An Automation And Data Management System For An Electronic Autobalance

by

Bertram W. Murphy

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED: A. W. Bennett

A. W. Bennett, Chairman

R A Thompson

R. A. Thompson

CW Bostian

C. W. Bostian

May 1977

Blacksburg, Virginia

LM/MRJ 6/9/77

ACKNOWLEDGEMENTS

I wish to express my appreciation to Dr. A. W. Bennett for his help and guidance on this project and his suggestions and advice on the preparation of this thesis. I also wish to thank Alvin Schmitt for his assistance in the use of the operating system and his insights into design philosophies of system routines. I would like to thank Barbara Baldwin for typing the thesis and all members of the project for their aid and considerations. Finally, I wish to thank my wife, Christy, for her help in finalizing many of my thoughts and ideas and for her love and understanding throughout my graduate career.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

Introduction

1.1 Laboratory Automation

The application of the digital computer to process control has developed rapidly since the late 1950's. Most of the severe limitations and problems that plagued early users have been solved, and the far reaching advantages and great potential of the digital computer as a controller has been recognized [8]. The move toward automation in industry has also had profound effects in laboratory environments. The overwhelming majority of laboratory equipment manufacturers are now producing equipment with digital and/or analog outputs that can easily be used directly as inputs to processing units [4]. Add to this the ever increasing availability of medium powered, low cost computers, and the stage is set for the rapid advance of laboratory automation system technology experienced in the last few years.

The advances in laboratory automation have brought new system requirements. Systems must not only interface with laboratory equipment, but must meet the often rigid demands imposed by each experiment, as well as be a willing servant to the experimenter [9]. A laboratory automation system must be capable of collecting all required data, analyzing it, and generating appropriate reports. All this must be done while remaining in the background, as the experiment itself is of primary importance, and not the mecahnics of acquisition and analysis [9].

When these requirements are met and a competent automation and management system is implemented, throughput can be increased at least forty percent, as time required for instrument attendance, report generation, calculations, recording, and recordkeeping is greatly reduced [4]. Human error is, to some extent, eliminated; and accuracy of results is usually increased. The added sensitivity of the computer over ordinary integrators or chart recorders also enhances system performance. In general, laboratory efficiency and productivity are increased by computer automation.

The subject of this thesis is the description of an electronic autobalance data acquisition and management package and its interaction with and control by such a laboratory automation system.

1.2 The Problem

An electronic autobalance is used by the Division of Consolidated Laboratory Services of the State of Virginia (DCLS) to determine the amount of solid pollutants in the air at various locations throughout the state. Clean paper filters weighed to a precision of ±0.0001 grams at the laboratory in Richmond. The weight and corresponding filter identification number are kept on file and the filters sent to the pollution monitoring stations located throughout the state. Air is passed through each filter to collect the solid particulate in the atmosphere, and the exposed filters are returned to the Richmond laboratory for reweighing. By subtracting the clean weight from the exposed weight and accounting for the volume of air passed through a given filter, an indication of air quality can be determined.

There is often several months delay between initial weighing of clean filters and the return of exposed filters for reweighing. As a result, approximately 12,000 clean filter weights and their identification numbers must be kept on file.

In order to maintain laboratory standards, various quality control analyses are periodically carried out on the system. This includes an initialization process in which 100 clean or exposed filters are weighed and set aside for later reweighing. This is done to determine the effects of variations in temperature, moisture content, etc. After a delay determined by the operator, filters are reweighed and weight differences used to establish upper and lower bounds on weight variation for future quality control work.

The next step in the analysis is to conduct one of three different random sample checks. These are a reduced, a normal, and a tightened sample check in which random filters (clean or exposed) are periodically reweighed in the above manner to ensure that weight variations fall within predetermined bounds. The three checks differ by the number of filters sampled and the required percentage of passing filters. The pass/fail history of the filters determines which of the sample checks is used [7]. A statistical analysis is also conducted to calculate the average and the distribution by range of the residue weights. These results are also kept on file.

When the weight of the pollutant residue is determined, a report is generated containing the filter identification number, the clean filter weight, the exposed filter weight, the residue weight, the air flow rate through the filter, the time, the temperature, the date, the operator number, and the experimental station number. Special reports are also generated for quality control and statistical information.

To facilitate these operations and increase throughput, it was decided to add the autobalance to the DCLS Laboratory Automation and Data Management System under developement by the Computer Engineering Laboratory at VPI&SU.

The autobalance automation system must determine and store the weight of each clean filter for later recall when the filter is returned for reweighing. When the exposed filters are received, the residue weight is calculated by subtracting the clean filter weight from the exposed filter weight. At the end of a weighing session, an appropriate report is generated. The exposed filters return randomly from the experimental stations, and some are never returned. Thus, the system must be able to purge old, unused records from the files. To compensate for the increase in throughput, the system would need to store up to 20,000 clean filter weights. The system must also be capable of conducting quality control and statistical analysis and providing reports on this phase of system performance.

1.3 Thesis Organization

In order to provide an orderly discussion of system development, this thesis is divided into five chapters. Chapter one presents a brief discussion of automated process control in general and gives an introduction to the use of the autobalance at the Division of Consolidated Laboratory Services (DCLS).

Chapter two given a brief overview of the DCLS Automation and Data Management System, as well as a general description of the proposed autobalance system.

Chapter three discusses each routine in the autobalance system. The chapter also includes flowcharts of each routine.

Chapter four illustrates typical weighing sessions and gives a feel for the flow of the system.

Finally, Chapter five summarizes the proposed autobalance system and gives brief suggestions for future action.

CHAPTER II

System Overview

2.1 System Description

The Laboratory Automation and Data Management System under development by the Computer Engineering Laboratory for DCLS is a modular system, utilizing a minicomputer to service a cluster of laboratory instruments. Each minicomputer is called a Remote Instrument Service Unit (RISU) and can handle a maximum of 16 instruments. The RISU's can be connected to a Central Control Unit (CCU) for reporting and control supervision. Each RISU (see Fig. 2.1.1) includes a Texas Instruments 980 B minicomputer, along with the necessary interface hardware and software to monitor and control a group of laboratory instruments [2]. This includes an Advanced Electronics Design floppy diskette controller, a 16 channel multiplexer, a maximum of three terminals, and a varying number of instrument interface boxes. Several systems are currently under development and will run on the RISU. These will service the following laboratory instruments: an autoanalyzer, a gas chromatograph, and a spectrophotometer.

The autobalance system will likewise run on a standard RISU. The need for mass storage will be satisfied by the floppy diskette, which is capable of storing up to 142,336 sixteen bit words on each side [6]. The autobalance will be connected to the system by one of the standard instrument interface boxes, with a special printed circuit card to handle the data requirements of the autobalance [1]. Data flows through the multiplexer to the 980 B. The weight calculations, quality control,

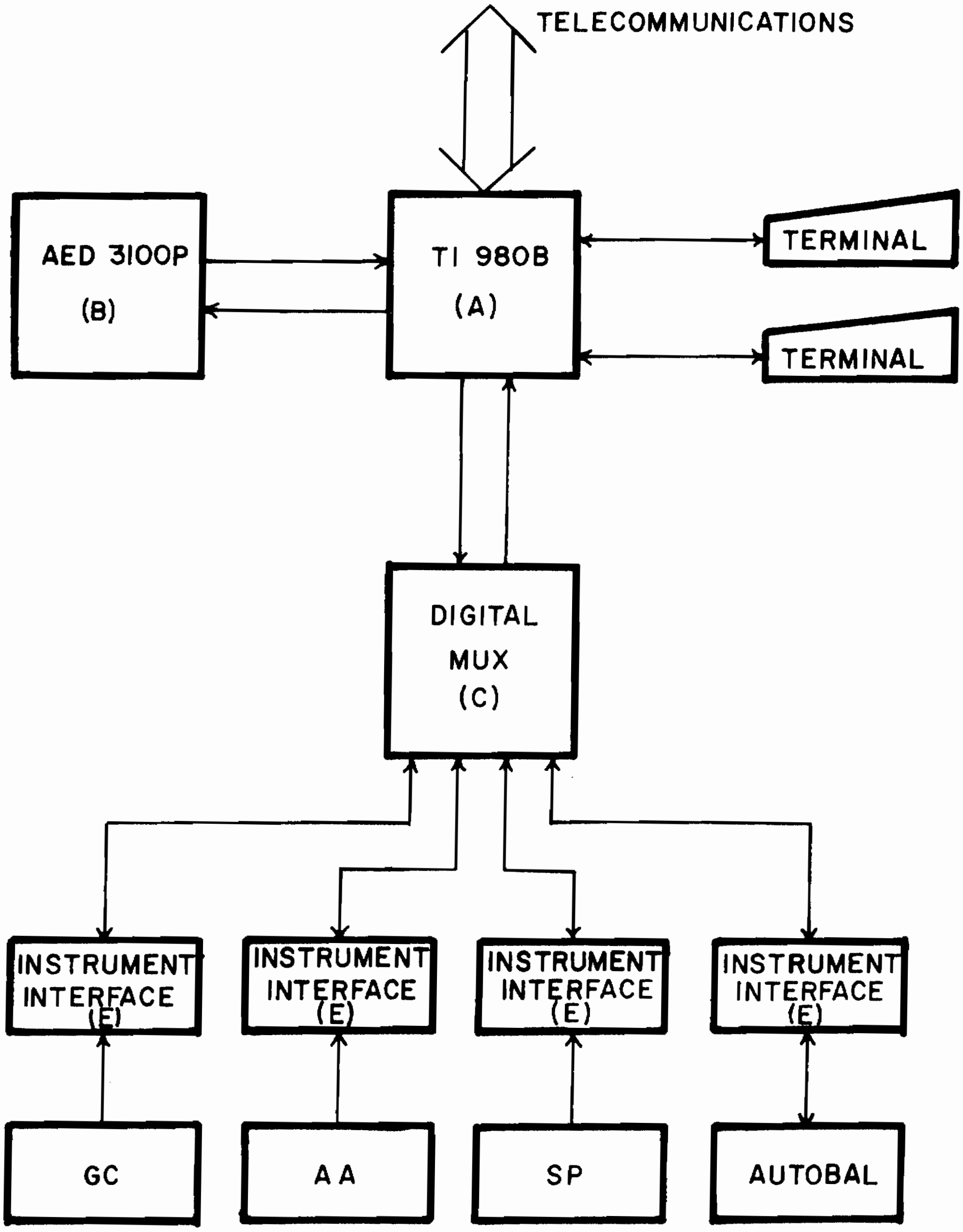

FIGURE 2.2.1 RISU Configuration

and statistical analyses are carried out by appropriate routines working in a multitasking environment. There will be a terminal for each autobalance, and all reports will be printed out on one of these. Implementing the autobalance system on a standard RISU will save both time and money; as much benefit will be gained from the extensive effort already conducted. In addition, it will add to the flexibility of the RISU concept.

2.2 System Hardware

The heart of each RISU is the TI 980 B minicomputer (block A of Fig. 2.1.1). This is a 16 bit, parallel operation machine especially suited for real-time process control [5]. The computer is divided by function into a central processing unit (CPU), a memory , an input/output (I/O) unit, and a power supply. The CPU has bipolar ROM control and features eight addressable registers, two's complement arithmetic, 750 nanosecond read or write cycle, and 500 nanosecond memory access. The computer offers up to 64K words of directly addressable, dynamic MOS/LSI semiconductor main memory with battery backup. Several addressing modes are available, including program counter relative, base register relative, index register relative, and single level indirect addressing. For I/O, the computer uses one direct memory access channel (expandable to eight) and a four port (expandable to 256) processor-controlled data buss. The 980 B also has power failure protection, a memory protect/privileged instruction feature, and a versatile set of 99 basic instructions [5].

The AED 3100P floppy disk controller (block B of Fig. 2.1.1) offers an on-line system bulk storage for the RISU. It contains one to four disk drive units which access one side of a floppy disk. The disk must be manually turned over to access the other side. Each drive has a rotational speed of 360 rpm. The data transfer rate of the controller is 250 kilobits/sec. Average latency (1/2 rotation) is 83 msec. Track to track access time is 11 msec., and head settling time at the last track is 11 msec [6]. Each disk is divided into 77 tracks (per side). The recording density of the disk is 3200 bpi on the inside track [6]. This allows storage of 1856 16-bit words per track. Each track is divided into 29 sectors of 64 words each, or 2233 sectors per side. This represents a storage capacity of approximately 142K 16-bit words per side. The controller presently being used has two drive units. This allows for approximately 284K words of on-line storage. The floppy disk controller is linked to the TI 980 B by a 16-bit parallel, bi-directional data bus and seven control lines. These are connected through an interface in the I/O expander rack of the TI 980 B [6].

The 16 channel digital multiplexer (block C of Fig. 2.1.1) is the link between the TI 980 B and each instrument interface box. It allows data transfer in either direction (to or from the CPU) and services each data channel ten times/sec.

The instrument interface (II) box (block E of Fig. 2.1.1) provides the data path between the laboratory instrument and the multiplexer. It also performs signal transformation or conversion necessary to insure the data is compatible to the multiplexer input. The instrument interface box will perform analog-to-digital conversion and/or code reduction.

There are five indicator lights and four control pushbutton switches on the front panel of the II box. The indicator lights are as follows: ready, running, report, attention, and power on. The switches provide the following functions: start, stop, abort, and special (user assigned). The switches are used to initiate or terminate an experiment session, and the lights indicate the status. Included in the II box is a program controllable relay. The contacts of this relay can be opened or closed as requested by the currently running routine.

2.3 System Software

The laboratory management system software consists of various service routines and a skeletal operating system "kernel". This operating system, know as DSOS (Deep Six Operating System), was developed for the TI 980 B by Dennis S. Frailey of Southern Methodist University while working as a consultant to Texas Instruments, Inc. [3]. It was designed primarily to support real-time operations. The operating system sets up a multitasking environment with priority scheduling of system routines. It also provides software interrupts, timers, and real-time clocks.

Under DSOS, memory is organized into a shell structure for task priority, with tasks located in shell zero having the highest priority. This structure also provides memory protection. Each shell is further divided into a code block and a changing data block. Space in the data block may be permanently assigned or dynamically allocated. No data can be located in the code block, and a check sum routine is provided to ensure the integrity of the block. The system is organized in this way to support re-entrant code. Once a routine is written, any number of instruments may call it, and all will be serviced. This is necessary to avoid duplication, as all routines must permanently reside in core. I/O routines and interrupt handlers are located in shell zero (highest priority). The majority of the operating system is located in shell one, and system routines are in shell two or higher. Shell seven contains the tasks with lowest priority.

Three of the system service routines work closely with the autobalance system. The Multiplexer Connection Software, the system Command Processor, and the Disk File Management Package are an existing part of the automation system and will be used by the autobalance system in performing its functions.

The multiplexer connection software is a group of routines that monitors the multiplexer and passes information to and from it. All data going to or coming from the instrument interface boxes must pass through the multiplexer connection software.

The system command processor prompts for and acts upon commands obtained from the terminal. Data is also passed from the terminal to the system through the command processor.

The Disk File Management Package (DFMP) is a set of routines created to communicate with the floppy disk. DFMP organizes the sectors on a disk into "files" determined by the users. To read from or write into a sector, the disk must be accessed, a file defined, and the file opened. Once this is done, any number of sector operations can be conducted. At the end of these operations, the file must be closed and the disk detached.

2.4 Autobalance Hardware

The specific balance that will operate under the control of the autobalance automation system at DCLS is a Hi-Vol Electronic Autobalance, Model BA28. Much of the customary signal processing and control is done by the autobalance. This simplifies the requirements of the instrument interface. In conducting its standard weighing operations, the autobalance associates a constant tare weight with each measurement. This is electronically subtracted from every weight. The weight is then sent to an LED display and to output circuitry in BCD form. To initiate data transfer, a "weight request" signal must be received by the balance; and the weight information, once issued, will remain on the balance output (and the display) as long as the weight request signal is present. If the scale is not stable (in motion) or is overloaded, the balance will lock out output until the error condition is corrected.

The autobalance is connected to the RISU through a standard instrument interface (II) box. A special printed circuit card is added to the II box. This contains the circuitry used to latch the weight from the autobalance [1]. The II box issues the weight request signal to the balance and holds it until the weight is transferred. The weight request is developed by the program controllable relay. The II box will also convert the BCD data to a binary form acceptable to the 980 B computer.

2.5 Autobalance Software

The autobalance automation software consists of five service routines and one disk resident, stand alone routine (see Fig. 2.5.1)

The five service routines operate in an asynchronous mode, with handshaking between routines accomplished by the use of common data areas that are opened or closed by software locks. The five routines are the Spin Task, the Block Handler, the Disk Routine, the Print Routine, and the Quality Control Routine. The stand alone routine is the Data Disk Formatter. In discussing the autobalance software, the routines will be discussed in the order of information flow. This will provide additional insight into the operation of the system.

The main responsibility of the Spin Task (see Fig. 2.5.2) is acquiring the filter weight and port number for the system while freeing the data blocks passed to it by the multiplexer connection software. The weight and port number are placed in main memory, the data block is returned, and a flag is set indicating a new weight value has been received.

The Block Handler (see Fig. 2.5.3) has two responsibilities. First, it is responsible for operation of the indicator lights and the relay and monitoring of the control switches in the instrument interface box. All of this is done through the multiplexer connection software. Second, it is responsible for taking the weight and port number from the Spin Task (through the common data area in main memory) and moving them to the sample table, also in main memory. The filter identification number is also placed in the sample table through commands executed by the system command processor.

The Disk Routine (see Fig. 2.5.4) next takes the identification number and filter weight and, depending on whether it is a clean or

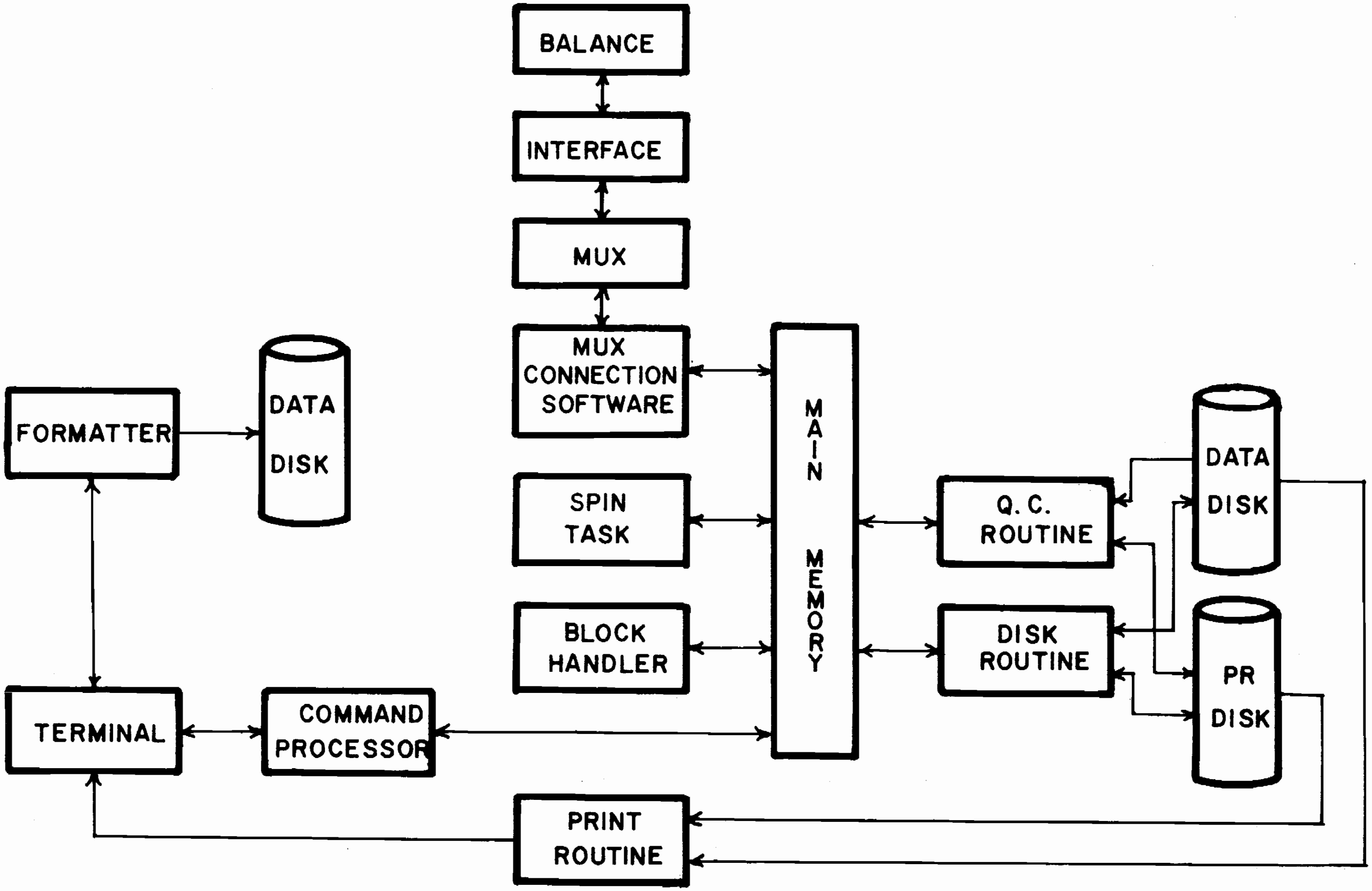

FIGURE 2.5.1 Autobalance System

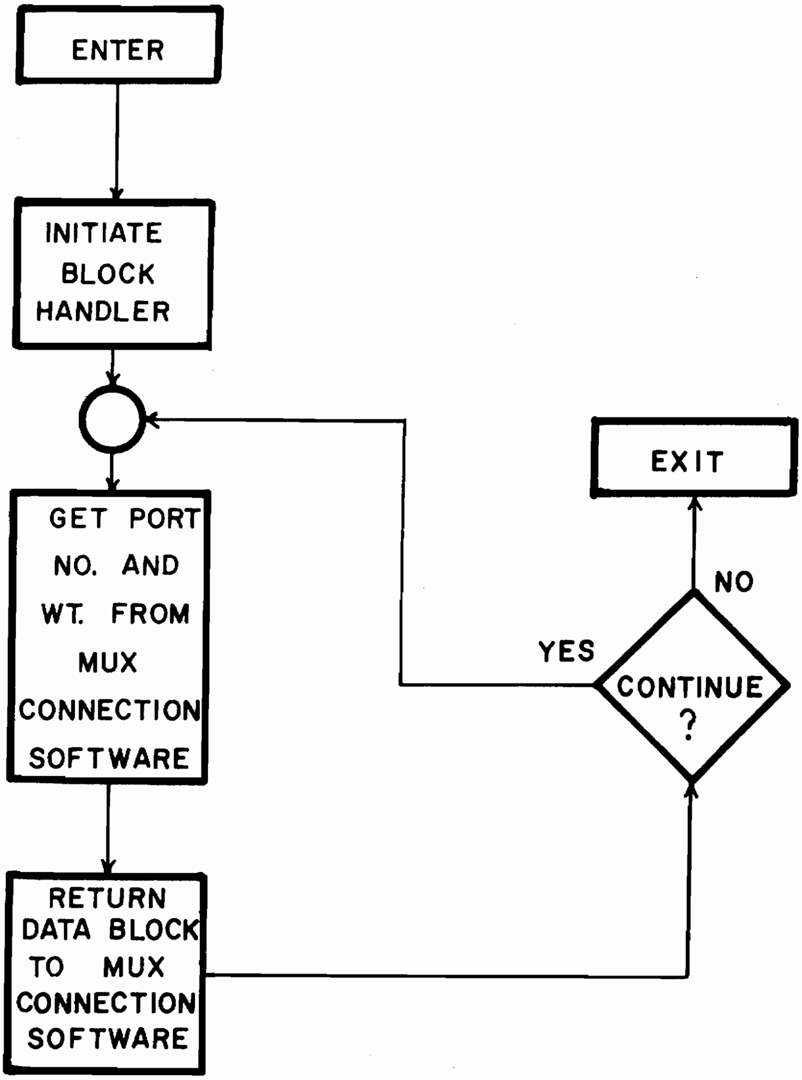
ENTER
INITIATE BLOCK HANDLER
GET PORT NO. AND WT. FROM MUX CONNECTION SOFTWARE
RETURN DATA BLOCK TO MUX CONNECTION SOFTWARE
CONTINUE ?
YES
NO
EXIT

FIGURE 2.5.2 Spin Task

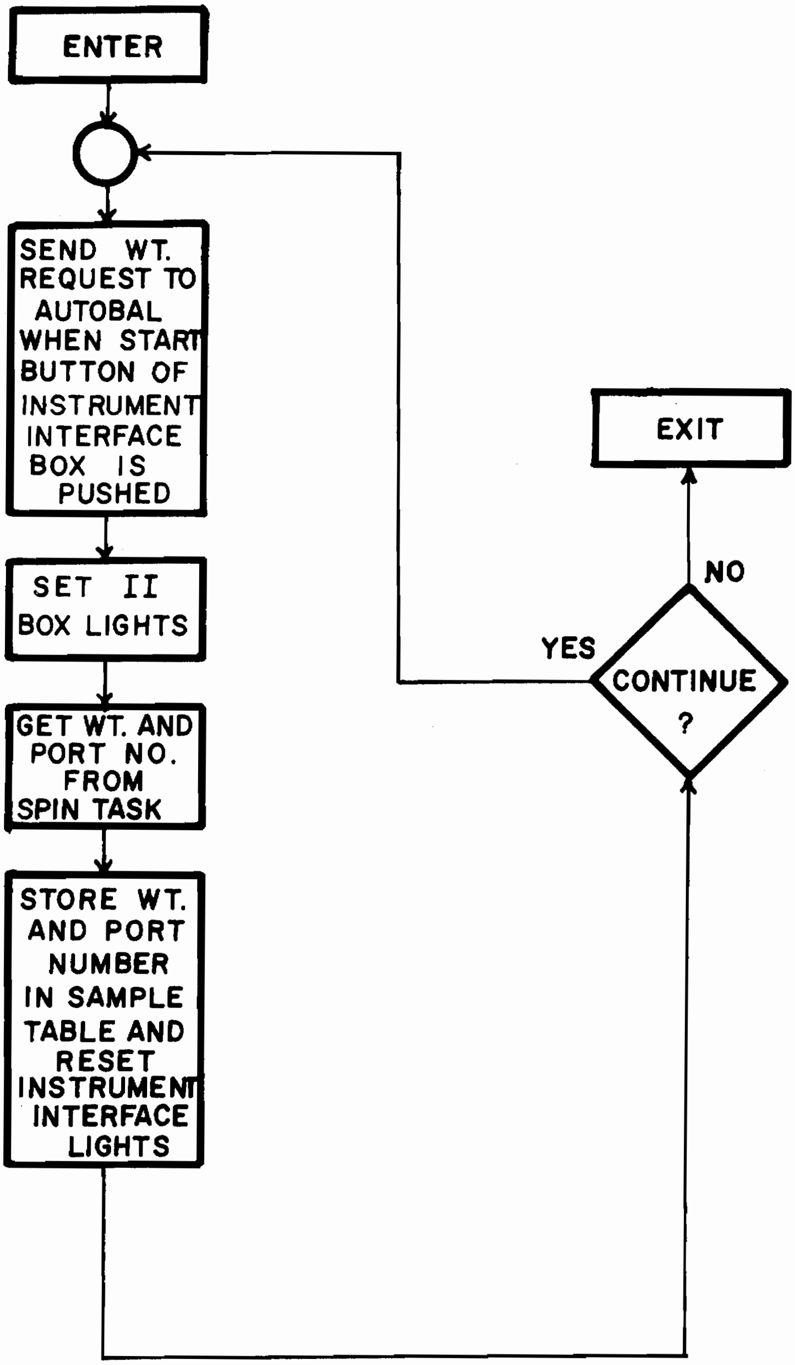
ENTER
SEND WT. REQUEST TO AUTOBAL WHEN START BUTTON OF INSTRUMENT INTERFACE BOX IS PUSHED
SET II BOX LIGHTS
GET WT. AND PORT NO. FROM SPIN TASK
STORE WT. AND PORT NUMBER IN SAMPLE TABLE AND RESET INSTRUMENT INTERFACE LIGHTS
CONTINUE ?
YES
NO
EXIT

FIGURE 2.5.3 BLOCK HANDLER

exposed filter weighing session, stores the weight on the data disk or calculates the residue weight and stores the sample information on the printout disk.

At the end of an exposed filter weighing session, the Print Routine (see Fig. 2.5.5) will output (printout on a standard terminal) the samples stored on the print disk. The Print Routine is also responsible for printing all error messages during system operation and for printing out any sample that is over a year old. This releases memory space by purging filter weights which have not been returned after one year.

The Quality Control Routine (see Fig. 2.5.6) executes the desired quality control and statistical operations and prints out the result.

As part of the method for sample addressing, the sectors on the data disk must be assigned consecutive addresses from 1-2080. The standard DSOS disk formatter and the Autobalance disk formatter work in sequence to accomplish this.

2.6 Summary

In this chapter an overview of the hardware and software of the Laboratory Automation and Data Management System has been presented. The chapter also includes a discussion of the autobalance system which provides a description of the hardware, the data structure, and an introduction to the routines required for operation. In Chapter three, each of the autobalance system routines will be discussed in detail.

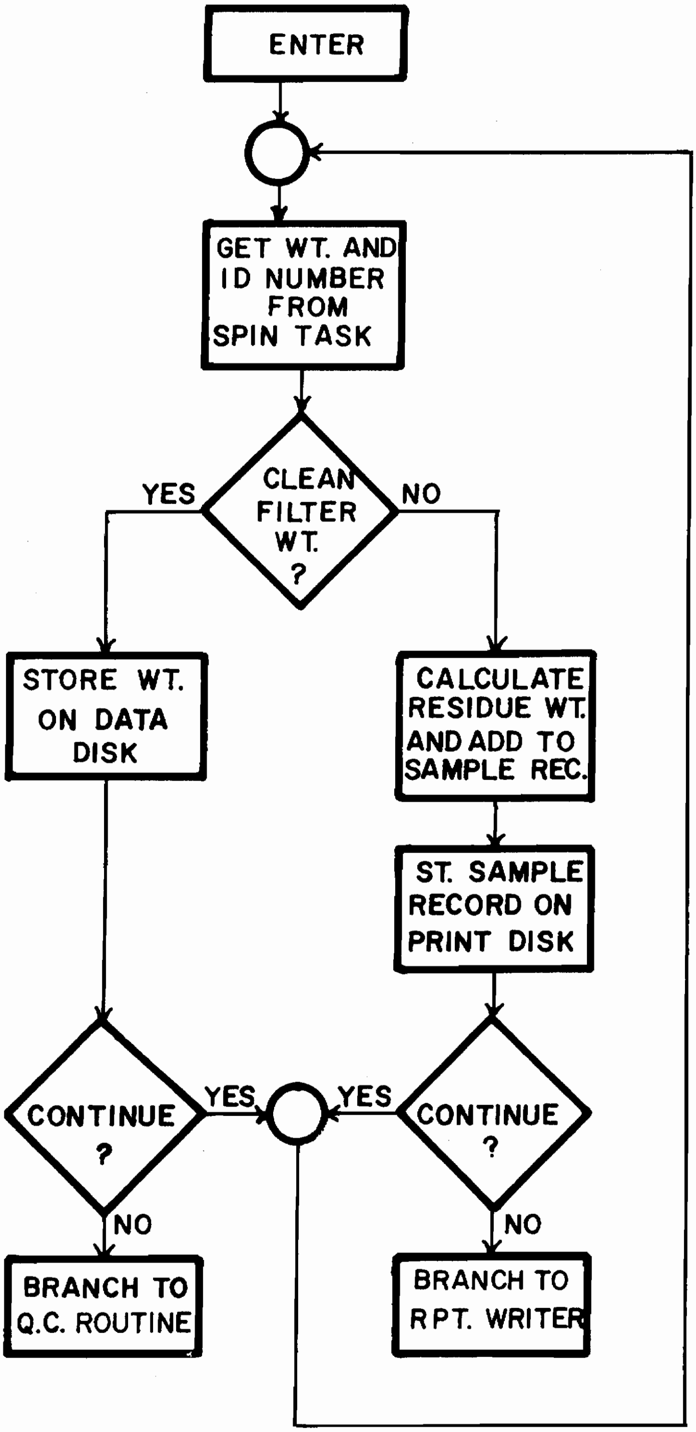

FIGURE 2.5.4 DISK ROUTINE

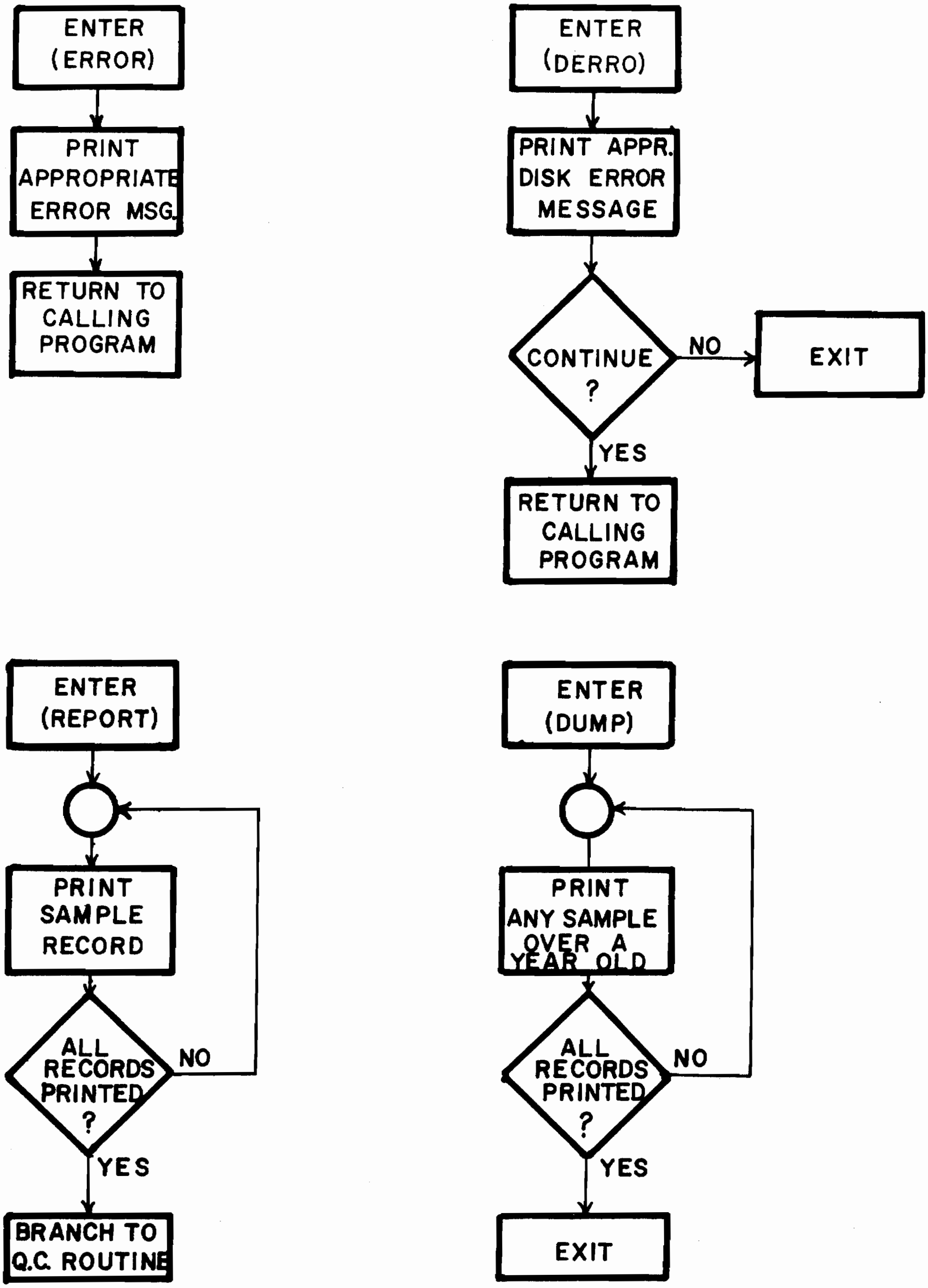

FIGURE 2.5.5 PRINT ROUTINE

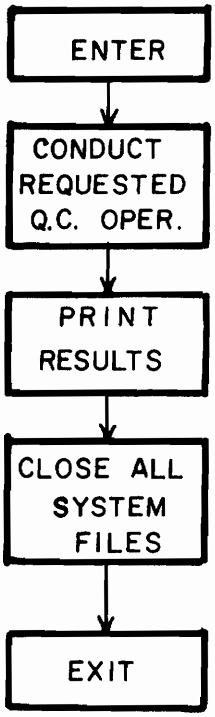

FIGURE 2.5.6 QUALITY CONTROL ROUTINE

CHAPTER III

Autobalance System Routines

3.1 Data Structure

The computer operations required to support the autobalance system can be divided into data management and data reduction. Although both operations are important, the data management required for this system is more complex. To provide a basis for a better understanding of the system service routines, the data management techniques will be discussed in this section. The methods of data reduction will be discussed in the sections on specific service routines.

The clean and exposed filter weights are gathered and processed by the autobalance system. During a clean filter weighing session, a six digit identification number is entered along with a corresponding filter weight; and the date is obtained from the DSOS system time/date service routine. The clean filter weight and the date constitute the clean filter sample record. Sample records are stored on the data disk (requiring four words of storage) and must be available for recall randomly as identified by their ID number.

To facilitate this, the data disk is organized to allow each ID number to act as a pointer to the disk location containing the weight of the corresponding filter. The data disk is first formatted by the standard DSOS formatter which establishes 2,233 64-word sectors. The autobalance formatter routine then assigns consecutive addresses (1,2080) to 2080 sectors, with the remianing sectors of the disk being required by the system Disk File Management Routines.

The first 2015 sectors (address number 1-2015) are used for clean filter sample record storage with ten samples/sector (six words/sample). This provides storage for a maximum of 20,150 sample records. The last 65 sectors (address number 2016-2080) are used as a sample record directory (see Fig. 3.1.1). Each sector in the directory is divided into 31 sector labels (two words each), a sector address word, and a word pointing to the next sector in the directory. These pointers link all the directory sectors, and the 2015 combined sector labels point to each of the 2015 sample record sectors (i.e., label one points to sector one, etc.). The five most significant digits of the six digit ID number (and the leading letter if used) are stored in a sector label. The ten sample records having an ID number with these five most significant digits are then stored in the sector pointed to by the label, with the least significant digit of the ID number used to point to one of the ten sample locations within the sector. At the beginning of every weighing sessions, the entire sample directory is read from the data disk into main memory. Only one disk read operation is then needed to obtain any random sample weight.

After a sample record containing a clean filter weight has been transferred from the data disk to main memory, the system can proceed to calculate the pollutant residue weight. Once this is complete, there is no longer any need for that sample to remain on the data disk; and the space should be available for reuse. Each sample weight only occupies one and one-half of the two words of storage assigned to it. This leaves eight bits free. One of these (designated the Weight Read

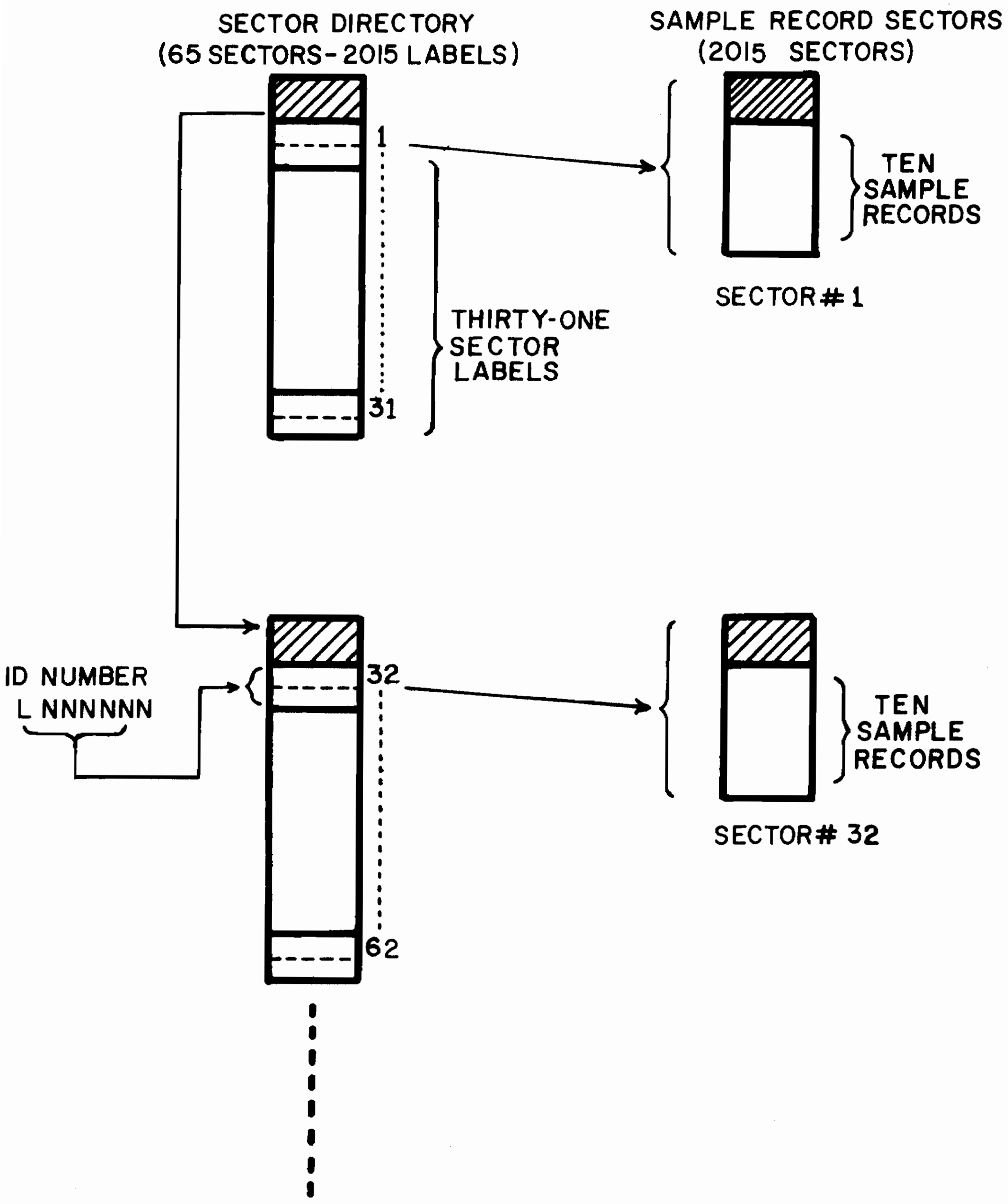

FIGURE 3.1.1 DATA DISK ORGANIZATION

Flag) is used to signify when a weight has been read back into the system to calculate the residue weight. When all sample weights in a sector have their Weight Read Flag set so as to indicate they have been read, a zero is placed in that sector's label in the sample directory. This indicates that this sector is available for reuse, and new clean filter sample records can be stored in the sector.

An exposed filter sample record consists of the clean filter weight (read from the data disk), the exposed filter weight (acquired from the autobalance), the calculated residue weight, and the filter ID number. At the time the exposed filter is returned for reweighing, the operator will enter the time (filter was exposed), temperature, date, experimental station number, the operator number, and the flow rate of the air through the filter at the time it was exposed. Each exposed filter sample record requires 15 words of storage and is stored on the print disk, four sample records per sector. At the end of the weighing session, all sample records are printed out consecutively (no printer interruptions by other system routines) to form the filter weight report.

3.2 Spin Task

Every tenth of a second, the multiplexer (mux) connection software checks the input to the mux for any available information from the instrument interface (II) boxes. The input present at each port is used to form a data block for that port and is passed by the multiplexer connection software to any currently running routine that requests it. If left unattended, the mux software will eventually run out of space

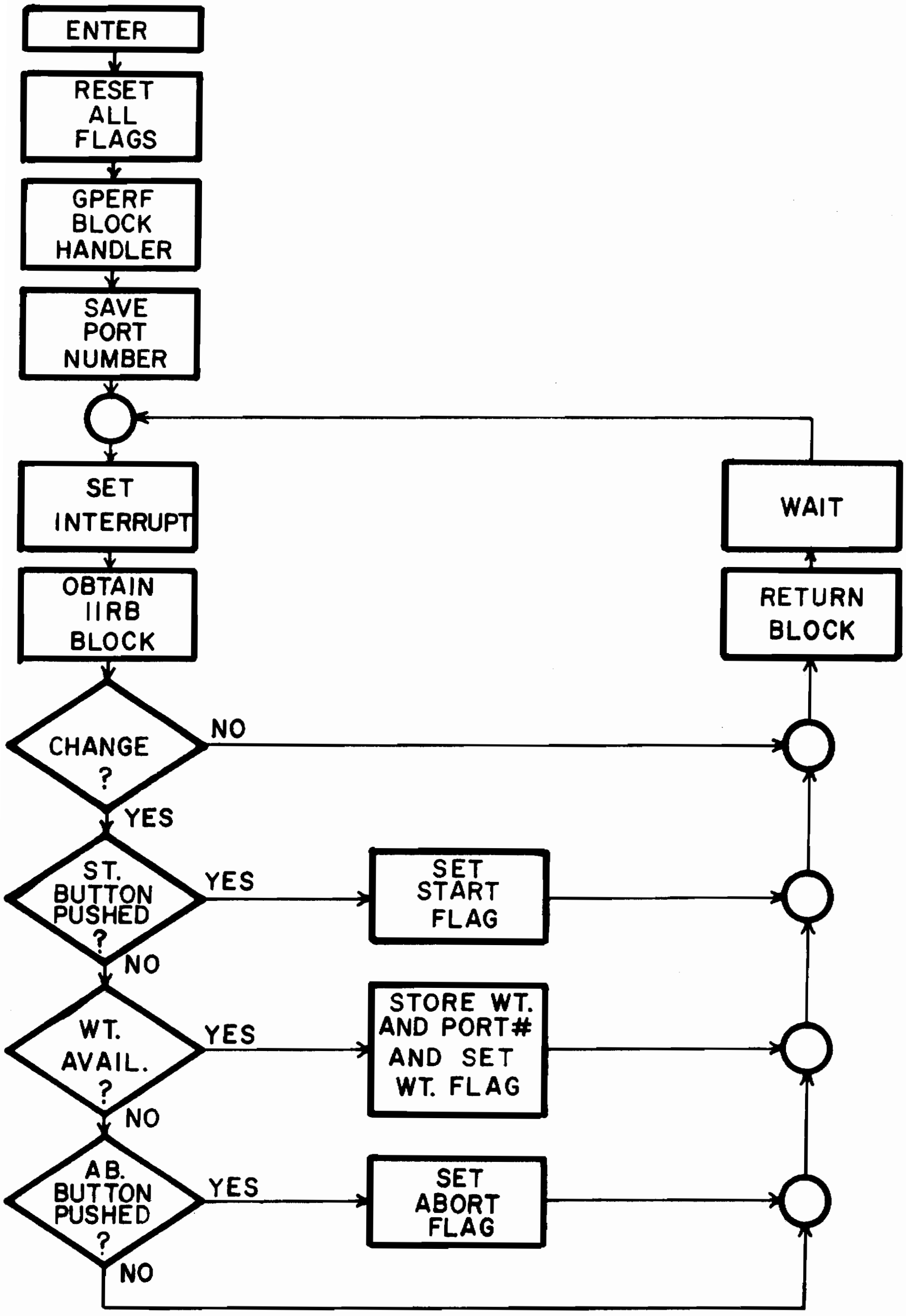

FIGURE 3.2.1 SPIN TASK FLOWCHART

to form new data blocks; and a system failure will occur.

The Spin Task Routine (see Fig. 3.2.1) is responsible for receiving these data blocks from an autobalance port, evaluating their contents, and returning them to the mux software for reuse. In evaluating these data blocks, the Spin Task is able to monitor the "start" and "abort" switches of the II box and receive any filter weight when available.

The status of the II control switches is part of the information received by the multiplexer and passed in a data block. When the block indicates that either the "start" or "abort" switch has been pushed, the Spin Task will set the Start or Abort Flag and return the data block by placing a minus one (-1) in the first location of the block. When a weight from the autobalance is present in the data block, the Spin Task will store this, along with the port number of the autobalance, and set a flag (Weight Flag) indicating a new weight is present. After returning the data block, a software interrupt is set to trigger the next data block check cycle.

As one of its first duties, the Spin Task starts (GPERFS) the Block Handler Routine. This routine is explained in the next section.

3.3 Block Handler

Running asynchronously with the Spin Task, the Block Handler Routine (see Fig. 3.3.1) interfaces between the Spin Task and the Disk Routine and services the instrument interface boxes.

When initiated, the Block Handler will send a control word to the II box (through the mux software) to turn on the Ready light. It then checks the Start Flag to see if it is set (signifying the Start button

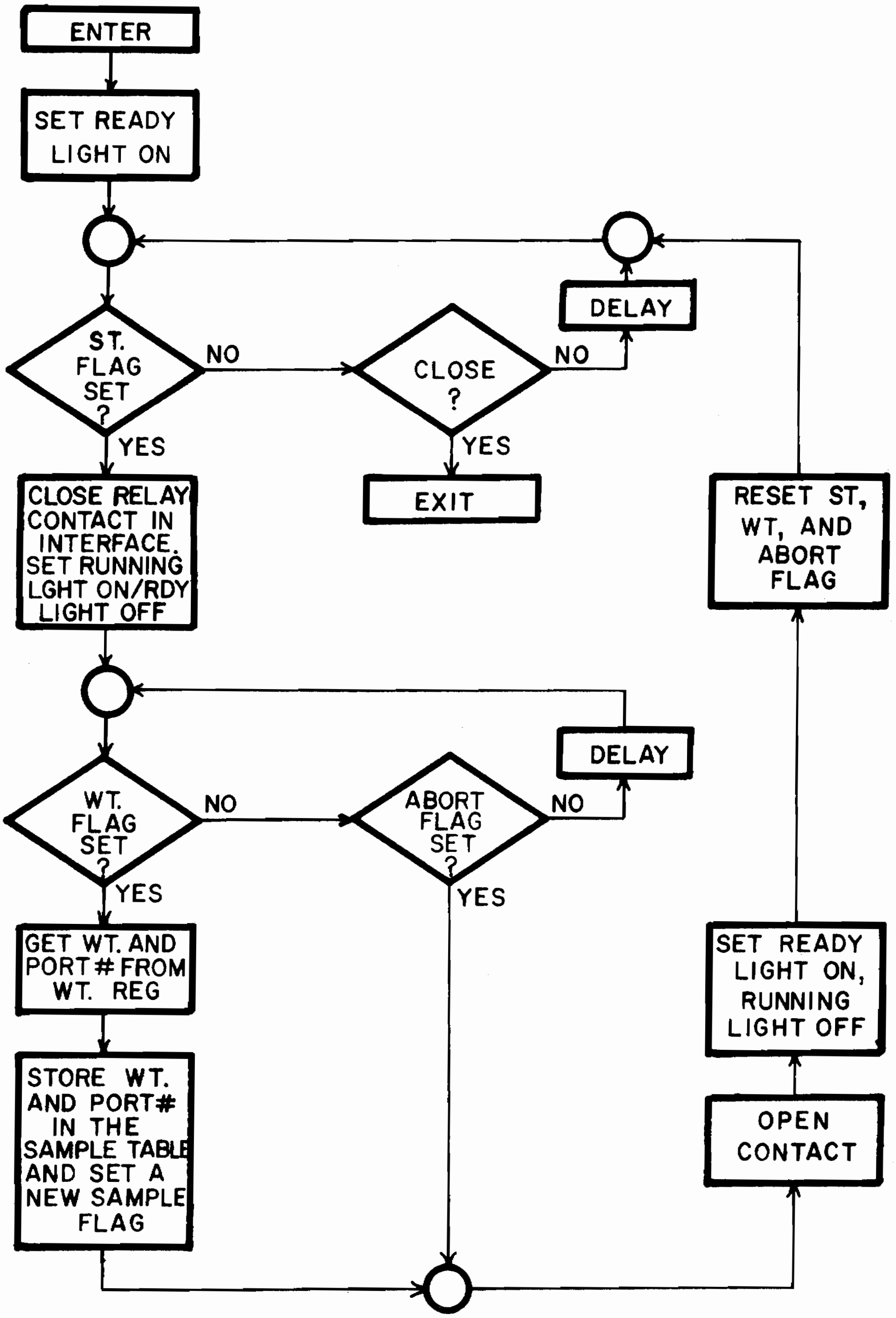
ENTER
SET READY LIGHT ON
ST. FLAG SET ?
NO
YES
CLOSE ?
NO
YES
DELAY
EXIT
CLOSE RELAY CONTACT IN INTERFACE. SET RUNNING LGHT ON/RDY LIGHT OFF
RESET ST, WT, AND ABORT FLAG
WT. FLAG SET ?
NO
YES
ABORT FLAG SET ?
NO
YES
DELAY
GET WT. AND PORT # FROM WT. REG
STORE WT. AND PORT# IN THE SAMPLE TABLE AND SET A NEW SAMPLE FLAG
SET READY LIGHT ON, RUNNING LIGHT OFF
OPEN CONTACT

FIGURE 3.3.1 BLOCK HANDLER FLOWCHART

on the II box has been pushed). If set, a control word will be sent to the II box to close the relay contacts and turn the Ready light off and the Running light on. Closing the II box relay sends a weight request to the autobalance. The Block Handler will then check the Weight Flag. If it is not set, the Abort Flag will be checked. If this is set, this particular weighing cycle will be aborted and the routine will return to the beginning for a new cycle. If the Abort Flag is not set, a delay will occur and the Weight Flag will be rechecked. This will continue until either the Weight Flag is set (filter weight available) or the Abort Flag is set (Abort button pushed.

When a weight becomes available, it is received from the Spin Task (through the common data area) and stored along with the port number in another data area in main memory designated the "Sample Table". The New Sample Flag for this particular location of the table is set. The Block Handler next opens the relay contacts in the II box, sets the Running light off and the Ready light on, and resets the Weight, Start, and Abort Flags; and a new weighing cycle begins.

3.4 Disk Routine

The Disk Routine (see Fig. 3.4.1) is the largest in the autobalance system and is responsible for all disk access operations.

During a clean filter weighing session, the routine continuously checks the Sample Table for new entries from each autobalance port. When one is available (New Sample Flag set), the weight is checked against predetermined limits (approximately 3.0 ± 0.5 grams). If the weight violates the limits, an error flag is set, the weight and any entered ID number for the weight are discarded, and the routine restarts. If the weight is valid, the routine checks to see if an ID number has been placed in the Sample Table by the command processor (ID Flag set). If not, an error flag is set, the weight is discarded, and the routine restarts. If an ID is available, the routine continues.

On the initial run, the data disk is accessed, the data file opened, and the sample directory is read into main memory. On each run, the directory is checked for a sample label for the particular ID number being processed. If one is not available, a label is created; and the defined sector is read into main memory. The sample record (weight and date) is placed in the sector data, and the data is read back into the sector. If a sector label already exists for the ID, the defined sector is read into core and the Weight Read Flag is checked. If it is set, a valid sample record already exists on the disk for the given ID number; and the Disk Routine checks to see if a correction (Correction Flag set) or quality control (Q.C. or Init Flags set) operation is desired. If not, an error flag is set, the ID and the specific New Sample Flags are reset, and the Disk Routine restarts. If a correction is desired, the current sample record replaces the old one; and the data is read back into the sector.

If the Q.C. and/or Init Flag is set, the duplicate sample records are placed in the Q.C. Table (in main memory); and a flag is set to indicate (to the Quality Control Routine) a new entry. In this case, the record on the data disk is not changed; and the Weight Read Flag

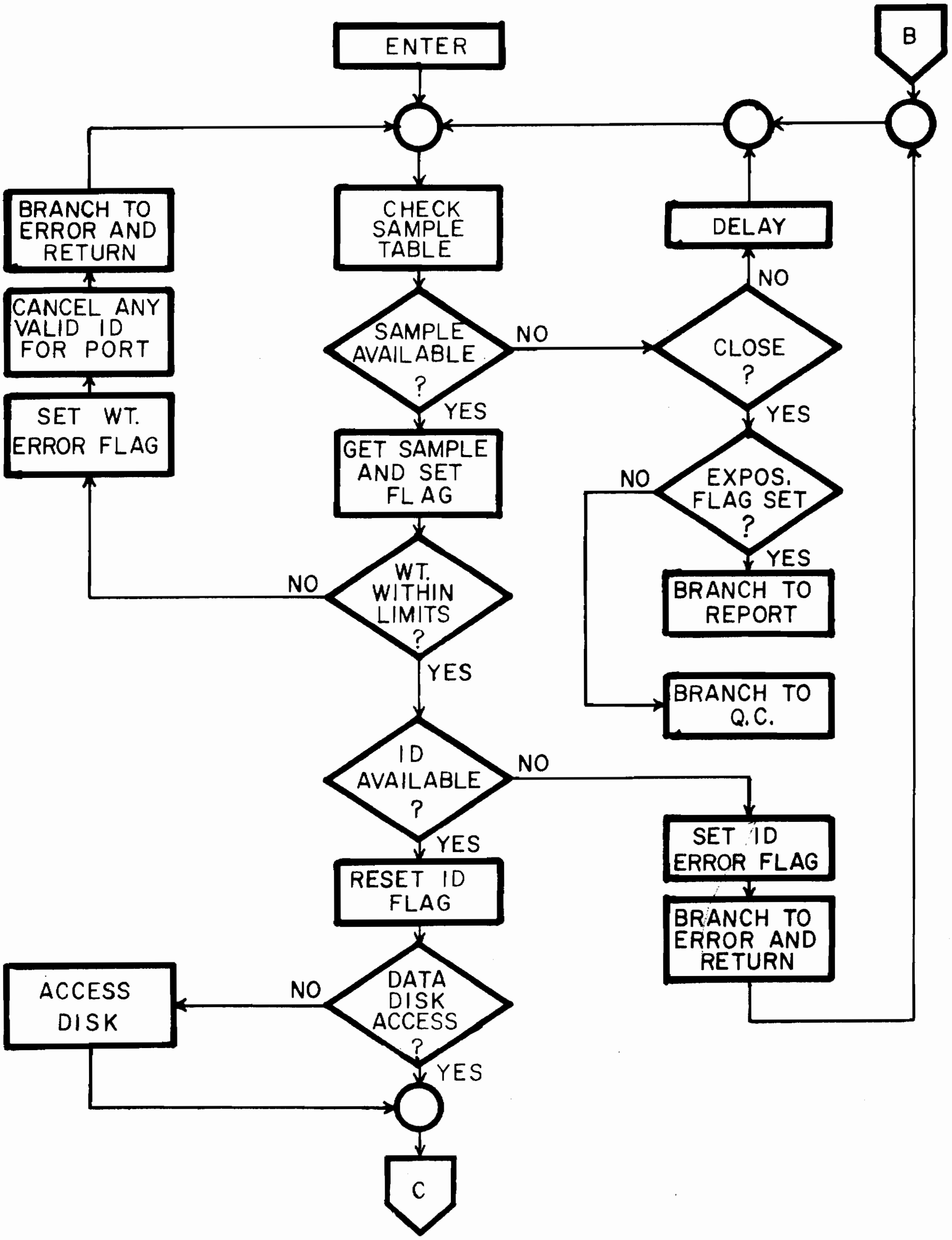

FIGURE 3.4.1 DISK ROUTINE FLOWCHART

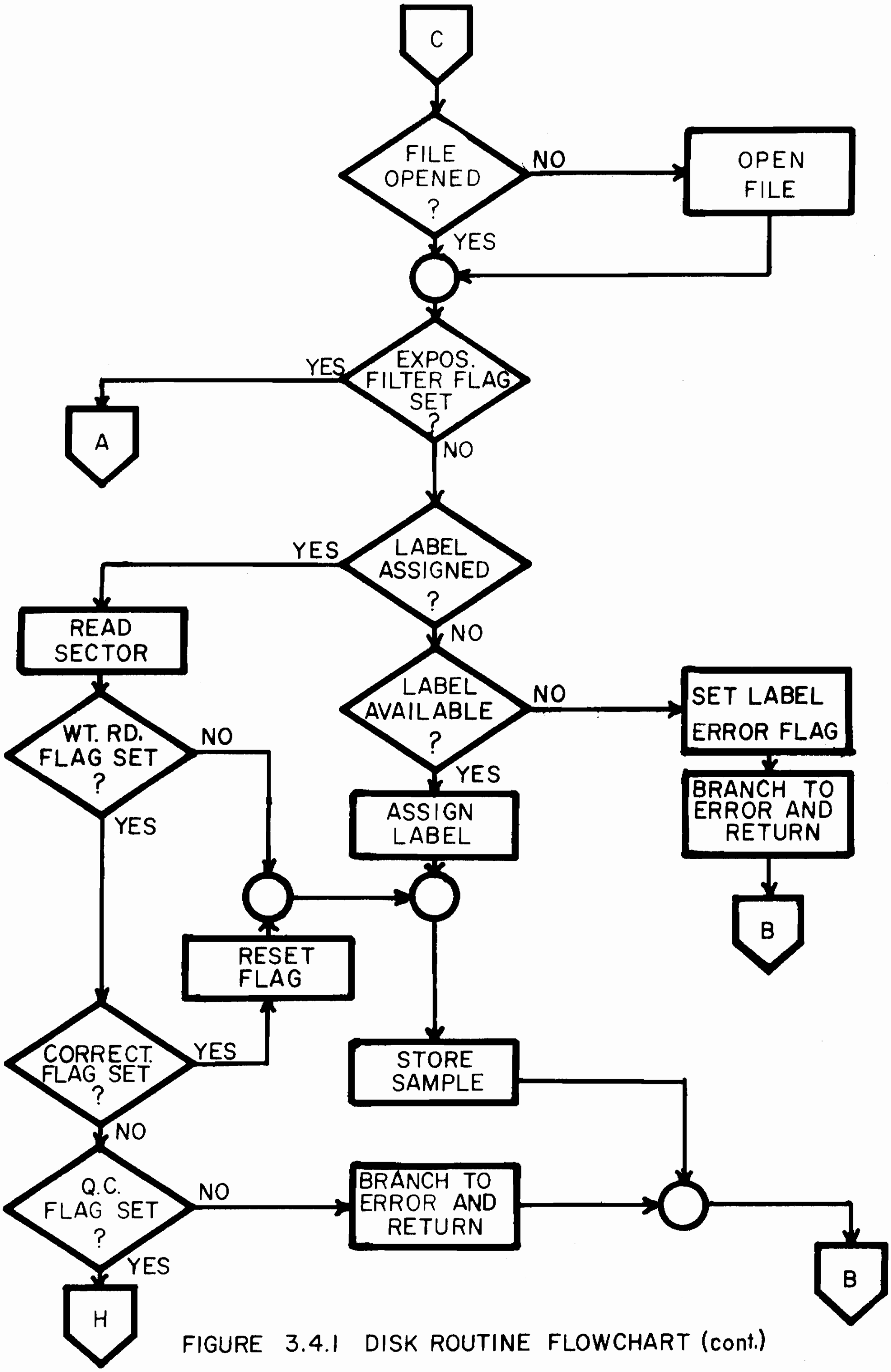

FIGURE 3.4.1 DISK ROUTINE FLOWCHART (cont.)

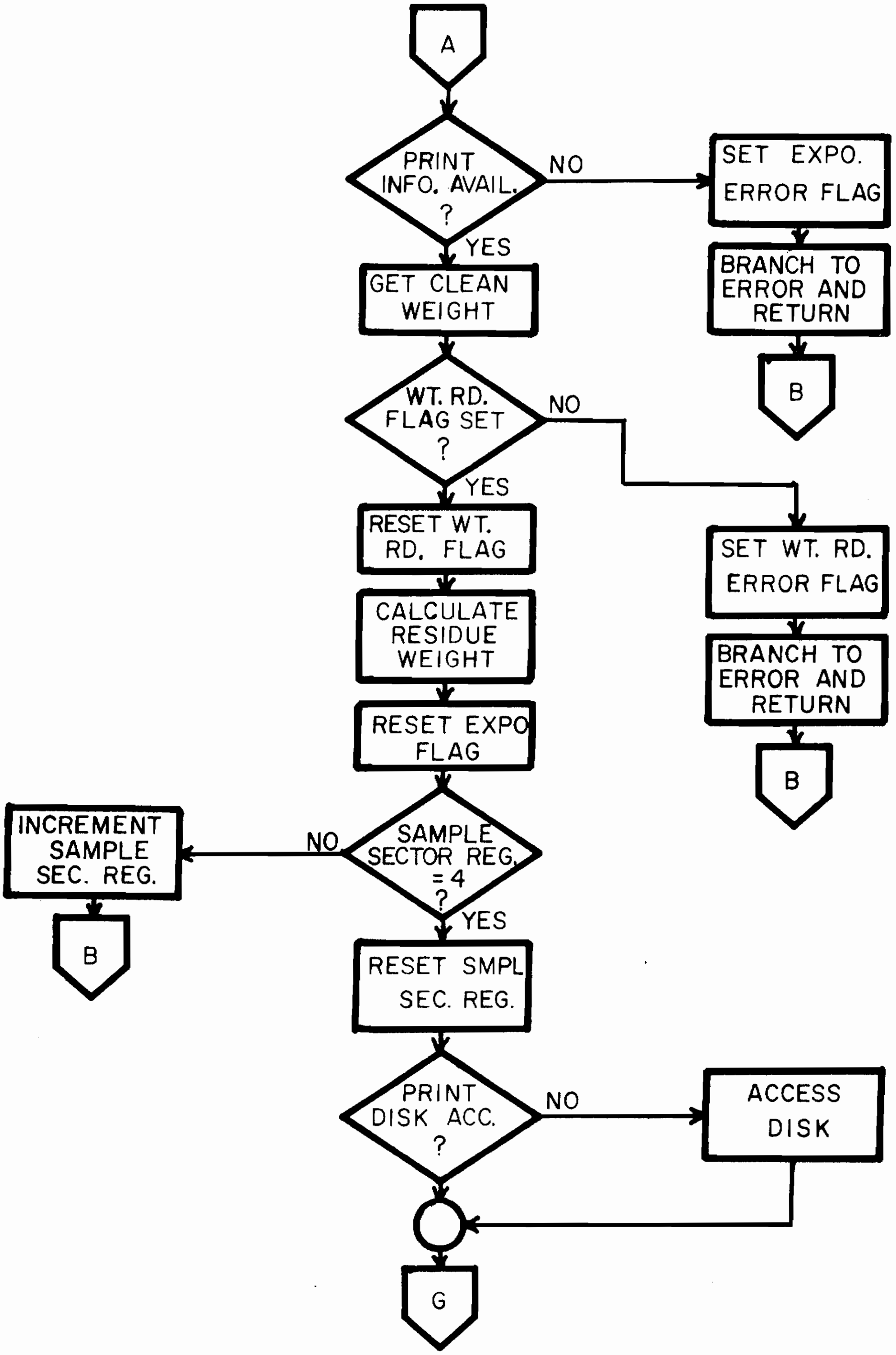

FIGURE 3.4.1 DISK ROUTINE (cont.)

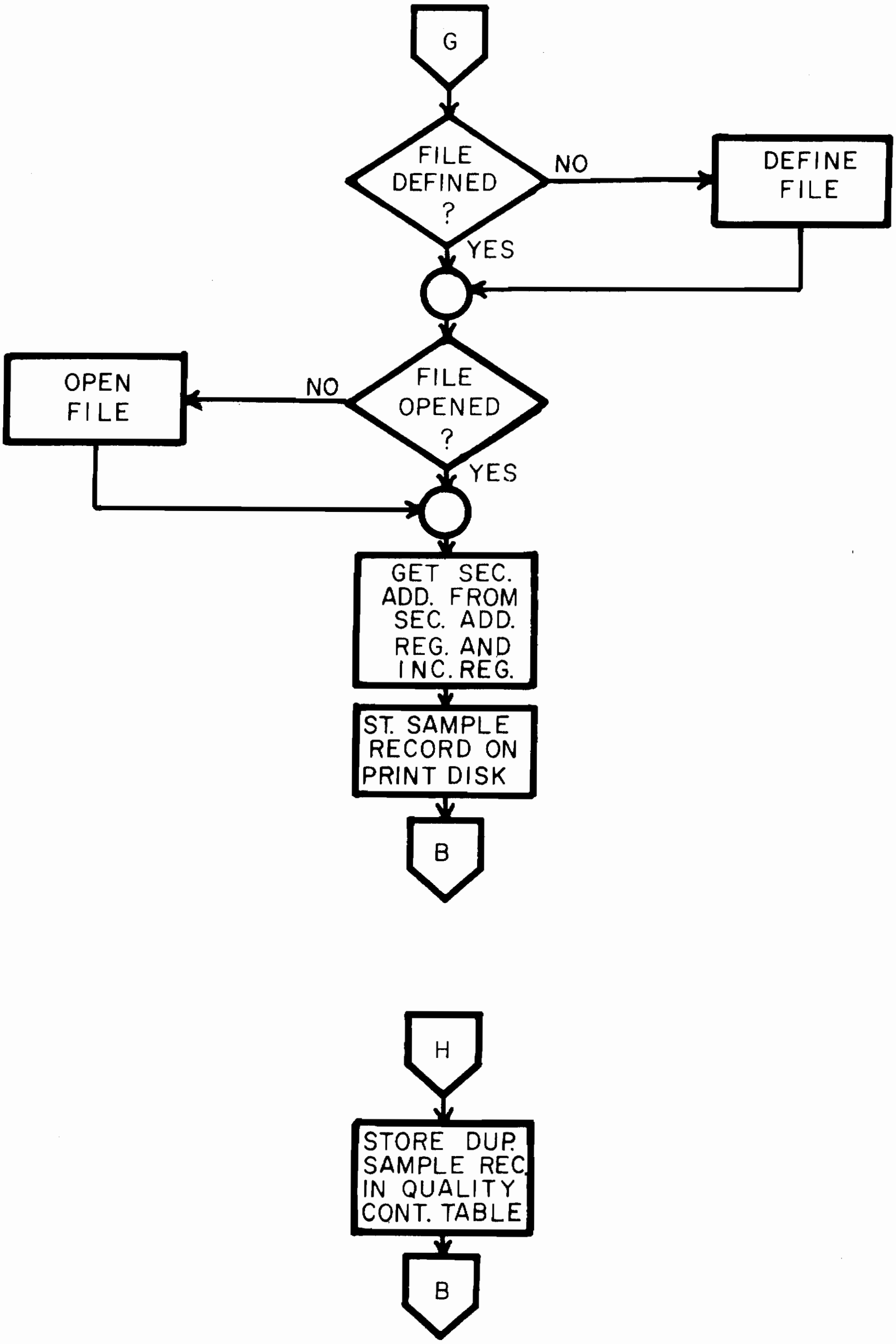
G
FILE DEFINED ?
NO
DEFINE FILE
YES
OPEN FILE
NO
FILE OPENED ?
YES
GET SEC. ADD. FROM SEC. ADD. REG. AND INC. REG.
ST. SAMPLE RECORD ON PRINT DISK
B
H
STORE DUP. SAMPLE REC. IN QUALITY CONT. TABLE
B

FIGURE 3.4.1 DISK ROUTINE FLOWCHART (cont.)

is not reset. If the Weight Read Flag is initially reset, the sample record is placed in the proper location of the sector data and the data read back into the sector. After the weight has been saved, the specific New Sample Flag in the Sample Table and the ID Flag are reset.

During the exposed filter weighing session, the procedures for obtaining and checking the weight and ID number, accessing the data disk, opening the data file, and reading the sample directory into main memory are the same as for the clean filter session. At this point, the Disk Routine checks for the extra printout material (station number, operator number, etc.) in the Sample Table. If this is not available, the sample record is discarded (appropriate flags reset), an error flag is set, and the routine restarts. Otherwise, the routine reads the appropriate clean filter weight from the data disk, calculates the residue weight, and stores this with the sample record. If a clean filter weight for the ID number does not exist on the data disk, the record is discarded, an error flag is set, and the routine restarts. Once the exposed filter sample record is complete, the Disk Routine accesses the print disk and defines and opens a file corresponding to the specific autobalance port number (if this is not already done). The sample record is then placed into a sector data block. When the block contains four sample records, it is read into a sector on the print disk.

After each disk operation, the Disk Routine checks for any disk errors. This is done by checking the error code word passed back to the routine by DFMP. If an error exists, the Disk Routine passes the error number and indicator number (indicates where in the program the error occurred) to the print routine.

After all weighing operations are complete and all sample records are stored, the Disk Routine branches to either the Quality Control Routine (after clean filter weighing session) or the Print Routine (after an exposed filter weighing session).

3.5 Print Routine

All information printed by the terminal for the autobalance system is handled by the Print Routine. This is actually three separate routines; the Error Writer, the Report Writer, and the Dump Routine (see Fig. 3.5.1).

The main section of the Print Routine is the Report Writer. This is branched to by the Disk Routine at the close of an exposed filter weighing session. The Report Writer is passed the address of the last sector stored on the print disk by the Disk Routine for the specific port being serviced. A report heading is printed at the terminal, and a printout of all sample records stored on the print disk (from the specific port) is executed (uninterrupted) by the Report Writer. An "end-of-record" statement signifies that all sample records for a given port have been printed. The Report Writer then branches to the Quality Control Routine.

The Error Writer is branched to by the Disk Routine whenever an error occurs (see Table 3.5.1). For all but disk errors, the Error Writer will print an error message corresponding to the particular

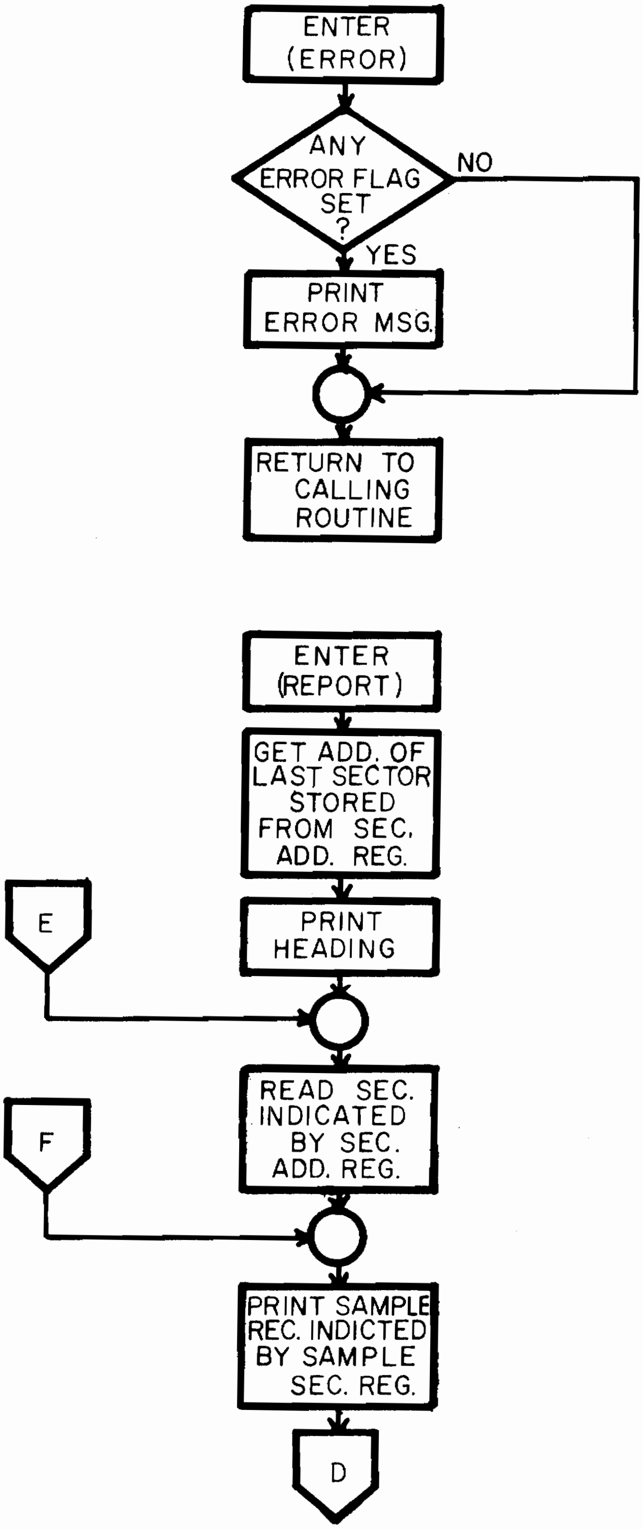
ENTER (ERROR)
ANY ERROR FLAG SET ?
NO
YES
PRINT ERROR MSG.
RETURN TO CALLING ROUTINE
ENTER (REPORT)
GET ADD. OF LAST SECTOR STORED FROM SEC. ADD. REG.
E
PRINT HEADING
READ SEC. INDICATED BY SEC. ADD. REG.
F
PRINT SAMPLE REC. INDICTED BY SAMPLE SEC. REG.
D

FIGURE 3.5.1 PRINT ROUTINE FLOWCHART

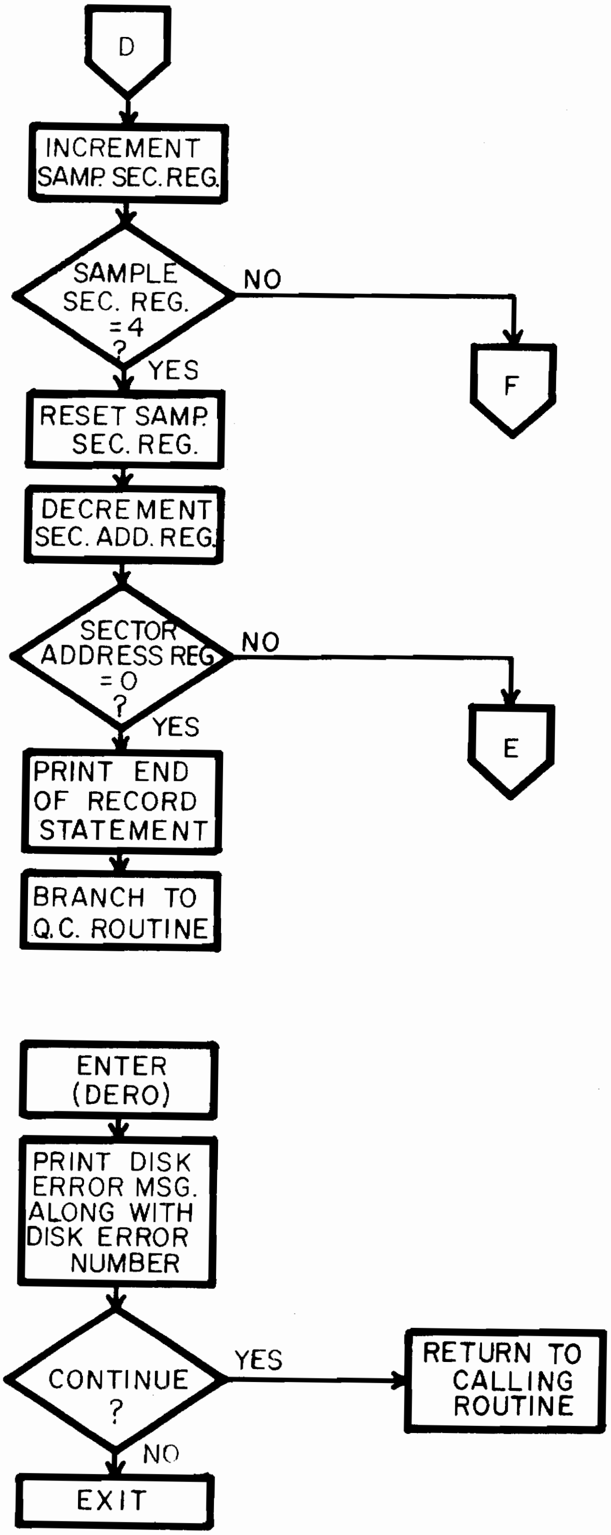

FIGURE 3.5.1 PRINT ROUTINE FLOWCHART (cont.)

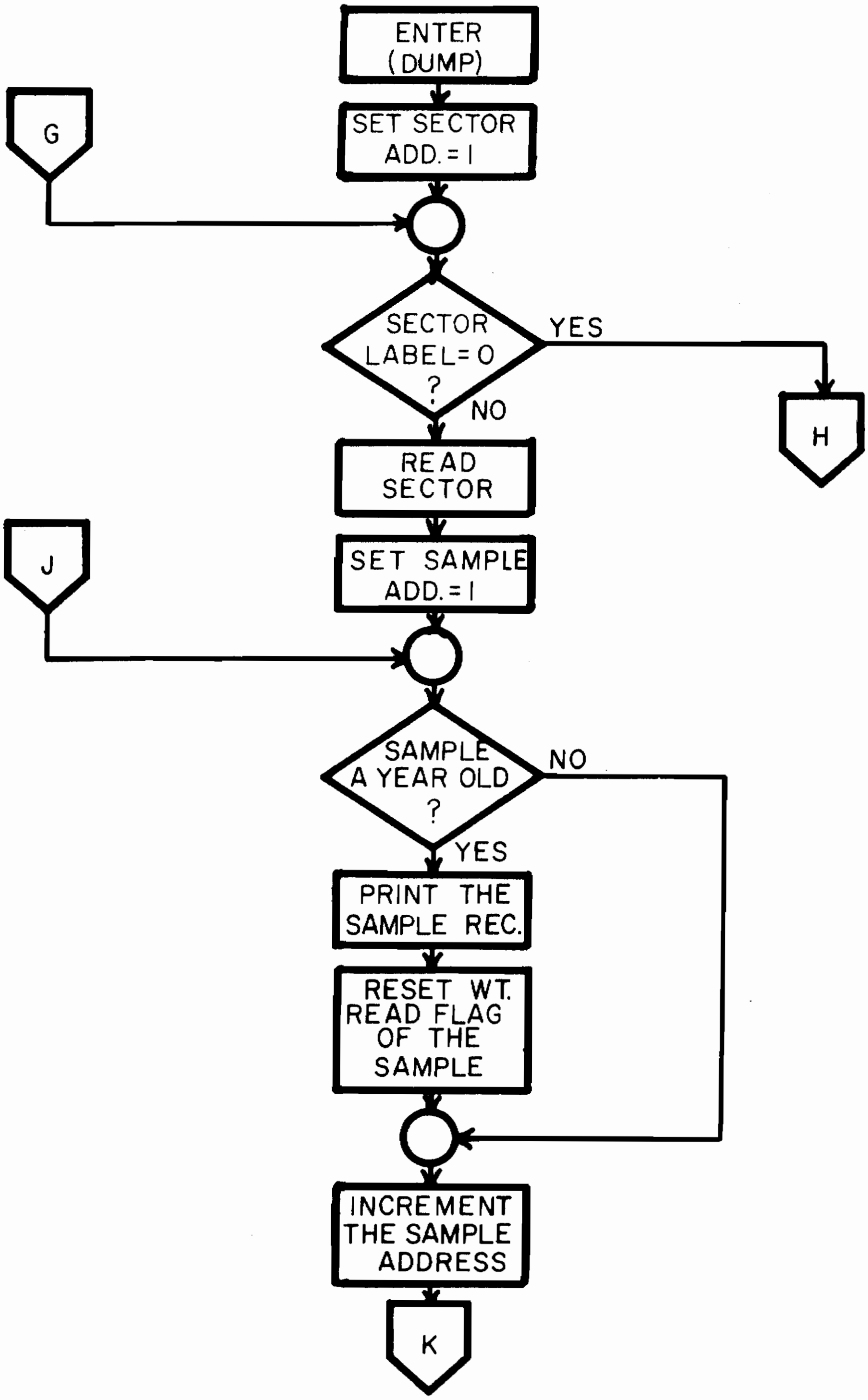

FIGURE 3.5.1 PRINT ROUTINE (cont.)

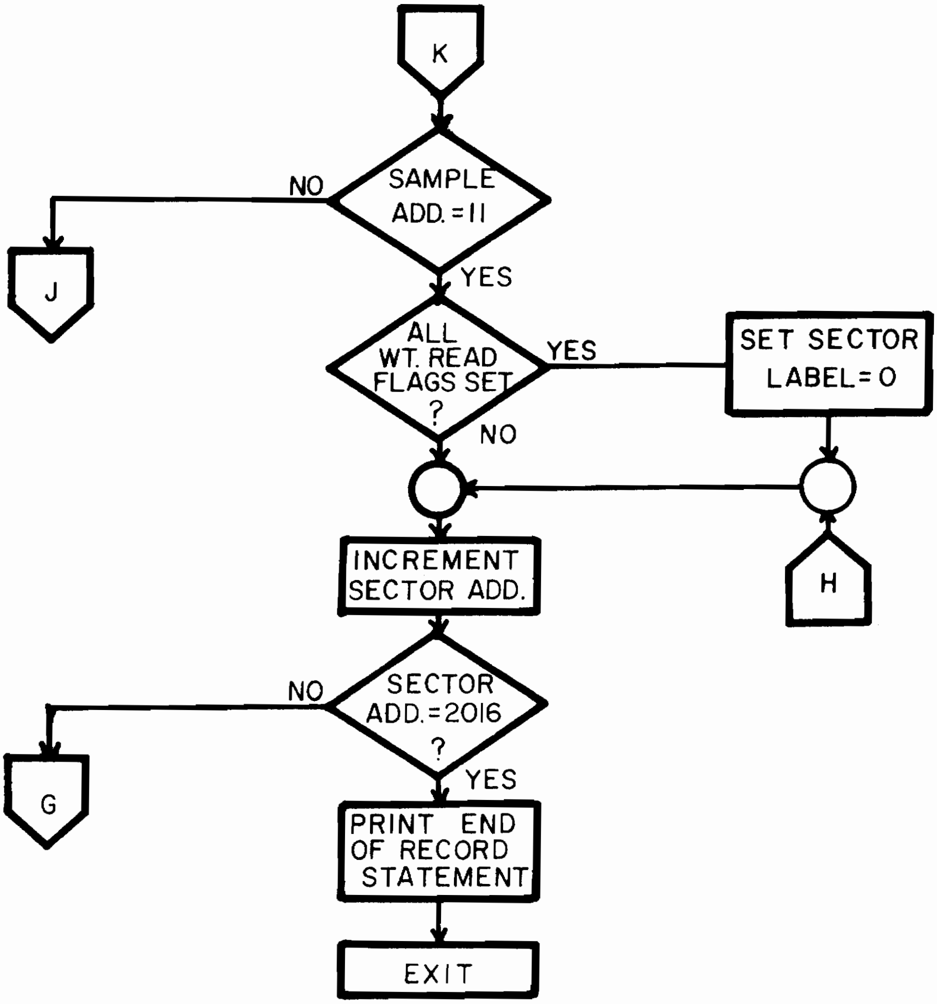

FIGURE 3.5.1 PRINT ROUTINE (cont)

Table 3.5.1 Error Messages

	ERROR MESSAGES		ERROR
1)	Invalid Weight	1)	Weight not within preset limits
2)	ID Not Entered	2)	ID number not entered before pushing START button
3)	Duplicate ID	3)	Duplicate ID number entered and Q.C. operation or correction not intended
4)	Extra Information Not Entered	4)	Extra printout information not entered before pushing START button (Exposed filter weighing session)
5)	Clean Weight Not Available	5)	Clean filter weight not found on data disk
6)	Data Disk Full	6)	No space available on data for sample record storage
7)	Print Disk Full	7)	No space available on print disk

error flag set, reset the error flag, and return to the Disk Routine. When disk errors occur, an error message is printed along with a description of the error location as specified by the indicator number passed by the Disk Routine. The user is then asked if he wishes to continue with the present disk. Depending on the answer, the routine will either clear the disk error word and return to the Disk Routine or it will end.

The Dump Routine is initiated by a command from the user and is usually executed during off hours. It accesses the data disk, opens the data file, and reads the sample directory into core. It then checks each sector label. If the label is zero, it moves to the next. If the label is not zero, the specified sector is read into core. The date of the initial clean filter weighing of the ten sample records from the sector are checked. Any sample records over a year old are printed at the terminal, and the specified Weight Read Flag is reset. The Dump Routine then checks the ten Weight Read Flags. If all are reset, a zero is placed in the sector label. If some are not reset, the data is read back into the sector. This is continued until all sectors are checked. The file is then closed, and the disk is detached.

3.6 Quality Control Routine

The Quality Control Routine (see Fig. 3.6.1) is branched to from the Disk Routine, at the end of clean filter sessions, or from the Print Routine, at the end of exposed filter sessions. Three routine operations are available, depending on the routine flag set. These are the quality control (Q.C.), the initialization (Init.), and the

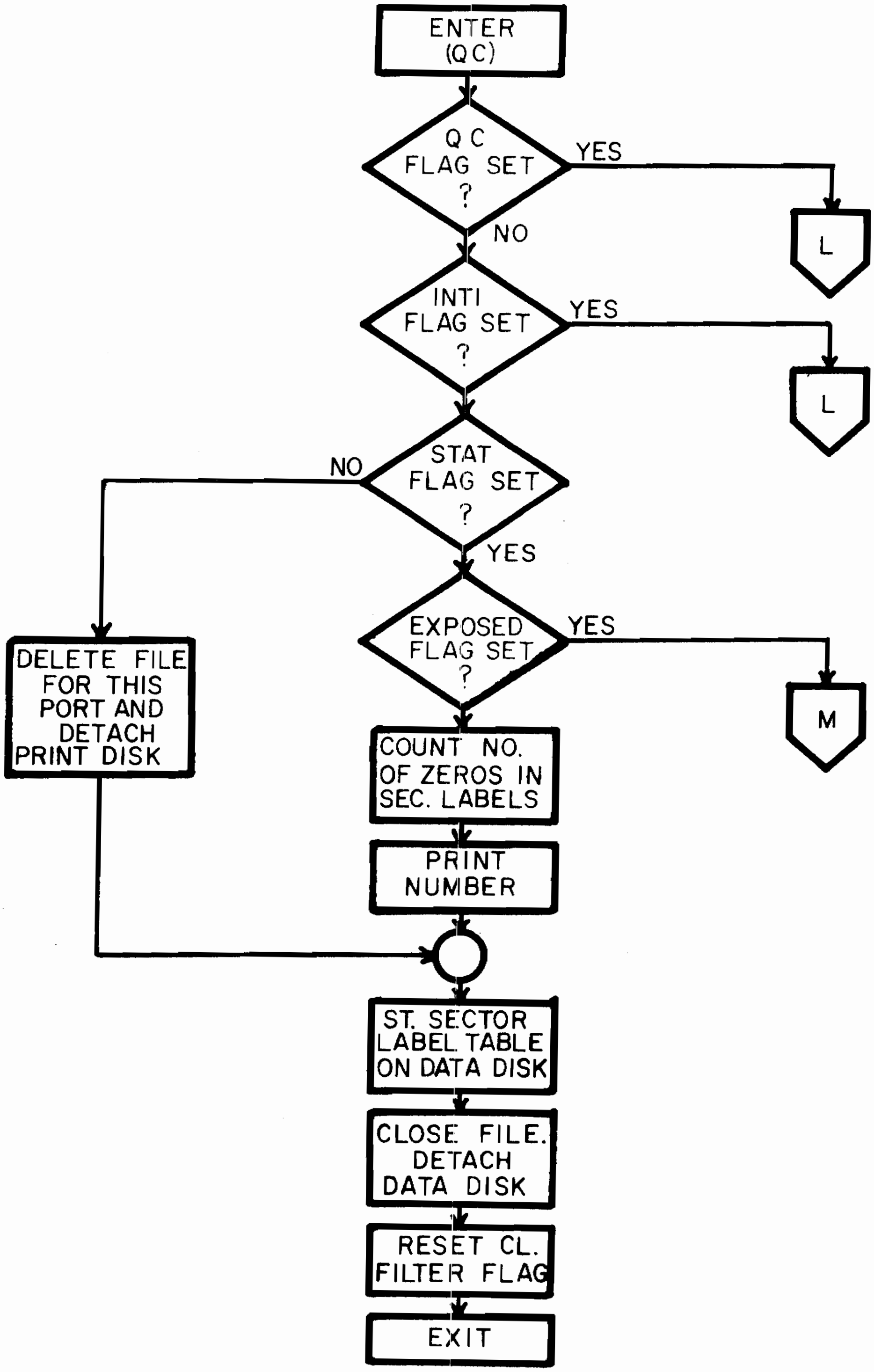
ENTER (QC)
QC FLAG SET ?
YES
NO
L
INTI FLAG SET ?
YES
L
STAT FLAG SET ?
NO
YES
EXPOSED FLAG SET ?
YES
M
DELETE FILE FOR THIS PORT AND DETACH PRINT DISK
COUNT NO. OF ZEROS IN SEC. LABELS
PRINT NUMBER
ST. SECTOR LABEL TABLE ON DATA DISK
CLOSE FILE. DETACH DATA DISK
RESET CL. FILTER FLAG
EXIT

FIGURE 3.6.1 QUALITY CONTROL ROUTINE FLOWCHART

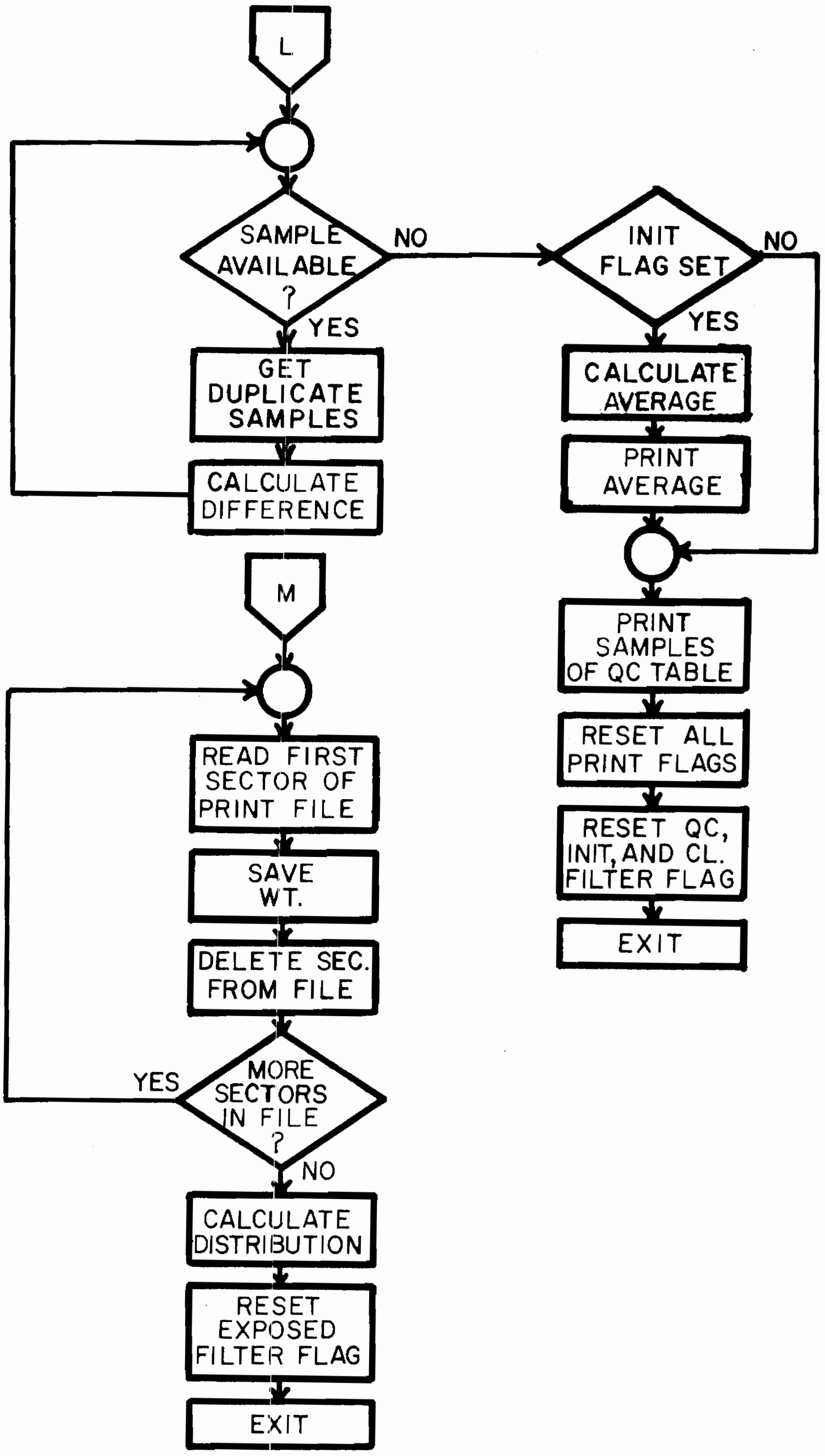

FIGURE 3.6.1 QUALITY CONTROL ROUTINE FLOWCHART (cont)

statistical (Stat.) operations. If no routine flags are set, the print file is deleted; and the print disk is detached (if applicable). The sample directory is stored on the data disk, the data file is closed, and the data disk is detached.

The quality control operations occur when the Q.C. flag is set and include calculating the difference in weight of duplicate sample records in the Q.C. Table. These weight differences, along with the ID number, are printed at the terminal.

The initialization operations (Init flag set) are the same as the quality control operations, except the average of all the weight differences is also calculated and printed.

The statistical operations (Stat flag set) include determining the number of free sectors (zeros in sample directory) on the data disk. Also, if the routine was called at the end of an exposed filter weighing session, the distribution by range of the sample residue weights of all samples from a specific port on the print disk is calculated and printed at the terminal. The print file is then deleted, and the print disk detached. The sample directory is stored on the data disk, the data file is closed, and the data disk is detached.

3.7 Format Routine

The Format Routine (see Fig. 3.7.1) is a stand alone, disk resident routine. Its main purpose is to prepare the data disk for the Disk Routine. The disk must first be formatted with the standard DSOS formatter. Two thousand and eighty sectors (2080) are then assigned consecutive number addresses (1-2080). This is done ten sectors at a

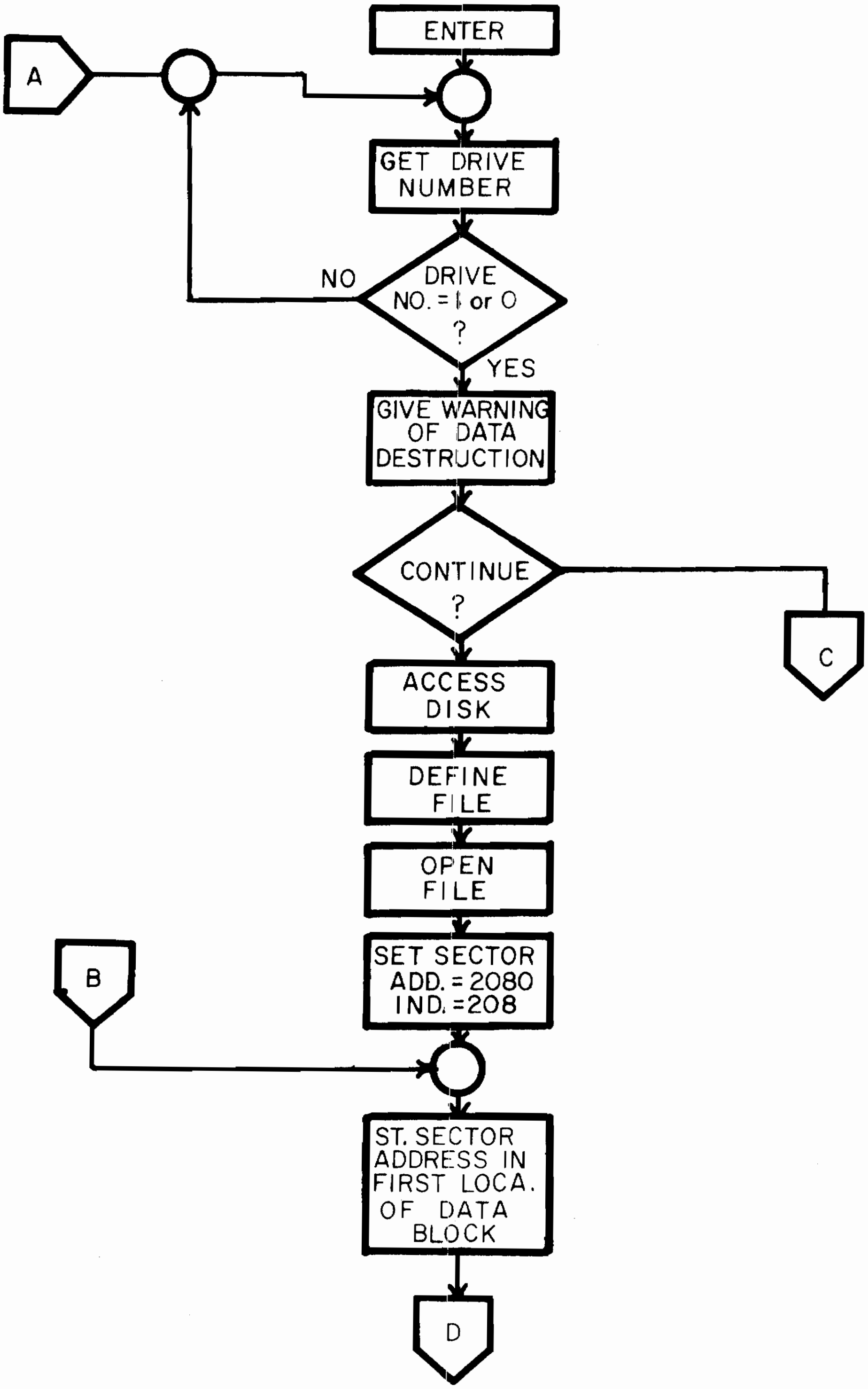
ENTER
A
GET DRIVE NUMBER
NO
DRIVE NO. = 1 or 0 ?
YES
GIVE WARNING OF DATA DESTRUCTION
CONTINUE ?
C
ACCESS DISK
DEFINE FILE
OPEN FILE
B
SET SECTOR ADD. = 2080 IND. =208
ST. SECTOR ADDRESS IN FIRST LOCA. OF DATA BLOCK
D

FIGURE 3.7.1 FORMAT ROUTINE FLOWCHART

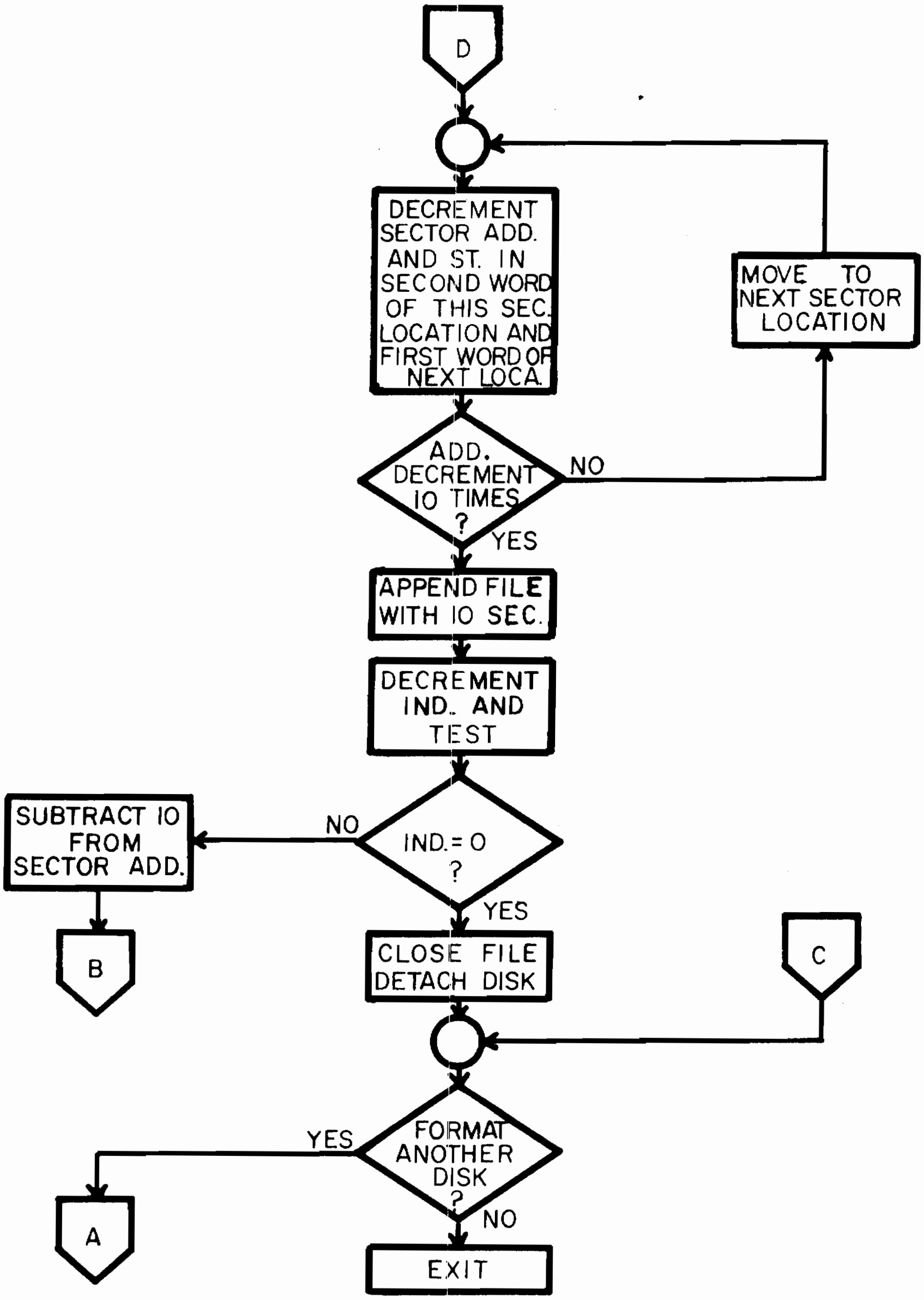
D
DECREMENT SECTOR ADD. AND ST. IN SECOND WORD OF THIS SEC. LOCATION AND FIRST WORD OF NEXT LOCA.
ADD. DECREMENT IO TIMES ?
NO
MOVE TO NEXT SECTOR LOCATION
YES
APPEND FILE WITH IO SEC.
DECREMENT IND. AND TEST
IND.= 0 ?
NO
SUBTRACT IO FROM SECTOR ADD.
B
YES
CLOSE FILE DETACH DISK
C
FORMAT ANOTHER DISK ?
YES
A
NO
EXIT

FIGURE 3.7.1 FORMAT ROUTINE FLOWCHART (cont)

time. A counter is set to 208 and a sector data area is established with ten data locations. The first word in the first location in initialized to 2080. The number is decremented (2079) and stored in the second word of the first location and the first word of the second location. This is continued until the first two words of all ten locations are loaded. The sector data area is then stored on the disk (in ten sectors), and the counter is decremented. The first word in the first location is then reinitialized to 2070. This process is continued until the counter is zero. Since the first word of each sector is its address indicator, the 2080 sectors now have addresses from 1 - 2080.

3.8 Summary

In this chapter, the basic data management techniques were discussed. The six autobalance system service routines were also explained in detail. The general flow of the service routines is controlled by commands entered at a terminal by the operator. These commands are explained and typical weighing sessions (command sequences) are presented in the next chapter.

CHAPTER IV

Autobalance System Operation

4.1 Commands

Communication between the autobalance system routines and the operator is carried out by means of a set of autobalance commands typed on a terminal by the operator. The commands initiate the entire autobalance system and dictate its functional flow. Commands perform various functions, including data entry.

The commands are processed by the system Command Processor, which is called by pushing the escape (interrupt, break, etc.) key of the terminal. When called, the Command Processor will respond with a prompt for a command by displaying or printing the message "comand?". The Command Processor then waits for a terminal entry by the operator. When data is entered by means of a command, the Command Processor will pass it to the system only if it is in the format predescribed by the system programmer.

The autobalance commands are organized into three command levels (see Fig. 4.1.1). The Command in the first level enables the acceptance of the commands in the second level. Two of the second level commands enable the acceptance of specific commands in the third level. In this way, commands are accepted in the proper order only.

The system initiator is the AUTOBL command. The format for the command is AUTOBL <N>, where N is the number of one of the sixteen multiplexer ports through which the desired autobalance is connected. The AUTOBL command opens the port and initiates the Spin Task. The

AUTOBL command also enables the acceptance of the clean filter (CLFILT), exposed filter (DRFILT), and the dump (CKBLWT) commands.

The clean filter command (CLFILT) is used to initiate a clean filter weighing session by setting the Clean Filter Flag (CFF). The command initiates the Disk Routine and allows the clean filter command chain in the third level (see Fig. 4.1.1).

The exposed filter command (DTFILT) is used to initiate an exposed filter weighing session by setting the Exposed Filter Flag (EFF). The command also initiates the Disk Routine and allows the exposed filter command chain in the third level (see Fig. 4.1.1).

The identification command, ID <L⌀NNNNNN> , is used to enter a filter identification number, where L⌀NNNNNN is the identification number. The format of the identification number is letter (L) followed by a blank (⌀) and a one-to-six decimal digit number (N). The number must include at least a letter and one decimal digit separated by one (or more) blanks (i.e., A1 or B 654326). Upon execution of the ID command, the ID Flag (IDF) is set and the autobalance system will not allow another ID to be entered into the Sample Table until the previous entry is processed. Thus, if the ID Flag is set, the Command Processor will wait until it is reset (signifying the ID has been used) before permitting a second entry. When it is reset, the Command Processor will store the ID number and set the ID Flag.

The correction command, CR <L⌀NNNNNN>, is similar to the ID command, except that the Correction Flag (CRF) is also set. This command is used to indicate to the Disk Routine that the specific ID number is

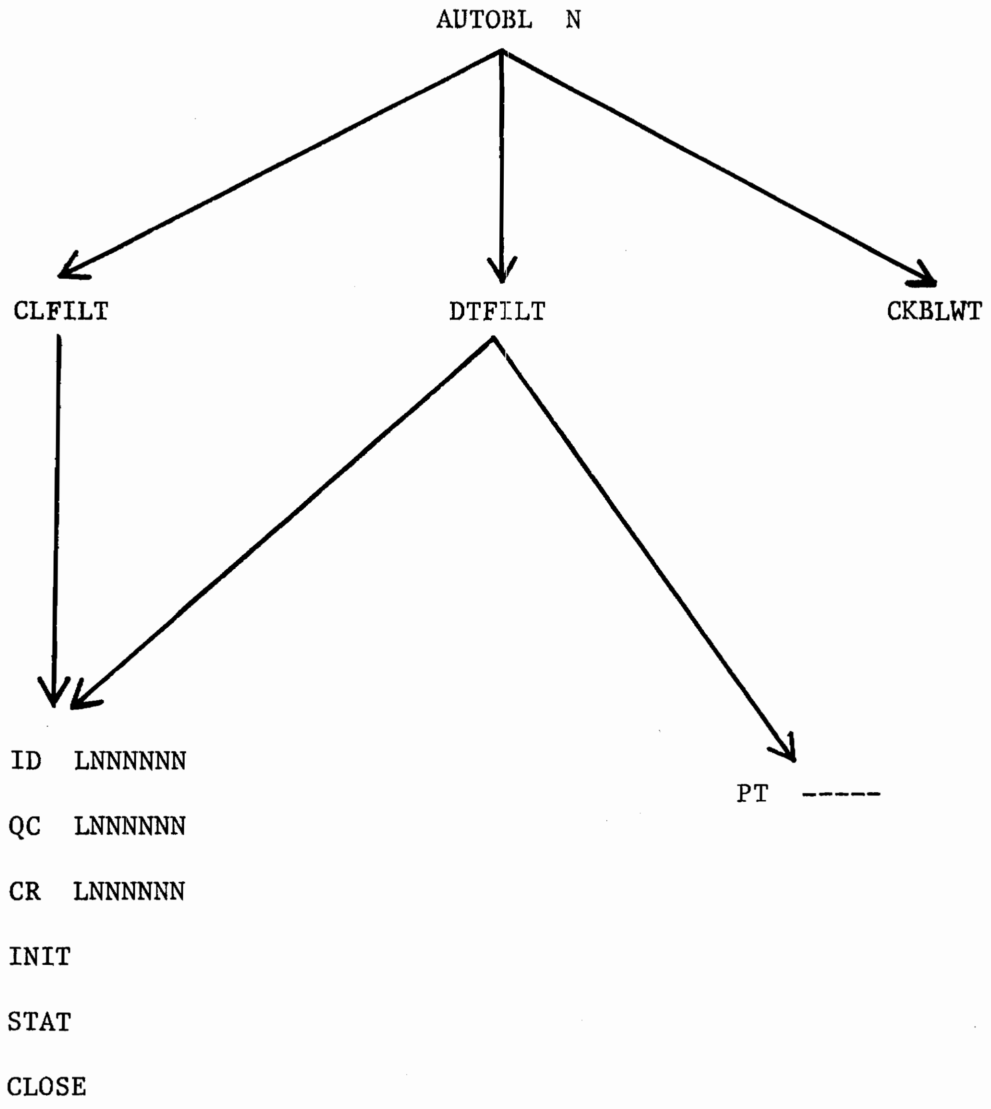

Figure 4.1.1 Command Levels

a duplicate of one already processed and that a correction of the corresponding sample record is desired. If a duplicate ID number is not on the data disk, an error flag is set.

The quality control command, QC <Lb̸NNNNNN>, is similar to the ID command, except that the Q.C. Flag (QCF) is also set. This command indicates to the Disk Routine that the specified ID number is a duplicate and two sample records (the current one and the one on the data disk) should be passed to the Q.C. Routine. If a duplicate ID number is not on the data disk, an error flag is set.

The printout command, PT <----->, is used to enter the extra information that should be stored for later printing with each final sample record. The information to be entered is as follows: time, temperature (when filter was exposed), flow rate of air through filter at time of exposure, experimental station number, and operator. The information will be entered as variable length tokens separated by one or more blanks. This information will be stored in the Sample Table if the Extra Printout (EXPO) Flag (EXF) is reset. If not, the Command Processor will wait until the flag is reset, store the information, and set the Expo Flag.

The initialization and statistical commands, INIT and STAT, indicate to the Quality Control Routine to conduct the initialization operations or the statistical operations, respectively, by setting the Init (IF) or the Stat (SF) Flag.

The CLOSE command is used to end a weighing session by setting the Close Flag (CLF). This terminates the Disk Routine and causes the execution of the Print Routine and the Q.C. Routine.

The dump command, CKBLWT, is a "stand alone" command that will usually be issued during off hours. The command initiates the Dump Routine. This conducts a lengthy operation of checking the date of all sample records on file and purging from the file those over a year old (see Section 3.5).

4.2 Typical Weighing Sessions

Tables 4.2.1 and 4.2.2 show the command flow of typical weighing sessions. It should be noted that the Quality Control Routine commands (Q.C., INIT, STAT) may or may not be executed during any given weighing session.

The output (printout at terminal) of the session always comes after the CLOSE command has been issued and depends on the type of weighing session as well as the commands issued during that session.

During a clean filter weighing session, if no quality control commands (QC, INIT, STAT) are issued, there will be no output. If the QC command has been issued, the output will include the ID number, the two filter weights, and the difference between these two weights for every sample entered with the command. The INIT command will give the same output for every duplicate ID entered after the command (not including those entered with the Q.C. or CR commands) and will also include the average of all the weight differences. The STAT command will give a printout of the number of free sectors on the data disk.

After an exposed filter weighing session (if no Quality Control Routine commands have been issued), the output will include the ID

Table 4.2.1 Typical Clean Filter Command Sequence

```
   i)  command ?              (Prompt from Command Processor)
       AUTOBL 1               (Autobalance of port 1 will be used)

  ii)  command ?
       CLFILT                 (Clean filter weighing session)

 iii)  command ?
       ID A 123456            (Filter ID number entered)

  iv)  command ?
       ID A 123457

   v)  command ?
       CR A 123457            (Filter weight corrected)
           .
           .
           .
           .
           .
           .
           .
   x)  command ?
       QC A 123456            (Quality control check)

  xi)  command ?
       STAT                   (Statistical command)

XII)  command ?
       CLOSE                  (Session closed)
```

Table 4.2.2 Typical Exposed Filter Command Sequence

```
  i)  command ?                 (Prompt from Command Processor)
      AUTOBL 1                  (Autobalance of port 1 will be used)

 ii)  command ?
      DTFILT                    (Exposed filter weighing session)

iii)  command ?
      ID A 123456               (Filter ID number entered)

 iv)  command ?
      PT -----                  (Extra printout information entered)

  v)  command ?
      ID A 123457

 vi)  command ?
      PT -----
         .
         .
         .
         .
         .
         .
         .
  x)  command ?
      CLOSE                     (Session closed)
```

number, the clean weight, the exposed weight, the residue weight, and the extra printout information for every sample entered during that session. The Q.C. and INIT commands will give the same additional information as during a clean filter weighing session. The STAT command will give the number of free sectors available on the data disk and also the distribution by range of the residue weights determined during the session.

4.3 Summary

In this chapter, the autobalance system commands were explained and the flow of these commands during typical weighing sessions was shown. Typical session output was also discussed.

It is suggested that, at the end of each day, the data disk be copied onto a backup disk. This will prevent the possible loss of sample records in the event the data disk is damaged or inadvertently erased.

CHAPTER V

Summary and Conclusion

5.1 Summary

The autobalance automation and data management system presented in this thesis provides for the collection and reduction of data from an electronic autobalance. The system utilizes the standard Remote Instrument Service Unit (RISU) computer system developed for the Division of Consolidated Laboratory Services (DCLS) of the Commonwealth of Virginia by the Computer Engineering Laboratory of VPI&SU. The autobalance is connected to the RISU through a special interface circuit in one of the instrument interface units developed as part of the laboratory automation and data management system. System software runs under the Deep Six Operating System (DSOS) and consists of six service routines working in an asynchronous environment. The routines are as follows: the Spin Task, the Block Handler, the Disk Routine, the Print Routine, the Quality Control Routine, and the Disk Formatter. The routines are controlled by a set of commands handled by a command processor.

At the writing of this thesis, the Disk Formatter has been implemented. The Spin Task, Block Handler, Disk Routine, and Print Routine have been coded and partially tested; and a skeletal outline has been written for the Quality Control Routine. The code for these routines is on file in the Computer Engineering Laboratory at VPI&SU.

5.2 Recommendation for Future Action

Immediate action should be taken to code and test the Quality Control Routine. Next, the entire system should be tested with all routines activated. The system should then be tested with one or more autobalances (or autobalance simulators) on line. The final phase would be the development of appropriate operation maintenance documentation.

BIBLIOGRAPHY

1. "Autobalance Interface Circuit", Computer Engineering Laboratory Internal Document, Virginia Polytechnic Institute and State University, 1977.

2. Bennett, A.W. and B.D. Osecky, "A Laboratory Automation and Data Management System for the Division of Consolidated Laboratory Services of the Commonwealth of Virginia", Project Proposal, Virginia Polytechnic Institute and State University, 1975.

3. Frailey, D.J., "DSOS - A Skeletal, Real-Time, Minicomputer Operating System", Software-Practice and Experience, Vol. 5, pp. 5-18, February 1974.

4. Merkley, Clifton D., "Laboratory Automation - Key to Profitable Processing", Quality Progress, pp 12-14, July 1976.

5. "Model 980 Computer Assembly Language Programmers Reference Manual", Publication Number 943013-9701, Texas Instruments, Inc., pp. 2.1-2.4.

6. Perdue, David T., "A Floppy Disk File Management System", M.S. Thesis, Virginia Polytechnic Institute and State University, 1976.

7. "Quality Control Program for Hi-Vol Filter Paper Weighing", DCLS Document, Unpublished, 1976.

8. Smith, Cecil L., Digital Computer Process Control, International Textbook Co., New York, 1972.

9. Tippie, J.W. and J.E. Kulago, "Modular-Programmed Data Acquisition/Analysis in Research", American Laboratory, Vol. 8, No. 10, pp 81-90, 1976.

VITA

Bertram Wayne Murphy was born in Lynchburg, Virginia on December 19, 1950. He attended E. C. Glass High School and graduated in June, 1969. He graduated from Virginia Polytechnic Institute and State University in June, 1974, with a Bachelor of Technology degree in Electrical Engineering Technology. Mr. Murphy was employed as an Electrical Engineer with Arch Mineral Corporation from July, 1974, to August, 1975. In September, 1975, he enrolled at Virginia Polytechnic Institute and State University and will receive his Master of Science degree in Electrical Engineering in May, 1977. After graduation, he will work for Babcock and Wilcox Corporation in Lynchburg, Virginia.

Mr. Murphy is a member of Phi Eta Sigma and Eta Kappa Nu.

Bertram W. Murphy

AN AUTOMATION AND
DATA MANAGEMENT SYSTEM
FOR AN ELECTRONIC AUTOBALANCE

by

Bertram W. Murphy

(ABSTRACT)

A typical application for an electronic autobalance is discussed, and the requirements for an autobalance automation system are developed. The design of six application programs that satisfy these requirements is presented, and the operation and interaction of these programs is discussed in detail. Typical weighing sessions of the autobalance while running under the automation system are described. Current status of the autobalance automation system is outlined, and recommendations for future action are made.