

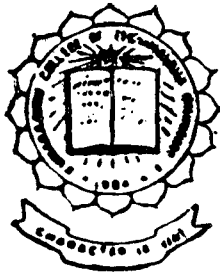
Computer Aided Speech Synthesizer

PROJECT REPORT

P-144

BY

Mr. S. MAHALINGAM
Mr. R. SHYAM SUNDAR
Mr. A. BASKARAN



1991-92

GUIDED BY

Miss K. RAMAPRABHA, B.E.

*Submitted in partial fulfilment of the
requirements for the award of the Degree of
Bachelor of Engineering in Electrical and
Electronics Engineering of the
Bharathiar University*

Department of Electrical and Electronics Engineering
Kumaraguru College of Technology

COIMBATORE-641 006.

Department of Electrical & Electronics Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

Certificate

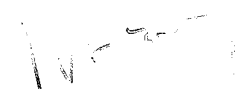
This is to certify that the report entitled
COMPUTER AIDED SPEECH SYNTHESIZER has been submitted by

Mr. S. MAHESHVARAN, B.E., MCA, JNTU, Hyderabad

In partial fulfilment for the award of Bachelor of Engineering in the
Electrical and Electronics Engineering branch of the Bharathiar University,
Coimbatore - 641 046 during the academic year 1991-92.



Guide



Dr. K. A. PALANISWAMY, B.E., M.Sc. (Engg.), Ph.D.
M.T.E.C., Engg (I), FIE.,

Professor and Head

Department of Electrical and Electronics Engineering,

Kumaraguru College of Technology,

Coimbatore - 641 006

Head of the Department

Certified that the Candidate with University Register No. was
examined in Project work Viva-Voce by us on

.....
Internal Examiner

.....
External Examiner

SYNOPSIS

The project aims at speech synthesizing by means of the speech synthesizer chip SP-0256 which employs linear predictive coding. This chip has phonemes, the basic unit of sound stored in its internal ROM. The combination of phonemes to form words and the combination of words to form sentences are controlled by the computer. The speech synthesizer module containing the buffer, the speech synthesizer chip, filter, audio amplifier and load speaker is connected to this computer through centronix interface. A software using C language has been developed to get the phonemes from the user and supply to the hardware thro the centronix interface. A menu has also been developed for a visual display of the phonemes entered.

CONTENTS

CHAPTER		PAGE	NUMBER
	CERTIFICATE		
	ACKNOWLEDGEMENT		
	SYNOPSIS		
	CONTENTS		
I	INTRODUCTION		1
	1.1 HOW SPEECH IS PRODUCED		
	1.2 CLASSIFICATION OF SPEECH SYNTHESIS		
	1.3 DIGITAL MODEL OF SPEECH SYNTHESIS		
II	DESIGN OF HARDWARE		10
	2.1 LPC SPEECH SYNTHESIS		
	2.1.1 SP-250		
	2.2 BLOCK DIAGRAM		
	2.2.1 BUFFER		
	2.2.2 SP0256 AL2		
	2.2.3 OUTPUT STAGE		
III	DESIGN OF SOFTWARE		19
	3.1 DEVELOPMENT		
	3.2 GENERATION OF PHENOMES CODES		
	3.3 PROGRAM LISTINGS		
IV	INTERFACING		25
V	CONCLUSION		26
	REFERENCE		
	APPENDIX		29

CHAPTER I

INTRODUCTION

In every language there is sound system consisting of consonants and vowels. Any particular word has got different types of pronunciation. Each sound, the presence or absence, will make a difference in meaning of a word is called a 'PHONEME'. Language whether it is written or spoken is made up of words (meaningful sound combinations) represented by phonemes joined together according to the rules of syntax of the particular language.

In language like Tamil, the writing system (alphabet) is such that each letter represent a phoneme. But in many other language like English there is no one to one correspondence between the letteres and phonemes. In such cases written representation of the phonemes is affected by

1. using some letters for equal number of sounds (eg) the letters P,L,M,N etc have a specific sound of their own.
2. using combinations of two or more letters for certain sounds (eg) the combination of 'C' and 'H' gives the first sound in "CHILD"
3. using the same letter for different sound in different words (eg) 'G' in 'Gin' and "G" in "Gun".

In this project we have developed a hardware which uses SP0256, AL2 74LS244 as a buffer, an analog filter, an amplifier and loud speaker to produce the speech sound according to the phonemes supplied.

A software have also been developed to get the phonemes from the user's terminal and to feed the hardware with the phonemes provided. The hardware being connected to the printer port of the personal computer through a Centronics interface.

1.1 HOW SPEECH IS PRODUCED

In normal speech production the chest cavity expands and contracts to force air from the lungs out through the treachea past the glottis. The sounds of speech are commonly divided into two types namely voiced and unvoiced.

Fig 1.1 indicates the model of human throat. If the vocal tracts are tensed as for voiced sounds like vowels they will vibrate in the mode of a relaxation oscillator modulating the air into discrete puffs or pulses. If the cords are spread apart the air stream passes through the paryux cavity; past the tongue and depending on the position of the trap door velum either through the mouth or nasal cavity. The air stream is expelled either thro the mouth or by the nose or by both and is preceived as speech. In the case of unvoiced sounds the vocal cords are spread apart and one of the two conditions prevail. Either a turbulent flow is produced as the air passes thro a narrow constriction in the vocal tract or a brief transient excitation occurs, following a build up of pressure behing a point of total closure along the tract. As the various articulators (eg) lips, tongue, jaw, velum change their position during continuous speech. The shapes of the various cavities undergo a drastic change.

Any sound can be made voiced one by making the vocal cords vibrate, while that sound is being produced nasalisation is done by allowing the air to escape partly through the mouth and partly through the nose consonants are aspirated by sending out an additional puff of air with the unaspirated sound. Fricative sounds are produced by squeezing out the air between the tongue and the upper part of the vocal chamber.

The vocal cords are a buzz source that whose fundamental frequency is approximately 100Hz and the harmonics frequency are essentially of equal amplitude. The vocal tract (ie throat and mouth) is a resonating chamber that can be tuned to different sets of frequencies by moving the tongue and lips. Each resonating frequency of a sound is called a formant. In meaningful speech the voiced or unvoiced sound is shaped by the resonant formants into a complex waveform called an allophone.

MODEL OF HUMAN THROAT

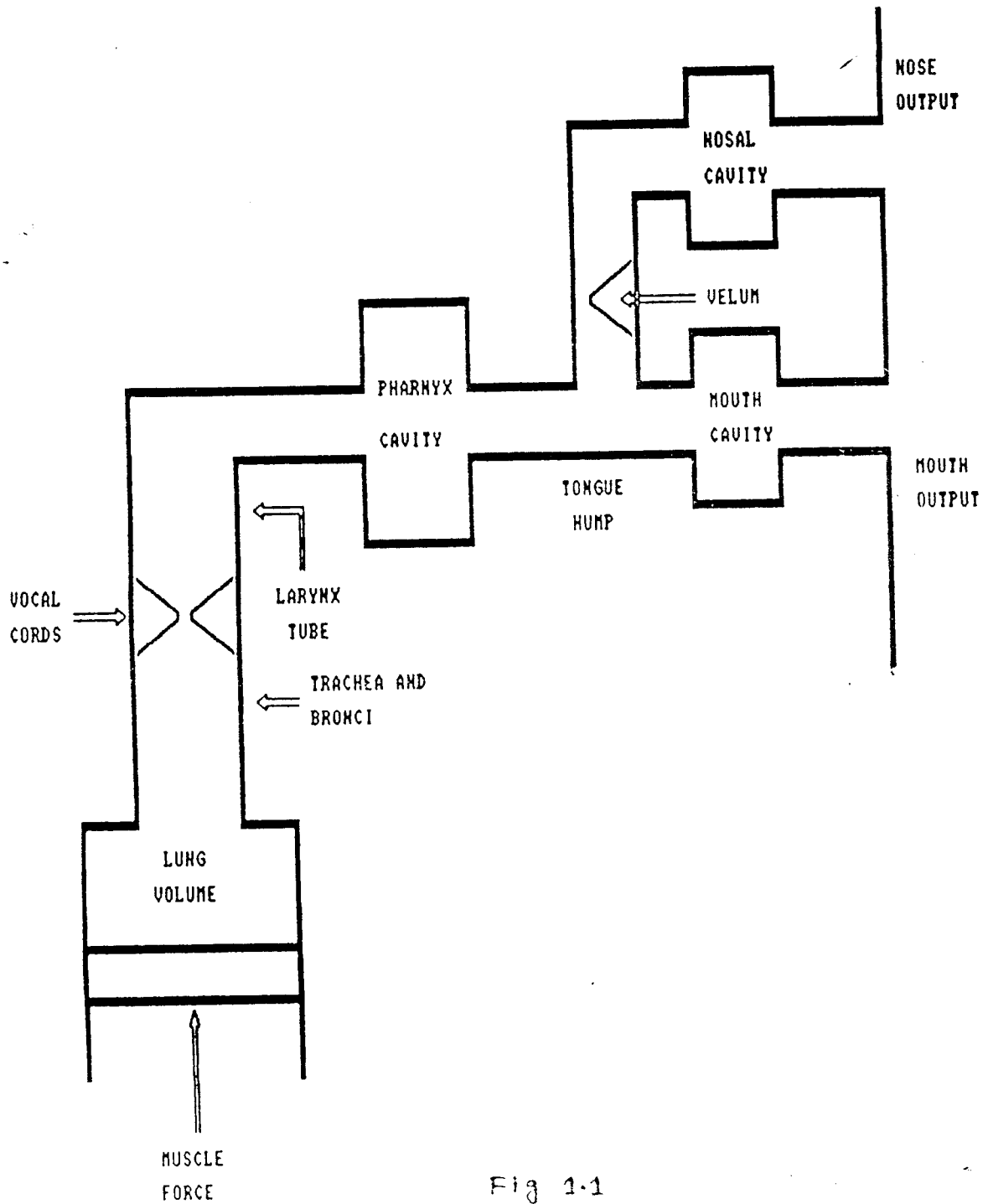


Fig 1-1

1.2 CLASSIFICATION OF SPEECH SYNTHESIS

The methods of speech synthesis can be broadly classified into 3 main categories

- i) Waveform coding
- ii) Hybrid coding
- iii) Source coding

i) Waveform coding:

This method is concerned with simply preserving the waveshape of the analog speech signal thro sampling and quantisation process. The waveform can be stored using techniques like pulse coded method, differential pulse coded method or adaptive differential pulse coded method etc

Advantages :

The advantages are

1. The quality of voice is good
2. It is easy to store the speech with minimum hardware

Disadvantages :

The main disadvantages of this method is that it requires a lot of memory

ii) Source coding:

Source coding are concerned with representing the speech signals as the coutput of a model for speech production. Then the parameter obtained for representing speech production model are conveniently classified as either

excitation parameters or vocal tract response parameters. One of the method of source coding is linear predictive coding and this is the one we have employed in our project work.

Advantages :

1. Source coding methods require less memory
2. Concatenation can be done easily

Disadvantages:

1. Parameter estimation is very difficult
2. It requires a lot of hardware

iii) Hybrid coding:

This is a combination of both the waveform coding and the source coding. This method is concerned generally with representing the vocal tract response by the source coding and the excitation source by waveform coding.

1.3 DIGITAL MODEL OF SPEECH SYNTHESIS

The basic assumption of almost all speech processing systems is that the source of excitation and the vocal tract system are independent. It is this source system independence that allows us to discuss the transmission function of the vocal tract and to let it be excited by any of the possible sources. The validity of the assumption above is quite good for the majority of cases of interest. There are some cases however when the assumption is invalid and the basic model breaks down, such as during transient sounds like p in pot,. based on the ideas above a simple digital model of speech production is shown in fig 1.2. The sources of excitation consists of an impulse generator (controlled from the outside world by the pitch period signal) and a random number generator. The impulse generator produces an impulse (corresponding to the initiation of a puff of air) for every samples. This duration is referred of the pitch frequency or rate of oscillation of the vocal cords.

The random number generator output simulator both the quasi-random turbulence and the pressure building waveform for unvoiced sounds. Either or both of these sources may be applied as input to a linear time varying digital filter. This filter simulates the vocal tract system and thus the filter coefficients specify, in some manner, the vocal tract as a function of time during continuous speech.

Finally a gain control between the source and system allows a certain flexibility in acoustic level of the output. The digital waveform at the output of the filter corresponds to the final speech output, sampled at the appropriate rate to control the model above requires a knowledge of the appropriate parameters. (pitch period, switch positions, amplitude and filter coefficients) as a function of time. This is the goal of almost all speech analysis system ie. to estimate the appropriate model parameters from real speech. The goal of most speech synthesis systems is to use these parameters obtained in any reasonable manner to derive a synthetic speech signal that is indistinguishable.

DIGITAL MODEL OF SPEECH SYNTHESIS

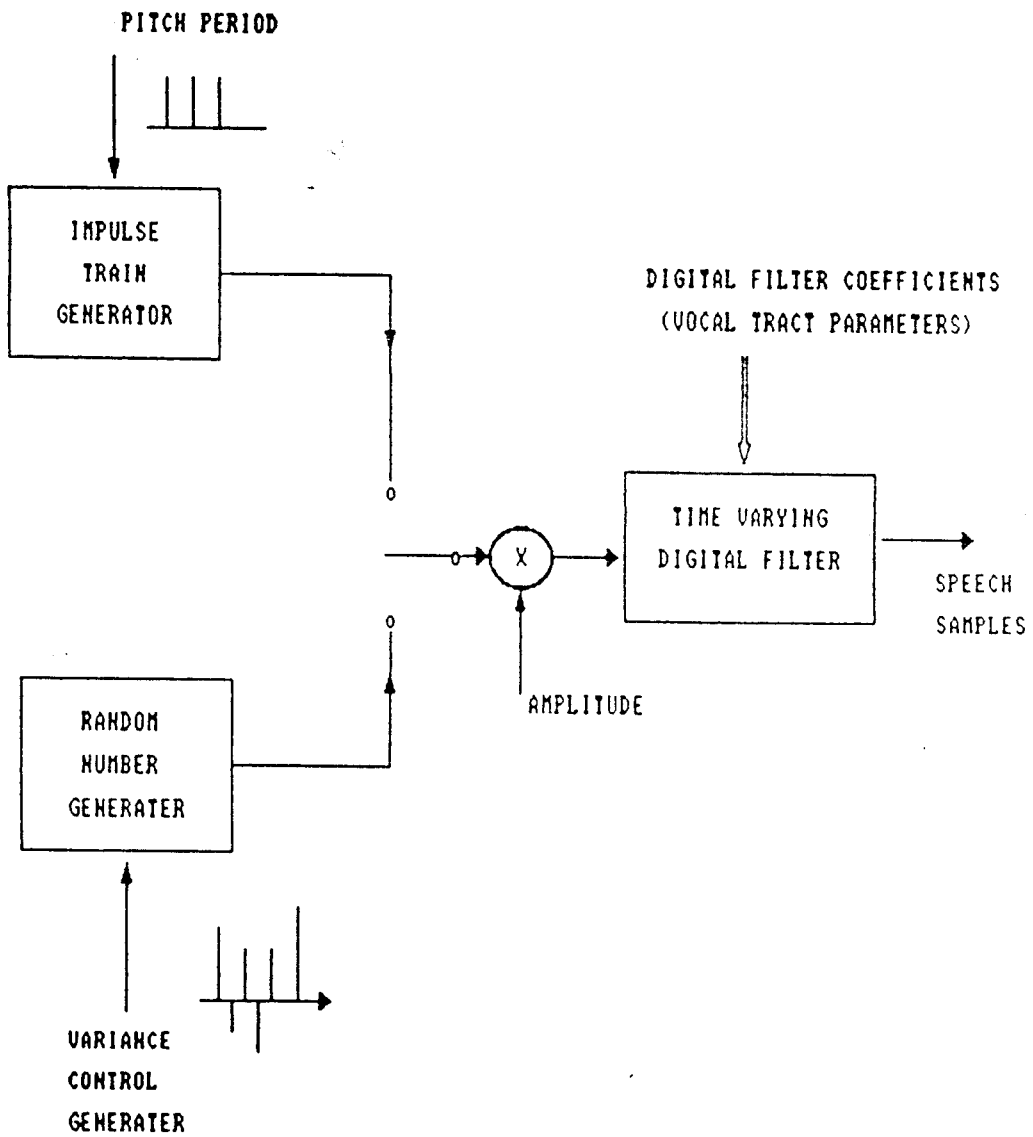


Fig 1.3

CHAPTER 2

DESIGN OF HARDWARE

2.1 LPC SPEECH SYNTHESIZER

Before analysing the various speech synthesization, some definitions have to be understood.

Frame length : This is the number of times per second that the synthesisation coefficients are updated.

Coefficient precision : This is the number of bits used to encode the various parameters.

Audio buffering and low pass filter requirements : To avoid a mechanical sound to smother the speech, reconstructed from digital data an analog filter must be used.

Typical data rate : The data rate for any given chip can vary widely depending on how many coefficients are used and how much editing time is spent on data compaction.

THE SP 250

The SP 250 from general instrument was one of the early synthesizer chips. Its vocal tract filter uses a series of six cascaded simple digital filter with each of the second order filter specifying one resonance of the voice sound.

One appealing aspect of this filter structure is that we can give an initiative meaning to each of the parameters sent to the filter. Two bytes specify each filter section.

One corresponds to the formant frequency while the other corresponds to the formant's bandwidth and amplitude. This intuitive meaning might be useful for implementing special effects or for reducing the basic data rate by specially coding the frequency and format value (which often change fairly slowly) and expanding these coded values to their full values before sending them to the synthesizer.

Each coefficient in this device can be specified with 8 bits of precision coefficients are updated after each pitch period. So the frame rate and the final data rate depend on the number of filter section used the pitch period and how many times the coefficient are repeated. The bit rate for the SP 250 is 350 to 600 bytes per second depending on how many coefficeints are used and how fancy the coefficients are coded.

The audio output from this chip is not an ordinary D/A converted output ; it is a pulse width modulatd TTI signal.

This signal must be low pass filtered and sent through a power amplifier IC before driving a speaker.

2.2 BLOCK DIAGRAM

The block diagram and circuit diagram are shown in fig 2.1 and fig 2.2 respectively . The main blocks of the ckt are explained as follows

2.2.1 Buffer :

Here 74LS244 have been used as the buffer chip. Th pin out diagram is shown in fig 2.2.1. The octal buffer 74LS224 is a typical example of a tristate buffer. When the enable line is low, the circuit functions as a buffer otherwise it stays in high impedance state. It is used to increased the driving capacity of the address bus in bus oriented system.

2.2.2 SP0256 AL2

The pinout diagram and block diagram is shown in fig 2.2.2 and 2.2.3 respectively. GI's SP-256 evolved from the SP250. The SP250 has an internal ROM that can be used for a custom vocabulary of greater interest to less ambitious. Users is GI's version of the chip that has a phoneme library on this ROM.

Beyond the speech synthesis circuits this chip has a simple front end microprocessor control. The SP256 can get voice data only from its internal ROM or from a special external voice ROM A microprocessor can sent voice data directly to the device but it sends the starting addresses to the chip and the synthesizer then goes to its own internal

or off chip local memory to begin reading a voice data. A stand by pin puts the chip in a power down mode which cuts its power requirements. The input of the chip can be connected to any standard centronics interface as only the phoneme data are transferred. The total data flow is very small on average eight bytes are sufficient for one second speech. The output from this chip is filtered through a low pass filter and given to a simple audio amplifier.

2.2.3 OUTPUT STAGE

Low Pass Filter:

In order to avoid external mechanical noise, and to smoothen the speech, reconstructed from the digital data, an analog low pass filter is constructed.

Amplifier unit:

An amplifier unit is constructed with LM386 chip. This unit is used to amplify the output signal from the low pass filter before driving the speaker.

Speaker:

To provide an audio output of the amplified signal an 8 ohm 1 watt speaker is used.

B L O C K D I A G R A M
O F
C O M P U T E R A I D E D S P E E C H S Y N T H E S I S

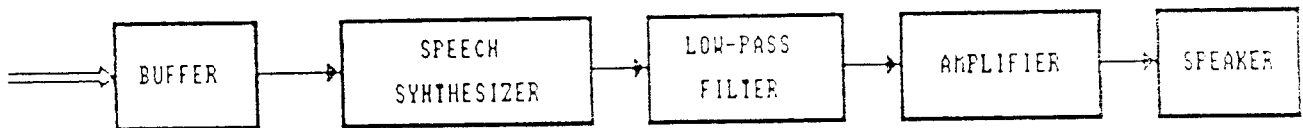


Fig 2.1

CIRCUIT DIAGRAM

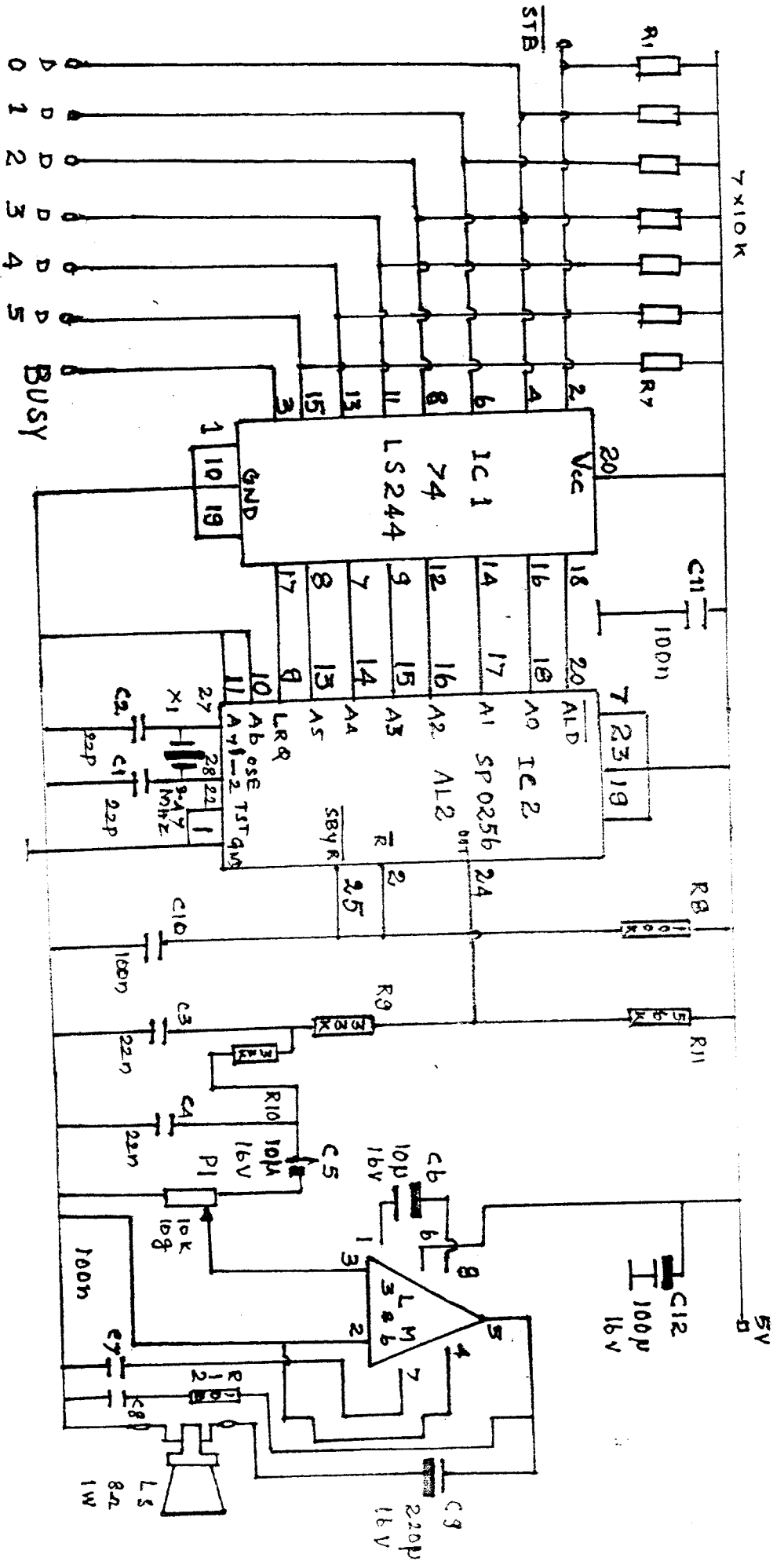
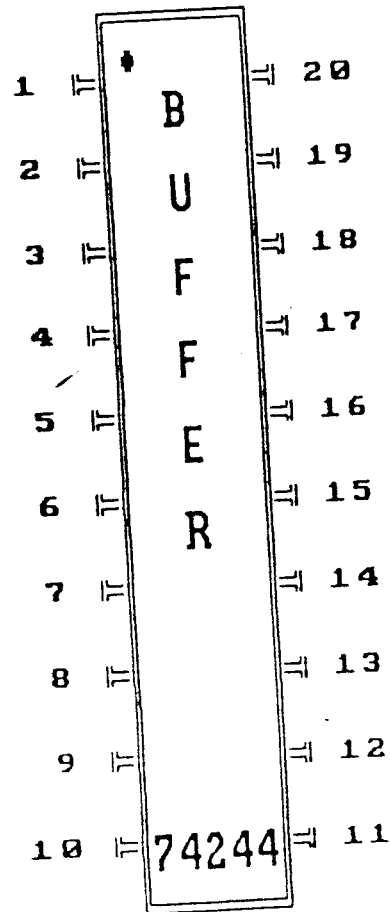


FIG 2.2

TRI - STATE BUFFER



PIN DETAILS

1	$\overline{1G}$ ENABLE FOR I NIBBLE	11	I/P FOR U	LINE	
2	I/P FOR I	LINE	12	O/P FOR IU	LINE
3	O/P FOR UIII	LINE	13	I/P FOR UI	LINE
4	I/P FOR II	LINE	14	O/P FOR III	LINE
5	O/P FOR UII	LINE	15	I/P FOR UII	LINE
6	I P FOR III	LINE	16	O/P FOR II	LINE
7	O/P FOR UI	LINE	17	I/P FOR UIII	LINE
8	I/P FOR IU	LINE	18	O/P FOR I	LINE
9	I/P FOR U	LINE	19	$\overline{2G}$ FOR HIGHER NIBBLE	
10	GND		20	$U_{CC} [+5V]$	

Fig 2.2.1

25

PIN DIAGRAM OF THE SP-0256A

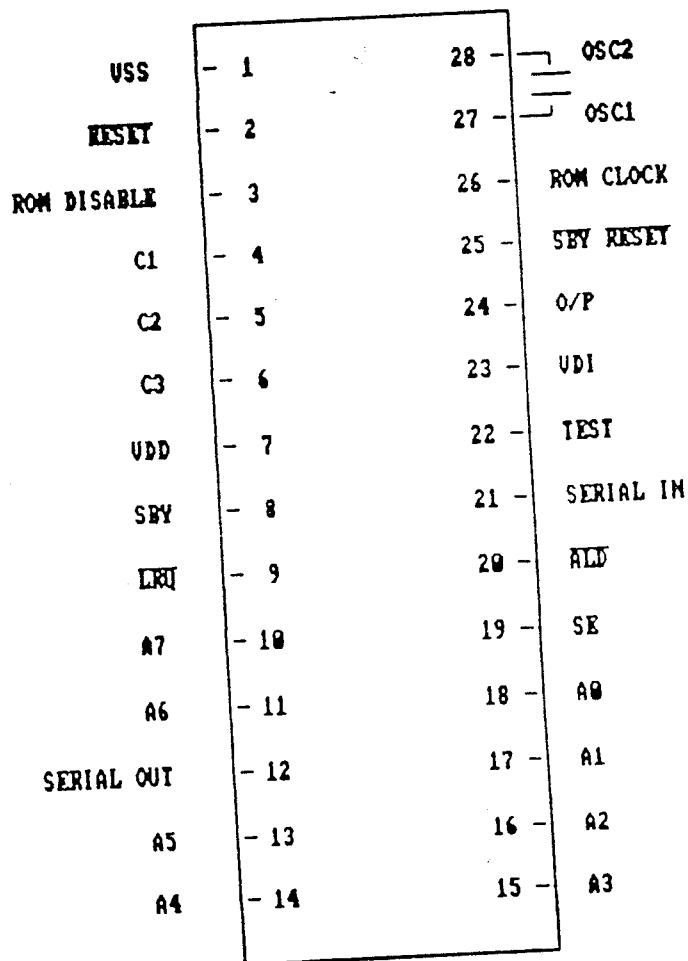


Fig 2.2.2

BLOCK DIAGRAM OF SP0256-AL2

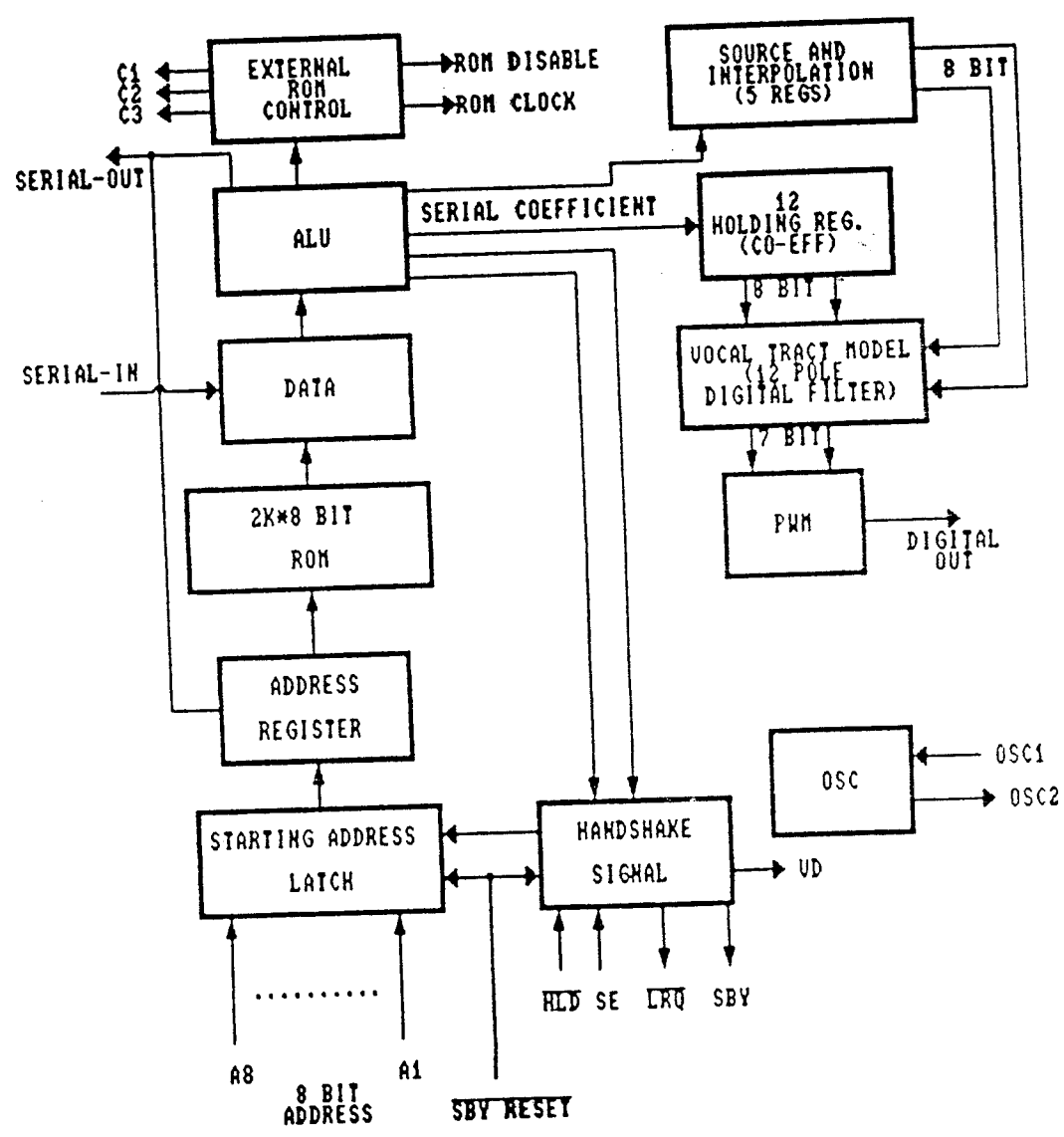


Fig 2.2.3

CHAPTER III
DESIGN OF SOFTWARE

3.1 DEVELOPMENT OF SOFTWARE

A software program has been written in 'C' for accepting characters from the keyboard. The language C is selected for that it allows to be used at a lower level, thus bridging the gap between machine language and the more conventional highlevel language. C is characterised by the ability to write a very concise source programs, due in part to the large number of operators included within the language.

A database in the form of a file is created which contain the information regarding the display format for all the characters menioned in the phoneme table (Appendix I). When a key is pressed, the program calculates the memory location of the display format for that particular character in the database. The character is displayed on the screen in this format. Before displaying the character, the type of the character must be checked for vowel or consonant. If the key pressed is a vowel then it is display as such . But if a consoannt key is pressed, then display of this consonant is postponed until the arrival of the next character. If the next character is also a consonant then the first character is displayed.

3.2 GENERATION OF PHONEME CODES:

Yet another database is created which contains the phoneme codes. This database contains two parts, the character and phoneme code for that character. The phoneme code represents the location in the ROM of the speech synthesizer chip. When a key is pressed, after displaying the character an access is made to the phoneme table to find the phoneme code for that character. This code is sent to the printer port because it is at the printer port that the speech synthesizer module is connected.


```

#include "my.c"

int tog=1;

main()
{
    struct WORDREGS
    {
        unsigned int    ax, bx, cx, dx, si, di, cflag, flags;
    };

    struct BYTEREGS
    {
        unsigned char    al, ah, bl, bh, cl, ch, dl, dh;
    };

    union    REGS    {
        struct    WORDREGS    w;
        struct    BYTEREGS    b;
    };

    union REGS    reg;
    char ch[100],gc='1';
    int i,n;
    menu();
    n=1;
    ch[0]=5;
    while (gc!=0x1b)
    {
        locate(16,40);
        clrscr();
        printf("%c",gc);

```

```

if (gc=='1')
{
clrscr();
locate(5,1);
printf ("Phoneme in memory ");
for (i=0;i<n;i++)
    printf ("%d ",ch[i]);
locate(10,10);
printf("Enter the no of phoneme: ");
scanf("%d",&n);
locate(12,10);
printf("Enter the phonemes ");
locate (14,10);
printf("Phoneme No");
for (i=0;i<n;i++) {
    locate(14,22);
    clrscr();
    printf("%d ",i+1);
    if (tog)
        scanf("%d",&ch[i]);
    else
        scanf ("%x",&ch[i]);
    ch[n]=0;
    menu();
}
}

```

```

if (gc=='2')
{
locate (20,1);
clreol();
printf("Speaking...");
for (i=0;i<n;i++)
{
reg.h.ah=0;
reg.h.al=ch[i];
reg.x.dx=0;
int86(0x17,&reg,&reg);
}
reg.h.ah=0;
reg.h.al=0;
int86(0x17,&reg,&reg);
locate(20,1);
clreol();
}
locate(20,1);
printf("Phoneme in Memory ");
locate (20,20);
for (i=0;i<n;i++)
if (tog)
printf("%d ",ch[i]);
else
printf ("%x ",ch[i]);
locate(16,40);
clreol();
gc=getch();

```

```

if (!gc)
    { gc=getch();
      if (gc=='D') tog=1-tog;
    }

locate (21,60);

if (tog)
    printf ("DEC");

else
    printf ("HEX");
}

}

menu()
{
clrscr();

locate(10,20);
printf("1. Enter the phoneme " );
locate(12,20);
printf("2. Speak");
locate(14,20);
printf("Esc Quit ");
locate(16,30);
printf ("Option");
locate (21,60);

if (tog)
    printf ("DEC");

else
    printf ("HEX");
}
}

```

CHAPTER IV

INTERFACING

The standard centronics interface is used for connecting the speech synthesizer module to the computer. The data input to the SP-0256 and the handshaken signals are separated from the PC by a buffer. The buffer used here is 74LS244 octal buffer which has eight input lines and eight output lines. Among the eight data bits coming out from the PC, one bit is the strobe signal, another one is the acknowledge signal and rest of the 6 bits correspond to a location in the ROM of the SP-0256. As the data is strobed in, the speech synthesizer starts synthesizing and this is indicated to the PC by making the busy line high. After synthesis the busy line becomes low at which time the next data is synthesized. The synthesizer chip requires a crystal of 3.47 MHz for its clock. The output of the audio signal is pulse width modulated. It is converted into analog signal by low pass filtering and amplifying and the audio output is obtained at the speaker.

CHAPTER V

CONCLUSION

A software has been successfully developed to pick up phonemes from the PC to SP0256 AL2 chip. The interfacing module is fabricated and tested. The phoneme code have been identified for different words.

The speech synthesizer chip used in this model do not have provision for pitch. This chip SP-0256 AL2 was mainly designed for British and American English. It has 64 phonemes in its ROM. Each phoneme has different duration which does not match for all other languages except English. Software can be developed to get the speech in any other language. In order to improve the quality of speech we can develop and use our own phoneme libraries instead of the one available. This library helps in choosing appropriate coding technique that will suit the utterance patterns of the language selected and the basic unit of speech system can also be varied. The project can be improved by developing software in Assembly language which would have been faster in displaying the characters than the program in C.

This project model is very useful for blind. When the text is typed, the corresponding speech is produced. So when they have pressed the wrong key, it can be corrected immediately. In the field of data entry in large amounts, there is possibility of wrong entry. Hence after entering the

whole data yet another person has to check again to correct the mistake. But if the speech synthesizer module is also implemented then it becomes easy for the operator and time can be saved. When interfaced to a computer which controls a lot of machines, it is possible, in case of a fault, to make the computer to announce where the fault is, so that it is easier for the common man to understand.

REFERENCES

1. BYRON S. GOTTFRIED - 'Programming with C' - Mc Graw Hill
Schaum's outline series
2. RAMESH GUPTA - 'Voice through Silicon chip' Electer India,
Apr. 1985.
3. RAMESH S. GOANKAR - 'Microprocessor architecture Programming
and applications with the 8085/8080A' 18th edition, 1991.
4. CRAIG BOLON - 'Mastering C' BPB publication, first edition,
1988.
5. BRAND KIM JON - 'Common C functions' Que corporation, 1985.
6. DOUGLAS V.HALL - 'Microprocessor and interfacing programming
and hardware' Mc Grew Hill international edition, second
edition, 1987.
7. LEVENTHAL '8080, 8085 Assembly language programming' Tata
Mc Grew Hill, New Delhi, Second Edition, 1986.

PHONEME TABLE

Decimal Code	Hexa Decimal Code	Allophone	Duration (ms)	Representative Sound (Bold letters)
00	00		10	PAUSE
01	01		20	PAUSE
02	02		50	PAUSE
03	03		100	PAUSE
04	04		200	PAUSE
05	05	OY	290	BOY
06	06	AY	170	FIVE
07	07	EH	50	LEFT
08	08	KX3	80	COUNT
09	09	PP	150	PEAK
10	0A	JH	100	JUMP
11	0B	MH1	170	NONE
12	0C	IH	50	IT
13	0D	TI2	100	TO
14	0E	RR1	130	RIGHT
15	0F	AX	50	TROUBLE
16	10	MM	180	MAGNET
17	11	TI1	80	PART
18	12	DH1	140	THEY
19	13	IY	170	SEE
20	14	EY	200	STAY
21	15	DD1	50	CARD
22	16	UW1	60	COMPUTER
23	17	AO	70	LONG
24	18	AA	60	HOT
25	19	YY2	130	YARD
26	1A	AE	80	MAH
27	1B	HH1	90	HE
28	1C	BB1	40	TROUBLE
29	1D	TH	130	THIN
30	1E	UH	70	PUSH-PULL
31	1F	UW2	170	FOOD
32	20	AW	250	SOUTH

Decimal Code	Hexa Decimal Code	Allophone	Duration (ms)	Representative Sound (Bold letters)
33	21	DD2	250	DO
34	22	GG3	120	JIG
35	23	VV	130	VERY
36	24	GG1	80	GO
37	25	SH	120	SHIFT
38	26	ZH	130	MEASURE
39	27	RR2	80	BRING
40	28	FF	110	FOR
41	29	KK2	140	SKIP
42	2A	KK1	120	ASK
43	2B	ZZ	150	ZERO
44	2C	NG	200	TALKING
45	2D	LL	80	LOOK
46	2E	WW	140	WIRE
47	2F	XR	250	DEAR
48	30	WH	150	WHERE
49	31	YY1	90	YES
50	32	CH	150	CHIP
51	33	ER1	110	COUNTER
52	34	ER2	210	TURN
53	35	OW	170	SLOW
54	36	DH2	180	LATHE
55	37	SS	60	STOP
56	38	NN2	140	NO
57	39	HH2	130	HERTZ
58	3A	OR	240	STORE
59	3B	AR	200	ARM
60	3C	YR	250	CLEAR
61	3D	GG2	80	GLUE
62	3E	EL	140	ANGLE
63	3F	BB2	60	BIT