MICROPROCESSOR BASED DIGITAL CONTROL OF NON-DISPERSIVE INFRARED SPECTROPHOTOMETER

(A SYSTEM DESIGN APPROACH)

A DISSERTATION

Submitted in partial fulfilment
of the requirements for *the* award of the Degree
of
MASTER OF ENGINEERING
in
ELECTRICAL ENGINEERING
(System Engineering & Operational Research)

By
ATAM P. DHAWAN

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE, (INDIA)
MAY, 1979

CERTIFICATE

Certified that the thesis entitled 'Microprocessor Based Digital Control of Non-Dispersive Infrared Spectrophotometer' which is being submitted by Mr. Atam P. Dhawan in the partial fulfilment of the requirements for the award of the Degree of Master of Engineering in Electrical Engineering, in System Engineering and Operational Research Group, of the University of Roorkee, Roorkee, India, is a record of student's own work carried out by him under my supervision and guidance. The matter embodied in this thesis, has not been submitted for the award of any other degree.

This is further to certify that he has worked for a period of about four and a half months (from January 1979 to May 1979) at this University for preparing this thesis.

M Vasantha

Roorkee
Dated : 24-5-'79

(M.K. Vasantha)
Reader,
Electrical Engg.
Department,
University of Roorkee,
Roorkee (India)

ACKNOWLEDGEMENT

Author wishes to express his profound gratitude and sincere thanks to his guide Sri M.K. Vasantha, Reader in Electrical Engineering Deptt., University of Roorkee, Roorkee for his erudite guidance, wise suggestions and encouragement, the author received from him.

A warm expression of gratitude is due towards all who took interest in completion of this dissertation work.

Atam P. Dhawan

(ATAM P. DHAWAN)

CONTENTS

CONTENTS

CHAPTER -I

INTRODUCTION

In the past decade computer technology and design (both analog and digital) and the development of low cost linear and digital 'integrated circuitry' have advanced at an almost unbelievable rate. Thus computers and quantitative electronic circuitry are now readily available to chemists, physicists, and other scientific groups interested in instrument design.

"The computer and integrated circuitry are revolutionizing measurement and instrumentation in science".

In general, chemist, physicists and others have just begun to realize and understand the potential of computer applications to their respective research and quantitative measurement. The basic applications are in the areas of data acquisition and reduction, simulation and instrumentation (on-line data processing and experimental control and/or optimization in real time).

For years, while much of the spectroscopic instrumentation market was ignoring the inroads being made into the laboratory by minicomputers and now by microprocessors, individual researchers were deeply involved in the making of laboratory 'workhorse' instruments and powerful laboratory computers. Infrared opectrophotometry has long had the reputation of being only semiquantitative at best. For this reason, some of the attention being lavished on the recently developed Fourier transform infrared instrumentation is derived from long-held misconceptions regarding conventional dispersive instrumentation. The dispersive instrument manufactures are moving slowly in the world of today, an era of sophisticated, computer based data acquisition and reduction. As they begin redesigning their 10-year-old dispersive/non-dispersive instrumentation with modern, high speed detectors and associate them with low cost, small computers and now by microprocessors based systems, the field will enjoy a second childhood. The huge advantage of microprocessor aided instrument is that these procedures can be absolutely standardized and are then not subject to error or variation from operator to operation. Another advantage is the uniformity of reporting format, usually taking the form of an alphanumeric printout under control of the microprocessor.

This scheme shows that an infrared absorption spectrometer, such as non-dispersive infrared (NDIR) spectrometer widely applied in the chemical industries and in air pollution control can be successfully married to a microprocessor revealing a lot of advantages being NDIR digital control system.

ABSORPTION SPECTROMETER

Radiation passed through semitransparent substances in characteristically absorbed selectively, the amount of energy transmitted being a function of the wavelength as well as the depth of the optical path through the sample and the concentration of the absorbing material (Beer's Law). Infrared spectra are generally unique and characteristic of particular organic and inorganic compounds as a result of the interaction of the infrared radiation with the molecular structure, exciting vibration and bending modes of motion and accounting for the energy absorption.

Absorption spectrometers fall into two categories dispensive and non-dispensive. The former type spreads out radiation from a source in space, using a prism or grating and selects a narrow wavelength by passing the dispersed radiation through a slit (this assembly being termed a monochrometer). The resulting beam is passed through a cell containing (the sample and the transmitted radiation is measured by a photoelectric detector. As the monochrometer setting is swept through a range of wavelengths, the spectrum can be plotted on a recorder. In the non-dispersive type, all the radiation, not just a small pass-band is passed through the sample cell. The transmitted energy is detected and then the same amount of source radiation is passed through a reference cell (usually containing a non-absorbing gas such as nitrogen) ; the two signals are then compared.

Since the NDIR type utilizes all the energy from the source which covers a broad band of wavelengths, it consequently generates a largest signal and is more sensitive.

The addition of digital data acquisition and reduction hardware to an inherently irreproducible instrument may seen like a foolish preposition at first. However, some of the simplest applications of digital data handling can significantly improve the quality of information from even the least expensive infrared instrumentation. The capability to spectrum-average is a valuable technique for improving the signal noise (S/N) ratio in regions of low transmittance. Averaging n spectra together to form one spectrum, provided that the response time of the detector is taken into account, produces an improvement in S/N equal to the square root of n. This technique has been employed to great advantage in Fourier transform infrared spectroscopy, where averages of 100 scans are often employed in order to inverse the S/N on a high resolution spectrum. Additional advantages are available to the spectroscopist with some kind of computing capability, such as slit function deconvolution mathematical smoothing, and separation of overlapping peaks.

It is always possible that some infrared spectrscopists will contend that the inherent properties of complex infrared spectra and some samples will make the acquisition of digital data a waste of time. Strongly absorbing samples, which require reference beam matching or severe attenuation to produce a satisfactory spectrum, suffer from the inability of many instruments to yield identical results from one-day or even hour, to the next. The drawing of base-lines is a difficult task for even the most experienced spectroscopist, when overlapping bands must be used in a quantitative determination. Few infrared spectroscopists will trust the drawing of base-lines to their computer, even though the computer's baseline efforts are much more predictable than most of a group of spectroscopists. Complex spectra, such as those of synthetic polymers, do not lend themselves to quantitative analysis of such things as plasticizers, filters etc., except where the substrate spectrum has no interfering absorption bands. With such complex spectra, though, it is often possible to obtain a reference sample which does not contain the analyte . The spectrum of the reference material can then be stored as a baseline to be subtracted from spectra of samples containing the analyte. Using a computer, some fraction of this baseline can be subtracted from the spectrum of a subsequent sample. In this manner, the complex baseline is completely removed from consideration, and the spectroscopist can then focus on the difference spectrum for this analysis.

The aim of this work is to present a suitable and adequate hardware for the interface of non dispensive infrared spectrometer to a microprocessor based digital control system and required software to overcome all the difficulties of spectroscopy analysis.

CHAPTER - II

INFRARED SPECTROSCOPY

The ordinary infrared region extends from 2.5 to 15 μ (4000 to 667 cm^{-1}); the region from 0.8 to 2.5 μ (12,500 to 4000 cm^{-1}) is called the near-infrared and the region from 15 to 200 μ(667 to 50 cm^{-1}) is called infrared. The wave number is directly proportional to the absorbed energy ($K = E/hc$), whereas the wavelength is inversely proportional to the absorbed energy ($\lambda = hc/E$; $\lambda = 1/K$).

A molecule is not a rigid assemblage of atoms. A molecule can be said to resemble a system of balls of varying masses, corresponding to the chemical bonds of a molecule. There are two kinds fundamental vibrations for molecules ; stretching in which the distance between two atoms increases or decreases, but the atom remains in the same bond axis, and bending (or deformation), in which the position of the atom changes relative to the original bond axis. The various stretching and bending of a bond occur at certain quantized frequencies. When infrared light of the same frequency is incident on the molecule, energy is absorbed and the amplitude of that vibration is increased. When the molecule reverts from the excited state to the original ground-state, the absorbed energy is released as heat.

A non-linear molecule that contains n atoms has

(3n-6) possible fundamental vibrational modes that can be responsible for the absorption of infrared light. In order for a particular vibration to result in the absorption of infrared energy, that vibration must cause a change in the dipole moment of the molecule. Thus molecule that contain certain symmetry elements will display somewhat simplified spectra. The C=C stretching vibration of ethylene and the symmetrical C-H stretching of the four C-H bonds of methane do not result in an absorption band in the infrared region. The predicted number of peaks will not be observed also, if the vibrations result in absorption that are so close that they can not be resolved, or if the absorption is of very weak intensity.

Additional (non-fundamental) absorption bands may occur because of the presence of overtones (or harmonics) that occur with greatly reduced intensity, at ($\frac{1}{2}$, $\frac{1}{3}$,....) of the wavelength (twice, three times....the wave number), combination bands (the sum of two or more different wave nos.) and difference bands (the difference of two or more different wave nos.).

An approximate value for the stretching frequency (ν in cm^{-1}) of a bond is related to the masses of two atoms (M_x and M_y in grams) the velocity of light (C), and the force constant of the bond (K, in dynes/cm)

$$\nu = \frac{1}{2\pi C}\sqrt{\frac{K}{M_x M_y/(M_x+M_y)}}$$

Single, double, and triple bonds have force constants that are approximately 5, 10 and 15x10^5 dynes/cm respectively.

The magnitude of the molar extinction coefficient in infrared-spectroscopy varies from near zero to around 2000. The value is proportional to the square of the change in the dipole moment of the molecule that the particular vibration causes. Absorption peaks caused by stretching vibrations are usually the most intense peaks in the spectrum.

MECHANICS OF MEASUREMENT

Infrared absorption spectra are usually obtained by placing the sample in one beam of double-beam infrared spectrophotometer and measuring the relative intensity of transmitted (and therefore absorbed) light energy versus wavelength (or wave number). A common light source for infrared radiation is the Nernst glower, a molded rod containing a mixture of zirconium oxide, yttrium oxide and erbium oxide that is heated to around 1500° by electrical means. Either optical prisms or gratings are used to obtain approximately monochromatic light; grating spectrophotometers give higher resolution. Glass and quartz absorb strongly throughout most of the infrared region, so they can not be used as cell containers or as optical prisms. Metal halides (e.g. sodium chloride) are commonly

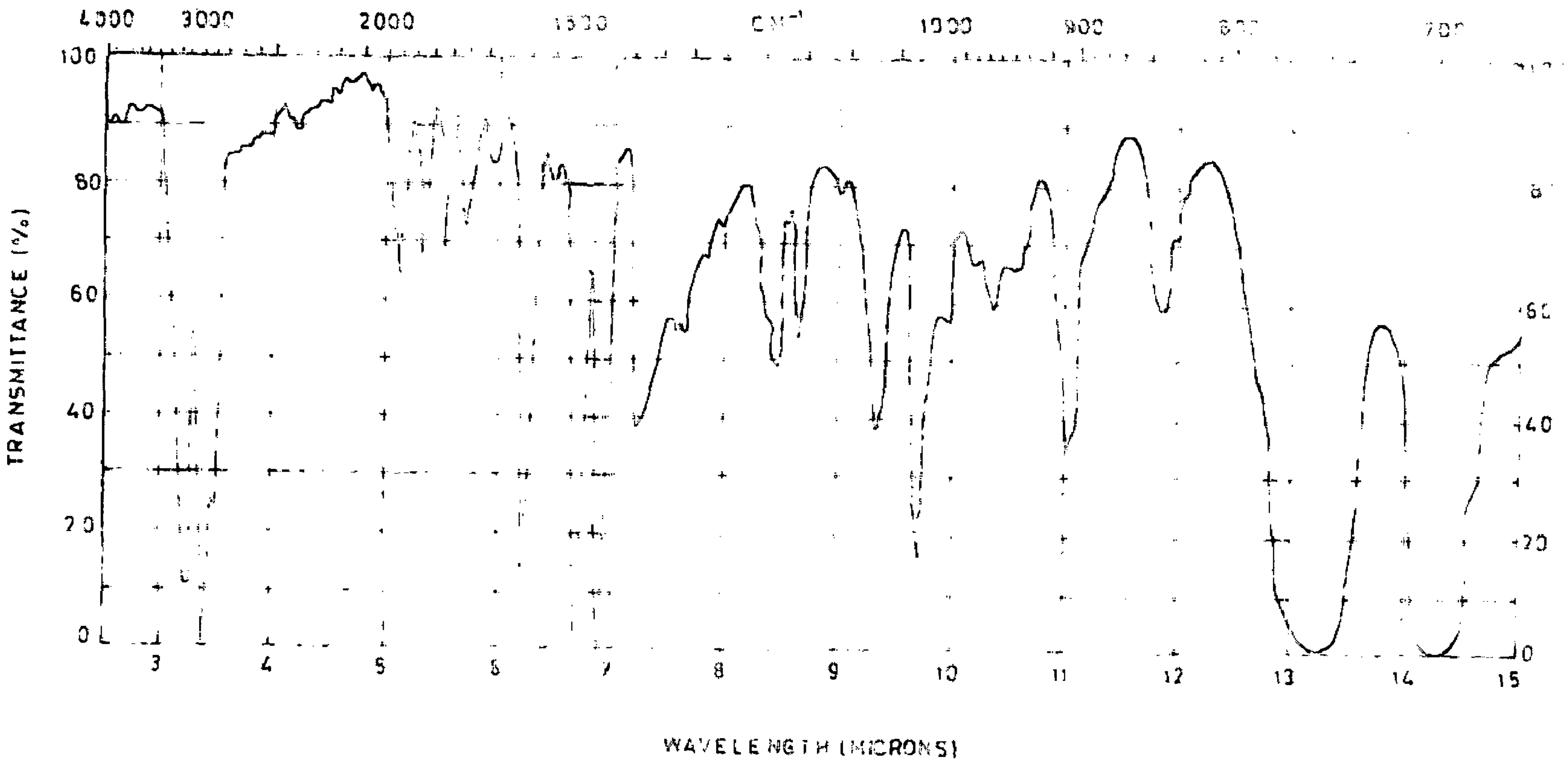


FIG. 1 POLYSTYRENE FILM NDIR SPECTRUM

used for these purposes. Recording spectrophotometers are available such that a complete spectrum (2.5 - 25 μ 4000 - 400 cm^{-1}) may be obtained in a matter of minutes.

When the spectrum is determined, a caliberation line is usually recorded on the paper. This is necessary because the recorder paper is fitted on a drum of the ordinary spectrophotometer, and it is not possible to place the paper in exactly the same position every time. One of several absorption peaks of polystyrene (Fig. 1) is commonly used for this purpose 3.509 μ (2850 cm^{-1}), 6.238 μ (1603 cm^{-1}) or 11.035 μ (906 cm^{-1}).

The spectrum may be determined if the sample is a gas, a solid, a liquid or in solution. The sample should be dry, because water absorbs strongly near 2.7 μ (3710 cm^{-1}) and near 6.15 μ (1630 cm^{-1}). These absorptions may obscure absorptions of substance being analyzed or may, more frequently, lead to erroneous assignments. The most commonly used cells are constructed by using sodium chloride windoes. Cell for the determination of the spectra of gaseous samples are available with path lengths up to several meters.

APPLICATION OF INFRARED SPECTROSCOPY

With an adequate number of examples of secure precedence, the chemist will usually know very closely what absorption position to expect from a certain functional obtained, there are a number of questions that can be answered fairly rapidly about what it contains ? and what is the substitutional type? . Answers to questions such as these will give many clues for chemical work that can lead to the conclusive identification of the compound. Thus infrared-spectroscopy has been proved of great importance in chemical process industries and air pollution control engineering.

group in a given environment. In the absence of steric or electrical effects that would affect the vibrational frequency of a given group, that group will absorb infrared energy of very nearly the same wavelength in all molecules. However, the infrared spectrum cannot commonly distinguish a pure sample from an impure sample. In general, however, the spectrum of a pure sample will have fairly sharp and well resolved absorption bands. The spectrum of a crude preparation that contains many different kinds of molecules will display broad and poorly resolved absorption bands because of the many absorptions that are present.

The examination of the infrared spectrum can aid a chemical investigation in many ways. The progress of most organic reactions can be followed readily by examining spectra of aliquots withdraws.

Perhaps the most powerful function of infrared spectroscopy is establishing conclusively the identity of two samples that have identical spectra when determined in the same medium. In order to define a functional group, the spectrum must be examined in detail for other diagnostic absorption bands and used in conjunction with classical chemical reactions and solubility determinations. If the spectrum does not contain an absorption typical of a certain functional group, the molecules does not contain that functional group. When the spectrum of an unknown material is

CHAPTER - III

INFRARED ABSORPTION SPECTROPHOTOMETER

We have discussed that when infrared radiation passed through a material, certain wavelengths are absorbed in proportion to the concentrations of the various compon ents in the material. Thus, evaluation of the type and the amount of energy absorbed provides a measure of the concentration of a component in a chemical mixture.

The process-stream analyzers fall into two categories. One type employs the dispersion of light into its component wavelengths, a prism generally being used to disperse the light. These instruments are known as 'dispersive' type.

The second approach does not employ the dispersion of light and is hence called 'non-dispersive' . The non - dispersive instruments are generally more suited to liquid systems than are the non-dispersive instruments though the latter can be used efficiently in some cases. The non-dispersive instruments, however, make use of an integrated total of the energy absorbed by a material from an infrared beam. This makes possible more signal for driving a recorder than is available when absorption at a single wavelength is used. The non-dispersive analyzer thus generally has a greater sensitivity.

NON-DISPERSIVE ANALYZERS

This type of instrument does not operate at any one specific wavelength but makes use of an integrated total of the energy absorbed by a sample from an infrared beam, to make possible more signal power to drive the recorder. Thus, the non-dispersive type of analyzer has a great increase in sensitivity and stability over the dispersive type of spectrophotometers, which is an important factor in plant-instrumentation. It is more suitable for gases and vapours.

Quantitative isolation of the desired component is obtained by sensitizing the analyzer to changes in the concentration of the desired component and greatly reducing or removing its responsiveness to changes in concentration of all other infrared absorbers present in the sample - stream. Although limitations may arise because of interference problems, combinations of stream components which would cause interference can often be predicted and thus can often be compensated for prior to installation. The source is a hot-wire filament or Globar element; the radiation is alternately passed through the sample or reference cell by means of a motor-driven optical shutter (chopper). The detector is a pneumatic cell filled with a pure sample of the gas to be detected - the sample component of interest (COI) - and equipped with a diaphragm electrically connected as a condenser microphone. If the sample cell in fact contains some of the COI, the radiant energy falling on the detector will fluctuate in synchronism with the chopper rotation, since some of the energy is absorbed passing through the sample cell but not through the reference cell. This fluctuation is converted by the microphone to an electrical signal. Since the detector (filled with COI) responds only to the radiation absorbed by the COI, it is relatively sensitive only to that compound, ensuring selectivity. The fluctuating signal varies in amplitude with the amount of COI present in the sample tube.

Although such a device is sensitive and selective, being capable of detecting many gases in the range of parts per million, it suffers from a number of shortcomings ; the one that concerns us here is that only a single COI can be detected per instrument. In many industrial and technological situations, it is desired to monitor several components of the gas stream simultaneously and more or less continuously. To overcome these defects and expand the sensitivity of the NDIR instrument a variation of the concept is employed. Instead of preumatic detector, a more sensitive and broad band solid-state detector is used. The rotating chopper and gas filled filter are combined; the filter wheel consists of pairs of these filters alternately interposed in the optical path of the infrared source, sample cell and detector. One is filled with COI and the other with nitrogen. When the COI filter is in

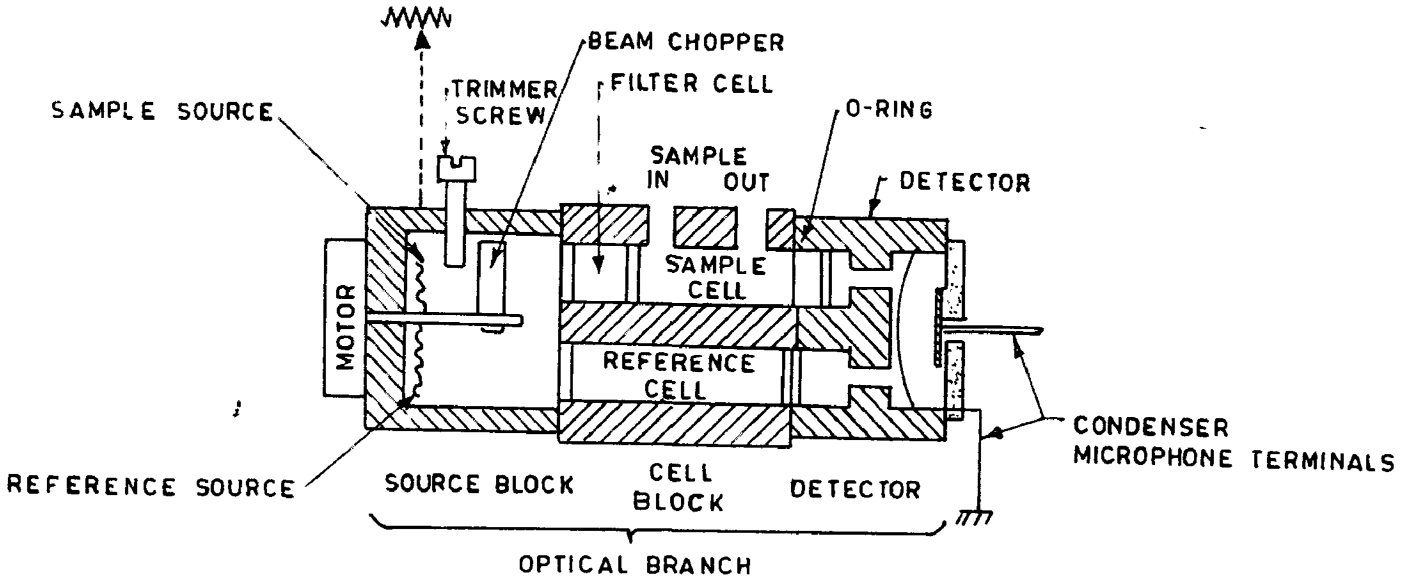


FIG. 2 CONVENTIONAL NDIR SPECTROMETER

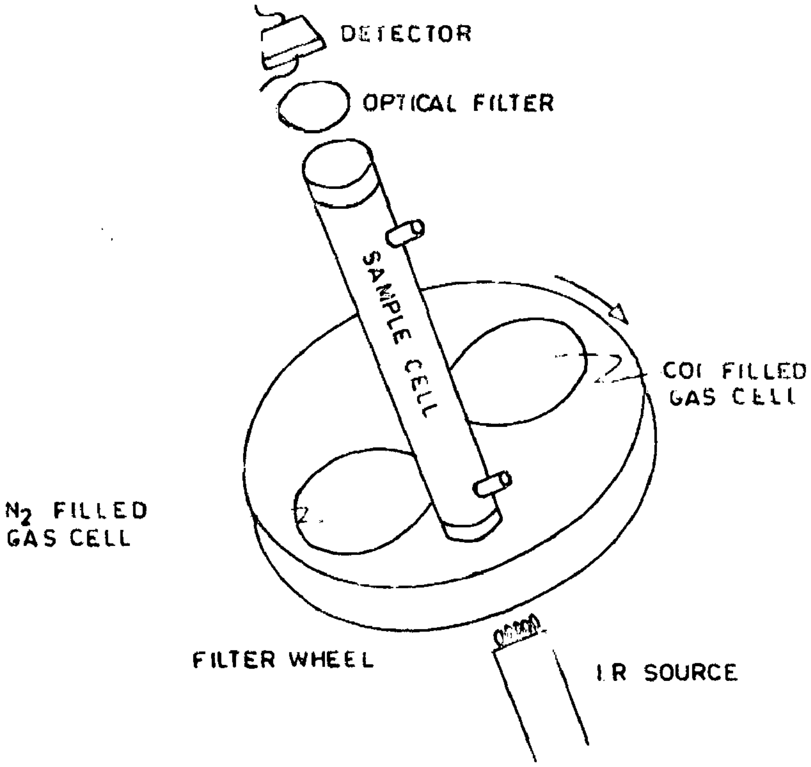


FIG. 3 FILTER WHEEL NDIR

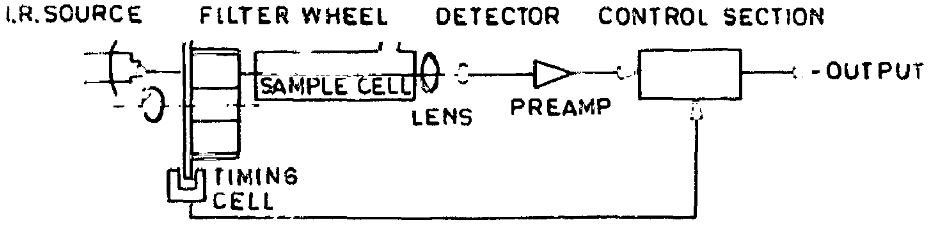


FIG. 4 MULTICOMPONENT NDIR SPECTROMETER
ANALYZER SECTION AND CONTROL SECTION

optical path the amount of COI present in the sample cell affects the detector output very little, because the radiant energy characteristic of that compound has already been largely absorbed by the concentrated COI in the filter cell, hence this filter is called the 'reference cell' . When its place is taken by a nitrogen cell, called the sensitive cell the amount of energy absorbed is proportional to the COI in the sample cell. Hence the detector output is an a.c. signal proportional to the COI in the sample cell and at the frequency of the filter wheel rotative or chopper. Since the selectivity of this instrument is obtained by means of the reference cell filter (plus optical filters to reduce interference) but without the highly specific detector, in Fig.2,3 the device can be multiplexed by employing several pairs of gas cells in the filter-wheel. This type of detector has been thought as the modification to the conventional NDIR Spectrometer.

Description of the Perkin-Elmer Model-180

The Perkin-Elmer Model-180 infrared spectrophotometer is basically a multigrating, dual beam, ratio-recording instrument. The key abscissa element, the grating carousel is driven by stepping motor pulses to a large frequency cam. As the frequency cam rotates an electromechanical encoder mounted on the cam-shift sends

/

signals (via-contact-closures) to the wave-number read-out-display and chart drive motor. At appropriate intervals, microswitches initiate grating changes at 2000, 1000, 500 and 250 cm^{-1}, and a filter cam triggers the positioning of filters in the recombined beam. A front panel switch enables the operator to step the monochrometer in 0.1 cm^{-1} or 0.01 cm^{-1} intervals. The fastest scanning speed of the 180 is 32 cm^{-1}/sec from 4000 to 2000 cm^{-1}, 16 cm^{-1}/sec from 2000 to 1000 etc. The thermopile detector has a fairly slow response time, however, and the fastest practical scanning speed for digital data acquisition is 5 digital data per second.

The biggest problem with the model 180 is the electromechanical encoder on the frequency cam. It consists of a spring wire 'Feelers' that rotates on a series of gold-plated annular rings on a small (4 in.) etched circuit board. These wire contacts may have a tendency to bounce when they hit a dust particle, scratch, or other impediment. Thus erroneous abscissa and ordinate data may spew randomly from the standard interface, requiring the user to carefully edit the incoming data with a software routine.

Fig. 5 illustrates the optical lay-out of P.E. 180. Some of its common features are as follows. There is an extra focal point in the source compartment,

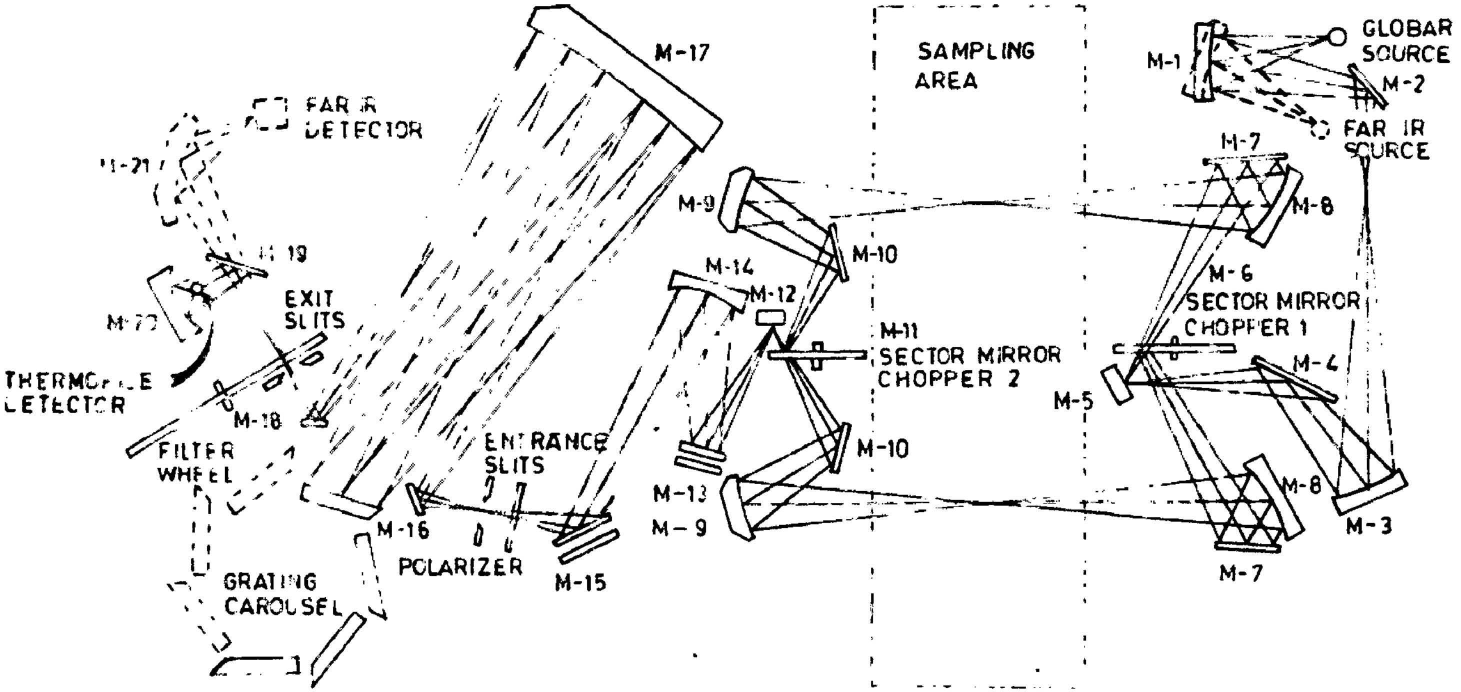


FIGURE 5 OPTICAL LAYOUT OF PERKIN ELMER MODEL 180 INFRARED SPECTROPHOTOMETER

which can be used if a separate source is desired. The sample and reference beams come to a focus (nominally about 11 mm wide) is the sample compartments, which means that many accessories do not work optimally in the 180 unless they are designed to take advantage of this focal point. The second chopper in the monochromator housing, is used to discriminate against self-emission. Shown in Fig. 5 is a polarizer mount between M-15 and the entrance slits, which holds a circular wire grid polarizer, a necessity for serious infrared work. The grating carousel has space for seven gratings. Only four of which are required for the normal range of 4000-180 cm^{-1}. The other three plus the mirror M-21 and the far-IR detector and source, shown in dashed lines, are used to extend the range of the 180 into the far-infrared, to 32 cm^{-1}.

Fig: illustrates the model 180 electrical, mechanical and optical layout to its stand alone configuration. A DPDT switch installed in the recorder circuitry enables the operator to disable the front panel recorder controls : TIME CONSTANT LINA/LINT, 100 %T ADJ, 0 %T ADJ, RATIO MODE (CI/Io), (Io/I) etc.), ORDINATE EXPANSION, ABSCISSA EXPANSION, VARIABLE EXPANSION and SCALE POSITION. In the 'COMPUTER' position, this DPDT turns over the control of the 180's recorder to NOVA and the associated software. In the 'SPECTROMETER' position, the operation can still acquire digital data with the NOVA, while allowing the spectrophotometer to produce an X-Y plot in its normal stand alone mode. Thus offering the operator a chance to monitor such things as the noise level, quality of the dry-air purge, level of transmittance etc., while storing the spectrum in digital form.

Nova is the particular name given to the interface adaptor, by manufacturers, which is provided with the model P.E. 180 spectrometer.

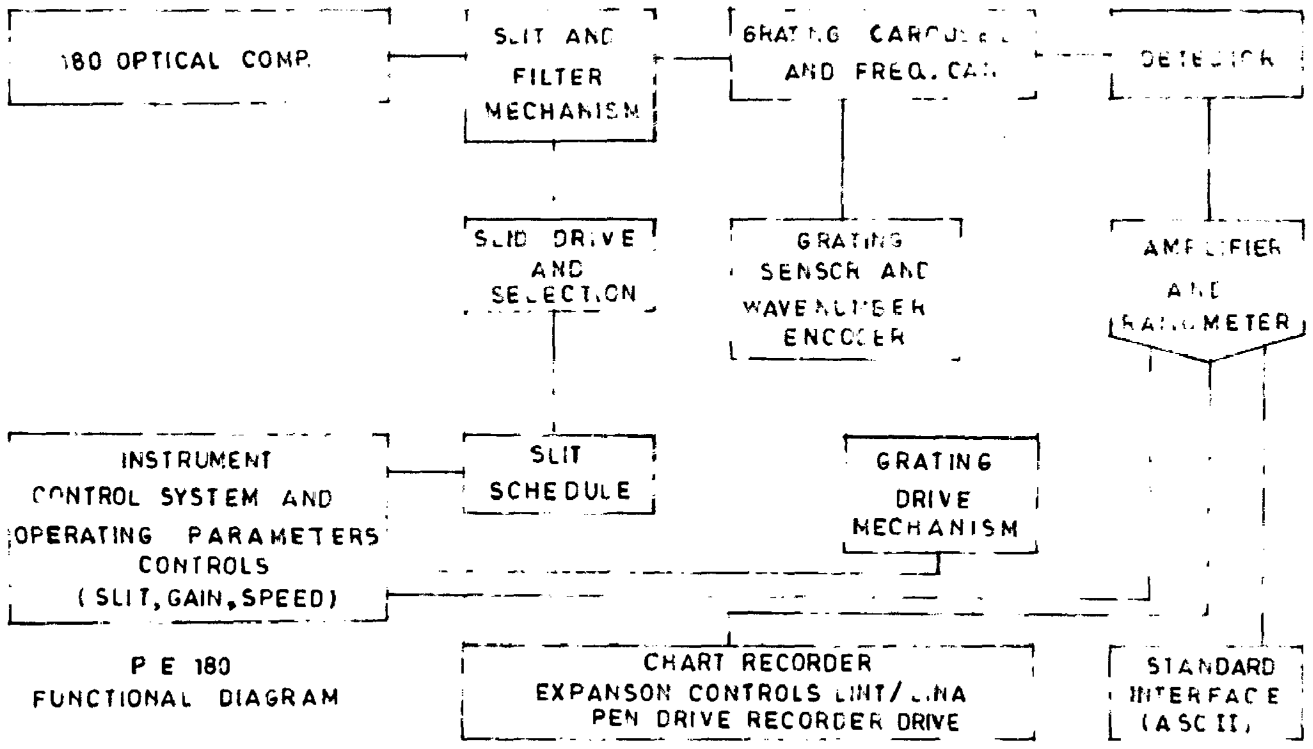

180 OPTICAL COMP.
SLIT AND FILTER MECHANISM
GRATING CAROUSEL AND FREQ. CAM
DETECTOR
SLID DRIVE AND SELECTION
GRATING SENSOR AND WAVENUMBER ENCODER
AMPLIFIER AND RATIOMETER
INSTRUMENT CONTROL SYSTEM AND OPERATING PARAMETERS CONTROLS (SLIT, GAIN, SPEED)
SLIT SCHEDULE
GRATING DRIVE MECHANISM
P E 180
FUNCTIONAL DIAGRAM
CHART RECORDER
EXPANSON CONTROLS LINT/LINA
PEN DRIVE RECORDER DRIVE
STANDARD INTERFACE (ASC II)

FIG. 6 BLOCK DIAGRAM OF ELECTRONICS OF MODEL 180 WITH STANDARD ENTERFACE ADDED.

CHAPTER - IV

A VIEW OF MICROPROCESSORS

Microprocessors are the most versatile and powerful tools electronics engineers have ever had. Right from the beginning, these large-scale-integrated circuit promised to spread into many diverse and complicated application, being both programmable and in terms of what they can do for the money-very low cost. They are rapidly fulfilling their promise. So many successful applications of microprocessors have proved now that they can be used as strongest tools in digital control system design. Microprocessor technology is developing towards the point where it will be economically feasible to design 'intelligence' into the environment at any desirable point. Microprocessors represent scale changes of several orders of magnitude in size, cost and power consumption.

A microprocessor is an LSI chip or collection of chips which performs the arithmatic and logic and the control logic that instructs and sequences the ALU. The same microprocessor can be a part of a host machine or instrument and become its controller, or it can be a unit of a general purpose microcomputer, The essential features of a processor can be specified by five operations :

1. READ the input symbol x
2. COMPARE x with z, the internal state of the processors.
3. WRITE the appropriate output symbol y.
4. CHANGE the internal state z to the new state z'
5. REPEAT the above sequence with a new input symbol x.

In approaching the design of microprocessor systems, the first requirement is to get acquainted with the specialized jargon. Some of the basic terms are defined below,

CENTRAL PROCESSING UNIT : a group of registers and logic that form the arithmatic/logic unit plus another group of registers with associated decoding logic that form the control unit. Most metal-oxide-semiconductor (MOS) devices are single-chip CPUs. In that registers hold as many bits as the word length of the unit (Intel 8080 and Motorola 6800, for example, are 8 bit devices and thus the basic registeres are eight bit wide). With bit-slice devices, however, central processing units of any bit width can be assembeled essentially by connecting the bit-slice parts in parallel. Externally, a bit-slice device will appear to be a coherent single CPU capable of handling words of the desired bit length.

REGISTER : logic elements (gates, flipflops, shift registeres), that, taken together, store 4,8, or 16-bit numbers. They are essentially for temporary storage, in that the contents usually change from one instruction cycle to the next. In fact, much of the microprocessor's operation can be learned by studing the registers, which take part in nearly all operations.

ACCUMULATOR : a register that adds an incoming binary number to its own contents and then substitutes the results for contents.

PROGRAM COUNTER : a register whose contents correspond to the memory address of the next instruction to be carried out. The count usually increases by one as each instruction is carried out, since instructions generally are stored in sequential locations.

INSTRUCTION REGISTER : storage for the binary code for the operation to be performed. Usually this instruction represents the contents of the address just designated by the program counter. However, the contents of the instruction register or the program counter may be changed by the computations. This, of course, represents one of the key ideas of a stored-program computer - instructions, as well as data, can be operated upon and subsequent operations will be determined by the results.

INDEX REGISTER : some memories are organized by index number (the contents of the index register). The address of the next instruction may be found by summing the contents of the program counter and the index register. Increasing the index register by one will cause the processor to go to a new section of memory.

STACK POINTER : a register which comes into use when the microprocessor must service an interrupt - a high priority call from an external device for the central processing unit to suspend temporarily its current operations and divert its attention to the interrupting task. A CPU must store the contents of its registers before it can move on to the interrupt operation. It does this in a stack, so named because information is added to its top, with the information already these being pushed further down. The stack, thus is a last-in-first-out type of memory. The stack pointer register contains the address of the next unused location in the stack.

FLAG : usually a flip-flop storing one bit that indicates some aspect of the status of the central processing unit. For example, a carry flag is set to one when an arithmatic operation produces a carry. A zero flag is set when the result is zero. These flags aid in interpreting the results of certain calculations. Others are sometimes provided to permit access by interupt request lines - for example, if a CPU is engaged in the highest priority of calculation, it may set all status flags to zero - which, loosely translated, means 'don't bother me now'.

If only some of these flags are set, then only certain interupt lines will be able to get through according to their priority.

DIRECT MEMORY ACCESS : a technique that permits a pheripheral device to enter or extract blocks of data from the microcomputer memory without involving the central processing unit. In some cases, a CPU can perform other functions while the transfer occurs.

ACCESS TIME : time between the instant that an address is sent to a memory and the instant that data returns. Since the access time to different locations (addresses) of the memory may be different, the access time specified in a memory device is the path that takes the longest time.

ADDRESS : a number used by the CPU to specify a location in memory.

BAUD : a communications measure of serial data transmission rate ; loosely bits per second but includes character framing START and STOP bits.

BENCHMARK PROGRAM : a sample program used to evaluate and compare computers.

BOOTSTRAP : technique or device for loading first instructions (usually only a few words) of a routine into memory; then using these instructions to bring in the rest of the routine.

The bootstrap loader is usually entered manually or by pressing s special console key.

BRANCH INSTRUCTION : a decision-making instruction that, on appropriate condition, forces a new address into the program counter. The conditions may be zero-result, overflow on add, an external flag raised etc. One of two alternate program segments in the memory are chosen, depending on the results obtained.

BUS : a group of wires that allow memory, CPU and I/O devices to exchange words.

BREAK POINT : a location specified by the user at which program execution (real or simulated) is to terminate. Used to aid in locating program errors.

DEBUG : to eliminate programming mistakes, including omissions, from a program.

DEBUG PROGRAM : help the programmer to find errors in his programs while they are running on the computer, and allow him to replace or patch instructions into (or out of) his program.

INSTRUCTION : a set of bits that defines a computer operation, and is a basic command understood by the CPU. It may move data, do arithmatic and logic functions, control I/O devices or make decisions as to which instruction to execute next.

REQUEST-INTERRUPT : a signal the computer that temporarily suspends the normal sequence of a routine and transfers control to a special routine. Operation can be resumed from this point later. Ability to handle interrupts is very useful in communication applications where it allows the microprocessor to service many channels.

MEMORY : the part that holds data and instructions. Each instruction or datum is assigned a unique address that is used by the CPU when fetching or storing the information.

MEMORY ADDRESS REGISTER : the CPU register that holds the address of the memory location being accessed.

ROM (Read Only Memory) : Read-Only Memory (Fixed Memory) is any type of memory that can not be readily rewritten. ROM requires a masking operation during production to permanently record or data patterns in it. The information is stored on a permanent basis and used repetitively. Such storage is useful for programs or tables of data that remain fixed and is usually randomly accessible.

RAM (Random Access Memory) : any type of memory that has both read and write capability. It is randomly accessible in the sense that the time required to read from or to write into the memory, is independent of the location of the memory where data was most recently read from or written into. In contrast, in a Serial Access Memory, this time is variable.

SERIAL ACCESS MEMORY : any type of memory in which the time required to read from or write into the memory is dependent on the location in the memory. This type of memory has to wait while undesired memory locations are accessed. Examples are paper tape, disc, magnetic tape etc. In a Randam Access Memory, access time is constant.

SCRATCH-PAD MEMORY : RAM or register that are used to store temporary intermediate results (data), or memory addressed (pointers).

PAGE : : a natural grouping of memory locations by higher order address bits. In an 8-bit microprocessor 2^8 = 256 consecutive bytes often may constitute a page. Then words on the same page only differ in the lower-order 8 bit address bits.

PLA (Programmable Logic Array) : a PLA is an array of logic elements that can be programmed to perform a specific logic function. In this sense, the array of logic elements can be as simple as a gate or as complex as a ROM. The array can be programmed (normally mask programmable) so that a given input combination produces a known output function.

PROM (Programmable Read-Only Memory) : an integrated circuit memory array that is manufactured with a pattern of either all logical zeros or ones and has a specific pattern writtn into it by the user by a

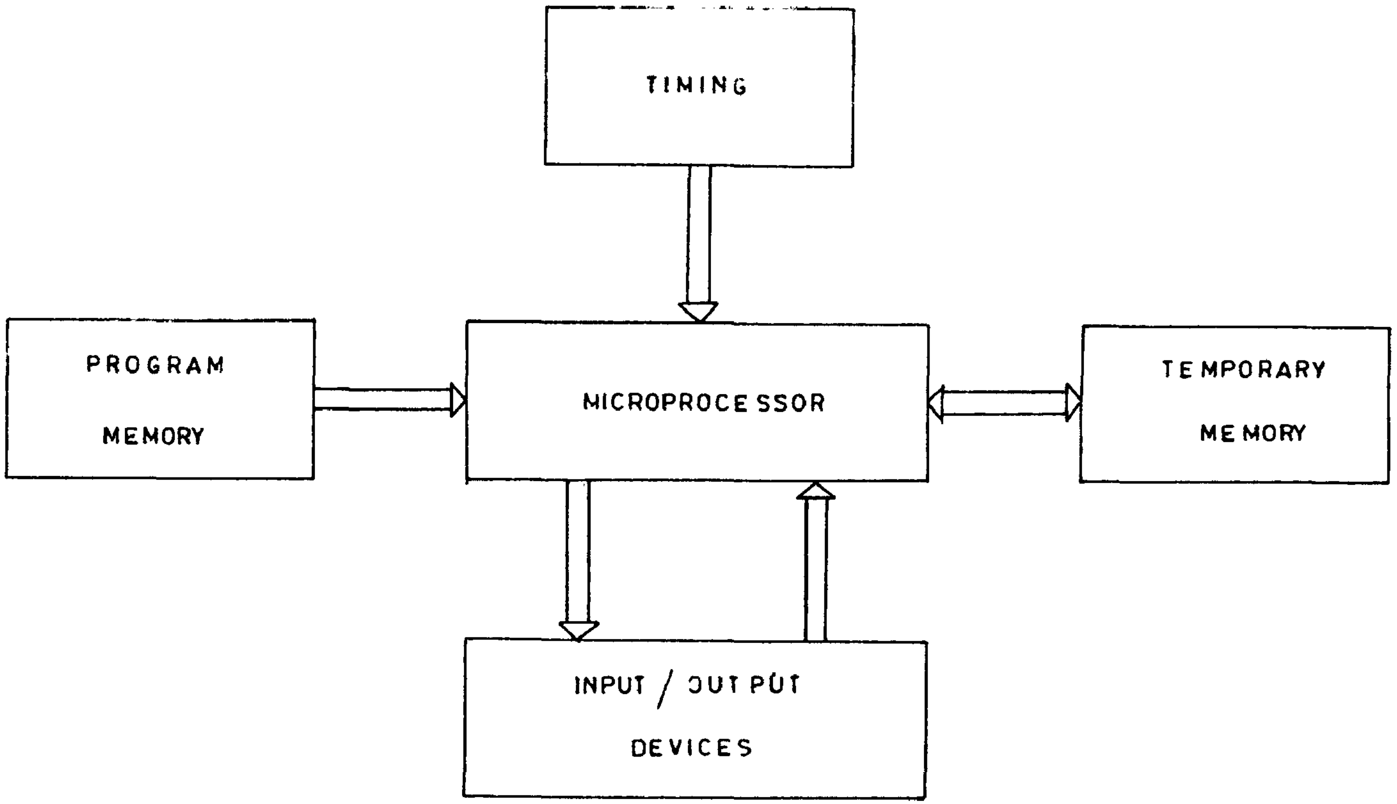


FIG. 7 MICROPROCESSOR CONTROLLER ORGANIZATION

special hardware programmer. Some PROMs called EAROMs, Electrically Alterable Read-Only Memory, can be erased and reprogrammed.

HARDWARE : a relatively permanent physical embodiment of an algorithmic process.

SOFTWARE : a relatively impermanent, informational embodiment of an algorithm.

The first microprocessor, the Intel 4004, was introduced in 1971 by Intel Corporation. This device, developed primarily for calculator-oriented applications, is a monolithic integrated circuit (IC) employing large-scale-integration (LSI) in metal-oxide-semiconductor (MOS) technology. The 4004 was soon follcwed by a variety of microprocessors, with most of the major semiconductor manufacturers producing one or more types. Most microprocessors use LSI technology employing either P-channel MOS, n-channel MOS, silicon-on-sapphire MOS, complementary MOS, or bipolar processes.

Microprocessors are having an impact on the design of virtually all digital systems. They are finding application in many systems which formerly used random logic. In complex systems, a microcomputer is often cheaper and convenient to use than random logic. One can expect, as time and technology progress, that the processor cost will continue to diminish in a manner reminiscent of that of operational amplifier IC.

CHAPTER - V

DIGITAL DATA PROCESSING

It is quite obvious to learn that the most precise spectrometers are those that incorporate a double-beam feature; that is, one optical path from the source to the detector passes through the sample, and the other travels directly or via a neutral or nonabsorbing filter. The absorption of the sample is measured as a ratio of the direct and indirect paths. Thus any variations in strength of the source or of the detector sensitivity, together with the effects of film or dirt obscuring the main optical path, are canceled out by the rationing process. In an NDIR spectrometer, a similar advantage is gained from the alternate presentation of the two filter cells used for each COI, hence it is desired to take the ratio of the signals obtained from each pair.

The filter wheel NDIR spectrometer has been called a 'double beam in time' . If the detector delivers and electrical signal voltage proportional to the transmission or absorption of radiation along each optical path, it desired to obtain an instrument output E_o equal to

$$E_o = A(e_b - \frac{e_r}{e_s}) \qquad (1)$$

where e_r and e_s are the reference and sensitive path voltages and e_b is a bias voltage added to the computed signal to give a zero reading when the conaentration of COI is zero in the sample cell. The division e_r/e_s in one version of the instrument is accomplished with analog circuitry using a voltage controlled amplifier or automatic-gain-control (AGC), as shown in Fig.

The ganged switches represent gates synchronized by the chopper filter wheel, alternatively emergizing the sense and reference channels. When the sense (COI filter) channel is active, the AGC circuit acts through the feedback integrating amplifier to make the sense output E_s equal to the fixed bias voltage. Thus the gain of the AGC circuit is proportional to $1/e_s$. When the switches alternately gate through the reference signal, the AGC gain remains at $1/e_s$, owing to a long time constant, and

$$E_R = e_r \left(\frac{E_s}{e_s} + Z \right) \qquad (2)$$

where Z is the zero-adjustment for the reference channel.

The complete expression for the output is obtained from the difference between E_k and E_s, since the differential amplifier G_o also has a long time constant compared with the pulse rates. Consequently

$$E_o = G_o \left[E_s - E_s\left(\frac{e_r}{e_s} \right) - Z \right]$$

$$= A \left[e_b - \frac{e_r}{e_s} \right] \qquad (3)$$

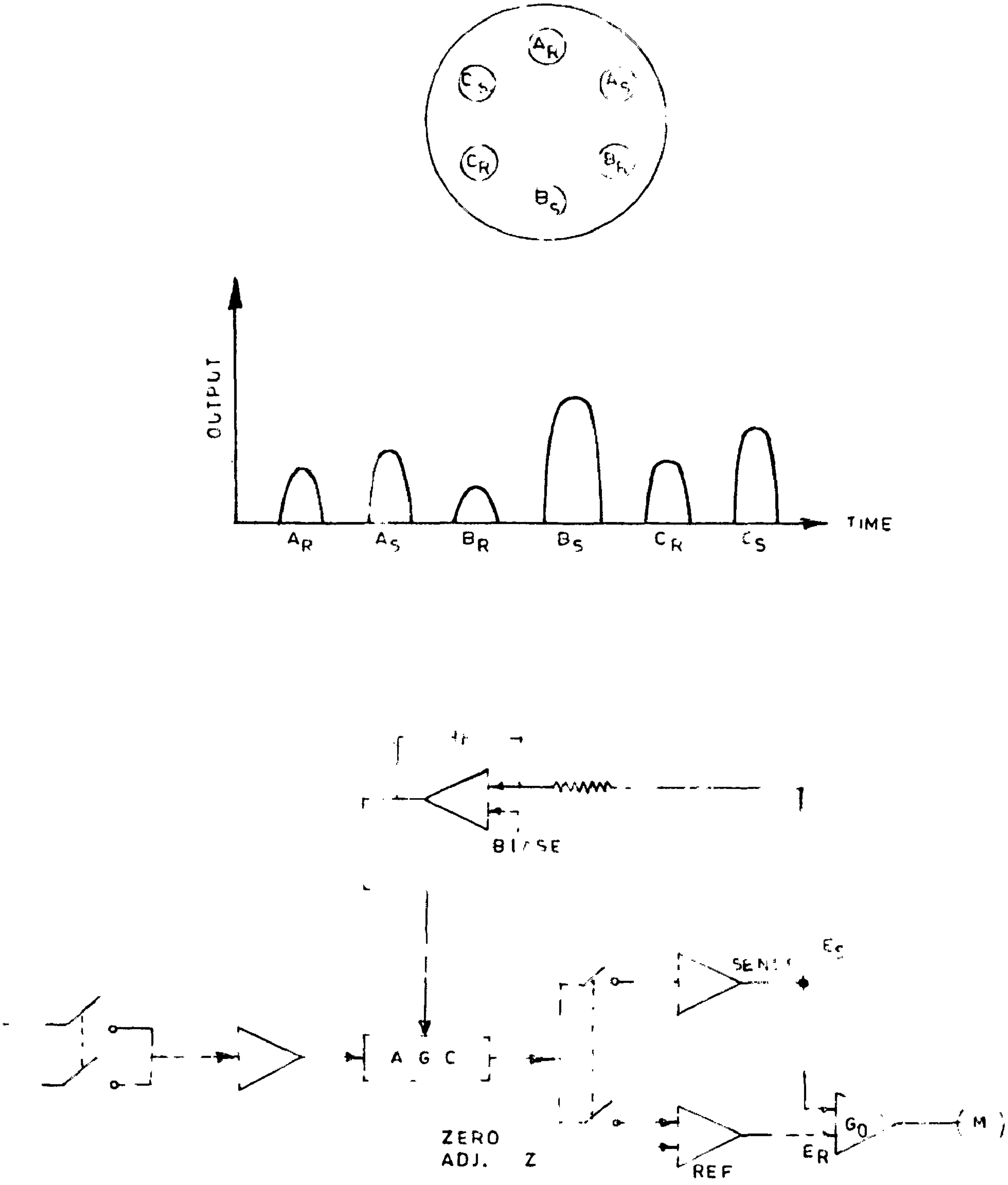


FIG. 8 ANALOG SIGNAL PROCESSING

which is of the same form as (1). A being equal to G_oE_s and

$$e_b = (1 - \frac{Z}{E_s}) \qquad (4)$$

Digital Data Processing

Although the analog circuitry developed for this device performs its task very well, there are several important advantages which may accrue from all-digital circuitry, specifically a microprocessor. First, the analog circuitry described, although time shared between sense and reference channels, must be duplicated for every component; that is, a three-component analyzer requires three AGCs and associated amplifiers and gates.

Second, a great deal of gating is required so that a certain portion of the circuitry is digital in any event.

Third, time constants are long in order to 'hold' voltages for subtraction and division - with time constants up to 12 sec a digital memory may save space and component dollers.

Finally, an analyzer of this type is necessarily customized to the extent of being adjustable for different COIs, requiring changes in a few components.

With a microprocessor, programs substitute for custom hardware, and the large dynamic range possible often eliminates the need entirely. Also the microprocessor allows signal processing techniques (such as filtering) to be added or altered with no new hardware, and digital processing generally adds to long term stability, particularly with respect to temperature changes.

For these reasons let us consider what may be involved in applying a microprocessor.to an instrument of this type. To make the problem more realistic, we assume a three-component instrument in which the filter wheel is driven by a pulse stepping motor rather than continuously. The microprocessor has the additional task of synchronizing the filter positions and the signal processing task for the three COIs (or more COIs). As still another complication, let the analog signal from the detector preamplifier be integrated for a precise fixed period before digital conversion and further processing in order to increase the signal strength.

Computer Spectrometer System

Fig. 9 shows a block diagram of the complete computer-spectrometer system. The basic block diagram for computer assisted system was developed at University of Miami.

We will use this basic block diagram as the base of the microprocessor based digital control system for NDIR spectrophotometer.

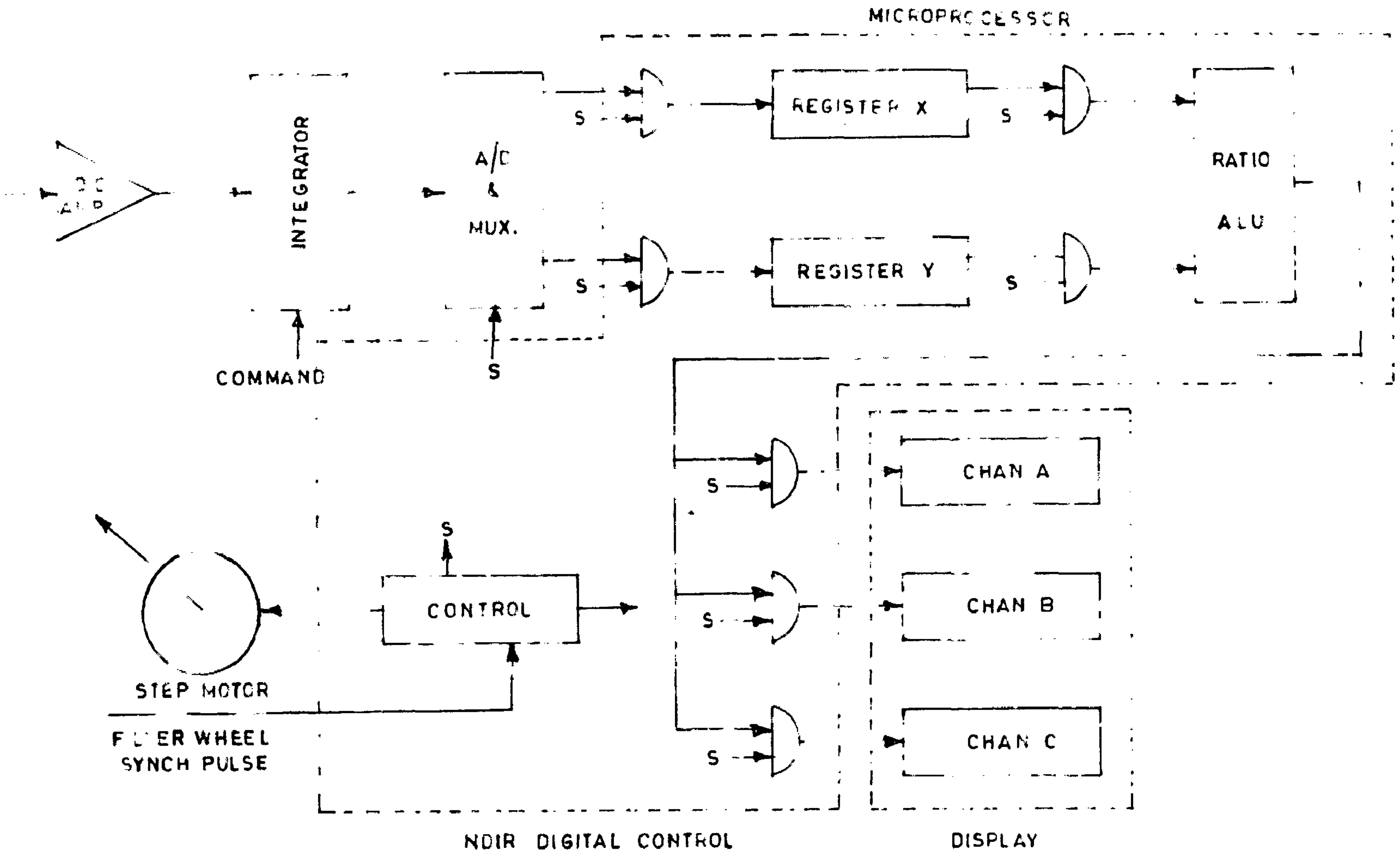


FIG. 9 NDIR DIGITAL CONTROL

Having received the external signal, the program initiates the following actions. Two counters (registers) termed the ratio counter and the display ~~counter,~~ used for switching logic are set to zero, and a signal is sent to the stepping motor to rotate the first filter, which is the reference filter of the first COI.

The program then delays for 200 msec while the first filter settles into place. At the end of this time, a signal is sent to the analog integrator to reset and begin integrating the pre-amplifier output. The integration continues until the program has measured a 600 msec delay, at which time the integrator stops. The A/D convertor then holds the current output of the integrator in digital form.

The ratio bit counter is then tested. If it holds a '0' , the A/D contents will be loaded into a register we shall call X. After a further 200 msec delay, the ratio counter bit is complemented - if it was '1' it becomes '0' and if '0' it becomes '1' . The filter wheel is stepped (to the sense filter of the first COI), another 200 msec delay occurs, and the integrator again zeroed and restarted. At the end of the integration, the ratio-counter shows '1' , so the A/D controls are loaded into register Y instead. The program Men computes the ratio X/Y which is the e_r/e_s required for (3).

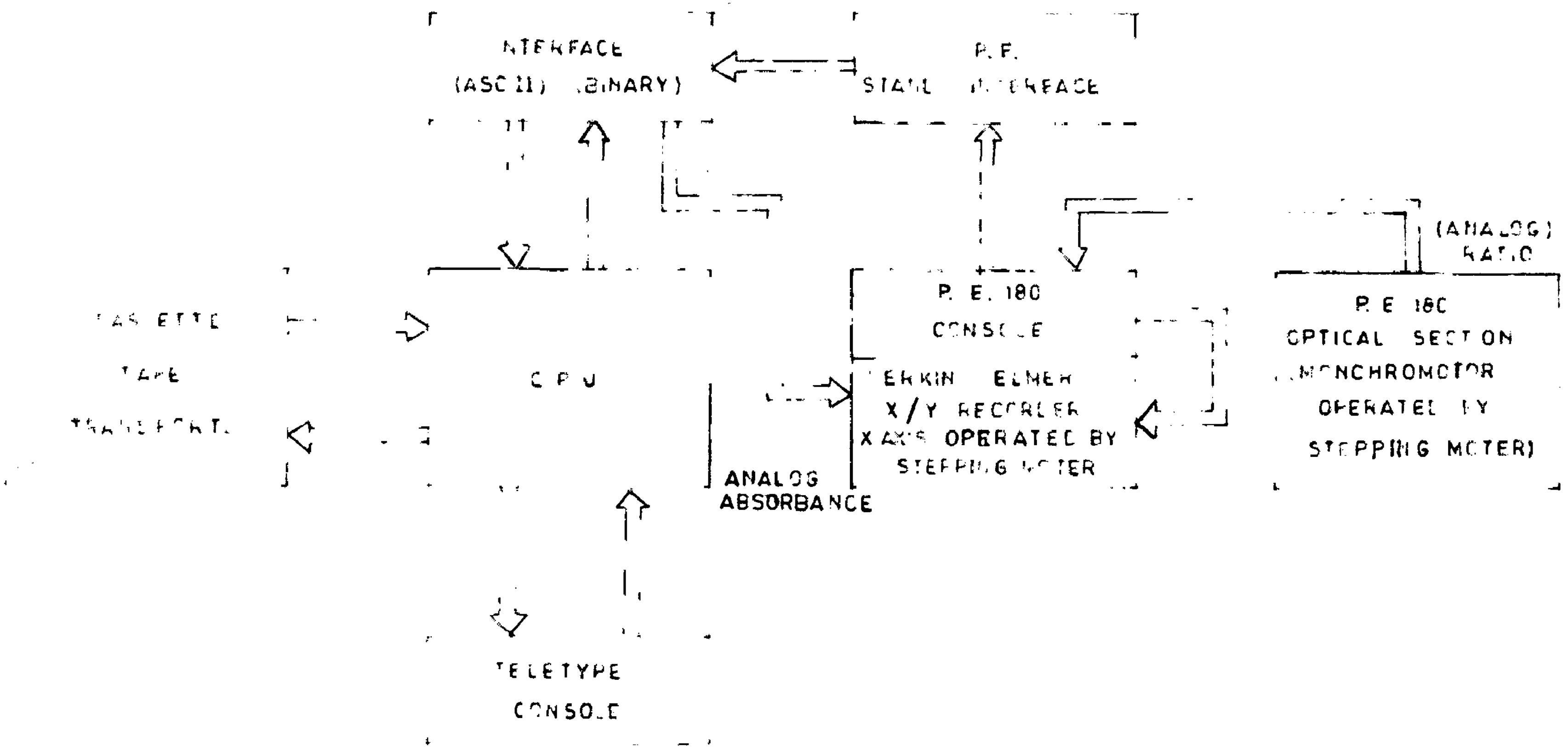


FIG. 10 BLOCK DIAGRAM OF COMPUTER SPECTROMETER SYSTEM

To anticipate, the division routine normally used by the microprocessor requires that the divident be smaller than the divisor. In the physical configuration, the reference signal is always smaller than the sense signal. Since the formal results from the absorption of pure COI. Hence, if this inequality condition does not hold, there is an error, such as a failure of the filter wheel to step. If this condition is sensed, an error routine is to be entered which continues to step the wheel and test for the synchronizing position.

If there is no error, the value of the ratio is sent to the correct display channel, utilizing the display counter. If the display counter holds '0' the results of the ratio calculation is sent to channel A. The display counter is then incremented. If the counter holds a '1' , the ratio is sent to channel B and the counter incremented again. If the counter holds any other value (more than '1'), the ratio is loaded into channel C and the program is returned to the initializing step.

DIGITAL CONTROL

For a moment, neglecting the problem of linearization, zero setting, span adjustment (system gain) and display, and consider the end product of the signal processing task is the ratio of the reference and sense channel voltages and variables. A block diagram of the control processing task (given in Fig. 10) shows the micro-processor tasks under consideration.

The time available for control and signal processing is more than adequate, considering typical micro-processors computing speeds. Table 1 indicates a typical time sequence for these operations, assuming each COI measurement is updated every 6 sec. The corresponding control flow chart is given in Fig. 11

Milliseconds	Action Required
0	Reset Sequences (from filter wheel synch. pulse).
0	Step filter wheel (motor)
200	Reset integrator and turn on
800	Integrator off. Store A/D output in register X
1000	Step filter wheel
1200	Reset integrator and turn on
1800	Integrator off. Store A/D output in register Y
1900	Compute register X/register Y ratio and gate result into channel A
2000	Step filter wheel
2200	Reset integrator and turn on
2800	Integrator. Store A/D output in register X
3000	Step filter wheel
3200	Reset integrator and turn on
3800	Integrator off. Store A/D output in register Y
3900	Compute register X/register Y ratio and gate result into channel B
4000	Step filter wheel
4200	Reset integrator and turn on
4800	Integrator off. Store A/D output in register X
5000	Step filter wheel
5200	Reset integrator and turn on
5800	Integrator off. Store A/D output in register Y
5900	Compute register X/register Y ratio and gate result into channel C
6000-0	Reset sequence - step filter wheel.

Table 1 Signal Processor Sequence Table

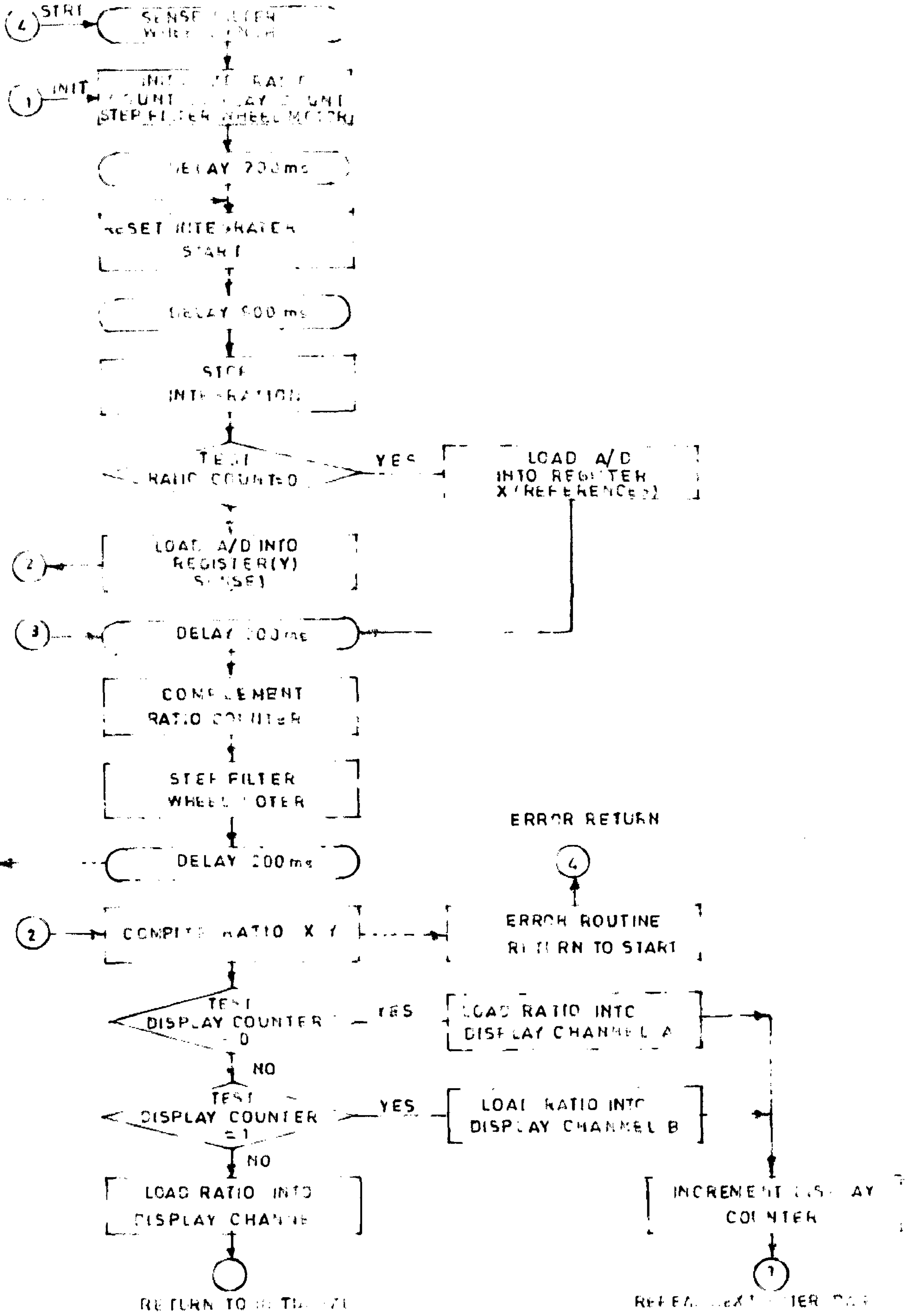


FIG. 11 NDIR CONTROL FLOW CHART

CHAPTER - VI

INFRARED DATA HANDLING AND REDUCTION

The main program, in all, carries out seven principal functions, selected by the operator in response to the query FUNCTION Using ↓ to represent a carriage return, the seven allowed responses are described below-

.P ↓enables operator to reset the zero and 100 % T counters from their default values of 100 and 10,000 or from previously set values.

C ↓ Reads floating point absorbance data from a file on a cassette tape, after echoing the ASCII title line making up the first record of the file.

W ↓ Write a data file and an operator written title line onto a cassette tape.

S ↓ Input to ASCII data point from the spectrophotometer to be stored in memory as a string of floating point ordinate values. The operator can choose to average any number of spectra under this mode of operation, combining them all into one spectrum.

T ↓ Type the ordinate values from the spectrum stored in core onto the teletype.

R ↓ Ratio two spectra stored as cassette tape files by subtracting absorbances (scale factor for subtrahend included)

M ↓ A response which indicate the operator's wish to either plot or smooth up to 4096 points stored in memory. An M ↓ response is followed tby he query 'SMOOTH OR PLOT'? requestion an operator's response of S↓ or P↓. S↓ results in the application of a 21 point quartic smooth to the stored spectrum. P ↓ causes the stored spectrum to be plotted on the spectrometer's X-Y recorder.

Detailed Description of Main Program Functions.

1. P↓ Adjusting Live Zero and 100 % T Counters.

The model 180 uses an electronic counter, which counts from zero to as high as 12,000, to represent the ratio of the saample beam intensity to the reference beam intensity (I/I_0). When the sample beam shutter is closed, a minimum count, usually between 50 and 200 counts, is produced. The minimum counts provides for a Live zero. , so that very low transmittance (I = 0 % T) can be recorded with about 1 % accuracy. The zero count can be manually adjusted via a trimpot on the analog filter p.c. board. The full count trimpot on the same p.c. board can be used to adjust 100 % T count close to 10,000, provided that the sample and reference beams have been properly phased. As a rule, neither the live zero count nor the full scale count match their nominal values of 100 and 10,000, respectively. For this reason, it is necessary to have this routine in order to adjust software counters to represent the actual

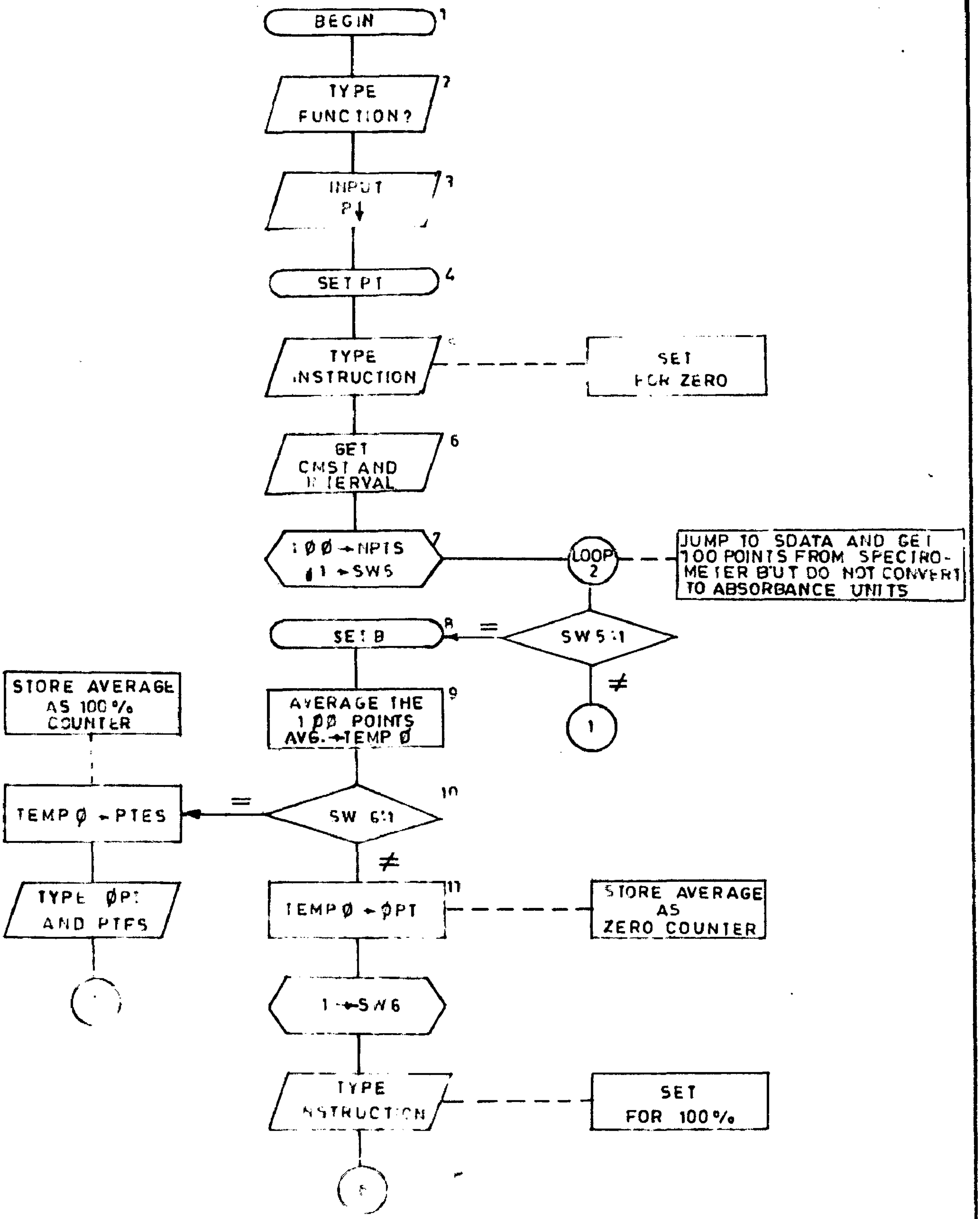


FIGURE 12 FLOW CHART FOR SETTING ZERO AND 100 % T

signal output of the spectrometer. This is the only way in which one can be certain of matching results over long period of time, over which time the source may be changed, p.c. boards may be replaced or adjusted, external temperature may change, and so on.

Fig.12 illustrates the straightforward logic flow employed to average 100 points with the sample shutter closed, then 100 points with both beams empty, to get representative values for the zero and 100 % T counters. This routine sets a switch (SW5) which enables it to use the spectrometer data acquisition routine (SDATA), but without modifying the I/I_0 counts in any way.

2. S ↓ Spectrometer Data Acquisition

A response of S ↓ to the query 'FUNCTION'? generates requests for the starting wave number (CMST), the data interval (INCRE), the number of points to be acquired (NPTS), whether or not averaging is desired, and if so, the number of spectra to be averaged (AVGN). After computer types 'BEGIN' , the operator may start the spectrophotometer scan after all of the instrument setting have been made.

According to Savitzky and Hannah [ ], a model 180 will not accurately respond faster than about five data intervals per second. In addition, our smoothing routine requires about a 21-point half width (FWHM) to avoid distorting the top of a peak. The two restrictions serve to guide to operator in setting the adjustable parameters of the spectrometer. For the narrow absorption bands of, say, a no.2 fuel oil ,data interval of 0.2 or 0.5 cm^{-1} may be necessary, alongwith a midest scanning speed (1 to 2.5 cm^{-1}/sec), and a narrow slit width. For broader absorption bands, say those of large proteins, a data interval of 1 or 2 cm^{-1} is sufficient, larger slit widths can be employed, as well as higher scanning speeds. Naturally the S/N is improved at larger slit openings, thus reducing the need to average spectra to those instances. There does not appear to be any point in scanning at slower speeds since no improvement in S/N should result. The S/N is determined by the absolute attenuation of the sample and reference beams, and the gain and slit settings of the spectrometer. When S/N is low, enhancement can be obtained through spectrum averaging and/or smoothing. The operator of the computer-spectrometer should be familiar enough with the available tradeoffs, both in the spectrometer and in the data handling, to be able to take full advantage of the capabilities of the system.

The flow chart for the acquisition of abscissa and ordinate data from the spectrometer is given. Using the floating point interpreter (F PI), the ASCII character strings are converted to floating point numbers. The

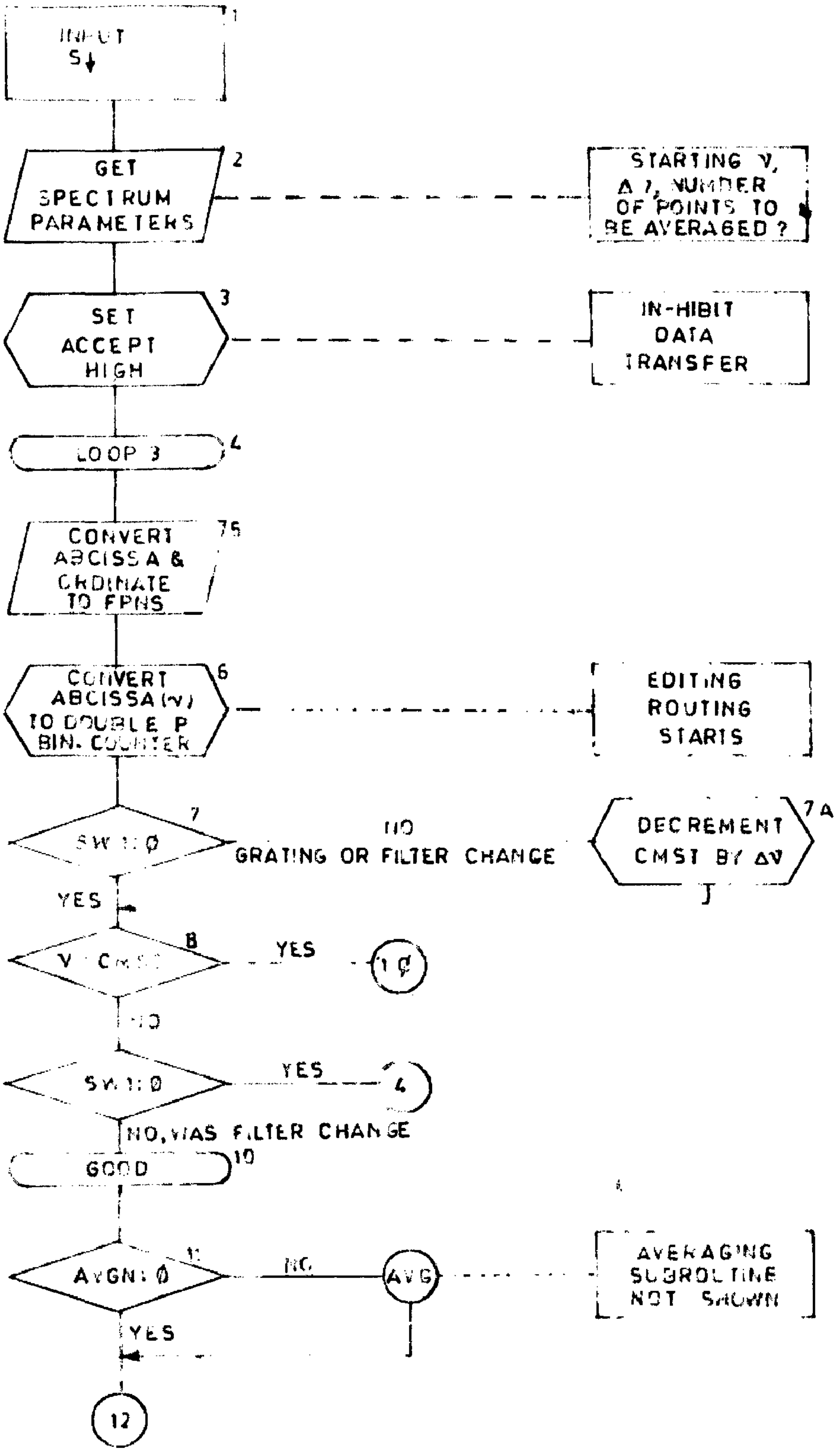


FIGURE 13 FLOW CHART SPECTROMETER DATA ACQUISITION

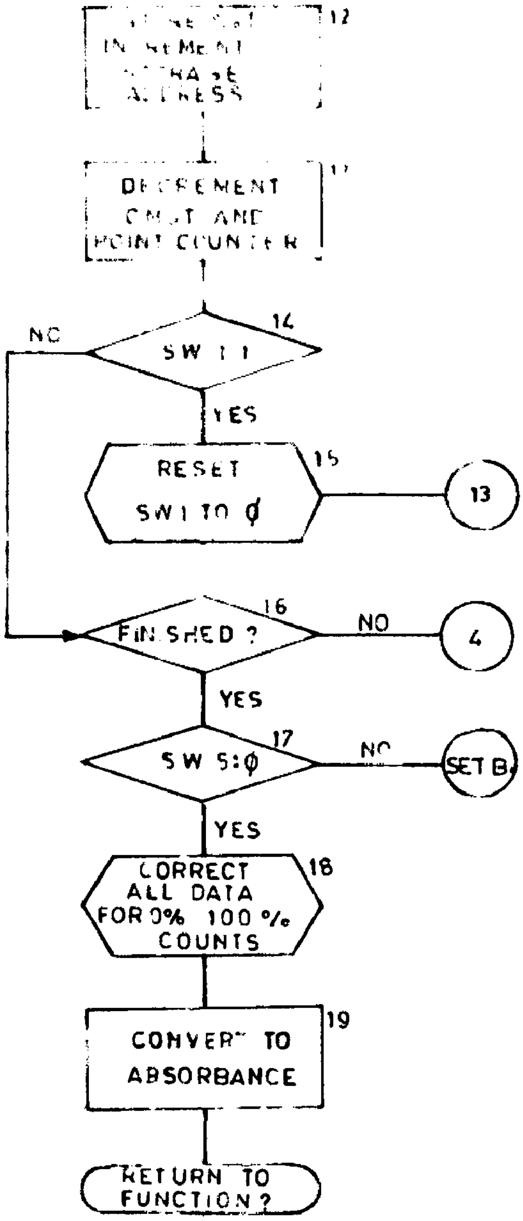


FIGURE 13 FLOW CHART SPECTRMETER DATA ACQUISITION

FPI requires that a get character routine be pointed to by an address in page zero location GETC. The get character subrouting used for the data acquisition from the 180 can be as follows (assembly language has been used).

```
GCHAR  :  DOB     Ø, Ø5    : Set accept low
          SKPON   Ø5       : Skip next instruction when
                             'done' is set
          JMP     .-1
          DIAP    Ø, Ø5    : Take character, set accept high
          LDA     2,MSK
          AND     2,Ø      : Right adjust
          NIOC    Ø5       : Clear 'done'
          JMP     Ø,3      : Return
MSK       177
```

The above subroutine will clarify the actions of the commands 'DOB' and 'DIAP' (the latter is a combination of 'DIA' followed by the special I/O pulses), if we have in mind some of the formerly written lines that the software sequence follows

1. RESET ACCEPT FF LOW
2. WAIT UNTIL READY IS HIGH
3. TAKE DATA AND SET ACCEPT FF HIGH

The appearance of Ø5 in the I/O commands (since here, we have taken five data interval per second) merely indicates the device code for the particular interface received, ν , using double precision binary subtraction rather than floating point arithmatic to eliminate round off errors. A precise agreement causes the exit at step 10 to 600D. Any difference between CMST and ν goes into TEMØ, where its absolute value is tested against EPSI ($\equiv$ INCRE/IØ) is reached to see if it is greater than CMST, and if so, the data are rejected and a new set of data requested. If not, it is tested to see if ν is only one data internal (±10 %) less than CMST. If so, a grating change has just been passed and the data are accepted, setting SWI in addition. The 180 does not transmit regular data at a grating change, but skips one data interval, thus SWI is set in order to store the subsequent datum twice. A precaution to the user is to avoid, setting up a scan so that a grating change is the last point in the scan. If this is done, the spectrometer must be allowed to pass through the grating change past the next data interval in order to acquire the final point. The editing routine is an absolute necessity to ensüre proper registry when plotting, ratioing or averaging spectras.

The remainder of the flow chart is straightforward. Once the abscissa is determined to be good, the routine decrements CMST by an data interval and tests AVGN to see if averaging was requested. If averaging is to take place the subroutine AVG (not shown in flow chart) takes the floating point ordinate, devides it by AVGN (the number of spectra being averaged), and adds the results to the previous by stored value for the same $^{\nu}$. As the data are entered, they are stored in T counts (between zero and 12,000 counts) in floating point notation. This enables the control to take data at very high speeds, much faster, in fact, than the commercial model 180 can scan. At the end of the spectrum (or spectra, if averaging) the entire spectrum is corrected for the live zero, than converted to absolute absorbances (ranging from about +0.05 to about 2 absorbance units).

3. M↓ Plotting or Smoothing Spectrum In Memory

The digital control system replies to a response of M↓ by requesting the number of points (NPTS), and then asking 'SMOOTH OR PLOT'? Flow change (given in Fig. 14) illustrates the logic flow followed in response to an S↓ or a P↓ response. The smoothing routine is based on 21-point smooth (as given in reference - J. Steiner, Y.J. Termonia and J. Deltour, Comments on smoothing and differentiation of data by simplified Least square procedure, Anal. Chem. 44:1906 (1972)). This requires that certain precautions be followed when recording the spectrum. First, the half width (FWHM) of the narrowest peak of interest should not be less than 21 data intervals, otherwise noticable degradation in the peak height will

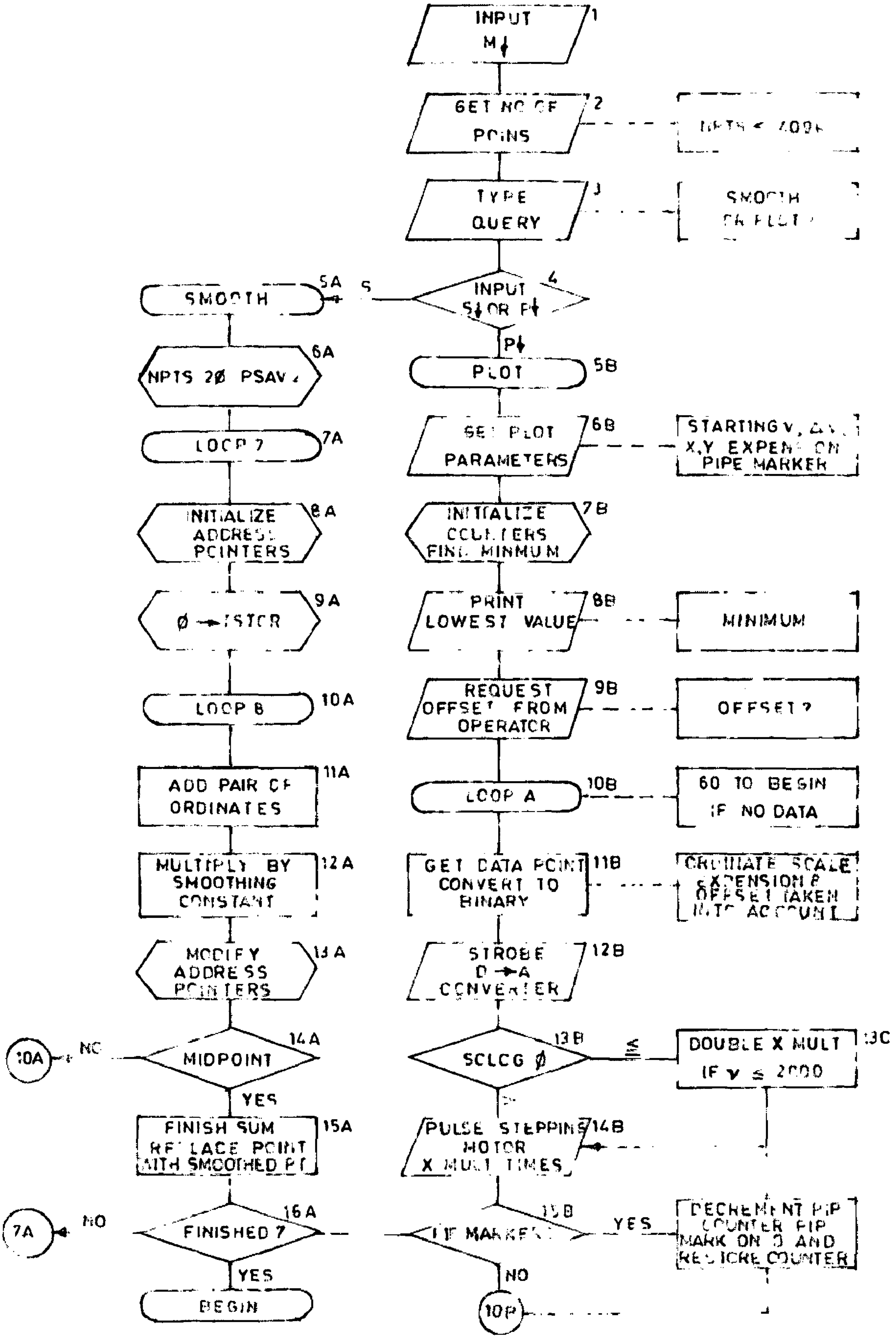

INPUT M
1
GET NO OF POINS
2
NPTS < 4096
TYPE QUERY
3
SMOOTH OR PLOT ?
INPUT S OR P
4
S
SMOOTH
5A
P
PLOT
5B
NPTS 20 PSAV
6A
SET PLOT PARAMETERS
6B
STARTING V, ΔV, X,Y EXPENSION PIPE MARKER
LOOP 7
7A
INITIALIZE COUNTERS FIND MINMUM
7B
INITIALIZE ADDRESS POINTERS
8A
PRINT LOWEST VALUE
8B
MINIMUM
0 → ISTOR
9A
REQUEST OFFSET FROM OPERATOR
9B
OFFSET ?
LOOP 8
10A
LOOP A
10B
GO TO BEGIN IF NO DATA
ADD PAIR OF ORDINATES
11A
GET DATA POINT CONVERT TO BINARY
11B
ORDINATE SCALE EXPENSION & OFFSET TAKEN INTO ACCOUNT
MULTIPLY BY SMOOTHING CONSTANT
12A
STROBE D → A CONVERTER
12B
MODIFY ADDRESS POINTERS
13A
MIDPOINT
14A
NO
10A
YES
SCLCG 0
13B
≤
DOUBLE X MULT IF y ≤ 2000
13C
>
FINISH SUM REPLACE POINT WITH SMOOTHED PT
15A
PULSE STEPPING MOTOR X MULT TIMES
14B
FINISHED ?
16A
NO
7A
YES
BEGIN
IF MARKER
15B
YES
DECREMENT PIP COUNTER PIP MARK ON 0 AND RESTORE COUNTER
NO
10P

FIGURE 14 SMOOTHING OR PLOTTING DATA IN MEMORY

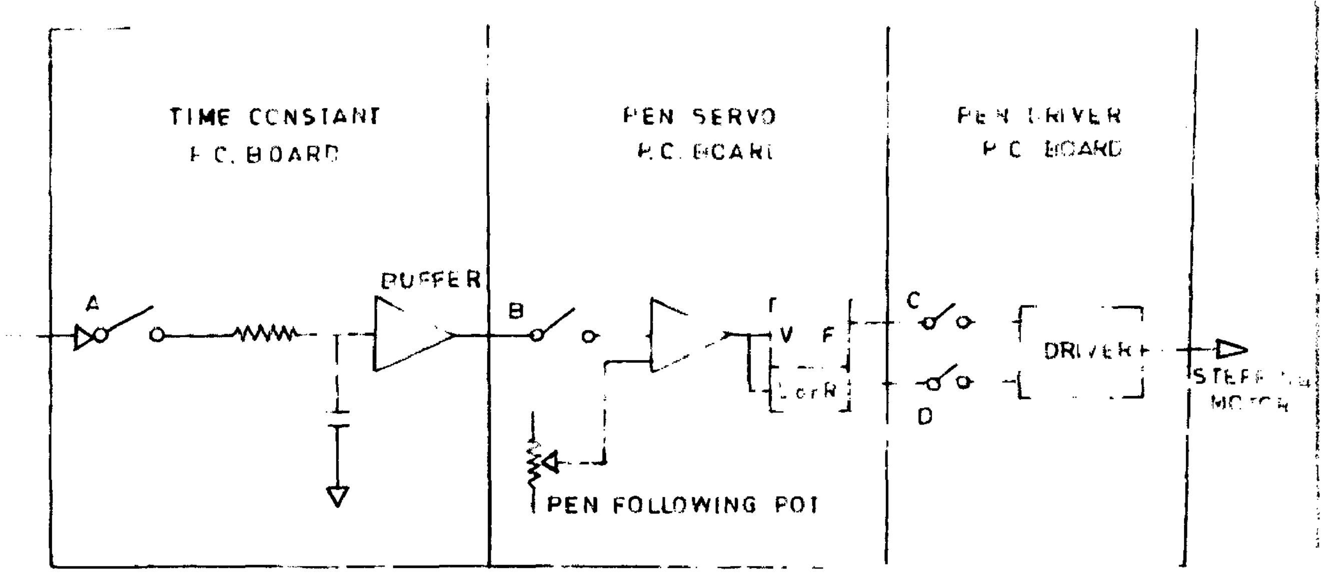
occur upon smoothing.

Second, adjacent peaks which are to remain separated after the smooth must have their peak maxima located at least 21 data intervals apart.

When these two conditions can not be met, the alternative available to the operator is to average four spectra together, with an attendant improvement of S/N of $2(=\sqrt{\text{AVGN}})$. In setting up the ν limits, onc must remember that, in the 21-point smoothing operation, the first and last 10 points of the spectrum are not modified in any way.

The course of action takes after a $P\downarrow$ response is shown in flow chart (Fig. 14). The computer control system requests sufficient information to set up the plot, including the starting wave number, the data interval, abscissa and ordinate expansion factors (the abscissa expansion factor must multiples of 1/2 and the ordinate expansion factor may be any positive number from zero to 10^{64}), whether or not a pip marker is desired, if so, the interval (in points) between pip marks . Next, the program searches the stored spectrum, finds the minimum absorbance value, prints it, and is prepared to plot the spectrum with the minimum absorbance adjusted to appear as zero on the plotted spectrum. Before plotting, however, the operator is allowed to enter an OFFSET either to displace the spectrum so that the plot will display absolute absorbance values, or to vertically offset the spectrum any amount desired. Negative values for OFFSET should be avoided, since the combination of two's-component logic in the computer, the single polarity of the D→A converter, and the 0 to 2.37 V range of the recorder all combines to make small negative absorbance values look like large positive values. (Data are taken from the P.E. 180 spectrophotometer system). The computer subtracts 2000 cm^{-1} from the starting frequency in order to set up a binary counter, SCLCG , which will automatically double the number of stepping motor pulses per abscissa when 2000 cm^{-1} is reached on the plot (plotted from high wave number to low). Each ordinate, varying between about 0 and 2 absorbance units, is first adjusted for the minimum and offset; i.e., ORDINATE' = ORDINATE - MINIMUM + OFFSET . Since a 10-bit D → A counter is employed, and the unexpanded ordinate scale of the 180 is from 0 to 1.5 absorbance units, each ordinate is multiplied by a scale factor of 682.67 before it is converted into truncated binary integer, and the most significant 10 bits are strobbed into the buffer of DAC. The pip marks are produced by simply adding ten percent of full scale to the contents of DAC buffer, without moving along the X-axis (See Fig.15B).

Fig.15B illustrates the possible inputs to the chart and pen drive circuitry of the 180. Point A allows



POINT A: 0-3.95V FULL SCALE

POINT B: 0-2.37V FULL SCALE

POINT C: 0-0.5V (0 400 pps PULSE RATE)

POINT D: MOTOR DIRECTION CONTROL 5V=RIGHT
0V=LEFT

FIG.15 B-POSSIBLE INPUTS TO MODEL 180 X/Y RECORDER CIRCUITRY

the operator to retain the function of the spectrometer's RC pen damping control. Our system uses point B for the input to ordinate values from the DAC. Two trim pots connected external to the DAC (not shown) adjust to full scale gain and zero offset for the ordinate. The motor pulse output from the interface goes directly to point C on the pen driver board where it drives the stepping motor (abscissa control) at a rate determined by a counter in the software.

Setting the spectrometer up for a plot requires throwing the DPDT switch to 'COMPUTER' , putting the pen to the right starting point, setting the monochromaton to CONSTANT CM(-1), turning the X-Y recorder ON, and pushing the SCAN button to lower the pen. During the plot, there must be some energy reaching the spectrophotometer detector, or the pen will automatically lift from the paper due to low energy.

W ↓ and C ↓ Write or Read a Cassette File

Data General's SOS (Stand-alone Operating System) provides the user with simple I/O calls to standard peripheral devices. Call such as

.RDL for read a line

.WRL for write a line

.RDS for read sequential

and .WRS for write sequential.

These calls make it easy to move data between cassette files and core memory. Flow chart (shown in Fig.15 illustrates both read and write flow diagrams. Each read or write command requires a byte counter and a byte pointer which are passed to SOS in one hardwave accumulators ACØ and ACI.

Each data file produced by the main program begins with an ASCII title line , which is typed in by the operator before the data are transferred to the tape. The title line can contain upto 80 characters (132 are allowed in DGC's default mode as allowed in 180 model) terminated by a carriage return. Pressing 'rubout' types a +, and deletes the last character typed. Pressing 'Shift-L' causes the entire line to be ignored and a new line requested. The title line should contain such information as the starting wave number, data interval, number of points, and some kind of sample identification. This makes it easy for multiple users to keep track of their data files over long periods of time.

When a file is to be input following a C$^{\downarrow}$ response, the title is typed on the teletype before the computer control requests additional information from the operator. Data are transferred, both in read and write modes, a single block of 4-NPTS bytes. Ordinate values are stored sequentially as floating point numbers, each occupying

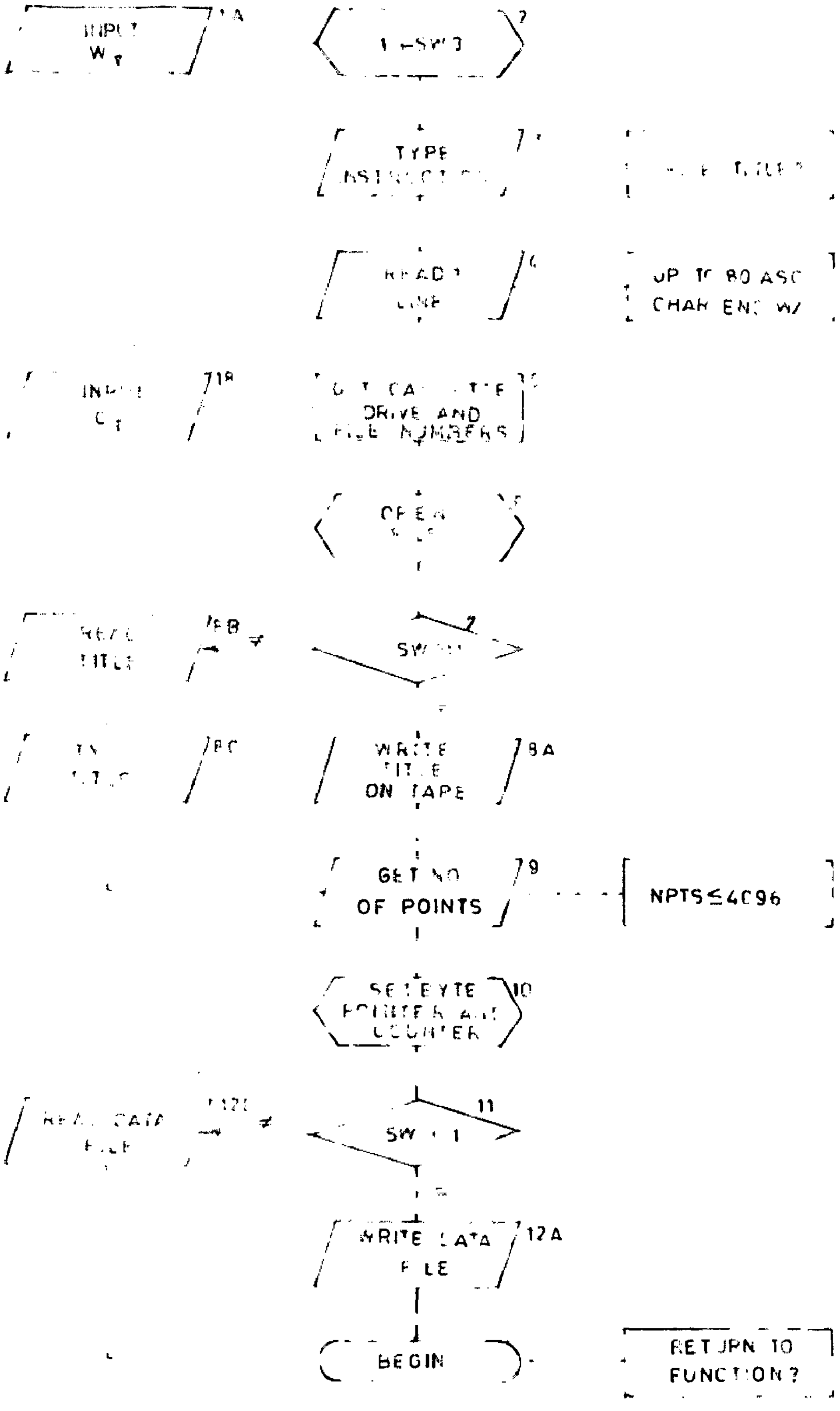


FIGURE 15 WRITE OR READ A CASSETTE TAPE FILE

two words. ASCII strings are stored one character to a byte, two bytes to a word.

5. R ↓ Ratio Two Spectra by Subtracting Absorbances

The spectrum which is minuend is placed on cassette drive (no.2) and the baseline spectrum goes on cassette drive (no.1). The computer control requests the file numbers for the two sets of data, then reads in the entire spectrum for the drive no.2. A scale factor is then requested from the operator, to be used in a scaled subtraction. Baseline ordinate values are then read in one at a time, multiplied by the scale factor, then subtracted from the minuend already stored. The difference between them replace the spectrum in memory, one point at a time.

Since SOS uses a software buffer to transfer bytes in blocks from a cassette tape, the operation of reading four bytes at a time does not cause jerky tape motion. The above routine does not conserve space, though, by not requiring storage for both spectra.

6. T↓ Type n points on Teletype.

This is a simple function, could be used to produce paper of spectra, but they are not convenient to handle. May be other sort of displays can be used instead of typing on teletype.

7. CH↓ There is one more task for the microprocessor,

can be applied here and that is to synchronize the channel of COI, as described in previous chapters. That, for example, if a filter wheel has arrangement for three channels, after one spectrum analysis is over, a certain delay must be offered to synchronize the second channel by rotating a step-motion and after then similarly third channel should be used for analysis. This will be included in Generalized Main Program.

CHAPTER - VII

MICROPROCESSOR SELECTION

The selection of a suitable microprocessor depends primarily on the particular application. Since the characteristics of the various processors are quite different, a number of factors must be considered in making a good choice. From the standpoint of the designer, the selection process involves investigating the software, hardware, and system design of a processor. Software design investigation requires examination of many features, including architecture (e.g., word length and speed) and programming flexibility. Hardware design investigation includes examining completeness (amount of support hardware necessary), and system design considers available design aids (both hardware and software).

Word length is the first feature usually considered. The determining requirement include analog resolution, computational accuracy, character length and width of parallel inputs or outputs. Also, when specifying word size, one should remember that ease of programming, not the efficiency with which the application is performed, is affected.

Other important architectural features include the number of CPU registers, type of return stock, interrupt capability, interface structures, and memory types. The number of registeres in the processor is obviously an important feature of its architocture. The registerers can reduce reference to main storage which may conserve both external memory and time.

The speed of the microprocessor has been ganged using numerous measures. Some include cycle time, state time, minimum instruction time, register-to-register addition time, and interrupt response time. But these measures should not be the only criteria used in timing estimation since they do not measure the power of the instruction set.

The degree of programming flexibility can be accessed by an examination of the processor instruction set. Multiple addressing modes conserve memory, simplify programming, and increase speed by using one-word memory reference instructions. Other important capabilities include arithmatic instructions (e.g. binary and BCD arithmatic, multiply and divide, double precision arithmatic) and logical and I/O control instructions.

The number of additional IC packages required for a microprocessor based control system is an indication of the completeness of the system. The feature of interest is how many number of additional hardware support is needed like clock generators, timing, memory, I/O control, data address buffers, multiplexer inputs, interrupt control, and power supply units.

In this case, dynamic range and precision are important considerations, since signal changes of only a few thousands of a percent can be measured. From this consideration, a 16 bit processor such as the CP,1600 (manufactured by General Instruments) or PACE (manufactured by National Semiconductors) can be a good choice. But CP1600 is having further advantages of the presence of eight program addressable registers which permit the programmer a great deal of flexibility. Also, as shown in figure, there are the four status flags for sign (0 = +), zero, overflow and carry (S,Z,OV,C), which are controlled from the results of the ALU operations. Alongwith these, 10 bits instruction word facility, instruction execution from register to register in just 2 μsec facility, with the power requirements of 5V and ±12 Volts are available. This is a 40 pin slice based on NMOS technology with the number of available softwares like cross-assembler, PL (Program library), Simulator and others. Also register holds 16 bits upto 64K words for the representation of memory or I/O locations.

The relatively high performance of the CP1600 makes the processor particularly suitable for high-speed real time applications, such as process control systems.

This CP1600 is organized around a 16-bit bidirectional data bus. Connected to the bus are the following active elements

a bidirectional I/O buffer

instruction register

ALU

eight general-purpose registers (16 bit)

The instruction register is connected to an IR Decode ROM, (Internal Read Only Memory), which decodes the bit sequence of the instructions in the instruction register to provide the appropriate control signals to the ALU, shifter and general purpose registers.

The ALU of the CP1600 operates on 8-bit words. It processes two 8 bit words from the 16 bit bidirectional data bus and produces an 8-bit resultant, which is forwarded to the shifter. The shifter may shift the resultant one bit right or left or may perform no operation at all. The shifter also affects the status flags, depending on the resultant. From the shifter, the data enters the Write Port in which it may be transferred to any one of eight-general-purpose registers. The Read Port associated with the registers can then transfer 16 bits in the form of two

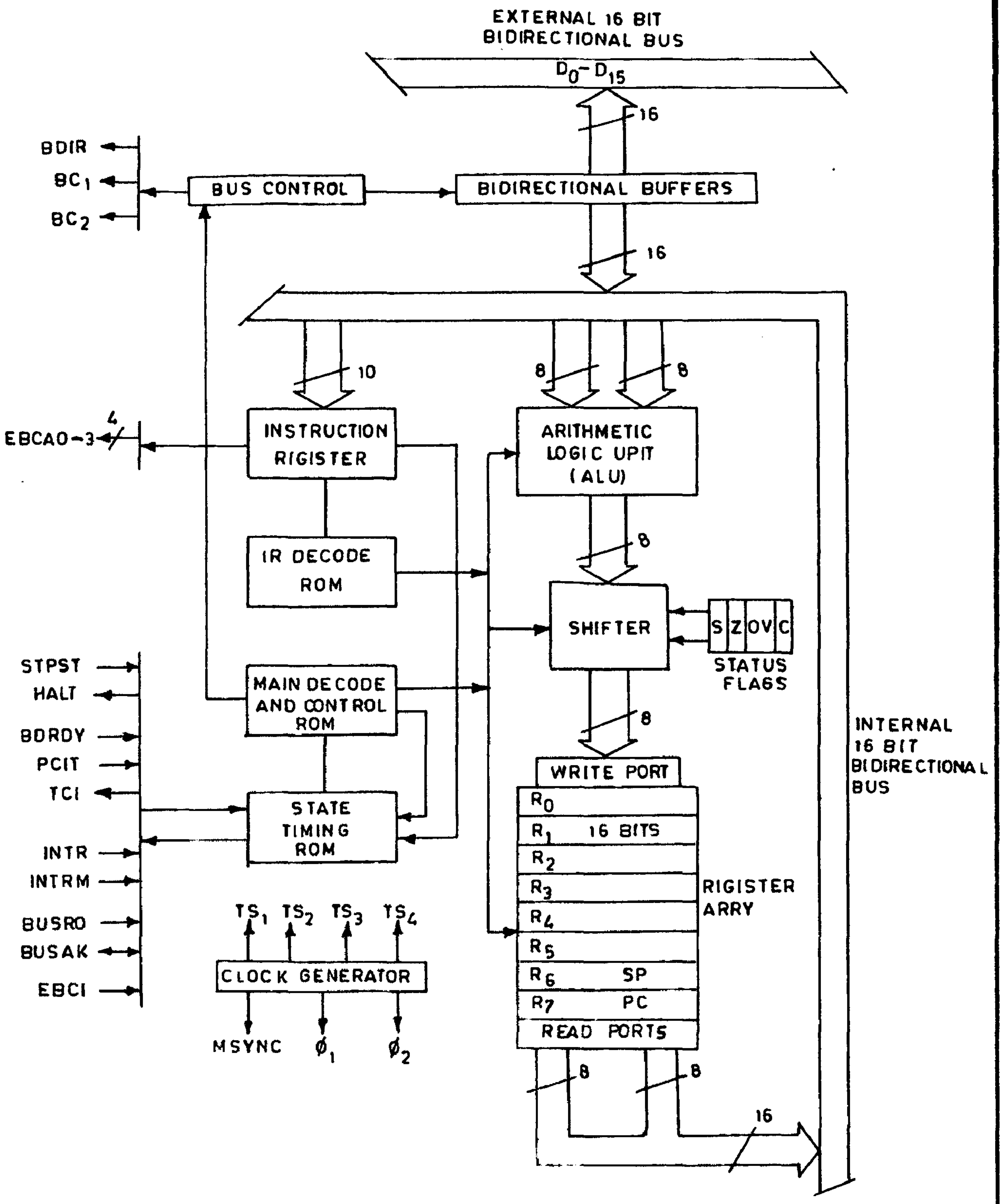


FIG. 16 GENERAL INSTRUMENTS [illegible]-1600 BLOCK DIAGRAM

8 bit words from registers to 16 bit bidirectional data bus.

CP1600 is driven by a two phase clock at 5 megahertz, thus producing a cycle time of 400 nanosecond. A state timing ROM decodes the state signals from external signals and transfers such information to the Main Decode and Control ROM.

In the CP1600 all operations to be performed are specified by 10 bit instruction words carried on the lower 10 bits of the 16 bit data bus. The structure of these codes is

Op code	RS	RD
4 bits	3 bits	3 bits

Where RS and RD represent three-bit codes specifying the source and destination registers respectively. The four op-code bits are decoded by the decoding and control (micro) ROM into 16 basic operations, 8 of which refer to external operations extending through the external data bus into the address space, to memory and peripherals (treated alike). The other eight are internal register to register instructions. The source and destination registers specified in the last two three-bit fields of the instruction refer to any of the eight addressable registers. Under certain conditions the last six bits are operation code modifiers instead of registers.

All external addresses are reached by register-addressing; that is, the register specified in the instructions contains a pointer to the location within the address space that actual holds or receives the data. Since the register holds 16 bits, up to 64 K words can be reached directly, representing memory or I/O locations.

The 16 basic instructions and their op-codes, for a generalized main program (as given earlier) are shown in table (Fig.). The total number of useful instructions that can be generated from one or more of these basic instructions, is well over 100.

Of the more than 100 instructions available in this language, only those necessary to understand this program, together with a listing of their appropriate mnemonics and meaning, is also given here. (Table).

add

cti

add.

the

dir

gene

tabl

tha

ins

lang

toge

mean

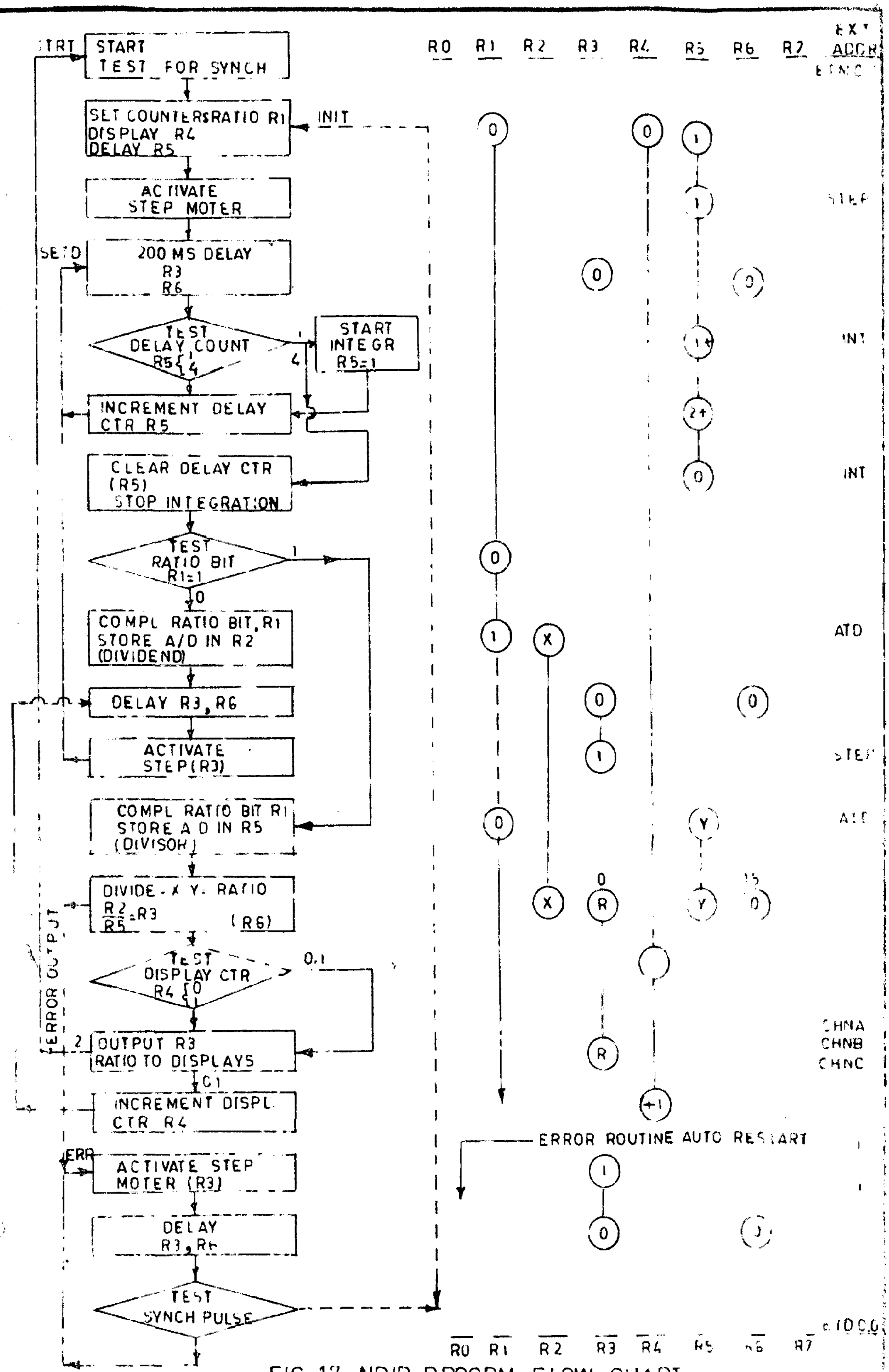


FIG. 17- NDIR PROGRM FLOW CHART

CHAPTER - VIII

GENERALIZED SOFTWARE USING CP1600

We have discussed till now the requirements of the digital control system desired for NDIR spectrometer and what functions, the main program or software of the system should perform. These functions and the corresponding software developed have been made to take care of all available features of Perkin-Elmer-180 Infrared Spectrophotometer model. But the software program required to channelize the spectrophotometer for various components of interest have not been taken into account yet.

Now let us consider a general spectrophotometer for the digital control desired and develop a generalized main program which can take care of only essential features of control, those are, to channelize spectrophotometer by moving a step-motor that drives the solid-state detector, calculations for ratio routines, storing the required data and the proper display corresponding to the proper channel of the component of interest. In this chapter, the generalized main program has been developed and it has been applied to the CP1600 microprocessor chip to work with.

Before going the entire generalized program, let us put a segment that performs the key operation of division in order to obtain the required signal ratio e_r/e_s. Since the divident is always smaller than the division, a simpler routine can be used which work only for this condition and will report an error if e_r is greater than e_s. As a further simplification of both signals must be positive, although this is not necessarily a software limitation. This particular subroutine is shown in Table. Appendic B.

The first step is to zero the register R_3, which is accomplished by the instruction CLR R_3. Next we wish to ensure that A is less than B, or that A-B is positive. This is accomplished by the second instruction CMP R5, R2 which temporarily subtracts R5 from R2 without changing either. In detail, this is a compound instruction. It requires I's complementing B, adding 1 to make the negative of 2's complement, and adding the result (-B) to A. If $A \geq B$, there is a carry bit, C = 1. If $A < B$, C = 0.

The next instruction BOC ERR (or BOHE ERR) requires a jump to the symbolic location ERR (not shown in Table.) if C = 1. This would put the system into an error routine.

If there is no branch, the program continues with MVI* 4,R6. This instruction moves the number 4, the number of places in the calculations, to the counter R6 where it will be used for control.

The instruction CRC clears the carry, after which the dividend is rotated left with the carry bit, R2-R5 is again examined with the CMP instruction and, if there is no carry (R5 is larger), there is a jump of DV2 on the BNC command. Since there is a carry in this case, the next operation, subtracting R5 from R2, is possible and

SUB R5, R2 takes place.

The next instruction shifts the carry bit into the LSB of R3 (the quotient) with SRLC S1, R3. R6 is then reduced by 1 (DECR R6), and if it is not get zero, the program jumps back to DV1 on BNZ DV1. Thus more iteration can be required in this case.

General Main Program Applied to CP1600

The main program is facilitated by preparing a more detailed flosheet (shown in Fig.17) specifying the microprocessor CP1600 activities. In the CP1600 it is worthful to include a listing if the registers in use at each step in the program. The main program shown in Table Appendix C is written in an assembly language.

Column 1, in program (shown in Table) shows descriptive amcomment and does not result in coding. The first three lines of coding define the assembly language and the memory location (relative). They symbols in the location field (column 1 to 4) are lebels which represents the memory location. EQU, a psuedo-operation which assigns a specific location to the label. For example STEP EQU 1437 assigns STEP to external word 1437. STEP is used as the address of the stepping motor drive flip flops (external peripheral).

Similarly, the address for the start-stop driver of the analog integrator (INT), the analog to digital (ATD), and the digital display channel gates CHNA, CHNB and CHNC, are defined.

The STRT routing senses the presence of the signal satisfying that the no. 6 filter is in place and the analyzer is ready to begin the first cycle. The BOS (Branch on sense) instruction refers to an addressable external multiplexer (MUX) in an experiment version of the microprocessor. The instruction BSNS can also be used and the sense signal be located with EQU.

If the signal is sensed, the program jumps to the ERR location with an unconditional branch (BUNC). This routine has the effect of rotating the filter wheels until the no.6 position is found.

When the program branches to INIT, the register RI is set to zero with the CLR instruction. R1 is selected as the counter to gate the ratio divisor and dividend signals into the correct registers when the divide routine is encountered. Similarly, R4 is chosen on the three display gate counter and is cleared R5 is used to count increments of the time delay (200 m sec) encountered in several places in the sequence and is also used for further tasks. In this case, it is incremented (INCR) to contain 1, and then the 1 is moved out to the location STEP to actuate the step motor. In this way no.1 filter is put into position (This is the reference filter of component A, containing

100% COI, therefore it will give the lowest signal and should be the divident in the ratio calculation).

The next coding is a JST (jump and store return) to the location SETD, which is the 200 m sec subroutine. JST stores the address of the next instruction in sequence so that, when the subroutine has been completed, one can return to the normal sequence. The address is stored in R6. After subroutine SETD has been completed the contents of R6 will be transferred to R7, which is the program counter in the microprocessor. The program will return to the address in the program counter.

The 200 m sec delay operates by moving a number (20000) into R3, subtracting 1, and branching back to RPET until the number is reduced to zero (BNZ). The NOP (no operation adjusts the timing of the repeating cycle to approximately 10 μ sec. At the end of the countdown, the program counter (R7) returns to main program.

At the end of this 200 m sec delay the '1' that is still in R5 is moved out to the location INT, starting the external integrator.

The integrator is to be turned off after a delay of 600 m sec or three increments of 200 m sec. This is accomplished by comparing the contents of the time delay counter R5 with the number 3. If it is smaller, 1 is added to R5 and we branch back to DLY, which sends us to subroutine for another 200 m sec. When the 600 m sec are up, we jump to location DEL8 , which clears R5 and stops the integrator by moving 0 in R5 to INT.

The next step is to test the last bit of R1, we reserved as a ratio-switch. By shifting the last bit (LSB) right, into the carry, we branch if it already contains a '1' to the place to store the sense signal. If the counter LSB is 0, however, R1 is complemented (COM) and the output of the ATD, recognized as the digitized integral of the reference signal, is placed in R2, the dividend location.

Next, we have a delay, an actuation of the step motor (MVOV STEP R3), and branch back to the delay and integration cycle (DLY). This time, the 1 in the LSB of R1 sends us to LRS1 when we store the ATD signal (identified as the sense signal) in R5.

We are then ready to enter the divide routine, which has already been explained. On completion of this segment, R3 contains the quotient. R4 is chosen as the display counter and is cleared to zero at the start of program. It is tested and if it still contains zero, the quotient is moved out to the display for component A, CHNA. If it contains 1, then CHNB is chosen and if more than 1 CHNC is chosen. This is accomplished by coding following division. At the end of the division, R6 contains a 0. If R4 also contains a 0, the CMP R6, R4 instruction bypasses the branch and moves R3 out to CHNA and returns the program to next filter wheel rotation and subsequent integration (after increasing the R4 counter by 1). Otherwise it branches to ONE where R6 is increased by 1 and the comparison repeated, this time moving R3's contents to CHNB. If R4 contains more than 1 (the third time around), we place R3's contents in CHNC and return to STRT, to repeat the entire process with a move to filter no.1.

CHAPTER - IX

HARDWARE SUPPORT

As a general purpose device, the microprocessor has made a large impact on many areas of engineering. The second generation of microprocessors includes some very powerful general-purpose processors like CP1600, as well as highly sophisticated special function devices. Hardware development kits and systems, software assemblers and compilers, and a wealth of-applications information have been produced, especially for the earlier market entries such as Intel 8080, Motorola 6800, and CP1600.

We have already discussed the contents and facilities provided by CP1600 microprocessor CPU chip. The system architecture is not very difficult and fairly complete set of compatible devices is available. These include a random-access memory chip (RAM), read-only-memory(ROM), a peripheral interface adapter (PIA), and an asynchronous communication interface (ASCI). It is possible to construct a simple system requiring very few interface circuits with the CP1600 chip family; a system may consists of a central processor (CPU), peripheral interface adapters, read-only-memories (ROM) for program storage, random-access-memories (ROM's) for data storage and power supplies and others. Interface between the various 'family' chips, when it is must, can be used. But is not very essential, if fewer chips (7 to 9) are used.

The central processing unit is the heart of the system, and normally controls all systems elements. The CPU accesses instructions stored in memory and manipulates data found in memory or its internal registers. It is capable of making decisions and altering program flow accordingly. It performs subroutines and can process interrupts/requests. Provisions are made to allow direct memory access by other devices, single instruction execution, and input/output interfacing; however somewhere external circuitry is required. Internally, the central processor is relatively straight-forward; instructions are contained in instruction registers (IR). Arithmatic operations and data manipulations are performed by the Arithmatic and Logic Unit (ALU) and the accumulators. Flags are contained in the register (as described earlier). Indexed addressing is made possible by index register (X).

Input/output (I/O) operations are greately simplified by the PIA chip, designed to interface directly with the CPU. Bidirectional bit programmable ports are provided with the associated control lines. The behaviour of the I/O lines and the action of the control lines are determined by the contents of two registers associated with each port; the data direction register and the control register. All PIA registers are accessed by the CPU in the same fashion as other memory locations, therefore, all arithmatic, logical and transfer instructions that may be used

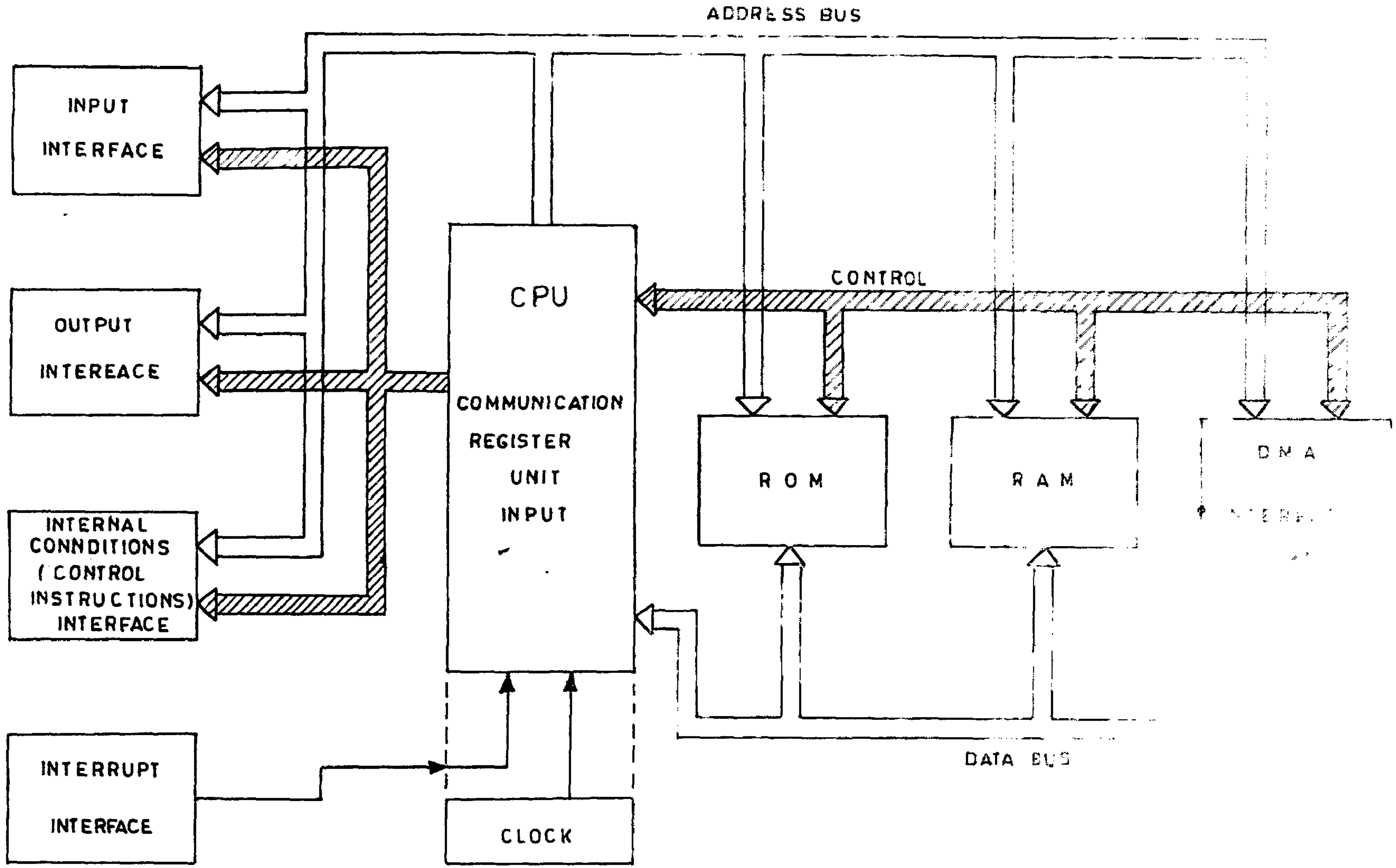


FIG. 18 HARDWARE SUPPORT

with memory data, may be used with I/O data. Separate input/output instructions are not generally required in such cases.

A very general and basic (PIA's and other details not whown) in Fig.18. Later step by step, discussing each requirement, we shall develop the final hardware system.

Addressing Structures

The ability to access information organized in a random fashion implies a unique name or address associated with each datum. The incorporation of an addressing subsystem capable of fetching instructions from memory provides the primary dist nction between microprocessors and previous integrated circuits, the latter requiring external address mechanism to access instructions or provide control signals.

Consideration of addressing subsystem can be fabricated by treating as independent processes -

(1) the formation of an address,

and (2) the presentation of this address to the target subsystem.

One simple address subsystem employed in the CP1600 is given in Fig.19. All address information appears on the data bus at the begining of each cycle and is latched for presentation to memory.

The timing signals are not available directly off the chip. A straightforward extension of the use of the

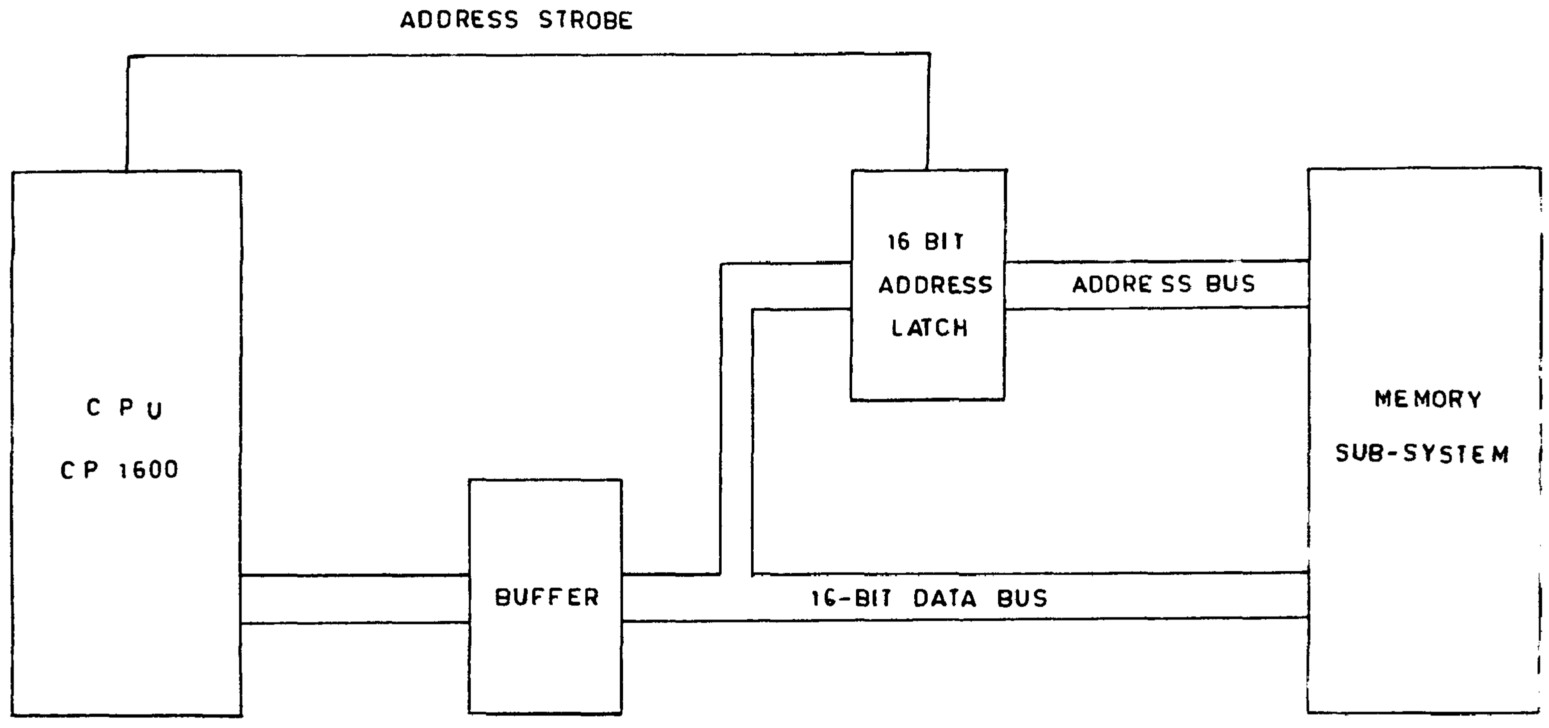


FIG. 19 ADDRESS SUBSYSTEM

data bus for supplying address information occurs in the CP1600 microprocessor. The use of a separate 16-bit address bus along with the 16-bit data bus in a 40 pin package is too costly, and an off-chip address latch must be provided. The full 16-bit address appears on the data bus at the begining of each cycle and must be latched at this time. The basic scheme is shown in Fig.20 . Either edge of the active LOW address strobe, that is available at one of the 40 pins, can be used to latch the address at the proper-time. The interpretation of the data appearing on the data-bus is always time-dependent in multiplexed-system, and the different time slots are shown for CP1600 microprocessor, relative to the system clock. All transactions consist of an address output internal followed by a data transfer interval. The presence of the address strobe timing pulse coming out of the processor reduces the design of this subsystem to the level.

Similar strobes are available for data flows to and from the addressed subsystem. The INPUT DATA STROBE (IDS) can be used to gate data from the target sybsystem into the CPV. The data must be valid when the IDS goes low. Set up and hold times are indicated in figure . The trailing edge of the OUTPUT DATA STROBE (ODS) can be used directly to latch data into the addressed target location. A seperate ready line extend the basic cycle for an integral number of clock-edges, giving the CPU - the ability to WAIT for the slow data sources.

Direct Addressing (Direct Memory Address)

An interesting and efficient approach to direct addressing is found in the CP1600. The CP1600 has a 16-line data bus over which all addresses are multiplexed.

The source of the address bite supplied to memory is termed the Memory Address Register (MAR) and can be located within the CPU, external to the CPU, or a combination of the low. The simplest conceptual addressing scheme is direct addressing in which the location of the operand is contained in the instruction proper. The upcode specifies the operation to be performed and the direct address specifies the location of the target, usually either a source or a destination register. The address is normally fetched into the CPU alongwith the instruction, and in the most straightforward case is simply loaded into the MAR. If the fetched address is to be modified in any fashion, this modification is performed in the CPU and the modified address.

In CP1600, the program counter is output at the begining of each instruction fetch cycle, and the opcode is fetched into the CPU as shown in Fig. 21. The direct address is contained in the 2-byte word following the opcode, and this address is used to select the operand. The CP1600 does no address modification and utilizes and

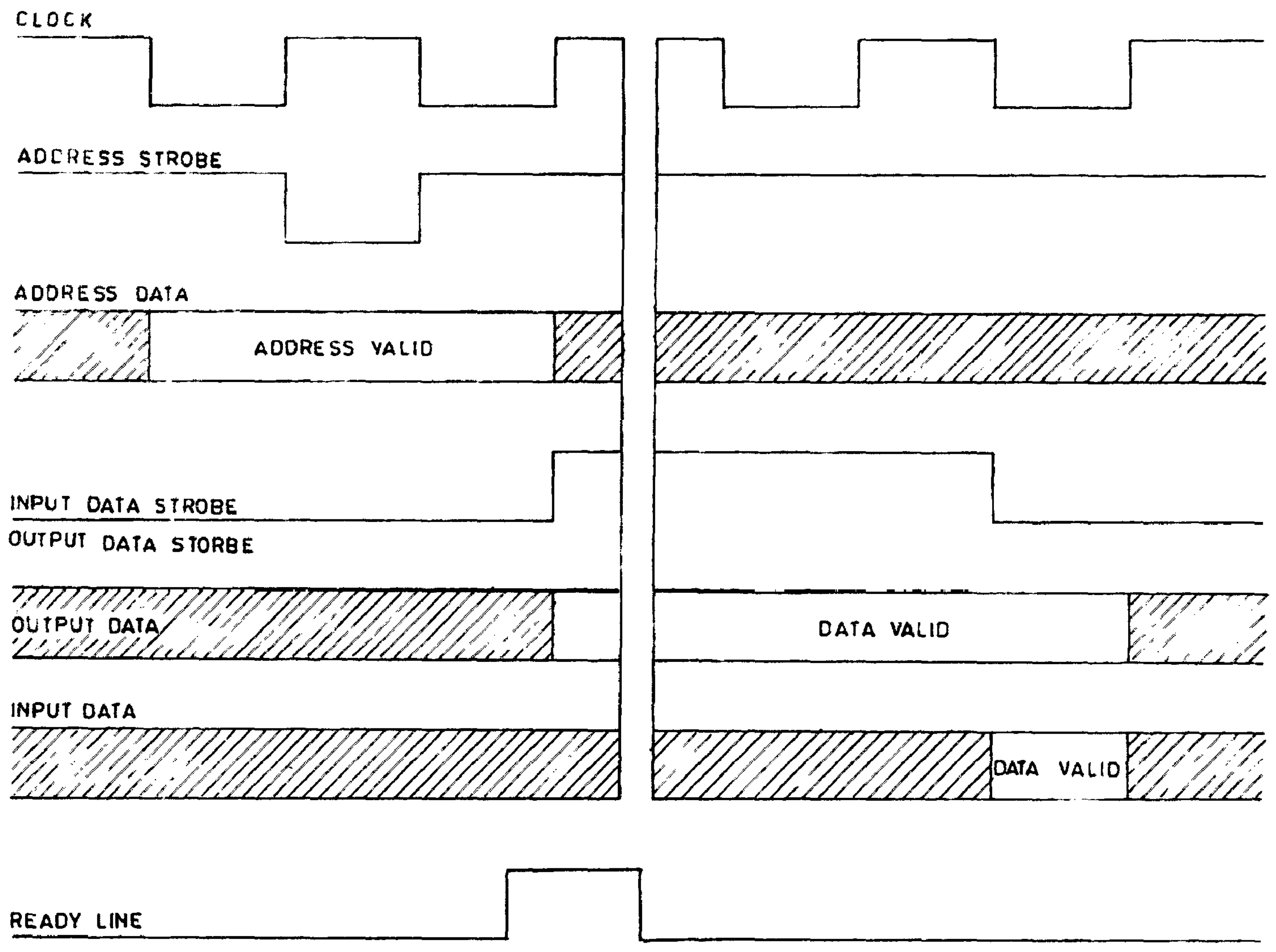


FIG. 20 TIMING SIGNALS FOR MULTIPLEXED ADDRESSING

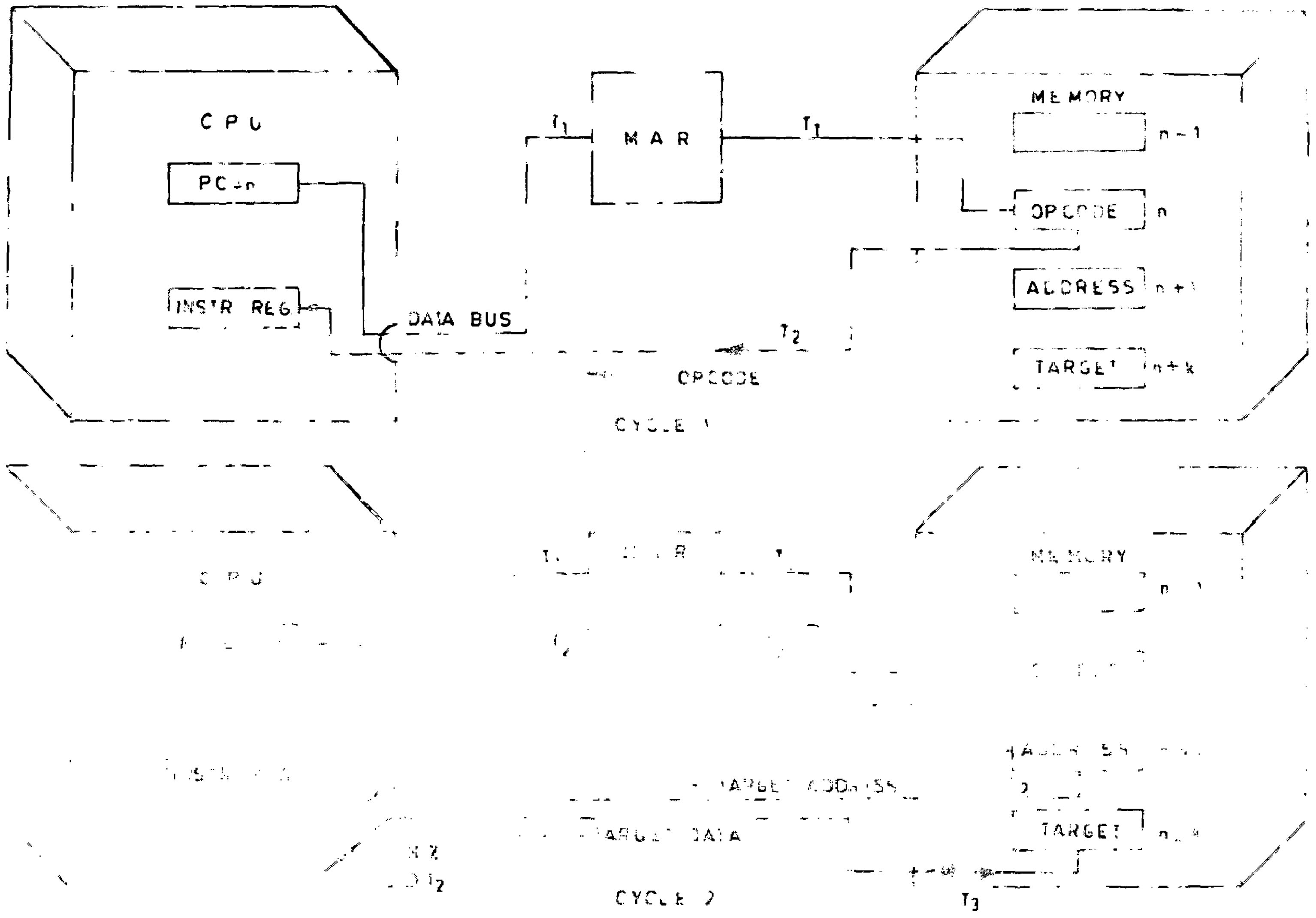


FIG 21 INDIRECT ADDRESSING

external memory address register, thus eliminating any need to fetch the direct address into the CPU. Rather than simply imputting the 16-bit address on the address-fetch cycle and then outputting it on the following data fetch (or data store) cycle, the same result is achieved via the use of an (encoded) control signal ($BC_1 = 1$ / $BC_2 = \emptyset$/ $BDIR = \emptyset$). This state signal is used to latch the address data appearing on the data bus from memory into the memory address register, completely bypassing the CPU. The signal is activated during all instructions that specify direct addressing. The CP1600 data bus is placed in the HIGH impedance state during this portion of the cycle.

During cycle -1, the program counter is used to fetch the opcode stored at location M in memory. This is decoded and determined to be a direct-addressing instruction. PC +1 is then sent to the MAR and the control lines are activated. The CPU data bus floats, the address contained in location M+1 is placed on the memory data bus and latched into the MAR. This address is then used to address the target location.

The timing diagram for the CP-1600 direct addressing subsystem shown in Fig. 22 , relates the condition if the state signals (used to control the data bus flow) to the data bus and the clock. These three signal lines (BC1, BC2, BDIR) can be directly decoded, using a 1 of 8

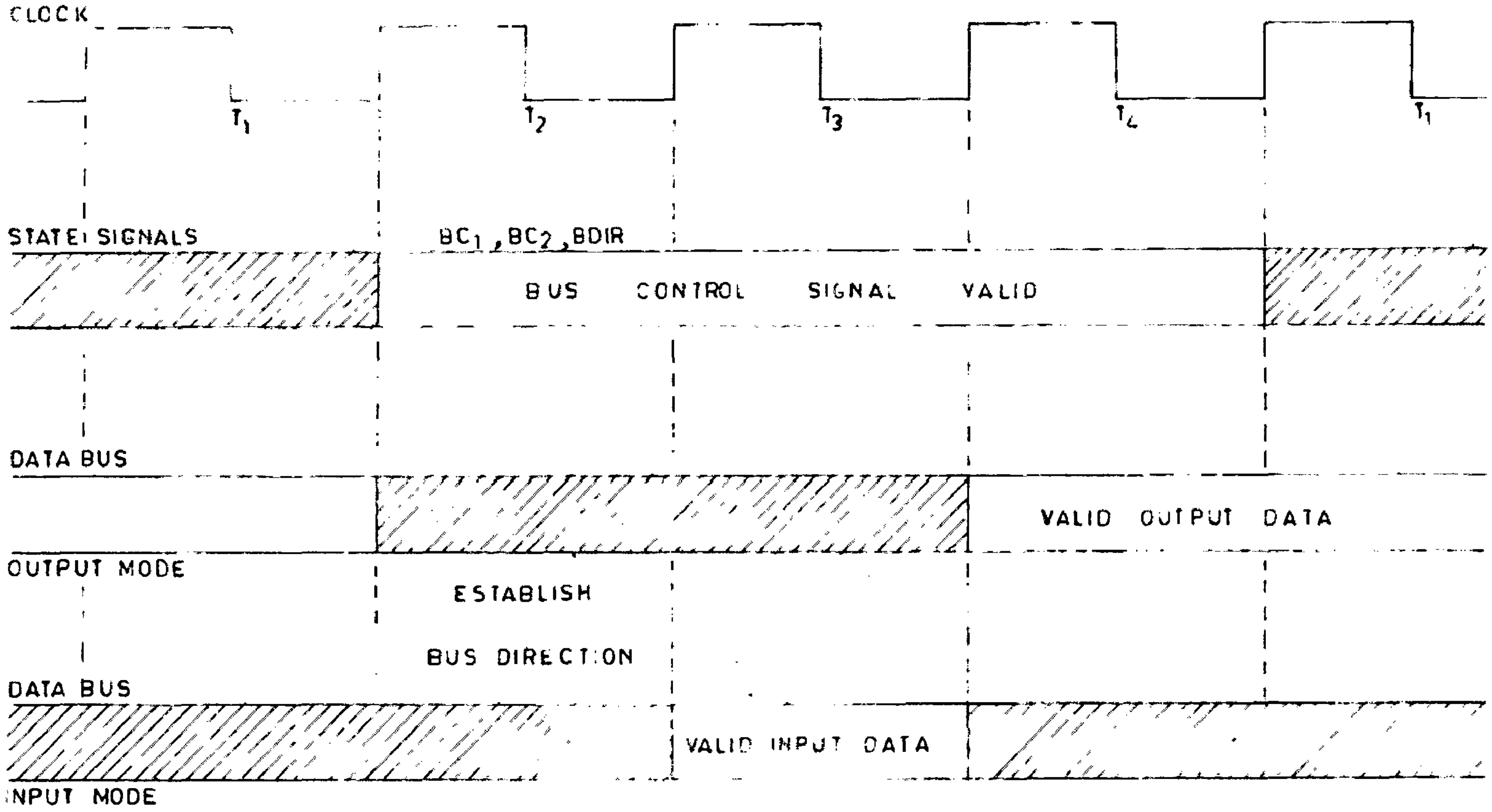


FIG. 22 DIRECT ADDRESSING TIMING DIAGRAM FOR CP 1600

decoder. They are treated in next Fig. 23. Also interrupt/request structures are needed which shall be discussed later. The CP-1600 processes a READY line for extending data fetch cycle with slow memories, as discussed earlier and WAIT state stops the interval clocks and cannot be indefinite duration because the dynamic CPU register will lose data in 40 µ sec.

The CP-1600 utilizes a 16-bit wide data bus over which all addresses and data words are bussed. The CPU utilizes memory mapped I/O and, therefore, possesses neither input nor output instructions. Communication with external devices, including memory, is channelled by three control lines, BUS CONTROL 1 (BC1), BUS CONTROL 2 (BC2) and BUS DIRECTION (BDIR). The use of these signals to latch direct address data from memory into the memory address register (MAR) was described earlier. During I/O operation, three of the eight combinations of primary interest are

BUS TO ADDRESS REGISTER — used to load data on bus into external register for addressing memory or Input/output device.

DATA WRITE STROBE — used as a write enable for the addressed memory or Input/output device.

DATA TO BUS — used to gate data from memory or any peripheral device onto the bus for input to the CPU.

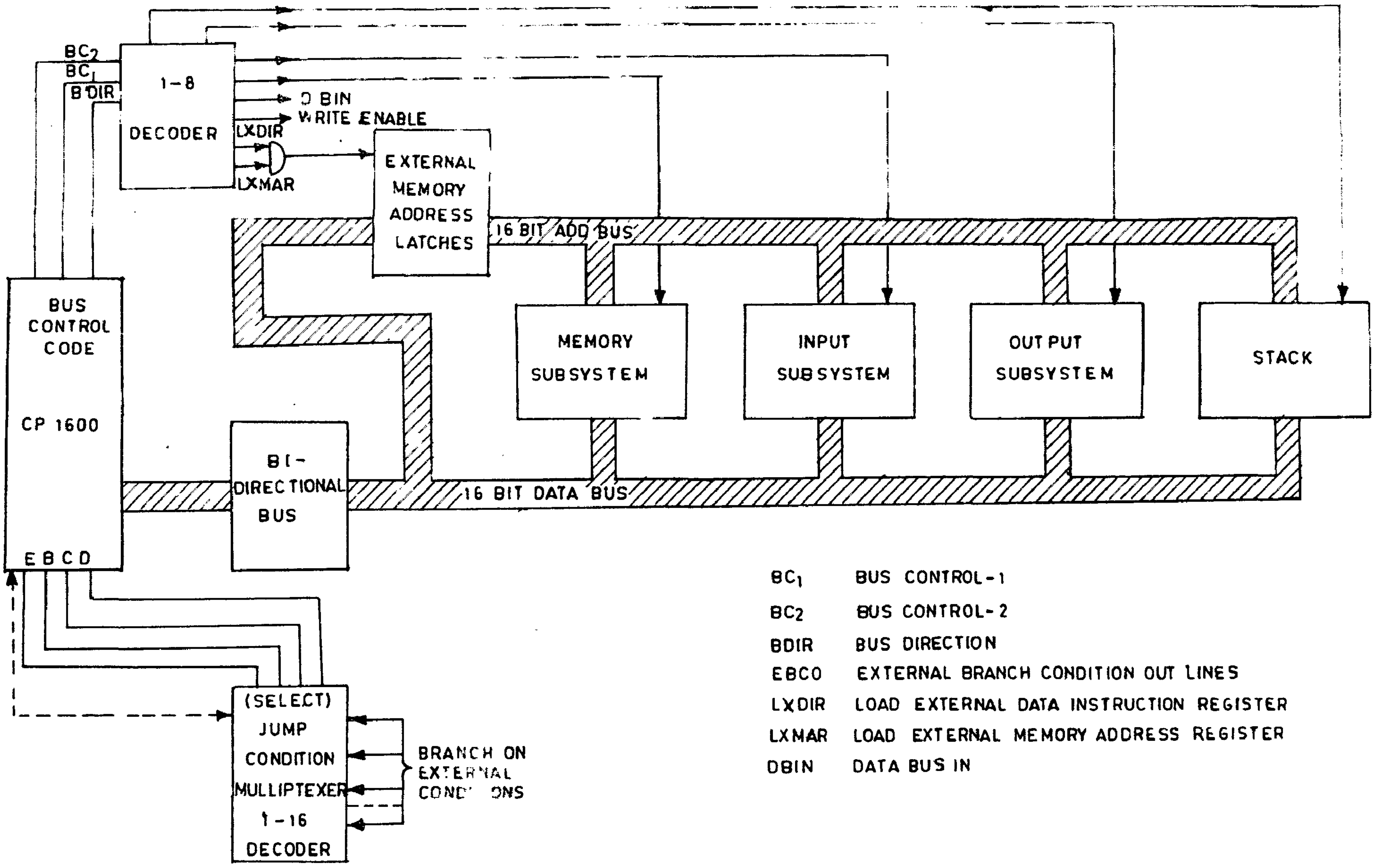
Several of the remaining codes are used in conjunction with the interrupt system , which shall be discussed later.

Test External Conditions

The CR 1600 is able to test both internal and external conditions via the use of Branch-on-condition instructions. The CP-1600 possesses a number of specific instructions that test internal status and a Branch-on-External Conditions (BEXT) that selects 1 of 16 external conditions to be tested. Since this is too many conditions to enter the CPU on individual lines, the JUMP CONDITION MULTIPLEXER has been moved outside the CPU and is internal 16-bit data paths and registers, the instruction decoder is only 10 bits wide. The six bits accompanying the opcode in a 16-bit memory are unused. The displacement address is in the second word and can usefully be 16 bits wide.

BEXT OPCODE				EXT.COND.CODE				
1	0	0	0	1	b3	b2	b1	bo

The four least significant bits of the BEXT instructions are buffered and presented to the External Branch Condition Outlines (EBCO 1-4). The selected input is tested and a branch occurs if the condition is true. The basic circuitry necessary to decode the status and channel information is shown in Fig. 23.



BC_1	BUS CONTROL-1
BC_2	BUS CONTROL-2
BDIR	BUS DIRECTION
EBCO	EXTERNAL BRANCH CONDITION OUT LINES
LXDIR	LOAD EXTERNAL DATA INSTRUCTION REGISTER
LXMAR	LOAD EXTERNAL MEMORY ADDRESS REGISTER
DBIN	DATA BUS IN

FIG. . 1 .S'C CIRCUITRY WITH DECODERS

Interrupt Structure

Information is a measure of 'surprise', and in many systems that equates as much to 'when' as to 'how much'. To find out 'when' an event occurs, it is possible to input and test status bits that are set by the occurances. For a number of systems this may require almost continuous sampling, while only a relatively few samples return much information. The machine can do no useful work while sampling and, thus, is inefficiently utilized. By allowing the events of interest to signal for the machine's 'attention', the efficiency can be vastly improved. Thus interrupt structures allow " event-driven" systems in which the concept of temporal containing has little relevance.

The ability to respond " instantly" to events allows processors to be shared by one or more processes if a means can be found to handle " simultaneous" events i.e. those events occuring within one basic system cycle - normally the current instruction cycle. The usual means of handling simultaneous " interrupt requests" is by embedding the " concurrent" processes within a priority structure.

The interrupt structure has been designed with one goal in mind - to share one CPU efficiently between several " concurrent" processes. This is accomplished via following procedure -

1. save state of current process
2. identify device requesting service
3. transfer control of CPU to this device
4. upon completion of service, restore state and
5. transfer control of CPU back to interrupted process.

The scheme is shown diagrammatrically in Fig.25

The elementry process that is crucial to an interrupt structure is the CALL/RETURN Transfer-of-control process. The subroutine is a sequence of code that is executed upon the invocation of its name and that returns control to the calling sequence upon completeing its execution. An interrupt process can be thought of as an unexpected or surprise subroutine call.

In a program the invocation is accomplished by inserting a call instruction at a known position in the instruction sequence. During interrupt processes, the invocation will occur at unknown positions in the control sequence. Thus, provision must be made for saving the return address in a known location for later retrieval. The complete sequence is composed of a set of elementry sequences, or control strings, that can be edited by real world systems to adapt to local conditions.

In the following, an INTERRUPT request signal has become active during the execution of an instruction fetched from memory at location one hundred. The program counter has been adjusted to point to the next instruction, and the current instruction completes its execution. The instruction can be one to five cycles in length where each cycle consists of these states.

T1 place address on address bus, status on data bus

T2 sample RDY, HOLD, HATAL (increment PL if used in T1) (TWAIT) optional : entered if $\overline{\text{RDY}}$ during T2.

T3 DBIN or $\overline{\text{WR}}$ timing signals for data flow to/ from CPU

$\left.\begin{array}{l}\text{T4}\\ \text{T5}\end{array}\right\}$ optional ; if needed for internal processing.

The completion of each instruction creation is followed by the FETCH cycle that brings in the next instruction. If, prior to the fetch, the interrupt system is enabled and an interrupt request has been presented to the system, the following event occur :

An internal interrupt F/F is set by the presence of an interrupt request. This F/F is used to

1. disable the interrupt system
2. set the INTA status bit
3. reset the MEMRead status bit
4. inhibit the store of the incremented program counter.

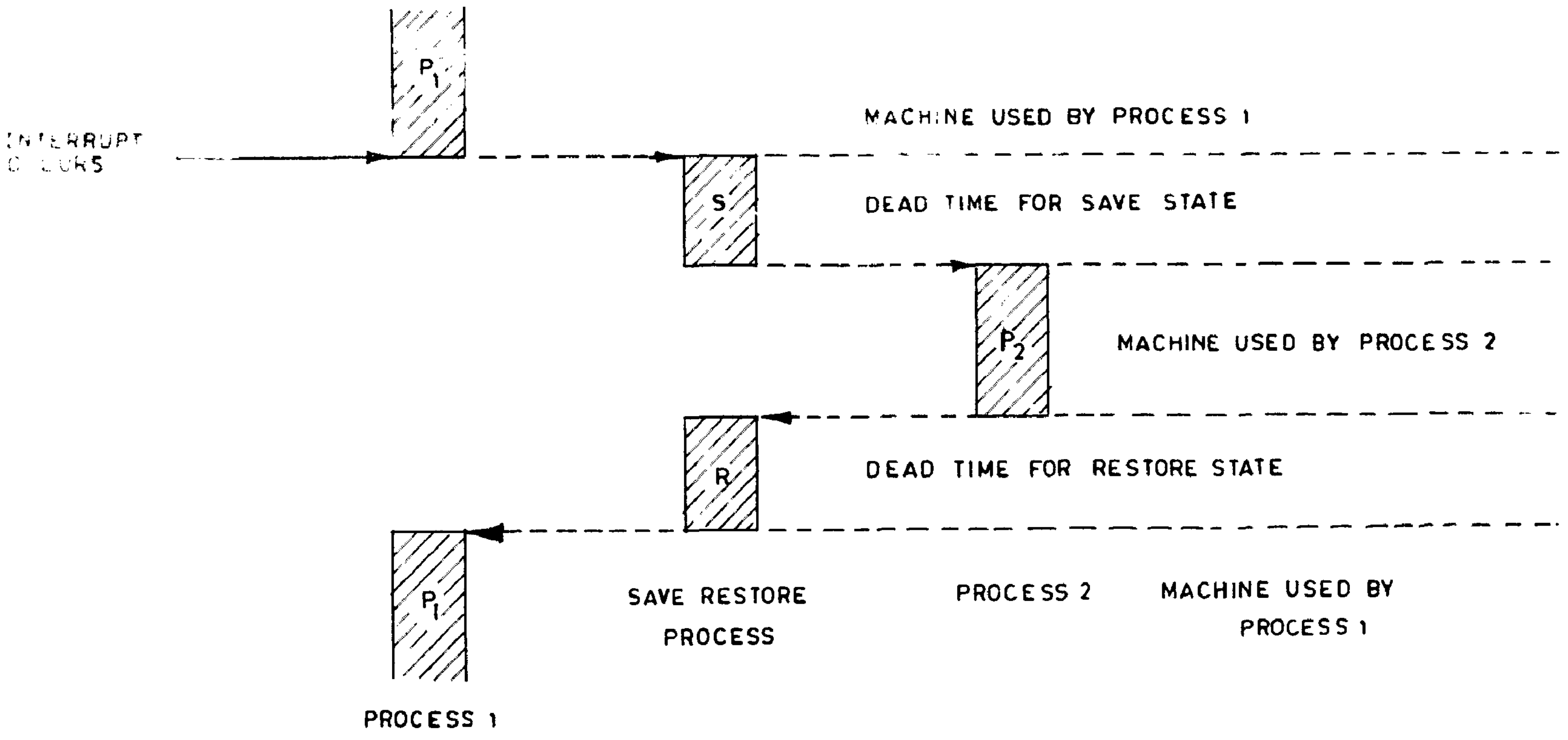


FIG. 24 THE INTERRUPT OCCURANCE

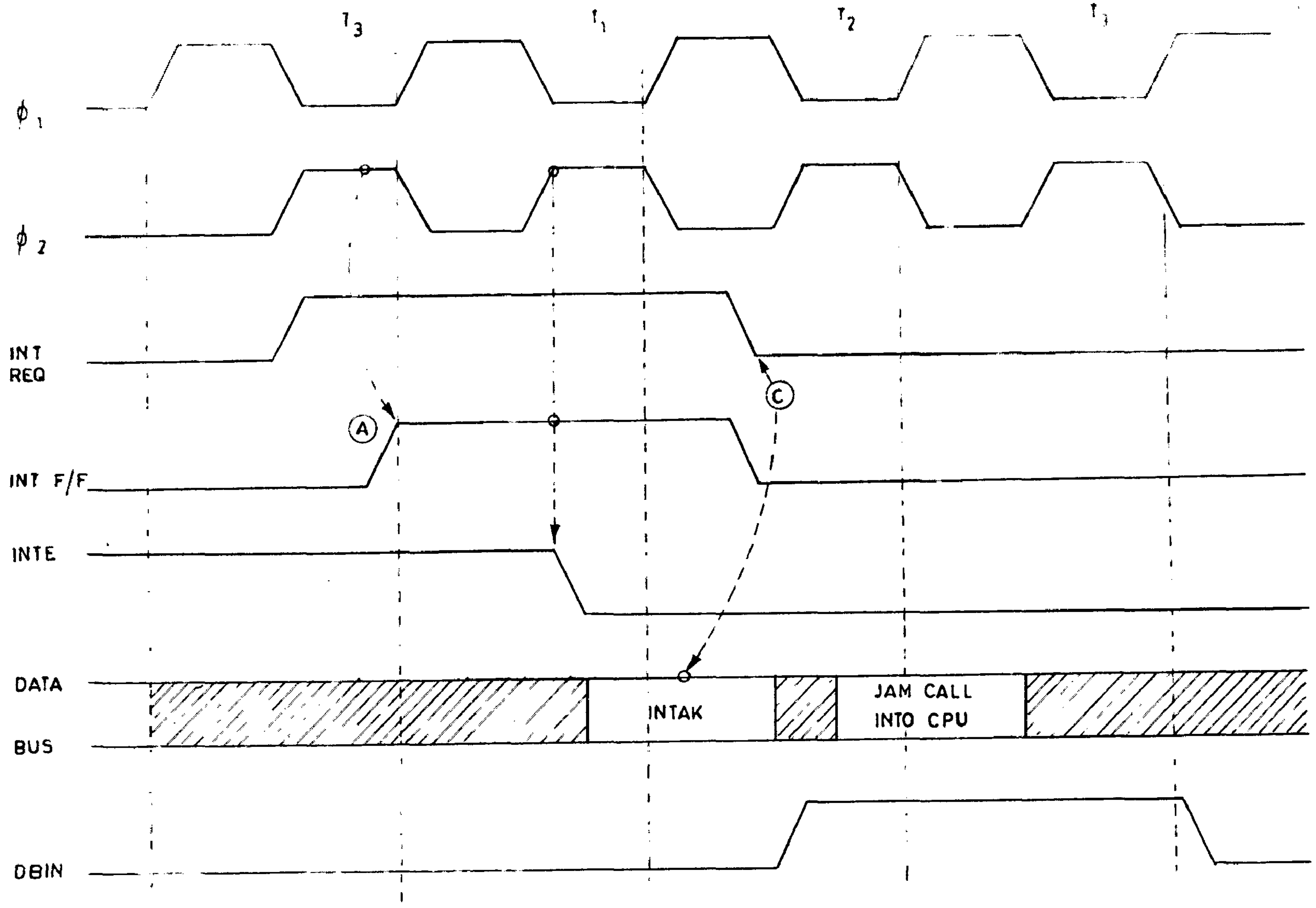


FIG.25 STATE TIMINING DIAGRAM OF A GENERAL INTERRUPT OCCURANCE

The sequence is shown in Fig.

The timing diagram indicates that the trailing edge of $\emptyset_2$ is used to set the internal INT flip-flop, (A), if the request is present for the specified setup time. The output of the flip/flop then gates a reset pulse to the INTE flip-flop at the next $\emptyset_2$ rise time, (B), thus disabling the system for further interrupts. Not shown, during T2, is an internal pulse that inhibits the store of the incremented program counter. The INTA status bit must then be employed by the designer the remove the (acknowledge) interrupt request, (C), Finally the same status bit must be used to select a subsystem that will apply a call to be jammed onto the system data bus.

Interrupt Structure Applied to CP1600

Each device (000 thru 111) can have a unique address associated with its service routine, and the hardware automatically provides a 1-byte cell instruction that causes transfer of control to this address. An interrupt system in which the hardware supplies a separate address for each interrupting device is called a VECTORed interrupt structure as opposed to the POLLed structure in which all devices trap to the same address, and device identification and conflict resolution are accomplished in software - VECTORed interrupt system provide the fastest possible interrupt serving because no time is wasted polling status bits.

Since the highest priority requests override all others, there must be a means of individually removing each request as if is serviced, so that lower priority requests can be seen. The lower requests must, therefore, remain active until their time comes. The 8214 Priority Interrupt Control Unit does not possess individual latches or clear lines, therefore it is necessary to precede the inputs with the flip-flops that possess individual set and reset lines; as shown in Fig. 26 The latches are set by the request from the appropriate device and cleared under software control by the service routine for the device. There is also a comparator override line that is shown in the 8214 diagram. This line is ORed with the comparator output so the device zero can generate requests when the threshold is completely open.

The decoder allows the use of three or more bits, if required for clearing the individual request latches. This can be accomplished at the same time that the threshold is set, and the decoder is then enabled by the same decoded I/O address as is used for selecting the current status latch. The WRITE ($\overline{WR}$) pulse is ANDed from the accumulator. The software completes the remaining part of the interrupt processing.

The OPREQ informs external devices that a new operation is beginning and that all CPU signal lines are valid. The INTAK is used to insert the second (address)

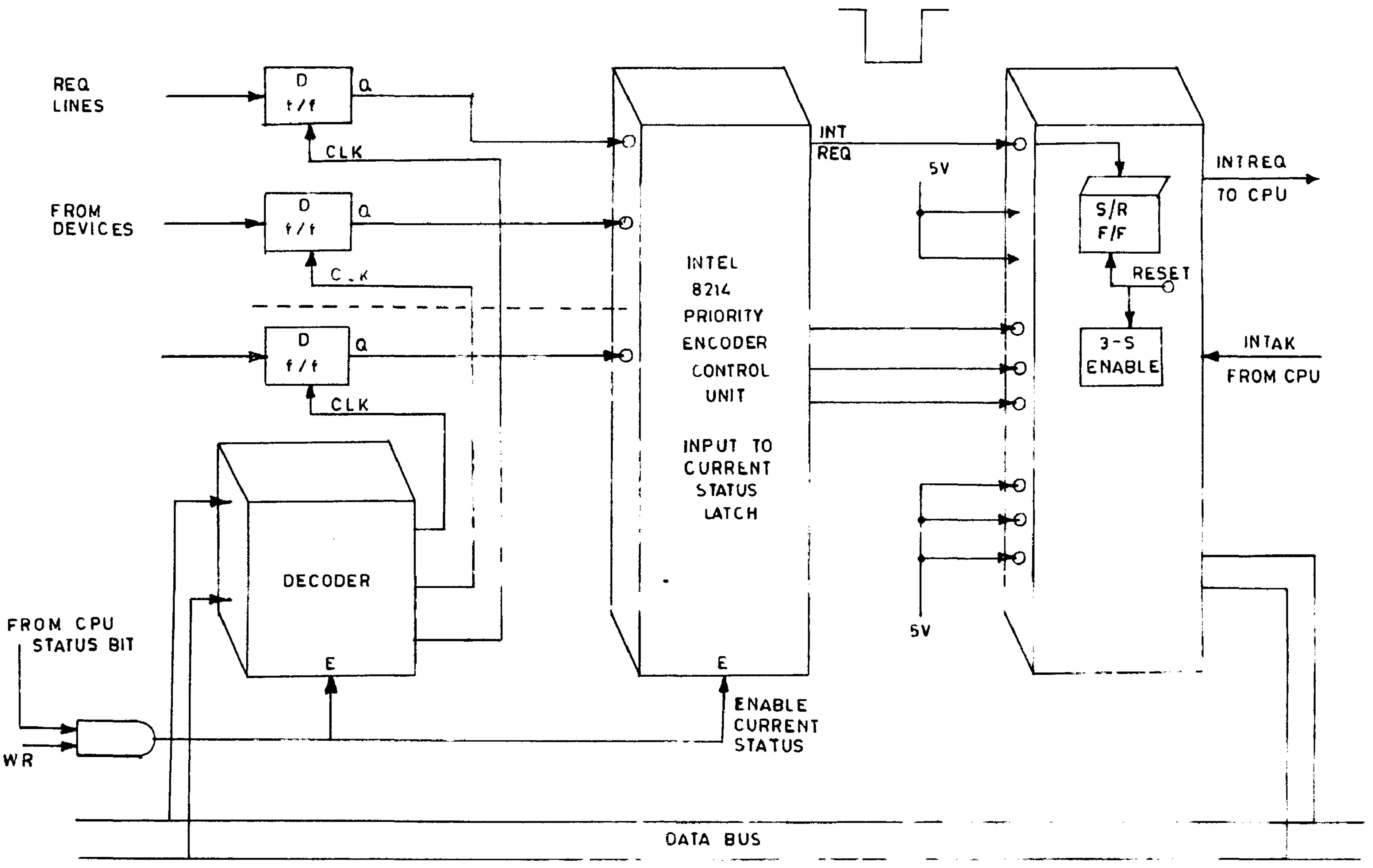


FIG. 26 PRIORITY ENCODER SCHEME

byte of ZBSR instruction into the CPU. Thus the interrupting device can supply the address vector to the service subroutine. The current program counter contents are stored on an internal eight-deep return stack. This stack is not available for storing the CPU state vector. The INTERUPT INHIBIT bit in the program status word can be reset under software control during the execution at the end of the routine. The INTAK signal should be used to clear the INTREQ signal. The timing diagram is shown in Fig. 28.

In the system shown in Fig. 27 the use of the priority encoder guarantees that the highest priority device, as determined by a rigid priority structure, will gain entry to its service routine at any time.

Thus CP1600 acknowledges an interrupt at the end of the instruction currently being executed and jams a self-identifying vector appearing on the data bus into the program counter. Two decoded state signals have been finally used to control the interrupt process : INTAK enables the priority resolution subsystems and signals the begining of the interrupt sequence. The Interrupt Address to Bus signal (IAB) is used to gate the interrupt vector onto the 16-bit data bus. The address is loaded into the program counter and causes a transfer of control to the interrupt service routine after the current program

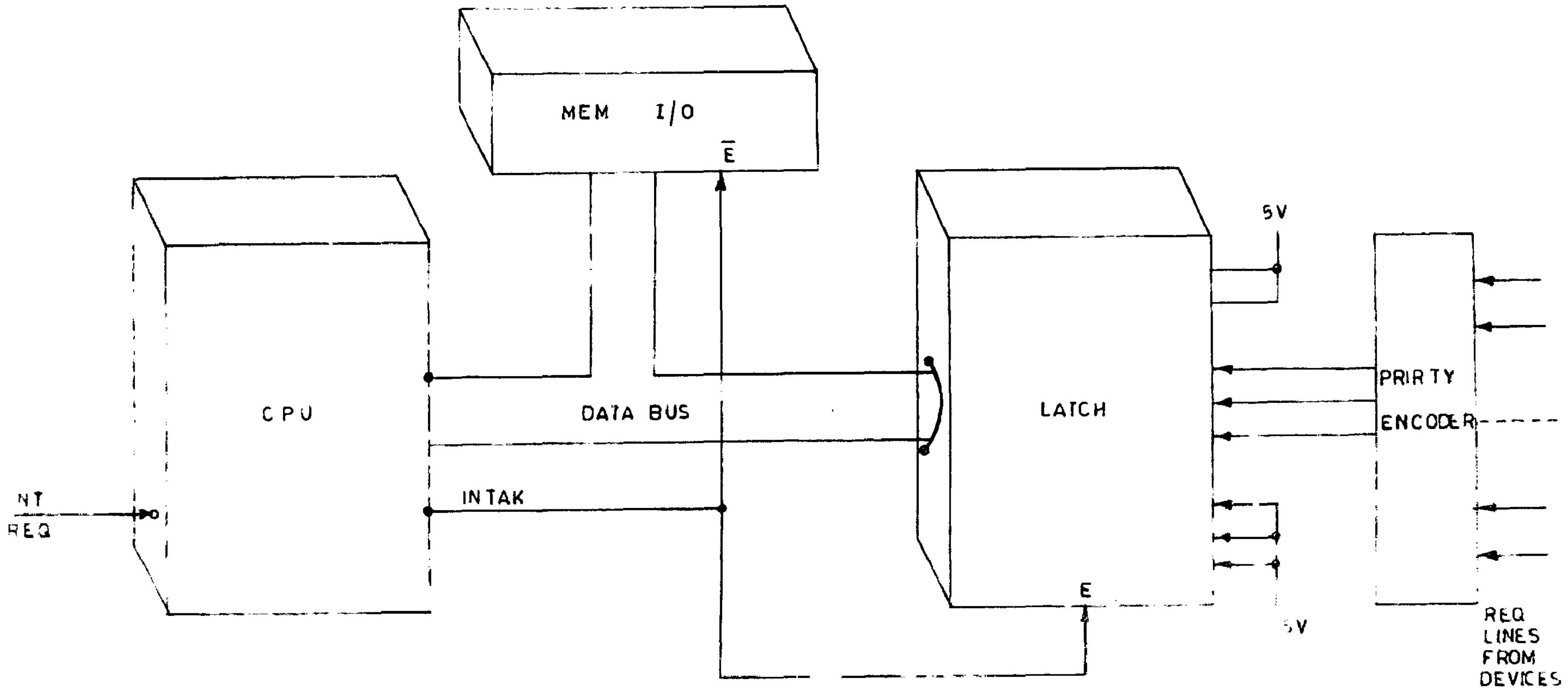


FIG 27 INTERRUPT STRUCTURE APPLIED TO CP 1600

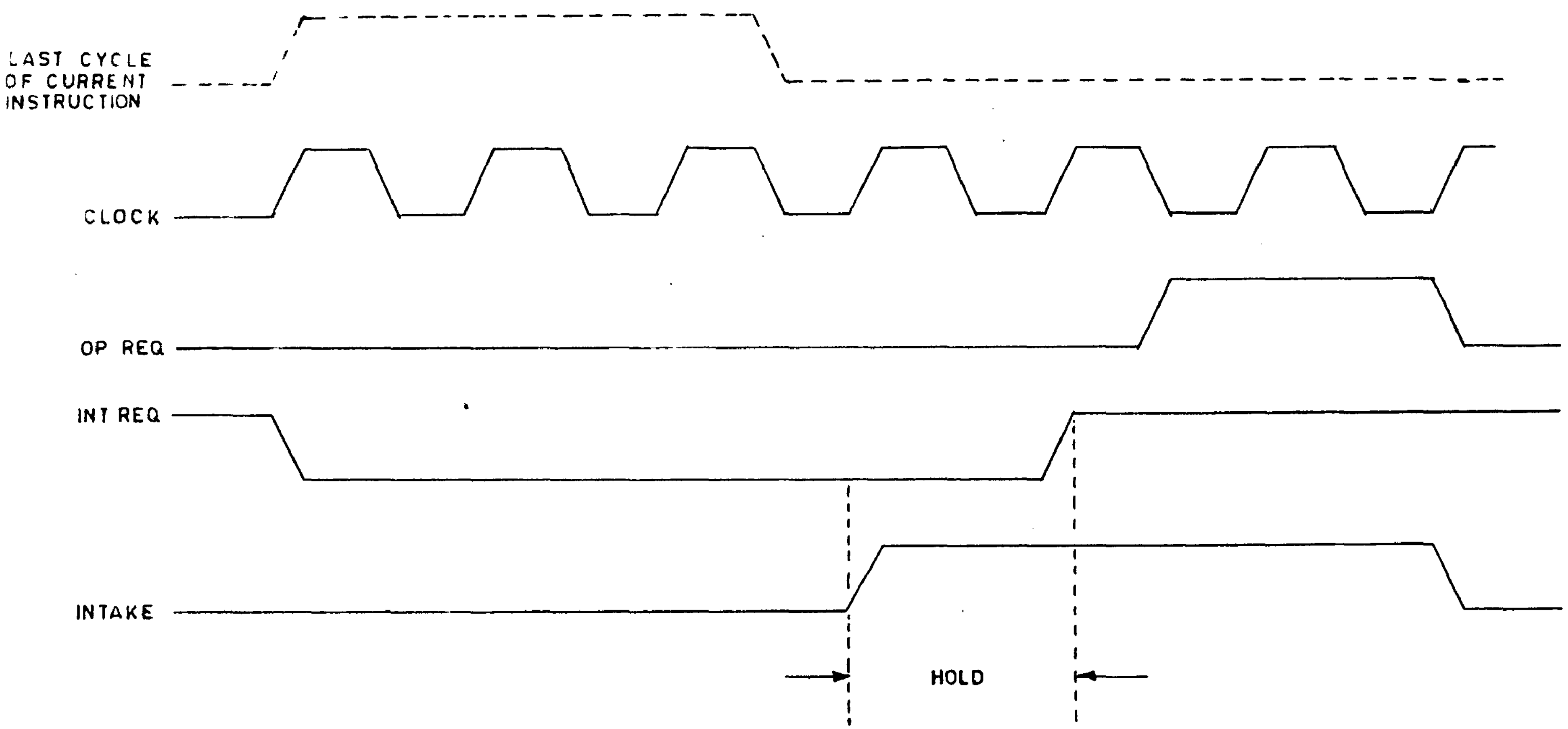
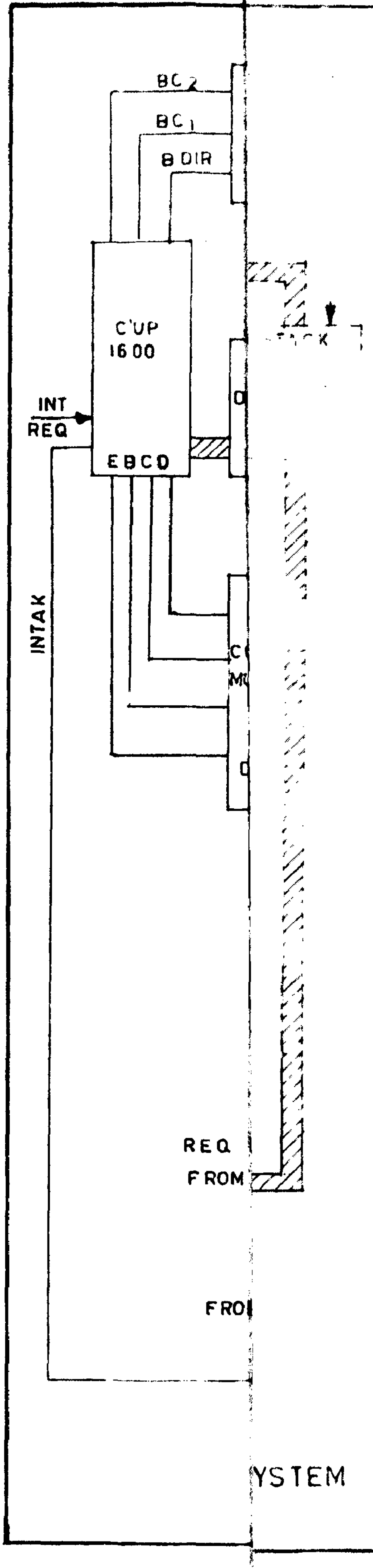


FIG. 28 STATE DIAGRAM OF INTERRUPT OCCURANCE IN CP 1600

counter has been PUSHed onto the stack.

Thus, the final hardware structure, is shown in Fig. . The input lines to this structure are connected with the output lines of the integrator, as shown earlier in the complete scheme of digital data processing of NDIR infrared spectrophotometer. The output lines of the developed structure are to be connected with the display devices and with control and command lines of step motor and integrator respectively to achieve the desired control described earlier. Using multiplexer decoder and universal TV transmeter /receiver, system can be married with the visual displays. More than one screen, can be used depending on the number of channels used in sample-cell of the spectrophotometer, giving out the corresponding channel display of the spectrum of the sample on the particular screen. One can also use, printer or X-Y recorder or a cassette tape to record the output, if he desires so, instead of using TV displays.

The connection of the various PIA's, ACIA's, RAM's and ROM's chips have been given in Fig.30, and with the interconnection to CPU in Fig. 31.



CPU	CP 1600
DECODER MUX	3205
CTIONAL BUS	8T26
AL LATCH	4
[illegible]	55 OR MCM 581 2102
	DYNAMIC ROM 21078/ 8
PR	EL 2708
A	MC 6820
A	MC 6850
PORT	8755
ONVE R	M 6912
NVERTER	
PRIOR NCO	TEL 8214
BUFFERS	
	8212
MIC	

AS ONOUS COMMU
ADC
PER AL INTER

YSTEM

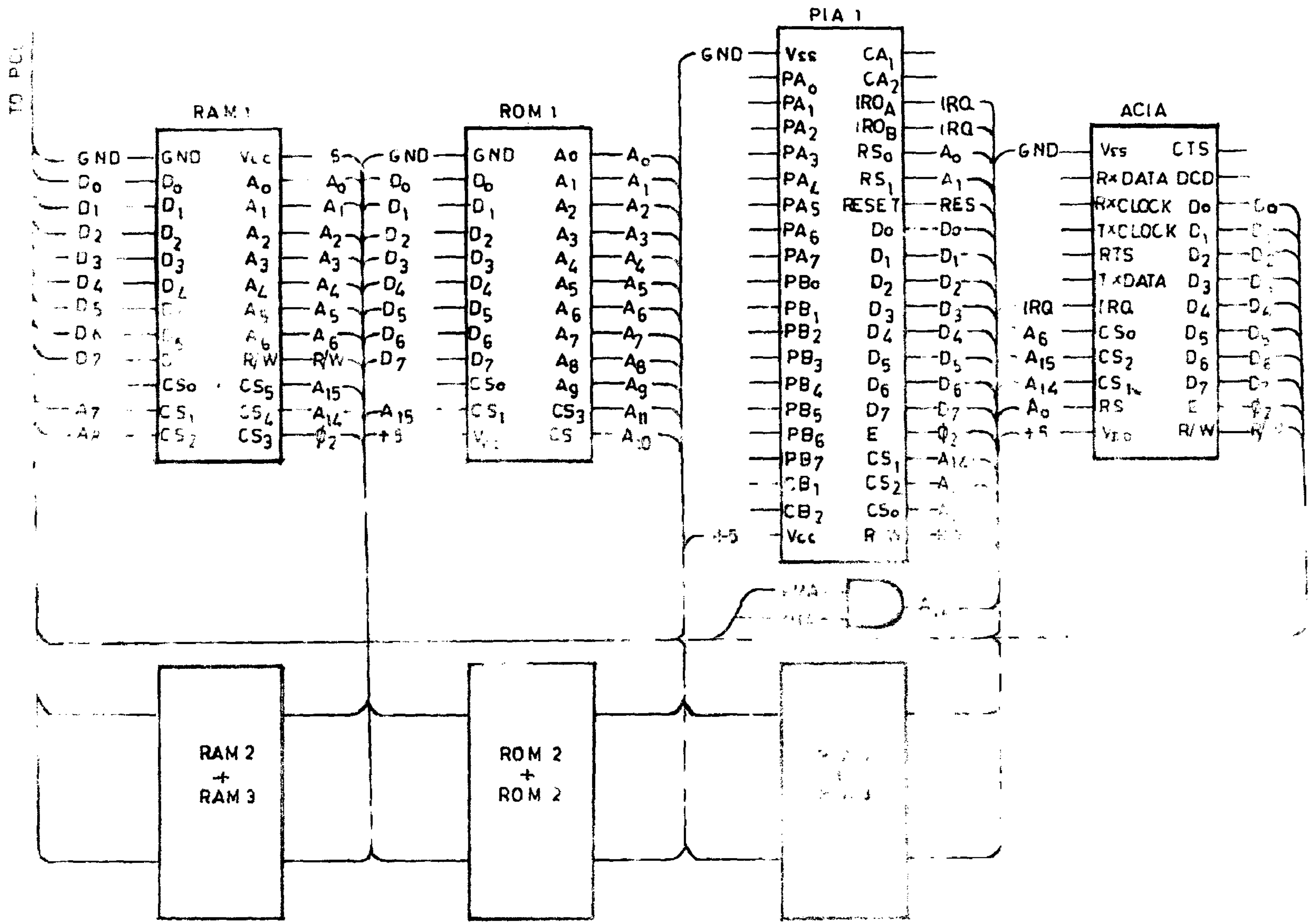


FIG. 30 INTERCONNECTIONS OF THE ADDITIONAL PERIPHERALS

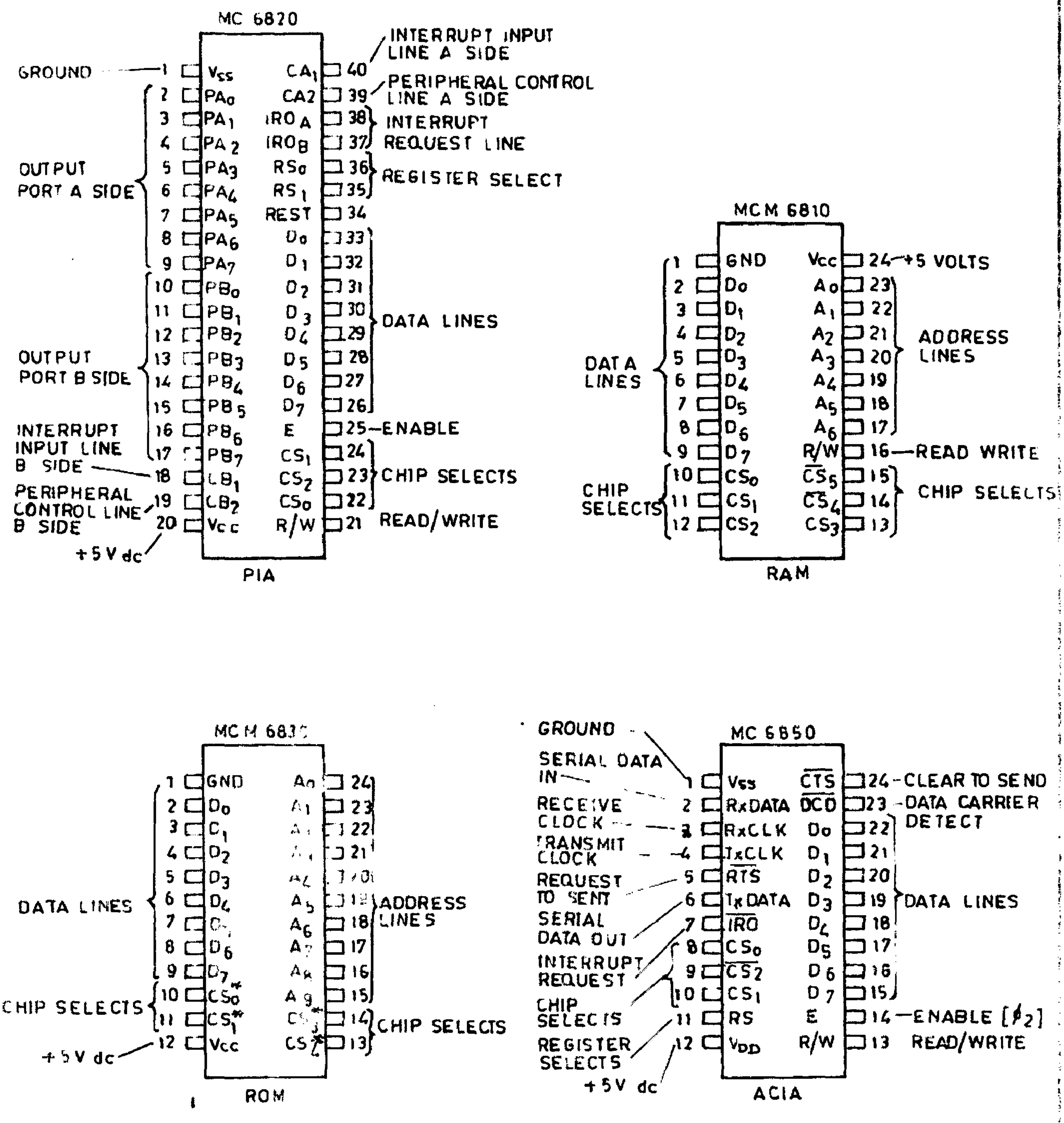


FIG. 31 PIN ASSIGNMENTS

CHAPTER - X

ADVANTAGES OF A MICROPROCESSOR BASED NDIR CONTROL SYSTEM

1. Improving Sensitivity and Signal to Noise Ratio.

The developed system has an additional advantage of digital data acquisition and reduction hardware alongwith the digital data handling of an NDIR infrared spectometer. The spectrum-average technique for improving the signal-to-noise ratio (S/N) is adequately precised and valuable technique even in the region of low transmittance. By having a number of spectra for averaging to form one spectrum and by taking response time factor of the detector into consideration, we can produce an appreciable improvement in S/N equal to the square root of the number of the spectra taken for averaging. Thus enhancing the S/N ratio of spectrometer data and improving the sensitivity over that available from the spectrometer itself, a better spectrum with more precised details can be obtained after mathematical smoothing.

The process of applying a least square smoothing function to noisy data is basically similar to the analog concept of an RC filter, except that the analog filter can only use data which have preceded the present measurement in time, and is thus one sided. Since the spectrum is already completely stored in the case of mathematical smoothing, the process can use equal intervals of data both before and after the point being smoothed. This is an additional advantage over the presently available control systems of spectrometer.

2. Scaled Spectrum Subtraction

One of the most difficult techniques to master in experimental infrared spectroscopy is differential spectroscopy. Differential spectroscopy involves the preparation of a reference cell system which matches the unwanted portion of the simple spectrum perfectly. For instance, aqueous solution spectra, in the transmission mode require extremely thin(3μm) cells of fairly high solute concentrations. Even with such thin cells, the water absorbs so much energy that a compensating cell must be placed in the reference beam, preferably one which can be adjusted until its transmittance matches that of sample cell.

But with the help of spectrum subtraction one can provide insight into the reproducibility of the spectrometer. For example, taking two independent polystyrene spectra, taken about 2 hrs apart. The two superimposed polystyrene spectra can be subtracted with the difference adequately small to retain the precision. And with the inherent capability of spectra subtraction (as they all have been already stored), even if one experts slightly slower scanning speed for the lower energy regions, by possible averaging and smoothing, the reproducibility of the polystyrene spectra can be improved.

3. Care for Several Components of Interest.

In many industrial and technological fields where it is required to get through a number of samples or components of interest quickly for the infrared spectroscopic analysis, this system developed with the filter-wheel and step-motor control can be proved suitable for appreciation. To moniter the desired several components of the gas-stream simultaneously and more or less continuously, the scheme can be used adequately.

4. Cheaper in Cost Consideration

The P.E. 180 model provides much more flexibility of operations. One may require about $ 15,000 on a data acquisition system for this type of model. But the investment required for the interface between an infrared spectrophotometer and the I/O bus of a microprocessor based digital control system is surprisingly small. For laboratories and new installations, the cost of computing hardware can be estimated of the order of about $ 6,000 only.

CHAPTER - X

CONCLUSION

A microprocessor based digital control system of a non-dispersive type Infrared Absorption Spectrophotometer has been developed using General Instruments CP1600 microprocessor chip. Possible modifications for the desired digital control of NDIR Spectrophotometer have been given regarding the filter-wheel type detector, step-motor controlled multi-channel system etc. to meet the vital requirements of the spectrophotometer in industrial field where response of spectrophotometer, alongwith the precision of spectra is desired. With the vital need of processing various spectra at a time for analysis has also been discussed and taken under consideration for the possible solution. Various other features for making the spectrophotometer digital control more useful and feasible to operate have been discussed.

Though the problem has been taken here, has taken various features of Perkin-Elmer model 180 Infrared Spectrophotometer, initially, to take care for developing the software, but later on efforts have been made to make the main program generalized, suitable for any spectrophotometer with the required but little changes. Similarly a generalized hardware scheme has been developed which would remain unchanged for any-type of output display except the required change in peripheral adapters and the additional output (write) software.

The CP1600 has been chosen because its system components features, interfacing, straightforward architecture and operation, and good engineering support both in hardware and software development etc.

REFERENCES

1. Robert J. Bibbero,'Microprocessors in Instruments and Control', A Wiley-Interscience Publication, New York, 1977, pp.176-202.

2. L.P. Morgenthaler and T.J. Poulos,'A Microcomputer System for Control of an Atomic Absorption Spectrometer', Am. Lab.,8 37-45(Aug. 1975). See also A.L. Robinson, Science 195(4284) 1314-1318, 1367(1977).

3. R.J. Bibbero and I.G. Young, System Approach to Air Pollution Control, Wiley Interscience, New York,1974, pp.401-409.

4. S. Siggia, Continuous Analysis of Chemical Process Systems, John Wiley, New York, 1959, Chap.6.

5. Specification Sheet No.SS-41-122, 'Multicomponent Type Infrared Gas Analyzer M-300', Yamataka-Honeywell,Tokyo, Japan, 1976.

6. J.H. Garfunkel, A.M.J. Pearman, and E.R. Rang, 'The Correlation Infrared Process Analyzer', Honeywell Research Centre, Bloomington, Minn.,Aug.1975.

7. A.B. Sidline, 'Operating System Software for a Microprocessor', Digest of Papers, Fall COMPCON, catalog no. 75CH 0988-6C,IEEE,New York,1975.

8. CDP3 Programmer's Reference Manual,System 700 Software, Honeywell Inc.,1975.

9. The writer is indebted to A Kegg of the Process Control Division (Ft.Washington Pa.),Honeywell Inc.,

who originated the program from which this adapted.

10. H.A. Laitinen, The seven ages of an analytical method, Anal. Chem., 45:2305, 1973.

11. R.N. Jones, T.E. Bach, H. Fuhrer, V.B. Kartha, J. Pitha, K.S. Seshadri, R. Venkatraghavan, and R.P. Young, Computer programs for absorption spectrophotometry, Bulletin No.11, National Research Council of Canada, Ottawa, 1968.

12. J. Pitha and R.N. Jones, Optimization methods for fitting curves to infrared band envelopes:Computer programs, Bulletin No.12, National Research Council of Canada, Ottawa, 1968.

13. R.N. Jones and R.P. Young, Additional computer programs for absorption spectrophotometry and band fitting, Bulletin No.13, National Research Council of Canada, Ottawa, 1969.

14. J.S. Mattson and A.C. McBride III, Digital data acquisition from a Hilger-Watts H-1200 infrared spectrophotometer, Anal.Chem.,43:1139 (1971).

15. R.W. Hannah and A. Savitzky, Rules for digital smoothing of spectral data--Part I, presented at the Eastern Analytical Symposium, Atlantic City, November,1972.

16. A. Savitzky and R.W. Hannah, Rules for digital smoothing of spectral data--Part II, presented at Pittsburgh Conference on Anal. Chem. and Appl. Spectry., Cleveland, March, 1973.

17. A. Savitzky and M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal.Chem. 36:1627(1964).

18. J. Steiner, Y. Termonia, and J. Deltour, Comments on smoothing and differentiation of data by simplified least square procedure, Anal. Chem.44:1906 (1972).

19. R.G.J. Miller and B.C. Stace (eds.), Laboratory Methods in Infrared Spectroscopy, 2nd ed., Heyden, London, 1972.

20. H. Hansdorff and H. Sternglanz, Differential recording with infrared spectrophotometers, Suppl. Vol.I, Ser.X, Del Nuovo Cimento, 1955:1, Perkin-Elmer Reprint No.IR-13.

21. F.S. Parker, Infrared spectroscopy in Biochemistry, Biology and Medicine, Plenum, New York,1971.

22. N.J. Harrick, Internal Reflection Spectroscopy, Wiley-Interscience, New York, 1967.

23. Edwin E. Klingman,'Microprocessor Systems Design', Prentice-Hall Inc.,New Jersey,1977.

24. John L. Hilburn, Paul M. Julich, 'Microcomputers/ Microprocessors' Prentice-Hall Inc.,N.J.,1976.

25. Daniel R. McGlynn, 'Microprocessors', A Wiley-Interscience Publication, Canada, 1976.

26. Laurence Altman, 'Applying Microprocessors',Electronics Magazine Book Series, McGraw-Hill Inc.USA,1976.

27. John R. Dyer, 'Applications of Absorption Spectroscopy of Orgnic Compounds' Prentice-Hall of India Ltd. 1969.

28. Thampy A. Thomos, 'Design techniques for Microprocessors Memory Systems', Computer Design, August 1975.

29. Mark Reed, H.W. Mergler, 'A Microprocessor-Based Control System', IEEE Trans. on Industrial Electronics and Control Instrumentation, vol.IECI 24, No.3, August 1977.

30. H.W. Mergler and K.H. Hubbard, 'Digital Control for Single Loop', Control Engineering, February 1965.

APPENDIX - A

Partial List for Instructions for CP1600

1. Register	Register	Instructions
ADD	RS,RD	Add RS into R4, R3 unchanged
SUB	RS,RD	Subtract RS from RD and put in RD, RS unchanged
MVR	RS,RD	Move RS into RD, RS unchanged
CMP	RS,RD	Examine RD minus RS to set status bits. RS and RD unchanged
2. Register - Address Space		
MVOV	SYM,RD	Move out RD to SYM (Symbolic address)
MVIV	SYM,RD	Move in value at SYM to RD
MVI*	NMBR,RD	Move in NMBR (number-inserted by assembler) to RD
CMP*	NMBR,RD	Compare RD with NMBR (RD minus NMBR) and set status bits, RD unchanged.
3. Shift - One Register		
SRLC	SI,RD	Rotate left with carry. MSB shifted to carry and carry bit to LSB.
SRRC	SI,RD	Rotate right with carry. LSB shifted to carry and carry to MSB.

4. Branch

{BNC BOL}	SYM	Branch on no carry (branch on lower).Branch to SYM if no carry bit (C=0),(unsigned).
BUNC	SYM	Branch unconditionally to SYM
BNZ	SYM	Branch if not zero, Branch to SYM (Z=1) if previous calculations did not result in zero.
BOC	SYM	Branch on carry (higher or equal). Branch to SYM
BOHE		if C = 1 (unsigned)
BGT	SYM	Branch on greater. Branch if previous comparison shown a greater number (signed)
BOS	N1,N2,SYM	Branch if logic 1 second (non-standard instruction)

5. Jump

JST	SYM,RD	Jump and store return. Jump to SYM and store next iteration address in RD.

6. Special and One Register

CLR	RD	Move zero into RD (0 → RD)
INCR	RD	RD + 1 → RZ
DECR	RD	RD - 1 → RD
NOP		No operation
COM	RD	Complement (1's complement)

HLT	Halt processor after insertion is executed.
CRC	Set carry bit to zero.

RS - any source register (addressable)

RD - any destination register (addressable)

SYM - external location symbol

NMBR - a number (inserted by assembler)

N1,N2 - external test multiplexer address (non-standard processor)

Status bits,(indicator flags) - S - sign (+ is 0)
Z - zero (1 if zero)
OV - overflow
C - carry bit

STPST : STOP START - a first transition used to stop the CPU after completion of an interruptable instruction followed by a second transition to start the CPU again.

BDRDY : BUS DATA READY - a signal used to resynchronize the CPU to slower peripherals by causing the CPU to wait.

PCIT* : PROGRAM COUNTER INHIBIT/TRAP* - a signal used to prevent the incrementing of the program counter during instruction execution; the pin may also be used to the output mode to indicate a software interrupt instruction.

INTR*: INTERRUPT REQUEST - an interrupt signal

INTRM* : INTERRUPT REQUEST MASKABLE, an interrupt signal which is executed only, if an internal interrupt flag is set

* indicates certain input pins represent negative logic.

Mnemonic	Op code	Meaning
External Reference		
MVOI	1001(RS,RD)	Move out, Send data from processor RD to location in RS in address space (memory or peripheral)
MVII	1010(RS,RD)	Move in, Bring data into processor RD from address in RS.
ADDI	1011(RS,RD)	Add contents of RS to RD; RS unchanged.
SUBI	1100(RS,RD)	Subtract contents of RS (2's component + 1)from RD. RS unchanged.
CMPI	1101(RS,RD)	Compare. Examine value of RD minus contents of RS; RS and RD unchanged.
ANDI	1110(RS,RD)	AND contents of RS to RD; RS unchanged.
XORI	1111(RS,RD)	Exclusive OR contents of RS to RD; RS unchanged.
B---	1000(----)	Branch on condition, using last six bits as modifiers
Internal reference		
MVR	0010(RS,RD)	Move contents of Rs to RD; RS unchanged.
ADD	0011(RS,RD)	RS+RD→RD,RS unchanged.
SUB	0100(RS,RD)	(-RS)+RD→RD,RS unchanged.
CMP	0101(RS,RD)	(-RS)+RD;RS and RD unchanged.
AND	0110(RS,RD)	RS and RD→RD;RS unchanged
XOR	0111(RS,RD)	RS ∀ RD→RD, RS unchanged
S____	0001(__,RD)	Shift using source register code bits on modifiers.
Xarious	0000(__,RD)	Ore operand instructions using source bits register code bits as modifiers.

∀ means exclusive OR

APPENDIX - B

```
GCHAR :   DOB     Ø,Ø5     : Set accept low
          SKPON   Ø5       : Skip next instruction when
                             'done' is set
          JMP     .-1
          DIAP    Ø,Ø5     : Take character, set accept high
          LDA     2,MSK
          AND     2,Ø      : Right adjust
          NIOC    Ø5       : Clear 'done'
          JMP     Ø,3      : Return
```

Get Character Subroutine for Spectrometer Data Acquisition System

```
STEP    SETD    MVS *         START OF SETTING FUNCTION PT
        INIT    CLR           DISPLAY AND SET PT
                SETD,Ro       SET FOR ZERO
CMST    BUNC    STRT          GET CMST AND INTERVAL

        MVIP    NPTS,Ro       JUMP TO SDATA AND GET 1ØØ POINTS
                SWS,Ro        FROM SPECTROMETER
        CMP     SWS,1         TEST '1' TO SW-5
        MVOV    AVGN          IF '1' MOVE AVERAGING THE 1Ø   PTS
        BUNC    STRT
        MVOV    SETP          IF '2' TO SW-5
                              MOVE START OF SETTING PT
        BUMC    STRT
AVGN    MVS*    AVG,TEMPØ     STORE AVERAGE
PTFS    CMP     SW6,1         TEST '1' TO SW6
        MVOV    Ro            IF '1' MOVE TEMPØ → OPT
        BUNC    STRT
        MVOV    PTFS          IF '2' MOVE TEMPØ → PTFS
        BUNC    STRT
        MVDV    SW6           MOVE  1  IN SW6
                SETD,Ro       SET FOR 100
```

Subroutine for 'Live Zero' and 1000% T setting.

Subroutine to check ASCII - to Floating Point Conversion Checkwords

```
Check   STA     3,PSAVØ       : Save return address
        LDA     2,WSA         : Get address of FPI area
        LDA     1,Ø,2         : Get OFLO/UFLO word
        LDA     Ø,1,2         : Get COK word
        SUBO    3,3           : Generate Ø in AC3
        STA     3,Ø,2         : Clear OFLO/UFLD word
        STA     3,1,2         : Clear COK word
        MOV     1,1,SNR       : Test OFLO/UFLO for Ø
        JMP     +13           : NO OFLO or UFLO
        SUBZL   3,3           : Generate +1 in AC3
        SUB     1,3,SZR       : Test for UFLD
        JMP     +6            : Was OFLO
        LDA     Ø,U           : Was OFLO
        JSR      @PUTC        : Type " U"  on TTY
        JSR      @CRLF        : CR-LF
        LDA     3,PSAUØ       : Get return address
        JMP     -4,3          : Bad datum, get another
        LDA     Ø,O           : Was OFLD
        JMP     -5            : Type " O"  on TTY, return
        LDA     3,PSAUØ       : Get return address
        MOV     Ø,Ø,SZR       : Test COK for Ø
        JMP     Ø,3           : Normal return
        JMP     -4,3          : No conversion, get another
PSAVØ:  Ø
WSA     BLK     1ØØ           : 1Ø0 -word storage area
U :     " U                   : ASCII " U"
O :     " O                   : ASCII " O"
PUTC :  (Address of subroutine to type single character)
CRFL :  (Address of subroutine to send carriage return-
        line feed)
```

STA	-	Store Accumulator at address
LDA	-	LoaD Accumulator from address
MOV	-	MOVe data
ADD	-	Address
JMP	-	Jump, Another data.

```
STPST   SETD     MVI         START OF SETTING FUNCTION IN
                 NMBR, RD    GET NUMBER OF POINTS
        BOHE     SYM         QUERY-SMOOTH OR PLOT
        MVOV     SYMP        PLOT
        BUNC     STRT
        MVOV     SYMS        SMOOTH
        BUNC     STRT
SYMP             STA         STORE PLOT PARAMETERS
        INIT     CLR         INITIALIZE COUNTERS
                 STAXD       STORE LOWEST VALUE
CNTS    CMP      NPTS        IF NO DATA  MOVE TO
        MVOV     BEGIN       BEGIN
        BONC     STRT
        BNC      SYM         CONVERT DATA INTO BINARY
        LDA      DATA
        COMP     SCLLG
        MVOV     DATB        IF '1' MOVE TO DATE
DATB    INCR     RD,RD       INCREMEND DATA BY DOUBLE IF
                             GREATER THAN '1' MOVE TO LDA
        MVOV     LDA         LOAD FOR PULSE FOR STEPPING
                             MOTOR
        BUNC     STRT
        COMP     PIPM        IF '1' MOVE TO DECR
DECR    MVOV     SUB,RS,RD
```

```
               Program                   Comments

SUBR       CLR R3                 Clear quotient register
           CMP R5, R2             Test for A< B
           BOC ERR                Error if A not < B
           MVI ≠ 4, R6            Initialize loop counter
           CRC                    Clear carry bit
DV1        SRLC SI, R2            Shift dividend
           CMP R5, R2             Compare with divisor
           BNC DV2                Jump if C = 0
           SUB R5, R2             Subtract divisor if C = 1
DV2        SRLC SI, R3            Shift carry into R3
           DECR R6                Decrement loop counter
           BNZ DV1                Loop until counter is 0
```

Subroutine for Divide Routine

```
          JMP   INTR2  :  Jump to interrupt-service
                       :  routine for device 2
INTR2 :   PUSH  RN1    :  Save RN1 and RN2 registers
          PUSH  RN3    :  Save RN3 and RN4 registers
          :
          PUSH  PSW    :  Save accumulator and status
          :
                       :  Transfer instruction for device 2
          :
          POP   PSW    :  Restore accumulator and status
          POP   RN1    :  Restore RN1 and RN2 registers.
          POP   RN3    :  Restore RN3 and RN4 registers
          :
          EI           :  Enable interrupt
          RET          :  Return to pre-interrupt location
          :
```

Software for Interrupt Service

APPENDIX - C

NDIR Control Main Program

```
STEP    EQU     '1012           DEFINES EXTERNAL ADDRESSES
INT     EQU     '1203
ATD     EQU     '1230
CHNA    EQU     '1234
CHNB    EQU     '1324
CHNC            '1423
STRT    BOS     0,0 INIT        0,0 IS ADDRESS OF FILTER SYNCH.
                                PULSE
        BUNC    E'RR
INIT    CLR     R1              DIVIDEND/DIVISOR GATE COUNTER
        CLR     R4              DISPLAY GATE COUNTER
        CLR     R5              TIME DELAY INCREMENT COUNTER
        INCR    R5              "1"  IN R5
        MVOV    STEP,R5         "1"  TURNS ON STEP MOTOR
PLY     JST     SETD,R6         START 200 MS DELAY SUBROUTINE
        CMP*      R5            START INTEGRATOR AFTER 200 MS DELAY
        BNZ     TDEL            ON MOVING
        MVOV    INT,R5          "1"  TO IHT LOCATION
TDEL    CMP*    3,R5
        BOHE    DELE            JUMP AFTER 600 MS DELAY
        INCR    R5
        BUNC    DELAY           REPEAT 200 MS DELAY LOOP
DELB    CLR     R5
        MVOV    IHT,R5          '0' TO INT STOPS INTEGRATOR
        SRRC    S1,R1           TEST LSB OF COUNTER
        BOC     LRS1        BRANCH IF '1'
        SRLC    S1,R1       REGISTER  GATE COUNTER
        COM     R1          COMPLEMENT COUNTER
        MVIV    ATD,R2      MOVE IN 'REFERENCE' SIGNAL TO
                            DIVIDEND
NEXT    JST     SETD        JUMP TO 200 DELAY
        INCR    R3
        MVOV    STEP,R3     ACTIVATE STEP MOTOR
        BUNC    DLY         REPEAT DELAY AND INTEGRATION CYCLE
LRGI    SRLC    S1,R1       RESTORE GATE COUNTER
        COM     R1          COMPLEMENT
        MVIV    ATD,R5      MOVE IN 'SENSE' SIGNAL TO DIVIDER
        CLR     R3          START OF DIVDE ROUTINE
        CMP     R5,R2
        BOC     ERR         EXIT TO ERROR ROUTINE, DIVDEND
                            AND DIVISOR
        MVI*    15,R6       SETS 15 BIT ACCURACY
        ADD     R3,R4
DVI     SRLC    S1,R2
        CMP     R5,R2
        BNC     DV2
        SUB     R5,R2
DV2     SRLC    S1,R3
        DECR    R5
        BNZ     DV1         END OF DIVISION ROUTINE
        COMP    R6,R4       TEST DISPLAY COUNTER FOR '0'
        BGT     ONE
        MVOV    CHNA,R3         IF '0' MOVE QUOTIENT TO DISPLAY 'A'
        BUNC    LRST
ONE     INCR    R6
        CMP     R6,R4           TEST FOR '1'
        BGT     TWO
        MVOV    CHNB,R3         IF '1' MOVE QUOTIENT TO DISPLAY 'B'
        BONC    LAST
TWO     MVOV    CHNC, R3        IF '2' MOVE QUOTIENT TO DISPLAY 'C'
        BUNC    START
LAST    INCR    R4
        BUNC    NEXT            JUMP TO NEXT RATIO CALCULATION
 *

ERR     INCR    R3              START OF ERROR ROUTINE
        MVOV    STEP,R3         STEP MOTOR
        MVI*    5,R4            AT THE SECOND INTERVAL
RPT2    JST     SETD,RL
        DECR    R4
        BNZ     RPT2
        BOS     0,0,INIT        STOP WHEN SENSE SYNCH SIGNAL
        BUNC    ERR             ELSE REPEAT LOOP
*
*
SETD    MVI     20000,R3        START OF 200 MS DELAY ROUTINE
RPET    DECR    R3
        NOP
        BNZ     REPT
        MVR     R6,R7           RETURN TO MAIN PROGRAM
 *
        END
```