CHAPTER 3

SYSTEM USED FOR
THE EXPERIMENT
CHAPTER 3

SYSTEM USED FOR THE EXPERIMENT

3.1 INTRODUCTION

Most of the modern signal analysis and processing is performed by digital computers. Using a small general purpose digital computer one can apply a wide variety of digital signal processing techniques to the speech communication problem. The complexity of analog circuit is very high compared to the digital processing. Moreover, the digital data is less prone to noise. In any speech analysis problem, one always faces the non-availability of speech data in a suitable digital form. Usually a large data base is required for speech experiments. Most preferably, the data should be available in the same system.

A good solution for this is, a digitizing system designed to work with the computer, used for the processing. So, the same computer can control the digitizer and get on-line data in the required form and can also store the data in its own memory. Ultimately one can think of a computer system, in which one function key converts analog speech to digital data and another key activates on-line processing and a third key converts the processed data into analog speech and so on.

55
In an application like speaker identification and verification, user expects a quick response from the system. Considering the huge amount of computational analysis and large number of comparisons involved in the problem, the system used for the purpose must be fast enough. Though there are not many real time identification or verification systems available, one expects a reasonably fast response and result. Availability of faster algorithms for analysis is another major issue. Above all, one has to think of an economic system with all these facilities. Taking all these factors into consideration, a system built around a personal computer (PC) is an optimized solution. It is the most inexpensive computer, with fast processing capabilities. There are many fast software algorithms, specially suited for digital signal processing, available in PC. Besides, the development of hardware on a Personal Computer is easy due to its open architecture.

3.2 DIGITAL CODING OF SPEECH

There are different schemes for representing the digital speech viz., PCM, DM, DPCM and ADPCM. Among all these coding schemes, PCM being direct and simple, is chosen for this present system. A digital representation of analog speech is always discrete in both time and amplitude. The sampling operation makes the analog signal discrete and the quantization in amplitude makes the signal completely digital.
For a faithful representation of speech signal, at least a 12 bit resolution is necessary which gives a reasonably good SNR also. Using a 12 bit digitizer along with a 16 bit computer makes the system faster and gives a good quality representation of speech. In order to check the quality of speech before and after processing, it is desirable to reconstruct the digital data to analog signal. For this, an 8 bit DAC is used with the system. Thus the whole system with a 12 bit ADC, 8 bit DAC and an IBM PC/AT gives the power of a speech workstation.

3.3 SYSTEM OVERVIEW

The block diagram of the system is shown in Fig.3.1. In this speech digitizing system one of the expansion slots in a Personal Computer is used with an interfacing card. The address lines from the computer are decoded in such a way that it avoids clash with standard I/O addresses used in computer. In the digitizing circuit, the microphone input is amplified properly and band limited to 4 KHz by an active low pass filter and fed to the ADC through a Sample and Hold circuit. Since the system was designed for PC and PC/XT, initially the 12 bit data had to be read into the computer through 2 latches and in two cycles to match the 8 data lines of 8088 CPU. The sampling rate of the ADC is controlled by feeding pulses to the Start of Conversion (SOC) pin of ADC. The End of Conversion (EOC) pin
Fig. 3.1 Block Diagram of Speech Digitizer.
is connected to the Sample/Hold pin of Sample and Hold circuit, so that the data will not change during conversion. The digitized data is stored in the main memory of the computer in a separate extra segment. This data can be viewed by plotting on the screen or listened back after reconstruction. For the reconstruction of digital data, a DAC is connected through a latch and the output of DAC is lowpass filtered and after proper amplification fed to an output speaker. The selected portion of data can be stored on to the secondary storage unit viz., Floppy Disk or Winchester, for later use of data.

The complete control program for the system was developed in a mixture of high level language and assembler. This gives the user an easier way of programming. One can make use of the memory management, graphic features and other advanced facilities available in the higher level language. At the same time, using the assembler enables, fine control of devices, which is not possible through high level language. Through the software, it is possible to select the sampling rate of data as well as the reproduction rate of output speech after reconstruction. Since 8088 and 80286 processors have segment addressing, the data can be stored in different segment addresses available in PC. Thus a large amount of data can be stored. At a normal sampling rate of 8 KHz, one segment of memory can hold about 4 sec. of speech. Data stored in the memory can be examined by graphically plotting
it on the graphic screen of PC, which is easily done through high level language. The system software developed is very powerful, that the system can handle many functions with the help of available hardware and deliver the power of a fairly good speech workstation.

3.4 HARDWARE DESCRIPTION

Only essential hardware is developed for this system. Special care has been taken to make use of all available facility of the PC. So, the hardware developed for this system is sharing most of the resources in the PC. The extra hardware developed is kept outside the PC and interfaced with an interface card plugged in the expansion slot of the PC mother board. On this interface card, we have bidirectional gated tristate buffers for data lines. This precaution has been taken to prevent any clash with the normal operation of the computer. These gated tristate buffers are enabled only in the selected address range by a decoder circuit. The I/O addresses are also chosen with special care to avoid clash with the standard I/O peripherals used with the PC. The range selected for this particular I/O operations is 8300H to 830FH, which is not used by any other devices. All the address lines are buffered and address range selection is done through a small logic circuit. An XUSER signal is generated using logic gates, which will be active only in the selected range of address. The interfacing circuit is shown in Fig.3.2.
The least significant 4 bits of address lines along with the \( \overline{\text{USER}} \) is decoded using a four-to-sixteen decoder, which generates 16 separate addresses in the range 8300H to 830FH. Then these decoded address lines along with \( \overline{\text{IN}} \) and \( \overline{\text{OUT}} \) signals from the computer are used to generate separate IN and OUT addresses using an OR gate. Thus the addresses 8300H through 8303H are decoded as input ports and 8304H through 8307H are decoded as output ports. These address lines control and monitor the devices and operations of the system. The decoder and address generation is shown in Fig.3.3.

In the digitizing circuit, the microphone is the front end. The signal from this microphone, which is only of the order of a few milli volts is amplified to appropriate level and fed to the Analog-to-Digital Converter. This ADC is working at bipolar ±10 volts, so that the level of the signal should be amplified using an IC 741 operational amplifier. Two 741 IC's are used for this amplification and a gain control potentiometer is provided in the second stage of the amplifier for proper control of input signal level. A second order Butterworth active low pass filter is designed for bandlimitting this amplified input signal at 4 KHz. This filter circuit is also designed based on an IC 741 op-amp. The band limited signal, which is the output of the filter circuit is now fed to a Sample and Hold circuit. An LF 398 along with a holding capacitance constitute a S/H circuit.
Fig. 3.3 Control Circuit - Decoder and Address Generator.
Binary (COB) for bipolar input signal ranges. In this present system the input range is selected as $\pm 5$ V and the output mode as Complementary Two's Complement (CTC).

Once the analog signal is digitized by the ADC, it has to be transferred to the computer. The EOC line status is checked by the computer by periodically scanning this line which is connected to a data line through a tristate buffer. Once the EOC signal is issued by the ADC, the processor senses it and then steps are taken to read the converted data into the memory. Since the card was originally designed for PC and PC/XT, the data width is considered to be 8 bit. Hence the 12 bit data from the ADC had to be read into the computer in two cycles. This is done by connecting this output lines from ADC to two latches and the output of the latches are connected to the 8 bit data bus. Two 8282, 8 bit bipolar latches with tristate output buffer are employed for this purpose in the circuit. The two latches are write enabled simultaneously by the EOC signal from ADC and the output data from the latches are enabled separately. The data read from the digitizer is appropriately stored in the main memory of the computer at a pre-specified extra segment. The conversion continues till the entire segment gets filled. Hence at a sampling frequency of 8 KHz, it can hold 4 seconds of 12 bit data. The complete digitizing circuit is shown in Fig.3.4.
Fig. 3.4 Analog-to-Digital Conversion Circuit.
The logic control input of the Sample and Hold is connected to END OF CONVERSION (EOC) pin of the ADC chip, so that the data will be held unchanged during the conversion of the analog signal by the ADC. The output of the Sample and Hold circuit is fed to the analog input pin of the ADC. The ADC used for this is Burr Brown ADC 85, which has 12 bit resolution and very fast conversion capability with internal clock, comparators, reference voltage and input buffer amplifier. The conversion time of this chip is 10 microseconds. The Start of Conversion (SOC) signal fed from the computer through the output address 8304H controls the sampling rate of the signal. This signal is generated by software. When an SOC is fed to ADC, the EOC pin will go low, which holds the signal in the holding capacitance of S/H circuit and at the same time start the conversion. After the conversion of data, the EOC signal goes high and the S/H circuit will be in sampling mode, which will start charging the capacitance. In the ADC 85, various analog input signal ranges are available with ±2.5 V, ±5 V, ±10 V, 0 to 5 V and 0 to 10 V which can be chosen according to the requirement. Facility is provided to trim externally the gain and offset error. The converted data from the chip can be taken out either in serial or parallel form according to the clock and status signal. The output is available in three different selectable binary codes. They are Complementary Straight Binary (CSB) for unipolar input signal range, Complementary Two's Complement (CTC) and Complementary Offset
There are many options for the user with the digitized data. This data can be checked by plotting or listened back by reconstructing or stored in the secondary storage units for further applications. One can also straightaway go for the analysis of speech. For most of these above mentioned processes, PC facilities are exploited. But for the reconstruction of the digitized data a special Digital-to-Analog Conversion circuit is required. This circuit consists of an 8 bit DAC (National DAC 0800), latches, filter and amplifier. The data bus of the computer is connected to the input digital lines of DAC through an 8282 latch, which is enabled when required using the address 8305H. This DAC 0800 is a high speed, monolithic and current output device. Since the output filter quality was not as demanding as the input filter, a simple RC low pass filter is used for the output analog signal from the DAC, which will band limit the signal at 4 KHz. The function of the filter is to smoothen the quantization effect and to move the high frequency noise from the system. A power amplifier, designed with 810 IC is used in the output side of the filter, which properly amplifies the signal and feeds to the speaker. This reconstruction facility will enable the speaker to check a processed data file for quality enhancement/deterioration by listening back in the subjective testing. Fig.3.5 shows this DAC circuit.
3.5 SOFTWARE FOR THE SYSTEM

The hardware explained in the above section is supported by appropriate software. The software developed for this system is exploiting many inherent facilities of PC using high level and machine level language. Some BIOS interrupts and subroutines are made use of at certain points of software development to make the system a self contained and powerful speech station. The main program is developed in BASIC and when there is some faster operation and some machine level control is required, separate subroutines in ASSEMBLER is developed and linked with the main BASIC program. This is achieved by the facility provided for calling external subroutines in the advanced BASIC available in PC. By developing programs in this mixed level, the programmer need not bother about the file management, screen manipulation, graphics and other complex features of the computer which will be taken care by the Operating System (OS) and compiler. But at the same time fine control is possible whenever required with machine level programs, especially in real time control of I/O devices. The approach adopted is that when a faster and flexible operation is required, separate assembler subroutines are developed and assembled into relocatable object files. The main program is also compiled to an object file. The subroutines are called from the main routine. The LINKER program, available in the PC, links all these relocatable object code files into a single executable file with absolute address.
When subroutine calls are made from the main program, there are some parameters to be passed between the modules. This parameter passing from high level to assembly level and vice-versa is a tricky affair. Instead of passing the parameter, its address is usually passed. The address of the parameter to be passed are kept in the stack just below the 32 bit return address (Segment and Offset: which is done automatically when a call is made from the program). Since no POP operation could be carried out first, the BP is pushed to the stack and then SP is copied to BP as the initial operation of the assembly subroutine. Now pointing the BP appropriately to the parameter addresses by adding number of locations, one can exactly access the parameter from the memory. This can be done by putting the address of the parameter in BX register and moving the contents of the memory points by BX to the required register. While returning from the subroutine, the BASIC program assumes that the parameters are lost from the stack. So to take care of this, one has to ignore the contents of the stack when the parameter address is fixed. Thus a RET 2*N instruction is used in the subroutine, which will discard 2*N bytes from the stack, where N is the number of parameters passed. The BP is popped from the stack before this RET instruction.

The software developed is user interactive. When the main program is invoked, it will display a main menu on the
screen as shown in Fig.3.6, from which the user can choose any function he desires. This menu has six options with six different functions. The functions available in the program are:

3.5.1 Zero Adjustment

This zero adjustment is meant for offset level of the input signal. When this function is invoked, the program jumps into the corresponding subroutine, which scans the data from the digitizer and displays it on the screen at the same time. The program initially issues an SOC signal to the ADC and then waits for EOC signal from the ADC. When an EOC signal is detected, this program will read the digitized data into a previously specified location. This data indicates the offset level of the input signal, without actual signal from the microphone. This is also displayed graphically and numerically on the screen. A scaled area corresponding to this value is displayed on the screen in the shape of a box. This box will have the same size in positive and negative offset, but numeric value displayed will indicate the sign. This complete process is repeated till another key is pressed. The user can adjust the offset to a minimum, bringing the size of the box to a point and thus ensuring the absence of noise in the input. A potentiometer is provided in the circuit for this purpose. Before all data storage sessions, this zero adjustment was made in order to avoid unnecessary noise in the digitized data.
What We can do.

1. Zero adjusting
2. Data acquisition
3. Speech reconstruction
4. Plotting the data
5. Read data from files
0. Exit to DOS

Fig. 3.6 Display of Main menu.
3.5.2 Data Acquisition

This function is responsible for digitizing the analog data and storing it in the appropriate memory location. When the user selects this function, the program will ask for sampling frequency. In this system the sampling frequency can be varied according to the need. Since the speech signal is bandlimited to 4 KHz in the input side, the sampling frequency should be higher than 8 KHz. In some application, 10 KHz is also chosen as the sampling frequency. The data has to be sampled at an interval of 125 \( \mu \text{scc} \) if the sampling frequency is 8 KHz. So a delay has to be introduced between an EOC and next SOC. This delay is generated by software instruction loops, after calculating the execution time of each instruction. Hence the sampling frequency is dependent on this clock frequency of each system. Initially the delay is calculated with a system clock of 6 MHz. If PC is changed, then the sampling frequency has to be scaled according to the system clock. When the users specify the sampling frequency, system produces a beep along with a prompt to wait, while it initializes the devices. The system starts sampling when the beep stops. The program sends out SOC signal with the preset sampling time. When the EOC signal is detected from ADC, the program issues two successive output enable (OE) signals through address lines to latches which take the digitized data into the computer and stores this data in the successive locations.
starting from OOOOH offset address of the pre-specified Extra Segment (ES). This offset address is then incremented after storing the current data till the 64 KB locations get exhausted. If one selects 8 KHz sampling frequency, then 4 sec. of speech can be stored in this 64 KB address available. If needed, similar available ES can be used, with each segment providing a 64 KB memory area. The MSDOS limitation in the main memory restricts this to a maximum size of 512 KB, which can hold about 32 sec. of speech data sampled at 8 KHz. But at present, only one segment is used for data storage. When this memory area gets filled, the program generates another beep along with a prompt message indicating that the time is up. Then the program automatically comes out of the function and displays the main menu.

3.5.3 Speech Reconstruction

The speech reconstruction function will enable the user to check whether the data stored is properly digitized by listening back the reconstructed speech. One can check the quality of a processed speech, either from the main memory or, by calling it from the secondary storage unit. In this subroutine, provision for reconstructing the whole segment of data or a selected portion of data is given. The program asks for number of samples to be reproduced and the starting point of the data so that exact desired portion can be listened to.
In this program also, the user has to specify the sampling rate at which rate the data is fed to the DAC input through a latch. The required delay between samples is generated by this subroutine. At present, an 8 bit DAC is used in this circuit. Hence the least significant four bits are truncated in the computer, and only the resultant 8 bit data is fed to the DAC. Facility is provided in this subroutine to plot or save the data which is reconstructed and listened back.

3.5.4 Plotting of Data

This function facilitates the user to visually inspect the data stored in the memory. For this function, the graphic function available in the PC is utilized and a waveform of the speech stored in the memory is displayed on the screen. In this routine also, one can view the complete data or a portion of the data stored in the memory. This selected view provides a sort of zooming facility, which enables the user for a closer examination of speech segments. A grid is also displayed for proper graduation of the waveform, which can be toggled to ON and OFF. The plotted data can be reconstructed for further checking or can be saved into a file.

3.5.5 File Saving

This facility helps the user to store the required portion or the full data files. The digitized data are stored
in the main memory of the computer. After examining this data, either by reconstruction or by plotting, the required portion is stored permanently in floppy or winchester with file names. The data can be stored in two file formats viz., Binary format and ASCII format. In the binary format, the data are directly stored in the floppy using the BSAVE statement available in BASIC. When the data is to be used by other programs, it should be in ASCII format. In order to store in this format, the corresponding ASCII is calculated by the subroutine and then saved. The user has to specify the file name and the format in which it is to be stored. The ASCII file takes almost 5 to 6 times space compared to the BIN file for the same number of data. But the BIN file cannot be typed on the screen directly or accessed by external programs.

3.5.6 Reading Data from Stored File

This function in the program extends a facility to read stored data file from the secondary storage media into the main memory. The same two formats viz., Binary and ASCII, are available in this function. The data stored in binary format is loaded with a BLOAD statement in BASIC directly into the segment at a specified location. ASCII data has to be reconverted into binary and stored in the main memory. Once the data is read into the main memory all the functions discussed earlier can be utilized. The user can even concatenate speech by loading different data files in successive locations.
3.6 CONCLUSION

The indigenous and economic speech digitizing system described above with all the software and hardware deliver a powerful facility of digitizing and storing the speech data without much information degradation. From the stored data, the features are extracted and used for the speaker recognition experiment. With proper development of software one can modify the system for online recognition purpose.