MOMENT ANALYZER:

MICROPROCESSOR BASED

by

D. S. Ho, B.A.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree of

Master of Engineering

McMaster University

May 1978

MOMENT ANALYZER:

MICROPROCESSOR BASED

MASTER OF ENGINEERING (1978)
(Electrical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Moment Analyzer: Microprocessor Based

AUTHOR: David Siu-Hang Ho

SUPERVISOR: Dr. R. Kitai, D.Sc.

NUMBER OF PAGES: viii, 101

ABSTRACT

A microprocessor based moment analyzer for the measurement of the first four statistical moments of a random or periodic signal has been developed and implemented. The processing circuitry uses the current state-of-the-art semiconductor components to produce an instrument that is cheaper and more compact than all previous implementations. It also provides more flexibility and demonstrates the current design philosophy of employing programmable LSIs as major building blocks. The instrument consists of two custom designed boards and utilizes the microprocessor as well as other facilities in an existing INTEL Prompt 80 microcomputer development system.

Acknowledgement

The author is greatly indebted to the supervisor, Dr. R. Kitai, for his constant encouragement and intuitive discussions during the course of the research. He also wishes to thank other faculty members of the Electrical Engineering Department of McMaster University and his research colleagues for making his stay in the university a very pleasant one.

The author is also very grateful for the invaluable time and effort that Miss Mary Yeh has devoted in typing and correcting this thesis. This research is supported by a grant from the National Research Council of Canada.

CONTENTS

		Page
Chapter 1	Introduction	1
Chapter 2	Algorithms for Statistical Moments	6
	2.1 Introduction	6
	2.2 Direct Accumulation	7
	2.3 Weighted Feed Algorithm	9
	2.4 Choice of Algorithm	14
Chapter 3	Adaptation to a Microcomputer System	16
	3.1 Introduction	16
	3.2 INTEL SBC 80/10 Computer	17
	3.3 INTEL Prompt 80	24
	3.3.1 General Description	24
	3.3.2 Display	26
	3.3.3 Keyboard	27
	3.3.4 Interrupts	28
	3.4 Hardware Requirements	30
	3.5 Software Requirements	35
Chapter 4	Hardware Design	39
	4.1 Introduction	39
	4.2 Sampling Circuit	40
	4.3 Timing and Control Circuit	43
	4.3.1 System Clock	43
	4.3.2 Process Sequencer	45
	4.3.3. Control PPI	53
	4.3.4 Counters and Start/Stop Logics	54
	4.4 Moment Generator	58
	4.5 Multiplexing Accumulator	63
	4.6 Microprocessor Bus Interfaces	66

Page

Chapter 5 Software Design 70

5.1 Introduction 70
5.2 Data Structure 70
5.3 Initialization Phase 75
5.4 Control Parameter Acquisition Phase 76
5.5 Measurement Phase 83
5.6 Result Processing and Display Phase 85

Chapter 6 Conclusions 92

6.1 Accuracy of the Instrument 92
6.2 Suggested Further Development 96

FIGURE CAPTIONS

Figure		Page
3.1	SBC 80/10 Block Diagram	18
3.6	System Block Diagram	34
3.7	Operating System Flowchart	38
4.1	Sampling Circuit	41
4.2	ADC Timing Diagram	44
4.3	Timing and Control Circuit Block Diagram	44
4.4	System Clock and Timing Diagram	46
4.5	Process Sequencer and Timing Diagram	48
4.6	Control Signals Timing Diagram	50
4.7	Sample Ready Latch and Multiplexing Accumulator Clearing Circuit	52
4.8	Control PPI Circuit	52
4.9	Counter and Start/Stop Logics	56
4.10	Moment Generator	60
4.11	Truncation Circuit	62
4.12	Multiplexing Accumulator	64
4.13	Microprocessor Bus Interface	68
5.2	Initilization Phase Flowchart	77
5.3	Command Loop Flowchart	79-80
5.4	Data Entry Loop Flowchart	81
5.6	Partial Accumulation Pick-up Loop Flowchart	86
5.7	Interrupt Service Procedures Flowchart	87

Figure		Page
5.8	Result Processing Flowchart	90
5.9	Display Functions Flowchart	91
6.4	Improved Moment Generator	97

LIST OF TABLES

Table		Page
3.2	Prompt 80 I/O Addresses	21
3.3	SBC 80/10 External Bus	22-23
3.4	Prompt 80 Memory Map	25
3.5	Prompt 80 Key Codes	29
5.1	Register Map of Instrument	72
5.5	Key Definitions for Instrument	82
6.1	DC Test Results	93
6.2	Sine Wave Test Results	94
6.3	Triangular Wave Test Results	94

Chapter 1

Introduction

In the past years, various techniques have been developed for the real time measurement of statistical moments associated with random or periodic signals. These experimental procedures are helpful in the characterizing of a signal since its statistical properties can be closely approximated once all the moments are determined. Thus a system can be identified by analyzing its outputs, from which the control parameters can also be derived. Earlier works in this area were concentrated around the application of digital measuring methods in most cases of which special purpose digital computers were involved. With the aid of fast and accurate analog to digital converters as a mean of quantization, these instruments have achieved better than 1% error in the measurement of all four moments.

A powerful Weighted Feed algorithm, which completely eliminated the requirement of multiplication in the generation of higher order moments from a sample, was developed by Deist and Kitai[1] in 1963. Later implementations by others[2, 3, 4] were based on this algorithm and extensive investigations of the error characteristics associated with such a process had been carried out. In 1971, the design and implementation of a special purpose computer for the real time measurement of all four moments was described by Majithia.[4] The instrument built was capable of accommodating input signals of up to 5 kHz and had a maximum sampling rate of 100 kHz. However, a total of more than 400 TTL (Transistor-Transistor Logic) chips were employed, which made it less attractive in many applications due to its size.

Recent advances in the state-of-art semiconductor technology have dramatically changed the design philosophy of digital systems. Especially after the introduction of the first four bit PMOS (P-channel Metal Oxide Semiconductor) microprocessor by INTEL in 1972, the world of digital signal processing has emerged into a new era. Versatile microprocessors are replacing numerous complicated hardwired logics in practically every possible environment where speed is not critical. This has resulted in enormous reduction in costs and great space efficiency in most cases. The power of microprocessors is further enhanced by the large varieties of supporting peripheral logic chips supplied by the manufacturers. These chips, into which thousands of logic gates are integrated, can be programmed to perform dedicated functions that could have required hundreds of discrete logic chips in the past. One area of application of these devices is in digital instrumentation in which the flexibility of a microprocessor is providing a significant degree of freedom in the control of measurements and the analysis of data. Many instruments of such a nature have been produced for various purposes and among them, the moment analyzer is just an example.

This thesis describes the design and implementation of a moment analyzer which resides in an existing microprocessor development system--the INTEL Prompt 80[5]--as a sub-system. A different approach of processing is considered, in which the feasibility of the Weighted Feed algorithm is challenged. Combined with the recent technological advances, it has resulted in an instrument which demonstrates comparable performance with more efficiency. Some salient characteristics of the instrument are as follows.

1. It accommodates input signals of up to 15 kHz.
2. The sampling rate is variable within a range of 3 to 9999 microsecond.
3. The sample size is variable within a range of 1 to 9999.
4. The results are available immediately after a measurement in sign and magnitude form in decimal code.
5. Measurement can be programmed for a fixed sample size or a complete period of the input signal.
6. For a one cycle measurement, the period of the input signal is available as a result.
7. The instrument is controlled solely by key entries through the keyboard and display assembly of the microprocessor development system. There are practically no switches to operate.
8. The results are displayed on the keyboard and display assembly, which also serves to monitor a measurement in progress.

Two algorithms for the computation of the four statistical moments from the digitized samples of a signal are presented in Chapter 2. The Weighted Feed algorithm is compared with the method of direct accumulation based on hardware requirements for the generation of high order moments. It is concluded that the former is less efficient in the sense that an extra bit is required to be handled if both are retaining the same resolution of eight bits in the quantization process. The second algorithm is therefore used in subsequent implementation of the instrument.

To adapt an instrument to a computer system as a sub-system, the key features of the host system have to be fully understood before any actual design can be initiated. Chapter 3 is devoted to this purpose. In this chapter, the structure and operation of the INTEL Prompt 80 microprocessor development system and its central processing element, the INTEL SBC 80/10[6] single board computer, are detailed. The adaptation has resulted in significant saving in software and hardware overhead, which are reflected in the discussion of the hardware and software requirements for configuring the instrument.

Chapter 4 concerns itself with the hardware design involved in the instrument. It follows the step by step processing of a quantized sample produced by the sampling circuit and describes the hardware that is designed for each of the processes. The current design philosophy of using LSI (Large Scale Integration) chips as principal building blocks has been adopted throughout the design process and is able to keep the chip count to about 80, which makes it possible to confine all hardware components to two 6.75 by 12 in. boards.

In Chapter 5, the software that controls the instrument is analyzed by tracing the flow of control during initialization and the course of a measurement. About one and a half kilobytes of machine code program have been written exclusively for running the instrument. This additional software module is designed to fit into the limited memory space in the Prompt 80 that is available to a user. It is inconspicuous to all operations of the Prompt 80 monitor although parts of the resident monitor have been altered to accommodate the extra interrupt originating from the instrument.

Concluding the thesis, Chapter 6 reveals the results of testing the performance of the instrument with AC and DC signals as inputs. From these results, the error of the instrument is estimated. These are followed by suggested further extensions of hardware and software that may serve to improve the instrument.

Chapter 2

Algorithms For Statistical Moments

2.1 Introduction

The use of digital equipment in the processing of analog data requires the analog signal to be sampled at discrete time intervals and digitized into finite-length digital words. Hardware and software algorithms can then be employed in the analysis of the digital data to produce meaningful results. This quantization of the signal governs the accuracy of an instrument as well as its complexity. Increase in the word length improves the accuracy but adds complexity as the number of processing elements is increased. It is therefore essential in the early stages of the development to determine the optimal combination of word length and algorithm from the points of view of efficiency and cost effectiveness.

Two algorithms for the computation of the four statistical moments are presented in this chapter. Each is discussed in detail and also with regard to the hardware generation of the higher order moments from the digitized samples. A comparision is made in the last section and the best algorithm and word length combination is chosen for subsequent development.

2.2 Direct Accumulation

The k^{th} statistical moment m_k of a time varying random signal v(t) for a finite period T is defined by the time average integral as

$$m_k = \frac{1}{T} \int_o^T v^k(t)\ dt \tag{2.1}$$

or by the expectation integral as

$$\begin{aligned} m_k &= E\ \{v^k(t)\} \\ &= \int_{-\infty}^{\infty} v^k(t) f(v)\ dv \end{aligned} \tag{2.2}$$

where f(v) is the probability density function of v(t).

Assume that v(t) is sampled at discrete time intervals of length Δt and quantized into n intervals on either side of zero with

$$n = 2^{\ell} \tag{2.3}$$

During the quantization process v(t) is quantized with reference to a fixed standard V_S to produce the digital word r of length ℓ according to the relation

$$r = \left| \ n \frac{v(t)}{V_s} \ \right| \tag{2.4}$$

If a total of M samples are taken during the period T such that

$$T = M\Delta t \tag{2.5}$$

then the time average integral in equation (2.1) can be written in a discrete form

$$m_k = \frac{1}{T} \sum_{i=1}^{M} v^k(i\Delta t)\ \Delta t$$

$$= \frac{1}{M\Delta t} \sum_{i=1}^{M} \{\mathrm{sign}(v_i)\ \frac{r_i\ V_s}{n}\}^k\ \Delta t$$

$$= \frac{V_s^{\ k}}{Mn^k} \sum_{i=1}^{M} \{\mathrm{sign}(v_i)\ r_i\}^k \qquad (2.6)$$

where v_i is the value of $v(t)$ at time of the i^{th} sample and r_i the corresponding digital word.

From equation (2.6), it is obvious that the k^{th} moment can be directly obtained by accumulating $\{\mathrm{sign}(v_i)\ r_i\}^k$ for every i and normalizing with the value $V_s^{\ k}/(Mn^k)$. The process is straight forward and simple except that it requires the value of $\{\mathrm{sign}(v_i)\ r_i\}^k$ to be available at time of each accumulation. For real time applications it means that r^k has to be generated from r in a period of time less than Δt after the sampling and digitizing have been completed. The normalization at the end of the accumulation can be simplified by suitable choice of V_s and n.

Multiple input-multiple output combinatorial circuit can be used to obtain r^k from r as they both have finite values. This is the fastest method but involves large amounts of hardware and excessive wiring. Simpler circuit designs result if multipliers or read-only-memory (ROM) look-up-table are used.

2.3 Weighted Feed Algorithm

With the same number of quantization levels as described in the previous section, equation (2.2) can also be written in a discrete form

$$m_k = \sum_{r=-n}^{n} v_r^k p(v_r) \tag{2.7}$$

where v_r is the value of $v(t)$ represented by level r and $p(v_r)$ the discrete probability density function of $v(t)$ for level r within period T. If v_r is represented by the mid-level value between the r^{th} and $(r+1)^{th}$ level such that

$$v_r = \frac{2r+1}{2n} V_s \tag{2.8}$$

then equation (2.7) becomes

$$m_k = V_s^k \sum_{r=-n}^{n} \left(\frac{2r+1}{2n}\right)^k p(v_r) \tag{2.9}$$

which can be rewritten as

$$m_k = V_s^k \sum_{r=-n}^{n} \left(\frac{2r+1}{2n}\right)(P_r - P_{r+1}) \tag{2.10}$$

where P_r is probability that the voltage exceeds the r^{th} level. Separating the positive and negative parts of equation (2.10) gives

$$m_{k+} = V_s^k \sum_{r=0^+}^{n} (P_r - P_{r+1})\left(\frac{2r+1}{2n}\right)^k \tag{2.11}$$

$$m_{k-} = V_s^k \sum_{r=0}^{-n} (P_r - P_{r+1})(\frac{2r+1}{2n})^k \quad (2.12)$$

The similarity of the two equations allows the development of the algorithm for one to be applied to both. Expanding m_{k+}, one obtains

$$m_{k+} = V_s^k \{(P_0{+} - P_1)(\frac{1}{2n})^k + (P_1 - P_2)(\frac{3}{2n})^k + \cdots$$

$$\cdots + (P_r - P_{r+1})(\frac{2r+1}{2n})^k + \cdots$$

$$\cdots + (P_{n-1} - P_n)(\frac{2n-1}{2n})^k\} \quad (2.13)$$

For practical pruposes the n^{th} level is never exceeded and the equation reduces to

$$m_{k+} = \frac{V_s^k P_0{+}}{(2n)^k} + \frac{1}{(2n)^k} \sum_{r=1}^{n-1} P_r\{(2r+1)^k - (2r-1)^k\} \quad . \quad (2.14)$$

The first term $(V_s^k P_0{+})/(2n)^k$ in equation (2.14) represents the contribution to m_{k+} due to samples in the first quantization level. It is usually small except for k=1 and small n. In general, however, if n is large enough, this term is negligible even for k=1. This approximation is necessary for further simplification of the equation to obtain the weighting numbers corresponding to the n levels of each moment.

Let C_r denote the number of samples that exceed the r^{th} level such that for total number of positive samples M_+,

$$P_r = \frac{C_r}{M_+} \quad (2.15)$$

Replacing P_r in equation (2.14) yields

$$m_{k+} = \frac{V_s^{\ k}}{M_+(2n)^k} \sum_{r=1}^{n-1} C_r\{(2r+1)^k - (2r-1)^k\} \tag{2.16}$$

The equation for each moment can then be obtained by substituting in various values of k.

For the first moment, k=1

$$m_{1+} = \frac{V_s}{M_+(2n)} \sum_{r=1}^{n-1} 2C_r$$

$$= \frac{V_s}{M_+n} \sum_{i=1}^{n-1} C_r \tag{2.17}$$

For the second moment, k=2

$$m_{2+} = \frac{V_s^{\ 2}}{M_+(2n)^2} \sum_{r=1}^{n-1} 8C_r \cdot r$$

$$= \frac{V_s^{\ 2}}{M_+n^2} \sum_{r=1}^{n-1} 2r \cdot C_r \tag{2.18}$$

For the third moment, k=3

$$m_{3+} = \frac{V_s^{\ 3}}{M_+(2n)^3} \sum_{r=1}^{n-1} C_r\{24r^2 + 2\}$$

$$= \frac{V_s^{\ 3}}{M_+n^3} \sum_{r=1}^{n-1} C_r\{3r^2 + \frac{1}{4}\}$$

$$\doteq \frac{V_s^{\ 3}}{M_+n^3} \sum_{r=1}^{n-1} 3r^2C_r \tag{2.19}$$

for $3r^2 \gg \frac{1}{4}$.

For the fourth moment, k=4,

$$m_{4+} = \frac{V_s^4}{M_+(2n)^4} \sum_{r=1}^{n-1} C_r\{64r^3 + 16r\}$$

$$= \frac{V_s^4}{M_+n^4} \sum_{r=1}^{n-1} C_r\ \{4r^3 + r\}$$

$$\doteq \frac{V_s^4}{M_+n^4} \sum_{r=1}^{n-1} 4r^3C_r \qquad (2.20)$$

for $4r^3 \gg r$.

In the last two equations, it is assumed that the full scale of the sampling and digitizing circuit is always utilized, so that the approximations hold. From these equations, a general form for m_{k+} can be derived

$$m_{k+} = \frac{V_s^k}{M_+n^k} \sum_{r=1}^{n-1} kr^{k-1}C_r \qquad (2.21)$$

This expression leads to the direct calculation of the weighting numbers.

Consider if a sample occurs that exceeds level r, then all levels from 0 to (r-1) are also exceeded. The total contribution to m_{k+} according to equation (2.21) is

$$W_{k,r} = \sum_{q=1}^{r} kq^{k-1} \qquad (2.22)$$

$W_{k,r}$ is termed the weighting factor for the r^{th} level of the k^{th} moment. This allows real time contribution of the sample to m_{k+} to be accumulated. If $W_{k,r}$ can be generated from r, equation (2.21) then becomes

$$m_{k+} = \frac{V_s^{\,k}}{M_+ n^k} \sum_{i=1}^{M+} W_{k,r_i} \tag{2.23}$$

Where W_{k,r_i} is the weighting factor for the i^{th} sample. The negative parts are similarly obtained and subtracted from the positive parts for the odd moments and added to the positive parts for the even moments. Normalization of the final result by $V_s^{\,k}/(Mn^k)$ produces the moments.

The weighting factors deduced from (2.22) for the individual moments are

$$W_{1,r} = r \tag{2.24}$$

$$W_{2,r} = r(r+1) \tag{2.25}$$

$$W_{3,r} = \frac{1}{2}r(r+1)(2r+1) \tag{2.26}$$

$$W_{4,r} = r^2(r+1)^2 \tag{2.27}$$

Using these and equation (2.23) the final form for the moments can be established

$$m_1 = \frac{V_s}{Mn} \sum_{i=1}^{M} \text{sign}(v_i) r_i \tag{2.28}$$

$$m_2 = \frac{V_s^{\,2}}{Mn^2} \sum_{i=1}^{M} \text{sign}(v_i)^2 r_i (r_i+1) \tag{2.29}$$

$$m_3 = \frac{V_s^{\,3}}{Mn^3} \sum_{i=1}^{M} \operatorname{sign}(v_i)^3 \, \frac{1}{2} \, r_i(r_i+1)(2r_i+1) \qquad (2.30)$$

$$m_4 = \frac{V_s^{\,4}}{Mn^4} \sum_{i=1}^{M} \operatorname{sign}(v_i)^4 \, r_i^{\,2}(r_i+1)^2 \qquad (2.31)$$

All three methods discussed in the previous section can also be employed for the generation of W_{k,r_i} from r_i. However, the word length requirement is different. A more detailed review is presented in the next section.

2.4 Choice of Algorithm

For a microcomputer based instrument, the word length usually resides with that of the microprocessor so that the capacity of the system can be utilized to its full extent. As the instrument is intended to be adapted to the Prompt 80 microprocessor development system which is developed around the eight bit SBC 80/10 computer, it is obvious that the word length is to be eight or a multiple of eight.

In this case, a word length of eight is chosen for the reason that word lengths of sixteen and above result in too big a system, the accuracy of which far exceeds that required by an averaging process. Eight bit quantization on both sides of zero gives a total range of more than 500 levels, which leads to a resolution of better than 0.1% of full scale as the uncertainty of the quantization process is in half the least significant bit. This provides enough accuracy for the intended instrument if proper scaling can be done before sampling.

Upon determination of the word length, the feasibilities of the algorithms can be analyzed with regard to their hardware and software requirements. In the case where k=1, equation (2.6) is identical to equation (2.28), which is merely the direct accumulation of the samples. For k>1, both algorithms require the generation of the values to be accumulated from the sample. Contrasting equations (2.29) to (2.31) with equation (2.6) reveals that in each case one more bit has to be handled if weighting numbers are to be used. That means extra processing elements are necessary. It also reflects the adverse effect of the approximations in the development of the weighted feed algorithm as equation (2.6) is obtained with no approximation at all.

The weighted feed algorithm is therefore inferior under these considerations no matter what method of generation and accumulation are to be used. Thus the method of direct accumulation is chosen and this allows the instrument to be configured after features of the host microcomputer system have been discussed in the next Chapter.

Chapter 3

Adaptation to Microcomputer System

3.1 Introduction

The speed of a microprocessor is limited. If it is programmed to produce the contribution r_i^k to each moment and to accumulate them, the amount of time required to process each sample will be excessive. This restricts the input signal bandwidth so that only slow varying signals can be analyzed. To overcome this, high speed hardware has to be employed to extend the upper frequency limit.

Under these considerations, the role of the microprocessor in the system can be very clearly defined. It takes up the processing of data where speed is not critical and provides the suitable man-machine interface by converting keyboard commands into control signals that drive the high speed hardware. The final results of a measurement can also be processed and displayed with the aid of the microprocessor.

It is intended that the instrument is to reside in an existing INTEL Prompt 80[5] development system as one of its sub-systems so that the CPU (Central Processing Unit) and other facilities of the host system can be utilized as parts of the instrument. In this way, the cost of the instrument and the time taken to design it can be significantly reduced. In this chapter, various features of the INTEL SBC 80/10[6] computer and its host system the INTEL Prompt 80 are presented. The major hardware and software building blocks that are required to configure the instrument are then identified and briefly described. These lead to the final detailed designs which are featured in the two subsequent chapters.

3.2 INTEL SBC 80/10 Computer

The SBC 80/10 is a self-contained single board OEM (Original Equipment Manufacturer) computer designed with full utilization of the current state-of-art LSI (Large Scale Integration) technology. It is complete in the sense that the CPU, minimal amount of memories, I/O (Input/Output) ports plus their necessary drivers are integrated into a 6.72 in. by 12 in. printed circuit board to form a stand-alone computer system that can be used in a great variety of environments. A block diagram of the computer is given in figure 3.1.

The CPU consists of three LSI chips, the most important of which is the INTEL 8080[7] microprocessor that is capable of processing eight bits of parallel data. It accepts crytal controlled two phase clock signals from a 8224[7] clock driver for the generation of internal and external timing signals necessary for processing of data and access to memories and I/O ports. Limited by its number of pins, some of the external control signals are sent out as a coded status word on the data bus during the first state of a machine cycle, which a 8238[7] system controller latches and decodes into more usable signals to be supplied to various part of the computer.

There are sixteen address lines for addressing memories, half of which also double as I/O address lines. This allows the CPU to address directly up to 65532 memory locations and 256 I/O ports. One special feature of the 8080 microprocessor is that it has a READY input which

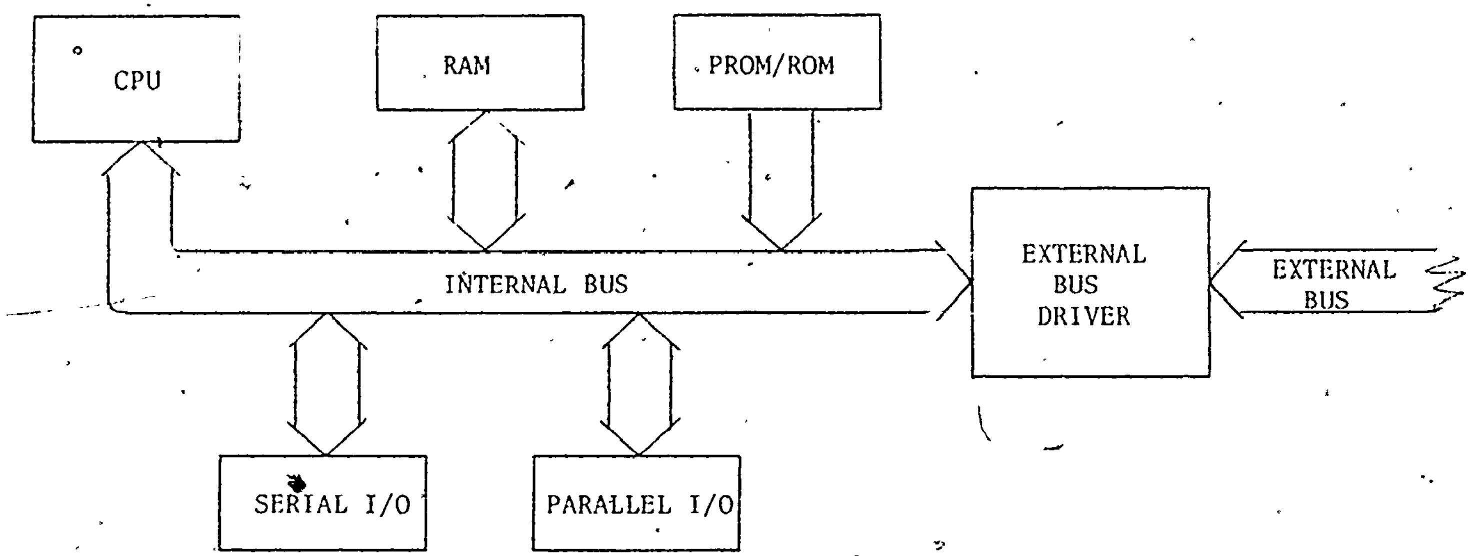

Fig. 3.1 SBC 80/10 BLOCK DIAGRAM

enables the CPU to access devices that are slow in response compared to its cycle time. When a read/write signal is sent to a device after the address is stable on the address bus, it does not proceed any further with the data on the data bus until the READY signal has become high. Wait states are inserted to achieve the necessary synchronization; however, the CPU will enter an infinite wait period if a non-existent device is addressed. In the SBC 80/10, this is remedied by triggering a monostable which automatically supplies a READY signal after 9 milliseconds through the RDYIN input of the 8224. READY signals from on board memories and I/O ports are also ORed to this line and for external modules, the AACK line on the external bus is provided to serve the same purpose.

Two types of memories are available to the user. The volatile RAM (Random Access Memory) consists of eight 256 by 4 bits 8111A[7] static RAM chips with a total of 1024 bytes. Addresses of the RAM are from 3CØØ (hexadecimal) to 3FFF. There are four 24 pin sockets for 4096 bytes of non-volatile ROM (Read Only Memory). Both 8708 EPROMs (Erasable and Electrically Reprogrammable ROM) and 8308[7] masked programmable ROMs with an organization of 1024 by 8 bits can be installed in these sockets. ROM addresses are from ØØØØ to ØFFF.

The computer communicates with the outside through two 8255 PPIs[7] (Programmable Peripheral Interfaces) and a 8251 USART[7] (Universal Synchronous/Asynchronous Receiver/Transmitter). Each PPI provides four parallel I/O ports three of which can be configured to exchange data with external peripherals in a great variety of I/O modes. The fourth port accepts only command words from the CPU for the control of the PPI. Sockets for TTL (Transistor Transistor Logic) buffers are provided for the I/O lines. The 8251 USART is a serial communication link that occupies two I/O addresses. Data can be transmitted and received serially through the 8251 in various software selectable modes and a common 20 mA current loop interface has been prewired to support a teletype. To select a specific mode command words are written into one of the I/O ports and the USART is programmed accordingly. The same port is used for reading in the status while the other is dedicated to data transfer. Baud rate for the USART is obtained by counting down the 18.432 MHz system clock and is jumper selectable. A list of I/O addresses are tabled in table 3.2.

All blocks in the computer are joined by the three internal buses which carry the data, addresses and timing signals. Provision for external expansion is through the external bus driver that buffers between the negative true external bus and the internal bus. Compatible modules can be attached to the external bus and addressed in a manner similar to the on-board blocks. The external bus comes out of the board through a 86 pin edge connector P1. A listing of the pin assignments for P1 is given in table 3.3.

As an OEM board, the SBC 80/10 comes with no system software. However, a two kilobyte system monitor resident on two 8708 EPROMs can be purchased from INTEL to configure a general purpose computer. This allows user programs to be loaded and executed via a serial terminal. Additional features include the examination and alteration of CPU registers and punching and reading of paper tape files.

Table 3.2 PROMPT 80 I/O ADDRESSES

I/O ADDRESS	FUNCTION
E4	PPI1 Port A
E5	PPI1 Port B
E6	PPI1 Port C
E7	PPI1 Control Port
E8	PPI2 Port A
E9	PPI2 Port B
EA	PPI2 Port C
EB	PPI2 Control Port
EC/EE *	USART Data Port
ED/EF *	USART Control Port

* Jumper Selectable

Table 3.3 SBC 80/10 EXTERNAL BUS

PIN	MNEMONIC	DESCRIPTION
1, 2	GND	Signal Ground
3, 4, 5, 6	VCC	+5V DC
7, 8	VDD	+12V DC
9, 10	VBB	-5V DC
11, 12	GND	Signal Ground
13	BCLK/	Bus Clock
14	INIT/	Initialize
15	BPRN	Bus Priority In
16		Not Used
17	BUSY/	Bus Busy
18		Not Used
19	MRDC/	Memory Read Command
20	MWTC/	Memory Write Command
21	IORC/	I/O Read Command
22	IOWC/	I/O Write Command
23	XACK/	External Transfer Acknowledge
24*	DONE	Done Pulse
25	AACK/	Advance Acknowledge
26*	Ø1	Instrument System Clock Ø1
27*	Ø2	Instrument System Clock Ø2
28*	RESET/	Instrument Reset
29*	Q2	Second Bit of Counter in Process Sequencer
30*	CCL/	Multiplexing Accumulator Clear
31	CCLK/	Constant Clock
32*	SS	Sign of Sample
33*	SØ	Digital Sample
34*	S1	Digital Sample
35*	S2	Digital Sample
36*	S3	Digital Sample
37*	S4	Digital Sample
38*	S5	Digital Sample
39*	S6	Digital Sample
40*	S7	Digital Sample
41*	T1/	Multiplexing Accumulator Strobe
42	INTR1/	Interrupt Request

*Used by instrument.

Table 3.3 SBC 80/10 EXTERNAL BUS (Cont'd)

PIN	MNEMONIC	DESCRIPTION
43	ADRE/	
44	ADRF/	
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	Address Bus
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADRØ/	
58	ADR1/	
59		Not Used
60*	04/	Multiplexing Accumulator Strobe
61		Not Used
62*	FLS/	Final Load Strobe
63		Not Used
64*	BLA/	Bus Latch Acknowledge
65		Not Used
66*	SRE	Bus Latch Strobe Enable
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	Data Bus
71	DAT2/	
72	DAT3/	
73	DATØ/	
74	DAT1/	
75, 76	GND	Signal Ground
77, 78	VBB	-10V DC
79, 80	VAA	-12V DC
81, 82, 83, 84	VCC	+5V DC
85, 86	GND	Signal Ground

*Used by instrument.

3.3 INTEL Prompt 80

3.3.1 General Description

The Prompt 80 is a microprocessor development system specially built to support software development of 8080 microprocessor based systems. It provides a user with all the facilities necessary for the running and debugging of machine code programs. All system activities of the Prompt 80 are controlled through a console keyboard with the associated results displayed on a set of 16 seven segment LED (Light Emitting Diode) displays and 20 discrete LED indicators. A total of 916 bytes of RAM are available for user programs and execution can start anywhere within this boundary. A memory map of the system is given in table 3.4.

There are two modes of execution that can be selected by the user. A user program can take complete control of the system or can be single-stepped through each instruction under control of the system monitor. In the latter mode, any of the CPU registers or memory locations can be examined and altered between the steps, thus allowing the program to be debugged. The system is also capable of programming and verifying the contents of a 8708 EPROM through a zero-insertion-force socket on the console. This provides a very convenient way of producing transferrable and alterable software for use in prototype development. In situations where paper tape files are required, a teletype can be directly connected to the system for the reading and punching of paper tapes.

The heart of the Prompt 80 is a SBC 80/10 computer with three 8708[7] EPROMs that contain the system monitor. A custom design display and keyboard assembly driven through the two PPIs on the SBC 80/10 provides the essential man-machine interface. One PPI controls the display

Table 3.4 PROMPT 80 MEMORY MAP

ADDRESS	CONTENTS
Ø - ØAE8	ROM: System monitor
ØAE9 - ØFFF	ROM: User program space
1ØØØ - 3BFF	UNUSED ADDRESS
3CØØ - 3CØ1	RAM: Reserved for monitor
3CØ2 - 3F9Ø	RAM: User program space
3F91 - 3FCF	RAM: Reserved for monitor
3FDØ - 3FDA	RAM: Reserved for monitor for single step and breakpoint status storage
3FEB	RAM: Display address counter
3FEC - 3FED	RAM: Pointer to display buffer pointer
3FEE - 3FFF	RAM: Display buffer

multiplexing and keyboard scanning while the other supports the EPROM programming circuit. The system has made very extensive and efficient use of software in the implementation of its functions. It is therefore feasible to look into the system in more details so as to make the best use of it.

3.3.2 Display

Besides the power on pilot LED, the rest of the LEDs can be devided in two major groups. One group consists of 16 seven segment LEDs and three indicator LEDs that are used to display register and memory data. They are divided into nine subgroups which are driven in a time multiplexed fashion with a nine millisecond cycle. A monostable interrupts the CPU at one millisecond intervals to initiate a service routine which controls the sequencing. This routine takes the character pair in the eighteen byte long display buffer to which the display pointer is pointing and transfers it to the display assembly through PPI1. The display pointer is then incremented and is reset to the bottom of the buffer again when the top is reached. Any hexadecimal data desired to be displayed must be stored in the display buffer in the form of display characters, which are the decoded low true seven segment equivalence of the data. Each bit of a character represents one segment and the most significant bit controls the decimal point on the right of the LED. This conversion is necessary because the display circuit is driven directly without any decoders to reduce hardware costs. A conversion routine has been incorperated into the monitor software for this purpose. Address of the buffer starts from 3FEE which corresponds to the right most seven segment LED to 3FFD which is the left most. The three dicrete LEDs are used for indicating the different register sets displayed have their control word stored in location 3FFE. Contents of location 3FFF are also used with the light control word to drive the display but do not contribute any effect.

User program can make use of the display by writing the desired character into the buffer or by calling the routine DBYT located at Ø3ØC which formats two hexadecimal digits in the A register into display characters and stores them in successive locations in the buffer as pointed by the register pair DE. However, one important point must be observed when manipulating the display; since it is interrupt driven, the user must ensure that the interrupt flag of the CPU and the display control bit (bit 7 of PPI1 port C) are enabled for proper functioning of the display assembly.

The second set of sixteen discrete LEDs are for monitoring the lines of port A and port B of PPI2. Port A is programmed in latched output mode and port B in input mode and these low true LEDs indicate the activities at these two ports. Besides EPROM programming, the three ports of PPI2 also double as user accessible I/O ports that can be accessed via a 50 pin connector on the console. The only restrictions are that port A and lines 7 to 4 of port C can only be configured as output lines and no external connection can be made to the console connector when EPROM programming is in progress.

3.3.3 Keyboard

The console keyboard is the only data entry channel in the Prompt 80 besides an optional paper tape reader that can be connected to the serial interface. There are a total of thirty-six keys on the console, of which only the twenty-four keys that are used for data entry and system function control can be scanned by the system monitor through PPI1. A set of eight keys are for setting the input lines to port B of PPI2 through S-R latches. These lines can be reset by a nineth key. Of the remaining three keys, one is for system reset and the rest are for generating interrupts. One generates the monitor interrupt which causes a branch to the monitor while the other cuases a user interrupt that branches to address 3C∅2, which is the beginning of the user RAM area.

The scanning of the keyboard is done by a software routine KI located at ∅7EE. It identifies a single key depressed in the set of twenty-four keys, debounces it by checking for the same entry after a sixteen millisecond delay and decodes it into a key character in the A register and a key position in the E register. A list of the keys and the corresponding values in the A and E registers are listed in table 3.5.

3.3.4 Interrupts

Most of the system functions in the Prompt 80 are implemented with the aid of interrupts. Besides the two console interrupts and the display interrupt mentioned earlier, there is a fourth interrupt that comes from the single step timing counter. This counter is loaded and enabled by bit 6 of port C of PPI1 every time a single step through a user program is initiated. It counts for fifteen memory read circles before interrupting the CPU, thus providing such accurate timing that the interrupt always occurs within the first cycle of the user instruction. The single step routine is then executed to save all CPU registers and display them.

Each of these interrupts, when it occurs, sets a latch and through arbitration logics, inhibits others from generating further interrupt

Table 3.5 PROMPT 80 KEY CODES

KEY	A CONTENT	E CONTENTS
0	CØ	0
1	F9	1
2	A4	2
3	BØ	3
4	99	4
5	92	5
6	82	6
7	F8	7
8	80	8
9	98	9
A	88	A
B	83	B
C	C6	C
D	A1	D
E	86	E
F	8E	F
Scroll Register Display	Ø8	10
Single Step	Ø3	11
Exam/Mod Register	46	12
Go	21	13
Dis/Mod Memory	Ø6	14
Clear Entry/Previous	Ø3	15
Next	FF	16
Execute/End	7F	17

signals. The CPU is vectored by a RST 7[8] instruction to location ØØ38 where a jump instruction to the central interrupt service routine is stored. This routine examines the latches through port A of PPI1 to determine the source of the interrupt, resets the latch and then branches to the appropriate service routine. The Prompt 80 has no provision for external interrupts except its own, thus restricting all hardware expansion to non-interrupt type unless the central interrupt service routine is altered to accommodate external interrupts.

3.3 Hardware Requirements

The design of a system is critical. It is necessary that the system specifications as a whole are known to the designer and a set of guidelines are established before design procedures can be initiated. Taking into consideration the instrument built by Majithia[4] and the latest advances in circuit elements, it is estimated that the instrument can be upgraded to have a sampling rate of less than 3 microseconds and an input bandwidth of better than 15 KHz. However, allowances are to be provided for further upgrading of the instrument in future.

Cost and space effectiveness are also the main concern of the design. It is intended that the instrument is to reside on one or two circuit boards that can be plugged directly into the bus of the host computer system. Efforts are spent to achieve this although trade offs in speed and bandwidth of the instrument are found inevitable. An optimal configuration which compromises between these constraints is to be designed.

Before committing to detail the circuit design of the instrument, the major building blocks have to be identified. This can easily be done by tracing the flow of data beginning from the input and specifying the processes that take place along the path. Each of the processes may be considered as a block with interfaces to the others. Specifications of the blocks and their interfaces are then defined and realized in the final design.

In the preceding chapter, it has been concluded that the analog signal is to be quantized into eight bits on both sides of zero. The quantized sample is then accompanied by a sign bit which indicates its polarity and these are passed to a moment generator to produce the higher order moments. Quantization is provided by an ADC (Analog to Digital Converter) but as most commercially available fast ADCs are of 8 bit resolution, an absolute value circuit has to be placed before it so that sampling can be done on an unipolar signal. The sign bit is produced along with the sample by a fast zero-crossing comparator.

There are several methods of producing the moments from the sampled data. An ideal case is to have three fast EPROMs of 256 byte capacity to act as look up tables and the sample is used directly to address the EPROMs simultaneously. Masked ROMs can also be used but are not considered because of the high cost of custom programming. However, EPROMs of such size with less than one microsecond access time are not available and the use of larger capacity 8708 EPROMs is not economical and is wasteful of space. Storing all three tables in one EPROM means that three accesses have to be made to fetch the moments and this slows down the process. The multiple supplies required by an EPROM is also another undesirable feature. Combinatorial logic has found to result in too complicated a circuit and the only alternative is to use an eight bit fast multiplier. Fortunately, it is available as a bipolar 40 pin DIP chip that is capable of completing a signed eight by eight bit multiplication in less than

150 nanoseconds,[9] which makes it possible to produce all the moments successively in a period of less than one microsecond. With this high speed, the front end can be upgraded to a one microsecond sampling rate when better ADCs are available in future.

To accumulate the moments, it is obvious that accumulators made up of adders and registers are to be used. If the positive and negative parts of a moment are processed separately, a total of six identical accumulators are required. This is however, not necessary if two's complement arithmetic is employed. The moments can be converted to their signed two's complement before the accumulation and a reduction to four accumulators is achieved. Further reduction can be obtained if one adder is multiplexed between four result registers. Since the moments are not coming simultaneously from the moment generator, there is no delay in the processing of the moments if an accumulation can be completed before the arrival of the next moment. For a sample rate of three microseconds a very safe margin is provided if 500 nanoseconds are allowed for each moment, which is long enough for a 32 bit accumulation using TTL adders.

The length of the accumulator depends on the maximum number of samples desired to be taken. It is possible to keep it to a minimum if the computer can pick up the accumulations and accumulate them in memory registers after a fixed number of samples. Calculations based on the time required by the computer to finish processing one set of accumulations indicate that the suitable number is 128. A 16 bit multiplexed accumulator is just enough in this case to avoid any overflow that may happen. To notify the computer that the partial results are ready, a counter counts the number of samples accumulated and sets a latch after every 128 samples. This is detected by the computer which resets the latch and picks up the partial results. At the end of a measurement, the computer is interrupted to pick up the last part of the results and proceeds with the final processing.

The fastest sampling rate of the system is set at three microseconds. An accurate system clock and a programmable counter are required to generate the triggering pulse for the ADC with variable repetition rate. The INTEL 8253 PIT[10] (Programmable Interval Timer) is suitable for this purpose. It has three 2 MHz 16 bit counters and each can be programmed to work in various modes. In this application, one counter is programmed to act as a rate counter while the rest are spared for counting total number of samples and the 128 samples required to set the sample ready latch. With a crystal controlled 2 MHz system clock, the rate counter can generate repetition rates varying from three microseconds to thirty-two milliseconds.

Besides clocking the rate counter, the system clock also drives a controller which co-ordinates all hardware activities. The controller is designed with high speed TTL and is responsible for the generation of sequences of timing signals that synchronize the flow of data. Supervision from the computer and the controller status are exchanged through two ports of a 8255 PPI. Each port provides one direction of data flow, the third being available for future system expansion.

With all the building blocks of the system identified, it is best to summarize them in a diagram. This is given in figure 3.6. The circuit designed in the next chapter is based on this configuration.

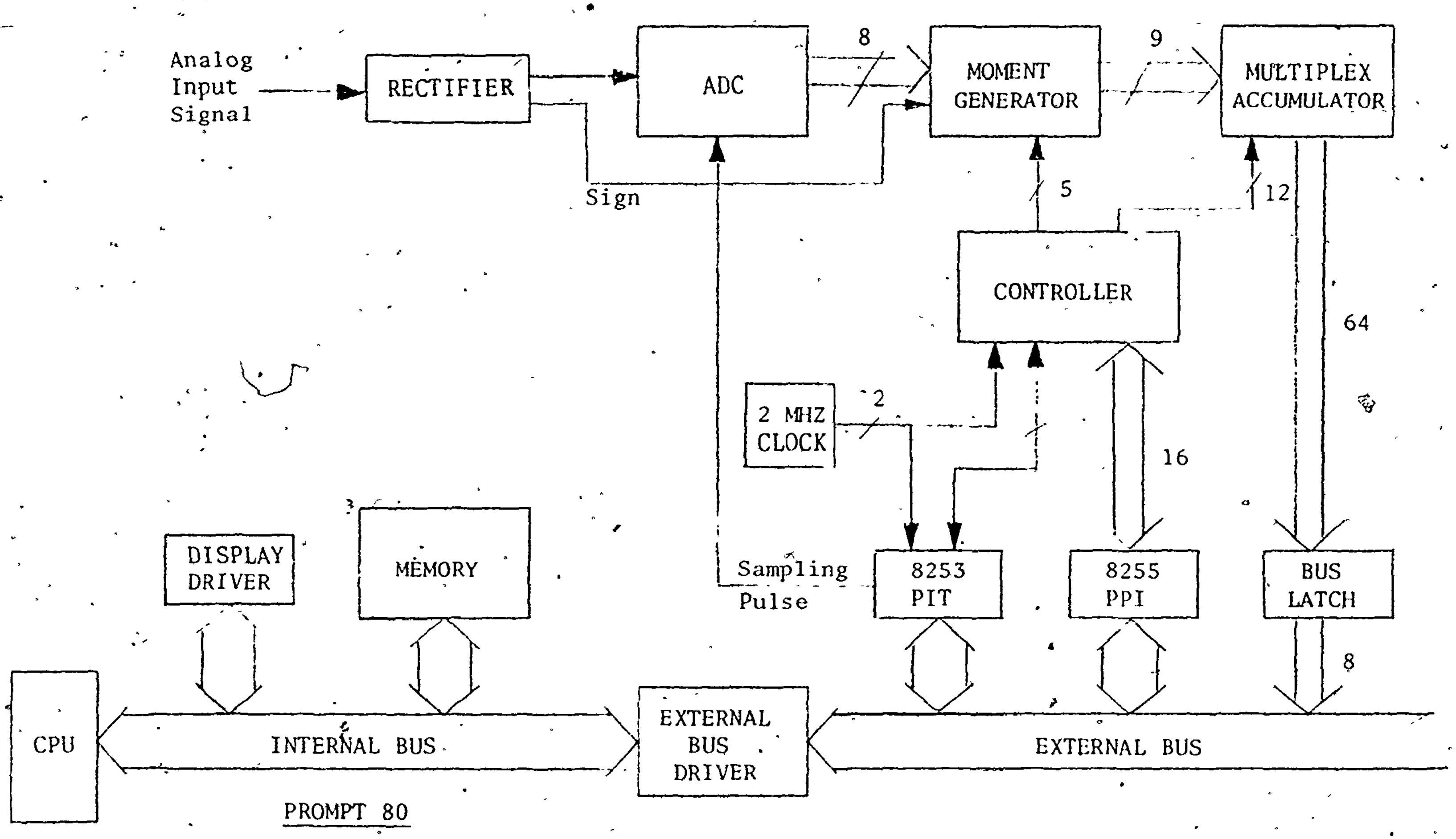

Fig. 3.6 SYSTEM BLOCK DIAGRAM

3.4 Software Requirements

Although there seems to be very extensive use of hardware in the instrument, the parts played by the software are also considered very important. Measurement parameters and control commands are acquired through the keyboard and sent to hardware control registers with the aid of software routines. Partial results are accumulated and the final results are calculated and formatted for display. All these are achieved with an OS (Operating Scheme) that supervises software as well as hardware activities. Since the Prompt 80 system monitor has occupied two and a half of the four kilobyte ROM space on the SBC 80/10, the OS is limited to one and half kilobyte in length. As an add-on software module, the OS does not interfere with normal operation of the Prompt 80 except when control is transferred to it. However, two routines and the display refresh service of the Prompt 80 monitor are utilized to minimize the length of the OS. Most of the functions are implemented with callable subroutines so that they can be shared by different parts of the module as well as Prompt 80 user programs. For a brief description, it is best to trace the actions right from the beginning.

Before any measurement can be initiated, it is necessary that all hardwares are set to known states and all memory registers are initialized to their desired values. This is taken care of at the beginning of the OS when control of the Prompt 80 console is first transferred to it. Measurement mode is set for one cycle of the input analog signal and the display is blanked by filling its buffer with blank characters. A measurement can then be started but as it is intended that measurement parameters such as sampling rate and number of samples are to be alterable, a control parameter entry loop is entered to allow for the changes.

Through the console keyboard, user desired parameters can be entered to override those that have been set during initialization or a previous measurement. Every relevant key entry is monitored on the display and all illegal entries are disregarded. The loop is exited upon the depressing of the execution key after which all measurement parameters are checked before the Start/Stop line is set to start a measurement. Any illegal parameter detected causes a branch back to the loop with the corresponding indication on the display. This eliminates any user error in using out of range parameters for a measurement. Routines for the manipulation of keyboard entries and display are essential parts of the loop and also a decision table is required for appropriate branching according to key entries.

During the measurement, the CPU loops until the sample ready latch is set and accumulates the partial results. Since it is a time critical process, the display is disabled so that no other interrupts can happen except the one that terminates the measurement, and in order to ensure that the accumulation is not disturbed, interrupt is only enabled during the checking loop. This restricts the end of measurement interrupt to be acknowledged after a complete set of partial results have been processed.

After the interrupt, the last set of partial results are accumulated and the OS commences with the final calculation of the results to be displayed. All four accumulated moments are normalized by dividing by the number of samples and from these values, the standard deviation and root mean square of the input signal are obtained. The period of the measurement is also calculated by multiplying the sampling rate with the number of samples. Display is then enabled and the OS enters a result display loop in which any of the above calculated results can be displayed with a corresponding label upon the depression of an appropriate key on the console. If further measurements are desired, it is possible to clear all the registers and branch back to the parameter entry loop through a single key entry.

With the major operations in the OS identified, a flow chart can be drawn for greater clarity. This is given in figure 3.7 in which the OS is divided into four phases, namely the initialization phase, the parameter acquisition phase, the measurement phase and the result processing and display phase. The initialization phase is executed only once upon the start up of the instrument while the others can be executed as many times as controlled by the keyboard. More detailed descriptions of the OS are found in Chapter 5.

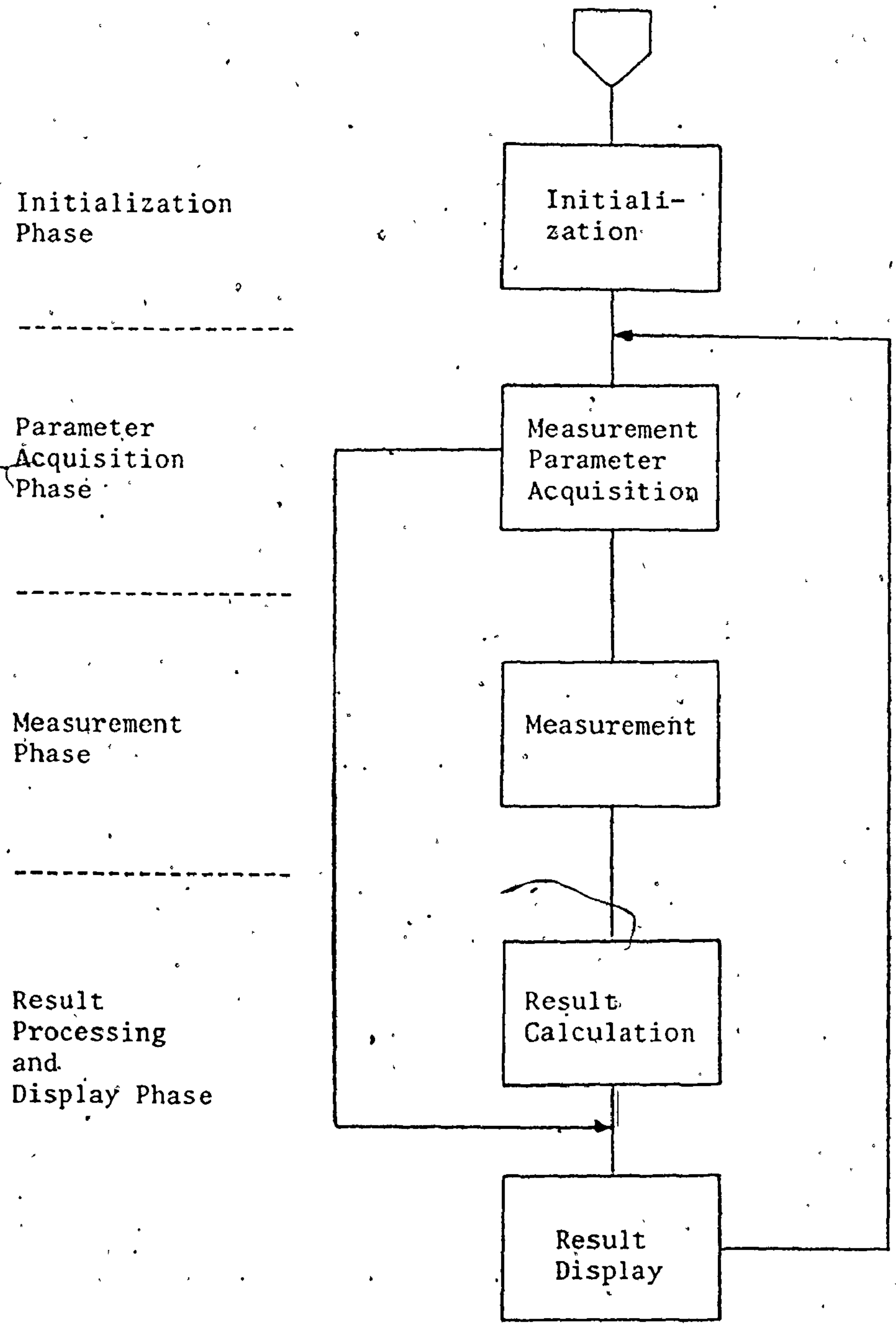

Fig. 3.7 OPERATING SCHEME FLOWCHART

Chapter 4

Hardware Design

4.1 Introduction

The principal duties of the hardware are to perform high speed date acquisition and processing and to achieve proper communication between the add-on sub-system and the host microcomputer system. It is designed with special considerations given to cost and space effectiveness. Every effort is made to keep the chip count to a minimum and in some parts the speed of a process is sacrified just to retain this rule. LSI chips are employed wherever possible to reduce chip count as well as the amount of connections between the chips although these complicated chips are found to be extremely difficult to test without the aid of sophisticated equipment that has only recently been developed.

The hardware can roughly be partitioned into five functional blocks of which the timing and control circuit is the most complicated. Two MOS and one Schottky bipolar LSI chips are involved in the design and the rest are either normal or Schottky TTL chips. The only analog circuitry is found in the sampling circuit in which three operational amplifiers, a high speed comparator and a monolithic hybrid ADC are used to produce an eight bit sample and a sign bit. These add up to a total of about 80 chips and are wire-wrapped on two 6.75 by 12 in. prototype boards that can be plugged into an INTEL SBC 614[11] cardcage specially manufactured for accommodating extension boards of the INTEL SBC series products. Communication between the two boards is achieved through unused lines of the SBC bus as listed in table 3.3.

The Prompt 80 is a self-contained microprocessor development system that has not been designed to allow for much hardware expansion. There are no provisions in the system for accommodating extra boards and the power supply unit has very little surplus power for driving external logics. To remedy this, an INTEL System 80/10[12] packaged computer that includes an SBC 80/10 card in an SBC 604 cardcage and a power supply unit is used instead of the SBC 80/10 in the Prompt 80. By connecting the keyboard and display assembly to this system through cables, a functionally identical and expandable Prompt 80 is obtained. The instrument is connected to the system by stacking the SBC 614 cage directly on the SBC 604 so that the buses of the cardcages are mated via their edge connectors.

4.2 Sampling Circuit

To produce a signed nine bit digital sample from an ADC with eight bit resolution, it is necessary that the input signal be rectified before sampling, and a comparator is used to produce the extra sign bit. All these are shown in figure 4.1.

The three analog amplifiers are Burr Brown 3507J[13] fast-slewing operational amplifiers. A1 acts as a unity gain buffer that serves to isolate the input signal source from the low input impedance absolute value circuit[14] formed by A2 and A3. The 220 pF capacitor at its input is for bypassing high frequency noise that may come with the input signal.

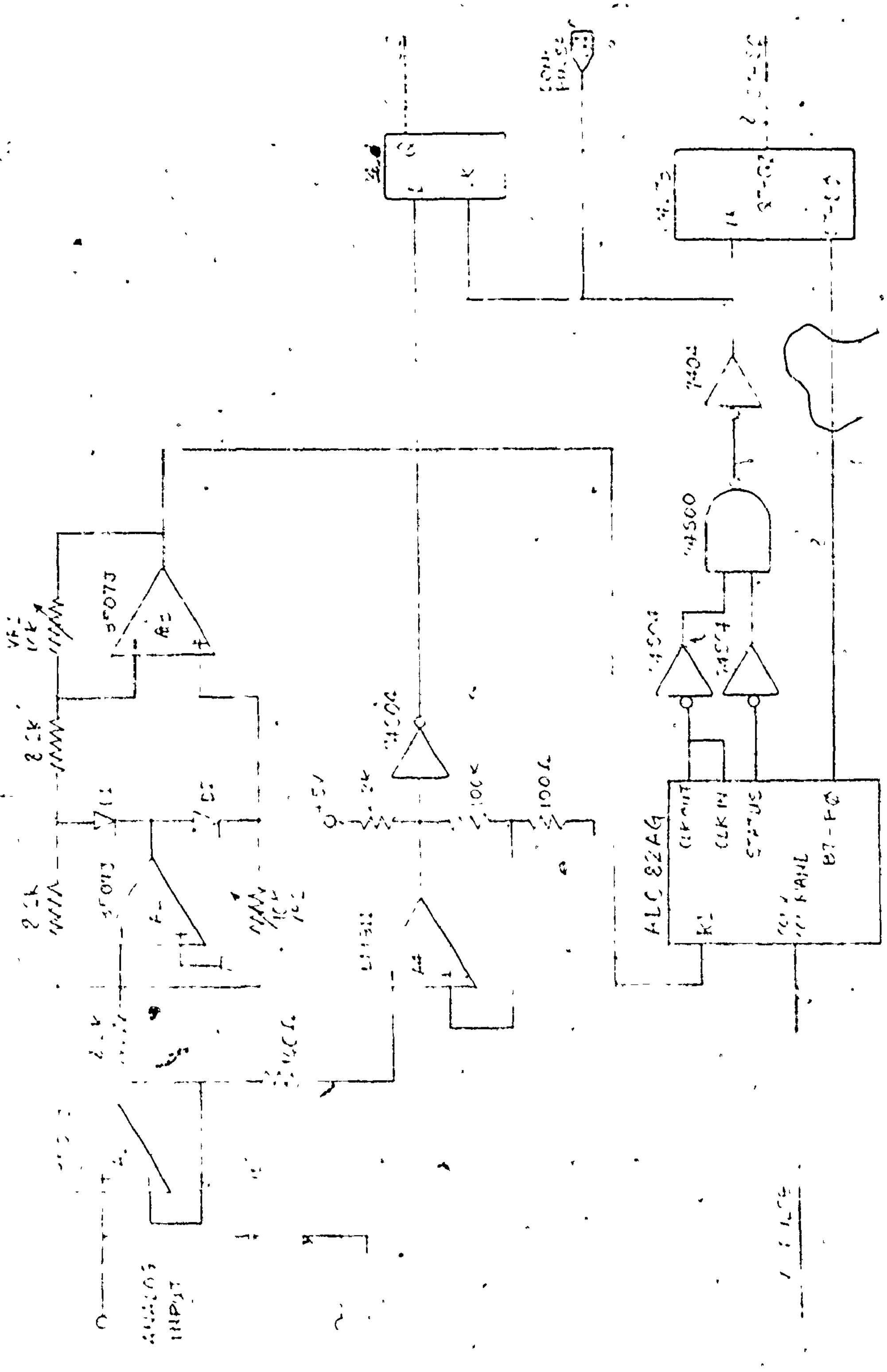

Figure 4.1 SAMPLING CIRCUIT

An LM311 high speed comparator produces the sign associated with the input signal and a 50mV hysterisis loop is introduced into the circuit through feed back resistors to prevent false triggering by noise during a one-cycle measurement. During the positive cycle of the input signal diode D2 at the output of A2 is off and A2 is connected as an unity gain inverter. Since the positive input of A3 is connected to the virtual ground of A2 through VR2, it also acts as an unity gain inverter and the double inversion passes the input signal to the output unchanged. During the negative cycle, D1 is turned off so that -2/3 of the input signal appears at the positive input of A3. A3 is now configured as a non-inverting amplifier with a gain of 3/2 by grounding its feed back resistors at the virtual ground of A2. Thus the input signal is inverted and the absolute value function is achieved.

Analog to digital conversion of the input signal is done by the ADC82AG[15] hybrid ADC that is wired to provide a 0 to 10V range. The ADC is of the successive approximation type with an internal 2.85 MHz clock source for clocking a 2.8 μs conversion. Upon triggering by the falling edge of a convert pulse, the status output goes high and nine clock pulses are generated by the internal clock for a conversion, after which the status output goes low again. If the convert pulse line turns high during a conversion, the process is aborted and another conversion can be initiated again at its falling edge. A timing diagram of these lines and the internal clock are given in figure 4.2. When a conversion is completed, a done pulse is issued to the arithmetic circuit to start the processing of the digital sample then available. This done pulse also latches the sign from the comparator and the complemented digital sample from the ADC into a 7474 D-type flip-flop and a 74273 octal D-type flip-flop respectively.

Generation of the done pulse is by the logical combination of the status line and the internal clock using high speed Schottky TTL gates and the timing is also shown in figure 4.2. It is important to note that the done pulse may be of variable width depending on the sampling rate and only the rising edge can be used for precise triggering of events. For the fastest sampling rate of 3 μs, the convert pulse must not exceed 200 ns and done pulses of at least 25 ns in width are generated.

The sampling circuit requires ±15V supplies for the analog amplifiers in addition to the +5V TTL supply. They are obtained from a modular +5V to ±15V DC to DC converter Model 546[16] from the Burr Brown Corp.

4.3 Timing and Control Circuit

This is the most complex part of the design and is made up of four functionally distinct blocks that are responsible for translating software commands from the microprocessor into hardware signals and sequencing the step by step processing of a digitized sample. A 2 MHz two phase clock provides accurate timing in the system for the generating of sampling commands and latching signals for the moment generator and multiplexing accumulator. Timing signals for other parts of the circuit are derived from these signals so that the amount of hardware can be reduced. A block diagram of this circuit is shown in figure 4.3.

4.3.1 System Clock

Sequential multiplexed processing requires accurate timing signals for clocking latches. If a single ~~phase~~ high speed clock is counted by a counter the outputs of which are decoded to generate the latching pulses, the length of the counter may be excessive and the decoding logic

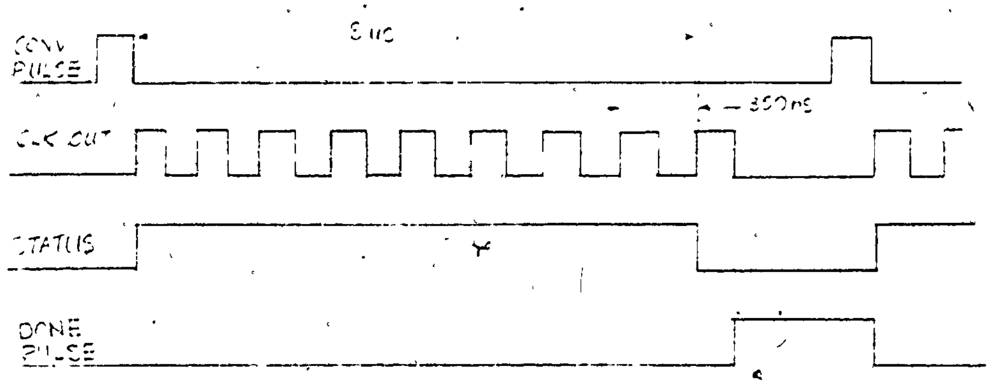

Figure 4.2 ADC TIMING DIAGRAM

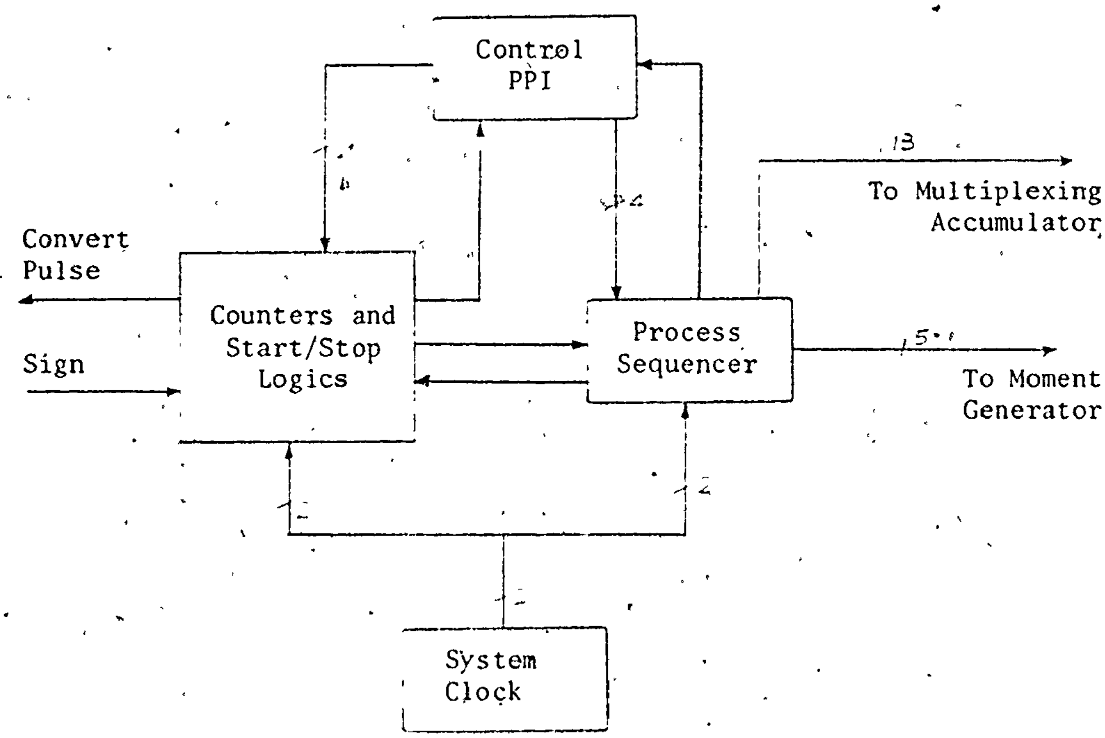

Figure 4.3 TIMING AND CONTROL CIRCUIT BLOCK DIAGRAM

may also be complicated. One way of avoiding this is to employ a multi-phase system clock which has a different duty cycle for each of the phases. Since by simple inversion, latching can also be done with a falling edge as well as a rising edge and vice versa, the multi-phase system clock has in fact more flexibility in addition to its simplicity in design. To determine the optimal number of clock phases in the system, various processing blocks are analyzed and it is found that the multiplexing accumulator requires at least two phases of the system clock for accumulating one of the moments. Thus a two phase 2 MHz clock is designed and the circuit and its timing diagram are given in figure 4.4.

A crystal controlled 20 MHz oscillator[7] is employed as a high frequency reference. It consists of two 74S04 inverters that are biased by 330Ω feedback resistors to work as small signal inverting amplifiers. By cascading these inverting amplifiers and using a crystal as a feedback element, a stable oscillator is obtained. The 680 pF capacitor between the inverters is for DC blocking, which is necessary for maintaining the DC operating point of the amplifiers. To obtain the two phase clock, the output of the oscillator is counted by a 74160 decade counter and its outputs are decoded to give two 2 MHz non-overlapping clock signals Ø1 and Ø2 of 40% and 20% duty cycles respectively. Most parts of the instrument are synchronized to this two phase clock and more detailed applications are reviewed in subsequent sections.

4.3.2 Process Sequencer

The multiplexing accumulator requires one cycle of the two phase system clock to accumulate one of the moments generated from a sample. Thus for four moments, a total of exactly four cycles of the system clock

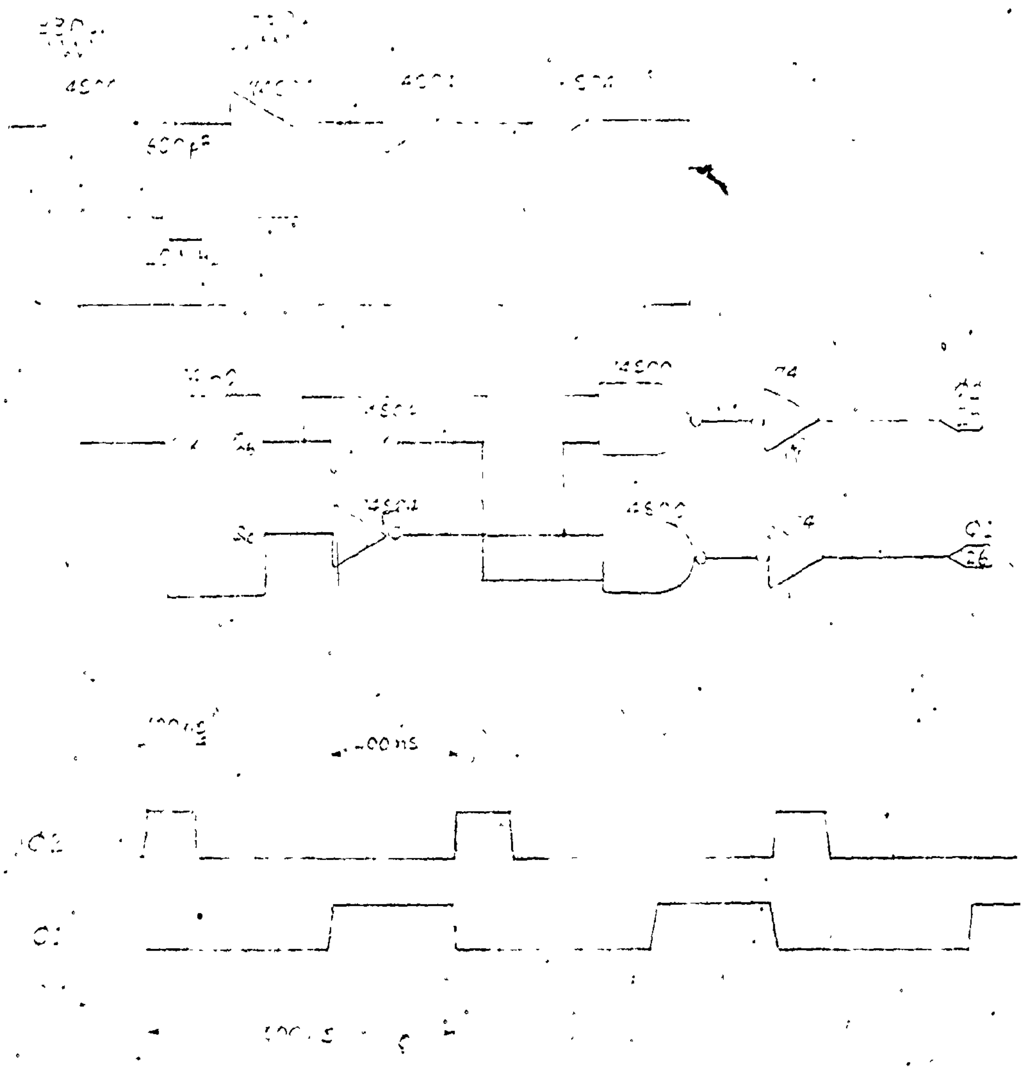

Figure 4.4 SYSTEM CLOCK AND TIMING DIAGRAM

is necessary. This is achieved by triggering with the done pulse from the sampling circuit a two bit synchronous counter that stops itself after counting four cycles of the system clock. By doing so, the arithmetic circuit is synchronized to the sampling circuit and its timing signal can be obtained from the decoded outputs of the counter. The circuit and the timing diagram is shown in figure 4.5.

In the stable state, all 7474 D-type flip-flops are cleared and the counter formed by the two 7476 JK flip-flops is in a 11 state. Since both clock sources into the 74155 one channel to four channel demultiplexers and the counter are gated by the Q output of FF1, no timing signals are produced and the circuit remains stable. When a done pulse arrives, the Q output of FF2 is set and it frees FF1 from its cleared state. The D input to FF1 has also turned high so that the rising edge of the next Ø2 sets FF1 and passes clock pulses to the counter as well as the demultiplexers. This guarantees that the first Ø2 pulse going to the counter must have a minimum width of 80 ns. Without FF1, a pulse of smaller width may occur if the Q output of FF2 is used for gating the clock source and FF2 is set when Ø1 or Ø2 is high. In this case, however, only the first Ø2 pulse of a cycle is compressed by 20 ns and the widths of the rest remain the same as the system clock.

The clock pulses passed to the counter and the multiplexers are inverted. Thus the counter is toggled every 500 ns at the rising edge of Ø2 due to the JK control of its flip-flops. Outputs of the counter are fed to the demultiplexers to control the passing of clock pulses to each channel. Channels T1/ to T4/ each get one negative Ø2 for each cycle of the counter and since the counter is also clocked by Ø2/, a delay of 40 ns is added to the Ø2/ pulses to the demultiplexer so that the low state of

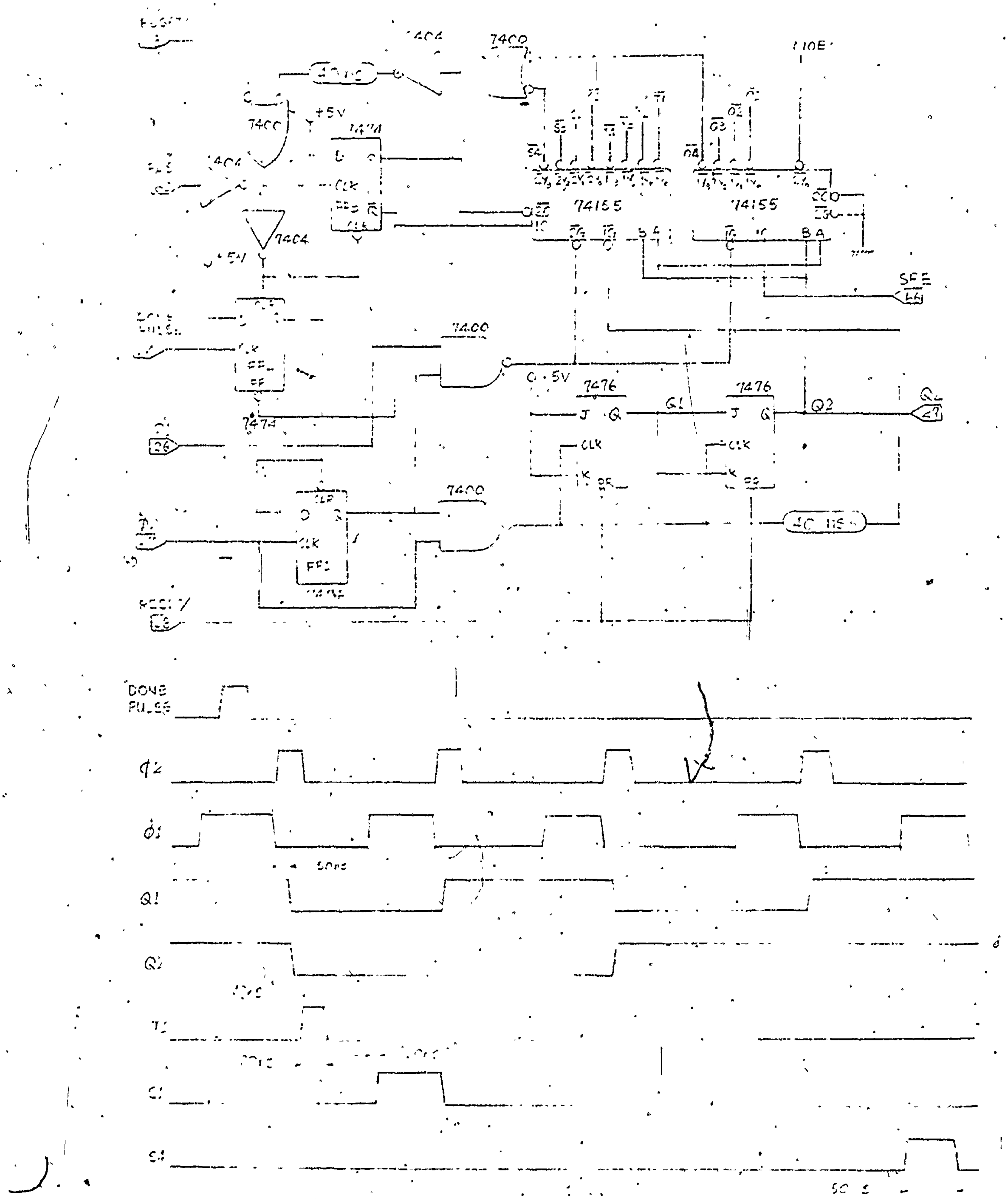

Figure 4.5 PROCESS SEQUENCER AND TIMING DIAGRAM

Ø2/ cannot coincide with the toggling of the counter. In this way, the splitting of a pulse between two channels is prevented. Similarly the four Ø1/ pulses during a counter cycle are distributed among channels S1/ to S4/ except that no delaying of the Ø1/ pulses to the demultiplexers is necessary. This also has the advantage of shortening the gap between the rising edge of Ø2/ and the falling edge of Ø1/ in corresponding channels, which is what suits the multiplexing accumulator.

The signals 01/ through 04/ coincide with S1/ through S4/ except that they only occur once in 128 counter cycles when the SRE line is high. They are for latching the contents of the multiplexing accumulator to a set of bus latches so that the partial accumulations can be picked up by the microprocessor. To generate the SRE signal, the Q2 output of the counter is counted by another counter which generates a carry at its 128th count. This is discussed in more detail in the next sub-section. One of the demultiplexers also produces the MOE/ signal that is low during the first state of the counter cycle. It is for routing of data in the moment generator and passes the first moment directly to the multiplexing accumulator. A timing diagram of these control signals is given in figure 4.5.

The counter cycle is terminated by S4/ which clears all the D-type flip-flops and restores the stable state. As there is only a delay of only about 150 ns in the loop before the gate for passing Ø1/ pulses to the demultiplexers is disabled, the width of the last Ø1/ pulse to the demultiplexer is shortened to 150 ns and so are S4/ and 04/. Another way of resetting the circuit to its stable state is by pulling the RESET/ line low which also clears all the D-type flip-flops and presets the counter. This is usually done at the start up of the system when a reset

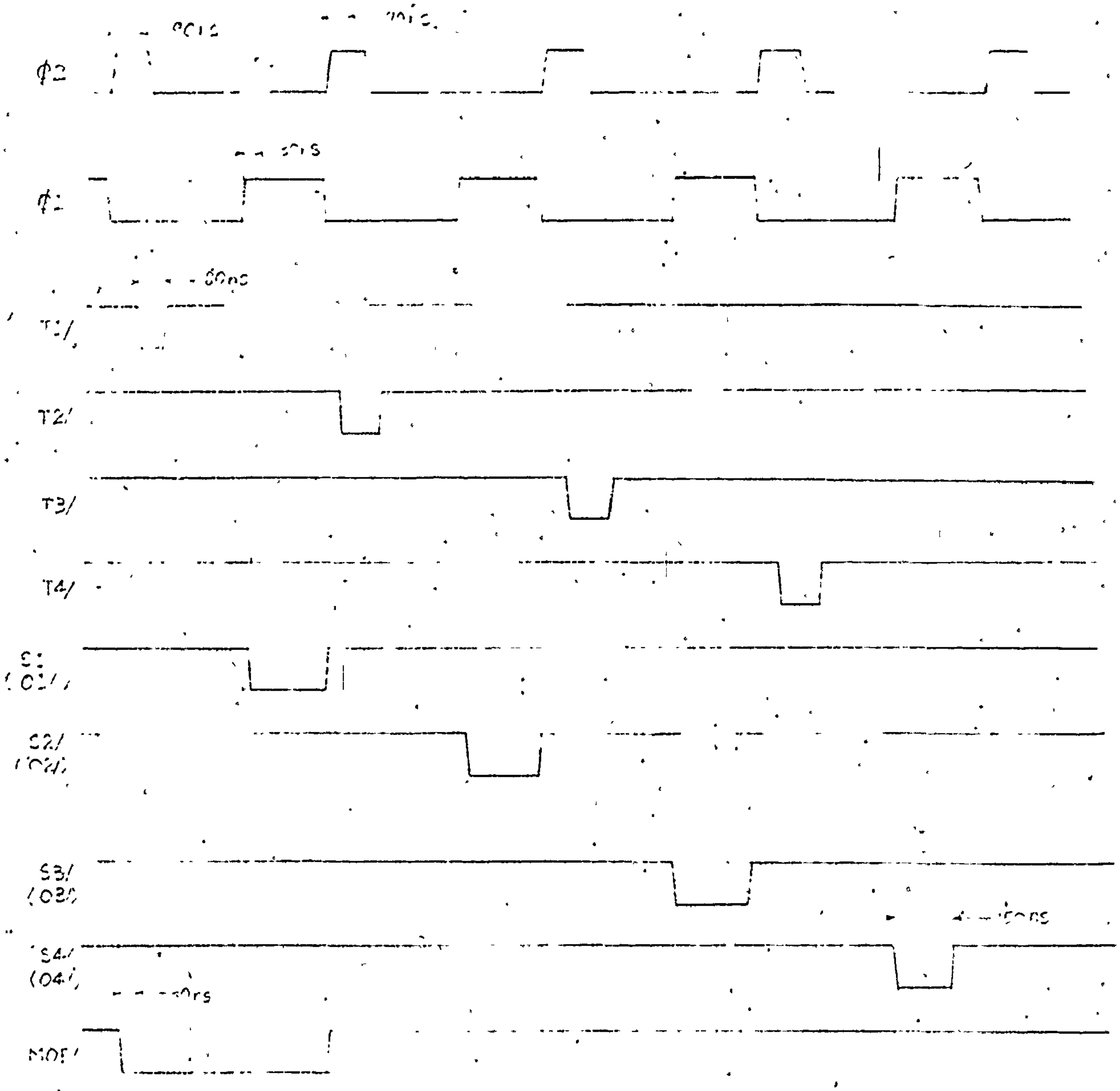

Figure 4.6 CONTROL SIGNALS TIMING DIAGRAM

signals is sent to all parts of the instrument. The D-type flip-flop FF3 is only activated at the end of a measurement when software has regained control of the processing. A pulse from the FLS/ line sets FF3 and induces an extra counter cycle in which only the signals 01/ to 04/ are generated. This is done by disabling the control inputs of channels T1/ through T4/ and S1/ through S4/ which are tied to the outputs of FF3. The MOE/ signal has no adverse effect during this operation and is allowed to remain.

After a set of partial accumulations are transferred to the bus latches, the microprocessor has to be notified so that it can receive them and the multiplexing accumulator has to be cleared. The signal 04/ is found to be most useful in this situation. By setting two latches with the delayed rising edge of 04/, both of the above tasks can be achieved simultaneously. The concept is depicted in figure 4.7 in which the Q output of L1 goes directly to line PB6 of port B of the control PPI. This port has been programmed to be in input mode and during the course of a measurement, the microprocessor keeps checking the status of line PB6. When it is detected set, the microprocessor clears L1 with the SRLC/ line and proceeds to pick up the partial accumulations. 04/ also latches a low into L2 and the Q/ output of L2 is NORed with the reset line to produce the signal CCL/ for clearing the multiplexing accumulator. Since during a measurement CCL/ must return high before the next sample is processed, L2 is preset with the next T1/ that marks the beginning of the next set of 128 samples.

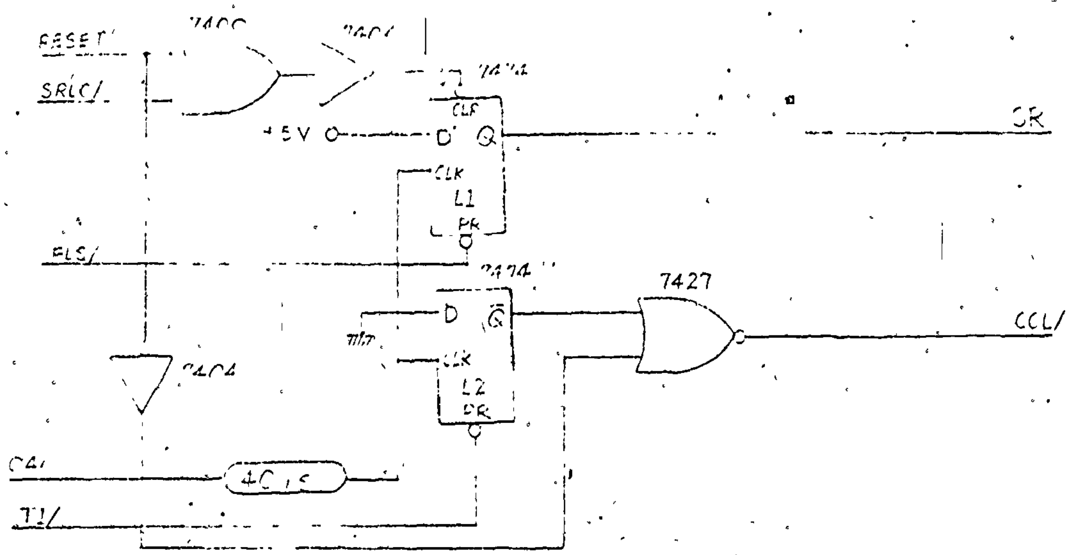

Figure 4.7 SAMPLE READY LATCH AND MULTIPLEXING ACCUMULATOR CLEARING CIRCUIT

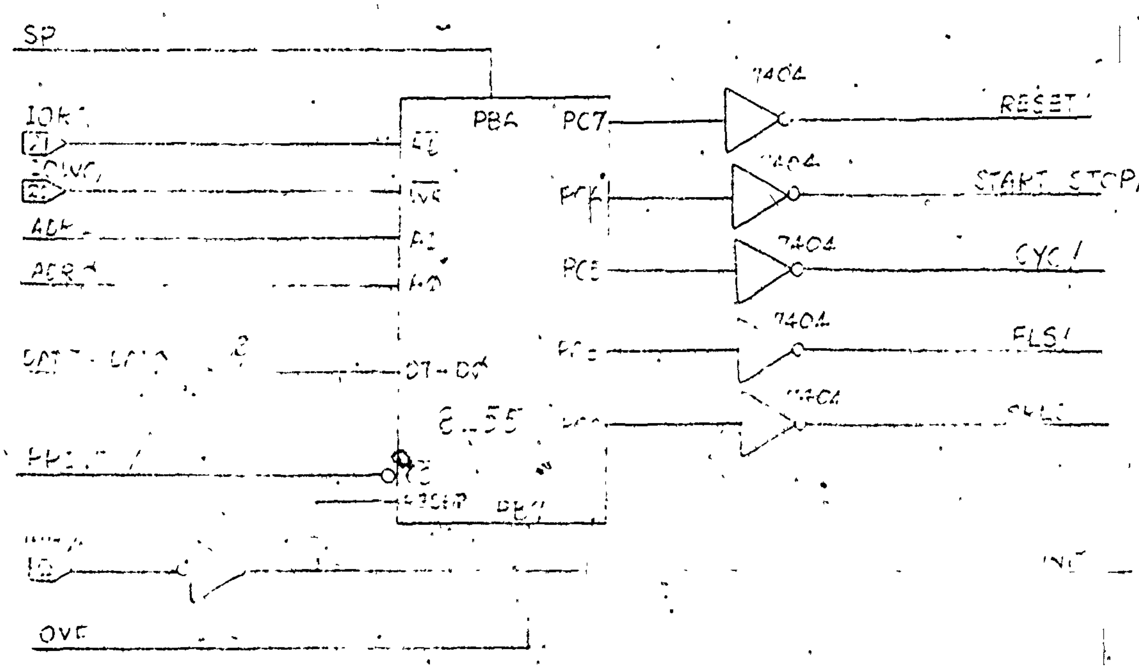

Figure 4.8 CONTROL PPI

4.3.3 Control PPI

Communication between the microprocessor and the hardware is achieved through two ports of an 8255 PPI[7] chip. Software commands for driving hardware are executed by setting or resetting bits of an output port to which hardware control lines are linked. The bit setting and resetting feature of port C of the PPI is extremely useful here. Port B is programmed as an input port for checking the status of the hardware. Since the PPI is a MOS device and its output lines can only drive one TTL load, these lines are all buffered by inverters before connection to other parts of the system. A brief description of the I/O lines used by the system is listed below and the circuit is presented in figure 4.8.

RESET	:	output line that resets all hardware in the instrument.
START/STOP	:	output line that starts an measurement when is set.
CYC	:	output line that sets the necessary logics for a one cycle measurement.
FLS	:	output line that strobes the final set of partial accumulations into the bus latches.
SRLC	:	output line that clears the sample ready latch.
SR	:	input line that monitors the sample ready latch.
OVF	:	input line that monitors the number-of-samples overflow latch.

Other lines into the PPI are from the bus interface that connects it to the SBC bus. These include the eight data lines, read/write signal lines, address lines for selecting the I/O ports and the chip select signal for enabling the PPI. There is also a reset line tied to the INIT/ signal of the SBC bus for power on reset.[12] The remaining I/O lines are left open and can be used for further expansion of the instrument.

4.3.4 Counters and Start/Stop Logics

Three software programmable counters are required by the instrument for generating sampling commands, recording the number of samples in a measurement and clocking the bus latches once in 128 samples. An 8253 PIT[10] (Programmable Interval Timer) is found to be most suitable for these purposes. It consists of three 16 bit programmable counters each of which can be individually programmed to achieve the three different functions mentioned above. Each counter has its own clock input and is controlled by a gate input that can inhibit the counting when pulled low. By setting different work modes for a counter, various kinds of signal synchronized to the input clock can be obtained from its output. All counters work in down count mode and are negative edge triggered. Interface of the PIT to the SBC bus is similar to that of the control PPI.

Counter Ø and counter 2 are programmed as rate generators. In this mode, the output of a counter goes low for one period of the input clock period when a programmed number of input pulses have been counted. The counter is then reloaded and can therefore continue with the counting process until it is inhibited. Counter Ø is for providing sampling commands and its rate is set before a measurement. The input clock is derived from Ø1 of the system clock by further dividing the 2 MHz clock by two with a D-type flip-flop wired to work in a toggle mode. Counter 2 is for generating the SRE signal that enables the signal O1/ through O4/ for latching the partial accumulations from the multiplexing accumulator to the bus latches. Its triggering negative edges are obtained from the Q2 output of the counter in the process sequencer so that the SRE signal can be synchronized to it. Since Q2 goes low at the start of every counter cycle of the process sequencer, the signal SRE turns high at the beginning of the 128th counter cycle and remains until counter 2 is triggered by the next falling edge of Q2, thus enabling the desired demultiplexer to produce signal O1/ to O4/.

Counter 1 is programmed to work in interrupt at terminal count mode. For a fixed sample measurement, its output goes low after the counter has been loaded with the desired number of samples and remains so until the counter reaches zero when it turns high again. The rising edge at zero count is used to set an interrupt latch which stops further generation of sampling commands and interrupts the microprocessor to signify the end of a measurement. Clocking of this counter is by inverted sampling commands generated by counter Ø. During a one cycle measurement, counter 1 is loaded with a maximum count of 10000 and if the counter reaches zero before one cycle of the input signal has elapsed, an overflow interrupt occurs and the same interrupt latch is set. The microprocessor distinguishes this overflow interrupt from the normal interrupt in a fixed sample measurement by checking the OVF line sustained by the interrupt latch through line PB7 of the control PPI during a one cycle measurement. A jump to display the word 'ERROR' on the console is executed if the OVF line is found true. Figure 4.9 depicts the counter circuit and its associated Start/Stop logics.

At power on of the instrument, the interrupt latches L3 and L4 are cleared with INIT signal available from the SBC bus. This is necessary because all control lines coming from the control PPI are also

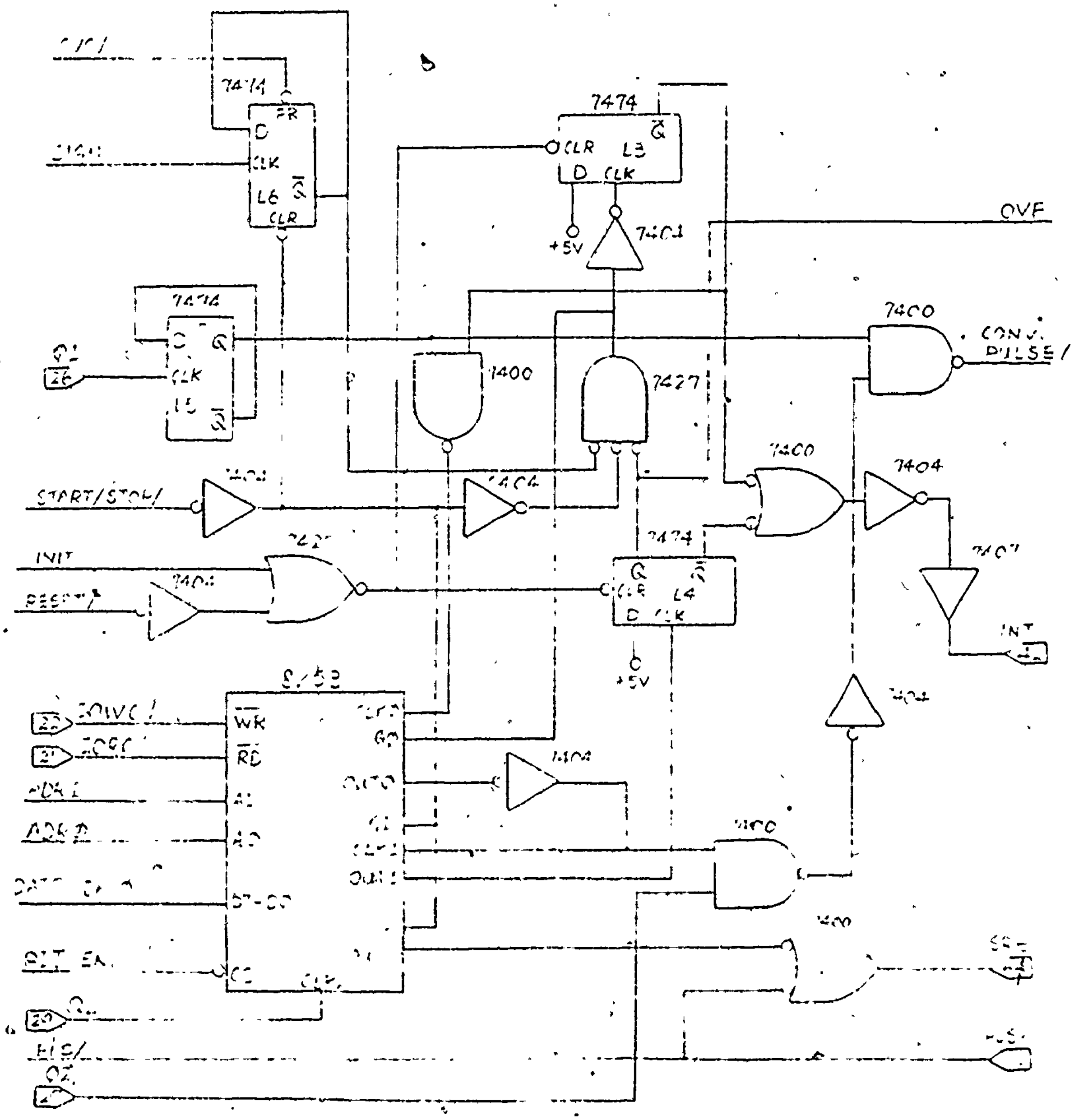

Figure 4.9 COUNTERS AND START/STOP LOGICS

reset to a high state and may cause undesirable interrupts. After the PPI has been set to the desired mode by software, all lines of port C turn low and the START/STOP line forces the circuit into a stable state. Counter 1 and counter 2 are inhibited directly through gates G1 and G2 and the measurement mode control latch L6 is forced to zero. Since only one input to the 7427 NOR gate is low, counter Ø is also disabled through gate GØ although the toggle flip-flop formed by L5 is still supplying a 1 MHz square wave through a NAND gate to its clock input.

The mode of a measurement is solely controlled by the CYC/ signal which sets up a fixed sample measurement when it is low. When the START/STOP/ line turns low and CYC/ is low, L6 is preset and the output of the 7427 NOR gate goes high. This enables counter Ø which starts generating sampling commands according to the rate set by software. The sampling commands are inverted to drive a NAND gate which passes negative Ø2 pulses to another NAND gate controlled by the Q output of L6. With this arrangement, the case of passing two convert pulses to the sampling circuit by the 1 μs wide sampling command pulse is avoided as L6 is toggled every 500 ns by Ø1. Since the START/STOP/ line also releases counters 1 and 2, the counting processes as described previously start and continue until L4 is set by the output of counter 1 at its terminal count. This disables counter Ø through the 7427 NOR gate and pulls the interrupt request line INT/ on the SBC bus low through a NAND gate, an inverter and an open collector driver to initiate an interrupt. The circuit can then be restored to a stable state by disabling the counters with the START/STOP/ line and clearing the interrupt latches L3 and L4 with the RESET/ line.

For a one cycle measurement, the CYC/ is held high and L6 is free to toggle according to the SIGN signal coming from the sampling circuit after the START/STOP line has turned low. A negative to positive zero crossing of the input analog signal toggles L6 from low to high and enables counter Ø through the 7427 NOR gate. This starts the sampling process and if no overflow of counter 1 occurs, the next negative to positive zero crossing of the input analog signal toggles L6 back to low and disables counter Ø. The falling edge at the output of the 7427 NOR gate is used to set L4, which, besides generating an interrupt request through the same series of gate as L3 does, alos stops the passing of further clock pulses to counter Ø so that it cannot generate sampling commands when L6 is toggled again. Restoring of the circuit to a stable state is also by the same sequence of signals as in a fixed sample measurement.

4.4 Moment Generator

The moment generator is responsible for the generating of higher order moments from a digital sample and its sign. Its central building block is an eight bit signed two's complement bipolar multiplier[9] that is capable of producing a fifteen bit signed two's complement product in 150 ns. Since the digital sample is a nine bit sign and magnitude number, it has to be converted into its two's complement and truncated to eight bits before it can be fed to the multiplier and the product at the output of the multiplier has to be also truncated to eight bit to be fed back to the multiplier so that the higher order moments can be produced. The truncation has reduced the accuracy of the moments except in the case of the first moment in which the nine bit two's complement number converted from the sample is passed directly to the multiplexing accumulator. This is made possible with the tri-state outputs of the multiplier by attaching to them a set of nine tri-state drivers through which the first moment can be passed to the multiplexing accumulator. The MOE/ signal from the process sequencer is specially designed to control this. It disables the multiplier outputs and enables the drivers during the time the first moment is accumulated. Summarized in figure 4.10 is a circuit of the moment generator.

The digital sample S7 through SØ from the sampling circuit is in complemented form. If the sign of the sample is positive as indicated by line SS being low, it has to be complemented before it can be passed with the sign SS through the tri-state drivers to the multiplexing accumulator. This is achieved by passing it first through a pass/complement element made up of two 74H87s which complements the sample when the line SS is low. Otherwise the sample is treated as negative and is passed directly to the drivers. The incrementing of the negative sample by one to form the two's complement is done at the multiplexing accumulator by presenting a one to its carry input CØ through line TØ at the time the sample is accumulated. By feeding the most significant seven bits from the pass/complement element and the sign SS to an eight bit adder that has been wired in such a way that only when the sign is negative a one is appended before the seven bits and the whole is incremented by one, an eight bit two's complement version of the sample is obtained. It is presented to the X inputs of the multiplier and is latched in by the rising edge of T1. The other eight bit two's complement number is presented to the Y inputs of the multiplier by the truncation circuit at the output of multiplier which truncates the nine bit first moment and fifteen bit second and third moments to eight bits. They are latched into the multiplier with the rising edges of T1/, T2/ and T3/ respectively so that

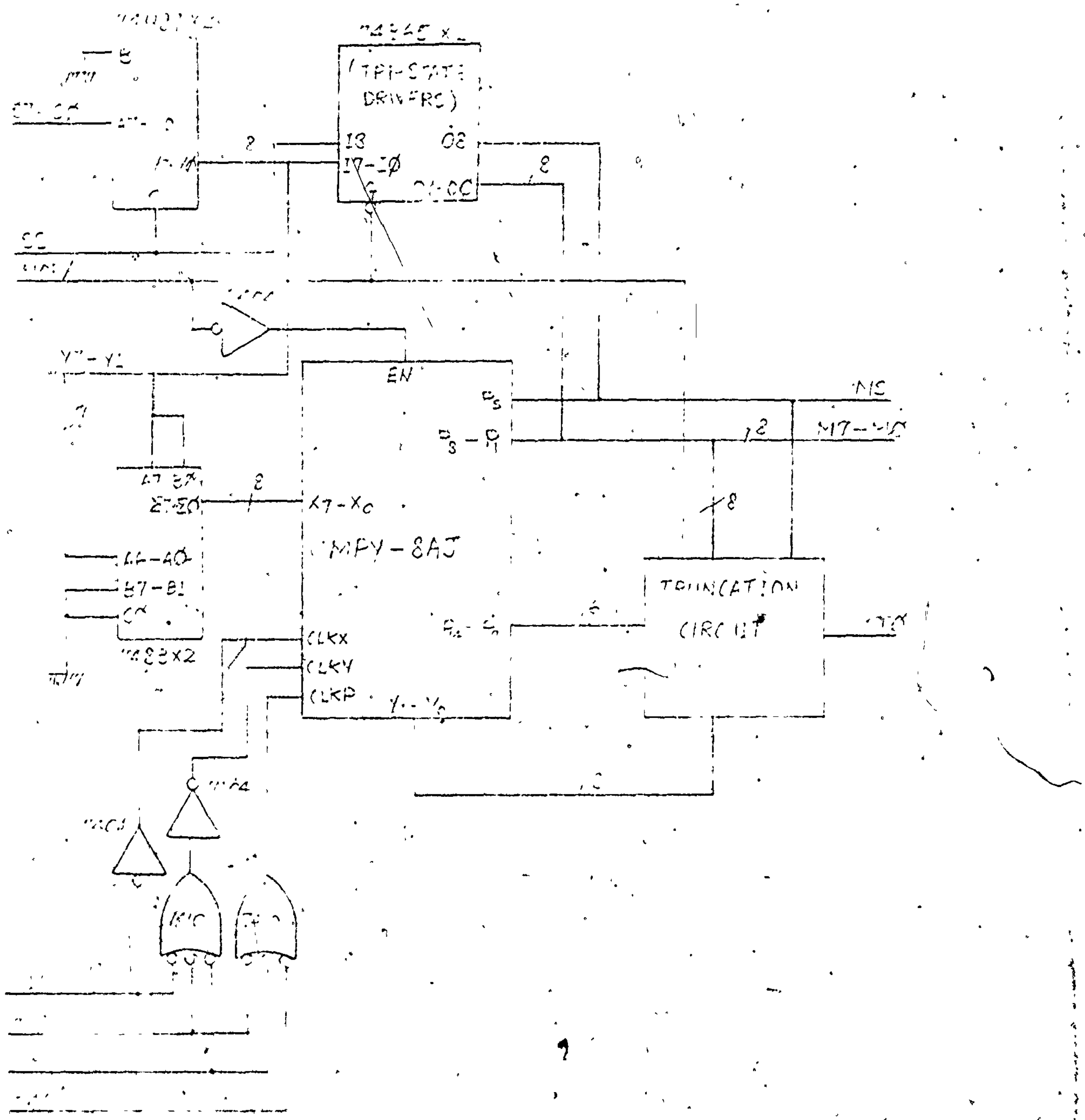

Figure 4.10 MOMENT GENERATOR

the latching always occurs about 100 ns after the moments are available on lines MS and M7 through MØ to allow for the settling time of the tri-state outputs and propagation delay through the truncation circuit. The higher order moments produced by the multiplier are latched to its outputs by the rising edges of T2, T3 and T4 during the time the output of the multiplier is enabled by the MOE/ line.

The truncation circuit is responsible for the supply of an eight bit two's complement number to the Y inputs of the multiplier and a nine bit two's complement number to the multiplexing accumulator from the output of the multiplier or the tri-state drivers. The circuit is depicted in figure 4.11. When the MOE/ line is low, the first moment is passed to lines MS and MØ through M7 by the tri-state drivers. If MS is high, the moment is negative and the combination of MS and inverted MOE/ forces TØ high, which is then added to the multiplexing accumulator through its CØ input to give an equivalent two's complement adding. The feedback of an eight bit two's complement number to the Y inputs of the multipler is through an eight bit adder which performs the same operation on the most significant seven bits M7 to M1 as the one preceding the X inputs except that the adding of one to the truncated value through BØ is controlled by the same combination that forces TØ to a one.

When the MOE/ line is high, a fifteen bit product is available from the multiplier. If it is negative, the truncation is done with an algorithm which increments the truncated value by one if any of the bits that have been truncated is a one. This is based on the fact that the only case that a one can propagate to the least significant bit of the truncated value during a two's complement to signed magnitude conversion before the truncation is when the truncated bits are all zeros. Any

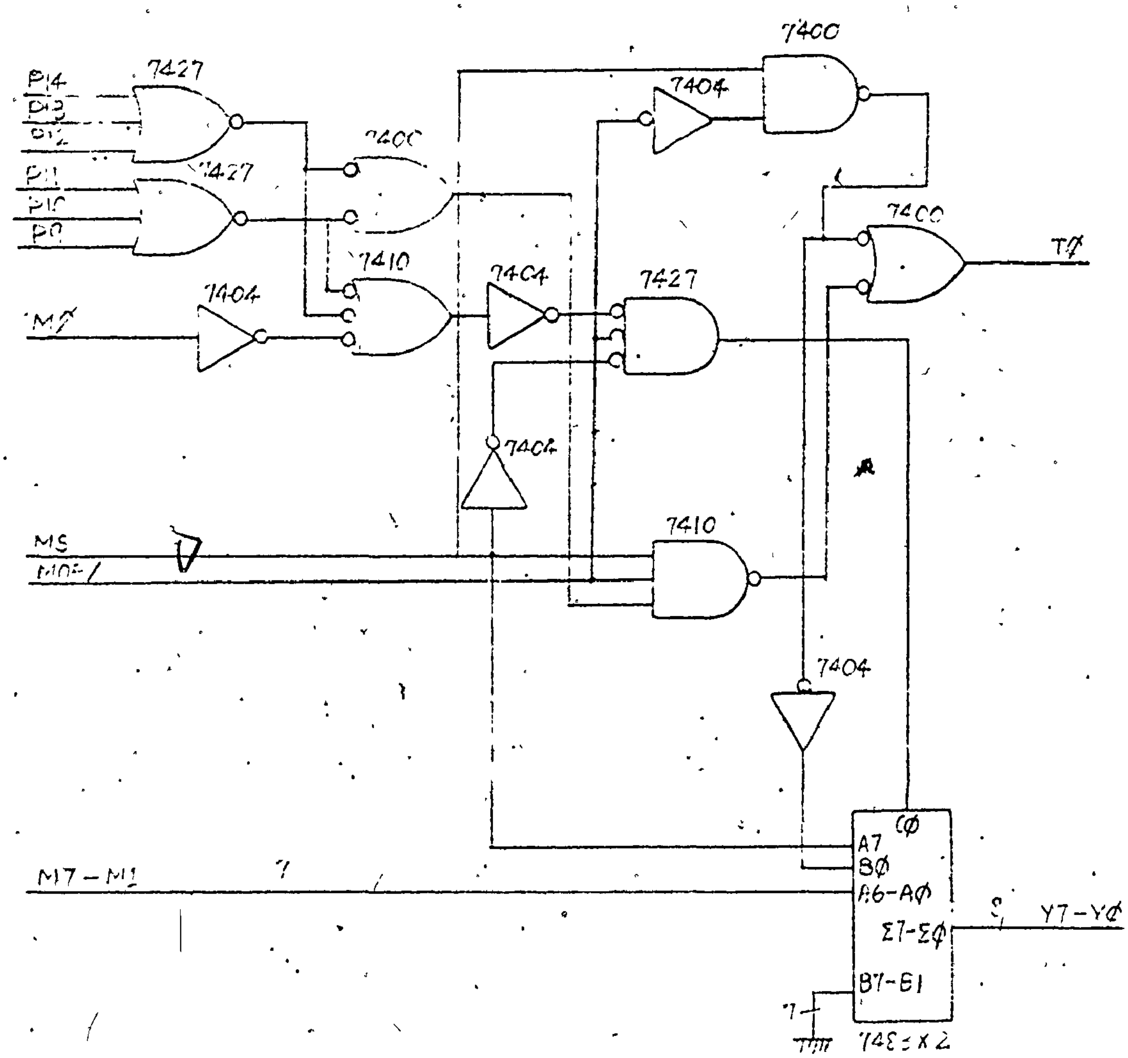

Figure 4.11 TRUNCATION CIRCUIT

truncation besides this resembles a truncation of a one's complement number and requires a one to be added to the least significant bit to form the two's complement. In the truncation circuit, the truncation to nine bits is done by ORing the least significant six bits of the product from the multiplier and forces TØ to high to add one at the multiplexing accumulator if MS, MOE/ and the result of the ORing are all high. For the truncation to eight bits to be fed to the Y inputs, the least significant seven bits of the product are ORed and then combined with MS and MOE/ to supply a one to the CØ input of the adder to be added to the truncated eight bit number. In all cases if the number to be truncated is positive, MS is low and TØ and the inputs to CØ and BØ are forced low, which means the truncation is simply done by dropping the desired number of bits.

According to the timing signals that control the moment generator, each of the first three moments appears successively as a nine bit two's complement number at the inputs to the multiplexing accumulator for 500 ns after the counter in the process sequencer is triggered to start a new cycle. The fourth moment follows but remains until the next processing cycle is triggered.

4.5 Multiplexing Accumulator

The part of the multiplexing accumulator for accumulating the first moment is shown in figure 4.12. During the first 500 ns of a processing cycle, the first moment is presented to the A inputs of the sixteen bit adder and extended to sixteen bits by feeding MS to the most significant eight bit inputs of the adder. At the beginning of this 500 ns period, the outputs of the sixteen bit tri-state latch formed by

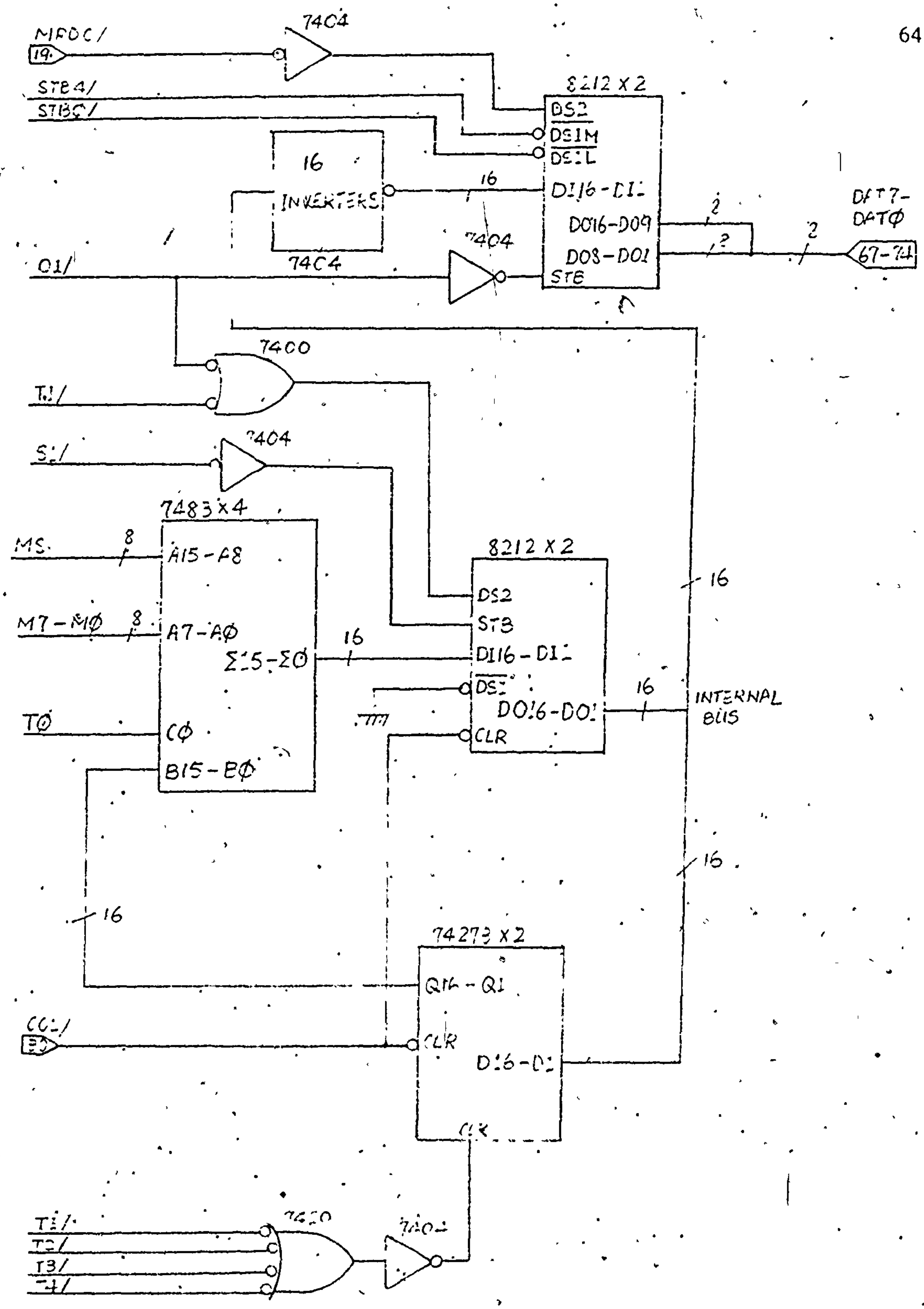

Figure 4.12 MULTIPLEXING ACCUMULATOR

two 8212[7] eight bit latches at the output of the adder are enabled by T1/ through the DS2 input for 100 ns. Since there is a delay of 50 ns before the data is valid on the internal bus, the rising edge of T1/ through the 7420 and an inverter can be used to clock the data into another sixteen bit latch formed by two 74273 octal D-type latches the outputs of which are fed to the B inputs of the adder to be added to the incoming sample. The outputs of the adder are latched back into the sixteen bit tri-state latch at the end of the 500 ns by the rising edge of S1/ that drives the STB input of the latch, thus completing the accumulation. At the 128th sample, the same sequence is repeated except that O1/ also enables the output of the tri-state latch during the last 200 ns as it coincides with S1/ and latches with its rising edge the inverted version of the data on the internal bus into two 8212s that are acting as bus latches for transferring the data to the microprocessor. Such latching is made possible by the special characteristic of the 8212 latch that the outputs of its eight internal D-type latches follow the inputs when the STB input is high and the latching occurs at the falling edge. The inversion before the bus latches is to compensate for the negative data structure of the SBC bus.

Each of these bus latches represents an unique read-only memory location that can be accessed by the microprocessor as their outputs are tied to the data lines of the SBC bus. Both STBØ/ and STB4/ signals coming from the bus interface logics are for enabling the latches and together with the MRDC/ signal, the most significant or least significant part of the partial accumulation can be presented to the SBC bus and picked up by the microprocessor. For the other moments, three other sets of sixteen bit tri-state latches are connected between the outputs of the adder and the internal bus for accumulating them and three corresponding sets of bus latches are connected to the output of the sixteen inverters to act as bus interfaces to the SBC bus. Signals T2/, S2/ and O2/ are used for clocking the accumulation of the second moment and the rest follow. After the partial accumulation of the fourth moment has been transferred to the bus latches at the 128th accumulation, all tri-state latches and the feedback latch are cleared so that another 128 accumulations can begin at the next processing cycle. This is repeated until the end of the measurement.

The multiplexing accumulator is a very efficient design that is capable of completing the accumulation of four moments in 2 μs with 500 ns being devoted to each moment. However, this is not its speed limit for a sixteen bit addition can be completed by the 7483 adders within 150 ns, which makes it possible to reduce the processing time to 1 μs for four moments. It also implies that the sampling frequency can be upgraded to 1 MHz if such sampling devices are available. Tighter timing requirements are necessary in this case.

4.6 Microprocessor Bus Interface

To the microprocessor, the instrument appears as a set of eight read-only memory locations from which the partial accumulations are picked up, and eight I/O ports through which a measurement is controlled. The reason for this configuration is that the partial accumulations are always fetched by the microprocessor in sequential order and the indirect memory addressing mode of the microprocessor is most efficient in handling this sort of fetch. On the order hand, the I/O parts are randomly accessed by the operating system before and during a measurement, which makes a memory-mapped I/O scheme comparatively slower and involves more software. Under such considerations, a combination of the two is found to be optimal. The circuit for such an interface is shown in figure 4.13.

The addresses of the bus latches for temporary storage of the partial accumulations range from FFF8 to FFFF, with the lowest address for the least significant part of the first moment and the next address for the most significant part. Thus the enable signals for the latches are obtained by ANDing the most significant thirteen bits of the address bus and using the result to enable a 74155 demultiplexer which is wired to act as a three line to eight line decoder for decoding the least significant three bits of an address into eight enable signals STBØ/ through STB7/. The result of the thirteen bit ANDing is also combined with the MRDC/ command from the microprocessor to produce the AACK/ signal which notifies the SBC 80/10 CPU that the data is ready.

Two I/O enable signals PPI EN/ and PIT EN/ are drived from the least significant eight bits of the address bus and are combined with the IORC/ or IOWC/ signals to produce the AACK/ signal necessary for a data transfer. Since the SBC data bus is negative true and the PPI and PIT require true data, an inverting bus transceiver formed by two 8226 inverting tri-state bus drivers is employed to provide bidirectional inverting buffering. This transceiver can be enabled by either I/O enable signals and the direction of the data flow is controlled by the IORC/ signal which only passes data from the PPI or PIT to the SBC bus when it is low. Addresses of the PIT and PPI start from F8 and end at FF.

The design of the interface circuit is based on the assumption that the peripheral chips are fast enough to respond during a read/write

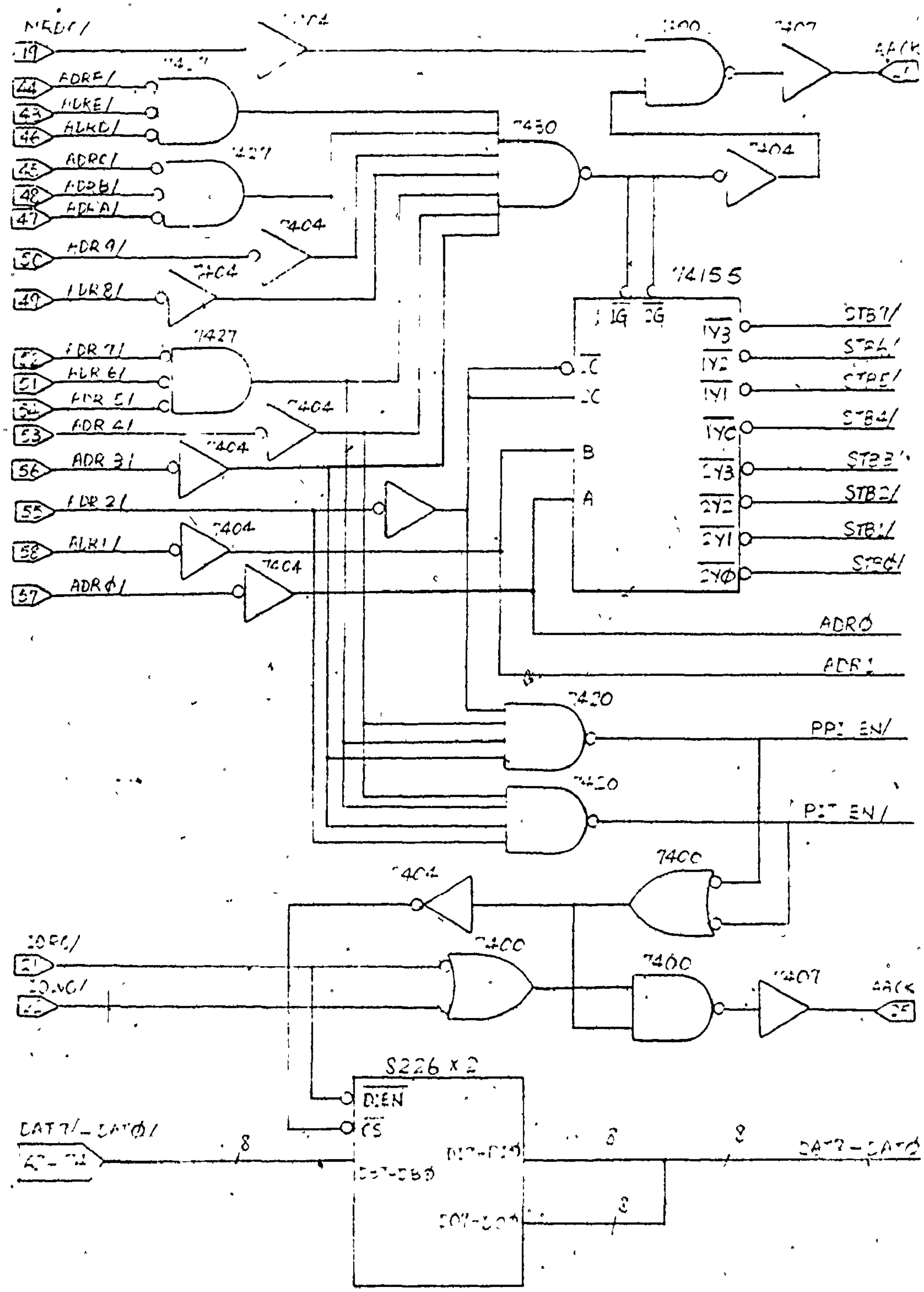

Figure 4.13 MICROPROCESSOR BUS INTERFACE

cycle of the microprocessor so that no wait state has to be inserted. Thus the AACK/ signal is returned to the CPU as soon as the address on the address bus is stable in case of an access. This is justified by the fact that all peripheral chips used are INTEL products, designed specifically for applications in 8080 microprocessor systems.

Chapter 5

Software Design

5.1 Introduction

For the ease of design and understanding, the OS has been partitioned into four major parts so that step by step development of software can be carried out. In order to avoid unnecessary confusions during the course of the design, it is also essential that the data structure of the system is clearly defined. In this chapter, the data structure of the system is first presented and this is followed by descriptions of each of the four major parts of the OS. Appropriate flowcharts are inserted wherever necessary to aid in understanding the flow of control and transfer of data. The software listing is not appended to this thesis, it is available in the operating manual specially written for this instrument.

5.2 Data Structure

The nine bit signed two's complement moments produced from each sample are accumulated for 128 samples before the partial accumulations are fed to the computer. For a maximum of 10^4 samples each of the result registers in the computer requires a length of 23 bits to guard against an overflow. Four three-byte registers are assigned for this purpose. At the end of a measurement, each accumulated moment is normalized to a two-byte fraction by dividing by the number of samples. This fraction is further converted into a four digit BCD (Binary Coded Decimal) fraction suitable for display. The standard deviation and r.m.s. values are similarly treated. Since four BCD digits can be stored in two-bytes of momory, a total of twelve two-byte registers are required. Another four two-byte registers are also reserved for the number of samples and the sampling rate, which allow a range of 0 to 9999 for both and for the period, two four-byte registers are used to store the product and its BCD equivalence. The sign associated with the first and third moment is isolated and stored as corresponding display character in a one byte location before the moments are normalized.

The OS also required eight bytes of momory as a scratch pad during the conversion of the results from binary to BCD digits. This adds up to a total of 62 bytes of RAM, excluding those used by the display interrupt service routine. A register map is shown in table 5.1. Twenty-two subroutines are called by the OS, a brief description of them is given below.

BCDBIN[17] : converts a two-byte (four digits) BCD number pointed by (DE) into a two-byte binary number in (HL). It is based on a procedure that isolates each digit and performs nested multiplication in binary arithmetic to product the binary equivalence in (HL).

BINBCD1 : converts a two-byte binary number pointed by (DE) into a two-byte (four digits) BCD number in locations pointed by (HL). A shift and decimal adjsut method[18] is used and no overflow will occur as the BCD number is limited to 9999.

Table 5.1 REGISTER MAP OF INSTRUMENT

ADDRESS	PURPOSE
3C1Ø - 3C17	Scratch Pad
3DØØ - 3DØ2	First Moment
3DØ3 - 3DØ5	Second Moment
3DØ6 - 3DØ8	Third Moment
3DØ9 - 3DØB	Fourth Moment
3DØC - 3DØD	Normalized First Moment
3DØE - 3DØF	BCD Equivalence Moment
3D1Ø - 3D11	Normalized Second Moment
3D12 - 3D13	BCD Equivalence Moment
3D14 - 3D15	Normalized Third Moment
3D16 - 3D17	BCD Equivalence Moment
3D18 - 3D19	Normalized Fourth Moment
3D1A - 3D1B	BCD Equivalence Moment
3D1C	Sign of Odd Moments
3D1D	Flag for Strobing Last Part of Partial Accumulations
3D2Ø - 3D21	Sampling Rate
3D22 - 3D23	BCD Equivalence Sampling Rate
3D24 - 3D25	Number of Samples
3D26 - 3D27	BCD Equivalence Number of Samples
3D3Ø - 3D33	Period
3D34 - 3D37	BCD Equivalence Period
3D3A - 3D3B	Standard Deviation
3D3C - 3D3D	BCD Equivalence Standard Deviation
3D3E - 3D3F	R.M.S.
3D4Ø - 3D41	BCD Equivalence R.M.S.

BINBCD2 : performs the same operation as in BINBCD1 except that a four-byte binary number is converted into a four-byte (eight digits) BCD number.

BLANK : blanks the display by filling the display buffer with the blank character FF.

BUMP : takes a display character in (A) and displaces four successive display characters in the display buffer pointed by (HL) by one character to the left. The leftmost character is lost. For example, if the display buffer contains {5F49} and (A) contains {7}, then after this subroutine is called the contents of the display buffer become {F497}.

BUMPD : takes a BCD digit in the lower order four bits of (E) and displaces the two-byte (four digits) BCD number pointed by (HL) by one digit. The procedure is similar to that of BUMP and the leftmost digit is dropped.

DBYT[5] : converts the two BCD digits in (A) into display characters and stores them in successive display buffer locations pointed by (DE).

DISB : displays a two-byte BCD number pointed by (HL) by converting the digits into display characters and storing them in successive display buffer locations pointed by (DE).

DISF1 : displays an odd moment with label in (A). The sign of the moment is also taken from location 3D1C and put before the displayed value. Pointer to the moment is in (HL).

DISF2 : displays an even moment with label in (A). Pointer to the moment is in (HL).

DIVID : divides a three-byte number pointed by (BC) by the two-byte number in (HL). The result produced by the non-restoring algorithm used is returned in (DE).

DRATE : displays the current sampling rate in the BCD sampling-rate register with a label.

DSAMP : displays the current value of the BCD number-of-samples register with a label.

FIX : first stores the sixteen bit fraction in (DE) into the register pointed by (BC) and then converts the fraction in (DE) into a four digit BCD fraction and stores it in the register pointed by (HL). The conversion is done by multiplying the fraction by 10000 and converting the integer part of the result into a four digit BCD number which is the desired BCD fraction.

KI[5] : gets a key entry from the Prompt 80 keyboard, debounces it and converts it into a key position number in (E) and the corresponding display character in (A).

MOMENT : transfers the accumulated moment pointed by (BC) to a scratch pad pointed by (DE) and divides it by the value in (HL). The result is returned in (DE).

MULTI : multiplies two two-byte numbers in (BC) and (DE). The four-byte result is returned in (DE) through (HL).

NSC : sets the number-of-samples counter with the value in the binary number-of-samples register.

ROT : transfers a BCD digit in the lower half of (E) into a byte pointed by (HL) and displaces one digit out. The digit displaced is held in the lower half of (B).

SPLIT : splits the content of the byte pointed by (DE) into two bytes and stores in scratch pad pointed by (BC).

SQRT[19] : calcualtes the square root of the number in (HL) and returns the result in (BC). The algorithm used is the common double shift procedure.

SRC : sets the sampling-rate counter with the value in the binary sampling-rate register.

5.3 Initialization Phase

This part of the OS is only executed when control is first transferred from the Prompt 80 monitor. It starts with disabling interrupt and sets the hardware control PPI to mode 0 with ports A and C as outputs and port B as input. The stack pointer is then defined and all hardware is reset. Counting modes for the three counters in the PIT are set and the counters are loaded with their default values. Counter 0 acts as the sampling-rate counter and is loaded with the 3 μs default rate. Counter 1 is set to interrupt at terminal count and is loaded with the maximum count of 10^4 for a one-cycle measurement. Counter 2 is fixed with a rate count of 128 for setting the sample-ready latch. All registers are cleared and the display is filled with blank characters. Interrupt and display are finally enabled and the two leftmost LEDs at port E8 of PPI2 are lit to mark the end of this phase before control is passed to the next phase.

Three routines are called by the OS, one blanks the display and others are for setting the sampling-rate counter and the number-of-samples counter. The names of these routines are listed with the flowchart in figure 5.2.

5.4 Parameter Acquisition Phase

This is the part of the OS which requires most user interaction. A command loop keeps checking for a keyboard entry which causes the OS to perform one of the four functions - display and set new sampling rate, display and set number of samples, set to one cycle measurement or start a measurement. Included in each of the first two procedures is a data entry loop which allows decimal data entries to override any previously set values. All key entries are input via the KI routine of the Prompt 80 monitor.

When the Display and Set Sampling-Rate key is pressed, the OS decodes the key and branches to display the present value in the BCD sampling-rate register. This is done by calling the DBYT routine twice with the pointer in the DE registers pointing to buffer locations 3FF7 and 3FF9. After the display is done, the OS enters the data entry loop to accept decimal data. Each key entry is checked to exclude out-of-range data and each legal entry is used to update the contents of the BCD sampling-rate register and the display by one digit. The loop terminates when the Continue key is detected and the BCD value in the BCD sampling-rate register is converted into binary and stored in the binary sampling-rate register; this content is then used by the SRC routine to set the sampling-rate counter. The command loop is finally re-entered after the

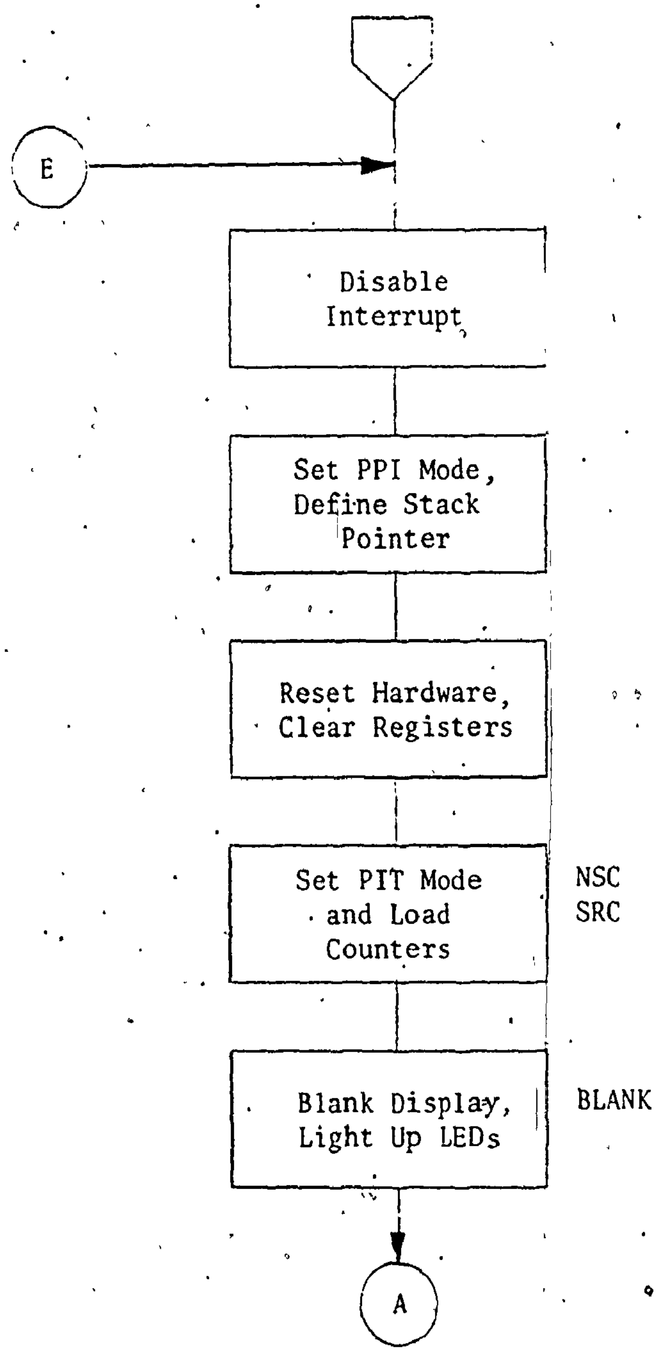

Figure 5.2 INITIALIZATION PHASE FLOWCHART

display is blanked. Similar actions are invoked when the Display and Set Number-of-Samples key is pressed.

The One Cycle Measurement key causes the BCD number of samples register to be displayed after it has been set to zero. This verifies that the command has been executed and pressing the Continue Key again causes the display to be blanked and a branch back to the command loop. The only exit from the command loop is by way of pressing the Start Measurement key which initiates the OS to set all necessary software and hardware logics for a measurement.

The number-of-samples counter is first checked to identify the mode of measurement and the corresponding hardware control line is set through the control PPI. Since the sampling rate has to be at least 3 μs, it is also checked and a branch back to the Display and Set Sampling-Rate loop occurs if an illegal value is found. The flag for strobing in the last part of the partial accumulations is cleared and the display is disabled before the Start/Stop line at the control PPI is set. Finally, all the LEDs at port E8 of PPI2 are lit to indicate that control has been passed to the measurement phase.

Nine routines are called in this segment of the OS. These include the three that have been described in the previous section and the KI routine for keyboard entry. Others are for displaying the counter registers and altering the registers and the display after a decimal key entry. A BCD to binary conversion routine is also required for converting the value in a BCD counter register to binary and store in the binary register before the corresponding counter is set. The flowcharts for the command loop and the data entry loop are shown in figures 5.3 and 5.4. New definitions for the keyboard are shown in table 5.5.

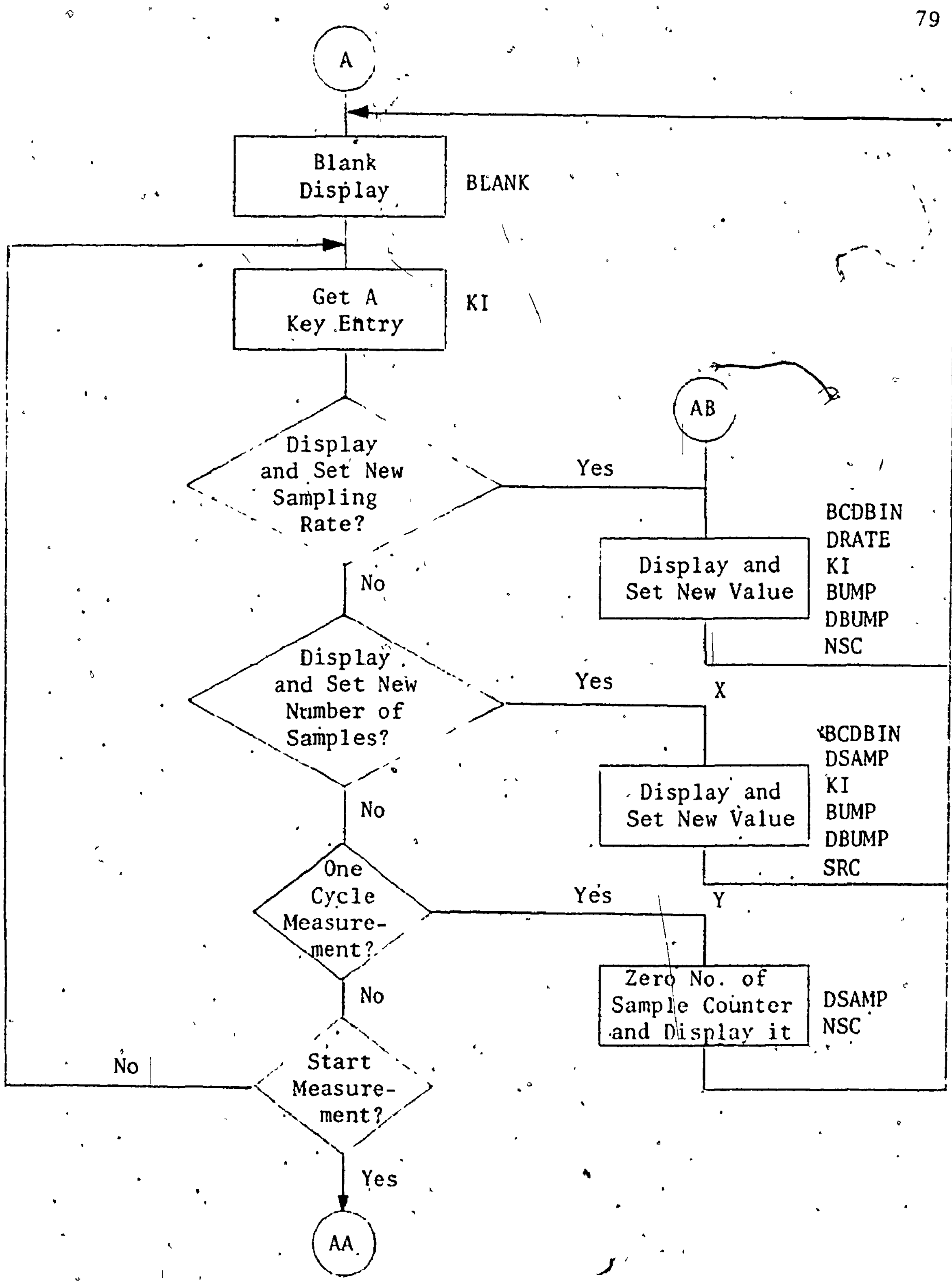

Figure 5.3 COMMAND LOOP FLOWCHART

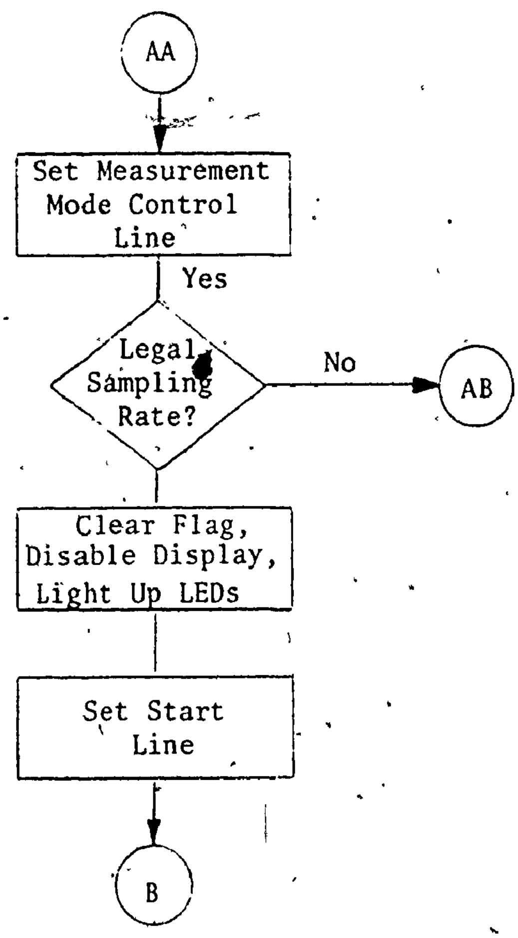

Figure 5.3 COMMAND LOOP FLOWCHART (Cont'd.)

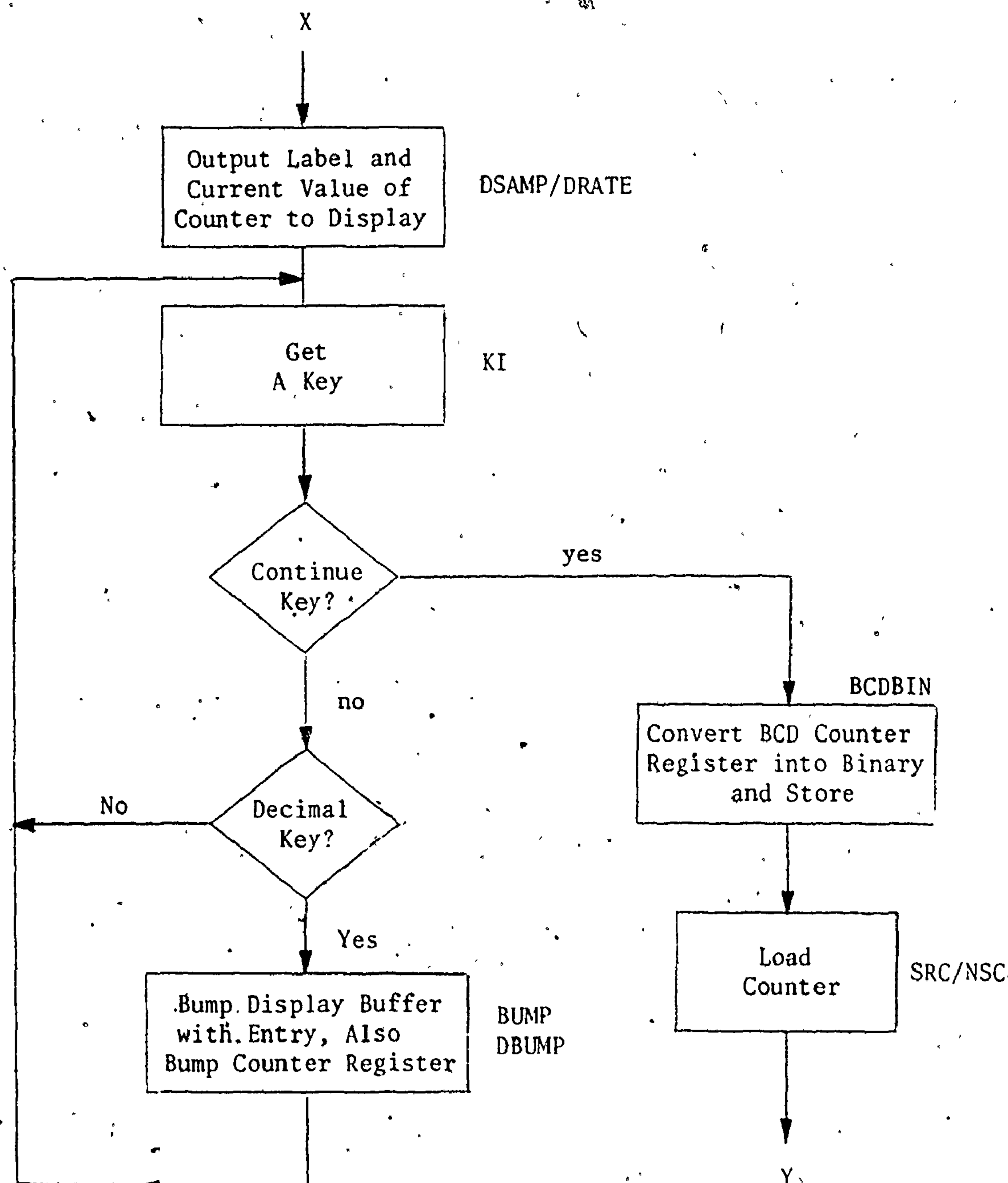

Figure 5.4 DATA ENTRY LOOP FLOWCHART

Table 5.5 KEY DEFINITIONS FOR INSTRUMENT

PROMPT 80 ORIGINAL	RECOGNIZED BY OS
0	Decimal 0
1	Decimal 1
2	Decimal 2
3	Decimal 3
4	Decimal 4
5	Decimal 5
6	Decimal 6
7	Decimal 7
8	Decimal 8
9	Decimal 9
A	Display First Moment
B	Display Second Moment
C	Display Third Moment
D	Display Fourth Moment
E	Display Standard Deviation
F	Display R.M.S.
SCROLL	Jump Back to Parameter Acquisition Loop
SINGLE STEP	One Cycle Measurement
EXAM. REG.	Display and Set Sampling Rate
GO	System Reset
DISPLAY MEM.	Display and Set No. of Samples
PREVIOUS	Display Period
NEXT	Continue
EXECUTE	Start Measurement

5.5 Measurement Phase

After the Start/Stop line is set, the starting addresses of the partial accumulation bus latches and the result registers are set up in two of the sixteen bit register pairs in the CPU. Other necessary constants are also stored in the CPU registers before the OS enters a loop which keeps checking the sample ready latch. It exists the loop when the sample ready signal is detected, disables interrupt and resets the sample ready latch before picking up the partial accumulations. After the partial accumulations are processed, interrupt is enabled and a branch back to the checking loop occurs. This sequence is repeated until the OS is interrupted by the hardware when the measurement has terminated or an overflow in number of samples has occurred.

Upon an interrupt, program control is vectored to address ØØ38 where a branch to the Prompt 80 central interrupt service routine is executed. At the beginning of this routine a jump instruction to location Ø7B9 has been inserted. This causes a block of nine additional instructions to be executed before control is passed back to the central interrupt service routine. The purpose of this is to check the end of measurement line at port FD of the control PPI to see if the interrupt is originating from the instrument. A jump to the instrument's interrupt service section occurs instead of branching back to the central interrupt service routine if this line is found set.

When the interrupt service section is entered, the Start/Stop line is reset to stop the hardware functions. The overflow flag is then checked to see if an overflow has occurred. A branch to the Prompt 80 error routine which causes the word 'ERROR' to be displayed is executed if it is found true. The sample ready flag is also checked to see if there is a set of accumulations just transferred to the bus latches and a return is executed to re-enter the partial accumulation pick-up loop to pick up this last set. Since the interrupt request latch of the instrument is not reset, another interrupt occurs when the enable interrupt instruction is executed at the end of loop. This time the sample ready latch is not set and the control passes to the next part which checks if the final accumulations in the multiplexed accumulator have been strobed to the bus latches and picked up. Location 3D1D is a flag for this purpose. It is reset before this phase is entered and indicates that this last part has not been attended to. The flag is set before the accumulations are strobed to the bus latches through the control PPI. A return is again executed to pick them up and at the end when interrupt occurs, the set flag detected causes the above procedures to be skipped.

After all the accumulations in the hardware have been picked up and processed, all hardware except the counters are reset with the reset line through the control PPI. If the measurement is for one cycle of the input signal, the number of samples is obtained by subtracting the count in the number of sample counter from 10^4 since the counter has been loaded with 10^4 before the measurement and works in a down-count mode. The result is stored in the number-of-samples register and also converted into BCD in the corresponding BCD register. This marks the end of the measurement phase and all processing thereafter has no interaction with the hardware except the display and keyboard assembly.

No subroutine is called during the execution of this phase except for the conversion of the number-of-samples to BCD so as to minimize the amount of time required by the software. The only complication is found in the multiple interrupt returns which are designed to avoid duplication of software required by different parts of the OS in the picking up of the partial accumulations from the bus latches. To aid in the understanding of this particualr structure, two flowcharts are provided in figures 5.6 and 5.7 which describe the partial accumulation pick-up loop and the interrupt service procedures.

5.5 Result Processing and Display Phase

This phase of the OS can be further sub-divided into two parts. One governs the generating of statistical results from the accumulated moments and formats them to be displayed. The other controls the display by monitoring keyboard commands and displays the requested results accordingly. Since there are a total of eleven possible commands, a branch table is used instead of a command loop for making branch decisions as the latter requires more memory space for implementation. Subroutines are used very extensively in both parts to avoid all possible software duplications and this makes it possible to pack all processing functions into 370 bytes of software.

The period is first calcualted by multiplying the sampling rate by the number of samples, both being sixteen bit binary numbers. A thirty-two bit binary number results and this is further converted into an eight digit BCD number suitable for display. Both numbers are then stored in corresponding period registers. Polarity of the odd moments is checked and a display character corresponding to a negative sign is

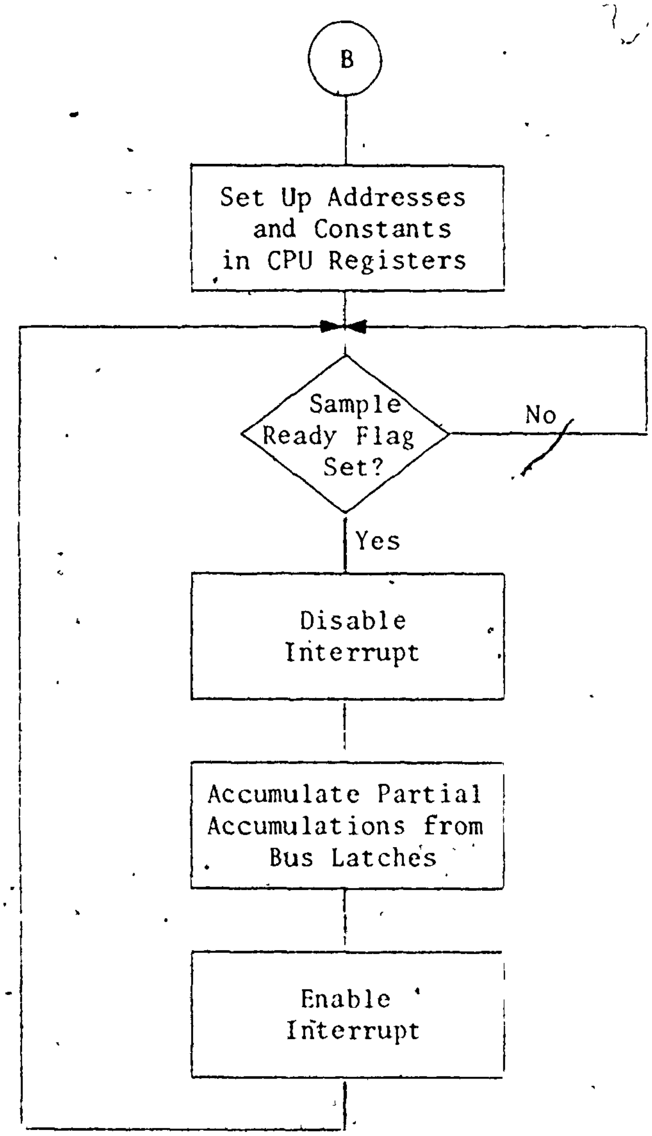

Figure 5.6 PARTIAL ACCUMULATION PICK-UP LOOP FLOWCHART

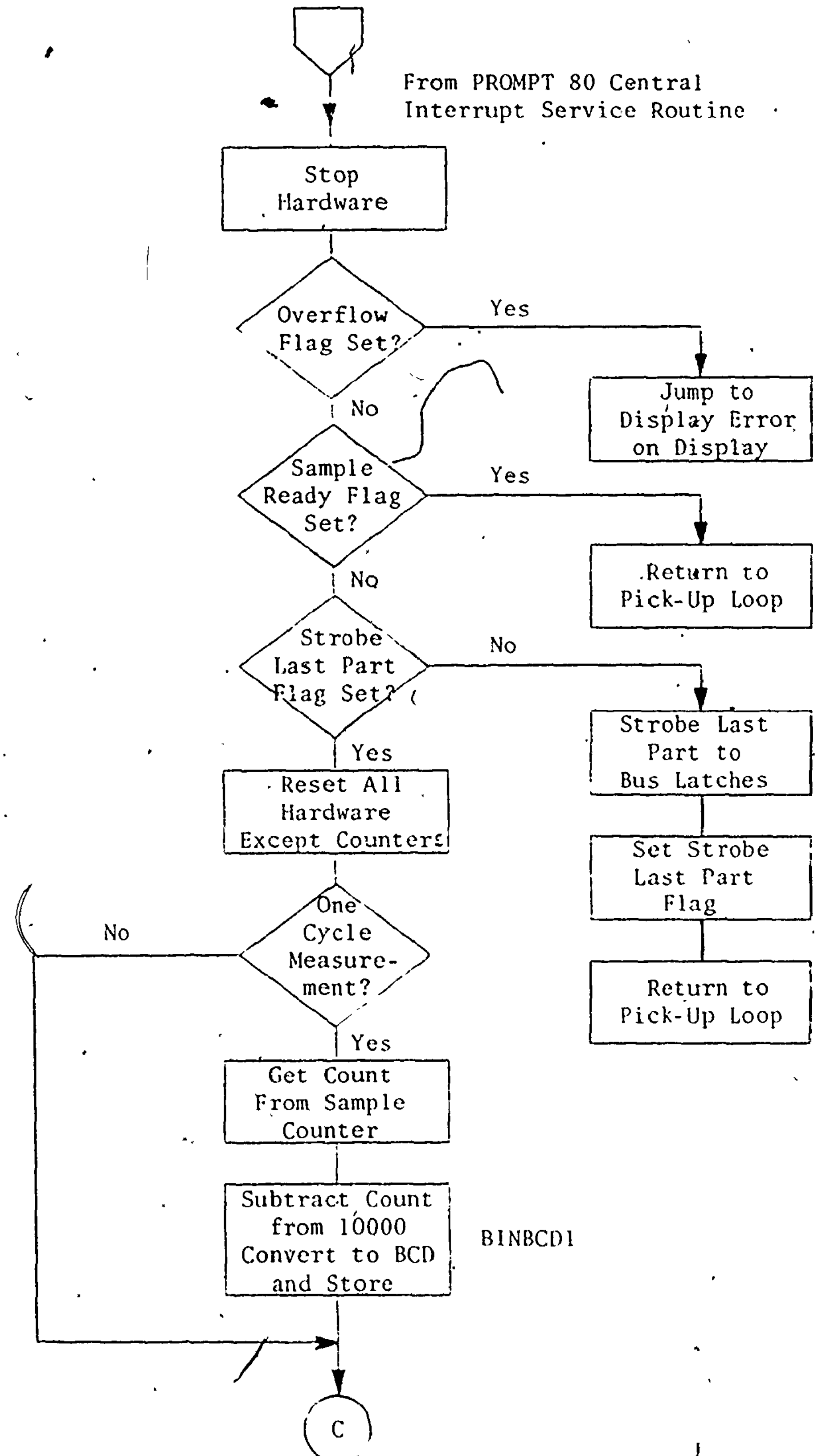

Figure 5.7 INTERRUPT SERVICE PROCEDURES FLOWCHART

stored in the sign register if they are found negative, otherwise this register is filled by a blank character. The four moments are normalized by dividing the absolute value of the accumulated moments by the number of samples using a 24 by 16 bit non-restoring division routine. A sixteen bit fraction results in each case and is converted into a four digit BCD fraction by first multiplying by 10^4 and then converting the integer part to a four digit BCD number. This number is the required fraction suitable for display and all resulting numbers are stored in corresponding registers.

From the normalized first and second moment, the standard deviation is obtained by square-rooting the difference between the second moment and the square of the first moment, in which case the squaring is done by self-multiplying and the square-rooting by the common double shift procedure. The root mean square is also calculated by square-rooting the second moment. Each process results in a sixteen fraction and is treated similarly as in the case of the normalized moments. A subroutine called FIX is written to do the fraction conversion and is shared by all six processes. After all the calculations and conversions are done, display and interrupt are enabled and the two right most LEDs at port E8 are lit to indicate that the results are ready to be displayed.

The second part begins with an unconditional branch to display the first moment with its label. When this is done, the keyboard is scanned for an entry which is then screened to exclude out-of-range commands. Upon a legal command, the key position value in the E register is used to obtain the location of a jump address in the jump table. This jump address is loaded into the HL registers and passed to the program counter to force an unconditional branch after the display is blanked.

Of the eleven possible branches, nine are for displaying the calculated results and the rest are used for exiting this final phase. One goes back to the parameter acquisition loop with the accumulated moments cleared so that another measurement can be started and the other jumps to the beginning of the initialization phase to reset the system completely.

Every one of the display functions branched to causes the desired value to be displayed in its appropriate format on the right hand seven segment LED display with a corresponding label on the left hand display. All statistical results are displayed as four digit BCD fractions and the rest are pure integers with eight digits for the period and four digits each for the sampling rate and the number of samples. The procedures followed by each display function are similar, requiring only the values in the corresponding BCD registers to be converted into display characters and transferred with a label to designated addresses in the display buffer. Two subroutines are designed for displaying the fractions; both include the turning on of a decimal point before the displayed number to indicate that it is a fraction and one also displays a minus sign before the odd moments. For the integer display functions, the Prompt 80 supplied routine DBYT is used in succession to cope with the multiple digits.

After a display function has been executed, the scanning of the keyboard resumes until another legal command is detected and causes a branch. This goes on until one of the exit commands is encountered or the system is switched off. The activities in this phase are summarized in the flowcharts in figures 5.8 and 5.9.

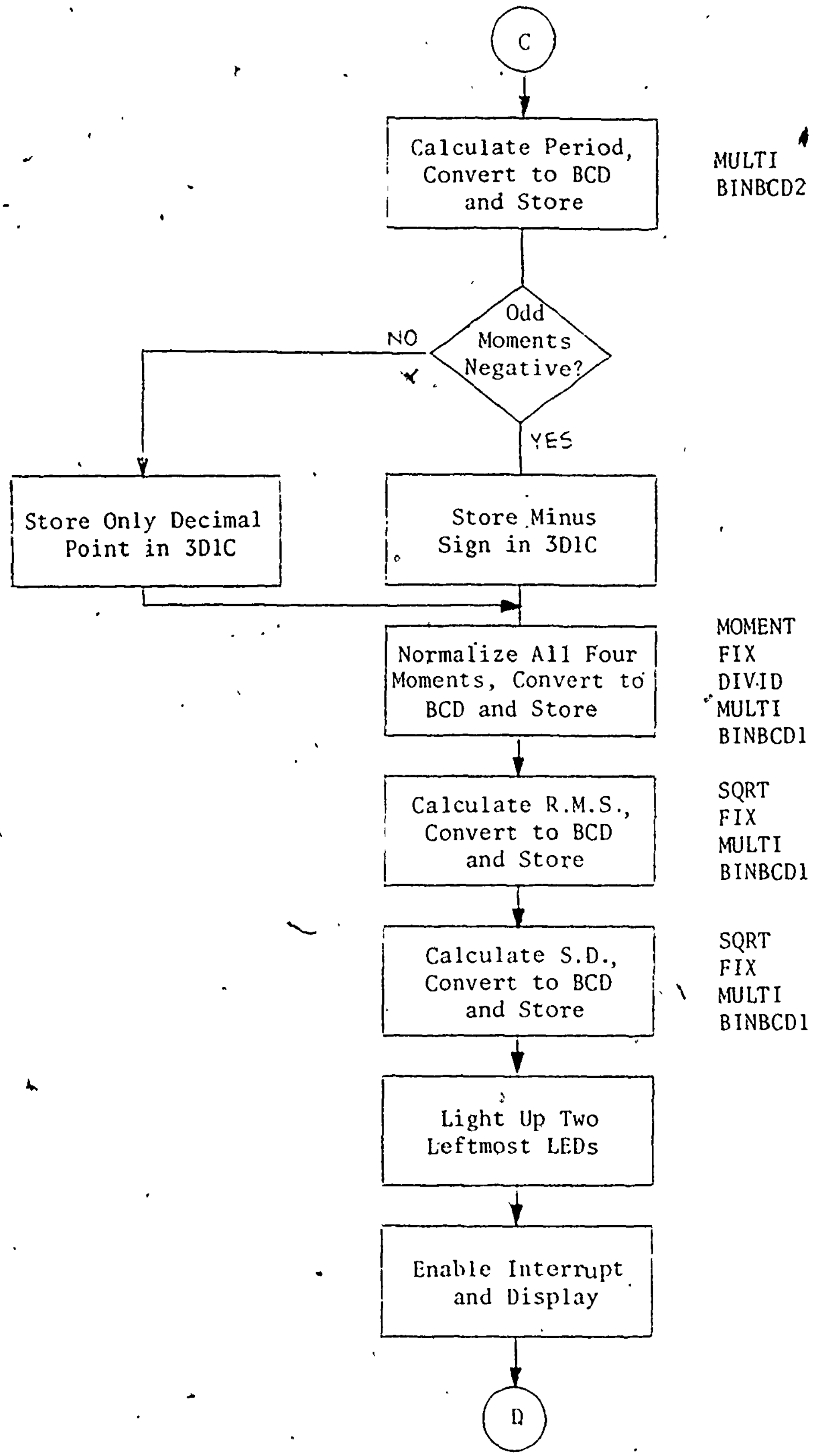

Figure 5.8. RESULT PROCESSING FLOWCHART

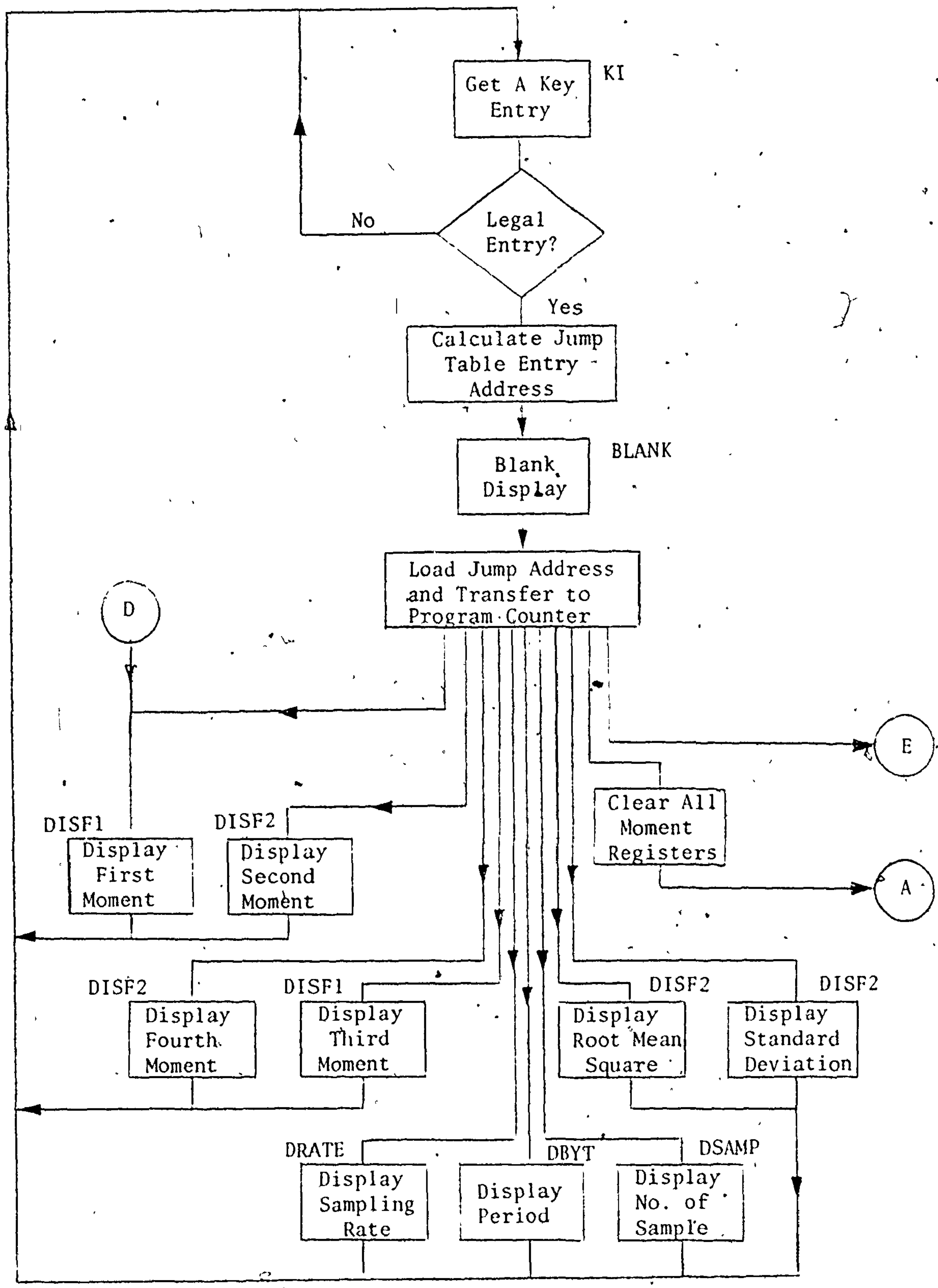

Figure 5.9 DISPLAY FUNCTIONS FLOWCHART

Chapter 6

Conclusions

6.1 Accuracy of the Instrument

The instrument implemented demonstrates accuracies comparable to all previous designs[2, 3, 4] in the measuring of all four moments of DC and AC test signals of various amplitudes. In all cases the error in the fourth moment is within 1% of full scale based on the average of five consecutive measurements with the maximum sampling rate of 3 μs. The moments in DC measurements are obtained for a sample size of 5000 and since even for indentical input signals, the last two digits of the first and second moments fluctuate slightly due to errors in the ADC, the averaging is necessary. AC measurements are conducted with both sine waves and triangular waves at a nominal frequency of 100 Hz so that a reasonably large sample size can be achieved. The test results are compared with the theoretical values in tables 6.1 to 6.3.

The readings from a Fluke 8100A[20] digital multimeter monitoring the input to the instrument are used as reference in the tables. During DC measurements, its DC range provides instant digital readouts of the DC input levels with which the variable power supply supplying the DC input signals can be adjusted to obtain the desired outputs. Similarly, the AC range of the multimeter is employed in AC measurements to produce r.m.s. readouts of the AC input signals from a signal generator except

Table 6.1 DC TEST RESULTS

(All values normalized to 10V)

DC Reading	M1		M2		M3		M4	
	Exact	Measured	Exact	Measured	Exact	Measured	Exact	Measured
0.1000	0.1000	0.0990	0.0100	0.0078	0.0010	0.0000	0.0001	0.0000
0.2000	0.2000	0.1998	0.0400	0.0358	0.0080	0.0045	0.0016	0.0000
0.3000	0.3000	0.3016	0.0900	0.0869	0.0270	0.0235	0.0081	0.0039
0.4000	0.4000	0.4022	0.1600	0.1563	0.0640	0.0587	0.0256	0.0196
0.5000	0.5000	0.4959	0.2500	0.2463	0.1250	0.1213	0.0625	0.0592
0.6000	0.6000	0.6031	0.3600	0.3587	0.2160	0.2142	0.1296	0.1244
0.7000	0.7000	0.7034	0.4900	0.4913	0.3430	0.3416	0.2401	0.2368
0.8000	0.8000	0.8051	0.6400	0.6432	0.5120	0.5108	0.4096	0.4053
0.9000	0.9000	0.9043	0.8100	0.8132	0.7290	0.7331	0.6561	0.6593

Table 6.2 SINE WAVE TEST RESULTS

(All values normalized to 10V)

AC Reading	P	$P^2/2$	Measured M2	$3P^4/8$	Measured M4
0.1404	0.1986	0.0197	0.0098	0.0006	0.0000
0.2878	0.4070	0.0828	0.0727	0.0103	0.0072
0.4283	0.6057	0.1834	0.1755	0.0505	0.0462
0.5759	0.8144	0.3316	0.3279	0.1650	0.1613
0.7102	1.0044	0.5044	0.5004	0.3816	0.3781

Table 6.3 TRIANGULAR WAVE TEST RESULTS

(All values normalized to 10V)

AC Reading	P	$P^2/3$	Measured M2	$P^4/5$	Measured M4
0.1142	0.2056	0.0141	0.0061	0.0004	0.0000
0.2286	0.4116	0.0565	0.0446	0.0057	0.0033
0.3397	0.6117	0.1247	0.1138	0.0280	0.0244
0.4532	0.8161	0.2220	0.2149	0.0887	0.0848
0.5601	1.0086	0.3391	0.3351	0.2069	0.2049

that the input amplitudes are roughly adjusted with the aid of an oscilloscope. Since the multimeter uses the average value of rectified AC signals to generate r.m.s. readings and is only calibrated for sine wave measurements, the readings obtained in measuring triangular waves have to be scaled down by a factor of 1.11 to give the average of the rectified triangular wave from which the peak and the r.m.s. values can be calculated. For a sine wave, the relations between the peak value P and the second and fourth moments are given by $P^2/2$ and $3P^4/8$ respectively. The corresponding values for a triangular wave are $P^2/3$ and $P^4/5$. From the readings on the AC range of multimeter, the peak values of the input sine or triangular wave can be calculated by using the fact that the average values of a rectified sine wave and a rectified triangular wave are $2P/\pi$ and $P/2$ respectively. Once the peak values are determined, the second and fourth moments can be obtained by the relations mentioned earlier. The theoretical values in tables 6.2, and 6.3 are all based on these calculations.

According to the specifications of the ADC82AG ADC, a total accuracy error of ± 1 LSB is possible at full scale, which means a maximum full scale error of $\pm 0.4\%$ for the first moment. This includes the effects of gain error, offset error, linearity error, differential linear error and power supply sensitivity. For the high order moments, the effect of truncating one bit of a nine bit two's complement number in the moment generator becomes very significant, especially when the sample values are small. This truncation adds 0.4% of uncertainty to the truncated number and since two such numbers are required to produce a higher order moment, an additional error of 0.8% at full scale has to be added for each increase in order of the moments from the first. Thus the maximum error in the fourth moment is 2.8% at full scale and this value only appears when all truncated bits in the above process are ones.

With the major sources of error in the instrument identified and the values of each individual error estimated, various changes can be made to improve its accuracy. These are presented as suggested further development in the next section.

6.2 Suggested Further Development

The instrument implemented may lack the desired accuracy in some applications. In situations where more accurate higher order moments are necessary, the error in these moments can be immediately halved if an eight bit unsigned multiplier is available and substituted into the moment generator. The truncation circuit can then be completely eliminated since the products from the multiplier are straight binary numbers and can be truncated by simply dropping the desired number of bits. A block diagram of the suggested new structure of the moment generator is shown in figure 6.4. The sign generator is for inserting a one at both lines MS and TØ during the times when the first and third moments are accumulated if the sample is negative.

Any further improvement in the overall accuracy of the instrument requires an increase in the word-length of all processes. The next logical step is to increase the quantization to twelve bits, which reduces the quantization uncertainty to ±0.012% of full scale. Error in the first moment is then dominated by the linearity error of the analog circuitry, which can be suppressed sufficiently by employing a better circuit or more carefully trimming of the resistors. A twelve bit multiplier has to be used to produce the higher order moments with the same truncation process

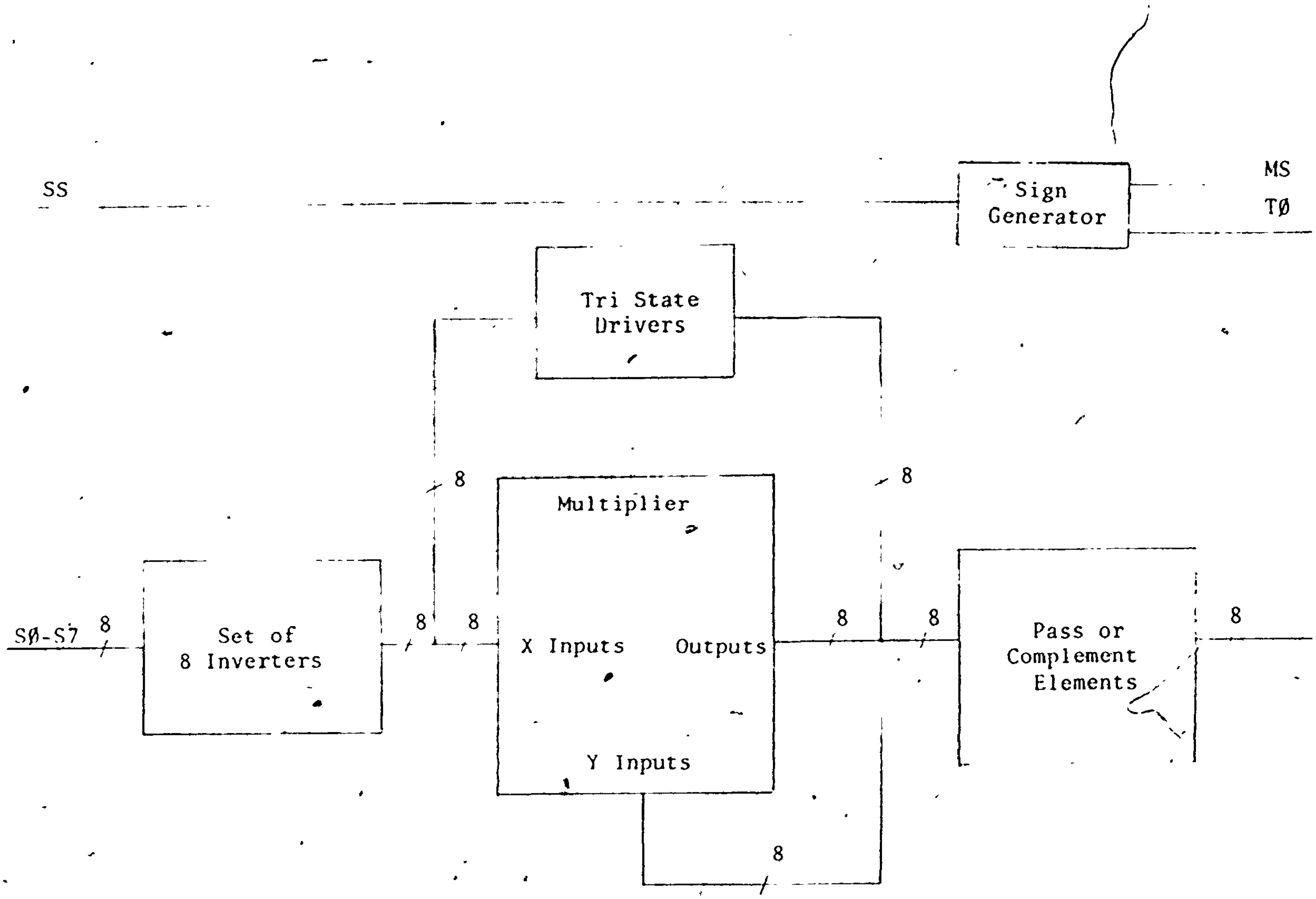

Figure 6.4 IMPROVED MOMENT GENERATOR

and the error in the fourth moment should be improved to within 1%. In the multiplexing accumulator, one additional tri-state latch is required by each moment as its timing structure does not allow an eight bit latch to be shared by two moments. Similar expansion in the bus latches is also necessary and thus if the adder in the multiplexing accumulator can be extended to 24 bits, more samples can be accumulated before the partial accumulations are latched to the bus latches to be picked up by the microprocessor. This releases the microprocessor from frequent processing of the partial accumulations and makes background processing of other tasks a possibility. The only software requirements here are a slightly more complicated interrupt structure in the operating system and corresponding increase in size of the storage registers.

In some applications, the sample size allowed by this instrument may be considered insufficient. This can be remedied by cascading another 8253 PIT to the one already in use in the instrument to double the order of magnitude of the capacities of the counters. Some software alterations are necessary for accommodating the increase in magnitude of the total samples accumulated and the expanded range of parameters that can be used to control a measurement. The potential duration of a measurement is also extended by such changes and the instrument may be used to cover a wider area of applications. If speed is a critical factor in the sampling process, the instrument can always be upgraded to have a sampling rate of 1 μs by using a faster ADC and tightening the timing of the processing circuit. To process one moment in 250 ns, the system clock has to be changed to 4 MHz and high speed Schottky TLL gates are necessary in parts of the timing and control circuit.

All future development suggested so far tends to imply that the architecture of the instruemnt is a flexible one and can be tailored to meet the requirements of a particular application. This is in fact one of the criteria based on which the instrument is designed. It also reflects the usefulness of a microprocessor in terms of its flexibility and intelligence but exposes its speed limitations so that a more efficient way of utilizing its power can be developed. The immediate future of intelligent digital instrumentation appears to be strongly dependent on microprocessors.

References

1. Deist, F., and Kitai, R.: "Digital Transfer Voltmeter: Principles and Error Characteristics", Proc. IEE, Vol. 110, No. 10, Oct. 1963, pp. 1887-1904.

2. Kitai, R., and Braithwaite, D. J.: "Digital Instrumentation of the Square of a Slow Fluctuating Voltage", IEEE Trans., IM-17, Sept. 1968, pp. 177-185.

3. Kitai, R., and Masuko, A.: "Digital Instrument for Measurement of Autocorrelation and Moments", Proc. IEE, Vol. 116, No. 11, Nov. 1969, pp. 1950-1956.

4. Majithia, J. C.: "Digital Moment Analyser: Design and Error Characteristics", Ph.D. Thesis, McMaster University, Hamilton, Ontario, Mar. 1971.

5. "Prompt 80 Microcomputer User's Manual", INTEL Corp., Santa Clara, Cal., U.S.A., Sept. 1976.

6. "SBC 80/10 Single Board Computer Hardware Reference Manual", INTEL Corp., Santa Clara, Cal., U.S.A., 1976.

7. "8080 Microcomputer System User's Manual", INTEL Corp., Santa Clara, Cal., U.S.A., Sept. 1975.

8. "8080 Assembly Language Programming Manual", INTEL Corp., Santa Clara, Cal., U.S.A., 1976.

9. "LSI Multipliers", Specification Notes, TRW Inc., Redondo Beach, Cal., U.S.A., Mar. 1977.

10. "MCS-85 Microcomputer User's Manual", INTEL Corp., Santa Clara, Cal., U.S.A., June 1977.

11. "INTEL 1977 Data Catalog", INTEL Corp., Santa Clara, Cal., U.S.A., 1977, pp. 12-60.

12. "System 80/10 Microcomputer Hardware Reference Manual", INTEL Corp., Santa Clara, Cal., U.S.A., 1976.

13. "Models 3505J and 3507J Fast Slewing Operational Amplifiers", Specification Notes, Burr Brown Corp., Tucson, Arizona, U.S.A., April 1974.

14. Graeme, J. G.: "Applications of Operational Amplifiers: Third Generation Techniques", Burr Brown Electronics Series, McGraw Hill, U.S.A., 1973, pp. 123-124.

15. "ADC82 IC A/D Converters", Specification Notes, Burr Brown Corp., Tucsan, Arizona, U.S.A., Oct. 1976.

16. "Dual Power Supply for Computer Interface: Model 546", Specification Notes, Burr Brown Corp., Tucson, Arizona, U.S.A., Nov. 1971.

17. "BCD to BIN Conversion Routine", Ref. No. BB11, Section 5, Microcomputer User's Library, INTEL Corp., Santa Clara, Cal., U.S.A., April 1977.

18. Barna, A., and Porat, D. I.: "Integrated Circuits in Digital Electronics", John Wiley & Sons, U.S.A., 1973, pp. 287-297.

19. "16 Bit Square Root Routine", Ref. No. BC11, Section 5, Microcomputer User's Library, INTEL Corp., Santa Clara, Cal., U.S.A., April 1977.

20. "Model 8100A Digital Multimeter", Instruction Manual, John Fluke Mfg. Co. Inc., Seattle, Wash., U.S.A., Oct. 1969.