

CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE



National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

TMS 320 BASED
DUAL TONE MULTIFREQUENCY
RECEIVER USING
LINEAR PREDICTION

by

Basavaraj I. Pawate

A thesis

presented to the School of Graduate Studies and Research
of the University of Ottawa
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
Electrical Engineering

Ottawa, Canada, 1985



Basavaraj I. Pawate, Ottawa, Canada, 1985



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

TO
MY FATHER
WHO TAUGHT ME
AND
MY MOTHER
WHO ENCOURAGES ME

ABSTRACT

A novel approach to Dual Tone Multi-frequency (DTMF) Receiver realization is presented. Linear prediction or Autoregressive (AR) time series modelling is used. Various modern spectral estimation techniques are explored because of potential properties that may overcome some of the disadvantages suffered by the existing DTMF receivers. Of the different methods in the class of parametric spectral estimation techniques, the AR model is chosen. Simulation studies regarding the proper choice of the number of data samples, model order and the number of points at which the power spectral density is estimated to match the receiver performance to the Bell system specifications is presented. Simulation studies of the performance of the receiver in a typical telephone environment prove that AR spectral estimation is an attractive alternative to existing methods employed in DTMF Receivers. The model is implemented in real-time on the TMS 320 digital signal processor.

ACKNOWLEDGEMENTS

It is with great pleasure that I acknowledge Dr. Willem Steenaart for having launched me into this challenging field of digital signal processing. I am also highly indebted to him for his inspiring and enthusiastic guidance given to me during the course of this work.

I would also like to thank Dr. Sorin Cohn Sfetou at Bell Northern Research, Ottawa, for his invaluable suggestions and help given during the development of the DTMF Receiver. Thanks are also due to V.I. Pawate for having introduced to me concepts of systems design and structured systems development.

For their cooperation and assistance, while I was completing my research, I would also like to thank all those in the department of Electrical Engineering at the University of Ottawa, Ottawa.

CONTENTS

1.	INTRODUCTION	
1.1.	Aim.....	1
1.2.	DTMF Signalling.....	2
1.3.	Multifrequency Signalling.....	5
1.4.	Approaches Currently Used.....	6
1.5.	Designers Dilemma.....	7
1.6.	Pötential Applications.....	8
1.7.	Outline of thesis	13
2.	WHY LINEAR PREDICTION	
2.1.	DTMF Receiver Task.....	15
2.2.	Processing Strategy.....	15
2.3.	Advantages of the AR Model.....	17
3.	THE AUTOREGRESSIVE MODEL	
3.1.	Modelling and Parameter Identification Approach.....	22
3.2.	The AR Process.....	23
3.3.	Basis of AR Model Quality.....	28
4.	APPLICATION OF LINEAR PREDICTION TO THE DTMF RECEIVER	
4.1.	Simulation Study.....	30
4.2.	Choice of N.....	33
4.3.	Choice of p.....	36
4.4.	Choice of M.....	39
5.	IMPLEMENTATION	
5.1.	DTMF Receiver Task.....	42
5.2.	Implementation Approach	
5.2.1.	Choice of a Microcomputer.....	43
5.2.2.	Programming Technique.....	46
5.3.	R(T), the Autocorrelation function.....	46
5.4.	AR Parameter Estimation.....	50
5.5.	PSD(f) Implementation.....	51
5.6.	Storage and Switching of Input Data.....	56
5.7.	High Level Decision.....	61
5.8.	Threshold Setting Study.....	68
5.9.	DTMF Receiver Test.....	73
6.	CONCLUSION.....	81
	REFERENCES.....	82
	APPENDIX A. PROGRAM LISTING	
	B. TMS320 INSTRUCTION SET.	

LIST OF FIGURES

Figure		page
1.2.1	DTMF Receiver.....	5
1.6.1	Conventional DTMF Receiver Application.....	8
1.6.2	Remote control application with "one-way" data transmission.....	10
1.6.3	Remote control application with "two-way" data transmission.....	11
1.6.4	Possible Industrial Application.....	12
1.6.5	Mobile Radio Telecommunication.....	13
2.3.1	PSD of two sinusoids	18
2.3.2	The AR model places no restriction on the number of points at which the PSD is estimated and also allows for arbitrary spacings between them.....	19
4.1.1	AR Spectral Estimator.....	31
4.2.1	Two Tone Spectrum, N=64.....	32
4.2.2	Two Tone Spectrum, N=128.....	33
4.3.1	Modelling error vs model order.....	39
4.4.1	Points at which PSD is evaluated....	41
5.6.1	TMS320 Memory Map.....	59
5.7.1	Detailed Flow-Chart.....	64
5.9.1	Experimental Setup.....	74
5.9.2	Digit Waveform in Time Domain.....	76

LIST OF TABLES

<u>Table</u>		<u>page</u>
1.2.1	Frequency assignment of the digits in Dual Tone Multifrequency signalling scheme.....	3
1.3.1	Frequency assignment of the digits in Multifrequency signalling scheme.	6
5.2.1	Comparison of Various DSPs.....	45
5.2.2	Benchmark comparison.....	46
5.5.1	Storage of cosine coefficients and computation of PRSUM for one frequency point.....	54
5.5.2	Storage of sine coefficients and computation of IRSUM for one frequency point.....	55
5.6.1	Memory utilization of TMS320.....	58
5.8.1	Maximum and Minimum Values taken by the modelling error at model order 1.....	70
5.8.2	Maximum and Minimum values taken by the modelling error at model order 14.....	71
5.8.3	σ_1 & σ_{14} Values when no tone-pair is input.....	73

CHAPTER 1

INTRODUCTION

1.1 Aim:

The aim of this thesis is to present an attractive alternative to existing methods employed in Dual Tone Multifrequency (DTMF) detection and to implement this technique. The suggested method should be as universal as possible.

With this view in mind, various modern spectral estimation techniques are explored because of potential properties that may overcome some of the disadvantages suffered by the existing DTMF receivers. One such estimation technique, the Autoregressive modelling approach, is chosen and its performance in a typical telephone environment studied. The model is then implemented on the TMS 320 digital signal processor.

Until now, these modern spectral estimation techniques were of mostly an academic interest because of their computation intensity. These complex algorithms require a large number of multiplications and additions. Real-time implementation of these techniques was not yet a reality for all except a few basic and simple algorithms. But with the Texas Instruments high performance 16/32 bit TMS 320

microcomputer, it is now possible to implement numerically intensive algorithms in real-time. This motivated us to implement the AR spectral estimator in real-time.

1.2 DTMF Signalling:

Introduced about 20 years ago, DTMF signalling [1], [2] (also known as Touch-Tone, Tel-Touch, Digitone etc.) has been steadily gaining ground at the expense of dial-pulse signalling. Its advantages are:

- o Convenience of use
- o Efficiency
- o Increased reliability

DTMF signalling is now no longer confined to telephony but is also applicable to other areas as answering machines, radio communications, data transmission and remote control.

In Touch-Tone signalling every digit is represented by a composite audio signal made up of the superposition of two tones, one from the low frequency group

697	770	852	941
-----	-----	-----	-----

and the other from the high frequency group

1209	1336	1477	1633
------	------	------	------

The digits and their corresponding frequencies are shown in table 1.2.1.

Table 1.2.1. Frequency assignment of the digits

	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Since the signalling frequencies fall inside the 0-4 KHz bandwidth used for transmitting speech special features had to be incorporated into the encoding scheme as well as in the detection criteria so as to distinguish real DTMF tones from the ones simulated by speech [3]. This led to the decision in the encoding scheme, to have the low group and the high group, and to represent each digit by two frequencies, one from the low group and the other from the high group. The particular frequencies were then chosen in

such a manner that neither their harmonics nor their intermodulation products fall in one of the tone bands.

For the detection criteria, it was decided that a tone should normally be present for 40 milliseconds since it is highly unlikely that speech simulated digits be present for such a "long" period of time, and it was felt that very few persons would generate speech pulses of less than 40 ms duration [3].

This lead to the following DTMF receiver specifications

Tone burst (min):	40 ms ON, 40 ms OFF
Repetition rate (max):	12 pps
Valid tone accept:	+1.5%
Invalid tone reject:	+3.5%
Input dynamic range:	-26 dBm to +6 dBm
Signal to noise ratio:	15 dB

The fourth column of push-buttons (1633 Hz tone) is usually not present in today's telephone sets. However it is available for future use. A 4x4 matrix of buttons is presently used in some military telephones and is suitable for implementation in telephone sets intended for multiple functions such as financial transaction terminals, credit checking machines and some PBX or CBX systems.

The dual tone signal is heard in the ear-piece and is transmitted over the telephone lines to the private branch exchange or directly to the local exchange where it is decoded into the digit that it represents. The decoder circuitry (Figure 1.2.1) entrusted with this task is called the DTMF receiver.

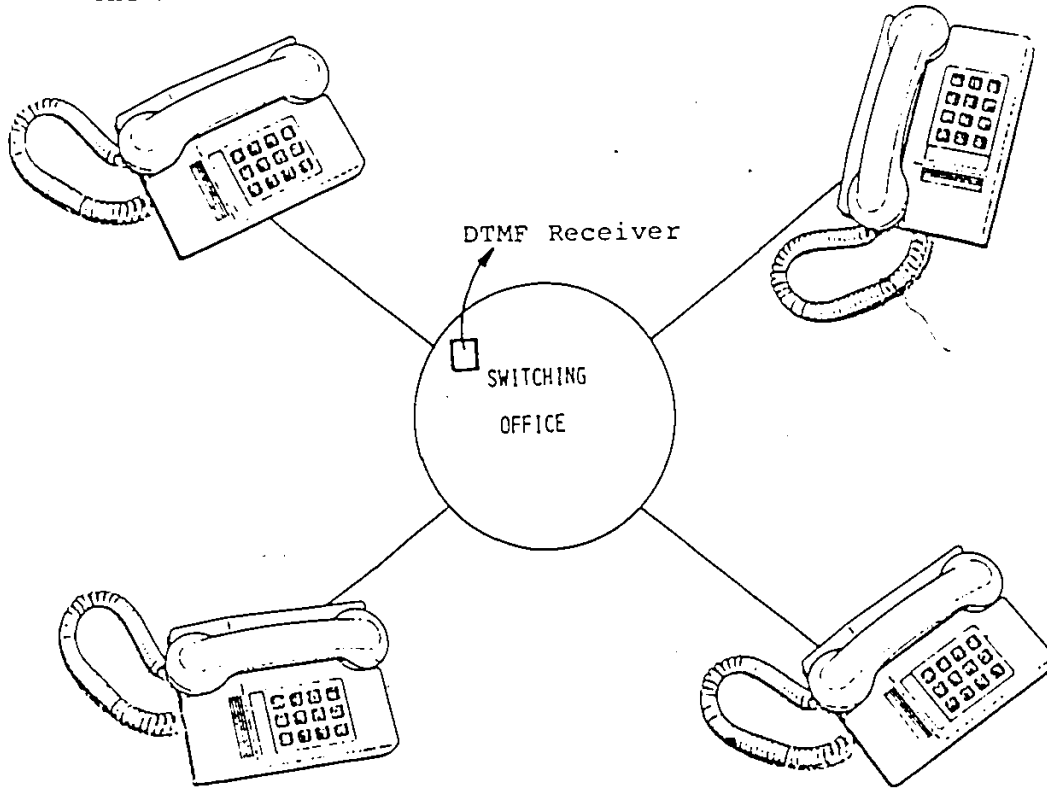


Figure 1.2.1 DTMF Receiver.

1.3 Multifrequency Signalling:

In this signalling scheme, every digit is represented by the simultaneous presence of two frequencies out of a set of six frequencies. The digits and their frequency assignments are shown in Table 1.3.1

Table 1.3.1. Frequency Assignment of the digits.

	1	2	3	4	5	6	7	8	9	0	A	B	P	D	S
700 Hz	*	*		*			*				*				
900 Hz	*		*		*			*				*			
1100 Hz		*	*			*			*				*		
1300 Hz				*	*	*				*				*	
1500 Hz							*	*	*	*					*
1700 Hz											*	*	*	*	*

This signalling scheme is used for inter-office signalling i.e. for signalling between two telephone exchanges.

It is clear that MF signalling scheme uses different frequencies from the DTMF case and the idea of low-group and high group frequencies does not exist.

1.4 Currently Employed Approaches:

A good discussion of these is given by S. Cohn-Sfetou and G. Klowak [4]. A few of them are:

- 1 Classic analog implementation using separate filters for each tone [3],[5]. But since this has been optimized for DTMF, the same design does not work equally well for MF.

- 2 Zero crossing detection [6] using digital period counting relying on a zero-crossing detector. This method is again not suitable for MF since the two tones are not separable as they are in the DTMF scheme.
- 3 Bandpass filtering (Recursive and FIR bandpass filtering) using active or passive filters. These seem to be popular and have been successfully implemented by Cohn Sfetou and others [7],[8].
- 4 Quadrature detection algorithm in which digital modulation is employed [9].
- 5 FFT-based receivers [10] in which the popular FFT approach for spectral estimation is used.

A new approach is sought for to provide the cost-advantage of a single chip implementation and the technical performance of the linear prediction technique.

1.5 Designers Dilemma:

The encoding scheme and the detection criteria evolved as a result of trying to distinguish true DTMF tones from the ones simulated by speech. However, the designer is always faced by a dilemma; if the "accept" criteria are too relaxed, DTMF tone-pairs are quickly recognized as valid and so are speech segments. If the "accept" criteria are too strict, detection

time is inordinately stretched and the receiver is no longer in compliance with timing requirements. Hence the implementation designs that have come up are a trade-off between the "talk-off" performance (immunity to tones simulated by speech) and the tolerance to received-tone imperfection and noise.

1.6 Potential Applications:

DTMF receivers are used mainly in central office and PABX equipment, as shown in Figure 1.6.1.

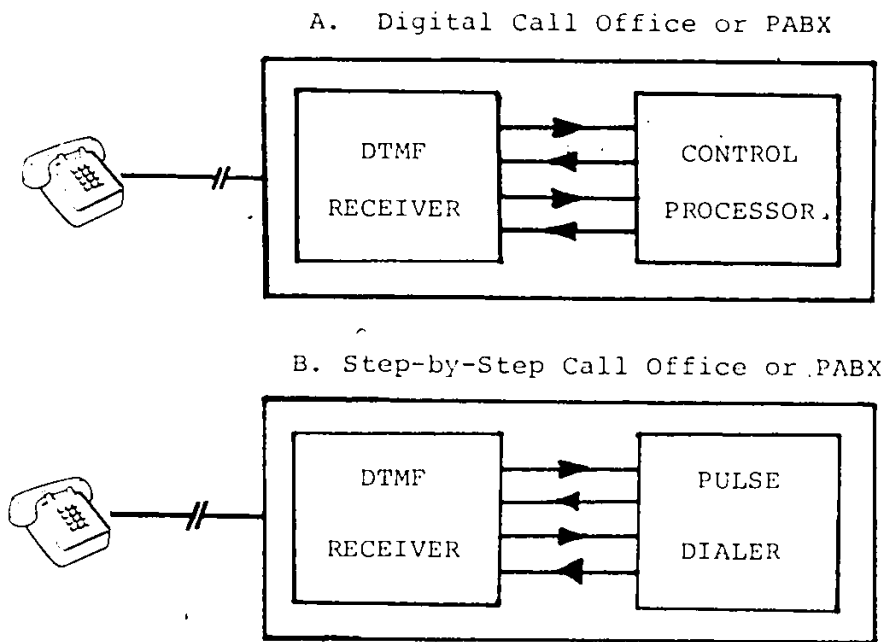


Figure 1.6.1. Conventional DTMF Receiver Applications

Some of the potential applications [6] are:

1. Use of receivers at the subscriber end of the installation, transforming the standard voice channel into an economical means of data transmission. The data may be entered through any standard (DTMF) telephone keypad eliminating the need for modems, acoustic couplers and other specialized interfacing equipment. The disadvantages would be

- o Data-transmission being one way only
- o Limited transmission speed
- o Limited character set as compared to modem solutions.

But there are a host of applications where these shortcomings do not impose a limitation e.g. applications requiring only numeric data plus the "A", "B", "C", "D", "*" and "#" and where data entry speed is limited by speed of the operator (human) input.

2. Use of DTMF in remote control applications namely in the home to control heating, appliances, pay TV etc. and in the industry to control remotely located installations such as data loggers, radio transmitters etc. The increase in cost would be only at the controlled location needing a DTMF receiver and the appropriate interface circuitry. The controlling location would be the Touch-Tone telephone. But the benefits would be:

- o Elimination of equipment at controlling location
- o Lower maintenance costs
- o User friendliness

e.g. a salesman on the road may enter his orders directly into a central computer from any remote location without specialized portable equipment.

Remote control and data-entry are both applications requiring essentially "one-way" data transmission as shown in Figure 1.6.2.

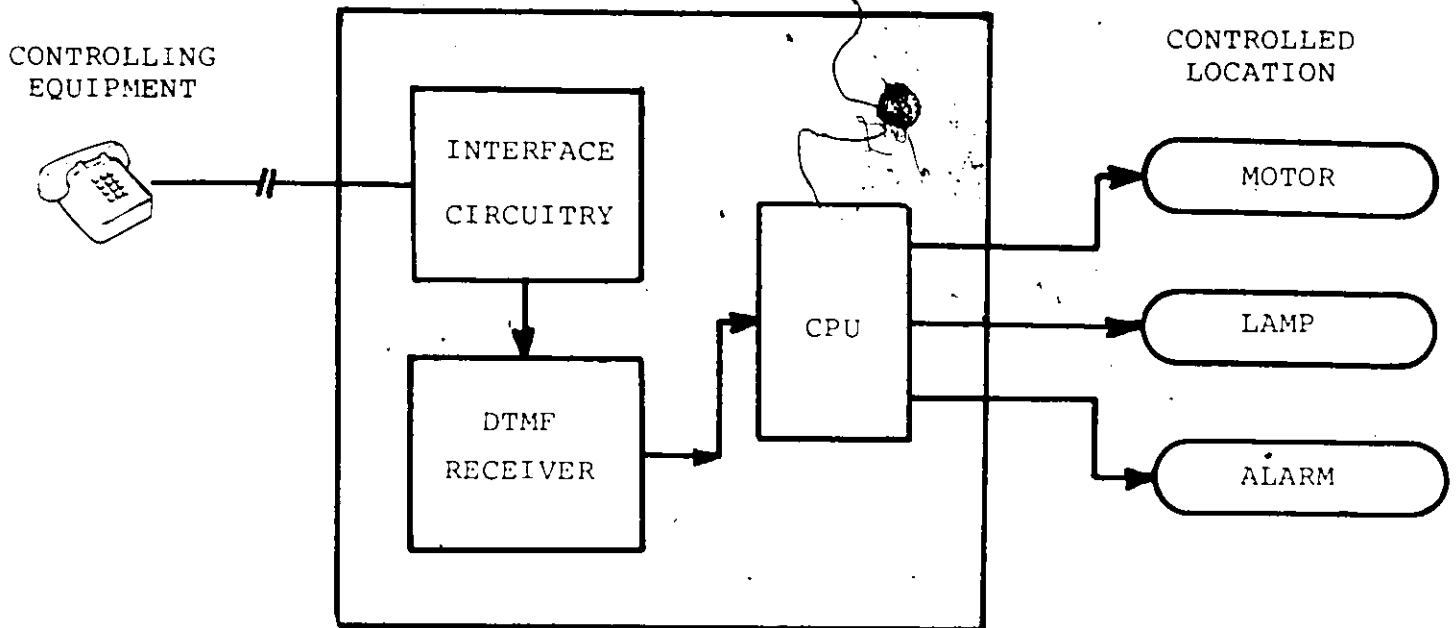


Figure 1.6.2. Remote control application with "one-way" data transmission

Some applications may require "two-way" data transmission where the called location is required to give some response. (Figure 1.6.3) e.g. credit card verification where a central computer must provide a confirmation number in response to a keyed in card number. Here voice synthesis could be used for detailed feed-back responses. This may also bring some of the functions of today's "electronic bank tellers" directly into the home, again without added subscriber equipment.

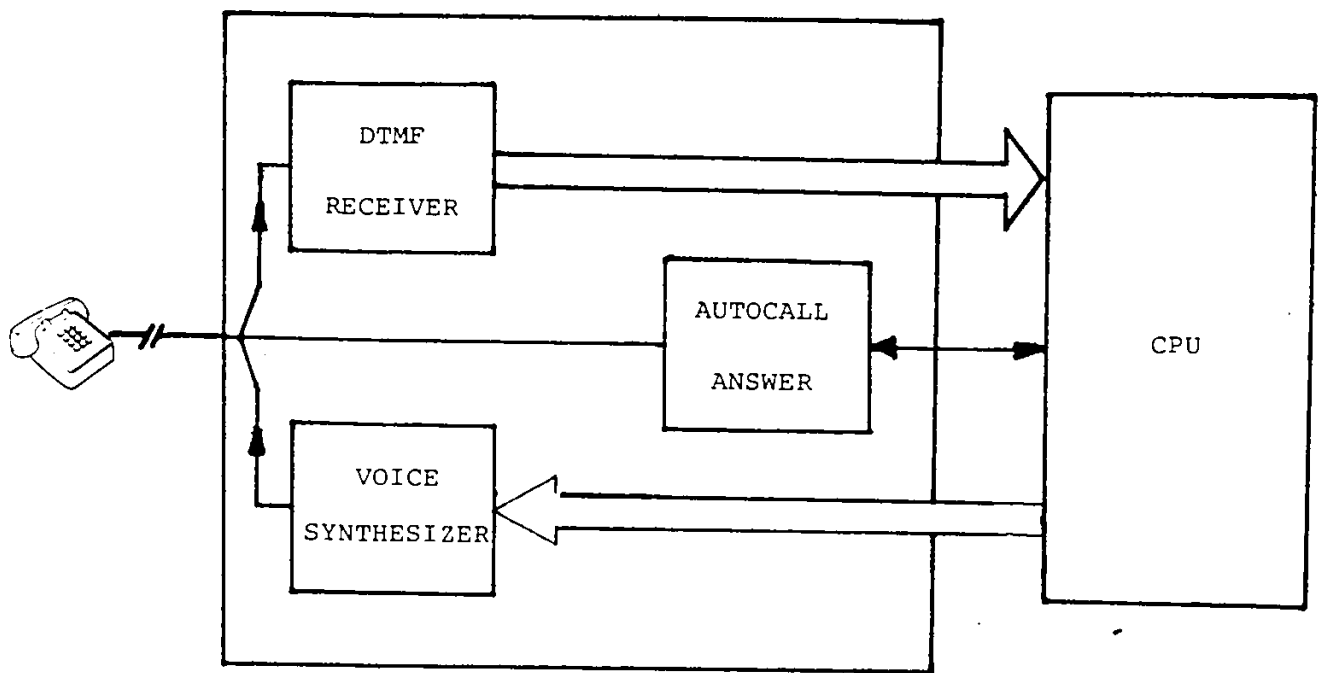


Figure 1.6.3: Remote control application with "two-way" data transmission

All these advantages stem from the fact that the calling location is a simple Touch-Tone telephone and the system can be accessed from any remote location.

- 3 The same approach can be used in industrial control systems. The system does not necessarily have any interface to the public telephone network as in Figure 1.6.4. Here the designer does not have to deal with the "talk-off" problem and hence can adjust the signalling parameters such as tone durations and amplitudes to suit his specific application.

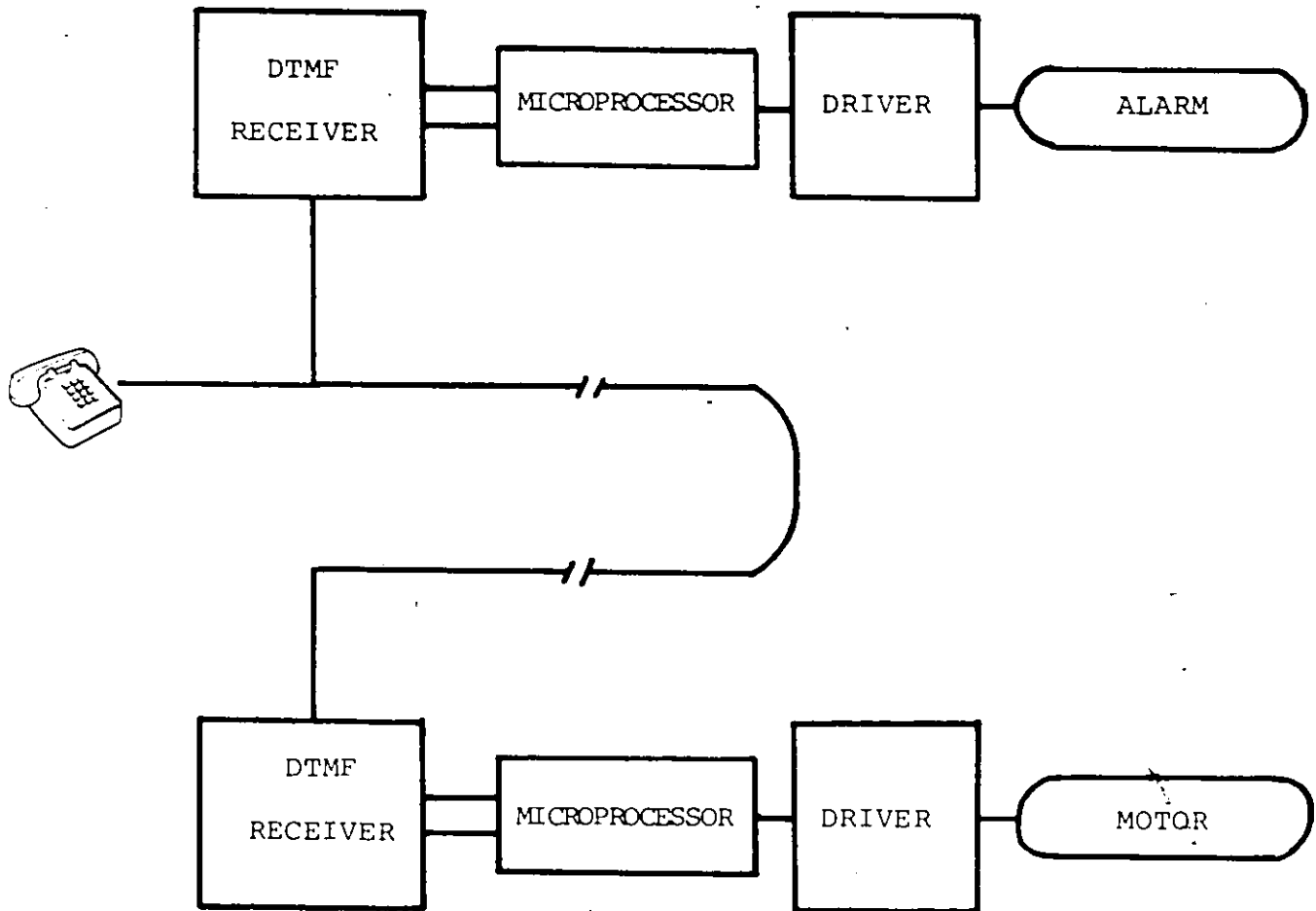


Figure 1.6.4. Possible Industrial Application

4 DTMF signalling is also being extended into mobile-radio telephone applications Figure 1.6.5. The radio link can plug in transparently to the telephone network, simplifying the interface and the same DTMF signalling can be used to transmit channel and other information to and from the mobile installation.

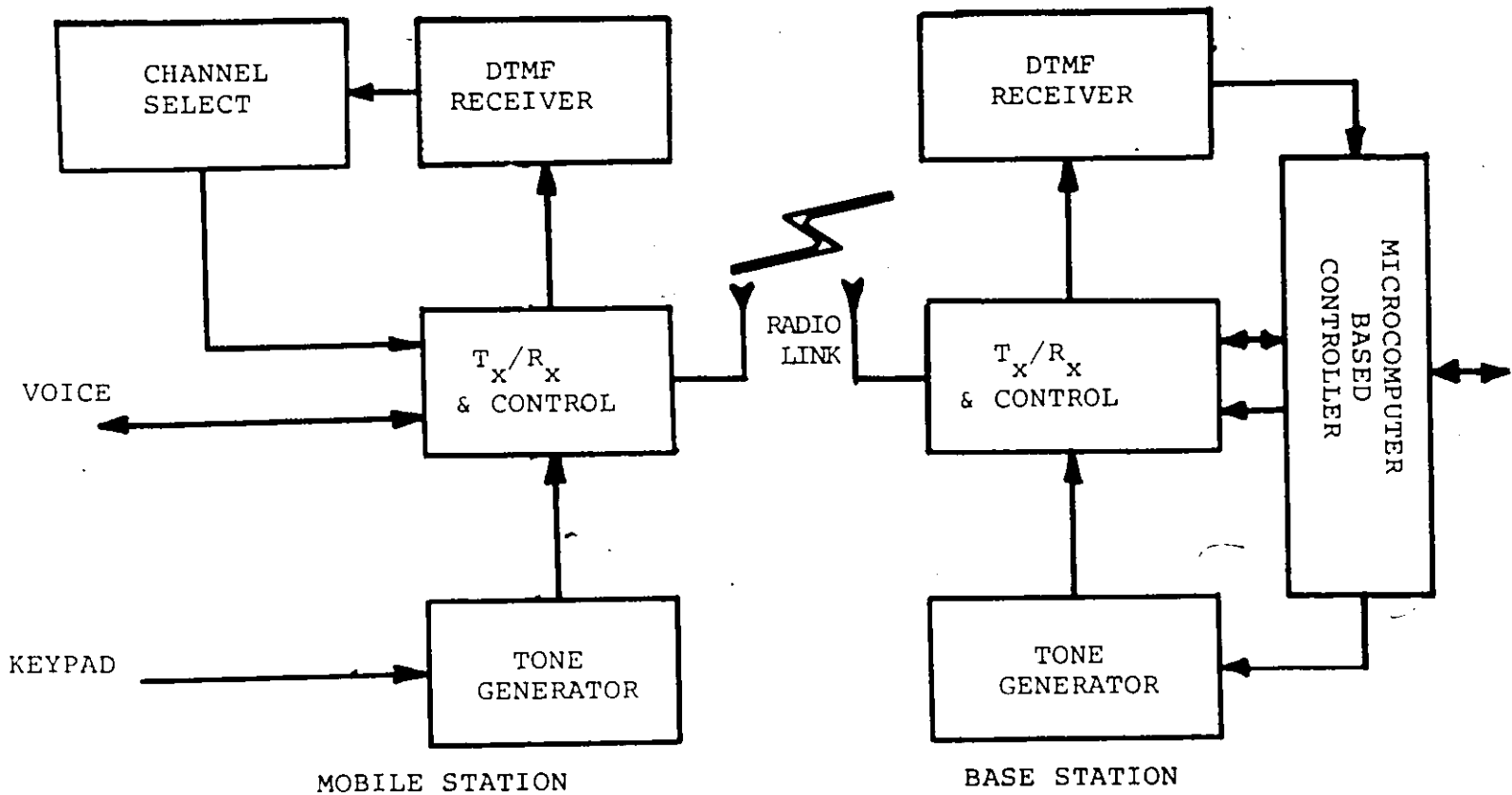


Figure 1.6.5: Mobile Radio Telecommunication

1.7 Outline of thesis

Chapter 2 presents the DTMF Receiver task and the processing strategy employed. Justification for the choice of the AR model as a processing strategy is then given.

Chapter 3 discusses in detail the AR process and basis of AR model quality.

Simulation studies regarding the proper choice of the number of data samples, model order and the number of points at which the power spectral density is estimated to match the receiver performance to the Bell System specification is given in chapter 4. Simulation studies of the performance of the receiver in a typical telephone environment prove that AR spectral estimation is an attractive alternative to existing methods employed in DTMF Receivers.

Chapter 5 presents the implementation of the model on the TMS320 digital signal processor. The approach and a technique for storage and switching of input data to overcome the relatively small data memory is presented. This technique proved to be very efficient in terms of additional hardware requirements.

Conclusions are given in Chapter 6. A novel approach to the DTMF Receiver realisation has thus been presented and implemented on the TMS320 digital signal processor.

CHAPTER 2

WHY LINEAR PREDICTION?

2.1 DTMF Receiver Task:

The main task of a DTMF Receiver is to identify the presence of a valid tone-pair entering the switching office. This means it has to process the incoming signal, examine the energies at different frequencies and then decide whether a digit has been transmitted or not. The task can thus be partitioned into

- o signal processing and
- o high level decision

Processing the signal is necessary to extract the energy information at the desired frequencies and high level decision is required to judge whether a digit has been transmitted or not based on the information obtained by processing the incoming signal.

2.2 Processing Strategy

Processing the incoming signal to obtain the tone energies at different frequencies requires analysis in the frequency domain and any one of the various spectral estimation techniques can be chosen. The periodogram (FFT approach) is

one such technique. However, other approaches to spectral estimation have been proposed. These modern approaches are often referred to in the literature as parametric spectral estimation techniques and include Autoregressive (AR) modelling, Moving Average (MA) modelling and Autoregressive Moving Average (ARMA) modelling [11]. A study of these indicated that some of their potential properties may overcome some of the disadvantages suffered by the present DTMF receivers. However, these advantages are at the cost of increased complexity of computation and pose a challenge for the implementation in real-time. But with the advent of microprocessors with instructions and processing capabilities more tuned to signal processing like the TMS 320 it is shown here that this challenge can be met.

Of the various methods in the class of parametric spectral estimation techniques, the AR model is chosen based on its probability of error performance in Gaussian noise and impulsive noise [12]. The origin of AR models may be found in economic time series forecasting [13], [14] and statistical estimation [15], [16]. It was developed for geophysical data processing, where it was termed the maximum entropy method (MEM) [17], [18]-[29], and has been used for applications in radar [30], [31], sonar [32], [33], imaging [34], radio astronomy [35], biomedicine [36], oceanography [37], ecological systems [38], and direction finding [30], [40]. The AR approach to spectrum analysis is closely related to linear

prediction (LPC) techniques used in speech processing [41], [42]. The AR model is also relatively easier to implement when compared with the other models in the class of parametric spectral estimation techniques. Linear prediction or AR modelling is thus a good strategy for processing the incoming signal.

2.3 Advantages of the AR Model:

The AR PSD estimator fits an AR model to the data. Its main advantages are:

- o Good spectral resolution
- o Significant reduction of sidelobes
- o No restriction on the number of points at which the Power Spectral Density (PSD) is estimated
- o Arbitrary spacings between frequency points allowed.

These advantages are obtained at the cost of increased complexity of computations and pose a challenge for implementation.

Figure 2.3.1. brings out the first two advantages. Comparison is made with an FFT based spectral estimator and the spectral representation of two frequency components shown. The same number of data points are used for both estimators, and rectangular windowing is applied to both.

In Figure 2.3.2 the two other advantages viz. absence of a constraint on the number of frequency

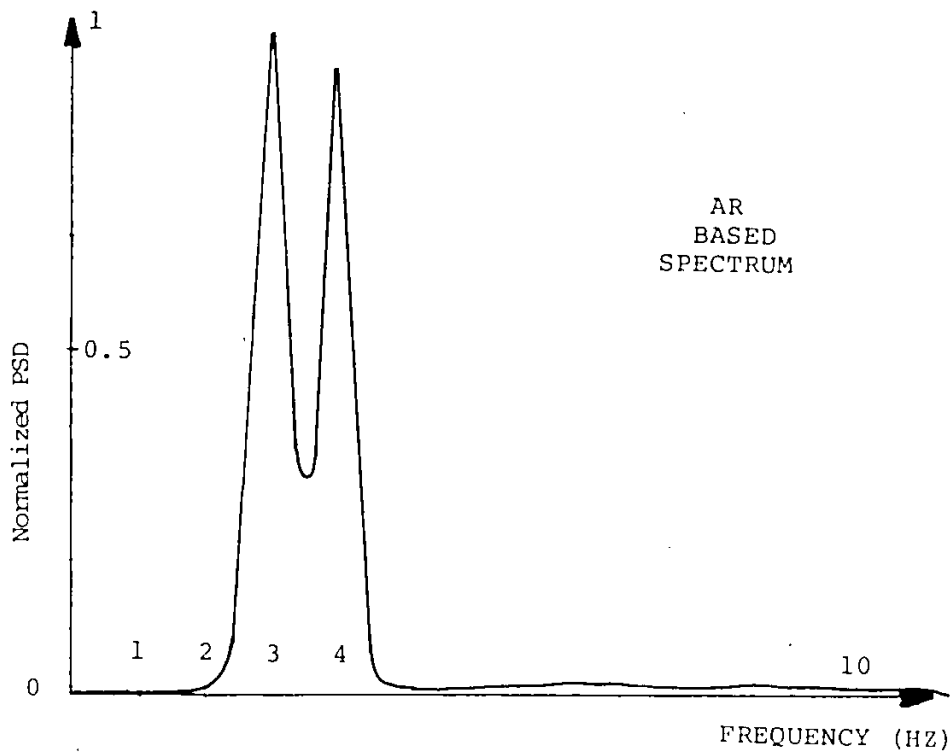
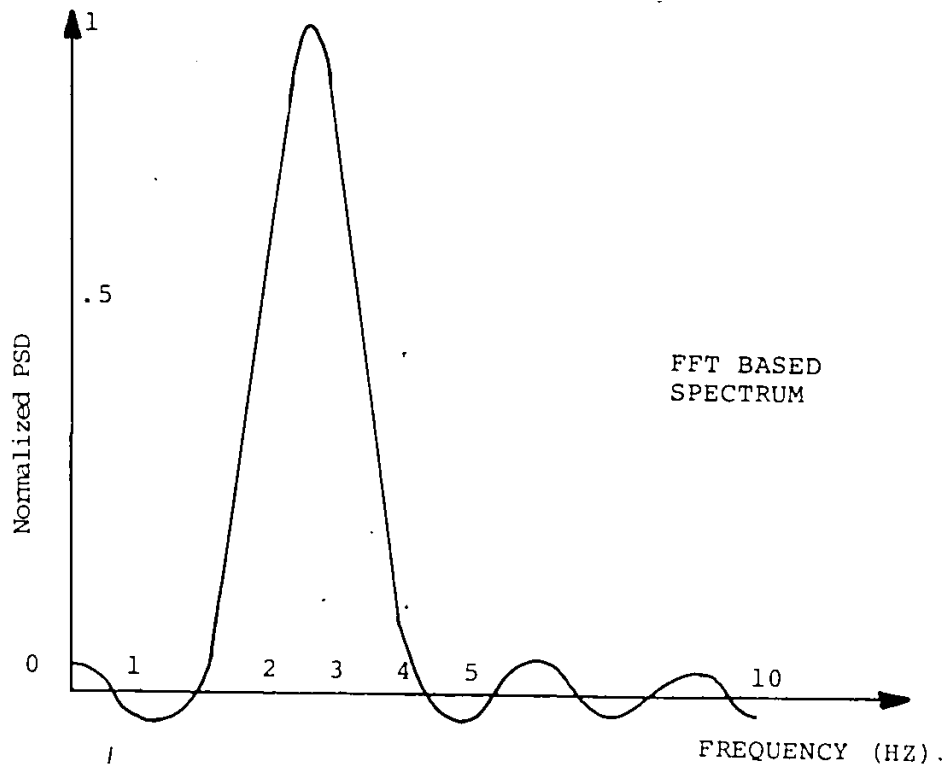


Figure 2.3.1. PSD of two sinusoids (3 Hz & 4 Hz)
 The AR model offers good resolution
 and a significant reduction of side-lobes.

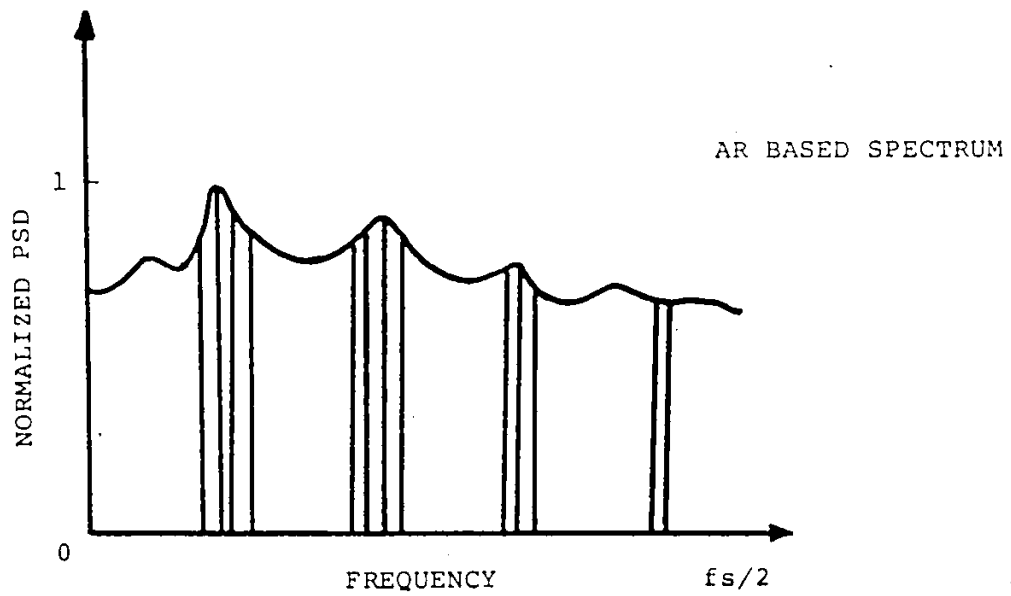
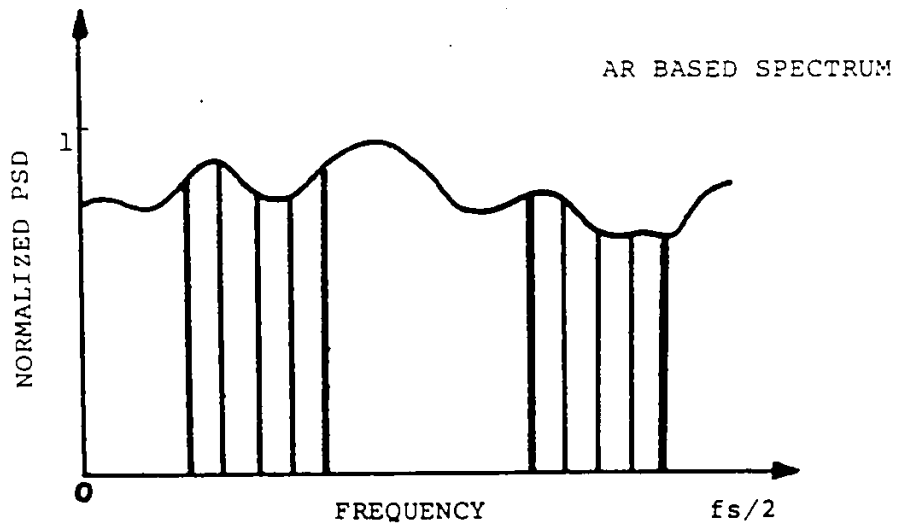


Figure 2.3.2. The AR model places no restriction on the number of points at which the PSD is estimated and also allows for arbitrary spacings between them

points and the arbitrary spacings allowed for between them. This suggests that the model can be used as a general spectral estimator. Depending on the particular application, the points at which the Power Spectral Density is evaluated can be chosen arbitrarily and the AR model can be set at its required model order. Consequently a wide range of applications for this model exist.

It is to be noted that the DTMF receiver requires extracting energy information at only the 8 discrete frequencies belonging to the low group and the high group. These frequencies are not evenly spaced. The AR model thus seems to be an excellent method to obtain the power at the eight desired frequencies.

Many of the conventional approaches employ windowed data. Windowing assumes that the values outside the window are zero or cyclic, an assumption that may or may not be true. This results in a poor spectrum with the following effects:

- o Resolution is dependent on the number of frequency points to be evaluated
- o Weaker signals are drowned by the side lobes of the adjacent stronger signals. This is because implicit windowing of the data manifests itself as leakage in the

spectral domain i.e. energy in the main-lobe of a spectral response leaks into the sidelobes, obscuring and distorting other spectral responses that are present. To avoid the distorting effects of sidelobe "leakage", special tapered windows have to be applied but always at the expense of reduced resolution.

These limitations of conventional approaches to spectral estimation are noticeable, especially in the analysis of short data records. But short data records are encountered often in many systems applications. Most of the processes are brief in duration or have slowly time varying spectra that may be considered "constant" for brief periods of time.

The AR model in such cases is preferable. This is because it effectively extends the autocorrelation function samples outside the interval rather than setting these to zero by windowing. The degree of improvement depends to a large extent on the ability to fit the all-pole model to the measured data with a model order that is large enough.

In conclusion, linear prediction or AR modelling is an attractive strategy for processing the incoming signal.

CHAPTER 3

THE AUTOREGRESSIVE MODEL

3.1. Modelling and Parameter Identification Approach.

Many of the conventional approaches employ windowed data with the implicit assumption that the values outside the window are zero or cyclic, which may or may not be true. Usually a poor spectral estimate is a consequence of windowing.

Generally one has more insight into the process that generated the data samples. This permits a more reasonable assumption to be made about the unobserved data. Thus "a priori" information can be used to select an exact or at least a good approximation model for the process that generated the data samples. It is then usually possible to obtain a better spectral estimate based on this model by determining the parameters of the model from the observations. Thus "parametric spectral estimation" approach involves the following three steps.

1. Select a time-series model,

2. Estimate parameters of this assumed model using either the available data samples or autocorrelation lags (either known or estimated from the data)
3. Obtain the theoretical PSD by substituting the model parameters.

3.2 The AR Process

The AR process [11],[43] was introduced by Yule (1927) and is an all-pole model. It is represented by the relationship

$$X_n = a_1 X_{n-1} + a_2 X_{n-2} + \dots + a_p X_{n-p} + \xi_n \quad (3.2.1)$$

where

$X_n, X_{n-1}, \dots, X_{n-p}$ are the data values at time indicated by their subscript

a_1, a_2, \dots, a_p are the AR parameters, ξ_n is the prediction error and p is the model order.

The model thus identifies X_n with a value predicted from

previous values of the process with a prediction error equal to ξ_n . The AR coefficients a_i are the multiplier coefficients of a p -point prediction filter.

The output of a linear filter, $P_Y(z)$, is related to the power spectrum of the input, $P_X(z)$ as follows: (this is derived in "Digital

Signal Processing" by Oppenheim and Schaefer, pp 391-396)

$$P_Y(z) = \left| H(z) \right|^2 P_X(z) \quad (3.2.2)$$

Since we have assumed an all-pole filter,

$$H(z) = \frac{1}{1 + \sum_{i=1}^p a_i z^{-i}} \quad (3.2.3)$$

If we denote the output of the AR model by $P_{AR}(z)$ and assume the input driving process to be a white noise sequence of zero mean and variance σ^2 , the PSD of the noise is then $\sigma^2 \Delta t$

With $z = e^{j\omega \Delta t}$, we obtain

$$P_{AR}(f) = \frac{\sigma^2 \Delta t}{\left| 1 + \sum_{k=1}^p a_k \exp(-j2\pi f k \Delta t) \right|^2} \quad (3.2.4)$$

where Δt - sampling interval.

$P_{AR}(f)$ is the PSD as suggested by the AR model at frequency f

From (3.2.4) it is clear that to estimate the PSD, one need

only estimate $\{a_1, a_2, \dots, a_p, \sigma^2\}$. To obtain these AR

parameters, their relationship with the autocorrelation function (known or estimated) of X_n is now presented. This

relationship is known as the Yule-Walker equation and is

linear. The derivation follows [12] The autocorrelation function is defined as:

$$R_{xx}(k) = E[X_{n+k} X_n^*] = E[X_n^* (-\sum_{l=1}^p a_l X_{n-l+k} + N_{n+k})]$$

$$= -\sum_{l=1}^p a_l R_{xx}(k-l) + E[N_{n+k} X_n^*] \quad (3.2.5)$$

Since $H(z)$ is assumed to be a stable, causal filter, we have

$$E(N_{n+k} X_n^*) = E[N_{n+k} \sum_{l=0}^{\infty} h_l N_{n-l}] = \sum_{l=0}^{\infty} h_l \sigma^2 \delta_{k+l}$$

$$= \sigma^2 h_{-k} = \begin{cases} 0, & \text{for } k > 0 \\ h_0 \sigma^2, & \text{for } k = 0 \end{cases} \quad (3.2.6)$$

δ_m is the discrete delta function, i.e. $\delta_m = 1$ if $m=0$ or 0 if $m \neq 0$. But $h_0 = \lim_{z \rightarrow \infty} H(z) = 1$ and hence

$$R_{xx}(k) = \begin{cases} -\sum_{l=1}^p a_l R_{xx}(k-l), & \text{for } k > 0 \\ -\sum_{l=1}^p a_l R_{xx}(-l) + \sigma^2, & \text{for } k = 0 \end{cases} \quad (3.2.7)$$

The expression for $R_{xx}(k)$ is called the Yule-Walker

equations. To determine the AR parameters, one needs only to choose p equations for $k > 0$; solve for $\{a_1, a_2, \dots, a_p\}$,

and then find σ^2 from (3.2.7) for $k=0$. The set of equations

which require the fewest lags of the autocorrelation function is the selection $k=1,2,\dots,p$. They can be expressed in the matrix form as

$$\begin{bmatrix}
 R_{xx}(0) & R_{xx}(-1) & \dots & R_{xx}(-p) \\
 R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(-p+1) \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot \\
 R_{xx}(p) & R_{xx}(p-1) & \dots & R_{xx}(0)
 \end{bmatrix}
 \begin{bmatrix}
 1 \\
 a_1 \\
 \cdot \\
 \cdot \\
 \cdot \\
 a_p
 \end{bmatrix}
 =
 \begin{bmatrix}
 \sigma_a^2 \\
 0 \\
 \cdot \\
 \cdot \\
 \cdot \\
 0
 \end{bmatrix}$$

(3.2.8)

Thus to determine the AR parameters and σ_a^2 , one must solve (3.2.8) with the $p+1$ estimated autocorrelation lags R_{xx}

$R_{xx}(0), \dots, R_{xx}(p+1)$ and use $R_{xx}(-m) = R_{xx}^*(m)$ where $*$ denotes

complex conjugate.

One way to solve the computationally burdensome matrix is to employ the Levinson-Durbin algorithm [43], [44], [45], [46]. This requires only in the order of p operations. It proceeds recursively to compute the parameter sets

$$\{a_{11}, \sigma_1^2\}, \{a_{21}, a_{22}, \sigma_2^2\}, \dots, \{a_{p1}, a_{p2}, \dots, a_{pp}, \sigma_p^2\}.$$

Note that an additional subscript has been added to the AR coefficients to denote the order. The recursive algorithm is initialized by

$$a_{11} = -R_{xx}(1)/R_{xx}(0) \tag{3.2.9}$$

$$\sigma_1^2 = (1 - |a_{11}|^2) R_{xx}(0)$$

with the recursion for $k=2,3,\dots,p$ given by

$$a_{kk} = -\{R_{xx}(k) + \sum_{l=1}^{k-1} a_{k-1,l} R_{xx}(k-1)\} / \sigma_{k-1}^2$$

$$a_{ki} = a_{k-1,i} + a_{kk} a_{k-1,k-i} \tag{3.2.10}$$

$$\sigma_k^2 = (1 - |a_{kk}|^2) \sigma_{k-1}^2$$

The final set at order p is the desired solution.

From equations (3.2.10) it is clear that the model order must be as small as possible to reduce the number of computations. At the same time the chosen model order must

$$\{a_{11}, \sigma_1^2\}, \{a_{21}, a_{22}, \sigma_2^2\} \dots \{a_{p1}, a_{p2}, \dots, a_{pp}, \sigma_p^2\}.$$

Note that an additional subscript has been added to the AR coefficients to denote the order. The recursive algorithm is initialized by

$$a_{11} = R_{xx}(1)/R_{xx}(0) \tag{3.2.9}$$

$$\sigma_1^2 = (1 - |a_{11}|^2) R_{xx}(0)$$

with the recursion for $k=2, 3, \dots, p$ given by

$$a_{kk} = \{R_{xx}(k) + \sum_{l=1}^{k-1} a_{k-1,l} R_{xx}(k-1)\} / \sigma_{k-1}^2$$

$$a_{k1} = a_{k-1,1} + a_{kk} a_{k-1,k-1} \tag{3.2.10}$$

$$\sigma_k^2 = (1 - |a_{kk}|^2) \sigma_{k-1}^2$$

The final set at order p is the desired solution.

From equations (3.2.10) it is clear that the model order must be as small as possible to reduce the number of computations. At the same time the chosen model order must

give the desired performance and also enable real-time implementation. In effect, too low a model order will result in a highly smoothed spectral estimate. Too high an order will introduce spurious details into the spectrum and increase the computational complexity as well. Thus an appropriate choice of the model order is warranted. Intuitively, one can use equation (3.2.10) to generate successively higher order models until the modelling error is reduced to a desired value. The point at which σ_p^2 does not change appreciably with increasing 'p' gives a practical choice of the model order. It can be shown that $a_{kk} < 1$, so

that $\sigma_{k+1}^2 < \sigma_k^2$ [47] [48]. This means that σ_k^2 first reaches its minimum at the correct model order.

3.5 Basis of AR Model Quality

An alternative expression for equation (3.2.4) is [49]

$$P_{AR}(f) = \frac{\sigma_p^2 \Delta t}{\left| 1 + \sum_{k=1}^p a_{pk} \exp(-j2\pi f k \Delta t) \right|^2} - \Delta t \sum_{n=-\infty}^{\infty} r_{xx}(n) \exp(-j2\pi f n \Delta t) \quad (3.5.1)$$

where

$$r_{xx}(n) = \begin{cases} R_{xx}(n), & \text{for } |n| \leq p \\ -\sum_{k=1}^p a_k r_{xx}(n-k), & \text{for } |n| > p \end{cases}$$

From (3.5.1) it is clear that the AR model effectively extends the autocorrelation function samples outside the interval rather than setting these to zero by windowing. Thus the AR spectra do not exhibit sidelobes. Also it is the implied extrapolation given by (3.5.1) that is responsible for the high resolution property of the AR spectral estimator.

CHAPTER 4

APPLICATION OF LINEAR PREDICTION TO THE DTMF RECEIVER

4.1 Simulation Study

So far we have seen that the AR Spectral Estimator has shown enough potential properties to make it seem attractive. However many issues have to be addressed to determine its suitability for real-time implementation. The number of data samples, the model order and the number of points at which the PSD has to be evaluated all have a bearing on the real-time implementation considerations. Various trade-offs may have to be made among these parameters. To this end, extensive simulation studies of the model in a typical telephone environment have been carried out [1]. Simulation studies were made regarding the proper choice of parameters N , p and M , where

N is the number of data-points

p is the model order and

M is the number of points at which the PSD is estimated.

N , p and M have to be chosen to allow for real-time implementation and to meet the specifications for tone detection. The steps involved in the calculation of the PSD are shown in Figure 4.1.1.

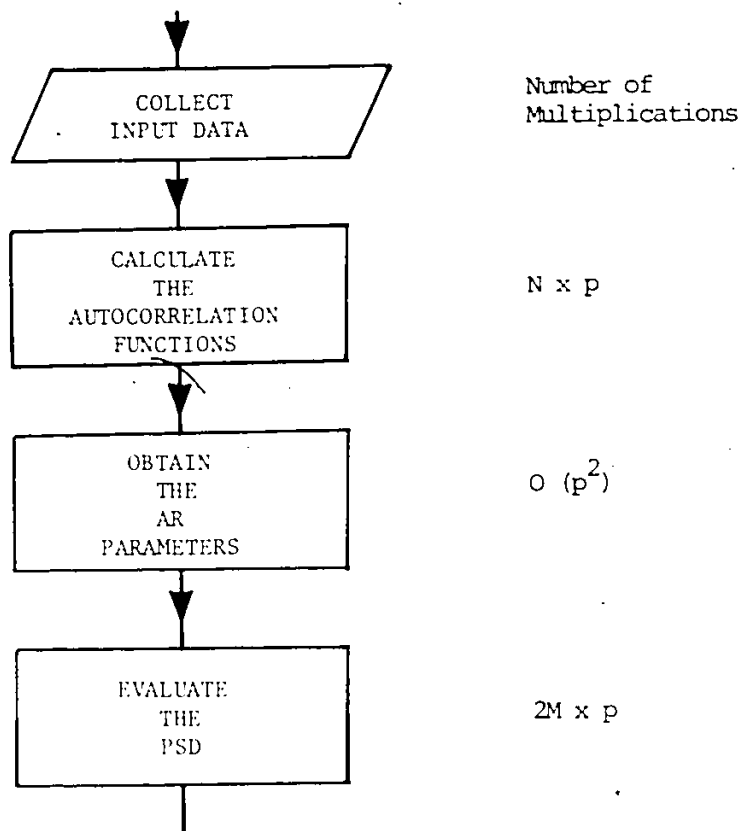


Figure 4.1.1 AR spectral estimator

It is to be noted that the model can be made to evaluate the PSD at only the frequencies of interest. This is particularly advantageous since the model places no constraints on the number or spacing of the spectral points. It also offers a high resolution and significant reduction of the side-lobes.

This is in contrast to say an FFT based estimator where the FFT grid would miss the frequency points of interest if the FFT used was of too low an order. The FFT spectral analysis would also provide information on frequencies which are outside the range of the TTR (Touch-Tone Receiver) i.e. it would provide spectral information before 697 Hz and after 1633 Hz as well.

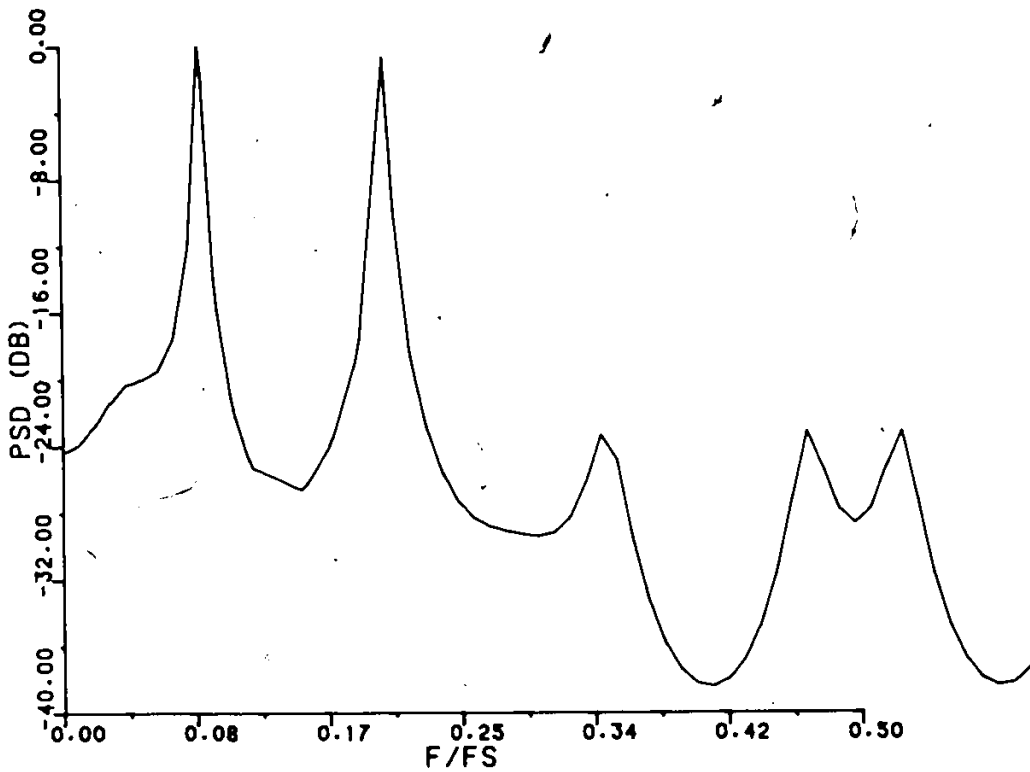


Figure 4.2.1. Two tone spectrum, N-64

4.2 Choice of N

The first goal of the simulation studies was to determine an appropriate value for N, the number of data samples. Figure 4.1.1 shows that N contributes significantly to the number of multiplications involved. In fact it is the largest contributor since N^2 multiplications are required for calculating the autocorrelation lags. Two tones embedded in white Gaussian noise were considered. Rectangular windowing was assumed and a signal to noise ratio of 15 dB taken. The spectrum obtained was then examined for different values of N from 32 upward. Attributes examined were the resolution,

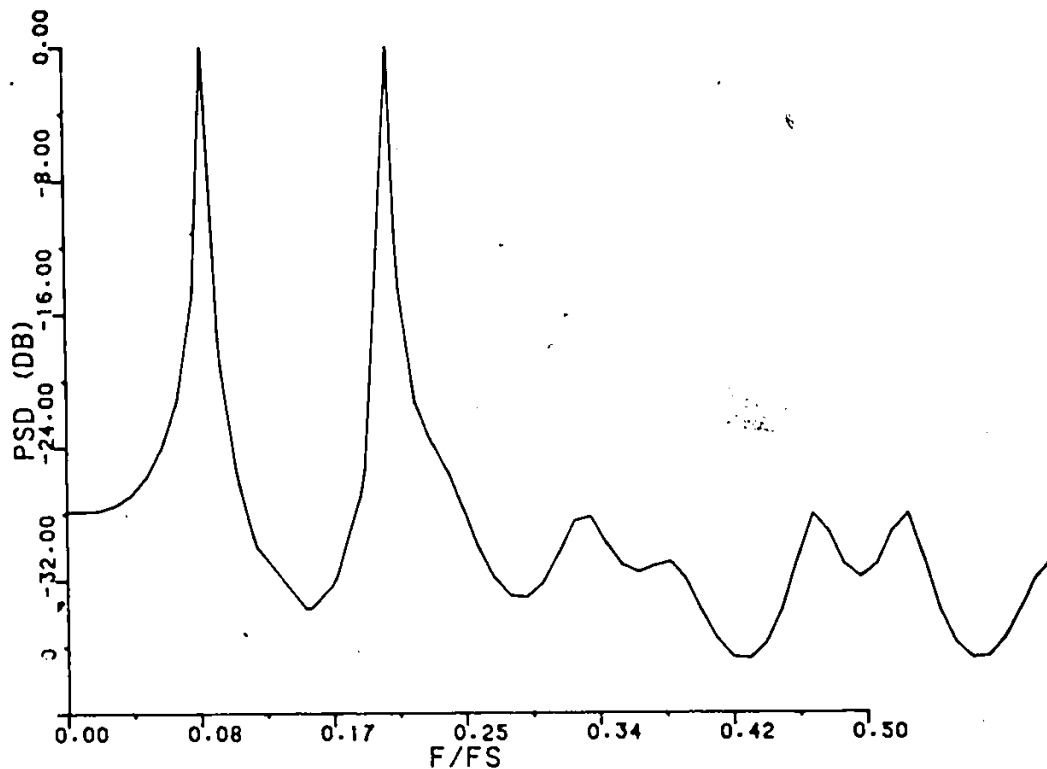


Figure 4.2.2. Two tone spectrum N-128

width of the peaks and the noise-floor. Sample spectra for N=64 and N=128 are shown below in Figures 4.2.1 and 4.2.2. The performance improved as N increased. This is expected since larger values of N contain more information on the signal. The autocorrelation estimates improve as N is increased, as shown below, and as a result the performance also improves.

For the solution of Yule-Walker equations, the autocorrelation matrix is to be known or given. But, in practice, the only available information is the finite data sequence. Hence the autocorrelation matrix (3.2.8) has to be replaced by a suitable estimate of the lags. Since the statistical autocorrelation function is not known, to make matters simple, it is assumed that the random process is ergodic. This property permits the substitution of time averages for ensemble averages. For an ergodic process, then, the statistical autocorrelation function may be equated to

$$R_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t+\tau)x^*(t) dt \quad (4.2.1)$$

An autocorrelation estimate is

$$\hat{R}_{xx}(m) = \frac{1}{N-m} \sum_{n=0}^{N-m-1} x_{n+m} x_n^* \quad (4.2.2)$$

for $m=0, \dots, M$ where $M < N-1$

where $\hat{\cdot}$ denotes an estimate

The negative estimates are obtained from the positive lag estimates as follows:

$$\hat{R}_{xx}(-m) = \hat{R}_{xx}^*(m) \quad (4.2.3)$$

in accordance with the conjugate symmetric property of the autocorrelation function of a stationary process. Instead of (4.2.2), which gives an unbiased estimator, Jenkins-Watts [50] and Parzen [51],[15] show that the autocorrelation estimate

$$\hat{R}_{xx}(m) = 1/N \sum_{n=0}^{N-m-1} \mathbf{X}_{n+m} \mathbf{X}_n^* \quad (4.2.4)$$

$m = 0, 1, \dots, M$

is better since it tends to have less mean square error than (4.2.2) for many finite data sets. (4.2.4) is known as a biased estimator.

From (4.2.4) it is clear that the autocorrelation estimates improve as N increases. The frequency peaks get narrower and the noise floor is pushed further down. This is clear from Figures 4.2.1 and 4.2.2. Consequently larger values of N are preferred. However, real-time and hardware considerations limit N . The hardware limitations are that the TMS320 has a 144 word data RAM and the 32 bit wide

accumulator allows two 12 bit numbers to be multiplied and accumulated without overflow for $N=128$. Beyond $N=128$ overflows may occur. Another hardware limitation is that the TMS320 has no hardware divider and a software subroutine has to be written for division. As a result division becomes a time consuming task and should be avoided as far as possible. It is preferred that N be a power of 2 since division then would mean shifting right appropriately. Thus as a trade-off between performance and hardware/real-time considerations, $N=128$ is chosen.

$N=64$ gives satisfactory results as seen in Figure 4.2.1. But for the reasons stated above $N=128$ is chosen.

4.3 Choice of the model order:

The choice of p , the model order, is critical because the system performance depends considerably on it. p affects

- o the spectrum evaluated
- o the number of calculations required
- o the evaluation time, which is defined as the time required to calculate the autocorrelation values, the AR parameters and the final PSD estimation at the appropriate frequencies.

Too low a model order results in a highly smoothed spectrum. When the model order chosen is too large relative to the

number of data points, the AR spectral estimate exhibits spurious peaks [52], [53]. Ideally if the autocorrelation lags are estimated without errors, then the estimated AR parameters for an AR(p) model would be:

$$a_{pi} = \begin{cases} a_{pi} & \text{for } i=1,2,\dots,p \\ 0 & \text{for } i=p+1,\dots,n \end{cases} \quad (4.3.1)$$

where a_{pi} are the AR(p) parameters. But since estimation errors in equation (4.2.4) are inevitable,

$$a_{pi} \neq 0 \text{ for } i > p.$$

As a consequence, (n-p) "extra" poles are introduced. When these "extra" poles occur near the unit circle, they are brought out as spurious spectral peaks. Hence it is a good rule of thumb to restrict p to be no greater than N/2.

The number of calculations increases with the model order. The number of multiplications, shown in Figure 4.1.1, is of the order of p^2 . The autocorrelation matrix (3.3.8) also gets larger as p increases. Thus p should be as small as possible.

The evaluation time increases also with the number of the calculations resulting from an increase in model order.

In contrast, it is shown in (3.2.1) which is reproduced here, that a large value of 'p' is necessary if the prediction error is to be minimized.

$$X_t = a_1 x_{t-1} + a_2 X_{t-2} + \dots + a_p X_{t-p} + \text{prediction error}$$

All these aspects thus bring out the fact that the choice of p should be adapted to each particular application.

As discussed in Chapter 3 the point at which σ_p^2 does not decrease with increasing p is a good way to choose the model order. Simulation shows the relationship between σ_p^2 and p. Again N=128 is chosen with rectangular windowing and a signal to noise ratio of 15 dB assumed. The results are shown in Figure 4.3.1

σ_p^2 does not change appreciably beyond p=10. This shows that the increase in performance diminishes rapidly as p increases beyond a threshold value (10 in this case). Hence p can be chosen in the range of 10-15; p=14 has been used here.

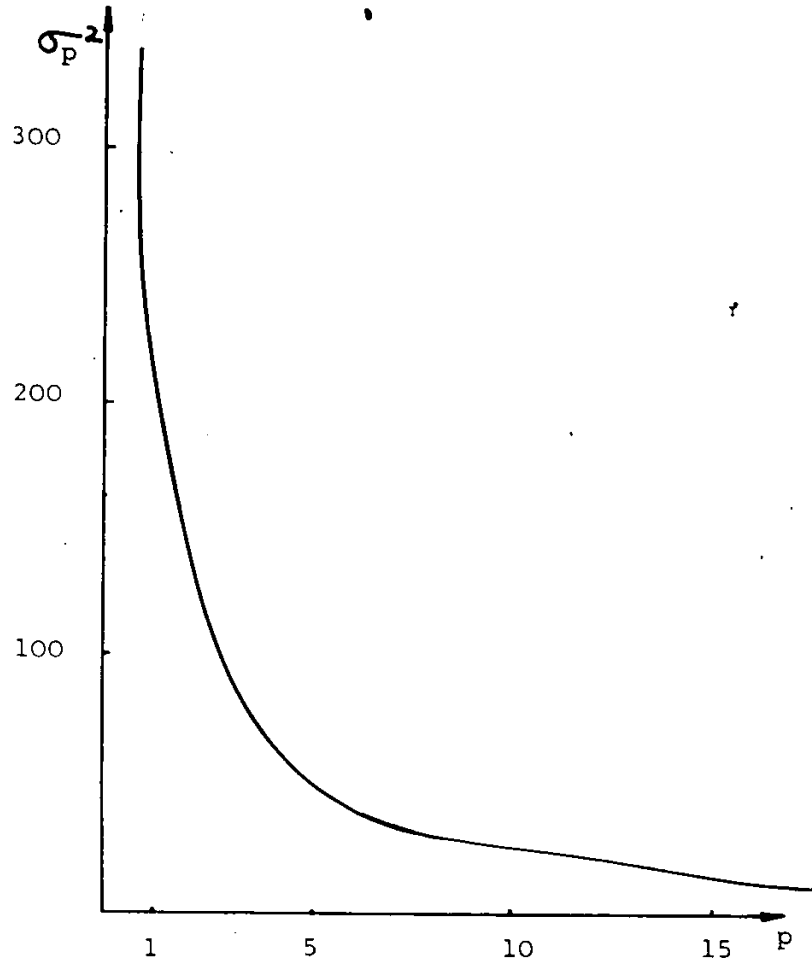


Figure 4.3.1 Modelling error vs model order

4.4 Choice of M:

In DTMF detection only the PSD at the eight discrete frequencies is required. It is unnecessary to evaluate the PSD at other frequency points.

The advantages of the AR modeling

- a. Good spectral resolution
- b. No restriction on the number of frequencies at which the PSD is estimated, and
- c. No restriction on the spacing between the frequency points

are exploited to evaluate the PSD at the eight frequencies of interest. Hence the number of frequency points is 8 i.e. $M=8$, as shown in Figure 4.4.1

A signal-to-noise ratio of 15dB is assumed and the PSD is obtained for all tone pairs with different levels according to the specifications. Two sample readings are given below. These readings show that the peaks are brought out clearly at the appropriate frequencies.

Tone 1 - 697 Hz	Tone 2 - 1633 Hz
N - 128	p - 14
697 - 0 dB	1209 - - 33 dB
770 - - 18 dB	1336 - - 32 dB
852 - - 26 dB	1477 - - 27 dB
941 - - 30 dB	1633 - 0 dB

This example is shown in Figure 4.4.1.

Tone 1 - 941 Hz

Tone 2 - 1633 Hz

N - 128

p - 14

697 - - 25 dB

1209 - - 27 dB

770 - - 28 dB

1336 - - 28 dB

852 - - 17 dB

1477 - - 26 dB

941 - 0 dB

1633 - 0 dB

In conclusion, simulation studies show that N-128, p-14 and M-8 provide satisfactory response and prove that AR modelling is an attractive alternative to existing methods in DTMF receivers.

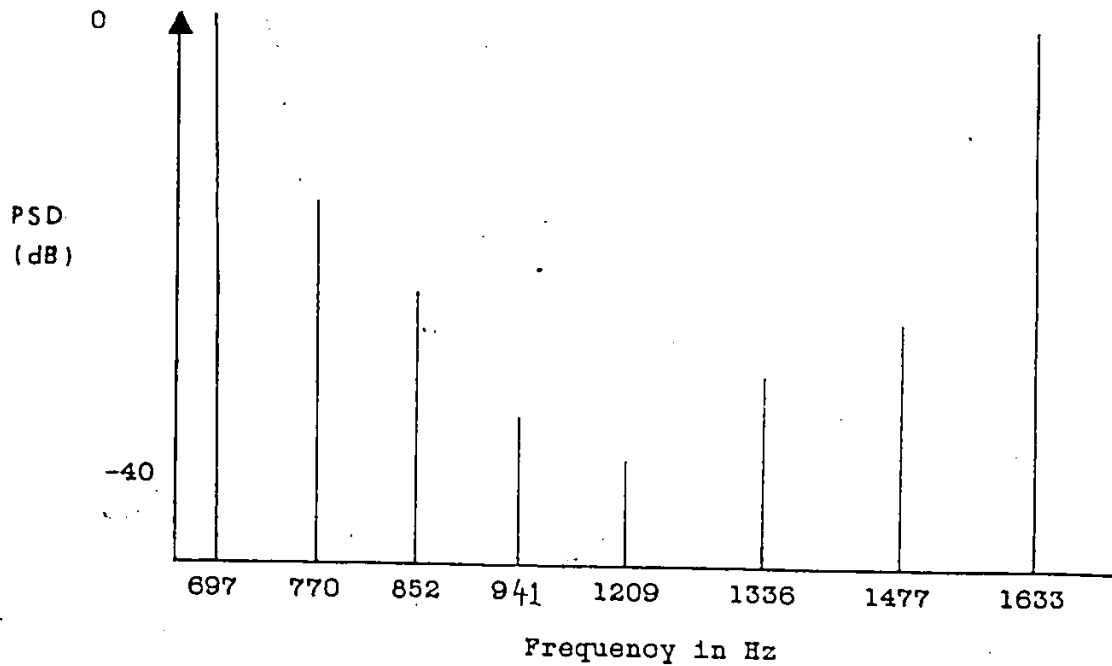


Figure 4.4.1: Points at which PSD is Evaluated

CHAPTER 5

IMPLEMENTATION

5.1 DTMF Receiver Task:

The main task of the DTMF Receiver is to identify the presence of a valid tone pair entering the switching office. To this end, the receiver estimates the power at the 8 discrete frequencies and determines that there is one valid tone in the low group and one valid tone in the high group. It then establishes the time duration of the tone pair. If the timing criteria is satisfied the receiver accepts the tone pair as a valid digit. The task of the receiver, then, could be sub-divided into the following sub-tasks viz.,

- 1 Collect N data samples (rectangular window assumed)
 X_n
- 2 Calculate $R(T)$, the autocorrelation function from X_n
- 3 Obtain the AR parameters
- 4 Estimate the PSD(f) at the low group and high group frequencies
- 5 Determine whether the digit is present or not.
- 6 Determine the time-interval the digit is present.
- 7 Output the digit if and only if the digit is present for 2 consecutive blocks and absent for 2 consecutive blocks. (1 block corresponds to 128 data samples)
- 8 Continue to repeat steps 1 to 7

It should be noted that steps 1 to 4 constitute the signal processing task of the receiver and steps 5 to 7 correspond to the high level decision task of the receiver.

5.2 Implementation Approach

It was decided to develop a microprocessor based system since it offers ease of implementation and flexibility.

The number of multiplications involved is:

$$N \times p + O(p^2) + 2p \times M$$

At the 8 KHz sampling rate, which is the standard in the telephone industry, collection of 128 data samples will require 16 msec. Hence for $N=128$, $p=14$ and $M=8$, multiplications of the order of 2000 would have to be performed in 16 mseconds.

Until now since multiplication took more time to be done than addition, a good measure of the complexity of algorithms was the number of multiplications involved. However with the advent of microprocessors like the TMS 32010 and 32020 which take the same time to execute addition or multiplication this measure of the complexity of an algorithm is no longer valid.

5.2.1 Choice of a Microcomputer

The computational complexity of the model and the real-time constraints rule out the use of general purpose microprocessors, more so because they usually do not have hardware multipliers. Hence the choice was narrowed down to programmable signal processors like the Texas Instruments TMS 320, NEC 7720 and Intel 2920. A comparative study (Table 5.2.1 and Table 5.2.2) indicated that the TMS 320 was best suited for the job.

The TMS 320 has been tailored, both in hardware and software, to meet general digital signal processing needs. It's instruction set, on-chip hardware multiplier (16 bit by 16 bit multiplication giving a clean 32 bit product), 200 ns instruction cycle, a 32 bit wide accumulator and 32 bit wide internal data path meets our requirements. Texas Instruments also supports an evaluation kit (development kit) and good software tools to facilitate development.

Table 5.2.1 Comparison of Various DSPs

Feature	TI TMS3210	NEC UPD7420	INTEL 2920
Data word			
size (bits)	16	16	25
Coefficients size (bits)	16	13	variable
Accumulation width (bits)	32	16	28
Saturation arithmetic	Hardware	Software	Hardware
Multiplier implementation	Hardware	Hardware	Software
Multiplier precision	16X16-32	16X16-32	12X25-
multiplication			
Time (nanoseconds)	200	250	4800
(Parallel I/O (bits)	16	8	4 in/8 o
Instruction word (bits)	16	23	24
Instruction cycle (ns)	200	250	400
Subroutine levels	>50	4	None
Iteration (loop counter)	YES	NO	NO
Conditional jumps	YES	YES	NO
Full speed external	YES	NO	NO
Memory expansion			
Instruction ROM (bits)	1536X16	512X23	192x24
Coefficient ROM	Not required	512x13	-
Data RAM (bits)	144C16	128X16	40x25
-1			
Z function	YES	YES	NO
Look up tables	YES	YES	NO

Table 5.2.2. Benchmark Comparison

	TI	NEC
	TMS32010	UPD7720
Biquad Filter Element	2.0 usecond	2.25 ⁴ us
Sine or Cosine	4.8 us	5.25 us
U/A law to		
Linear conversion	0.8 us	0.5 us
32 point complex FFT	0.254 ms	0.7 ms
64 point complex FFT	0.580 ms	1.6 ms

5.2.2: Programming Technique:

In programming two approaches viz.

- a) Normal programming and
- b) In-line coding

were considered.

Normal programming requires less memory but takes a longer time to execute. Inline code takes less time to execute but requires more memory space. Hence it was decided to use the normal programming technique and see if real-time criteria were met. If not, then, only those time-critical blocks (blocks that require more of processing time as compared to others) could be coded using inline code.

5.3 R(T), the Autocorrelation function

Fixed point arithmetic is assumed. The equations to be implemented are:

$$R(m) = \frac{1}{N} \sum_{n=0}^{N+m-1} X_n X_{n+m}, \quad m = 0, 1, \dots, (p+1). \quad (5.3.1)$$

It should be noted that this set of equations requires $N \times p$ multiplications and contains the maximum number of multiplication as compared to the other operations required. Consequently, it contributes to a major portion of the total execution time and care must be exercised to keep the execution time to a minimum.

For input data of 12 bits and a maximum value of X equal to 2047,

$$R(0) = 1/N \left[X_0^2 + X_1^2 + \dots + X_n^2 \right]$$

$$= 1/128 \left[X_0^2 + X_1^2 + \dots + X_{127}^2 \right]$$

Then

$$R(0)_{\max} = 1/128 \left[2047^2 + 2047^2 + \dots + 2047^2 \right] \quad (5.3.2)$$

To evaluate the above expression, one has to add 24 bit words and since in the worst case all X_n can be 2047, adding 2047^2

128 times gives an answer that is 29 bits long. Since the TMS

320 has a 32 bit accumulator, overflow problems do not arise in calculating R(T). R(T) are scaled down by 2047 i.e.

$$R_1(T) = R(T)/2047 \quad (5.3.3)$$

Again to simplify division,

$$R'(T) = R(T)/2048 \quad (5.3.4)$$

is done since this means shifting R(T) obtained in the accumulator by 11 bits. This also shows that by scaling by 2048 the maximum word length of R(T) will be 11 bits and this answer can be stored in the 16 bit registers

i.e. $\left[X_0^2 + X_1^2 + \dots + X_{127}^2 \right]$ requires 29 bits

Dividing by 128 implies

$$R(T) = 1/128 \left[X_0^2 + X_1^2 + \dots + X_{127}^2 \right]$$

requires 29 - 7 = 22 bits

Scaling by 2048 implies

$$R'(T) = R(T)/2048 \text{ requires } 22-11 = 11 \text{ bits}$$

The sampled data is read into the data memory and $R(T)$ calculated. Looping is used and both the auxiliary registers are used as pointers. Now out of the 144 word data RAM, 128 words are occupied by data; hence only 15 registers are left. The maximum model order that can be implemented is thus 14, since one register is used as a counter. This way the data RAM has been used to its capacity. An efficient way of calculating $R(T)$ is to use the following loop.

```

LARP 1
ZAC
MPYK 0
MT  LT *,0      *
    MPY * -.1   * Form product
    APAC        * Accumulate
BANZ MT

```

The data memory occupied is 128 words for digitized data samples and 15 words for R(T)

The execution time is 2.0184 milliseconds

5.4 AR Parameter Calculation:

In this block the calculated a_{kk} and a_{ki}

values are stored into data RAM locations 71, 72, ..., 71 + k and then copied into locations 21, 22, ..., 21 + k before proceeding to calculate the next set of AR parameters

The execution time is 1.3064 milliseconds

5.5 PSD(f) Calculation

The power spectral density is given by (3.2.4) and is repeated here

$$\text{PSD}(f) = \frac{\sigma_p^2 \Delta t}{\left| 1 + \sum_{k=1}^p a_{pk} e^{-j2\pi f k \Delta t} \right|^2} \quad (5.5.1)$$

For each frequency, the exponent has to be evaluated p times; then p multiplications have to be performed followed by a division. Thus the evaluation of the PSD for each frequency is a time-consuming task.

By simulation, it was observed that the denominator of the above expression took large values and required more than 16 bits to represent. Division, by itself, should be avoided since the signal processors do not yet have a "hardware" divider. Close examination of (5.5.1) reveals that since σ_p^2 and Δt are independent of the frequency at which PSD (f) is estimated, the PSD can be evaluated by calculating only the denominator. This is further indicated by the fact that since we are interested only in the maxima of the PSD, we could as well look for the minima of the denominators. This would eliminate the division involved for estimation of the PSD at each frequency and hence save a lot of processing time.

An efficient way of calculating the PSD at each frequency point is accomplished by representing the denominator of (5.5.1) by

$$\text{DENOM}(f) = \left| 1 + \sum_{k=1}^P a_{pk} e^{-j2\pi f k \Delta t} \right|^2 \quad (5.5.2)$$

$$= |A|^2$$

Now

$$A = 1 + \sum_{k=1}^P a_{pk} \left(\cos 2\pi f k \Delta t + j \sin 2\pi f k \Delta t \right)$$

$$= 1 + a_{p1} \cos 2\pi f \Delta t + a_{p2} \cos 2\pi f 2\Delta t + \dots + a_{pP} \cos 2\pi f P \Delta t$$

$$+ j \left(a_{p1} \sin 2\pi f \Delta t + a_{p2} \sin 2\pi f 2\Delta t + \dots + a_{pP} \sin 2\pi f P \Delta t \right)$$

$$= \text{PRSUM} + j \text{IRSUM}$$

where

$$\text{PRSUM} = 1 + \left\{ a_{p1} \cos 2\pi f \Delta t + a_{p2} \cos 2\pi f 2\Delta t + \dots + a_{pP} \cos 2\pi f P \Delta t \right\}$$

$$\text{and IRSUM} = \left\{ a_{p1} \sin 2\pi f \Delta t + a_{p2} \sin 2\pi f 2\Delta t + \dots + a_{pP} \sin 2\pi f P \Delta t \right\}$$

$$\therefore \text{DENOM}(f) = \text{PRSUM}^2 + \text{IRSUM}^2 \quad (5.5.3)$$

The large number of calculations and house keeping required for each frequency shows the need to keep the number of frequencies to a minimum.

A way to reduce the execution time is to precompute the cosine and sine values for the 8 frequencies. This requires that 112 cosine and 112 sine values be stored in a table. This is shown in Tables 5.5.1 and 5.5.2

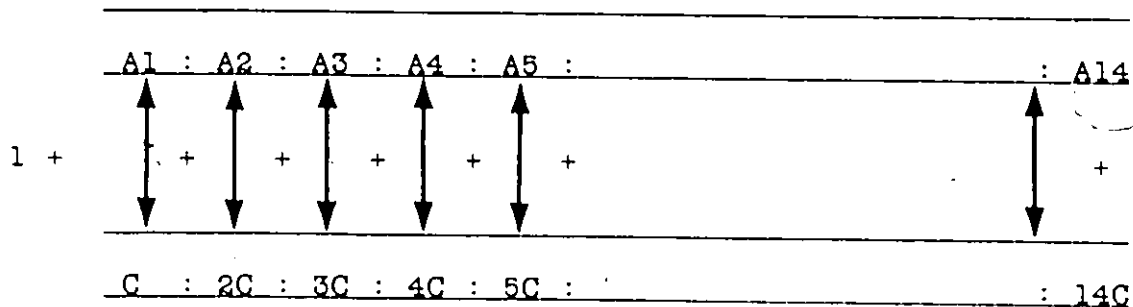
TABLE 5.5.1: Storage of Cosine Co-efficients and computation of PRSUM for one frequency point.

F1	:	C1	:	2C1	:	3C1	:	4C1	:	5C1	:	14C1	:
F2	:	C2	:	2C2	:	3C2	:	4C2	:	5C2	:	14C2	:
F3	:	C3	:	2C3	:	3C3	:	4C3	:	5C3	:	14C3	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:
F8	:	C8	:	2C8	:	3C8	:	4C8	:	5C8	:	14C8	:

COSINE COEFFICIENTS TABLE

F1, F2, F3, F4, F8 are the eight frequencies of interest.
 C1, C2, 14C8 are the pre-computed cosine values. (Total 112 values)

PRSUM COMPUTATION FOR ONE FREQUENCY POINT.



PRSUM

$$\text{PRSUM} = 1 + A1 \cdot C + A2 \cdot 2C + A3 \cdot 3C + \dots + A14 \cdot 14C$$

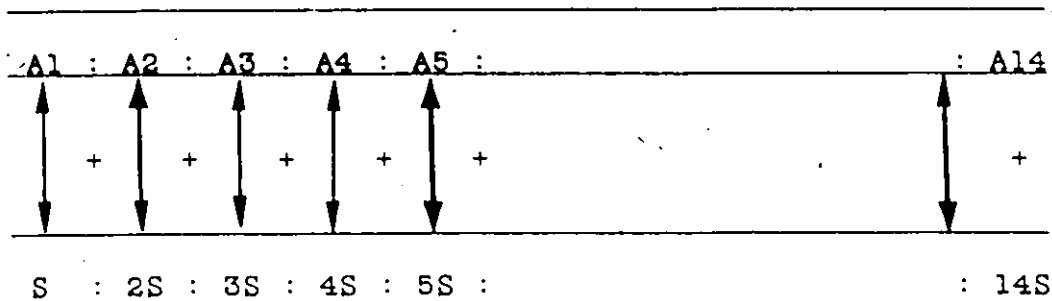
TABLE 5.5.2: Storage of sine coefficients and computation of IRSUM for one frequency point.

F1	:	S1	:	2S1	:	3S1	:	4S1	:	5S1	:	:	14S1	:
F2	:	S2	:	2S2	:	3S2	:	4S2	:	5S2	:	:	14S2	:
F3	:	S3	:	2S3	:	3S3	:	4S3	:	5S3	:	:	14S3	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:	:
.	:	:	:	:	:	:	:	:	:	:	:	:	:	:
F8	:	S8	:	2S8	:	3S8	:	4S8	:	5S8	:	:	14S8	:

SINE COEFFICIENTS TABLE

F1, F2, F3, F4, F8 are the eight frequencies of interest.
 S1, S2, 14S8 are the pre-computed sine values. (Total 112 values)

IRSUM COMPUTATION FOR ONE FREQUENCY POINT.



IRSUM

$$\text{IRSUM} = A1 \cdot S + A2 \cdot 2S + A3 \cdot 3S + \dots + A14 \cdot 14S$$

Since the cosine and sine values are precomputed and stored, one need only multiply the AR parameters with the corresponding cosine/sine values and accumulate the sums. Thus the execution time is significantly reduced.

The execution time is 0.6752 milliseconds

5.6 Storage and Switching of Input Data

The model requires that all N data samples be available simultaneously before calculations can begin. Since input data is entering continuously, some sort of storage has to be provided for in order to capture the next set of samples while processing the current set of N samples.

The TMS 320 has a data memory of 144 words. Since $N=128$ and $p=14$ the data RAM has been used to its capacity.

One possible approach can be to increase the data RAM from 144 words to 288 words or more. One part of it could be used to collect the data entering during processing of the current set of samples. Subsequently the part of the RAM that had collected the second set of data could be switched in and the first part of the data RAM could be made to collect the next samples. But this requires an increase both in hardware and in design complexity.

There is a way around this problem. The TMS320 has a Harvard architecture. This means that the program memory is separate from the data memory. There are 1536 words of program RAM provided on chip. Program memory expansion is implemented by an external pin defined as MC/Mp. Table 5.6.1 illustrates the program memory capability of the TMS320 family.

Table 5.6.1

MODEL	PROGRAM MEMORY OPTIONS	MICROCOMPUTER MODE MEMORY	MICROPROCESSOR MODE MEMORY
TMS 5320M10	MICROCOMPUTER AND MICROPROCESSOR MODES	1536 WORDS ON CHIP MEMORY AND 2560 WORDS	4096 WORDS OF EXTERNAL MEMORY
TMS 32010	MICROPROCESSOR MODE ONLY	NONE	4096 WORDS OF EXTERNAL MEMORY

The microcomputer mode is defined by a one level on the MC/Mp pin. In this mode, 1524 words of on-chip memory are available. The device architecture allows for an additional 2560 words of external program memory.

The microprocessor mode is defined by a zero level on the MC/Mp pin. All memory (4096 words) is external in this mode. The memory map of the TMS 320 is shown in Figure 5.6.1.

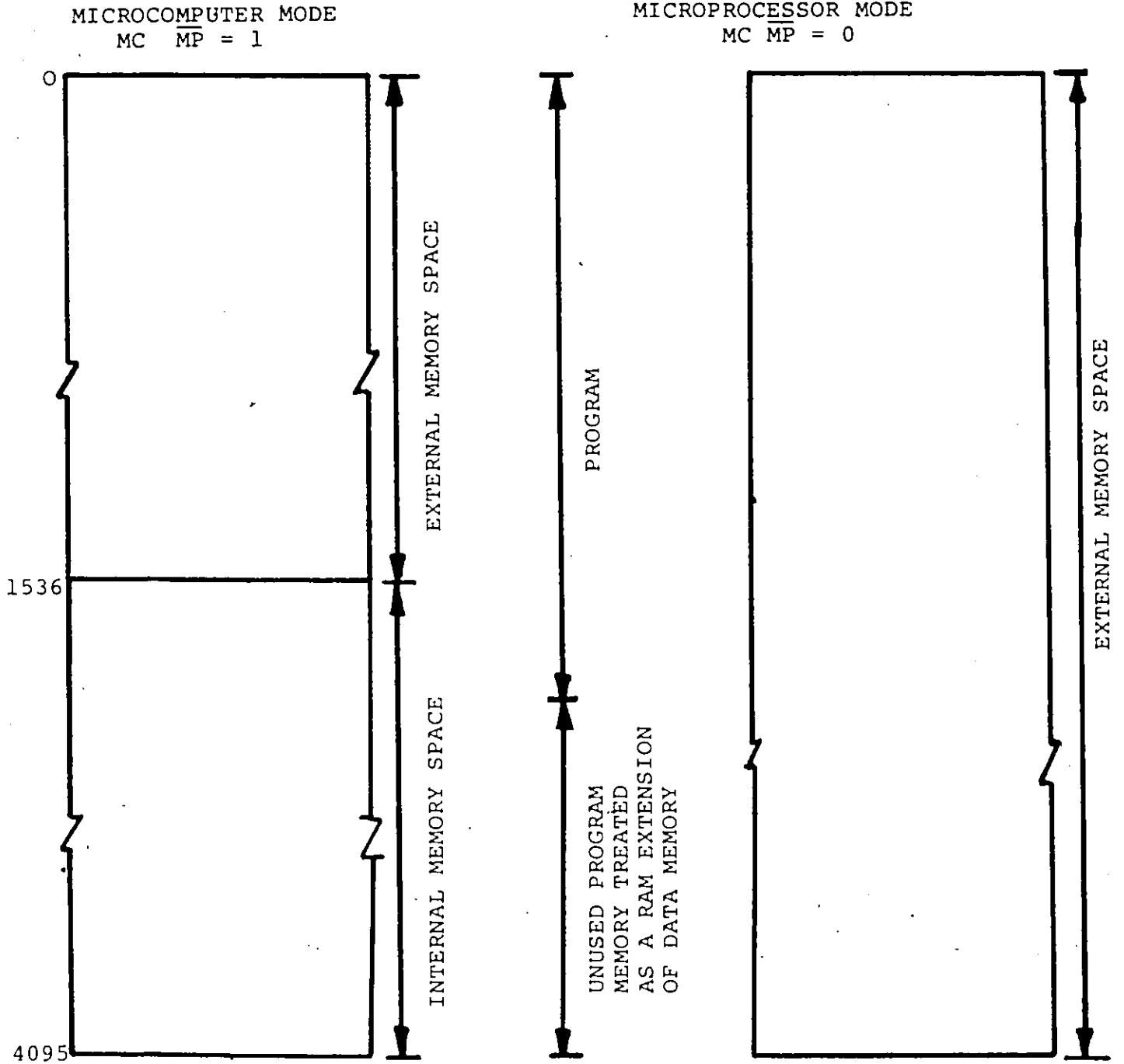


Figure 5.6.1. TMS 320 Memory Map

This large program memory of 4k words suggests that the unused program memory space can be treated as a RAM extension of the data memory. This technique is very efficient in terms of additional hardware requirements since it would just mean adding another RAM chip to the TMS 320. The only resulting complexity would be in the software.

This concept has been utilized in the receiver implementation to overcome the disadvantage of the small data memory. When the current set of samples (which is in the data RAM) is being processed, the next set of samples is collected by the interrupt mode and stored in the program RAM. A variable called the data pointer is used to point to the next memory location and another variable is used to indicate the number of data samples collected. After execution of the main program, data from the program RAM is transferred to the data RAM and the remaining data samples collected.

Our idea that the remaining space in the program RAM could be treated as an extension of the data RAM was tested. It takes 21,902 clock-cycles i.e. 4.3804 ms to execute the main program. Sampling at 8KHz would mean 35 data should be collected during this time. It was verified that 35 data samples were in fact collected. Overhead incurred by the interrupt routine - $120/625 = 19.2\%$

We have experienced that the notion that the TMS 320 has a relatively small data memory is incorrect. For most telephony and speech processing applications, the sampling frequency is 8KHz. This implies that there is 125 usecond interval between two samples. This provides ample time to transfer data from program RAM to data RAM as and when needed. If the application program is restricted to occupy less than 3500 lines (which is true for most applications) the remaining space in the program memory can be treated as a ~~RAM~~ extension of the 144 word data memory.

5.7 High Level Decision:

The signal-processing task of the receiver estimates the PSD at the appropriate frequencies. It is now the responsibility of the high-level decision function of the receiver to determine whether a digit has been transmitted or not. To this end, the receiver scans the eight PSD values and a digit hypothesis is generated whenever

1. A valley point LGMIN (local minimum in the low-group frequencies) and a valley point HGMIN (local minimum in the high-group frequencies) have been confirmed.
2. The threshold criteria are met. (These are discussed in detail in the next section)

A digit will then be declared to have been recognized under two conditions being satisfied viz., .

1. The same tone-pair is continuously present for at least two consecutive sets of data.
2. The above tone-pair is then (i.e. after it is present for two consecutive sets of data) followed by a silence (i.e. no tone-pair) for at least two consecutive sets of data.

The complete flowchart is given in Figure 5.7.1 and the flow-chart for the timing decisions is shown in Figure 5.7.1.(b) & (c). VDPR (valid digit present) is a variable that's incremented by 1 every time a valid tone pair has been detected. The first time i.e. when VDPR=1, the generated digit hypothesis is stored in CD1 (current digit for the first pass). If the tone-pair is present in the succeeding set of data as well, VDPR becomes 2, the generated digit hypothesis CD2 (current digit for second pass) is now compared with CD1. If $CD1 = CD2$, then it is decided that the digit is present for two consecutive sets of data by setting PRESENT = 2, and the digit hypothesis is stored in CD3; else if $CD1 \neq CD2$ then the digit hypothesis in CD1 is discarded, and PRESENT is set to 0 i.e. PRESENT = 0. The receiver now waits to observe a silence before confirming the hypothesis.

Thus when the tone-pair is absent for the succeeding set of data, VDPR along with CD1 and CD2 is reset. Now VDAB, valid digit absent, is incremented. The receiver then tests the variable PRESENT; if PRESENT = 2 it implies that a digit hypothesis was generated in the preceding sets of data. If PRESENT = 2, AD1, absent digit, is set to zero. When the digit is absent even in the next succeeding set of data, AD2, absent digit for second pass, is set to zero. Now AD1 is compared with AD2. If AD1 = AD2, the generated digit hypothesis is confirmed and the receiver outputs the valid digit; else the generated digit hypothesis is discarded, PRESENT, VDAB, AD1 and AD2 are all reset.

It is to be noted that the structured software enables different combinations of intervals of the digit being present and absent, to be incorporated if needed. Thus it is quite easy to make the logic accept a digit when the tone-pair is present for two consecutive sets of data and absent for one set of data or any suitable combination.

The software for the timing decisions occupies a relatively large memory space. This is because a certain amount of book-keeping has to be done. But as multiplications and loops are not present it does not take too large a portion of the total processing time. This also brings out the fact that the present signal processors are good number crunching microcomputers; however they are not so easy to program

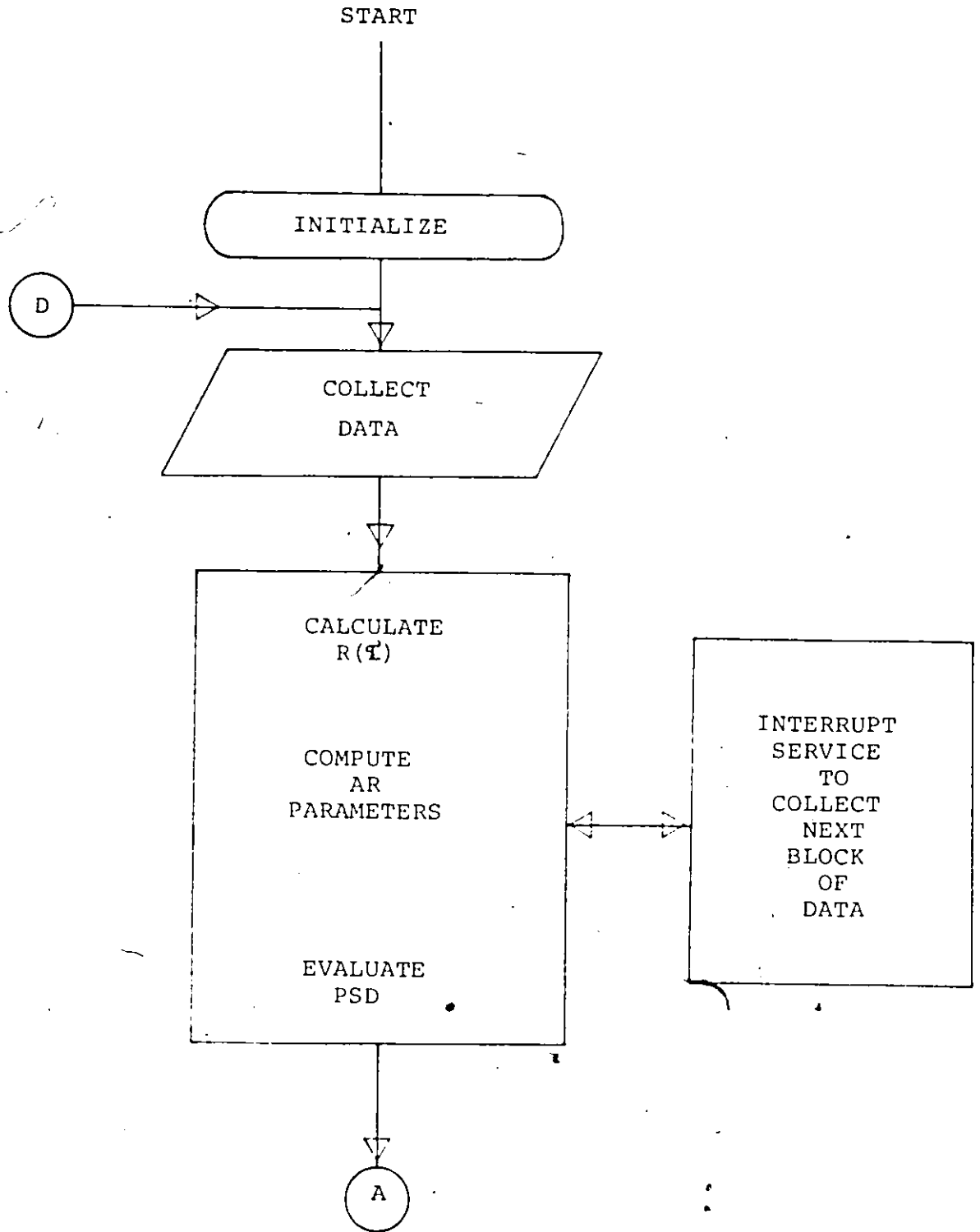


Figure 5.7.1(a) Detailed Flow-Chart

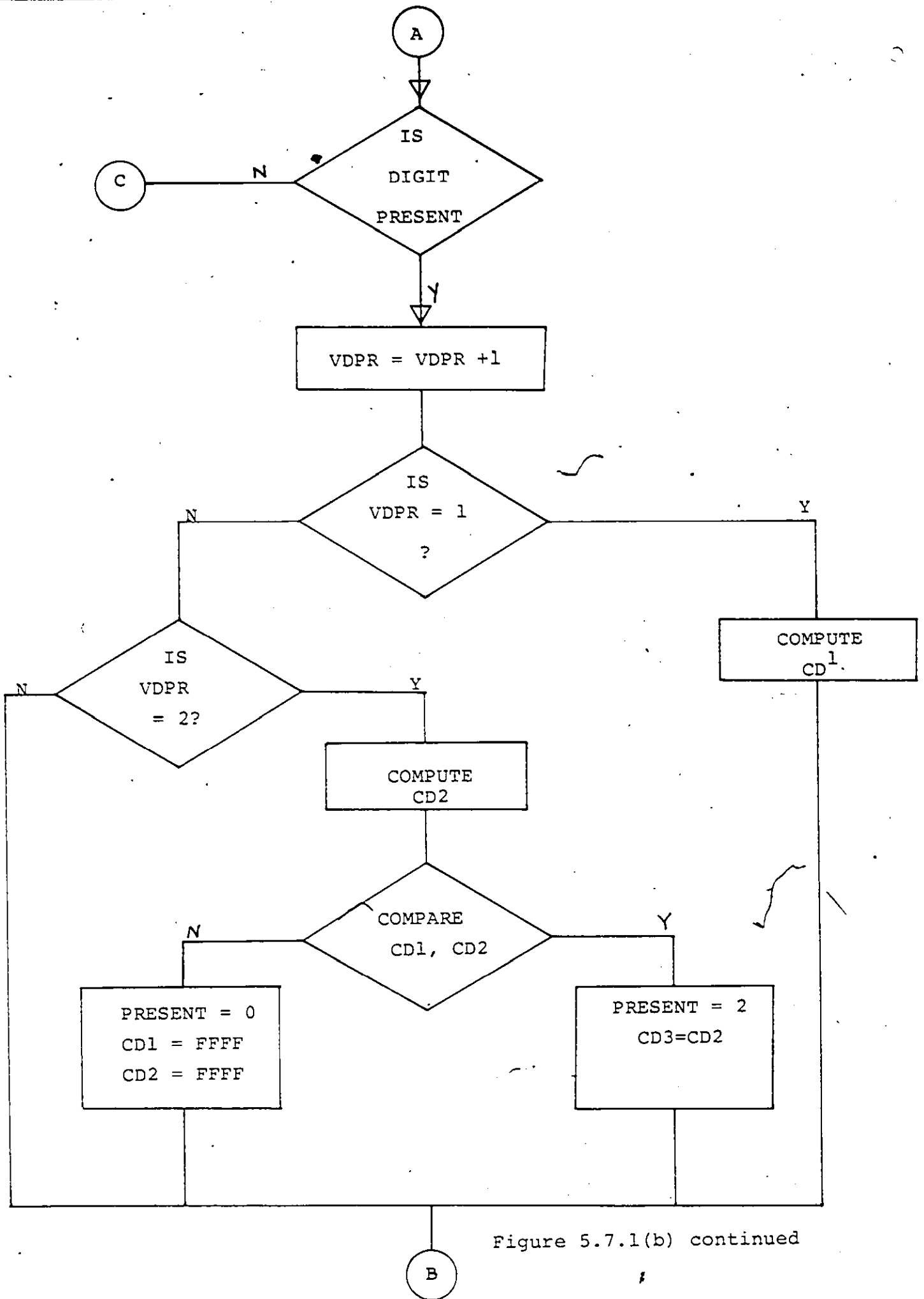


Figure 5.7.1(b) continued

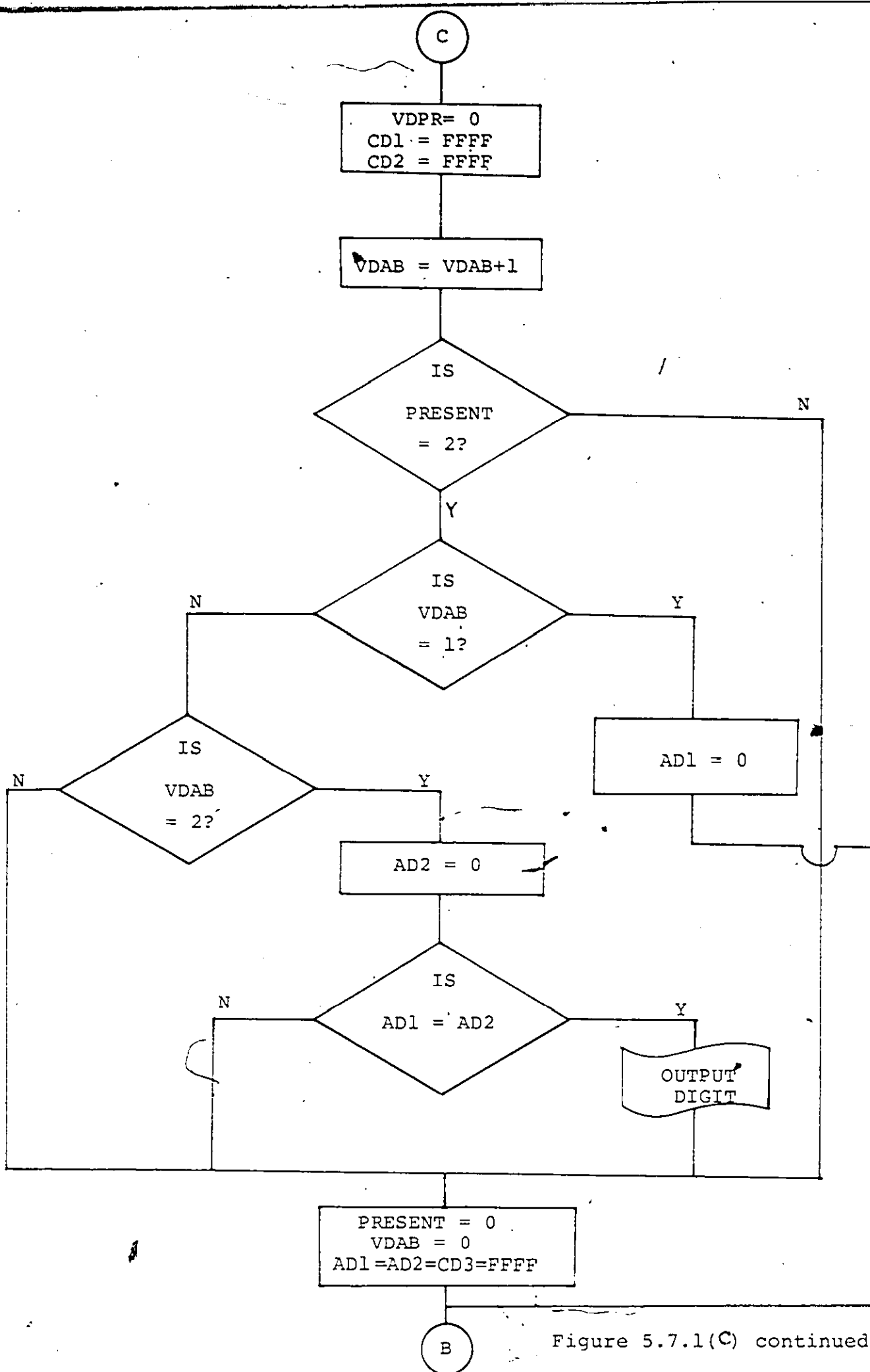
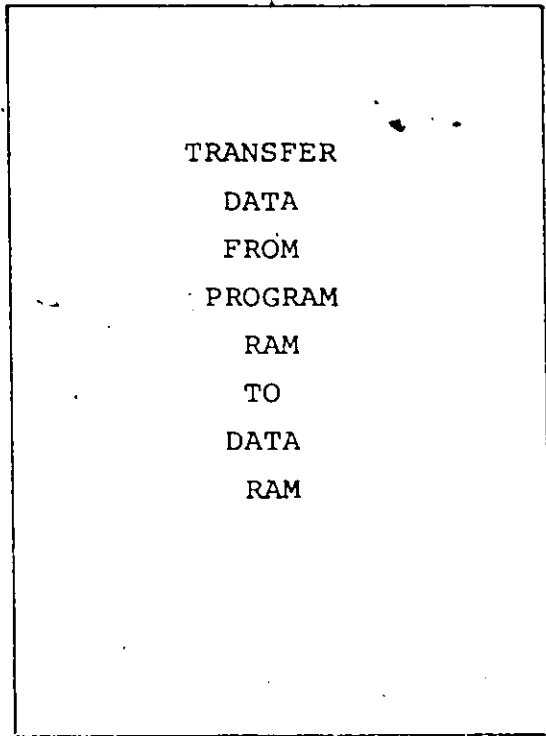


Figure 5.7.1(C) continued

B



D

logic involving a lot of decisions to be made e.g. if a number of

IF (conditions) THEN

ELSE

statements are to be implemented.

5.8 Threshold Setting Study

The Touch-Tone receiver computes the PSD at the eight desired frequencies. It then obtains the minimum in the low-group, LGMIN, and the minimum in the high-group, HGMIN. It has to now decide whether there is a valid tone-pair or not. It does this by comparing LGMIN against KL, the low-group threshold and HGMIN against KH, the high-group threshold. A valid tone-pair is one in which

a) $LGMIN < KL$

and

b) $HGMIN < KH$

The threshold values are determined empirically. The experimental setup used in this study is shown in Fig. 5.9.1. It is to be noted that each frequency is derived from a different source. The tones are added and then fed to the A/D converter. Each frequency combination is generated and the 8 PSD's computed by the AR model displayed and noted down. The experiment is repeated 10 times for each frequency combination. Since there are 16 different combinations of the frequencies, in effect 160 trials were performed. From this study it was decided that $KL=55$ and $KH=80$.

While observing the PSD's for each trial, σ_1 and σ_{14} the modelling errors at order 1 and order 14 were also noted. These are shown in Tables 5.8.1 and 5.8.2 . Each row of the table shows the maximum and minimum values taken by the modelling error for each trial.

Table 5.8.1: Maximum and Minimum Values
 taken by $\hat{\sigma}_1$, the modelling error
 at model order 1. Each row shows
 the Maximum and Minimum values
 taken out of 10 trials.

Maximum	Minimum
1602	1440
1442	1353
1499	1399
1520	1377
1465	1384
1478	1380
1465	1361
1526	1409
1529	1402
1586	1408
1561	1359
1579	1503
1602	1402
1584	1408
1541	1455
1548	1385

Table 5.8.2. Maximum and Minimum Values taken
by σ_{14} , the Modelling Error at order
14. Each row shows the Maximum and
Minimum values taken out of ten trials.
Thus 16X10 = 160 Trials were performed.

Maximum	Minimum
1090	927
1388	924
1032	823
1097	937
1075	944
1035	917
990	765
1497	861
951	723
908	847
965	793
962	789
971	756
986	809
1141	929
948	828

Maximum of σ_1 Maxima - 1602

Maximum of σ_1 Minima - 1442

Maximum of σ_1 Maxima - 1503

Maximum of σ_1 Minima - 1353

Hence KS_1 , the threshold for 1 is 1100 i.e.

$$KS_1 = 1100$$

Maximum of σ_{14} Maxima - 1497

Maximum of σ_{14} Minima - 908

Maximum of σ_{14} Maxima - 944

Maximum of σ_{14} Minima - 723

Hence KS_{14} , the threshold for 14 can be fixed as 500 i.e.

$$KS_{14} = 500$$

The logic for introducing the thresholds KS_1 and KS_{14} instead of just KL and KH is that more number of thresholds may mean greater immunity of the receiver against false detection.

Table 5.8.3 list the values taken by σ_1 and σ_{14} when no tone-pair is input i.e. when there is only noise.

Table 5.8.3: 1 and 14 Values When No Tone Pair Is Input.

1	14
110	37
124	57
112	34
110	40
110	7

Thus a tone-pair would be accepted as a valid tone-pair if and only if

$$\sigma_1 \rightarrow KS1 = 1100$$

and

$$\sigma_2 \rightarrow KS14 = 500$$

Table 5.83 shows that when no tone-pair is present and σ_1 and σ_{14} computed, condition

$\sigma_1 \rightarrow KS1$ would not be satisfied; similarly condition

$\sigma_{14} \rightarrow KS14$ would also not be satisfied.

5.9 DTMF Receiver Test

The experimental set-up used is shown in Figure 5.9.1. The Mitel DTMF Receiver test cassette has been used. Two tones are generated and then corrupted by white Gaussian noise. The resulting signal is then passed through a band-pass filter where the higher cut-off frequency is 4 KHz. The

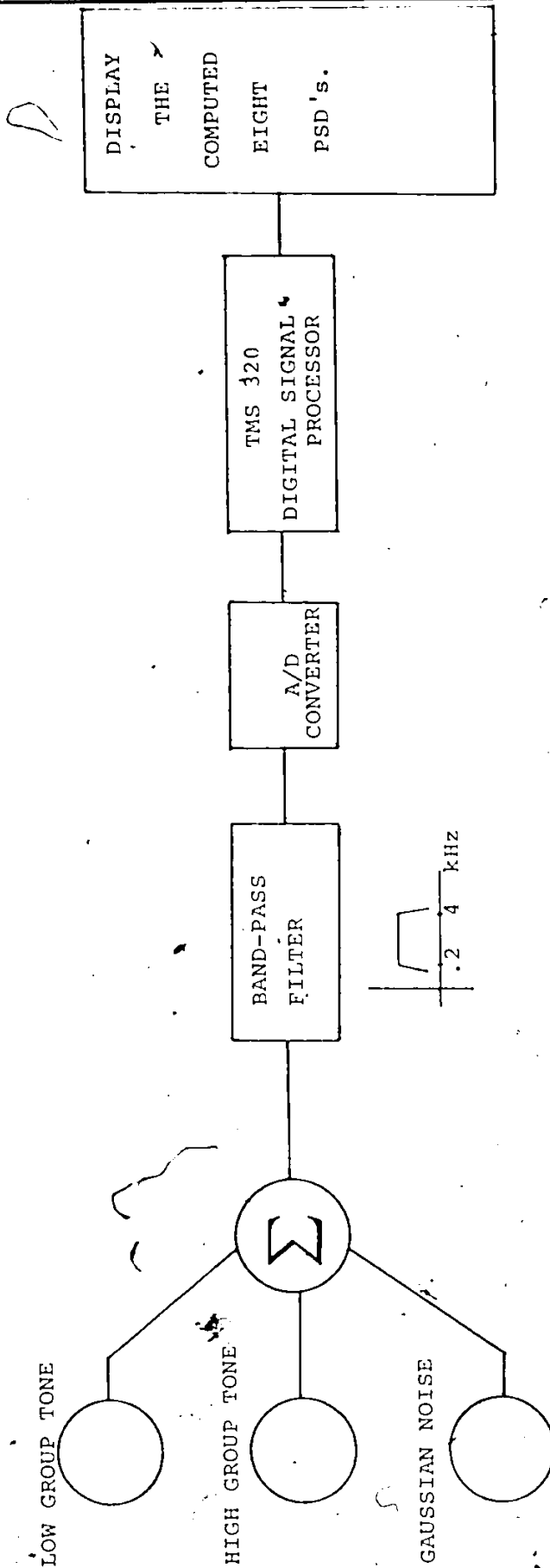


Figure 5.9.1. Experimental Setup

filter output is fed to an A/D converter and the TMS320 collects the data samples via the interrupt mode. Power of each tone is measured after removing the other tone and the noise signal. It is measured at the A/D input. The set-up enables the tones to be varied independently, both in frequency and in signal level. In Figure 5.9.2 a digit waveform in the time domain without noise is shown. The several tests conducted are then shown, with their results, in the following sections.

1. Decode Check

All tone-pairs associated with standard 4x4 keypad digits are generated and the receiver tested. It is observed that the receiver responded to all tone-pairs it is designed to receive. Figure 5.9.2 shows a digit waveform as a function of time.

A sample reading is given below.

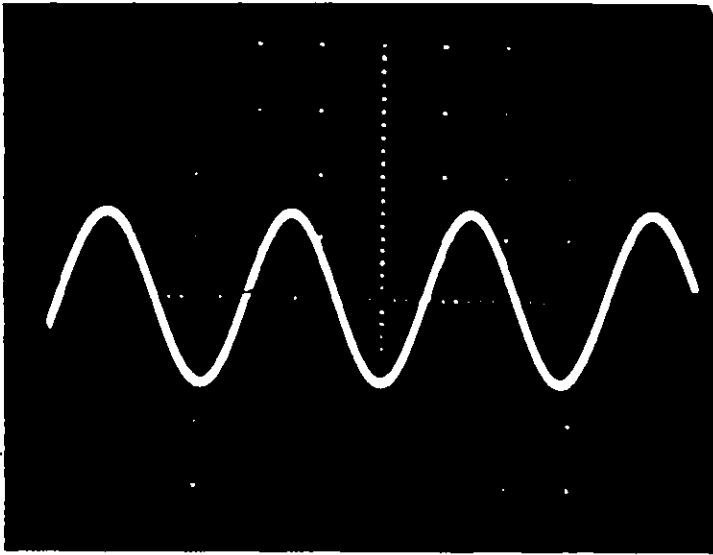
Tone 1 - 697 Hz at 0.2 V
rms

Tone 2 - 1336 Hz at 0.2 V
rms

S/N - 20

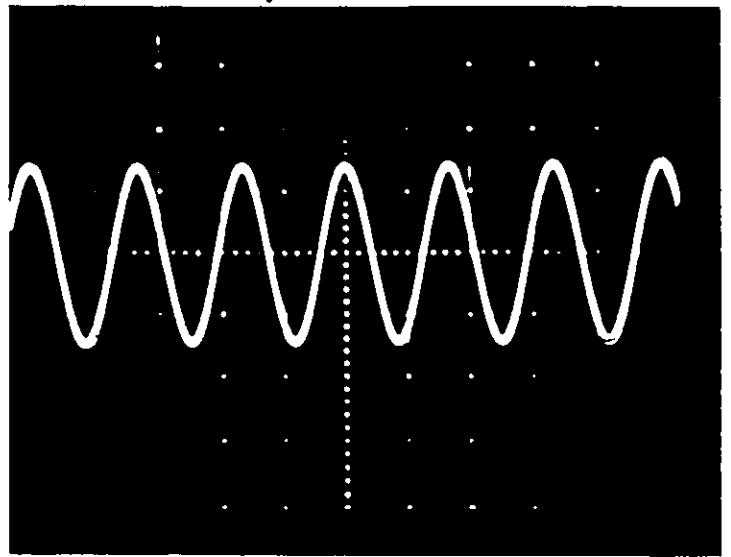
Frequency deviation - 0%

of data samples - 128



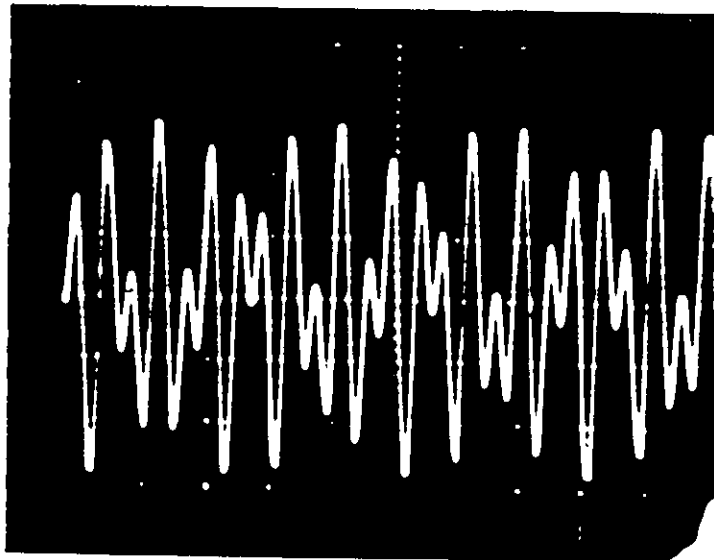
a) Tone 1 at 697 Hz

Scale: Vertical - 1 volt/div.



b) Tone 2 at 1209 Hz

Scale: Vertical 1volt/div.



c) Sum of Tones 1 & 2

Scale: Vertical - 1 v/div.

Horizontal - 0.5 m/div

Figure 5.9.2 Digit Waveform in Time-Domain

S/N - 15 dB

697 - 48

770 - 398

852 - 1218

941 - 1645

1209 - 78

1336 - 708

1477 - 3225

1633 - 3845

S/N - 10 dB

697 - 208

1209 - 181

770 - 752

1338 - 948

852 - 1452

1477 - 3242

941 - 1645

1633 - 3624

We observe that at S/N - 10 dB, eventhough the valleys are obtained at the appropriate frequencies, they exceed the thresholds for the low group (KL-55) and high group (KH-80).

3. Talk-Off Test*

The test consists of recordings of conversations on telephone trunks made over a long period of time and condensed into a 30 minute period. Receiver immunity to talk-off is determined by the number of responses occurring during this test. According to Mitel (DTMF Receiver Test Cassette CM7291) [7] the receivers response to this test may vary from 20 to 100. In the experiment performed the receivers response was 41 indicating a satisfactory talk-off performance.

4. Dynamic Range

This test utilizes the tone-pair 697,1209. The amplitude of each is gradually attenuated to -38 dB. The receiver performed well in the amplitude range 0 to -18 dBm0.

A few sample readings are given below.

Tone 1 - 697 at 0.7 V

Tone 2 - 1209 at 0.7 V

697 - 42

1209 - 54

770 - 408

1336 - 7845

852 - 756

1477 - 2248

941 - 985

1633 - 3124

Tone 1 - 697 at 0.1 V

Tone 2 - 1209 at 0.1V

697 - 68

1209 - 74

770 - 298

1336 - 652

852 - 945

1477 - 2348

941 - 1243

1633 - 2802

Performance deteriorated below 0.01 V.

In order to meet the dynamic range of 26 dB as specified, the threshold values will have to be modified. But modification of the threshold values to meet this specification resulted in an increase in the number of false alarms to speech.

CHAPTER 6

CONCLUSION

A novel approach to the Dual Tone Multifrequency Receiver has been presented. Linear prediction or autoregressive time series modelling is used. The approach offers the advantages of improved spectral selectivity and greater resolution. The model has been implemented on the TMS320 digital signal processor.

The objectives of suggesting an attractive alternative to existing methods in DTMF receiver realization and implementing it have thus been met. It has also been shown that highly numeric intensive algorithms which had so far been of purely an academic interest, can now be implemented in real-time, thanks to the high performance TMS320 digital signal processor.

- [1] B. I. Pawate, W. Steenaart and B. Sankur, "The DTMF Receiver Based on Linear Prediction", 27th Midwest Symposium on Circuits and Systems, Morgantown, W.V., pp. June 11-12, 1984 pp. 258-260.
- [2] L. Schenker, "Pushbutton Calling with a Two-Group Voice-Frequency Code", Bell System Technical Journal, New York, N.Y., vol. 39, Jan 1960, p.235
- [3] R.L. Deininger, "Human Factors Engineering Studies of the Design and Use of Pushbutton Telephone Sets", *ibid.*, vol. 39, May 1960, p.995
- [4] R.N. Battista, C.G. Morrison, D.H. Nash, "Signalling System and Receiver for Touch-Tone calling", IEEE Proc., March 1963, pp. 9-17
- [5] Greg Klowak and Sorin Cohn-Sfetou, "Effective Multifrequency Receiver Design", ICASSP 85 Proc., vol. 4 of 4, March 1985, pp 1629-1632
- [6] E. Fok, J. Mullen, "DTMF Receivers for Touch-Tone Data Terminals, "Comm. International, pp. 42-46, Feb 1963.
- [7] Mitel's Microelectronic Products for Telecommunication.

- [8] S. Cohn-Sfetou and E. Munter, "Digital DT/MF Receivers," BNR Memo, file H160, July 1976
- [9] J.R. Bobbie, N. Sachs, J. Tow, "Receiver for Touch-Tone Service," B.S.T.J., vol. 60, no 7, pp. 1573 -1583, Sept. 1981.
- [10] C.R. Baugh, "Design and Performance of a Digital MF Receiver," IEEE Trans. on Comm. June 1977.
- [11] I. Koval, G. Gara, "Digital MF Receivers using DFT," IEEE Trans. on Comm., vol. com. - 211, no. 12, pp. 1331 - 1335, Dec. 1973.
- [12] Steven M. Kay, Stanley L. Marple, Jr., "Spectrum Analysis - A Modern Perspective", Proc. IEEE, vol. 69, no. 11, Nov. 1981, pp. 1380 - 1419.
- [13] B. Sankur, "An Investigation of DTMF Receivers Based on Non-Linear Spectral Estimation Algorithms," Univ. of Ottawa, Dept. of Elect. Eng., Feb. 1983
- [14] G.E.P. Bon and G.M. Jenkins, "Time Series Analysis: Forecasting and Control", San Francisco, CA; Holden-Day, 1970.

- [8] S. Cohn-Sfetou and E. Munter, "Digital DT/MF Receivers," BNR Memo, file H160, July 1976
- [9] J.R. Bobbie, N. Sachs, J. Tow, "Receiver for Touch-Tone Service," B.S.T.J., vol. 60, no 7, pp. 1573 -1583, Sept. 1981.
- [10] C.R. Baugh, "Design and Performance of a Digital MF Receiver," IEEE Trans. on Comm. June 1977.
- [11] I. Koval, G. Gara, "Digital MF Receivers using DFT," IEEE Trans. on Comm., vol. com. - 211, no. 12, pp. 1331 - 1335, Dec. 1973.
- [12] Steven M. Kay, Stanley L. Marple, Jr., "Spectrum Analysis - A Modern Perspective", Proc. IEEE, vol. 69, no. 11, Nov. 1981, pp. 1380 - 1419.
- [13] B. Sankur, "An Investigation of DTMF Receivers Based on Non-Linear Spectral Estimation Algorithms," Univ. of Ottawa, Dept. of Elect. Eng., Feb. 1983
- [14] G.E.P. Bon and G.M. Jenkins, "Time Series Analysis: Forecasting and Control", San Francisco, CA; Holden-Day, 1970.

- [15] G.U. Yule, "On a method of investigating periodicities in disturbed series with special reference to Wolfer's sunspot numbers," *Philosophical Trans. Royal Soc. London, Series A*, vol. 226, pp: 267-298, July 1927.
- [16] J.G. Berryman, "Choice of operator length for maximum entropy spectral analysis," *Geophys.*, vol. 43, no. 1383-1391, Dec. 1978.
- [17] E. Parzen, "Multiple time series modelling," *Dep. Statistics, Stanford, Univ., Stanford, CA, Tech. Rep. 12*, July 8, 1968 (Contract Nonr-225-80)
- [18] I. Barrodale and R.E. Erickson, "Algorithms for least-squares linear prediction and maximum entropy spectral analysis - Part I: Theory and Part II: FORTRAN Program," *Geophys.*, vol. 45, pp. 420-446, Mar. 1980.
- [19] J.P. Burg, "Maximum entropy spectral analysis," in *Proc. 37th Meeting Society of Exploration Geophysicists (Oklahoma City, OK)*, Oct. 31, 1967.

- [20] J. P. Burg, "A new analysis technique for time series data," NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, Enschede, The Netherlands, Aug. 12-23, 1968.
- [21] J. P. Burg, "New concepts in power spectral estimation," in Proc. 40th Annu. Int. Society of Exploration Geophysicists (SEG) Meeting (New Orleans, LA) Nov. 11, 1970.
- [22] J.F. Claerbout, Fundamentals of Geophysical Data Processing. New York: McGraw-Hill 1976
- [23] L.J. Griffens and R. Prieto-Diaz, "Spectral analysis of natural seismic events using autoregressive techniques," IEEE Trans. Geosci. Electron. vol. GE-15, pp. 13-25, Jan. 1977.
- [24] R.T. Lacoss, "Data adaptive spectral analysis methods," Geophysics, vol. 36, pp. 661-675, Aug. 1971.
- [25] T.E. Landers and R.T. Lacoss, "Some geophysical applications of autoregressive spectral estimates," IEEE Trans. Geosci. Electron., vol. GE-15, pp. 26-32, Jan. 1977.

- [26] R. N. McDonough, "Maximum entropy spatial processing of array data," *Geophysios*, vol. 39, pp. 843-851, dec. 1974.
- [27] D. E. Smylie, G.K.C. Clarke, and T.J. Ulrych, "Analysis of irregularities in the earth's rotation," in *Methods in Computational Physios*, vol. 13. B.A. Bolt, Ed. New York: Academic Press, 1973, pp. 391-430.
- [28] D.J. Thomson, M.F. Robbins, C.G. MacLennan, and L.J. Lanzerotti, "Spectral and windowing techniques in power spectral analyses of geomagnetic data," *Phys. Earth Planetary Interiors*, vol. 12, pp. 217-231, Aug. 1976.
- [29] T.J. Ulrych, "Maximum entropy power spectrum of long period geomagnetic reversals," *Nature*, vol. 235, pp. 218-219, Jan. 28, 1972.
- [30] T.J. Ulrych, "Maximum entropy power spectrum of truncated sinusoids," *J. Geophysical Research*, vol. 77, pp. 1396-1400, March 10, 1972.

- [31] J. Gibson, S. Haykin, and S. B. Kesler, "Maximum entropy (adaptive) filtering applied to radar clutter," in Rec. 1979 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, pp. 166-169.
- [32] S.S. Haykin, Ed., "Nonlinear Methods of Spectral Analysis". New York: Springer-Verlag, 1979.
- [33] S.M. Kay, "Autoregressive spectral analysis of narrowband processes in white noise with application to sonar," Ph.D. dissertation, Georgia Institute of Technology, Mar. 1980.
- [34] G. Prado and P. Moroney, "Linear predictive spectral analysis for Doppler sonar applications," Charles Stark Draper Laboratory, Tech. Rep. R-1109, Sept. 1977.
- [35] M. Hsu, "Maximum entropy principle and its application to spectral analysis and image reconstruction," Ph.D. dissertation, Ohio State Univ., 1975.
- [36] Dr. R. Moorcroft, "Maximum entropy spectral analysis of radioauroral signals," Department of Physics and Centre for Radio Science, Univ. Western Ontario, London, Ontario, Canada.

- [37] W. Gersch, "Spectral analysis of EEG's by autoegressive decomposition of time series," Math. Biosci., vol. 7, pp. 105-22, 1970
- [38] S. Holm and J.M. Hovem, Estimation of scalar ocean wave spectra by the maximum entropy method," IEEE J. Ocean. Eng., vol. OE-4, pp. 76-83, July 1979.
- [39] C.S. Hacker, "Autoregressive and transfer function models of mosquito populations," in Time Series and Ecological Processes, H.M. Shugat, Ed. Boulder, CO: Siam, 1978, pp. 294-303.
- [40] W.F. Gabriel, "Spectral analysis and adaptive array superresolution techniques," Proc. IEEE, vol. 68, pp. 654-666, June 1980.
- [41] ^{J. P.} ~~J.P.~~ Burg, "The relationship between maximum entropy and maximum likelihood spectra," Geophys., vol. 37, pp. 375-376, Apr. 1972.
- [42] A.H. Gray, Jr. and D.Y. Wong, "The Burg algorithm for LPC speech analysis/synthesis," IEEE Trans. Acoustics, Speech, Signal Process., vol. ASSP-28, pp. 609-615, Dec. 1980.

l

- [43] H. Kobatake et al., "Linear predictive coding of speech signals in a high ambient noise environment," in Conf. rec. 1978 ICASSP. pp. 472-475.

APPENDIX A

```

0001 *****
0002 *
0003 * THE DTMF RECEIVER BASED ON LINEAR PREDICTION
0004 *
0005 *****
0006 IDT 'DTMFRLP'
0007 *
0008 * ALL VARIABLES USED DECLARED
0009 *
0010 *
0011 0000 CNTRL AORG 0
0012 0001 CLCK AORG 1
0013 0002 MASK AORG 2
0014 005A RS AORG 90
0015 005B ASTART AORG 91
0016 005C SIGPTR AORG 92
0017 005D K AORG 93
0018 005E K1 AORG 94
0019 005F T AORG 95
0020 0060 NUMERA AORG 96
0021 0061 DENOM AORG 97
0022 0062 QUOT AORG 98
0023 0063 T1 AORG 99
0024 0064 T2 AORG 100
0025 0065 FIFTY AORG 101
0026 0066 TWONE AORG 102
0027 0067 REG AORG 103
0028 0068 KE AORG 104
0029 0069 KL AORG 105
0030 006A KR AORG 106
0031 006B KK AORG 107
0032 006C ACOUNT AORG 108 * ACOUNT CONTAINS THE MODEL ORDER-1
0033 006D P1 AORG 109 * I.E. IT CONTAINS (P-1)
0034 006E P2 AORG 110
0035 0070 TBLPT AORG 112
0036 0071 FREQ AORG 113
0037 0072 CCOUNT AORG 114
0038 007D ONE AORG 125
0039 0074 PRSUM AORG 116
0040 0075 IRSUM AORG 117
0041 0076 SINST AORG 118
0042 007E TEMP1 AORG 126
0043 007F TEMP2 AORG 127
0044 0079 DENOM1 AORG 121
0045 007A A2048 AORG 122
0046 007B TEMSGN AORG 123
0047 007C T3 AORG 124
0048 0073 COUNT AORG 115
0049 0077 ATEMP AORG 119
0050 0042 LGMIN AORG 66 * CONTAINS THE MINIMUM IN THE LOW GROUP
0051 0043 HGMIN AORG 67 * CONTAINS THE MINIMUM IN THE HIGH GROUP
0052 0045 LGADD AORG 69 * POINTS TO THE ADDRESS OF THE LGMIN
0053 0046 HGADD AORG 70 * POINTS TO THE ADDRESS OF THE HGMIN
0054 0047 NTHRL AORG 71 * CONTAINS NOISE THRESHOLD FOR LOW GROU
0055 0048 NTHRH AORG 72 * CONTAINS NOISE THRESHOLD FOR HIGH GRO
0056 0049 SEPRH AORG 73 * CONTAINS SEPERATION THRESHOLD FOR
0057 * THE HIGH GROUP POINTS

```

```

0058          *
0059 004A     DIGPT   AORG 74      * REGISTER USED TO DETERMINE TIME THE
0060 004B     DIGAB   AORG 75      * PRESENT(DIGPT)/ ABSENT(DIGAB)
0061 004C     DIGTB   AORG 76      * POINTS TO START OF TABLE CONTAINING
0062 0051     L1      AORG 81
0063 0052     L2      AORG 82
0064 0053     L3      AORG 83
0065 0054     L4      AORG 84
0066 0055     H1      AORG 85
0067 0056     H2      AORG 86
0068 0057     H3      AORG 87
0069 0058     H4      AORG 88
0070 0033     S1      AORG 51
0071 0040     S14     AORG 64
0072          *
0073          0065 VDPR   EQU 101   * VALID DIGIT PRESENT COUNTER
0074          0066 CD1   EQU 102   * CURRENT DIGIT FOR 1ST TIME
0075          0067 CD2   EQU 103   * CURRENT DIGIT FOR 2ND TIME
0076          0068 CD3   EQU 104   * HOLDS THE VALID DIGIT ( ACTUALLY
0077          *                               THE NO. OF PULSES TO BE OUTPUT ).
0078          0069 PREST EQU 105   * CONTAINS 2 WHEN DIGIT IS CONTINUOUSLY
0079          *                               PRESENT FOR 2 BLOCKS.
0080          006A VDAB   EQU 106   *DIGIT ABSENT COUNTER
0081          006B AD1   EQU 107   * CURRENT DIGIT ABSENT FOR 1ST BLOCK
0082          006C AD2   EQU 108   * CURRENT DIGIT ABSENT FOR 2ND BLOCK
0083          006E DIGIT EQU 110
0084          *
0085          007E TEMP1 EQU 126
0086          007F TEMP2 EQU 127
0087          007D ONE   EQU 125
0088          0045 LGADD EQU 69
0089          0046 HGADD EQU 70
0090          0063 T1    EQU 99
0091          004C DIGTB EQU 76
0092          0051 L1    EQU 81
0093          0052 L2    EQU 82
0094          0053 L3    EQU 83
0095          0054 L4    EQU 84
0096          0055 H1    EQU 85
0097          0056 H2    EQU 86
0098          0057 H3    EQU 87
0099          0058 H4    EQU 88
0100          0042 LGMIN EQU 66
0101          0043 HGMIN EQU 67
0102          0047 KLM   EQU 71
0103          0048 KH    EQU 72
0104          0049 KS1   EQU 73
0105          004A KS14  EQU 74
0106          007A A2048 EQU 122
0107          *
0108          * SHIFT RIGHT 11 MACRO
0109          *
0110          SR11      $MACRO
0111          SACL P1
0112          SACH P2
0113          ZAC
0114          LAC P1,5      *}

```

```

0115          SACH P1          *}SHIFT P1 LEFT 11 BITS
0116          LACK 31
0117          AND P1
0118          SACL P1
0119          ZAC
0120          LAC P2,5
0121          OR P1
0122          $END
0123          *
0124          * MACRO TO COMPARE A TO B
0125          *
0126          CMP $MACRO A,B
0127          LAC :A:,0
0128          SUB :B:,0
0129          $END
0130          *
0131 0000          PSEG
0132          OPTION XREF
0133          *
0134          *
0135          *
0136          *
0137          *
0138          *
0139          *
0140          * THE FOLLOWING DATA ARE COSINE COEFFECEINTS AND ARE STORED
0141          *
0142          * IN THE PRAM STARTING FROM ADDRESS 3200.
0143 0C80          AORG 3200
0144 0C80 06D4          DATA >06D4
0145 0C81 03AA          DATA >03AA
0146 0C82 FF6E          DATA >FF6E
0147 0C83 FB5C          DATA >FB5C
0148 0C84 F8A6          DATA >F8A6
0149 0C85 F815          DATA >F815
0150 0C86 F9D5          DATA >F9D5
0151 0C87 FD63          DATA >FD63
0152 0C88 01B3          DATA >01B3
0153 0C89 0585          DATA >0585
0154 0C8A 07BA          DATA >07BA
0155 0C8B 07AC          DATA >07AC
0156 0C8C 0561          DATA >0561
0157 0C8D 0182          DATA >0182
0158 0C8E 0694          DATA >0694
0159 0C8F 02D3          DATA >02D3
0160 0C90 FE13          DATA >FE13
0161 0C91 FA00          DATA >FA00
0162 0C92 F80F          DATA >F80F
0163 0C93 F8EF          DATA >F8EF
0164 0C94 FC50          DATA >FC50
0165 0C95 0100          DATA >0100
0166 0C96 0556          DATA >0556
0167 0C97 07C7          DATA >07C7
0168 0C98 0776          DATA >0776
0169 0C99 047F          DATA >047F
0170 0C9A FFF0          DATA >FFF0
0171 0C9B FB67          DATA >FB67

```

0172	0C9C	0646	DATA	>0646
0173	0C9D	01D7	DATA	>01D7
0174	0C9E	FC9E	DATA	>FC9E
0175	0C9F	F8DA	DATA	>F8DA
0176	0CA0	F82B	DATA	>F82B
0177	0CA1	FADD	DATA	>FADD
0178	0CA2	FFC7	DATA	>FFC7
0179	0CA3	04C8	DATA	>04C8
0180	0CA4	07BA	DATA	>07BA
0181	0CA5	0757	DATA	>0757
0182	0CA6	03C9	DATA	>03C9
0183	0CA7	FE9A	DATA	>FE9A
0184	0CA8	FA05	DATA	>FA05
0185	0CA9	F804	DATA	>F804
0186	0CAA	05E9	DATA	>05E9
0187	0CAB	00BD	DATA	>00BD
0188	0CAC	FB2F	DATA	>FB2F
0189	0CAD	F824	DATA	>F824
0190	0CAE	F933	DATA	>F933
0191	0CAF	FDCE	DATA	>FDCE
0192	0CB0	038F	DATA	>038F
0193	0CB1	0774	DATA	>0774
0194	0CB2	0776	DATA	>0776
0195	0CB3	0393	DATA	>0393
0196	0CB4	FDD3	DATA	>FDD3
0197	0CB5	F935	DATA	>F935
0198	0CB6	F823	DATA	>F823
0199	0CB7	FB2B	DATA	>FB2B
0200	0CB8	04A8	DATA	>04A8
0201	0CB9	FD6C	DATA	>FD6C
0202	0CBA	F858	DATA	>F858
0203	0CBB	F9AA	DATA	>F9AA
0204	0CBC	0048	DATA	>0048
0205	0CBD	06AA	DATA	>06AA
0206	0CBE	077A	DATA	>077A
0207	0CBF	0209	DATA	>0209
0208	0CC0	FAE6	DATA	>FAE6
0209	0CC1	F806	DATA	>F806
0210	0CC2	FBD1	DATA	>FBD1
0211	0CC3	031B	DATA	>031B
0212	0CC4	07CD	DATA	>07CD
0213	0CC5	05F9	DATA	>05F9
0214	0CC6	03FC	DATA	>03FC
0215	0CC7	FBF9	DATA	>FBF9
0216	0CC8	F801	DATA	>F801
0217	0CC9	FC0F	DATA	>FC0F
0218	0CCA	0412	DATA	>0412
0219	0CCB	07FF	DATA	>07FF
0220	0CCC	03E5	DATA	>03E5
0221	0CCD	FBE3	DATA	>FBE3
0222	0CCE	F801	DATA	>F801
0223	0CCF	FC26	DATA	>FC26
0224	0CD0	0428	DATA	>0428
0225	0CD1	07FF	DATA	>07FF
0226	0CD2	03CF	DATA	>03CF
0227	0CD3	FBCD	DATA	>FBCD
0228	0CD4	0331	DATA	>0331

```
-0229 0CD5 FA8E      DATA >FA8E
0230 0CD6 F875      DATA >F875
0231 0CD7 FF6D      DATA >FF6D
0232 0CD8 0715      DATA >0715
0233 0CD9 063C      DATA >063C
0234 0CDA FDE6      DATA >FDE6
0235 0CDB F816      DATA >F816
0236 0CDC FBC9      DATA >FBC9
0237 0CDD 048C      DATA >048C
0238 0CDE 07D9      DATA >07D9
0239 0CDF 01B8      DATA >01B8
0240 0CE0 F987      DATA >F987
0241 0CE1 F91C      DATA >F91C
0242 0CE2 0246      DATA >0246
0243 0CE3 F94B      DATA >F94B
0244 0CE4 F9EA      DATA >F9EA
0245 0CE5 033F      DATA >033F
0246 0CE6 07EE      DATA >07EE
0247 0CE7 0143      DATA >0143
0248 0CE8 F8CA      DATA >F8CA
0249 0CE9 FAA3      DATA >FAA3
0250 0CEA 0429      DATA >0429
0251 0CEB 07BB      DATA >07BB
0252 0CEC 003B      DATA >003B
0253 0CED F867      DATA >F867
0254 0CEE FB73      DATA >FB73
0255 0CEF 0502      DATA >0502
0256
0257
0258
0259
0260
0261 0D48          AORG 3400
0262 0D48 0429      DATA >0429
0263 0D49 071C      DATA >071C
0264 0D4A 07FA      DATA >07FA
0265 0D4B 0684      DATA >0684
0266 0D4C 0325      DATA >0325
0267 0D4D FEDD      DATA >FEDD
0268 0D4E FAE8      DATA >FAE8
0269 0D4F F871      DATA >F871
0270 0D50 F82F      DATA >F82F
0271 0D51 FA37      DATA >FA37
0272 0D52 FDEF      DATA >FDEF
0273 0D53 0241      DATA >0241
0274 0D54 05EB      DATA >05EB
0275 0D55 07DB      DATA >07DB
0276 0D56 048C      DATA >048C
0277 0D57 077B      DATA >077B
0278 0D58 07C3      DATA >07C3
0279 0D59 054A      DATA >054A
0280 0D5A 00F0      DATA >00F0
0281 0D5B FC42      DATA >FC42
0282 0D5C F8E7      DATA >F8E7
0283 0D5D F811      DATA >F811
0284 0D5E FA0B      DATA >FA0B
0285 0D5F FE22      DATA >FE22
```

```
*
*
*
*
*
```

WRITE THE 112 SINE COEFFECIENTS INTO PROM STARTING FROM 34

0286	0D60	02E2	DATA	>02E2
0287	0D61	069D	DATA	>069D
0288	0D62	07FF	DATA	>07FF
0289	0D63	068B	DATA	>068B
0290	0D64	04F6	DATA	>04F6
0291	0D65	07C8	DATA	>07C8
0292	0D66	073F	DATA	>073F
0293	0D67	0396	DATA	>0396
0294	0D68	FE61	DATA	>FE61
0295	0D69	F9DF	DATA	>F9DF
0296	0D6A	F801	DATA	>F801
0297	0D6B	F997	DATA	>F997
0298	0D6C	FDF1	DATA	>FDF1
0299	0D6D	032D	DATA	>032D
0300	0D6E	070B	DATA	>070B
0301	0D6F	07E0	DATA	>07E0
0302	0D70	054F	DATA	>054F
0303	0D71	0073	DATA	>0073
0304	0D72	0563	DATA	>0563
0305	0D73	07F7	DATA	>07F7
0306	0D74	0662	DATA	>0662
0307	0D75	0179	DATA	>0179
0308	0D76	FBCC	DATA	>FBCC
0309	0D77	F84F	DATA	>F84F
0310	0D78	F8D6	DATA	>F8D6
0311	0D79	FD1B	DATA	>FD1B
0312	0D7A	02E1	DATA	>02E1
0313	0D7B	0728	DATA	>0728
0314	0D7C	07B2	DATA	>07B2
0315	0D7D	0439	DATA	>0439
0316	0D7E	FE8C	DATA	>FE8C
0317	0D7F	F9A1	DATA	>F9A1
0318	0D80	0681	DATA	>0681
0319	0D81	0792	DATA	>0792
0320	0D82	024F	DATA	>024F
0321	0D83	FB1E	DATA	>FB1E
0322	0D84	F802	DATA	>F802
0323	0D85	FB94	DATA	>FB94
0324	0D86	02D8	DATA	>02D8
0325	0D87	07BC	DATA	>07BC
0326	0D88	0628	DATA	>0628
0327	0D89	FF70	DATA	>FF70
0328	0D8A	F92F	DATA	>F92F
0329	0D8B	F8A1	DATA	>F8A1
0330	0D8C	FE3D	DATA	>FE3D
0331	0D8D	0551	DATA	>0551
0332	0D8E	06EF	DATA	>06EF
0333	0D8F	06E9	DATA	>06E9
0334	0D90	FFF4	DATA	>FFF4
0335	0D91	F90A	DATA	>F90A
0336	0D92	F91E	DATA	>F91E
0337	0D93	0019	DATA	>0019
0338	0D94	06FC	DATA	>06FC
0339	0D95	06DC	DATA	>06DC
0340	0D96	FFDA	DATA	>FFDA
0341	0D97	F8FE	DATA	>F8FE
0342	0D98	F92B	DATA	>F92B

```

0343 0D99 0033      DATA >0033
0344 0D9A 0708      DATA >0708
0345 0D9B 06CE      DATA >06CE
0346 0D9C 0755      DATA >0755
0347 0D9D 05DB      DATA >05DB
0348 0D9E FD58      DATA >FD58
0349 0D9F F806      DATA >F806
0350 0DA0 FC49      DATA >FC49
0351 0DA1 0502      DATA >0502
0352 0DA2 07B7      DATA >07B7
0353 0DA3 0126      DATA >0126
0354 0DA4 F934      DATA >F934
0355 0DA5 F96C      DATA >F96C
0356 0DA6 018A      DATA >018A
0357 0DA7 07D0      DATA >07D0
0358 0DA8 04B2      DATA >04B2
0359 0DA9 FBF1      DATA >FBF1
0360 0DAA 07AB      DATA >07AB
0361 0DAB 045C      DATA >045C
0362 0DAC FAD0      DATA >FAD0
0363 0DAD F8B1      DATA >F8B1
0364 0DAE 0108      DATA >0108
0365 0DAF 07E6      DATA >07E6
0366 0DB0 0375      DATA >0375
0367 0DB1 FA11      DATA >FA11
0368 0DB2 F92C      DATA >F92C
0369 0DB3 020C      DATA >020C
0370 0DB4 07FF      DATA >07FF
0371 0DB5 027F      DATA >027F
0372 0DB6 F96D      DATA >F96D
0373 0DB7 F9C4      DATA >F9C4
0374
0375
0376
0377
0378
*****
*
*                               INITIALIZATION
*
*****
0379 0640      AORG 1600
0380 0640 0641  DATA >641      * INITIALIZE DATA POINTER AT ROM LOC.160
0381
*                               NEXT SET OF DATA IS STORED FROM ROM LOC
0382 07D0      AORG 2000
0383 07D0 000A  DATA >000A      * INITIALIZE ROM. LOC +2000= S1 = +10
0384
*
*
0385
* INITIALIZE THE DIGIT TABLE. DIGIT TABLE POINTER = 1344
0386
*
0387
* THIS TABLE ENTRIES GIVES THE # OF PULSES TO BE OUTPUT
0388
*
0389
* CORRESPONDING TO EACH DIGIT.
0390
*
0391
*
0392
0393 0546      AORG 1350
0394 0546 0000  DATA >0000
0395 0547      AORG 1351
0396 0547 0001  DATA >0001
0397 0548      AORG 1352
0398 0548 0002  DATA >0002
0399 0549      AORG 1353

```

```

0400 0549 000A      DATA >000A
0401 054E          AORG 1358
0402 054E 0003      DATA >0003
0403 0550          AORG 1360
0404 0550 0004      DATA >0004
0405 0552          AORG 1362
0406 0552 0005      DATA >0005
0407 0554          AORG 1364
0408 0554 000B      DATA >000B
0409 0560          AORG 1376
0410 0560 0006      DATA >0006
0411 0564          AORG 1380
0412 0564 0007      DATA >0007
0413 0568          AORG 1384
0414 0568 0008      DATA >0008
0415 056C          AORG 1388
0416 056C 000C      DATA >000C
0417 0588          AORG 1416
0418 0588 000E      DATA >000E
0419 0590          AORG 1424
0420 0590 0009      DATA >0009
0421 0598          AORG 1432
0422 0598 000F      DATA >000F
0423 05A0          AORG 1440
0424 05A0 000D      DATA >000D
0425                *****
0426                *
0427                *
0428                *
0429                *****
0430 0002          AORG 2
0431 0002 F900      B INTPC          * BRANCH VECTOR FOR INTERRUPT
0432 0003 048E      0003 048E
0432 0004 F900      B YB20
0432 0005 0008
0433                *
0434                *
0435                *
0436                *
0437 0008          AORG 8
0438 0008 7F82      YB20  EINT
0439 0009 7F8B      BEGIN SOVM  * SET TMS 320 IN THE OVERFLOW MODE
0440                *
0441                *
0442                *
0443                *
0444                *
0445 000A 6E01      LDPK 1
0446 000B 700A      LARK 0,10
0447 000C 3000      SAR 0,CNTRL  *
0448                *
0449 000D 7014      LARK 0,20
0450 000E 3001      SAR 0,CLCK
0451 000F 6A01      LT CLCK
0452 0010 8015      MPYK 21
0453 0011 7F89      ZAC
0454 0012 7F8F      APAC

```

```

0455 0013 5001 *      SACL CLCK,0 *
0456 *
0457 0014 6E00      LDPK 0
0458 *
0459 * NOW START GETTING THE DATA SAMPLES
0460 *
0461 0015 717F      LARK 1,127
0462 0016 7000      LARK 0,0
0463 0017 6E01      LDPK 1
0464 0018 4901      OUT CLCK,PA1 * SAMPLING FREQ.= 8 KHZ.
0465 0019 4800      OUT CNTRL,PA0 *A/D IS SET TO CONTINUOUS MODE
0466 *
0467 001A 7F82 LOOP  EINT * WAIT FOR AN INPUT WORD
0468 001B 7F89      ZAC
0469 001C FF00 HERE  BZ HERE * HERE YOU TEST WHETHER THE ACC = 0. IF YO
      001D 001C
0470 * HAVE JUST RETURNED FROM SERVICING THE INTERRUPT ACC=11. SO
0471 * BRANCH TO LOOP WHERE YOU ENABLE THE INTERRUPT; OTHERWISE RE
0472 * JUST TO 'HERE' WOULD NOT ENABLE THE INTERRUPT WHICH HAD BE
0473 * DISABLED IN THE 'GET' SERVICE ROUTINE.
0474 001E 6881      LARP 1
0475 001F F400      BANZ LOOP * TO COLLECT 127 DATA SAMPLES
      0020 001A
0476 0021 F900      B YY11
      0022 0028
0477 *
0478 0023 7F81 GET  DINT
0479 0024 6880      LARP 0
0480 0025 42A8      IN *,2
0481 0026 7E0B      LACK 11
0482 0027 7F8D      RET
0483 *
0484 0028 7F82 YY11 EINT * ENABLE INTERRUPT AFTER COLLECTING
0485 0029 7EFA      LACK 250 * 128 DATA SAMPLES.
0486 002A 507E      SACL TEMP1 *
0487 002B 237E      LAC TEMP1,3 * SET UP ACC = ADDRESS 2000
0488 002C 706F      LARK 0,111 *
0489 002D 307E      SAR 0,TEMP1 *
0490 002E 7D7E      TBLW TEMP1 * SET ROM LOC. 2000 = 111
0491 *
0492 **
0493 *
0494 * SET UP A/D AFTER TBLW
0495 *
0496 *
0497 002F 6E00      LDPK 0
0498 0030 700A      LARK 0,10
0499 0031 307E      SAR 0,TEMP1
0500 0032 7014      LARK 0,20
0501 0033 307F      SAR 0,TEMP2
0502 0034 6A7F      LT TEMP2
0503 0035 8015      MPYK 21
0504 0036 7F89      ZAC
0505 0037 7F8F      APAC
0506 0038 507F      SACL TEMP2 *
0507 *
0508 0039 497F      OUT TEMP2,PA1 * SET SAMPLING FREQ. = 8 KHZ.

```

```

0509 003A 487E      OUT TEMP1,PA0 * SET A/D IN CONTINUOUS MODE
0510                *
0511 003B          *
0512                *
0513 003B 6E00      LDPK 0
0514 003C 701E      LARK 0,30
0515 003D 3002      SAR 0,MASK
0516 003E 6A02      LT MASK
0517 003F 7F89      ZAC
0518 0040 8888      MPYK 2184
0519 0041 7F8F      APAC
0520 0042 5002      SACL MASK,0 * MASK = >FFF0 USED TO INVERT
0521                *
0522                *
0523 0043 7100      LARK 0,127
0524 0044 6881      LARK 1,0
0525 0045 6E01      INVERT LARP 1 *
0526 0046 6602      LDPK 1 *
0527 0047 7888      ZALS MASK *
0528 0048 50A8      XOR *
0529 0049 6880      SACL *+
0530 004A F400      LARP 0
0531 004B 0044      BANZ INVERT
0531 004C 6E00      LDPK 0
0532                *
0533                * THE DATA SAMPLES ARE NOW IN THE DATA RAM FROM
0534                * ADDRESS 0 TO 127.
0535                *
0536 004D 717F      LARK 1,127
0537 004E 7000      LARK 0,0
0538 004F 6880      CST LARP 0
0539 0050 2C88      LAC *,12 *}SHIFT RIGHT BY 4 BITS
0540 0051 58A8      SACH *+,0 *}
0541 0052 6881      LARP 1
0542 0053 F400      BANZ CST
0542 0054 004F
0543                *
0544                *
0545                *
0546                *
0547                * THE FOLLOWING BLOCK CALCULATES R(0),R(1),....R(14) AND *
0548                *
0549                *
0550                *
0551                *
0552                * INITIALIZE T=128,ARO=127,AR1=127-(T-128)
0553                *
0554                *
0555 0055 6E01      LDPK 1 *} T(RAM LOCATION 143)
0556 0056 7180      LARK 1,128 *} IS SET TO 128
0557 0057 310F      SAR 1,15 *}
0558                *
0559                *
0560 0058 380F      RT LAR 0,15 *STORE T
0561 0059 7180      LARK 1,128 *
0562 005A 200F      LAC 15,0 *}
0563 005B 310F      SAR 1,15 *} T-128

```

```

0564 005C 100F      SUB 15,0      *}
0565 005D 500F      SACL 15,0    * T-128 GOES INTO 143
0566 005E 7E7F      LACK 127     *}
0567 005F 100F      SUB 15,0    *} 127-(T-128)
0568 0060 500F      SACL 15,0    *]
0569 0061 390F      LAR 1,15    *] AR1=127-(T-128)
0570 0062 300F      SAR 0,15    *RESTORE T
0571 0063 707F      LARK 0,127  * AR0=127
0572                *
0573                *
0574                * LOOP BEGINS
0575                * ZERO THE ACC AND THE P REGISTERS BEFORE ENTERING THE LOOP
0576 0064 6881      LARP 1
0577 0065 7F89      ZAC
0578 0066 8000      MPYK 0      * PREG=0
0579 0067 6A80      MT         LT *,0
0580 0068 6D91      MPY *-,1
0581 0069 7F8F      APAC
0582 006A F400      BANZ MT
0583                *
0584                *
0585                * SCALE R(T) AND STORE
0586                *
0587 006C 6880      LARP 0
0588 006D 6E01      LDPK 1
0589 006E 380F      LAR 0,15
0590 006F 5888      SACH *,0
0591 0070 2E88      LAC *,14
0592 0071 5888      SACH *,0
0593 0072 68A8      MAR **
0594 0073 300F      SAR 0,15
0595 0074 7F89      ZAC
0596 0075 7E8F      LACK 143
0597 0076 100F      SUB 15
0598 0077 FD00      BGEZ RT
0599                *
0600                *
0601                *****
0602                *
0603                * THE FOLLOWING BLOCK CALCULATES THE A PARAMETERS *
0604                *
0605                *****
0606                *
0607 0079 6E00      LDPK 0
0608                * INITIALIZING THE VARIOUS POINTERS
0609 007A 7080      LARK 0,128
0610 007B 307A      SAR 0,A2048
0611 007C 247A      LAC A2048,4 *}
0612 007D 507A      SACL A2048 *} A2048=2048
0613 007E 7001      LARK 0,1    *]
0614 007F 305A      SAR 0,RS    *] RS=1
0615 0080 7015      LARK 0,21   *}
0616 0081 305B      SAR 0,ASTART *]ASTART=21
0617 0082 7033      LARK 0,51   *]
0618 0083 305C      SAR 0,SIGPTR *] SIGPTR=51

```

```

0619 0084 7032      LARK 0,50      *
0620 0085 3065      SAR 0,FIFTY   * FIFTY=50
0621 0086 7014      LARK 0,20
0622 0087 3066      SAR 0,TWONE   *TWONE=20
0623 0088 700D      LARK 0,13
0624 0089 306C      SAR 0,ACOUNT
0625 008A 7001      LARK 0,1      *
0626 008B 307D      SAR 0,ONE     * ONE = 1
0627
0628      *CALCULATION OF All AND SIGMAL BEGINS
0629      *
0630      *
0631 008C 6E01      LDPK 1
0632 008D 3900      LAR 1,0       *FETCH R(0)
0633 008E 3801      LAR 0,1       *FETCH R(1)
0634 008F 6E00      LDPK 0
0635 0090 3161      SAR 1,DENOM
0636 0091 3060      SAR 0,NUMERA
0637 0092 7F81      DINT
0638 0093 F800      CALL DIVIDE   *All=R(1)/R(0)
           0094 04F9
0639 0095 7F82      EINT
0640 0096 2C62      LAC QUOT,12   *]
0641 0097 5862      SACH QUOT,0   *] SHIFT QUOT RIGHT BY 4 TO PUT
0642      * All IN 2048 SCALE
0643 0098 7F89      ZAC           *]
0644 0099 1062      SUB QUOT      *} NEGATE QUOT
0645 009A 5015      SACL 21      *STORE All IN RAMLOC 21
0646
0647      *
0648      * SIGNAL CALCULATION
0649      *
0650 009B 2B7A      LAC A2048,11 *]
0651 009C 6A15      LT 21        *}(2048*2048 - (All**2))
0652 009D 6D15      MPY 21      *]
0653 009E 7F90      SPAC        *}
0654      SR11
0001 009F 506D      SACL P1
0002 00A0 586E      SACH P2
0003 00A1 7F89      ZAC
0004 00A2 256D      LAC P1,5    *]
0005 00A3 586D      SACH P1     *}SHIFT P1 LEFT 11 BITS
0006 00A4 7E1F      LACK 31
0007 00A5 796D      AND P1
0008 00A6 506D      SACL P1
0009 00A7 7F89      ZAC
0010 00A8 256E      LAC P2,5
0011 00A9 7A6D      OR P1
0655 00AA 5033      SACL 51,0   *STORE AT RAMLOC 51
0656 00AB 7F89      ZAC
0657 00AC 6A33      LT 51       *]
0658 00AD 6E01      LDPK 1      *
0659 00AE 6D00      MPY 0       *}((2048**2 - All**2)/2048)*R(0)=SIGM
0660      *
0661 00AF 7F8F      APAC
0662 00B0 6E00      LDPK 0
0663      SR11

```

```

0001 00B1 506D   SACL P1
0002 00B2 586E   SACH P2
0003 00B3 7F89   ZAC
0004 00B4 256D   LAC P1,5      *}
0005 00B5 586D   SACH P1      *}SHIFT P1 LEFT 11 BITS
0006 00B6 7E1F   LACK 31
0007 00B7 796D   AND P1
0008 00B8 506D   SACL P1
0009 00B9 7F89   ZAC
0010 00BA 256E   LAC P2,5
0011 00BB 7A6D   OR P1
0664 00BC 5033   SACL 51,0      *STORE SIGNAL AT RAMLOC 51
0665
0666
0667
0668
0669 00BD 700D   LARK 0,13
0670 00BE 6880   LARP 0
0671 00BF 6E01   LDPK 1
0672 00C0 200E   LAC 14,0
0673 00C1 5098   SACL *-,0
0674 00C2 200D   LAC 13,0
0675 00C3 5098   SACL *-,0
0676 00C4 200C   LAC 12,0
0677 00C5 5098   SACL *-,0
0678 00C6 200B   LAC 11,0
0679 00C7 5098   SACL *-,0
0680 00C8 200A   LAC 10,0
0681 00C9 5098   SACL *-,0
0682 00CA 2009   LAC 9,0
0683 00CB 5098   SACL *-,0
0684 00CC 2008   LAC 8,0
0685 00CD 5098   SACL *-,0
0686 00CE 2007   LAC 7,0
0687 00CF 5098   SACL *-,0
0688 00D0 2006   LAC 6,0
0689 00D1 5098   SACL *-,0
0690 00D2 2005   LAC 5,0
0691 00D3 5098   SACL *-,0
0692 00D4 2004   LAC 4,0
0693 00D5 5098   SACL *-,0
0694 00D6 2003   LAC 3,0
0695 00D7 5098   SACL *-,0
0696 00D8 2002   LAC 2,0
0697 00D9 5098   SACL *-,0
0698 00DA 2001   LAC 1,0
0699 00DB 5098   SACL *-,0
0700 00DC 6E00   LDPK 0
0701
0702
0703
0704
0705
0706 00DD 395A   AKK LAR 1,RS
0707 00DE 6881   LARP 1
0708 00DF 2B98   LAC *-,11     *LOAD ACC WITH R(K)*2048
0709 00E0 385B   LAR 0,ASTART * ARO POINTS TO RAMLOC 21

```

```

0710 00E1 8000          MPYK 0          * PREG =0
0711 00E2 6880      GT  LARP 0
0712 00E3 6AA1          LT ** ,1
0713 00E4 6D88          MPY *          *WHEN LOOP EXITED,ARO POINTS TO AKK
0714 00E5 7F8F          APAC
0715 00E6 F400          BANZ GT          * R(K)+{A(K-1,L)*R(K-L) FORMED
          00E7 00E2          *
          *          AFTER EXITING THE LOOP
0716
0717          SR11
0001 00E8 506D          SACL P1
0002 00E9 586E          SACH P2
0003 00EA 7F89          ZAC
0004 00EB 256D          LAC P1,5          *}
0005 00EC 586D          SACH P1          *}SHIFT P1 LEFT 11 BITS
0006 00ED 7E1F          LACK 31
0007 00EE 796D          AND P1
0008 00EF 506D          SACL P1
0009 00F0 7F89          ZAC
0010 00F1 256E          LAC P2,5
0011 00F2 7A6D          OR P1
0718 00F3 5060          SACL NUMERA,0
0719 00F4 395C          LAR 1,SIGPTR      *FETCH ADDRESS OF SIGMA(K-1)
0720 00F5 6881          LARP 1
0721 00F6 20A8          LAC **          *LOAD ACC WITH SIGMA(K-1)
          *          AND INC SIGMAPOINTER
0722
0723 00F7 5061          SACL DENOM,0
0724 00F8 305F          SAR 0,T
0725 00F9 7F81          DINT
0726 00FA F800          CALL DIVIDE
          00FB 04F9
0727 00FC 7F82          EINT
0728 00FD 2C62          LAC QUOT,12      * SHIFT QUOT 4 RIGHT
0729 00FE 5862          SACH QUOT,0
0730 00FF 7F89          ZAC
0731 0100 1062          SUB QUOT
0732 0101 5062          SACL QUOT
0733 0102 385F          LAR 0,T
0734 0103 2062          LAC QUOT          *}
0735 0104 6880          LARP 0          *} STORE AKK
0736 0105 5088          SACL *          *}
0737
0738 0106 3063          SAR 0,T1          *T1 NOW CONTAINS K
0739 0107 6800          MAR -            *DEC T1
0740 0108 3064          SAR 0,T2          *STORE IN T2
0741 0109 385B          LAR 0,ASTART     *}K1=21
0742 010A 305E          SAR 0,K1          *}
0743
0744
0745          *CALCULATION OF A(K,I) BEGINS
0746
0747
0748 010B 385F          LAR 0,T
0749 010C 306B          SAR 0,KK
0750 010D 6898          MAR *-
0751 010E 3069          SAR 0,KL
0752 010F 2069          LAC KL,0
0753 0110 0065          ADD FIFTY

```

```

0754 0111 5068          SACL KE
0755 0112 385B          LAR 0,ASTART
0756 0113 306A          SAR 0,KR
0757                    *
0758                    *
0759 0114 6880          LARP 0
0760 0115 386B          EXTEND LAR 0,KK
0761 0116 6A88          LT *                * AKK LOADED IN T REG
0762 0117 386A          LAR 0,KR
0763 0118 7F89          ZAC
0764 0119 6DA8          MPY **            *]AKK*AKI. INC ARO
0765 011A 7F8F          APAC
0766 011B 507E          SACL TEMP1        * SAVE ACCL
0767 011C 587F          SACH TEMP2        * SAVE ACCH
0768 011D 306A          SAR 0,KR
0769 011E 7F89          ZAC
0770 011F 3869          LAR 0,KL          *
0771 0120 2B98          LAC *- ,11        *}AK1,I *2048 LOADED IN ACC
0772 0121 3069          SAR 0,KL          * RESTORE UPDATED KL
0773 0122 607F          ADDH TEMP2        *AK1,I*2048 +AKK*AK1,K2
0774 0123 617E          ADDS TEMP1
0775                    SR11
0001 0124 506D          SACL P1
0002 0125 586E          SACH P2
0003 0126 7F89          ZAC
0004 0127 256D          LAC P1,5          *}
0005 0128 586D          SACH P1          *}SHIFT P1 LEFT 11 BITS
0006 0129 7E1F          LACK 31
0007 012A 796D          AND P1
0008 012B 506D          SACL P1
0009 012C 7F89          ZAC
0010 012D 256E          LAC P2,5
0011 012E 7A6D          OR P1
0776 012F 3868          LAR 0,KE          *
0777 0130 5098          SACL *-          *
0778 0131 3068          SAR 0,KE          *
0779                    *
0780                    *
0781                    *HAVE WE REACHED END OF CALCULATING AK,I
0782                    *
0783                    *
0784 0132 2069          LAC KL
0785 0133 1066          SUB TWONE
0786 0134 FE00          BNZ EXTEND
0135 0115
0787                    *
0788                    *TO SWAP THE ATEMP TABLE INTO ASTART TABLE
0789                    *
0790 0136 6880          LARP 0
0791 0137 386B          LAR 0,KK
0792 0138 6898          MAR *-
0793 0139 3067          SAR 0,REG
0794 013A 2067          LAC REG,0        * ARO POINTS TO ASTART
0795 013B 0065          ADD FIFTY
0796 013C 5067          SACL REG
0797 013D 3967          LAR 1,REG        *AR1 POINTS TO ATEMP
0798                    *

```

```

0799 013E 6881      LARP 1
0800 013F 2090      CON   LAC *-,0,0
0801 0140 5091      SACL *-,0,1
0802 0141 3067      SAR 0,REG
0803 0142 2067      LAC REG,0
0804 0143 1066      SUB TWONE
0805 0144 FE00      BNZ CON
      0145 013F

0806                *
0807                *
0808                * CALCULATION OF SIGMA K BEGINS
0809                *
0810 0146 2B7A      LAC A2048,11
0811 0147 6880      LARP 0
0812 0148 386B      LAR 0, KK
0813 0149 6A88      LT *
0814 014A 6D88      MPY *
0815 014B 7F90      SPAC
0816                SR11
0001 014C 506D      SACL P1
0002 014D 586E      SACH P2
0003 014E 7F89      ZAC
0004 014F 256D      LAC P1,5      *}
0005 0150 586D      SACH P1      *}SHIFT P1 LEFT 11 BITS
0006 0151 7E1F      LACK 31
0007 0152 796D      AND P1
0008 0153 506D      SACL P1
0009 0154 7F89      ZAC
0010 0155 256E      LAC P2,5
0011 0156 7A6D      OR P1
0817 0157 5067      SACL REG
0818 0158 6A67      LT REG
0819 0159 385C      LAR 0, SIGPTR
0820 015A 7F89      ZAC
0821 015B 6DA8      MPY *+      * INC SIGPTR
0822 015C 7F8F      APAC
0823 015D 305C      SAR 0, SIGPTR
0824                SR11
0001 015E 506D      SACL P1
0002 015F 586E      SACH P2
0003 0160 7F89      ZAC
0004 0161 256D      LAC P1,5      *}
0005 0162 586D      SACH P1      *}SHIFT P1 LEFT 11 BITS
0006 0163 7E1F      LACK 31
0007 0164 796D      AND P1
0008 0165 506D      SACL P1
0009 0166 7F89      ZAC
0010 0167 256E      LAC P2,5
0011 0168 7A6D      OR P1
0825 0169 385C      LAR 0, SIGPTR
0826 016A 6880      LARP 0
0827 016B 5088      SACL *      *
0828                *
0829                *
0830 016C 385A      LAR 0, RS      *
0831 016D 6880      LARP 0      *SET RS TO POINT TO NEXT SET OF
0832 016E 68A8      MAR *+      * A PARAMETERS

```

```

0833 016F 305A      SAR 0,RS      *
0834                *
0835                *
0836 0170 206C      LAC ACCOUNT
0837 0171 107D      SUB ONE
0838 0172 506C      SACL ACCOUNT
0839 0173 FE00      BNZ AKK      * BRANCH IF CALCULATION OF THE 14TH SET
      0174 00DD
0840                * OF A PARAMETERS IS NOT OVER TO CALCULATE
0841                * THE NEXT SET
0842                *
0843                * *****
0844                *
0845                * THE FOLLOWING BLOCK CALCULATES THE PSD FOR 8 FREQ PTS.
0846                * *****
0847                *
0848                *
0849                *
0850                *
0851                *
0852                *
0853                * INITIALIZATION
0854                *
0855                *
0856                *
0857 0175 7EFF      LACK 255
0858 0176 507E-     SACL TEMP1
0859 0177 277E      LAC TEMP1,7
0860 0178 7A7E      OR TEMP1
0861 0179 5064      SACL 2      * T2 = >7FFF = 32,767 = SATURATING VALUE
0862                *
0863                *
0864 017A 7001      LARK 0,1
0865 017B 307D      SAR 0,ONE
0866 017C 6A7D      LT ONE
0867 017D 8064      MPYK 100
0868 017E 7F8E      PAC
0869 017F 5070      SACL TBLPT,0  *TBLPT=(SO THAT 100*2**5 = 3200)
0870 0180 7000      LARK 0,0
0871                * LARK 0,150
0872                * SAR 0,TEMP1
0873                * LT TEMP1
0874                * ZAC
0875                * MPYK 8
0876                * APAC
0877                * SACL SINST,0
0878 0181 6A7D      LT ONE
0879 0182 8352      MPYK 850
0880 0183 7F8E      PAC
0881 0184 5076      SACL SINST,0
0882 0185 2276      LAC SINST,2
0883 0186 5076      SACL SINST,0  * SINST = 3400 = ADDRESS OF SINE TABLE
0884 0187 7001      LARK 0,1
0885 0188 307D      SAR 0,ONE
0886 0189 7020      LARK 0,32      *}
0887 018A 307E      SAR 0,TEMP1    *}
0888 018B 6A7E      LT TEMP1      *}

```

```

0889 018C 7F89      ZAC
0890 018D 8040      MPYK 64      * }SET A2048=2048
0891 018E 7F8F      APAC
0892 018F 507A      SACL A2048,0 *
0893 0190 7000      LARK 0,0
0894 0191 3072      SAR 0,CCOUNT * SET CCOUNT=0
0895
0896                * INITIALIZE P1,P2(DOUBLE WORD) TO 2048**2 +{(2048**2)*14}
0897                *
0898 0192 700D      LARK 0,13
0899 0193 6880      LARP 0
0900 0194 7F89      ZAC
0901 0195 8000      MPYK 0
0902 0196 6A7A      PQ1 LT A2048
0903 0197 6D7A      MPY A2048
0904 0198 7F8F      APAC
0905 0199 F400      BANZ PQ1
      019A 0196
0906 019B 6A7A      LT A2048
0907 019C 6D7A      MPY A2048
0908 019D 7F8F      APAC
0909 019E 586D      SACH P1      * P1= 03C0
0910 019F 506E      SACL P2      * P2= 0000
0911
0912 01A0 7007      LARK 0,7
0913 01A1 3071      SAR 0,FREQ   * SET FREQ=7 BECAUSE WE HAVE TO CALCULATE
0914                * THE PSD AT 8 POINTS
0915 01A2
0916
0917                *
0918                *ACTUAL CALCULATION BEGINS
0919                *
0920                *
0921                *
0922                * READ IN THE COSINE COEFFECIENTS
0923                *
0924 01A2 3071      PSD SAR 0,FREQ   * SAVE THE FREQPOINTER .THIS IS NEEDED AF
0925                * * JUMPING BACK TO PSD LABEL.THIS WAY
0926                * * THE BANZ USING ARO WILL WORK PROPER
0927 01A3 2570      LAC TBLPT,5  * LOAD ACC WITH STARTING ADDRESS OF COSIN
0928 01A4 6A72      LT CCOUNT
0929 01A5 800E      MPYK 14      *
0930 01A6 7F8F      APAC
0931                *
0932                *
0933                *
0934                *
0935 01A7 700D      LARK 0,13    *ARO IS SET UP AS A COUNTER TO READ IN
0936 01A8 7100      LARK 1,0     * IN THE 14 COSINE COEFFECIENTS
0937 01A9 6881      COOK LARP 1
0938 01AA 67A0      TBLR **+,0
0939 01AB 007D      ADD ONE      * INC ADDRESS IN ACC
0940 01AC F400      BANZ COOK
      01AD 01A9
0941
0942
0943                * START FORMING THE PRSUM

```

```

0944 *
0945 01AE 7022 LARK 0,34
0946 01AF 710D LARK 1,13
0947 01B0 7F89 ZAC
0948 01B1 6881 LARP 1
0949 01B2 6A80 PLOOP LT *,0
0950 01B3 6D91 MPY *-,1
0951 01B4 7F8F APAC
0952 01B5 F400 BANZ PLOOP
    01B6 01B2

0953 *
0954 *
0955 01B7 6A7A LT A2048 * T REG = 2048
0956 01B8 6D7A MPY A2048
0957 01B9 7F8F APAC
0958 *
0959 01BA 7F88 ABS
0960 01BB 5C74 SACH PRSUM,4
0961 01BC 626D SUBH P1
0962 01BD 636E SUBS P2 * PRSUM(32 BITS) - (P1,P2) --> ACC.
0963 01BE FA00 BLZ SQP2 * IS PRSUM > PRSUM_MAX ?
    01BF 01C2

0964 *
0965 01C0 246D LAC P1,4 * YES !
0966 01C1 5074 SACL PRSUM,0 * SET PRSUM = 3C00
0967 *
0968 *
0969 * READ IN THE SINE COEFFECIENTS
0970 *
0971 01C2 2076 SQP2 LAC SINST,0 * ACC= STARTING ADDRESS OF SINE TABLE
0972 01C3 6A72 LT CCOUNT
0973 01C4 800E MPYK 14
0974 01C5 7F8F APAC
0975 *
0976 01C6 700D LARK 0,13 * ARO AS COUNTER TO READ 14 SINE COEFFS
0977 01C7 7100 LARK 1,0
0978 01C8 6881 SOOK LARP 1
0979 01C9 67A0 TBLR **+,0
0980 01CA 007D ADD ONE
0981 01CB F400 BANZ SOOK
    01CC 01C8

0982 *
0983 *
0984 *
0985 * START FORMING THE IRSUM
0986 *
0987 *
0988 01CD 7022 LARK 0,34
0989 01CE 710D LARK 1,13
0990 01CF 7F89 ZAC
0991 01D0 6881 LARP 1
0992 01D1 6A80 ISPOP LT *,0
0993 01D2 6D91 MPY *-,1
0994 01D3 7F8F APAC
0995 01D4 F400 BANZ ISPOP
    01D5 01D1

0996 *

```

```

0997 01D6 7F88      ABS
0998                *
0999 01D7 5C75      SACH IRSUM,4
1000 01D8 626D      SUBH P1
1001 01D9 636E      SUBS P2
1002 01DA FA00      BLZ DP3R
      01DB 01DE
1003                *
1004 01DC 246D      LAC P1,4
1005 01DD 5075      SACL IRSUM,0
1006                *
1007 01DE 7F89      DP3R ZAC
1008 01DF 6A74      LT PRSUM
1009 01E0 6D74      MPY PRSUM
1010 01E1 7F8F      APAC
1011 01E2 6A75      LT IRSUM
1012 01E3 6D75      MPY IRSUM
1013 01E4 7F8F      APAC
1014                *
1015 01E5 7F88      ABS
1016 01E6 5C7E      SACH TEMP1,4
1017                *
1018 01E7 207E      LAC TEMP1,0
1019 01E8 597F      SACH TEMP2,1
1020                *
0001 01E9 207F      LAC TEMP2,0
0002 01EA 107D      SUB ONE,0
1021 01EB FA00      BLZ TOK1
      01EC 01EF
1022                *
1023 01ED 2064      LAC T2,0
1024 01EE 507E      SACL TEMP1,0 * SATURATE DENOM
1025                *
1026                *
1027 01EF 7F89      TOK1 ZAC *
1028 01F0 7E51      LACK 81 *
1029 01F1 0072      ADD CCOUNT *
1030                *
1031 01F2 507F      SACL TEMP2
1032 01F3 387F      LAR 0,TEMP2
1033 01F4 207E      LAC TEMP1
1034 01F5 6880      LARP 0
1035 01F6 5088      SACL * * STORE DENOM(F)
1036                *
1037 01F7 6880      LARP 0 * }
1038 01F8 3872      LAR 0,CCOUNT * } INC CCOUNT
1039 01F9 68A8      MAR ** * }
1040 01FA 3072      SAR 0,CCOUNT * }
1041                *
1042                *
1043                * NOW TEST WHETHER ALL FREQ PTS HAVE BEEN CALCULATED.
1044                *
1045                *
1046 01FB 3871      LAR 0,FREQ
1047 01FC 6880      LARP 0
1048 01FD F400      BANZ PSD
      01FE 01A2

```

```

1049 01FF 7F80      NOP
1050      *
1051      *
1052      *
1053      *
1054      * NOW MAKE SURE THAT ALL THE PSD'S ARE POSITIVE.
1055      *
1056      *
1057 0200 2051      LAC L1,0
1058 0201 7F88      ABS
1059 0202 5051      SACL L1,0
1060      *
1061 0203 2052      LAC L2,0
1062 0204 7F88      ABS
1063 0205 5052      SACL L2,0
1064      *
1065 0206 2053      LAC L3,0
1066 0207 7F88      ABS
1067 0208 5053      SACL L3,0
1068      *
1069 0209 2054      LAC L4,0
1070 020A 7F88      ABS
1071 020B 5054      SACL L4,0
1072      *
1073 020C 2055      LAC H1,0
1074 020D 7F88      ABS
1075 020E 5055      SACL H1,0
1076      *
1077 020F 2056      LAC H2,0
1078 0210 7F88      ABS
1079 0211 5056      SACL H2,0
1080      *
1081 0212 2057      LAC H3,0
1082 0213 7F88      ABS
1083 0214 5057      SACL H3,0
1084      *
1085 0215 2058      LAC H4,0
1086 0216 7F88      ABS
1087 0217 5058      SACL H4,0
1088      *
1089      **
1090      *
1091 0218 7F80      NOP
1092      *
1093      *
1094      *
1095      *
1096      *
1097      *
1098      *
1099      * NOW FIND THE MINIMUM IN THE LOW GROUP
1100      * AND THE MINIMUM IN THE HIGH GROUP AND THEIR
1101      * CORRESSPONDING ADDRESSES.
1102      *
1103      *
1104 0219 7001      LARK 0,1
1105 021A 307D      SAR 0,ONE

```

```

1106 021B 3851          LAR 0,81
1107 021C 3042          SAR 0,LGMIN * LGMIN = L1
1108 021D 7001          LARK 0,1
1109 021E 3045          SAR 0,LGADD * LGADD = 1
1110
1111
1112                    CMP LGMIN,L2 * L1 - L2 = ACC
0001 021F 2042  LAC  LGMIN,0
0002 0220 1052  SUB  L2,0
1113 0221 FA00          BLZ N1 * IS L1 < L2 ?
      0222 0227
1114
1115 0223 2052          LAC L2,0 * NO! SO MAKE L2
1116 0224 5042          SACL LGMIN,0 * AS CURRENT LGMIN
1117 0225 7E02          LACK 2
1118 0226 5045          SACL LGADD,0 * CURRENT LGADD = 2
1119
1120                    N1  CMP LGMIN,L3 * LGMIN - L3 = ACC
0001 0227 2042  LAC  LGMIN,0
0002 0228 1053  SUB  L3,0
1121 0229 FA00          BLZ N2 * IS CURRENT LGMIN < L3 ?
      022A 022F
1122
1123 022B 2053          LAC L3,0 * NO! SO MAKE L3 AS
1124 022C 5042          SACL LGMIN,0 * CURRENT LGMIN
1125 022D 7E03          LACK 3
1126 022E 5045          SACL LGADD,0
1127
1128                    N2  CMP LGMIN,L4
0001 022F 2042  LAC  LGMIN,0
0002 0230 1054  SUB  L4,0
1129 0231 FA00          BLZ N3 * IS CURRENT LGMIN < L4 ?
      0232 0237
1130
1131 0233 2054          LAC L4,0 * NO! SO LGMIN = L4
1132 0234 5042          SACL LGMIN,0 *
1133 0235 7E04          LACK 4
1134 0236 5045          SACL LGADD,0 * LGADD = 4
1135
1136
1137
1138
1139 0237 2055  N3  LAC H1,0 *
1140 0238 5043          SACL HGMIN,0 * SET CURRENT HGMIN = H1
1141 0239 7E05          LACK 5
1142 023A 5046          SACL HGADD,0 * SET CURRENT HGADD = 5
1143
1144                    CMP HGMIN,H2
0001 023B 2043  LAC  HGMIN,0
0002 023C 1056  SUB  H2,0
1145 023D FA00          BLZ N4 * IS HGMIN < H2 ?
      023E 0243
1146
1147 023F 2056          LAC H2,0 * NO! SO SET
1148 0240 5043          SACL HGMIN,0 * CURRENT HGMIN = H2
1149 0241 7E06          LACK 6
1150 0242 5046          SACL HGADD,0 * HGADD = 6

```

```

1151      *
1152      N4      CMP HGMIN,H3
0001 0243 2043  LAC  HGMIN,0
0002 0244 1057  SUB   H3,0
1153 0245 FA00  BLZ  N5      * IS HGMIN < H3 ?
      0246 024B
1154      *
1155 0247 2057  LAC  H3,0      * NO! SO SET
1156 0248 5043  SACL HGMIN,0  * CURRENT HGMIN = H3
1157 0249 7E07  LACK 7
1158 024A 5046  SACL HGADD,0 *
1159      *
1160      N5      CMP HGMIN,H4
0001 024B 2043  LAC  HGMIN,0
0002 024C 1058  SUB   H4,0
1161 024D FA00  BLZ  N6      * IS HGMIN < H4
      024E 0253
1162      *
1163 024F 2058  LAC  H4,0      * NO! SO SET
1164 0250 5043  SACL HGMIN,0  * HGMIN = H4
1165 0251 7E08  LACK 8
1166 0252 5046  SACL HGADD,0 * HGADD = 8
1167      *
1168 0253 7F80  N6      NOP
1169      *
1170      *
1171      * NOW THE LOW GROUP MINIMUM IS IN LGMIN AND ITS ADDRESS
1172      * IN LGADD. THE HIGH GROUP MINIMUM IS IN HGMIN AND ITS ADDRE
1173      * IN HGADD.
1174      *
1175 0254 7EFF  LACK 255
1176 0255 507E  SACL TEMP1,0
1177 0256 287E  LAC  TEMP1,8
1178 0257 7A7E  OR   TEMP1
1179 0258 5063  SACL T1,0      * INITIALIZE T1 = FFFF
1180 0259 7001  LARK 0,1
1181 025A 307D  SAR 0,ONE      ) * SET ONE = 1
1182      *
1183      *
1184      * NOW DETERMINE WHETHER THE DIGIT IS PRESENT OR NOT.
1185      *
1186      * TO DO THIS THE FOLLOWING TESTS ARE DONE
1187      *
1188      *      1) IS LGMIN < KL.
1189      *
1190      *      2) IS HGMIN < KH.
1191      *
1192      *      3) IS S1 > KS1
1193      *
1194      *      4) IS S14 > KS14
1195      *
1196      * THE DIGIT IS PRESENT IF AND ONLY IF LGMIN < KL .AND. HGMIN
1197      *
1198      * .AND S1 > KS1 .AND. S14 > KS14 .
1199      *
1200 025B 7E32  LACK 50
1201 025C 5048  SACL KH,0      * INITIALIZE KH = 50

```

```

1202          *
1203 025D 7E64      LACK 100
1204 025E 5069      SACL KL,0      * INITIALIZE KL = 100
1205          *
1206 025F 7E01      LACK 1
1207 0260 507D      SACL ONE,0
1208 0261 6A7D      LT ONE
1209 0262 844C      MPYK 1100
1210 0263 7F8E      PAC
1211 0264 5049      SACL KS1      * INITIALIZE KS1= 1100
1212          *
1213 0265 6A7D      LT ONE
1214 0266 81F4      MPYK 500
1215 0267 7F8E      PAC
1216 0268 504A      SACL KS14,0      * INITIALIZE KS14 = 500
1217          *
1218 0269 2043      LAC HGMIN,0
1219 026A 1048      SUB KH
1220 026B FA00      BLZ CIP      * IS HGMIN < KH ?
      026C 026F
1221          *
1222 026D F900      B IVDIG      * NO! SO BRANCH TO IVDIG.
      026E 038F
1223          *
1224 026F 2042      CIP      LAC LGMIN,0
1225 0270 1069      SUB KL
1226 0271 FA00      BLZ GCP      * IS LGMIN < KL ?
      0272 0275
1227          *
1228 0273 F900      B IVDIG      * NO! SO BRANCH TO IVDIG.
      0274 038F
1229          *
1230 0275 2033      GCP      LAC S1,0
1231 0276 1049      SUB KS1
1232 0277 FC00      BGZ BASU      *
      0278 027B
1233          *
1234 0279 F900      B IVDIG
      027A 038F
1235          *
1236 027B 2040      BASU      LAC S14,0
1237 027C 104A      SUB KS14
1238 027D FC00      BGZ PARU
      027E 0281
1239          *
1240 027F F900      B IVDIG
      0280 038F
1241          *
1242          * THIS PROGRAM IS TO BE EXECUTED WHEN THE DIGIT IS PRESENT
1243          *
1244 0281 7EFF      PARU      LACK 255      *} SO HGMIN < KH .AND. LGMIN <KL
1245 0282 507E      SACL TEMP1,0      *} SET UP ADDRESS IN ACC.
1246 0283 237E      LAC TEMP1,3      *}
1247 0284 507F      SACL TEMP2,0
1248 0285 6765      TBLR VDPR      * FETCH VDPR
1249          *
1250 0286 2065      LAC VDPR,0

```

```

1251 0287 007D      ADD ONE
1252 0288 5065      SACL VDPR,0      * VDPR = VDPR + 1
1253 0289 207F      LAC TEMP2,0
1254 028A 7F81      DINT
1255 028B 7D65      TBLW VDPR      * RESTORE VDPR
1256                *
1257                *
1258                * SET UP A/D AFTER TBLW.
1259                *
1260                *
1261 028C 6E00      LDPK 0
1262 028D 700A      LARK 0,10
1263 028E 307E      SAR 0,TEMP1
1264 028F 7014      LARK 0,20
1265 0290 307F      SAR 0,TEMP2
1266 0291 6A7F      LT TEMP2
1267 0292 8015      MPYK 21
1268 0293 7F89      ZAC
1269 0294 7F8F      APAC
1270 0295 507F      SACL TEMP2
1271                *
1272 0296 497F      OUT TEMP2,PA1
1273 0297 487E      OUT TEMP1,PA0 *
1274                *
1275 0298 7F82      EINT
1276                *
1277 0299 2065      LAC VDPR,0
1278 029A 107D      SUB ONE      * VDPR - 1 = ACC
1279 029B FF00      BZ SANJ      * IS VDPR = 1 ?
      029C 02A6
1280                *
1281 029D 2065      LAC VDPR,0      * NO!
1282 029E 107D      SUB ONE      *} VDPR - 2 = ACC
1283 029F 107D      SUB ONE      *}
1284 02A0 FF00      BZ RAJAM      * IS VDPR = 2 ?
      02A1 031F
1285                *
1286 02A2 7E02      LACK 2      * NO!
1287 02A3 5069      SACL PREST,0 * SET PRESENT = 2
1288 02A4 F900      B'DACOL
      02A5 0453
1289                *
1290 02A6 7EA8      SANJ LACK 168      * YES! VDPR = 1.
1291 02A7 507E      SACL TEMP1,0
1292 02A8 237E      LAC TEMP1,3
1293 02A9 504C      SACL DIGTB      * SET DIGTB = 1344
1294 02AA 2045      LAC LGADD,0
1295 02AB 0046      ADD HGADD,0
1296 02AC 507E      SACL TEMP1      * TEMP1 = LGADD + HGADD
1297                *
1298 02AD 7E01      LACK 1
1299 02AE 507F      SACL TEMP2,0
1300                *
0001 02AF 2045      LAC LGADD,0      * LGADD - TEMP2 --> ACC.
0002 02B0 107F      SUB TEMP2,0
1301 02B1 FF00      BZ Y16      * IS LGADD = 1 ?
      02B2 02D7

```

```

1302          *
1303 02B3 7E02          LACK 2
1304 02B4 507F          SACL TEMP2,0
1305          CMP LGADD,TEMP2  * LGADD - TEMP2 --> ACC.
0001 02B5 2045  LAC  LGADD,0
0002 02B6 107F  SUB  TEMP2,0
1306 02B7 FF00          BZ  Y17          * IS LGADD = 2 ?
      02B8 02EF
1307          *
1308 02B9 7E03          LACK 3
1309 02BA 507F          SACL TEMP2,0
1310          CMP LGADD,TEMP2
0001 02BB 2045  LAC  LGADD,0
0002 02BC 107F  SUB  TEMP2,0
1311 02BD FF00          BZ  Y18          * IS LGADD = 3 ?
      02BE 0307
1312          *
1313 02BF 237E          LAC TEMP1,3          * SO LGADD = 4 !!
1314 02C0 004C          ADD DIGTB
1315 02C1 6766          TBLR CD1          * SET CD1 = PARTICULAR DIGIT
1316          *
1317 02C2 7EFF          LACK 255
1318 02C3 507E          SACL TEMP1,0
1319 02C4 237E          LAC TEMP1,3
1320 02C5 007D          ADD ONE
1321 02C6 7F81          DINT
1322 02C7 7D66          TBLW CD1          * RESTORE CD1 AT ROM +2041
1323          *
1324          *
1325          * SET UP A/D AFTER TBLW.
1326          *
1327          *
1328 02C8 6E00          LDPK 0
1329 02C9 700A          LARK 0,10
1330 02CA 307E          SAR 0,TEMP1
1331 02CB 7014          LARK 0,20
1332 02CC 307F          SAR 0,TEMP2
1333 02CD 6A7F          LT TEMP2
1334 02CE 8015          MPYK 21
1335 02CF 7F89          ZAC
1336 02D0 7F8F          APAC
1337 02D1 507F          SACL TEMP2  *
1338          *
1339 02D2 497F          OUT TEMP2,PA1
1340 02D3 487E          OUT TEMP1,PA0
1341          *
1342 02D4 7F82          EINT
1343 02D5 F900          B  DACOL
      02D6 0453
1344          *
1345 02D7 207E  Y16 LAC TEMP1,0          * SO LGADD = 1 !
1346 02D8 004C          ADD DIGTB
1347 02D9 6766          TBLR CD1          * SET CD1 = PARTICULAR DIGIT
1348          *
1349 02DA 7EFF          LACK 255
1350 02DB 507E          SACL TEMP1,0
1351 02DC 237E          LAC TEMP1,3

```

```

1352 02DD 007D          ADD ONE
1353 02DE 7F81          DINT
1354 02DF 7D66          TBLW CD1          * RESTORE CD1 AT ROM +2041
1355                    *
1356                    *
1357                    * SET UP A/D AFTER TBLW.
1358                    *
1359                    *
1360 02E0 6E00          LDPK 0
1361 02E1 700A          LARK 0,10
1362 02E2 307E          SAR 0,TEMP1
1363 02E3 7014          LARK 0,20
1364 02E4 307F          SAR 0,TEMP2
1365 02E5 6A7F          LT TEMP2
1366 02E6 8015          MPYK 21
1367 02E7 7F89          ZAC
1368 02E8 7F8F          APAC
1369 02E9 507F          SACL TEMP2      *
1370                    *
1371 02EA 497F          OUT TEMP2,PA1   *
1372 02EB 487E          OUT TEMP1,PA0   *
1373                    *
1374 02EC 7F82          EINT
1375 02ED F900          B DACOL
      02EE 0453
1376                    *
1377 02EF 217E          Y17 LAC TEMP1,1   * SO LGADD = 2 ?
1378 02F0 004C          ADD DIGTB
1379 02F1 6766          TBLR CD1
1380 02F2 7EFF          LACK 255
1381 02F3 507E          SACL TEMP1,0
1382 02F4 237E          LAC TEMP1,3
1383 02F5 007D          ADD ONE
1384 02F6 7F81          DINT
1385 02F7 7D66          TBLW CD1          * RESTORE CD1 AT ROM +2041
1386                    *
1387                    *
1388                    * SET UP A/D AFTER TBLW.
1389                    *
1390                    *
1391 02F8 6E00          LDPK 0
1392 02F9 700A          LARK 0,10
1393 02FA 307E          SAR 0,TEMP1
1394 02FB 7014          LARK 0,20
1395 02FC 307F          SAR 0,TEMP2
1396 02FD 6A7F          LT TEMP2
1397 02FE 8015          MPYK 21
1398 02FF 7F89          ZAC
1399 0300 7F8F          APAC
1400 0301 507F          SACL TEMP2      *
1401                    *
1402 0302 497F          OUT TEMP2,PA1
1403 0303 487E          OUT TEMP1,PA0   *
1404                    *
1405 0304 7F82          EINT
1406 0305 F900          B DACOL
      0306 0453

```

```

1407          *
1408 0307 227E Y18  LAC TEMP1,2      * SO LGADD = 3 ?
1409 0308 004C      ADD DIGTB
1410 0309 6766      TBLR CD1        * SET CD1 = PARTICULAR DIGIT
1411 030A 7EFF      LACK 255
1412 030B 507E      SACL TEMP1,0
1413 030C 237E      LAC TEMP1,3
1414 030D 007D      ADD ONE
1415 030E 7F81      DINT
1416 030F 7D66      TBLW CD1        * RESTORE CD1 AT ROM +2041
1417          *
1418          *
1419          * SET UP A/D AFTER TBLW.
1420          *
1421          *
1422 0310 6E00      LDPK 0
1423 0311 700A      LARK 0,10
1424 0312 307E      SAR 0,TEMP1
1425 0313 7014      LARK 0,20
1426 0314 307F      SAR 0,TEMP2
1427 0315 6A7F      LT TEMP2
1428 0316 8015      MPYK 21
1429 0317 7F89      ZAC
1430 0318 7F8F      APAC
1431 0319 507F      SACL TEMP2
1432          *
1433 031A 497F      OUT TEMP2,PA1    * SET SAMPLING FREQ. = 8 KHZ.
1434 031B 487E      OUT TEMP1,PA0    * SET A/D IN CINTINOUS MODE
1435          *
1436 031C 7F82      EINT
1437 031D F900      B DACOL
      031E 0453
1438          *
1439 031F 7EA8      RAJAM LACK 168      * YES ! VDPR = 2
1440 0320 507E      SACL TEMP1,0
1441 0321 237E      LAC TEMP1,3
1442 0322 504C      SACL DIGTB      * SET DIGTB = 1344
1443 0323 2045      LAC LGADD,0
1444 0324 0046      ADD HGADD,0
1445 0325 507E      SACL TEMP1      * TEMP1 = LGADD + HGADD
1446          *
1447 0326 7E01      LACK 1
1448 0327 507F      SACL TEMP2,0
1449          CMP LGADD,TEMP2    * LGADD - TEMP2 --> ACC.
0001 0328 2045      LAC LGADD,0
0002 0329 107F      SUB TEMP2,0
1450 032A FF00      BZ Y16Z        * IS LGADD = 1 ?
      032B 033F
1451          *
1452 032C 7E02      LACK 2
1453 032D 507F      SACL TEMP2,0
1454          CMP LGADD,TEMP2    * LGADD - TEMP2 --> ACC.
0001 032E 2045      LAC LGADD,0
0002 032F 107F      .SUB TEMP2,0
1455 0330 FF00      BZ Y17Z        * IS LGADD = 2 ?
      0331 0344
1456 0332 FF00      BZ Y17Z        * IS LGADD = 1 ?

```

```

0333 0344
1457 *
1458 0334 7E03 LACK 3
1459 0335 507F SACL TEMP2,0
1460 CMP LGADD,TEMP2
0001 0336 2045 LAC LGADD,0
0002 0337 107F SUB TEMP2,0
1461 0338 FF00 BZ Y18Z * IS LGADD = 3 ?
0339 0349
1462 *
1463 033A 237E LAC TEMP1,3 * SO LGADD = 4 !!
1464 033B 004C ADD DIGTB
1465 033C 6767 TBLR CD2 * SET CD2 = PARTICULAR DIGIT
1466 033D F900 B MANJU
033E 034C
1467 *
1468 033F 207E Y16Z LAC TEMP1,0 * SO LGADD = 1 !
1469 0340 004C ADD DIGTB
1470 0341 6767 TBLR CD2 * SET CD2 = PARTICULAR DIGIT
1471 *
1472 0342 F900 B MANJU
0343 034C
1473 *
1474 0344 217E Y17Z LAC TEMP1,1 * SO LGADD = 2 !
1475 0345 004C ADD DIGTB
1476 0346 6767 TBLR CD2
1477 0347 F900 B MANJU
0348 034C
1478 *
1479 0349 227E Y18Z LAC TEMP1,2 * SO LGADD = 3 !
1480 034A 004C ADD DIGTB
1481 034B 6767 TBLR CD2 * SET CD2 = PARTICULAR DIGIT
1482 *
1483 034C 7EFF MANJU LACK 255
1484 034D 507E SACL TEMP1,0
1485 034E 237E LAC TEMP1,3
1486 034F 007D ADD ONE
1487 0350 6766 TBLR CD1 * FETCH CD1
1488 *
1489 CMP CD1,CD2
0001 0351 2066 LAC CD1,0
0002 0352 1067 SUB CD2,0
1490 0353 FF00 BZ GPM * IS CD1 = CD2 ?
0354 0374
1491 *
1492 0355 7F89 ZAC * NO!
1493 0356 5069 SACL PREST * SET PRESENT = 0 .
1494 0357 2063 LAC T1,0 *
1495 0358 5066 SACL CD1,0 * SET CD1 = FFFF
1496 0359 5067 SACL CD2,0 * SET CD2 = FFFF
1497 *
1498 035A 7EFF LACK 255
1499 035B 507E SACL TEMP1,0 *
1500 035C 237E LAC TEMP1,3 * RESTORE CD1,CD2,PRESENT
1501 035D 007D ADD ONE *
1502 035E 7F81 DINT
1503 035F 7D66 TBLW CD1 *

```

```

1504 0360 007D      ADD ONE
1505 0361 7D67      TBLW CD2
1506 0362 007D      ADD ONE
1507 0363 007D      ADD ONE
1508 0364 7D69      TBLW PREST
1509
1510
1511
1512
1513
1514 0365 6E00      LDPK 0
1515 0366 700A      LARK 0,10
1516 0367 307E      SAR 0,TEMP1
1517 0368 7014      LARK 0,20
1518 0369 307F      SAR 0,TEMP2
1519 036A 6A7F      LT TEMP2
1520 036B 8015      MPYK 21
1521 036C 7F89      ZAC
1522 036D 7F8F      APAC
1523 036E 507F      SACL TEMP2
1524
1525 036F 497F      OUT TEMP2,PA1
1526 0370 487E      OUT TEMP1,PA0
1527
1528 0371 7F82      EINT
1529 0372 F900      B DACOL
    0373 0453
1530
1531 0374 7E02      GPM LACK 2
1532 0375 5069      SACL PREST,0
1533 0376 7E03      LACK 3
1534 0377 507F      SACL TEMP2,0
1535 0378 7EFF      LACK 255
1536 0379 507E      SACL TEMP1,0
1537 037A 237E      LAC TEMP1,3
1538 037B 007F      ADD TEMP2
1539 037C 7F81      DINT
1540 037D 7D67      TBLW CD2
1541 037E 007D      ADD ONE
1542 037F 7D69      TBLW PREST
1543
1544
1545
1546
1547
1548 0380 6E00      LDPK 0
1549 0381 700A      LARK 0,10
1550 0382 307E      SAR 0,TEMP1
1551 0383 7014      LARK 0,20
1552 0384 307F      SAR 0,TEMP2
1553 0385 6A7F      LT TEMP2
1554 0386 8015      MPYK 21
1555 0387 7F89      ZAC
1556 0388 7F8F      APAC
1557 0389 507F      SACL TEMP2
1558
1559 038A 497F      OUT TEMP2,PA1

```

* SET UP A/D AFTER TBLW.

* SET UP A/D AFTER TBLW.

* SET UP A/D AFTER TBLW.

* SET UP A/D AFTER TBLW.

* YES ! CD1 = CD2
* SET PRESENT = 2

* STORE CD2 AT ROM LOC. +2043 = CD3
* RESTORE PRESENT @ +2044

```

1560 038B 487E      OUT TEMP1,PA0 *
1561                *
1562 038C 7F82      EINT
1563 038D F900      B DACOL
      038E 0453
1564                *
1565                *
1566                * THIS PROGRAM IS EXECUTED WHEN THE DIGIT
1567                *
1568                * IS NOT PRESENT. ✓
1569                *
1570 038F 7F89      IVDIG ZAC          * SET VDPR = 0
1571 0390 5065      SACL VDPR,0        *      CD1 = FFFF
1572 0391 2063      LAC T1,0          *      CD2 = FFFF .
1573 0392 5066      SACL CD1,0
1574 0393 5067      SACL CD2,0
1575 0394 7EFF      LACK 255
1576 0395 507E      SACL TEMP1,0
1577 0396 237E      LAC TEMP1,3
1578 0397 7F81      DINT
1579 0398 7D65      TBLW VDPR
1580 0399 007D      ADD ONE
1581 039A 7D66      TBLW CD1
1582 039B 007D      ADD ONE
1583 039C 7D67      TBLW CD2
1584                *
1585                *
1586                * SET UP A/D AFTER TBLW.
1587                *
1588                *
1589 039D 6E00      LDPK 0
1590 039E 700A      LARK 0,10
1591 039F 307E      SAR 0,TEMP1
1592 03A0 7014      LARK 0,20
1593 03A1 307F      SAR 0,TEMP2
1594 03A2 6A7F      LT TEMP2
1595 03A3 8015      MPYK 21
1596 03A4 7F89      ZAC
1597 03A5 7F8F      APAC
1598 03A6 507F      SACL TEMP2
1599                *
1600 03A7 497F      OUT TEMP2,PA1
1601 03A8 487E      OUT TEMP1,PA0
1602                *
1603 03A9 7F82      EINT
1604                *
1605 03AA 7E05      LACK 5
1606 03AB 507F      SACL TEMP2,0
1607 03AC 7EFF      LACK 255
1608 03AD 507E      SACL TEMP1,0
1609 03AE 237E      LAC TEMP1,3
1610 03AF 007F      ADD TEMP2
1611 03B0 507F      SACL TEMP2,0
1612                *
1613 03B1 676A      TBLR VDAB          * FETCH VDAB FROM ROM +2045
1614 03B2 206A      LAC VDAB,0
1615 03B3 007D      ADD ONE

```

```

1616 03B4 506A      SACL VDAB,0      * VDAB = VDAB + 1
1617 03B5 207F      LAC TEMP2,0
1618 03B6 7F81      DINT
1619 03B7 7D6A      TBLW VDAB      * RESTORE VDAB
1620
1621
1622      * SET UP A/D AFTER TBLW.
1623
1624
1625 03B8 6E00      LDPK 0
1626 03B9 700A      LARK 0,10
1627 03BA 307E      SAR 0,TEMP1
1628 03BB 7014      LARK 0,20
1629 03BC 307F      SAR 0,TEMP2
1630 03BD 6A7F      LT TEMP2
1631 03BE 8015      MPYK 21
1632 03BF 7F89      ZAC
1633 03C0 7F8F      APAC
1634 03C1 507F      SACL TEMP2      *
1635
1636 03C2 497F      OUT TEMP2,PA1    *
1637 03C3 487E      OUT TEMP1,PA0    *
1638
1639 03C4 7F82      EINT
1640
1641 03C5 7E04      LACK 4
1642 03C6 507F      SACL TEMP2,0
1643 03C7 7EFF      LACK 255
1644 03C8 507E      SACL TEMP1,0
1645 03C9 237E      LAC TEMP1,3
1646 03CA 007F      ADD TEMP2
1647 03CB 6769      TBLR PREST      * FETCH PRESENT FROM ROM +2044
1648
1649 03CC 7E02      LACK 2
1650 03CD 1069      SUB PREST
1651 03CE FF00      BZ PBM          * IS PRESENT = 2 ?
      03CF 03F6
1652
1653 03D0 7F89      JEQS ZAC        * NO!
1654 03D1 5069      SACL PREST,0    * SET PRESENT = 0
1655 03D2 506A      SACL VDAB,0      * VDAB = 0
1656 03D3 2063      LAC T1,0        * AD1 = FFFF
1657 03D4 506B      SACL AD1        * AD2 = FFFF
1658 03D5 506C      SACL AD2        * CD3 = FFFF
1659 03D6 5068      SACL CD3        *
1660
1661 03D7 7E03      LACK 3
1662 03D8 507F      SACL TEMP2,0
1663 03D9 7EFF      LACK 255
1664 03DA 507E      SACL TEMP1,0
1665 03DB 237E      LAC TEMP1,3
1666 03DC 007F      ADD TEMP2
1667 03DD 7F81      DINT
1668 03DE 7D68      TBLW CD3
1669 03DF 007D      ADD ONE
1670 03E0 7D69      TBLW PREST
1671 03E1 007D      ADD ONE

```

```

1672 03E2 7D6A      TBLW VDAB
1673 03E3 007D      ADD ONE
1674 03E4 7D6B      TBLW AD1
1675 03E5 007D      ADD ONE
1676 03E6 7D6C      TBLW AD2
1677                *
1678                *
1679                * SET UP A/D AFTER TBLW.
1680                *
1681                *
1682 03E7 6E00      LDPK 0
1683 03E8 700A      LARK 0,10
1684 03E9 307E      SAR 0,TEMP1
1685 03EA 7014      LARK 0,20
1686 03EB 307F      SAR 0,TEMP2
1687 03EC 6A7F      LT TEMP2
1688 03ED 8015      MPYK 21
1689 03EE 7F89      ZAC
1690 03EF 7F8F      APAC
1691 03F0 507F      SACL TEMP2      *
1692                *
1693 03F1 497F      OUT TEMP2,PA1      *
1694 03F2 487E      OUT TEMP1,PA0      *
1695                *
1696 03F3 7F82      EINT
1697 03F4 F900      B DACOL
      03F5 0453
1698                *
1699 03F6 7E01      PBM LACK 1
1700 03F7 106A      SUB VDAB
1701 03F8 FF00      BZ JMTP      * IS VDAB = 1 ?
      03F9 0400
1702                *
1703 03FA 7E02      LACK 2      * NO!
1704 03FB 106A      SUB VDAB
1705 03FC FF00      BZ JMSQ      * IS VDAB = 2 ?
      03FD 0419
1706                *
1707 03FE F900      B JEQS      * NO! GO TO RESET.
      03FF 03D0
1708                *
1709 0400 7F89      JMTP ZAC      * YES ! VDAB = 1
1710 0401 506B      SACL AD1,0      *
1711 0402 7E06      LACK 6      * SET AD1 = 0
1712 0403 507F      SACL TEMP2,0
1713 0404 7EFF      LACK 255
1714 0405 507E      SACL TEMP1,0
1715 0406 237E      LAC TEMP1,3
1716 0407 007F      ADD TEMP2
1717 0408 7F81      DINT
1718 0409 7D6B      TBLW AD1      * RESTORE AD1 AT ROM +2046
1719                *
1720                *
1721                * SET UP A/D AFTER TBLW.
1722                *
1723                *
1724 040A 6E00      LDPK 0

```

```

1725 040B 700A      LARK 0,10
1726 040C 307E      SAR 0,TEMP1
1727 040D 7014      LARK 0,20
1728 040E 307F      SAR 0,TEMP2
1729 040F 6A7F      LT TEMP2
1730 0410 8015      MPYK 21
1731 0411 7F89      ZAC
1732 0412 7F8F      APAC
1733 0413 507F      SACL TEMP2      * CLCK = +420 SO THAT FS=8KHZ.
1734                *
1735 0414 497F      OUT TEMP2,PA1
1736 0415 487E      OUT TEMP1,PA0  *
1737                *
1738 0416 7F82      EINT
1739 0417 F900      B DACOL
      . 0418 0453
1740                *
1741 0419 7F89      JMSQ  ZAC
1742 041A 506C      SACL AD2,0
1743 041B 7E06      LACK 6
1744 041C 507F      SACL TEMP2,0
1745 041D 7EFF      LACK 255
1746 041E 507E      SACL TEMP1,0
1747 041F 237E      LAC TEMP1,3
1748 0420 007F      ADD TEMP2
1749 0421 676B      TBLR AD1      * FETCH AD1 @ ROM +2046
1750 0422 206B      LAC AD1,0
1751 0423 106C      SUB AD2
1752 0424 FF00      BZ JQPS      * IS AD2 = AD1 ?
      0425 0428
1753                *
1754 0426 F900      B JEQS      * NO! SO GO TO RESET
      0427 03D0
1755                *      PRESENT = 0 , VDAB = 0
1756                *      AD1 = FFFF , AD2 = FFFF
1757                *      CD3 = FFFF .
1758                *
1759 0428 7E03      JQPS  LACK 3
1760 0429 507F      SACL TEMP2,0
1761 042A 7EFF      LACK 255
1762 042B 507E      SACL TEMP1,0
1763 042C 237E      LAC TEMP1,3
1764 042D 007F      ADD TEMP2
1765 042E 677F      TBLR TEMP2   * FETCH CD3 @ ROM +2043
1766                *
1767                *
1768                * OUTPUT THE DIGIT
1769                *
1770                **
1771 042F 7002      LARK 0,2
1772 0430 307E      SAR 0,TEMP1
1773 0431 487E      OUT TEMP1,PA0
1774 0432 397F      LAR 1,TEMP2  *
1775 0433 7080      LARK 0,128
1776 0434 307A      SAR 0,A2048
1777 0435 247A      LAC A2048,4
1778 0436 507A      SACL A2048,0  * INITIALIZE A2048 = +2048

```

```

1779 0437 7E96 Y61 LACK 150
1780 0438 507E SACL TEMP1,0
1781 0439 217E LAC TEMP1,1
1782 043A 507E SACL TEMP1,0
1783 043B 387E LAR 0,TEMP1 * ARO = ONTIME = 300
1784 043C 6880 LARP 0
1785 043D 4A7A Y62 OUT A2048,PA2 * +5 -->
1786 043E 7F80 NOP
1787 043F F400 BANZ Y62
    0440 043D
1788 *
1789 0441 7E96 LACK 150
1790 0442 507E SACL TEMP1,0
1791 0443 217E LAC TEMP1,1
1792 0444 507E SACL TEMP1,0
1793 0445 387E LAR 0,TEMP1 * ARO = OFFTIME = 300
1794 *
1795 0446 7F89 ZAC
1796 0447 107A SUB A2048
1797 0448 507F SACL TEMP2 * TEMP2 = -2048
1798 0449 6880 LARP 0
1799 044A 4A7F Y63 OUT TEMP2,PA2 *
1800 044B 7F80 NOP
1801 044C F400 BANZ Y63
    044D 044A
1802 *
1803 044E 6881 LARP 1
1804 044F F400 BANZ Y61
    0450 0437
1805 *
1806 0451 F900 B JEQS * NOW THAT YOU HAVE OUTPUT
    0452 03D0
1807 * THE DIGIT GO TO RESET
1808 * PRESENT = 0
1809 * VDAB = 0, AD1 = FFFF, AD2 = FFFF
1810 * CD3 = FFFF.
1811 *
1812 *
1813 * NOW TRANSFER DATA FROM PROGRAM RAM TO DATA RAM
1814 *
1815 *
1816 *
1817 0453 7F81 DACOL DINT
1818 0454 7EFA LACK 250 *}
1819 0455 507E SACL TEMP1 *}
1820 0456 237E LAC TEMP1,3 *} SET ROM LOC. 2000 TO +10
1821 0457 700A LARK 0,10 *}
1822 0458 307E SAR 0,TEMP1 *}
1823 0459 7D7E TBLW TEMP1 *}
1824 *
1825 *
1826 045A 7E01 LACK 1
1827 045B 507D SACL ONE * INITIALIZE ONE= 1
1828 045C 7EC8 LACK 200
1829 045D 507E SACL TEMP1 *}
1830 045E 237E LAC TEMP1,3 *} SET UP ADDRESS 1600 IN ACC.
1831 045F 507E SACL TEMP1 *} TEMP1 = 1600

```

```

1832      *
1833      *
1834 0460 677F      TBLR TEMP2      * GET DATA POINTER @ 1600
1835      * NOV. 20
1836 0461 007D      ADD ONE          *}
1837 0462 507E      SACL TEMP1         *}
1838 0463 107D      SUB ONE           *} NOW SET ROM.LOC.+1600 TO +1601
1839 0464 7D7E      TBLW TEMP1         *} SO THAT THE NEXT SET OF DATA
1840 0465 507E      SACL TEMP1         *} WILL BE STORED FROM +1601 ONWARDS
1841      * NOV. 20
1842      *
1843      * SET UP A/D AFTER THE TBLW.
1844      *
1845 0466 6E01      LDPK 1
1846 0467 700A      LARK 0,10
1847 0468 3000      SAR 0,CNTRL      *
1848      *
1849 0469 7014      LARK 0,20
1850 046A 3001      SAR 0,CLCK
1851 046B 6A01      LT CLCK
1852 046C 8015      MPYK 21
1853 046D 7F89      ZAC
1854 046E 7F8F      APAC
1855 046F 5001      SACL CLCK,0      *
1856      *
1857 0470 4901      OUT CLCK,PA1     *
1858 0471 4800      OUT CNTRL,PA0    *
1859 0472 6E00      LDPK 0
1860      *
1861      *
1862      *
1863 0473 207F      LAC TEMP2,0      *
1864 0474 107E      SUB TEMP1        * 1600+N-1600 = N DATA SAMPLES HAVE BEEN
1865      *                               COLLECTED.
1866 0475 107D      SUB ONE
1867 0476 107D      SUB ONE
1868 0477 507E      SACL TEMP1
1869 0478 397E      LAR 1,TEMP1     * SET UP ARI AS COUNTER
1870 0479 387E      LAR 0,TEMP1     *AR0=ADDRESS @ RAM FROM WHICH DATA
1871      * HAS TO LOADED AND ARO MUST BE DECREMENTED.
1872 047A 207F      LAC TEMP2,0
1873 047B 107D      SUB ONE
1874 047C 6880      Y50 LARP 0
1875 047D 6798      TBLR *-          * TRANSFER DATA FROM PRAM
1876 047E 107D      SUB ONE          * TO DRAM.
1877 047F 6881      LARP 1
1878 0480 F400      BANZ Y50
1879      *
1880      *
1881      *
1882      * PROGRAM TO COLLECT REMAINING DATA.
1883      *
1884      *
1885 0482 207E      LAC TEMP1,0
1886 0483 007D      ADD ONE
1887 0484 507E      SACL TEMP1

```

```

1888 0485 387E      LAR 0,TEMP1
1889 0486 7E80      LACK 128
1890 0487 107E      SUB TEMP1
1891 0488 107D      SUB ONE
1892 0489 507E      SACL TEMP1
1893 048A 397E      LAR 1,TEMP1      * ARI NOW CONTAINS THE NO. OF REMAINING
1894 *                                     DATA SAMPLES TO BE COLLECTED.
1895 *
1896 048B
1897 048B 7F80      NOP
1898 048C F900      B LOOP
      048D 001A
1899 *
1900 *
1901 *
1902 *
1903 *      INTERRUPT SERVICE ROUTINE
1904 *
1905 *
1906 048E 7C0E      INTPC  SST 14      * SAVE STATUS @ DRAM >142
1907 048F 6E00      LDPK 0
1908 0490 507E      SACL TEMP1      *}
1909 0491 6E01      LDPK 1
1910 0492 580F      SACH 15      *} SAVE ACCUMULATOR
1911 0493 6E00      LDPK 0
1912 0494 7F9D      POP
1913 0495 507C      SACL T3      * SAVE STACK 0 @ T3
1914 0496 7F9C      PUSH      * RESTORE STACK 0 AS IT IS
1915 *                                     NEEDED IF YOU ARE GETTING DATA('GET'ROU
1916 *
1917 *      NOW TEST SWITCH S1
1918 *
1919 0497 7EFA      LACK 250
1920 0498 507F      SACL TEMP2
1921 0499 237F      LAC TEMP2,3
1922 049A 677F      TBLR TEMP2      * TEMP2= S1
1923 049B 7E0A      LACK 10
1924 049C 107F      SUB TEMP2
1925 049D FF00      BZ GET
      049E 0023
1926 *
1927 *
1928 *      ROUTINE TO BE SERVICED IF INTERRUPT
1929 *      OCCURED IN THE MAIN PROGRAM
1930 *
1931 *
1932 *      SAVE CONTEXT
1933 *
1934 *
1935 049F 7E01      LACK 1
1936 04A0 507D      SACL ONE      * INITIALIZE ONE = 1
1937 04A1 7EBC      LACK 188
1938 04A2 507F      SACL TEMP2
1939 04A3 237F      LAC TEMP2,3      * SET UP ADDRESS >1504, IN ACC.
1940 04A4 307F      SAR 0,TEMP2
1941 04A5 7D7F      TBLW TEMP2      * SAVE ARO @ PROM >1504
1942 04A6 007D      ADD ONE

```

```

1943 04A7 317F      SAR 1,TEMP2
1944 04A8 7D7F      TBLW TEMP2      * SAVE ARI @ PROM >1505
1945 04A9 007D      ADD ONE
1946 04AA 7D7E      TBLW TEMP1      * SAVE ACCL @ PROM >1506
1947 04AB 708F      LARK 0,143
1948 04AC 6880      LARP 0
1949 04AD 007D      ADD ONE
1950 04AE 7D88      TBLW *          * SAVE ACCH @ PROM >1507
1951 04AF 507F      SACL TEMP2
1952 04B0 8001      MPYK 1
1953 04B1 7F8E      PAC
1954 04B2 6E01      LDPK 1
1955 04B3 500F      SACL 15         * SAVE T REG. @ DRAM >143
1956 04B4 6E00      LDPK 0
1957 04B5 207F      LAC TEMP2,0
1958 04B6 007D      ADD ONE
1959 04B7 708F      LARK 0,143
1960 04B8 6880      LARP 0
1961 04B9 7D98      TBLW *-        * SAVE T REG. @ PROM >1508
1962 04BA 007D      ADD ONE
1963 04BB 7D88      TBLW *         * SAVE STATUS @ PROM >1509
1964 04BC 6E00      LDPK 0
1965
1966
1967
1968
1969
1970 04BD 70C8      LARK 0,200
1971 04BE 307E      SAR 0,TEMP1
1972 04BF 237E      LAC TEMP1,3    * SET UP ADDRESS OF DATA POINTER
1973 04C0 677E      TBLR TEMP1     * TEMP1 = DATA POINTER
1974
1975 04C1 427F      IN TEMP2,2
1976 04C2 207E      LAC TEMP1
1977 04C3 7D7F      TBLW TEMP2     * STORE DATA SAMPLE @ ADDRESS
1978
1979
1980 04C4 207E      LAC TEMP1
1981 04C5 007D      ADD ONE        * INC DATA POINTER
1982 04C6 507E      SACL TEMP1
1983 04C7 70C8      LARK 0,200
1984 04C8 307F      SAR 0,TEMP2
1985 04C9 237F      LAC TEMP2,3
1986 04CA 7D7E      TBLW TEMP1     * RESTORE DATA POINTER
1987
1988
1989
1990
1991
1992 04CB 6E00      LDPK 0
1993 04CC 700A      LARK 0,10
1994 04CD 307E      SAR 0,TEMP1
1995 04CE 7014      LARK 0,20
1996 04CF 307F      SAR 0,TEMP2
1997 04D0 6A7F      LT TEMP2
1998 04D1 8015      MPYK 21
1999 04D2 7F89      ZAC

```

```

*
*
* SERVICE
*
*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

*

```

```

2000 04D3 7F8F          APAC
2001 04D4 507F          SACL TEMP2      *
2002                    *
2003 04D5 497F          OUT TEMP2,PA1   *
2004 04D6 487E          OUT TEMP1,PA0   *
2005                    *
2006                    *
2007                    * RESTORE CONTEXT
2008                    *
2009                    *
2010 04D7 7EBC          LACK 188
2011 04D8 507F          SACL TEMP2
2012 04D9 237F          LAC TEMP2,3     * ACC = +1504
2013 04DA 007D          ADD ONE
2014 04DB 007D          ADD ONE
2015 04DC 708E          LARK 0,142
2016 04DD 6880          LARP 0
2017 04DE 67A8          TBLR *+         * DRAM >142= ACCL
2018 04DF 007D          ADD ONE
2019 04E0 6788          TBLR *         * DRAM >143=ACCH
2020                    *
2021 04E1 7EBC          LACK 188
2022 04E2 507F          SACL TEMP2
2023 04E3 237F          LAC TEMP2,3     * ACC = 1504
2024 04E4 677F          TBLR TEMP2
2025 04E5 387F          LAR 0,TEMP2    * RESTORE ARO
2026 04E6 007D          ADD ONE
2027 04E7 677F          TBLR TEMP2
2028 04E8 397F          LAR 1,TEMP2    * RESTORE ARI
2029                    *
2030 04E9 007D          ADD ONE
2031 04EA 007D          ADD ONE
2032 04EB 007D          ADD ONE
2033 04EC 677F          TBLR TEMP2
2034 04ED 6A7F          LT TEMP2        * RESTORE T REG.
2035 04EE 007D          ADD ONE
2036 04EF 677F          TBLR TEMP2     * TEMP2 = STATUS
2037 04F0 207C          LAC T3,0
2038 04F1 7F9C          PUSH           * RESTORE STACK 0
2039 04F2 6E01          LDPK 1
2040 04F3 650F          ZALH 15        * RESTORE ACCH
2041 04F4 610E          ADDS 14        * RESTORE ACCL
2042 04F5 6E00          LDPK 0
2043 04F6 7B7F          LST TEMP2      * RESTORE STATUS
2044 04F7 7F82          EINT
2045 04F8 7F8D          RET
2046                    *
2047                    *
2048                    *
2049                    *
2050                    *
2051 04F9 6880          DIVIDE LARP 0
2052 04FA 6A60          LT NUMERA
2053 04FB 6D61          MPY DENOM
2054 04FC 7F8E          PAC
2055 04FD 587B          SACH TEMSGN
2056 04FE 2061          LAC DENOM

```

```
2057 04FF 7F88      ABS
2058 0500 5061      SACL DENOM
2059 0501 6560      ZALH NUMERA
2060 0502 7F88      ABS
2061 0503 700E      LARK 0,14
2062 0504 6461      KPDVNG SUBC DENOM
2063 0505 F400      BANZ KPDVNG
      0506 0504
2064 0507 5062      SACL QUOT
2065 0508 207B      LAC TEMSGN
2066 0509 FD00      BGEZ DONE
      050A 050E
2067 050B 7F89      ZAC
2068 050C 1062      SUB QUOT
2069 050D 5062      SACL QUOT
2070 050E 7F8D      DONE RET
2071                *
2072                *****
2073                *
2074                *
2075                *
2076                *
2077                *
2078                *
2079                *
2080                END
NO ERRORS, NO WARNINGS
```


Y50	047C	1874	1878
Y61	0437	1779	1804
Y62	043D	1785	1787
Y63	044A	1799	1801
YB20	0008	0438	0432
YY11	0028	0484	0476

APPENDIX B

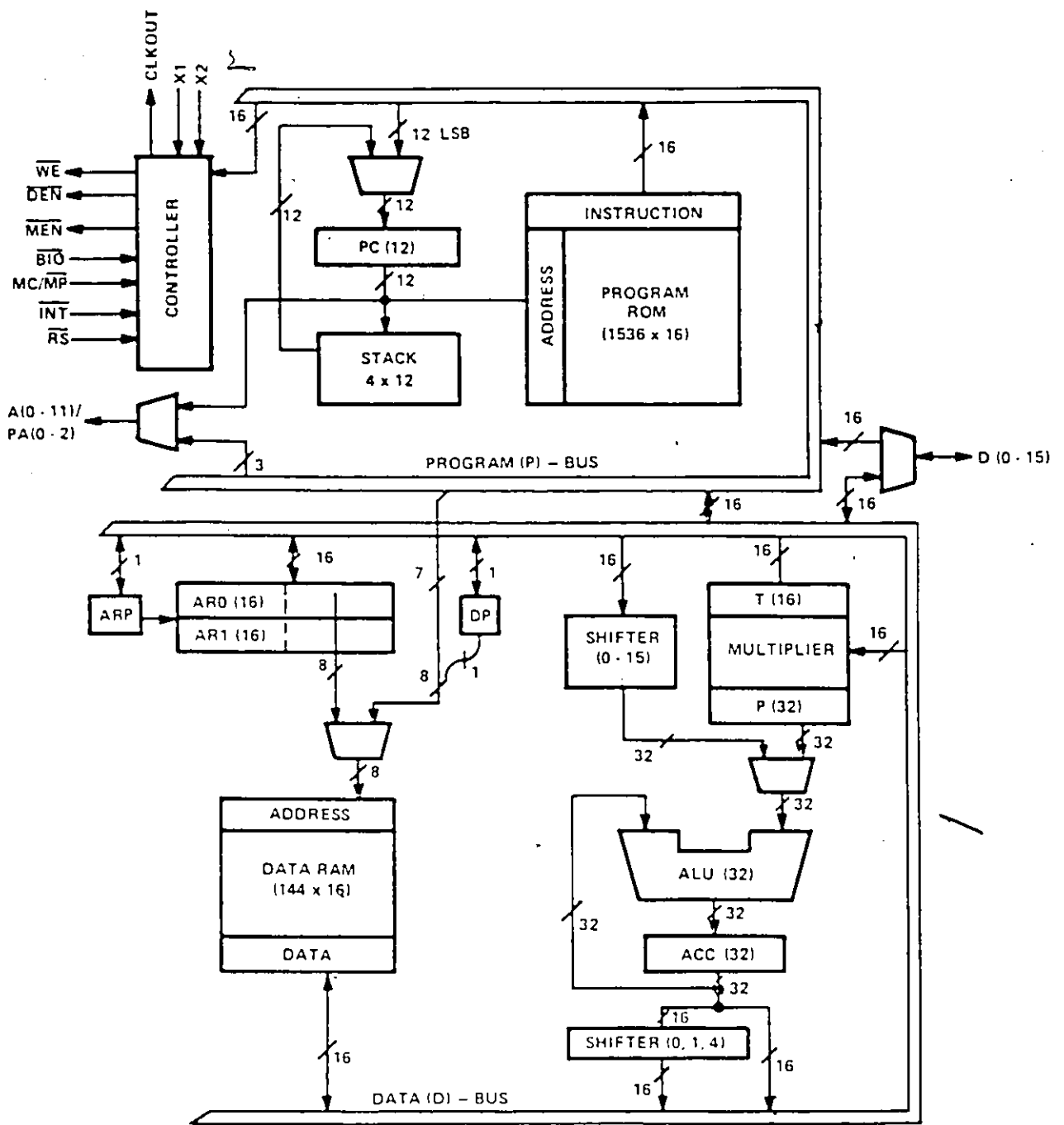


FIGURE 2.2.1

4.5 INSTRUCTION CODE SUMMARY

INST	INSTRUCTION REGISTER										CY	WD	DESCRIPTION							
	15	14	13	12	11	10	9	8	7	6				5	4	3	2	1	0	
ADD	0	0	0	0	C							S					1	1	ADD TO ACC WITH SHIFT	
SUB	0	0	0	1	C							S					1	1	SUBTRACT FROM ACC W/ SHIFT	
LAC	0	0	1	0	C							S					1	1	LOAD ACC WITH SHIFT	
SAR	0	0	1	1		R						S					1	1	STORE AUXILIARY REGISTER	
LAR	0	0	1	1		R						S					1	1	LOAD AUXILIARY REGISTER	
IN	0	1	0	0		PA						S					2	1	INPUT DATA FROM PORT	
OUT	0	1	0	0		PA						S					2	1	OUTPUT DATA TO PORT	
SACL	0	1	0	1		X						S					1	1	STORE ACC LOW	
SACH	0	1	0	1		X						S					1	1	STORE ACC HIGH WITH SHIFT	
ADDH	0	1	1	0	0	0	0	0	0			S					1	1	ADD TO ACC HIGH	
ADDS	0	1	1	0	0	0	0	0	1			S					1	1	ADD TO ACC WITH SIGN-EXTENSION SUPPRESSED	
SUBH	0	1	1	0	0	0	1	0				S					1	1	SUBTRACT FROM ACC HIGH	
SUBS	0	1	1	0	0	0	1	1				S					1	1	SUBTRACT FROM ACC WITH SIGN-EXTENSION SUPPRESSED	
SUBC	0	1	1	0	0	1	0	0				S					1	1	CONDITIONAL SUBTRACT FOR DIVIDE	
ZALH	0	1	1	0	0	1	0	1				S					1	1	ZERO ACC & LOAD HIGH	
ZALS	0	1	1	0	0	1	1	0				S					1	1	ZERO ACC & LOAD LOW WITH SIGN-EXTENSION SUPPRESSED	
TBLR	0	1	1	0	0	1	1	1				S					3	1	TABLE READ	
MAR	0	1	1	0	1	0	0	0				S					1	1	MODIFY AUXILIARY REGISTER AND POINTER	
DMOV	0	1	1	0	1	0	0	1				S					1	1	DATA SHIFT IN MEMORY	
LT	0	1	1	0	1	0	1	0				S					1	1	LOAD T-REGISTER	
LTD	0	1	1	0	1	0	1	1				S					1	1	LOAD T-REG., ACC P-REG. & SHIFT DATA MEMORY	
LTA	0	1	1	0	1	1	0	0				S					1	1	LOAD T-REG. & ACCUMULATE	
MPY	0	1	1	0	1	1	0	1				S					1	1	MULTIPLY WITH DATA WORD	
LDPK	0	1	1	0	1	1	1	1	0								1	1	LOAD DATA MEMORY PAGE IMMEDIATE	
LDP	0	1	1	0	1	1	1	1		0	0	0	0	0	0	0		1	1	LOAD DATA MEMORY PAGE
LARK	0	1	1	1	0													1	1	LOAD AUXILIARY REGISTER IMMEDIATE
XOR	0	1	1	1	1	0	0	0				S					1	1	EXCLUSIVE OR	
AND	0	1	1	1	1	0	0	1				S					1	1	LOGICAL AND	
OR	0	1	1	1	1	0	1	0				S					1	1	LOGICAL OR	
LST	0	1	1	1	1	0	1	1				S					1	1	LOAD STATUS	
SST	0	1	1	1	1	1	0	0				S					1	1	STORE STATUS	
TBLW	0	1	1	1	1	1	0	1				S					3	1	TABLE WRITE	
LACK	0	1	1	1	1	1	1	1	0								1	1	LOAD ACC IMMEDIATE	
NOP	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0		1	1	NO OPERATION
DINT	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	DISABLE INTERRUPT	
EINT	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	ENABLE INTERRUPT
ABS	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0		1	1	ABSOLUTE VALUE
ZAC	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1		1	1	ZERO ACC
ROVM	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0		1	1	RESET OVERFLOW MODE
SOVM	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1		1	1	SET OVERFLOW MODE
CALA	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0		2	1	CALL SUBROUTINE INDIRECT
RET	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1		2	1	RETURN

4.5 INSTRUCTION CODE SUMMARY (CONTINUED)

INST	INSTRUCTION REGISTER														CY	WD	DESCRIPTION		
	15	14	13	12	11	10	9	8	7	6	5	4	3	2				1	0
LARP	0	1	1	0	1	0	0	0					K				1	1	LOAD AUXILIARY REG. POINTER IMMEDIATE
PAC	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	LOAD ACC WITH P-REGISTER
APAC	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	ADD P-REGISTER TO ACC
SPAC	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	SUBTRACT P-REG. FROM ACC
MPYK	1	0	0									M					1	1	MULTIPLY IMMEDIATE
BANZ	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	2	2	BRANCH ON AUX. REG. NOT ZERO
EV	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	2	2	BRANCH ON OVERFLOW
BIOZ	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	2	2	BRANCH ON I/O ZERO
CALL	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	2	2	CALL SUBROUTINE DIRECT
B	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	2	2	BRANCH UNCONDITIONALLY
BLZ	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	2	2	BRANCH IF < 0
BLEZ	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	2	2	BRANCH IF ≤ 0
BGZ	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	2	2	BRANCH IF > 0
BGEZ	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	2	2	BRANCH IF ≥ 0
ENZ	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	2	2	BRANCH IF ≠ 0
BZ	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	2	2	BRANCH IF = 0
PUSH	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	2	1	PUSH FROM ACC TO STACK
POP	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	2	1	POP TO ACC FROM STACK

4.6 NOMENCLATURE

Instruction set nomenclature is defined in the following way:

- () = The contents of the location pointed to, e.g., (A) means the contents of data memory location at address 'A'.
- = Replacement
- :
- = Concatenation

Reference to contents of the program counter (PC) is made only when relevant, i.e., in multiword branch or call instructions, table-look-up, CALA or return instructions when the stack registers are used.

The increment of the program counter, PC + 1, due to the prefetch, is implied for every instruction.