A COMPUTER BASED DATA ACQUISITION SYSTEM FOR FAST PULSED POWER EXPERIMENTS

by

ALEX KAMLUN AUYEUNG, B.S.E.E.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved

Accepted

May, 1981

ACKNOWLEDGMENTS

I am deeply indebted to Dr. Marion Hagler and Dr. Magne Kristiansen for their help and guidance throughout this project. I would like to thank Dr. Donald Gustafson and Dr. James Archer for serving on my committee and for their helpful criticism. I would also like to thank my wife Doris for typing the manuscript of this thesis. Finally, I would like to thank my fellow graduate students for their help and cooperation in the laboratory.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ii

LIST OF TABLES vi

LIST OF FIGURES vii

I. INTRODUCTION 1

II. THE IEEE STANDARD 488-1978 6

2.1 Introduction 6

2.2 A Typical GPIB System 6

2.3 IEEE 488 Signal Lines 8

2.4 Bus Messages 13

III. EQUIPMENT 15

3.1 Introduction 15

3.2 The PDP 11/34 Computer System 15

3.3 Tektronix 7912AD Programmable Digitizer . 19

3.4 HP 12050A Fiber Optic Data Link 21

3.5 The CAMAC System 22

IV. SOFTWARE 26

4.1 Introduction 26

4.2 System Information 27

4.3 Data Disk 32

4.4 BASIC System Programs 34

4.5 DATACQ.BAS 35

4.6 Hardware Configuration File 36

4.7 STORE1.BAS 36

4.8 Data File 39

4.9 GRAPH1.BAS 41

4.10 READ1.BAS 41

4.11 STATS1.BAS 44

4.12 System Support Programs 44

4.13 Analysis Programs 46

V. OPERATING INSTRUCTIONS 47

5.1 Introduction 47

5.2 Turn-on and Bootstrap Procedure 47

5.3 Creating a Data Disk 50

5.4 Running of the Storage Routine 50

5.5 System Commands 52

5.6 Recovery From a Crash 56

VI. SYSTEM CHECKOUT 58

6.1 Introduction 58

6.2 Hardware Setup and Test Procedure 58

6.3 Execution of System Programs 60

VII. CONCLUSION 79

REFERENCES 82

APPENDIX A. PROGRAM LISTINGS AND DISCUSSIONS OF STORAGE ROUTINES 83

DATACQ.BAS 84

STORE1.BAS 93

GRAPH1.BAS 95

STATS1.BAS 97

READ1.BAS 99

CREATE.BAS 111

CATDIR.BAS 114

FILTR1.BAS 116

ANALYS.BAS 119

CURFIT.BAS 121

APPENDIX B. DATA ACQUISITION SYSTEM COMMANDS 124

APPENDIX C. CHARACTERISTICS OF THE OVERALL SYSTEM . . . 125

LIST OF TABLES

Table		Page
3.1	Components of the PDP - 11/34 Computer System . .	16
4.1a	Directory of System Disk DXO	28
4.1b	Directory of System Disk DXO	29
4.1c	Directory of System Disk DXo	30
4.2	Directory of Data Disk DX1	31

LIST OF FIGURES

Figure		Page
1.1	A Simplified Block Diagram of a Pulsed Power Network	2
1.2	A Block Diagram of the Data Acquisition System	3
2.1	A Typical System Based on the IEEE 488 Bus	7
2.2	A Typical Handshake Sequence	10
3.1a	Tektronix 7912AD Scan Converter	20
3.1b	Scan Converter Target	20
4.1	Flow Diagram of the Initialization Portion of DATACQ.BAS	37
4.2	Hardware Configuration File	38
4.3	Flow Diagram of STORE1.BAS	40
4.4	Typical Parameter File for Laser Triggered Spark Gap Experiment	42
4.5	Flow Diagram of GRAPH1.BAS	43
4.6	Flow Diagram of READ1.BAS	45
6.1	A Block Diagram of Hardware Setup for Testing	59
6.2a	Bootstraping and Loading of DATACQ.BAS	63
6.2b	Execution of DATACQ.BAS	63
6.3	Execution of ACQUIRE Command	64
6.4	Execution of GRAPH Command	65
6.5	Execution of STORE Command	66
6.6	Execution of STATUS Command	66
6.7	Execution of CREATE.BAS	67
6.8	Execution of CATDIR.BAS	67

Figure		Page

Figure		Page
6.9	Execution of ANALYS.BAS and Plot Command . . .	68
6.10	Execution of PEAK Command	69
6.11	Execution of INTEGRATE Command	70
6.12	Execution of DIFFERENTIATE Command	71
6.13	Execution of CURFIT.BAS	72
6.14a	9th Order Curvefitting	73
6.14b	11th Order Curvefitting	73
6.15	Execution of FILTR1.BAS	74
6.16a	Original Signal of Data Record 801118.001 . . .	75
6.16b	Filtered Signal of 801118.001	75
6.17	Filtered Signal of 801118.001	76
6.18a	Band Pass Signal of 801118.001	77
6.18b	High Pass Signal of 801118.001	77
6.19	Data Record Plotted by the READ Command	78
7.1	Scanning of the Target by the Reading Beam . .	126
7.2	A Portion of the Raw Data Array	130
7.3	A Portion of the Processed Data Array	131

CHAPTER I

INTRODUCTION

Pulsed power technology involves the rapid delivery of energy to the load at very high power levels. The general method of achieving such pulsed power is by slowly storing energy in a storage element and then switching the stored energy to a load so that a short, high power pulse is obtained. A block diagram for a non typical pulsed power network is shown in Fig. 1.1.

Data obtained from these experiments are in the form of voltage and current waveforms which are normally photographed by a camera mounted oscilloscope. As experiments and diagnostic methods become more complex, the amount of data required increases tremendously. Data organization and analysis will then become the most time consuming task of the experiment. To improve the efficiency of operation, a versatile and flexible computer-based data acquisition system has been designed and developed for pulsed power experiments at Texas Tech University. The development of this system includes a) up-grading of an old data acquisition system by replacing the old PDP 11/04 central processor with a PDP 11/34 processor, b) adding on a new fast digitizer and supporting hardware, c) writing control and data analysis programs for the system. The present data acquisition system is depicted by Fig. 1.2. Blocks marked with a "*" are the hardware used in the old system. This system allows transient waveforms with nanosecond time scale to be recorded

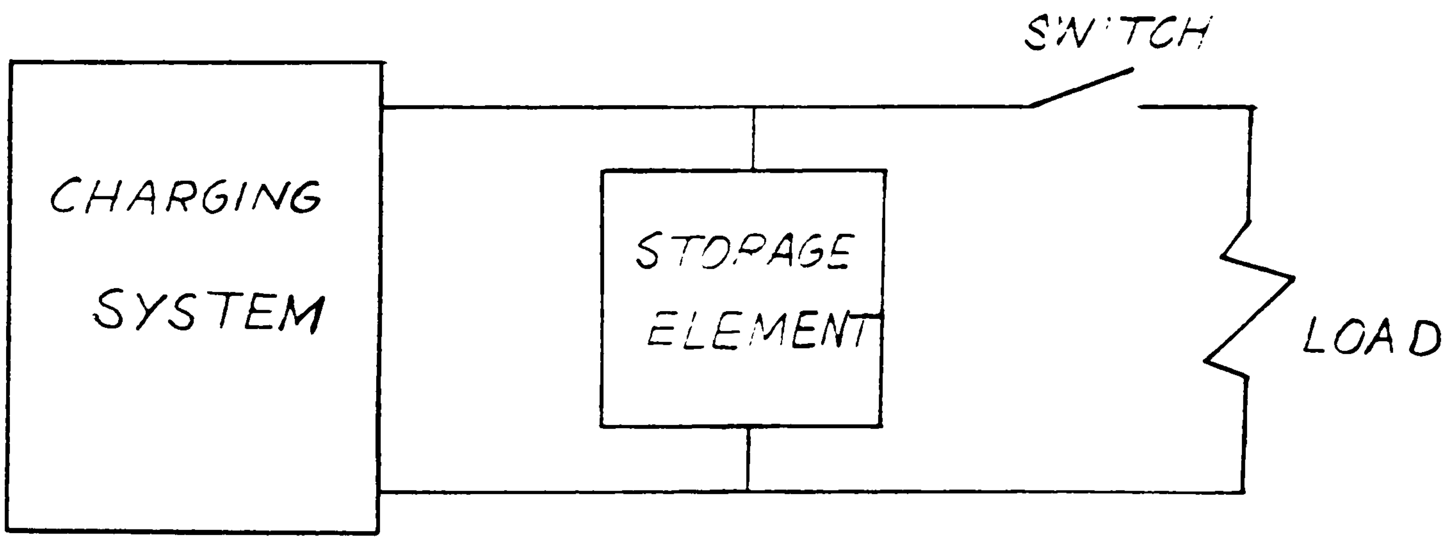


Fig. 1.1 A Simplified Block Diagram of a Pulsed Power Network

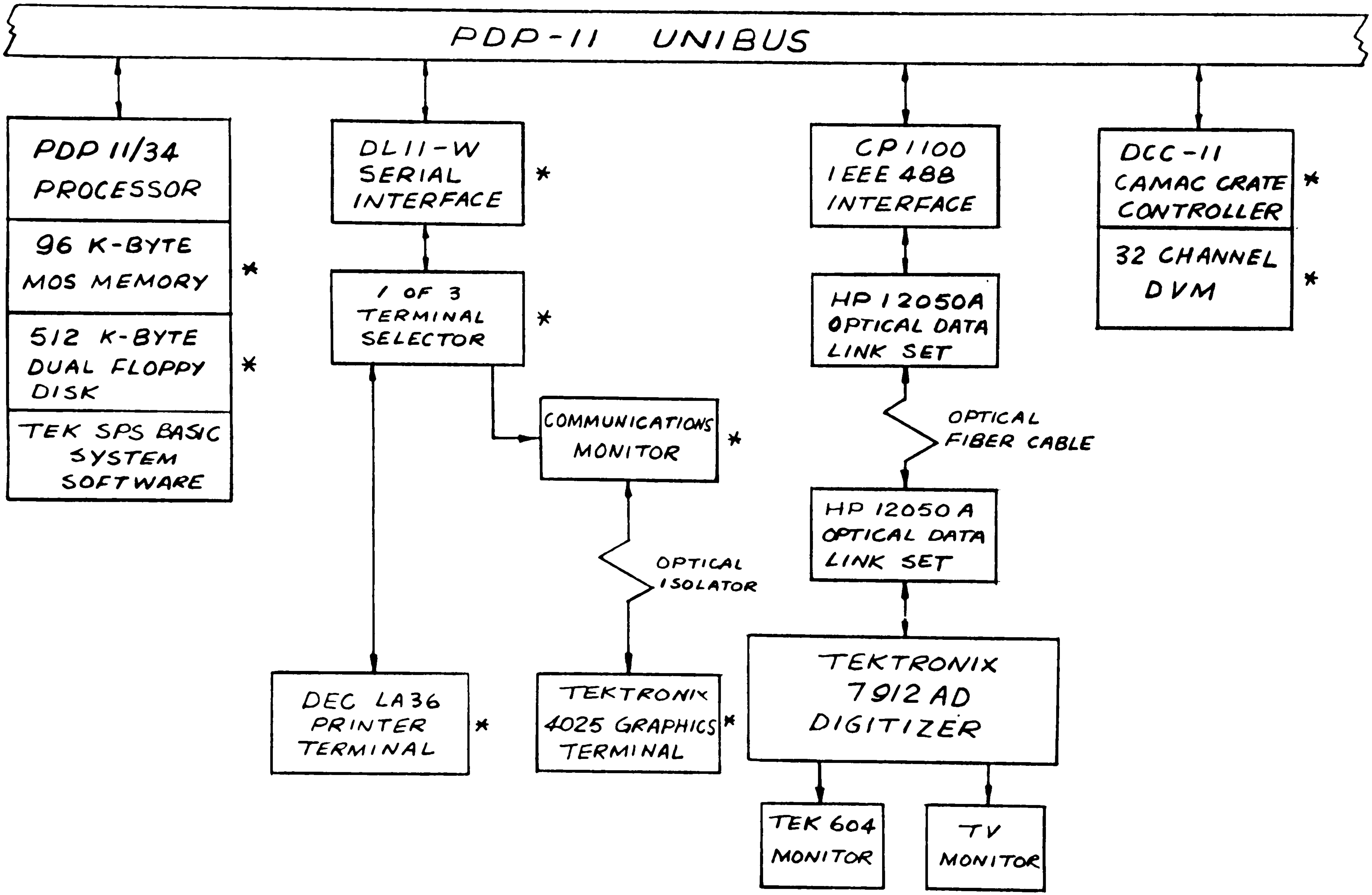


Fig. 1.2 A Block Diagram of the Data Acquisition System

in digital form. Waveform data are archived in such a way that they can be retrieved easily for analysis and comparison. Permanent records of data are stored on magnetic floppy disks and can be retrieved at any time for analysis. Graphs of both raw data and processed data can be plotted for comparison. Numerical analysis methods can be applied to the acquired data to greatly enchance the capability for data interpretation.

Figure 1.2 shows a block diagram of a Data Acquisition System which provides the features mentioned in the previous paragraph. A DEC PDP-11/34 minicomputer system (described in Ch. III, Sec. 3.2) is selected both to control all other equipment connected to the system and to run data analysis programs. This system includes 48-k words (16 bit) of dynamic RAM and a 512 k-byte dual floppy disk drive. Communication to and from the computer occurs via a DEC LA36 ASCII printer terminal. A Tektronix 7912AD programmable digitizer which is capable of capturing a transient waveform with a bandwidth of up to 1 GHz is interfaced to the PDP 11/34 system through a IEEE 488 interface bus[1] (described in Ch. II and Ch. IV). A CAMAC[2] (described in Ch. III Sec. 3.5) 32 channel scanning digital voltmeter (DVM) is used to record the slowly varying parameters of the experiments. The extension of the distance (up to 100 meters) and isolation between the PDP 11/34 computer and the Tektronix 7912AD digitizer are achieved by using a set of HP 12050A optical data link units. A Tektronix 4025 graphics terminal serves both as a communication terminal between the operator and the computer and as a graphics display. Digitized waveform data can be displayed immediately after digitization by a Tektronix 604 display monitor so that the operator can determine the validity of the data before storing it on the disk. A hard copy of the data displayed on the screen of the 4025 terminal can be made via a Tektronix 4631 hard copy unit which connects to the rear of the graphics terminal. The system software used is the TEK SPS BASIC (described in Ch. IV) in which the storage programs and analysis programs are written.

In the following chapters, a detailed description of the whole system, both the hard ware and software, is provided. Chapter II discusses the IEEE-488 interface bus system[1]. Each piece of hardware is discussed in Chapter III. The system software as well as the user written programs are discussed in Chapter IV. Chapter V presents an orderly set of operational instructions for acquiring and storing data with this system. Chapter VI describes the check-out of the system with the similation of the signal from a delay generator. In addition, a few examples of the execution of the main data strage program, system support programs and analysis programs are illustrated. Finally, Appendix A includes a listing of all programs used in this system. Appendix B lists all the commands used in this system. Appendix C describes the characteristics of the overall system.

CHAPTER II

THE IEEE STANDARD 488-1978

Sec. 2.1 Introduction

This chapter introduces the IEEE 488 standard[1] which describes a general purpose bus for instrument systems. The IEEE 488 bus, also known as the General Purpose Interface Bus (GPIB) is used in this data acquisition system to interconnect the Tektronix 7912AD digitizer and the PDP 11/34 computer. Since both the control messages and data transfer are carried via the GPIB, understanding the IEEE 488 standard will be helpful to those who write control programs for the Tektronix 7912AD digitizer.

The purpose of the GPIB is to provide an effective communications link over which messages can be carried between instruments in a clear and orderly maner. Instruments designed to operate according to the standard can be connected directly to the bus and operated by a controller (e.g. PDP 11/34) with appropriate programming.

This bus uses eight data and eight control lines. Information is transferred bit-parallel, byte-serial by an asynchronous handshake.[1] This arrangement allows instruments with different transfer rates to operate together.

Sec. 2.2 A Typical GPIB System

A typical GPIB system shown in Fig. 2.1 could include a controller (e.g. a PDP 11/34 with a GP1100/IEEE 488 interface card installed), a talker (e.g. a counter or digital

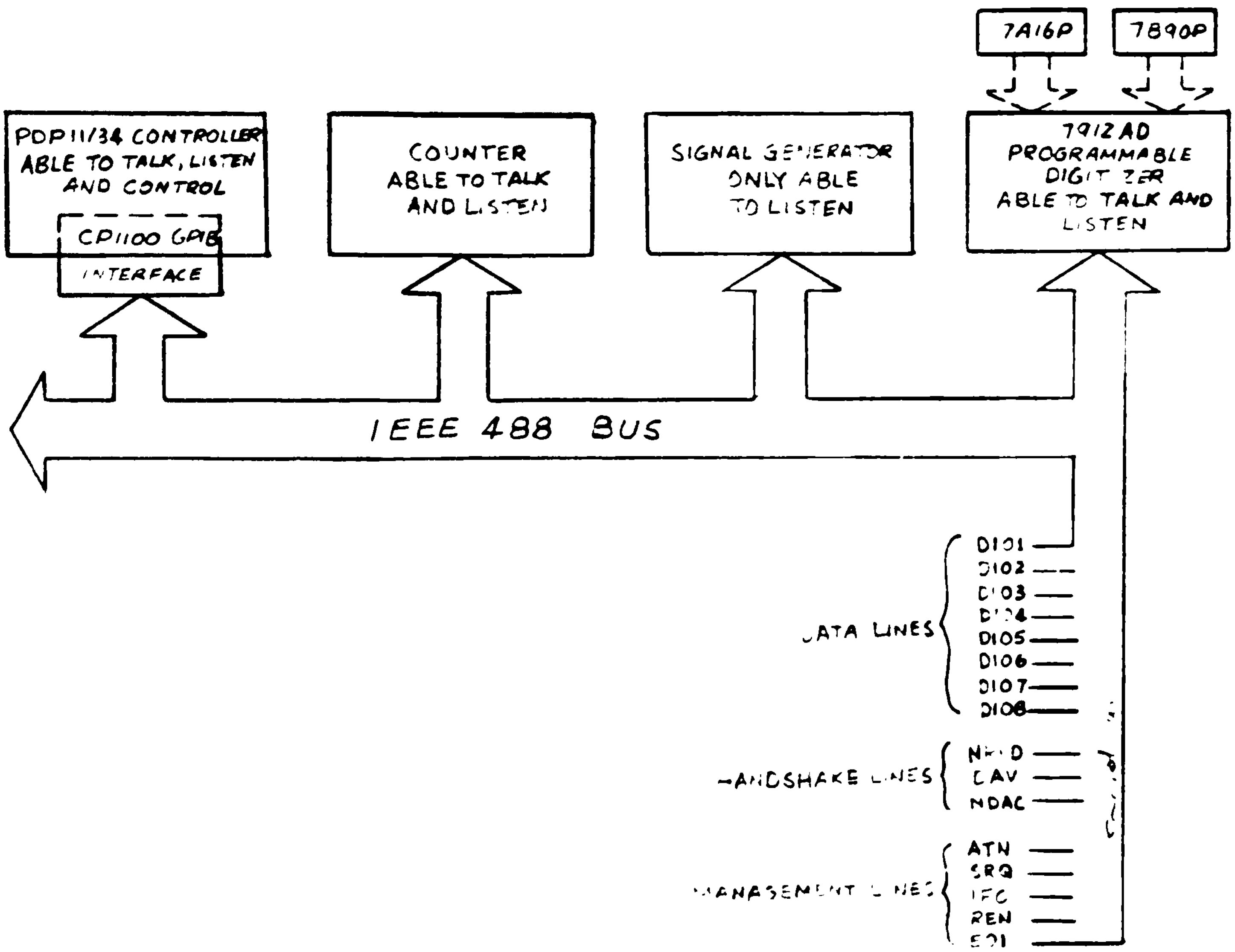


Fig. 2.1 A typical System Based on the IEEE 488 Bus

multimeter), and a listener (e.g. a line printer or signal generator). More than one function can be combined in a single instrument. For example, the Tektronix 7912AD Programmable Digitizer has both listener and talker functions.

Up to 15 devices, distributed over no more than 20 meters total cable length, can be connected to a single IEEE 488 bus. More than 15 devices can be interfaced if they do not connect to the bus directly but are interfaced through another device. Such a scheme is used for programmable plug-ins housed in the 7912AD mainframe; the 7912AD provides a transparent interface between the bus and the plug-ins.

Messages on the bus are either interface messages or device-dependent messages. Interface messages are used to manage the interface functions of the instruments. They designate talkers and listeners. Device-dependent messages are either remote-control messages which set the operation mode or the data transfer mode of an instrument.

Sec. 2.3 IEEE 488 Signal Lines

The IEEE 488 bus is functionally divided into eight data lines and eight control lines. The eight control lines consist of three handshake lines and five management lines. This bus structure is shown in Fig. 2.1.

Data Lines

The eight Data Input/Output lines (DIO1 through DIO8) are bi-directional active-low lines used to carry data or device-dependent messages. Device addresses and universal commands are also transferred over these lines when ATN is asserted. One byte of information is transferred over the bus at a time. DIO1 represents the least significant bit in the byte; DIO8 represents the most significant bit. Data is transferred in byte-serial, bit-parallel fashion. Data bytes can be formatted in ASCII code, or in machine-dependent binary code. The term "machine-dependent binary code" refers to an internal binary format used by a device to store certain programs and data.

Control Lines

The three handshake lines are used to communicate a handshake sequence that is executed between the talker and all designated listeners each time a byte is transferred over the data lines. This handshake sequence prevents the talker from placing a new byte on the bus until the slowest listener has received the previous byte. Thus the talker cannot transmit at a rate faster than can be received by the slowest listener. The three active-low handshake lines are NRFD, DAV, and NDAC. A basic timing diagram of the handshake sequence is shown in Fig. 2.2.

NRFD (Not Ready For Data)-This signal line is asserted until all assigned listeners are ready to receive the next data byte. When all of the assigned listeners have released NRFD, the NRFD signal is unasserted, thus allowing the talker to place the next byte on the data lines.

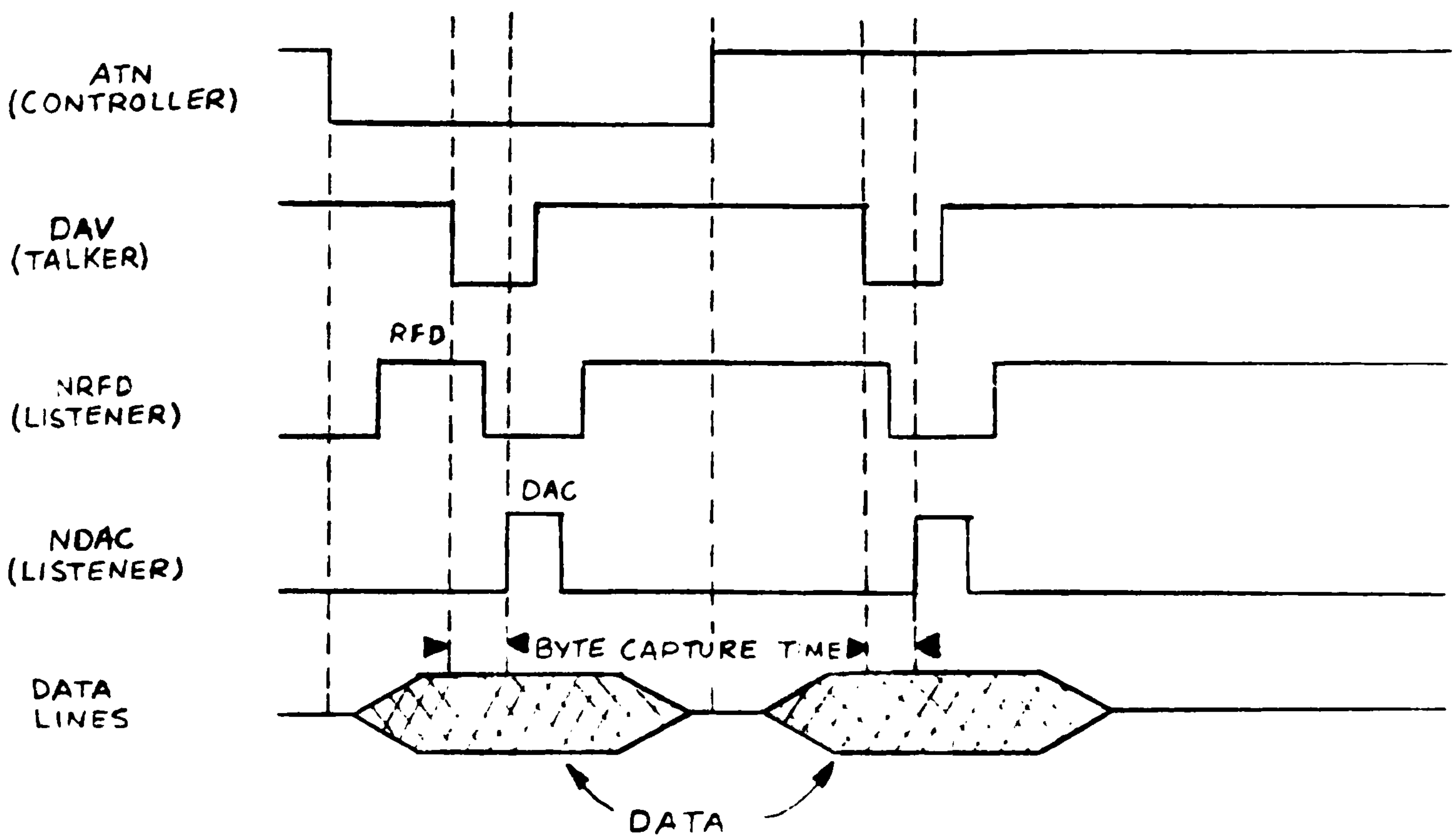


Fig. 2.2 A Typical Handshake Sequence

DAV (Data Valid) - The DAV signal line is asserted by the talker shortly after placing a valid byte on the data lines. This tells each listener to receive the byte presently on the data lines. DAV cannot be asserted until NRFD has been unasserted.

NDAC (Not Data Accepted) - This signal line is asserted until all the listeners have captured the byte currently on the data lines and released NDAC. When the slowest listener has received the data byte and released NDAC, NDAC is unasserted thus allowing the talker to remove the byte from the data lines. At that point, the DAV line is unasserted and the entire handshake cycle is repeated.

The five management lines are used to control data transfers over the data lines. The management lines perform important interface operations such as detecting an interrupt from a device, setting a device to remote control, and flagging the end of a message. These five signal lines are ATN, IFC, SRQ, REN, and EOI; their functions are:

ATN (Attention) - Asserted by the controller-in-charge to specify how information on the data lines is to be interpreted. When ATN is not asserted, the information on the data lines is interpreted as device-dependent messages and data. When ATN is asserted, the data lines carry universal commands, addressed commands, talk addresses, listen addresses, or secondary address.

Which addresses and commands are sent depends upon the byte currently on the data lines. The codes corresponding to various commands and addresses are defined in Appendix E of the IEEE 488 Standard.[1]

IFC (Interface Clear) - Asserted by the system controller to initialize the interface functions of all instruments to an inactive state and return control to the system controller. The IFC function effectively performs an UNListen, an UNTalk, and a Serial Poll Disable and resets all devices except the system controller to the idle state.

SRQ (Service Request) - Asserted by an instrument to request service from the controller. The controller usually interrupts its current task and conducts a serial poll to determine which device asserted SRQ. The controller can then branch to an interrupt service routine where appropriate action is taken. After the interrupt has been processed, the controller may resume execution of the previous task.

EOI (End of Identify) - Asserted by a talker to indicate the last byte of its message. When EOI is asserted with ATN, the controller is conducting a parallel poll of the devices connected to the bus. REN (Remote Enable) - Asserted by the system controller to allow devices on the bus to go to Remote mode, thus allowing remote control of their programmable functions. When in Remote mode, the front panels of the instruments ~~are instruments~~ are disabled except for any non-programmable functions.

Sec. 2.4 Bus Messages

As previously mentioned, messages on the data lines are either interface messages or device-dependent messages. When the ATN line is asserted by the controller, all devices "pay attention" since interface messages are to be transferred over the data lines. (By "pay attention" it is meant that all devices handshake and process all bytes transferred on the bus). Interface messages can be classified as follows:

1) talk address
2) listen address
3) secondary address
4) universal commands
5) addressed commands

The first three categories refer to how a device is to be addressed. That is, they designate a device either as a talker or a listener. To designate a device as a talker, the controller asserts ATN and places the device talk address to the data lines. Similarly, the controller designates a listener by asserting ATN and placing the device listen address on the data lines. In cases where secondary addressing is designed into a particular device (e.g. 7912AD), it is necessary to transmit the device secondary address with ATN asserted following the primary talk or listen address.

The fourth category listed (universal commands) consists of those interface commands which affect all devices connected to the bus, regardless of whether they are currently addressed as talker or listeners. Examples of universal commands are LLO (Local Lockout) and DCL (Device Clear).

The fifth category listed (address commands) consists of those interface commands which affect all devices currently addressed as listeners. A complete list of universal and addressed commands is provided in Appendix E of the IEEE 488 standard.[1]

In contrast to interface messages, device-dependent messages are sent with ATN unasserted and are transmitted only between a designated talker and one or more designated listeners. A device-dependent message can be either an instruction or data. Instructions and data are normally coded in ASCII or binary, but this is not required by the IEEE Standard.

The above discussion is only a brief introduction to the IEEE 488 interface. Detailed information can be found in Ref. 1.

CHAPTER III

EQUIPMENT

Sec. 3.1 Introduction

The equipment making up this data acquisition system is shown in Fig. 1.1 of Chap. I. It consists of a PDP 11/34 computer system (see Table 3.1) and several peripheral devices. These devvices are a RX01 dual floppy disk drive, a Tektronix 7912AD digitizer, a CAMAC[2] crate with controller, a 32 channel CAMAC[2] DVM, a Tektronix 4025 graphics terminal and a DEC LA36 printer terminal. They communicate with the PDP 11/34 computer through interface cards which are installed within the computer backplane[3]. In this chapter, the general function of each piece of equipment is discussed with respect to the overall operation of the system. Detailed descriptions can be found in the manufacturer's documentation.

Sec. 3.2 The PDP 11/34 computer system

One of the major components of this data acquisition system is the PDP 11/34 computer. It controls the operation of the 7912AD as well as data transfer among peripheral devices. A list of components which make up the 11/34 system is shown in Table 3.1.

All components of the PDP 11/34 system are connected to each other by a single bus known as the Unibus[4] (see Fig. 1.1 of Ch. I). Address, data and control information are sent along the 56 lines of the bus. Each unibus device is

TABLE 3.1

Components of the PDP - 11/34 Computer System

The 11/34 central processor (M7265 and M7266)

96 kb of active MOS dynamic RAM (3XM7847)

KY11-LB programmer's console (M7859)

Bootstrap ROM module (M9301-YA)

Parity generator/checker module (M7850)

DL11-W serial interface board (2XM7856)

Unibus terminator (M9302)

Floppy disk drive controller (M8264)

512 kb dual floppy disk drive (RX01)

CP1100 IEEE 488 bus interface board

LA36 printer terminal

TEK SPS BASIC V02 system software

Documentation kit

assigned an address on the bus so that the central processor can access each device like a memory location.

11/34 central processor - The 16 bit central processor of the 11/34 is contained in two multilayer circuit boards (M7265 and M7266). It is connected to the computer via the Unibus[4]. It controls peripheral devices, and performs arithmetic operations, logic operations, and instruction decoding. This processor also contains memory management logic which allows memory extention from 28k to 124k words.

KY11-LB programmer's console - This console contains a 7-segment LED display and a keypad for entering and verifying data as well as for controlling basic computer operations. Contents of any of the eight general purpose registers in the central processor or any memory location can be examined by using the keypad and the LED display. The console is interfaced to the Unibus via the console module (M7959).

Terminators - The PDP 11/34 contains a special terminator module (M9301) that contains the required unibus resistors and 512 words of read-only-memory (ROM). The ROM contains a bootstrap program which loads the operation system from the disk when the bootstrap characters are entered through a terminal. Another unibus terminator module contains terminating resistors and logic circuits which indicate to the processor that the physical end of the bus is reached.

Memory - There are three 16k words MOS memory boards

(M7847) installed in this system. Micro switches are available for setting the address when these boards are installed so that the system knows the amount of memory available. The parity of the stored data in memory is checked by a parity controller (M7850).

Floppy disk controller - The module (M8264) is the floppy disk RX01 controller which interfaces a dual floppy disk drive to the Unibus. Up to 256,256 8-bit bytes of information can be stored on a single floppy disk. A total storage capacity of 512,512 bytes is available on both disks. The average access time is 483 milliseconds. The whole diskette can be read sequentially in about 30 seconds.

Serial Interface - There are two DL11-W (M7856) interface modules installed in this system. Only one is used to interface to the Tektronix 4025 terminal. The other DL11-W is reserved for use in multi-terminal, multi-user systems. The DL11-W provides an asynchronous serial line interface to an ASCII terminal (e.g. LA36, Tektronix 4025) and a line frequency clock. The serial line interface can handle data transfer rates from 110 to 9600 baud (bit/ second) and provides serial-to-parallel (and vice versa) data conversion for information transfer to or from the Unibus. The line clock senses the 60 Hz line frequency for internal timing.

CP1100 interface - The CP1100 is a IEEE-488 interface module which interfaces GPIB devices (e.g. Tektronix 7912AD) to the Unibus. The CP1100 module maps the IEEE 488

Bus to eight unibus addresses. Data transfer between the Unibus and the IEEE 488 Bus is controlled by the processor through these eight registers.

Sec. 3.3 Tektronix 7912AD programmable digitizer

The Tektronix 7912AD programmable digitizer is the instrument which enables the transient waveform signals in nanosecond (10^{-9} second) time frames to be recorded in digital form. Because of the limitation of switching speed and time delay of electronic switches, the conventional "sample and hold" circuit is not used in this digitizer. Instead, a scan converter is used.

The scan converter (see Fig. 3.1a) consists of a write gun, a read gun and a semiconductor target. The target is an array of diodes formed on an n-type silicon wafer (see Fig. 3.2b). A transient signal is written on the target by the high speed electron beam of the write gun. Then the target is scanned by the read gun. Data on the target are converted into digital words and stored in local memory by a microprocessor within the 7912AD digitizer.

The 7912AD can be operated manually or by computer control. Two modes of operation are provided, The digital and the TV mode. In the digital mode, the 7912 digitizes the waveform signal and stores it for internal processing or for output on the IEEE 488 bus. Analog outputs are also available to display the digitized waveform on an X-Y-Z monitor (Tektronix 604). In the TV mode, the 7912AD converts

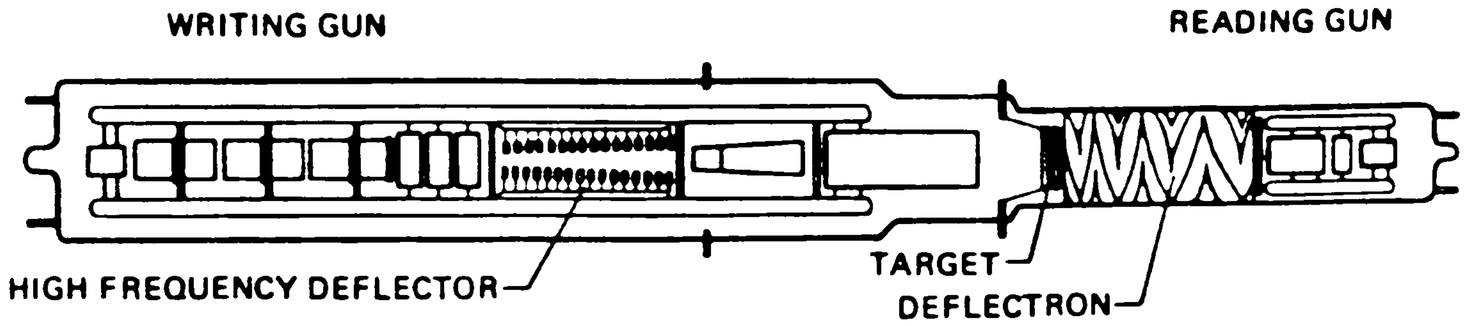


Fig. 3.1a Tektronix 7912AD Scan Converter

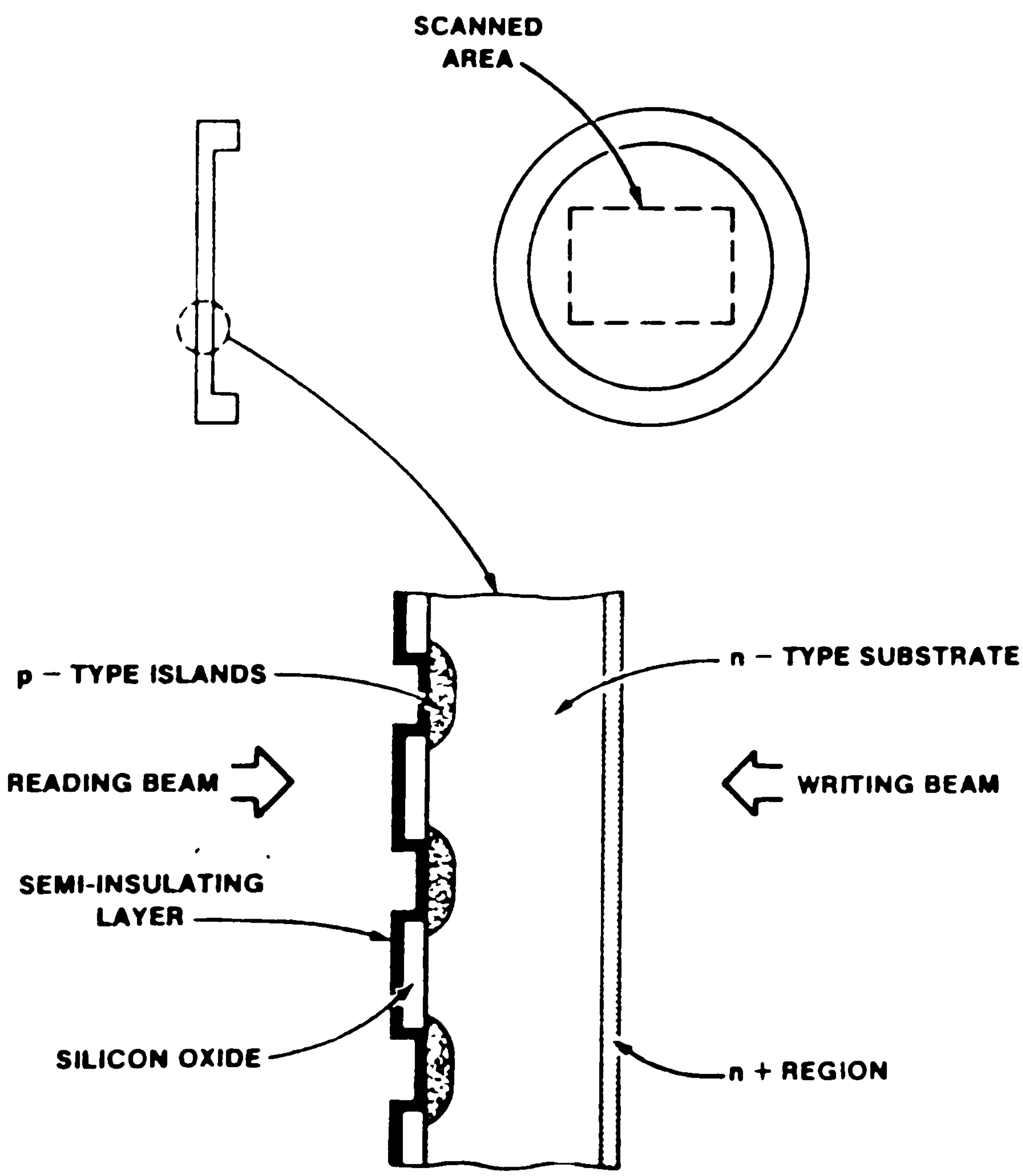


Fig. 3.1b Scan Converter Target

the waveform signal to a composite video output to be displayed on a TV monitor. The 7912AD operates much like an oscilloscope. It has both vertical and horizontal amplifier plug-ins (presently the 7A29 and the 7B80). With the Tektronix 7A21N plug-in installed, this unit can digitize an analog signal with a bandwidth of up to 1 GHz. The time window of the signal can be selected between 10 milliseconds and 5 nanoseconds using the Tektronix 7000-series time base plug-in. The equivalent sampling rate ranges from 50 kHz to 100 GHz.

Sec. 3.4 HP 12050A fiber optic data link

When a pulsed power experiment is in progress, a lot of electromagnetic interference will be generated. To ensure the integrity of the measurement and to protect digital equipment against induced voltages, the computer system should be located in an area away from the experiment. According to the IEEE - 488 standard, the maximum distance permitted between the controller (e.g. 11/34) and an IEEE - 488 device is 20 meters using a GPIB cable. The HP fiber optic data link unit is used to overcome the distance limitation.

The HP 12050A fiber optic data link provides the following features:

1) extension of the GPIB up to 100 meters
2) high speed, real time performance to a data transfer rate of 20,000 bytes/second

3) excellent noise immunity in a high EMI environment. With the fiber optic system installed, it is possible to locate the computer in an area that is far away from the high voltage experiment area.

The fiber optic data link requires two 12050A units, one at each end of the link, functioning as transmitter/ receiver and connected by a dual fiber optic cable. When commands or data are sent, the GPIB format signals are converted into a bit serial stream that is transmitted optically. At the remote site, the optical signal is converted back to the electrical GPIB format. The HP 12050A is controlled by a microprocessor within the unit. When installed, the HP 12050A is transparent to the 11/34 and the 7912AD. No software which controls the 7912AD needs to be modified.

Sec. 3.5 The CAMAC system

The CAMAC system[2,5] consists of a crate, a crate controller (Standard Engineering Co. DCC-11) and a 32 channel scanning DVM module (LeCroy 2232A). In order to interconnect the PDP 11/34 system and the CAMAC system, the Unibus is brought outside the computer cabinet via Unibus extension cable. The extension cable interconnects the Unibus and the crate. Since the crate becomes the remote end of the nibus, the unibus terminator (M9302) should be installed at the crate end.

CAMAC crate - the CAMAC crate is a compartment that houses the individual CAMAC instrument modules. It supports up to 25 modules and supplies ± 6 VDC and ± 24 VDC to the modules. The back of the crate is the dataway[2,5] which connects the CAMAC modules electrically. The dataway is a set of wires that transfers address, data, and control signals among the CAMAC modules. More information can be found in Refs. 2 and 5.

Crate controller - Because of the different structures, the dataway and the Unibus are not directly compatible. A Standard Engineering Co. DCC-11 crate controller is used to interface the two buses. The DCC-11 controller maps the CAMAC modules onto a block of 1024 Unibus addresses. Each module will have its own address and can be accessed by the 11/34. The DCC-11 also decodes instructions from the Unibus and sends out CAMAC command codes on the dataway to control the CAMAC instrument modules. The DCC-11 controller also acts as a data buffer during data transfer between the CAMAC modules and the 11/34 computer.

At present, two 8-channel digitizers and one 32-channel DVM modules are included in the CAMAC system but the CAMAC digitizers are not in use. In order to incorporate the two 8-channel digitizers into the system, MACRO-11[6] assembly routines must be written and linked to the system software.

Terminals (LA36, Tektronix 4025) - The LA36 is a general purpose printer terminal capable of printing 30 characters per second. It has a standard ASCII keyboard and a control keypad. The baud rate (300 at present), capital character lock, etc. can be set by the control keypad. Detailed information can be found in the LA36 User's Manual.

The Tektronix 4025 is a microprocessor-based graphics terminal using an ASCII keyboard for input and a CRT for output. It has 4 k-bytes of local memory and contains its own firmware (program stored in ROM). Functions such as data transmission and reception rates, moving text line up and down the CRT screen, etc. can be set by a program or through the use of the keyboard. For plotting graphs, the screen is divided into two halves. The upper half is the "work space" where graphs are plotted. The lower half is the "monitor" where commands entered by the operator or messages sent by the computer are printed.

Switch panel - A switch panel is installed above the disk drive to enable the operator to select the use of terminals (LA36 or 4025). There are also three micro-switch packs, hard-wired to the DL11-W interface board, to allow the change of baud rates, data bit length, and parity check. Further information can be found in Ref. 5.

Optical isolator - The 4025 terminal is usually used as a remote control terminal. To protect it against demage by the large induced voltage during experiment, an optical isolator is installed between the 11/34 and the 4025 terminal. Refer to Ref. 5 for further information.

Communication interface - The main function of the interface is to selectively connect the terminal between two computers. It can also be used to monitor the data flow between computer and terminal. At present, this unit is used only to monitor data flow.

CHAPTER IV

SOFTWARE

Sec. 4.1 Introduction

The system software used in this data acquisition system is the TEK SPS BASIC V02[7]. It is a general-purpose programming language for control of instrument plus the acquisition processing, storage, and display of data. It gives the programmer access, through BASIC, to features of the operating system usually accessible only through assembly-level interaction.

A graphics package is available to help display data in an easy-to-understand format. A set of signal analysis commands is also available that greatly simplifies computations such as Fourier Transforms, differentiation, and integration. A General Purpose Interface Bus (GPIB) driver (instrument control routine) is included in the SPS BASIC enabling easier control of the 7912AD programmable digitzer.

The data storage program consists of a number of BASIC routines that operates the 7912AD, acquires data from the digitizer and does the error checking. During the execution of this program, commands are entered by the operator through the terminal keyboard and executed by subroutines. Subroutines addresses are stored in an address table. New command subroutines can be added to the main program by writing the new routines and adding the new address to the address table. This chapter will discuss the structure and flow of the main storage program, supporting programs and the command subroutines.

Sec. 4.2 System Information

The data storage program has been written to run on the 11/34 system as described in Sec. 3.2 of Ch. III. if there is any change in the system hardware, corresponding modification of the software must be made. For example, if the cartridge disk drive RL01 is used instead of the floppy disk drive RX01, all the "DX's" appearing in the program must be changed to "DL". The RX01 floppy disk drive is a file structure device. Manipulation of files on a disk can be done by using the SPS BASIC commands or by execution of the storage program. A directory (table of file names and pointers to where files are stored) is installed on each disk. Files are accessed by searching the directory for the file name and using the associated pointer to locate the file on a disk. As each file is stored or canceled on the disk, the directory is updated to indicate the new or canceled file.

A directory of the disks used with routines is shown in Table 4.1. The directory, which is printed with the "DIR" command, contains the names and extension of all program and all data files with the number of blocks they occupy and date of creation. Before using a brand new disk, it must be initialized with the "ZERO" command (this is the command that "erases" the contents of the medium). After

```
DIR DX0:

 19-NOV-80
DX0:
TRAN  .OVL     5  8-FEB-80
UNTRAN.OVL     5  8-FEB-80
CLK   .SPS     2  8-FEB-80
DX    .SPS     7  8-FEB-80
GPI   .SPS     5  8-FEB-80
KBG   .SPS     5  8-FEB-80
KBN   .SPS     4  8-FEB-80
KBT   .SPS     4  8-FEB-80
LP    .SPS     2  8-FEB-80
VM    .SPS     6  8-FEB-80
ABORT .SPS     1  8-FEB-80
ATAN2 .SPS     2  8-FEB-80
BOOT  .SPS     1  8-FEB-80
CANCEL.SPS     1  8-FEB-80
CHAIN .SPS     1  8-FEB-80
CHANGE.SPS     4  8-FEB-80
CLEAR .SPS     1  8-FEB-80
COPY  .SPS     3  8-FEB-80
DATE  .SPS     1  8-FEB-80
DEFINE.SPS     1  8-FEB-80
DIR   .SPS     3  8-FEB-80
EOF   .SPS     1  8-FEB-80
FORMAT.SPS     1  8-FEB-80
GET   .SPS     1  8-FEB-80
GETBLK.SPS     1  8-FEB-80
GETFRE.SPS     1  8-FEB-80
GETLIN.SPS     1  8-FEB-80
GETLOC.SPS     1  8-FEB-80
GETPRI.SPS     1  8-FEB-80
GETSTA.SPS     1  8-FEB-80
OINPUT.SPS     2 15-AUG-80
OPRINT.SPS     2 15-AUG-80
OSET  .SPS     1 15-AUG-80
VARCLR.SPS     1 15-AUG-80
CONFIG.INF     2  1-OCT-80
PARM1 .LST     4  1-OCT-80
SPSDX .LDA    83  8-FEB-80
LOADER.SAV     5  8-FEB-80
DIFF  .SPS     2  8-FEB-80
INT   .SPS     1  8-FEB-80
POLAR .SPS     3  8-FEB-80
RFFT  .SPS     7  8-FEB-80
DISPLA.SPS     2  8-FEB-80
DRAW  .SPS     1  8-FEB-80
DRAWON.SPS     1  8-FEB-80
GIN   .SPS     1  8-FEB-80
GRAPH .SPS     9  8-FEB-80
INITG .SPS     1  8-FEB-80
MOVE  .SPS     1  8-FEB-80
PAGE  .SPS     1  8-FEB-80
RDRAW .SPS     1  8-FEB-80
RESETG.SPS     1  8-FEB-80
RMOVE .SPS     1  8-FEB-80
RSDRAW.SPS     1  8-FEB-80
RSMOVE.SPS     1  8-FEB-80
SDRAW .SPS     1  8-FEB-80
SETGR .SPS     2  8-FEB-80
SEEVIE.SPS     1  8-FEB-80
SEEWIN.SPS     1  8-FEB-80
SGIN  .SPS     1  8-FEB-80
SMOVE .SPS     1  8-FEB-80
VIEWPO.SPS     1  8-FEB-80
WINDOW.SPS     1  8-FEB-80
READ1 .BAS     2  2-OCT-80
```

Table 4.1a Directory of System Disk DX0

SETDAT.SPS	1	8-FEB-80
SETTIM.SPS	1	8-FEB-80
SIFCOM.SPS	1	8-FEB-80
SIFLIN.SPS	1	8-FEB-80
SIFTO .SPS	1	8-FEB-80
SQUISH.SPS	7	8-FEB-80
STATUS.SPS	3	8-FEB-80
STERMC.SPS	1	8-FEB-80
SYSBLD.SPS	4	8-FEB-80
TIFL .SPS	1	8-FEB-80
TIME .SPS	1	8-FEB-80
UNSCHE.SPS	1	8-FEB-80
VARTST.SPS	1	8-FEB-80
VERSIO.SPS	1	8-FEB-80
WAIT .SPS	1	8-FEB-80
WASCII.SPS	2	8-FEB-80
WBYTE .SPS	1	8-FEB-80
WHEN .SPS	1	8-FEB-80
WRITE .SPS	1	8-FEB-80
WRITEU.SPS	2	8-FEB-80
ZERO .SPS	1	8-FEB-80
EDGEAD.SPS	2	15-AUG-80
NORMAD.SPS	3	15-AUG-80
MAPAD .SPS	6	15-AUG-80
ZREF .SPS	1	15-AUG-80
REJECT.SPS	1	15-AUG-80
INSTAD.SPS	3	15-AUG-80
IFCTIM.SPS	2	15-AUG-80
BITCMP.SPS	2	15-AUG-80
BITCLR.SPS	1	15-AUG-80
BITSET.SPS	1	15-AUG-80
BITTST.SPS	1	15-AUG-80
GETR5 .SPS	1	15-AUG-80
IV .SPS	2	15-AUG-80
GIFES .SPS	1	8-FEB-80
HOOK .SPS	4	8-FEB-80
HOOKQ .SPS	4	8-FEB-80
IFDTM .SPS	1	8-FEB-80
IGNORE.SPS	1	8-FEB-80
INPREQ.SPS	1	8-FEB-80
INPUT .SPS	2	8-FEB-80
LIST .SPS	3	8-FEB-80
LISTVA.SPS	3	8-FEB-80
LOCKKB.SPS	1	8-FEB-80
MATCH .SPS	1	8-FEB-80
ONERR .SPS	1	8-FEB-80
OVERLA.SPS	1	8-FEB-80
OVLOAD.SPS	2	8-FEB-80
OVLSAV.SPS	2	8-FEB-80
POLL .SPS	2	8-FEB-80
PPOLL .SPS	1	8-FEB-80
PRINT .SPS	2	8-FEB-80
PRIORI.SPS	1	8-FEB-80
PUT .SPS	1	8-FEB-80
PUTBLK.SPS	1	8-FEB-80
PUTLOC.SPS	1	8-FEB-80
RANDOM.SPS	1	8-FEB-80
RASCII.SPS	2	8-FEB-80
RBYTE .SPS	1	8-FEB-80
READU .SPS	2	8-FEB-80
RENAME.SPS	1	8-FEB-80
RENUM .SPS	2	8-FEB-80
REPLAC.SPS	1	8-FEB-80
RESCHE.SPS	1	8-FEB-80
RESET .SPS	1	8-FEB-80
REWIND.SPS	1	8-FEB-80
SAVE .SPS	1	8-FEB-80
SCHEDU.SPS	2	8-FEB-80

Table 4.1b Directory of System Disk DX0

```
READ1 .BAS      2  2-OCT-80
STORE1.BAS      3  2-OCT-80
STATS1.BAS      2  2-OCT-80
GRAPH1.BAS      3  2-OCT-80
FILTR1.BAS      4  2-OCT-80
ANALYS.BAS      7  2-OCT-80
CATDIR.BAS      3  2-OCT-80
< UNUSED >      7
CREATE.BAS      3  2-OCT-80
DATACQ.BAS     18 31-OCT-80
CURFIT.BAS      7 18-NOV-80
< UNUSED >     87
    94 FREE BLOCKS

READY
*
```

Table 4.1c Directory of System Disk DX0

```
DIR DX1:

 19-NOV-80
DX1:
INITIA.LIZ      1  1-SEP-80
< UNUSED >      0
CATALO.G        2  2-OCT-80
< UNUSED >      6
TITLE .TXT      1  2-OCT-80
< UNUSED >    470
   476 FREE BLOCKS

READY
*_
```

Table 4.2 Directory of Data Disk DX1

initialization, the disk will contain 480 free blocks for storing either programs or data (A block is a space which can store 512 bytes of information).

The system disk (DXO) contains the required SPS programs, instrument drivers, graphic and signal processing package, storage programs and supporting programs. The disk in drive DX1 is a specially formatted disk for data storage.

Files on the system disk shown in Table 4.1 can be classified into four categories:

1) files required by the SPS BASIC system
2) files required to run the storage routines
3) support programs
4) Analysis programs

Files in the first category include:

All files with extension ".SPS"

SPSDX.LDA LOADER.SAV

Files in the second category include:

DATACQ.BAS STORE1.BAS READ1.BAS

GRAPH1.BAS STATS1.BAS

Files in the third category include:

CATDIR.BAS CREATE.BAS

Files in the fourth category include:

ANALYS.BAS CURFIT.BAS FILTRI.BAS

Sec. 4.3 Data Disk

Before using a blank disk, it must be initialized with the "ZERO" command. If the disk is used as a data disk, it must be formatted by the "CREATE.BAS" program. This program creates the files as shown in Table 4.2.

The "TITLE.TXT" file contains the title of the disk which will always have the form:

DATXXX (e.g. DAT001)

where "XXX" is a three digit sequence number. It also contains the date of initialization and any comments (up to 400 characters) entered by the operator during the execution of the program "CREATE.BAS".

When a file is canceled or updated, its physical location may change. If this process is performed frequently, holes of disk space due to the deletion of a file will be left behind. To prevent this from happening, the program "CREATE.BAS" creates the files "TITLE.TXT", "CATALO.G" and "INITIA.LIZ" on top of the disk with 8 blocks of blank space reserved for the updating of these files. A listing of "CREATE.BAS" and a line by line discussion is included in Appendix A. An example of the execution of this program is shown in Fig. 6.7.

The file "INITIA.LIZ" on the data disk contains the date and shot number of the last data file. The shot number is automatically incremented and the "INITIA.LIZ" file is updated each time data is stored on disk. If the storage program is restarted after the system has been crashed or turned off, the software checks the "INITIA.LIZ" file and asks the operator to verify the shot number. Thus, chances of losing data are greatly reduced.

The "CATALO.G" file on the data disk contains a directory of all data files on that disk. The file name for each data file is created by the storage routine in the form as follows:

File name = YYMMDD.XXX e.g. (800901.001)

where: YY = the last two digits of the year

MM = the number of the month (1-12)

DD = the day of the month (01-31)

XXX = shot number.

Zeros are added when neccessary to ensure that the file name contains nine digits. The "CATALO.G" can be updated by running the program "CATDIR.BAS". A listing of "CATDIR.BAS" is included in Appendix A and an example of the execution of this program is shown later in Fig. 6.8.

Sec. 4.4 BASIC system programs

The data acquisition system software consists of system support programs, storage routines and analysis programs. The system support programs "CREATE.BAS" and "CATDIR.BAS" have been discussed in the last section. The storage routines include five BASIC programs and two files:

DATACQ.BAS READ1.BAS STORE1.BAS

GRAPH1.BAS STATUS.BAS CONFIG.INF

PARM1.LST

"DATACQ.BAS" is the main program that contains routines to control the operation and data transfer of the 7912AD digitizer. It also permits other programs with ".BAS" extension to be overlaid onto a specified area of the main program. The result of an overlay is equivalent to typing the new program in from the keyboard. The old lines with corresponding line numbers in the overlay will be replaced by the new lines. All variable values and dimension statements are preserved and execution resumes at the first line of the overlay segment. Lines 20000 to 20260 are designated to be the overlay area. The programs requiring overlay are the graphing program (GRAPH1. BAS), the storing program (STORE1.BAS), the data file reading program (READ1. BAS) and the 7912AD status reading program (STATS1.BAS).

Sec. 4.5 DATACQ.BAS

This program contains the 7912AD control routines, the commands address table, most of the simple command routines and an initialization routine that erases itself after execution. During execution of the program, it first defines a few string variables and constants. Then five keys of the 4025 terminal are "learned"[8] (programmed to perform a special function). The date and time of the system are checked; if they are not set, an error message is printed on the terminal and the program stops. The operator must enter the correct date and time in order to rerun the program. The operator is then asked to verify the shot number and the "INITIA.LIZ" file is updated. The contents of the hardware configuration file are printed in the workspace of the 4025 terminal to review the hardware setup to the operator. The routine also sets the default crate address and station number, defines three waveform arrays, sets the addresses of the 7912AD and erases the first 51 lines except lines 300 to 340. A flow diagram of the intialization portion of this program is shown is Fig. 4.1. A list of DATACQ.BAS and the discussion is included in Appendix A.

Sec. 4.6 Hardware Configuration File

A typical hardware configuration file is shown in Fig. 4.2. It is printed in the workspace by the initialization portion of the DATACQ.BAS for the operator to check against the actual configuration of the system. The file provides information about the Tektronix digitizer and the CAMAC modules connected to the system. The Tektronix system information indicates that a 7912AD is tied to the system. It also shows that it is a IEEE 488 device interfaced to the computer by the interface card labeled #0. The 7912AD can be accessed by using (0+32), (0+64) and (0+96) as listener, talk and secondary address. The CAMAC system information indicates that three units are presently installed in the crate: two 8-channel digitizers and one 32-channel DVM. They use the CAMAC 583 bus to transfer data between the computer and the crate. The CAMAC module locations are described by the crate number and slot number.

Sec. 4.7 STORE1.BAS

After DATACQ.BAS has been loaded from system disk into

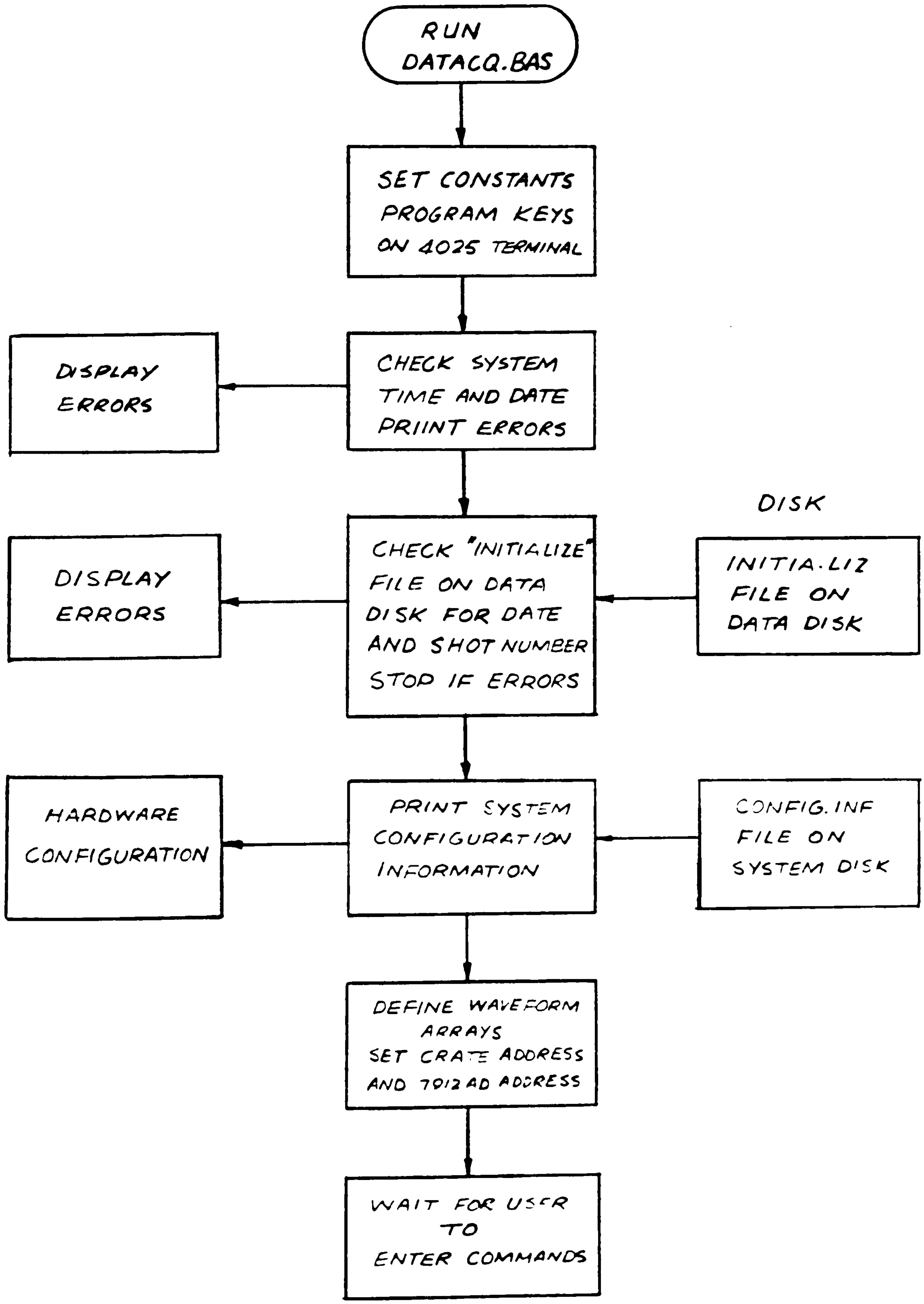


Fig. 4.1 Flow Diagram of the Initialization Portion of DATACQ.BAS

```
                    *****CAMAC SYSTEM*****
UNIT # ,  BUS #  , CRATE # , SLOT #  ,MNF. TYP# POSTRIG ,SERIAL # ,
     1,     583,         2,        6,     2264,       1,   169436
     2,     583,         2,       16,     2264,       2,   169995
     3,     583,         2,       15,   2232.1,      00,   169994
                   *****TEKTRONIX SYSTEM*****
UNIT # ,  BUS #  , INTF. # ,MNF.TYP# , L. ADDR., T. ADDR., S. ADDR.,
     1,     488,         0,   7912AD,     0+32,     0+64,     0+96,
```

Fig. 4.2 Hardware Configuration File

memory, it contains a designated area (2000 - 20260) for new program segment overlay. When commands "GRAPH", "STORE", "READ" OR " STATUS" are executed, the corresponding program will be overlaid and execution begins at line 20000. After a transient waveform has been digitized by the 7912AD, data can be stored on the data disk (DX1) by executing the "STORE" command. Lines 20000 to 20070 of the "STORE1.BAS" check the shot number, covert the date from "DO-MMM-YY" to "YYMMDD" format, and print the parameter file into the workspace for modification. The data header is assembled by combining the date, time, shot number, and operator's name with the modified parameter file. The data header and the waveform data are put together to become the data file which is stored on DX1 with a data file name "YYMMDD.sht#" (e.g. 800901.001). Before the data file is stored, the program checks for error. If an error exists, the store operation is aborted and the program returns to the "ENTER COMMAND" state. Errors must be corrected by the operator before the data file can be stored on the data disk by using the store command again. A flow diagram of STORE1 is shown in Fig. 4.3 and a listing is included in Appendix A.

Sec. 4.8 Data File

The data file consists of two parts, the data header and the binary waveform data. The first part contains information about the date, time, shot number and the name of the operator. It also contains the parameter file which describes the conditions for which the shot is fired. An

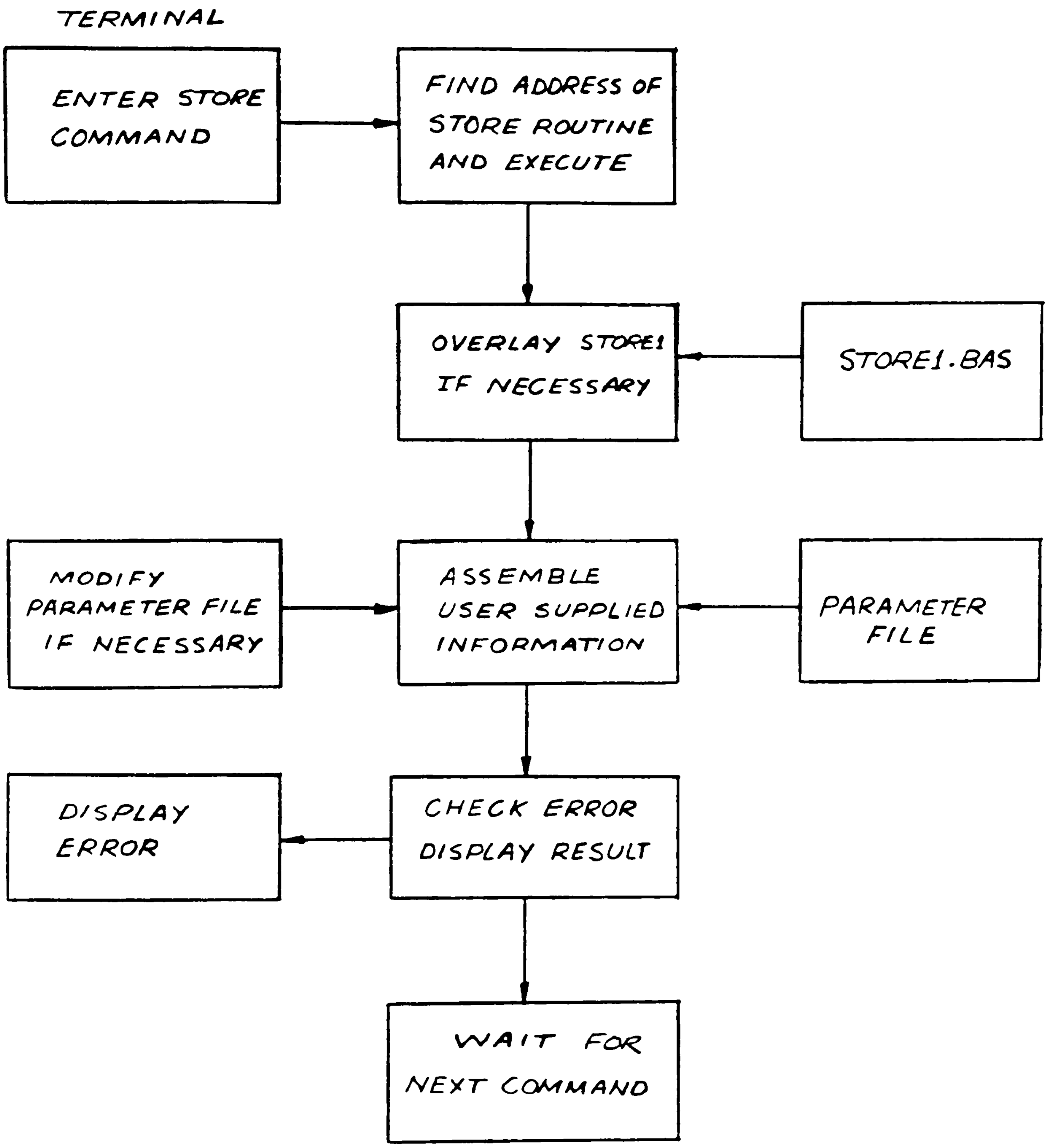


Fig. 4.3 Flow Diagram of STORE1.BAS

example of the parameter file is shown in Fig. 4.4. This file describes the parameters of a laser triggered spark gap experiment. It records the gas pressure and mixture of the chamber, the charging voltage of the Marx bank, the gap voltage, the laser power delivered to the gap and the cable delays. All this information can be entered from the keyboard before the data of a shot are stored.

The second part of the data file comes from the 7912AD. It contains the vertical and horizontal scales and units (front panel settings of the 7912AD plug-ins), and the binary waveform data.

Sec. 4.9 GRAPH1.BAS

Data stored on the data disk can be graphed and displayed in the workspace of the 4025 terminal by executing the "GRAPH" command. The corresponding program being overlaid by this command is "GRAPH1.BAS". Up to three data curves can be graphed by this command. An example of the execution is shown later in Fig. 6.4. A flow diagram of this program is shown in Fig. 4.5 and a listing is included in Appendix A.

Sec. 4.10 READ1.BAS

Data file stored on the data disk can be read by the execution of the "READ" command. The program "READ1.BAS" is overlaid into the main program segment by this command. It reads a data file, prints the data header in the monitor area of the 4025 terminal and plots the data waveform with

```
1001
1002 CHAMBER
1003   GAS ID#  PRESSURE(T)  GAS NAME  PRESENTAGE(%)
1004       01,        960,  NITROGEN,       50,
1005       02,        960,     ARGON,       50,
1006
1007 MAX BANK
1008   CHAR.VOLT.(KV)  TIME CONST.(NS)  TRIG.TIME
1009        250,              250,          —MS,
1010
1011 GAP VOLTAGE(KV) BEFORE LASER TRIGGERED
1012        200,
1013
1014 LASER
1015   TYPE    POWER(MW)  PULSE WITH(NS)  OPTICAL FIBER
1016   RUBY,       2,             15,          1MM
1017
1018 CABLE DELAYS(NS)
1019   POC.CEL  dI/dt  FAST TRIG.  LASER P. V. RAM
1020      40,     30,     20,          20,    250,
1021
1022
1023
1024
1025
1026
1027
1028 COMMENTS
1029 TYPICAL LASER TRIGGERED SHOT
1030 END OF PARM FILE
```

Fig. 4.4 Typical Parameter File for Laser Triggered Spark Gap Experiment

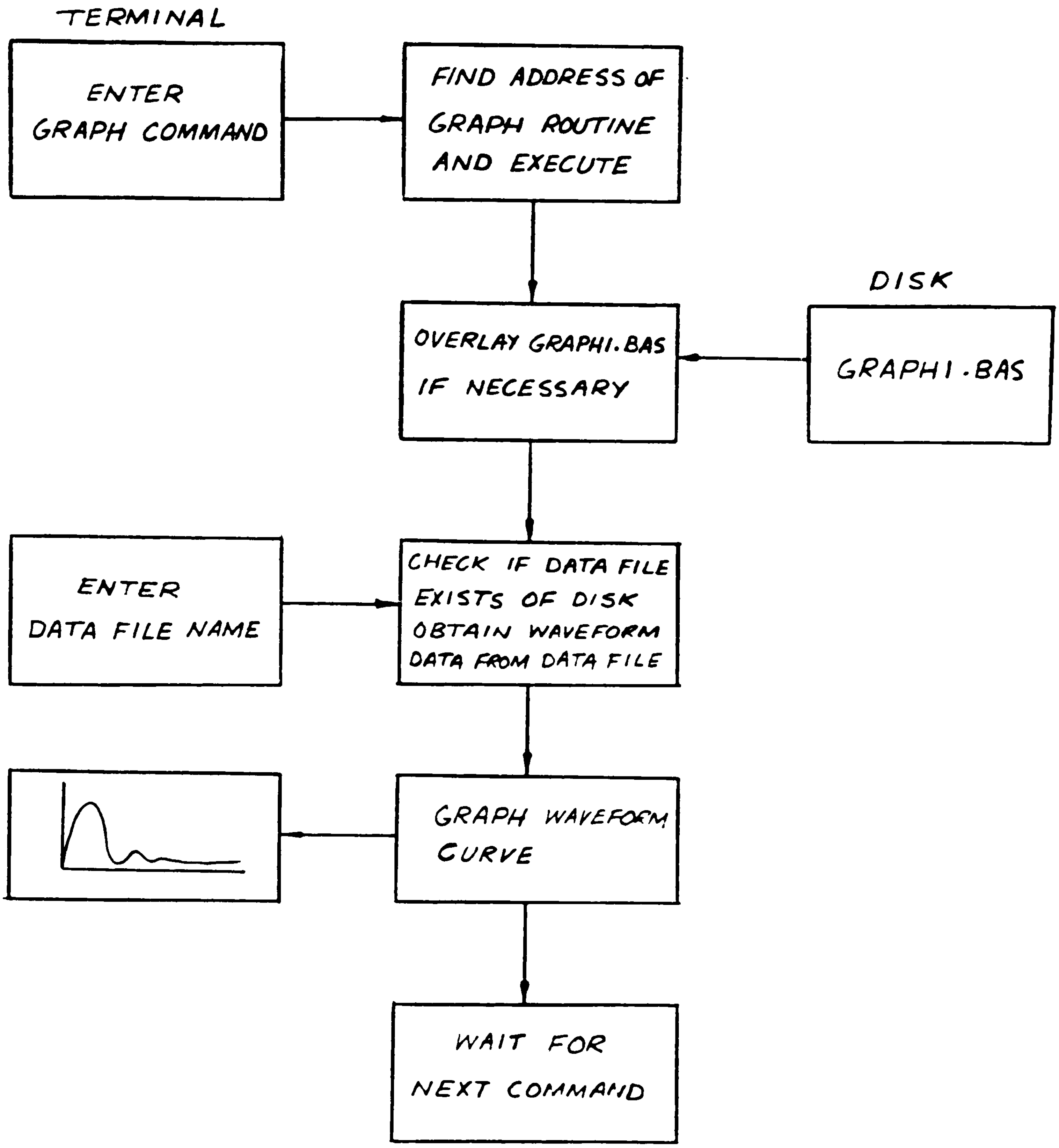


Fig. 4.5 Flow Diagram of GRAPH1.BAS

vertical and horizontal labels in the workspace. This program contains an error handling routine that prints out error messages on the terminal and transfers control to the "ENTER COMMAND" state if an error is found. A flow diagram is shown in Fig. 4.6 and a listing is included in Appendix A.

Sec. 4.11 STATS1.BAS

STATS1.BAS is also one of the BASIC program that is overlaid into the main program when the "STATUS" command is executed. It reads the status register of the CP1100 interface board (discussed in Sec. 3.2 Ch.IV) and decodes the content. The status messages of the 7912AD are printed on the 4025 terminal. A listing and description of STATS1.BAS are included in Appendix A.

Sec. 4.12 System Support Programs

Two BASIC programs initialize and organize the data disks. "CREATE.BAS" initialize a blank disk, creates the "TITLE.TXT", "INITIA.LIZ" and "CATALO.G" files. This program allocates the first ten blocks for these three files and the rest of the disk for storing data files.

"CATDIR.BAS" is usually run at the end of the day after the experiment has been shut down. This program extracts the directory of the data disk and stores it in the CATALO.G file. The CATALO.G file then contains the updated information of the data files stored on the data disk. A listing of these two programs is included in Appendix A.

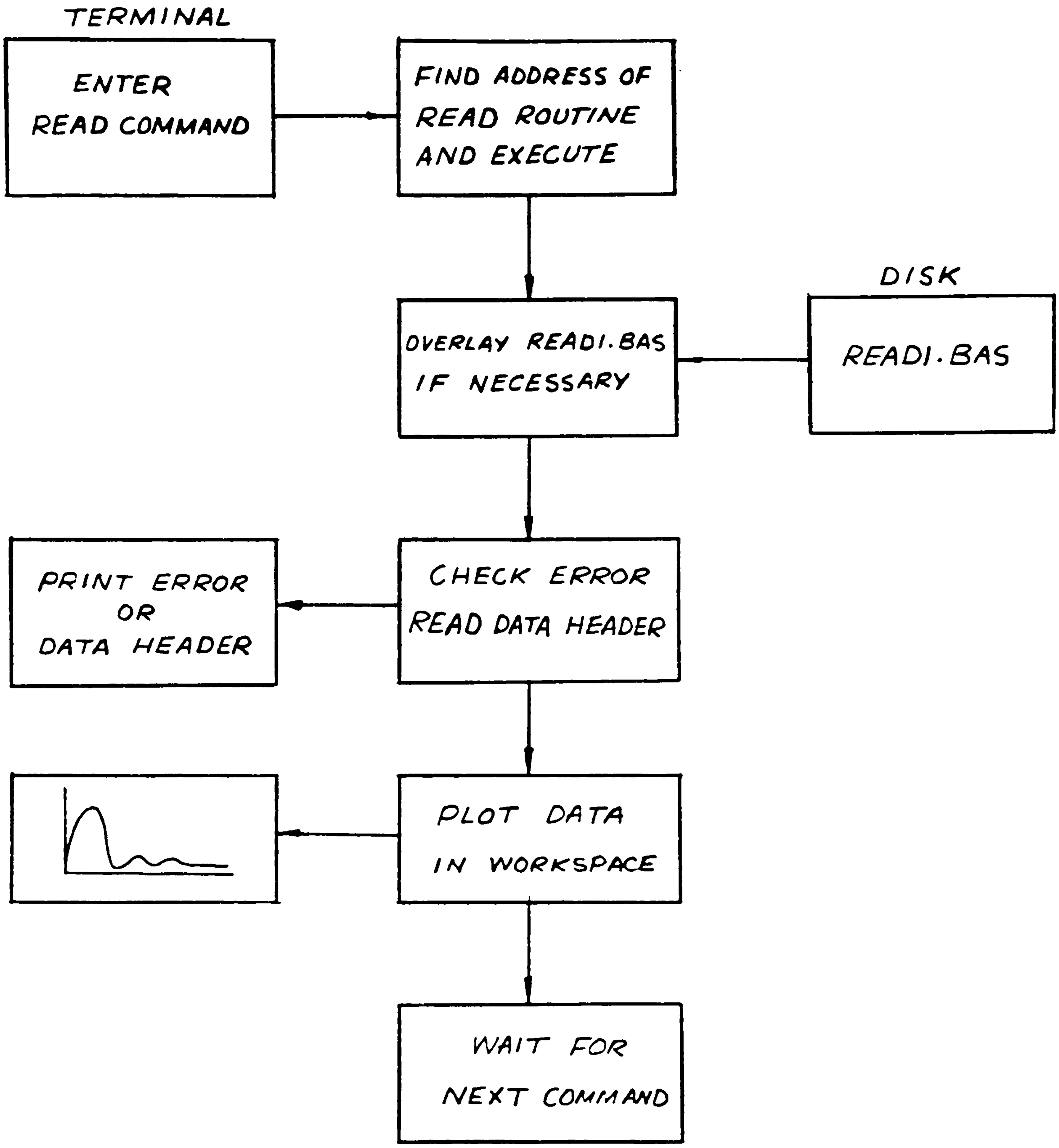


Fig. 4.6 Flow Diagram of READ1.BAS

Sec. 4.13 Analysis Programs

Three analysis programs have been written to analyze the experimental data. Data can be retrieved from the data disk and analyzed numerically. Such analysis includes integration, differentiation, locating the peak valve, least square polynomial curve fitting, and filtering of waveform data. Examples of the execution of these programs are shown in Chapter VI. Listings and descriptions of each of these three programs are included in Appendix A.

CHAPTER V

OPERATING INSTRUCTIONS

Sec. 5.1 Introduction

This chapter discusses the procedure to bootstrap the PDP 11/34 ccmputer, to turn on different pieces of equipment, and to run the storage routines and analysis programs.

Sec. 5.2 Turn-on and Bootstrap Procedure

Turn on AC power to all equipment. The 4025 terminal requires about 15 seconds to warm up. The equipment requiring AC power is:

1) the computer (front panel switch)
2) the LA36 terminal (AC power controlled by the same switch on the computer. When the computer is on, the LA36 is also on)
3) the 4025 graphics terminal (switch on the right side)
4) the CAMAC crate (power <u>must</u> be turned on to use computer, because the Unibus terminator is located there)
5) the 7912AD digitizer (main power switch is at the back, secondary on/off button on front panel. Before turning on the AC power, be sure the main intensity knob is at minimum, all the way counter clockwise)
6) the 12050A optical data link (if used)
7) Optical isolator (two separate supplies)

8) the communication interface

9) TV monitor

10) 604 monitor

Since the computer accepts only commands in capital letters, the capital "Lock on" key must be on at both terminals. On the LA36 terminal depress the "CAP's Lock" key (it should stay in down position), while on the 4025 terminal depress the "TTY Lock" key (an indicator light should be lit). The baud rates that the computer uses to communicate with the LA36 and the 4025 are 300 and 4800 respectively. They can be set through the micro-switch packs located at the front panel above the disk drive. The baud rate of the LA36 is also set at 300 by depressing the key "300" at the left corner of the key board. The baud rate of the 4025 can be set by typing "! BAU 4800" (see 4025 user's manual). The 4025 has 4 k-bytes of local memory. It remembers the settings even after the power is turned off. It is therefore, not necessary to set the baud rate each time the terminal is turned on.

Set the terminal select switch to "GRAPHICS"; set the communications interface for "TTY——PDP" (see Fig. 3-1 to Fig. 3-4 of Ref. 5) with:

Monitor all data: disabled

TTY2: disabled

TTY1: enabled

Put the data acquisition system disk in drive DX0 and a data disk in drive DX1. At the front panel of the computer, hold down the "CNTRL" button and press the "HLT/SS" at the same time, then press "BOOT". The computer should print four numbers on the terminal screen.

"XXXXXX XXXXXX XXXXXX XXXXXX"

"$_" ("_" is the cursor of the 4025)

Those four numbers correspond to the contents of registers 0,4,6 and 7. R0 and R4 are general purpose registers. R6 and R7 are used as the stack pointer and the program counter. These registers contain information about what the computer was doing when it halted. To run the bootstrap routine, type in:

DX (CR) ("CR" is carriage return)

The bootstrap routine will load the operating system from the system disk to memory. The SPS BASIC software is automatically loaded. It takes about 20 seconds before the computer responds by printing a message and a prompt character "*".

```
TEK SPS BASIC V02-01-COPYRIGHT 1979-TEKTRONIX, INC.
ALL RIGHTS RESERVED. SERIAL NO. 10248.
YOU HAVE 10886 WORDS OF FREE MEMORY.
READY
*
```

Now the SPS BASIC is resident and ready to accept command.

Since the date and time must be set in order to run the storage routines, it is appropriate at thispoint to set the system calender and clock by using the "SETDATE" and "SETTIME" commands. Type in:

SETDATE "DD-MMM-YY" (CR) e.g. SETDATE "29-OCT-80"

SETTIME "HH:MM:SS" (CR) e.g. SETTIME "14:30:00"

Sec. 5.3 Creating a Data Disk

Before a blank disk can be used to store data, it must be formatted by the program "CREATE.BAS". The program first initializes the disk by clearing the old directory and contents, then reserves a new space for the directory. It also creates files "TITLE.TXT", "INITIA.LIZ" and "CATALO.G". To format a new data disk, type in:

OLD DX: "CREATE.BAS" (CR)

The system will load "CREATE.BAS" from the system disk to memory, and respond with a prompt character "*". Insert the new data disk in drive DX1 and type "RUN" to execute the program. The program will ask for information such as disk number and comments during execution. Several disks can be formatted at this time. An example of the execution of this program is shown later in Fig. 6.7.

Sec. 5.4 Running of the Storage Routine

Both the main program "DATACQ.BAS" and the 7912AD driver "GPI.SPS" must be resident to run the storage routine. To load the 7912AD driver, type in:

LOAD "GPI.SPS" (CR), ".SPS" is optional

The system will respond with "*". Then load "DATACQ.BAS" by typing in:

OLD DX: "DATACQ.BAS" (CR)

It takes approximately 20 seconds to load this program.

The system prints "*" to show that "DATACQ.BAS" has been loaded. Type in:

RUN (CR)

to execute the program. The program first defines a few constants and four keys of the 4025 by using the "LEARN" (refer to 4205 user's manual) statements. The four keys are:

F1 - CURSOR CONTROL KEY;
Shift F1, cursor move from monitor to workspace.

F6 - LIST KEY, list the program from memory onto the workspace.

F7 - EDIT KEY, allow modification to be made on the contents in workspace.

F8 - SEND KEY, send contents from workspace to computer.

The program then checks date and time. If they are not set, the system prints an error message and asks the operator to set the date and time. When these conditions are met, the system prints the hardware configuration file (see Fig. 4.2) on the workspace and asks the operator to enter parameter file and the operator's name.

The first portion of the program automatically erases itself after execution. If the program halts (transfers control back to BASIC) due to execution of illegal commands entered by the operator, it can be restarted by typing "GO TO 1000" (CR). If a "CLEAR" command has been excuted while the system is in "BASIC" control, the "DATACQ.BAS" must be reloaded from the system disk and restarted by typing in:

RUN (CR)

The execution of some system commands requires a new program segment to be overlaid. These commands are "GRAPH", "STORE", "READ" and "STATUS". When these commands are entered, the system loads a new program segment from the system disk and overlays it onto a reserved area in memory. Execution starts at the first line of the overlay.

Sec. 5.5 System Commands

There are altogether 17 system commands, 13 from the main program "DATACQ.BAS" and 4 from the analysis program "ANALYS.BAS". These commands can be executed in response to an "ENTER COMMAND" prompt from the main program or a "DATA FILE" prompt from "ANALYS.BAS". Commands are recognized either by the first three letters or the first letter. The program checks the first three letters of a command entry with the address table (lines 1010 to 1050 in "DATACQ.BAS"). If no match is found, the command is checked again with the single lettered command address table (lines 1110 to 1120). The command will be executed if it matches with those in the address tables. Otherwise, the system asks for new command entry again. The analysis program commands work in a similar way. The available system commands are listed below and in Appendix B.

In the following discussion, (CR) indicates the carriage return key. All messages printed by the computer are in double quotation marks.

Three Lettered Commands (DATACQ.BAS)

ACQUIRE: Initiates the 7912AD to digitize waveform data, and obtains data from the 7912AD local memory. To complete this operation, some of the settings on the 7912AD have to be adjusted. An example of the use of this command is shown later in Fig. 6.3.

CLEAR: Clears all registers and local memory of the 7912AD digitizer. The listener and secondary addresses must be supplied by the operator.

Example: CLE (CR)

"LISTENER ADDRESS?" 64 (CR)

"SECONDARY ADDRESS?" 96 (CR)

DEFECT: With the main intensity at minimum, the 7912AD digitizes the target 25 times and records any target defect data. If no defect is found, the computer prints "NO DEFECTS ON TARGET" and then executes the ACQUIRE command automatically.

Example: DEF (CR)

"DIGITIZE CRT TARGET DEFECTS 25 TIMES"

"NO DEFECTS ON TARGET"

GRAPH: Plots the stored data from the disk into the workspace of the 4025 terminal. Up to three graphs can be plotted in the workspace.

Example: GRA (CR)

"DATA FILE?" 800901.001 (CR)

"TYPE STOP AFTER DATA FILE?"

"IF NO MORE GRAPH TO PLOT"

"DATA FILE?" STOP (CR)

NAME: Stores operator's name

Example: NAM (CR)

"ENTER OPERATOR'S NAME:-?" JOHN (CR)

PARMETER: Prints a parameter file into the workspace of the 4025 and allows the operator to make changes, and saves the new file (or replaces the old file by using the same name).

Example: PAR (CR)

"PARAMETER FILE?" PARM1.LST (CR)

"MAKE CHANGES NOW THEN ANSWER (Y/N)"

(by moving cursor to workspace, return to monitor when finished)

"SAVE NEW FILE?" ANY NAME (CR)

(use of the same name will replace the old file with the new one)

READ: Overlays "READ1.BAS" from system disk and plots data of data file in workspace and prints data header in monitor.

Example: REA (CR)

"DATA FILE?" 800901.001 (shot #1 of 9/1/80)

STATUS: Checks the status of the 7912AD digitizer and prints one or more of the following messages.

COMMAND ERROR

EXECUTION ERROR

POWER FAIL

INTERNAL ERROR

POWER UP

SERVICE REQUEST

OPERATION COMPLETE

STORE: Stores the waveform data currently digitized by the 7912AD digitizer. The date, shot number, operator's name and the parameter file chosen by the operator are assembled and stored as the data header. Example of the execution of this command is shown in Fig. 6.5.

One Letter Commands (DATACQ.BAS)

Only the first letter of these commands needs to be entered.

CRATE: Changes the default crate and slot numbers. The present crate and slot numbers are 2 and 15.

Example: C (CR)

"ENTER CRATE #?" 2 (CR)

"ENTER SLOT #?" 15 (CR)

(module located at slot 15 of crate 2)

ALL: Reads all 32 channels of the 32 channel DVM. Stores the readings in an array N(1) to N(32) and prints them on the 4025 terminal.

HALT: Stops the execution of the program and returns control to BASIC.

VOLT: Reads the value of a single channel of the 32 channel DVM selectively.

Example: V (CR)

"ENTER CHANNEL #?" 1 (CR)

"CHANNEL 1 V=5 volts"

Commands Recognized by "ANALYS.BAS"

PLOT: Plots data files onto workspace of 4025 terminal. Up to 5 curves can be plotted with the same coordinates (see Fig. 6.9).

Example: PLOT (CR)

"NO OF GRAPHS?" 3

"DATA FILES"

"?" 800901.001 (CR)

"?" 800901.002 (CR)

"?" 800901.003 (CR)

PEAK: Prints the maximum value of a waveform on the workspace (next to the curve).

INTEGRATE: Performs numerical integration on the waveform data and plots the result onto the workspace of the 4025.

Example: INT (CR)

"1st or 2nd INTEGRAL? 1 or 2?" 1 (CR)

DIFFERENTIATE: Performs numerical differentiation on the waveform data and plots the result onto the workspace of the 4025.

Example: DIF (CR)

"1st or 2nd DERIVATIVE (1 or 2)?" 1 (CR)

Sec. 5.6 Recovery From a Crash

Error checking and handling routines have been written into the storage program. When the operator enters invalid commands or tries to access a nonexistant data file, the system prints an error messages and asks for command entry again. If the operator tries to access equipment which is not connected to the system, or has been turned off, the system prints an error code and crashes to BASIC. Such a crash can be recovered from by typing in "GO TO 1000". The system then starts execution from statement 1000 by responding with "ENTER COMMAND?". If the "CLEAR" statement has been executed while control is in BASIC, the storage program can only be restarted by reloading the program "DATACQ.BAS" from the system disk and typing in "RUN" to execute the program.

CHAPTER VI

SYSTEM CHECKOUT

Sec. 6.1 Introduction

This data acquisition system has been designed as a general system that can be interfaced to a great variety of pulsed power experiments. To check out both the hardware and software of the system, the output of a delay generator (Cordin model 437-B) has been used to simulate the waveforms of an actual experiment. These waveform data have been digitized, stored and retrieved for analysis. This chapter describes the setting up of the hardware and the procedure of testing the system. Besides the main storage programs the support programs and the analysis programs have been executed for illustration purposes.

Sec. 6.2 Hardware Setup and Test Procedure

A block diagram of the hardware configuration is shown in Fig. 6.1. The signal output from the delay generator is connected to the vertical plug-in of the 7912AD. A trigger signal is connected to both the "EXT TRIG" of the 7912AD and the "SCAN TRIGGER" of the CAMAC 32 channel DVM. When the system is triggered, the DVM records the parameters of the experiment while the 7912AD starts digitizing the transient waveform. In an actual experiment, a certain amount of delay must be added to the trigger signal to ensure that the readings on the DVM are those parameters of the experiment just before the shot.

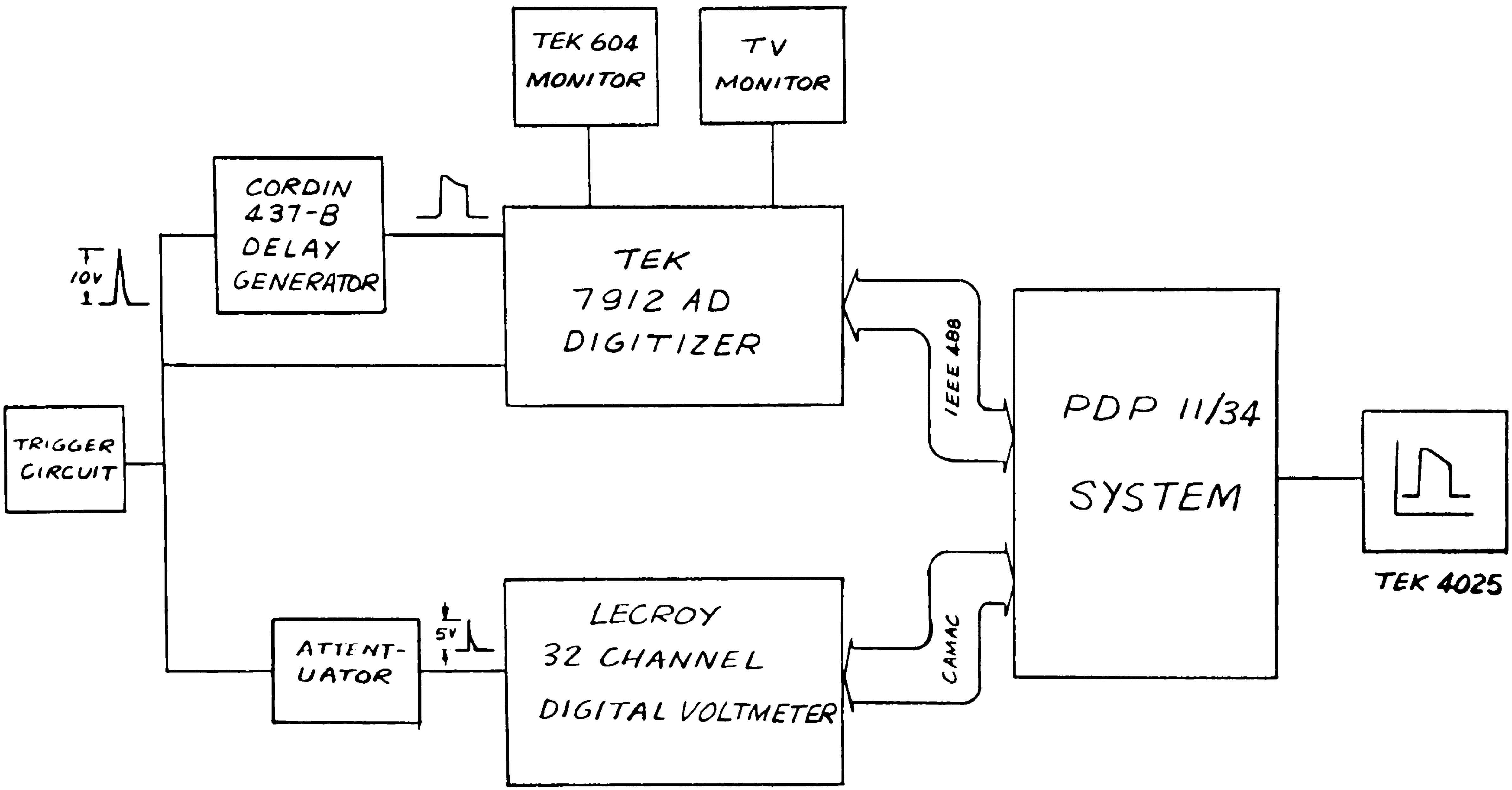


Fig. 6.1 A Block Diagram of the Hardware Setup for Testing

During the execution of the storage program, the computer will ask the operator to enter information and to make certain adjustments on the 7912AD digitizer. Before turning on the 7912AD, both the main and graticule intensity must be set at minimum. The settings on the horizontal plug-in should be: a) Trigger mode in "P-P auto", b) Coupling in "AC", and c) Source in "INT". The main intensity is increased slowly until a visible trace is seen on the TV monitor. A zero-reference trace can now be obtained by grounding the vertical plug-in. To digitize a transient waveform, the external trigger mode is used. When "EXT" is entered in response to the "TRIGGER MODE? INT/EXT?" prompt, the buttons "EXT" and "SINGLE SWP" of the horizontal plug-in must be pressed down. When the digitizer is triggered, the digitized waveform is displayed on the 604 monitor from the local memory of the 7912AD. This waveform can be examined and if it is valid data, it can be sent to the 11/34 by typing in "CONTINUE". If the waveform on 604 indicates that it is not satisfactory, the data stored in the local memory of the 7912AD can be erased by typing in "SCRATCH". The digitizer is then reset and the "SINGLE SWP" is armed.

Sec. 6.3 Execution of System Programs

The following is a sequence of printouts and curves showing the execution of the main storage program and the support programs. These programs are: a. DATACQ.BAS b. ANALYS.BAS c. FILTRI.BAS d. CURFIT.BAS e. CATDIR.BAS and f. CREATE.BAS. These printouts serve as demonstrations of and instructions on running these programs. Figure 6.2a shows the printout of the bootstraping routine, the GPIP driver 'GPI' and the DATACQ.BAS. Figure 6.2b shows the printout by the execution of DATACQ.BAS. Figure 6.3 shows the execution of the ACQUIRE command. The acquired data are displayed in the workspace. Figure 6.4 shows the execution of the GRAPH command; three data curves are plotted in the workspace. Figure 6.5 shows the execution of the STORE command. The contents of the parameter file are printed in the workspace for modification and are used for the assembly of the data header. Figure 6.6 is the execution of the STATUS command. Figures 6.7 and 6.8 show the execution of the programs CREATE.BAS and CATDIR.BAS. Figure 6.9 shows the execution of the PLOT command of ANALYS.BAS. Three data curves are plotted on the same coordinates so that the delay and jitter of each shot can be compared. The execution of the PEAK command is shown in Fig. 6.10 when the data curve is plotted with the maximum value of the curve printed next to it. The execution of INTEGRATE and DIFFERENTIATE commands are shown in Fig. 6.11 and Fig. 6.12. Figures 6.13 and 6.14 show the execution of CURFIT.BAS program. The execution of FILTRI.BAS is shown in Fig. 6.15. The frequency spectrum of the data is first plotted on the screen. Figure 6.16a shows the original signal waveform while Fig. 6.16b shows the low pass waveform with the upper cutoff frequency set at 500 MHz. Figure 6.17 shows two curves of the filtered signal with upper cutoff frequences set at 400 MHz and 200 MHz, respectively. Figure 6.18a and Fig. 6.18b show the band pass signal and the high pass signal. Figure 6.19 is the data plot by the <u>READ</u> command.

```
000204 000617 000037 000217
$DX
TEK SPS BASIC V02-01 — COPYRIGHT 1979 — TEKTRONIX, INC.
ALL RIGHTS RESERVED. SERIAL NO. 10248.

YOU HAVE 10886 WORDS OF FREE MEMORY.


READY
*SETDATE '1-SEP-80'

READY
*SETTIME '13:00:00'

READY
*SETTIME '13:30:00'

READY
*LOAD 'GPI.SPS'

READY
*OLD DX:'DATACQ.BAS'

READY
*RUN_
```

Fig. 6.2a Bootstraping and Loading of DATACQ.BAS

```
                         *****CAMAC SYSTEM*****
 UNIT # ,  BUS #  , CRATE # , SLOT #  ,MNF. TYP# POSTRIG ,SERIAL # ,
     1,      583,         2,       6,     2264,        1,    169436
     2,      583,         2,      16,     2264,        2,    169995
     3,      583,         2,      15,   2232.1,       00,    169994
                        *****TEKTRONIX SYSTEM*****
 UNIT # ,  BUS #  , INTF. # ,MNF.TYP# , L. ADDR., T. ADDR., S. ADDR.,
     1,      488,         0,   7912AD,     0+32,      0+64,      0+96,


 DEFAULT VALUES OF CRATE AND SLOT ARE 2 AND 15

 IF PROGRAM SHOULD STOP USE "GOTO 1000" TO REENTER PROGRAM

 ENTER PARAMETER FILE NAME?PARM1.LST
 ENTER OPERATOR'S NAME-?ALYEUNG
```

Fig. 6.2b Execution of DATACQ.BAS

```
ENTER COMMANDS
?ACQ

TO ACQUIRE ZERO-REFERENCE TRACE, GROUND VERTICAL PLUG-IN
ADJUST INTENSITY, POSITION TRACE AND PRESS ANY KEY

TRIGGER MODE?INT/EXT?EXT

TO ACQUIRE WAVEFORM TRACE,UNGROUND VERTICAL PLUG-IN
ESTABLISH WAVEFORM, ADJUST INTENSITY AND PRESS ANY KEY

TRIGGER DIGITIZER,ENTER "CONTINUE" TO ACQUIRE DATA
"SCRATCH" TO RESET DIGITZER
?CONTINUE
```

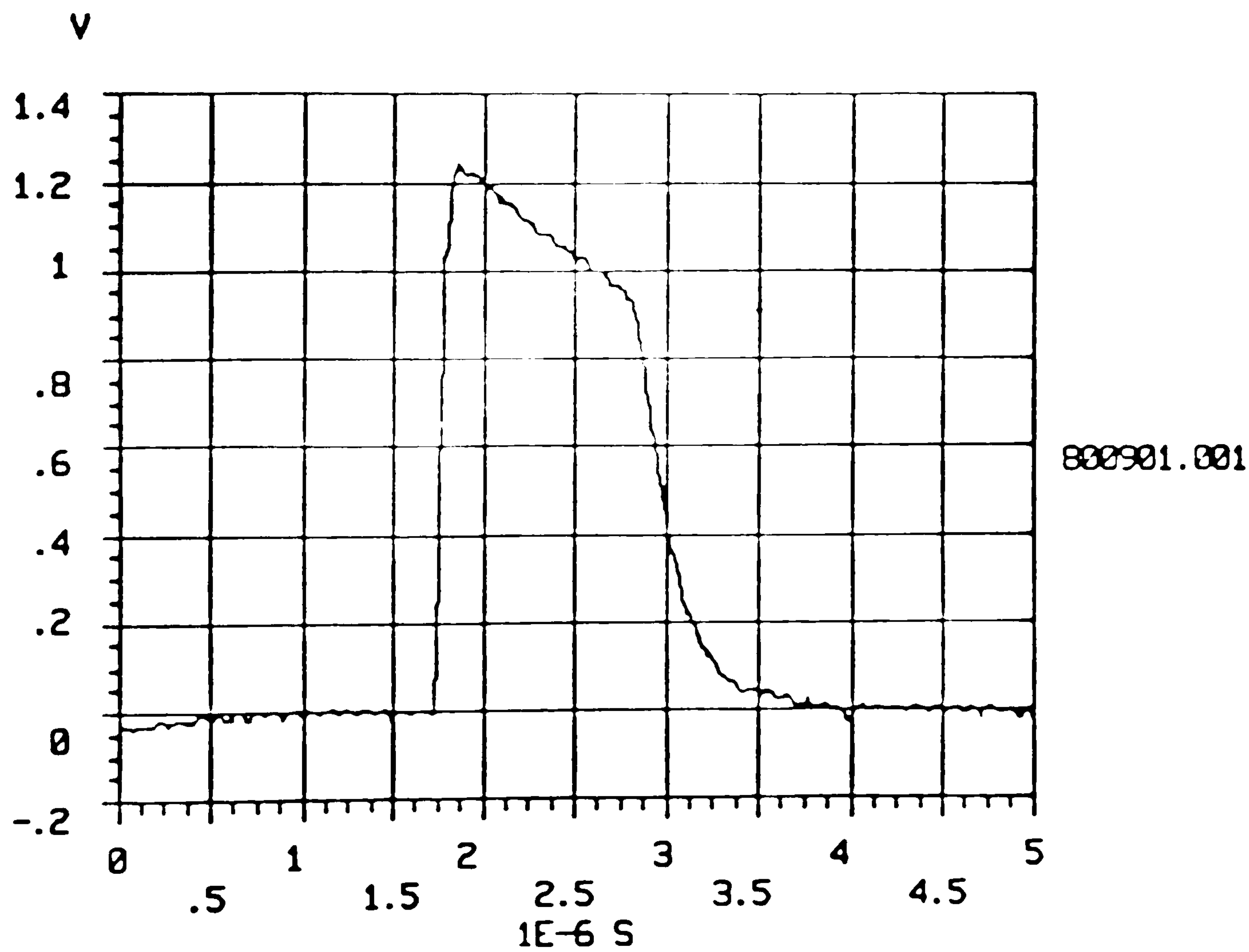


Fig. 6.3 Execution of ACQUIRE Command

```
ENTER COMMANDS
?GRA

DATA FILE?800901.003
TYPE STOP AFTER DATA FILE?
IF NO MORE GRAPH TO PLOT
DATA FILE?800901.001
TYPE STOP AFTER DATA FILE?
IF NO MORE GRAPH TO PLOT
DATA FILE?800901.002
```

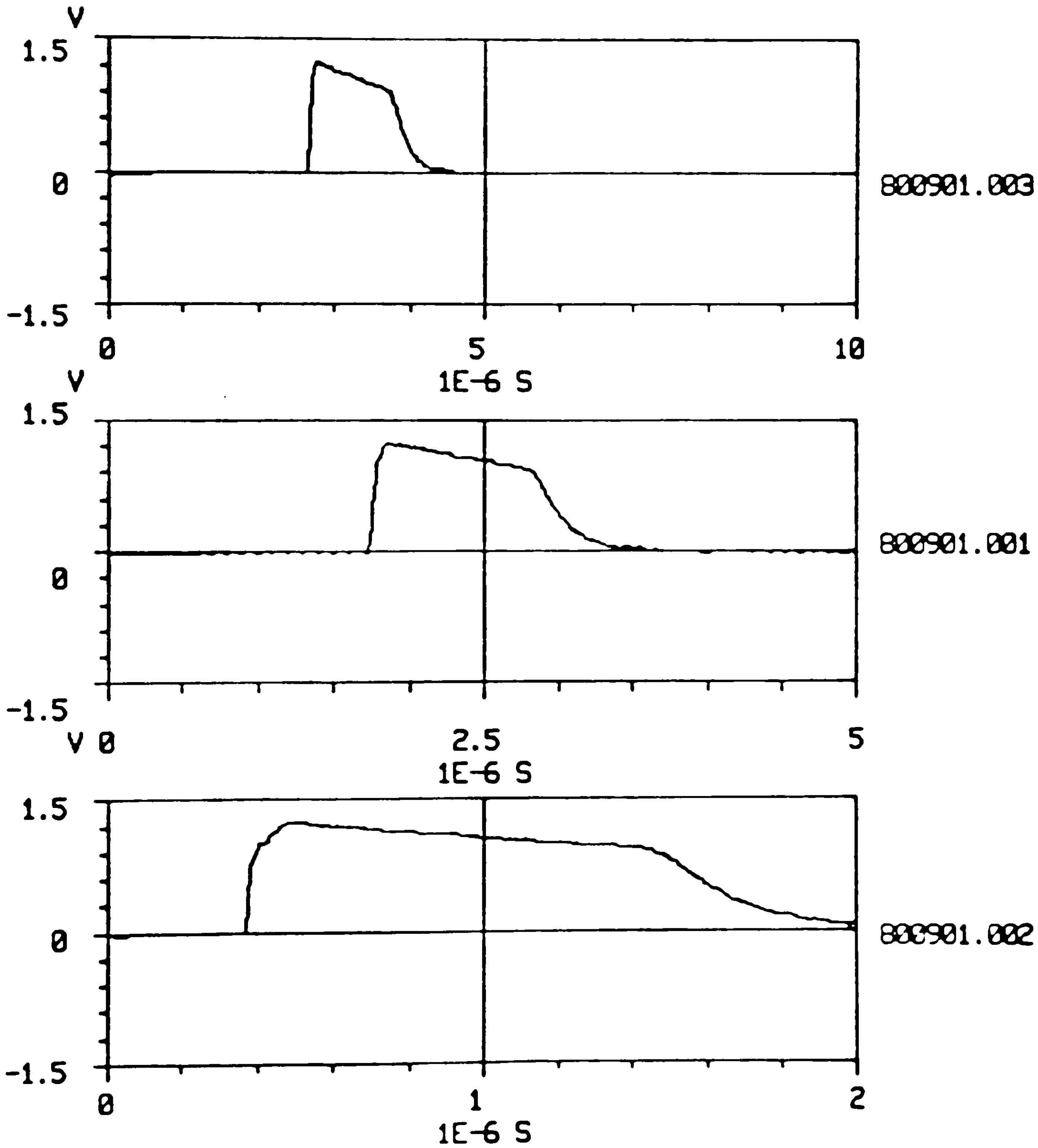


Fig. 6.4 Execution of GRAPH Command

```
ENTER COMMANDS
?STO_


1002  CHAMBER
1003    GAS ID#  PRESSURE(T)  GAS NAME  PRESENTAGE(%)
1004         01,         960,  NITROGEN,       50,
1005         02,         960,     ARGON,       50,
1006
1007  MAX BANK
1008    CHAR.VOLT.(KV)  TIME CONST.(NS)  TRIG.TIME
1009        250,               250,          --MS,
1010
1011  GAP VOLTAGE(KV) BEFORE LASER TRIGGERED
1012        200,
1013
1014  LASER
1015    TYPE    POWER(MW)  PULSE WITH(NS)  OPTICAL FIBER
1016    RUBY,       2,              15,          1MM
1017
1018  CABLE DELAYS(NS)
1019    POC.CEL  dI/dt  FAST TRIG.  LASER P. V. RAM
MAKE CHANGES NOW THEN ANSWER (Y/N)
SAVE NEW FILE?N
ENTER PARAMETER FILE'S NAME?PARM1.LST
800901.001 STORED ON DX1:
```

Fig. 6.5 Execution of STORE Command

```
ENTER COMMANDS
?STA
SERVICE REQUEST
POWER UP

ENTER COMMANDS
?
```

Fig. 6.6 Execution of STATUS Command

```
OLD DX:'CREATE.BAS'

READY
*RUN
THIS PROGRAM WILL CLEAR THE DIRECTORY AND CONTENTS OF DX1:
ARE YOU SURE THAT YOU WANT TO INITIALIZE DX1: (Y/N)?Y
ENTER DISC #(JUST THE NUMBER)?1
PLEASE WAIT 25 SECONDS
ENTER COMMENTS FOR DAT001 ENDING WITH "END" AS THE FIRST THREE
CHARACTERS OF A NEW LINE,THERE ARE 5 LINES FOR COMMENTS
?DATA DISK #1 FOR TEK 7912 DIGITIER
?END
INITIALIZATION OF DAT001 IS COMPLETE
ENTER -1 FOR NEXT DISC # TO EXIT PROGRAM
ENTER DISC #(JUST THE NUMBER)?-1
FINISHED

STOP  AT LINE   310
READY
*_
```

Fig. 6.7 Execution of CREATE.BAS

```
OLD DX:'CATDIR.BAS'

READY
*RUN
** WARMING-THIS DISK MAY HAVE BEEN CATALOGUED BEFORE
 THE FIRST LINE OF DX1:CATALO.G IS--
EDITEK.BAS      2 12-AUG-80
DO YOU STILL WANT TO CATALOGUE IT (Y/N)?Y
THIS DISK HAS BEEN CATALOGUED
DO YOU WANT TO HAVE THE PRINT OUT OF THE CATALOG (Y/N)?N

STOP AT LINE    90
READY
*
```

Fig. 6.8 Execution of CATDIR.BAS

```
OLD DX:'ANALYS.BAS'

READY
*RUN
TYPE 'STOP' TO HALT PROGRAM,'PLOT' TO PLOT GRAPHS
DATA FILES?PLOT

NO OF GRAPH?3

DATA FILES
?800901.001
?800901.002
?800901.003
```

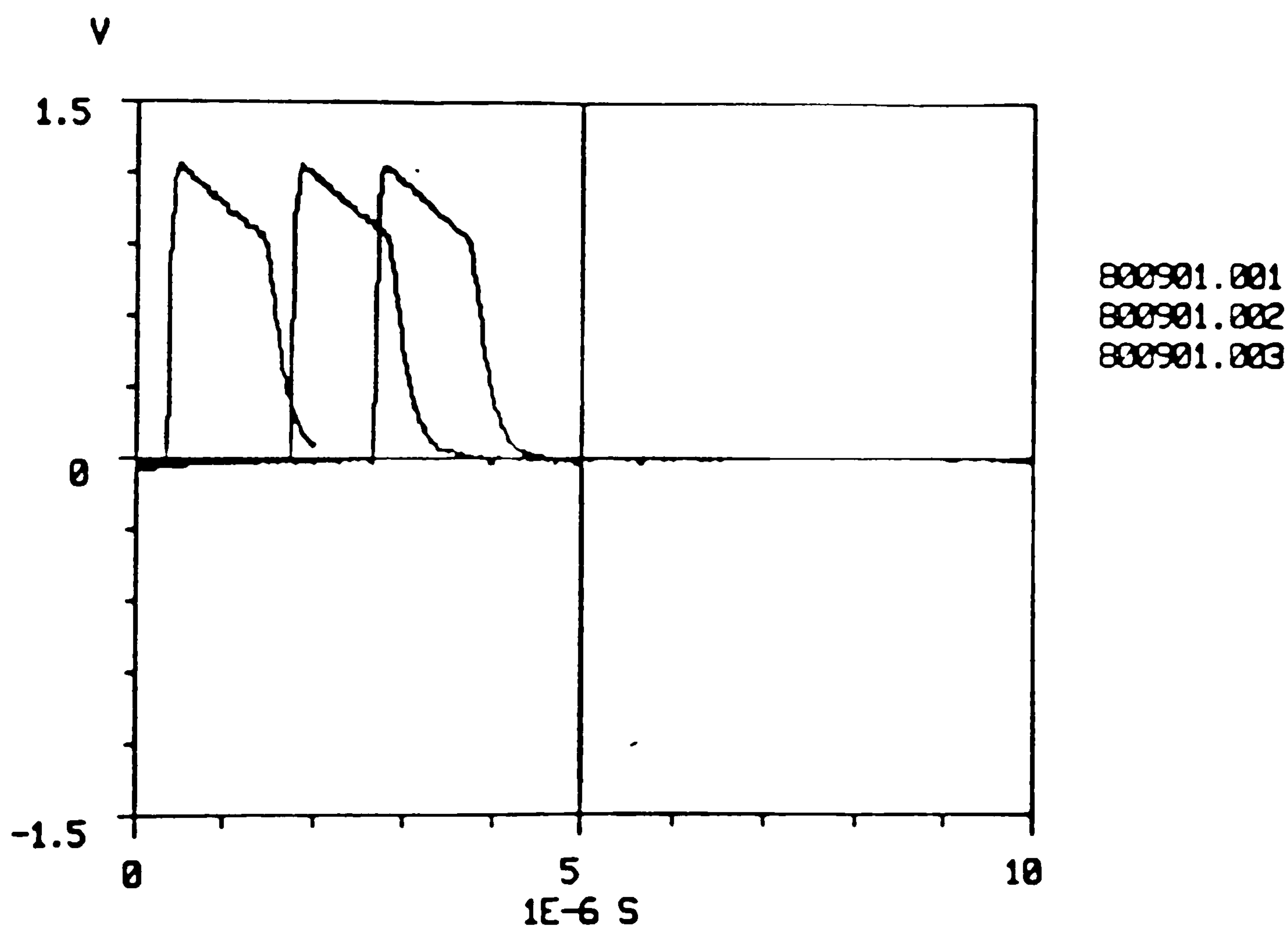


Fig. 6.9 Execution of ANALYS.BAS and PLOT Command

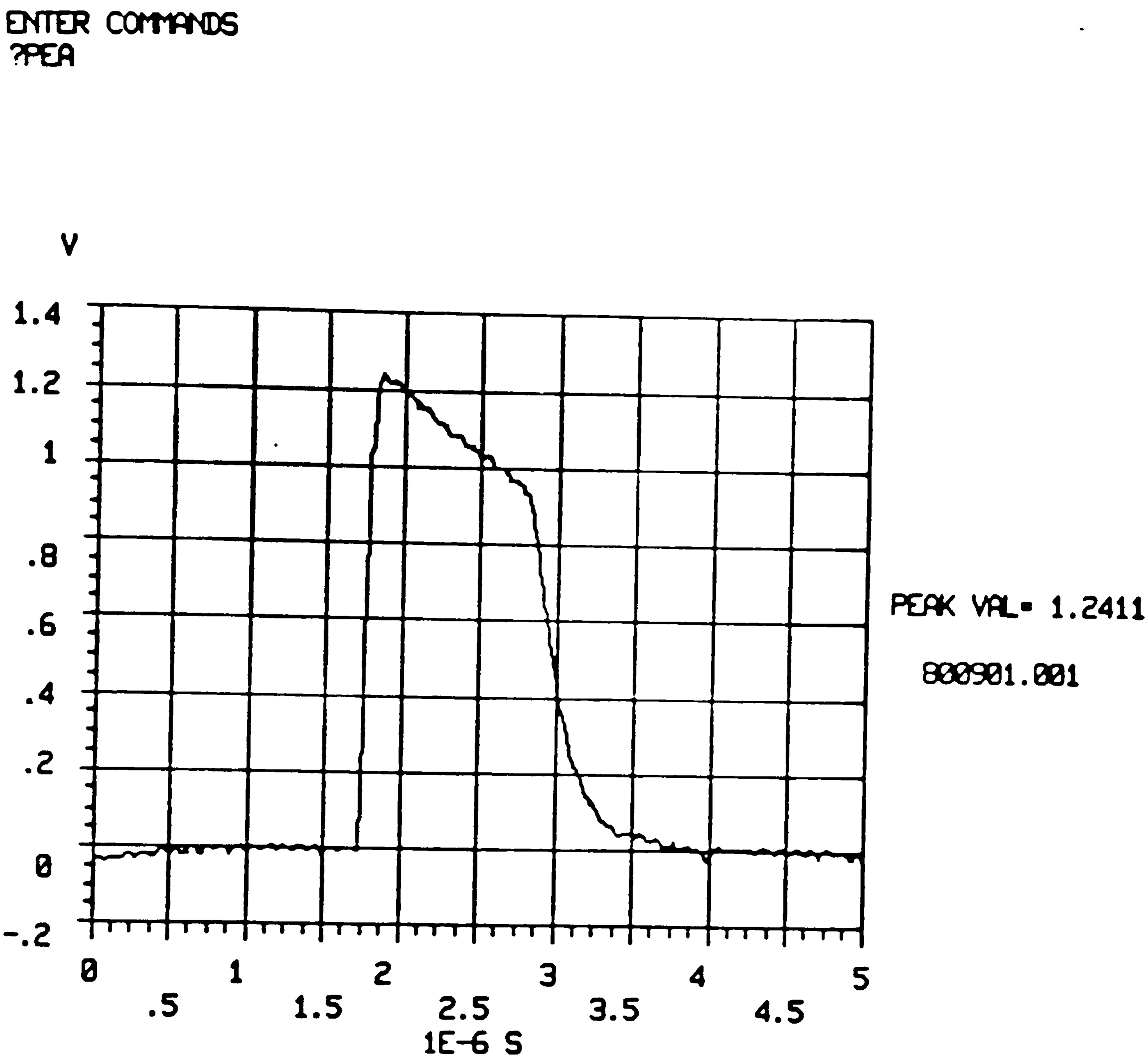


Fig. 6.10 Execution of PEAK Command

```
ENTER COMANDS
?INT
1ST OR 2ND INTEGRAL?1 OR 2?1
TYPE 'STOP' TO HALT PROGRAM,'PLOT' TO PLOT GRAPHS
DATA FILES?_
```

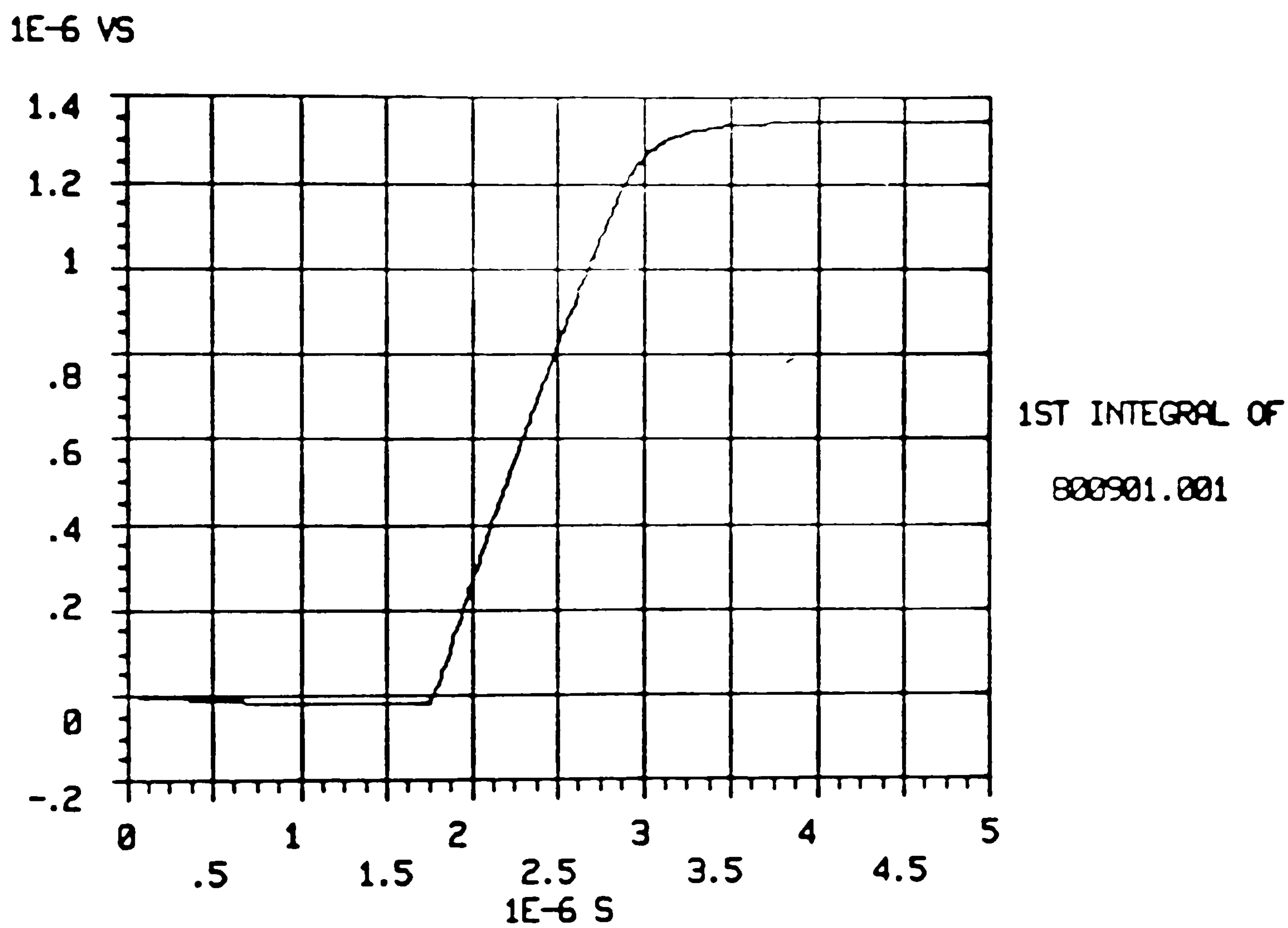


Fig. 6.11 Execution of INTEGRATE Command

```
ENTER COMANDS
?DIF
1ST OR 2ND DERIVATIVE(1 OR 2)?1
TYPE 'STOP' TO HALT PROGRAM,'PLOT' TO PLOT GRAPHS
DATA FILES?STOP

STOP  AT LINE   505
READY
*_
```

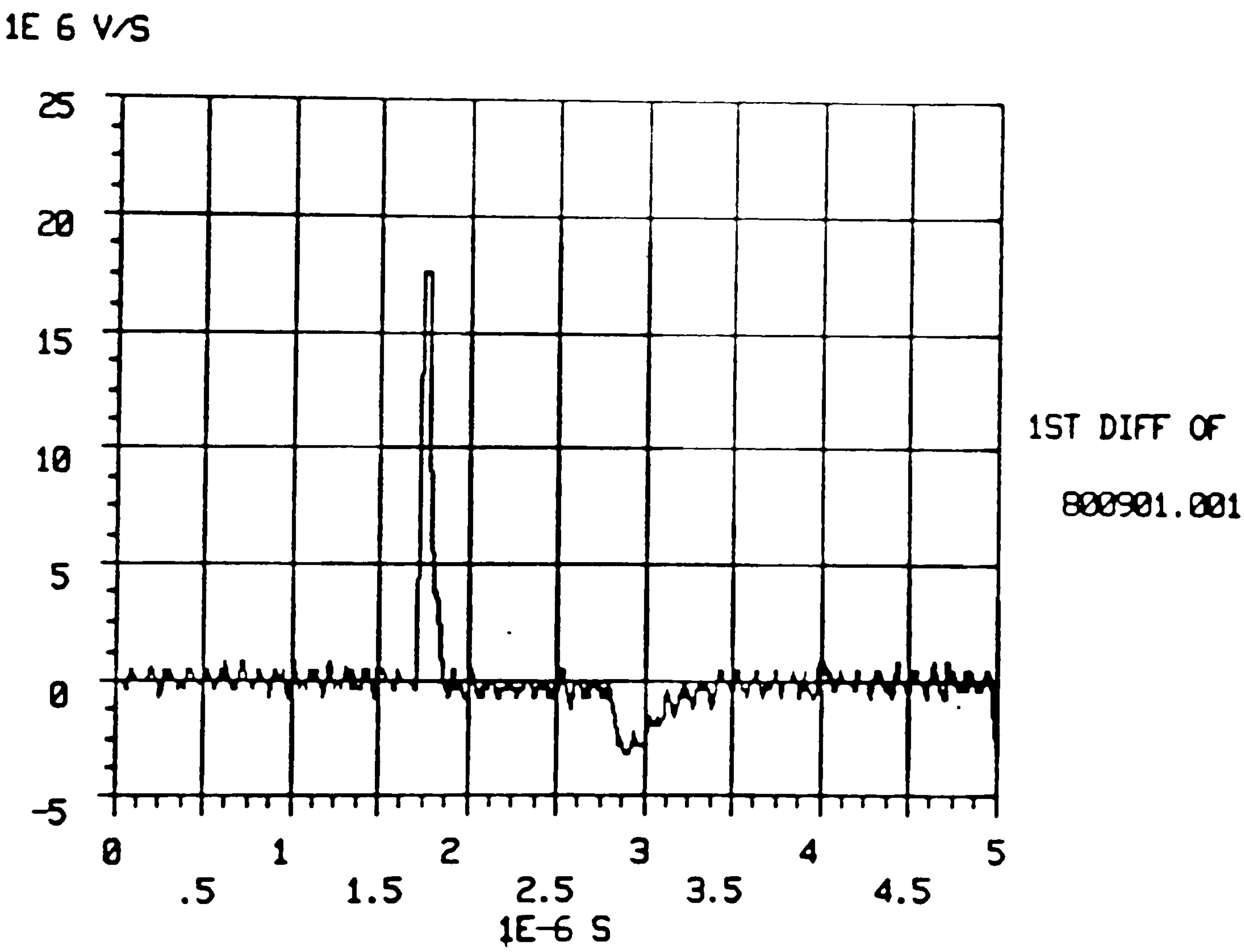
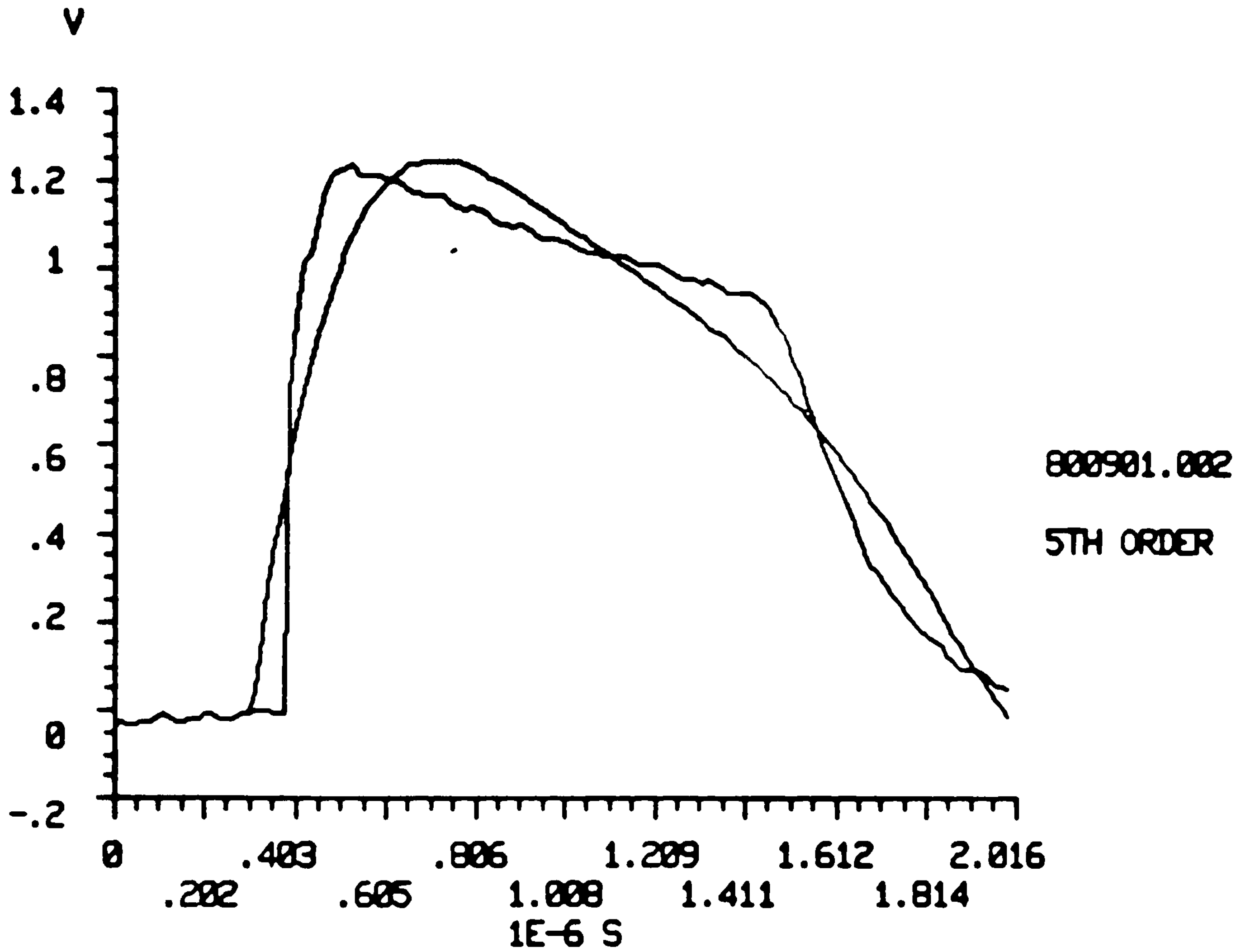


Fig. 6.12 Execution of DIFFERENTIATE Command

```
OLD DX:'CURFIT.BAS'

READY
*RUN
POLYNOMIAL CURVEFITTING ROUTINE,IT TAKES 129 DATA POINTS
              THE HIGHEST ORDER IS 16
DATA FILE?800901.002
ENTER ORDER, -1 STOPS PROGRAM?5_
```




```
C( 0)= .0487633
C( 1)= 1.27103
C( 2)=-.480637
C( 3)= .0809813
C( 4)=-6.53684E-03
C( 5)= 1.98681E-04
ENTER ORDER, -1 STOPS PROGRAM?_
```

Fig. 6.13 Execution of CURFIT.BAS

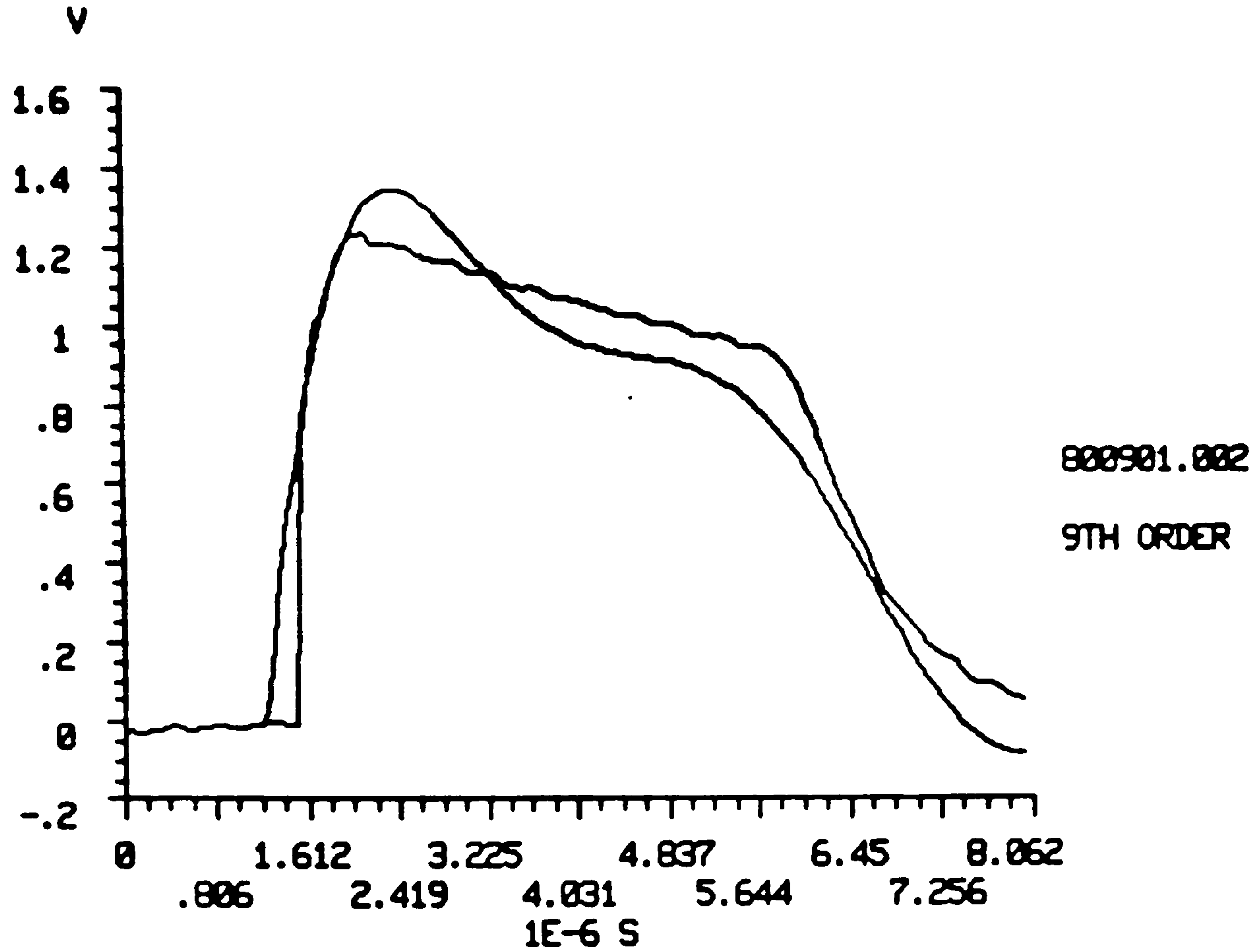


Fig. 6.14a 9th Order Curvefitting

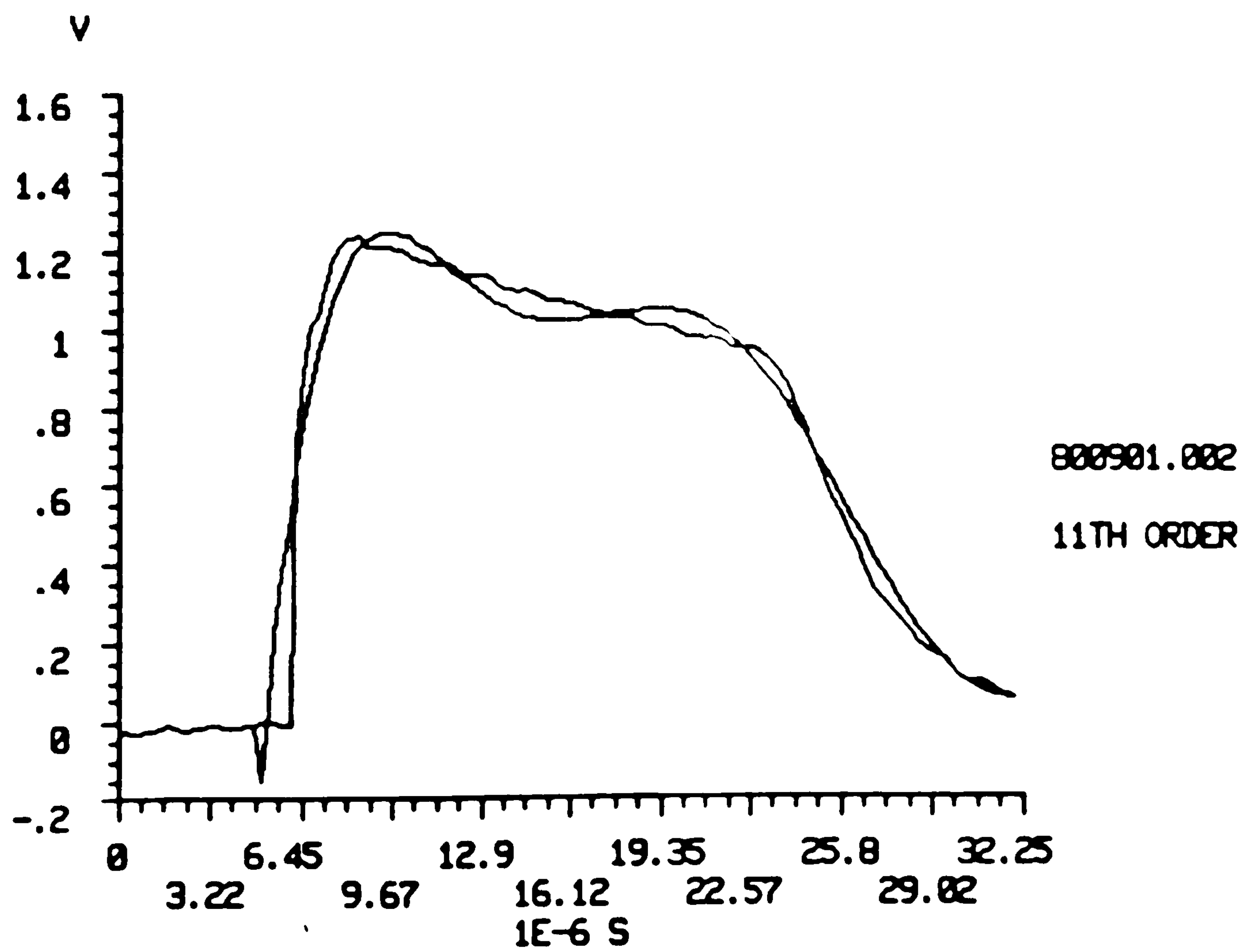


Fig. 6.14b 11th Order Curvefitting

```
OLD DX:'FILTR1.BAS'

READY
*RUN
DATA FILE?801118.001_

ENTER UPPER CUTOFF FREQ IN KHZ,-1 FOR HIGH PASS
?500000
ENTER LOWER CUTOFF FREQ IN KHZ,-1 FOR LOW PASS
?-1_
```

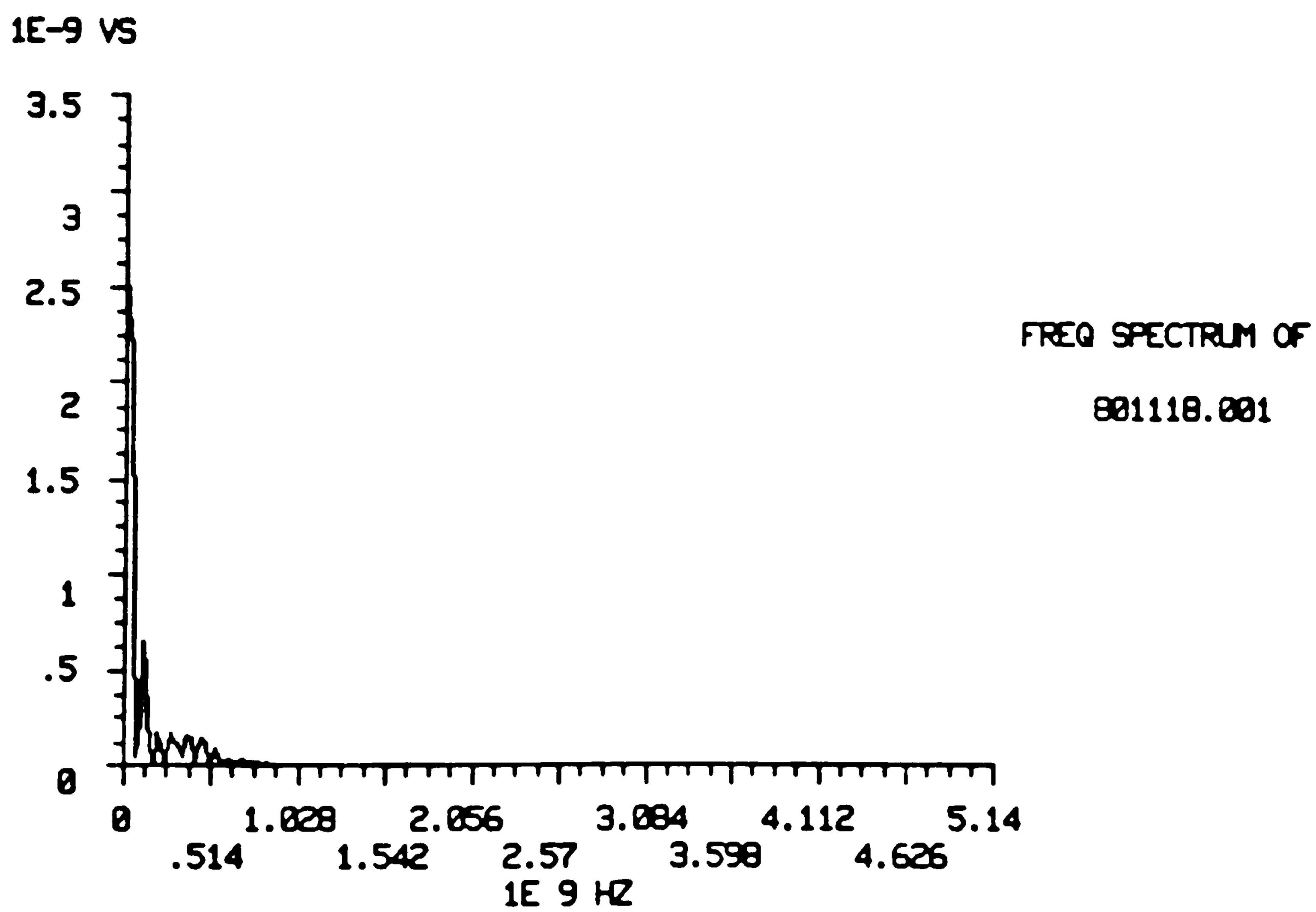


Fig. 6.15 Execution of FILTR1.BAS

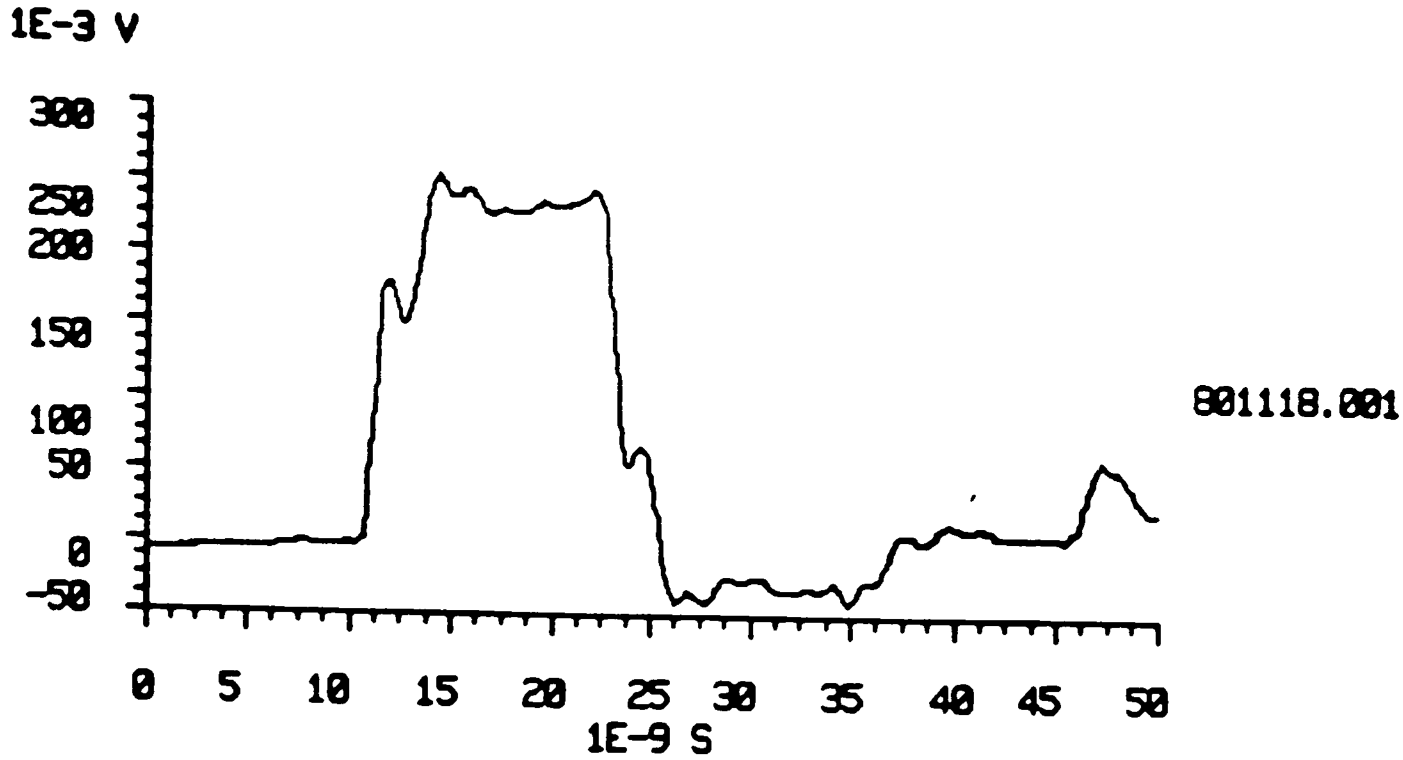


Fig. 6.16a Original Signal of Data Record 801118.001

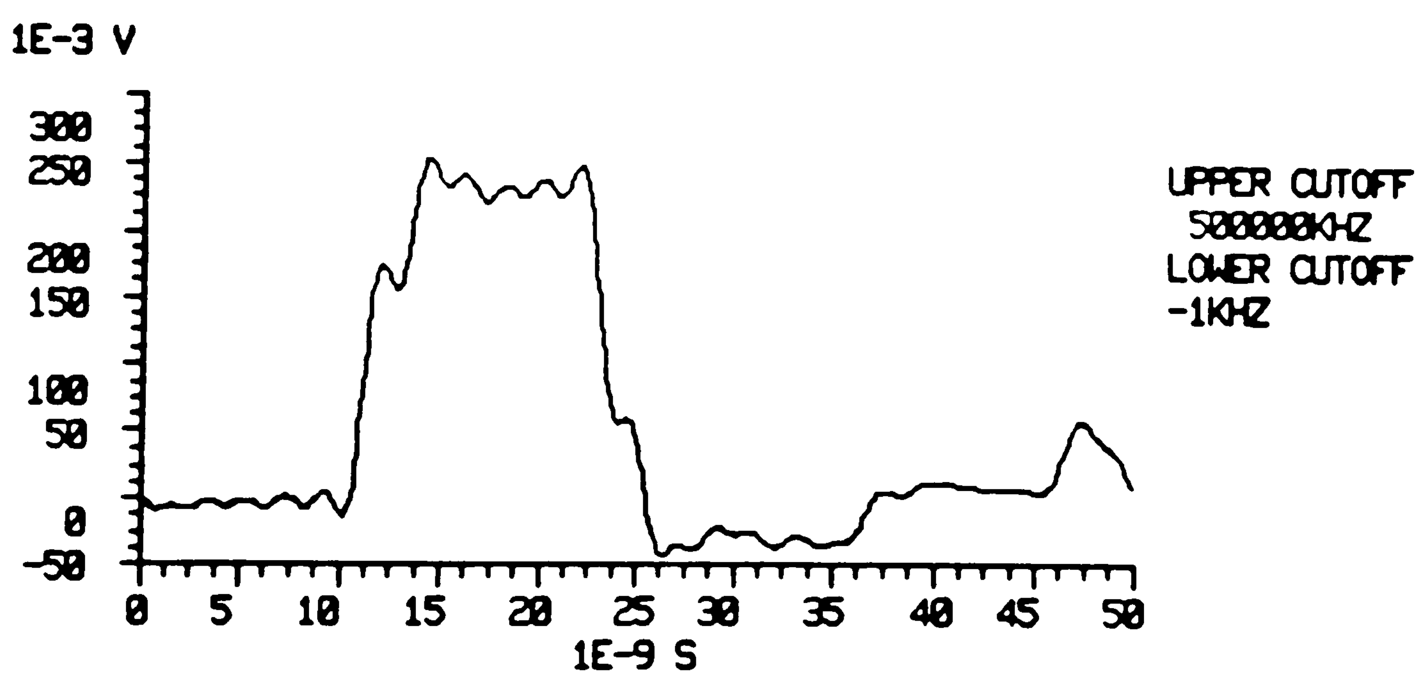
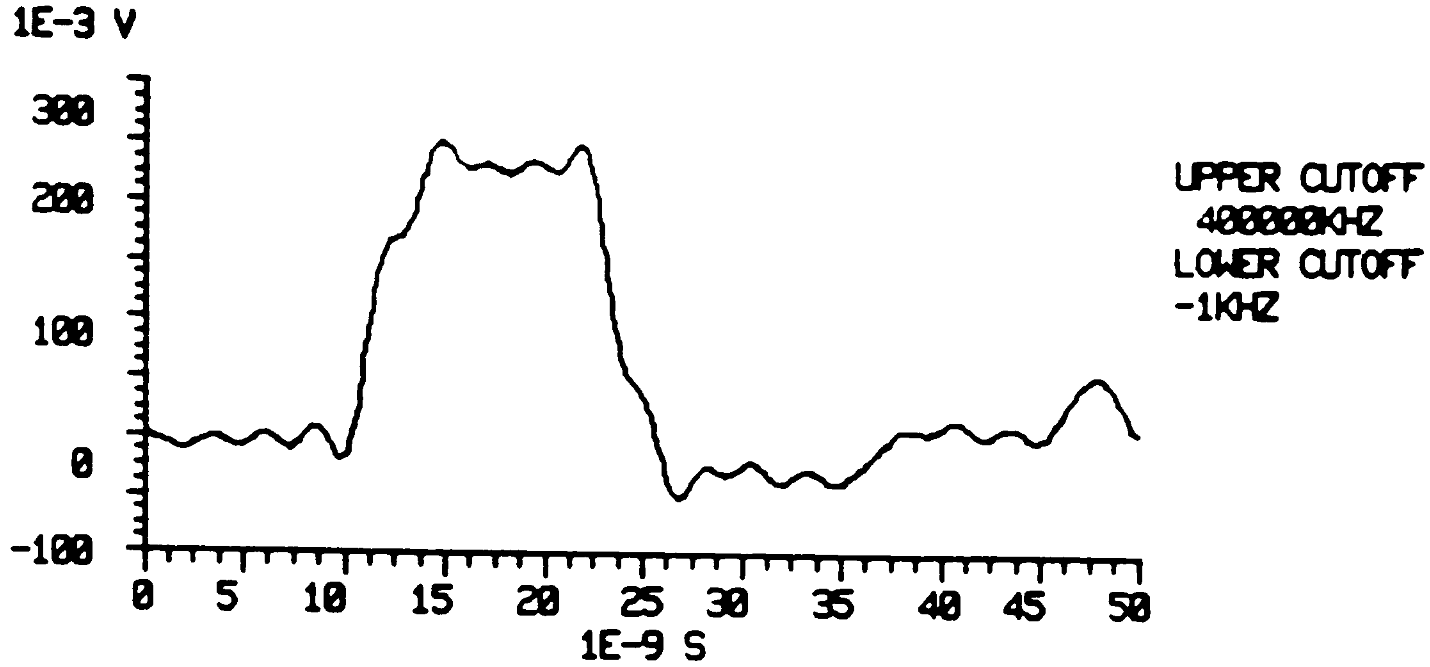


Fig. 6.16b Filtered Signal of 801118.001



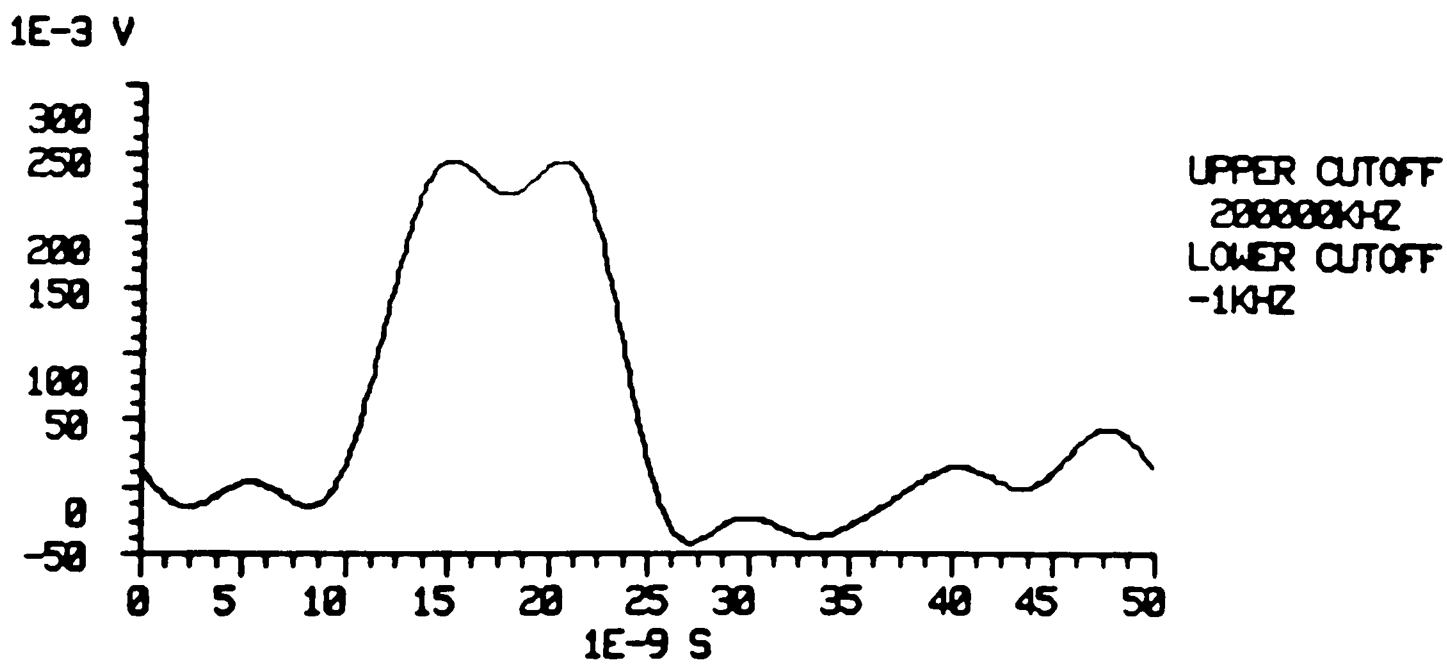


Fig. 6.17 Filtered Signal of 801113.C01

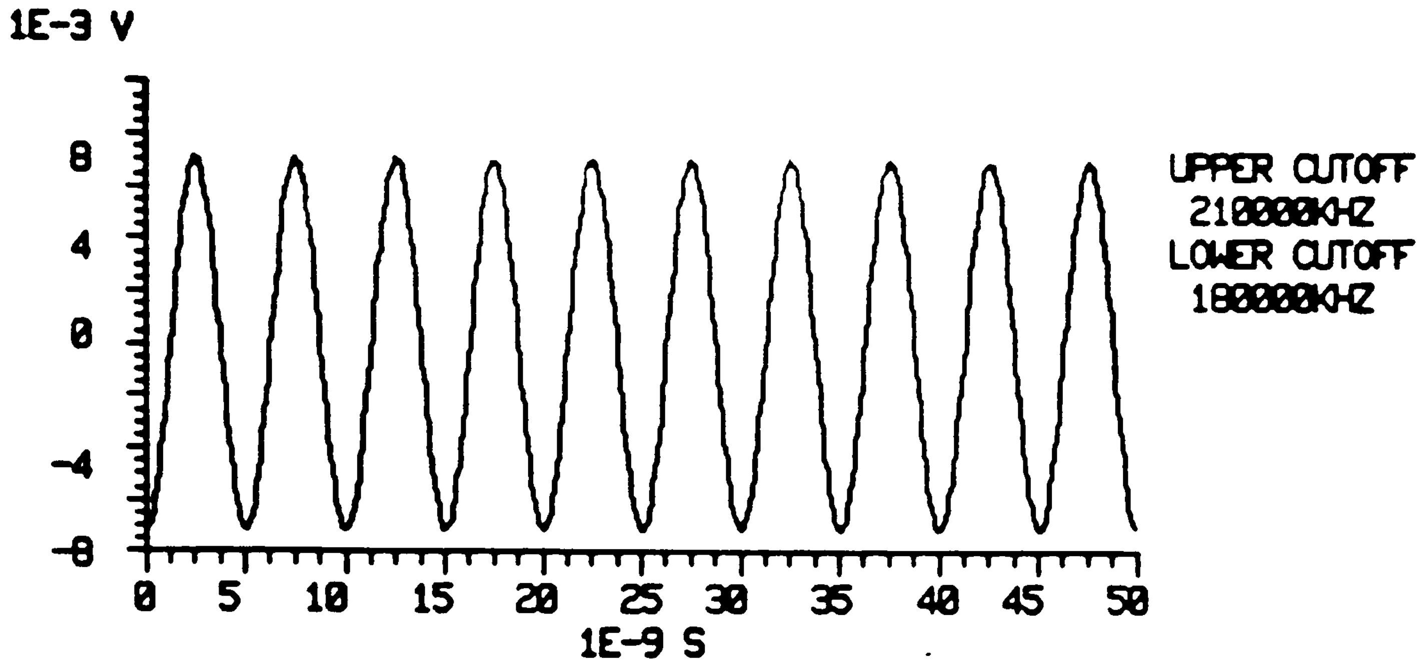


Fig. 6.18a Band Pass Signal of 801118.001

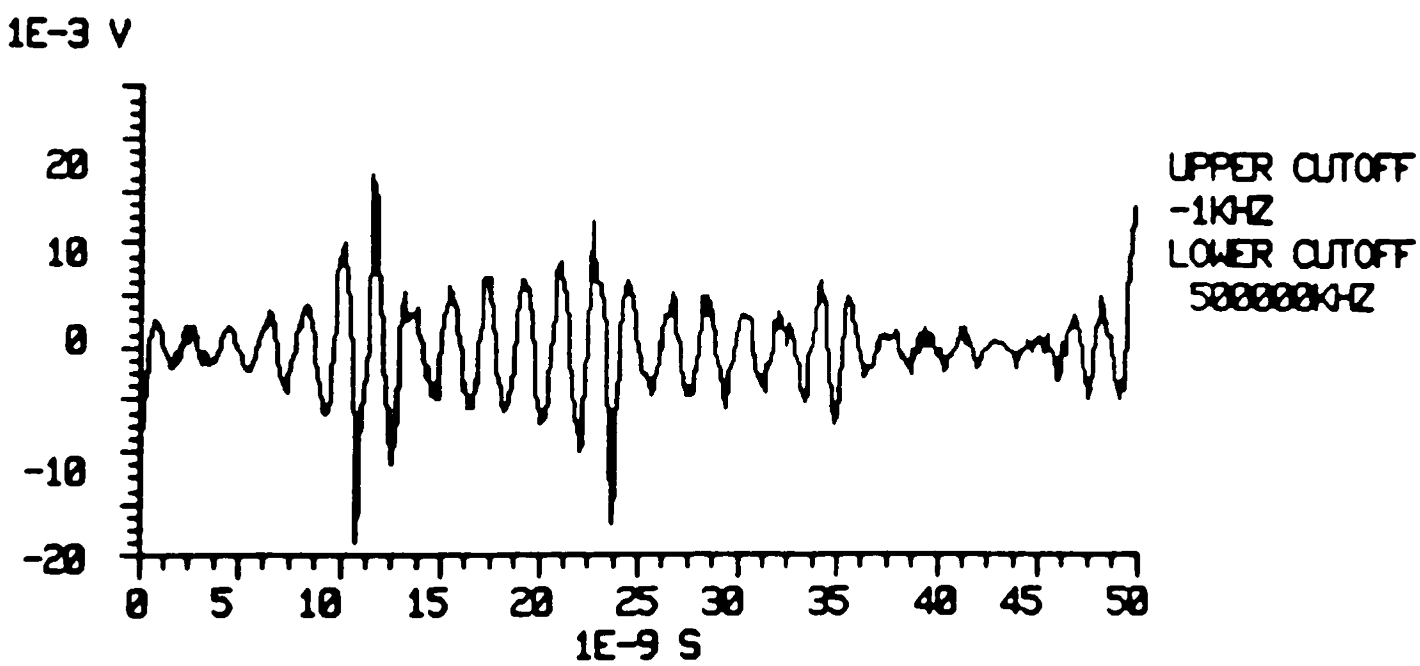


Fig. 6.18b High Pass Signal of 801118.001

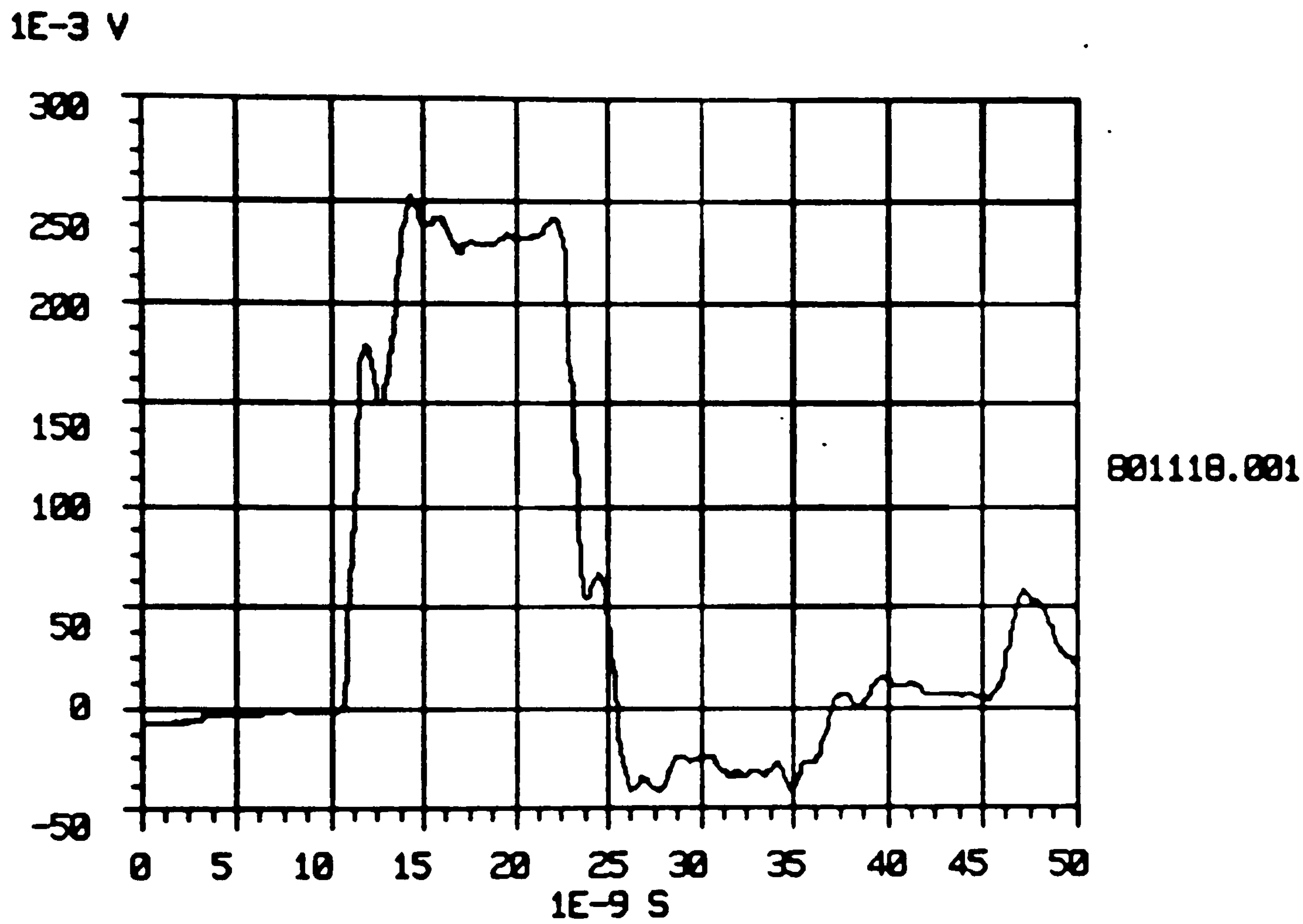


Fig. 6.19 Data Record Plotted by the READ Command

CHAPTER VII

CONCLUSION

A general purpose minicomputer based data acquisition and retrieval system has been designed for fast pulsed power experiments. The system includes a PDP - 11/34 minicomputer, a Tektronix 7912AD digitizer, a CAMAC 32 channel DVM, an HP 12050A optical data link set, a Tektronix graphics terminal, and BASIC programs to control the system. Three BASIC data analysis programs have been written which allow stored data to be studied in great detail. Data can also be filtered digitally so that meaningful information can be extracted from apparently bad data.

The versatility and flexibility of this system come from the fact that the 11/34 computer is interfaced to two digital communication and control systems; the IEEE 488 interface bus system and the CAMAC system. New instruments designed and built according to these two systems' standard can be connected to the computer and operated with the appropriate program. The utilization of the scan coverter and microprocessor control by the 7912AD allows fast transient signals up to 1 GHz bandwidth to be recorded digitally. This greatly enchances the diagnostic capability of this system. Through the HP 10250A optical fiber data link, the distance between the computer unit and the 7912AD can be extend up to 100 meters. Thus the computer based data acquisition system becomes available to different experiments located at various locations other than the computer site.

The 7912AD digitizes a waveform and stores it with 512 data points. For practical purposes, half of the number of data points (256) will provide a reasonably good resolution of the signal. If a dual trace plug-in is installed, a minor modification can be made that enables the horizontal sweep to switch from channel A to channel B. The channel A signal is stored by data points 0 to 255 while the channel B signal is stored by points 256 to 511.

At present, the system software is the TEK SPS BASIC V02. This software supports only 32 k-words of memory. In order to use the whole 48 k-words of available memory, the SPS BASIC extended memory system (SPS BASIC V02 XM) must be used. If the TEK SPS BASIC assembly level support package is included in the system software, Marco-11 assembly routines which control the CAMAC 8 channel digitizer can be linked to the SPS BASIC so that the system can utilize all of the existing hardware.

The present system can easily be expanded by adding on new equipment such as an additional 7912AD digitizer or CAMAC crate. Since the GPIB supports up to 15 devices, additional 7912AD's can be connected to the system through a GPIB cable. New addresses designated to the 7912AD must be added to the address table in the BASIC program "DATACQ.BAS" so that the software can access the new device. A new crates can also be added to the system through an IEEE 488 interface module that interfaces the GPIB bus to the CAMAC Dataway. With the present system setup, both the IEEE 488 standard and CAMAC standard instruments can "plug in" to the system with no hardware modification. The only requirement is writing software control routines for the new instrument.

REFERENCES

(1) IEEE standard 488-1978 (Revision of ANSI/IEEE standard 488-1975), IEEE Standard Digital Interface for Programmable Instrumentation, November 30, 1978.

(2) Institute of Electrical and Electronics Engineers CAMAC Instrumentation and Interface Standards (John Wiley & Sons Inc., New York, 1976).

(3) System User's Manual, PDP 11/34 (Digital Equipment Corp., Maynard, MA, 1977), Ch. 1, Sec. 1.2.7.

(4) PDP-11 Peripherals Handbook (Digital Equipment Corp., Maynard, MA, 1976), Ch. 5.

(5) S.R. Beckerich, "A Computer Based Data Acquisition System for the Texas Tech Tokamak," M.S. Thesis, Dept. of Electrical Engineering, Texas Tech University (1980), Ch's. 2, 3.

(6) PDP-11 Software Handbook (Digital Equipment Corp., Maynard, MA, 1978), Ch. 15.

(7) Tek SPS Basic V02/V02XM System Software Manual (Tektronix Inc., Beaverton, OR, 1980)

(8) Tektronix 4025 Computer Display Terminal Operator's Manual, (Tektronix Inc., Beaverton, OR, 1978), pp.3-18 to 3-21.

APPENDIX A

PROGRAM LISTINGS AND DISCUSSIONS

A listing of each program written in BASIC is included in this section. A line by line discussion is also included.

DATACQ.BAS

DATACQ.BAS is the main program that contains the data acquisition routine and most of the system commands. It also contains the logic to overlay other basic program segments. Lines 20000 to 20260 of this program are reserved for overlaying new program segments.

LINES	PURPOSE
10	Set C$=! as the command character (see the 4025 User's Manual); set B$= "Bell"; set constant values.
15 to 75	Define keys F1, F6, F7 and F8 to special functions as discussed in Sec. 5.4.
110 to 250	Check system date and time; set E1=1 if date or time not set.
240 to 290	Convert the data from "DD-MMM-YY" to the "YYMMDD" format, e.g. "10-SEP-80" to "800910"; store in variable E3$
300 to 340	Check INITIA.LIZ file on data disk to determine if this is a new start or a restart; if it is a restart, ask operator to verify shot number and update INITIA.LIZ file.
500 to 520	Print hardware configuration file into workspace on 4025 terminal.

800 to 820	Set default CSR and crate addresses.
900 to 920	Print rerun message, accept parameter file name and operator's name entries.
930 to 945	Set constant values, define array and waveform dimension.
950	Set interface time out value to 3000 milliseconds; clear interface and 7912AD.
950-965	Set the listener's, talker's and secondary addresses of the 7912AD.
966	Erase lines 10 to 290 and lines 500 to 945.
970	Set up an error handling routine.
1000	Print "ENTER COMMANDS" prompt.
1010 to 1130	Address table for command subroutine.
2032 to 2400	Check if arrays and waveforms have been defined; if not define them, execute "DEFECT" commands.
2405 to 2406	Check if position of the zero reference trace has been moved.
2410 to 2440	Put the 7912AD to local and TV mode.
2450 to 2460	Wait for keyboard entry and acquire raw data.
2470 to 2540	Take the average of the upper and

	lower edge data and store it as the zero - reference trace data.
2542 to 2580	Set trigger mode of 7912AD and wait for keyboard entry.
2600 to 2730	Obtain the signal waveform; take the average to center data from upper and lower edge; normalize the waveform trace and plot the waveform on the workspace of the 4025.
2740	Execute subroutine which reads the CAMAC 32 channel DVM.
3030 to 3055	Determine if the waveform is continuous or transient; set 7912 accordingly and digitize the waveform signal.
3056 to 3058	Reset the 7912AD if "SCR" is entered; continue execution if "CON" is entered.
3060 to 3120	Invoke a subroutine to actually acquire the array data from the 7912AD.
3130 to 3270	Obtain the horizontal scale factor, vertical scale factor, horizontal and vertical unit from the 7912AD.
4000 to 4200	Subroutine to obtain data array from the 7912AD.

5000 to 5080	Subroutine to reject target defects
6000 to 6005	Execution of "STORE" command
6010 to 6015	Execution of "STATUS" command
6020 to 6025	Execution of "READ" command
6030 to 6035	Execution of "GRAPH" command
6500 to 7540	Subroutine responds to "C", "A" and "V" commands; set new crate addresses; read all 32 channels of the CAMAC VDM or a specified channel of the VDM.
8000 to 8070	Execution of "CLEAR" command
9000 to 9070	Error messages
2000 to 20220	Program segment reserved for overlay.
30000 to 30210	Execution of "PARAMETER" command; allows operator to make changes on the parameter file and store the new file on the disk.

12-DEC-80 01:15 [OPTION I, LPRINT V5:BB]

DX1:DATACQ.BAS Page 1

```
10 C$=CHR(33)\B2$=CHR(7)\E1=0\S5=1\A5=0\A6=0\A7=0\F1=0
15 PRINT '!LEARN F1 Z!MON KZ13'
20 PRINT '!LEARN S1 Z!WOR KZ13'
25 PRINT '!LEARN 173 Z!MON 30 H KZ13'
30 PRINT CHR(33);'LEA F6 Z';CHR(33);'WOR 25 H K';CHR(33);'JUM Z13-'
35 PRINT 'ZENTER LINE NUMBERS';CHR(33);'MON K ; LIST Z'
40 PRINT CHR(33);'LEA F7 Z';CHR(33);'WOR K';CHR(33);'MON HZ-'
45 PRINT 'Z';CHR(33);'UP 2';CHR(33);'COM # Z 13 ZLAST LINE !BUF NZ-'
50 PRINT 'Z!EOL 13 !DEL 50 !FRO 10 Z 13 Z #COM !Z-'
55 PRINT 'Z';CHR(33);'DLI 2';CHR(33);'JUM';CHR(33);'DLI 1 Z 13'
60 PRINT CHR(33);'LEA F8 Z';CHR(33);'DEL 400';CHR(33);'MON H K Z-'
65 PRINT 'Z';CHR(33);'ERA H';CHR(33);'BUF Z 13-'
70 PRINT 'Z';CHR(33);'FRO 10 42 Z 13 Z';CHR(33);'SEN Z 13'
75 PRINT CHR(33);'MON H'
100 REM CHECK DATE
110 DATE D$\IF D$<>'   ' THEN 200\E1=1\PRINT B2$\PRINT 'SET DATE ****'
200 REM CHECK TIME
210 TIME T1$\IF VAL(SEG(T1$,1,2))>4 THEN 230
215 E1=1\PRINT B2$\PRINT 'SET TIME ****'
230 IF E1<>1 THEN 240\PRINT 'ERROR IN DATE OR TIME SETTING'\STOP
240 B3$='JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC'
250 B4$='010203040506070809101112'
260 FOR I=1 TO 36 STEP 3\B5$=SEG(B3$,I,I+2)
265 IF B5$=SEG(D$,4,6) THEN 270\NEXT I
270 IF I=1 THEN 280\I=I/3*2
280 B6$=SEG(B4$,I,I+1)\B7$=SEG(D$,1,2)
285 IF ASC(SEG(D$,1,1))=32 THEN B7$='0'&SEG(D$,2,2)
290 B8$=SEG(D$,8,9)&B6$&B7$\REM B8$ IS THE YYMMDD
300 OPEN #1 AS DX1:'INITIA.LIZ' FOR READ\READ #1,D1$,S1\CLOSE #1
315 IF D1$<>D$ THEN 330
320 PRINT 'LAST SHOT #=';S1\PRINT 'ENTER TRUE LAST SHOT #';\INPUT S5
330 CANCEL DX1:'INITIA.LIZ'\OPEN #1 AS DX1:'INITIA.LIZ' FOR WRITE INTO 1
340 WRITE #1,D$,S5\CLOSE #1
500 OPEN #1 AS DX0:'CONFIG.INF' FOR READ\PRINT C$;'WOR 20 H'
510 EOF #1 GOTO 520\READ #1,D2$\PRINT D2$\GOTO 510
520 CLOSE #1\PRINT C$;'MON H'
800 PRINT 'DEFAULT VALUES OF CRATE AND SLOT ARE 2 AND 15'
810 B1=61184\REM CSR ADDRESS
820 A1=60896\REM CRATE #2,SLOT #15 ADDRESS
900 PRINT\PRINT 'IF PROGRAM SHOULD STOP USE 'GOTO 1000' TO REENTER PROGRAM'
910 PRINT B2$\PRINT 'ENTER PARAMETER FILE NAME';\INPUT P1$
920 PRINT 'ENTER OPERATOR'S NAME-';\INPUT N1$
925 E=0\A8=0
930 INTEGER A(511),B(511)
935 WAVEFORM WA IS A,IA,HA$,VA$
940 WAVEFORM WB IS B,SP,HB$,VB$
945 WAVEFORM WC IS C(511),SC,HC$,VC$
950 SIFTO @0,3000\SIFLIN @0,'IFC'\SIFCOM @0,'DCL'
955 LA=0+32
960 TA-0+64
965 SA=0+96
966 DELETE 10,290\DELETE 500,945
970 WHEN @0 HAS 'ERR' AT 126 AS TASK 5 GOSUB 9000
1000 PRINT B2$\PRINT 'ENTER COMMANDS'
1010 INPUT C2$\C2$=TRM(C2$)\C1$=SEG(C2$,1,3)
1020 IF C1$='STO' THEN 6000\IF C1$='STA' THEN 6010\IF C1$='CLE' THEN 9050
```

```
1030 IF C1$='DEF' THEN 2000\IF C1$='NAM' THEN 8000
1040 IF C1$='ACQ' THEN 2000\IF C1$='PAR' THEN 29990
1050 IF C1$='REA' THEN 6020\IF C1$='GRA' THEN 6030
1100 C3$=SEG(C2$,1,1)
1110 IF C3$='C' THEN 6500\IF C3$='V' THEN 7000
1120 IF C3$='A' THEN 7500\IF C3$='H' THEN 6110
1130 GOTO 1000
2000 REM
2010 REM PROGRAM TO ACQUIRE A NORMALIZED 7912AD WAVEFORM
2020 REM          WITH CRT TARGET DEFECT REJECTION
2030 REM
2032 IF A8=0 THEN IF A7=0 THEN 2400\DELETE QQ,P,A,B,C,WA,WB,WC
2033 INTEGER A(511),B(511)
2034 WAVEFORM WA IS A,IA,HA$,VA$\WAVEFORM WB IS B,SB,HB$,VB$
2036 WAVEFORM WC IS C(511),SC,HC$,VC$
2400 IF C1$='DEF' THEN GOSUB 5000
2405 IF A6=0 THEN 2410\PRINT\PRINT 'HAVE YOU REPOSITIONED TARGET TRACE Y/N';
2406 INPUT G2$\IF G2$='N' THEN 2542
2410 PUT 'MODE TV' INTO @0,LA,SA
2415 PRINT B2$
2420 PRINT 'TO ACQUIRE ZERO-REFERENCE TRACE, GROUND VERTICAL PLUG-IN'
2430 PRINT 'ADJUST INTENSITY, POSITION TRACE AND PRESS ANY KEY'
2440 SIFCOM @0,LA,SA,'GTL'
2450 WAIT
2460 GOSUB 3000
2470 EDGEAD QQ,P,A,B
2540 ZREF A,B,ZR\A6=1
2542 PRINT\PRINT 'TRIGGER MODE?INT/EXT';\INPUT G3$\IF G3$='EXT' THEN F1=1
2545 PRINT B2$
2550 PRINT 'TO ACQUIRE WAVEFORM TRACE,UNGROUND VERTICAL PLUG-IN'
2560 PRINT 'ESTABLISH WAVEFORM, ADJUST INTENSITY AND PRESS ANY KEY'
2570 SIFCOM @0,LA,SA,'GTL'
2580 WAIT
2590 GOSUB 3000
2600 EDGEAD QQ,WP,WA,WB
2670 NORMAD WA,WB,WC,ZR,VS
2680 PRINT C$;'WOR 25'\PRINT C$;'GRA 1,35'\PRINT C$;'SHRINK'
2690 PAGE\VIEWPORT 200,800,200,600\SETGR VIEW
2710 DELETE QQ,P,A,B\A8=1
2717 PRINT C$;'WOR H'
2720 GRAPH WC
2730 PRINT CHR(33);'MON HK'
2740 GOSUB 7500
2830 GOTO 1000
3000 REM
3010 REM  PROGRAM RO ACQUIRE WAVEFORM RAW DATA
3030 INTEGER P(511)
3040 WAVEFORM WP IS P,SP,HP$,VP$
3045 IF F1=1 THEN 3055
3050 PUT 'DIG DAT' INTO @0,LA,SA\GOTO 3060
3055 PUT 'SSW ARM' INTO @0,LA,SA\PUT 'DIG SSW' INTO @0,LA,SA
3056 PRINT B2$\PRINT 'TRIGGER DIGITIZER,ENTER 'CONTINUE' TO ACQUIRE DATA'
3057 PRINT ''SCRATCH' TO RESET DIGITIZER'\INPUT AA$
3058 BB$=SEG(AA$,1,3)\IF BB$='SCR' THEN 3055\IF BB$='CON' THEN 3060\GOTO 3056
3060 PUT 'READ PTR,VER' INTO @0,LA,SA\F1=0
3070 GOSUB 4000
3080 P=QQ
3090 GOSUB 4000
```

```
3100 IF E=1 THEN 3130
3120 REJECT QQ,P,DF
3130 PUT 'MODE TV' INTO @0,LA,SA
3140 PUT 'HS1?' INTO @0,LA,SA
3150 GET A$ FROM @0,TA,SA
3160 SP=VAL(SEG(A$,5,LEN(A$)-1))
3170 SP=SP/51.2
3180 PUT 'VS1?' INTO @0,LA,SA
3190 GET A$ FROM @0,TA,SA
3200 VS=VAL(SEG(A$,5,LEN(A$)-1))
3210 PUT 'HU1?' INTO @0,LA,SA
3220 GET A$ FROM @0,TA,SA
3230 HF$=SEG(A$,5,LEN(A$)-1)
3240 PUT 'VU1?' INTO @0,LA,SA
3250 GET A$ FROM @0,TA,SA
3260 VF$=SEG(A$,5,LEN(A$)-1)
3270 RETURN
4000 REM
4010 REM  SUBROUTINE TO READ DATA ARRAY
4020 REM
4030 DELETE QQ
4040 IFDTM @0,'UNP'
4050 GET X FROM @0,TA,SA
4060 IF CHR(X)<>'%' THEN STOP
4070 IFDTM @0,'PAK','HBF'
4080 GET CW FROM @0,TA,SA
4090 IF CW>=2 THEN 4130
4100 GET Y FROM @0,TA,SA
4110 IF CW=1 THEN IF Y=-197 THEN PRINT 'NO DEFECTS ON TARGET'\E=1\RETURN
4120 PRINT 'ERROR IN TRANSMISSION OF TARGET DEFECT ARRAY'\STOP
4130 CW=(CW-1)/2-1
4140 INTEGER QQ(CW)
4150 GET QQ FROM @0,TA,SA
4160 IFDTM @0,'UNP'
4170 GET X FROM @0,TA,SA
4180 GET X FROM @0,TA,SA
4190 IF CHR(X)<>';' THEN STOP
4200 RETURN
5000 REM
5002 REM SUBROUTINE TO REJECT TARGET DEFECTS
5005 REM
5010 PRINT\PRINT 'DIGITIZE CRT TARGET DEFECTS 25 TIMES'
5020 PUT 'DIG DEF,25' INTO @0,LA,SA
5030 PUT 'READ DEF' INTO @0,LA,SA
5040 GOSUB 4000
5050 IF E<>1 THEN 5060\RETURN
5060 DELETE DF
5070 INTEGER DF(SIZ(QQ)-1)
5080 DF=QQ\RETURN
6000 IF F2=1 THEN 20000\DELETE A,B\OVERLAY DX0:'STORE1.BAS'
6005 F2=1\F3=0\F4=0\F5=0\GOTO 20000
6010 IF F3=1 THEN 20000\DELETE A,B,C\OVERLAY DX0:'STATS1.BAS'
6015 F3=1\F2=0\F4=0\F5=0\GOTO 20000
6020 IF F4=1 THEN 20000\DELETE A,B,C\OVERLAY DX0:'READ1.BAS'
6025 F4=1\F2=0\F3=0\F5=0\GOTO 20000
6030 IF F5=1 THEN 20000\DELETE A,B,C\OVERLAY DX0:'GRAPH1.BAS'
6035 F5=1\F2=0\F3=0\F4=0\GOTO 20000
6110 STOP
```

```
6500 REM ROUTINE TO READ CAMAC 32 CHANNEL ADC
6510 PRINT 'ENTER CRATE #';\INPUT C
6520 PRINT 'ENTER SLOT #';\INPUT S
6530 IF C=1 THEN A1=59392+S*32
6540 IF C=2 THEN A1=60416+S*32
6550 IF C>=3 THEN A1=49152+(C-3)*1024
6560 IF C>=10 THEN 6510
6570 IF S>23 THEN 6510\GOTO 1000
7000 PRINT 'ENTER CHANNEL #'\F=0\INPUT D
7010 IF D>32 THEN 7000\IF D>=17 THEN 7020\A1=A1+(D-1)*2\GOTO 7030
7020 A1=A1+(D-17)*2\F=1
7030 PUTLOC B1,F\GETLOC A1,V\V=V/4096*10-5
7040 PRINT 'CHANNEL #';D;'    V=';V;'  VOLTS'
7050 GOTO 1000
7500 DIM N(32)\F=0\FOR I=1 TO 32
7510 IF I>=17 THEN F=1\IF I=17 THEN A1=A1-32
7520 PUTLOC B1,F\GETLOC A1,M\N(I)=M/4096*10-5\A1=A1+2\NEXT I
7530 PRINT 'ALL 32 CHANNELS HAVE BEEN READ,VALUES STORED IN N(1)-N(32)'
7535 FOR K=1 TO 32\PRINT 'N';K;'=';N(K)\NEXT K
7540 IF C3$='A' THEN 1000\RETURN
8000 PRINT B2$\PRINT 'ENTER OPERATOR'S NAME:-';\INPUT N1$\GOTO 1000
8050 PRINT B2$\PRINT 'LISTENER ADDRESS';\INPUT C1
8060 PRINT 'SECONDARY ADDRESS';\INPUT C2
8070 SIFLIN @0,'IFC'\SIFCOM @0,C1,C2,'SDC'\GOTO 1000
9000 GIPES @0,E2
9010 GOTO E2 OF 9020,9030,9020,9030,9020,9030,9020
9020 PRINT 'DMA ERROR'
9030 GOTO E2 OF 9050,9040,9050,9050,9040,9040
9040 PRINT 'WRITE ERROR'
9050 GOTO E2 OF 9070,9070,9070,9060,9060,9060
9060 PRINT 'WRITE TIMING ERROR'
9070 RETURN
20000 REM
20010 REM
20020 REM
20030 REM
20040 REM
20050 REM
20060 REM
20070 REM
20080 REM
20090 REM
20100 REM
20120 REM
20130 REM
20140 REM
20150 REM
20160 REM
20170 REM
20180 REM
20190 REM
20200 REM
20210 REM
20220 REM
29990 PRINT 'PARAMETER FILR';\INPUT P1$
30000 PRINT C$;'WOR 20 H'
30005 ONERR AR GOTO 30190
30010 OPEN #1 AS DX0:P1$ FOR READ\L2=0
```

```
30020 READ #1,H$\PRINT H$\L2=L2+1
30030 EOF #1 GOTO 30050
30040 GOTO 30020
30050 CLOSE #1\PRINT C$;'MON H'
30060 PRINT C$;'FOR Y'\PRINT C$;'JUM'
30070 PRINT 'MAKE CHANGES NOW THEN ANSWER (Y/N)'
30080 PRINT 'SAVE NEW FILE';\INPUT H$\PRINT C$;'FOR N'
30090 IF SEG(H$,1,1)<>'Y' THEN 30140\PRINT 'NEW NAME';\INPUT P2$
30095 IF P1$<>P2$ THEN 30100\CANCEL DX0:P1$
30100 OPEN #1 AS DX0:P2$ FOR WRITE INTO 5
30110 PRINT C$;'PRO 10/?/'\PRINT C$;'BUF'\PRINT C$;'SEN'
30120 FOR I1=1 TO L2\INPUT H$\WRITE #1,H$\NEXT I1
30130 GOSUB 30150\CLOSE #1
30140 IF C1$='STO' THEN RETURN\GOTO 1000
30150 PRINT C$;'REP 00'\FOR I5=1 TO 150\INPUT H$
30160 IF SEG(H$,1,4)=CHR(33)&'ANS' THEN 30180\NEXT I5
30170 PRINT 'NO REPLY FROM 4025'\GOTO 1000
30180 PRINT C$;'BUF N'\PRINT C$;'PRO'\PRINT 'END WS'\A7=0\RETURN
30190 PRINT C$;'MON H'\CLOSE #1\IF AR(0)=0 THEN ONERR RETURN
30195 IF AR(0)<>30010 THEN 30210
30200 PRINT P1$;' DOES NOT EXIST'\GOTO 1000
30210 PRINT CHR(AR(1));AR(2);'ERROR IN LINE';AR(0)\GOTO 1000
```

STORE1.BAS

STORE1.BAS is overlaid on lines 20000 to 20260 when the "store" command is executed. It stores the waveform data on the data disk with the data header. The file name of the data file is the combination of the date and shot number.

LINES	PURPOSE
20000 to 20070	Obtain the time from system clock; assemble data file name in the form of YYMMDD.Sht#; print parameter file in workspace of 4025.
20080 to 20090	Assemble the first line of data header; check error; go to error handling routine if error occurs; ask for parameter file.
20110 to 20160	Assemble data file and store it on data disk; print message after finished.
20180 to 20260	Error handling routine.

```
12-DEC-80  01:11  [OPTION I, LPRINT V5:BB ]
                                                  DX1:STORE1.BAS Page 1

20000 TIME T1$\IF S5<9 THEN 20060\IF S5>99 THEN 20180\A7=0
20010 S4=S5/10\S6=ITP(S4)\IF S6=1 THEN S7=49\IF S6=2 THEN S7=50
20020 IF S6=3 THEN S7=51\IF S6=4 THEN S7=52\IF S6=5 THEN S7=53
20030 IF S6=6 THEN S7=54\IF S6=7 THEN S7=55\IF S6=8 THEN S7=56
20040 IF S6=9 THEN S7=57\S8=(S5-S6*10)+48
20050 B9$=CHR(48)&CHR(S7)&CHR(S8)\GOTO 20070
20060 B9$=CHR(48)&CHR(48)&CHR(48+S5)\REM B9$=SHOT #
20070 F$=B8$&'.'&B9$\GOSUB 30000
20080 A2$='1000 '&'DATE '&B8$&' , '&'TIME '&SEG(T1$,1,5)&' , '
20090 A3$='SHOT # '&B9$&' , '&'OPERATOR '&N1$\A4$=A2$&A3$
20100 ONERR AR GOTO 20190\PRINT 'ENTER PARAMETER FILE'S NAME';\INPUT P1$
20110 OPEN #1 AS DX0:P1$ FOR READ
20120 OPEN #2 AS DX1:F$ FOR WRITE\WRITE #2,A4$
20130 FOR J=1 TO 30\READ #1,A1$\WRITE #2,A1$\NEXT J
20140 WRITE #2,C,SC,HC$,VC$
20150 CLOSE #1\CLOSE #2
20160 PRINT F$;' STORED ON DX1:'\S5=S5+1
20170 ONERR\GOTO 1000
20180 PRINT B2$\PRINT\PRINT '# OF SHOOTS STORED EXCEEDED 99'
20190 CLOSE ALL\IF AR(0)=0 THEN ONERR RETURN
20200 IF AR(1)<>ASC('P') THEN 20230\IF AR(2)=11 THEN 20240
20210 IF AR(2)=16 THEN 20240
20220 IF AR(2)=5 THEN 20250\IF AR(2)=12 THEN 20260
20230 PRINT CHR(AR(1));AR(2);'ERROR IN LINE';AR(0)\GOTO 20170
20240 PRINT 'NOT ENOUGH ROOM ON DX1:'\GOTO 20170
20250 PRINT F$;' ALREADY EXISTS ON DATA DISK'\GOTO 20170
20260 PRINT F$;' ALREADY OPEN'\CLOSE ALL\GOTO 20170
```

GRAPH1.BAS

GRAPH1.BAS is overlaid on lines 20000 to 20260 when the "GRAP" command is executed. It reads the data file from the data disk and plots it on the workspace of the 4025 terminal. Up to three graphs can be plotted.

LINE	PURPOSE
20000 to 20010	Prepare workspace of 4025 for plotting graphs; delete old waveforms and array.
20010 to 20030	Accept input from operator.
20040 to 20050	Define waveform WD; initiate error checking and error handling routine.
20060 to 20070	Read data file from data disk; put waveform data in WD.
20090 to 20180	Plot data curves in workspace according to their order.
20190 to 20200	Return control to monitor after finished plotting; return to main program.
20210 to 20220	Program messages.
20230 to 20260	Error handling routines.

```
12-DEC-80  01:09  [OPTION I, LPRINT V5:BB ]
                                                    DX1:GRAPH1.BAS Page 1

20000 PRINT C$;'WOR 25'\PRINT C$;'GRA 1,35'\A7=1
20010 A9=0\DELETE A,WA,B,WB,C,WC
20020 PRINT 'DATA FILE';\INPUT G1$
20030 IF G1$='STOP' THEN 20190
20040 WAVEFORM WD IS D(511),SD,HD$,VD$
20050 ONERR AR GOTO 20240
20060 OPEN #1 AS DX1:G1$ FOR READ
20070 FOR I=0 TO 30\READ #1,G$\NEXT I
20080 READ #1,D,SD,HD$,VD$\CLOSE #1
20090 IF A9<>0 THEN 20110\IF A7=1 THEN 20110
20100 PRINT C$;'WOR 25'\PRINT C$;'GRA 1,35'\A7=1
20110 PRINT C$;'SHR'\PRINT C$;'WOR H'
20120 IF A9<>0 THEN 20140\VIEWPORT 200,800,580,760\SETGR VIEW,TICS 2,2,5,5
20130 GRAPH WD\RESETG\MOVE 820,670\PRINT G1$\GOTO 20200
20140 IF A9<>1 THEN 20160\VIEWPORT 200,800,320,500\SETGR VIEW,TICS 2,2,5,5
20150 GRAPH WD\RESETG\MOVE 820,410\PRINT G1$\GOTO 20200
20160 IF A9<>2 THEN 20180\VIEWPORT 200,800,60,240\SETGR VIEW,TICS 2,2,5,5
20170 GRAPH WD\RESETG\MOVE 820,150\PRINT G1$\GOTO 20200
20180 GOTO 20020
20190 PRINT C$;'MON H'\DELETE D,WD\ONERR\GOTO 1000
20200 IF A9=2 THEN 20190
20210 PRINT C$;'MON H'\PRINT 'TYPE STOP AFTER DATA FILE?'
20220 PRINT 'IF NO MORE GRAPH TO PLOT'
20230 A9=A9+1\A5=1\GOTO 20020
20240 CLOSE #1\IF AR(0)=0 THEN ONERR RETURN\IF AR(0)<>20050 THEN 20260
20250 PRINT CHR(AR(1));AR(2);'ERROR IN LINE';AR(0)\GOTO 20190
20260 PRINT G1$;' DOES NOT EXIST'\GOTO 20190
```

STATS1.BAS

STATS1.BAS is also overlaid on lines 20000 to 20260 when the "STATUS" command is executed. It reads the status register of the 7912AD, decodes its content and prints the status message on the 4025 terminal.

LINES	PURPOSE
20070 to 20090	Define the string array ER$; assign status code messages to ER$.
20100 to 20200	Read status register of the 7912AD, decode the content, print messages and return to main program.

```
20000 REM ROUTINE TO READ THE STATUS REGISTOR OF THE 7912AD
20010 REM
20020 REM
20030 REM
20040 REM
20050 REM
20060 REM
20070 DIM ER$(4)\ER$(0)='ILLEGAL CODE FOR 7912AD'\ER$(1)='COMMAND ERROR'
20080 ER$(2)='EXECUTION ERROR'\ER$(3)='INTERNAL ERROR'
20090 ER$(4)='POWER FAIL ERROR'
20100 GETSTA @0,ST,TA,SA\VARTST ST,'20',BZ\IF BZ=1 THEN PRINT 'DEVICE BUSSY'
20110 VARTST ST,'100',B2\IF B2=0 THEN 20200\PRINT 'SERVICE REQUEST'
20120 VARTST ST,'40',B2\IF B2=0 THEN 20140\M8=ST-ITP(ST/8)*8
20130 IF M8>4 THEN M8=0\PRINT ER$(M8)
20140 VARTST ST,'200',B2\IF B2=0 THEN 20160\PRINT 'REMOTE REQUEST'
20150 REM GOTO TO SUBROUTINE TO READ FRONT PANEL\GOTO 1000
20160 VARTST ST,'1',B2\IF B2=0 THEN 20170\PRINT 'POWER UP'\GOTO 1000
20170 VARTST ST,'2',B2\IF B2=0 THEN 20190\PRINT 'OPERATION COMPLETE'\GOTO 1000
20180 VARTST ST,'357',B2\IF B2=1 THEN 20190\PRINT 'NO CONDITION'\GOTO 1000
20190 PRINT ER$(0)\GOTO 1000
20200 PRINT 'NO SERVICE REQUEST'\GOTO 1000
```

READ1.BAS

READ1.BAS reads a data file from the data disk, prints the data header in the monitor and plots the data curve in the workspace. READ1.BAS is overlaid to the main program in response to the Read Command.

LINES	PURPOSE
20120	Delete old arrays and waveforms.
20130	Ask for data file.
20140	Define new waveform WD.
20150	Initiate error checking and handling routine.
20160 to 20230	Read the data file; print data header in monitor. Plot data curve in work-space; return to main program after finished.
20240 to 20260	Error handling routine.

```
12-DEC-80  01:13  [OPTION I, LPRINT V5:BB ]
                                                        DX1:READ1.BAS Page 1

20000 REM  ROUTINE TO READ WAVEFORM DATA ,PRINT PARAMETER FILE IN MONITOR
20010 REM  AND PLOT WAVEFORM IN WORKSPACE
20020 REM
20030 REM
20040 REM
20050 REM
20060 REM
20070 REM
20080 REM
20090 REM
20100 REM
20110 REM
20120 DELETE A,WA,B,WB,C,WC
20130 PRINT B2$\PRINT 'DATA FILE';\INPUT E$
20140 WAVEFORM WD IS D(511),SD,HD$,VD$
20150 ONERR AR GOTO 20240
20160 OPEN #1 AS DX1:E$ FOR READ\PRINT C$;'WOR 25'
20170 PRINT C$;'MON H'\FOR I=0 TO 30\READ #1,G$\PRINT G$\NEXT I
20180 READ #1,D,SD,HD$,VD$\CLOSE #1\PRINT C$;'ERA W'
20190 PRINT C$;'WOR H'\PRINT C$;'GRA 1,35'\PRINT C$;'SHRINK'\A7=1
20200 RESETG\MOVE 820,400\PRINT E$
20210 VIEWPORT 200,800,200,600\SETGR VIEW
20220 GRAPH WD\PRINT C$;'MON H'
20230 DELETE D,WD\ONERR\GOTO 1000
20240 CLOSE ALL\IF AR(0)=0 THEN ONERR RETURN\IF AR(0)<>20160 THEN 20260
20250 PRINT E$;' DOES NOT EXIST'\GOTO 20230
20260 PRINT CHR(AR(1));AR(2);'  ERROR IN LINE';AR(0)\GOTO 20230
```

CREATE.BAS

CREATE.BAS is used to format a blank disk before it can be used as a data disk. It initializes the disk by creating a directory area and three files. These files are "TITLE.TXT", "INITIA.LIZ" and "CATALO.G", and are located on the top of the disk so that they do not mix with the data files.

LINES	PURPOSE
10 to 30	Program message; ask operator's decision.
40	Initialize disk in drive 1 (DX1)
50	Check date; stop if date not set.
60	Enter disk sequence number; stop program if less than or equal to zero.
90 to 1000	Create a one block INITIA.LIZ file; enter the message, "EMPTY DATA DISC"
110 to 140	Create an 8 block CATALO.G file filled with "*" characters.
150	Assemble disk title DATXXX where XXX is the disk sequence number.
160 to 240	Create a one block TITLE.TXT file containing the disk title, sequence number, and comment entered by operator.
250	Delete the old CATALO.G to create 8 blocks of free space between the INITI.-LIZ and TITLE.TXT files.
260 to 290	Create a new 2 block CATALO.G file in the 8 blocks of empty space; enter the

	message, "DATA ON THIS DISC HAS NOT BEEN CATALOGED".
300 to 320	Return for next disk or exit program

```
12-DEC-80  00:59  [OPTION I, LPRINT V5:BB ]
                                                        DX1:CREATE.BAS Page 1

10 PRINT 'THIS PROGRAM WILL CLEAR THE DIRECTORY AND CONTENTS OF DX1:'
20 PRINT 'ARE YOU SURE THAT YOU WANT TO INITIALIZE DX1: (Y/N)';\INPUT M$
30 IF SEG(M$,1,1)='Y' THEN 40\STOP
40 ZERO DX1:
50 DATE D$\IF D$<>'         ' THEN 60\PRINT 'SET DATE'\STOP
60 PRINT 'ENTER DISC #(JUST THE NUMBER)';\INPUT N\IF N>0 THEN 80
70 GOTO 310
80 PRINT 'PLEASE WAIT 25 SECONDS'
90 OPEN #1 AS DX1:'INITIA.LIZ' FOR WRITE INTO 1
100 WRITE #1,'      EMPTY DATA DISC      ',0\CLOSE #1
110 OPEN #1 AS DX1:'CATALO.G' FOR WRITE INTO 8
120 FOR I=0 TO 55
130 A$='****************************************************************'
140 WRITE #1,A$\NEXT I\CLOSE #1
150 T1$='000'&STR(N)\T1$='DAT'&SEG(T1$,LEN(T1$)-2,LEN(T1$))
160 PRINT 'ENTER COMMENTS FOR ';T1$;' ENDING WITH 'END' AS THE FIRST THREE'
170 PRINT 'CHARACTERS OF A NEW LINE,THERE ARE 5 LINES FOR COMMENTS'
180 OPEN #1 AS DX1:'TITLE.TXT' FOR WRITE
190 WRITE #1,T1$,'  INTIALIZED ON ',D$\L=0
200 INPUT L$\IF L$='  ' THEN 230\IF L$='END' THEN 240
210 WRITE #1,L$\L=L+1\IF L=4 THEN PRINT 'THIS IS THE LAST LINE FOR COMMENTS'
220 IF L=5 THEN 240
230 GOTO 200
240 CLOSE #1\PRINT 'INITIALIZATION OF ';T1$;' IS COMPLETE'
250 CANCEL DX1:'CATALO.G'
260 OPEN #1 AS DX1:'CATALO.G' FOR WRITE INTO 2
270 WRITE #1,'DATA ON THIS DISC HAS NOT BEEN CATALOGED'
280 FOR I=1 TO 12\WRITE #1,'
290 NEXT I\CLOSE #1
300 PRINT 'ENTER -1 FOR NEXT DISC # TO EXIT PROGRAM'\GOTO 60
310 CLOSE ALL\PRINT 'FINISHED'\STOP
320 END
```

CATDIR.BAS

CATDIR.BAS is used at the end of a data run to enter new data file information into CATALO.G file on each data disk. The catalog of the data disk can be printed out, together with the title of the disk and comments entered by the operator, for a permanent record.

LINES	PURPOSE
10 to 70	Define B$="BELL", C1$="DX1: CATALO.G"; check "CATALO.G" to determine if it has been cataloged before; input message.
80 to 100	Enter "Y" to continue, 'N' to abort program.
110 to 220	Obtain directory information and put it in a temporary file "DIRECT.TMP"; Organize the information in "DIREC.TMP" and store in "CATALO.G"; Delete "DIRECT. TMP".
240 to 250	Enter "Y" to print catalog, "N" to stop program.
260 to 330	Read and assemble files "TITLE.TXT" and "CATALO.G"; print these files on the 4025 terminal.

12-DEC-80 01:00 [OPTION I, LPRINT V5:BB]

DX1:CATDIR.BAS Page 1

```
10 B$=CHR(7)&'** WARMING-'\C1$='DX1:CATALO.G'
20 OPEN #1 AS DX1:'CATALO.G' FOR READ
30 READ #1,A$\IF SEG(A$,1,4)='DATA' THEN 100
40 PRINT B$;'THIS DISK MAY HAVE BEEN CATALOGUED BEFORE'
50 PRINT ' THE FIRST LINE OF ';C1$;' IS--'
60 PRINT A$
70 PRINT 'DO YOU STILL WANT TO CATALOGUE IT (Y/N)';
80 INPUT L$\IF SEG(L$,1,1)='Y' THEN 100\IF SEG(L$,1,1)='N' THEN 240
90 CLOSE ALL\STOP
100 CLOSE #1
110 DIR DX1:TO DX0:'DIRECT.TMP'
115 CANCEL DX1:'CATALO.G'
120 OPEN #2 AS DX0:'DIRECT.TMP' FOR READ
130 OPEN #3 AS DX1:'CATALO.G' FOR WRITE INTO 7
140 READU #2,A1$=20
150 E$=' '
160 READU #2,G$=1
170 IF ASC(G$)=0 THEN 220\IF ASC(G$)=10 THEN 210
180 E$=E$&G$\GOTO 160
190 WRITE #3,E$\GOTO 150
200 GOTO 90
210 E$=SEG(E$,2,LEN(E$)-1)\GOTO 190
220 CLOSE ALL\CANCEL DX0:'DIRECT.TMP'
230 PRINT 'THIS DISK HAS BE CATALOGUED'
240 CLOSE #1\PRINT 'DO YOU WANT TO HAVE THE PRINT OUT OF THE CATALOG (Y/N)';
250 INPUT L1$\IF SEG(L1$,1,1)='N' THEN 90
255 FOR I=0 TO 4\PRINT\NEXT I
260 OPEN #1 AS DX1:'TITLE.TXT' FOR READ
270 EOF #1 GOTO 290
275 READ #1,F$,G$,H$\PRINT F$;G$;H$
280 READ #1,F$\PRINT F$\GOTO 280
290 CLOSE #1\PRINT\PRINT
300 OPEN #2 AS DX1:'CATALO.G' FOR READ
310 EOF #2 GOTO 325
320 READ #2,I$,J$\PRINT I$;'       ';J$\GOTO 320
325 FOR I=0 TO 4\PRINT\NEXT I
330 CLOSE #2\GOTO 90
```

FILTR1.BAS

This program does the digital filtering of the data by taking the Fast Fourier transform which can be done by executing the "RFFT" command of the SPS signal processing package. The undesired elements of the frequency spectrum are set to zero and the reverse of the transform is then performed. This program first displays the frequency spectrum of the data in the workspace of the 4025. The operator can then choose the upper and lower frequency limits. Both the original and filtered signal are displayed on the screen of the 4025 terminal.

LINES	PURPOSE
10 to 40	Define four waveforms, AA, BB, CC, and EE.
50	Enter data file.
60 to 90	Read data from data disk; perform the Fast Fourier Transform; results are stored in BB and CC.
100 to 160	Convert BB and CC into polar coordinates; store result in BB; plot BB, which is the frequency spectrum array, in the workspace; perform Fast Fourier Transform again on AA.
170 to 250	Enter upper and lower cutoff frequencies.
260 to 270	Assign zero values to the undesired frequency components.
280	Perform inverse Fast Fourier Transform.

290 to 370	Plot both the original signal and the filtered signal in the workspace.
390 to 400	Repeat the process if "Y" is entered.

12-DEC-80 01:07 [OPTION I, LPRINT VS:BB]

DX1: FILTR1.BAS Page 1

```
10 WAVEFORM AA IS A(511),DA,HA$,VA$
20 WAVEFORM BB IS B(256),DB,HB$,VB$
30 WAVEFORM CC IS C(256),DC,HC$,VC$
40 WAVEFORM EE IS E(511),DE,HE$,VE$
50 PRINT 'DATA FILE';\INPUT F$
60 OPEN #1 AS DX1:F$ FOR READ
70 FOR I=0 TO 30\READ #1,G$\NEXT I
80 READ #1,AA\CLOSE #1\GF=0
90 RFFT AA,BB,CC
100 POLAR BB,CC
110 PRINT CHR(33);'WOR 25'\PRINT CHR(33);'GRA 1,35'\PRINT CHR(33);'SHR'
120 PRINT CHR(33);'WOR H'
130 VIEWPORT 200,800,200,600\SETGR VIEW,GRAT 2,2\GRAPH BB
140 RESETG\MOVE 820,450\PRINT 'FREQ SPECTRUM OF'\MOVE 820,420\PRINT '    ';F$
150 PRINT CHR(33);'MON H'
160 RFFT AA,BB,CC
170 PRINT 'ENTER UPPER CUTOFF FREQ IN KHZ,-1 FOR HIGH PASS'
180 INPUT U
190 PRINT 'ENTER LOWER CUTOFF FREQ IN KHZ,-1 FOR LOW PASS'
200 INPUT L
210 IF U=-1 THEN 240
220 U1=U*1000/DB
230 FOR I=U1 TO 256\B(I)=0\C(I)=0\NEXT I
240 IF L=-1 THEN 270
250 L1=L*1000/DB
260 FOR I=0 TO L1\B(I)=0\C(I)=0\NEXT I
270 IF U=-1 THEN IF L=-1 THEN 170
280 RFFT EE,BB,CC,'INV'
290 IF GF<>1 THEN 310\PRINT CHR(33);'ERA W'\PRINT CHR(33);'GRA 1,35'
300 GOTO 320
310 PRINT CHR(33);'GRA 36,71'
320 PRINT CHR(33);'SHRINK'\PRINT CHR(33);'WOR H'
330 VIEWPORT 200,800,500,760\SETGR VIEW,GRAT 2,2\GRAPH AA
340 VIEWPORT 200,800,80,320\SETGR VIEW,GRAT 2,2\GRAPH EE
350 RESETG\MOVE 820,280\PRINT 'UPPER CUTOFF'\MOVE 820,260\PRINT U;'KHZ'
360 MOVE 820,240\PRINT 'LOWER CUTOFF'\MOVE 820,220\PRINT L;'KHZ'
370 MOVE 820,620\PRINT F$
380 PRINT CHR(33);'MON H'\GF=1
390 PRINT 'DO YOU WANT TO TRY OTHER FREQUENCIES Y/N';\INPUT Q$
400 IF Q$='Y' THEN 160
410 STOP
```

ANALYS.BAS

This program allows the operator to plot up to five data waveforms on the same coordinate axes. It also takes the 1st and 2nd integral and derivative of the stored data. The peak value of a waveform can be found by executing the "PEAK" command.

LINES	PURPOSE
10	Define string variables C$ and B2$.
500 to 506	Enter stop, plot commands or data file.
510	Initiate error handling routine.
520 to 560	Define waveform functions; read data file and print data header in monitor.
1000 to 1050	Accept "PEAK", "INTEGRATE" and "DIFFEREN-TIATE" commands.
1500 to 1520	Execution of "PEAK" command.
2000 to 2080	Enter 1st or 2nd integral; perform integration on data; plot the result in workspace.
2500 to 2580	Enter 1st or 2nd derivative; perform differentiation on data; plot the result in workspace.
10000 to 10040	Plotting subroutine
26000 to 26210	Define the necessary waveforms for "PLOT" command (maximum 5 waveforms); ask for data files names; read data disk; store in waveform arrays; plot all data curves on the same set of coordinate axes.
30000 to 31000	Error handle routine

```
12-DEC-80  01:02  [OPTION I, LPRINT V5:BB ]
                                                  DX1:ANALYS.BAS Page 1
10 C$=CHR(33)\B2$=CHR(7)
500 PRINT 'TYPE 'STOP' TO HALT PROGRAM,'PLOT' TO PLOT GRAPHS'
502 PRINT 'DATA FILES'!\INPUT E$\IN=0
505 IF E$='STOP' THEN STOP
506 IF E$='PLOT' THEN 26000
510 ONERR AR GOTO 30000
520 WAVEFORM WA IS A(511),IA,HA$,VA$
522 WAVEFORM WB IS B(511),IB,HB$,VB$
523 WAVEFORM WC IS C(511),IC,HC$,VC$
530 OPEN #1 AS DX1:E$ FOR READ\PRINT C$!'WOR 25'
540 PRINT C$!'MON H'\FOR I=0 TO 30\READ #1,G$\PRINT G$\NEXT I
550 READ #1,A,IA,HA$,VA$\CLOSE #1
560 GOSUB 10005\PRINT C$;'MON H'
1000 PRINT B2$\PRINT 'ENTER COMANDS'
1010 INPUT C2$\C2$=TRM(C2$)\C1$=SEG(C2$,1,3)
1020 IF C1$='PEA' THEN 1500\IF C1$='INT' THEN 2000
1030 IF C1$='DIF' THEN 2500\IF C1$='CUR' THEN 25000
1050 GOTO 1000
1500 LET M=MAX(WA)
1510 PRINT C$;'WOR H'\GOSUB 10035\PRINT 'PEAK VAL='!M\PRINT C$!'MON H'
1520 GOTO 1000
2000 PRINT '1ST OR 2ND INTEGRAL?1 OR 2';\INPUT IN
2010 IF IN>2 THEN 2000\IF IN<=0 THEN 2000
2020 WAVEFORM WB IS B(511),IB,HB$,VB$
2030 INT WA,WB\IF IN=1 THEN GOSUB 10000\IF IN=2 THEN 2040\GOTO 2065
2040 INT WB,WC\GOSUB 10000\GOTO 2065
2065 PRINT C$;'WOR H'
2070 IF IN=1 THEN PRINT '1ST INTEGRAL OF'\IF IN=2 THEN PRINT '2ND INTEGRAL OF'
2080 DELETE WA,WB,WC\GOTO 31000
2500 PRINT '1ST OR 2ND DERIVATIVE(1 OR 2)';\INPUT IN
2510 IF IN>2 THEN 2500\IF IN<=0 THEN 2500
2530 DIFF WA,WB\IF IN=1 THEN GOSUB 10000\IF IN=2 THEN 2540\GOTO 2565
2540 DIFF WB,WC\GOSUB 10000\GOTO 2565
2565 PRINT C$;'WOR H'
2570 IF IN=1 THEN PRINT '1ST DIFF OF'\IF IN=2 THEN PRINT '2ND DIFF OF'
2580 DELETE WA,WB,WC\GOTO 31000
10000 PRINT C$;'ERA W'
10005 PRINT C$;'GRA 1,35'\PRINT C$;'SHR'\PRINT C$;'WOR H'
10010 VIEWPORT 200,800,200,600\SETGR VIEW
10020 IF IN=0 THEN GRAPH WA\IF IN=1 THEN GRAPH WB\IF IN=2 THEN GRAPH WC
10030 RESETG\MOVE 850,380\PRINT E$
10035 MOVE 820,420
10040 RETURN
26000 DIM B$(5)\PRINT\PRINT 'NO OF GRAPH';\INPUT G\IF G>5 THEN 26130
26010 WAVEFORM WA IS A(511),IA,HA$,VA$\IF G=1 THEN 26060
26020 WAVEFORM WB IS B(511),IB,HB$,VB$\IF G=2 THEN 26060
26030 WAVEFORM WC IS C(511),IC,HC$,VC$\IF G=3 THEN 26060
26040 WAVEFORM WD IS D(511),ID,HD$,VD$\IF G=4 THEN 26060
26050 WAVEFORM WE IS E(511),IE,HE$,VE$
26060 PRINT B2$\PRINT 'DATA FILES '\FOR L=1 TO G\INPUT B$(L)\NEXT L
26070 FOR I=1 TO G\OPEN #1 AS DX1:B$(I) FOR READ
26080 FOR J=0 TO 30\READ #1,A$\NEXT J\GOSUB 26140\CLOSE #1\NEXT I
26090 PRINT C$;'WOR 25'\PRINT C$;'GRA 1,35'\PRINT C$;'SHR'
26100 VIEWPORT 200,800,200,600\SETGR VIEW,TICS 2,2,5,5
26110 PRINT C$;'WOR H'\GOSUB 26170\PRINT C$;'MON H'
26120 DELETE A,WA,B,WB,C,WC,D,WD,E,WE\GOTO 1000

                                                  DX1:ANALYS.BAS Page 2
26130 PRINT 'NO OF GRAHPS EXCEEDED 5,TRY AGAIN'\GOTO 26000
26140 IF I=1 THEN READ #1,A,IA,HA$,VA$\IF I=2 THEN READ #1,B,IB,HB$,VB$
26150 IF I=3 THEN READ #1,C,IC,HC$,VC$\IF I=4 THEN READ #1,D,ID,HD$,VD$
26160 IF I=5 THEN READ #1,E,IE,HE$,VE$\RETURN
26170 IF I=2 THEN GRAPH WA\IF I=3 THEN GRAPH WA,WB\IF I=4 THEN GRAPH WA,WB,WC
26180 IF I=5 THEN GRAPH WA,WB,WC,WD\IF I=6 THEN GRAPH WA,WB,WC,WD,WE\A1=500
26190 FOR I=1 TO G\RESETG
26200 MOVE 850,A1
26210 PRINT B$(I)\A1=A1-20\NEXT I\PRINT C$!'MON H'\GOTO 500
30000 CLOSE ALL\IF AR(0)=0 THEN ONERR RETURN\IF AR(0)<>530 THEN 30020
30010 PRINT E$;' DOES NOT EXIST'\GOTO 31000
30020 PRINT CHR(AR(0))!AR(2);' ERROR IN LINE';AR(0)\GOTO 31000
31000 PRINT C$;'MON H'\GOTO 500
```

CURFIT.BAS

The CURFIT.BAS program takes 129 data points out of a waveform data array and performs the least square polynomial curve fit. Both the original curve and the fitted curve are plotted in the workspace for comparison (see Fig. 6.13, 14). The coefficients of the polynomial are printed on the monitor.

LINES	PURPOSE
10	Clear all variables and arrays.
40 to 60	Define the arrays for storing data points; assign 129 values to independent array X.
70	Enter data file
80 to 160	Assign waveforms WD, WY, WZ; read waveform data from data disk; extract 129 data points from waveform data.
170 to 180	Enter order; -1 stop program.
200 to 310	Find the coefficients S(I) and T(I) required for solving the set of linear equations.
320 to 740	Find the (C) matrix by performing matrix inversion: (S) (C) = (T) $(C) = (T)\ (S)^{-1}$.
750 to 1005	Calculate the data array using the expression $f(X) = C_0 + C_1X + C_2X^2 + \cdots C_nX^n$.
1010 to 1100	Plot the original data curve and the curve generated by the polynomial expression in the workspace of the 4025.

12-DEC-80 00:55 [OPTION I, LPRINT VS:BB]

DX1:CURFIT.BAS Page 1

```
10 CLEAR
20 PRINT 'POLYNOMIAL CURVEFITTING ROUTINE,IT TAKES 129 DATA POINTS'
30 PRINT '                 THE HIGHEST ORDER IS 16'
40 DIM Y1(128),Z(128),Z1(128)
50 DIM T(50),X(128),Y(128),S(50)
60 FOR I=0 TO 128\X(I)=I*.1\NEXT I
70 PRINT CHR(7);'DATA FILE';\INPUT E$
80 WAVEFORM WD IS D(511),SD,HD$,VD$
90 WAVEFORM WY IS Y,SD,HD$,VD$\WAVEFORM WZ IS Z,SD,HD$,VD$
100 OPEN #1 AS DX1:E$ FOR READ
110 FOR I=0 TO 30\READ #1,G$\NEXT I
120 READ #1,D,SD,HD$,VD$\CLOSE #1
130 C1=CRS(D,0)\C1=ITP(C1)\J=0
140 FOR I=C1 TO 511 STEP 4
150 Y(J)=D(I)\J=J+1\NEXT I
160 SD=SD*4
170 PRINT CHR(7);'ENTER ORDER, -1 STOPS PROGRAM';\INPUT M
180 IF M=-1 THEN 1100\IF M>16 THEN 1110
190 REM TO FIND THE COEFFICICIENTS REQUIRED FOR SOLVING THE SET OF EQUATION
200 FOR K=0 TO 2*M\FOR I=0 TO 128
210 IF K=0 THEN X1=1\IF K=0 THEN 270
220 IF K=1 THEN S(K)=S(K)+X(I)
230 X1=X(I)\IF K=1 THEN 280
240 FOR J=1 TO K-1
250 X1=X1*X(I)
260 NEXT J
270 S(K)=S(K)+X1
280 IF K>M THEN 300
290 T(K)=T(K)+Y(I)*X1
300 NEXT I
310 NEXT K
320 REM MATRIX INVERSION FOR THE S(40,40) MATRIX
330 DIM S1(40,40)
340 N=M+1
350 FOR I=1 TO N
360 FOR J=1 TO N
370 S1(I,J)=S(J+I-2)
380 NEXT J\NEXT I
390 I=1\NX=N+1\NY=2*N
400 FOR J=NX TO NY\S1(I,J)=1
410 I=I+1\NEXT J
420 L=1\K=2
430 XM=S1(L,L)
440 FOR J=L TO NY
450 S1(L,J)=S1(L,J)/XM
460 NEXT J
470 FOR I=K TO N
480 X2=S1(I,L)
490 FOR J=L TO NY
500 S1(I,J)=S1(I,J)-S1(L,J)*X2
510 NEXT J\NEXT I
520 L=L+1\K=K+1
530 IF L-N<0 THEN 430\IF L-N=0 THEN 430
540 L=N
550 LZ=L-1
560 FOR K=1 TO LZ
```

```
570 I=L-K
580 Y2=S1(I,L)
590 FOR J=L TO NY
600 S1(I,J)=S1(I,J)-S1(L,J)*Y2
610 NEXT J\NEXT K
620 L=L-1
630 IF (L-1)>0 THEN 550
640 FOR I=1 TO N
650 REM TO OBTAIN THE COEFFICIENTS C(I) OF THE PLOYNOMIAL
660 DIM A(20,20),B(20,20),C(20)
670 J1=1
680 FOR J=NX TO NY
690 A(I,J1)=S1(I,J)\J1=J1+1
700 NEXT J\NEXT I
710 J=1\FOR I=1 TO N\B(I,J)=T(I-1)\NEXT I
720 FOR I=1 TO N\FOR K=1 TO N
730 C(I)=C(I)+A(I,K)*B(K,J)
740 NEXT K\NEXT I
750 REM R IS THE INDEPENDENT VARIABLE
760 FOR R1=0 TO 128\R=R1*.1
770 IF R=-1 THEN STOP
780 AA=C(1)+C(2)*R+C(3)*R*R+C(4)*R*R*R+C(5)*R*R*R*R
790 BB=C(6)*R*R*R*R*R+C(7)*R*R*R*R*R*R+C(8)*R*R*R*R*R*R*R
800 CC=C(9)*R*R*R*R*R*R*R*R+C(10)*R*R*R*R*R*R*R*R*R
810 DD=C(11)*R*R*R*R*R*R*R*R*R*R+C(12)*R*R*R*R*R*R*R*R*R*R*R
820 EE=C(13)*R*R*R*R*R*R*R*R*R*R*R*R
830 FF=C(14)*R*R*R*R*R*R*R*R*R*R*R*R*R+C(15)*R*R*R*R*R*R*R*R*R*R*R*R*R*R
840 GG=C(16)*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R
850 HH=C(17)*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R
860 II=C(18)*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R
870 JJ=C(19)*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R*R
880 YY=AA+BB+CC+DD+EE+FF+GG+HH+II+JJ
890 Z(R1)=YY\NEXT R1
900 C2=ITP(C1/4)\C3=C2*4\J=0
910 FOR I=0 TO C3 STEP 4\Y1(J)=D(I)\J=J+1\NEXT I
920 J=0
930 FOR I=C2+1 TO 128\Y1(I)=Y(J)\J=J+1\NEXT I
940 Y=Y1\DELETE Y1
950 DELETE A,B,T
960 J=0
970 FOR I=0 TO C3 STEP 4\Z1(J)=D(I)\J=J+1\NEXT I
980 J=0
990 FOR I=C2+1 TO 128\Z1(I)=Z(J)\J=J+1\NEXT I
1000 Z=Z1\DELETE Z1
1005 DELETE S,S1
1010 PRINT CHR(33);'WOR 25'\PRINT CHR(33);'GRA 1,35'
1020 PRINT CHR(33);'SHRINK'\PRINT CHR(33);'WOR H'
1030 VIEWPORT 200,800,200,600\SETGR VIEW,GRAT 2,2\GRAPH WY,WZ
1060 RESETG\MOVE 820,400\PRINT E$\MOVE 810,350\PRINT M;'TH ORDER'
1070 PRINT CHR(33);'MON H'
1080 FOR I=1 TO N\PRINT 'C(';I-1;')=';C(I)\NEXT I
1085 C=0
1090 DIM S(50),S1(40,40),T(50),Z1(128),Y1(128)
1095 GOTO 130
1100 STOP
1110 PRINT 'EXCEEDED THE HIGHEST ORDER'\GOTO 130
11092 Z=0\S=0\S1=0\T=0\Z1=0\Y1=0
```

APPENDIX B

COMMANDS

There are 17 commands recognized by the system. The commands are listed in alphabetical order, each followed by a brief description. A detailed description of each command can be found in Sec. 5.5.

ALL	Read all 32 channels of the DVM.
ACQUIRE	Digitize waveform data with the 7912AD.
CLEAR	Clear the 7912AD digitizer.
CRATE	Change crate address and subaddress.
DEFECT	Digitize target defects on the 7912AD.
DIFFERENTIATE	Perform differentiation on waveform data.
GRAPH	Plot stored data.
HALT	Terminate program execution.
INTEGRATE	Perform integration on waveform data.
PEAK	Find the maximum value of a data waveform.
PLOT	Plot different waveform data on the same set of coordinate axes.
NAME	Set operator's name.
PARMETER	Modify parameter file.
READ	Plot data stored on the data disk.
STATUS	Return status of the 7912AD.
STORE	Store data on data disk.

Appendix C

CHARACTERISTICS OF THE OVERALL SYSTEM

I. Speed

The speed of the whole system is limited by the slowest device of the system. The major components of the system are: a. the 7912AD digitizer, b. the HP12050A optical data link and c. the PDP 11/34 computer system.

The 7912AD works similar to an oscilloscope. The analog bandwidth is dependent upon the vertical plug-in amplifier. With the 7A21N vertical plug-in installed, a bandwidth of 1 GHz is obtained. The 7B92A horizontal plug-in provides a maximum sweep rate of 500 ps/div. The plug-ins for the existing system are 7A29 and 7B80. They provide a bandwidth of 500 MHz and a maximum sweep speed of 1 ns/div, respectively.

The 7912AD digitizes a waveform signal by writing the signal on a semiconductor target. Because the target is small, the writing beam need only be deflected over a small area (about 1.3 X 0.95 centimeters). Furthermore, only the writing beam scans at high-velocity (typical speed is about 8 div/ns); the reading beam scans more slowly. When the digitized waveform is read by the read gun, the target is scanned vertically by the reading beam in a 512 x 512 point format (Fig.7.1). These data are stored in the local memory of the 7912AD. The time required to read and store a waveform in the 7912AD memory is approximately 16.4 milliseconds. Data are transfered from the 7912AD to the PDP 11/34 through the IEEE 488 interface at a maximum rate of 710 k-bytes/second.

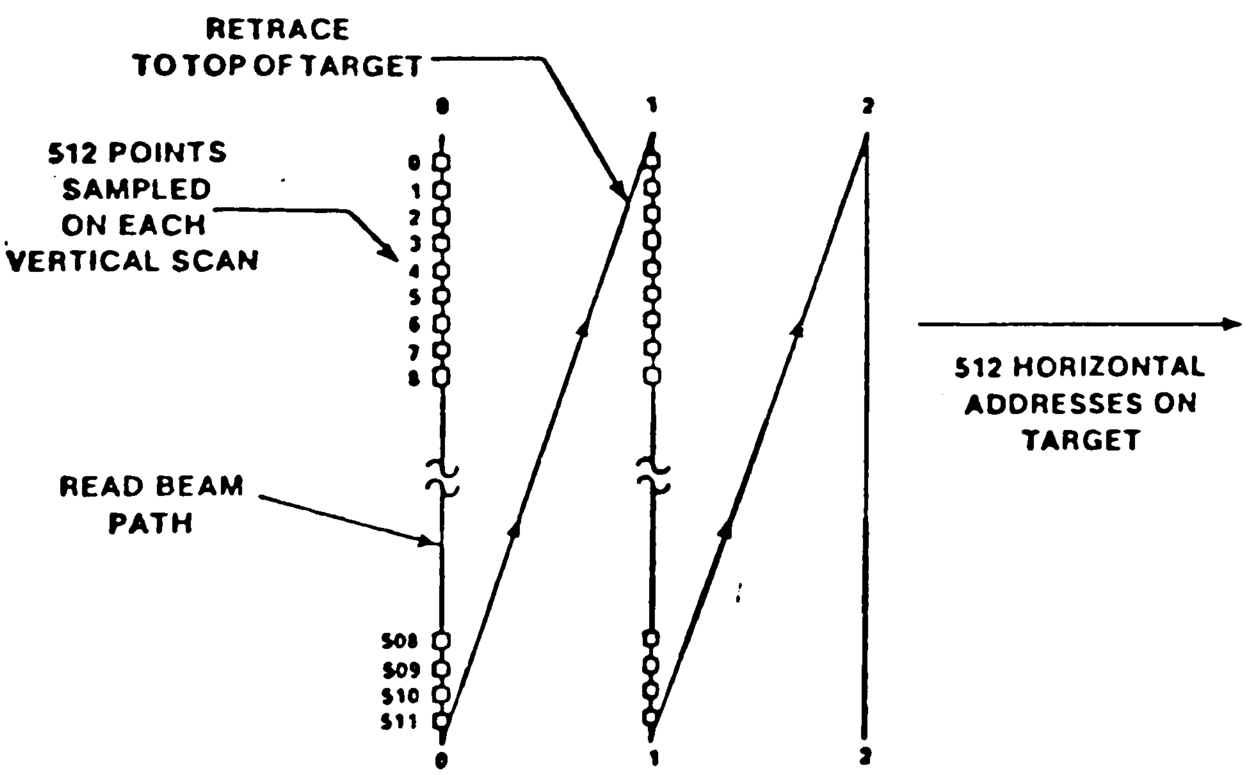


Fig. 7.1 Scanning of the Target by the Reading Beam

To transfer a waveform consisting of 1024 data points (512 vertical and 512 horizontal) takes a maximum time of 3 milliseconds. The digitizer data transfer time (T_{ddt}) is calculated as follows:

$$T_{ddt} = \frac{1024x2}{710} \text{ milliseconds} \quad \text{(required two bytes to represent a data point)}$$

$$= 2.88 \text{ milliseconds.}$$

Data are then stored in the MOS memory of the PDP 11/34. The typical access time of the MOS memory in use is about 400 ns; therefore, it requires about .8 millisecond to store the waveform data into the computer memory.

$$T_{ms} = 400 \times 10^{-9} \times (1024 \times 2) \text{ seconds}$$

$$= 8.192 \times 10^{-4} \text{ seconds}$$

$$= .8192 \text{ milliseconds}$$

(T_{ms} = memory storage time)

The longest time required is the time needed to store the data from memory on the floppy disk, because the average access time is 483 milliseconds. Since the maximum data transfer rate of the HP 12050A optical data link is 20,000 bytes/seconds, the maximum data transfer time is:

$$T_{odt} = \frac{1024 \times 2}{20} \text{ milliseconds}$$

$$= 102.4 \text{ milliseconds } (T_{odt} = \text{ optical data transfer time}).$$

Based on the above calculations, the addition of the HP12050A slows the system by a factor of 30, but it does not affect the speed of storing data because the I/O time of the floppy disk is about 4 times longer than the data transfer time of the optical data link. The following summarizes the speed performance of the system.

Device	Characterics
TEK 7912AD Digitizer	typical writing speed 8 division/ nanosecond, approximately 16.4 ms per waveform to read and store in local memory.
CP1100 IEEE 488 interface	710 kilobytes/second maximum data transfer rate. It takes approximately 3 milliseconds to transfer a waveform.
PDP 11/34 MOS Memory	Typical access time is 400 ns. It takes approximately 1 milli-second to store waveform data.
RX01 Floppy Disk	It takes approximately 483 milliseconds to complete its storage operation.
HP12050A Optical data link	The maximum data transfer rate is 20k-bytes/second. It takes about 102 milliseconds to complete the data transfer per waveform.

II. Resolution

Since the target array is 512 x 512, it requires 9 bits to represent 512 positions. The waveform data are stored in a 10 - bit word memory. The most significant bit is used as a flag. When this flag is set, it indicates that the number is a target defect instead of a valid data point. The other nine bits represents the physical position of the signal on the target.

III. Accuracy

The Accuracy of the digitizer depends mainly on the vertical plug-in amplifier. The relative accuracy of the present vertical plug-in is 2%. The raw data digitized by the 7912AD are sent to the PDP 11/34 where the binary data are processed and converted to a floating point number array. Figures 7.2 and 7.3 show a portion of the raw data array and a portion of the processed array. The raw data array consists of 9 bit binary numbers represented in decimal numbers, while the processed data array consists of floating point numbers which can be expressed as a decimal value ranging from +1.70141 38 to -1.70141E-38.

PRINT QQ

84	74	81	69	74
68	72	66	71	64
69	62	67	56	64
54	59	53	58	51
57	49	55	43	53
42	48	39	45	38
44	35	42	30	39
28	36	26	32	25
30	23	28	18	27
15	23	12	19	10
18	4	16	4	14
3	12	3	11	3
10	3	10	3	10
3	11	3	11	3
12	4	12	4	13
5	13	5	14	6
14	6	15	7	16
7	16	8	17	8
17	9	18	9	18
10	19	10	19	11
20	11	20	12	21
13	21	13	22	14
22	14	23	15	23
15	24	16	24	16
25	17	25	17	26
18	27	18	27	19
27	20	28	20	28
21	29	21	29	22
30	22	31	23	31

Fig. 7.2 A Portion of the Raw Data Array

PRINT C

.0244449	.0238199	.0230386	.0221011	.0214761
.0205386	.0197574	.0186636	.0181949	.0174136
.0166324	.0156949	.0146011	.0139761	.0133511
.0125699	.0114761	.0106949	9.75738E-03	9.13239E-03
8.19489E-03	7.41364E-03	8.03864E-03	6.94489E-03	6.31989E-03
5.38239E-03	4.60113E-03	3.97614E-03	2.56989E-03	1.94489E-03
1.47614E-03	1.00739E-03	8.51135E-04	6.94885E-04	8.51135E-04
8.51135E-04	1.16364E-03	1.16364E-03	1.47614E-03	1.47614E-03
1.78864E-03	1.78864E-03	2.10114E-03	2.25739E-03	2.41364E-03
2.56989E-03	2.72614E-03	3.03864E-03	3.03864E-03	3.35114E-03
3.35114E-03	3.66364E-03	3.66364E-03	3.97614E-03	3.97614E-03
4.28864E-03	4.28864E-03	4.60113E-03	4.75739E-03	4.91364E-03
4.91364E-03	5.22614E-03	5.22614E-03	5.53864E-03	5.69489E-03
5.85114E-03	6.00739E-03	6.16364E-03	6.47614E-03	6.47614E-03
6.78864E-03	6.78864E-03	7.10113E-03	7.10113E-03	7.10113E-03
7.41364E-03	7.56988E-03	7.72614E-03	8.03864E-03	8.03864E-03
8.35114E-03	8.35114E-03	8.66363E-03	8.66363E-03	8.97614E-03
8.97614E-03	9.13239E-03	9.28864E-03	9.60113E-03	9.60113E-03
9.91364E-03	9.91364E-03	.0102261	.0102261	.0105386
.0105386	.0108511	.0108511	.0111636	.0111636
.0114761	.0114761	.0117886	.0117886	.0121011
.0121011	.0124136	.0124136	.0127261	.0127261
.0128824	.0130386	.0130386	.0133511	.0133511
.0136636	.0139761	.0139761	.0141324	.0142886
.0144449	.0146011	.0147574	.0149136	.0152261
.0152261	.0153824	.0155386	.0156949	.0158511
.0158511	.0161636	.0164761	.0164761	.0164761
.0167886	.0169449	.0171011	.0172574	.0174136

Fig. 7.3 A Portion of the Processed Data Array