XCHG    ACCA      /* STEP BITS */
0130 0068      OP      MOV     @RP,B
0131 0069      OP      ADD     ACCA,IDB
0131 0069      OP      MOV     @NON,RO    /* ADD CORD */
0132 006A      OP      ADD     ACCA,IDB
0132 006A      OP      MOV     @NON,TR    /* ADD SIGN BIT */
0133 006B      JMP      BIT          /* JMP TO INVERT BIT */
0134 006C      C3:     OP      MOV     @A,K
0135 006D      OP      SUB     ACCA,IDB    /* FIND CORD */
0135 006D      OP      MOV     @NON,RO
0135 006D      RPDEC
0136 006E      JSA0    C2
0137 006F      OP      SHL4    ACCA
0138 0070      OP      ADD     ACCB,IDB
0138 0070      OP      MOV     @NON,RP
0139 0071      OP      XCHG    ACCA      /* STEP BITS */
0139 0071      OP      MOV     @RP,B
0140 0072      OP      ADD     ACCA,IDB
0140 0072      OP      MOV     @NON,RO    /* ADD CORD */
0141 0073      OP      ADD     ACCA,IDB
0141 0073      OP      MOV     @NON,TR    /* ADD SIGN BIT */
0142 0074      JMP      BIT          /* JMP TO INVERT BIT */
0143 0075      C2:     OP      MOV     @A,K
0144 0076      OP      SUB     ACCA,IDB    /* FIND CORD */
0144 0076      OP      MOV     @NON,RO
0144 0076      RPDEC
0145 0077      JSA0    C1
0146 0078      OP      SHR1    ACCA
0147 0079      OP      SHR1    ACCA
0148 007A      OP      ADD     ACCB,IDB
0148 007A      OP      MOV     @NON,RP
0149 007B      OP      SHR1    ACCA      /* STEP BITS */
0149 007B      OP      MOV     @RP,B
0150 007C      OP      ADD     ACCA,IDB
0150 007C      OP      MOV     @NON,RO    /* ADD CORD */
0151 007D      OP      ADD     ACCA,IDB
0151 007D      OP      MOV     @NON,TR    /* ADD SIGN BIT */
0152 007E      JMP      BIT          /* JMP TO INVERT BIT */
0153 007F      C1:     OP      MOV     @A,K
0154 0080      OP      SUB     ACCA,IDB    /* FIND CORD */

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0154 0080          MOV      QNON,RO
0154 0080          RPDEC
0155 0081          JSAO     CO
0156 0082          OP      SHR1  ACCA
0157 0083          OP      SHR1  ACCA
0158 0084          OP      ADD    ACCB,IDB
0158 0084          OP      MOV    QNON,RP
0159 0085          OP      MOV    QRP,B
0160 0086          OP      ADD    ACCA,IDB
0160 0086          OP      MOV    QNON,RO      ; /* ADD CORD */
0161 0087          OP      ADD    ACCA,IDB
0161 0087          OP      MOV    QNON,TR      ; /* ADD SIGN BIT */
0162 0088          JMP      BIT              ; /* JMP TO INVERT BIT */
0163 0089  CO:      OP      MOV    QA,K
0164 008A          OP      SHR1  ACCA
0165 008B          OP      ADD    ACCA,IDB
0165 008B          OP      MOV    QNON,TR      ; /* ADD SIGN BIT */
0166 008C  BIT:    LDI      QB,00FFH
0167 008D          OP      XOR    ACCB,IDB
0167 008D          OP      MOV    QNON,A      ; /* GET NON-INVERT BITS */
0168 008E          OP      XCHG   ACCB
0168 008E          RET
0169 008F  CCLIP:  LDI      QB,DRABS
0170 0090          OP      OR     ACCB,IDB      /* ABS. VALUE OF SN */
0170 0090          OP      MOV    QA,SIN      ; /* INPUT SN */
0171 0091          OP      XOR    ACCB,IDB      /* RESET ACCB */
0171 0091          OP      MOV    QRP,B      ; /* MU-TO-LINEAR */
0172 0092          OP      XCHG   ACCA      /* CHECK SIGN OF SN */
0172 0092          OP      MOV    QMEM,RO      ; /* ABS. VALUE OF SN */
0173 0093          JNSAO    T1NEG      ; /* JUMP IF SN < 0 */
0174 0094          OP      INC     ACCB
0175 0095  T1NEG:  LDI      QRP,RPINI
0176 0096          OP      XOR    ACCA,IDB
0176 0096          OP      MOV    QNON,A
0177 0097          OP      MOV    QKLR,MEM      /* [SN]*(1-ALFA1) */
0177 0097          OP      ADD    ACCA,RAM      /* A=[SN] */
0177 0097          RPDEC
0177 0097          M1
0178 0098          OP      SUB     ACCA,M      /* A=[SN]-[SN]*(1-ALFA1) */
0178 0098          OP      MOV    QKLR,MEM      ; /* M=<E>*(1-ALFA2) */
0179 0099          OP      ADD     ACCA,M      ; /* NEW <E> */
0180 009A          OP      MOV    QMEM,A      /* STORE NEW <E> */
0180 009A          OP      M1
0181 009B          OP      SUB     ACCA,RAM      ; /* CHECK <E>-[SN] */
0182 009C          JNSAO    T100      ; /* JUMP IF >= 0 */

```


(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0183 009D T101: OP INC ACCB
0183 009D RET ;
0184 009E T100: OP XOR ACCB,IDB ;
0184 009E MOV @NON,B ;
0184 009E RET ;
0185 009F CON5: OP ADD ACCB,M ;
0185 009F MOV @KLM,MEM /* B(2)/2*~E(N-1)/2 */
0185 009F M1 ;
0186 00A0 OP ADD ACCB,M ;
0186 00A0 MOV @KLM,MEM ; /* B(1)/2*~E(N)/2 */
0187 00A1 LDI @DP,50H ;
0188 00A2 OP ADD ACCB,M ;
0188 00A2 MOV @MEN,DR /* ~R(N+1) */
0189 00A3 LDI @DP,0DH ;
0190 00A4 OP MOV @NON,A ;
0190 00A4 XOR ACCA,IDB ;
0191 00A5 OP SHL1 ACCB ;
0192 00A6 OP MOV @MEN,B /* STORE ~P(N+1)/2 */
0192 00A6 XOR ACCB,IDB ;
0192 00A6 M3 ;
0193 00A7 /* LDI @DP,5FH ; TDHS LOOP BACK */
0193 00A7 OP NOP ;
0194 00A8 OP ADD ACCA,RAM ;
0194 00A8 MOV @TR,MEM ;
0194 00A8 DPCLR ;
0194 00A8 M3 ;
0195 00A9 LDI @DP,00H ; /* TDHS LOOP BACK */
0196 00AA JSAO RNEG ;
0197 00AB OP INC ACCB ;
0197 00AB DPDEC ;
0198 00AC OP MOV @MEN,A ;
0198 00AC XOR ACCA,IDB ;
0199 00AD JMP RCLIP ;
0200 00AE RNEG: OP SUB ACCB,IDB ;
0200 00AE MOV @NON,A ;
0200 00AE DPDEC ;
0201 00AF OP MOV @MEN,B ;
0201 00AF XOR ACCB,IDB ;
0202 00B0 OP MOV @A,B ;
0203 00B1 RCLIP: LDI @RP,RPINI ;
0204 00B2 OP MOV @KLR,MEM /* [RN]*(1-ALFA1) */
0204 00B2 ADD ACCA,RAM /* [RN] */
0204 00B2 M1 ;
0204 00B2 RPDEC ;
0205 00B3 OP SUB ACCA,M /* A=[RN]-[RN]*(1-ALFA1) */

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0205 00B3      MOV      @KLR,MEN      /* M=<E>*(1-ALFA2) */
0205 00B3      DPCLR
0205 00B3      M7
0206 00B4      OP      MOV      @MEN,DR      /* ~R(N-T) */
0206 00B4      DPDEC
0206 00B4      M7
0207 00B5      OP      ADD      ACCA,M      /* NEW <E> */
0208 00B6      OP      MOV      @MEN,A      /* STORE NEW <E> */
0208 00B6      M1
0209 00B7      OP      SUB      ACCA,RAM      /* CHECK <E>-[SN] */
0209 00B7      M3
0210 00B8      JSAO
0211 00B9      R00:    OP      MOV      @MEN,TR
0211 00B9      M1      /* STORE R(N) */
0212 00BA      OP      MOV      @NON,B
0212 00BA      XOR      ACCB,IDB
0213 00BB      OP      MOV      @MEN,B
0213 00BB      DPCLR
0213 00BB      M5      /* STORE [R(N)] */
0214 00BC      JMP
0215 00BD      R01:    OP      MOV      @MEN,TR      /* STORE R(N) */
0215 00BD      SHL2      ACCB
0215 00BD      M1
0216 00BE      OP      INC      ACCB
0217 00BF      OP      MOV      @MEN,B
0217 00BF      DPCLR
0217 00BF      M5
0218 00C0      CLIPT2: JNSIAK CLIPT2      /* WAIT */
0219 00C1      LDI      @SR,SR8I
0220 00C2      OP      MOV      @MEN,DR      /* ~R(N-T+1) */
0220 00C2      M7
0220 00C2      DPINC
0221 00C3      CALL
0221 00C3      OP      CCLIP
0222 00C4      OP      DPDEC
0223 00C5      OP      MOV      @SON,MEN      /* OUT ~S(N-1) */
0223 00C5      M1
0224 00C6      OP      MOV      @MEN,SIN      /* S(N+1) */
0224 00C6      M3
0225 00C7      OP      MOV      @MEN,B      /* [S(N+1)] */
0225 00C7      DPINC
0225 00C7      M6
0226 00C8      TDHE:   LDI      @B,00FFH
0227 00C9      OP      MOV      @NON,DRNF
0227 00C9      AND      ACCB,IDB      /* T */
0228 00CA      LDI      @SR,SR8D

```

(TDHS N 4)

STNO	LOC.	SOURCE	STATEMENT
0229	00CB	OP	MOV QDR, MEM /* [S(N)] */
0229	00CB		DPDEC
0229	00CB		M7 ;
0230	00CC	OP	XCHG ACCB ;
0230	00CC		MOV QTR, B ;
0231	00CD	JSBO	RNDEL
0232	00CE	RPDEL: LDI	QB, RRP ; /* JUMP IF NO CHANGE */
0233	00CF	OP	SUB ACCB, IDB ;
0233	00CF		MOV QNON, TR /* T-POINTER */
0233	00CF		M7 ;
0234	00D0	OP	MOV QRP, B ;
0235	00D1	LDI	QA, ZERO ;
0236	00D2	OP	MOV QB, RD ;
0236	00D2		ADD ACCA, RAM
0236	00D2		M6 ; /* ~S(N)=~R(K) */
0237	00D3	OP	SHR1 ACCB ;
0238	00D4	OP	MOV QDR, MEM /* [S(N+1)] */
0238	00D4		SHR1 ACCB
0238	00D4		DPDEC
0238	00D4		M6 ;
0239	00D5	OP	SHR1 ACCB ;
0240	00D6	OP	MOV QMEN, B ;
0241	00D7	LDI	QB, -32768 ;
0242	00D8	OP	SUB ACCB, RAM ;
0242	00D8		DPINC
0242	00D8		M7 ;
0243	00D9	OP	MOV QMEN, B /* STORE BETA(I+1) */
0243	00D9		DPINC ;
0244	00DA	JMP	CTDHE ;
0245	00DB	RNDEL: OP	MOV QA, MEM /* BETA(I) */
0245	00DB		DPDEC
0245	00DB		M7 ;
0246	00DC	OP	SUB ACCA, RAM /* BETA(I+1) */
0246	00DC		MOV BL, A
0246	00DC		DPINC
0246	00DC		M7 ;
0247	00DD	OP	MOV QMEN, A /* STORE BETA(I+1) */
0247	00DD		M1 ;
0248	00DE	OP	MOV QDR, MEM /* [S(N+1)] */
0248	00DE		M6 ;
0249	00DF	OP	MOV QB, MEM /* GET ~R(K) */
0249	00DF		M2 ;
0250	00E0	OP	SUB ACCB, RAM /* ~R(K)-~R(K-T) */
0250	00E0		MOV QA, MEM /* ~R(K-T) */
0250	00E0		DPINC

(TDHS W 4)

STNO LOC. SOURCE STATEMENT

```

0250 00E0      M5
0251 00E1      OP      MOV      QK,B      ; /* ~R(K)-~R(K-T) */
0252 00E2      OP      ADD      ACCA,M      ; /* ~S(N) */
0253 00E3      CTDHE: CALL      LTU      ;
0254 00E4      OP      MOV      QMEN,B      ; /* ~S(N) */
0254 00E4      DPDEC
0254 00E4      M3      ;
0255 00E5      OP      MOV      QDR,MEN      ; /* S(N+1) */
0255 00E5      M2      ;
0256 00E6      OP      MOV      QA,MEN      ; /* BETA(I) */
0256 00E6      DPDEC
0256 00E6      M7      ;
0257 00E7      OP      SUB      ACCA,RAM      ; /* BETA(I+1) */
0257 00E7      MOV      QL,A      ;
0257 00E7      DPINC
0257 00E7      M7      ;
0258 00E8      OP      MOV      QMEN,A      ; /* STORE BETA(I+1) */
0258 00E8      M6      ;
0259 00E9      OP      MOV      QB,MEN      ; /* GET ~R(K+1) */
0259 00E9      M2      ;
0260 00EA      OP      SUB      ACCB,RAM      ; /* ~R(K+1)-~R(K-T+1) */
0260 00EA      MOV      QA,MEN      ; /* ~R(K-T+1) */
0260 00EA      DPINC
0261 00EB      OP      MOV      QK,B      ; /* ~R(K+1)-~R(K-T+1) */
0261 00EB      XOR      ACCB,IDB      ;
0262 00EC      OP      ADD      ACCA,M      ; /* ~S(N) */
0263 00ED      LDI      QB,ZERO      ; /* ZERO CODE WORD */
0264 00EE      OP      SUB      ACCB,RAM      ;
0265 00EF      JNZB      OUTC      ;
0266 00F0      LDI      QB,000FH      ;
0267 00F1      LDI      QDP,DPQ1      ;
0268 00F2      OP      XOR      ACCB,RAM      ;
0269 00F3      OP      MOV      QMEN,B      ;
0270 00F4      LDI      QDP,DPQ2      ;
0271 00F5      OP      MOV      QMEN,B      ;
0272 00F6      OUTC: OP      MOV      QDR,MEN      ; /* Q(K) */
0273 00F7      LDI      QDP,DPQ3      ;
0274 00F8      CALL      LTU      ;
0275 00F9      LDI      QSR,SR160      ;
0276 00FA      OP      MOV      QDR,MEN      ; /* ~R(N) */
0276 00FA      DPINC
0276 00FA      M3      ;
0277 00FB      OP      MOV      QMEN,B      ; /* ~S(N+1) */
0278 00FC      JMP      WAIT2      ;
0279 00FD      ORG      100H      ;

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0280 0100 BEGIN: LDI      @DP,01H      ;
0281 0101      LDI      @SR,SR8I      ;
0282 0102      OP       MOV      @NON,DR      ; /* DUMMY */
0283 0103      CALL     CCLIP      ;
0284 0104 TDHC:  LDI      @A,DRINI      ;
0285 0105      OP       MOV      @NON,DR      ;
0285 0105      OR       ACCA,IDB      ; /* INPUT S(I) */
0285 0105      M2              ;
0286 0106      OP       MOV      @SON,MEN      ; /* OUTPUT ~S(N) */
0286 0106      M1              ;
0287 0107      OP       MOV      @MEN,SIN      ; /* IN S(I) */
0287 0107      M7              ;
0288 0108      OP       MOV      @MEN,B      ; /* [S(N)] CLIP */
0288 0108      M1              ;
0289 0109      OP       NOP              ;
0290 010A      OP       MOV      @RP,A      ;
0291 010B      LDI      @B,00FFH      ;
0292 010C      OP       MOV      @NON,DR      ;
0292 010C      AND      ACCB,IDB      ; /* T */
0293 010D      OP       XCHG      ACCB      ;
0293 010D      MOV      @TR,B      ;
0294 010E      JSBO     TNDLDT      ; /* JUMP IF NO CHANGE */
0295 010F TPDELT: LDI      @A,TRP      ;
0296 0110      OP       SUB      ACCA,IDB      ;
0296 0110      MOV      @NON,TR      ; /* T-POINTER */
0297 0111      OP       MOV      @B,RO      ; /* R(I) */
0298 0112      OP       MOV      @RP,A      ;
0298 0112      M3              ;
0299 0113      OP       MOV      @A,RO      ;
0300 0114      OP       SHR1      ACCA      ;
0301 0115      OP       SHR1      ACCA      ;
0302 0116      OP       MOV      @NON,DR      ;
0302 0116      SHR1      ACCA      ;
0303 0117      OP       MOV      @MEN,A      ;
0304 0118      LDI      @A,-32768      ;
0305 0119      OP       SUB      ACCA,RAM      ;
0305 0119      M3              ;
0306 011A      OP       MOV      @MEN,A      ; /* STORE BETA(I+1) */
0306 011A      M5              ;
0306 011A      DPINC      ;
0307 011B      JMP      ARC      ;
0308 011C TDDEL: OP       MOV      @A,MEN      ; /* BETA(I) */
0308 011C      M3              ;
0309 011D      OP       SUB      ACCA,RAM      ; /* BETA(I+1) */
0309 011D      MOV      @L,A

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0309 011D      M3
0310 011E      OP      MOV      @MEN,A      ; /* STORE BETA(I+1) */
0310 011E      M5
0310 011E      DPINC
0311 011F      LDI      @B,DRINI
0312 0120      OP      MOV      @NON,DR
0312 0120      OR      ACCB,IDB      ; /* INPUT S(I+T) */
0313 0121      OP      MOV      @A,RO      ; /* GET S(I) */
0314 0122      OP      MOV      @RP,B      ; /* A-TO-LINEAR */
0315 0123      OP      SUB      ACCA,IDB      ; /* S(I)-S(I+T) */
0315 0123      MOV      @B,RO      ; /* S(I+T) */
0316 0124      OP      MOV      @K,A      ; /* S(I)-S(I+T) */
0317 0125      OP      ADD      ACCB,M      ; /* R(I) */
0318 0126      ARC:    OP      SHR1      ACCB
0319 0127      OP      SHR1      ACCB
0320 0128      LDI      @DP,5FH      ; /* TDHS LOOP BACK */
0321 0129      OP      MOV      @MEN,B      ; /* TDHS LOOP BACK */
0322 012A      LDI      @DP,02H      ; /* TDHS LOOP BACK */
0323 012B      LDI      @RP,RPTH
0324 012C      /*      QUANTIZER
0324 012C      OP      SUB      ACCB,RAM      /* E(N)/2=S(N)/2-P(N)/2 */
0324 012C      MOV      @KLM,RO      /* <E>*TH1/2 */
0324 012C      DPDEC
0324 012C      M7
0324 012C      RPDEC
0325 012D      JSBO      TENEG
0326 012E      TEPOS:  LDI      @TR,-22
0327 012F      JMP      TQ1
0328 0130      TENEG:  OP      XOR      ACCB,IDB
0328 0130      MOV      @TR,B
0329 0131      OP      SUB      ACCB,IDB
0329 0131      MOV      @NON,TR
0330 0132      LDI      @TR,-15
0331 0133      TQ1:    OP      MOV      @L,RO      /* <E>*TH2/2 */
0331 0133      SUB      ACCB,M
0331 0133      RPDEC
0332 0134      JNSBO      TQ2
0333 0135      LDI      @TR,-15
0334 0136      JMP      TQ1
0335 0137      TQ2:    OP      MOV      @L,RO      /* <E>*TH3/2 */
0335 0137      SUB      ACCB,M
0335 0137      RPDEC
0336 0138      JSBO      TQ1
0337 0139      OP      MOV      @L,RO      /* <E>*TH4/2 */
0337 0139      SUB      ACCB,M

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0337 0139          RPDEC          ;
0338 013A          JSB0          TQP1          ;
0339 013B          OP            MOV            @L,RO          /* <E>*TH5/2 */
0339 013B          SUB            ACCB,M          ;
0339 013B          RPDEC          ;
0340 013C          JSB0          TQP1          ;
0341 013D          OP            MOV            @L,RO          /* <E>*TH6/2 */
0341 013D          SUB            ACCB,M          ;
0341 013D          RPDEC          ;
0342 013E          JSB0          TQP1          ;
0343 013F          OP            MOV            @L,RO          /* <E>*TH7/2 */
0343 013F          SUB            ACCB,M          ;
0343 013F          RPDEC          ;
0344 0140          JSB0          TQP1          ;
0345 0141          OP            SUB            ACCB,M          ;
0345 0141          RPDEC          ;
0346 0142          JSB0          TQP1          ;
0347 0143          OP            RPDEC          ;
0348 0144          /* INVERSE-QUANTIZER WITH BIAS CHECK */
0348 0144          TQP1: LDI            @B,-05          ;
0349 0145          OP            ADD            ACCB,IDB          ;
0349 0145          MOV            @NON,RP          ; /* ADJUST RP */
0350 0146          OP            MOV            @RP,B          ;
0351 0147          OP            MOV            @L,RO          ;
0352 0148          OP            ADD            ACCB,IDB          ;
0352 0148          MOV            @NON,TR          ;
0353 0149          OP            MOV            @RP,B          ;
0354 014A          OP            MOV            @A,RO          ;
0355 014B          OP            XCHG          ACCA          ;
0356 014C          OP            MOV            @MEM,A          ;
0356 014C          XOR            ACCA,IDB          ;
0356 014C          DPINC          ;
0356 014C          M3          ;
0357 014D          LDI            @B,-16          ;
0358 014E          OP            ADD            ACCB,IDB          ;
0358 014E          MOV            @NON,RP          ;
0359 014F          CALL          COMN          ;
0360 0150          JMP            CON6          ;
0361 0151          COMN: OP            MOV            @RP,B          ;
0361 0151          XOR            ACCB,IDB          ;
0362 0152          OP            ADD            ACCB,M          ;
0363 0153          OP            SHL1          ACCB          ;
0363 0153          MOV            @L,RO          ; /* CALCULATE "E(N)/8 */
0364 0154          LDI            @RP,RP2          ;
0365 0155          OP            SUB            ACCB,IDB

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0365 0155      MOV      @NON,RO      ;
0366 0156      JSB0     TBIAS      ;
0367 0157      OP      ADD      ACCB,IDB      ;
0367 0157      MOV      @NON,RO      ;
0368 0158      TCCC:    OP      ADD      ACCA,M      ;
0368 0158      MOV      @MEN,B      /* STORE <E(N+1)> */
0368 0158      RPDEC
0368 0158      DPINC
0368 0158      M4      ;
0369 0159      JSA0     TEHN      ;
0370 015A      JZA      TEHZ      ;
0371 015B      JMP      TEHP      ;
0372 015C      TBIAS:  OP      ADD      ACCA,M      ;
0372 015C      MOV      @MEN,RO      /* STORE <E(N+1)> */
0372 015C      DPINC
0372 015C      RPDEC
0372 015C      M4      ;
0373 015D      JSA0     TEHN      ;
0374 015E      JZA      TEHZ      ;
0375 015F      /*      PREDICTOR COEFFICIENTS UPDATE      */
0375 015F      TEHP:   OP      MOV      @NON,B      ;
0375 015F      XOR      ACCB,IDB      ;
0376 0160      OP      SHL1      ACCA      ;
0377 0161      OP      ADD      ACCB,RAM      /* TEST SIGN OF ~E(N-4)/2 */
0377 0161      MOV      @KLM,RO      /* (1-BETA)*B(4)/2 */
0377 0161      RPDEC
0377 0161      M1      ;
0378 0162      CON:    JSB0     TEH4N      ;
0379 0163      TEH4P:  OP      MOV      @TR,A      /* TR=~E(N)/4 */
0379 0163      XOR      ACCA,IDB      ;
0380 0164      OP      ADD      ACCA,RAM      /* TEST SIGN OF ~E(N-3)/2 */
0380 0164      MOV      @B,RO      /* B=CB/2 */
0380 0164      M4      ;
0380 0164      RPDEC
0381 0165      OP      ADD      ACCB,M      ;
0381 0165      MOV      @K,MEN      /* (1-BETA)*B(3)/2 */
0381 0165      RPDEC
0381 0165      M1      ;
0382 0166      OP      MOV      @MEN,B      /* STORE B(4)/2 */
0382 0166      XOR      ACCB,IDB      ;
0382 0166      M4      ;
0383 0167      JSA0     TEH3N      ;
0384 0168      TEH3P:  OP      MOV      @MEN,A      /* SHIFT */
0384 0168      M2      ;
0385 0169      OP      ADD      ACCB,RAM      /* TEST SIGN OF ~E(N-2) */

```


(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0385 0169      MOV      QA,RO      /* B=CB/2 */
0385 0169      M4
0385 0169      RPDEC
0386 016A      OP      ADD      ACCA,M
0386 016A      MOV      QK,MEN      /* (1-BETA)*B(2)/2 */
0386 016A      M3
0386 016A      RPDEC
0387 016B      OP      MOV      QMEN,A      /* STORE B(3)/2 */
0387 016B      XOR      ACCA,IDB
0387 016B      M4
0388 016C      JSBO
0389 016D      TEH2P: OP      MOV      QMEN,B      /* SHIFT */
0389 016D      M2
0390 016E      OP      ADD      ACCA,RAM      /* TEST SIGN OF ~E(N-1) */
0390 016E      MOV      QB,RO      /* B=CB/2 */
0390 016E      M4
0390 016E      RPDEC
0391 016F      OP      ADD      ACCB,M
0391 016F      MOV      QK,MEN      /* (1-BETA)*B(1)/2 */
0391 016F      M1
0391 016F      RPDEC
0392 0170      OP      MOV      QMEN,B      /* STORE B(2)/2 */
0392 0170      M4
0393 0171      JSBO
0394 0172      TEH1P: OP      MOV      QB,TR      /* B=~E(N)/4 */
0395 0173      OP      MOV      QMEN,A      /* SHIFT */
0395 0173      SHL1      ACCB
0395 0173      M1
0396 0174      OP      MOV      QMEN,B      /* STORE ~E(N)/2 */
0396 0174      DPDEC
0396 0174      M3
0397 0175      OP      ADD      ACCB,RAM      /* ~S(N)/2 */
0397 0175      MOV      QA,RO      /* B=CB/2 */
0397 0175      DPINC
0397 0175      M7
0397 0175      RPDEC
0398 0176      OP      ADD      ACCA,M
0398 0176      MOV      QTR,B      /* TR=~S(N)/2 */
0398 0176      RPDEC
0399 0177      OP      MOV      QMEN,A      /* STORE B(1)/2 */
0399 0177      SHR1      ACCB
0399 0177      DPDEC
0400 0178      JMP      TSD
0401 0179      /* BRANCHES FOR COEFFICIENTS UPDATE */
0401 0179      TEH4N: OP      MOV      QTR,A      /* TR=~E(N)/4 */

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0401 0179          XOR      ACCA,IDB
0401 0179          RPDEC
0402 017A          OP      ADD      ACCA,RAM      ; /* TEST SIGN OF ~E(N-3)/2 */
0402 017A          MOV      QB,RO      /* B=CB/2 */
0402 017A          M4
0402 017A          RPDEC
0403 017B          OP      ADD      ACCB,M
0403 017B          MOV      QK,MEN      /* (1-BETA)*B(3)/2 */
0403 017B          M1
0404 017C          OP      MOV      QMEN,B      /* STORE B(4)/2 */
0404 017C          XOR      ACCB,IDB
0404 017C          M4
0405 017D          JNSAO    TEH3P
0406 017E          TEH3N: OP      MOV      QMEN,A      /* SHIFT */
0406 017E          RPDEC
0406 017E          M2
0407 017F          OP      ADD      ACCB,RAM      /* TEST SIGN OF ~E(N-2) */
0407 017F          MOV      QA,RO      /* B=CB/2 */
0407 017F          M4
0407 017F          RPDEC
0408 0180          OP      ADD      ACCA,M
0408 0180          MOV      QK,MEN      /* (1-BETA)*B(2)/2 */
0408 0180          M3
0409 0181          OP      MOV      QMEN,A      /* STORE B(3)/2 */
0409 0181          XOR      ACCA,IDB
0409 0181          M4
0410 0182          JNSBO    TEH2P
0411 0183          TEH2N: OP      MOV      QMEN,B      /* SHIFT */
0411 0183          RPDEC
0411 0183          M2
0412 0184          OP      ADD      ACCA,RAM      /* TEST SIGN OF ~E(N-1) */
0412 0184          MOV      QB,RO      /* B=CB/2 */
0412 0184          M4
0412 0184          RPDEC
0413 0185          OP      ADD      ACCB,M
0413 0185          MOV      QK,MEN      /* (1-BETA)*B(1)/2 */
0413 0185          M1
0414 0186          OP      MOV      QMEN,B      /* STORE B(2)/2 */
0414 0186          M4
0415 0187          JNSAO    TEH1P
0416 0188          TEH1N: OP      MOV      QB,TR      /* B=~E(N)/4 */
0417 0189          OP      MOV      QMEN,A      /* SHIFT */
0417 0189          SHL1    ACCB
0417 0189          M1
0418 018A          OP      MOV      QMEN,B      /* STORE ~E(N)/2 */

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0418 018A          DPDEC
0418 018A          RPDEC
0418 018A          M3          ;
0419 018B          OP          ADD          ACCB,RAM          /* ~S(N)/2 */
0419 018B          MOV          @A,RO          /* B=CB/2 */
0419 018B          DPINC
0419 018B          M7
0419 018B          RPDEC          ;
0420 018C          OP          ADD          ACCA,M          ;
0420 018C          MOV          @TR,B          ; /* TR=~S(N)/2 */
0421 018D          OP          MOV          @MEN,A          /* STORE B(1)/2 */
0421 018D          SHR1          ACCB
0421 018D          DPDEC          ;
0422 018E          JMP          TSD          ;
0423 018F          /* COEFFICIENTS UPDATE FOR ~E(N)=0 */          .
0423 018F          TEHZ: OP          MOV          @NON,B
0423 018F          XOR          ACCB,IDB          ;
0424 0190          OP          SHL1          ACCA          ;
0424 0190          M1          ;
0425 0191          OP          MOV          @TR,A          /* TR=~E(N)/4 */
0425 0191          XOR          ACCA,IDB          ;
0426 0192          OP          MOV          @A,MEN          /* ~E(N-3)/2 */
0426 0192          M1          ;
0427 0193          OP          MOV          @MEN,A          /* SHIFT */
0427 0193          M2          ;
0428 0194          OP          MOV          @A,MEN          /* ~E(N-2) */
0428 0194          M3          ;
0429 0195          OP          MOV          @MEN,A          /* SHIFT */
0429 0195          M2          ;
0430 0196          OP          MOV          @A,MEN          /* ~E(N-1) */
0430 0196          M1          ;
0431 0197          OP          MOV          @B,TR          /* B=~E(N)/4 */
0432 0198          OP          MOV          @MEN,A          /* SHIFT */
0432 0198          SHL1          ACCB
0432 0198          M1          ;
0433 0199          OP          MOV          @MEN,B          /* STORE ~E(N)/2 */
0433 0199          DPDEC
0433 0199          M3          ;
0434 019A          OP          ADD          ACCB,RAM          /* ~S(N)/2 */
0434 019A          M7          ;
0435 019B          OP          MOV          @TR,B          /* TR=~S(N)/2 */
0436 019C          OP          SHR1          ACCB          ;
0437 019D          JMP          TSD          ;
0438 019E          /* COEFFICIENTS UPDATE FOR ~E(N)<0 */
0438 019E          TEHN: OP          MOV          @NON,B

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0438 019E      XOR      ACCB,IDB      ;
0439 019F      OP      SHL1      ACCA      ;
0440 01A0      OP      ADD      ACCB,RAM      ;
0440 01A0      MOV      @KLM,RO      /* TEST SIGN OF ~E(N-4)/2 */
0440 01A0      RPDEC      /* (1-BETA)*B(4)/2 */
0440 01A0      M1      ;
0441 01A1      OP      RPDEC      ;
0442 01A2      JMP      CON      ;
0443 01A3      /*      UPDATE DELTA (SPEECH/DATA DISCREMINATOR) */
0443 01A3      TSD:    LDI      @RP,RPPRE      ;
0444 01A4      OP      SUB      ACCB,RAM      /* ~S(N)/4-~SD(N)/4 */
0444 01A4      MOV      @A,B      ;
0444 01A4      M6      ;
0445 01A5      JSB0      TSD1N      ;
0446 01A6      TSD1P:  OP      SUB      ACCA,RAM      /* ~S(N)/4-~SS(N)/4 */
0446 01A6      MOV      @KLM,RO      /* (1-GAMA)*DELTA */
0446 01A6      RPDEC      ;
0446 01A6      M7      ;
0447 01A7      JSA0      TSD2N      ;
0448 01A8      TSD2P:  OP      SUB      ACCB,IDB      ;
0448 01A8      MOV      @NON,A      ;
0449 01A9      JSB0      TSD3N      ;
0450 01AA      TSD3P:  OP      MOV      @A,RO      /* CDELTA */
0450 01AA      RPDEC      ;
0451 01AB      OP      RPDEC      ;
0452 01AC      JMP      TSD4      ;
0453 01AD      /*      BRANCHES FOR DELTA UPDATE */
0453 01AD      TSD2N:  OP      ADD      ACCB,IDB      ;
0453 01AD      MOV      @NON,A      ;
0454 01AE      JNSB0      TSD3P      ;
0455 01AF      TSD3N:  OP      RPDEC      ;
0456 01B0      OP      MOV      @A,RO      /* CDELTA */
0456 01B0      RPDEC      ;
0457 01B1      JMP      TSD4      ;
0458 01B2      TSD1N:  OP      SUB      ACCA,RAM      /* ~S(N)/4-~SS(N)/4 */
0458 01B2      MOV      @KLM,RO      /* (1-GAMA)*DELTA */
0458 01B2      RPDEC      ;
0458 01B2      M7      ;
0459 01B3      JSA0      TSD5N      ;
0460 01B4      TSD5P:  OP      ADD      ACCB,IDB      ;
0460 01B4      MOV      @NON,A      ;
0461 01B5      JSB0      TSD3P      ;
0462 01B6      OP      RPDEC      ;
0463 01B7      OP      MOV      @A,RO      /* CDELTA */
0463 01B7      RPDEC      ;

```

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0464 01B8      JMP      TSD4
0465 01B9      TSD5N:  OP      SUB      ACCB,IDB
0465 01B9              MOV      @NON,A
0466 01BA              JSB0      TSD3P
0467 01BB              OP      RPDEC
0468 01BC              OP      MOV      @A,RO      /* CDELTA */
0468 01BC              RPDEC
0469 01BD      /*      4TH-ORDER SPEECH-TYPE FIXED PREDICTOR */
0469 01BD      TSD4:  OP      ADD      ACCA,M
0469 01BD              MOV      @KLR,MEM      /* AS(4)/2*S(N-3)/2 */
0469 01BD              M3
0469 01BD              RPDEC
0470 01BE              OP      MOV      @MEN,A      /* STORE DELTA */
0470 01BE              XOR      ACCA,IDB
0470 01BE              M7
0471 01BF              OP      ADD      ACCA,M
0471 01BF              MOV      @KLR,MEM      /* AS(3)/2*S(N-2)/2 */
0471 01BF              M1
0471 01BF              RPDEC
0472 01C0              OP      ADD      ACCA,M
0472 01C0              MOV      @KLR,MEM      /* AS(2)/2*S(N-1)/2 */
0472 01C0              M5
0472 01C0              RPDEC
0473 01C1              OP      ADD      ACCA,M
0473 01C1              MOV      @KLR,TR      /* AS(1)/2*S(N)/2 */
0473 01C1              RPDEC
0473 01C1              RET
0474 01C2      /*      4TH-ORDER DATA-TYPE PREDICTOR */
0474 01C2      CON6:  OP      ADD      ACCA,M
0474 01C2              MOV      @KLR,MEM      /* AD(4)/2*S(N-3)/2 */
0474 01C2              M7
0474 01C2              RPDEC
0475 01C3              OP      MOV      @MEN,A      /* STORE SS(N+1)/4 */
0475 01C3              XOR      ACCA,IDB
0475 01C3              M3
0476 01C4              OP      MOV      @KLR,MEM      /* AD(3)/2*S(N-2)/2 */
0476 01C4              ADD      ACCA,M
0476 01C4              M4
0476 01C4              RPDEC
0477 01C5              OP      MOV      @MEN,K      /* SHIFT */
0477 01C5              ADD      ACCA,M
0477 01C5              M5
0478 01C6              OP      MOV      @KLR,MEM      /* AD(2)/2*S(N-1)/2 */
0478 01C6              M1
0478 01C6              RPDEC

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0479 01C7      OP      MOV      @MEM,K          /* SHIFT */
0479 01C7      ADD      ACCA,M
0479 01C7      M1
0480 01C8      OP      MOV      @KLR,TR        /* AD(1)/2*S(N)/2 */
0480 01C8      RPDEC
0481 01C9      OP      ADD      ACCA,M
0481 01C9      MOV      @MEM,K          /* SHIFT */
0481 01C9      DPINC
0481 01C9      M3
0482 01CA      /*      4TH-ORDER ADAPTIVE NON-RECURSIVE PREDICTOR */
0482 01CA      OP      MOV      @KLN,MEN        /* B(4)/2*E(N-3)/2 */
0482 01CA      M7
0482 01CA      DPDEC
0483 01CB      OP      MOV      @MEM,A          /* STORE ~SD(N+1)/4 */
0483 01CB      M6
0484 01CC      OP      SUB      ACCA,RAM        /* ~SD(N+1)/4-~SS(N+1)/4 */
0484 01CC      MOV      @B,A
0485 01CD      OP      ADD      ACCB,M
0485 01CD      MOV      @KLN,A          /* DELTA*ACCA */
0485 01CD      DPINC
0486 01CE      OP      SUB      ACCB,M
0486 01CE      MOV      @KLN,MEN        /* B(3)/2*E(N-2)/2 */
0486 01CE      M3
0487 01CF      OP      ADD      ACCB,M
0487 01CF      MOV      @KLN,MEN        /* B(2)/2*E(N-1)/2 */
0487 01CF      M1
0488 01D0      OP      ADD      ACCB,M
0488 01D0      MOV      @KLN,MEN        /* B(1)/2*E(N)/2 */
0488 01D0      DPDEC
0488 01D0      M3
0489 01D1      OP      ADD      ACCB,M
0490 01D2      OP      MOV      @NON,A
0490 01D2      XOR      ACCA,IDB
0491 01D3      OP      SHL1      ACCB
0492 01D4      OP      MOV      @MEM,B          /* STORE ~P(N+1)/2 */
0492 01D4      XOR      ACCB,IDB
0492 01D4      DPCLR
0493 01D5      ARCR:  LDI      @A,000FH
0494 01D6      LDI      @SR,SR16I
0495 01D7      OP      AND      ACCA,IDB
0495 01D7      MOV      @NON,DR
0496 01D8      LDI      @B,TABLE
0497 01D9      OP      SUB      ACCB,IDB
0497 01D9      MOV      @NON,A
0498 01DA      OP      MOV      @RP,B

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0498 01DA      XOR      ACCB,IDB      ;
0499 01DB      OP      MOV      QB,RO      ;
0500 01DC      OP      SHL 4      ACCB      ;
0501 01DD      OP      XCHG      ACCB      ;
0502 01DE      LDI      QA,000FH      ;
0503 01DF      OP      AND      ACCA,IDB      ;
0503 01DF      OP      MOV      QNON,B      ;
0503 01DF      OP      DPDEC      ;
0504 01E0      LDI      QB,RPE      ;
0505 01E1      OP      SUB      ACCB,IDB      ;
0505 01E1      OP      MOV      QNON,A      ;
0505 01E1      OP      DPDEC      ; /* ADJUST RP */
0506 01E2      OP      MOV      QRP,B      ;
0506 01E2      OP      DPDEC      ;
0507 01E3      OP      MOV      QKLN,RO      /* CALCULATE <E(N+1)> */
0507 01E3      OP      M4      ;
0508 01E4      LDI      QB,RPOS      ;
0509 01E5      OP      SUB      ACCB,IDB      ;
0509 01E5      OP      MOV      QNON,A      ;
0510 01E6      OP      MOV      QNON,A      ;
0510 01E6      OP      XOR      ACCA,IDB      ;
0511 01E7      CALL     COMN      ;
0512 01E8      LDI      QDP,40H      ;
0513 01E9      OP      MOV      QNEM,DR      /* "R(N) */
0514 01EA      LDI      QDP,6DH      ;
0515 01EB      /*      4TH-ORDER DATA-TYPE PREDICTOR */
0515 01EB      OP      ADD      ACCA,M      ;
0515 01EB      OP      MOV      QKLR,MEN      /* AD(4)/2*"S(N-3)/2 */
0515 01EB      OP      M7      ;
0515 01EB      OP      RPDEC      ;
0516 01EC      OP      MOV      QNEM,A      /* STORE "SS(N+1)/4 */
0516 01EC      OP      XOR      ACCA,IDB      ;
0516 01EC      OP      M3      ;
0517 01ED      OP      MOV      QKLR,MEN      /* AD(3)/2*"S(N-2)/2 */
0517 01ED      OP      ADD      ACCA,M      ;
0517 01ED      OP      M4      ;
0517 01ED      OP      RPDEC      ;
0518 01EE      OP      MOV      QNEM,K      /* SHIFT */
0518 01EE      OP      ADD      ACCA,M      ;
0518 01EE      OP      M5      ;
0519 01EF      OP      MOV      QKLR,MEN      /* AD(2)/2*"S(N-1)/2 */
0519 01EF      OP      M1      ;
0519 01EF      OP      RPDEC      ;
0520 01F0      OP      MOV      QNEM,K      /* SHIFT */
0520 01F0      OP      ADD      ACCA,M

```

** UPD7720 ASSEMBLY LIST ** (7/5/83) MODULE TDHSI4.SRC PAGE 0023

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

```

0520 01F0      M1
0521 01F1      OP      MOV      @KLR,TR      ; /* AD(1)/2*S(N)/2 */
0521 01F1      RPDEC
0522 01F2      OP      ADD      ACCA,M
0522 01F2      MOV      @MEN,K      /* SHIFT */
0522 01F2      DPINC
0522 01F2      M3
0523 01F3 /* 4TH-ORDER ADAPTIVE NON-RECURSIVE PREDICTOR */
0523 01F3      OP      MOV      @KLN,MEN      /* B(4)/2*E(N-3)/2 */
0523 01F3      M7
0523 01F3      DPDEC
0524 01F4      OP      MOV      @MEN,A      ; /* STORE ~SD(N+1)/4 */
0524 01F4      M6
0525 01F5      OP      SUB      ACCA,RAM      ; /* ~SD(N+1)/4-~SS(N+1)/4 */
0525 01F5      MOV      @B,A
0526 01F6      OP      ADD      ACCB,M
0526 01F6      MOV      @KLN,A      /* DELTA*ACCA */
0526 01F6      DPINC
0527 01F7      OP      SUB      ACCB,M
0527 01F7      MOV      @KLN,MEN      /* B(3)/2*E(N-2)/2 */
0527 01F7      M3
0528 01F8      JMP      CONS
0529 01F9      EOF

```

ERROR COUNT = 0000

** UPD7720 ASSEMBLY LIST ** (7/5/83) MODULE TDHSI4.SRC PAGE 0024

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

```

0001 0000      MPROG                      ;
0002 0000 /*      *****
0002 0000      *
0002 0000      *                      DATA ROM FOR TDHS-ARC
0002 0000      *
0002 0000      *****
0002 0000
0002 0000      AUTHOR: MAWLIN YEH
0002 0000      DATE: 9/8/83
0002 0000
0002 0000      PROGRAM: TDHSD4.SRC
0002 0000
0002 0000      VERSION #2: 4/9/83
0002 0000
0002 0000      VERSION #3: 6/29/83
0002 0000                      NEW CODING TABLE
0002 0000                      4TH-ORDER FIXED PREDICTOR
0002 0000
0002 0000      VERSION #4: 7/5/83
0002 0000                      ARCJ ALGORITHM
0002 0000
0002 0000      THIS DATA-ROM FILE INCLUDES THE FOLLOWING PARAMETERS:
0002 0000      1. PITCH PERIOD TO INVERSE OF (PITCH PERIOD-1) CONVERSION
0002 0000      2. COEFFICIENTS FOR THE FIRST-ORDER FILTER IN THE CENTER-
0002 0000          CLIP ROUTINE.
0002 0000      3. COEFFICIENTS FOR LINEAR TO MU-LAW CONVERSION.
0002 0000      4. COEFFICIENTS FOR 4th-ORDER MODEM-TYPE RECURSIVE FIXED
0002 0000          PREDICTOR.
0002 0000      5. COEFFICIENTS FOR 4th-ORDER VOICE-TYPE RECURSIVE FIXED
0002 0000          PREDICTOR.
0002 0000      6. COEFFICIENTS TO UPDATE THE VOICE/MODEM DATA DISCRIMINATOR
0002 0000          PARAMETER.
0002 0000      7. COEFFICIENTS TO UPDATE THE ADAPTIVE PREDICTOR COEFFICIENTS.
0002 0000      8. INVERSE QUANTIZER OUTPUT SCALING FACTORS.
0002 0000      9. OUTPUT QUANTIZER 4-BIT CODES
0002 0000      10. COEFFICIENTS TO UPDATE THE ADAPTIVE QUANTIZER STEP SIZE.
0002 0000      11. QUANTIZER THRESHOLDS.
0002 0000      12. MU-LAW TO LINEAR CONVERSION TABLE.
0002 0000
0002 0000 /*
0002 0000      DATA      0000H
0003 0001      DATA      0000H
0004 0002      ORG      2H
0005 0002      DATA      2224
0006 0003      DATA      2240

```

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

0007	0004	DATA	2256	:
0008	0005	DATA	2280	:
0009	0006	DATA	2296	:
0010	0007	DATA	2320	:
0011	0008	DATA	2344	:
0012	0009	DATA	2360	:
0013	000A	DATA	2384	:
0014	000B	DATA	2408	:
0015	000C	DATA	2424	:
0016	000D	DATA	2448	:
0017	000E	DATA	2472	:
0018	000F	DATA	2496	:
0019	0010	DATA	2520	:
0020	0011	DATA	2544	:
0021	0012	DATA	2568	:
0022	0013	DATA	2592	:
0023	0014	DATA	2624	:
0024	0015	DATA	2648	:
0025	0016	DATA	2672	:
0026	0017	DATA	2704	:
0027	0018	DATA	2728	:
0028	0019	DATA	2760	:
0029	001A	DATA	2792	:
0030	001B	DATA	2816	:
0031	001C	DATA	2848	:
0032	001D	DATA	2880	:
0033	001E	DATA	2912	:
0034	001F	DATA	2944	:
0035	0020	DATA	2976	:
0036	0021	DATA	3016	:
0037	0022	DATA	3048	:
0038	0023	DATA	3088	:
0039	0024	DATA	3120	:
0040	0025	DATA	3160	:
0041	0026	DATA	3200	:
0042	0027	DATA	3240	:
0043	0028	DATA	3280	:
0044	0029	DATA	3320	:
0045	002A	DATA	3360	:
0046	002B	DATA	3408	:
0047	002C	DATA	3448	:
0048	002D	DATA	3496	:
0049	002E	DATA	3544	:
0050	002F	DATA	3592	:
0051	0030	DATA	3640	:

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

0052	0031	DATA	3696	:
0053	0032	DATA	3744	:
0054	0033	DATA	3800	:
0055	0034	DATA	3856	:
0056	0035	DATA	3912	:
0057	0036	DATA	3968	:
0058	0037	DATA	4032	:
0059	0038	DATA	4096	:
0060	0039	DATA	4160	:
0061	003A	DATA	4232	:
0062	003B	DATA	4296	:
0063	003C	DATA	4368	:
0064	003D	DATA	4440	:
0065	003E	DATA	4520	:
0066	003F	DATA	4600	:
0067	0040	DATA	4680	:
0068	0041	DATA	4768	:
0069	0042	DATA	4856	:
0070	0043	DATA	4944	:
0071	0044	DATA	5040	:
0072	0045	DATA	5144	:
0073	0046	DATA	5240	:
0074	0047	DATA	5352	:
0075	0048	DATA	5464	:
0076	0049	DATA	5576	:
0077	004A	DATA	5696	:
0078	004B	DATA	5824	:
0079	004C	DATA	5960	:
0080	004D	DATA	6096	:
0081	004E	DATA	6240	:
0082	004F	DATA	6392	:
0083	0050	DATA	6552	:
0084	0051	DATA	6720	:
0085	0052	DATA	1104	:
0086	0053	DATA	1112	:
0087	0054	DATA	1128	:
0088	0055	DATA	1136	:
0089	0056	DATA	1144	:
0090	0057	DATA	1152	:
0091	0058	DATA	1168	:
0092	0059	DATA	1176	:
0093	005A	DATA	1184	:
0094	005B	DATA	1200	:
0095	005C	DATA	1208	:
0096	005D	DATA	1216	:

(TDHS N 4)

STND LOC. SOURCE STATEMENT

0097 005E	DATA	1232	:
0098 005F	DATA	1240	:
0099 0060	DATA	1256	:
0100 0061	DATA	1264	:
0101 0062	DATA	1280	:
0102 0063	DATA	1288	:
0103 0064	DATA	1304	:
0104 0065	DATA	1320	:
0105 0066	DATA	1328	:
0106 0067	DATA	1344	:
0107 0068	DATA	1360	:
0108 0069	DATA	1376	:
0109 006A	DATA	1384	:
0110 006B	DATA	1400	:
0111 006C	DATA	1416	:
0112 006D	DATA	1432	:
0113 006E	DATA	1448	:
0114 006F	DATA	1464	:
0115 0070	DATA	1480	:
0116 0071	DATA	1496	:
0117 0072	DATA	1512	:
0118 0073	DATA	1536	:
0119 0074	DATA	1552	:
0120 0075	DATA	1568	:
0121 0076	DATA	1592	:
0122 0077	DATA	1608	:
0123 0078	DATA	1632	:
0124 0079	DATA	1648	:
0125 007A	DATA	1672	:
0126 007B	DATA	1688	:
0127 007C	DATA	1712	:
0128 007D	DATA	1736	:
0129 007E	DATA	1760	:
0130 007F	DATA	1784	:
0131 0080	DATA	1808	:
0132 0081	DATA	1832	:
0133 0082	DATA	1856	:
0134 0083	DATA	1888	:
0135 0084	DATA	1912	:
0136 0085	DATA	1944	:
0137 0086	DATA	1968	:
0138 0087	DATA	2000	:
0139 0088	DATA	2032	:
0140 0089	DATA	2064	:
0141 008A	DATA	2096	:

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0142 008B	DATA	2128	:
0143 008C	DATA	2168	:
0144 008D	DATA	2200	:
0145 008E	DATA	2240	:
0146 008F	DATA	2280	:
0147 0090	DATA	2320	:
0148 0091	DATA	2360	:
0149 0092	DATA	2408	:
0150 0093	DATA	2448	:
0151 0094	DATA	2496	:
0152 0095	DATA	2544	:
0153 0096	DATA	2592	:
0154 0097	DATA	2648	:
0155 0098	DATA	2704	:
0156 0099	DATA	2760	:
0157 009A	DATA	2816	:
0158 009B	DATA	2880	:
0159 009C	DATA	2944	:
0160 009D	DATA	3016	:
0161 009E	DATA	3088	:
0162 009F	DATA	3160	:
0163 00A0	DATA	3240	:
0164 00A1	DATA	3320	:
0165 00A2	DATA	32440	; /* ALFA2=0.01 */
0166 00A3	DATA	32344	; /* G=1.3 1-ALFA2*G */
0167 00A4	DATA	010H	; /* CORD 1 */
0168 00A5	DATA	020H	:
0169 00A6	DATA	030H	:
0170 00A7	DATA	040H	:
0171 00A8	DATA	050H	:
0172 00A9	DATA	060H	:
0173 00AA	DATA	070H	; /* CORD 7 */
0174 00AB	DATA	032	:
0175 00AC	DATA	096	; /* CORD THRESHOLD */
0176 00AD	DATA	0224	:
0177 00AE	DATA	0480	:
0178 00AF	DATA	0992	:
0179 00B0	DATA	02016	:
0180 00B1	DATA	4064	:
0181 00B2	DATA	8160	; /* CORD THRESHOLD */
0182 00B3	DATA	12904	; /* AD(1)=0.7878/2 */
0183 00B4	DATA	-16944	; /* AD(2)=-1.034/2 */
0184 00B5	DATA	9640	; /* AD(3)=0.5884/2 */
0185 00B6	DATA	-9408	; /* AD(4)=-.5742/2 */
0186 00B7	DATA	52648	; /* AS(1)=1.99268/2 */

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0187	00B8	DATA	-25976	; /* AS(2)=-1.58545/2 */
0188	00B9	DATA	9864	; /* AS(3)=0.602051/2 */
0189	00BA	DATA	-1672	; /* AS(4)=-.102051/2 */
0190	00BB	DATA	64	; /* GAMA-CDELTA */
0191	00BC	DATA	392	; /* GAMA CDELTA=.01 */
0192	00BD	DATA	32376	; /* 1-GAMA GAMA=0.012 */
0193	00BE	DATA	184	;
0194	00BF	DATA	-184	; /* -CB/2 CB=.011 */
0195	00C0	DATA	184	; /* CB/2 CB=.011 */
0196	00C1	DATA	-184	;
0197	00C2	DATA	184	;
0198	00C3	DATA	-184	;
0199	00C4	DATA	184	;
0200	00C5	DATA	-184	;
0201	00C6	DATA	184	; /* CB/2 */
0202	00C7	DATA	32608	; /* 1-BETA BETA=.005 */
0203	00C8	DATA	20H	; /* <E>MIN */
0204	00C9	DATA	29200	; /* QS(14)=6.48/8*SMIN */
0205	00CA	DATA	19736	; /* QS(13)=4.38/8*SMIN */
0206	00CB	DATA	13112	; /* QS(12)=2.91/8*SMIN */
0207	00CC	DATA	8424	; /* QS(11)=1.87/8*SMIN */
0208	00CD	DATA	5088	; /* QS(10)=1.13/8*SMIN */
0209	00CE	DATA	2792	; /* QS(9)=.62/8*SMIN */
0210	00CF	DATA	1168	; /* QS(8)=.26/8*SMIN */
0211	00D0	DATA	-29200	; /* QS(7)=-6.48/8*SMIN */
0212	00D1	DATA	-19736	; /* QS(6)=-4.38/8*SMIN */
0213	00D2	DATA	-13112	; /* QS(5)=-2.91/8*SMIN */
0214	00D3	DATA	-8424	; /* QS(4)=-1.87/8*SMIN */
0215	00D4	DATA	-5088	; /* QS(3)=-1.13/8*SMIN */
0216	00D5	DATA	-2792	; /* QS(2)=-.62/8*SMIN */
0217	00D6	DATA	-1168	; /* QS(1)=-.26/8*SMIN */
0218	00D7	DATA	0000	; /* QS(0)=0 */
0219	00D8	DATA	0070H	;
0220	00D9	DATA	0060H	; /* Q=14 */
0221	00DA	DATA	0130H	; /* Q=13 */
0222	00DB	DATA	0350H	; /* Q=12 */
0223	00DC	DATA	08B0H	; /* Q=11 */
0224	00DD	DATA	0220H	; /* Q=10 */
0225	00DE	DATA	0580H	; /* Q=9 */
0226	00DF	DATA	0900H	; /* Q=8 */
0227	00E0	DATA	0F00H	; /* Q=7 */
0228	00E1	DATA	0E10H	; /* Q=6 */
0229	00E2	DATA	0C90H	; /* Q=5 */
0230	00E3	DATA	0440H	; /* Q=4 */
0231	00E4	DATA	0DC0H	; /* Q=3 */

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

0232 00E5	DATA	0AA0H	; /* Q=2 */
0233 00E6	DATA	06D0H	; /* Q=1 */
0234 00E7	DATA	08E0H	; /* Q=0 */
0235 00E8	DATA	17592	; /* (1-ALFA+ALFA*[QS])/2 */
0236 00E9	DATA	17136	; /* (1-ALFA+ALFA*[QS])/2 */
0237 00EA	DATA	16816	; /* (1-ALFA+ALFA*[QS])/2 */
0238 00EB	DATA	16592	; /* (1-ALFA+ALFA*[QS])/2 */
0239 00EC	DATA	16432	; /* (1-ALFA+ALFA*[QS])/2 */
0240 00ED	DATA	16320	; /* (1-ALFA+ALFA*[QS])/2 */
0241 00EE	DATA	16240	; /* (1-ALFA+ALFA*[QS])/2 */
0242 00EF	DATA	17592	; /* (1-ALFA+ALFA*[QS])/2 */
0243 00F0	DATA	17136	; /* (1-ALFA+ALFA*[QS])/2 */
0244 00F1	DATA	16816	; /* (1-ALFA+ALFA*[QS])/2 */
0245 00F2	DATA	16592	; /* (1-ALFA+ALFA*[QS])/2 */
0246 00F3	DATA	16432	; /* (1-ALFA+ALFA*[QS])/2 */
0247 00F4	DATA	16320	; /* (1-ALFA+ALFA*[QS])/2 */
0248 00F5	DATA	16240	; /* (1-ALFA+ALFA*[QS])/2 */
0249 00F6	DATA	16184	; /* (1-ALFA+ALFA*[QS])/2 */
0250 00F7	DATA	32264	; /* QTH(7)=TH(7)-TH(6)=1.79/2*SMIN *
0251 00F8	DATA	22528	; /* QTH(6)=TH(6)-TH(5)=1.25/2*SMIN *
0252 00F9	DATA	16040	; /* QTH(5)=TH(5)-TH(4)=.89/2*SMIN */
0253 00FA	DATA	11176	; /* QTH(4)=TH(4)-TH(3)=.62/2*SMIN */
0254 00FB	DATA	7928	; /* QTH(3)=TH(3)-TH(2)=.44/2*SMIN */
0255 00FC	DATA	5584	; /* QTH(2)=TH(2)-TH(1)=.31/2*SMIN */
0256 00FD	DATA	2344	; /* QTH(1)=TH(1)=.13/2*SMIN SMIN=1.1
0257 00FE	ORG	100H	:
0258 0100	DATA	-32128	:
0259 0101	DATA	-31104	:
0260 0102	DATA	-30080	:
0261 0103	DATA	-29056	:
0262 0104	DATA	-28032	:
0263 0105	DATA	-27008	:
0264 0106	DATA	-25984	:
0265 0107	DATA	-24960	:
0266 0108	DATA	-23936	:
0267 0109	DATA	-22912	:
0268 010A	DATA	-21888	:
0269 010B	DATA	-20864	:
0270 010C	DATA	-19840	:
0271 010D	DATA	-18816	:
0272 010E	DATA	-17792	:
0273 010F	DATA	-16768	:
0274 0110	DATA	-16000	:
0275 0111	DATA	-15488	:
0276 0112	DATA	-14976	:

(TDHS N 4)

STNO LOC. SOURCE STATEMENT

0277 0113	DATA	-14464	:
0278 0114	DATA	-13952	:
0279 0115	DATA	-13440	:
0280 0116	DATA	-12928	:
0281 0117	DATA	-12416	:
0282 0118	DATA	-11904	:
0283 0119	DATA	-11392	:
0284 011A	DATA	-10880	:
0285 011B	DATA	-10368	:
0286 011C	DATA	-9856	:
0287 011D	DATA	-9344	:
0288 011E	DATA	-8832	:
0289 011F	DATA	-8320	:
0290 0120	DATA	-7936	:
0291 0121	DATA	-7680	:
0292 0122	DATA	-7424	:
0293 0123	DATA	-7168	:
0294 0124	DATA	-6912	:
0295 0125	DATA	-6656	:
0296 0126	DATA	-6400	:
0297 0127	DATA	-6144	:
0298 0128	DATA	-5888	:
0299 0129	DATA	-5632	:
0300 012A	DATA	-5376	:
0301 012B	DATA	-5120	:
0302 012C	DATA	-4864	:
0303 012D	DATA	-4608	:
0304 012E	DATA	-4352	:
0305 012F	DATA	-4096	:
0306 0130	DATA	-3904	:
0307 0131	DATA	-3776	:
0308 0132	DATA	-3648	:
0309 0133	DATA	-3520	:
0310 0134	DATA	-3392	:
0311 0135	DATA	-3264	:
0312 0136	DATA	-3136	:
0313 0137	DATA	-3008	:
0314 0138	DATA	-2880	:
0315 0139	DATA	-2752	:
0316 013A	DATA	-2624	:
0317 013B	DATA	-2496	:
0318 013C	DATA	-2368	:
0319 013D	DATA	-2240	:
0320 013E	DATA	-2112	:
0321 013F	DATA	-1984	:

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0322	0140	DATA	-1888	:
0323	0141	DATA	-1824	:
0324	0142	DATA	-1760	:
0325	0143	DATA	-1696	:
0326	0144	DATA	-1632	:
0327	0145	DATA	-1568	:
0328	0146	DATA	-1504	:
0329	0147	DATA	-1440	:
0330	0148	DATA	-1376	:
0331	0149	DATA	-1312	:
0332	014A	DATA	-1248	:
0333	014B	DATA	-1184	:
0334	014C	DATA	-1120	:
0335	014D	DATA	-1056	:
0336	014E	DATA	-992	:
0337	014F	DATA	-928	:
0338	0150	DATA	-880	:
0339	0151	DATA	-848	:
0340	0152	DATA	-816	:
0341	0153	DATA	-784	:
0342	0154	DATA	-752	:
0343	0155	DATA	-720	:
0344	0156	DATA	-688	:
0345	0157	DATA	-656	:
0346	0158	DATA	-624	:
0347	0159	DATA	-592	:
0348	015A	DATA	-560	:
0349	015B	DATA	-528	:
0350	015C	DATA	-496	:
0351	015D	DATA	-464	:
0352	015E	DATA	-432	:
0353	015F	DATA	-400	:
0354	0160	DATA	-376	:
0355	0161	DATA	-360	:
0356	0162	DATA	-344	:
0357	0163	DATA	-328	:
0358	0164	DATA	-312	:
0359	0165	DATA	-296	:
0360	0166	DATA	-280	:
0361	0167	DATA	-264	:
0362	0168	DATA	-248	:
0363	0169	DATA	-232	:
0364	016A	DATA	-216	:
0365	016B	DATA	-200	:
0366	016C	DATA	-184	:

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0367 016D	DATA	-168	:
0368 016E	DATA	-152	:
0369 016F	DATA	-136	:
0370 0170	DATA	-120	:
0371 0171	DATA	-112	:
0372 0172	DATA	-104	:
0373 0173	DATA	-96	:
0374 0174	DATA	-88	:
0375 0175	DATA	-80	:
0376 0176	DATA	-72	:
0377 0177	DATA	-64	:
0378 0178	DATA	-56	:
0379 0179	DATA	-48	:
0380 017A	DATA	-40	:
0381 017B	DATA	-32	:
0382 017C	DATA	-24	:
0383 017D	DATA	-16	:
0384 017E	DATA	-8	:
0385 017F	DATA	0	:
0386 0180	DATA	32128	:
0387 0181	DATA	31104	:
0388 0182	DATA	30080	:
0389 0183	DATA	29056	:
0390 0184	DATA	28032	:
0391 0185	DATA	27008	:
0392 0186	DATA	25984	:
0393 0187	DATA	24960	:
0394 0188	DATA	23936	:
0395 0189	DATA	22912	:
0396 018A	DATA	21888	:
0397 018B	DATA	20864	:
0398 018C	DATA	19840	:
0399 018D	DATA	18816	:
0400 018E	DATA	17792	:
0401 018F	DATA	16768	:
0402 0190	DATA	16000	:
0403 0191	DATA	15488	:
0404 0192	DATA	14976	:
0405 0193	DATA	14464	:
0406 0194	DATA	13952	:
0407 0195	DATA	13440	:
0408 0196	DATA	12928	:
0409 0197	DATA	12416	:
0410 0198	DATA	11904	:
0411 0199	DATA	11392	:

** UPD7720 ASSEMBLY LIST ** (7/5/83) MODULE TDHSD4.SRC PAGE 0011

(TDHS # 4)

STND LOC. SOURCE STATEMENT

0412 019A	DATA	10880	:
0413 019B	DATA	10368	:
0414 019C	DATA	9856	:
0415 019D	DATA	9344	:
0416 019E	DATA	8832	:
0417 019F	DATA	8320	:
0418 01A0	DATA	7936	:
0419 01A1	DATA	7680	:
0420 01A2	DATA	7424	:
0421 01A3	DATA	7168	:
0422 01A4	DATA	6912	:
0423 01A5	DATA	6656	:
0424 01A6	DATA	6400	:
0425 01A7	DATA	6144	:
0426 01A8	DATA	5888	:
0427 01A9	DATA	5632	:
0428 01AA	DATA	5376	:
0429 01AB	DATA	5120	:
0430 01AC	DATA	4864	:
0431 01AD	DATA	4608	:
0432 01AE	DATA	4352	:
0433 01AF	DATA	4096	:
0434 01B0	DATA	3904	:
0435 01B1	DATA	3776	:
0436 01B2	DATA	3648	:
0437 01B3	DATA	3520	:
0438 01B4	DATA	3392	:
0439 01B5	DATA	3264	:
0440 01B6	DATA	3136	:
0441 01B7	DATA	3008	:
0442 01B8	DATA	2880	:
0443 01B9	DATA	2752	:
0444 01BA	DATA	2624	:
0445 01BB	DATA	2496	:
0446 01BC	DATA	2368	:
0447 01BD	DATA	2240	:
0448 01BE	DATA	2112	:
0449 01BF	DATA	1984	:
0450 01C0	DATA	1888	:
0451 01C1	DATA	1824	:
0452 01C2	DATA	1760	:
0453 01C3	DATA	1696	:
0454 01C4	DATA	1632	:
0455 01C5	DATA	1568	:
0456 01C6	DATA	1504	:

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0457	01C7	DATA	1440	:
0458	01C8	DATA	1376	:
0459	01C9	DATA	1312	:
0460	01CA	DATA	1248	:
0461	01CB	DATA	1184	:
0462	01CC	DATA	1120	:
0463	01CD	DATA	1056	:
0464	01CE	DATA	992	:
0465	01CF	DATA	928	:
0466	01D0	DATA	880	:
0467	01D1	DATA	848	:
0468	01D2	DATA	816	:
0469	01D3	DATA	784	:
0470	01D4	DATA	752	:
0471	01D5	DATA	720	:
0472	01D6	DATA	688	:
0473	01D7	DATA	656	:
0474	01D8	DATA	624	:
0475	01D9	DATA	592	:
0476	01DA	DATA	560	:
0477	01DB	DATA	528	:
0478	01DC	DATA	496	:
0479	01DD	DATA	464	:
0480	01DE	DATA	432	:
0481	01DF	DATA	400	:
0482	01E0	DATA	376	:
0483	01E1	DATA	360	:
0484	01E2	DATA	344	:
0485	01E3	DATA	328	:
0486	01E4	DATA	312	:
0487	01E5	DATA	296	:
0488	01E6	DATA	280	:
0489	01E7	DATA	264	:
0490	01E8	DATA	248	:
0491	01E9	DATA	232	:
0492	01EA	DATA	216	:
0493	01EB	DATA	200	:
0494	01EC	DATA	184	:
0495	01ED	DATA	168	:
0496	01EE	DATA	152	:
0497	01EF	DATA	136	:
0498	01F0	DATA	120	:
0499	01F1	DATA	112	:
0500	01F2	DATA	104	:
0501	01F3	DATA	96	:

** UPD7720 ASSEMBLY LIST ** (7/5/83) MODULE TDHSD4.SRC PAGE 0013

(TDHS # 4)

STNO LOC. SOURCE STATEMENT

0502	01F4	DATA	88	:
0503	01F5	DATA	80	:
0504	01F6	DATA	72	:
0505	01F7	DATA	64	:
0506	01F8	DATA	56	:
0507	01F9	DATA	48	:
0508	01FA	DATA	40	:
0509	01FB	DATA	32	:
0510	01FC	DATA	24	:
0511	01FD	DATA	16	:
0512	01FE	DATA	8	:
0513	01FF	DATA	0	:
0514	0200	EOF		:

ERROR COUNT = 0000

** UPD7720 ASSEMBLY LIST ** (7/5/83) MODULE TDHSD4.SRC PAGE 0014

APPENDIX C

PRODUCTION COST ESTIMATE

This section presents an estimate of the production cost for the TDHS-ARC system. Two approaches are considered. The first approach utilizes off-the-shelf LSI components for fabrication of the TDHS-ARC system. The second considers the design based on a custom LSI TDHS-ARC transceiver.

C.1 Off-the-Shelf Component Design

The cost estimate given here is for the TDHS-ARC system as delivered as part of this contract (design and package). The current design has considerable potential for optimization for a large production schedule from the point of hardware complexity as well as reliability and testing. These would be incorporated before a production was undertaken.

Cost/unit for < 1000 units - \$2400

Cost/unit for 10,000 units
in batches of 2000 units - \$1920

C.2 Custom LSI TDHS-ARC Transceiver Design

The second approach to manufacturing TDHS-ARC systems is based on a custom LSI TDHS-ARC transceiver chip. The total system would in such a case consist of the custom transceiver, an analog interface and a digital interface. This section briefly considers the contents and size of the transceiver chip.

This analysis of the size of the chip is based on 2 micron NMOS technology and is derived by comparison with currently available signal processing chips, and 256 kilobit RAM chips by appropriate scaling. The transceiver chip would require an approximate NEC 7720 equivalent (40,000 transistors), 30 kilobits of additional RAM (30,000 transistors), and some additional logic for pitch extraction, code-word synchronization, and clock and timing (9000 transistors). The total estimated area is approximately 30 mm². Figure C.1 shows

PROCESSING AND CONTROL: INCLUDES NEC PROCESSING AND CONTROL

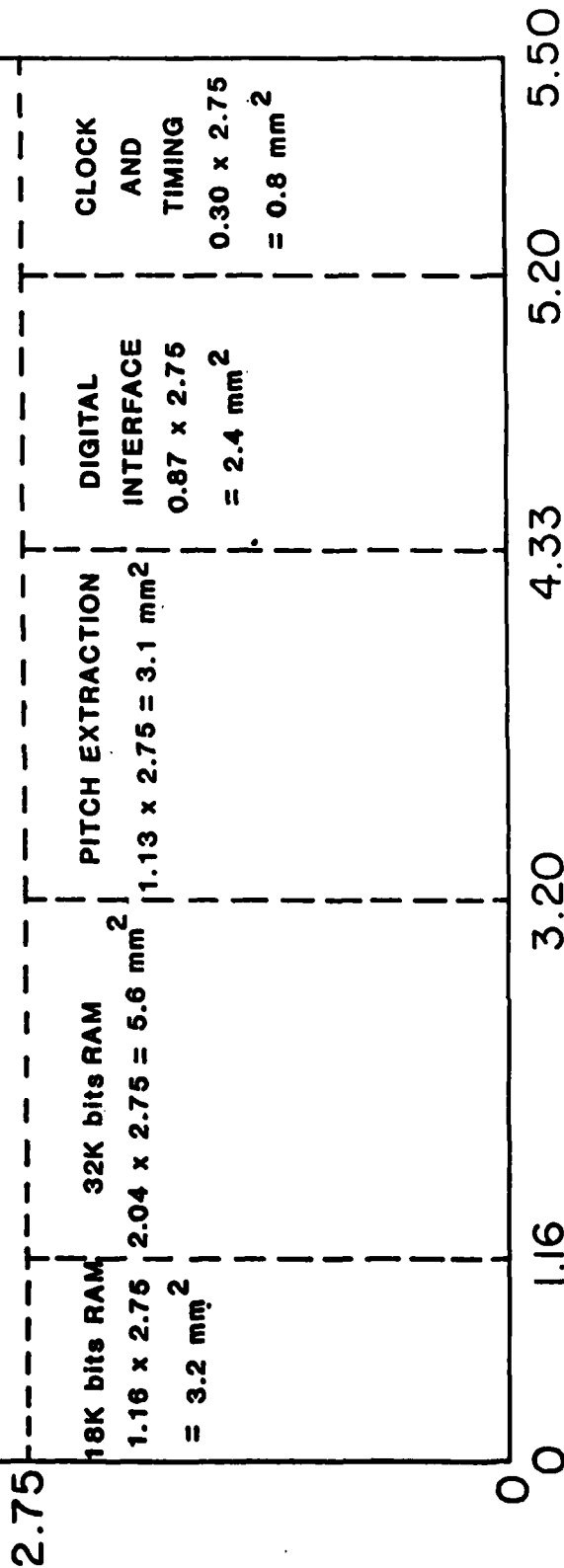
LOGIC (14.4 mm^2) AND COMPUTATION CENTER LOGIC ($.2 \text{ mm}^2$)TOTAL AREA $5.5 \times 2.65 = 14.6 \text{ mm}^2$ 

Fig. C.1: Die Segmentation of a Custom TDHS-ARC Transceiver

a functional breakdown.

NEC chip equivalent	:	17.9	mm ²
30 Kilobits RAM	:	5.3	mm ²
Random logic	:	6.5	mm ²
Total	:	29.7	mm ²

Based on the die area of this chip, and by comparison with the custom LSI convolutional processor (developed for Tellabs, Inc. for use in echo cancellation), the custom TDHS-ARC transceiver would cost about \$30 to \$60 per chip, after development cost and process stabilization. The cost estimate of the TDHS-ARC system built using such a custom transceiver would be as follows:

Cost/unit for 10,000 units	
in batches of 2000 units	- \$800

END

DATE
FILMED

12-8

DTIC