DESIGN OF A MICROPROCESSOR BASED INSTRUMENTATION MODULE FOR SIGNAL PROCESSING APPLICATIONS

by

ASHOK RAMACHANDRAN

B.S., College of Engineering, Guindy, Madras, India, 1981

A MASTER'S THESIS

Submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1984

Approved by

M&P Lucas

Major Professor

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

II. DESIGN OF THE OPTOPOD 3
2.1 The need for a fast data acquisition system 3
2.2 Hardware requirements of the Optopod 4
2.3 Basic operation of the Optopod using NSC 800 CPU ... 6
2.4 NSC 800 Microprocessor 7
2.5 Analog to Digital Converter 23
2.6 Flash Analog to Digital converter 24
2.7 Optical Fiber Communication 28
2.8 Theory of Optical Fibers 29
2.9 Reflection and Refraction at a Dielectric Interface. 30
2.10 NSC 858 - UART 32
2.11 Limitations of the System 33

III. SOFTWARE DESCRIPTION 35

IV. HARDWARE DESCRIPTION FOR FLASH ADC MODULE USING 8751 39
4.1 Choice of a suitable Processor 39
4.2 Intel-8751 Microcomputer 39
4.3 Memory organisation of the 8751 41
4.4 Programming the On-board EPROM of 8751 44
4.5 Direct Memory Access 45
4.6 Counter technique to implement DMA 46
4.7 Sampling frequency control 54
4.8 Phase Locked Loop 54
4.9 Auto-ranging of the Flash ADC 66
4.10 RS-232C Standard 68

V. 5.1 SOFTWARE DESCRIPTION FOR THE 8751 70
5.2 Designing the 8751 Monitor Program 71
5.3 Serial Port and Timer configuration 73

VI. 6.1 EVALUATION OF THE TWO DESIGNS OF THE TEST MODULE ... 75
6.2 System implementation for Phase Measurement 76
6.3 Development of a new Instrumentation System 78

VII. CONCLUSION 80

APPENDICES.

APPENDIX A. Board layout and Power consumption details of the System 82

APPENDIX B. Calculation of Current limiting resistors for Optic Fiber Cable 83

APPENDIX C. BASIC Program from Control Computer to invoke the Optopod 84

APPENDIX D. Monitor program Listing in Z-80 Code for the Optopod 85

APPENDIX E. BASIC Program to invoke the Auto-ranging test module 87

APPENDIX F. Flowchart for the monitor program of 8751 88

APPENDIX G. Monitor Program listing in mnemonic code for the 8751 System 103

APPENDIX H. Calculation of Signal conditioning circuitry components for Auto-ranging the Flash ADC 112

APPENDIX I. Components list for the module using 8751 CPU ...116

BIBLIOGRAPHY 118

ACKNOWLEDGEMENTS 119

LIST OF FIGURES

Figures		Page
1	Basic block diagram of a data acquisition system	2
2	Block diagram of the existing Automated test system .	3
3	Block schematic of an Optopod based Instrumentation System	6
4	Demultiplexing the NSC 800 bus	7
5	Memory Map of the NSC 800	9
6	Block schematic of the Optopod using the NSC 800	10
7	Schematic diagrams	
	7.a Showing the CPU, Address/Data demultiplexing and device select logic	12
	7.b Showing Memory devices and Flash ADC connected to NSC 800 Bus	13
	7.c Showing Serial Communication using Optical Fibers	14
8	8.a.1,8.a.2,8.b.1,8.b.2 Printed Circuit board layouts for the NSC 800 System	16
9	Timing Diagrams	
	9.1 Read timing diagram of the NSC 800 with HM 6116 RAM	20
	9.2 Write timing diagram of the NSC 800 with HM 6116 RAM	21
	9.3 Read timing diagram of the NSC 800 with NMC 27C32 EPROM	22
10	Theoretical transfer function of an ADC	23
11	Schematic diagram of an 8-bit Flash ADC	25
12	Read timing diagram of the NSC 800 with CA 3308 Flash ADC	27
13	A basic Optical fiber system	28
14	Light rays at a dielectric interface	30
15	Device block diagram of the Optic fiber link	32

16	Flowcharts 16.a For Optopod to receive request from Computer (Interface) and collect 2K samples of the signal .. 16.b Showing stored samples of signal transmitted to Computer(Interface)	36 37
17	Input/Output lines of the 8751	41
18	Memory Map of the 8751	42
19	EPROM Programming configuration of the 8751	44
20	Timing diagram of data transfer from CA 3308 Flash ADC to the HM 6116 RAM	48
21	Block schematic showing the data transfer from Flash ADC into memory through DMA	50
22	Read timing diagram for the 8751 with the HM 6116 RAM	53
23	A Phase locked loop	55
24	A practical Phase locked loop that could be implemented in a microcomputer based system	56
25	Block diagram of the programmable PLL MC 145143	58
26	Block schematic for frequency control through 8751 ..	60
27	Configuration of 74LS161 as a programmable divider ..	61
28.a,b	Printed circuit board layout for the 8751 system	64
29	Diagram to show the need for auto-ranging a Flash ADC	66
30	Signal conditioning circuitry for Auto-ranging an ADC	67
31	RS-232C Interface circuit for the 8751	69
32	32.a System implementation for Phase measurement .... 32.b Zero crossing detection technique	77 77
33	Block diagram for developing a new Instrumentation system using Optical fibers	79

CHAPTER-1

INTRODUCTION

The purpose of this report is to discuss two designs for a high speed data acquisition unit that could be used for Digital Signal-Processing applications. In contemporary instrumentation, instruments are normally in large boxes, rack-mounted or plugged into a motherboard similar to the Hewlett-Packard 6940B Multiprogrammer. The Multiprogrammer houses several hardware modules such as an Analog to Digital Converter, Voltage to Frequency Converter and a Voltage Controlled Oscillator apart from the relay cards and power supply unit.

The basic idea of this work was to evaluate the feasibility of constructing special purpose test modules that could be placed in close proximity to the circuit under test. One particularly useful test module would contain an auto-ranging Flash Analog to Digital converter. This module could be used for specific applications such as the measurement of phase using signal processing techniques. Typical algorithms would include the Adaptive filtering, Fast Fourier Transform, Correlation and numerical methods.

The basic block diagram of a modular test system would look as shown in Fig 1.

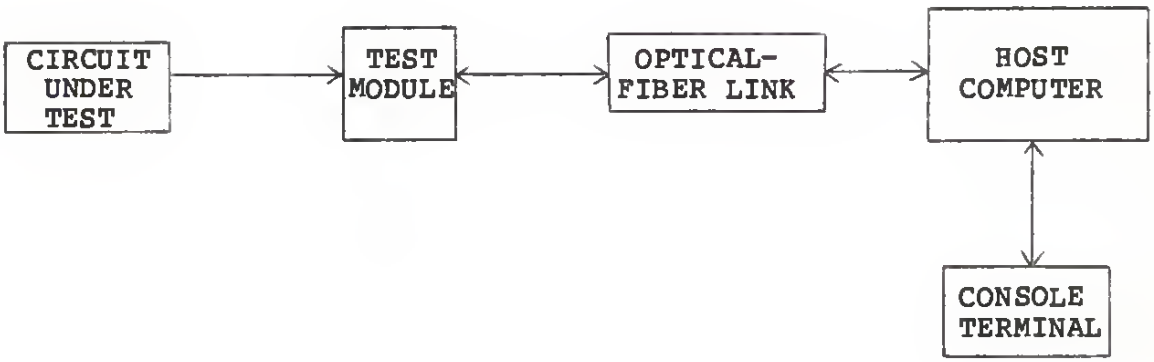


Fig. 1

BLOCK DIAGRAM OF A MODULAR TEST SYSTEM

The development of such Microcomputer controlled test pods would lead to instrumentation-systems with distributed intelligence. The communication between the pod and host-computer by Optical fibers would, in turn lead to faster measurements. Compared to the flat wire cables conventionally used for data links the Optical fibers would help in making measurements even in electrically noisy environments.

CHAPTER-2

DESIGN OF THE OPTOPOD

2.1 THE NEED FOR A FAST DATA ACQUISITION SYSTEM.

In recent times Computer aided instrumentation has been extensively used for the measurement of responses in electrical networks. In one existing system in the instrumentation laboratory, Electrical Engineering Department, Kansas State University a Hewlett-Packard Multiprogrammer has been connected to a Northstar Microcomputer through a M6802 interface for data acquisition. The basic block diagram of the existing system is shown in Fig.2.

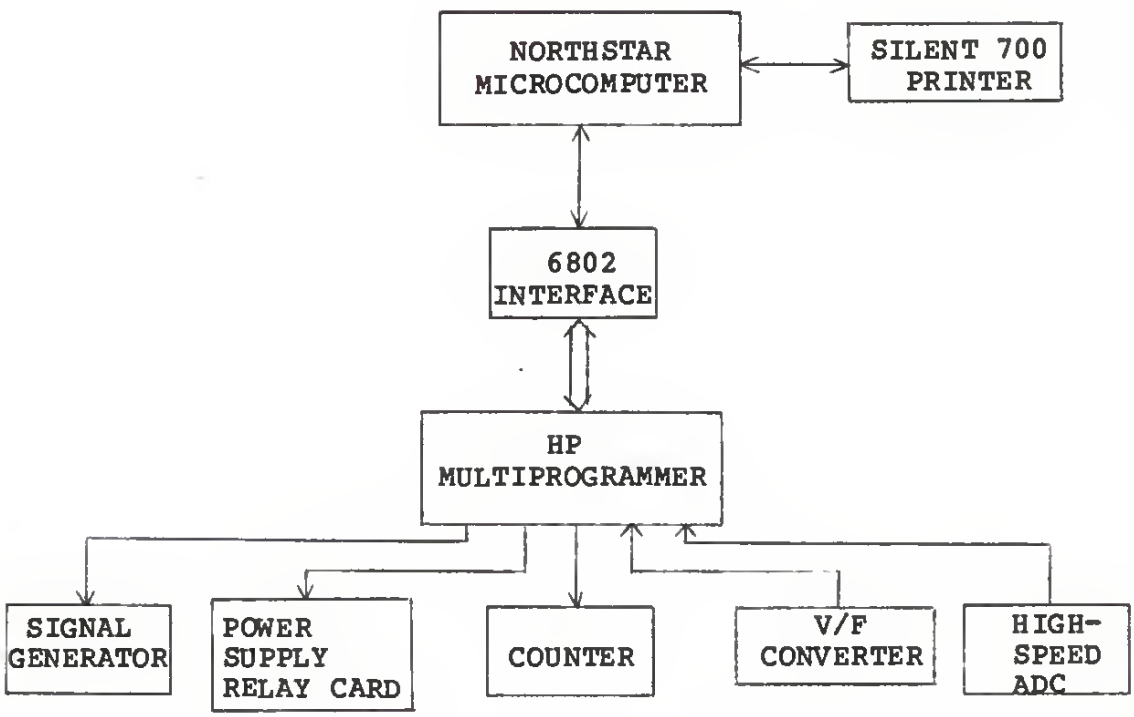


Fig. 2

BLOCK DIAGRAM OF THE EXISTING AUTOMATED TEST SYSTEM

The HP Multiprogrammer has a variety of separate input cards such as the ADC,V/F and VCO for frequency and voltage measurements. The Multiprogrammer returns 12 bit binary dataupon request from the Northstar Microcomputer through the M6802 interface. The MC 6802 interface serves as the communication link between the Northstar and the HP Multiprogrammer. It converts the 12 bit binary data from the HP Multiprogrammer to ASCII code and sends it serially to the Northstar through an RS-232 link.

There is a high speed ADC card in the Multiprogrammer that monitors bipolar dc levels in ranges upto $\pm$ 100V and returns a 12 bit 2's complement digital word to the 6802 to indicate the magnitude and sign of measured voltage. Although the minimum conversion time for the ADC is 50 μsecs the actual conversion rate is 9.12 K samples per sec due to the program controlled data acquisition of the samples. This system has been found to be slow due to motherboard nature of the input cards and also because of long cable connections. Noise and DC offset problems have also been encountered; so it was decided to design a special-purpose test-module for an Auto-ranging Flash ADC application which would perform better as a dedicated system.

2.2 HARDWARE REQUIREMENTS OF THE POD.

The design of the Pod was made after careful consideration as to its requirements. The basic considerations that were taken into account were :

1) To design a high speed data acquisition unit which consumes low power,

2) To digitize an input signal upto a maximum sampling rate of 8 MHz,

3) To use Optical fibers for communication between the Pod and the Computer instead of the RS-232 link so that the data transfer rate could be increased up to 1 M Baud,

4) To minimize the board size of the pod as much as possible so that it could be placed next to the circuit under test for accurate measurements, and

5). To automatically set the range of the Flash ADC with respect to the amplitude of signal and sample the signal at a frequency as required by the user.

It was decided to use a Flash ADC that would convert the input-analog signal to an 8 bit digital word in one clock-cycle. The first design was made using the NSC 800 Microprocessor to achieve low power consumption. Due to the constraints of processor capability and space limitations only three requirements could be fulfilled. Hence the second design was made using the Intel- 8751 Microcomputer which could handle all the requirements.

Since the Intel 8751 serial interface could transfer at a maximum baud rate of only 666.6 K baud it was decided to use the RS-232 link at 9600 Baud for design purpose. Further improvement in speed up to 1 M Baud would then lead to using the optical fiber cable for this design too. The limitations of this design and the improvements that are made in the system designed using the 8751 are discussed in later chapters. A comparative study of the two designs is also given.

2.3 BASIC OPERATION OF THE OPTOPOD USING THE NSC 800 CPU.

The Optopod would act as an intelligent terminal and would operate upon request from the host computer. The Pod would sample the input signal at a fixed sampling frequency under the control of the NSC 800 Microprocessor and store data from the ADC into a scratch pad memory in successive locations upto 2 K bytes. After acquiring a block of data in memory it is sent serially to the host computer using optical fiber cables. Further processing of samples of the signal could then be done using the computer. Since the optical link communicates at a data rate of 1 M baud it would be necessary to send data from the pod to an interfacing processor which would then send data to the parallel ports of the computer. The basic block diagram of the optopod would look as shown in Fig.3.

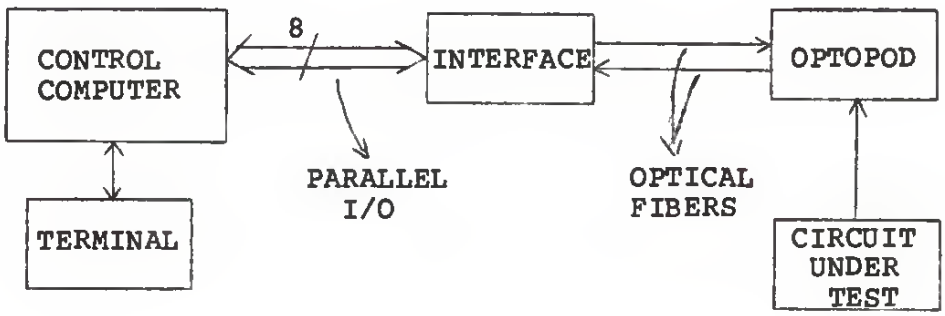


Fig. 3

BLOCK SCHEMATIC OF AN OPTOPOD BASED INSTRUMENTATION SYSTEM

2.4 NSC 800 MICROPROCESSOR.

The NSC 800 is an 8 bit low power Microprocessor fabricated using P2CMOS technology. It features low power consumption; typically 50 mW at 5 V Vcc, a variable power supply of 2.4-6V, the powerful Z-80 compatible instruction set, and a 1 μsec instruction cycle time with a 4 MHZ clock. It is capable of addressing 64 K bytes of memory, 256 I/O devices, has 10 addressing modes, and 22 internal registers. The NSC 800 has a multiplexed bus structure with the lower order address lines and the data bus being multiplexed. So de-multiplexing and buffering of data and address bus are done using a bi-directional transceiver and an Input/Output port respectively. Control signals ALE and $\overline{RD}$ are used to latch the address and data respectively into these devices. On the falling edge of ALE address A0-A7 is latched to the 82PC12 which serves as an output port. The bidirectional transceiver 82PC08 transmits data into memory on a write cycle and receives data on a read cycle. The $\overline{RD}$ of the NSC 800 is connected to the T/$\overline{R}$ of the transceiver. The demultiplexing of the address and data bus is shown in the Fig.4.

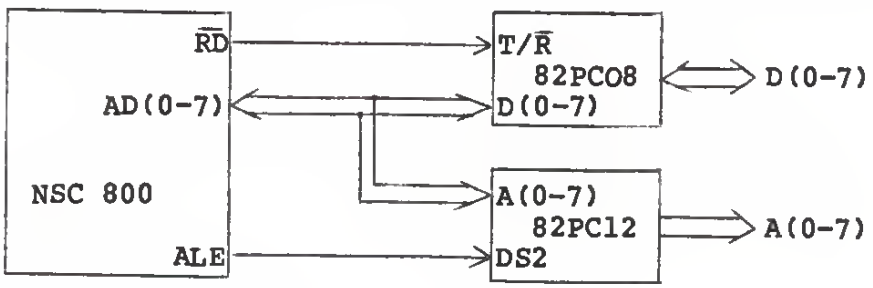


Fig. 4

DEMULTIPLEXING OF THE NSC 800 BUS

With the NSC 800 running at 4 MHZ the basic-hardware modules that are required for the construction of the Optopod are:

1) An HM6116 (2K*8) RAM to store sampled data from the ADC,

2) An NMC 27C32 (4K*8) EPROM to store the monitor program,

3) A CA 3308, 8 bit Flash ADC which samples the input signal and converts into digital word within one clock cycle,

4) An NSC 858 UART to transmit and receive data to and from the interface at a data rate of 1 M Baud, and

5) An HFBR 1501/2501 Optical fiber cable of 5 meter length which could communicate optically through the transmitter and receiver coupler.

The block diagram of the Optopod is shown in Fig 6. It should be noted that all devices listed above are fabricated using CMOS technology and would contribute to low power consumption of the overall system. All devices are memory mapped in the 64 K memory space of the NSC 800 and address space of each device is shown in the memory map in Fig.5. It should be noted that on resetting the CPU clears the Program counter to X'0000 and the program execution starts at memory location zero. Hence the monitor program which resides in the EPROM starts at location zero instead of the conventional way of putting the EPROM on the top of memory.

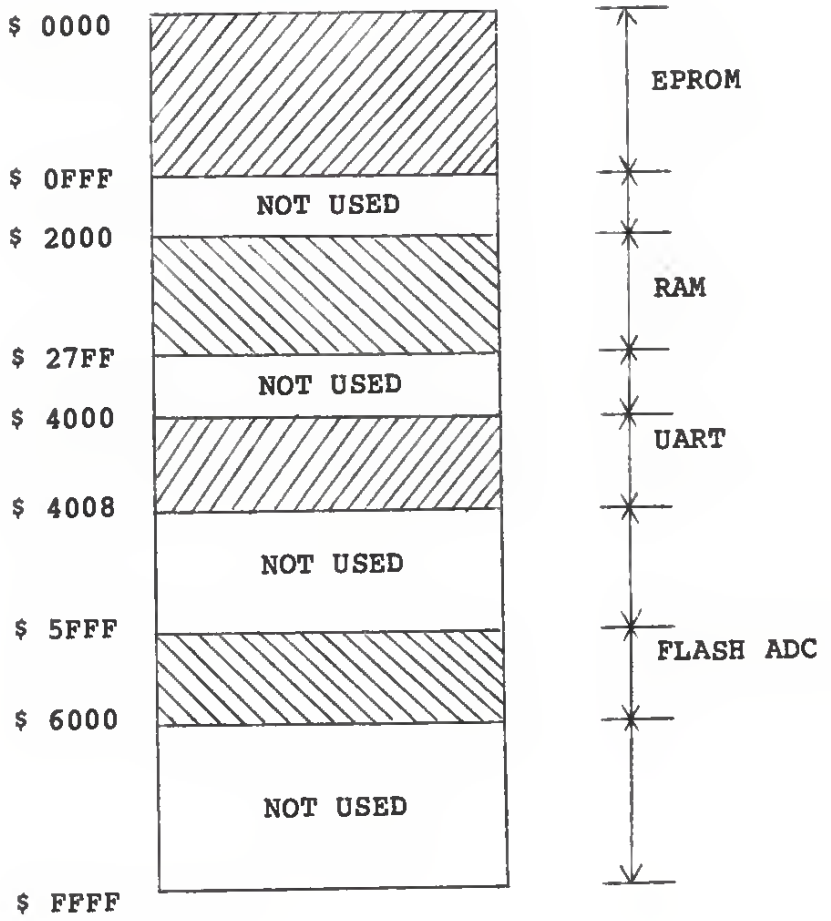


Fig. 5

MEMORY MAP OF THE NSC 800

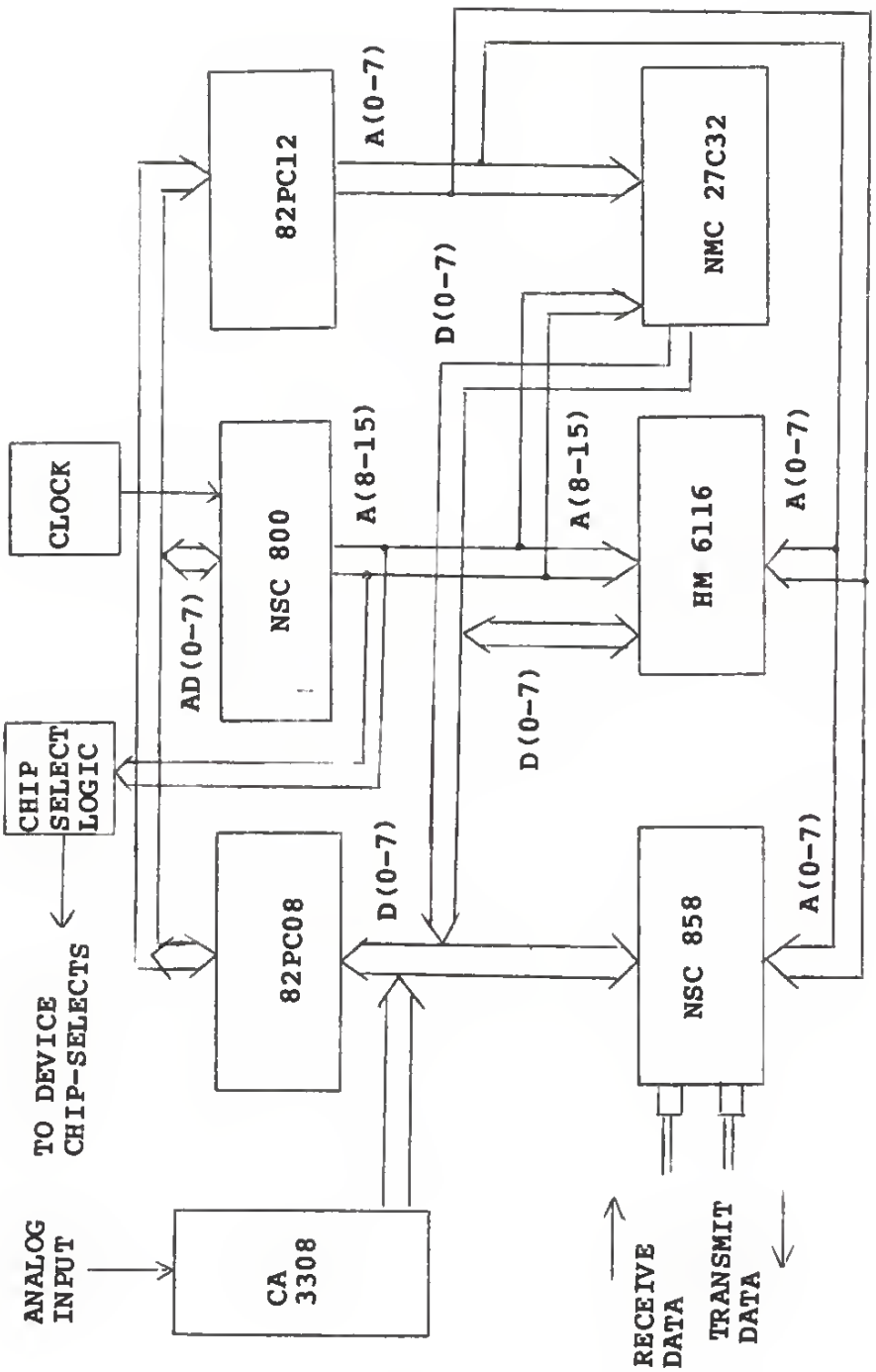


Fig. 6

BLOCK DIAGRAM OF THE OPTOPOD USING THE NSC 800

The address decoding for the various devices is done using the higher order address lines A15, A14, and A13. Hex-inverter and 3-Input Nand Gates are used .Two Control latches, Flip-Flops with light-emitting diodes are provided to alert the user as to the operational state of the CPU.

The schematics for the system interconnections are shown in Fig 7.a, 7.b and 7.c.

After designing the system, quite a bit of time was spent drawing different printed circuit board layouts. At first it was thought that the pod would only occupy a 5 inch by 3 inch board. But even with the very minimal hardware involved it was found that this estimate was much too low. The first layout was found to be double the size of what was expected . This was possibly due to the complicated bus structure of the NSC 800 which required the lower order address and data to be latched into buffers and also the circuitry for the serial interface consumed quite a bit of space. Improvements on the layouts were made by trying to efficiently place the circuitry in such a way so that the total area of the IC chips would be almost equal to the board size area. After trying different layouts it was found that board size could not be reduced any further and based on that decision it was decided to lay the circuit in two boards which could be interconnected using board connectors. By this way the effective board area could be brought down to what was originally expected. Details of the board size and effective area that could be used by the circuitry are given in Appendix A.

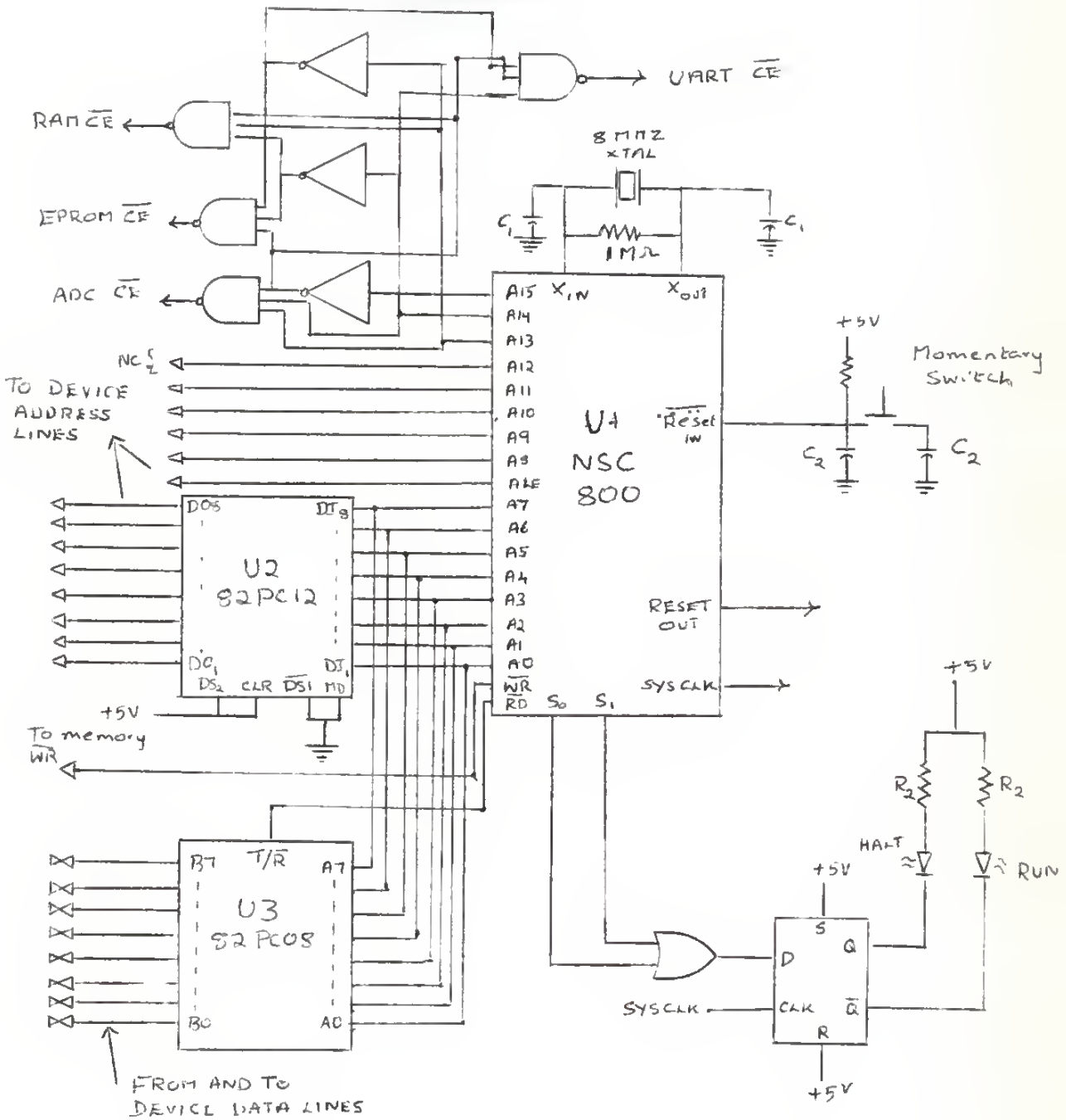


Fig. 7.a SCHEMATIC DIAGRAM SHOWING CPU, ADDRESS/DATA DEMULTIPLEXING AND DEVICE SELECT LOGIC

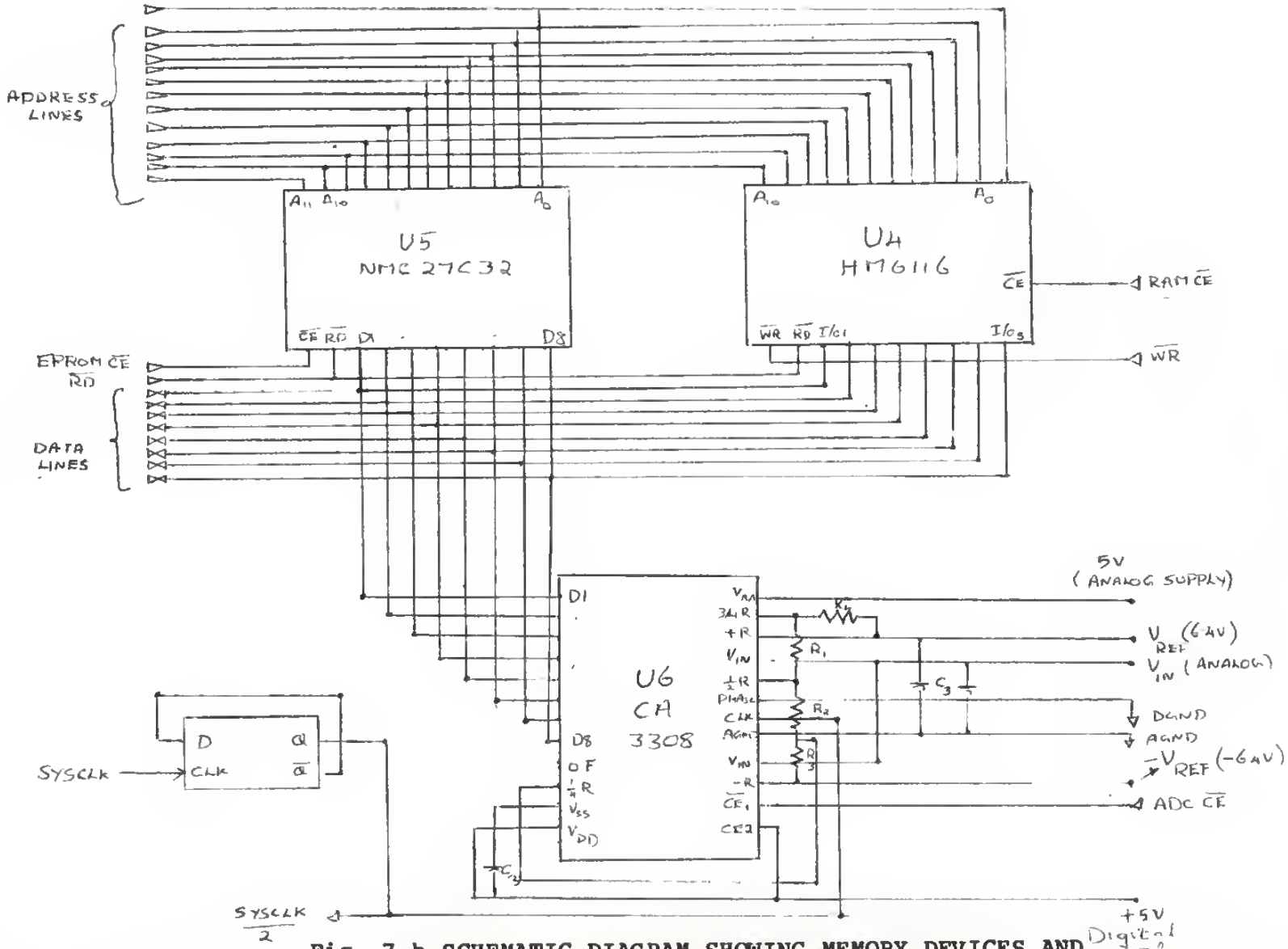


Fig. 7.b SCHEMATIC DIAGRAM SHOWING MEMORY DEVICES AND FLASH ADC AS CONNECTED TO THE NSC 800 BUS

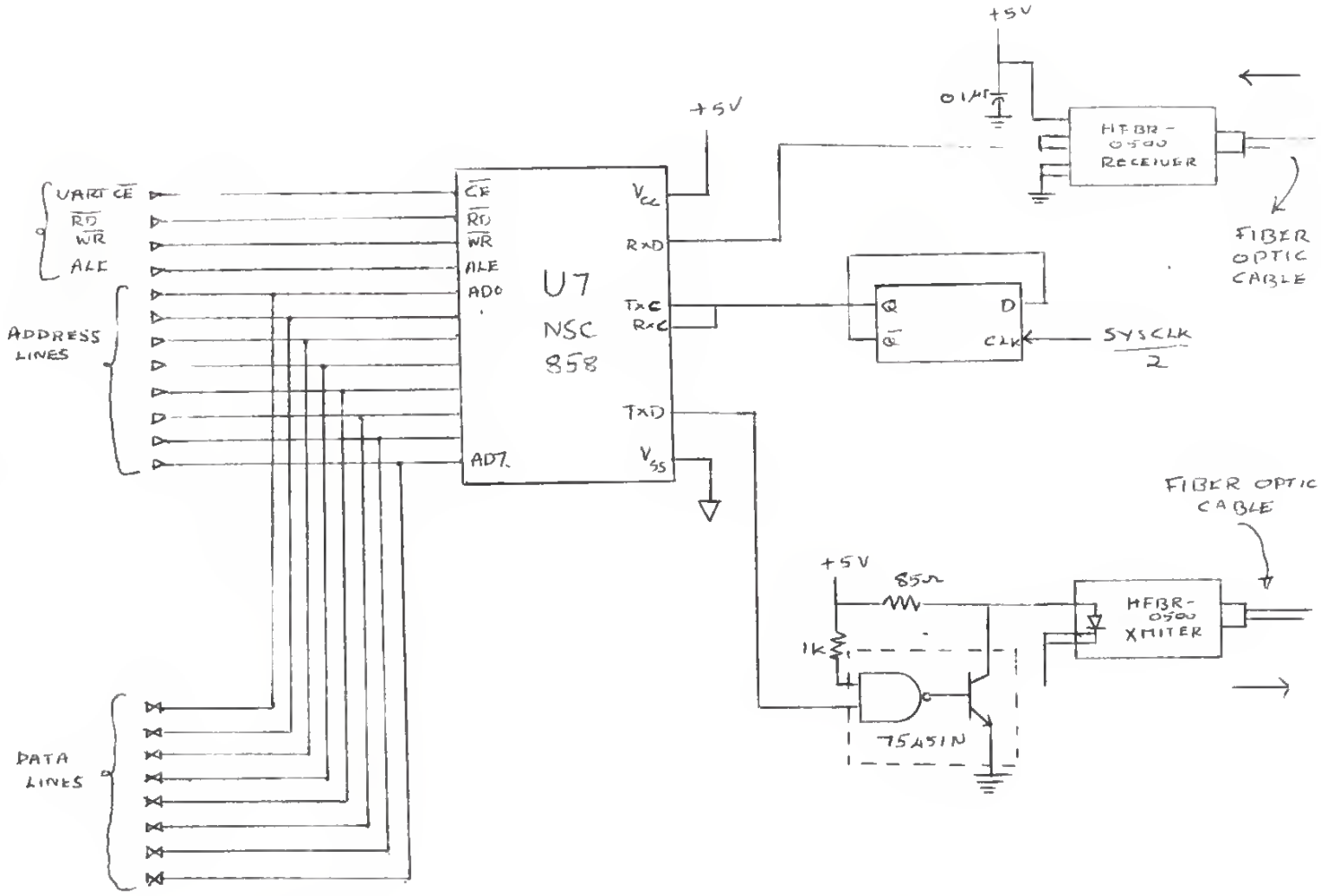


Fig. 7.c SCHEMATIC DIAGRAM SHOWING SERIAL COMMUNICATION USING OPTICAL FIBERS

The printed circuit board layouts for the two interconnected boards of the Optopod are shown in Figs. 8.a.1, 8.a.2 , 8.b.1 and 8.b.2.

Before deciding on microprocessor and memory devices a thorough study of the timing diagrams for the NSC 800 for the read and write cycle with the HM6116 RAM and the read cycle with NMC 27C32 EPROM was made and it was found that these memory devices would be compatible with the NSC 800 at a clock frequency of 4 MHZ. The read and write timing diagrams for HM6116 and the read timing diagrams for the NMC27C32 EPROM are shown in Figs 9.a, 9.b and 9.c respectively. It was found that the HM6116 RAM would be compatible with the NSC 800 while the EPROM specification sheet specifies a maximum access time of 450 nsecs. From the read timing diagram it was found that maximum access time required by the microprocessor was around 375 nsecs. So it was assumed that this memory could be accessed around 300-350 nsecs. A memory of faster access time say 200-300 nsecs is however recommended.

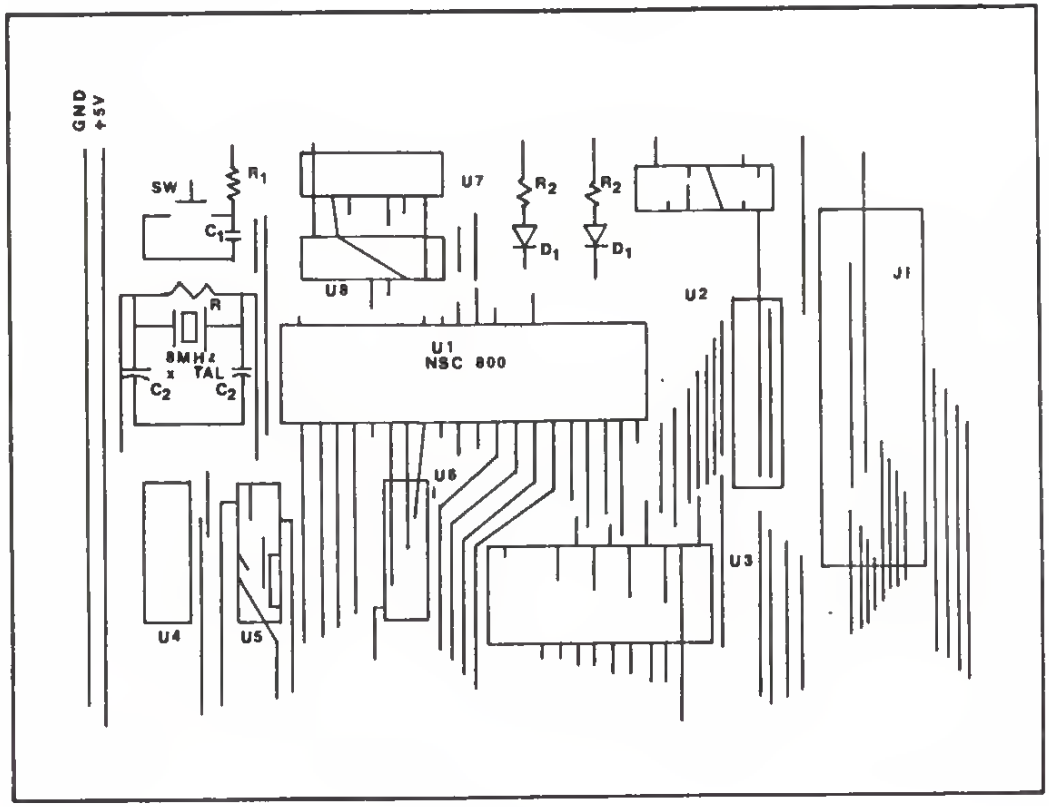


Fig. 8.a.1

PRINTED CIRCUIT BOARD LAYOUT FOR THE NSC 800 BASED SYSTEM
(TOP BOARD- COMPONENT SIDE)

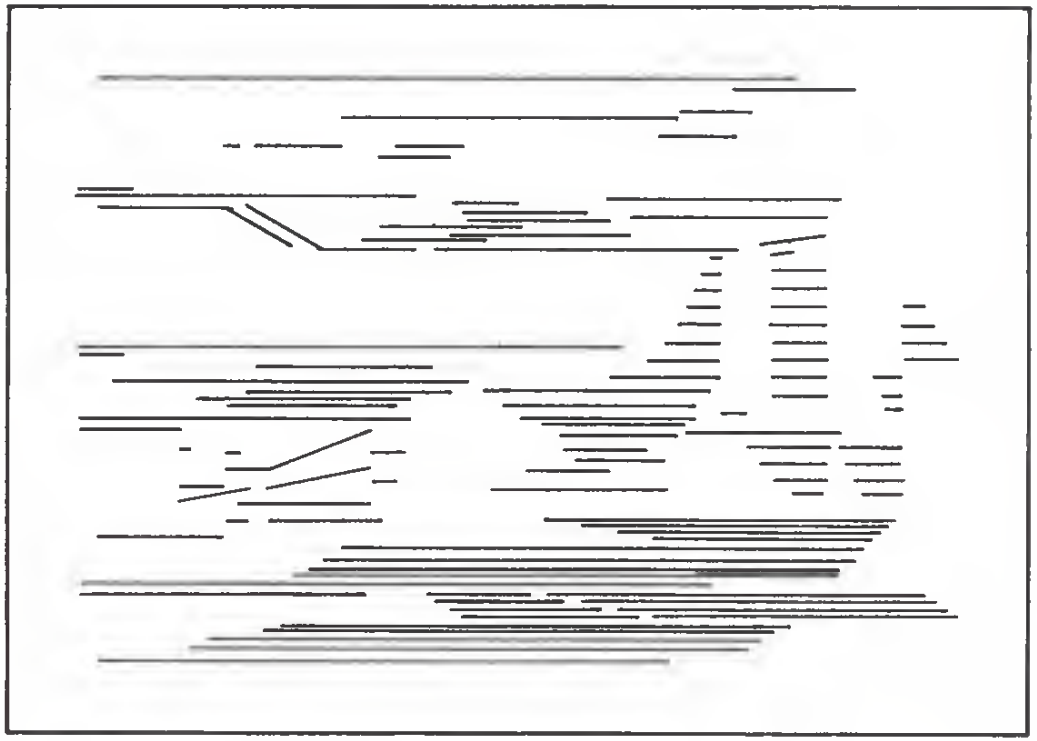


Fig. 8.a.2

PRINTED CIRCUIT BOARD LAYOUT FOR THE NSC 800 BASED SYSTEM
(TOP BOARD- CIRCUIT SIDE)

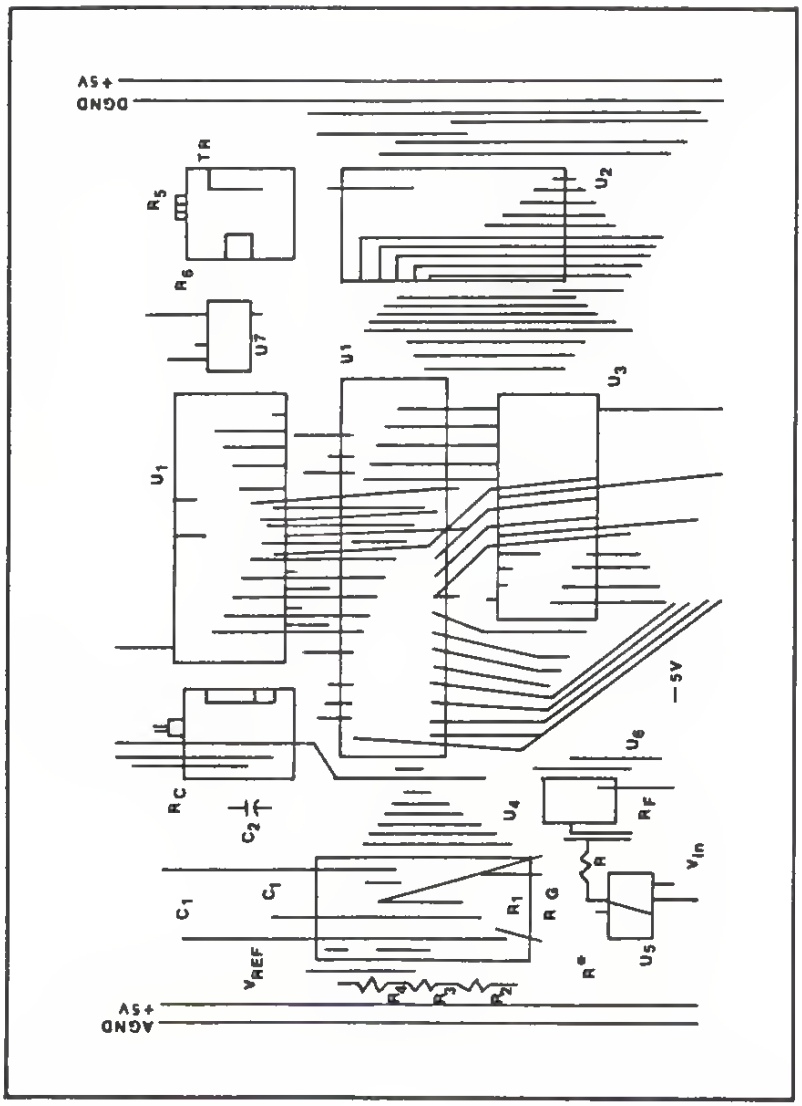


Fig. 8.b.1

PRINTED CIRCUIT BOARD LAYOUT OF THE NSC 800 BASED SYSTEM
(BOTTOM BOARD- COMPONENT SIDE)

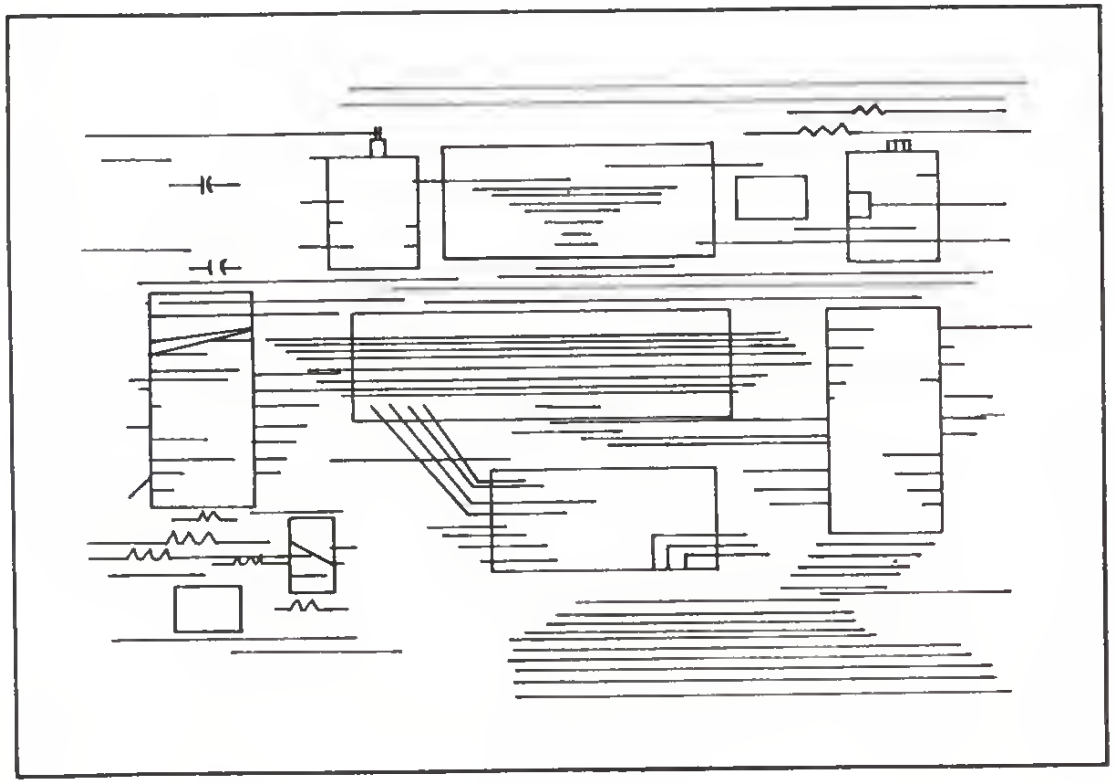


Fig. 8.b.2

PRINTED CIRCUIT BOARD LAYOUT OF THE NSC 800 BASED SYSTEM
(BOTTOM BOARD- CIRCUIT SIDE)

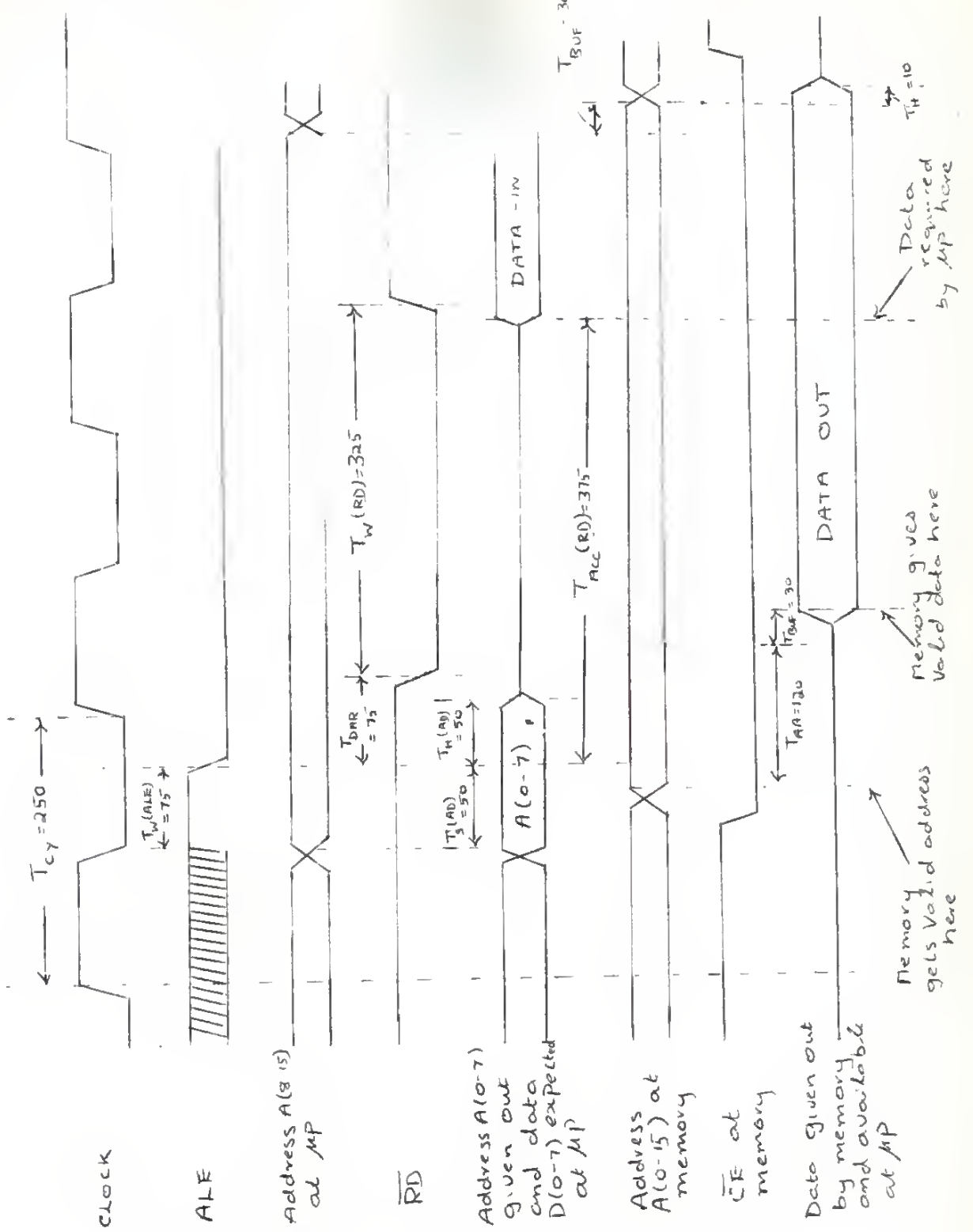


Fig. 9.a READ TIMING DIAGRAM OF THE NSC 800 WITH THE HM 6116 RAM

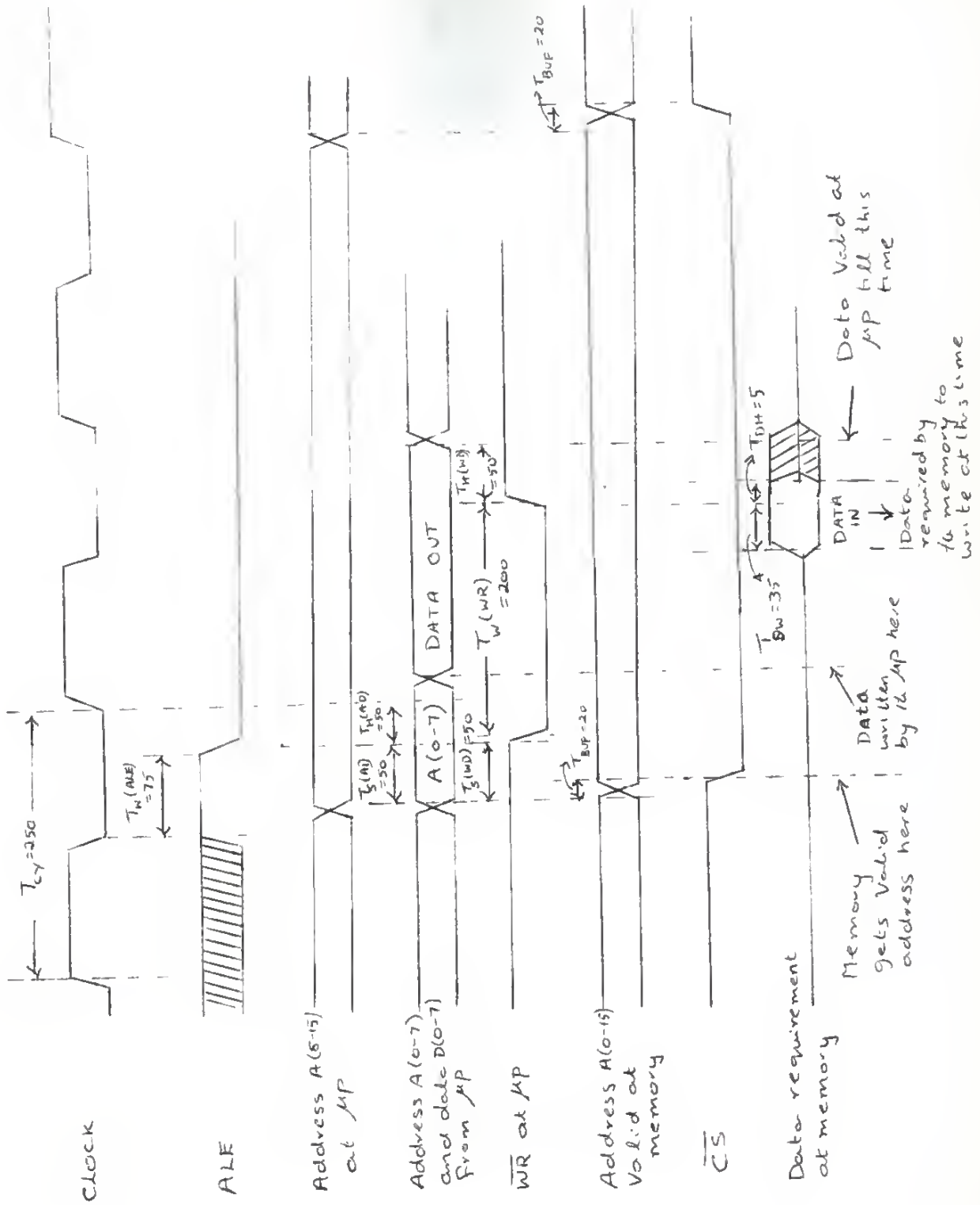


Fig. 9.b WRITE TIMING DIAGRAM OF THE NSC 800 WITH THE HM 6116 RAM

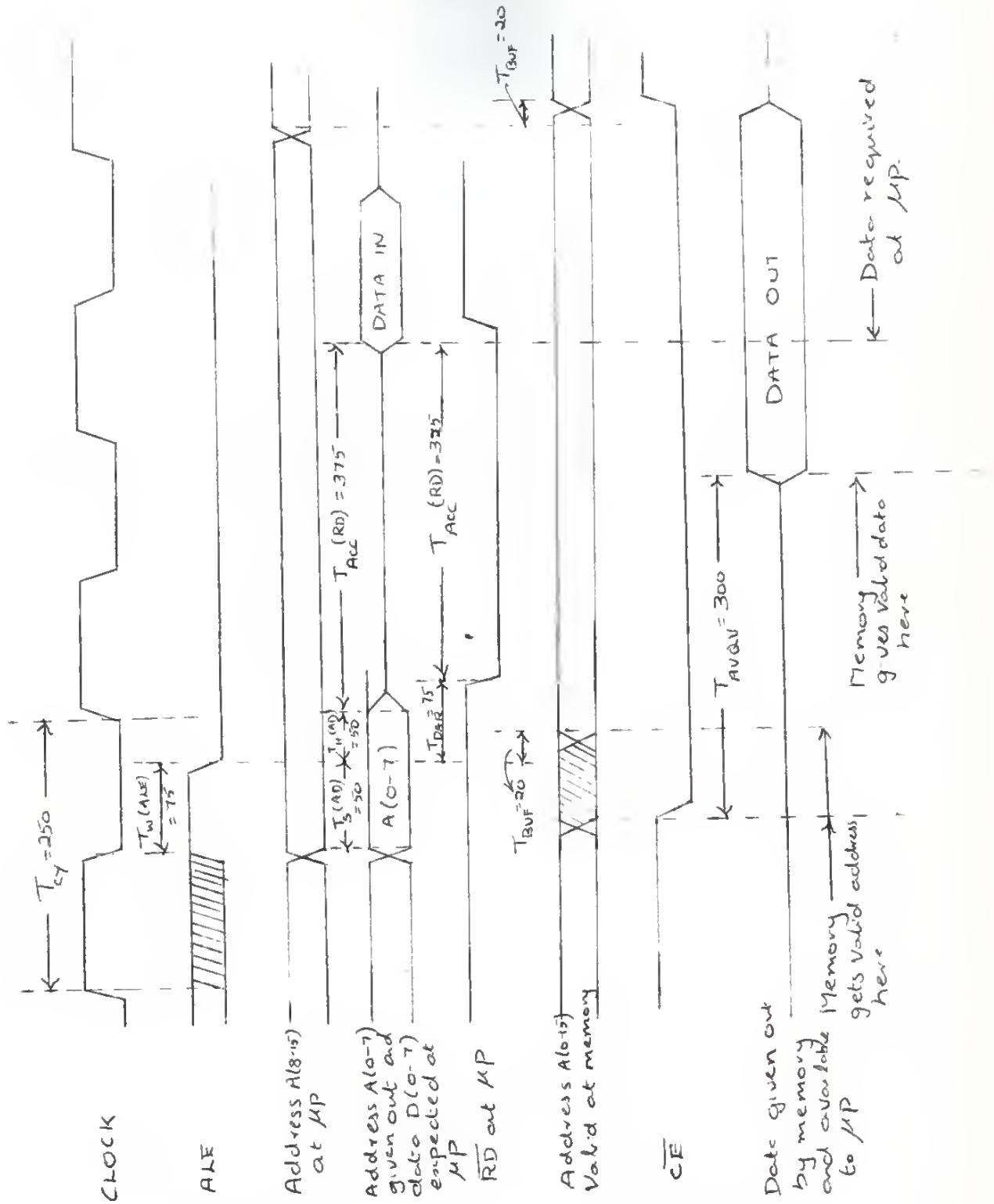


Fig. 9.c READ TIMING DIAGRAM OF THE NSC 800 WITH THE NMC 27C32 EPROM

2.5 ANALOG TO DIGITAL CONVERTER.

An Analog to digital converter or ADC is a device that accepts an analog input signal (which may have any value falling between minimum and maximum of the rated input range) and generates a digital output signal (a coded set of ONE/ZERO levels). The output of an ADC is ideally the precise digital representation of the analog input to within ± 0.5 LSB which is known as the quantization error. The theoretical transfer function of an ADC with first three least significant bits only is shown in Fig.10.

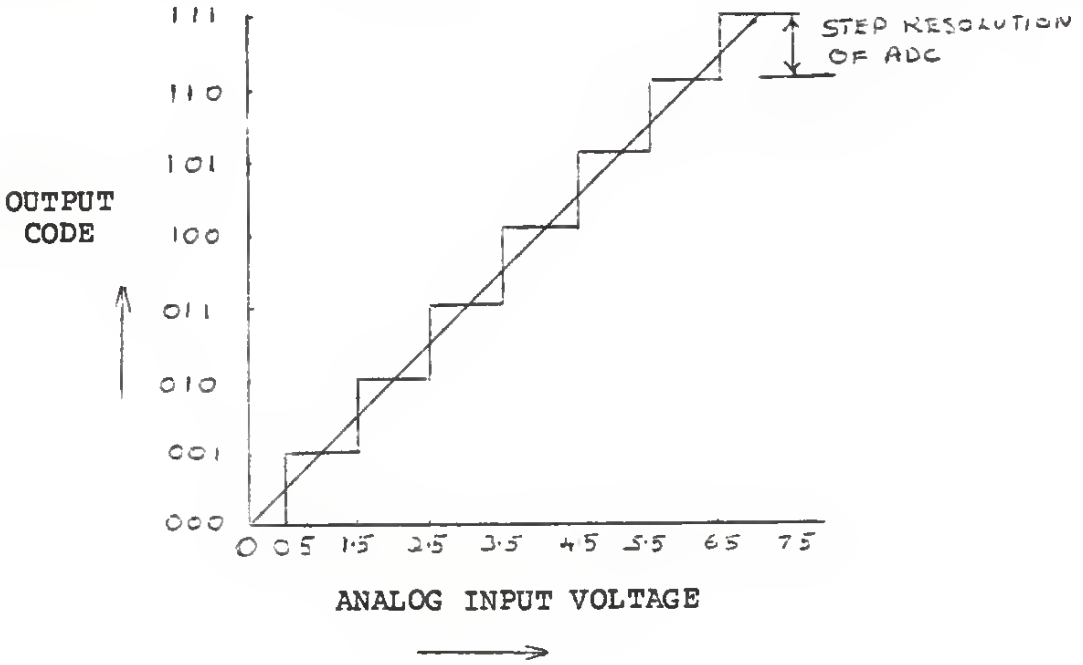


Fig. 10

THEORETICAL TRANSFER FUNCTION OF AN ADC (3 LSB'S ONLY)

2.6 FLASH ANALOG TO DIGITAL CONVERTER.

A Flash ADC is a very specialized analog to digital converter where the input analog voltage is applied in parallel to a group of comparators simultaneously. For an 8 bit converter the number of comparators needed are 2**8 - 1=255 comparators. Each comparator in the group is referenced to one set of voltages obtained by putting an equal resistor string across a very stable, precisely settable reference voltage. There exists therefore, a comparator level for each possible quantization level. Thus for any input level, all comparators referenced to quantization levels below that input will be actuated and all comparators referenced to quantization levels above that input level will not be actuated. The outputs of all the comparators are applied to the decoding logic which yields the parallel output code. This form of converter, perhaps best characterized as "brute force" in structure, is extremely fast. It is limited in speed only by the reaction time of the decoding comparators and the logic. The number of bits could be increased for better resolution by connecting ADC's in tandem. The block diagram for an 8 bit flash ADC is shown in Fig. 11.

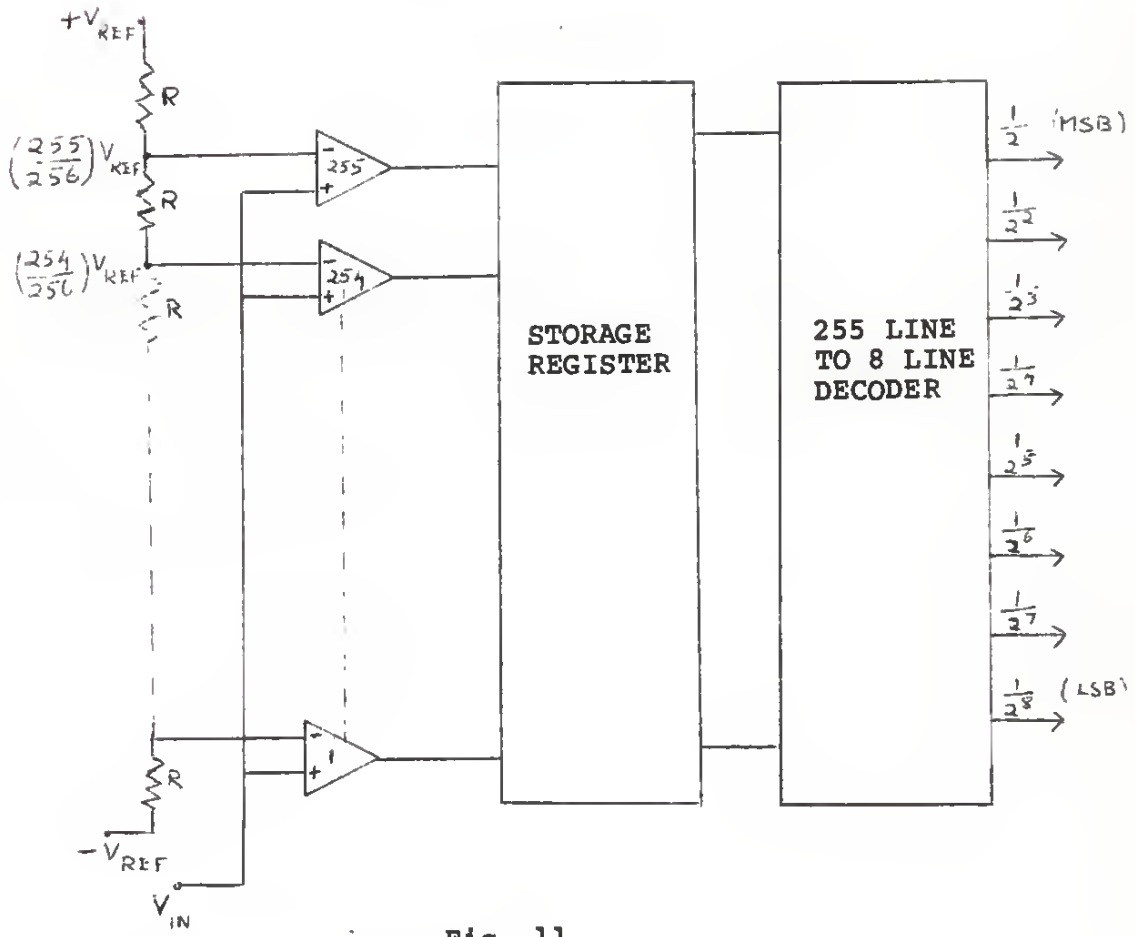


Fig. 11

SCHEMATIC DIAGRAM OF AN 8 BIT FLASH ADC

CA 3308 is an 8 bit CMOS Flash ADC which has a reference input voltage of ±6.4 V. The digital word corresponding to an input analog signal is available in one clock cycle. The power consumption of the CA 3308 is less than 150 mW at a clock frequency of 15 MHZ when operated at 5 V supply. There are two chip enable signals; one active high, and the other active low.

These chip enables control the tri-state outputs of the converter. $\overline{CE1}$ will independently disable bit 1 to 8 when it is in high state. Similarly CE2 will independently disable all the bits when it is in the low state.

The higher order address lines are used in selecting the Flash ADC in the address 6000H. The Flash ADC is continously clocked by the system clock of the NSC 800 at 2 MHZ. The voltage resolution that could be achieved with 8 bits with $\pm$6.4 V reference voltage is (12.8 /2**8)= .05 V and the quantization error is .025V.

Although the Flash ADC could sample at a maximum rate of 15 MHz, due to program controlled data acquisition the effective sampling rate is reduced to 100 KHZ. Reading one sample from the ADC by the NSC 800 and storing it in memory takes 40T states which is about 10 secs. Hence the maximum effective sampling rate is considerably reduced to 100 KHZ. So signals of frequency greater than 50 KHZ would not preserve the required information according to Nyquist criterion. Hence it would probably be better to clock the ADC at 2 MHZ since there would be no use in sampling at a higher frequency. The read timing diagrams for NSC 800 with CA3308 is shown in Fig.12.

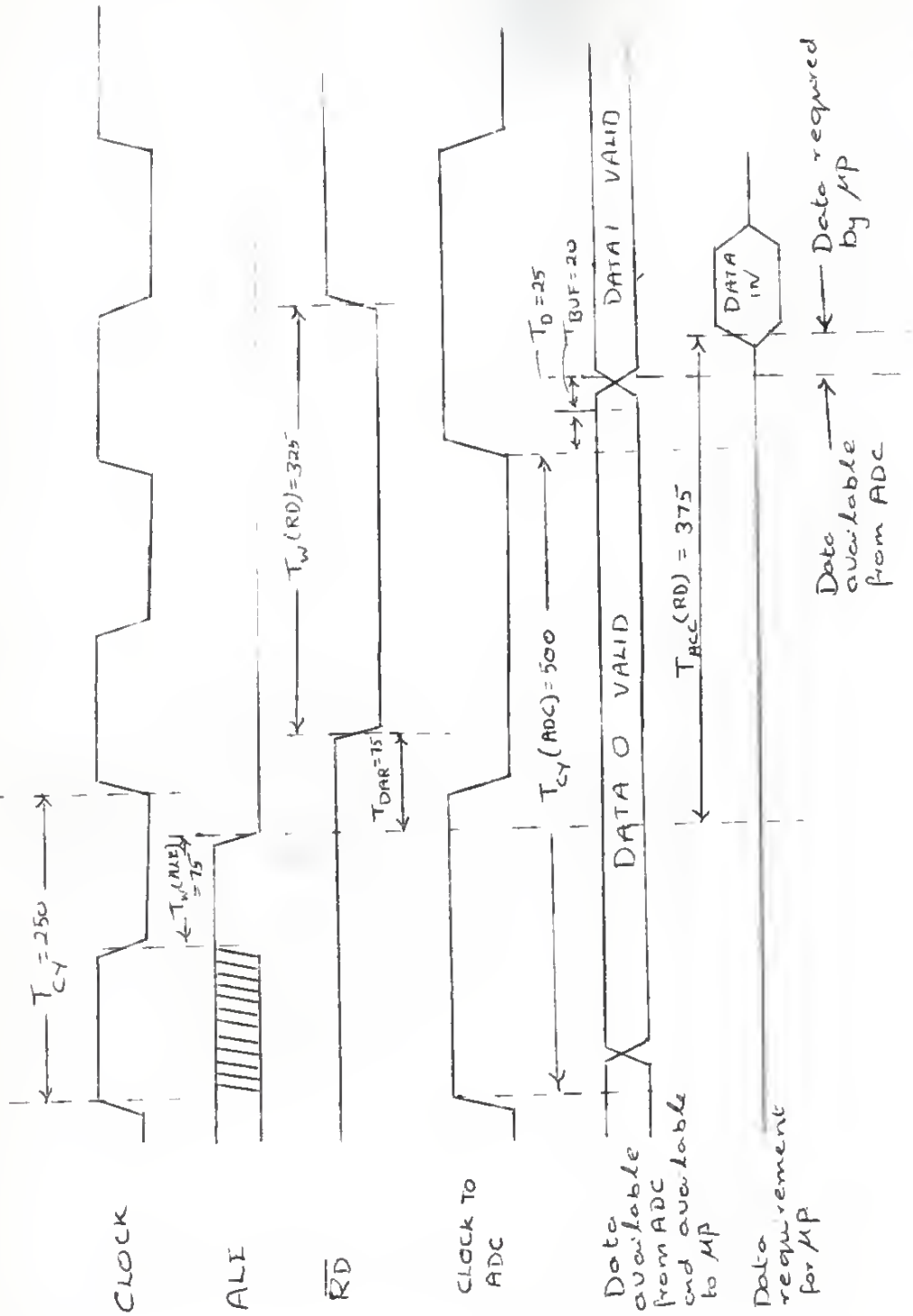


Fig. 12

READ TIMING DIAGRAM OF THE NSC 800 WITH CA 3308 ADC

2.7 OPTICAL FIBER COMMUNICATION.

Since the introduction of optical fibers, digital data communication has been growing in an ever increasing level of technology. With the advantage of high-speed,low-noise data transmission, optical fibers have shown excellent data communication abilities. Actually, the Bell system has made a commitment to manufacture and install fiber optic systems as an alternative to wire cable.

The basic optical fiber system is illustrated in Fig.13. It consists of a transmitter which transforms an electrical signal to be transmitted into an optical signal, a receiver which converts the optical signal back to the original electrical form, and a fiber transmission line which conducts the optical signal from the transmitter to the receiver.

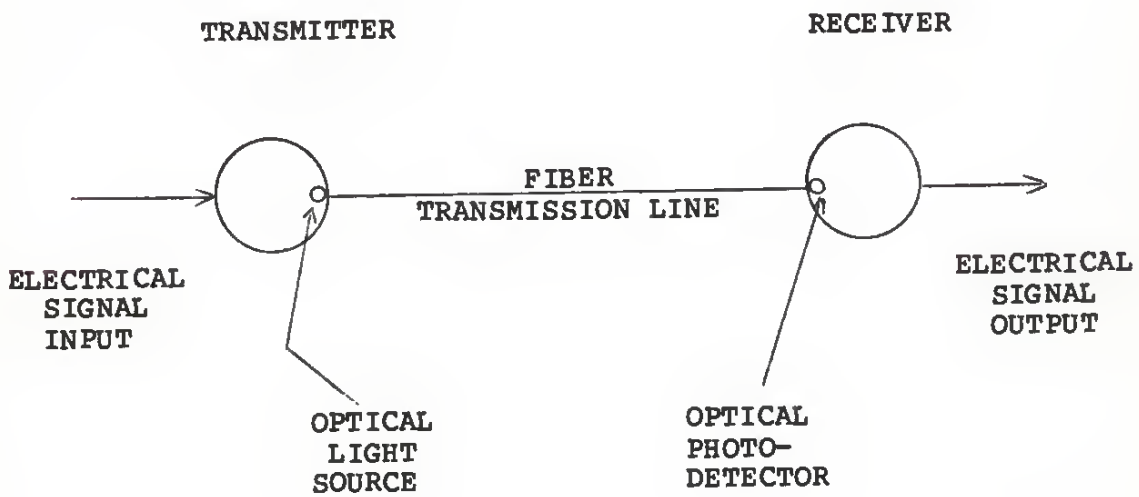


Fig. 13

A BASIC OPTICAL FIBER SYSTEM

Three new components are involved : the light source, the photodetector, and the optical fiber transmission line. The optical light source generates the optical energy which serves as the information carrier and the photodetector detects the optical energy and converts it into an electrical form.

The HFBR-1501 transmitter modules incorporate a 665 nm LED targeted at the low attenuation window for the HFBR-3500 plastic fiber optic cable. The transmitters can be easily interfaced to standard TTL logic. The optical power output of the HFBR-1501 is specified at the end of 0.5m of cable. The HFBR-2501 receiver modules feature a shielded photodetector and a wide bandwidth DC amplifier for high EMI immunity. A Schottky clamped open-collector output transistor allows interfacing to common logic families. An integrated 1000 ohm resistor internally connected to Vcc may be externally jumpered to provide a pull-up for ease of use with +5V logic.

2.8 THEORY OF OPTICAL FIBERS.

Central to optical fiber systems is the optical fiber waveguide which is threadlike in structure. In its simplest form it has a light guiding region referred to as the core, surrounded by a layer of material, a coaxial outer region, known as the cladding. The information carrying capacity of the fiber depends on the fiber design, the fiber material properties, and the spectral width of the electromagnetic energy source.

Total internal reflection, which occurs when a light beam emerges from a denser to a rarer medium, is the basic mechanism involved in the transmission of light along the fiber.

The principle of operation of the fiber can be explained rigorously in terms of geometric optics.

An optical fiber is a long cylindrical structure, usually with a circular cross section. In its simplest form it has two coaxial regions, the inner region is the light guiding core while the cladding completes the light guiding region in such a way that the outer surface can be handled with little disturbance to the propagation characteristics. The fiber with a core of refractive index n_1 surrounded by a cladding of constant refractive index n_2 is referred to as a step-index fiber wave guide. For light guidance to occur $n_1 > n_2$ is required.

2.9 REFLECTION AND REFRACTION AT A DIELECTRIC INTERFACE.

When a plane wave front is incident at a boundary of two dielectrics with different refractive indices n_1 and n_2, the incident wave is reflected and refracted at the boundary as shown in Fig.14.

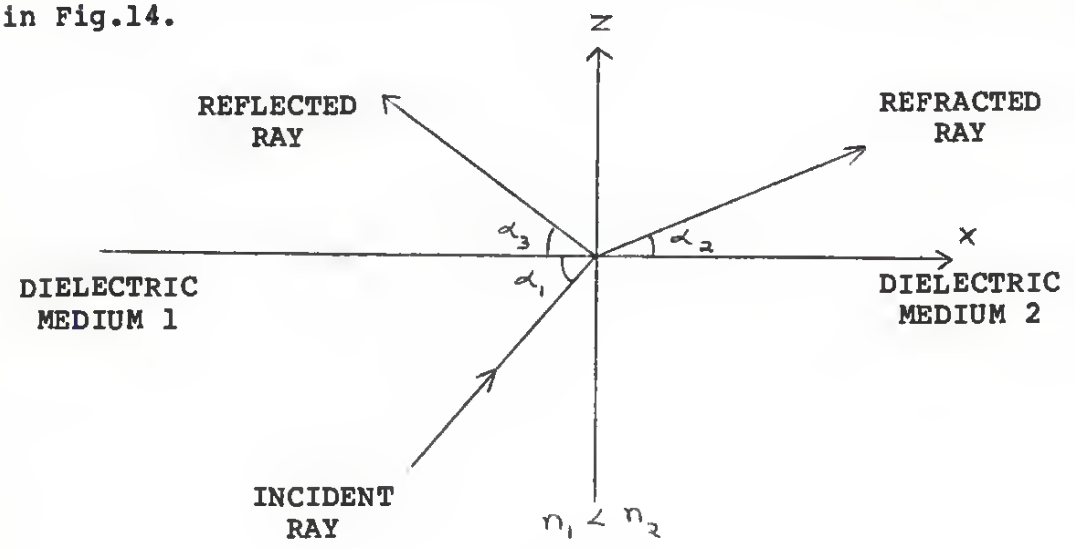


Fig. 14

LIGHT RAYS AT A DIELECTRIC INTERFACE

According to the laws of reflection and refraction it is found that the angle of reflection α_3 is always equal to the angle of incidence α_1: $\alpha_1 = \alpha_3$. The law of refraction which is known as Snell's law satisfies the relationship

$$\frac{\text{Sin}(\alpha_1)}{\text{Sin}(\alpha_2)} = \frac{n_2}{n_1}$$

For $n_1 > n_2$, Sin $_2$ approaches α_1 to $\alpha_c = \text{Sin}^{-1}(n_2/n_1)$. This angle is referred to as the critical angle. When is infinitesimally greater than α_c, no refracted wave can exist. This effect is the total internal reflection. This is always valid when $\alpha > \alpha_c$. The critical angle allows the light to be bounded within the denser medium and permits lossless propogation.

The fibers being used are the Hewlett Packard HFBR-0500 snap in optical fiber links. The couplers are TTL compatible and can be run at a 5 M baud maximum data transmit and receive rate. Also an additional peripheral driver 75451 is needed to run the transmit coupler. The driver takes TTL levels from the UART and boosts the high signal current for proper operation of the transmit coupler LED. For a 5 meter length of the cable the driver components, current limiting resistors were designed. The specification sheets gives the necessary equations, graphs for calculation of the components. The device block diagram using the optical fiber link is shown in Fig.15.

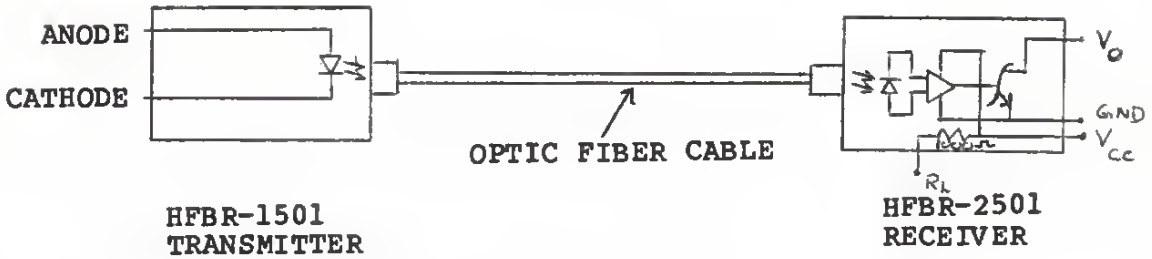


Fig. 15

DEVICE BLOCK DIAGRAM OF THE OPTIC FIBER LINK

2.10 NSC 858- UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER.

The NSC 858 is a P2CMOS UART compatible with the NSC 800 Microprocessor and could transmit and receive data at 1 Mbaud maximum rate. It converts parallel data into serial and shifts it through the T*D pin and also receives through R*D pin and converts it to 8 bit parallel binary word to be read by the processor. The system clock of the NSC800 which is 4 MHz is divided by 4 for 1 MHZ external clock to the UART. The receiver clock is synchronized automatically when receiving data. This is necessary so that the device samples incoming data on every high to low transition and if the transition is not near the center of the data pulse the received data may not be necessarily valid. The NSC 858 has a multiplexed address/data bus so the ALE of the NSC 800 is used the strobe the address from the Input/Output port. Data is latched into the UART using the RD signal of NSC800 from the bidirectional transceiver.

The calculations for the driver coupler, current limiting resistors and the necessary graphs are in Appendix B.

2.11 LIMITATIONS OF THE SYSTEM.

The Optopod designed using the NSC 800 for high speed data acquisition has several limitations due to numerous constraints:

1) The sampling frequency of the ADC is not variable and remains constant irrespective of the input frequency. Auto ranging of the Flash ADC with respect to the amplitude of the signal is not attempted.

2) The Flash ADC has a maximum sampling rate of 15 MHZ but due to the time delay of 10 μsecs for every memory access the effective sampling rate is reduced to just 100 KHZ thus even losing the objective in choosing a Flash ADC for data acquisition.

3). In spite of the minimal hardware that is used the size of the board of the Optopod is not as small as expected. However this could be taken care of by using two boards that could be interconnected using board connectors.

The sampling frequency control was not incorporated in this design because of the hardware expansion and due to the space constraints of the board size. Hence input signals of frequency less than 50 KHZ could only be sampled. The effective sampling rate of the ADC is kept at 100 KHZ in spite of the clock of 2 MHZ thus reading one sample in every twenty. The process of reading the sample and storing it in memory slows the sampling rate by quite a bit that any processor even with a faster clock-rate would hardly improve the situation. The Intel-8751 which runs at 8 MHz clock was found to run no better and its effective

maximum sampling rate was found to be 150 KHZ. So only the process of Direct Memory Access could improve the situation. By direct memory access data available from the ADC after one clock cycle is available directly to be stored in memory without processor intervention. By this process a maximum sampling rate upto 8 MHZ could be achieved; the only disadvantage being the increase in hardware. For the same reason, sampling frequency control was not incorporated in the NSC 800 system. Both these features are incorporated in the system designed using the 8751.

CHAPTER-3

SOFTWARE DESCRIPTION

The software controls the acquisition of data and also the fiber optic data transmission. As stated earlier there is need for an interfacing processor between the optopod and the control computer because the speed of communication is different. Hence the interface would send and receive data in serial form from the Optopod and communicate with the host computer through its parallel ports in a handshaking mode. Data is converted from binary to ASCII and ASCII to binary by the interface depending on the direction of data flow.

The flow charts that are shown in Fig.16.a & 16.b essentially perform two functions: 1). Receive a character from the Computer (interface) to start sampling the signal and continue until 2K samples are stored in memory.
2). Send the stored data in memory serially to the computer (interface).

The NSC 800 has a Z-80 compatible instruction set which is used for the software driven routines. The NSC 858 UART is used to control the data bits, speed etc. The addressable registers in the UART in memory are shown below:

4000H - R*HOLDING REGISTER/T*HOLDING REGISTER
4001H - RECEIVER MODE REGISTER
4002H - TRANSMITTER MODE REGISTER
4003H - GLOBAL MODE REGISTER
4004H - COMMAND REGISTER
4008H - RECEIVER/TRANSMITTER STATUS REGISTER

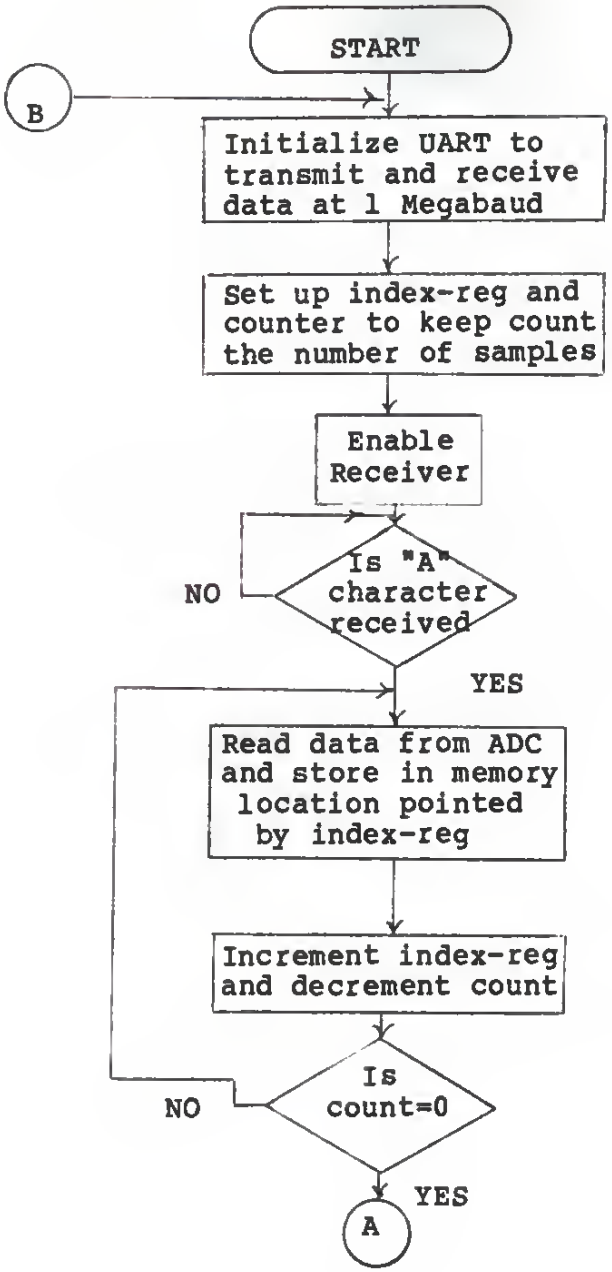


Fig. 16.a FLOWCHART FOR THE OPTOPOD TO RECEIVE REQUEST FROM COMPUTER(INTERFACE) AND STORE 2K SAMPLES IN MEMORY

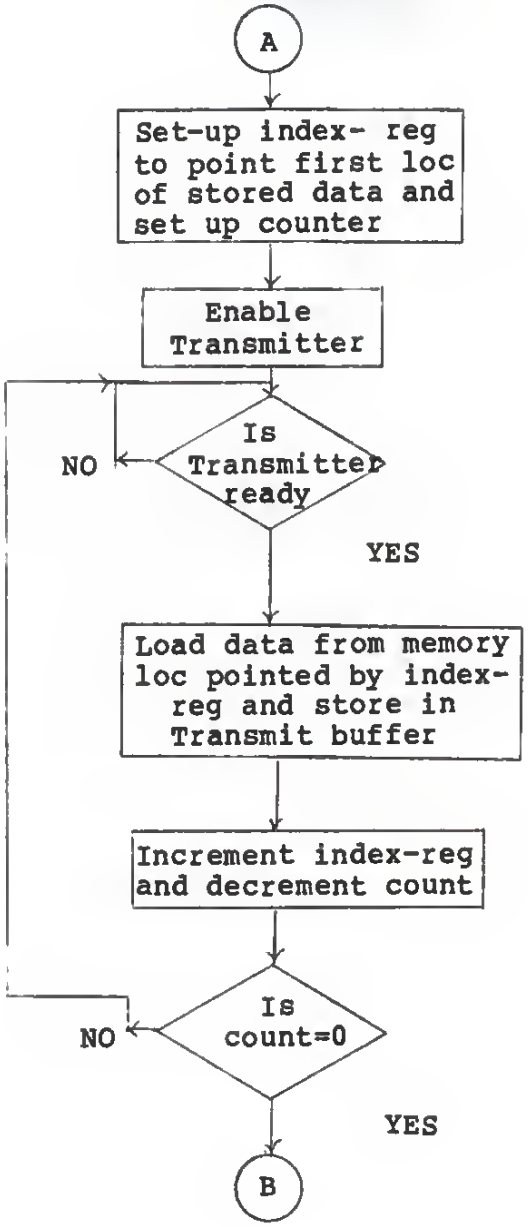


Fig. 16.b FLOWCHART SHOWING STORED DATA IN MEMORY TRANSMITTED TO COMPUTER(INTERFACE)

The Flash ADC is addressable at 6000H in memory and the scratch pad memory is addressed between 2000H-3FFFH . The receiver and transmitter-holding register receives and sends data while the receiver/transmitter mode-register fixes the character length, parity and also the clock mode. The Global mode register controls the clock factor and the number of stop bits to be used in data transmission. Receiver/Transmitter status-register has bits 1 and 0 which indicate whether data is ready on the receive side and whether the transmitting buffer is empty. The command register enables/disables the receiver/transmitter.

The BASIC program that initiates sampling of data is shown in Appendix C. It also stores the data in an array and scales the 8 bit binary words into voltage values. The monitor program in the NSC 800 first initializes the pod to receive a character from the computer (interface). When the character 'A' is received, data from the ADC is read and stored in memory in successive locations upto 2 K bytes. On completion the transmitter is enabled and data is sent out to the interface. The program written in Z-80 assembly does not do the debugging for data communication but acquires data on request and sends it to the interface. Error checking on data would lead to slower data sampling and hence is not pursued at this instance. The assembly language listing are shown in Appendix D.

CHAPTER-4

HARDWARE DESCRIPTION FOR FLASH ADC MODULE USING 8751

4.1 CHOICE OF A SUITABLE PROCESSOR.

The design of the Optopod using the NSC 800 Microprocessor was based upon considerations of low power consumption and high speed optical fiber communication. Due to program controlled data acquisition the Optopod designed could sample at a fixed maximum sampling frequency of 100 KHZ. It was found that if maximum efficency of the Flash ADC test module had to be achieved extra hardware would have to be used to fulfill this requirement. An Auto-ranging Flash ADC test module could then be constructed with sampling frequency control upto 8 MHZ.

A processor with more in-built features such as extra control ports, serial-port, on-board program memory was the one which could serve this particular application. At first a study was made on the INTEL-8744 Microcontroller and it was found that this would be suitable for the Optopod application. However because of the unavailabilty of this chip and also on account of its cost the design was not pursued. The INTEL-8751 Microcomputer was found to be more suitable and inexpensive and hence design using the 8751 was started.

4.2 INTEL-8751 MICROCOMPUTER.

INTEL-8751 is a stand-alone high performance 8 bit single chip microcomputer intended for use in sophisticated real-time applications, such as instrumentation, industrial control, and intelligent computer peripherals. It provides hardware features, architectural enhancements and new instructions that

make it a powerful and cost effective controller for applications requiring up to 64K bytes of Program memory and/or up to 64K bytes of data storage.

The 8751 contains a user programmable/UV erasable EPROM (4K*8) bits and (128*8) bits of RAM resident on the chip. It has 32 I/O lines, two 16 bit timer/event counters, high performance full duplex UART, an On-chip oscillator and Clock circuit.

With regard to software the 8751 is efficient both as a controller and also as an arithmetic processor. Efficient use of Program memory results from an instruction set consisting of 44% single-byte, 41% double-byte and 15% three-byte instructions. Most instructions are executable in 1 μsec. However the 8 bit unsigned multiplication and divisions are executable in 4 μsecs.

Since the 8751 has an on-board EPROM of 4K bytes, 128 bytes of RAM and a full duplex UART resident on the chip it would reduce the overall hardware requirement of this instrument module as compared to the NSC 800 based system. The effective area saved on using this integrated chip could be utilized for use of extra hardware for programmable sampling frequency control and for direct memory access.

As discussed earlier the in-built features of the 8751 could be used most effectively for control in a data acquisition system. The 8751 has 32 bi-directional I/O lines which could be individually addressed as a bit or as 4 eight bit parallel I/O ports. These ports are all multi-functional. Port 0 is the multiplexed address/data bus and port 2 is the higher order address bus for external memory access. Port 3 could be

configured individually to provide external interrupt request input, counter input, the serial port's receiver input and transmitter output and to generate the control signals for reading and writing external data memory. Port 1 is an 8 bit port which could be used for any specific purpose. Fig. 17 shows the 32 I/O lines and their functions.

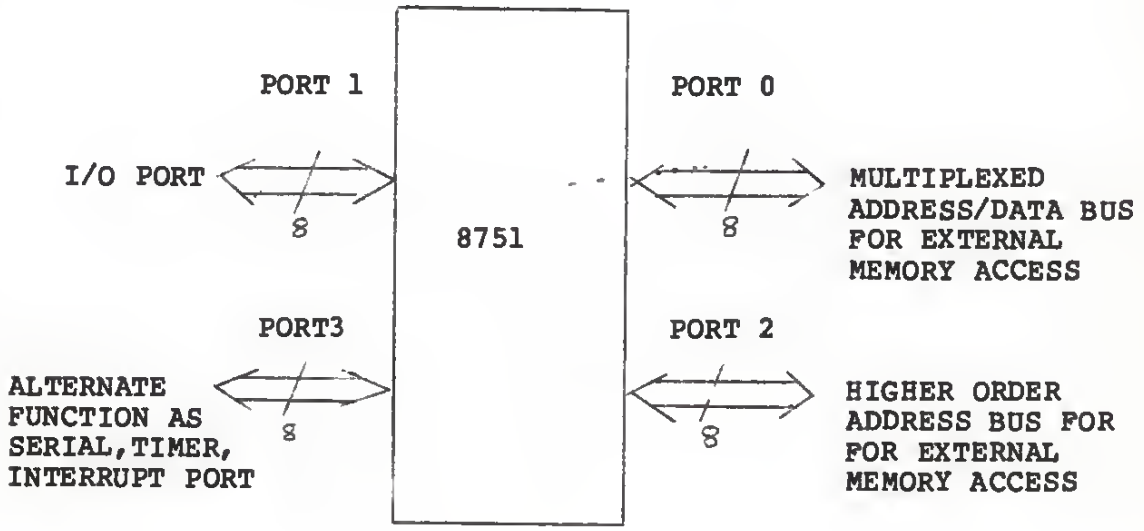


Fig. 17

INPUT/OUTPUT LINES OF 8751

4.3 MEMORY ORGANISATION OF 8751.

The 8751 CPU manipulates operands in four memory spaces. These are the 64K byte Program memory, 64K byte Data memory, 384 byte internal Data memory and 16 bit program counter spaces. The internal data memory address space is further divided into 256 byte internal data RAM and 128 byte special function register address space as shown in Fig. 18. Four register

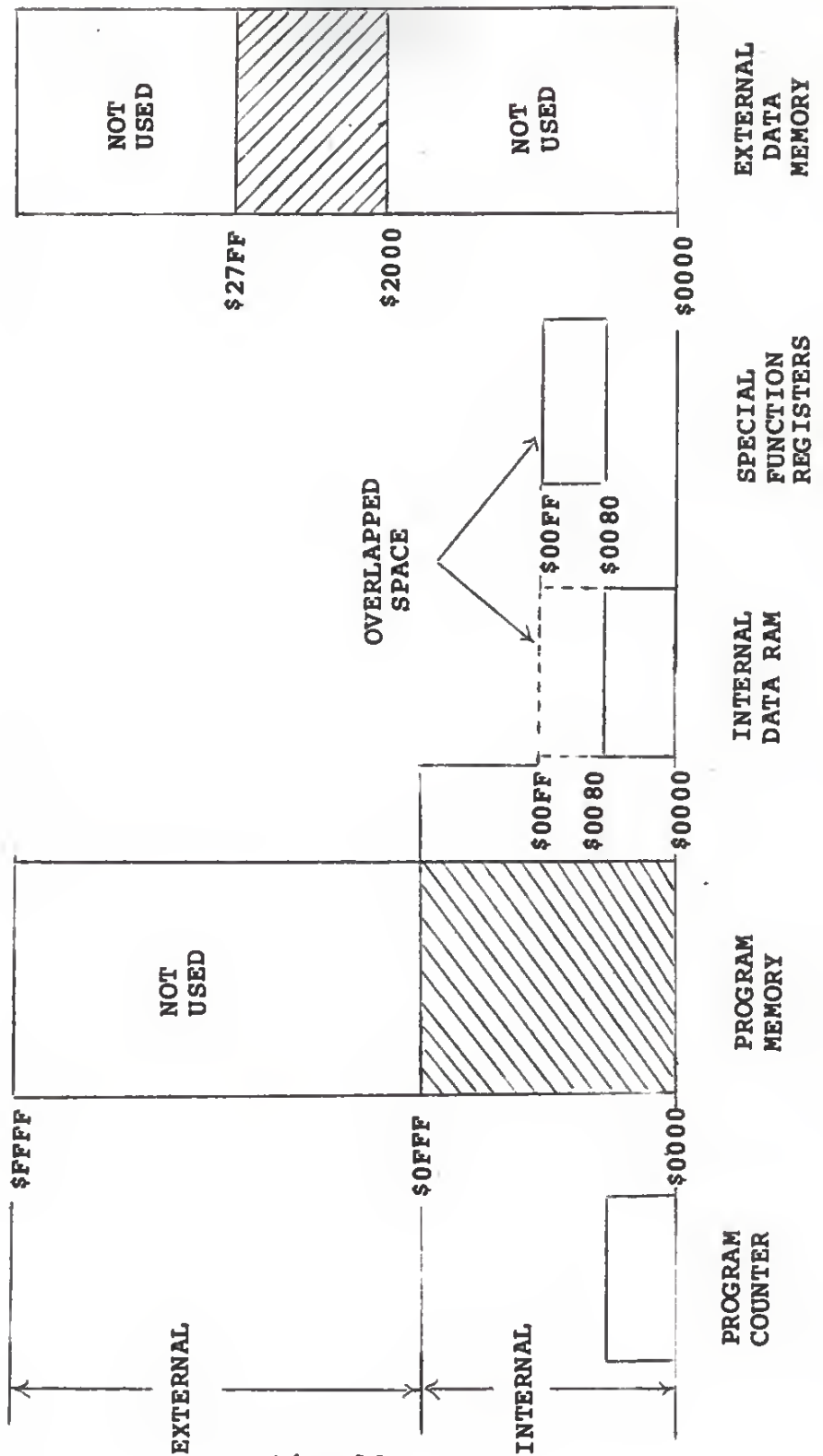


Fig. 18

MEMORY MAP OF THE 8751

banks(each with 8 registers), 128 addressable bits and the stack reside in the internal data RAM. The stack depth is limited only by the available internal data RAM and its location is determined by the 8 bit stack pointer. All registers except the program counter and the 4 eight register banks reside in the special function address space. The memory mapped registers include arithmetic registers, pointers, I/O port, interrupt system registers, timers and serial port. 128 bit locations in the SFR address space are addressable as bit.

Fig.18 shows the memory map of the 8751 in which an HM 6116 RAM of 2K bytes is placed in the external data memory space 2000H-3FFFH. The address decoding for the RAM is done using the higher order address lines A15, A14, A13 with logic Nand and Inverter gates.

Instructions from the internal EPROM are executed in a sequence from location 0000H by holding the $\overline{EA}$/Vdd high. This pin also receives the 21V EPROM programming voltage. Since Port 0 is a multiplexed address/data bus demultiplexing and buffering of data and address is done using a bi-directional transceiver and I/O port as in the NSC 800 system. ALE/$\overline{PROG}$ pin provides the latch enable output used for latching the address into external memory during normal operation. This pin also receives program pulse input during EPROM programming.

4.4 PROGRAMMING THE ON-BOARD EPROM OF 8751.

To program the EPROM, the 8751 must be running with a 4 to 6 MHZ oscillator. This is because the internal bus is being used to transfer address and program data to the appropriate registers. The address of an EPROM location to be programmed is applied to Portl and pins P2.0-P2.3 of Port2, while the data byte is applied to Port 0. Pins P2.4-P2.6 and PSEN should be held low, and P2.7 and RST high. $\overline{EA}$/VPP is held normally high and is pulsed to +21V. While $\overline{EA}$/VPP is at 21V, the ALE/PROG pin which is normally being held high is pulsed low for 50 msec. Then $\overline{EA}$/VPP is returned to high. Fig. 19 illustrates the programming configuration of the 8751. The Universal PROM programmer is an Intellec System peripheral capable of programming and verifying the 8751 when UPP-851 is inserted. Programming and verification operations are initiated from the Intellec development system console and are controlled by the Universal PROM Mapper(UPM) program.

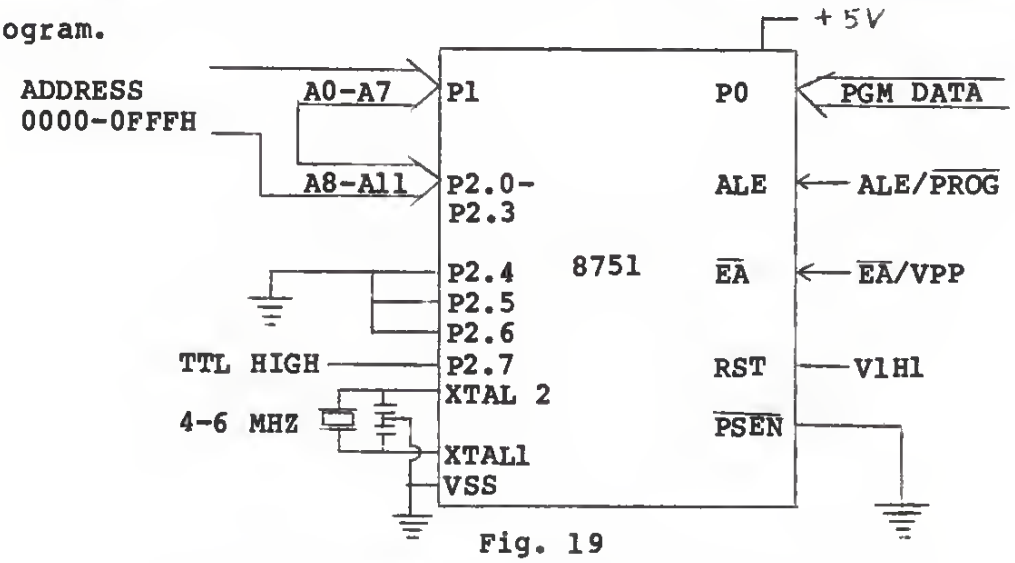


Fig. 19

EPROM PROGRAMMING CONFIGURATION OF THE 8751

4.5 DIRECT MEMORY ACCESS.

Direct memory access is one of the techniques used for access from an I/O device directly into memory asynchronously with the system clock. This method is used widely in systems where the I/O device speed is much higher than the processor speed. For instance the 8751 is clocked at 8 MHZ and has a minimum execution time of 1 μsec whereas the CA3308 Flash ADC has data available in 1 clock cycle i.e. 150 nsecs. Under program controlled data acquisition the 8751 could read one sample in 7 μsecs. Hence the effective sampling rate is reduced to about 150 KHZ although data could be sampled at a maximum rate of 15 MHZ. In other words, with program controlled data acquisition one sample in every hundred could only be read thus defeating the purpose of using a Flash converter. Based on these considerations it was thought that Direct memory access from Flash ADC into memory would yield better results. Through direct memory access, data available from the ADC in one clock cycle is directly stored in memory without CPU intervention. Extra hardware had to be incorporated for this process and the steps involved in this design are explained further in this chapter.

The conventional techniques used for DMA are 1) Halting the processor, 2) Cycle stealing and 3) Tri-state control method. However in this application a counter technique is used similar to Tri-state Control method to acquire data in one clock cycle. In this DMA technique the 8751 is not halted but it's address and data bus are tri-stated for the Flash ADC to access memory. Port 1 is used as the control port to control the whole process of

data acquisition while port0 and port3 which are the multiplexed address/data bus and serial port during normal operation are used for synthesizing the clock of required sampling frequency for the ADC and counter.

4.6 COUNTER TECHNIQUE TO IMPLEMENT DMA.

Data to be stored in memory has to be given the necessary address and data information at a particular instant of time and also for a particular duration. For a memory space of 2K bytes, a 12 bit counter could be used to address the 2048 locations successively. Samples of signal from the Flash ADC should be synchronized with the address for the data to be stored in successive memory locations.

A thorough study was made for DMA operation from CA 3308, an 8 bit Flash ADC to HM 6116 , a 2K byte CMOS RAM. A detailed analysis on the timing diagram revealed that data transfer could be achieved up to a maximum sampling frequency of 8 MHZ. Fig.20 shows the timing diagram for the DMA operation and explanation for these diagrams are also shown.

It should be noted that the address and data lines of the HM6116 RAM are being accessed not only by the ADC and counter for data storage but also by 8751 during the data transfer to the Control computer. Hence the address and data lines of the 6116 RAM have to be multiplexed so that this could be achieved. Multiplexing of these lines is accomplished by using Tri-state buffers between the ADC and memory. A similar approach is used for address information from the counter to memory. These tri-state buffers act transparently during data acquisition and are

tri-stated when 8751 reads the data from memory.

Port 1, the bi-directional I/O port with no primary function is used to control the whole operation of DMA. Sampling of the signal is started by using a bit of this port and until 2K samples are stored the sampling process is continued. On storing 2K samples the address and data bus of the 8751 are used to transmit data to the computer. All 8 bita of Portl are used, some of them are used for DMA and the rest for frequency synthesis. Some of the functions of Port 1 as related to DMA are listed below:

Bit 0- Output: To initiate the Flash ADC and binary counter for sampling the signal.

Bit 1- Output: To enable/disable the latch pin of the tri-state buffers.

Bit 2- Output: To disable/enable the bi-directional transceiver and I/O port from the data and address lines of the 8751.

Bit 7- Input: To check if Q_{12} of the binary counter has been set.

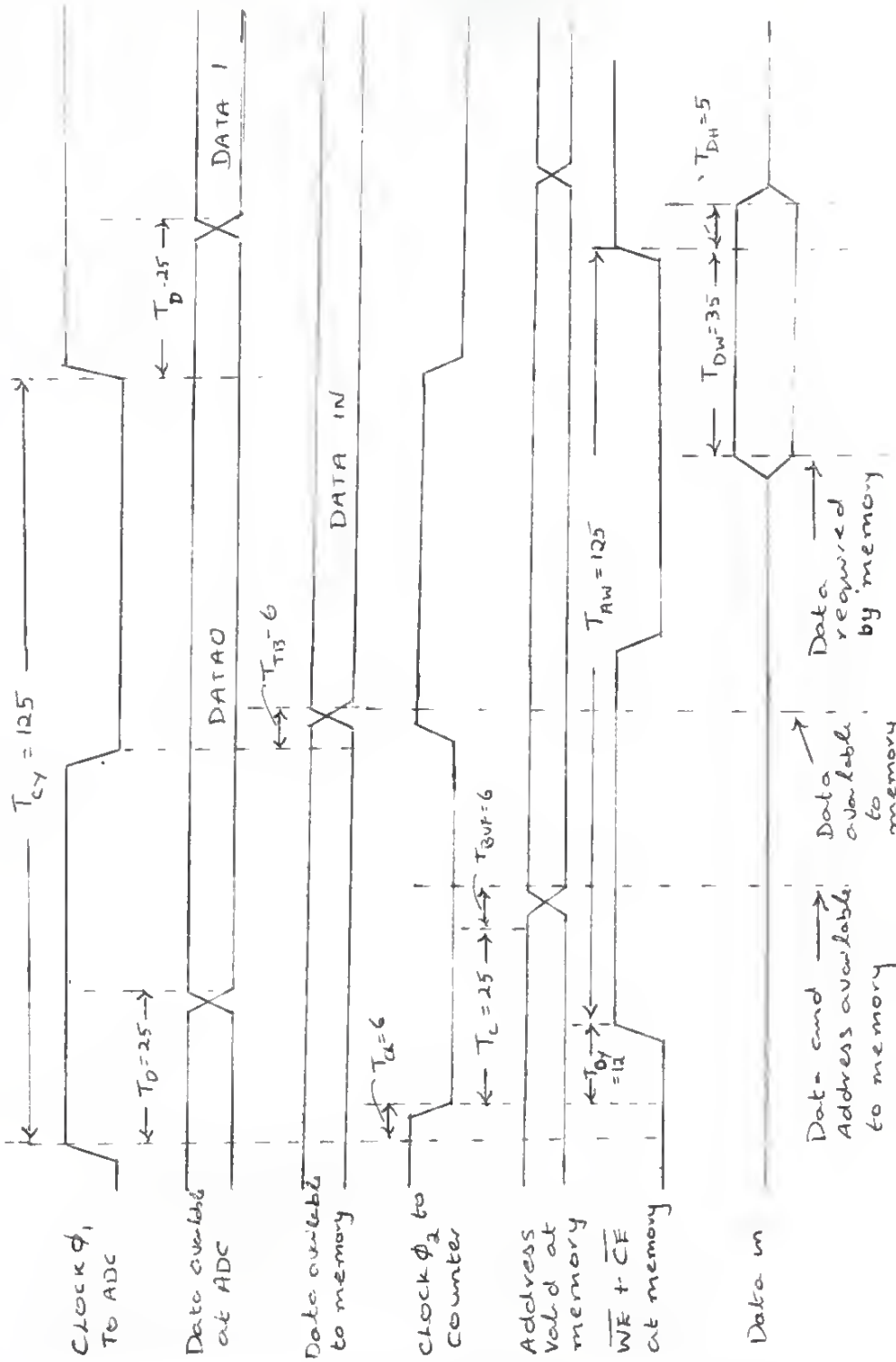


Fig. 20

TIMING DIAGRAM OF DATA TRANSFER FROM FLASH ADC CA 3308 TO THE HM 6116 RAM

4.6.b. EXPLANATION OF THE TIMING DIAGRAM FOR THE DATA WRITTEN INTO MEMORY FROM FLASH ADC.

Maximum clock frequency to Flash ADC and counter= 8 MHZ = ϕ_1

Data 0 valid at the ADC at time t_D= 25 nsecs from the rising edge of the clock. Data 1 is sampled at the rising edge of the clock.

Data 0 is latched from ADC into memory on falling edge of the clock with a buffer delay of 6 nsecs.

Data 0 available to memory = 63 + 6 = 69 nsecs from the starting point of sampling.

Data 0 available to memory from 69 nsecs to 194 nsecs.

Clock to counter ϕ_2 is $\overline{\phi_1}$ delayed by 6 nsecs.

Address valid at memory = falling edge of ϕ_2+T_C+T_{BUF} nsecs

= 31 nsecs from falling edge of ϕ_2

where: T_C= Delay for counter to count after clock ϕ_2 is low, and T_{BUF} = Delay at Tri-state buffer.

$\overline{WE}$ and $\overline{CE}$ are derived from ϕ_1 delayed by 12 nsecs.

Data required by the memory is T_{DW} nsecs before the end of write period and T_{DH} after end of chip disable.

From the timing requirements of the memory and timing cycles of the Flash ADC and counter it looks reasonable for the data transfer to occur.

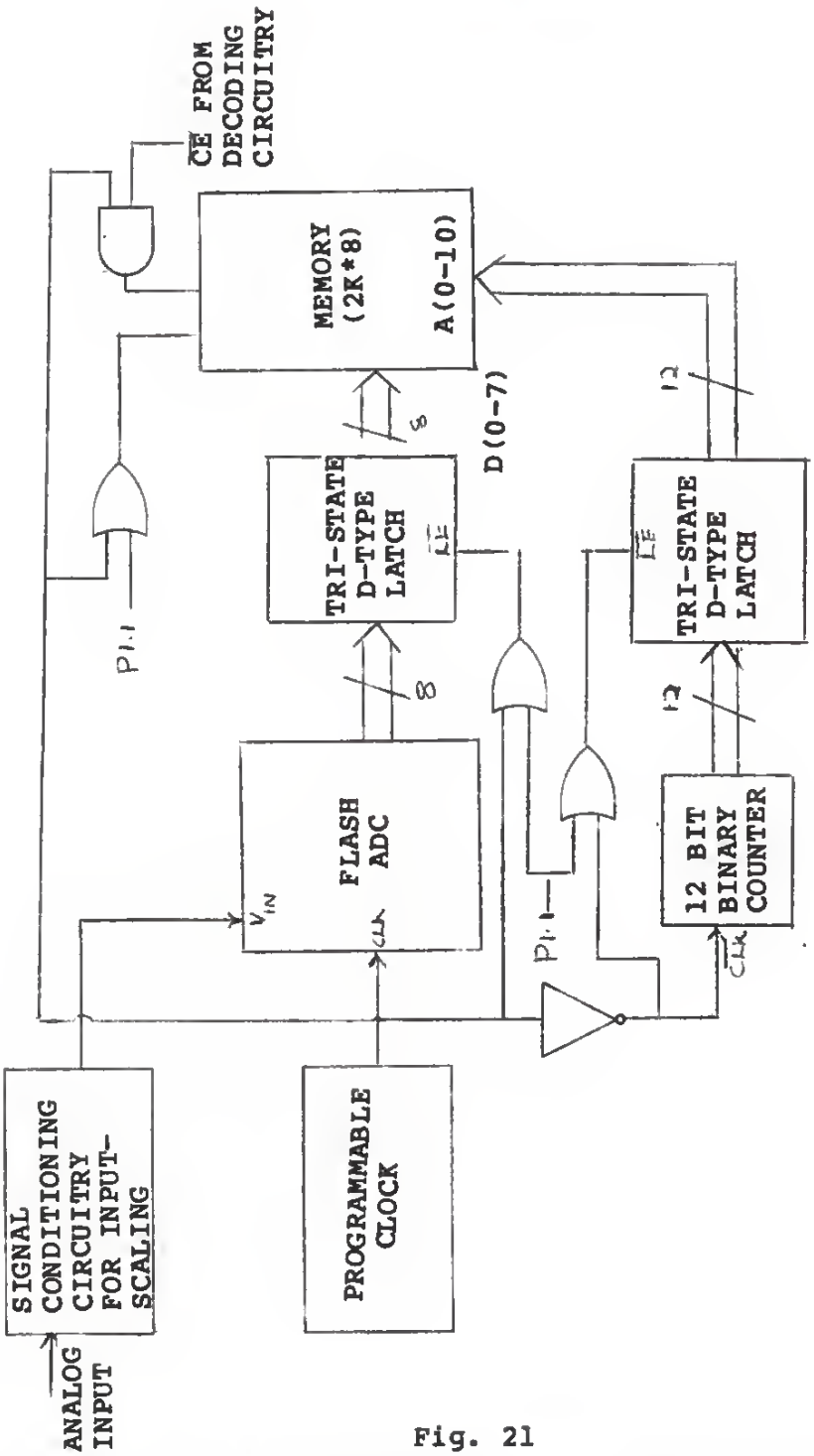


Fig. 21

BLOCK DIAGRAM SHOWING DATA TRANSFER FROM FLASH ADC TO MEMORY THROUGH DMA

A detailed study of the DMA process can be made with the help of Fig. 21. As indicated earlier since the memory lines are accessed by the 8751 microcomputer and also by the Flash ADC and counter logic gates are used to facililate multiplexed operation. The $\overline{WR}$ of 6116 RAM is controlled by an OR-gate whose logical inputs are the sampling clock and bit 1 of portl. Data is written into memory by a high to low transition of the clock signal. Bit 1 is pulled high to disable the memory write process through bitl. The memory is selected through the logic AND of the $\overline{CE}$ from the decoding circuit and the sampling clock. So the memory chip is enabled on all high to low transitions of the clock for writing data. When data is read by the 8751 microcomputer the sampling clock has little significance. Data and address from ADC and counter are latched respectively into tri-state buffers on the high to low transitions of their clocks. Bit 1 initially is set low for latching to take place and is pulled high when $\overline{WR}$ is disabled.

The various steps that are involved in initiating DMA through Port 1 are as follows:

1) Set bit 2 high to tri-state the bi-directional transceiver and I/O port,

2) Clear bit 1 to enable tri-state buffers to latch the address and data on high to low transition of sampling clock,

3) Clear bit 0 to trigger the Flash ADC and binary counter, and

4) Check if bit 7 i.e. Q_{12} of counter is set. If bit 7 is set 2048 samples are already stored in memory then steps 2,3 and 1 are undone in a sequence. This process is delayed until Q_{12} is

found to be high.

Data is read by the 8751 Microcomputer from HM6116 memory successively and sent to the computer through the T*D pin. The read timing diagram for 8751 with the 6116 RAM was made and the study revealed that this memory was compatible up to a frequency of 8 MHZ. The read timing diagram is shown in Fig. 22.

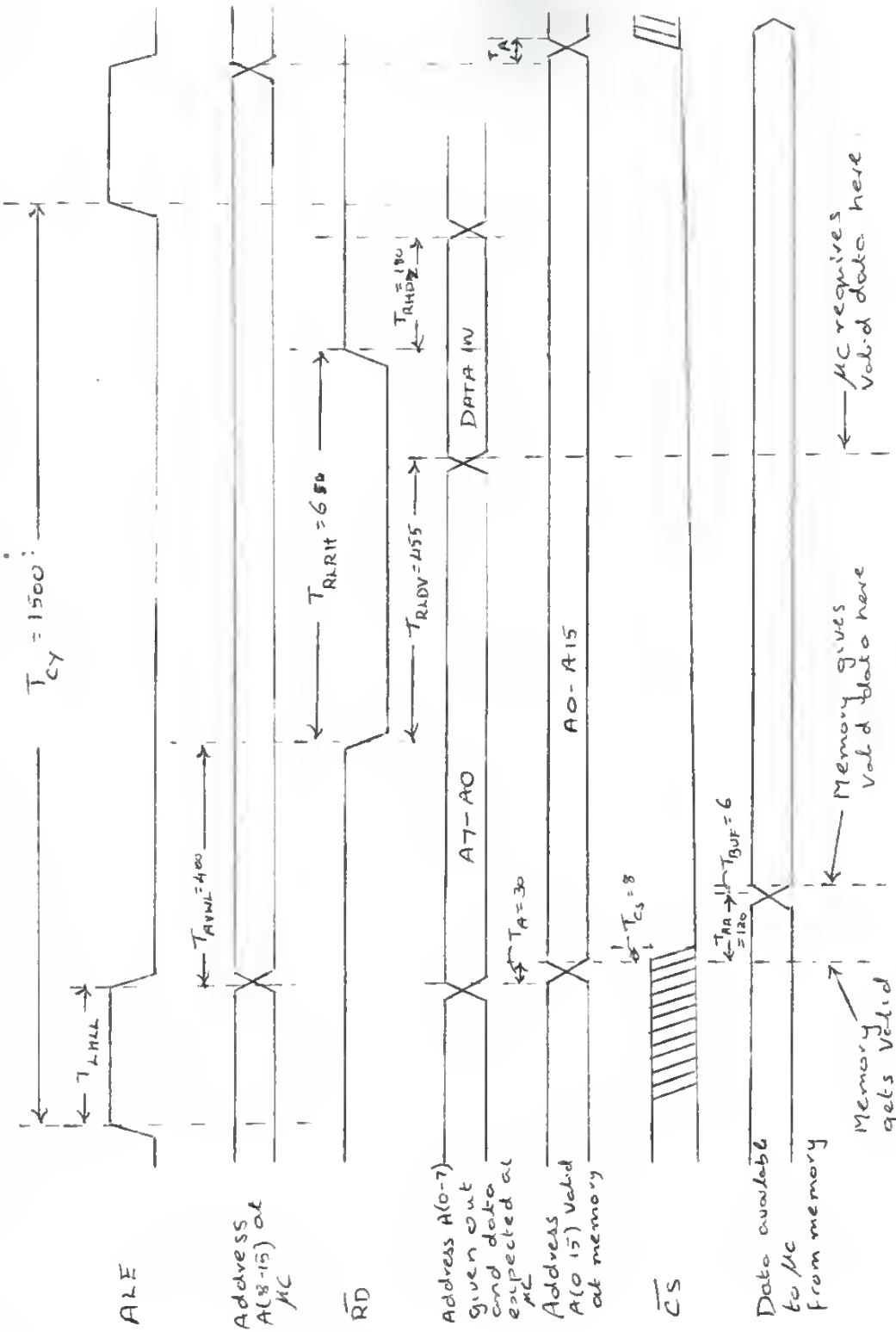


Fig. 22

READ TIMING DIAGRAM OF THE 8751 WITH HM 6116 RAM

4.7 SAMPLING FREQUENCY CONTROL.

There would be very little meaning in sampling a signal at a constant high frequency in the order of Megahertz when the input signal frequency is in the order of hundreds of hertz. So it is necessary to vary the sampling frequency of the Flash ADC depending on the range of input frequency. The sampling frequency could be varied so that it is greater than at least twice the frequency of the input signal through a programmable clock circuit. This frequency could be synthesized in coarse steps using a programmable Phase Locked loop. The process of Direct Memory Access and PLL frequency control are linked in such a way so that the clock frequency of the required range is synthesized before the actual sampling is started. A brief discussion on the basic operation of Phase Locked Loop is followed by the method of frequency control using 8751.

4.8 PHASE LOCKED LOOP.

The majority of new frequency synthesizers utilize the Phase Locked Loop. With the realization of a PLL in an integrated circuit, an inexpensive frequency synthesizer could be constructed.

A phase locked loop includes a phase detector, low pass filter and a voltage controlled oscillator as shown in Fig. 23. The term phase locked loop refers to a feedback loop in which the input and feedback parameters are either the relative phases or frequency of the waveform. The function of the PLL is to track the small difference in phase or frequency between the input and feedback signal.

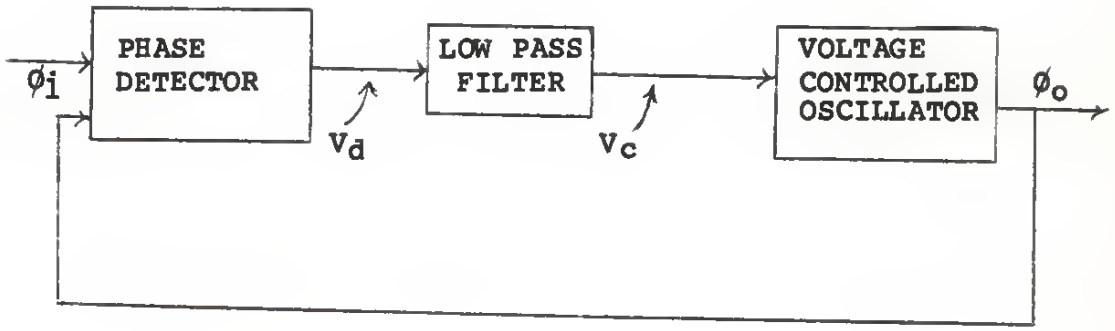


Fig. 23

A PHASE LOCKED LOOP

The Phase detector measures the phase difference between its two inputs. The Phase detector output is then filtered by a low pass filter and applied to a voltage controlled oscillator. The VCO input voltage changes its frequency in a direction that reduces the phase difference between the input signal and the local oscillator. The loop is said to be Phase-locked when the phase difference is reduced to zero.

The Phase detector although a non-linear device could be modelled as a linear one when the loop is in lock. When the loop is locked the Phase detector output voltage is proportional to the difference in phase between inputs ie $V_d = K(\phi_i - \phi_o)$ where ϕ_i and ϕ_o are the phases of the input signal and VCO output respectively and K is the Phase detector gain factor. The output frequency of the VCO deviates from its free running frequency by an increment $w = K_o V_e$ where Ve is the output of

the low pass filter and K_o is the VCO gain factor.

The configuration shown in Fig.23 is probably the most simple and minimum configuration of the PLL. Since there is no divider in the loop the output frequency of the VCO and the reference input are the same. However to synthesize a higher frequency and for better control of frequency ,PLL digital dividers and phase/frequency comparators are used so that the output frequency can be as high as 10 MHZ when the Phase/Frequency comparision is done around 5-10 KHZ. The digital frequency divider steps down the high frequency of the VCO output to a smaller range for better control. Fig. 24 shows the block diagram of a practical Phase Locked Loop that could be implemented in a microcomputer based system.

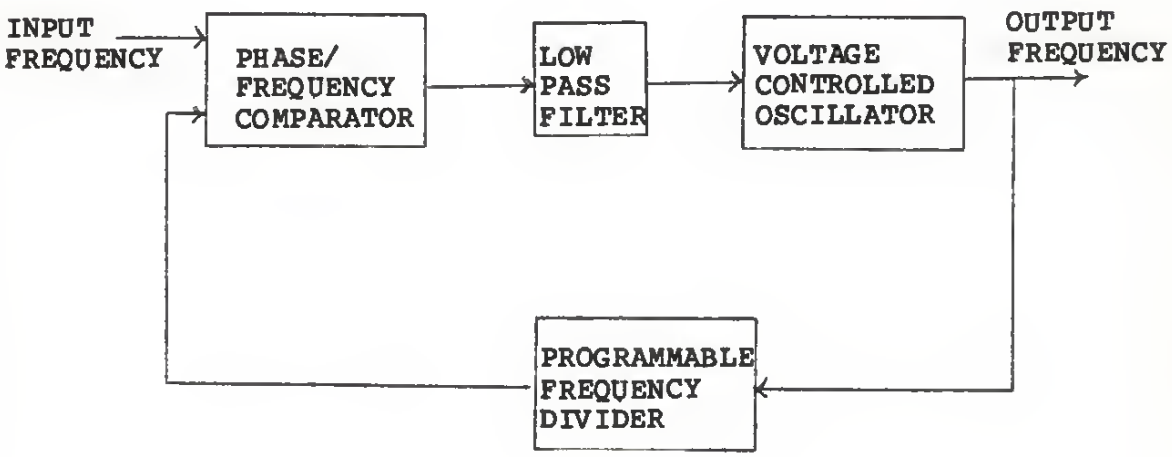


Fig. 24

A PRACTICAL PHASE LOCKED LOOP THAT COULD BE IMPLEMENTED IN A MICROCOMPUTER BASED SYSTEM

The digital Phase/Frequency comparator has two ranges for acquisition. One is called the lock-in range and the other is called the pull-in range. The total time to acquire both frequency and phase lock is called the digital acquisition time. The tuning range of the Voltage Controlled Oscillator is the frequency range over which the VCO could be tuned with the available control voltage. When the loop is switched on for the first time, it is far from being locked and the VCO frequency can be anywhere within the tuning range. Very few loops acquire locking by themselves, a process which is called self-acquisition. Generally the tuning range is larger than the acquisition range. If the loop is closed for the first time the "pull-in" process will occur. The oscillator frequency together with the reference frequency will generate a beat-note and a dc control voltage of such phases that the VCO is pulled in a direction of frequency lock. As the oscillator itself generates noise in the form of residual frequency modulation, the oscillator is constantly trying to break out of lock, and the loop is constantly monitoring the state and reasserting lock.

There are practical difficulties involved in the Phase Locked Loop operation like noise and spurious response in the loop although the in theory they seem to look pretty simple. The PLL frequency synthesis is probably one of the most reliable methods to synthesize frequency upto 25 MHZ. Coarse control of the frequency could be achieved by making the PLL programmable.

MC145143 is a CMOS Phase Locked Loop frequency synthesizer that has an in-built programmable-reference-divider. The block diagram of the MC 145143 is shown in Fig. 25.

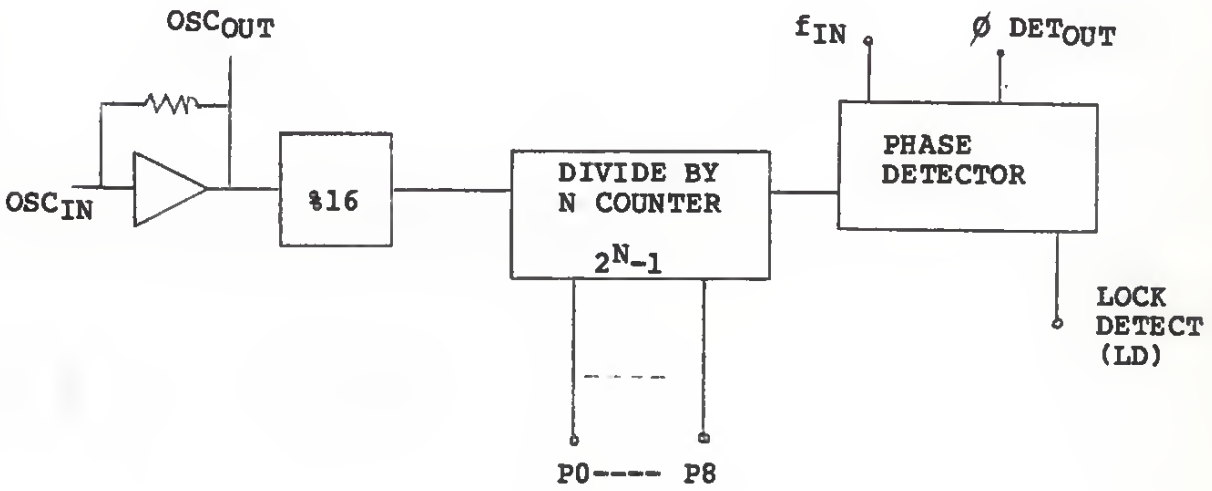


Fig. 25

BLOCK DIAGRAM OF THE PROGRAMMABLE PLL MC 145143

This device contains the oscillator circuitry to operate with fundamental mode crystals upto 10.24 MHZ. The oscillator circuit is connected to a Phase-detector through a divide-by-16 and divide by N (2^N-1) counter. The reference oscillator can be divided in steps of 16 between 32 and 8176 before interfacing with the Phase detector. P0-P8 are the programmable divider inputs in binary that divide the reference oscillator between 32 and 8176 (16*2-16*511). f_{in} the frequency derived from the VCO output is an input to the Phase detector. The Phase detector output ϕ Det $_{out}$ is a signal for control of external VCO. The output is high when f_{in} is greater than the

reference frequency; output low when f_{in} is less than the reference frequency. The signal LD, Lock Detector output is a signal to test to check if the loop is locked. The signal output is low when out of lock.

When data is acquired by memory through DMA the address and data buses of the 8751 are not used. The data and address buffers (Octal bidirectional transceiver and I/O port) are tri-stated to disable these lines from the 8751 to memory. The serial Port's transmitter output and receiver input are also disabled during this time. Hence Port 0 and Port 3 could be used for the program the PLL frequency synthesizer for the required sampling frequency before starting DMA.

Eight of the 9 binary inputs of MC145143 PLL are used to divide the reference oscillator of frequency 8 MHZ in the range up to 10 KHz. These 8 inputs divide in the range between 16 and 4080 (16*1-16*255) the oscillator frequency suitably for better control of phase lock.

NE 564 is the external VCO used for this design with the MC145143 PLL frequency synthesizer. The VCO section of the PLL is used for this design since the NE 564 does not possess programmable input for microcomputer control. NE 564 is a versatile PLL designed for operation up to 50 MHZ. An emitter coupled oscillator is used in the VCO. The free-running frequency of the oscillator is given by the following equation:

$f_0 = 1/16\ R_C C_1$ where R_C =100 ohms (internal)

and C_1= external setting capacitor. The variation of frequency of the VCO depends on the capacitor setting and also on the Phase

detector output.

The principle behind the PLL control of sampling frequency through the 8751 Microcomputer could be explained using the block diagram shown in Fig. 26.

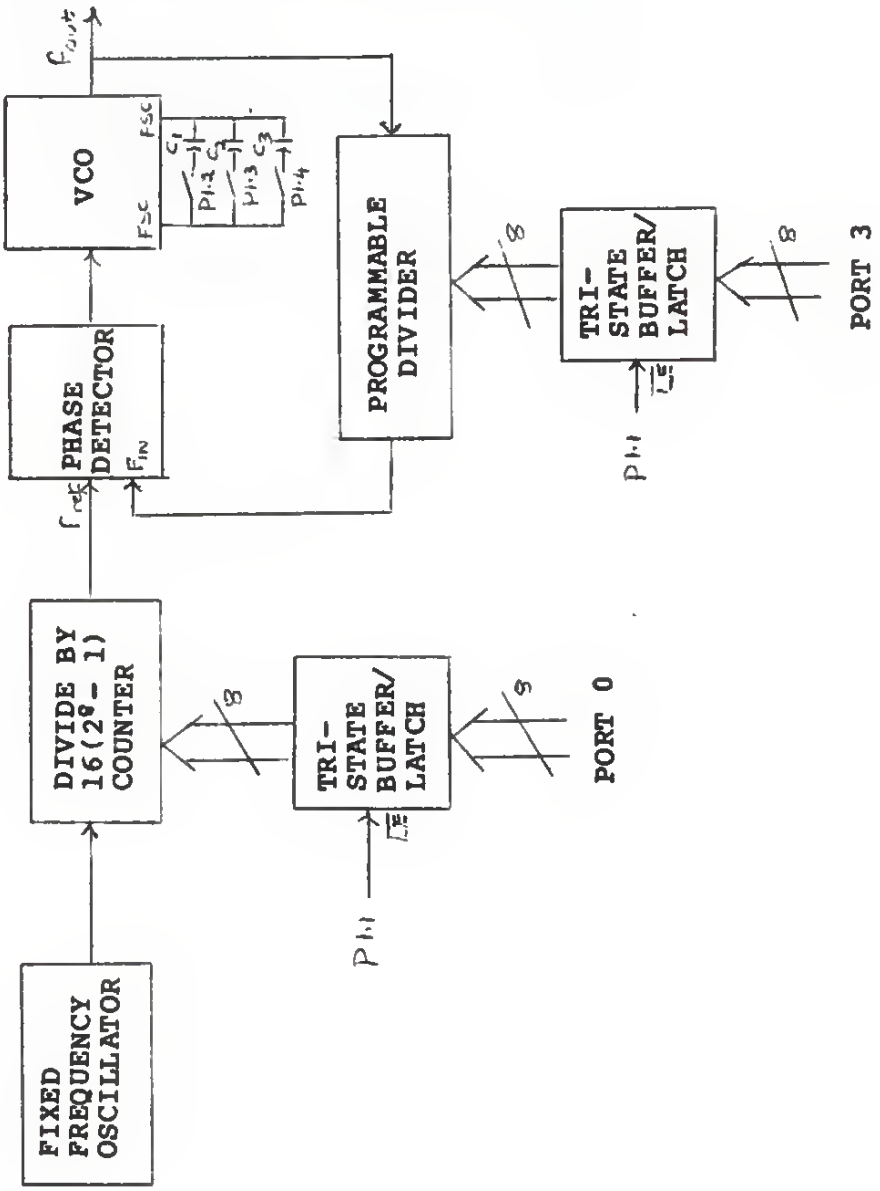


Fig. 26

BLOCK SCHEMATIC FOR THE PLL FREQUENCY CONTROL THROUGH THE 8751

Port 0 is used to send a control word for dividing the reference oscillator suitably so that frequency/phase comparision done at a frequency around 10 KHZ. The Phase detector compares the reference divided frequency with the VCO output frequency to produce an output. Port 3 is used to control the programmable digital divider through appropriate control words to the synchronous presettable counter 74LS161. Fig. 27 shows the configuration of the 74LS161 used as a programmable divider.

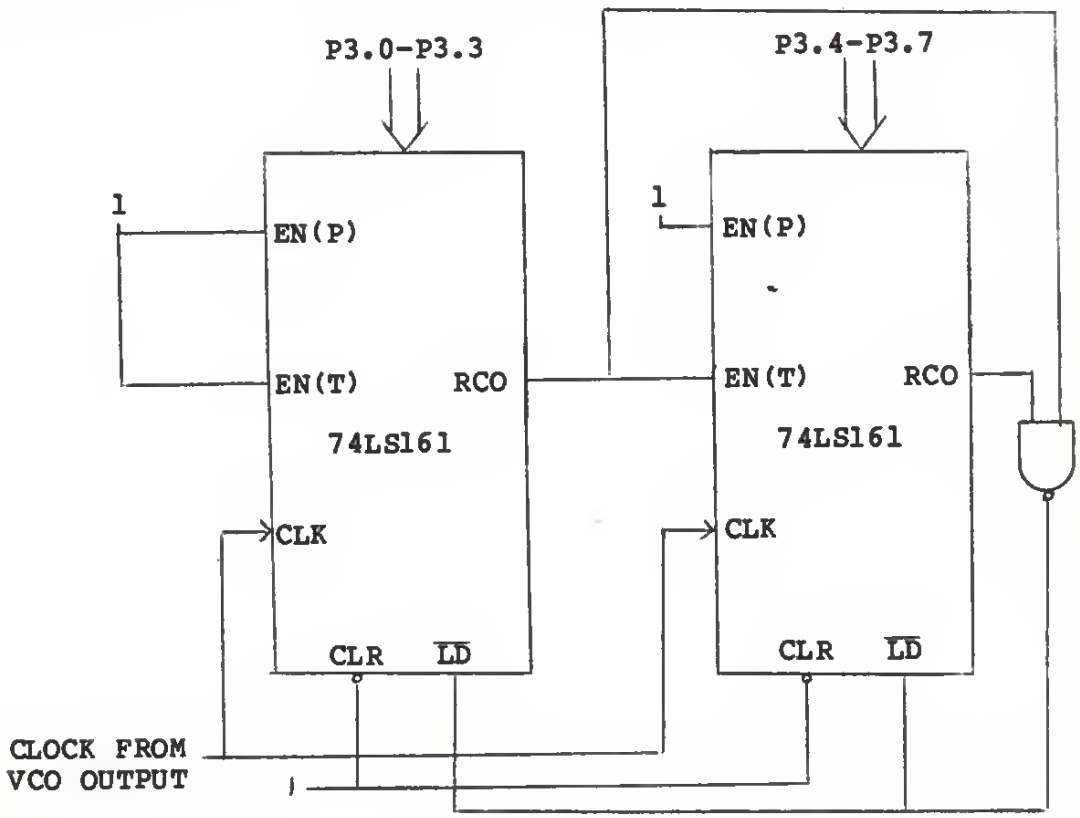


Fig. 27

CONFIGURATION OF THE 74LS161 AS A PROGRAMMABLE DIVIDER

So the process of locking the PLL in the sampling frequency range is done by sending control words from port 0 and 3 and also by suitably switching the frequency setting capacitor of the VCO . Bits 3,4 and 5 are used for switching the capacitor for suitable VCO frequency through a CMOS bilateral switch. Since port 0 and 3 are used as data bus and serial port during memory access and serial reception, tri-state buffers are used between these ports and the PLL and counters respectively. These buffers act transparently during frequency synthesis and are tri-stated during memory access by the 8751.

The various steps that are involved for frequency synthesis using the programmable PLL are as follows:

1) Switch capacitor through bits 3,4 or 5 of port 1 depending on the required output frequency.

2) Bit 2 of Portl is set to tri-state the bi-directional transceiver and I/O port which serve as data and address buffer during memory access by the 8751.

3) Bit 1 is set low to enable the tri-state buffers to latch control word from the port to programmable PLL and divider.

4) Control word from port 0 and 3 are sent for synthesizing the required frequency.

5) Bit 6 monitors the LD output of the PLL to check if loop has been locked. If LD is high, frequency locking has occurredand hence the sampling process could be started. If LD is low, frequency locking has not occurred and the sampling processis delayed until the loop is in lock.

Thus frequencies of the required range could be synthesized in coarse steps using the programmable PLL. The capacitors used in the VCO operated for usually two decades of frequency and hence with the use of 3 switching capacitors frequencies upto 10 MHZ could be synthesized.

With all the above requirements of the test-module in mind printed circuit board layouts were made. Due to the extra-hardware that was incorporated for the Direct-memory access and PLL sampling frequency control the required board size of 3 in by 5 in could not be achieved. However different types of layouts were tried out so that board area could be utilized to the maximum. Fig 28.a,b show the printed circuit board layouts of the ADC test module using the 8751 Microcomputer. Serial communication to the control computer using RS-232C interface is used.

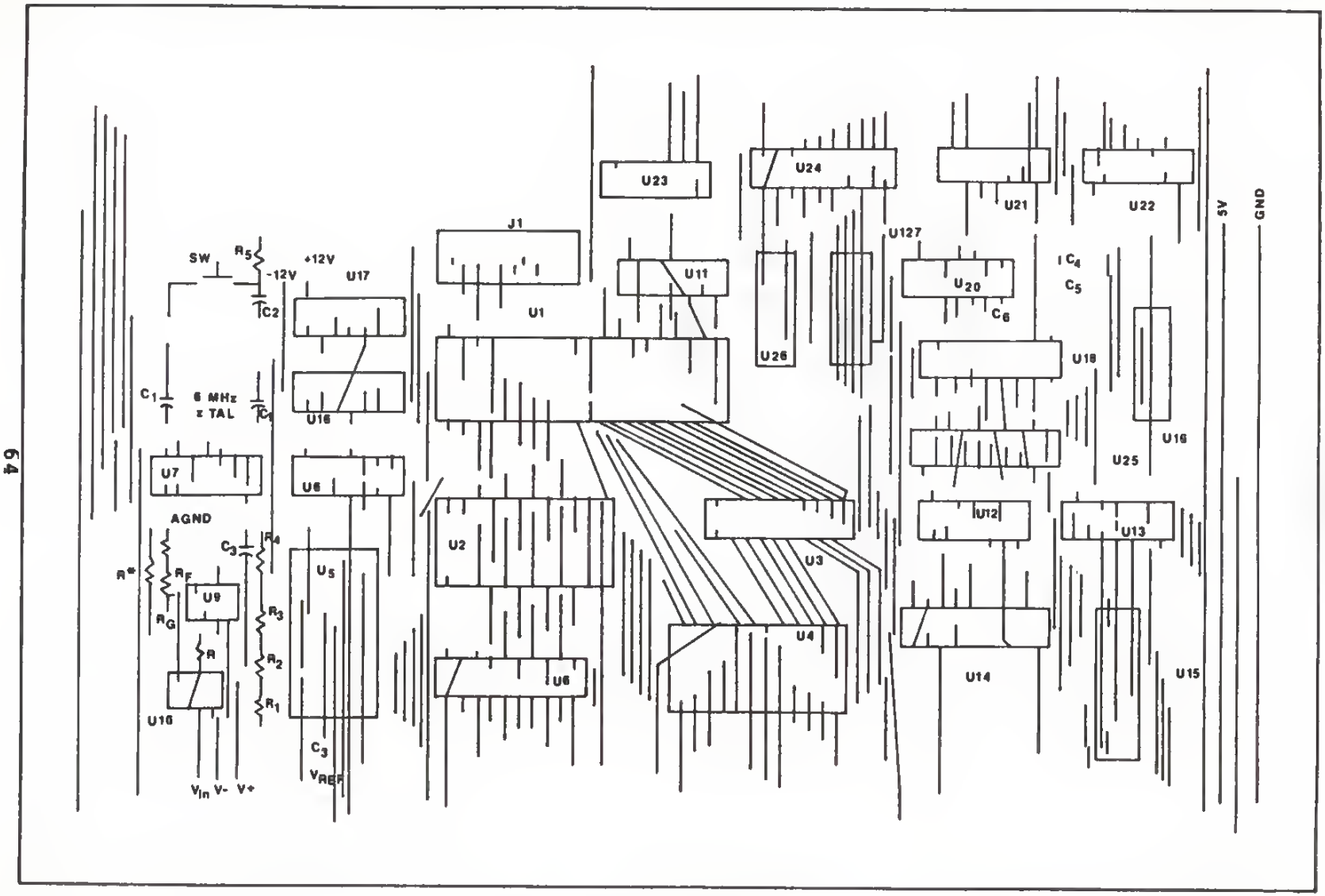


Fig. 28.a

PRINTED CIRCUIT BOARD LAYOUT OF THE INTEL-8751 BASED SYSTEM
(COMPONENT SIDE)

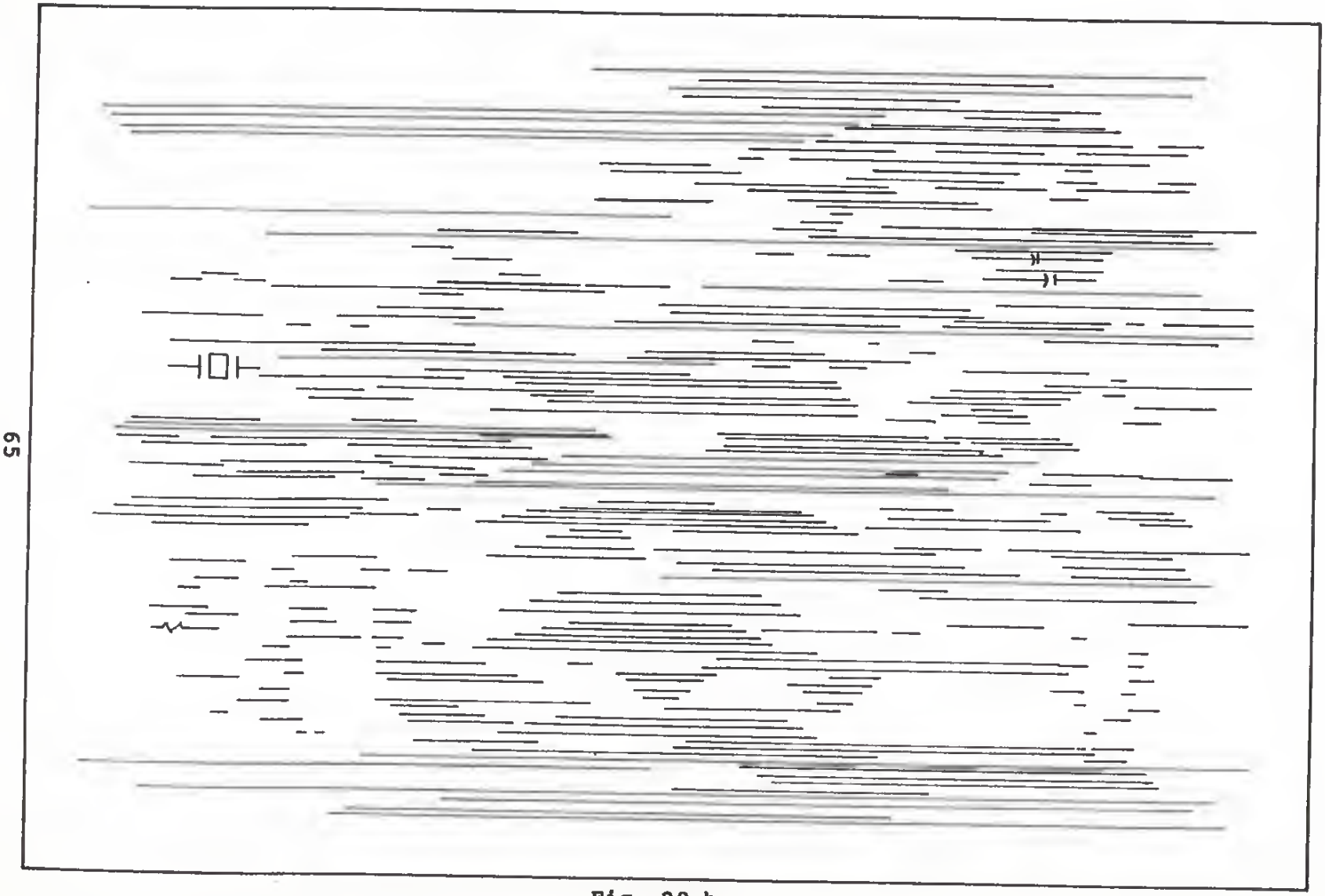


Fig. 28.b

PRINTED CIRCUIT BOARD LAYOUT OF THE INTEL-8751 BASED SYSTEM
(CIRCUIT SIDE)

4.9 AUTO-RANGING OF THE FLASH ADC.

The Flash ADC module has several input ranges like ±320 MV, ±3.2V, ±6.4V and ±9.6V. However the Flash ADC has a maximum reference of ± 6.4V and hence scaling of the input has be done so that the maximum reference voltage is not exceeded. Since the amplitude range of signal is not known it has be automatically set to a range so that the Flash ADC has a good resolution for the digital representation of the signal. Fig. 29 shows the digitized signal and explains why the auto-ranging is necessary.

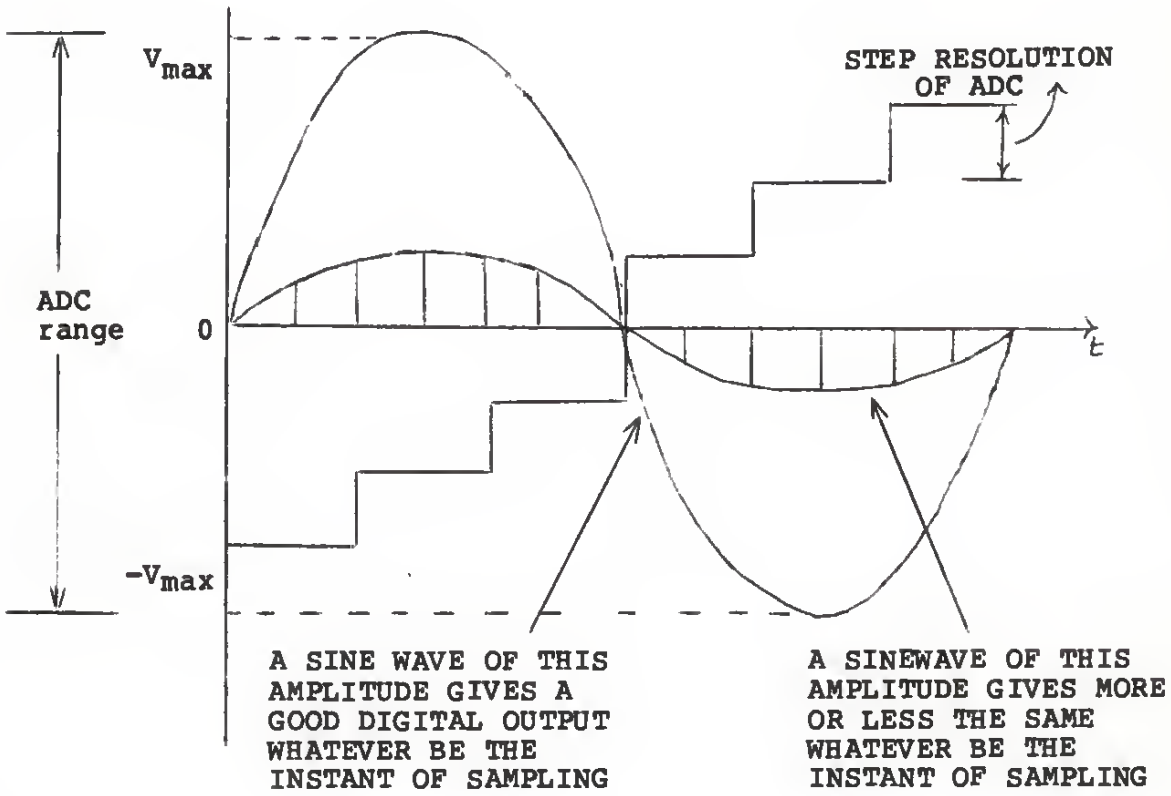


Fig. 29

DIAGRAM TO SHOW THE NEED FOR AUTO-RANGING AN ADC

Auto-ranging is accomplished by using signal conditioning circuitry as shown in Fig.30. As shown in the figure switches are used to scale the input voltage to less than the reference voltage. A resistor corresponding to maximum voltage rangeis initially closed on system reset. Then a finite number of samples say 128 are tested. If the initial voltage range fixed is very high which corresponds to constant digital words then next lowest range is checked similarly. Hence until the correct voltage range is fixed the real acquisition of data is not started. The software controls the whole auto-ranging process for make before break switching operation . The calculation of the various circuit components for signal conditioning circuitry is shown in Appendix H. Bits of Port 2.0-2.3 are used for resistance switching and P2.4 is used to check the counter for 128 samples. Appendix H shows how Port 2 is used to scale the input signal.

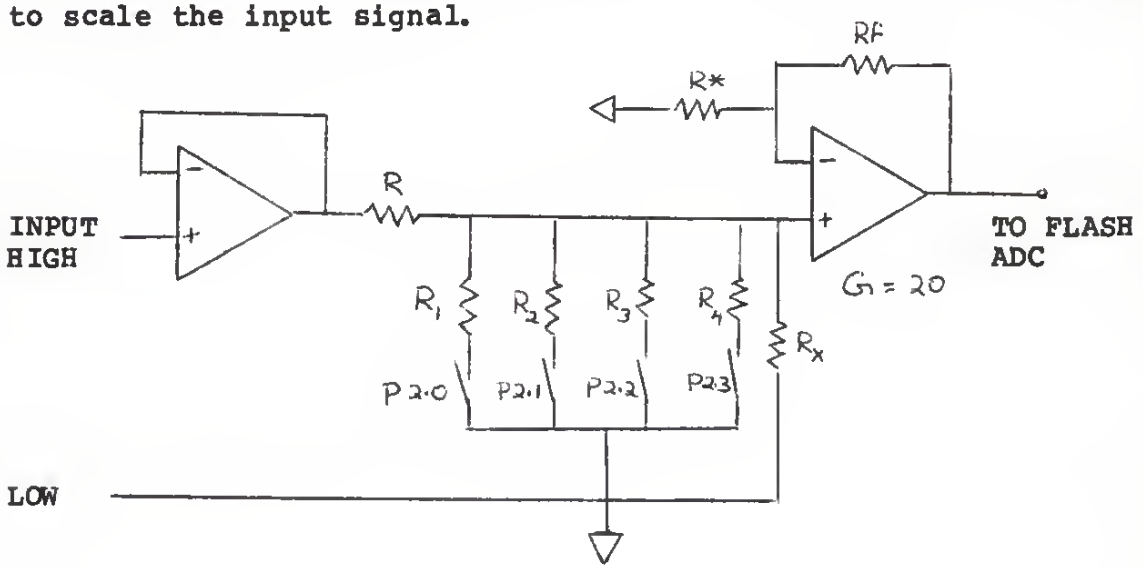


Fig. 30

SIGNAL CONDITIONING CIRCUITRY FOR AUTO-RANGING THE ADC

4.10 RS-232C STANDARD.

The signals going in and coming out of a processor are not adequate to communicate with a peripheral controlled by the processor. Thus signals are seldom sent directly to the interfaces. The terminal transfers characters of data in the ASCII form. Special hardware is used to form a standardized communication bus to which memory and peripherals are interfaced. A widely used standard wherein the terminal uses serial communication is the Electronic Industry Associates (EIA) specification RS-232 standard.

All RS-232 signals must be within the limits shown below.

1. +3 Volts to +15 Volts for a Zero.

2. -3 Volts to -15 Volts for a One.

Thus level translators are required between the TTL levels and the MODEMs. The level translators require additional power supplies (-12 V and +12V) to generate the required signal voltages. The two popular translators are:

1. MC 1488: Quad TTL to RS-232C. This for the data going to the MODEM.

2. MC 1489: QUAD RS-232C to TTL. This is for the data coming from the modem to the device.

The full RS-232 interface consists of 25 data lines; Most of these are undefined. In most computer terminals only 3 to 5 data lines are required for operation. The serial interface connections between the 8751 serial port and the computer data terminal is shown in Fig 29. Binary data available in the 8751 is converted to ASCII and sent to the data terminal of the computer

and ASCII characters sent by the computer to the 8751 are converted to binary for communication. ASCII to Binary conversion and Binary to ASCII conversion is done through software.

The serial interface connections between the 8751 and the Computer data terminal is shown in Fig. 31.

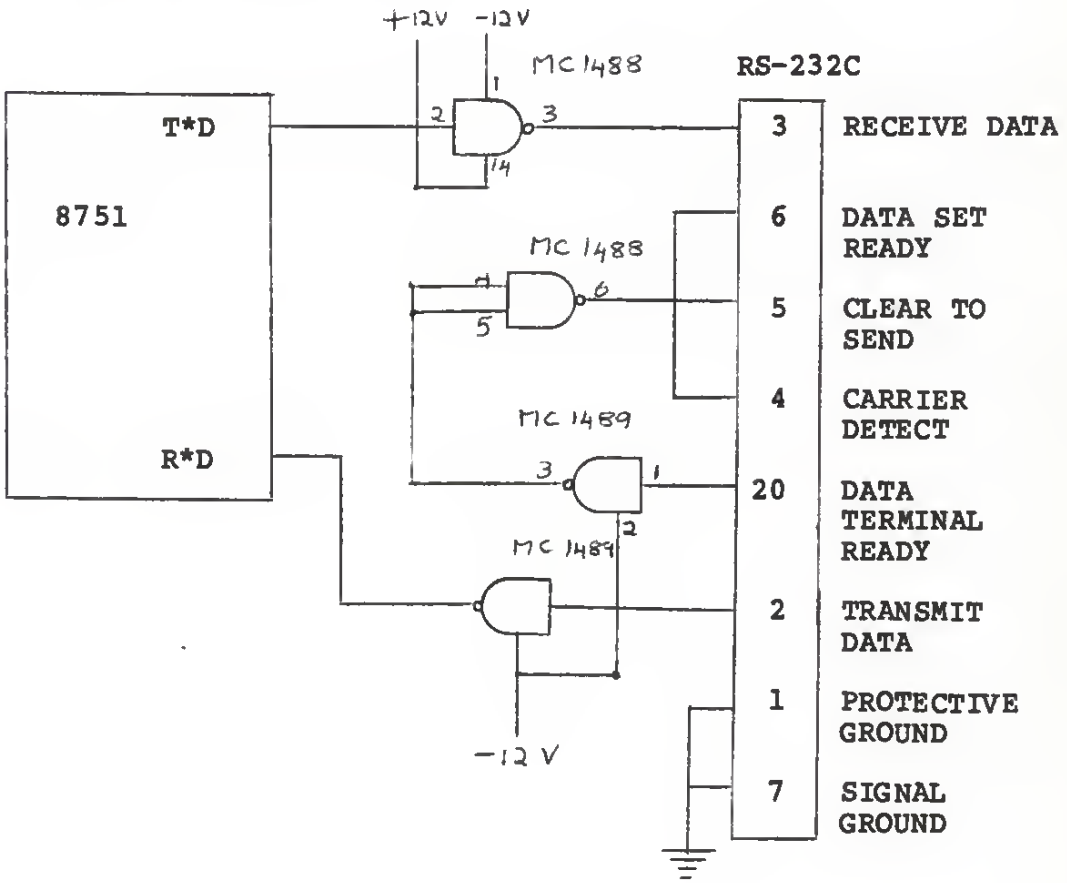


Fig. 31

RS-232C INTERFACE CONNECTIONS BETWEEN THE COMPUTER AND THE 8751

CHAPTER-5

5.1 SOFTWARE DESCRIPTION FOR THE 8751.

The heart of the Flash ADC test-module is the 8751 which acts as a controller for the system. The software designed for the Flash ADC test-module using the 8751 microcomputer basically serves four functions:

1) To communicate with the control computer through RS-232 serial interface,

2) To synthesize the clock of required frequency for the Flash ADC and binary counter through a programmable PLL,

3) To control the process of data acquisition through direct memory access (DMA), and

4) To automatically vary the input range of the signal so that an accurate digital representation of the input signal could be obtained by switching resistors from highest range to the lowest range.

The BASIC program that invokes the test module is shown in Appendix E.

COMMAND FORMAT FOR INVOKING THE FLASH ADC TEST MODULE.

```
    PRINT #1, "A$R$FF$"
 70 DIM D(2048)
190 FOR L= 1 TO 2048
200 INPUT #1,D$
210 D(L)=VAL(D$)
220 NEXT L
```

This command will cause the computer to receive 2048 (2K) samples of signal from the test module. The test module expects the command "A" to start the data acquisition. "R" specifies the range of frequency in the range of 1-3, where 1

corresponds to a frequency range of 100HZ-10KHZ, 2 corresponds to the range 10KHZ-1MHZ, 3 denotes 1-8MHZ range. The control word "FF" denotes the sampling frequency word 0 to 99 in the range R. Through specification of R and FF the exact sampling frequency is fixed. For example with R=1 and FF= 50 the sampling frequency of the required clock is fixed at 5KHZ. The frequency resolution that could obtained in ranges 1,2 and 3 are 100HZ, 10 KHZ and 90K HZ respectively within reasonable limits.

5.2 DESIGNING THE 8751 MONITOR PROGRAM.

The main driver routine initializes the serial portto receive a character at 9600 baud rate.On receiving thecommand, range, and control words from the computer the PLL is programmed to synthesize a clock of required sampling frequency range. On detection of phase lock, 128 samples of signal are sampled to test the input range. Initially the highest input range is selected so that the input is protected. All the samples are individually tested for the range. If the highest range is found to be unsuitable, the next lowest range is fixed and the signal is once tested for the correct range. Real data acquisition is not started until a suitable input range is fixed. The various subroutines that are called by the driver routine and their functions are summarized below:

1) GETCHAR: This routine enables the receiver of the serial port SBUF to receive a character from the computer and store it in the accumulator,

2) PRPLL: To program the PLL through Ports 0 and 3 to achieve the required sampling frequency as indicated by the input statement

of the BASIC program,

3) SETUP: This routine is used to configure Port 1 and 0 as outputs except bits 6&7 of Portl. The bi-directional transceiver and I/O port are tristated and the latch enable of the tri-state buffers enabled to program the PLL and control DMA,

4) READ: This routine is used to read the ASCII character got from GETCHAR convert it to binary code and store in temporary locations for frequency range control,

5) SWCAP: This is used to switch the capacitor for controlling the VCO output frequency of the PLL and also to fetch the data-pointer for required control words to the PLL and programmable divider,

6) CTRTBL: To output control words to the PLL and programmable divider for locking the PLL in the sampling frequency range,

7) COSEND: This routine calls house-keeping routines such as TRIST, CONVERT, TRANSMIT, and SENDCR to send the stored data to the computer,

8) TRIST: This routine stops data acquisition through Port 1 and reconfigures Port 0 and 3 as address/data bus and serial port respectively,

9) CONVERT: This routine converts the binary data to BCD and then to ASCII before sending them to the computer,

10) SUB8: is the 8 bit subtraction routine to convert binary data to BCD digits,

11) TRANSMIT: is the routine to transmit the ASCII coded digits through SBUF,

12) SENDCR: This routine sends a carriage return character to the

computer after every data sample is sent,

13) AUTORANG: This routine does the range switching of the input so that a good digital representation of signal is obtained,

14) TRANS: A character corresponding to the switched resistor is sent to the computer for fixing up the resolution of the Flash ADC,

15) BOOKEEP: 128 samples of signal are stored in memory through this routine for checking the input range, and

16) TIMOUT: A time delay routine to give allowance for the resistance switching.

The flowcharts for the logic design of the software are shown in Appendix F. The monitor program listing for the 8751 that is to be coded from location zero of the internal EPROM is shown in Appendix G.

5.3 SERIAL PORT AND TIMER CONFIGURATION.

Configuring the 8751's serial port for a given data rate and protocol requires essentially three sections of software. On power-up or hardware reset the serial port and timer control words must be initialized to appropriate values. Additional software is needed to unload data as it is received in the GETCHAR routine and load serial port data register in the TRANSMIT routine.

To communicate with the Computer, the serial port is initialized to an 8 bit UART in Mode 1 configuration through SCON, TMOD and TCON registers. Timer 1 is configured in the auto-reload mode (TMOD.5=1, TMOD.4=0) as a baud rate generator. The Timer must be running (TCON.6=1) and to keep the overflows from

generating unnecessary interrupts Timer/Counter 1 interrupt is disabled (IE.3=0). To achieve a data rate of 9600 baud the timer1 would automatically reload a constant values into the timer through software. The Timer 1 must divide the (8/12) MHZ clock by ((8/12)*10**6)/(32*9600) which equals 2 instruction cycles approximately. The Timer TH1 must reload the value -02H as shown in the initialization routine SPINIT.

CHAPTER-6

6.1 AN EVALUATION OF THE TWO DESIGNS OF THE TEST MODULE.

Two designs of a microprocessor based test module were made for Flash ADC applications. With the use of such a module samples of the input signal could be obtained and sent to the host-computer for subsequent processing. Signal-processing techniques like Adaptive filtering, FFT and Correlation algorithms can be used for digital phase measurement methods. Also the Flash ADC will enable the host computer to emulate a digital oscilloscope.

The first design using the NSC 800 Microprocessor has limited application due to hardware and space constraints. The fast ADC module could acquire 2K data samples under program control at a fixed sampling frequency of 100 KHZ. The input range is fixed at $\pm$6.4V. Due to software controlled data acquisition, data is read once in 10 secs thus reducing the effective sampling rate to 100 KHZ. This design is however an improvement of the available system for low power application. Through the use of optical fibers, noise-free communication is achieved.

It was found that if higher sampling rates were to be achieved, direct memory access should be used by which data from the Flash ADC available in one clock cycle is directly stored in memory through the use of control logic and extra hardware. With the use of the Intel 8751 an auto ranging test module could be built for a variable sampling frequency option. Variable sampling

frequency is achieved through a programmable PLL controlled by the software. Auto-ranging is also done through the control of the processor for more accurate measurement. It is assumed that the sampling frequency for the Flash ADC is known to the user beforehand so that the BASIC program which initiates the sampling also sends the required control word for synthesizing the clock of sampling frequency. However if sampling frequency is not known to the user then a procedure similar to the amplitude scaling should be done for setting the correct frequency range. Only the software would have to be modified for doing the auto-ranging of sampling frequency. The only problem that could be expected would be to find a way of setting the sampling frequency at least twice as great as the input frequency. A sufficiently large number of samples would be needed in order to get enough information on the input signal frequency.

6.2 SYTEM IMPLEMENTATION FOR PHASE MEASUREMENT.

This module could be used for measuring the phase difference between two continuous time signals. Actual data-samples of the two signals are acquired by the module and sent to the host computer. Suitable algorithms like adaptive filtering, FFT, Correlation could be used for processing the actual data samples for measurement of phase. Since two signals have to be sampled synchronously for accurate measurement a set up as shown in Fig.32 would be necessary. Since the test module has only one ADC, a zero crossing detection technique could be used to sample the two signals alternately through a trigger reference. With a highly stable trigger reference it would look as if the signals

are sampled synchronously. The property of the two signals that is exploited in this method is that x(t) and y(t) are continuous time signals and periodic. Hence the signals by themselves do not change with time and with respect to themselves.

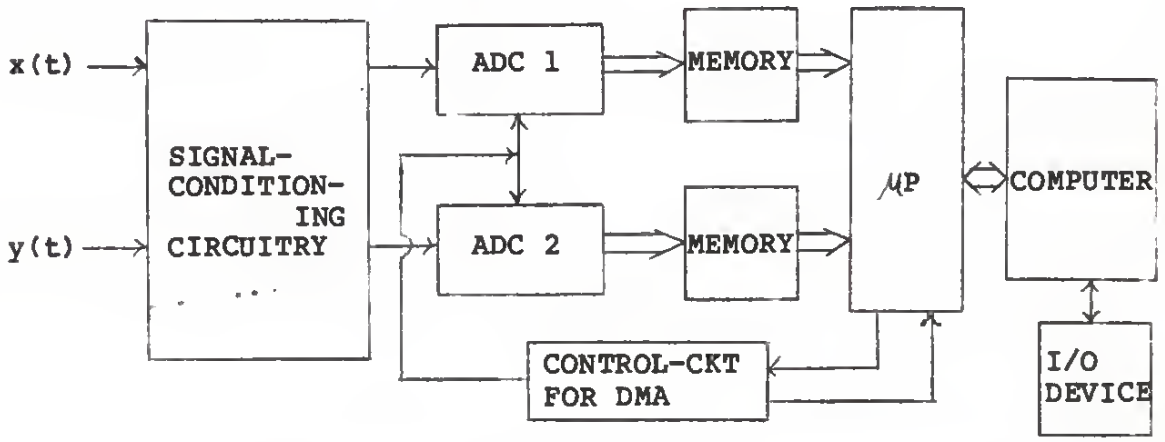


Fig. 32.a SYSTEM IMPLEMENTATION FOR PHASE MEASUREMENT

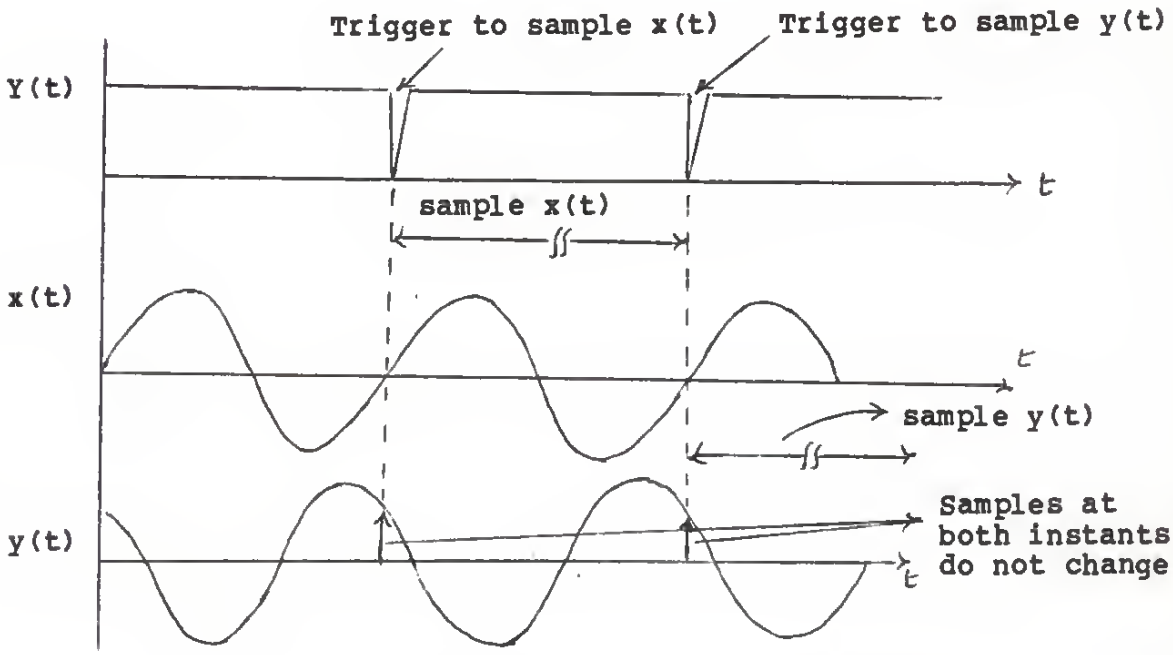


Fig. 32.b ZERO CROSSING DETECTION TECHNIQUE

6.3 DEVELOPMENT OF A NEW INSTRUMENTATION SYSTEM.

Test modules like the Flash ADC, precision ADC, Waveform generator, DVM, programmable power supply are some of the few instrumentation modules that are required for laboratory measurement. However with the development of optical fibers, these modules could be placed next to the circuit under test for measurement. With the increase of optopods each module would require a fiber optic transmitter and receiver. However a simpler method could be suggested using a multiplexed optical fiber link. The arrangement for the system would look as shown in Fig. 33.

Hewlett-Packard has come up with the 39301 time-division data multiplexer. This full duplex part operates independently of the asynchronous protocols of any CPU or peripheral because its time division multiplexing allows each of its 16 channels to operate by protocol-independent signal sampling. With such a multiplexer at both ends 16 modules could be able to communicate with the CPU through a time multiplexed optic fiber cable. Fig. 33- shows the arrangement of the new instrumentation system using optical fibers.

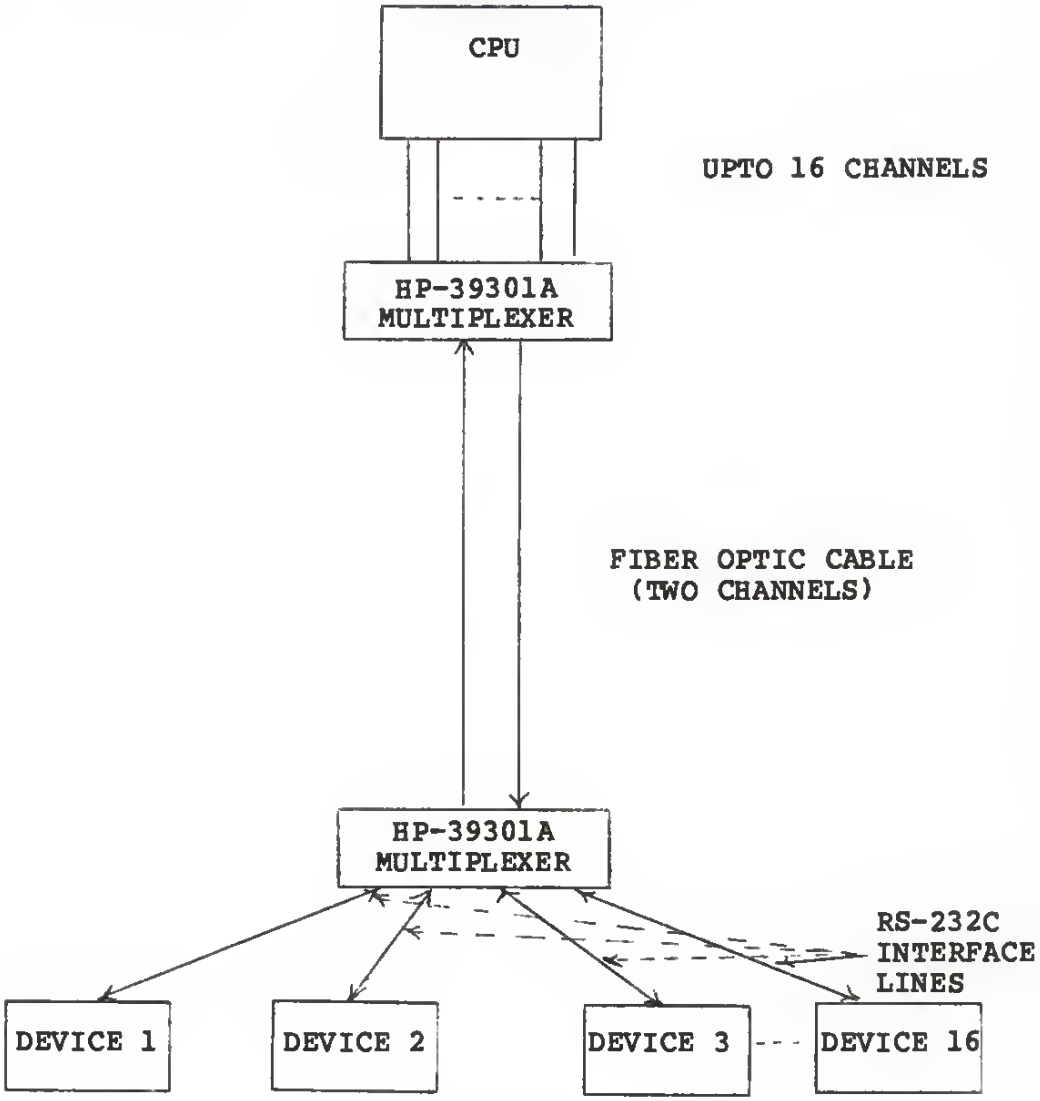


Fig. 33

BLOCK DIAGRAM FOR THE DEVELOPMENT OF A NEW INSTRUMENTATION SYSTEM USING OPTIC FIBER CABLES

CHAPTER-7

CONCLUSION

Before attempting to conclude we must make a comparative study on the two design methods. Both the designs were made after careful consideration with regard to high speed data acquisition and communication, low power consumption, and for small sized modules. This module could be used for emulating a digital oscilloscope, Phase measurement between two signals and another potential application is in sensor systems in industrial robotics.

With the objective of improving the data-communication between the Instrumentation-module and host-computer optical-fiber communication is used in the first method instead of the conventional RS-232 interface; also a data rate of 1 Megabaud could be achieved by this method. The same method could be used for the second design using the 8751. With the 8751 Microcomputer operating at a clock frequency of 8 MHZ the serial interface could operate at a maximum baud rate of 666.6 K baud. With the use of a faster memory of access time 80 nsecs, a 12 MHZ clock could be used for the 8751 for memory access. With a 12 MHZ clock optical-fiber communication at 1 Megabaud could also be achieved. However for design purpose, RS-232C interface for a baud rate of 9600 is used.

As far as power consumption is concerned the first design used the NSC 800 Microprocessor with all CMOS components. This would contribute to the overall low power consumption as indicated in Appendix A. With regard to space requirements

various design were tried out for efficient layout and a reasonable size has been achieved with two boards of 3 in by 5 in each which are connected by board connectors. A complete study has been made on the software in the second design for direct implementation. The design of software for the Optopod using the NSC 800 depends largely on the interface to the computer and hence only a partial study on the software design was attempted. The second design using the 8751 is an auto-ranging module with options of variable sampling frequency as required by the user and auto-amplitude scaling of input signal. Since the second design is more versatile than the first it is the one which is recommended for implementation.

APPENDIX-A

BOARD LAYOUT AND POWER CONSUMPTION DETAILS OF THE SYSTEM

****

NSC 800 SYSTEM.
****

Printed circuit board layouts of the NSC 800 based system are shown in Fig. 8. Two boards of size 3.6 in by 5 in are used. Effective area utilized in these boards = Chip cell size area =10 $inch^2$. Power consumption of the system = 1000 mW (approx) at 4 MHZ.

****

INTEL-8751 SYSTEM.
****

Printed circuit board layouts of the 8751 based system are shown in Fig. 28.a, 28.b. Board size of 6 in by 10 in is used. Effective area utilized in this board = 15 in^2. A double board of size 3 in by 5 in could be possibly made for smaller sized module.

****

APPENDIX-B

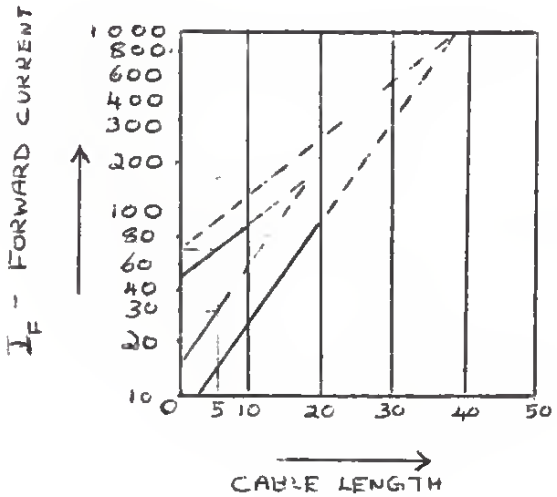
CALCULATION OF CURRENT LIMITING RESISTOR FOR OPTIC FIBER CABLE.

From the specification manual of the HFBR-0500 optic fiber cable it was found the typical forward voltage V_f for the transmitter is 1.5 volts. Also from the graph of fiber cable length versus I_C, the forward current requirement corresponding to 5 meter length cable in the range of 0-70 C is around 40 mA. Hence the current limiting resistor for the transmitter for a forward voltage V_f=1.5 Volts is given by the equation

$$R = (V_{CC} - V_f)/ I_f.$$

For V_{CC} = 5 V, V_f = 1.5 V and forward current I_f = 40 mA

$$R = (5-1.5)/.04 = 87.5 \text{ ohms.}$$



APPENDIX-C

```
**********************************************************************
BASIC  PROGRAM  FROM CONTROL COMPUTER TO GET 2K SAMPLES OF  INPUT
SIGNAL.  THIS PROGRAM INITIATES THE  OPTOPOD TO SAMPLE THE SIGNAL
AT A  FIXED  SAMPLING  FREQUENCY OF 100 KHZ IN THE INPUT RANGE OF
±6.4 VOLTS. THE NUMBER OF SAMPLES COULD BE VARIED AS  REQUIRED BY
THE USER BY CHANGING THE ARRAY CONSTANT.
**********************************************************************

          10   DIM D(2048)
          20   PRINT #1,"A"
          30   FOR L=1 TO 2048
          40   INPUT #1,D$
          50   D(L)=VAL(D$)
          55   D(L)=D(L) - 128
          60   NEXT L
          70   FOR L=1 TO 2048
          80   D(L)=D(L)*.05    \ REM THE INPUT RANGE IS ± 6.4V.
          90   PRINT L,"TH SAMPLE=",D(L)," ",
         100   NEXT L
         110   END
**********************************************************************
```

APPENDIX-D

A SAMPLE PROGRAM IN Z-80 ASSEMBLER CODE FOR THE NSC 800 SYSTEM

```
;*****************************************************************
;IN THIS PART OF THE PROGRAM DATA FROM ADC IS READ BY THE        *
;NSC 800 MICROPROCESSOR AND STORED IN MEMORY ON REQUEST FROM THE*
;CONTROL COMPUTER.                                               *
;*****************************************************************
          ORG       0000 ; MONITOR PROGRAM STARTS AT LOCN ZERO
CNST1     EQU       07FFH ; NUMBER OF DATA SAMPLES
PNTR1     EQU       2000H ; POINTER TO STARTING LOCATION IN MEM
RTHR      EQU       4000H ; RECEIVE/TRANSMIT HOLD REGISTER
RMDR      EQU       4001H ; RECEIVER MODE REGISTER
TMDR      EQU       4002H ; TRANSMIT MODE REGISTER
GLMR      EQU       4003H ; GLOBAL MODE REGISTER
COMR      EQU       4004H ; COMMAND REGISTER
RTSR      EQU       4008H ; RECEIVER/TRANSMITTER STATUS REGISTER
;
START     LD        HL,CNST1
          LD        IX,PNTR1
          LD        A,30H
          LD        (RMDR),A ; RECEIVER SET TO RECEIVE 8 CHARS
;                            ; WITH NO PARITY
          LD        (TMDR),A ; SAME WITH TRANSMITTER
          LD        A,00H
          LD        (GLMR),A ; SET UP CLK FACTOR,NO OF STOP BITS
          LD        A,01H
          LD        (COMR),A ; ENABLE RECEIVER
          SCF
          CCF
AGAIN     LD        A,(RTSR) ; CHECK IF CHARACTER IS RECEIVED
          RRC       A
          JP        NC,AGAIN ; IF NOT CHECK AGAIN
          LD        A,(RTHR) ; RECEIVE CHARACTER
          CP        41H      ; IF YES CHECK IF "A" IS RECEIVED
          JP        Z,SAMPLE
          JR        AGAIN
SAMPLE    LD        A,(6000H) ; SAMPLE SIGNAL FROM ADC
          LD        (IX),A    ; STORE SAMPLE IN MEMORY
          INC       IX
          DEC       HL
          JP        NZ,SAMPLE ; UNTIL 2K SAMPLES ARE STORED
;
;*****************************************************************
; THIS IS THE TRANSMIT ROUTINE WHERE DATA STORED IN MEMORY IS    *
; SENT TO THE INTERFACE (COMPUTER). NO ERROR CHECKING IS DONE    *
;*****************************************************************
TRANS     LD        HL,CNST1
          LD        IX,PNTR1
          LD        A,02H
          LD        (COMR),A
```

```
NEW         SCF
            CCF
            LD          A,(IX)
BACK        LD          A,(RTSR)
            RRC         A       ; CHECK IF TRANSMIT BUFFER IS EMPTY
            JP          NC,BACK ; IF NOT GO BACK
            LD          A,(IX)
            INC         IX
            LD          (RTHR),A
            DEC         HL
            JP          NZ,NEW ; TRANSMIT DATA IF 2K SAMPLES NOT SENT
            JR          START
            END
*****************************************************************
ERRORS=0000
```

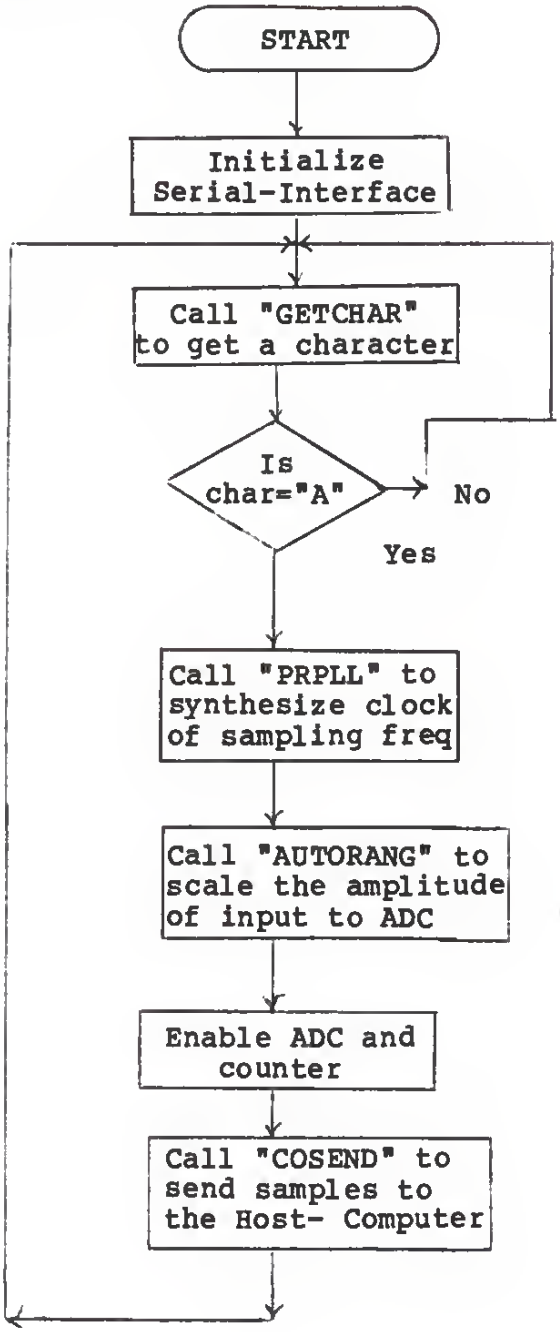
APPENDIX-E

BASIC PROGRAM TO INVOKE THE AUTO-RANGING TEST MODULE

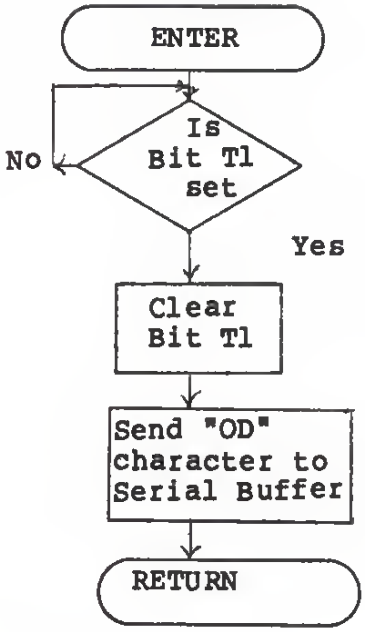
```
100 REM*************************************************************
110 REM THIS BASIC PROGRAM FETCHES 2K SAMPLES OF THE SIGNAL FROM
120 REM THE TEST MODULE. THE REQUEST FOR SAMPLING THE SIGNAL IS
130 REM INVOKED BY RECEIVING THE CHARACTER "A" IN THE MONITOR
140 REM PROGRAM. ACTUAL RANGE OF SAMPLING FREQUENCY IS FIXED BY
150 REM "R" & "FF" COMMAND.
151 REM THE VOLTAGE VALUE IS FIXED BY THE RETURNING VALUE OF
152 REM  SWITCH CLOSED i.e.BY RECEIVING A CHARACTER "X$" FOR THE
153 REM  DIFFERENT VOLTAGE RANGES ± 320 mV,± 3.2V,± 6.4V,± 9.6V.
160 REM *************************************************************
170 DIM D(2048)
110 PRINT #1,"A$R$FF$"
115 REM *************************************************************
120 PRINT " R DENOTES FREQUENCY RANGES 1 TO 3"
130 PRINT " RANGE 1 CORRESPONDS TO 100-10KHZ, 2 CORRESPONDS TO"
140 PRINT " 10KHZ-1MHZ, 3 CORRESPONDS TO 1-10MHZ"
150 PRINT " FF DENOTES SAMPLING FREQUENCY WORD IN RANGE OF 0-99"
160 PRINT " IN THE SPECIFIED RANGE R. 0 REFERS TO MINIMUM AND"
170 PRINT " 99 REFERS TO MAXIMUM. Eg. FF=50 IN RANGE R=1 REFERS"
180 PRINT " TO 5KHZ"
181 REM *************************************************************
182 INPUT #1,R$
183 IF ASC(R$)=31 THEN RS=.0025 \ INPUT RANGE IS ±320 mV.
184 IF ASC(R$)=32 THEN RS=.025  \ INPUT RANGE IS ±3.2 V.
185 IF ASC(R$)=33 THEN RS=.05   \ RANGE IS ± 6.4V.
186 IF ASC(R$)=34 THEN RS=.075  \ RANGE IS ± 9.6V.
190 FOR L=1 TO 2048
200 INPUT #1,D$
210 D(L)=VAL(D$)
220 NEXT L
230 FOR L=1 TO 2048
240 D(L)=D(L)-128
250 D(L)=D(L)* RS \  SCALE THE DIGITAL WORD TO VOLTAGE VALUE
260 PRINT L,"TH SAMPLE =", D(L)," ",
270 NEXT L
280 END
*************************************************************************
```

APPENDIX-F

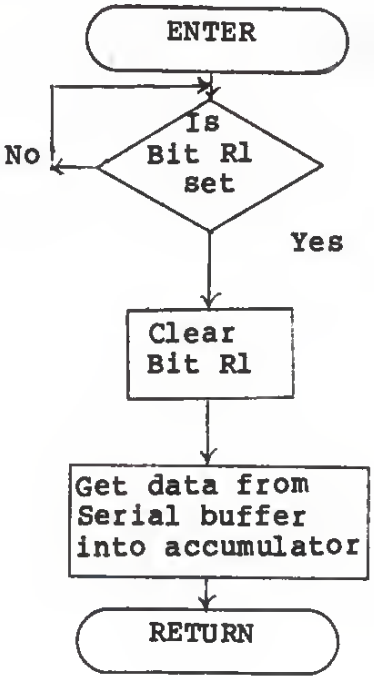
MAIN PROGRAM FLOWCHART.



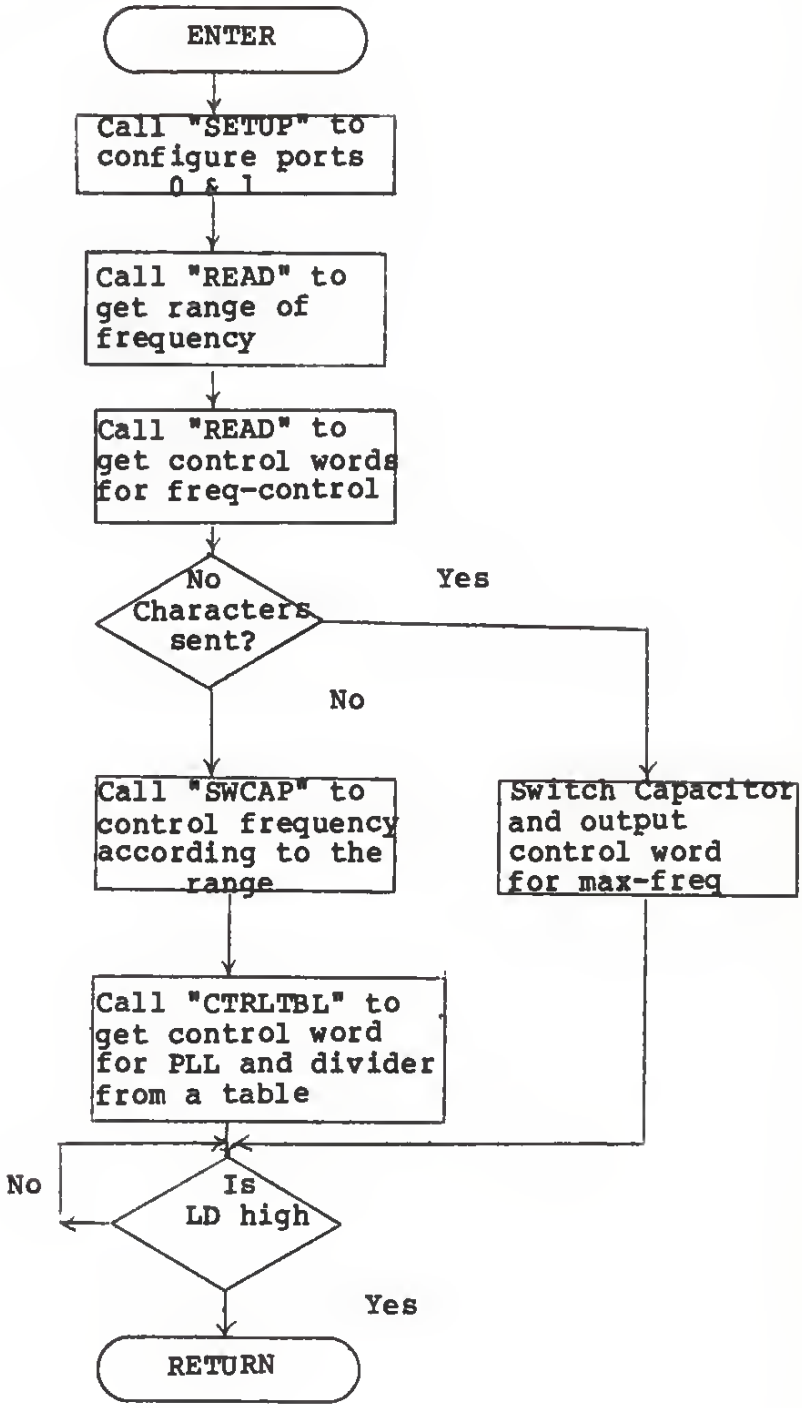
SENDCR ROUTINE FLOWCHART.



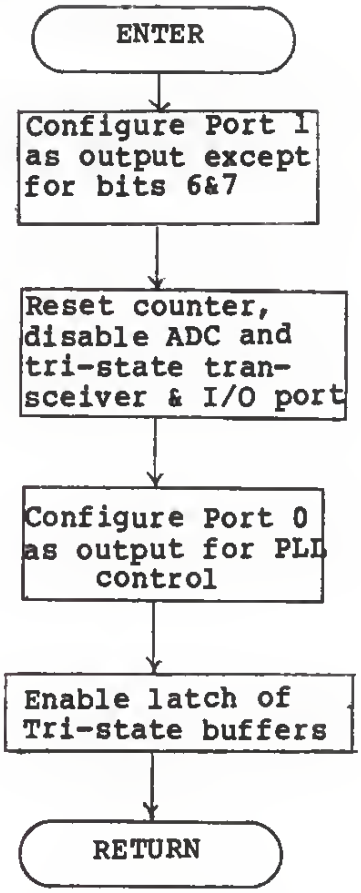
GETCHAR ROUTINE FLOWCHART.



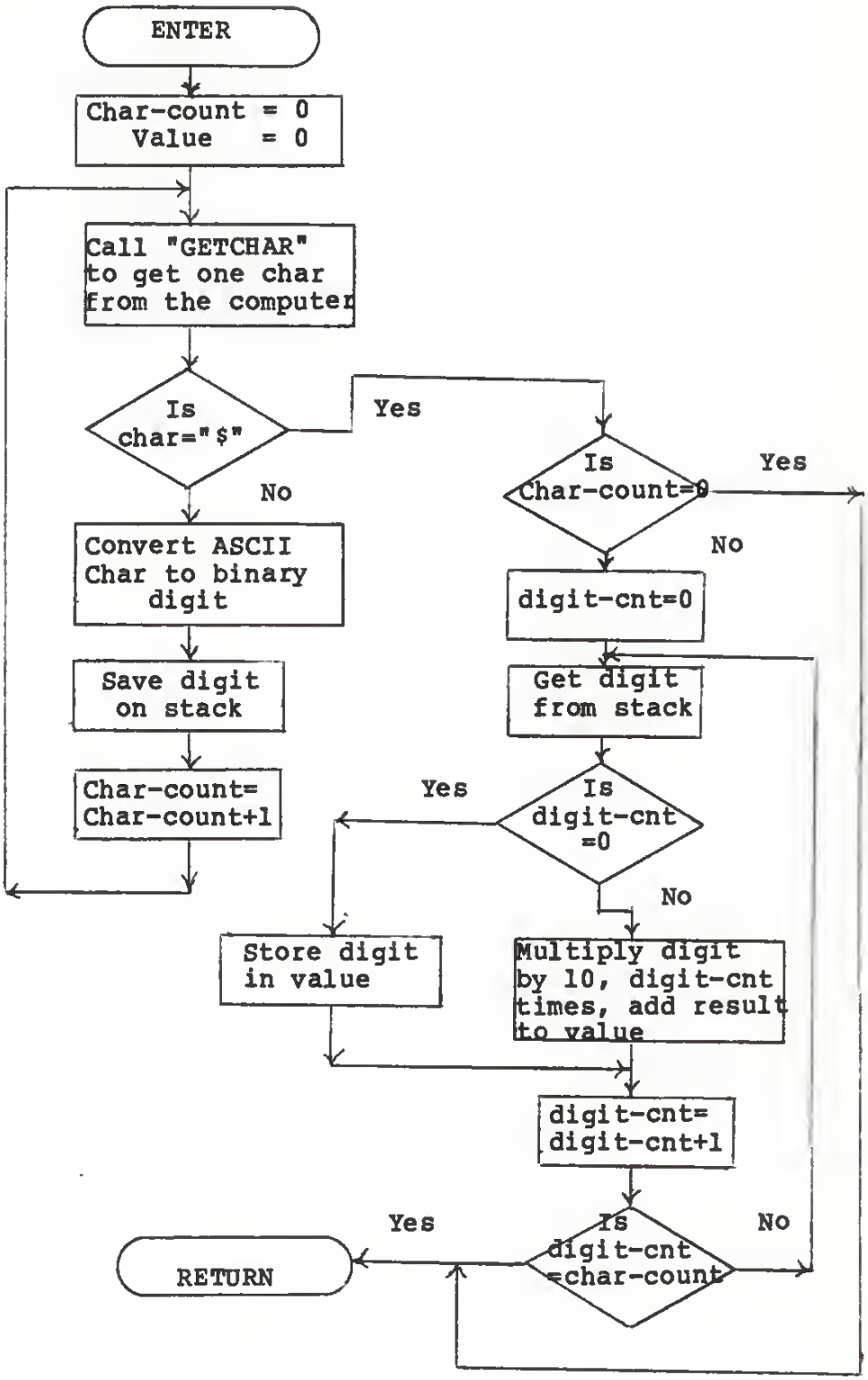
PRPLL ROUTINE FLOWCHART.



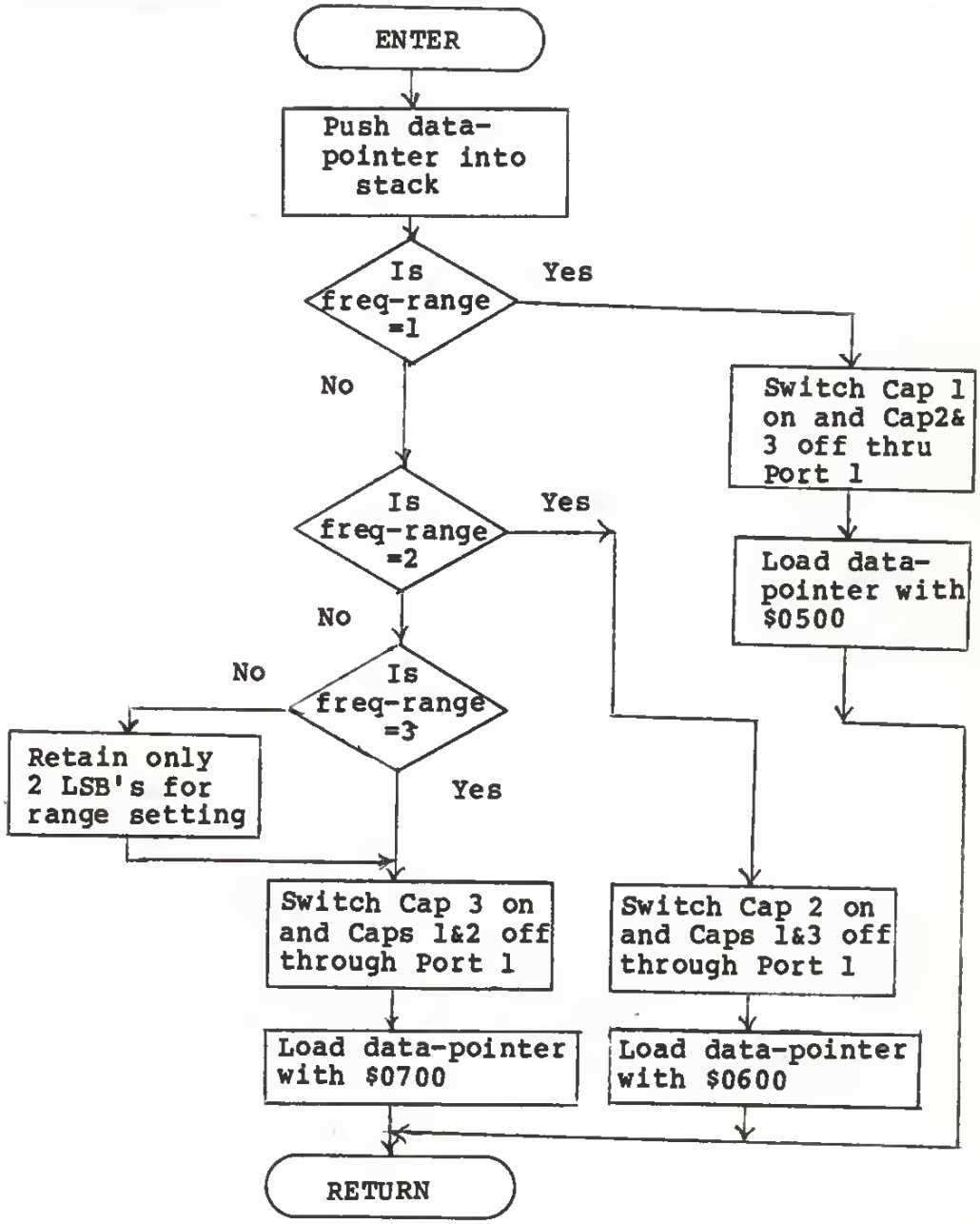
SETUP ROUTINE FLOWCHART.



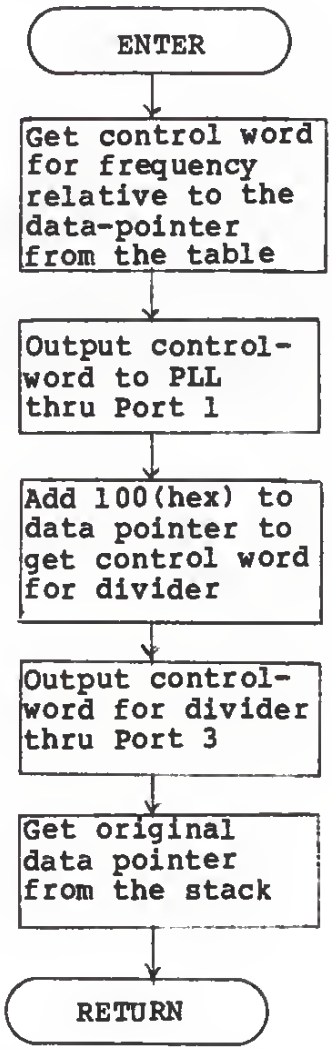
READ ROUTINE FLOWCHART.



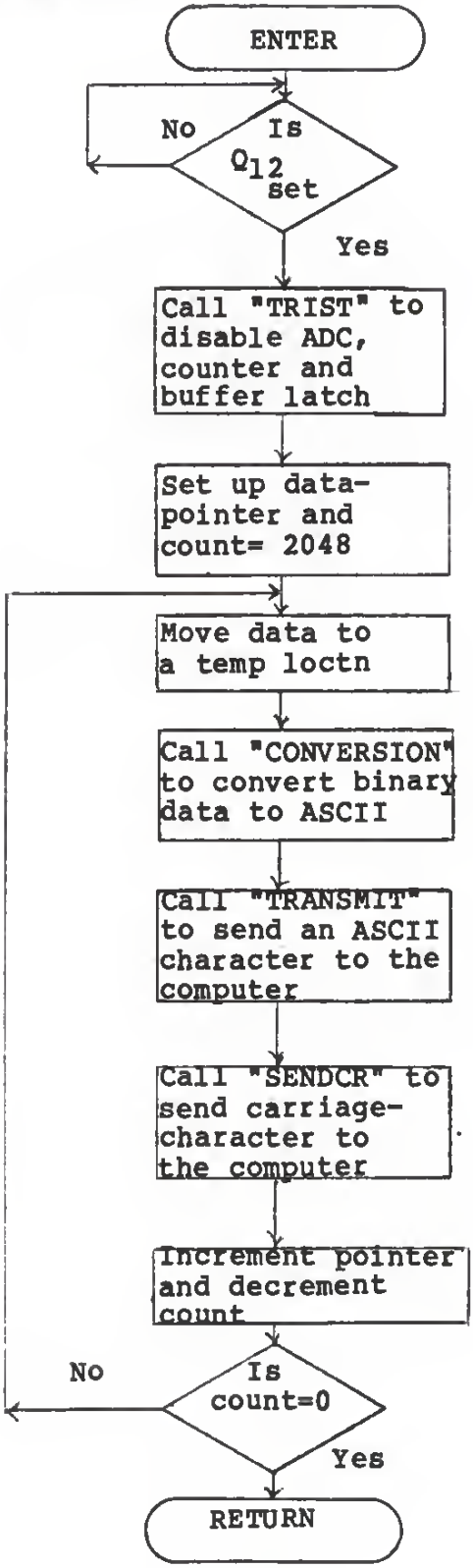
SWCAP ROUTINE FLOWCHART.



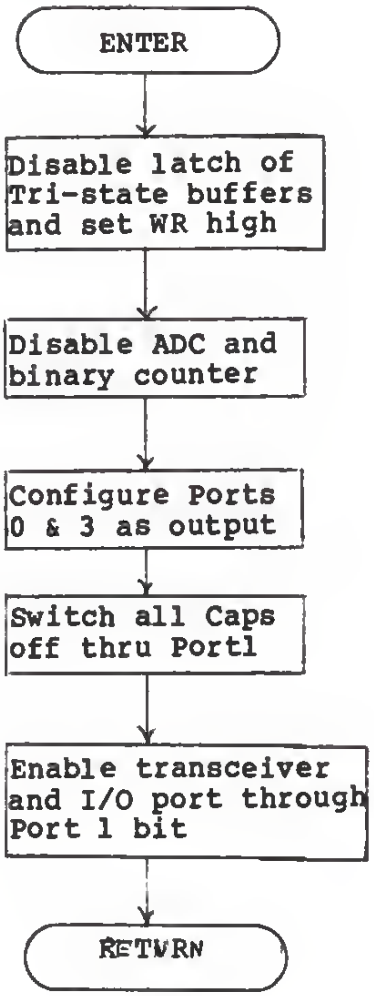
CTRLTBL ROUTINE FLOWCHART.



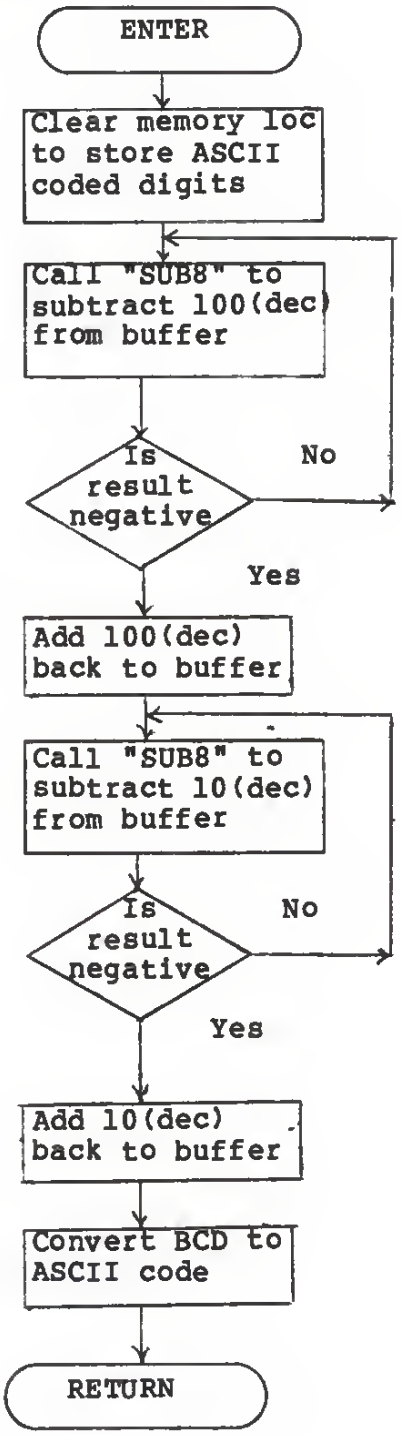
COSEND ROUTINE FLOWCHART.



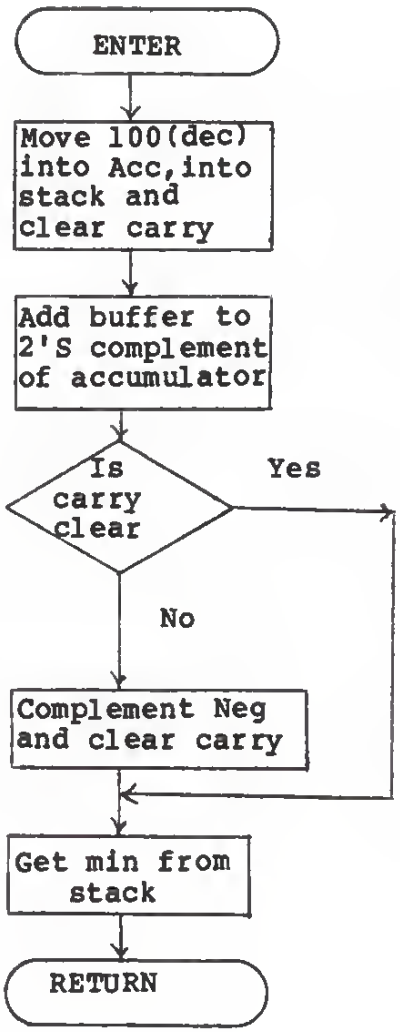
TRIST ROUTINE FLOWCHART.



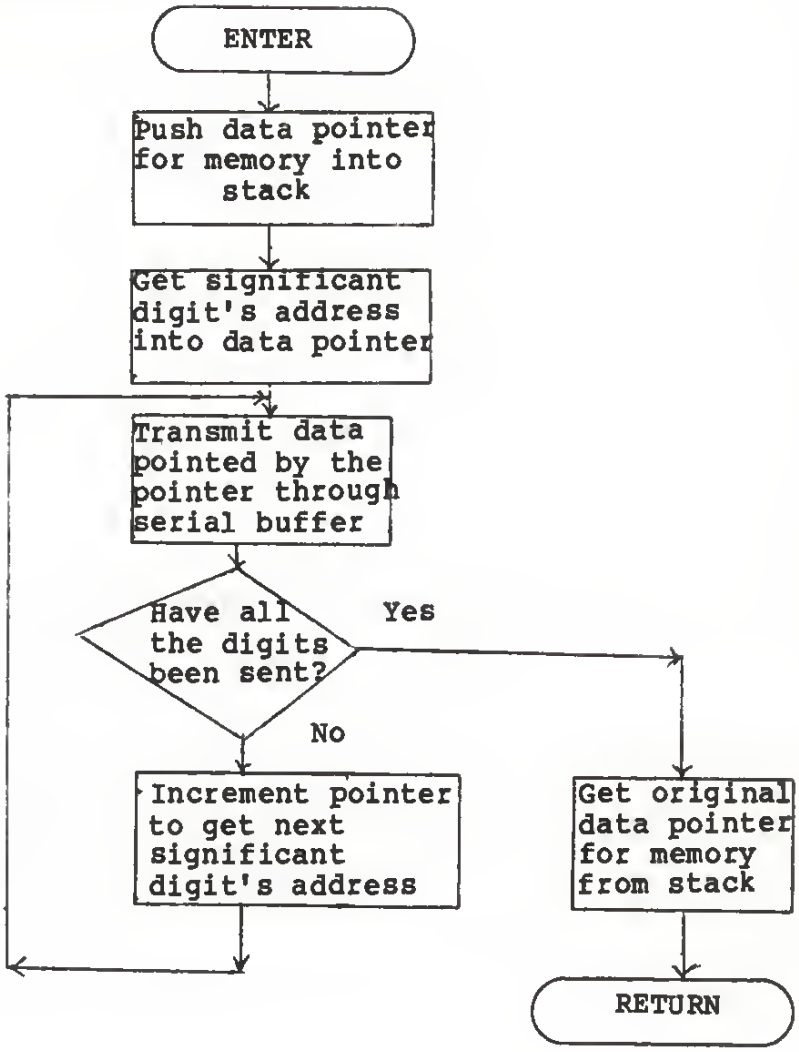
CONVERSION ROUTINE FLOWCHART.



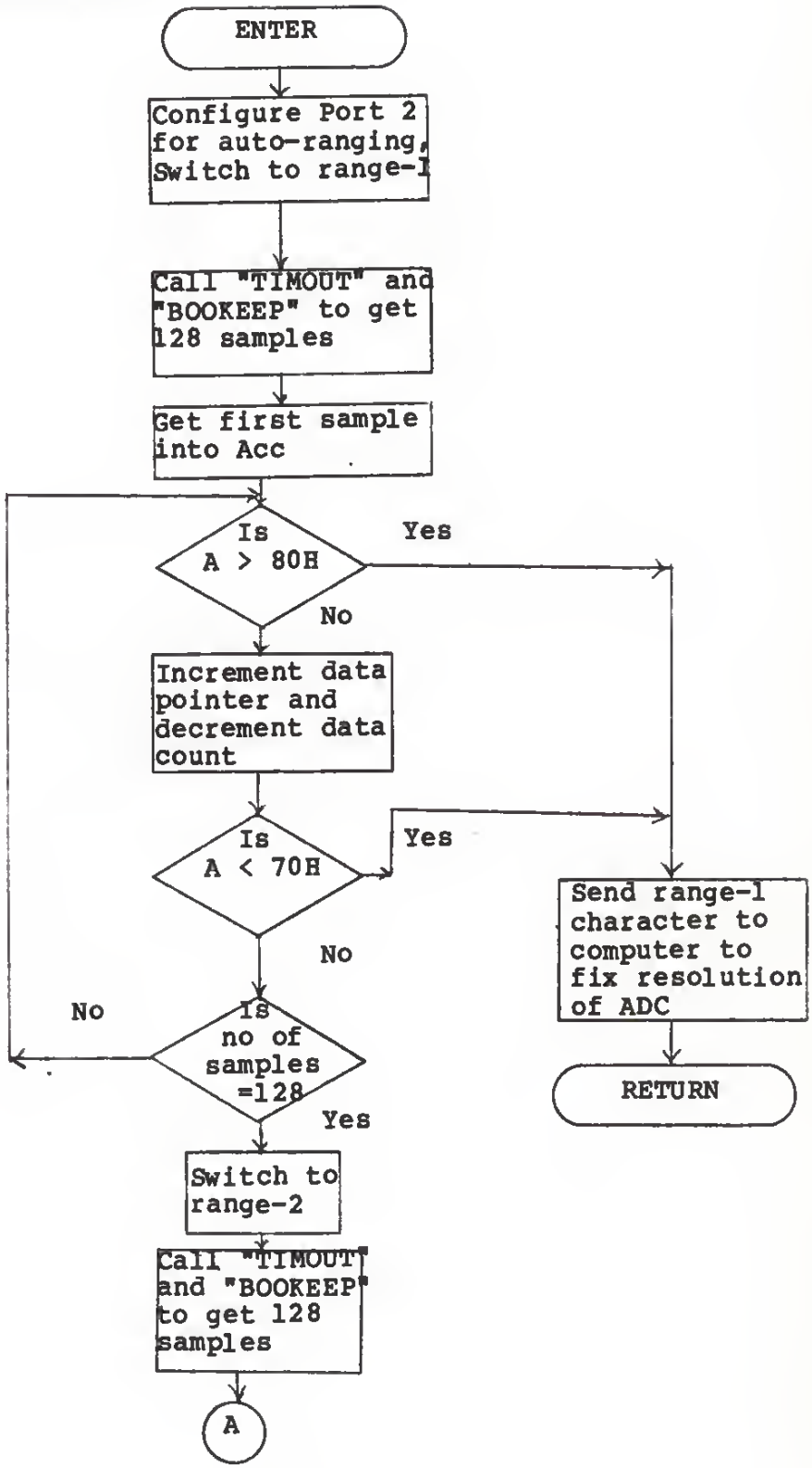
SUB8 ROUTINE FLOWCHART.

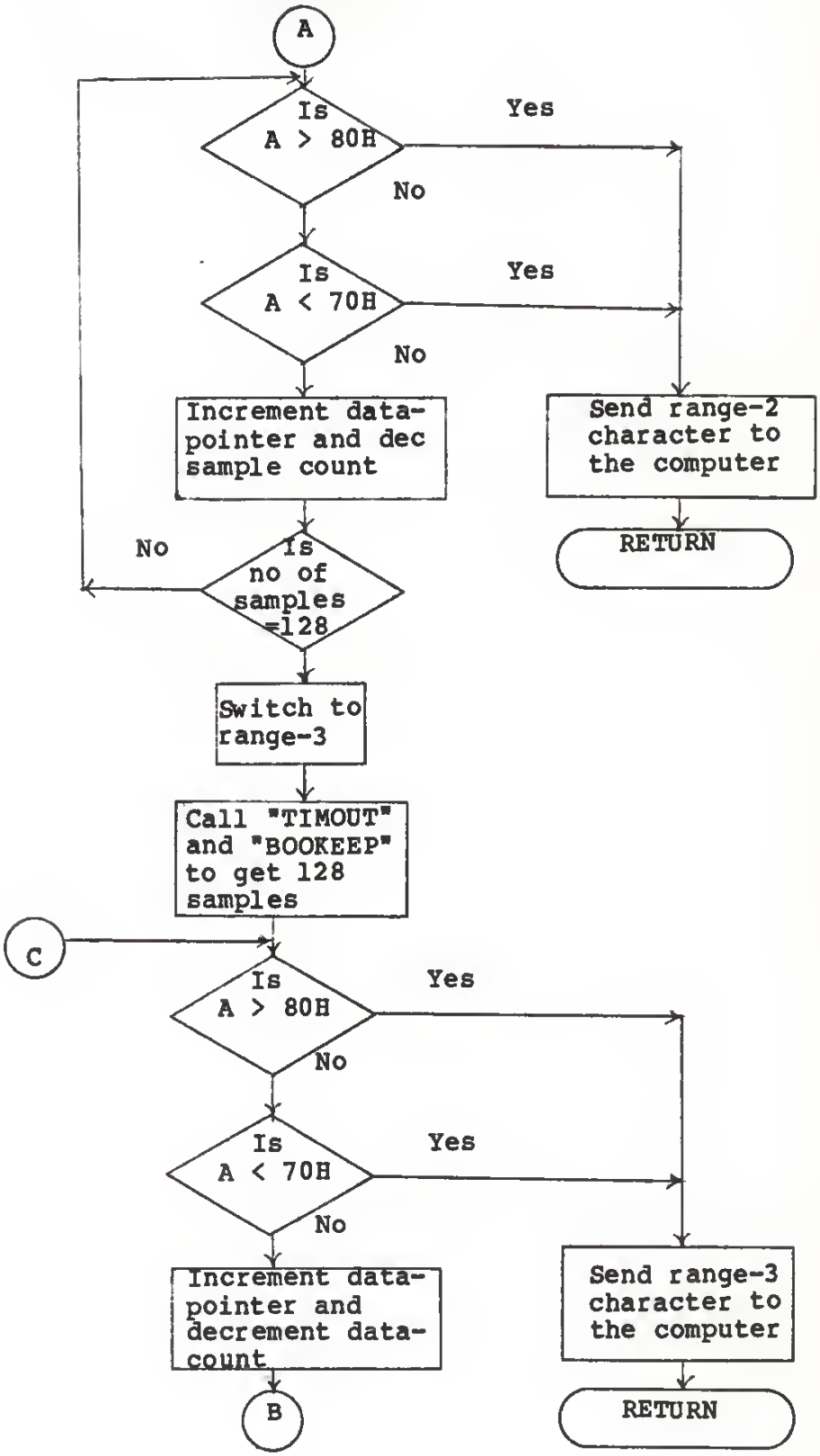


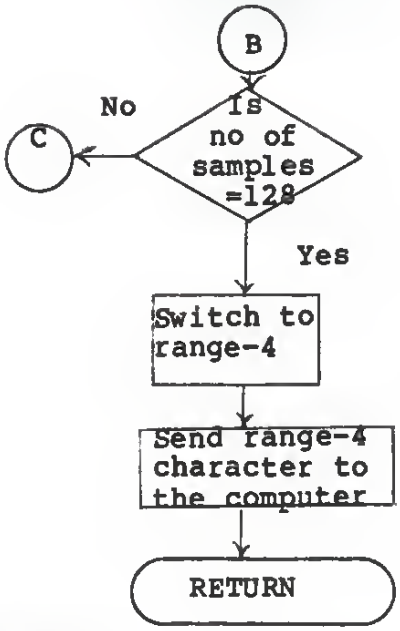
TRANSMIT ROUTINE FLOWCHART.



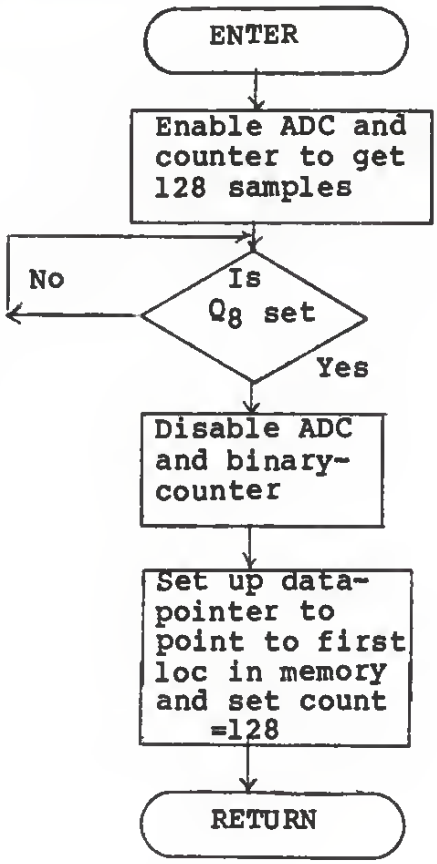
AUTORANG ROUTINE FLOWCHART.




A
Is A > 80H
Yes
No
Is A < 70H
Yes
No
Increment data-pointer and dec sample count
Send range-2 character to the computer
RETURN
No
Is no of samples =128
Switch to range-3
Call "TIMOUT" and "BOOKEEP" to get 128 samples
C
Is A > 80H
Yes
No
Is A < 70H
Yes
No
Increment data-pointer and decrement data-count
Send range-3 character to the computer
RETURN
B



BOOKEEP ROUTINE FLOWCHART.



APPENDIX-G

PROGRAM LISTING IN MNEMONIC CODE FOR THE 8751 MONITOR

```
*****************************************************************
                    NAM       INSTMOD
     BUF            EQU       $70
     NEG            EQU       $71
     LSD2           EQU       $72
     LSD1           EQU       $73
     LSD0           EQU       $74
     MIN            EQU       $75
     VALUE          EQU       R1
     CC             EQU       R0
     TEMVAL         EQU       R2
     CNT            EQU       R3
     TEMP1          EQU       R4
     TEMP2          EQU       R5
     P1.6           BIT       LOCDT
     P1.7           BIT       CNTDT
     P2.4           BIT       CNT128

*****************************************************************
*MAIN PROGRAM                                                   *
*****************************************************************

                    ORG       $0000
     SPINIT         MOV       SCON,#01010001B ;CONFIGURE SERIAL
                    MOV       TCON,#10000000B ;PORT IN MODE 1IN
                    MOV       TH1,#$FD ; AUTO-RELOAD MODE WITH
                    SETB      TR1       ; TIMER1 AS BAUD GEN
     AGAIN          LCALL     GETCHAR
                    CJNE      A,#$41,AGAIN
                    LCALL     PRPLL
                    LCALL     AUTORANG
                    ORL       P1,#00001100B ; ENABLE ADC,COUNTER
                    LCALL     COSEND
                    SJMP      AGAIN

*****************************************************************
*GETCHAR - THIS ROUTINE RECEIVES A CHARACTER FROM THE COMPUTER *
* THRU SERIAL BUFFER REG TO INITIATE THE SAMPLING PROCESS      *
*****************************************************************

     GETCHAR        JNB       R1,$
                    CLR       R1
                    MOV       A,SBUF
                    RET
```

```
**********************************************************************
*READ - THIS ROUTINE READS ASCII CHAR RECEIVED BY THE GETCHAR  *
*ROUTINE CONVERTS IT TO BINARY CODE FOR FREQUENCY RANGE CONTROL*
**********************************************************************

READ        CLR     A
            MOV     CC,A
            MOV     VALUE,A
            MOV     TEMVAL,A
            MOV     CNT,A
NEXT        LCALL   GETCHAR
            CJNE    A,#$24,CHK
            SJMP    CHECK
CHK         SUB B   A,#$30
            PUSH    A
            INC     CC
            SJMP    NEXT
CHECK       MOV     A,CC
            JNZ     A,CNTU
            SJMP    XREAD
CNTU        POP     A
            CJNE    CNT,#$00,POS1
            MOV     VALUE,A
            INC     CNT
            MOV     A,CC
            CJNE    A,CNT,CNTU
            SJMP    XREAD
POS1        CLR     C
            RCL     A
            MOV     TEMVAL,A
            CLR     C
            RCL     A
            CLR     C
            RCL     A
            CLR     C
            ADDC    A,TEMVAL
            ADDC    VALUE,A
            MOV     VALUE,A
            DEC     CNT
            MOV     A,CNT
            CJNE    A,CC,POS1
XREAD       RET
```

```
*********************************************************************
*CONVERT-THIS ROUTINE CONVERTS BINARY DATA TO BCD AND THEN TO  *
*ASCII BEFORE SENDING IT TO THE COMPUTER                       *
*********************************************************************

CONVERT     CLR     A
            MOV     LSD0,A
            MOV     LSD1,A
            MOV     LSD2,A
            MOV     MIN,#$64
BACK1       LCALL   SUB8
            CLR     A
            CJNE    A,NEG,POS2
            INC     LSD2
            SJMP    BACK1
POS2        MOV     A,#$64
            ADD     A,BUF
            MOV     BUF,A
            MOV     MIN,#$0A
            LCALL   SUB8
            CLR     A
            CJNE    A,NEG,POS3
            INC     LSD1
            SJMP    BACK2
POS3        MOV     A,#$0A
            ADD     A,BUF
            MOV     LSD0,A
            MOV     R6,#$30
            ADD     A,R6
            MOV     LSD0,A
            MOV     A,LSD1
            ADD     A,R6
            MOV     LSD1,A
            MOV     A,LSD2
            ADD     A,R6
            MOV     LSD2,A
            RET

*********************************************************************
*SUB8- 8 BIT SUBTRACTION ROUTINE TO CONVERT BINARY TO BCD DIGIT*
*********************************************************************

SUB8        MOV     A,MIN
            PUSH    A
            MOV     NEG,#$00
            CPL     A
            INC     A
            ADDC    A,BUF
            MOV     BUF,A
```

```
                JNC     NEW
                MOV     NEG,#$FF
                CLR     C
NEW             POP     A
                RET

*****************************************************************
*SENDCR- THIS ROUTINE SENDS CARRIAGE RETURN CHAR TO COMPUTER    *
*AFTER  EVERY DATA SENT TO INDICATE END OF DATA                 *
*****************************************************************

SENDCR          JNB     T1,$
                CLR     T1
                MOV     A,#$0D
                MOV     SBUF,A
                RET

*****************************************************************
*TRANSMIT-ROUTINE TO SEND THE ASCII DATA TO COMPUTER            *
*****************************************************************

TRANSMIT        PUSH    DPH
                PUSH    DPL
                MOV     DPTR,#$LSD2
                MOV     A,@DPTR
                MOV     R7,#$03
BACK4           CJNE    A,#$30,TXEN
                SJMP    ZERO
TXEN            JNB     T1,$
                CLR     T1
                MOV     SBUF,A
                INC     DPTR
                DJNE    R7,PATCH
                SJMP    XTRANS
ZERO            CJNE    R7,#$01,NOT
                SJMP    TXEN
NOT             DEC     R7
                INC     DPTR
                SJMP    BACK4
PATCH           MOV     A,@DPTR
                SJMP    TXEN
XTRANS          POP     DPL
                POP     DPH
                RET
```

```
*********************************************************************
*COSEND- CALLS HOUSEKEEPING ROUTINES LIKE TRIST,CONVERT,            *
*TRANSMIT TO SEND THE STORED DATA TO COMPUTER                       *
*********************************************************************

     COSEND          JNB       CNTDT,COSEND
                     LCALL     TRIST
                     MOV       DPTR,#$2000
                     MOV       R1,#$20
     AGA2            MOV       R2,#$40
     AGA1            CLR       A
                     MOVX      A,@DPTR
                     MOV       BUF,A
                     LCALL     CONVERT
                     LCALL     TRANSMIT
                     LCALL     SENDCR
                     INC       DPTR
                     DJNZ      R2,AGA1
                     DJNZ      R1,AGA2
                     RET

*********************************************************************
*SETUP- ROUTINE TO CONFIGURE PORT 0 AND 1 AS OUTPUTS                *
*EXCEPT BITS 6&7 OF PORT 1,TRISTATE THE TRANSCEIVER & I/O PORT      *
* AND ENABLE LATCH OF TRI-STATE BUFFERS TO PROGRAM THE PLL AND      *
* CONTROL DMA                                                       *
*********************************************************************

     SETUP           CLR       A
                     ORL       A,#11000000B
                     MOV       P1,A
                     ORL       P1,#00000101B
                     ANL       A,#$00
                     MOV       P0,A
                     RET
*********************************************************************
*TRIST- THIS ROUTINE DISABLES THE LATCH ENABLE OF BUFFERS TO        *
*STOP DMA AND RE-CONFIGURE PORT 0,2 AND 3 FOR NORMAL OPERATION     *
*SERIAL RECEPTION IS ALSO ENABLED                                  *
*********************************************************************

     TRIST           MOV       P1,#00000011B
                     ORL       P0,#$FF
                     ORL       P3,#$FF
                     ORL       P2,#$FF
                     MOV       SCON,#01010001B
                     RET
```

```
*********************************************************************
*PRPLL-ROUTINE TO PROGRAM THE PHASE LOCKED LOOP THROUGH PORTS 0*
*AND 3 TO ACHIEVE THE SAMPLING FREQUENCY CLOCK AS REQUIRED BY  *
*THE INPUT STATEMENT OF BASIC PROGRAM                          *
*********************************************************************

     PRPLL          LCALL     SETUP
                    LCALL     READ
                    MOV       TEMP1,R1
                    ORL       A,#00000011B
                    LCALL     READ
                    MOV       TEMP2,R1
                    CLR       A
                    CJNE      A,R0,NEXTS
                    MOV       SCON,#$00
                    ORL       P1,#00100101B
                    SJMP      CHKLOOP
     NEXTS          LCALL     SWCAP
                    LCALL     CTRLTBL
     CHKLOOP        JNB       LOCDT,CHKLOOP
                    RET

*********************************************************************
*SWCAP-ROUTINE TO SWITCH CAPACITOR FOR REQUIRED FREQUENCY AND  *
*ALSO FETCH DATA POINTER FOR THE REQUIRED CONTROL WORDS  TO    *
*TO PROGRAM THE PLL AND DIVIDER                                *
*********************************************************************

     SWCAP          PUSH      DPH
                    PUSH      DPL
                    MOV       R6,TEMP1
                    CJNE      R6,#$01,RANG2
                    ORL       P1,#00100101B
                    LCALL     TIMOUT
                    MOV       DPTR,#$0500
                    SJMP      XCAP
     RANG2          CJNE      R6,#$02,RANG3
                    ORL       P1,#00110101B
                    LCALL     TIMOUT
                    ANL       P1,#00010101B
                    MOV       DPTR,#$0600
                    SJMP      XCAP
     RANG3          ORL       P1,#00011101B
                    LCALL     TIMOUT
                    ANL       P1,#00001101B
                    MOV       DPTR,#$0700
     XCAP           RET
```

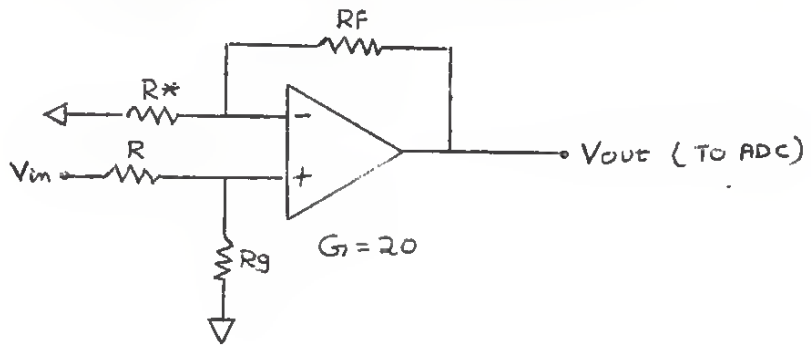
```
*********************************************************************
*CTRLTBL- ROUTINE TO OUTPUT CONTROL WORDS TO PLL AND DIVIDER        *
*FOR LOCKING THE PLL IN SAMPLING FREQUENCY RANGE                    *
*********************************************************************

     CTRLTBL        MOV       A,TEMP2
                    MOVC      A,@A+DPTR
                    MOV       P0,A
                    MOV       A,TEMP2
                    ADD       A,#$80
                    MOVC      A,@A+DPTR
                    POP       DPL
                    POP       DPH
                    RET

*********************************************************************
*AUTORANG-ROUTINE TO SWITCH THE INPUT RANGE OF SIGNAL SO THAT       *
*A GOOD DIGITAL REPRESENTATION OF THE SIGNAL IS OBTAINED IN         *
* THAT RANGE-MAKE BEFORE BREAK SWITCHING OPERATION                  *
*********************************************************************

     AUTORANG       CLR       A
                    MOV       R1,A
                    ORL       A,#11110000B
                    MOV       P2,A
                    ANL       P2,#00000001B
                    LCALL     TIMOUT
                    LCALL     BOOKEEP
     AG1            MOVX      A,@DPTR
                    CLR       C
                    PUSH      A
                    SUB B     A,#$80
                    JNC       XAUTO1
                    POP       A
                    CLR       C
                    SUB B     A,#$70
                    JNC       AG2
                    SJMP      XAUTO1
                    INC       DPTR
                    DJNZ      R1,AG1
     AG2            ANL       P2,#00000011B
                    LCALL     TIMOUT
                    ANL       P2,#00000010B
                    LCALL     BOOKEEP
     AG3            MOVX      A,@DPTR
                    CLR       C
                    PUSH      A
                    SUB B     A,#$80
                    JNC       XAUTO2
                    POP       A
                    CLR       C
                    SUB B     A,#$70
                    JNC       AG4
```

```
                SJMP    XAUTO2
AG4             INC     DPTR
                DJNZ    R1,AG3
                ANL     P2,#00000110B
                LCALL   TIMOUT
                ANL     P2,#00000100B
                LCALL   BOOKEEP
AG5             MOVX    A,@DPTR
                CLR     C
                PUSH    A
                SUB B   A,#$80
                JNC     XAUTO3
                POP     A
                CLR     C
                SUB B   A,#$70
                JNC     AG6
                SJMP    XAUTO3
AG6             INC     DPTR
                DJNZ    R1,AG5
                ANL     P2,#00001100B
                LCALL   TIMOUT
                ANL     P2,#00001000B
                MOV     R1,#$34
                LCALL   TRANS
                SJMP    EXIT
XAUTO1          MOV     R1,#$31
                LCALL   TRANS
                SJMP    EXIT
XAUTO2          MOV     R1,#$32
                LCALL   TRANS
                SJMP    EXIT
XAUTO3          MOV     R1,#$33
                LCALL   TRANS
EXIT            RET

*****************************************************************
*TIMOUT- A DELAY ROUTINE TO STABILIZE THE INPUT AFTER SWITCHING*
*THE RESISTANCE THROUGH PORT 2 BITS P2.0-P2.3                   *
*****************************************************************

TIMOUT          MOV     R7,#$FF
WASTE           DEC     R7
                CJNE    R7,#$00,WASTE
                RET
*****************************************************************
*BOOKEEP-128 SAMPLES OF SIGNAL ARE STORED TO CHECK INPUT RANGE *
*****************************************************************

BOOKEEP         ORL     P1,#00001100B ;ENABLE ADC,COUNTER
RANG            JNB     CNT128,RANG
                ORL     P1,#00000011B
                MOV     DPTR,#$2000
                MOV     R1,#$FF
```

```
                    RET
*********************************************************************
*TRANS- A ROUTINE TO TRANSMIT CHARACTER CORRESPONDING TO THE    *
*SWITCHED RESISTOR FOR FIXING THE RESOLUTION OF FLASH ADC IN    *
*THE BASIC PROGRAM                                              *
*********************************************************************
     TRANS          ANL       A,#$00
                    MOV       SCON,#01010001B
                    MOV       P3,A
                    MOV       A,R1
                    JNB       T1,$
                    CLR       T1
                    MOV       SBUF,A
                    LCALL     SENDCR
                    MOV       SCON,#$00
                    RET
*********************************************************************
```

APPENDIX-H

CALCULATION OF SIGNAL CONDITIONING CIRCUITRY COMPONENTS FOR AUTO-RANGING THE FLASH ADC



$$V' = \frac{Vin\ Rg}{R + Rg} \quad , \quad Vout = V' \left(\frac{Rf}{R^*} + 1 \right)$$

$$Vout = \frac{Vin\ Rg}{R + Rg} \left(\frac{Rf}{R^*} + 1 \right)$$

Case-1.

Vin ⟶ Vout

320 mV ⟶ 6.4V

$$\text{Choose } \frac{Rf}{R^*} + 1 = 21$$

$\frac{Rf}{R^*} = 20$ If Rf = 62K and R* = 3K

then $\frac{Rf}{R^*} = 20.67$

$$V' = \frac{Vout}{20.67} = \frac{6.4}{20.67} = .3096V$$

$$0.3096 = \frac{0.32\ Rg}{R + Rg}$$

For R = 100K (2W)
Rg= 2.984 M Ohm

If Rg = 2.8 M Ohm

$$V' = \frac{0.3 * 2.8}{0.1 + 2.8} = 0.3089 \text{ V}$$

Vout = 0.3089 * 20.67 = 6.38 V (max)

For ±320 mV range Rg = R_1 = 2.8 M Ohm.

Case-2.

Vin ⟶ Vout

3.2V ⟶ 6.4 V V' = 0.3096 V

$$0.3096 = \frac{3.2 * Rg}{100 + Rg}$$

Rg = 10.7 K

Let Rg = 10.5 K

$$V' = \frac{3.2 * 10.5}{100 + 10.5} = 0.304 \text{ V}$$

Vout = 0.304 * 20.67 = 6.285 V (max)

For ±3.2 V range Rg = R_2 = 10.5 K

Case-3.

Vin ⟶ Vout V' = 0.3096 V

6.4 V ⟶ 6.4 V

$$0.3096 = \frac{6.4 * Rg}{100 + Rg}$$

6.1 Rg = 30.96 K Rg = 5.075 K

Let Rg = 5 K then $V' = \frac{6.4 * 5}{100 + 5} = 0.3047$ V

Vout = 20.67 * 0.3047 = 6.3 V (max)

For $\pm$6.4 V range Rg = R_3 = 5 K Ohm

Case-4

Vin $\longrightarrow$ Vout

9.6 V $\longrightarrow$ 6.4 V $\qquad V' = 0.3096$ V

$$0.3096 = \frac{9.6 * Rg}{100 + Rg}$$

Rg = 3.33 K

Let Rg = 3.2 K

then $V' = \frac{9.6 * 3.2}{100 + 3.2} = 0.2976$ V

Vout = 20.67 * 0.2976 = 6.15 V (max)

For $\pm$9.6 V range Rg = R_4 = 3.2 K

Maximum Voltage range	Resistor Values
±320 mV	2.8 M
± 3.2 V	10.5 K
± 6.4 V	5.0 K
± 9.6 V	3.2 K

APPENDIX-I

COMPONENT LIST FOR THE 8751 BASED SYSTEM

NO OF PINS	PART NO	COMPONENT NAME	QUANTITY
40	U1	INTEL-8751 Microcomputer	1
24	U2	INTEL 8212 Input/Output Port	1
20	U3	FAIRCHILD 74F588 Octal - Bi-directional transceiver	1
24	U4	HITACHI HM 6116P-2 (2K*8) Static RAM	1
24	U5	RCA CA-3308, 8 Bit Flash A/D	1
20	U6,U14, U15,U24	FAIRCHILD 74F244 Octal- Tri-State Buffer	4
20	U25,U26	FAIRCHILD 74F373 Tri-state D-Type Latch	2
14	U7	FAIRCHILD 74F10 3 Input Nand Gate	1
14	U8	FAIRCHILD 74F08 Hex Inverter	1
8	U9,U10	MICRO POWER-SYSTEMS Op-Amps OP-227	1
14	U11	FAIRCHILD 74F32 Quad 2 Input OR-Gate	1
14	U12,U13	NATIONAL 74HC024 7 Stage Binary Counter	2
14	U16	MOTOROLA MC 1489 Quad-line receiver	1
14	U17	MOTOROLA MC 1488 Quad-line driver	1
16	U18	MOTOROLA MC 145143 PLL Frequency synthesizer	1
16	U19	SIGNETICS NE 564 Phase Locked Loop or MC 4024 VCO	1
14	U20,U27	NATIONAL CD4066 Quad Bilateral CMOS Switch	2
16	U21,U22	FAIRCHILD 74F161 Sychnronous Presettable counter	2
14	U23	FAIRCHILD 74F00 Quad 2 Input Nand Gate	1
20	J1	RS-232C Interface Connector	1
		8 MHZ XTAL Oscillator	2

IC SOCKETS

PINS	QUANTITY
40	1
24	3
20	7
16	4
8	2
14	10

RESISTORS

$R_1=R_2=R_3=R_4$ = 100 Ohm (1/8 Watt)

R_5=1 M Ohm , R^*=3 K, R_F=62 K

R_G=2.8 M Ohm , 10.7 K , 5K, 3.2K.

CAPACITORS

C_1=10 PF - 2 NOS

C_2=0.22 uF, C_3=0.2 uF, C_4=0.125 uF

C_5=0.0125 uF, C_6=1250 pF.

BIBLIOGRAPHY

[1]. NSC 800 Microprocessor Family Handbook, National Semiconductor, 1981.

[2]. Dwight.W.Gordon, An NSC 800 development system, A Master's Report, 1981.

[3]. Charles.K.Kao, Optical fiber Systems Technology, design and applications, McGraw Hill 1982.

[4]. Hewlett Packard, Snap in Fiber Optic Link Technical Data, July 1982.

[5]. Chris Duffey & Vince Deems, Optical fiber Communication, Microcomputer System Design Project, 1983.

[6]. J.D.Greenfield & Wray, Using Microprocessors and Microcomputers; The 6800 family, John Wiley, 1981.

[7]. Microcontroller Handbook, Intel Corporation, 1983.

[8]. Ulrich.L.Rohde, Digital PLL frequency synthesizer, Theory and design, Prentice Hall Inc, Englewood Cliffs, NJ 07632.

[9]. John. Allocca & Allen Stuart, Electronic Instrumentation, Reston publishing company Inc, 1983.

[10]. N.L.Fernandes, An Automated laboratory test system, A Master's Report, 1983.

[11]. M.Duggirala, Applications of Signal Processing techniques on Microprocessor based Instrumentation, A Master's Thesis, 1984.

ACKNOWLEDGEMENTS

I take this opportunity to sincerly thank my major advisor Dr. M.S.P. Lucas for his excellent guidance, helpful suggestions and encouragement during my graduate program.

Thanks are due to my committee members, Dr.Hummels and Dr.G. Hadjipanayis for their suggestions and support. I also express my appreciation to Mr. Tim Diekker and Mr.Hank Lanford for preparing the printed circuit layouts at such short notice. Special thanks to Mr. Phil Buckland for initiating me on the spinwriter, which made the typing of this report a pleasure.

Gratitude is due to my parents whose encouragement contributed towards the sucessful completion of my graduate program.

DESIGN OF A MICROPROCESSOR BASED INSTRUMENTATION MODULE FOR SIGNAL PROCESSING APPLICATIONS

by

ASHOK RAMACHANDRAN

B.S., College of Engineering, Guindy, Madras, India, 1981

AN ABSTRACT OF A MASTER'S THESIS

Submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1984

ABSTRACT

Studies were made of two different designs of a Microprocesor based test-module and comparisions were made between them. With the objective of improving the existing Automated-test system several new features have been suggested for noise-free measurements. The test-module which would act as an intelligent terminal to the host-computer would lead to accurate and faster measurements. The first design was made for a constant sampling frequency of 100 KHz using the NSC 800 Micro-processor for low-power applications. An Auto-ranging test module for variable-sampling frequency up to 8 MHZ was made using Intel-8751. Optical fiber-communication is suggested instead of the RS-232C serial communication. The hardware and software design for the test-module using the 8751 are made for direct implementation.

Phase measurement, emulation of digital oscilloscope are some of the possible applications of this test module. The development aspects of a new Instrumentation test system consisting of several such test modules are also discussed.