TRW Library

25990-H040-RO-00

NASA CR-
147451

TRW NOTE NO. 74-FMT-946

SHUTTLE

TASK JSC/TRW 542

PROGRAM MANUAL FOR THE SHUTTLE ELECTRIC POWER SYSTEM ANALYSIS COMPUTER PROGRAM (SEPS)

VOLUME I OF PROGRAM DOCUMENTATION

JUNE 1974

Prepared by

ELECTRICAL POWER SECTION

Prepared for

MISSION PLANNING AND ANALYSIS DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

JOHNSON SPACE CENTER

HOUSTON, TEXAS

NAS 9-13834

18692-A SSC

TRW NOTE NO. 74-FMT-946

SHUTTLE

TASK JSC/TRW 542

PROGRAM MANUAL FOR THE SHUTTLE ELECTRIC POWER SYSTEM ANALYSIS COMPUTER PROGRAM (SEPS)

VOLUME I OF PROGRAM DOCUMENTATION

JUNE 1974

Prepared for
MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
JOHNSON SPACE CENTER
HOUSTON, TEXAS
NAS 9-13834

Prepared by: R. W. Bains
H. A. Herwig
J. K. Luedeman
E. M. Torina

Approved by R. A. Mintz, Head
Electric Power Section

Approved by W. B. Warren, Acting Manager
Electronic Systems
Engineering Department

Approved by D. M. Austgen, Manager
Mission Trajectory Control
Program

ABSTRACT

The Shuttle Electric Power System Analysis Computer Program (SEPS) was developed by TRW under JSC/TRW Task 542 for the Consumables Analysis Section of the Mission Planning and Analysis Division. The SEPS program has two major uses; first, to perform detailed load analysis including predicting energy demands and consumables requirements when the shuttle electric power system is operated and perturbed in accordance with premission flight plans; and second, to perform parametric and special case studies on the Shuttle electric power system. As an additional feature, the SEPS program can be and has been used to analyze the ASTP Apollo electric power system. No program changes are required to use the SEPS program for analysis of the ASTP Apollo electric power system.

The SEPS Computer Program is written in FORTRAN V for use on the UNIVAC 1108 under the EXEC II operating system.

Documentation of the SEPS program is divided into two separate volumes.

VOLUME I - Program Manual contained herein

VOLUME II - User's Manual contains the information necessary for a user to adequately understand and use the SEPS Computer Program

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

1.0 INTRODUCTION

1.0 INTRODUCTION

This document contains information pertaining to the Program Manual, Programmer Guide, and Program Utilization of the Shuttle Electrical Power System (SEPS) computer program. The main objective of this manual is to provide the information necessary to interpret and use the routines comprising the SEPS program.

The subroutine descriptions are divided into four categories; control, Phase I, Phase II, and analysis routines. The subroutine descriptions include the name, purpose, method (if applicable), variable definitions and logic flow.

The SEPS User's Manual provides the information necessary for a user to adequately understand and use the SEPS computer program.

2.0 PROGRAM DESCRIPTION

2.0 PROGRAM DESCRIPTION

2.1 PURPOSE

The TRW Shuttle EPS Analysis Computer Program (SEPS) was developed for use as a premission evaluation tool. The purposes of the program are to (a) predict EPS performance and EPS consumables usage when the system is operated and perturbed in accordance with premission flight plans, and (b) perform parametric and special case studies on the Shuttle EPS.

2.2 SEPS PHASE I

2.2.1 Discussion

The SEPS Computer Program Phase I, using a mission event timeline, develops an electrical load profile and provides subsystem and mission analyses of the power and energy demands for Shuttle missions. The analysis is based on a 28 VDC load bus voltage. The program utilizes a data base which describes all Shuttle electrical power consuming equipment in terms of power requirements and relating all the equipment to subsystems. This data combined with a desired mission event timeline provides the basis for the output interface tape consisting of event time point data and a listing of the activated components. The interface tape is utilized as the input driver for the Phase I COMUSE analyses and Phase II. The Phase I COMUSE analyses are discussed in more detail in the User's Manual, Section 2.1 (Formatted Printout Description).

The Phase I output and Phase I COMUSE output analyses have been adjusted to include an assumed 4% average line loss and an inverter efficiency of 80%. These data have been hard coded into the program and would require a change in Subroutine JVSEPS to revise these assumptions. The respective words are RESLOS for the line loss factor and PFEFF for the inverter efficiency. The line loss factor and inverter efficiency are not used in the Phase II analyses.

2.3 PHASE II

2.3.1 Discussion

The SEPS Phase II program integrates the various math models that define the operating characteristics of the power sources, distribution and equipment of the Shuttle electrical power system. This provides the capability of simulating the total electrical power system with which system design and design/mission requirements compatibility can be analyzed and parametric studies performed. Another capability of SEPS Phase II is the fuel cell cryogenic requirements which result from the mission profile electrical power demand and operating procedures.

The SEPS Phase II program utilizes the Phase I interface tape to provide the electrical load profile and active components. The user also has the capability to change or modify the configuration or input data through the use of an input card alter deck.

2.3.2 Math Models

In order to accomplish the SEPS Phase II capabilities several math models were required. A description of the math models, their intended use, and primary subroutine follows.

EPS Distribution Circuit Math Model - This model is a representation of the Shuttle dc electrical power distribution and control system. Using a node analysis technique and the load profile contained on the Phase I interface tape, the distribution circuit model will determine the system bus voltages and currents and the load voltages and currents. The primary subroutine is DCSOLV.

Fuel Cell Math Model - This model is a representation of the Shuttle 7 KW average, 12 KW peak fuel cells. The fuel cell model is called by the distribution circuit model to provide source voltage as a function of source current. The fuel cell model also provides the cryogenic use rates to the cryogenics model. The primary subroutines are FUELIV and FUCLTM.

Inverter System Math Model - The inverter model is a simplified representation of the Shuttle 9 inverter/3 phase ac system. This model will calculate the inverter no load and load losses and reflect the ac inverter loads to the dc distribution system. The primary subroutine is ACINVT.

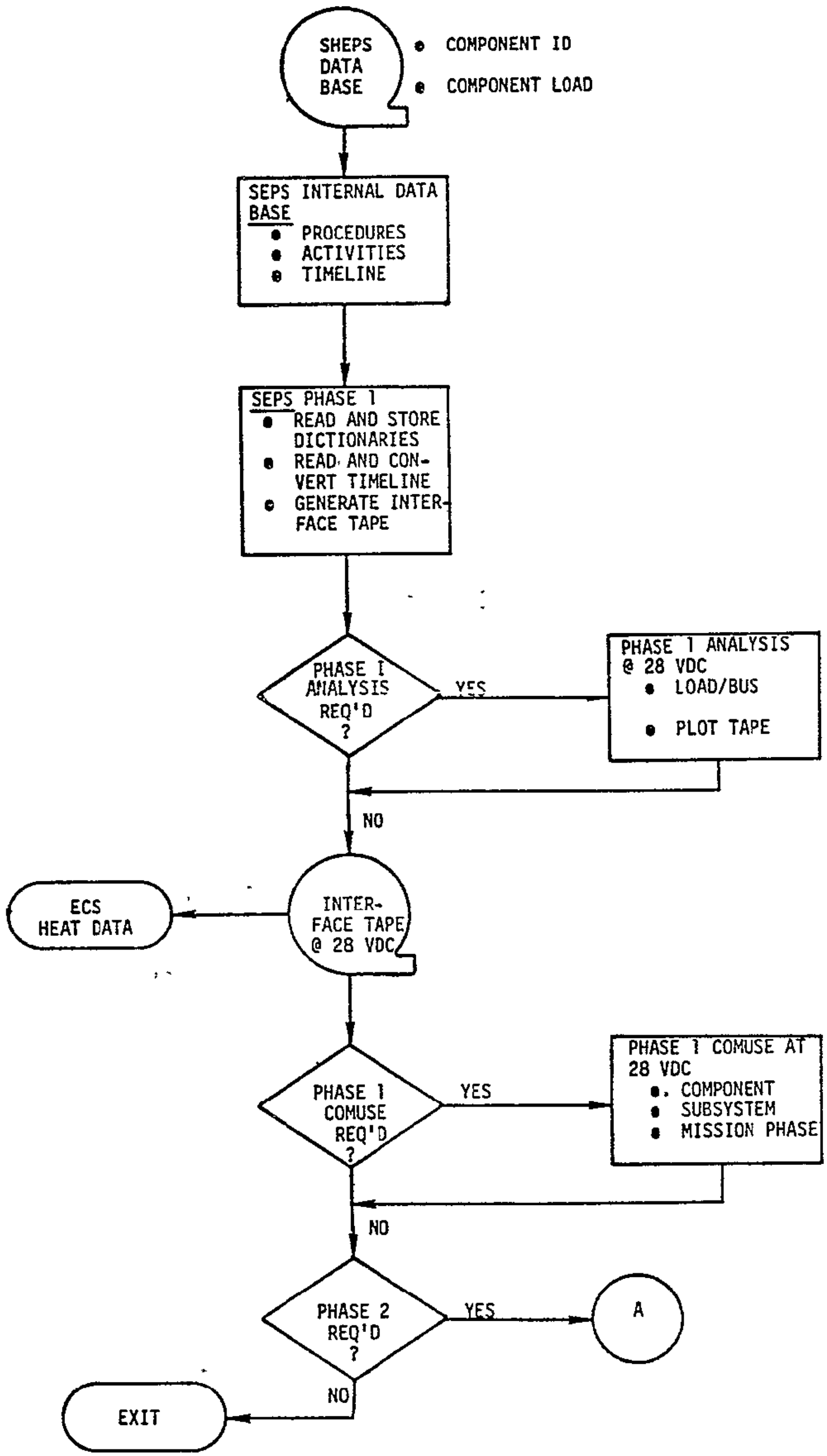
Constraints Model - The constraints model will provide for automatic program checking and flagging of distribution, power source, and reactant storage system constraint violations. As actual performance, test and limit data becomes available, the constraints model can be updated. The primary subroutine is REDLIN.

2.3.3 Phase I/Phase II Interface

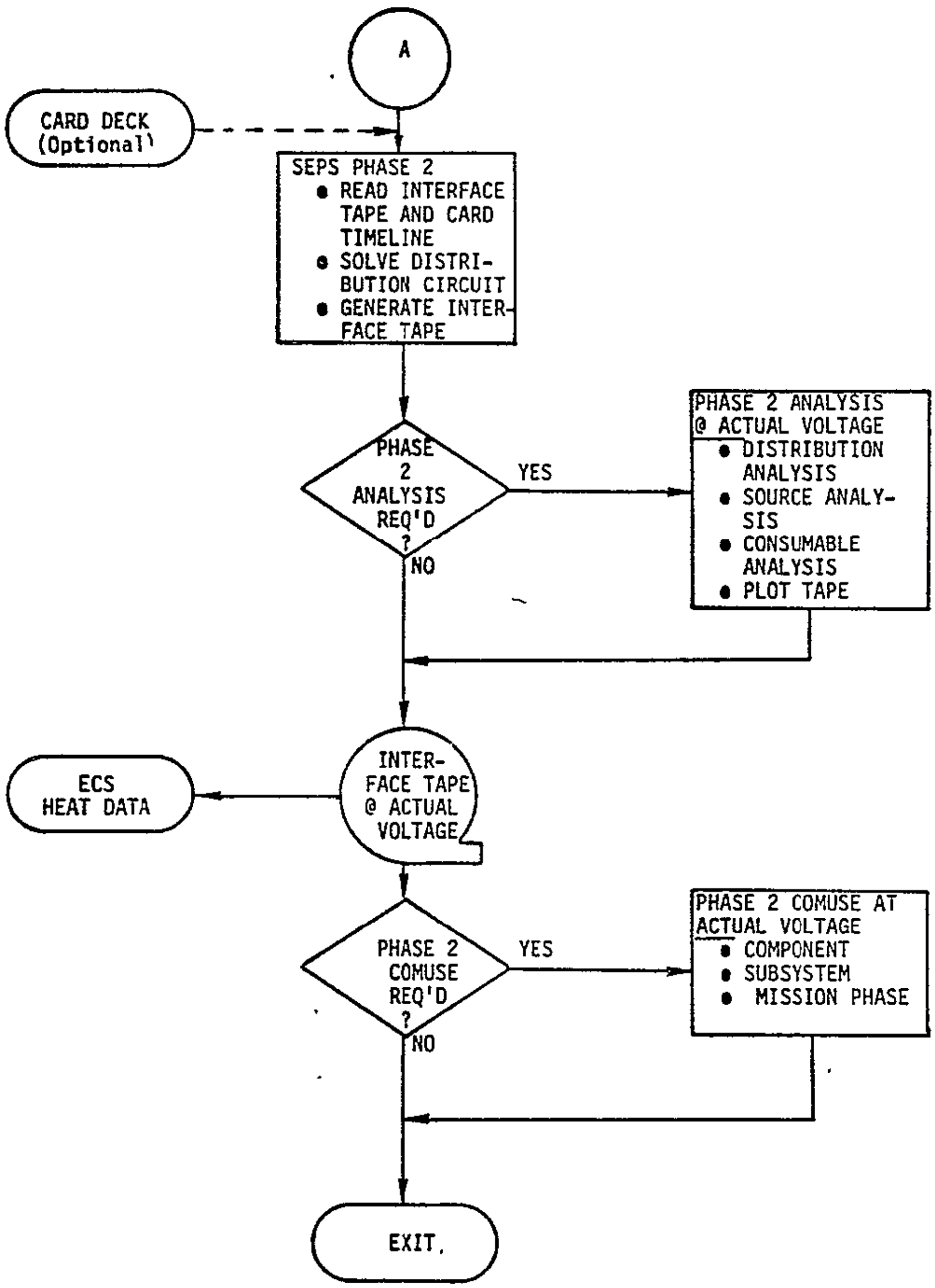
2.3.3.1 Discussion

The SEPS Phase I/Phase II interface has been designed to allow maximum utilization of analysis subroutines, output subroutines, and user interface through program option and control cards. Section 2.3.3.2 flow charts the Phase I/Phase II program. The flow chart shows a commonality in program output and the use of the same analysis subroutines for both Phase I and Phase II. The user through program option and control cards can either allow or suppress virtually any portion of the combined Phase I/Phase II program outputs. The output interface tape of Phase I is the input driver for Phase II. Modifications to the interface tape data can be inserted by the user through an input card deck.

2.3.3.2 Functional Flow Diagram



2.3.3.2 Functional Flow Diagram (Continued)



2.4 EXTERNAL PROGRAM INTERFACES

To facilitate the SEPS program in meeting its defined requirements, the following interfacing programs have been developed:

1. CIFLIS

   This program is used to list a given number of files from a card image tape.

2. CMPDAT

   This program is used to give a component and/or mission phase comparison of two interface tapes

3. CREDJR

   This program is used to make changes to the component definition card image tape file.

4. FILDUP

   This program is used to duplicate card image files.

5. JVMMPS

   This program is used to concentrate several timeline files together and to time order the resulting file.

6. NEWHLP

   This program is used to construct a tape able to be plotted showing user designated component's time history of operation.

7. STLPLT

   Generates CALCOMP plots from the unformatted interface output tapes.

8. WLCCIT

   This program is used to create or alter a card image tape file, as a by-product the file is listed and each entry is numbered.

3.0 SUBROUTINE DESCRIPTION

3.0 SUBROUTINE DESCRIPTION

3.1 CONTROL SUBROUTINE

3.1.1 Subroutine: JVSEPS

PURPOSE: This routine controls the SEPS program execution.

METHOD: This routine controls the following functions:

1. Determines total area of random access available
2. Reads the option-units cards
3. Reads the abort time
4. If applicable, calls for the mission phase definition cards to be read
5. If applicable, calls for the compacted component dictionary to be read for use in a Phase II only run utilizing an interface tape
6. Controls the execution of
   a. Phase I
   b. Phase I COMUSE
   c. Phase II
   d. Phase II COMUSE

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.1.1. See Appendix for definition of all variables.

```
INCLUDE STRAGA
INCLUDE STRAGI
INTEGER QQASG
DIMENSION IFLAG(40)
DIMENSION TITLT(2,20)
DIMENSION TITLO(3,40)
DIMENSION NTAPE(20)
DIMENSION ALPHA(29)
```

```
DIMENSION ALPHAT(20)
```

```
DATA TITLT /
"COMPONENTS  ","PROCEDURES  ","ACTIVITIES  ","TIMELINE    ",
"FIXED DATA  ","RESTART OUT ","RESTART IN  ","INTERFACE 1 ",
"PLOT 1      ",'INTERFACE 2 ",'PLOT 2      ","DICTIONARY  ",
"CIRCUIT DEF ","            ","            ","            ",
"            ","            ","            ","            "/
```

```
DATA TITLO /
"EXECUTE PHASE 1   ","EXECUTE PHASE 2   ","PLOT ON PRINTER 1 ",
"WRITE RESTART TAPE","USE 3 PT LOAD DATA","                  ",
"                  ","                  ","                  ",
"PRINT EACH INPUT 2","                  ","                  ",
"                  ","                  ","CONSTRAINT ANAL  2",
"                  ","                  ","                  ",
"                  ","PHASE 2 COMUSE    ","                  ",
```

```
"                  ","                  ","                  ",
"MISSION PHASE AN 2","                  ","                  ",
"                  ","SUPPRESS ANL PRT 1","PHASE 1 COMUSE    ",
"ECHO COMPONENTS   ","ECHO PROCEDURES   ","ECHO ACTIVITIES   ",
"ECHO TIMELINE     ","SUP COMUSE CMP ANL","AT MP LIST ACT CMP",
"MISSION PHASE AN 1","SUPPRESS CMP CHNG ","SUPPRESS LOAD/BUS ',
"SUPPRESS SUBSYSTEM"/
```

```
DATA ALPHA /
"A    ","B    ","C    ","D    ","     ","     ","E    ",
"F    ","G    ","H    ","I    ","J    ","K    ","L    ",
"M    ","N    ","     ","O    ","P    ","Q    ","R    ",
"S    ","T    ","U    ","V    ","W    ","X    ","Y    ",
"Z    "/
```

```
DATA NTAPE /20*6H      /
DATA PWRTOT /-10./
```

CONT. ON PG 2

JVSEPS
PG 1 OF 7

FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS

ORIGINAL PAGE IS
OF POOR QUALITY

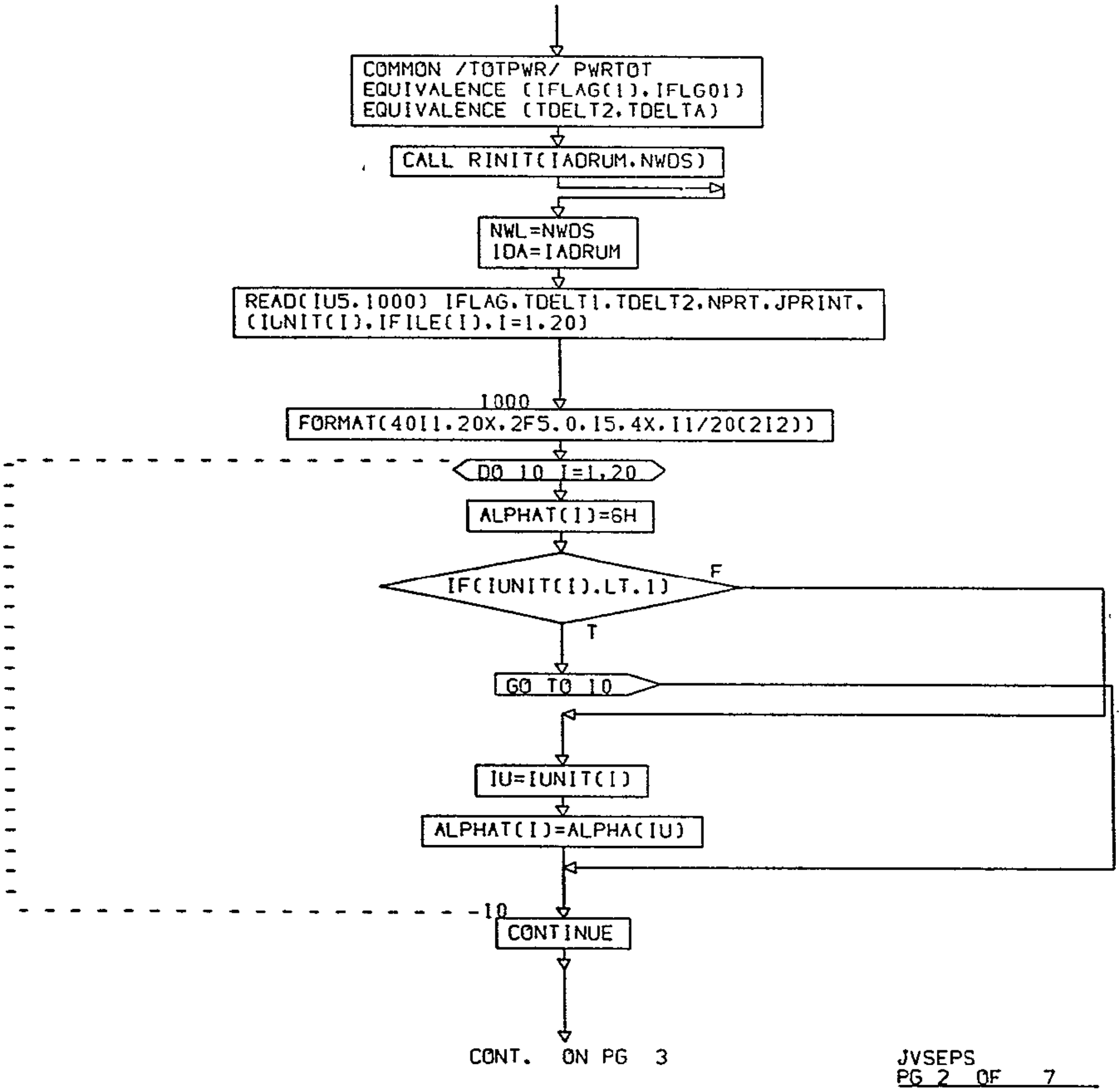
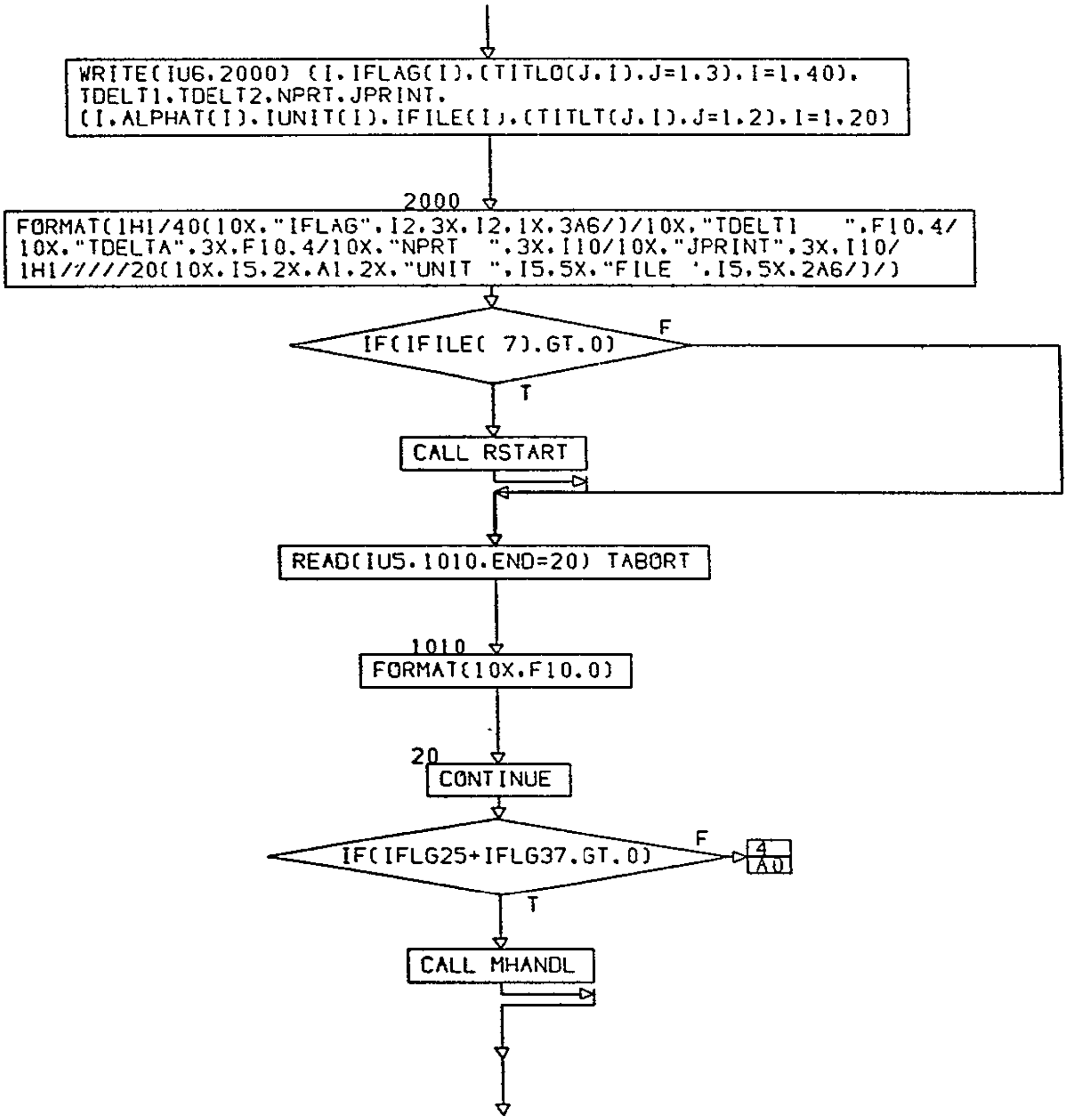


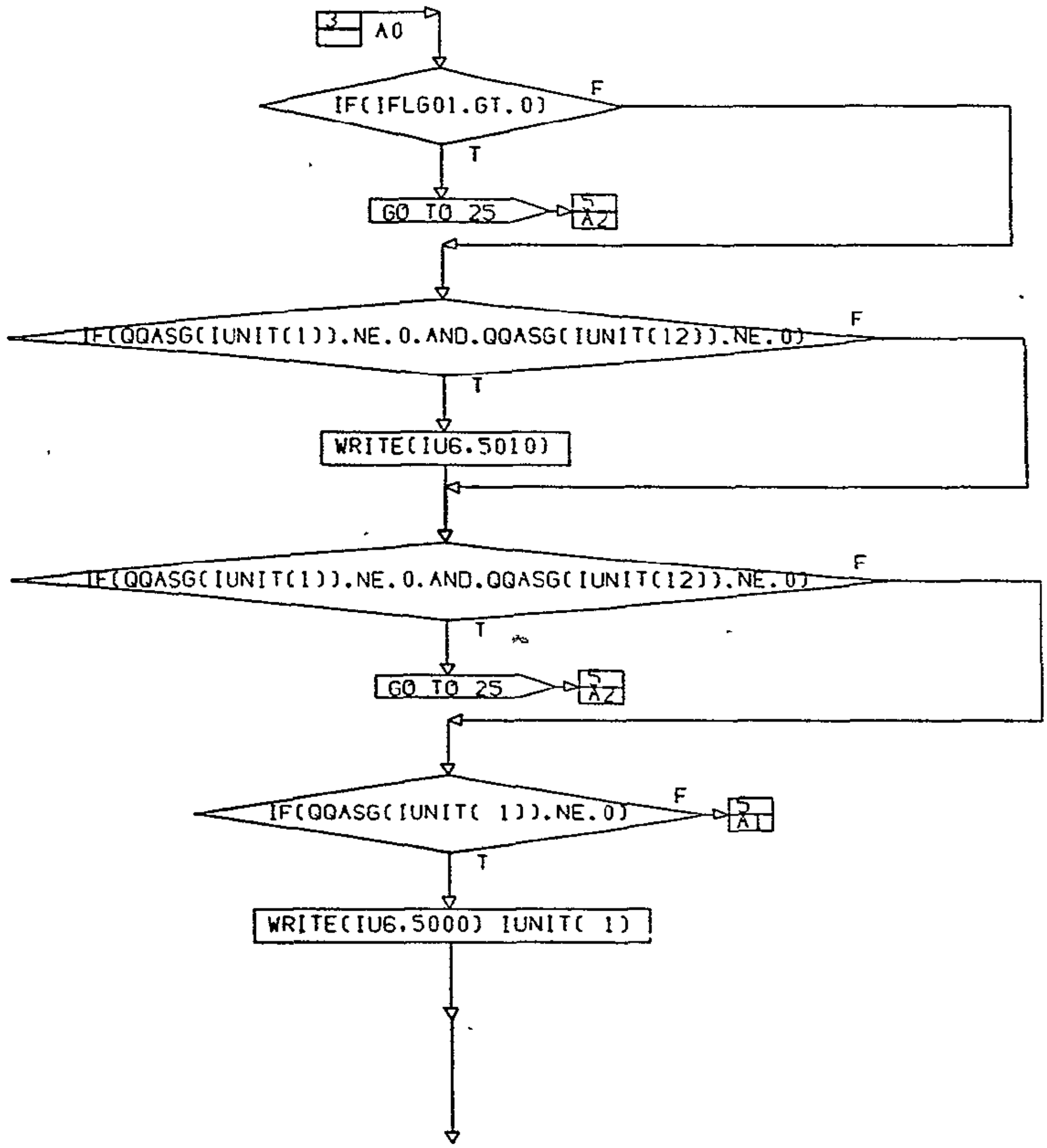
FIGURE 3.1.1 FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)



CONT. ON PG 4

JVSEPS
PG 3 OF 7

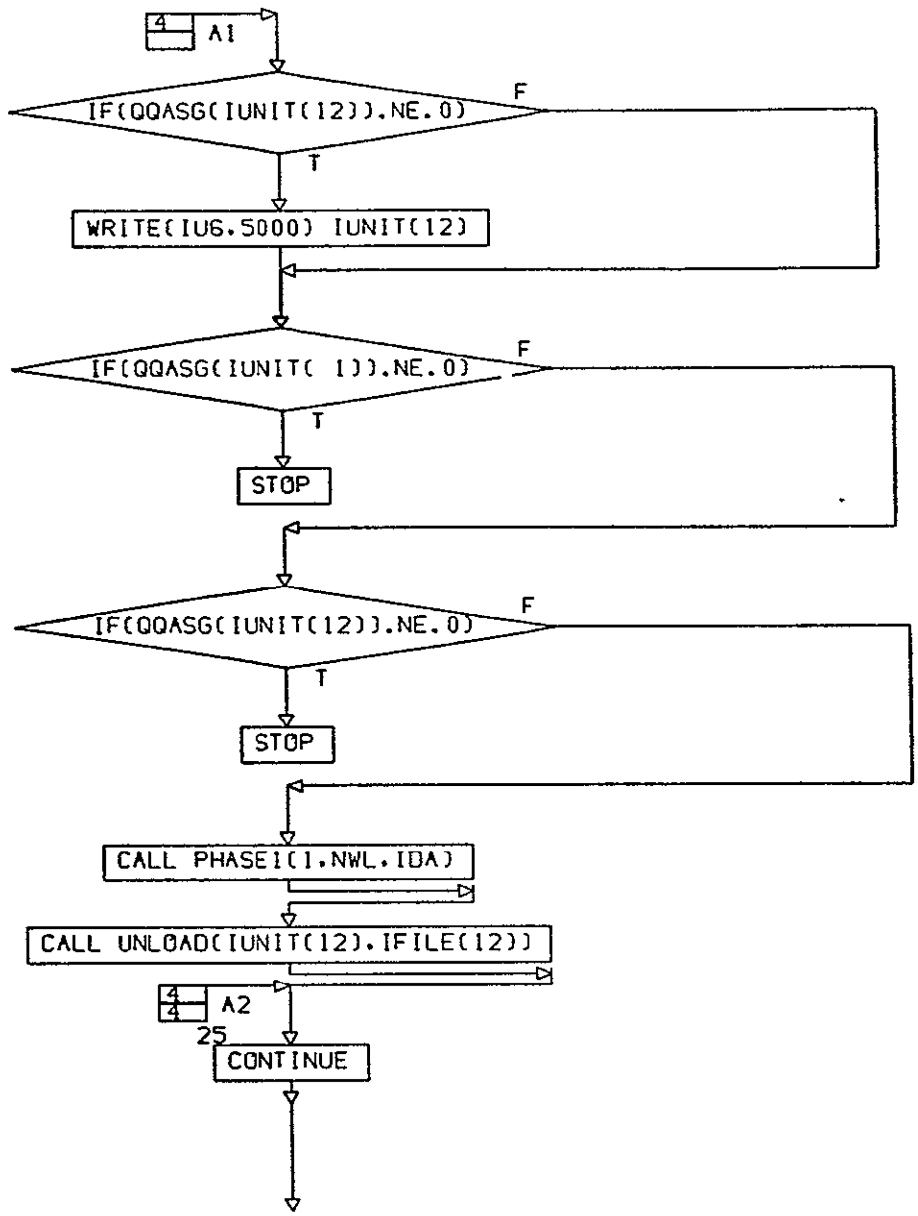
FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)



CONT. ON PG 5

JVSEPS
PG 4 OF 7

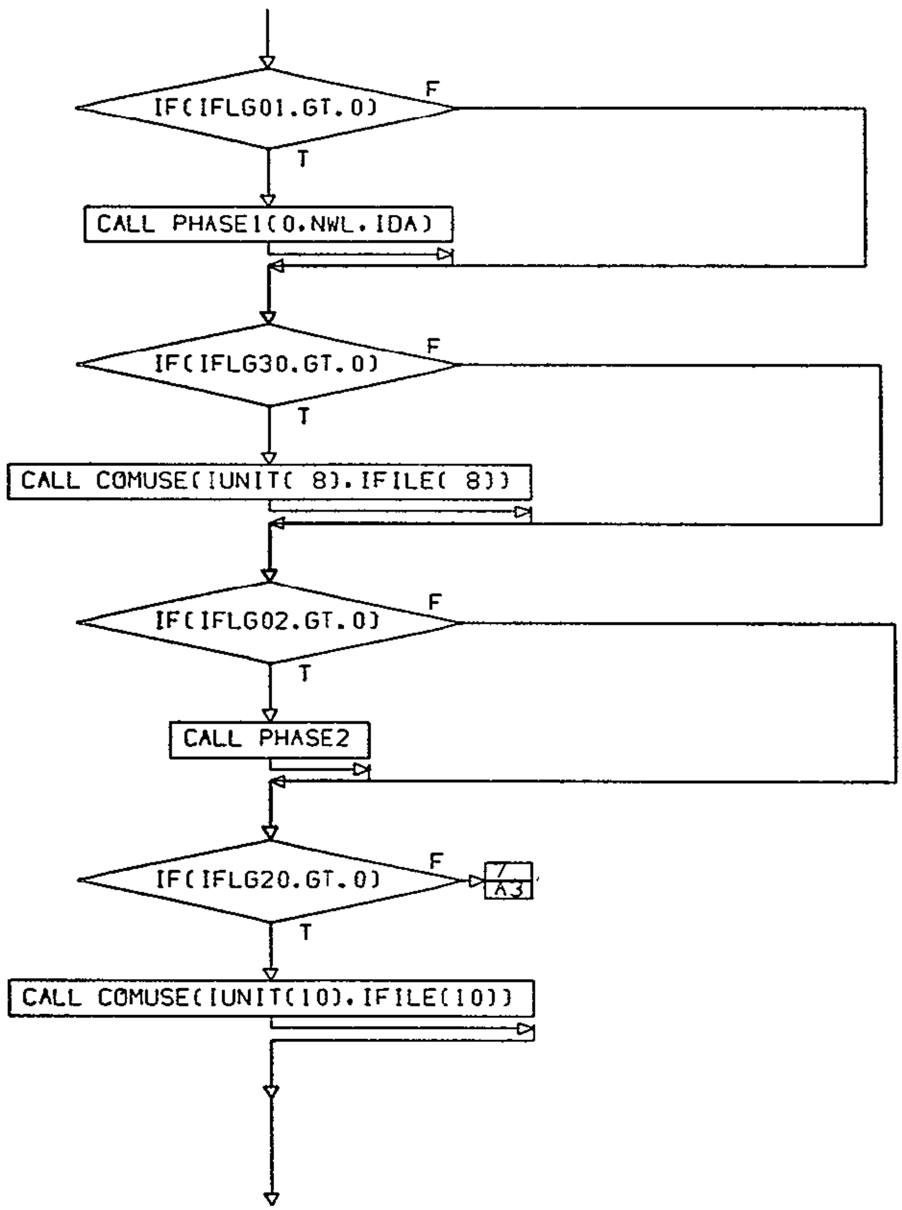
FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)



CONT. ON PG 6

JVSEPS
PG 5 OF 7

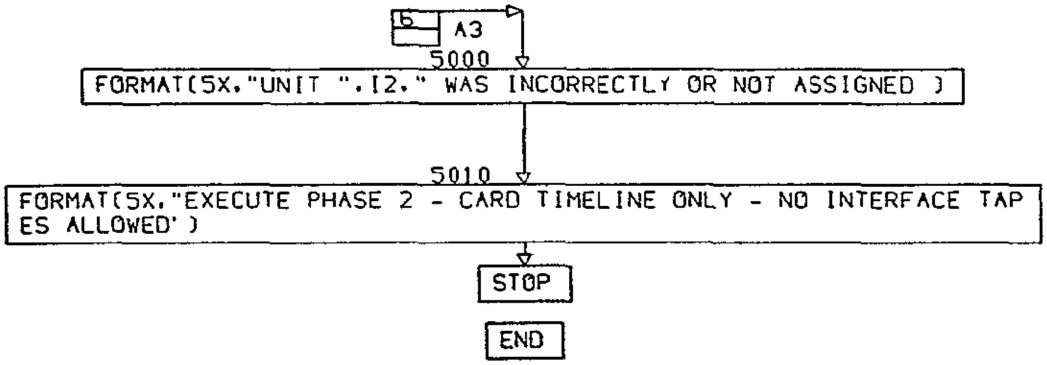
FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)



CONT. ON PG 7

JVSEPS
PG 6 OF 7

FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)



JVSEPS
PG 7 FINAL

FIGURE 3.1.1. FUNCTIONAL FLOWCHART OF SUBROUTINE: JVSEPS (CONTINUED)

3.2 PHASE I SUBROUTINES

3.2.1 Subroutine: PHASE1

PURPOSE: To control the creation of a load profile

METHOD: This routine controls the following functions

1. Creates a component dictionary
2. Creates a procedure dictionary
3. Creates an activity dictionary
4. Reads a timeline consisting of activities, procedures, components, switches, and cyclic elements and converts it to a component event timeline
5. Compacts out the unused components in the component dictionary
6. Analyzes the component event timeline

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.1. See Appendix for definition of all variables.

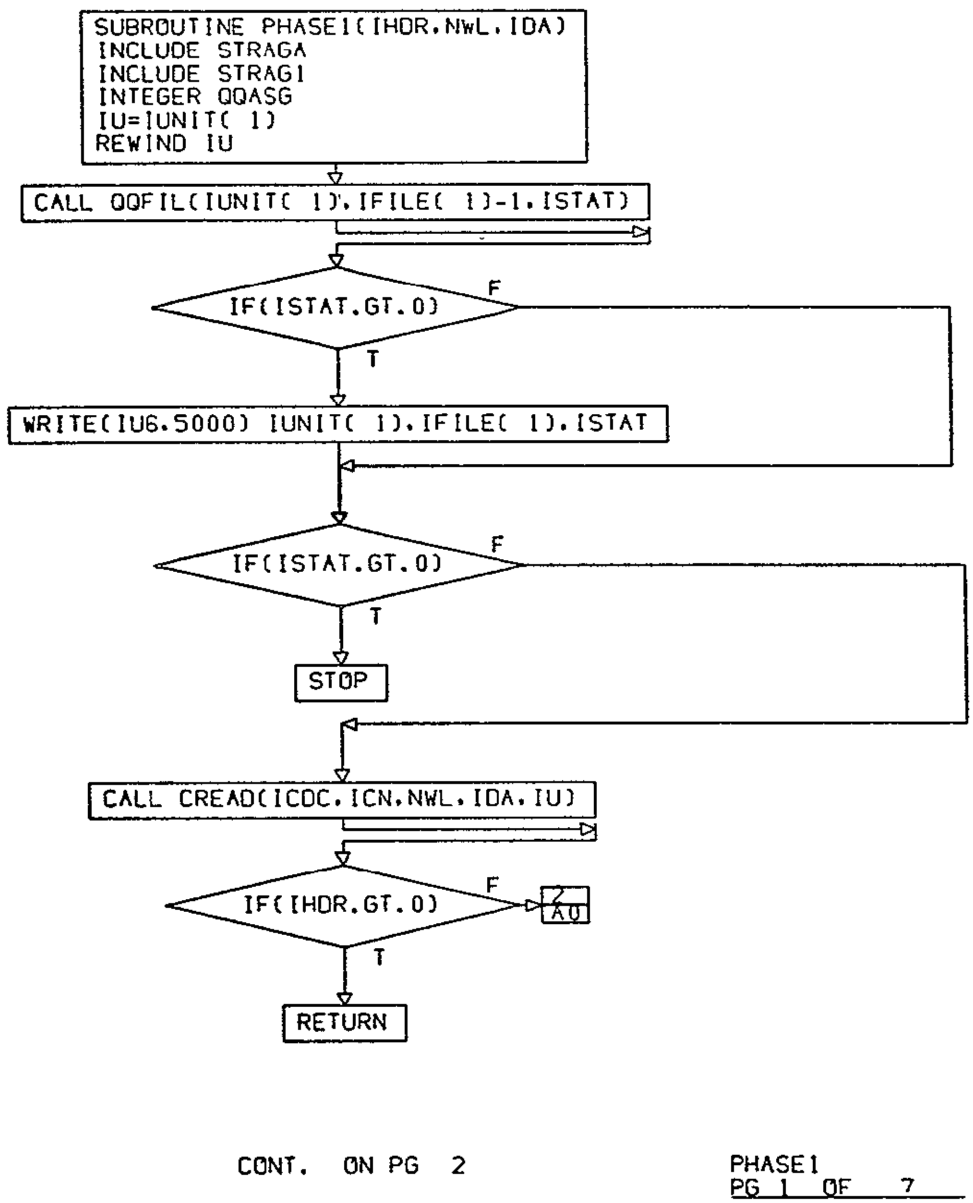


FIGURE 3.2.1 FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I

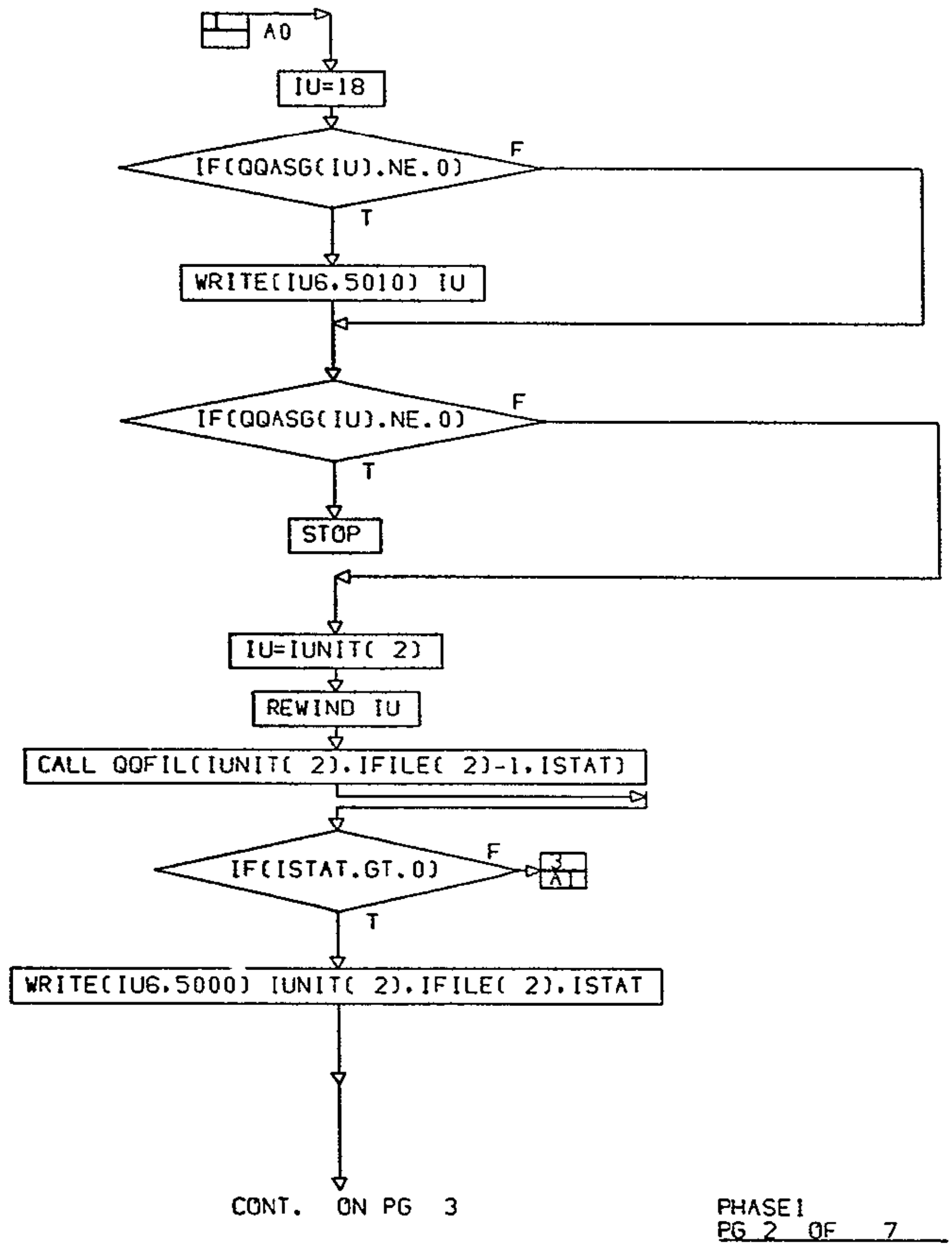
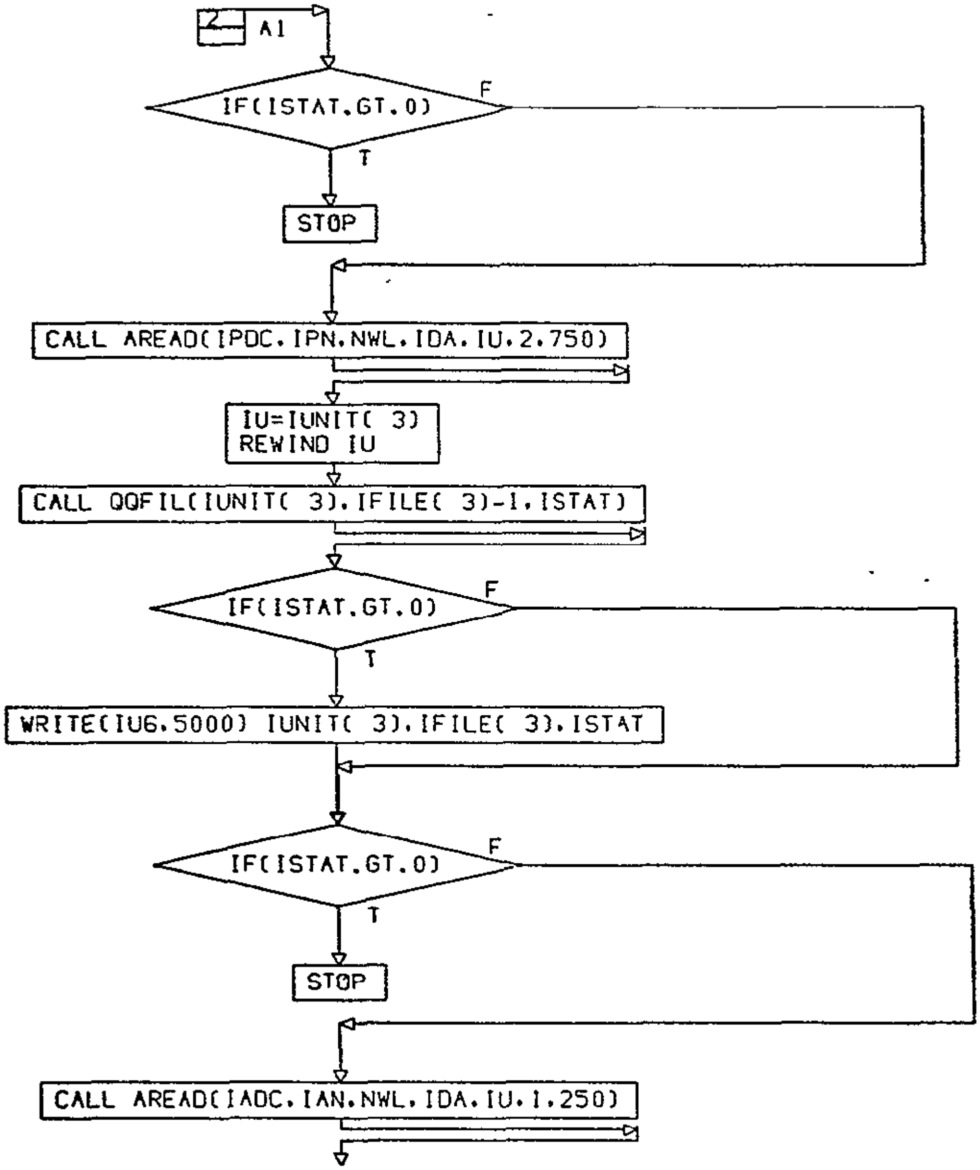


FIGURE 3.2.1.. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)

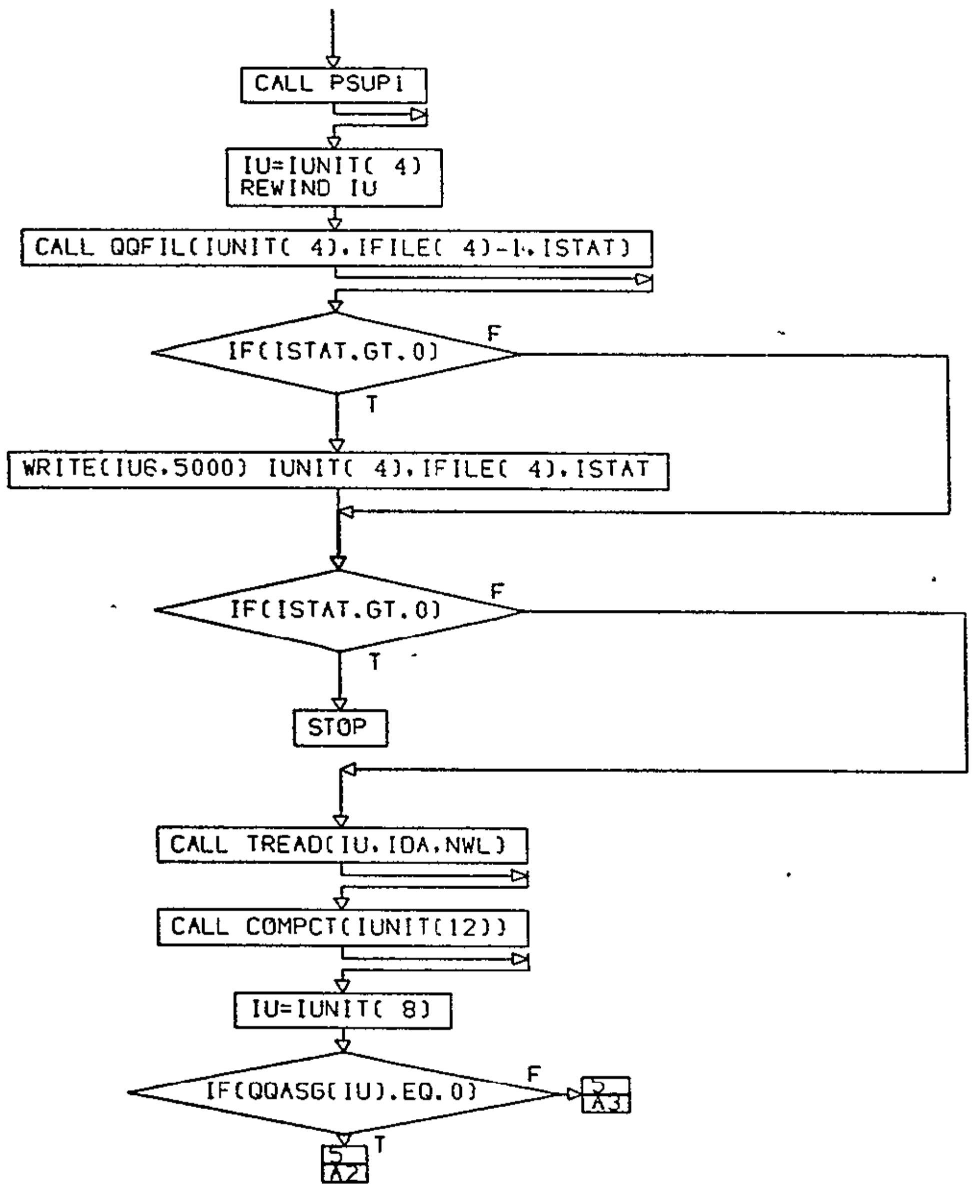
ORIGINAL PAGE IS
OF POOR QUALITY



CONT. ON PG 4

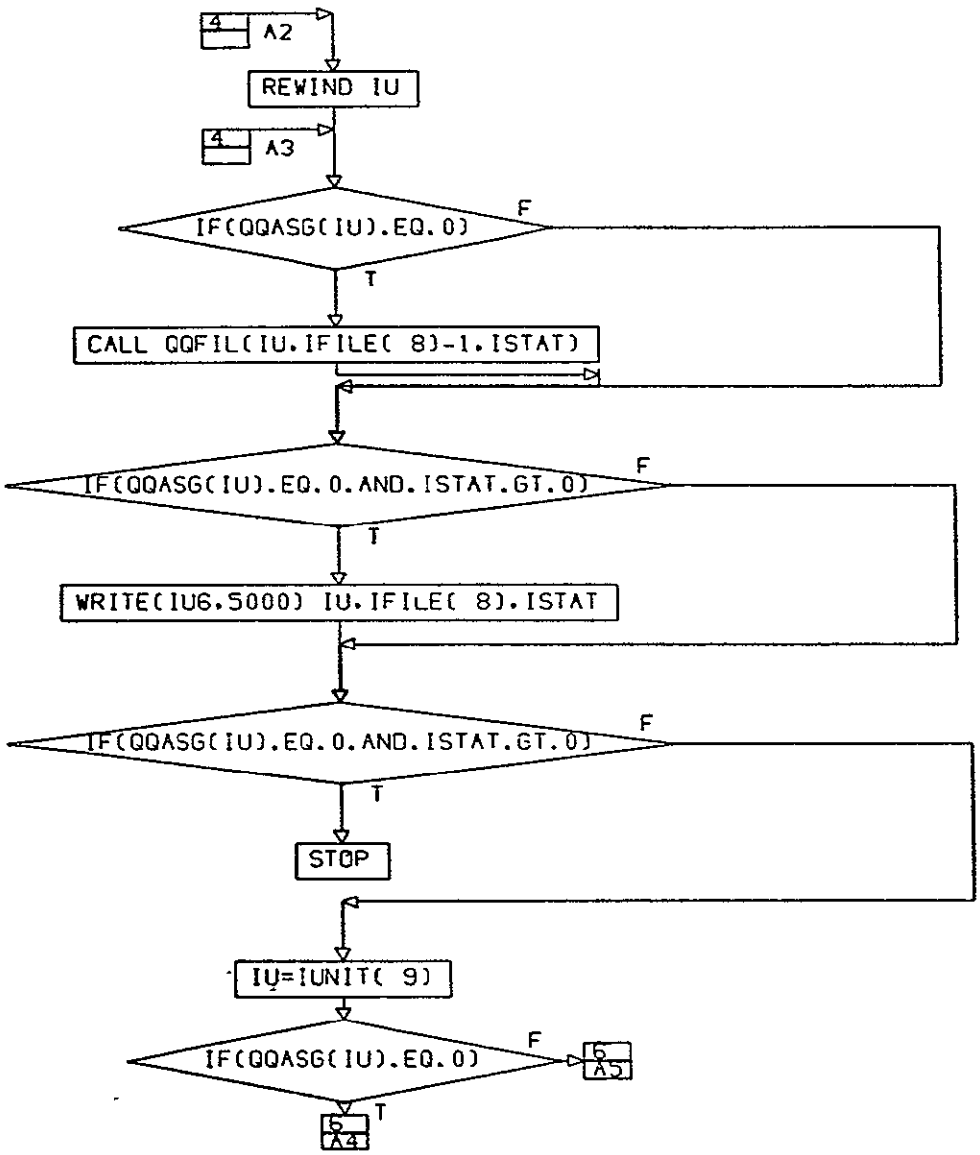
PHASE1
PG 3 OF 7

FIGURE 3.2.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)



CONT. ON PG 5

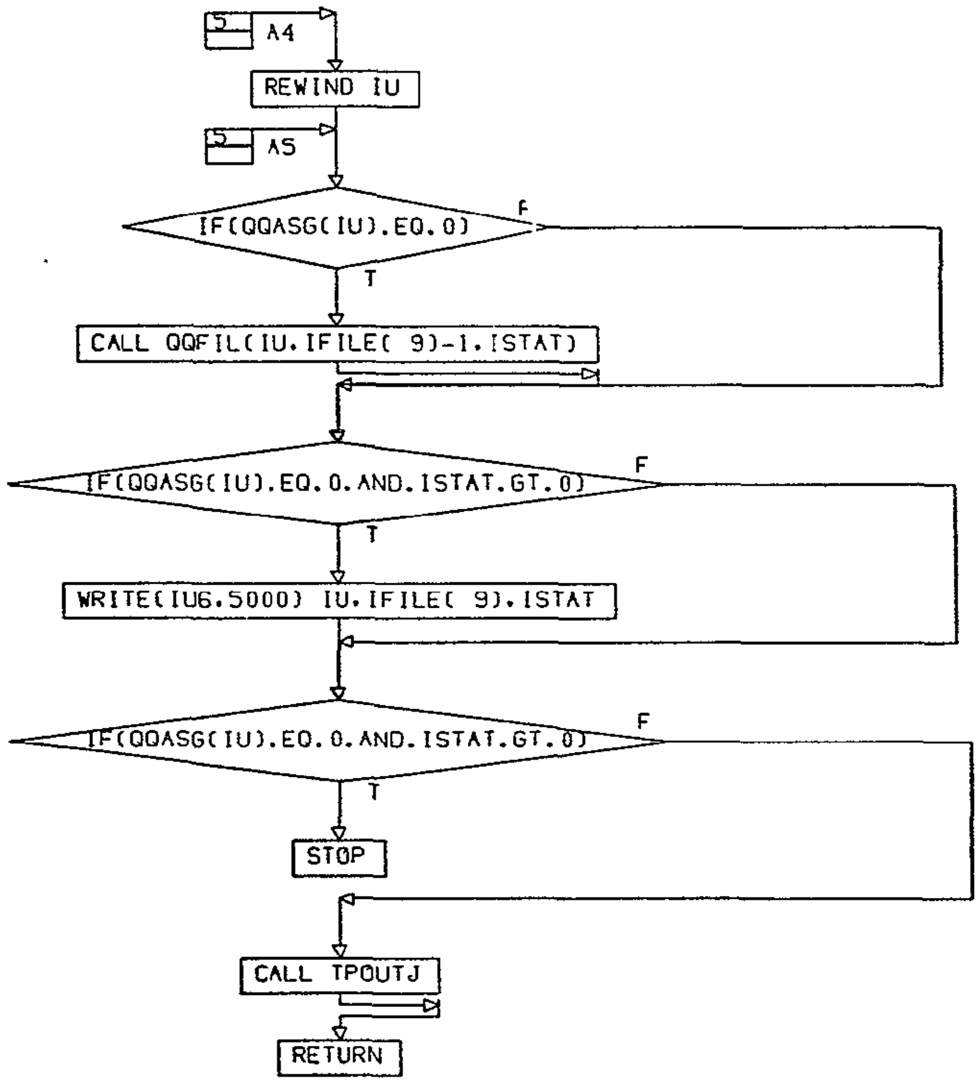
FIGURE 3.2.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)



CONT. ON PG 6

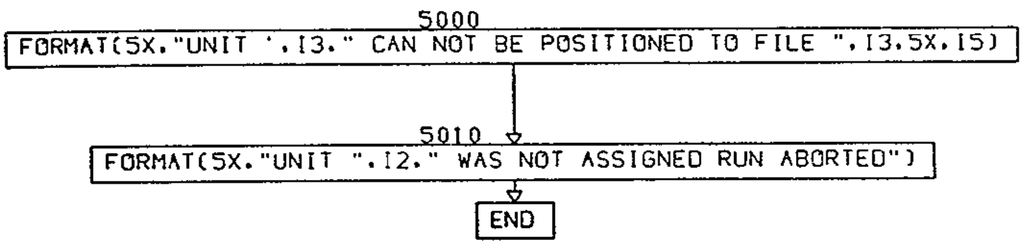
PHASEI
PG 5 OF 7

FIGURE 3.2.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)



CONT. ON PG 7

FIGURE 3.2.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)



PHASE I
PG 7 FINAL

FIGURE 3.2.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE I (CONTINUED)

3.2.2 Subroutine: ACYCLE

PURPOSE: This routine converts an Activity into Procedures and Components.

METHOD: This routine interrogates the Activity dictionary and calls the appropriate subroutines to correctly handle Procedures and Components. If the Activity cannot be found, the following diagnostic is generated.

REQUESTED ACTIVITY NNNNNNNNNN IS NOT IN THE DICTIONARY

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.2. See Appendix for definition of all variables.

NOTE: Subroutine ACYCLE is essentially identical to Subroutine AHANDL. The requirement for these subroutines is dictated by the program logic.

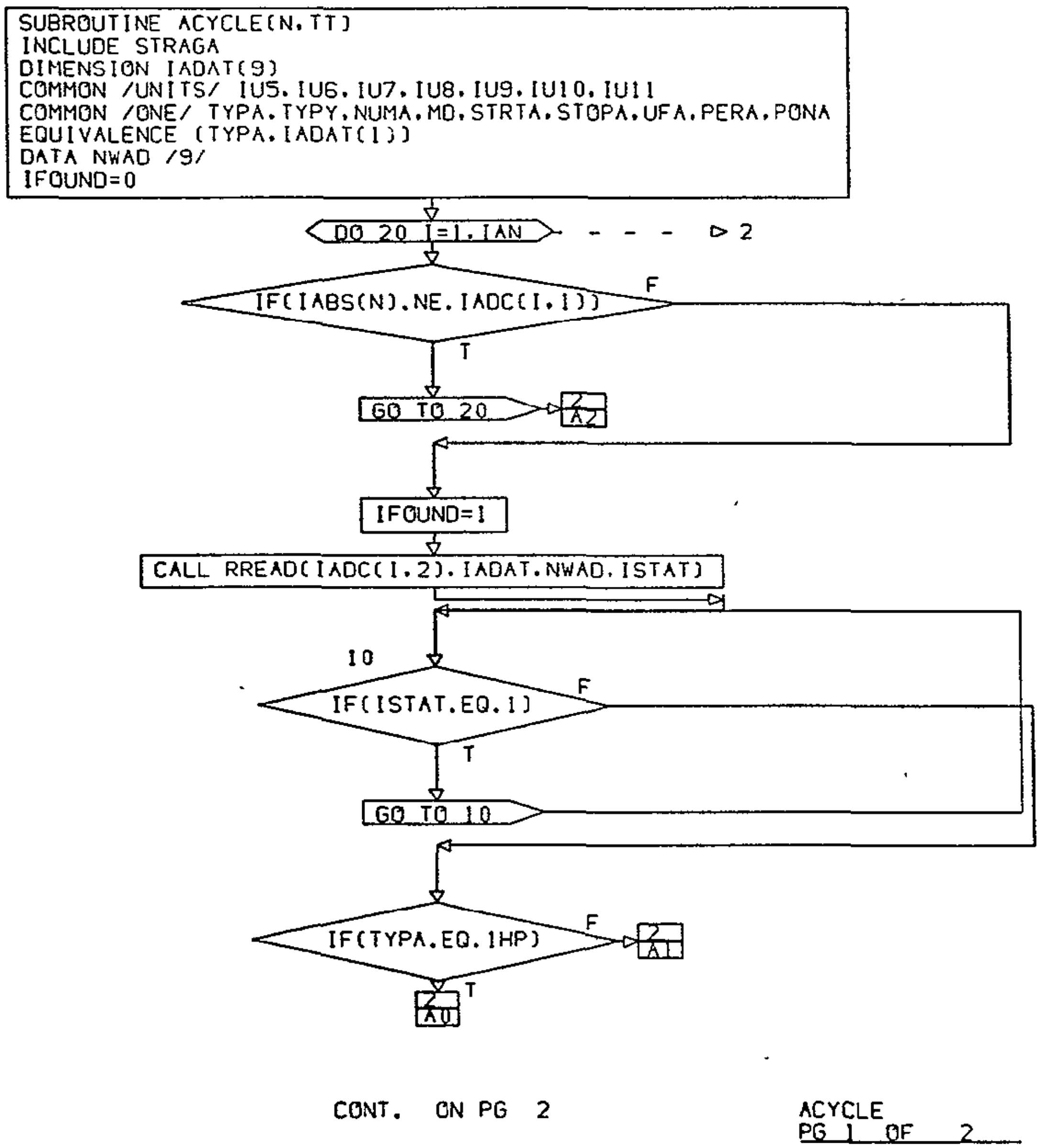


FIGURE 3.2.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACYCLE

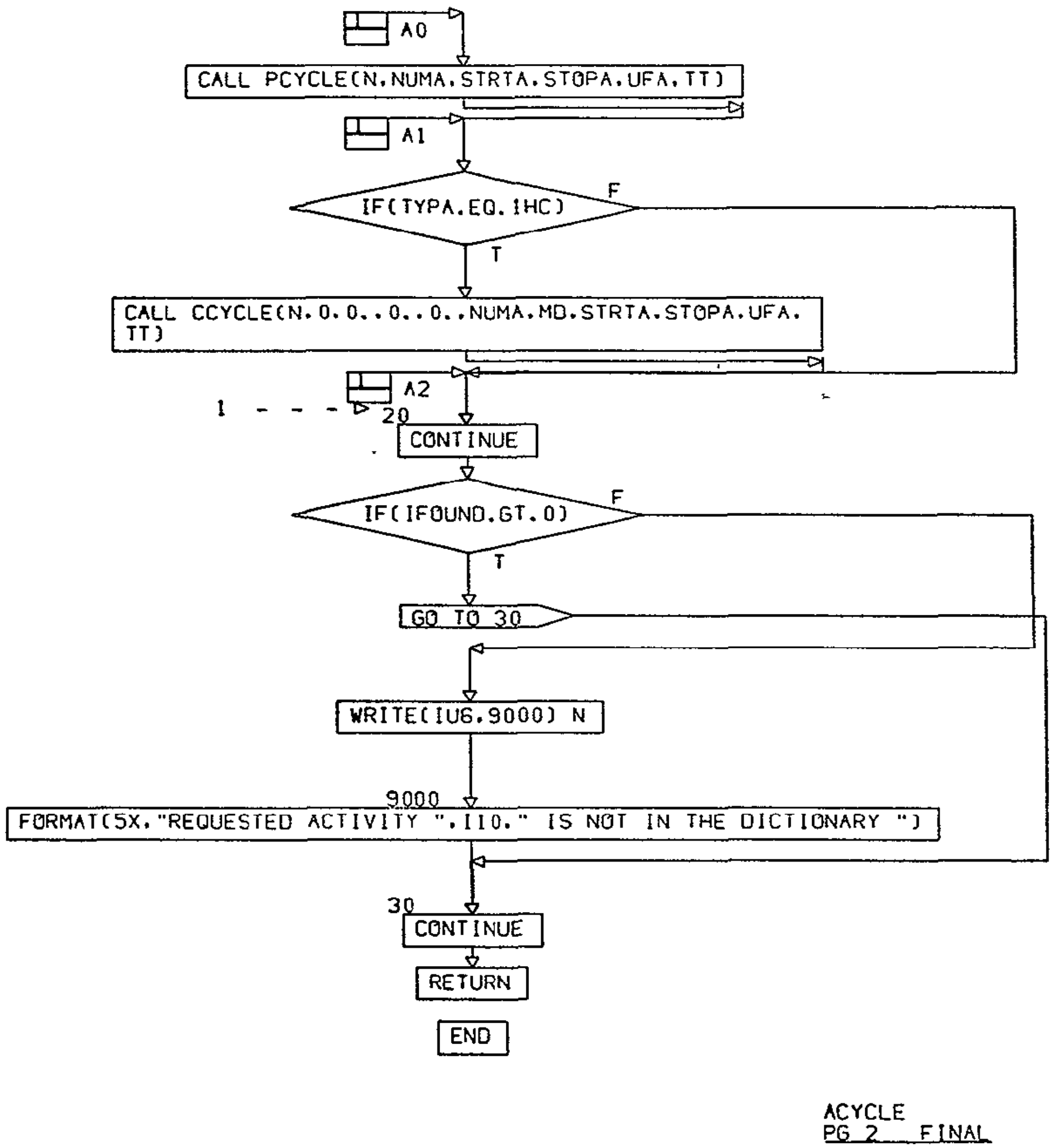


FIGURE 3.2.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACYCLE (CONTINUED)

3.2.3 Subroutine: AHANDL

PURPOSE: This routine converts an Activity into Procedures, Components and Switches.

METHOD: This routine interrogates the Activity dictionary and calls the appropriate subroutines to correctly handle Procedures, Components, and Switches. If the Activity cannot be found, the following diagnostic is generated.

REQUESTED ACTIVITY NNNNNNNNNN IS NOT IN THE DICTIONARY

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.3. See Appendix for definition of all variables.

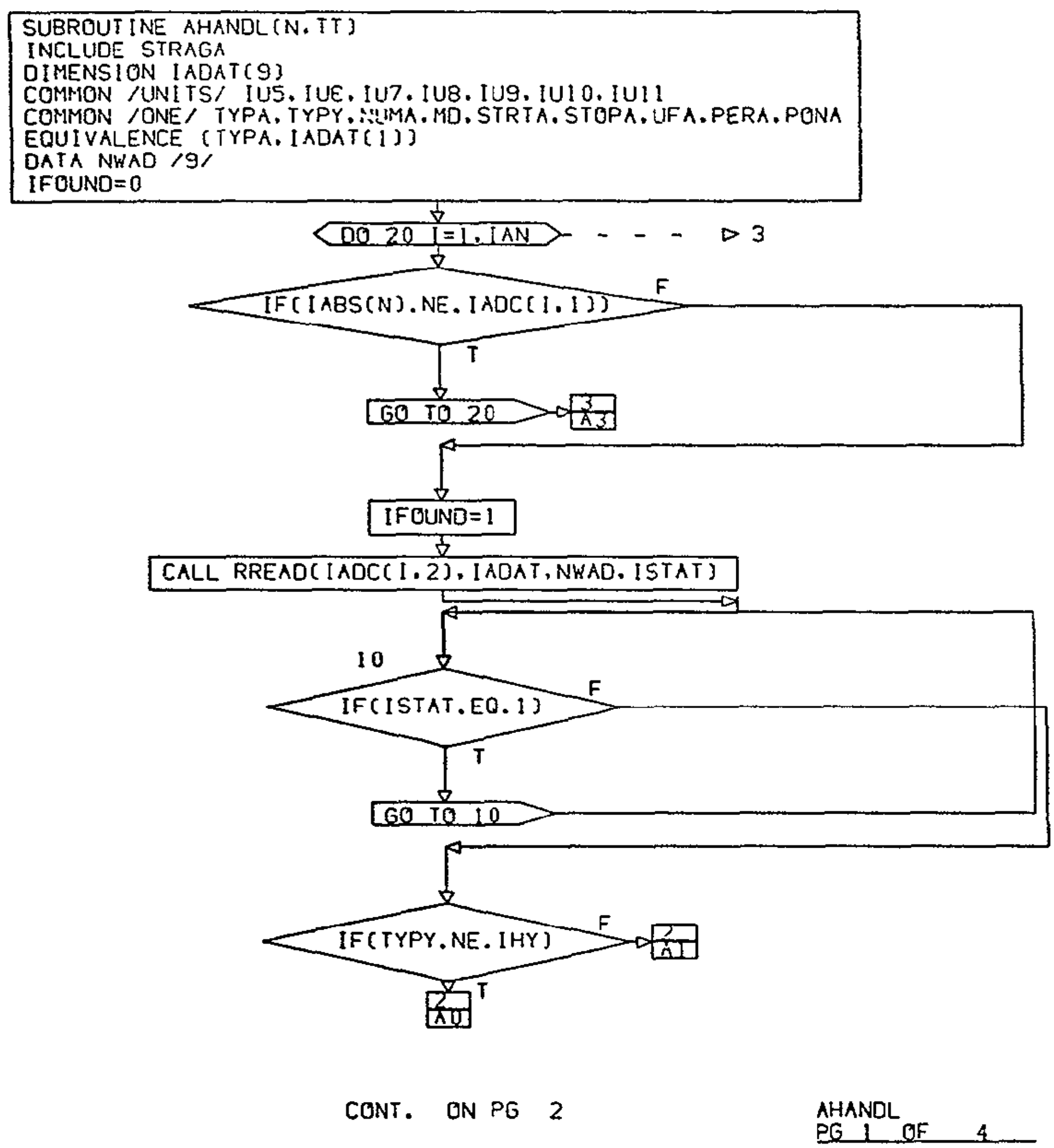
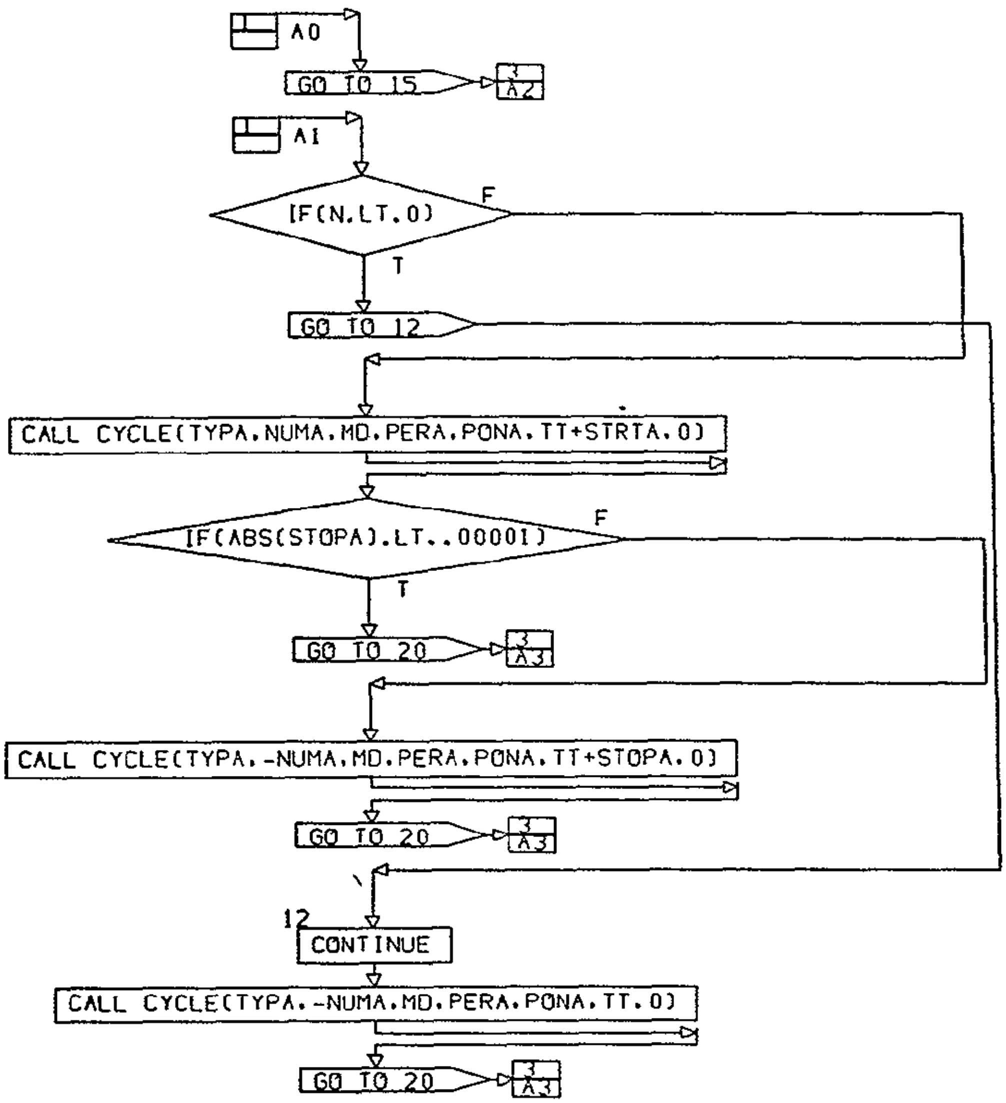


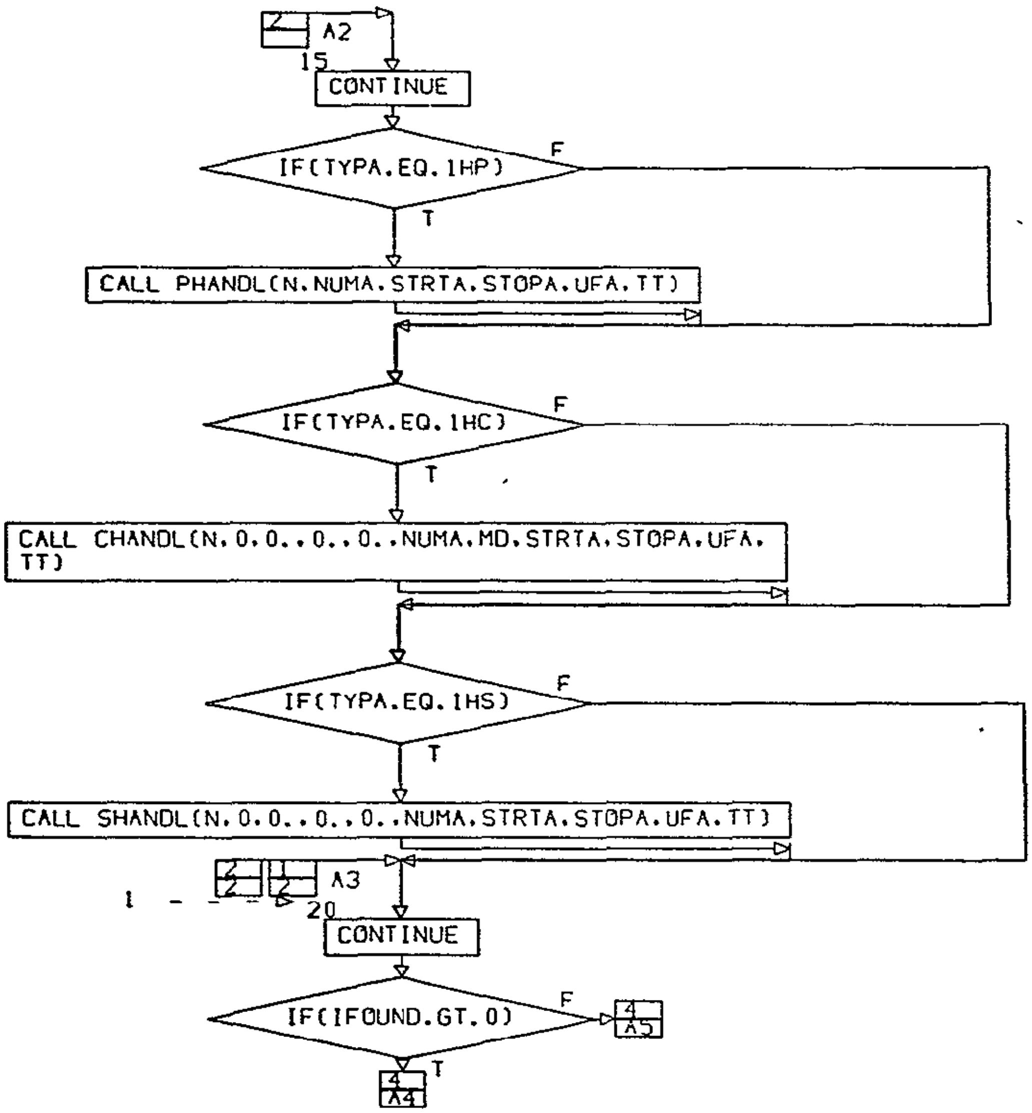
FIGURE 3.2.3. FUNCTIONAL FLOWCHART OF SUBROUTINE AHANDL



CONT. ON PG 3

AHANDL
PG 2 OF 4

FIGURE 3.2.3. FUNCTIONAL FLOWCHART OF SUBROUTINE AHANDL (CONTINUED)



CONT. ON PG 4

AHANDL
PG 3 OF 4

FIGURE 3.2.3. FUNCTIONAL FLOWCHART OF SUBROUTINE AHANDL (CONTINUED)

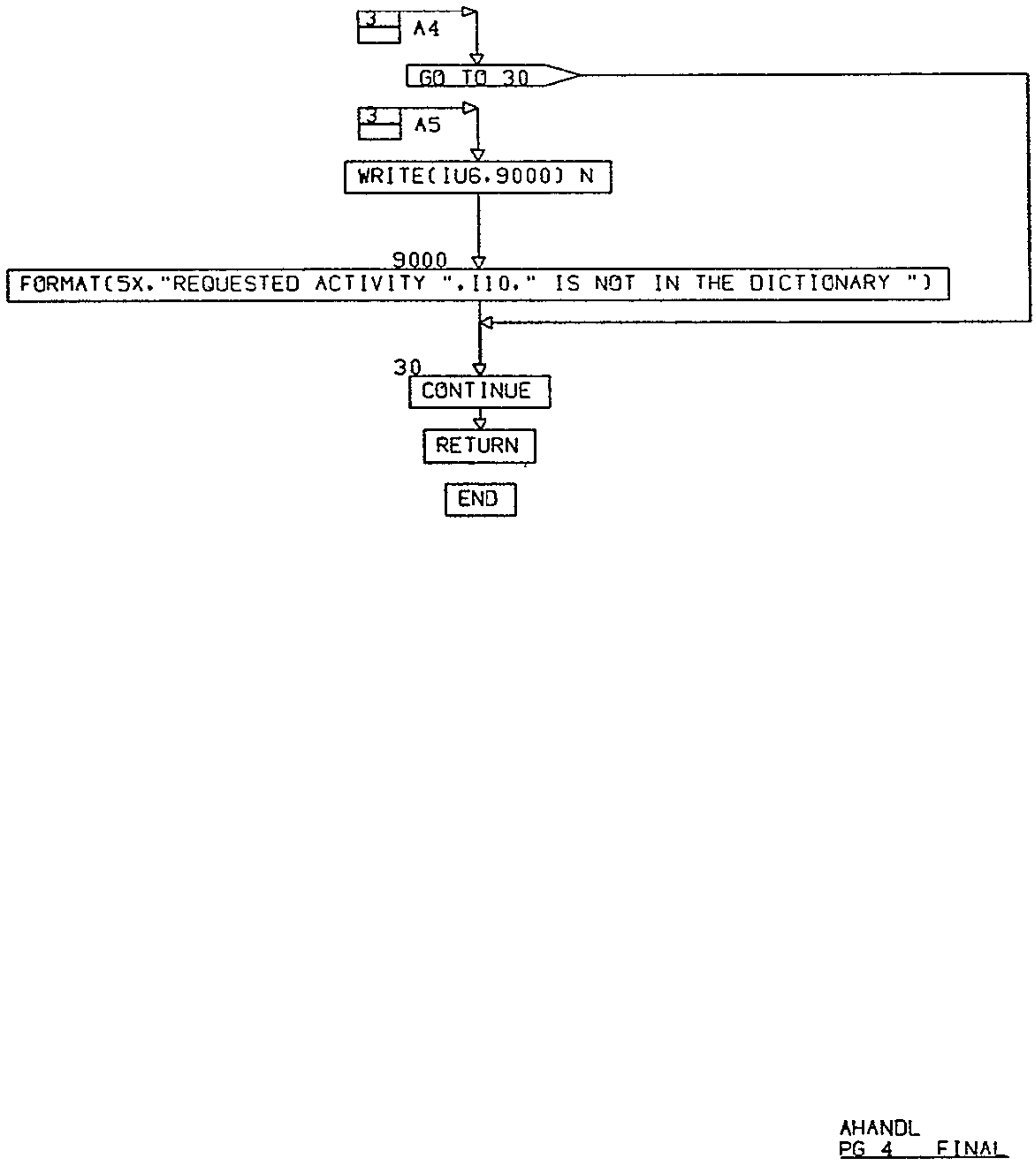


FIGURE 3.2.3. FUNCTIONAL FLOWCHART OF SUBROUTINE AHANDL (CONTINUED)

3.2.4 Subroutine: AREAD

PURPOSE: This routine reads the Activity/Procedure definition cards and creates the Activity/Procedure dictionary.

METHOD: This routine is called twice; once to create the Activity dictionary and once to create the Procedure dictionary.

Each card as it is read in is tested to determine if it is a File Title card, a Card Type 1 (an Activity/Procedure title card), or a Card Type 2 (an Activity/Procedure definition card). As each Card Type 2 is read, it is related to its card type 1 and stored in a Sequentially/indexed file and its location in the file is stored in the dictionary.

The following diagnostics are provided:

INCORRECT TYPE AANNNNNNN NOT A {ACTIVITY / PROCEDURE}

NO MORE DRUM SPACE AVAILABLE - THE LAST CARD PROCESSED WAS AA

ID NNNNNNN NUMBER NNNNNNNNNN RUN ABORTED

READ ERROR ON CARD AANNNNNNN {ACTIVITY / PROCEDURE}

VARIABLES: Calling Arguments:

IDICT - where the dictionary is to be stored

ICD - number of entries in IDICT

NWL - words of drum remaining

IDA - drum address

IU - unit definitions' are to be read form

ITY - entry type 1 - procedure 2 - activity

IDEM - IDICT dimension

The remaining variables listed in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.4. See Appendix for definition of all variables.

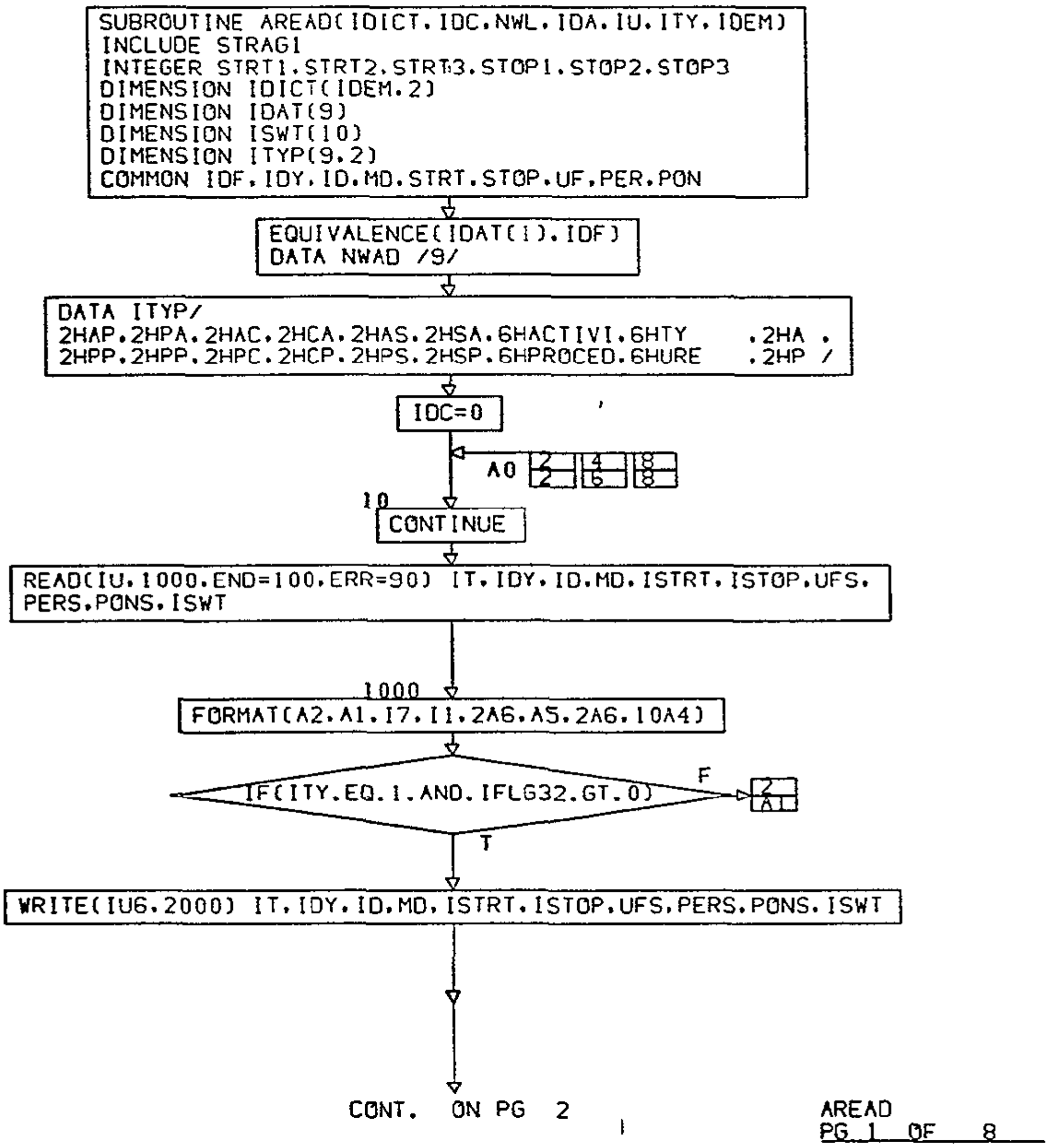


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD

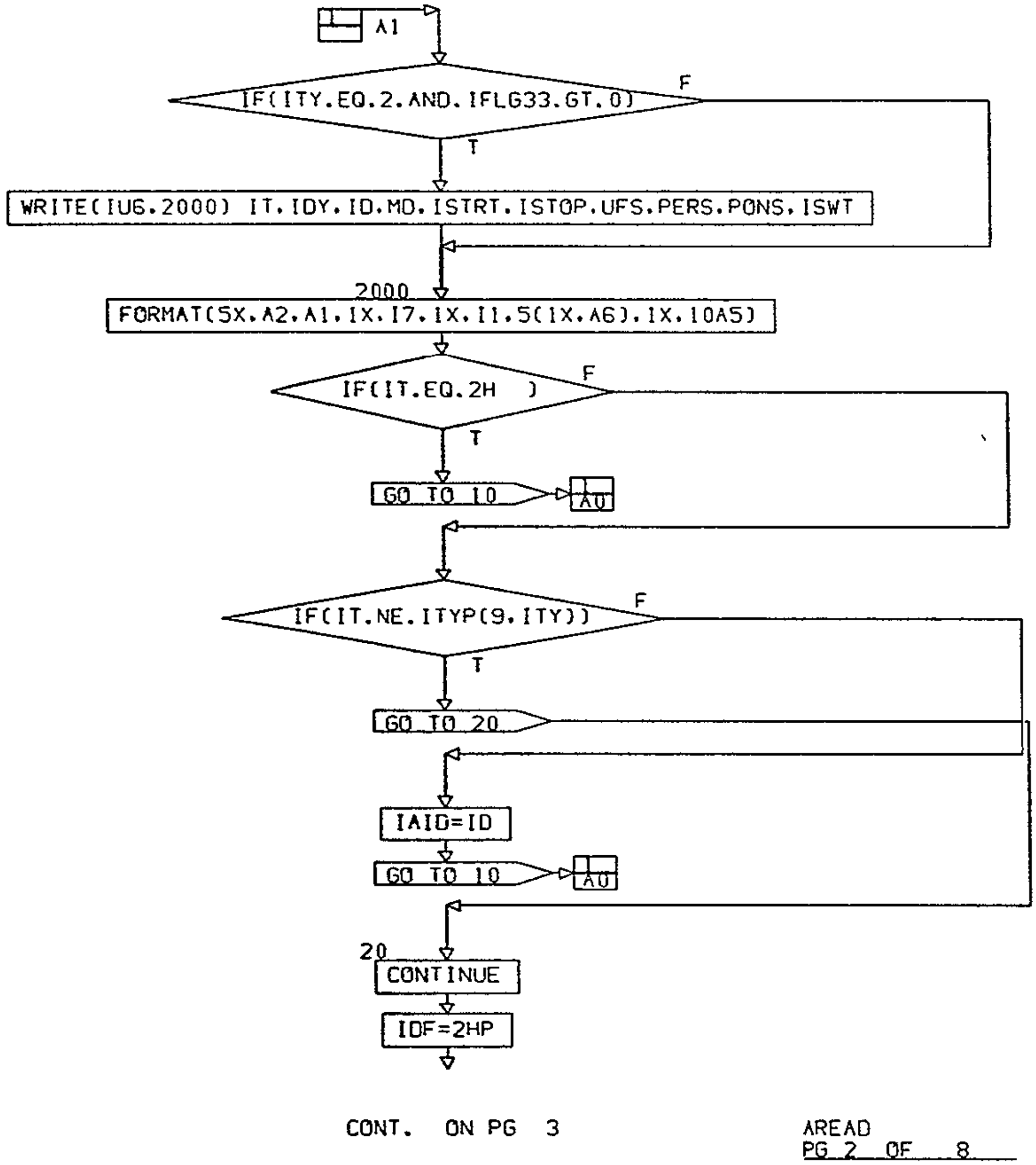
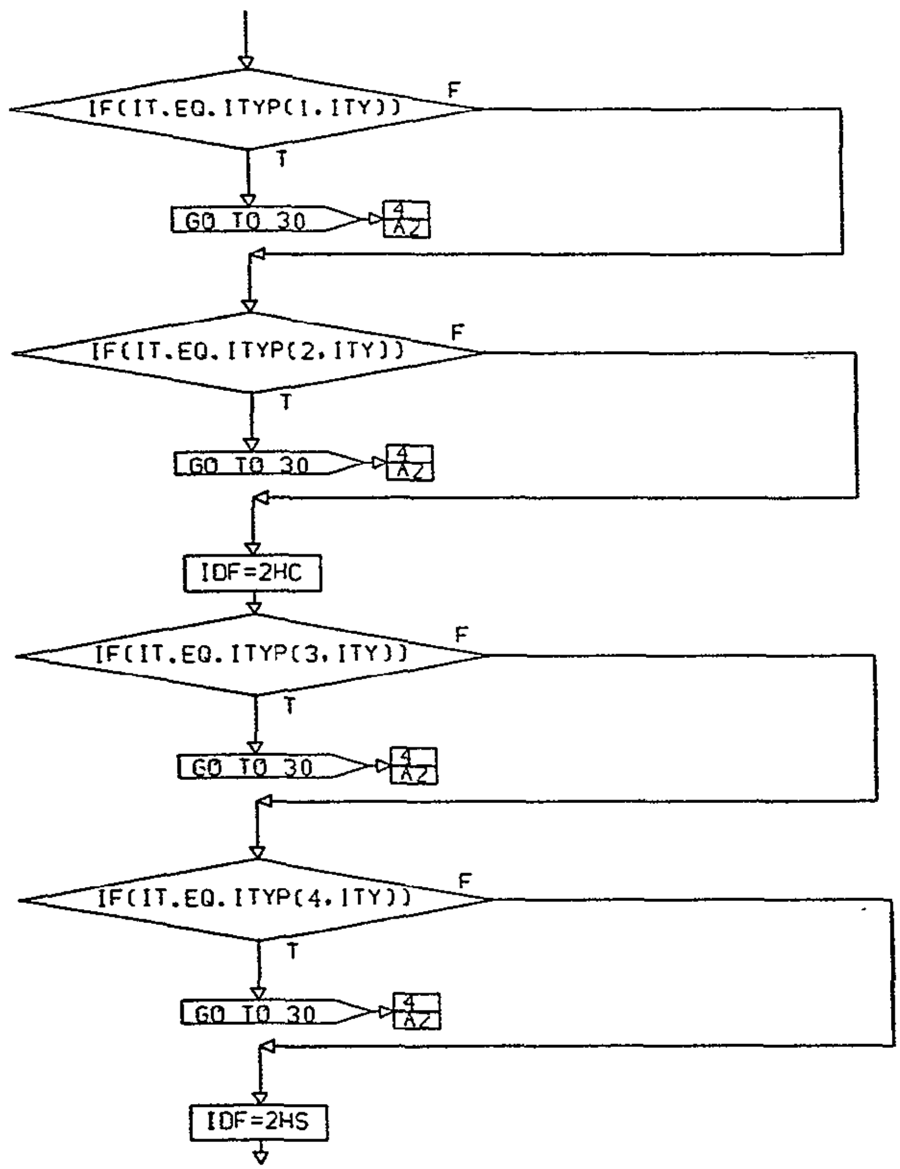


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)



CONT. ON PG 4

AREAD
PG 3 OF 8

FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)

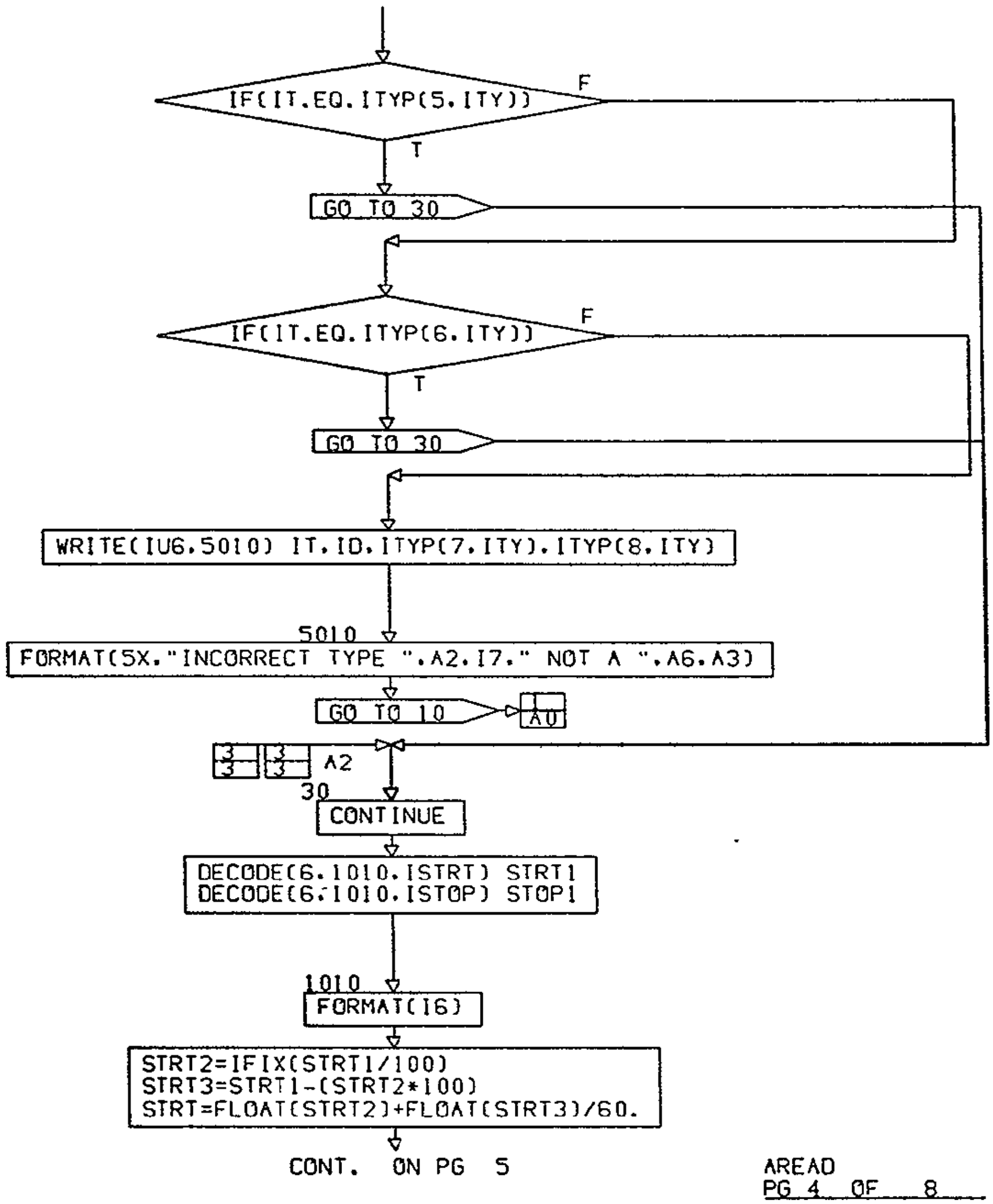


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)

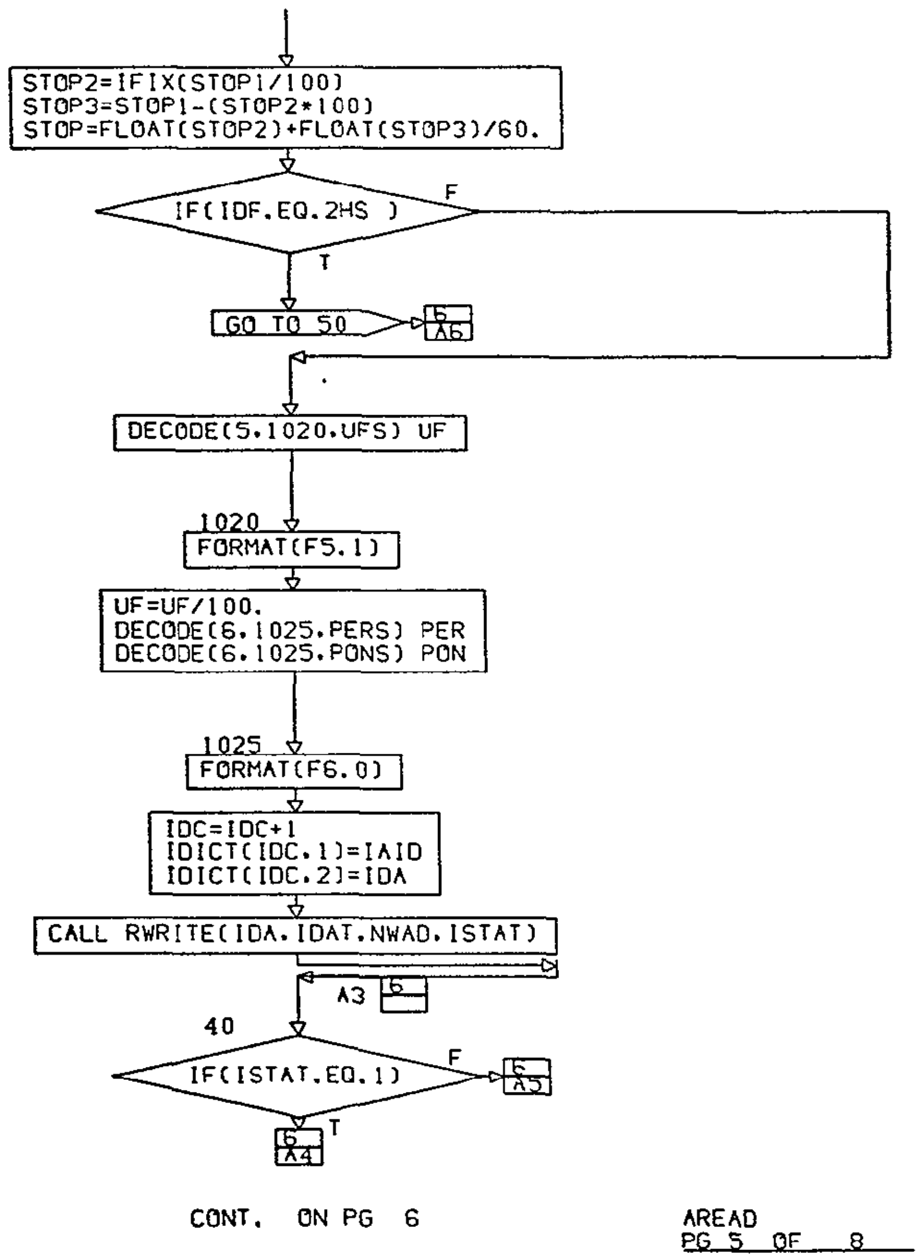


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)

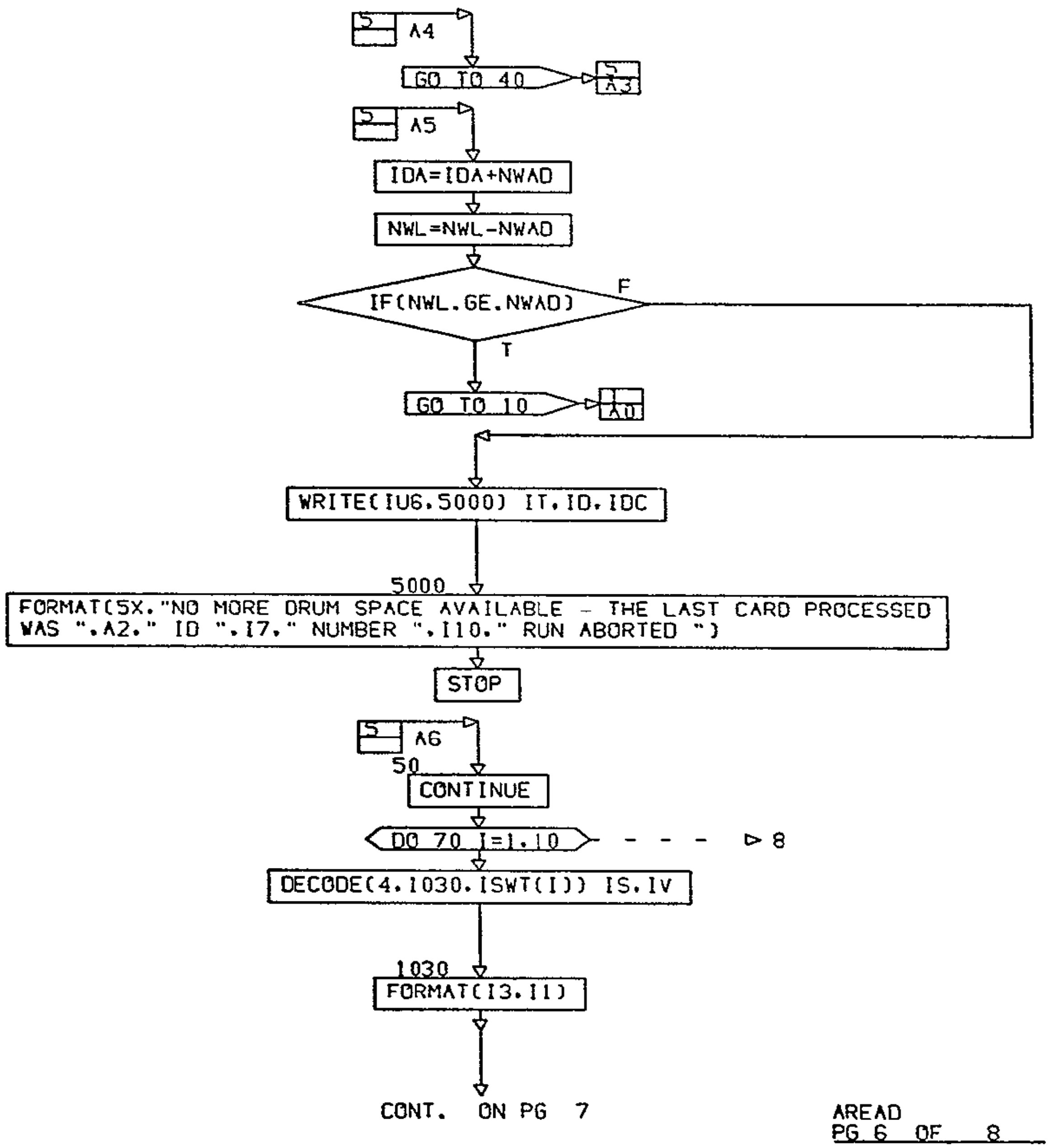
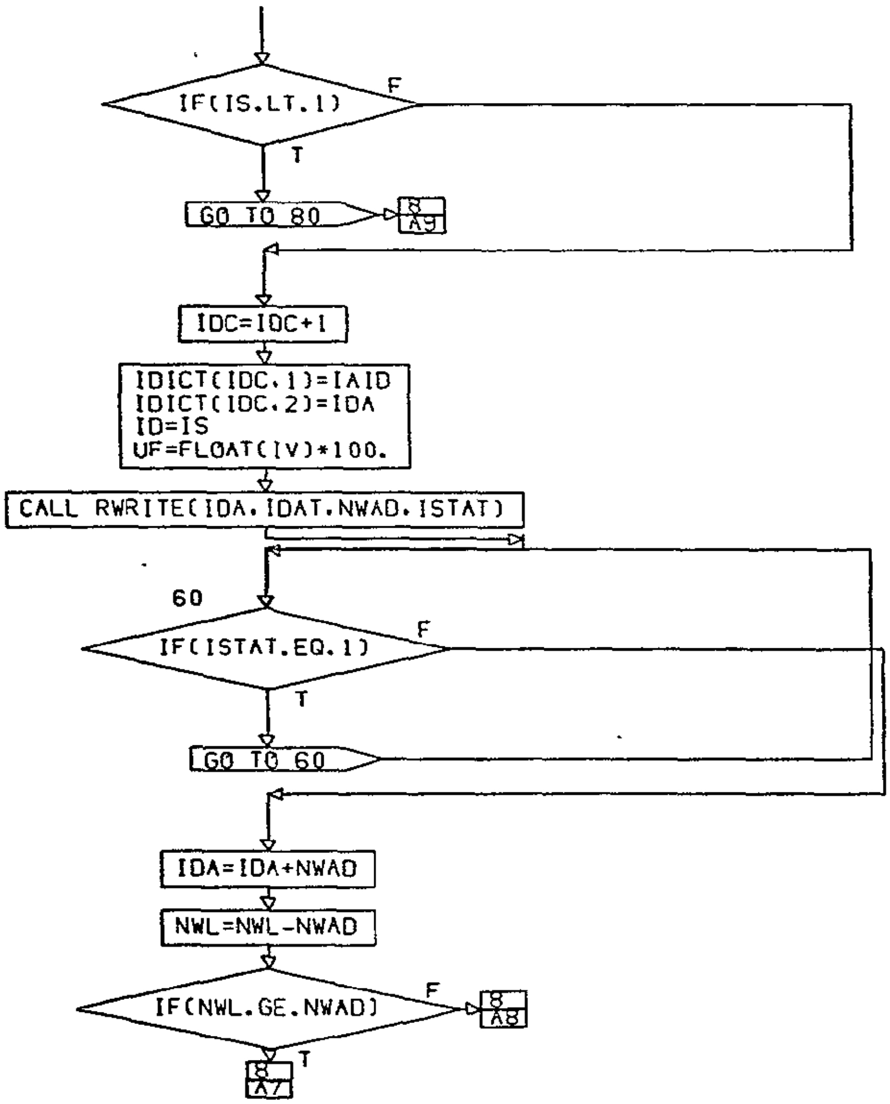


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)



CONT. ON PG 8

FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)

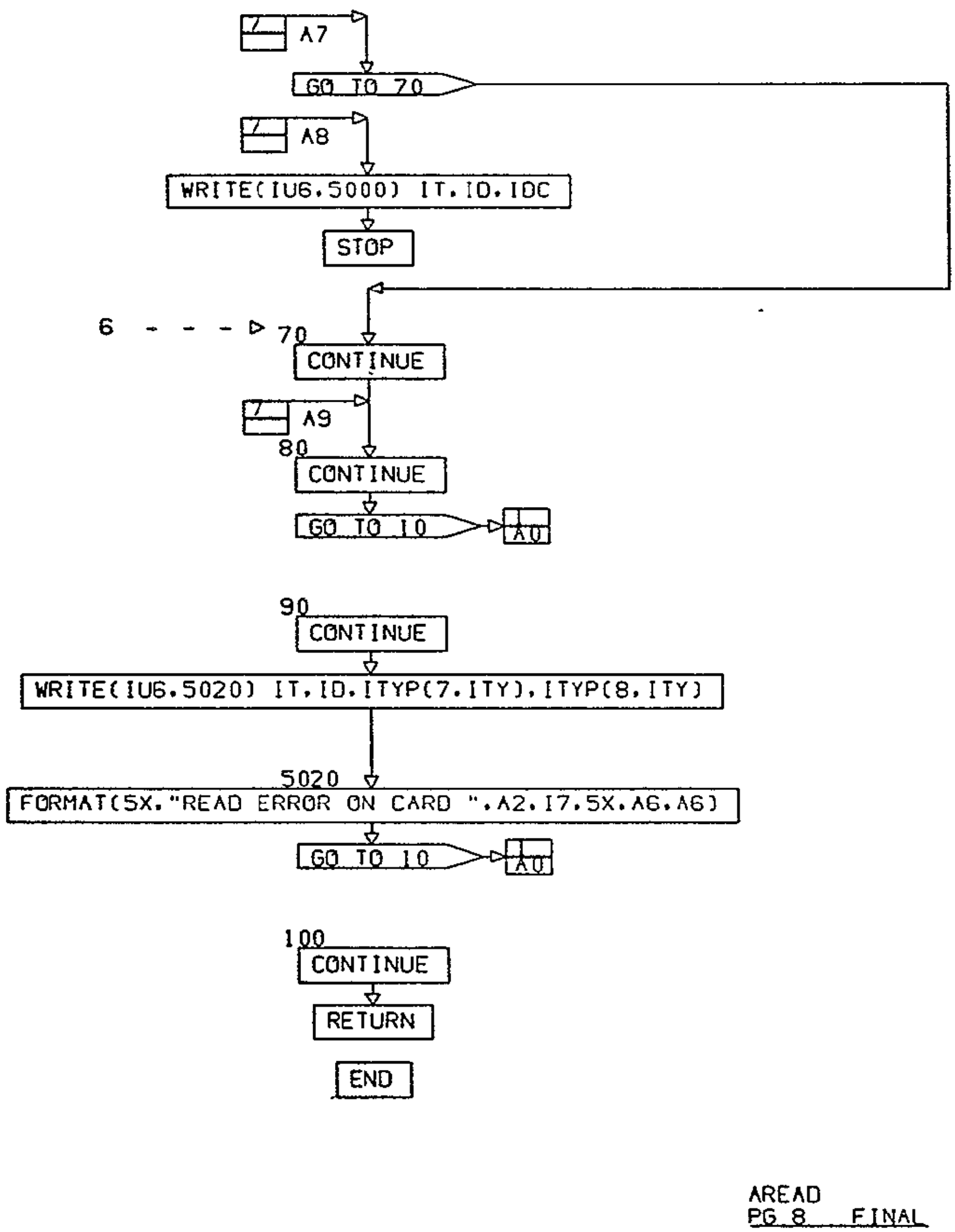


FIGURE 3.2.4. FUNCTIONAL FLOWCHART OF SUBROUTINE AREAD (CONTINUED)

3.2.5 Subroutine: CCYCLE

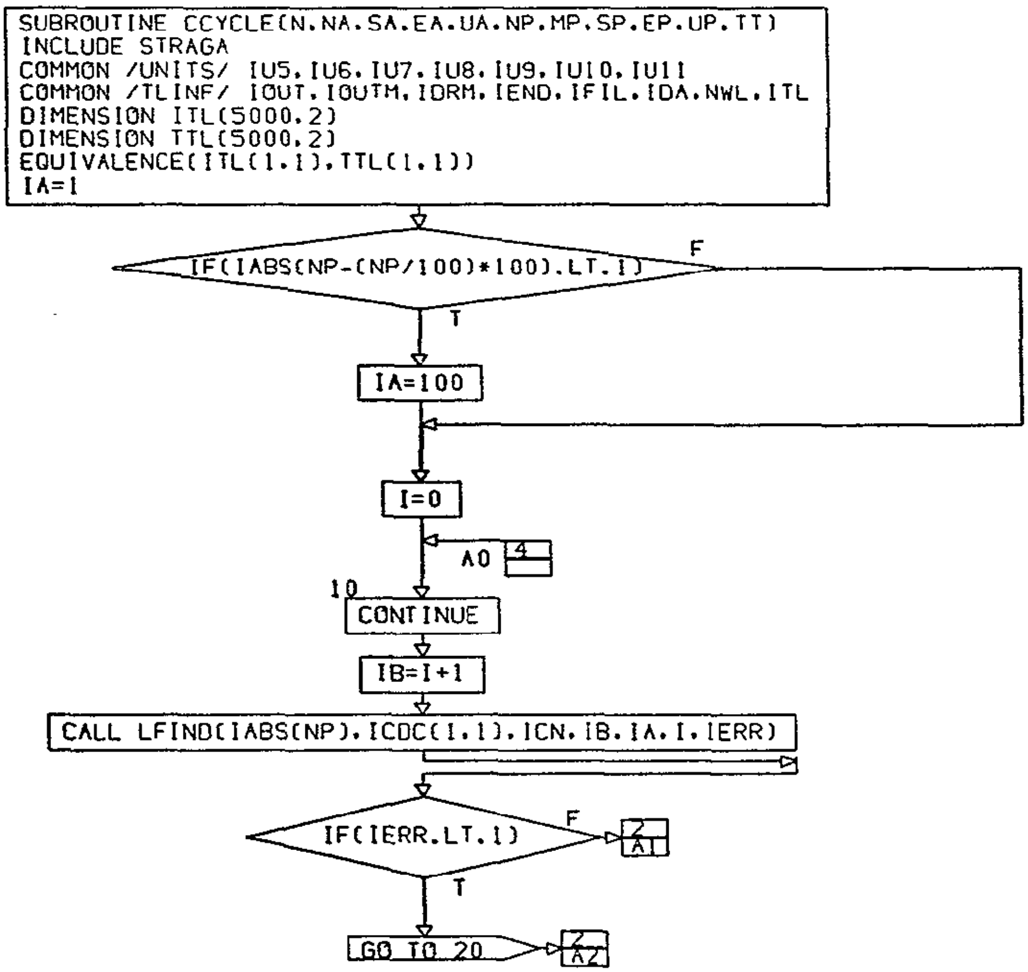
PURPOSE: This routine creates the component portion of the event timeline.

METHOD: For each component the following are determined:

1. Location in the component dictionary
2. Shows the component as "active" and to be included in the compacted dictionary
3. Determines the event on and off time
4. Stores the event in the timeline array
5. Writes the event on drum

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.5. See Appendix for definition of all variables.

NOTE: Subroutine CCYCLE is essentially identical to Subroutine CHANDL. The requirement for these subroutines is dictated by the program logic.



CONT. ON PG 2

CCYCLE
PG 1 OF 5

FIGURE 3.2.5. FUNCTIONAL FLOWCHART OF SUBROUTINE CCYCLE

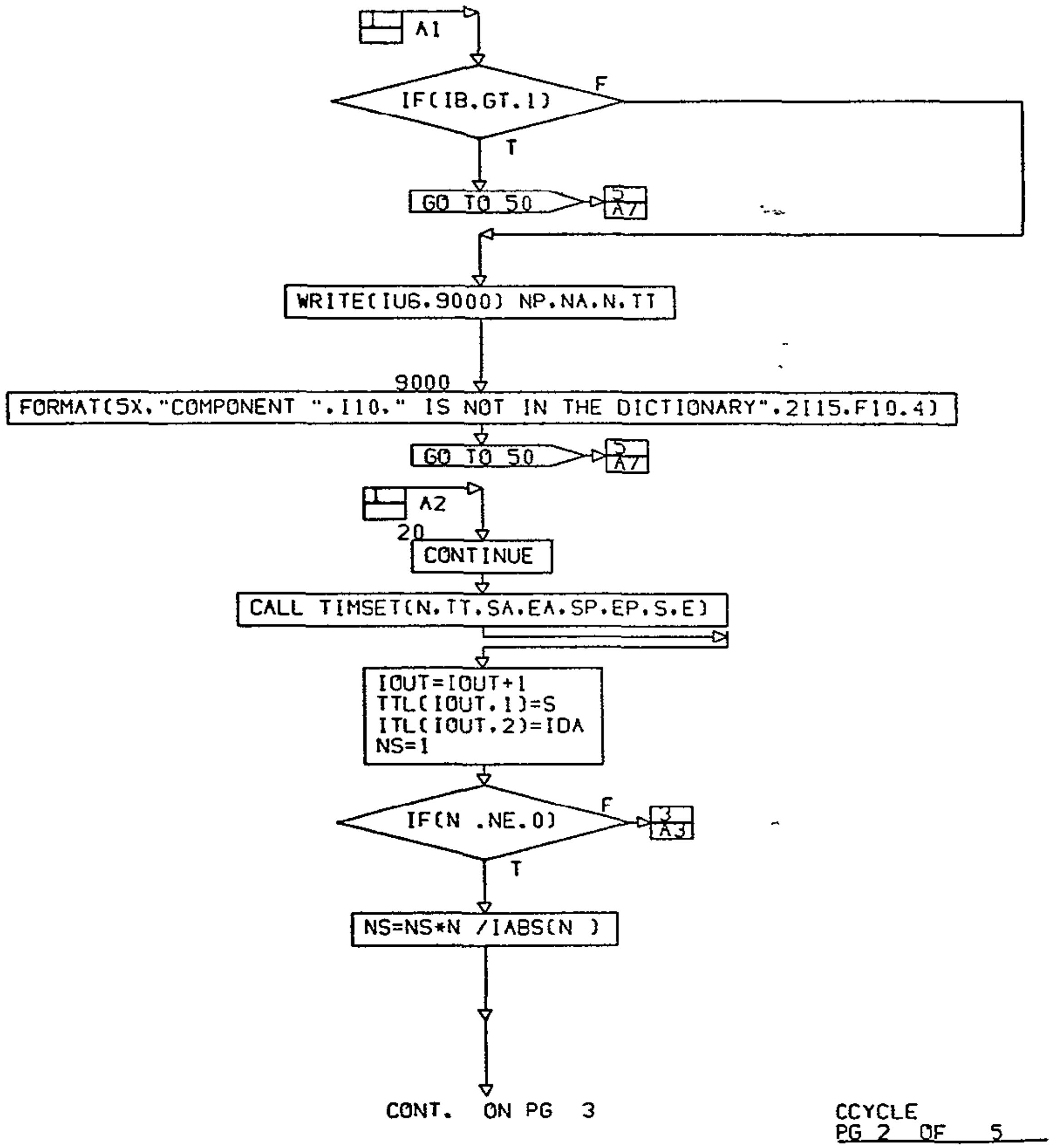
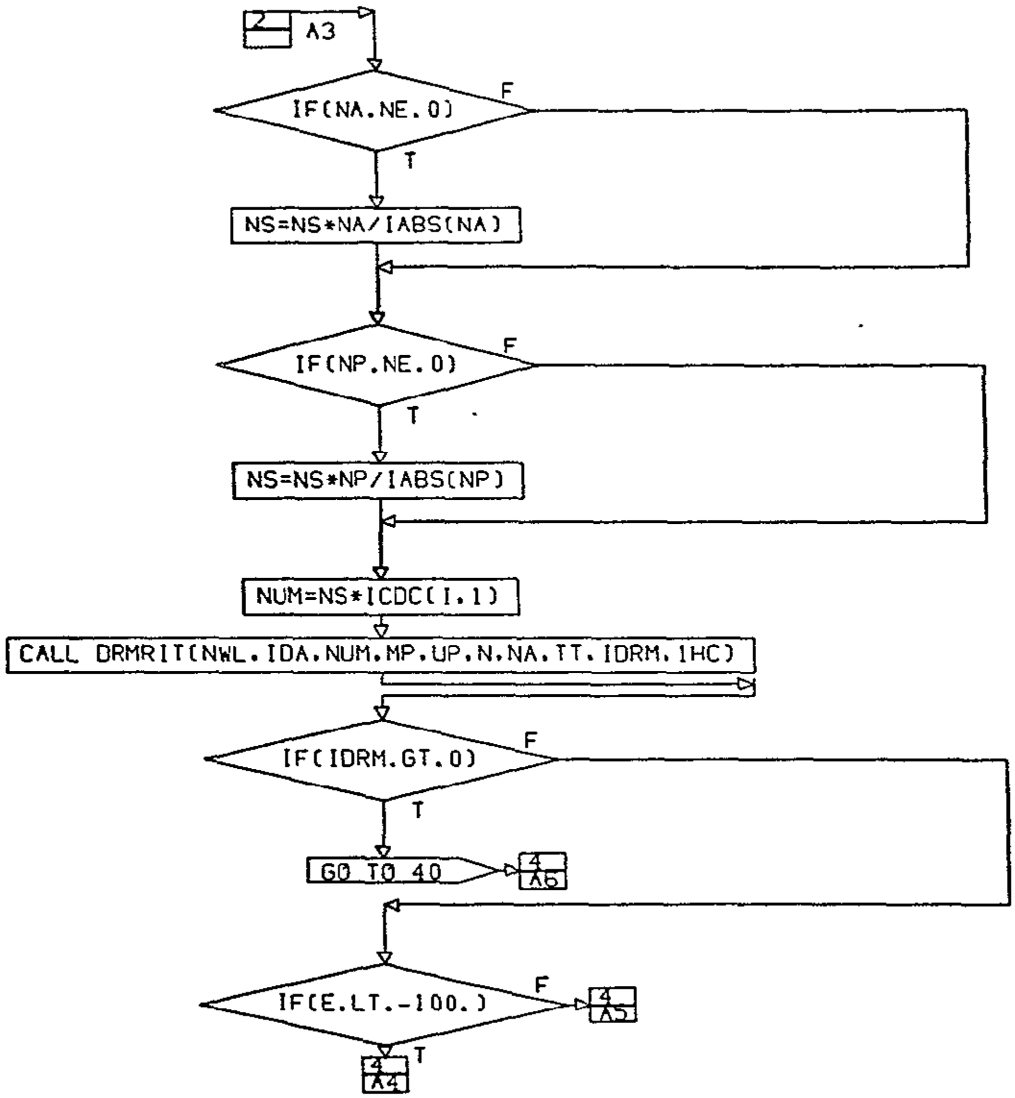


FIGURE 3.2.5. FUNCTIONAL FLOWCHART OF SUBROUTINE CCYCLE (CONTINUED)



CONT. ON PG 4

CCYCLE
PG 3 OF 5

FIGURE 3.2.5. FUNCTIONAL FLOWCHART OF SUBROUTINE CCYCLE (CONTINUED)

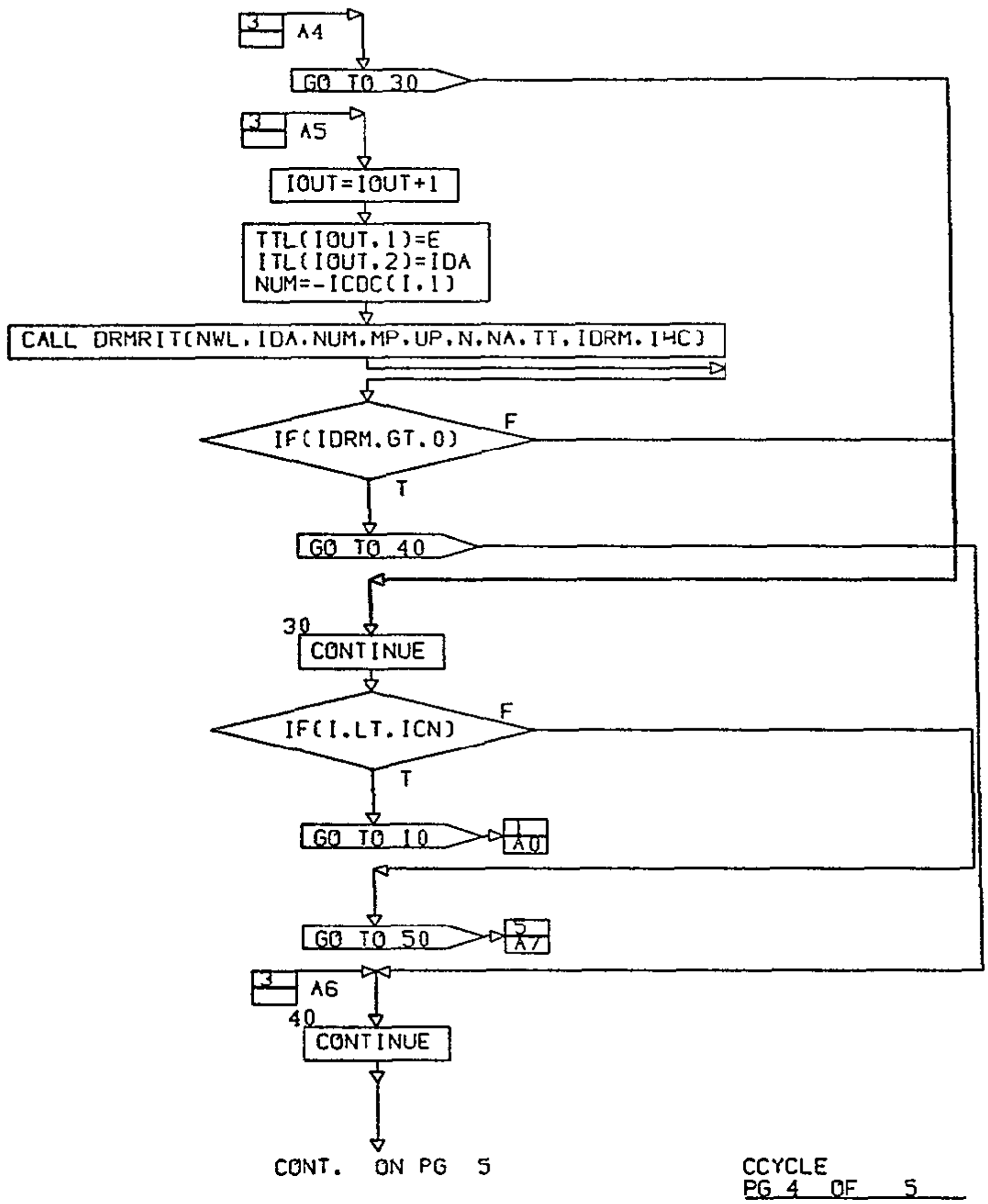
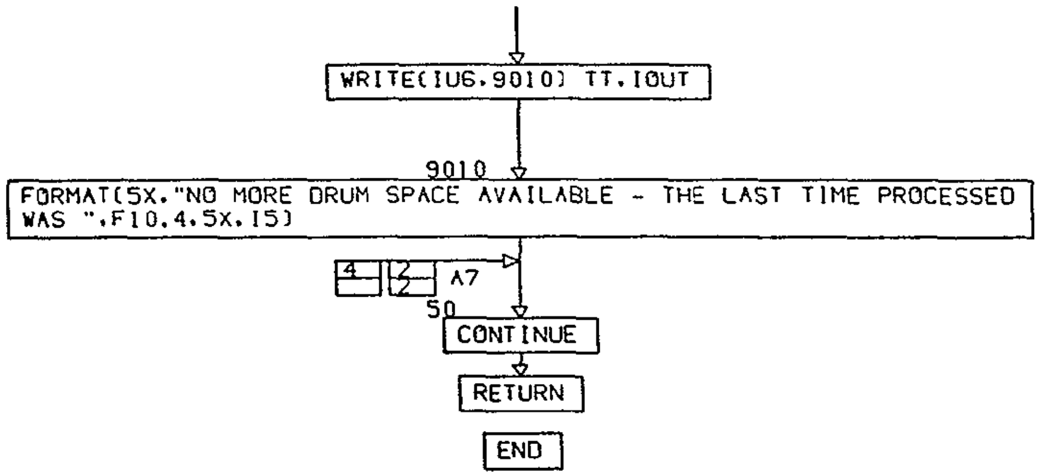


FIGURE 3.2.5. FUNCTIONAL FLOWCHART OF SUBROUTINE CCYCLE (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY



CCYCLE
PG 5 FINAL

FIGURE 3.2.5. FUNCTIONAL FLOWCHART OF SUBROUTINE CCYCLE (CONTINUED)

3.2.6 Subroutine: CHANDL

PURPOSE: This routine creates the component portion of the event timeline.

METHOD: For each component the following are determined:

1. Location in the component dictionary

2. Shows the component as "active" and to be included in the compacted dictionary

3. Determines the event on and off time

4. Stores the event in the timeline array

5. Writes the event on drum

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.6. See Appendix for definition of all variables.

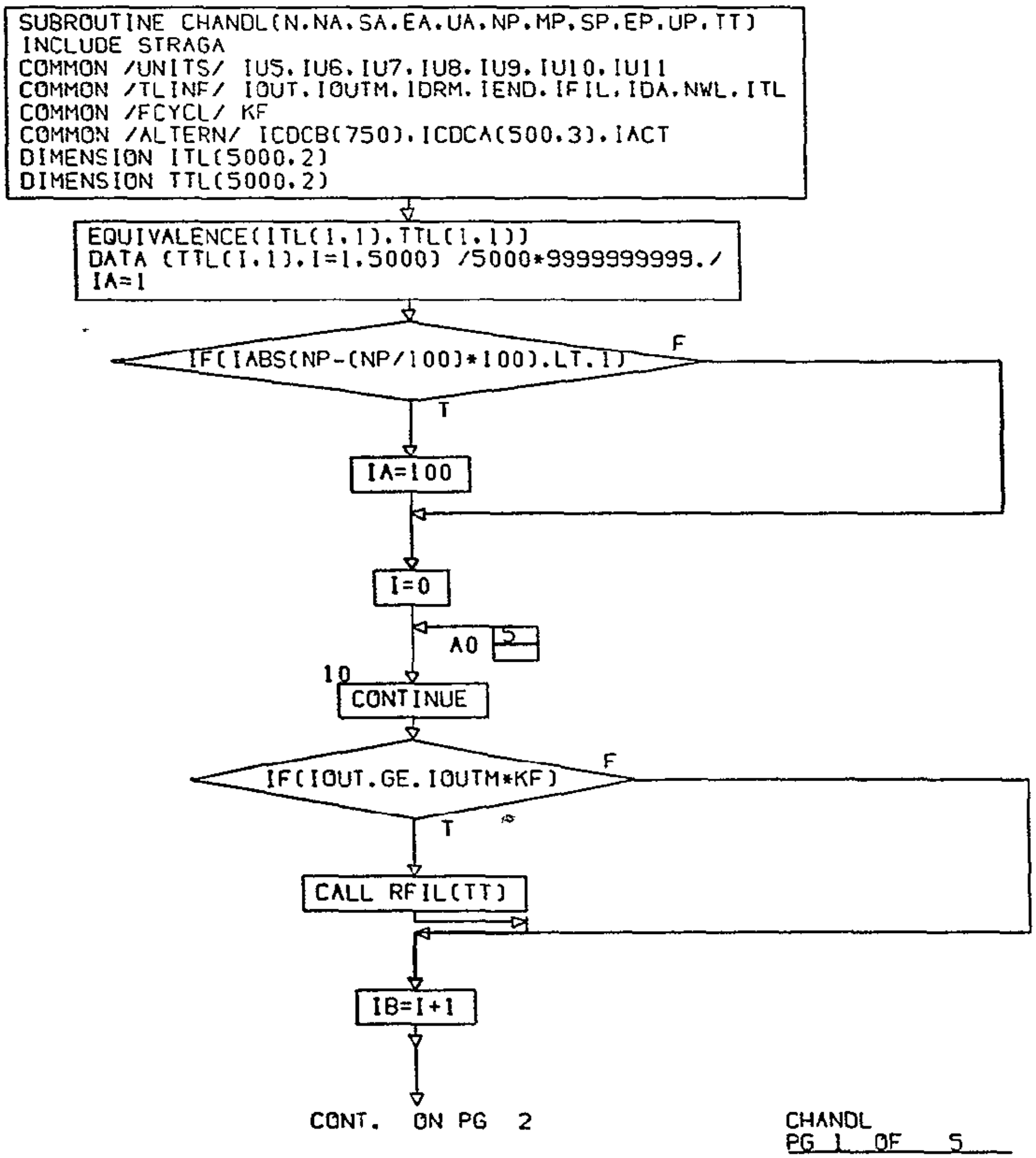


FIGURE 3.2.6. FUNCTIONAL FLOWCHART OF SUBROUTINE CHANDL

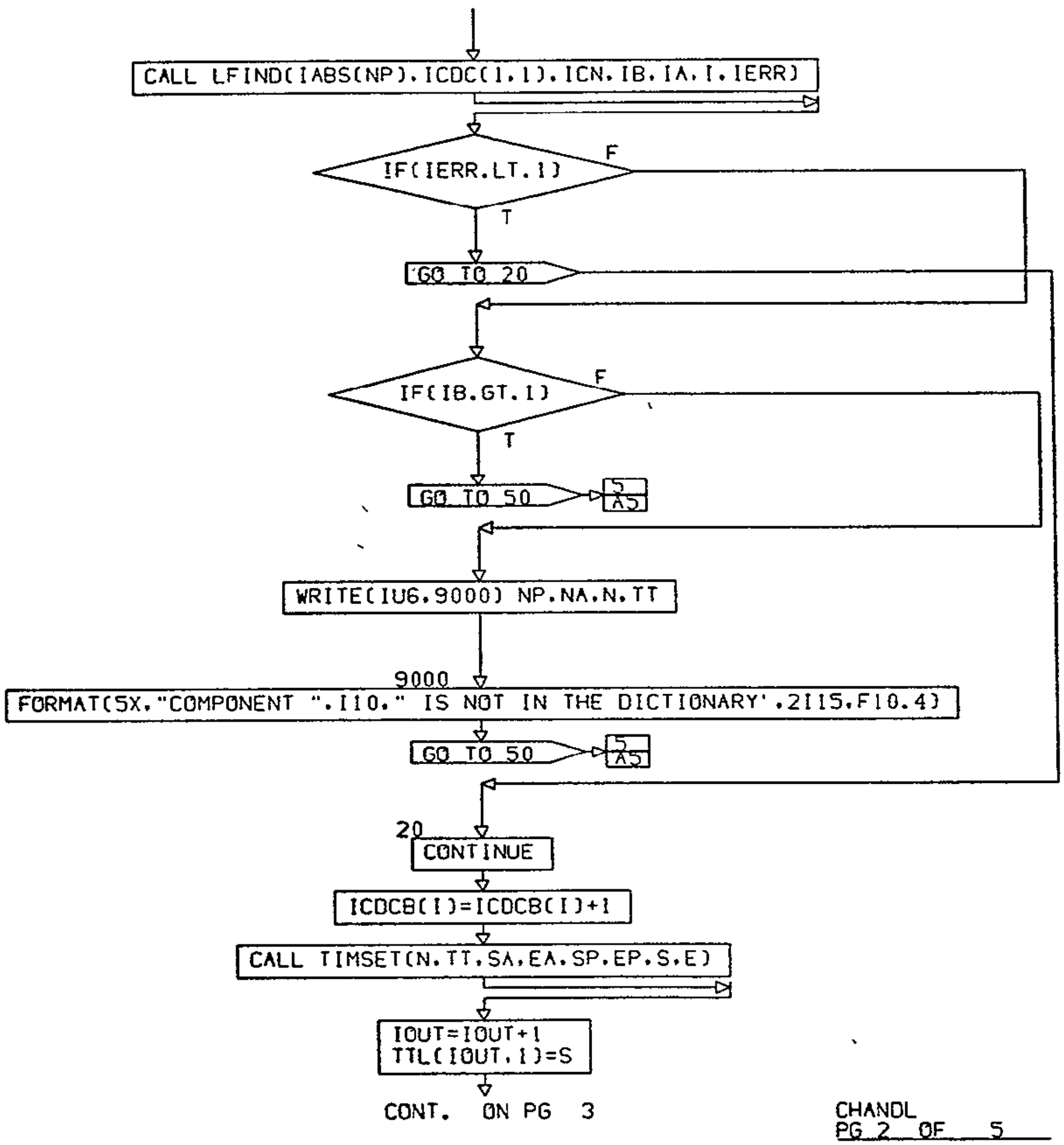


FIGURE 3.2.6. FUNCTIONAL FLOWCHART OF SUBROUTINE CHANDL (CONTINUED)

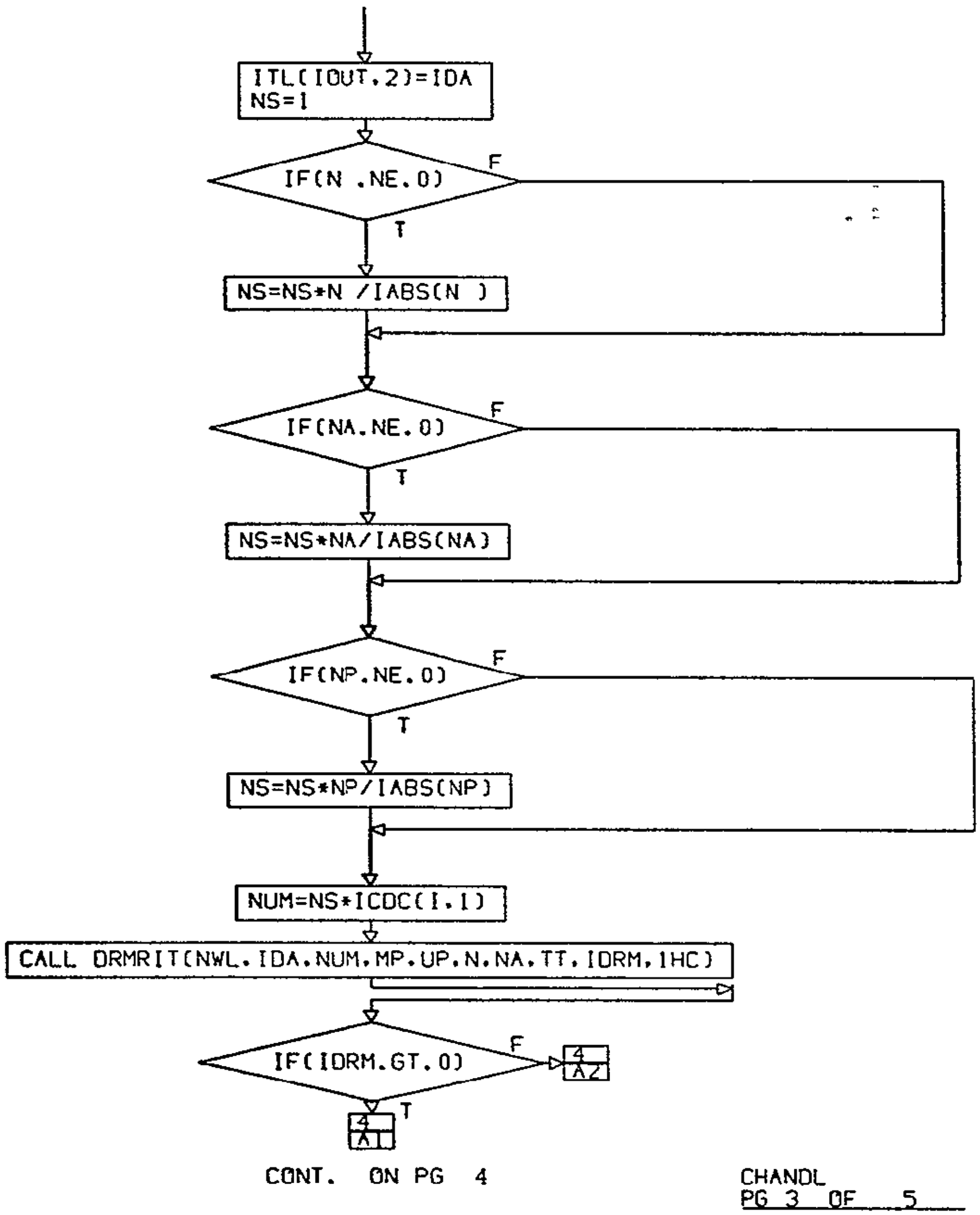
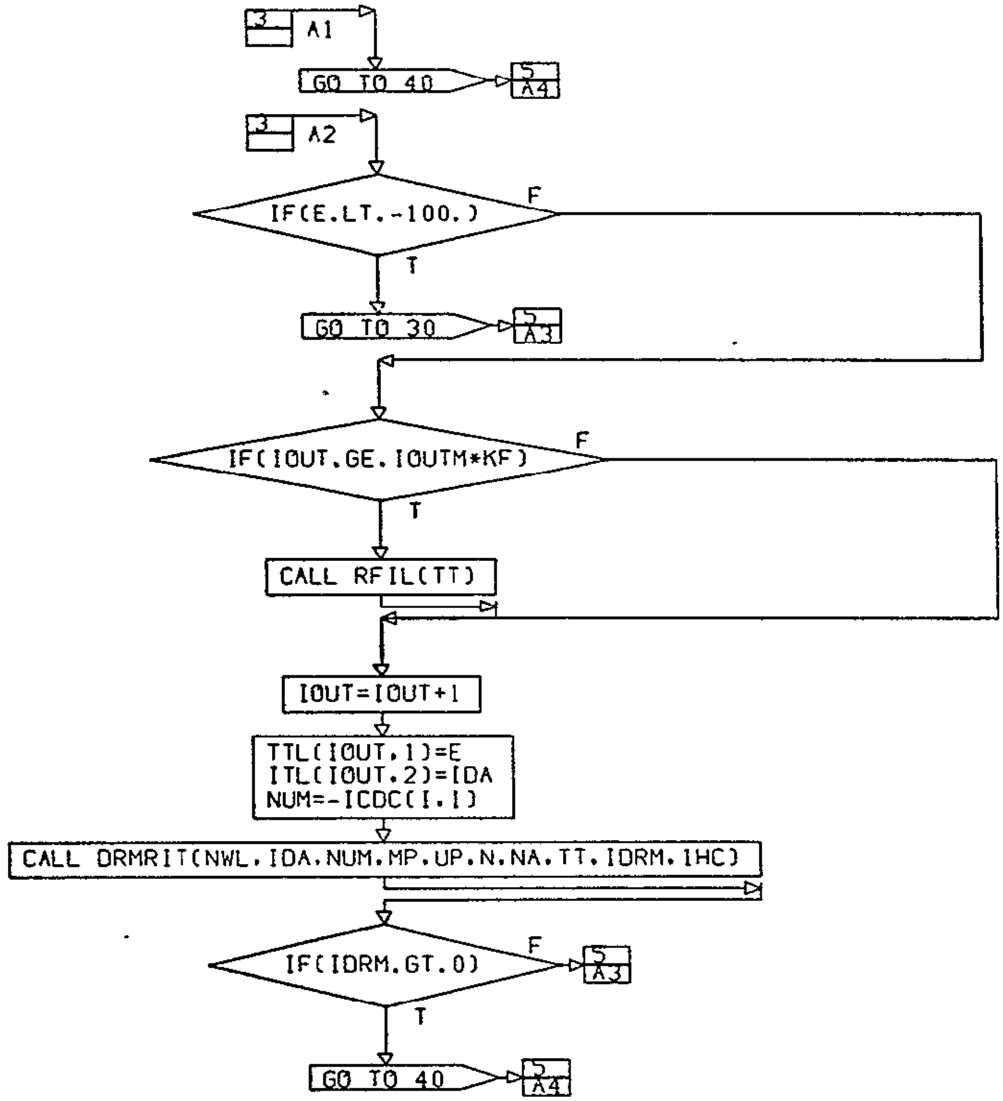


FIGURE 3.2.6. FUNCTIONAL FLOWCHART OF SUBROUTINE CHANDL (CONTINUED)



CONT. ON PG 5

CHANDL
PG 4 OF 5

FIGURE 3.2.6. FUNCTIONAL FLOWCHART OF SUBROUTINE CHANDL (CONTINUED)

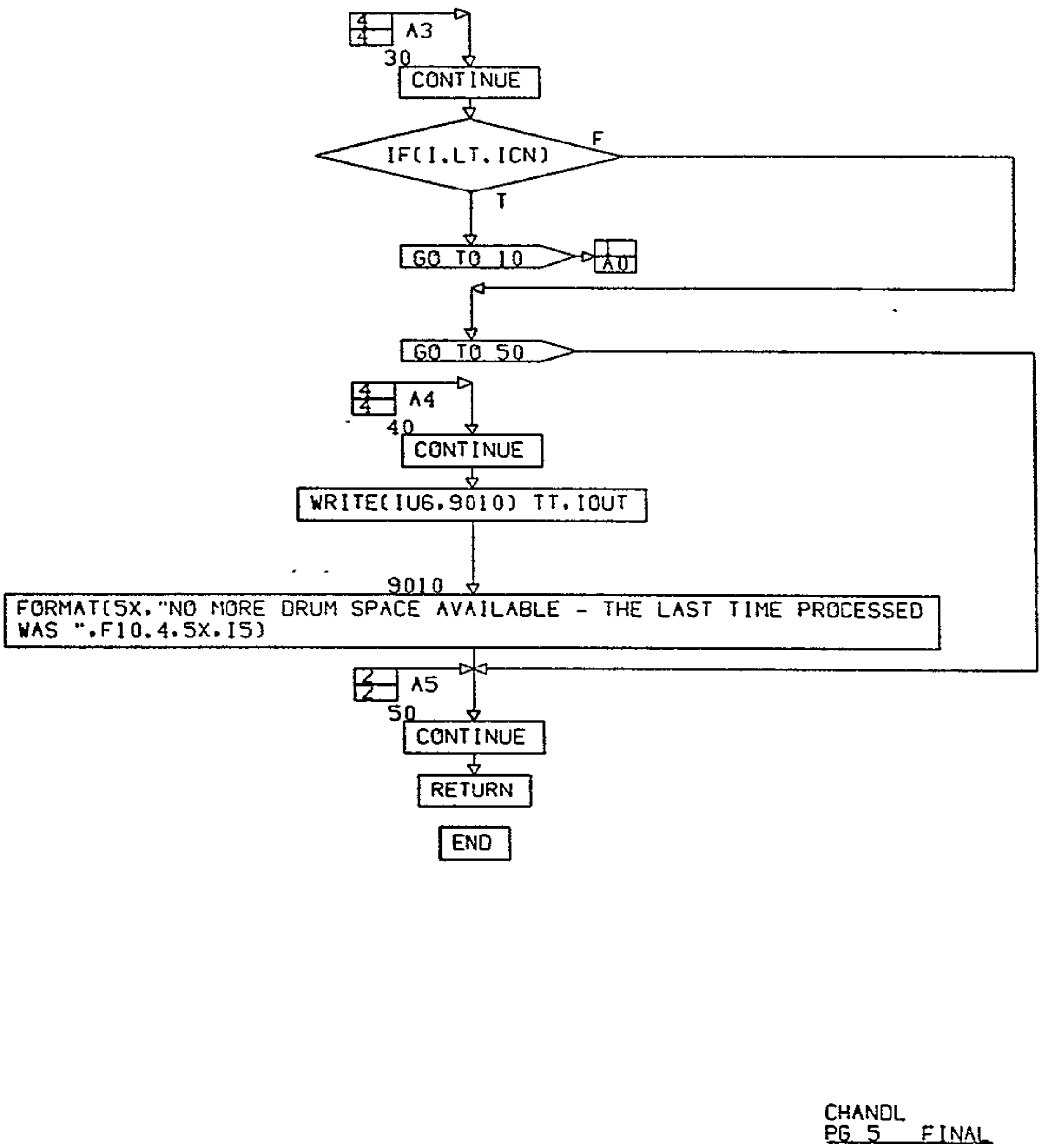


FIGURE 3.2.6. FUNCTIONAL FLOWCHART OF SUBROUTINE CHANDL (CONTINUED)

3.2.7 Subroutine: COMPCT

PURPOSE: To eliminate unused components from the component dictionary

METHOD: Only components found to be "active" are stored in the compacted dictionary. If requested, the compacted dictionary is written to an auxiliary unit for use in a later execution.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.7. See Appendix for definition of all variables.

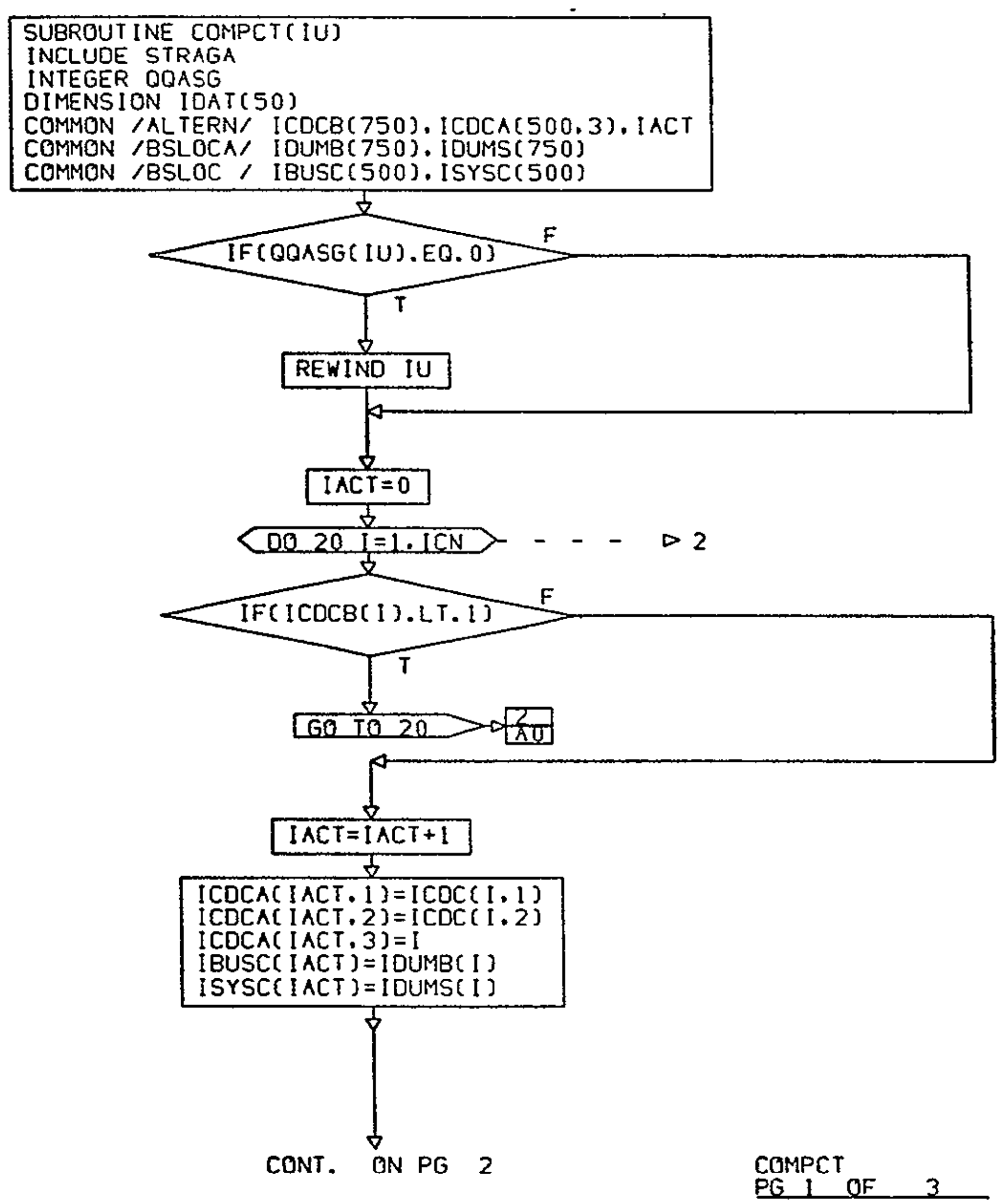


FIGURE 3.2.7. FUNCTIONAL FLOWCHART OF SUBROUTINE COMPCT

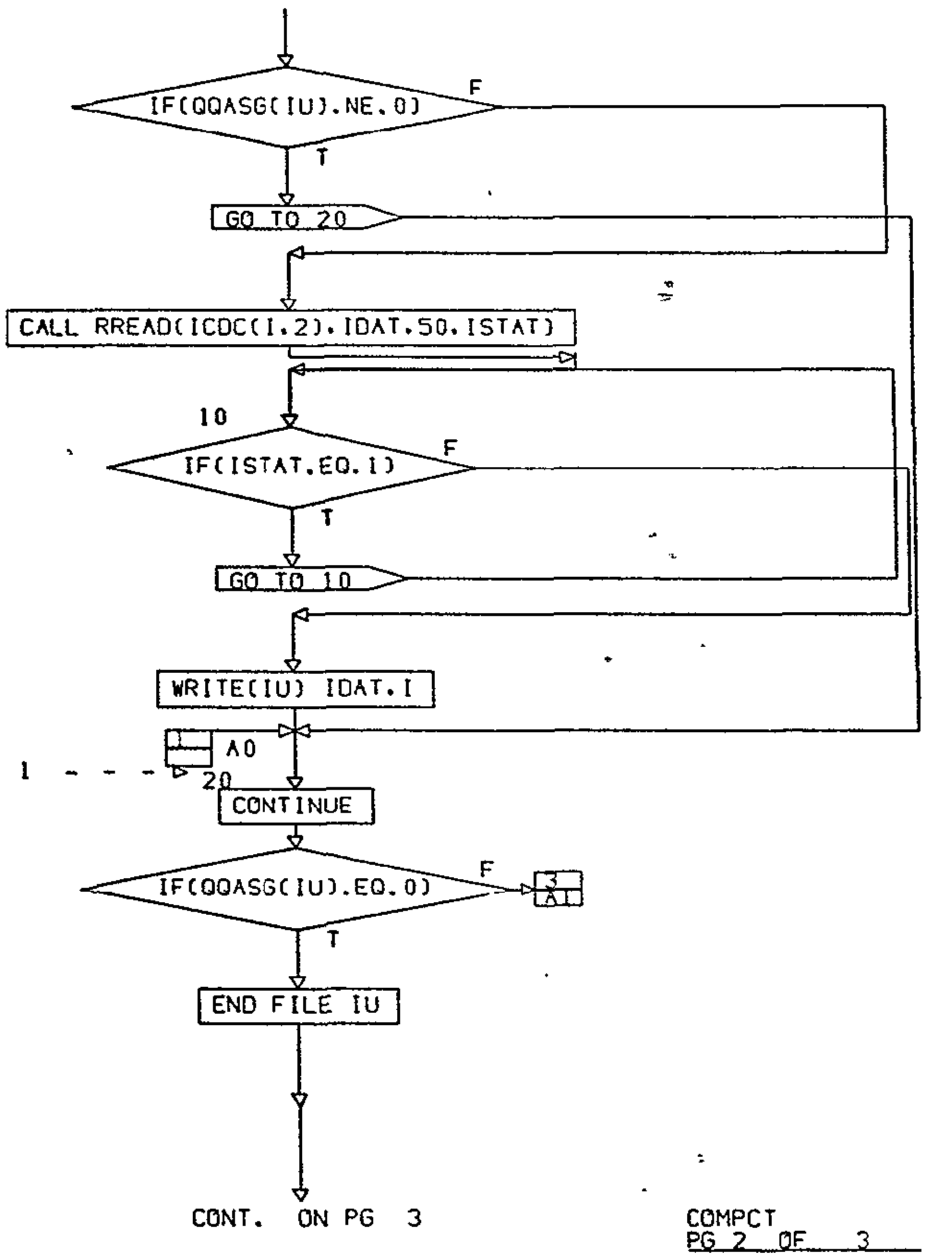
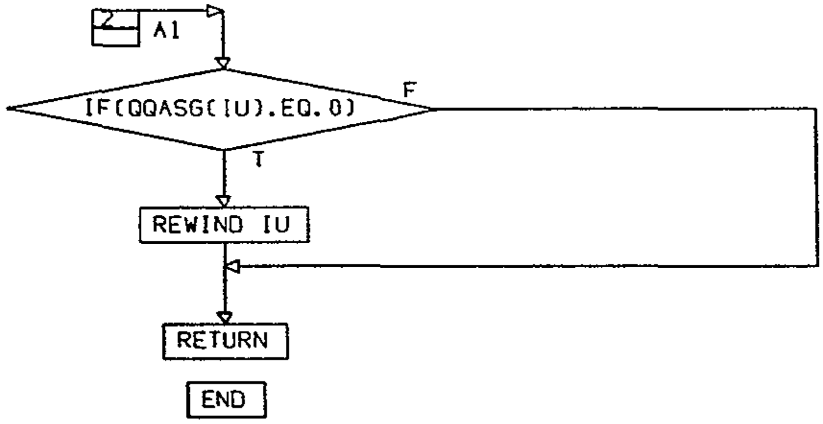


FIGURE 3.2.7. FUNCTIONAL FLOWCHART OF SUBROUTINE COMPCT (CONTINUED)



COMPCT
PG 3 FINAL

FIGURE 3.2.7. FUNCTIONAL FLOWCHART OF SUBROUTINE COMPCT (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

3.2.8. Subroutine: CREAD

PURPOSE: This routine creates the component dictionary.

METHOD: The component definition is read. The component modes are gathered together and written randomly on drum. The component dictionary consists of component ID number and drum location.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.8. See Appendix for definition of all variables.

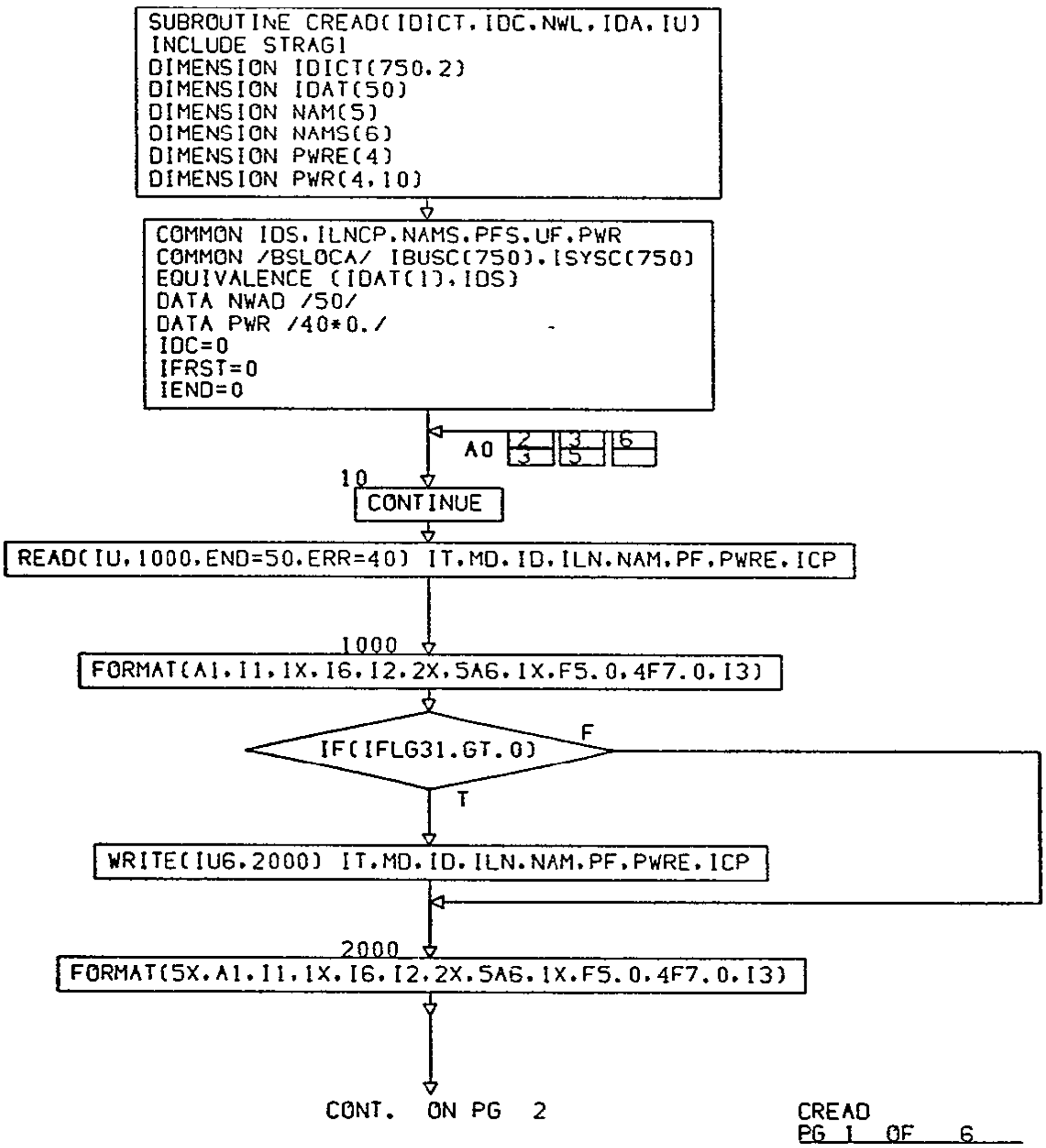
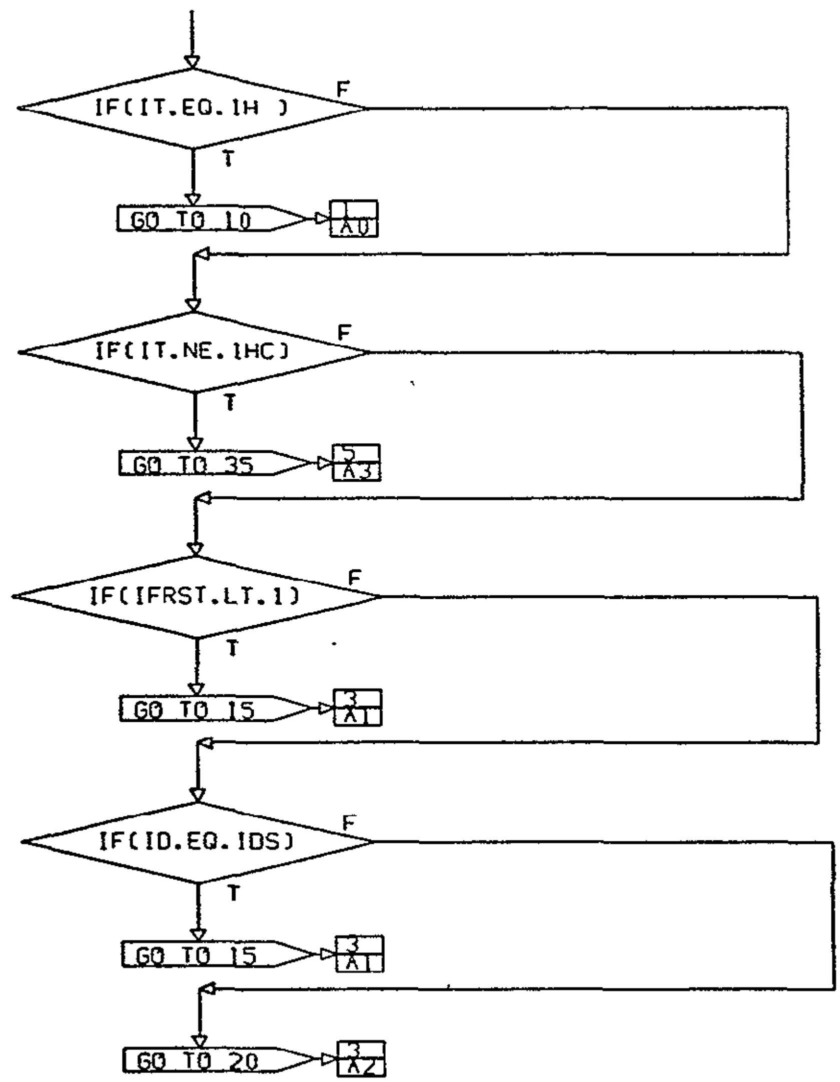


FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD



CONT. ON PG 3

CREAD
PG 2 OF 6

FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD (CONTINUED)

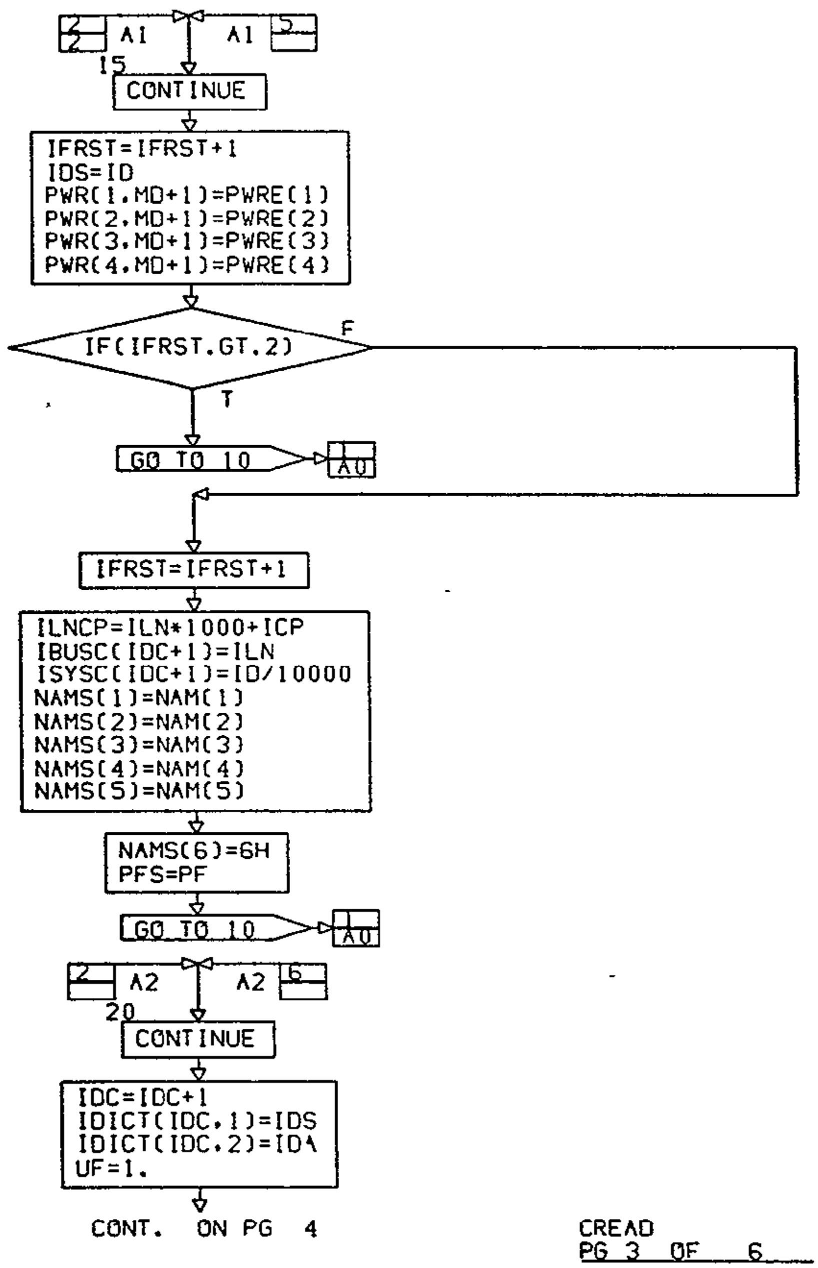
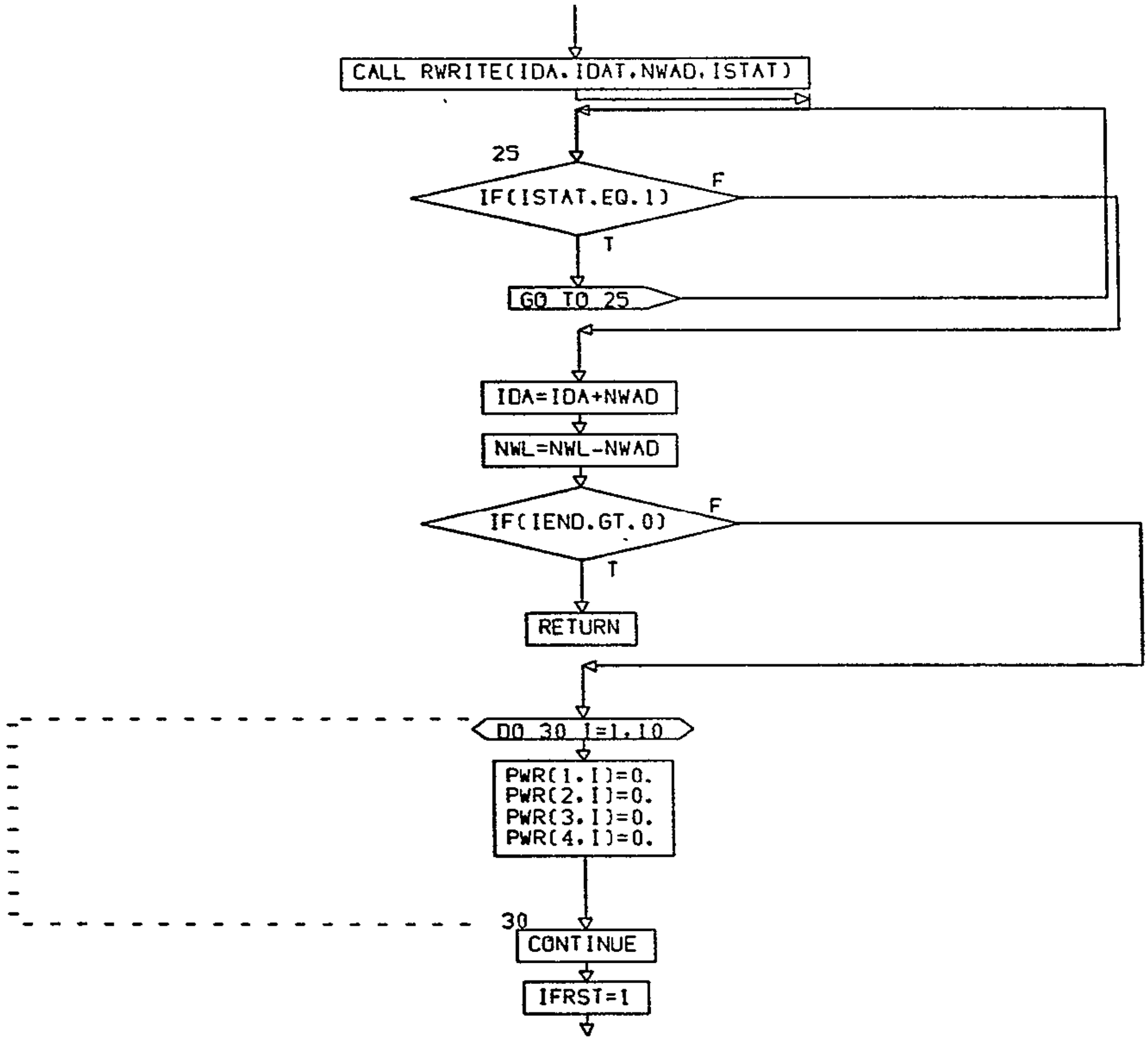


FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD (CONTINUED)



CONT. ON PG 5

CREAD
PG 4 OF 6

FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD (CONTINUED)

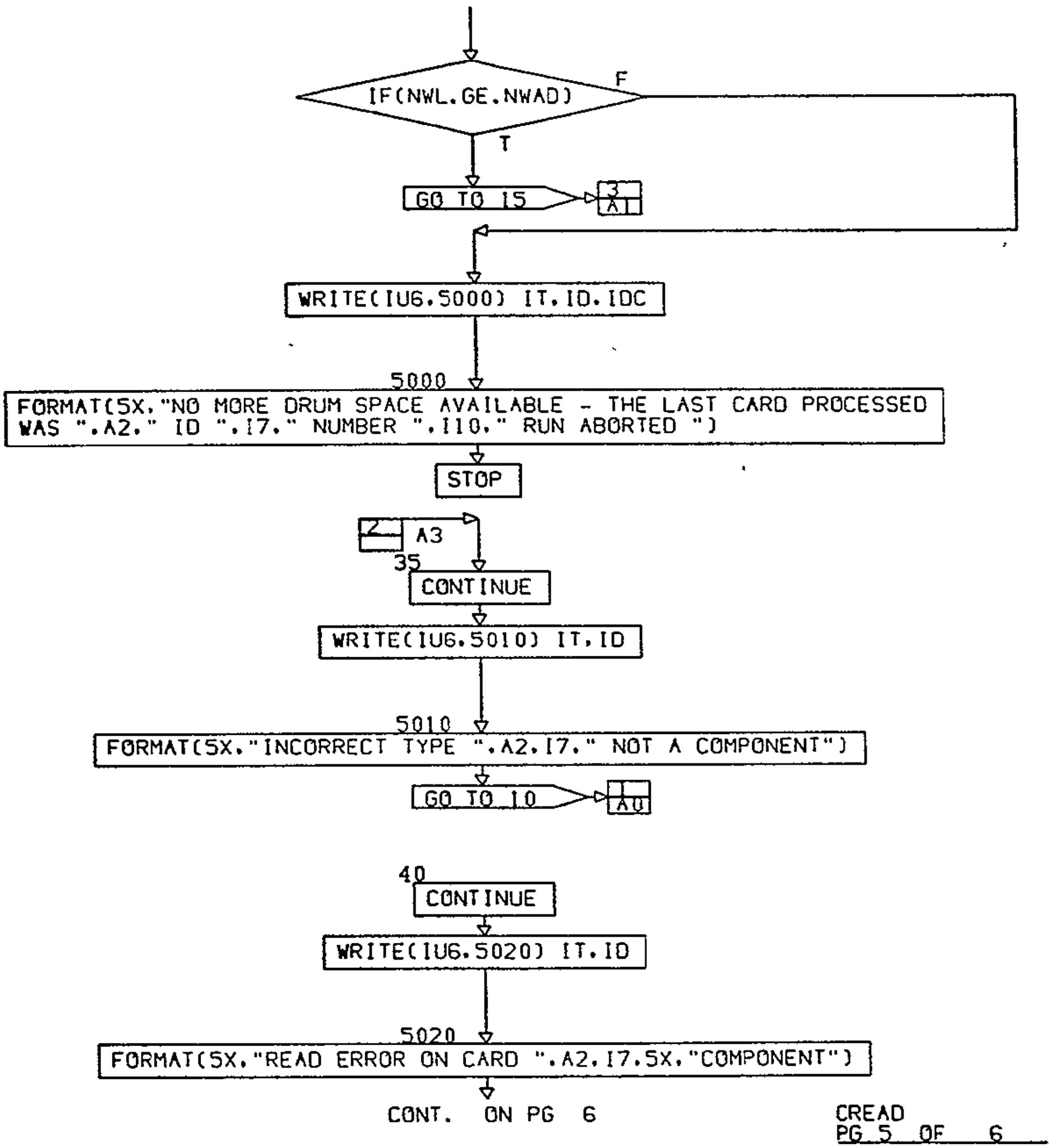
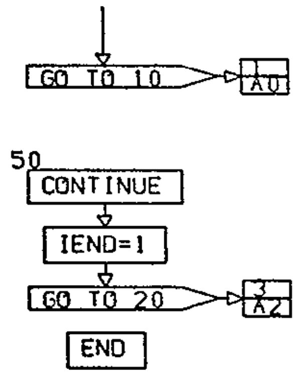


FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD (CONTINUED)



CREAD
PG 6 FINAL

FIGURE 3.2.8. FUNCTIONAL FLOWCHART OF SUBROUTINE CREAD (CONTINUED)

INTENTIONALLY

LEFT

BLANK

3.2.9 Subroutine: CTAPE

PURPOSE: To write the Phase I interface tape.

METHOD: The changes of component loads and switch positions from the previous time are determined. Only the changes are written on the interface tape. The first record contains zero values for all switch positions. The first record is used for initialization.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 2.3.9. See Appendix for definition of all variables.

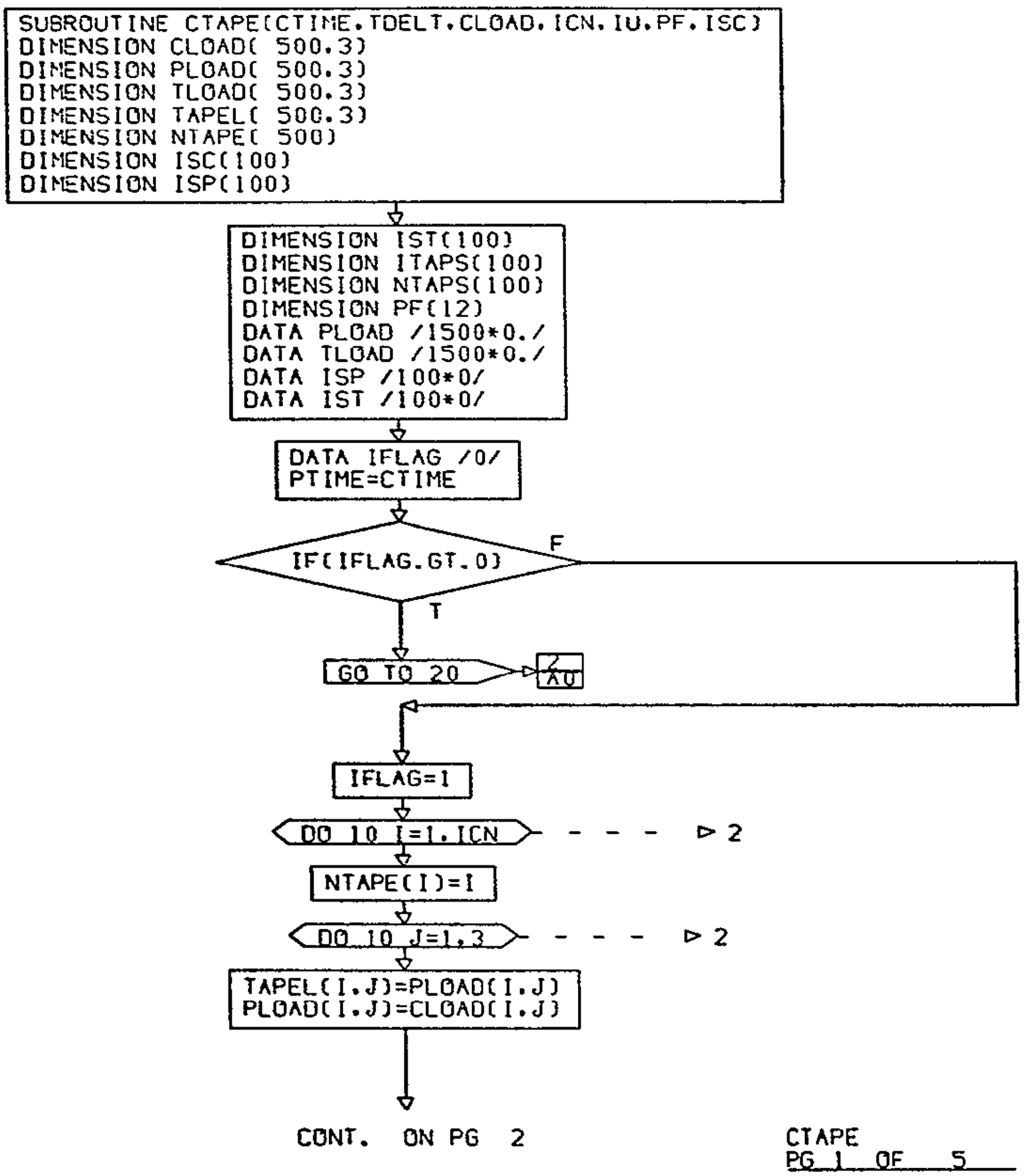


FIGURE 3.2.9. FUNCTIONAL FLOWCHART OF SUBROUTINE CTAPE

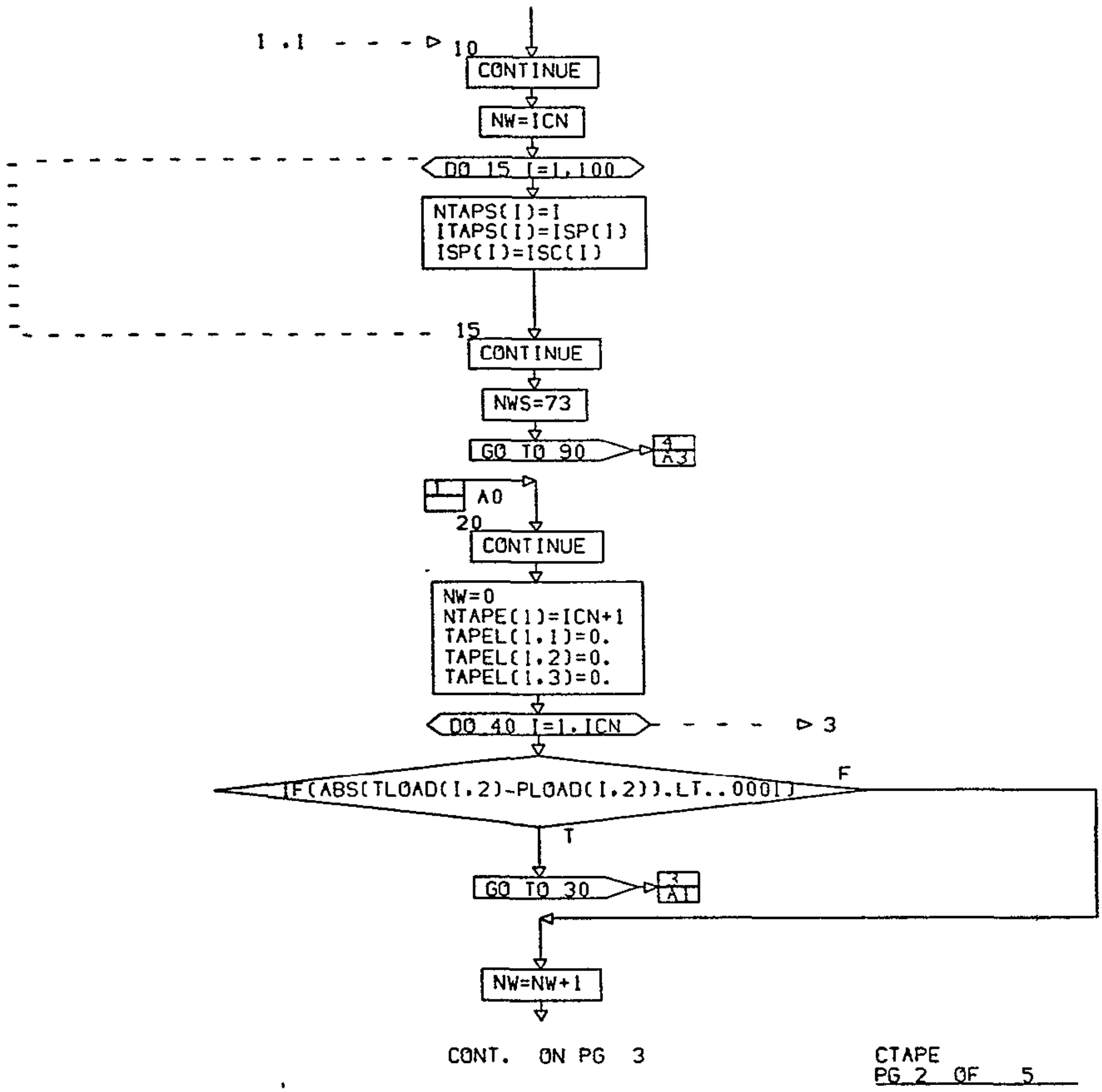
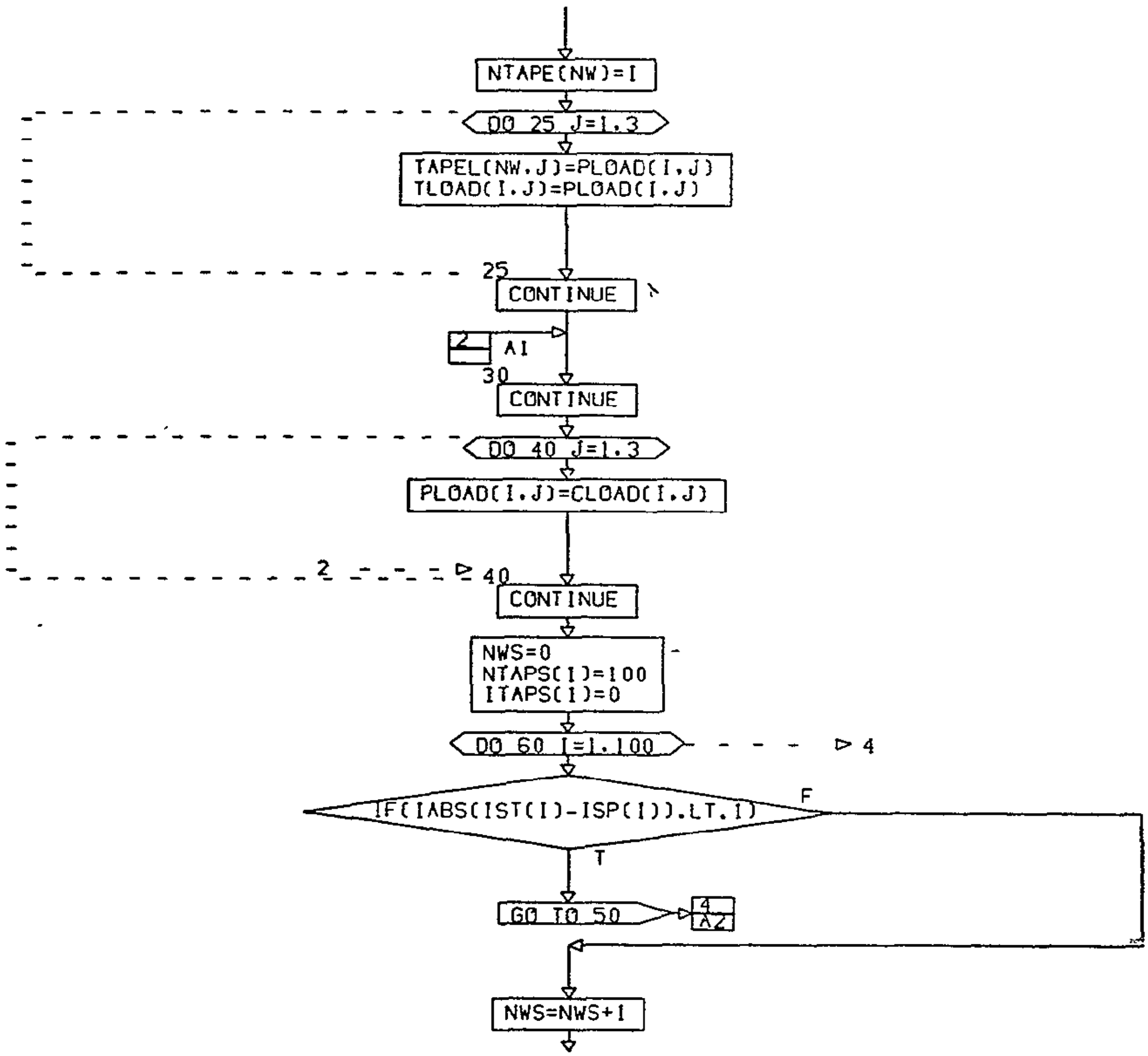


FIGURE 3.2.9. FUNCTIONAL FLOWCHART OF SUBROUTINE CTAPE (CONTINUED)



CONT. ON PG 4

CTAPE
PG 3 OF 5

FIGURE 3.2.9. FUNCTIONAL FLOWCHART OF SUBROUTINE CTAPE (CONTINUED)

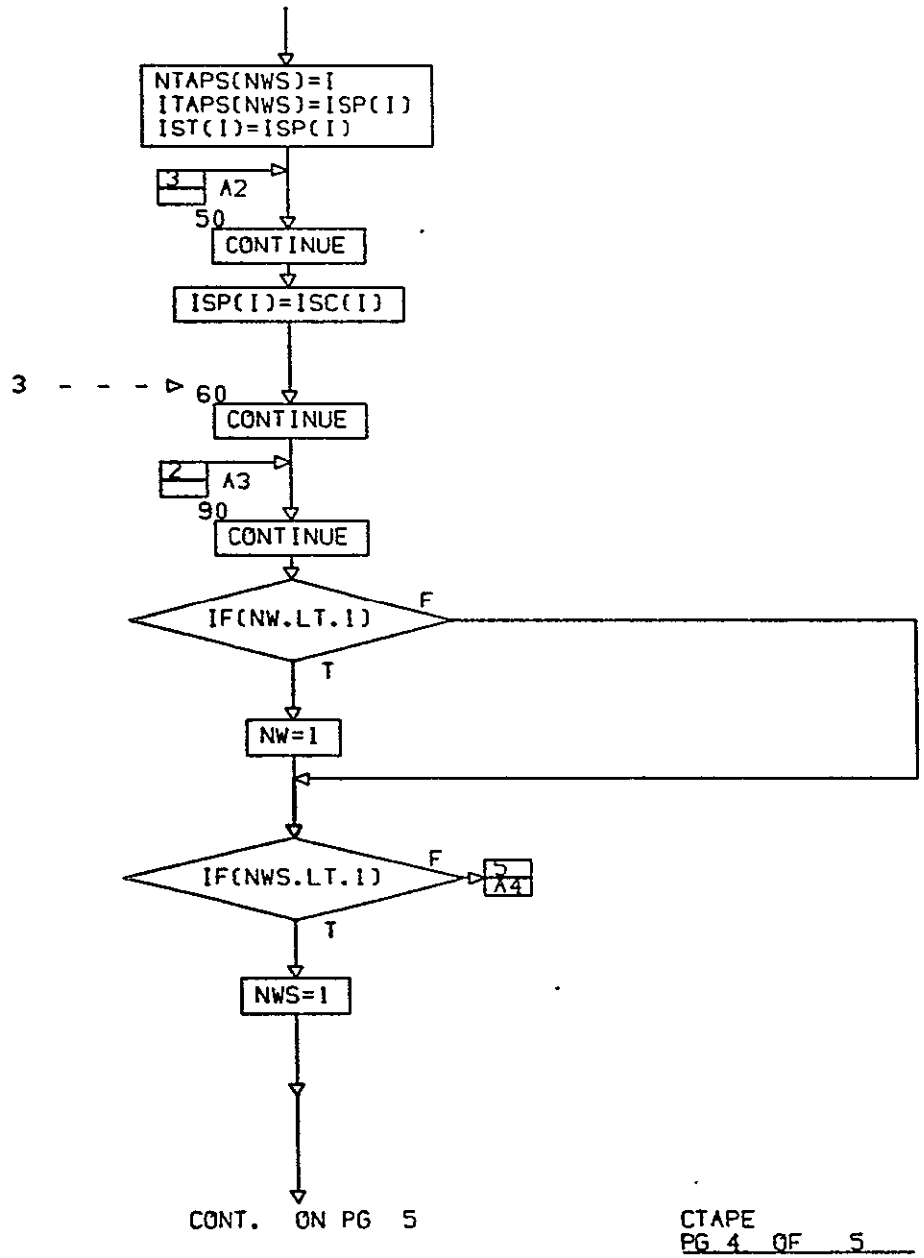
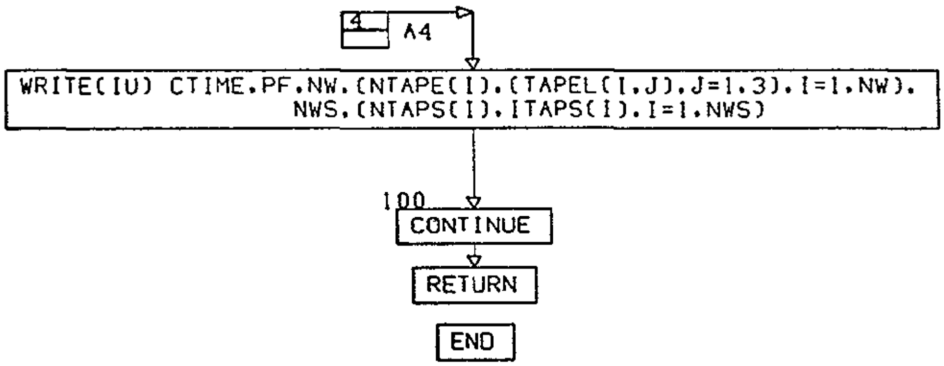


FIGURE 3.2.9. FUNCTIONAL FLOWCHART OF SUBROUTINE CTAPE (CONTINUED)



CTAPE
PG 5 FINAL

FIGURE 3.2.9. FUNCTIONAL FLOWCHART OF SUBROUTINE CTAPE (CONTINUED)

3.2.10 Subroutine: CYCLE

PURPOSE: To control all cyclic elements.

METHOD: When a cyclic element is encountered it is stored in the cyclic definition table. Periodically this routine is called to update the cyclic's condition by calling either AHANDL, PHANDL, or CHANDL depending upon the type of cyclic element.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.10. See Appendix for definition of all variables.

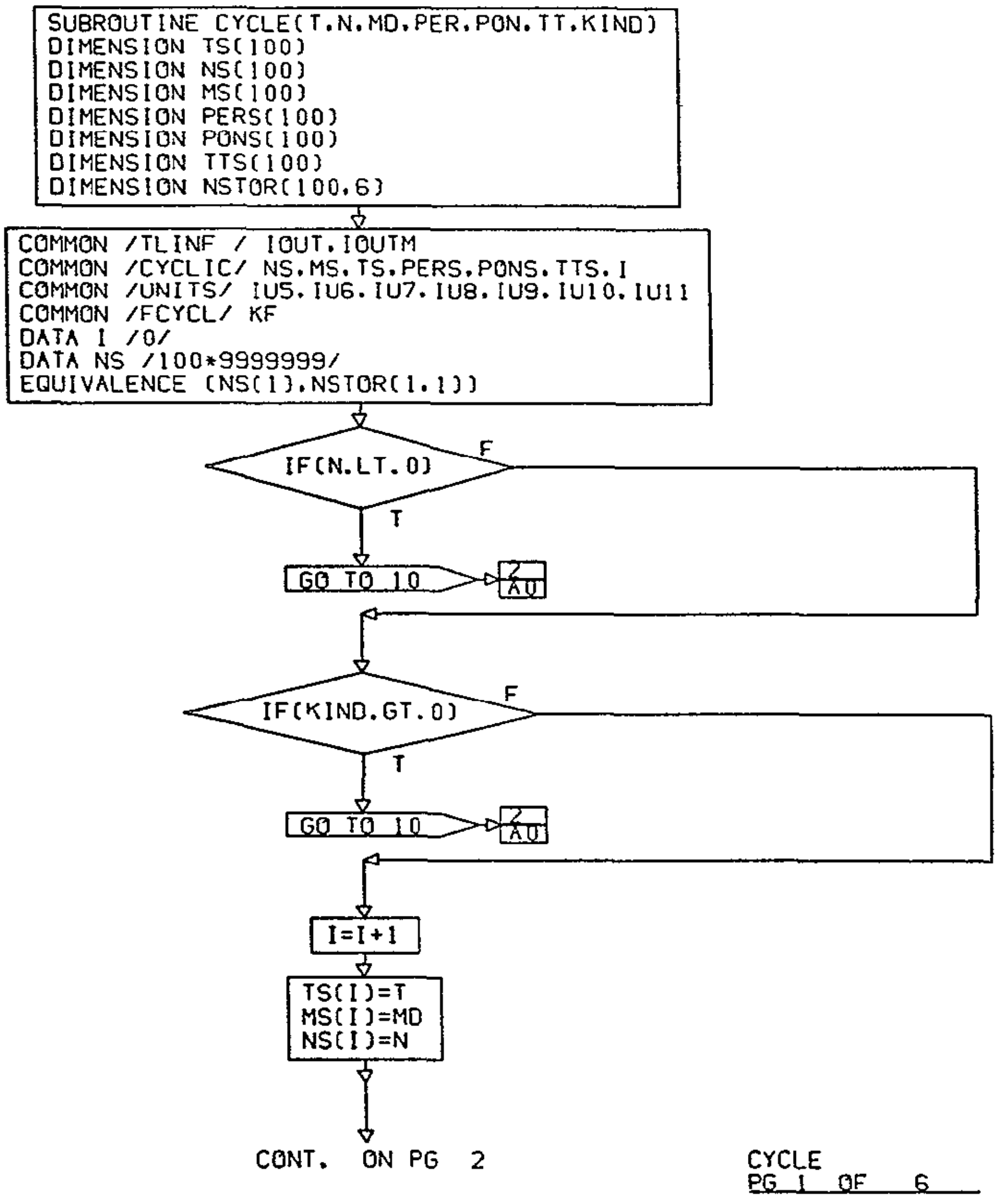


FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE

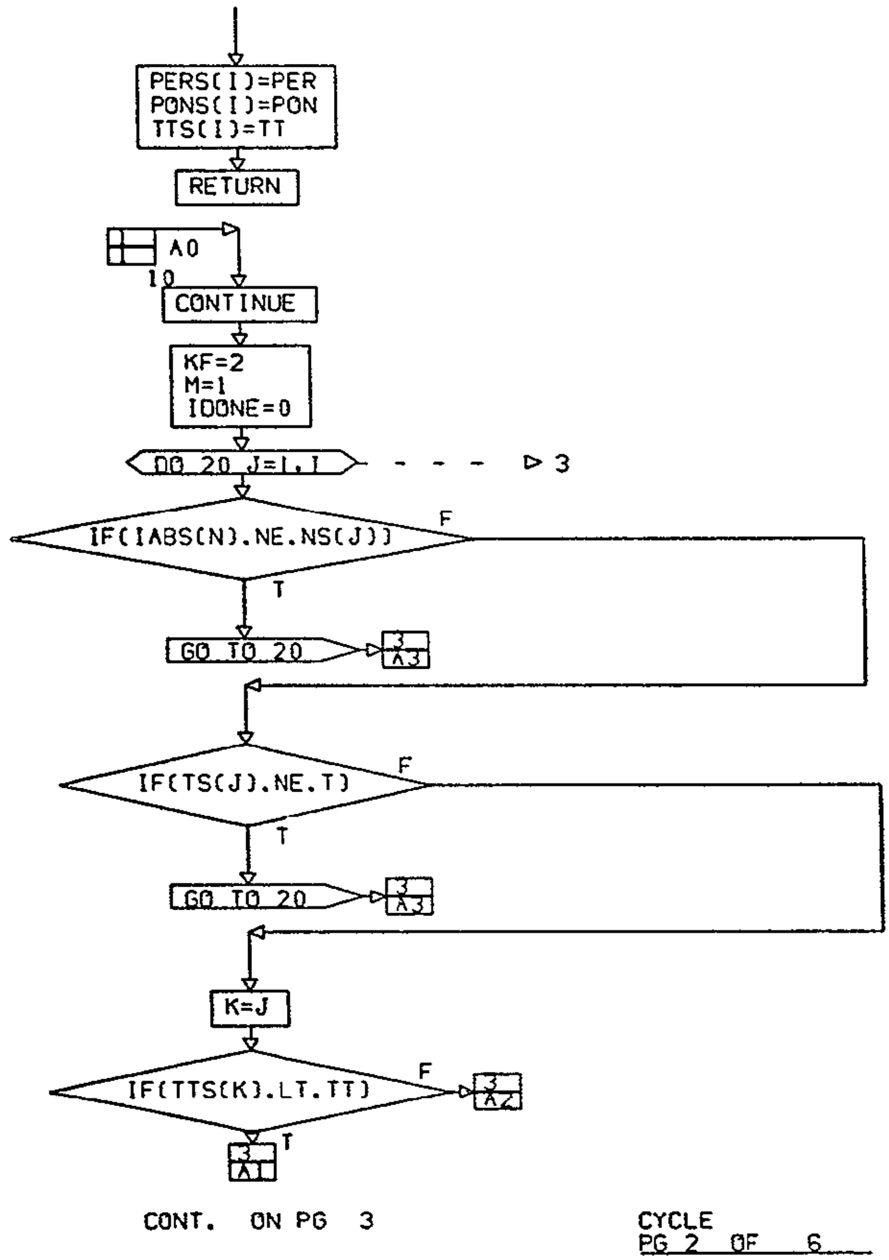


FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE (CONTINUED)

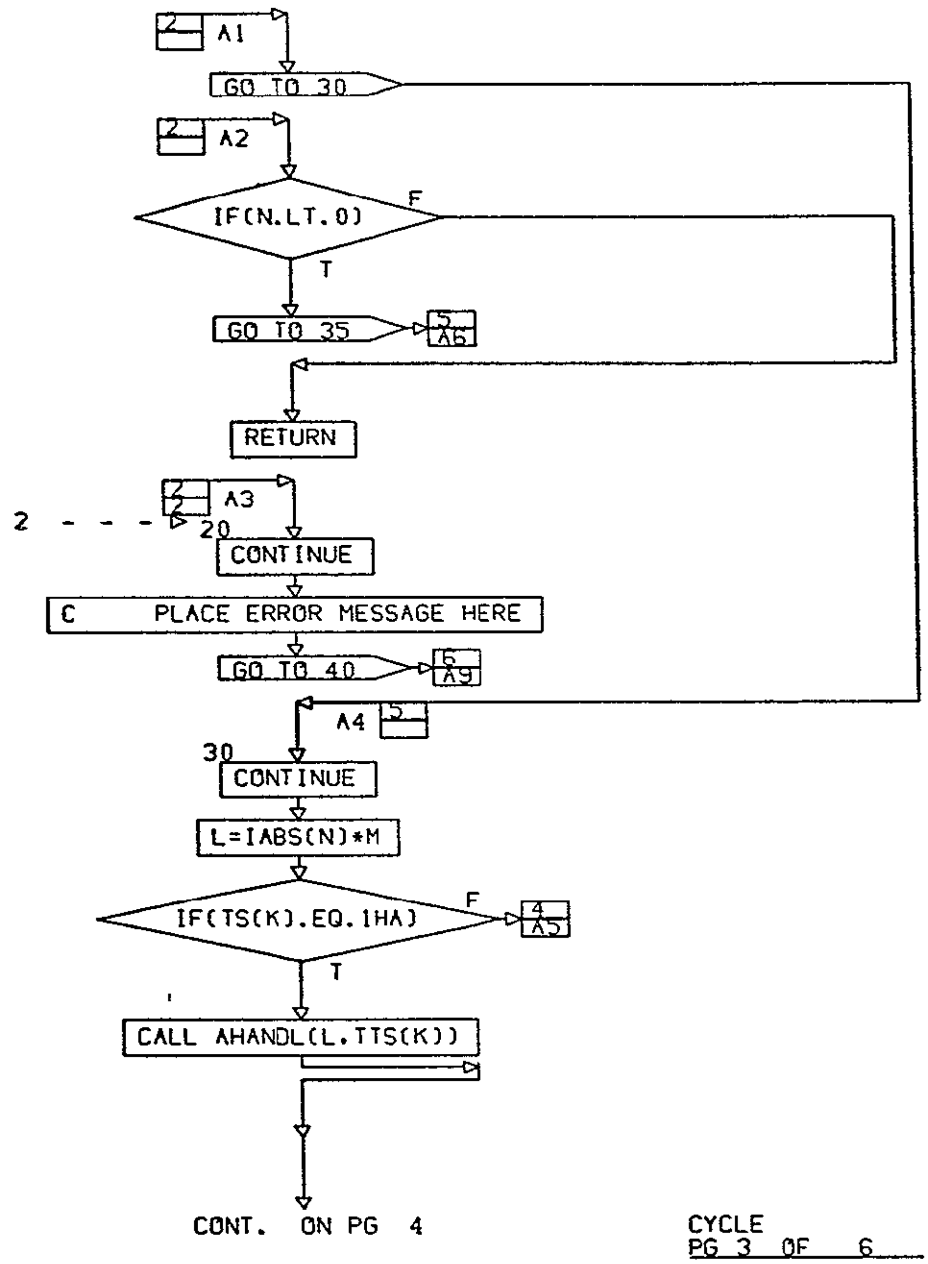


FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE (CONTINUED)

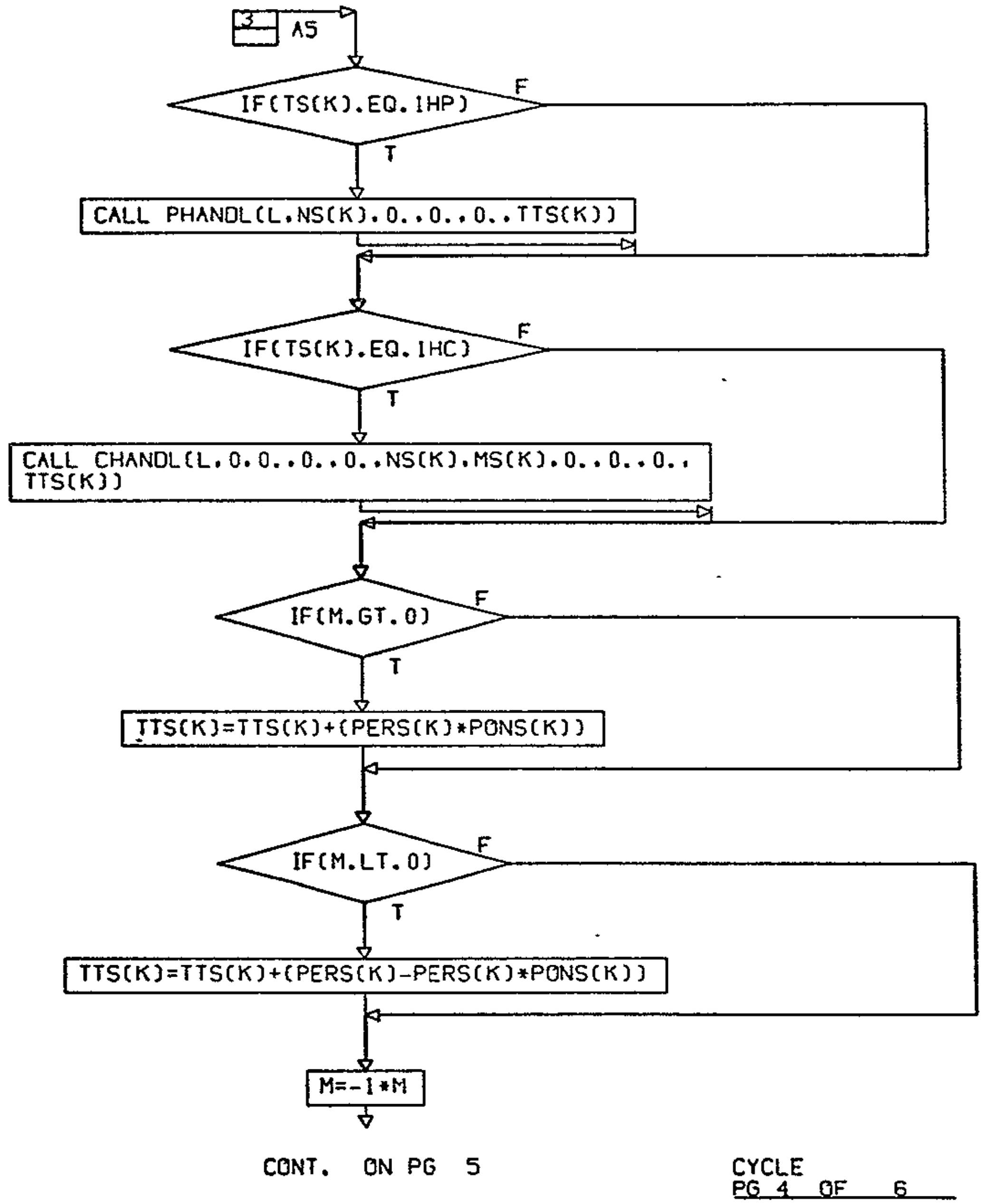


FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE (CONTINUED)

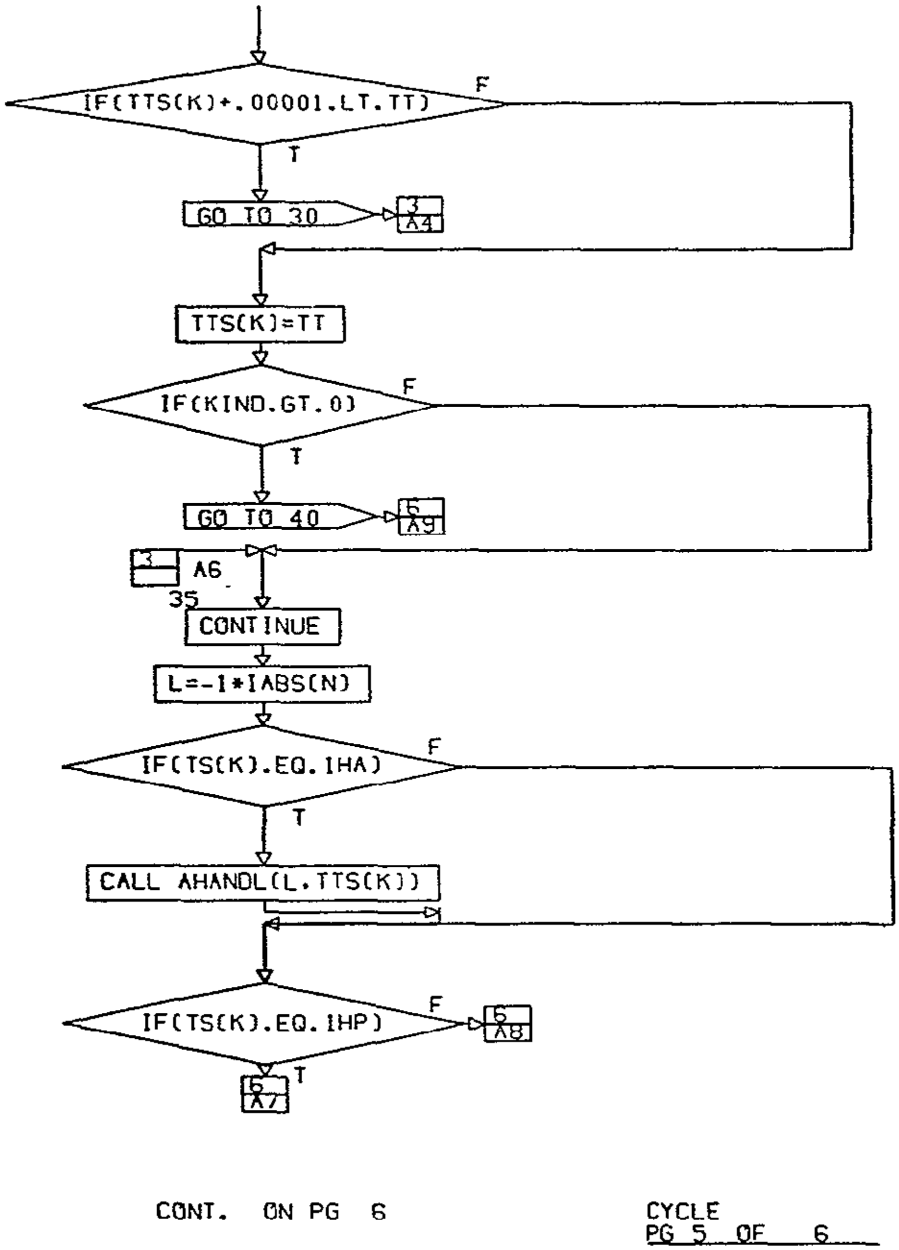
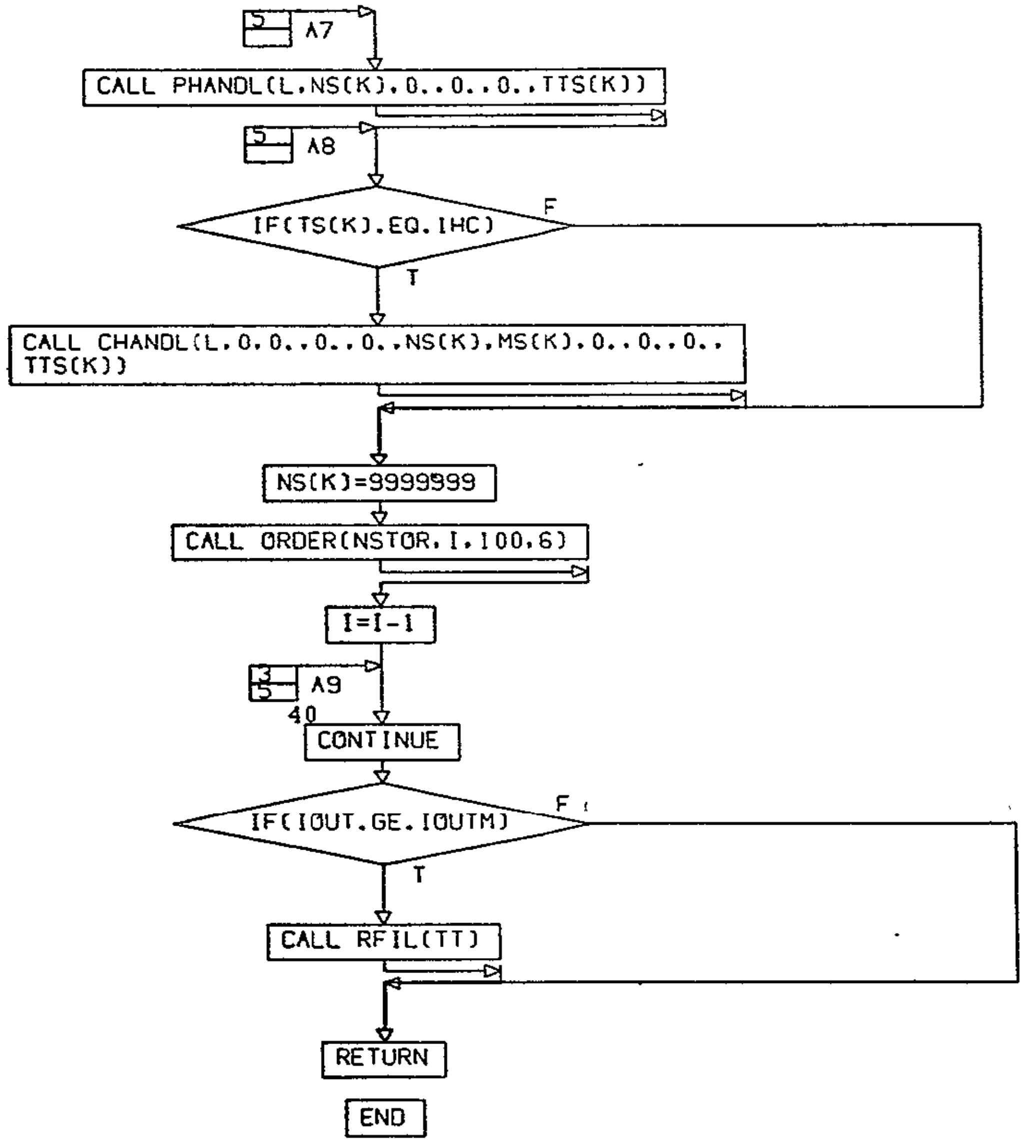


FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE (CONTINUED)



CYCLE
PG 6 FINAL

FIGURE 3.2.10. FUNCTIONAL FLOWCHART OF SUBROUTINE CYCLE (CONTINUED)

3.2.11. Subroutine: MHANDL

PURPOSE: Create the Mission Phase Definition Dictionary

METHOD: Read the Mission Phase Definition cards and store them in an array for later use.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.11. See Appendix for definition of all variables.

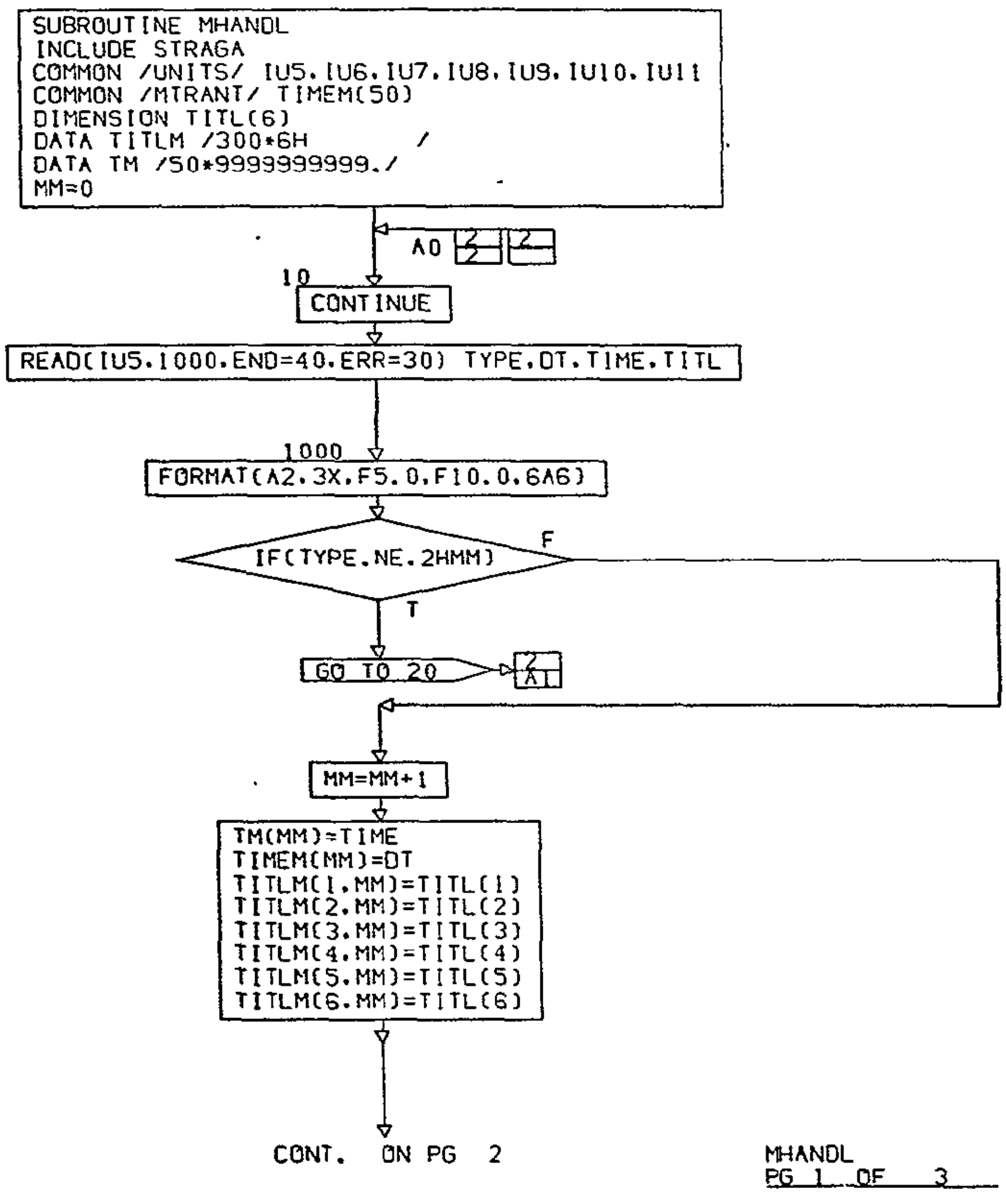


FIGURE 3.2.11. FUNCTIONAL FLOWCHART OF SUBROUTINE MHANDL

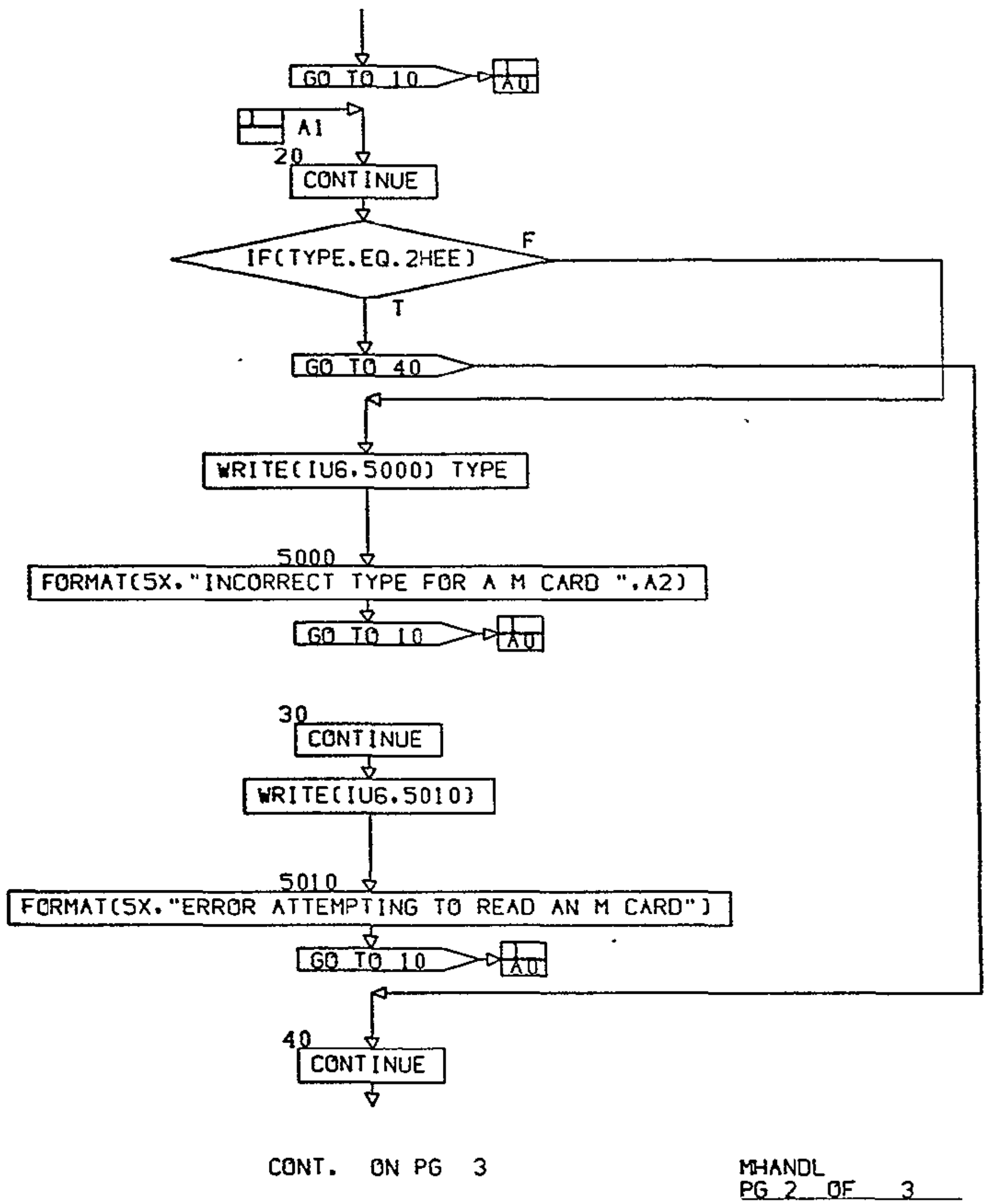


FIGURE 3.2.11. FUNCTIONAL FLOWCHART OF SUBROUTINE MHANDL (CONTINUED)

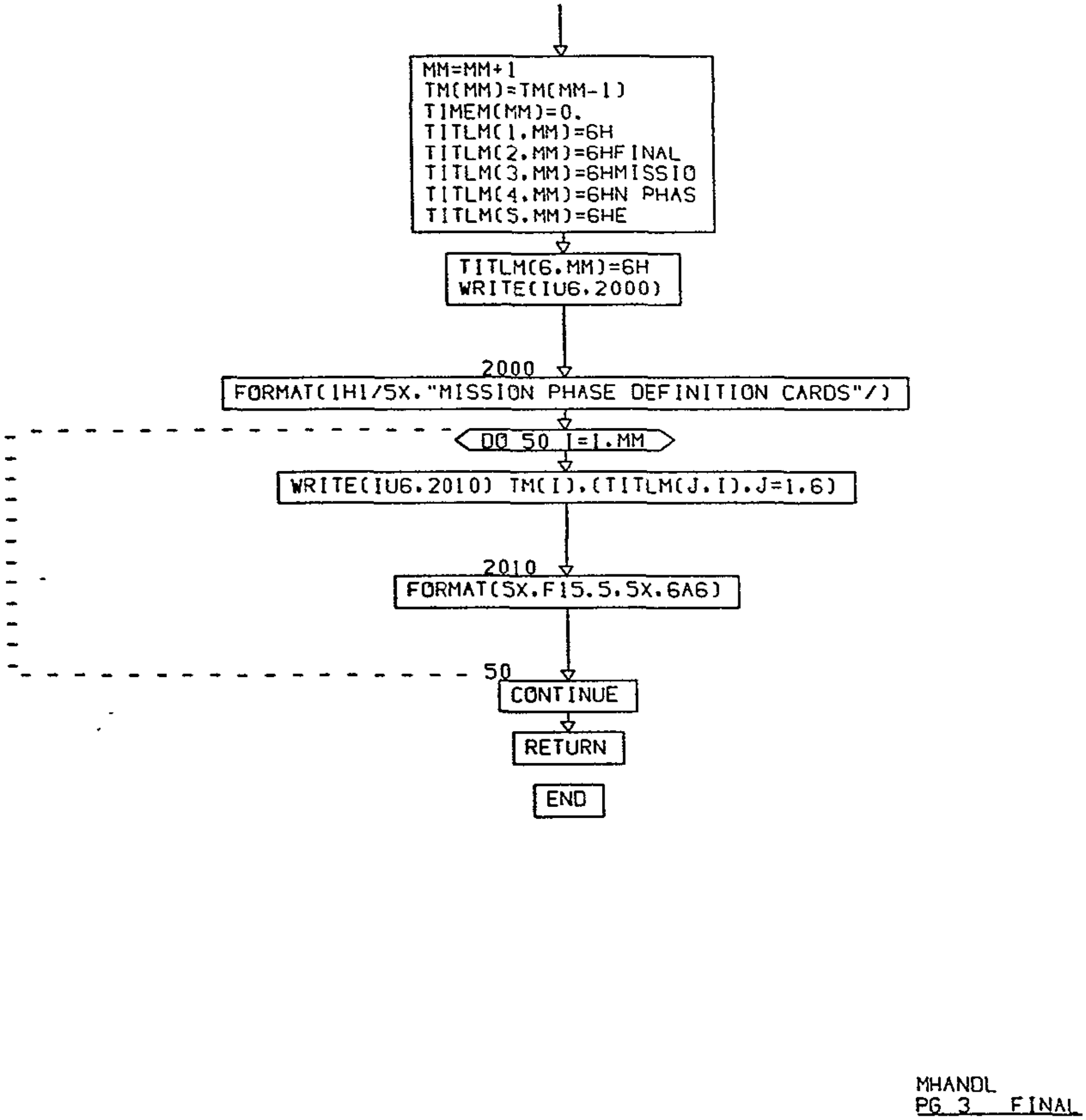


FIGURE 3.2.11. FUNCTIONAL FLOWCHART OF SUBROUTINE MHANDL (CONTINUED)

3.2.12 Subroutine: PCYCLE

PURPOSE: This routine converts Procedures into Components.

METHOD: This routine interrogates the Procedure dictionary and calls the appropriate subroutine to correctly handle Components. If the Procedure cannot be found, the following diagnostic is generated.

REQUESTED PROCEDURE NNNNNNNNNN IS NOT IN THE DICTIONARY

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.12. See Appendix for definition of all variables.

NOTE: Subroutine PCYCLE is essentially identical to Subroutine PHANDL. The requirement for these subroutines is dictated by the program logic.

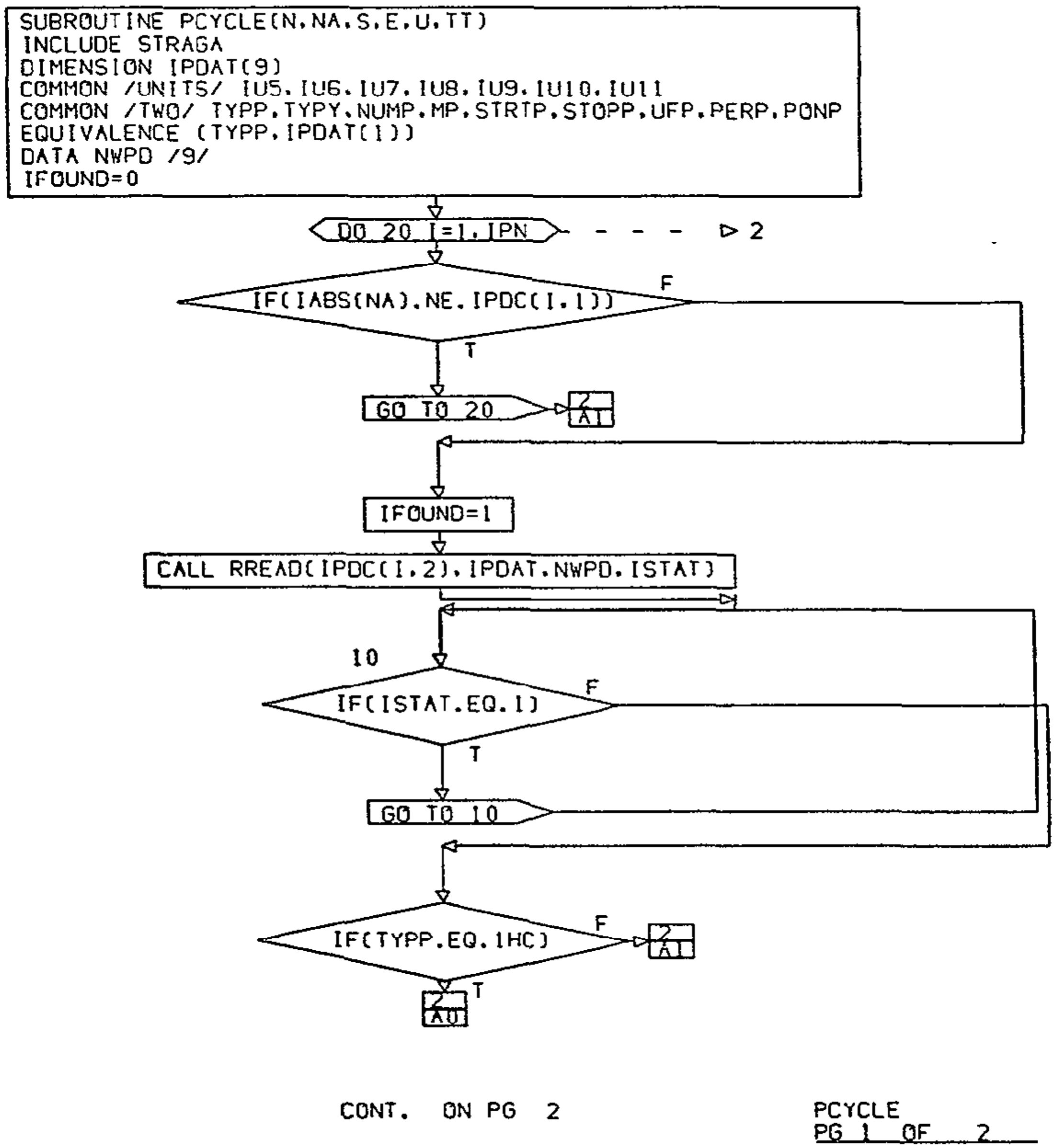


FIGURE 3.2.12. FUNCTIONAL FLOWCHART OF SUBROUTINE PCYCLE

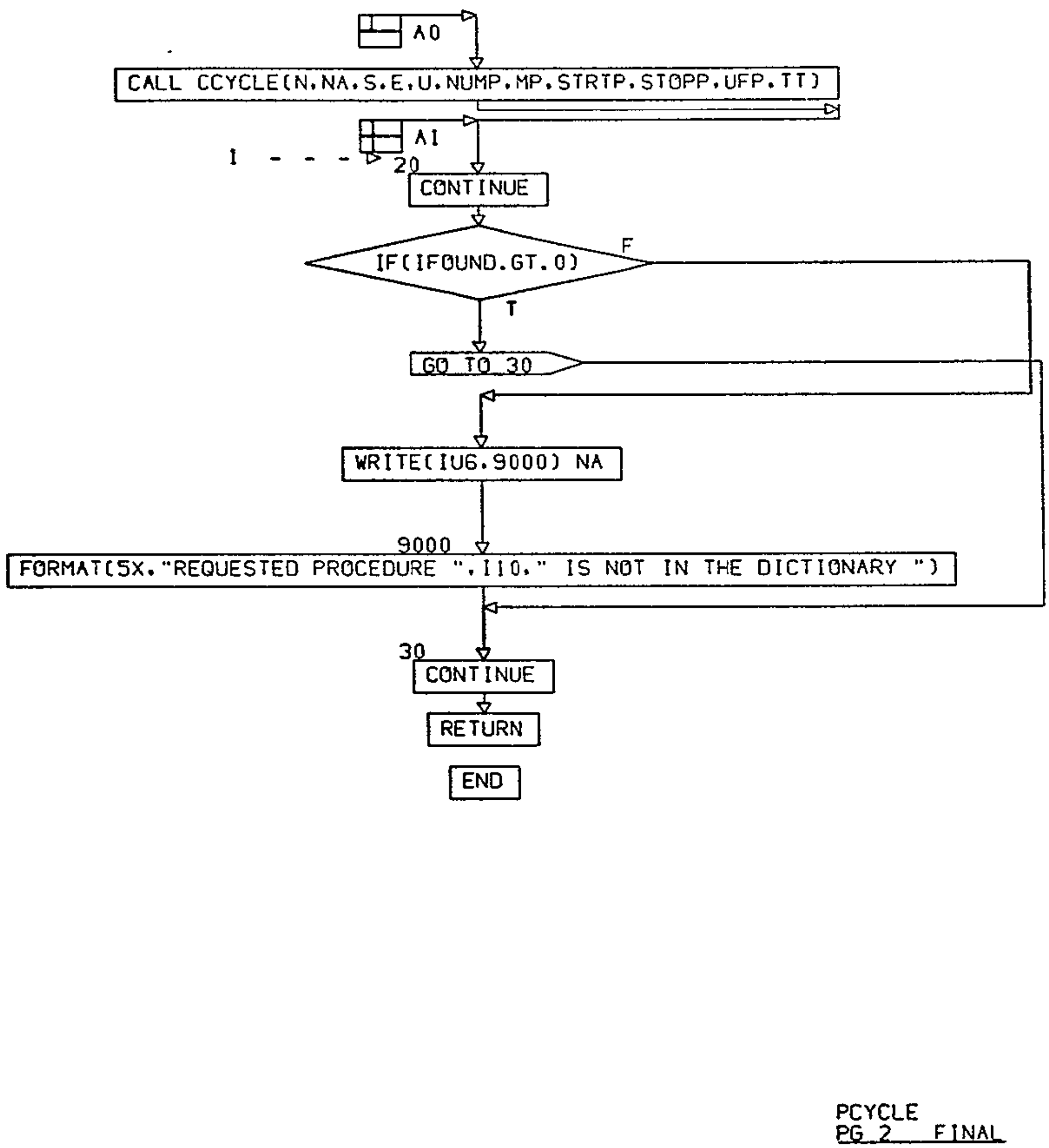


FIGURE 3.2.12. FUNCTIONAL FLOWCHART OF SUBROUTINE PCYCLE (CONTINUED)

3.2.13 Subroutine: PHANDL

PURPOSE: This routine converts a Procedure into Components and switches.

METHOD: This routine interrogates the Procedure dictionary and calls the appropriate subroutines to correctly handle Components and Switches. If the Procedure cannot be found, the following diagnostic is generated.

REQUESTED PROCEDURE NNNNNNNNNN IS NOT IN THE DICTIONARY

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.13. See Appendix for definition of all variables.

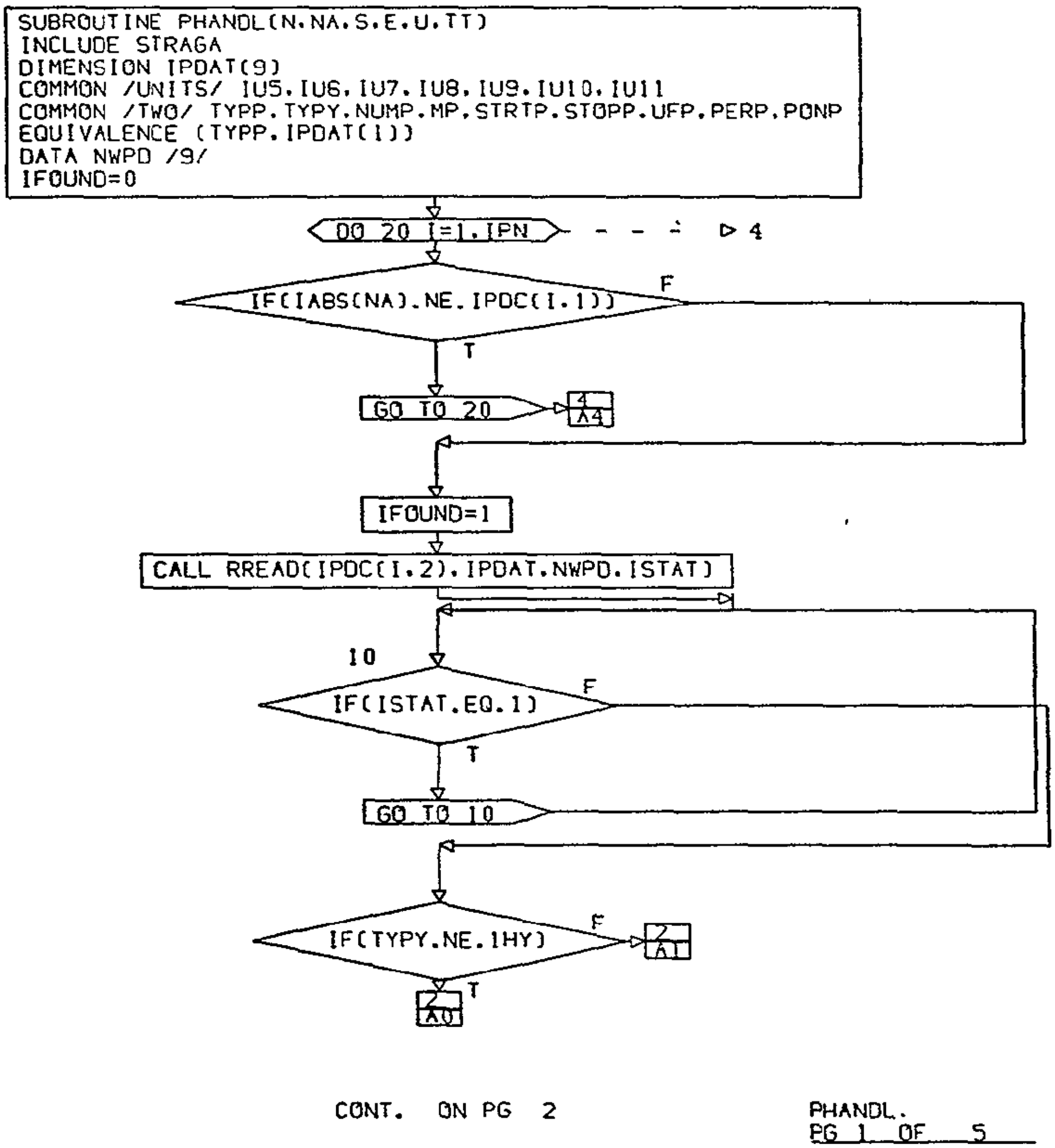
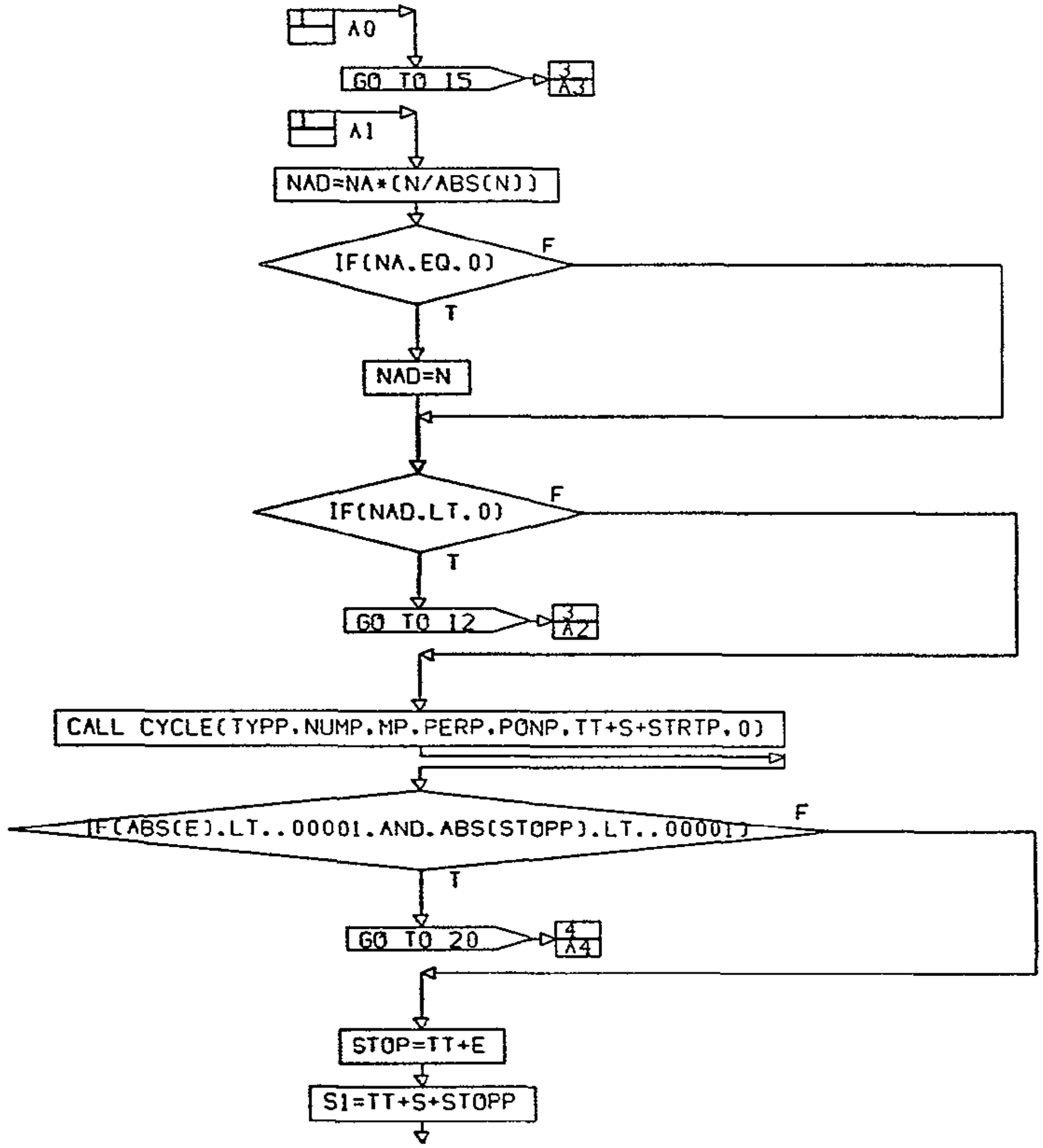


FIGURE 3.2.13. FUNCTIONAL FLOWCHART OF SUBROUTINE PHANDL

ORIGINAL PAGE IS
OF POOR QUALITY



CONT. ON PG 3

PHANDL
PG 2 OF 5

FIGURE 3.2.13. FUNCTIONAL FLOWCHART OF SUBROUTINE PHANDL (CONTINUED)

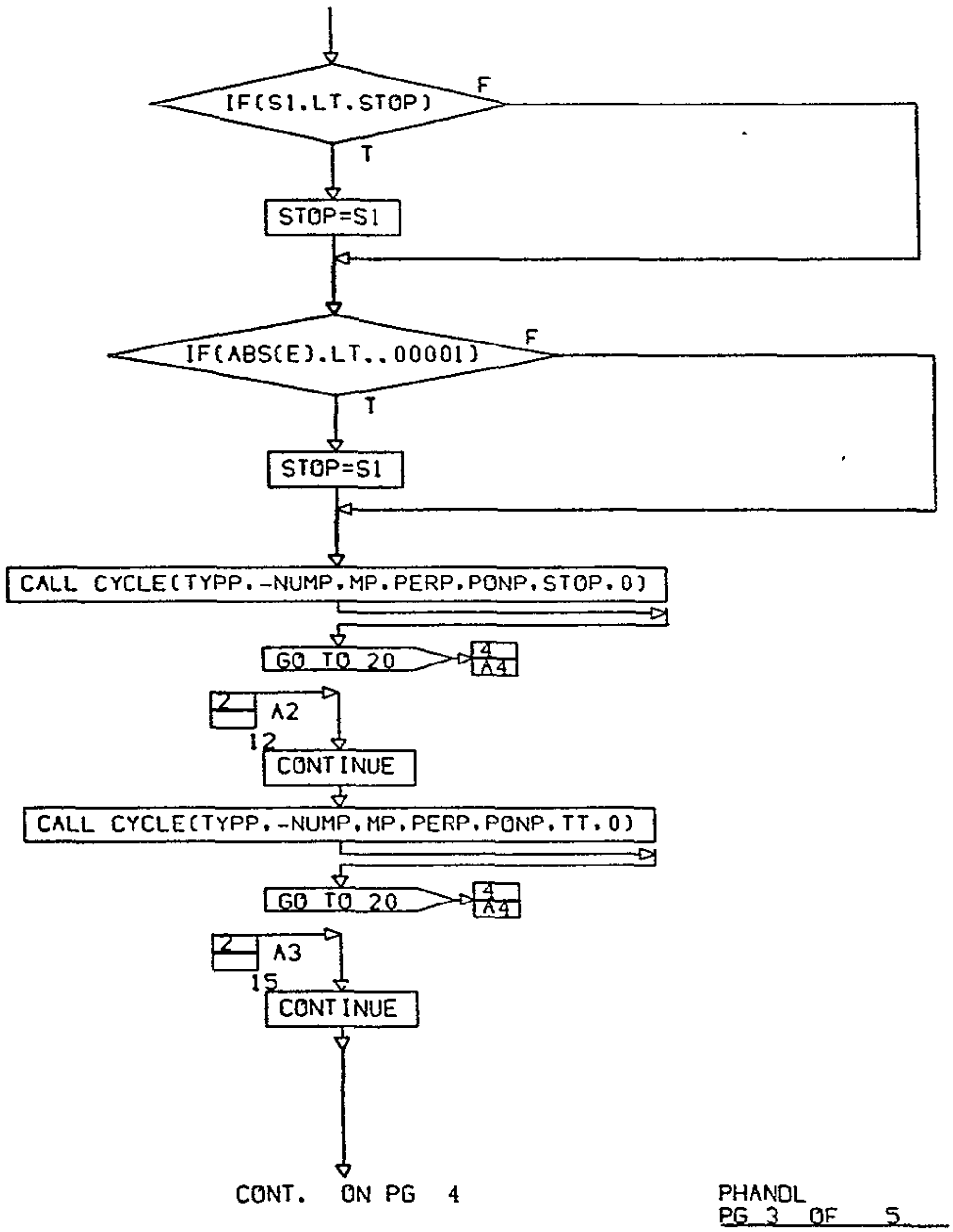
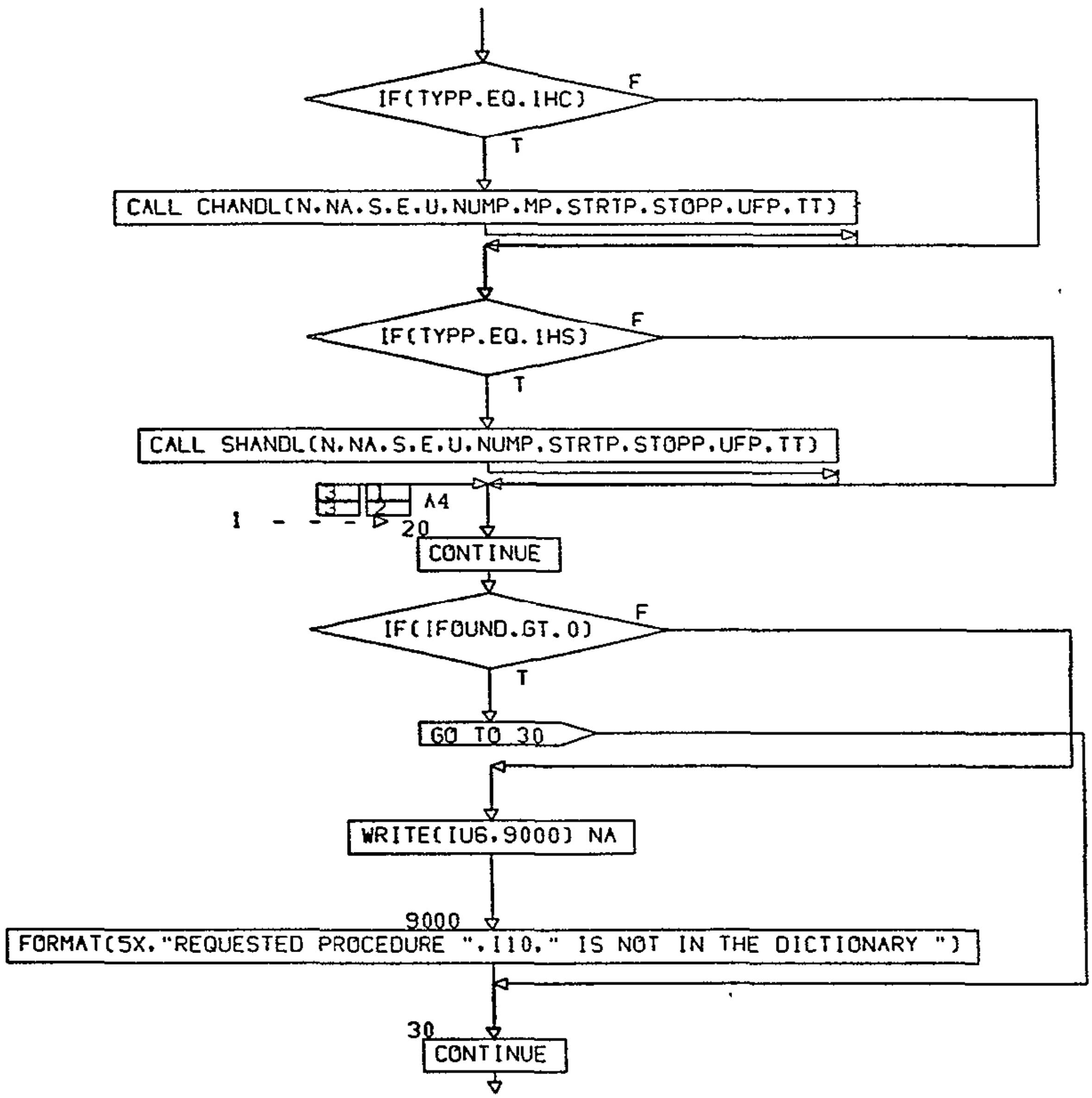


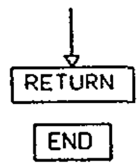
FIGURE 3.2.13. FUNCTIONAL FLOWCHART OF SUBROUTINE PHANDL (CONTINUED)



CONT. ON PG 5

PHANDL
PG 4 OF 5

FIGURE 3.2.13. FUNCTIONAL FLOWCHART FOR SUBROUTINE PHANDL (CONTINUED)



PHANDL
PG 5 FINAL

FIGURE 3.2.13. FUNCTIONAL FLOWCHART OF SUBROUTINE PHANDL (CONTINUED)

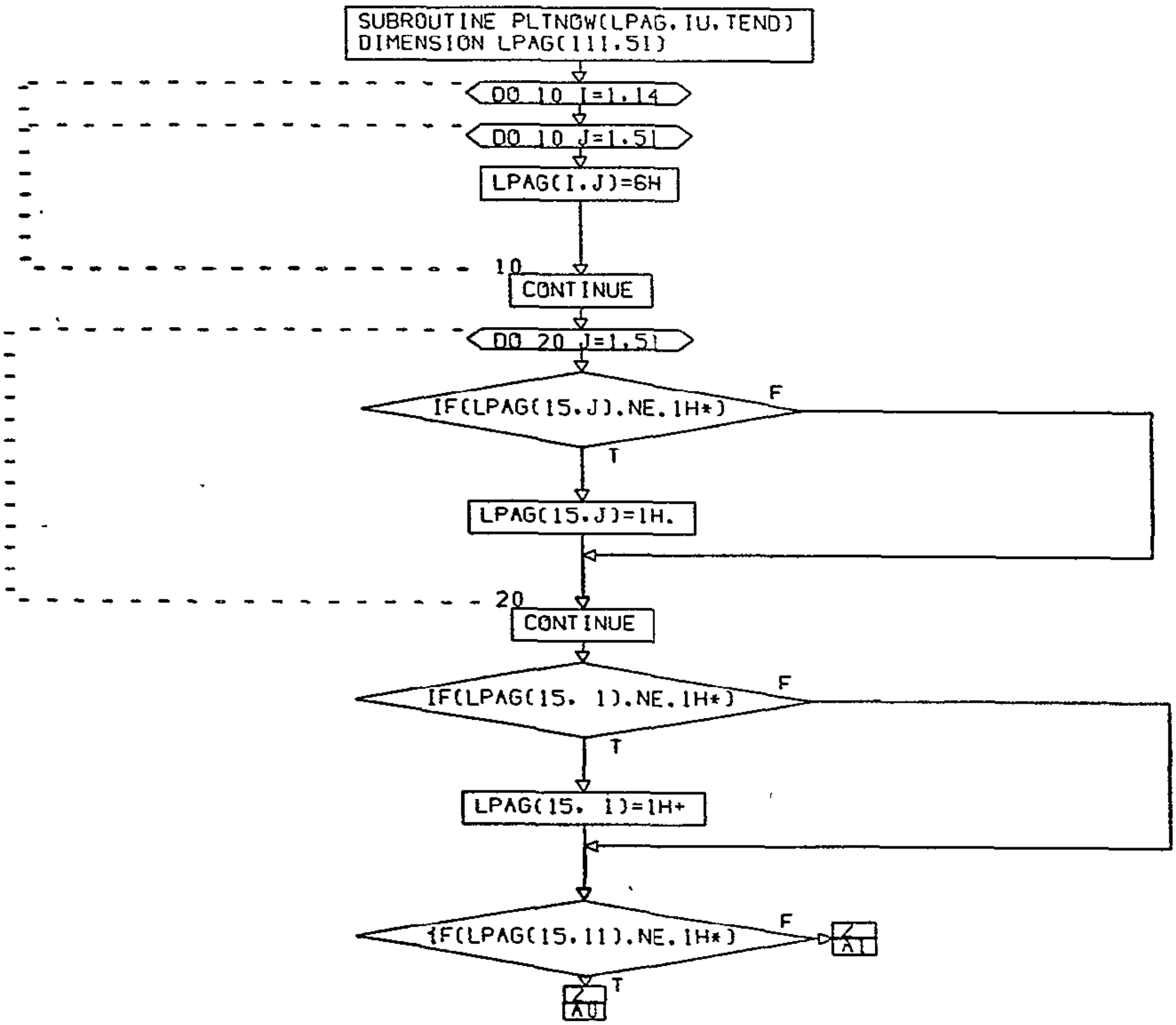
3.2.14 Subroutine: PLTNOW

PURPOSE: Write the plot of total source power versus time, as determined in Phase I, on the printer.

METHOD: Create headings necessary for the plot and write the plot on the printer.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.14. See Appendix for definition of all variables.

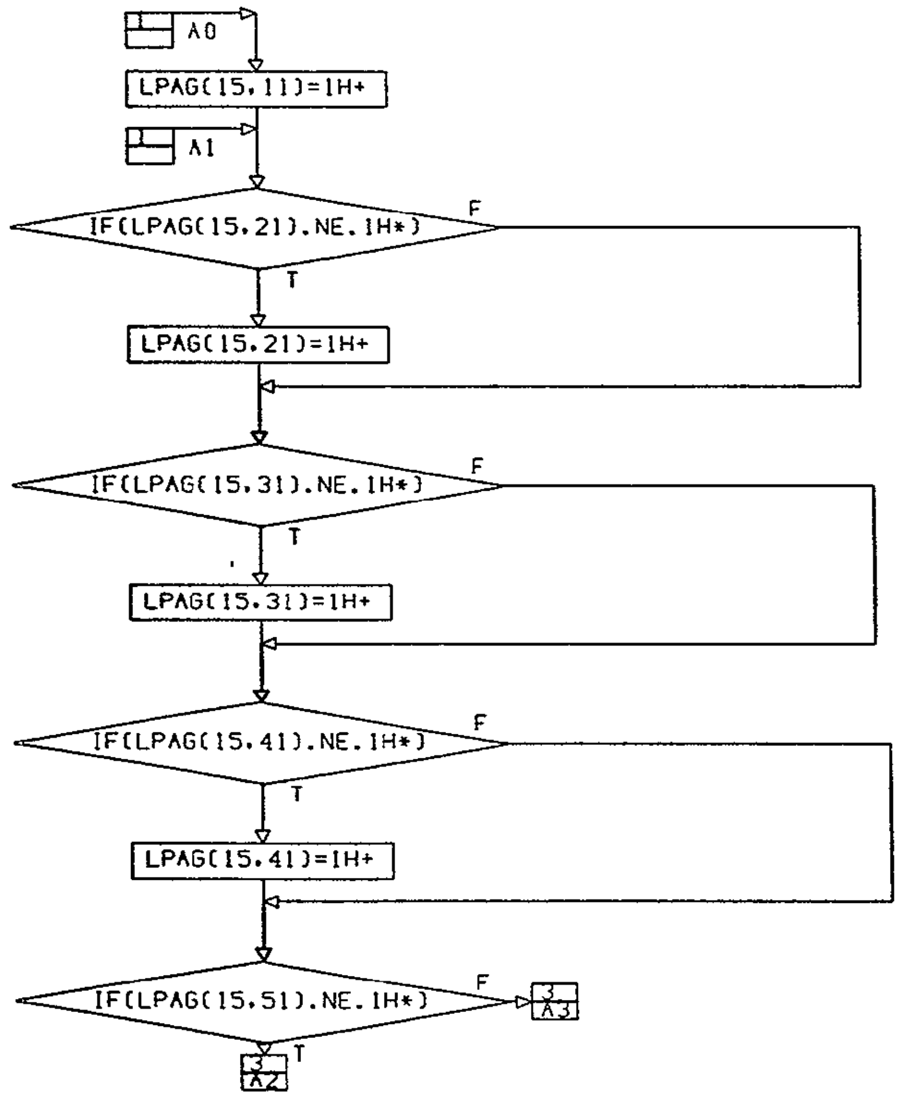
C.2



CONT. ON PG 2

PLTNOW
PG 1 OF 5

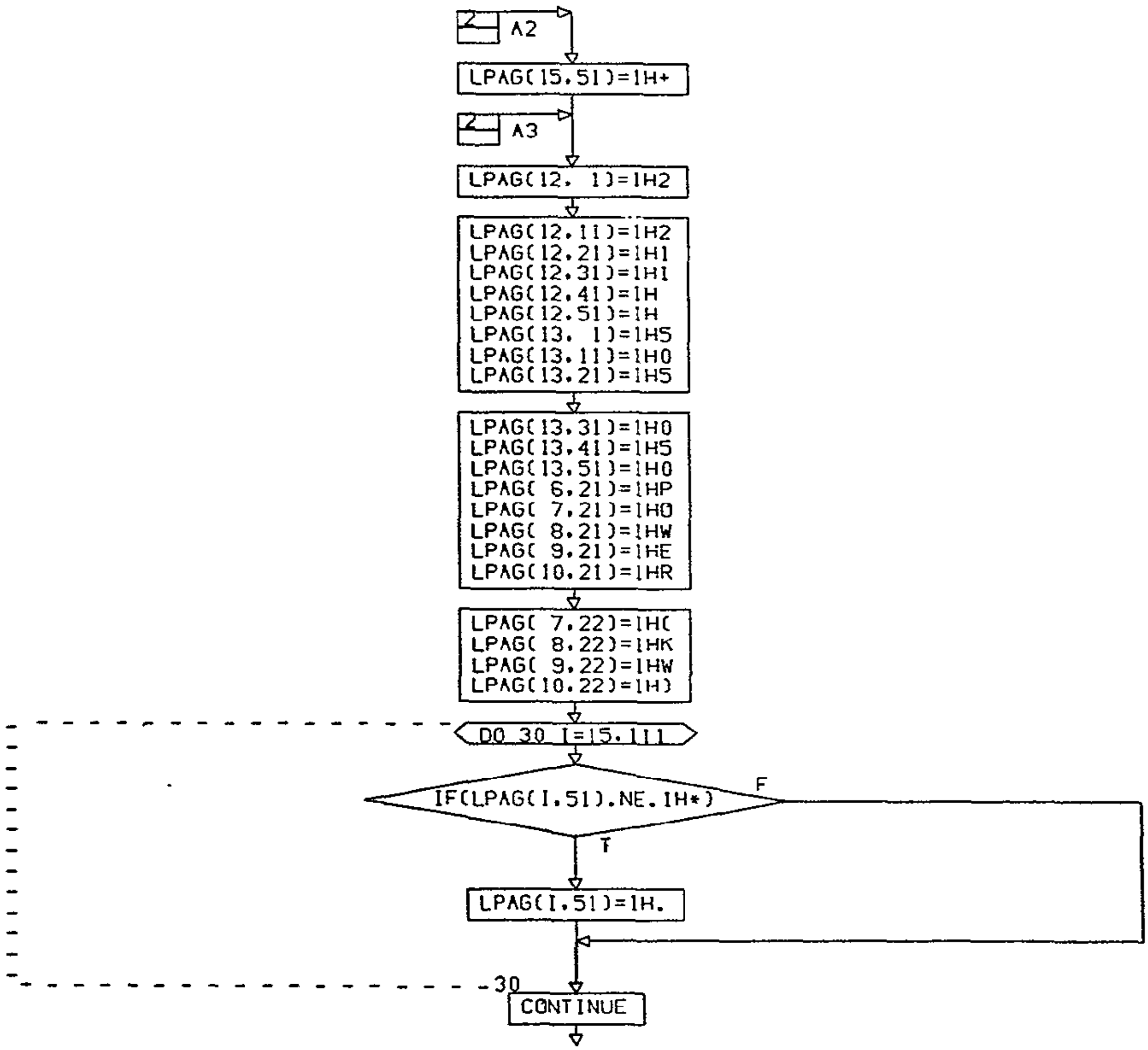
FIGURE 3.2.14. FUNCTIONAL FLOWCHART OF SUBROUTINE PLTNOW



CONT. ON PG 3

PLTNOW
PG 2 OF 5

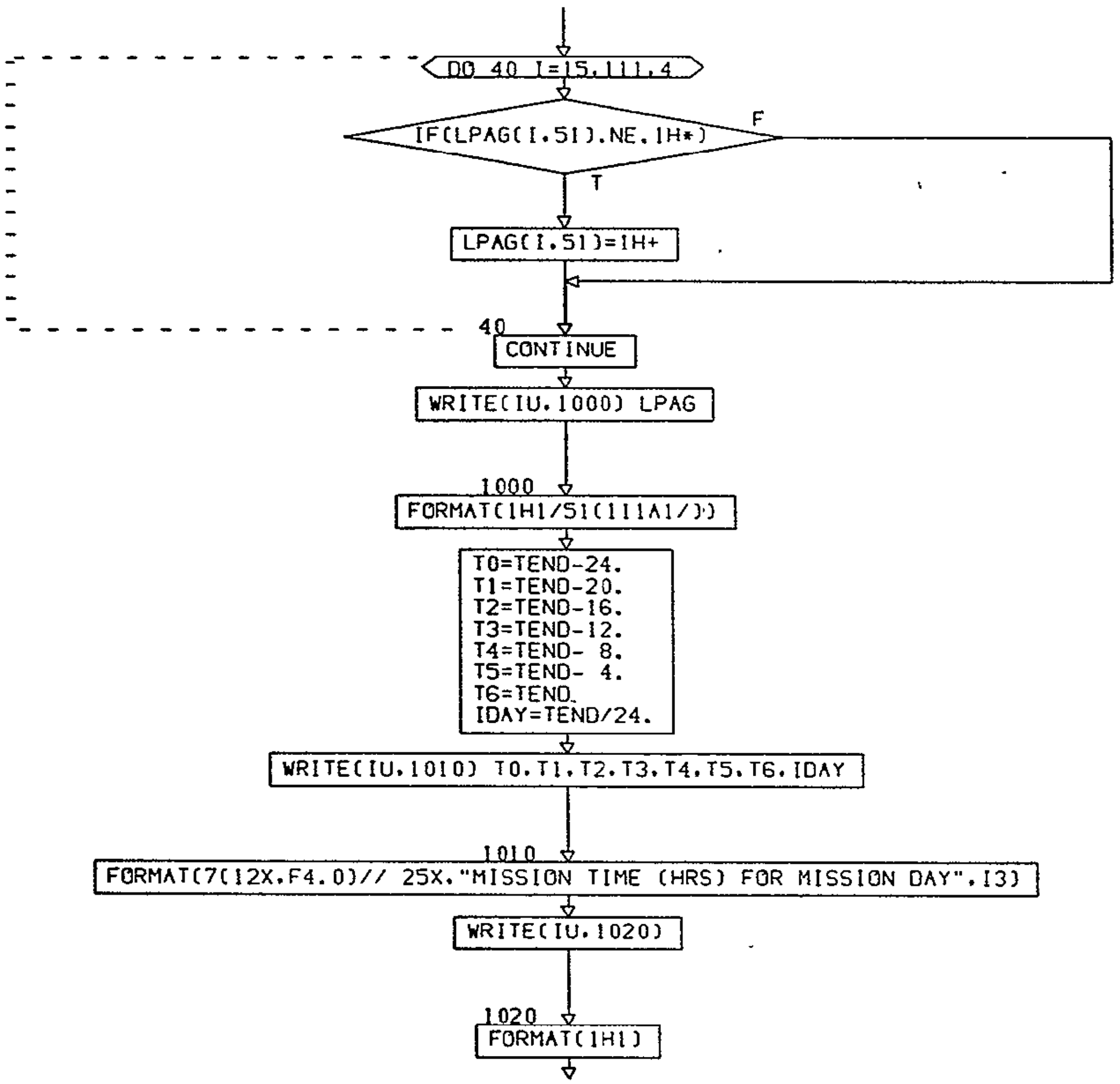
FIGURE 3.2.14. FUNCTIONAL FLOWCHART OF SUBROUTINE PLTNOW (CONTINUED)



CONT. ON PG 4

PLTNOW
PG 3 OF 5

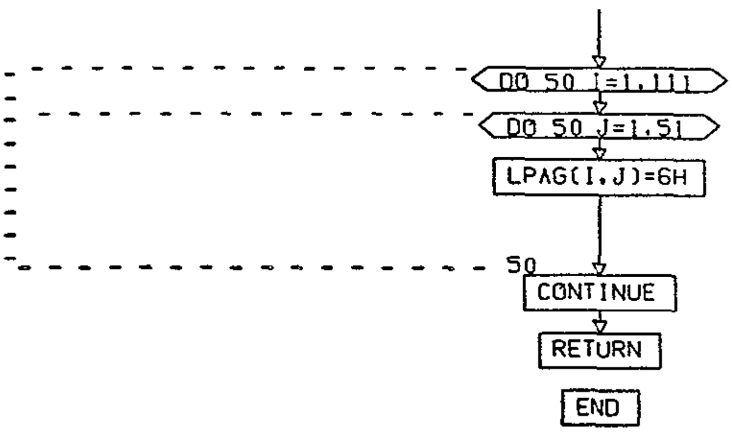
FIGURE 3.2.14. FUNCTIONAL FLOWCHART OF SUBROUTINE PLTNOW (CONTINUED)



CONT. ON PG 5

PLTNOW
PG 4 OF 5

FIGURE 3.2.14. FUNCTIONAL FLOWCHART OF SUBROUTINE PLTNOW (CONTINUED)



PLTNOW
PG 5 FINAL

FIGURE 3.2.14. FUNCTIONAL FLOWCHART OF SUBROUTINE PLTNOW (CONTINUED)

3.2.15 Subroutine: PRTPLT

PURPOSE: Create a plot of total source power versus time, as determined in Phase I.

METHOD: Determine the scaling necessary for this point. Store the point in the plot array. If the time elapsed has exceeded 24 hours, call PLTNOW to print the plot.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.15. See Appendix for definition of all variables.

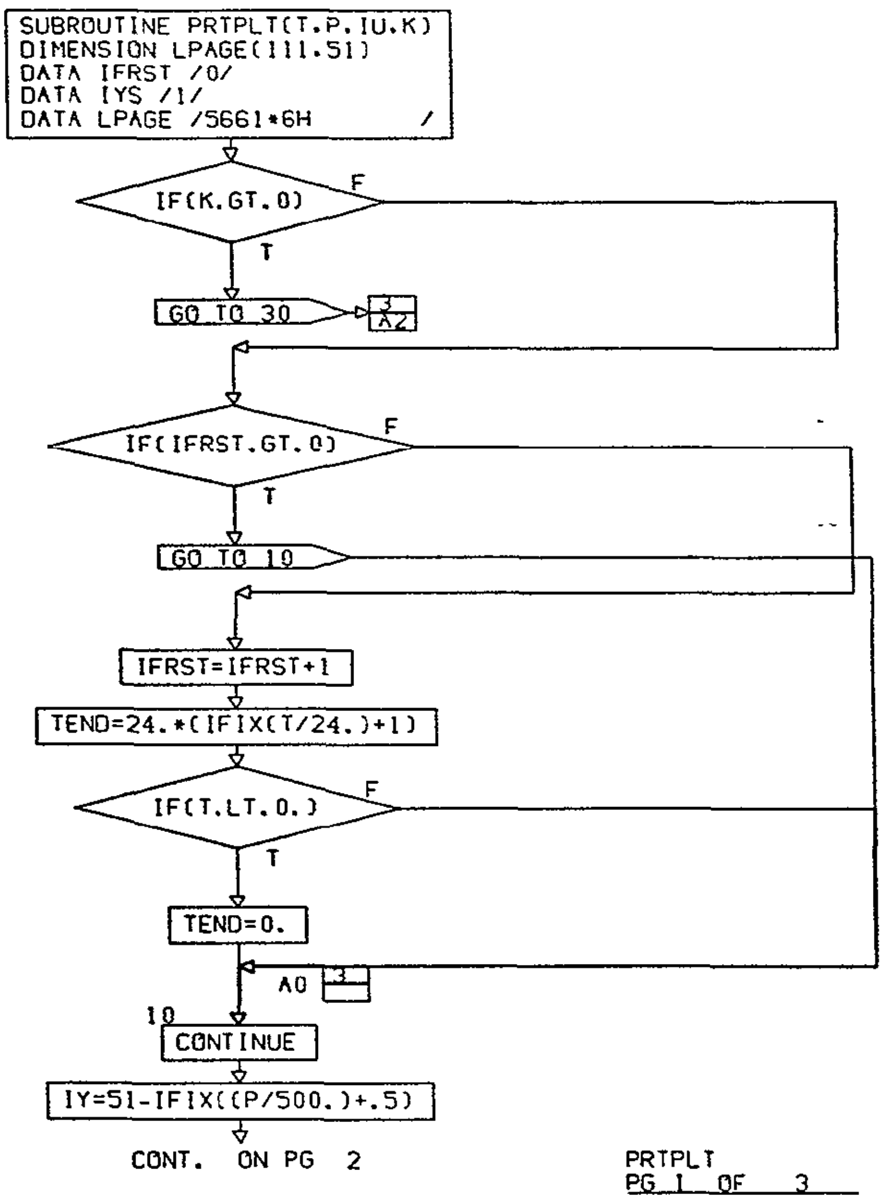


FIGURE 3.2.15. FUNCTIONAL FLOWCHART OF SUBROUTINE PRTPLT

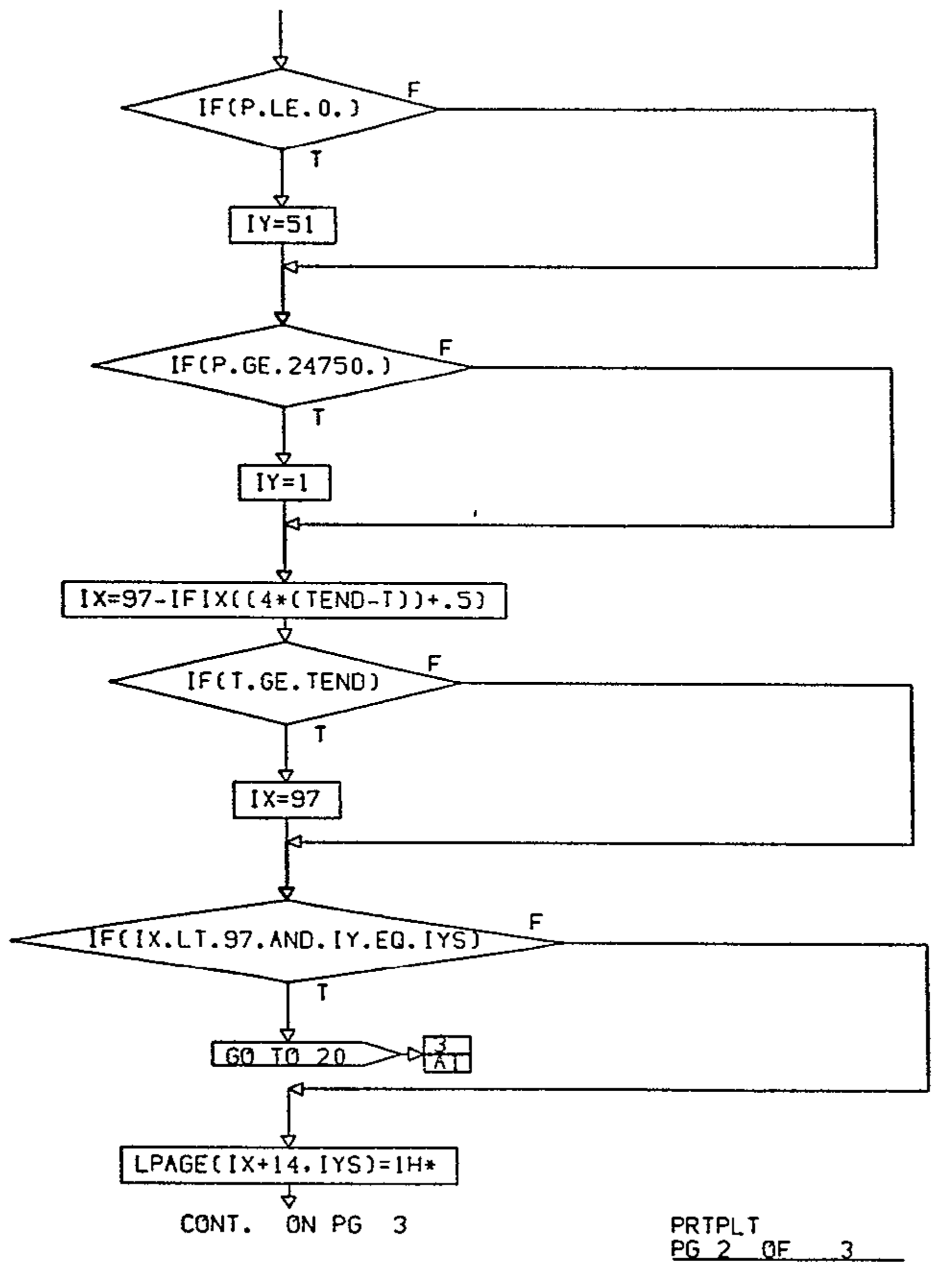


FIGURE 3.2.15. FUNCTIONAL FLOWCHART OF SUBROUTINE PRTPLT (CONTINUED)

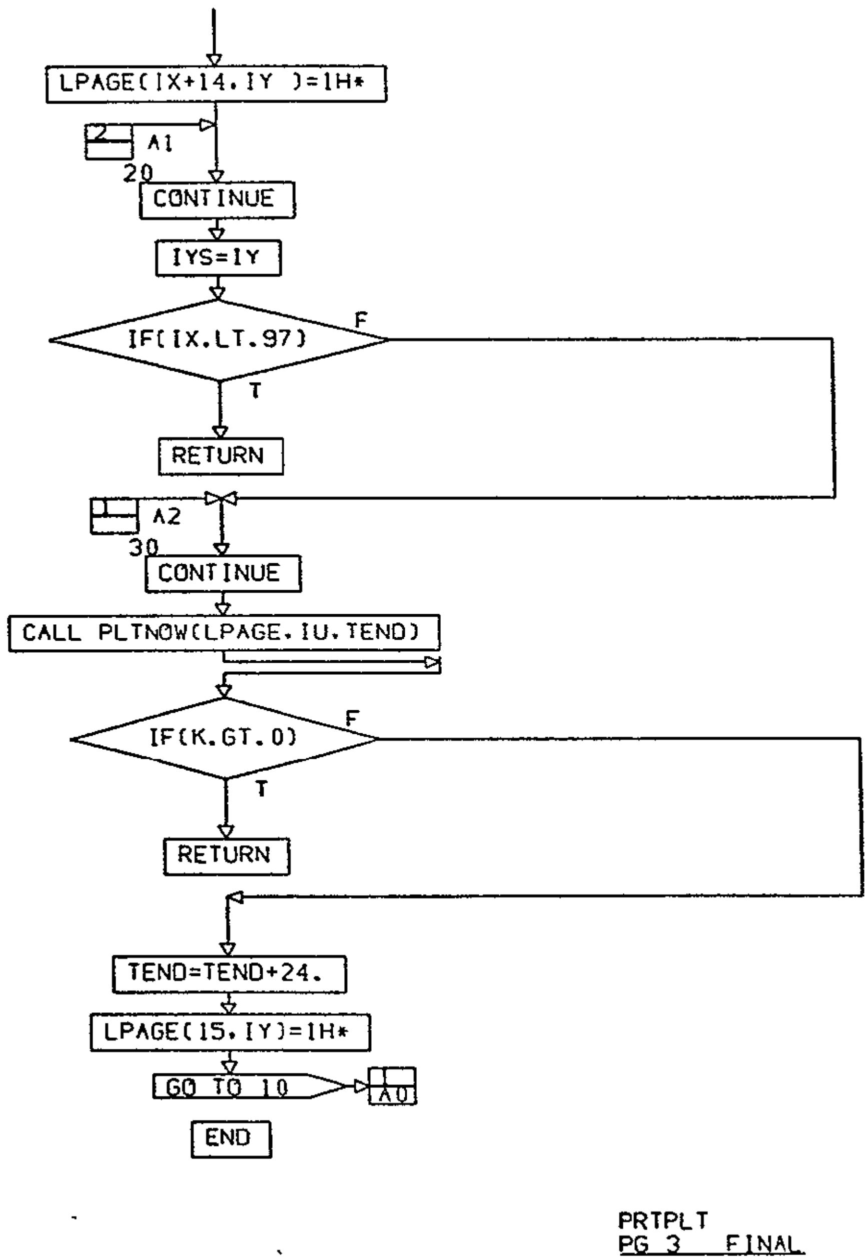


FIGURE 3.2.15. FUNCTIONAL FLOWCHART OF SUBROUTINE PRTPLT (CONTINUED)

3.2.16 Subroutine: RCYCLE

PURPOSE: To control all cyclic elements.

METHOD: Periodically this routine is called to update the cyclic's condition by calling either AHANDL, PHANDL, or CHANDL depending upon the type of cyclic element.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.16. See Appendix for definition of all variables

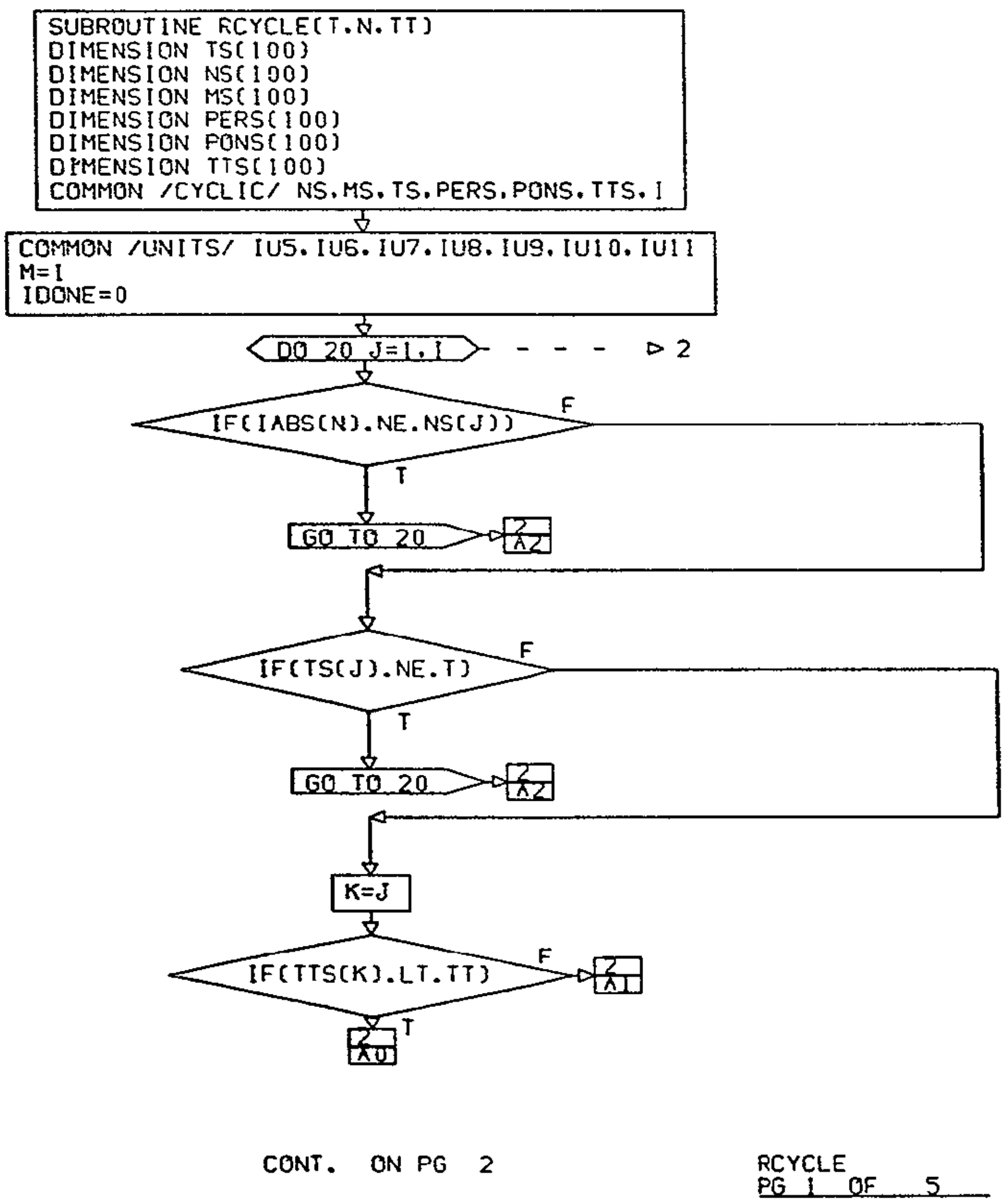


FIGURE 3.2.16. FUNCTIONAL FLOWCHART OF SUBROUTINE RCYCLE

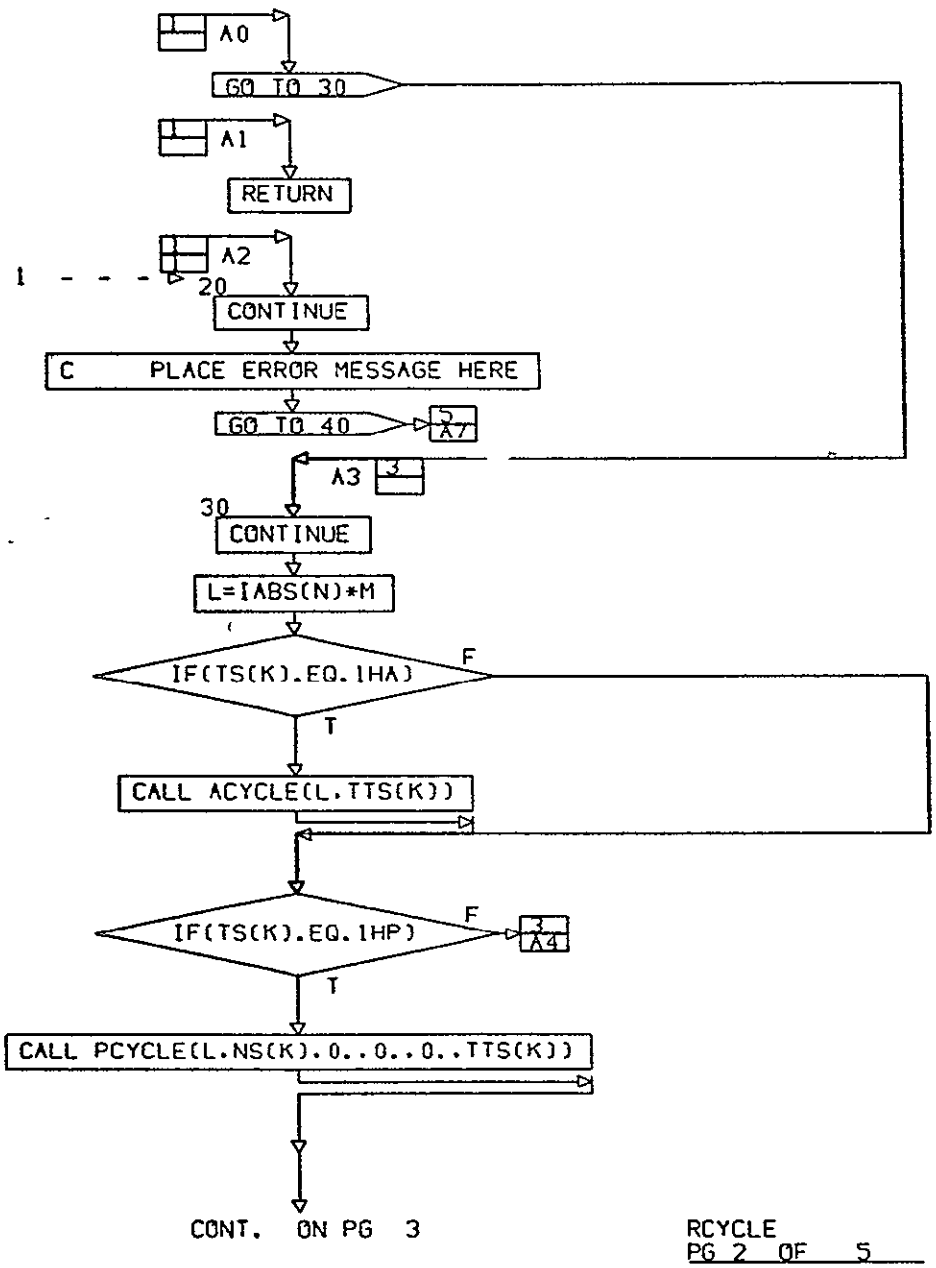
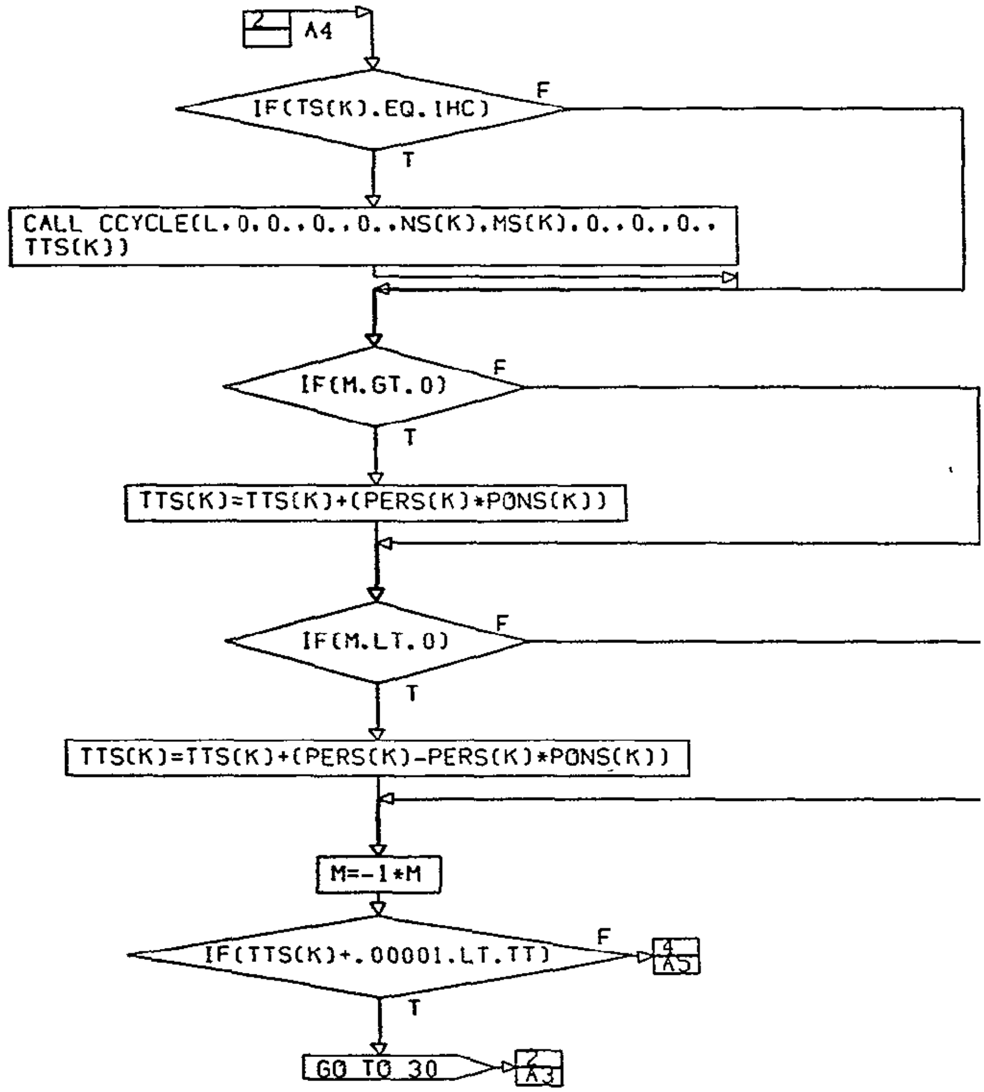


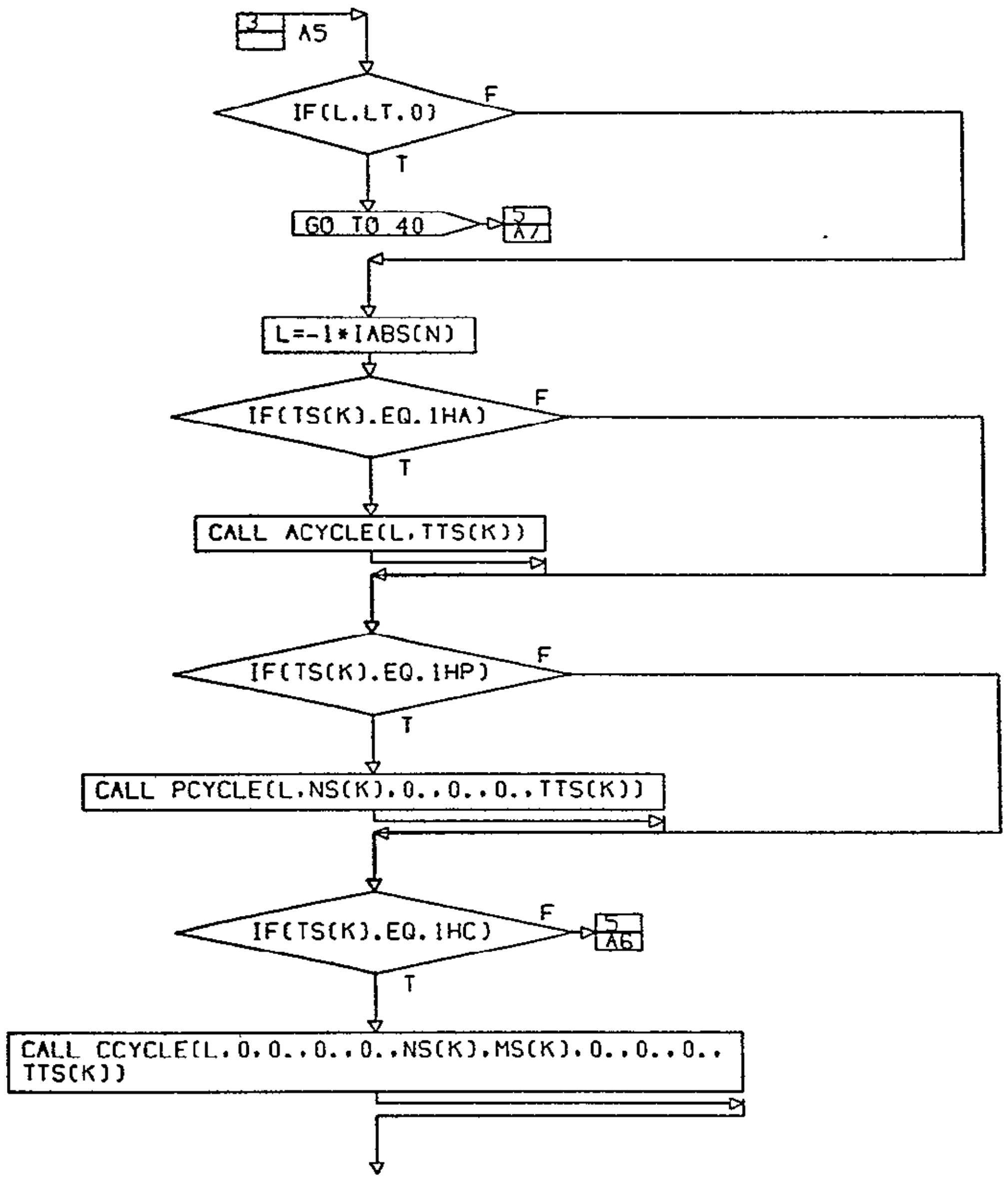
FIGURE 3.2.16. FUNCTIONAL FLOWCHART OF SUBROUTINE RCYCLE (CONTINUED)



CONT. ON PG 4

RCYCLE
PG 3 OF 5

FIGURE 3.2.16. FUNCTIONAL FLOWCHART OF SUBROUTINE RCYCLE (CONTINUED



CONT. ON PG 5

RCYCLE
PG 4 OF 5

FIGURE 3.2.16. FUNCTIONAL FLOWCHART OF SUBROUTINE RCYCLE (CONTINUED)

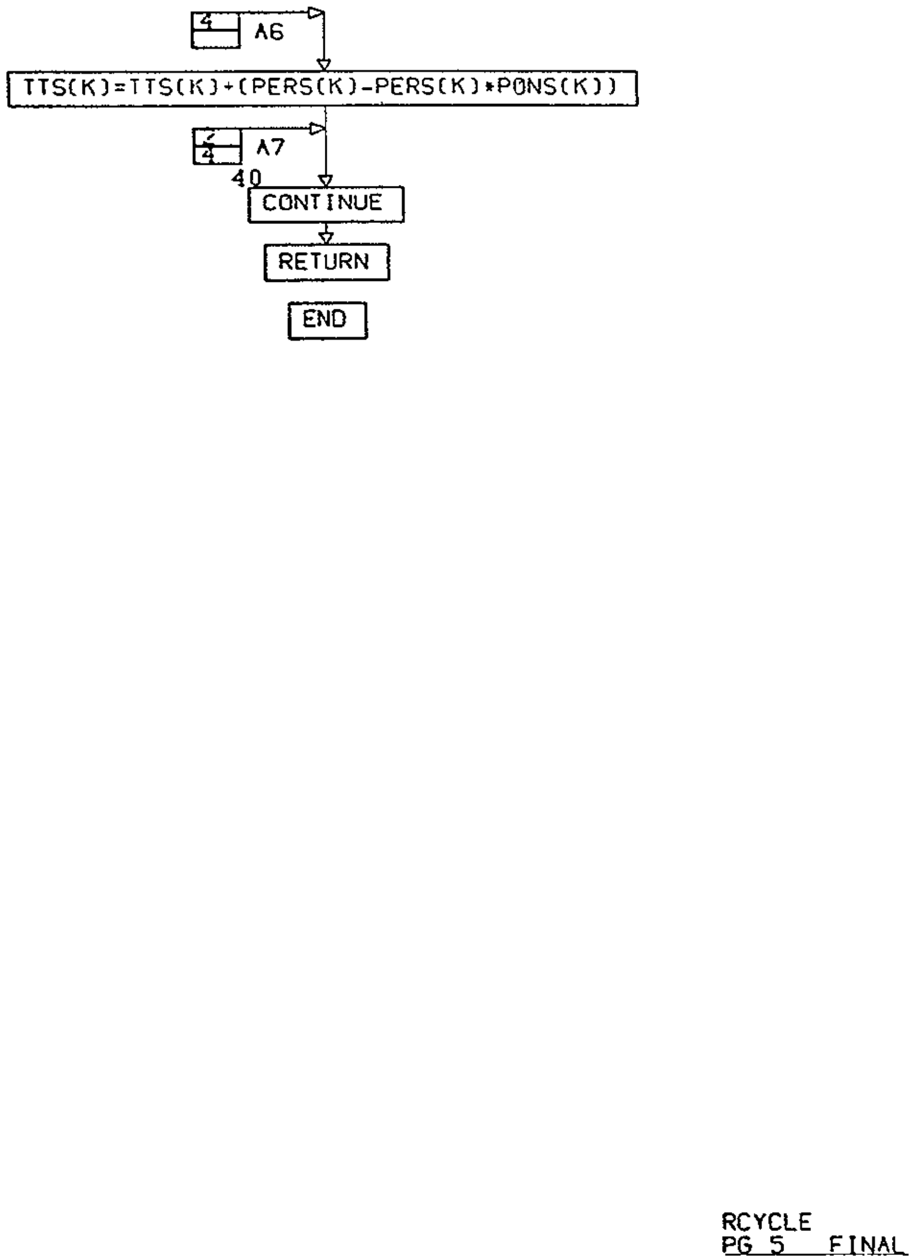


FIGURE 3.2.16. FUNCTIONAL FLOWCHART OF SUBROUTINE RCYCLE (CONTINUED)

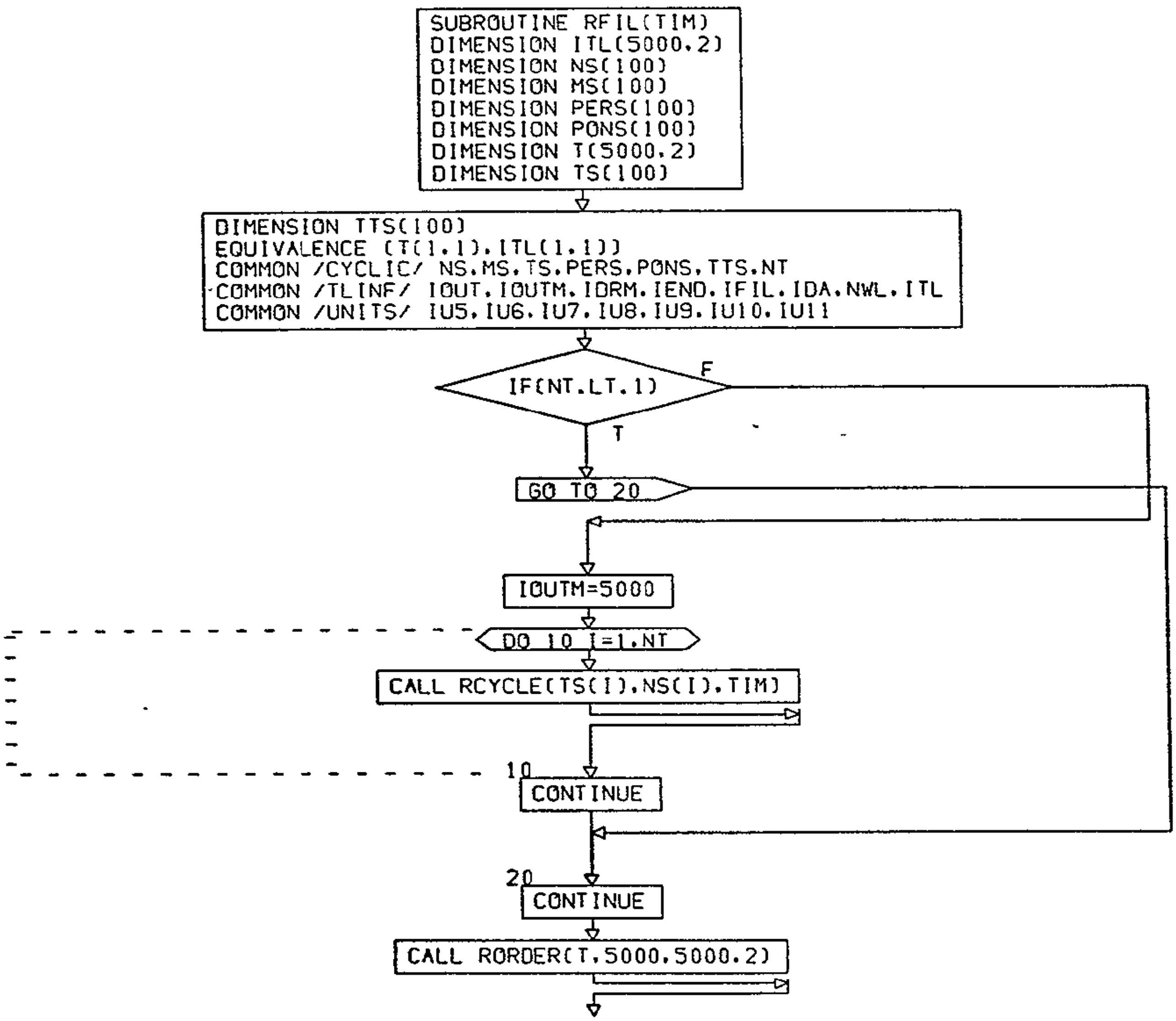
ORIGINAL PAGE IS
OF POOR QUALITY

3.2.17 Subroutine: RFIL

PURPOSE: To handle the situation created when the event array overloads.

METHOD: If necessary, all cyclic elements status are brought up to the present time. The event array is ordered on time. The number of points to be written to the event timeline unit (0) is determined. The points are written to the tape and removed from the event array.

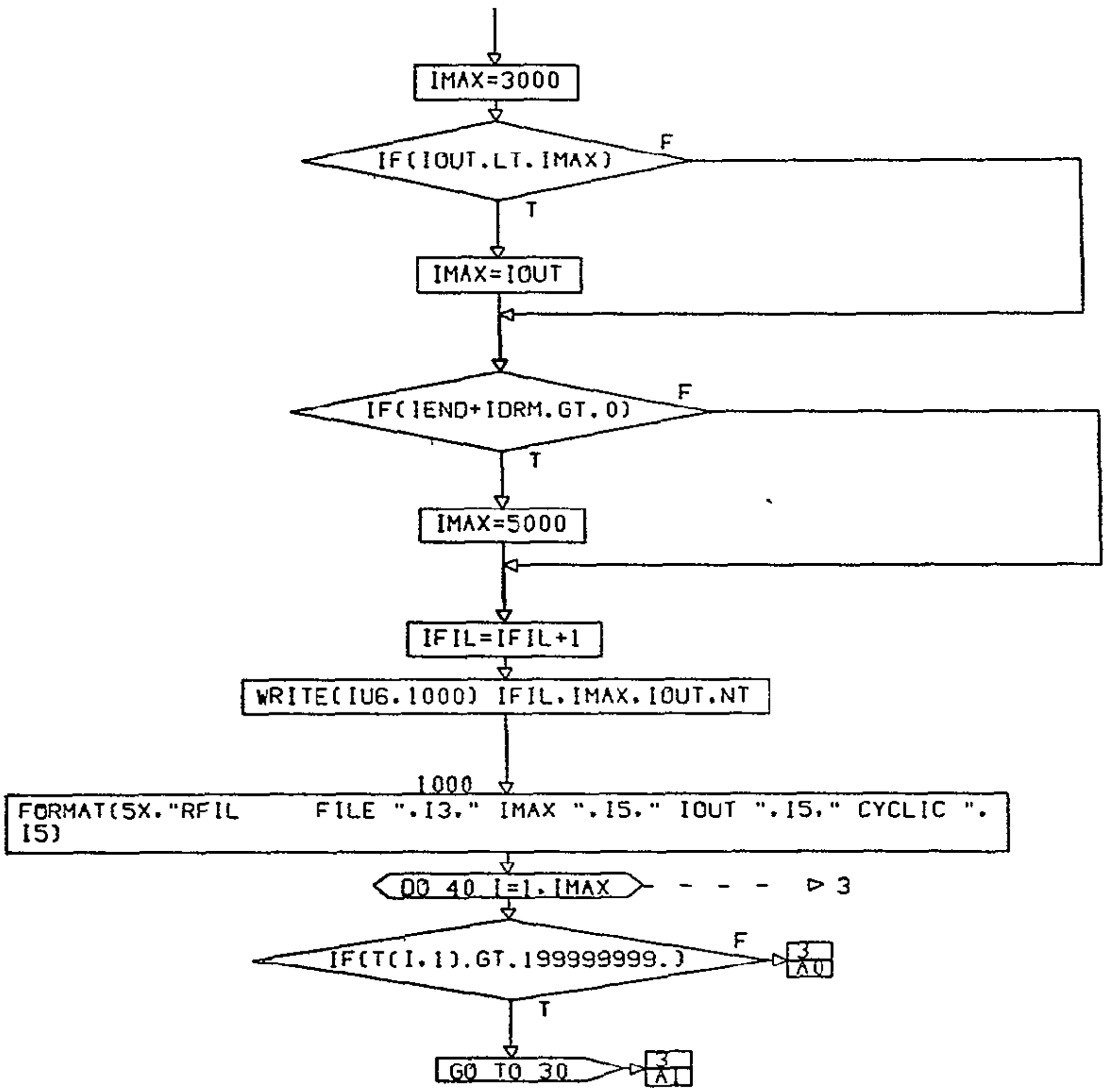
VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.17. See Appendix for definition of all variables.



CONT. ON PG 2

RFIL
PG 1 OF 3

FIGURE 3.2.17. FUNCTIONAL FLOWCHART OF SUBROUTINE RFIL



CONT. ON PG 3

RFIL
PG 2 OF 3

FIGURE 3.2.17. FUNCTIONAL FLOWCHART OF SUBROUTINE RFIL (CONTINUED)

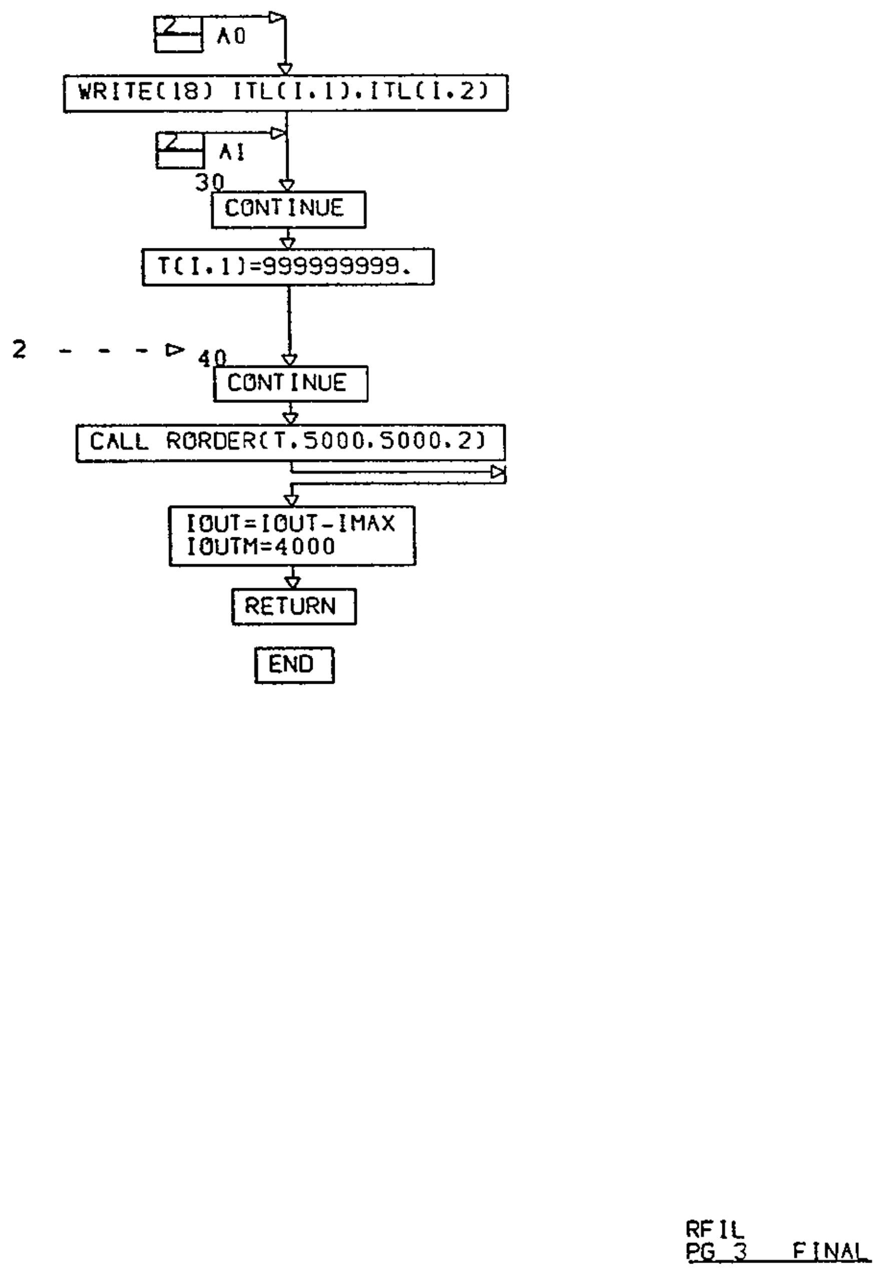


FIGURE 3.2.17. FUNCTIONAL FLOWCHART OF SUBROUTINE RFIL (CONTINUED)

3.2.18 Subroutine: SHANDL

PURPOSE: This routine creates the switch portion of the event timeline.

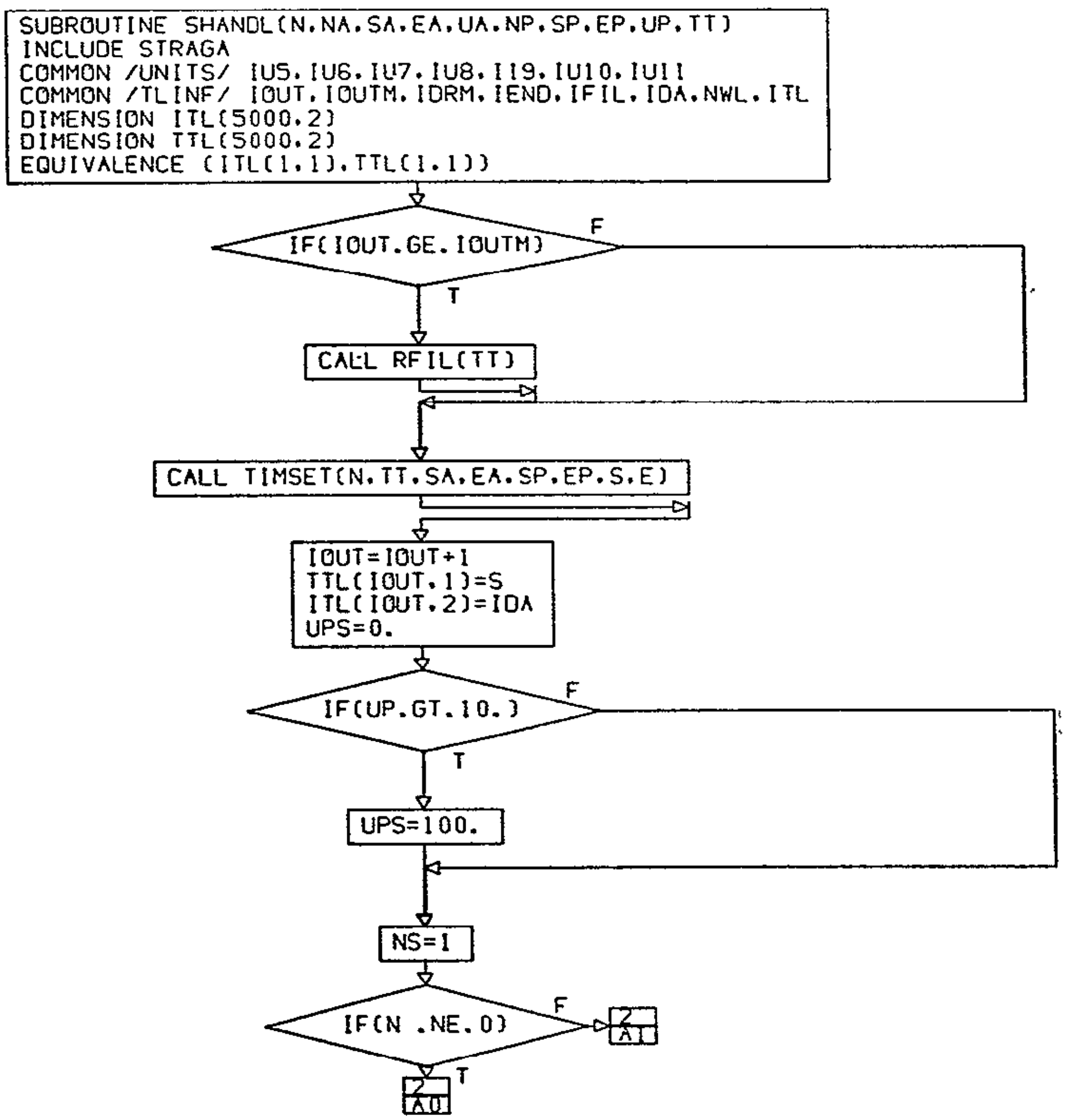
METHOD: For each switch the following are determined:

1. Determines the event on and off time

2. Stores the event in the timeline array

3. Writes the event on drum

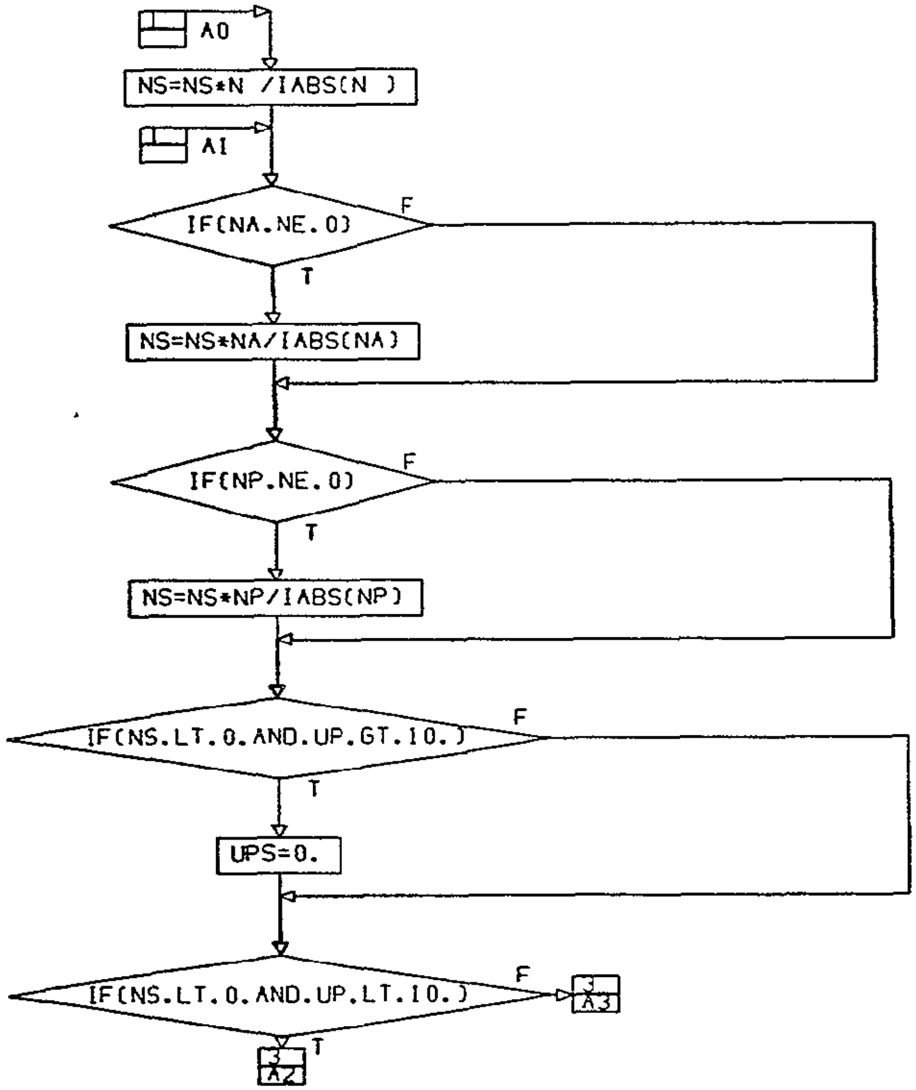
VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.18. See Appendix for definition of all variables.



CONT. ON PG 2

SHANDL
PG 1 OF 5

FIGURE 3.2.18. FUNCTIONAL FLOWCHART OF SUBROUTINE SHANDL



CONT. ON PG 3

SHANDL
PG 2 OF 5

FIGURE 3.2.18. FUNCTIONAL FLOWCHART OF SUBROUTINE SHANDL (CONTINUED)

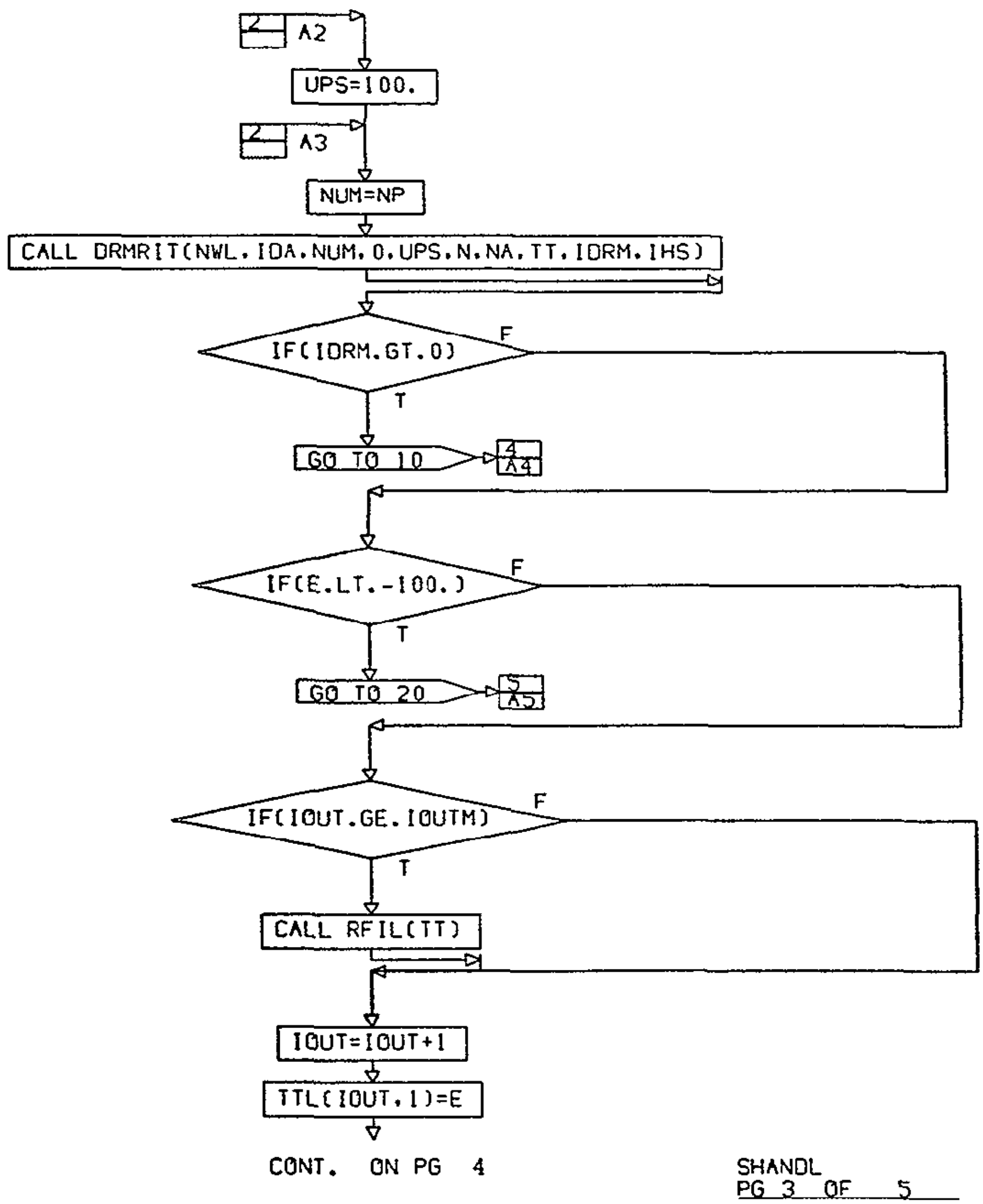


FIGURE 3.2.18. FUNCTIONAL FLOWCHART OF SUBROUTINE SHANDL (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

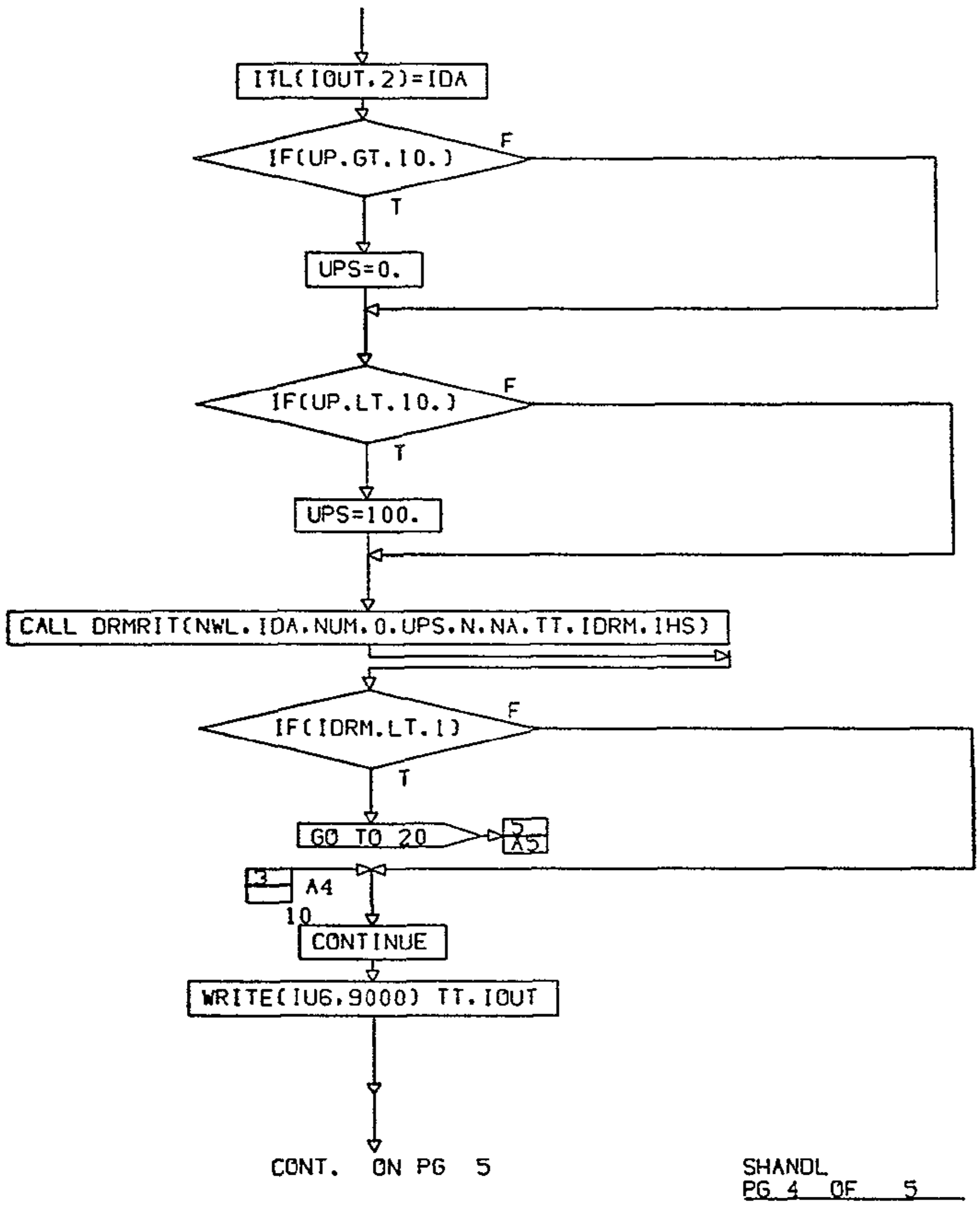
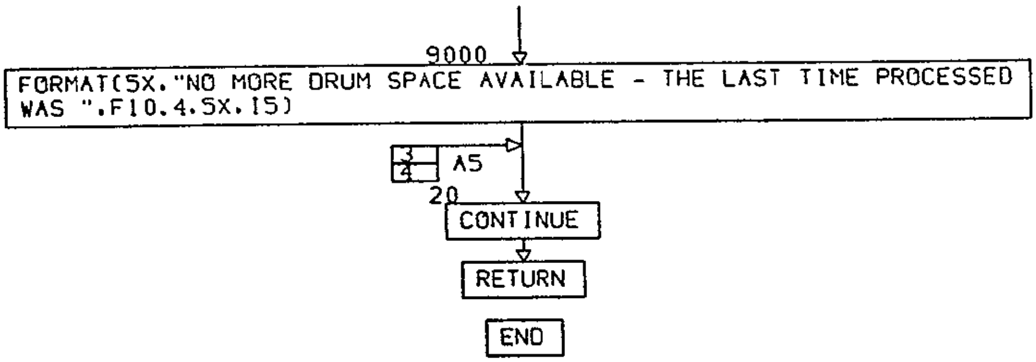


FIGURE 3.2.18. FUNCTIONAL FLOWCHART OF SUBROUTINE SHANDL (CONTINUED)



SHANDL
PG 5 FINAL

FIGURE 3.2.18. FUNCTIONAL FLOWCHART OF SUBROUTINE SHANDL (CONTINUED)

3.2.19 Subroutine: TPOUTJ

PURPOSE: Analyze the event timeline and provide the interface tape, plots, and all formatted printouts from Phase I.

METHOD: The event timeline is read to determine the change of status for switches and components. If the change of status effects a component certain checks are made to insure the component is operating at its highest use factor. Changes in status happening within an input delta time of each other are grouped together and printed and plotted as if they occurred at the same time.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.19. See Appendix for definition of all variables.

```
SUBROUTINE TPOUTJ
INCLUDE STRAG1
INTEGER CDAT(50)
INTEGER TDAT(8)
INTEGER TDATA(8)
INTEGER QQASG
DIMENSION PWR(3)
DIMENSION TRILD( 500.3)
```

```
DIMENSION TRILDE(3)
DIMENSION TRILDP(3)
DIMENSION BUS(24)
DIMENSION PFB(12)
DIMENSION ACLOAD(12)
DIMENSION IS(100)
DIMENSION TLOAD(100)
DIMENSION ITLINE(3000.3)
```

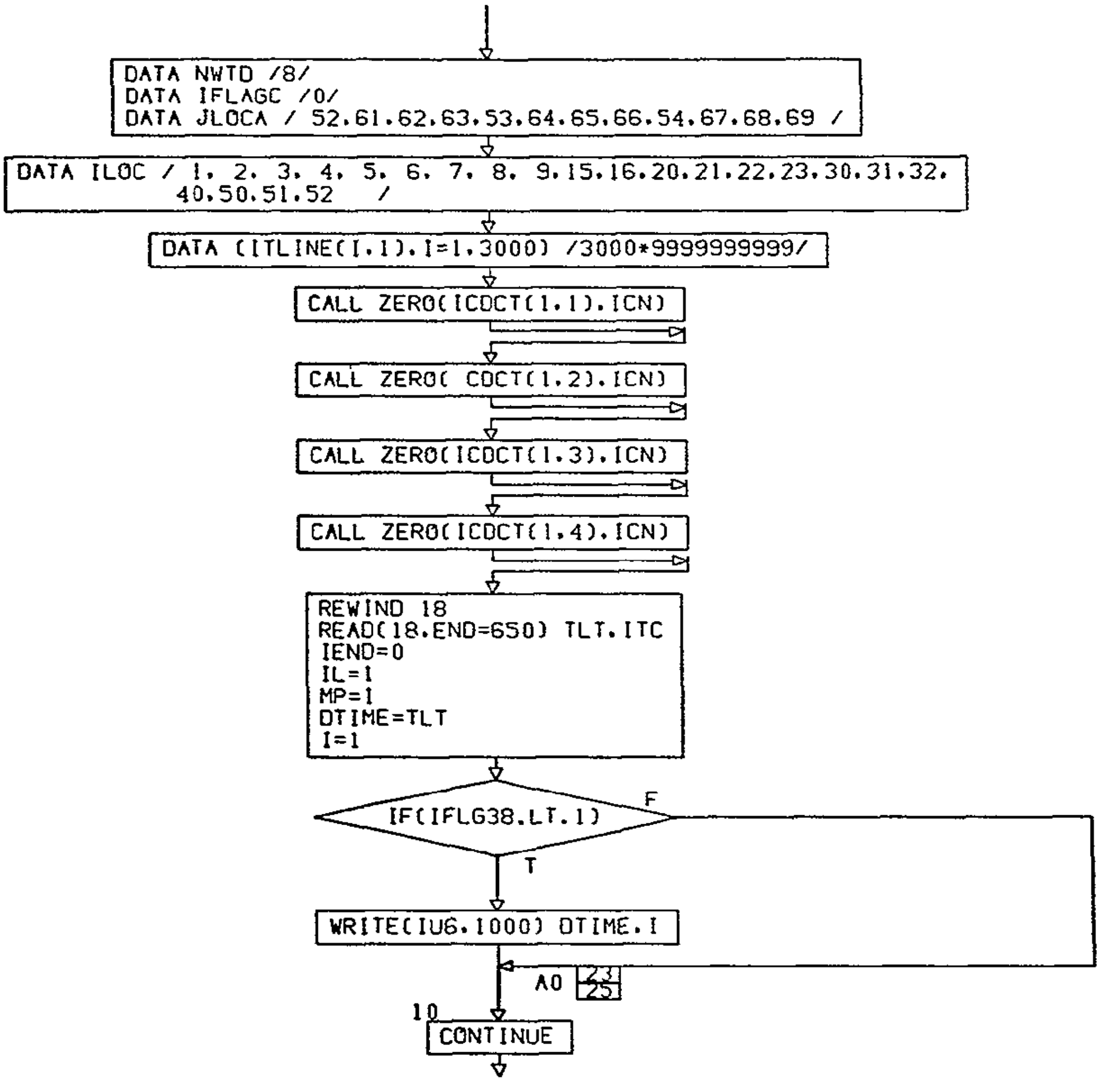
```
DIMENSION CDCT (500.4)
DIMENSION ICDCT(500.4)
DIMENSION NAM(6)
DIMENSION PWRM(4.10)
DIMENSION SUBSYS(100)
DIMENSION ILOC(22)
DIMENSION JLOCA(12)
COMMON /THREE/ ID.ILNCP.NAM.PF.USF.PWRM
```

```
COMMON /FOUR / NUM .MD .UST .IAC .IPC .ISF .TC .TYP
COMMON /FIVE / NUMA.MDA.USTA.IACA.IPCA.ISFA.TCA.TYPA
COMMON /MISPHS/ STORMP
COMMON /ACPOWF/ PFEFF.RESLOS
COMMON /ALTERN/ ICDCB(750).ICDC(500.3).ICN
COMMON /BSLOC/ IBUSC(500).ISYSC(500)
COMMON /PHAS1/ IDUM(3503).TM(50).MM.TITLM(6.50).TDELT1.TABORT
EQUIVALENCE (CDCT(1.1).ICDCT(1.1))
```

```
EQUIVALENCE (CDAT(1).ID)
EQUIVALENCE (TDAT(1).NUM)
EQUIVALENCE (TDATA(1).NUMA)
EQUIVALENCE ( IU.IUNIT( 8))
EQUIVALENCE (IUP.IUNIT( 9))
DATA PFEFF / 1.2/
DATA RESLOS / 1.04/
DATA NWAD /50/
```

CONT. ON PG 2

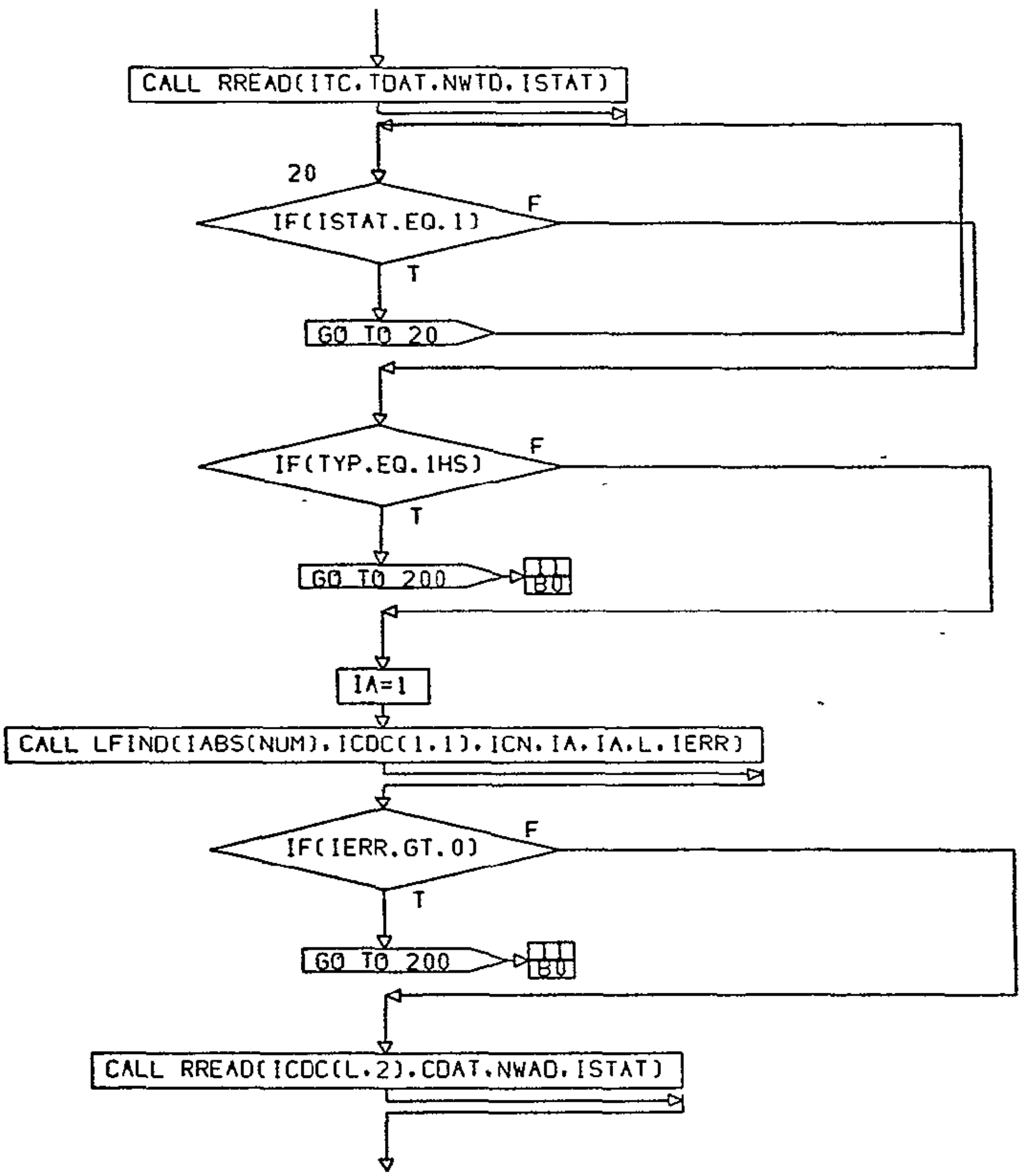
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ



CONT. ON PG 3

TPOUTJ
PG 2 OF 32

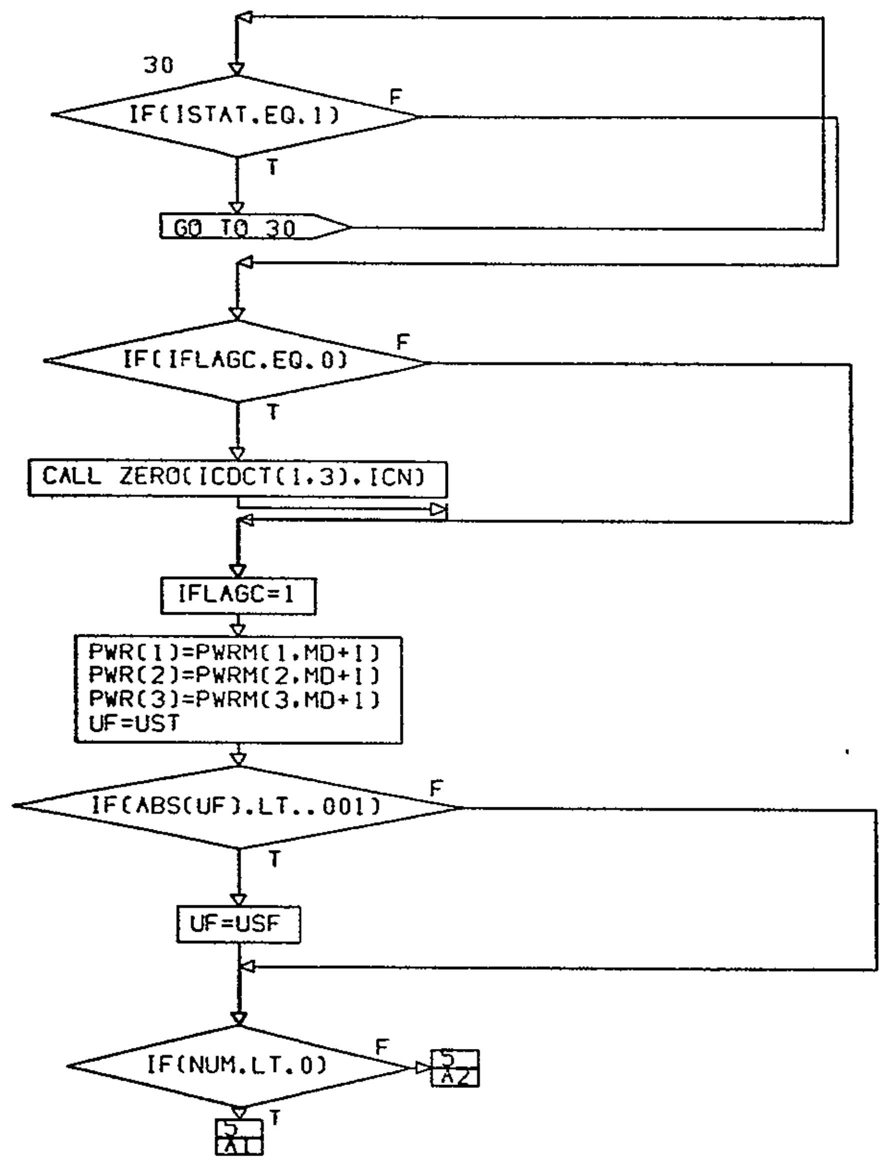
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 4

TPOUTJ
PG 3 OF 32

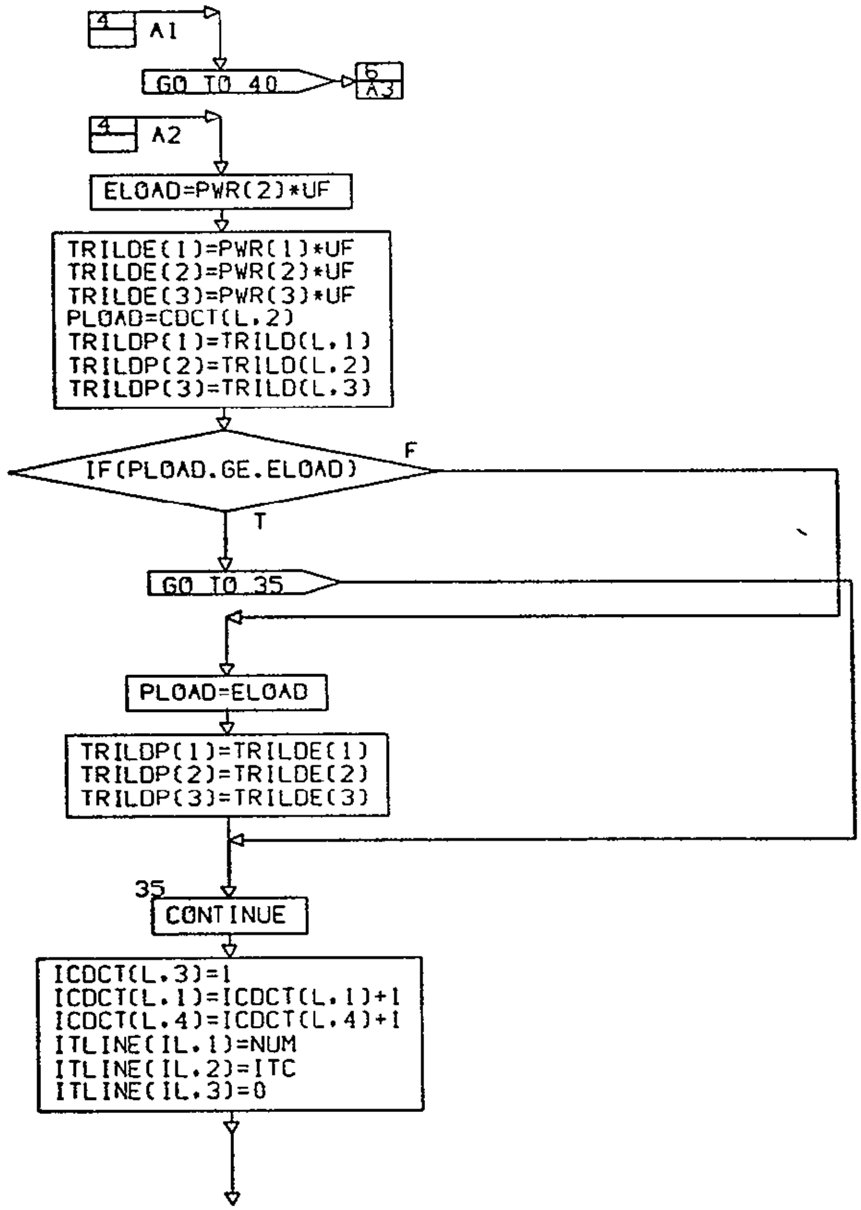
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 5

TPOUTJ
PG 4 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 6

TPOUTJ
PG 5 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

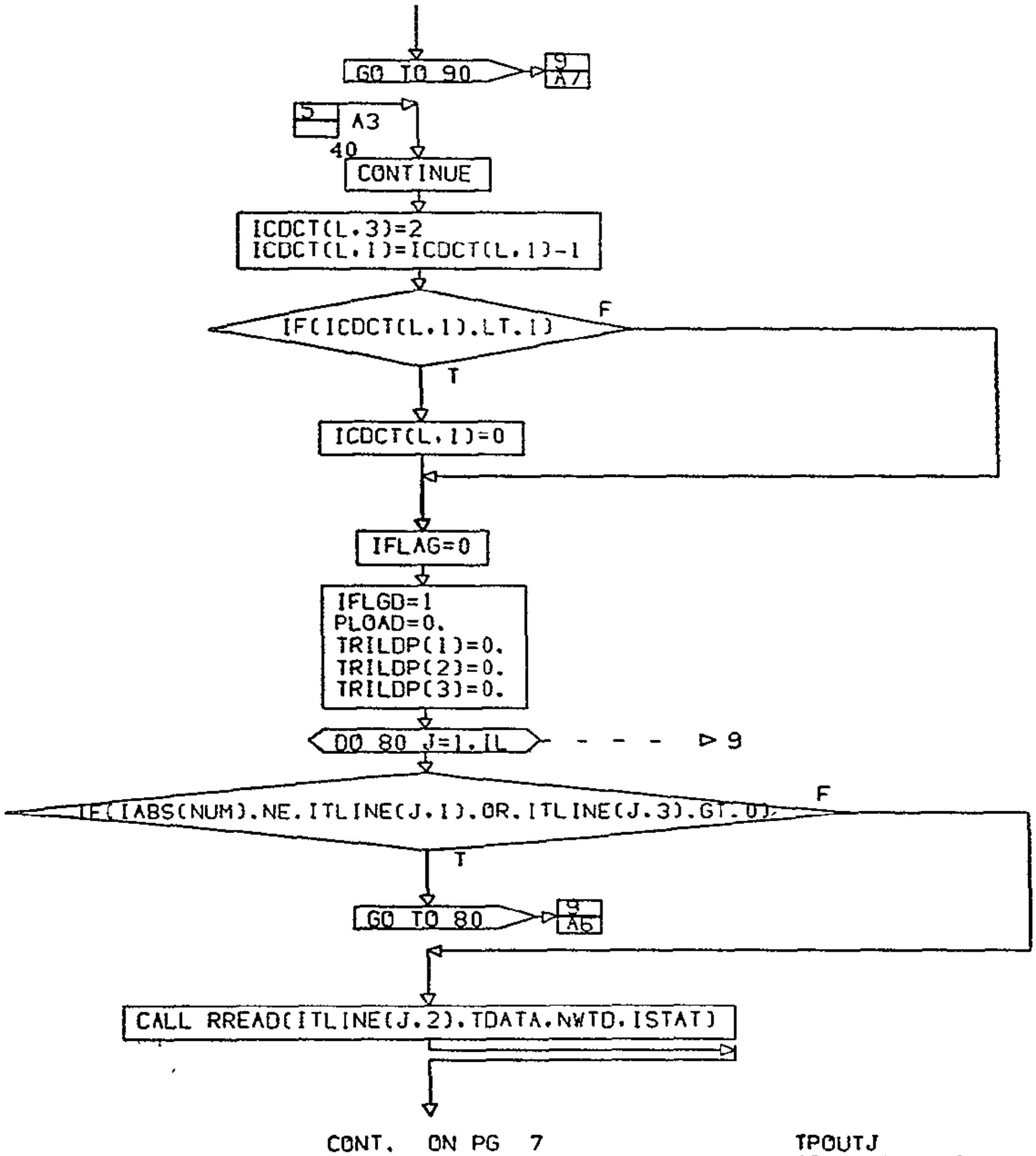
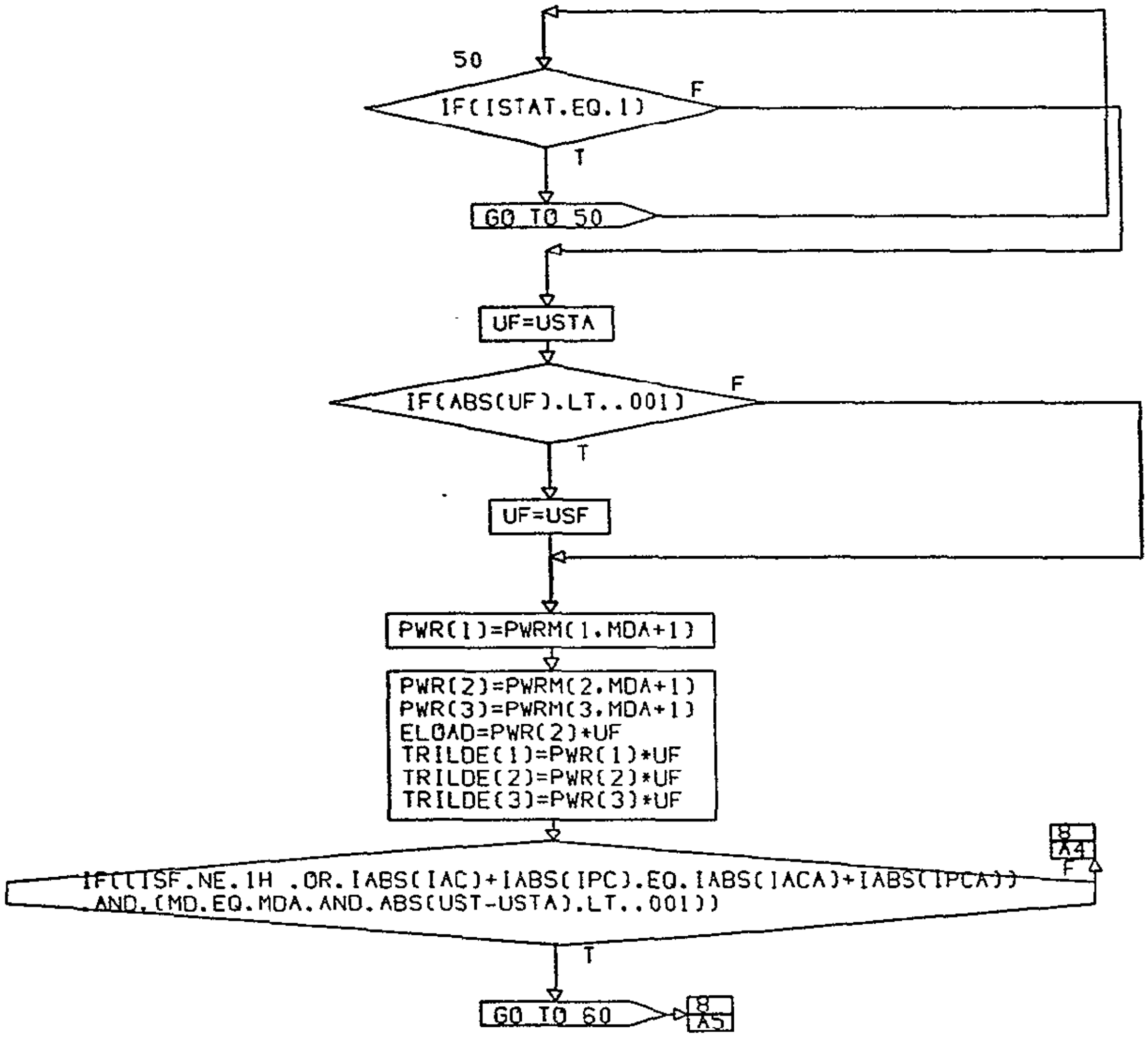


FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 8

TPOUTJ
PG 7 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

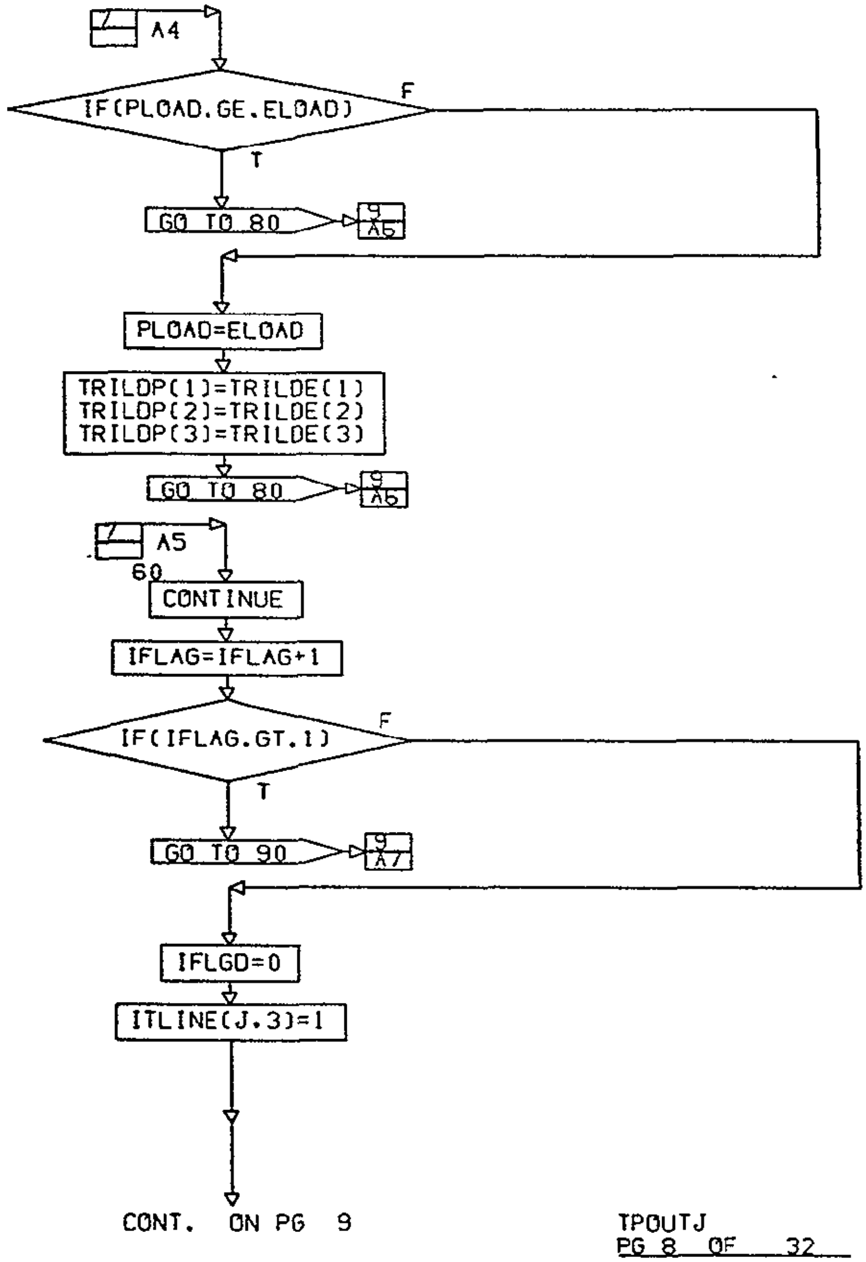


FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

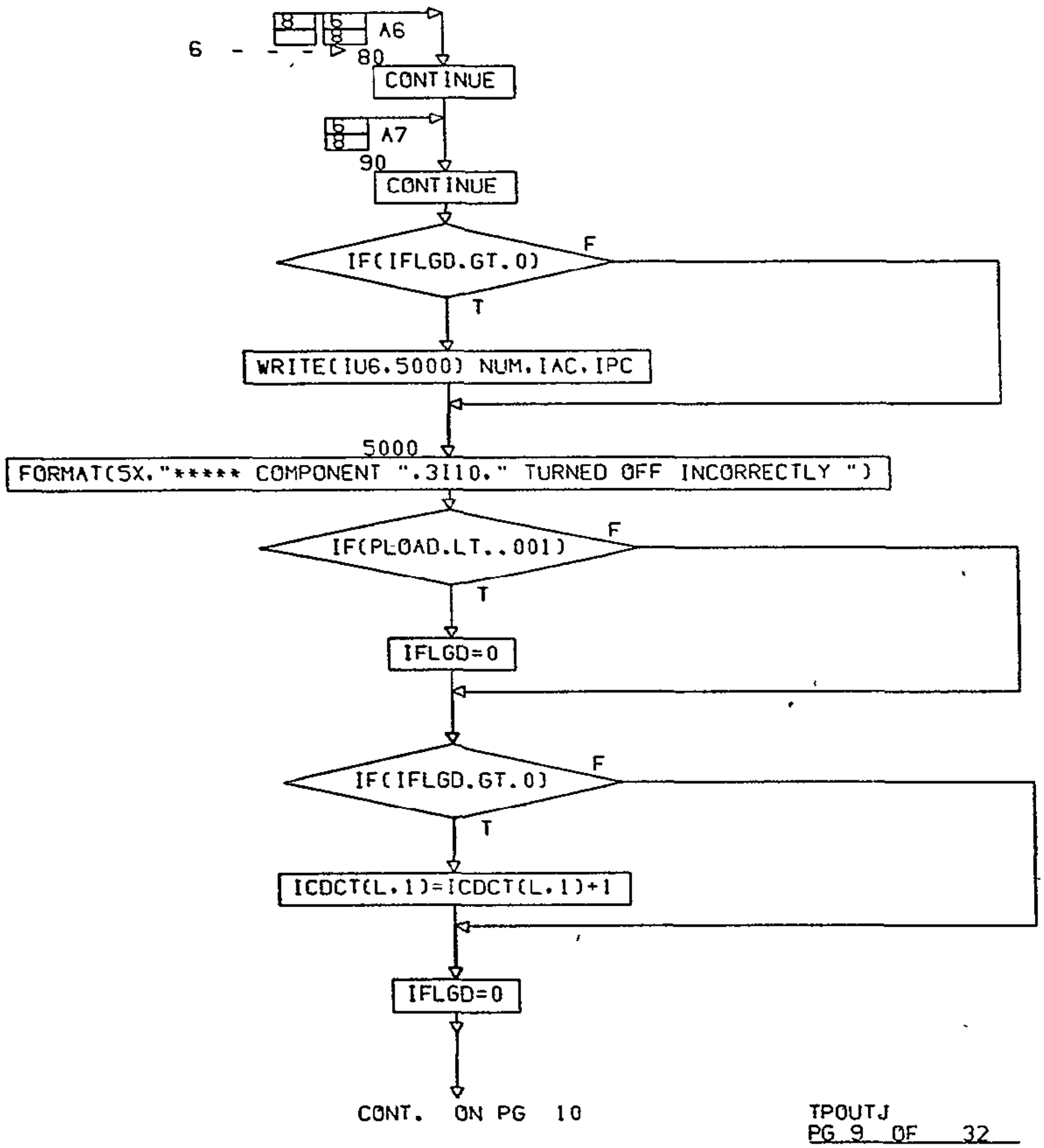
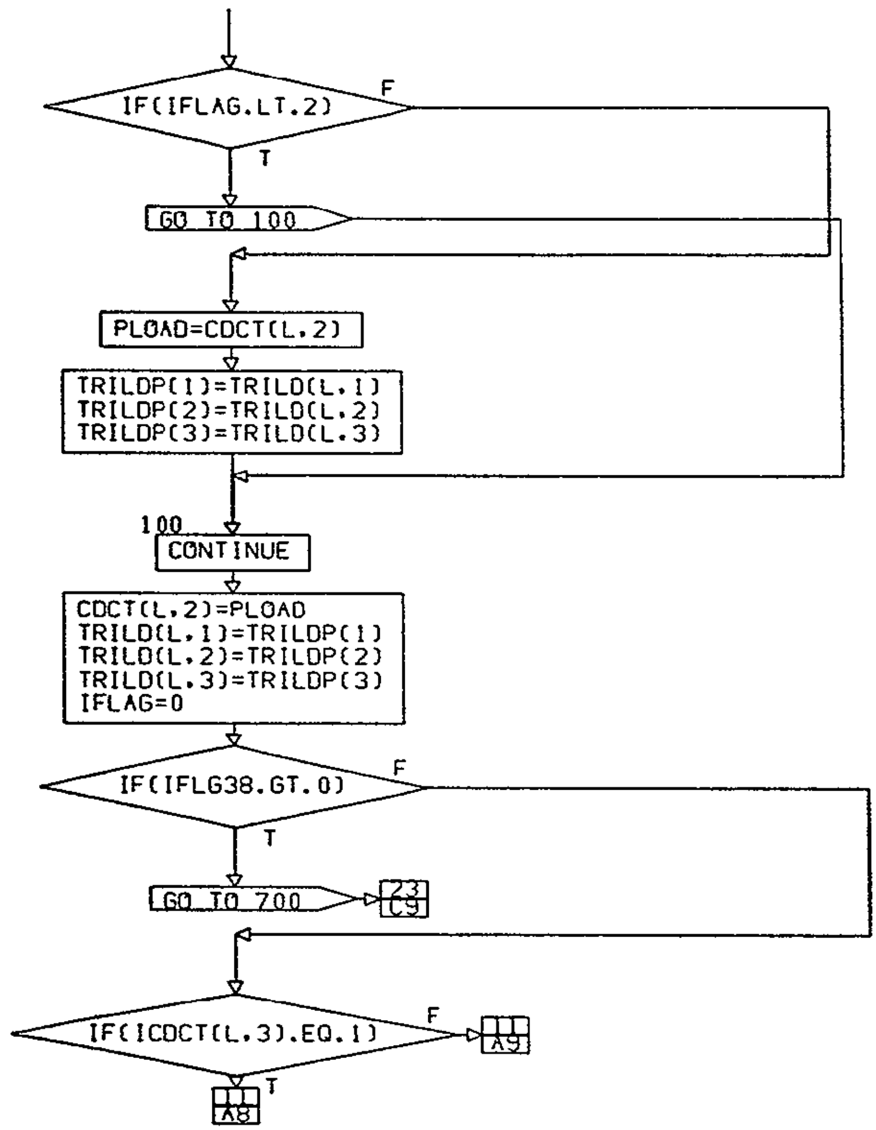


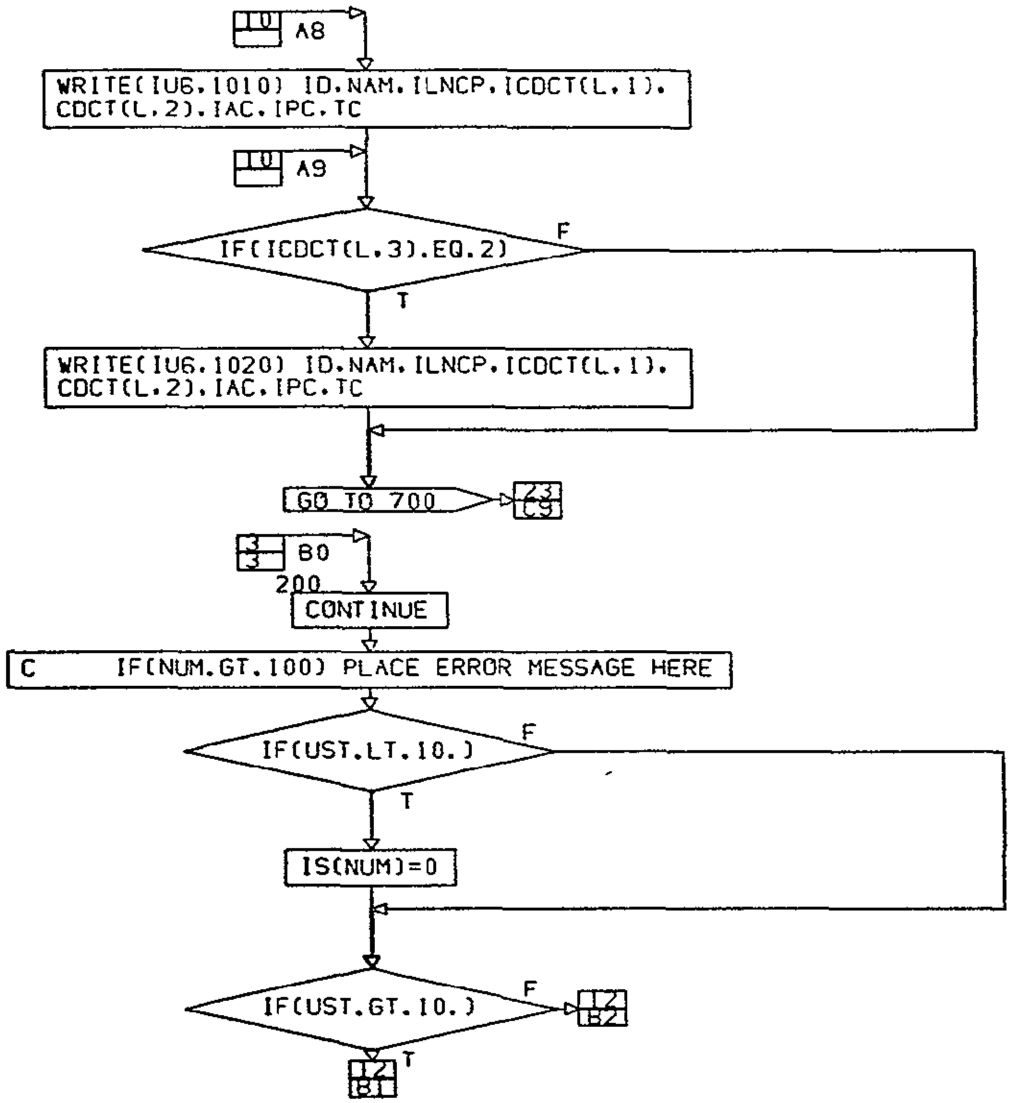
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 11

TPOUTJ
PG 10 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 12

TPOUTJ
PG 11 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

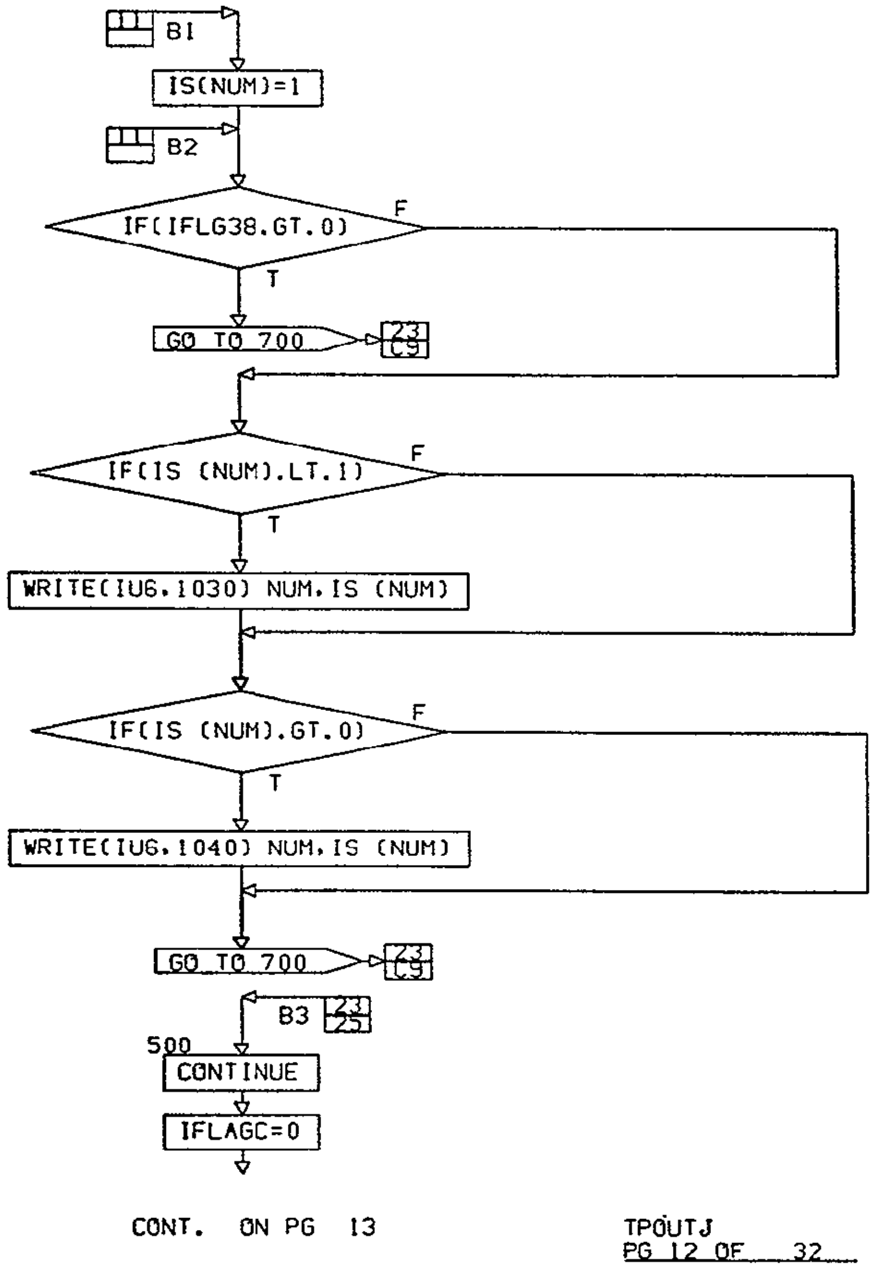
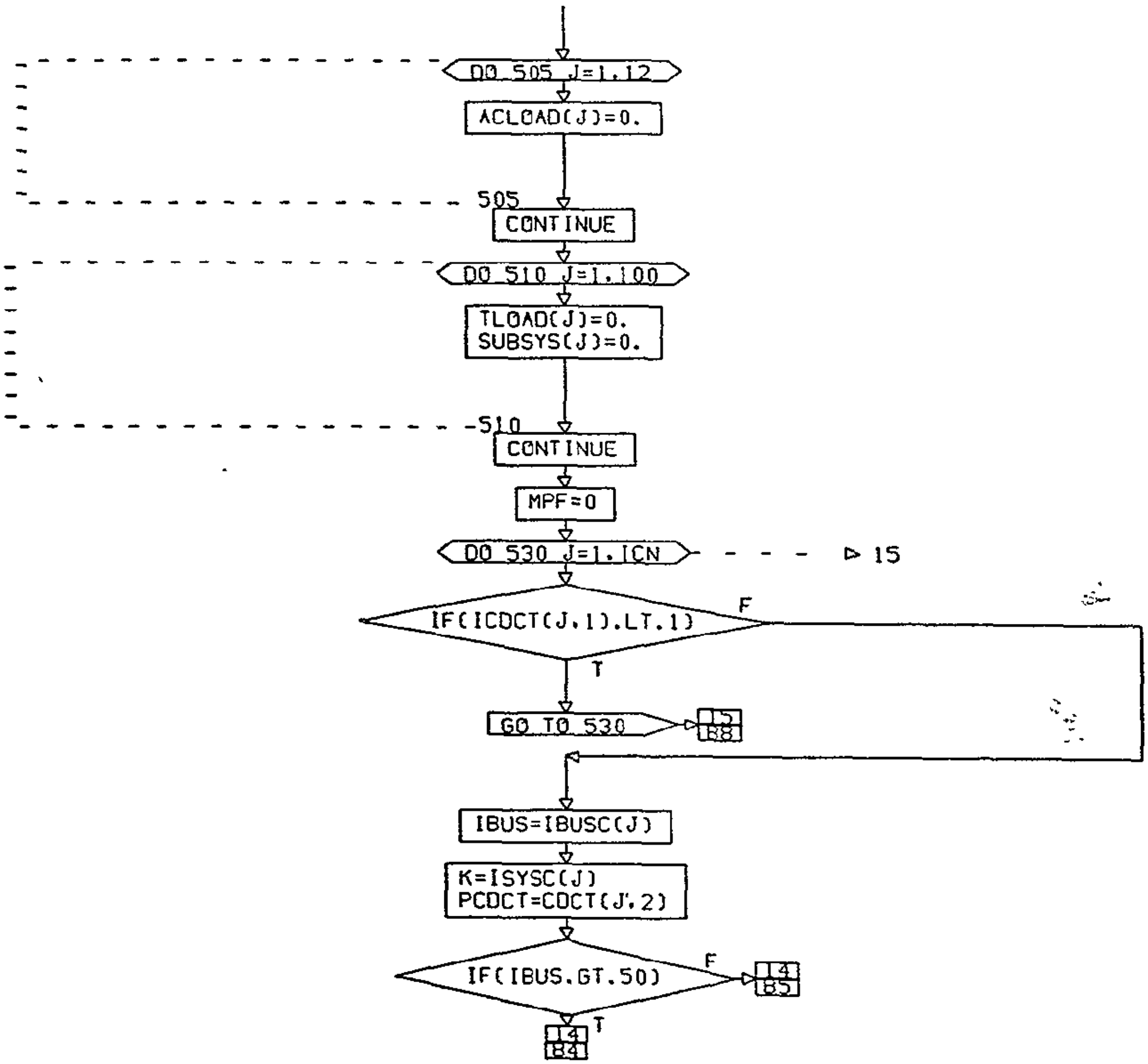


FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

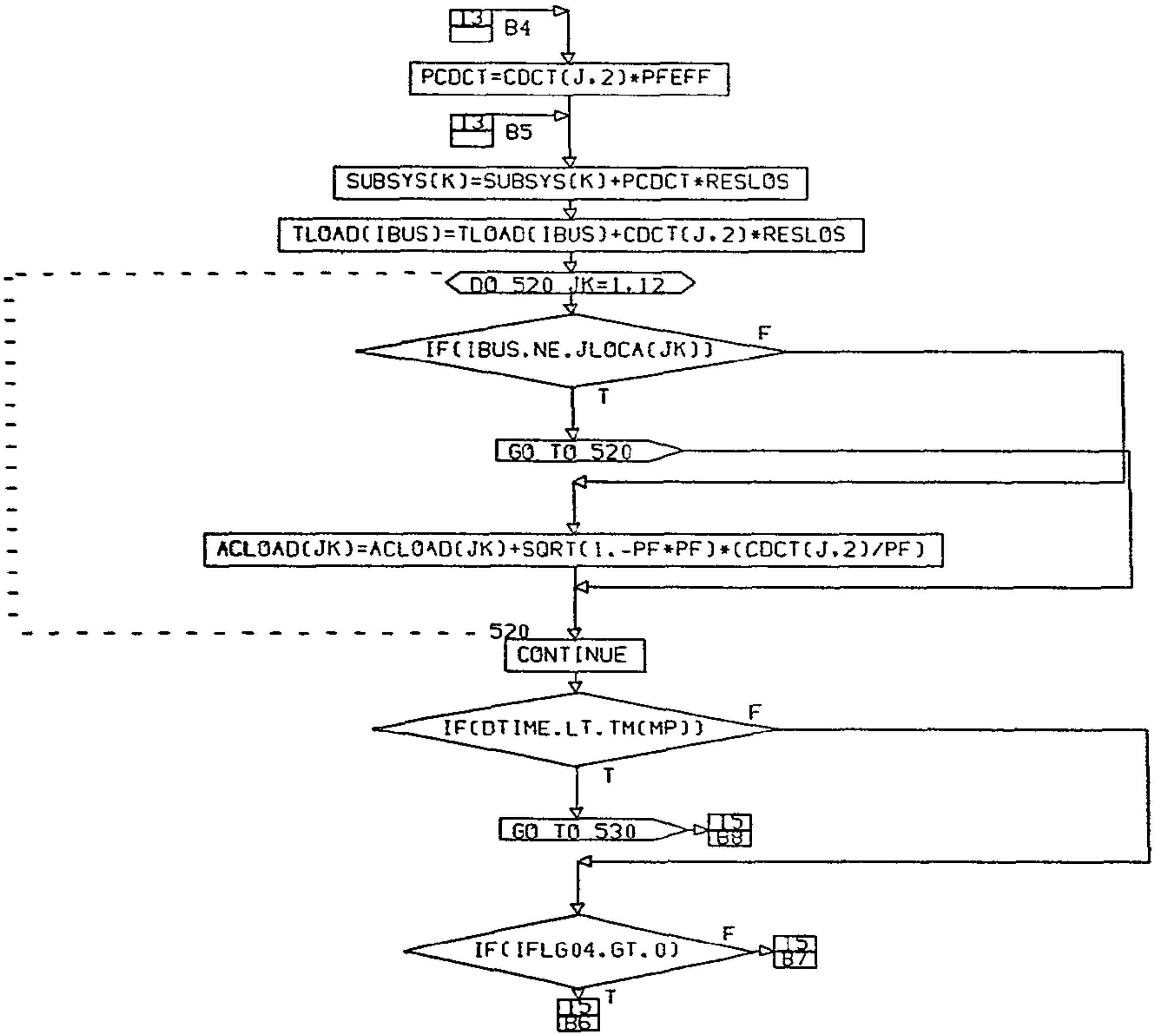


CONT. ON PG 14

TPOUTJ
PG 13 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY



CONT. ON PG 15

TPOUTJ
PG 14 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

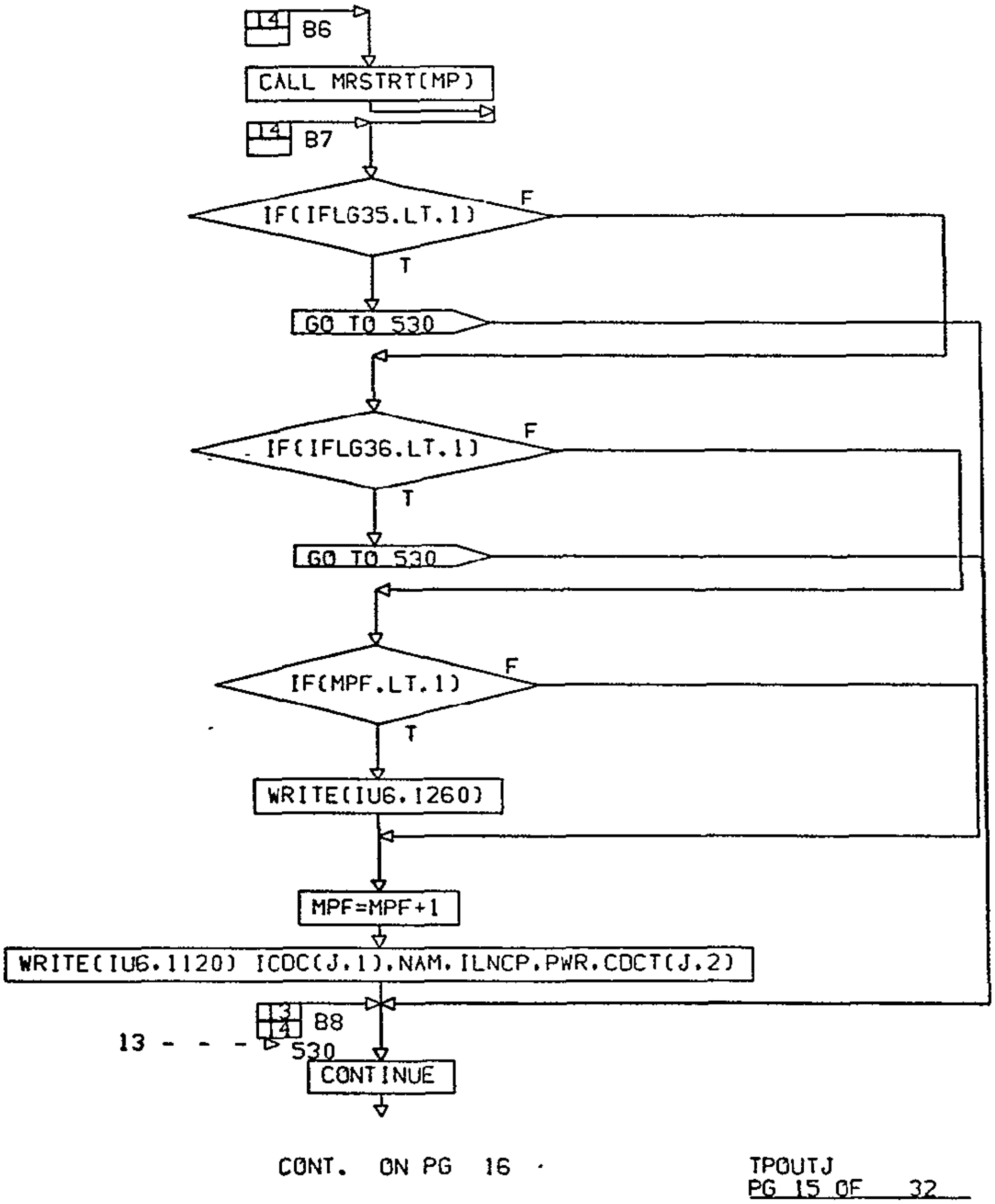
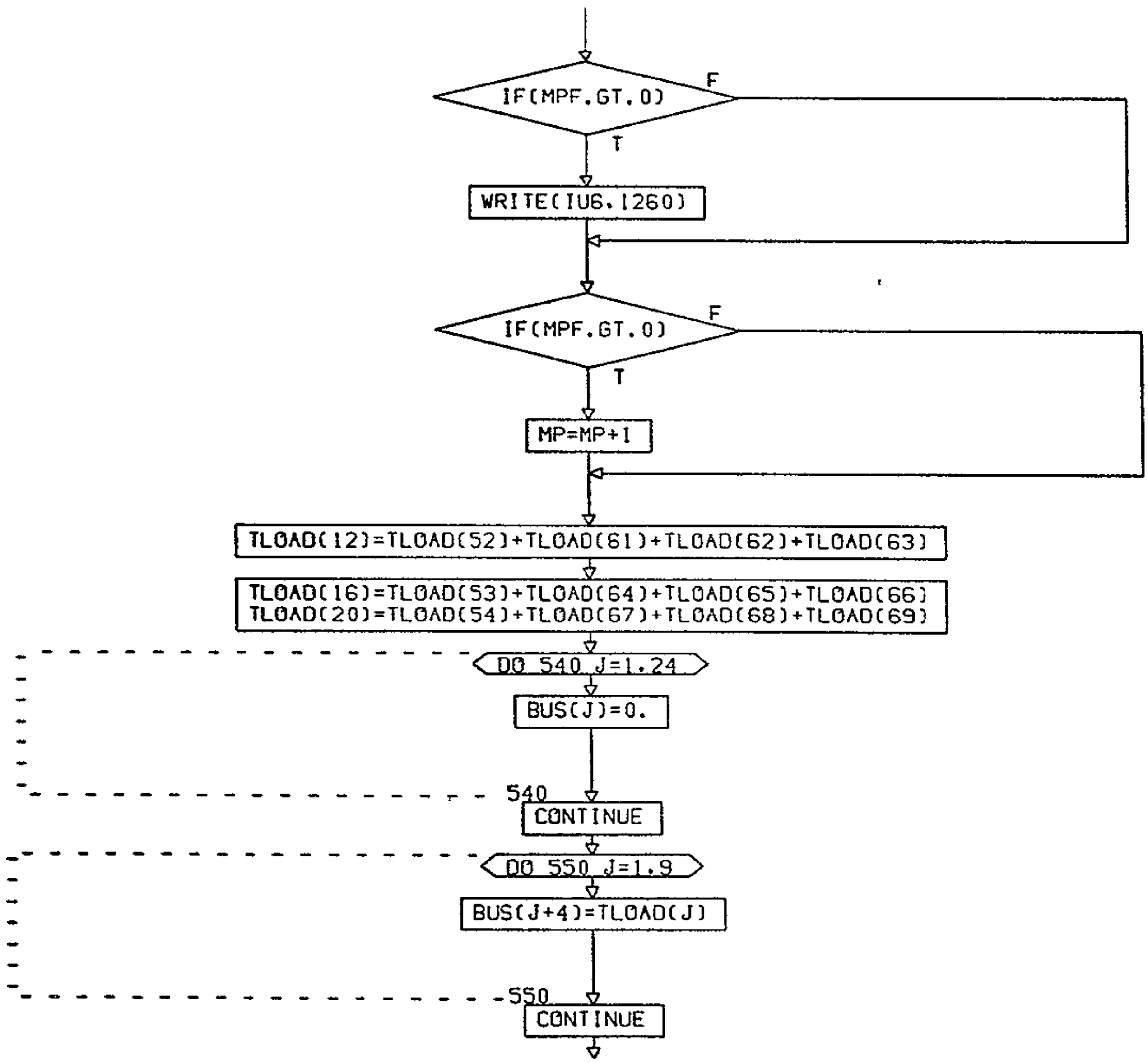


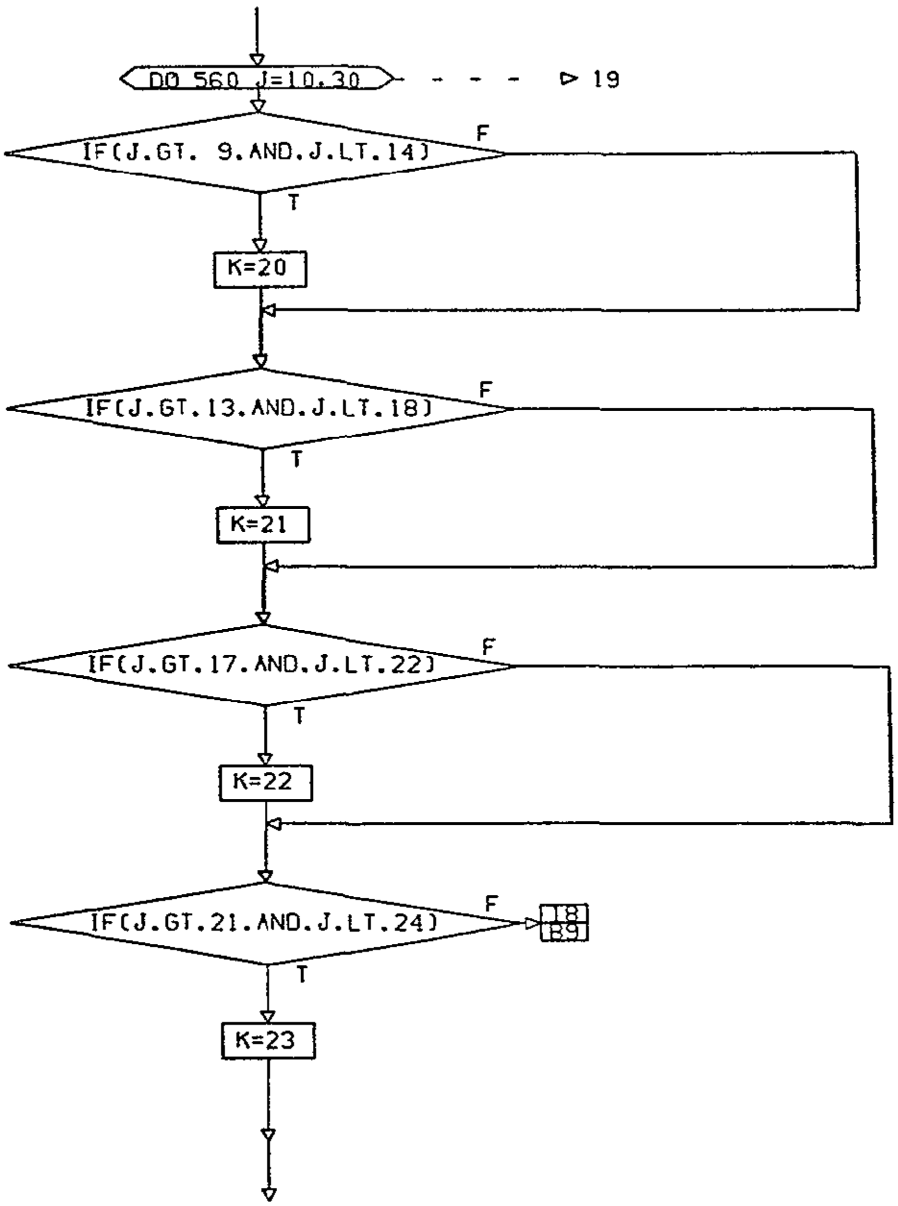
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 17

TPOUTJ
PG 16 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 18

TPOUTJ
PG 17 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

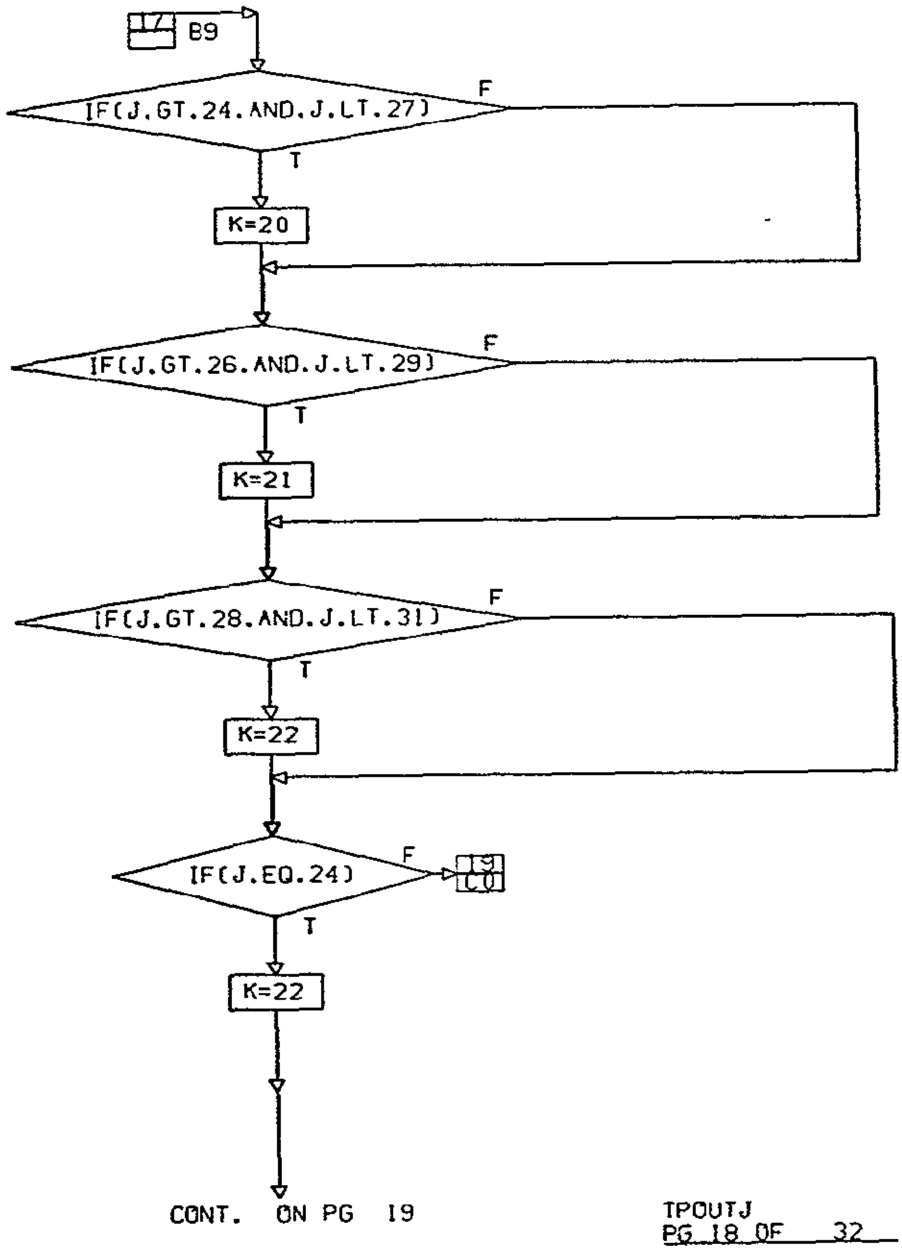
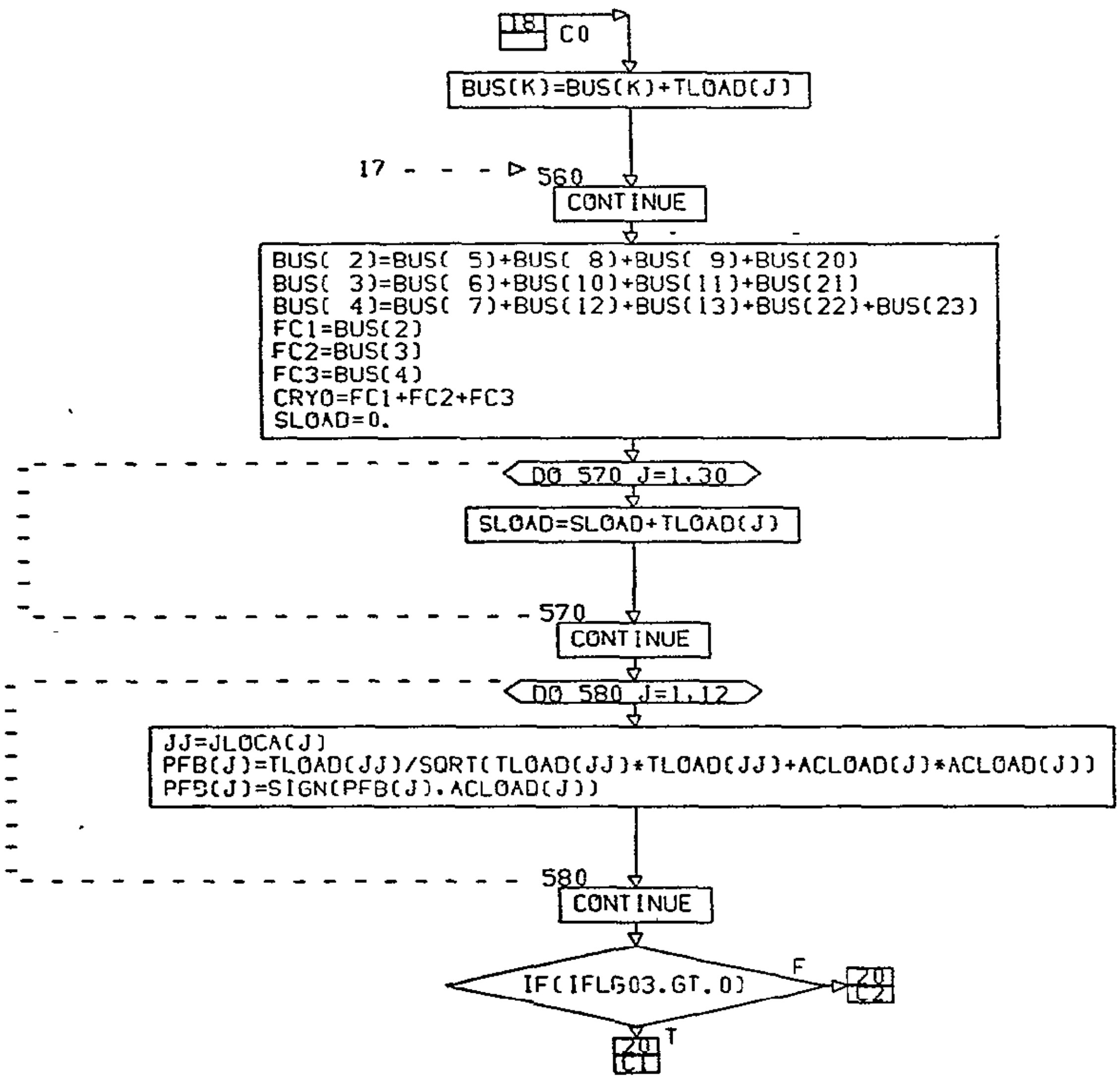
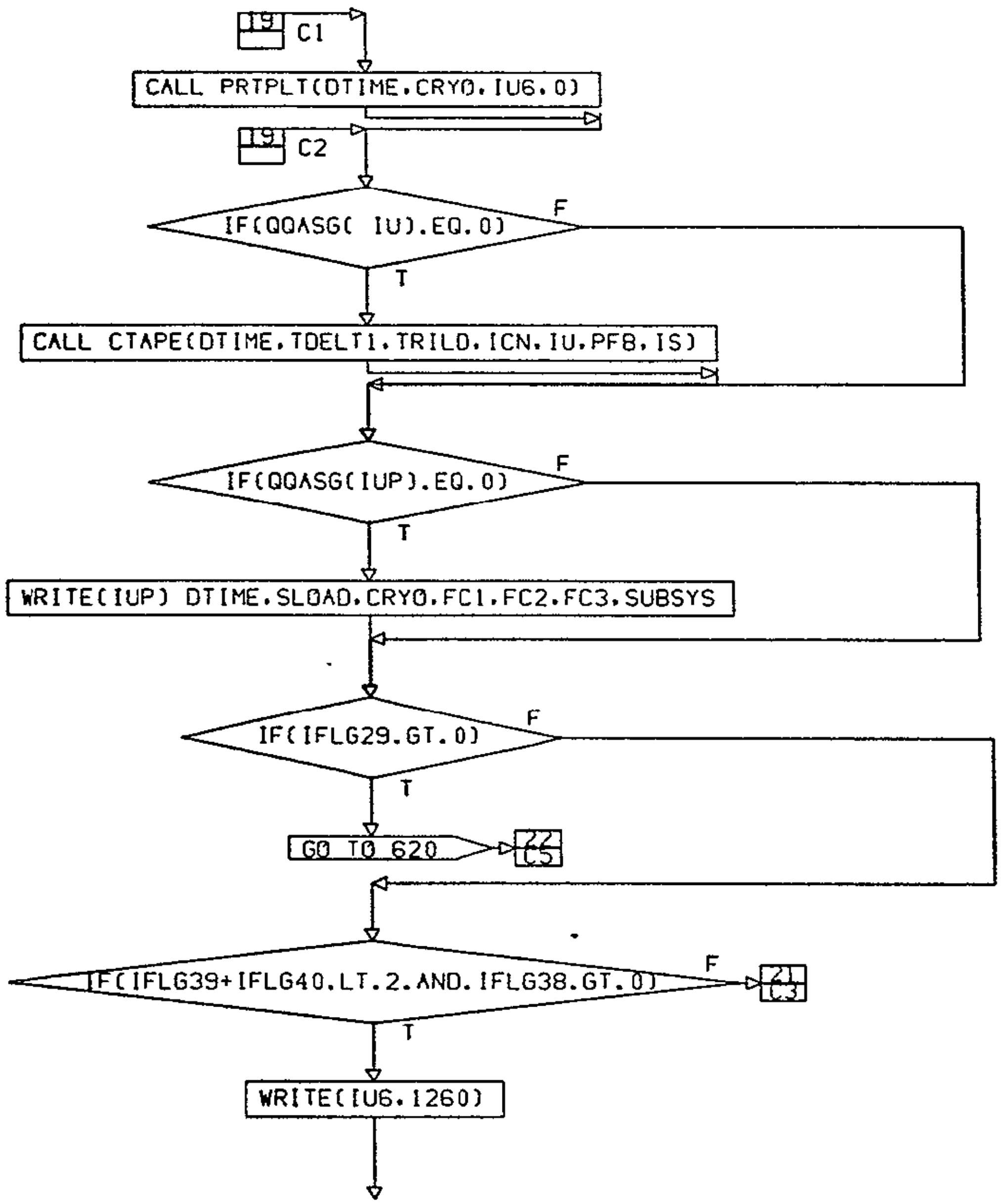


FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 20

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 21

TPOUTJ
PG 20 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

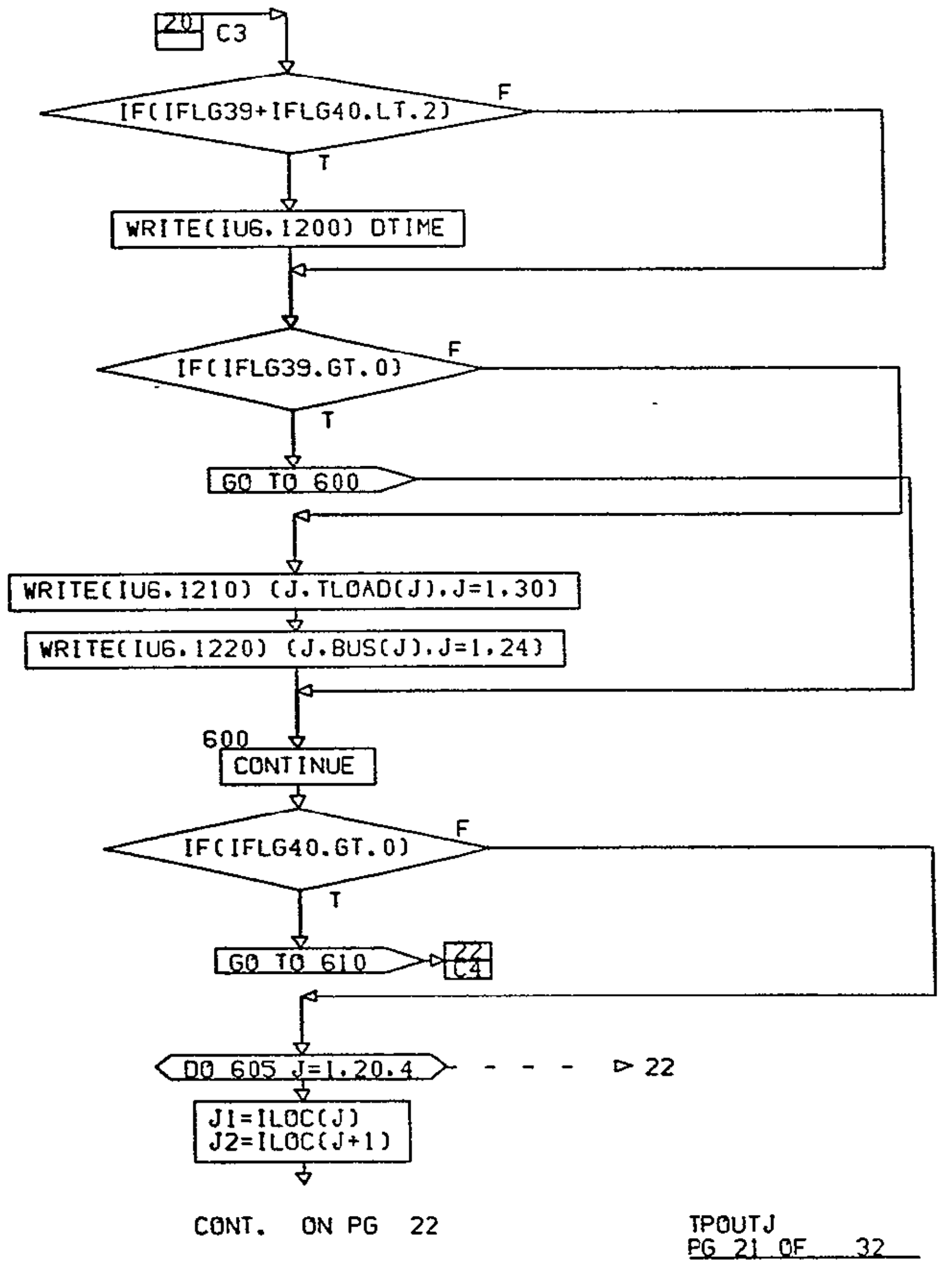
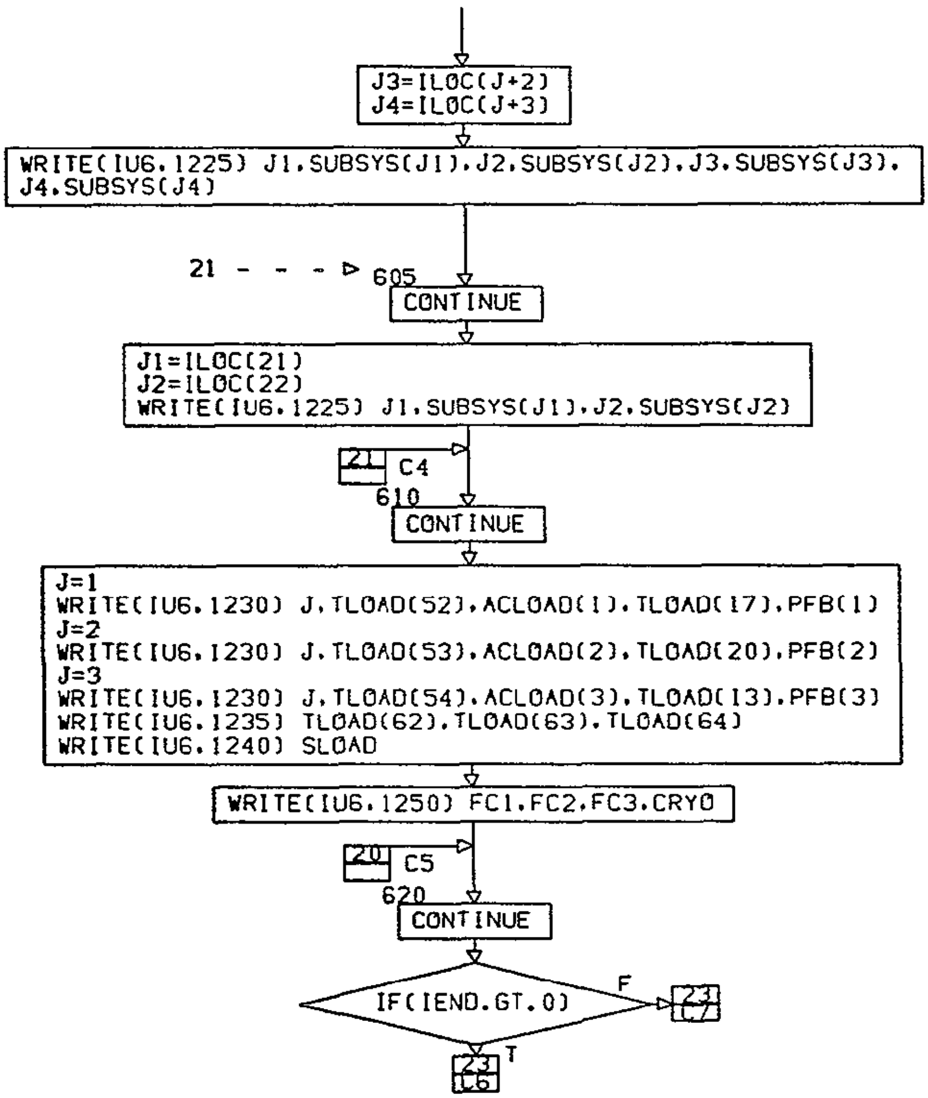


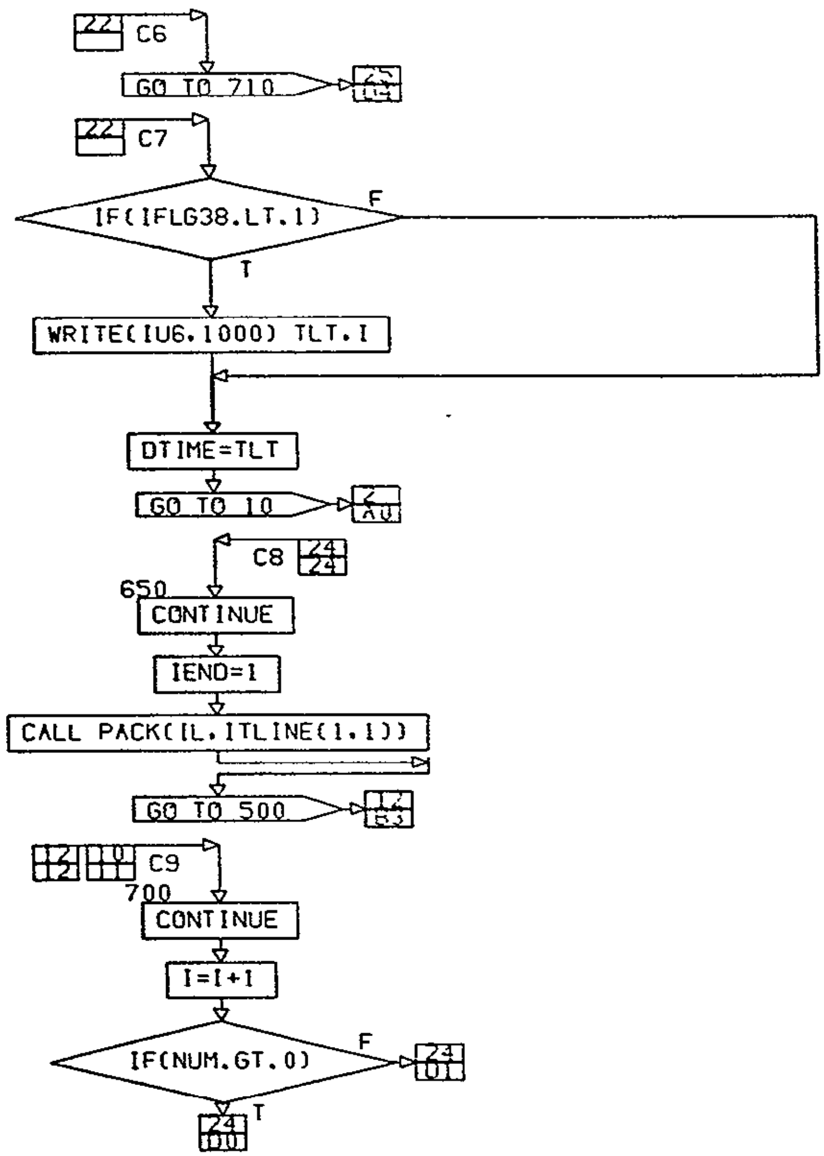
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 23

TPOUTJ
PG 22 OF 32

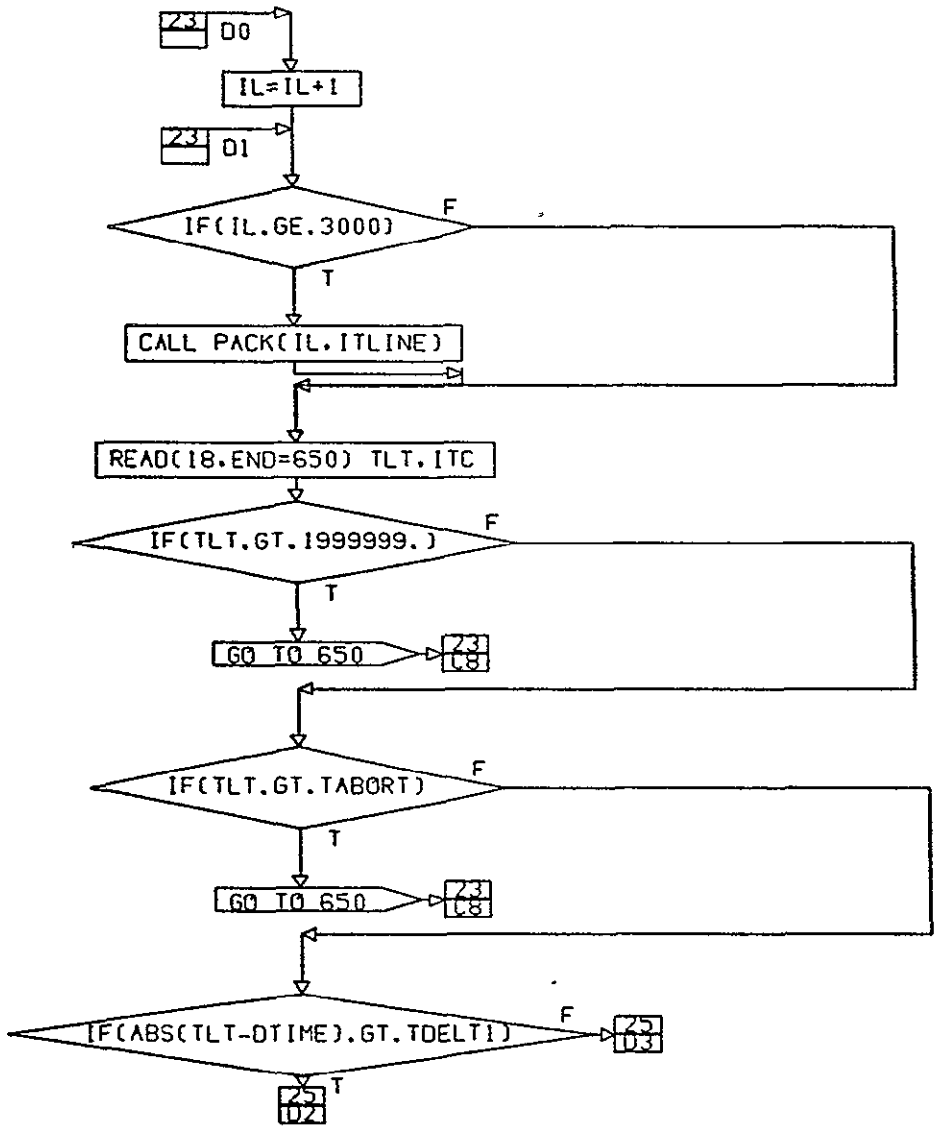
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 24

TPOUTJ
PG 23 OF 32

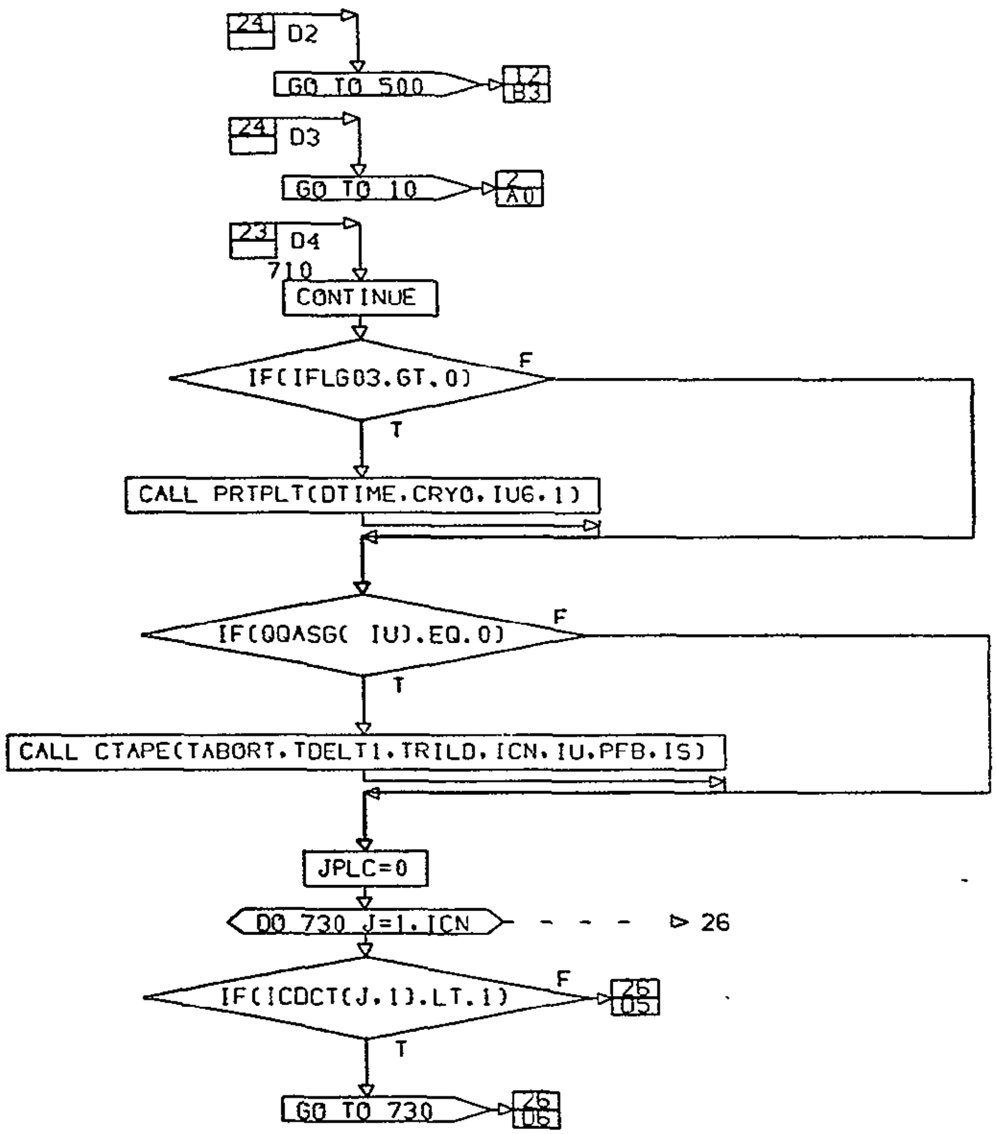
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 25

TPOUTJ
PG 24 OF 32

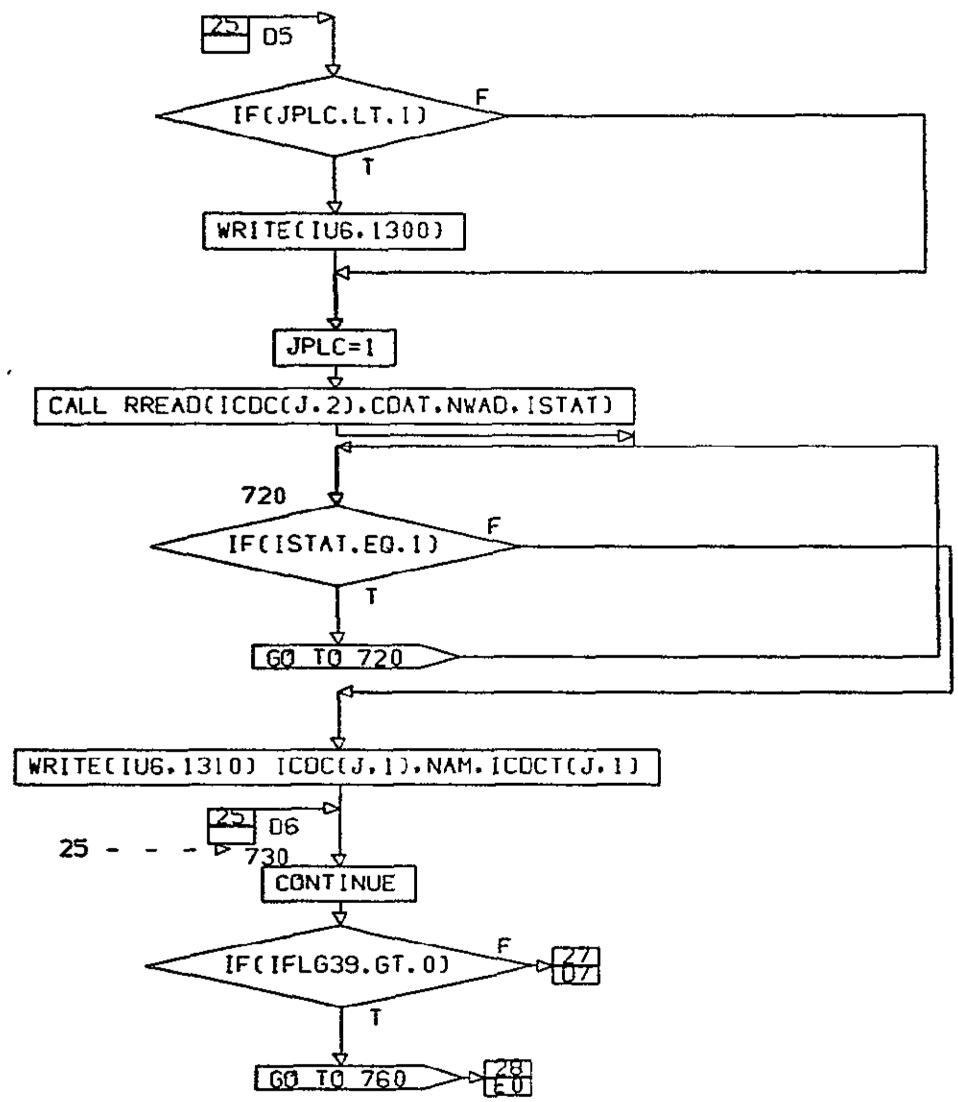
FIGURE 3.2.19. FUNCTINAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 26

TPOUTJ
PG 25 OF 32

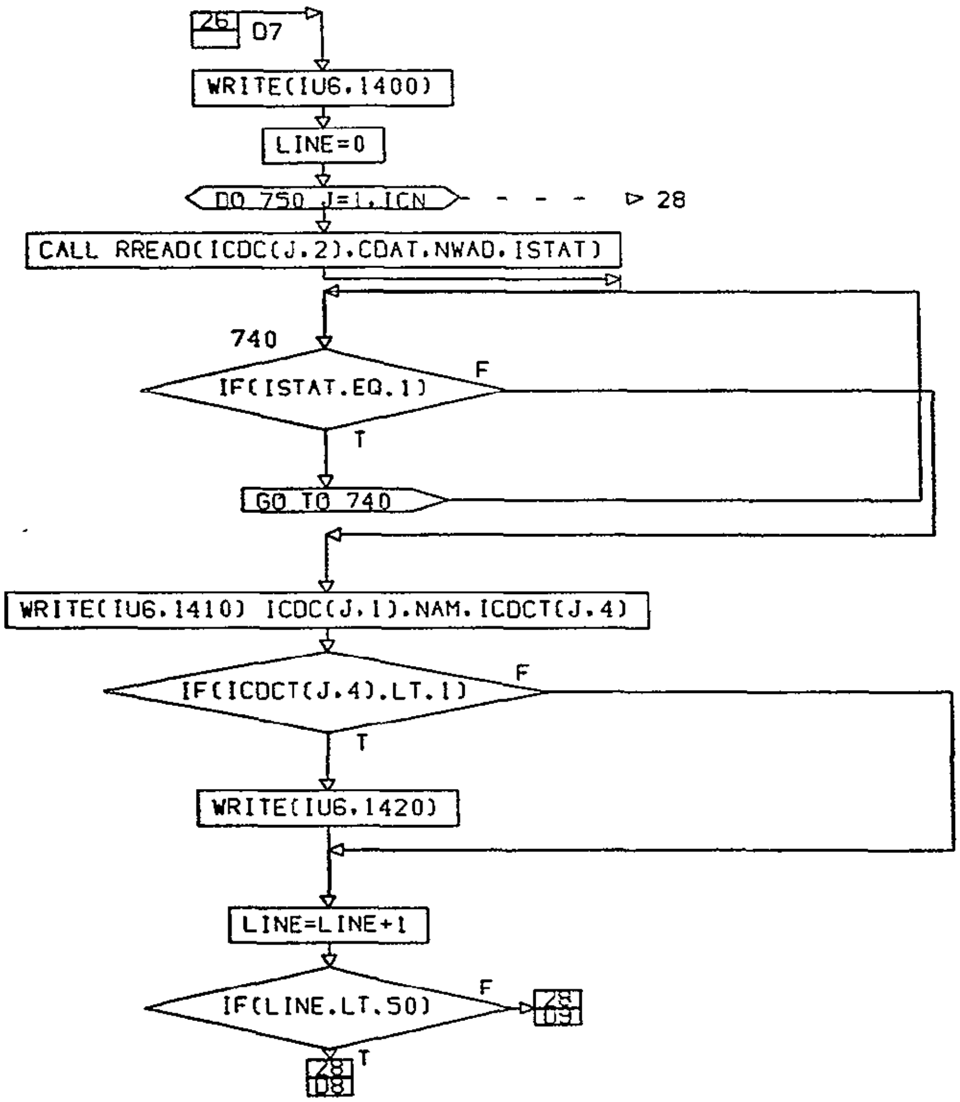
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 27

TPOUTJ
PG 26 OF 32

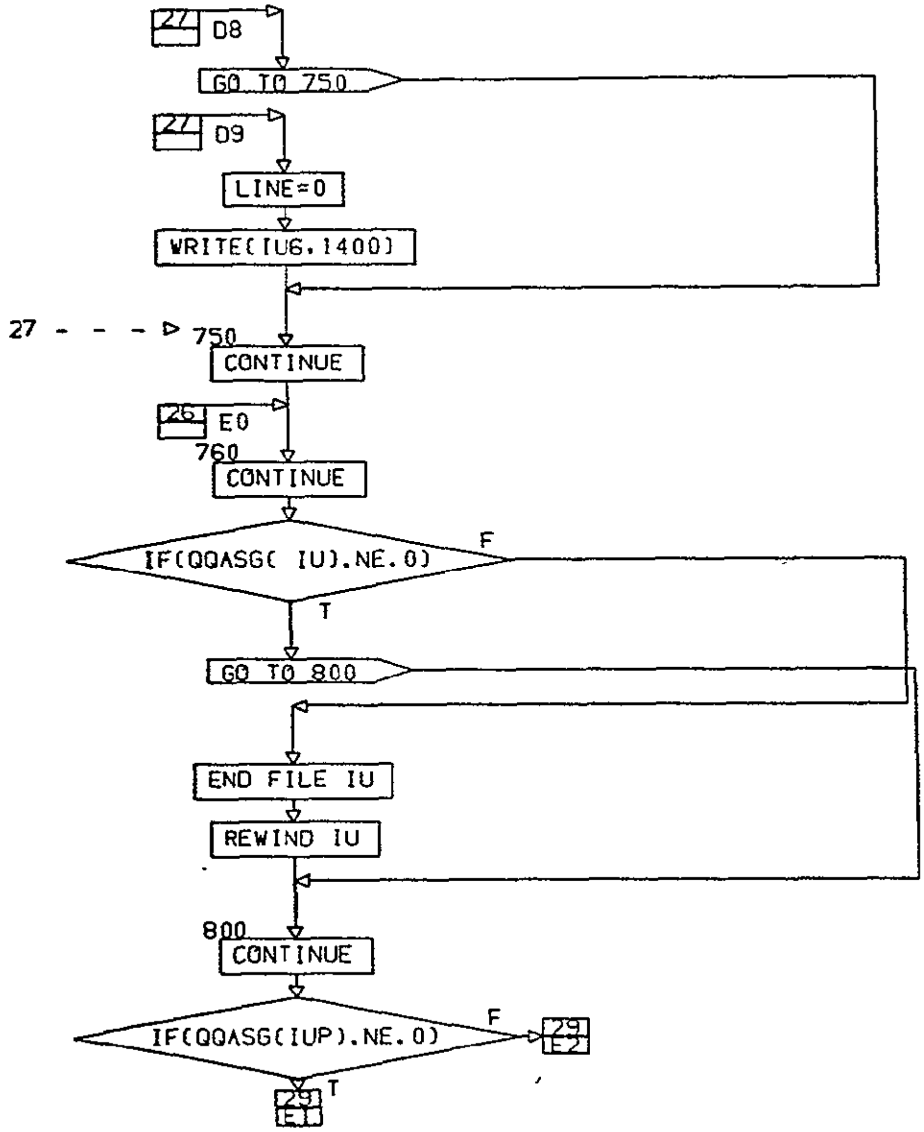
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 28

TPOUTJ
PG 27 OF 32

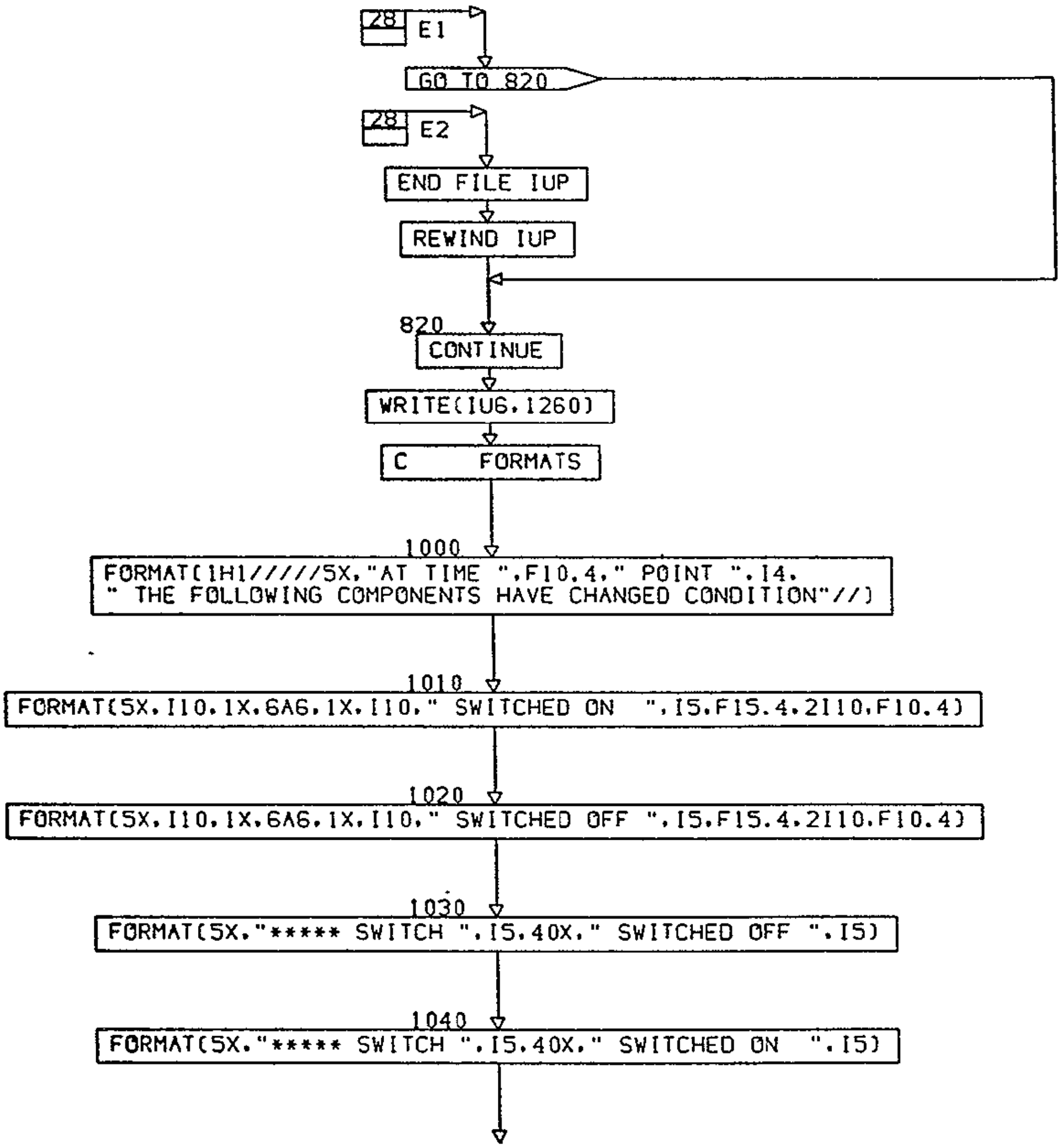
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 29

TPOUTJ
PG 28 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 30

TPOUTJ
PG 29 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

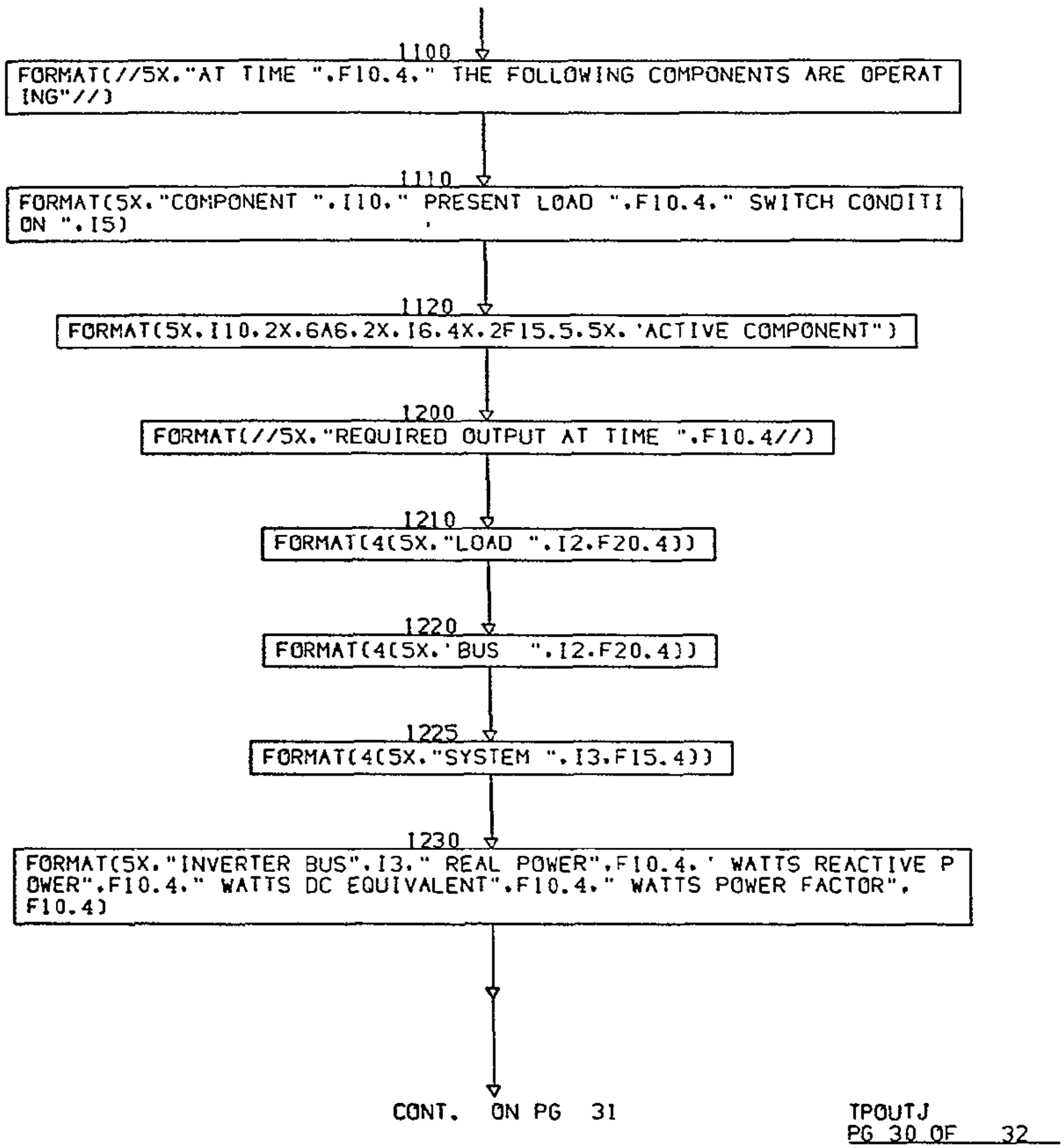
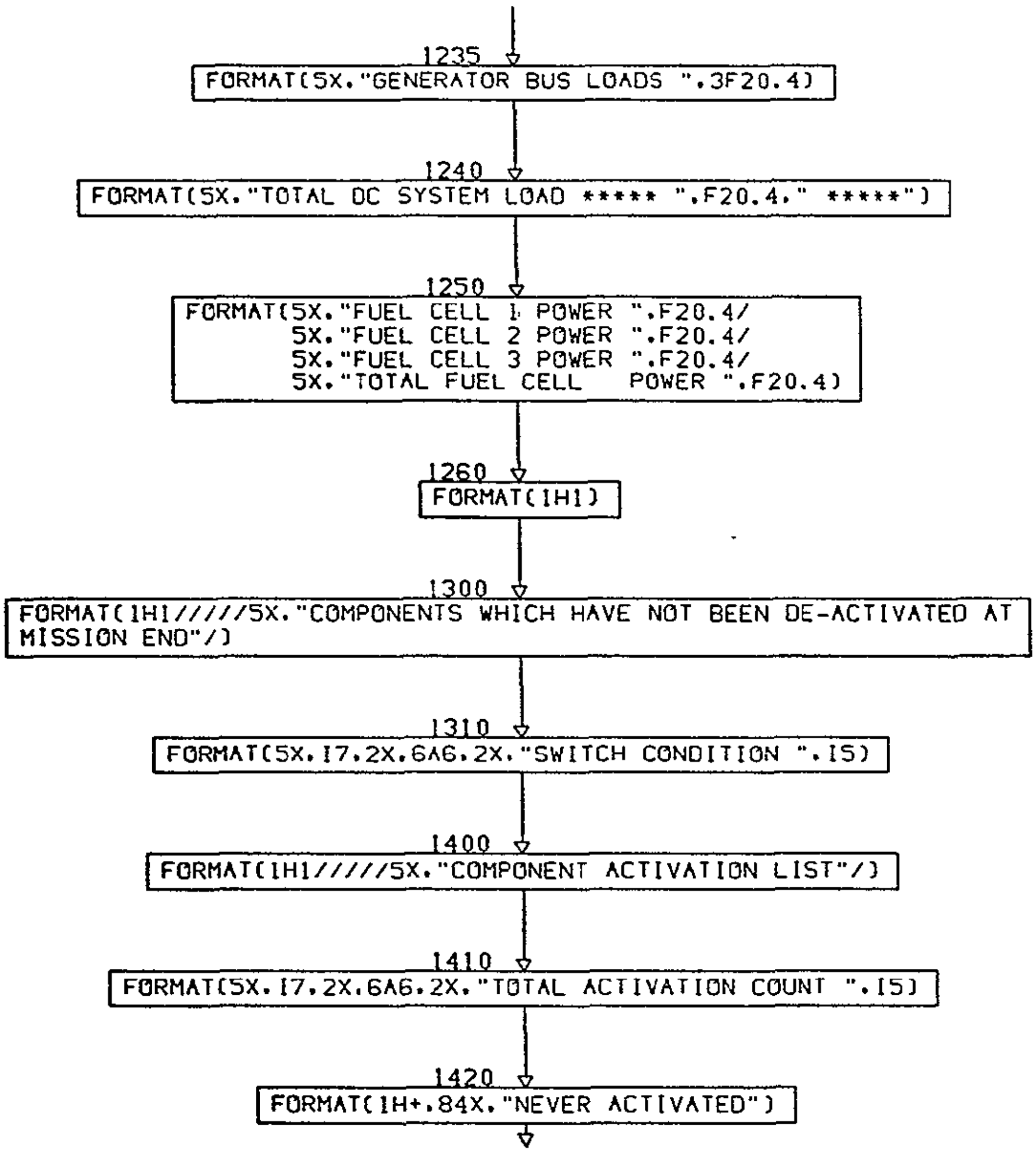


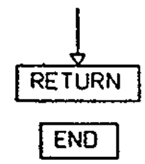
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



CONT. ON PG 32

TPOUTJ
PG 31 OF 32

FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)



TPOUTJ
PG 32 FINAL

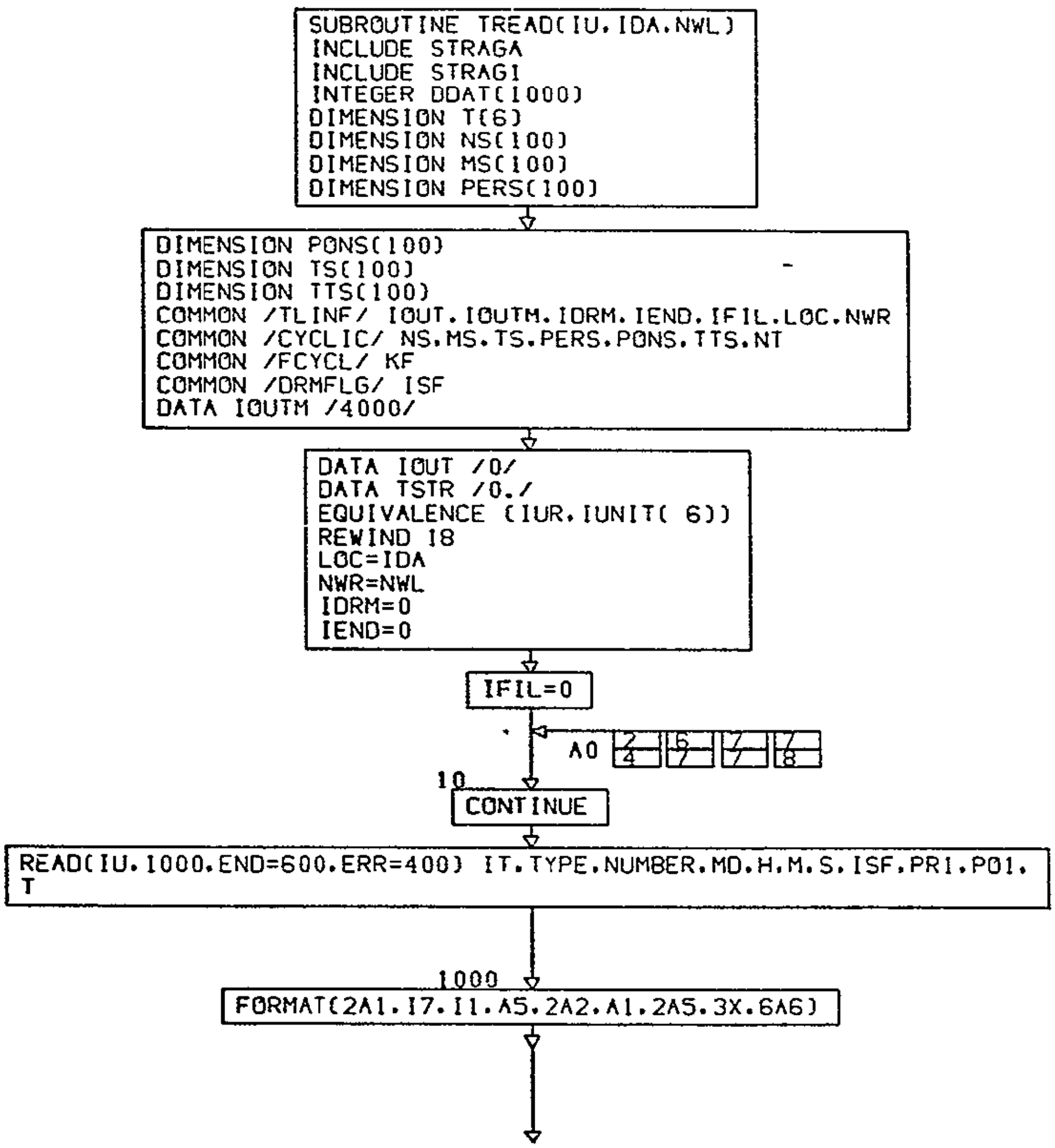
FIGURE 3.2.19. FUNCTIONAL FLOWCHART OF SUBROUTINE TPOUTJ (CONTINUED)

3.2.20 Subroutine: TREAD

PURPOSE: Read the input timeline and control the creation of the event timeline.

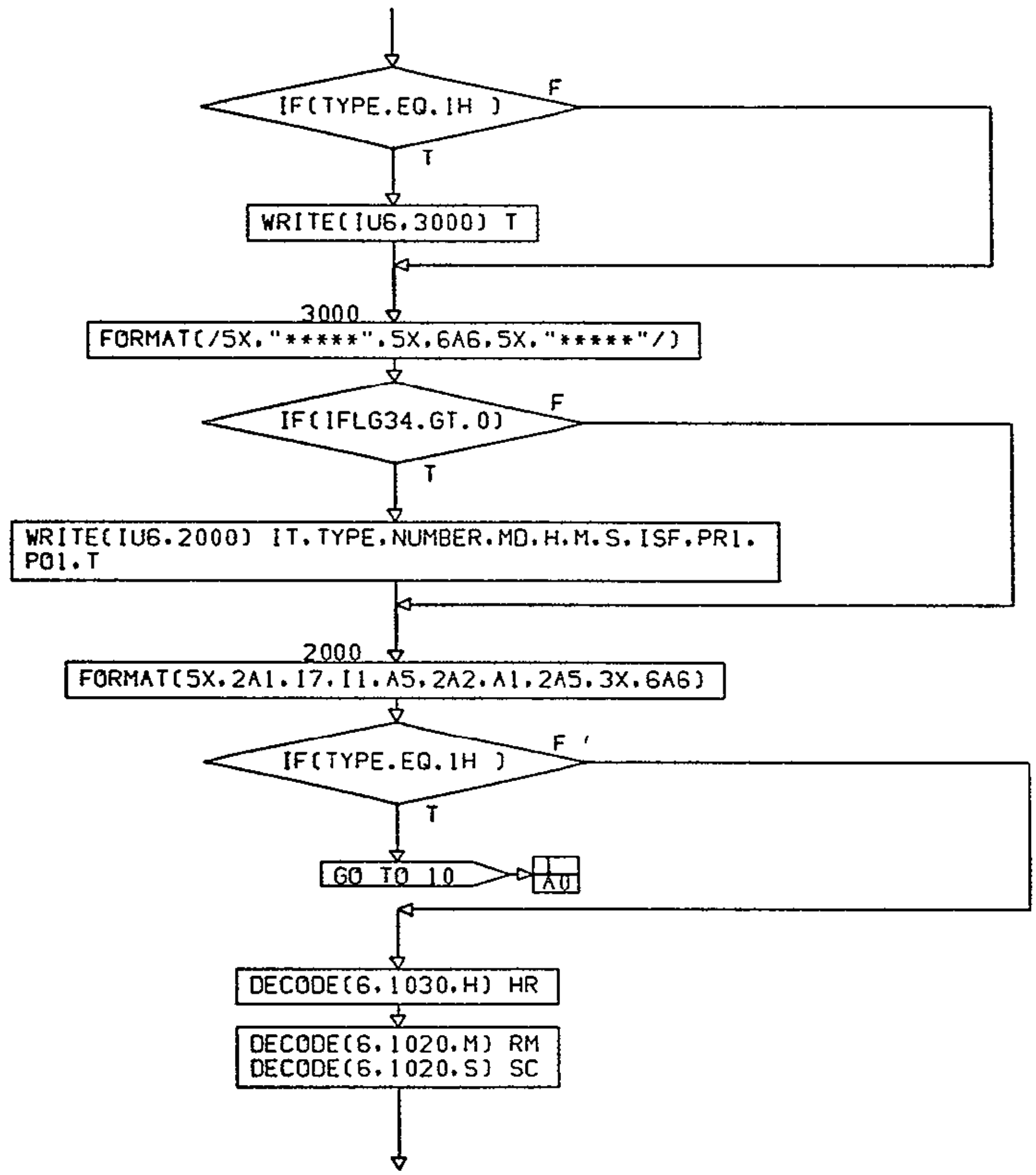
METHOD: This routine reads an input timeline card, determines its type, and calls the correct routine to handle the type.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.2.20. See Appendix for definition of all variables.



CONT. ON PG 2

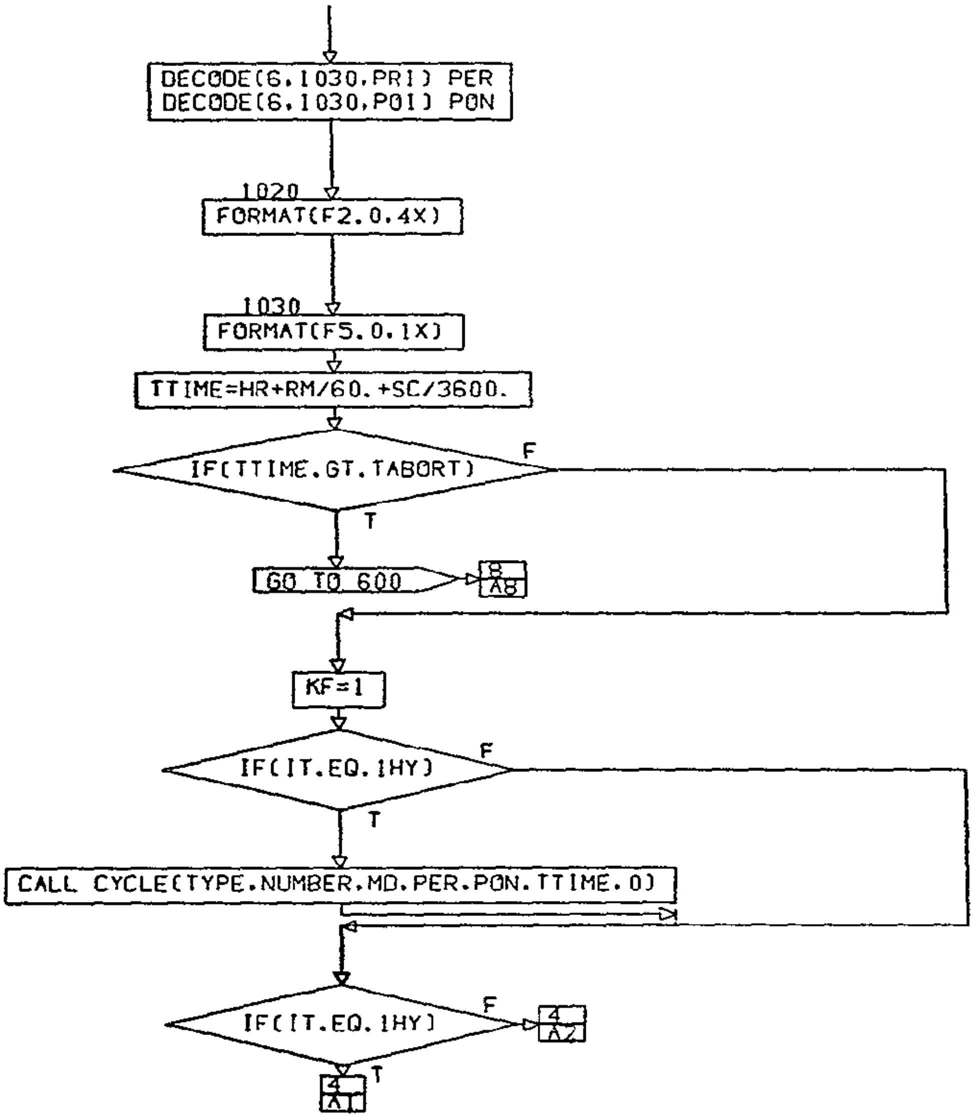
FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD



CONT. ON PG 3

TREAD
PG 2 OF 9

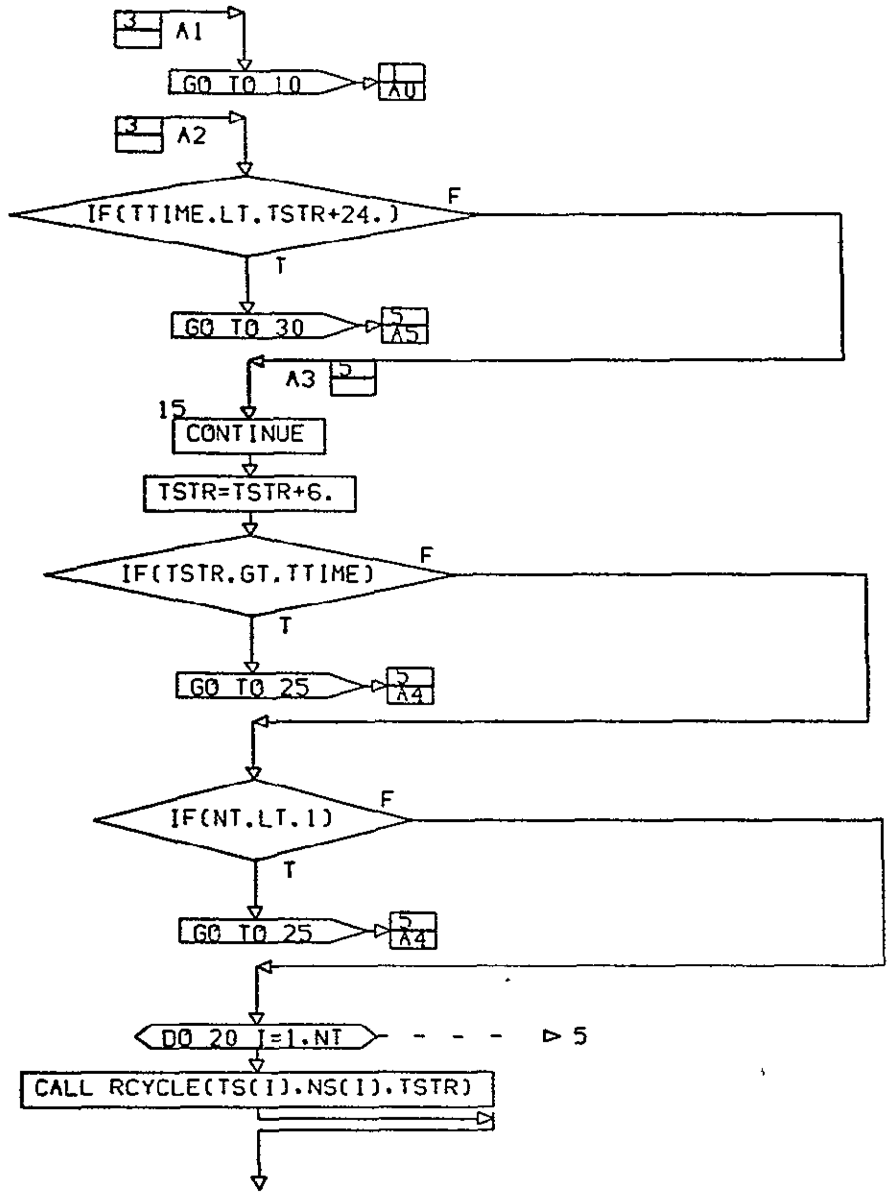
FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)



CONT. ON PG 4

TREAD
PG 3 OF 9

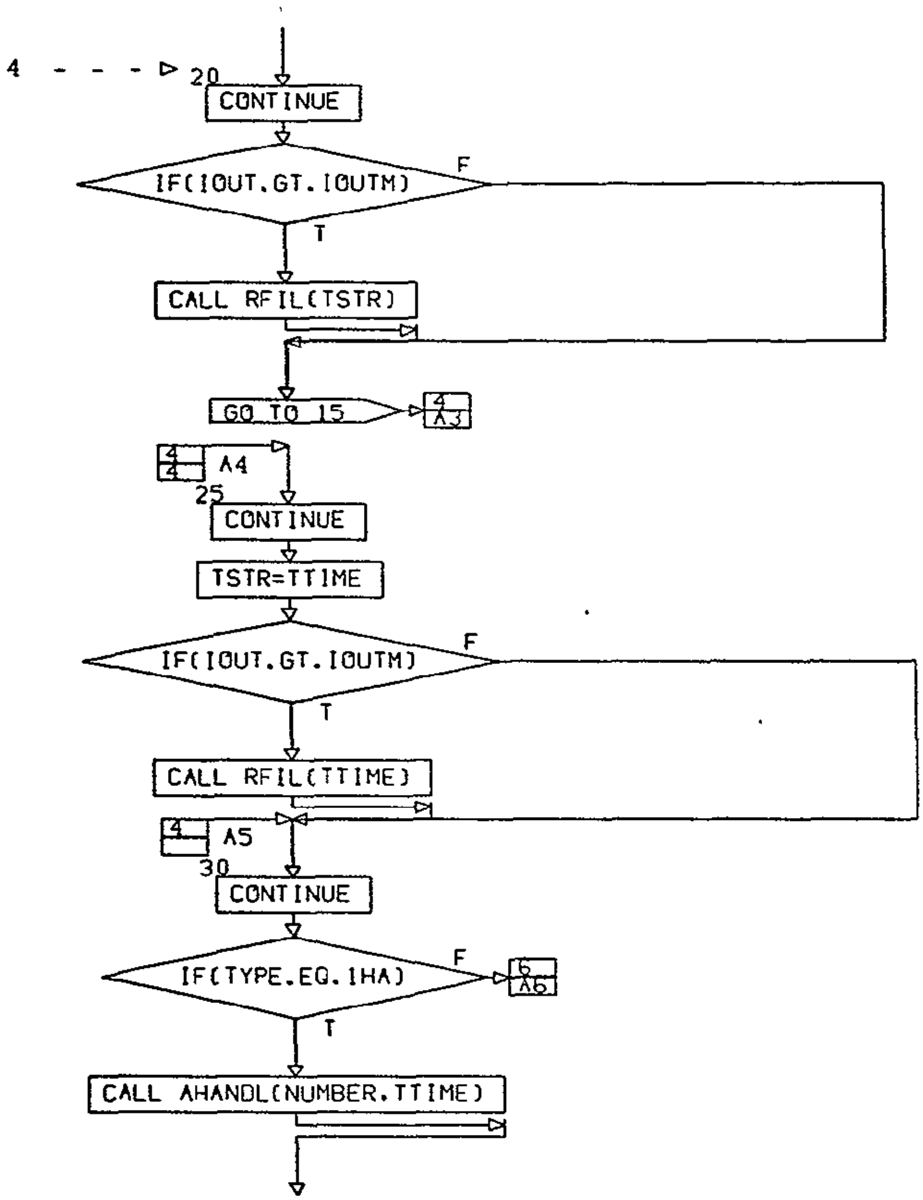
FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)



CONT. ON PG 5

TREAD
PG 4 OF 9

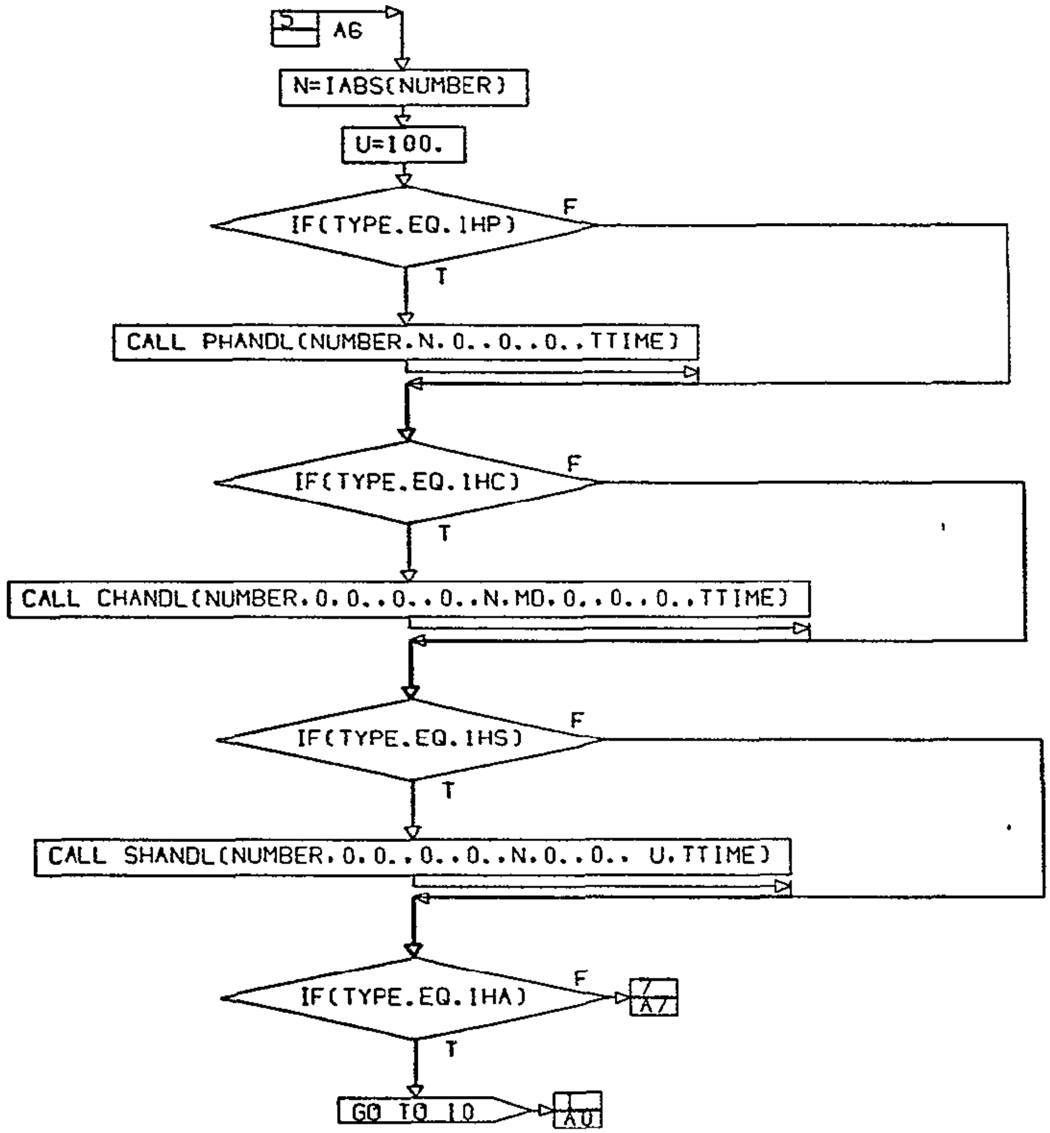
FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)



CONT. ON PG 6

TREAD
PG 5 OF 9

FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)



CONT. ON PG 7

TREAD
PG 6 OF 9

FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)

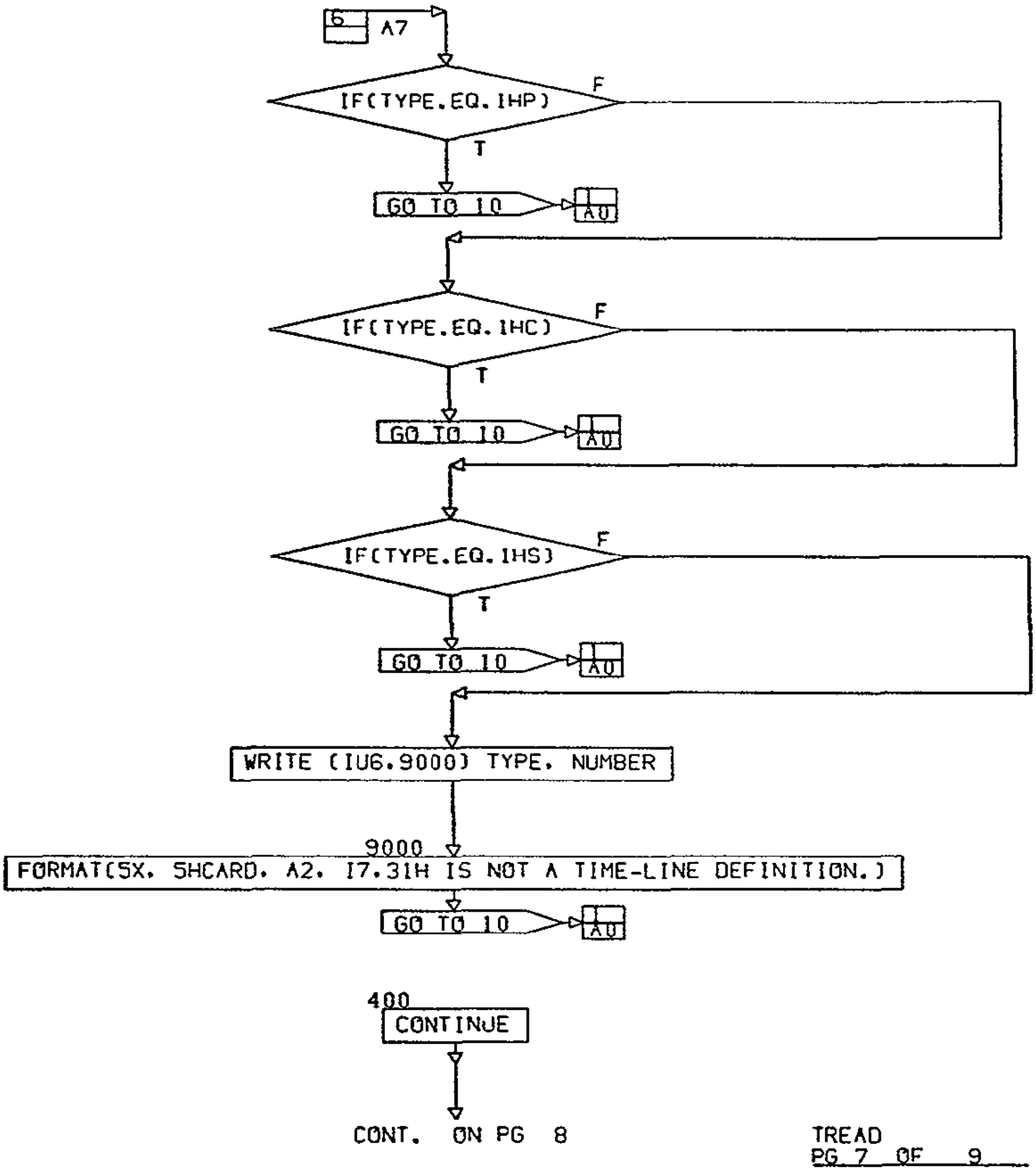


FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)

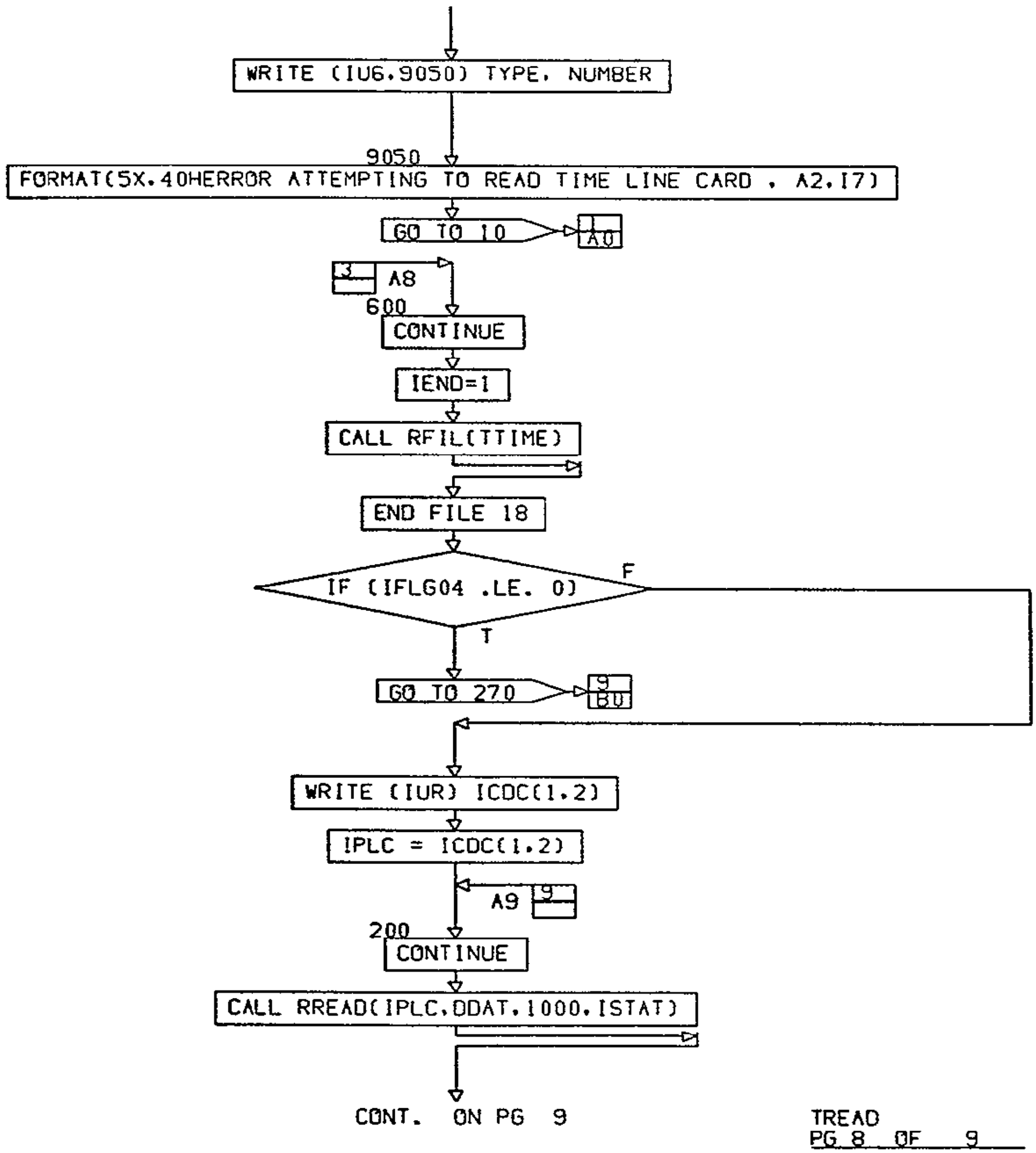


FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)

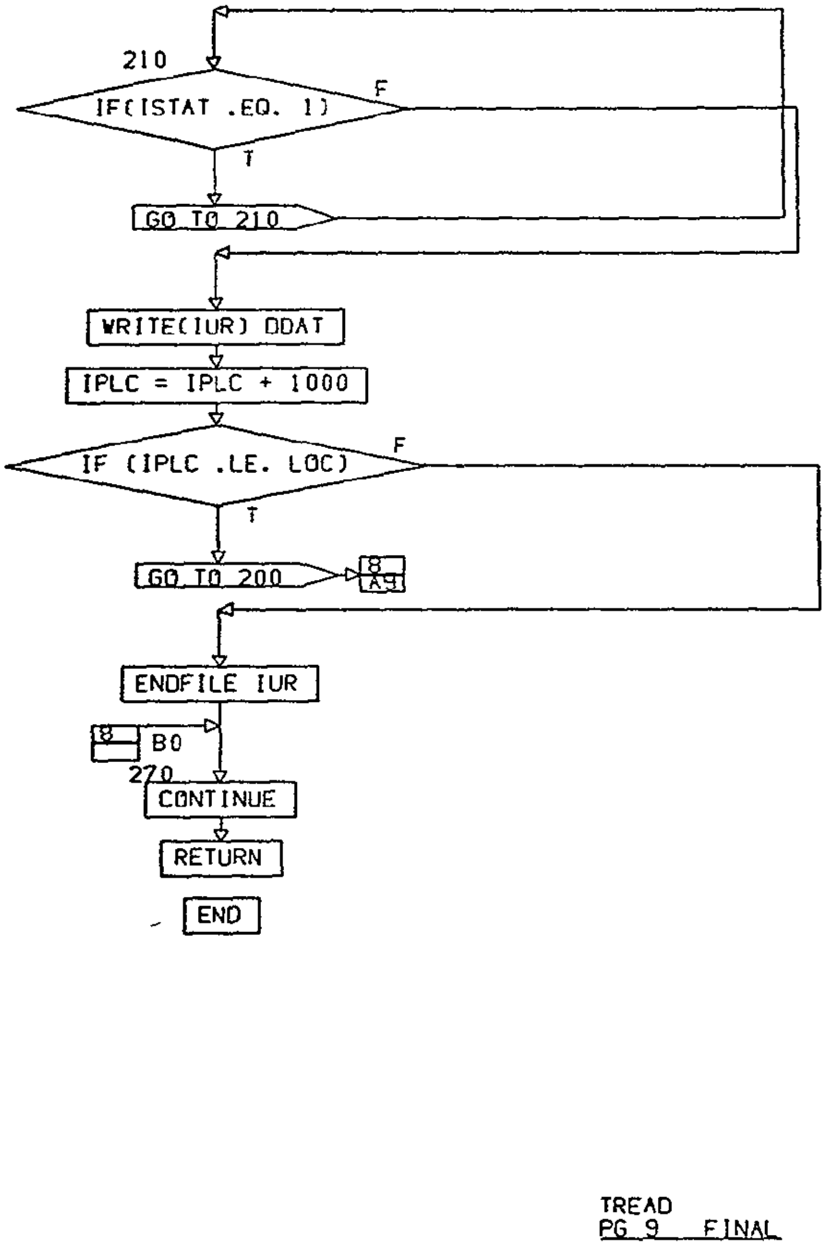


FIGURE 3.2.20. FUNCTIONAL FLOWCHART OF SUBROUTINE TREAD (CONTINUED)

3.3. PHASE II SUBROUTINES

3.3.1 Subroutine: PHASE2

PURPOSE: This routine controls the simulation of the Shuttle vehicle's electrical power system.

METHOD: Using a user supplied time step this routine controls the following functions to simulate the vehicle from some input simulation start time to some input simulation abort time.

1. Reads initialization data and solves the initialization calculations
2. Determines the source I-V characteristics
3. Solves the distribution system to find node voltages and branch currents
4. Checks for constraint violations
5. Provides the required outputs
6. Repeats Steps 2 through 5 until the end of the simulation

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.1. See Appendix for definition of all variables.

```
SUBROUTINE PHASE2
```

```
C      ALL COMMUNICATION IS TO TAKE PLACE THROUGH PDP ELEMENTS
C      STRAG1    ALL DRIVER RELATED ELEMENTS
C      STRAG2    ALL DC SYSTEM RELATED ELEMENTS
C      STRAG3    ALL AC SYSTEM RELATED ELEMENTS
C      STRAG4    ALL BATTERY RELATED ELEMENTS
C      STRAG5    ALL FUEL CELL RELATED ELEMENTS
C      STRAG6    ALL TR UNIT RELATED ELEMENTS
C      PROGRAM FLOW
```

```
C          INITAL - READ OPTION CARD, FIXED DATA, INITIALIZE ALL SOURCES
C                   READ DOWN THE TIMELINE, DETERMINE INITIAL LOADING
C          CIRSOL - CIRCUIT SOLUTION DRIVER
C                   CALLS A. ACINVT - SOLVES AC INVERTERS APPLIES AS LOA
C                                     TO DC CIRCUIT
C                         B. DCSOLV - SOLVES DC CIRCUIT
C                         C. TRAPLY - APPLIES TR UNIT AS LOAD TO AC CIRC
C                         D. ACSOLV - SOLVES AC CIRCUIT
```

```
C          REDLIN - POWER MANAGEMENT
C          INCRA  - CONTROLS TIMESTEP IF REQUIRED READS TIMELINE INPUT
C                   CALLS A. TLREAD - READS CARD TIMELINE
C          CHARGE - DETERMINES WHICH, IF ANY, 10 AH BATTERY IS CONNECTED
C                   TO THE BATTERY CHARGER AND UPDATES THAT BATTERYS STA
C          QCLTMP - UPDATES BATTERY STATUS
C          BATTIV - CALCULATES CORRECT BATTERY OPERATING IV CURVE
C          FUCLTM - UPDATES FUEL CELL STATUS
```

```
C          FUCLIV - CALCULATES FUEL CELL OPERATING IV CURVE
C          TRUNIT - CALCULATES TRANSFORMER-RECTIFIER UNIT OPERATING IV C
```

```
INCLUDE STRAG1
INCLUDE STRAG2
INCLUDE STRAG3
INCLUDE STRAG4
INCLUDE STRAG5
INCLUDE STRAG6
INTEGER QQASG
DIMENSION SSOC(6)
```

```
COMMON /PRINT/ ICARD,ITAPE,IPRNT
COMMON /TOTPWR/ PWRTOT
COMMON /ACPOWF/ PFEFF,RESLOS
```

CONT. ON PG 2

PHASE2
PG 1 OF 8

FIGURE 3.3.1 FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2

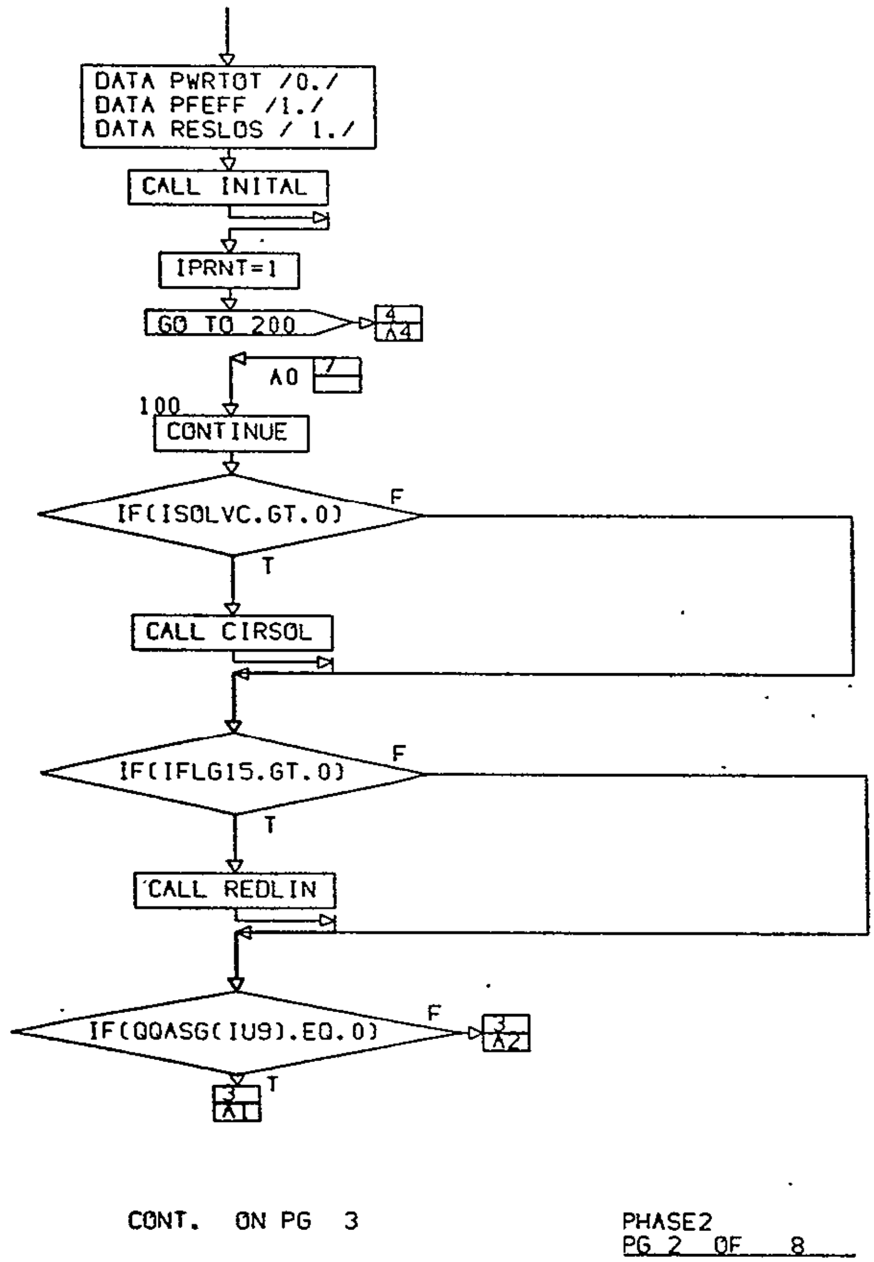
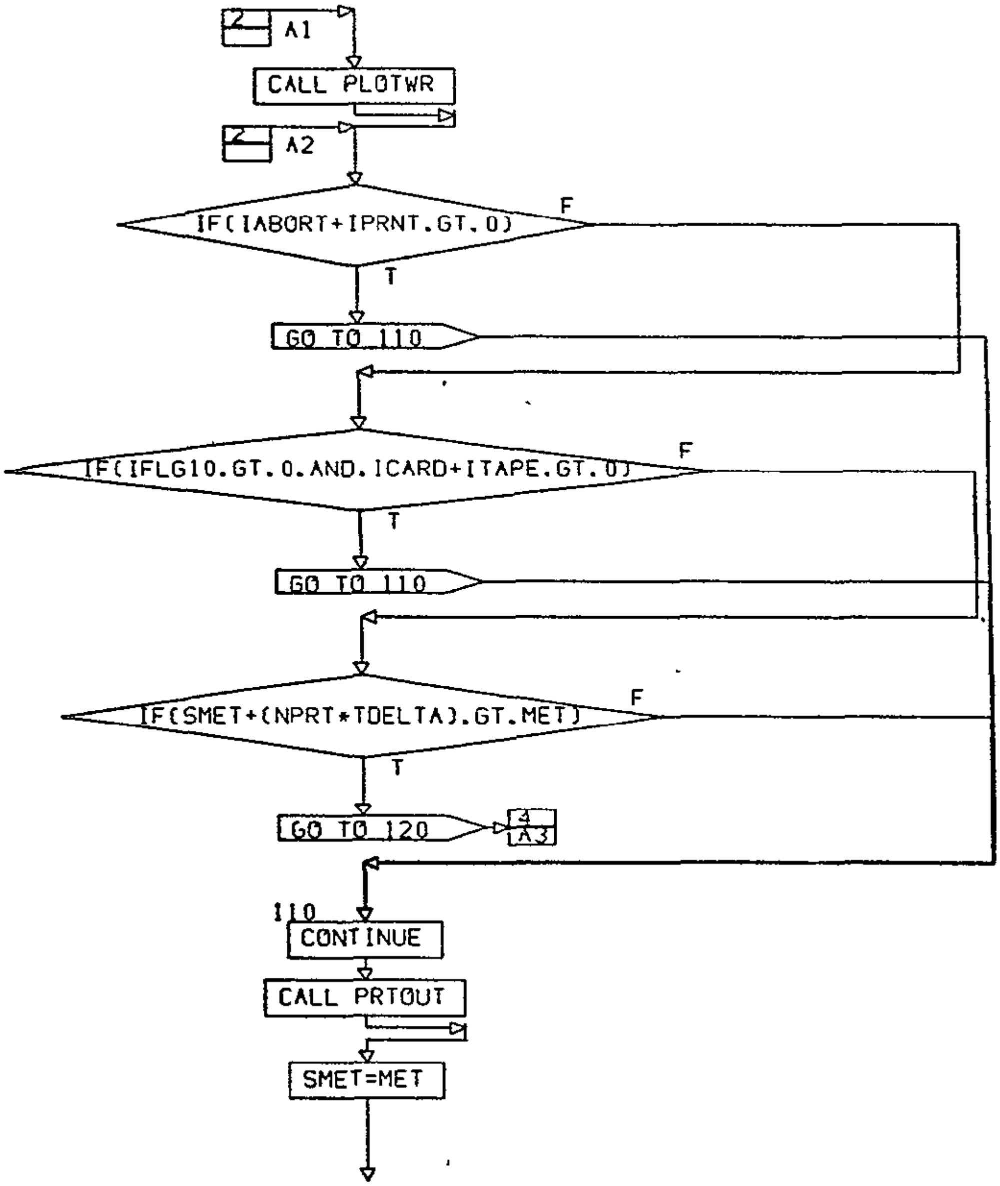


FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED).



CONT. ON PG 4

PHASE2
PG 3 OF 8

FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)

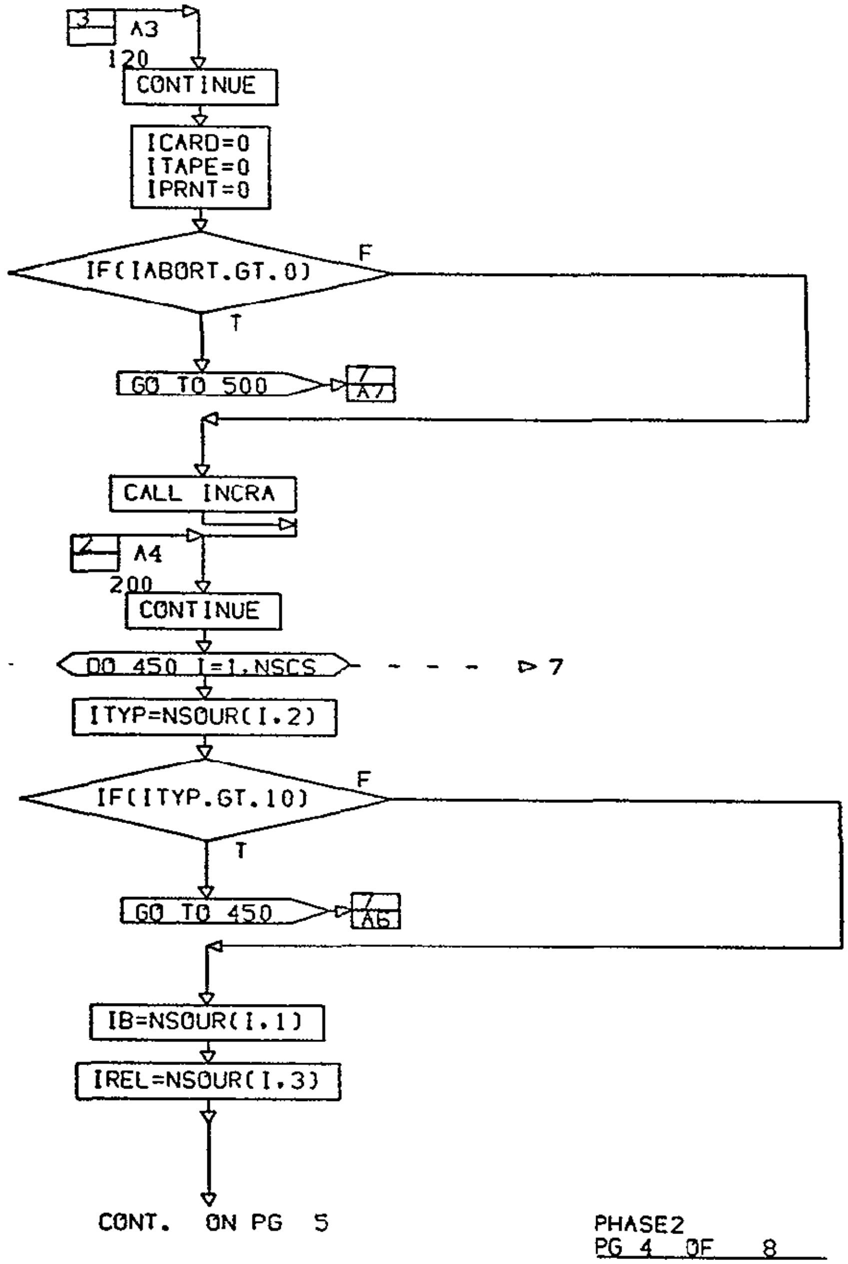
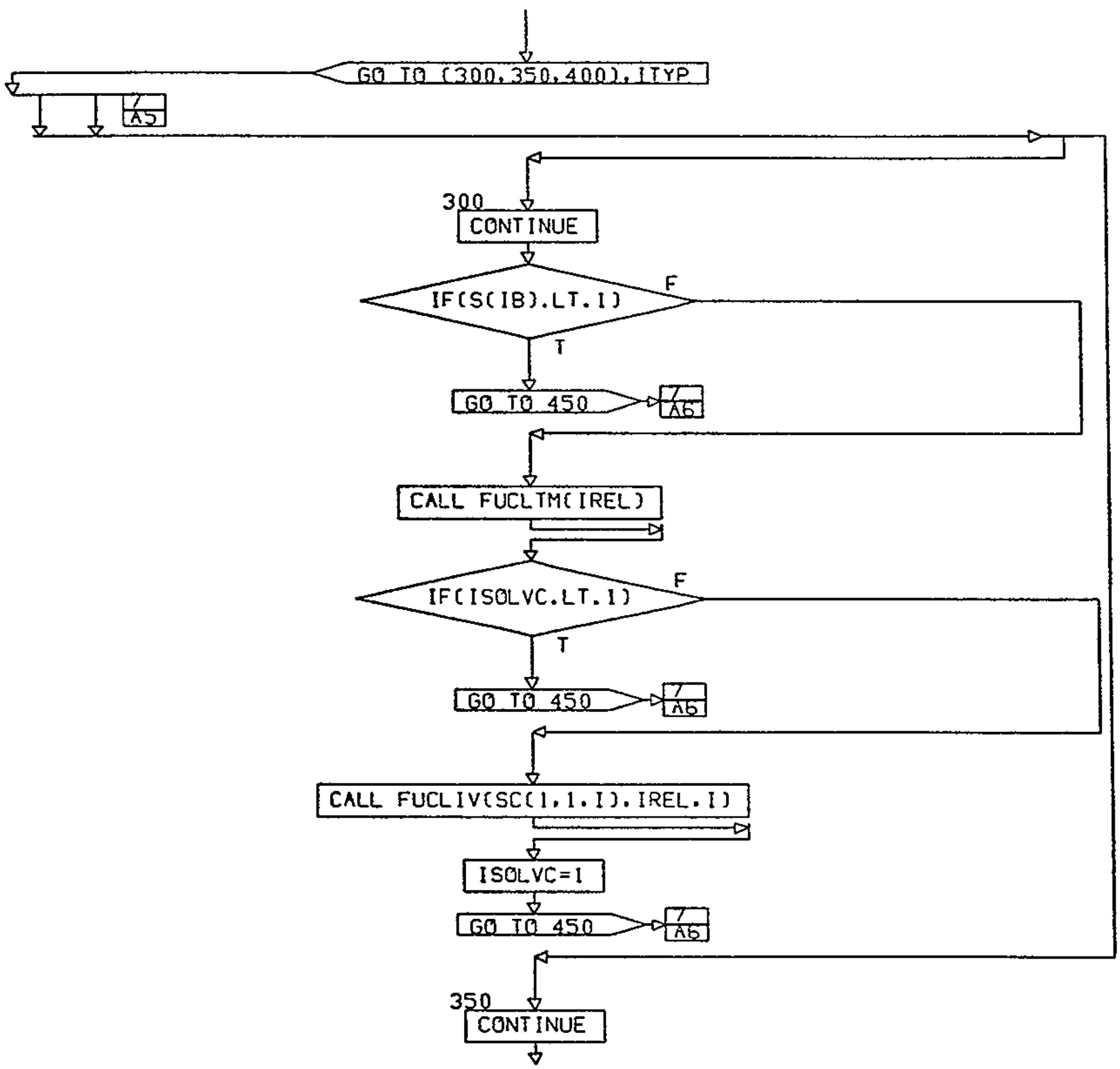


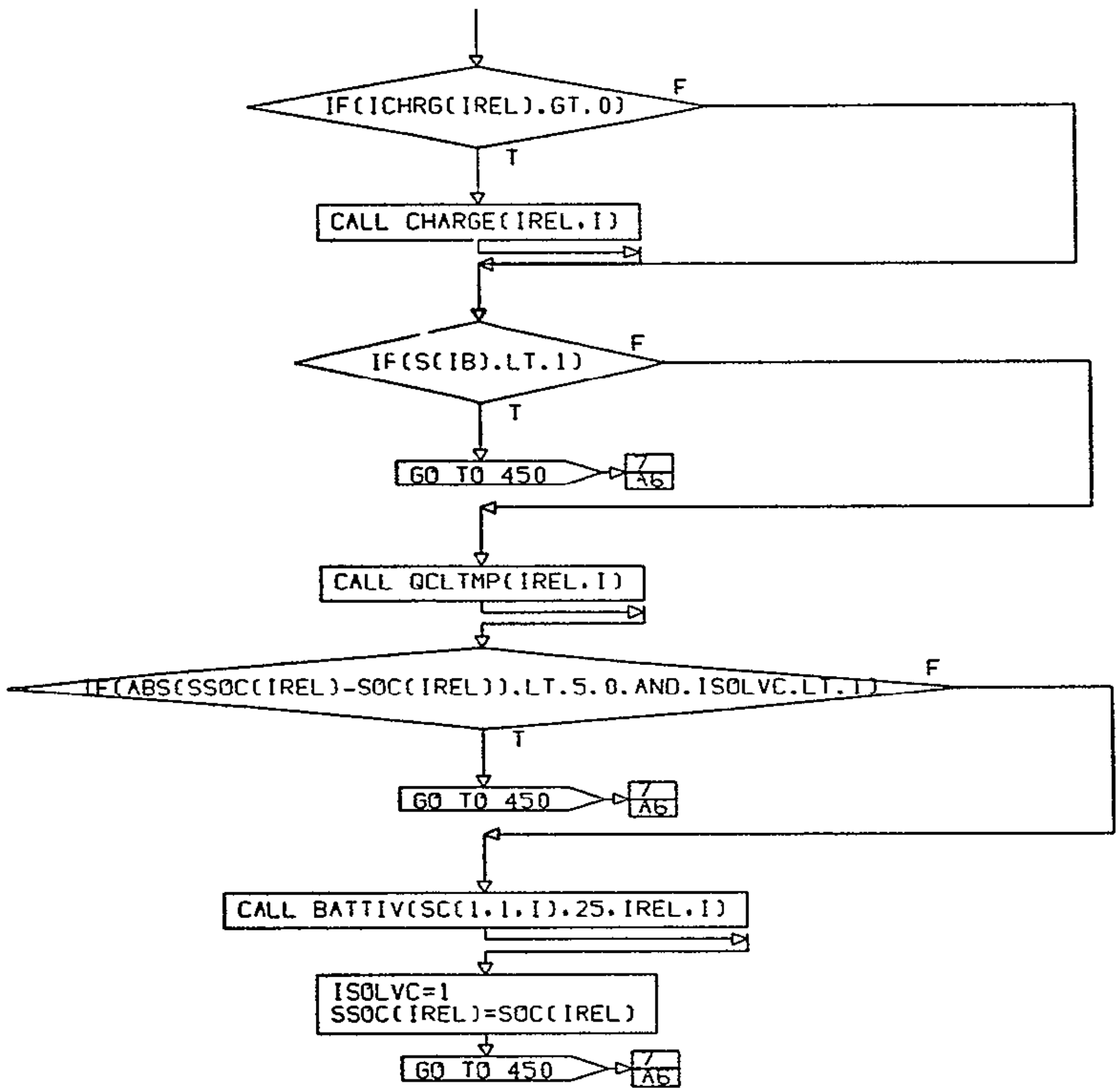
FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)



CONT. ON PG 6

PHASE2
PG 5 OF 8

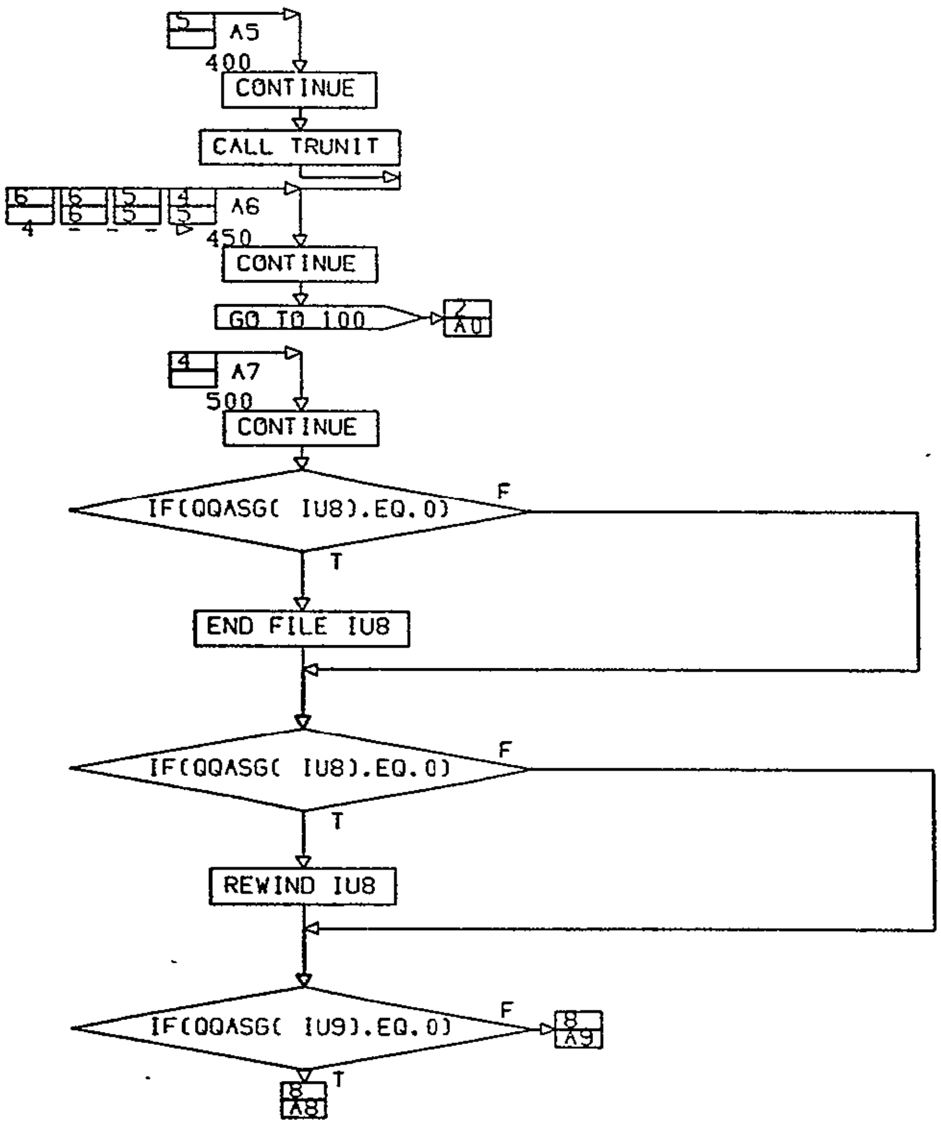
FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)



CONT. ON PG 7

PHASE2
PG 6 OF 8

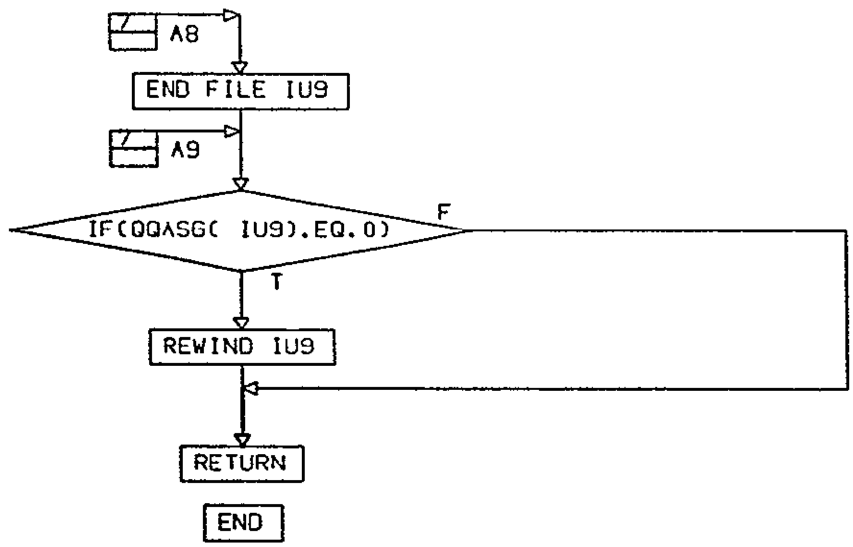
FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)



CONT. ON PG 8

PHASE2
PG 7 OF 8

FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)



PHASE2
PG 8 FINAL

FIGURE 3.3.1. FUNCTIONAL FLOWCHART OF SUBROUTINE PHASE2 (CONTINUED)

3.3.2 Subroutine: ACINVT

PURPOSE: To simulate the operation of the onboard dc-ac inverters

METHOD: The ac load and power factor for each inverter is calculated. These values are used to calculate the equivalent dc load and inverter efficiency. If the inverter is not carrying an ac load or the ac load it is carrying is less than the input dc no-load value, the inverter is set to dc no-load value.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.2 See Appendix for definition of all variables.

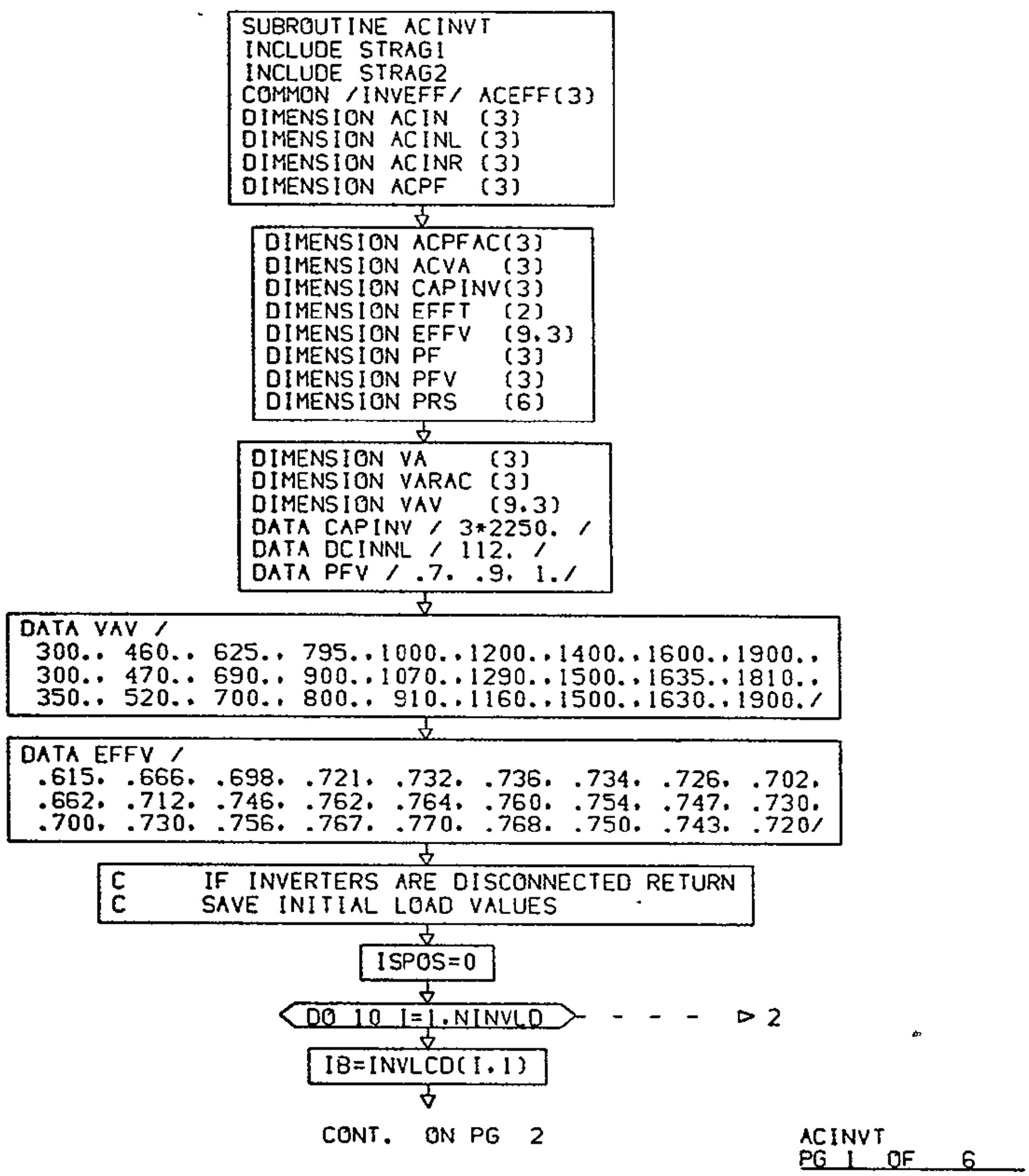


FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT

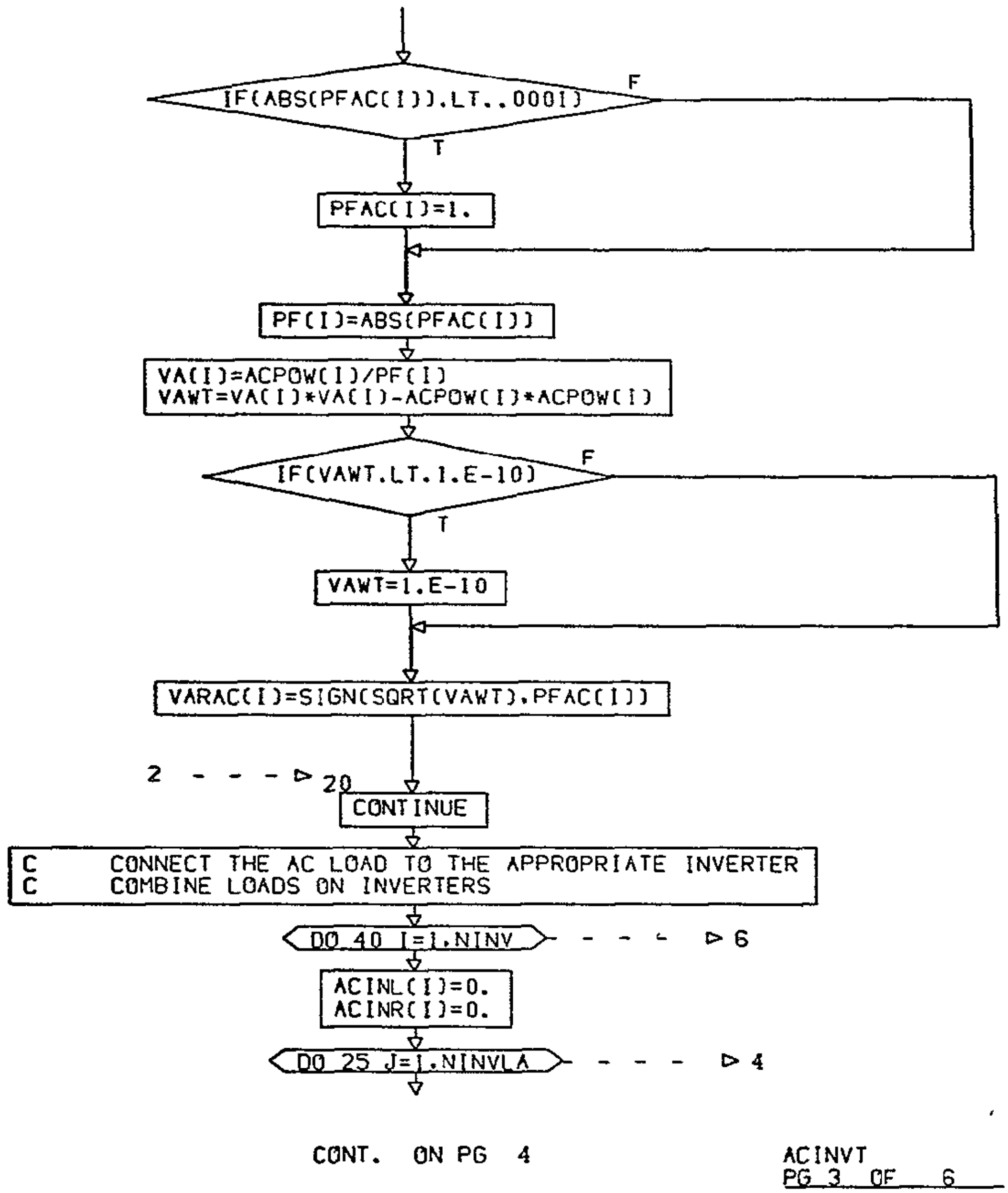


FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT (CONTINUED)

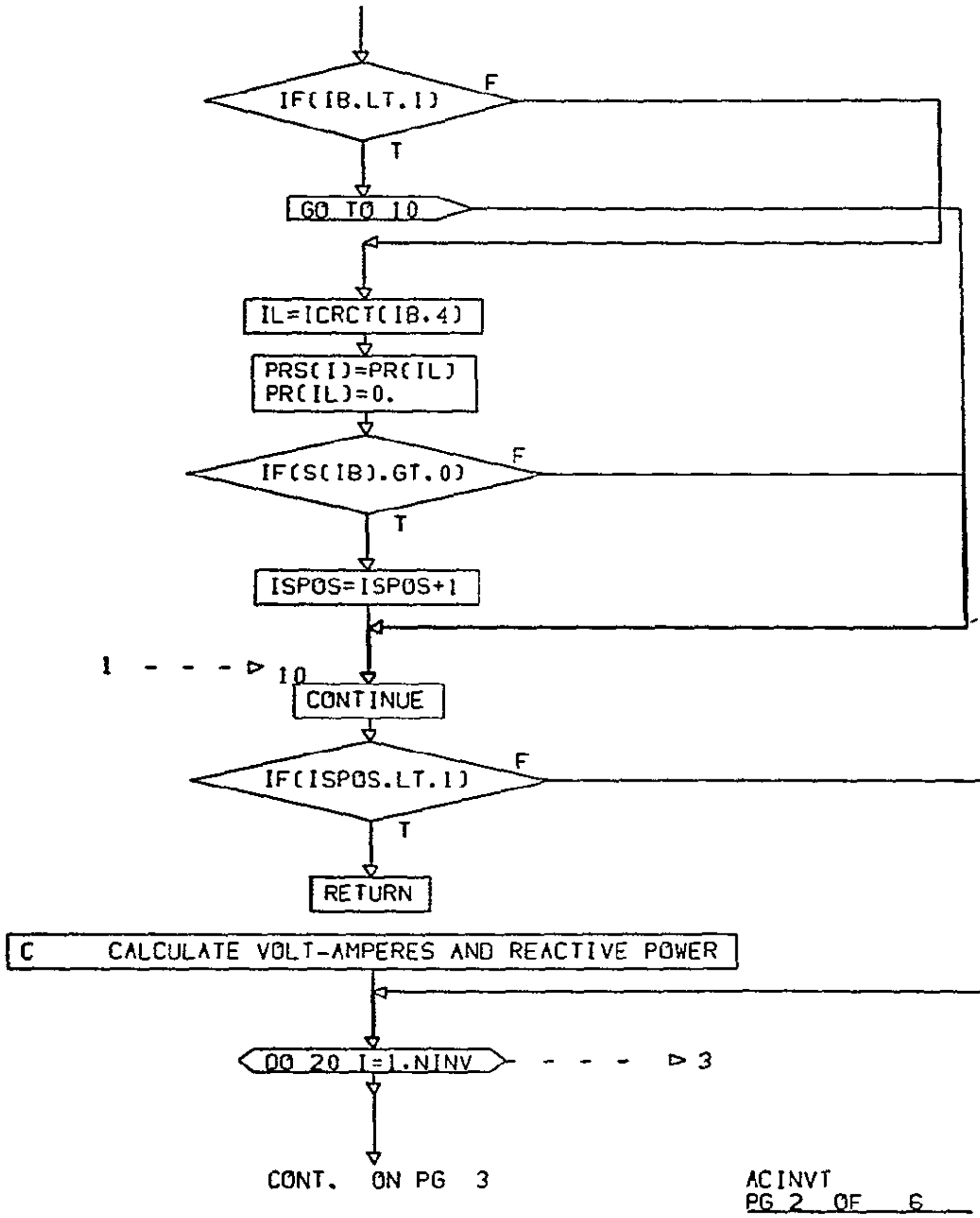


FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT (CONTINUED)

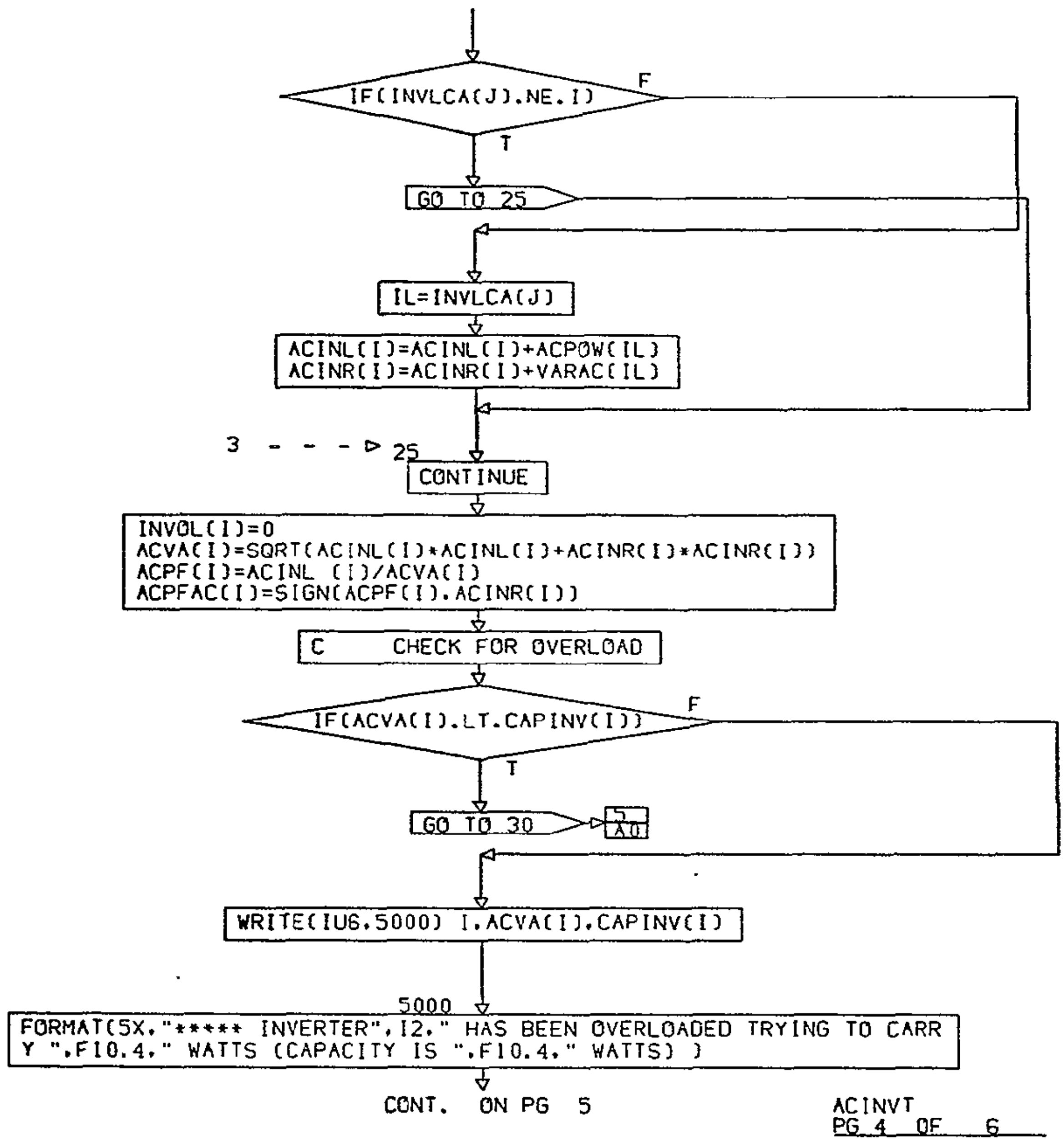
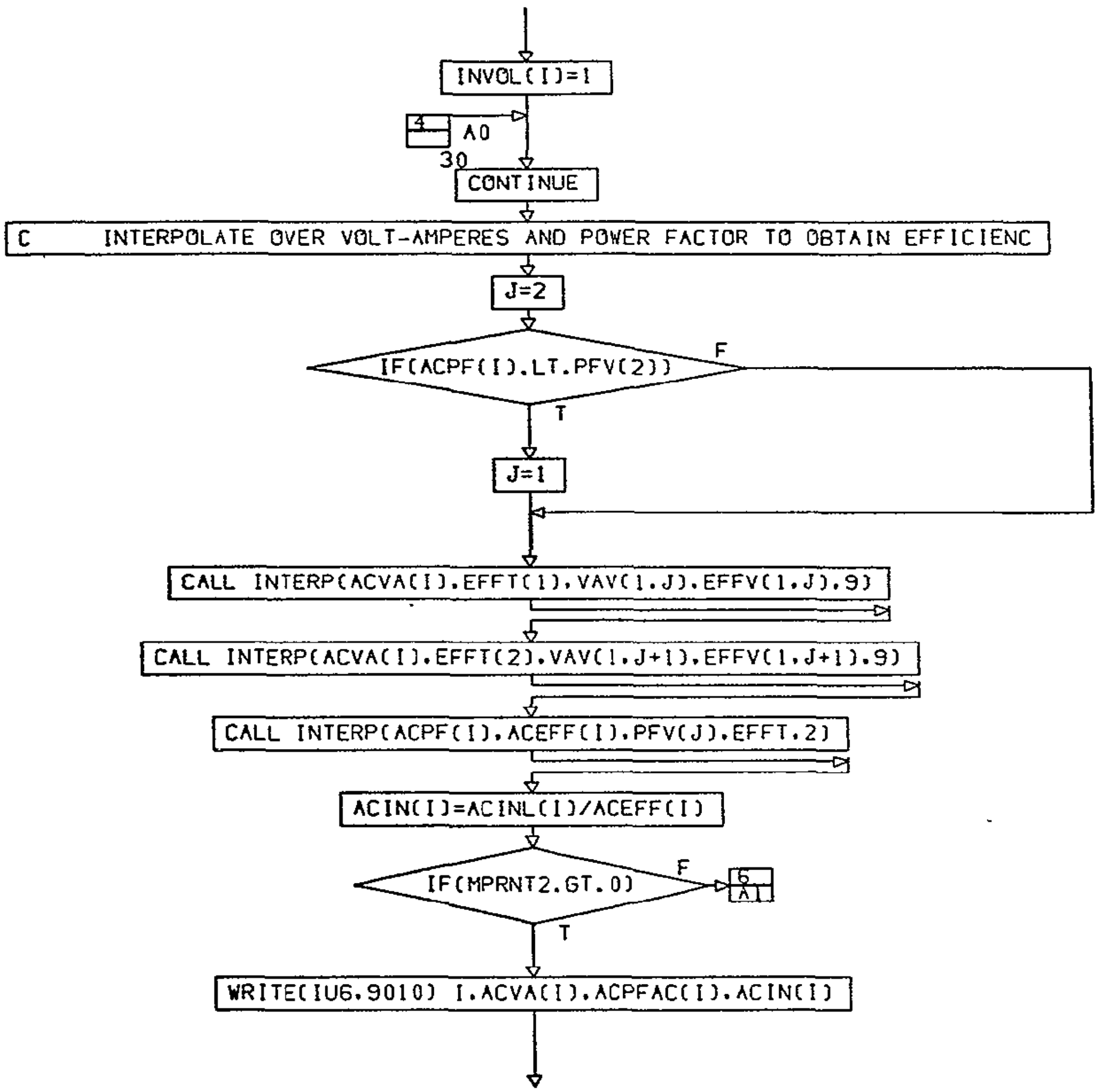


FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT (CONTINUED)



CONT. ON PG 6

ACINVT
PG 5 OF 6

FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT (CONTINUED)

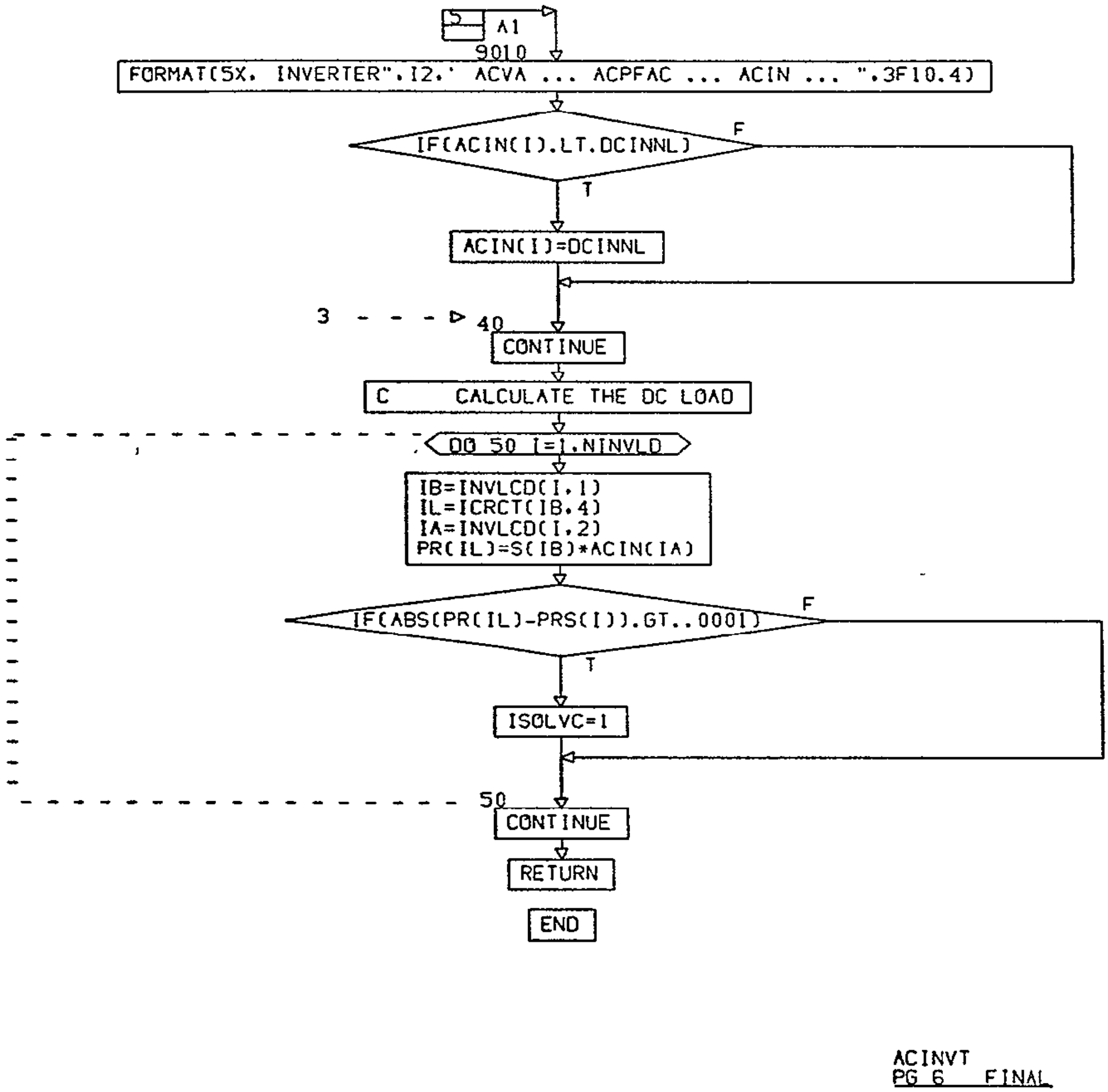


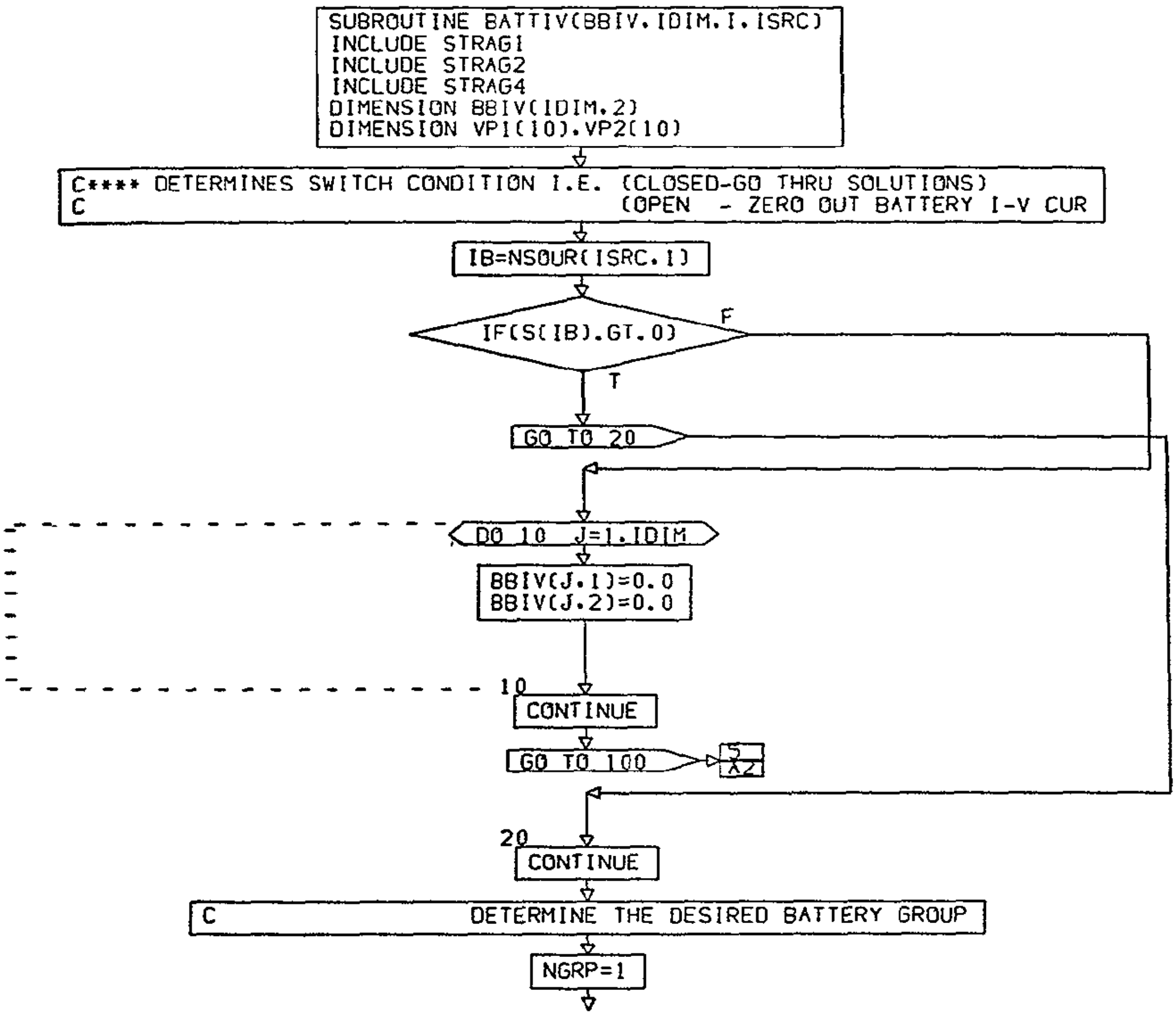
FIGURE 3.3.2. FUNCTIONAL FLOWCHART OF SUBROUTINE ACINVT (CONTINUED)

3.3.3 Subroutine: BATTIV

PURPOSE: To create the I-V curves used in the onboard battery simulations.

METHOD: After determining the type of battery to be simulated, an interpolation is made into a set of current-voltage curves as a function of temperature to determine the battery I-V curves at its operating temperature.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.3. See Appendix for definition of all variables.

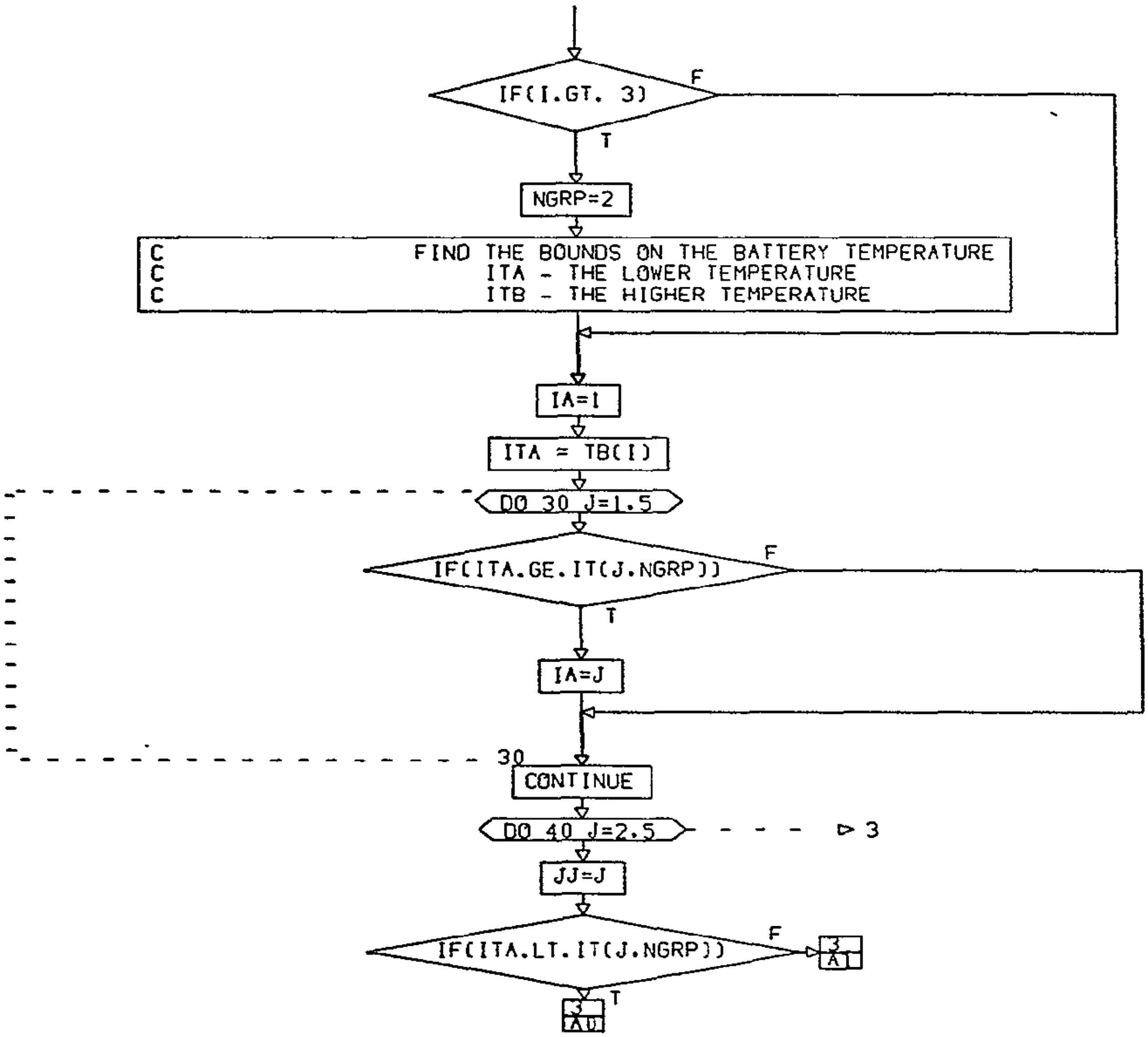


CONT. ON PG 2

BATTIV
PG 1 OF 5

FIGURE 3.3.3. FUNCTIONAL FLOWCHART OF SUBROUTINE BATTIV

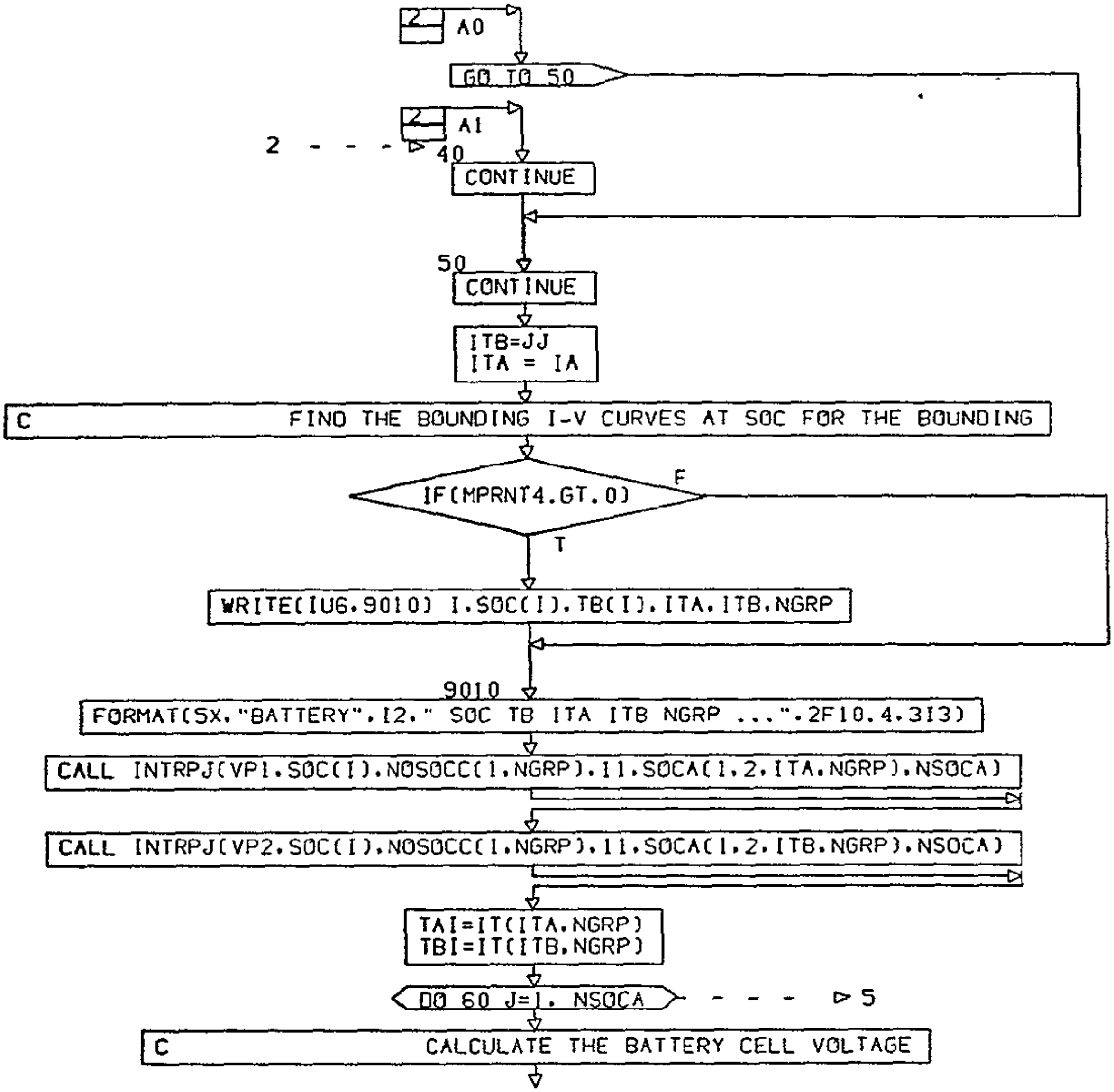
ORIGINAL PAGE IS
OF POOR QUALITY



CONT. ON PG 3

BATTIV
PG 2 OF 5

FIGURE 3.3.3. FUNCTIONAL FLOWCHART OF SUBROUTINE BATTIV (CONTINUED)



CONT. ON PG 4

BATTIV
PG 3 OF 5

FIGURE 3.3.3. FUNCTIONAL FLOWCHART OF SUBROUTINE BATTIV (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

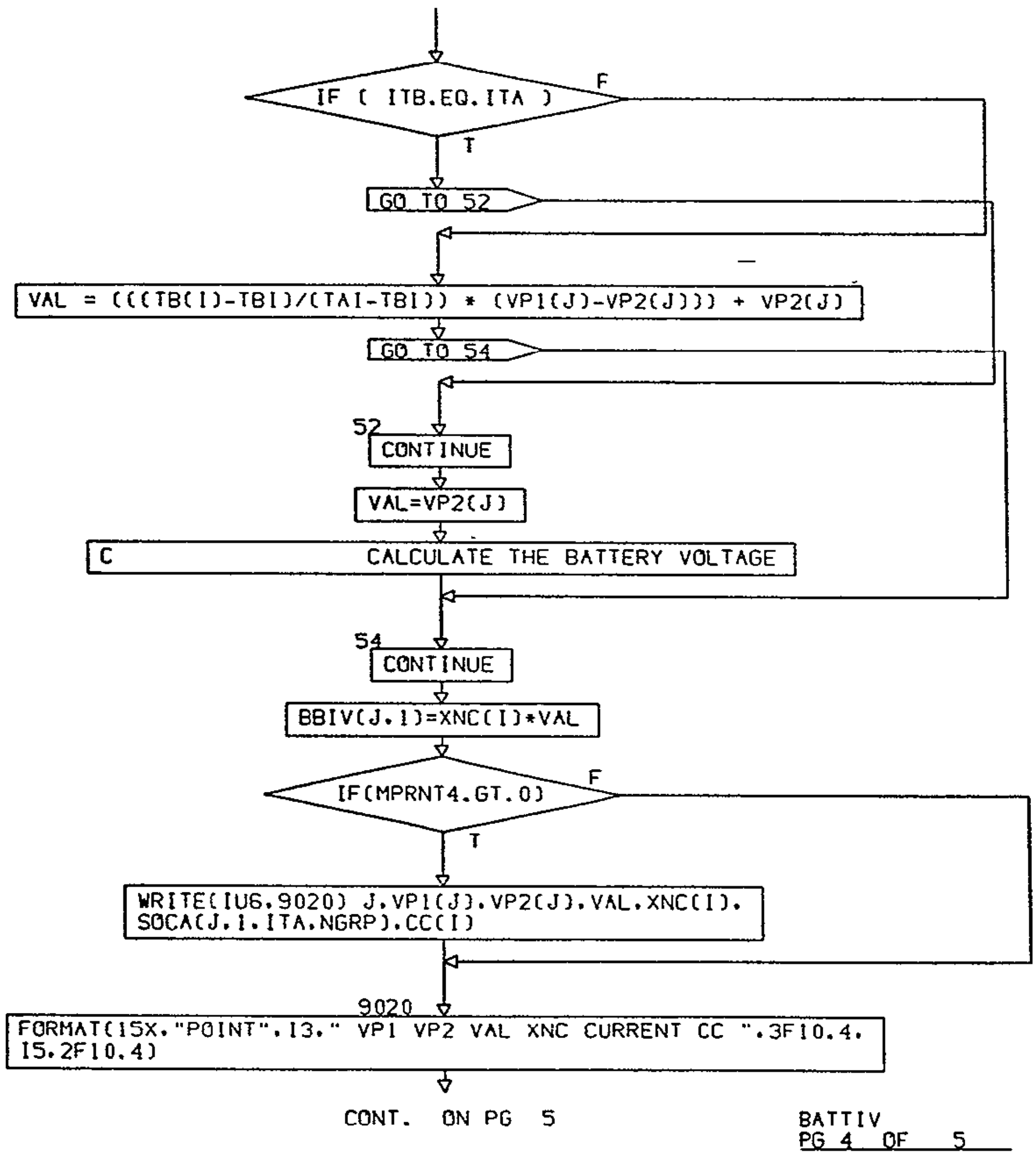


FIGURE 3.3.3. FUNCTIONAL FLOWCHART OF SUBROUTINE BATTIV (CONTINUED)

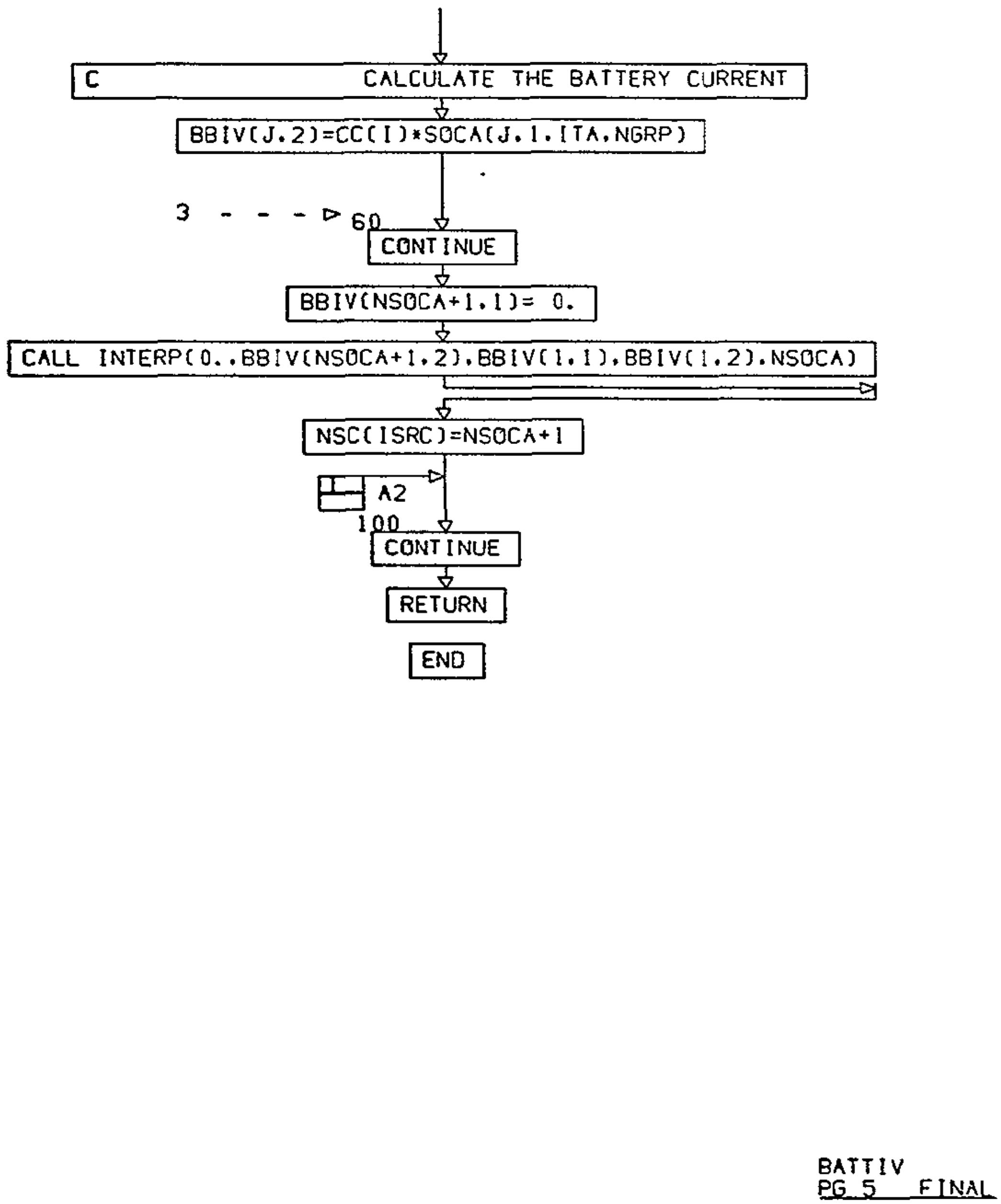


FIGURE 3.3.3. FUNCTIONAL FLOWCHART OF SUBROUTINE BATTIV (CONTINUED)

3.3.4 Subroutine: CHARGE

PURPOSE: To simulate the operation of the battery charger.

METHOD: A charge curve is calculated based on depth-of-discharge and time since the last discharge. The curve is rate of charge versus time on charge. When the battery state-of-charge reaches 100 percent, the battery is removed from the charger.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.4. See Appendix for definition of all variables.

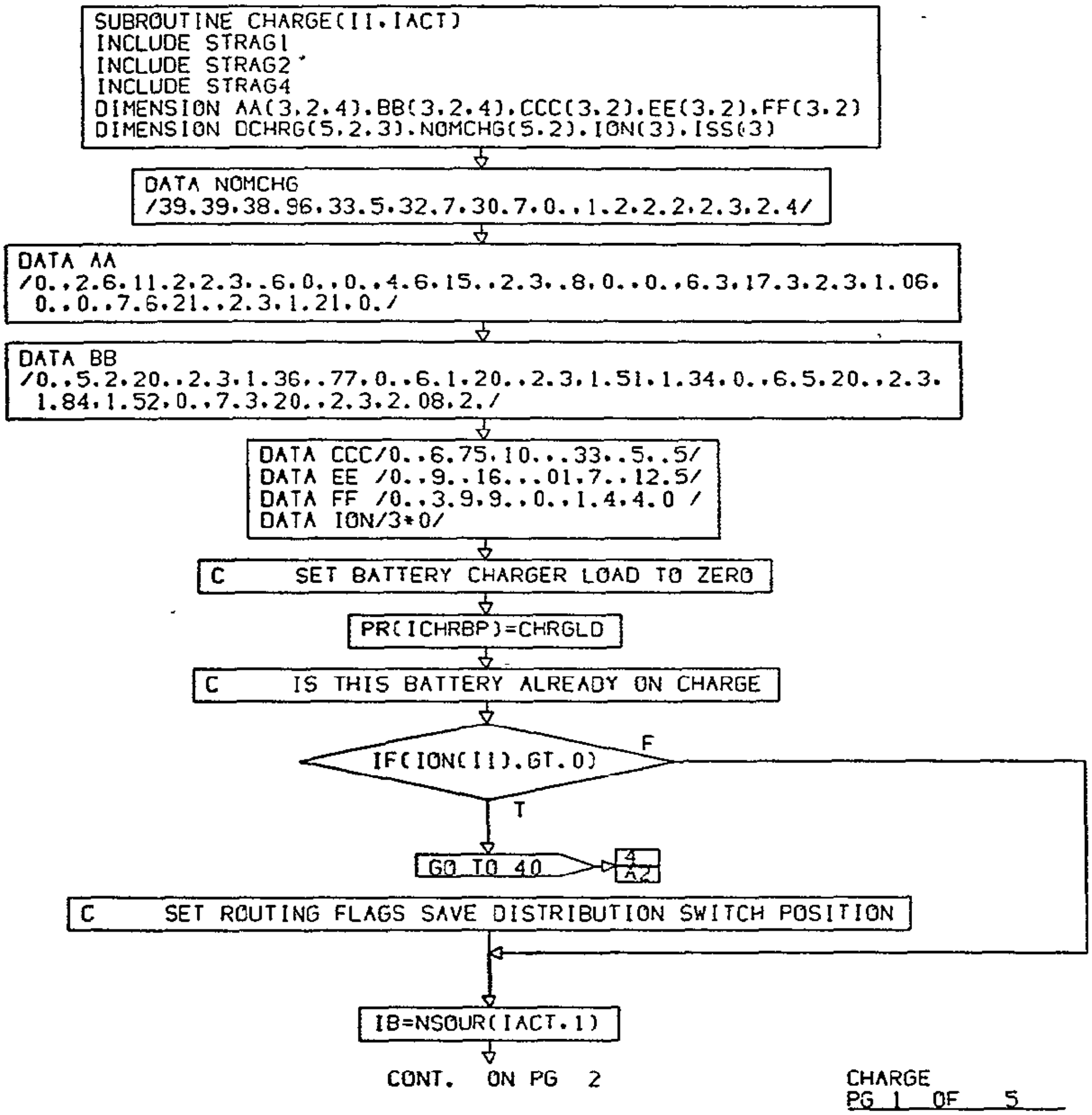
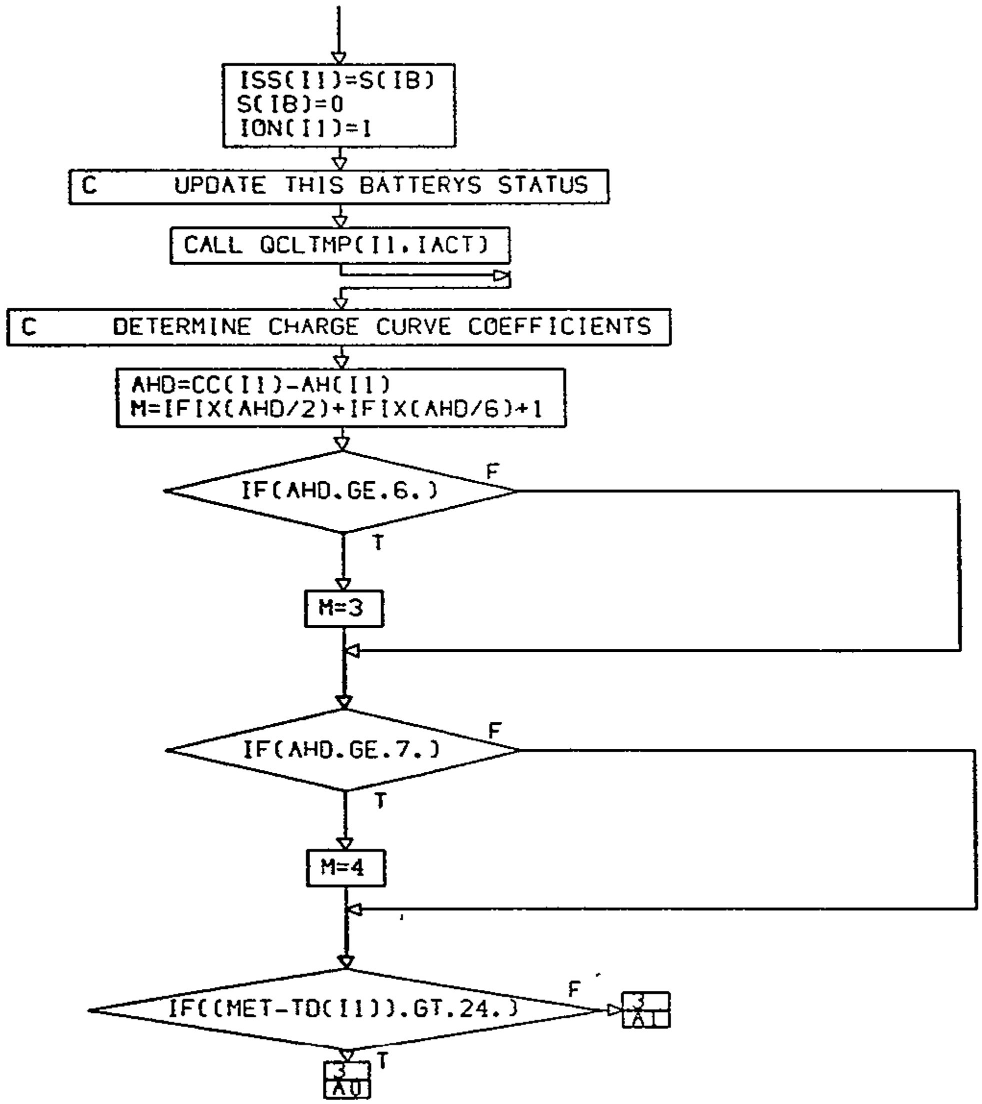


FIGURE 3.3.4. FUNCTIONAL FLOWCHART OF SUBROUTINE CHARGE



CONT. ON PG 3

CHARGE
PG 2 OF 5

FIGURE 3.3.4. FUNCTIONAL FLOWCHART OF SUBROUTINE CHARGE (CONTINUED)

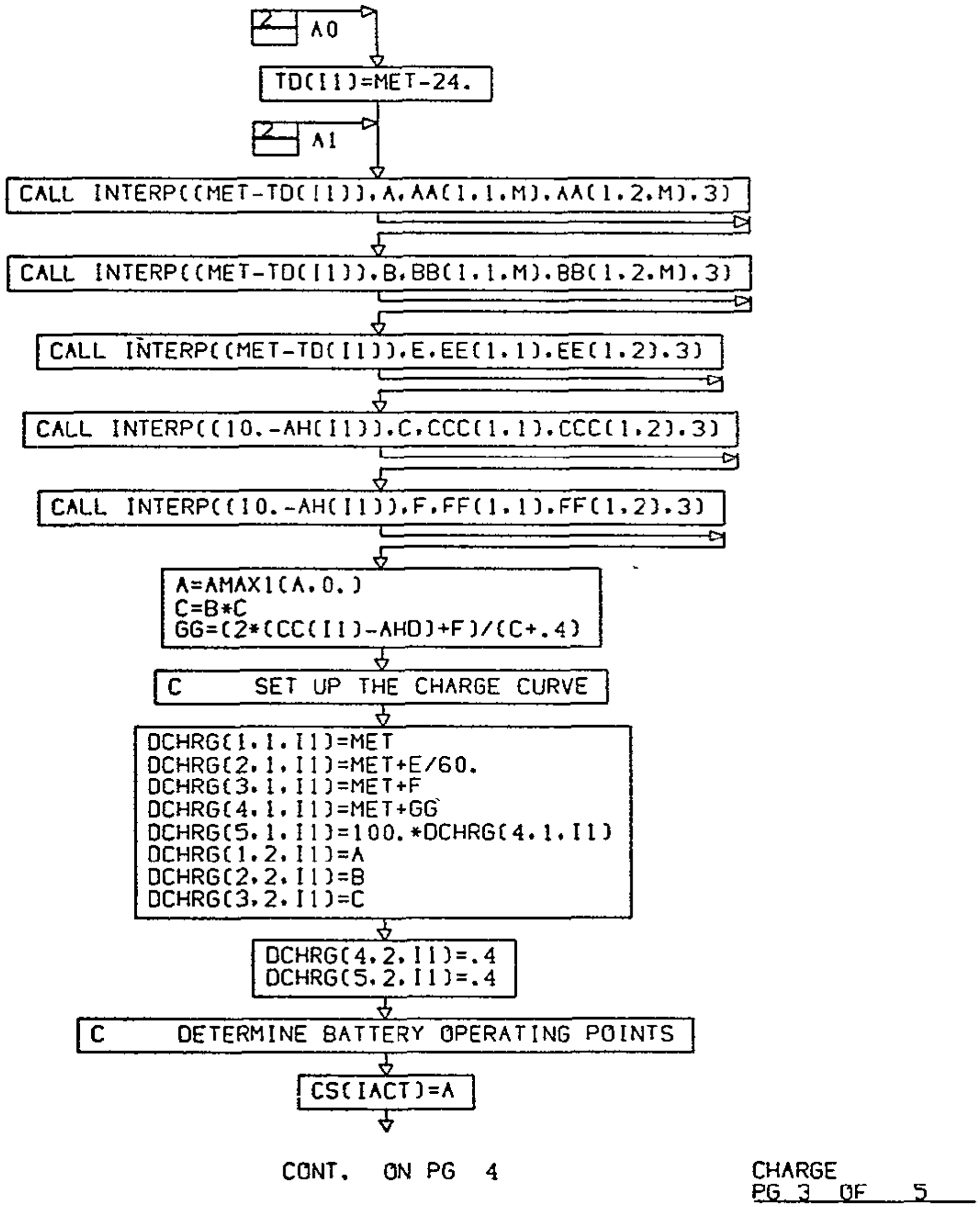
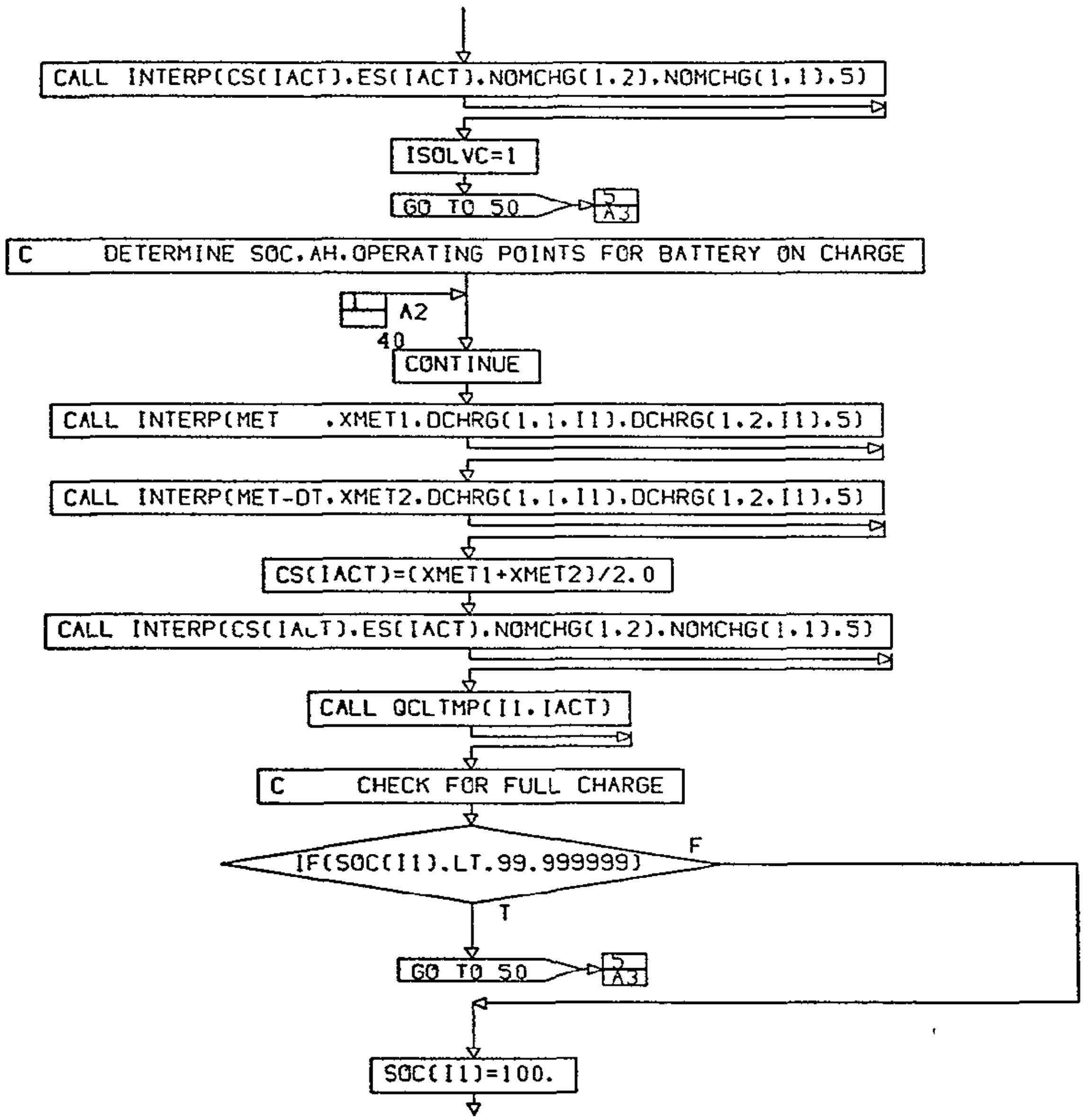


FIGURE 3.3.4. FUNCTIONAL FLOWCHART OF SUBROUTINE CHARGE (CONTINUED)



CONT. ON PG 5

CHARGE
PG 4 OF 5

FIGURE 3.3.4. FUNCTIONAL FLOWCHART OF SUBROUTINE CHARGE (CONTINUED)

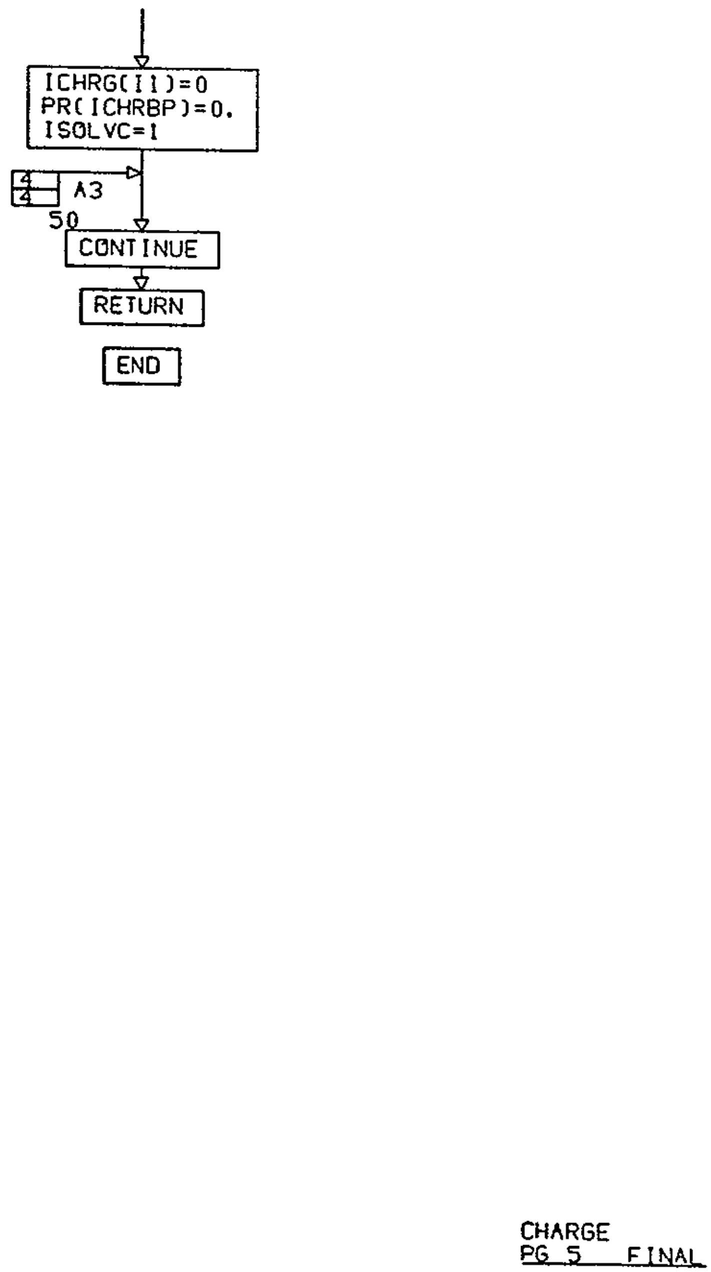


FIGURE 3.3.4. FUNCTIONAL FLOWCHART OF SUBROUTINE CHARGE (CONTINUED)

3.3.5 Subroutine: DCSOLV

PURPOSE: To determine the operating characteristics of the EPDC.

METHOD: Utilizing a user provided distribution circuit this routine calculates the nodal equations of the circuit and solves these equations by means of the Gauss-Jordan method. This process is done iteratively until the change-in node voltages are less than some input value.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.5. See Appendix for definition of all variables.

C.3

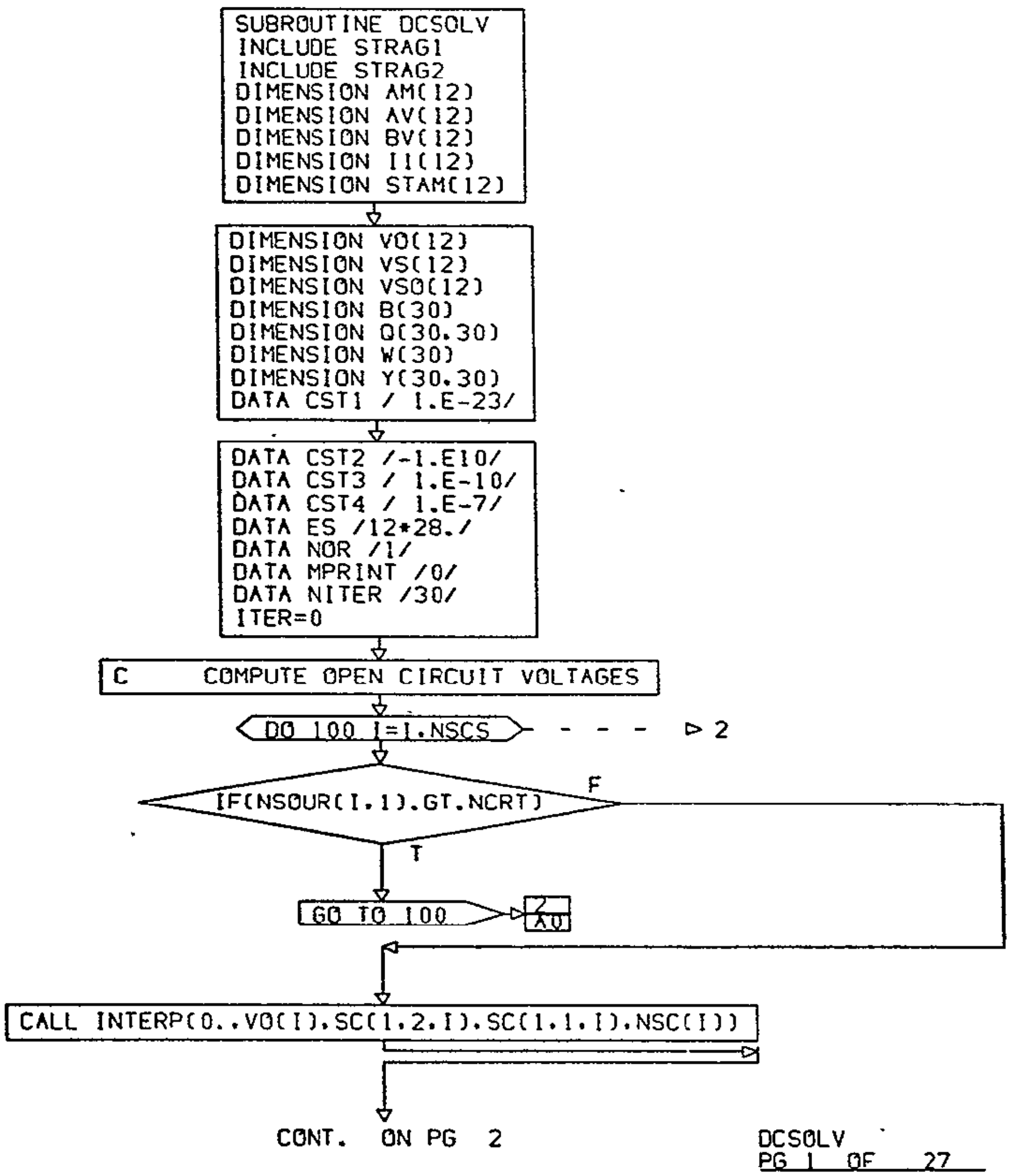
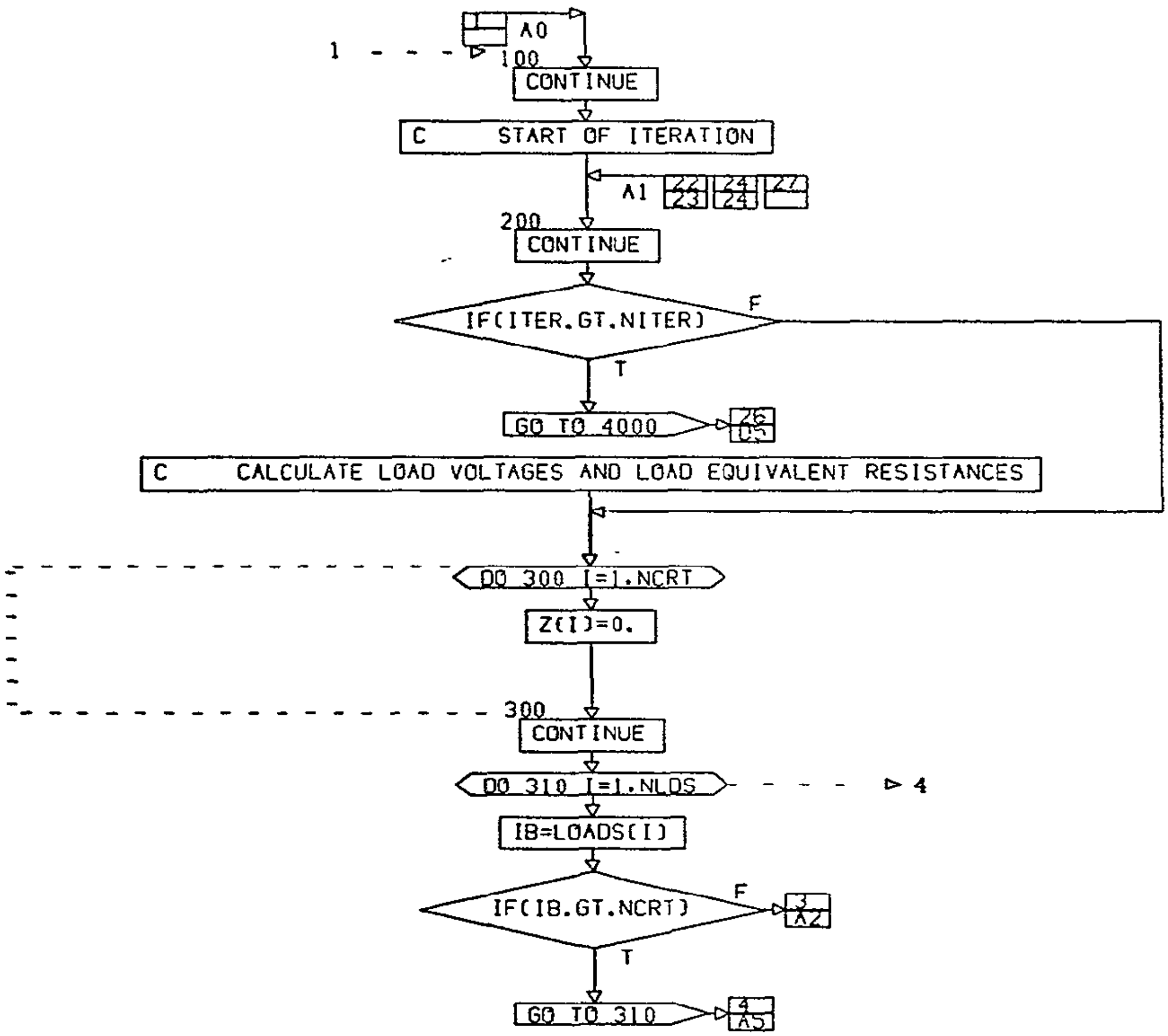


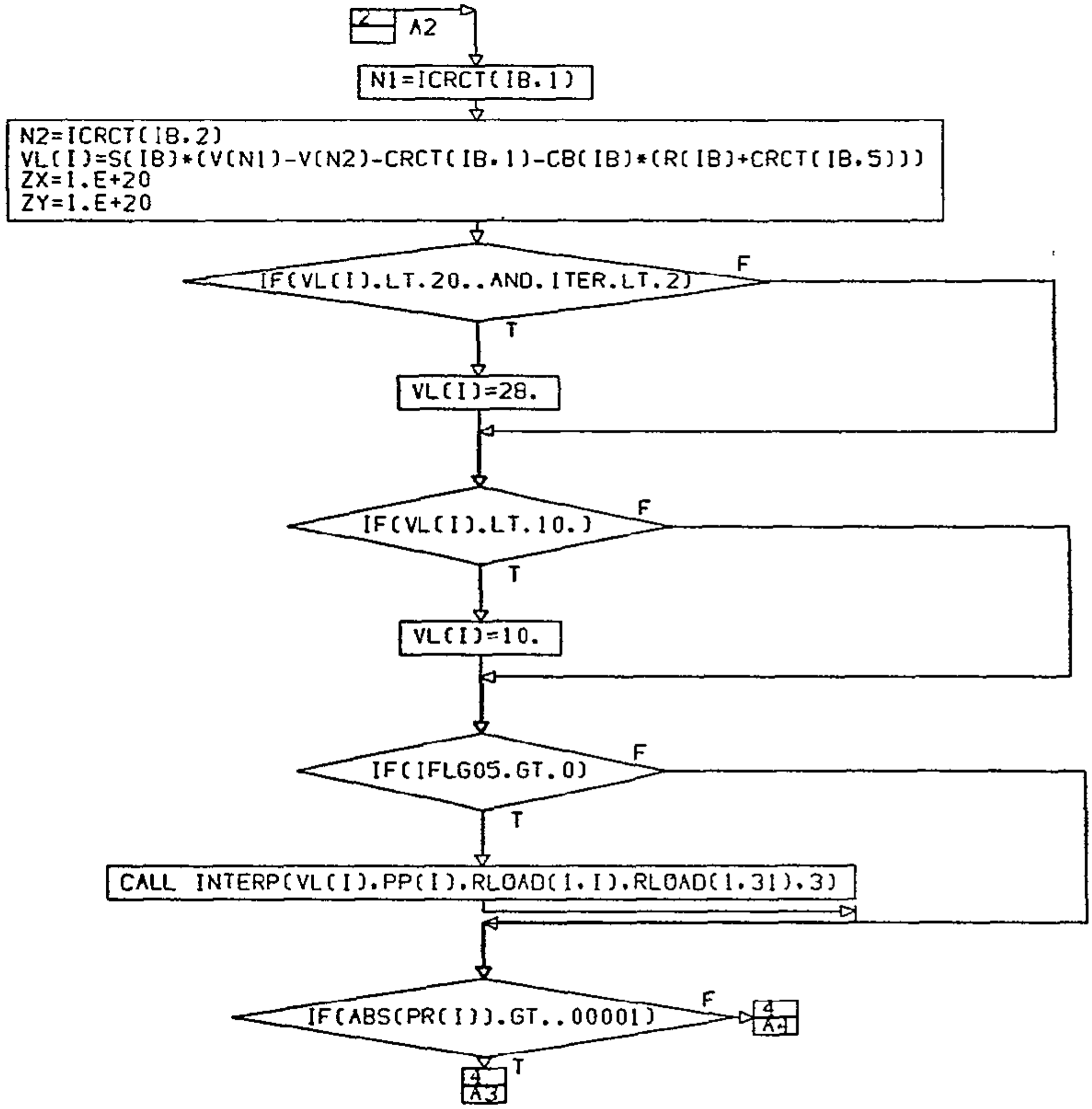
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV



CONT. ON PG 3

DCSOLV
PG 2 OF 27

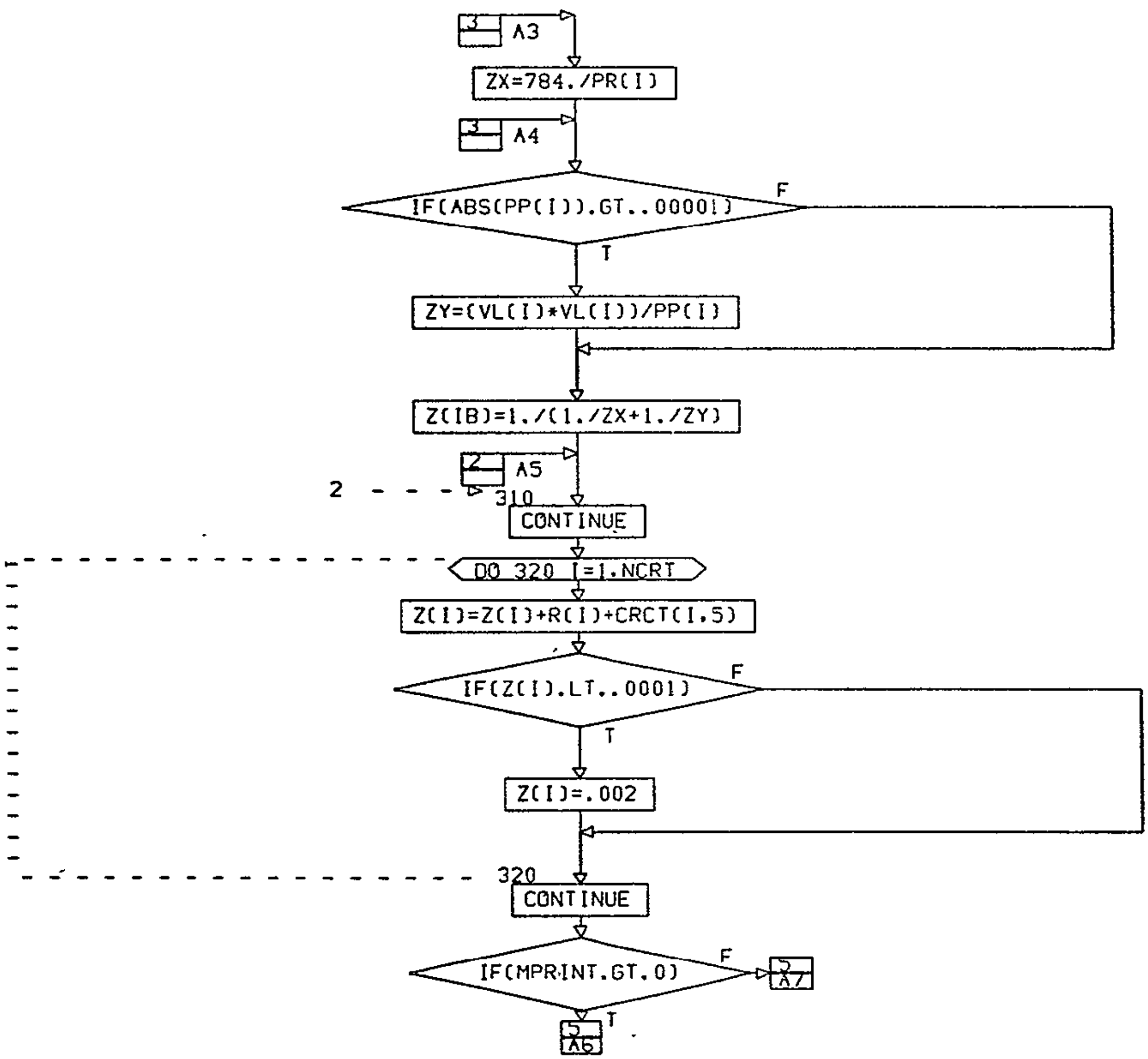
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 4

DCSOLV
PG 3 OF 27

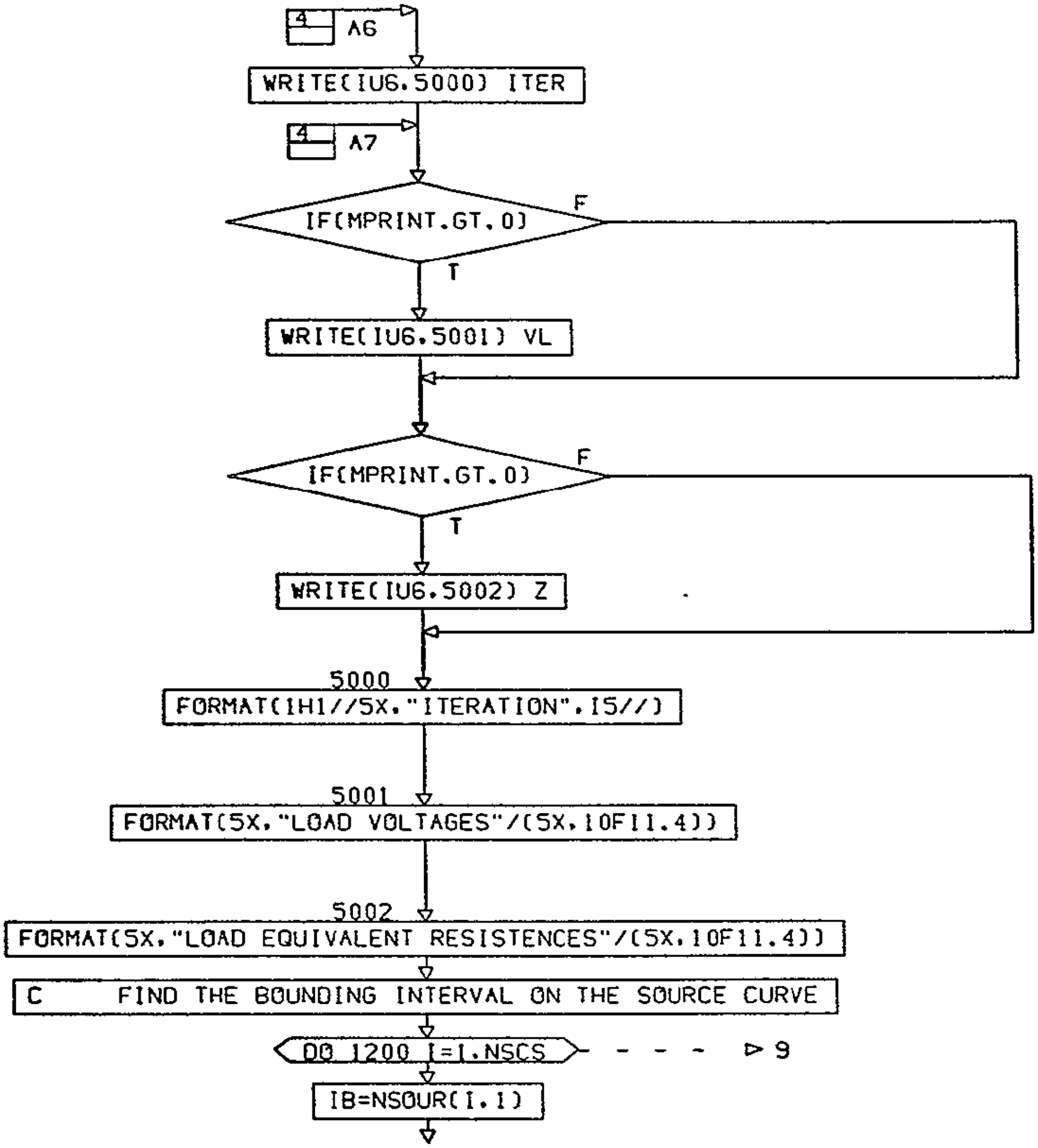
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 5

DCSOLV
PG 4 OF 27

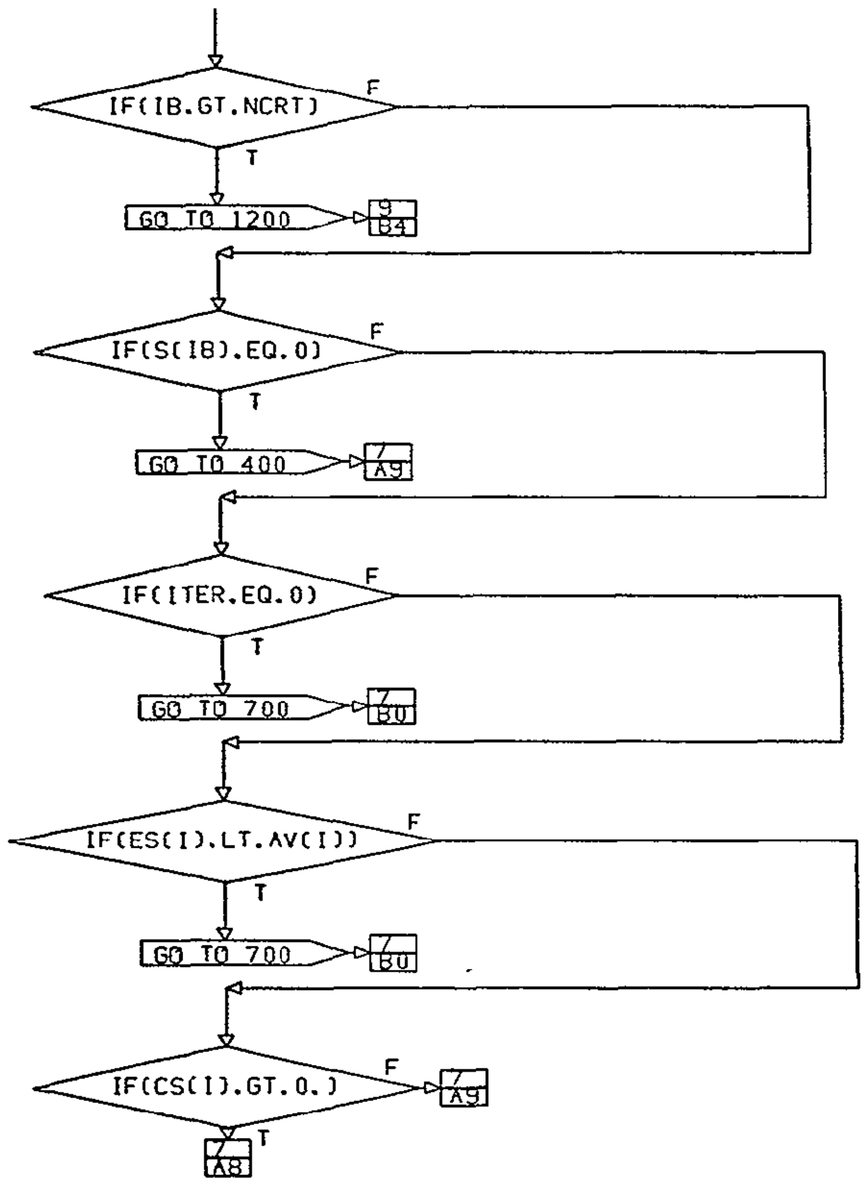
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 6

DCSOLV
PG 5 OF 27

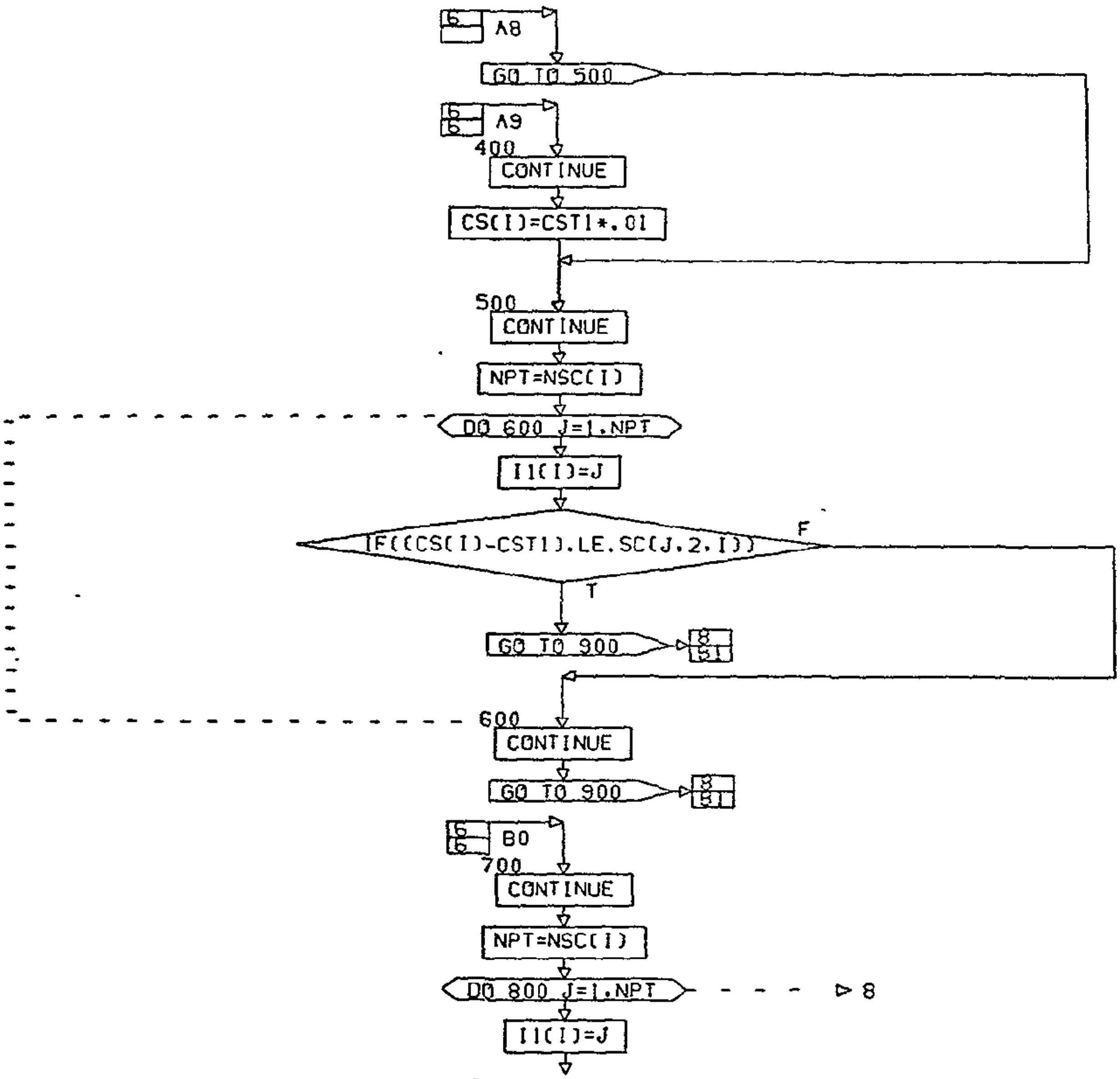
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 7

DCSOLV
PG 6 OF 27

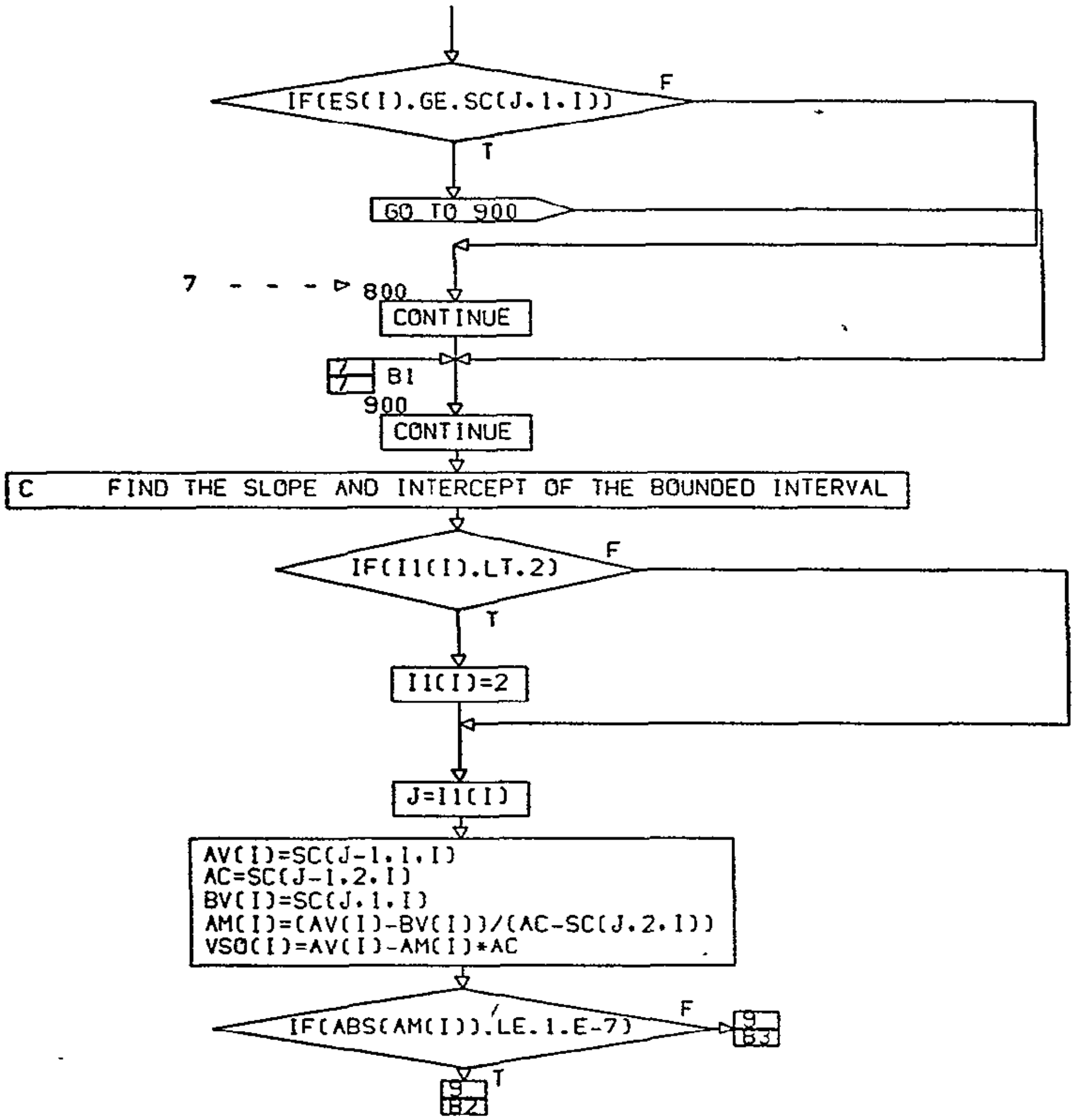
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 8

DCSOLV
PG 7 OF 27

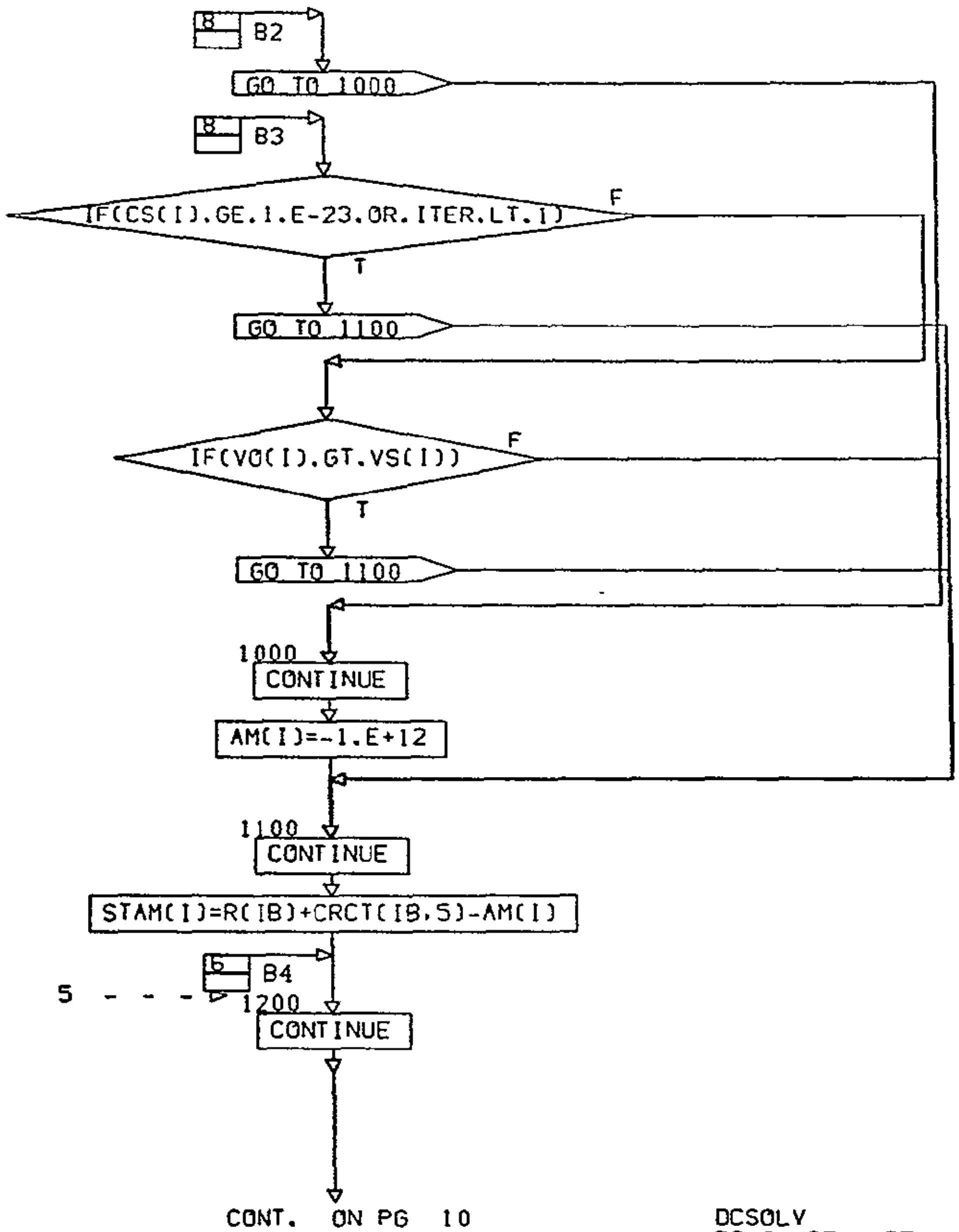
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 9

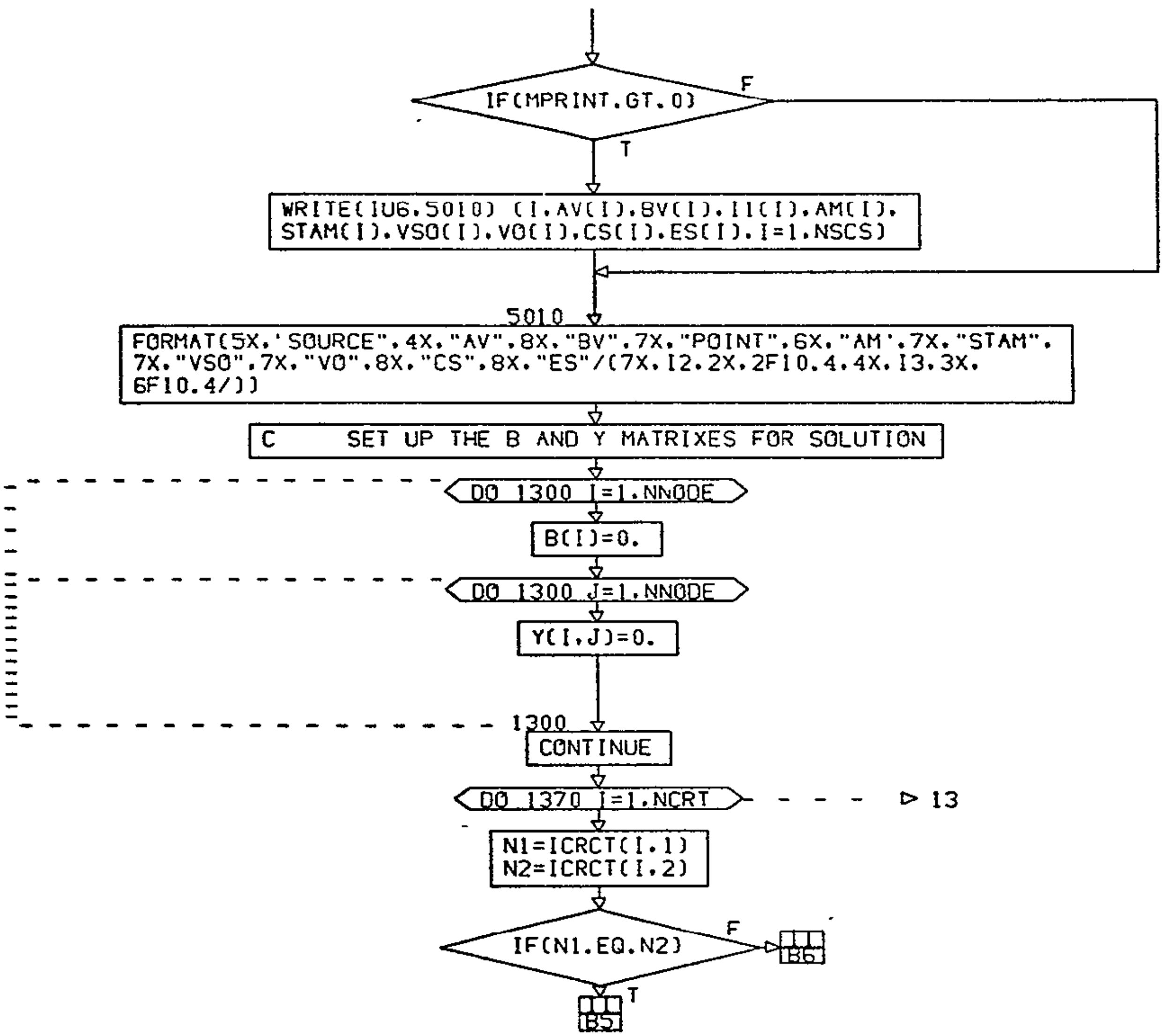
DCSOLV
PG 8 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



DCSOLV
PG 9 OF 27

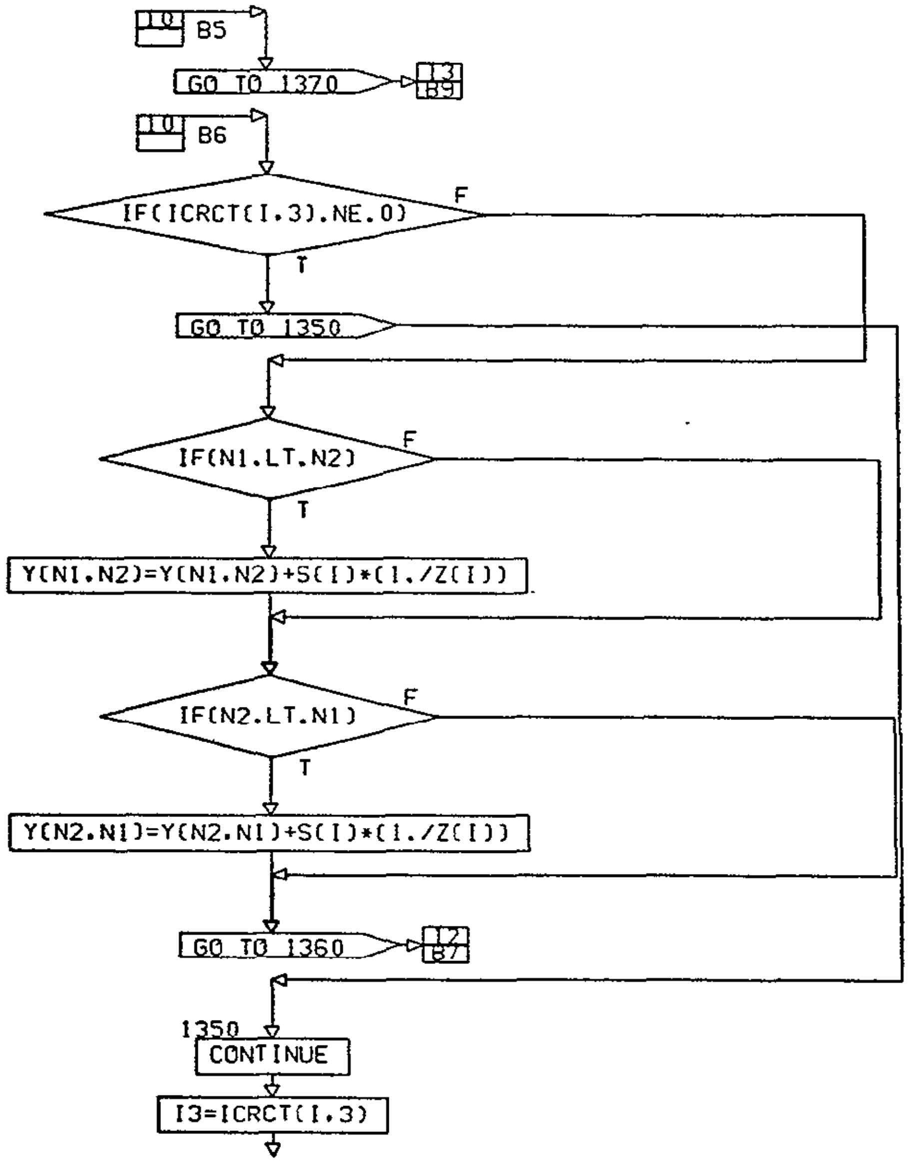
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 11

DCSOLV
PG 10 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 12

DCSOLV
PG 11 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

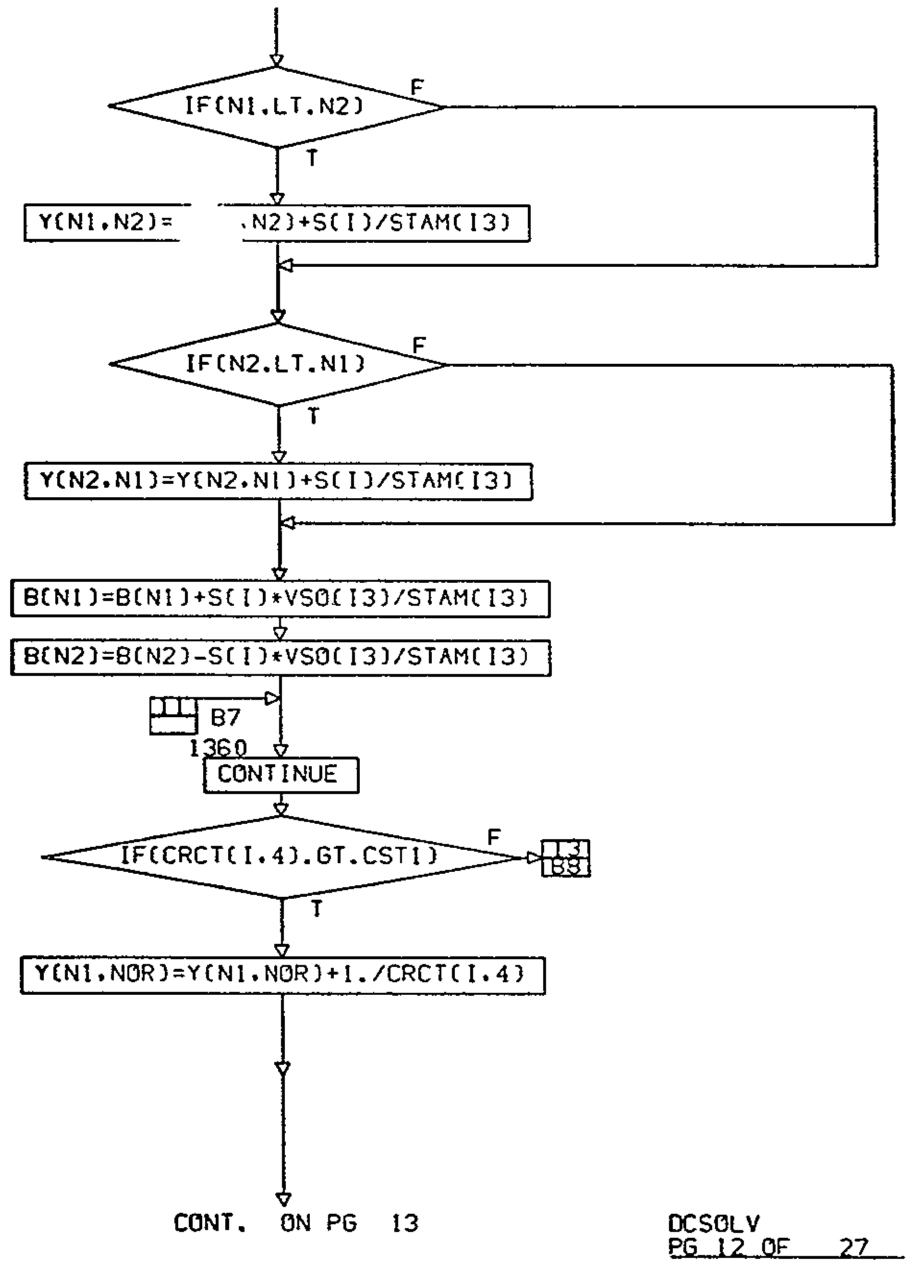
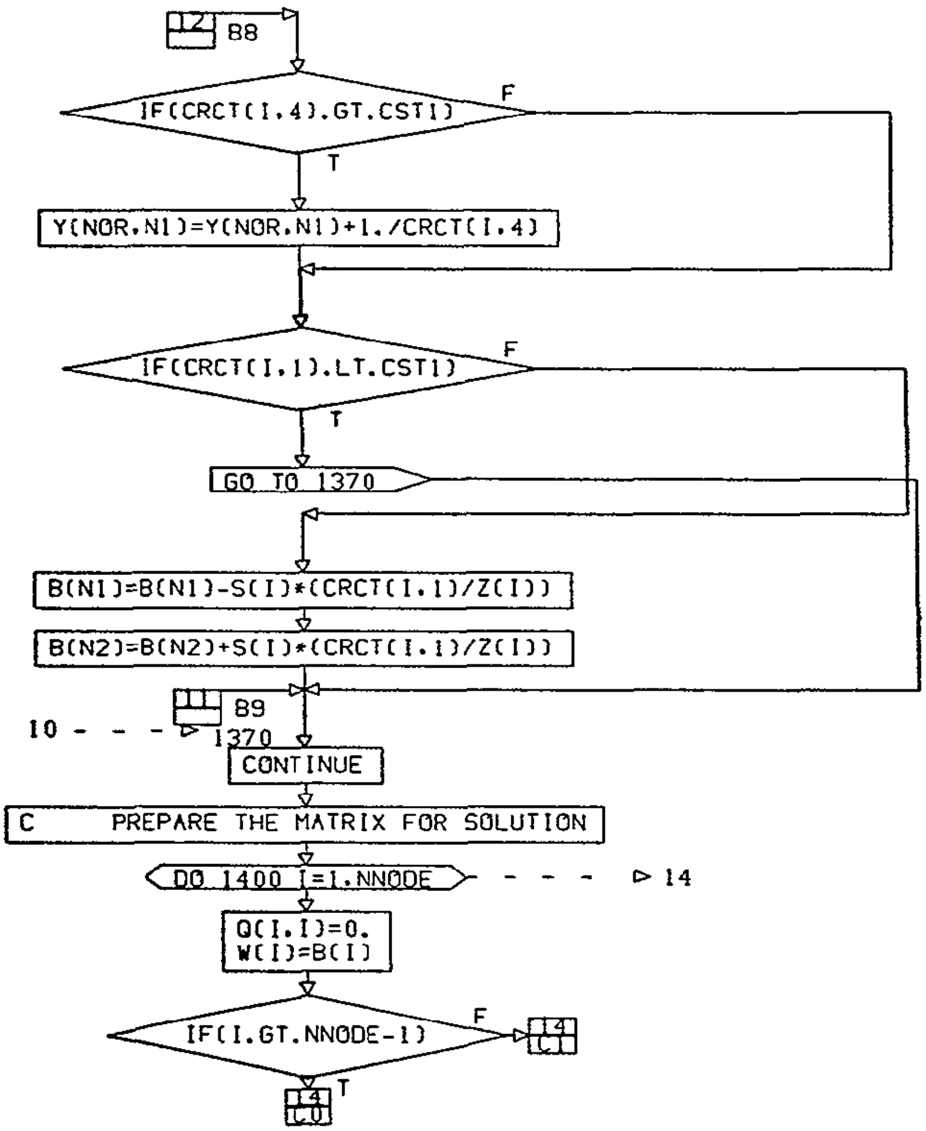


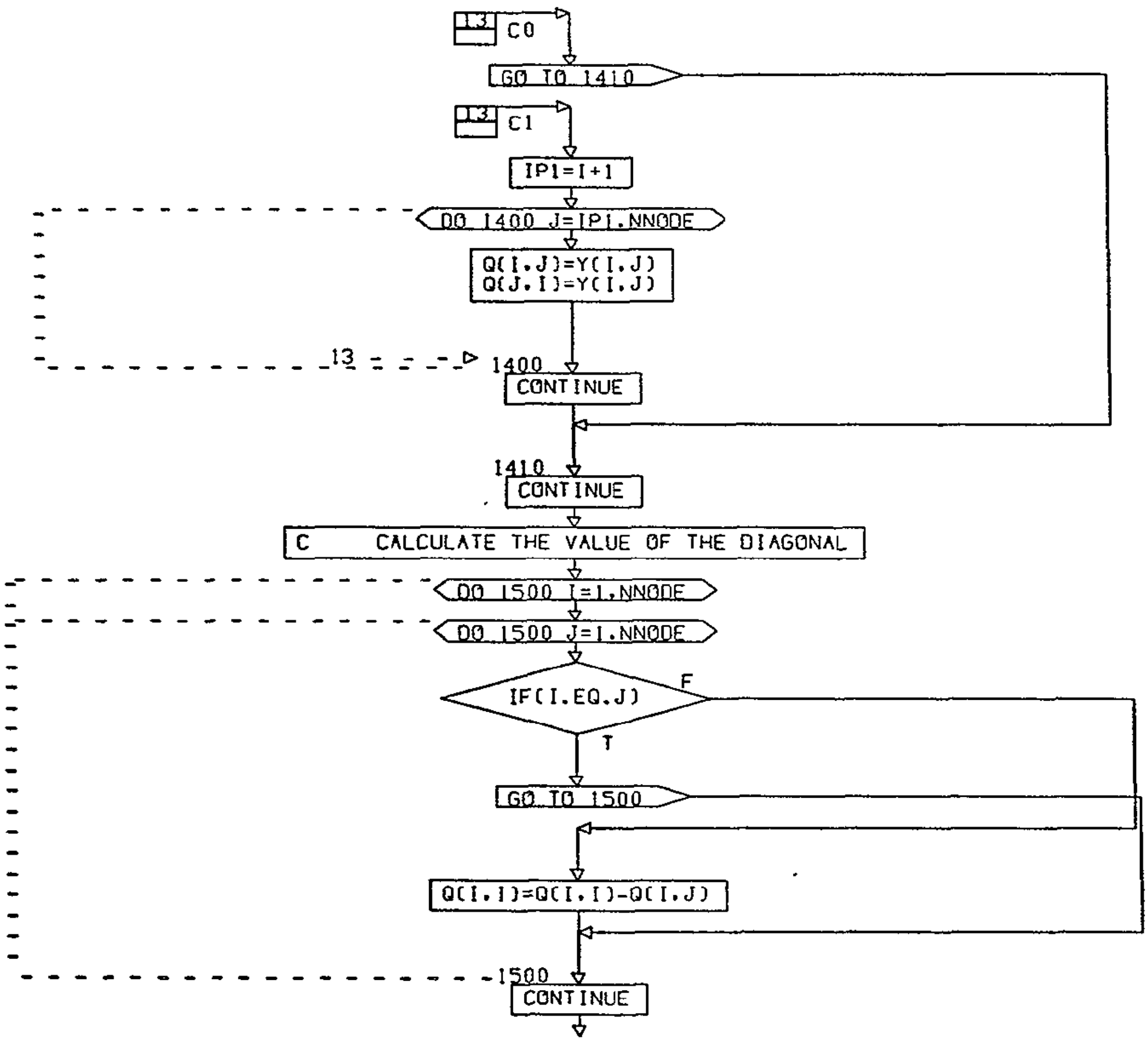
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 14

DCSOLV
PG 13 OF 27

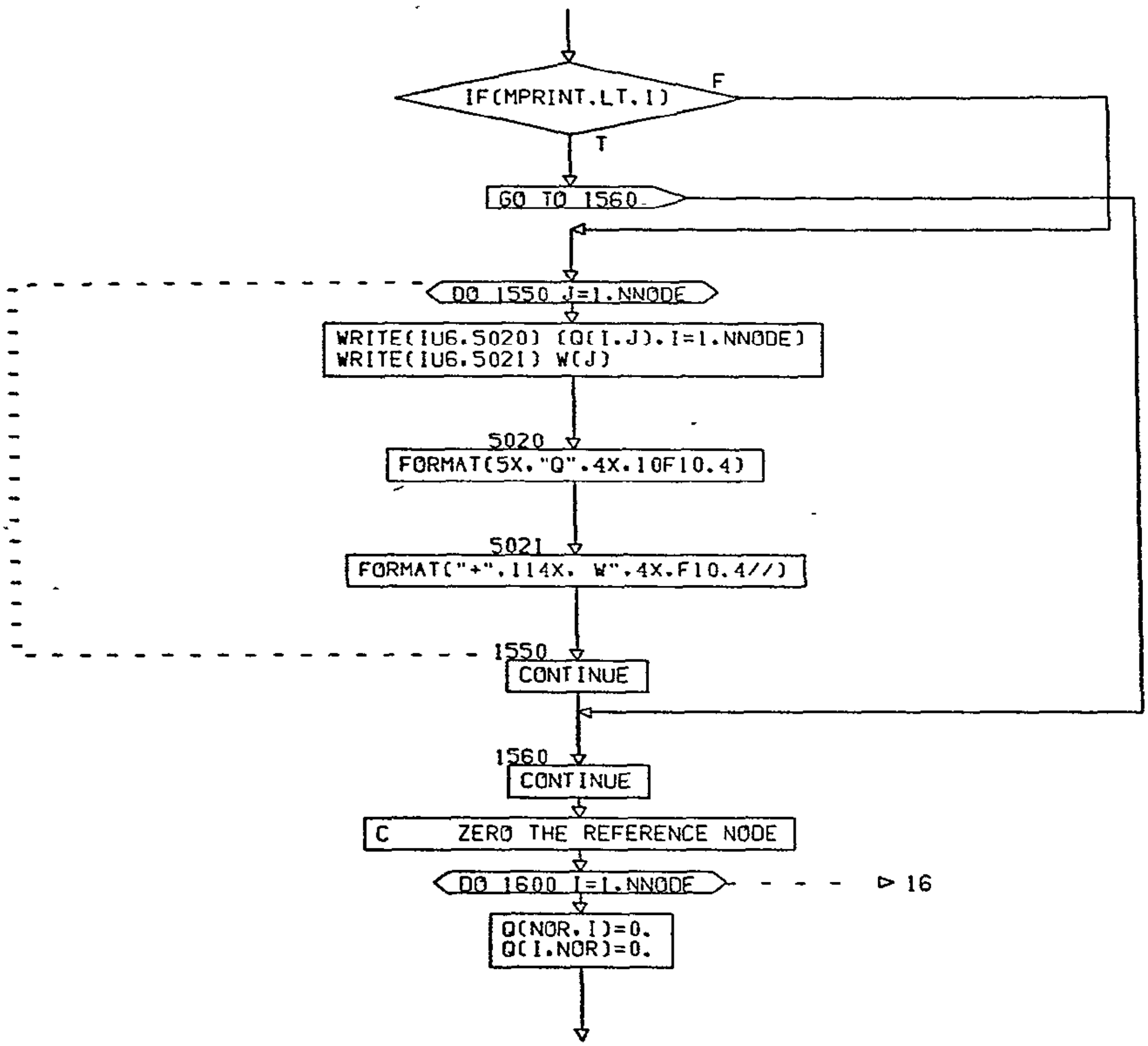
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 15

DCSOLV
PG 14 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 16

DCSOLV
PG 15 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

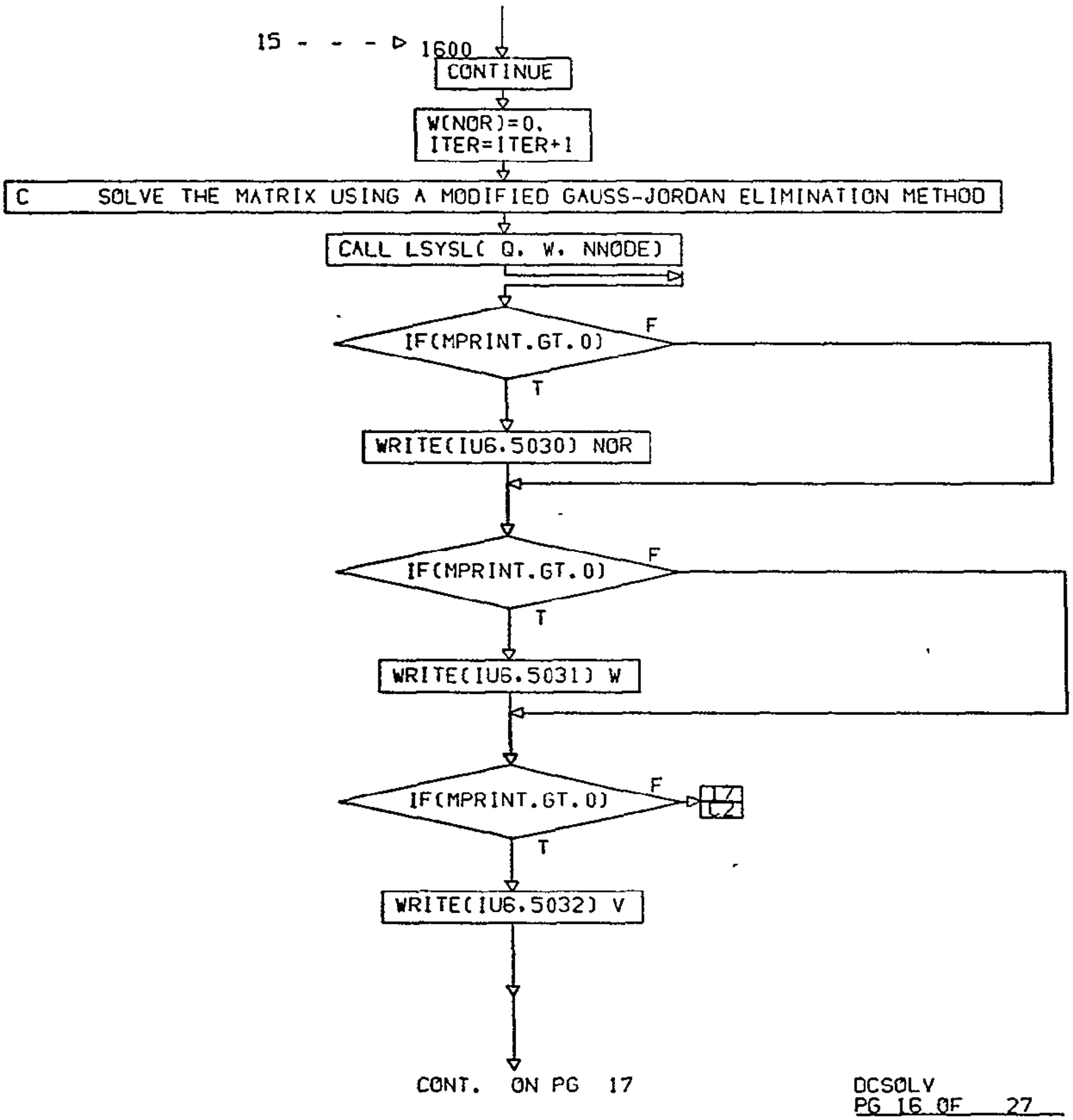
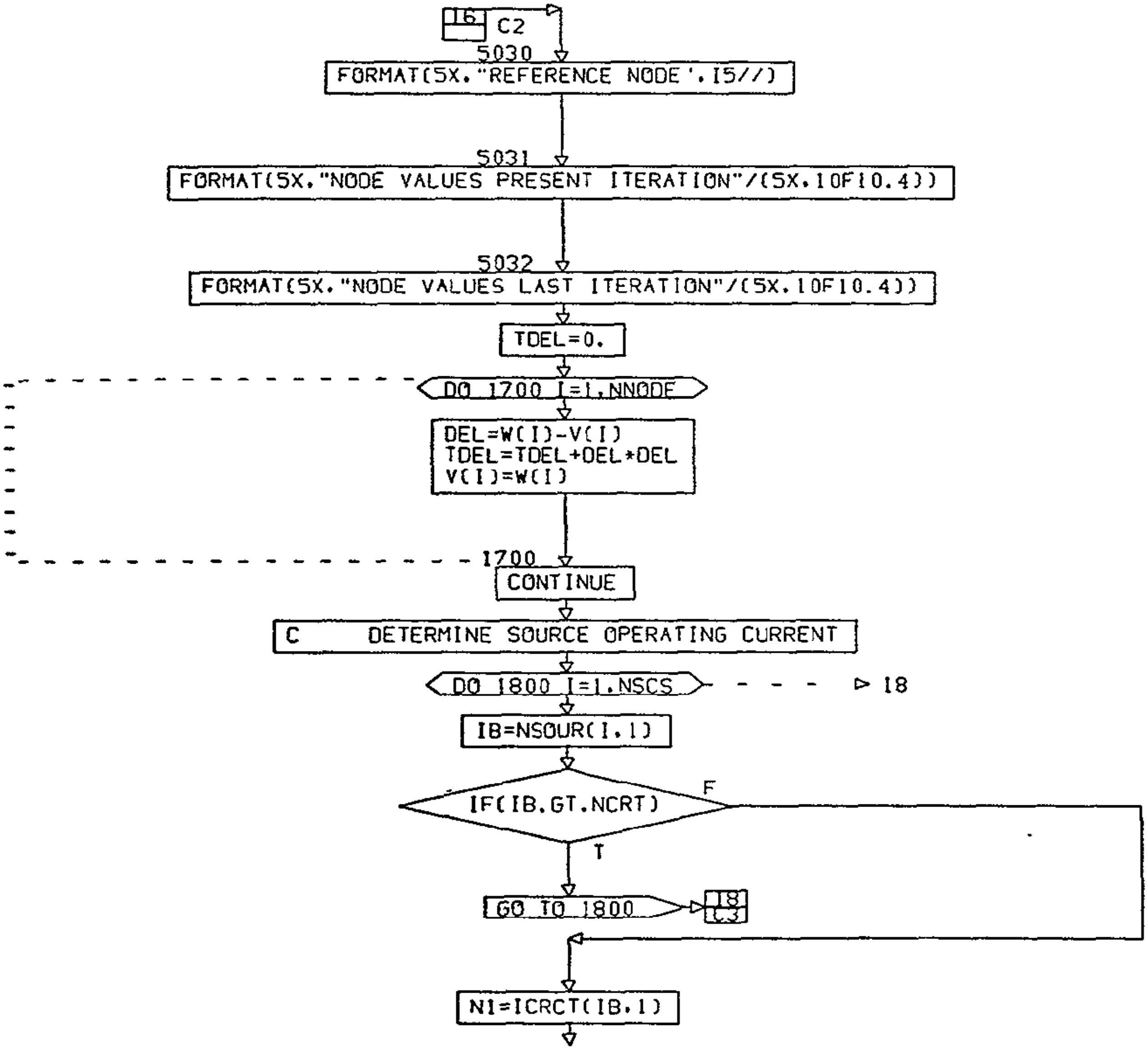


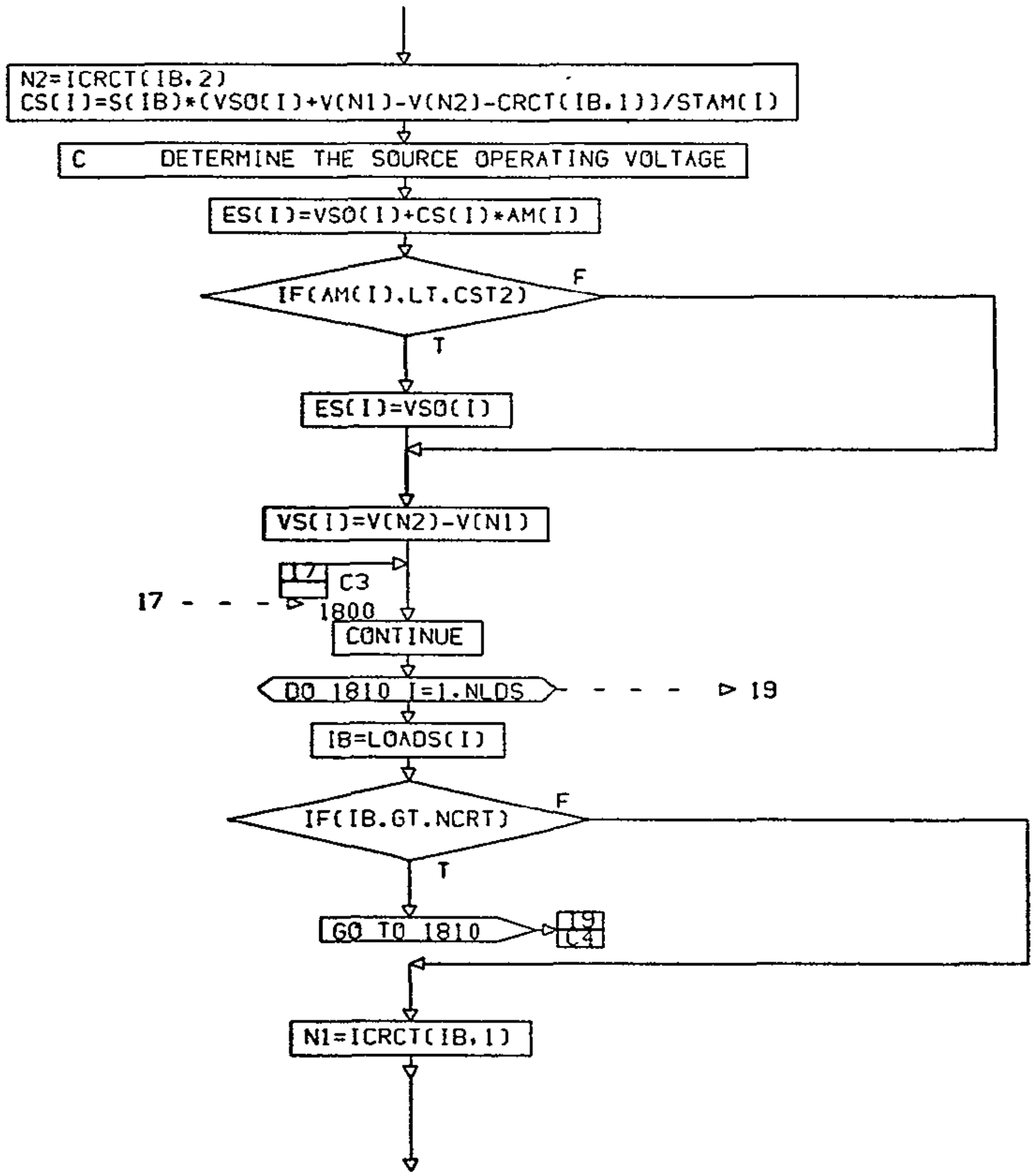
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 18

DCSOLV
PG 17 OF 27

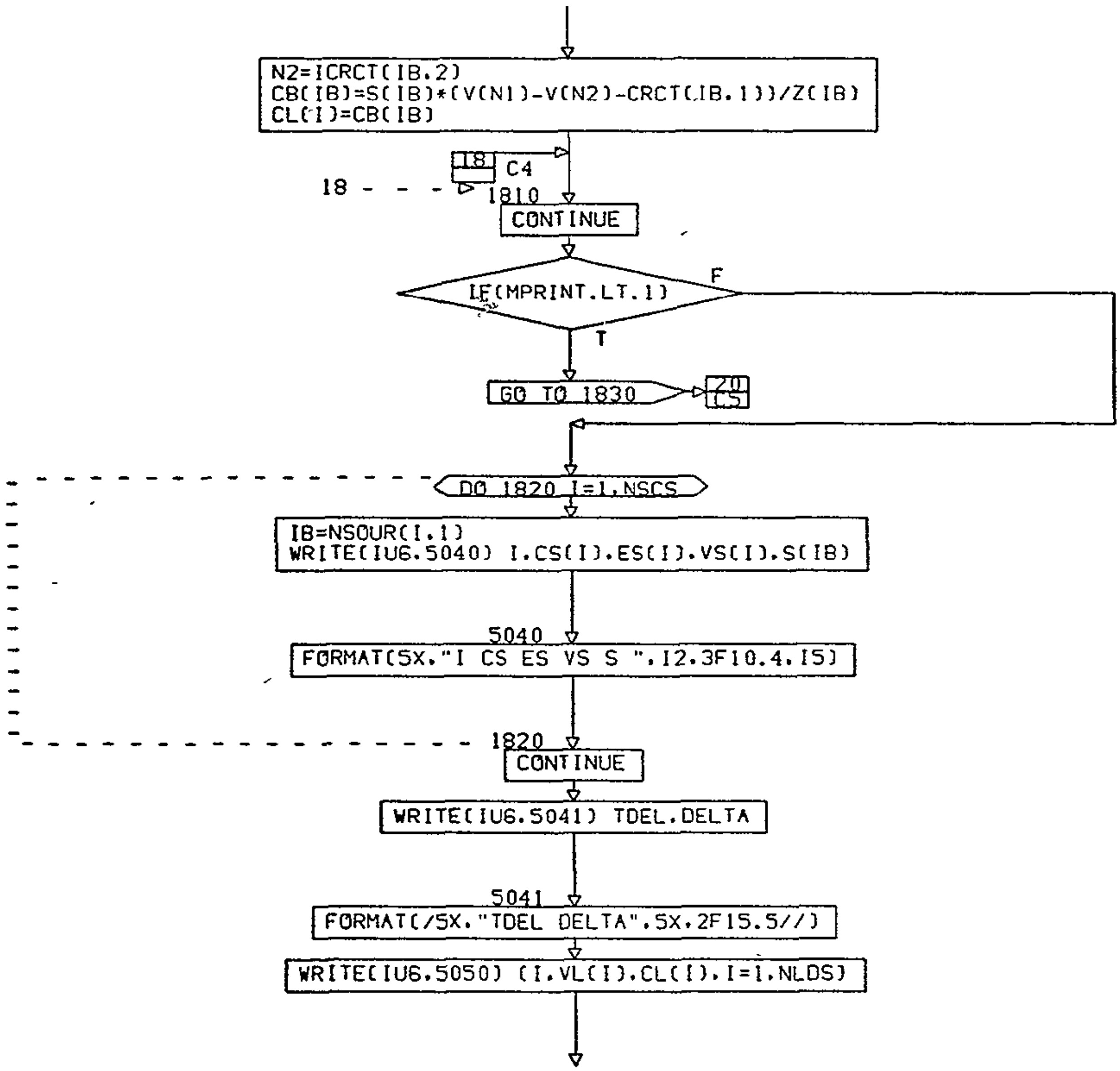
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 19

DCSOLV
PG 18 OF 27

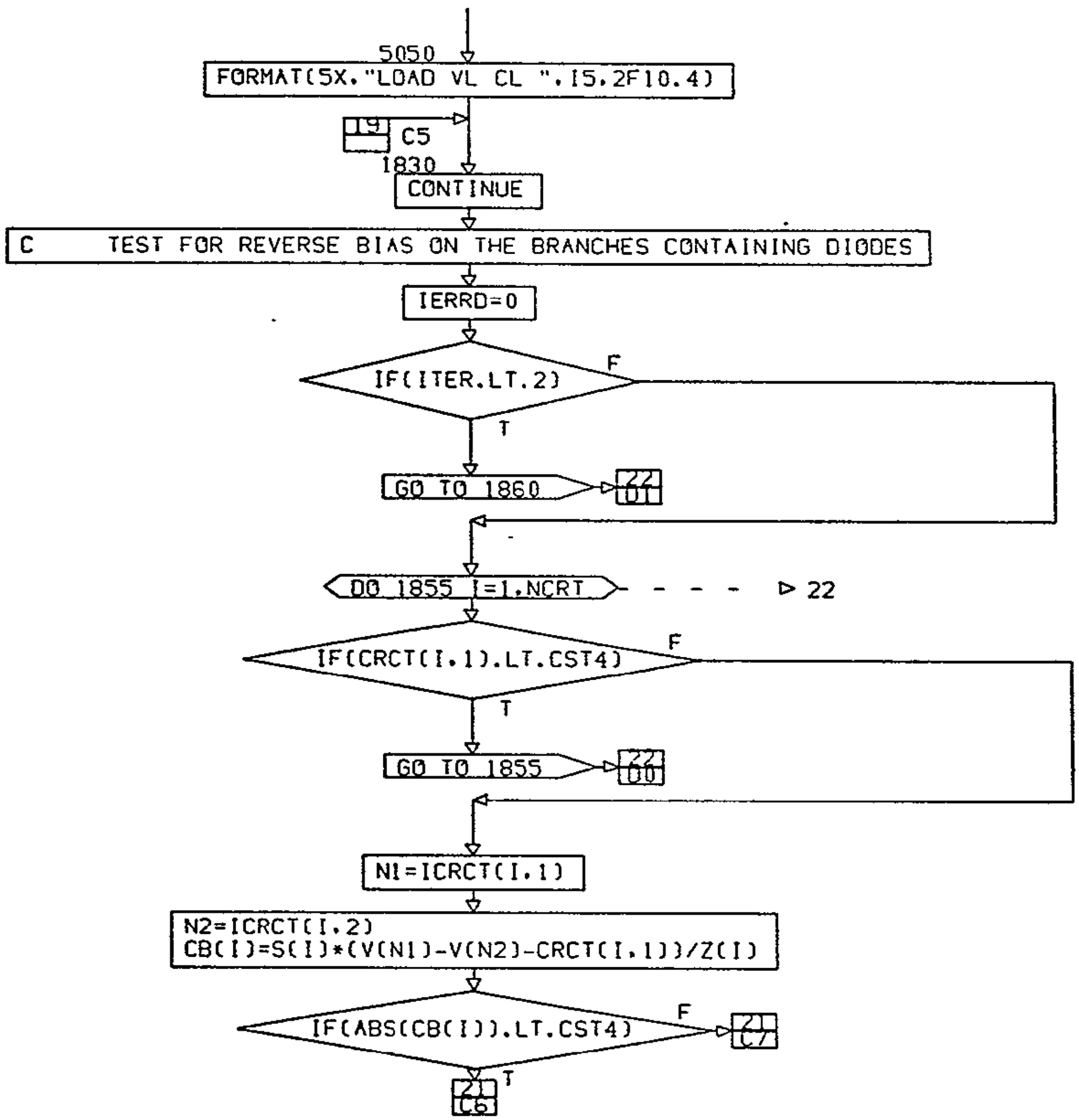
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 20

DCSOLV
PG 19 OF 27

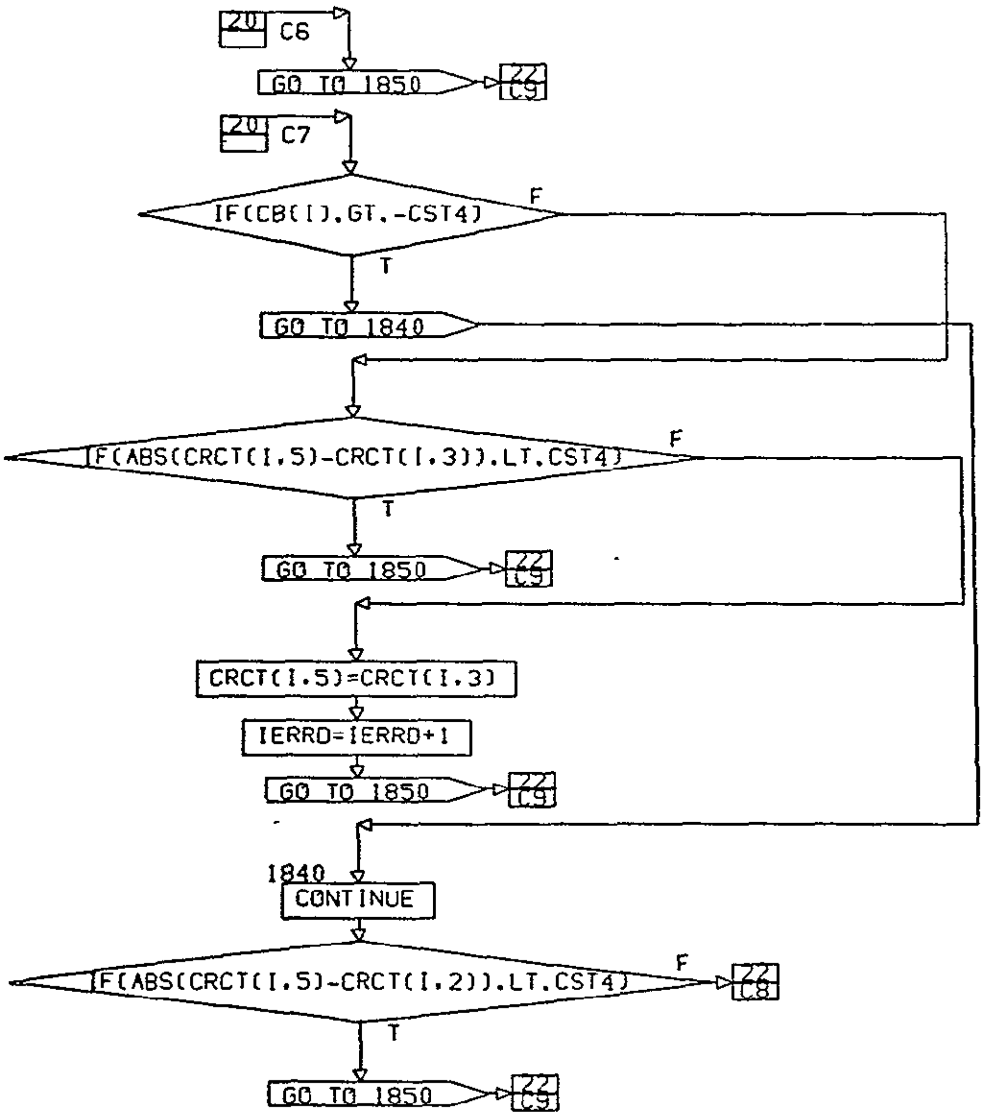
FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 21

DCSOLV
PG 20 OF 27

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 22

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

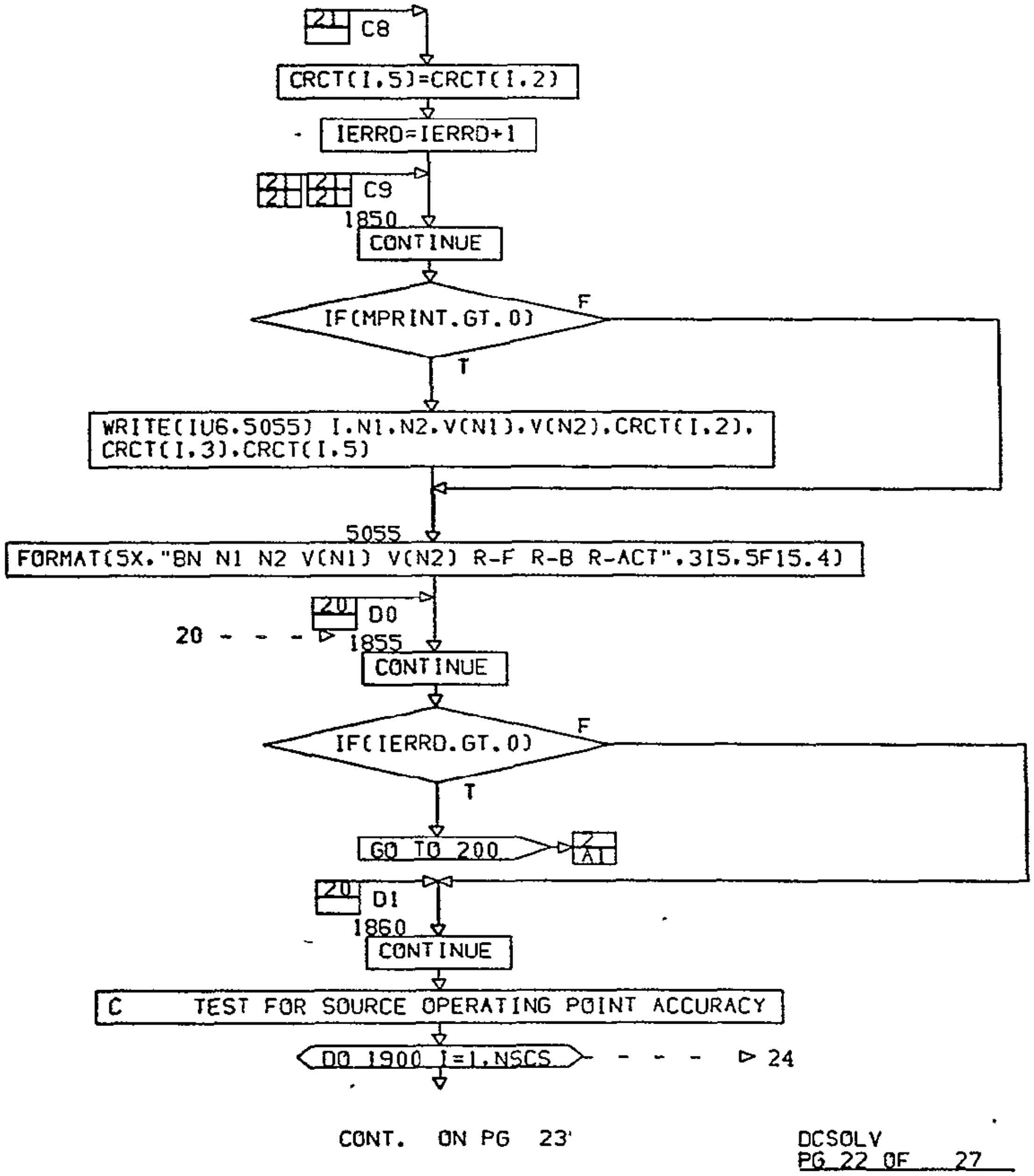
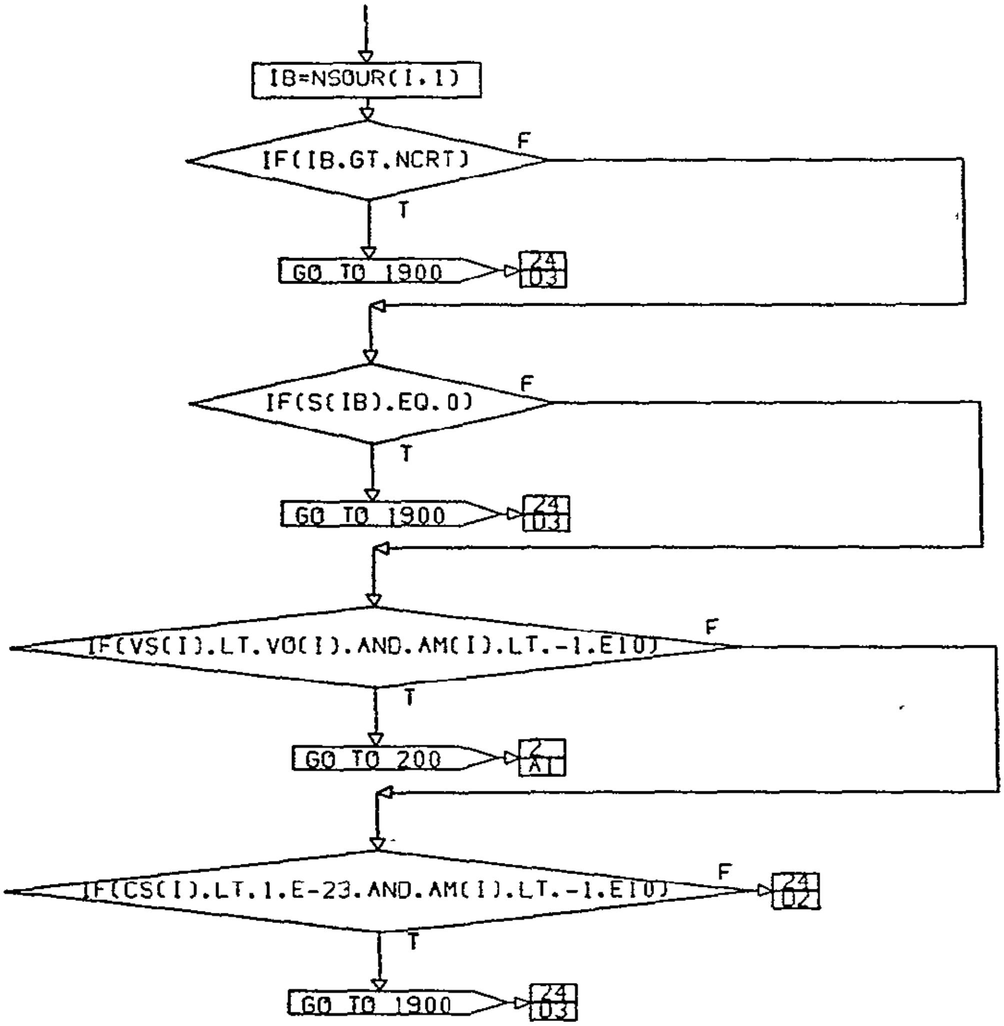


FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



CONT. ON PG 24

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

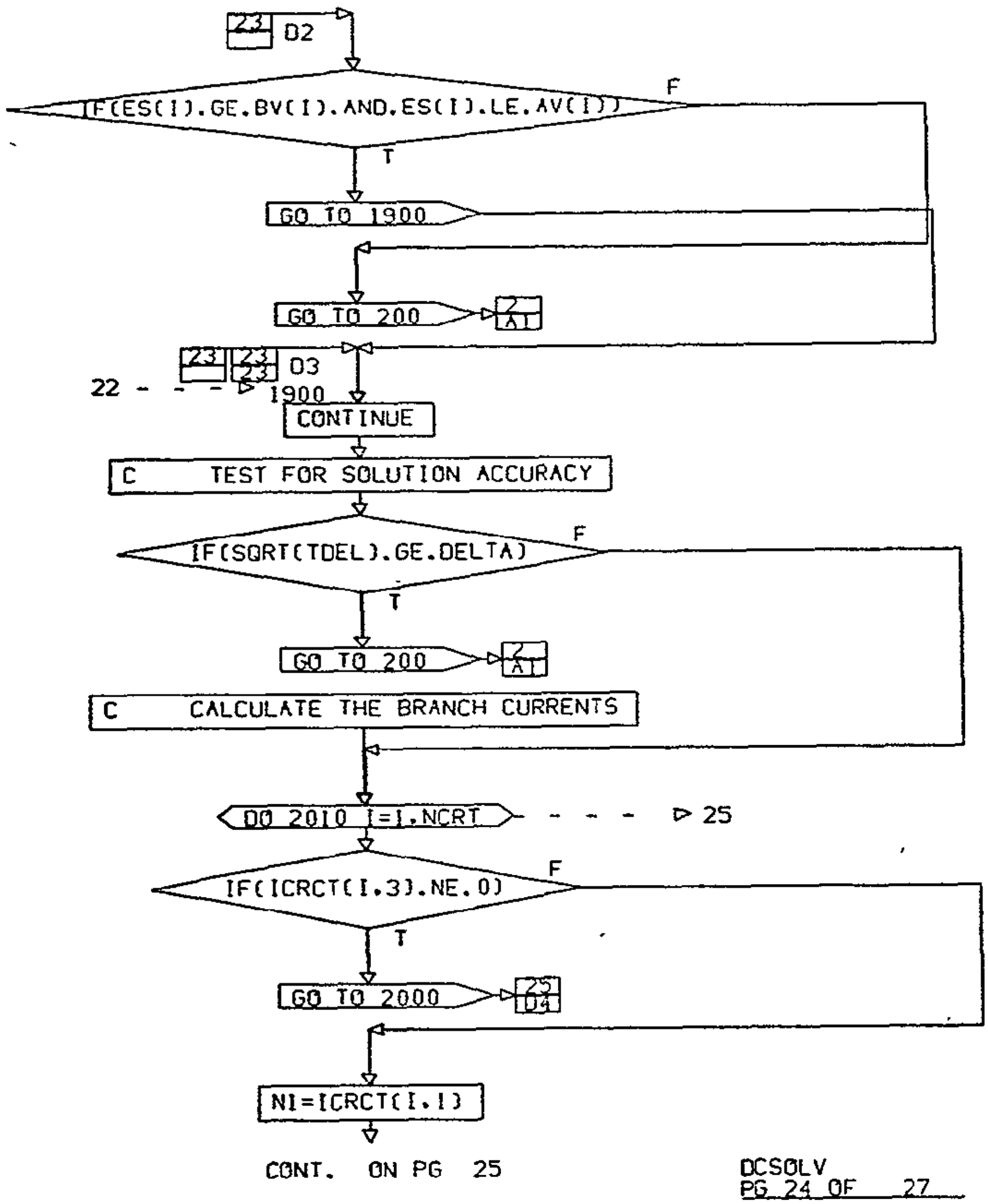


FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

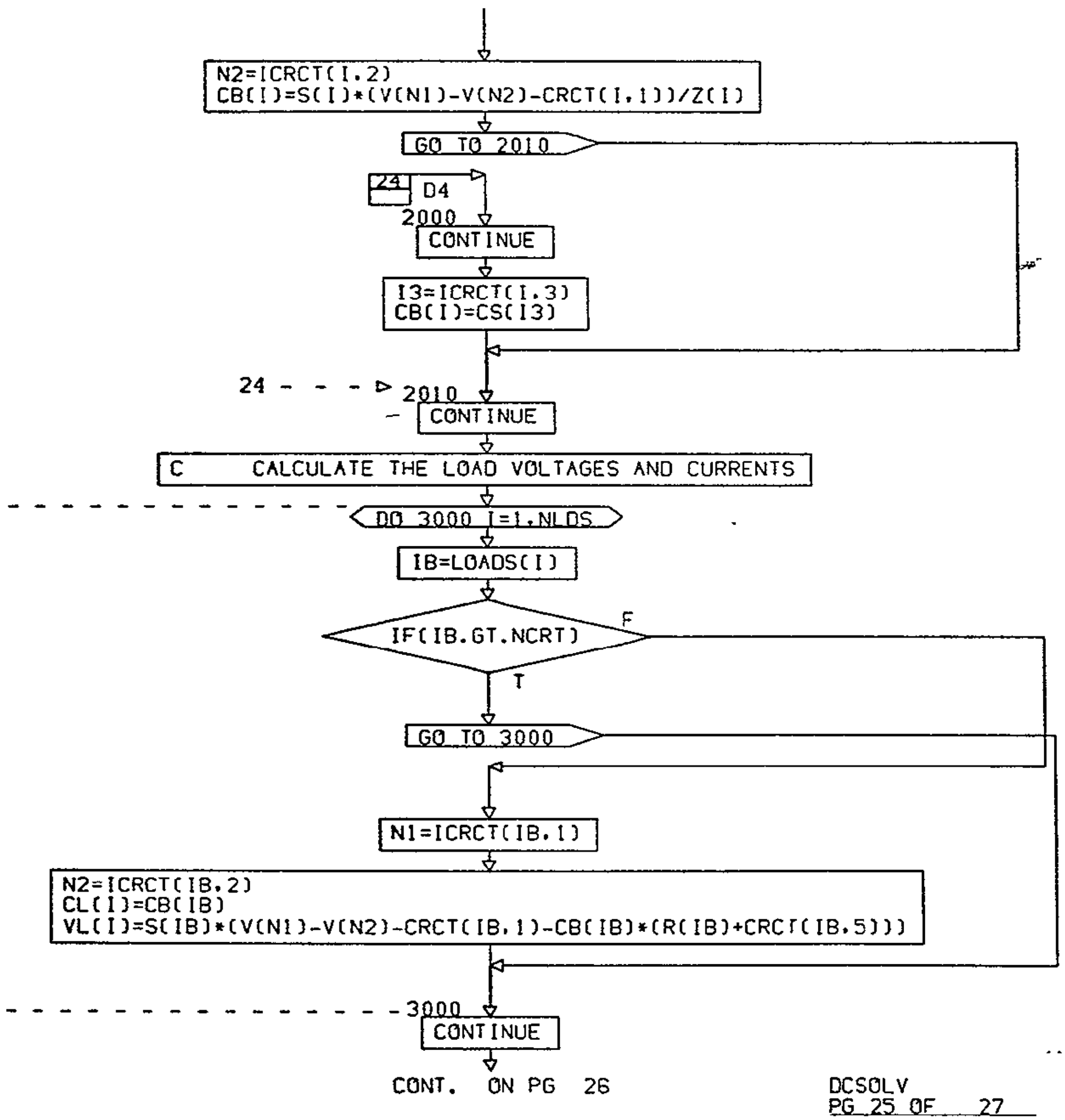


FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

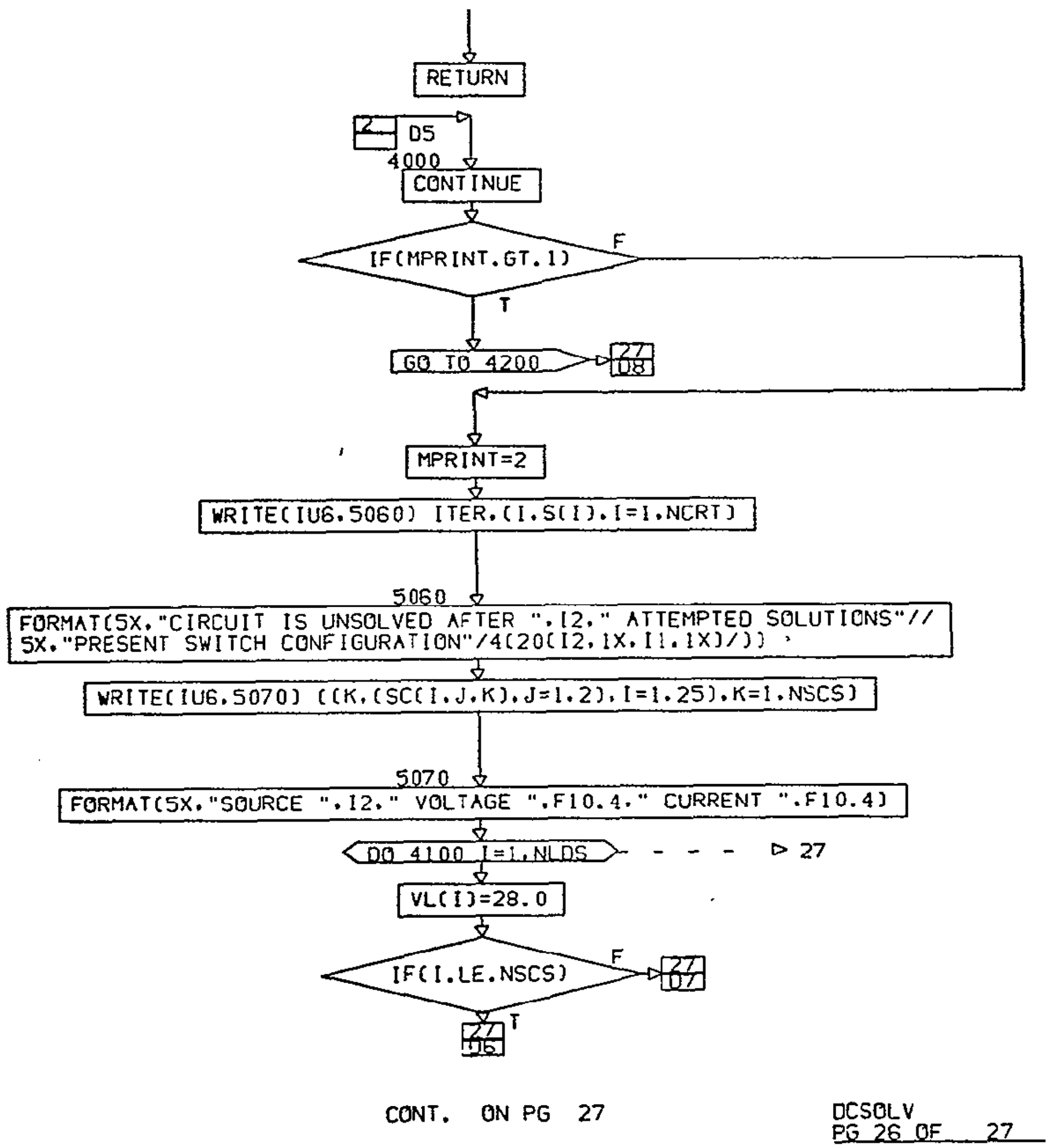
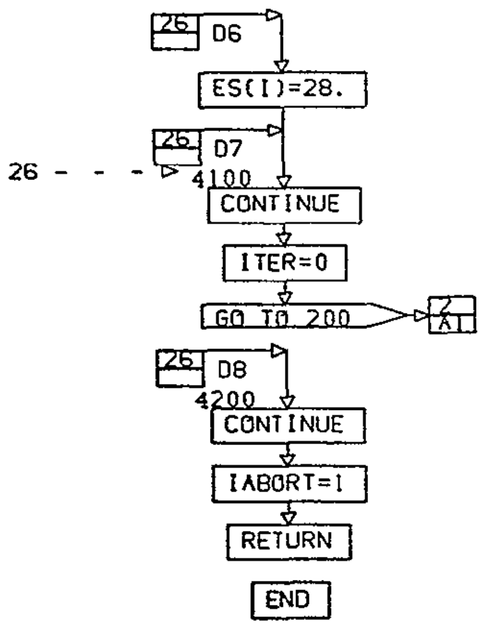


FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)



DCSOLV
PG 27 FINAL

FIGURE 3.3.5. FUNCTIONAL FLOWCHART OF SUBROUTINE DCSOLV (CONTINUED)

3.3.6 Subroutine: FUCLIV

PURPOSE: To create the I-V curves used in simulating the onboard fuel cells

METHOD: An interpolation is made into a set of current-voltage curves as a function of temperature to determine the fuel cell I-V curve at its operating temperature. The curve is further degraded by the parasitic load to be carried by the fuel cell.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.6. See Appendix for definition of all variables.

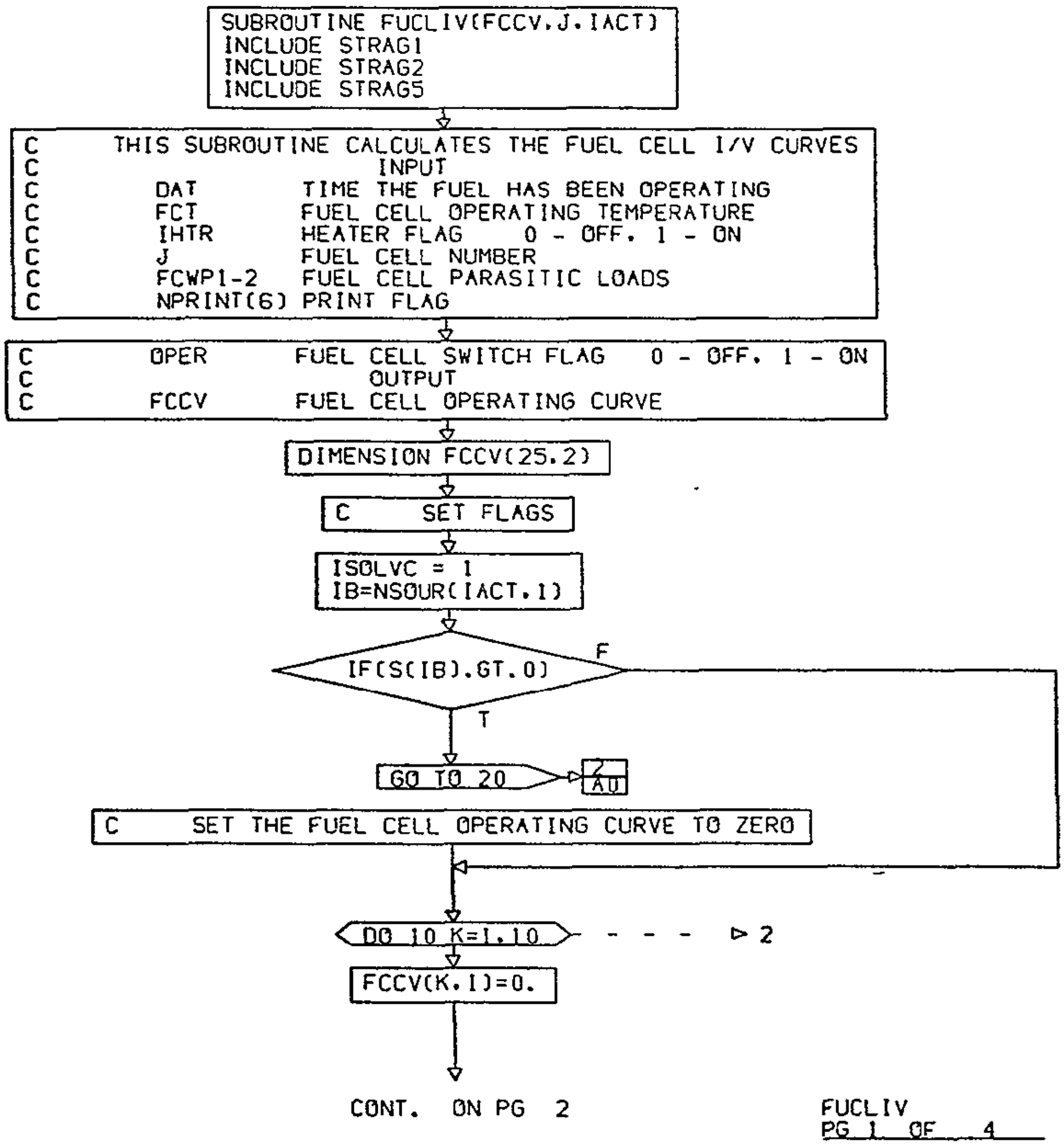


FIGURE 3.3.6. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLIV

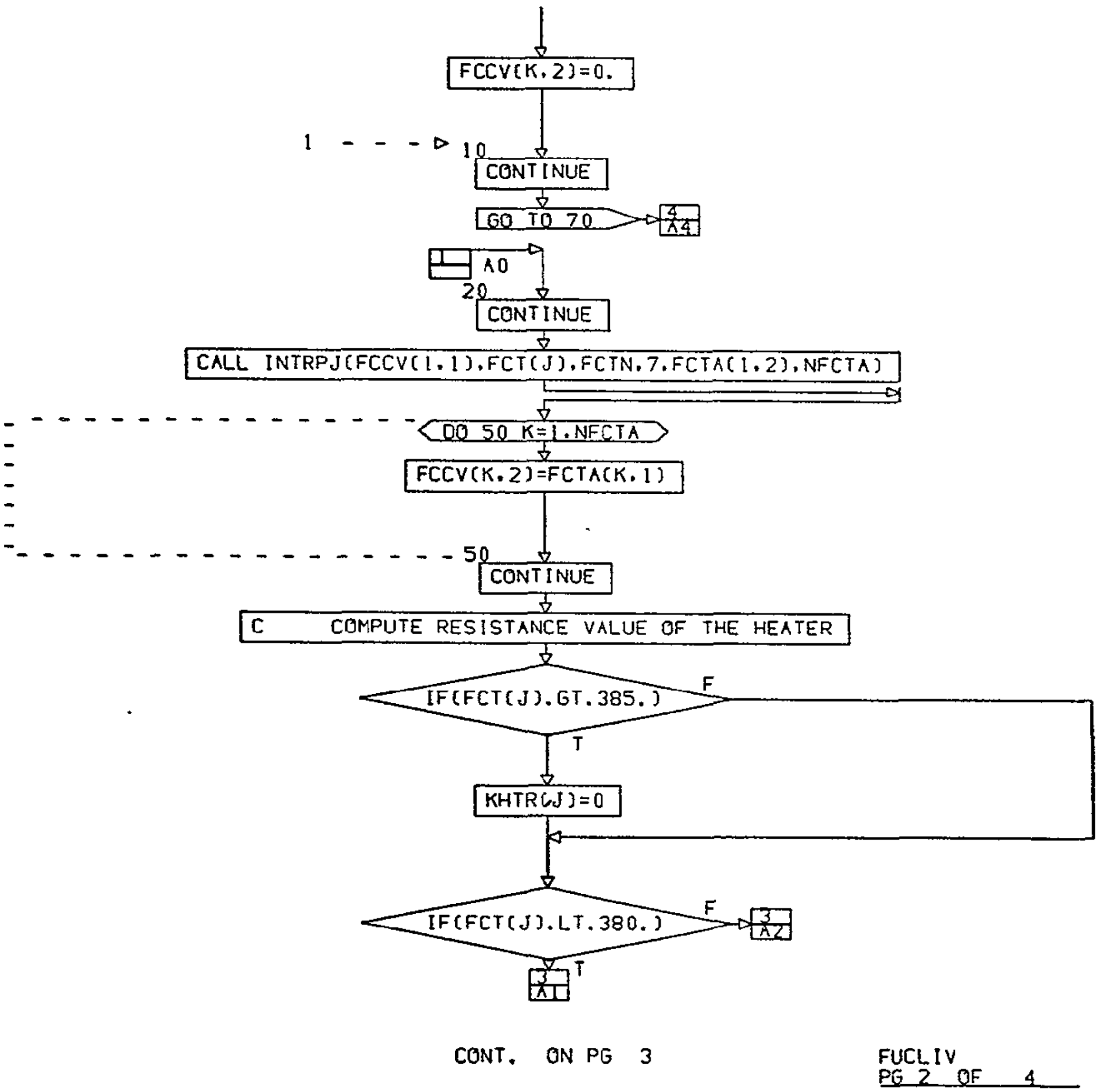
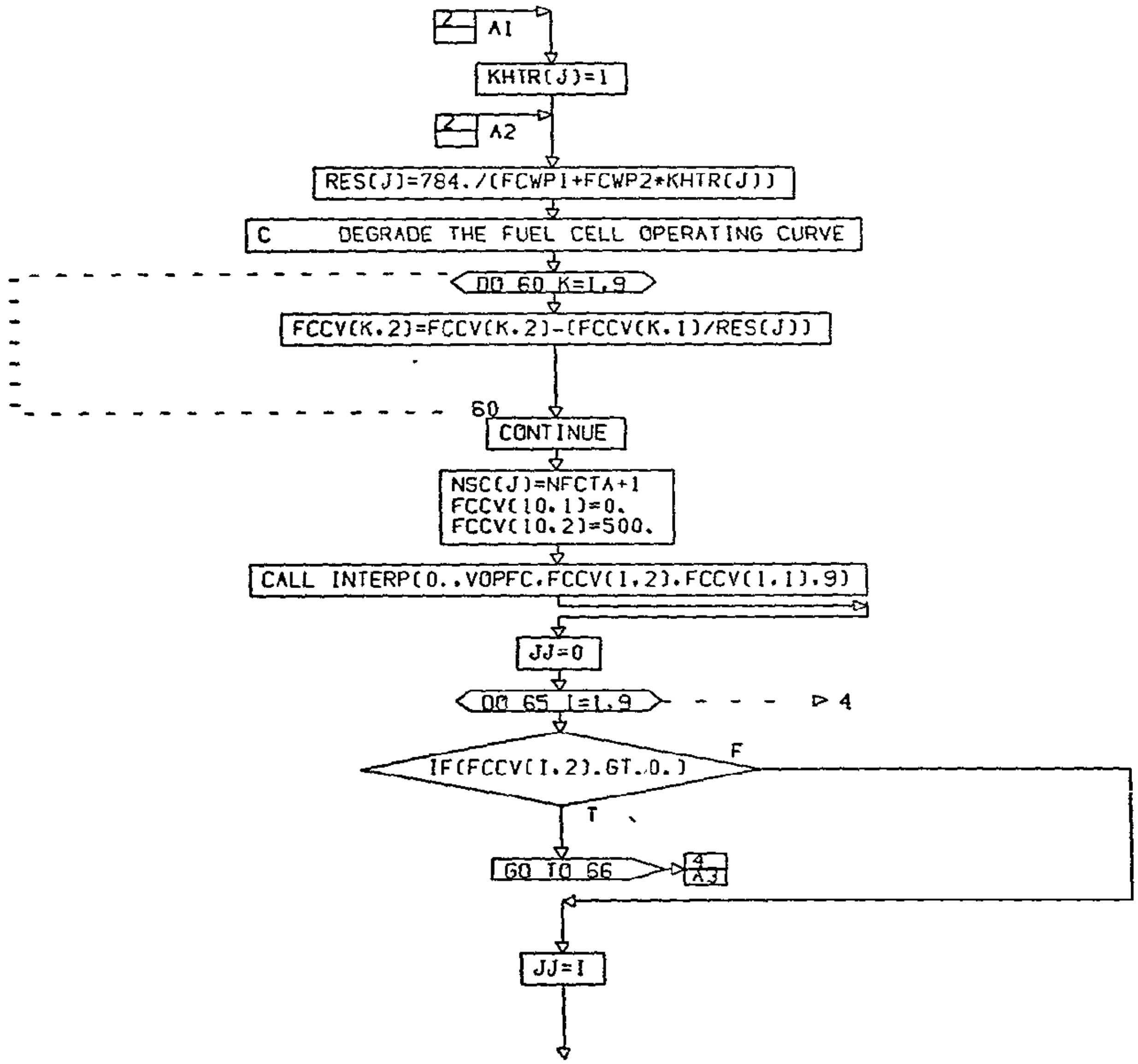


FIGURE 3.3.6. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLIV (CONTINUED)



CONT. ON PG 4

FUCLIV
PG 3 OF 4

FIGURE 3.3.6. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLIV (CONTINUED)

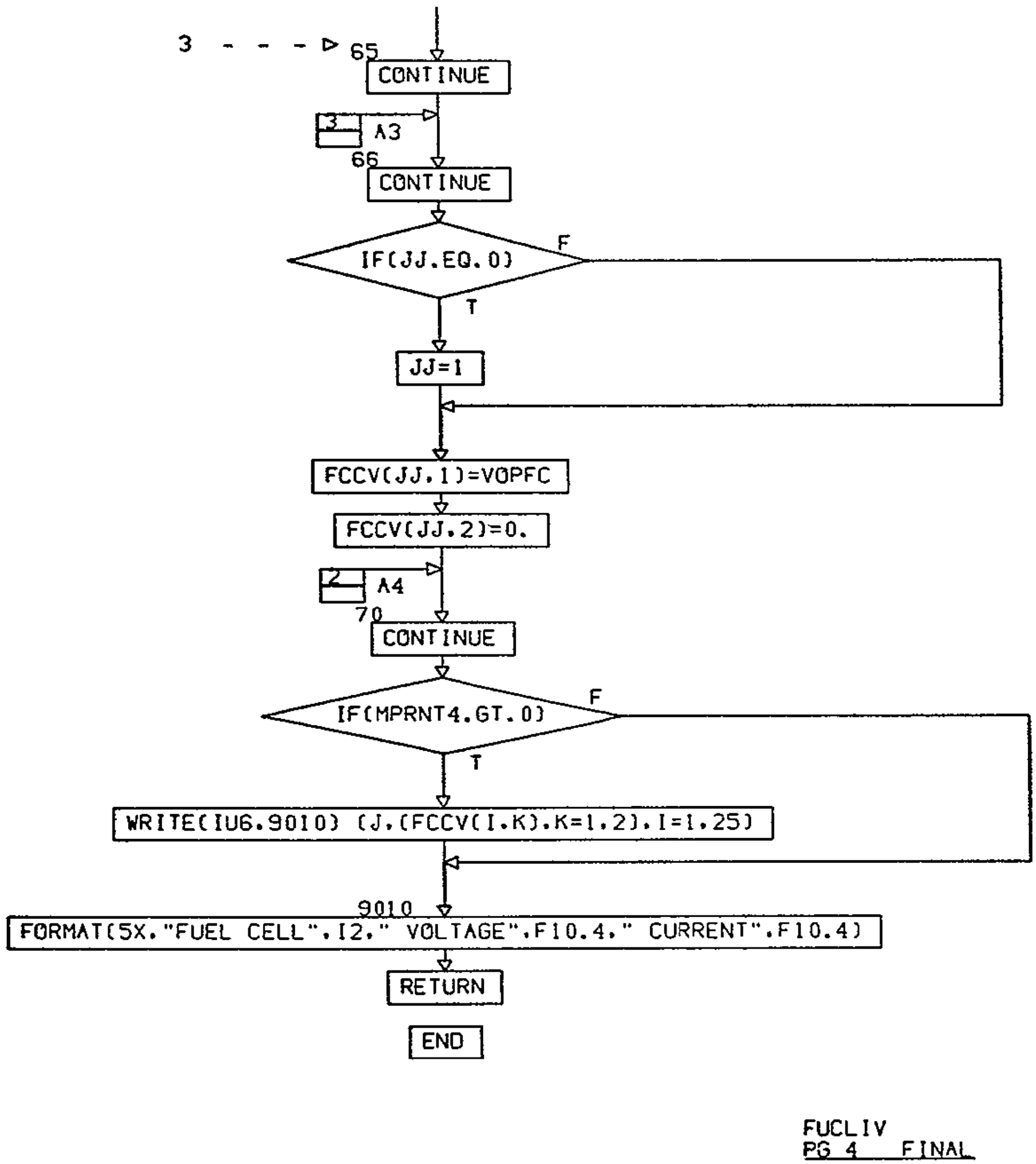


FIGURE 3.3.6. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLIV (CONTINUED)

3.3.7 Subroutine: FUCLTM

PURPOSE: To determine the change in fuel cell temperature and certain gross cyrogenic quantities.

METHOD: Based on the current fuel cell operating temperature and steady state temperatures versus current curves an ideal operating current is determined. The difference between the ideal current and the operating current is used to determine the change in fuel cell temperature.

Based on input purge times and rates, and input usage rates, the amount of oxygen and hydrogen used and the quantity of water produced is calculated.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.7. See Appendix for definition of all variables.

```
SUBROUTINE FUCLTM(J)
INCLUDE STRAG1
INCLUDE STRAG5
```

```
C     THIS SUBROUTINE PERFORMS THE FOLLOWING FUEL CELL CALCULATIONS
C        HYDROGEN AND OXYGEN DEPLETION
C        WATER PRODUCTION
C        HYDROGEN AND OXYGEN PURGES
C        CRYOGENIC DEPLETION
C                       INPUT
C        DTM        TIME STEP
C        FCCP       OPERATING CURRENT OF FUEL CELL
```

```
C        HR         HYDROGEN PURGE RATE
C        HPT        TIME REQUIRED FOR HYDROGEN PURGE
C        HUR        HYDROGEN USAGE RATE
C        J          FUEL CELL NUMBER
C        OR         OXYGEN PURGE RATE
C        OPT        TIME REQUIRED FOR OXYGEN PURGE
C        OUR        OXYGEN USAGE RATE
C        PIH        INTERVAL BETWEEN HYDROGEN PURGES
```

```
C        PIO        INTERVAL BETWEEN OXYGEN PURGES
C        TMAXFC     MAX ALLOWABLE TIME FOR ONE PASS THRU  FUCLTM
C        TPH        TIME OF LAST HYDROGEN PURGE
C        TPO        TIME OF LAST OXYGEN PURGE
C        WPR        WATER PRODUCTION RATE
C                       INPUT/OUTPUT
C        H2OT       TOTAL AVAILABLE WATER
C        H2T        TOTAL AVAILABLE HYDROGEN
```

```
C        O2T        TOTAL AVAILABLE OXYGEN
C                       OUTPUT
C        DAT        TIME FUEL CELL HAS BEEN OPERATING
C        H2         HYDROGEN DEPLETED THIS TIMESTEP
C        O2         OXYGEN DEPLETED THIS TIMESTEP
C        H2O        WATER PRODUCED THIS TIMESTEP
C     COMPUTE INTERNAL TIME VARIABLES
```

```
KMAX=(DTM/TMAXFC)+1.
DTFC=DTM/KMAX
```

CONT. ON PG 2

FUCLTM
PG 1 OF 6

IGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM

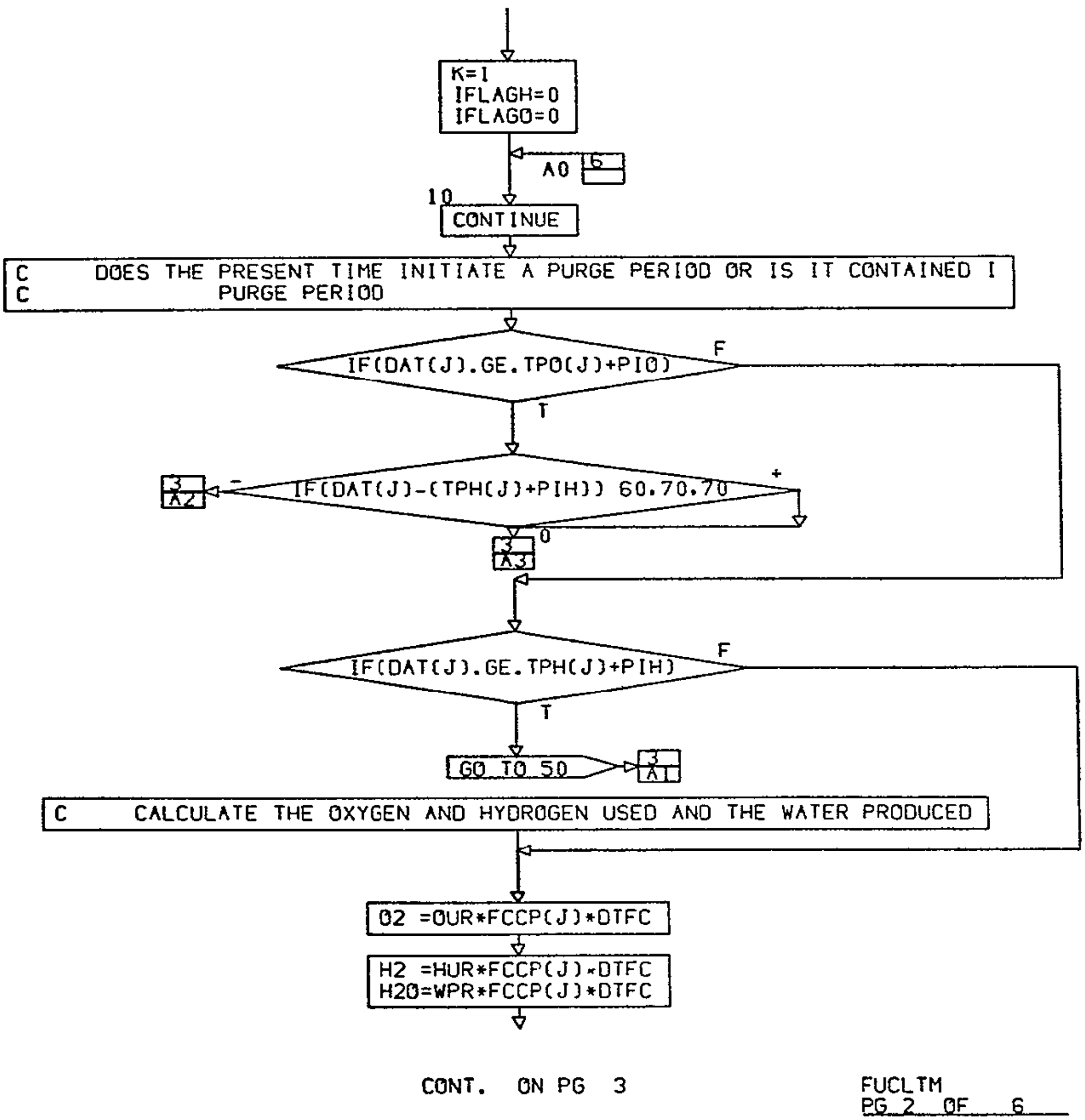


FIGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM (CONTINUED)

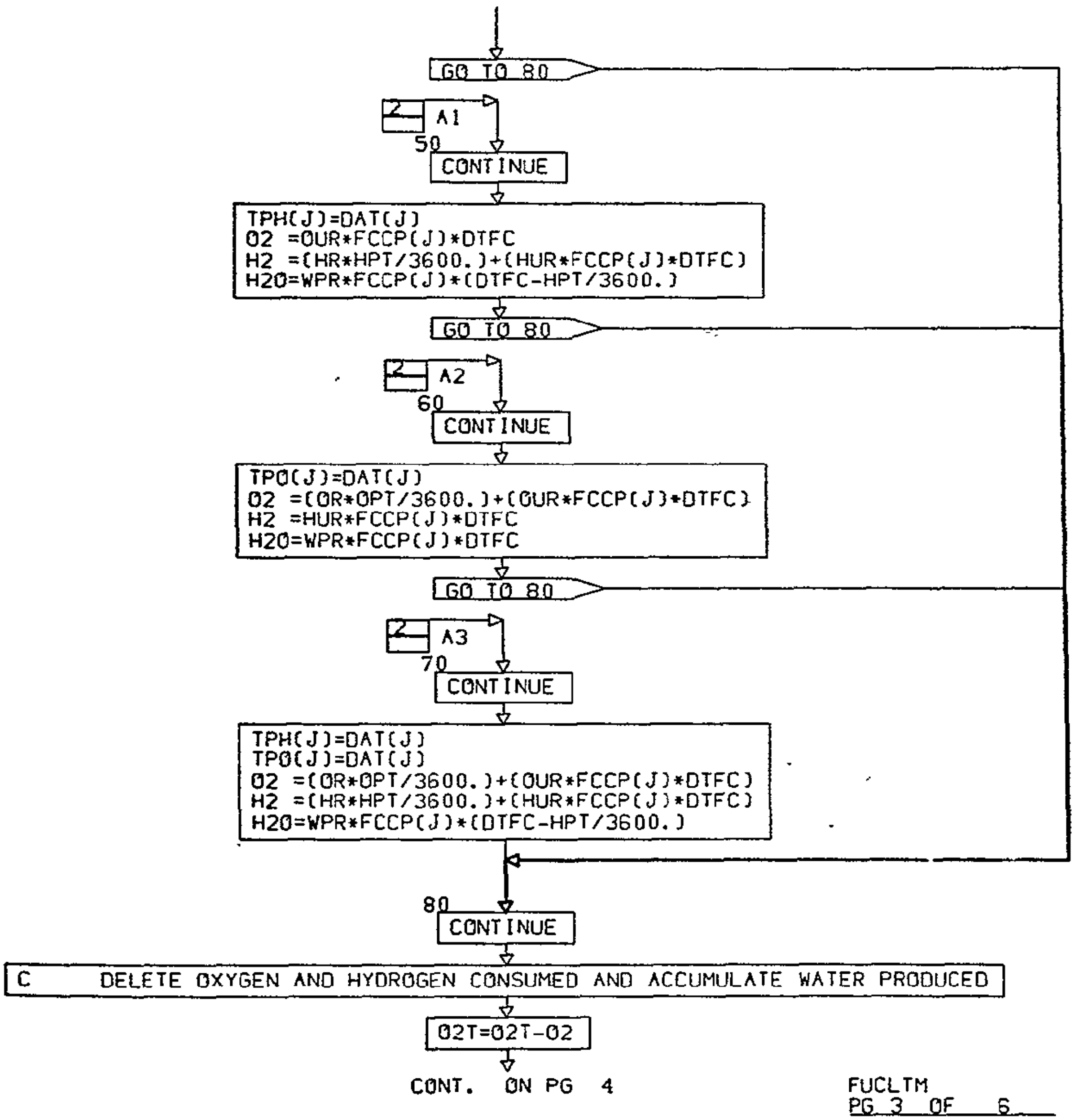


FIGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

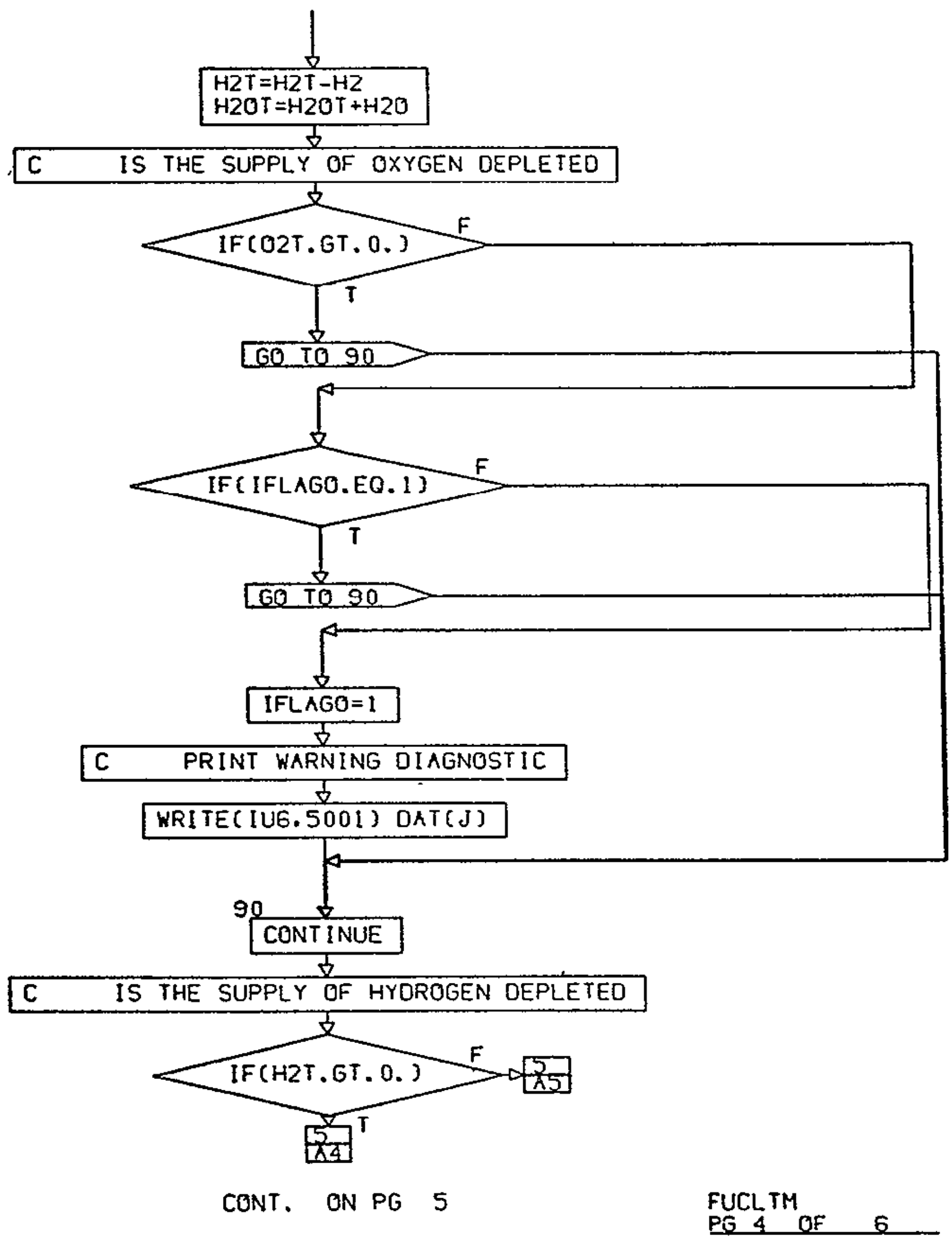


FIGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM (CONTINUED)

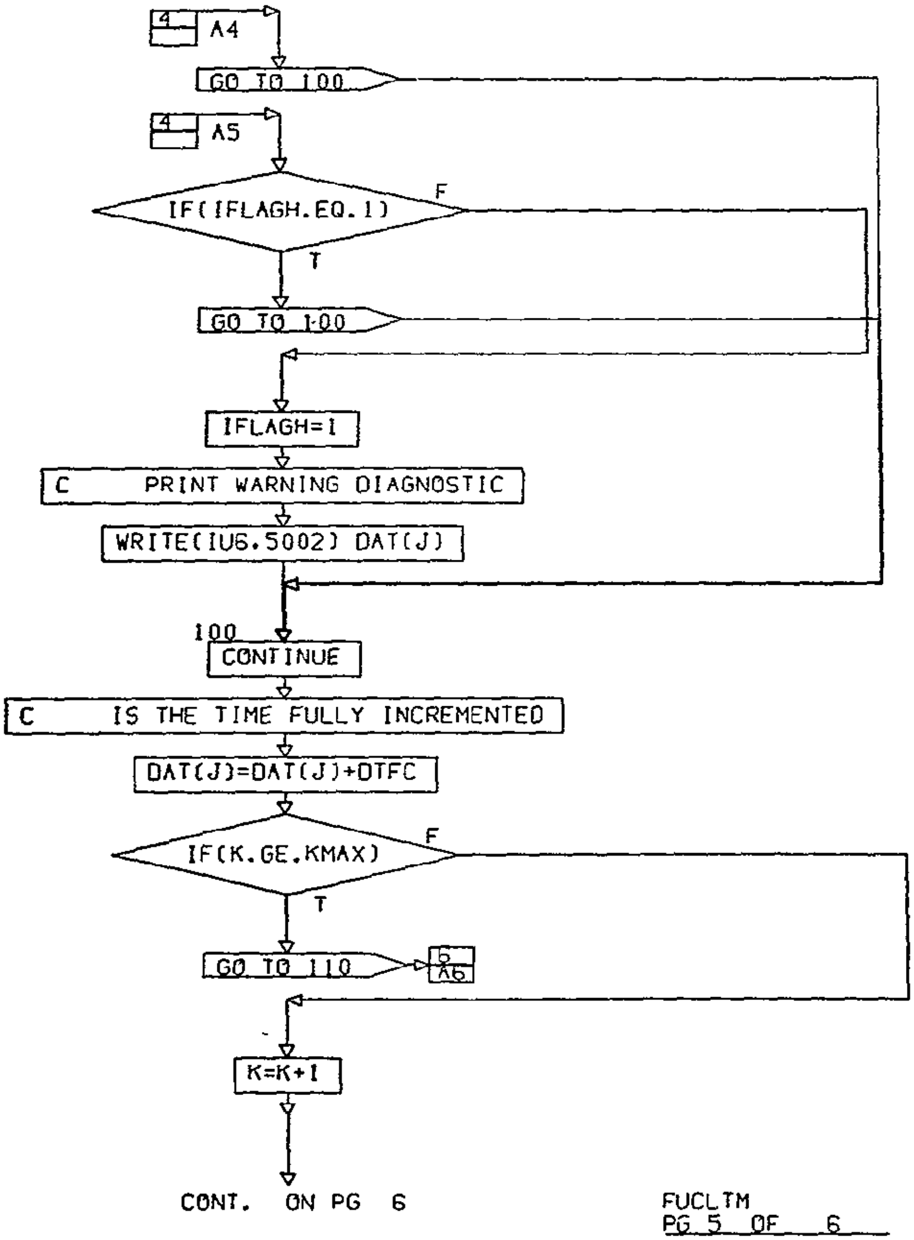
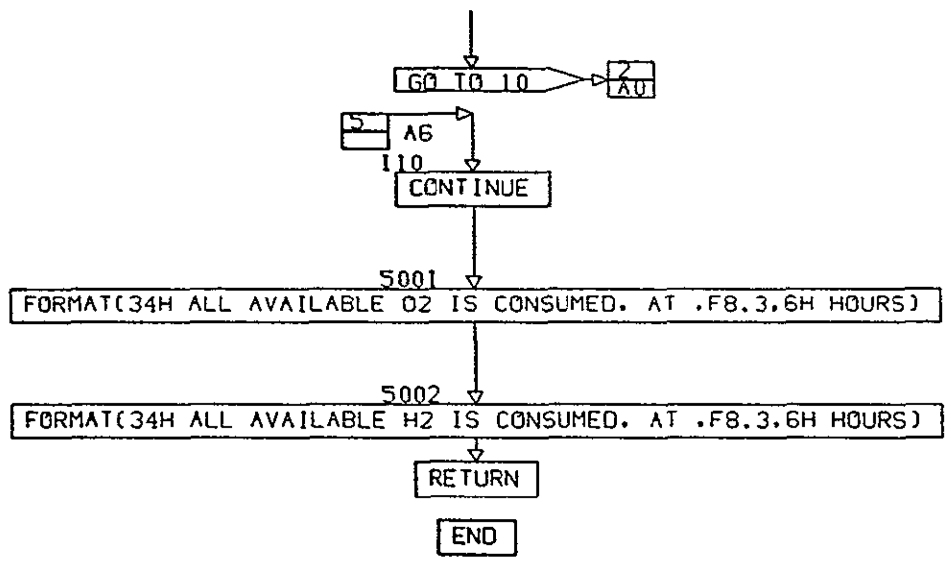


FIGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM (CONTINUED)



FUCLTM
PG 6 FINAL

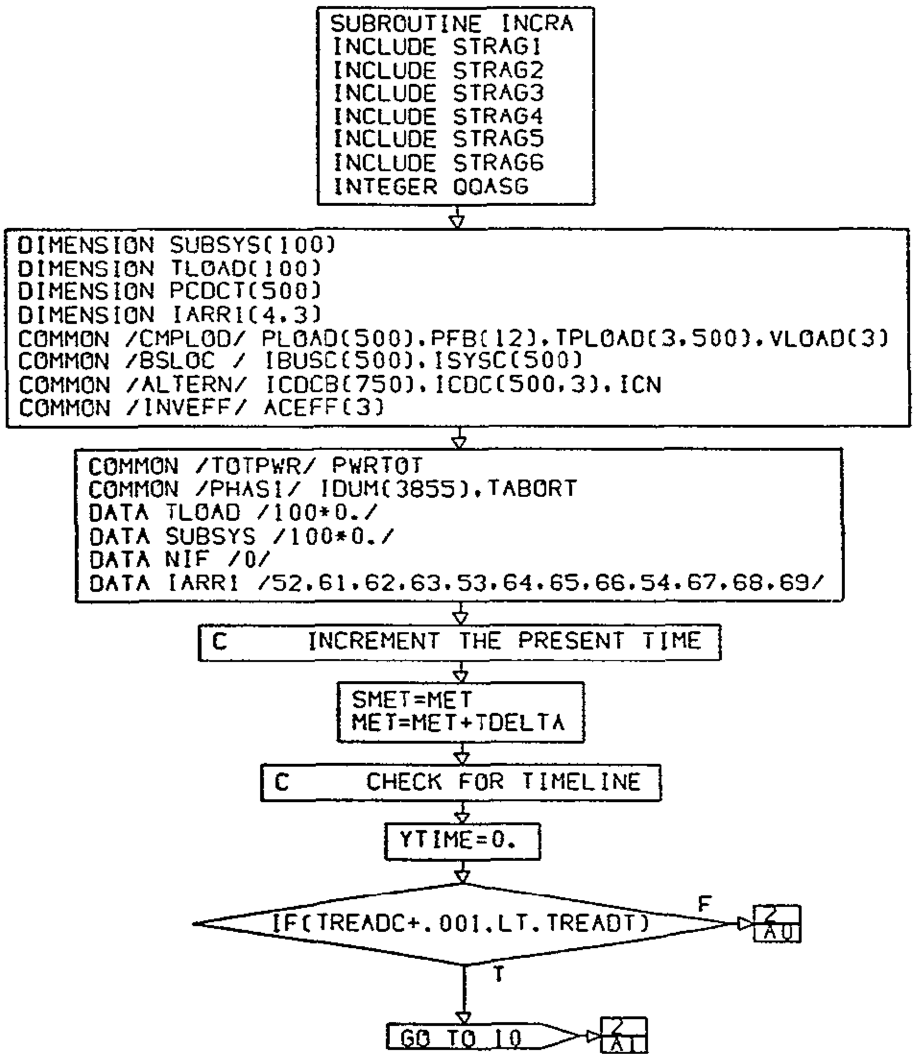
FIGURE 3.3.7. FUNCTIONAL FLOWCHART OF SUBROUTINE FUCLTM (CONTINUED)

3.3.8 Subroutine: INCRA

PURPOSE: To control the sequential time dependent operation of Phase II.

METHOD: The present time (MET or GET) is incremented by the input time step. Both the input card timeline and the interface tape are checked to see if either or both should be read. If both are to be read, the interface tape is read first, then the card timeline.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.8. See Appendix for definition of all variables.



CONT. ON PG 2

INCRA
PG 1 OF 8

FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA

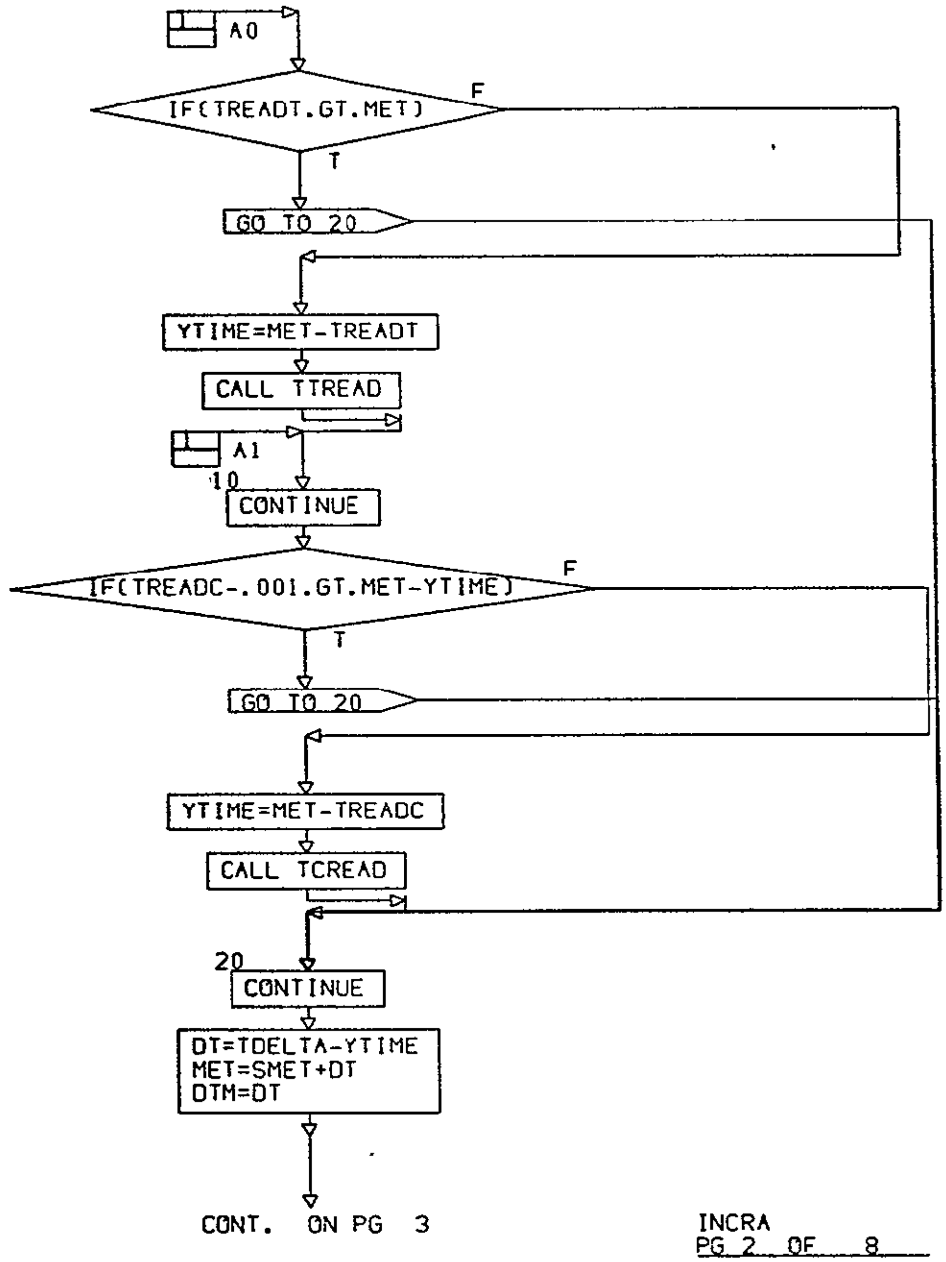


FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)

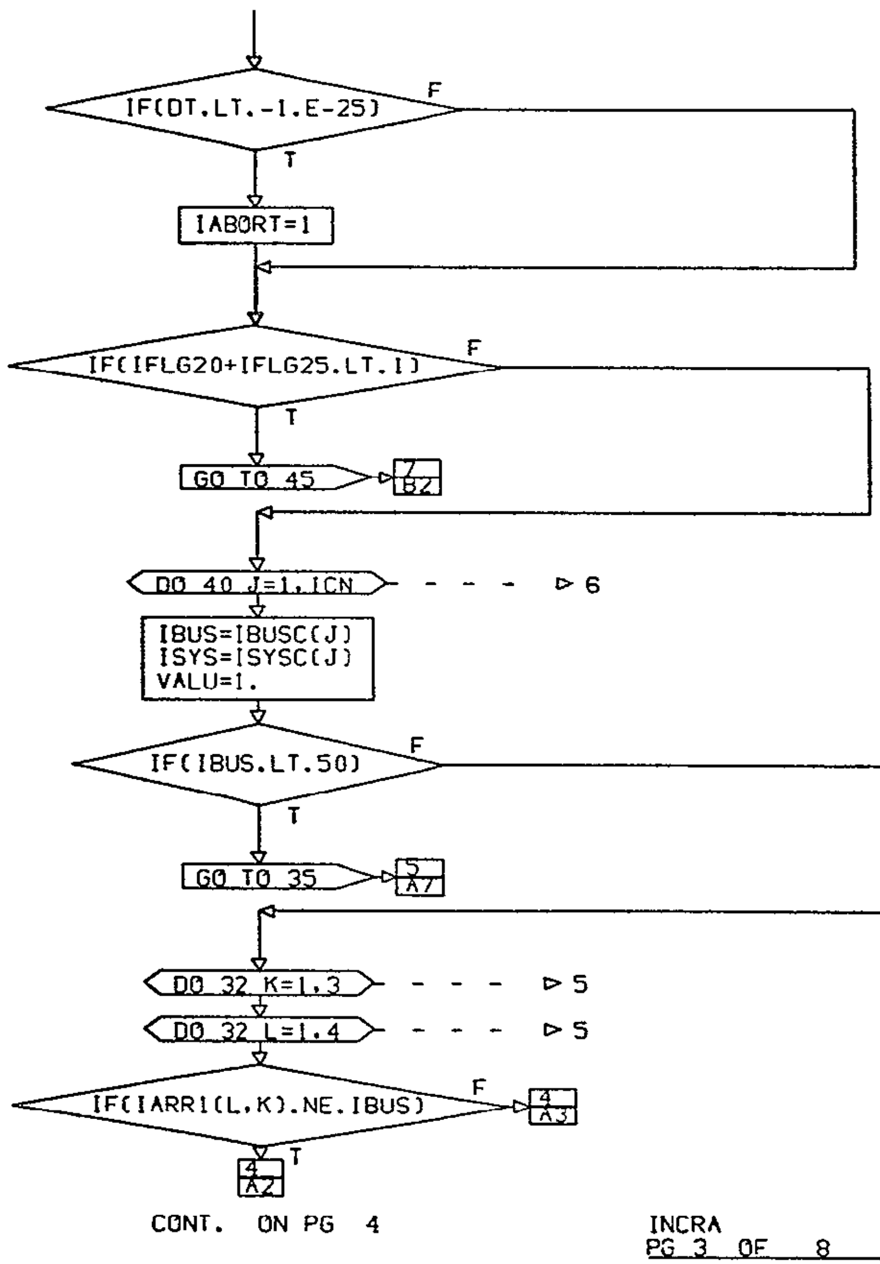


FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)

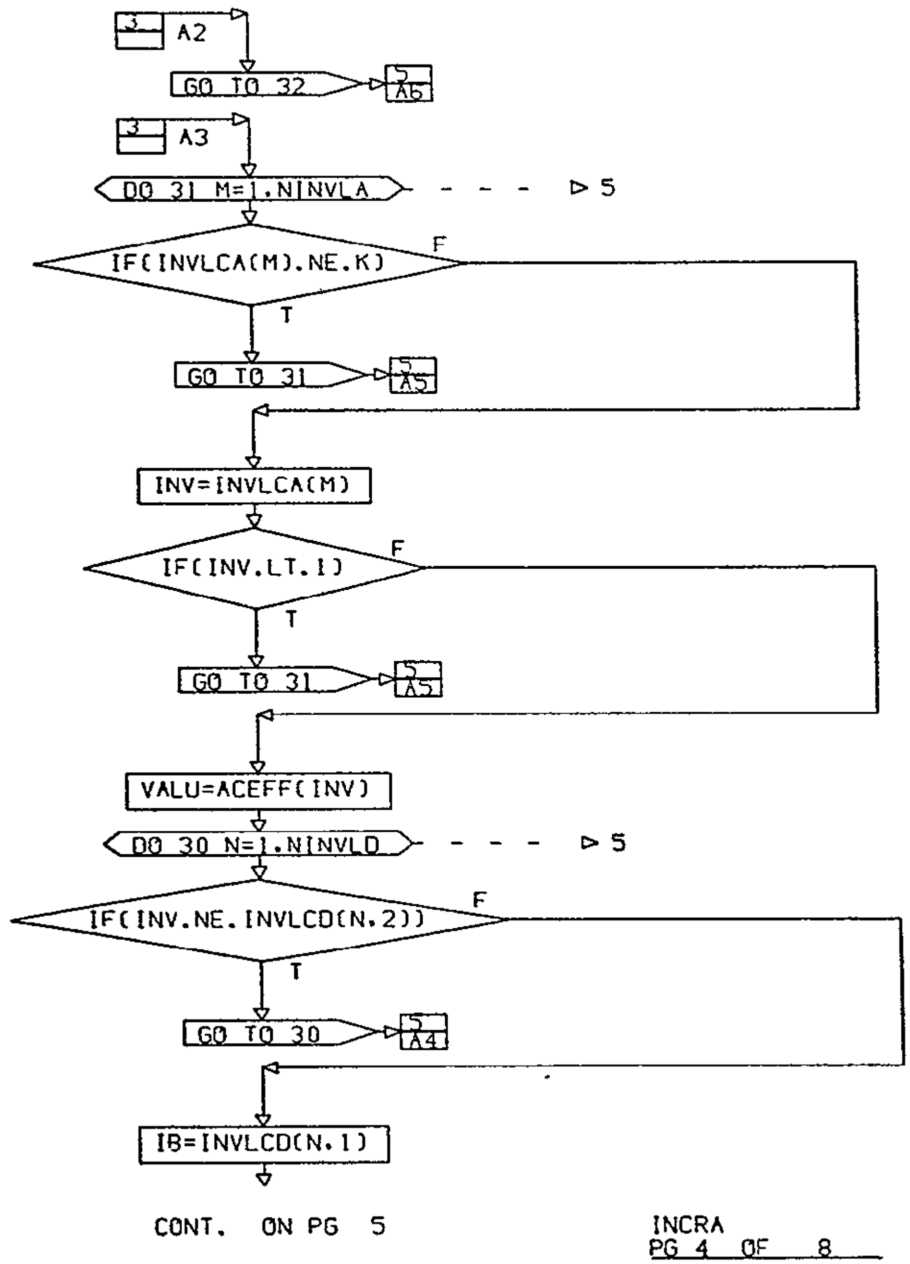
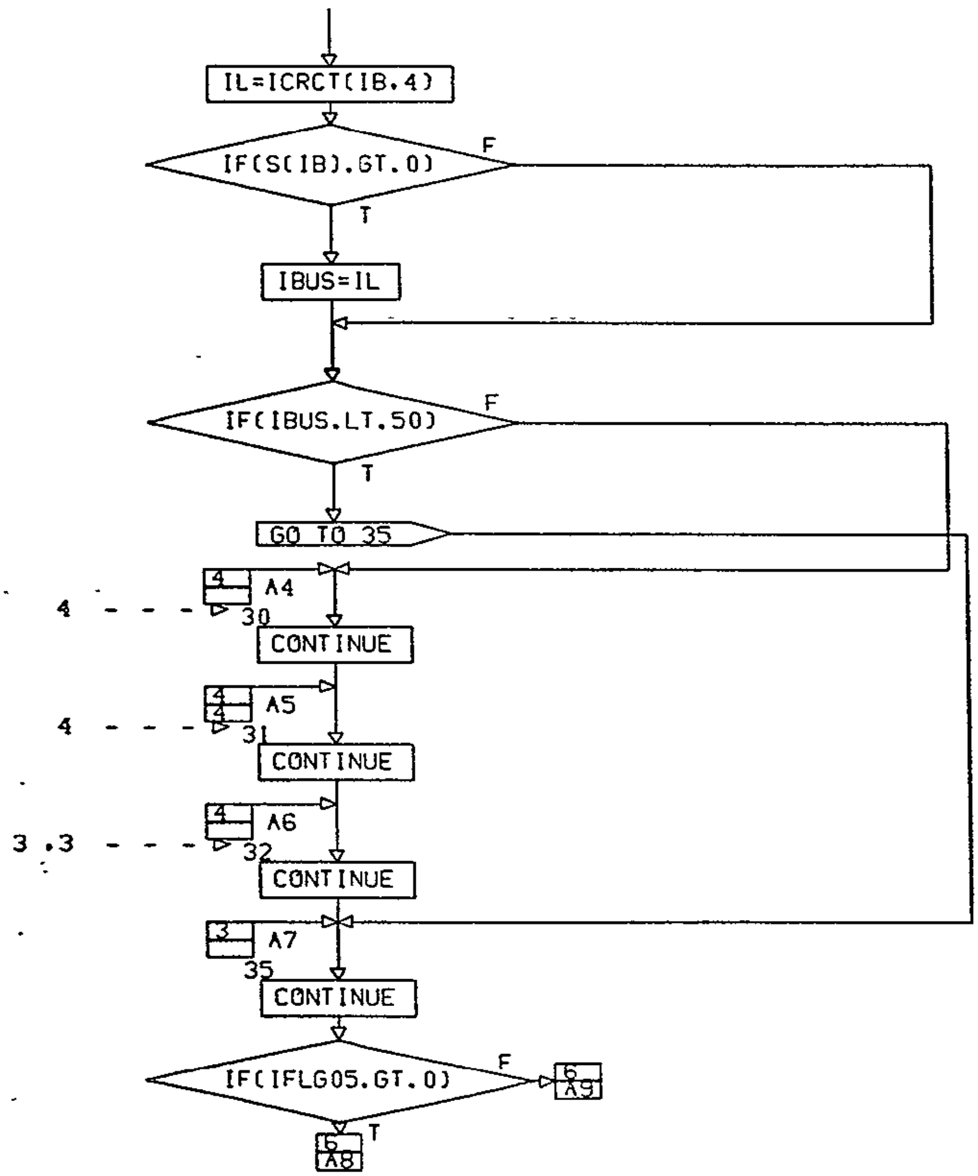


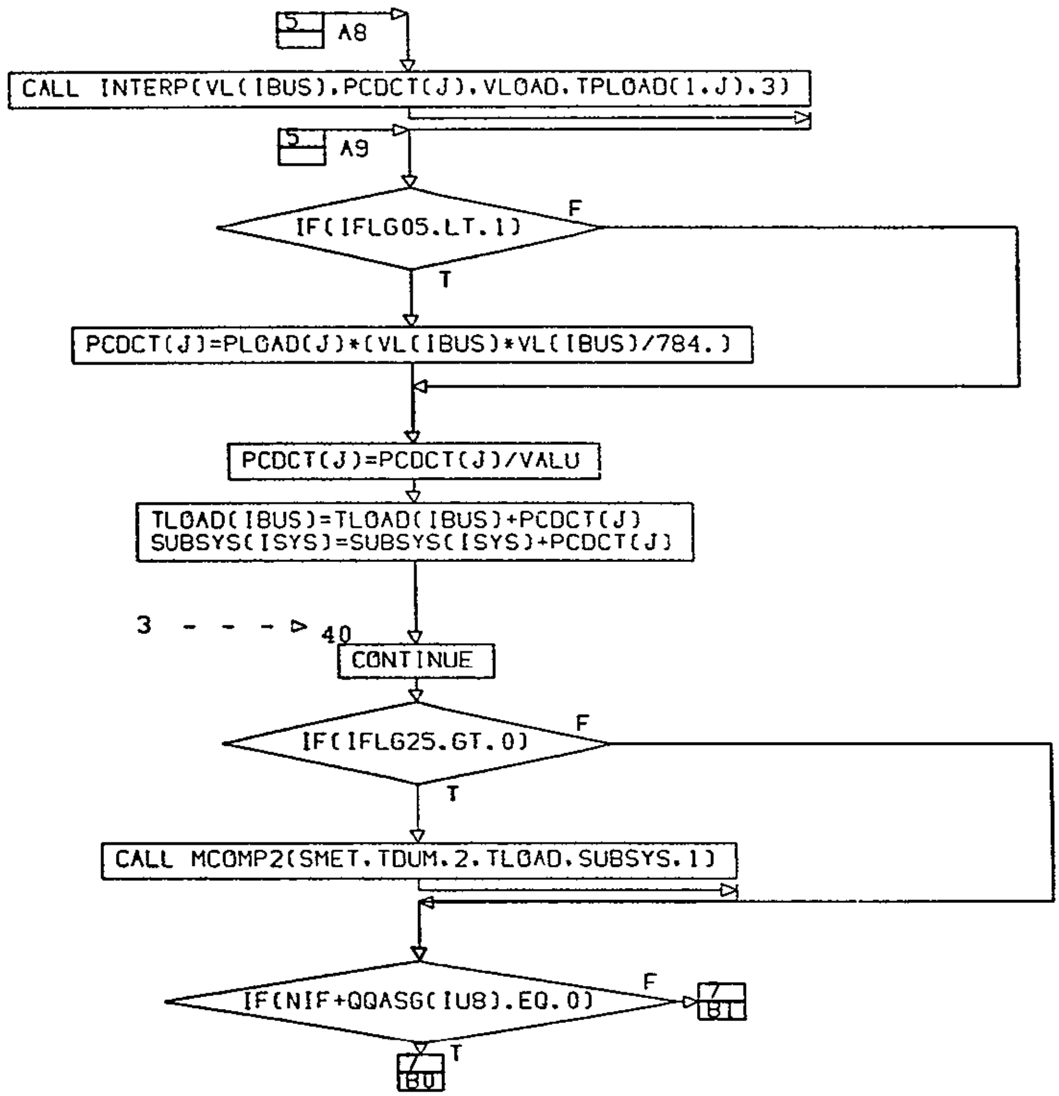
FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)



CONT. ON PG 6

INCRA
PG 5 OF 8

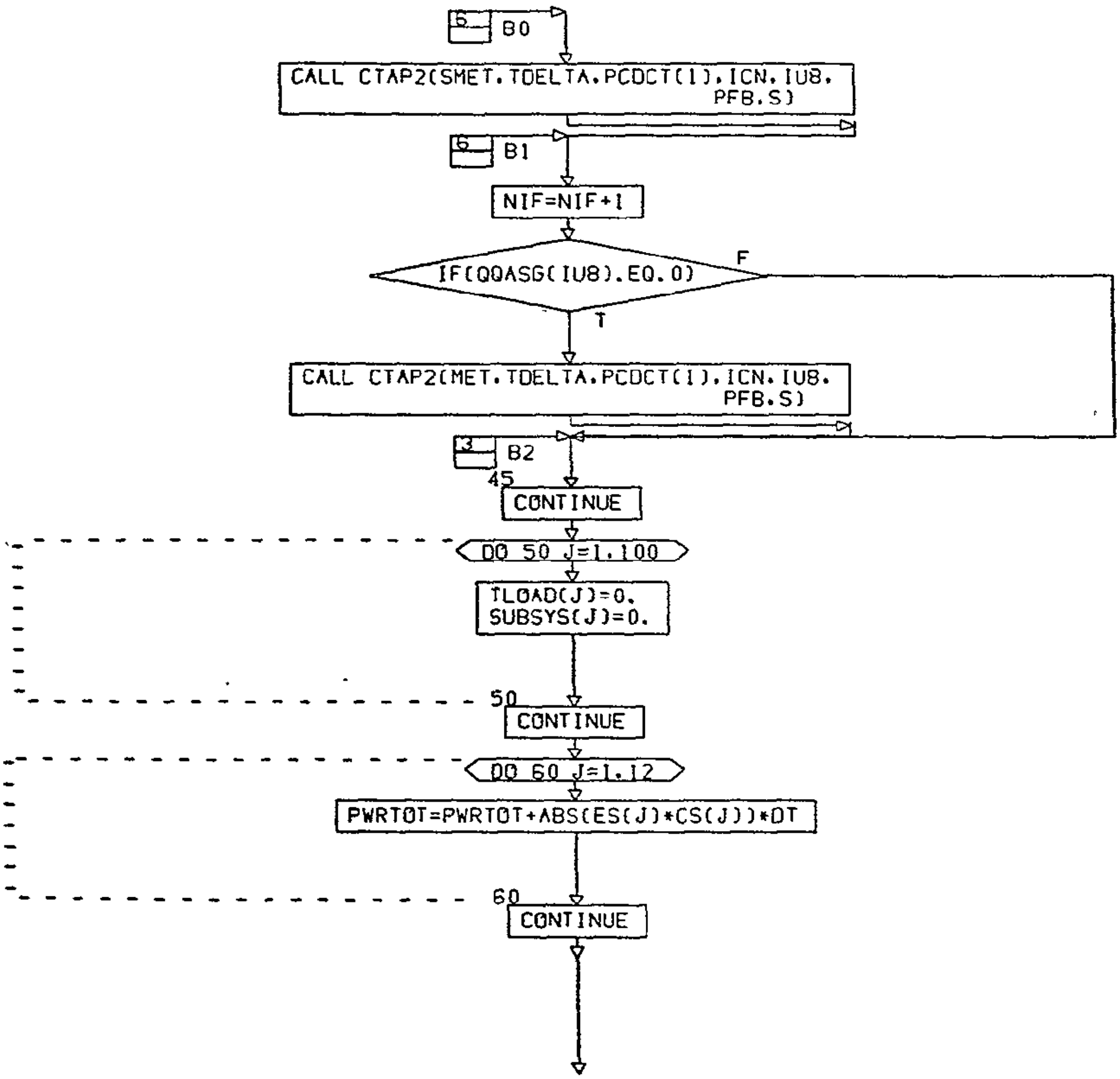
FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)



CONT. ON PG 7

INCRA
PG 6 OF 8

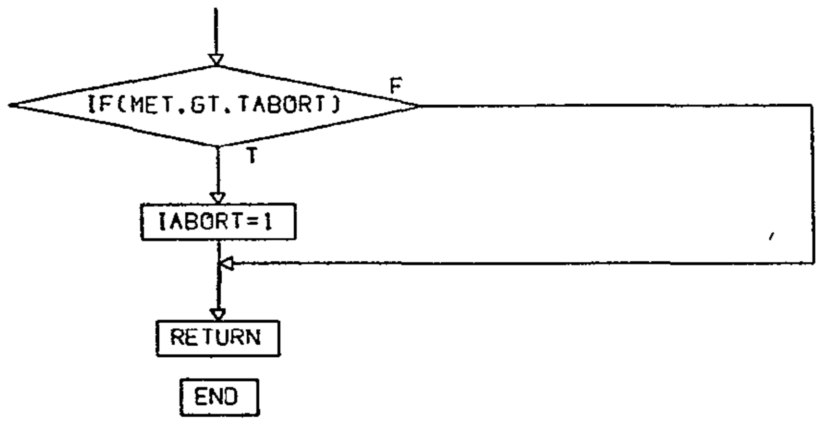
FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)



CONT, ON PG 8

INCRA
PG 7 OF 8

FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)



INCRA
PG 8 FINAL

FIGURE 3.3.8. FUNCTIONAL FLOWCHART OF SUBROUTINE INCRA (CONTINUED)

3.3.9 Subroutine: INITAL

PURPOSE: To prepare Phase2 for execution.

METHOD: All initialization procedures are followed and all necessary initial quantities are calculated.

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.9. See Appendix for definition of all variables.

```
SUBROUTINE INITAL
INCLUDE STRAG1
INCLUDE STRAG2
INCLUDE STRAG3
INCLUDE STRAG4
INCLUDE STRAG5
INCLUDE STRAG6
DIMENSION A(80)
```

```
DIMENSION TLOAD(100)
DIMENSION SUBSYS(100)
INTEGER QQASG
DATA S/100*1/
DATA TREADC /1.E6/
DATA TREADT /1.E6/
DATA NSC /12*25/
DATA CHRGLD /30./
```

```
DATA NFCTA/9/
DATA TLOAD /100*0./
DATA SUBSYS /100*0./
DATA MET / -999./
```

```
NAMELIST /CRDSF /
ACPOW.CSUBD.DELTA.EFFAVR.FCT.FCTA.FCTN.FCWP1.FCWP2.
HPT.HR.HUR.H2T.IT.NFCTA.NOSOCC.NSOCA.NSC.
OPT.OR.OUR.O2T.
PFAC.PIH.PIO.PP.PR.P1.P2.P3.P4.R.
S.SOC.SSTVI.TB.TMAXFC.TPH.TPO.WPR.XNC.SOCA
```

```
NAMELIST /CRDSI /
ACPOW.DELTA.FCT.H2T.INVLCA.MET.O2I.PFAC.PP.PR.S.SOC.TB.
NSC.MPRINT.MPRNT1.MPRNT2.MPRNT3.MPRNT4.MPRNT5.MPRNT6
```

```
NAMELIST /CRDST /
ACPOW.DELTA.CHRGLD.FCT.IABORT.ICHRG.INVLCA.ISOLVC.
NOR.NPRT.PFAC.PP.PR.S.SOC.TB.TDELTA
```

```
WRITE(IU6.9000)
IU7 =IUNIT( 5)
IF7 =IFILE( 5)
IU8 =IUNIT(10)
```

CONT. ON PG 2

INITAL
PG 1 OF 10

FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL

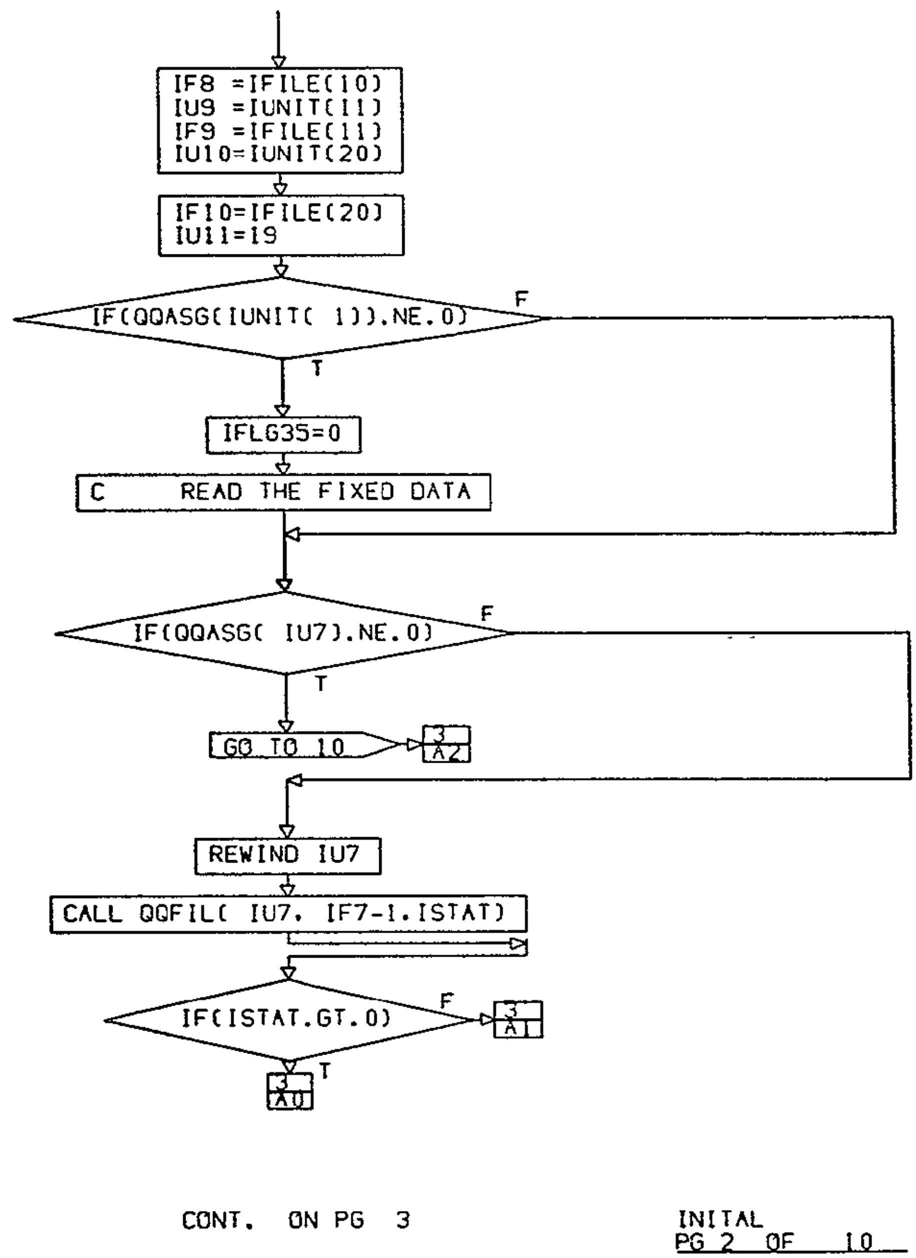


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

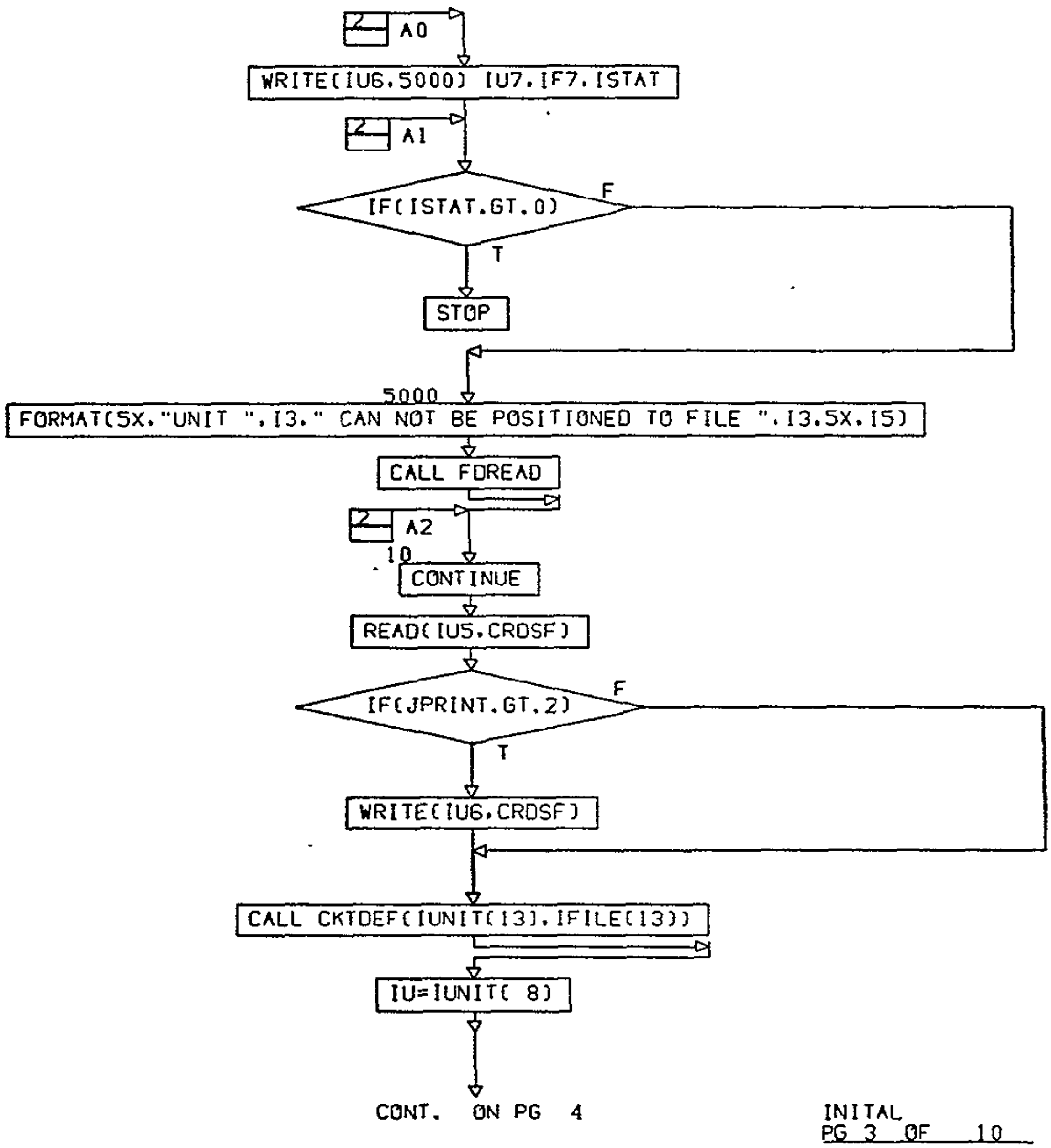


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

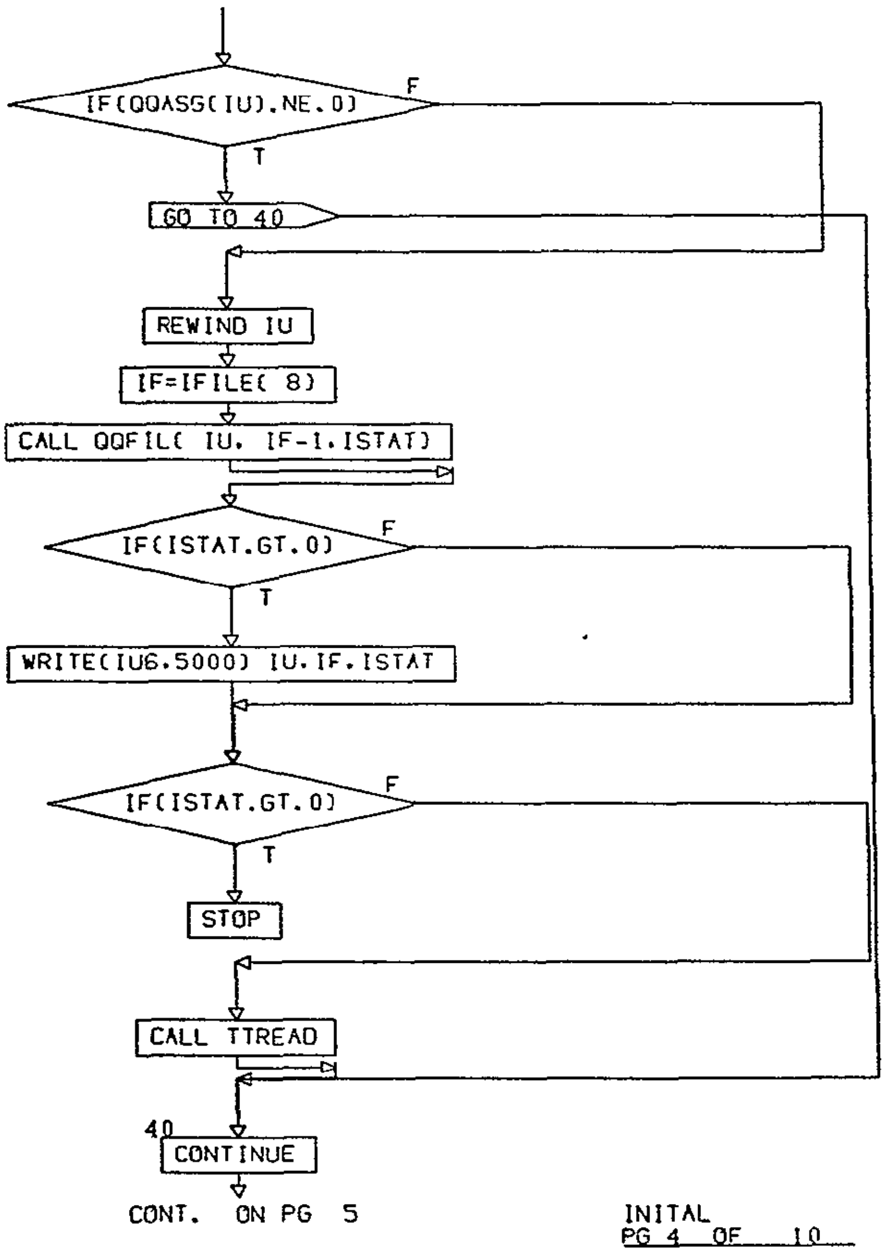
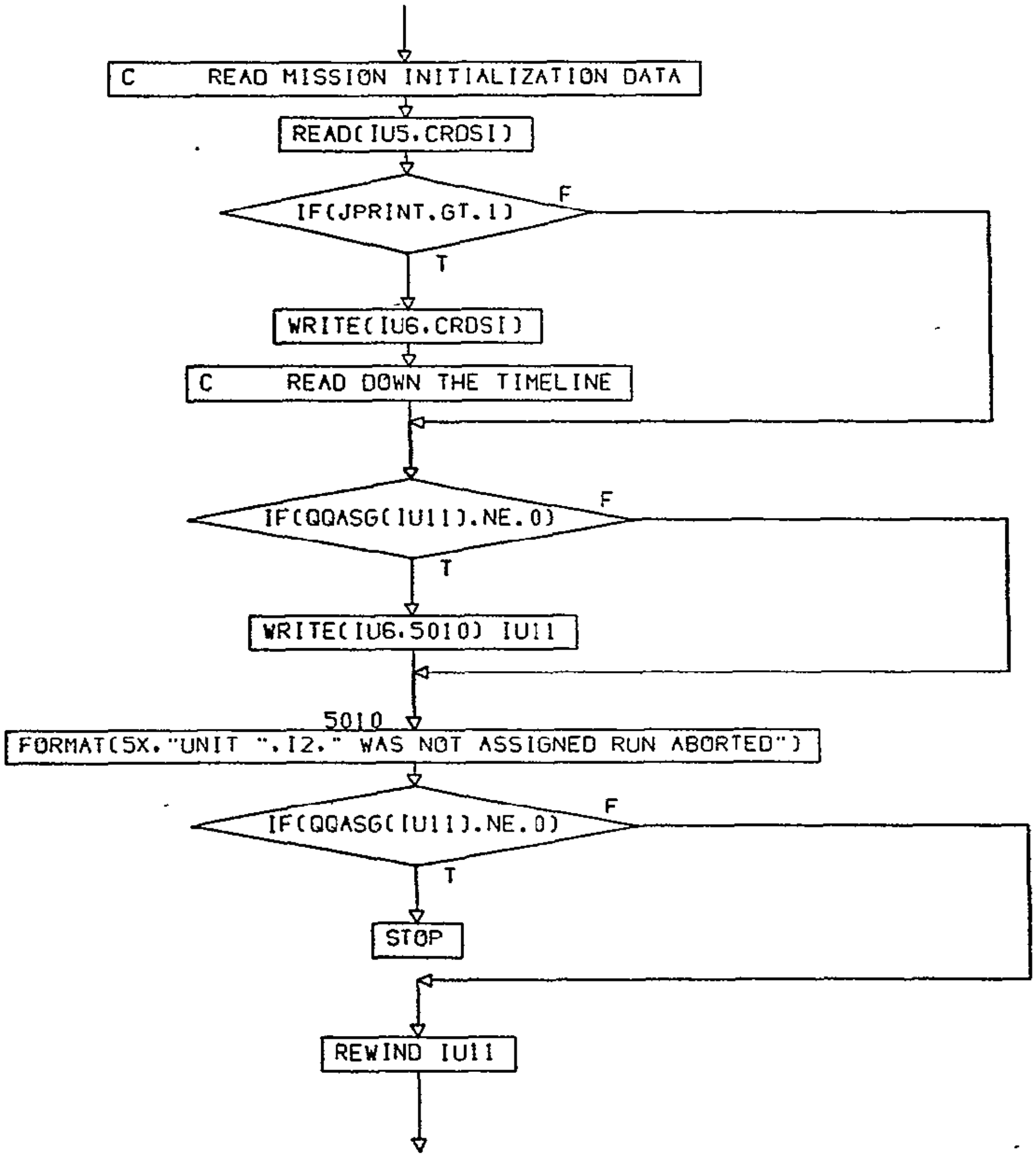


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)



CONT. ON PG 6

INITAL
PG 5 OF 10

FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

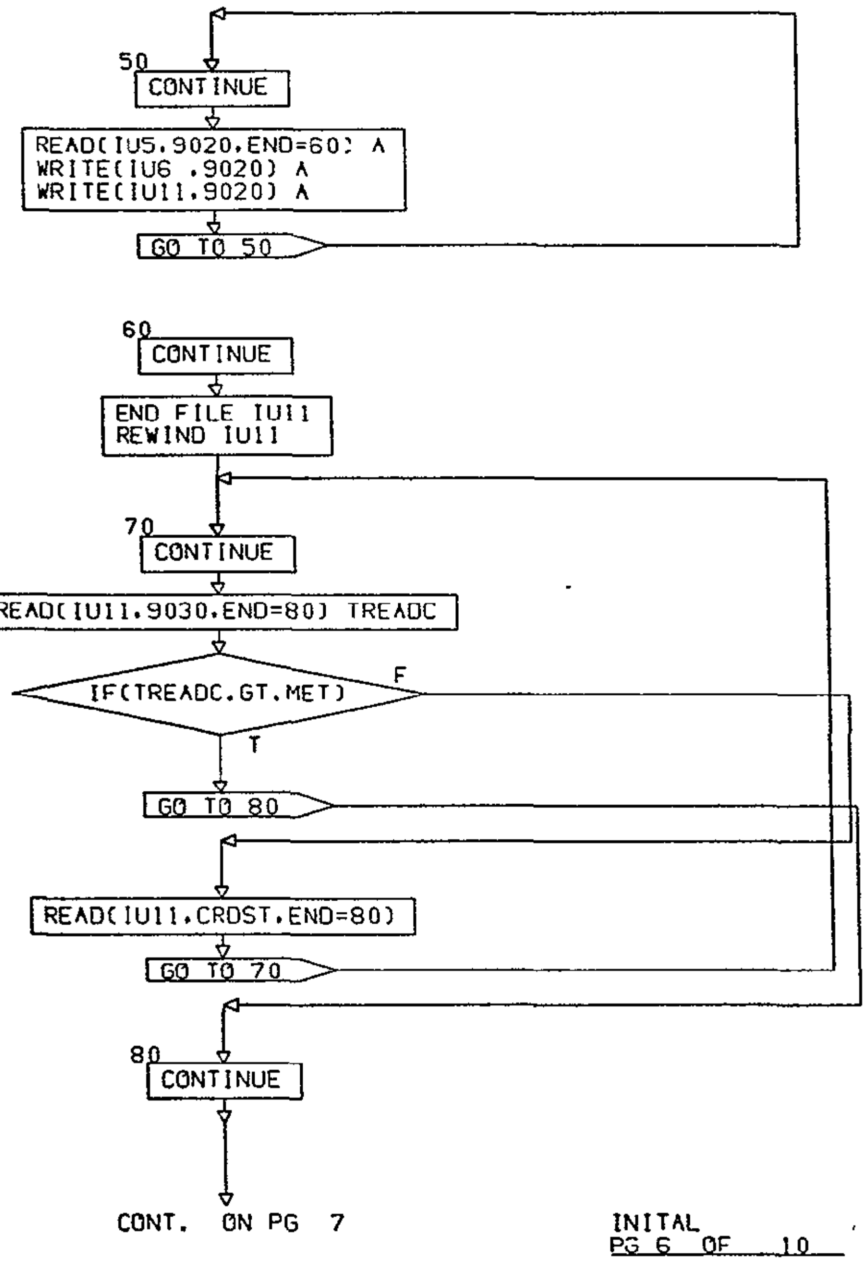
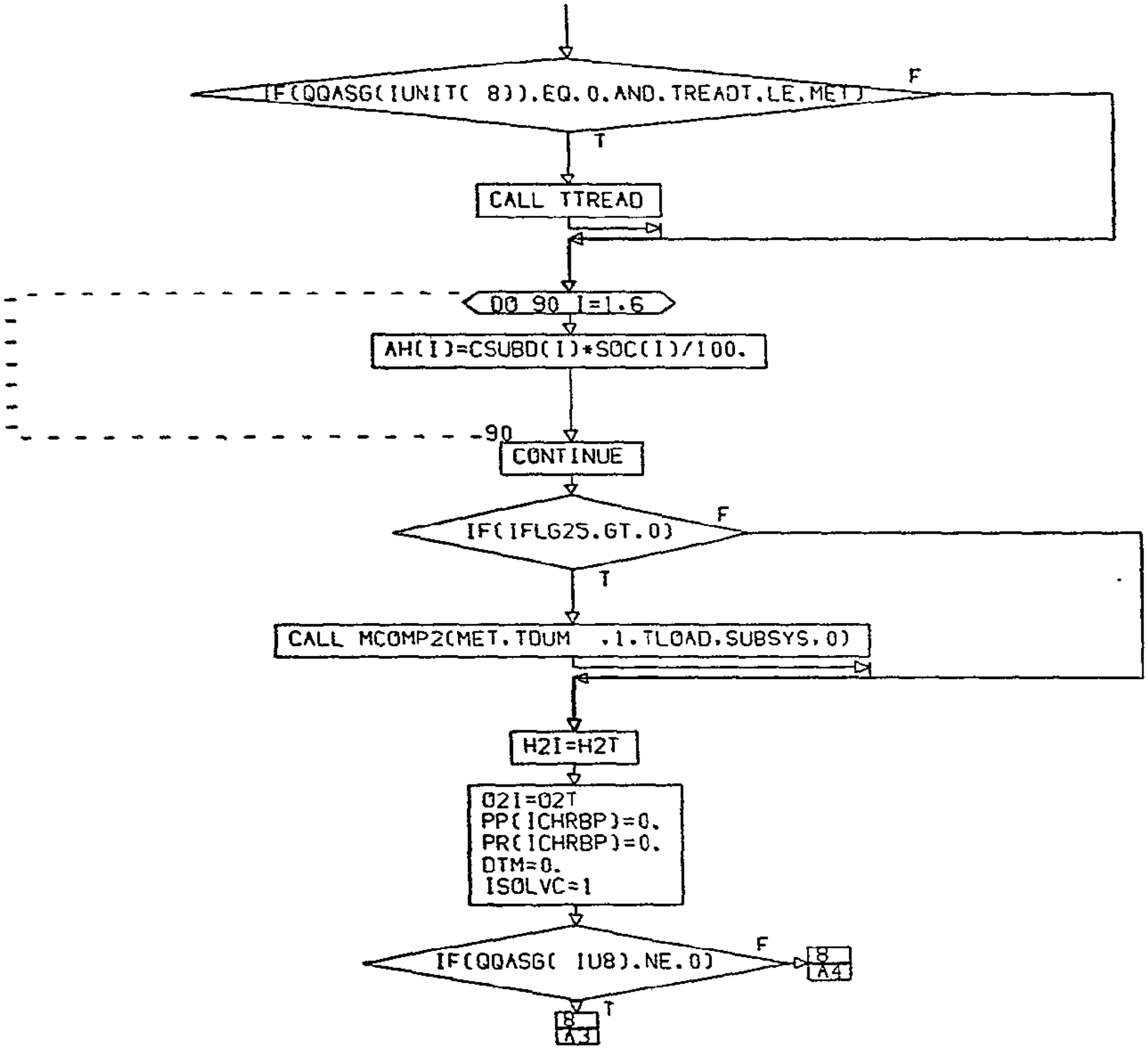


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)



CONT. ON PG 8

INITAL
PG 7 OF 10

FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

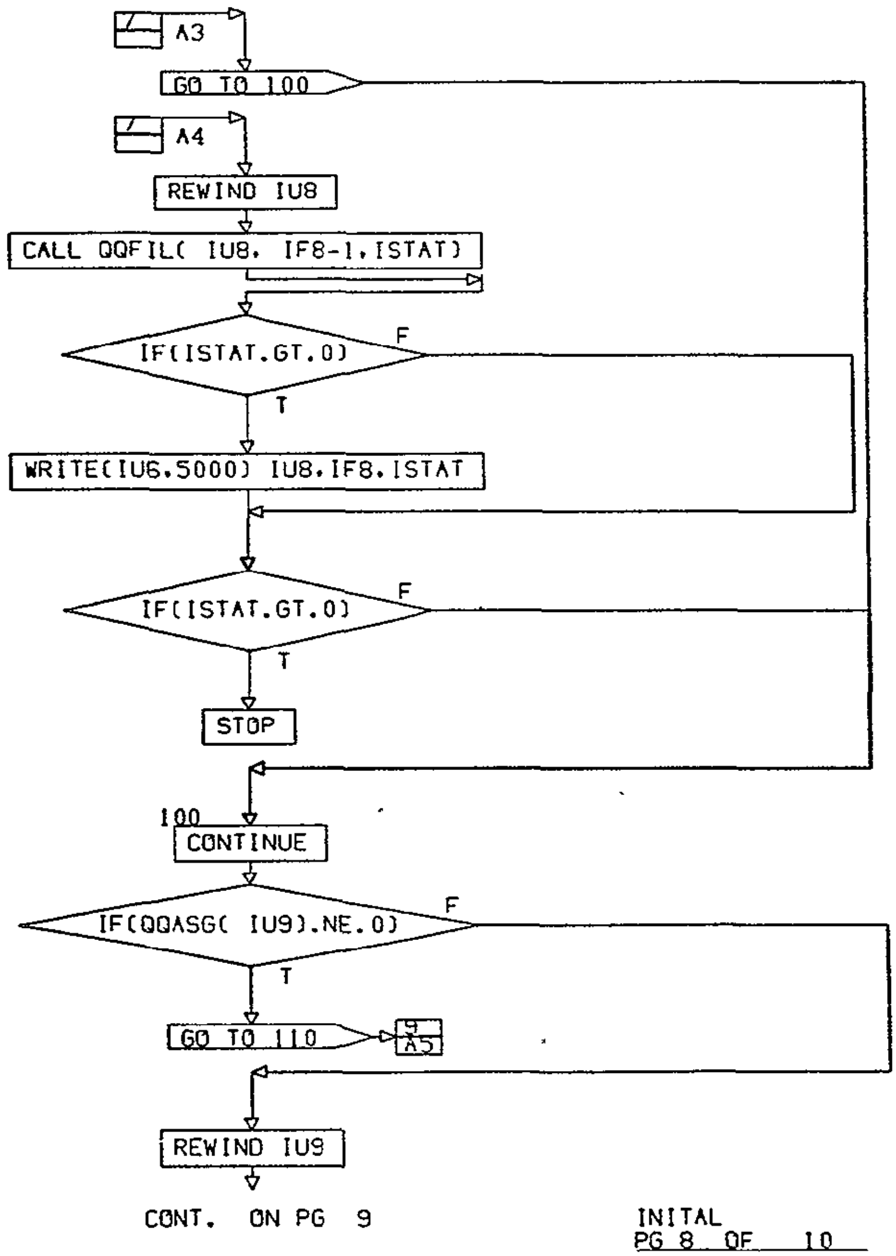


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

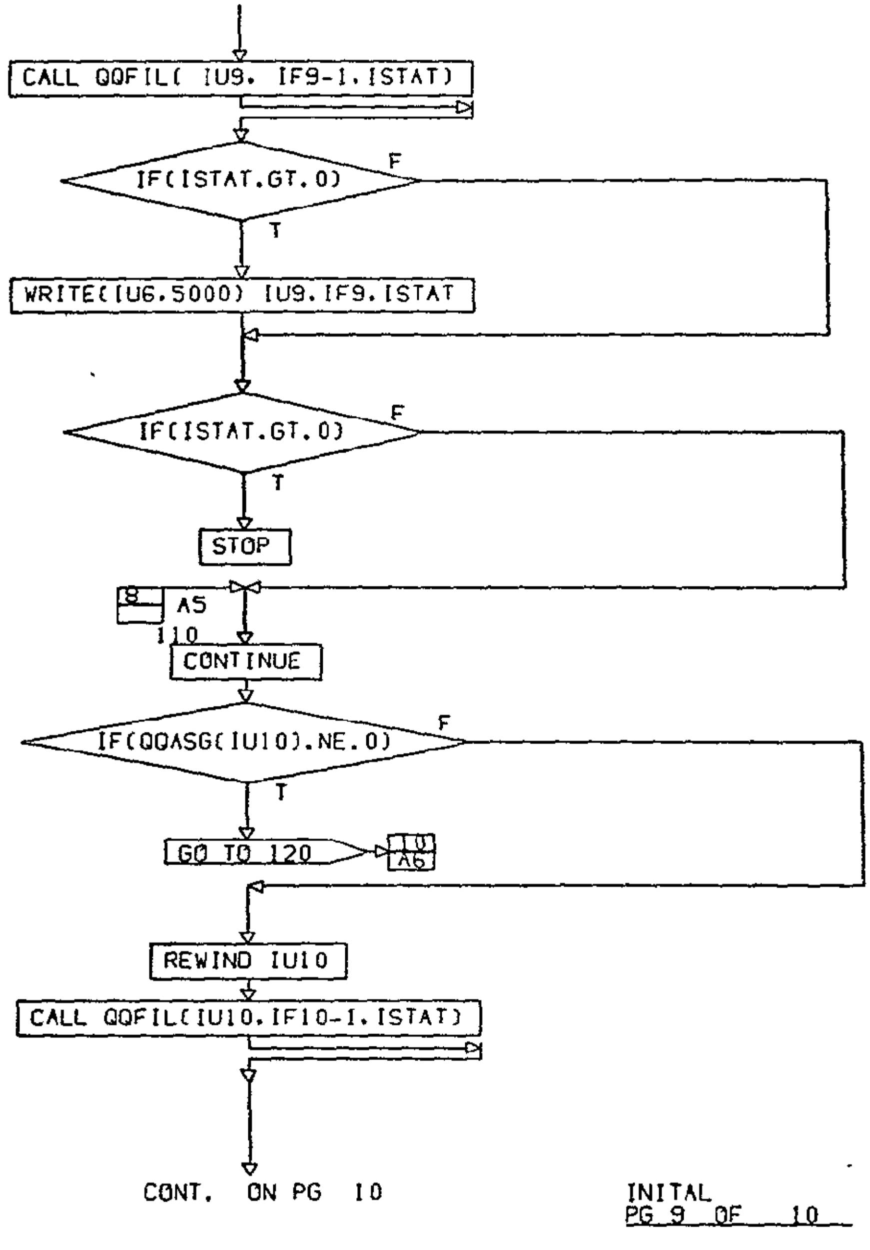


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

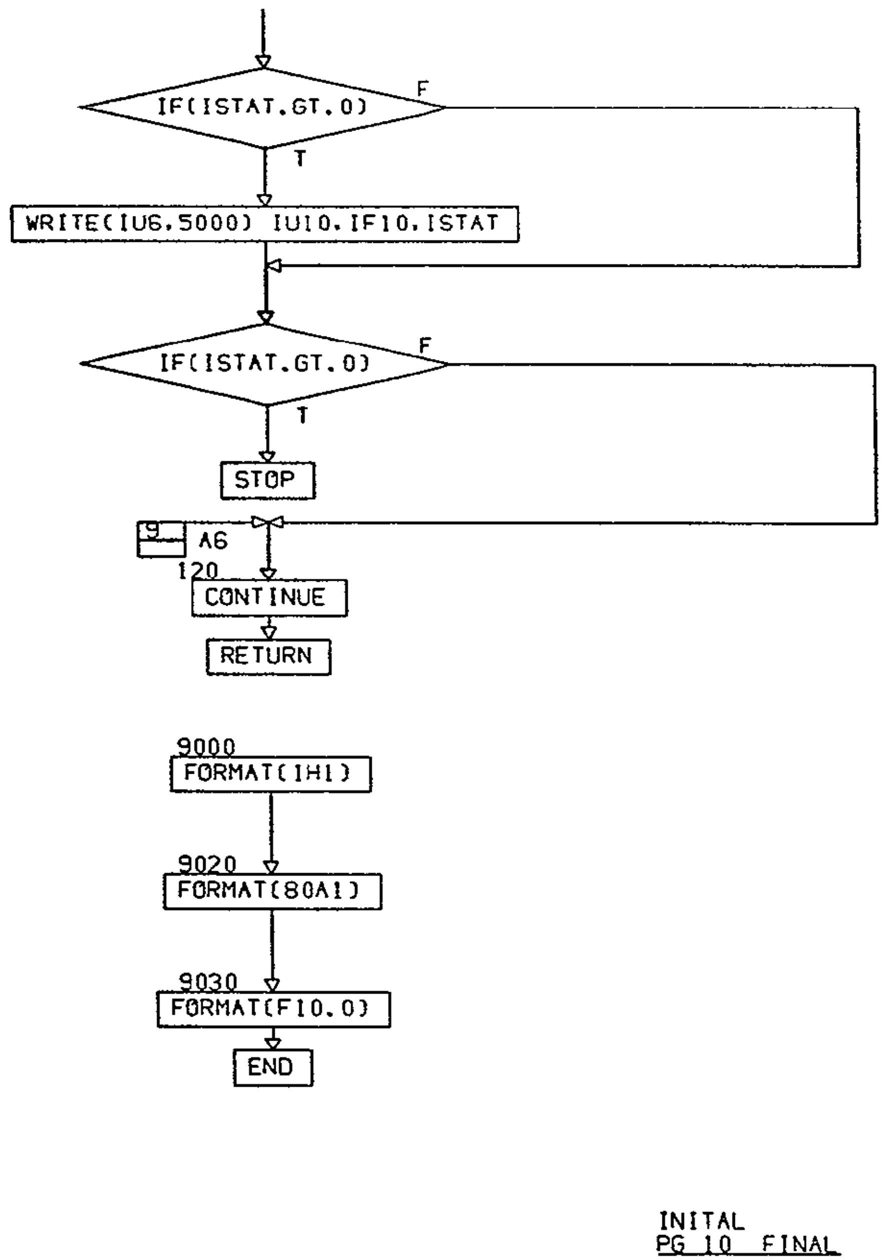


FIGURE 3.3.9. FUNCTIONAL FLOWCHART OF SUBROUTINE INITAL (CONTINUED)

3.3.10 Subroutine: QCLTMP

PURPOSE: To update battery status

METHOD: Based upon time elapsed, operating temperature, and operating current and voltage the following quantities are calculated:

1. Charge and discharge efficiency
2. Heat produced
3. Change in capacity
4. Temperature
5. New ampere-hour status
6. State-of-charge

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.10. See Appendix for definition of all variables.

```
SUBROUTINE QCLTMP(I,IACT)
INCLUDE STRAG1
INCLUDE STRAG2
INCLUDE STRAG4
```

```
C     PRIMARY BATTERY SUBROUTINE   CALCULATES A-H STATUS, SOC, DEGRATION
C     TURNS ON HEATERS, CALCULATES HEAT GENERATED, AND TAKES DEPLETED
C     BATTERIES OUT OF LINE
C                       INPUT
C        CS          OPERATING CURRENT OF BATTERY
C        CSUBD       BATTERY DESIGN CAPACITY
C        DT          TIMESTEP
C        EFF         DISCHARGING BATTERY EFFICIENCY
```

```
C          EFFAVR    CHARGING BATTER EFFICIENCY
C          P1,P2,P3,P4     CONSTANTS FOR HEAT GENERATION EQUATION
C          QHI       HEATER OUTPUT CAPABILITY (WATTS)
C          TB        BATTERY TEMPERATURE
C          VP        OPERATING VOLTAGE OF BATTERY
C          TRFC(J,I) TEMP VS TEMP RATE OF INCREASE
C                    I=1 TEMP RATE FOR CURRENT CDX1
C                    I=2 TEMP RATE FOR CURRENT CDX2
```

```
C                    I=3 TEMP RATE FOR CURRENT CDX3
C                    I=4 TEMP IN DEGREES F
C                    J= POINTS
C                        OUTPUT
C          AH        AMP HOUR STATUS OF BATTERY
C          CC        AMP HOUR CAPACITY OF BATTERY
C          DQ        HEAT PRODUCED (WATTS)
C          QH        HEATER OUTPUT (WATTS)
```

```
C          SOC       PRESENT STATE-OF-CHARGE OF THE BATTERY
C          TD        TIME ON LAST MAJOR BATTERY DISCHARGE
```

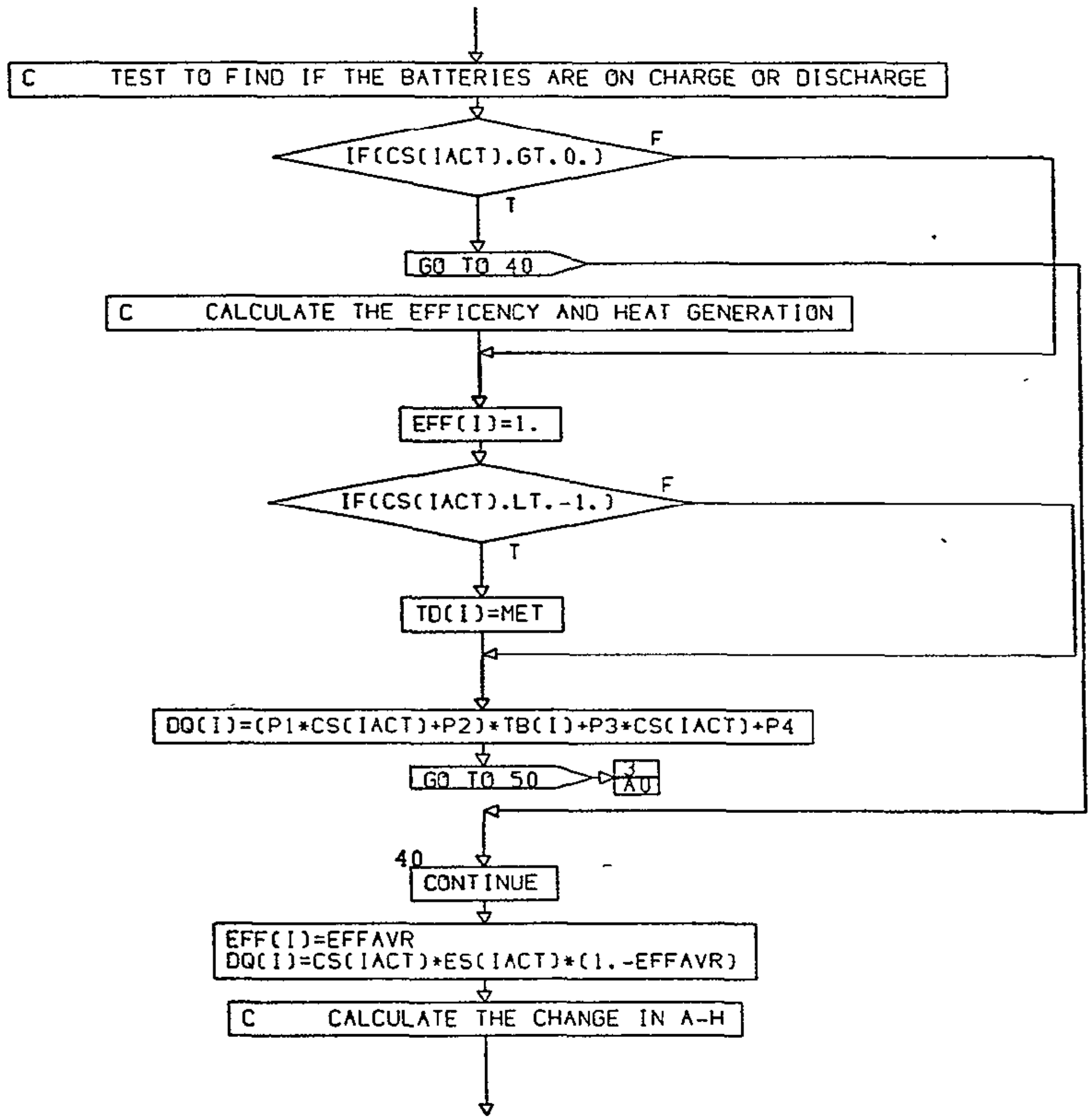
```
DIMENSION TRFC(8,4)
DATA CDX1,CDX2,CDX3 /15.,25.8,36.7/
```

```
DATA TRFC /7.,6.6,5.4,3.8,1.5,-1.2,-5.,-9.4,
          9.2,9.,8.1,6.8,5.,2.7,0.,-3.8,
          12.6,12.5,12.,11.2,9.8,8.,5.9,3.,
          30.,40.,60.,80.,100.,120.,140.,160.     /
```

CONT. ON PG 2

QCLTMP
PG 1 OF 6

FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP



CONT. ON PG 3

QCLTMP
PG 2 OF 6

FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP (CONTINUED)

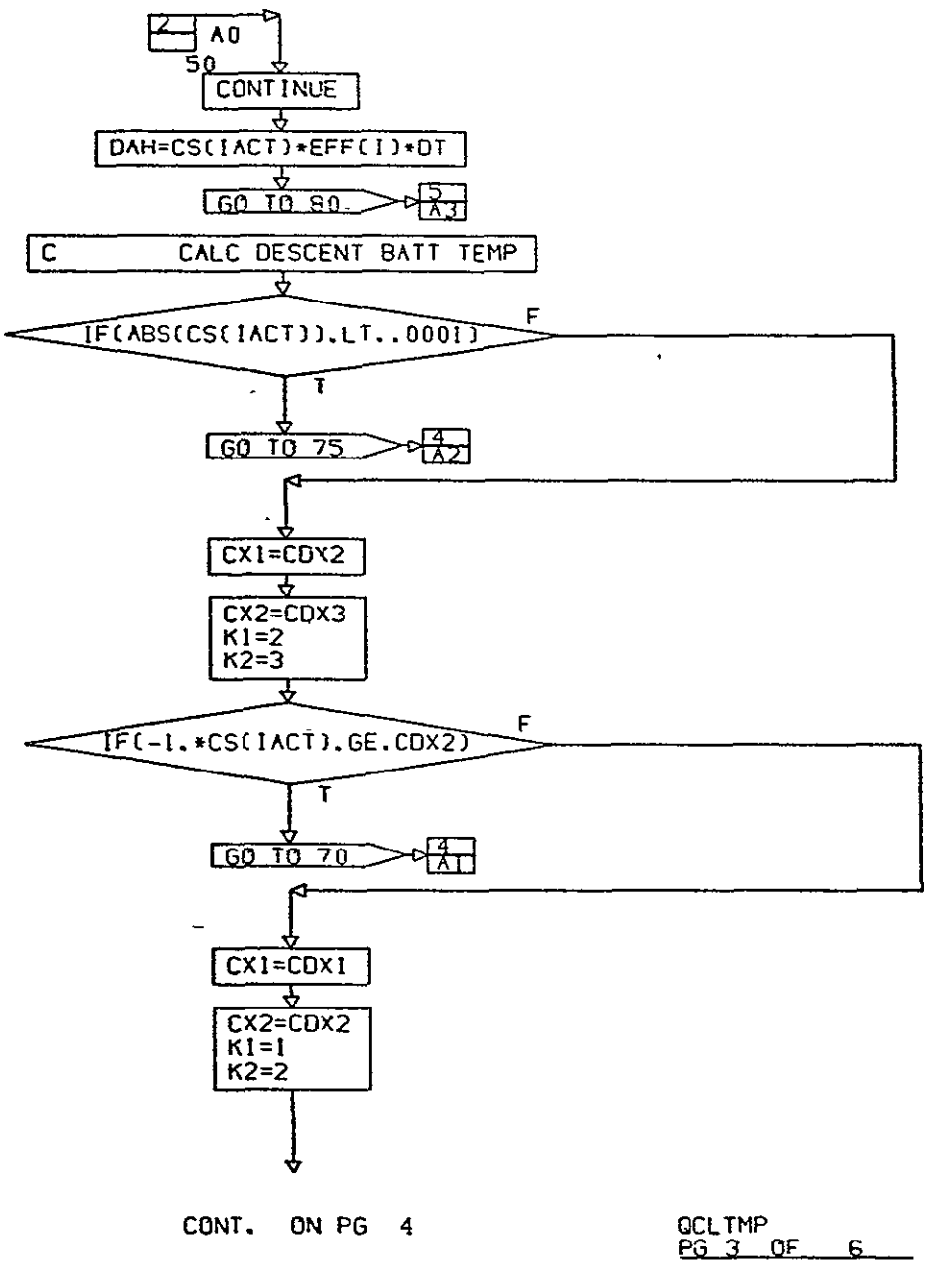
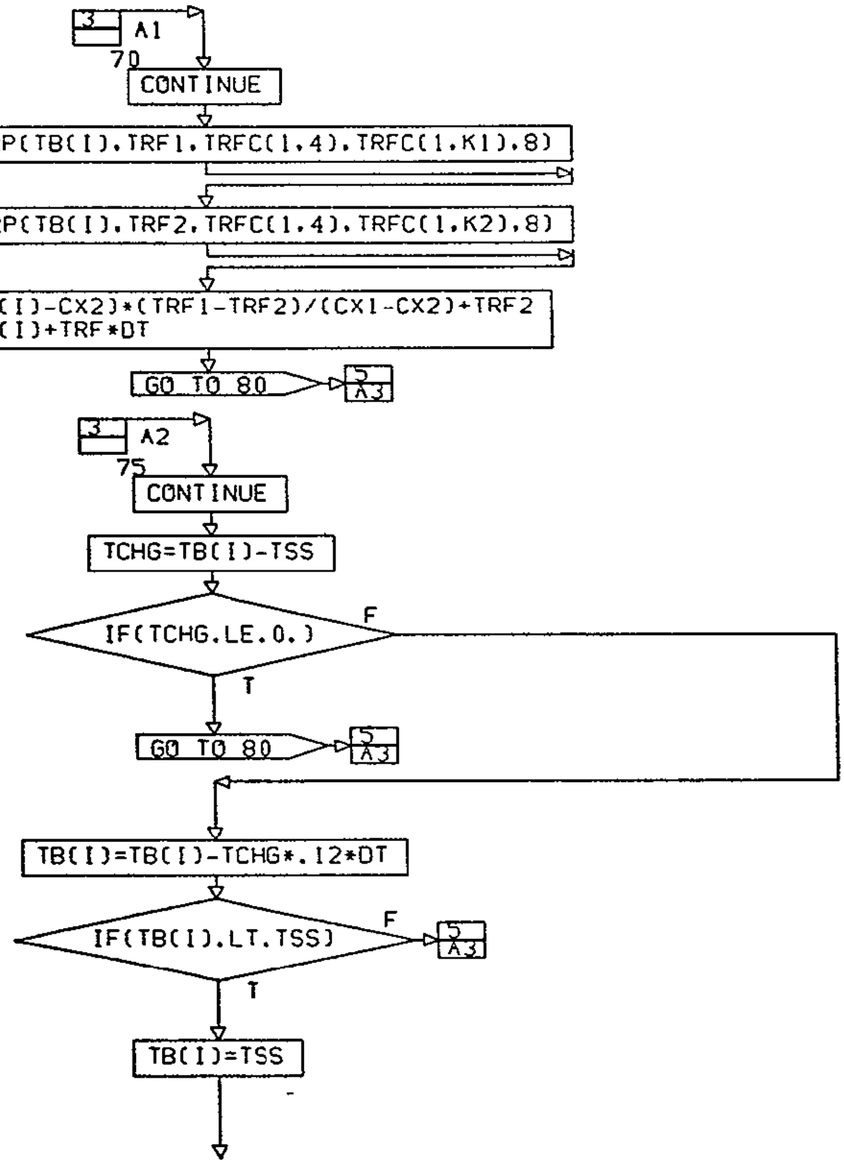


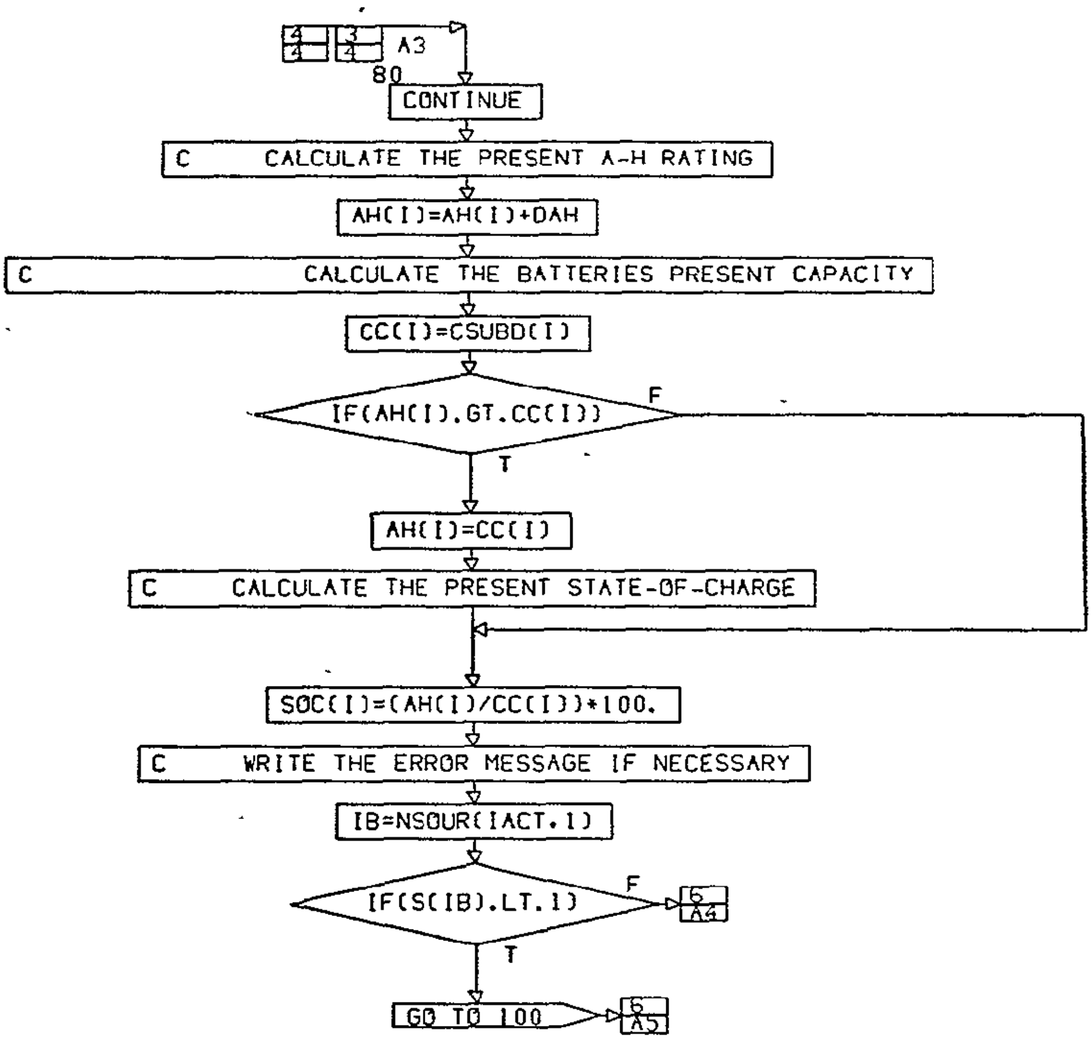
FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP (CONTINUED)



CONT. ON PG 5

QCLTMP
PG 4 OF 6

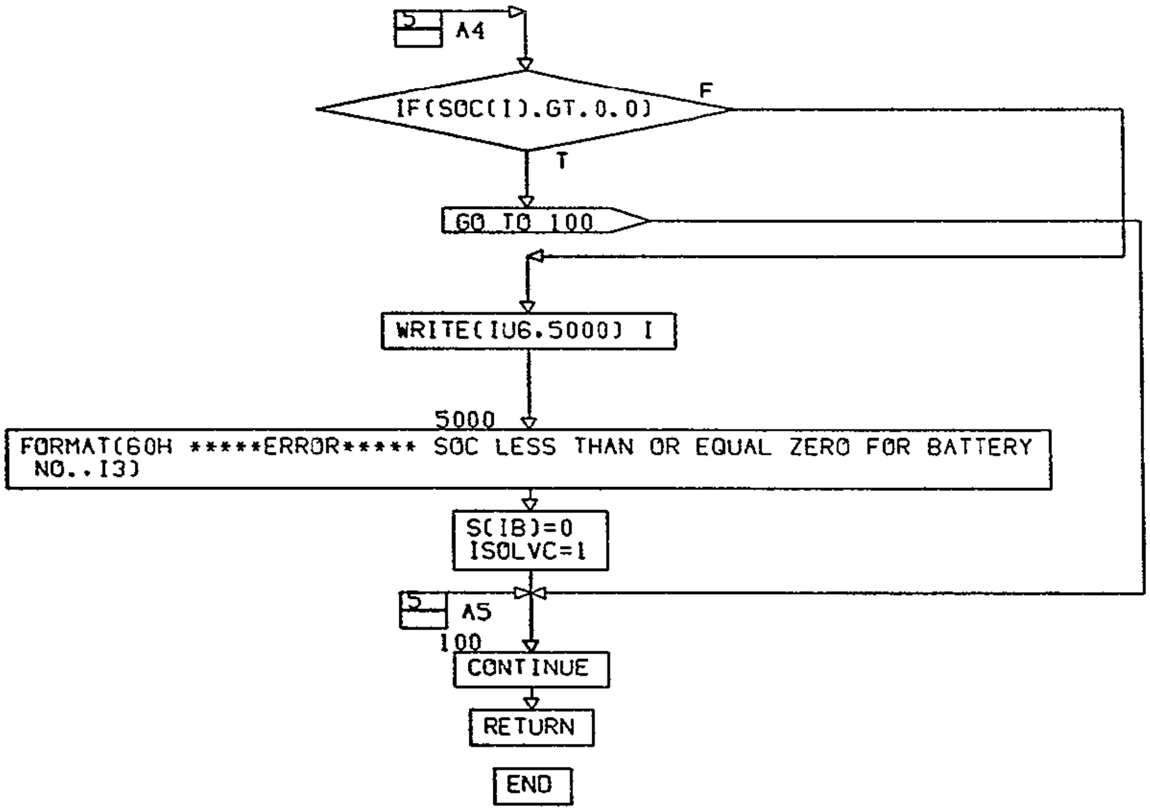
FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP (CONTINUED)



CONT. ON PG 6

QCLTMP
PG 5 OF 6

FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP (CONTINUED)



QCLTMP
PG 6 FINAL

FIGURE 3.3.10. FUNCTIONAL FLOWCHART OF SUBROUTINE QCLTMP (CONTINUED)

3.3.11 Subroutine: REDLIN

PURPOSE: To detect and identify EPS values which violate established limits

METHOD: The subroutine consists of logic for tests on six individual EPS parameters. The parameters are:

1. Node Voltage - tested for violation of under voltage limit

2. Inverter Volt-amps - checked for overload

3. Branch Current - tested for current limit

4. Fuel Cell Power

   a. Peak Power - tested for power level exceeding peak power limit

   b. Continuous Power - tested for power level exceeding continuous power limit

   c. Minimum Power - tested for power level under minimum power level

5. Cryogen Level

   a. O_2 - checked for depletion

   b. H_2 - checked for depletion

6. Battery State-of-Charge - checked for depletion

The tests are made by comparing an EPS parameter value to its appropriate limit from the fixed data tape. If the parameter value violates the limit, a message is constructed which contains the following data:

1. Time of Violation

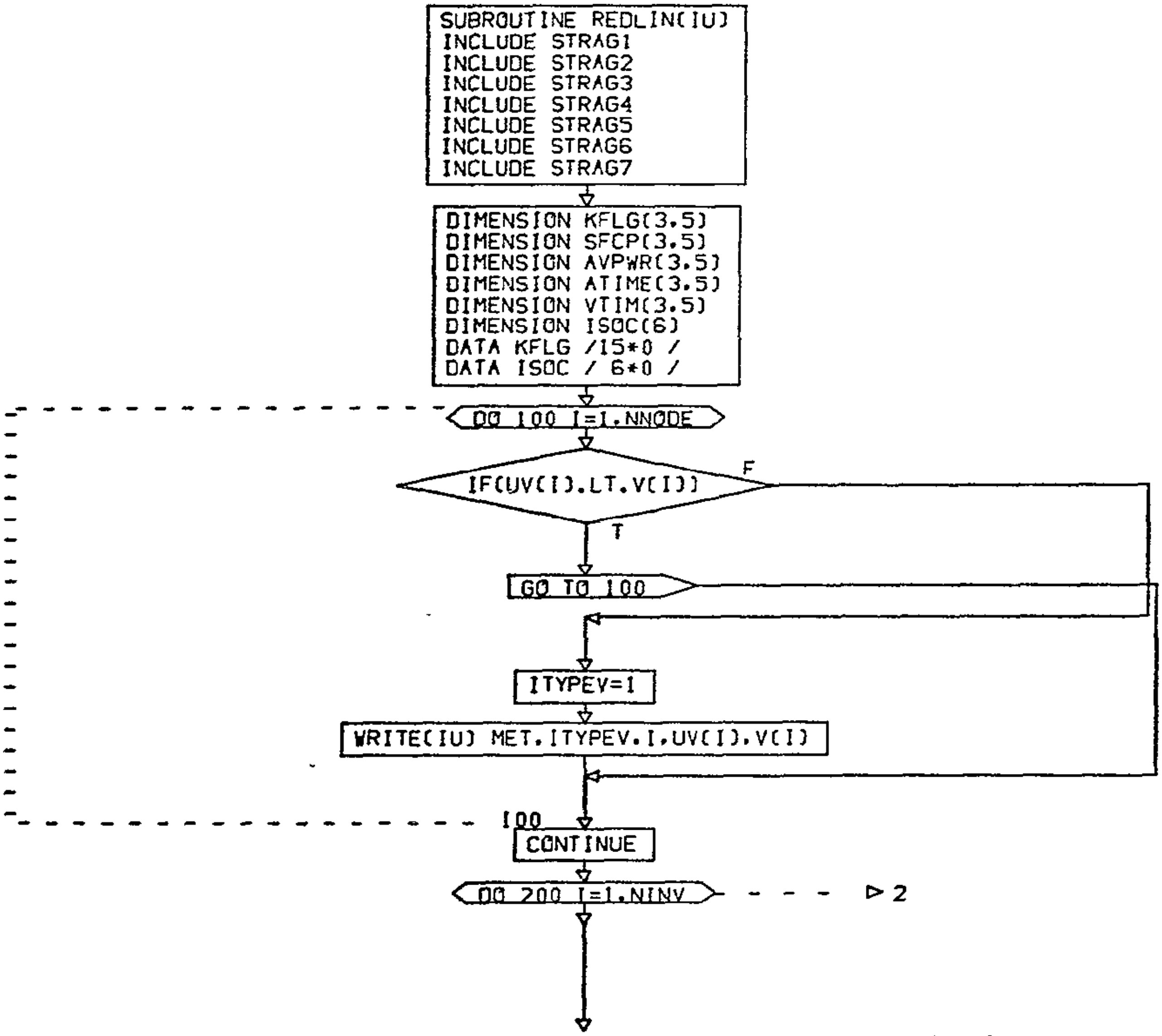
2. Type of Violation

3. Value of Limit

4. Parameter Value

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.3.11. See Appendix for definition of all variables.

PRECEDING PAGE BLANK NOT FILMED



CONT. ON PG 2

REDLIN
PG 1 OF 9

FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN

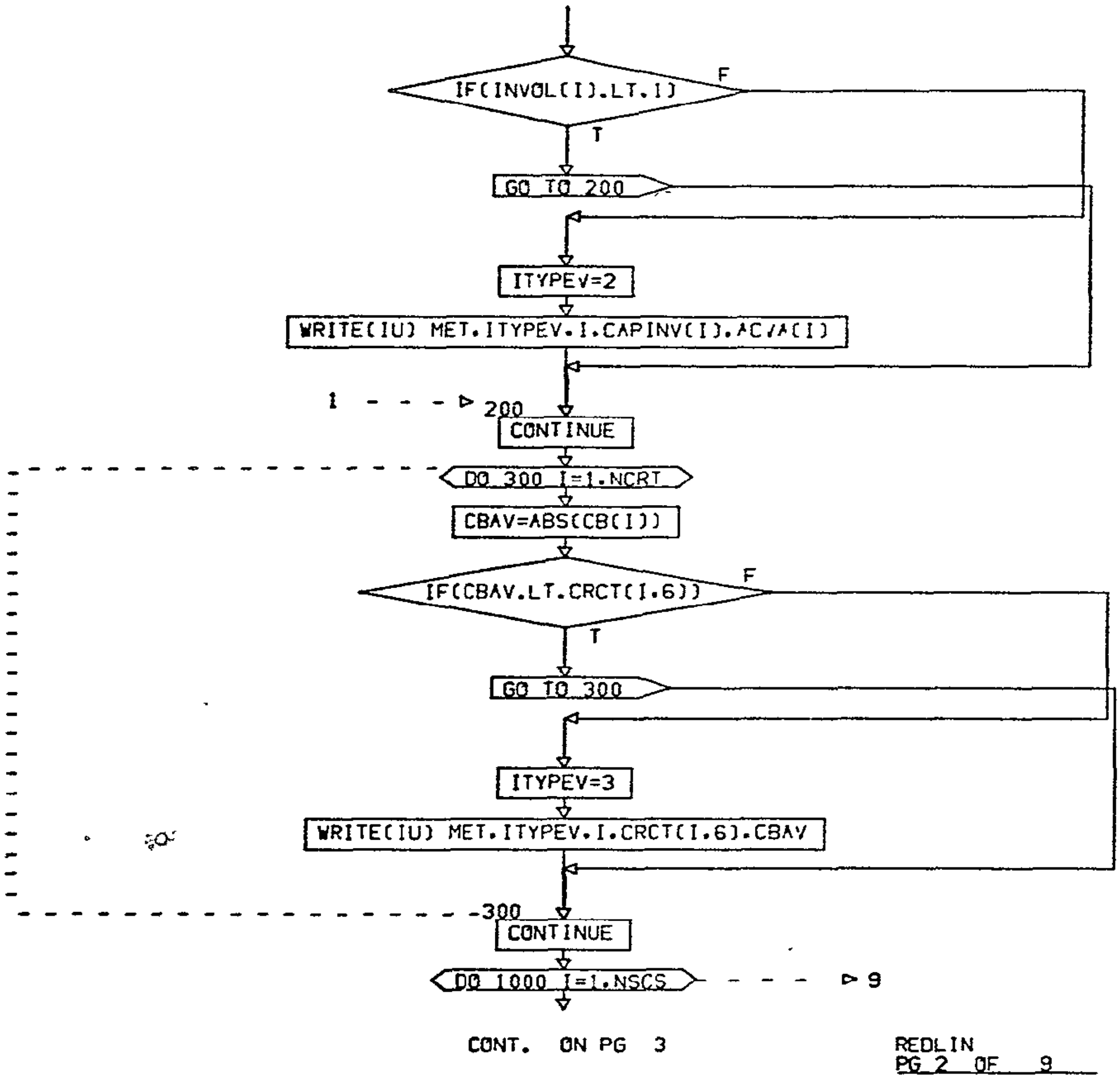


FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)

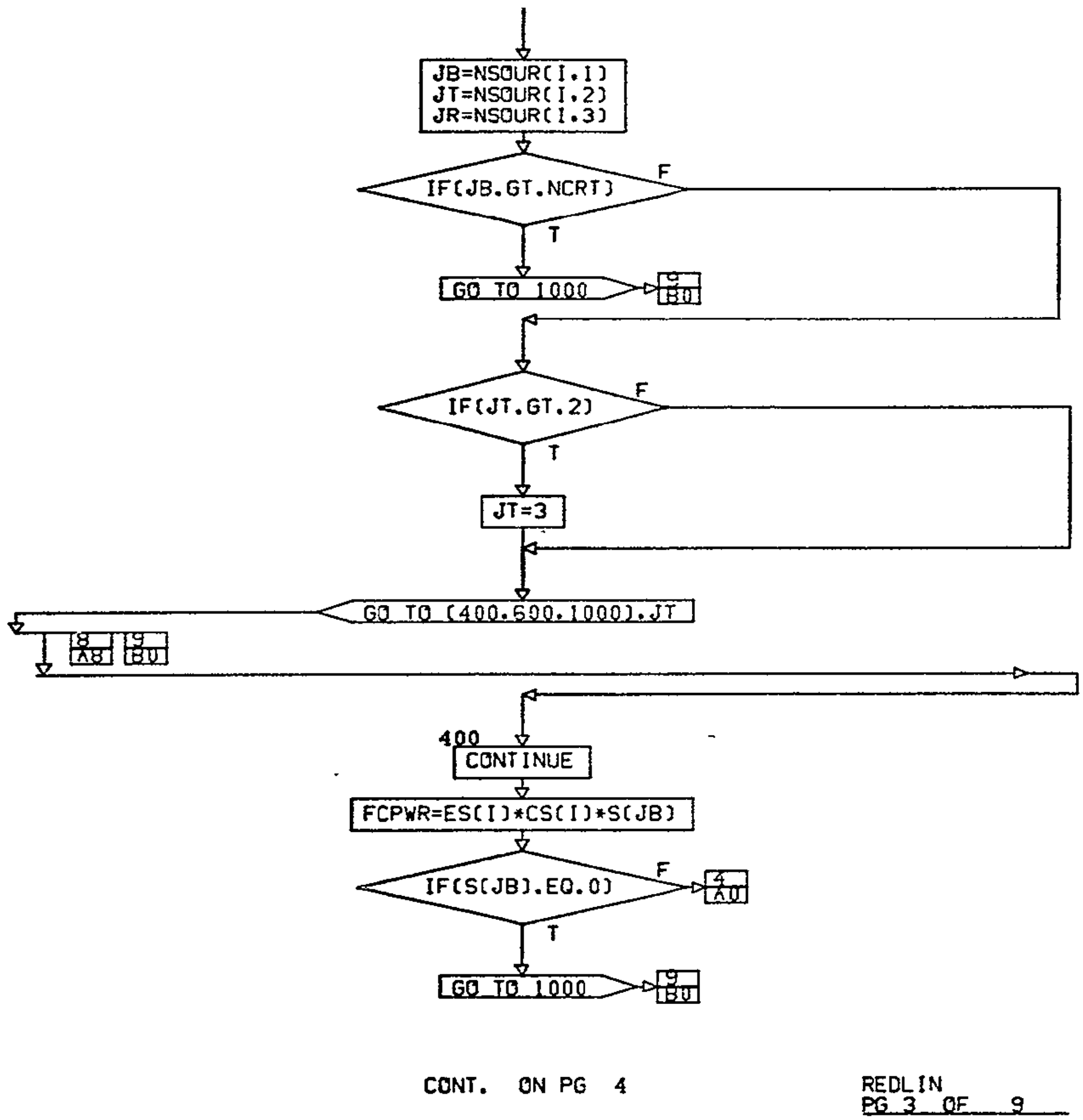


FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

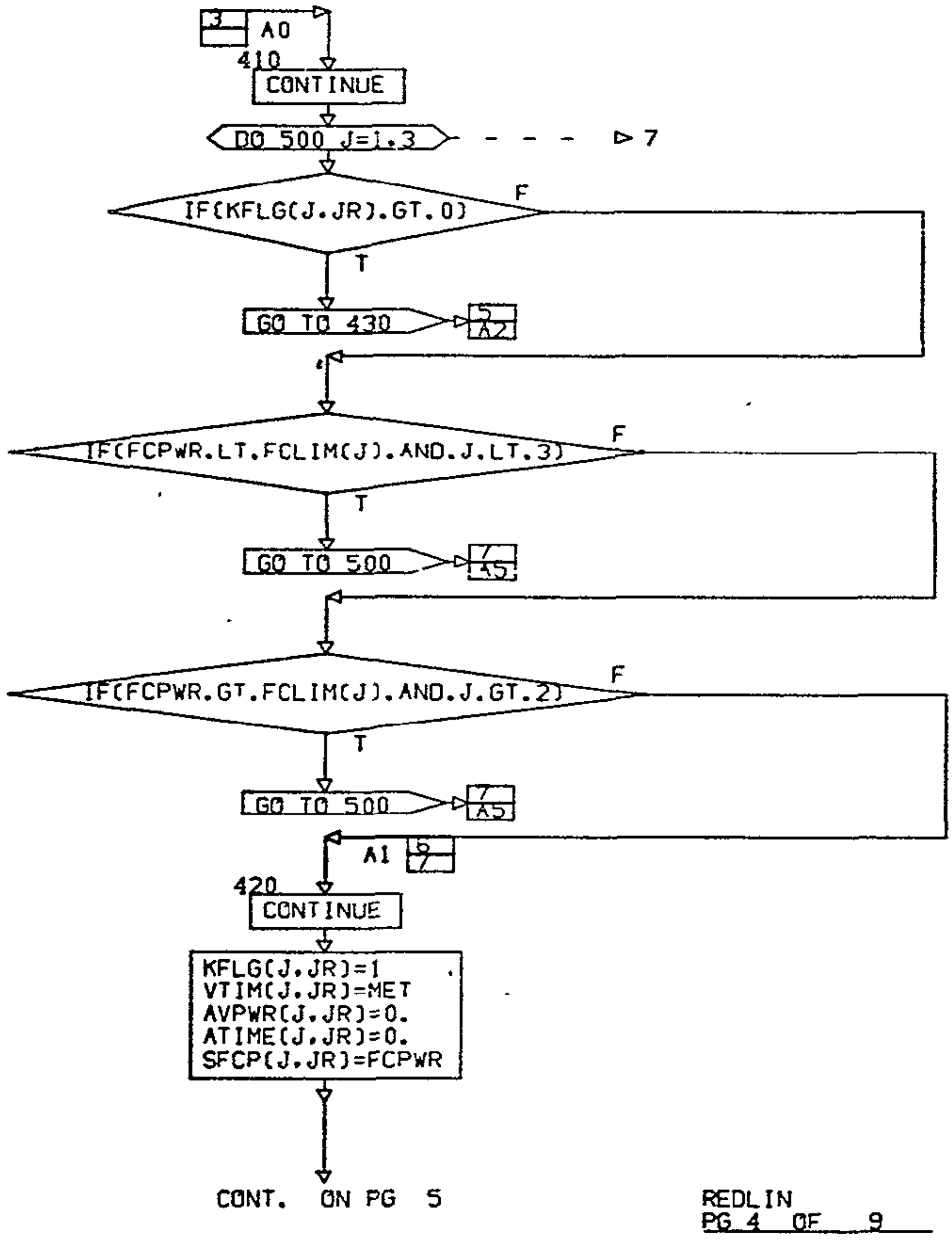
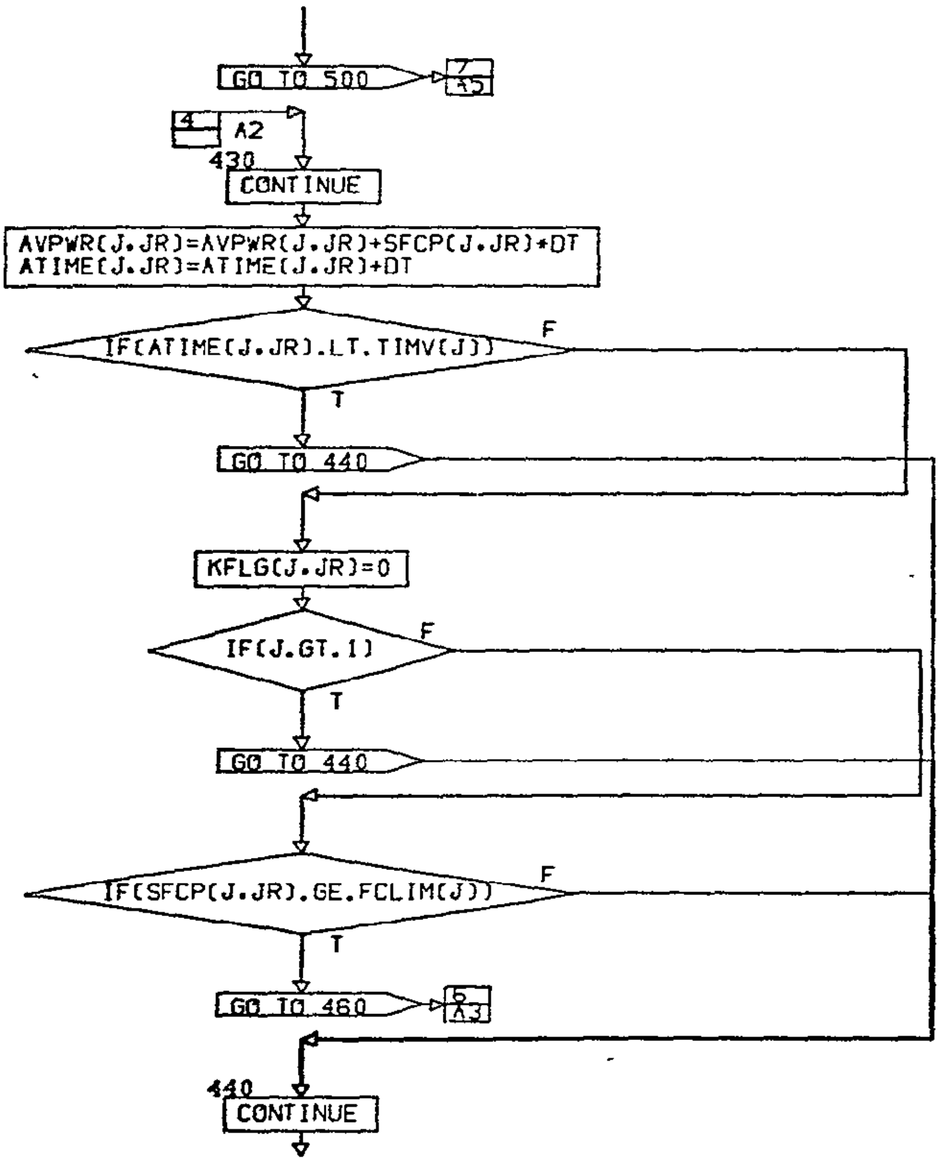


FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)



CONT. ON PG 6

REDLIN
PG 5 OF 9

FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)

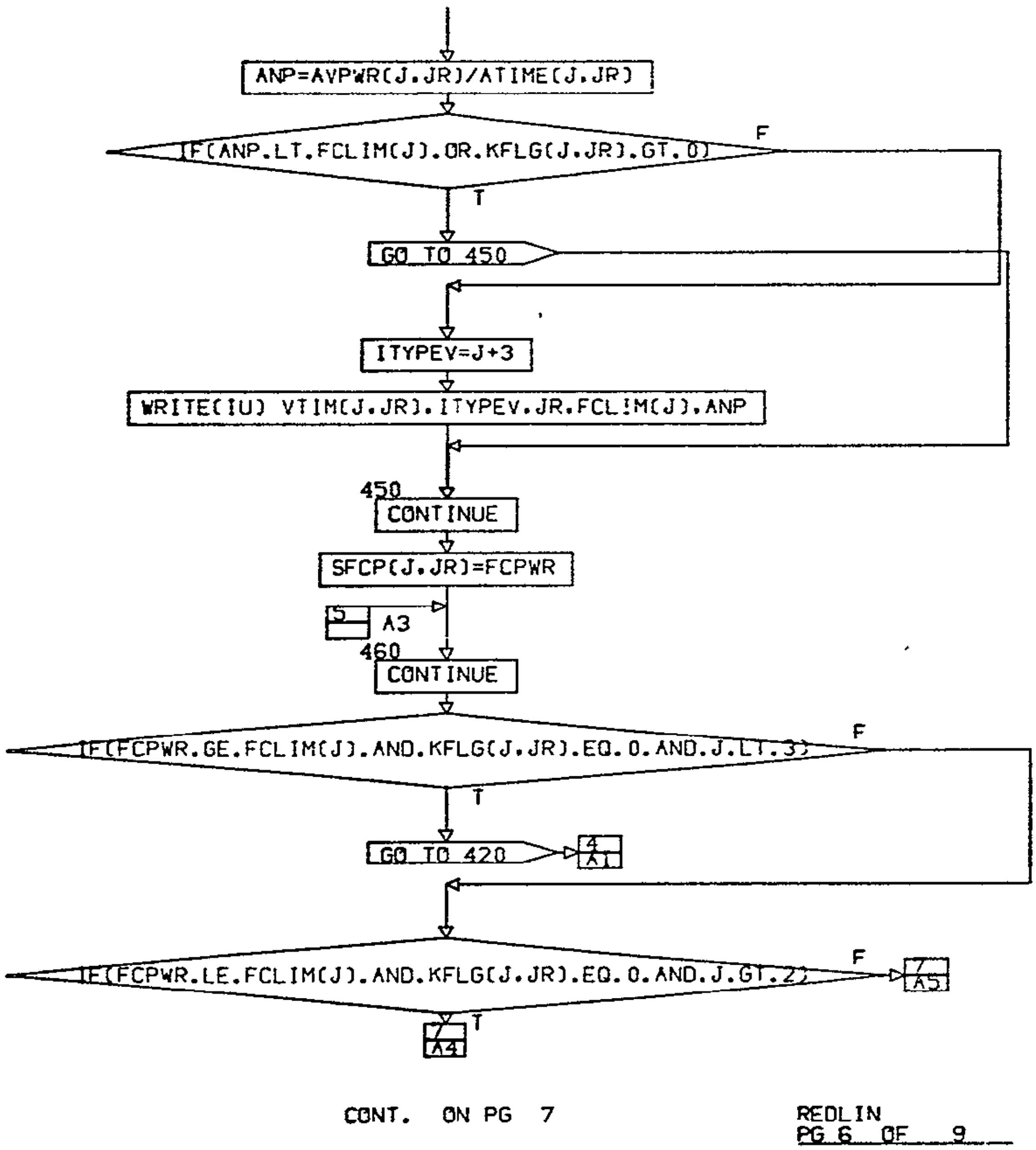
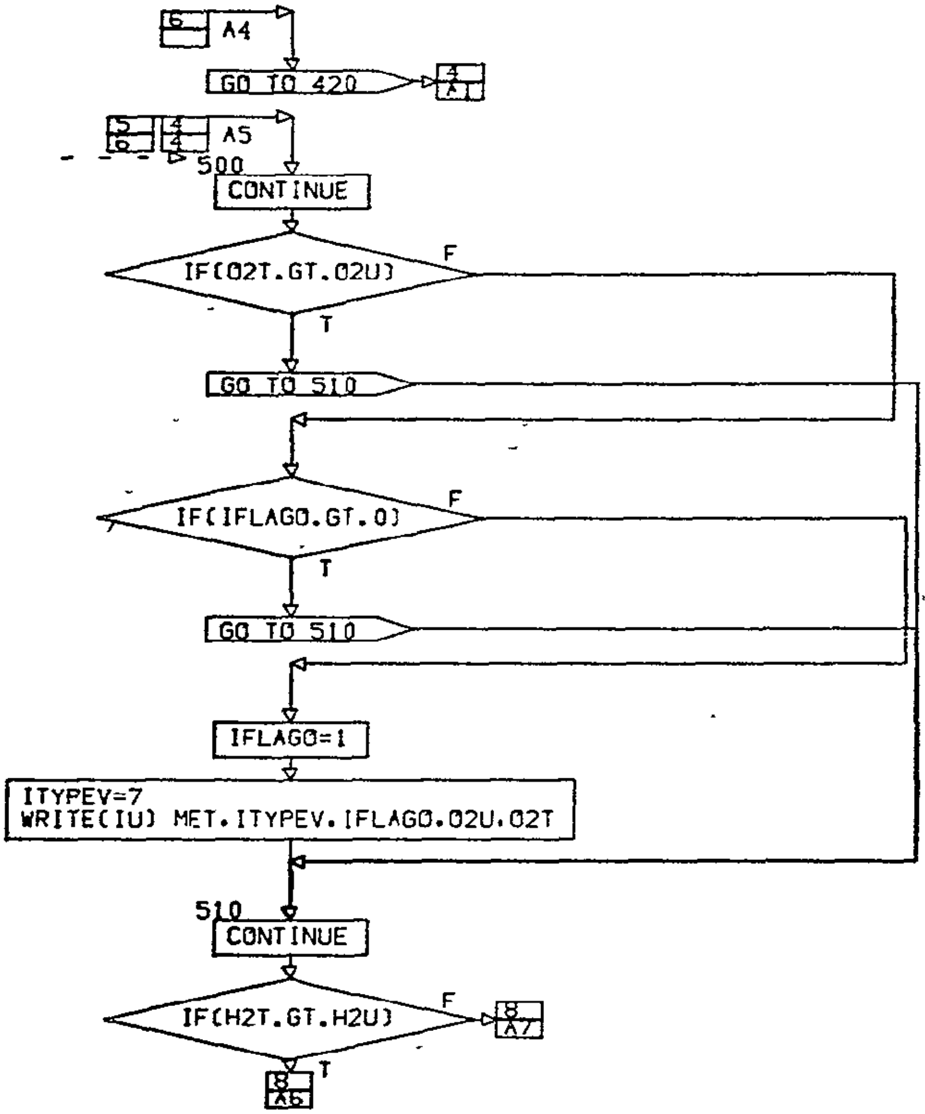


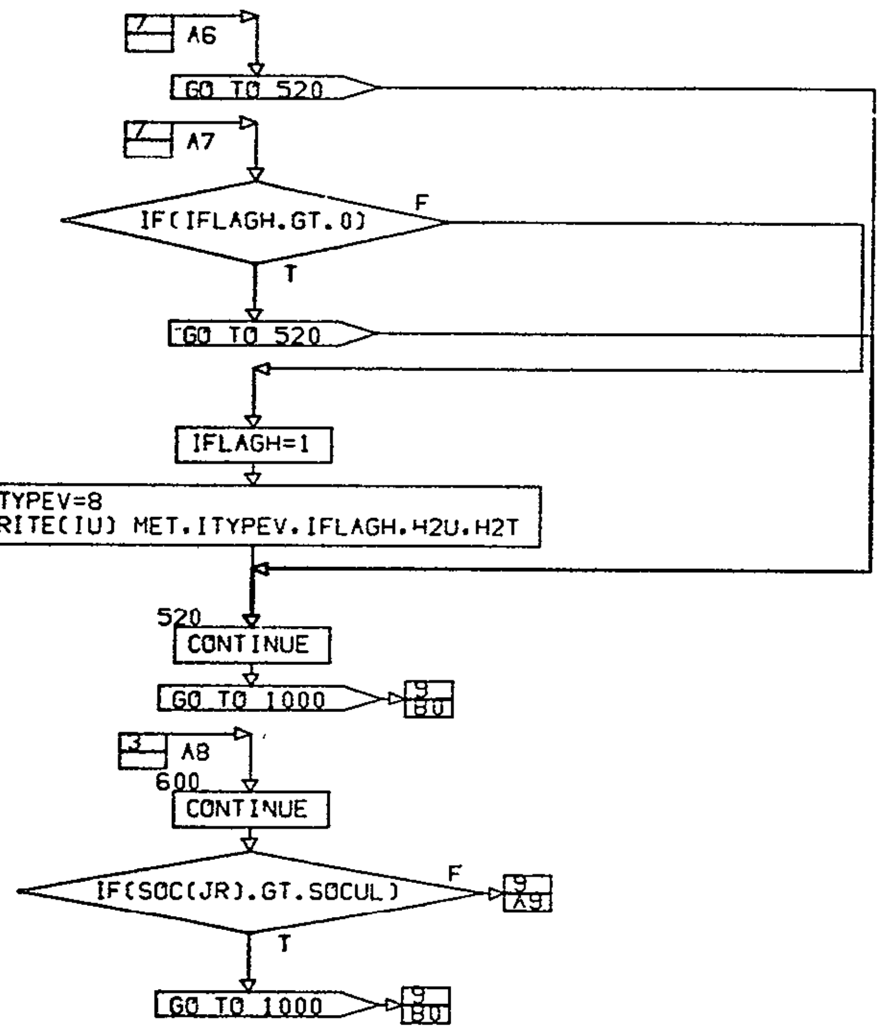
FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)



CONT. ON PG 8

REDLIN
PG 7 OF 9

FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)



CONT. ON PG 9

REDLIN
PG 8 OF 9

FIGURE 3.3.11. FUNCTIONAL FLOW CHART OF SUBROUTINE REDLIN (CONTINUED)

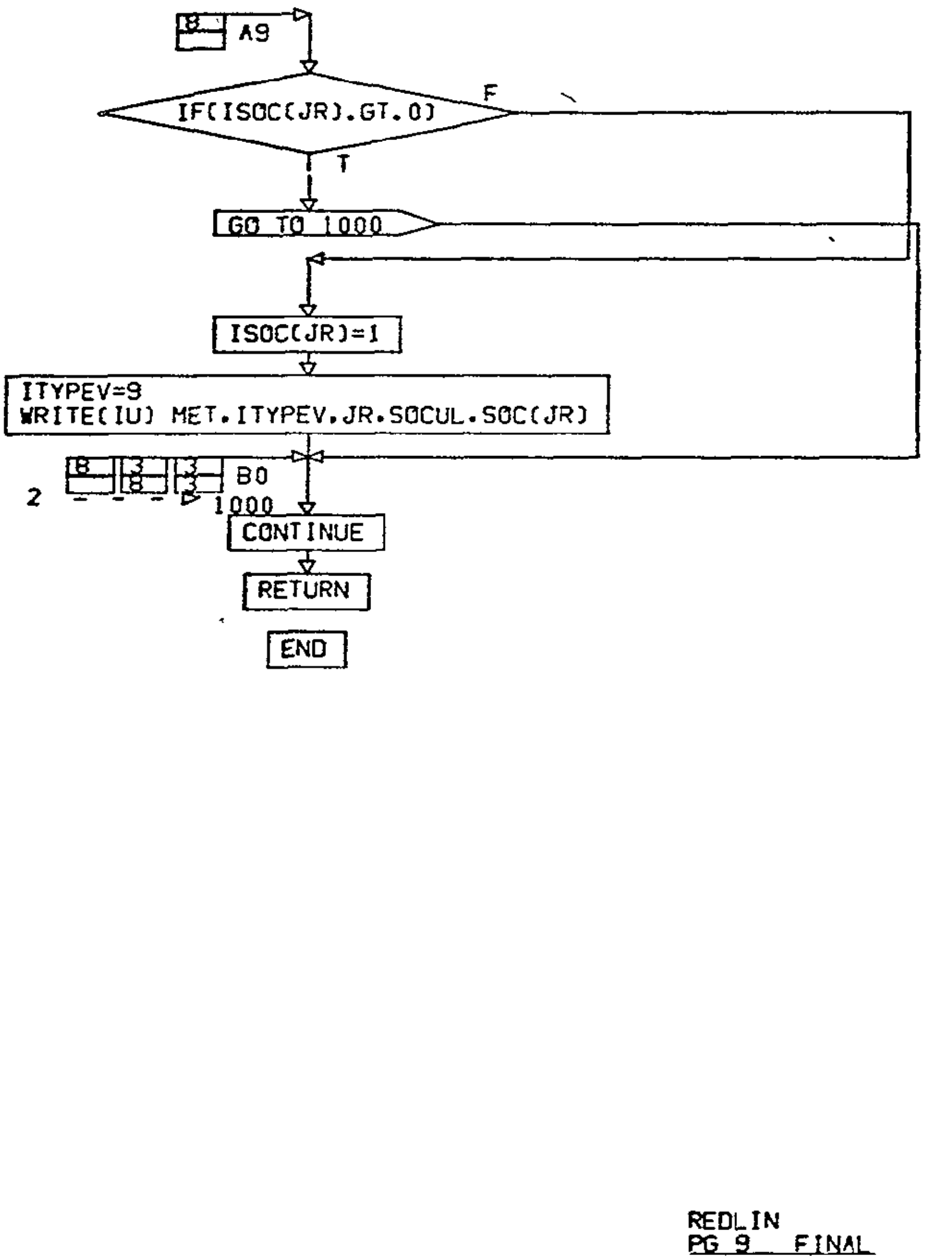


FIGURE 3.3.11. FUNCTIONAL FLOWCHART OF SUBROUTINE REDLIN (CONTINUED)

3.4 ANALYSIS SUBROUTINES

3.4.1 Subroutine: COMUSE

PURPOSE: To provide a component analysis

METHOD: Using the component dictionary, compacted dictionary, and interface tape the following quantities are determined for:

1. Mission Phase:

    Subsystem Analysis
    - Average kilowatts (KW)
    - Percent of total KW
    - Maximum KW
    - Time of maximum

    Phase Analysis
    - Total kilowatt hours (KWH) required
    - Accumulated KWH
    - Average KW for mission phase
    - Maximum KW
    - Time of maximum (KW)

2. Component
    - Average use factor
    - Total on time
    - Component energy required (WH)
    - Percent of total mission energy required

3. Subsystem

    By component
    - Average use factor
    - Total on time
    - Component energy required (WH)
    - Percent of subsystem energy required

VARIABLES: The variables used in this subroutine are listed in the common blocks of the functional flowchart, Figure 3.4.1. See Appendix for definition of all variables.

```
SUBROUTINE COMUSE(IU,IF)
INCLUDE STRAGA
INCLUDE STRAGI
INTEGER QQASG
DIMENSION TLOAD( 750)
DIMENSION PLOAD( 750)
DIMENSION IPLOD( 750)
DIMENSION LNUM ( 750)
```

```
DIMENSION TIMON( 750)
DIMENSION AUF  ( 750)
DIMENSION POWER( 750)
DIMENSION PFAC ( 750)
DIMENSION NAME ( 750.6)
DIMENSION IDUM (100)
DIMENSION SUBT(100)
DIMENSION ICDAT(50)
```

```
DIMENSION NAM(6)
DIMENSION PWRM(4.10)
DIMENSION D1(4)
DIMENSION D2(4)
DIMENSION D3(4)
DIMENSION IALL ( 750.13)
DIMENSION ALL  ( 750.13)
DIMENSION JS(750)
```

```
DIMENSION PS(750)
COMMON /SIX/ ID.ILNCP.NAM.PF.UF.PWRM
COMMON /ACPOWF/ PFEFF.RESLOS
COMMON /ALTERN/ ICDCB(750).ICDCA(500.3).IACT
EQUIVALENCE (ID.ICDAT(1))
EQUIVALENCE (IPLOD(1).PLOAD(1))
EQUIVALENCE (IALL(1.1).ALL(1.1))
DATA NWCD /50/
```

```
DATA TLOAD / 750*0./
DATA TIMON / 750*0./
DATA NAME  /4500*6H      /
DATA SUBT /100*0./
```

CONT. ON PG 2

COMUSE
PG 1 OF 12

FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE

ORIGINAL PAGE IS
OF POOR QUALITY

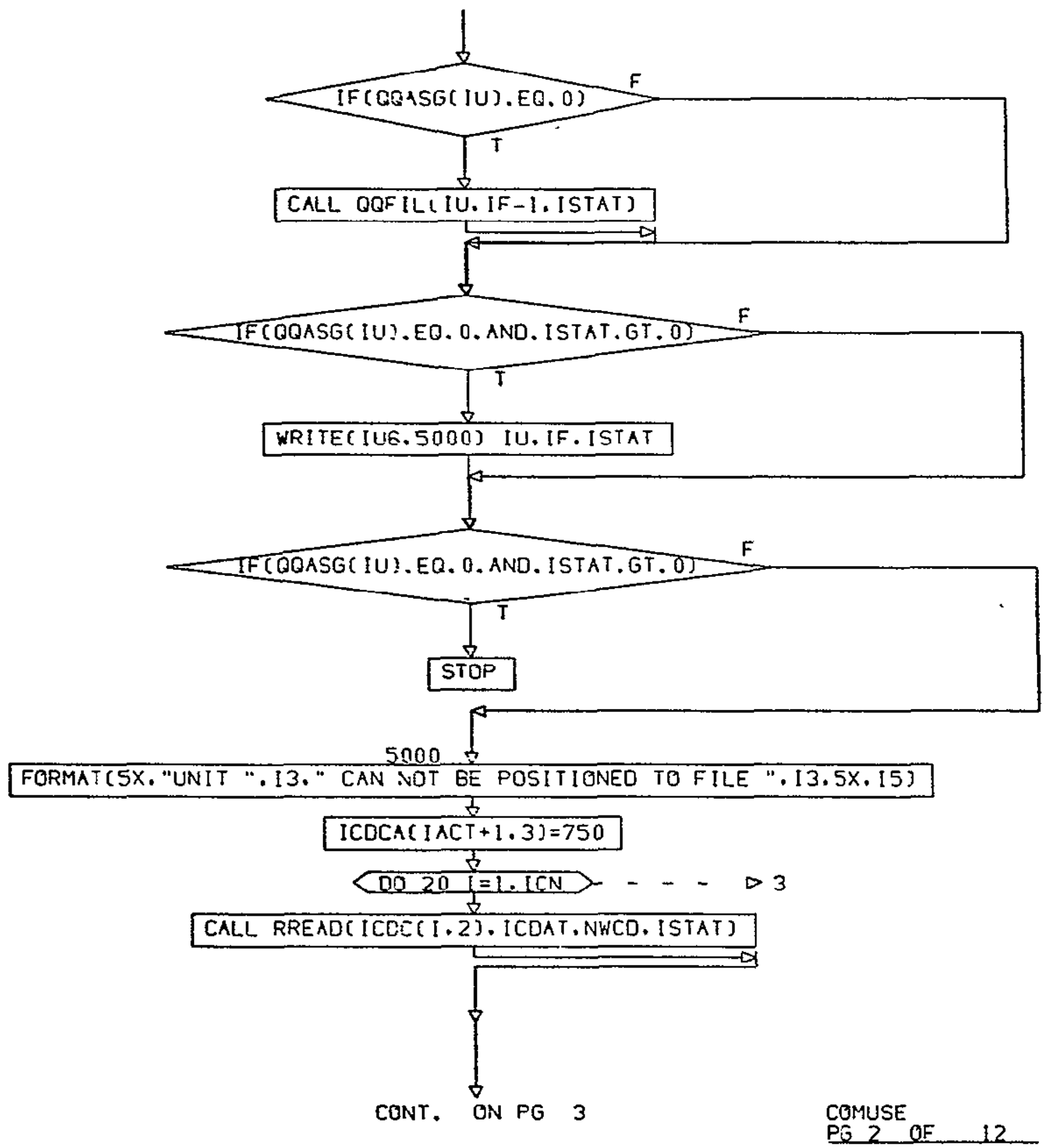


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

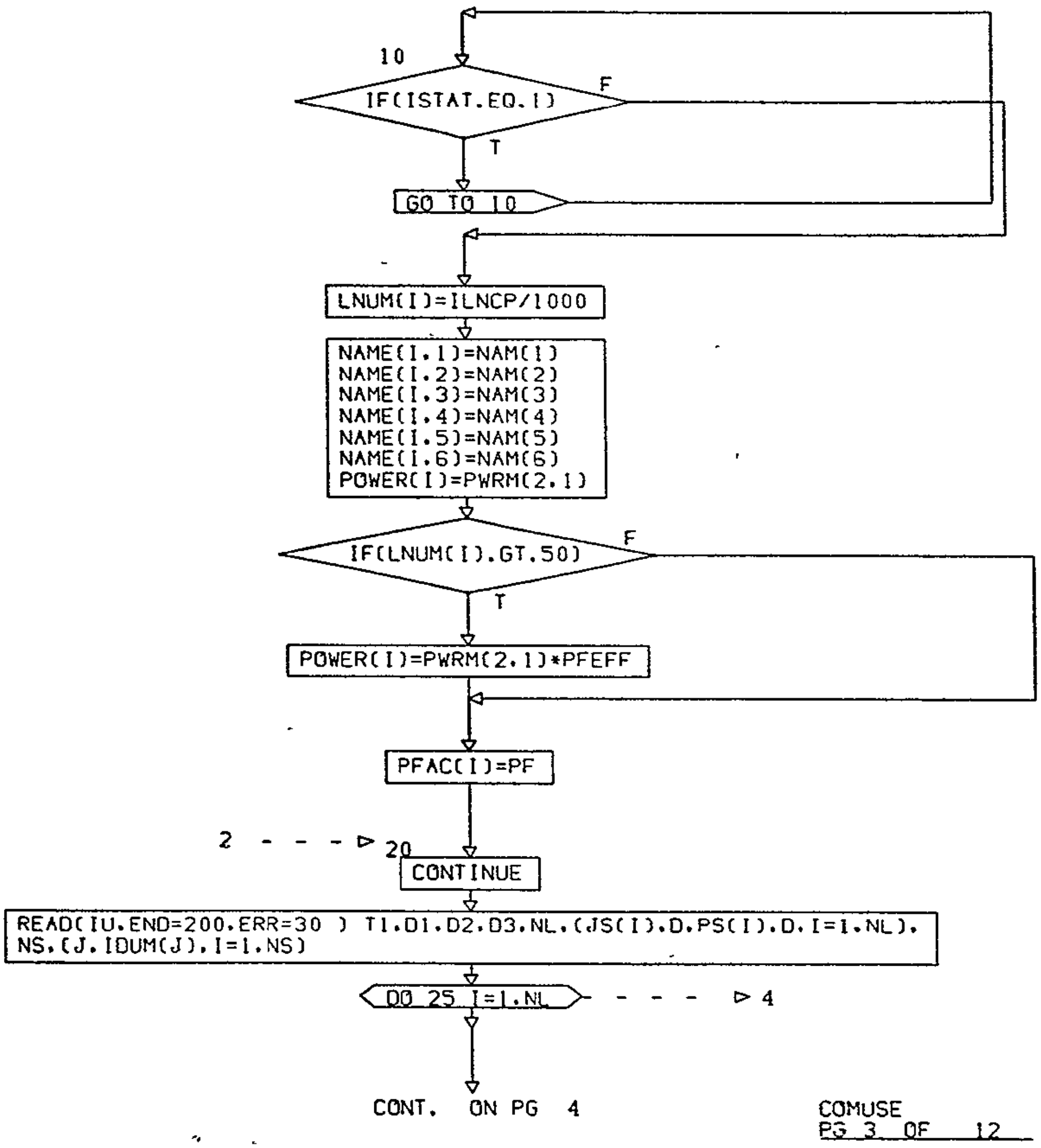
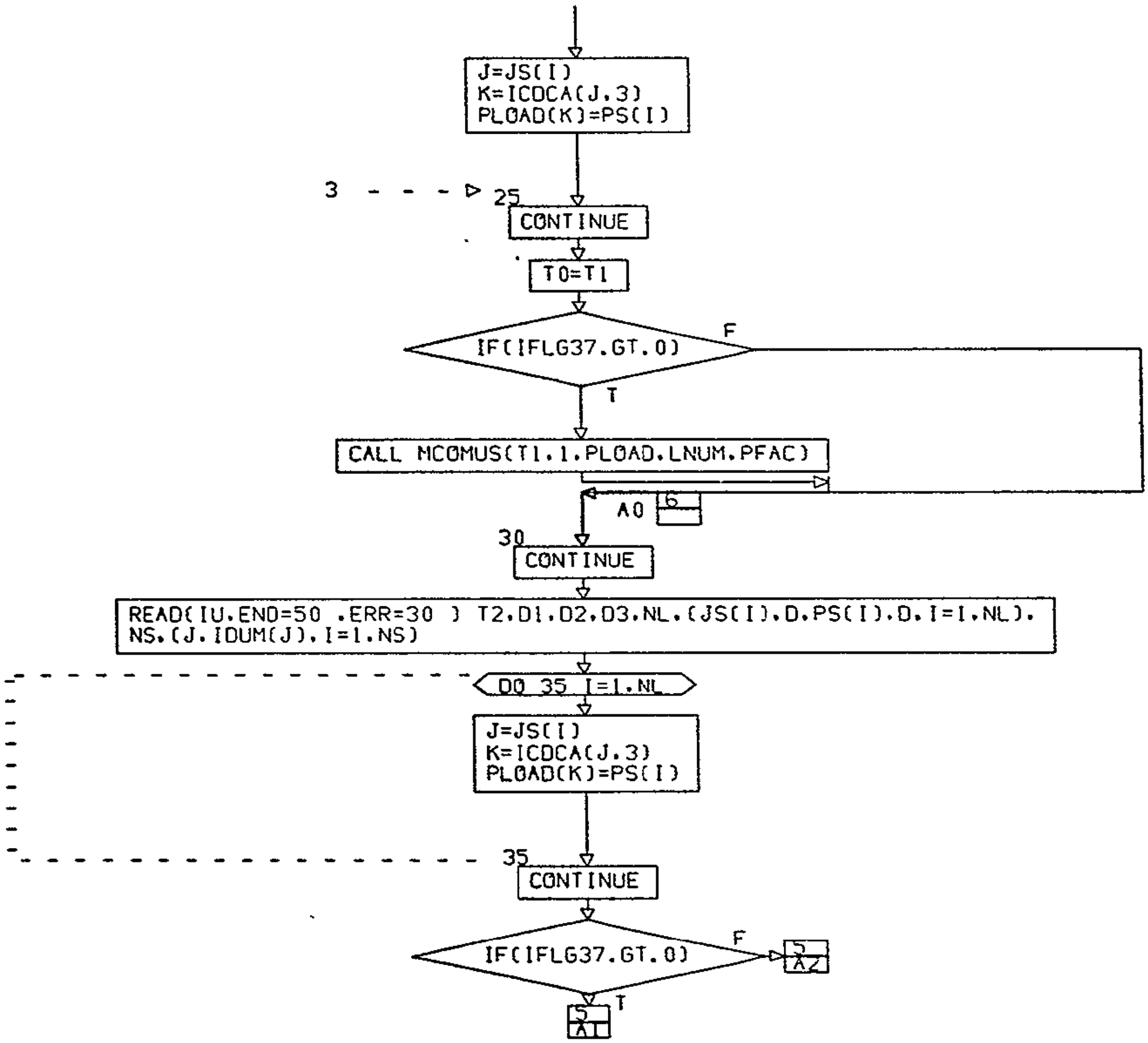


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

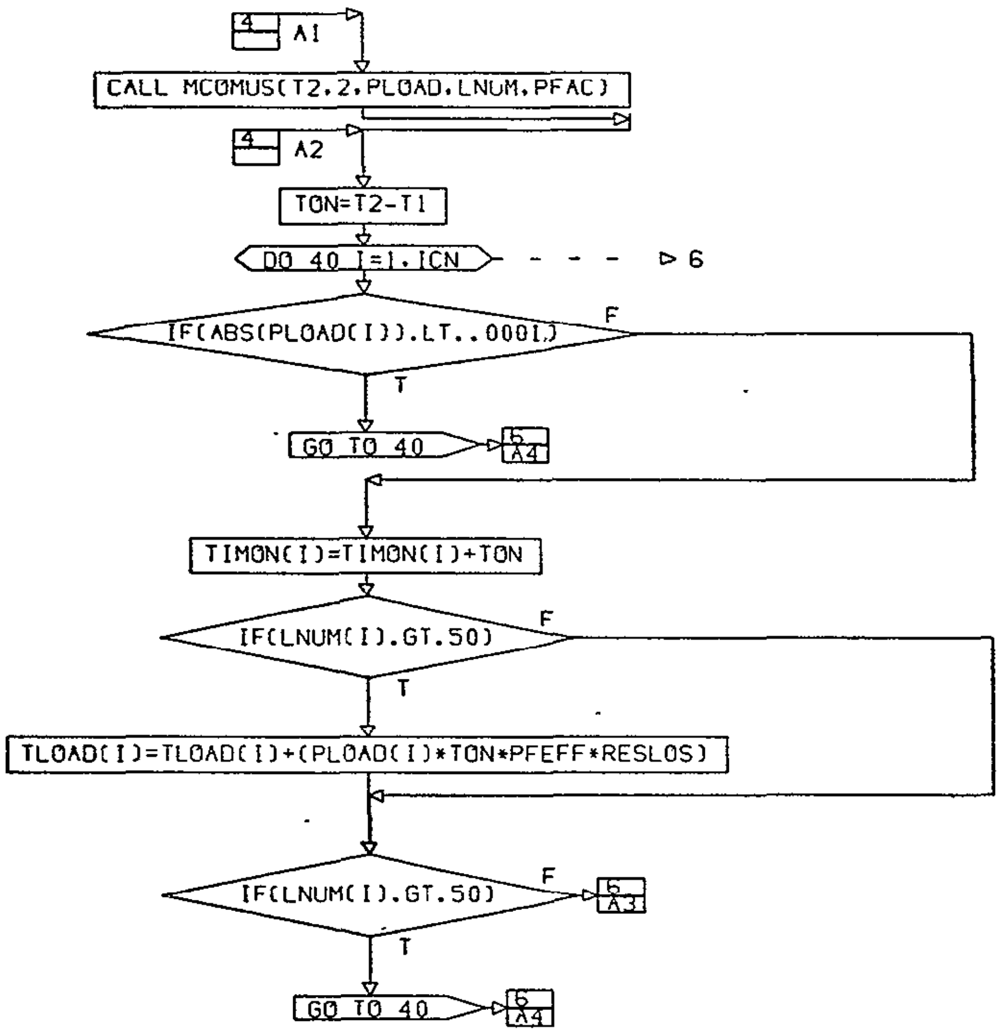
ORIGINAL PAGE IS
OF POOR QUALITY



CONT. ON PG 5

COMUSE
PG 4 OF 12

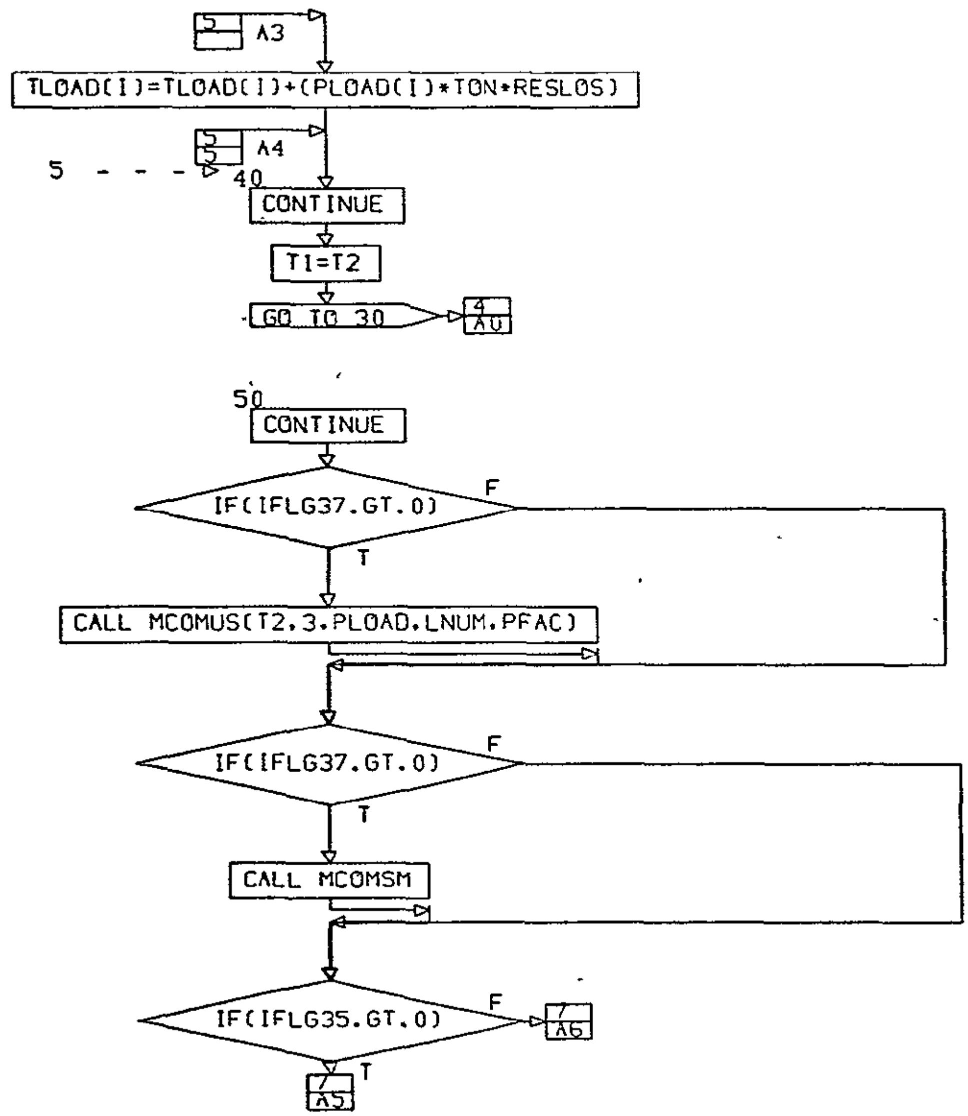
FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)



CONT. ON PG 6

COMUSE
PG 5 OF 12

FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)



CONT. ON PG 7

COMUSE
PG 6 OF 12

FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

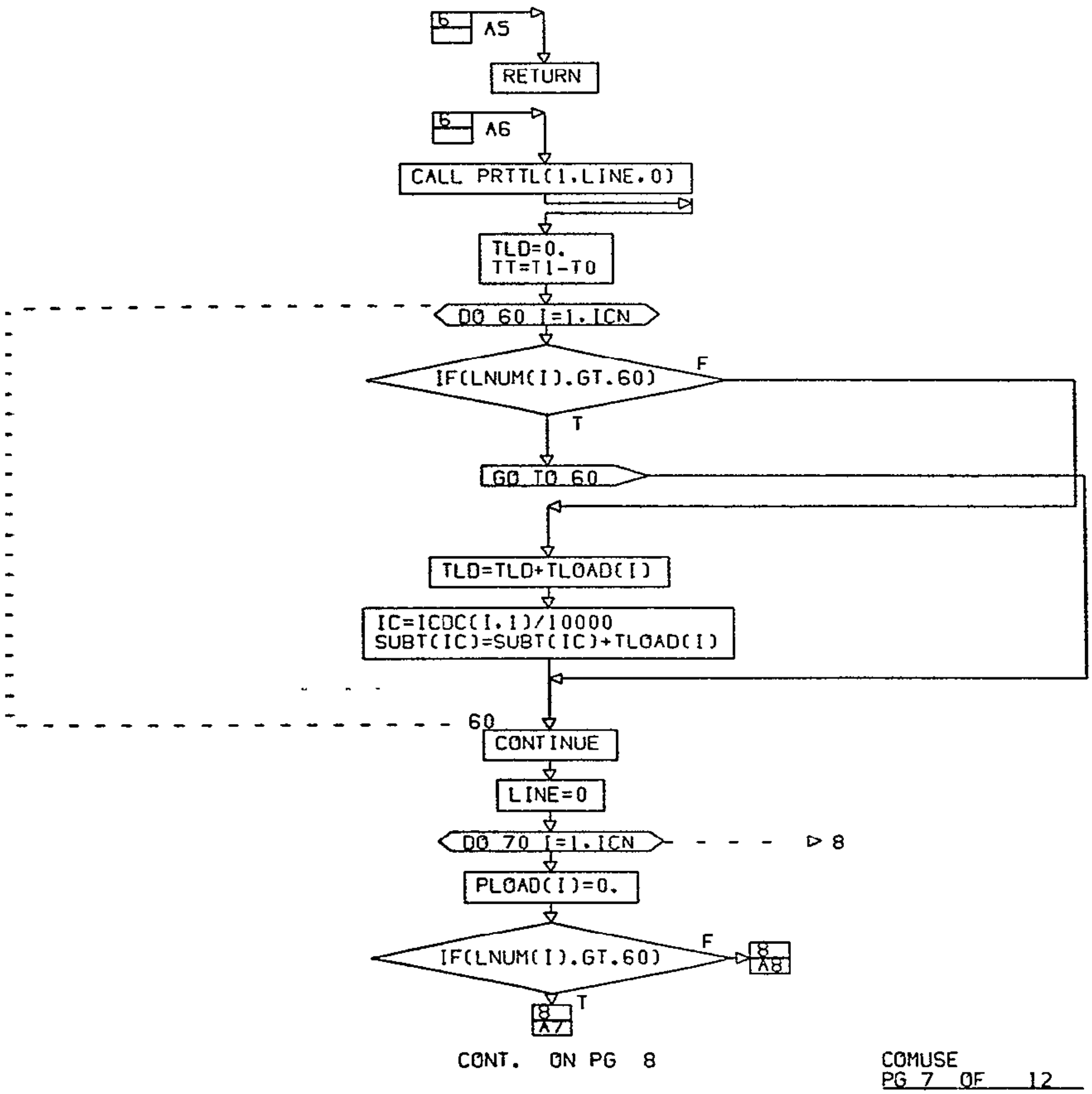
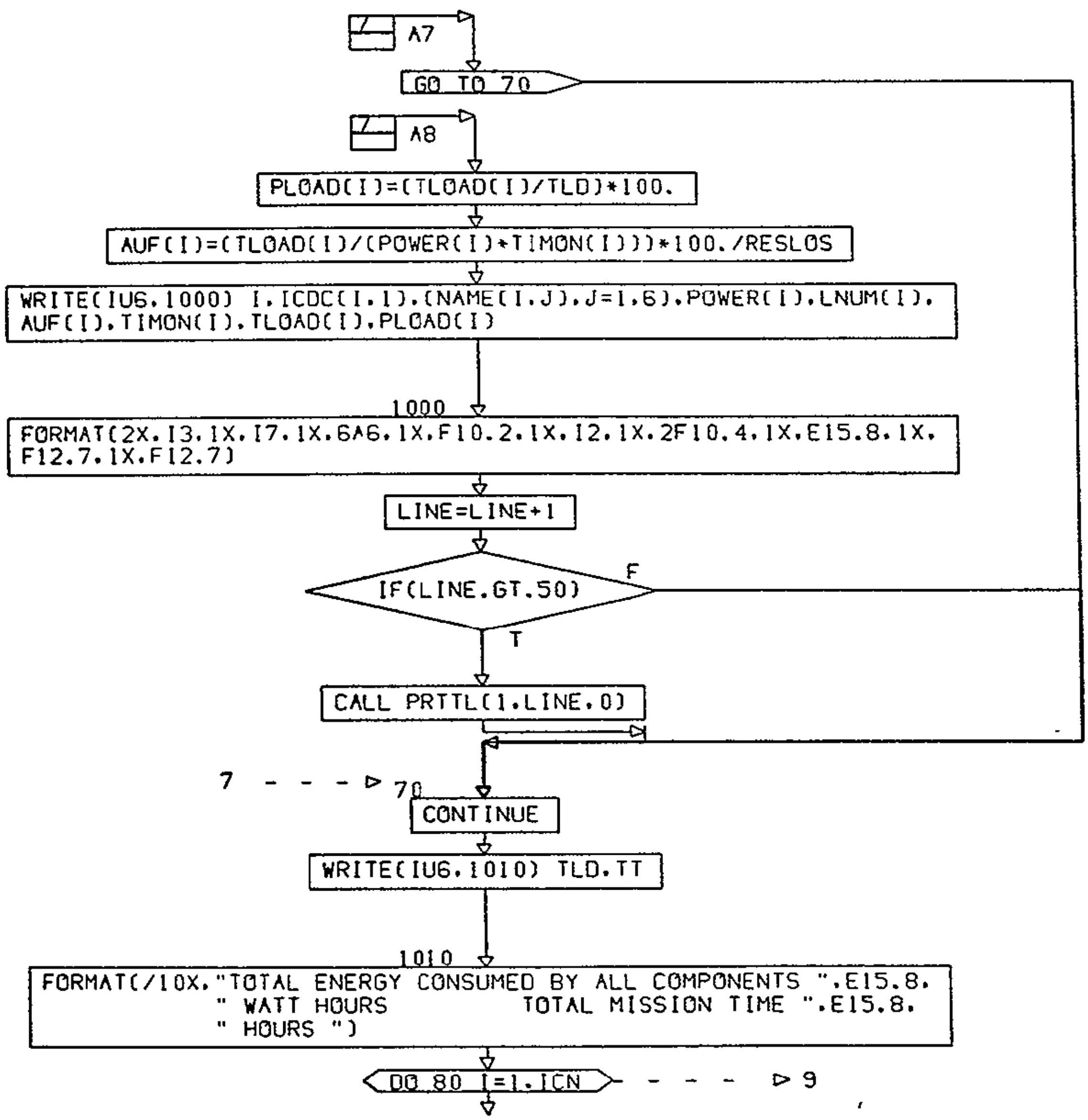


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)



CONT. ON PG 9

COMUSE
PG 8 OF 12

FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

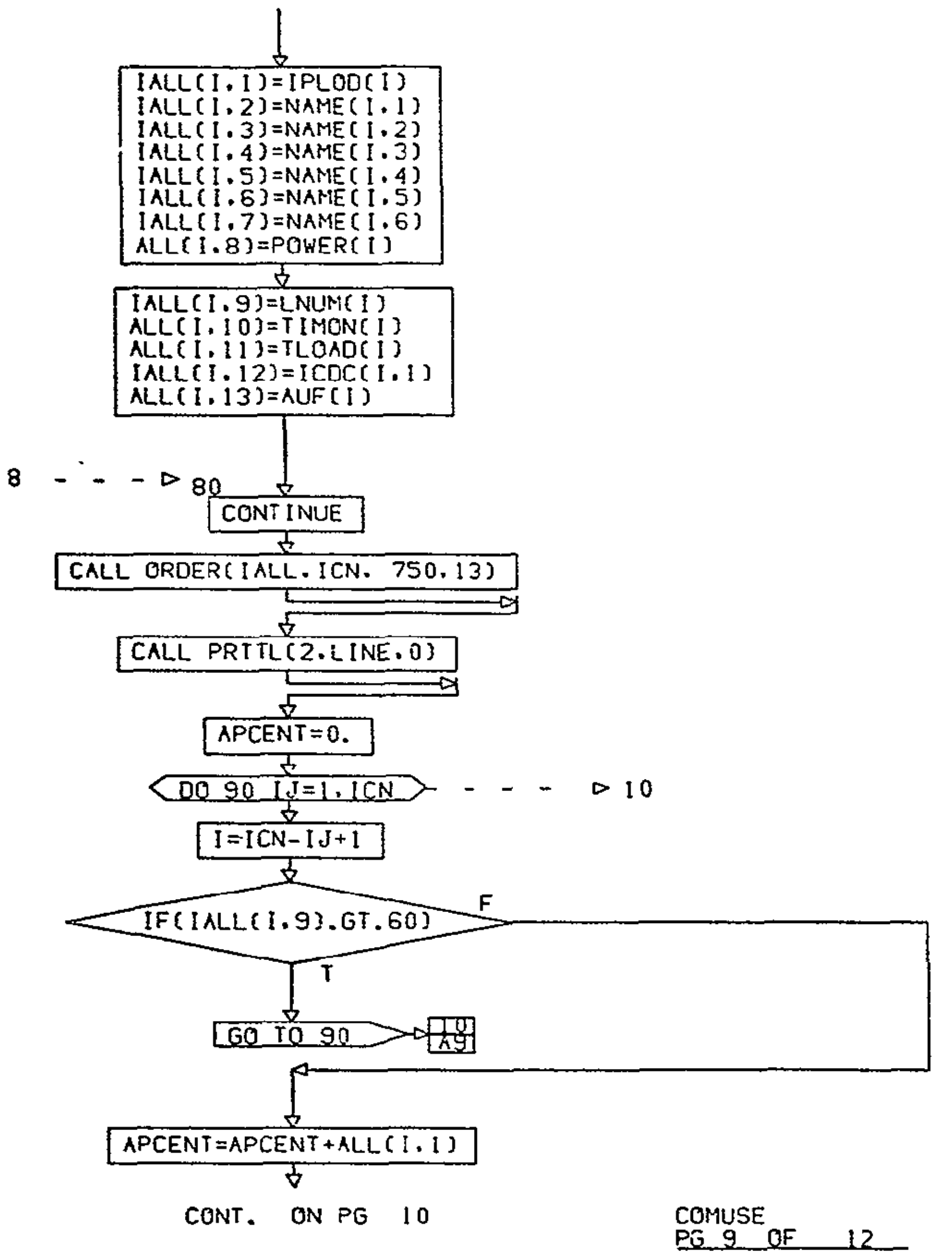


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

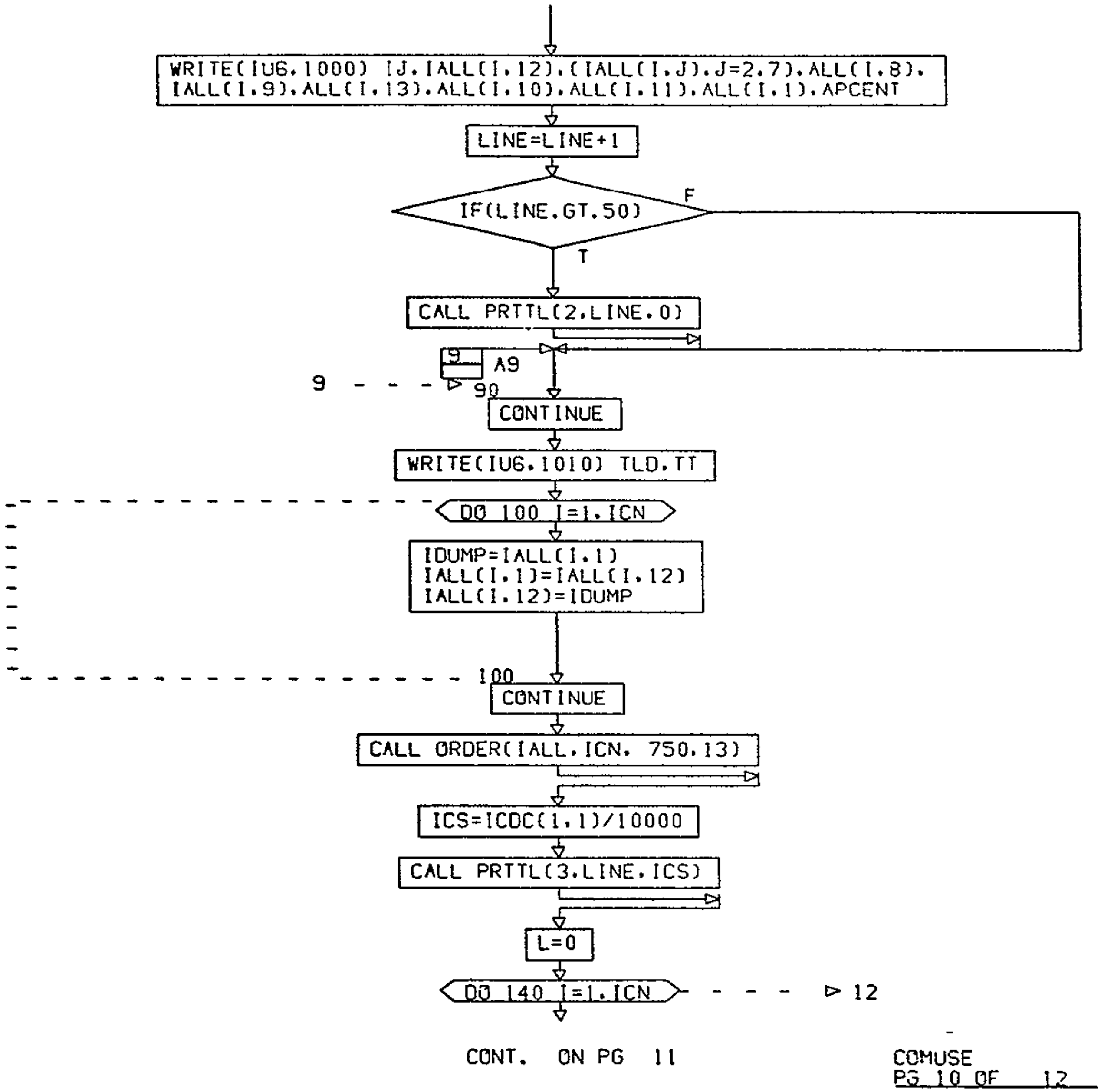
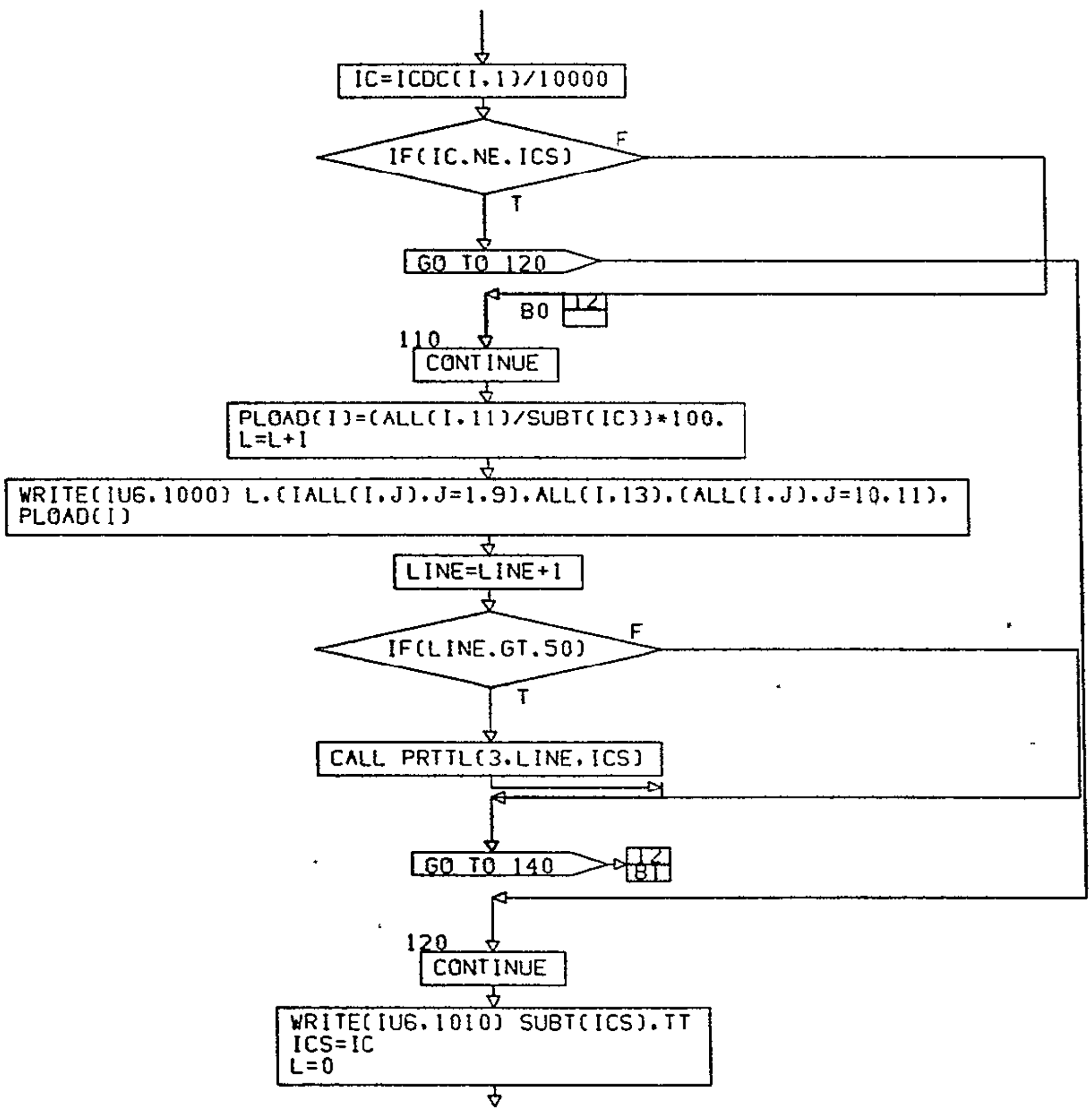


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)



CONT. ON PG 12

COMUSE
PG 11 OF 12

FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

ORIGINAL PAGE IS
OF POOR QUALITY

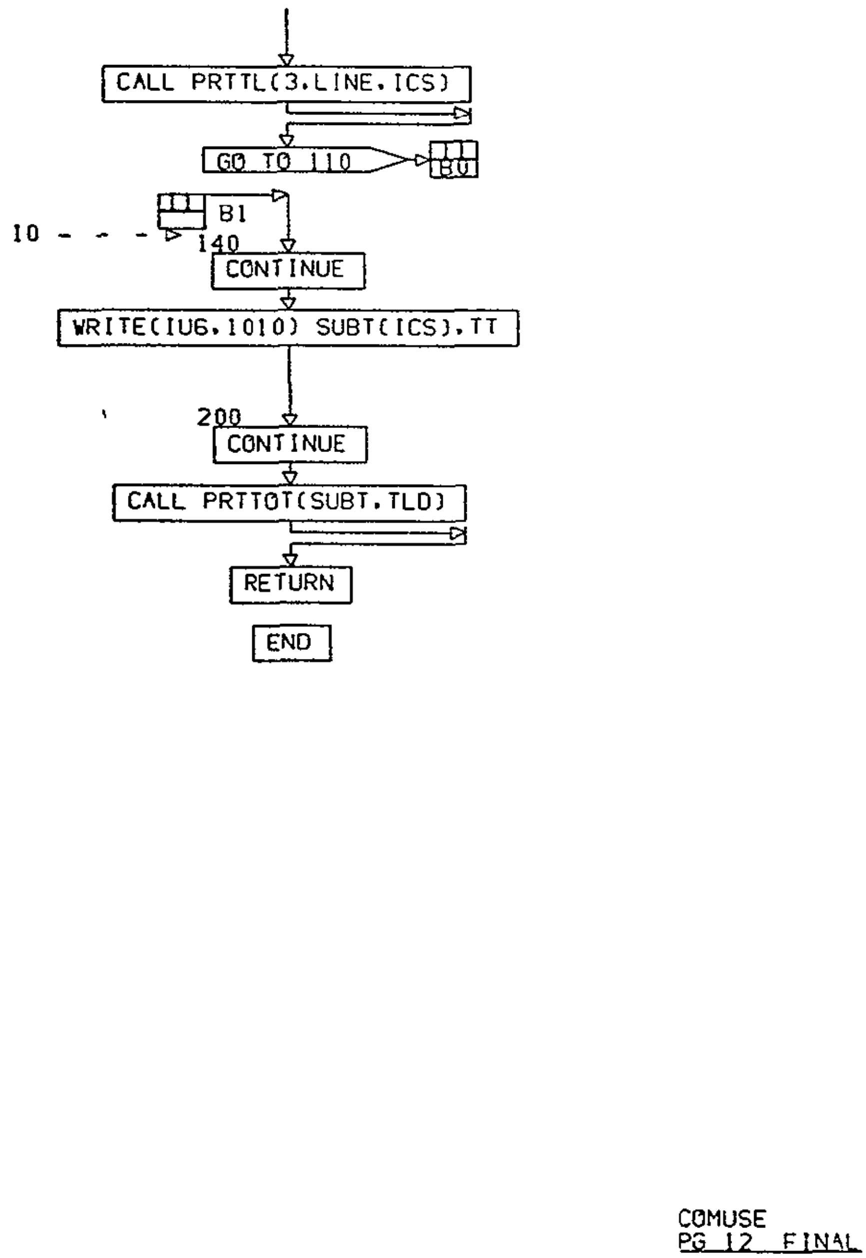


FIGURE 3.4.1. FUNCTIONAL FLOWCHART OF SUBROUTINE COMUSE (CONTINUED)

APPENDIX

APPENDIX

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONTRL	DT				Actual time interval between successive time steps in decimal hours
		IFLG01				Option Flag > 0 - Execute Phase I = 0 or blank - Do not execute Phase I
		IFLG02				> 0 - Execute Phase II = 0 or blank - Do not execute Phase II
		IFLG03				> 0 - Plot on Printer 1 = 0 or blank - Do not plot on Printer 1
		IFLG04				Not Used
		IFLG05				> 0 - Use 3 point load data = 0 or blank - Do not use 3 point load data
		IFLG06				Not Used
		IFLG07				Not Used
		IFLG08				Not Used
		IFLG09				Not Used
		IFLG10				> 0 - Print each Phase II timepoint = 0 or blank - Do not print each Phase II timepoint
		IFLG11				Not Used
		IFLG12				Not Used
		IFLG13				Not Used
		IFLG14				Not Used
		IFLG15				Not Used
		IFLG16				Not Used
		IFLG17				Not Used

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONTRL	IFLG18				Not Used
		IFLG19				Not Used
		IFLG20				> 0 - Execute Phase II COMUSE = 0 or blank - Do not execute COMUSE
		IFLG21				Not Used
		IFLG22				Not Used
		IFLG23				Not Used
		IFLG24				Not Used
		IFLG25				Not Used
		IFLG26				Not Used
		IFLG27				Not Used
		IFLG28				Not Used
		IFLG29				> 0 - Suppress analysis part 1 = 0 or blank - Do not suppress analysis part 1
		IFLG30				> 0 - Execute Phase I COMUSE = 0 or blank - Do not execute Phase I COMUSE
		IFLG31				> 0 - Print out input components = 0 or blank - No action taken
		IFLG32				> 0 - Print out input procedures = 0 or blank - No action taken
		IFLG33				> 0 - Print out input activities = 0 or blank - No action taken
		IFLG34				> 0 - Print out input timeline = 0 or blank - No action taken
		IFLG35				> 0 - Suppress COMUSE component analysis = 0 or blank - Do not suppress COMUSE component analysis

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONTRL	IFLG36				> 0 - At the end of each mission phase provide a listing of all active components = 0 or blank - No action taken
		IFLG37				> 0 - Mission phase analysis = 0 or blank - No mission phase analysis
		IFLG38				> 0 - Suppress cycled component listing = 0 or blank - Do not suppress cycled component listing
		IFLG39				> 0 - Suppress print of compacted dictionary = 0 or blank - Do not suppress compacted dictionary printout
		IFLG40				> 0 - Suppress subsystem analysis at each time point = 0 or blank - Do not suppress subsystem analysis
		IABORT				Set > 0 to abort simulation
		IFILE	20			
						I=1 File containing components
						I=2 File containing procedure
						I=3 File containing activities
						I=4 File containing timelines
						I=5 File containing fixed data
						I=6 Not Used
						I=7 Not Used
						I=8 File containing Phase I interfaces
						I=9 File containing Phase I plot
						I=10 File containing Phase II interface

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONTRL	IFILE				I=11 File containing Phase II plot
						I=12 File containing compacted dictionary
						I=13 File containing circuit definition
						I=14 File containing constraint violations
						I=15 File containing subsystem names
						I=16 File containing mission phase
						I=17 Not Used
						I=18 Not Used
						I=19 Not Used
						I=20 Not Used
		ISOLVC				Flag to request a circuit solution at a particular card timeline point. Set = 1 to request solution
		IUNIT	20			
						I=1 Unit containing components
						I=2 Unit containing procedures
						I=3 Unit containing activities
						I=4 Unit containing timeline
						I=5 Unit containing fixed data
						I=6 Not Used
						I=7 Not Used
						I=8 Unit containing Phase I interface
						I=9 Unit containing Phase I plot

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONTRL	IUNIT				I=10 Unit containing Phase II interface
						I=11 Unit containing Phase II plot
						I=12 Unit containing compacted dictionary
						I=13 Unit containing circuit definition
						I=14 Unit containing constraint violations
						I=15 Unit containing subsystem names
						I=16 Unit containing mission phases
						I=17 Not Used
						I=18 Not Used
						I=19 Not Used
						I=20 Not Used
		JPRINT				Flag to request initialization data to be printed out. Value >1
		NPRT				Formatted printout interval as a multiple of TDELTA
		MET				Simulation start time
		TDELTA				Maximum simulator time increment
		TREADC				Next time to read card input
		TREADT				Next time to read tape input
	DEBUG	MPRNT				Debug print control
		MPRNT1				Debug print control
		MPRNT2				Debug print control

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	DEBUG	MPRNT3				Not Used
		MPRNT4				Debug print control
		MPRNT5				Not Used
		MPRNT6				Not Used
	CMPCNT	CVAL	25			Component power value to be used in Phase II calculations in lieu of the component loads in TPLOAD. NOTE: LOC 1-5 for F/C 1-5 noncyclic loads and LOC 6-10 for F/C 1-5 cyclic loads
	CMPCNT	NCNT	25			Component numbers associated with the component loads in CVAL
		NCNTC				Number of loads defined in CVAL
		NCTP	25			For each LOAD in CVAL, defines if the load is constant power (>0) or constant resistance (=0)
	UNITS	IU5				Internal unit designation
		IU6				Internal unit designation
		IU7				Internal unit designation
		IU8				Internal unit designation
		IU9				Internal unit designation
		IU10				Internal unit designation
		IU11				Internal unit designation
	CONSTR	ACVA	9			AC volt-ampere load on inverter
		CAPINV	9			Inverter (I) maximum overload limit
		FCLIM	3			Fuel cell power limits, 1 = peak, 2 = average, 3 = minimum
		H2U				Unusable hydrogen quantity
		O2U				Unusable oxygen quantity

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG1	CONSTR	SOCUL				Minimum battery SOC limit
		TIMV	3			Length of time that fuel cell power limits apply, 1 = peak, 2 = average, 3 = minimum
STRAG2	DCCRCT	CB	100			Branch current
		CL	50			Load current operating point
		CS	12			Source current operating point
		DELTA				Tolerance on node voltage solutions (normally 10^{-5})
		ES	12			Source voltage operating point
		ITER				Circuit solution internal iteration counter
		NITER				Maximum allowable iterations in the circuit polution
		NOR				Circuit reference node
		NSC				Number of I-V points in Source (I) used in the circuit solution
		NT3S				Number of points used in T3SRCS
		NT4S				Number of points used in T4SRCS
		PP				Constant power load for LOAD (I), variable not used when Phase II is driven by an interface tape
		PR				Constant resistive power load for LOAD (I), PP(I) and/or PR(I) may be used to represent LOAD (I), variable not used when Phase II is driven by an interface load
		R				Branch (I) line resistance
		RLOAD	3	51		Three point equivalent load I = load at 24v, 28v, 32v J = 50 possible loads J = 51 voltage equivalent

C.4

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG2	DCCRCT	S	100			Branch (I) switch position (1 = closed, 0 = open)
		SC	15	2	12	Source curves I= Number of points in curve J=1 Voltage J=2 Current K= Source number
		T3SRCS	15	2		Third source type I= Number of points in curve J=1 Voltage J=2 Current
		T4SRCS	15	2		Fourth source type I= Number of points in curve J=1 Voltage J=2 Current
		V	30			Node voltage
		VL	50			Load voltage operating point
		Z	100			Admittance of the branch
	CRTFLG	IACSOL				AC circuit solution flag
		IDCSOL				DC circuit solution flag
	INVERT	ACPOW	9			AC load for inverter AC BUS (I) variable not used when Phase II is driven by an interface tape
		INVOL	10			Inverter overload flag
		PFAC	9			AC load power factor for AC BUS (I) corresponding to loads in ACPOW(I). Not used with interface tape

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG2	DEFCRT	CRCT	100	6		Branch (I) diode or RPC definition
						I= Branch No.
						J=1 Diode voltage drop
						J=2 Diode forward resistance
						J=3 Reverse resistance
						J=4 RPC no load resistance
						J=5 Forward or reverse resistance being used
						J=6 Branch current limit
		ICHRBP				Load location of the battery charger
		ICRCT	100	4		Branch (I) definition
						I= Branch No.
						J=1 Node number current out
						J=2 Node number current in
						J=3 Source number
						J=4 Load number
		INVLCA	9			Inverter number connected to ac inverter bus (I)
		INVLCD	10	2		Table of branch no. vs inverter in the branch
						I= No. of entries in the table
						J=1 Branch No. containing dc load
						J=2 Associated inverter no. for above dc load
		LOADS	50			Branch locations of the loads
		NCRT				Maximum or highest branch number in ICRCT

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG2	DEFCRT	NINVLA				No. of inverters
		NINVLA				No. of ac load buses supplied by inverters
		NINVLD				No. of dc loads used for inverters
		NLDS				Highest load number
		NNODE				Highest node number used in circuit
		NSCS				Number of sources
		NSOUR	12	3		Source definition table
						I= Source number
						J=1 Branch location
						J=2 Source type
						J=3 Source reference numbers
		UV	30			Undervoltage limit for node (I)
STRAG3						Not Used
STRAG4	BATTRY	AH	6			Actual number of ampere hours remaining
		BC	6			Operating current point of the battery
		BV	6			Operating voltage point of the battery
		CC	6			Ampere hour capacity of the battery
		CHRGLD				DC power required by battery charger when charging batteries
		CSUBD	6			Battery amp-hour capacity for battery (I)
		DQ	6			Amount of heat generated by the battery in watts
		EFF	6			Battery decimal efficiency during charge and discharge

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG4	BATTRY	EFFAVR				Average amp-hour efficiency during charging
		ICHRG	6			Charge flag to initiate charging of battery (I)
		IT	5	2		Battery temperatures used in SOCA, I= Temperature J= Battery group
		NSOCA				Number of points used in SOCA for each I-V curve
		P1				Constant used in the battery heat generation equation
		P2				Constant used in the battery heat generation equation
		P3				Constant used in the battery heat generation equation
		P4				Constant used in the battery heat generation equation
		SOC	6			Battery (I) initial state-of-charge
		SOCA	7	6	2	Battery I-V curves versus temperature, and battery group I= Number of points J=1 Current J=2-6 Voltage at temperature of IT K= Battery group 1 or 2 (1 = group of 3 common batteries) (2 = group of common batteries) (per cell voltage)
		TB	6			Battery (I) temperature
		TD	6			Time of last major battery discharge
		TSS				Battery steady state temperature

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG4	BATTRY	XNC	6			Number of cells in battery I
STRAG5	FUSEL	DAT	5			Time the fuel cell has been operating in decimal hours (MET)
		DTM				Internal time step in FUCLTM
		FCCP	5			Fuel cell current
		FCHOL				Fuel cell high temperature limit - heater turns "OFF"
		FCHTL				Fuel cell redline limit - diagnostic warning
		FCLTL				Fuel cell lower temperature limit - heaters turn "ON"
		FCT	5			Operating temperature of fuel cell (I)
		FCTA	9	7		Array containing fuel cell I-V curves as a function of temperature I= Number of points J=1 Current value (amps) J=2 thru J=7 Voltage level at each temperature of FCTN
		FCTN	6			Temperatures associated with the I-V curves of FCTA
		FCWP1	5			Parasitic pump and logic loads for fuel cell (I) - constant power
		FCWP2	5			Parasitic heater cyclic load for fuel cell (I) constant resistance
		HPT				Hydrogen purge time
		HR				Hydrogen purge rate
		HUR				Hydrogen use rate based on amp-hour requirements
		H2I				Initial quantity of H_2 loaded in lbs

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG5	FUSEL	H2OT				Total quantity of H_2O accumulated in lbs
		HDUM1				Dummy filler variable
		HDUM2				Dummy filler variable
		HDUM3				Dummy filler variable
		H2T				Initial quantity of hydrogen available
		KHTR	5			Fuel cell parasitic load flag Value 0 implies no parasitic load Value 1 implies parasitic load
		NFCTA				Number of current points used in FCTA
		OPER	5			Flag indicating the on/off condition of the fuel cell
		OPT				Oxygen purge time
		OR				Oxygen purge rate
		OUR				Oxygen use rate based on amp-hour requirements
		O2I				Initial quantity of O_2 loaded in lbs
		O2T				Initial quantity of oxygen available
		PIH				Interval between hydrogen purges
		PIO				Interval between oxygen purges
		RES	5			Equivalent resistance of the fuel cell parasitic load at 28 VDC
		SSTVI	10	2		The T-I curve which the fuel cell follows as the fuel cell temperature reaches its steady state value J=1 Temperature J=2 Current
		TMAXFC				Maximum time step through feul cell thermal model

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
STRAG5	FUSEL	TPH	5			Time of last hydrogen purge for fuel cell (I)
		TPO	5			Time of last oxygen purge for fuel cell (I)
		WPR				Fuel cell water production rate
STRAG6						Not Used
STRAG7						Not Used
STRAGA	PHAS1	IADC	250	2		Activity dictionary I= Dictionary element J=1 Activity number J=2 Drum address
		IAN				Number of dictionary entries
		ICDC	750	2		Component dictionary I= Dictionary element J=1 Component number J=2 Drum address
		ICN				Number of dictionary entries
		IPDC	750			Procedure dictionary I= Dictionary element J=1 Procedure number J=2 Drum address
		IPN				Number of dictionary entries
		MM				Number of mission phases
		TABORT				Mission elapsed time to end the simulation, default is 500 hours

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
N/A	ACPOWF	PFEFF				Inverter efficiency
		RESLOS				Total system line loss
N/A	ALTERN	IACT				Number of entries in the compact dictionary
		ICDCA	500	3		Compact component dictionary
						I= Number of entries
						J=1 Component number
						J=2 Corresponding drum address
						J=3 Relative location in the component dictionary
		ICDCB	750			Component usage count
N/A	BSLOC	IBUSC	500			Component load assignment (COMPACT)
		ISYSC	500			Component system assignment (COMPACT)
N/A	BSLOCA	IBUSC	750			Component load assignment
		ISYSC	750			Component systems assignment
N/A	CYCLIC	I				Number of entries
		MS	100			Cyclic mode
		NS	100			Cyclic number
		PERS	100			Cyclic period
		PONS	100			Cyclic percent on
		TS	100			Cyclic type
		TTS	100			Cyclic time to start cycle

SEPS DATA VARIABLES

PDP ELEMENT	COMMON BLOCK	VARIABLE NAME	I	J	K	DEFINITION
N/A	DRMFLG	ISF				Override control on illegal component turn off
N/A	FCYCL	KF				Type of entry indicator
N/A	INVEFF	ACEFF	3			Inverter efficiencies
N/A	INVPF	ACPFAC	9			AC load power factors
N/A	MPSF	ISF29	50			Same as IFLG29
		ISF36	50			Same as IFLG36
		ISF38	50			Same as IFLG38
		ISF40	50			Same as IFLG40
N/A	MTRANT	TIMEM	50			Mission elapsed time at end of mission phase
N/A	PRINT	ICARD				Card input read
		IPRNT				Input print required
		IRESET				Simulation reset point
		ITAPE				Tape input read
N/A	SUBSTM	ILOC	25			Subsystem number
		ILOCN				Number of subsystems
		TITLE	6	25		Subsystem name
N/A	TLINF	IDA				Present drum address
		IDRM				Drum full flag
		IEND				End of Phase I flag
		IFIL				Output file number
		IOUT				Output record number
		IOUTM				Maximum number of output records
		NWL				Number of drum words left
N/A	TOTPWR	PWRTOT				Total source power