A MICROPROCESSOR BASED AUDIO TELECONFERENCING SYSTEM

BY

MICHAEL C. NDINECHI
(89EE/00025/G)

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

OF THE

FEDERAL UNIVERSITY OF TECHNOLOGY
OWERRI,

MAY 1991.
ABSTRACT

In this report, the design and simulation of a microprocessor controlled audio-teleconferencing system is presented. Top-down design technique is adopted and the system divided into modules. The modules, with the exception of the power supply and the mixer, interact by means of the 8085A microprocessor.

A mixer circuit is incorporated to combine the inputs from all the microphones to the system and to give the same output to the individual speakers so as to enable participants have discussion in a more natural atmosphere. The present design is intended to accommodate sixteen people in a group discussion with three groups operating simultaneously. However, the system can be readily expanded far beyond this capacity with minimum components increase because it is software based.

"Add-On" bridging technique which allows a conference participant to call up others to add each to the conference is aimed at in the design.

Some modules of the design were implemented and tests carried out in the laboratory to demonstrate their workability. Software development for the system was based on the Intel 8085A microprocessor instructions and the program was written in assembly language. System operation was simulated using the Microprocessor Applications Trainer-MAT 385 and the results show the program to work.
DEDICATION

TO MY LATE BROTHER
SAM. O. NOINECHI
DECLARATION

I hereby declare that this thesis has been composed by me and that it is the record of my work. It has not been presented in any previous application for a higher degree.

All quotations are distinguished by quotation marks and the sources of information are specifically acknowledged by means of references at the end of the thesis.

Ndinechi, Michael Chukwudi
May 1991
ACKNOWLEDGEMENT

I wish to express my appreciation to Engr. (Prof.) C. O. G. Obah and Engr. Dr. J. I. Ejimanya for their advice to pursue a graduate course. Their painstaking supervision has contributed immensely to the success of this work.

My thanks go to Engr. Dr. S. O. E. Ogbogu, Dr. R. S. U. Dike and Dr. F. C. F. Nzeribe for their contributions. Also, to Mr. P. C. Abafar, Mr. O. Nweke and Mr. C. M. Ugoanochie for their assistance at various stages of the work.

My deep gratitude goes to my brothers and sister whose financial support saw me through the graduate work. Also, to Nky who always utters words of encouragement.

Finally to God for his protection and guidance throughout my stay in the University, I bow.
TABLE OF CONTENTS

TITLE PAGE ........................................ 1
ABSTRACT ......................................... ii
DEDICATION ....................................... iii
DECLARATION ...................................... iv
ACKNOWLEDGEMENT .................................. v
TABLE OF CONTENTS ................................. vi
LIST OF FIGURES .................................. ix

CHAPTER ONE:
Introduction ...................................... 1

CHAPTER TWO: LITERATURE REVIEW
2.1 Origin of Teleconferencing .................. 4
2.2 Types of teleconferencing system .......... 4
2.2.1 Video Teleconferencing ................... 4
2.2.2 Electronic Mail and Computer conferencing 5
2.2.3 Audio Conferencing ........................ 6
2.3 Components of Teleconferencing system .... 6
2.4 Impacts of Teleconferencing ................. 9
2.4.1 Positive Impacts ........................... 9
2.4.2 Negative Impacts ......................... 10
2.5 Failure of the existing Teleconferencing Systems ....... 10

CHAPTER THREE: SYSTEM DEVELOPMENT
3.1 Aims and Objectives ......................... 12
3.2 Components Selection ....................... 12
3.2.1 General Considerations ................. 12
3.2.2 Selecting the Microprocessor .......... 13
3.2.3 Keyboard Interface ....................... 14
3.2.4 Memory ... ... ... ... ... 18
3.2.5 Programmable Input/output port ... 21
3.2.6 Buffers ... ... ... ... ... 21
3.2.7 The Clock .. ... ... ... ... 23
3.2.8 Decoders ... ... ... ... ... 24
3.2.9 The Mixer ... ... ... ... ... 24
3.2.10 The Switching Bank ... ... ... ... 25
3.2.11 Power Supply ... ... ... ... ... 27

CHAPTER FOUR: SYSTEM DESIGN AND ANALYSIS

4.1 System Layout ... ... ... ... ... 28
4.2 Processing Unit .. ... ... ... ... 30
4.3 Keyboard Encoding Unit . ... ... ... 35
4.4 Decoding Unit ... ... ... ... ... 37
4.5 Switching Unit ... ... ... ... ... 38
4.6 Mixer Unit ... ... ... ... ... 40
4.6.1 Audio Pre-amplifier ... ... ... ... 40
4.6.1.1 Null Offsetting of Operational amplifiers 46
4.6.2 Voice Activated Switch . ... ... ... 48
4.6.3 Mixer Pre-amp and Audio Amplifier .. 50
4.7 Power Supply ... ... ... ... ... 52

CHAPTER FIVE: SOFTWARE DESIGN AND DEVELOPMENT

5.1 Development Procedure .. ... ... ... 56
5.2 Program Development ... ... ... ... 57

CHAPTER SIX: IMPLEMENTATION AND TESTING

6.1 Implementation ... ... ... ... ... 66
6.1.1 Program Testing .. ... ... ... ... 67
6.1.2 Cassette Interface Operation . ... ... 69
6.2 Problems Encountered 70
6.3 System Operation 71

CHAPTER SEVEN: CONCLUSION AND RECOMMENDATION

7.1 Conclusions 73
7.2 Recommendation 74

References 75
Appendix 79
LIST OF FIGURES

Fig: 2.1 Components of Teleconferencing System.

Fig: 3.1 8085A architecture.

Fig: 3.2 Keyboard switch matrix.

Fig: 3.3 Hardware structure for software scanning of key matrix.

Fig: 3.4 Hardware Debouncing circuit.

Fig: 3.5 Memory classification.

Fig: 3.6 Mixing audio signals.

Fig: 4.1 System Block diagram.

Fig: 4.2 Central processing Unit.

Fig: 4.3 Power-on reset circuitry.

Fig: 4.4 Port initialization information.

Fig: 4.5 Keyboard encoding unit

Fig: 4.6 Decoding unit.

Fig: 4.7 Internal switching arrangement of MC 140538

Fig: 4.8 Audio pre-amplifier.

Fig: 4.9 Op-amp biasing technique.

Fig: 4.10 Voice activated switch.

Fig: 4.11 Mixer pre-amplifier and audio amplifier
Fig: 4.12  Power supply unit.

Fig: 5.1  Main program flow chart.

Fig: 5.2  Keyboard read routine flowchart.

Fig: 6.1  Cassette recorder connection
CHAPTER ONE

INTRODUCTION

Teleconferencing is defined as group communication through electronic means\(^1\). It is a generic term for the whole range of electronic meeting aids. Teleconferencing need not be a 'meeting' in the same sense as face-to-face meeting but it attempts to provide as much as possible, the natural atmosphere that exists when people come together for the purpose of having discussion. With it, widely separated people can conduct a meeting by typing messages at their terminals\(^2\). Attendees may leave at will, and find out what they missed when they come back.

The term teleconferencing is used for a variety of telecommunications media ranging from\(^3\)

(a) voice only conference - where only audio signals are being transmitted.

(b) Voice with still pictures or voice with motion picture - where audio with moving images are transmitted simultaneously.

(c) Electronic mail which is oriented towards person-to-person communications and

(d) Computer conferencing which supports conferencing at which participants are not present simultaneously.

Teleconferencing has a spectrum of options with no clear indication that any of the options is the superior, though there are definite price trade-offs between the transmission bandwidths required as well as various equipment configurations\(^4\).
Therefore, a tendency exists to view audio teleconferencing as "bottom of the line". The obvious cost differences make this a logical observation. Also, teleconferencing being a developing technology, it is early to assume that any of today's products define the limit of how these systems will develop. In any case, audio teleconferencing is as good and sometimes better than motion video teleconferencing for specific applications. Some of such applications where audio teleconferencing is attractive are

(a) When there is no restriction on when and how long a user is entitled to use it.

(b) When the fund available to the user cannot support the video teleconferencing.

(c) Where the people involved in the conference are already familiar.

In the design reported here, an attempt is made to use a microprocessor to control the connections of individuals in order to have an audio conference. These individuals are physically separated but in an area of interest. The design is also expected to support more than one such conference going on at a time.

The microprocessor controls the connections and disconnects all the connections made to the system as well as rejecting all the connections that have been made. The use of a microprocessor for the control reduces the number of components required for the system design. Also, since the system can be programmed to meet individual users needs, the use of a microprocessor provides the best means of achieving the connections.
The design is broken down into modules for ease of implementation and testing and also for future fault isolation and component replacement. Some of these modules are located at the users premises while the others are to be located at a central place. The choice of components for the design was greatly influenced by availability.

Chapter 2 of this report gives a review of the existing teleconferencing systems. An attempt is also made in this chapter to analyse the impacts of teleconferencing on individuals and organisations as well as the reasons why it has not lived up to expectation. Chapter 3 analyses the various circuits used in the system design and development. In this chapter, the selection of the microprocessor is highlighted. The use of decoders and digitally controlled analogue switches are discussed as well as the programmable keyboard interface and programmable input/output (I/O) ports. In chapter 4, the system hardware design and analysis is presented while chapter 5 focuses on the system software design and development. Chapter 6 discusses the implementation of some of the modules simulation procedures and test results. The simulation was carried out in the laboratory using the MAT 385. Voice activated switch was demonstrated as well as mixing of audio signals. Chapter 7 concludes the work done and advances some recommendations for future expansion of the system.
CHAPTER TWO

LITERATURE REVIEW

2.1 ORIGIN OF TELECONFERENCING

Teleconferencing came into existence in the early 1970s. At that time, it was conceived as home information service that would use the television set as a display for the text received from a remote database, accessed over ordinary phone lines – videotex. This service later culminated into what is known today as computer teleconferencing, video teleconferencing, audio teleconferencing and electronic mail. These are more useful to the business world than homes. Some of the teleconferencing systems in use are briefly discussed below.

2.2 TYPES OF TELECONFERENCING SYSTEMS

2.2.1 VIDEO TELECONFERENCING

Full motion video teleconferencing is the most prominent multimedia system available today. This form of teleconferencing is group-to-group communication from specially constructed rooms. Such systems are typically not geared towards more individual uses or personalized computing capabilities. Furthermore, most video teleconferencing systems have been designed to simulate face-to-face meetings rather than to exploit the capabilities of electronic meetings.

The most widely used video teleconferencing is the one developed by Dr. Rohman and his colleagues which is a desktop model. This connects up to five sites on a broadband, full-duplex bridge that offers split-screen video and stereophonic audio.
2.2.2 **ELECTRONIC MAIL AND COMPUTER CONFERENCING**

The meaning of electronic mail and computer conferencing, most of the time are being misunderstood. Electronic mail is oriented towards person-to-person communication and is much more broadly available in today's marketplace. It is becoming so popular that it is now the first experience most people have with computer-based communications and like computer conferencing, it can introduce the powers of store-and-forward communications.

Computer conferencing, on the other hand, is designed to maintain an ongoing "transcript" of the interactions among many people discussing a topic. In this sense, it is a conference and the electronic file of entries is the proceedings. Computer conferencing systems, however, usually have awkward mechanisms for storing personal files. Electronic mail, then is a person-to-person medium while computer conferencing is group oriented.

Computer conferencing is mainly employed in the industry and universities for group problem solving and for generating taskforce reports. A major efficiency of computer conferencing is that it speeds the information dissemination and decision process especially when the participants' schedules do not permit them to set aside a common block of time for an in-person meeting or for an audio graphic teleconferencing.
2.2.3 Audio Teleconferencing

The greatest limitation to electronic mail is the computer terminal it requires. Audio teleconferencing eliminates this need, since it can be used through a standard touch-tone telephone.

In this area, a number of systems are already in use. Two most widely used are the ones developed by Dr. Robert Springer in the United States of America and Dr. Yamaguchi and his colleagues in Japan. The design by Robert Springer features the use of telephones lines on commercial audio bridges to support multi-point communications.

Audio/Graph mode also exists which provides enhancements to basic audio system which offers visual as well as audio interaction.

2.3 Components of Teleconferencing System

The components which make up a teleconferencing system include audio-visual terminal equipment, bridging equipment and telephone lines or other transmission media to interconnect these facilities as shown in figure 2.1.

Commonly used terminal equipment include telewriter systems which support interaction among remotely dispersed individuals for interactive writing, typing and drawing.

A variety of shared and dedicated telecommunication services that can be used to link together the remotely located participants of a teleconferencing system abound. The public switched telephone network,
FIG. 2-1 COMPONENTS OF TELECONFERENCING SYSTEM.
can be used to interconnect audio systems, telewriters and document scanners. Satellite transmission facilities can be used to interconnect point-to-point or point-to-multipoint compressed full motion video locations.

When audiographic terminal equipment is interconnected via telephone line, a teleconference bridge is the means by which multiple locations are tied together during a meeting. Some of the terminal equipment in use over standard telephone lines requires two telephone lines per site one for interconnecting the audio equipment and the other for interconnecting graphic devices.

Two basic types of teleconferencing bridges for terminal equipment that interwork with the telephone network exist namely a premises bridge and a network bridge. A premises bridge is located on the premises of a teleconferencing service vendor. A network bridge, on the other hand, is embedded in the public telephone network and usually provides for echo control and automatic volume level control. A bridge may have one or more of the following features:

(a) The "Meet-Me" feature which allows individuals and groups to dial into the bridge to participate in a scheduled conference. Some bridges answer automatically and add the participants to the conference. Others have the option to allow an operator to intercept the calling party and introduce him to the conference group.

(b) The "Add-On" feature which allows a conference participant or designated alternative to call up others to add each to the conference.

(c) The "Dial-Out" feature which requires a conference operator to dial out to bring participants into a conference.
(d) The "Blast-Up" feature which uses a computer to simultaneously dial-out the numbers of the conference participants and add those that answer to the conference.

A widely used add-on teleconferencing system is the AT and T Alliance Teleconference Service¹⁵. This is a public network service in the United States of America which does not require an advance reservation and allows one to add international locations that can be reached through direct distance dialing.

The design reported here utilizes the advantages of the add-on system. One advantage of the add-on Bridging system over others is that unwanted person cannot link his or her line to the bridge and listen to the conference. A conference participant uses the keyboard in his office to call the numbers of individuals he wishes to participate in the conference. However, the line of the called individuals are connected to the bridge irrespective of whether the called party answers or not.

2.4 IMPACTS OF TELECONFERENCING
2.4.1 POSITIVE IMPACTS

The positive impacts of teleconferencing on the society are many and some of them are stated here¹⁶.

(a) It reduces net travel expenses
(b) It reduces nonproductive time while travelling as well as travel fatigue.
(c) It reduces the number of mistakes made because the right person was not present at meeting
(d) It shortens time necessary for some business
cycles to be completed or for key decisions to be made.

(e) It provides faster and better responses to emergencies when key people are separated geographically.

(f) It provides the ability to share "people resources" when people needed are geographically separated.

(g) It makes job assignments less dependent on where a given employee happens to live.

2.4.2 NEGATIVE IMPACTS

Teleconferencing also has some negative impacts on organisations that make use of teleconferencing facilities. Some of these negative impacts are itemized below:

(a) It increases nonproductive time spent in meetings, because teleconference meetings are easier to arrange and unnecessary meetings may be arranged.

(b) It allows too much dependence on technology to develop, thereby creating a potential for breakdowns or even sabotage.

2.5 FAILURE OF THE EXISTING TELECONFERENCING SYSTEMS

Teleconferencing has not measured up to the market level originally predicted by its originators although several surveys carried out in the United States show not only interest but a great demand for teleconferencing. For instance, Bellcore researchers in the United States of America ran an experiment with a commercial audio bridge in the "meet-me" mode. Up to seven people can have an audio teleconference by
simply agreeing to call a single number to meet in audio space. Although the service was offered at no charge to over five hundred people, the service was used less than five times over a period of three years.

One reason for this low usage is due to the proliferation of inexpensive telephone systems. This seriously degraded the overall effectiveness of the electronic meetings. Another reason is the problem associated with teleconferencing acoustical engineering designs. The acceptable range in video quality is usually much broader than that of audio. Slightly fuzzy pictures of participants or less crisp letters in viewgraphs will not destroy a meeting but the inability to hear clearly what is being said discourages many people from making use of teleconferencing systems.

Also, the fact that each person has to connect himself to the audio bridge in some teleconferencing systems discourages people from discussing confidential matters over such facilities. This greatly contributed to the limited use made of some teleconferencing systems.

The design aimed at in this report makes use of the "add-on" bridging technique to eliminate this problem.
CHAPTER THREE

SYSTEM DEVELOPMENT

3.1 AIMS AND OBJECTIVES

At the beginning of the project work, certain aims and objectives were defined and are mentioned here.

To design:

(a) A system which can allow individuals in a particular area of interest to connect one another to have group discussion while sitting in their offices.

(b) A system that can connect about sixteen people for the purpose of group discussion.

(c) A system which can be expanded in future to accommodate more groups or people and

(d) to simulate and test the system using the Microprocessor Applications Trainer - MAT 385\textsuperscript{21}.

3.2 COMPONENTS SELECTION

3.2.1 General Considerations

In the design, a lot of factors were considered before components were chosen with a view to making the design cheap, reliable, efficient and work effectively. Compromise was therefore drawn between conflicting factors.

The major considerations in the choice of components were cost and availability. Where options were available, reliability and familiarity were considered.
The choice of circuits was based mainly on simplicity to enhance quick and easy implementation. In some cases, system performance and quality were given priority.

3.2.2 SELECTING THE MICROPROCESSOR
The selection of a microprocessor for any given application depends on quite a number of factors. In most of the applications, the following will need to be considered:

(a) Cost of the Microprocessor
(b) Reliability of the microprocessor
(c) Performance with regards to its processing power.
(d) Its ease of use
(e) The range of complementary hardware
(f) Special environmental constraints
(g) Availability of the microprocessor.

With these factors in mind, an appraisal was made of some of the existing 8-bit microprocessors in the market and the Intel 8085A CPU was eventually chosen for the design. Its pin configuration is shown in Appendix 2. The choice of this microprocessor over others like the z-80 and MC 6802 was greatly influenced by the fact that it is available in the laboratory. Also, its support hardware as well as sufficient documentation on it was readily available in the University laboratory.
The Intel 8085A is a single-chip, NMOS device implemented with approximately 6200 transistors, on a single chip mounted on a 40-pin dual-in-line package. The instruction set of the 8085A consists of seventy-four instructions. It has the advantage of internal clock circuitry over z-80 CPU and is capable of directly addressing up to 64k memory locations with its 16-bit address. Figure 3.1 shows the internal organisation of the Intel 8085A CPU while its instruction set is contained in Appendix I.

3.2.3 **KEYBOARD INTERFACE**

Keyboards containing a large number of push button switches or keys are commonly used when manual entry of data symbols is required for microprocessor applications. Each key is associated with a particular symbol or binary value. When a key is pressed, it generates a corresponding binary code. Key enclosures can be encoded into parallel data using combination circuitry when the number of keys is less than or equal to sixteen. This is known as linear arrangement of keys.

An alternative to this linear arrangement of keys is for the keys to be arranged at the intersection of wires that form a matrix network as shown in figure 3.2. This arrangement is more advantageous particularly when a large number of keys are involved because it allows a reduction in the amount of hardware required for the system design. The matrix is scanned continuously to determine that a key has been pressed,
FIG. 3.1 8085A ARCHITECTURE.
and to identify that key.

The scanning is done either by hardware or software designs by making all the rows of the matrix logic 0 and sensing the logic values of the columns. If one or more columns is logic 0, then one or more keys has been pressed. To encode the key, each horizontal wire is, in turn made logic 0 with all other horizontal wires logic 1, while a single horizontal wire is logic 0, each of the vertical wires is examined to see whether it is logic 0. When a vertical line is logic 0, the number of that line, together with the number of the logic 0 row, identifies the pressed key.

Figure 3.3 shows an example of the hardware used in key closure detection and key scanning with software. The rows of the key matrix are controlled by an output port and its columns are sensed through an input port.

To preclude a key being pressed and released without detection, software scan subroutine which is called up repetitively is used. It is also necessary to distinguish between a key being pressed several times or simply being pressed and held for a long period of time. There are two solutions to this problem - hardware or software debouncing. Software debouncing was chosen for this design because of the number of keys required so as to reduce component count. Hardware debouncing
FIG. 3.2 KEYBOARD SWITCH MATRIX.

FIG. 3.3 HARDWARE STRUCTURE FOR SOFTWARE SCANNING OF KEY MATRIX.
circuit is shown in figure 3.4.

Intel 8279 together with a decoder chip 74LS156 provides the keyboard interfacing unit capable of carrying out the matrixing function described above. The 8279 is a general purpose programmable keyboard and display input/output interface device designed for use with Intel microprocessors. The keyboard portion of it provides scanned interface to a 64-contact key matrix. Keyboard entries which is automatically debounced in a 8-bit character FIFO (first in first out) automatically sets the interrupt output line to the CPU.

3.2.4 MEMORY

Within any computer, both instructions and data are stored internally in the memory as binary numbers. This is because the computer can only understand and execute instructions coded in the binary code appropriate to the particular computer. Each instruction in the machine code instruction set of the computer is normally represented by one or more words in the computer memory. Each word is represented physically by a number of digits (bits) in parallel. Data is also stored as words within the computer memory and the word-length of the computer therefore defines the number of binary digits which the computer can manipulate simultaneously.
FIG. 3.4 HARDWARE DEBOUNCING CIRCUIT.
Memories are classified in a number of ways. Some utilize semiconductor devices and are called semiconductor memories while others utilize magnetic material. In the event of power failure, most semiconductor memories lose their stored information so they are called volatile memories. Magnetic memories retain their stored information even when power is turned off, so they are called non-volatile memories.

Another classification of memories centres on the way data can be written on them or extracted from them. In this group is the sequential access memory. Sequential access memories read or write data in sequence hence they are slow but inexpensive. The magnetic forms of sequential access memories can store very large amounts of data.

The fastest form of memory is the random access memory (RAM) where any memory location can be addressed without having to sequence through others. It is used for storing applications programs, fed in, for example, from a tape, disc or printer. They are also used to store intermediate results during program execution. Information stored in RAM can readily be modified. RAM memory is normally volatile.

Another form of memory is called the read only memory (ROM). This is different from RAM because data can only be read from it but cannot be written to. It is employed to store permanently required programs — the firmware which is usually "burnt"
into its store during manufacture or possibly by the user. ROM is non-volatile. The classification of memory technologies now available for microprocessor systems are summarized in figure 3.5.

Two Intel 8755 UV EPROMs each having 2k-byte memory capacity are used to provide a 4k-byte memory and one INTEL 8155 provides 256 bytes of random access memory. The 8755 chip incorporates two programmable input/output ports while the 8155 has three ports—one timer and two programmable input/output ports.

3.2.5 PROGRAMMABLE INPUT/OUTPUT PORT

An input port is that circuit that connects signals from external device to the microprocessor while an output port is that circuit that allows the microprocessor to output signals to external devices.

The functions of the input and the output ports are often combined into a single chip similar in size to the microprocessor chip package and is controlled by the microprocessor to input information when required from a data source and output data as instructed by the program being executed.

The Intel 8755 and 8155 contain programmable dual and triple port device respectively that provides a TTL compatible interface between peripheral devices and the 8085A CPU, and is used in the design.

3.2.6 BUFFERS

A buffer is an amplifier that can be used to increase the current drive capability of a microprocessor.
FIG. 3.5 MEMORY CLASSIFICATION
signal line. Buffers with tri-state outputs can be used to provide electrical isolation between portions of a microprocessor system. An important use of a buffer is in a microprocessor input port in which case it serves to isolate input data from the microprocessor data bus until input data is requested by the microprocessor.

If a device that drives the bus cannot meet the total current requirements of the devices to which it is connected, then its output must be buffered. The 8085A has a modest output current drive capability. The design makes use of 74LS245 octal buffer as the enhancing chip.

3.2.7 THE CLOCK

The clock signals in a microprocessor system are timing waveforms that are used to synchronize the system's operation. Some microprocessors do not have an internal oscillator to generate a clock signal and therefore require more complicated external circuitry to generate the waveform. Others like the 8085A have an internal oscillator to generate a clock signal; these microprocessors require only external passive timing elements.

The 8085A microprocessor also has provision for externally generated signal in which case the clock input is connected to the X₁ pin. Otherwise crystal oscillator is connected to the X₁ and X₂ pins. This later method is utilized in the design.
3.2.8 DECODERS

The decoder is a combinational circuit designed in such a way that at least one of the several outputs will respond to a unique input code. It is an MSI device available to the logic designers for use in a variety of applications. For each possible input code applied to the decoder, one of the outputs will differ (asserted) from all the rest because of this input condition. When the code is changed, another output will respond and the original output will return to its NOT ASSERTED condition

A decoder is a combinational circuit with \( n \) inputs and \( m \) outputs where \( m \leq 2^n \). Commericially available decoders are 2-to-4 decoders, 3-to-8 decoders and 4-to-16 decoders. Other powers of decoders could be obtained by connecting the available ones in matrix or stack form. Decoders are mainly used in computer systems for selecting memory address lines and I/O devices.

The design made use of six 74154 decoders (4-to-16 decoder) connected in stack form to select 80 lines. Latching device is connected to the output of the decoder so as to hold the asserted condition of any selected line.

3.2.9 THE MIXER

In order to obtain the natural response that exists when two or more people engaged in a discussion, an audio mixer circuit is necessary. This circuit enables the people to talk and be heard simultaneously.
A mixer therefore is a circuit which accepts more than one audio signal input and gives out single output with individual inputs still being distinct.

The mixer employed in the design used two μA741 (audio preamplifier) to boost the signal from the microphone preamplifier output so as to drive the power amplifier to give the required power output. The power amplifier uses two TDA 2002 IC chips.

The common way of mixing audio signals is shown in fig.3.6.

3.2.10 THE SWITCHING BANK

A switch is a device that passes or interrupts the current in an electric circuit. The most widely used switch in electrical connections is the relay. They are operated by an electromagnet. When an appropriate current flows through the coil, a magnetic force displaces the armature of the relay, in turn causing the switch contacts to open or close. The position of the switch contacts when the coil is not energized is known as the normal position. Thus, a relay may have both normally open (NO) and normally closed (NC) contacts.

The more useful form of switching device in computer applications is the multiplexer or selector. The basic function of this circuit is to select one of several inputs to connect to a single output line.
FIG. 3.6 MIXING AUDIO SIGNALS
3.2.11 POWER SUPPLY

Microprocessor applications require one or more power supply voltage(s). These voltages must be held within five percent of their nominal value. Microprocessor power supply design is greatly simplified by making use of three terminal integrated circuit voltage regulators such as the TTL 78XX series for positive voltages and 79XX series for negative voltage.

In designing with IC regulators, attention must be given to the power dissipation capabilities of the regulators. Such a regulator can typically dissipate one or two watts of power, depending on the adequacy of the heat sink to which it is attached. The power dissipation in watts in the regulator is given by the product of the load current and the difference between the input voltage and output voltages.

CPUs can be operated directly from batteries without the need for a voltage-regulated supply. The 8085A requires only a single +5V power supply. A voltage regulator IC - the 7805 is used in the design.
CHAPTER FOUR

SYSTEM DESIGN AND ANALYSIS

4.1 SYSTEM LAYOUT

The hardware design details of the microprocessor controlled audio teleconferencing system is shown in the system diagram, figure AT.1 inserted at the end of this report. The diagram is a complete working drawing of all the components adequately specified.

Top-down design approach was used. Highly interrelated parts of the system are in the same unit or module. These modules are namely

(a) Processing unit
(b) Keyboard encoding unit
(c) Decoding unit
(d) Switching unit
(e) Mixer unit
(f) Power supply unit.

This modular design approach simplified the design, fault isolation in case of system mal-functioning as peculiar faults could be attributed to a particular unit and also in testing of the system. It gives room for easy replacement of broken down units and components, and allows for future expansion of the system. Figure 4.1 shows the system block diagram of a microprocessor controlled audio teleconferencing system.
FIG. 4.1 SYSTEM BLOCK DIAGRAM
4.2 PROCESSING UNIT

This unit consists of the microprocessor, power-on reset circuitry and the crystal oscillator since the 8085A makes use of the internally generated clock signal. In this unit also is included the memory, the programmable input/output and the memory decoder chips.

$A_0^\text{D} - A_{15}^\text{D}$ form the 16-bit address bus, while $A_0^\text{AD} - A_7^\text{AD}$ form the multiplexed address/data bus. Some of these lines, $A_0^\text{AD} - A_{10}^\text{AD}$, are connected to the memory unit where they are used to achieve the required memory map. Address lines $A_{11}^\text{AD} - A_{15}^\text{AD}$ are connected to the data and enable lines of the 8205 chip for chip selection. Intel 8205 is a 3 - to - 8 line decoder compatible with the 8085 microprocessor system as shown in figure 4.2.

$A_0^\text{AD} - A_7^\text{AD}$ lines constitute the 8-bit data bus. These bus lines are extended to the respective pins of the memory unit. The address latch enable ALE pin of the microprocessor is connected to the corresponding ALE pins of the memory chips. This serves to distinguish when data is placed on the multiplexed data/address line and when the bus is carrying address. The IO/M, Clk, WR, RD pins of the microprocessor are connected to the corresponding pins of the memory chips. The reset out pin of the microprocessor is extended to the reset-in
of the memory chips while the outputs of the 8205 chip are connected to the chip enable of the memory chips.

The power-on reset circuitry is shown in figure 4.3 and is connected to the reset-in of the microprocessor. The crystal oscillator is connected to the X1 and X2 pins of the microprocessor. The crystal oscillator used for the system has a frequency of 5MHz. All other microprocessor pins which are not employed are left unconnected.

The memory mapping of the system is as follows:

0000 - 07FFH EPROM 1 2k byte
0800 - 0FFFH EPROM 2 2k byte
1000 - 10FFH RAM 256 bytes

When a memory location within the address range 0000 - 07FFH is addressed, pin 15 of the 8205 chip, the IO/M and the RD pins of the microprocessor will go low thus enabling data to be read from EPROM 1. Similarly, when a memory location within the range of 0800 - 0FFFH is addressed, pin 14 of the 8205 chip, the IO/M and the RD outputs of the microprocessor will go low, thereby enabling data to be read from EPROM 2. Memory locations 0000 - 00CE were used for the system software.

The memory chips (8155 and 8755) also contain the input/output ports. The logic value of the IO/M determines whether the address refers to memory or input/output. The address and value of IO/M are latched on the falling edge of the ALE. The ports can be configured to serve as an input or output port as shown in figure 4.4 (a).

The Intel 8155 and 8755 require a single 8-bit byte command information to initialize the ports. Each of the 8-bit byte is assigned some command significance as shown in
FIG. 4.3 POWER-ON RESET.
FIG. 4.4 PORT INITIALIZATION INFORMATION.
figure 4.4 (b). The command byte is transferred to the 8155 or 8755 using a specific address and the programmed output instruction, OUT. Thus the instruction

OUT address

transfers the contents of the processor A-register to the addressed input/output device. Table 4.1 shows information on programming the ports.

<table>
<thead>
<tr>
<th>CHIP</th>
<th>PORT ADDRESS</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8755</td>
<td>00</td>
<td>Command status register/</td>
</tr>
<tr>
<td>ROM</td>
<td>00</td>
<td>Port A (8bit I/O)</td>
</tr>
<tr>
<td>PIO1</td>
<td>01</td>
<td>Port B (8bit I/O)</td>
</tr>
<tr>
<td>8755</td>
<td>08</td>
<td>Command status register</td>
</tr>
<tr>
<td>ROM</td>
<td>09</td>
<td>Port A (8 bit I/O)</td>
</tr>
<tr>
<td>PIO2</td>
<td>0A</td>
<td>Port B (8 bit I/O)</td>
</tr>
<tr>
<td>8155</td>
<td>10</td>
<td>Command status register</td>
</tr>
<tr>
<td>ROM</td>
<td>11</td>
<td>Port A (8 bit I/O)</td>
</tr>
<tr>
<td>PIO3</td>
<td>12</td>
<td>Port B (8 bit I/O)</td>
</tr>
</tbody>
</table>

**TABLE 4.1 PORT PROGRAMMING INFORMATION**

In this design, Port 'A' is configured as an output port and Port 'B' as an input port. Port 'A' lines are connected to Port 'B' lines in such a way that the output is equal to the input.

4.3 **KEYBOARD ENCODING UNIT**

This unit consists of the matrix keyboard located at the users end and the programmable keyboard interface – Intel 8279 chip, the keyboard decoder – 74LS156 which is a dual 2-to-4 line decoder configured to work as a 3-to-8 line decoder. The line enhancing chips are located in the users end and the central location as shown in

---
The keypads are connected to the read lines of the programmable keyboard interface chip. This chip has three scan lines, SLO, SL1 and SL2. SLO is connected to the data line 'A' of the keyboard decoder chip. SL1 and SL2 are connected to data lines 'B' and 'C' respectively while the reset line is connected to the strobe input 'G' of the decoder.

The data lines of the 8279 chip are connected to the multiplexed address and data lines of the microprocessor and the memory chips. The interrupt pin is connected to the A55.5 pin of the microprocessor.

When the system is powered on, the 8279 chip starts scanning the keyboard. This is achieved by placing 000 binary digits on the scan lines. When this is done, the 2Y0 output of the decoder becomes asserted and the keys connected to this line are scanned. If any key is pressed, then the interfacing chip recognizes the key and generates the appropriate binary digits. In the event of no key being pressed, the scan lines generate 001 binary digits which then selects 2Y1 output, thus enabling the keys in that line to be scanned.

After two keys have been pressed, the binary equivalent of the two keys in 8-bit first-in first-out characters set the interrupt output line to the microprocessor for processing.

4.4 DECODING UNIT
The decoding unit makes use of six 74LS154 chips labelled IC 9 to IC 14 to select a total of 80 lines. The 74LS154 is a 4 line to 16 line decoder.
Figure 4.6 shows the decoder chips connected in stack form to achieve the required number of lines.

The four most significant bits of the eight lines from the output port of the I/O chip IC6 to IC8 are connected to the input line of IC9 while the four least significant lines are extended to the input lines of ICs 10, 11, 12, 13 and 14. The first five outputs of the IC 9 are then connected to the strobe lines of ICs 10, 11, 12, 13 and 14.

Each output line of the decoder is connected to a latching device so as to hold on the output immediately it is asserted. For example if 15H is pressed in the keyboard, the binary equivalent 00010101 is placed in the output lines of the output port (input to the decoders) and this causes line 21 to be selected.

The latching is provided by the Intel 8212 chips labelled IC 15 to IC 24 and connected as shown in the system diagram of fig. AT 1.

Pin 1 of the IC 10 is extended to the clear inputs of ICs 15 - 24. This serves to clear the system any time 00 is keyed-in in the keyboard.

The output of the latching chips are each connected to the control inputs of the switching unit i.e., pins 9, 10 and 11 of chips labelled IC 25 to IC 34 as shown in figure AT 1.

4.5 SWITCHING UNIT

The switching unit serves to make proper connections of the lines by linking up the appropriate speakers and microphones to the mixer. This is achieved by the use
of the MC 140538 chips labelled ICs 25 to 34 in the system
diagram of figure 4.1. The MC 140538 analogue multiplexer
is a digitally controlled analogue switch which implements
a triple single pole double throw operations.

Figure 4.7 shows the internal switching arrangement
of the MC 140538 chip. Pins 9, 10 and 11 are the control
input pins. These pins are connected to the output lines
of the decoding unit. Pins 1, 5 and 13 are not connected
and represents the unconnected position of the switches.
All other pins are connected as labelled in figure 4.7

4.6 MIXER UNIT

The mixer is simply a summing amplifier, the output
voltage being a vector sum of the input voltages. The
most common way of mixing audio signals is by the parallel
combination shown in figure 3.6.

The mixer unit in this work consists of three
sub-units. These are
* audio pre-amplifier
* the summing amplifier and
* audio amplifier.

4.6.1 AUDIO PRE-AMPLIFIER

The circuit diagram of the pre-amplifier is shown in
figure 4.8. It makes use of a two-stage transistor
amplifier using BC148 connected in the common-emitter
configuration. The choice of this configuration is
based on the fact that the input impedance of the microphone is of medium range value, and also to achieve high
stability.
FIG. 4.7 INTERNAL SWITCHING ARRANGEMENT
OF MC 14053B.
In amplifier design calculations, the knowledge of the gain, input impedance and the desired stability are always the starting point. In this design, an overall gain of not less than 250 is expected. The input impedance of the microphone is measured to be 1.8kilo-ohms which is a carbon type of microphone.

Calculations:
transistor parameters:
\[ h_{FE} = 100, \text{ germanium}. \]
It is required to produce quiescent collector current of 2mA and the load resistance as seen by the first stage is fixed at 4k
Then
\[ R_2 = R_B = \frac{V_{CE} - V_{BE}}{I_B} \quad \text{(1)} \]

Where
- \( R_B = \text{Self bias resistor} = R_2 \)
- \( V_{CE} = \text{Collector - Emitter Voltage} \)
- \( V_{BE} = \text{Base - Emitter Voltage} = 0.2V \)
- \( I_B = \text{Base current} \)

But
\[ I_B = \frac{I_C}{h_{FE}} \quad \text{(2)} \]
where \( I_C = \text{collector current} \)
Therefore
\[ I_B = \frac{2}{100} \times 10^{-3} = 0.02mA \]
\[ V_{CE} = V_{CC} - R_3 (I_C + I_B) \quad \text{(3)} \]
\[ V_{CC} = \text{Supply Voltage} \]
\[ V_{CE} = 12 - 4000 (2.02 \times 10^{-3}) \]
\[ V_{CE} = 3.92 \text{ volts} \]
From equation (1)
\[ q = 3.92 - 0.2 \]
\( R_4 \) is calculated by equating the voltage drops across 
\( R_3 \), the transistor and \( R_4 \) to the supply voltage given as in 
equation 4 below:

\[
V_{CE} = R_3 I_C + V_{CE} + R_4 (I_C + I_B) \quad (4)
\]

\[
12 = 4000 \times 2 \times 10^{-3} + 3.92 + (2.02 \times 10^{-3})R_4
\]

\( R_4 = 39.6 \text{ ohms} \)

chose a value of 47 ohms

The value of \( R_2 \) is chosen to be a little above the cal-
culated value so as not to drive the transistor to saturation.
Capacitor \( C_2 \) couples the signal source - the microphone to the
base of the transistor. \( R_5 \) and \( C_3 \) form a compensating network.
This serves to minimize oscillations and instability associated
with high frequencies. It also reduces the possibility of the
transistor picking feedback from echo effects especially from
long cables used in this work. \( C_3 \) serves also to alternate the
high frequency (carrier) signals that may be associated with
the local radio station.

To bias transistor \( TR_2 \) properly

\[
V_{BB} = \frac{R_7}{R_7 + R_6} \frac{V_{CC}}{12} \quad (5)
\]

\( V_{BB} \) is usually small and is taken to be 0.7 V

\( R_7 \) is fixed at 47k

From equation 5

\[
0.70 = \frac{47000}{47000 + R_6} \times 12
\]

\( R_6 = 705k \)

\( R_8 \) is fixed by allowing the current through it to be 100 times
that through \( R_6 \). This gives the value of \( R_8 \) to be 7.05k
FIG. 4.8 AUDIO PRE-AMPLIFIER.
Since a voltage gain of about 100 is required and

\[ A_V = \frac{R_C}{R_E} \]  \hspace{1cm} (6)

where

- \( R_C \) = Collector resistance
- \( R_E \) = Emitter resistance
- \( A_V \) = Voltage gain

From equation 6

\[ R_E = \frac{7050}{100} = 70 \text{ ohms} \]

choose a value of 100 ohms.

\( R_E \) has no bypass capacitor and represents a negative feedback, which reduces gain and increases stability.

The output of the second stage is fed to the UA741 IC (IC35) to further amplify the signal so as to drive the voice activated switch.

The feedback resistor \( RV2 \) is a variable one with a value of 1 Mohms since a high gain is expected. With a gain of 100 in mind, then \( R_{10} \) is taken to be \( \frac{1}{100} \) of the mid value of \( RV2 \). Therefore, a value of 5k ohms is chosen.

\[ R_{11} = R_{10} // RV2 \]  \hspace{1cm} (7)

Therefore \( R_{11} = 4.95k \text{ ohms} \)

Choose a value of 4.7k ohms.

Pin 7 of IC 35 is connected to the \( V_{CC} \). Pin 2 is the signal input while pin 3 is connected to ground through \( R_{11} \). The feedback resistor is connected between pin 6 which is the output and pin 2. \( RV3 \) is connected to pins 1 and 5 while the variable arm is connected to \( -V_{CC} \) and pin 4. This setup achieves null offset and is discussed briefly in section 4.6.1.1.
4.6.1.1 NULL OFFSETTING OF OPERATIONAL AMPLIFIERS

When a short circuit is applied between the inputs of an ideal op-amp, the output voltage is zero. In actual op-amps, however, imperfect matching of components within the op-amp lead to a non-zero output voltage when the inputs are shorted together.

Most op-amps have external terminals to which appropriate variable resistor is connected for the purpose of zeroing or nulling out this non-zero output. Fig 4.9 shows a typical op-amp biasing technique for null setting.\(^{35}\)

If it is assumed that equal bias current, \(i_B\), flows into both input leads, the voltage on the positive terminal is

\[
V_+ = -i_B R_B \quad (8)
\]

\(R_B = \) Bias resistance

Because of the negative feedback, it is assumed that

\[
V_+ = V_-
\]

Therefore

\[
I_s = \frac{V_s + i_B R_B}{R_B} \quad (9)
\]

Where

\(V_s = \) signal input

\(R_s = \) input resistance

Because \(i_B\) is flowing in the negative terminal, the current through \(R_F\) is

\[
I_F = I_s - i_B \quad (10)
\]

Writing Kirchhoff's voltage law for this

\[
V_0 = V_+ - i_F R_F \quad (11)
\]

\[
= \frac{R_F}{R_s} V_s + \left[ R_F - \frac{R_F}{R_s} \right] i_B \quad (12)
\]
FIG. 4.9  OP-AMP BIASING TECHNIQUE.
Closer examination of equation 12 shows that if

\[ R_B = R_s / / R_F = \frac{R_F R_s}{R_F + R_s} \]  \hspace{1cm} (13)

Where \( R_F \) = Feedback resistance

The bias current produces no output error.

### 4.6.2 VOICE ACTIVATED SWITCH

A voice activated switch is incorporated in the design. This serves to isolate the participants speaker when he is speaking in order to avoid the microphone picking up the output of the speaker as this may introduce echo.

The voice activated switch in this design makes use of the ST 151 thyristor and a diode as shown in figure 4.10. The output of the UA741 is fed through a small value resistor to the gate of the thyristor.

The diode D2 provides half wave rectification of the audio signal output to provide the firing pulses to the gate \(^{36,37}\). This rectification is necessary since the gate current should not be allowed to flow during the half cycle when the thyristor anode is negative which will lead to excessive leakage current.

The alternative way of achieving this is to have a fullwave rectifier circuit with the output connected to the gate of the thyristor. However, this method has a serious limitation since it requires external circuitry trying to turn off the thyristor once it is latched on.

A normally closed relay of 5kilo-ohms resistance is used so as not to exceed the holding current of the thyristor. A switch is included in the circuit to provide the participant the option of using the voice activated switch or not.
FIG. 4.10 VOICE ACTIVATED SWITCH.
4.6.3 Mixer Pre-Amplifier and Audio Amplifier

IC 36 of figure 4.11 was wired as an investing amplifier. RV4 and RV5 are used to provide bass and treble respectively. The output of the audio pre-amplifier is fed through capacitor C24 to IC 36. \( R_{16} \) and \( R_{17} \) provide the potential divider network for the treble and bass, and the paralleled combination of these provides the input impedance.

The gain of this amplifier is varied by RV4 and RV5. These controls vary the gain of the amplifier by adjusting the negative feedback.

\[ R_{16} \] and \( R_{17} \) are calculated from the formula

\[ R_{16} = \frac{(R_{17} \times Z_{in})}{(R_{17} - Z_{in})} \quad (14) \]

Where \( Z_{in} \) = the desired input impedance = 2k

\( R_{17} \) is set at 10k

From equation 14

\[ R_{16} = \frac{10000 \times 2000}{10000 - 2000} = 2.5k \]

\( R_{20} \) and \( R_{21} \) are the feedback resistors

Using equation 13 and expecting a gain of 10, \( R_{21} \) which formed part of the feedback is set at 15k while the RV5 formed the remaining part.

\[ R_{18} = \frac{R_{21} \times R_{17}}{R_{21} + R_{17}} \]

\[ R_{18} = 40k \]

The output is fed to the audio amplifier through capacitor C14. The audio amplifier makes use of two TDA 2002 chips, ICs 37 and 38. IC 37 is wired as an investing amplifier. The output of IC 37 is connected to pin 2 of IC 38 which is the inverting input so their
FIG. 4.11 MIXER PRE-AMPLIFIER & AUDIO AMPLIFIER.
outputs are in anti-phase. The voltage gain of the circuit was found to be approximately 150. RV6 varies the offset voltage of the circuit and is adjusted under no-signal condition. This was achieved by the input shorted to supply and ground with no load connected until the DC voltage between the output terminals became zero.

Transformer coupling is to be used to match multiple speakers to the output of the audio amplifier. However, the output of the audio amplifier was monitored in the oscilloscope. The biasing components for the TDA2002 assume the standard values available in the data book.\(^{32}\)

4.7 **POWER SUPPLY UNIT**

The power supply unit is shown in figure 4.12. A fast acting fuse \(F_1\) is included in the 240 Volts ac input to protect the transformer. \(T1\) is a 240V-15vac step-down transformer with centre tapping. The Secondary output voltage of \(T1\) is fed to a full-wave bridge rectifier. This is made up of diodes \(D_3-D_6\) which is the IN4001 series. This series is used because, at low voltages it drops about 0.6v of the voltage fed to it.

The rectified output is fed to a smoothing capacitor which removes the pulsating dc seen at the output of the bridge rectifier circuit. A capacitor of high value is chosen to give better ripple reduction. The voltage rating of the smoothing capacitors \(C_8\) and \(C_9\) demands that it be in excess of the voltage demand of the load. Therefore, a 2200uf, 20v capacitors are used in the design.

The series regulator technique is used to obtain the required dc volts\(^{38}\). This technique uses series resistors \(R_{14}\) and \(R_{15}\) and zener diodes \(D_7\) and \(D_8\) as the regulating
FIG. 4.12 POWER SUPPLY UNIT.

D3-D6 = IN4001
C12 = 33µF, 10V

240V

F1

R14

10

15Vw

15Vw

2200µF

20V

2200µF

20V

C8

C9

C11

C10

D7

D6

D5

D4

D3

7805

C13

R15

10
elements. This provides a good regulation of the output voltage as any increase in input Vdc is seen across the series resistors hence keeping the output voltage fairly constant at the zener voltage Vz. The design aimed for a maximum power output of 16 watts.

A voltage regulator chip, the 7805 is used to obtain a smooth regulated output voltage for the +5v. The +5v output from the power supply unit is fed to the processing unit, the switching unit, and the keyboard interfacing unit. The +12v supply serves the mixer action while the -12v supply is used in the null offset of the ICs.

Calculations:

Using a zener diode which has a rating of 12V 40 watts.

\[ I_z = \frac{P_z}{V_z} \]  \hspace{2cm} (15)

where

- \( I_z \) = Maximum zener current
- \( P_z \) = Maximum power rating of the zener diode
- \( V_z \) = Voltage rating of the zener diode

From equation 15

\[ I_z = \frac{40}{12} = 3.3 \text{ Amp.} \]

For a variable load current and variable input voltage

\[ R_{14} = \frac{V_{in} (\text{Min}) - V_z}{1.1 I_l(\text{max})} \]  \hspace{2cm} (16)

\( V_{in}(\text{min}) \) = the minimum input voltage

\( I_l(\text{max}) \) = maximum current drawn by load

Current requirement of the load = 3.0 Amp.

\[ V_{dc} = \frac{2V_m}{\sqrt{2}} \]  \hspace{2cm} (17)

\( V_m \) = peak-to-peak ac volts.

\[ V_{dc} = V_{ac} \sqrt{2} \]  \hspace{2cm} (18)

But \( V_{ac} = 15 \text{ volts} \)
Therefore, \( V_{dc} = 15\sqrt{2} = 21 \) volts.

Nominal value of \( R_{14} = \frac{V_{dc} - V_z}{3.0} \) \hspace{1cm} (19)

\( V_{dc} \) = input to the zener diode which is the output of the bridge rectifier.

\( R_{14} \) = compensating resistor

\( V_z \) = zener diode voltage rating.

\( R_{14} = \frac{21 - 12}{3} = 3 \) ohms

However, \( R_{14} \) and \( R_{15} \) are given a value of 10 ohms to ensure that the zener diodes D7 and D8 do not heat up excessively.
CHAPTER FIVE

5.0 SOFTWARE DESIGN AND DEVELOPMENT

5.1 DEVELOPMENT PROCEDURE

Software development is a process that culminates with the creation and verification of a pattern of 0s and 1s. This, when placed in the appropriate memory locations of a specific microprocessor system and executed, cause the system to implement its intended function. In general, software development consists of five steps:

1. **Design**: The determination from the functional specification of the overall structure of the program and the data, as well as a determination of the algorithms to implement the necessary functions.

2. **Coding**: The actual writing of instructions in a specific programming language to implement the algorithms of step one.

3. **Translation**: The creation of the patterns of 0s and 1s (the object code) from the program (source program) created in step two.

4. **Testing**: The determination of whether the object code, when executed, actually caused the system to implement its intended function.

5. **Debugging**: The process of determining the source of any failure found during testing in order to eliminate them. Correction of a failure, requires a repetition of these steps, starting with the first or second one.

Software is best written by a process of stepwise refinement which starts by expressing the program algorithm or flowchart in a very broad term, and then breaks each
block down into successively more detailed sub-blocks. This technique of software design is known as top-down strategy and is employed in this design.

5.2 PROGRAM DEVELOPMENT

The program is written in assembly language for Intel 8085A cpu and is made up of two main sections - the main routine and the keyboard read routine.

At power on or reset, the program starts from the memory address 0000. It starts first by initializing programmable input-output port PIO one. Port A is initialized as output while port B is initialized as an input port. Also, the command/status register is initialized as well as the stack pointer. The main program starts by setting the maximum number of people that are required to participate in a group discussion. This is achieved by the instruction

\[
\text{MVI C 0 F H}
\]

\[
\text{DCR C}
\]

The keyboards are polled by unmasking the interrupts and enabling them. If any data is found, it is stored in the location 10F5 which has been set aside as the input buffer address. Also, as the data is being sent to the output, a look-up table is created where these data are being stored for further reference. Group connection is terminated automatically when the number of connections made are up to sixteen and manually when FFH is pressed in the keyboard. The program also checks the calls already
made before connecting the second and third groups. At the end of any group discussion, the code combinations 00, 01 and 02 are used to disconnect all the engaged lines for groups 1, 2 and 3 respectively.

The flowchart for the main program is shown in figure 5.1 while the program is contained in Appendix 3.

The program for the keyboard read routine is designed in such a way that when a key is pressed, the value is stored in location 10FE. The value in this location is then moved to the accumulator when needed. The flowchart for the keyboard read routine is shown in figure 5.2.
START

INITIALIZE PIO one

STACK POINTER

CALL RDKBD
ROUTINE

IS DATA AVAILABLE

CONDITION DATA FOR OUTPUT

CALL RDKBD ROUTINE

A

B

C

Fig. 5.1: Main Program Flowchart.
IS DATA AVAILABLE?

YES
ADD THE CONTENTS OF B REGISTER

IS RESULT = FFH?

NO
OUTPUT RESULT.
STORE DATA IN D REGISTER PAIR

C

NO

IS COUNT UP TO 16?

YES
INITIALIZE PIO 2
A=OUTPUT, B=INPUT

K

D

MAIN PROGRAM FLOWCHART CONTINUES.
CALL RDKBD

IS DATA AVAILABLE?

YES
CONDITION DATA FOR OUTPUT.
STORE DATA IN B REGISTER.

CALL RDKBD

IS DATA AVAILABLE?

NO

YES
ADD THE CONTENTS OF B REGISTER

IS DATA = FF?

YES G

NO E
THE SAME WITH STORED DATA?

OUTPUT RESULT

STORE DATA

IS COUNT UP TO 16?

G

YES

INITIALIZE PIO 3
A=OUTPUT B=INPUT

CALL RDKBD

IS DATA AVAILABLE?

J

YES

CONDITION DATA FOR OUTPUT
STORE DATA IN B REGISTER

CALL RDKBD

IS DATA AVAILABLE?

YES

ADD THE CONTENTS OF B REGISTER

IS RESULT = FFH?

YES -> 1

NO

THE SAME WITH STORED DATA?

YES -> J

NO

OUTPUT RESULT

MAIN PROGRAM FLOWCHART CONTINUES.
FIG. 5.2 KEYBOARD READ FLOWCHART.
CHAPTER SIX

6.0 IMPLEMENTATION AND TESTING

6.1 IMPLEMENTATION

The initial assembly of the constituent components of the audio preamplifier unit, the voice activated switch, mixer preamplifier and audio amplifier were done on a breadboard. This was to ensure the working credibility of the design.

Although it was expected that the performance of the units whilst on the breadboard would not be exactly as that on a printed circuit board, a working performance was all that was expected at this stage.

As soon as these units were constructed on the breadboard, test procedures were initiated. These procedures were carried out with the aid of various equipment in the laboratory such as the MAT 385, the oscilloscope and the power supply module, PS 441. Although the final product was anticipated to operate using 12 volts dc, 5 volts dc and -12 volts dc, for test purposes +15v and -15v dc were used. This was obtained by the use of a feedback PS441 power supply unit available in the laboratory.

Two audio preamplifiers were constructed and these were used to test the ability of the mixer to work properly. Oscilloscope observations showed both the level of amplification achieved and mixing when an audio frequency signal was fed into the
system via the microphone.

6.1.1 PROGRAM TESTING

The assembly language program was written using
the Intel 8085A instructions and was tested on the
Microprocessor Applications Trainer - MAT 385
available in the Laboratory.

The program was loaded in the non-volatile read
only memory of the MAT 385 starting from location
8000. The MAT 385 has only one input and one output
port that can be used without external interfacing. As
a result of this, the program was modified so as to
show all the outputs in port 22 and to use port 21
as the input port.

The switches of port 21 are arbitrarily set and the
program was executed by issuing the command EXEC. -
GO - 8000 - EXEC. On issuing this command, the
letter E appears in the address field in the front
panel of the MAT 385 as indicated in figure 6.1.
Then two keys were pressed and the binary equivalent
of the keys in first-in first-out order was displayed
in port 22. This was repeated many times to
ascertain the credibility of the program.

After testing the program and having found it
working to specification, it was stored in a cassette
for future use. The procedure used in storing the
program and to retrieve the stored program for use
in the MAT 385 is described in section 6.1.2.
6.1.2 **CASSETTE INTERFACE OPERATION**

Figure 6.1 shows how the cassette recorder was connected to the MAT 385 so as to store the program in a cassette. The arrangement is also the same when the MAT 385 is being loaded from a cassette.

The Teletype/VOU switch in the front panel was set to the TELETYP. The interface cable was connected to the appropriate pins at the back of the MAT 385, and was extended to the mic and earphone socket of the tape recorder.

The number of blocks of data was entered into register 'A' by using the command EXAM REG A. Since the length of the program was not up to a block which is 256 bytes, the number 01 was entered into register 'A'. The address field displayed A and the data field displayed 01. The NEXT key was pressed seven times. At the seventh time, 'H' was displayed in the address field and 80 was keyed in. Pressing NEXT again displayed 'L' and 00 was entered. NEXT key was pressed and then the EXEC. key. At this point, section 'a' or 'b' below was followed depending on whether data was being loaded from the MAT 385 memory to cassette or from cassette recorder to MAT 385 memory.

(a) To load data from memory (MAT 385) to Tape, the following instructions were carried out.

(i) 'GO' key was pressed.
(ii) 1000 was entered in the address field.

(iii) The recorder was started and few seconds were allowed for the speed to stabilise.

(iv) Then 'EXEC. ' key was pressed.

Immediately this was down, the 'DATA' indicator lights showing that data was being transferred to the tape. After about one minute, the address field indicated 'EOP' signifying end of data transfer.

(b) To load data from Tape to Memory,

(i) The output level on tape recorder was set high enough (more than half,).

(ii) The 'GO' key was pressed

(iii) The program start address 1013 was entered.

(iv) The 'EXEC.' key was pressed

(v) The recorder was started (playback).

The 'DATA' indicator lights immediately to show flow of data. 'EOP' appears in the address field to signify end of data flow.

6.2 PROBLEMS ENCOUNTERED

Many problems were encountered during the course of the research reported in this write-up, with the availability of components being the most pronounced. This led to many disassembling and reconstruction, sometimes redesigning anytime a component becomes extremely difficult to come by in and around Owerri.
The problem of adequate access to relevant journals must be mentioned. Text books on related subjects were in extremely short supply as well as journals and magazines and so to acquire information from them proved difficult.

Initially, signal generator was used to feed signal to the audio amplifier. This gave distorted output because the minimum signal obtained from the signal generator is high enough to drive the transistors to saturation. This led to the belief that the design was not practical until a real microphone was used to pick normal human voice.

As regards the overall work, finance proved to be one of the major obstacles in the way of success, but ways of circumventing this problem were found.

6.3 SYSTEM OPERATION

The system starts operating once its power supply switch is set. The program starts at location 00000, thus the keyboard is continuously scanned for inputs.

By pressing the appropriate keys on the keyboard, a string of binary digits appear at the output port. These binary digits are fed to the decoding unit which selects one out of eighty lines. The selected output is latched to the switching unit which in turn connects the appropriate line corresponding to the keys pressed on the keyboard.

The system continues to connect up lines until a total of sixteen lines are connected or the end of
connection signal (FFH) is received. When any of these occurs the second group connections is automatically set and is ready to make connections. The procedure is repeated.

At the end of the group discussion, 00 is pressed in the keyboard by the participant that convened the meeting. This action clears or disconnects all the connections made on the group one. Similarly 01 and 10 disconnects all the connections made to groups two and three respectively. This is achieved by connecting the outputs of the decoders corresponding to these numbers to the clear inputs of the latches - the 8212 chips.
CHAPTER SEVEN

7.0 CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

Teleconferencing system is a growing technology that if well developed will enhance productivity both in industrial and academic environments.

A reliable, cheap and multipurpose teleconferencing system suitable for use by many different groups of people in an area of common interest, the object of this work reached a satisfactory stage, with the successful design of all the units, and implementation and testing of some of the units. The units implemented included the audio pre-amplifier, voice activated switch, mixer pre-amplifier and the audio amplifier.

The design of the micro-computer unit comprising the keyboard unit, the processing unit and the decoder was also completed but due to the problem of securing components and cost, was not implemented. However, the MAT 385 in the laboratory provided the microcomputer with which the program was tested.

The choice of the Intel 8085A CPU as opposed to other types of CPUs was based on availability rather than performance or reliability.

The attractive feature of the system so designed is that it is open ended. The modular design of the system gives room for future modifications and expansion. For example, the number of people that can participate in
a group discussion can be increased or decreased by just changing the content of the memory location containing the previously set number with the new one.

7.2 RECOMMENDATIONS

The design carried out in this report can be extended far beyond the design capacity. This is achieved easily because of its software base. The following recommendations will be useful for users of the system so designed.

(1) The system will be more cost-effective when many people are to be connected

(2) Input and Output ports are the only hardwares necessary for future expansion of the system.

(3) Complete restructure of the software to suit immediate user's need.
REFERENCES


3. Ibid p.6


24. Ibid, p.371


Appendix 1 The Intel 8085 Instruction Set

The information in the tables in this appendix are reproduced by courtesy of Intel Corporation. The following symbols and abbreviations are used.

A. B. C, D, E, H, L represent one of the internal processor registers
M represents the memory address currently held in register pair HL
byte represents an 8-bit (2 hex character) data quantity
dble represents a 16-bit (2 byte) data quantity
addr represents a 16-bit (2 byte) memory address
port represents an 8-bit I/O port address

Register pairs are denoted as follows:

PSW represents register pair AF
B represents register pair BC
D represents register pair DE
H represents register pair HL
SP represents the 16-bit stack pointer
PC represents the 16-bit program counter

The processor flag bits are

CY carry flag
Z zero flag
S sign flag
P parity flag
AC auxilliary carry
This group of instructions transfers data to and from registers and memory. No condition flags are affected by any instructions in this group.

<table>
<thead>
<tr>
<th>Move</th>
<th>Move (contd.)</th>
<th>Move Immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV-</td>
<td>MOV-</td>
<td>MOV-</td>
</tr>
<tr>
<td>A, A</td>
<td>7F</td>
<td>A, byte</td>
</tr>
<tr>
<td>A, B</td>
<td>78</td>
<td>B, byte</td>
</tr>
<tr>
<td>A, C</td>
<td>79</td>
<td>C, byte</td>
</tr>
<tr>
<td>A, D</td>
<td>7A</td>
<td>D, byte</td>
</tr>
<tr>
<td>A, E</td>
<td>7B</td>
<td>E, byte</td>
</tr>
<tr>
<td>A, H</td>
<td>7C</td>
<td>H, byte</td>
</tr>
<tr>
<td>A, L</td>
<td>7D</td>
<td>L, byte</td>
</tr>
<tr>
<td>A, M</td>
<td>7E</td>
<td>M, byte</td>
</tr>
<tr>
<td>MOV-</td>
<td>MOV-</td>
<td>MOV-</td>
</tr>
<tr>
<td>B, A</td>
<td>47</td>
<td>Load Immediate</td>
</tr>
<tr>
<td>B, B</td>
<td>40</td>
<td>(Reg. Pair)</td>
</tr>
<tr>
<td>B, C</td>
<td>41</td>
<td>B, dble</td>
</tr>
<tr>
<td>B, D</td>
<td>42</td>
<td>LXI</td>
</tr>
<tr>
<td>B, E</td>
<td>43</td>
<td>D, dble</td>
</tr>
<tr>
<td>B, H</td>
<td>44</td>
<td>H, dble</td>
</tr>
<tr>
<td>B, L</td>
<td>45</td>
<td>SP, dble</td>
</tr>
<tr>
<td>B, M</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>MOV-</td>
<td>MOV-</td>
<td>Load/Store A direct</td>
</tr>
<tr>
<td>C, A</td>
<td>4F</td>
<td>LDA addr</td>
</tr>
<tr>
<td>C, B</td>
<td>48</td>
<td>STA addr</td>
</tr>
<tr>
<td>C, C</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>MOV-</td>
<td>MOV-</td>
<td>Load/Store A indirect</td>
</tr>
<tr>
<td>C, D</td>
<td>4A</td>
<td>LDAX B</td>
</tr>
<tr>
<td>C, E</td>
<td>4B</td>
<td>LDAX D</td>
</tr>
<tr>
<td>C, H</td>
<td>4C</td>
<td>STAX B</td>
</tr>
<tr>
<td>C, L</td>
<td>4D</td>
<td>STAX D</td>
</tr>
<tr>
<td>C, M</td>
<td>4E</td>
<td></td>
</tr>
<tr>
<td>MOV-</td>
<td>MOV-</td>
<td>Load/Store HL direct</td>
</tr>
<tr>
<td>D, A</td>
<td>57</td>
<td>LHLD addr</td>
</tr>
<tr>
<td>D, B</td>
<td>50</td>
<td>SHLD addr</td>
</tr>
<tr>
<td>D, C</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>D, D</td>
<td>52</td>
<td>Exchange HL/DE</td>
</tr>
<tr>
<td>D, E</td>
<td>53</td>
<td>XCHG</td>
</tr>
<tr>
<td>D, H</td>
<td>54</td>
<td>EB</td>
</tr>
<tr>
<td>D, L</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>D, M</td>
<td>56</td>
<td></td>
</tr>
</tbody>
</table>
This group of instructions performs arithmetic operations on data in registers and memory.

<table>
<thead>
<tr>
<th>Add*</th>
<th>Add/Subtract Immediate*</th>
<th>Double Add†</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 87</td>
<td>ADI byte C6</td>
<td>B 09</td>
</tr>
<tr>
<td>B 80</td>
<td>ACI byte CE</td>
<td>D 19</td>
</tr>
<tr>
<td>C 81</td>
<td>SUI byte D6</td>
<td>H 29</td>
</tr>
<tr>
<td>D 82</td>
<td>SBI byte DE</td>
<td>SP 39</td>
</tr>
<tr>
<td>E 83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H 84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M 85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Increment/Decrement**

<table>
<thead>
<tr>
<th></th>
<th>A 3C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 8F</td>
<td></td>
</tr>
<tr>
<td>B 04</td>
<td></td>
</tr>
<tr>
<td>C 0C</td>
<td></td>
</tr>
<tr>
<td>D 14</td>
<td></td>
</tr>
<tr>
<td>E 1C</td>
<td></td>
</tr>
<tr>
<td>H 24</td>
<td></td>
</tr>
<tr>
<td>L 23</td>
<td></td>
</tr>
<tr>
<td>M 34</td>
<td></td>
</tr>
</tbody>
</table>

**Increment/Decrement Register Pair**

<table>
<thead>
<tr>
<th></th>
<th>A 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 8E</td>
<td></td>
</tr>
<tr>
<td>B 05</td>
<td></td>
</tr>
<tr>
<td>C 0D</td>
<td></td>
</tr>
<tr>
<td>D 15</td>
<td></td>
</tr>
<tr>
<td>E 1D</td>
<td></td>
</tr>
<tr>
<td>H 25</td>
<td></td>
</tr>
<tr>
<td>L 2D</td>
<td></td>
</tr>
<tr>
<td>M 35</td>
<td></td>
</tr>
</tbody>
</table>

**Subtract**

<table>
<thead>
<tr>
<th></th>
<th>B 03</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 97</td>
<td></td>
</tr>
<tr>
<td>B 13</td>
<td></td>
</tr>
<tr>
<td>H 23</td>
<td></td>
</tr>
<tr>
<td>M 33</td>
<td></td>
</tr>
</tbody>
</table>

**Decimal Adjust A**

<table>
<thead>
<tr>
<th></th>
<th>B 0B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 9F</td>
<td></td>
</tr>
<tr>
<td>B 2F</td>
<td></td>
</tr>
</tbody>
</table>

**Complement A**

<table>
<thead>
<tr>
<th></th>
<th>CMA 2F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 9B</td>
<td></td>
</tr>
<tr>
<td>B 99</td>
<td></td>
</tr>
</tbody>
</table>

Note

* All flags (CY, Z, S, P, AC) affected
** All flags except carry affected
† Only carry affected
‡‡ No flags affected
Data Manipulation Group - Logical

This group of instructions perform logical operations on data in registers and memory.

<table>
<thead>
<tr>
<th><strong>AND</strong>*</th>
<th><strong>OR</strong>*</th>
<th><strong>Exclusive-OR</strong>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A7</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A0</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>A1</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>A2</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>A3</td>
<td>E</td>
</tr>
<tr>
<td>H</td>
<td>A4</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>A5</td>
<td>L</td>
</tr>
<tr>
<td>M</td>
<td>A6</td>
<td>M</td>
</tr>
<tr>
<td>ANI</td>
<td>byte</td>
<td>ORI</td>
</tr>
<tr>
<td></td>
<td>E6</td>
<td>byte</td>
</tr>
<tr>
<td></td>
<td>F6</td>
<td>XRI</td>
</tr>
<tr>
<td></td>
<td>byte</td>
<td>byte</td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Compare</strong>*</th>
<th><strong>Rotate ️†</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>BF</td>
</tr>
<tr>
<td>B</td>
<td>B8</td>
</tr>
<tr>
<td>C</td>
<td>B9</td>
</tr>
<tr>
<td>D</td>
<td>BA</td>
</tr>
<tr>
<td>E</td>
<td>BB</td>
</tr>
<tr>
<td>H</td>
<td>BC</td>
</tr>
<tr>
<td>L</td>
<td>BD</td>
</tr>
<tr>
<td>M</td>
<td>BE</td>
</tr>
<tr>
<td>CPI</td>
<td>byte FE</td>
</tr>
<tr>
<td></td>
<td>RLC 07</td>
</tr>
<tr>
<td></td>
<td>RRC 0F</td>
</tr>
<tr>
<td></td>
<td>RAL 17</td>
</tr>
<tr>
<td></td>
<td>RAR 1F</td>
</tr>
</tbody>
</table>

Note

* All flags affected
† Only carry affected

Transfer of Control Group

This group of instructions alter normal sequential program flow. The following condition codes are used:

- **NZ**: not zero (Z = 0)  
- **Z**: zero (Z = 1)  
- **NC**: no carry (CY = 0)  
- **C**: carry (CY = 1)  
- **PO**: parity odd (P = 0)  
- **PE**: parity even (P = 1)  
- **P**: plus (S = 0)  
- **M**: minus (S = 1)
### Jump

<table>
<thead>
<tr>
<th>Jump</th>
<th>Call</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMP addr</td>
<td>C3</td>
<td>CALL addr</td>
</tr>
<tr>
<td>JNZ addr</td>
<td>C2</td>
<td>CNZ addr</td>
</tr>
<tr>
<td>JZ addr</td>
<td>CA</td>
<td>CZ addr</td>
</tr>
<tr>
<td>JNC addr</td>
<td>D2</td>
<td>CNC addr</td>
</tr>
<tr>
<td>JC addr</td>
<td>DA</td>
<td>CC addr</td>
</tr>
<tr>
<td>JPO addr</td>
<td>E2</td>
<td>CPO addr</td>
</tr>
<tr>
<td>JPE addr</td>
<td>EA</td>
<td>CPE addr</td>
</tr>
<tr>
<td>JP addr</td>
<td>F2</td>
<td>CP addr</td>
</tr>
<tr>
<td>JM addr</td>
<td>FA</td>
<td>CM addr</td>
</tr>
</tbody>
</table>

### Jump Indirect

**PHCL** E9

### Input/Output Group

This group of instructions perform I/O operations between the A-register and a specified port.

- IN port DB
- OUT port D3

### Machine Control Group

This group of instructions manipulate the contents of the stack and alters/controls the state of the processor.

#### Stack Operations (Register Pairs)

<table>
<thead>
<tr>
<th>PUSH</th>
<th>POP</th>
</tr>
</thead>
<tbody>
<tr>
<td>B C5</td>
<td>B C1</td>
</tr>
<tr>
<td>D D5</td>
<td>D D1</td>
</tr>
<tr>
<td>H E5</td>
<td>H E1</td>
</tr>
<tr>
<td>PSW F5</td>
<td>PSW F1</td>
</tr>
</tbody>
</table>

- **XTHL** E3 \((L) \leftrightarrow ((SP))\)
- **SPHL** F9 \((SP) \leftrightarrow (H) \leftrightarrow (L)\)

#### Interrupt Control

<table>
<thead>
<tr>
<th>Insert Code</th>
<th>Description</th>
<th>Restart Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI FB</td>
<td>(Enable Interrupts)</td>
<td>0 C7</td>
</tr>
<tr>
<td>DI F3</td>
<td>(Disable Interrupts)</td>
<td>1 CF</td>
</tr>
<tr>
<td>RIM 20</td>
<td>(Read Interrupt Mask)</td>
<td>2 D7</td>
</tr>
<tr>
<td>SIM 30</td>
<td>(Set Interrupt Mask)</td>
<td>3 DF</td>
</tr>
<tr>
<td>RST</td>
<td>(Reset)</td>
<td>4 E7</td>
</tr>
</tbody>
</table>

#### Processor Control Operations

<table>
<thead>
<tr>
<th>Insert Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP 00</td>
<td>(No Operation)</td>
</tr>
<tr>
<td>HLT 76</td>
<td>(Halt)</td>
</tr>
</tbody>
</table>
FIG. A-2 MICROPROCESSOR PIN OUT DIAGRAM.
APPENDIX 3

ASSEMBLY LANGUAGE PROGRAM FOR A MICROPROCESSOR-BASED TELECONFERENCE SYSTEM.

LANGUAGE USED IS THE INTEL 8085A ASSEMBLY LANGUAGE.

0000  P10 1 OUT EQU 00H
0001  P10 1 IN  EQU 01H
0009  P10 2 OUT EQU 09H
000A  P10 2 IN  EQU 0AH
0011  P10 3 OUT EQU 11H
0012  P10 3 IN  EQU 12H
0000  PIU 1 C/Y EQU 00H
0008  PIU 2 C/Y EQU 08H
0010  PIU 3 C/Y EQU 10H
10FE  KBDBUFF EQU 10FEH

0000  20  3E 01  MVI A, 01 :INITIALIZE PIU 1 PORTS
          :A=OUTPUT, B=INPUT.
0002  22  D3 00  OUT 00 :INITIALIZE CONTROL/STATUS REGISTER.
0004  23  31 00 10  LXI SP 1000 :INITIALIZE STACK POINTER.
0007  24  0E 0F  MVI C, OF :STATE MAX. NUMBER OF PARTICIPANTS.
0009  25  11 1F 10  LXI D, 101F :INITIALIZE THE START OF THE LOOK-UP
          :TABLE.
000C  27  3E 08  MVI A, 08 :USE 8085A SIM INSTRUCTION .
000E  28  30  SIM       :UNMASK INTERRUPT.
000F 29 FB E1 :Enable interrupt.
0010 30 CD BF 00 CALL RDKBD :Read keyboard value into register A.
0013 31 07 RLC :Condition data for output.
0014 32 07 RLC :
0015 33 07 RLC :
0016 34 07 RLC :
0017 35 47 MOV B, A :Store high order bits in B register.
0018 36 FB E1 :Ready for data from keyboard ?
0019 37 CD BF 00 CALL RDKBD :Read keyboard value into register A.
001C 39 80 ADD B :Update data for output.
001D 40 FE FF CPI FFH :Is it end of connections ?
001F 41 CA 2A 00 JZ 002A :Yes - begin next group connections.
0022 42 B3 00 OUT 00 :No - make connections.
0024 43 12 STAX D :Keep connection in D register pair.
0025 44 13 INX D :
0026 45 0D DCR C :Is maximum number of participants connected ?
0027 47 C2 0F 00 JNZ 000F :Yes - begin next group connections.
002A 48 3E 01 MVI 01H :Initialize PIO 2
002B 49 :A=Output, B=Input.
002C 50 D3 08 OUT 08 :Initialize command/status register.
002E 51 0E OF MVI C, OFH :Set maximum number of participants.
0030 52 DB 01 IN 01 :Read PIO 1 port B.
0032 53 FE 00 CPI 00 :Is PIO 1 port B engaged ?
0034 54 CA 00 00 JZ 0000 :No - make connections in port one.
0037 55 3E 08 MVI A, 08 :Use 8085A SIM instruction.
0039 56 30 SIM :Unmask interrupt.
003A 57 FD EI :ENABLE INTERRUPT.
003B 58 CD BF 00 CALL ROKBD :READ KEYBOARD VALUE INTO REGISTER A.
003E 59 07 RLC :CONDITION DATA FOR OUTPUT.
003F 60 07 RLC .
0040 61 07 RLC .
0041 62 07 RLC .
0042 63 47 MOV B, A :STORE HIGH ORDER BITS IN B REGISTER.
0043 64 FB EI :ENABLE INTERRUPT.
0044 65 CD BF 00 CALL ROKBD :READ KEYBOARD VALUE INTO REGISTER A.
0047 66 80 ADD, B :UPDATE DATA FOR OUTPUT.
0048 6E FF FF CPI FFH :IS IT END OF CONNECTIONS ?
004A 68 CA 66 00 JZ 0066 :YES - BEGIN NEXT GROUP CONNECTIONS.
004D 69 47 MOV B, A .
004E 70 36 OF MVI M, OFH .
0050 71 1A LXAX D :LOAD ACCUMULATOR WITH CONTENT OF D .
0051 73 B9 CMP B :PATH CONNECTION MADE BEFORE ?
0052 74 CA 3A 00 JZ 003A :YES - DISCARD IT.
0055 75 13 INC D :CHECK NEXT STORED NUMBER.
0056 76 35 DCR M ;
0057 77 C2 51 00 JNZ 0051 ;
005A 78 11 34 10 LXI D, 1034 :START NEW LOOK - UP TABLE.
005D 79 78 MUV A, B .
005E 80 D3 09 OUT 09 :MAKE CONNECTIONS TO FIG 2 A.
0060 81 12 STAX D :STORE NUMBER IN THE LOOK - UP TABLE.
0061 83 13 INX D ;
0062 84 0D DCR C :IS MAXIMUM NUMBER OF PARTICIPANTS
88.
85
0063 86 C2 3A 00 JNZ 003A :CONNECTED ?
0064 87 3E 01 MVI A, 01 :NO - MAKE MORE CONNECTIONS.
0065 88   :INITIALIZE PIO 3 PORTS
0066 89 D3 10 OUT 10 :A = OUTPUT  B = INPUT.
0067 90 0E 0F MVI C, OF :SET COMMAND/STATUS REGISTER
0068 91 DB 01 IN 01 :SET MAXIMUM NUMBER OF PARTICIPANTS.
0069 92 FE 00 CPI 00 :IS GROUP ONE FREE ?
0070 93 CA 00 00 JZ 0000 :YES - MAKE CONNECTIONS TO GROUP ONE.
0071 94 DB 0A IN 0A :IS GROUP TWO FREE ?
0072 95 FE 00 CPI 00 :
0073 96 CA 2A 00 JZ 002A :YES - MAKE CONNECTIONS TO GROUP TWO.
0074 97 3E 08 MVI A, 08 :USE SIM INSTRUCTION.
0075 98 30 SIM :UNMASK INTERRUPT.
0076 99 FB EI :ENABLE INTERRUPT.
0077 100 CD BF 00 CALL RDKBD :READ KEYBOARD VALUE INTO REGISTER A.
0078 101 07 RLC :CONDITION DATA FOR OUTPUT.
0079 102 07 RLC :
0080 103 07 RLC :
0081 104 07 RLC :
0082 105 47 MUV B, A :STORE HIGH ORDER BITS IN REGISTER B.
0083 106 FB EI :
0084 107 CD BF 00 CALL RDKBD :READ KEYBOARD VALUE INTO REGISTER A.
0085 108 80 ADD B :UPDATE DATA FOR OUTPUT.
0086 109 FE FF CPI FF :IS IT END OF CONNECTIONS ?
0087 110 CA A7 00 JZ 00A7 :YES - CHECK FOR FREE GROUP.
0088 111 47 MUV B, A :
0091 112  36  1F  MVI M, 1F  
0093 113  1A  LDAX D  :LOAD ACCUMULATOR WITH D REGISTER.  
0094 114  BB  CMP B  :IS CONNECTION MADE BEFORE?  
0095 115  CA  7D  00  JZ  007D  :YES - DISCARD IT.  
0098 116  13  INX D  :  
0099 117  35  DCR M  :  
009A 118  CA  A0  00  JZ  00A0  :  
009D 119  C3  94  00  JMP 0094  :NO - CHECK NEXT NUMBER IN THE LOOK-UP TABLE.  
120  
00A0 121  78  MUV A, B  :  
00A1 122  D3  11  OUT 11  :  
00A3 123  0D  DCR C  :IS MAXIMUM NUMBER OF PARTICIPANTS CONNECTED?  
124  
00A4 125  C2  7D  00  JNZ 007D  :NO - MAKE MORE CONNECTIONS.  
00A7 126  DB  01  IN 01  :IS GROUP ONE FREE?  
00A9 127  FE  00  CPI 00  :  
00AB 128  CA  00  00  JZ  0000  :YES - MAKE CONNECTIONS TO GROUP ONE.  
00AE 129  DB  0A  IN 0A  :NO - IS GROUP TWO FREE?  
00B0 130  FE  00  CPI 00  :  
00B2 131  CA  2A  00  JZ  002A  :YES - MAKE CONNECTIONS TO GROUP TWO.  
00B5 132  DB  12  IN 12  :NO - IS GROUP THREE FREE?  
00B7 133  FE  00  CPI 00  :  
00B9 134  CA  66  00  JZ  0066  :YES - MAKE CONNECTIONS TO GROUP THREE.  
00BC 135  C3  A7  00  JMP 00A7  :NO - CHECK ALL OVER FOR FREE GROUP.  
136  
137  READ KEYBOARD ROUTINE.  
00BF 138  21  FE  10  LXI H, 10FE  :GET INPUT BUFFER ADDRESS.  
00C2 139  7E  MUV A, M  :GET BUFFER CONTENTS.
```
<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Instruction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>00C3:0</td>
<td>140</td>
<td>B7 DRA A</td>
<td>IS CHARACTER AVAILABLE?</td>
</tr>
<tr>
<td>00C4:0</td>
<td>141</td>
<td>F2 CB 00 JP 00CB</td>
<td>YES - EXIT FROM LOOP.</td>
</tr>
<tr>
<td>00C7:0</td>
<td>142</td>
<td>FB E1</td>
<td>NO - READY FOR CHARACTER FROM KEYBOARD?</td>
</tr>
<tr>
<td>00CB:0</td>
<td>144</td>
<td>C3 BF 00 JMP 00DF</td>
<td>YES - GO READ KEYBOARD.</td>
</tr>
<tr>
<td>00CB:0</td>
<td>145</td>
<td>36 80 MVI M, 80</td>
<td>SET BUFFER EMPTY FLAG.</td>
</tr>
<tr>
<td>00CD:0</td>
<td>146</td>
<td>F3 D1</td>
<td>RETURN WITH INTERRUPTS DISABLED.</td>
</tr>
<tr>
<td>00CE:0</td>
<td>147</td>
<td>C9 RET.</td>
<td>END OF PROGRAM.</td>
</tr>
<tr>
<td>148</td>
<td>00D</td>
<td>ENDS.</td>
<td></td>
</tr>
</tbody>
</table>
```
## COMMON COMPONENTS TO ALL THE CHANNELS

<table>
<thead>
<tr>
<th>#</th>
<th>COMPONENTS</th>
<th>TYPE</th>
<th>QTY</th>
<th>UNIT VALUE (£)</th>
<th>QTY COST (£)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Microprocessor</td>
<td>8085</td>
<td>1</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>2</td>
<td>Memory (ROM &amp; I/O)</td>
<td>8755</td>
<td>1</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>Memory (RAM &amp; I/O)</td>
<td>6155</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>Keyboard Interface</td>
<td>8279</td>
<td>1</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>Memory Decoder</td>
<td>6205</td>
<td>1</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>Keyboard Decoder</td>
<td>74L3156</td>
<td>1</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>Buffer</td>
<td>74HC45</td>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Crystal</td>
<td>74001</td>
<td>4</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>Diodes</td>
<td>240-15V Al</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>Transformer</td>
<td>7805</td>
<td>1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Voltage Regulator</td>
<td>7505</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Diodes</td>
<td>15548</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Capacitors</td>
<td>Electrolytic</td>
<td>7</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>Resistors</td>
<td>Carbon</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Components per Channel

<table>
<thead>
<tr>
<th>#</th>
<th>COMPONENTS</th>
<th>TYPE</th>
<th>QTY</th>
<th>UNIT VALUE (£)</th>
<th>QTY COST (£)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>4-to-16 Decoder</td>
<td>74154</td>
<td>5</td>
<td>30</td>
<td>150</td>
</tr>
<tr>
<td>16</td>
<td>Latch</td>
<td>8212</td>
<td>10</td>
<td>25</td>
<td>250</td>
</tr>
<tr>
<td>17</td>
<td>Digital Switches</td>
<td>74LS38</td>
<td>50</td>
<td>15</td>
<td>750</td>
</tr>
<tr>
<td>18</td>
<td>Audio Amplifiers</td>
<td>UA 741</td>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>19</td>
<td>Audio Pre-Amplifier</td>
<td>TA 2002</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>Resistors</td>
<td>Carbon</td>
<td>12</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>21</td>
<td>Capacitors</td>
<td>Various</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

### Components for Station

<table>
<thead>
<tr>
<th>#</th>
<th>COMPONENTS</th>
<th>TYPE</th>
<th>QTY</th>
<th>UNIT VALUE (£)</th>
<th>QTY COST (£)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Microphones</td>
<td>Carbon</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>23</td>
<td>Transistor</td>
<td>BS 140</td>
<td>10</td>
<td>50</td>
<td>500</td>
</tr>
<tr>
<td>24</td>
<td>Op-Amp</td>
<td>UA 741</td>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>25</td>
<td>relay</td>
<td>7515</td>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>26</td>
<td>Thyristor</td>
<td>74L5245</td>
<td>2</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>27</td>
<td>Buffer</td>
<td>Carbon</td>
<td>14</td>
<td>5</td>
<td>70</td>
</tr>
<tr>
<td>28</td>
<td>Resistors</td>
<td>Various</td>
<td>6</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>29</td>
<td>Capacitors</td>
<td>IN4146</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>Diodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: £1752