NASA CR-165540
FWS 7621

DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM (PART III: UNATTENDED POWER PERFORMANCE MONITOR)

by A. Hallett, D. Heidkamp, H. Neustadter, R. Olson

FAIRCHILD WESTON SYSTEMS INCORPORATED
DATA SYSTEMS DIVISION

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract DEN 3-98

NASA CR-165540
FWS 7621

DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM (PART III: UNATTENDED POWER PERFORMANCE MONITOR)

by A. Hallett, D. Heidkamp, H. Neustadter, R. Olson

FAIRCHILD WESTON SYSTEMS INCORPORATED
DATA SYSTEMS DIVISION

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract DEN 3-98

1. Report No.
NASA CR 165540

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM
(PART III: UNATTENDED POWER PERFORMANCE MONITOR)

5. Report Date
JANUARY 1983

6. Performing Organization Code

7. Author(s)
A. HALLETT, D. HEIDKAMP, H. NEUSTADTER, R. OLSON

8. Performing Organization Report No.

9. Performing Organization Name and Address
FAIRCHILD WESTON SYSTEMS INCORPORATED
POST OFFICE BOX 3041
SARASOTA, FLORIDA 33578

10. Work Unit No.

11. Contract or Grant No.
DEN 3-98

12. Sponsoring Agency Name and Address
NASA LEWIS RESEARCH CENTER
21000 BROOKPARK ROAD
CLEVELAND, OHIO 44135

13. Type of Report and Period Covered
CONTRACTOR REPORT

14. Sponsoring Agency Code
DOE/NASA

15. Supplementary Notes
PROJECT MANAGER: DR. HAROLD E. NEUSTADTER, NASA/LRC

16. Abstract
THIS MANUAL PROVIDES SOFTWARE DOCUMENTATION, OPERATIONAL PROCEDURES, AND DIAGNOSTIC INSTRUCTIONS FOR DEVELOPMENT VERSION OF AN UNATTENDED WIND TURBINE PERFORMANCE MONITORING SYSTEM DESIGNED TO BE USED FOR OFF-LINE INTELLIGENT DATA ACQUISITION IN CONJUNCTION WITH THE NASA LEWIS RESEARCH CENTER CENTRAL HOST COMPUTER.

17. Key Words (Suggested by Author(s))
PERFORMANCE MONITOR

18. Distribution Statement
UNCLASSIFIED, LIMITED

19. Security Classif. (of this report)
UNCLASSIFIED

20. Security Classif. (of this page)
UNCLASSIFIED

21. No. of Pages
53

22. Price*

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

SECTION I GENERAL DESCRIPTION

Section I

GENERAL DESCRIPTION

Introduction

This manual provides software documentation, operational procedures and diagnostic instructions for development version of an unattended wind turbine performance monitoring system designed to be used for off-line intelligent data acquisition in conjunction with the NASA Lewis Research Center central host computer.

References

(1) PDP11 Software Handbook, Digital Equipment Corporation 1980.

(2) Microcomputer Processors, Digital Equipment Corporation 1978.

(3) Microcomputer Interface Handbook, Digital Equipment Corporation 1980.

(4) PDP-11/70 Processor Handbook, Digital Equipment Corporation 1975.

(5) PDP11 Processor Handbook, Digital Equipment Corporation 1975.

(6) RT-11 Programmers Reference Manual, Digital Equipment Corporation 1980.

(7) PDP-11 MACRO-11 Language Reference Manual, Digital Equipment Corporation 1980.

(8) RT-11 Software Support Manual, Digital Equipment Corporation 1980.

(9) Peripherals Handbook, Digital Equipment Corporation 1980.

(10) Elite 1520P Technical Manual, Datamedia Corporation 1976.

Purpose

The purpose of this manual is to describe a software system which (when mated with appropriate hardware) collects power system performance data from a wind turbine station, processes the raw data and stores pertinent information in a data base which can be queried periodically via a commercial telephone link.

Scope

The Documentation contained in this manual describes the software system and related hardware components for a power system performance monitor centered around an existing LSI11/2 computer and R/T operating system. In its basic form this system monitors the parameters of as many as three wind turbines.

Hardware Requirements

Processor: The CPU is a DEC LSI11/2. It is configured with 32K of random access memory.

FWS 56A-7621 System Clock: This clock stores days, hours, minutes, and seconds in hardware registers. It also provides an external priority pulse to the CPU at a rate of 10SPS. This model is battery powered to be immune to station power fluctuations. It can be set to the proper date and time, both manually and from the software.

QSB-11 BUBBL-BOARD

The QSB-11 BUBBL-BOARD® is a complete non-volatile mass storage system implemented on a single dual height module suitable for operation in the LSI-11 microcomputer. The QSB-11 provides 128 K-bytes of mass storage in an all solid state package, completely eliminating all external chassis, power supplies, moving parts and mechanical storage media.

The QSB-11 is comprised of one bubble-memory device (with its attendant support circuitry), plus a complete controller which emulates the DEC RX01 single density floppy-disk system. The system also includes a built-in bootstrap.

The QSB-11 is compatible with the RT-11 operating system and is operated as the system mass-storage device using DEC's standard "DX" floppy-disk handler.

The QSB-11 plugs directly into the LSI-11 backplane bus. No other electrical connections are required. All required power (+5 and +12 volts) is derived from the bus.

The QSB-11 BUBBL-BOARD module is removable and transportable without loss of data contained within the magnetic bubble device. Therefore, data can be moved from one machine to another simply by pulling the BUBBL-BOARD out of one machine and installing it into the bus of another.

The QSB-11 system provides built-in self diagnostics which can be invoked with simple ODT commands. The self test mode is intended for use in acceptance testing, and as a field maintenance aid.

An activity indicator light (labeled RUN) on the QSB-11 indicates when a bubble device on the module is being accessed. Another indicator labeled ERROR illuminates if the controller ever detects an error in the bubble-stored data.

NOTE: DEC, LSI-11, AND RT-11 are registered trademarks of Digital Equipment Corporation, Maynard, Massachusetss, USA.

QSB-11 HARDWARE/SOFTWARE SPECIFICATIONS

Software Compatibility:	Command compatible with DEC RXV11/RX01 floppy disk software.	
Memory Capacity:	128 K-bytes (131,072 8-bit bytes).	
Command Functions:	Fill Buffer Empty Buffer Write Sector Read Sector Read Status	Read Error Message Initialize Enable/Disable Interrupt Run Self Test
Sector Buffer Size:	128 bytes	
Device Address Select:	177170	
Interrupt Vector Select:	264	
Bootstrap Address:	173000	

Access Time (first byte):	Less than 41 msec average, 82 msec maximum
Transfer Rate:	In excess of 68,000 bits per second
Sector Transfer Time:	Less than 18 milliseconds
Error Protection:	14-bit Fire Code. Corrects error bursts up to 5 bits long. Automatic re-try for soft errors.
Bubble Device Formatting:	Automatic synchronization by controller at power-up.
Power Required (maximum):	+5 volts DC at 1.4 amps. +12 volts DC at 0.05 amps, static; 0.2 amps when accessed.
Size:	Standard dual-height (double-width) module.
Operating Temperature:	0 to 70 degress Celsius, maximum, ambient.
Storage Temperature:	-40 to 100° C, without loss of stored data.
Weight:	14 ounces.

Overall Block Diagram - Description

A block diagram representing the major components of the software system is given in Figure 1.4A.

The "event enable" block is the start of the program proper. It enables the required interrupts and sets system variables to their proper initialization values. The background logic serves two major functions. One is to provide a "wait loop" from which "foreground processing" (events) may interrupt the system. The other is to interrogate the modem logic to determine when a "ring" signal has been received. "Foreground" processing is derived from an external clock operating at 10 SPS. The ADV11A Handler is responsible for bringing in A to D data, accumulating same, averaging data every 6 minutes, storing averages in the data block, looking for power outage conditions and updating system pointers. The Modem Logic is accessed via telephone communication. It establishes caller protocol, priorities and protections, data block transmissions for certain callers, and permits diagnostic and program development measures to be taken. The Diagnostic Software facilitates hardware and software maintenance by detecting and isolating system malfunctions.

PERFORMANCE MONITOR

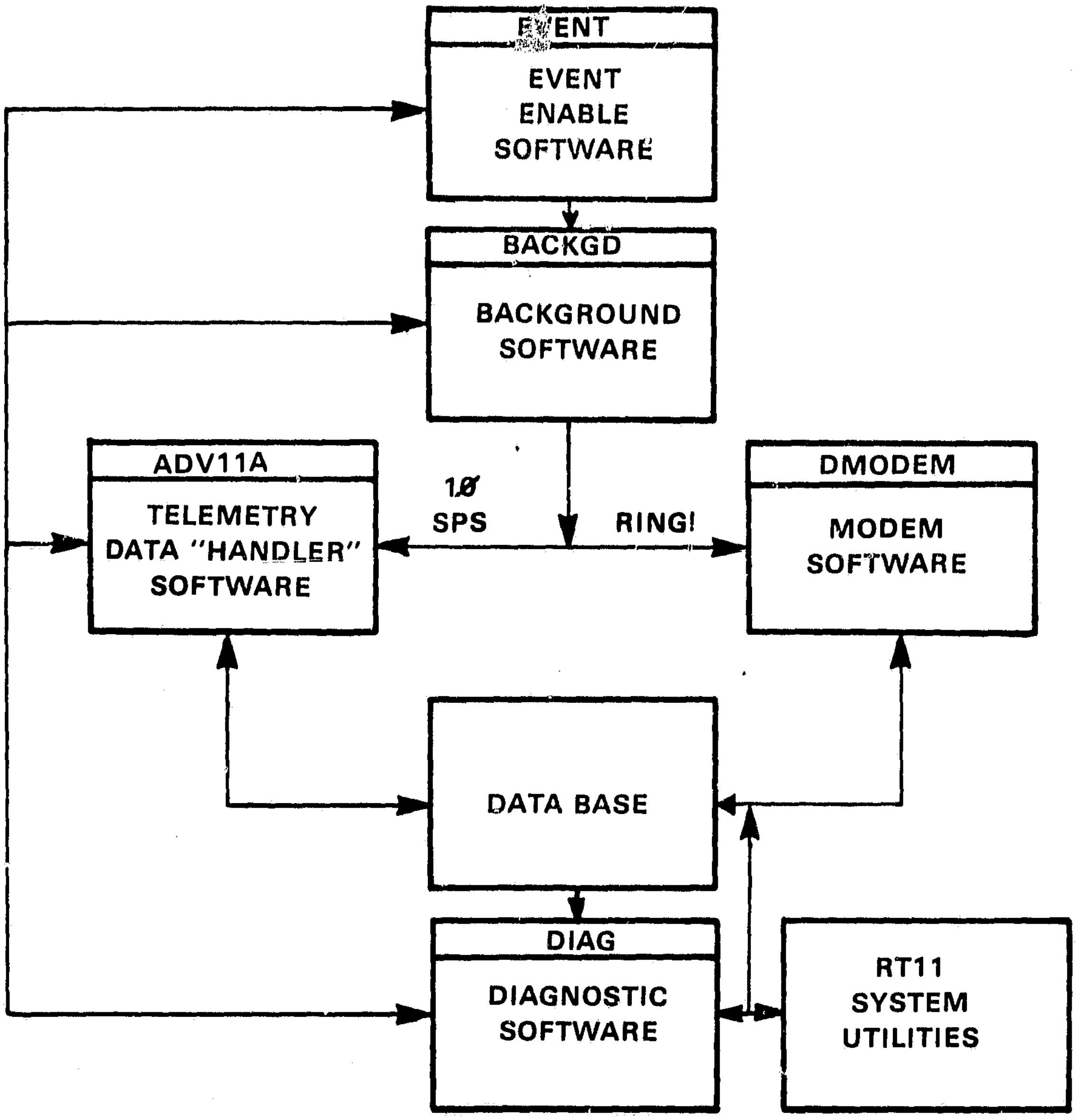


FIGURE 1.4A MAJOR SOFTWARE SYSTEM COMPONENTS

SECTION II HOUSEKEEPING SOFTWARE

PROGRAM: EVENT

PURPOSE: Initialize system.

CALLING
SEQUENCE: Executed via System Bootstrap Loader.

Description:

Enables required interrupts, conditions A/D channels, establishes FWA of the Data Base and initializes system variables.

Initialization Procedure:

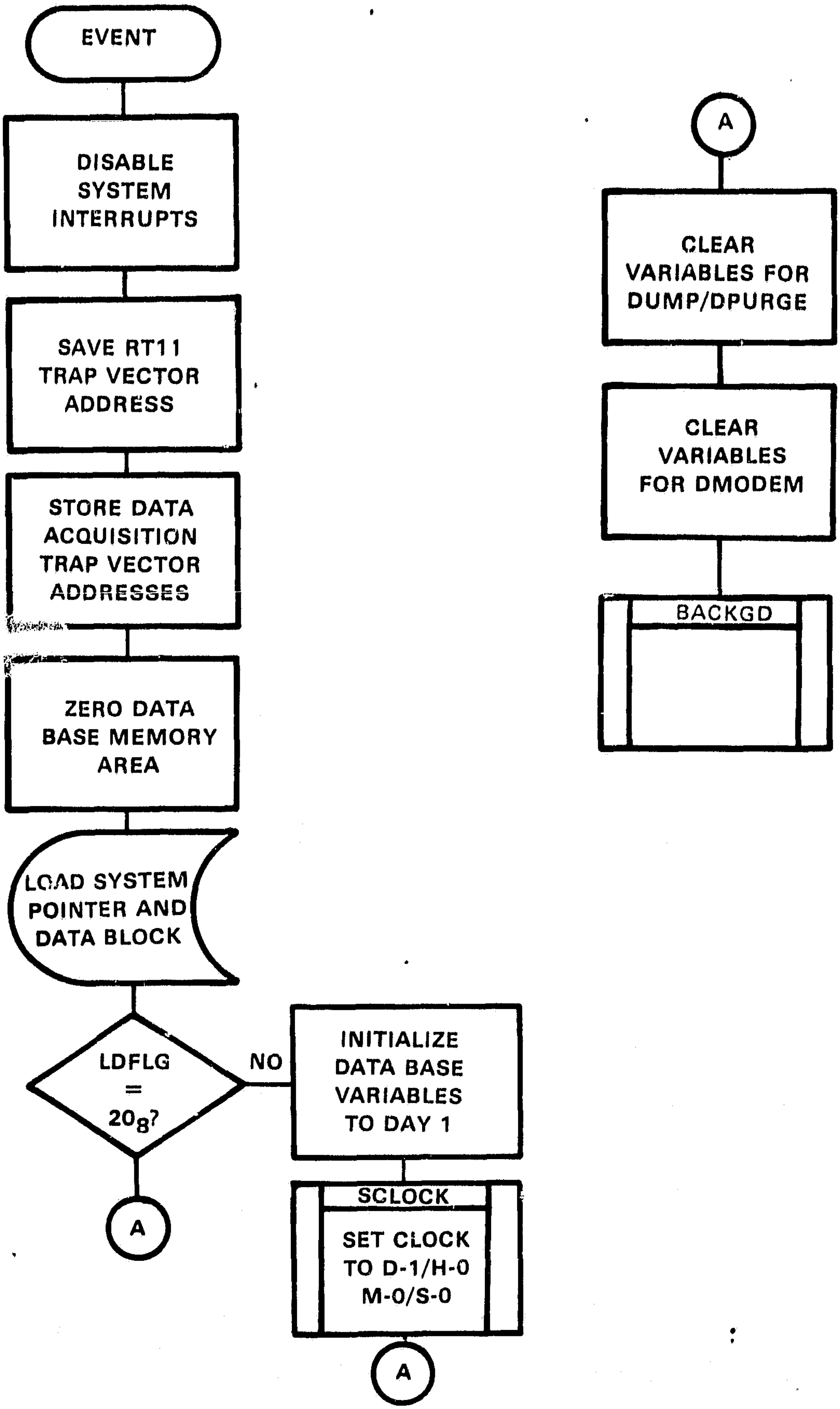
1. Set R4 to FWA of Data Base.
2. Clear Data Base, 8K locations.
3. Establish maximum number of channels to be monitored.
4. Set 4 locations in Purge equal to 1 corresponding to 4 days.
5. Specify storage used for counters.
6. Initialize day counters.
7. Clear storage used for counters.
8. Clear accumulator buffers.
9. Clear MODEM variables.

Special Conditions:

Prior to entry, the global "LDFLG" must be cleared, otherwise, system variables will not be initialized.

Globals:

Data Base Variables, MINCNT, OFF, DLOSS, CAL1, CAL2, CAL3, CAL4, CAL5, CAL6, CAL7, TB1180, TB124, XXCNT, XCNT, STARTD, STOPD, R2S, R0S, MSGFLG, PASS, ERR, CODES, CODEF, REQA, VBASE, LDFLG


EVENT
DISABLE SYSTEM INTERRUPTS
SAVE RT11 TRAP VECTOR ADDRESS
STORE DATA ACQUISITION TRAP VECTOR ADDRESSES
ZERO DATA BASE MEMORY AREA
LOAD SYSTEM POINTER AND DATA BLOCK
LDFLG = 20_8?
NO
INITIALIZE DATA BASE VARIABLES TO DAY 1
SCLOCK
SET CLOCK TO D-1/H-0 M-0/S-0
A
A
A
CLEAR VARIABLES FOR DUMP/DPURGE
CLEAR VARIABLES FOR DMODEM
BACKGD

PROGRAM: BACKGD

PURPOSE: Background routine which has program control between external clock interrupts.

CALLING
SEQUENCE: BR BACKGD

Real Time Entry:

While in the idle mode BACKGD monitors bit 15 of the DLV11-E RCSR and the diagnostice "REMFLG" in anticipation of a Modem status change (ring) or a diagnostic monitor request. When a modem "ring" is detected BACKGD transfers program control to subroutine DMODEM. When the diagnostic REMFLG is set BACKGD returns the RT11 operating system to the caller.

NOTE: DLV11-E Receiver Control Status Register is defined in DMODEM Documentation.

Special Conditions:

Upon every system "start up" and prior to entering BACKGD the system is checked for a power out. When a power out is not detected BACKGD is executed immediately. When a power out is detected a verification is made to determine whether power has been out for one or more minutes and/or one or more days.

When power is out for one or more days, the data block day pointers are reset equal to the current day. All data blocks associated with the power out day(s) are purged.

When power is out for one or more minutes the data loss word (200_8) is stored into the data block of the current day starting at the current and subsequent locations of the pointer "n" until "n" is equal to the increment past midnight "IPM".

NOTE: Power out recovery as stated above, takes place on an as available basis and in between clock interrupts.

Special Requirements:

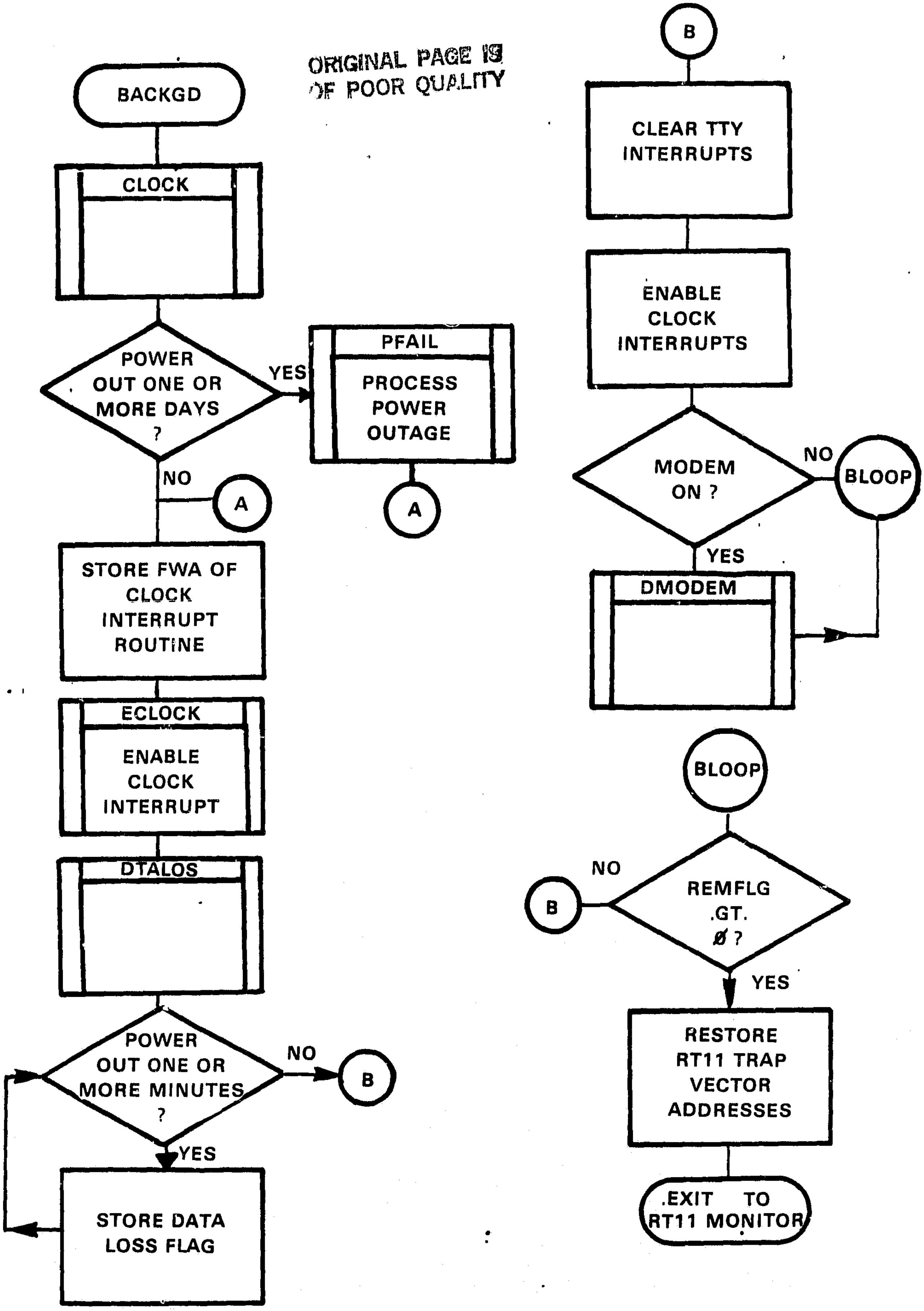
LSI11/2, DLV11-E Asynchronous Line Interface, Bell 103 Modem or equivalent, RS232 CPU Interface.

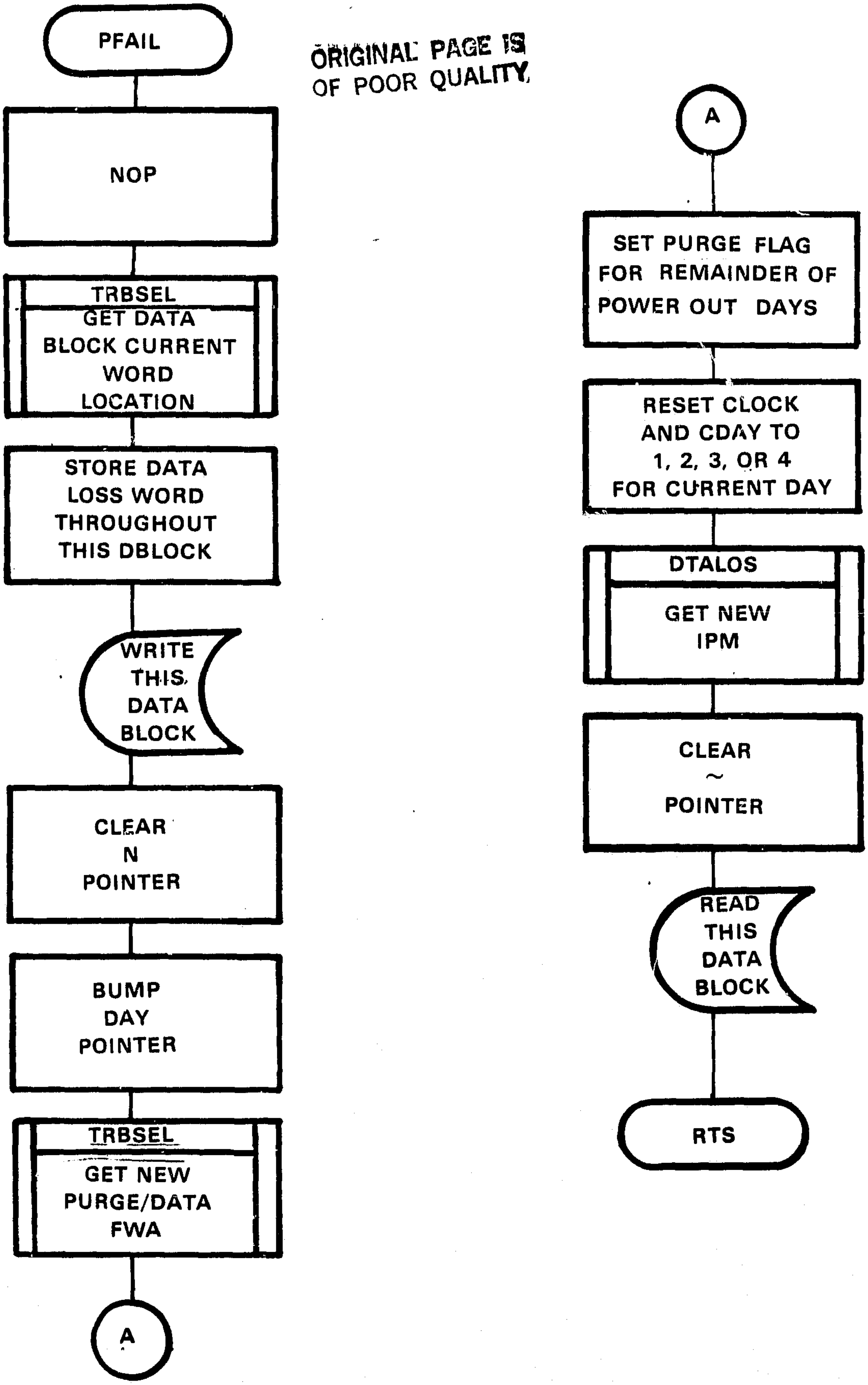
External Requirements:

DMODEM, PFAIL, CLOCK, DTALOS

Globals:

RCSR, REMFLG, IPM, N


BACKGD
CLOCK
POWER OUT ONE OR MORE DAYS ?
YES
PFAIL
PROCESS POWER OUTAGE
A
NO
A
STORE FWA OF CLOCK INTERRUPT ROUTINE
ECLOCK
ENABLE CLOCK INTERRUPT
DTALOS
POWER OUT ONE OR MORE MINUTES ?
NO
B
YES
STORE DATA LOSS FLAG
B
CLEAR TTY INTERRUPTS
ENABLE CLOCK INTERRUPTS
MODEM ON ?
NO
BLOOP
YES
DMODEM
BLOOP
NO
B
REMFLG .GT. Ø ?
YES
RESTORE RT11 TRAP VECTOR ADDRESSES
.EXIT TO RT11 MONITOR


PFAIL
NOP
TRBSEL
GET DATA BLOCK CURRENT WORD LOCATION
STORE DATA LOSS WORD THROUGHOUT THIS DBLOCK
WRITE THIS DATA BLOCK
CLEAR N POINTER
BUMP DAY POINTER
TRBSEL
GET NEW PURGE/DATA FWA
A
A
SET PURGE FLAG FOR REMAINDER OF POWER OUT DAYS
RESET CLOCK AND CDAY TO 1, 2, 3, OR 4 FOR CURRENT DAY
DTALOS
GET NEW IPM
CLEAR ~ POINTER
READ THIS DATA BLOCK
RTS

SECTION III REMOTE COMMUNICATIONS SOFTWARE

PROGRAM: DMODEM

PURPOSE: To maintain and support data communications between the DLV11-E Asynchronous Line Interface and the T103J Modem.

CALLING SEQUENCE: DMODEM assumes the standard calling conventions which are established for MACRO compatibility.

Serial Communications Line Interface

GENERAL DESCRIPTION:

The DLV11-E asynchronous line interface module interfaces the LSI-11 bus to the Bell T103J Data Modem. This module receives serial data from peripheral devices, assembles it into parallel data and transfers it to the LSI-11 bus. In addition, the DLV11-E accepts data from the LSI-11 bus, converts it into serial data and transmits it to the peripheral devices.

DLV11-E OPERATIONS:

Baud Rate - 300

Data bit selection - 8 bits, EVEN Parity-generated, Received and Transmitted by hardware.

Sync - Two STOP bits; frames the data bits received and transmitted by hardware. "Halt on Framing Error-Disabled."

Receiver control/status register (RCSR) - Console module at memory address 177560.

Receiver Data Buffer (RBUF) - Console module at memory address 177562.

Transmit Control/Status (XCSR) - Console module at memory address 177564.

Transmit Data Buffer (XBUF) - Console module at memory address 177566.

Interrupts - Receiver and Transmitter are disabled by software. Console module address 300 and 304 are therefore not used by DMODEM software.

PERIPHERAL CHARACTERISTICS:

CPU Interface - RS-232-C

Alternate Mode - Places the keyboard or terminal in teletype mode, producing upper case characters.

Line - Full Duplex; input received from line can go to tape, memory or printer; output from any device other than line can only go to line. In addition, control characters for reading and writing can only come from the line or keyboard.

Format - USASCII code, all data internal to the terminal is handled with a 7 bit format and EVEN parity.

Real Time Entry:

DMODEM is entered via BACKGD (Background subroutine) upon acknowledgement of a modem status change. This status change is determined by monitoring bit 15 of the RCSR register. The DLV11-E hardware asserts bit 15 when a transition of any of the following lines is detected.

.RING

.CLEAR TO SEND

.SECONDARY RECEIVE DATA

.DATA SET READY

RING (bit-15) - Reflects the state of the modem ring line, and when set indicates that a ring signal is being received from the modem.

CLEAR TO SEND (bit 14) - Reflects the state of the clear-to-send line from the modem, and when set indicates that the modem is ready to accept data from the interface for transmission.

CARRIER (bit 13) - Reflects the state of the modem carrier, and when set indicates that the carrier is up.

SECONDARY RECEIVED DATA (bit 11) - Provides a receive channel for supervisory data from the modem to the processor.

DATA SET READY (bit 10) - Reflects the state of the data set ready line from the modem, and when set indicates that the modem is powered-up and ready.
The RCSR Interrupt is enabled when bit 7 of the RCSR is set to '1'. However, this interrupt is not enabled for this system.

Real Time Entry Continued:

The current configuration of program DMODEM is set up such that all of the following conditions must be determined .TRUE. before data communication can begin.

1. Ring bit asserted

2. Modem power on

3. Modem ready to receive data from the interface for transmission (clear to send).

The .TRUE. state of the third condition is determined by the state of bit 15 RCSR. When set to '1' this bit reflects a .TRUE. condition. When set to a '0' this bit reflects a .NOT TRUE. condition. DMODEM monitors this bit periodically during data communication and transfers program control to the calling program when a .NOT TRUE. condition is acknowledged. In the event that all of the above conditions are .TRUE. all software requirements have been satisfied and data communication begins.

Telephone Caller Protocol:

The following information describes basi~ I/O between DMODEM and the telephone caller. In an effort to simplify the understanding of this basic I/O we shall first define the control characters that are "invisible" to the caller and required by both DMODEM and the Lerc IBM 3033/TSS System.

The control characters, DC1 and DC3, have been employed so that the TSS System can pace the input and output of DMODEM according to its TSS scheduling process.

SYSTEM REQUIREMENTS:

The DC1 (x-on) control character must be supplied by the callers system to initiate character (message) transmissions from DMODEM to the caller.

The DC3 (x-off) control character must be supplied by the callers system to stop character (message) transmissions from DMODEM to the caller. Characters that are transmitted to DMODEM in response to messages sent from DMODEM should always follow the DC3 control character.

Considering the above system requirements, we shall talk about "telephone caller protocol" as it is understood by the caller/operator.

Upon entry, DMODEM transfers a message to the telephone caller, then waits for a response. Message transmissions are in groups of 8-bit (EVEN Parity), ASCII characters. Character transmissions are complete when a carriage return is recognized by the caller, or when a DC3 character is recognized by DMODEM.

While waiting for the telephone caller's response, DMODEM monitors status lines of the DLV11-E. These status lines indicate whether the telephone caller hung up his/her telephone or whether a character has been received.

Each time a character is received from the caller, the Modem Logic determines whether an error was generated during transmission (Receive Error).

ERROR PROCEDURES:

Upon recognition of an error, the type of error is determined, then DMODEM waits for the completion of the telephone caller's character transfers. Thereafter, the incorrect characters and an error message are transmitted prior to the acknowledgement of the error. The telephone caller may make corrections, if applicable, then re-transmit.

NOTE: Incorrect password characters are not echoed.

DATA COMMUNICATION PROCEDURES:

When valid character strings have been received from the telephone caller, DMODEM translates and verifies the characters to determine whether or not the caller responded correctly. When it has been determined that the telephone caller's response is correct, DMODEM acts upon the response, transmits another message in a sequence, then re-initializes its protocol. When the last message in the sequence is transmitted and the telephone caller responds correctly, DMODEM transmits the first message in the sequence and communication continues.

Upon recognition of an incorrect response from the telephone caller, DMODEM performs the error procedures stated above.

Data communication and Modem Logic execution is discontinued when (1) the Modem power is turned off, or (2) the telephone caller hangs up his/her telephone.

Message I/O: Communication between DMODEM and the telephone caller is shown below. The DC1's, DC3's and CR's (carriage return) characters are displayed in an effort to give a complete understanding of the message I/O. However, DC1's and DC3's are supplied by the hardware and therefore "invisible" to the caller/operator.

Line in: <DC1>

Line out: ******* (80 total) ....<CR>

Line in: <DC3> <DC1>

Line out: Who is calling please? <CR>

Line in,
Response: <DC3> (F)airchild (W)eston <CR> <DC1>
or
<DC3> (PL)umbrook <CR> <DC1>

Line out: Please give function. <CR>

Note: DC1/DC3 characters are no longer required if the caller is Fairchild Weston.

Line in,
Response: <DC3> (DU)MP <CR> <DC1>
or
<DC3> (D)UMP and (P)urge <CR> <DC1>
or
<DC3> (CH)ange Password <CR> <DC1>
or
<DC3> (DI)agnostic <CR> <DC1>

Line out: Please give password. <CR>

Line in,
Response: <DC3> Enter password characters, 78 characters maximum. <CR> <DC1>

NOTE: DMODEM transfers program control to subroutine DUMP, DPURGE, CHPSWD or DIAG to have the desired function performed. Subroutines DPURGE, CHPSWD, and DIAG require password identification before execution.

The following messages shall be transmitted to the caller upon recognition of an incorrect response or a Receiver Error.

Response Errors:

'Sorry Wrong Number'

'Sorry Wrong Function'

'Sorry Wrong Password'

RECEIVE ERRORS:

OVERRUN ERROR - CHARACTER LOST - The processor has failed to service the RX DONE flag within the time required to load another character into the Receiver Data Buffer, i.e., (1/baud rate) x (bits per character) seconds. Hence, the previous character was overwritten (lost).

FRAMING ERROR - STOP BIT MISSING - The character received was not followed by a valid stop bit. In the Asynchronous mode, each received character must be preceded by a start bit and succeeded by a stop bit, which serves to synchronize the DRV11-E receive logic.

PARITY ERROR - REQUIRE EVEN PARITY - The parity of the received character does not agree with the parity programmed (EVEN).

Special Conditions:

Upon entry, the trap counters are interrogated to determine whether a system trap failure occurred since the last data transmission. When true, DMODEM sets the trap flag "TPERR" then prepares for telephone caller protocol. At such times as the caller properly identifies him/herself, the appropriate trap message is transmitted.

When not true, DMODEM continues the logic flow as stated above.

Special Requirements:

LSI11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

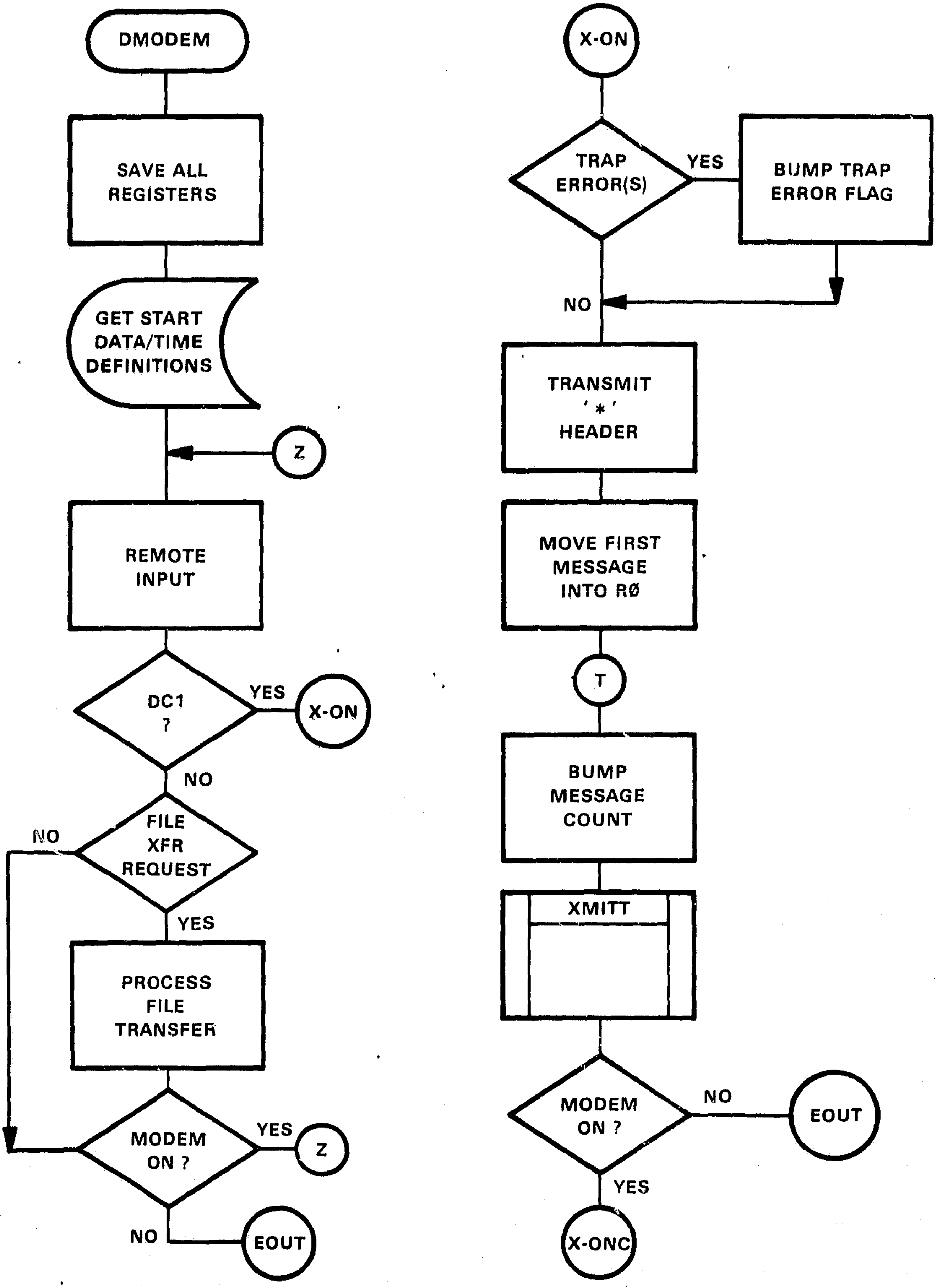
External Requirements:

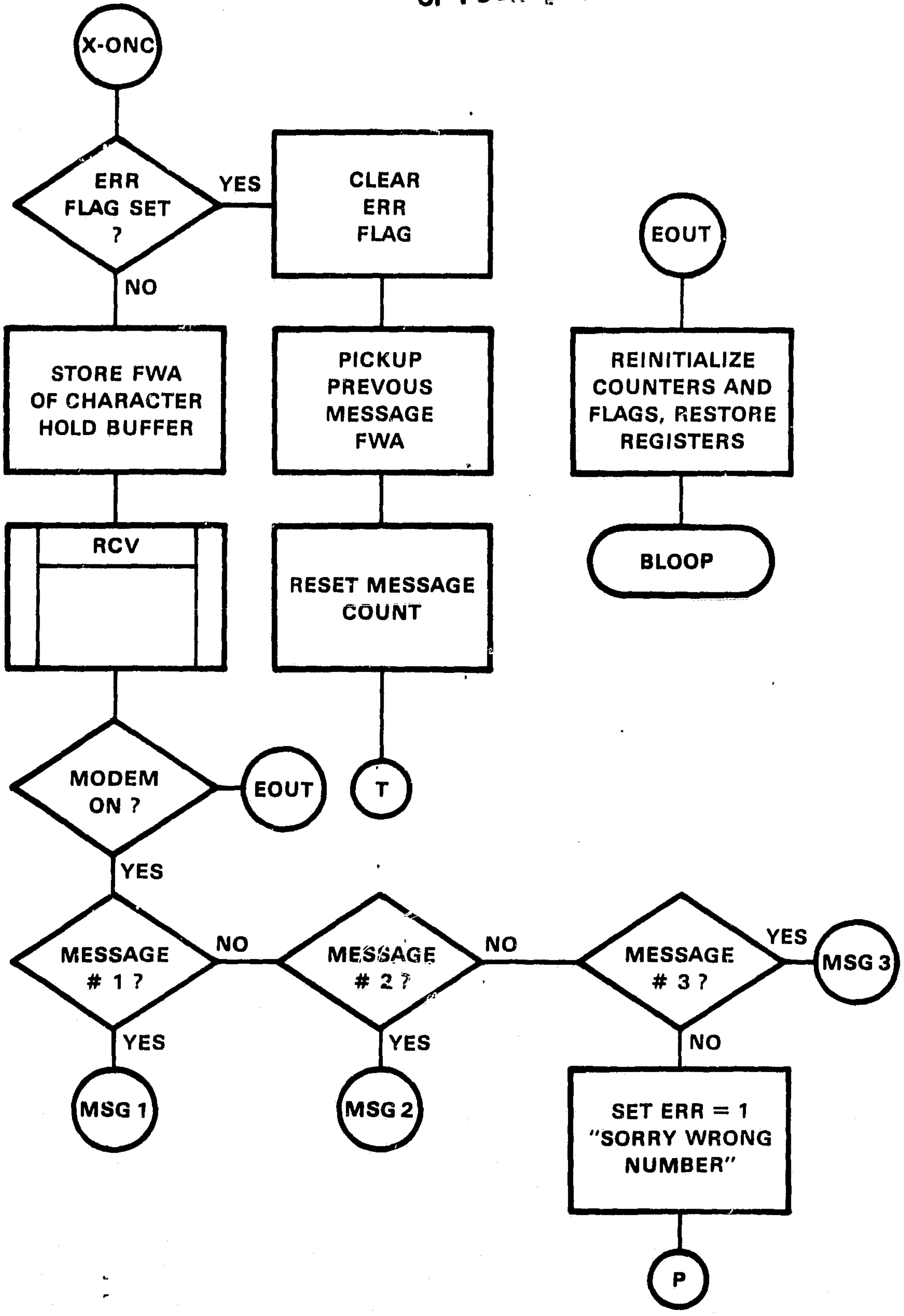
DUMP, DPURGE, DIAG, CHPSWD, XMITT, RCV

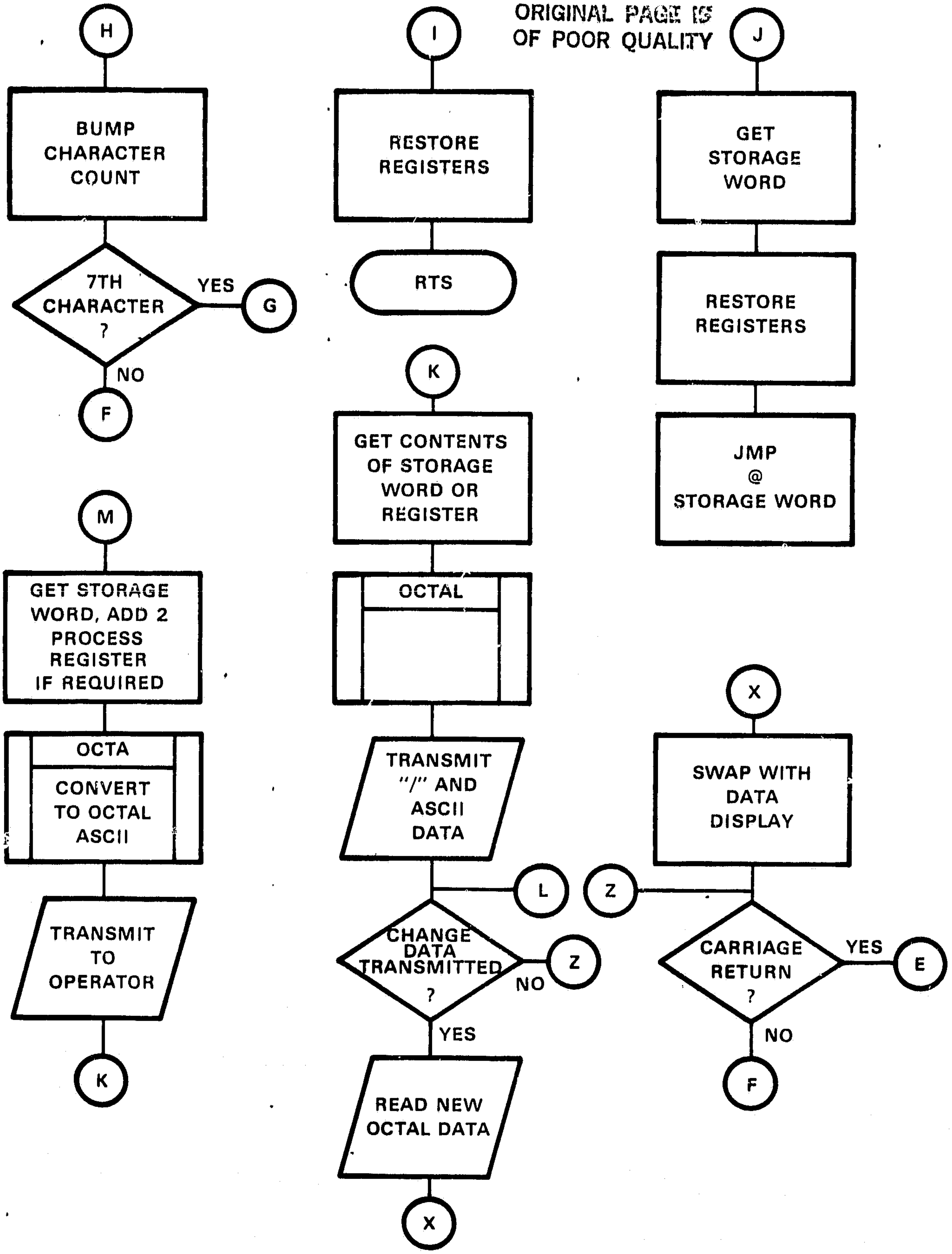
Globals:

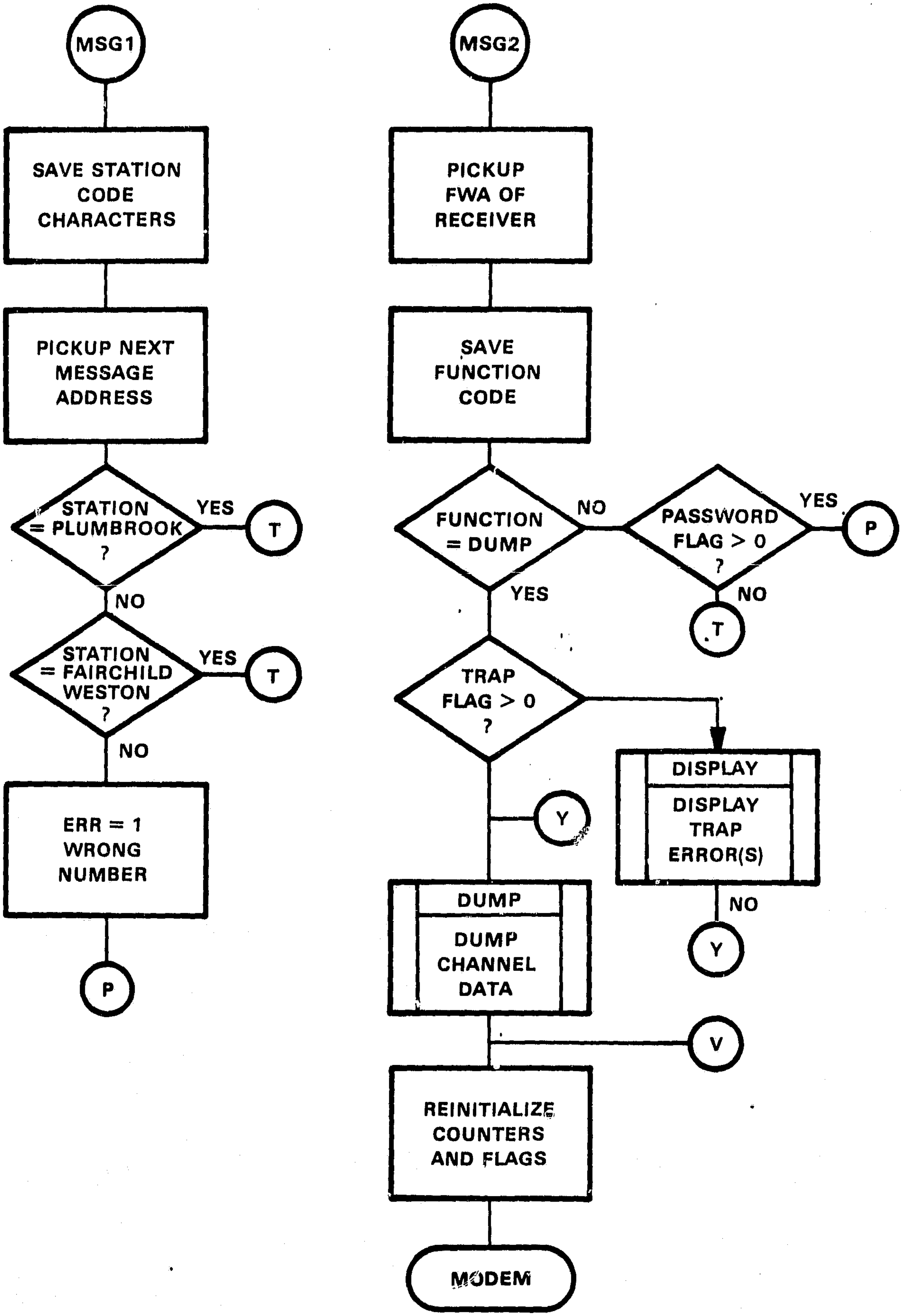
FWPSWD, PLPSWD, MSGFLG, ERR, SAVEB, PASS, CODES, CODEF, MSG, MSG2, MSG3, ERR1,

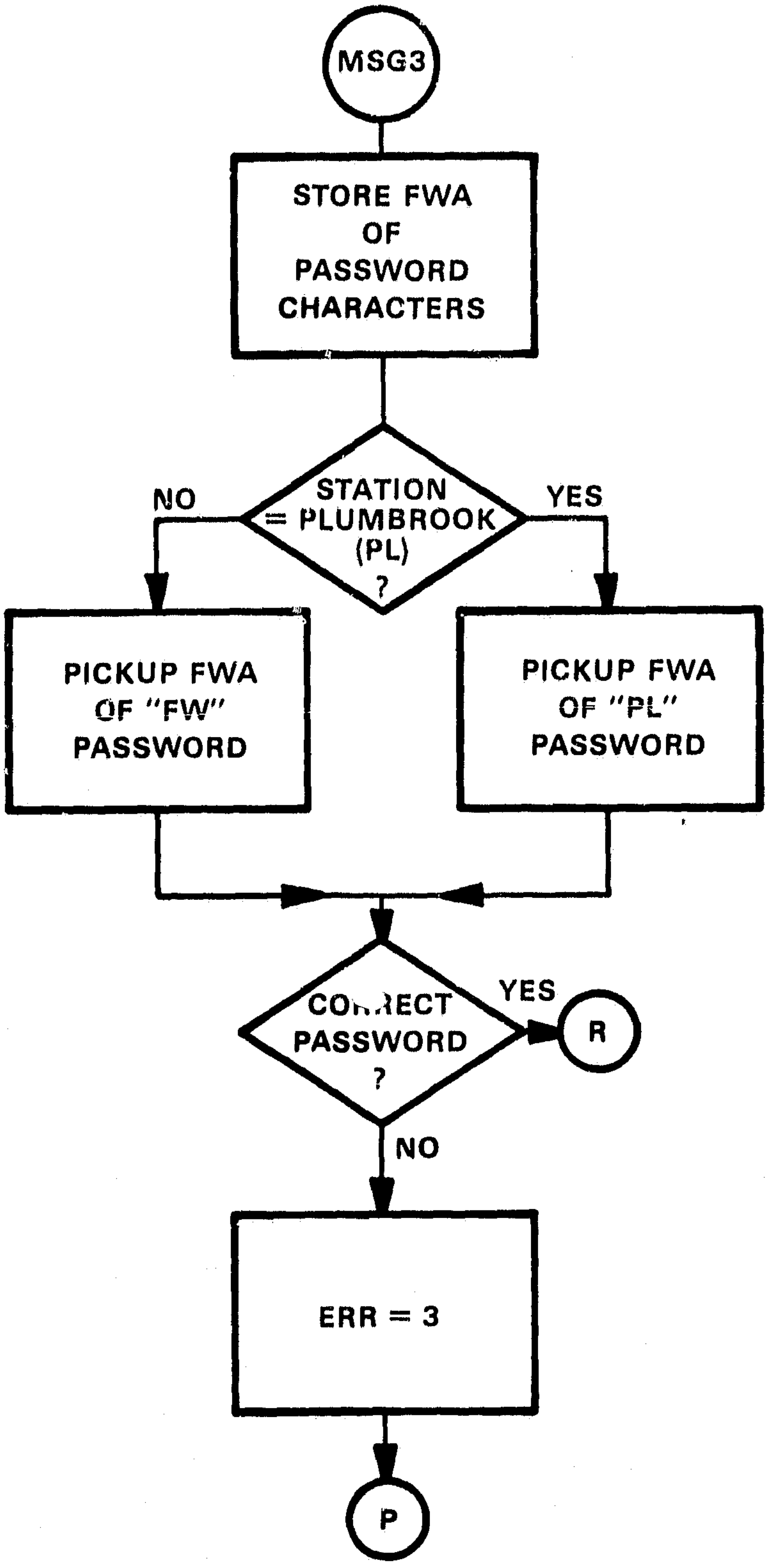
ERR3, ERR4, ERR5, ERR6, F,U,NC,T,I,O,NS, RCSR, RBUF, XCSR, XBUF, MODEMF, DRCSR.


DMODEM
SAVE ALL REGISTERS
GET START DATA/TIME DEFINITIONS
Z
REMOTE INPUT
DC1 ?
YES
X-ON
NO
FILE XFR REQUEST
NO
YES
PROCESS FILE TRANSFER
MODEM ON ?
YES
Z
NO
EOUT
X-ON
TRAP ERROR(S)
YES
BUMP TRAP ERROR FLAG
NO
TRANSMIT '*' HEADER
MOVE FIRST MESSAGE INTO RØ
T
BUMP MESSAGE COUNT
XMITT
MODEM ON ?
NO
EOUT
YES
X-ONC


X-ONC
ERR FLAG SET ?
YES
NO
CLEAR ERR FLAG
STORE FWA OF CHARACTER HOLD BUFFER
PICKUP PREVOUS MESSAGE FWA
EOUT
REINITIALIZE COUNTERS AND FLAGS, RESTORE REGISTERS
BLOOP
RCV
RESET MESSAGE COUNT
MODEM ON ?
EOUT
T
YES
MESSAGE # 1 ?
NO
MESSAGE # 2 ?
NO
MESSAGE # 3 ?
YES
MSG 3
YES
YES
NO
MSG 1
MSG 2
SET ERR = 1 "SORRY WRONG NUMBER"
P


H
BUMP CHARACTER COUNT
7TH CHARACTER ?
YES
G
NO
F
I
RESTORE REGISTERS
RTS
J
GET STORAGE WORD
RESTORE REGISTERS
JMP @ STORAGE WORD
K
GET CONTENTS OF STORAGE WORD OR REGISTER
OCTAL
TRANSMIT "/" AND ASCII DATA
L
CHANGE DATA TRANSMITTED ?
NO
Z
YES
READ NEW OCTAL DATA
X
M
GET STORAGE WORD, ADD 2 PROCESS REGISTER IF REQUIRED
OCTA
CONVERT TO OCTAL ASCII
TRANSMIT TO OPERATOR
K
X
SWAP WITH DATA DISPLAY
Z
CARRIAGE RETURN ?
YES
E
NO
F


MSG1
SAVE STATION CODE CHARACTERS
PICKUP NEXT MESSAGE ADDRESS
STATION = PLUMBROOK ?
YES
T
NO
STATION = FAIRCHILD WESTON ?
YES
T
NO
ERR = 1 WRONG NUMBER
P
MSG2
PICKUP FWA OF RECEIVER
SAVE FUNCTION CODE
FUNCTION = DUMP
NO
PASSWORD FLAG > 0 ?
YES
P
NO
T
YES
TRAP FLAG > 0 ?
DISPLAY
DISPLAY TRAP ERROR(S)
NO
Y
Y
DUMP
DUMP CHANNEL DATA
V
REINITIALIZE COUNTERS AND FLAGS
MODEM


MSG3
STORE FWA OF PASSWORD CHARACTERS
STATION = PLUMBROOK (PL) ?
NO
YES
PICKUP FWA OF "FW" PASSWORD
PICKUP FWA OF "PL" PASSWORD
CORRECT PASSWORD ?
YES
R
NO
ERR = 3
P

PROGRAM: DUMP DPURGE

PURPOSE: Dump and/or Dump and Purge data blocks, transferring same via the DLV11-E Line Interface module and the T103J Modem.

CALLING SEQUENCE:

```
JSR PC,DUMP
    or
JSR PC,DPURGE
```

Data Block Transmissions:

Four data blocks per channel are contained in the Data Base. Each block is comprised of 120 words of storage, enough space to store processed information for up to 16 channels for four consecutive days. Each data block is maintained and updated by various subroutines and sub-processes described in this manual. Ref... Documentation for "BASE", this manual.

Real Time Entry:

DUMP: Transmits data blocks that are complete and that have not been purged.

DPURGE: Transmits data blocks that are complete and that have not been previously purged. Sets a purge flag for each data block transmitted. Password identification is required before execution.

Special Conditions:

Converts data to Hexadecimal then to ASCII. Transmits 4 MSB of byte followed by 4 LSB. Stops when all data for all channels of a data block have been transmitted. In addition, transmits a carriage return as "End of Block" indicator. Dumps all or selected data blocks. Selected data blocks may be dumped via the command "Du n" where "n" equals the data blocks associated Julian day number 0 to 365.

Special Requirements:

LSI11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

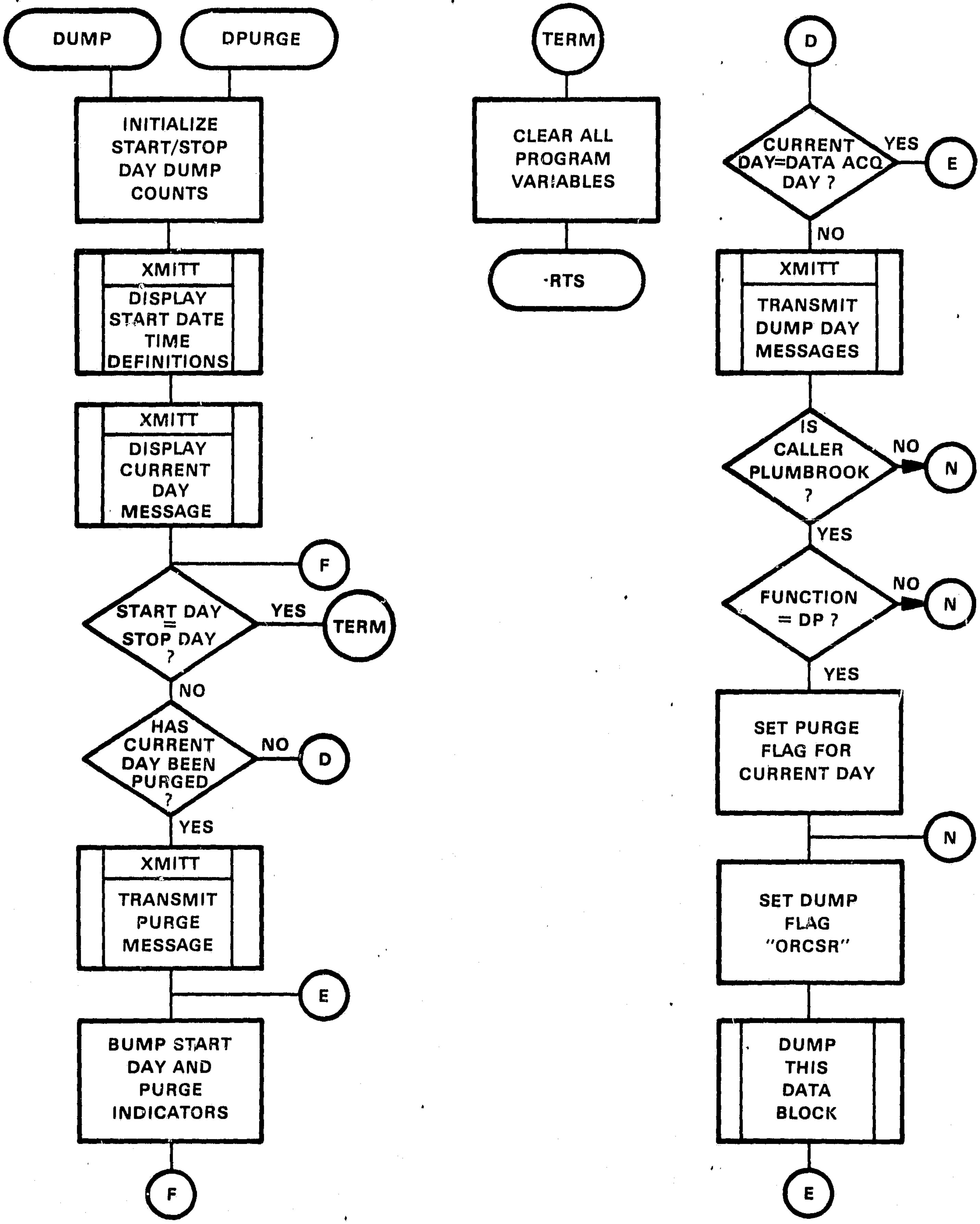
External Requirements:

XMITT

Globals:

XCNT, XXCNT, STARTD, STOPD, R2S, ROSS, ROSA, HEXASC, PURGE,M ADC1M1, MAXCH,

SR1, LF, CR, CODES, CODEF, M, XCSR, RCSR, XBUF


DUMP
DPURGE
INITIALIZE START/STOP DAY DUMP COUNTS
XMITT
DISPLAY START DATE TIME DEFINITIONS
XMITT
DISPLAY CURRENT DAY MESSAGE
F
START DAY = STOP DAY ?
YES
TERM
NO
HAS CURRENT DAY BEEN PURGED ?
NO
D
YES
XMITT
TRANSMIT PURGE MESSAGE
E
BUMP START DAY AND PURGE INDICATORS
F
TERM
CLEAR ALL PROGRAM VARIABLES
·RTS
D
CURRENT DAY=DATA ACQ DAY ?
YES
E
NO
XMITT
TRANSMIT DUMP DAY MESSAGES
IS CALLER PLUMBROOK ?
NO
N
YES
FUNCTION = DP ?
NO
N
YES
SET PURGE FLAG FOR CURRENT DAY
N
SET DUMP FLAG "ORCSR"
DUMP THIS DATA BLOCK
E

PROGRAM: CHPSWD - Subroutine for DLV11-E/T103J Modem communications

PURPOSE: Change Your Password

CALLING
ROUTINE: JSR PC, CHPSWD

I/O Communications:

CHPSWD performs the following functions:

1. Transmits the message "Enter new Password."
2. Receives new password characters and checks for receiver errors. When Receiver errors are found, CHPSWD determines the type, transmits an error message then solicits the caller to "Enter New Password." When a new password is correctly received, CHPSWD changes the caller's password then,
3. Exits.

Special Conditions.

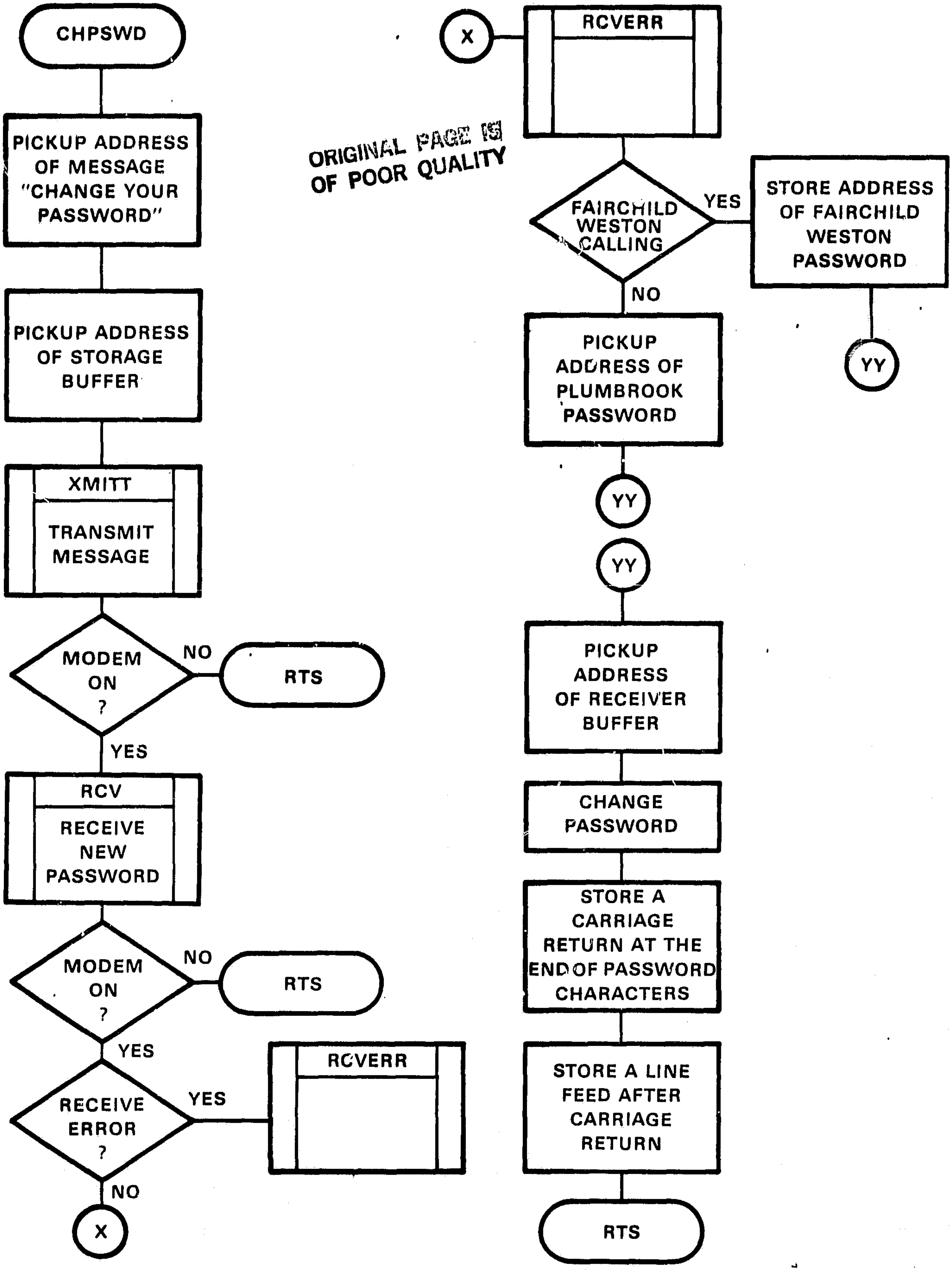
Password characters must not exceed 78 characters.

Special Requirements:

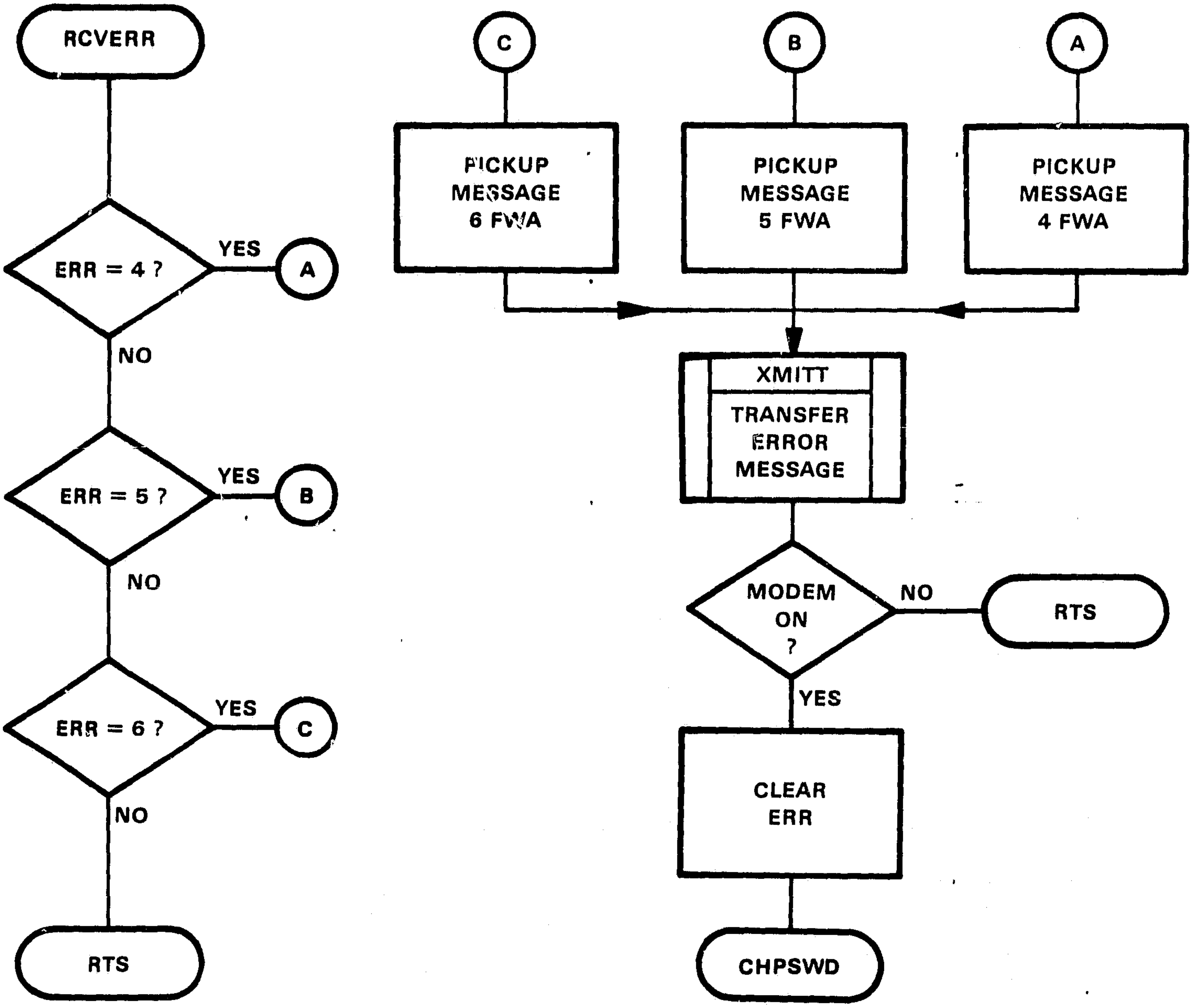
LSI11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements: XMITT, RCV.

Globals: ERR, CR, LF, ERR4, ERR5, ERR6, FWPSWD, PLPSWD, SAVEB, CODES.


CHPSWD
PICKUP ADDRESS OF MESSAGE "CHANGE YOUR PASSWORD"
PICKUP ADDRESS OF STORAGE BUFFER
XMITT
TRANSMIT MESSAGE
MODEM ON ?
NO
RTS
YES
RCV
RECEIVE NEW PASSWORD
MODEM ON ?
NO
RTS
YES
RECEIVE ERROR ?
YES
RCVERR
NO
X
X
RCVERR
FAIRCHILD WESTON CALLING
YES
STORE ADDRESS OF FAIRCHILD WESTON PASSWORD
YY
NO
PICKUP ADDRESS OF PLUMBROOK PASSWORD
YY
YY
PICKUP ADDRESS OF RECEIVER BUFFER
CHANGE PASSWORD
STORE A CARRIAGE RETURN AT THE END OF PASSWORD CHARACTERS
STORE A LINE FEED AFTER CARRIAGE RETURN
RTS

ORIGINAL PAGE IS
OF POOR QUALITY



PROGRAM: XMITT RCV - Subroutines for DLV11-E/T103J Modem Communications

PURPOSE:

XMITT: Consecutively, transmit 8-bits of a single word.

RCV : Consecutively, receive and transfer 8-bits of a single word to a buffer area supplied by the calling program. In addition, tests for receiver errors and sets an error flag when applicable.

CALLING SEQUENCE:

```
                  MOV   #TBUF,RO     ; RO=FWA of user transmitter buffer.
                  JSR   PC,XMITT

        EXIT1:    BR    OFF          ; Return here if Modem power off

        EXIT2:    NOP                ; Return here if Modem power on
                        and/or
                  MOV   #RBUF,R1     ; R1=FWA of used receiver buffer
                  JSR   PC,RCV

        EXIT3:    BR    OFF          ; Return here if Modem power off

        EXIT4:    NOP                ; Return here if Modem power on
```

XMITT/RCV:

Prior to receiving or transmitting characters, RCV and XMITT monitor bit 14 of the RCSR. When set to '1', this bit indicates that the Clear to Send line of the DLV11-E is asserted and that the T103J Modem is powered on and ready. When set to 0, this bit indicates that the Clear to Send line has been disabled and that the T103J Modem is power off.

Dependent on the state of RSCR bit 14, XMITT performs the following functions:

NOTE: Upon entry, XMITT enables the DLV11-E RCSR Request to Send Line and prepares to accept transmission from the caller.

Bit 14 Asserted:

1. Test XCSR bit 8 (Transmitter Ready) to determine whether the DLV11-E Transmitter Buffer can accept another character.

2. When not true, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.

3. When true, test receiver buffer (RBUF) for a DC1 (x-on) character.

4. When not true, continues monitoring RCSR bit 14/8 and testing RBUF for DC1 character.

5. When true, transfers 8 bits of the current buffer location. Bumps R1 to point to the next 8 bit character. Checks for a line feed character and transfers program control to EXIT2 when encountered. Otherwise, continues monitoring RCSR bit 14 and bit 8, repeating the functions as stated above.

Bit 14 Disabled:

1. Returns program control to EXIT1 of the calling program.
Dependent on the state of RCSR bit 14, RCV performs the following functions:

NOTE: Upon entry, RCV disables the DLV11-E RCSR Request to Send Line and prepares to accept transmission from the caller.

Bit 14 Asserted:

1. Tests RCSR bit 8 (Receiver Done) to determine whether an entire character has been received.

2. When not true, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.

3. When true, test receiver buffer (RBUF) for a DC3 (x-off) character.

4. When not true, continues monitoring bits 14 and 8 and testing for a DC3 character.

5. When true, transfers the received character to the user receiver buffer. Checks for receiver errors. Sets an error flag when necessary. Makes provisions for rubout characters. Returns program control to EXIT4 upon receipt of a carriage return character. Otherwise, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.

Bit 14 Disabled:

1. Returns program control to EXIT3 of the calling program.

User Receiver Buffer Upon Exit:

Character data is stored sequentially into the current and subsequent byte locations of the receiver buffer followed by a carriage return and a line feed.

Error Conditions:

The global ERR is set to a #4, 5, or 6 when a receiver error is encountered. These error numbers are defined as follows:

ERR		
4	=	Overrun Error, character lost
5	=	Framing Error, stop bit missing
6	=	Parity Error, EVEN parity required

ERROR RECOVERY: At the discretion of the calling program.

Special Conditions:

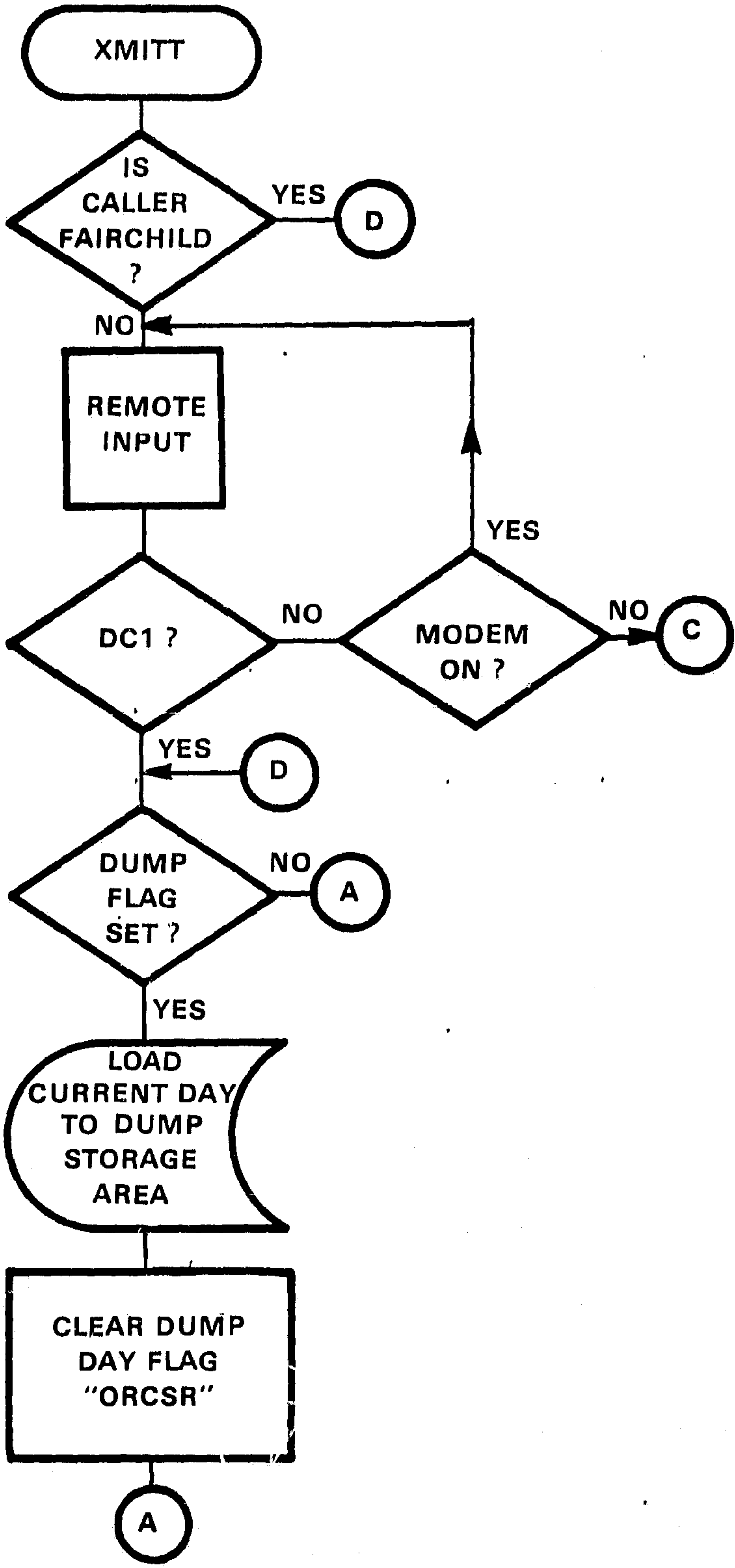
Under normal conditions (Modem power on), the user supplied transmitter buffer (TBUF) must contain a line feed character in the last byte location to terminate transmission. Receiver operations are terminated only after a carriage return character is received. Also, the user should allocate an additional byte location at the end of RBUF to allow for a line feed character, supplied by RCV.

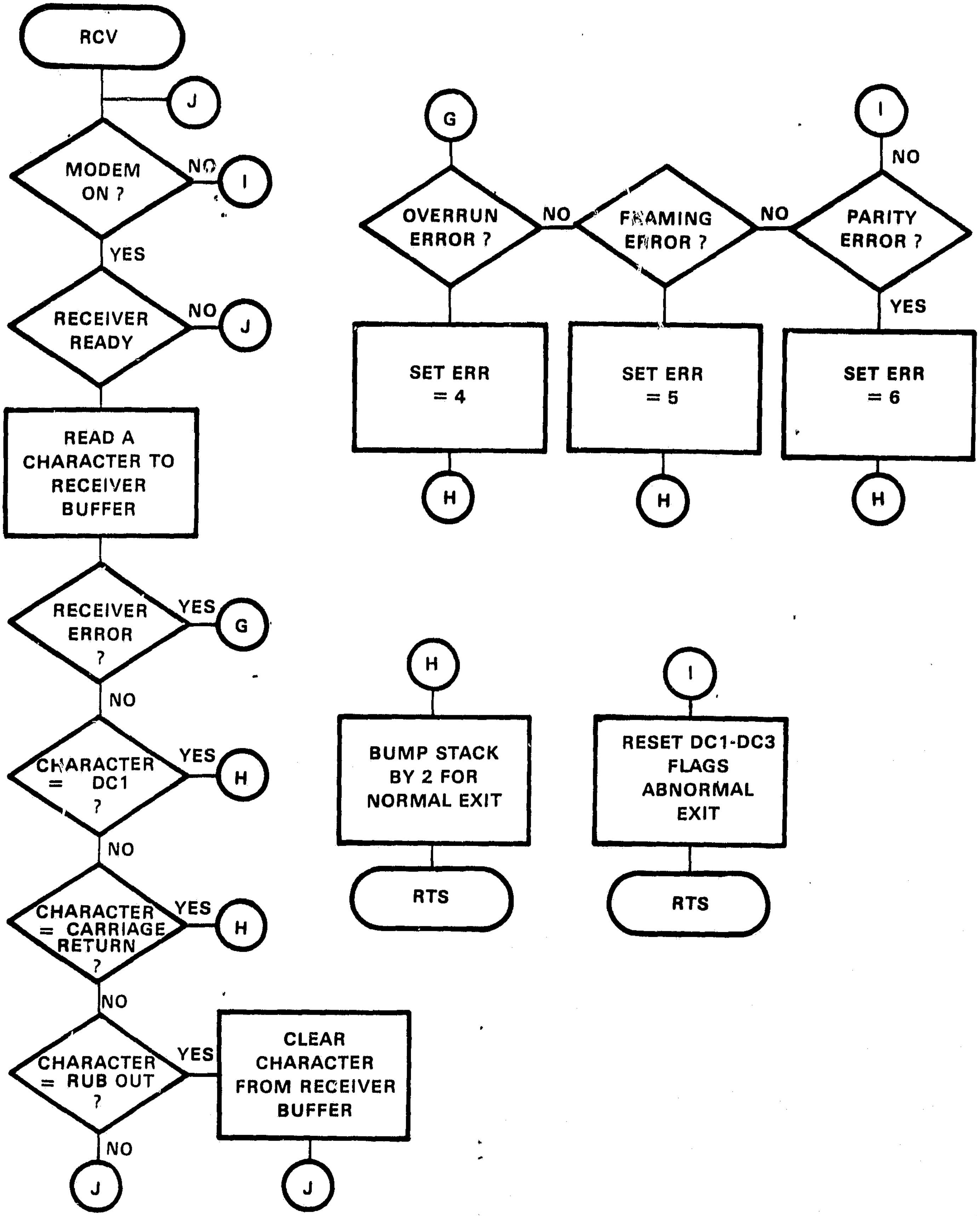
Special Requirements:

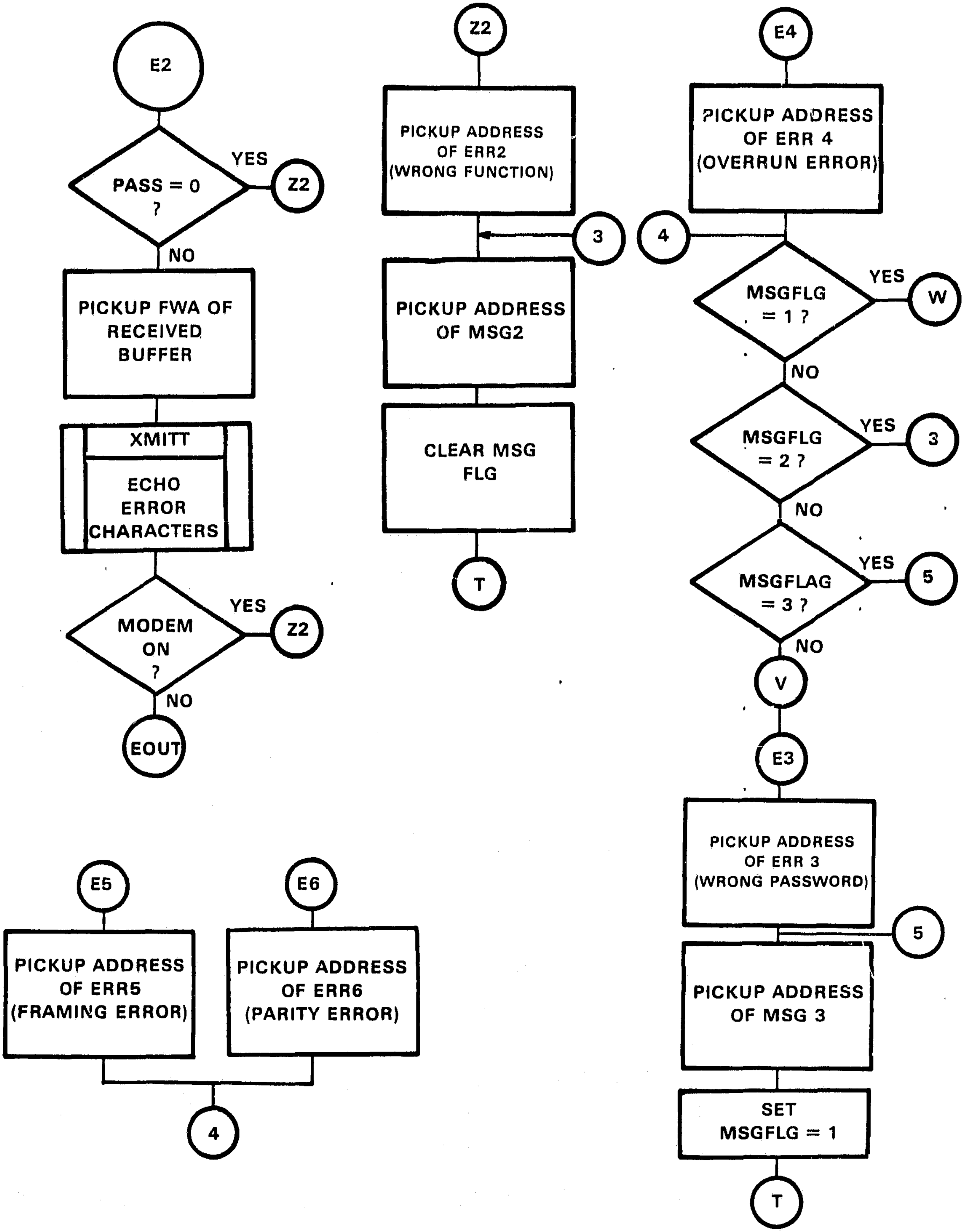
LSI11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

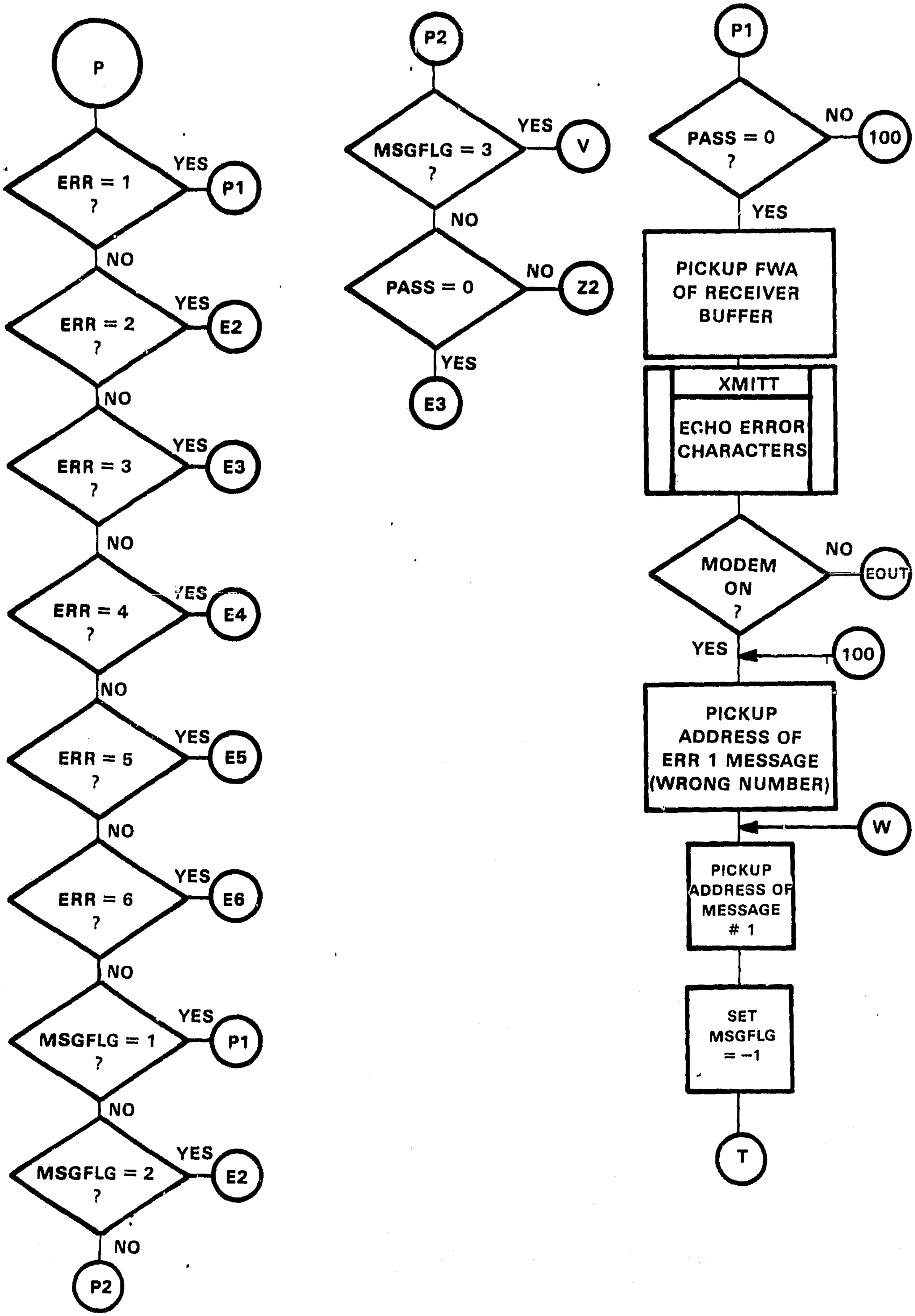
External Requirements: None

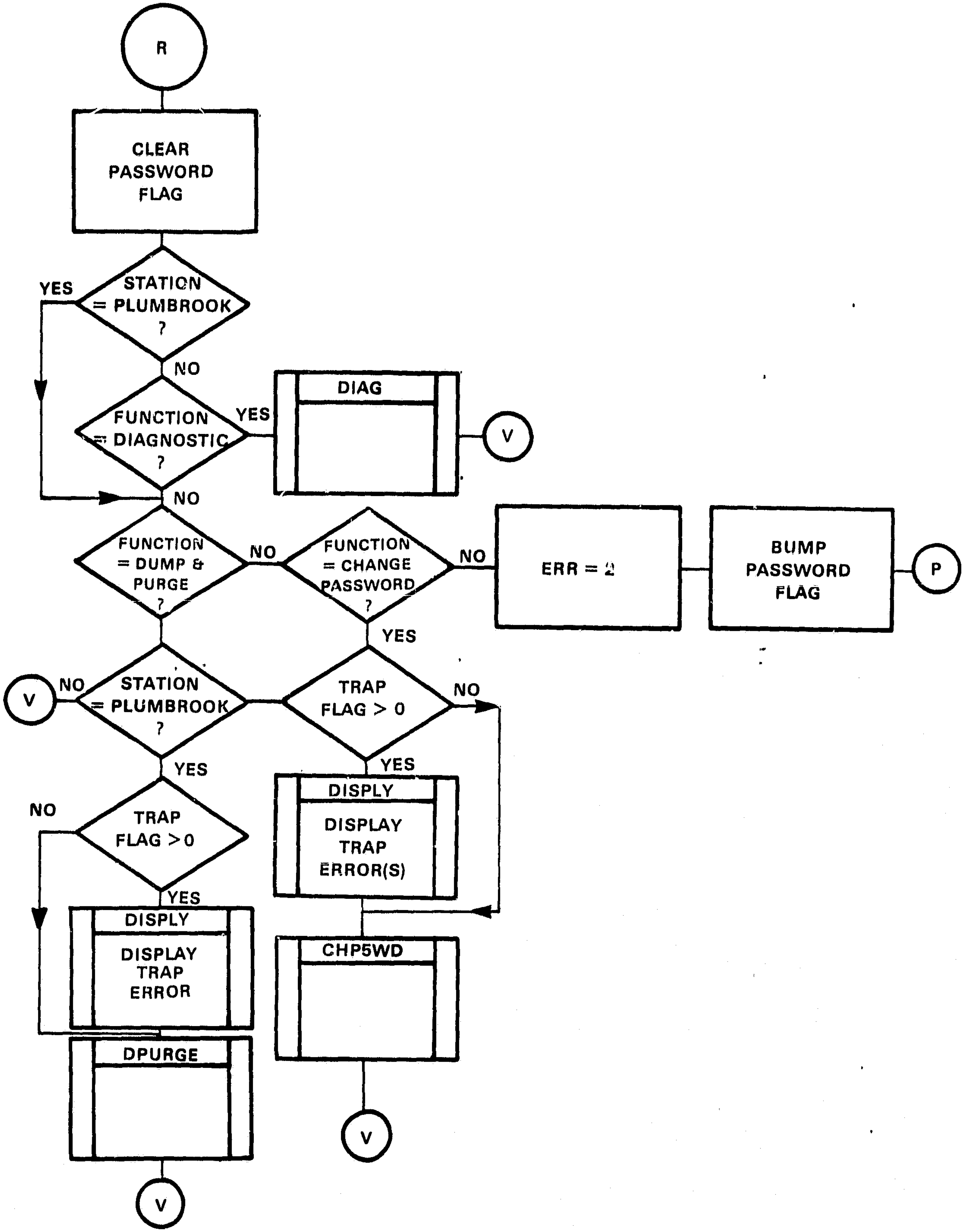
Globals: ERR, RCSR, XCSR, RBUF, XBUF, LF, CR, SR4

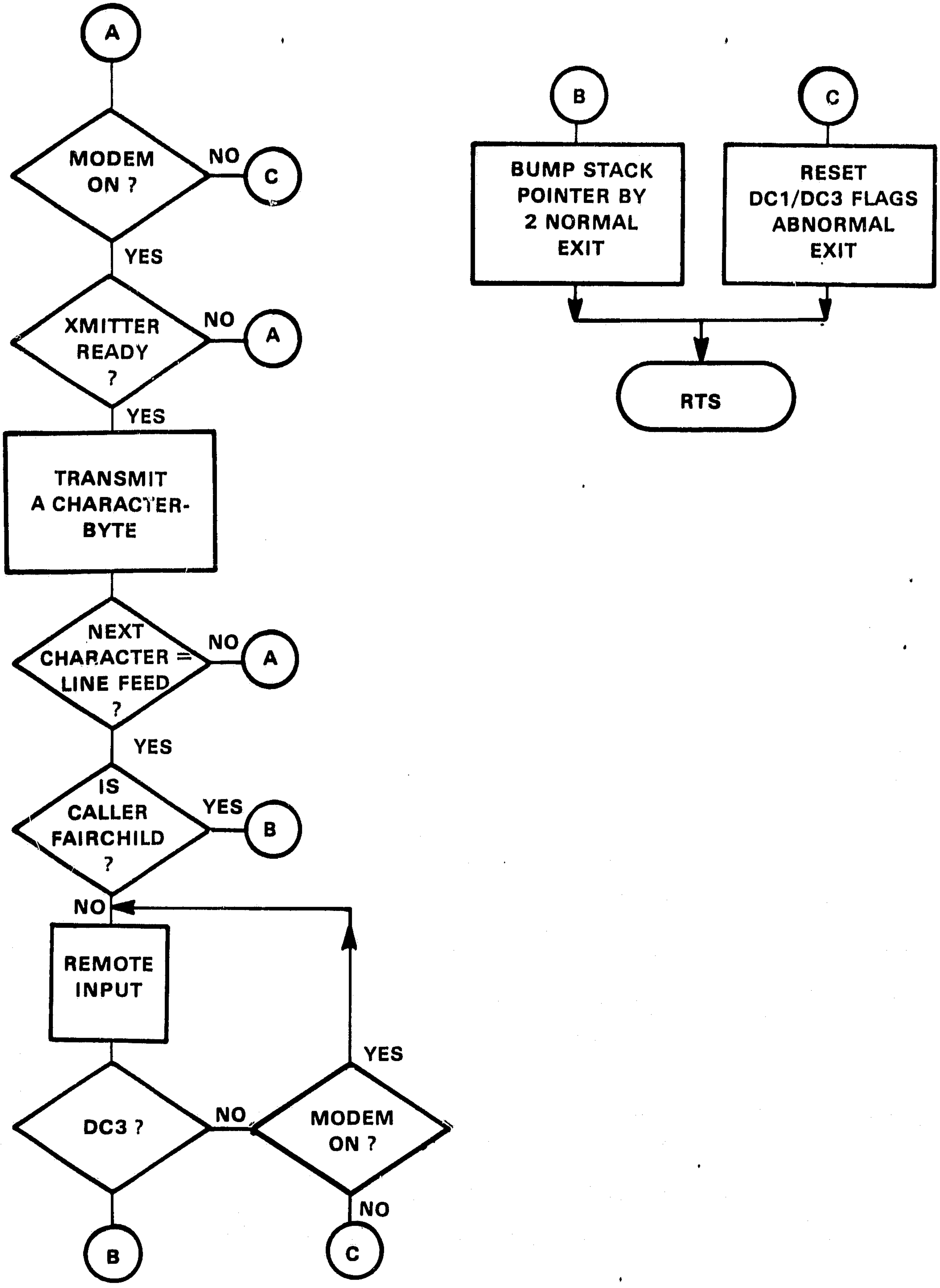

XMITT
IS CALLER FAIRCHILD ?
YES
D
NO
REMOTE INPUT
YES
DC1 ?
NO
MODEM ON ?
NO
C
YES
D
DUMP FLAG SET ?
NO
A
YES
LOAD CURRENT DAY TO DUMP STORAGE AREA
CLEAR DUMP DAY FLAG "ORCSR"
A


RCV
J
MODEM ON ?
NO
I
YES
RECEIVER READY
NO
J
READ A CHARACTER TO RECEIVER BUFFER
RECEIVER ERROR ?
YES
G
NO
CHARACTER = DC1 ?
YES
H
NO
CHARACTER = CARRIAGE RETURN ?
YES
H
NO
CHARACTER = RUB OUT ?
YES
CLEAR CHARACTER FROM RECEIVER BUFFER
J
NO
J
G
OVERRUN ERROR ?
NO
FRAMING ERROR ?
NO
PARITY ERROR ?
I
NO
YES
SET ERR = 4
SET ERR = 5
SET ERR = 6
H
H
H
H
BUMP STACK BY 2 FOR NORMAL EXIT
RTS
I
RESET DC1-DC3 FLAGS ABNORMAL EXIT
RTS


E2
PASS = 0 ?
YES
Z2
NO
PICKUP FWA OF RECEIVED BUFFER
XMITT
ECHO ERROR CHARACTERS
MODEM ON ?
YES
Z2
NO
EOUT
Z2
PICKUP ADDRESS OF ERR2 (WRONG FUNCTION)
3
PICKUP ADDRESS OF MSG2
CLEAR MSG FLG
T
E4
PICKUP ADDRESS OF ERR 4 (OVERRUN ERROR)
4
MSGFLG = 1 ?
YES
W
NO
MSGFLG = 2 ?
YES
3
NO
MSGFLAG = 3 ?
YES
5
NO
V
E3
PICKUP ADDRESS OF ERR 3 (WRONG PASSWORD)
5
PICKUP ADDRESS OF MSG 3
SET MSGFLG = 1
T
E5
PICKUP ADDRESS OF ERR5 (FRAMING ERROR)
E6
PICKUP ADDRESS OF ERR6 (PARITY ERROR)
4


P
ERR = 1 ?
YES
P1
NO
ERR = 2 ?
YES
E2
NO
ERR = 3 ?
YES
E3
NO
ERR = 4 ?
YES
E4
NO
ERR = 5 ?
YES
E5
NO
ERR = 6 ?
YES
E6
NO
MSGFLG = 1 ?
YES
P1
NO
MSGFLG = 2 ?
YES
E2
NO
P2
P2
MSGFLG = 3 ?
YES
V
NO
PASS = 0
NO
Z2
YES
E3
P1
PASS = 0 ?
NO
100
YES
PICKUP FWA OF RECEIVER BUFFER
XMITT
ECHO ERROR CHARACTERS
MODEM ON ?
NO
EOUT
YES
100
PICKUP ADDRESS OF ERR 1 MESSAGE (WRONG NUMBER)
W
PICKUP ADDRESS OF MESSAGE # 1
SET MSGFLG = −1
T


R
CLEAR PASSWORD FLAG
STATION = PLUMBROOK ?
YES
NO
FUNCTION = DIAGNOSTIC ?
YES
DIAG
V
NO
FUNCTION = DUMP & PURGE ?
NO
FUNCTION = CHANGE PASSWORD ?
NO
ERR = 2
BUMP PASSWORD FLAG
P
YES
STATION = PLUMBROOK ?
NO
V
TRAP FLAG > 0
NO
YES
YES
DISPLY
DISPLAY TRAP ERROR(S)
CHP5WD
V
NO
TRAP FLAG > 0
YES
DISPLY
DISPLAY TRAP ERROR
DPURGE
V


A
MODEM ON ?
NO
C
YES
XMITTER READY ?
NO
A
YES
TRANSMIT A CHARACTER-BYTE
NEXT CHARACTER = LINE FEED ?
NO
A
YES
IS CALLER FAIRCHILD ?
YES
B
NO
REMOTE INPUT
DC3 ?
NO
MODEM ON ?
YES
NO
B
C
B
C
BUMP STACK POINTER BY 2 NORMAL EXIT
RESET DC1/DC3 FLAGS ABNORMAL EXIT
RTS

PROGRAM: DISPLY

PURPOSE: Determine the type of system failure and transmit an appropriate message.

CALLING SEQUENCE: JSR PC,DISPLY

Upon entry, determines type of system failure then transmits an appropriate message to the telephone caller. This message will print prior to a channel dump, however, the dump will not be inhibited. The message(s) printed should be used when making decisions to troubleshoot the system via program DIAG.

The trap flag "TPERR" is reinitialized by DIAG.

Message Format:

TRAP VECTOR # - DESCRIPTION

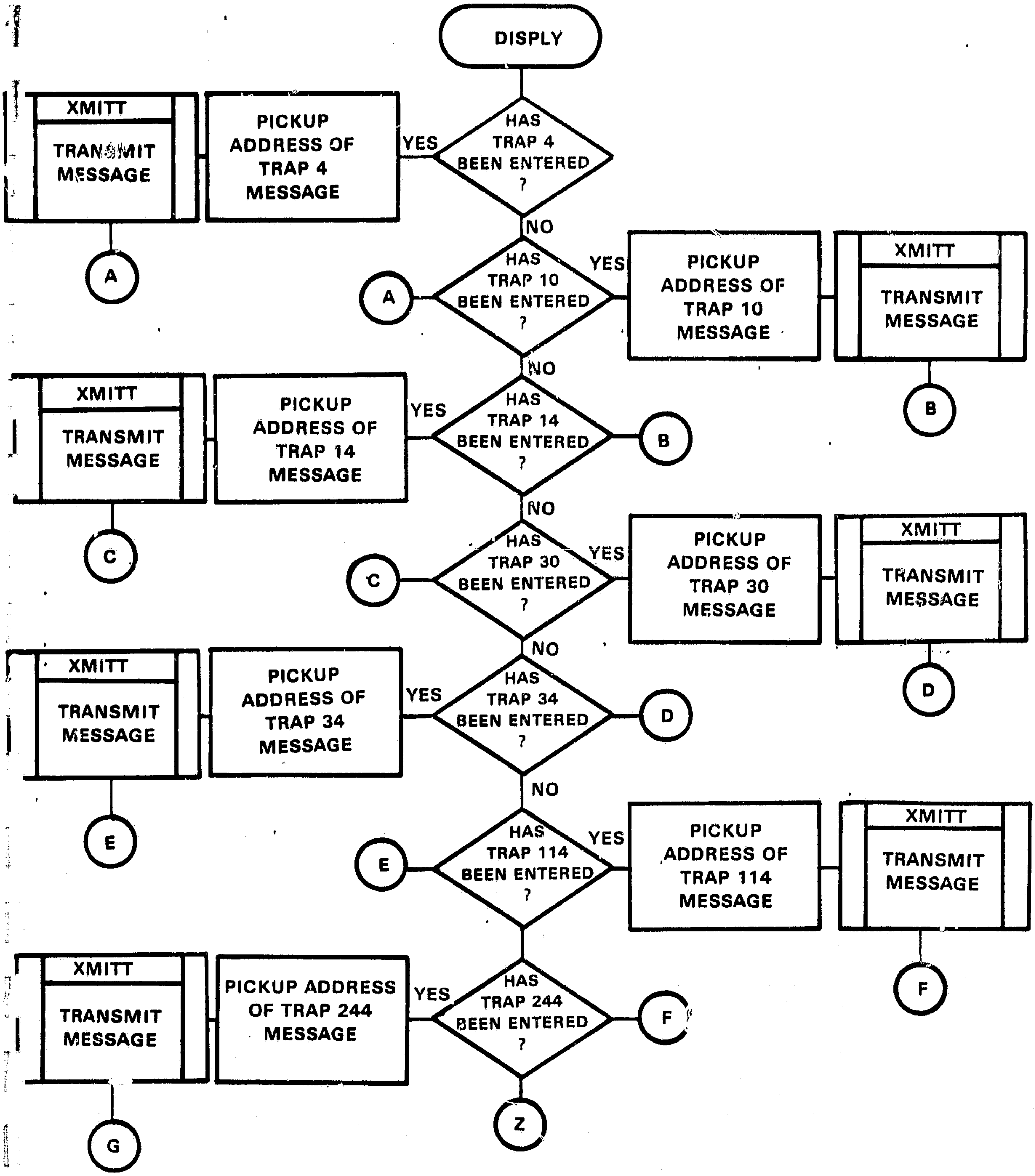
Special Requirements: None

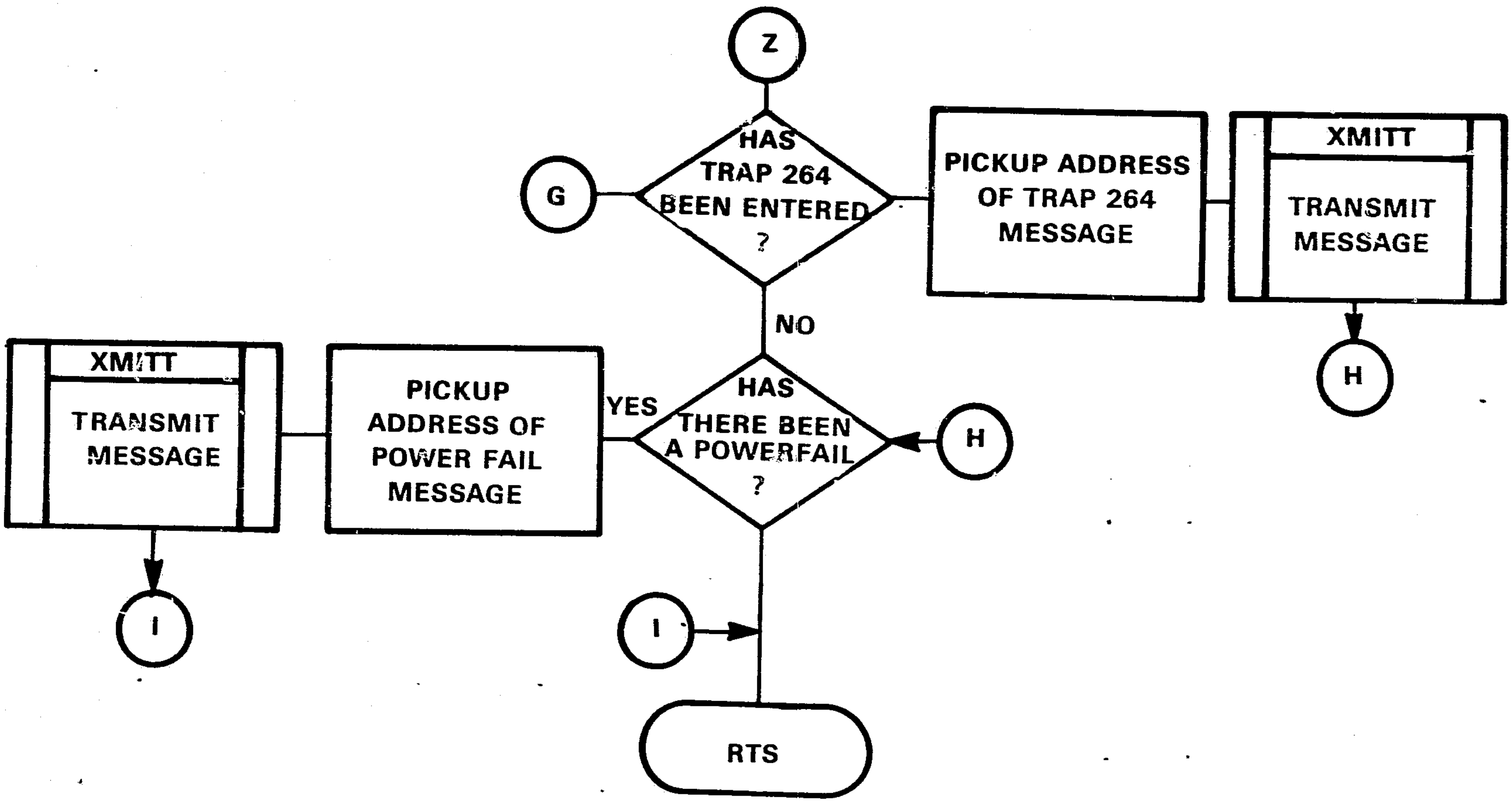
External Requirements: None

Globals: None

MODEM STORAGE AREA

MSG	/Who is calling please?/ CR LF
MSG2	/Please give function code/ CR LF
MSG3	/Please give password/ CR LF
MSG4	/Enter new password/ CR LF
ERR1	/Sorry wrong number/ CR LF
ERR2	/Sorry wrong function/ CR LF
ERR3	/Sorry wrong password/ CR LF
ERR4	/Overrun Error Character Lost/ CR LF
ERR5	/Framing error stop bit missing/ CR LF
ERR6	/Parity error require even parity/ CR LF
CODES	Station code characters
CODEF	Function code characters
MSGFLG	Message flag
ERR	Error number
PASS	Password error flag
SAVEB	FWA of password characters
FWPSWD	Fairchild Weston Password Storage
PLPSWD	Plumbrook Password Storage
PRG0	/Current Day= / CR LF
PRG1	/Buffer has been purged/ CR LF
DMPDAY	/Buffer to follow/ CR LF
CHN	/Channel / CR LF
F	/DU/ CR LF
U	/DUMP/ CR LF
NC	/DUMP AND PURGE CR LF
T	/DUMP PURGE/ CR LF
I	/CH/ CR LF
O	/DI/ CR LF
NS	/DP/ CR LF


DISPLY
HAS TRAP 4 BEEN ENTERED ?
YES
PICKUP ADDRESS OF TRAP 4 MESSAGE
XMITT
TRANSMIT MESSAGE
A
NO
A
HAS TRAP 10 BEEN ENTERED ?
YES
PICKUP ADDRESS OF TRAP 10 MESSAGE
XMITT
TRANSMIT MESSAGE
B
NO
HAS TRAP 14 BEEN ENTERED ?
B
YES
PICKUP ADDRESS OF TRAP 14 MESSAGE
XMITT
TRANSMIT MESSAGE
C
NO
C
HAS TRAP 30 BEEN ENTERED ?
YES
PICKUP ADDRESS OF TRAP 30 MESSAGE
XMITT
TRANSMIT MESSAGE
D
NO
HAS TRAP 34 BEEN ENTERED ?
D
YES
PICKUP ADDRESS OF TRAP 34 MESSAGE
XMITT
TRANSMIT MESSAGE
E
NO
E
HAS TRAP 114 BEEN ENTERED ?
YES
PICKUP ADDRESS OF TRAP 114 MESSAGE
XMITT
TRANSMIT MESSAGE
F
HAS TRAP 244 BEEN ENTERED ?
F
YES
PICKUP ADDRESS OF TRAP 244 MESSAGE
XMITT
TRANSMIT MESSAGE
G
Z


Z
HAS TRAP 264 BEEN ENTERED ?
G
PICKUP ADDRESS OF TRAP 264 MESSAGE
XMITT
TRANSMIT MESSAGE
H
NO
XMITT
TRANSMIT MESSAGE
PICKUP ADDRESS OF POWER FAIL MESSAGE
YES
HAS THERE BEEN A POWERFAIL ?
H
I
I
RTS

PROGRAMS: DIAG CHETR1/2 RONV PROBEM COPY DELETE DUMP - Diagnostic's

PURPOSE: To facilitate hardware and software maintenance via a remote communication link.

OPERATOR INITIATION: ;REF...DMODE1 Documentation

```
          Please enter function.          DI<CR>
          Please enter password.          80 characters max <CR>
```

CALLING SEQUENCE: JSR PC,DIAG

OVERVIEW

The diagnostic software is a segment of the ADV11A Data Acquisition software and facilitates hardware and software maintenance by detecting and isolating system malfunctions. In addition, the diagnostic software affords the system program manager with the capability of locating and correcting any errors in programs written for or supplied by the user. Diagnostic program execution for all routines except CHETR1/2, Boot and Monitor takes place on an as available basis in between interrupts. Therefore, data acquisitioning is not inhibited during program/operator I/O.

DIAG

DIAG is the program shell of the diagnostic operation, supervisor of all diagnostic subroutine execution and the intermediary of operator/program communication. In addition, DIAG performs housekeeping functions which:

(1) Optionally allow data acquisition start date and start time inputs to be made permanent on disk for program retrieval and display,
(2) Process and print statistical information pertinent to the data acquisitioning of up to 3 wind turbines,
(3) Redirect diagnostic program control to the RT11 operating system and
(4) Permit system bootstrapping.

PROGRAM/OPERATOR COMMUNICATION

PROGRAM: SELECT A DIAGNOSTIC

OPERATOR: Input one of the diagnostic names in the menu selection or type "MENU" if uncertain.

MENU SELECTION

Diagnostic Name	Comment
Prints	Print System Statistics
VCLOCK	Verify clock time increments and clock loading.
RONV	Display selected channels data.
CLEARP	Clear power out and system status flags.
PROBEM	Verify and/or correct system failures, or Load and/or execute PDP11 machine language instructions.
BOOT	Boot in new NASA Software System.
MONITOR	Return to RT11 System Level.
DATE - MM-DD-YY	Define data acquisition start date.
TIME - HH:MM:SS	Define data acquisition start time.
COPY	Copy files on disk.
DELETE	Delete files from system disk.
DUMP	Dump file blocks from disk to line.
MENU	Display this menu.
EXIT	Leave diagnostic mode.

PRINTS

PRINTS prints the following information to the remote terminal.

(1) Total amount of elapsed clock time since the start of data acquisition.
(2) Total amount of elapsed clock time during power outs.
(3) The number of the current data block being updated.
(4) The number(s) of the data blocks that have been purged.
(5) The current on/off status of up to 3 turbines.
(6) Messages identifying system failures.

CLEARP

CLEARP resets power out and system status flags to inhibit power fail and system failure messages that are printed to the remote terminal.

BOOT

BOOT reloades the RT11 operating system and the ADV11-A Data Acquisition Software.

MONITOR

MONITOR terminates Data Acquisitioning and transfers program control to the RT11 operating system at system level.

DATE/TIME

DATE and TIME inputs are stored on disk in the event of power outages. Once defined, the time and date should remain unchanged until data acquisition is reinitialized to Julian Day 1.

COPY

COPY performs basic disk file transfer operations for system software maintenance. The command line format is as follows:

```
*DISK:FILEOUT.EXT=DISK:FILEIN.EXT <CR>
```

DELETE

DELETE is a software maintenance process which deletes files from the system disk that are outdated or obsolete. The command line format is as follows:

```
*FILENAME.EXT <CR>
```

DUMP

DUMP utilizes the remote communications driver BB1.SYS and dumps file blocks from the system disk to modem line. The command line format is as follows:

```
*BB:FILEOUT.EXT=DISK:FILEIN.EXT <CR>
```

Files are transmitted to line in 512 decimal ASCII word blocks.

MENU

Prints the menu selection to the remote terminal.

EXIT

Terminates diagnostic operations and returns program control to the calling program.

DIAGNOSTIC SUBROUTINES

CHETR1/2

CHETR1 determines whether or not the interrupt service of the 50A-7621 system clock maintains a consistency without error at one second intervals over an infinite period of time.

CHETR2 determines whether or not the 50A-7621 system clock programs correctly when loaded under software control.

CHETR1/2 are executed in sequence when the clock verification function "VCLOCK" is requested by the operator. Both routines handle and disclose to the operator any errors found applicable to their operation.

Upon acknowledgement of errors CHETR1/2 disclose the following information.

CHETR1

```
Output message...                    Octal Data
          Clock SYL1 = ......................... ;minutes/seconds
          Computed SYL1 = ...................... ;minutes/seconds
          Clock SYL2 = ......................... ;days/hours
          Computed SYL2 = ...................... ;days/hours
```

Clock syllable 1/2 represents the current clock time taken from the software clock when the error occured. The computed syllables 1/2 represent the time computed by software based on an anticipated one second interval generated by the clocks 1 P.P.S. interrupt. When the clock syllables and computed syllables agree the 50A-7621 system clock is said to be incrementing correctly and providing a consistent one second interrupt pulse. When the syllables do not agree the messages shown above are disclosed to reflect the acknowledgement of an inconsistency in clock increments and interrupt pulses.

CHETR2

```
Output message...                Programming Table

            ****************************************************
            * Seconds = 14xx   minutes = 24xx   hours = 44xx   *
            *       Days = 104xx    Hun's Days = 204xx         *
            ****************************************************

                                   CLOCK ERROR!!
                                    Octal-Data
          Load Data = .........
          SYL1      = .........           ;minutes/seconds
          SYL2      = .........           ;days/hours
```

"LOAD DATA" represents the data written to the clock upon detection of an error. Syllable 1/2 represent the data read from the clock upon detection of an error. These error messages are disclosed when the data written to the clock is not equal to the data read.

The program table is printed to aid in diagnosing the clock error(s) and should be referenced when defining the octal value in the "octal data" column of the message.

CHETR1/2 PROGRAM TERMINATION

A termination sequence begins when the caller hangs up his/her phone. At the completion of the sequence the RT11 operating system is rebooted and the data acquisition initialized to start acquiring data for Day 1. The start date and start time must be redefined by the operator.

RONV - Display ADV11-A Data

The RONV diagnostic reads selected channels data from the ADV11-A analog to digital converter and displays same on a remote terminal device.
Upon entry, the following program/operator interchange is initiated.

PROGRAM: Enter channel number.
OPERATOR: Input the desired channel number in decimal, 16 channels max.
PROGRAM: How many samples this channel?
OPERATOR: Input any decimal number from 1 to 32767.

After the channel number and number of samples have been input RONV responds by printing the total number of samples requested for the channel. Each sample is printed in octal ASCII and represent the channels most recent data conversion. The samples printed are "raw data" with 12 significant bits.

At the completion of each channel data print segment the operator is permitted to continue by inputting another channel and sample count. The operator may terminate input by entering "EXIT" when the channel number is requested by the program.

PROBEM - Memory Probe

PROBEM performs functions similar to the RT11 Online Diagnostic Routine (ODT). Basically, PROBEM makes available a method by which the system manager can manipulate memory addresses and locations thereof for program testing and system analysis.

PROGRAM/OPERATOR communication begins when an "at" (@) character prompt is printed to the remote terminal. The operator need only enter the address of interest followed by a slash character (/) and the program displays the address contents. This address can be altered by entering an octal value on the same line followed by a carriage return or line feed. Any READ/WRITE location of memory may be displayed and/or changed. Any number of memory locations containing valid program instructions can be executed by entering a first word address followed by a "G" character. A "P" character may be entered to terminate the PROBEM operation.

The following examples illustrate program/operator communication.
FORMAT

```
        ADDRESS   CONTENTS
       @xxxxxx  /  xxxxxx
```

(1) Display the contents of memory location 1000_8.

```
       @1000 / 000416
```

(2) Change the contents of memory location 1000_8.

```
@1000 / 000416  207<CR>        ;before
@1000 / 000207  <CR>           ;after
@                              ;prompt after carriage return
```

(3) Display memory location 1000_8 through 1010_8.

```
@1000 / 000207  <LF>
@1002 / 40260   <LF>
@1004 / 000000  <LF>
@1006 / 000000  <LF>
@1010 / 000000  <CR>
@
```

(4) Change memory locations 1000_8 through 1002_8.

```
@1000 / 000207  240  <LF>      ;before
@1002 / 40260   207  <CR>      ;before
@1000 / 000240  <LF>           ;after
@1000 / 000207  <CR>           ;after
```

(5) Execute the program starting at memory address 1000_8.

```
@1000G
```

(6) Boot the RT11 operating system and restart the data acquisitioning.

```
@173000G
```

(7) Display the contents of the general purpose registers and set R2 equal to 10_8.

```
@R0 / 000121  <LF>
@R1 / 000000  <LF>
@R2 / 000000  10  <LF>
@R3 / 000002  <LF>
@R4 / 40260   <LF>
@R5 / 000001  <LF>
@R6 / 000776  <CR>
@
```

(8) Terminate the PROBEM operation and return to the caller.

```
@P
```

```
Select a diagnostic   ;output by program DIAG upon return.
```

NOTE: All memory and register changes are made permanent upon termination of the PROBEM operation.

SPECIAL CONDITIONS

The operator is required to enter a special access code when requesting execution of the following routines:

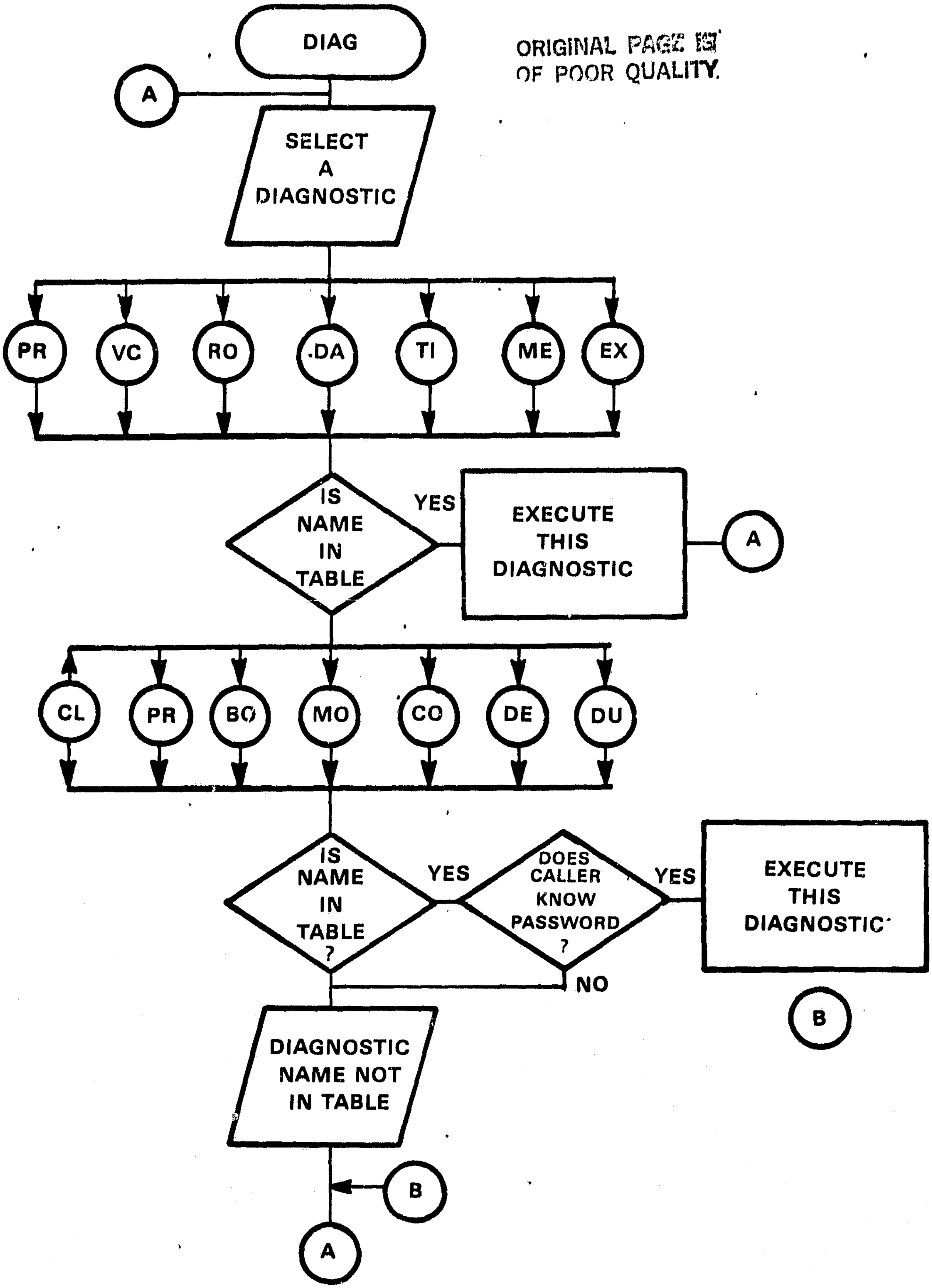
(1) CLEARP
(2) PROBEM
(3) BOOT
(4) MONITOR
(5) COPY
(6) DELETE
(7) DUMP

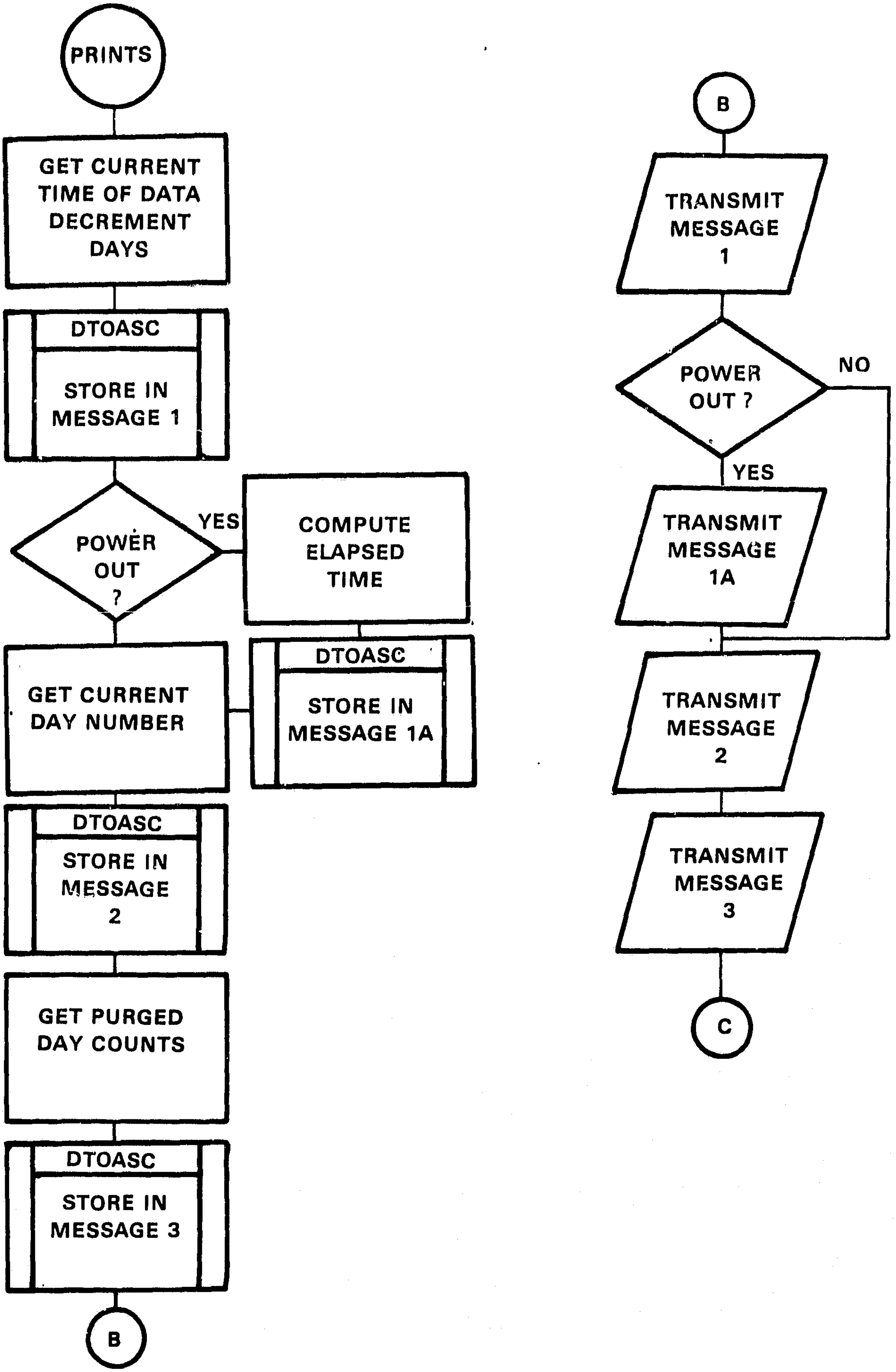
SPECIAL REQUIREMENTS

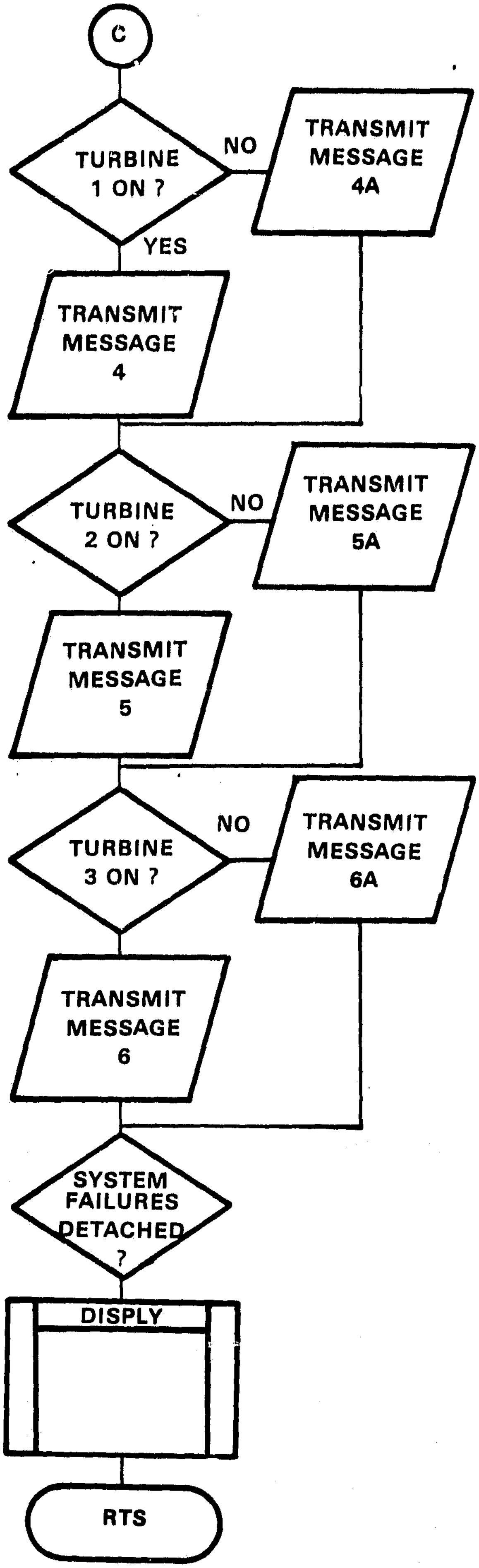
Data terminal must be set for 8 bits even parity.
ADV11A Data Acquisition Software.

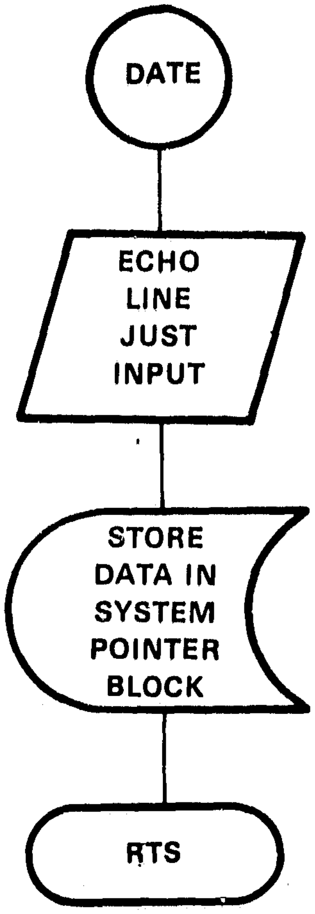
LIMITATIONS: None

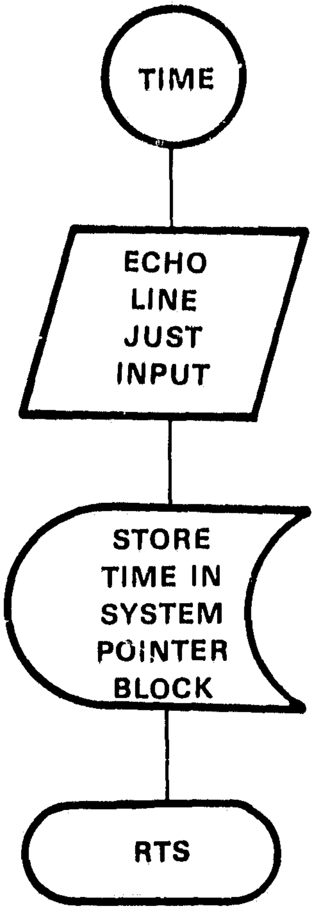
ORIGINAL PAGE IS
OF POOR QUALITY

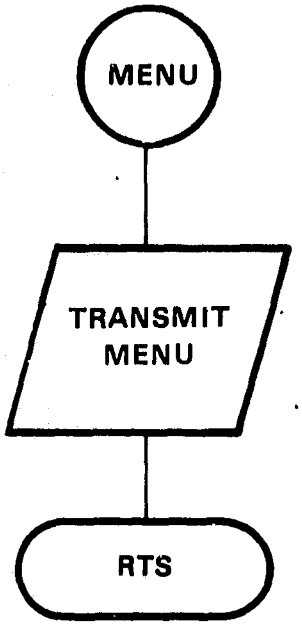


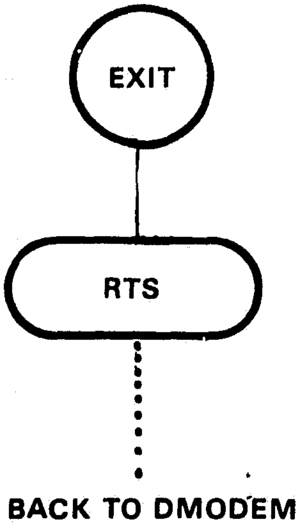

PRINTS
GET CURRENT TIME OF DATA DECREMENT DAYS
DTOASC
STORE IN MESSAGE 1
POWER OUT ?
YES
COMPUTE ELAPSED TIME
DTOASC
STORE IN MESSAGE 1A
GET CURRENT DAY NUMBER
DTOASC
STORE IN MESSAGE 2
GET PURGED DAY COUNTS
DTOASC
STORE IN MESSAGE 3
B
B
TRANSMIT MESSAGE 1
POWER OUT ?
NO
YES
TRANSMIT MESSAGE 1A
TRANSMIT MESSAGE 2
TRANSMIT MESSAGE 3
C

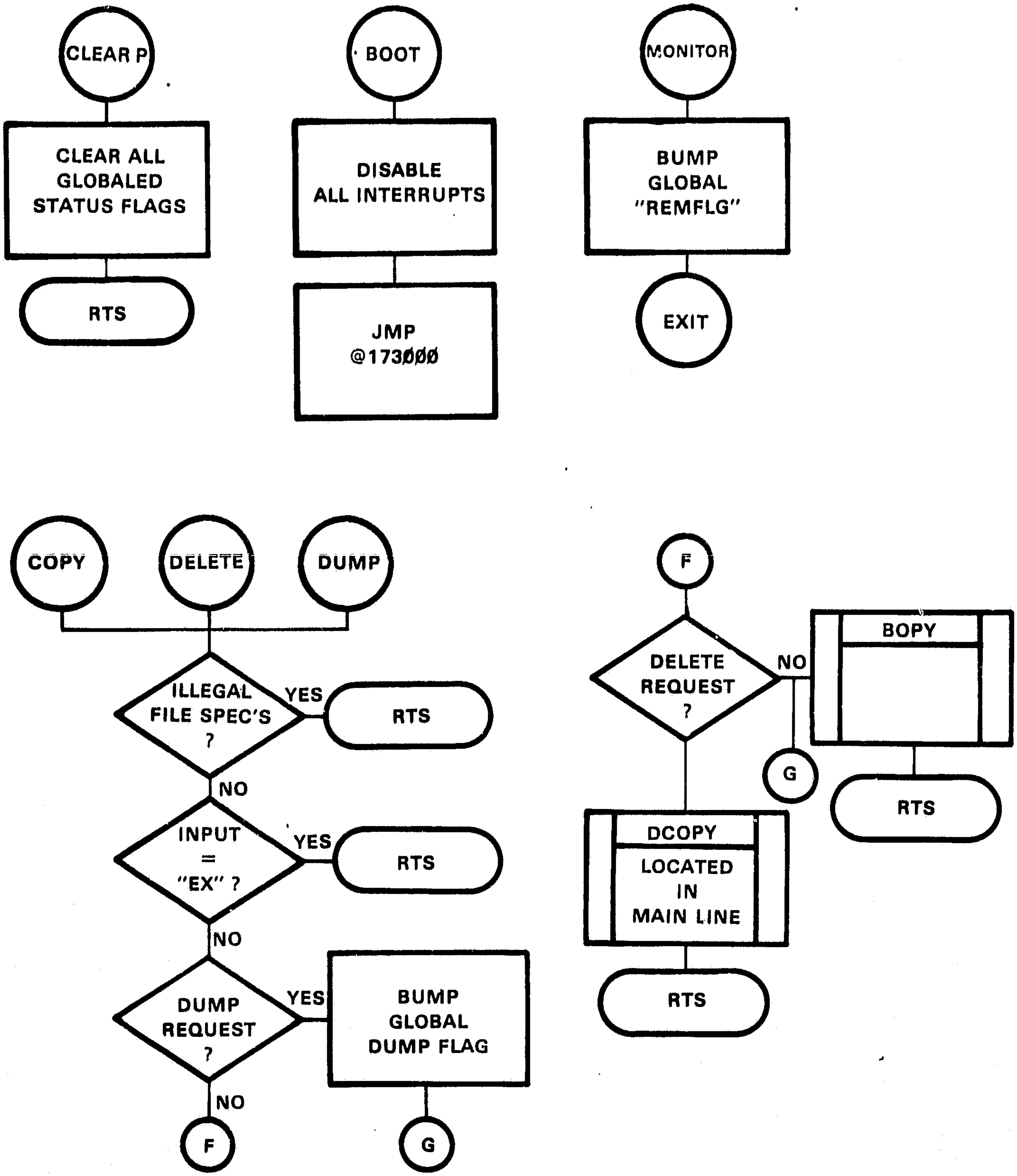

C
TURBINE 1 ON ?
NO
YES
TRANSMIT MESSAGE 4A
TRANSMIT MESSAGE 4
TURBINE 2 ON ?
NO
TRANSMIT MESSAGE 5A
TRANSMIT MESSAGE 5
TURBINE 3 ON ?
NO
TRANSMIT MESSAGE 6A
TRANSMIT MESSAGE 6
SYSTEM FAILURES DETACHED ?
DISPLY
RTS

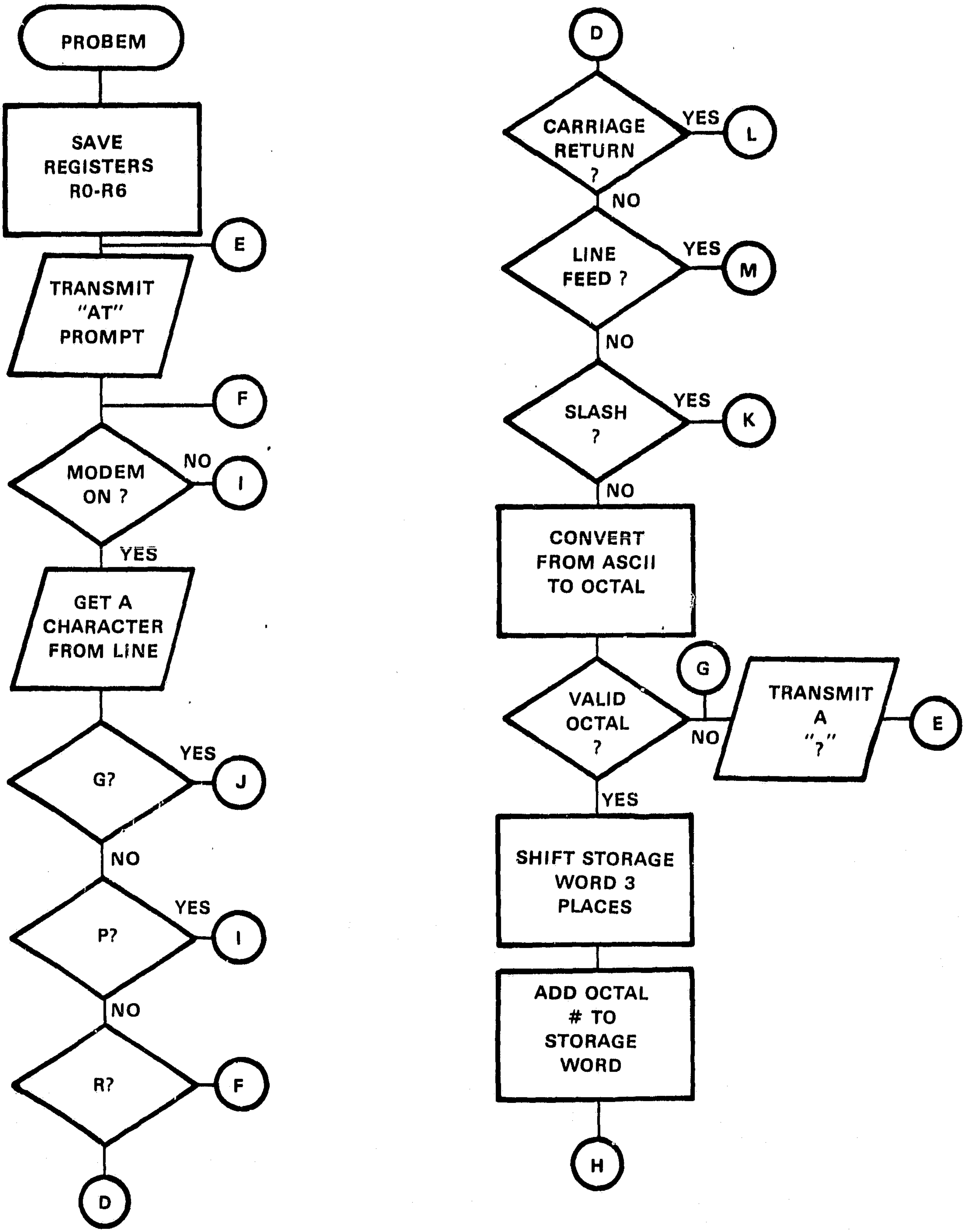

DATE
ECHO LINE JUST INPUT
STORE DATA IN SYSTEM POINTER BLOCK
RTS

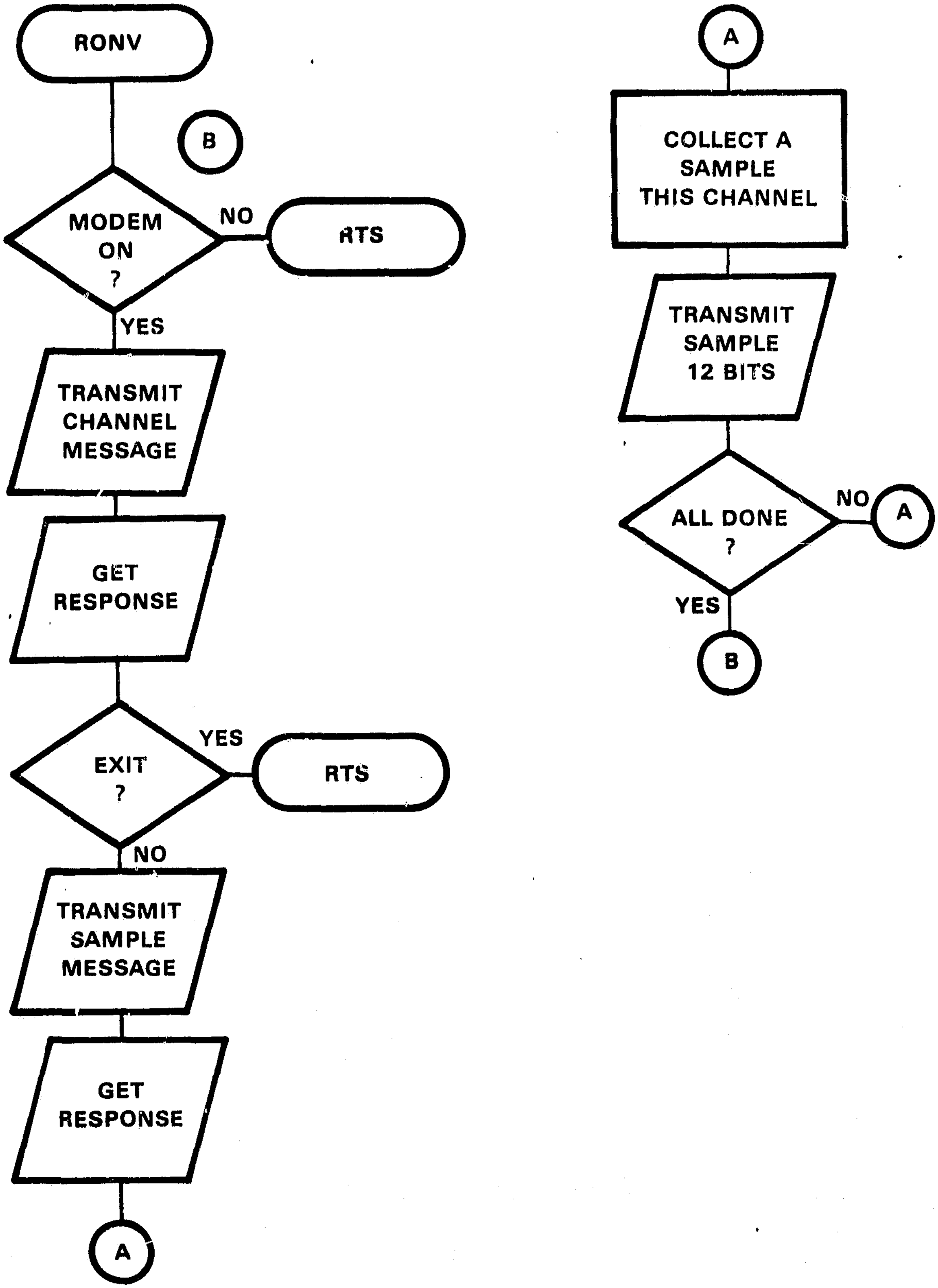

TIME
ECHO LINE JUST INPUT
STORE TIME IN SYSTEM POINTER BLOCK
RTS

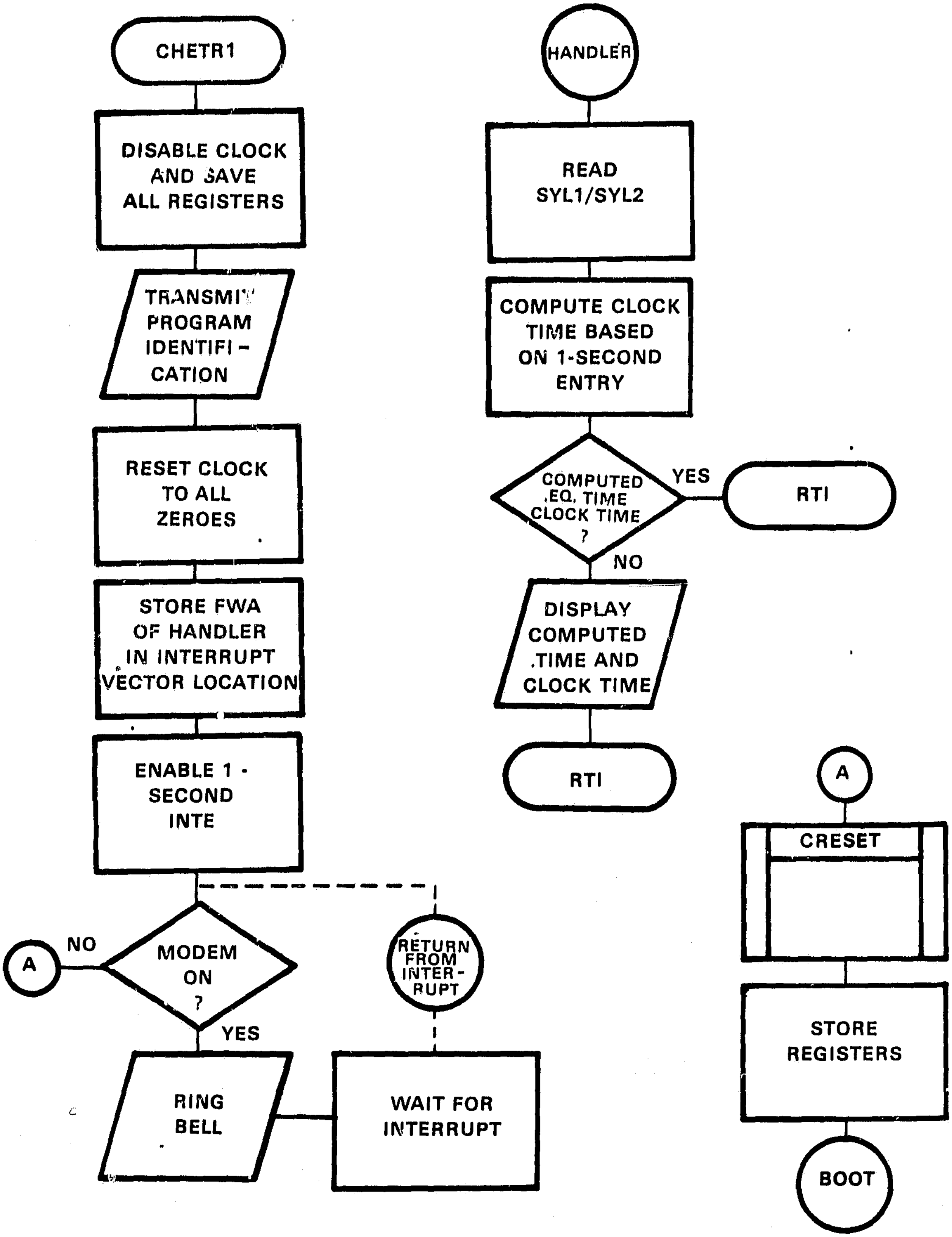

MENU
TRANSMIT MENU
RTS

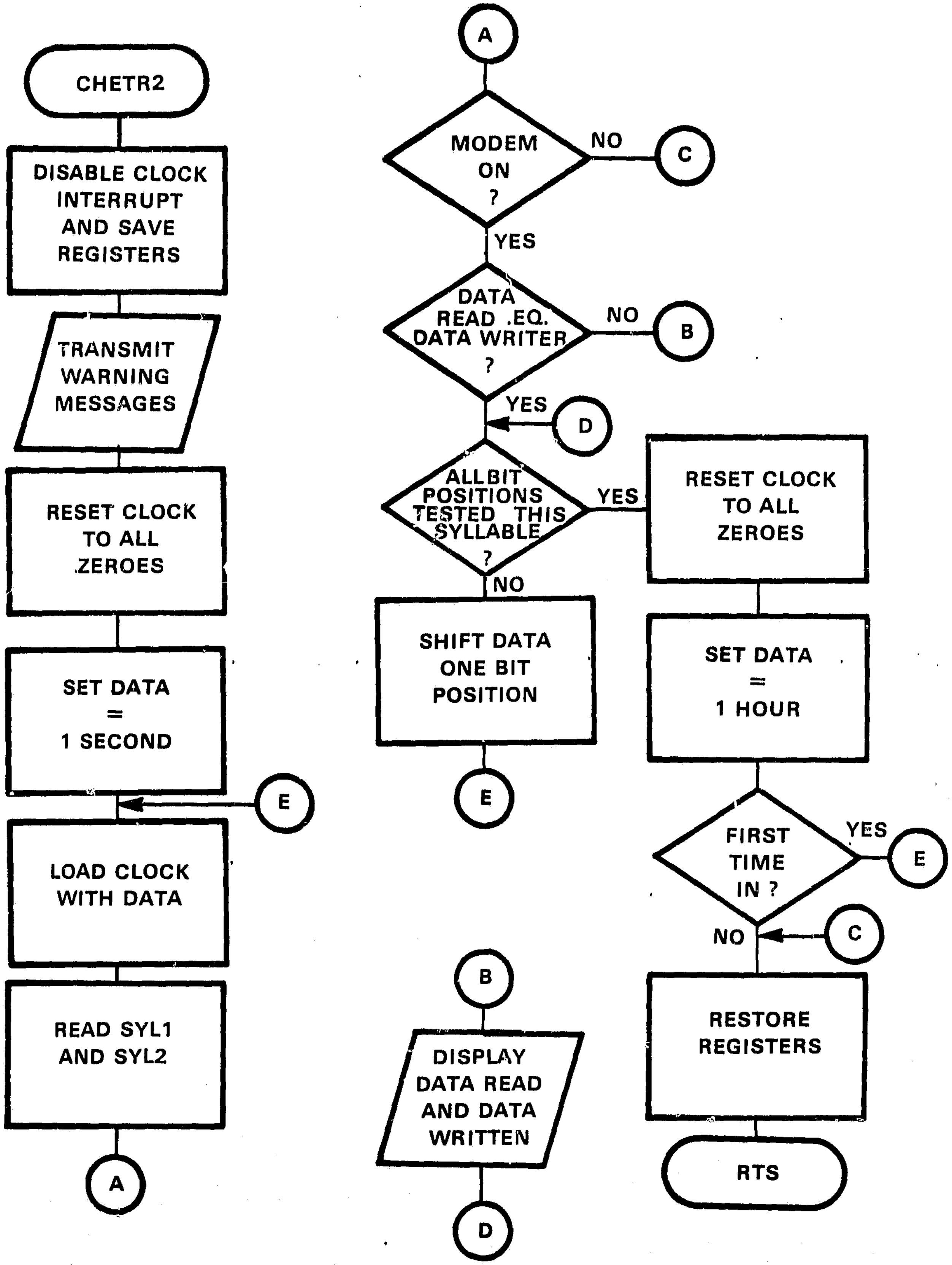

EXIT
RTS
BACK TO DMODEM


CLEAR P
CLEAR ALL GLOBALED STATUS FLAGS
RTS
BOOT
DISABLE ALL INTERRUPTS
JMP @173ØØØ
MONITOR
BUMP GLOBAL "REMFLG"
EXIT
COPY
DELETE
DUMP
ILLEGAL FILE SPEC'S ?
YES
RTS
NO
INPUT = "EX" ?
YES
RTS
NO
DUMP REQUEST ?
YES
BUMP GLOBAL DUMP FLAG
G
NO
F
F
DELETE REQUEST ?
NO
BOPY
RTS
G
DCOPY
LOCATED IN MAIN LINE
RTS


PROBEM
SAVE REGISTERS R0-R6
E
TRANSMIT "AT" PROMPT
F
MODEM ON ?
NO
I
YES
GET A CHARACTER FROM LINE
G?
YES
J
NO
P?
YES
I
NO
R?
F
D
D
CARRIAGE RETURN ?
YES
L
NO
LINE FEED ?
YES
M
NO
SLASH ?
YES
K
NO
CONVERT FROM ASCII TO OCTAL
VALID OCTAL ?
G
NO
TRANSMIT A "?"
E
YES
SHIFT STORAGE WORD 3 PLACES
ADD OCTAL # TO STORAGE WORD
H


RONV
B
MODEM ON ?
NO
RTS
YES
TRANSMIT CHANNEL MESSAGE
GET RESPONSE
EXIT ?
YES
RTS
NO
TRANSMIT SAMPLE MESSAGE
GET RESPONSE
A
A
COLLECT A SAMPLE THIS CHANNEL
TRANSMIT SAMPLE 12 BITS
ALL DONE ?
NO
A
YES
B


CHETR1
DISABLE CLOCK AND SAVE ALL REGISTERS
TRANSMIT PROGRAM IDENTIFI - CATION
RESET CLOCK TO ALL ZEROES
STORE FWA OF HANDLER IN INTERRUPT VECTOR LOCATION
ENABLE 1 - SECOND INTE
A
NO
MODEM ON ?
YES
RING BELL
WAIT FOR INTERRUPT
RETURN FROM INTER - RUPT
HANDLER
READ SYL1/SYL2
COMPUTE CLOCK TIME BASED ON 1-SECOND ENTRY
COMPUTED .EQ. TIME CLOCK TIME ?
YES
RTI
NO
DISPLAY COMPUTED .TIME AND CLOCK TIME
RTI
A
CRESET
STORE REGISTERS
BOOT


CHETR2
DISABLE CLOCK INTERRUPT AND SAVE REGISTERS
TRANSMIT WARNING MESSAGES
RESET CLOCK TO ALL ZEROES
SET DATA = 1 SECOND
E
LOAD CLOCK WITH DATA
READ SYL1 AND SYL2
A
A
MODEM ON ?
NO
C
YES
DATA READ .EQ. DATA WRITER ?
NO
B
YES
D
ALLBIT POSITIONS TESTED THIS SYLLABLE ?
YES
NO
SHIFT DATA ONE BIT POSITION
E
B
DISPLAY DATA READ AND DATA WRITTEN
D
RESET CLOCK TO ALL ZEROES
SET DATA = 1 HOUR
FIRST TIME IN ?
YES
E
NO
C
RESTORE REGISTERS
RTS

SECTION IV ADC CONTROLLER SOFTWARE

PROGRAM: ADV11A LOGIC1 LOGIC6

PURPOSE: Controls and maintains the sampling selection of one of sixteen channels of the ADV11-A analog to digital converter. Isolates data samples, selecting only those of interest and transfers same to various subprocess.

CALLING SEQUENCE: Entered 10 times per second via the system interrupt address 300_8.

Real Time Entry:

Upon acknowledgement of the external clock event, 10 times per second, all registers are saved, then set up to contain the following:

R0 = Converted data
R1 = FWA of next subprocess, turbine data
R2 = FWA of channel table
CHANNEL = Data Base Storage for Current Channel Number.

DATA REDUCTION AND ACQUISTION:

Figure 4.3A outlines sample tables consisting of subprocess addresses. The table in 4.3A allocates sixteen 16-bit memory locations, one location per address. Each address is referenced after the current channel data is (1) converted from serial to parallel, (2) reduced from 12 to 7 bits and (3) stored in R0.

DATA PROCESSING:

With each table reference, ADV11A transfers program control to a subprocess whose address is in an active location of the table. Location #1 in the table contains the FWA of a subprocess for channel #1 data processing, location 2, channel 2 data processing, and so on.

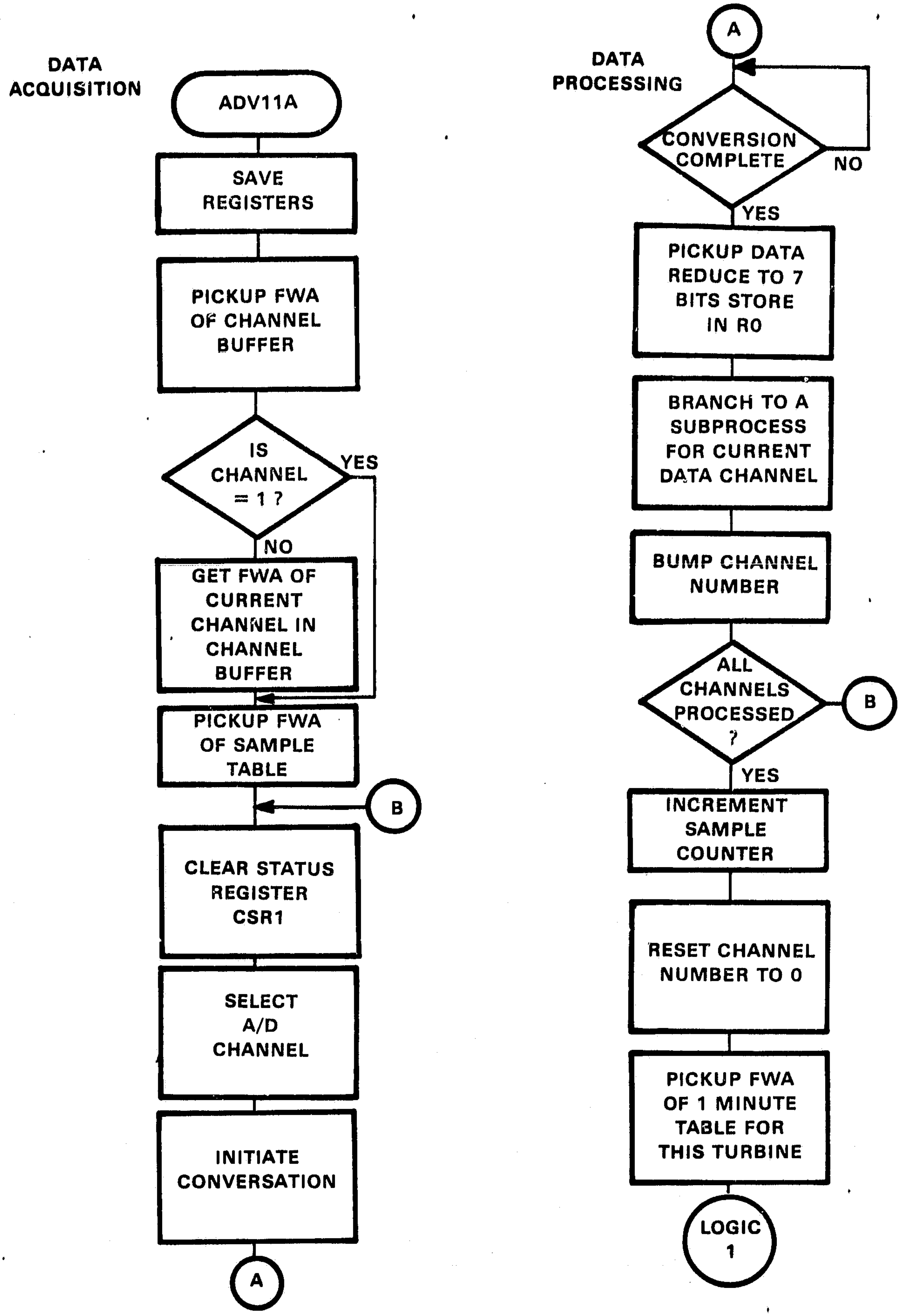
Upon return from each subprocess the current channel number and table pointer are incremented to point to the next channel number and associated subprocess. The channel number is compared with a maximum channel count (MAXCH) to determine whether all channels have been processed. In the event that all channels have not been processed, ADV11A repeats data reduction as stated above. If however, all channels have been processed ADV11A increments a sample counter (NSAMP) then reinitializes the channel number (CHANNEL). Thereafter, program control is transferred to LOGIC1.

NOTE: "NSAMP" is used by the subprocess AVGS and represents the number of samples processed per channel over a period of six minutes.

	Corresponding Channel
FWA OF AVGS	1
FWA OF AVGS	2
FWA OF AVGS	3
FWA OF AVGS	4
FWA OF AVGS	5
FWA OF AVGS	6
FWA OF AVGS	7
FWA OF AVGS	8
FWA OF AVGS	9
FWA OF UNITS	10
FWA OF AVGS	11
FWA OF AVGS	12
FWA OF AVGS	13
FWA OF AVGS	14
FWA OF AVGS	15
FWA OF UNITS	16

FIGURE 4.3A

Subprocess Table
For Sample Entries


DATA ACQUISITION
ADV11A
SAVE REGISTERS
PICKUP FWA OF CHANNEL BUFFER
IS CHANNEL = 1 ?
YES
NO
GET FWA OF CURRENT CHANNEL IN CHANNEL BUFFER
PICKUP FWA OF SAMPLE TABLE
B
CLEAR STATUS REGISTER CSR1
SELECT A/D CHANNEL
INITIATE CONVERSATION
A
A
DATA PROCESSING
CONVERSION COMPLETE
NO
YES
PICKUP DATA REDUCE TO 7 BITS STORE IN R0
BRANCH TO A SUBPROCESS FOR CURRENT DATA CHANNEL
BUMP CHANNEL NUMBER
ALL CHANNELS PROCESSED ?
B
YES
INCREMENT SAMPLE COUNTER
RESET CHANNEL NUMBER TO 0
PICKUP FWA OF 1 MINUTE TABLE FOR THIS TURBINE
LOGIC 1

TITLE: LOGIC1

PURPOSE: Transfers program control to predefined subprocesses whose objective is to process those channels data collected and saved during ADV11A data processing, performing various software functions at one minute intervals.

CALLING SEQUENCE:

```
MOV  #T1MIN,R1     ;    R1 = FWA of one minute subprocess entries
BR   LOGIC1
```

Real Time:

Upon entry the global NSAMP is compared with the minute count variable MINS to determine whether one minute has elapsed since the last entry. If one minute has not elapsed LOGIC1 restores those registers saved by ADV11A then issues a return from interrupt (RTI). If one minute has elapsed, NSAMP is reinitialized and program control is transferred to each subprocess in the T1MIN table, Figure 4.8A.

Upon completion of the subprocess transfers, the channel count (CHANNEL) is reinitialized and a check is made to determine whether six minutes has elapsed. In six minute intervals, LOGIC1 reinitializes the global MINCNT (one minute counter) then transfers program control to subroutine LOGIC6, otherwise LOGIC1 restores ADV11A registers then issues an RTI.

TITLE: LOGIC6

PURPOSE: Transfers program control to predefined subprocesses whose objective is to process and/or store those channels data collected, saved and preprocessed by ADV11A and LOGIC1. The memory image of the data block and the system pointer block is written to disk.

CALLING SEQUENCE:

```
MOV   #T1SIX,R1    ;    FWA of subprocess six minute entries
BR    LOGIC6
```

Real Time:

Upon entry, LOGIC6 transfers program control to the subprocesses whose FWA's are located in the first and subsequent active locations of table T1SIX (Six minute table, Figure 4.8B).

After completing the subprocess transfers, LOGIC6 reinitializes the channel counter (CHANEL), writes the data block to disk, then transfers program control to subroutine PSTAT (performs day rollover, housekeeping functions). Upon return from PSTAT, LOGIC6 prepares for its exit, restoring all registers saved by ADV11A, then reinitializing the data loss flag. Thereafter, an RTI is issued to return from interrupt.

NOTE: When telephone communciations are in progress the data block is written to disk only if day rollover occurs at the current 6 minute entry into LOGIC6. Otherwise, the memory image of the data block is updated and gets written to disk when telephone communications are terminated.

	Corresponding Channel
FWA OF RTSB	1
FWA OF RTSB	2
FWA OF RTSB	3
FWA OF RTSB	4
FWA OF RTSB	5
FWA OF RTSB	6
FWA OF RTSB	7
FWA OF RTSB	8
FWA OF RTSB	9
FWA OF UNITM	10
FWA OF RTSB	11
FWA OF RTSB	12
FWA OF RTSB	13
FWA OF RTSB	14
FWA OF RTSB	15
FWA OF UNITM	16

FIGURE 4.8A

Subprocess Table
For Minute Entries

	Corresponding Channel
FWA OF AVG6	1
FWA OF AVG6	2
FWA OF AVG6	3
FWA OF AVG6	4
FWA OF AVG6	5
FWA OF AVG6	6
FWA OF AVG6	7
FWA OF AVG6	8
FWA OF RTSB	9
FWA OF UNIT6	10
FWA OF AVG6	11
FWA OF AVG6	12
FWA OF AVG6	13
FWA OF AVG6	14
FWA OF AVG6	15
FWA OF UNIT6	16

FIGURE 4.8B

Subprocess Table
For Six Minute Entries

Special Conditions:

None

Special Requirements:

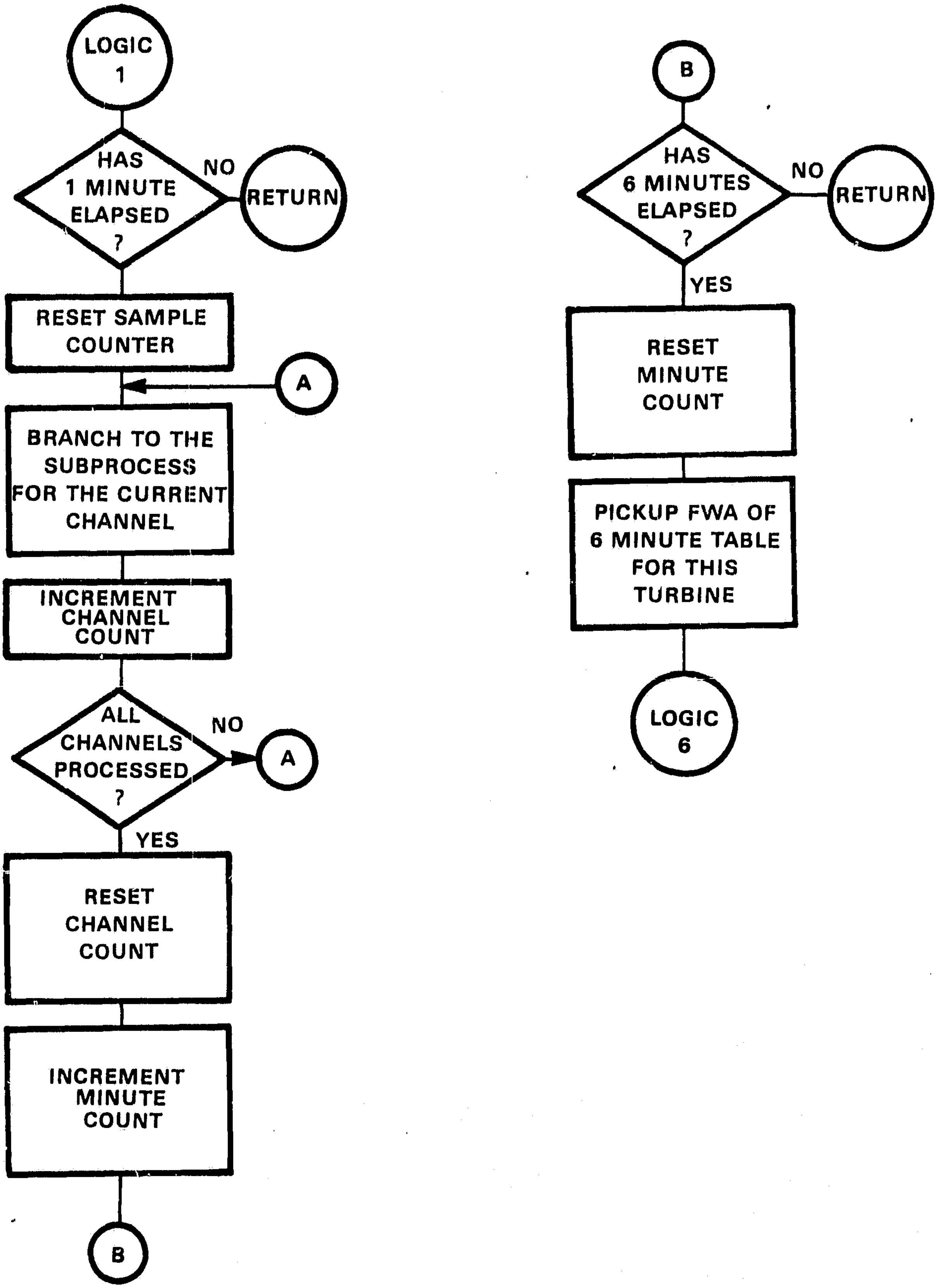
Digital (DEC) ADV11-A A/D Converter.

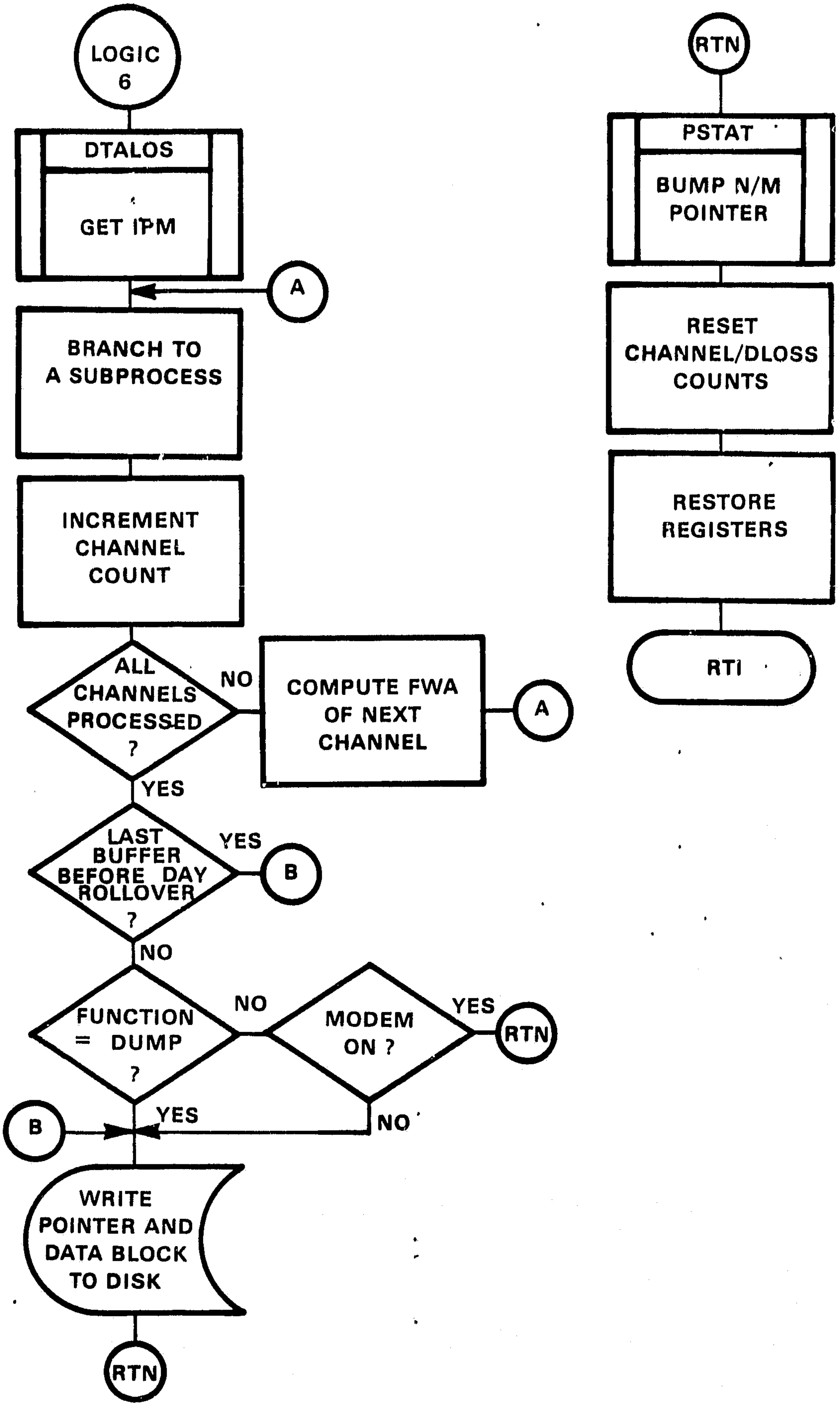
External Requirements:

DTALOS, PSTAT

Globals:

CALB, CALON, CALERR, CHANL, CHANNEL, MAXCH, MINS, MINCNT, TIMIN NSAMP, SAMP, SIX, T1SIX AND DLOSS.


LOGIC 1
HAS 1 MINUTE ELAPSED ?
NO
RETURN
RESET SAMPLE COUNTER
A
BRANCH TO THE SUBPROCESS FOR THE CURRENT CHANNEL
INCREMENT CHANNEL COUNT
ALL CHANNELS PROCESSED ?
NO
A
YES
RESET CHANNEL COUNT
INCREMENT MINUTE COUNT
B
B
HAS 6 MINUTES ELAPSED ?
NO
RETURN
YES
RESET MINUTE COUNT
PICKUP FWA OF 6 MINUTE TABLE FOR THIS TURBINE
LOGIC 6


LOGIC 6
DTALOS
GET IPM
A
BRANCH TO A SUBPROCESS
INCREMENT CHANNEL COUNT
ALL CHANNELS PROCESSED ?
NO
COMPUTE FWA OF NEXT CHANNEL
A
YES
LAST BUFFER BEFORE DAY ROLLOVER ?
YES
B
NO
FUNCTION = DUMP ?
NO
MODEM ON ?
YES
RTN
NO
YES
B
WRITE POINTER AND DATA BLOCK TO DISK
RTN
RTN
PSTAT
BUMP N/M POINTER
RESET CHANNEL/DLOSS COUNTS
RESTORE REGISTERS
RTI

PROGRAM: DTALOS

PURPOSE: Compute increment past midnight and look for data loss.
CALLING
SEQUENCE: JSR PC,DTALOS

Real Time Entry:

DTALOS is Executed every 6 minutes to compute the increment past midnight (IPM), The computed IPM is compared with the data block pointer "n" to determine whether a data loss occurred since the last entry. Upon recognition of a data loss the data loss flag "DLOSS" is set equal to a number greater than zero. This flag is monitored by processes at the six minute entries.

IPM - Computes as Follows:

IPM = (Hours x 10 + (minutes/6))
Hours x 10 = The total number of increments per hour.
Minutes/6 = The total number of increments per six minute interval within a period of one hour.

Special Conditions: None

External Requirements:

CLOCK, CRESET, TRBSEL, DTALOS

Globals:

CDAY, IDAY, JDAY, IFLAG, JFLAG, SAVEM, M, N, IPM, MAXCH, CHANEL, DLOSS, TIME

PROGRAM: DALOS

PURPOSE: Stores Data Loss word/s.
CALLING
SEQUENCE: SSR PC,DALOS

Real Time Entry: R5 = FWA of Day-M(n)

Upon entry, DALOS stores the data loss word (200$_8$) into the data base starting at the current and subsequent position of the pointer "n" until "n" is equal to the increment past midnight (IPM).

Special Conditions:

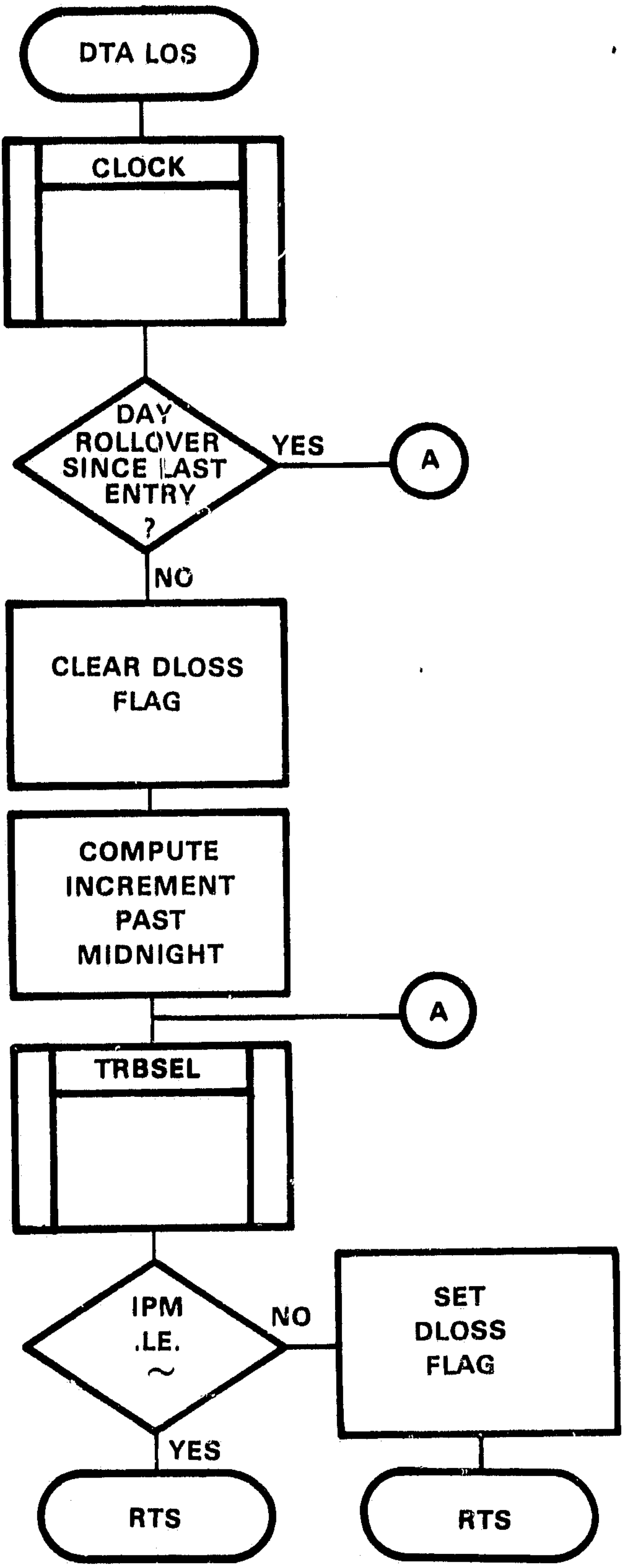
DALOS should be entered only when data losses are detected. The data loss flag "DLOSS" should be tested for > zero before entry.

Special Requirements: None

External Requirements: None

Globals:

IPM, N, DLOSS


DTA LOS
CLOCK
DAY ROLLOVER SINCE LAST ENTRY ?
YES
A
NO
CLEAR DLOSS FLAG
COMPUTE INCREMENT PAST MIDNIGHT
A
TRBSEL
IPM .LE. ~
NO
SET DLOSS FLAG
YES
RTS
RTS

PROGRAM: PSTAT

PURPOSE: Adjusts program variables

CALLING SEQUENCE: JSR PC,PSTAT

Real Time Entry:

Updates, if necessary, the current day counters, N pointer, external clock and FWA of the next day's data block. Also makes provision for leap year clock rollover.

Special Conditions: None

Special Requirements: None

External Requirements:

SCLOCK, TRBSEL

Globals

M, N, IDAY, JDAY, PURGE, IPM

PROGRAM: TRBSEL

PURPOSE: Establishes the FWA of the Data Block and purge flag for the current day.

CALLING SEQUENCE: JSR PC,TRBSEL

Real Time Entry:

Upon Entry, TRBSEL verifies the globaled "M" count to determine the "current day." Thereafter, general purpose register R5 is set equal to the FWA of the Data Block for that day. R5 is then adjusted to point to the current Data Storage location for A/D channel #1. The purge flag pointer is positioned at the current days purge flag location. In addition, the GLOBL "DBLOCK" is setup to contain the FWA of the command string for data block disk write functions (performed by BOPY).

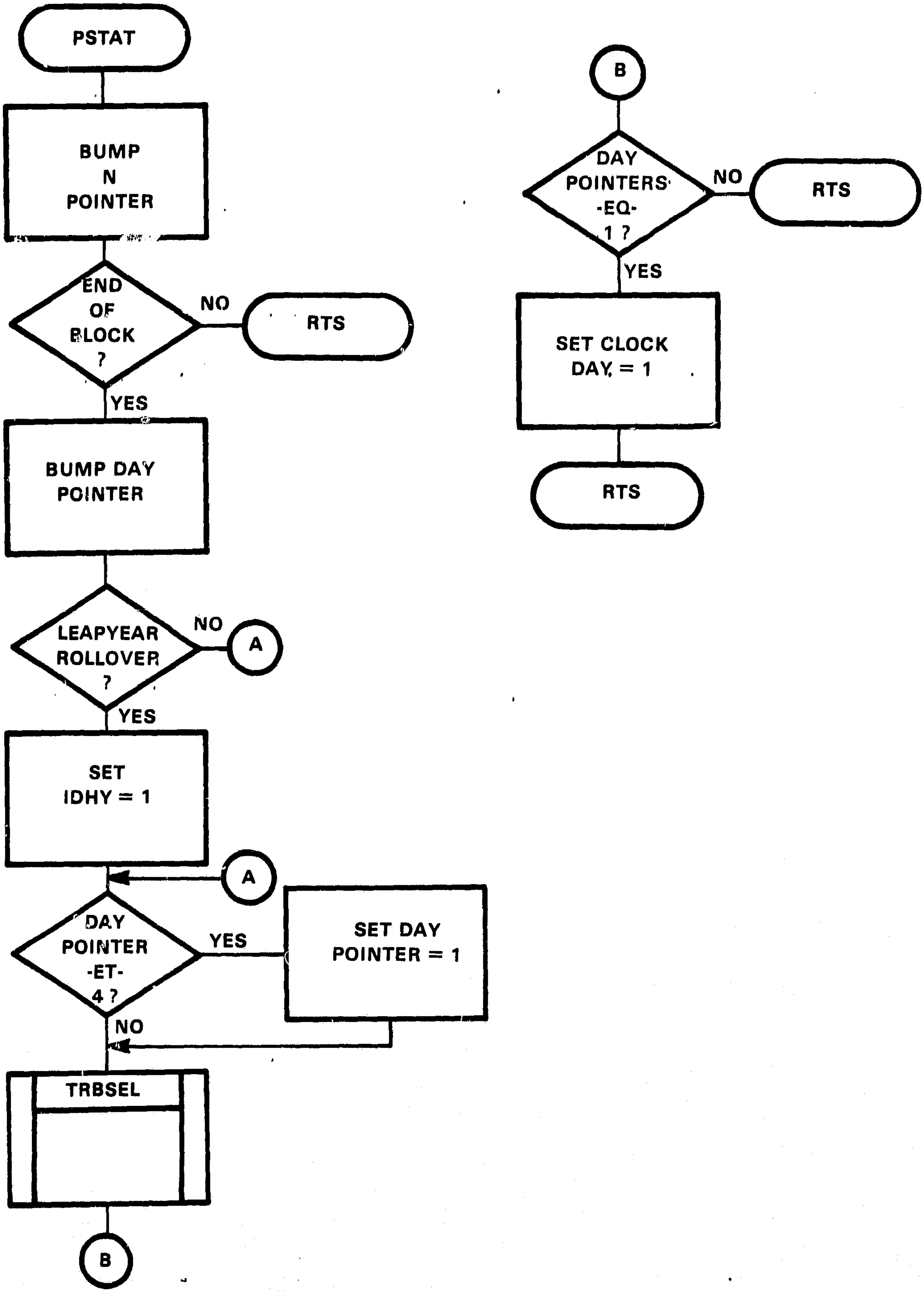
Special Conditions: None

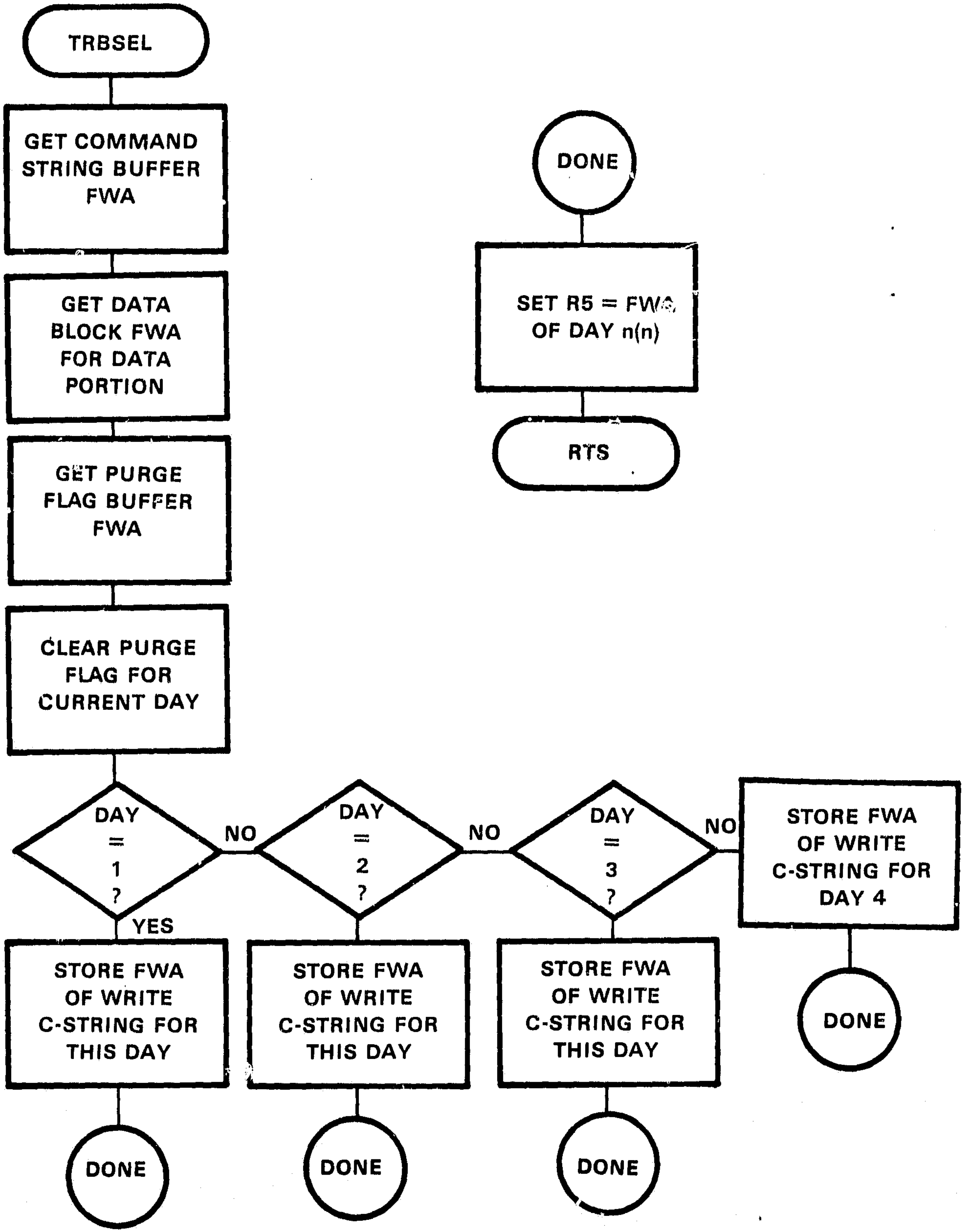
Special Requirements: None

External Requirements: None

Globals:

PURGE, M, N, ADC1M1


PSTAT
BUMP N POINTER
END OF BLOCK ?
NO
RTS
YES
BUMP DAY POINTER
LEAPYEAR ROLLOVER ?
NO
A
YES
SET IDHY = 1
A
DAY POINTER -ET- 4 ?
YES
SET DAY POINTER = 1
NO
TRBSEL
B
B
DAY POINTERS -EQ- 1 ?
NO
RTS
YES
SET CLOCK DAY, = 1
RTS


TRBSEL
GET COMMAND STRING BUFFER FWA
GET DATA BLOCK FWA FOR DATA PORTION
GET PURGE FLAG BUFFER FWA
CLEAR PURGE FLAG FOR CURRENT DAY
DAY = 1 ?
YES
NO
DAY = 2 ?
NO
DAY = 3 ?
NO
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE
STORE FWA OF WRITE C-STRING FOR DAY 4
DONE
DONE
SET R5 = FWA OF DAY n(n)
RTS

PROGRAM: CLOCK

PURPOSE: Updates the Data Base "TIME" locations, once every 6 minutes.

CALLING
SEQUENCE: JSR PC,CLOCK

Real Time Entry:

Upon entry, CLOCK sends a freeze command $(400)_8$ to the DRV11 output register. This freeze command switches the output data mux to syllable 1, minutes and seconds, and insures that the output data remains stable for at least 50 msec. After the freeze command is sent, syllable 1 is read from the DRV11 data register 167774 and saved. Thereafter, the DRV11 automatically switches the clock output data to syllable 2, hours and days, which is also read from register 167774 and saved. Syllable 1 and 2 are converted from BCD to Decimal then the Data Base Locations reserved for days, hours, minutes, and seconds are updated accordingly. In addition, the global CDAY is set equal to the current clock day. Program control is returned to the caller after the above procedures are complete.

CLOCK DESCRIPTION:

The external clock is fully programmable and provides BCD days, hours, minutes and seconds. The clock provides an external priority pulse to the CPU at the rate of 10SPS. It is battery powered to be immune from station power fluctuations.

Specifications:

1. Power requirements are .060 amps at 5v.DC, power provided by an internal power supply and battery backup.
2. Interfaces to an LSI/11 CPU via a DEC DRV11 parallel line unit.
3. Provides julian date or elapsed time to the DRV11 input port in two syllables; Syllable 1 is minutes and seconds, Syllable 2 is days and hours.
4. Time may be set to any valid time via the output port of the DRV11.
5. Provides two interrupt rates:

```
                                    Enable Interrupt

   (1)  REQ A at 10 P.P.S.      ;   MOV   #177770, R0
                                ;   MOVB  #100,(R0)
   (2)  REQ B at 1 P.P.S.       ;   MOV   #177770, R0
                                ;   MOVB  #10, (R0)
```

FORMAT OF TIME DATA READ FROM THE EXTERNAL CLOCK:

<table>
<tr><td></td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>Syllable 2</td><td colspan="9">DAYS</td><td colspan="7">HOURS</td></tr>
<tr><td>Syllable 1</td><td>X</td><td colspan="7">MINUTES</td><td>X</td><td colspan="7">SECONDS</td></tr>
</table>

Remote Programming:

	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
CLEAR 37000	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
FREEZE 400	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
SET SECONDS 14XX	0	0	0	0	0	0	1	1	0	X	X	X	X	X	X	X
SET MINUTES 24XX	0	0	0	0	0	1	0	1	0	X	X	X	X	X	X	X
SET HOURS 44XX	0	0	0	0	1	0	0	1	0	0	X	X	X	X	X	X
SET DATS 104XX	0	0	0	1	0	0	0	1	X	X	X	X	X	X	X	X
SET HUNDS DAYS 2040X	0	0	1	0	0	0	0	1	0	0	0	0	0	0	X	X

Programming Example:

```
                    SETDAY:  MOV   #177772,R0   ;  R0 = FWA of DRV11 output register.
                             MOV   #37000,(R0)  ;  Clear days, hours, minutes and
                                                   seconds.
                             MOV   #10401,(R0)  ;  Set clock Day = 1, Hours, minutes
                                                   and seconds remain unchanged.
                            .END
```

Special Conditions:

In addition to updating the data base "TIME" locations, CLOCK also processes and stores the current time into the data base "PTIME" locations. These time locations are used when computing the total amount of elapsed time during power out.

Special Requirements:

DRV11 Parallel Line Unit, FWS 56A-7621 Battery Powered External Clock.

External Requirements:

BCDDEC

Globals:

SRO, TIME, DRIBUF, DCAY, DRCSR, DROBUF, IDAY

PROGRAM: ECLOCK SCLOCK CRESET

PURPOSE: Enable, Set, and Reset the external clock.

CALLING SEQUENCE:

```
JSR  PC,ECLOCK
JSR  PC,SCLOCK
JSR  PC,CRESET
```

Real Time Unit:

ECLOCK: Enables the external clock interrupt by placing a '1' bit in the 7th bit of the clock control status register.

SCLOCK: Sets the external clock equal to Day 1, 0 Hours, 0 Minutes, 0 Seconds.

CRESET: Clears the GLOBL "LDFLG" and writes system pointer block to disk.

Special Conditions: None

Special Requirements:

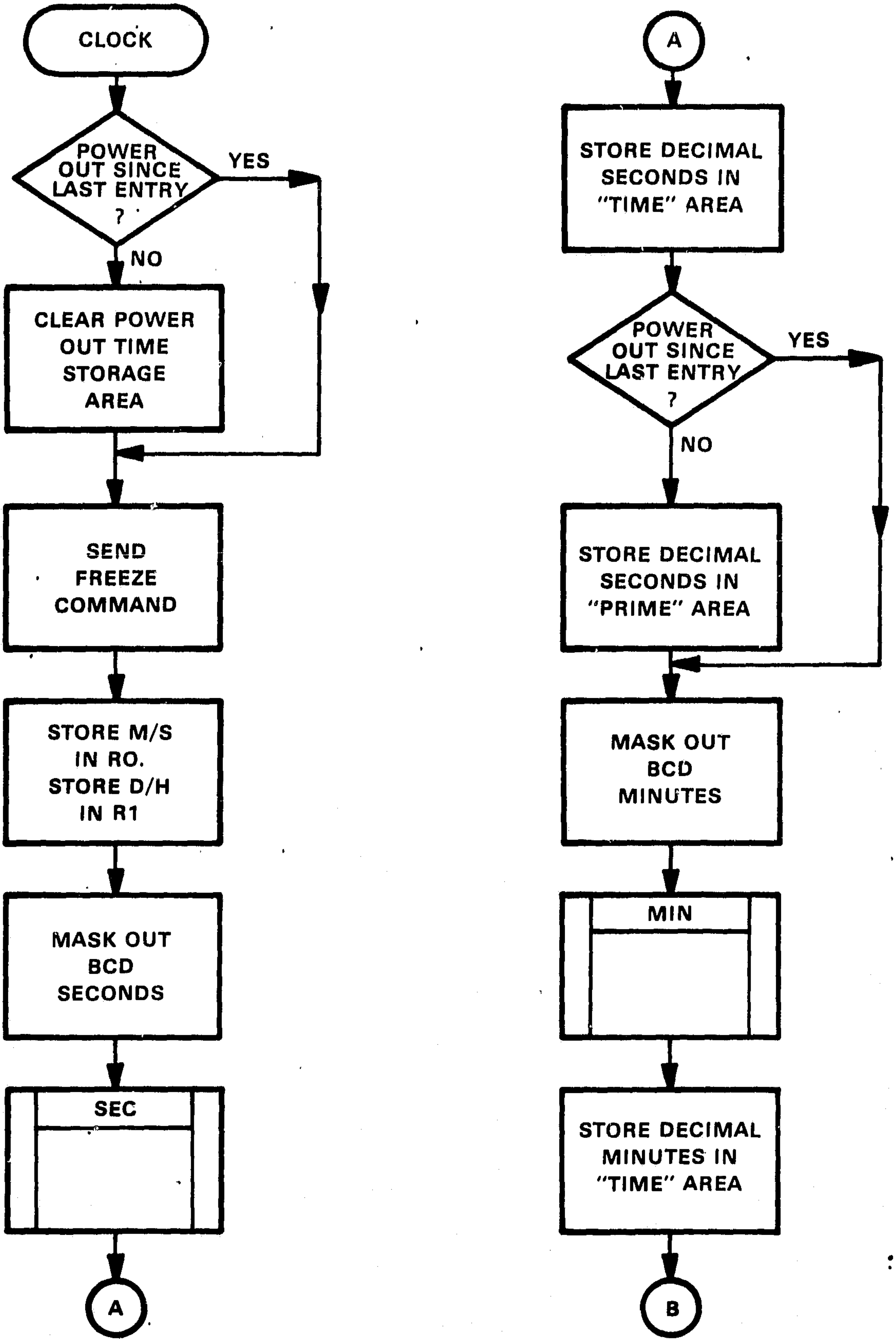
FWS 56A-7621 Battery Powered External Clock, DLV11 Parallel Line Unit.

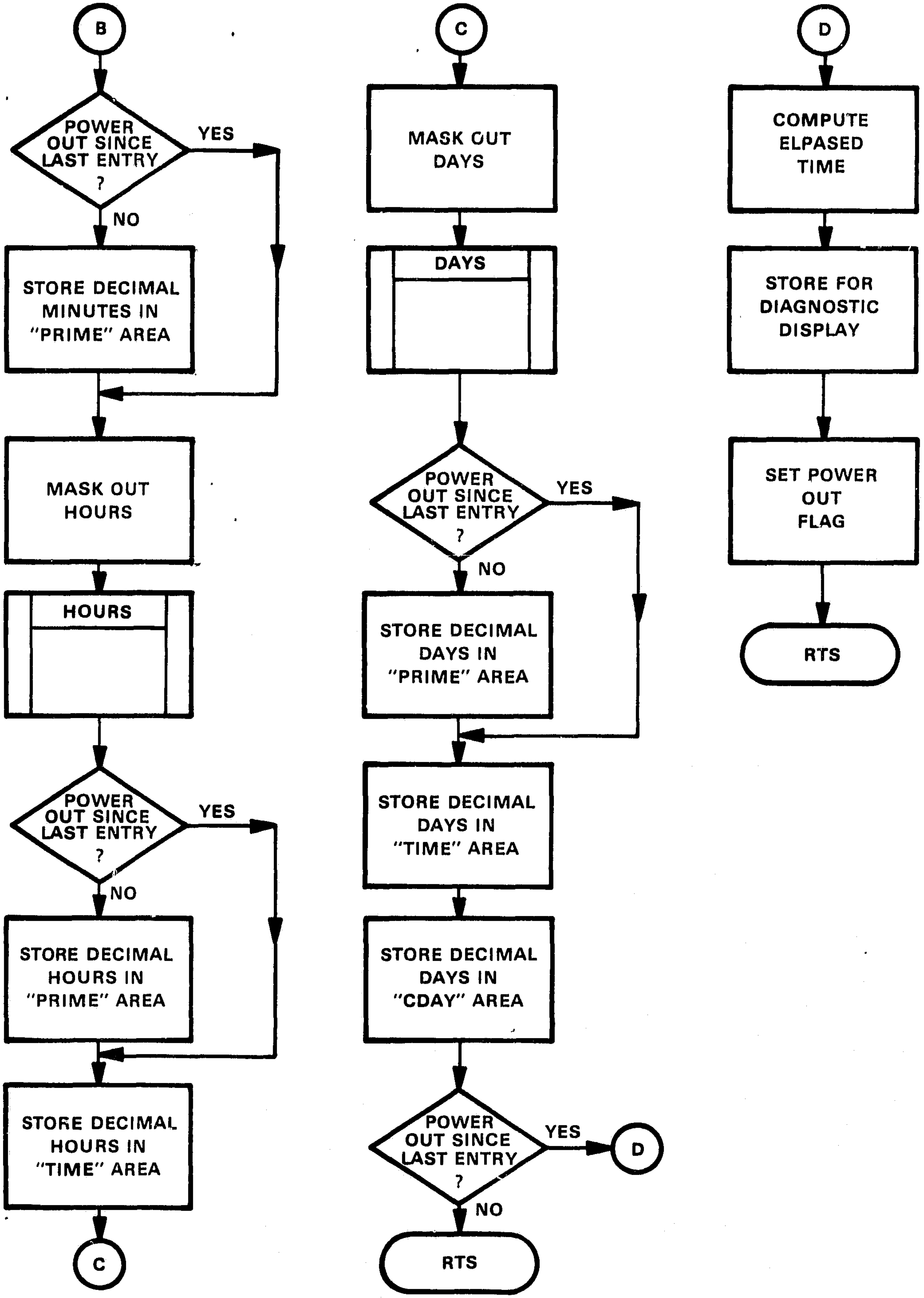
External Requirements: None

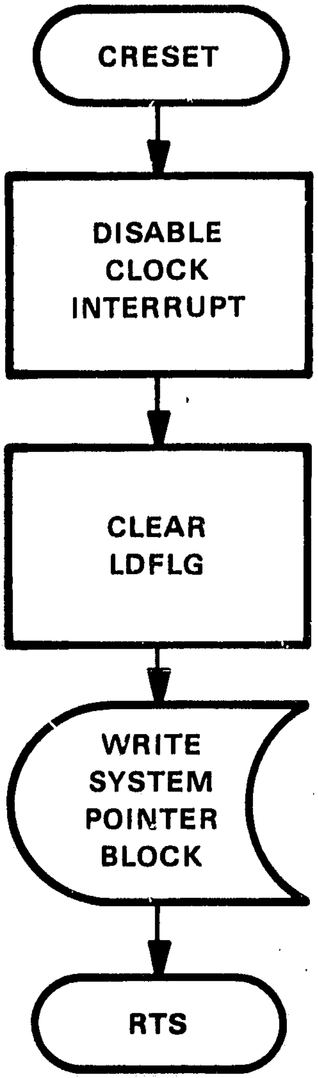
Globals:

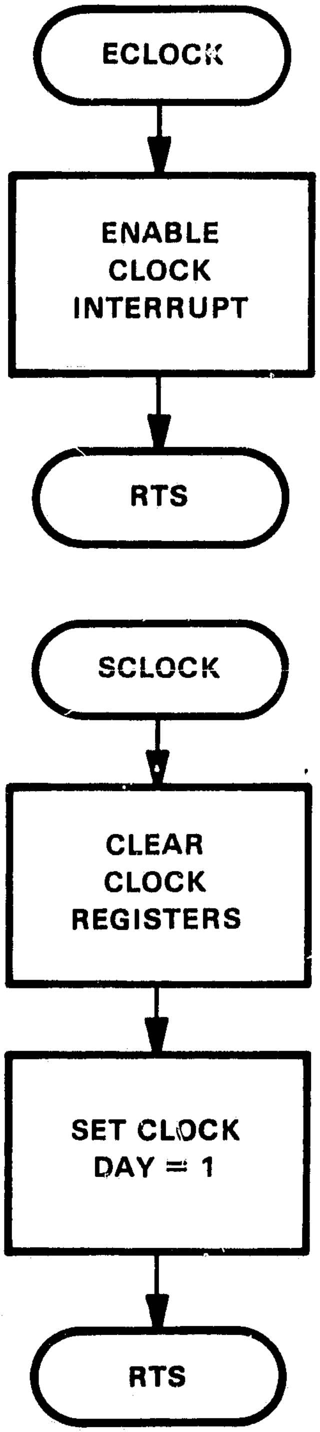
DROBUF, DRSCR, IDAY

ORIGINAL PAGE IS
OF POOR QUALITY


CLOCK
POWER OUT SINCE LAST ENTRY ?
YES
NO
CLEAR POWER OUT TIME STORAGE AREA
SEND FREEZE COMMAND
STORE M/S IN RO. STORE D/H IN R1
MASK OUT BCD SECONDS
SEC
A
A
STORE DECIMAL SECONDS IN "TIME" AREA
POWER OUT SINCE LAST ENTRY ?
YES
NO
STORE DECIMAL SECONDS IN "PRIME" AREA
MASK OUT BCD MINUTES
MIN
STORE DECIMAL MINUTES IN "TIME" AREA
B


B
POWER OUT SINCE LAST ENTRY ?
YES
NO
STORE DECIMAL MINUTES IN "PRIME" AREA
MASK OUT HOURS
HOURS
POWER OUT SINCE LAST ENTRY ?
YES
NO
STORE DECIMAL HOURS IN "PRIME" AREA
STORE DECIMAL HOURS IN "TIME" AREA
C
C
MASK OUT DAYS
DAYS
POWER OUT SINCE LAST ENTRY ?
YES
NO
STORE DECIMAL DAYS IN "PRIME" AREA
STORE DECIMAL DAYS IN "TIME" AREA
STORE DECIMAL DAYS IN "CDAY" AREA
POWER OUT SINCE LAST ENTRY ?
YES
D
NO
RTS
D
COMPUTE ELPASED TIME
STORE FOR DIAGNOSTIC DISPLAY
SET POWER OUT FLAG
RTS


CRESET
DISABLE CLOCK INTERRUPT
CLEAR LDFLG
WRITE SYSTEM POINTER BLOCK
RTS


ECLOCK
ENABLE CLOCK INTERRUPT
RTS
SCLOCK
CLEAR CLOCK REGISTERS
SET CLOCK DAY = 1
RTS

```
PROGRAM:    BCDDEC
CALLING
SEQUENCE:   MOV  DAYS,R0
            MOV  R0,SR0
            JSR  PC,DAYS
               or
            MOV  HOURS,R0
            MOV  R0,SR0
            JSR  PC,HOURS
               or
            MOV  MIN,R0
            MOV  R0,SR0
            JSR  PC,MIN
               or
            MOV  SEC,R0
            MOV  R0,SR0
            JSR  PC,SEC
```

Real Time Entry:

Upon entry,
R0 = BCD Day, hours minutes or seconds (LSB of R0).
SR0 = Globaled storage area set equal to R0.
Converts days, hours, minutes, and/or seconds from BCD to decimal.

Globals:

SR0

```
PROGRAM:    BOPY - Disk I/O Handler

PURPOSE:    Used by ADV11A software to transfer data blocks from memory to disk
            and vice versa.
CALLING
SEQUENCE:   DBLOCK = 1014; FWA of command string
            MOV  #Cstring, DBLOCK; store your string address.
            JSR  PC,BOPY
CSTRING:   .ASCIZ  /BB:FILEOUT.EXT=DX:FILEIN.EXT/
```

Upon entry, the command string supplied by the caller is transferred to the MACRO Command String Interpreter (CSI). CSI interrogates the command string for legal and illegal file specifications. When CSI determines that the command string is in an acceptable RT11 format .LOOKUP and/or .ENTER request are performed on the files. CSI aborts the operation and returns to the caller when illegal file specs are found.

Memory location 1012 contains error status set and cleared by BOPY. This memory location should be monitored by the caller after each call to BOPY. Before exiting BOPY location 1012 will be set to zero if the operation is successful, less than zero if a write error is detected or greater than zero if a read error is detected.

After it has been determined that the command string supplied by the user is in a legal RT11 format BOPY performs a copy operation using synchronous I/O, and the output file is made permanent on the device.

Read/write operations are performed on channels 3 and 0 respectively and are transferred in 512 decimal word blocks.

SPECIAL CONDITIONS

Any driver defined in the command string must be resident on the system storage medium.

When linking, BOPY must be the last object in the link string.

SPECIAL REQUIREMENTS

MACRO library must be resident on the system storage medium before linking.

EXTERNAL REQUIREMENTS

.CSIGEN, .READW, .WRITW, .CLOSE, .SRESET, BASE

GLOBALS

DSPACE, DEXT, SPACE, OUTSP, INSP, DBASE

ASSEMBLY

BOPY=BOPY,BASE

PROGRAM: BASE

PURPOSE: Used to define storage allocation for all data base variable and constants.

ASSEMBLY: MACRO
*BOBY=BOBY,BASE

The module base contains the data base variable described below. The variables names used in the data base are all globled and should not be duplicated by newly incorporated software routines.

GENERAL:

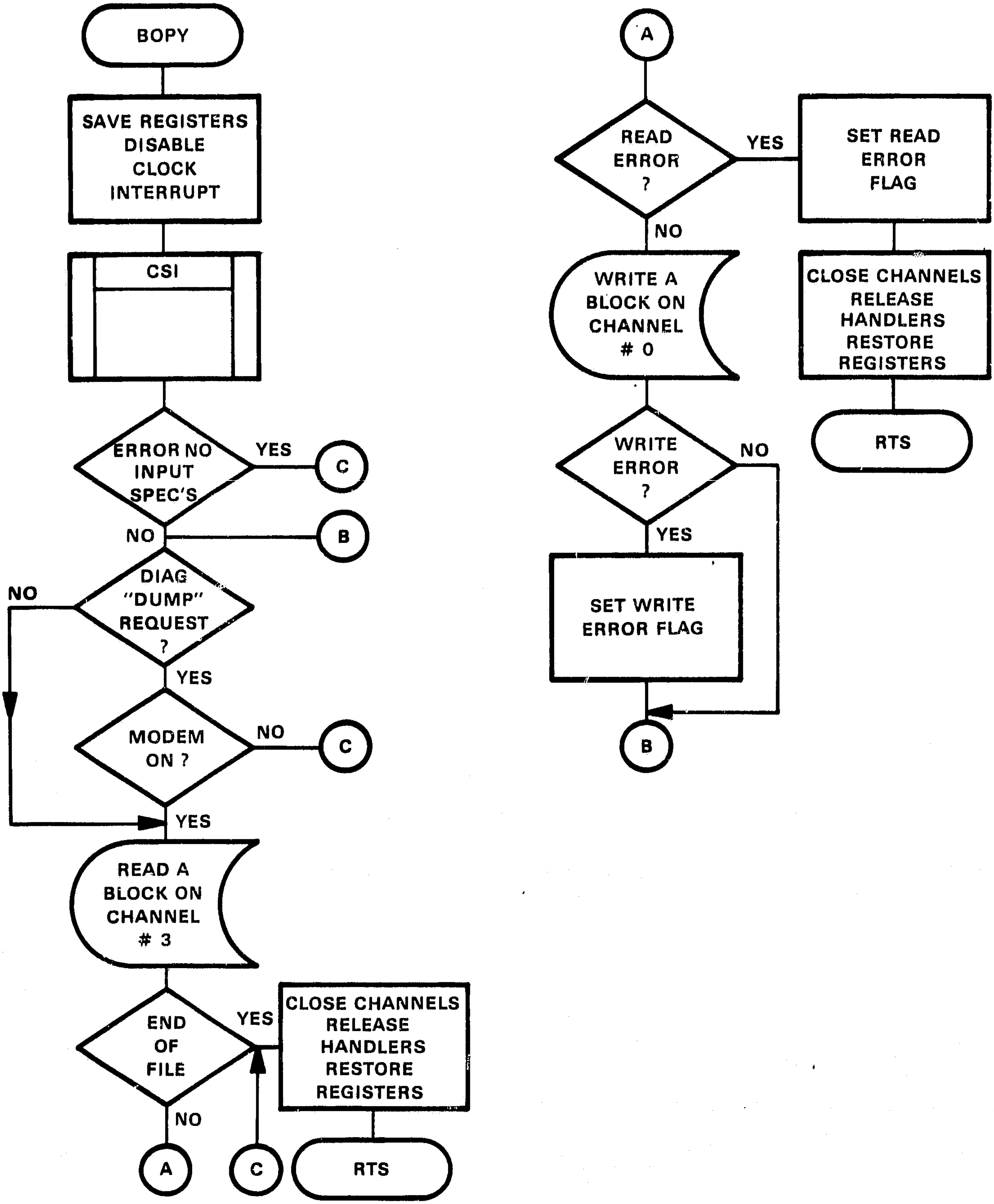
The data base allocates 8K words of memory and is comprised of the following:

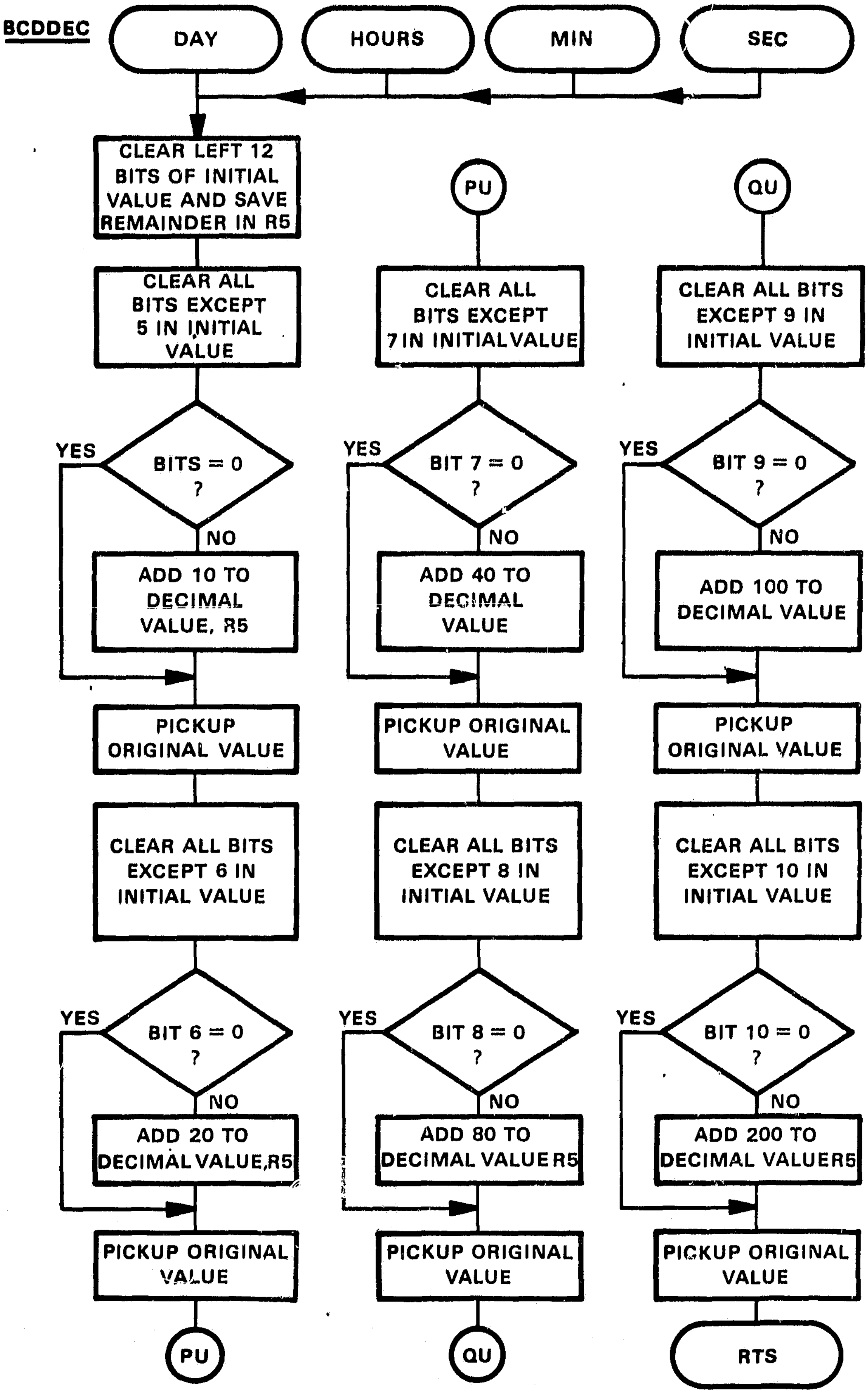
1. System pointers, program variables that identify relative memory locations within the data block.

2. System flags, program variables that assist in the decision making process of the incorporated software routines.

3. System counters, program variables that establish and retain points of reference during data processing and program execution.

4. Disk storage allocation. Data blocks, four (4) total, comprised of memory storage locations, which allocate enough space to store data for up to 16 A/D channels consecutively for four days. The composite size for each block (Data Only) is 3840K words of memory of which 240K bytes are reserved per channel. In addition, 128K bytes of storage is made available for system pointers.


BOPY
SAVE REGISTERS DISABLE CLOCK INTERRUPT
CSI
ERROR NO INPUT SPEC'S
YES
C
NO
B
DIAG "DUMP" REQUEST ?
NO
YES
MODEM ON ?
NO
C
YES
READ A BLOCK ON CHANNEL # 3
END OF FILE
YES
CLOSE CHANNELS RELEASE HANDLERS RESTORE REGISTERS
NO
A
C
RTS
A
READ ERROR ?
YES
SET READ ERROR FLAG
NO
WRITE A BLOCK ON CHANNEL # 0
CLOSE CHANNELS RELEASE HANDLERS RESTORE REGISTERS
RTS
WRITE ERROR ?
NO
YES
SET WRITE ERROR FLAG
B


BCDDEC
DAY
HOURS
MIN
SEC
CLEAR LEFT 12 BITS OF INITIAL VALUE AND SAVE REMAINDER IN R5
CLEAR ALL BITS EXCEPT 5 IN INITIAL VALUE
YES
BITS = 0 ?
NO
ADD 10 TO DECIMAL VALUE, R5
PICKUP ORIGINAL VALUE
CLEAR ALL BITS EXCEPT 6 IN INITIAL VALUE
YES
BIT 6 = 0 ?
NO
ADD 20 TO DECIMAL VALUE,R5
PICKUP ORIGINAL VALUE
PU
PU
CLEAR ALL BITS EXCEPT 7 IN INITIALVALUE
YES
BIT 7 = 0 ?
NO
ADD 40 TO DECIMAL VALUE
PICKUP ORIGINAL VALUE
CLEAR ALL BITS EXCEPT 8 IN INITIAL VALUE
YES
BIT 8 = 0 ?
NO
ADD 80 TO DECIMAL VALUE R5
PICKUP ORIGINAL VALUE
QU
QU
CLEAR ALL BITS EXCEPT 9 IN INITIAL VALUE
YES
BIT 9 = 0 ?
NO
ADD 100 TO DECIMAL VALUE
PICKUP ORIGINAL VALUE
CLEAR ALL BITS EXCEPT 10 IN INITIAL VALUE
YES
BIT 10 = 0 ?
NO
ADD 200 TO DECIMAL VALUER5
PICKUP ORIGINAL VALUE
RTS

Main memory storage allocation allocates enough storage for two data blocks. The first is used for current day data storage and the second is used for "DUMP" data storage. In addition, 128K bytes of storage have been added to maintain system pointers and program variables. This block is stored on disk every 6 minutes and may be referenced as "SYSTEM.PNT".

"DUMP" reads data blocks from disk to main memory and transfers same via a modem link.

The current day data block is updated every 6 minutes by the data acquisition software and written to disk. This procedure is repeated for 24 hours. At the 24 hour mark the data block last written will be located on disk as DAY1.DAT, DAY2.DAT, DAY3.DAT or DAY4.DAT.

Once all four data blocks have been written to disk the above procedure is repeated, overlaying DAY1.DAT, 2, 3 and so on.

5. Common Storage, global scratch area shared by all system software routines.

```
;*******************************************************************************
;                  BEGIN DATA BASE VARIABLES ADDRESS ASSIGNMENTS HERE
;*******************************************************************************
;
;
;COMMON STORAGE AREA
;
;
;GLOBLS....
                .GLOBL  DBASE,LDFLG,IPM,N,CHANEL,IUCNT,IUCNT1,IUCNT2,JDAY
                .GLOBL  MINS,CDAY,LEEPYR,SIX,SAVES,SAVES1,SAVES2,CSTAT
                .GLOBL  CALON,TRBN,MAXCH,IDAY,M,TIME,PURGE,PTIME,DABAS
                .GLOBL  DTBAS,ENDFLE,ADC1M1,STOPE,NSAMP,SRO,SR1,SR2,SR3
                .GLOBL  SR4,SR5,SHIFT,SHIFT1,SHIFT2,ERBYT,DBLOCK,IFLAG
                .GLOBL  JFLAG,MODEMF,DMPS
DBASE:          .WORD   0   ;FWA OF DATA BASE
LDFLG=BASE              ;LOAD FLAG,...EVENT
IPM=LDFLG + 2           ;INCREMENT SINCE MIDNIGHT
N=IPM+2         ;RELATIVE POINTER TO CURRENT "M" ARRAYS
CHANEL=N+2              ;CURRENT A/D CHANNEL 0-15
IUCNT=CHANEL + 2        ;UNITARY BIT COUNT, TRBN ON/OFF STATUS BITS
IUCNT1=IUCNT + 2        ;UNITARY BIT COUNT, TRB 1
IUCNT2=IUCNT1 + 2       ;UNITARY BIT COUNT, TRB 2
IUCNT3=IUCNT2 + 2       ;UNITARY BIT COUNT, TRB 3
JDAY=IUCNT3 + 2         ;JULIAN DAY COUNT
MINS=JDAY + 2           ;ONE MINUTE COUNT
CDAY=MINS + 2           ;CURRENT DAY COUNTER
LEEPYR=CDAY + 2         ;LEEP YEAR COUNT
SIX=LEEPYR + 2          ;SIX MINUTE COUNT
SAVES=SIX + 2           ;MOST RECENT SAMPLE,
SAVES1=SAVES+2          ;MOST RECENT SAMPLE TRB 1 ON/OFF
SAVES2=SAVES1+2         ;MOST RECENT SAMPLE TRB 2 ON/OFF
SAVES3=SAVES2+2         ;MOST RECENT SAMPLE TRB 3 ON/OFF
CSTAT=SAVES3 + 2                ;CAL ON/OFF STATUS BITS
CALON=CSTAT + 2         ;CALABRATION ON/OFF FLAG
TRBN=CALON+2            ;TURBINE  #'S, 0=NONE, 1=THREE
MAXCH=TRBN + 2          ;MAXIMUM CHANNEL
IDAY=MAXCH + 2          ;INITIAL DAY COUNT
M=IDAY + 2              ;RELATIVE DAY POINTER TO DATA BLOCK ARRAYS
TIME=M + 2              ;EXTERNAL CLOCK TIME DECIMAL
;                       ;       DAYS, HOURS, MINUTES, SECONDS
;                       ;UPDATED EVERY 6 MINUTES
PURGE=TIME+10           ;PURGE FLAG LOCATIONS, DAY 1-4
PTIME=PURGE+6           ;POWER OUT TIME STORAGE
DABAS=PTIME+12          ;TIME RESTORE...STORAGE, ASCII STRING
DTBAS=DABAS+42.         ;DATE RESTORE...STORAGE, ASCII STRING
ENDFLE=DTBAS+44         ;       END OF SYSTEM POINTERS, EOF
;
;DATA ONLY!, STORED EVERY 6 MINUTES
;
ADC1M1=ENDFLE + 2               ;DBLOCK-M, CH-1  CURRENT DAY STORAGE
;
ADC2M1=ADC1M1 + 360             ;DBLOCK-M, CH-2  CURRENT DAY STORAGE
;
ADC3M1=ADC2M1 + 360             ;DBLOCK-M, CH-3  CURRENT DAY STORAGE
;
ADC4M1=ADC3M1 + 360             ;DBLOCK-M, CH-4  CURRENT DAY STORAGE
```

```
ADC5M1=ADC4M1 + 360        ;DBLOCK-M, CH-5  CURRENT DAY STORAGE
;
ADC6M1=ADC5M1 + 360        ;DBLOCK-M, CH-6  CURRENT DAY STORAGE
;
ADC7M1=ADC6M1 + 360        ;DBLOCK-M, CH-7  CURRENT DAY STORAGE
;
ADC8M1=ADC7M1 + 360        ;DBLOCK-M, CH-8  CURRENT DAY STORAGE
;
A9 = ADC8M1 + 360          ;DBLOCK-M, CH-9  CURRENT DAY STORAGE
A10 = A9 + 360             ;DBLOCK-M, CH-10 CURRENT DAY STORAGE
A11 = A10 + 360            ;DBLOCK-M, CH-11 CURRENT DAY STORAGE
A12 = A11 + 360            ;DBLOCK-M, CH-12 CURRENT DAY STORAGE
A13 = A12 + 360            ;DBLOCK-M, CH-13 CURRENT DAY STORAGE
A14 = A13 + 360            ;DBLOCK-M, CH-14 CURRENT DAY STORAGE
A15 = A14 + 360            ;DBLOCK-M, CH-15 CURRENT DAY STORAGE
A16 = A15 + 360            ;DBLOCK-M, CH-16 CURRENT DAY STORAGE
;
STOPE=A16+362        ;EOF....6 MINUTE DATA STORAGE AREA.
;
;THE FOLLOWING VARIABLES MUST BE INITIALIZED UPON POWER UP
;
;FROM POWER OUT.
NSAMP=STOPE+2        ;# OF ACCUMULATED SAMPLES
SRO=NSAMP + 2        ;REGISTER
SR1=SR0 + 2
SR2=SR1 + 2
SR3=SR2 + 2          ;STORAGE
SR4=SR3 + 2
SR5=SR4 + 2          ;AREA
SHIFT=SR5 + 2        ;SHIFT COUNT,...UNITM
SHIFT1=SHIFT+2       ;TRB #1 SHIFT COUNT
SHIFT2=SHIFT1+2      ;SHIFT COUNT FOR TRB 2
SHIFT3=SHIFT2+2      ;TRB #3 SHIFT COUNT
ERBYT=SHIFT3 + 2             DISK ERROR FLAG
DBLOCK=ERBYT+2       ;FWA OF CURRENT COMMAND LINE FOR BOPY'
IFLAG=DBLOCK+2       ;INITIAL DAY FLAG
JFLAG=IFLAG+2        ;JULIAN DAY FLAG
MODEMF=JFLAG + 2             ;MODEM "QUE" FLAG
;
;BEGIN DUMP/DPURGE DATA STORAGE AREA HERE....
;
DMPS=MODEMF+2        ;STORAGE AREA FOR DUMP/DPURGE
;
;BEGIN DRIVER LOAD AREA HERE....
;
OUTSP=DMPS+10004     ;DRIVERS
INSP=OUTSP+36        ;START
SPACE=INSP+39.       ;LOADING
DSPACE=OUTSP         ;HERE
```

PROGRAM: BB.SYS BB1.SYS - Device Handlers

PURPOSE: Handle I/O from disk to memory and visa versa.

BB.SYS: Reads and prepares data blocks for transmission to disk as files.

BB1.SYS: Reads files from disk and transmits same into prescribed memory locations or remotely across telephone lines.

CALLING SEQUENCE: Envoked via RT11 command string interpreter (CSI).

BB/BB1

BB and BB1 where written and structured according to standard RT11 programming conventions for device handlers. Each contain the following sections:

- Preamble
- Header
- I/O Initiation
- Interrupt Service
- I/O Completion
- Handler Termination

These sections are discussed in the RT11 Software Support Manual - section 7.2.

LOADING AND EXECUTING

Read/write request are submitted to BB/BB1.SYS via the ADV11A Data Acquisition software or the ADV11A Diagnostic Software. Both utilize the file copy subroutine BOPY to define command string interpreter (CSI) file specifications and load BB/BB1 and disk drivers.

BB.SYS

BB.SYS is a read only driver whose basic function is to pass data from a data block or system pointer block storage area in memory to an I/O buffer area shared by resident disk drivers. The data in the I/O buffer area is used by the disk drivers to create files on disk.

When entered, BB.SYS first determines whether or not a request has been made to read a file from disk. If a "READ FILE" request is issued BB.SYS transfers program control to BB1.SYS. If not, BB.SYS determines whether or not a request has been made to write a data block or system pointer block to disk.

BB.SYS obtains the appropriate memory address of the data block or system pointer block then begins transferring the data contained in the block to the I/O buffer shared by a resident disk driver.

The disk driver transfers the data in the I/O buffer area to disk in 512 decimal word blocks. When the last block of the data block or system pointer block is written to disk a filename and extension is written in the RT11 header block (block 6) and made permanent on the disk medium. The file name/extension written is passed to the disk driver by the CSI processor.

BB.SYS terminates its operation after passing eight 512 decimal word blocks of the data block or after passing one 512 decimal word block of the system pointer block.

BB1.SYS

BB1.SYS is a write only driver whose basic function is to read and transmit files from the disk drivers I/O buffer area to memory locations reserved for data block and system pointer storage. In addition, BB1.SYS performs the function of transmitting files data across the phone line during diagnostic dump operations.

BB1.SYS is entered via BB.SYS. Its first operation makes a decision whether or not to transfer the I/O buffer data to the data block and system pointer storage area or across the phone line.

The total number of blocks (8 or 1) for the data block or system pointer file is stored in their respective memory storage area when the user request is not a diagnostic dump operation.

When a diagnostic dump request is issued BB1.SYS reads the files data from the I/O buffer, unpacks and converts each word to ASCII then transmits same down the phone line. Each 512 decimal word block transmitted is followed by a carriage return line feed until the last block is transmitted. At the completion of the diagnostic dump operation the ASCII characters ".EXIT" are transmitted and BB1.SYS aborts.

SPECIAL REQUIREMENTS

Modem device assignments must start at memory address 177560_8.
MACRO library.

EXTERNAL REQUIREMENTS

.DRBEG, .DRAST, .FORK, .DRFIN, .DREAD

LIMITATIONS

Restricted for use by ADV11A Data Acquistion and Diagnostic Software only.

PROGRAM: RD RP - RCP Device Handlers

PURPOSE: Remote communications I/O processing.

RD.SYS: Read data from Line/Write to Disk.

RP.SYS: Read data from Disk/Write to Line.

CALLING SEQUENCE: Envoked via RT11 command string interpreter (CSI).

RD/RP

RD and RP were written and structured according to standard RT11 programming conventions for device handlers. Each contain the following sections:

- Preamble
- Header
- I/O Initiation
- Interrupt Service
- I/O Completion
- Handler Termination

These sections are discussed in the RT11 Software Support Manual section 7.2.

RD.SYS

RD.Sys is a read only driver whose basic function is to read data from an incoming modem line and write same to disk.

Upon entry, RD transmits a DC1 (21_8) character down the line to initiate telephone conversation. The driver at the other end (RP) is expected to respond by sending the total block count of the file to be read followed by the files contents.

RD reads and packs bytes from the line, storing same into an I/O buffer area shared by a resident disk handler. This buffer area is 512 decimal words in length. Each time the buffer area is exhausted RD decrements the block count and the buffer area gets written to disk. This procedure is repeated until the block count equals zero or until the modem carrier is dropped.

Block transfers less than 512 are written to disk with trailing zeros.

After the last block is written to disk RD performs handler termination procedures then exits to the caller.

RP.SYS

RP.Sys is a write only driver whose basic function is to transmit data from a file located on a resident disk to a modem line.

Upon entry, RP sits in a loop awaiting a DC1 (21_8) character from the incoming line. If the DC1 character is received RP transmits a block count followed by a file of data in 512 decimal word blocks. If the DC1 character is not received and the carrier drops RP exits to the caller.

The block count of the file is preloaded by the RCOPY/RPIP routines and is resident within these programs at memory location 10028.

The data from the file to be transferred is stored in an I/O buffer area shared and maintained by a resident disk handler. The resident disk handler fills the buffer area with data from the file and RP extracts it and sends byte for byte down the line. After the last block of the file is transmitted RP performs handler termination procedures then exits to the caller.

Special Conditions - RP/RD

Following each byte transmission RP transmits a status byte. The status byte when set to a 1 indicates to RD.Sys that the byte preceding the status should have bit 8 set. RD sets bit 8 dependent on the status byte then packs same into its I/O buffer. The status byte is merely tested then ignored.

Status byte transmissions have been incorporated to accommodate the 8-bit even parity configuration of the modem.

LOADING AND EXECUTING

See RCOPY/RPIP Documentation, this manual.

LIMITATIONS

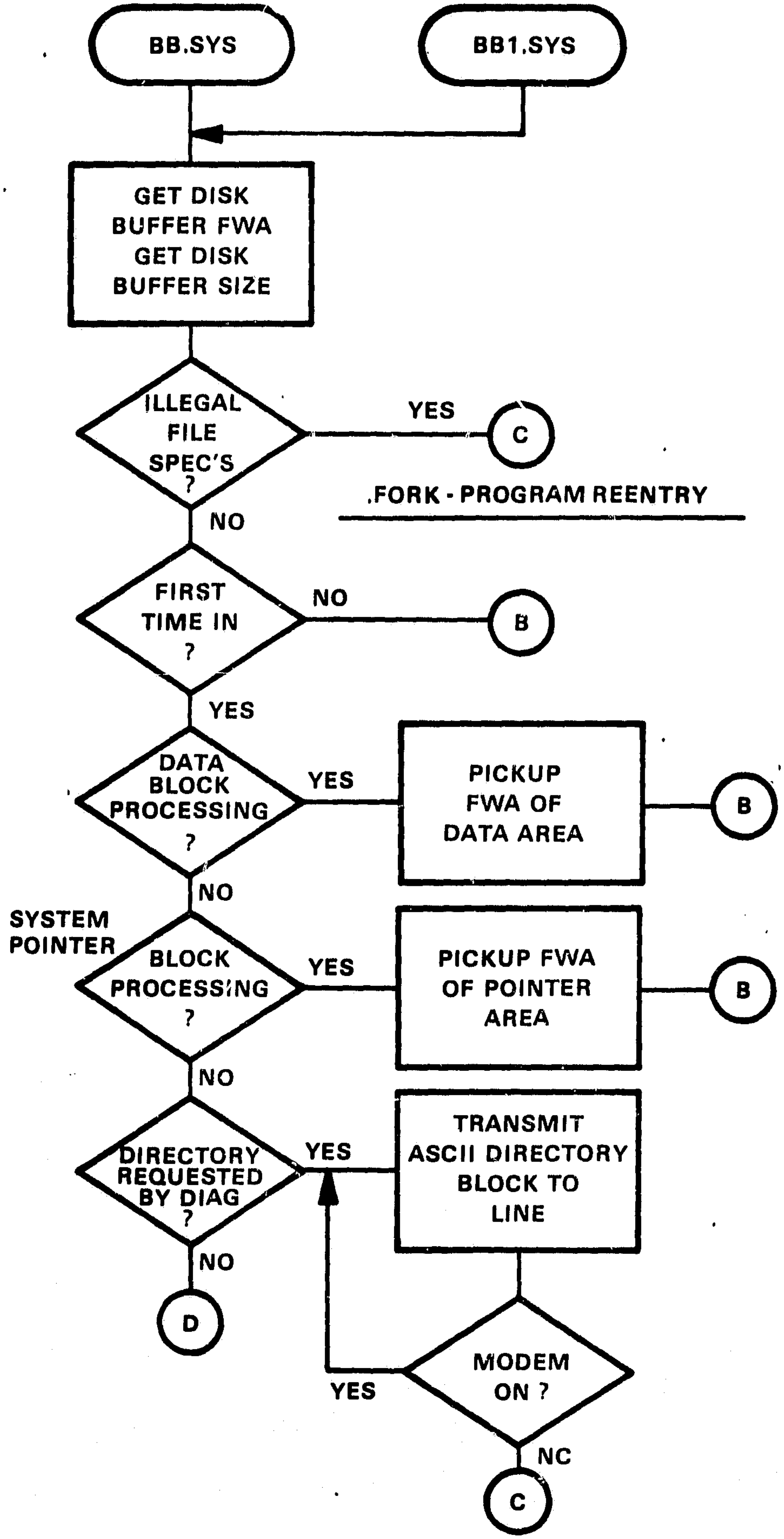
File transfers are restricted to 127 decimal blocks maximum, per transfer.

SPECIAL REQUIREMENTS

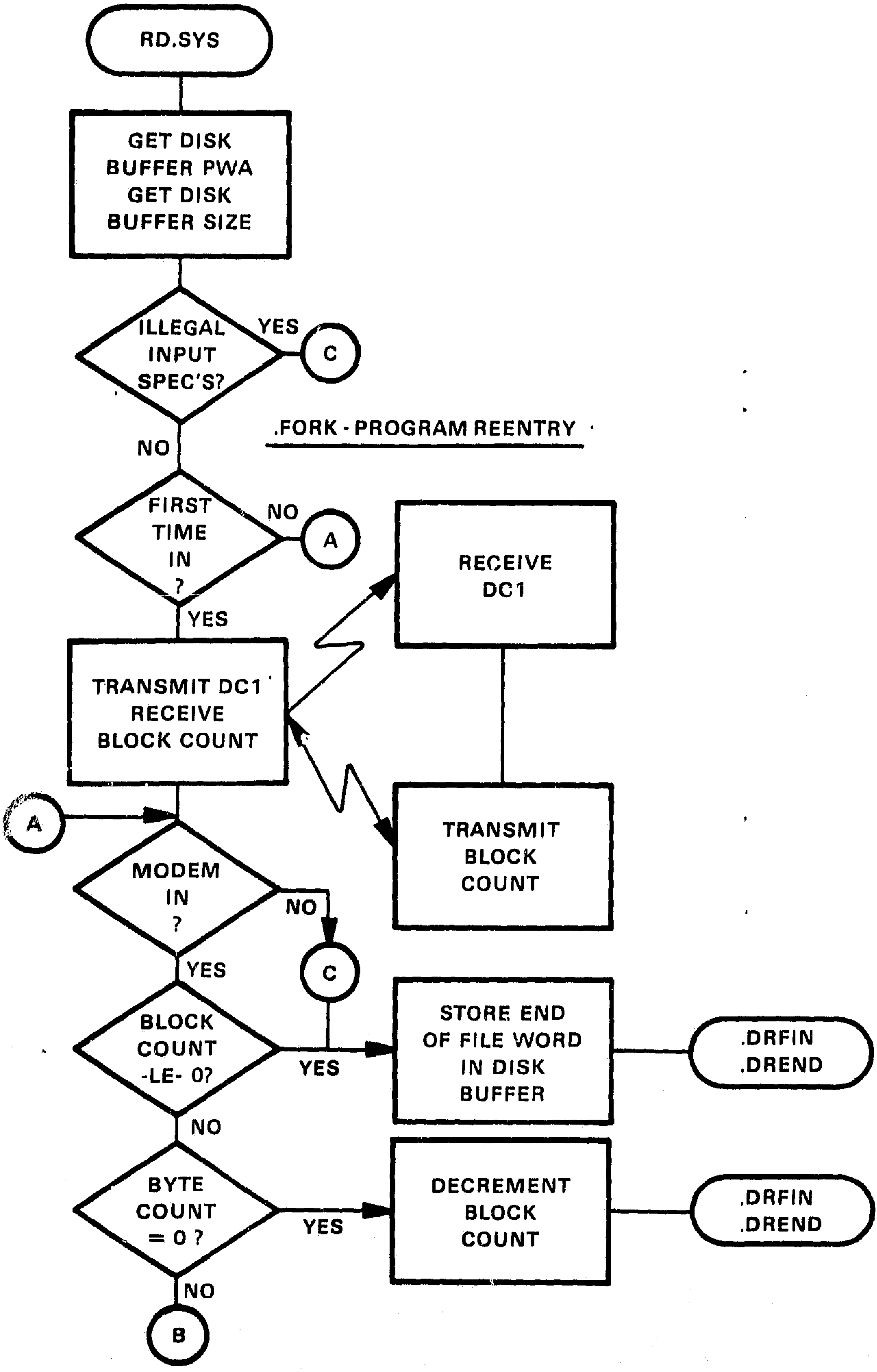
The CSR address for the disk driver must be set at 177170.
Modem device assignments must start at 177560.
MACRO library must be resident on the system storage medium before linking.

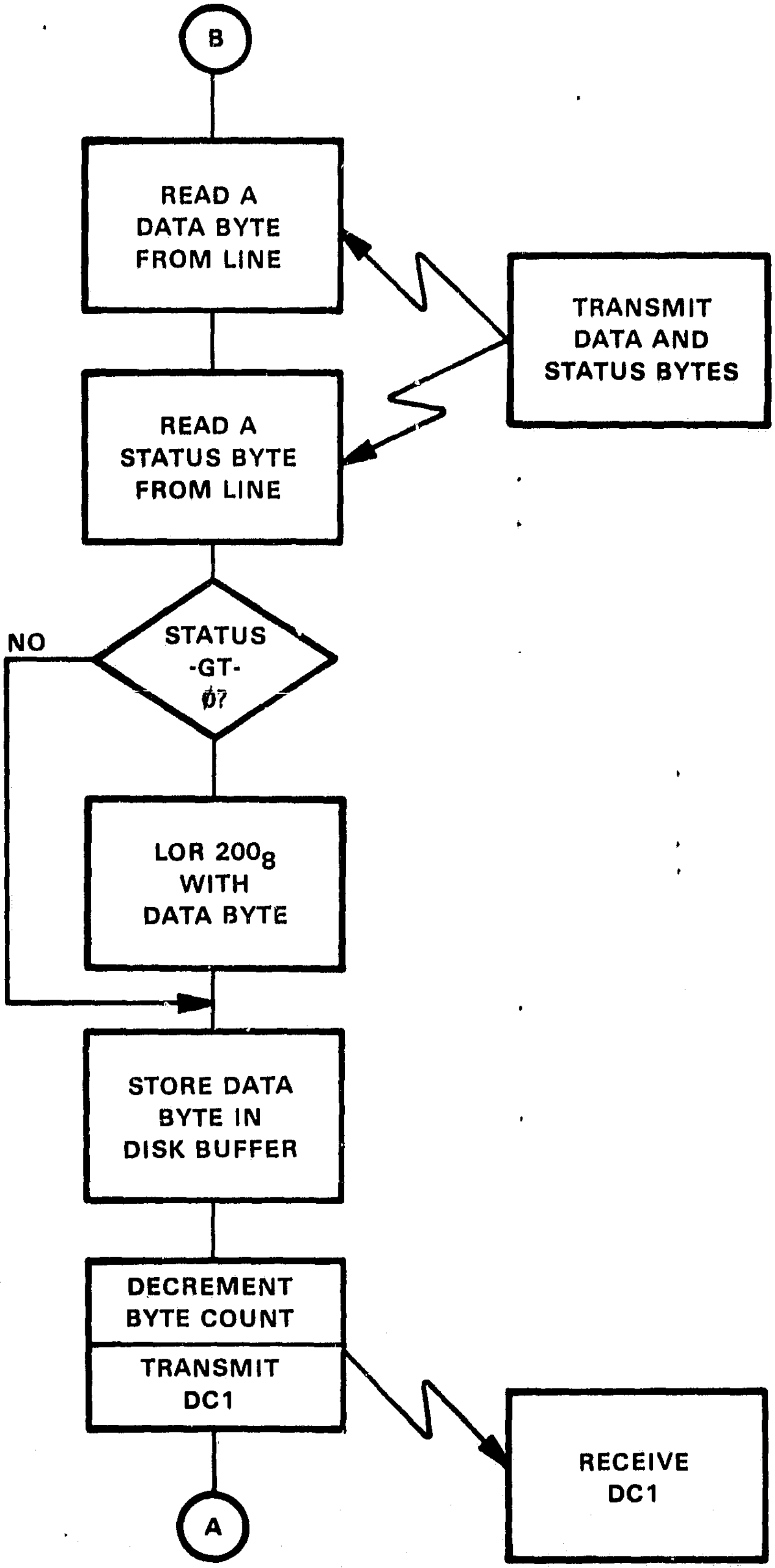
EXTERNAL REQUIREMENTS

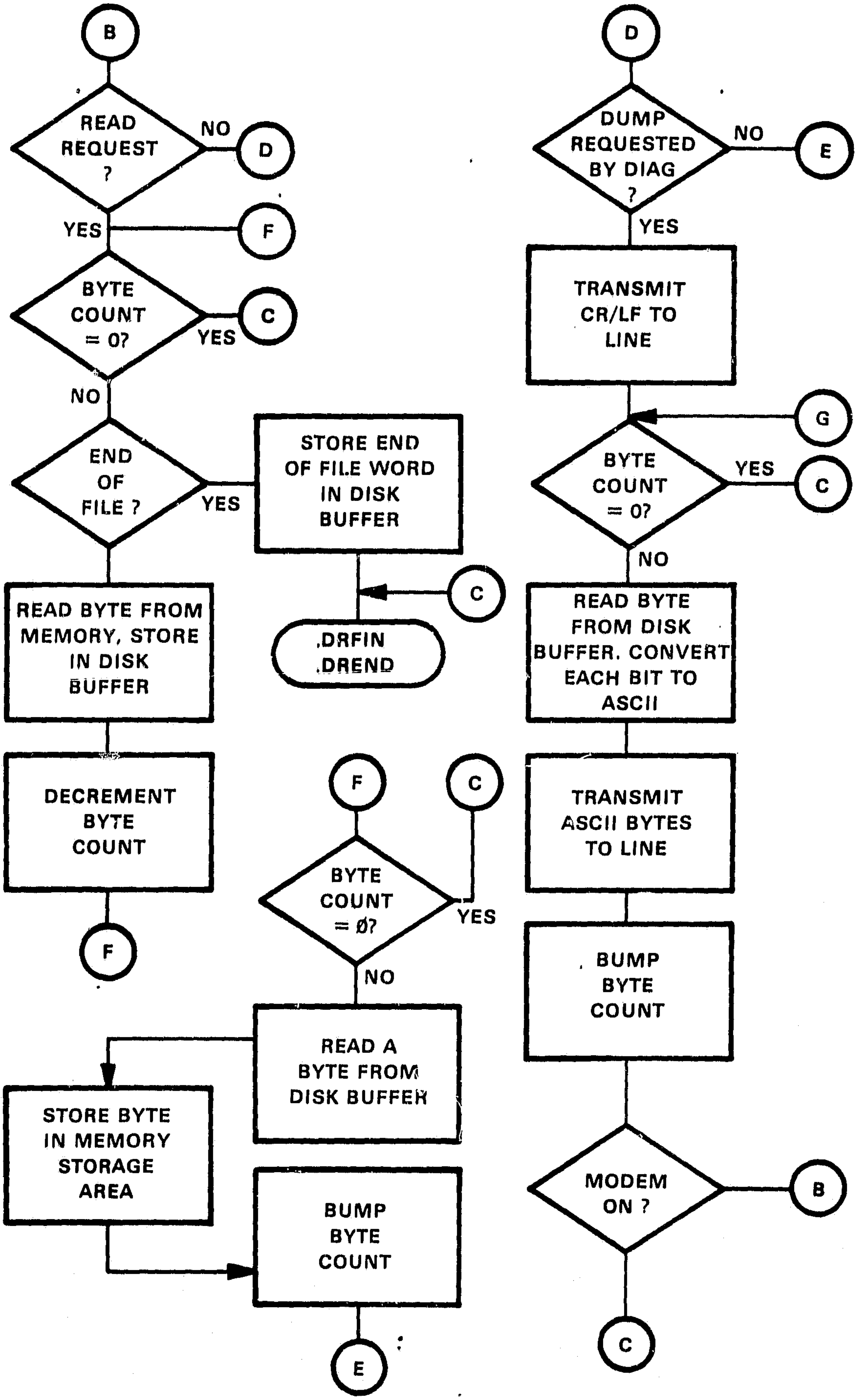
.DRFIN, .DREND, .DRBEG, .DRAST, .FORK

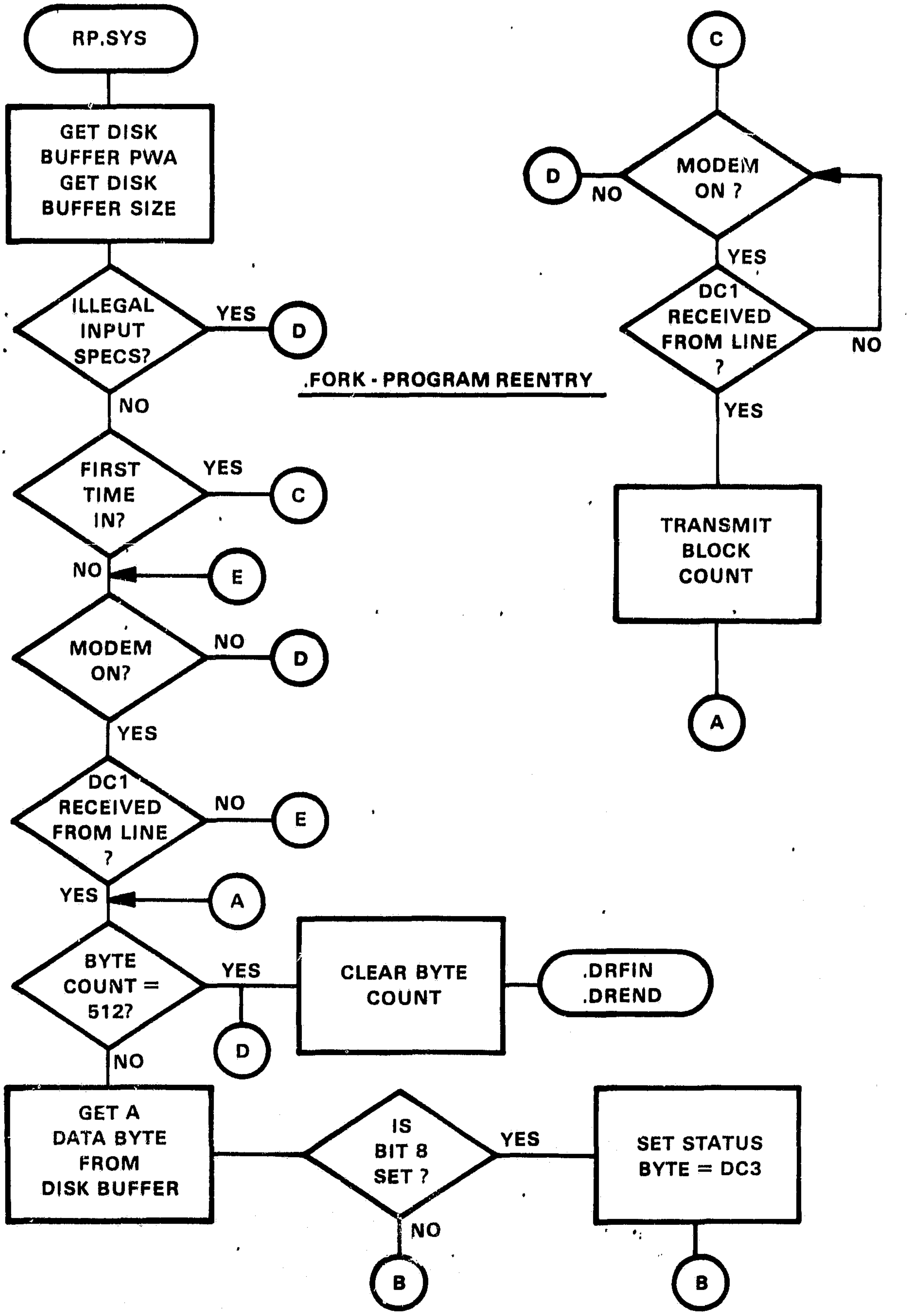

BB.SYS
BB1.SYS
GET DISK BUFFER FWA GET DISK BUFFER SIZE
ILLEGAL FILE SPEC'S ?
YES
C
NO
FIRST TIME IN ?
NO
B
YES
DATA BLOCK PROCESSING ?
YES
PICKUP FWA OF DATA AREA
B
NO
SYSTEM POINTER BLOCK PROCESSING ?
YES
PICKUP FWA OF POINTER AREA
B
NO
DIRECTORY REQUESTED BY DIAG ?
YES
TRANSMIT ASCII DIRECTORY BLOCK TO LINE
NO
D
MODEM ON ?
YES
NC
C

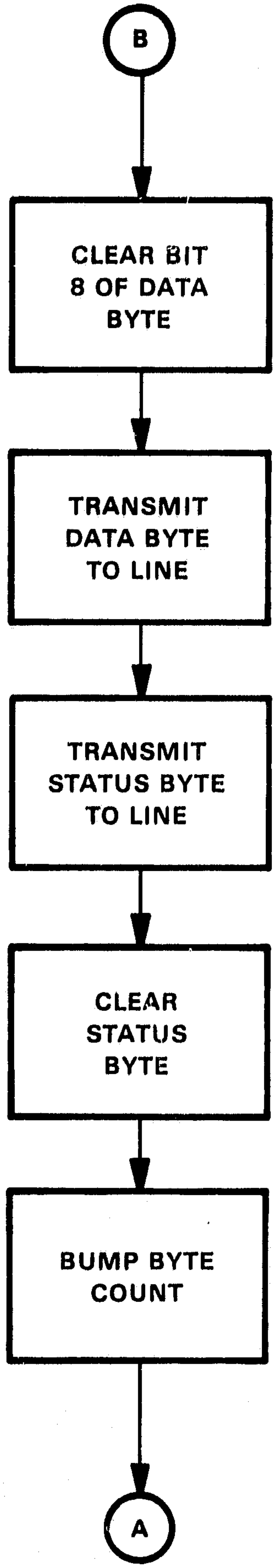
.FORK - PROGRAM REENTRY


RD.SYS
GET DISK BUFFER PWA GET DISK BUFFER SIZE
ILLEGAL INPUT SPEC'S?
YES
C
NO
.FORK - PROGRAM REENTRY
FIRST TIME IN ?
NO
A
YES
RECEIVE DC1
TRANSMIT DC1 RECEIVE BLOCK COUNT
TRANSMIT BLOCK COUNT
A
MODEM IN ?
NO
C
YES
BLOCK COUNT -LE- 0?
YES
STORE END OF FILE WORD IN DISK BUFFER
.DRFIN .DREND
NO
BYTE COUNT = 0 ?
YES
DECREMENT BLOCK COUNT
.DRFIN .DREND
NO
B


B
READ A DATA BYTE FROM LINE
TRANSMIT DATA AND STATUS BYTES
READ A STATUS BYTE FROM LINE
NO
STATUS -GT- Ø7
LOR 200_8 WITH DATA BYTE
STORE DATA BYTE IN DISK BUFFER
DECREMENT BYTE COUNT
TRANSMIT DC1
RECEIVE DC1
A


B
READ REQUEST ?
NO
D
YES
F
BYTE COUNT = 0?
YES
C
NO
END OF FILE ?
YES
STORE END OF FILE WORD IN DISK BUFFER
C
.DRFIN .DREND
READ BYTE FROM MEMORY, STORE IN DISK BUFFER
DECREMENT BYTE COUNT
F
F
C
BYTE COUNT = Ø?
YES
NO
READ A BYTE FROM DISK BUFFER
STORE BYTE IN MEMORY STORAGE AREA
BUMP BYTE COUNT
E
D
DUMP REQUESTED BY DIAG ?
NO
E
YES
TRANSMIT CR/LF TO LINE
G
BYTE COUNT = 0?
YES
C
NO
READ BYTE FROM DISK BUFFER. CONVERT EACH BIT TO ASCII
TRANSMIT ASCII BYTES TO LINE
BUMP BYTE COUNT
MODEM ON ?
B
C


RP.SYS
GET DISK BUFFER PWA GET DISK BUFFER SIZE
ILLEGAL INPUT SPECS?
YES
D
NO
FIRST TIME IN?
YES
C
NO
E
MODEM ON?
NO
D
YES
DC1 RECEIVED FROM LINE ?
NO
E
YES
A
BYTE COUNT = 512?
YES
D
CLEAR BYTE COUNT
.DRFIN .DREND
NO
GET A DATA BYTE FROM DISK BUFFER
IS BIT 8 SET ?
YES
SET STATUS BYTE = DC3
NO
B
B
.FORK - PROGRAM REENTRY
C
D
NO
MODEM ON ?
YES
DC1 RECEIVED FROM LINE ?
NO
YES
TRANSMIT BLOCK COUNT
A


B
CLEAR BIT 8 OF DATA BYTE
TRANSMIT DATA BYTE TO LINE
TRANSMIT STATUS BYTE TO LINE
CLEAR STATUS BYTE
BUMP BYTE COUNT
A

ADV11A STORAGE AREA

CHANL	(Ch-1)		Specified channel number contained in most significant byte of each word.
	(Ch-2)		
	(Ch-3)		
	(Ch-4)		
	(Ch-5)		
	(Ch-6)		
	(Ch-7)		
	(Ch-8)		
	(Ch-9)		
	(Ch-10)		
	(Ch-11)		
	(Ch-12)		
	(Ch-13)		
	(Ch-14)		
	(Ch-15)		
	(Ch-16)		
SAMP	AVGS	Ch-1	Address of appropriate subprocess for each channel for each sample taken.
	AVGS	Ch-2	
	AVGS	Ch-3	
	AVGS	Ch-4	
	AVGS	Ch-5	
	AVGS	Ch-6	
	UNITS	Ch-7	
	RTSB	Ch-8	
	RTSB	Ch-9	
	RTSB	Ch-10	
	RTSB	Ch-11	
	RTSB	Ch-12	
	RTSB	Ch-13	
	RTSB	Ch-14	
	RTSB	Ch-15	
	RTSB	Ch-16	
T1MIN	RTSB	Ch-1	Address of appropriate subprocess for each channel. Entered once per minute.
	RTSB	Ch-2	
	RTSB	Ch-3	
	RTSB	Ch-4	
	RTSB	Ch-5	
	RTSB	Ch-6	
	RTSB	Ch-7	
	UNITM	Ch-8	
	RTSB	Ch-9	
	RTSB	Ch-10	
	RTSB	Ch-11	
	RTSB	Ch-12	
	RTSB	Ch-13	
	RTSB	Ch-14	
	RTSB	Ch-15	
	RTSB	Ch-16	

T1SIX	AVG6	Ch-1	
	AVG6	Ch-2	
	AVG6	Ch-3	
	AVG6	Ch-4	
	AVG6	Ch-5	
	AVG6	Ch-6	
	AVG6	Ch-7	Address of appropriate subprocess for each channel. Entered once every six minutes.
	AVG6	Ch-8	
	AVG6	Ch-9	
	UNIT6	Ch-10	
	AVG6	Ch-11	
	AVG6	Ch-12	
	AVG6	Ch-13	
	AVG6	Ch-14	
	AVG6	Ch-15	
	UNIT6	Ch-16	
VBASE	16	Maximum channel	
	6	6 minute count constant	
	600	minute count constant	
	366	leap year constant	

SECTION V SUBPROCESS ENTRIES

```
PROGRAM:    RTSB

PURPOSE:    No Operation
CALLING
SEQUENCE:   MOV  #TABLE,R1    ;  #TABLE = FWA of subprocess table
            JSR  PC,@(R1)
```

Real Time Entry:
The address of process RTSB, a no operation, is contained in the inactive locations of the subprocess tables, SAMP, TIMIN, and TISIX. The execution of the process RTSB initiates a Return From Subroutine (RTS).

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals: None

```
PROGRAM:    AVGS

PURPOSE:    Accumulate and average groups of sample data
CALLING
SEQUENCE:   MOV   #SAMP,R1    ;    FWA of sample table
            JSR   PC,@(R1)
```

Upon entry, R0 = FWA of Turbine Data current channel.

Real Time Entry:
When no power outages are evident, AVGS is entered 10 times per second, via the A/D software driver ADV11A. AVGS accumulates data samples in Table TB1180, takes an average of the result, then accumulates the average quotient in a common storage area, TAble TB124. This procedure takes place 4 times per minute and 24 times per six minutes. At the six minute mark, the quotients summed for each channel are averaged and stored into the data block by process AVG6. Within each one minute period, the first three accumulations consist of 180 data samples per channel. The fourth accumulation consists of 60 data samples per channel. The common storage areas TB1180 and TB124 are set equal to 0 by EVENT and LOGIC6 routines.

The Mathematical formula used to compute the average of 180 data samples per channel is as follows:

$$\text{Common (channel)} = \frac{\sum_{i=1}^{180} \text{Data } i}{180} + \text{Common (channel)}$$

where ... Common = common storage area for quotient accumulations
Channel = Channel data locations 1 through 7 in the common storage area, stated above.
Data = 180 data samples per channel

The mathematical formula used to compute the average of 60 data samples per channel is as follows:

$$\text{Common (channel)} = \frac{\sum_{i=1}^{60} \text{Data } i + \text{Common (channel)}}{60}$$

where... Common = common storage area for quotient accumulations
Channel = channel data locations 1 through 7 in the common storage area, stated above.
Data = 60 data samples per channel

NOTE: Each TB1180 Location is reinitialized before the respective quotient is accumulated in TB124.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:

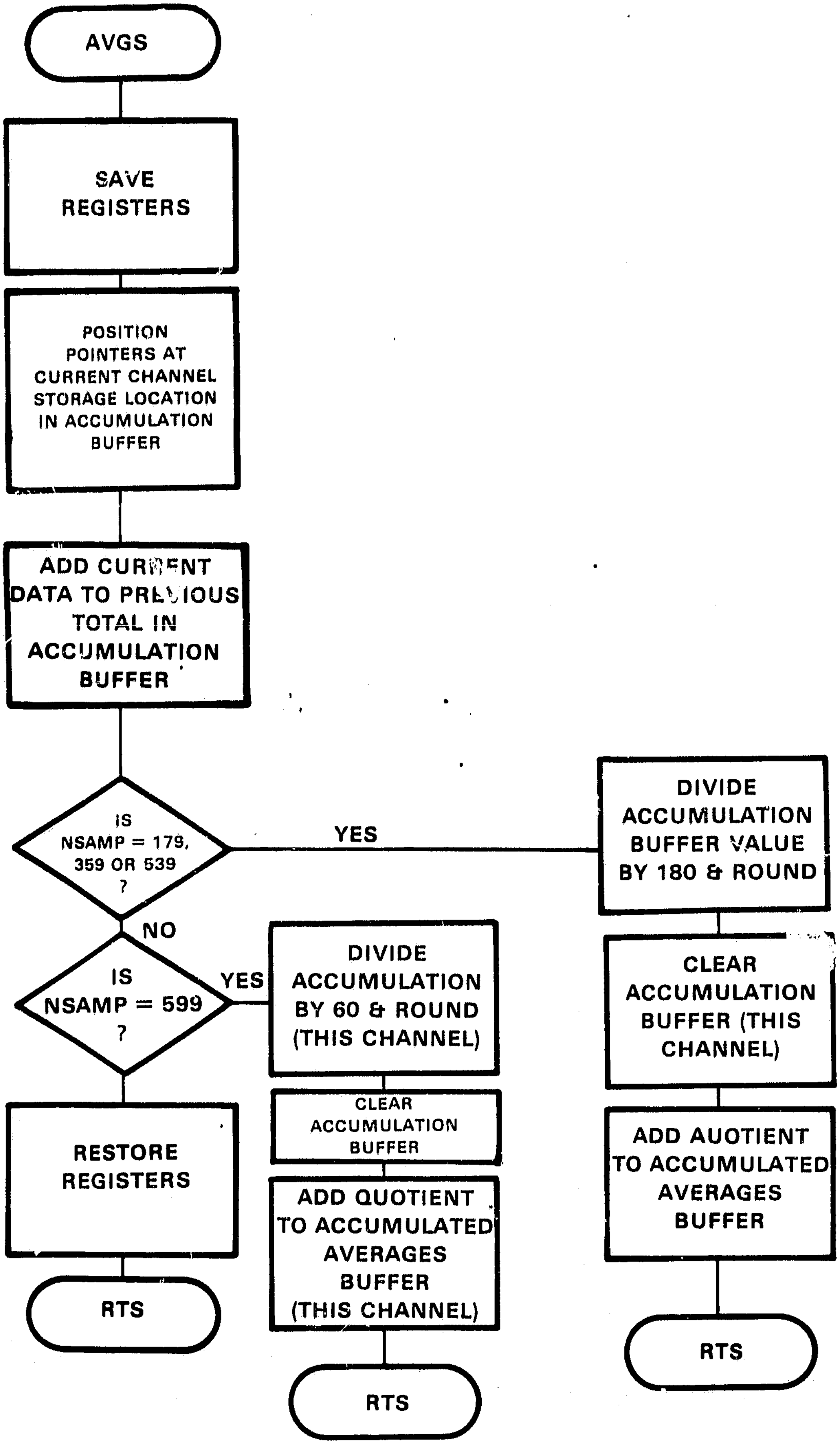
TB1180, TB124, CHANEL, NSAMP

C-2

AVGS STORAGE AREA

TB1180	(Ch-1)
	(Ch-2)
	(Ch-3)
	(Ch-4)
	(Ch-5)
	(Ch-6)
	(Ch-7)
	(Ch-8)
	(Ch-9)
	(Ch-10)
	(Ch-11)
	(Ch-12)
	(Ch-13)
	(Ch-14)
	(Ch-15)
	(Ch-16)

Stores sum of data samples


AVGS
SAVE REGISTERS
POSITION POINTERS AT CURRENT CHANNEL STORAGE LOCATION IN ACCUMULATION BUFFER
ADD CURRENT DATA TO PREVIOUS TOTAL IN ACCUMULATION BUFFER
IS NSAMP = 179, 359 OR 539 ?
YES
DIVIDE ACCUMULATION BUFFER VALUE BY 180 & ROUND
CLEAR ACCUMULATION BUFFER (THIS CHANNEL)
ADD AUOTIENT TO ACCUMULATED AVERAGES BUFFER
RTS
NO
IS NSAMP = 599 ?
YES
DIVIDE ACCUMULATION BY 60 & ROUND (THIS CHANNEL)
CLEAR ACCUMULATION BUFFER
ADD QUOTIENT TO ACCUMULATED AVERAGES BUFFER (THIS CHANNEL)
RTS
RESTORE REGISTERS
RTS

PROGRAM: UNITS

PURPOSE: Save present ON/OFF turbine status

CALLING
SEQUENCE:
```
MOV #SAMP,R1    ;    FWA of sample table
JSRPC,@(R1)
```

Real Time Entry:
R0 - FWA of turbine data.
UNITS monitors channels data 10 times per second over a period of one minute to determine (1) whether the turbine is on or off and/or (2) whether "Data Calibration" is in progress. The turbine is determined to be on or off based on a tolerance defined and supplied by the user. The data associated with the channel connected to the process UNITS is compared to this tolerance and when greater than or equal to denotes the on status of the turbine.

The globaled SAVES is set to a '1' or '0' after the turbine is determined to be on or off. When set to a '0', SAVES reflects the OFF status of the turbine. When set to a '1', SAVES reflects the ON status of the turbine. This ON/OFF status bit is unitarily packed into 6 bits of a single word every minute for 6 minutes then stored in the data block. Packing and storing channels data is a function of the processes UNITM and UNIT6.

Data calibration is considered "in progress" when the channels data is stable at 0 to 15 counts for a period of at least 30 seconds. Consequently, the globaled "CALON" flag is set to a 1 to reflect the calibration ON status and to a 0 to reflect the calibration OFF status as determined by the state of the channels data within this 30 second period. Within the subprocess UNITM and once per minute, CALON bit 1 is unitarily packed into 6 bit positions of the calibration status word CSTAT. CSTAT represents the calibration status over a period of 6 minutes.

In addition, UNITS monitors data calibration to determine whether any channel calibrates consecutively for more than 1 minute. When any calibration channel exceeds a 1 minute calibration limit the calibration error flag, CALERR, is set to 1. This flag is monitored by the sub-process CALB6. CALB6 sets an error bit in bit position #8 of CSTAT as determined by the state of CALERR.

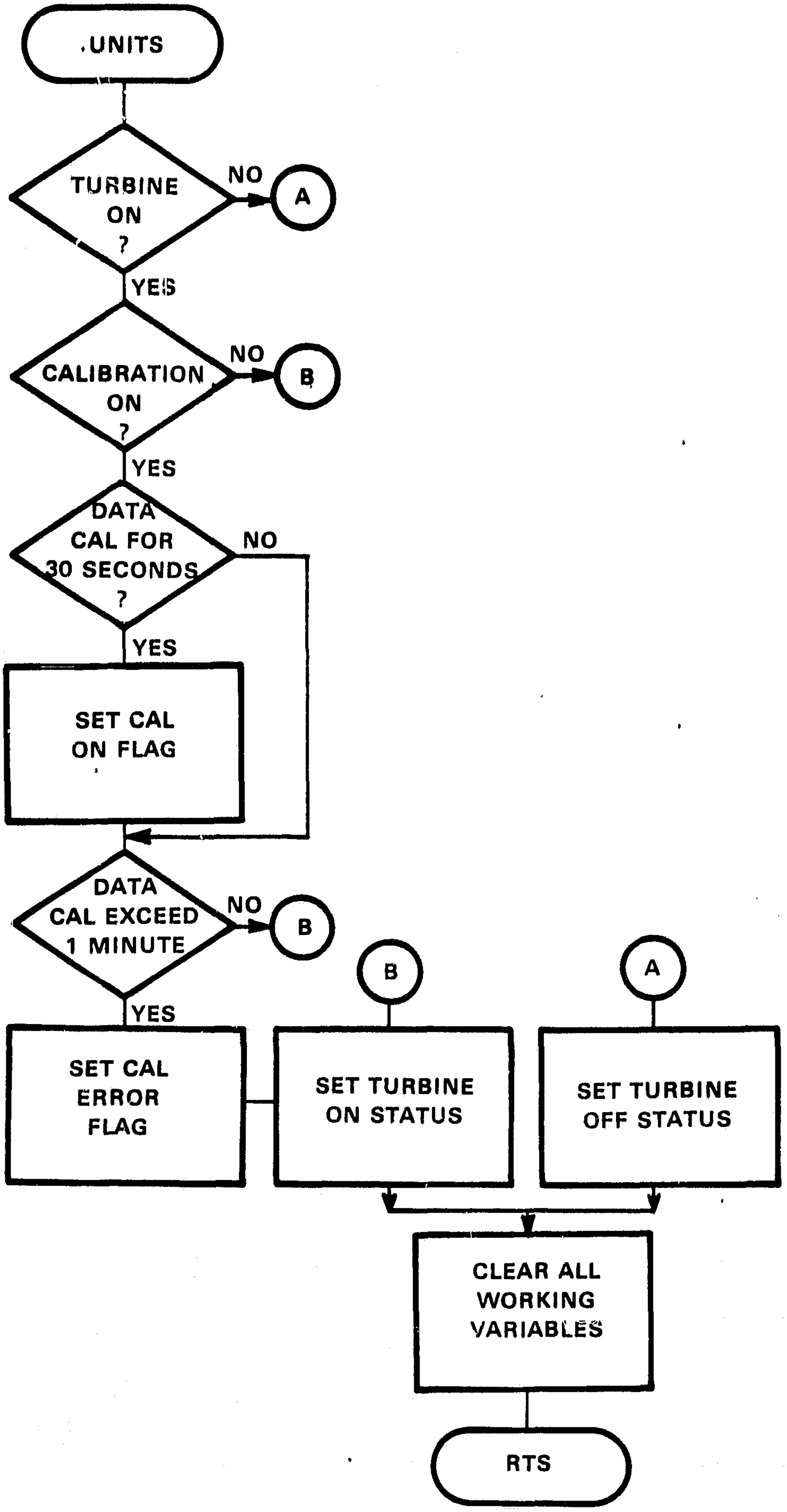
Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:

SAVES, MINS


.UNITS
TURBINE ON ?
NO
A
YES
CALIBRATION ON ?
NO
B
YES
DATA CAL FOR 30 SECONDS ?
NO
YES
SET CAL ON FLAG
DATA CAL EXCEED 1 MINUTE
NO
B
YES
SET CAL ERROR FLAG
B
SET TURBINE ON STATUS
A
SET TURBINE OFF STATUS
CLEAR ALL WORKING VARIABLES
RTS

PROGRAM: UNITM

PURPOSE: Pack turbine and calibration ON/OFF status

CALLING
SEQUENCE:

```
MOV  #T1MIN,R1    ;    FWA of minute table
JSR  PC,@(R1)
```

Real Time Entry:

Upon entry,

SAVES = Turbine ON/OFF status.

CALON = Calibration ON/OFF status.

UNITM is entered once per minute to retrieve and unitarily pack channels data into 6 bit positions of the globaled IUCNT and CSTAT. Each of the globals are contained in the data base and made accessible to all subprocesses.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:

SAVES, CALON, SHIFT, IUCNT, CSTAT, SHIFT1, SHIFT2, IUCUT1, IUCUT2, SAVES1, SAVES2, SAVES3, IUCNT3

PROGRAM: AVG6

PURPOSE: Computes final average of data for a six minute period.

CALLING
SEQUENCE:

```
MOV  #T1SIX,R1    ;    R1 = FWA of 6 minute table
JSR  PC,@(R1)
```

Real Time Entry:

Upon entry,

R5 = FWA of Day-M(n) for the current channel, located within the data block of the current day.

AVG6 divides the accumulated averages of Table TB124 by 24 and stores the quotient in the data block. These quotients are the final average for each data channel over a six minute period and are within 1% of the absolute value.

Special Conditions: None

Special Requirements: None

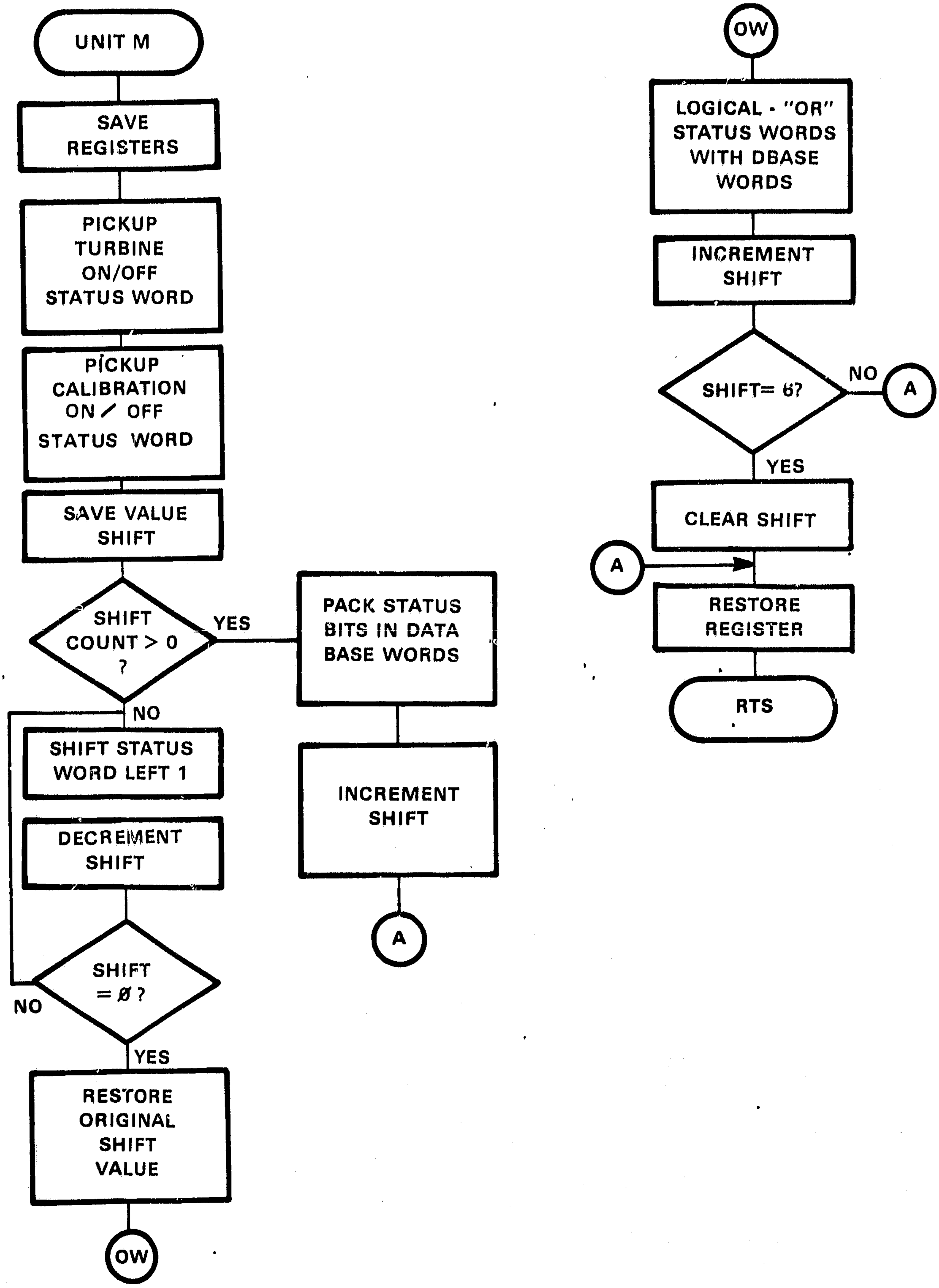
External Requirements: None

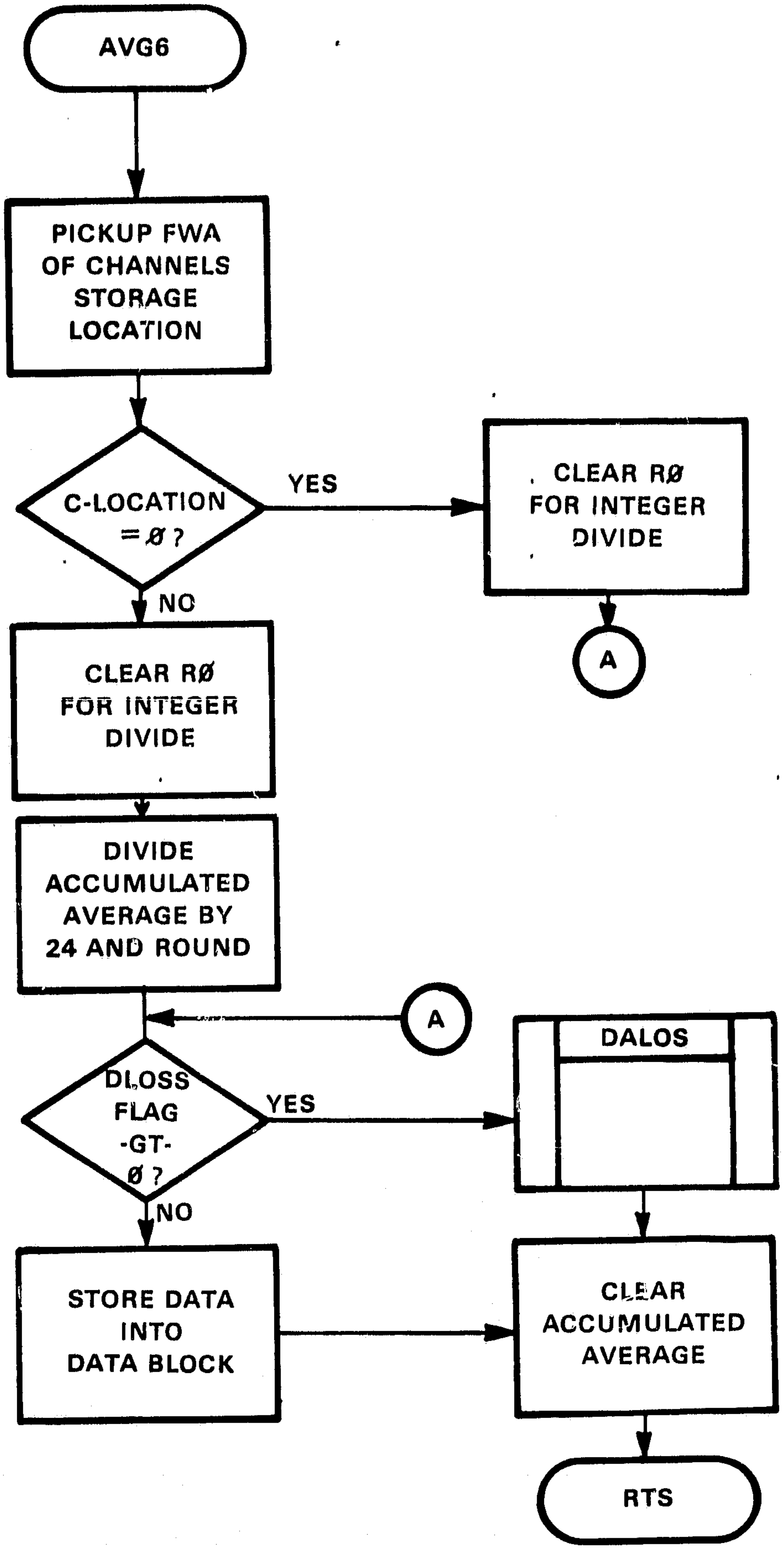
Globals:

TB124, CHANEL

AVG6 STORAGE AREA

TB 124	(Ch-1)	
	(Ch-2)	
	(Ch-3)	
	(Ch-4)	
	(Ch-5)	
	(Ch-6)	
	(Ch-7)	
	(Ch-8)	Stores sum of averaged data samples
	(Ch-9)	
	(Ch-10)	
	(Ch-11)	
	(Ch-12)	
	(Ch-13)	
	(Ch-14)	
	(Ch-15)	
	(Ch-16)	


UNIT M
SAVE REGISTERS
PICKUP TURBINE ON/OFF STATUS WORD
PICKUP CALIBRATION ON / OFF STATUS WORD
SAVE VALUE SHIFT
SHIFT COUNT > 0 ?
YES
PACK STATUS BITS IN DATA BASE WORDS
INCREMENT SHIFT
A
NO
SHIFT STATUS WORD LEFT 1
DECREMENT SHIFT
SHIFT = Ø ?
NO
YES
RESTORE ORIGINAL SHIFT VALUE
OW
OW
LOGICAL - "OR" STATUS WORDS WITH DBASE WORDS
INCREMENT SHIFT
SHIFT= 6?
NO
A
YES
CLEAR SHIFT
A
RESTORE REGISTER
RTS


AVG6
PICKUP FWA OF CHANNELS STORAGE LOCATION
C-LOCATION =Ø ?
YES
CLEAR RØ FOR INTEGER DIVIDE
A
NO
CLEAR RØ FOR INTEGER DIVIDE
DIVIDE ACCUMULATED AVERAGE BY 24 AND ROUND
A
DLOSS FLAG -GT- Ø ?
YES
DALOS
NO
STORE DATA INTO DATA BLOCK
CLEAR ACCUMULATED AVERAGE
RTS

PROGRAM: UNIT6

PURPOSE: Stores turbine ON/OFF status word in the data block .

```
CALLING
SEQUENCE:  MOV  #T1SIX,R1     ;  six minute table
           JSR  PC,@(R1)
```

Real Time Entry:
Upon entry,
R5 = FWA of Day-M(n) located within the data block of the current day.
IUCNT = Turbine ON/OFF status word, updated by UNITM.
Every six minutes UNIT6 stores the turbine ON/OFF status word into the data block at a location reserved for the channel. Upon exit, IUCNT is reinitialized.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:

IUCNT, CALERR, CSTAT, IUCNT1, IUCNT2, IUCNT3

PROGRAM: CALB6

PURPOSE: Stores calibration status word and checks for a calibration error.

```
CALLING
SEQUENCE:  MOV  #T1SIX,R1     ;    R1 = FWA of 6 minute table
           JSR  PC,@(R1)
```

Real Time Entry:
Upon entry,
R5 = FWA of DayM(n) located within the data block of the current day.
CALERR = Calibration error flag, set by UNITS
CSTAT = Calibration ON/OFF status word, updated by UNITM
Every 6 minutes CALB6 performs the following functions:
1. Verifies the CALERR flag;
2. Sets bit 8 of CSTAT to '1' when CALON is greater than 0,
3. And/or, stores CSTAT into the current data block location.
Upon exit CSTAT and CALERR are reinitialized.

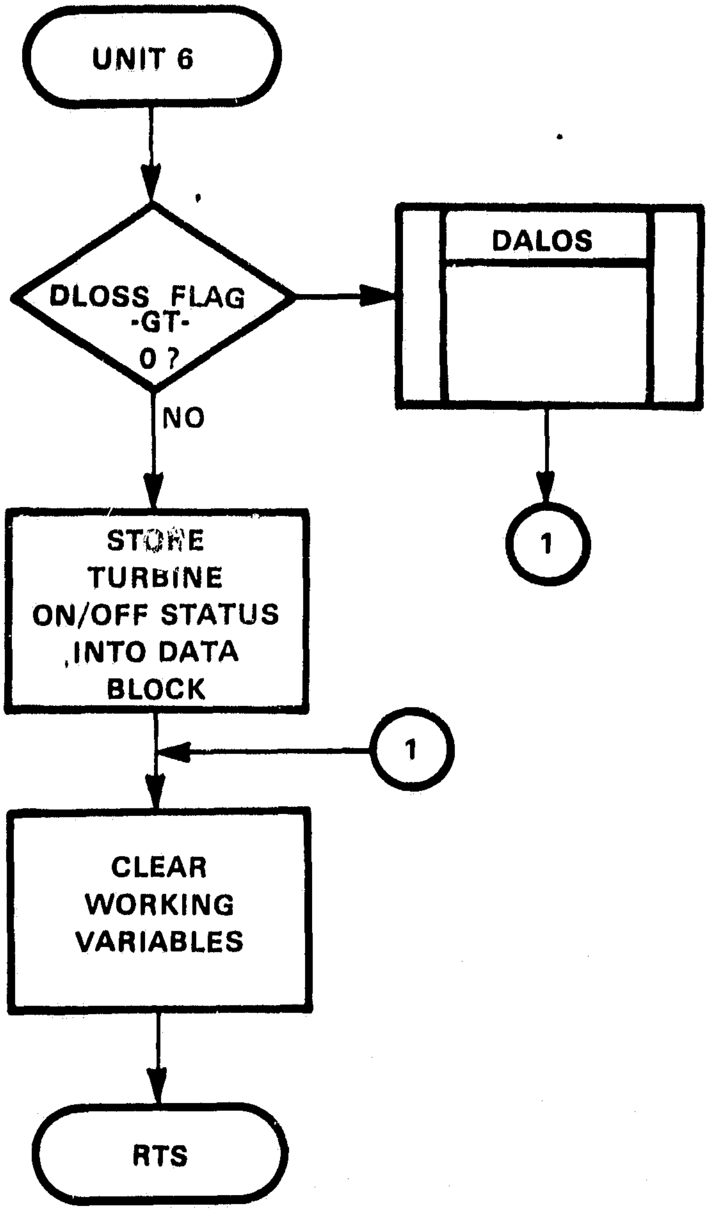
Special Conditions: None

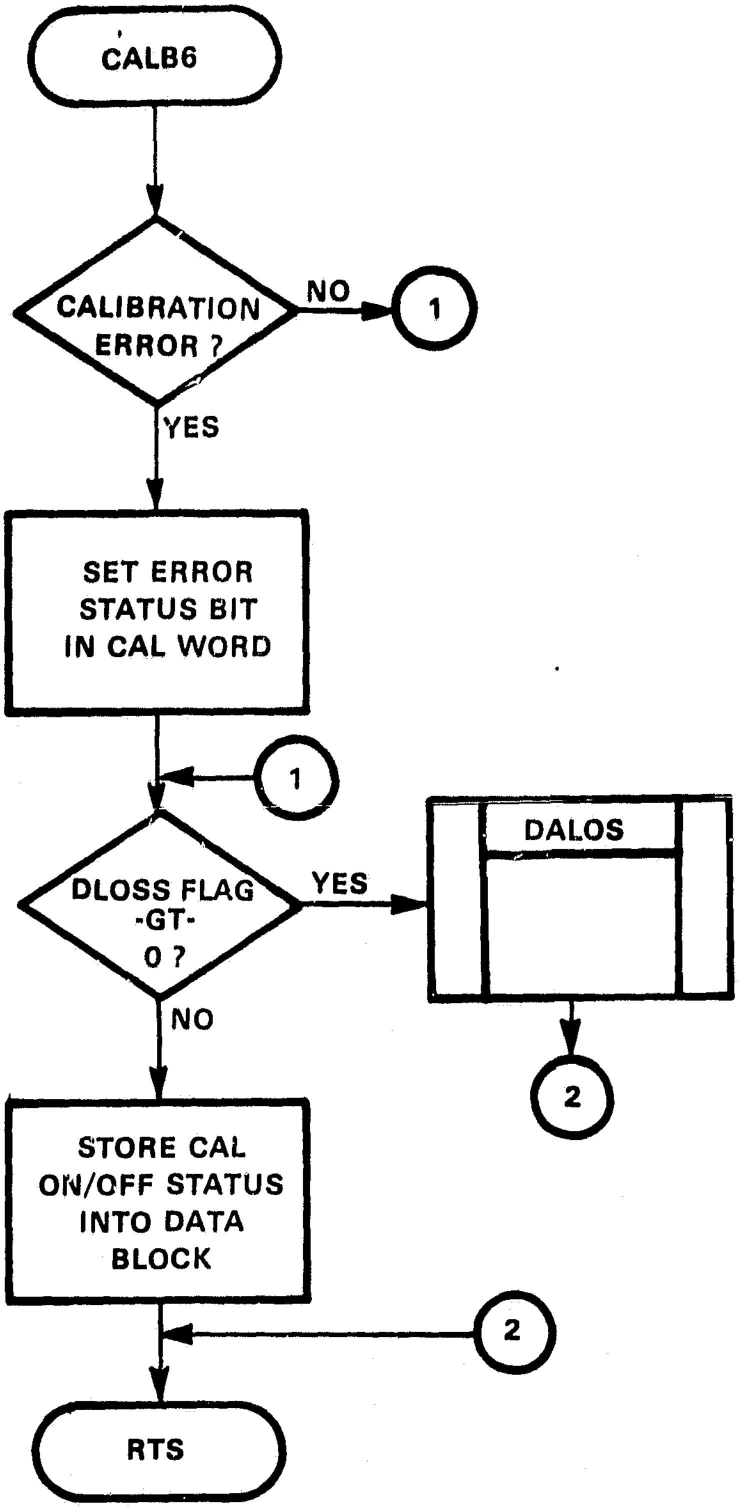
Special Requirements: None

External Requirements: None

Globals:

CALERR, CSTAT, IUCNT


UNIT 6
DLOSS FLAG -GT- 0 ?
DALOS
NO
1
STORE TURBINE ON/OFF STATUS INTO DATA BLOCK
1
CLEAR WORKING VARIABLES
RTS


CALB6
CALIBRATION ERROR ?
NO
1
YES
SET ERROR STATUS BIT IN CAL WORD
1
DLOSS FLAG -GT- 0 ?
YES
DALOS
2
NO
STORE CAL ON/OFF STATUS INTO DATA BLOCK
2
RTS

SECTION VI SYSTEM UTILITIES

PROGRAM: TRAP

PURPOSE: ADV11A Software System Trap Recovery and Maintenance.
CALLING
SEQUENCE: Executed by the RT11 monitor via some trap vector address (described below).

Vector	Trap Entry Name	Trap Description
004	TRAP4	Caused by a reference to a non-existing device or odd address, or by a bus time-out.
010	TRAP10	Caused by an attempt to execute a reserved or illegal instruction.
014	TRAP14	Breakpoint Trap, usually issued by a debugging utility program such at ODT.
020	TRAP20	Input Output TRAP, Program I/O TRAP.
030	TRAP30	EMT Emulator Trap
034	TRAP34	Program Trap Instruction, issued by a program to change the flow of execution.
114	T114	Memory Parity Trap, caused by a memory parity error.
244	T244	FPU Trap, caused by the Floating Point Unit exception or error.
264	T264	Disk I/O error.

Special Conditions:
The address of each entry per trap must be stored in the respective Trap Vector Address before program execution can take place.

Globals:

TP4, TP4A, TP10, TP10A, TP30, TP30A, TP34, TP34A, TP114, TP114A, TP244, TP244A, TP264, TP264A, TP14, TP14A

PROGRAM: RCOPY RPIP - Remote file XFR processing.

PURPOSE: Establish and oversee I/O between the remote communications drivers RD.SYS and RP.SYS.

CALLING
SEQUENCE: RUN RCOPY
or
RUN RPIP

RCOPY.SAV
RCOPY communicates with the performance monitor data acquisition software via modem.

In an effort to simplify yet afford RCOPY with adaptability it has been incorporated as a stand alone system "RUN" routine for a RT11 base system.

In its basic form RCOPY solicits information from the user via the system terminal. This information is used when preparing communication between the RCP drivers.

Command strings describing the drivers and file names of the files to be read remotely and transferred locally are input to RCOPY by the user. RCOPY interrogates the remote command string and looks for valid RCP driver definitions and string formats. Invalid string formats are not accepted. The user is requested to reenter the command upon error. However, valid command strings are passed down the modem line to the data acquisition software. When received, the data acquisition software loads and executes the appropriate drivers. Thereafter the drivers await file transmission.

After RCOPY transmitts the remote command string the user is asked to input the block count of the file. Once entered, RCOPY decodes and stores the block count for later use by the RP driver. The block count is resident at memory location 1002_8.

Next, RCOPY envokes the command string interpreter (CSI) to solicite from the user a local command string file specifications. CSI interrogates the string for legal and illegal file specifications. Illegal file specs are not accepted. The user is requested to reenter the specs upon error.

CSI performs .LOOKUP and/or .ENTER request and loads the drivers defined in the local command string, when legal file specs are input by the user.

After the local drivers are loaded RCOPY issues READ/WRITE commands to the drivers.

Upon each READ/WRITE request the drivers perform the functions defined in the local command string file specification and transmitts down the modem line the block count followed by file data. A file transfer is completed when the last block of the file is transmitted. The block count is transmitted once per file.

At the completion of each file transfer RCOPY is reinitialized and the above procedures repeated.

RPIP.SAV

RPIP performs functions identical to RCOPY. However, as a run routine RPIP communicates with the performance monitor at the RT11 executive or monitor level. That is, RPIP is designed to communicate with the RT11 system routine PIP (Peripheral Interchange Program).

RPIP.SAV was incorporated to safeguard and reassure file transfer capability in the event that the performance monitor data acquisition software gets corrupted and is unusable.

PROGRAM/OPERATOR INTERCHANGE - Example Format

PROGRAM: Enter command line for remote filespecs - [Type "HELP" if not sure]
OPERATOR: DXO:FILEOUT:EXT=RD:FILEIN.EXT

PROGRAM: Enter block count for local filespecs (Octal)
OPERATOR: 41

PROGRAM: Enter command line for local filespecs.
OPERATOR: RP:FILEOUT.EXT=DYO:FILEIN.EXT

In the example above, FILEOUT.EXT is the name of the file to be written, FILEIN.EXT is the name of the file to be read. Simularly, FILEIN.EXT will be read locally from DYO: and written to modem via RP:. RD will read FILEOUT.EXT from RP and transfer same to DXO:. The total number of blocks transferred for FILEIN or FILEOUT is 41_8. When the transfer is complete DXO: will contain a file of 41_8 blocks named FILEOUT.EXT.

Filenames FILEOUT.EXT and FILEIN.EXT may be replaced by a file name resident on the local and/or remote disk storage medium.

I/O ERRORS

PROGRAM: Illegal command line for remote FILESPECS.
OPERATOR: Reenter legal command line.

PROGRAM: Input error.
OPERATOR: The input filename can not be found on the local storage medium.
Reenter new local filespecs.

PROGRAM: Output error.
OPERATOR: The channel to the modem is not on. Dial and connect modem to the channel. Operator must start RPIP/RCOPY program from the beginning.

DIRECTORY LIST

File names, extentions and block counts may be obtained via the following directory command.

```
.DIR/OCTAL  FILENAME.EXT
```

Special Conditions: None

Special Requirements:

MACRO library must be resident on the system medium before linking.

External Requirements:

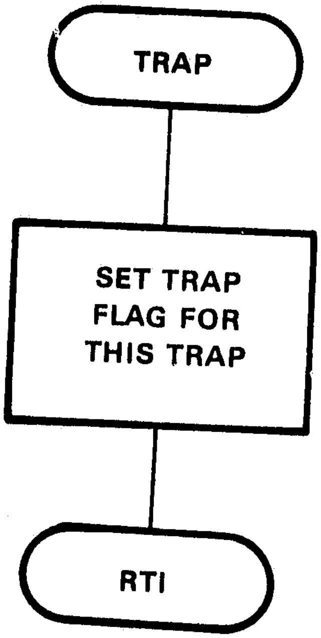
.CSIGIN, .PRINT, .EXIT, .READW, .WRITW, .CLOSE, .SRESET

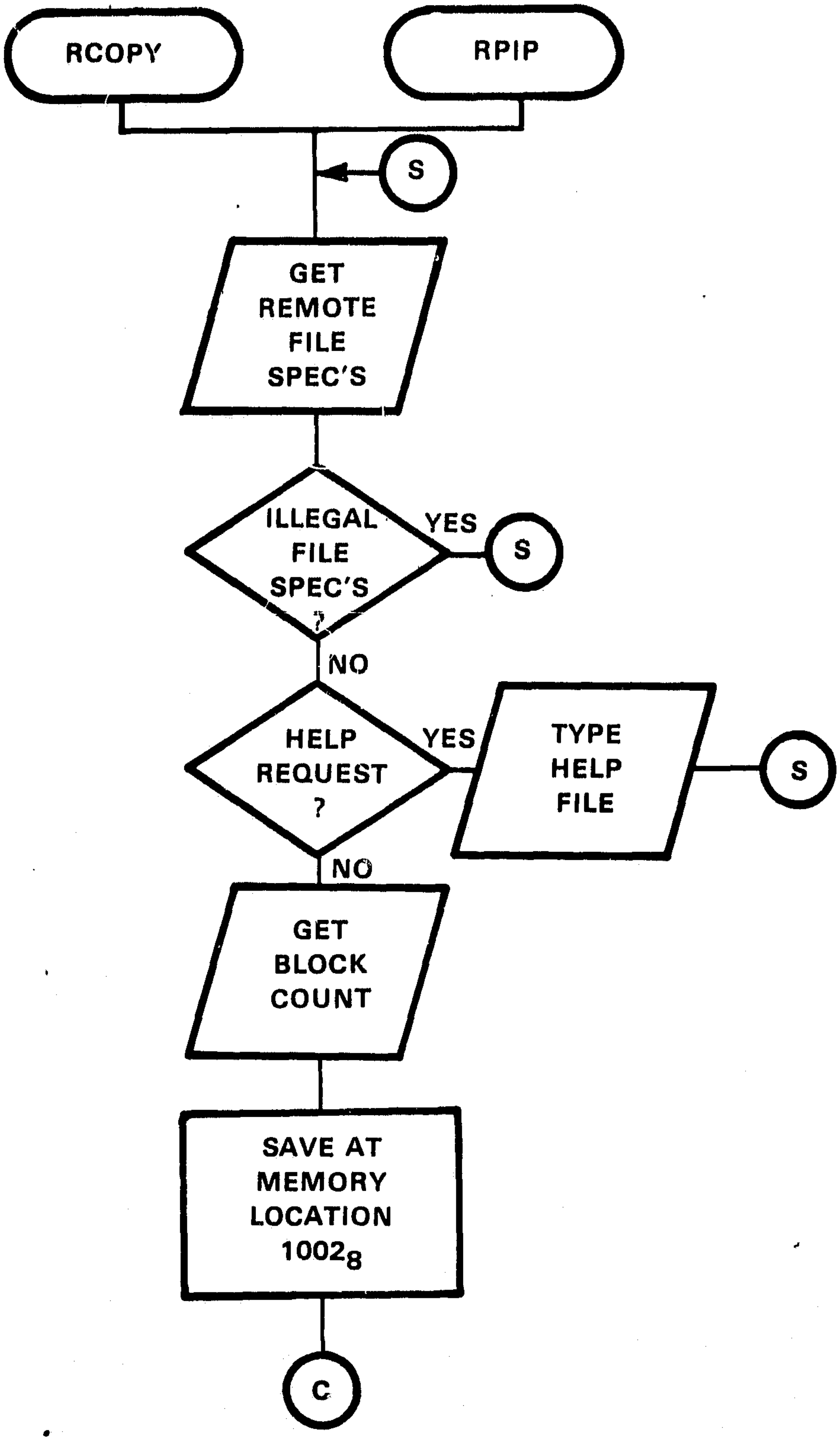
ASSEMBLY

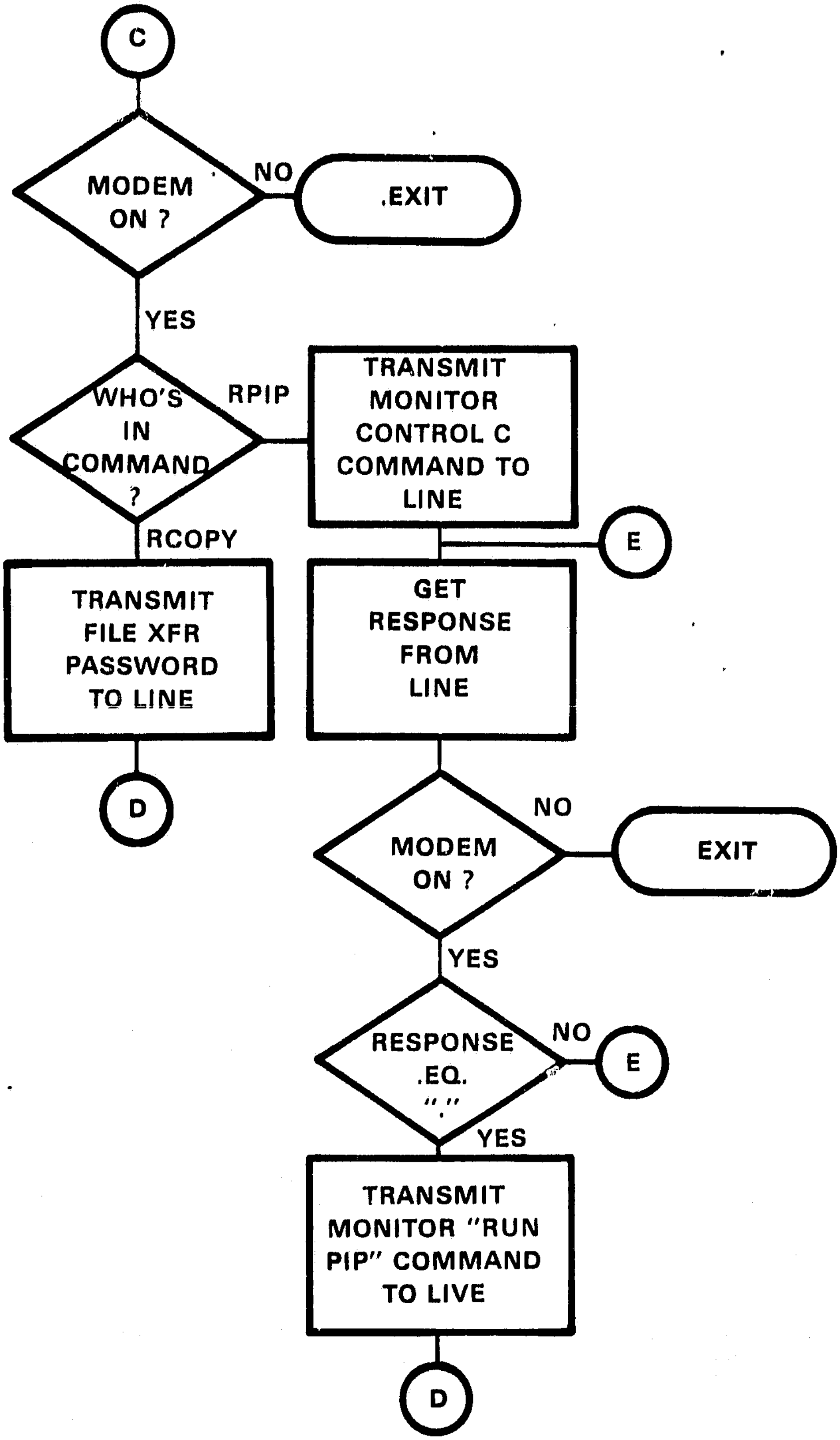
```
RCOPY = RCOPY
RPIP  = RPIP
```

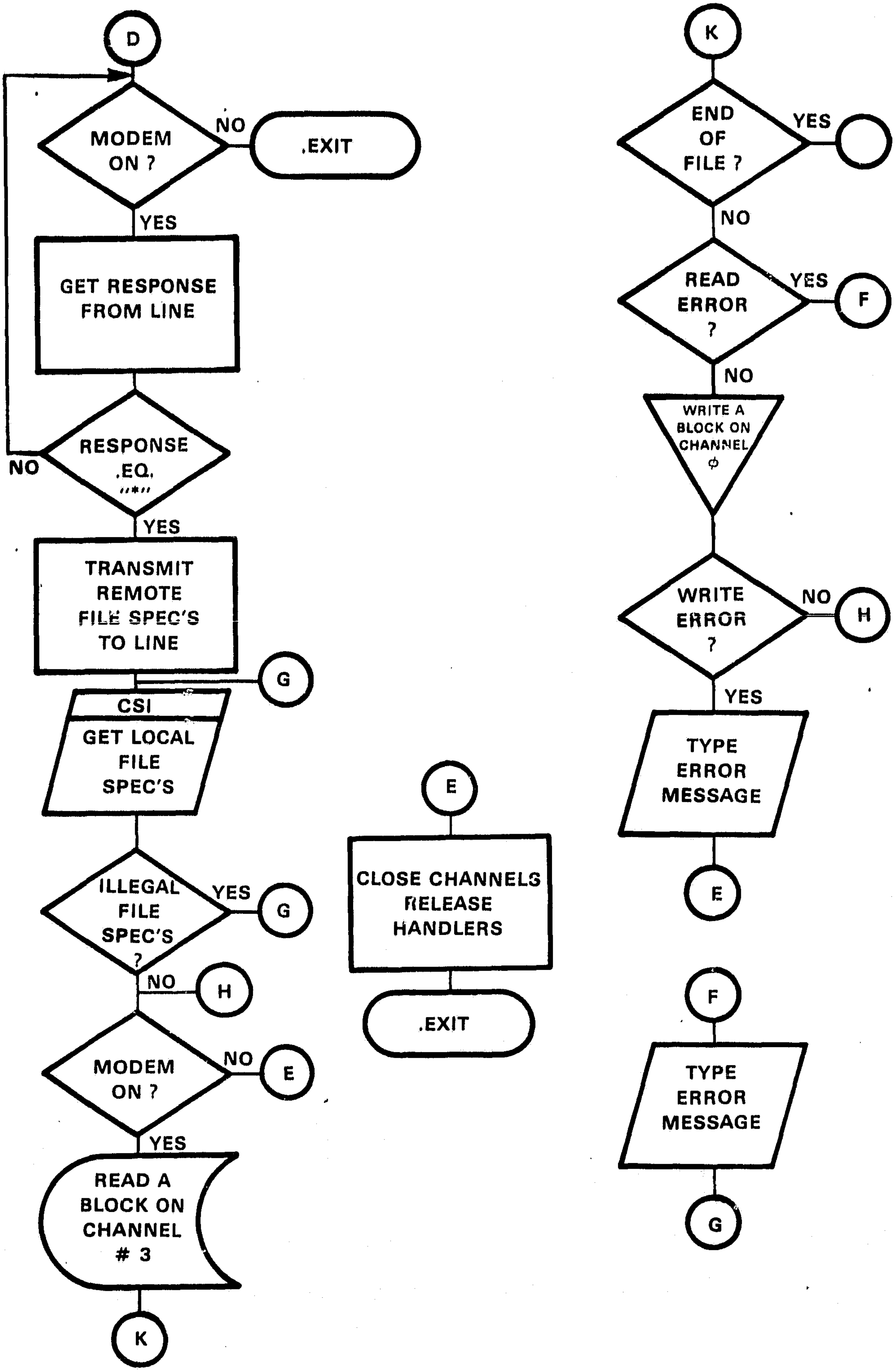
LINKING

```
RCOPY = RCOPY
RPIP  = RPIP
```


TRAP
SET TRAP FLAG FOR THIS TRAP
RTI


RCOPY
RPIP
S
GET REMOTE FILE SPEC'S
ILLEGAL FILE SPEC'S ?
YES
S
NO
HELP REQUEST ?
YES
TYPE HELP FILE
S
NO
GET BLOCK COUNT
SAVE AT MEMORY LOCATION 1002₈
C


C
MODEM ON ?
NO
.EXIT
YES
WHO'S IN COMMAND ?
RPIP
TRANSMIT MONITOR CONTROL C COMMAND TO LINE
E
RCOPY
TRANSMIT FILE XFR PASSWORD TO LINE
D
GET RESPONSE FROM LINE
MODEM ON ?
NO
EXIT
YES
RESPONSE .EQ. "."
NO
E
YES
TRANSMIT MONITOR "RUN PIP" COMMAND TO LIVE
D


D
MODEM ON ?
NO
.EXIT
YES
GET RESPONSE FROM LINE
RESPONSE .EQ. "*"
NO
YES
TRANSMIT REMOTE FILE SPEC'S TO LINE
G
CSI
GET LOCAL FILE SPEC'S
ILLEGAL FILE SPEC'S ?
YES
G
NO
H
MODEM ON ?
NO
E
YES
READ A BLOCK ON CHANNEL # 3
K
E
CLOSE CHANNELS RELEASE HANDLERS
.EXIT
K
END OF FILE ?
YES
NO
READ ERROR ?
YES
F
NO
WRITE A BLOCK ON CHANNEL φ
WRITE ERROR ?
NO
H
YES
TYPE ERROR MESSAGE
E
F
TYPE ERROR MESSAGE
G

PROGRAM: DTOASC OCTA

PURPOSE:

DTOASC: Decimal to ASCII Conversion process
OCTA: Octal to ASCII Conversion process.

CALLING
SEQUENCE:

DTOASC

```
        .GLOBL  NDTA,NDTAB
        MOV     Value,NDTA   ;NData=Decimal Value
        JSR     PC,DTOASC
        NOP
        .MOV    #NDTAB,R0    ;R0=FWA of 3 word ASCII Decimal Value
        .
        .
        .
        .END
```

OCTA

```
        .GLOBL  BDATAB,CDATAC
        MOV     Value,BDATAB; BDATAB=Octal Value
        JSR     PC,OCTA
        NOP
        MOV     #CDATAB,R0;R0=FWA of 3 word ASCII Octal Value
        .
        .
        .
        .END
```

Special Conditions: None

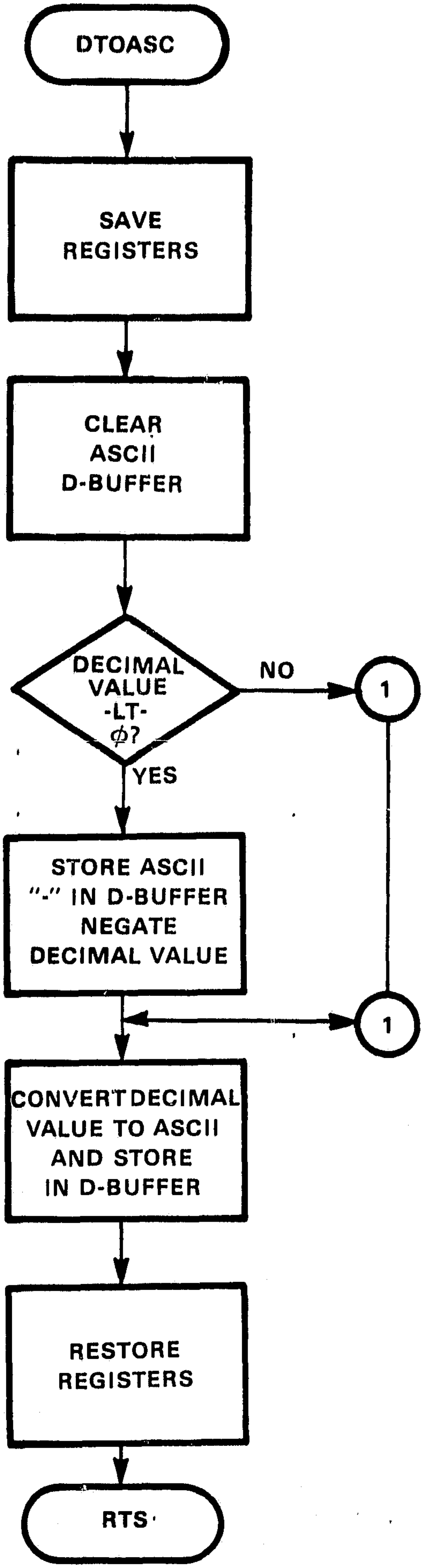
Special Requirements:

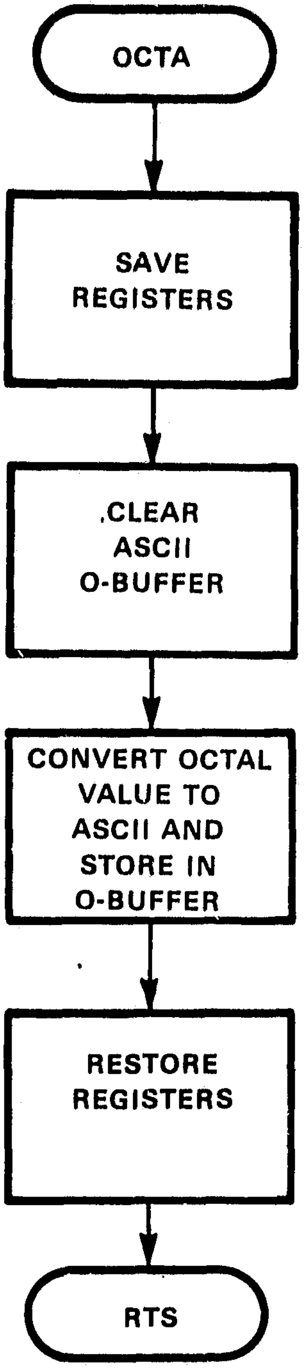
DTOASC, Floating Point Hardware

External Requirements: None

Globals:

DTOASC - NDTA, NDTAB
OCTA - BDATAB,CDATAB


DTOASC
SAVE REGISTERS
CLEAR ASCII D-BUFFER
DECIMAL VALUE -LT- ϕ?
NO
1
YES
STORE ASCII "-" IN D-BUFFER NEGATE DECIMAL VALUE
1
CONVERT DECIMAL VALUE TO ASCII AND STORE IN D-BUFFER
RESTORE REGISTERS
RTS


OCTA
SAVE REGISTERS
CLEAR ASCII O-BUFFER
CONVERT OCTAL VALUE TO ASCII AND STORE IN O-BUFFER
RESTORE REGISTERS
RTS

SECTION VII DEFINITIONS

DEFINITIONS

absolute address - A binary number that is assigned as the address of a physical memory storage location.

algorithm - A prescribed set of well-defined rules or processes for the solution of a problem; a program.

analog - Data characters which are not coded into discreet separate pulses or signal levels.

ASCII - The American Standard Code for Information Interchange, consisting of 128 7-bit binary codes for upper and lower case letters, numbers, punctuation, and special communication control characters.

asynchronous - A mode of operation in which an operation is started by a signal that the operation on which it depends is completed. When referring to hardware devices, it is a method in which each character is sent with its own synchronizing information. The hardware operations are scheduled by ready and done signals rather than by time intervals. In addition, it implies that a second operation can begin before the first operation is completed.

baud - (a) A unit of signaling speed equal to the number of signal events per second. (b) For asynchronous events, the unit of modulation rate corresponding to one unit interval per second. If, for example, the length of the unit's interval is 25 milliseconds, the modulation rate is 40 baud. Baud is frequently, though erroneously, used as a synonym for bits per second.

BCD - Binary Coded Decimal

bit - a binary digit.

bootstrap loader - A routine whose first instructions are sufficient to load the remainder of itself into memory from an input device and normally start a complex system of programs.

buffer - A storage area used to temporarily store information being transferred between two devices or between a device and a memory. A buffer is often a special register or a designated area of memory.

byte - The smallest memory addressable unit of information in a computer system. A byte is equivalent to 8 bits.

calibration - The encoding, per channel, of standard voltages by which data may be sequentially adjusted for slope and offset errors.

carrier - Two sets of frequencies used to transmit data over lines where Frequency 1 represents a '0' and Frequency 2 represents a '1'.

console module - A hardware device which is programmable through specified hardware registers.

data base - A collection of interrelated data items organized by a consistent scheme that allows one or more applications to process the items without regard to physical storage locations.

diagnostic - Pertaining to the detection and isolation of malfunctions or mistakes.

digital - Data characters which are coded into discreet separate pulses or signal levels.

dump - To copy the content of all or part of core memory, usually onto an external storage medium. Also the copy so produced.

FWA - First Word Address

function code - The name of an algorithm contained in the system software.

global - A value defined in one program module and used in others. Globals are often referred to as entry points in the module in which they are defined, and externals in the other modules which use them.

hexadecimal - Pertaining to the Base 16 number system.

interrupt - A signal which, when activated, causes a transition of control to a specific location in memory, thereby breaking the normal flow of control of the routine being executed. An interrupt is normally caused by an external event such as a "done" condition in a peripheral. It is distinguished from a trap which is caused by the execution of a processor instruction.

Julian day - Yearly day count 1 to 365.

MCR (Monitor Control Routine) - The executive routine that allows the user to communicate with the system using an on-line terminal device. MCR accepts and interprets commands typed on the terminal keyboard and calls appropriate routines to execute the specified requests.

msec - milliseconds, (10^{-3}) seconds.

mux - (multiplexing) Use of one communications line circuit for two or more simultaneous data paths.

parity - When a binary bit is appended to a group of bits to make the sum of all the bits always odd (ODD Parity) or always even (EVEN Parity).

purge - to delete

SPS - Samples per Second

TPERR - Trap Error flag

trap - The central processor immediately going to a fixed memory location because of an error or programming condition.

TTY - teletype

XXX_8- Representation of an Octal number.

TYPICAL 24 HOUR DATA DUMP

DATA TAKEN 29TH OF OCTOBER 1981

FROM THE MODOA @ CLAYTON, N.M.

ORIGINAL PAGE IS
OF POOR QUALITY

CHANNEL CONVERSION IN UNITS OF MEAS.

6 MIN AVG	CH1	CH2	CH3	CH4	CH5	CH6	CH7	
	POWER	WIND SPDM	WIND SPDN	NAC.DIR	YAW ERROR	BLADE ANGLE	KVARS	
IN HEX	KW	MPH	MPH	DEG	DEG	DEG	ON/OFF	
7F	-300	0.0	50.0	0	+90	+10	-300	
79	-270	2.5	47.5	18	+81	+ 5	-270	
73	-240	5.0	45.0	36	+72	0	-240	
6C	-210	7.5	42.5	54	+63	- 5	-210	
66	-180	10.0	40.0	72	+54	-10	-180	
60	-150	12.5	37.5	90	+45	-15	-150	
59	-120	15.0	35.0	108	+36	-20	-120	
53	- 90	17.5	32.5	126	+27	-25	- 90	
4C	- 60	20.0	30.0	144	+18	-30	- 60	
46	- 30	22.5	27.5	162	+9.0	-35	- 30	
3F	00	25.0	25.0	180	0.0	-40	0.0 (OFF)	
39	+ 30	27.5	22.5	198	-9.0	-45	+ 30	
33	+ 60	30.0	20.0	216	-18	-50	+ 60	ON
2C	+ 90	32.5	17.5	234	-27	-55	+ 90	
26	+120	35.0	15.0	252	-36	-60	+120	
1F	+150	37.5	12.5	270	-45	-65	+150	
19	+180	40.0	10.0	288	-54	-70	+180	
13	+210	42.5	7.5	306	-63	-75	+210	
0C	+240	45.0	5.0	324	-72	-80	+240	
06	+270	47.5	2.5	342	-81	-85	+270	
00	+300	50.0	0.0	360	-90	-90	+300	

ORIGINAL PAGE IS
OF POOR QUALITY

DAY 4 TO FOLLOW: 29 OCTOBER, 1981

CHANNEL1

```
00003F3F3F3F3F3F3F3F3F3E3332303438373C3734383C3F3F3F3F352D2C3034393B3C3A37312C282B2A302E2D2F322F2F33363A3E3D3D3C3C3C3E3A313131382F
3536373232333322F313222E3232353D24313B403F3A33363033342D292726241C2227131A161A1C1A1817161718161917161618181A1E1A1616171617161616161716
161616161616161616161616161616161616161616161516151715161616161615161616161616161616161616181716171616171816151616161616171A1C1A171719
1A191F21211E2022231D1B191618171A181A1C1F1E20201F1C1C1D1D1C1D21201C1E18171717191C1B20
```

CHANNEL2

```
75747374747474747374736E6C6C6B6A6C6F6D6F6D6B6C6F7171716E6B69696A6C6E6F6F6F6D6B696769686B6A6A6A6B6A6A6B6C6E70707070707070716F6B6B6B6D6A
6B6B6C6B6B6C6B6A6B6B6B696B6B6D706C6B6F73706D6B6C6A6B6B69676666676466666626160626362605C5D5F615F635F616060606161615F5B5C5C5F5B5A595A5B5C
59585E5757585857585B5A5A5B5B595A5A5B5B595D56585C5553535656595B5957595357575956555658585E5C5F5E5E5F5C606261605F5F5E5F5F6062626363626162
6362636464646464656464636261636364636464656565656564646464646463666564646362616262636364
```

CHANNEL3

```
13151613101214141513151E1E1D1F23201C1D191B1D1A161819191D202526231E181716181C2022242423202222221F22211F1F1817151616171516171E1E1E1A22
211C1C1F1F1E1E21201E211E1F1C151D1F1912191E1F1C211E1E23272526272B27252E32353230323737 3B3531352E3433333 2531302E313B4140413E42434648413D
3F4047424746444A403F41413E42494446424142414A433B4A52514B453E4049454E4843454 74C4848413D3E313337393B3735333939373939393937333132363635
3435312F2F2E2D2C2A2C2E323330312E2F2E2C2B2D2C2B2B2E2E2E2F2F2F2D2D312F3335363534322D2C
```

CHANNEL4

```
2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2C26262829292929292B2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2F2A2A232628282929 2D2928282828 2A
2C27262626262626262626262626282C2C2D2D2C2625303131313131313131313131313131313131313131313131313131313131313131313134393838383838383439393939393939
39393939393939393939393939393939393939393939393939393939393939393939393939393F404040404040404040404040404041404141414141414141414141414141
41414141403E3E3E3E3E3E3E3E3E3E3E3E3E404141414141414141414141414141414141414141413D3D
```

CHANNEL5

```
4C50554D4545494C47433E3C3C3D3E40424242473E393D3F3F40403A394243443F3D3F3E423F3E3B3B3C3C3F3E3E3E41414042453F49383C424038453D3A393A3E3B
403E423E3E3D3B3E413C3B3B383E403F3F463B293F413F40404241403E3D3C3B3B3E3C403E424041433B3A3A3C3B3B363737323439403B414646453E43454849493D
3B3C3B3E3C3D3F3F3D3F414145414B4545463E3E3D413E3F43423F4145423A3F3C393D3E3A334044434240403D3D40414140403E40403F3F3F414143434242424243
45444545443F3F3F3E3C3C3C3C3B3A3B39393B3C3D3F3E3E3D3E3F40404142424243414 424143453C3F
```

CHANNEL6

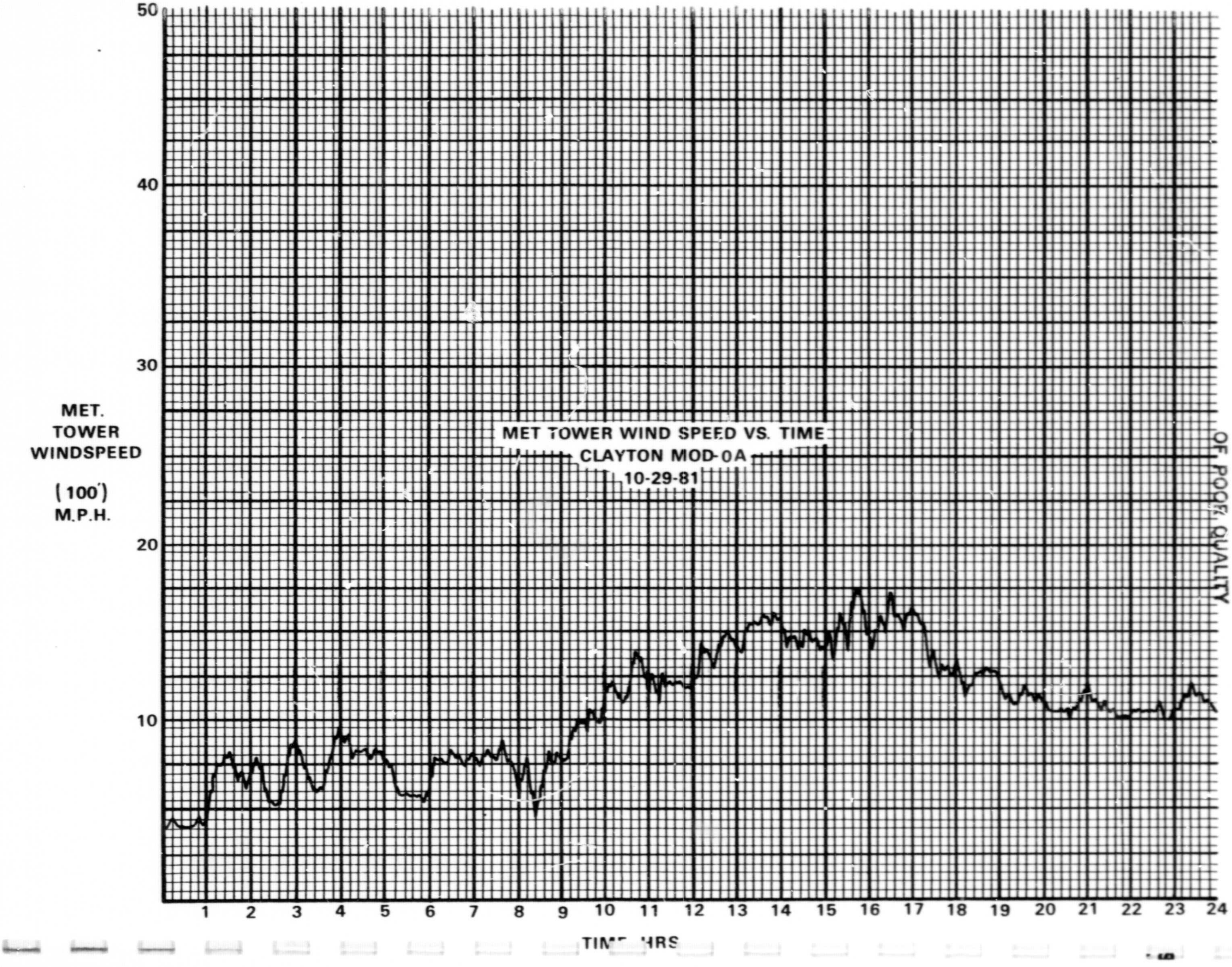
```
0000000000000000000000000046F71717171717171717171717171200000056F71717171717171717171717171717171717171717171717171717171717171717171
71717171717171717171717171717171717171716300060717171717171717171717171717171706F6E706F6E6E6B6A6C6E6D6F6D6D6D6D6E6F706F6C6B6B6A6C6A6A6A696C6A
686865686666666865686A69696C696A6A696B69686866686B67646366696B68666666365686666666867696B6B6F6E6D6D6C6E6F706D6D6E6D6D6D6E6F717171707070
71717171717171717171717171706F7070717071717171717171717171717171717171717171707171717171
```

CHANNEL7

```
28300031040C321A0003411173E3F3F3F3F3F3F3F3F3F3F3F08202A303E3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F
3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F0F173F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F
3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F
3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F
```

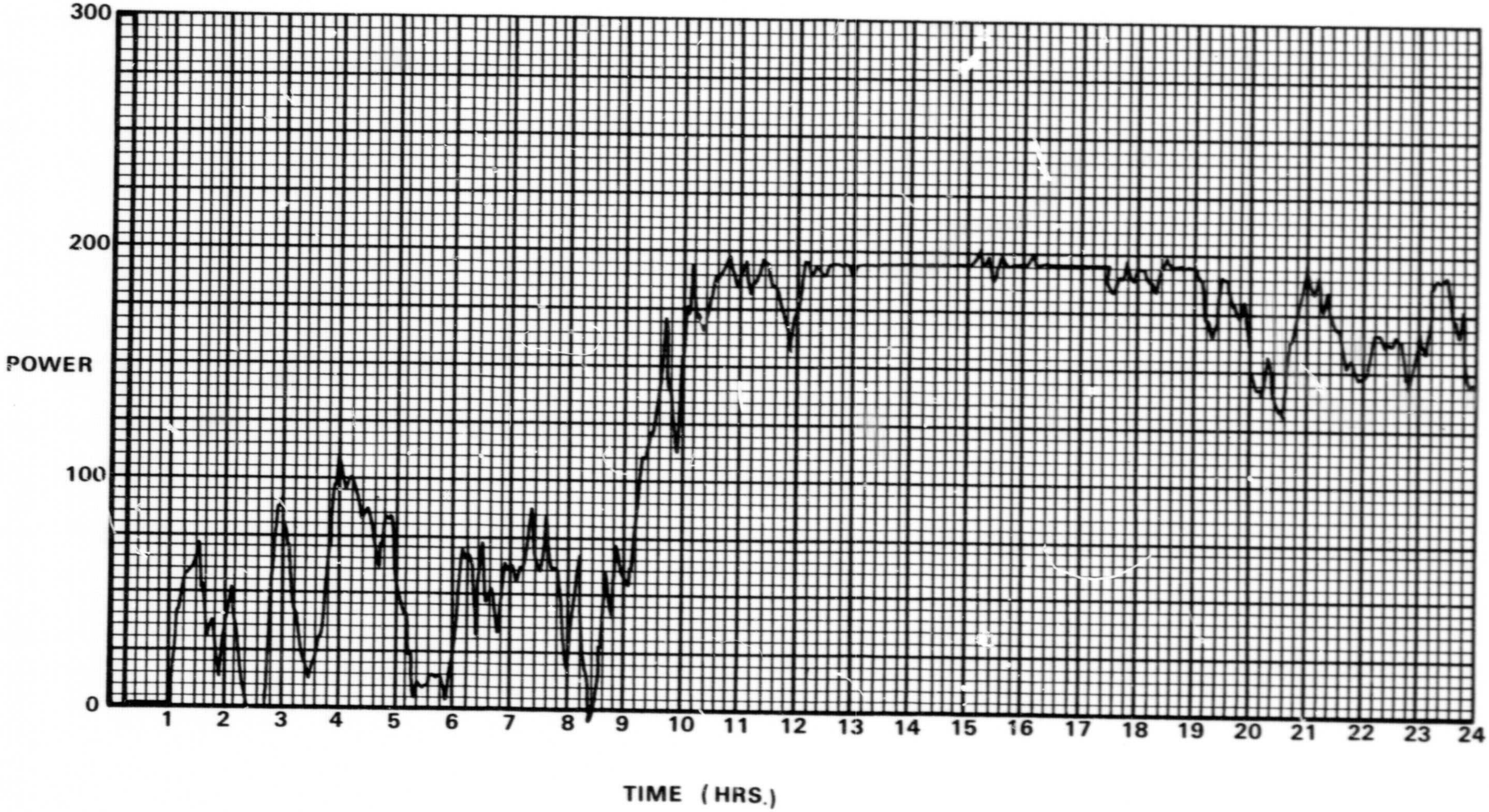
CHANNEL8

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```



MET TOWER WIND SPEED VS. TIME
CLAYTON MOD-0A
10-29-81

POWER 10/29/81
CLAYTON MOD-0A



ORIGINAL PAGE IS
OF POOR QUALITY