Voice Recognition System

Project Report 1998-99

P-1345

Submitted by

ANAND KUMAR V.
ANEETHA VENU
PRAKASH N.
RAJANIKANT M
RAMANATHAN S.P

Guided by

Mrs. H. MANGALAM M.E, MISTE

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF **BACHELOR OF ENGINEERING IN ELECTRONICS & COMMUNICATION ENGINEERING** OF THE BHARATHIAR UNIVERSITY COIMBATORE

Department of Electronics & Communication Engineering

Kumaraguru College of Technology

Coimbatore-641 006

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE -641 006

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
CERTIFICATE

This is to certify that the Report entitled
"VOICE RECOGNITION SYSTEM"
has been submitted by

Ms. *Aneetha Venu,*

Mr. *Anand Kumar .V, Prakash .N, Rajanikant .M, Ramanathan .S.P*

In partial fulfillment of the requirements for the award of **Degree of Bachelor Of Engineering in the Electronics and Communication Engineering** *branch of the Bharathiar University, Coimbatore – 641 046 during the academic year 1998 – 1999.*

13/3/99
(Guide) *(Head of Department)*

Certified that the Candidate was Examined by us in the Project Work Viva – Voce Examination held on 15-3-99 *and the University Register Numbers are* 9527DO178, 9527DO176, 9527DO206, 9527DO207, 9527DO212.

15/3
(Internal Examiner)

15/3/99
(External Examiner)

DEDICATED

TO

SIR C. V. RAMAN

In remembrance of his

contribution to Science.

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Principal, **Dr. K. K. Padmanabhan,** *B.Sc. (Engg),M.Tech., Ph.D., for his unflinching support during the course of the project.*

We are highly indebted to our **Prof. M. Ramasamy,** *M.E., MIEEE (U.S.A), MISTE , Professor and Head of the Department of ELECTRONICS AND COMMUNICATION ENGINEERING for his valuable advice and timely guidance.*

We are extremely thankful to our guide **Mrs. H. Mangalam,** *M.E., MISTE, for the moral support and immense help she provided for the completion of this project.*

Last but not the least we thank all the **faculty members** *and our friends for their help and support.*

CONTENTS

CONTENTS

SYNOPSIS

SYNOPSIS

Speaker recognition nowadays is finding extensive applications in the business field. This project is aimed at implementing a possible application of speaker recognition in the form of an intelligent answering machine giving customized messages depending on the callers voice.

In this project the more conventional parametric approach to speech processing is used, which involves the comparison of the LPC coefficients of the speech obtained in real time to the speech obtained in real time to the LPC coefficients of the reference templates.

The extraction of the LPC coefficients is done, using the popular Levinson Durbin algorithm, due to its inherent advantage of stability. The comparison of the templates is done after endpoint detection of the vectors using a process called Dynamic Time Warping (DTW).

A hardware interface between the telephone line and the computer has also been developed using the CODEC chip for Analog to Digital operations. The memory and the processing constraints for this operation has been found suitable for real time. The efficiency for the process of speaker recognition has been found to be high enough for implementation although improvements could be made by using the statistical approach.

INTRODUCTION

1. INTRODUCTION

For many years, speech recognition by machine existed only in the minds of science-fiction writers. However, in recent years, the real problem of automatic speech recognition (ASR) has been addressed in many research laboratories throughout the world. The ultimate goal of this research is to produce a machine which will recognize accurately normal human speech from any speaker. Such a machine could be used in a wide variety of applications including speech input to computers , office automation , factory automation, security systems, aids for the handicapped , consumer products and almost any job involving hands-busy/eyes-busy activities.

In an ever-expanding information technology and advanced telecommunications age, speech recognition and synthesis devices would permit remote access to a wide variety of information services over the telephone.

1.1 AIM OF THE PROJECT

The aim of this project is to implement a speaker recognition system which works on the principle of Linear Predictive Coding (LPC) and Dynamic Time Warping Technique (DTW). The system extracts features from a speech segment and performs a match with an already existing template of the speaker. Depending on the results of the match the identity of the speaker is established.

BLOCK DIAGRAM

2. BLOCK DIAGRAM

This project can be divided into two modules, namely

(1) Hardware Module

(2) Software Module

which is represented in the figure 2.1.

2.1 HARDWARE MODULE

This module involves the development of an Add-on-Card, with a CODEC chip and an external switching device. Initially, when the telephone call is made, a timer is initiated. If the telephone handset is not picked up within three rings, a relay is activated which switches the telephone signal onto the Add-on-Card. The Add-on-Card is an interface between the system's parallel ports and the telephone line. It digitizes the speech signal at a sampling rate of 8Khz (set by software) into an 8 bit digital format.

2.2 SOFTWARE MODULE

This module involves the processing and coding of signals using conventional methods. This coded format of speech is compared with the reference templates and a decision as to who the caller, is taken based on the voice of the caller. Based on this decision , a personalized message which is stored in memory is conveyed to the caller.

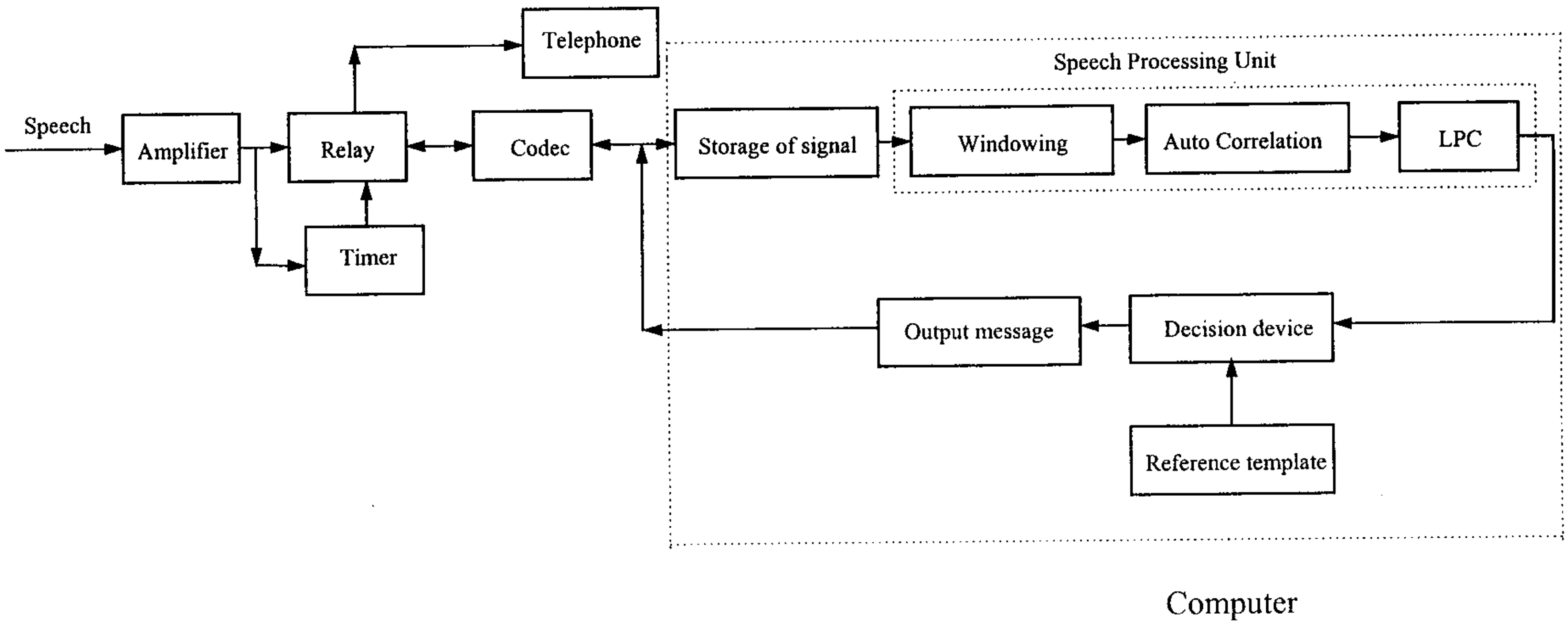


FIG 2.1 **BLOCK DIAGRAM**

HARDWARE INTERFACE

3. HARDWARE INTERFACE

The hardware interface involves two modules, represented in figure 3.1.

(1) Switching the telephone line on to the interface board.

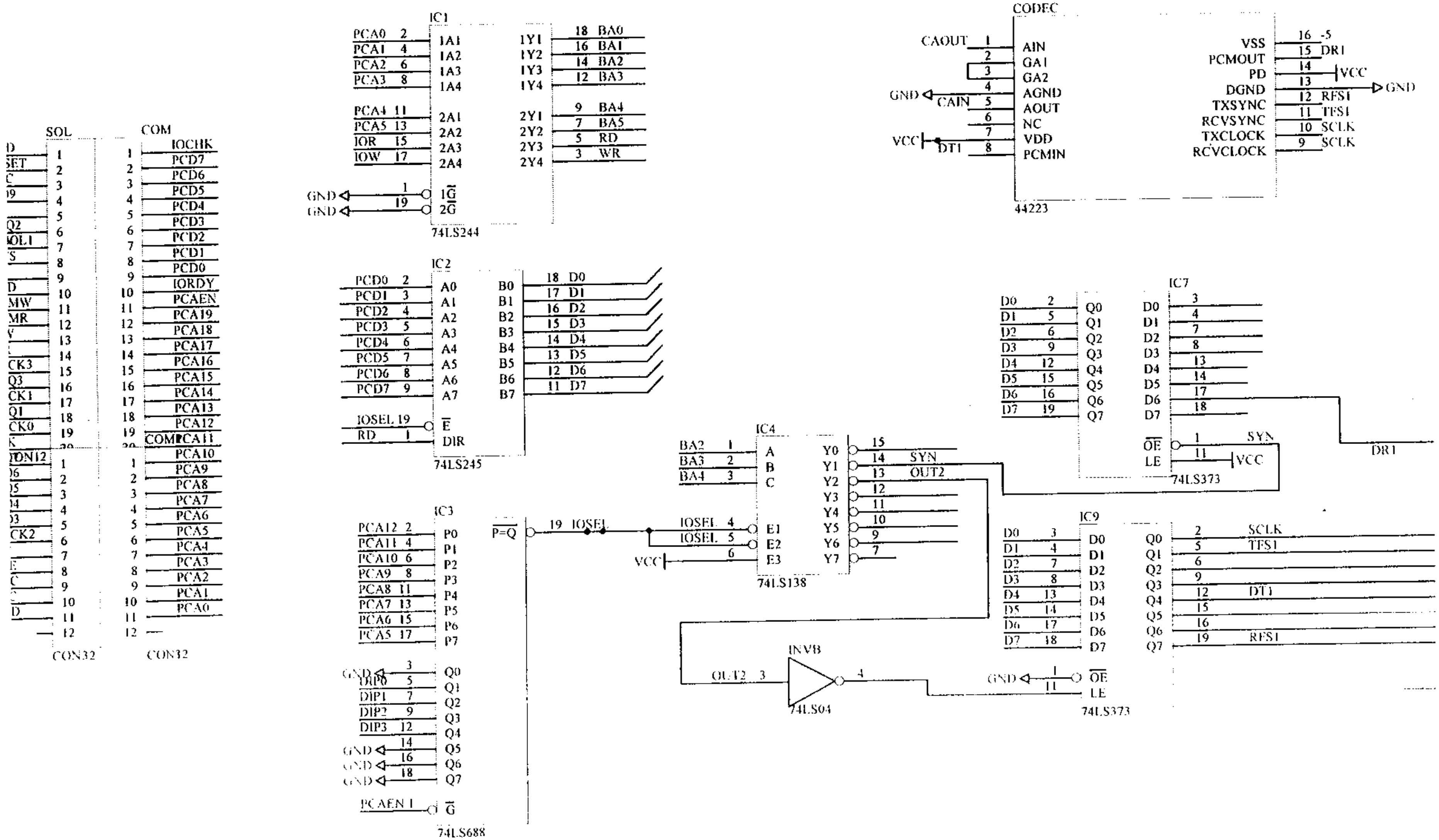
(2) Development of an interface between the telephone line and parallel port of the computer.

When a telephone call is made, a ringing tone is sent, which is detected and this activates the timer. For a time duration equal to three rings, if the telephone handset is not picked up, then the relay is activated and switches it to the Add-on-Card .

The interface board involves the development of an Add-on-Card. The system contains the 74LS244(address bus buffer) and 74LS245(data bus buffer) to exclude the parallel port of the buffer from the current variations in the rest of the circuit. The address bus is connected to the 74LS688 (comparator). The address lines A5 through A12 is compared with the address reference of the comparator which are specified through connections to the Ground and Supply. The output of the comparator is used to select the

74LS138 decoder which decodes the address lines A2, A3 and A4. The 74LS138 chooses either the input port or the output port latch (74LS373). The output latch is enabled through the 74LS04 NOT gate. The CODEC chip is connected to these latches. The clock for the CODEC chip (1.536 MHz) is provided by a software through the output port. The data is obtained in a serial format from and to the CODEC chip in serial form such that each bit of the data is transmitted for a clock pulse. The output and input operations are enabled through the transmit and receive sync pulses which are also provided by software. The conversion from Analog to Digital and vice-versa in the CODEC chip is also controlled by the transmit and receive sync pulses provided to the chip.

The data acquired by this process is stored in the form of a binary data file. Similarly, the information to be transmitted is also stored as a binary data file which is output to the telephone lines whenever the message is required.



CODEC
44223
AIN
GA1
GA2
AGND
AOUT
NC
VDD
PCMIN
VSS
PCMOUT
PD
DGND
TXSYNC
RCVSYNC
TXCLOCK
RCVCLOCK
CAOUT
CAIN
GND
VCC
DT1
DR1
RFS1
TFS1
SCLK
IC1
74LS244
IC2
74LS245
IC3
74LS688
IC4
74LS138
IC7
74LS373
IC9
74LS373
INVB
74LS04
SYN
OUT2
IOSEL
SOL
COM
CON32

```
/* ASSEMBLY  LANGUAGE  PROGRAM  */

#include<stdio.h>
#include<dos.h>
#include<stdlib.h>

void main()
{
 FILE *ptr;
int i,*addint;
char *address;
char k;
address= 0x6000;
addint=0x6000;
top:
ptr=fopen("data.dat","w+");
asm{
mov bx,[0x6000];
mov ch,08h;
mov dx,308h;
mov ax,0000h;
out dx,ax;
push cx;
pop cx;
nop;
nop;
}
loop:

asm{
mov ax,0001h;
out dx,ax;
mov ax,0000h;
nop;
nop;
nop;
nop;
nop;
nop;
```

```
out dx,ax;
mov ax,0002h;
out dx,ax;
nop;
nop;
mov ax,0001h;
out dx,ax;
mov dx,304h;
in ax,dx;
mov bx,ax;
mov dx,308;
mov ax,0000h;
out dx,ax
inc bx;
inc bx;
nop;
nop;
nop;
dec ch;
jnz loop;
}
fclose(ptr);
goto top;
}
```

/* ASSEMBLY LANGUAGE PROGRAM */

```
#include<stdio.h>
#include<dos.h>
#include<stdlib.h>

void main()
{
 FILE *ptr;
 int i;
char *address,*crap;
char k;
address= 0x6000;
crap=0x6100;
top:
ptr=fopen("data.dat","r+");
for(i=0;i<=250;i++)
{
fscanf(ptr,"%c",&k);
*address=k;
address++;
}
asm{
mov bx,0x6000
mov ch,08h;
mov dx,308h;
mov ax,0000h;
out dx,ax;
push cx;
pop cx;
nop;
nop;
}
loop:

asm{
mov ax,0001h;
out dx,ax;
mov ax,0000h;
```

```
nop;
nop;
nop;
nop;
nop;
nop;
out dx,ax;
mov ax,0080h;
out dx,ax;
nop;
nop;
mov ax,0001h;
out dx,ax;
mov ax,[bx];
mov cl,10h;
and al,cl;

//  remove start
mov [0x6100],ax;
}

printf(" the read value is=%c",*crap);
getchar();

asm{

// end remove

nop;
nop;
nop;
nop;
nop;
out dx,ax;
inc bx;
//inc bx;
nop;
nop;
nop;
dec ch;
```

```
jnz loop;
}
fclose(ptr);
goto top;
}
```

SOFTWARE

4. SOFTWARE

The software module consists of processing the incoming speech signal and then extracting certain features in order to compare the incoming speech signal with the stored reference templates so as to identify the speaker. For this the following operations are performed.

4.1 WINDOWING

The process of truncating the infinite impulse response of a system to a finite set of values is known as windowing. It is called so, as it is equivalent to looking at the samples of the system through a finite size window.

4.2 NEED FOR WINDOWING

- *The speech signal is not known for all the time and it is impossible and impractical to compute the infinite summations required to obtain the autocorrelation values.*
- *It is necessary to re-calculate a new set of coefficients every 10 to 30ms (we use 30ms intersecting windows) in order to reflect the changing nature of the speech signal.*

4.3 TYPES OF WINDOWS

The various types of windows are given below.

- *Rectangular window*
- *Hanning window*
- *Kaiser window*
- *Hamming window*

(1) Rectangular Window

The time domain representation is given by

$W_R(n) = 1 , \quad -M \leq n \leq M$

$\quad\quad 0 , \quad \text{otherwise}$

(2) Hanning Window

It is also known as Raised Cosine Window.

The time domain representation is given by

$W_H'(n) = 0.5 + 0.5 \, Cos(2\Pi n/(N-1))$

(3) Kaiser Window

It is also known as I_o-Sinh window.

The time domain representation is given by

$W_K(n) = \{I_o\, [\alpha[((N-1)/2)^2-(n-((N-1)/2))^2\,]^{1/2}]\, \}/I_o\, [\alpha\, [1/2]\,]$

where $0 \leq n \leq N-1$

(4) Hamming Window

It is also known as Raised Cosine Window with a platform. The time domain representation is given by

$W_H(n) = 0.54 + 0.46\ Cos(2\Pi n/(N-1))$

The comparison of the time domain representations of the various windows is given in figure 4.1.

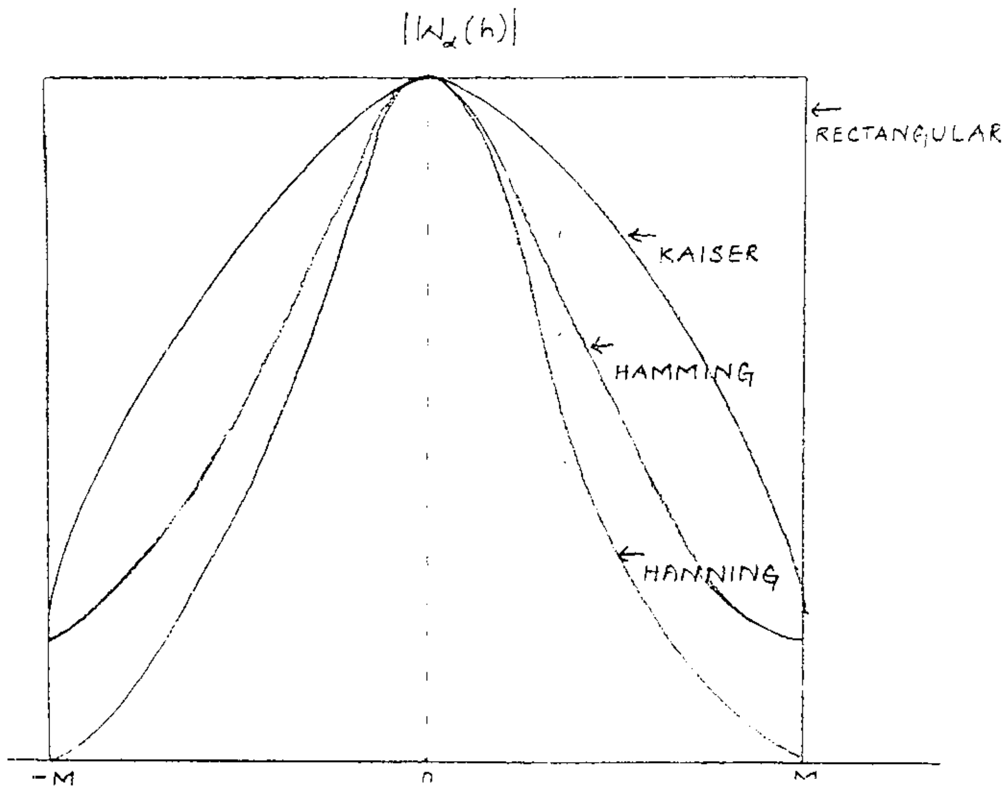


FIGURE : 4.1 COMPARISON OF TIME DOMAIN REPRESENTATION OF THE VARIOUS WINDOWS

As far as speech analysis is considered , we require a **soft window function** *in order to reduce the prediction error at the beginning and end of the speech segment. Large prediction errors will arise at the start of the interval (o ≤ n ≤ p-1) since the predictor is effectively being required to predict the signal from samples which have arbitrarily been set to zero, while at the end of the interval (N ≤ n ≤ N+p-1) it is endeavouring to predict zero signal from samples that are non-zero. Thus we choose the Hamming Window.*

4.4 WINDOWING PERIOD

A window of duration 30ms i.e. 240 samples at a 8khz rate, is used and overlapping of adjacent windows is normally used to give an overall frame rate in the range 10-20ms. Here we use a intersection of 100 samples between adjacent windows.

4.5 CODING TECHNIQUES

There are a number of coding techniques which can be used to code the signal after windowing. They are as follows.

- *Short time Fourier analysis*
- *Sub- band Coding*
- *Adaptive Transform Coding*
- *Linear Predictive Coding*
- *Adaptive Predictive Coding*

Among the various coding techniques mentioned above, we go in for Linear Predictive Coding.

4.6 LINEAR PREDICTIVE CODING (LPC)

One of the most powerful speech analysis techniques is the method of linear predictive analysis. Linear prediction gives us an aspect of time series analysis and also an approach in modeling dynamic systems. For example

LPC finds its application in neurophysics , geophysics , speech communication, etc. This method has become the predominant technique for estimating the basic speech parameters, example pitch formats, spectra, vocal tract area functions , and for representing speech for low bit rate transmission or storage.

The importance of this method lies both in its ability to provide extremely accurate estimation of the speech parameters and in its relative speed of computation.

The basic idea behind linear predictive analysis is that a speech sample can be approximated as a linear combination of past speech samples. The application of linear predictive analysis to estimate speech parameters is often called linear predictive coding (LPC).

By minimising the sum of the squared distances (over a finite interval) between the actual speech samples and the linear predicted ones , a unique set of predictor coefficients can be determined.

The reason for choosing LPC is as follows :

- It overcomes disadvantages found in other coding methods
- It is the most efficient way of obtaining data compression.

- *It provides a robust , reliable and accurate method for estimating the parameters that characterize the linear time-varying system*
- *It is a very important and powerful speech processing technique used for speech synthesis, speech recognition and s peech coding.*

4.7 METHODS OF LPC PARAMETERS EXTRACTION

As applied to speech processing the term linear prediction refers to a variety of essentially equivalent formulations of the problem of modelling the speech waveform. The three different methods of formulation of LPC parameters are as follows :

- *Autocorrelation method*
- *Covariance method*
- *Lattice or Burg method*

Among the three methods mentioned above, the Autocorrelation method is used since it guarantees the stability of the resulting filter.

BASIC IDEA

The basic idea behind this method is that sample values of speech , x[n] , can be approximated as a linear combination of the past p speech samples as shown in the figure 4.2.

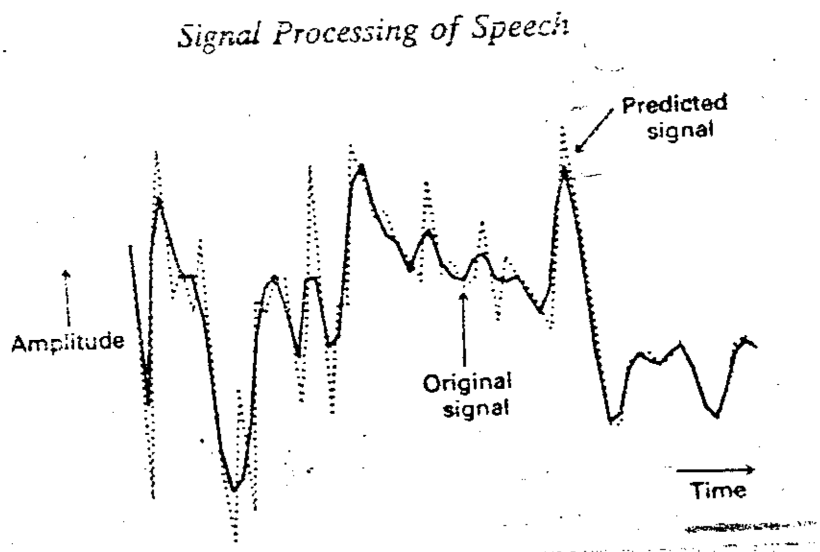


FIGURE : 4.2 GRAPHICAL INTERPRETATION OF LINEAR PREDICTION

EQUATION

The linear predictor is described mathematically, by the following equation

$$X[n] = a_1x[n-1] + a_2x[n-2] + \ldots + a_px[n-p]$$

$$= \sum_{k=1 \text{ to } p} a_kx[n-k]$$

where $X[n]$ *is the Predicted sample at instant n and*

$a_1, a_2, \ldots, a_p$ *are Predictor Coefficients*

PREDICTION ERROR

It is generally impossible to predict each signal sample exactly and this leads to an error known as Prediction Error. The prediction error $e[n]$ *at each sample instant is given by*

$$e[n] = x[n] - X[n]$$

By minimising the mean squared error between the actual speech samples and the linearly-predicted ones, the predictor coefficients (that is, the weighting coefficients of the linear combination) can be determined by solving a set of linear equations. A set of predictor coefficients can predict the speech signal reasonably accurately over stationary portions . In order to match the time-varying properties of the speech signal, a new set of predictor coefficients are calculated every 10-30 ms.

The problem in linear prediction is to determine the a_k coefficients so as to minimize the mean square error, E, over a specified number of samples.

$$E = \sum_n e^2[n] = \sum_n \{ x[n] - X[n] \}^2$$

$$= \sum_n \{ x[n] - \sum_{k=1 \text{ to } p} a_k . x[n-k] \}^2$$

If E is to be minimized by the appropriate choice of the a_k coefficients, then the partial derivative of E with respect to each coefficient a_j, $j=1,2,\ldots,p$ should be zero, that is

$$\delta E/\delta a = -2\Sigma x[n-j].[x[n]-\Sigma_{k=1 \text{ to } p} a_k x[n-k]]=0$$

$x[n-k] \} = 0$

So,

$$\sum_{k=1 \text{ to } p} a_k \sum_n x[n-j].\ x[n-k] = \sum_n x[n].x[n-j]$$

$j=1,2,\ldots,p.$

The above equation represents a set of p linear equations for the p unknowns a_k. Therefore, it should be possible to find a solution by matrix inversion. However, finding the solution to a system of equations in perhaps 10-15 unknowns is not a trivial problem even if the equations are linear. We solve this system by applying autocorrelation method.

Suppose initially that we assume that the signal is stationary with finite energy, which of course is not the case for speech, and the range of summation is from $-\infty$ to $+\infty$ with $x[n]$ being defined as zero for $n<o$, then

$$\sum_{n=-\infty \text{ to } \infty} x[n-j].x[n-k] = \sum_{n=-\infty \text{ to } \infty} x[n-j+1].\ x[n-k+1] = \ldots$$

$$= \sum_{n=-\infty \text{ to } \infty} x[n].\ x[n+j-k]$$

Therefore the system of equations can be written as

$$\sum_{k=1 \text{ to } p} a_k \sum_{n=-\infty \text{ to } \infty} x[n].\ x[n+j-k] = \sum_{n=-\infty \text{ to } \infty} x[n].\ x[n-j]$$

$$j = 1,2,3,\ldots,p$$

The multipliers of the a_k coefficients and the right hand sides of the system of equations are in the form of autocorrelation values of the speech signal for specific time (sample) shifts. If $R(k)$ is defined as the autocorrelation value for a shift of k samples, that is

$$R(k) = \sum_{n=-\infty \text{ to } \infty} x[n] \, . \, x[n+k]$$

The system of equations can be written as

$$\begin{vmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(0) & \ldots & R(p-2) \\ . & . & \ldots & . \\ . & . & \ldots & . \\ R(p-1) & R(p-2) & \ldots & R(0) \end{vmatrix} \quad \begin{vmatrix} a_1 \\ a_2 \\ . \\ . \\ a_p \end{vmatrix} \quad \begin{matrix} = \\ = \\ = \\ = \\ = \end{matrix} \quad \begin{vmatrix} R(1) \\ R(2) \\ . \\ . \\ R(p) \end{vmatrix}$$

The above matrix is symmetric and all the diagonal elements are the same. It is known as a Toeplitz matrix. The above matrix can be solved efficiently by

Durbin-Levinson method. The Durbin-Levinson method requires much less computational effort than is generally needed for solving a system of linear equations. Of course, the speech signal is not known for all time and it is impossible and impractical to compute the infinite summations required to obtain the autocorrelation values. In addition, it is necessary to re-calculate a new set of coefficients, a_k, every 10 – 30 ms to reflect the changing nature of the speech signal, and hence short- time autocorrelation values are used. These are computed by first multiplying the speech signal $x[n]$ by a soft window function, $w[n]$ (for example, a Hamming window), of duration N samples as shown in fig.4.3 and the autocorrelation values are calculated from

$$R(k) = \sum_{n=0 \text{ to } N-1} \{w[n].x[n]\}.\{w[n+k].x[n+k]\},$$

$K=0,1,2,\ldots,p$ & N = No: of samples

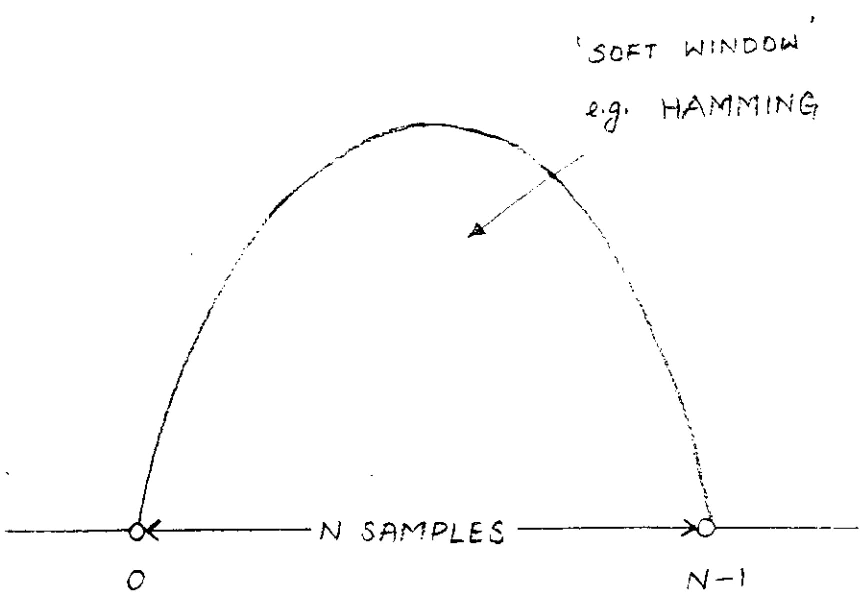


FIGURE:4.3 ILLUSTRATION OF ANALYSIS INTERVALS FOR AUTOCORRELATION METHOD

In speech analysis this method of computing the predictor coefficients has become known as the autocorrelation method because of the presence of autocorrelation values in the system of equations for solving the predictor coefficients. Typically, a window of duration 20 – 30 ms (240 samples at 8khz rate) is used and overlapping of windows is normally used to give an overall frame rate in the range 10 – 20 ms. A soft window function is essential in order to reduce the prediction error at the beginning and the end

Much of the computational effort in speech pattern matching is in deriving a near optimal alignment function. This can be achieved by a technique called dynamic time-warping(DTW) as discussed below.

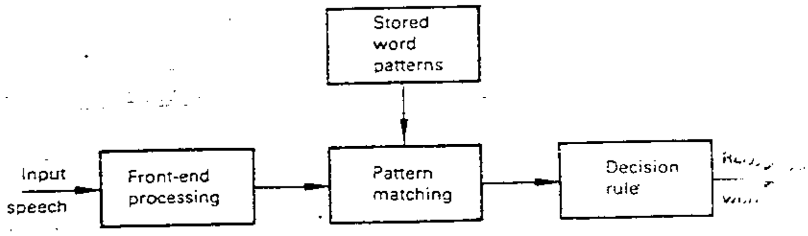


FIGURE:4.4 BASIC ISOLATED-WORD RECOGNITION SYSTEM

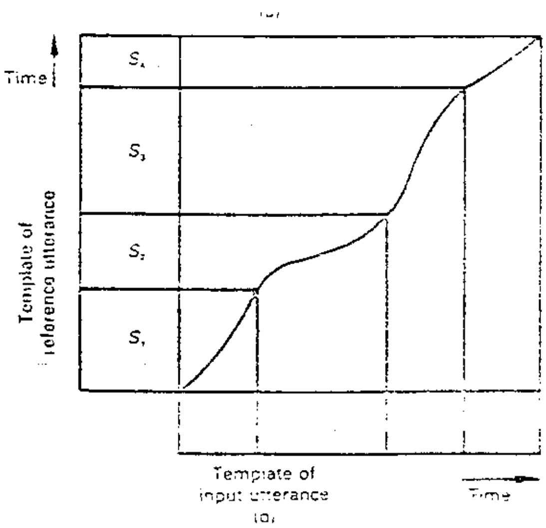


FIGURE:4.5(a) ILLUSTRATION OF LINEAR TIME ALIGNMENT OF UTTERANCES

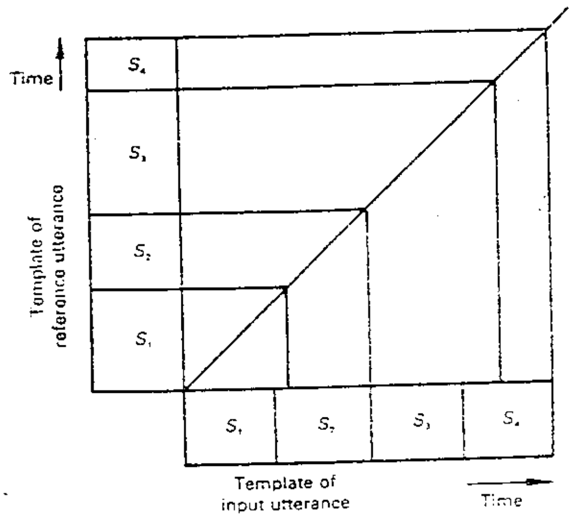


FIGURE:4.5(6) ILLUSTRATION OF NON-LINEAR TIME ALIGNMENT (DYNAMIC TIME-WARPING)

Let $P_1(n)$,n=1,2,...N and $P_2(m)$, m=1,2,...M denote the two patterns to be matched .P_1 and P_2 consist of N and M frames respectively , of a multidimensional feature vector. The time- alignment of these two patterns is done by illustrating the patterns as one-dimensional functions as shown in the fig.4.6. It has been assumed that both the beginning and the end points of the two patterns have been accurately determined. The goal is to find an alignment curve which maps $P_1(n)$ onto the corresponding parts of $P_2(m)$. In other words, an alignment function of the form m=w(n) is required , which relates the n time axis of P_1 to m time axis of P_2. The constrained beginning and end points in fig3 can formally be expressed as constraints on w(n) , that is $w(1)=1$ and $w(N)=M$. For any two patterns, $w(n)$ is that function or path which minimises the distance between them. In mathematical terms, $w(n)$ can be determined by solving the optimization problem expressed as

$$D^* = Min\{\sum_{n=1 \text{ to } N} d[P_1(n), P_2(w(n)]\}$$

where $d[P_1(n), P_2(w(n)]$, is the distance between frame n of P_1 and frame $w(n)$ of P_2 and D^* is the accumulated distance between P_1 and P_2 over the optimal path $w(n)$. The solution to this problem may be found out by a technique called dynamic programming which was first applied in the design of optimal control systems.

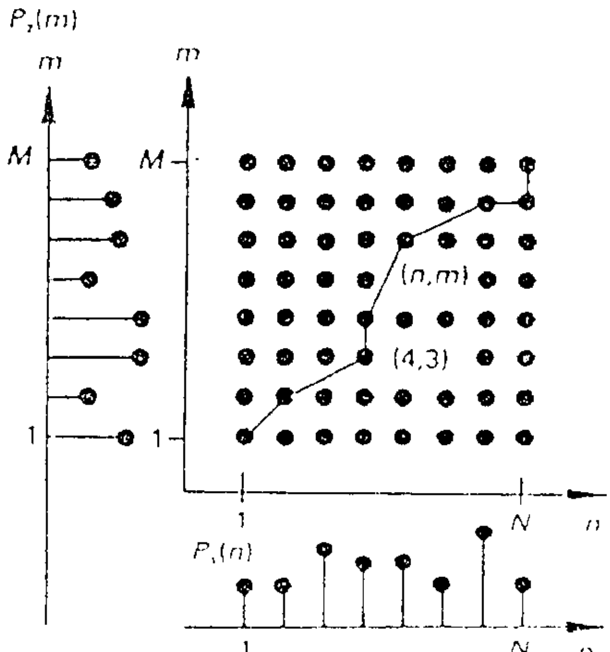


FIGURE:4.6 TIME-ALIGNMENT OF TWO ONE-DIMENSIONAL PATTERNS

Dynamic programming is a technique used in multi-stage decision processes where decisions are made at discrete time-intervals during the course of some transaction in which the outcome of the transaction is a function of all the decisions. The set of rules for making decisions throughout a particular process is called a policy ; and a policy which results in an optimal process is called an optimal policy. Fundamental to the method of Dynamic programming is the 'principle of optimality' which states – 'An optimal policy has the property that ,whatever the initial state and initial decision may be , the remaining decisions must constitute an optimal policy with respect to the state resulting from the first decision'. In other words the last part of an optimal process is itself an optimal process. The principle of optimality applied to pattern matching framework of fig4.6. may be stated as follows – ' If the best path from the grid-point (1,1) to the general grid-point (n,m) goes through the grid-point (4,3) , for example , then the best path from (1,1) to (4,3) is part of the best path from (1,1) to (n,m)'. In other words, the optimum global path can be obtained by always locally choosing the path which minmises the distances between the two patterns.

In addition to the endpoint constraints mentioned above, a number of other restrictions are placed on the time-alignment path. For example, it would generally be unrealistic to permit a path which resulted in excessive compression or expansion of the time-scale. Another obvious restriction is that the slope of the path can never be negative, that is time-order must be preserved by making $w(n)$ increase monotically. These local constraints are incorporated by specifying the full path in terms of simple local paths which may be pieced together to form larger paths. The path to the grid-point (n,m) can only come from one of three points, that is $(n,m-1)$,$(n-1,m-1)$ or $(n-1,m)$.

If $D_A(n,m)$ denotes the minimum accumulated distance along any path from the grid-point (1,1) to the grid-point (n,m), $D_A(n,m)$ can be computed recursively using dynamic programming , that is

$$D_A(n,m)=d[P_1(n),P_2(m)]+Min\{D_A(n,m-1),D_A(n-1,m-1),D_A(n-m)\} \text{ ----(1)}$$

The above equation indicates that the minimum accumulated distance from the grid-point (1,1) to the grid-point (n,m) consists of the local distance between frame n of pattern P_1 and frame m of pattern P_2 plus the minimum of the accumulated distances to the grid-points $(n,m-1)$,$(n-1,m-1)$ and $(n-1,m)$. The above equation forms the basis of a dynamic time-warping algorithm for pattern matching in speech recognition. Initially $D_A(1,1)$ is set equal to $d[P_1(1),P_2(1)]$, that is the local distance between the first frames of of the segment. Large prediction errors will arise at the start of the interval ($0 \leq n \leq p-1$) since the predictor is effectively being required to predict the signal from samples which have arbitrarily been set to zero, while at the end of the interval ($N \leq n \leq N+p-1$) it is endeavouring to predict zero signal from samples that are non-zero.

4.8 PATTERN MATCHING

The function of the pattern matching block of the isolated-word speech recognition system in fig4.4 is to determine the similarity between the input word pattern and the stored word patterns. This involves not only distance computation but also time alignment of the input and the reference patterns because a word spoken on different occasions, even by the same speaker, will exhibit both local and global variation in its time-scale. The simplest method of time-aligning two patterns of unequal length is to map the time-axis of one onto the time-axis of the other in a linear fashion as shown in fig 4.5(a). It is clear that this method is not entirely satisfactory since it does not guarantee that the internal parts of the patterns will be properly aligned although it does give proper alignment of the beginning and end of the patterns. What is really required is an alignment function which properly matches the internal features of the pattern as shown in fig 4.5(b).

the two patterns. Then $D_A(n,m)$ is computed sequentially for all values of n and m in the range n=1 to N and m=1 to M. Finally, the accumulated distance D^* over the optimal path is given by $D_A(N,M)$. The optimum time-alignment path w(n) can be determined by backtracking from the end of the path , although for most speech recognition applications only the optimal accumulated distance is required and computation of the actual warping path is of no benefit.

The process of Dynamic Time Warping is clearly depictedin figure 4.7

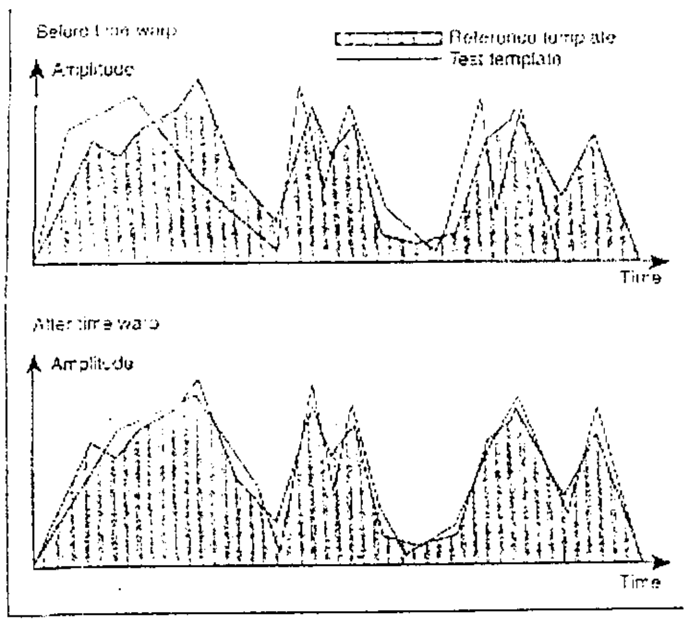
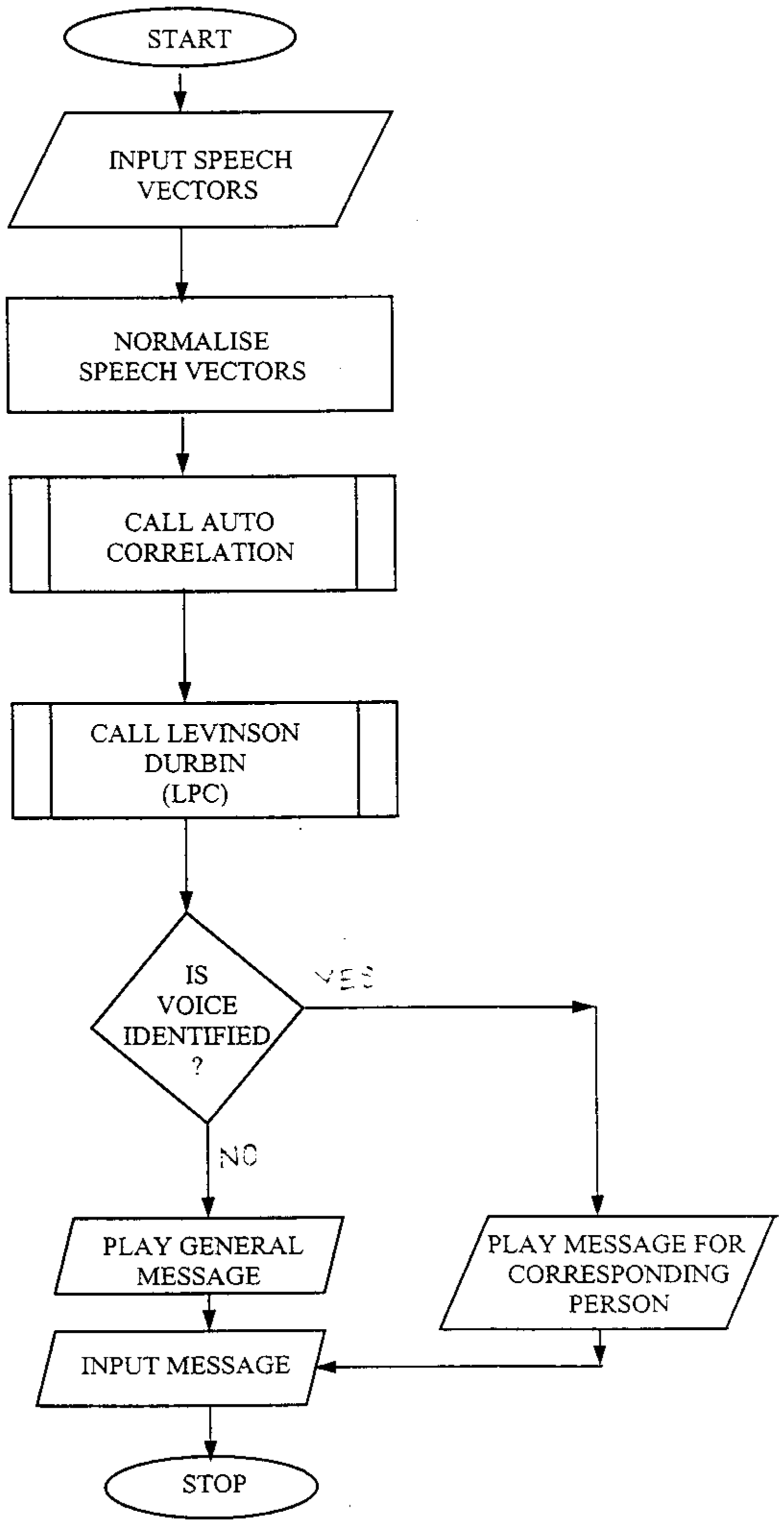


FIGURE 4.7 PROCESS OF DTW

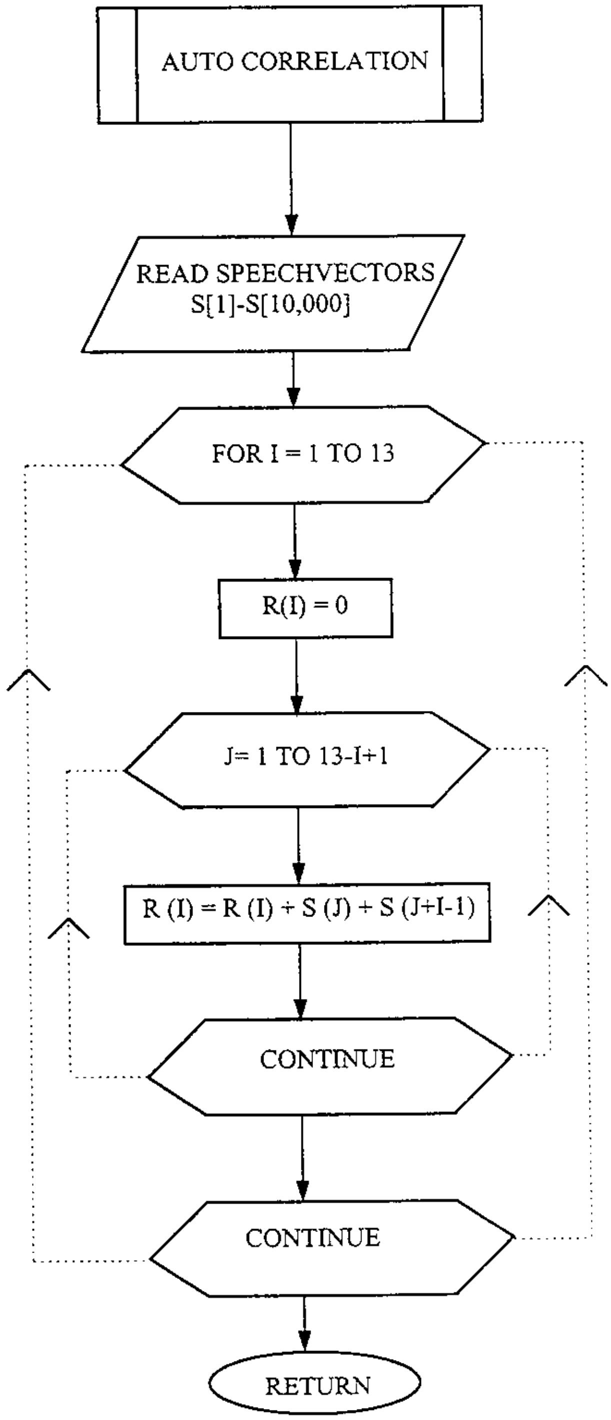
FLOWCHART



START
INPUT SPEECH VECTORS
NORMALISE SPEECH VECTORS
CALL AUTO CORRELATION
CALL LEVINSON DURBIN (LPC)
IS VOICE IDENTIFIED ?
YES
NO
PLAY GENERAL MESSAGE
PLAY MESSAGE FOR CORRESPONDING PERSON
INPUT MESSAGE
STOP

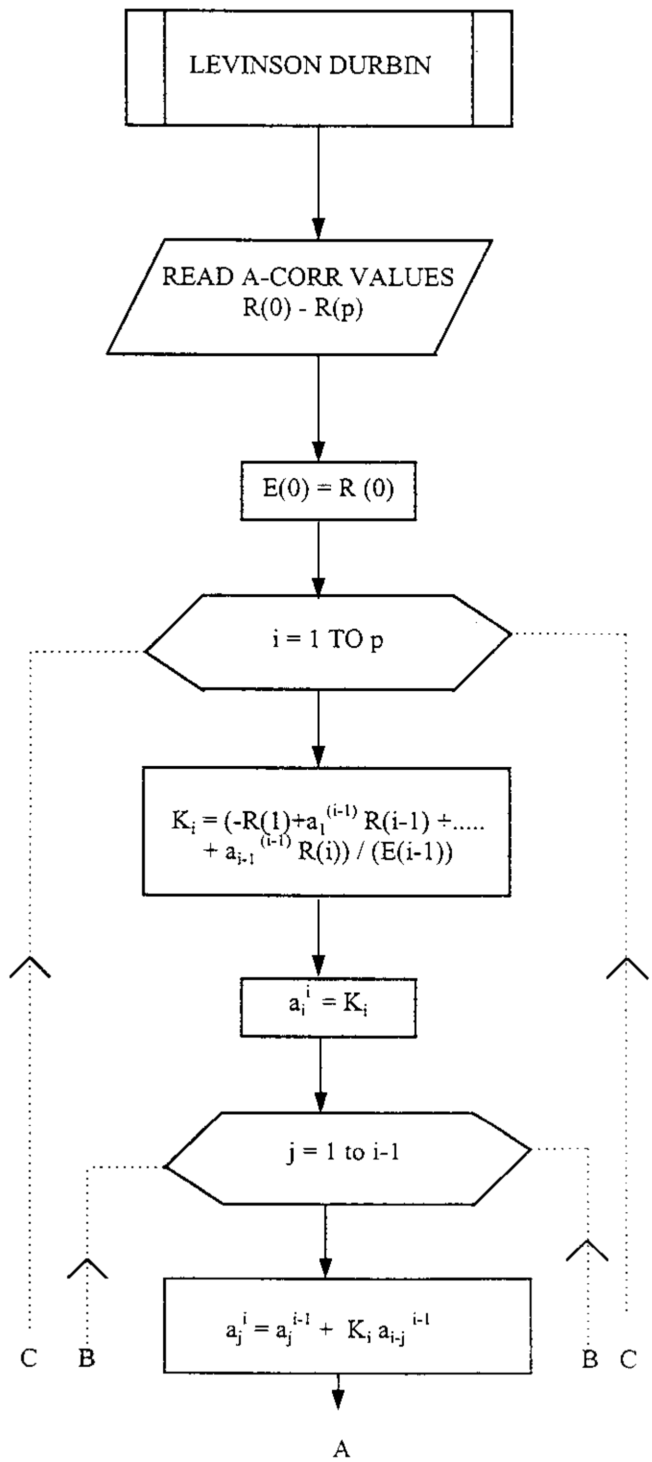
AUTO CORRELATION

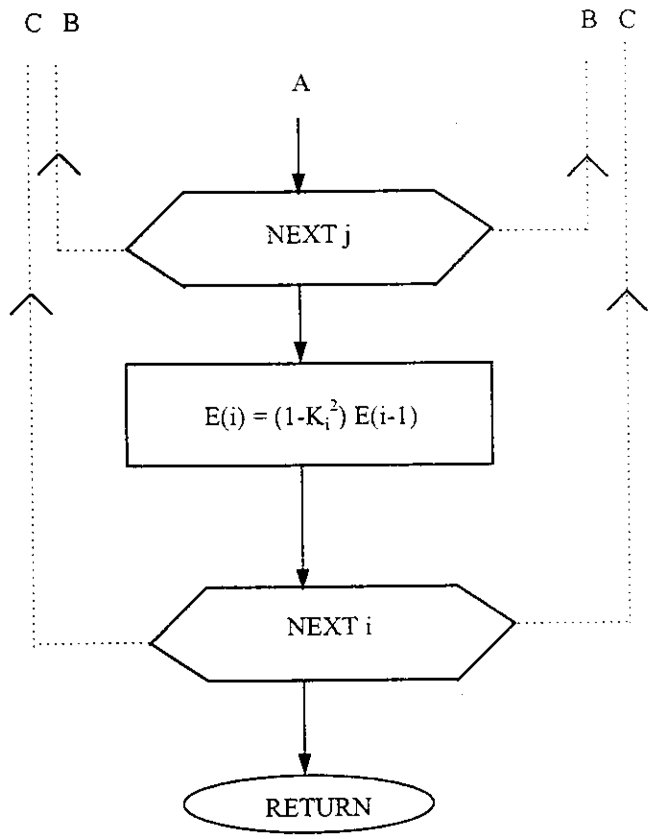
R(I) = AUTOCORRELATION COEFFICIENTS, S(I) = SPEECH VECTORS



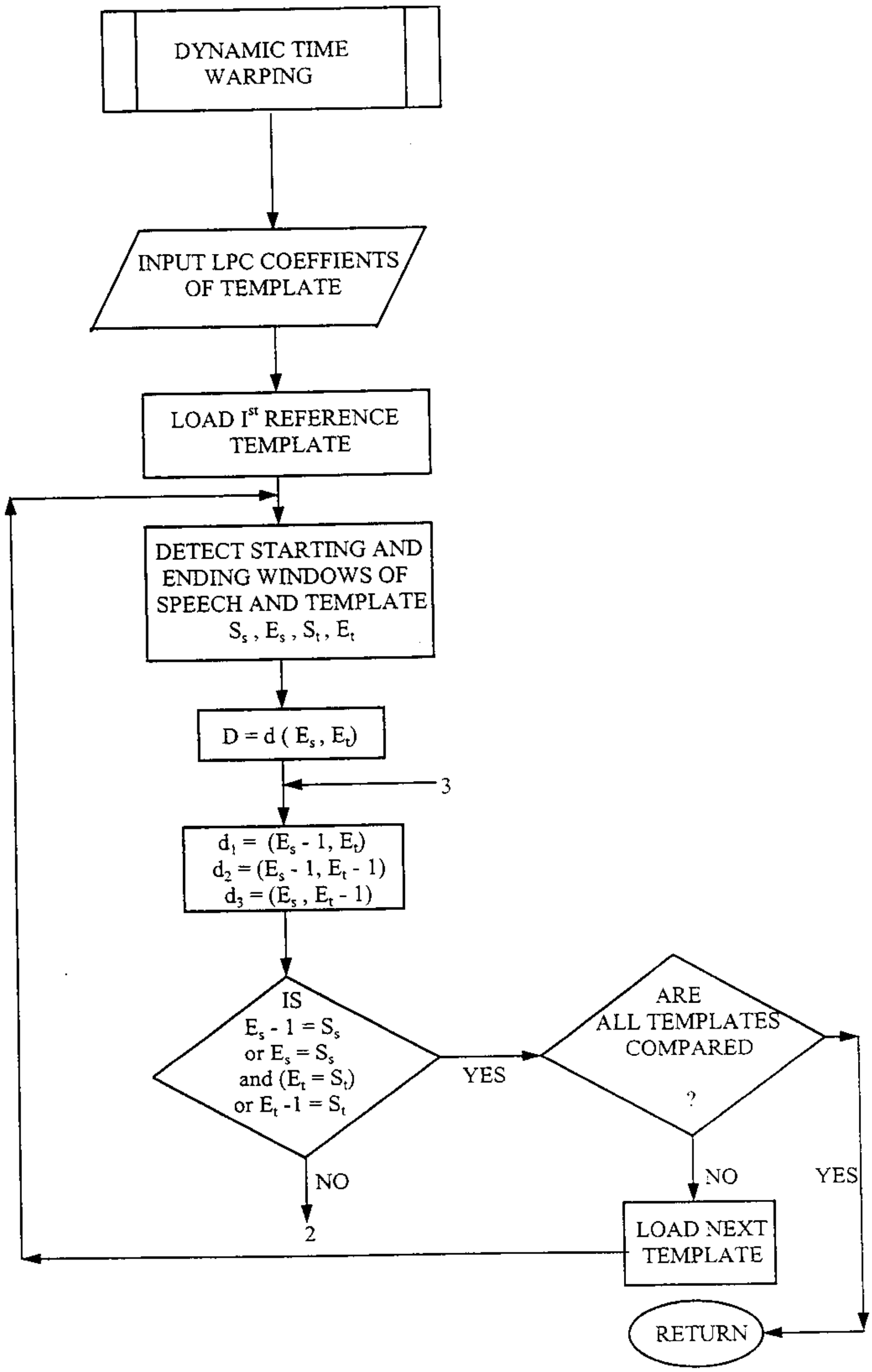
LEVINSON DURBIN ALGORITHM FOR CALCULATING LPC

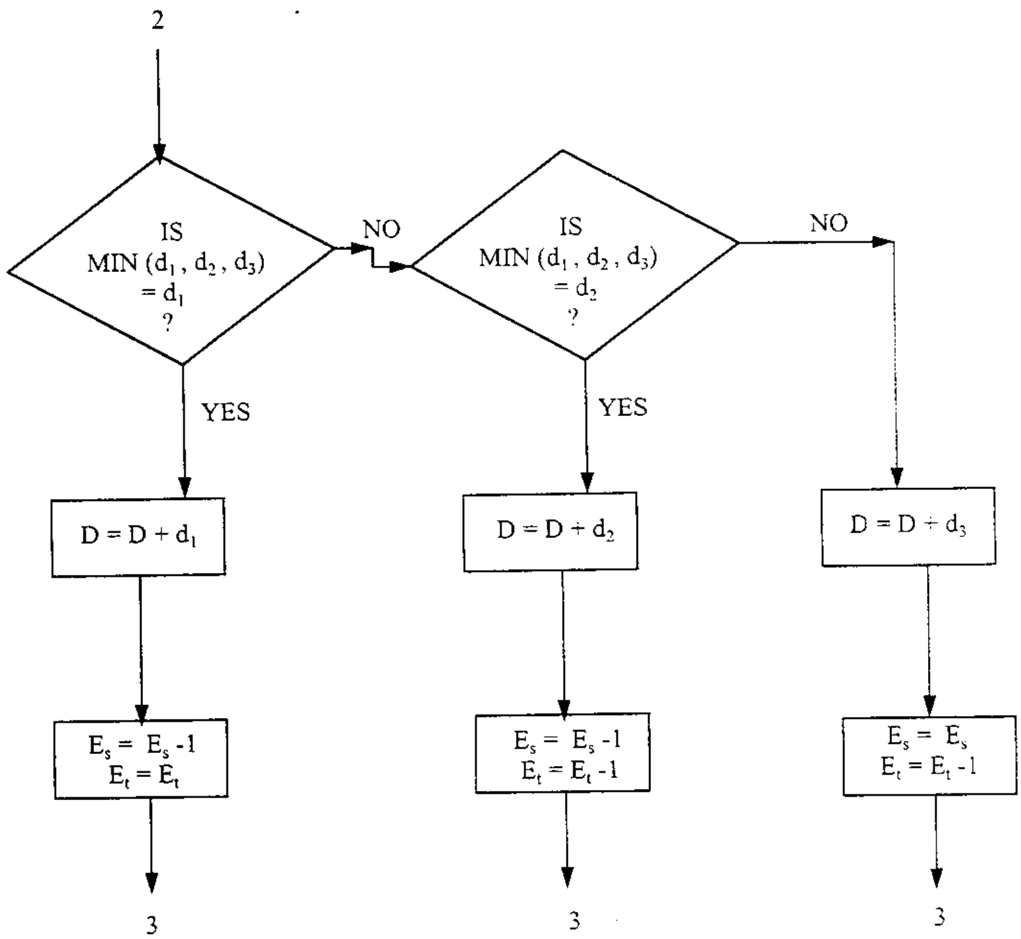
R(I) = AUTOCORRELATION COEFFICIENTS, K(I) = REFLECTION COEFFICIENT
a_j^i = LPC values, p = 12 (12^{th} order LPC)




C B
B C
A
NEXT j
E(i) = (1-K_i^2) E(i-1)
NEXT i
RETURN

DYNAMIC TIME WARPING




2
IS MIN (d_1, d_2, d_3) = d_1 ?
NO
IS MIN (d_1, d_2, d_3) = d_2 ?
NO
YES
YES
$D = D + d_1$
$D = D + d_2$
$D = D + d_3$
$E_s = E_s - 1$
$E_t = E_t$
$E_s = E_s - 1$
$E_t = E_t - 1$
$E_s = E_s$
$E_t = E_t - 1$
3
3
3

/* FEATURE EXTRACTION FOR THE REFERENCE TEMPLATE */

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<dos.h>
#define pi 3.14
void main()
{
int a;
void ham_gen();
void hamming();
void autos();
void lpcs();
ham_gen();
hamming();
autos();
lpcs();
printf(" program ends:");
}
```

/* FUNCTION 1: TO GENERATE HAMMING WINDOW */

```
void ham_gen()
{
FILE *wind;
int i,a;
float h[240];
wind=fopen("ham.raj","w+");
for(i=1;i<=240;i++)
        {
        h[i]=0.54-((1-0.54)*(cos(2*pi*(i-1)/240)));
        fprintf(wind,"%f",h[i]);
        }
fclose(wind);
return;
}
```

/* END OF FUNCTION 1 */

/* FUNCTION 2:TO MULTIPLY SPEECH AND HAMMING WINDOW COEFFICIENTS */

```
void hamming()

{
      int v,p=0,u,s=0,m=240;
      int i,j,x,counter=0;
      float a[10000],maximum;
      float  k,c,f[250],h[250];
      FILE *p1,*wind,*wind_vals;
      p1=fopen("sn2.dat","r+");
      for(i=1;i<=9000;i++)
      {
      fscanf(p1,"%f",&a[i]);
      }
      maximum=abs(a[1]);
      for(i=2;i<=9000;i++)
      {
      if(abs(a[i])>maximum)
      {
      maximum=abs(a[i]);
      }
      }

 for(i=1;i<=9000;i++)
 wind=fopen("ham.raj","r+");
 for(i=1;i<=240;i++)
             {
              fscanf(wind,"%f",&h[i]);
             }
  wind_vals=fopen("windowtsp.raj","w+");
  w:   for(v=s;v<=m;v++)
             {
             for(u=1;u<=240;u++)
             {
              if( (v-(p*140))-u==0)
                      { k=a[v]*h[u];
                       fprintf(wind_vals,"%f",k);
```

```
            counter++;
                    }
            }
    }

        s=140;m=m+140;p=p+1;
        if(m<=9000)
                {
                fflush(stdin);
                goto w;
                }
        else
                goto q;
    q:      fclose(wind);
    fclose(p1);
    fclose(wind_vals);
    return;
}
```

/* END OF FUNCTION 2 */

/* FUNCTION 3:TO FIND AUTOCORRELATION FOR ALL WINDOWS */

```
void autos()
{
    int z,i,k,j,count=1,mov=1,counter=1;
    float s[14000],r[13],maxi;
    FILE *autocorr,*wind_vals;
    wind_vals=fopen("windowtsp.raj","r+");
    autocorr=fopen("acortsp.raj","w+");
for(z=1;z<=57;z++)
 {
      for(i=mov;i<mov+240;i++)
            {
            fscanf(wind_vals,"%f",&s[i]);
            }
an:  for (k=1;k<=13;k++)
            {
            r[k]=0;
```

```
              for (j=1;j<=240-k+1;j++)
                     r[k]=r[k]+s[j]*s[j+k-1];
             }
             maxi=abs(r[1]);
for (k=2;k<=13;k++)
             {
             if (abs(r[i])>maxi)
              maxi=abs(r[i]);
           }
 for(k=1;k<=13;k++)
  {
  r[k]=r[k]/maxi;
  }
  for(k=1;k<=13;k++)
  {
  fprintf(autocorr,"%f",r[k]);
  }
     count=count+1;
     fflush(stdin);
      }
 fclose(wind_vals);
 fclose(autocorr);
 return;
}

 /* END OF FUNCTION 3 */


 /* FUNCTION 4:TO FIND LPC COEFFICIENTS */

void lpcs()
{
 int i,j,l,p=12,x,coef,counter;
 float k[14],a[12][12],E[14],A=0;
 float coer,r[14];
 FILE *lpcs,*autocorr;
 autocorr=fopen("acortsp.raj","r+");
 lpcs=fopen("lpctsp.raj","w+");
```

```
for(counter=1;counter<=57;counter++)
{
     for (i=0;i<=p;i++)
     {
      fscanf(autocorr,"%f",&r[i]);
      }
a1:E[0]=r[0];
for(i=1;i<=p;i++)
{
     A=0;
     coef=(i-1);
     for(j=1;j<=(i-1);j++)
             {
             if ((i-1)>0)
                     {
                     A=A+(a[j][i-1]*r[coef]);
                             coef=coef-1;
                     }
             }
     k[i]=(-1)*(r[i]+A)/E[i-1];
     a[i][i]=k[i];
     for(j=1;j<=(i-1);j++)
             {
                     if((i-1)>0)
                             {
                             a[j][i]=a[j][i-1]+(k[i]*a[(i-j)][(i-1)]);
                             }
             }
     E[i]=(1-k[i]*k[i])*(E[i-1]);
}
     i=p;
     for(j=1;j<i;j++)
      {
      fprintf(lpcs,"%f",a[j][i]);
      }
      for(j=1;j<i;j++)
      {
      }
```

```
fflush(stdin);
}
fclose(lpcs);
fclose(autocorr);
return;
}

/* END OF FUNCTION 4 */
```

```
/* FEATURE EXTRACTION FOR THE INPUT SPEECH SIGNAL */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<dos.h>
#define pi 3.14
void main()
{
 int a;
 void ham_gen();
 void hamming();
 void autos();
 void lpcs();
 ham_gen();
 hamming();
 autos();
 lpcs();
 printf(" program ends:");
 }

/* FUNCTION 1: TO GENERATE HAMMING WINDOW */

void ham_gen()
{
 FILE *wind;
 int i,a;
 float h[240];
      wind=fopen("ham.raj","w+");
       for(i=1;i<=240;i++)
       {
              h[i]=0.54-((1-0.54)*(cos(2*pi*(i-1)/240)));
              fprintf(wind,"%f",h[i]);
       }
       fclose(wind);
       return;
}

/* END OF FUNCTION 1 */
```

/* FUNCTION 2:TO MULTIPLY SPEECH AND HAMMING WINDOW COEFFICIENTS */

```
void hamming()
{
      int v,p=0,u,s=0,m=240;
      int i,j,x,counter=0;
      float a[10000],maximum;
      float  k,c,f[250],h[250];
      FILE *p1,*wind,*wind_vals;
      p1=fopen("sn2.dat","r+");
      for(i=1;i<=9000;i++)
      {
       fscanf(p1,"%f",&a[i]);
      }
      maximum=abs(a[1]);
      for(i=2;i<=9000;i++)
      {
      if(abs(a[i])>maximum)
      {
       maximum=abs(a[i]);
      }
      }
    for(i=1;i<=9000;i++)
    wind=fopen("ham.raj","r+");
              for(i=1;i<=240;i++)
              {
              fscanf(wind,"%f",&h[i]);
              }
    wind_vals=fopen("windows.raj","w+");
    w:   for(v=s;v<=m;v++)
         {
          for(u=1;u<=240;u++)
                 {
                  if( (v-(p*140))-u==0)
                          { k=a[v]*h[u];
                            fprintf(wind_vals,"%f",k);
```

```
        counter++;
                        }
                }
        }

                s=140;m=m+140;p=p+1;
                if(m<=9000)
                        {
                        fflush(stdin);
                        goto w;
                        }
                else
                         goto q;
        q:      fclose(wind);
 fclose(p1);
 fclose(wind_vals);
 return;
 }
```

/* END OF FUNCTION 2 */

/* FUNCTION 3:TO FIND AUTOCORRELATION FOR ALL WINDOWS */

```
void autos()
{
      int z,i,k,j,count=1,mov=1,counter=1;
      float s[14000],r[13],maxi;
      FILE *autocorr,*wind_vals;
      wind_vals=fopen("windows.raj","r+");
      autocorr=fopen("acors.raj","w+");
for(z=1;z<=57;z++)
{
        for(i=mov;i<mov+240;i++)
              {
              fscanf(wind_vals,"%f",&s[i]);
              }
  an:  for (k=1;k<=13;k++)
              {
```

```
            r[k]=0;
             for (j=1;j<=240-k+1;j++)
                    r[k]=r[k]+s[j]*s[j+k-1];
            }
            maxi=abs(r[1]);
 for (k=2;k<=13;k++)
            {
            if (abs(r[i])>maxi)
             maxi=abs(r[i]);
           }
 for(k=1;k<=13;k++)
  {
  r[k]=r[k]/maxi;
  }
  for(k=1;k<=13;k++)
  {
  fprintf(autocorr,"%f",r[k]);
  }
     count=count+1;
     fflush(stdin);
 }
     fclose(wind_vals);
     fclose(autocorr);
     return;
}

/* END OF FUNCTION 3 */


/* FUNCTION 4:TO FIND LPC COEFFICIENTS */

void lpcs()
{
 int i,j,l,p=12,x,coef,counter;
 float k[14],a[12][12],E[14],A=0;
 float coer,r[14];
 FILE *lpcs,*autocorr;
 autocorr=fopen("acors.raj","r+");
```

```
lpcs=fopen("lpcs.raj","w+");
for(counter=1;counter<=57;counter++)
{
      for (i=0;i<=p;i++)
      {
       fscanf(autocorr,"%f",&r[i]);
    }
a1:E[0]=r[0];
for(i=1;i<=p;i++)
{
      A=0;
      coef=(i-1);
      for(j=1;j<=(i-1);j++)
               {
               if ((i-1)>0)
                        {
                        A=A+(a[j][i-1]*r[coef]);
                                 coef=coef-1;
                        }
               }
      k[i]=(-1)*(r[i]+A)/E[i-1];
      a[i][i]=k[i];
      for(j=1;j<=(i-1);j++)
               {
                        if((i-1)>0)
                                 {
                                 a[j][i]=a[j][i-1]+(k[i]*a[(i-j)][(i-1)]);
                                 }
               }
      E[i]=(1-k[i]*k[i])*(E[i-1]);
 }
      i=p;
      for(j=1;j<i;j++)
      {
      fprintf(lpcs,"%f",a[j][i]);
      }
      for(j=1;j<i;j++)
      {
```

```
        }
 fflush(stdin);
 }
 fclose(lpcs);
 fclose(autocorr);
 return;
}

/* END OF FUNCTION 4 */
```

/* PROGRAM TO CALCULATE DTW - MAIN PROGRAM */

```
#include<stdio.h>
#include<stdlib.h>
#include "dynasp.c"
#include "dynar.c"
#include "dynaav.c"
void main()
{
 float dsp,dr,dav;
 dsp=finalsp();
 dr=finalr();
 dav=finalav();
 printf("dsp=%f\n,dr=%f,dav=%f",dsp,dr,dav);
}
```

```
/* PROGRAM TO CALCULATE DTW - SUBPROGRAM */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
float temps[700];
float signs[700];
int n,m,starts,ends,startt,endt;
float d1sp(int,int);
float dminsp();
float finalsp();

/* FUNCTION BEGINS */
float finalsp()
{
 FILE *temp,*signal;
 int i,j;
 float dist;
 void findingsp();
 void findingtempsp();
 temp=fopen("lpctsp.raj","r+");
 signal=fopen("lpcs.raj","r+");
 for (i=0;i<=683;i++)
 {
      fscanf(temp,"%f",&temps[i]);
      fscanf(signal,"%f",&signs[i]);
      if((i>=200)&&(i<250))
      {
      printf("\n temp[%d]=%f\tsigns[%d]=%f",i,temps[i],i,signs[i]);
      }
           }
    fflush(stdin);
    findingsp();
    findingtempsp();
    n=endt;m=ends;
    dist=d1sp(n,m);
```

```
do
{
dist=dist+dminsp();
}while((n!=startt)&&(m-1!=starts)&&(n-1!=startt)&&(m!=starts));
if((n==startt)||(m-1==starts)||(n-1==startt)||(m==starts))
{
dist=dist+d1sp(startt,starts);
}
return (dist);
}


/* FUNCTION TO CALCULATE THE DISTANCE OF 2 WINDOWS */
float d1sp(n,m)
int n;
int m;
{
 int i,j;
 float dist=0,sum=0,zigma=0;
 for(i=0;i<12;i++)
 {
 zigma=(temps[(n*12+i)]-signs[(m*12+i)])*(temps[(n*12+i)]-
                                              signs[(m*12+i)]);
 sum=sum+zigma;
     }
 dist=sqrt(sum);
 return(dist);
 }

 /* FUNCTION TO CALCULATE THE DISTANCE & FIND THE MINIMUM OF
 3 VALUES */
 float dminsp()
 {
 int tm=0,tn=0;
 float dist1=0,dist2=0,dist3=0,min=0;
 tm=m;tn=n;
 dist1=d1sp(n,m-1);
 dist2=d1sp(n-1,m-1);
 dist3=d1sp(n-1,m);
```

```
min=dist1;
m=tm-1;
n=tn;
if(dist2<min)
{
min=dist2;
n=tn-1;
m=tm-1;
}
if(dist3<min)
{
min=dist3;
n=tn-1;
m=tm;
}
return(min);
}
```

/* FUNCTION TO FIND THE STARTING AND ENDING POINTS OF INPUT SPEECH SIGNAL */

```
void findingsp()
{
float speech,e[2],sum[57];
int i,j;
FILE *wind_sigs;
e[1]=0;
wind_sigs=fopen("windows.raj","r+");
for(i=0;i<57;i++)
sum[i]=0;
for(j=0;j<=240;j++)
{
fscanf(wind_sigs,"%f",&speech);
sum[0]=sum[0]+(speech*speech);
}
e[0]=sum[0];
for (i=1;i<57;i++)
{
```

```
for(j=0;j<240;j++)
{
fscanf(wind_sigs,"%f",&speech);
sum[i]=sum[i]+(speech*speech);
}
if (sum[i]>e[1])
e[1]=sum[i];
}
for(j=0;j<57;j++)
{
if((sum[j+1]-sum[j])>.1)
{
starts=j;
break;
}
}
for(j=56;j>=0;j--)
{
if((sum[j-1]-sum[j])>.1)
{
ends=j;
break;
}
}
return;
}

/* FUNCTION TO FIND STARTING AND ENDING POINTS OF TEMPLATE */

void findingtempsp()
{
float speech,e[2],sum[57];
int i,j;
FILE *wind_sigs;
e[1]=0;
wind_sigs=fopen("windowtsp.raj","r+");
for(i=0;i<57;i++)
 sum[i]=0;
```

```
for(j=0;j<=240;j++)
{
fscanf(wind_sigs,"%f",&speech);
sum[0]=sum[0]+(speech*speech);
}
e[0]=sum[0];
for (i=1;i<57;i++)
{
 for(j=0;j<240;j++)
{
fscanf(wind_sigs,"%f",&speech);
sum[i]=sum[i]+(speech*speech);
}
if (sum[i]>e[1])
e[1]=sum[i];
}
for(j=0;j<57;j++)
{
if((sum[j+1]-sum[j])>.1)
{
startt=j;
break;
}
}
for(j=56;j>=0;j--)
{
if((sum[j-1]-sum[j])>.1)
{
 endt=j;
 break;
 }
 }
 return;
 }
```

CONCLUSION

5. CONCLUSION

In this project, a sincere attempt has been made, to effectively use the power of speech processing , for an unique application, which is bound to have immense practical value. Due to time and technical constraints, the efficiency of this system is limited to around 80% accuracy. The implementation of more complex approaches , like 'Hidden Markov Model' and 'Neural Networks' is bound to increase the efficiency to around 95% to 99%. This shall make the project, a commercially marketable product in the near future.

APPENDIX

recommended operating conditions

PARAMETER		SN54LS' MIN	SN54LS' NOM	SN54LS' MAX	SN74LS' MIN	SN74LS' NOM	SN74LS' MAX	UNIT
V_{CC}	Supply voltage (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.7			0.8	V
I_{OH}	High-level output current			−12			−15	mA
I_{OL}	Low-level output current			12			24	mA
T_A	Operating free-air temperature	−55		125	0		70	°C

NOTE 1: Voltage values are with respect to network ground terminal.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		SN54LS' MIN	SN54LS' TYP‡	SN54LS' MAX	SN74LS' MIN	SN74LS' TYP‡	SN74LS' MAX	UNIT
V_{IK}		V_{CC} = MIN, I_I = −18 mA				−1.5			−1.5	V
Hysteresis ($V_{T+} - V_{T-}$)		V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = MAX, I_{OH} = −3 mA		2.4	3.4		2.4	3.4		V
V_{OH}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.5 V, I_{OH} = MAX		2			2			V
V_{OL}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = MAX	I_{OL} = 12 mA			0.4			0.4	V
V_{OL}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = MAX	I_{OL} = 24 mA						0.5	V
I_{OZH}		V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = MAX	V_O = 2.7 V			20			20	μA
I_{OZL}		V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = MAX	V_O = 0.4 V			−20			−20	μA
I_I		V_{CC} = MAX, V_I = 7 V				0.1			0.1	mA
I_{IH}		V_{CC} = MAX, V_I = 2.7 V				20			20	μA
I_{IL}		V_{CC} = MAX, V_{IL} = 0.4 V				−0.2			−0.2	mA
I_{OS}§		V_{CC} = MAX		−40		−225	−40		−225	mA
I_{CC}	Outputs high	V_{CC} = MAX, Output open	All		17	27		17	27	mA
I_{CC}	Outputs low	V_{CC} = MAX, Output open	'LS240		26	44		26	44	mA
I_{CC}	Outputs low	V_{CC} = MAX, Output open	'LS241, 'LS244		27	46		27	46	mA
I_{CC}	All outputs disabled	V_{CC} = MAX, Output open	'LS240		29	50		29	50	mA
I_{CC}	All outputs disabled	V_{CC} = MAX, Output open	'LS241, 'LS244		32	54		32	54	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	TEST CONDITIONS		'LS240 MIN	'LS240 TYP	'LS240 MAX	'LS241, 'LS244 MIN	'LS241, 'LS244 TYP	'LS241, 'LS244 MAX	UNIT
t_{PLH}	R_L = 667 Ω, See Note 2	C_L = 45 pF,		9	14		12	18	ns
t_{PHL}	R_L = 667 Ω, See Note 2	C_L = 45 pF,		12	18		12	18	ns
t_{PZL}	R_L = 667 Ω, See Note 2	C_L = 45 pF,		20	30		20	30	ns
t_{PZH}	R_L = 667 Ω, See Note 2	C_L = 45 pF,		15	23		15	23	ns
t_{PLZ}	R_L = 667 Ω, See Note 2	C_L = 5 pF,		10	20		10	20	ns
t_{PHZ}	R_L = 667 Ω, See Note 2	C_L = 5 pF,		15	25		15	25	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

recommended operating conditions

	PARAMETER	SN54S' MIN	SN54S' NOM	SN54S' MAX	SN74S' MIN	SN74S' NOM	SN74S' MAX	UNIT
V_{CC}	Supply voltage, (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.8			0.8	V
I_{OH}	High-level output current			−12			−15	mA
I_{OL}	Low-level output current			48			64	mA
	External resistance between any input and V_{CC} or ground			40			40	kΩ
T_A	Operating free-air temperature (see Note 3)	−55		125	0		70	°C

NOTES: 1. Voltage values are with respect to network ground terminal.
3. An SN54S241J operating at free-air temperature above 116°C requires a heat sink that provides a thermal resistance from case to free-air $R_{\theta CA}$, of not more than 40°C/W.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†			SN54S' MIN	SN54S' TYP‡	SN54S' MAX	SN74S' MIN	SN74S' TYP‡	SN74S' MAX	UNIT
V_{IK}		V_{CC} = MIN,	I_I = −18 mA				−1.2			−1.2	V
Hysteresis ($V_{T+} - V_{T-}$)		V_{CC} = MIN			0.2	0.4		0.2	0.4		V
V_{OH}		V_{CC} = MIN, I_{OH} = −1 mA	V_{IH} = 2 V,	V_{IL} = 0.8 V,				2.7			V
V_{OH}		V_{CC} = MIN, I_{OH} = −3 mA	V_{IH} = 2 V,	V_{IL} = 0.8 V,	2.4	3.4		2.4	3.4		V
V_{OH}		V_{CC} = MIN, I_{OH} = MAX	V_{IH} = 2 V,	V_{IL} = 0.5 V,	2			2			V
V_{OL}		V_{CC} = MIN, I_{OL} = MAX	V_{IH} = 2 V,	V_{IL} = 0.8 V,			0.55			0.55	V
I_{OZH}		V_{CC} = MAX, V_{IL} = 0.8 V,	V_{IH} = 2 V,	V_O = 2.4 V			50			50	μA
I_{OZL}		V_{CC} = MAX, V_{IL} = 0.8 V,	V_{IH} = 2 V,	V_O = 0.5 V			−50			−50	μA
I_I		V_{CC} = MAX,	V_I = 5.5 V				1			1	mA
I_{IH}		V_{CC} = MAX,	V_I = 2.7 V				50			50	μA
I_{IL}	Any A	V_{CC} = MAX,	V_I = 0.5 V				−400			−400	μA
I_{IL}	Any G	V_{CC} = MAX,	V_I = 0.5 V				−2			−2	mA
I_{OS}§		V_{CC} = MAX			−50		−225	−50		−225	mA
I_{CC}	Outputs high	V_{CC} = MAX,	Outputs open	'S240		80	123		80	135	mA
I_{CC}	Outputs high	V_{CC} = MAX,	Outputs open	'S241, 'S244		95	147		95	160	mA
I_{CC}	Outputs low	V_{CC} = MAX,	Outputs open	'S240		100	145		100	150	mA
I_{CC}	Outputs low	V_{CC} = MAX,	Outputs open	'S241, 'S244		120	170		120	180	mA
I_{CC}	Outputs disabled	V_{CC} = MAX,	Outputs open	'S240		100	145		100	150	mA
I_{CC}	Outputs disabled	V_{CC} = MAX,	Outputs open	'S241, 'S244		120	170		120	180	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5$ V, $T_A = 25°C$

PARAMETER	TEST CONDITIONS		'S240			'S241, 'S244			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	$R_L = 90\ \Omega$, See Note 4	$C_L = 50$ pF,		4.5	7		6	9	ns
t_{PHL}				4.5	7		6	9	ns
t_{PZL}				10	15		10	15	ns
t_{PZH}				6.5	10		8	12	ns
t_{PLZ}	$R_L = 90\ \Omega$, See Note 4	$C_L = 5$ pF,		10	15		10	15	ns
t_{PHZ}				6	9		6	9	ns

NOTE 4: See General Information Section for load circuits and voltage waveforms

'LS682, 'LS684, 'LS686, 'LS688

recommended operating conditions

	SN54LS'			SN74LS'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	µA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†			SN54LS'			SN74LS'			UNIT
						MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage					2			2			V
V_{IL}	Low-level input voltage							0.7			0.8	V
$V_{T+} - V_{T-}$	Hysteresis	P or Q inputs	V_{CC} = MIN				0.4			0.4		V
V_{IK}	Input clamp voltage		V_{CC} = MIN,	I_I = −18 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IL} = V_{IL}max,	V_{IH} = 2 V, I_{OH} = −400 µA		2.5			2.7			V
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max		I_{OL} = 12 mA		0.25	0.4		0.25	0.4	V
					I_{OL} = 24 mA					0.35	0.5	
I_I	Input current at maximum input voltage	Q inputs, 'LS682	V_{CC} = MAX,	V_I = 5.5 V				0.1			0.1	mA
		All other inputs	V_{CC} = MAX,	V_I = 7 V								
I_{IH}	High-level input current		V_{CC} = MAX,	V_I = 2.7 V				20			20	µA
I_{IL}	Low-level input current	Q inputs, 'LS682	V_{CC} = MAX,	V_I = 0.4 V				−0.4			−0.4	mA
		All other inputs						−0.2			−0.2	
I_{OS}§	Short-circuit output current		V_{CC} = MAX,	V_O = 0		−20		−100	−20		−100	mA
I_{CC}	Supply current	'LS682	V_{CC} = MAX,	See Note 2			42	70		42	70	mA
		'LS684					40	65		40	65	
		'LS686					44	75		44	75	
		'LS688					40	65		40	65	

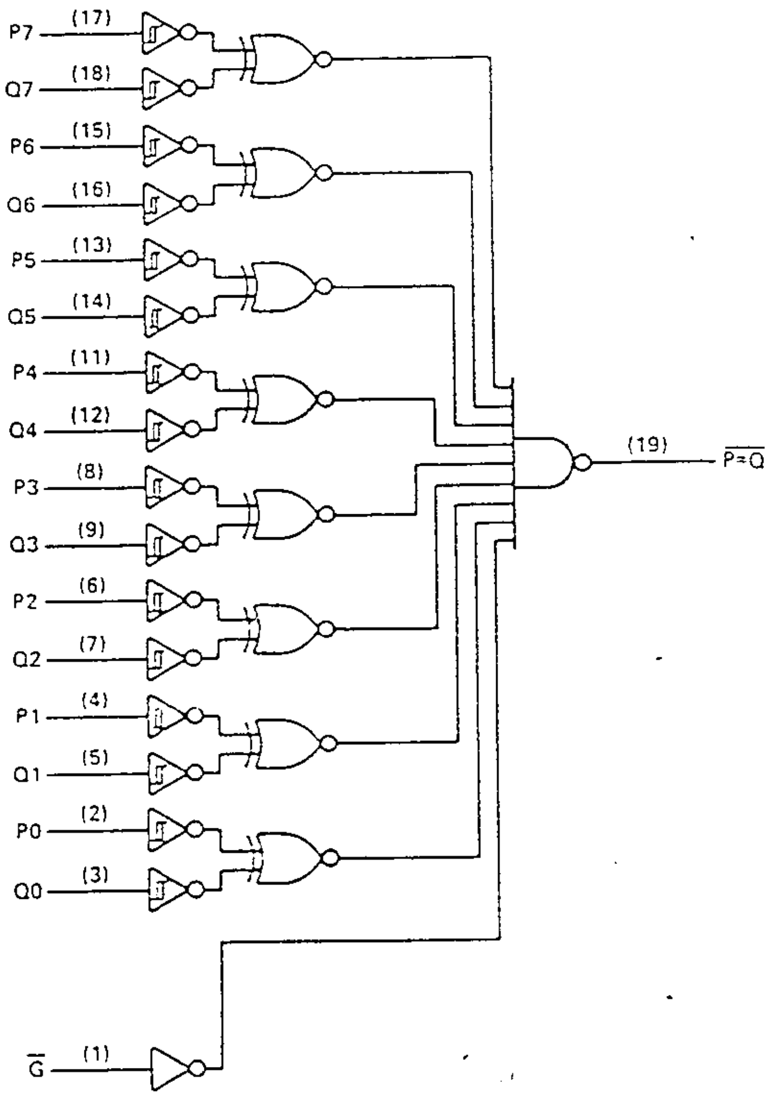
†All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.
NOTE 2: I_{CC} is measured with any $\bar{G}$ inputs grounded, all other inputs at 4.5 V, and all outputs open.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER#	FROM (INPUTS)	TO (OUTPUT)	TEST CONDITIONS	'LS682 MIN	'LS682 TYP	'LS682 MAX	'LS684 MIN	'LS684 TYP	'LS684 MAX	'LS686 MIN	'LS686 TYP	'LS686 MAX	'LS688 MIN	'LS688 TYP	'LS688 MAX	UNIT
t_{PLH}	P	$\overline{P = Q}$	R_L = 667 Ω, C_L = 45 pF, All other inputs low, See Note 3		13	25		15	25		13	25		18	27	ns
t_{PHL}					15	25		17	25		20	30		20	30	
t_{PLH}	Q	$\overline{P = Q}$			14	25		16	25		13	25		18	27	ns
t_{PHL}					15	25		15	25		21	30		20	30	
t_{PLH}	$\bar{G}$, $\bar{G}1$	$\overline{P = Q}$									11	20		12	18	ns
t_{PHL}											19	30		13	20	
t_{PLH}	P	$\overline{P > Q}$			20	30		22	30		19	30				ns
t_{PHL}					15	30		17	30		15	30				
t_{PLH}	Q	$\overline{P > Q}$			21	30		24	30		18	30				ns
t_{PHL}					19	30		20	30		19	30				
t_{PLH}	$\bar{G}2$	$\overline{P > Q}$									21	30				ns
t_{PHL}											16	25				

#t_{PLH} ≡ propagation delay time, low-to-high-level outputs; t_{PHL} ≡ propagation delay time, high-to-low-level output.
NOTE 3: See General Information Section for load circuits and voltage waveforms

'LS688, 'LS689 logic diagram (positive logic)



Pin numbers shown on logic notation are for DW, J or N packages

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage (see Note 1)		7 V
Input voltage: Q inputs of 'LS682 and 'LS683		5.5 V
All other inputs		7 V
Off-state output voltage: 'LS683, 'LS685, 'LS687, 'LS689		7 V
Operating free-air temperature range:	SN54LS682 thru SN54LS689	−55°C to 125°C
	SN74LS682 thru SN74LS689	0°C to 70°C
Storage temperature range		−65°C to 150°C

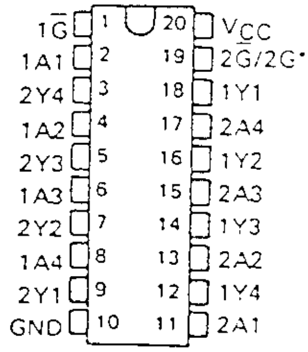
NOTE 1. Voltage values are with respect to network ground terminal.

SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

REVISED APRIL 1985

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- PNP Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

SN54LS', SN54S' . . . J PACKAGE
SN74LS', SN74S' . . . DW, J OR N PACKAGE
(TOP VIEW)

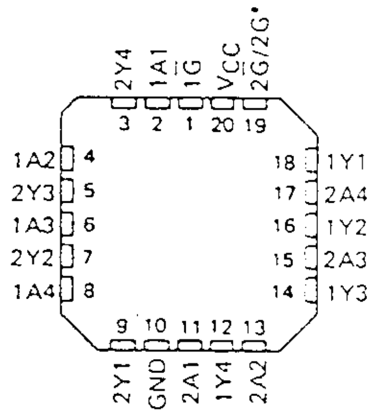


description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical G̅ (active-low output control) inputs, and complementary G and G̅ inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

The SN54' family is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74' family is characterized for operation from 0°C to 70°C.

SN54LS', SN54S' . . . FK PACKAGE
SN74LS', SN74S' . . . FN PACKAGE
(TOP VIEW)

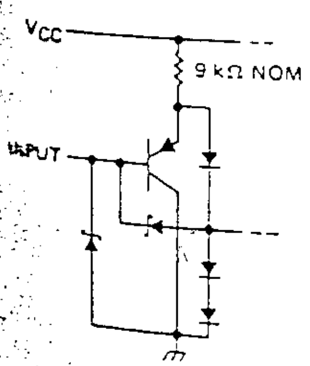


*2G for 'LS241 and 'S241 or 2G̅ for all other drivers.

schematics of inputs and outputs

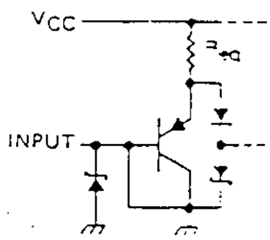
'LS240, 'LS241, 'LS244

EQUIVALENT OF EACH INPUT



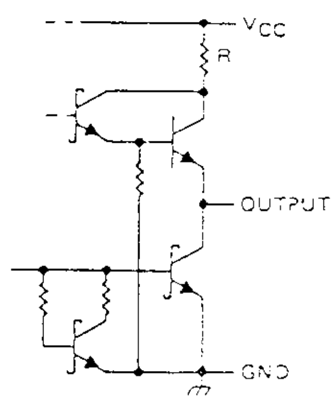
'S240, 'S241, 'S244

EQUIVALENT OF EACH INPUT



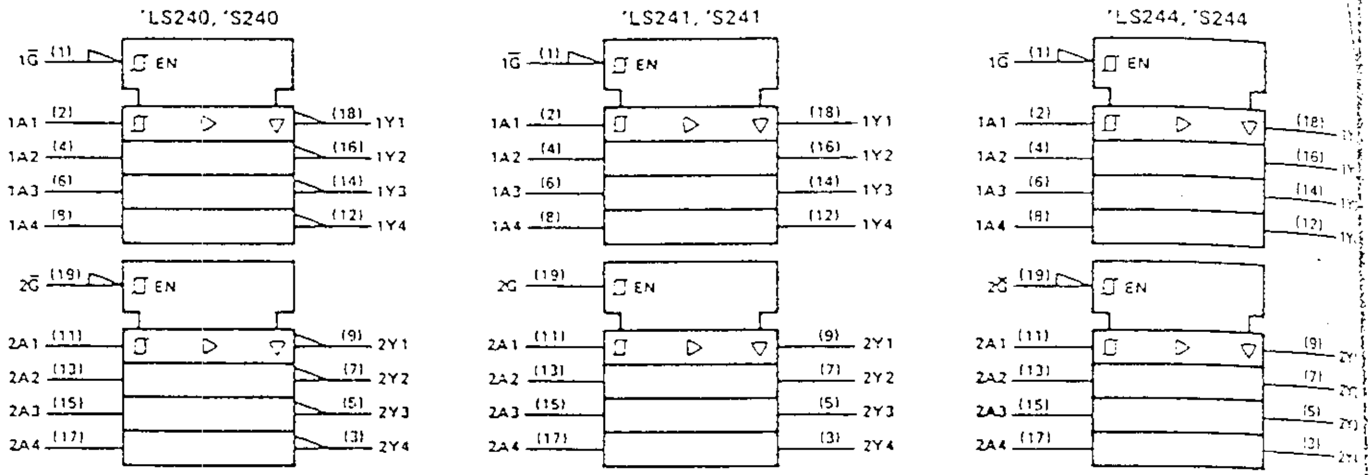
G and G̅ inputs: R_{eq} = 2 kΩ NOM
A inputs: R_{eq} = 2.8 kΩ NOM

TYPICAL OF ALL OUTPUTS

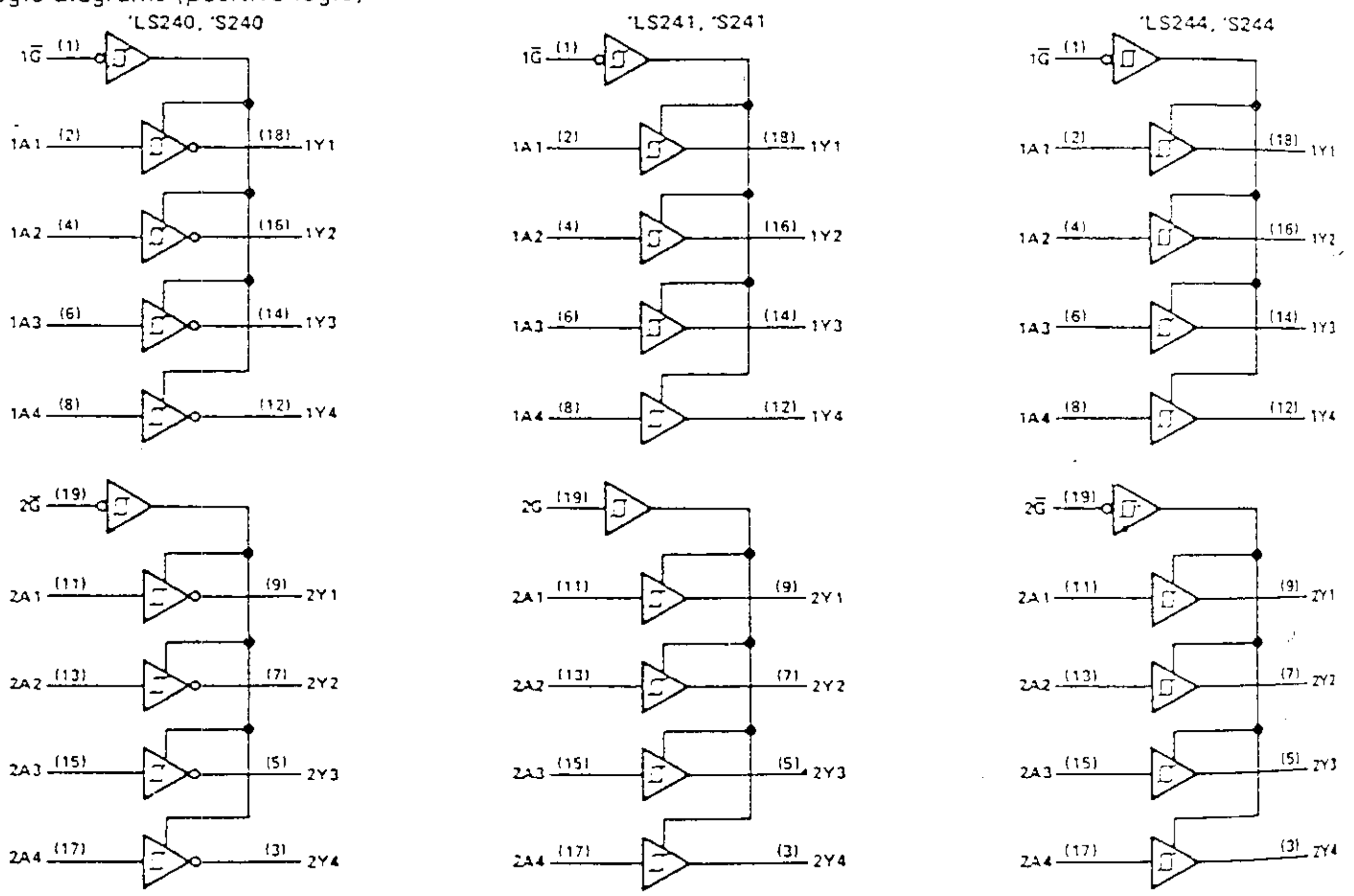


'LS240, 'LS241, 'LS244,
R = 50 Ω NOM
'S240, 'S241, 'S244
R = 25 Ω NOM

logic symbols



logic diagrams (positive logic)

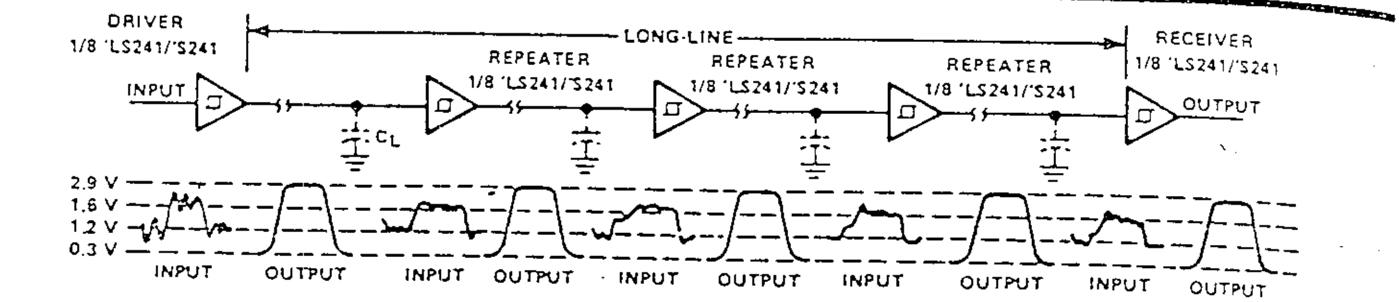


Pin numbers shown on logic notation are for DW, J or N packages.

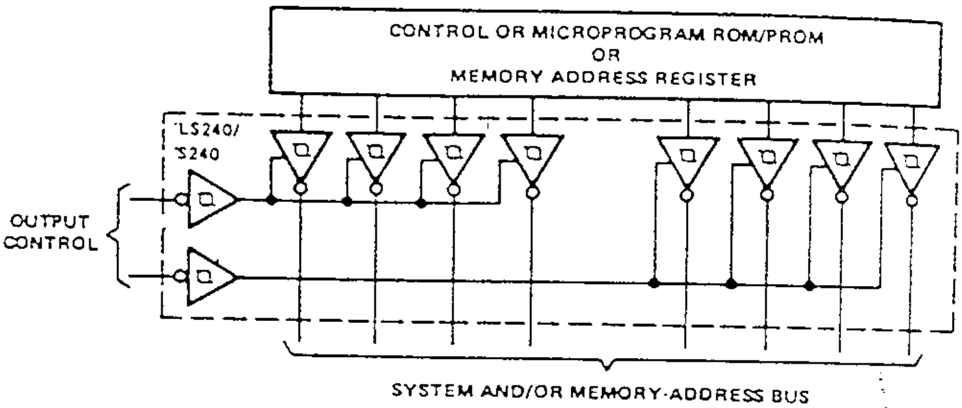
absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)		7 V
Input voltage:	'LS Circuits	7 V
	'S Circuits	5.5 V
Off-state output voltage		5.5 V
Operating free-air temperature range:	SN54LS', SN54S' Circuits	−55°C to 125°C
	SN74LS', SN74S' Circuits	0°C to 70°C
Storage temperature range		−65°C to 150°C

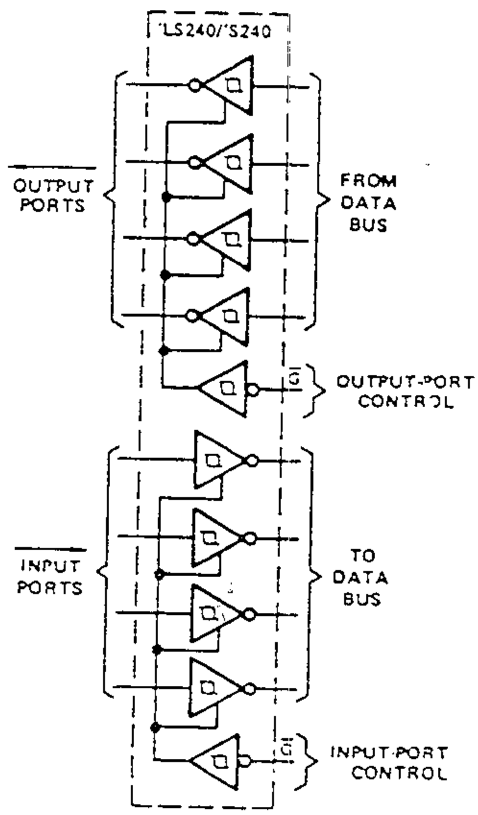
NOTE 1: Voltage values are with respect to network ground terminal.



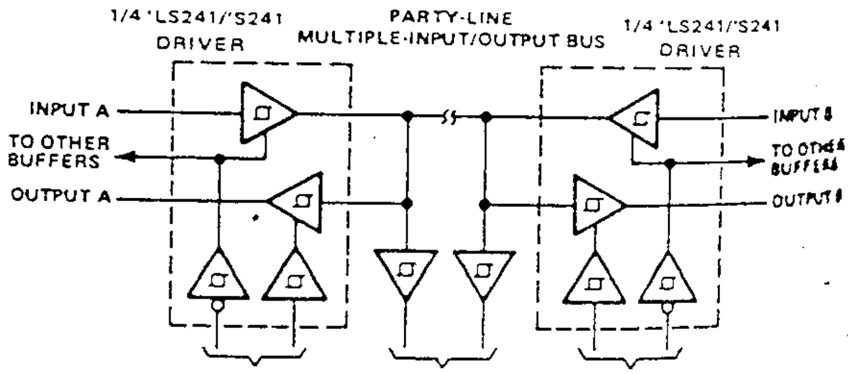
'LS241, 'S241 USED AS REPEATER/LEVEL RESTORER



'LS240/'S240 USED AS SYSTEM AND/OR MEMORY BUS DRIVER—4-BIT ORGANIZATION CAN BE APPLIED TO HANDLE BINARY OR BCD



INDEPENDENT 4-BIT BUS DRIVERS/RECEIVERS IN A SINGLE PACKAGE



BUS CONTROL		RECEIVERS INPUT	RECEIVERS OUTPUT	BUS CONTROL	
H	H	B	A	L	L
H	L	B	B	H	L
L	L	A	B	H	H
L	H	A	A	L	H
H	L	NONE	NONE	L	H

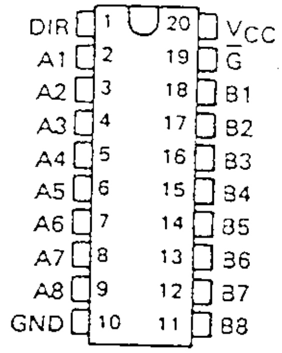
PARTY-LINE BUS SYSTEM WITH MULTIPLE INPUTS, OUTPUTS, AND RECEIVERS

OCTOBER 1976 — REVISED APRIL 1985

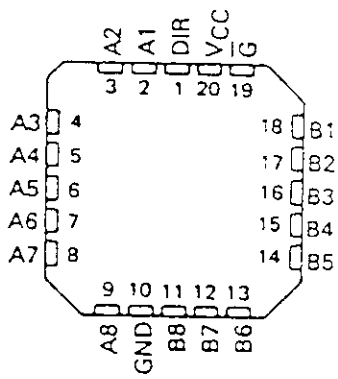
- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- PNP Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port . . . 8 ns

TYPE	I_{OL} (SINK CURRENT)	I_{OH} (SOURCE CURRENT)
SN54LS245	12 mA	-12 mA
SN74LS245	24 mA	-15 mA

SN54LS245 . . . J PACKAGE
SN74LS245 . . . DW, J OR N PACKAGE
(TOP VIEW)



SN54LS245 . . . FK PACKAGE
SN74LS245 . . . FN PACKAGE
(TOP VIEW)



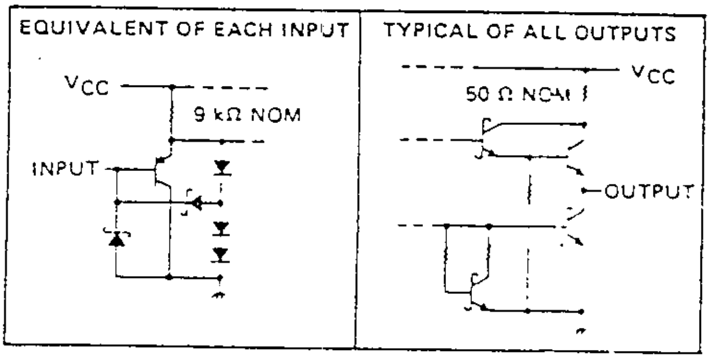
description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The devices allow data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input ($\bar{G}$) can be used to disable the device so that the buses are effectively isolated.

The SN54LS245 is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74LS245 is characterized for operation from 0°C to 70°C.

schematics of inputs and outputs



FUNCTION TABLE

ENABLE $\bar{G}$	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = high level, L = low level, X = irrelevant

recommended operating conditions

PARAMETER	SN54LS245 MIN	SN54LS245 NOM	SN54LS245 MAX	SN74LS245 MIN	SN74LS245 NOM	SN74LS245 MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−12			−15	mA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted

PARAMETER		TEST CONDITIONS†		SN54LS245 MIN	SN54LS245 TYP‡	SN54LS245 MAX	SN74LS245 MIN	SN74LS245 TYP‡	SN74LS245 MAX	UNIT
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage					0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN,	$I_I = -18$ mA			−1.5			−1.5	V
	Hysteresis ($V_{T+} - V_{T-}$) A or B input	V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, $V_{IL} = V_{IL}$ max	$I_{OH} = -3$ mA	2.4	3.4		2.4	3.4		V
			I_{OH} = MAX	2			2			V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, $V_{IL} = V_{IL}$ max	I_{OL} = 12 mA			0.4			0.4	V
			I_{OL} = 24 mA						0.5	V
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, $\bar{G}$ at 2 V	V_O = 2.7 V			20			20	μA
I_{OZL}	Off-state output current, low-level voltage applied		V_O = 0.4 V			−200			−200	μA
I_I	Input current at maximum input voltage — A or B	V_{CC} = MAX,	V_I = 5.5 V			0.1			0.1	mA
	Input current at maximum input voltage — DIR or $\bar{G}$		V_I = 7 V			0.1			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX,	V_{IH} = 2.7 V			20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX,	V_{IL} = 0.4 V			−0.2			−0.2	mA
I_{OS}	Short-circuit output current¶	V_{CC} = MAX		−40		−225	−40		−225	mA
I_{CC}	Supply current — Total, outputs high	V_{CC} = MAX,	Outputs open		48	70		48	70	mA
	Supply current — Total, outputs low				62	90		62	90	mA
	Supply current — Outputs at Hi-Z				64	95		64	95	mA

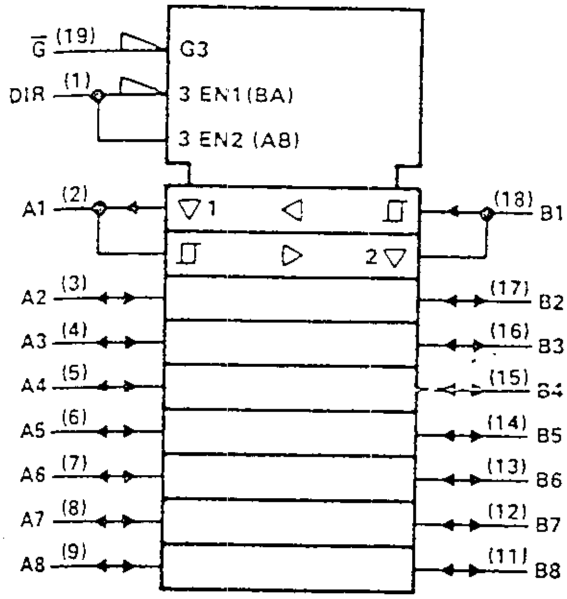
†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
¶Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

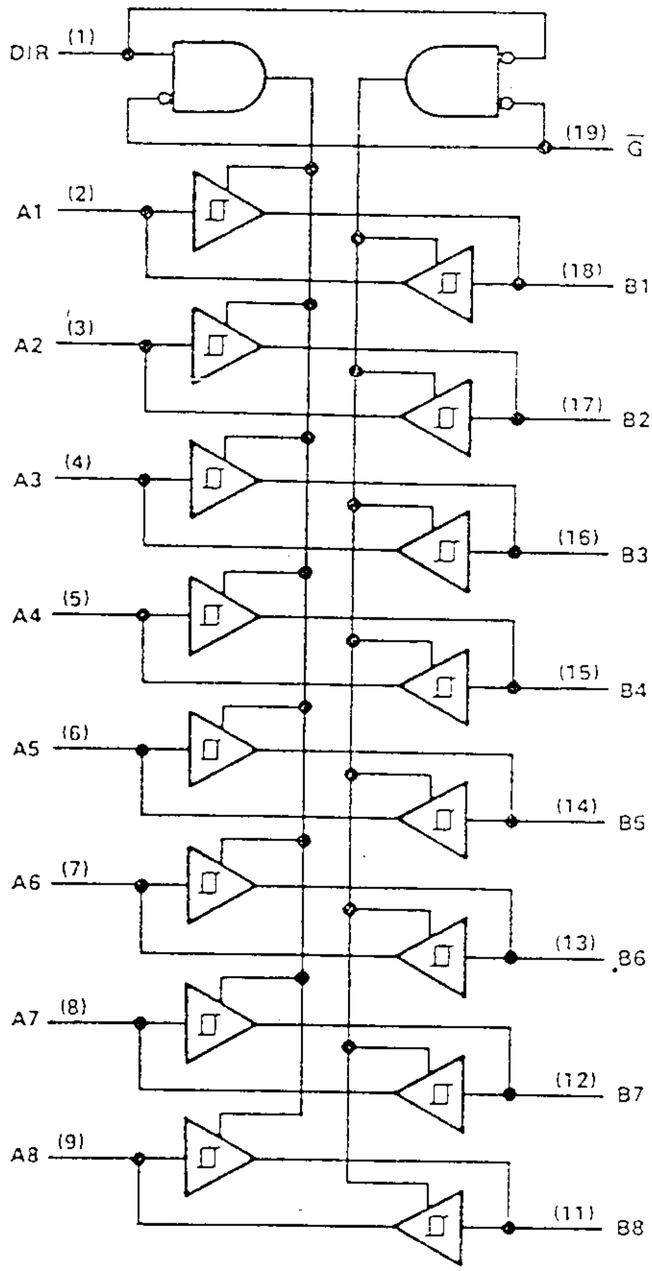
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 45 pF, R_L = 667 Ω, See Note 2		8	12	ns
t_{PHL}	Propagation delay time, high-to-low-level output			8	12	ns
t_{PZL}	Output enable time to low level			27	40	ns
t_{PZH}	Output enable time to high level			25	40	ns
t_{PLZ}	Output disable time from low level	C_L = 5 pF, R_L = 667 Ω, See Note 2		15	25	ns
t_{PHZ}	Output disable time from high level			15	28	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

logic symbol



logic diagram (positive logic)



Pin numbers shown on logic notation are for DW, J or N packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)		7 V
Input voltage		7 V
Off-state output voltage		5.5 V
Operating free-air temperature range:	SN54LS'	−55°C to 125°C
	SN74LS'	0°C to 70°C
Storage temperature range		−65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . 7 V
Input voltage . . . . 7 V
Off-state output voltage . . . . 5.5 V
Operating free-air temperature range: SN54LS' . . . . −55°C to 125°C
SN74LS' . . . . 0°C to 70°C
Storage temperature range . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

			SN54LS'			SN74LS'			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		4.5	5	5.5	4.75	5	5.25	V
V_{OH}	High-level output voltage				5.5			5.5	V
I_{OH}	High-level output current				−1			−2.6	mA
I_{OL}	Low-level output current				12			24	mA
t_w	Pulse duration	CLK high	15			15			ns
		CLK low	15			15			
t_{su}	Data setup time	'LS373	5↓			5↓			ns
		'LS374	20↑			20↑			
t_h	Data hold time	'LS373	20↓			20↓			ns
		'LS374†	0↑			0↑			
T_A	Operating free-air temperature		−55		125	0		70	°C

† The t_h specification applies only for data frequency below 10 MHz. Designs above 10 MHz should use a minimum of 5 ns.

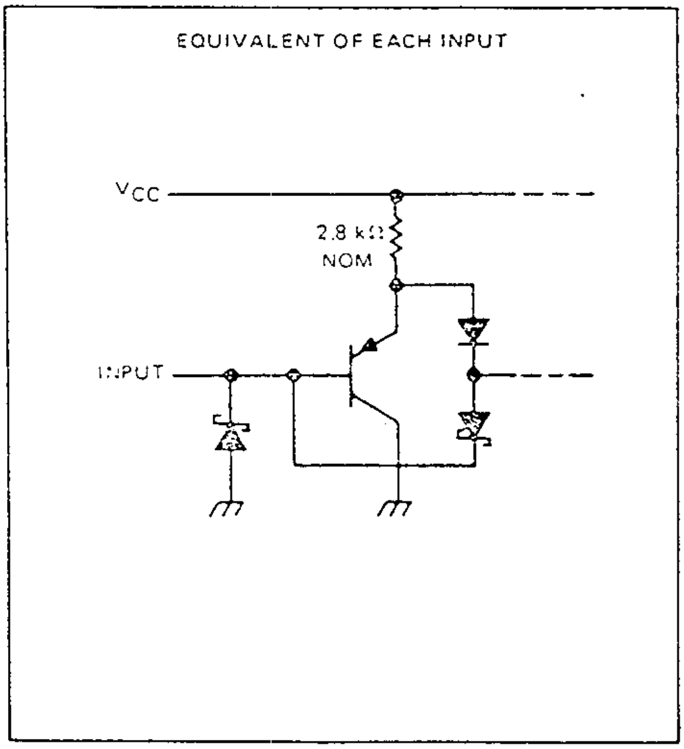
electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

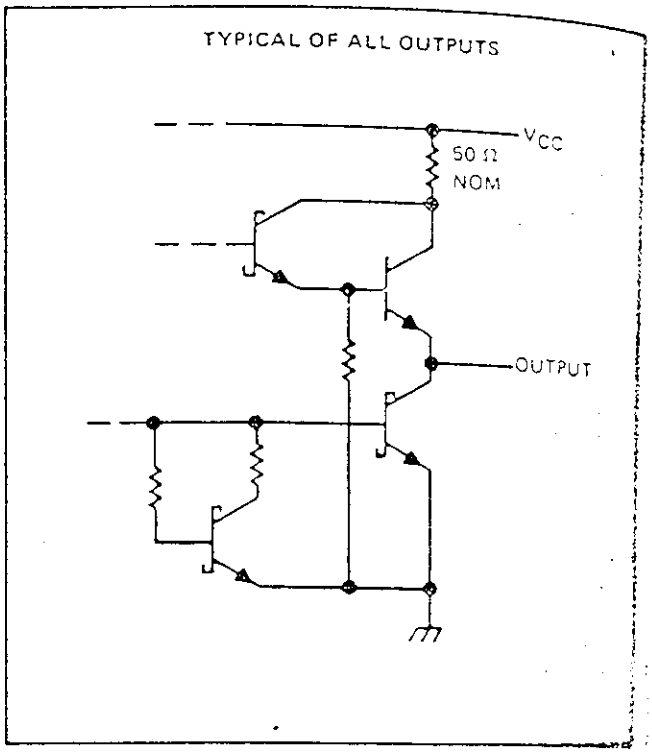
PARAMETER		TEST CONDITIONS†		SN54LS' MIN	SN54LS' TYP‡	SN54LS' MAX	SN74LS' MIN	SN74LS' TYP‡	SN74LS' MAX	UNIT
V_{IH}	High-level input voltage			2			2			[illegible]
V_{IL}	Low-level input voltage					0.7			0.8	[illegible]
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18 mA				−1.5			−1.5	[illegible]
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max, I_{OH} = MAX		2.4	3.4		2.4	3.1		[illegible]
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max	I_{OL} = 12 mA		0.25	0.4		0.25	0.4	[illegible]
			I_{OL} = 24 mA					0.35	0.5	
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2 V, V_O = 2.7 V				20			20	[illegible]
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX, V_{IH} = 2 V, V_O = 0.4 V				−20			−20	[illegible]
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7 V				0.1			0.1	[illegible]
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7 V				20			20	[illegible]
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4 V				−0.4			−0.4	[illegible]
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−30		−130	−30		−130	[illegible]
I_{CC}	Supply current	V_{CC} = MAX, Output control at 4.5 V	'LS373		24	40		24	40	[illegible]
			'LS374		27	40		27	40	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

schematic of inputs and outputs





absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Off-state output voltage	5.5 V
Operating free-air temperature range: SN54S'	−55°C to 125°C
SN74S'	0°C to 70°C
Storage temperature range	−65°C to 150°C

NOTE 1 Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54S'			SN74S'			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, V_{OH}				5.5			5.5	V
High-level output current, I_{OH}				−2			−6.5	mA
Width of clock/enable pulse, t_w	High	6			6			ns
	Low	7.3			7.3			
Data setup time, t_{su}	'S373	0↓			0↓			ns
	'S374	5↑			5↑			
Data hold time, t_h	'S373	10↓			10↓			ns
	'S374	2↑			2↑			
Operating free-air temperature, T_A		−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†			MIN	TYP‡	MAX	UNIT
V_{IH}					2			V
V_{IL}							0.8	V
V_{IK}		V_{CC} = MIN, I_I = −18 mA					−1.2	V
V_{OH}	SN54S'	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = MAX			2.4	3.4		V
	SN74S'				2.4	3.1		
V_{OL}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA					0.5	V
I_{OZH}		V_{CC} = MAX, V_{IH} = 2 V, V_O = 2.4 V					50	μA
I_{OZL}		V_{CC} = MAX, V_{IH} = 2 V, V_O = 0.5 V					−50	μA
I_I		V_{CC} = MAX, V_I = 5.5 V					1	mA
I_{IH}		V_{CC} = MAX, V_I = 2.7 V					50	μA
I_{IL}		V_{CC} = MAX, V_I = 0.5 V					−250	μA
I_{OS}§		V_{CC} = MAX			−40		−100	mA
I_{CC}		V_{CC} = MAX	'S373	outputs high			160	mA
				outputs low			160	
				outputs disabled			190	
			'S374	outputs high			110	
				outputs low			140	
				outputs disabled			160	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'S373 MIN	'S373 TYP	'S373 MAX	'S374 MIN	'S374 TYP	'S374 MAX	UNIT
f_{max}			C_L = 15 pF, R_L = 280 Ω, See Notes 2 and 4				75	100		MHz
t_{PLH}	Data	Any Q			7	12				ns
t_{PHL}					7	12				
t_{PLH}	Clock or enable	Any Q			7	14		8	15	ns
t_{PHL}					12	18		11	17	
t_{PZH}	Output Control	Any Q			8	15		8	15	ns
t_{PZL}					11	18		11	18	
t_{PHZ}	Output Control	Any Q	C_L = 5 pF, R_L = 280 Ω, See Note 3		6	9		5	9	ns
t_{PLZ}					8	12		7	12	

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
4. See General Information Section for load circuits and voltage waveforms

f_{max} ≡ maximum clock frequency
t_{PLH} ≡ propagation delay time, low-to-high-level output
t_{PHL} ≡ propagation delay time, high-to-low-level output
t_{PZH} ≡ output enable time to high level
t_{PZL} ≡ output enable time to low level
t_{PHZ} ≡ output disable time from high level
t_{PLZ} ≡ output disable time from low level

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		'LS373 MIN	'LS373 TYP	'LS373 MAX	'LS374 MIN	'LS374 TYP	'LS374 MAX	UNIT
f_{max}			C_L = 45 pF, R_L = 667 Ω See Notes 2 and 3					35	50		MHz
t_{PLH}	Data	Any Q				12	18				ns
t_{PHL}						12	18				
t_{PLH}	Clock or enable	Any Q				20	30		15	28	ns
t_{PHL}						18	30		19	28	
t_{PZH}	Output Control	Any Q				15	28		20	26	ns
t_{PZL}						25	36		21	28	
t_{PHZ}	Output Control	Any Q	C_L = 5 pF, R_L = 667 Ω See Note 3	SN54		28	32		28	32	ns
				SN74		15	25		15	28	
t_{PLZ}	Output Control	Any Q				12	20		12	20	ns

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
3. See General Information Section for load circuits and voltage waveforms.

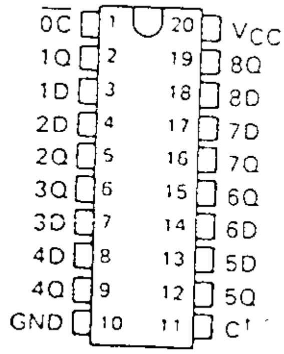
f_{max} ≡ maximum clock frequency
t_{PLH} ≡ propagation delay time, low-to-high-level output
t_{PHL} ≡ propagation delay time, high-to-low-level output
t_{PZH} ≡ output enable time to high level
t_{PZL} ≡ output enable time to low level
t_{PHZ} ≡ output disable time from high level
t_{PLZ} ≡ output disable time from low level

OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

OCTOBER 1975 – REVISED APRIL 1985

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)

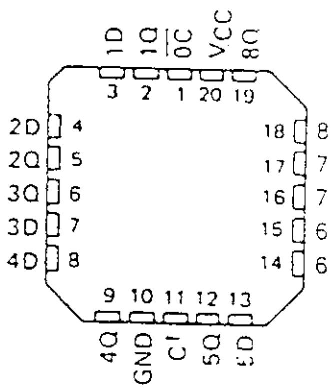
SN54LS373, SN54LS374, SN54S373, SN54S374 . . . J PACKAGE
SN74LS373, SN74LS374, SN74S373, SN74S374 . . . DW, J OR N PACKAGE
(TOP VIEW)



'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . FK PACKAGE
SN74LS373, SN74LS374, SN74S373, SN74S374 . . . FN PACKAGE
(TOP VIEW)



'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

†C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

description

These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

description (continued)

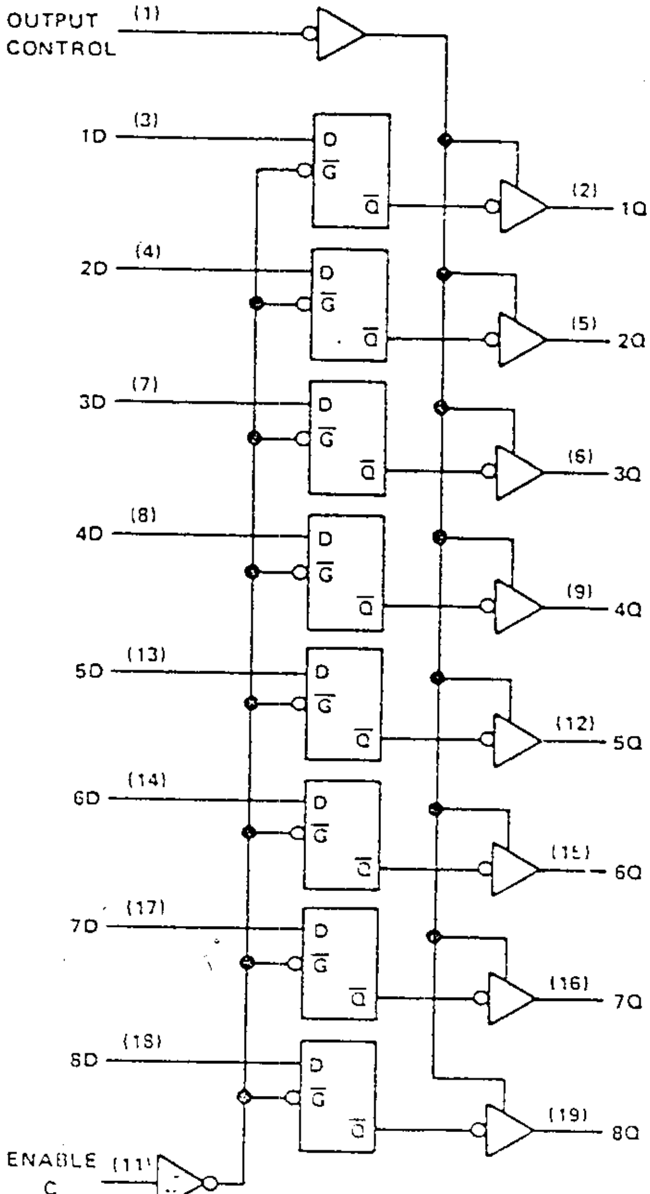
The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices, simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

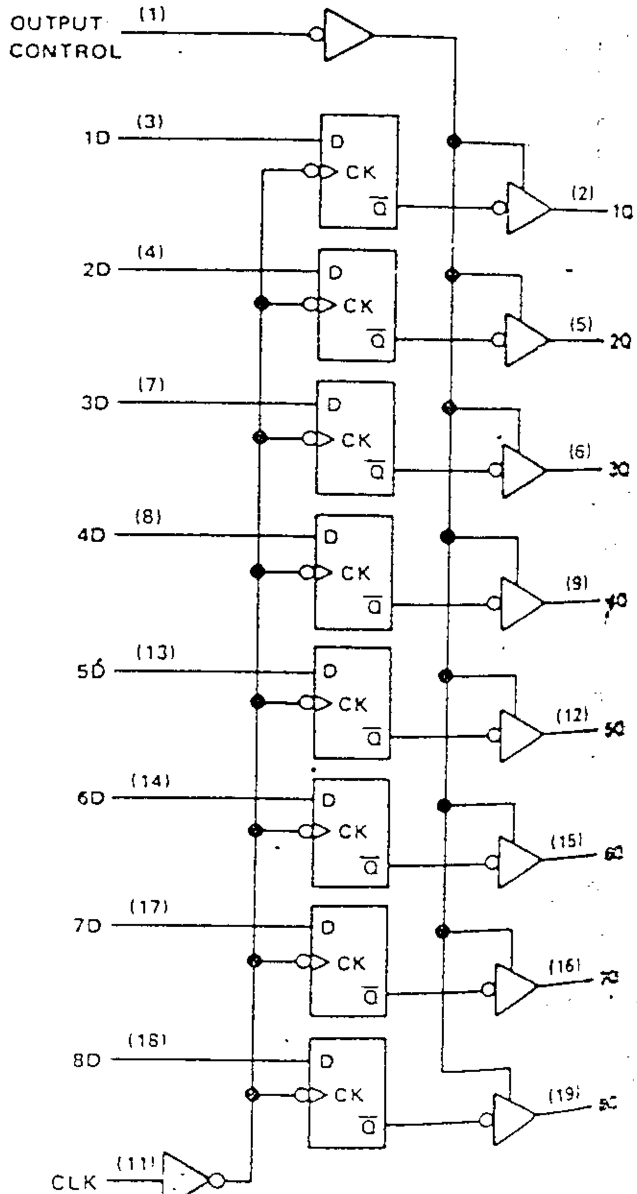
logic diagrams

'LS373, 'S373
TRANSPARENT LATCHES



for 'S373 only

'LS374, 'S374
POSITIVE-EDGE-TRIGGERED FLIP-FLOPS

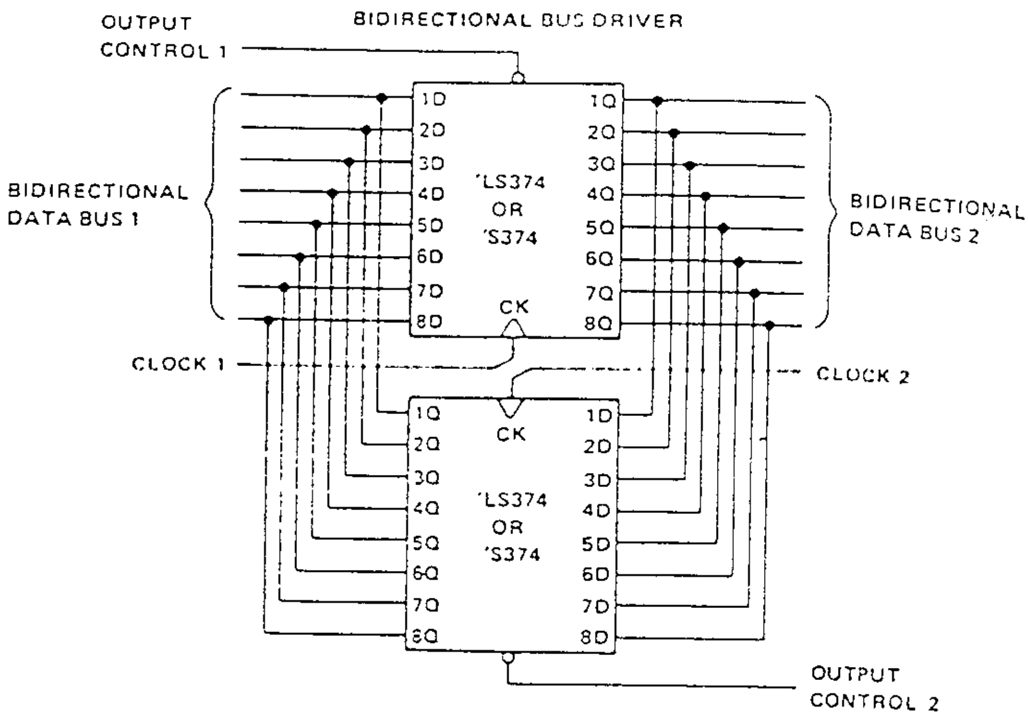


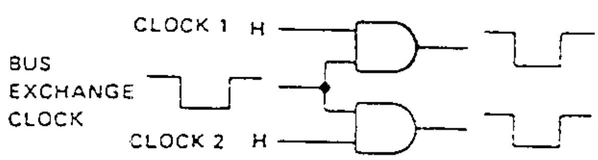
for 'S374 only

Pin numbers shown on logic notation are for DW, J or N packages.

TYPICAL APPLICATION DATA

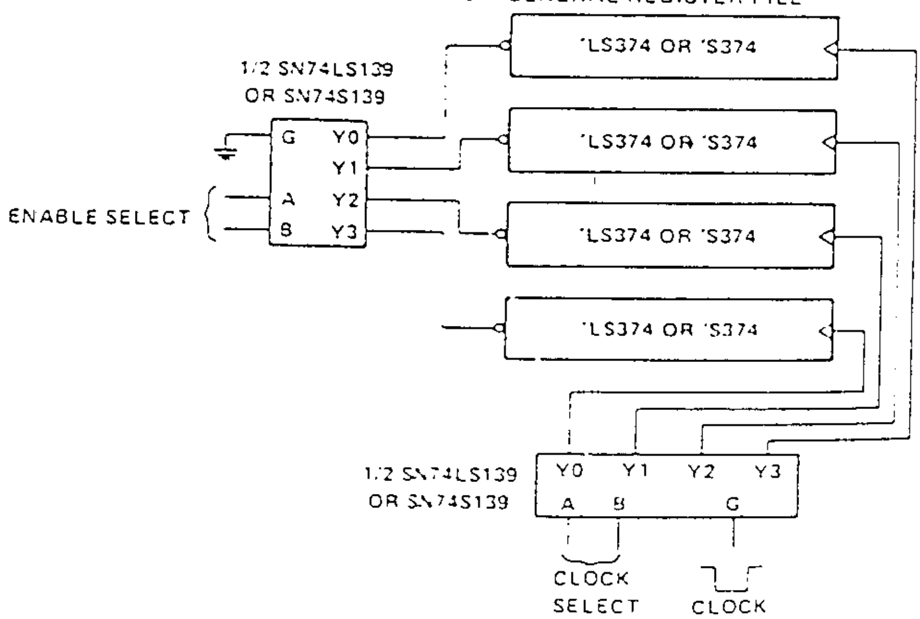
BIDIRECTIONAL BUS DRIVER





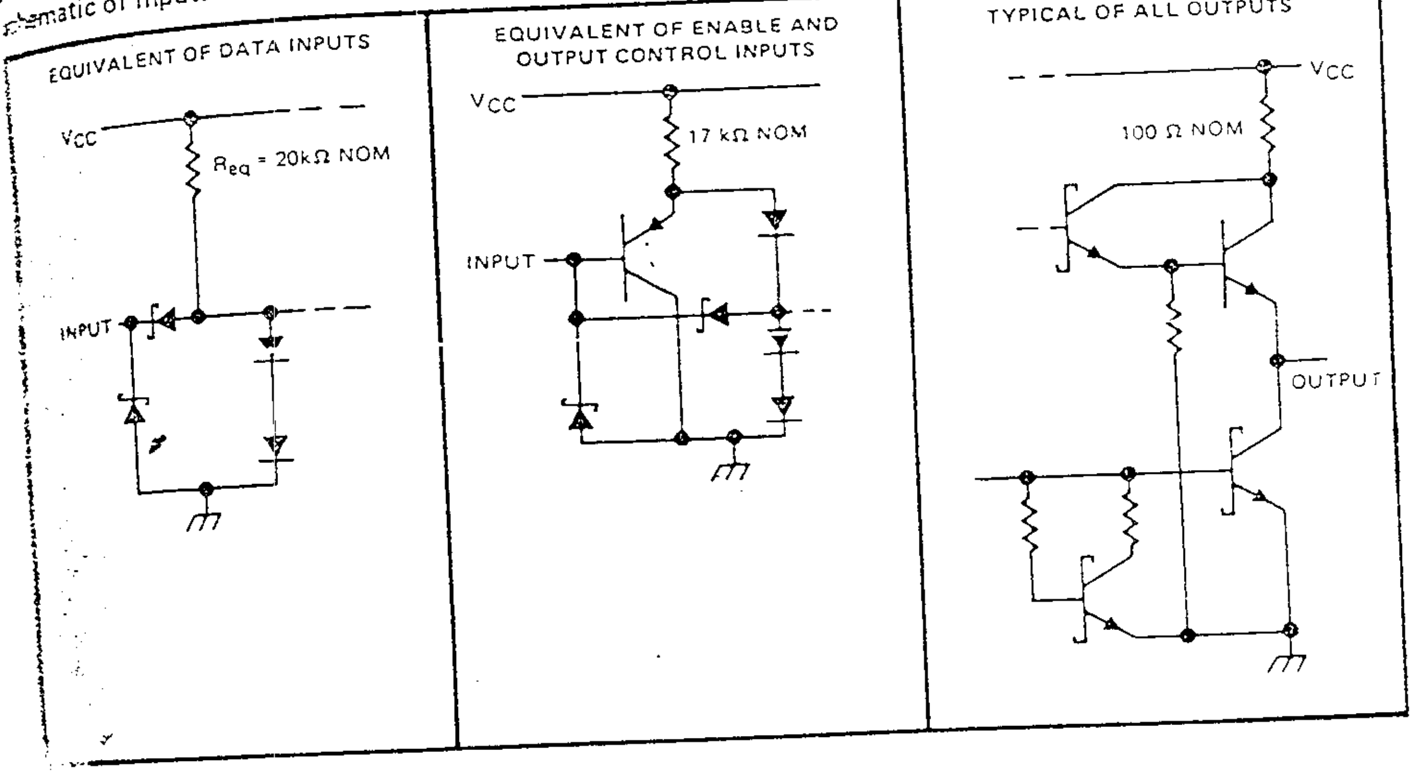
CLOCK CIRCUIT FOR BUS EXCHANGE

EXPANDABLE 4-WORD-BY-8-BIT GENERAL REGISTER FILE

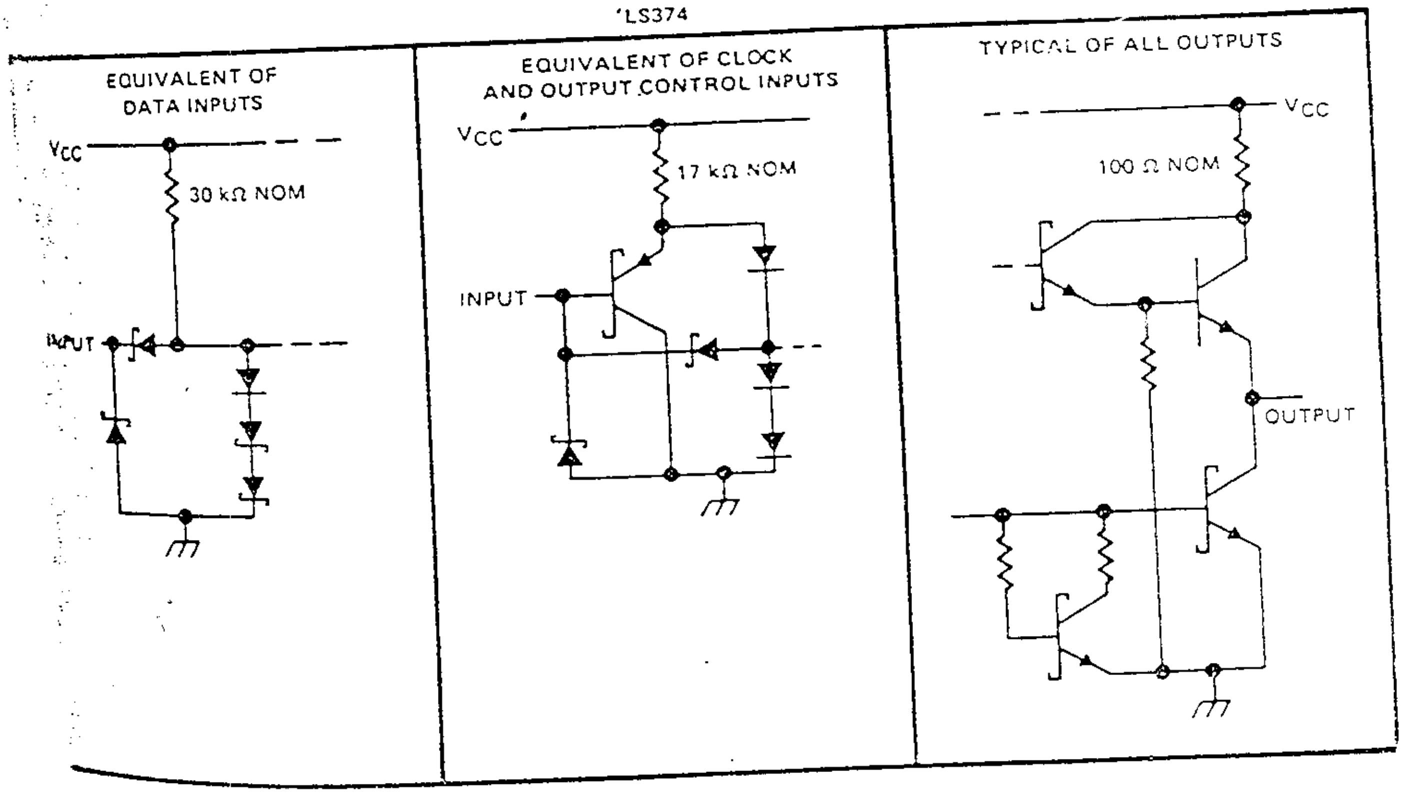


schematic of inputs and outputs

'LS373

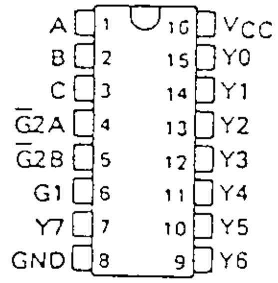


'LS374

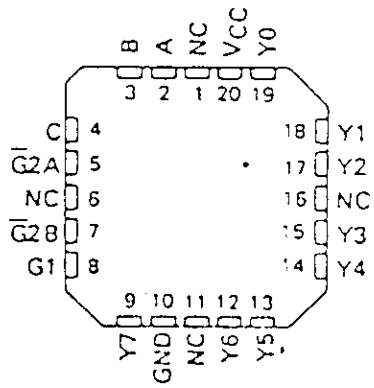


- Designed Specifically for High-Speed:
  Memory Decoders
  Data Transmission Systems
- 3 Enable Inputs to Simplify Cascading and/or Data Reception
- Schottky-Clamped for High Performance

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . D, J OR N PACKAGE
(TOP VIEW)



SN54LS138, SN54S138 . . . FK PACKAGE
SN74LS138, SN74S138 . . . FN PACKAGE
(TOP VIEW)



NC – No internal connection

scription

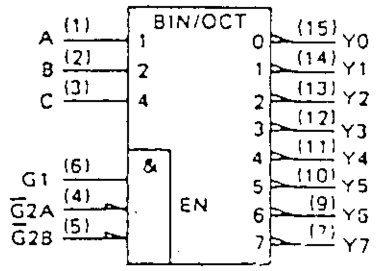
These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'LS138 and 'S138 decode one of eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

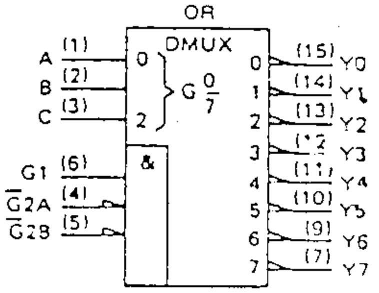
All of these decoder/demultiplexers feature fully buffered inputs, each of which represents only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and to simplify system design.

The SN54LS138 and SN54S138 are characterized for operation over the full military temperature range of −55°C to 125°C. The SN74LS138 and SN74S138 are characterized for operation from 0°C to 70°C.

logic symbols



OR



Pin numbers shown on logic notation are for D, J or N packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . 7 V
Input voltage . . . . 7 V
Operating free-air temperature range: SN54LS138 . . . . −55°C to 125°C
SN74LS138 . . . . 0°C to 70°C
Storage temperature range . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54LS138			SN74LS138			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.7			0.8	V
I_{OH}	High-level output current			−0.4			−0.4	mA
I_{OL}	Low-level output current			4			8	mA
T_A	Operating free-air temperature	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS †		SN54LS138			SN74LS138			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	V_{CC} = MIN, I_I = −18 mA				−1.5			−1.5	V
V_{OH}	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = MAX, I_{OH} = −0.4 mA		2.5	3.4		2.7	3.4		V
V_{OL}	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = MAX	I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
		I_{OL} = 8 mA					0.35	0.5	
I_I	V_{CC} = MAX, V_I = 7 V				0.1			0.1	mA
I_{IH}	V_{CC} = MAX, V_I = 2.7 V				20			20	μA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V	Enable			−0.4			−0.4	mA
		A, B, C			−0.2			−0.2	
I_{OS} §	V_{CC} = MAX		−20		−100	−20		−100	mA
I_{CC}	V_{CC} = MAX, Outputs enabled and open			6.3	10		6.3	10	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time.

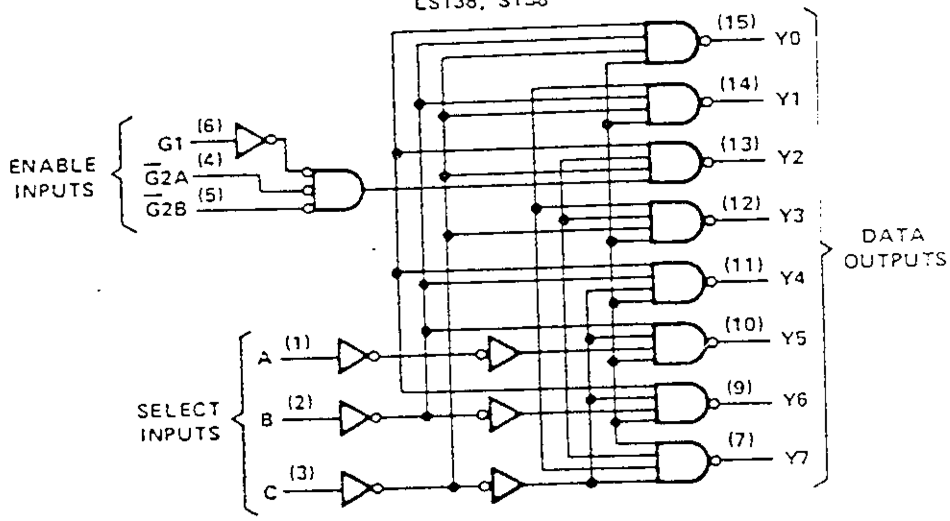
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER ¶	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54LS138 SN74LS138			UNIT
					MIN	TYP	MAX	
t_{PLH}	Binary Select	Any	2	R_L = 2 kΩ, C_L = 15 pF, See Note 2		11	20	ns
t_{PHL}						18	41	ns
t_{PLH}			3			21	27	ns
t_{PHL}						20	39	ns
t_{PLH}	Enable	Any	2			12	18	ns
t_{PHL}						20	32	ns
t_{PLH}			3			14	26	ns
t_{PHL}						13	38	ns

¶ t_{PLH} = propagation delay time, low-to-high-level output; t_{PHL} = propagation delay time, high-to-low-level output.
NOTE 2: See General Information Section for load circuits and voltage waveforms.

logic diagram and function table

'LS138, 'S138



Pin numbers shown on logic notation are for D, J or N packages.

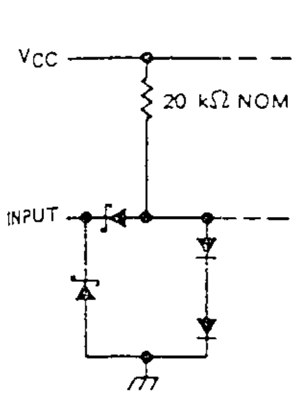
'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	$\overline{G}$2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

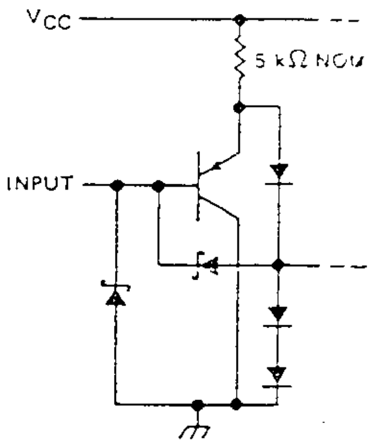
*$\overline{G}$2 = $\overline{G}$2A + $\overline{G}$2B
H = high level, L = low level, X = irrelevant

schematics of inputs and outputs

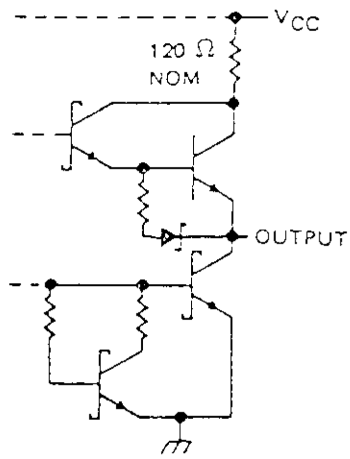
EQUIVALENT OF EACH ENABLE INPUT OF 'LS138



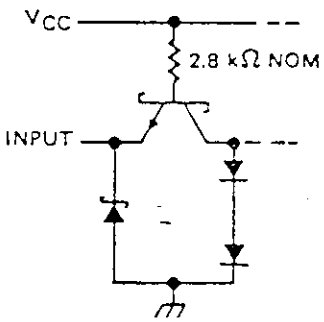
EQUIVALENT OF EACH SELECT INPUT OF 'LS138



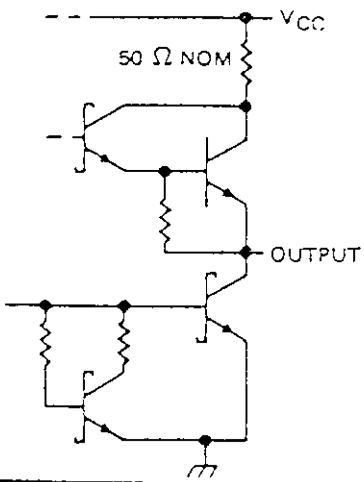
TYPICAL OF OUTPUTS OF 'LS138



EQUIVALENT OF EACH INPUT OF 'S138



TYPICAL OF OUTPUTS OF 'S138



recommended operating conditions

		SN54LS04			SN74LS04			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.7			0.8	V
I_{OH}	High-level output current			−0.4			−0.4	mA
I_{OL}	Low-level output current			4			8	mA
T_A	Operating free-air temperature	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

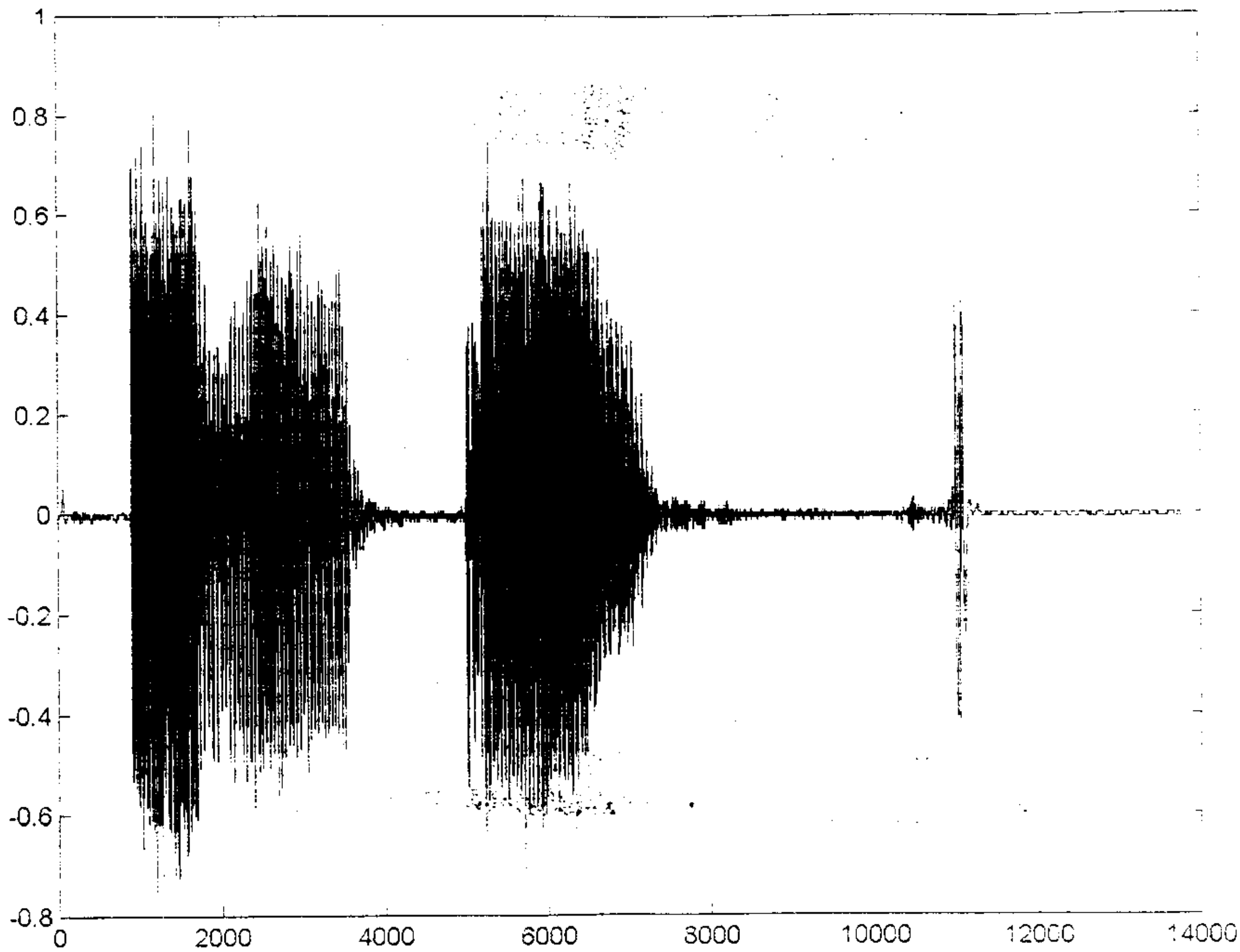
PARAMETER	TEST CONDITIONS †	SN54LS04			SN74LS04			UNIT
		MIN	TYP ‡	MAX	MIN	TYP ‡	MAX	
V_{IK}	V_{CC} = MIN, I_I = −18 mA			−1.5			−1.5	V
V_{OH}	V_{CC} = MIN, V_{IL} = MAX, I_{OH} = −0.4 mA	2.5	3.4		2.7	3.4		V
V_{OL}	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = 4 mA		0.25	0.4			0.4	V
	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = 8 mA					0.25	0.5	
I_I	V_{CC} = MAX, V_I = 7 V			0.1			0.1	mA
I_{IH}	V_{CC} = MAX, V_I = 2.7 V			20			20	µA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V			−0.4			−0.4	mA
I_{OS} §	V_{CC} = MAX	−20		−100	−20		−100	mA
I_{CCH}	V_{CC} = MAX, V_I = 0 V		1.2	2.4		1.2	2.4	mA
I_{CCL}	V_{CC} = MAX, V_I = 4.5 V		3.6	6.6		3.6	6.6	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

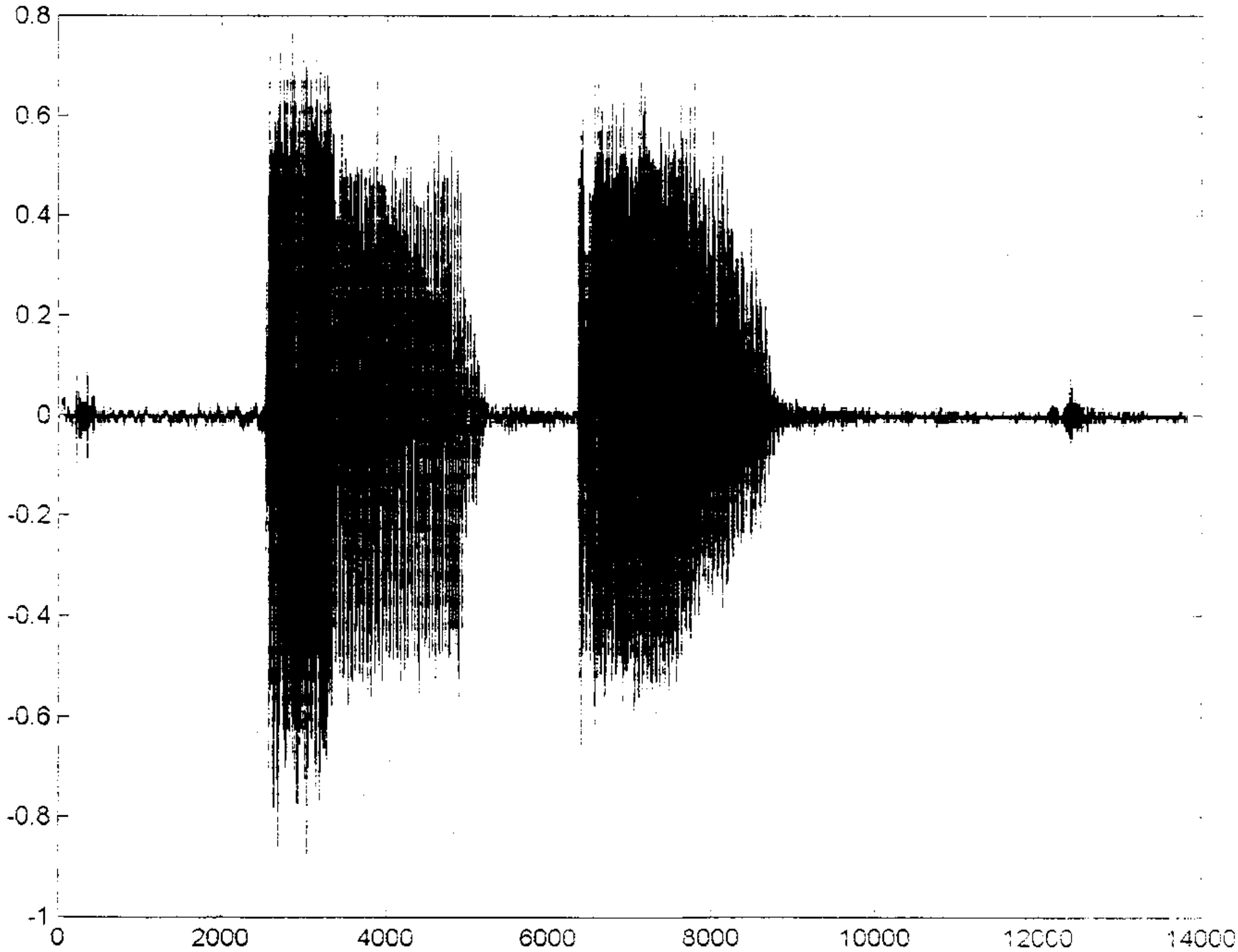
switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	R_L = 2 kΩ, C_L = 15 pF		9	15	ns
t_{PHL}					10	15	ns

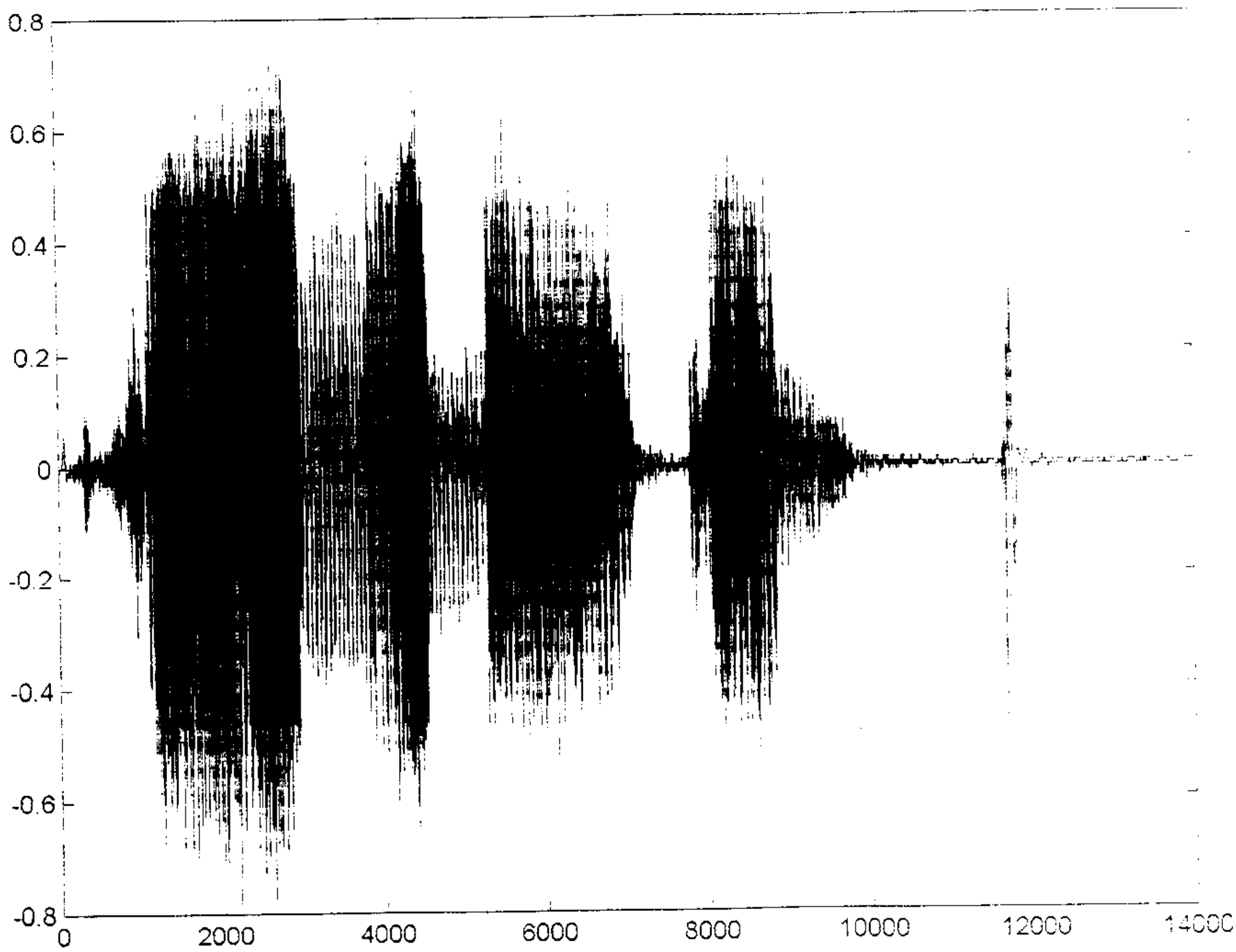
NOTE 2: See General Information Section for load circuits and voltage waveforms.



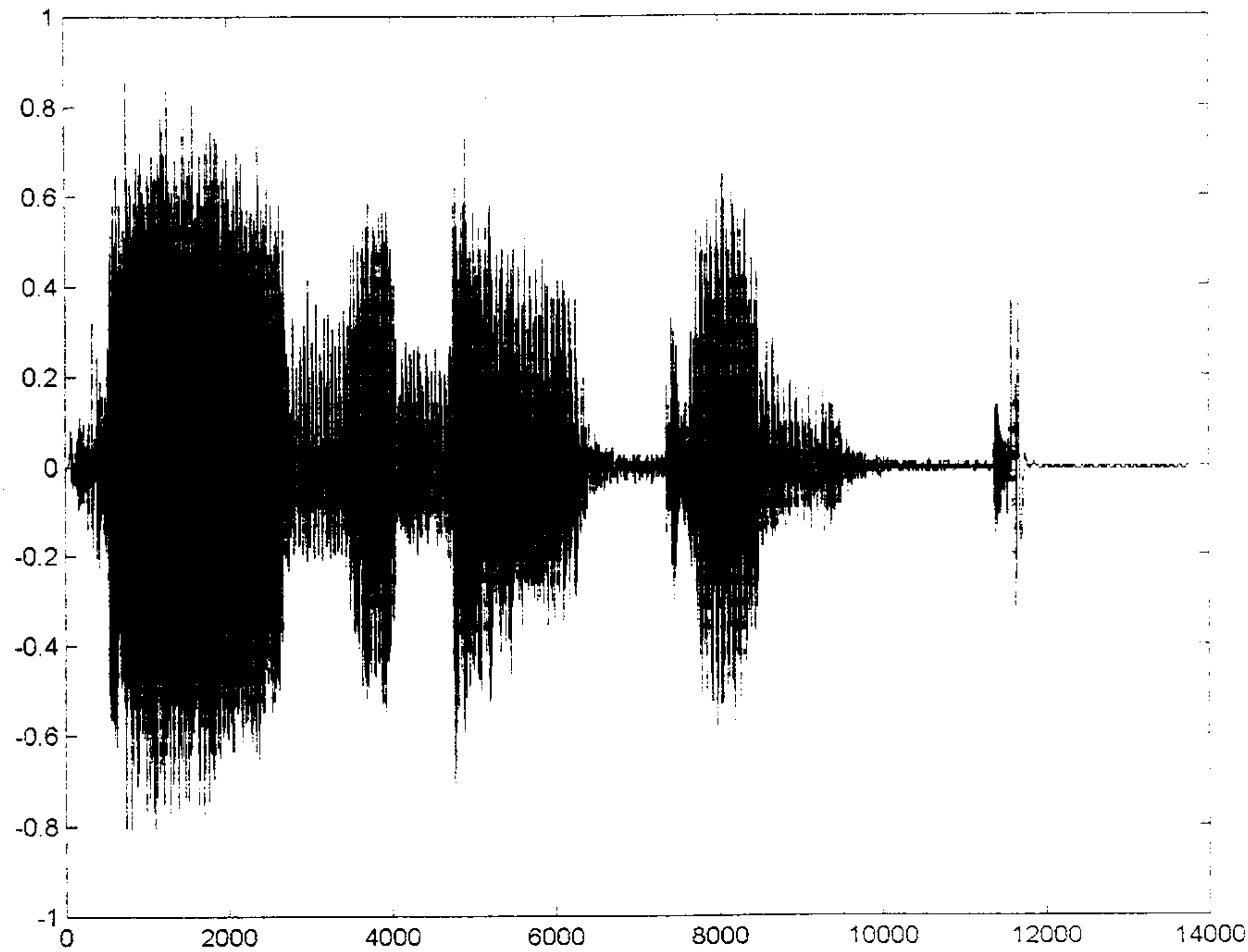
VOICE I TEMPLATE A



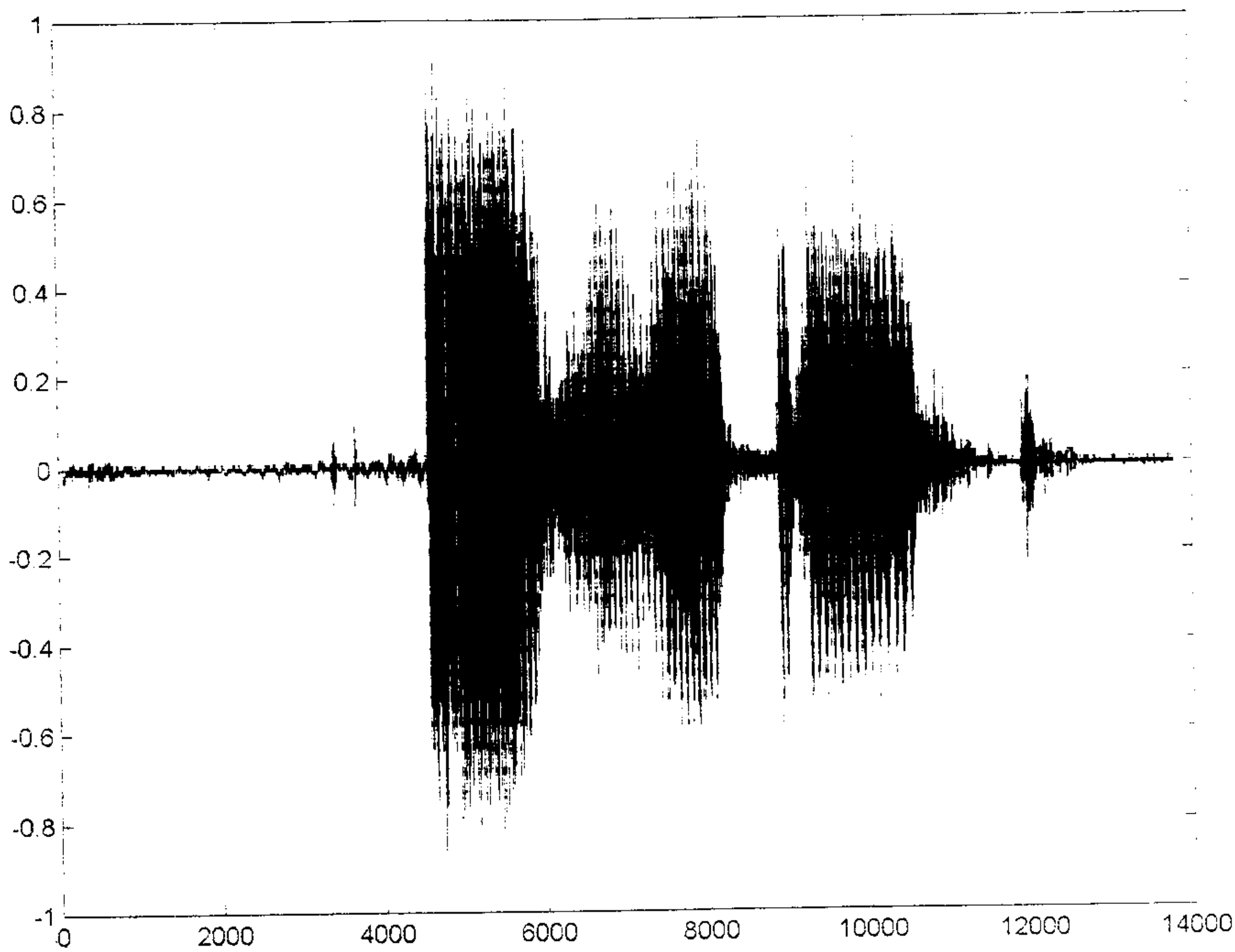
VOICE I TEMPLATE B



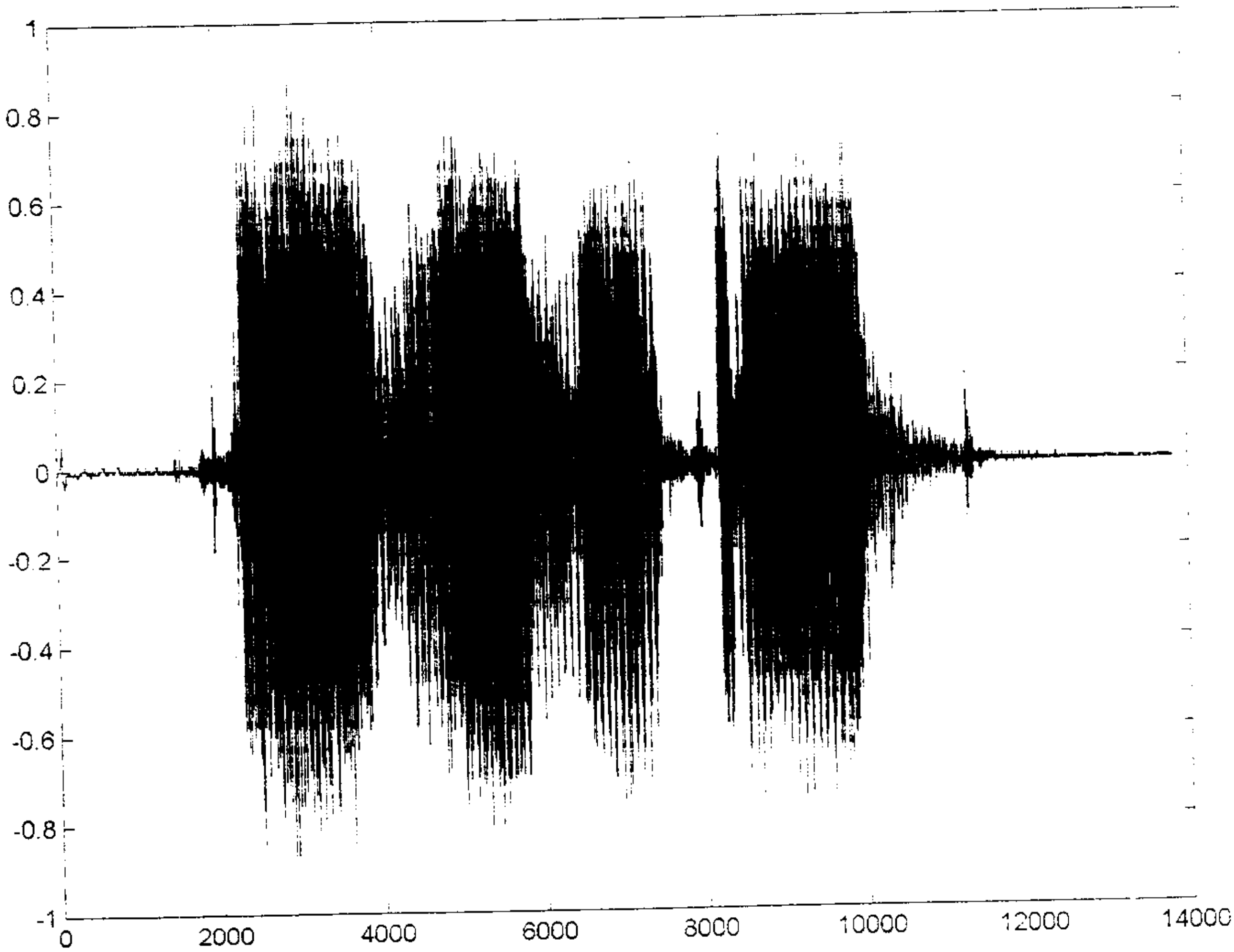
VOICE II TEMPLATE A



VOICE II TEMPLATE B



VOICE III TEMPLATE A



VOICE III TEMPLATE B

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Panos E. Papamichalis, " Practical Approaches to Speech Coding", Texas Instruments, Inc., Rice University.
2. John G. Proakis & Dimitris G. Manolakis, " Digital signal Processing – Principles,Algorithms and Applications", PHI.
3. Roger Moor, "Speech Processing",1990.
4. Steve Young,"A review of Large Vocabulary Continuous Speech Recognition", IEEE communications magazine, September 1996.
5. Richard D. Peacock and Daryl H. Graph, "An introduction to Speech and Speaker Recognition", IEEE Computers, August 1990.
6. Herbert Gish and Michal Schmidt,"Text independent Speaker Recognition", IEEE S.P. magazine,1994.
7. Luciano et al, "Lexical Access to Large Vocabularies for Speech Recognition ",IEEE Speech and Audio Processing,1989.