THE DEVELOPMENT OF AN INEXPENSIVE, PORTABLE, VERSATILE, MICROPROCESSOR BASED DATA ACQUISITION SYSTEM

UNIVERSITY OF NAIROBI
CHIROMO LIBRARY

By

Aggrey Madahana

THIS THESIS HAS BEEN ACCEPTED FOR
THE DEGREE OF....MSc(1990)...
AND A COPY MAY BE PLACED IN THE
UNIVERSITY LIBRARY.

This thesis has been submitted in partial fulfillment for the Master of Science degree of the University of Nairobi.

JANUARY, 1990

DECLARATION

This thesis is my original work and has not been presented to any other University.

Hana

AGGREY MADAHANA
DEPARTMENT OF PHYSICS
UNIVERSITY OF NAIROBI

This thesis has been submitted to the University of Nairobi for examination with my approval as supervisor.

Prof. WILLIAM H. DRAKE
DEPARTMENT OF PHYSICS
UNIVERSITY OF MALAWI

To my fiancee Judy Wambui Njuguna

Oh, who can lift enquiring eyes,
And scan the star-bespangled skies,
Yet argue, earthbound as the cod
Mid wonders such, there is no God?

The flaming splendor of the noon,
The gentler beams of silver moon,
And far flung systems everywhere
Their woundorous Architect declare.

Yet though His glory they reveal,
Himself, His nature, they conceal;
We grope for Him whom they declare
Yet can but dimly find Him there.

Oh, love outshining starry gleam,
That He should suffer to redeem!
That He who all the heavens built
Once bled to bear my sin and guilt!

Now skies indeed are softer blue,
And every flower has lovelier hue;
The one whom stars proclaim above
As savior now I know and love.

J. Sidlow Baxter.

Acknowledgement

The knowledgeable man inevitably comes to the conclusion that the horizon of his knowledge is also the frontier of his ignorance. Frontiers pose new challenges as well as fears of pursuing a new experience. Our success in the new frontiers is partly dependent in one way or another on the contribution made to our work by various people whom we must always humbly acknowledge.

Special thanks to my supervisor Prof. William H. Drake for his invaluable contribution through out my research period on this work. His wide experience and far-sightedness kept me from deviating from the original proposal for this work. It was indeed his inspiration and inovativeness that led me to pursue this research.

I wish to also acknowledge the following people who have made some significant contribution to this work :- Gerald Kotonya and Silvester Namuye both of Egerton University in Njoro, Phil Bates who is currently working on a microcomputer project in the University of Nairobi, for his help on using the Icebox emulator. Mjomba Kale a tutorial fellow in the Physics Department for letting me use his Amstrad computer to design the PCB artwork.

Lastly, Iam grateful to Francis Juma a technician in the Physics Department University of Nairobi, and to all persons who have made a contribution to this work, but are not mentioned here by name.

Abstract

An inexpensive, portable, versatile, microprocessor based data acquisition system is presented. The design and development of the system hardware and software is described. The system is software intensive consisting of 2 Kbytes of monitor program and 1 Kbyte of application programs. It is configured to operate on a Z-80 microprocessor running at 2 MHz. It is programmed using a microcomputer through a serial port to meet specific requirements of data to be acquired which include calibrating specific input devices. A maximum of 16 analog signals in the range of 0-5 Volts can be simultaneously monitored by this system. Data is logged in the system in two modes. In the normal mode it acquires data at a maximum rate of 1 reading in 2 microseconds. Whereas in the interrupt mode with the real time clock it takes 1 reading in 10 milliseconds. The main board storage capacity is 5 Kbytes but it has an additional capacity of 40 Kbytes on the RAM expansion card. The operation of the equipment has been tested by acquiring data from a network of resistors with voltage variations simulated manually by varying a potentiometer, over a period of twenty four hours, with readings taken at 60 minutes intervals. A minimum current of 0.35 Amp. at 5V is sufficient to drive the system when using normal chips and an estimated minimum of 0.10 Amp. for their CMOS versions.

Contents

Chapter One: System Design

Chapter Two: The Integrated System

Chapter Three: Hardware Development

Chapter Four: Software Development

CHAPTER FIVE : SYSTEM EVALUATION AND TESTING

CHAPTER SIX: CONCLUSION AND FUTURE WORK

Introduction

In the field of scientific and engineering study, there are two main ways of obtaining information; analysis and experiment. Analysis is the use of agreed upon theories and mathematical formulae to predict and analyse physical situation. This method of obtaining information is the basis of theoretical physics. In experimentation we turn to the actual measurement of what happens and make conclusions based on the data gathered as well as our experience[1].

Scientific and technical advances in the past have provided a large quantity of theory and analysis. However inspite of this quantity, most general theory and analysis fail to provide sufficient detail of many phenomena of interest to us, or at least do not fully explain all phenomenon observed. For us to understand and capture detail, in different scientific phenomena, we must still have recourse to experiment.

The art of measurement which encompasses detection, acquisition, control and analysis is either done manually or automatically[2][3][4]. In performing an experiment manually, an observer (the experimenter) equipped with the appropriate measuring equipment observes an experiment and records the readings of his instruments as they respond to event changes from initial conditions. Detection in this first case depends to a large degree on the human senses sight, touch, smell, taste, sound, and mental perception. These senses serve as interfaces between the detecting devices or measuring equipment and the observer. Instrument readings (data acquired) and observations are recorded on paper and later translated into a form suitable for analysis such as tables, graphs, charts etc.

This method of data acquisition is limited to data collected from experiments whose time interval in detection is enough (60 seconds on the average) to accommodate the fastest human response to the observation and subsequent recording of the change. Furthermore the number of observations which can be monitored consecutively with a negligible time difference are also limited by the same factors mentioned above. Data acquired manually also pose tremendous challenges to storage, data compression, processing, and analysis. Manual acquisition of data therefore restricts the experimenter to a single event confined to a limited area or point in space. This restriction cannot be tolerated in experiments which require simultaneous acquisition of data, and those that require a well spatially distributed sample of readings from a chosen area. Such an area can only be observed well by an instrument capable of making several almost simultaneous observations.

Measurements taken manually are highly subjective, with errors resulting from human factors such as misreading of scales, reaction time, misrepresentation, and zero reading error. What's more, large amounts of experimental data acquired and recorded on paper cannot be translated easily into a format which would readily facilitate the use of a microcomputer for analysis, thus limiting the speed at which one would arrive at a conclusion.

This method of acquiring data is not adequate for most scientific experiments involving more than one reading or many physical parameters to be monitored simultaneously; and is very limiting in current research where measuring accuracy, data processing speed and storage are critical factors in the analysis of most scientific phenomena.

On the other hand, automatic data acquisition involve using electronic instruments to take measurements and acquire data. The human element is minimised, and he becomes a mere operator of the 'automatic observer'. When all measurements are fully automated, human errors are substantially reduced.

This work is primarily aimed at helping to partly resolve measurement problems encountered by a team of microclimatology researchers from the Departments of Geography and Botany in the University of Nairobi. Research in this area involves simultaneous measurements of temperature, soil moisture, light radiation, and humidity, usually done over a period of weeks at predetermined time intervals. A versatile equipment capable of acquiring data of all the physical parameters mentioned in real time, at pre-programmed intervals of time, would be the most suitable device for these kind of measurements.

The main purpose of this work was to develop a versatile data acquisition system based on a microprocessor to be used for measurements in most scientific applications. Even though the original requirements were those of the microclimatology research, Other requirements based on general observations from three undergraduate experiments in the Physics Department were considered. The three experiments involve:-

1. The performance of solar cells,
2. Thermal conductivity, and
3. The Coffee cooling problem in the simulation laboratory exercises.

It was not essential to examine any of these experiments in detail since the requirements have to be as general as possible for the development of a versatile general purpose equipment.

In this thesis chapter 1 discusses the design requirements for the system, system. Chapter two examines the resulting integrated system. Chapter three deals with the development of the hardware on the Nascom microcomputer and chapter four covers the software aspects of the design. Chapter five deals with the testing and performance of the system in real time. In the concluding chapter evaluation of the project is presented and recommendation made for future work where necessary.

Chapter 1

System design

1.1 Introduction

In general precision and accuracy become better as human errors are minimised in the process of taking measurements although the acquisition instrument and transducers do introduce their own intrinsic errors. An illustration of a general automatic data acquisition system is shown in fig. 1.1.1[5].

Data acquisition systems are used to measure and read signals obtained in basically two ways:-

1. Signals originating from direct measurements of electrical quantities; dc and ac voltages, frequency and resistance.

2. Signals originating from transducers, strain gauges, thermo-

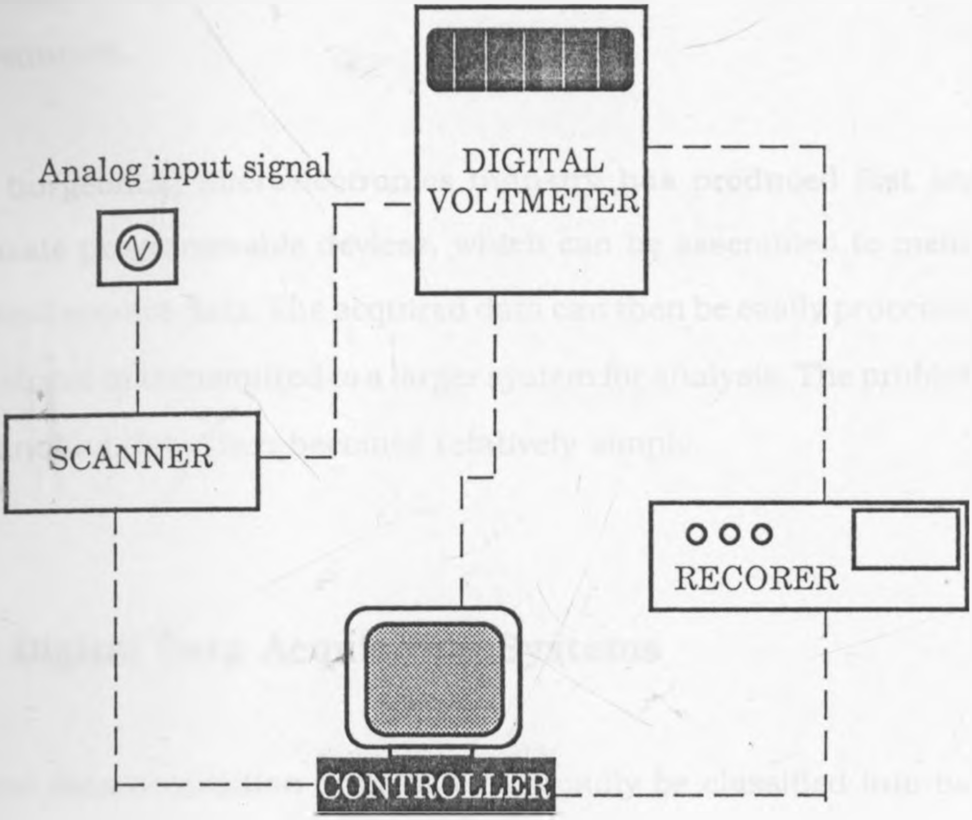


Fig. 1.1.1 A General Automatic Data Acquisition System.

couples, light sensors etc.

The equipment used may be either digital or analog. depending on the type of application. Accuracy and bandwidth are prime factors to be considered in the design of data acquisition equipment which range from single channel to multi-channel systems[6]. Single channel systems can monitor only one parameter at a time, while multi-channel data acquisition systems are equipped to make several readings in one cycle from either the same transducer or different transducers.

The burgeoning microelectronics industry has produced fast and accurate programmable devices, which can be assembled to measure and acquire data. The acquired data can then be easily processed and stored or transmitted to a larger system for analysis. The problem of handling data then becomes relatively simple.

1.2 Digital Data Acquisition Systems

Digital data acquisition systems can broadly be classified into two types. Those that require a host computer (usually a microcomputer) to acquire data and stand alone systems. The former are laboratory based systems designed to operate around an existing computer with a suitable interface circuit. The control and acquisition software on these systems are usually written in a high level language. These kind of systems are suitable for on-line control and processing of the acquired data, and results can be acquired in the form of graphical print-outs or charts. Some examples of systems that have been designed to operate from a host computer system include:- A paleomagnetic spinner magnetometer, which was interfaced to the KIM-1 microcomputer and developed by Thompson and Drake[7] in the

Physics Department, University of Nairobi. Later, another microcomputer based data acquisition and instrument control system for an absorption calorimeter, was developed in the same Department by Mate[8]. In Mate's design, the host computer used for control and acquisition was the BBC microcomputer, the software being written in BASIC. Two other pieces of work centred around the Apple-2 microcomputer namely; A microprocessor based data acquisition system for thermo-fluids laboratory[9] and a microcomputer interface board for time resolving multi-channel scaling respectively[10]. In all the four designs, suitable interface circuits are utilised to read and transmit data from transducers to the host microcomputer system which is invoked to issue control signals when necessary to operate the experimental devices set up. The microcomputers mentioned are also used for other functions besides acquiring data.

On the other hand, stand alone data acquisition systems are developed solely for the purpose of taking measurements, strictly, they are not general purpose microprocessor equipment, although some may be utilised for functions other than the acquisition of data, by altering the software and incorporating additional hardware on the system.

We shall now review, briefly, two data acquisition systems currently available on the market and used extensively in the field (outside the laboratory environment), especially by environmental biology and microclimatology research teams in the Departments of Geography, Crop Science and Botany of the University of Nairobi.

1.2.1 The Squirrel Data Acquisition System

The basic squirrel system used in field experiments is dedicated and programmed to measure one physical parameter. Thermistors have been hard wired to it and it is therefore configured to measure temperature only. However the versatile version of the squirrel is capable of measuring humidity, temperature, small ac and dc signals. It operates on an eight bit microprocessor. Data is logged by commanding the system to operate its logging program through a keypad. Under normal operating conditions the Squirrel's current consumption is about 20 milli-Amps. Data acquired by the squirrel can directly be down loaded into an IBM personal computer(pc) and used in a LOTUS 1-2-3 package for analysis[11].

1.2.2 The Campbell Versatile Programmable Data Logger

The campbell scientific inc. 21X micrologger is a versatile programmable data acquisition system based on the Hitachi 6303 CMOS central processing unit(CPU). The 21X system combines precision measurement with processing and control capabilities in one single battery operated system. Programs can be entered via a keypad which instructs the 21X to initiate measurement or control functions to process input data and store it. The system has 16K of ROM and 40K of RAM on board. It's workspace and user programming area utilises 11K of the RAM, so only 19K is available for low resolution data storage. Under normal logging conditions this logger consumes 60 milli-Amps of current for analog measurements and 10 milli-Amps in the quiescent condition. 21X has a throughput of 100 data values per second through its 16 channels of analog inputs, which can be expanded to a maximum of 192 channels. The acquired data is transmitted in ASCII format through a serial port into a larger system for analysis. When logging data, the system's real time clock is pre-programmed to invoke the CPU to run the data acquisition program at pre-determined intervals of time[12]. The two data acquisition instruments described above and others available locally are imported at quite exorbitant prices. oft-times they do not often meet the specification and requirements of local researchers fully. Further the unavailability of replacement components for maintenance is a major handicap to the continual operation of these equipment. There is a need to design equipment which meet local specifications and yet easy to handle and use from the electronic layman's point of view.

1.3 Project Requirements

The project design requirements based on the environmental biology and physics experiments have broadly been summarized into the following areas and are briefly discussed below.

1. Detection.
2. Transducer requirement.
3. Data storage.
4. Data transfer.
5. Equipment physical dimension.
6. Cost.
7. System flexibility.
8. Power consumption.
9. Timing device.

The conceptual design of the system is shown in fig 1.3.1 :-

The area under investigation in a microclimatology system requires simultaneous readings of data from selected points which are well distributed around the isolated area under consideration. The physics experiments cited in introduction chapter also require simultaneous reading of data. A set of readings for both investigations has to be taken simultaneously from several varying physical parameters. At least twelve input channels are required for the acquisition of data.

Data is to be acquired from transducers without the need of an operator switching the system from one channel to another. The equipment should be able to monitor the specified region over a period of at least 48 hrs, by automatically scanning all the input channels to acquire data from transducers.

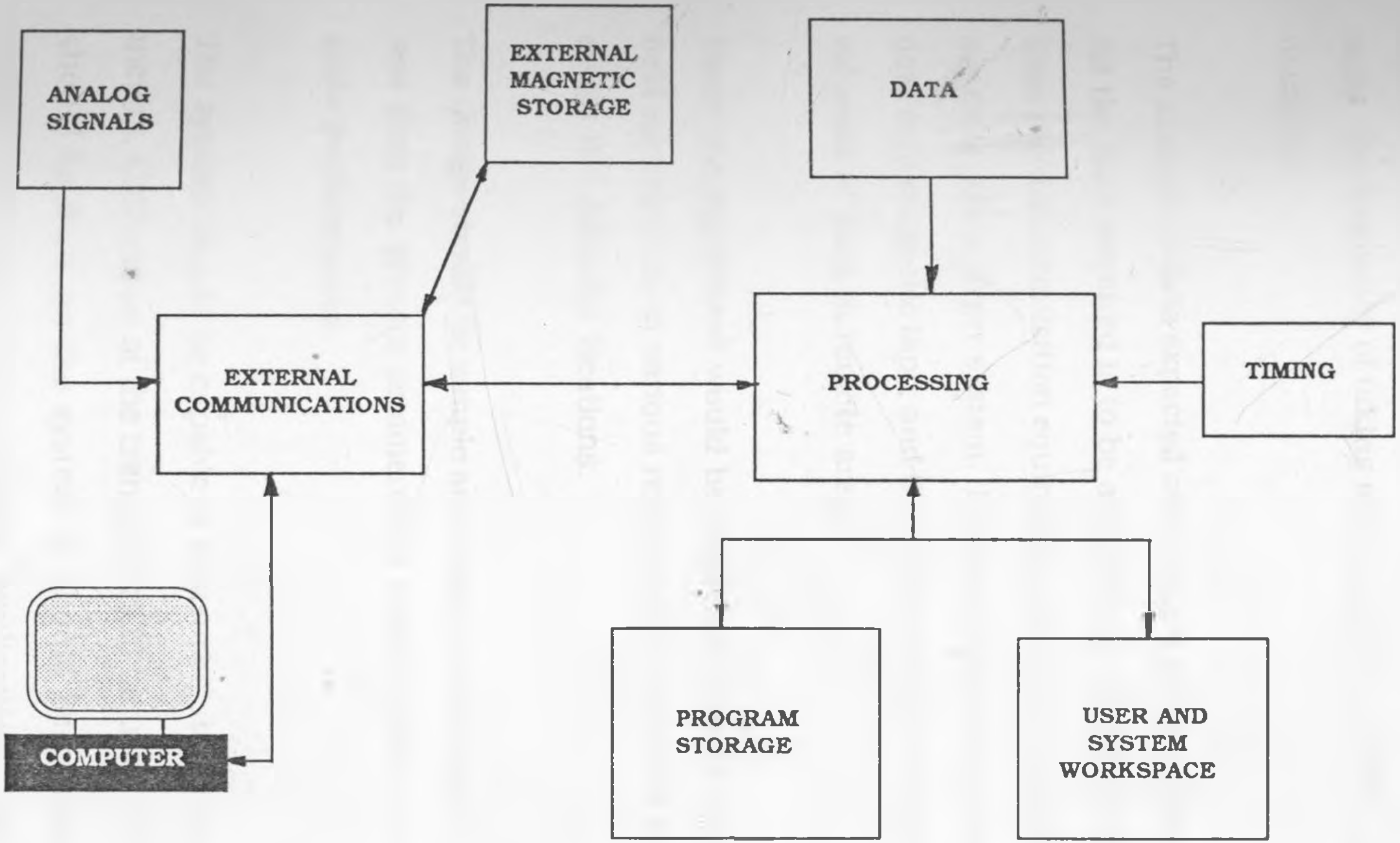


Fig. 1.3.1. Conceptual Block Diagram of the Data Acquisition System

The time interval of acquiring data would vary from one experiment to another. The programs should incorporate flexibility to predetermine the frequency of taking measurements and storing the data in memory.

The amount of data expected over a given period of time will be large. All the data acquired is to be analysed by a microcomputer system, thus the data acquisition equipment should be capable of downloading data into a larger system. It should also have a means of storing data on a magnetic tape, and thus facilitating the acquisition of large volumes of data in remote areas.

Since the equipment would be employed to take measurements in field experiments in various remote sites, it should be easy to move about the different locations.

The design should be simple and relatively affordable. It should cost less than the present commercially available systems with comparable performances.

The system should be capable of measuring different types of parameters. Calibration of the transducers to be used for measurement should be done on the system. It should be possible to alter the systems software to suit specific applications. The processes of operating the system should be simple and easy to learn with as little detailed training as possible on the use of the equipment.

The equipment should be usable in regions where there is no electricity. It should be designed to operate on low power for reasonable periods of time, at least 48 hours.

Most of the experiments will involve accurate time measurements. Time should be measured in two ways. First the time of the day would be required, mainly for field experiments, and secondly a clock that can be reset and started at any desired moment to measure relative time intervals would be utilised especially for laboratory experiments (like a stop watch). In which case the microprocessors system clock would not be sufficient for both purposes and an external real time clock is required.

1.4 The Design Philosophy

1.4.1 Design Criteria

The development of this data acquisition system was influenced by the following factors:-

1. The development systems and equipment available in the Physics Department and the University of Nairobi as a whole.
2. The requirements already discussed in the previous section.

The design criteria is summarized as follows:-

1. The system was developed in modules on the Nascom microcomputer system which is described later in chapter two. The initial hardware design utilised all the existing Nascoms hardware and peripherals, useful to in it's development, such as interface kits, disc drive and a printer. While other parts of the system were separately developed by interfacing them to the Nascom microcomputer system and the software written for each of the modules.

2. The integrated system was assembled from all modules as separately developed on the Nascom microcomputer system. The components used were all available locally at affordable costs.

3. The circuit was kept as simple as possible without trading off the capabilities of the resulting system. The resulting system is designed to be easy to maintain and operate. The design was optimised by reducing the chip count, with an aim of lowering the cost of the integrated system.

4. The circuit board was fabricated on a double sided PCB board to ensure close fitting of components and thus decrease the physical dimensions of the integrated system.

5. The instrument capabilities were tested by a simple method to read voltage variation from a set of 16 resistors of differing values. The voltage variations were simulated manually by a potentiometer.

1.4.2 Conceptual Design of the Data Acquisition System.

A block diagram of the design of the system is shown in figure 1.3.1 The main parts of the system are:-

1. Data storage:

The data storage comprises of two sections

i. Systems work space

ii. Temporary Data storage area

The systems workspace is the area needed to store temporary data used by the monitor program and is relevant to the operation of the system. The temporary data storage is required for the storage of data before it is transferred to a permanent storage such as the magnetic tape.

2. Program storage:

The programs were envisioned to be of two kinds: i. Monitor programs ii Application programs;

The monitor program is the main program required to manage the system, while the applications programs cover the following; applications Real time clock, acquisition, calibration, basic user routines, cassette tape interface and down loading programs.

3. processing unit:

i. Systems microprocessor and clock

ii Time measuring device. The system will have a microprocessor and time measuring device for the acquisition of data.

4. External world communications:

Four essential means of communicating with the outside world utilised in the system are outlined below.

i. Analog to digital converter.

ii. Parallel input/output devices.

iii. Serial input/output device.

iv. Visual and audio cues.

1.4.3 Choice of Components

The microprocessor utilised is the Z80 since it is readily available and it also has a wide range of industry standard input and output support chips. The other main devices will include a 7071 real time clock selected for it's ease in programming and operation in different modes ; namely as a time keeping device and interrupt generating device.

The parallel input/output ports are designed using the 8255 PIO chip, while external serial communications is via the 6204.
The AD8016 is preferred for the analog inputs since it allows more than twelve inputs of analog signals.

Address decoding is done using the 74ls138 and the 74ls139 decoder ICs.

The 4k memory 2716 ROMs are employed for the permanent storage of programs while the RAMs are implemented with several 4K and 8K 6116 and 6264 chips respectively.

1.5 Conclusion

We have discussed in this chapter the rationale behind the development and design of a versatile general purpose microprocessor based data acquisition system. The conceptual design of the system has also been presented.

Chapter 2

The Integrated System

2.1 General Introduction

The design and construction of the integrated system was based on concepts developed and discussed in chapter three and four. The individual modules developed on the Nascom system are integrated into one whole to operate as a stand alone system. The heart of this system is the Z80A microprocessor which replaces the Nascom computer in the conceptual design. The input/output operations and communications with the external world are accomplished through either the 8255 parallel ports for all operations which can be synchronised with the microprocessor's speed, or serially through the 6402 UART for asynchronous communications. Both the 8255 and 6402 chips replace the MIS kits employed in the development stage. The system is commanded or interrogated through the hexadecimal key pad provided. It can also be programmed by a host computer such as the BBC through the serial port.

The integrated system components were chosen to be compatible with the Z80 microprocessor, but not limited to the Z80 family devices, with a great deal of consideration being given to the cost and availability of the chips locally. A block diagram of the integrated system is shown in fig. 2.1.1.

2.2 Construction Details

Despite the complexity of the circuit, the construction was made relatively easy by using a double sided printed circuit board designed with the help of a computer aided design (CAD) package.

2.2.1 Circuit Layout Artwork Design

The artwork was designed and produced by the aid of "smARTWORK" PCB[28] layout CAD package. SmARTWORK is a PCB layout CAD package designed to produce and simplify the process involved in artwork design. Unlike the traditional artwork generated by using special tape on sheets of drafting film which is slow and requires considerable skill and experience to produce reasonable work. A CAD package such as smARTWORK enables both the novice and the expert in PCB design to produce professional artwork.

The following hardware was used in the PCB layout design process:-

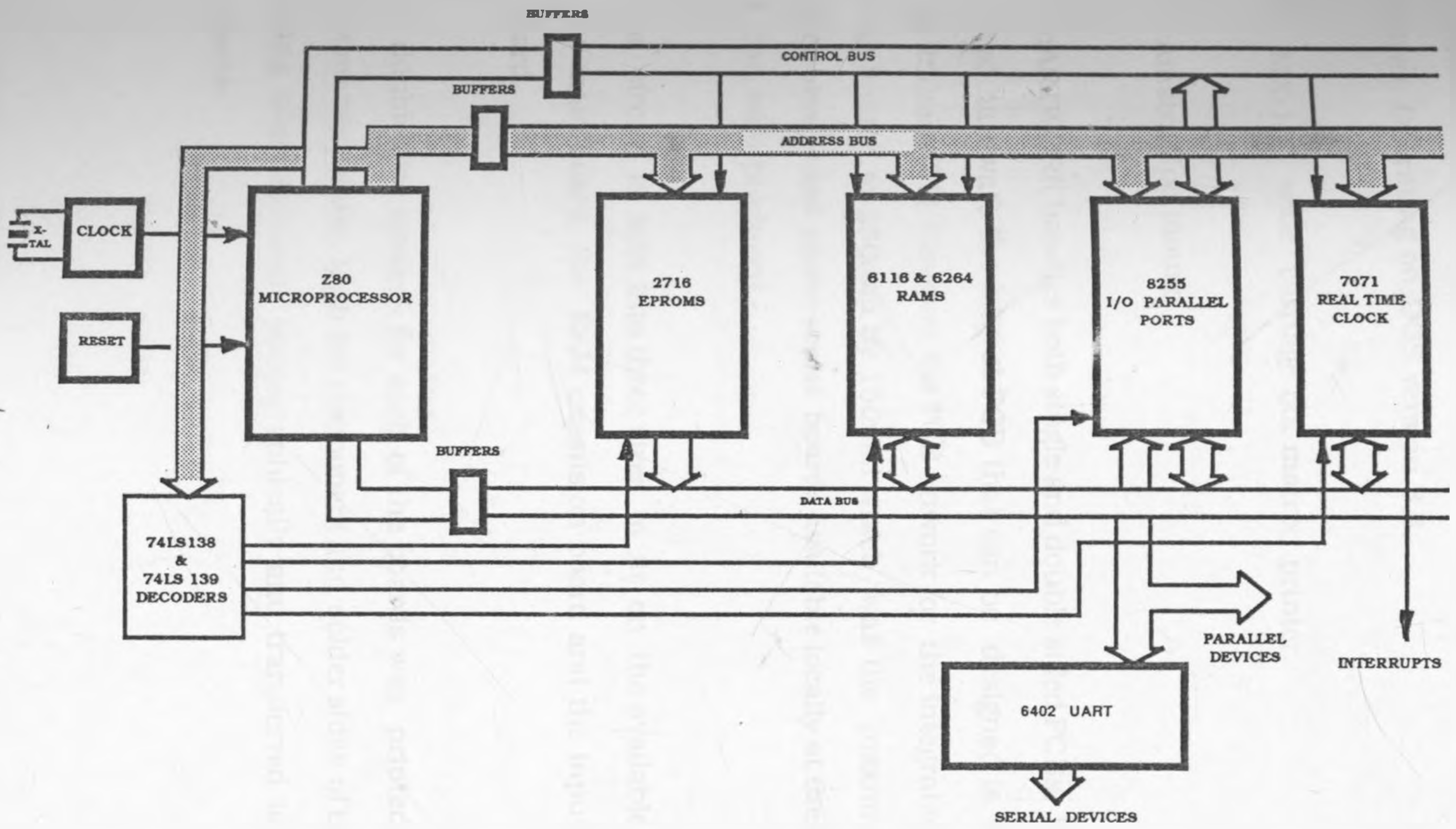


Fig. 2.1.1 The Integrated System Block Diagram

1. Amstrad pc 1640 microcomputer with 640K RAM, and two disk drives. Operating on DOS version 3.3.

2. MX-100 wide carriage dot matrix printer.

3. Amstrad pc mouse.

SmARTWORK handles both single and double sided PCBs. With this listed hardware the largest PCB that can be designed is 400mm. by 250mm [28]. However the PCB artwork for the integrated system was limited to 250mm by 150mm which was the maximum size of double sided photo-resist boards available locally at the time the PCBs were produced.

The circuit is split into three sizes to fit on the available boards. The main board, the RAM expansion board and the input/output board.

A double size artwork for each of the boards was printed on the dot matrix printer. Both the component and solder sides of the board being later reduced photographically and transferred to acetate sheets.

2.2.2 Fabrication

The fabrication commenced by exposing both sides of the photo-resist board sandwiched between two sheets of acetate to UV light, for approximately 3 minutes. The exposed board was then developed, etched, striped and tinned to protect the copper tracks from corroding. This process took about an hour to complete production. Finally 1mm. and 0.8mm. holes were drilled on the board and components placed and soldered in position. Appendix D1 to D3 contain diagrams for the layout of components on the PCBs. Appendix D4 to D7 contain copies of the printed circuit board diagrams and a sample copy of a negative obtained from photographically reducing the double sized artwork of the PCB layout transferred to the acetate sheets.

2.2.3 Circuit Check

On completion, the boards were first inspected for any short circuits using a multimeter and any breaks in the circuit tracks were monitored physically. The performance of the main board was then tested using the Icebox emulator which confirmed that the decoding had been correctly designed and that all the mapped memories and input/output ports were not conflicting.

The RAM and I/O boards were then connected to the main board with 50 way ribbon cables plugged into the edge connectors and similarly their performance tested.

2.3 Physical Layout

The system has three main boards:-

1. The main board, consisting of the microprocessor Z80 with its clock circuit, the real time clock circuitry; centred around the ICM 7071 real time clock chip, 2K ROMs - 2716 for the monitor program, and 2K 6116 RAMS together with the associated decoding circuitry. The physical layout of the main board is shown in appendix D4.

2. The input/output board, having the analog to digital converter circuit implemented by the the ADC 0817 chip, the 8255 parallel ports, and the 6402 UART. Appendix D5 shows the physical layout of the input/output board.

3. The RAM expansion board. The physical layout of the RAM expansion board is shown in appendix D6. It has a capacity of 50 Kbytes memory space. The main components are five 2k 6116 RAMs and five 8k 6264 RAMs.

2.4 The Z80 Microprocessor

It has been mentioned that the heart of this data acquisition system is the Z80 microprocessor. A pin diagram of the Z80 is shown in figure 2.4.1. The address bus is represented by signals A15, which is the most significant bit, through A0 the least significant bit.

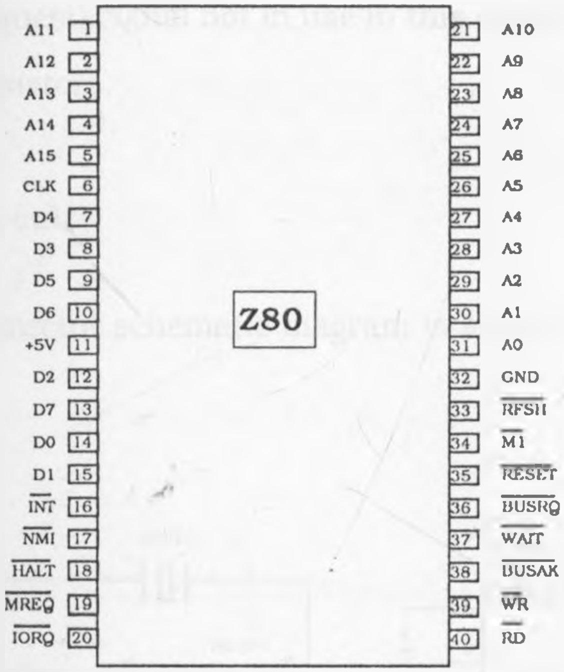


Fig. 2.4.1 The Z80 Pin diagram

All the address lines are tri-state active high signals and are configured to address a total of 64Kbytes of memory. It can also address an additional 256 input/output devices. The input/output device address is held on only eight of these lines ie. A7 through A0.

The data bus signals are also tri-state active high, ranging from

D7 the most significant bit, through D0 the least significant data bit. Whereas the address bus is uni-directional, the data bus is bi-directional permitting data to be transferred either to the CPU from external devices or memory and vice-versa. The WAIT signal, and the BUSRQ (bus request) signal not in use in this design are held high through 10K resistors.

2.4.1 Clock Circuit

The Z80 clock circuit schematic diagram is shown in fig 2.4.1.1.

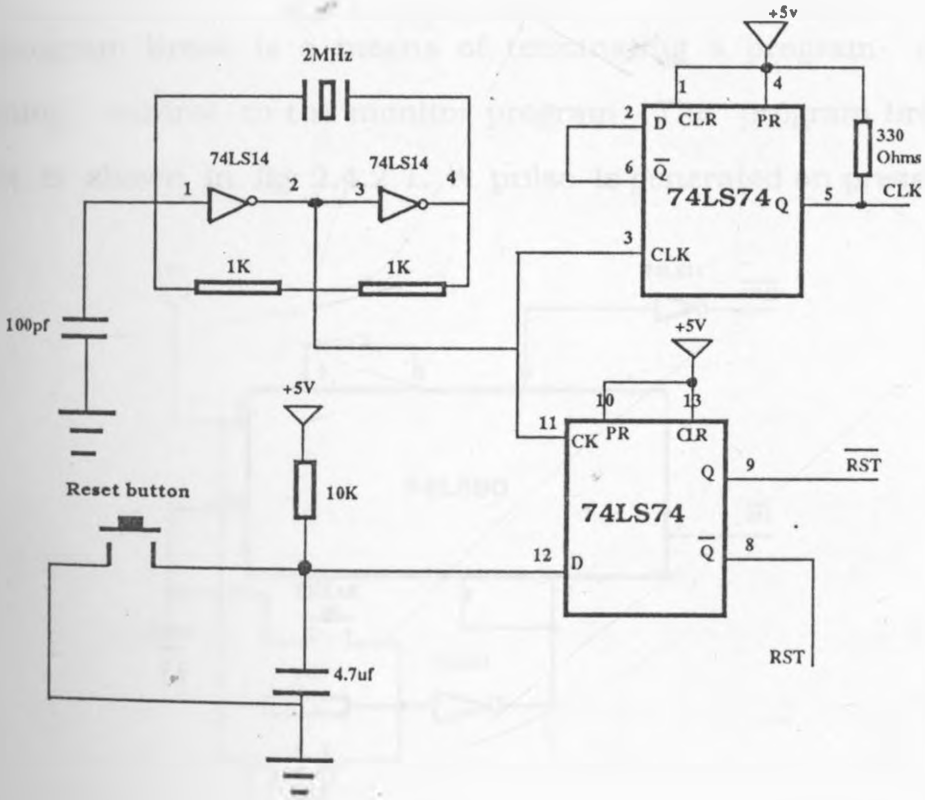


Fig. 2.4.1.1 System Clock and Reset Circuit

The initial signal is generated by a 2 Mhz crystal oscillator. A schmitt trigger circuit is used to generate square wave pulses from the crystal pulses that are later divided by two with the 74LS74 to give a 1 Mhz signal. An external clock pull up resistor of 330 Ohms used. adequately meets all the external ac and dc clock requirements. A 1 Mhz speed is chosen because speed is not so much a critical factor in this work and it also suits the clock requirements for the analog to digital converter circuit, without the use of additional dividing hardware.

2.4.2 Program Break and Restart Signals

The program break is a means of terminating a program and returning control to the monitor program. The program break circuit is shown in fig 2.4.2.1. A pulse is generated on pressing

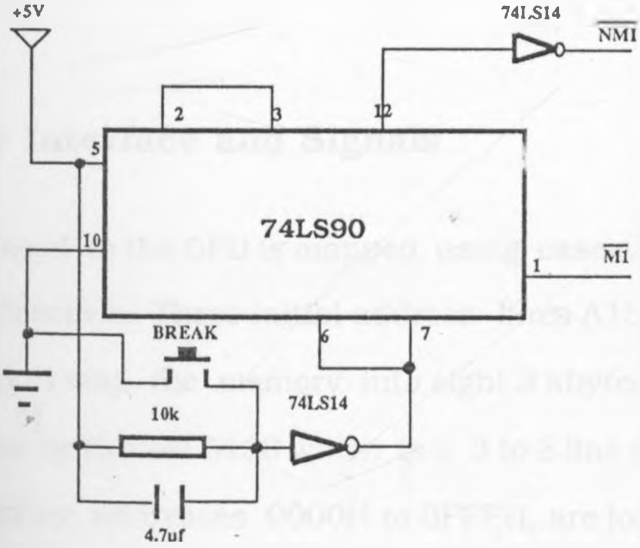


Fig. 2.4.2.1 Program Break Circuit

the monitor key, to reset the 74LS90 decade counter to 9 and gate in the M1 instruction fetch signal. The decade counter then issues a low signal input to the non maskable interrupt input of the microprocessor, which then terminates the execution of the program, and returns control to the monitor program. In this case the system is not initialized again.

The system is restarted by using the reset key provided. The D flip-flop 74LS74 is used to trim the reset signal produced on pressing the reset key. The RST signal is sent to the CPU. The inverse of this signal RST is sent to the 8255 chip. Unlike the program break key, the whole system is initialized on resetting. In this case all interrupts are disabled except the NMI signal. The IV and R registers are both set to 0000, and the system begins execution of the program at ROM address location 0000H again. The restart circuit is as shown in figure 2.4.1.1

2.5 Memory Interface and Signals

Memory interfaced to the CPU is mapped using cascaded 74LS138 and 74LS139 decoders. Three initial address lines A15, A14, and A13 are utilized to map the memory into eight 8 kbytes areas. This decoding is done by the 74LS138 which is a 3 to 8 line decoder. The first 8K of memory, addresses 0000H to 3FFFH, are located on the main board. The remaining addresses 4000H to FFFFH are channelled to the bus for additional data acquisition storage. The RAM expansion card then utilizes some of this memory.

Address 0000H to 3FFFH (on the main board) are further decoded into four 2K areas using two cascaded 74LS139, 2 to 4 line decoders. The memory map of this address section is shown in figure 2.5.1.

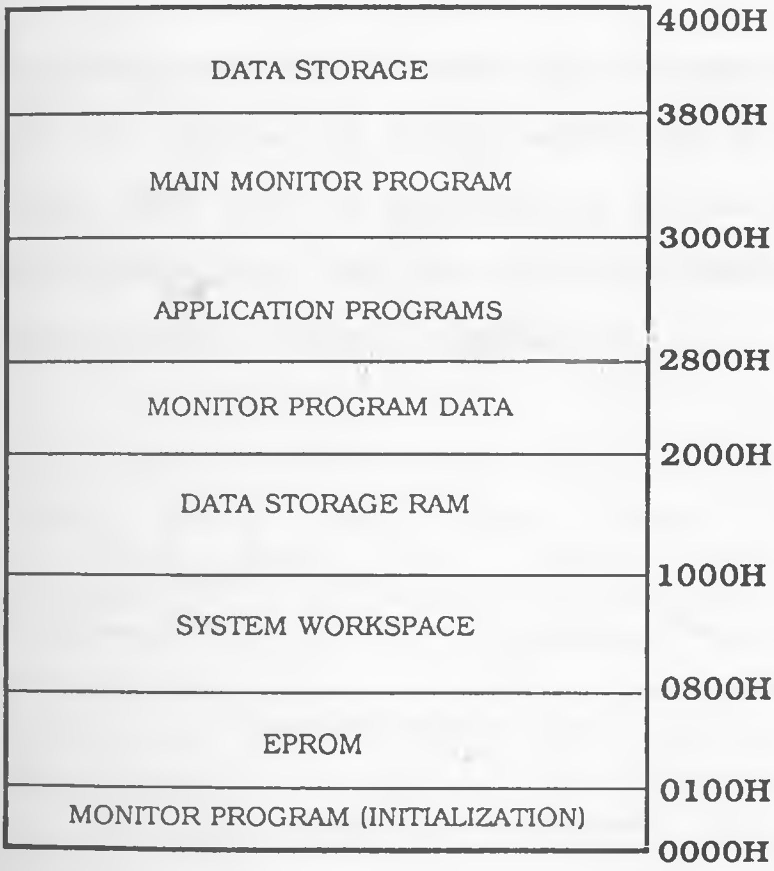


Fig 2.4.1 System Memory Map for the Main Board

The memory map was determined by the program development on the Nascom computer. The main program ROM is assigned location

0000H to 3FFFH because the program was developed from location 3000H on the Nascom microcomputer. Program execution for the Z80 microprocessor commences at location 0000H, therefore part of the initialization program begins at location 0000H. Control, is then later transferred to the monitor program at location 3000H after initialization. Each ROM and RAM chip is accessed when the right address appears on the address lines to enable the appropriate chip.

The ROMs are accessed when both the enable signal (EN) generated by the decoders with signals from address signals A15 to A11, and the read signal (RD) from the microprocessor are both low, while the EPROM program pin is held high all this time. The timing diagram for this operation is shown in figure 2.5.2.

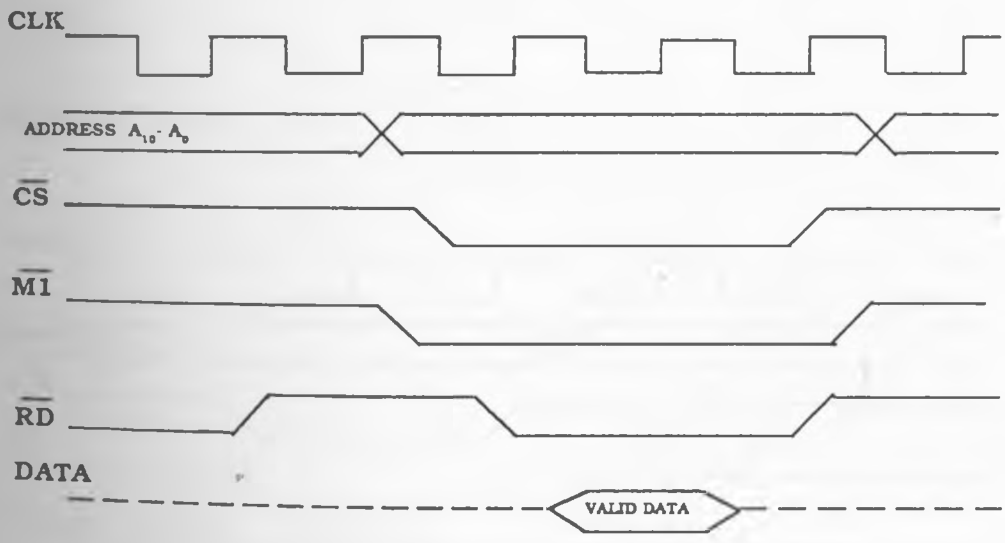
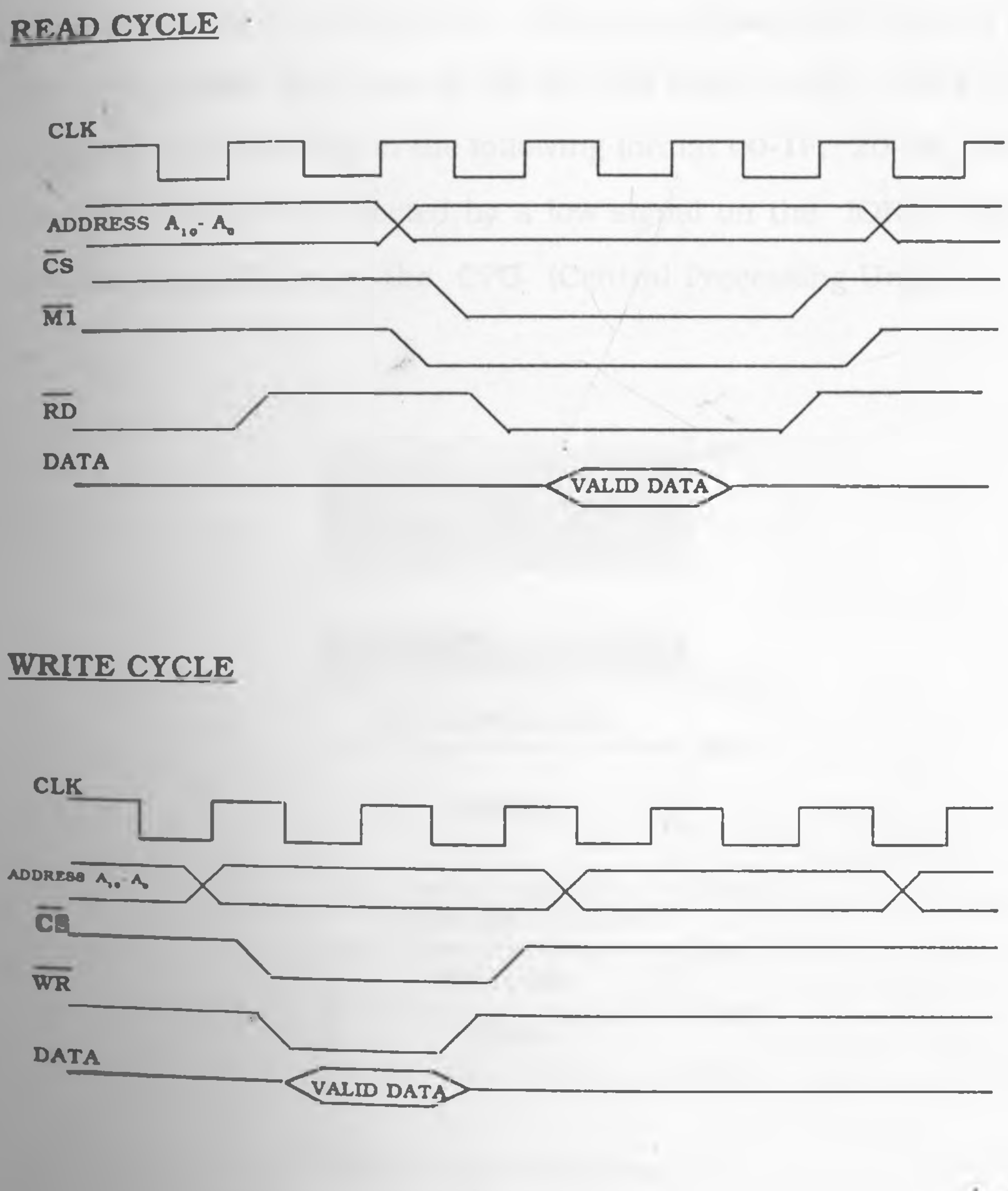


Fig. 2.5.2. ROM Read Cycle - Timing Diagram

he RAMS are accessed in the same way as the ROMS but they have an additional write line for transferring data into memory. The write signal - WR must go low before data is written in the RAM selected by the chip select (CS) signal from the decoding chips. The timing diagram of the RAM read and write cycles are shown in figure 2.5.3.


READ CYCLE

WRITE CYCLE

Fig. 2.5.3. RAM Read and Write Cycles - Timing Diagram

2.6 Input /Output Interface and Signals

The input/output map is shown in figure 2.6.1. The three prime I/O devices are the real time clock, the input output parallel ports and the analog to digital converter. The input output device is selected by the address lines A7 to A0. The 74LS138 decoder is used to decode the initial three addresses, these range from A7 to A5. Using these three signals, all the 256 input/output ports are mapped in to 16 areas in the following format 00-1F, 20-3F... etc. The I/O devices are selected by a low signal on the IORQ (Input Output Request) pin on the CPU (Central Processing Unit).

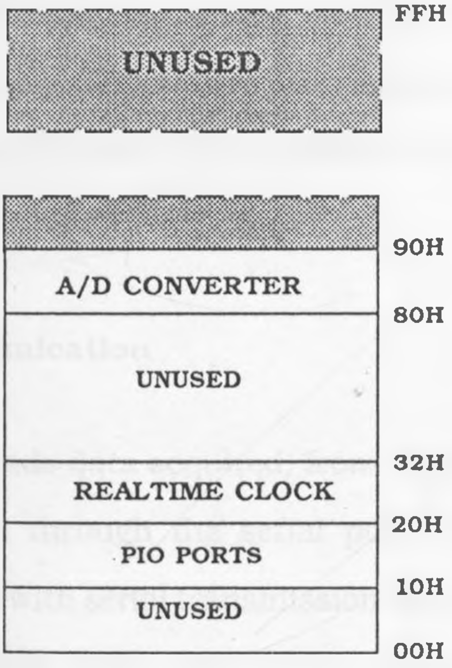


Fig 2.6.1. Input/ Output Map

2.6.1 Input /Output Ports

The Nascom interface kits were replaced by the 8255 parallel input/output ports. The 8255 was preferred to the components used in the MIS modules because they can be programmed and are cheaper to implement. The 8255 is a general purpose parallel microprocessor compatible input/output peripheral chip. This chip has three ports A, B, and C which are programmed to be either input or output ports. In this work, two chips are used and are mapped from location 10H to 17H. Ports 10H, and 12H are programmed as input ports and port 11H is configured to operate as an output port. The remaining ports 14H to 16H are not initialized in the monitor program even though they are used by some applications programs such as the printing service routine. They are available for any applications required and are programmed by writing a word to the control Ports 13H and 17H. A table of the control words is shown in table 4.4.3.2 of chapter 4

2.6.2 Serial communication

The system down loads data acquired, from memory into a larger system for analysis, through the serial port (6402 UART). The hardware associated with serial transmission is relatively simple. All the address lines, data lines and control signals are channelled through the parallel output ports. Data is transferred from memory to the UART when the input transfer buffer is empty and the system is ready to send some more data. This communication through a parallel port provides a means of isolating any external influences interfering with the system.

2.6.3 Audio Tape Interface

The data to be stored on a magnetic tape is transmitted serially to the tape recorder via port 14H of the 8255, through a four wire cable.

2.7 Buses

The three PCBs are inter-linked by using two buses created for this purpose. The buses are:-

1. The General bus
2. The Input/Output bus

UNIVERSITY OF NAIROBI
CHIROMO LIBRARY

The General bus can be used for interfacing memory or various Input/Output devices to the main board. The General purpose bus lines are shown in figure 2.7.1 including all the signal lines available on the Z80, +5V, the Ground and also avails six 8K memory select lines mapped from location 4000H onwards. The RAM expansion board uses all six of these decoded signals to access the memory. The decoding for the RAM board is shown in figure 2.7.2.

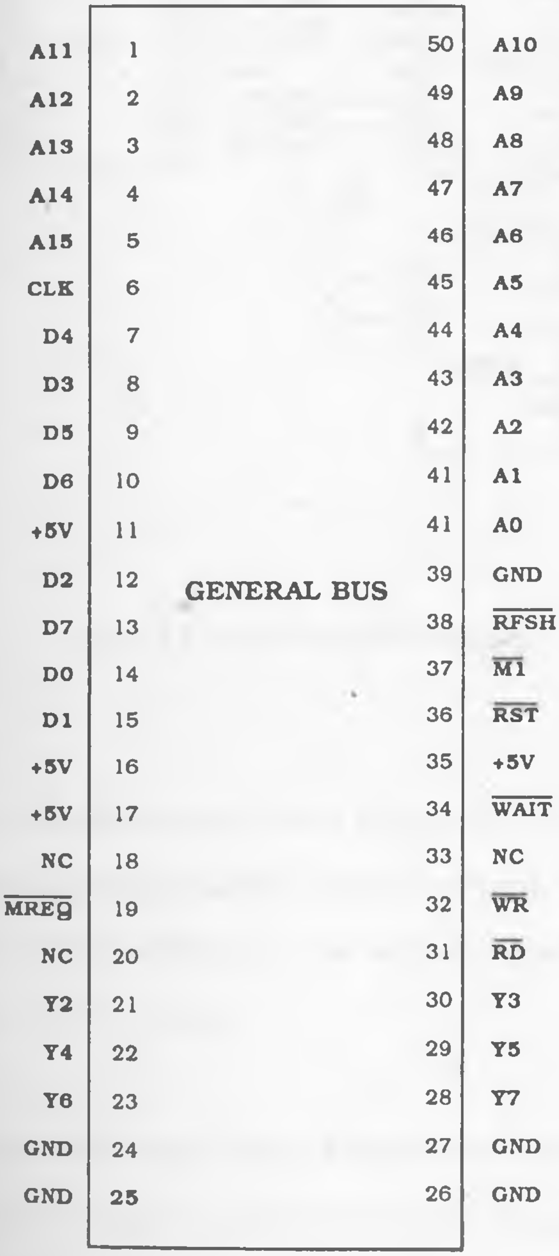


Fig. 2.7.1 The General Bus Pin Connections

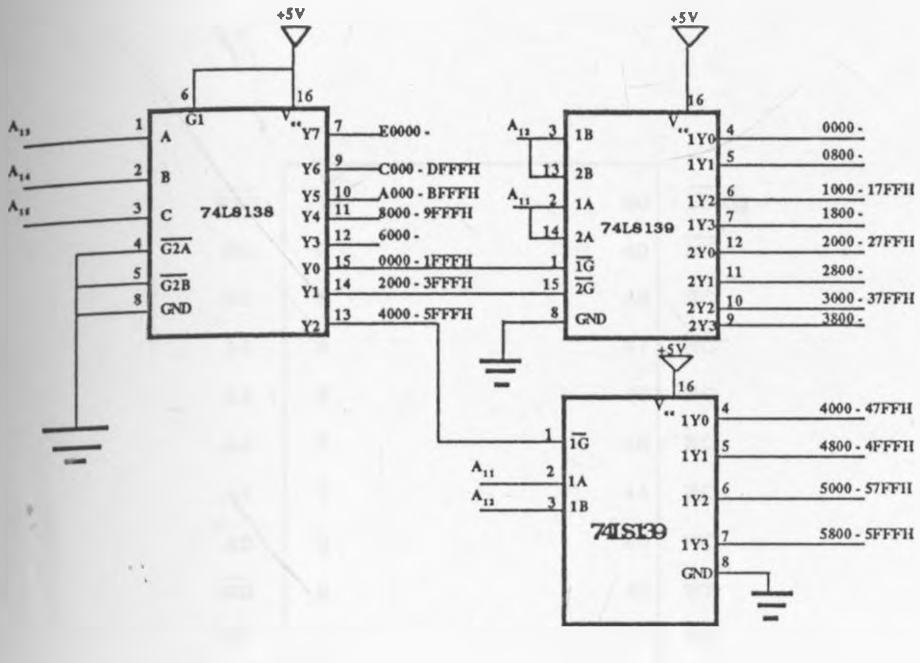


Fig. 2.7.2. Main Board RAM Decoding

The board is not buffered since it was designed to be fully compatible with the main board whose bus is already buffered. Memory locations on the RAM board are accessed in the same manner to that described for the main board memory.

A second bus on the main board avails only the relevant signals required for input output operations Unlike the general bus it is limited in its scope of application. The Input/Output board is designed to be plugged into this bus. A pin diagram of the Input/ Output bus is shown in figure 2.7.3.

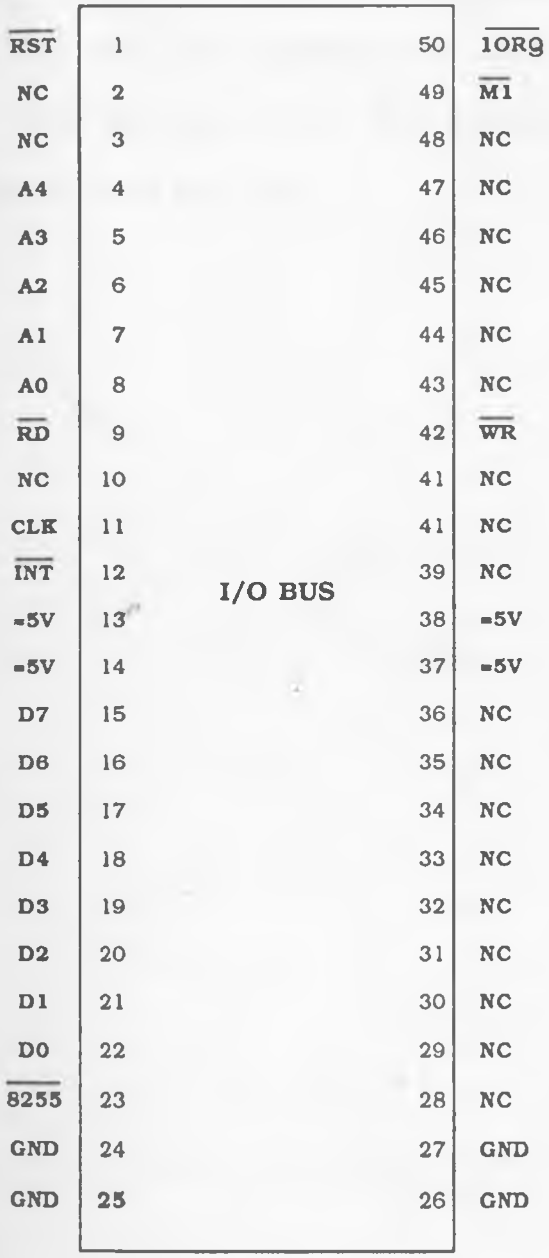


Fig. 2.7.2. The I/O Bus Pin Connections

2.8 Conclusion

The integrated system hardware has been presented and described in this section. Only the bare minimum of hardware is utilized zxin it's design. The development of this integrated system is presented in chapters three and four.

Chapter 3

Hardware Development

3.1 NASCOM MICROCOMPUTER SYSTEM GENERAL OVERVIEW

The Nascom-2 microcomputer system used in the development of this project has a Z80 microprocessor operating on a 4MHZ clock. It has a user friendly screen editor, and monitor program that allows quick entry and editing of programs in machine code. It is equipped with a dis-assembler (NAS-DIS) that provides a means of dis-assembling automatically the hand-coded program entered in the computer. Therefore, errors resulting from the hand-coded programs and mistakes made in entering the code in the computer can easily be detected and corrected. The output format of the dis-assembled program can be determined by the user on executing NAS-DIS, which resides in three EPROMs from locations C400H to CFFFH. The dis-assembler operates in two main modes; a simple mode for direct examination of memory and a more complex one, for producing labeled listings and source files[13]. When operating in the advanced mode the user has the flexibility of selecting the desired options in the dis-assembled document which include among others a cross reference listings of all the labels used in a program, and the assembler (ZEAP) source files.

Programs can also be directly developed using ZEAP which is available on disc and normally occupies and utilizes locations 1000H to 2000H as workspace (commonly referred to as an edit buffer). ZEAP is cold started at location 1000H and all its edit parameters are initialized to their default values and the workspace cleared. If desired it can be warm started at location 1003H without changing the edit buffer. The debugging of programs is readily facilitated by the use of single stepping facility which also provides a full display of all machine registers.

The Nascom system utilizes all the available data lines, address lines and control signals on its Z80 microprocessor to create a bus (NAS-BUS) around which the data acquisition system was developed. Fig. 3.1.1 shows the pin connections on the NAS-BUS. Using this bus system the computer can be easily expanded to meet user hardware requirements, ranging from memory expansion to interfaces for varied interactions with the external world. The system is equipped

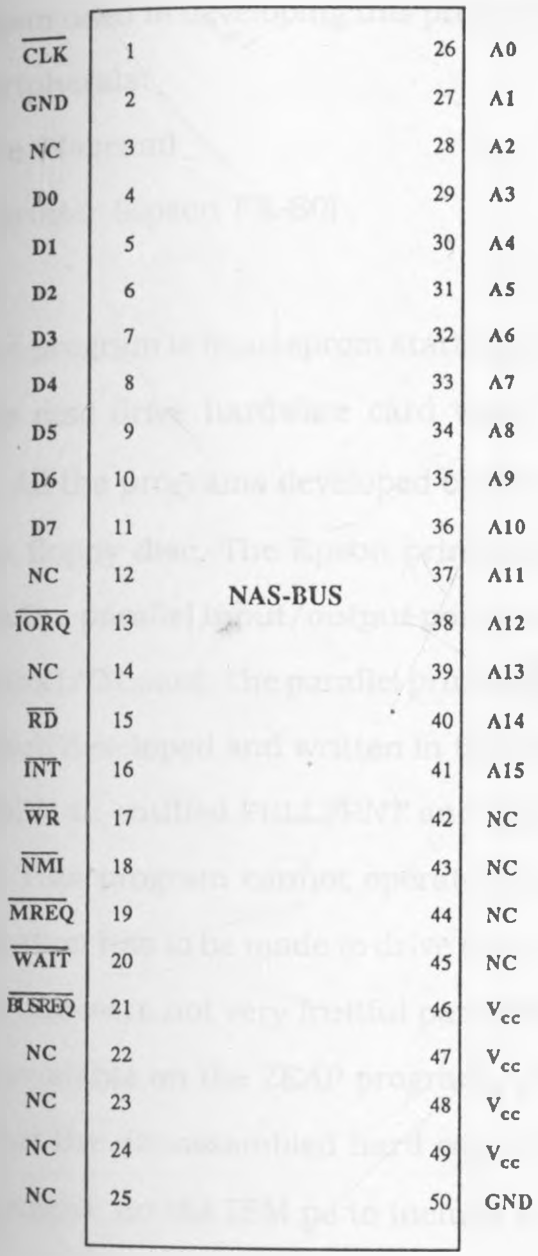


Fig. 3.1.1. The NAS-BUS Pin Connections

to fully support serial terminals. However, with little modification in software and hardware, it can also support parallel terminals. The Nascom system used in developing this project was supported by the following peripherals:

1). Disc drive (Nascom)

2). Parallel printer (Epson FX-80)

The disc drive program is in an eprom starting at location D000H and supports the disc drive hardware card which fits directly on the system bus. All the programs developed on the Nascom were stored on a 5.5inch floppy disc. The Epson printer was connected to the computer via the parallel input/output port PL4. This port is located on the Nascoms I/O board. The parallel printer is driven by a relatively simple program developed and written in the Department of Physics by John Nicoll[14]; entitled PRLLPRNT and starts executing at location 0C80H. This program cannot operate directly with ZEAP and some modification has to be made to drive the printer from ZEAP. Attempts to do this were not very fruitful partly due to the insufficient information available on the ZEAP program. The printer was useful in printing out the dis-assembled hard copy of the program which was further edited on the IBM pc to include comments and proper labels.

The Nascom system also has other capabilities that were not directly useful to the development of this project but are worth mentioning.

here. The main one is its BASIC program that occupies the area E000H to FFFFH, and enables the system to be programmed in BASIC. The real time clock module in the development stage, was tested by a program written in BASIC. The listing of this program is shown in appendix C1. The Nascoms microprocessor clock can be switched to operate at two speeds, 4Mhz and 2Mhz.

The following were the disadvantages and shortcomings of the Nascom-2 as a development system that made the evolution of this project cumbersome and difficult:-

1. There is no repeat facility for the key board thus the speed of editing the program was severely reduced as time was wasted in moving the edit cursor on the screen.

2. The NAS-DIS system has to be executed every time the system is reset.

3. The source program could not be printed from the ZEAP dis-assembler hence no proper assembly program listing is available.

4. The Nascom is not a full development system hence it could not be used to check the hardware of the resulting system. Since it does not have 'ready made' programs for hardware debugging.

5. The program development was highly dependent on the Nascom's

memory map, hence the memory map of the resulting system is not optimally designed.

3.2 The Microcomputer Interface (MIS) Kits

These kits, described by W.H.Drake[15] and P.R.Kirk[15a] were jointly designed by the Department of Physics, University of Nairobi and Department of Electronics, University of York. They were developed for quick and easy interactions of The Nascom system and the external world. Only two of the modules utilized for this work are described in this report. The MIS kits are designed to simplify the user input/output operations. The kits provide the user with a virtual device to communicate with the external world without much concern of the details of operations taking place in the communication, at the hardware level.

The kits are connected to the Nascom computer using a ribbon cable that avails all the systems control signals, address signals and data lines.

The two modules used were the Decode module and the Digital to Analog converter, (D/A) module which are described in the next paragraphs.

3.2.1 Decode Module

The top view of the Decode module is shown in appendix A1. All the necessary connections are shown on the module. The desired circuit can be connected using jumper wires, plugged into pins on top of the module, hence all changes and modifications were made with little difficulty. Eight address lines A0 - A7 have been decoded to provide a total of 256 input/output ports (I/O) all of which are made available to the user. However, only 16 ports are used simultaneously on the same module, which is more than what is needed on the average for most applications. In the development of this work only four ports were used simultaneously. Decoding is implemented by using two 74LS154 ICs. The upper address lines A4 - A7 are decoded to provide the upper nibble of the port address which enables the chips' decoding the lower address lines A0 - A3. The control lines are buffered for use with the ports on the same module. The following control signals are available :

1). Input/Output request ($\overline{\text{IORQ}}$)
2). Read signal($\overline{\text{RD}}$)
3). Write signal ($\overline{\text{WR}}$)
4). Memory request ($\overline{\text{MREQ}}$)

Also on the same board is an array of eight NOR and OR gates that are provided for any additional logic required for development and interfacing.

3.2.2 Digital to analog module

The D/A module utilizes the 74LS363 input/output tri-state buffer for the data signals which are enabled by signals from the decode module. The data is accessed by enabling the buffers, using appropriate control signals and Likewise the data is latched into the buffer at the right instant. The buffers are cleared externally by applying a negative going pulse to the clear terminal, or through a software routine that sets its data to 0. The top view of the D/A module is as shown in Appendix A-2.

Both modules form a complete interface system. Interactions with the external world is realized through the instructions IN(port) for input communications and OUT(port) for output communications both of which are supported by the Z80 instruction set. Each of these modules has one input and one output port, hence three modules were used in the development phase to provide a total of six ports required in the development.

3.3 Hardware Development

The development of this project depended on the existing Nascom computer and the MIS kits hardware, which have been discussed and described in sections 3.2.1 and 3.2.2 of this chapter. The block diagram for the system development environment is shown in fig. 3.3.1.

The main parts of the system at this stage were:-

1). The Nascom computer

2). The MIS kits

3). Real time clock

4). Key Board

5). The display unit, and

6). The analog to digital converter

The Nascom system provided the memory and the Z80 microprocessor for the system as well as the program development environment for the software. The microprocessor and memory were enslaved to emulate the single board data acquisition system.

Only 8K the of Nascoms RAM was available for use during the development. The memory map of the resulting unit was dictated by the user RAM area on the Nascom system. This is because the program once developed could not be shifted easily to operate from another location due to the use of absolute jumps and the numerous vectors in the writing of the program. The memory map of the development system is shown in fig. 3.3.2.

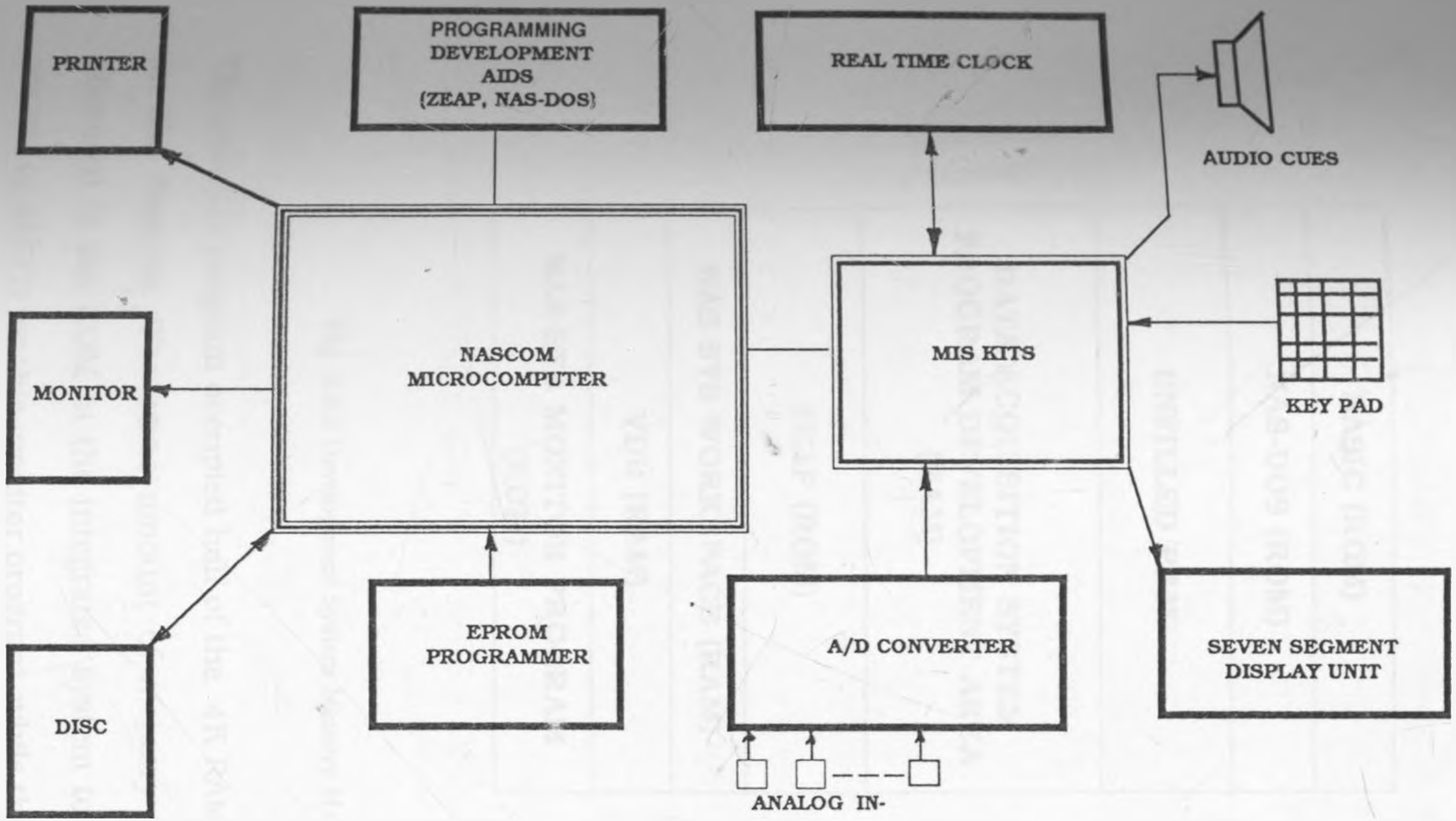


Fig 3.3.1 System Development Block Diagram

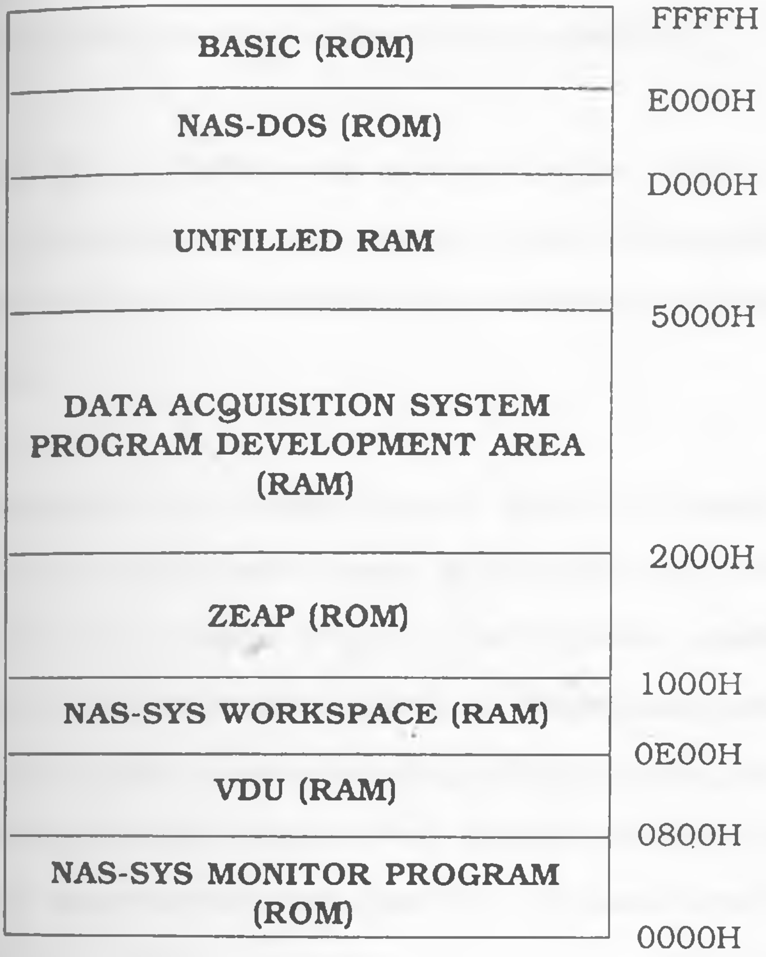


Fig. 3.3.2 Development System Memory Map

The monitor program occupied half of the 4K RAM space available on the Nascom. The same amount of memory space was later allocated to the ROM in the integrated system to occupy address 3000H to 37FFH for this monitor program, while the remaining area from 37FFH to 4000H was reserved for any other additional monitor program subroutines and for application programs. The stack was not initialized in the development system since an attempt to do so interfered with the operation of the Nascom computer.

Locations 2000H to 2260H were reserved for the storage of data required by the system monitor program, and locations 2661H to 3000H were used both for the system's workspace and temporary data storage.

The Z80 microprocessor always begins to execute the monitor program at address 0000H when power is switched on or when the system is reset. The monitor program must therefore commence at this address. It was not possible however to develop the monitor program for this acquisition system starting at that address, because it is obviously the starting address of the Nascoms monitor program. The program was therefore developed from location 3000H and a solution sought to shift it to 0000H before blowing an EPROM for the integrated system.

Either a hardware or software solution could be implemented to solve this problem. In the hardware approach to the solution of this problem, the main program and initialization subroutine are developed from memory address 3000H and are blown on an EPROM. The appropriate microprocessor address lines are then inverted later, to start executing at location 3000H. This design is however wasteful as 8K of memory is lost which is 12% of the 64k capacity for the whole of the acquisition system. The software solution was therefore preferred; this involved developing the main program from address 3000H and an initialization subroutine routine from address 1000H on the Nascom microcomputer. The initialization program was then shifted to location 0000H by changing the most significant digit of the addresses from 1 to 0 including all the call and jump commands pointing to addresses in these locations, as the EPROM was blown. No errors resulted in this change because the initialization subroutine was relatively short.

The fixed data for the program was allocated location 2000H to 23FFH. This includes any data locations needed for additional programs that may later be developed and added to the main monitor program. Locations 1200H to 13FFH are reserved for the system and user stack, while the system RAM location starts at location 3800H to 4500H.

3.4 Keypad

3.4.1 Physical Layout

Fig. 3.4.1.1 shows the physical layout of the keypad. It has two main sections, the Numeric section that has the hexadecimal digits 0 to F, and the register and function keys section for editing the program

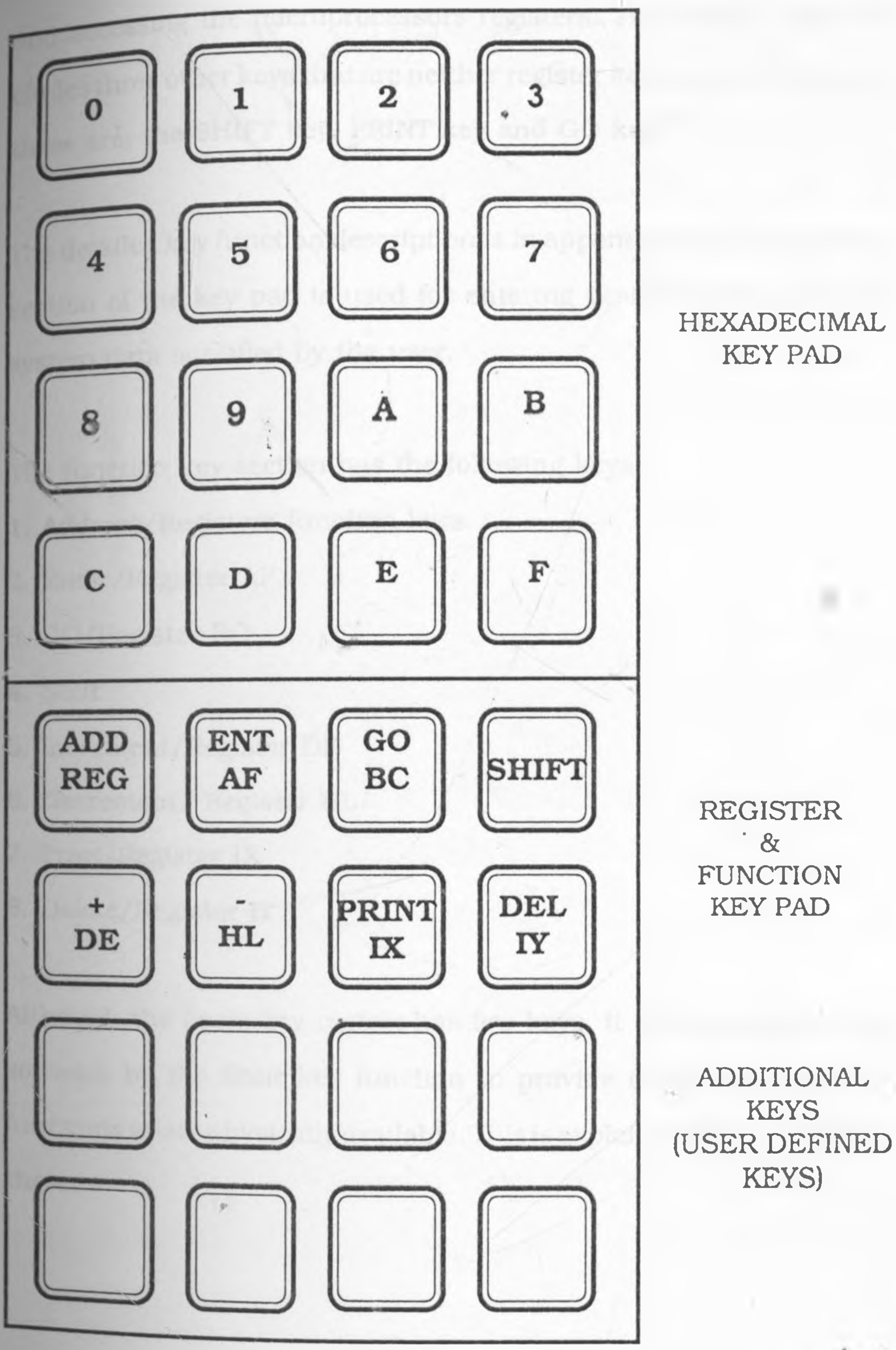


Fig. 3.4.1.1 Keypad Physical Layout

and accessing the microprocessors registers. This section also includes three other keys that are neither register keys nor editing keys, these are; the SHIFT key, PRINT key and GO key.

The detailed key function description is in appendix B2. The numeric section of the key pad is used for entering desired addresses and system data specified by the user.

The function key section has the following keys :

1. Address/Registers function keys
2. Enter/Register AF.
3. GO/Register BC
4. Shift
5. Increment/Register DE
6. Decrement/ Register HL
7. Print/Register IX
8. Delete/Register IY

Although the basic key matrix has few keys, it is expanded in the software by the Shift key function to provide twice as many key functions as are physically available. This is explained later in chapter three.

3.4.2 Circuit Configuration

The design of the key pad circuit is displayed in fig. 3.4.2.1. The pad which is non-encoded consists of pressure (or touch) activated switches arranged in a 6 x 4 matrix fashion[16]. Apart from the voltage pull-up resistors, it does not include any hardware to detect a key or hold data until a new key is pressed. All these functions are analysed and performed by a software routine with the minimal hardware. Thus possible hardware malfunction would only result from either broken terminals or dirty switch contacts.

Two common problems associated with keypads have been considered in the design, these are:-

1). Keybounce, which refers to the fact that when contacts of a mechanical switch close, they vibrate for a short time before closing[17]. The key bounce problem is illustrated in figure 3.4.2.2.

2). Roll-over, which is the problem caused by multiple keys pressed simultaneously.

Both keybounce and roll-over are remedied by software routines KYBNCE and RLLVER respectively.

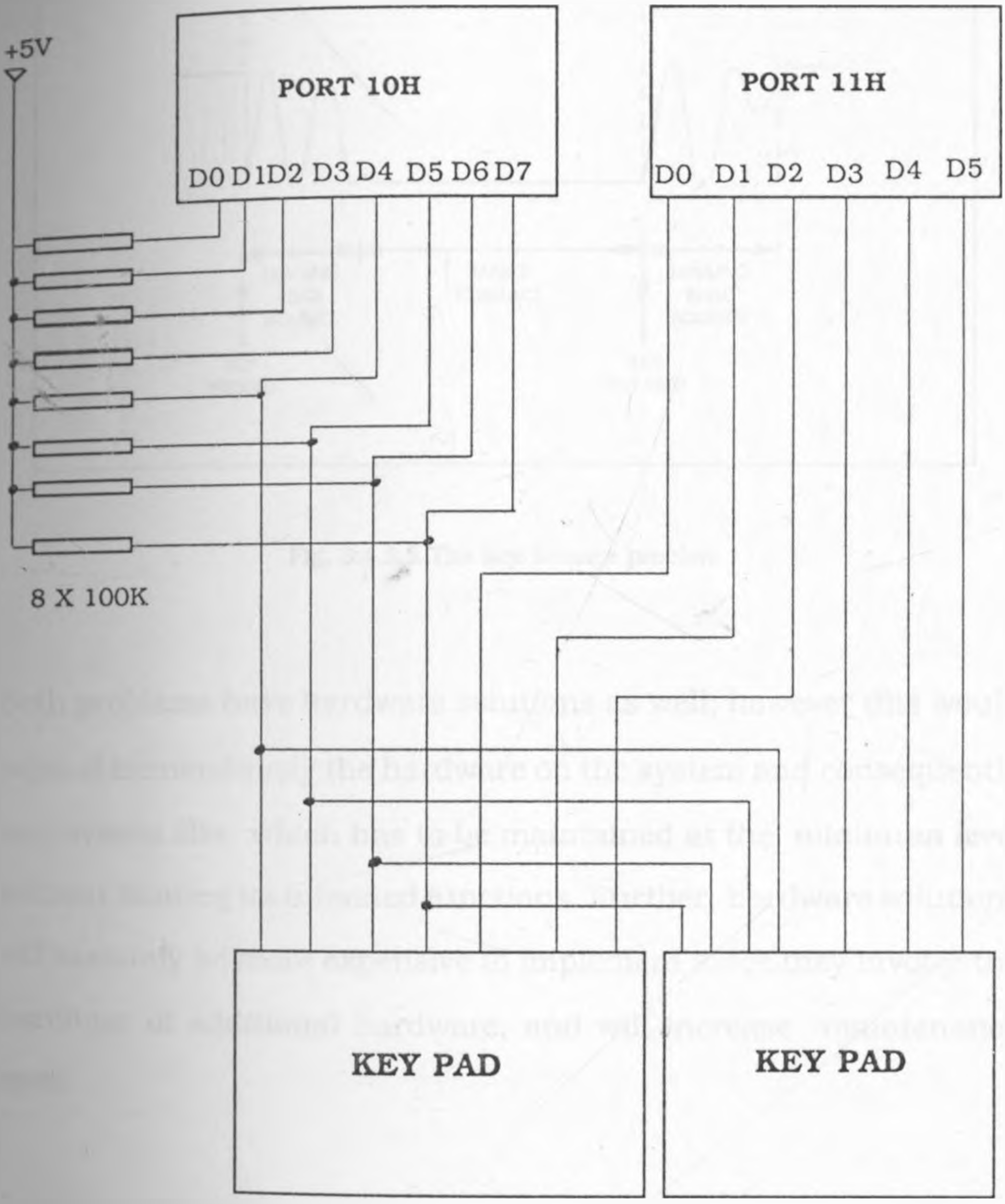


Fig. 3.4.2.1. Key pad Circuit Diagram

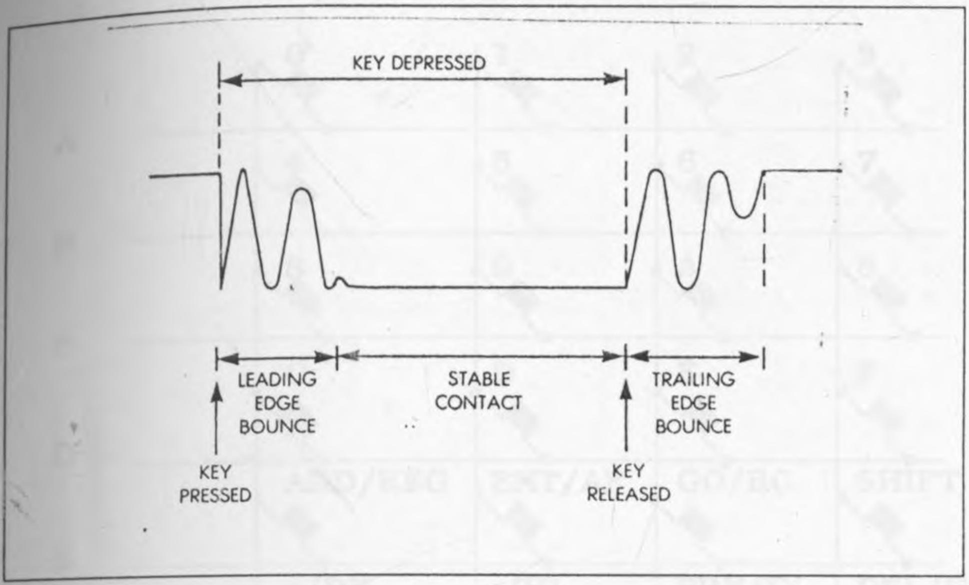


Fig. 3.4.2.2 The Key bounce problem

Both problems have hardware solutions as well, however this would expand tremendously the hardware on the system and consequently its physical size, which has to be maintained at the minimum level without limiting its intended functions. Further, hardware solutions will certainly be more expensive to implement since they involve the purchase of additional hardware, and will increase maintenance costs.

3.4.3 Pin Allocation:-

The matrix is connected to the input and output ports of the interface kits as shown in fig. 3.4.3.1. The ports are scanned by a software -

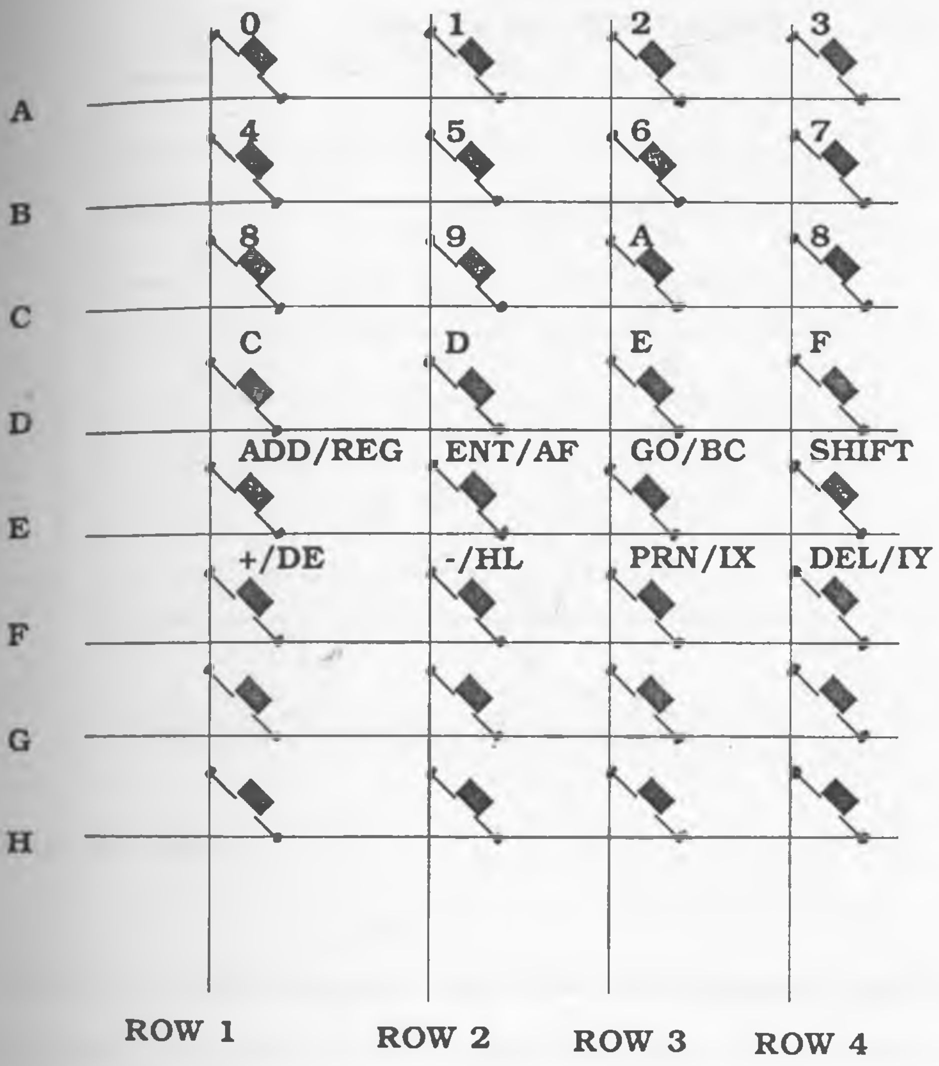


Fig. 3.4.3.1. Key Pad Matrix Schematic Diagram.

routine which writes to the keypad rows and reads the columns of the keypad matrix to detect a coincidence. The pins are allocated to the input port 10H and output port 11H as follows:-

PIN POSITION D0 - D7	IN/OUT PORT	
	PORT 10H (IN) COLUMNS	PORT 11H (OUT) ROWS
0	1	A
1	2	B
2	3	C
3	4	D
4	HI	E
5	HI	F
6	HI	G
7	HI	H

Table 3.4.3.1 In/Out Ports Key Pad Allocation

3.5 Display Unit

The display unit consisted of an array of six seven segment display units arranged to enable the user to read the input when entering programs or specifying control parameters. The status of the system is therefore monitored through these LEDs. A red filter is used to enhance the visibility of the display. The seven segment display units are TTL compatible and are directly interfaced to the output ports on the interface kits using transistors to drive the individual segments, and to enable each unit of the seven segment display system. The transistors also sink the excess current (40mA) to the LED as shown in fig 3.5.1.

The current limiting resistors R were determined by the following simple equation:

$I = (V_{cc} - (V_d + V_{cc(sat)}))/R$

where the used variables have the following meaning:

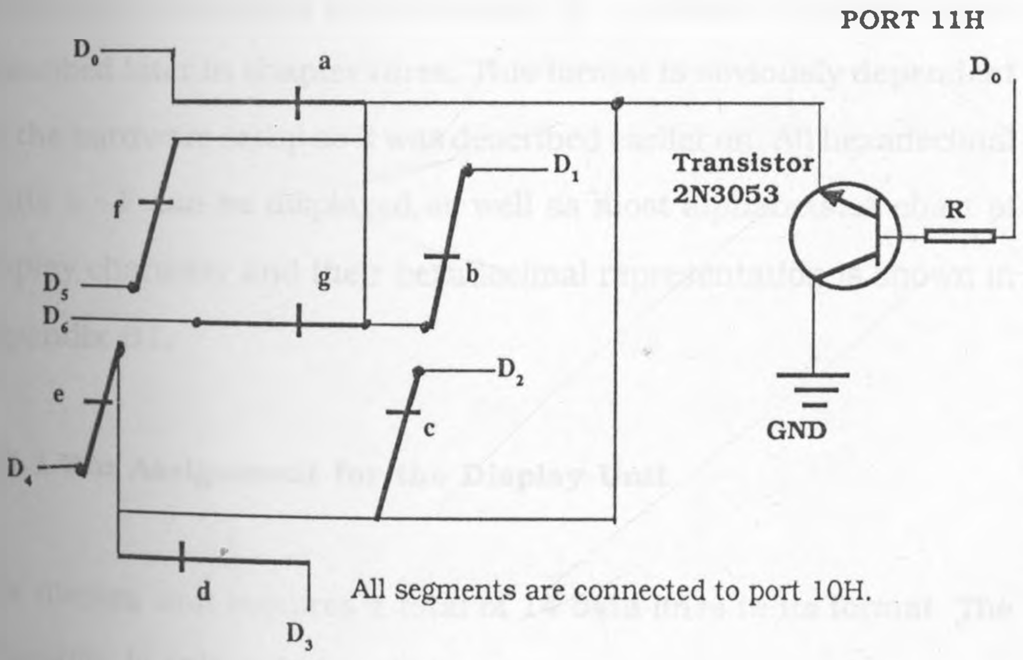
V_{cc} - the supply voltage (+5V)

V_d - the port output voltage (+4v)

V_{cc}(sat) - the collector/emitter saturation voltage (+0.5V)

I - the Maximum allowable LED current

The resistance R from the equation above was determined as 1.2K.



All segments are connected to port 10H.

Fig. 3.5.1 A Single LED Interface

Figure 3.5.2 shows the complete detail of circuit interconnection for the display system. The segments are driven by the pnp transistors 2N2904 and the units are enabled through the npn transistors 2N3053.

Transistors have been preferred in this design to IC drivers even though the later would be more compact and would reduce the number of components greatly. This is because the transistor design is easier to implement and certainly cheaper to maintain than the corresponding IC unit. The driver chips are not readily available locally whereas the transistors are. This advantage overrides all the other considerations.

The format of display is determined in a software routine and is described later in chapter three. This format is obviously dependent on the hardware setup as it was described earlier on. All hexadecimal digits 0 - F can be displayed as well as most alphabets. A chart of display character and their hexadecimal representation is shown in appendix B1.

3.5.1 Pin Assignment for the Display Unit

The display unit requires a total of 14 data lines in its format. The allocation is split into two groups:-

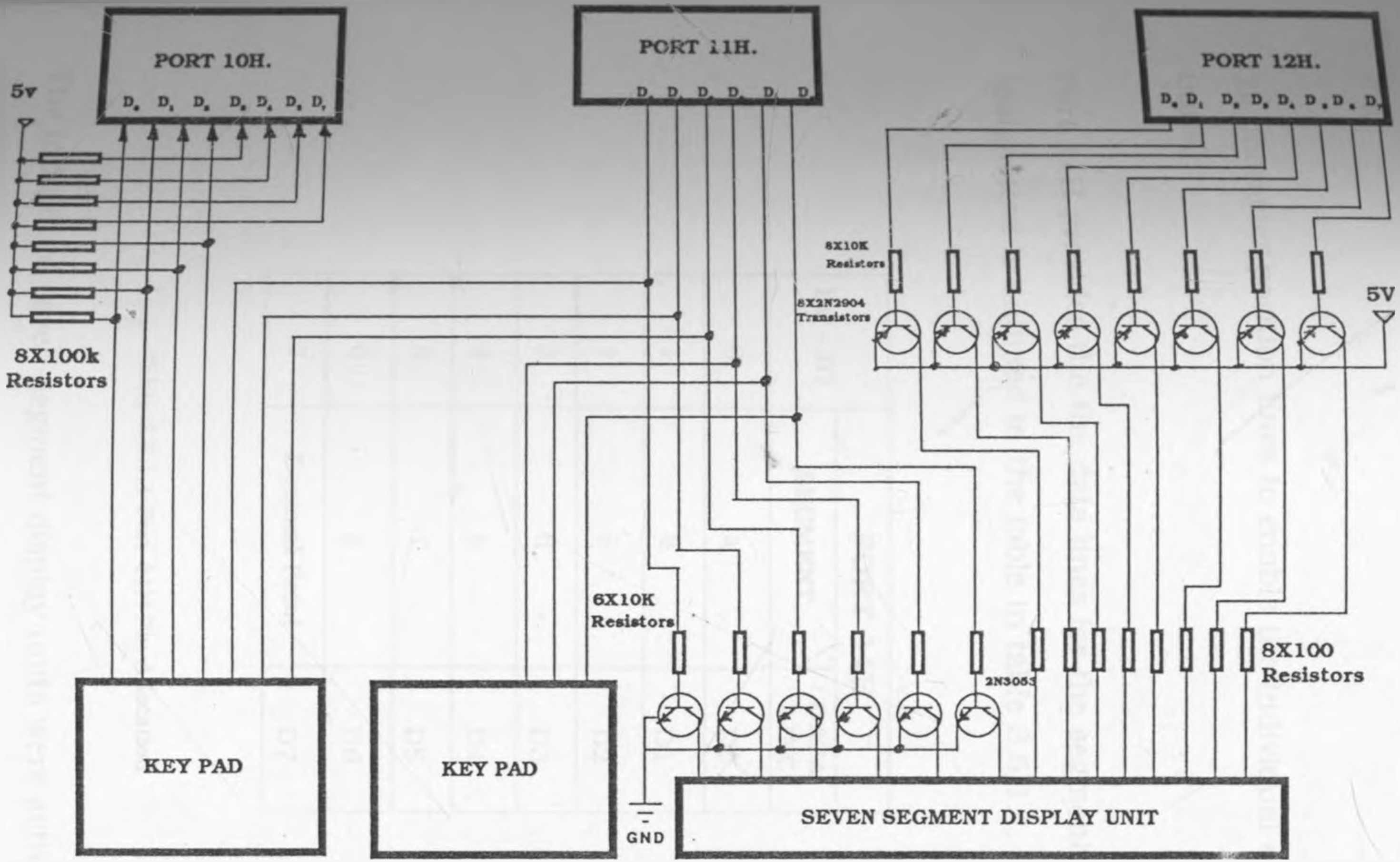


Fig. 3.5.2 Seven Segment Display System Schematic Diagram

1). Data lines to enable the individual segments

and

2). character position lines to enable the individual seven-segment units.

Port 11H provides the the data lines for the segments. and the pin assignment is outlined in the table in table 3.5.1.1.

D0 - D7	PORT 11H	
	SEGMENT	DATA LINE
0	a	D0
1	b	D1
2	c	D2
3	d	D3
4	e	D4
5	f	D5
6	g	D6
7	Decimal Point	D7

Table 3.5.1.1 Port 11H Pin Allocation

The individual seven-segment display units were activated through port 10H. The pin assignment for this is shown in table 3.5.1.2.

UNIT POSITION (Left to Right)	PORT 10H
	DATA LINE
0	D5
1	D4
2	D3
3	D2
4	D1
5	D0

Table 3.5.1.2 Port 10H Pin Allocation

3.5.2 Operation of the Display Unit

The display unit has six LEDS in a row each LED representing a character position numbered from left to right. Eight lines are used for the segment control and six for character position select. To display a character a corresponding word is issued to the segment lines and character position line is enabled.

3.5.3 Scanning the Seven segment Display

Two outputs are necessary for the seven segment display format: An eight bit output for the word to be displayed Wa, Wb,...... to Wg and decimal points as shown in fig. 3.5.3.1. A six bit control word is issued to position enable lines P1, P2,......,P6 for character position select. The seven segment display unit is continually scanned by the subroutine DISPLY in the monitor program.

With reference to figure 3.5.3.1 the principle of scanning is as follows:-

The program outputs a digital signal to select the character position. The hardware is configured to have only six such character display positions. The segment control lines Wa to Wg of the corresponding word for display are also activated at the instance the character position select signal is active. For instance if the digit position selection line is P1, then the first position ie. the left most position of the seven segment display unit is activated. If the segment control !ines are holding a word 7F (BCD value 01111111) all the segment lines are then active, an eight("8") will be displayed in the left most position of the display unit.

The procedure of scanning the seven segment display unit method is summarized as follows:-

Apply a signal voltage to the character position selection lines P0, P1,......... Pn in the desired sequence. When the character selection line is activated, a voltage signal containing the corresponding output word format is applied to the segment control lines Wa, Wb,......Wg. After the digits are scanned once, the scanning is then repeated from

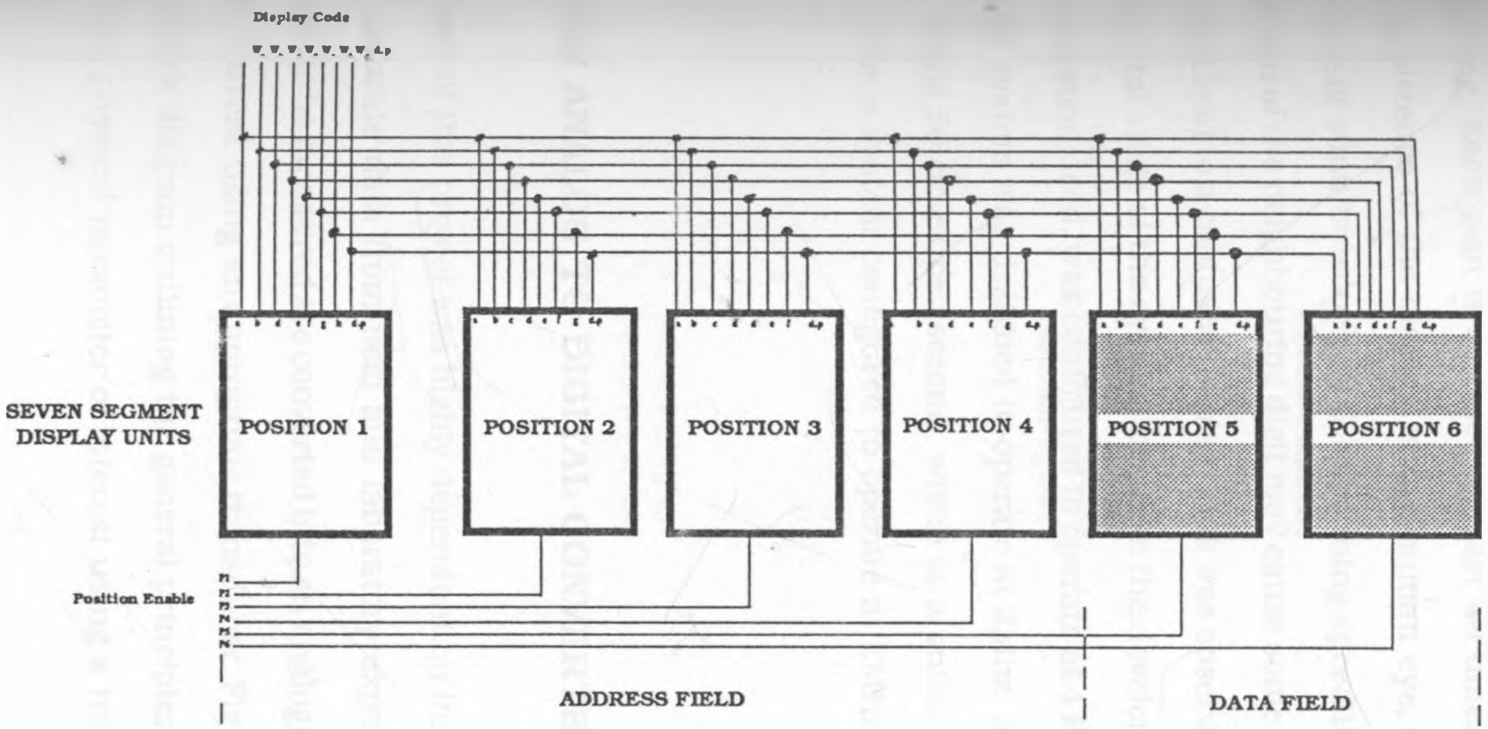


Fig. 3.5.3.1. Seven segment display scanning.

the beginning. Each digit is scanned at least 40 times per second. Due to persistence of the display to the human eye, all the digits appear to be lit simultaneously. If the scanning speed is too high the residual light of the neighbouring digit may cause some ghost images which might lead to a confused display. This was observed during the developmental stage of the system, because the development microcomputer system used, was configured to operate at 4 MHz while the acquisition system was designed to operate at 2Mhz. DISPLY scans the key board 50 times per second, which is a sufficient scanning frequency for a system configured to operate at 2MHz.

3.6 THE ANALOG TO DIGITAL CONVERTER

The success of this project was highly dependent on its capability to acquire accurate data from field and laboratory experiments. The parameters to be measured are converted into an analog signal, either voltage or current using an appropriate transducer. Fig. 3.6.1 below shows a block diagram outlining the general principles of measurement of any physical parameter of interest using a transducer[18].

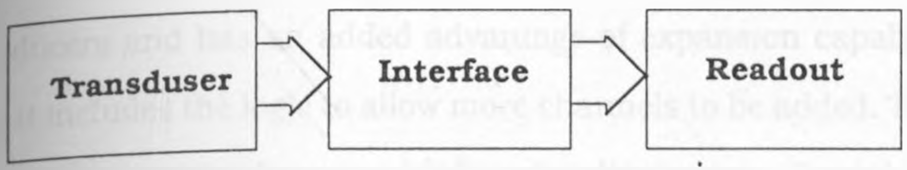


fig. 3.6 1. Principle of Measurement of a Physical parameter using a Transduser[19]

In this project the physical parameters of interest deduced from the project requirements of section 1.5 include:- the measurement of light intensities, temperature, and current or voltage. The analog to digital converter is chosen to meet the general requirement for measurements of the above parameters. The transducer analog signal is converted into digital form with the 16-channel multiplexed A/D converter AD0816, which is described in the next section.

3.7 General Description of the A/D Converter

The AD0816 CMOS device incorporates both a multiplexer and an analog to digital converter on a single chip. The successive approximation conversion technique is implemented. This features have resulted into a system with a high resolution, fast, accurate and cheaper to use than discrete ICs. The conversion from analog to digital is achieved entirely by the hardware. However, the channel and data select and storage is done by software routines. The 16 channel multiplexer is used to access any of the 16 input analog signals from the transducers and has an added advantage of expansion capability, since it includes the logic to allow more channels to be added. There is no need for external zero and full scale adjustments. The address inputs are already latched and decoded internally. Therefore it is simple to interface to a microprocessor. Data is transferred through the TTL tri-state outputs. It has an added advantage of low power consumption of 5mW at 298 Kelvin and can withstand temperatures in the range of 218k to 423k which covers sufficiently the laboratory and field operating temperature range requirements of between 10C and 40C.

The device comes in 40 pin DIP. The pin diagram of AD0816 is shown in fig 3.7.1. The analog voltage is quantized into 256 levels and takes 100uS to convert the signal. This is equivalent to 10 clock pulses when the device is operating at 1MHZ[20].

3.7.1 Pin Description and Circuit Operation

Fig. 3.7.1.1 and 3.7.1.2 show the circuit diagram and the development block diagram respectively of the A/D converter stage of the acquisition system. A small dc signal from a signal generator varying in the range of 20 60 mili-Amps was used to test the operation of this

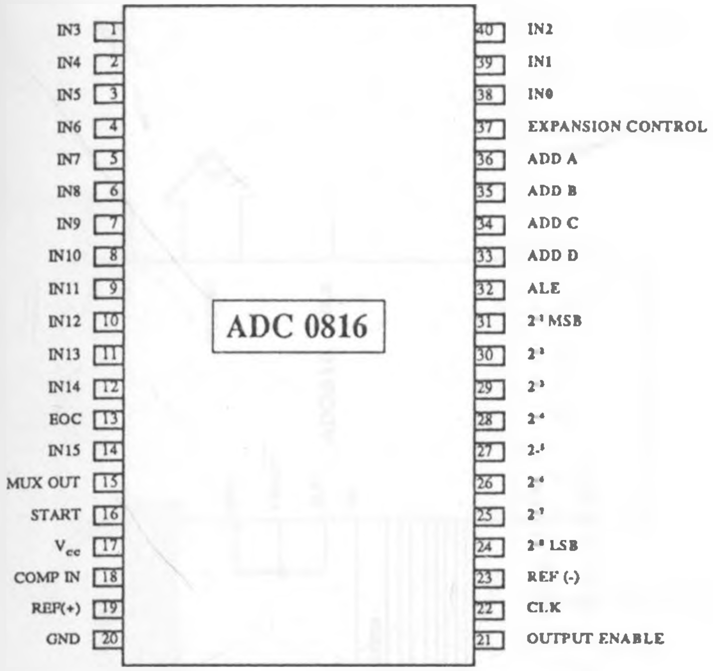


Fig. 3.7.1. Pin Configuratuion of ADC 0816

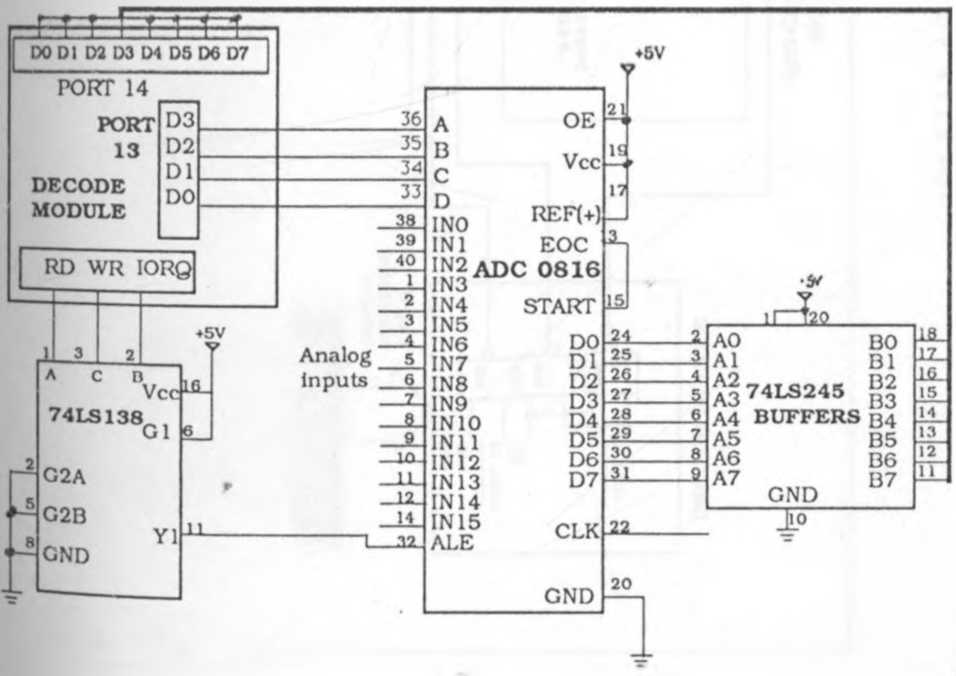


Fig. 3.7.1.1 ADC 0816 Interface Circuit Diagram

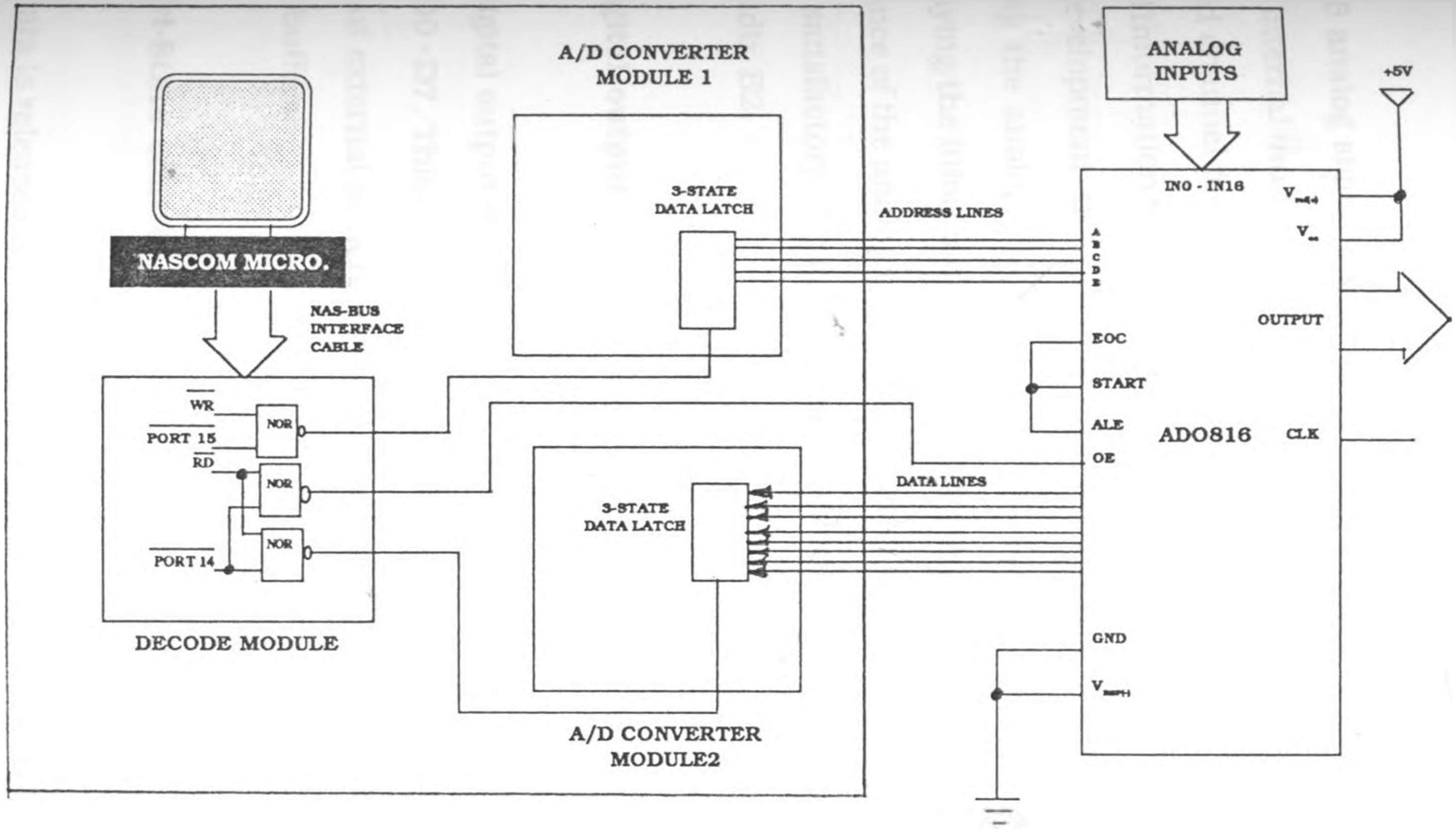


Fig. 3.7.1.2. A/D converter interface diagram

circuit.

i. Analog input pins (IN0 - IN15)

The 16 analog signal inputs to be measured are directly connected to the internal multiplexer through the input pins IN0 - IN15 and the desired channel monitored through a software routine that outputs digital information to the device to select the channel required. During the development the 16 channels were tested independently by shifting the analog signal input as the program was running and displaying the information on the Nascom system monitor. The performance of the analog to digital converter within the laboratory was quite satisfactory. Typical values obtained in this test are shown in appendix B2.

ii. Digital output data (D0 -D1)

The digital output of the converted analog voltage was monitored on pins D0 - D7. This output is connected directly to the microprocessor, without external buffers for data lines, since it includes its own tri-state buffers.

iii. Tri-State control.

The data is released from the buffers when the signal on the tri-state buffer control is high (HI). In this design it is enabled by "ORing" the READ and the IORQ signals using the logic available on the decode module.

iv. Address signals (ADD A - ADD D)

The channels are selected and controlled in a software routine that outputs the control and select word at port 15H the pin assignment for the control word is shown in table 3.7.1.1.

PORT 15H	
ADDRESS LINE	PIN ALLLOCATION
A	D3
B	D2
C	D1
D	D0

Table. 3.7.1.1. Port 15H Pin Allocation

the address of the selected analog channels are shown in table 3.7.1.2.

v. Expansion control

The channel select is toggled between the ON and OFF positions using the expansion control line. It is particularly useful in switching off the A/D converter when the microprocessor is no longer receiving

SELECTED CHANNEL	ADDRESS LINES				DECODED CHANNEL	EXPANSION CONTROL
	D	C	B	A		
IN0	L	L	L	L	80	HI
IN1	L	L	L	H	81	HI
IN2	L	L	H	L	82	HI
IN3	L	L	H	H	83	HI
IN4	L	H	L	L	84	HI
IN5	L	H	L	H	85	HI
IN6	L	H	H	L	86	HI
IN7	L	H	H	H	87	HI
IN8	H	L	L	L	88	HI
IN9	H	L	L	H	89	HI
IN10	H	L	H	L	8A	HI
IN11	H	L	H	H	8B	HI
IN12	H	H	L	L	8C	HI
IN13	H	H	L	H	8D	HI
IN14	H	H	H	L	8E	HI
IN15	H	H	H	H	8F	HI

Table 3.7.1.2 A/D Converter Address Lines Input States

data from the later. This line however is continually enabled in the design since there is no need of having additional software to control this line whereas the real time clock performs more or less the same function.

vii. Clock

The clock driving the A/D is operating at 1 MHz this was chosen since it also is the operating speed of the microprocessor in the integrated system, thus eliminating any need for additional circuitry to divide the clock pulses.

viii. Comparator IN

The comparator IN signal takes the signal from the multiplexer or some external circuitry such as an amplifier or signal conditioning circuit. This pin was connected to the multiplexer output - common. The end of conversion signal was connected to the start and address latch enable signals. In this case the converter will continuously convert the analog input signals. This design reduces the complexity of extra control signals that are required to co-ordinate the A/D converter functions.

3.8 The Real Time Clock

The real time clock is used for measuring time intervals between events in the acquisition process. Data can therefore be acquired in real time and analysed later. The development of the real time clock centred around the Nascom microcomputer only (i.e The interface modules were not utilised). The ICM 7170 real time clock chip was used. The pin diagram for the 7170 is shown in fig. 3.8.1[21].

The clock is configured to operate in two modes. First is the interrupt mode in which the time interval for interrupts is pre-programmed by the user to issue an interrupt signal at a desired time of the day.

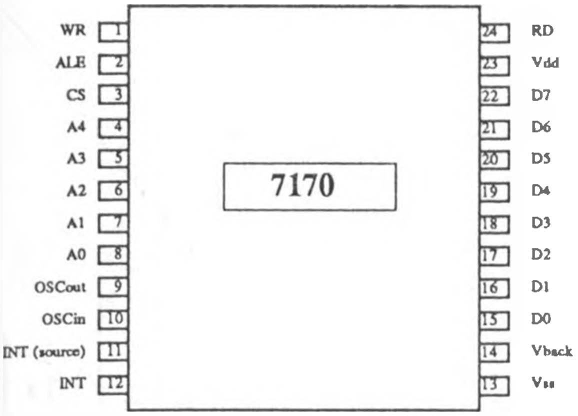


Fig. 3.8.1 7170 Real Time Clock Pin Diagram

Second is the normal mode in which the time at which a measurement is taken is stored in the RAM area of the real time clock which in turn issues an interrupt signal at the set time. All the data and address lines are buffered through the 74LS245 bi-directional buffers both to protect the chip and to drive the data and address lines. A diagram of the real time clock interface is shown in figure 3.8.2.

The address signals A0 to A7 are decoded to access the clock registers from 80H to 91H. The clock design incorporates a 3V battery backup which supports it when the system power goes down or is turned off. The timing diagram for the real time clock interface is shown in fig 3.8.3.

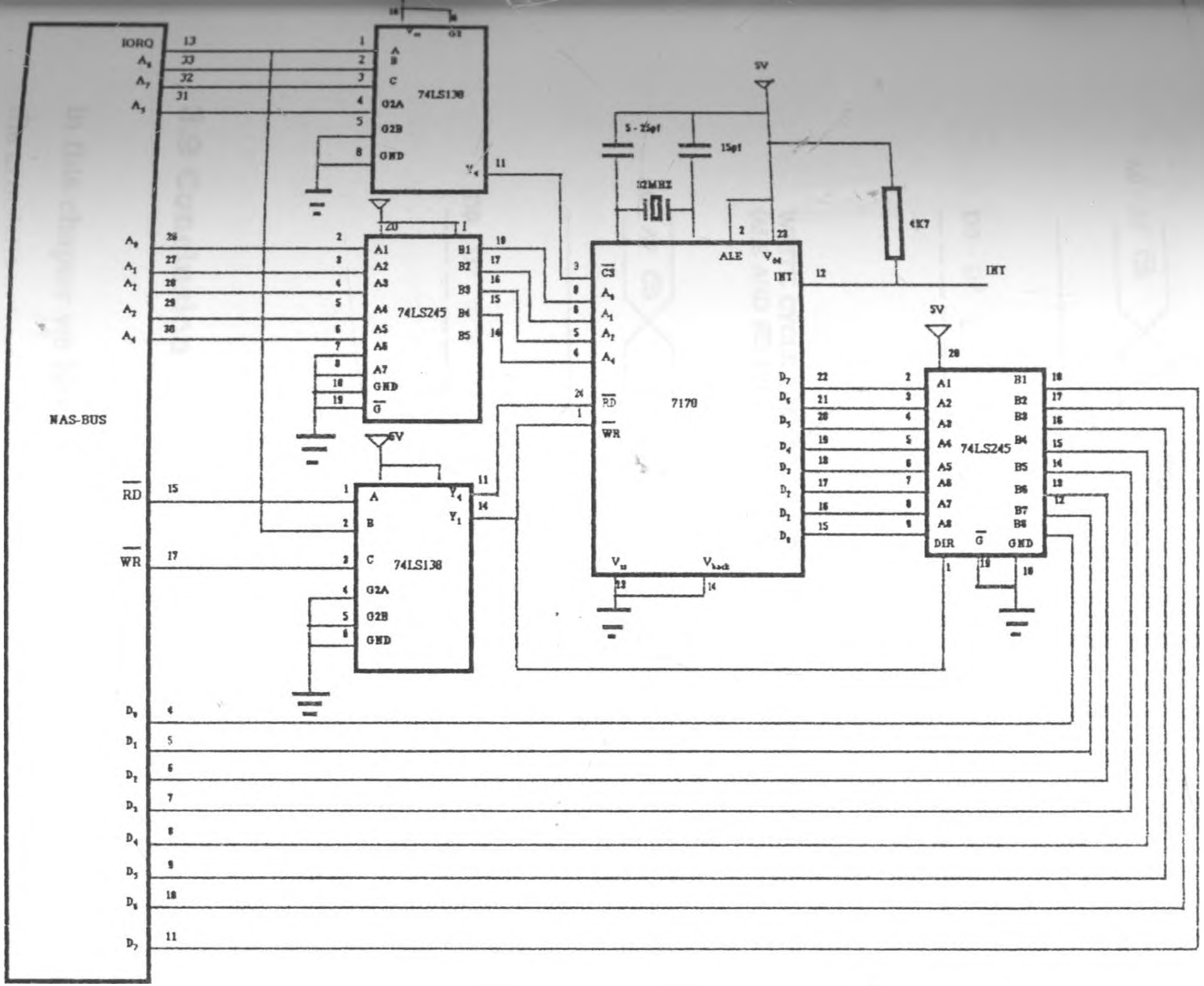
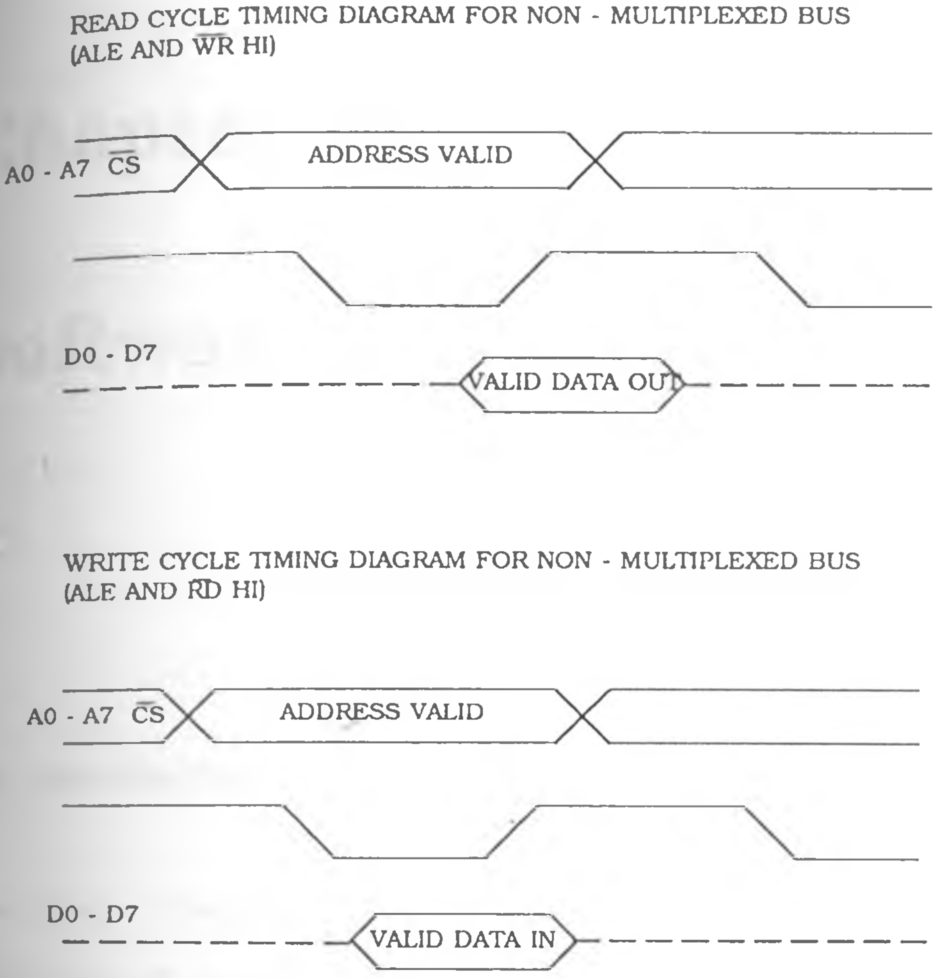


Fig. 3.8.2 The real time clock interface



READ CYCLE TIMING DIAGRAM FOR NON - MULTIPLEXED BUS (ALE AND $\overline{WR}$ HI)

WRITE CYCLE TIMING DIAGRAM FOR NON - MULTIPLEXED BUS (ALE AND $\overline{RD}$ HI)

Fig. 3.8..3. 7170 Timing Diagrams

3.9 Conclusion

In this chapter we have presented the hardware of each module of the acquisition instrument developed on the Nascom microcomputer system. . The separate modules developed and described here were integrated to form the hardware for the stand alone system, which was discussed earlier in chapter two.

Chapter 4

Software Development

4.1 Introduction

This chapter describes the software development on the Nascom microcomputer system along with explaining the operation of the major programs of the acquisition system. The software development is based on a single board microcomputer monitor program which was initially designed by Gerald Kotonya, whose work is gratefully acknowledged.

4.2 Software Development Environment

The software was developed on the Nascom system and designed to run on any other computer or microprocessor based system with a similar hardware configuration. The Nascom together with the interface kits, provided a virtual machine which formed the frame work of the acquisition system's hardware.

Programs were first written in the ZEAP assembler and fixed to operate from location 3000H. The program in ZEAP could not be printed out because there was no serial printer available for use with the Nascom computer. Hard copies of the program were printed on the Epson FX80 parallel printer connected through port PL4 on the Nascom microcomputer, and driven by a printer routine program PRLLPRNT, which is listed in appendix C1. A complete list of the system programs for the acquisition instrument are shown in appendix C2.

4.2.1 Program design

The main program is broken into subroutines which are easily incorporated in application programs. A workspace is set aside for the main program's temporary data storage area. All the fixed data in the ROM is transferred to this area on initialization.

In designing the program the following considerations were taken:-

1. Acquisition of input signals and data.
2. Generation and conversion of output signals and data.

3. Memory allocation for main monitor program.

4. System initialization and constants in the program.

5. Timing sequences.

6. Memory allocation of data.

7. Memory allocation of systems workspace.

8. Length and precision of data.

9. External interrupt devices.

10. Real time clock control.

11. Nascom system capabilities. 12. User interaction.

4.3 Development Language

The software was developed using Z80 assembly language mnemonics, because the development system (The Nascom computer) uses a Z80 microprocessor and is limited in its high level language capabilities.

Program development is usually done on a full development system capable of fully emulating the system being designed, such as the Ice box by Micrologics UK Ltd,[30] but at the time this project was proposed none was available in the University. The Department of Physics however has since acquired the ICE box. The programs were blown into two 2716 EPROMs using an EPROM programmer. For the sake of clarity the programs are classified into two main areas:-

1. Main monitor program.
2. Applications programs.

4.4 Monitor Programs

The monitor program is the main system program. It initializes the system, checks the memory RAMs, scans the key board for any inputs, displays information on the seven segment display, detects errors resulting from key board entries and provides program editing and execution capabilities. All of the following subroutines associated with the monitor program namely; INTLZE, LOOK, DSPLY, SOUND, CONVRT are described in the following sections. All the above subroutines are in the program listing in Appendix C3.

4.4.1 Initializing

The system is initialized by clearing the RAM, transferring program variable data into the temporary work space from the ROM data area, setting the system stack and initializing the input/output ports as well as configuring the 8255 programmable I/O device for either input or output operations. All these functions are implemented in subroutine INTLZE. This subroutine like most of the others can work independently . It can therefore be called by a subroutine CALL statement by a user who wishes to develop his own application programs for the acquisition system which is independent of the monitor program. The initialization of the program fixes Four main program functions; Beginning of stack, the audio cue frequency, the acquisition operation mode,the time interval for taking measurements, and configures the input/output operating modes. These fixed functions specify the default parameters used by the monitor program. The same parameters can be altered by a user applications program to meet the users unique measurement specifications.

4.4.1.1 The Stack

Only one stack is supported in the Z80 hardware even though more stacks can be simulated in the software. However software stacks have the disadvantage of increasing the complexity of the program and utilizing a lot of memory space [22]. For the above reasons only one stack has been used in this program. It has been employed for the following functions :-

1. To store the program environment during the real time clock interrupt processing.

2. Temporary storage of CPU registers in both the monitor and application program.
3. For saving the environment before transferring to or from subroutines.
4. Transfer of data between CPU and registers.

Subroutine INTLZE sets the stack to begin at location 0FFFH and reserves 500 bytes for this purpose. The growth of the stack is not limited to this dimension and therefore the user has to limit it if any additional application programs are written to avoid it growing into the systems workspace. The stack register has also been used to facilitate the processing of strings of data acquired. When using the stack for this purpose no maskable or non maskable interrupts are permitted and no other subroutines that utilize the stack may be employed during the time of processing[23]. The program structure which was adopted for processing using the stack is shown below:-

```
LD   (SAVP),SP                    ;Save current stack pointer.
LD   SP, (DATA_ADDRESS)           ;Initialize SP to data.
POP BC                            ;First byte in C, and next in B.
  ----------
(PROCESSES DATA)                  ;Process here.
  ----------
LD SP,(SAVP)                      ; Restore SP to original stack.
```

4.4.2 Audio Cues

Audio cues for this system tell the user some things that cannot be observed on the display unit, but may have taken place in the system. Cues are employed in this work to alert the user, of a task already accomplished by the microprocessor such as end of data transfer, errors in keypad entries. Cues have also been used to indicate a response to key inputs thus confirming that the depressed key has been detected. Subroutine SOUND is used to generate the audio output at a frequency of 1KHz for 30 milliseconds. The time duration of the cue can be altered by changing its value at location 2030H after the system is initialized. SOUND outputs a square wave signal through output port 11H on the 8255 to an external speaker.

4.4.3 Programming the Peripheral interface Adapter

The 8255 ports are utilized in this design to make the system more versatile. Subroutine INTLZE programs the 8255s to configure them as either input or output ports. The control word is issued to ports 13H and 17H respectively for both the 8255 chips. A summary of the 8255 port configuration is shown in figure 4.4.3.1.

The 8255 can be set to operate in any of the following three modes:-

1. Mode 0 - Basic Input/Output.
2. Mode 1 - Strobed Input/Output.
3. Mode 2 - Bi-directional Bus

INTLZE programs the 8255 to operate in mode 0. But since it is programmable the users may define their own routines to reprogram the chip for different configurations.

PORT ADDRESS	PORT	8255 IC1
10	A	KEY PAD
11	B	KEY PAD/DISPLAY
12	C	DISPLAY
13		CONTROL
		8255 IC2
14	A	UART
15	B	PRINTER
16	C	-
17		CONTROL

Table 4.4.3.1 8255 Port Addresses and Functions

Table 4.4.3.2 summarizes the mode 0 port definition. The appropriate control word sent to ports 13H and 17H is deduced from this table [24].

The control word number six (see table 4.4.3.1) issued to port 13H configures the first 8255 chip to operate with one input port and two output ports. Port A of this chip whose address is 10H becomes an input, port B whose address is 11H becomes an output and port C whose address is 12H in this instance also becomes an output. Similarly the second 8255 is programmed by issuing control word number 5 to port 17H which configures it as follows; port A whose address is 14H becomes an output, port B whose address is 15H also becomes an output and port c whose address is 16H becomes an input. Each of these ports has its associated subroutines for the

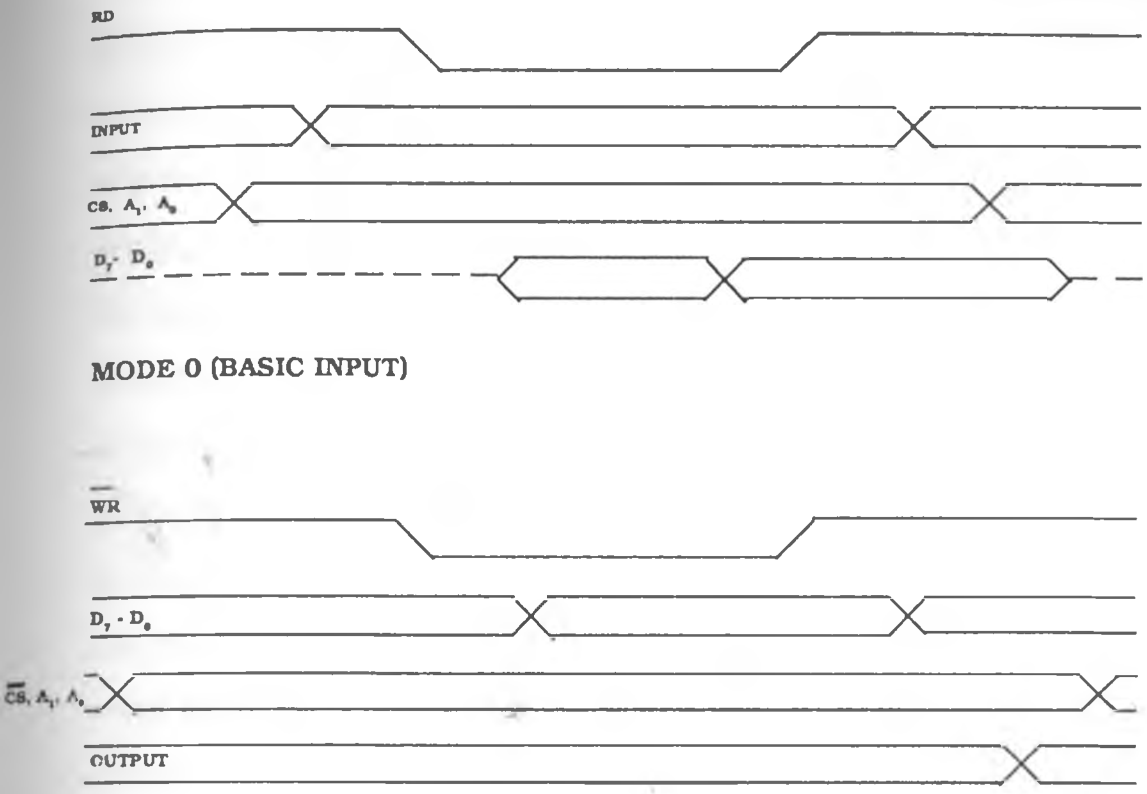
A		B		GROUP A			GROUP B	
D3	D2	D1	D0	PORT A	PORT C (UPPER)	NO.	PORT C (LOWER)	PORT B
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	INPUT	OUTPUT
0	0	1	0	OUTPUT	OUTPUT	2	OUTPUT	INPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	INPUT	OUTPUT
0	1	1	0	OUTPUT	INPUT	6	OUTPUT	INPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	INPUT	OUTPUT
1	0	1	0	INPUT	OUTPUT	10	OUTPUT	INPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	INPUT	OUTPUT
1	1	1	0	INPUT	INPUT	14	OUTPUT	INPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

Table 4.4.3.2. Mode 0 Port Defination

specified function. The timing diagrams for mode 0 operations are shown in fig 4.4.3.1.

4.4.4 System Control Mode and Data Acquisition Time Interval

The system is designed to operate in two modes described in section 3.8 of chapter three. The Normal mode requires few parameters to be initialized in the program. To operate the system in this mode



MODE 0 (BASIC INPUT)

MODE 0 (BASIC OUTPUT)

Fig. 4.4.3.1 Mode 0 Basic I/O timing diagrams

program DATAINN and it's associated subroutine MEMO described in section 4.6 of this chapter are executed and then the system acquires data automatically. The default time interval between consecutive readings is 60 minutes. This time interval is set on initializing the system and may be altered at location 203AH by specifying the desired interval between measurements before executing the acquisition program DATINI in this particular mode.

If the interrupt method is desired for data acquisition the user has to specify the time at which a reading is to start and the interval required between readings. This is done by altering the default settings at locations 2040H 2041H and 2042H to set a new time for the commencement of measurement and location 2043H to 2046H to set the time interval between the measurements in hours, minutes, seconds and hundredths of seconds respectively. The default value for the time interval in this mode is 1 second. The reading commences as soon as the program is initiated. When this program is executed, the default values for time intervals and measurement commencement time are set. The first interrupt will then be issued at the programmed time of commencement to invoke measurement. The advantage of operating the system in this mode is that while not taking measurements it can be utilized to perform other functions like data transfer to the printer or to some external storage device. In effect the interrupt method involves programming the real time clock.

4.5 Keypad Scanning Program

Keypad scanning is implemented by the LOOK subroutine. On executing LOOK, a counter monitoring the number of times the keypad matrix columns and rows are scanned is set to zero. The scanning proceeds by progressively issuing a voltage to each of the rows of the matrix connected to port 10H beginning from D0. The columns of the matrix connected to port 11H are then scanned for any voltage coincidences. The first word to be issued to port 10H on scanning is 01H (binary 00000001). columns D0 to D7 are then examined in sequence. If a key is depressed the corresponding column will have a voltage if it also is in the first column. This key is identified, it's internal code is stored in the systems workspace, and the correct display pattern shown at the desired position on the display unit. After the first scan the scanning counter is incremented and a voltage is issued to the second row in a similar manner. The columns are then scanned again for any voltages. This process continues till all the rows are completed after which it is repeated. If no key is found depressed the counter value in the program is also incremented and the scanning done again until a depressed key is encountered. The counter is used to store the position code of the key depressed and it is reset once the scanning goes through one complete cycle.

4.5.1 Scanning Period and Key Bounce

The keypad is usually depressed by hand to enter data or issue a command e.g. run a program or interrupt a program. As already described in the previous section, the microprocessor scans the keypad continuously to find any keys depressed. When a key is depressed and released it bounces for some time, so that the voltage on the key matrix column oscillates between 0 and 5 volts for a short period thus creating the same effect as if the key is depressed in succession. A typical time response diagram for a depressed key is shown in figure 4.5.1.1. To avoid the key bounce problem the rate

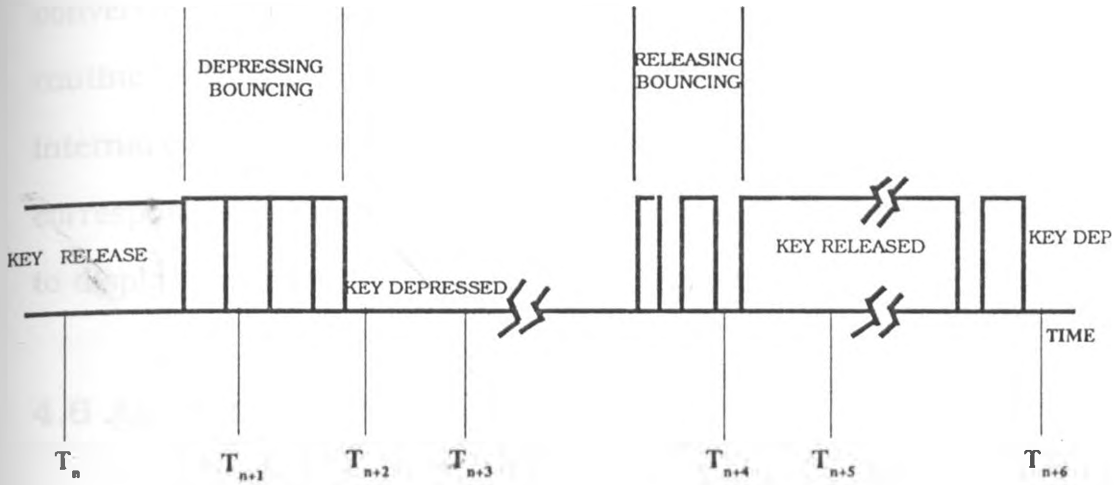


Fig. 4.6.1.1. Keypad Scanning Time Response[25].

of scanning the keypad has been selected taking into account the bounce period. The bounce time for most keys does not exceed 10 milliseconds, therefore the period of scanning the keypad is set to about 20 milliseconds. This is sufficient enough to ensure an error free scan. The program also takes into consideration multiple key depressions. Any multiple key depressions will return a key code that is none existent such a code will be ignored and the program LOOK searches the keypad again for a single depressed key.

4.5.2 Hexadecimal to Seven Segment Conversion

The format and method of displaying data and information visually on the display unit has already been discussed. Information is converted into the seven segment display format by the conversion routine CONVRT. When a key is detected during one LOOK cycle the internal code for the hexadecimal value of the key is converted to the corresponding seven segment display format. DISPLY is then called to display the key value on the seven segment display.

4.6 Application Programs

The application programs are the actual programs that determine the systems function. The main application program is composed of several subroutines and service programs to facilitate the processes of acquiring data, storing it in the battery backed RAMs and ultimately transferring it to other computer systems for analysis. The process of acquiring data is performed by two application programs DATAINN and DATAINI together with subroutine MEMO for storing data into memory.

DATAINN reads the real time clock register and the analog to digital converter inputs thus recording the measurements. The index register IX is used to point to the storage memory location. Time is stored in the first four consecutive locations beginning with that pointed by

IX and the corresponding data acquired from the digital to analog converter is stored in the next 16 consecutive locations ie. IX+4 for channel 1 to IX+19 for the last channel (channel 16). The storage memory location is then incremented by 15H and the next set of readings are taken. In the interrupt mode however the time indicated is the actual time of the day recorded from the real time clock registers at the instance the measurement is made. Three memory locations to store the time are used when acquiring data in this mode. The seconds value is stored in the location pointed by IX, the minutes and hours will follow in the next two locations, then comes the data in the next 16 locations making a total of 19 readings for each cycle of measurements. The routine for taking readings at the actual time of the day is RAMDATINI which is an extension of DATAINI, and is used in conjunction with the corresponding ram storage routine RAM-MEMO. The mode of operation of acquiring data is determined by the user. Both the normal and the interrupt modes can both be used with little adjustments in the software. This provision is not considered since only a few experiments may require a combination of both methods to acquire data.

4.6.1 BCD to ASCII Conversion

Data is acquired and stored in binary and has to be converted to a suitable format for down loading to a larger computer system for analysis. Conversions are done by subroutine BASCII which converts eight bit binary presentations into ASCII representation. Numbers larger than a byte are stored in consecutive locations in memory and are preceded with an ASCII word for a carriage return to avoid ambiguity when transmitting.

4.7 Data Retrieval and Serial I/O

Most data is transferred to peripherals:- the printer and magnetic tape via the 8255 parallel ports. Port 15H is used in the transfer of data to a parallel printer by utilizing the user control functions. Data transfer to and from a tape is channelled through the same port as the printer by the program TAPEWR for writing to the tape and TAPERD for reading from a tape respectively. Subroutine PRINTER controls the output to the printer. This routine continually outputs data from a specified memory area to the printer. Serial communications are made through the 6402 Universal Asynchronous Receiver and Transmitter (UART) and is controlled by the service routine SERL.

4.7.1 Serial Data Communication Software

Serial communication software in this project has been adopted from earlier work done by Shiyukah[26]. In his work, the British Broadcasting Corporation (BBC) microcomputer system was interfaced to the Microprofessor (MPF) single board computer. The interfacing consists of two wires to interconnect the two systems. The MPF PIO is used for down loading data to the BBC through the RS 232 serial port. Software was written for the BBC to receive the incoming data and for the MPF to transmit it at the same baud rate as received. Some modifications were made on Shiyukah's programs for serial communication and used for the data acquisition system. The resulting program is entitled SERL in this work.

In the câse of Shiyukah the MPF continually outputs data to the BBC. It is the easiest way of writing a program for serial communications, however there is a likelihood of missing out some data if the BBC is switched on after the MPF has began it's transmission. Therefore, in designing the down loading program of the acquisition system a different protocol is used for communication. Two signals are required for handshaking; the ready to send and the ready to transmit signals, which are used in addition to the existing basic communication lines ie. the signal (data) transmit and Ground.

The microprocessor on the acquisition system reads the Transmitter Buffer Register Empty status (TBRE) of the UART to inquire if it is ready to receive data for transmission. If TBRE is high, then a byte of data is transferred to the UART from memory for transmission to external devices.' The transmitter Buffer Register Load (TBRL) is forced low by the ready to receive line on the BBC. The UART at this point transmits the data in its registers to the BBC. For simplicity in software, the framing error, overrun error and data received error on the 6402 chip are disabled. The transmitted word has an additional stop bit but no parity bit. Transmission is done at a baud rate of 2400 [27]. After receiving a byte of data and storing it into memory, the BBC sends a handshaking signal again to indicate that it is ready to receive another byte of data. The ready to receive data line is connected to TBRL line of the UART. when TBRL goes low, data is transferred to the transmitter buffer register of the 6402 for transmission to the BBC. A timing diagram for the control of the UART is shown in fig 4.7.1.1.

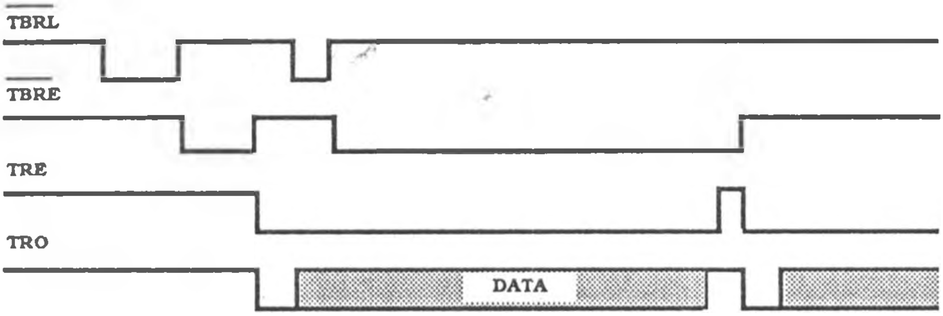


Fig. 4.7.1.1. UART Timing Diagram

4.7.2 The Printer Subroutine

Data in BCD format is dumped into a printer using the PRINTER routine. The beginning and ending of the data block to be printed are specified at locations 2031H and 2034H respectively. The program is then executed at location 34B2H which invokes PRINTER to send the data byte by byte to the printer connected to port 15H. The number of bytes to be printed on a single line can be controlled by the line size in location 2035H. The default is 20 bytes per line.

4.8 Summary

The monitor and application programs provide the basis of the acquisition processes. The performance of the system is dependent on the efficiency of the developed software. In the next chapter we will discuss the resulting integrated system which marries the developed software and hardware.

Chapter five

System testing and Evaluation

5.1 Testing and Evaluation Overview

No instrument can be regarded functionally useful until it has undergone the appropriate tests and evaluation procedures to ascertain its' performance and capabilities thus confirming its' usefulness. Any method of testing can be employed, provided the parameters and functions to be evaluated are clearly specified. Two methods of testing have been employed in this work to evaluate the integrated system and to test it's performance.

The first method utilizes an automatic test equipment (A.T.E). An A.T.E can be employed at it's lowest level of sophistication to carry out simple GO/NO GO type of tests[29]. In this simple mode of A.T.E operation, no attempt is made to determine the actual values of variables involved in the operation of the system. Even though this mode of employing the A.T.E is simple it is nonetheless powerful enough to identify and isolate circuit malfunctions, especially those resulting from poor fabrication and faulty designs. At a higher level of sophistication, A.T.Es may be used to carry out a sequence of patterns to check the correct operation of test units under different operating conditions. The Icebox emulator was used as the A.T.E to test the integrated system.

A second method involved the designing of a simple experiment to simulate a data acquisition environment. This simple experimental scenario proved to be a reliable method of testing the actual performance of the integrated system in the "real world". Both methods are described in the preceding sections.

5.2 The ICE box Emulator

5.2.1 General overview

The Ice box emulator utilizes two microprocessors in its design; An internal control processor which controls all the emulator hardware e.g. Keyboard, display, serial communication etc., and an emulation processor which resides on the emulation interface board and whose sole function is that of emulation. This use of two processors has several advantages, the main one being that the emulator is protected from any faulty target hardware hence allowing diagnostics to be performed. Secondly, the emulator CPU can be as close as possible to target system thus providing the best possible emulation[30]. Icebox can function in four modes:-

1. Local/stand alone mode, using the in-built keyboard and hexadecimal display.

2. Terminal mode, by connecting it (Icebox) to a standard serial terminal via the RS232 port.

3. Computer mode intended for use when Icebox is connected to a host computer via the RS232 link.

4. Automatic test equipment (A.T.E) mode, in which Icebox operates as an automatic programmable test equipment.

5.2.2 In Circuit tests using ICEBOX

All the three boards designed were separately tested using the Icebox emulator in both the computer and A.T.E mode. The IBM personal system/2 computer was used as the host system. Figure 5.2.2.1 shows the block diagram of the complete set up of the equipment for circuit tests.

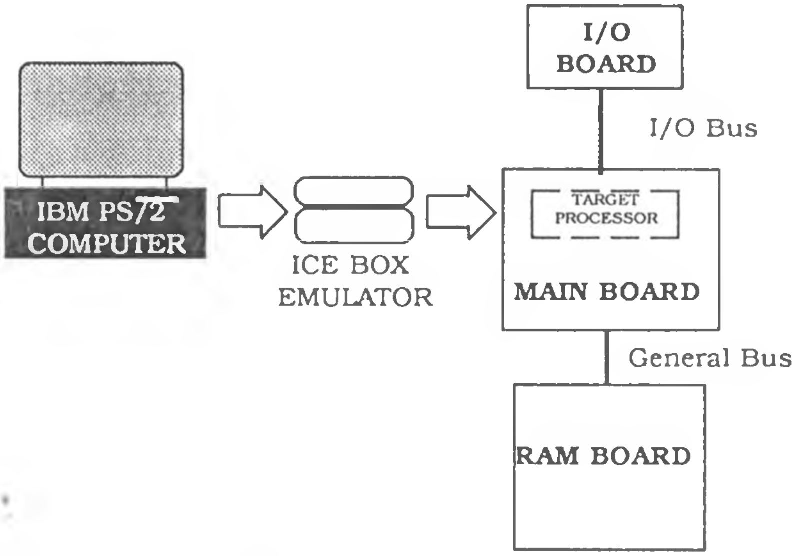


Fig. 5.2.2.1.ICE BOX In-Circuit Tests Block diagram.

The arget processor used on the icebox emulator interface was the Z80. This was then plugged into the mother board using the emulation header dual in line (D.I.L) plug,'Which is connected directly to the processor on the interface to be tested, except for it's data lines which are buffered. Simple diagnostics were performed on the main board to verify three main functions of the design.

1. Memory and input output decoding.
2. Monitor program performance.
3. Real time clock performance.

5.2.2.1 Memmory and I/O Decoding

Icebox was used to check and verify that the memory and input/ output decoding on the main board had no hardware errors resulting from PCB fabrication. This was done by accessing each of the decoded memory chips as well as input/output ports thus verifying whether the memory is a RAM or ROM, according to the designed memory maps. Input/output performance testing was achieved in a similar manner. A small program was used to write random data into all the RAM's to evaluate their performance. The memory contents of the RAMs was then verified by displaying it on the VDU.

5.2.2.2 Monitor Program Verification

The monitor program was verified by the dis-assembly utility which directly dis-assembled the program to the terminal. The dis-assembled output consists of address, data, and mnemonics operand. All hexadecimal values were issued with leading zeros. A file was also created from the ROM readings and compared with that created from the EPROM programmer. Various subroutines of the monitor program were then executed and the output ports monitored for the correct output.

5.2.2.3 Real Time Clock Performance

The RAM registers of the real time clock was tested in the same way as that of the main board. The RAMs were accessed from the host-computer system and programmed to issue interrupts at 60 second intervals. The service routine associated with the real time clock in-the monitor program was executed and the interval at which the interrupts were issued was monitored by an oscilloscope.

5.3 Data Logging tests

A rather crude but simple experiment was devised to test the performance of the data acquisition system. No specific transducer was employed in this test, instead direct measurements of voltages from an array of sixteen resistors was monitored. An advantage of using voltage measurements is that ultimately all physical parameters to be measured are converted to voltage values using appropriate transducers.

5.3.1 The Experimental Set-up

Fig 5.3.1.1 shows a diagram of the experiment set up to test the systems performance in acquiring data. Sixteen readings are taken from varying voltages across resistors representing the different readings which can be obtained from a set of transducers. The current in each of the resistors was individually varied by adjusting the potentiometer connected in series with each respective resistor. The voltage across the whole resistor network was varied by a single potentiometer connected to the whole network. The data acquisition instrument analog inputs were connected to the array of sixteen resistors in the network. Readings were then obtained both manually

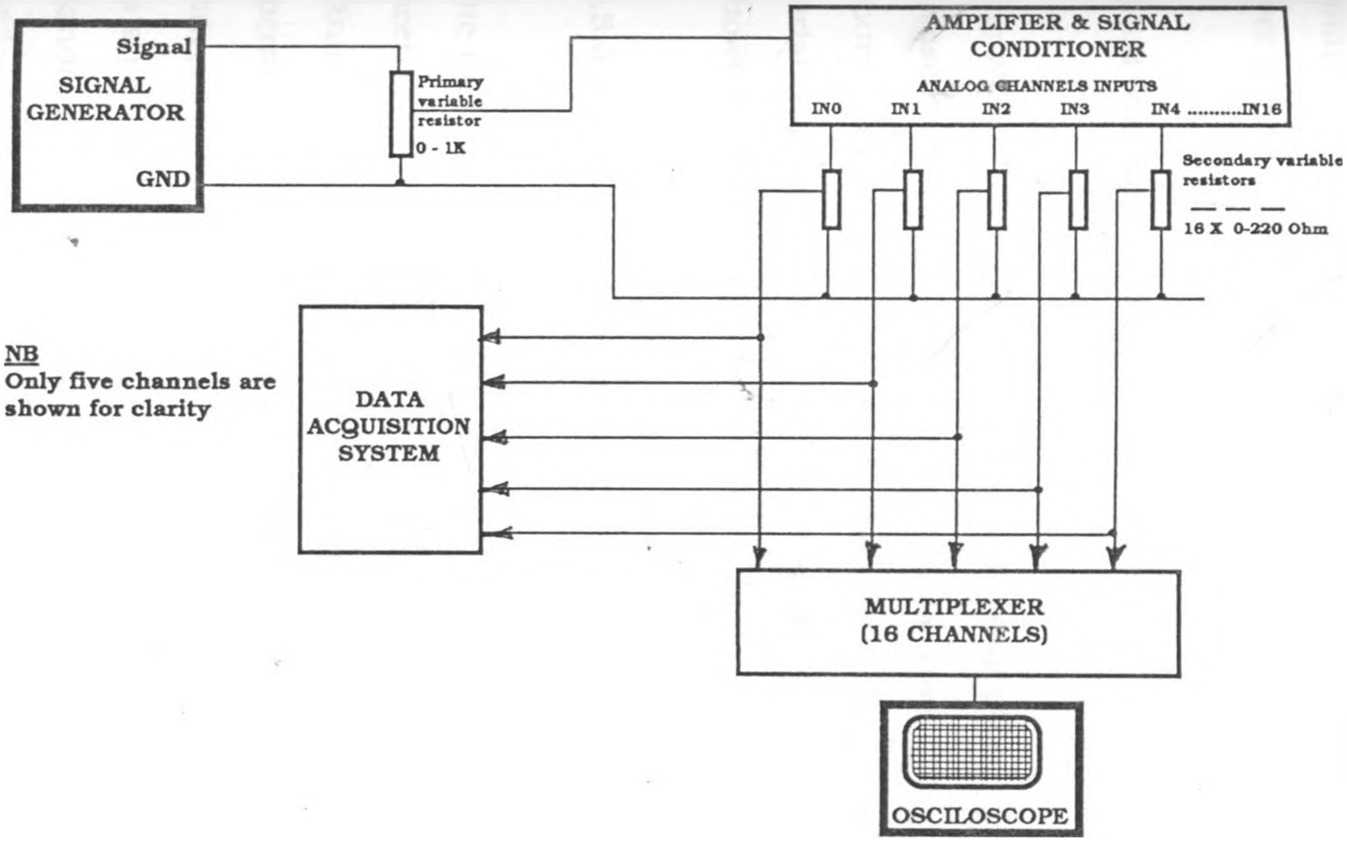

SIGNAL GENERATOR
Signal
GND
Primary variable resistor
0 - 1K
AMPLIFIER & SIGNAL CONDITIONER
ANALOG CHANNELS INPUTS
IN0
IN1
IN2
IN3
IN4 ....IN16
Secondary variable resistors
16 X 0-220 Ohm
NB
Only five channels are shown for clarity
DATA ACQUISITION SYSTEM
MULTIPLEXER (16 CHANNELS)
OSCILOSCOPE

Fig. 5.3.1.1 Data acquisition test experiment

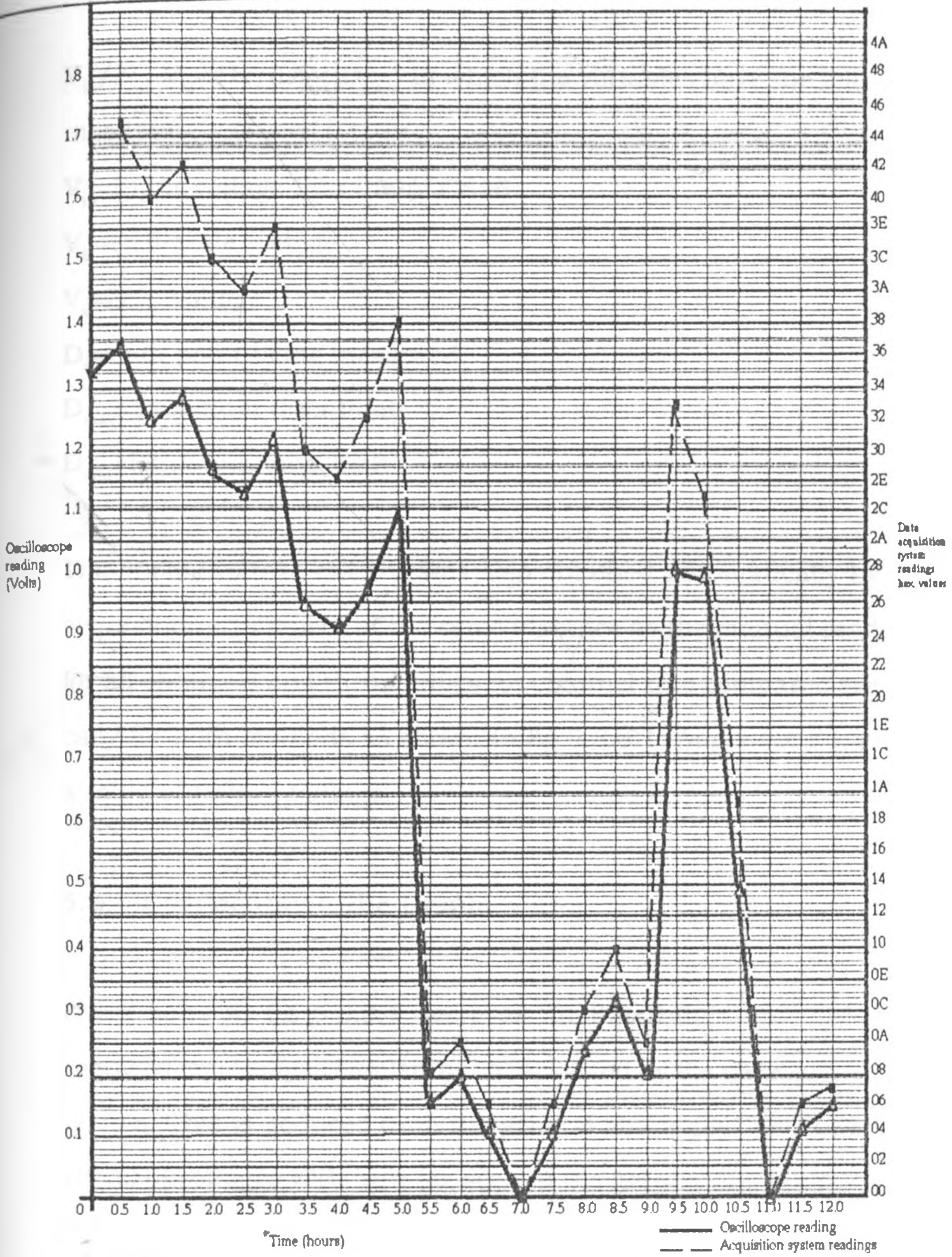
and automatically through measuring the voltage across each resistor which was arbitrarily varied by adjusting the potentiometer to simulate changes in a physical parameter with time. This was done over a period of 12 hours.

5.3.2 Manual Readings

The voltage across the resistor network was taken manually using an oscilloscope. The list of readings obtained over the period of twelve hours for all channels is shown in appendix E1. A graph of the voltage variations (from some sample points)with time for channel zero is shown in fig 5.3.3.1.

5.3.3 Acquisition Systen Calibration and Readings

The calibration of the integrated system is done in the software to meet the requirement for the measurement of different physical parameters. During this test, the A/D converter was configured to operate in the ratiometric conversion mode[31]. In a ratiometric conversion system the physical variable being measured is expressed as a percentage of full scale not necessarily related to an absolute standard. The software calibration was not therefore necessary in this mode. The voltage input to the AD0816 was determined in the following equation:-



Channel o data

Time(Hr.)		0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0	10.5	11.0	11.5
Oscilloscope reading (Volts)±10 mV		1.33	1.35	1.25	1.29	1.17	1.13	.21	0.94	0.90	0.97	1.09	0.16	0.19	0.12	0.00	0.12	0.23	0.31	0.20	1.00	0.99	0.49	0.00	0.12
IPMDAS Reading	Hex	44	45	40	42	36	3A	3E	30	2E	32	38	08	0A	06	00	06	0C	10	0A	33	2D	19	00	06
	Decimal	68	69	64	66	54	58	62	48	46	50	56	8	10	6	0	6	12	16	10	51	45	25	0	6

Fig. 5.3.3.1. Voltage graph for Manual and Automated Readings

$V_{in}/(V_{fs}-V_{z}) = Dx/(D_{max}-D_{min})$ in which the variables have the following meaning:-

V_{in}=Input voltage into the AD0816

V_{fs}=Full-scale voltage

V_{z}=Zero voltage

D_{x}=data point being measured

D_{max}=Maximum data limit

D_{min}=Minimum data limit

The analog voltage reading was also obtained automatically by the acquisition equipment at the same instance as the manual readings for all channels and the readings for channel zero compared graphically for both methods as shown in fig 5.3.3.1. Appendix E1 shows a table of these readings.

5.3.4 Comments on the Graph

The graphs in fig 5.3.3.1 serve as an illustration that the data acquired manually is similar to that acquired by the data acquisition system. This is deduced from the general shape and identical variations of voltages with time (as seen in the graph). The resolution of the acquisition system is 20mv, and the manual readings taken from the centronix oscilloscope is 100mv, thus the readings are quite comparable. A further confirmation of the validity of the data is demonstrated in appendix E2.

$V_{in}/(V_{fs}-V_{z)}= Dx/(D_{max}-D_{min)}$ in which the variables have the following meaning:-

V_{in}=Input voltage into the AD0816

V_{fs}=Full-scale voltage

V_{z}=Zero voltage

D_{x}=data point being measured

D_{max}=Maximum data limit

D_{min}=Minimum data limit

The analog voltage reading was also obtained automatically by the acquisition equipment at the same instance as the manual readings for all channels and the readings for channel zero compared graphically for both methods as shown in fig 5.3.3.1. Appendix E1 shows a table of these readings.

5.3.4 Comments on the Graph

The graphs in fig 5.3.3.1 serve as an illustration that the data acquired manually is similar to that acquired by the data acquisition system. This is deduced from the general shape and identical variations of voltages with time (as seen in the graph). The resolution of the acquisition system is 20mv, and the manual readings taken from the centronix oscilloscope is 100mv, thus the readings are quite comparable. A further confirmation of the validity of the data is demonstrated in appendix E2.

5.3.5 Test of Transmission to the BBC Microcomputer

The data acquired from the experiment described in section 5.3 was transmitted to the BBC microcomputer to verify the capability of the system to transmit data to the BBC. The protocol of communication was described in chapter four. The data received by the BBC microcomputer shown in appendix E3 was identical to the one listed in appendix E2. Hence there was no apparent loss of data.

5.4 System Throughput

The throughput rate of a system is the rate at which a measurement can be taken, scaled to the equipments measuring units and the reading stored to final storage. We can generalize this by saying that the systems throughput is the rate at which data is transferred between different devices or components. The throughput for this data acquisition depends upon two major factors:-

1. The sample rate.
2. The amount of processing specified.

5.4.1 The Sample Rate

The fastest rate at which any measurement (sample rate) can be taken by an automatic equipment is limited by the transducer's response time, ie. the minimum time required for the transducer to record a change in the varying physical parameter. Digital equipment have a further limitation dependent upon the rate at which an analog signal can be converted to a digital reading. The A/D converter in this design is configured to operate at 1KHZ. At this speed, it takes approximately 200 micro-seconds to convert a single analog input signal into digital. The A/D converter therefore takes 3.2 milliseconds to scan 16 transducers. This is the minimum time needed to acquire data through all the 16 channels in one cycle of operation if the A/D converter is operating in the stand alone mode with no external program control or in the single instruction mode with multiple repetitions. For this system the rate of scanning the transducers is much lower because of the external control program required for the A/D converter.

5.4.2 The amount of processing Required

The primary factor affecting throughput is the amount of processing specified by the program for taking measurement. All processing called for by an instruction must be completed before moving on to the next instruction. The number of clock cycles in the processing program will therefore determine the actual throughput. using program DATAINN (which is the simplest in this system), the maximum throughput is 128 readings per second.

5.4.3 Current Consumption

The current consumption of the system was monitored during the system tests under laboratory conditions. The current supplied to the system was varied as the tests were carried and the mean value of the minimum current required to drive the system was found to be 0.354 Amp. at a 5V supply voltage.

5.5 Conclusion

Due to time constraints, not all modules developed and presented in chapters two and three are tested in the integrated system. The testing and possible debugging and these modules are left for future work. From the two tests of the performance of the system, we conclude that the system has demonstrated satisfactorily it's ability to acquire data, store it in memory and transmit it to another system for analysis.

Chapter 6

Conclusion and Future work

6.1 Hardware/Software partitioning

On examining the requirements of this project listed in chapter one section 1.5, the partitioning of the hardware and software has been carefully made to optimize the design of the system under discussion.

The hardware is kept to a minimum and most of the functions have been implemented in the software. The program is written to utilize this minimum hardware configuration without trading off the systems capabilities. This has resulted into an integrated system which is inexpensive, portable and powerful yet easy to maintain and handle both in the field and laboratory. The resulting system is also versatile and thus suitable for most scientific data acquisition. An inclination to software offers three main advantages namely; Flexibility, Development speed and Cost effectiveness.

6.2 Conclusion

In this thesis the development of a general purposes, inexpensive, portable, microprocessor based data acquisition system (IPMDAS) has been discussed and the designed presented, based on general requirements from two scientific areas of interest:-

1. Environmental monitoring and
2. Student laboratory experiments.

The equipment has been tested by simulating a data acquisition process in real time. It should be noted that this experiment was not meant to be an exhaustive treatment for the IPMDAS performance but rather a simple and sufficient test for evaluating its' usefulness. Further field tests need to be carried out for specific types of acquisition. The testing has proved that the system is capable of logging data at a considerable rate, and subsequently transferring the data to secondary storage or down loading the acquired data to the BBC microprocessor for analysis. The down loading feature is particularly useful because of the ease of manipulation of data in a high level language for analysis, and the use of already existing commercial packages such as the Instat and Lotus 1-2-3 for handling the row data statistically.

IPMDAS has a main advantage of hardware simplicity compared to imported data acquisition systems. The system's hardware maintenance is likely to involve the replacement of a faulty chip or transistor only, which can easily be identified in case of a failure.

A summary of IPMDAS advantages over standard fixed configuration system designs are as follows:-

1. A particular requirement of any application can be catered for.

2. The data logger configuration can be easily extended to cope with any increase in requirements thus no need of purchasing a new system.

3. maintenance of the system is easy and faults can easily be traced.

4. The design allows for enhancement to incorporate better modules.

5. The system can be calibrated and used for a wide variety of applications with suitable transducers.

Incorporation of a facility to enter and edit a program is a useful dimension in the programability of IPMDAS. With this facility, temporary user defined programs can easily be incorporated in the system to meet their own peculiar data acquisition requirements such-as calibration and software correction to experimental readings.

IPMDAS reflects an effort to design a cost effective programmable, easy to use, and powerful instrument for data acquisition.

6.3 Recomendation for Future Work

Due to time constraints the project has been developed up to the stage hitherto described. It is reasonable enough for use in the field and laboratory environments but still requires some improvements to make it more suitable for acquisition tasks.

We shall look at the future work required in two main areas namely, hardware and software. These recommendations are a reflection of the intended final IPMDAS design.

6.3.1 Hardware

IPMDAS can accommodate a large variety of transducers because of it's versatility. A provision is given for the sixteen multiplexed signals to be modified before the acquisition begins. Every transducer used in IPMDAS has to be evaluated to ascertain that it's output voltage level can be accommodated without the need of any modifications. Some signals will certainly require pre-processing such as Amplification, signal conditioning, linearization, filtering etc, before the acquisition. Standard modules to meet these pre-processing requirements from a selected list of frequently used transducers need to be designed. These modules will increase the scope of measurements which IPMDAS can take.

An interface can also be designed to expand the number of input channels of IPMDAS. The environmental monitoring experiments may require more than sixteen channels for analog inputs and will therefore utilize these analog input channels expansion board.

The hardware has not been tested for use in severe environmental conditions because we did not address ourselves to such issues of the environmental impact on the system in this thesis. Since the instrument is intended for use in areas where it is likely to encounter a harsh environment, further research is needed to evaluate IPMDAS performance under these field conditions.

6.3.2 Software

The hardware on IPMDAS must be minimized at all times when modifications are made, unless costs and other factors strongly suggest that the hardware ought to take a higher precedence over the software in partitioning. The elegance of IPMDAS lies in it's software, however the existing application programs for acquisition are fairly general. Application programs for more specific types of acquisition should be designed and incorporated in the system these programs can reside in the unused ROM areas of IPMDAS.

The two IPMDAS programs i.e DATAINN and DATAINI have been designed with the assumption that all sixteen channels for acquisition are always in use. All these channels are therefore continuously scanned for inputs, however it serves no useful purpose to scan channels that have no data inputs. These programs need to be modified to allow for the number of channels in use to be specified. Raw data often has very little meaning to the researcher. To be useful, this data must be linearlized and scaled with the aid of a calibration curve in order to determine the real value of the variables monitored in appropriate engineering units. To achieve this we recommend the development of suitable algorithms for calibration, reduction and simple manipulation of data.

There are no limits other than memory capacity in the development of programs for IPMDAS. Other application programs and software changes not recommended here can still be made because of the simplicity and generality in IPMDAS's hardware, to improve it's performance.

References

[1]. Campbell, D.T and Stanley, J,

Experimental & Quasi-Experimental Design. American Education Research, 1963, pp. 1-25.

[2]. Gordon, B,

Principles of Experimentation and Measurements. Prentice-Hall Inc., pp. 1-10.

[3]. Sharma, G.R, Rangan, C.S and Mani V.S.V,

Instrumentation Devices and Systems. Tata McGraw Hill Co. Ltd., 1983, pp. 1-7.

[4]. Cooper, W.D,

Electronic Instrumentation and Measurement Techniques. Prentice - Hall inc. 1978, pp. 450-453.

[5]. Fritschen, L.J and Lloyd W.G,

Environmental Instrumentation. Springer-Verlag, 1979, pp. 195 - 209.

[6]. Ibid, pp. 206 - 207.

[7]. Thompson, J.D and Drake, W.H,

Paleomagnetic spinner Magnetometer interface, Unpublished, 1980

[8]. Mate, P,

The Design of a Microprocessor Based Data Acquisition System and Instrument Control System for an Absorption calorimeter; University of Nairobi, MSc thesis, 1989.

[9]. Ball, S.K and Farouk, B,

Microprocessor based data acquisition system for thermo-fluids laboratory; Rev. of Sc. Inst. vol. 58 No.4, April 1987, pp. 657 - 659.

[10]. Martin, M.M and Cobb, C.L,

A Microcomputer Interface Board for the Resolved Multichannel Scaling; Rev. of sc. inst. Vol.58 No.8, August, 1987.

[11]. Squirrel Manual. Grant Instruments, 1982.

[12]. Campbell User manual, Campbell scientific inc., Logan,Utah. 1984.

[13]. Nascom 2 Microcomputer Documentation; Disassembler Manual, April, 1980. Nascom Microcomputers Division of

Lucas Logic Ltd.

[14]. Nicoll, J,

Parallel Printer Interface Driver Program, unpublished 1980.

[15]. Drake, W.H,

Low Cost Microprocessor Teaching System. ICTP Asian Regional College on Microprocessors. Colombo, Sri-Lanka 1985.

[15a]. Kirk, P.R,

Microprocessor Basics; Electronics Service, University of York, 1980.

[16]. Zaks, R,

From Chips to Systems: An Introduction to Microprocessors. Sybex inc., 1981, pp. 350 - 353.

[17]. Ibid. pp. 354

[18]. Namuye, S

A Low.Cost, Low Power Data Acquisition System; University of Nairobi Msc thesis, 1986, pp. 10b.

[19]. Data Acquisition Hand book, National Semiconductor Corp., 1978, pp. 3.

[20]. Ibid. pp. 2-29 - 2-38.

[21]. Intel Semi Conductor Inc., 7071 Real Time Clock Data Sheets 1989

[22]. Barden, W.jr.,

The Z-80 Microcomputer Hand Book. Howard W. Sams and co., Inc., 1978, pp. 154 - 155.

[23]. Ibid. pp 156.

[24]. Intel Semiconductor Inc., 8255A/8255A-5 Programmable Peripheral Interface Data Sheets.

[25]. Microprofessor User Manual, Multitech inc. 1986 pp. 38.

[26]. Shiyukah, D

Microprocessor Control of and Electronic Vehicle; University of Nairobi postgraduate Diploma in Computer Science Project report, 1987, pp. 27 - 40.

[27]. Ibib. pp 38

[28]. Smartwork User Manual; Wintek inc., 1984, pp. 1 - 5

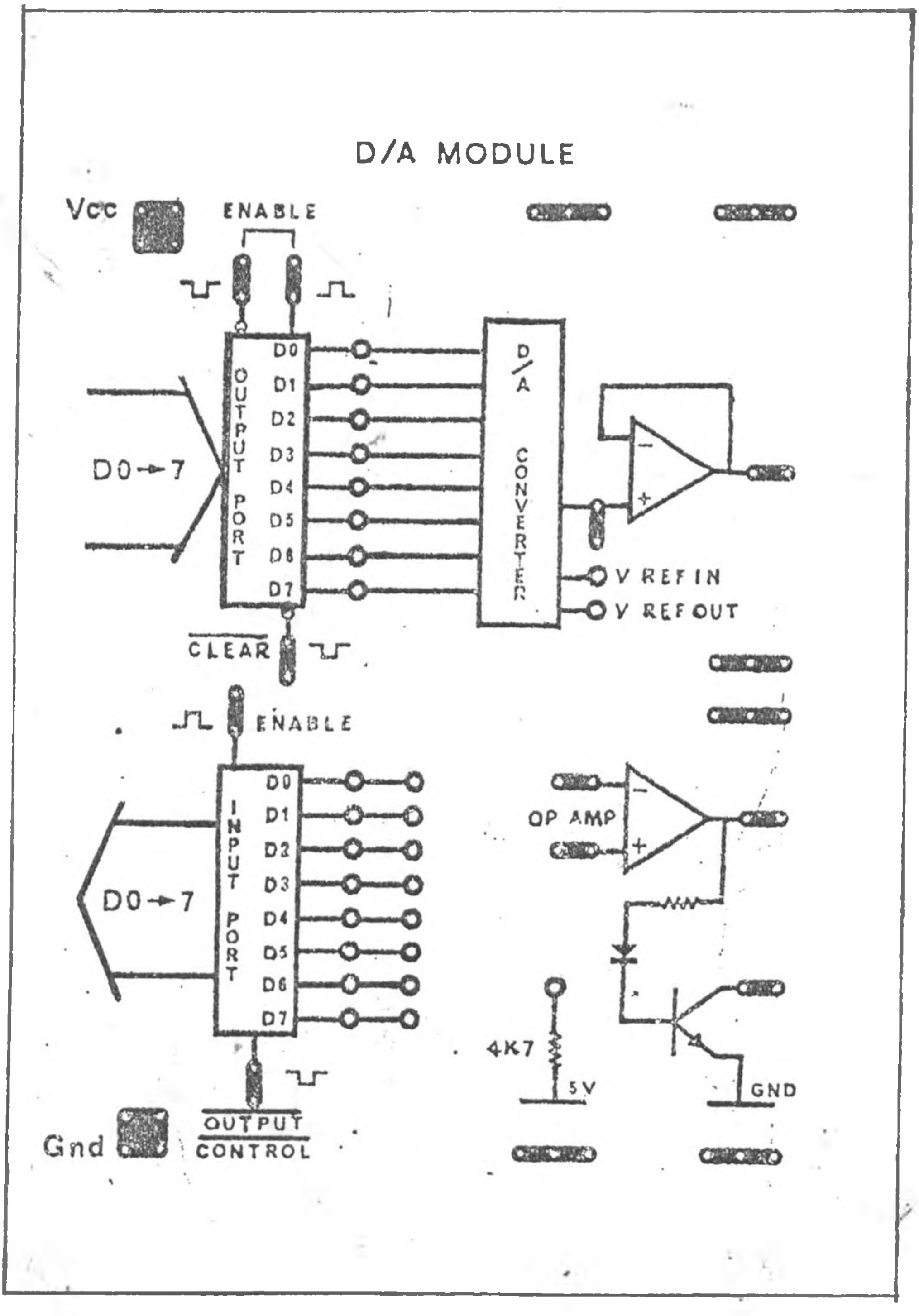
[29]. Reference [4] pp. 440-450 [30]. Icebox Operating instructions handbook; Noral Logics Ltd UK. 1984, Section 1.5

[31]. Reference [19], pp. 2-36 - 2-37.

[32]. Reference [18], pp. 17.

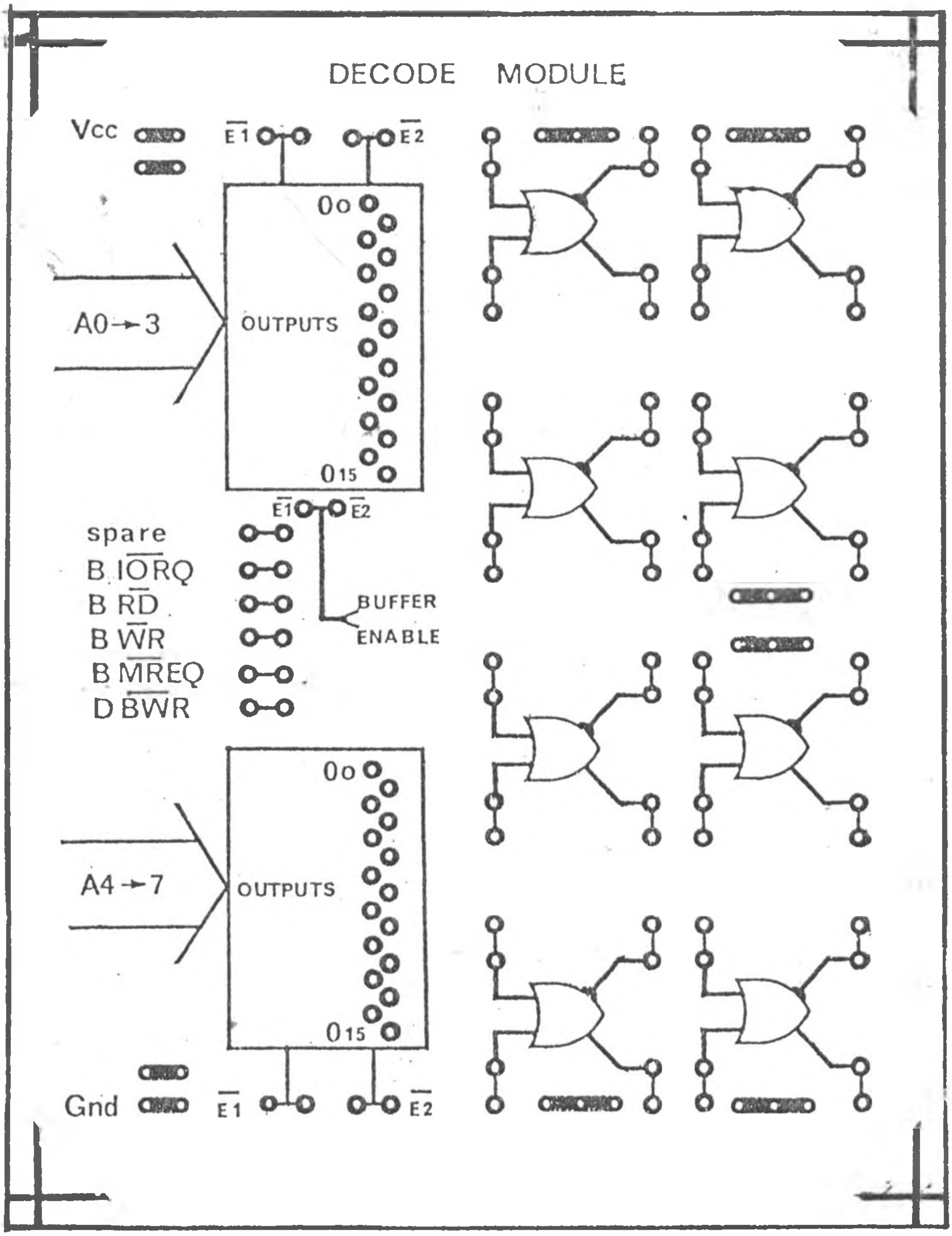
Appendix A1

Top view D/A Module



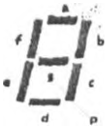
Appendix A2

Top View Decode Module



Appendix B1

Hexadecimal to Seven Segment Display Conversion table



7	6	5	4	3	2	1	0
d	p	c	b	a	f	g	e

DISPLAY FORMAT:

CODE	BD	30	9B	BA	36	AE	AF	38	BF	BE	3F	A7	8D	B3
DATA	0	1	2	3	4	5	6	7	8	9	A	B	C	D
DISP	0	1	2	3	4	5	6	7	8	9	A	b	C	d
CODE	8F	0F	AD	37	09	01	97	85	2B	23	A3	1F	3E	03
DATA	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
DISP	E	F	G	H	ī	J	K	L	ñ	n	o	P	q	r
CODE	A6	87	B5	B7	A9	07	B6	8A	83	A2	32	02	C0	00
DATA	S	T	U	V	W	X	Y	Z	(	)	+	−	,	
DISP	S	t	U	V	ū	X	Y	Ξ	c	ɔ	+	-	_.	

Appendix B3

Key Function Descreption

Numeric pad (0 - F)

The first four rows of the Keypad is the numeric pad it contains the hexadecimal numbers 1 to F and is used for entering the desired address, and data into a specified location or register.

Address/register key (Addr./reg.)

This key sets a memory address and is activated when the SHIFT is off. It must be followed by the specific address desired. SHIFT on activates the register key which sets a register name. The desired register is displayed on pressing the appropriate key.

Enter/register AF key (ent/AF)

Enter is activated with a SHIFT off and is used to confirm the set address and register the microprocessor responds by displaying the data contained in this address or register. When the SHIFT is on, the AF register key is activated it is used to access the data in register AF for any changes to be made.

GO/Register BC key

The Go key operates with the SHIFT off and it will commence program execution at the address shown on the display. The register BC key access the contents of this register.

Shift key

The shift key toggles all the keys between upper functions and lower functions except the numeric keys. The lower functions are activated when the shift is off and the upper functions are activated on a shift on its default value is off.

Increment/register DE (+/DE) key

This is activated with a shift off and increments the memory address or register by one. The contents of register DE can be examined with this key when the shift is on.

Decrement/HL (-/HL) key

Decrement key is active when the shift is off location by one. and it decrements the address or register when the shift is on the HL register contents can be modified.

Print/IX (P/IX) key

The print key allows the contents of memory to be printed and is active with a shift off contents of the IX register can be examined and modified by the IX key which is active when the shift is on.

Delete/register IY (del/IY).key

When the shift key is off this key deletes one byte from the memory. This key is also used to examine the content of register IY with the shift on.

appendix C1

Parallel Printer routine program listing

```
;NAS-SYS Parallel printer driver

PIOADAT    EQU  04              ;Data output port
PIOBDAT    EQU  05              ;Bit 0 is busy input
PIOACTL    EQU  06              ;Bit 1 is strobe
PIOBCTL    EQU  07
UOUT       EQU  OC77

BEGIN     LD   C,PIOCTL
LD   HL,OPTAB
CALL OUT4
INC  C                    ;c=  PIOBCTL
CALL OUT4                  ;Set up PIO
;Setup dump at UOUT to point to driver
LD   HL,PARAOUT
LD   (UOUT+1),A

;Set printer strobe HI

LD   A,OFFH
OUT  (PIOBDAT),A
RST  28H                   ;OOH ends print message
DEFM "PARALLEL PRINTER DRIVER DRIVER INSTALLED"
DEFB ODH, OAH              ;CR & LF
DEFM "TOGGLE WITH 'U' AND 'N' COMMANDS"
DEFB ODH, OAH, OOH
DEFB ODFH, 05BH             ;Return to NAS-SYS

;Printer driver proper is here

PAROUT     PUSH AF
OUT  (PIOADAT),A            ;Must be preserved
P1         IN   A,(PIOADAT)
BIT  O,A                   ;Look at printer busy
JR   NZ,PI                  ;Wait for no busy
LD   A,00
OUT  (PIOBDAT),A            ;Keep strobe low
P2         DEC  A
JR
```

```
DEC  A
OUT  (PIOBDAT,A           ;set srobe HI
POP  AF
RET
OUT4       LD   B,04
OUT4A      LD   A,(HL)        ;Output 4 bytes (HL) to (c)
OUT  (C),A
DJNZ OUT4A
RET
OPTAB      EQUB 0FFH,0          ;port a is mode 3 all outputs
EQUB 73H,07          ;Disable all interrupts
EQUB 0FFH,01         ;port B is mode 3 bit 1 out
EQUB 73H,07          ;bit 0 in
END
```

Appendix C3

Main Program Listing

```
 LOC   OBJ CODE  LINE   LABEL   SRCE CODE        COMMENTS

                 0001  ;===================================
                 0002  ;  SINGLE BOARD DATA ACQUISITION
                 0003  ;            SYSTEM
                 0004  ;  PROGRAM DEVELOPED ON THE NASCOM
                 0005  ;  UNIVERSTY OF NAIROBI
                 0006  ;  PHYSICS DEPARTMENT
                 0007  ;  1989
                 0008  ;  BY AGGREY MADAHANA
                 0009  ;  SURPERVISED BY Prof. W.H. DRAKE
                 0010  ; ==================================

0000   C30010    0011             JP   1000       ; START HERE
1000   CD9017    0012             CALL 179D       ;POWER ON BEEP
1003   D305      0013             OUT  (13H),05   ;SET 8255 IC1
1007   D306      0014             OUT  (17H),06   ;SET 8255 IC2
1008   C0030     0015             JP   3000       ; EXECUTE MONITOR PROGRAM
10D7   CDE010    0075   L10D7     CALL 10E0       ;KEY INPUT BEEP
10DA   CDCA12    0076             CALL 12CA
10DD   C9        0077             RET
10DE   00        0078             NOP             ;RESERVED FOR MAINTENANCE
10DF   00        0079             NOP             ;**********************************
10E0   E5        0080   L10E0     PUSH HL         ; Subroutine Delay time
10E1   D5        0081             PUSH DE         ;*********************************
10E2   211000    0082             LD   HL,0010
10E5   29        0083             ADD  HL,HL
10E6   110100    0084             LD   DE,0001
10E9   3E80      0085             LD   A,80       ;OUTPUT BEEP
10EB   D310      0086             OUT  A,80
10ED   41        0087             LD   B,C
10EE   10FE      0088   L10EE     DJNZ 10EE
10F0   EE80      0089             XOR  80         ;PIN 7 ONLY
10F2   ED52      0090             SBC  HL,DE
10F4   20F5      0091             JR   NZ,10EB
10F6   3E00      0092             LD   A,00
10F8   D310      0093             OUT  (10),A     ;CLEAR PORT
10FA   D1        0094             POP  DE
10FB   E1        0096             POP  HL
10FC   C9        0097             RET
10FD   00        0098             NOP               ;RESERVE FOR MAINTENCE
10FE   00        0099             NOP
10FF   00        0100             NOP
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE          COMMENTS
                      ;================================
                      ;Monitor program proper begins here
                      ;================================
3000  C32C32    0100         JP   322C      ;Goto startup location
3003  3E2A      0101         LD   A,2A      ;Delimeter routine
3005  3E2220    0102         LD   (2022),A
3008  3E2A      0103         LD   A,2A
300A  322320    0104         LD   (2023),A
300D  C37E31    0105         JP   317E      ;Execute PRINTER
3010  33        0106         INC  SP
3011  33        0107         INC  SP
3012  CDAC31    0108         CALL 31AC      ;Display data of break address
3015  C332 32   0109         JP   3232
3018  D9        0110         EXX
3019  08        0111         EX   AF,AF     ;Enter routine
301A  2A0045    0112         LD   HL,(4500)
301D  3A0245    0113         LD   A,(4502)
3020  77        0114         LD   (HL),A
3021  C33232    0115         JP   3232
3024  00        0116         NOP             ;USER DEFINED KEYS
3025  00        0117         NOP             ;CAN BE PROGRAMMED HERE
3026  00        0118         NOP
3027  00        0119         NOP
3028  00        0120         NOP
3029  00        0121         NOP
302A  00        0122         NOP
302B  00        0123         NOP
302C  00        0124         NOP
302D  00        0125         NOP
302E  00        0126         NOP
302F  00        0127         NOP
3030  0602      0128         LD   B,02
3032  CDED31    0129         CALL 31ED      ;Address routine
3035  C3E130    0130         JP   30E1
3038  00        0131         NOP
3039  00        0132         NOP
303A  00        0133         NOP
303B  CD1932    0134         CALL 3219      ;Ready prompt
303E  1E04      0135         LD   E,04
3040  FD212020  0136         LD IY,2020
3044  FD7300    0137         LD (IY),E
3047  CD3232    0138  L3047  CALL 3232
304A  F5        0139         PUSH AF         ;save data to be used in key
304B  C5        0140         PUSH BC         ;processing
304C  DD212120  0141         LD   IX,2021
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE              COMMENTS


3050  DD7E00     0142           LD   A,(IX)
3053  DD212420   0143           LD   IX,2024
3057  0602       0144           LD   B,02
3059  DD4E00     0145   L3059   LD   C,(IX)
305C  DD7700     0146           LD   (IX),A
305F  DD23       0147           INC  IX
3061  DD7E00     0148           LD   A,(IX)
3064  1809       0149           JR   306F
3066  08         0150           EX   AF,AF'
3067  D9         0151           EXX                       ;Real time clock
3068  CD6435     0154           CALL 3365                 ;interrupt
306B  D9         0155           EXX                       ;operates with
306C  08         0156           EX   AF,AF'               ;routine DATAINI
306D  ED45       0157           RETN
306F  DD71 00    0158   L306F   LD   (IX),C
3072  DD23       0159           INC  IX
3074  10E3       0160           DJNZ 3059
3076  DD212720   0161           LD   IX,2027
307A  C1         0162           POP  BC
307B  F1         0163           POP  AF
307C  18C9       0164           JR   3047
307E  D9         0165   L307E   EXX
307F  08         0166           EX   AF,AF'              ;Go routine
3080  E9         0167           JP   (HL)
3081  D9         0168           EXX                      ;Incr routine
3082  23         0169           INC  HL
3083  CD4F31     0170           CALL 314F
3086  D9         0171           EXX                      ;Decr routine
3087  C33232     0172           JP   3232
308A  D9         0173           EXX
308B  2B         0174           DEC  HL
308C  CD4F31     0175           CALL 314F      ;Scan the key board 1 cycle
308F  D9         0176           EXX                       ;Brk pnt routine4
3090  C33232     0177           JP   3232
3093  D9         0178           EXX
3094  08         0179           EX   AF,AF'
3095  220045     0180           LD   (4500),HL
3098  7E         0181           LD   A,(HL)
3099  320245     0182           LD   (4502),A
309C  3EDF       0183           LD   A,DFH                ;set restart loc
309E  77         0184           LD   (HL),A
309F  C37E30     0185           JP   307E
30A2  00         0186           NOP                       ;user key vector
30A3  00         0187           NOP                       ;location
30A4  00         0188           NOP
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE              COMMENTS

30A5  00         0189           NOP
30A6  00         0190           NOP
30A7  00         0191           NOP
30A8  3A0038     0192   L30A8   LD   A,(3800)         ;Shift key
30AB  FE34       0193           CP   34               ;routine
30AD  201B       0194           JR   30CA
30AF  3E00       0195           LD   A,00
30B1  D314       0196           OUT  (14),A
30B3  3E30       0197           LD   A,30
30B5  320038     0198           LD   (3800),A
30B8  210023     0199           LD   HL,2300
30BB  CD5236     0200           CALL 3652             ;V table
30BE  3ECB       0201           LD   A,CB
30C0  320039     0202           LD   (3900),A
30C3  3E3B       0203           LD   A,3B
30C5  322920     0204           LD   (2029),A
30C8  1814       0205           JR   30DE
30CA  3E01       0206   L30CA   LD   A,01
30CC  D314       0207           OUT  (14),A
30CE  3E34       0208           LD   A,34
30D0  320038     0209           LD   (3800),A
30D3  3E00       0210           LD   A,00
30D5  322920     0211           LD   (2029),A
30D8  211023     0212           LD   HL,2310
30DB  CD5236     0213           CALL 3652
30DE  C37336     0214   L310E   JP   3673
30E1  CD3132     0215   L31E1   CALL 3131
30E4  4F         0216           LD   C,A
30E5  1E04       0217           LD   E,04
30E7  78         0218           LD   A,B
30E8  FE00       0219           CP   00
30EA  280A       0220           JR   Z,30F6
30EC  CB21       0221   L30EC   SLA  C
30EE  1D         0222           DEC  E
30EF  20FB       0223           JR   NZ,31EC
30F1  CD3131     0224           CALL 3131
30F4  1818       0225           JR   310E
30F6  CB26       0226   L31F6   SLA  (HL)
30F8  1D         0227           DEC  E
30F9  20FB       0228           JR   NZ,30F6
30FB  79         0229           LD   A,C
30FC  86         0230           ADD  A,(HL)
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE           COMMENTS


3101  3D         0232           DEC  A
3102  322A20     0233           LD   (202A),A
3105  FE01       0235           CP   01
3107  281D       0236           JR   Z,3126
3109  CD4F31     0237           CALL 314F
310C  1821       0238           JR   312F
310E  81         0239    L310E  ADD  A,C        ;subrotine decode
310F  4F         0240           LD   C,A
3110  7B         0241           LD   A,B
3111  FE00       0242           CP   00
3113  2811       0243           JR   Z,3126
3115  69         0244           LD   L,C
3116  FE01       0245           CP   01
3118  2809       0246           JR   Z,3123
311A  1E08       0247           LD   E,08       ;loop 8 times
311C  CB25       0248   L311C   SLA  L
311E  CB14       0249           RL   H
3120  1D         0250           DEC  E
3121  20F9       0251           JR   NZ,311C
3123  05         0252   L3123   DEC  B
3124  18BB       0253           JR   30E1
3126  23         0254   L3126   INC  HL
3127  3E03       0255           LD   A,03
3129  322A20     0256           LD   (202A),A
312C  CD4F31     0257           CALL 314F
312F  18B0       0258   L312F   JR   30E1
3131  D9         0259   L3131   EXX
3132  CD3731     0260           CALL 3137
3135  D9         0261           EXX
3136  C9         0262           RET
3137  ED4600     0263   L3137   LD   B,(IY)
313A  78         0264           LD   A,B
313B  FE00       0265           CP   00
313D  2002       0266           JR   NZ,3141
313F  180A       0267           JR   314B
3141  DD7E00     0268   L3141   LD   A,(IX)
3144  DD2B       0269           DEC  IX
3146  05         0270           DEC  B
3147  FD7000     0271           LD   (IY),B
314A  C9         0272           RET             ;*****************************
314B  CD3232     0273   L314B   CALL  3232      ;subroutine LOOK1
314E  C9         0274           RET             ;& SCAN1 begin here
314F  D5         0275   L314F   PUSH DE         ;******************************
3150  C5         0276           PUSH  BC        ;scans the keyboard and the
3151  F5         0277           PUSH  AF        ; &the display system
```

```
LOC   OBJ CODE  LABEL   SRCE CODE             COMMENTS

3152  112220            LD    DE,2022   ;routine scan  does  not  return
3155  7E                LD    A,(HL)    ;a key is pressed
3156  E60F              AND   0F
3158  12                LD    (DE),A
3159  4E                LD    C,(HL)
315A  CD7131            CALL  3172      ;Delay
315D  7D                LD    A,L
315E  E60F              AND   0F
3160  12                LD    (DE),A
3161  4D                LD    C,L
3162  CD7131            CALL  3171
3165  7C                LD    A,H
3166  E60F              AND   0F
3168  12                LD    (DE),A
3169  4C                LD    C,H
316A  CD7131            CALL  3171      ;Check if address to be  inserted
316D  F1                POP   AF         ;is in RAM
316E  C1                POP   BC
316F  D1                POP   DE
3170  C9                RET
3171  13        L3171   INC   DE        ;check for RAM address
3172  0604              LD    B,04
3174  CB39      L3174   SRL   C
3176  10FC              DJNZ  3174
3178  79                LD    A,C
3179  E60F              AND   0F
317B  12                LD    (DE),A
317C  13                INC   DE
317D  C9                RET
317E  212720    L317E   LD    HL,2027
3181  7E                LD    A,(HL)
3182  CD9431            CALL  3194
3185  86                ADD   A,(HL)
3186  320144            LD    (4401),A
3189  2B                DEC   HL
318A  7E                LD    A,(HL)
318B  CD9431            CALL  3194      ;Load parameters from
318E  86                ADD   A,(HL)    ;step buffer into registers
318F  320044            LD    (4400),A  ;check if parameters
3192  1808              JR    319C      ;are legal. if yes
3194  0604      L3194   LD    B,04      ; calculate the sum of all data
3196  CB27      L3196   SLA   A         ;to be output
3198  10FC              DJNZ  3196
319A  2B                DEC   HL
319B  C9                RET
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE             COMMENTS



319C  212720    0323  L3196  LD   HL,2027
319F  0604      0324         LD   B,04      ;loop 4 times
31A1  D9        0325  L31A1  EXX
31A2  CD3232    0326         CALL 3232
31A5  D9        0327         EXX
31A6  77        0328         LD   (HL),A
31A7  2B        0329         DEC  HL
31A8  10F7      0330         DJNZ 31A1
31AA  18F0      0331         JR   3196
31AC  212720    0332  L31AC  LD   HL,2027    ;Dispaly data of break address
31AF  7E        0333         LD   A,(HL)
31B0  CD9432    0334         CALL 3194       ;Output 2K square wave  for  4000
31B3  86        0335         ADD  A,(HL)     ; cycles
31B4  320344    0336         LD   (4403),A   ; load parameters into registers
31B7  2B        0337         DEC  HL
31B8  7E        0338         LD   A,(HL)
31B9  CD9432    0339         CALL 3194       ;Dispalay data of address
31BC  86        0340         ADD  A,(HL)
31BD  320244    0341         LD   (4402),A
31C0  ED5B0244  0342         LD   DE,(4402)
31C4  2A0044    0343         LD   HL,(4400)
31C7  C3B535    0344         JP   35B5
31CA  E5        0345  L31CA  PUSH HL               ;Scan display
31CB  D5        0346         PUSH DE               ;& process address & keys
31CC  C5        0347         PUSH BC
31CD  F5        0348         PUSH AF
31CE  0E01      0349         LD   C,01
31D0  212220    0350         LD   HL,2022
31D3  0606      0351         LD   B,06
31D5  7E        0352  L31D5  LD   A,(HL)
31D6  1622      0353         LD   D,22
31D8  5F        0354         LD   E,A
31D9  1A        0355         OUT  (11),A
31DC  79        0356         LD   A,C
31DD  D310      0357         OUT  (10),A
31DF  23        0358         INC  HL
31E0  CB21      0359         SLA  C
31E2  10F1      0360         DJNZ 32D5
31E4  3E00      0361         LD   A,00
31E6  D310      0362         OUT  (10),A    ;clear the display
31E8  F1        0363         POP  AF
31E9  C1        0364         POP  BC
31EA  D1        0365         POP  DE
31EB  E1        0366         POP  HL
31EC  C9        0367         RET
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE            COMMENTS


31ED  C5         0368  L31ED    PUSH BC        ;Address Subroutine
31EE  212720     0369           LD   HL,2027
31F1  CD0032     0370           CALL 3200
31F4  57         0371           LD   D,A
31F5  CD0032     0372           CALL 3200      ;*****************************
31F8  5F         0373           LD   E,A       ;PRINTER subroutine
31F9  EB         0374           EX   DE,HL     ;******************************
31FA  CD4F31     0375           CALL 314F      ;prints a block of specified output
31FD  FB         0376           EX   DE,HL     ;to be invoked by the user
31FE  C1         0377           POP  BC        ;the top and bottom of the block
31FF  C9         0378           RET            ;is to specified at locations
3200  0604       0379  L3200    LD   B,04      ;200BH. and 200CH.
3202  7E         0380           LD   A,(HL)
3203  CB27       0381  L3203    SLA  A
3205  10FC       0382           DJNZ 3203
3207  2B         0383           DEC  HL
3208  86         0384           ADD  A,(HL)
3209  2B         0385           DEC  HL
320A  C9         0386           RET
320B  212720     0387  L320B    LD   HL,2027
320E  3E10       0388           LD   A,10
3210  0606       0389           LD   B,06
3212  77         0390  L3212    LD   (HL),A
3213  3C         0391           INC  A
3214  2B         0392           DEC  HL
3215  00         0393           NOP
3216  10FA       0394           DJNZ 3212
3218  C9         0395           RET
3219  E5         0396  L3219    PUSH HL        ;load initializing word
321A  C5         0397           PUSH BC
321B  F5         0398           PUSH AF
321C  212720     0399           LD   HL,2027
321F  3E16       0400           LD   A,16
3221  0606       0401           LD   B,06
3223  77         0402  L3223    LD   (HL),A
3224  3C         0403           INC  A
3225  2B         0104           DEC  HL
3226  10         0405           DJNZ 3223
3228  F1         0406           POP  AF
3229  C1         0407           POP  BC
322A  E1         0408           POP  HL
322B  C9         0409           RET
322C  CDBB32     0410  L322C    CALL 32B8           ;Look at key board
322F  CD0B32     0411           CALL 320B           ;check hex to 7seg. display val.
3232  CDCA31     0412  L3232    CALL 31CA           ;Scan display
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE              COMMENTS


3235  DB10      0413         IN   A,(10)
3237  FEFF      0414         CP   FF
3239  20F7      0415         JR   NZ,3223
323B  CDCA31    0416  L323B  CALL 31CA
323E  DB10      0417         IN   A,(10)
3240  FEFF      0418         CP   FF
3242  28F7      0419         JR   Z,323B
3244  57        0420         LD   D,A
3245  0E05      0421         LD   C,05
3247  06FF      0422  L3247  LD   B,FF
3349  10FE      0423  L3249  DJNZ 3249
324B  0D        0424         DEC  C
324C  20F9      0425         JR   NZ,3347
324E  DB10      0426         IN   A,(10)
3250  BA        0427         CP   D
3251  20DF      0428         JR   NZ,3232
3253  37        0429         SCF
3254  06FE      0430         LD   B,FE
3256  78        0431  L3256  LD   A,B
3257  D310      0432         OUT  (10),A    ;send to display
3259  DB10      0433         IN   A,(11)    ;get keyboard input
325B  FEFF      0434         CP   FF
325D  2004      0435         JR   NZ,3263
325F  CB10      0436         RL   B
3261  18F3      0437         JR   3256
3263  48        0438  L3263  LD   C,B
3264  2620      0439         LD   H,20
3266  7A        0440         LD   A,D
3267  CDAB33    0441         CALL 32AB      ;Display register and its state
326A  50        0442         LD   D,B
326B  79        0443         LD   A,C
326C  CDAB33    0444         CALL 32AB
326F  78        0445         LD   A,B
3270  0E03      0446         LD   C,03
3272  80        0447         ADD  A,B
3273  0D        0448         DEC  C
3274  20        0449         JR   NZ,3272
3276  82        0450         ADD  A,D
3277  6F        0451         LD   L,A
3278  7E        0452         LD   A,(HL)
3279  322120    0453         LD   (2021),A
327C  3E0F      0454         LD   A,0F
327E  95        0455         SUB  L
327F  3033      0456         JR   NC,3384
3281  3A2920    0457         LD   A,(2029)
```

```
LOC   OBJ CODE  LINE   LABEL   SRCE CODE               COMMENTS


3284  FE00      0458           CP   00
3286  281D      0459           JR   Z,32A5
3288  3A2920    0460           LD   A,(2029)
328B  47        0461           LD   B,A
328C  3A2120    0462           LD   A,(2021)
328F  FEA8      0363           CP   A8
3291  2812      0464           JR   Z,32A5
3293  B8        0465           CP   B
3294  209C      0466           JR   NZ,3232
3296  FE30      0467           CP   30
3298  2006      0468           JR   NZ,32A0
329A  33        0469           INC  SP
329B  33        0470           INC  SP
329C  3E        0471           LD   A,00
329E  1802      0472           JR   33A2
32A0  3E30      0473   L32A0   LD   A,30
32A2  322920    0474   L32A2   LD   (2029),A
32A5  6E        0475   L32A5   LD   L,(HL)
32A6  3A0038    0476           LD   A,(3800)
32A9  67        0477           LD   H,A
32AA  E9        0478           JP   (HL)
32AB  37        0479   L32AB   SCF               ;Register status subroutine
32AC  0608      0480           LD   B,08
32AE  05        0481   L32AE   DEC  B
32AF  CB27      0482           SLA  A
32B1  38FB      0483           JR   C,32AE
32B3  C9        0484           RET
32B4  CDCA31    0485   L32B4   CALL 31CA
32B7  3A2120    0486           LD   A,(2021)
32BA  C9        0487           RET
32BB  D9        0488   L32BB   EXX               ; Subrotine Look
32BC  08        0489           EX   AF,AF'
32BD  00        0490           NOP
32BD  00        0491           NOP
32BE  00        0492           NOP
32BF  00        0493           NOP
32C0  3E3B      0494           LD  A,3B
32C2  322920    0495           LD  (2029),A
32C5  3E03      0496           LD  A,03
32C7  322A20    0497           LD  (202A),A
32CA  3E10      0498           LD  A,10           ;Printer width
32CC  320043    0499           LD  (4300),A
32CF  3ECB      0500           LD  A,CB
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE              COMMENTS

32D1  320039    0501         LD  (3900),A
32D4  3ECC      0502         LD  A,CC
32D6  32FF45    0503         LD  (45FF),A         ;Clock setup loc.
32D9  FB        0504         EI                 ;subroutine DATAINI aquires data
32DA  CDEE32    0505         CALL 32EE          ;in the interupt mode
32DB  CD9A36    0506         CALL 369A          ;Acquire data
32DC  F3        0507         DI                 ;disable interupt
32DD  C9        0508         RET
32DE  3E30      0509         LD   A,30
32E0  320038    0510         LD   (3800),A
32E3  3E00      0511         LD   A,00
32E5  D314      0512         OUT  (14),A          ;Initialize port 14H.
32E7  210023    0513         LD   HL,2300
32EA  CD5236    0514         CALL 3652            ;Delay
32ED  C9        0515         RET
32EE  FD210746  0516  L32EE  LD   IY,4607         ;RTclock subroutine
32F2  DD210046  0517         LD   IX,4600         ;Scans the analog to digital
32F6  2608      0518         LD    H,08         ;converter  unit  when  interupt
32F8  DD7E00    0519  L32F8  LD   A,(IX)          ;mode
32FB  CD3833    0520         CALL 3334          ;
32FE  FD7700    0521         LD   (IY),A
32FF  77        0522         LD   (HL),A
3300  00        0523         NOP
3301  DD23      0524         INC   IX
3303  FD2B      0525         DEC   IY
3305  25        0526         DEC   H
3306  20F0      0527         JR    NZ,32F8
3308  3AFF45    0528         LD    A,(45FF)
330B  CD3833    0529         CALL  3334
330E  FE00      0530         CP    00
3310  C27036    0531         JP    NZ,3670
3313  0E0F      0532         LD    C,0F
3315  210746    0533         LD    HL,4607
3318  7F        0534  L3318  LD    A,(HL)
3319  CD4B36    0535         CALL 364B          ;shift data for display
3320  ED79      0536         OUT  (C),A
332E  0D        0537         DEC  C
331F  7E        0538         LD   A,(HL)
3320  E60F      0539         AND  0F
3322  ED79      0540         OUT  (C),A
3324  2B        0541         DEC  HL
3325  0D        0542         DEC  C
3326  20F0      0543         JR   NZ,3318
3328  7F        0544         LD   A,(HL)
3329  E60F      0545         AND  0F
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE              COMMENTS


332B  CB57      0546         BIT  2,A
332D  C27036    0547         JP   NZ,3670
3330  ED79      0548         OUT  (C),A
3332  3CCC      0549         LD   A,CC
3334  2AFF45    0550         LD   HL,(45FF)
3337  C9        0551         RET
3338  0E06      0552  L3438  LD   C,06
333A  5F        0553         LD   E,A
333B  D60A      0554         SUB  0A
333D  3823      0555         JR   3462
333F  7B        0556         LD   A,E
3340  E60F      0557         AND  0F
3342  CD7036    0558         CALL 3670         ;DATAIN
3344  CD3616    0559         CALL 3616   ;this subroutine reads data for
3347  3E10      0560         LD   A,10        ;in normal mode
3348  80        0561         ADD  A,B
334A  47        0562         LD B,A
334C  C9        0563         RET
334F  7B        0564         LD   A,E
3350  E6F0      0565         AND  F0
3352  0604      0566         LD   B,04
3354  CB3F      0567  L3354  SRL  A
3356  10FC      0568         DJNZ 3354
3358  47        0569         LD   B,A
3359  AF        0570         XOR  A          ;clear carry flag
335A  81        0571  L335A  ADD  A,C
335B  10FD      0572         DJNZ 335A
335D  4F        0573         LD   C,A
335E  7B        0574         LD   A,E
335F  91        0575         SUB  C
3360  1801      0576         JR   3363
3362  7B        0577  L3362  LD   A,E
3363  C9        0578  L3363  RET
3364  DB00      0579         IN   A,(00)
3366  210048    0580  L3366  LD   HL,4800
3369  0E0D      0580         LD   C,0D
336B  ED78      0581  L336B  IN   A,(C)
336D  CD1234    0582         CALL 3410
3370  0D        0583         DEC  C
3371  ED40      0584         IN   B,(C)
3373  80        0585         ADD  A,B
3374  77        0586         LD   (HL),A
3375  2B        0587         DEC  HL
3376  0D        0588         DEC  C
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE             COMMENTS

3377  3E        0589         LD   A,01
3379  B9        0590         CP   C
337A  20EF      0592         JR   NZ,3368
337C  DB00      0592         IN   A,(00)
337E  20E6      0593         JR   NZ,3366
3380  1603      0594         LD   D,03
3382  0E10      0595         LD   C,10
3384  210648    0596  L3487  LD   HL,4806   ;Subroutine update the RT clock
3387  ED78      0597         IN   A,(C)     ;for display
3389  CD1234    0598         CALL 3410
338C  ED40      0599         IN   B,(C)
338E  80        0600         ADD  A,B
338F  77        0601         LD   (HL),A
3390  23        0602         INC  HL
3391  15        0603         DEC  D
3392  20F3      0604         JR   NZ,3387
3394  CDAD34    0605         CALL 34AD
3397  21AD35    0606         LD   HL,225C
339A  CD5E36    0607         CALL 365E
339D  0603      0608         LD   B,03
339F  210348    0609         LD   HL,4803
33A2  7E        0610  L33A2  LD   A,(HL)
33A3  4F        0611         LD   C,A
33A4  CD1636    0612         CALL 3616
33A7  05        0613         DEC  B
33A8  280A      0614         JR   Z,33B4
33AA  3E2F      0615         LD   A,2F
33AC  D312      0616         OUT  (12),A    ;send to the out put port
33AE  CD3C37    0617         CALL 363C
33B1  23        0618         INC  HL
33B2  18EE      0619         JR   33A2
33B4  216022    0620  L33B4  LD   HL,2260
33B7  CD5E36    0621         CALL 365E
33BA  0603      0622         LD   B,03H     ;loop 3 times
33BC  210248    0623         LD   HL,4802
33BF  7E        0624  L34BF  LD   A,(HL)
33C0  4F        0625         LD   C,A
33C1  CD1636    0626         CALL 3616
33C4  05        0627         DEC  B
33C5  280A      0628         JR   Z,33D1
33C7  3E3A      0629         LD   A,3A
33C9  D312      0630         OUT  (12),A
33CB  CD3C36    0631         CALL 363C
33CE  2B        0632         DEC  HL
33CF  18EE      0633         JR   33BF
33D1  3E0D      0634  L33D1  LD   A,0D
33D3  D312      0635         OUT  (12),A
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE                COMMENTS


33D5  CD3C36     0636           CALL 363C
33D8  216422     0637           LD   HL,2264
33DB  CD5E36     0638           CALL 365E
33DE  210648     0639           LD   HL,4806
33E1  0602       0640           LD   B,02
33E3  CD1636     0641   L33E3   CALL 3616       ;set decimal locations
33E6  23         0642           INC  HL
33E7  10FA       0643           DJNZ 33E3
33E9  3E2F       0644           LD   A,2F
33EB  D312       0645           OUT  (12),A
33ED  CD3C36     0646           CALL 363C
33F0  216822     0647           LD   HL,2268
33F3  CD5E36     0648           CALL 365E
33F6  210848     0649           LD   HL,4808
33F9  CD1636     0650           CALL 3616
33FC  3E0D       0651           LD   A,0D
33FE  D312       0652           OUT  (12),A
33FF  12         0653           LD   (DE),A
3400  CD3C36     0654           CALL 363C
3403  C9         0655           RET
3404  CDEE32     0656           CALL 32EE      ';subrotine to intialize RTClock
3407  CDED35     0657           CALL 35ED      ;
340A  CD3232     0658           CALL 3232
340D  C3A830     0659           JP   30A8
3410  0604       0660           LD   B,04
3412  CB27       0661   L3512   SLA  A
3414  10FC       0662           DJNZ 3512
3416  C9         0663           RET
3417  D9         0664           EXX             ;Register keys subroutine
3418  08         0665           EX   AF,AF'     ;this subroutine invokes others in
3419  3FCD       0666           LD   A,CD       ;processing the register keys to
341B   320039     0667          LD   (3900),A  ;access registers and display  the
341E  ED430240   0668           LD   (4002),BC ;the data
3422  ED530440   0669           LD   (4004),DE
3426  ED060640   0670           LD   (4006),HL
3429  F5         0671           PUSH AF
342A  E1         0672           POP  HL
342B  220040     0673           LD   (4000),HL
342E  DD220840   0674           LD   (4008),IX
3432  FD220A40   0675           LD   (400A),IY
3436  ED730C40   0676           LD   (400C),SP
343A  D9         0677           EXX
343B  08         0678           EX    AF,AF'
343C  ED431040   0679           LD    (4010),BC
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE               COMMENTS


3440  ED531240   0680           LD    (4012),DE
3444  221440     0681           LD    (4014),HL
3447  F5         0682           PUSH  AF
344B  E1         0683           POP   HL
3449  220E40     0683           LD    (400E),HL
344C  C38D36     0684           JP    368D              ;Branch if error.
344F  3A0039     0685           LD    A,(3909)
3452  FECD       0686           CP    CD
3454  C27036     0687           JP    NZ,3670
3457  1601       0688           LD    D,01
3457  1E04       0689           LD    E,04
345B  216622     0691           LD    HL,2266
345E  0E0B       0692           LD    C,0B
3460  0601       0693           LD    B,01
3462  3E20       0694   L3462   LD    A,20     ;output control byte
3464  D312       0695   L3464   OUT   (13),A
3466  CD3C36     0696           CALL  363C
3469  10F7       0694           DJNZ 3462
346B  0602       0698           LD   B,02
346D  7E         0699           LD   A,(HL)
346E  D312       0700           OUT  (12),A
3470  CD3C36     0701           CALL 363C
3473  23         0702           INC  HL
3474  10F7       0703           DJNZ 346D
3476  3E00       0704           LD   A,00
3478  BA         0705           CP   D
3479  2809       0706           JR   Z,3484
347B  3E05       0707           LD   A,05
3470  B9         0708           CP   C
347E  20CD       0709           JR   NZ,348D
3480  1600       0710           LD   C,00
3482  1809       0711           JR   348D
3484  3E27       0712   L3484   LD   A,27
3486  D312       0713           OUT  (12),A
3488  CD3C36     0714           CALL 363C
348B  1E03       0715           LD   E,03
348D  43         0716   L348D   LD   B,E
348E  3E20       0717   L348E   LD   A,20
3490  D312       0718           OUT  (12),A
3492  CD3C36     0719           CALL 363C
3495  10F7       0720           DJNZ 348E
3497  0D         0721           DEC  C
3498  20D1       0722           JR   NZ,346B
349A  3E0D       0723           LD   A,0D
349C  D312       0724           OUT  (12),A    ;Display output
349E  CD3C37     0725           CALL 363C
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE            COMMENTS


349E  210040    0726         LD   HL,4000
34A4  1E01      0727         LD   E,01
34A6  1616      0728         LD   D,16
34A8  7E        0729  L34A8  LD   A,(HL)
34A9  4F        0730         LD   C,A
34AA  CD1637    0731         CALL 3616
34AD  3E02      0732         LD   A,02
34AF  BB        0733         CP   C
34B0  200D      0734         JR   NZ,34BF
34B2  0602      0735         LD   B,02
34B4  3E20      0736   L34B4 LD   A,20
34B6  D312      0737         OUT  (14),A
34B8  CD3C36    0738         CALL 363C
34BB  10F7      0739         DJNZ 34B4
34BD  1E00      0740         LD   E,00
34BF  23        0741   L35BF INC  HL
34C0  1C        0742         INC  E
34C1  15        0743         DEC  C
34C2  20E4      0744         JR   NZ,34A8
34C4  3E0D      0745         LD   A,0D
34C6  D312      0746         OUT  (12),A
34C8  CD3C36    0747         CALL 363C
34CB  C33232    0748         JP   3232
34CE  212720    0749         LD   HL,2027  ;register AF.
34D1  360A      0750         LD   (HL),0A
34D3  2B        0751         DEC  HL
34D4  360F      0752         LD   (HL),0F
34D6  ED5B0040  0753         LD   DE,(4000)
34DA  C37A35    0754         JP   357A
34DD  212720    0755         LD   HL,2027  ;register BC
34E0  3608      0756         LD   (HL),08
34E2  2B        0757         DEC  HL
34E3  360C      0758         LD   (HL),0C
34E5  ED5B0240  0759         LD   DE,(4002)
34E9  C37A35    0760         JP   357A
34EC  1810      0761         JR   35FE
34EE  181C      0762         JR   350C
34F0  1828      0763         JR   351A
34F2  1834      0764         JR   3528
34F4  1840      0765         JR   3536
34F6  184C      0766         JR   3544
34F8  1858      0767         JR   3552
34FA  1864      0768         JR   3560
34FC  1870      0769         JR   356E
34FE  212720    0770  L34FE  LD   HL,2027  ;registerDE
```

```
LOC   OBJ CODE   LINE   LABEL   SRCE CODE              COMMENTS


3501  00         0771           NOP
3500  00         0772           NOP
3502  0D         0773           DEC   C
3503  2B         0774           DEC   HL
3504  360E       0775           LD    (HL),0E
3506  ED5B0440   0776           LD    DE,(4004)
350A  186E       0777           JR    357A
350C  212720     0778   L350C   LD   HL,2027    ;Register HL
350F  364A       0779           LD   (HL),4A
3511  2B         0780           DEC  HL
3512  364B       0781           LD   (HL),4B
3514  ED5B0640   0782           LD   DE,(4006)
351B  1860       0783           JR   357A
351A  212720     0784   L351A   LD   HL,2027    ;Register IX
351D  3601       0785           LD   (HL),01
351F  2B         0786           DEC  HL
3520  364C       0787           LD   (HL),4C
3522  ED5B0840   0788           LD   DE,(4008)
3526  1852       0789           JR   357A
3528  212720     0790   L3528   LD   HL.2027    ;Register IY
352B  3601       0791           LD   (HL),01
352D  2B         0792           DEC  HL
352E  364D       0793           LD   (HL),4D
3530  ED5B0A40   0794           LD   DE,(400A)
3534  1844       0794           JR   357A
3536  212720     0796   L3536   LD   HL,2027    ;Stack Pointer
3539  3605       0797           LD   (HL),05
353B  2B         0798           DEC  HL
353C  364E       0799           LD   (HL),4E
353E  ED5B0C40   0800           LD   DE,(400C)
3542  1836       0801           JR   357A
3544  212720     0802   L3544   LD   HL,2027    ;Register AF'
3547  360A       0803           LD   (HL),0A
3549  2B         0804           DEC  HL
354A  360F       0805           LD   (HL),0F
354C  ED5B0F40   0806           LD   DE,(4004)
3550  1828       0807           JR   357A
3552  212720     0808   L3552   LD   HL,2027    ;Register BC'
3555  360B       0809           LD   (HL),0B
3557  2B         0810           DEC  HL
3558  360C       0811           LD   (HL),0C
355A  ED5B1040   0812           LD   DE,(4010)
355E  181A       0813           JR   357A
3560  212720     0814   L3560   LD   HL,2027    ;Register DE'
3563  360D       0815           LD   (HL),0D
3565  2B         0816           DEC  HL
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE         COMMENTS

3566  360E      0817         LD   (HL),0E
3568  ED5B1240  0818         LD   DE,(4012)
356C  180C      0819         JR   357A
356E  212720    0820  L356E  LD   HL,2027    ;Register HL'
3571  364A      0821         LD   (HL),4A
3573  2B        0822         DEC  HL
3574  364B      0823         LD   (HL),4B
3576  ED5B1440  0824         LD   DE,(4014)
357A  3A0039    0825  L366A  LD   A,(3900)
357D  FECD      0826         CP   CD
357F  C27035    0827         JP   NZ,3570
3582  DD212420  0828         LD   IX,2024
3586  4A        0829         LD   C,D
3587  CD9135    0830         CALL 3591
358A  4B        0831         LD   C,E
358B  CD9135    0832         CALL 3591
358E  C33232    0833         JP   3232
3591   79       0834   L3591 LD    A,C       ;Register  7-segment   display
3592  E60F      0835         AND  0F         ;subroutine
3594  CDA635    0836         CALL 35A6
3597  DD7700    0837         LD   (IX),A
359A  DD23      0838         INC  IX
359C  DD7100    0839         LD   (IX),C
359F  DD2B      0840         DEC  IX
35A1  DD2B      0841         DEC  IX
35A3  DD2B      0842         DEC  IX
35A5  C9        0843         RET
35A6  0604      0844  L35A6  LD   B,04       ;Shift data for register display
35AB  CB39      0845  L35A8  SRL  C
35AA  10FC      0846         DJNZ 35A8
35AC  C9        0847         RET
35AD  3E0D      0848  L35AD  LD   A,0D       ;Subroutine resets the printer
35AF  D312      0849         OUT  (13),A
35B1  CD3C36    0850         CALL 363C
35B4  C9        0851         RET
35B5  CDAD35    0852  L35B5  CALL 35AD       ;Printer subroutine
35B8  3E00      0853         LD   A,00       ;Prints a block of data from memory
35BA  320043    0854         LD   (4300),A
35BD  7C        0855  L35BD  LD   A,H
35BE  4F        0856         LD   C,A
35BF  CD1636    0857         CALL 3616
35C2  7D        0858         LD   A,L
35C3  4F        0859         LD   C,A
35C4  CD1636    0860         CALL 3616
35C7  0602      0861         LD   B,02
```

```
LOC   OBJ CODE   LINE  LABEL  SRCE CODE            COMMENTS

35C9  3E20       0862  L35C9  LD   A,20
35CB  D312       0863         OUT  (15),A    ;output tp printer port
35CD  CD3C36     0864         CALL 363C
35D0  10F7       0865         DJNZ 35C9
35D2  3E20       0866  L35D2  LD   A,20
35D4  D312       0867         OUT  (15),A
35D6  CD3C36     0868         CALL 363C
35D9  7E         0869         LD   A,(HL)
35DA  4F         0870         LD   C,A
35DB  CD1636     0871         CALL 3616
35DE  7D         0880         LD   A,L
35DF  BB         0881         CP   E
35E0  2018       0882         JR   NZ,35FA
35E2  7C         0883         LD   A,H
35E3  BA         0884         CP   D
35E4  2014       0885         JR   NZ,35FA
35E6  3E0D       0886         LD   A,0D
35E8  D312       0887         OUT  (15),A
35EA  CD3C36     0888         CALL 363C
35ED  0604       0889  L35ED  LD   B,04      ;save data to be used with RTC
35EF  212720     0890         LD   HL,2027
35F2  3E1E       0891         LD   A,1E
35F4  77         0892  L35F4  LD   (HL),A
35F5  2B         0893         DEC  HL
35F6  3C         0894         INC  A
35F7  10FB       0895         DJNZ 35F4
35F9  C9         0896         RET
35FA  23         0897  L35FA  INC  HL
35FB  3A0043     0898         LD   A,(4300)
35FE  FE17       0899         CP   17
35FF  17         0900         RLA
3600  200E       0901         JR   NZ,3610
3602  3E00       0902         LD   A,00
3604  320043     0903         LD   (4300),A
3607  3E0D       0904         LD   A,0D
3609  D312       0905         OUT  (15),A
360B  CD3C36     0906         CALL 363C
361E  18AD       0907         JR   35BD
3610  3C         0908  L3610  INC  A
3611  320043     0909         LD   (4300),A
3614  18BC       0910         JR   35D2
3616  CD2A36     0911  L3616  CALL 362A      ;convert Hex to ASCII and send to
3619  D312       0912         OUT  (15),A    ;printer port
361B  CD3C36     0913         CALL 363C
361E  79         0914         LD   A,C
361F  E60F       0915         AND  0F
```

```
LOC   OBJ CODE  LINE  LABEL  SRCE CODE              COMMENTS

3621  CD3036    0916         CALL 362A
3624  D312      0917         OUT  (15),A
3626  CD3C36    0918         CALL 363C
3629  C9        0919         RET
362A  0604      0920  L362A  LD   B,04      ;convert key position to
362C  CB3F      0921  L362C  SRL  A         ;key internal code
362E  10FC      0922         DJNZ 362C
3630  FE0A      0923  L3630  CP   0A
3632  C23936    0924         JP   P,3639
3635  C630      0925         ADD  A,30
3637  1802      0926         JR   353B
3639  C637      0927  L3639  ADD  A,37
363B  C9        0928  L363B  RET
363C  DB12      0929  L363C  IN   A,(12)    ; Subroutine  for handshaking
363E  E601      0930         AND  01        ;lsb of bit 7 indicates ready
3640  20FA      0931         JR   NZ,363C
3642  3E00      0932         LD   A,00
3644  D313      0933         OUT  (13),A
3646  3E01      0934         LD   A,01
3648  D313      0935         OUT  (13),A
364A  C9        0936         RET
364B   0604     0937  L364B  LD   B,04      ;Rotate four bits for  the  clock
364D  CB3F      0938  L364D  SRL  A         ;display in the address field.
364F  10FC      0939         DJNZ 364D
3651  C9        0940         RET
3652  111020    0941   L3652 LD   DE,2010   ;subrotine v - get display code
3655  060F      0942         LD   B,0F
3657  7E        0943   L3657 LD   A,(HL)
3658  12        0944         LD   (DE),A
3659  23        0945         INC  HL
365A  13        0946         INC  DE
365B  10FA      0947         DJNZ 3657
365D  C9        0948         RET
365E  0604      0949   L365E LD   B,04      ;subrotine OUT
3660  7E        0950   L3660 LD   A,(HL)
3661  D312      0951         OUT  (12),A    ;sends data to the display unit
3663  CD3C36    0952         CALL 363C
3666  10F8      0953         DJNZ 3660
3668  3E2F      0954         LD   A,2F
366A  D312      0955         OUT  (12),A
366C  CD3C36    0956         CALL 363C
367F  C9        0957         RET
3670  C33232    0958  L3670  JP   3232
3673  3A0038    0959  L3673  LD   A,(3800)
3676  FE30      0960         CP   30
```

```
LOC   OBJ CODE  LINE  LABEL  SPCE CODE                COMMENTS


3678  2004      0961         JR   NZ,367E
367A  3E3E      0962         LD   A,3E
367C  1802      0963         JR   3680
367E  3E44      0964  L367E  LD   A,44
3680  21272     0965  L3680  LD   HL,2027
3683  0606      0966         LD   B,06
3685  77        0967  L3685  LD   (HL),A
3686  3C        0969         INC  A
3687  2B        0969         DEC  HL
3688  10FB      0970         DJNZ 3689
368A  C33232    0971         JP   3232
358D  212720    0972         LD   HL,2027
368A  C33232    0973         JP   3232
368D  212720    0974  L368D  LD   HL,2027
3690  0606      0975         LD   8,06
3692  3E24      0976         LD   A,24
3594  77        0977  L3694  LD   (HL),A
3695  3C        0978         INC  A
3696  2B        0979         DEC  HL
3697  10FB      0980         DJNZ 3694
3699  C33232    0981         JP   3232
369A  770040    0982         LD HL, 4000     ;data input subroutine
369D  0610      0983         LD B, 10H        ;Scan 16 channells
369F  78        0984         LD A,B
3700  D314      0985         OUT (14),A      ;Output to address control chanell
3702  DB13      0986         IN A,(13)       ;read data
3704  77        0987         LD (HL),A
3705  22        0988         INC HL
3706  1006      0989         DJNZ
3708  C9        0990         RET
```

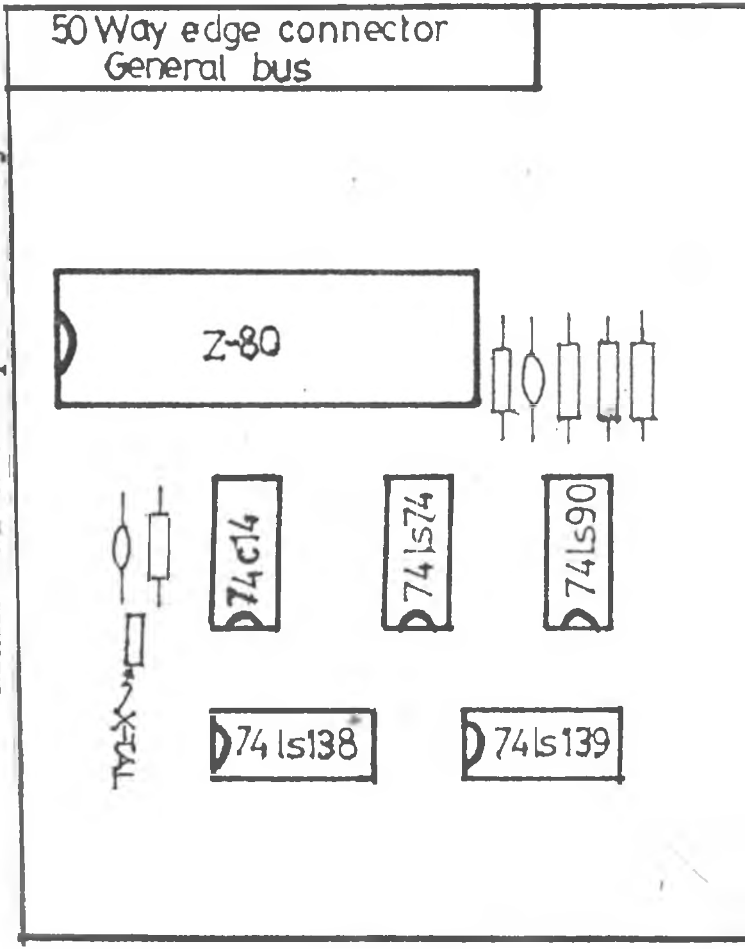
Appendix C4

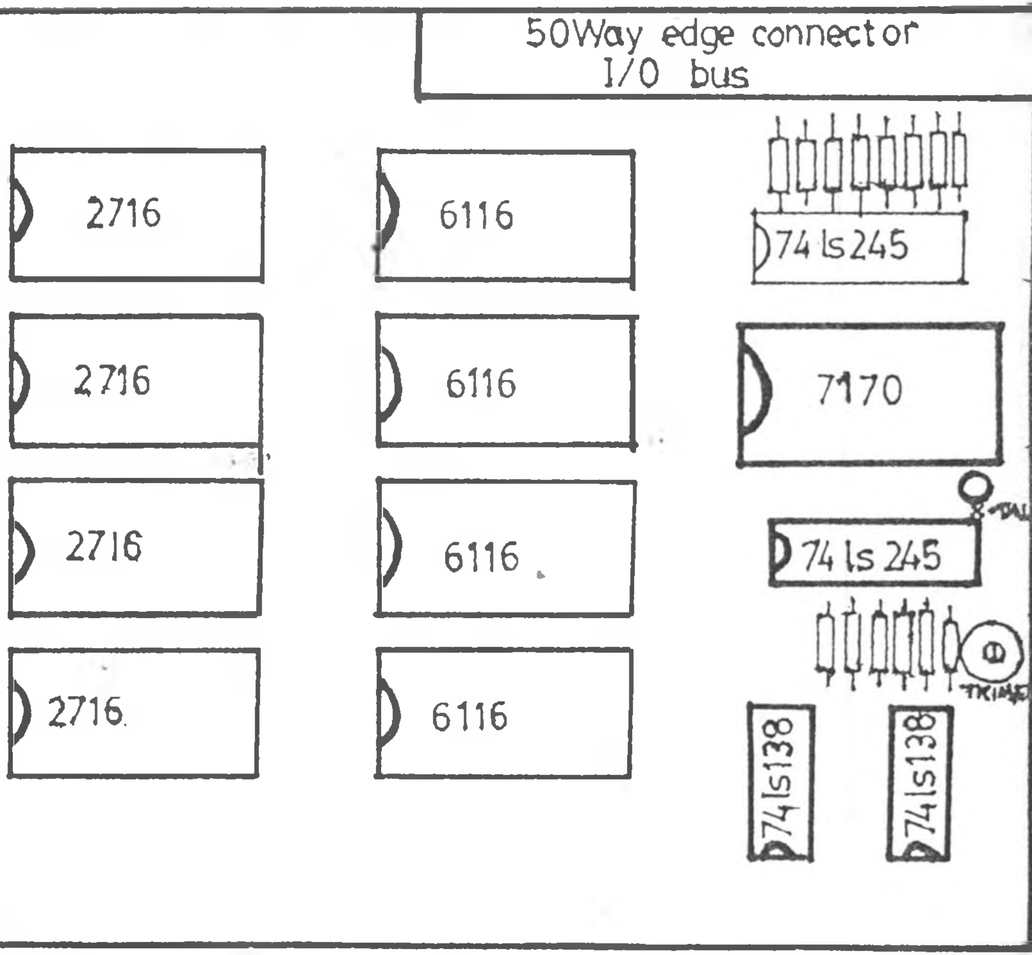
Real Time Clock BASIC Test Program

```
10 REM REAL TIME CLOCK TEST PROGRAM
20 REM -------------------------------------------
30 REM PROGRAMMED BY MADAHANA A.
40 REM SUPERVISED BY PROF. W. H. DRAKE
50 REM UNIVERSITY OF NAIROBI
60 REM DEPARTMENT  OF PHYSICS
65 REM UNIVERSITY OF NAIROBI
70 REM ---------------------------------------------
80 REM THE CLOCK IS PROGRAMMED FOR 12 HR. MODE
90 CN = 19 :CLS
100 OUT 14,3 :REM REAL TIME CLOCK CONTROL PORT ; f = 1.38Mhz
110 OUT 145,24 :REM 12 HR. MODE
120 HSC = INP(128) AND 127: REM READ HUNDREDS OF SECONDS
130 SC = INP(131) AND 63 : REM READ SECONDS
140 MIN = INP(129) AND 15 : REM READ MINUTES
150 SCREEN 2,10 :PRINT " TIME"
160 SCREEN 7,10 :PRINT ":"
170 SCREEN 11,10 :PRINT "MIN"
180 SCREEN 14,10 :PRINT ":"
190 SCREEN 15,10 :PRINT"SEC"
200 SCREEN 19,10 :PRINT ":"
210 SCREEN 20,10 :PRINT "HSEC"
220 GOSUB 290
230 M = INP (200)  :REM PM OR AM
240 IF M = 0 GOTO 270
250 SCREEN 23,10  :PRINT "PM"
260 GOTO 120
270  SCREEN 23,10 :PRINT "AM"
280 GOTO 120
290 MTH = INP (132) AND 15 :REM READ MONTH
300 DTE = INP (133) AND 31 :REM READ DATE
310 YR = INP (134) AND 127 :REM READ YEAR
320 DAY = INP (135) AND 7 :REM READ DAY OF WEEK
330 REM FIND WHICH DAY OF THE WEEK
340 IF DAY  = O GOSUB 500
350 IF DAY  = 1 GOSUB 520
360 IF DAY  = 2 GOSUB 540
370 IF DAY  = 3 GOSUB 560
380 IF DAY  = 4 GOSUB 580
390 IF DAY  = 5 GOSUB 600
400 IF DAY  = 6 GOSUB 620
410 REM PROCESS DATE HERE
420 SCREEN 2,6 :PRINT "DATE"
430 SCREEN 6,8 :PRIINT ":"
440 SCREEN 8,8 :PRINT DTE
450 SCREEN 10,8 :PRINT   "/"
460 SCREEN 11,8 :PRINT MTH
470 SCREEN 12,8 :PRINT "/"
480 SCREEN 15,8 :PRINT  YR
500 SCREEN 2,6 :PRINT "MONDAY     " :RETURN
520 SCREEN 2,6 :PRINT  "TUESDAY    " :RETURN
540 SCREEN 2,6 :PRINT "WEDNESDAY " :RETURN
560 SCREEN 2,6 :PRINT " THURSDAY     " :RETURN
580 SCREEN 2,6 :PRINT  "FRIDAY    ":RETURN
600 SCREEN 2,6 :PRINT "SUTURDAY  " :RETURN
620 SCREEN 2,6 :PRINT "SUNDAY       " :RETURN
640 END
```

Appendix D1

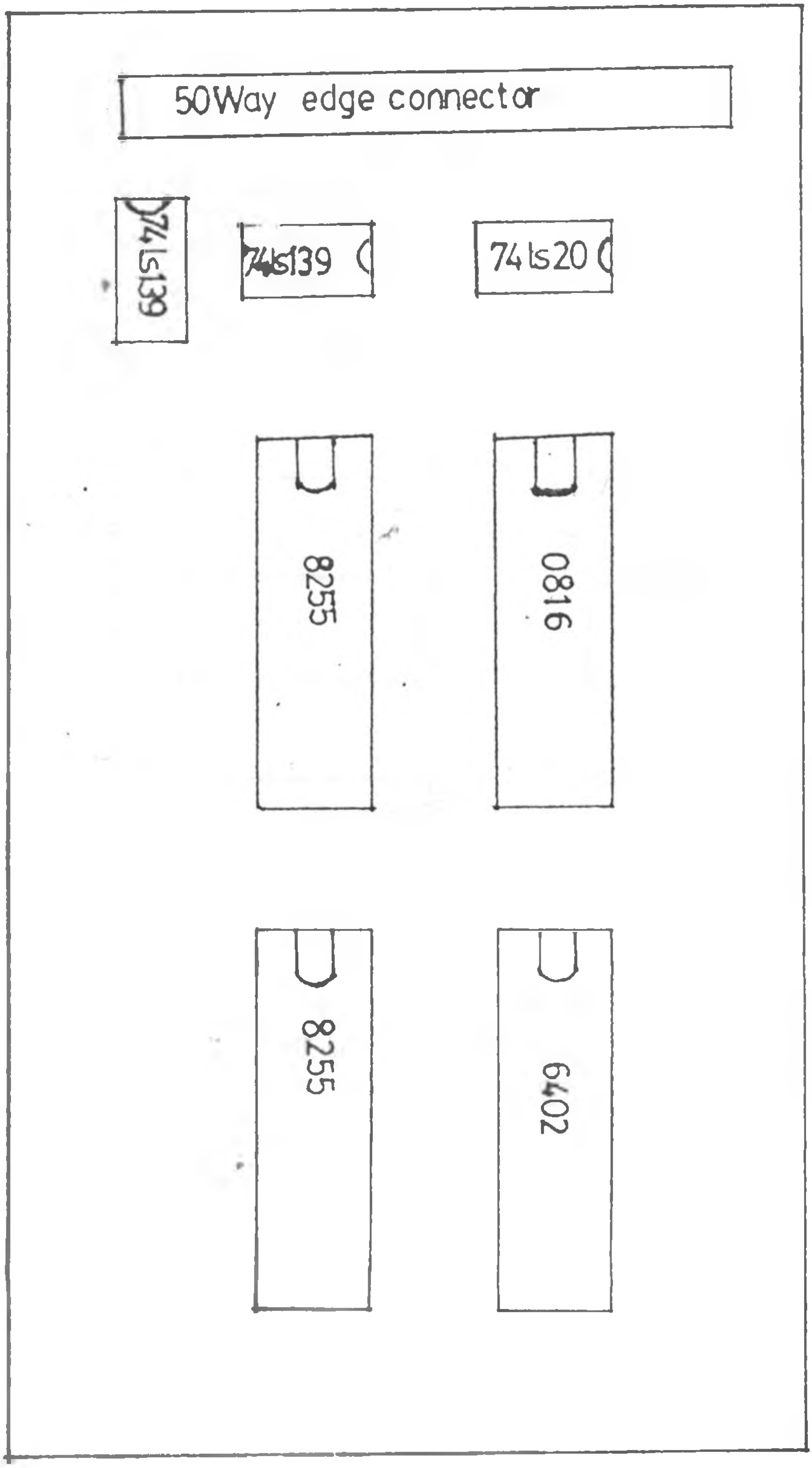
Main Board Components Layout




50Way edge connector
I/O bus
2716
2716
2716
2716
6116
6116
6116
6116
74 ls245
7170
XTAL
74 ls 245
TRIME
74ls138
74ls138

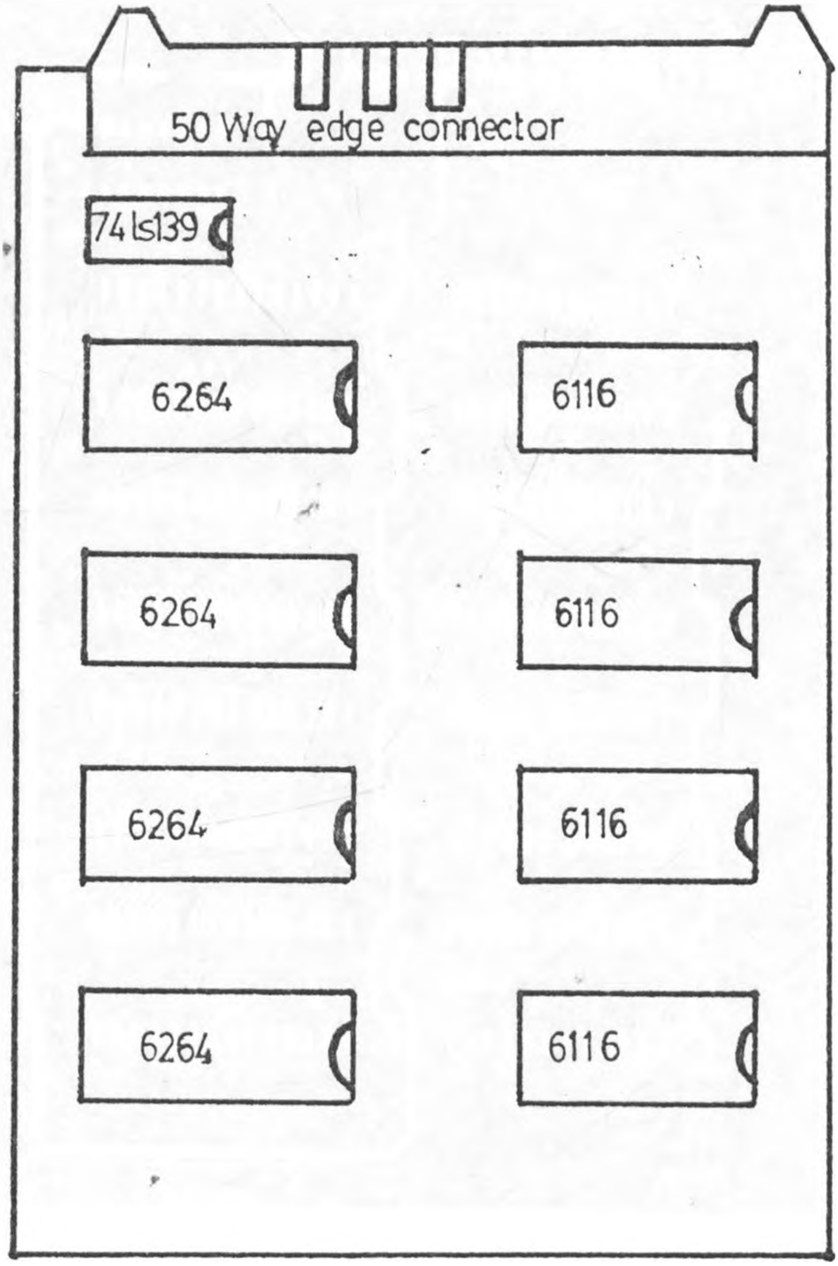
Appendix D2

I/O Board components Layout



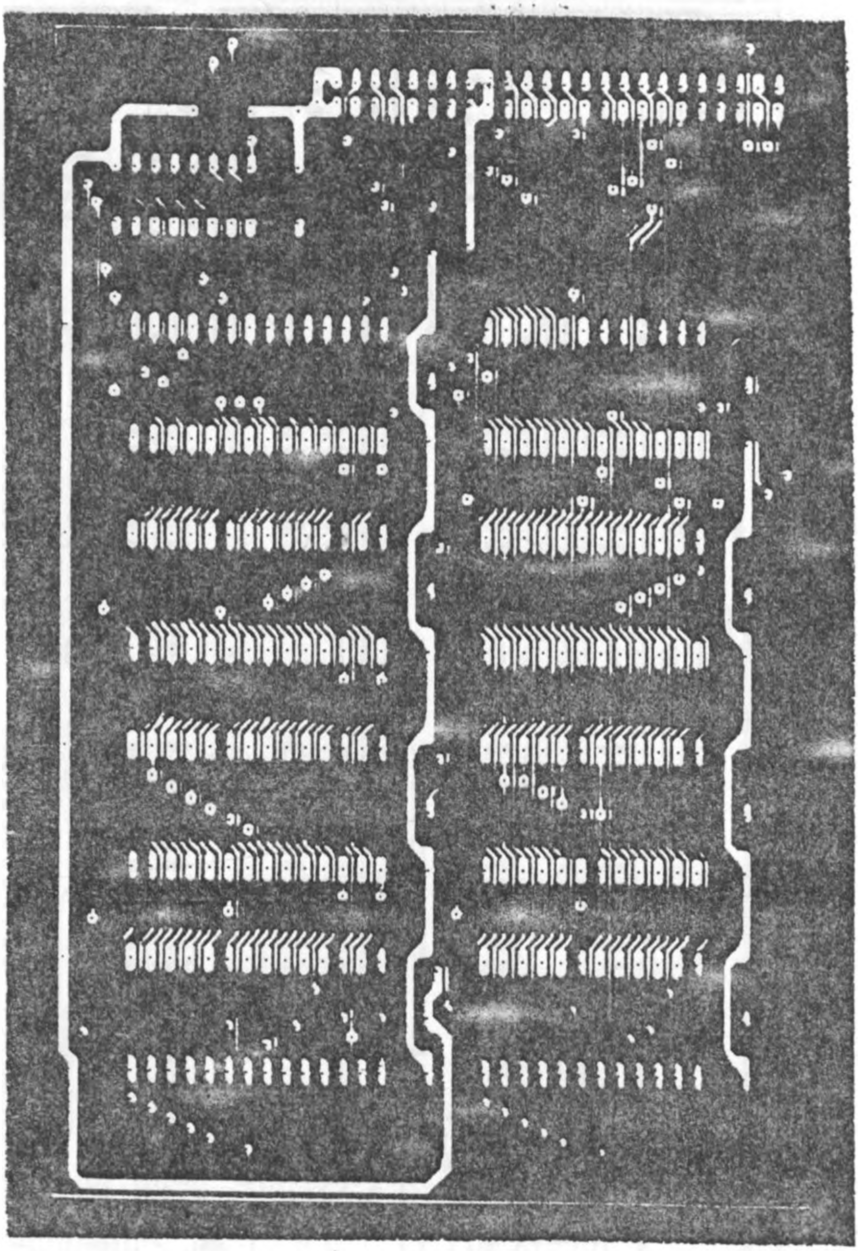
Appendix D3

Ram Board components Layout



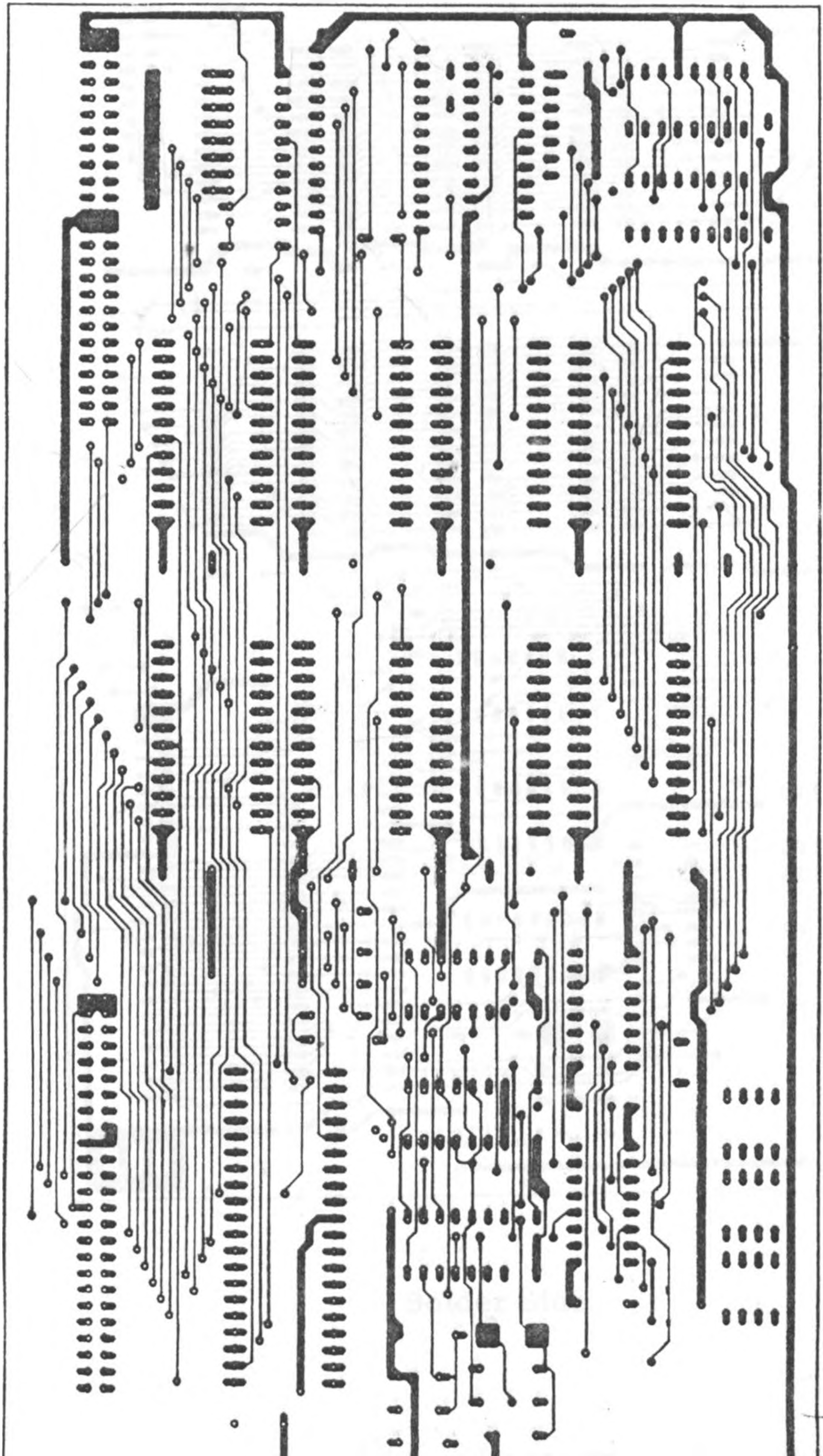
Appendix D4

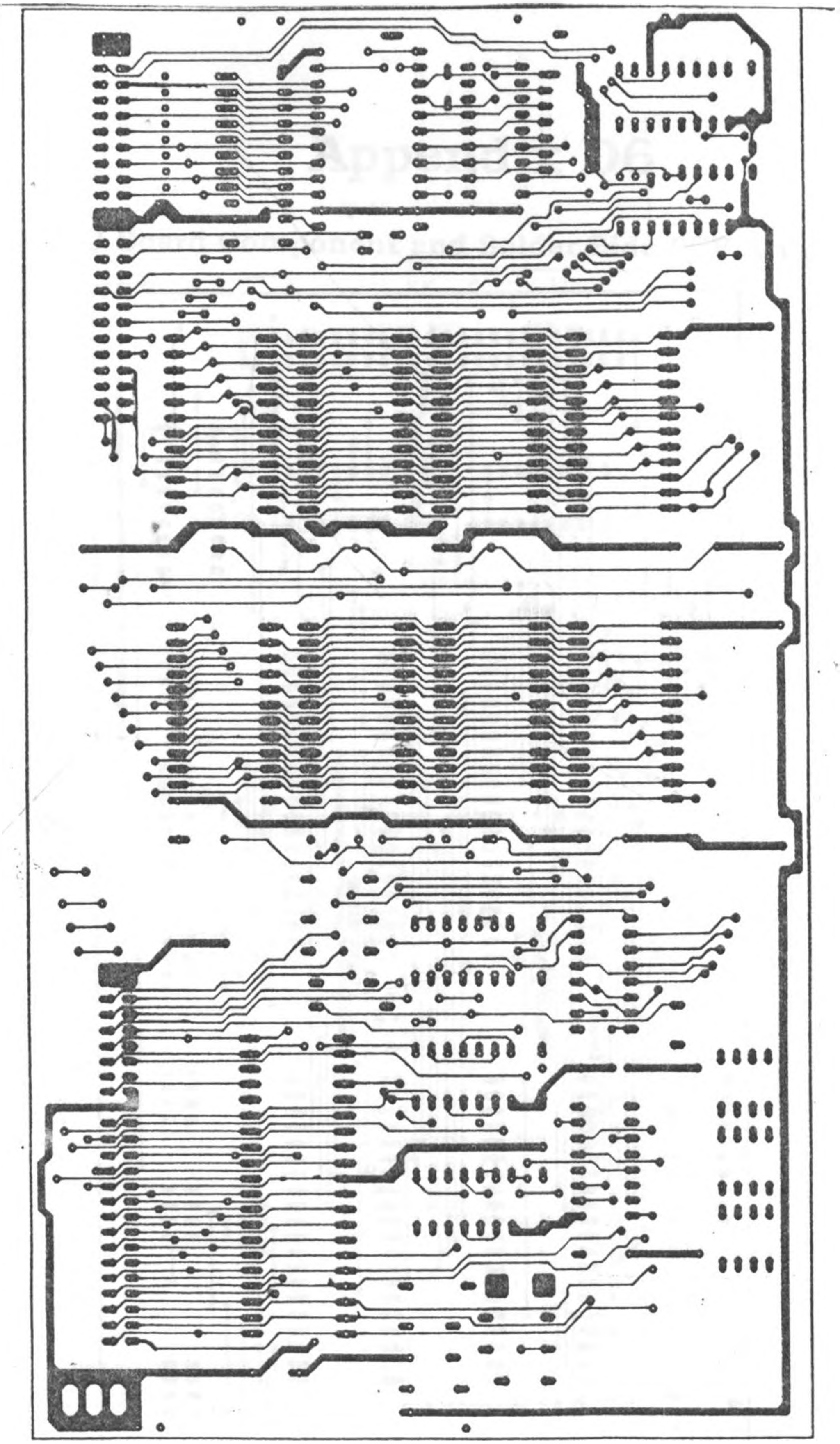
Ram Board PCB Artwork Negative



Appendix D5

Main Board Component and solder side PCB Artwork

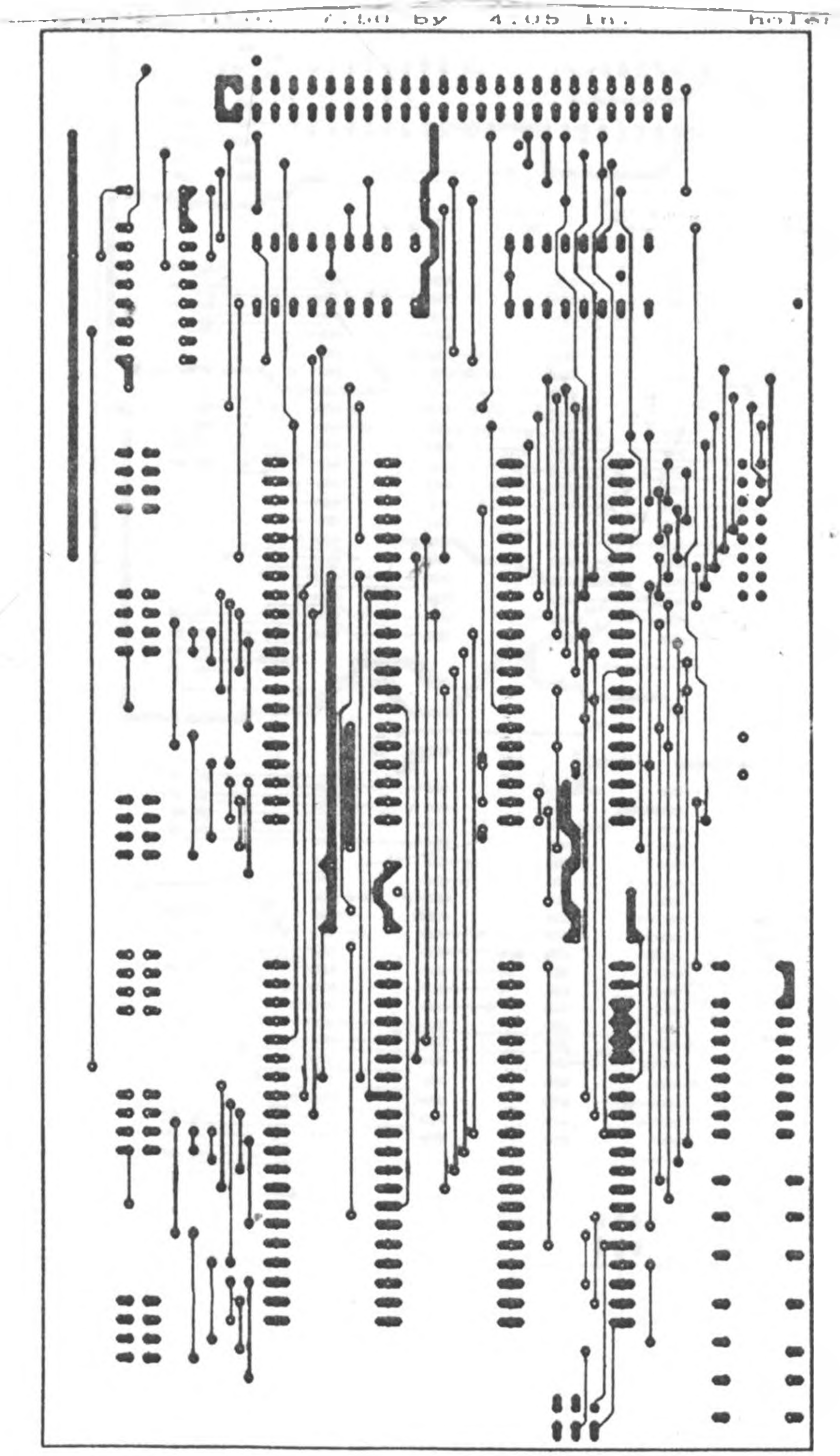




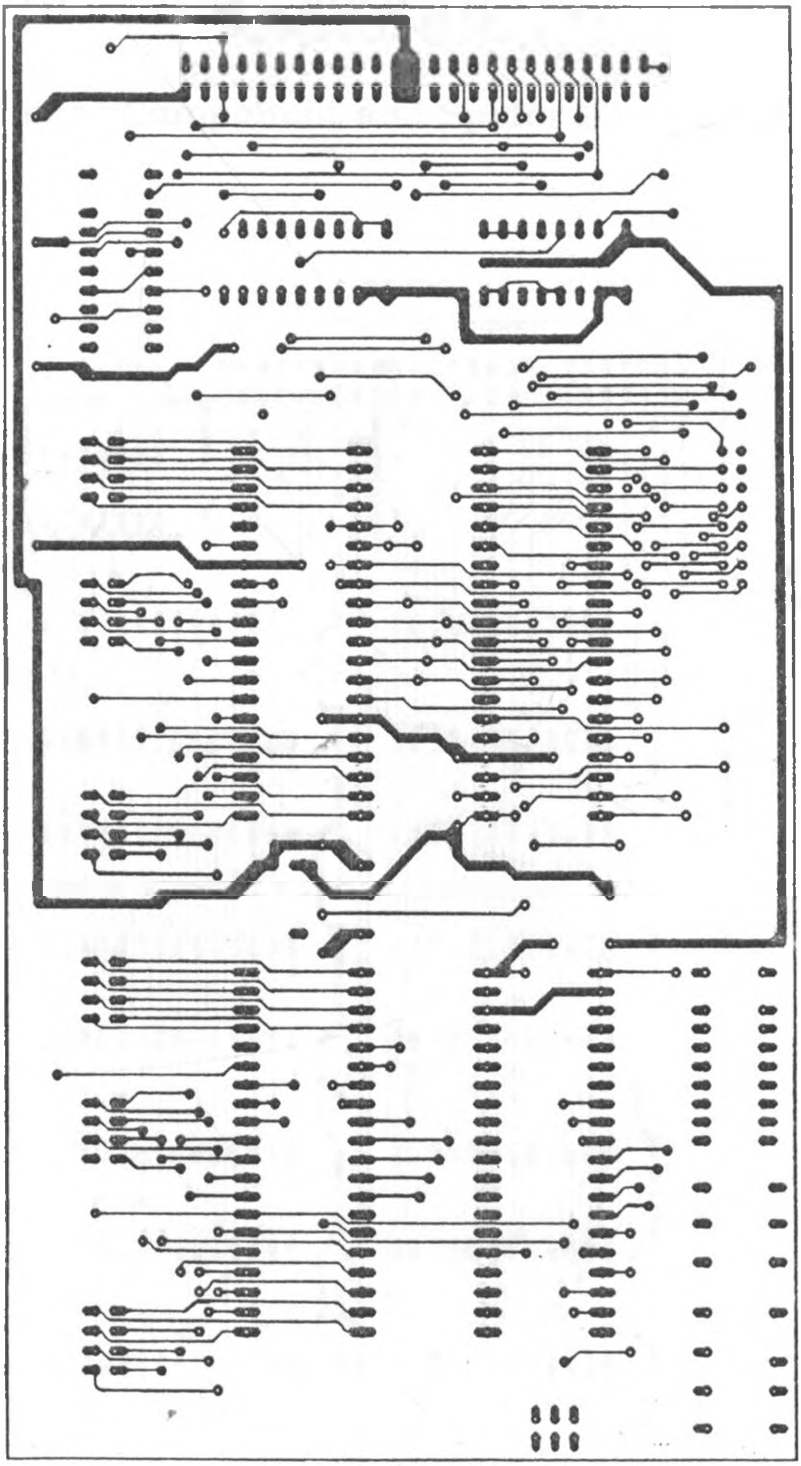
Solder Side

Appendix D6

I/O Board Component and Solder Side PCB Artwork



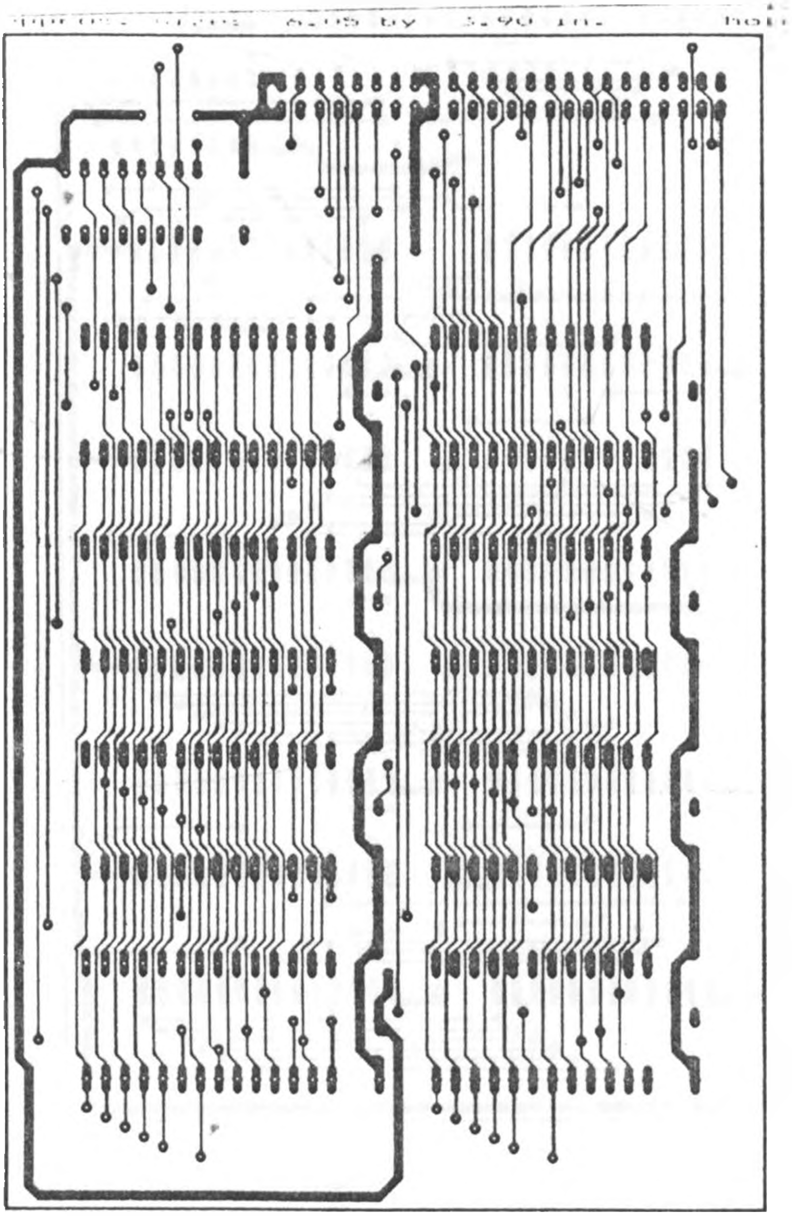
Component Side



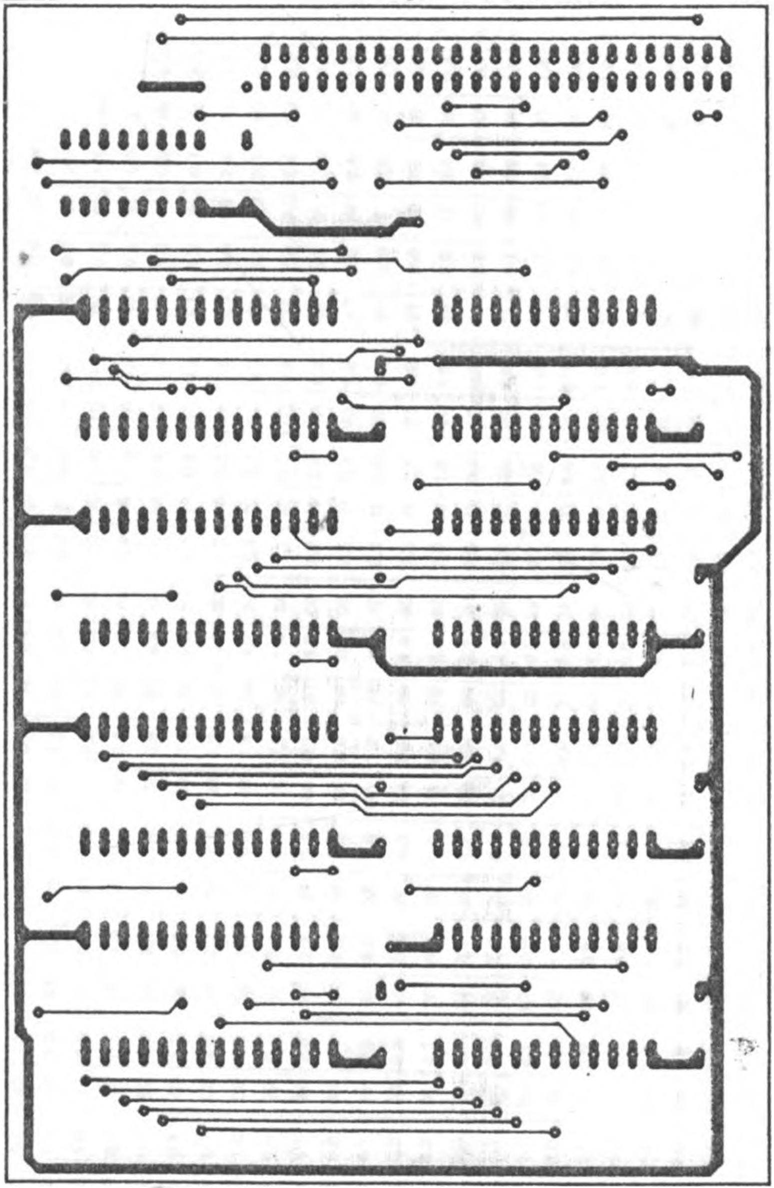
Solder Side

Appendix D7

Ram Board Component and Splder Side PCB Artwork



Component Side



Solder Side

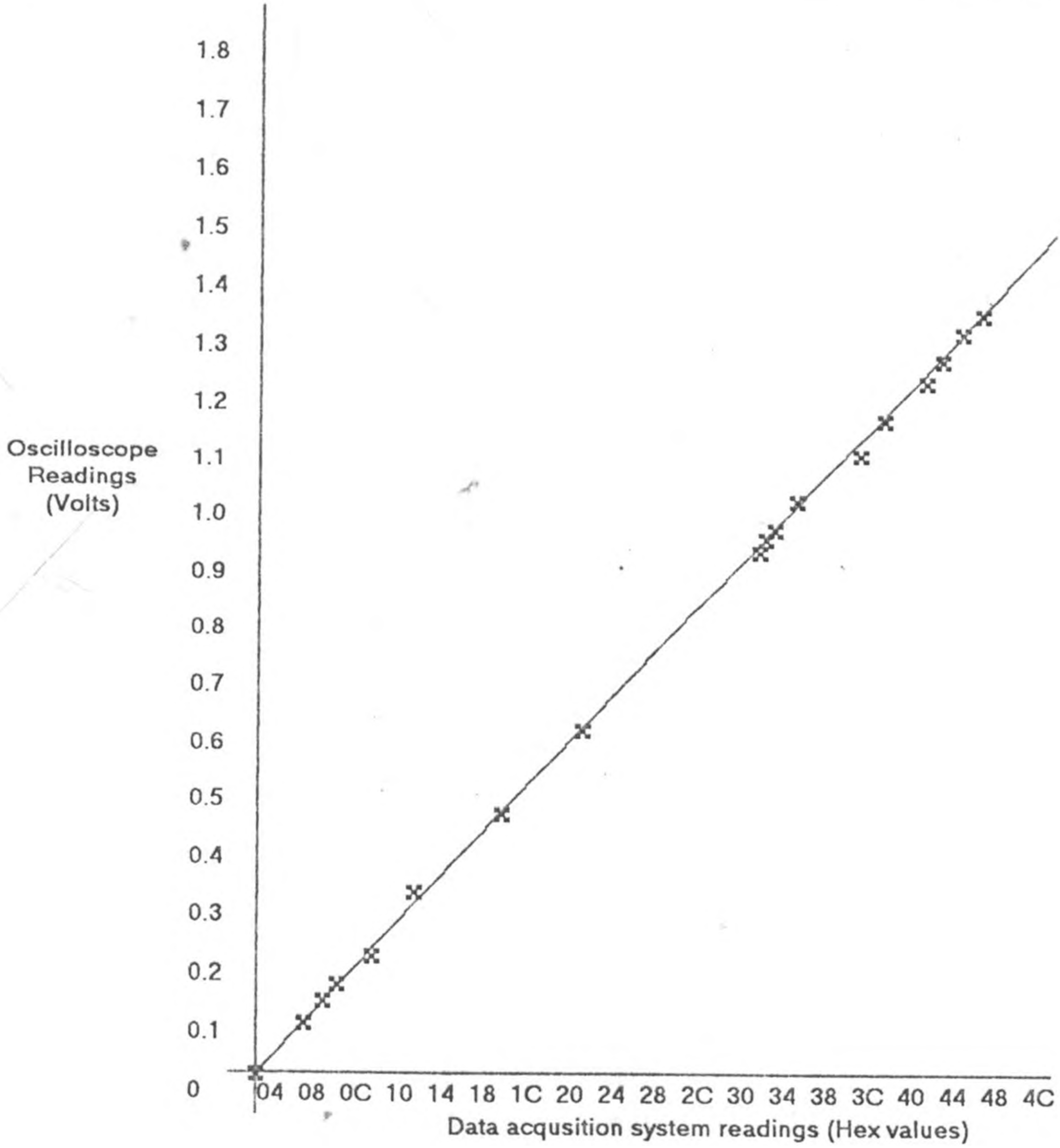
Appendix E1

IPMDAS Laboratory Test Data

Time (Hr.)	CHANNEL																															
	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15	
	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT	HEX	VOLT
0	02	0.0	4E	1.5	39	1.1	83	2.6	1B	0.5	CD	4.0	81	2.5	3B	1.2	08	0.2	3B	1.2	84	2.6	23	0.7	2D	0.9	C0	3.6	6D	2.1	48	1.4
0.5	38	1.1	6D	2.1	77	2.3	D7	4.2	21	0.6	42	1.3	53	1.6	55	1.7	03	0.1	18	0.5	67	2.0	3F	1.2	53	1.6	2D	0.9	0E	0.3	02	0.0
1.0	02	0.0	44	1.3	7A	2.4	0A	0.2	26	0.7	AA	3.3	6A	2.1	52	1.6	3C	1.2	D4	4.5	92	2.9	23	0.7	1B	0.5	1E	0.6	90	2.8	16	0.4
1.5	37	1.1	48	1.4	8C	3.6	73	2.2	35	1.0	42	0.8	71	2.2	4D	1.5	0C	0.2	A3	3.2	35	1.0	3D	1.0	03	0.1	06	0.2	5B	4.8	4B	1.5
2.0	11	0.2	3o	1.4	67	2.0	4C	1.4	16	0.4	18	0.5	5F	1.9	34	1.0	19	0.4	62	1.9	64	1.9	45	1.3	2D	0.9	03	0.1	35	1.0	27	0.8
2.5	23	0.6	4B	1.5	AB	3.4	43	1.3	23	0.7	BC	3.6	9D	3.0	52	1.6	2E	0.9	31	0.8	46	1.4	2A	0.8	53	1.6	4F	1.5	01	0.0	30	0.9
3.0	02	0.0	74	2.3	44	1.4	10	0.3	11	0.3	4F	1.5	09	0.2	37	1.0	20	0.6	1E	0.6	3A	1.1	2A	0.8	47	1.4	F1	4.7	25	0.7	18	0.5
3.5	26	0.7	0C	0.3	08	0.2	9E	3.1	04	0.1	CE	4.0	70	2.2	1E	0.6	3B	1.2	C0	3.8	3D	1.3	3B	1.2	34	1.0	C9	3.9	96	2.9	16	0.4
4.0	32	1.0	50	1.6	2F	0.7	4D	1.5	35	1.0	D8	4.5	54	1.6	12	0.4	4B	1.5	D4	4.1	20	0.6	26	0.7	27	0.7	07	0.1	6D	2.1	0D	0.3
4.5	03	0.0	47	1.4	46	1.4	AB	3.3	0F	0.3	28	0.8	82	2.5	15	0.4	1C	0.5	46	1.4	3A	1.1	26	0.7	0D	0.3	73	2.2	AC	3.4	2D	0.9
5.0	36	1.1	3F	1.2	66	2.0	4B	1.5	3A	1.1	84	2.6	74	2.3	44	1.3	06	0.1	B1	3.5	05	0.2	13	0.4	1A	0.5	28	0.8	71	2.2	16	0.4
5.5	09	0.2	27	0.8	37	1.1	10	0.3	20	0.6	44	1.3	2A	0.8	44	1.3	0F	0.3	03	0.1	90	2.8	10	0.3	4D	1.5	A7	3.3	AF	3.4	4B	1.5
6.0	03	0.0	6E	2.1	8D	2.8	49	1.4	09	0.2	10	0.3	9A	3.0	2A	0.8	41	1.3	A5	3.2	1D	0.6	11	0.3	27	0.8	81	2.5	7	2.5	50	1.6
6.5	14	0.4	49	1.4	64	1.9	0A	0.2	0D	0.3	A9	3.3	15	0.4	0F	0.3	0A	0.2	33	1.0	3C	1.2	48	1.4	07	0.2	31	0.9	82	2.5	07	0.2
7.0	35	1.0	0E	0.3	96	2.9	98	2.9	11	0.3	01	4.1	2F	0.9	0B	0.2	19	0.5	1B	0.5	32	0.9	52	1.6	34	1.0	28	0.5	16	0.4	10	0.6
7.5	04	0.1	0E	0.3	7C	2.4	06	0.1	01	0.0	73	2.2	04	0.1	59	1.7	02	0.0	69	2.1	7F	2.5	24	0.7	1C	0.5	83	2.6	8C	2.8	57	1.7
8.0	15	0.4	1F	0.6	90	2.8	A6	3.2	37	1.1	27	0.8	7A	2.4	2E	0.9	51	1.6	CC	4.0	65	1.9	0C	0.2	36	1.1	04	0.1	66	2.0	3A	1.1
8.5	31	0.9	64	1.9	1C	0.6	29	0.8	22	0.7	6C	2.1	87	2.6	3E	1.2	0A	0.2	02	0.0	25	0.7	51	1.6	33	1.0	CD	0.3	AD	3.4	47	1.4
9.0	27	0.8	0A	0.2	1B	0.5	91	2.8	02	0.0	1D	0.6	4C	1.5	15	0.4	31	0.9	31	0.9	18	0.5	2D	0.6	26	0.7	CE	4.0	94	2.9	16	0.4
9.5	2B	0.9	28	0.8	8C	2.7	57	1.7	3A	1.1	0F	0.3	0E	0.3	48	1.4	10	0.3	59	1.7	94	2.9	2B	0.8	50	1.6	63	1.9	07	0.2	62	0.9
10.0	0A	0.2	24	0.7	89	3.3	46	1.4	30	0.4	01	0.0	1D	0.6	47	1.4	3F	1.2	01	0.0	A4	3.2	24	0.7	11	0.3	13	0.4	49	1.4	14	0.4
10.5	1D	0.5	24	0.7	A9	0.6	25	1.7	2F	0.9	68	2.0	87	2.6	3D	1.2	22	0.7	E7	4.5	5F	1.9	50	1.6	33	1.0	5D	1.8	7E	2.5	30	0.9
11.0	0E	0.2	09	0.2	21	0.4	C3	3.8	01	0.0	D8	4.2	3E	1.2	4A	1.4	15	0.4	55	1.7	01	0.0	40	1.3	3A	1.1	39	1.1	4D	1.5	3B	1.1
11.5	04	0.0	72	2.2	14	0.3	86	2.7	0D	0.3	86	2.6	57	1.7	28	0.5	3A	1.1	79	2.3	03	0.1	23	0.7	29	3.8	C4	0.8	A9	3.3	26	0.7
12.0	03	0.0	2F	0.9	B5	3.5	E0	4.4	07	0.1	A4	3.2	57	1.7	45	1.3	2F	0.9	D4	4.4	64	2.0	54	1.6	4D	4.8	DA	1.5	26	0.7	21	0.6

Appendix E2

Channel 0 Test Data Graph



The straight line indicates that the readings acquired byu the instrument agree well with those obtained manually for most points.

Appendix F1

System Schematic Diagrams

UNIVERSITY OF NAIROBI
CHIROMO LIBRARY