NASA-RP-1043 19790025020

NASA Reference Publication 1043

Processing of On-Board Recorded Data for Quick Analysis of Aircraft Performance

Norman H. Michaud

SEPTEMBER 1979

NASA

NASA Reference Publication 1043

Processing of On-Board Recorded Data for Quick Analysis of Aircraft Performance

Norman H. Michaud
Wallops Flight Center
Wallops Island, Virginia

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1979

Foreword

The National Aeronautics and Space Administration (NASA), Wallops Flight Center participated in the flight testing operations of the NASA/ARMY Rotor Systems Research Aircraft (RSRA) project. Flight testing began at Wallops in July 1977 and scheduled to terminate upon acceptance of the aircraft in August 1979.

This publication is based upon the set of computer programs designed to meet the 24 hour data processing requirement for "quick look" analysis. The entire set of programs have been designed under one software system called EASE (Early Analysis System Evaluation). This system can be adapted to similar helicopter projects requiring "quick look" information on helicopter performance. The basic structure of the EASE system is the design of the Raw Data File. Hence, modifications of project requirements based upon the method of input or output to the Raw Data File can be easily incorporated into the EASE system.

The EASE system was successfully designed with the aid of Mr. Ken Lewis of Sikorsky Aircraft and Mr. Jon Stripling of NASA, Wallops Flight Center. Individual software contributions to the EASE system were performed by Mr. Terry D. Sommers, and Mr. David L. Davis, and Mr. Donald Woodward of the Computer Applications Section, NASA Wallops Flight Center; Mrs. Jeannette Wessells, Mr. Tom Taylor, Mr. Joseph Lapierre, Mr. Gus Dovi, and Mr. Leslie Brimer of Computer Science Corporation.

CONTENTS

PROCESSING OF ON-BOARD RECORDED DATA FOR QUICK ANALYSIS OF AIRCRAFT PERFORMANCE

Norman H. Michaud
NASA Wallops Flight Center

INTRODUCTION

The Early Analysis System Evaluation (EASE) is a collection of independent software programs designed to provide an overview of flight performance within twenty-four hours. Because of the short time frame for processing flight data, this system does not provide detailed analysis of flight performance. However, sufficient information is available for such analysis. There are twelve major tasks that compose the EASE software system. These are:

(1) SETUP reads the information necessary to process data for a particular flight operation. These data are then stored on a random file for access and updating by other tasks in the software system.

(2) SCAN reads the digitized data tapes, performs minor editing, stores the data in the raw data file and computes the statistics in raw units.

(3) RDFRD is a set of subroutines used by the remaining tasks in accessing data and statistics from the raw data file.

(4) ABTASK computes the linear coefficients for the data based upon known laboratory measurements and pre-flight and post-flight calibrations.

(5) STATPRINT provides a formatted listing by sensor of the computed statistics from either the SCAN or DPTASK programs. Data can be represented either in raw units or engineering units.

(6) DPTASK derives certain information from the aircraft's instrumented data. These parameters, such as airspeed, are computed as a function of specific instrumented data. Such information is computed, then stored on the raw data file along with its statistics.

(7) DATASK provides a different formatted listing of the statistical data than the STATPRINT task. Up to 18 different sensors are grouped on an output listing in column format for comparative analysis.

(8) CYCLE COUNTS of certain data indicating aircraft stress points. By comparing a known stress point concern level value to the flight data, an output listing is generated by this task to indicate the number of data points exceeding 75 percent of this level in 25 percent incremental intervals.

(9) TIME HISTORY is a plotting routine of the data in engineering units versus time. Up to twelve different sensors can be plotted on one plot page.

(10) HARMONIC ANALYSIS computes the harmonic coefficients on sensor data for given flight maneuvers. The first ten coefficients for both the sine and cosine terms are computed as well as the resultant coefficients and the associated phase angles.

(11) UPCGEN maintains a catalogue of sensor identifications that are used by various tasks for titling the data on output listings. Data is referenced initially by mnemonic names through the SETUP task. These mnemonics are used to identify the information contained in the Universal Parameter Catalogue (UPC) file.

(12) TPDMP and TABLE PRINT perform utility functions for the EASE software. TPDMP provides a selected dump of the magnetic tape digitized by the EMR 6130 computer system. TABLE PRINT provides a listing of the current contents of the Table File portion of the Raw Data File.

During flight operations, data is being recorded only during experimental maneuvers. The system requires the data to be identified and processed by each maneuver of the flight. A d.c. signal is recorded during flight operations which determines the stream of data being recorded for a given flight maneuver. When this signal, called the run tone, is "on" the information contained on the data tapes is valid experimental data for a given flight maneuver. A run tone "off" condition indicates the termination of data gathering for that maneuver. Using this recorded run tone signal, each burst of data represents the information recorded for one flight maneuver.

Processing of the flight data through the EASE system groups the data by the identified flight maneuvers and calculates statistics for each flight maneuver. The data and statistics are stored on the raw data files (RDF). These files are classified as random files so that specified portions of the flight data are quickly accessible.

The data are initially recorded in analog form in two different modes: pulse coded modulation (PCM), and frequency modulation (FM). Different processing techniques are applied to the PCM and FM recorded data. PCM data is digitized at a fixed time sampling rate. The FM data is first processed through a peak stress analyzer. This step in the processing system is a hardware function that samples a channel of FM data at a sample rate based upon the revolution of the helicopter's main rotor or tail rotor. A sampling rate of five samples per main rotor revolution will produce five data samples for one 360 degree revolution of the main rotor.

A channel of data input to the peak stress analyzer produces two channels of output data. One channel contains the peak or maximum values of the input signal while the other channel contains the valley or minimum values. The peak stress analyzer used for the project can process up to twenty input channels of data at one time producing a total of forty channels of data -- twenty peak channels and twenty valley channels. These output channels of the peak stress analyzer are then digitized.

Because of the hardware limitation of twenty data channels available on the peak stress analyzer and the fact that data processing requirements dictate different main and tail rotor sampling rates for different types of data, data digitization must be performed in multiple passes of the data. Selected channels of data are processed together through the peak stress analyzer at a given sampling rate and digitized from beginning to end of the flight. This is defined as a pass of data. A second grouping of data channels is similarly processed, yielding a second pass of data. This continues until all possible FM data configurations have been processed through the peak stress analyzer.

The digitization process of the flight data therefore produces two types of digital data on 9-track magnetic tape. Both types of data not only use different processing techniques, but also produce two different types of formats. These formats are defined in the SCAN routine. The EASE system accepts both the PCM and FM data for processing.

All the information including the digitized data and computed statistics is maintained on the Raw Data File (RDF) for each flight. The RDF is a set of random physical files that contain all the necessary flight information. There are four basic file configurations established for the EASE system: the Table File, the Statistics File, the Data Files, and the Universal Parameter Catalogue (UPC) File. The UPC File is a common file used by every flight in the project while the other three are uniquely built for each flight. The Table File is initially built by the SETUP routine. It is a random file assigned to file code 07. The entire file is 42 logical records in length with each record fixed at 600 words. The SETUP routine documentation provides a detailed description of the Table File.

The Statistics File contains the statistics in raw (machine count) units for each sensor and computed for each flight maneuver. The file is a random file with each logical record having a fixed length of 400 words. This file is built by the SCAN routine onto file code 09 as it reads and stores the digitized data.

The configuration of the Statistics File conforms to the pass and flight maneuver constraint definition of the system. These constraints are defined so that data for no more than 40 sensors is to be grouped together for flight processing during one pass. The flight being processed can have no more than 100 defined flight maneuvers. Using these constraints, the positioning of data on the Statistical File can be defined.

Each sensor is assigned by the SETUP routine to a particular pass and to one of the 40 sensor positions of the pass. Each pass is numbered from 1 to 20 and the assigned position of the sensors in each pass is numbered from 1 to 40. The flight maneuvers are also assigned numerical values (called run numbers) from 1 to 100 by the SETUP routine.

The statistical data is randomly stored based upon this indexing scheme. The record address is a function of the pass number and run number. It is defined as:

100*(IPASS-1)+IRUN,

where IPASS is the pass number and IRUN is the run number.

There are 10 data words assigned for the statistical elements for each sensor. Since there is a maximum of 40 sensors per pass, each data record contains the statistics computed for each flight maneuver for a given pass. The defined position of each sensor through the SETUP routine determines which set of 10 data words contains the statistics for a given sensor. A set of statistics is found by the formula:

10*(IELEM-1)+IPOS,

where IELEM is the assigned element number of the sensor through the SETUP routine and IPOS is an indexed value from 1 to 10 for each statistical element for that sensor. The statistics stored within the 10 words index by IPOS are:

(1) the maximum value,

(2) the corresponding vibratory component of peak stressed FM data at the time of maximum steady value or the corresponding steady value at the time of maximum vibratory,

(3) the minimum value,

(4) the corresponding vibratory or steady component at the time of minimum steady or vibratory value, respectively,

(5) the average value,

(6) the standard deviation,

(7) the 95 percentile value for the vibratory FM data only,

(8) the total number of points processed for this maneuver,

(9) the total number of error points for the maneuver, and

(10) a 6-character error code status word. Characters 1, 2, and 3 are not used; character 4 non-zero value signals errors due to loss of synchronization; character 5 non-zero value signals over full scale data error; and character 6 non-zero value signals data parity error. A detailed description of the computation of these statistics and their meanings is given under the SCAN task.

The data file is also configured as random with each pass of data being assigned to one physical file. The file code for each data file is assigned a number that is 10 plus the assigned pass number. For example, the data file for pass 5 is assigned to file code 15.

The physical size of a record within these files are 640-words in length. The records are blocked by flight maneuvers. That is, whether or not a complete record is filled with data, a new record is begun with a change in flight maneuver. The configuration of the 640-word data file record is of two types: one type of configuration for data being processed and recorded through the SCAN routine, and the other for data being processed and recorded through the DPTASK routine. In either type, data is grouped into a frame of data with multiple frames within the 640-word record. A frame of data is defined as an array of data sampled or derived at a given time for all sensors described for a given pass. That is, the sensor grouping described through the SETUP routine for a given pass determines the array configuration for a frame of data. In both types of data configuration, the first two words of the data record contain the run number assigned to the flight maneuver and the number of frames of data that are stored on this record.

For the data processed through the SCAN routine, 11 data words are used for a frame of data. The first word of each frame is the recorded time in milliseconds for the data samples being recorded. Since the data being scanned has only a 9-bit accuracy, four data samples are packed into one 36-bit data word. Hence, ten words contain 40 samples of data. A maximum of 58 frames of data can be recorded into one 640-word record.

The derived parameter data, however, is calculated in engineering units and requires a full data word for each sample. There are, therefore, 41 data words required for a frame of derived parameter -- the first word being the time tag and the next 40 being the data samples. Fifteen such data frames can be recorded on a 640-word record.

The EASE software system package is flexible in its mode of operation. The distinct tasks that are required for a particular operation are executed as separate activities in the order that they are required. The normal flow of processing is first to define the data configuration for flight and sensor data constants through the SETUP routine; second to read the flight data and compute their statistics through the SCAN routine; third to compute the engineering unit conversion factors through the ABTASK routine; fourth to compute the derived parameters through the DPTASK routine; and fifth to provide the requested output data through the STATPRINT, DATASK, CYCLECOUNT, and TIME HISTORY plot routines. However, either because of the lack of complete information or because of data errors, this normal flow of EASE activities may have to be altered. These circumstances must be investigated by the system user when preparing a run for processing. Figure 1 gives a visual description of the EASE digital processing scheme.

Each task requires a specific card input setup that can be classified into three categories: first is the program deck with its associated job control cards; second is the job control cards for the file configuration; and third is the data cards required for flight parameters.

The deck setup for the computer program and its associated job control language is the standard configuration for the computer system being used. The job control cards for the data file configuration is also a function of the requirements for the computer system's file description requirements. The software requires specific file codes for each data file and expects the files to be configured in a random mode. The description of these files to the computer's operating system is dictated by the particular computer file description requirement. Appendix A summarizes the job control language required to execute each program on the Honeywell 600 computer system. The remainder of this section describes the use of each task in the EASE system along with their data card descriptions.

SETUP

The purpose of the SETUP routine is to build the Table File. The Table File is used to describe the Data File format and flight parameters for a given flight. There are five tables that are defined within the Table File. These are the Header Table, the Pass Table, the Sensor Table, the Calrun Table, and the Event Table. These tables are built through card input.

The format of the Table File is defined as a random file consisting of a total of 42 physical records. Each record is pre-defined as a particular type of table. The structure of the Table File is described in figure 2. The SETUP routine initially builds these files through card input.

This task is designed for interaction with an input card deck for properly transferring the table information into the respective record areas of the Table File. Communication with the program is primarily performed by using unique two-character codes in columns one and two of the input card deck. The necessary communication required by the program is divided into three major categories. The first category is called the "L1" level of input. This level of input informs the software to begin processing in either an initialize or update mode and to terminate processing. These cards must be the first and last cards of the input data deck setup. The second category, called the "L2" level of input, informs the software that the following data cards in the deck setup contain the data values for a particular table. All data cards following an "L2" level card up to the next "L2" level card or the final "L1" level card contain the data values for the table specified by the leading "L2" card. The detailed entries for each table form the third category of data.

The order of input for the Header Table and the Event Table is unimportant. However, the information for the Pass Tables is located on an "L2" level card that precedes the Sensor Table data for that named pass. Also, the Calrun Table data indicates which events contain pre- and post-calibration data for a named sensor. The program requires that the Sensor Table be present in order to cross-reference the two tables.

The "L1" level cards used in the SETUP routine are described in figure 3. The option name specified on the first card informs the program that the execution of this task is either to initialize the Table File or update the current Table File with the input data cards. The update option assumes that only those input cards present require value changes in the current data file. There are a few remarks that must be stated for proper execution of this program in the update mode:

(1) If the value of only one data field is to be modified, all information that is specified for that card image must be present. The SETUP routine reads a card image and will modify each data value that is expected on that data card.

(2) Because of the interrelationship of the Pass Tables, Sensor Tables, and Calrun Tables, any modification to a given Pass Table or to any one sensor referenced to a given pass requires that the Pass Table, all sensors referenced to that pass and the Calrun Tables must be included in the deck setup.

(3) A modification to the Calrun Tables do not require the inclusion of the Pass or Sensor Table data. However, the entire set of Calrun Table data cards must be included for one change in the Calrun Table setup.

One additional remark that pertains to the execution of this task in either the initialize or update mode is that the Calrun Tables must be placed after the Pass and Sensor Table data cards.

The Header Table consists of four data cards preceded by an "L2" level card. Input for the initialize mode will store blank or zero values for data fields left blank or for data fields defined for a missing data card. The update mode requires all the data fields on a supplied input card to contain the proper data values. Data fields that are blank will cause blank or zero data to be stored in that Header Table data element. Only those card images requiring data modifications need to be supplied. Figure 4 contains a description of the data elements in the Header Table and figure 5 gives the data card image format for this table.

The Pass Table is contained in records 2 through 21 of the Table File. Each record contains information pertaining to one defined data pass. The EASE system is thus limited to a maximum of 20 data passes. Each data pass is assigned, through card input, a pass number from 1 to 20. This pass number is used to assign the record number for storing the Pass Table data in the Table File. Pass Table data for pass "i" is assigned to record number "i + 1" of the Table File.

A pass has been defined as the processing of flight data for a group of sensors from beginning to end of the flight at a common sampling rate. Since the peak stress converter is capable of processing no more than 40 output channels of data at one time, the EASE system is designed to process no more than 40 channels of data on one pass also. This is true in the EASE system even in handling the PCM data which is digitized with more than 40 channels of data at a time.

There is a natural association of the Pass Table with the Sensor Table because of the definition of a pass. The order of card input for the Pass Table information on an assigned pass is associated with a specific set of sensors. The card image description for the Pass Table is, therefore, described along with the Sensor Table in figure 8.

The data elements within each Pass Table record is described in figure 6. The data contents of the Pass Tables will change to reflect information derived from the processing of other tasks. In particular, the pass status word, ISTATC, will be updated by the SCAN routine to indicate the data is now available and it will be updated by the ABTASK routine to indicate the engineering unit conversion factors are now available. The array IADDR will be updated to reflect the data file addresses for each maneuver.

The Sensor Table contains information necessary to fully describe each data parameter processed in each pass. There is a maximum of 40 sensors described in each pass of data. The set of parameters processed together for a given pass are being sampled at a common sampling rate. This is the basic requirement for grouping parameters for a defined pass. The Sensor Table information is stored in records 22 through 41 of the Table File. The Sensor Table data for pass "i" is found in record number "i + 21" of the Table File. By using the address of the Table File records, the EASE system can therefore determine the pass associated with the Sensor Table data. Using a similar design feature, the contents of each Sensor Table record defines the ordering of the parameters within a pass.

Each parameter requires 13 data words for its description. With a maximum of 40 parameters per pass, a Sensor Table data record contains a maximum of 520 data words. The 13 data words for a given sensor "j" are then found in data word location "13j - 12" through "13j." The information contained in the 13 data words for each sensor are described in figure 7. Figure 8 outlines the data card images for the pass and sensor data for a given flight pass.

Data processed through the peak stress converter require specific ordering for Sensor Table entries. One data signal entering the peak stress converter produces two components of the data - a peak and a valley. In the case of data defined with preprocessing codes of vibratory and steady, the peak and valley components are mathematically combined to produce these two components. In the case of amplitude and phase components, two data signals are entered into the peak stress converter - one is the amplitude signal and the other the phase signal. Four data components are then produced - two for the amplitude and two for the phase.

The order of card input to the Sensor Table is important for data processed in this manner. For the vibratory and steady component sensor definition, the card image describing the vibratory component must precede the card image defining the steady component; for amplitude - phase data, the amplitude component must precede the phase component. The assignment of storage location for these components in the Statistics and Raw Data Files must also be in a similar ordering scheme.

The Calrun Table is used to associate the correct pre- and post-calibration maneuver number to every sensor on the flight. There are three different types of calibrations performed both before the flight begins and soon after the flight ends. Data is recorded with full loads on the instrumentation for the R-Cal; with no loads for the Z-Cals; and full stick and pedal positions for the transducer or X-Cals. Each type of calibration is performed in a definite sequence and the data for each calibration step is recorded as a burst similar to a flight maneuver.

Not all calibration steps are applicable to every sensor. A series of calibrations are performed so that the necessary calibrations are available for each active sensor. The Calrun Tables are used to indicate which data maneuver numbers contain the correct information for each sensor.

There are two types of input cards required for the Calrun Tables. One specifies the six assigned maneuver numbers for pre- and post-calibrations for a group of sensors. The other specifies the sensors associated with that calibration scheme. The order of input is important. The card containing the calibration run numbers must precede those containing the sensor mnemonics associated with that set of run numbers. The cards containing the run numbers are identified by the code "CT" punched in columns 1 and 2. The cards containing the sensor mnemonics are identified by the code "CS" in columns 1 and 2.

The first set of calibration run numbers are associated with a majority of the sensors activated for the flight. Rather than list such a long list of sensors mnemonics, the first grouping of data cards for the Calrun Table contains the characters "AC" punched in columns 4 and 5 of the "CS" card that are associated with the "CT" card read. At most, eight sensor mnemonics can be specified on one "CS" card. Since more than eight sensors can be associated with a defined calibration run scheme, more than one "CS" card is acceptable for input. The procedure of associating a sensor to the last table of calibration run numbers is by either encountering a blank field for a sensor mnemonic on a "CS" card or encountering a new "CT" card. A maximum of 20 calibration run table entries can be specified.

The entry numbers associated with a calibration run table are calculated by the software. Each set of tables are numbered consecutively as they are read by the software. These entry numbers are entered into the Sensor Tables as the value of CALADR for the proper sensor.

This entry number also is used in the formatting of the Calrun Table record of the Table File. All Calrun Table entries are recorded in record number 42 of the Table File. Each entry contains six values that are the calibration run numbers. With a maximum of 20 entries, the Calrun Table can have at most 120 words of data. Data words "6i-5" through "6i" contain the calibration run numbers for entry "i". Figure 9 is a description of the Calrun Table and figure 10 is a description of the card images for this table.

The Event Table is a list of the flight maneuver information. It contains a 24 character field used to define a title for the experimental maneuver, the type of maneuver being performed and the time the event began. This table is an important aspect of the system since the data is being evaluated for each maneuver.

The card input requirements for this table are relatively straightforward compared to the Pass, Sensor, and Calrun Tables. The information required for each event is placed on a single input card. The order of input is dictated by the order of the maneuvers as given on the Flight Log. A run number is assigned to each maneuver.

The information for the Event Table is stored in record 43 of the Table File. There is a maximum of 100 maneuvers that can be defined for a given flight. Each maneuver requires six data words for its second entry making the Event Table size a maximum of 600 data words in length. The entry for run number "i" is placed in data words "6i-5" through "6i".

The data field contents for each maneuver in the Event Table is described in figure 11. The card image description for each maneuver is described in figure 12. The order of input for the Event Table relative to the input of the other four tables is irrelevant. Only the order of the data cards within the Event Table is important.

The SETUP routine provides the initial description of flight parameters. Items that are included in this description are the aircraft configuration information, the configuration of the digitizing process by passes, the sensor configuration of the data being processed, the pre- and post-calibration configuration for each sensor, and the description of the events that were performed on the flight.

This routine can be operated in two modes. The first is the initialize mode where all data entries for each table are initialized to the data provided on input cards. The second mode is the update mode where information supplied on data cards alter the previously stored Table File data.

The SETUP routine provides a limited editing capability on the set of input data cards. When errors are found in the setup of the input data card deck that can be checked by SETUP, appropriate error messages are printed and none of the information provided by the execution of this job is recorded. This is to insure that additional errors are not included into the Table File especially during an update mode. Items that are checked include the correct order of data cards, data fields using a specific numeric or character code entries contain only the allowable entries, and the maximum allowable sizes of each table is not exceeded. Items that cannot be edited include such things as incorrect spelling of character entries, incorrect entries of allowable codes, and incorrect data value entries that fall within the allowable range of the data. Care should therefore be taken to insure all entries to the Table File are correct. A detailed check using the output listing provided by the SETUP routine should be made to insure correctness of the information recorded in the Table File.

SCAN

This program is used to record the digitized data onto the Raw Data File, and compute and store the defined statistics. This program is the data processing task for the EASE system. The SCAN program accepts digitized data produced in the Televent II[1] format and sequentially reads each time-tagged frame of data.

Each frame of data is a set of digitized data samples in the form of an array. The association of a data sample within a frame of data to a given sensor is defined through the SETUP program for each flight. The variable, "IPARIN" of the Sensor Table, indexes the appropriate data word within the data frame. Each referenced data sample is also stored into the Raw Data File in a position determined by the variable "IPARDF" of the Sensor Table. The flight data to be processed is divided into passes, digitized by different pre-processing schemes, originally recorded as different types of data, and partitioned according to a prescribed set of flight maneuvers.

A pass of data is a complete set of data frames for a flight configured according to the Sensor Table for that pass. Each pass is numbered through input to the SETUP program. The location of data for each sensor is then fixed during the processing of a pass by the set of "IPARIN" indices noted in the Sensor Table for that pass.

The type of data being processed must also be known since the data is digitized differently for each type of data. There are only two types of data that are recorded· FM and PCM data. Both use the Televent II recording scheme but the length of their data frames and the bit configuration of each data sample are different. A PCM data frame is 110 data words in length and each data sample contains nine bits of data plus one odd-parity bit. An FM data frame is 42 data words in length and each data sample contains nine bits of data plus three bits of "noise" data. Hence, the decoding of information requires that SCAN recognizes these differences. This is given through the variable "SRDIN" of the Pass Table input through SETUP.

The pre-processing code specified through the variable "PPC" of the Sensor Table informs the SCAN program on how to handle the input data. The pre-processing for FM data may require that the information pass through the peak stress converter before digitizing. This would provide a pair of data points for each sensor. These points are the maximum and minimum data values within the specified sampling interval. These points are then combined in SCAN to produce the vibratory and steady components of the sensor. Data having pre-processing codes of amplitude and phase are also processed through the peak stress converter but are processed in the SCAN program differently than vibratory and steady data.

[1] See Appendix B for a description of the Televent II format.

Finally, the partitioning of the data by a prescribed flight maneuver scheme must also be known by the SCAN program. This is normally provided by a run tone signal that is digitized and recorded within each frame of data. When the run tone is "on," that is, a high data value recorded, the data contained within that frame of data is to be processed. When the run tone value drops to a minimum value, this signals the end of data sampling for a flight maneuver. The run tone switching to a high value in a successive frame of data signals the beginning of a set of data frames for the next maneuver. Such a scheme for partitioning the data into separate maneuvers is not assumed to be foolproof so that a means of identifying the partitioning scheme is provided as card input to the SCAN program. The partitioning of sets of data frames into bursts permits the program to calculate statistics for each maneuver.

The SCAN program calculates the required statistics for each sensor as the data is being read. There are ten statistical values recorded for each flight maneuver on each sensor. These are the maximum and minimum points, average, standard deviation, total number of samples in the event, number of error points and coded word denoting the type of data error. In addition to these seven values, data processed through the peak stress components for a sensor require the corresponding component value at the time of maximum and minimum recorded value for the vibratory and steady data. That is, the data value of the vibratory component at the time of maximum and minimum steady is recorded in the statistics record. The steady values are similarly recorded at the time of maximum and minimum vibratory values. The 95th percentile point is the tenth statistic. It is only computed for vibratory component data.

It is then evident that the processing requirements are rather complex and require additional information to relate the SCAN program as to how the data is presented in the input file and how it is to be processed and transferred to the raw data file. Some of the information is available from the Table File stored by the SETUP program. However, some of the information that is needed must be supplied directly to the SCAN program. This is done through card input.

There are three types of data cards that are used by the SCAN program. Each is distinguished by a two character coded value in the first two columns. The card coded by "SD" is the pass configuration card. The card coded by "BT" is the burst processing card and the one coded by "L1" is a processing termination card. Figures 13, 14, and 15 outline the information required by each type of input card.

The order of card input follows the logical sequence of processing the data. The first data card must be an "SD" card in order to inform the SCAN program as to which pass is to be processed and where the data is to be found on the input file. Following the "SD" card are the "BT" cards that instruct the program to process those data bursts that cannot be processed by the run tone signal. If more than one pass is to be processed, another "SD" card followed by its required "BT" cards are then read. This scheme continues until all passes to be processed are defined. The last card of the input data deck is the "L1" card.

There are four types of conditions for processing a burst of data. One is by the run tone signal, another by time, a third by skipping an erroneous burst of data and lastly by defining a burst when one was not present on the input file. These four types of conditions are defined by the variable "ICOND" on the "BT" card. "BT" cards need not be present to describe the processing mode of each burst of data. Only those conditions where processing of a data burst other than run tone need to be included in the input deck. If a "BT" card is not present for a numbered burst, it will automatically be processed using the run tone. Figure 14 contains the coded values of "ICOND" for the different burst selection conditions.

In order to process data through the SCAN program, supporting documentation must be available with the input file. This includes a burst interval listing and a flight log. The digitized data is pre-processed using the run tone signal onto a magnetic tape. At the time of digitization, a burst interval listing is generated noting the time interval for detected bursts by run tone. The total number of these intervals must coincide with the total number of events outlined on the flight log. When there is a discrepency between these two pieces of information, both forms must be studied to determine where the discrepency is and how to correct it. This is done by inserting "BT" cards into the input data deck for the SCAN program specifying the processing condition for that burst of data. For example, it may be that burst number ten is a set of data frames with the run tone having been keyed on when an actual maneuver did not take place. This burst would then have to be skipped in the processing of the data tape by using a "BT" card for burst number ten with "ICOND" equal to one. Another possibility is that the run tone was not turned off between two maneuvers so that two maneuvers have been detected as one burst of data. In this case, the "BT" cards would contain a value of two for "ICOND" along with the maneuver time intervals. The SCAN program would then process these bursts by time intervals. Finally, there may have been a defined maneuver according to the flight log but the run tone signal was not on. In this case no data is present on the input tape and a "BT" card would be needed with "ICOND" equal to three.

The SCAN program is the main processing routine for the EASE system. It requires execution of the SETUP program in order to define the configuration of the data on the pre-processed magnetic tape. A study of the flight log and a listing of the burst intervals contained on the input file is required to properly align each burst of data to the prescribed flight maneuvers. Samples of the burst interval listing and flight log are found in figures 16 and 17.

RDFRD

This is the subprogram for accessing data and statistics from the Raw Data File (RDF). It is designed to use as a sub-module of each program requiring access to information on the RDF. This subprogram consists of three subroutines - RDFRD, RDFSUB, and UNPACK. The subroutine, RDFRD, accesses data samples or statistics requested through the input argument list and returns the information to the calling program either in raw units or in engineering units. If the requested data is not available, RDFRD returns the appropriate error codes which can be interpreted by the calling program. The subroutine RDFSUB is used when the calling program furnishes only the mnemonic name, pre-processing code and sample rate for the requested data. RDFSUB then searches the Table File to determine the pass and element number as well as the engineering unit coefficients so that RDFRD can locate and convert the requested information. UNPACK is used by the subroutine RDFRD when data samples are requested by the calling program. Each data sample within the RDF has been packed so that each computer word contains four successive data sample values. UNPACK extracts the appropriate "byte" for the requested data samples.

Figure 18 is a list of the arguments for subroutines RDFRD and RDFSUB. Since the subprogram RDFRD is not an independent program, there is no card input directly required by this subprogram. All information is furnished through the appropriate program that accesses the RDF and thus requires the use of the RDFRD subprogram.

ABTASK

This program computes the engineering unit coefficients for linear conversion of the data contained on the RDF. The operation of this program depends upon information on the Table File. No card input is required at the time of execution of the ABTASK program. The engineering unit coefficients are computed for every sensor defined in the Table File whose data has been entered into the RDF by the SCAN program. Hence, successful execution of the SETUP program and SCAN program is required before this program can perform meaningfully.

ABTASK first checks the present value of "ISTATC" of each Pass Table record (see figure 6). The engineering unit coefficients will only be calculated if "ISTATC" equals one. This indicates that the data has been scanned and entered into the RDF but the engineering unit coefficients have not yet been calculated. Once a pass has been found with such a condition, each entry of the Sensor Table for that pass is polled for additional information.

Within the Sensor Table, the data field contents of "PPC," "IPARDF," "CALTYP," "EUCV1," "EUCV2," and "CALADR" are needed by ABTASK. (See figure 7 for the definition of these variables.) The engineering unit coefficients, calculated by ABTASK, are stored in the Sensor Table under the variables "A" and "B". Some of the information extracted from the Sensor Table play a minor role in the logical flow of the program. The value of "PPC" satisfies the requirement for computing the engineering unit terms for vibratory data. For the vibratory component data, the "A" term is as computed for the associated steady component while its "B" term is zero. The variable "IPARDF" is used to locate the appropriate average and standard deviation values for each sensor from the Statistics File, and the variable "CALADR" references the appropriate Calibration Run Table entries for that sensor. The variable "CALTYP" determines the type of calibration procedure required for each sensor. Because of the varied types of instrumentation available, several types of calibration procedures have been established. These procedures then require different types of linear conversion equations to compute the engineering unit terms. These have been categorized into five types for ABTASK: ACX, ACZ, PC, AB1, and AB2.

In addition to defining the type of calibration for each sensor, the total range and offset values in engineering units are supplied through the variables "EUCV1" and "EUCV2" respectively. The actual raw unit measurements for these values must be supplied also. These are furnished through the pre- and post- calibration steps performed for each flight. This information is contained on the RDF and is determined by the Calrun Table referenced by each sensor.

The equations for computing the A and B terms for each type of calibration is as follows: For the ACZ type of calibration,

$$A = \frac{EUCV1}{\overline{R} - \overline{Z}} ,$$

where

$$\overline{R} = (PRE*\overline{R}_e + POST*\overline{R}_t)/(PRE+POST):$$

$\overline{R}_e$ = average in raw units (RU) of the pre- R cal run;

$\overline{R}_t$ = average RU of post- R cal run;

PRE = 0 if there is no pre- R cal, = 1 if there is;

POST = 0 if there is no post- R cal, = 1 if there is;

and

$$\overline{Z} = (PRE*\overline{Z}_e + POST*\overline{Z}_t)/(PRE+POST);$$

$\overline{Z}_e$ = average RU of pre- zero cal run;

$\overline{Z}_t$ = average RU of post- zero cal run;

and

$$B = EUCV2 - (A*\overline{Z}).$$

For the ACX type of calibration, the engineering unit terms are

$$A = \frac{EUCV1}{\overline{R} - \overline{Z}} ,$$

and

$$B = EUCV2-(A*\bar{X})$$

where

$$\bar{X} = (PRE*\bar{X}_e + POST*\bar{X}_t)/(PRE+POST);$$

$\bar{X}_e$ = the average RU of the pre- X ducer cal run;

$\bar{X}_t$ = the average RU of the post- X ducer cal run.

For the physical calibrations (type PC)

$$A = \frac{EUCV2-EUCV1}{\bar{H} - \bar{L}}$$

where

$$\bar{H} = (PRE*\bar{H}_e + POST*\bar{H}_t)/(PRE+POST):$$

$\bar{H}_e$ = average in RU of the pre- physical high cal run;

$\bar{H}_t$ = average in RU of the post- physical high cal run;

and

$$\bar{L} = (PRE*\bar{L}_e + POST*\bar{L}_t)/(PRE+POST):$$

$\bar{L}_e$ = average in RU of the pre- physical low cal run;

$\bar{L}_t$ = average in RU of the post- physical low cal run;

and

$$B = EUCV1-(A*\bar{L}).$$

For the AB1 type of calibration, no data is used for calculations
and

$$A = EUCV1,$$
$$B = EUCV2.$$

And for the AB2 type of calibration, no data is required for calculation,

$$A = \frac{EUCV1}{(EUCV2/100.0)*ACS}$$

where

$$ACS = 512.0$$

and

$$B = 0.0$$

The average raw unit values used in the calculations for the ACZ, ACX and PC types of calibrations are referenced by the run number values specified in the Calrun Tables (see figure 9). If any entry to the Calrun Table is zero, there is no average value for that type of calibration. This method is used when either the pre- or post-calibration values are required for a given sensor. In those cases where the Calrun entry is zero or the average value is null for both the pre- and post-calibrations the A and B terms cannot be calculated. The program then assigns A = 1.0 and B = 0.0. The calculated A and B terms are then entered into the appropriate Sensor Table position. When all sensors for a given pass have their engineering unit coefficients calculated, the "ISTATC" variable in the Pass Table is redefined as "2."

Three types of printout formats are produced by ABTASK. These reports summarize, on a pass basis, the information used to calculate the engineering unit coefficients as well as the coefficient values. Figure 19 lists the event numbers used for each Calrun Table entry used for the sensors in a pass. These run number entries correspond to the assigned event numbers associated with the Event Table (see figure 11).

The average value in raw units, its percent of full scale, the percent full scale shift between the pre- and post-calibration values and the standard deviation values is presented in figure 20. Percent full scale is based on the mid-range of the data being 0 percent. A value of 511 for the average value in raw units represents the null value. An outline of the final A and B terms calculated for each sensor along with the type of calibration, the calibration run table entry number and the given EUCV1 and EUCV2 terms is contained in figure 21.

STATPRINT

The STATPRINT program furnishes a tabulation of the statistical values for each flight maneuver for a given sensor. The statistical values are presented either in raw units or engineering units. Figure 22 is a sample of the report produced by the STATPRINT program. The purpose of this program is to furnish a detailed report on the statistical values for a given sensor. It is not a practical summary list for normal flight analysis due to the bulk of data that would be produced on all sensors of a given flight. Its primary use is in furnishing this data when detailed analysis is required.

The mode of operation for the STATPRINT program is controlled by card input. There is only one input card format required for any mode of operation. Figure 23 outlines the input card format for this program. There are three basic modes of program execution: a request for statistics of a specified sensor; a request for statistics for all sensors of a specified pass; and a request for statistics on all sensors within a given flight. The reports will be in either raw or engineering units depending upon the value of "ICONV" specified on the input card. There can be any number of input cards depending on what statistical reports are requested.

In order to furnish statistical data for a given sensor, information must be furnished for the mnemonic name, pre-processing code, and sample rate of the data

("NAME," "PPC," "SR"). This must be identical to the same information present in the Table File as input through the SETUP program. The STATPRINT program utilizes the subprogram RDFRD to locate the position of the statistics on the RDF by searching the Table File for sensors having the information from the cards read by STATPRINT. Failure to find the entry in the Table File results in an error message rather than the required statistics tabulation. When all sensors for one pass is required as output, the correct pass number in variable "IPASS" is all that is required for input. To process the statistical tabulations for all sensors of a given flight, no pass or sensor information is given on the input card.

In addition to specifying which of the three modes of output is requested, the request must include the appropriate code for "ICONV" for either raw or engineering values. An error will be noted if engineering unit values are requested and ABTASK has not been executed for the requested data.

DPTASK

The Derived Parameter or DPTASK program computes the values of selected parameters that are not directly sensed and recorded during flight. These parameters are derived as functions of directly sensed data. The input data is extracted from the RDF to calculate the derived parameter data. Statistics on the derived parameters are computed in the same manner as the directly sensed data.

The data and statistics computed in the DPTASK program is in engineering units. This is the exception to the raw data file format where all data stored in the RDF from the SCAN program is in raw units. Hence, ABTASK need not be executed for the derived parameter data. Consequently, this data cannot be extracted in raw units nor can the data values be packed into four data samples per computer word.

The calculation of the derived parameters follow the same general format for processing directly sensed data. Parameters are grouped similar to the normal grouping of sensor data with no more than 40 sensors per derived parameter pass. The definition of these sensors is controlled by the subroutine DPDEF which assigns the mnemonic names, pre-processing codes, and sample rates. This information is transferred to the Table File making appropriate entries to the Pass and Sensor Tables. The data and statistics are then assigned to the RDF according to the entries to the Table File. Accessing of derived parameter data can thus be performed through the RDFRD subprogram with only minor modifications due to the data not being packed. Calculation of the defined parameters is also performed through the subroutine DPDEF. Modifications to the set of derived parameters requires software changes to the subroutine DPDEF only. These modifications are transparent to the remainder of the DPTASK program or to the rest of the programs in the EASE system. Appendix C contains the current set of derived parameters defined by subroutine DPDEF.

There is one data card required as input to the DPTASK program. As mentioned earlier, the subroutine DPDEF assigns a set of parameters to a given pass and builds the Pass and Sensor Tables for these sensors onto the Table File. Within DPDEF, the sensors are grouped into passes called derived parameter passes. The derived parameter pass must be assigned a unique pass number to fit within the other assigned pass numbers of a given flight. The input data card is used to assign a flight pass number to a derived parameter pass being processed. Figure 24 defines the card image for input to DPTASK.

DATASK

The Data Analysis or DATASK program furnishes the user a report of the pertinent statistics for up to 18 sensors on one report page. This provides a comparative view on the behavior of interrelated sensors during each flight maneuver. A sample of the printout report provided by the DATASK program is found in figure 25. There are several aspects of this printout that must be explained in order to understand the information presented.

Each page of printout contains four lines of title information, noted as (1) on figure 25. This is provided on input data cards. The column titles (2), are also initially provided on input cards that specify the mnemonic name, pre-processing code, and sample rate of the requested data. The mnemonic name and pre-processing code is then translated to a column title (3), if found in the Universal Parameter Catalogue (UPC) File (see UPCGEN program). The sample rate of the data (4) is listed under the sensor's column title. The titles provided for each line (5) or series of lines (6) is the name of the flight maneuver. This information is extracted from the Event Table of the Table File. Within the Event Table, each flight maneuver is either classified as a calibration, steady or transient maneuver. The DATASK program only lists the statistical data for non-calibration maneuvers unless specified differently by card input to this program. Only one line of output is required for steady maneuvers (5) and three lines are required for transient maneuvers (6).

The statistics listed for a steady maneuver are the 95th percentile for vibratory data and the average for all other data. For a transient maneuver the data is listed in the following format:

	PPC =D	PPC =V	PPC =S	PPC =A	PPC =P
Line 1	STAT5	STAT4	STAT3	STAT5	STAT5
Line 2	STAT1	STAT2	STAT1	STAT1	-
Line 3	STAT3	STAT1	STAT2	STAT3	-

where

STAT1 = the maximum value for the flight maneuver,

STAT2 = for vibratory, the corresponding value of the vibratory component of the sensor when the steady component is its maximum value; for steady, the corresponding steady value for the maximum vibratory value.

STAT3 = the minimum value for the flight maneuver,

STAT4 = the corresponding vibratory value for the minimum steady value,

STAT5 = the average value of the data for the flight maneuver,

PPC = pre-processing code, where D = Direct, V = Vibratory, S = Steady, A = Amplitude, and P = Phase.

The report does not explicitly indicate the type of flight maneuver but it is implied by the number of output lines presented. The DATASK program determines this by checking the event type code for each maneuver referenced in the Event Table. One further remark in regard to the DATASK output report is that a blank field indicates that there are no statistics for that parameter during the indicated flight maneuver. An entire column of blanks could indicate that the referenced parameter was not on during the entire flight. Or, such a condition could also indicate that the input card may be incorrectly referencing a parameter either due to a mispunch of the mnemonic name, preprocessing code, or the sample rate.

There are three types of input cards required by the DATASK program: the title cards, the maneuver specification cards, and the parameter specification cards. The title cards must be the first four cards of the input card deck and each contains any alphanumeric characters in columns 1 through 72.

The maneuver specification cards are next in order of card input. These input cards allow specific flight maneuvers only to be reported on the printout. The flight maneuver number associated with each run as described for the Event Table is entered right-justified in columns 2-6. As many maneuver specification cards as required are read by the DATASK program. A card with the word "END" in column 1-3 terminates the processing of maneuver specification cards. Under normal operations, the requirements for the DATASK output request all non-calibration runs to be processed. Rather than input a large set of data cards for this mode of operation, the DATASK program assumes such a procedure when no maneuver specifications card except the "END" card is present.

The parameter specification cards are then processed into the program. These cards are grouped into pages with each page containing up to 18 parameter cards each. An input data card with the word "PAGE" in columns 1-4 is required in describing a page grouping. This card is then followed with up to 18 parameter identification cards with one sensor described per card. The final page group being processed must be followed by a card with the word "END" in columns 1-3.

The format for the parameter identification cards are: Columns 1-8 (left-justified) is the sensor mnemonics; column 10 is the pre-processing code; and columns 15-17 (right-justified) is the sample rate. The input card description for the DATASK program is outlined in figure 26.

CYCLE COUNTS

This program operates only on the vibratory component of parameters pre-processed through the Peak Stress Converter. It provides a count of the number of vibratory data values which occur in each of eight class intervals. The lower edge of the first class interval is defined as 75 percent of the stated concern level value in the Sensor Table for that parameter. Each successive class interval up to the eighth and final class is defined in increasing steps of 25 percent of the concern level. The final class interval counts the number of data samples above 250 percent of the stated level of concern.

A sensor's vibratory component will be cycle counted for a given flight maneuver only if the following conditions are satisfied: (1) a concern level value greater than zero has been defined for the sensor in its Sensor Table; (2) either the 75 percent level is reached for at least one data sample within an event or that one-half the difference between the absolute maximum and absolute minimum is reached during an event; and (3) cycle counting has been specified by card input for a given flight maneuver or parameter. The absolute maximum level is defined as the maximum of the sum of the steady and vibratory components of the sensor (S+V) and the absolute minimum level as the minimum of (S-V).

The type of card inputs which define what vibratory data is to be cycle counted are the maneuver specification cards and the parameter specification cards. Maneuver specification cards are required only when specific events are to be cycle counted. Otherwise, all non-calibration events will be cycle counted. Parameter specification cards are required for two modes of parameter selection: all available vibratory parameters for a specified pass; or only for specified vibratory parameters. The default mode is to cycle count all vibratory components for the flight. A four-card title specification is also included as part of the input data deck. The title cards must be present to execute the CYCLE COUNT program. Figure 27 outlines the data card input requirements.

A sample of the output report produced by this program appears in figure 28. The area on the report indicated by (1) is the title specification information as it appeared on the input cards. The line indicated by (2) includes the sensor name, sample rate, and the value of its concern level. The column titles indicated by (3) define the statistical values presented for each flight maneuver that was cycle counted. The column, "N TOTAL," (4) indicates the total number of data samples for that flight maneuver. The remaining column headings indicate the eight classes (5). The last line of the report (6) contains the absolute maximum and absolute minimum for the flight and the total number of samples in each of the eight class intervals. Each sensor that has been cycle counted is reported in this manner.

TIME HISTORY

The function of the Time History program is to generate a plot of time history data from the RDF. The basic format consists of a 25 x 42 cm plot of up to 12 parameters. Arrangement of the parameters is four rows of three parameters each. Parameters from the first (left most) column are plotted with a solid line. Parameters from the center and the right most columns are plotted with long and short dashed lines, respectively. All y-axes for each parameter are four cm in length.

The time scale (x-axis) is fixed at 30 cm in length. The time scale is automatically adjustable to 1.5×10^J, 3.0×10^J, or 6.0×10^J, where J is an integer which will create a sufficiently large scale in seconds for the time span plotted. The time scale is defined as elapsed time of a flight maneuver.

The arrangement of the 12 parameters on a plot is defined as a plot group format (PGF). Provisions have been established to permit up to 40 parameters to be defined each in up to nine PGFs per program execution. This permits data comparison of flight information for different parameters. In the plotting of data values, if null samples are encountered, the plotting pen is lifted resulting in a visible discontinuity in the graph.

Most of the labeling and PGF assignments are controlled by card input. There are five types of input cards described for the program. These types, in their prescribed order of input, are: (1) General Title Card; (2) Plot Row Labels Card; (3) Parameter Specifications Cards; (4) Heading Label Cards; and (5) Burst Selection Cards. The General Title Card must be the first card of the input data deck. It contains any alphanumeric characters in columns 2 to 73. The information contained on the card is printed at the top of all plots. (See (1) of figure 33.)

The Plot Row Labels card prints a two-line row label on the plot, each line containing at most 18 characters each. There are four rows per plot and they are numbered from one to four - top to bottom. The row labels are specified by PGF and row number. Card format for the Plot Row Labels is found in figure 29. An example of the labeling of the plot row label is designated as (2) in figure 33. A maximum of 36 (9 PGFs x 4 rows) plot row label cards may be included for a job run. Termination of the plot row labels card input is with a card containing zero in column 5. It is not necessary to provide any information for row labeling. However, the deck setup must include the card with zero in column 5.

The Parameter Specification Cards consist of two input cards per parameter. Figure 30 describes the format of these cards. Up to 40 parameter card groups may be defined for plotting. Termination of this group of card input is by including a card containing blanks in columns 1-8 at the end of this group.

The fourth group of input cards are the Heading Label Cards. This information is placed beneath the General Title Card information on the plot and is designated as (3) in figure 33. Up to eight heading items may be defined on a plot. Termination of the Heading Label Cards is by including a card with zeroes in columns 1-5. Description of the Heading Label Cards format is found in figure 31.

The final set of input cards are the Burst Selection Cards. Two cards are required per burst. A description of these cards are found in figure 32. The Burst Selection Card group also requires a termination card which contains a zero in column 5.

Direct output of the Time History Plot Program is a tape formatted to produce the requested plots on a CALCOMP plotter. The information has been scaled to a metric scale and produced on a 25 x 42 cm area. A sample of a plot produced by this program is found in figure 33.

HARMONIC ANALYSIS

This program determines the harmonic coefficient and phase angles for the Fourier series expressed as:

$$Y = A_0 + A_1 \text{Cos}X + B_1 \text{Sin}X + A_2 \text{Cos}2X + \ldots\ldots$$

and

$$Y = C_0 + C_1 \text{Cos}(X-\phi_1) + C_2 \text{Cos}(2X-\phi_2) + \ldots\ldots\ldots$$

Given a set of data values for a given parameter sampled at a fixed sample rate, the coefficients, A_i, B_i, and C_i, and the phase angles, ϕ_i, are computed on a cycle by cycle basis. The values of each coefficient and phase angle computed during each cycle are averaged and presented in a printout on a flight maneuver basis. A sample of the printout report is found in figure 34.

The definition of the data set used to compute the harmonic coefficients and phase angles is established through card input. Two types of information must be included for calculating the harmonic coefficients: a list of the flight maneuvers and a list of sensors and number of cycles to be processed. The order of card input is first the flight maneuver or "run" cards, then the sensor mnemonic information. Formats for these card images is found in figure 35. The program first reads a list of flight maneuver numbers from the run cards and continues to read this information until the characters "RUN" do not appear in columns 1-3 of the input card. The program then assumes that such a data card contains the sensor information and processes the card image under that assumption. The program continues to process data and calculate the harmonic coefficients and phase angles for each defined parameters for the given set of flight maneuvers. The program will terminate execution after the final input sensor has been processed.

UPCGEN

The Universal Parameter Catalogue Generator or UPCGEN program maintains the Universal Parameter Catalogue (UPC) File and provides a sorted listing of the file. This file is a catalogue of all known sensor mnemonics with their pre-processing codes and contains the column and line titles and a data precision code for the respective sensors. This information is used in the various reports generated by the software in the EASE System.

The UPC file is a random file and resides permanently in the computer system for access by the EASE software system. The entries to the UPC file are arranged alphabetically by sensor mnemonic and pre-processing code. Each record with the file is 280 words in length. The first record within the UPC file is a table of contents which contains a list of the mnemonic names and pre-processing codes of the first entry of each successive 280 word record of the UPC file. This record is uniquely defined with the characters "UPC" in the first word of the record. The second word contains the total number of entries in the UPC file. Each successive two-word entries contain the mnemonic name and pre-processing code of the first entry of each data record in the file. A maximum of 139 entries are possible in this table of contents record.

Each 280-word record in the UPC file contains catalogue information for 20 UPC entries. Hence, the UPC file can grow to a maximum of 2760 entries (20 words per record, with a maximum of 139 records). Each catalogue entry requires 14 data words. A new catalogue entry begins at word number 14*(i-1)+1 where i is the catalogue entry number within a record (i=1,20). A description of the 14 data words for each catalogue entry is found in figure 36.

The UPCGEN program maintains the UPCFILE by allowing addition and deletion of catalogue entries through card input. A set of command cards are used to indicate to the program what type of processing should be performed on the catalogue entries being read. The allowable commands by the UPCGEN program are: "INITIA," "DELETE," "INSERT," "ROLLEM." When the UPCGEN program encounters an input data card containing these words in columns 1-6, the appropriate program logic to perform these commands is set into operation. The catalogue entries following such a command will be processed according to the logic set into operation by that command. The command "INITIA" is used to initialize the UPC file. It assumes no UPC file has been initiated previously so that no entries exist on a UPC file. The program then processes the input catalogue entries and begins to build the UPC file. Each entry is read into a scratch file for sorting before placing the entries into the permanent UPC file. The sorting and building of the UPC file is not performed until the command card "ROLLEM" is encountered in the input deck. This command triggers the logic that assumes all data entries have been placed on the temporary scratch file and the sorting and building procedures can now begin.

The commands "DELETE" and "INSERT" are used to modify an existing UPC file. The "DELETE" command is used to eliminate an entry from the UPC file where the "INSERT" places a new entry, in its proper sorted sequence, onto the UPC file.

Again, these two commands only indicate to the program how the data following these command cards are to be processed. The command "ROLLEM" must be included before the actual deletions and insertions can take place. Catalogue entry modifications are combinations of deletions and insertions. The present entry is commanded to be deleted followed by the corrected entry commanded to be inserted. For any type of processing to be performed on the UPC file, the command card "ROLLEM" is the last card of the input deck.

The order of input for the various modes of processing defined by the UPCGEN program is as follows:

1. to initialize a UPC file
   "INITIA"
   Catalogue entries (one entry per card)
   "ROLLEM";
2. to add new entries to an existing UPC file
   "INSERT"
   Catalogue entries (one entry per card)
   "ROLLEM";
3. to delete existing UPC file data
   "DELETE"
   Catalogue entries (one entry per card)
   "ROLLEM";

4. and to modify existing entries
"DELETE"
Catalogue entries (one entry per card)
"INSERT"
Catalogue entries (one entry per card)
"ROLLEM".

The card image format for a catalogue entry is described in figure 37.

The UPCGEN program provides a list of the current UPC file after the requested processing has been completed. The file list is provided as a sorted list by sensor mnemonic and pre-processing code and list sorted first by category then by sensor mnemonic and pre-processing code. Each sensor is assigned a two-character category code which allows the entries to be defined into a subset of sensors with similar characteristics. Figures 38 and 39 are sample printouts of the two lists provided by this program.

TPDMP AND TABLE PRINT

There are two programs written for the EASE system to verify data being processed. One program, TPDMP, provides a selected listing of the contents of the digitized data tape as received for initial processing. The other program, TABLE PRINT, provides a listing of the current contents of a flight's Table File.

The TPDMP program reads a 9-track digital tape that is configured in the Televent II (see Appendix B) format and provides a listing of the specified data words within a frame of data for either a given time interval or a given number of data frames. The TPDMP program determines what portion of the input tape is to be listed from information supplied through input cards. Figure 40 contains the card format required by this program and figure 41 is a sample of the output provided. The data values are listed as octal numbers.

The TABLE-PRINT program requires no card input for execution. Each of the tables defined in the Table File (see "SETUP") are listed with their current contents. This program is useful particularly for those data elements generated by the SETUP, SCAN, and ABTASK programs. Figures 42a through 42e are samples of the output listing provided by the TABLE-PRINT program.

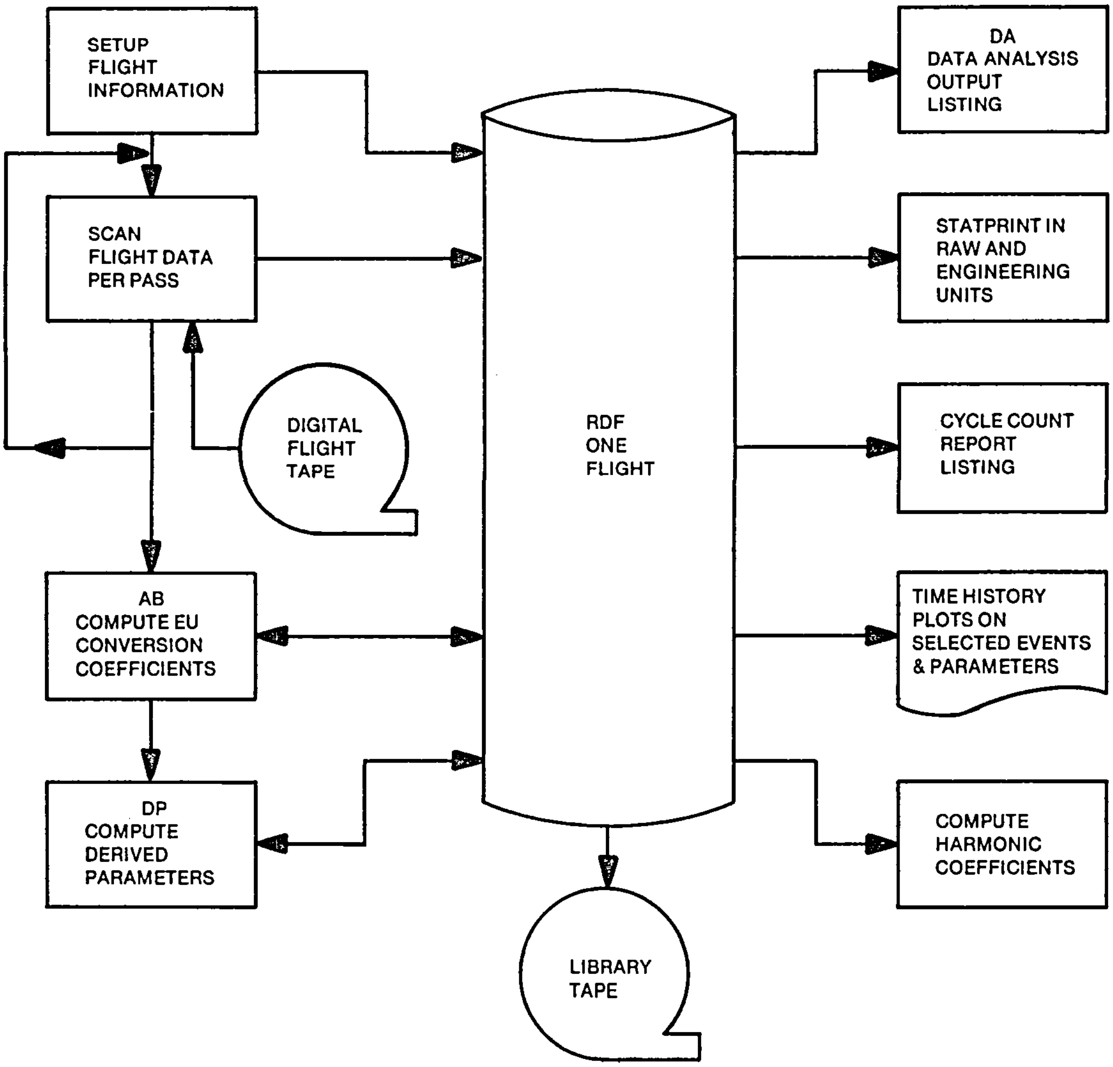


Figure 1 - EASE digital processing scheme.

RECORD NO.	TABLE FILE CONTENTS
1	Header Table
2	Pass 1 Table
3	Pass 2 Table
21	Pass 20 Table
22	Sensor Table for Pass 1
23	Sensor Table for Pass 2
41	Sensor Table for Pass 20
42	Calrun Table
43	Event Table

Figure 2 - Table file record layout

CARD NO.	COLS.	CARD FIELD CONTENTS
1	1-2	Data value "L1"
	6-10	Data value "SETUP"
	21-30	Data value "INITIALIZE" when program is to run in the initialize mode.
		Data value "UPDATE" when program is to run in the update mode. (left justified)
Last	1-2	Data value "L1"
	6-14	Data value "END SETUP"

NOTE: These two cards must be the first and last data cards of the input data cards.

Figure 3 - Description of card image for "L1" level input.

VARIABLE NAME	DESCRIPTION	WORD(S) NO.	DATA TYPE
MAXPAS	Maximum pass number used	1	I
MEVENT	Number of flight maneuvers	2	I
	Not used	3	
FTIME	Beginning time of flight (secs)	4	F.P.
FLIGHT	Flight identification	5	C
DATE	Date of Flight (DDMMYY)	6	C
TAPENO	Aircraft tape number	7	C
AIRCR	Aircraft name	8, 9	C
PILOT	Pilot name	10, 11	C
COPIL	Copilot name	12, 13	C
OBSERV	Observer Name	14, 15	C
ETP	Engineering test plan	16	C
ESGW	Estimated gross weight	17	F.P.
KPR	Airspeed probe recovery factor	19	F.P.
HPC	T-58 engine chaff (horsepower at 100%)	20	F.P.
MRC	Main rotor rpm at 100%	21	F.P.
R	Main rotor radius in feet	22	F.P.
RT	Tail rotor radius in feet	23	F.P.
GT	Tail rotor to main rotor rpm ratio	24	F.P.
GTS	Tail rotor shaft-rpm	25	F.P.
RELHU	Relative humidity	26	F.P.
RFBOOM	Not used	27	F.P.

Figure 4 - Header table data description.

CARD NO.	COLS.	CARD FIELD CONTENTS
1	1-2	Data value of "L2"
	6-11	Data value of "HEADER"
2	1-2	Data value of "HT"
	4	Data value of "1"
	7-8	MAXPAS (right-justified)
	11-13	MEVENT (right-justified)
	21-29	FTIME (HHMMSS.SS)
3	1-2	Data value of "HT"
	4	Data value of "2"
	6-11	FLIGHT (left-justified)
	12-17	DATE (left-justified)
	24-29	TAPENO (left-justified)
	31-42	AIRCR (left-justified)
	44-55	PILOT (left-justified)
	56-67	COPIL (left-justified)
	68-79	OBSERV (left-justified)
4	1-2	Data value of "HT"
	4	Data value of "3"
	5-10	ETP (left-justified)
	11-20	ESGW
	21-30	ESCG
	31-40	KPR
	41-50	HPC
	51-60	MRC
	61-70	R
5	1-2	Data value of "HT"
	4	Data value of "4"
	11-20	RT
	21-30	GT
	31-40	GTS
	41-50	RELHU

Figure 5 - Header table card image description.

VARIABLE NAME	DESCRIPTION	WORD(S) NO.	DATA TYPE
MAXPAR	Actual number of parameters defined for this pass	1	I
SRDIN	Sample rate of input data and type of sampling rate NOTE: Sample rate type is a 1-character code The acceptable codes are: X = Sampling rate is based on time through the PCM data stream M = Sampling rate is based on the main rotor revolution rate through the FM data stream T = Sampling rate is based on the tail rotor revolution rate through the FM data stream Blank = Sampling rate is based on time through the FM data stream	2	C
SRDATA	Sample rate of EASE data file and type of sampling rate (see Note above for sample rate type codes)	3	C
ISTATC	Status code for data availability Code = 0, data for this pass is not available in the raw data file = 1, data for this pass is available in the raw data file (RDF) = 2, data for this pass is available in the RDF for output in engineering units.	4	I
IADDR	Array containing the record address at which the data for a given maneuver is first found for this pass. The entries for this array are indexed by the assigned number for each flight maneuver. The number of entries for this array is one more than the number of flight maneuvers. The last entry to this array is the next available address for data on the RDF for this pass.	5-105	I

Figure 6 - Pass table data description.

VARIABLE NAME	DESCRIPTION	WORD NO.* 13 (j-1)+	DATA TYPE
SNAM	Mnemonic name of sensor	1, 2	C
PPC	Pre-processing code. Valid entries are "D" for direct processing,"V" for FM vibratory, "S" for FM steady, "A" for amplitude, and "P" for phase components of processed data.	3	C
IPARIN	Assigned word location within data frame formatted by the TELEVENT II system (see Appendix B)	4	I
IPARDF	Assigned word location of data transferred to the raw data file (RDF)	5	I
CALTYP	Type of calibration algorithm to be applied in ABTASK routine. Valid entries are: Z, X, P, 1, 2 (see ABTASK section of documentation)	6	C
EUCV1	Engineering unit value for sensor's full deflection (used in ABTASK algorithms)	7	F.P.
EUCV2	Engineering unit value for sensor's zero offset (used in ABTASK algorithms)	8	F.P.
A	A - term for sensor as computed by ABTASK	9	F.P.
B	B - term for sensor as computed by ABTASK	10	F.P.
CALADR	Calrun table entry index (automatically inserted when calrun table entries are processed)	11	I
	Not used	12	
CONLVL	Level of concern (applicable to FM vibratory components only and used for cycle counting)	13	F.P.

* Each sensor table record is 520 words in length, with 13 words per sensor. The data description for the "jth" sensor is found in words 13j-12 through 13j of this record.

Figure 7 - Sensor table data description.

CARD NO.	COLS.	CARD FILED CONTENTS
1	1-2	Data value of "L2"
	6-9	Data value of "PASS" (left-justified)
	20-21	Assigned pass number (right-justified)
	25-26	Numerical value of SRDIN (right-justified)
	27	Type code of SRDIN
	32-33	Numerical value of SRDATA (right-justified)
	34	Type code of SRDATA
2-41	1-2	Data value of "ST"
(one per	3-10	SNAM (left-justified)
sensor)	12	PPC
	14-16	IPARIN (right-justified)
	18-20	IPARDF (right-justified)
	22-23	FM track number assigned (right-justified)
	25-26	FM channel number assigned (right-justified)
	28	CALTYP
	30-39	EUCV1
	40-49	EUCV2
	50-55	Analog tape designation (left-justified)
	56-65	CONLVL

Figure 8 - Description of card image for pass & sensor tables.

VARIABLE NAME	DESCRIPTION	WORD(S) NO.* 6 (i-1)+	DATA TYPE
ICE(1)	Pre-Z or Pre-H calibration run number	1	I
ICE(2)	Pre-R or Pre-L calibration run number	2	I
ICE(3)	Pre-X calibration run number	3	I
ICE(4)	Post-Z or Post-H calibration run number	4	I
ICE(5)	Post-R or Post-L calibration run number	5	I
ICE(6)	Post-X calibration run number	6	I

* Each calrun table record is 120 words in length, with six words per entry group. The entry group for the "ith" calrun table is found in words 6i-5 through 6i.

Figure 9 - Calrun table data description.

CARD NO.	COLS	CARD FIELD CONTENTS
1	1-2	Data value of "L2"
	6-11	Data value of "CALRUN"
2	1-2	Data value of "CT"
	6-8	ICE(1) (right-justified)
	11-13	ICE(2) (right-justified)
	16-18	ICE(3) (right-justified)
	21-23	ICE(4) (right-justified)
	26-28	ICE(5) (right-justified)
	31-33	ICE(6) (right-justified)
3	1-2	Data value of "CS"
	4-5	Data value of "AC"

NOTE: Cards 2 & 3 describe the calrun entry for the "Auto Cals.". This group of calruns will be applied to all non-AB1 and non-AB2 type of cals for each sensor. The remaining groups of calrun entries will modify the calrun entries for those sensors specified on the "CS" cards. The order of input for additional entries to the calrun table is one "CT" card followed by as many "CS" cards that are required to include the sensors to be referenced.

CARD NO.	COLS	CARD FIELD CONTENTS
TYPE "CT"		Card image is the same as defined for card 2 above
TYPE "CS"	1-2	Data value of "CS"
	4-11 13-20 22-29 31-38 40-47 49-56 58-65 67-74	Each entry contains the sensor name, "SNAM" as entered in the sensor table (left-justified). Note: A blank field terminates a grouping of sensors referencing a calrun entry. A "CS" card with each field containing a sensor name will check the card code to see if more sensors are given. ("CS" or a new entry "CT").

Figure 10 - Calrun table card image description.

VARIABLE NAME	DESCRIPTION	WORD(S) NO.* 6 (i-1) +	DATA TYPE
EVENT	Event name.	1,2,3,4	C
ETYPE	Event type code. Allowable codes are: Z = zero calibration, R = full load calibration, X = transducer cal event, T = transient type of data maneuver, S = steady type of data maneuver, I = ignore-numbered event contains no applicable data.	5	C
ETIME	Beginning time of event (seconds).	6	F.P.

* Event table record contains 600 words for a maximum of 100 events. Each event uses six words of the record. The "ith" numbered event is found in words 6i-5 through 6i.

Figure 11 - Event table data description.

CARD NO.	COLS	CARD FIELD CONTENTS
1	1-2	Data value of "L2"
	6-10	Data value of "EVENT"
2 thru run no. plus 1	1-2	Data value of "ET"
	5-7	Run number assigned to maneuver (right-justified)
	12-34	EVENT (left-justified)
	38	ETYPE
	41-49	ETIME (HHMMSS.SS) - not a required input. SCAN task calculates this data entry to be the minimum data frame time tag of the first frame of data of each event for each PCM pass. Card input is provided to correct possible erroneous time data values contained on the digitized input tape.

Figure 12 - Event table card image description.

CARD NO.	COLS.	CARD FIELD CONTENTS
"SD"	1-2	Data value "SD".
	9-10	Pass number (corresponds to pass number defined in the SETUP program) (right-justified).
	14-15	File code of input data tape (right-justified) (corresponds to the file code designation on the job control card-see Appendix A).
	16-20	File number for this pass on input data tape (right-justified)
	25	Time option-applicable for PCM data only = 0 use time recorded during flight which is contained in the data frame; = 1 use time tag furnished when data was digitized.
	28-30	Total number of data bursts to be processed for this pass (right-justified).
	31-35	Code to signify pass to be scanned contains data requiring tracking filter analysis (right-justified) = 0 no tracking filter analysis for this pass; = 1 tracking filter analysis to be performed.

Figure 13 - Card format for "SD" cards of SCAN.

CARD NO.	COLS.	CARD FIELD CONTENTS
"BT"	1-2	Data value "BT".
	6-10	Burst number (right-justified).
	15	"ICOND", Condition Code ICOND = 0, process burst by run tone ("BT" card not required for burst processed for ICON=0); ICOND = 1, skip burst of data defined by run tone signal; ICOND = 2, process burst of data by time interval specified on this card; ICOND = 3, burst of data not on data tape for defined maneuver. Process a "dummy" record of data & statistics for this maneuver.
	21-32	Beginning time of burst (required for ICOND = 2) Format is HH-MM-SS.SSS.
	34-45	Ending time of burst (required for ICOND = 2) Format is HH-MM-SS.SSS.

Figure 14 - Card format for "BT" cards for SCAN.

CARD NO.	COLS.	CARD FIELD CONTENTS
"L1"	1-2	Data value "L1". Must be present as last card of input data deck.

Figure 15 - Card format for "L1" card for SCAN.

RSRA 545-49 FLT 3 72G3

	START		STØP	
	ØB	GS	ØB	GS
01	900-00-00-00,013	095-20-39-52,888	900-00-00-00,851	095-20-40-19,709
02	900-00-00-00,988	095-20-40-23,831	900-00-00-00,688	095-20-40-51,514
03	900-00-00-00,201	095-20-40-55,616	900-00-00-00,238	095-20-41-23,636
04	900-00-00-00,213	095-20-41-27,700	900-00-00-00,826	095-20-41-55,296
05	900-00-00-00,938	095-20-41-59,357	900-00-00-00,376	095-20-42-29,779
06	900-00-00-00,922	095-20-43-45,591	900-00-00-00,847	095-20-44-14,500
07	900-00-00-00,297	095-20-44-18,359	900-00-00-00,422	095-20-44-49,466
08	900-00-00-00,910	095-20-44-53,329	900-00-00-00,910	095-20-45-21,318
09	900-00-00-00,097	095-20-45-25,163	900-00-00-00,635	095-20-45-53,690
10	900-00-00-00,085	095-20-45-57,558	900-00-00-00,847	095-20-46-24,312
11	900-00-00-00,772	095-20-46-28,167	900-00-00-00,047	095-20-47-01,433
12	900-00-00-00,217	095-20-47-05,592	900-00-00-00,104	095-20-47-12,477
13	901-00-00-00,742	095-20-47-16,586	901-00-00-00,804	095-20-47-23,646
14	903-00-00-00,004	095-20-47-27,761	903-00-00-00,754	095-20-47-33,510
15	900-00-00-00,742	095-20-47-37,640	900-00-00-00,992	095-20-47-45,888
16	901-00-00-00,967	095-20-47-50,002	901-00-00-00,192	095-20-47-56,226
17	903-00-00-00,617	095-20-48-00,339	903-00-00-00,929	095-20-48-07,650
18	900-00-00-00,979	095-20-48-11,798	900-00-00-00,079	095-20-48-19,895
19	900-00-00-00,329	095-20-48-24,013	900-00-00-00,054	095-20-48-33,735
20	900-00-00-00,017	095-20-48-37,871	900-00-00-00,617	095-20-48-47,468
21	900-00-00-00,104	095-20-48-51,586	900-00-00-00,242	095-20-48-59,720
22	900-00-00-00,935	095-20-49-03,935	900-00-00-00,448	095-20-49-09,445
23	900-00-00-00,585	095-20-49-13,587	900-00-00-00,448	095-20-49-23,446
24	900-00-00-00,448	095-20-49-27,560	900-00-00-00,523	095-20-49-30,634
25	900-00-00-00,223	095-20-49-34,761	900-00-00-00,060	095-20-49-37,597
26	900-00-00-00,323	095-20-49-41,734	900-00-00-00,810	095-20-49-46,219
27	000-00-16-17,455	095-20-49-50,546	000-00-16-24,005	095-20-49-57,094
28	000-00-16-25,555	095-20-49-58,643	000-00-16-30,180	095-20-50-03,267
29	000-00-31-57,216	095-20-50-07,527	000-00-32-07,028	095-20-50-17,338
30	000-00-32-26,603	095-20-50-21,530	000-00-32-36,916	095-20-50-31,841
31	000-00-32-49,703	095-20-50-36,018	000-00-32-57,678	095-20-50-43,992
32	000-00-42-59,928	095-20-50-48,149	000-00-43-38,741	095-20-51-26,958
33	000-00-53-28,278	095-20-51-31,070	000-00-54-59,803	095-20-53-02,592
34	900-01-35-04,441	095-20-53-06,372	900-01-35-09,741	095-20-53-11,673
LØS	900-01-35-23,566	095-20-53-19,967	900-01-35-23,616	095-20-53-20,017
LØS	900-01-35-23,616	095-20-53-20,017	900-01-35-23,653	095-20-53-20,054
LØS	900-01-35-23,666	095-20-53-20,067	900-01-35-23,728	095-20-53-20,117
LØS	900-01-35-23,728	095-20-53-20,117	900-01-35-23,766	095-20-53-20,154
LØS	900-01-35-23,766	095-20-53-20,154	900-01-35-38,"58	095-20-53-20,242
LØS	900-01-35-23,903	095-20-53-20,292	900-01-35-23,966	095-20-53-20,354

Figure 16 - Burst interval listing.

FLIGHT LOG

PAGE 1 OF 1

F.T.P. 72G-2	G.W. 19418	C.G. 302.18	FLT. DATE 9/19/77	FLT. NO. 1
FUEL TAKE OFF 990/ 740 LANDING 1190/ 900	TAPE NO. 545-41A&B		FLT. TIME 0.0	
WIND 250 / 6	OAT 26°C	hp +110	WEATHER CAVU	
A/C NO. 545	PILOT Reine	CO-PILOT Graham	OBSERVER	

TEST TYPE Taxi runs w/compound tail

A/C CONFIGURATION Compound tail installed (Lower stab fixed @ + 2°) + Helo

RUN NO.		REGIME			
1/35	X	X Heading = 180°			
2/36	R	R Pads 1 Pitch = +2.0 Roll = +0.4			
3/37	Z	Z			
4/38	X	X			
5/39	R	R Pads 2			
6/40	Z	Z			
7/31	X	X = Lead stop, sideslip = +20°			
8/32		Lag stop, sideslip = -20°			
9/33		T.R. OTBD, S.L. = 0, α=+20°			
10/34		T.R. INBD, S.L. =100, α=-20°			
11/27		LT-LT-LO-FWD			
12/28		LT-LT-HI-FWD			
13/29		RT-RT-LO-AFT			
14/30	↓	RT-RT-HI-AFT NR			
15	S	GRD. Run - Flat Pitch ∿ 94%			
16		96%			
17		98%			
18		100%			
19		102%			
20		104%			
21		106%			
22		↓ ↓ 108%			
23		TAXI - 20 KIAS ∿ 104%			
24		30			
25		40			
26	↓	↓ 50 ↓ ↓			

SA 7861 REV. A

Figure 17 - Flight log.

SUBROUTINE RDFRD (ITOS, ITYPE, ICONV, IRUN, NPAR, NAMES, PPC, SR, ISMPL, LSMPL, OUTBUF, ISZBUF, TOS, NFOUT, MSG, IER)

ARG	DEFINITION
1. ITOS	= 0, when initial call with given set of param spec's which must be used to search sensor tables. = 1, when previous call with same set of param spec's has been used to define table of subscripts argument (TOS) thus averting search of sensor table.
2. ITYPE	= 0, when data samples are being requested. = 1, when statistics are being requested.
3. ICONV	= 0, if results are to be in raw units. = 1, if results are to be in eng. units.
4. IRUN	Run (event) number requested.
5. NPAR	Number of parameters requested.
6. NAMES	Array of parameter mnemonics (NPAR in length).
7. PPC	Array of parameter preprocess codes (NPAR in length).
8. SR	Array of parameter sample rates (NPAR in length). If data is being requested, all must be same.
9. ISMPL	Beginning sample number requested, if data.
10. LSMPL	Ending sample number requested, if data.
11. OUTBUF	Output buffer for data or statistics
12. ISZBUF	Size of OUTBUF: if asking for stats, size should be (10xNPAR) if asking for data, size should be (LSMPL-ISMPL+1)x(NPAR).
13. TOS	Table of subscripts, four words per parameter built within routine when ITOS = 0 for possible subsequent calls with same set of parameters. Used by routine, thus averting search of sensor table; when ITOS = 1, the four words are: TOS (1) = PASS number TOS (2) = ELEMENT number TOS (3) = A TOS (4) = B } when parameter is available. Zeros are inserted in all four words when parameter is not available.
14. NFOUT	Output array of actual number of frames returned for each parameter (NPAR in length). For stats, NFOUT will equal 10.
15. MSG	Array of output parameter messages (NPAR in length). = 0, no errors = 1, less data returned than requested because ending sample requested is greater than last sample available. = 2, beginning sample requested is greater than last sample available.
16. IER	Output error code determined as the sum of the following individual codes: = 0 No errors, = 1 CAL run requested, = 2 Parameter level errors exist, = 4 No such run, = 8 Buffer size error.

Figure 18 - Subprogram RDFRD argument list.

/C545 R FLIGHT FLT=5 PASS 9 DATE 041078

CALIBRATION RUN TABLE (CRNTAB) SUMMARY

ENTRY	PRE=Z/LO	PRE=R/HI	PRE=XDUC	POST=Z/LO	POST=R/HI	POST=XDUC
1	4	5	4	40	41	40
2	3	2	1	39	38	37
3	1	2	1	37	38	37
4	1	3	1	37	39	37
6	12	13	0	30	31	0
7	13	12	0	31	30	0
8	3	2	11	39	38	29
9	11	14	0	29	32	0

Figure 19 - ABTASK calibration run table summary.

/C545 R FLIGHT FLT=[illegible] PASS 9 DATE 041078

PARAMETER		******* AVERAGE RAW UNITS ********						**** PERCENT FULL SCALE *****									** STD. DEVIATION (RAW UNITS) **					
		PRE - FLIGHT			POST - FLIGHT			PRE - FLIGHT			POST - FLIGHT			SHIFT			PRE - FLIGHT			POST - FLIGHT		
		Z/LO	R/HI	X	Z/LO	R/HI	X	Z/LO	R/HI	X	Z/LO	R/HI	X	Z/LO	R/HI	X	Z/LO	R/HI	X	Z/LO	R/HI	X
HEAD180	D	255	374	378	256	374	329	0	46	48	0	46	28	0	0	-20	0.	0.1	0.1	0.	0.1	0.1
NR	D	257	506	8	257	506	8	0	97	-96	0	97	-96	0	0	0	0.	0.	0.	0.0	0.	0.
NRQ1	D	257	407	557	257	406	257	0	59	0	0	58	0	0	-1	0	5.1	5.1	5.1	4.6	4.7	4.6
PITCHATT	D	256	363	564	256	363	265	0	42	3	0	42	3	0	0	0	0.3	0.2	0.8	0.3	0.1	0.6
ROLLATT	D	257	382	556	257	382	258	0	49	0	0	49	0	0	0	0	0.1	0.5	1.1	0.2	0.5	1.1
WIACLTUP	D	258	385	558	258	385	258	1	50	1	1	50	1	0	0	0	1.3	1.4	1.3	0.5	0.3	0.5
WIACLTLO	D	257	383	557	255	381	255	0	49	0	0	49	0	0	0	0	0.8	0.7	0.8	0.7	0.7	0.7
WIACRTUP	D	257	379	557	257	379	257	0	48	0	0	48	0	0	0	0	2.4	2.6	2.4	1.5	1.4	1.6
PITCHRAT	D	257	505	556	256	506	257	0	97	0	0	97	0	0	0	0	0.	0.5	0.1	0.1	0.5	0.
ROLLRAT	D	257	506	558	257	506	259	0	98	0	0	98	1	0	0	1	0.	0.1	0.5	0.0	0.2	0.
WIACRTLO	D	256	382	556	255	382	255	0	49	0	0	49	0	0	0	0	1.5	1.7	1.5	1.6	1.6	1.6
TRIMPIT	D	104	388	511	104	388	511	-59	51	0	-59	51	0	0	0	0	0.	0.	99.9	0.2	0.4	99.9
YAWRAT	D	255	505	556	255	505	256	0	97	0	0	97	0	0	0	0	0.1	0.2	0.4	0.2	0.3	0.3
VIPBOOM	D	8	508	8	7	508	7	-96	98	-96	-96	98	-96	0	0	0	0.	0.1	0.	0.4	0.	0.4
IW	D	256	379	139	256	379	139	0	48	-45	0	48	-45	0	0	0	0.1	0.1	0.3	0.	0.2	0.1
LGMIXIR	D	107	393	511	107	394	511	-58	54	0	-58	54	0	0	0	0	0.	0.	99.9	0.1	0.1	99.9
LATMIXIP	D	166	421	511	166	421	511	-34	64	0	-34	64	0	0	0	0	0.2	0.	99.9	0.5	0.	99.9
COLMIXIP	D	266	488	511	270	488	511	3	90	0	5	90	0	2	0	0	0.1	0.	99.9	0.5	0.	99.9
LATSTKP	D	49	357	511	48	358	511	-80	39	0	-80	40	0	0	1	0	0.2	0.2	99.9	0.2	0.5	99.9
LGSTKP	D	64	334	611	62	334	511	-74	30	0	-75	30	0	-1	0	0	0.2	0.1	99.9	0.3	0.4	99.9
COLLSTKP	D	11	446	511	12	447	511	-95	74	0	-94	74	0	1	0	0	0.5	0.4	99.9	0.2	0.6	99.9
PEDP	D	329	164	511	329	163	511	28	-35	0	28	-36	0	0	-1	0	0.1	0.	99.9	0.	0.	99.9
PITCHACC	D	256	506	556	256	506	256	0	98	0	0	98	0	0	0	0	0.3	0.3	0.5	0.2	0.3	0.5
ROLLACC	D	256	506	556	256	506	256	0	98	0	0	98	0	0	0	0	0.2	0.3	0.3	0.2	0.3	0.5
YAWACC	D	256	506	556	256	506	256	0	98	0	0	97	0	0	-1	0	0.2	0.3	0.4	0.2	0.4	0.5
LOADFACT	D	256	506	554	256	506	254	0	97	0	0	97	0	0	0	0	0.4	0.4	0.5	0.4	0.5	0.5
NO1QPCT	D	260	399	560	260	399	260	1	55	1	1	56	1	0	1	0	2.5	2.1	2.5	2.5	2.1	2.5
NO2QPCT	D	260	398	560	260	398	260	1	55	1	1	55	1	0	0	0	0.9	2.1	0.9	1.0	2.0	1.0
NO1NFPCT	D	257	509	4	256	509	4	0	98	-98	0	98	-98	0	0	0	0.	0.2	0.4	0.1	0.0	0.4
NO2NFPCT	D	260	509	11	260	509	11	1	98	-95	1	98	-95	0	0	0	0.	0.1	0.	0.	0.0	0.
AILPOSR	D	338	13	511	338	17	511	32	-94	0	32	-93	0	0	1	0	0.1	0.	99.9	0.1	0.	99.9
FLAPPOSR	D	69	402	511	69	401	511	-72	57	0	-72	56	0	0	-1	0	0.5	0.	99.9	0.2	0.5	99.9
HBCOM	D	258	456	557	258	456	256	0	78	0	0	78	0	0	0	0	0.3	0.5	0.5	0.5	0.5	0.6
ITATBOOM	D	257	382	599	257	383	301	0	49	16	0	49	17	0	0	1	0.3	0.3	0.5	1.5	1.4	2.4

Figure 20 - ABTASK raw units statistic summary.

/C545 R FLIGHT ELT-5 PASS 9 DATE 041078

PARAMETER		CRNTAB,ENT	CAL-TYPE	EUCV1	EUCV2	A TERM	B TERM
HEAD180	D	2	ACZ	71,3000002	180.0000000	0.59921891E 00	0.26599960E 02
NR	D	2	ACZ	61,0000000	61.0000000	0.24497992E 00	-0.19598394E 01
MRQ1	D	3	ACX	45600,0000000	0.	0.30438403E 03	-0.78306273E 05
PITCHATT	D	2	ACX	24,8000000	3.0000000	0.23161020E 00	-0.58348392E 02
ROLLATT	D	2	ACX	46,0999999	0.5000000	0.36780883E 00	-0.94196186E 02
WIACLTUP	D	1	ACX	1616,0000000	0.	0.12778579E 02	-0.33054675E 04
WIACLTLO	D	1	ACX	1632,0000000	0.	0.12920761E 02	-0.33105260E 04
WIACRTUP	D	1	ACX	1508,0000000	0.	0.12320266E 02	-0.31720582E 04
PITCHRAT	D	2	ACX	75,3000002	0.	0.30186559E 00	-0.77578133E 02
ROLLRAT	D	2	ACX	146,7000008	0.	0.58687822E 00	-0.15182900E 03
WIACRTLO	D	1	ACX	1649,0000000	0.	0.13034498E 02	-0.33355501E 04
TRIMPIT	D	7	PC	-6,2000000	25.7000000	0.11228718E 00	-0.17680419E 02
YAWRAT	D	2	ACX	74,8999996	0.	0.30066699E 00	-0.77022924E 02
VIPBOOM	D	4	ACX	0,6410700	0.0100000	0.12818122E-02	-0.96593265E-04
IW	D	8	ACX	-11,7000000	10.0000000	-0.95097376E-01	0.23309844E 02
LGMIXIP	D	6	PC	0,	100.0000000	0.34723268E 00	-0.37155332E 02
LATMIXIP	D	6	PC	0,	100.0000000	0.39331694E 00	-0.65586433E 02
COLMIXIP	D	7	PC	0,	100.0000000	0.45529998E 00	-0.12218639E 03
LATSTKP	D	6	PC	0,	100.0000000	0.32385986E 00	-0.15654429E 02
LGSTKP	D	6	PC	0,	100.0000000	0.37014018E 00	-0.23664537E 02
COLLSTKP	D	7	PC	0,	100.0000000	0.22988750E 00	-0.28003789E 01
PEDP	D	9	PC	0,	100.0000000	-0.60421578E 00	0.19878928E 03
PITCHACC	D	2	ACX	152,2000008	0.	0.60886704E 00	-0.15631570E 03
ROLLACC	D	2	ACX	307,7000008	0.	0.12310259E 01	-0.31622976E 03
YAWACC	D	2	ACX	156,2500000	0.	0.62526488E 00	-0.16057217E 03
LOADFACT	D	2	ACX	-2,0700000	1.0000000	-0.82843084E-02	0.31104607E 01
NO1QPCT	D	3	ACX	72,8000002	0.	0.52258934E 00	-0.13597577E 03
NO2QPCT	D	3	ACX	74,0100002	0.	0.53688072E 00	-0.13994077E 03
NO1NFPCT	D	2	ACZ	62,9770002	62.0310001	0.24988571E 00	-0.21869678E 01
NO2NFPCT	D	2	ACZ	61,5999999	62.3000002	0.24738956E 00	-0.20212851E 01
AILPOSR	D	6	PC	383,0000000	63.0000000	0.98767632E 00	0.48184855E 02
FLAPPOSR	D	6	PC	0,	22.0000000	0.66263983E-01	-0.46209528E 01
HBOOM	D	2	ACZ	4041,0000000	-41.0000000	0.20377895E 02	-0.53017223E 04
ITATBOOM	D	2	ACZ	29,7400000	0.2200000	0.23620864E 00	-0.60536559E 02

Figure 21 - ABTASK engineering unit summary.

```
58&43  72G-3   FLT-5
STATISTICS PRINTOUT          RAW UNITS     PASS=   8
TRR2    LVL   1T          TR BLD STRESS P-2  PSI V
```

RUN	MANEUVER	MAXIMUM	MINIMUM	AVERAGE	STD.DEV.	PERCENTILE	N POINTS	ER COUNT	ER TYP
1	XCAL RADS 1	5.000	4.000	4.667	0.4764	5.000	48	0	000000
2	RCAL RADS 1	5.000	4.000	4.694	0.4628	5.000	109	1	000000
3	ZCAL RADS 1	5.000	4.000	4.586	0.4962	5.000	70	0	000000
4	XCAL RADS 2	5.000	4.000	4.696	0.4635	5.000	69	0	000000
5	RCAL RADS 2	4.000	1.000	1.734	0.6422	3.875	94	0	000000
6	ZCAL RADS 2	6.000	4.000	4.640	0.5056	5.000	89	0	000000
7	LEAD STOP.SIDESLIP=+20	5.000	4.000	4.663	0.4748	5.000	104	0	000000
8	LAG STOP,SIDESLIP= +20	5.000	4.000	4.678	0.4699	5.000	90	0	000000
9	TR OUTBD,TF34 MAX,A=+20	6.000	4.000	4.686	0.4914	5.000	86	0	000000
10	TR INBD,TF34 G.I.,A=-20	5.000	4.000	4.712	0.4542	5.000	146	0	000000
11	LT=LT=LO=FWD,FLAPS UP	6.000	4.000	4.674	0.4949	5.000	89	0	000000
12	LT=LT=HI=FWD,FLAPS UP	5.000	4.000	4.718	0.4519	5.000	110	0	000000
13	RT=RT=LO=AFT,FLAPS DWN	5.000	4.000	4.710	0.4576	5.000	62	0	000000
14	RT=RT=HI=AFT,FLAPS DWN	5.000	4.000	4.705	0.4589	5.000	88	0	000000
15	L/O AND CLIMB AT 70 KTS	43.00	3.000	18.82	9.189	31.90	1715	0	000000
16	RT CLIMBING TURN,70 KTS	37.00	16.00	24.67	3.376	31.58	1419	0	000000
17	LF 70 KTS. AUX G.I.	31.00	12.00	18.92	3.427	25.80	299	0	000000
18	LF 80 KTS. AUX G.I.	40.00	19.00	26.18	3.000	32.19	471	0	000000
19	LF 90 KTS. AUX G.I.	40.00	24.00	30.42	2.481	35.55	541	0	000000
20	LF 100 KTS. AUX G.I.	38.00	24.00	28.79	2.410	33.71	492	0	000000
21	RT TURN 80 KTS. 15/30DG	39.00	11.00	20.87	4.382	29.72	1381	0	000000
22	LF 60 KTS. AUX G.I.	30.00	12.00	18.62	2.953	24.73	781	0	000000
23	LT TURN 80KTS. 15/30DEG	36.00	14.00	22.55	3.577	29.98	1192	0	000000
24	LF 80 KTS. GEAR UP	36.00	18.00	23.88	3.114	30.71	494	0	000000
25	RPD 500 FPM. 80KTS. GU	30.00	7.000	14.72	3.182	21.01	987	0	000000
26	CL 500 FPM. 80KTS. GU	44.00	9.000	28.25	7.064	36.67	1333	0	000000
27	RPD 500 FPM.LT/RT TN.GU	33.00	6.000	15.33	3.657	22.43	1111	0	000000
28	APPROACH AND LANDING	32.00	2.000	12.05	4.844	20.85	3708	0	000000
29	LT=LT=LO=FWD. FLAPS UP	5.000	4.000	4.625	0.4885	5.000	56	0	000000
30	LT=LT=HI=FWD. FLAPS UP	5.000	4.000	4.637	0.4838	5.000	80	0	000000
31	RT=RT=LO=AFT. FLAPS DWN	5.000	3.000	4.549	0.5208	5.000	71	0	000000
32	RT=RT=HI=AFT. FLAPS DWN	5.000	4.000	4.503	0.4935	5.000	58	0	000000
33	LEAD STOP.SIDESLIP=+20	5.000	4.000	4.636	0.4855	5.000	55	0	000000
34	LAG STOP, SIDESLIP=+20	5.000	4.000	4.681	0.4695	5.000	72	0	000000
35	TR OUTBD.TF34 MAX/A= 20	5.000	4.000	4.679	0.4699	5.000	87	0	000000
36	TR INBD. TF34 G.I.,A=-20	5.000	4.000	4.633	0.4846	5.000	90	0	000000
37	XCAL RADS 1	5.000	4.000	4.637	0.4834	5.000	91	0	000000
38	RCAL RADS 1	5.000	4.000	4.684	0.4681	5.000	79	0	000000

Figure 22 - Sample output for STATPRINT program.

CARD NO.	COLS.	CARD FIELD CONTENTS
1 → n	1-2	IPASS - Pass number requested (right-justified) - only required if all sensors of one pass are to be tabulated.
	3-10	NAME - Mnemonic name of sensor (left-justified)
	15	PPC - Pre-processing code of sensor
	16-20	SR - Sample rate at which sensor has been processed onto the RDF (right-justified) NOTE: NAME, PPC, SR to be furnished when a specific sensor is to be tabulated
	25	ICONV - = 0 data tabulated in raw units = 1 data tabulated in engineering units

Figure 23 - Description of card image for STATPRINT.

CARD NO.	COLS	CARD FIELD CONTENTS
1	1	NPASS - Number of derived parameter passes to be processed.
	2-3	JPASS(1) - Derived parameter pass number to be processed first (right-justified)
	4-5	KPASS(1) - Pass number assigned to first derived parameter pass (right-justified)
	6-7	JPASS(2) - Derived parameter pass number processed second (right-justified)
	8-9	KPASS(2) - Pass number assigned to second derived parameter pass (right-justified)
	10-11	JPASS(3) - Derived parameter pass number processed third (right-justified)
	12-13	KPASS(3) - Pass number assigned to third derived parameter pass (right-justified)
	14-15	JPASS(4) - Derived parameter pass number processed fourth (right-justified)
	16-17	KPASS(4) - Pass number assigned to fourth derived parameter pass (right-justified)
	18-19	JPASS(5) - Derived parameter pass number processed fifth (right-justified)
	20-21	KPASS(5) - Pass number assigned to fifth derived parameter pass (right-justified)

Figure 24 - Card image for DPTASK program.

(1)
FTP 720-3 FLT. 5, DATE 04-10-78
A/C NO. 545 GROSS WEIGHT 27110 LBS, C.G. 302.1 IN
TEST TYPE FIRST FLIGHT,HANDLING QUALITIES AND STRUCTURAL
A/C CONFIG. FULL COMP.CONF. CPU PITCH.100-50 ROLL 100-100 YAW 100-100

(2) (3) (6) (5)	CALIB A/S BOOM KN	MAIN ROTOR SPEED %	MR BLADE PITCH DEG (V)	MR BLADE PITCH DEG (S)	MR BLADE FLAP DEG (V)	MR BLADE FLAP DEG (S)	MR BLADE MONT DEG (V)	MR BLADE LAG DEG (S)	ALPHA #1 PUSH ROD#1 UPPER LBXV)	ALPHA #1 PUSH ROD#1 UPPER LBXS)	RT LAT STA STAR LD LB(V)	RT LAT STA STAR LD LB(S)	MAIN ROTOR SHAFT TORQ FT-LB	TAIL ROTOR TORQ #1 FT-LB (S)	FREE AIR TEMP BOOM DEG C	DENS ALT BOOM FT	LOAD FACT G
(4)	20X	20X	1M	1M	1M	1M	1M	1M	1M	1M	5M	5M	20X	5M	20X	20X	20X
L/O AND CLIMB AT 70 KTS	69.7	103	4.7	4.3	2.8	-0.4	0.3	2.1	217	-117	30	-290	30153	59	10.2	-601	1.00
	58.4	107	8.4	12.6	0.6	5.4	0.6	11.1	72	40	100	40	51666	349	12.9	-327	1.24
	0.	100	7.3	12.3	4.1	-0.2	0.9	10.5	217	-105	200	10	7834	154	8.1	-902	0.82
RT CLIMBING TURN,70 KTS	80.5	102	8.8	12.1	1.1	4.6	0.8	10.3	205	-93	70	-170	46327	200	8.3	-106	1.01
	85.4	103	7.0	12.6	0.7	5.1	0.5	11.0	169	-44	110	-30	49839	299	10.3	262	1.06
	77.1	101	7.1	12.3	1.2	4.7	0.8	10.6	205	-81	150	-90	43143	231	6.5	-592	0.96
LF 75 KTS, AUX G.I.	82.5	103	5.6	10.3	0.7	4.3	0.6	6.7	181	-51	190	-80	31080	151	7.6	286	1.01
LF 85 KTS, AUX G.I.	72.3	103	5.1	11.2	0.6	4.3	0.8	8.0	181	-29	140	-120	36202	222	7.9	273	1.01
LF 95 KTS, AUX G.I.	100.3	102	5.4	11.5	0.9	4.2	0.8	8.4	193	-20	130	-120	38484	226	8.8	445	1.00
LE 100 KTS, AUX G.I.	195.5	102	5.4	11.9	0.9	4.1	0.8	9.0	205	-20	130	-130	40030	179	9.3	511	1.00
RT TURN 80 KTS, 1573056	88.9	103	5.1	10.8	0.7	4.0	0.5	7.5	217	-69	110	-190	36487	109	9.1	641	1.04
	94.5	104	5.1	11.3	0.3	4.9	0.5	8.7	181	-8	200	-30	41317	231	10.5	824	1.13
	85.7	101	6.8	11.4	0.9	4.7	0.9	8.0	217	-69	230	-40	31381	118	6.8	399	0.96
LE 85 KTS, AUX G.I.	73.2	103	3.2	10.1	1.1	4.2	0.4	6.3	193	-81	180	-140	31283	145	3.9	565	1.00
	80.0	104	3.3	10.5	0.6	4.6	0.6	7.1	151	-44	180	-10	34925	213	13.4	779	1.05
	69.5	102	3.5	10.2	1.3	4.5	0.8	6.7	230	-57	210	-140	27011	163	5.5	215	0.96
LT TURN 80KTS, 1573303G	87.3	105	3.8	11.0	0.6	4.1	0.5	7.3	205	-69	100	-190	36933	104	8.0	577	1.04
	93.5	104	3.6	11.7	0.4	4.8	0.5	8.5	169	-8	200	-30	41012	209	9.6	765	1.13
	84.0	101	6.4	11.2	0.9	4.2	0.8	8.2	205	-57	200	-30	32490	191	5.8	284	0.93
LF 85 KTS, GEAR UP	83.7	103	3.8	10.8	0.5	4.0	0.5	7.3	157	-32	130	-170	33948	177	7.9	1234	1.00
	91.7	103	3.9	11.1	0.5	4.3	0.5	7.8	157	-25	110	-70	36751	218	9.1	1377	1.01
	55.1	102	3.9	11.0	0.6	4.2	0.5	7.6	169	-25	130	-160	30968	195	6.3	1000	0.98
RPB 500 FPM, 85KTS, G3	87.4	103	4.1	8.3	0.4	3.3	0.4	3.9	157	-44	70	-190	22140	86	7.5	1162	1.00
	59.1	105	3.2	9.9	0.3	3.7	0.4	6.2	157	-9	130	-70	30968	168	8.9	1486	1.05
	55.4	103	3.3	9.8	0.5	3.7	0.4	6.2	169	-20	150	-170	17270	91	5.6	786	0.95
CL 500 FPM, 80KTS, GU	58.7	102	4.3	8.7	0.3	3.5	0.4	4.3	169	-57	110	-220	41166	91	7.8	967	1.01
	92.0	105	7.0	12.7	0.7	5.0	0.5	10.9	157	-20	130	-60	49231	336	9.2	1404	1.06
	56.1	108	7.4	12.7	1.1	4.7	0.8	10.1	193	-44	160	-160	19401	172	5.8	616	0.97

Figure 25 - Sample printout for DATASK program.

CARD NO.	COLS.	CARD FIELD CONTENTS
1-4	1-72	Title Cards: Any alphanumeric characters
5-n		Maneuver specification cards (one per run).
	2-6	Run number (right-justified). NOTE: No run numbers specified indicates all non-calibration runs to be processed.
n+1	1-3	Value of "END" required input card even when no maneuver identification cards have been included.
n+2→m		Parameter specification cards.
n+2	1-4	Value of "Page".
n+3 etc.	1-8	Sensor mnemonic (left-justified).
	10	Pre-processing code.
	15-17	Sample rate (right-justified).
		NOTE: Up to 18 parameter identification cards allowed within a page grouping. Repeat page grouping as necessary beginning with data card containing "Page" in columns 1-4.
Last	1-3	Value of "END".

Figure 26 - Card description for DATASK program.

CARD NO.	COLS.	CARD FIELD CONTENTS
1-4	2-72	Title specification cards containing any alphanumeric characters.
		Maneuver specification cards
1→n	(1-5)	Flight maneuver number (right-justified) one entry per card. No maneuver specification cards will default the program to cycle count all non-calibration maneuvers.
n+1	1-3	Value of "END" Note: This card must be present even if no cards containing flight maneuver numbers are present.
n+2→m		Parameter specification cards Type of input for parameter specification determines three modes of operation: Mode = 1, all vibratory components with concern level> 0 to be cycle counted; Mode = 2, all vibratory components with concern level> 0 within stated pass number to be cycle counted; Mode = 3, specified sensor's vibratory component to be cycle counted if concern level> 0.
		Type of input required for each mode: Mode = 1, one blank card.
	4-5	Mode = 2, pass number (right-justified).
		Mode = 3,
	11-18	sensor mnemonic name (left-justified).
	20-25	sample rate (right-justified).

Figure 27 - Card input for cycle count program.

(1)
FTP 72G-3 FLT? 5, DATE 04-10-78
A/C NO. 545 GROSS WEIGHT 27110 LBS, C.G. 302.1 IN

TEST TYPE FIRST FLIGHT, HANDLING QUALITIES AND STRUCTURAL

A/C CONFIG, FULL COMP, GDNF, CPU PITCH, 100-50 ROLL 100-100 YAW 100-100

(2) (3) TR PITCH BEAM LD #5 LB-V SR 1T (5) VIBRATORY CYCLE COUNT CONCERN LEVEL C = 90

MANEUVER	RUN NO.	STEADY (S) MEAN	STEADY (S) MAX	STEADY (S) @MAX (V)	ABSOLUTE LEVELS MAX	ABSOLUTE LEVELS MIN	VIBRATORY (V) @MAX (S)	VIBRATORY (V) 95 PTILE	VIBRATORY (V) MAX	(4) N TOTAL	.75 TO 1.00 C	1.00 TO 1.25 C	1.25 TO 1.50 C	1.50 TO 1.75 C	1.75 TO 2.00 C	2.00 TO 2.25 C	2.25 TO 2.50 C	> 2.50 C
L/O AND CLIMB AT 70 KTS	15	30	59	54	135	-39	66	71	81	1715	196							
RT CLIMBING TURN, 70 KTS	16	37	54	47	123	-39	65	66	76	1419	30							
LF 80 KTS, AUX G.I.	18	31	46	34	110	-42	63	66	71	471	11							
LF 90 KTS, AUX G.I.	19	29	44	41	118	-50	63	71	78	541	82							
LF 100 KTS, AUX G.I.	20	24	38	29	101	-51	60	68	70	492	21							
RT TURN 80 KTS, 15/30DEG	21	23	42	29	105	-42	63	57	71	1381	2							
LT TURN 80KTS, 15/30DEG	23	28	42	29	97	-40	52	61	65	1192								
LF 80 KTS, GEAR UP	24	30	47	31	107	-39	53	63	70	494	1							
CL 500 FPM, 80KTS, GU	26	40	67	54	146	-42	79	79	86	1333	717							
(6) FLIGHT SUMMARY					146	-51					1060							

Figure 28 - Sample printout for cycle count program.

CARD NO.	COLS.	CARD FIELD CONTENTS
1→36	5	Page group format number (integer value 1→9)
	10	Row number (integer 1→4 for rows numbered from top (1) to bottom (4)
	11-18	Alphanumeric information for first line of label.
	29-46	Alphanumeric information for second line of label.
Last	5	Data value of "0". NOTE: Last card must be included in deck setup even if no plot row label cards are included.

Figure 29 - Time history plot row labels card format.

CARD NO.	COLS.	CARD FIELD CONTENTS
2n-1*	1-8	Mnemonic for sensor (left-justified)
	9	Preprocessing code
	12-4	Sample rate (right-justified)
	19-36	18 Character Y-Axis label - first line (See(4), Fig. 33)
	37-54	18 Character Y-Axis label - second line (See(5), Fig. 33)
	55-58	Four character direction note for upper level of scale (See(6), Fig. 33)
	59-62	Four character direction note for lower level of scale (See(7), Fig. 33)
	63-64	Parameter's plot position number for PGF#1 (right-justified)
	65-66	Parameter's plot position number for PGF#2 (right-justified)
	79-80	Parameters plot position number for PGF#9 (right-justified) NOTE: Plot positions are number 1-12 beginning with the top left-most position down to the bottom right-most position.
2n*	1-10	Range of primary scale (floating point).
	11-20	Value at bottom of primary scale (floating point).
	21-30	Range of secondary scale (floating point).
	31-40	Value at bottom of secondary scale (floating point). NOTE: Primary scale will be used unless data exceeds scale by 20%.
	41-42	Scale precision index number for printing scale (right-justified).
	44	Option flag to reference all value to first non-null value. If entry is "1", the first non-null data sample will be subtracted from all subsequent samples. *Above 2-card formats to be repeated for a maximum of 40 parameters. (n=1→40)
Last	1-8	Blank - required card to terminate parameter specification card group.

Figure 30 - Time history parameter specification card format.

CARD NO.	COLS.	CARD FIELD CONTENTS
1→8	5	Heading number (1-8)
	6-13	Eight character information for line 1.
	14-21	Eight character information for line 2.
	22-29	Eight character information for line 3.
Last	5	Data value of "0" - must be included to terminate input of heading label card group even if none have been included in the deck setup.

Figure 31 - Time history heading label card format.

CARD NO.	COLS.	CARD FIELD CONTENTS
2m-1*	1-5	Flight maneuver number to be plotted (right-justified).
	6-10	PGF number to be used for this plot (right-justified).
	11-20	Start time of plot in seconds of elapsed time into maneuver (floating point) (0.0 sec. would designate a start time at the beginning of a maneuver).
	21-30	Elapsed stop time of plot in sec (floating point).
	31-78	Alphanumeric title of maneuver (See(8), Fig. 33).
	80	Option to list data being plotted: = 0 no printout to be produced, = 1 printout of data values produced.
2m*	1-10	Line 4 of heading label for heading #1.
	11-10	Line 4 of heading label for heading #2.
	21-30	Line 4 of heading label for heading #3.
	31-40	Line 4 of heading label for heading #4.
	41-50	Line 4 of heading label for heading #5.
	61-70	Line 4 of heading label for heading #6.
	71-80	Line 4 of heading label for heading #7. (See(9) of Fig. 33) *NOTE: Above 2-card format to be repeated for each plot to be generated
Last	5	Data value of "0" - must be included as last card of this group.

Figure 32 - Time history burst selection card format.

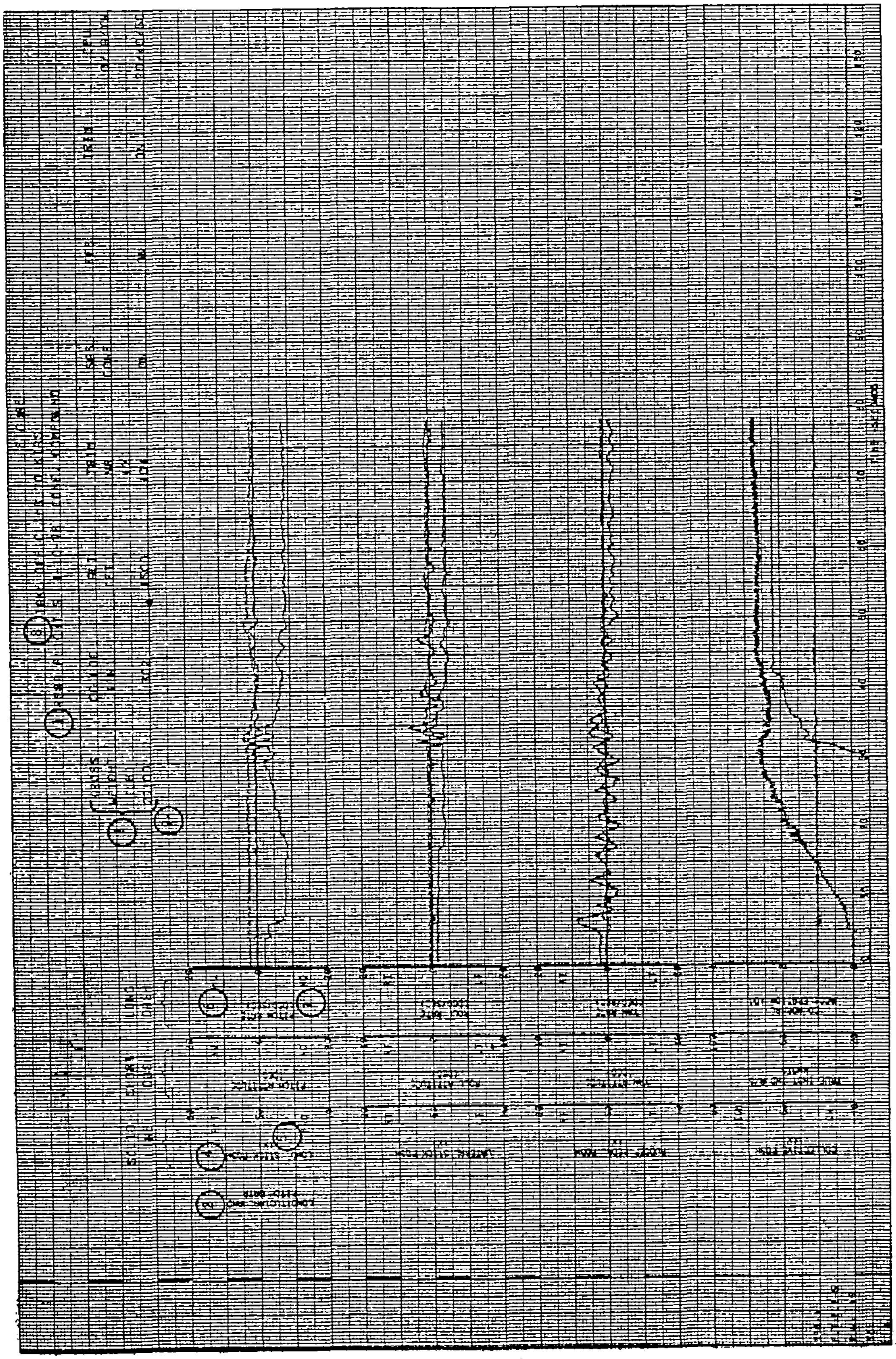


Figure 33 - Sample time history plot.

```
                    THIS IS THE FLIGHT TITLE FOR THE RSRA HARMONIC ANALYSIS VALIDATION RUN
                                     WHAT GOES UP MUST COME DOWN...

HARMONIC ANALYSIS VCOPFV        VCOPFV      D

   RUN  25        L.F.100KT.104% NR                CYCLES     1TO   5

          N             AN                BN                 CN                PHIN            STD DEV(C)

          0            1.1792                               1.1792                                0.0568
          1           -0.0118            0.0736             0.0745             99.1417            0.1345
          2           -0.0273            0.0059             0.0280            167.7505            0.1208
          3           -0.0235           -0.0203             0.0311           -139.0679            0.0565
          4            0.0452            0.0022             0.0452              2.8270            0.0789
          5           -0.0800            0.0826             0.1150            134.0885            0.0504
          6           -0.0327           -0.0434             0.0544           -126.9962            0.1057
          7            0.0020           -0.0485             0.0486            -87.6382            0.0994
          8            0.0297           -0.0252             0.0390            -40.3282            0.0849
          9            0.0375            0.0260             0.0456             34.8022            0.1040
         10           -0.2442            0.1550             0.2893            147.5923            0.0653
                    THE ABOVE AVERAGE VALUES WERE CALCULATED BASED ON  5 CYCLES OF DATA
                   0 CYCLES CONTAINED NULL SAMPLES

   RUN  26        L.F.128KT.104% NR                CYCLES     1TO   5

          N             AN                BN                 CN                PHIN            STD DEV(C)

          0            1.1165                               1.1165                                0.0990
          1           -0.0328            0.1287             0.1328            104.2729            0.1070
          2           -0.0022            0.0542             0.0543             92.2758            0.0374
          3           -0.0546           -0.0086             0.0553           -171.0564            0.0494
          4            0.0948            0.0332             0.1005             19.3145            0.0825
          5           -0.1775            0.0406             0.1821            167.1213            0.0972
          6           -0.0501           -0.0007             0.0501           -179.1943            0.0453
          7           -0.0104           -0.0076             0.0129           -143.6939            0.0068
          8            0.0224           -0.0135             0.0261            -31.1213            0.0180
          9           -0.0588            0.0238             0.0634            157.9815            0.0642
         10           -0.0289           -0.0583             0.0650           -116.3418            0.0912
                    THE ABOVE AVERAGE VALUES WERE CALCULATED BASED ON  2 CYCLES OF DATA
                   3 CYCLES CONTAINED NULL SAMPLES
```

Figure 34 - Sample listing of Harmonic Analysis program.

CARD NO.	COLS.	CARD FIELD CONTENTS
1 & 2	1-72	General title cards.
3→n	1-3	Data value "RUN".
	6-10	Flight maneuver numbers to be processed (right-justified).
	11-15	NOTE: First blank field encountered on card image terminates the processing of input flight maneuver numbers.
	16-20	
	21-25	
	26-30	
	31-35	
	76-80	
n+1→m	1-8	Sensor mnemonic name (left-justified).
	11-15	Sample rate of sensor data (right-justified).
	16-20	Beginning cycle number of data within flight maneuver to commence calculation of harmonic coefficients and phase angles (right-justified).
	21-25	Ending cycle number (right-justified).

Figure 35 - Card image for Harmonic Analysis program.

VARIABLE NAME	DESCRIPTION	WORD(S) NO.	DATA TYPE
DNAME	Concatenated name and pre-processing code of sensor (first eight characters are mnemonic name and eleventh character is pre-processing code).	1,2	C
PC	Parameter category code.	3	C
JINDEX	Integer precision code index.	4	I
CTITL	Six word column title.	5 10	C
LTITL	Four word line title.	11 14	C

Figure 36 - Data description of UPC file data entry.

CARD NO.	COLS.	CARD FIELD CONTENTS
1→n	1-2	Parameter category code.
	4-11	Sensor mnemonic name (left-justified).
	13	Preprocessing code for sensor entry.
	16-17	Integer precision code index (right-justified).
	19-54	The 36-character column title.
	57-80	The 24-character line title.

Figure 37 - UPCGEN program data card description.

```
FWDFUS10  V SA   1  FWD   FUS   #10          PSI  (V) FWD FUS #10 (V)
FWDFUS1   S SA   1  FWD   FUS   #1           PSI  (S) FWD FUS #1  (S)
FWDFUS1   V SA   1  FWD   FUS   #1           PSI  (V) FWD FUS #1  (V)
FWDFUS2   S SA   1  FWD   FUS   #2           PSI  (S) FWD FUS #2  (S)
FWDFUS2   V SA   1  FWD   FUS   #2           PSI  (V) FWD FUS #2  (V)
FWDFUS3   S SA   1  FWD   FUS   #3           PSI  (S) FWD FUS #3  (S)
FWDFUS3   V SA   1  FWD   FUS   #3           PSI  (V) FWD FUS #3  (V)
FWDFUS4   S SA   1  FWD   FUS   #4           PSI  (S) FWD FUS #4  (S)
FWDFUS4   V SA   1  FWD   FUS   #4           PSI  (V) FWD FUS #4  (V)
FWDFUS5   S SA   1  FWD   FUS   #5           PSI  (S) FWD FUS #5  (S)
FWDFUS5   V SA   1  FWD   FUS   #5           PSI  (V) FWD FUS #5  (V)
FWDFUS6   S SA   1  FWD   FUS   #6           PSI  (S) FWD FUS #6  (S)
FWDFUS6   V SA   1  FWD   FUS   #6           PSI  (V) FWD FUS #6  (V)
FWDFUS7   S SA   1  FWD   FUS   #7           PSI  (S) FWD FUS #7  (S)
FWDFUS7   V SA   1  FWD   FUS   #7           PSI  (V) FWD FUS #7  (V)
FWDFUS8   S SA   1  FWD   FUS   #8           PSI  (S) FWD FUS #8  (S)
FWDFUS8   V SA   1  FWD   FUS   #8           PSI  (V) FWD FUS #8  (V)
FWDFUS9   S SA   1  FWD   FUS   #9           PSI  (S) FWD FUS #9  (S)
FWDFUS9   V SA   1  FWD   FUS   #9           PSI  (V) FWD FUS #9  (V)
FX        D SM   0  MAIN  ROTOR LONG  FORCE  LBS (S)  MR LONG FORCE  LBS (S)
FY        D SM   0  MAIN  ROTOR LAT   FORCE  LBS (S)  MR LAT FORCE   LBS (S)
FZ        D SM   0  MAIN  ROTOR VERT  FORCE  LBS (S)  MR VERT FORCE  LBS (S)
HPBOOM    D RK   0  PRESS ALT   BOOM  FT              PRESS ALTITUDE(BOOM) FT
HDBOOM    D RK   0  DENS  ALT   BOOM  FT              DENSITY ALT(BOOM) FT
HEAD180   D HA  -1  COMP  HEAD  +/-   180    DEG      COMPASS HEAD +/- 180-DEG
HLATSTRM  D HM  -1  HELO  LAT   SERVO TRIM   DEG      HELO LAT SERVO TRIM-DEG
HLGSTRM   D HM  -1  HELO  LONG  SERVO TRIM   DEG      HELO LG SERVO TRIM-DEG
HPMR      D PT   0  MAIN  ROTOR POWER HP              MAIN ROTOR POWER HP
HPG       D PT   0  TOTAL STRN  GAGE  POWER           TOTAL STRAIN GAGE POWER
HPT       D PT   0  TOTAL ENG   POWER HP              TOTAL ENGINE POWER HP
HPTR      D PT   0  TAIL  ROTOR POWER HP              TAIL ROTOR POWER HP
HZSTBACT  S SA   1  HORZ  STAB  ACT   ROD    PSI  (S) HORZ STAB ACT ROD PSI(S)
HZSTBACT  V SA   1  HORZ  STAB  ACT   ROD    PSI  (V) HORZ STAB ACT ROD PSI(V)
HZSTELEP  S SA   0  HORZ  STAB  ELEV  POS    %    (S) HZ STAB ELEV POS-% (S)
HZSTELEP  V SA   0  HORZ  STAB  ELEV  POS    %    (V) HZ STAB ELEV POS-% (V)
IAU1      S SM   1  ISOL  AXIAL LOAD  #1        LB(S)ISOL AXIAL LOAD #1   (S)
IAU1      V SM   1  ISOL  AXIAL LOAD  #1        LB(V)ISOL AXIAL LOAD #1   (V)
IAU2      S SM   1  ISOL  AXIAL LOAD  #2        LB(S)ISOL AXIAL LOAD #2   (S)
IAU2      V SM   1  ISOL  AXIAL LOAD  #2        LB(V)ISOL AXIAL LOAD #2   (V)
IAU3      S SM   1  ISOL  AXIAL LOAD  #3        LB(S)ISOL AXIAL LOAD #3   (S)
IAU3      V SM   1  ISOL  AXIAL LOAD  #3        LB(V)ISOL AXIAL LOAD #3   (V)
IAU4      S SM   1  ISOL  AXIAL LOAD  #4        LB(S)ISOL AXIAL LOAD #4   (S)
IAU4      V SM   1  ISOL  AXIAL LOAD  #4        LB(V)ISOL AXIAL LOAD #4   (V)
ICEBATH   D RK   0  ICE   BATH  REF   TEMP DEG C      ICEBATH REF TEMP DEG C
```

Figure 38 - UPCGEN sample listing of sorted sensor data.

ELEVGRDR	S	SA	0	ELEV	CONT	ROD R	LBS	(S)	ELEV CONT ROD R LBS (S)
ELEVGRDR	V	SA	0	ELEV	CONT	ROD R	LBS	(V)	ELEV CONT ROD R LBS (V)
ELEVGROD	S	SA	0	ELEV	CONT	ROD	LBS	(S)	ELEV CONT ROD LBS (S)
ELEVGROD	V	SA	0	ELEV	CONT	ROD	LBS	(V)	ELEV CONT ROD LBS (V)
FLAPGPOS	0	SA	0	FLAP	CONT	POS	%		FLAP CONT POS=%
FWBFUS10	S	SA	1	FWD	FUS	#10		PSI	(S) FWD FUS #10 (S)
FWBFUS10	V	SA	1	FWD	FUS	#10		PSI	(V) FWD FUS #10 (V)
FWBFUS1	S	SA	1	FWD	FUS	#1		PSI	(S) FWD FUS #1 (S)
FWBFUS1	V	SA	1	FWD	FUS	#1		PSI	(V) FWD FUS #1 (V)
FWBFUS2	S	SA	1	FWD	FUS	#2		PSI	(S) FWD FUS #2 (S)
FWBFUS2	V	SA	1	FWD	FUS	#2		PSI	(V) FWD FUS #2 (V)
FWBFUS3	S	SA	1	FWD	FUS	#3		PSI	(S) FWD FUS #3 (S)
FWBFUS3	V	SA	1	FWD	FUS	#3		PSI	(V) FWD FUS #3 (V)
FWBFUS4	S	SA	1	FWD	FUS	#4		PSI	(S) FWD FUS #4 (S)
FWBFUS4	V	SA	1	FWD	FUS	#4		PSI	(V) FWD FUS #4 (V)
FWBFUS5	S	SA	1	FWD	FUS	#5		PSI	(S) FWD FUS #5 (S)
FWBFUS5	V	SA	1	FWD	FUS	#5		PSI	(V) FWD FUS #5 (V)
FWBFUS6	S	SA	1	FWD	FUS	#6		PSI	(S) FWD FUS #6 (S)
FWBFUS6	V	SA	1	FWD	FUS	#6		PSI	(V) FWD FUS #6 (V)
FWBFUS7	S	SA	1	FWD	FUS	#7		PSI	(S) FWD FUS #7 (S)
FWBFUS7	V	SA	1	FWD	FUS	#7		PSI	(V) FWD FUS #7 (V)
FWBFUS8	S	SA	1	FWD	FUS	#8		PSI	(S) FWD FUS #8 (S)
FWBFUS8	V	SA	1	FWD	FUS	#8		PSI	(V) FWD FUS #8 (V)
FWBFUS9	S	SA	1	FWD	FUS	#9		PSI	(S) FWD FUS #9 (S)
FWBFUS9	V	SA	1	FWD	FUS	#9		PSI	(V) FWD FUS #9 (V)
HZSTBACT	S	SA	1	HORZ	STAB	ACT	ROD	PSI	(S) HORZ STAB ACT ROD PSI(S)
HZSTBACT	V	SA	1	HORZ	STAB	ACT	ROD	PSI	(V) HORZ STAB ACT ROD PSI(V)
HZSTELEP	S	SA	0	HORZ	STAB	ELEV	POS	%	(S) HZ STAB ELEV POS=% (S)
HZSTELEP	V	SA	0	HORZ	STAB	ELEV	POS	%	(V) HZ STAB ELEV POS=% (V)
LAICB	S	SA	1	LEFT	ARON	INPUT	CONT	ROD	LB(S)LT ARON INPUT CR LB (S)
LAICB	V	SA	1	LEFT	ARON	INPUT	CONT	ROD	LB(V)LT ARON INPUT CR LB (V)
LBRGBKCR	S	SA	1	LEFT	DRAG	BRAKE	CONT	ROD	PSI=SLT CRAG BRAKE CR PSI(S)
LBRGBKCR	V	SA	1	LEFT	DRAG	BRAKE	CONT	ROD	PSI=VLT CRAG BRAKE CR PSI(V)
LEGRGR	S	SA	1	LEFT	ELEV	GRND	ROD	PSI	(S) LT ELEV GRND ROD PSI (S)
LEGRGR	V	SA	1	LEFT	ELEV	GRND	ROD	PSI	(V) LT ELEV GRND ROD PSI (V)
LGFORCE	0	SA	-1	LONG	STK	FORCE	LB		**AFTLONG STICK FORCE=LB
LHSDRAGL	S	SA	1	LOWER	HORZ	STAB	DRAG	LEFT	PSI(SLOW HZ STAB DRAG LT PSIS
LHSDRAGL	V	SA	1	LOWER	HORZ	STAB	DRAG	LEFT	PSI(VLOW HZ STAB DRAG LT PSIV
LHSDRAGR	S	SA	1	LOWER	HORZ	STAB	DRAG	RIGHT	PSI(SLOW HZ STAB DRAG RT PSIS
LHSDRAGR	V	SA	1	LOWER	HORZ	STAB	DRAG	RIGHT	PSI(VLOW HZ STAB DRAG RT PSIV
LHSLIFTL	S	SA	1	LOWER	HORZ	STAB	LIFT	LEFT	PSI(SLOW HZ STAB LIFT LT PSIS
LHSLIFTL	V	SA	1	LOWER	HORZ	STAB	LIFT	LEFT	PSI(VLOW HZ STAB LIFT LT PSIV
LHSLIFTR	S	SA	1	LOWER	HORZ	STAB	LIFT	RIGHT	PSI(SLOW HZ STAB LIFT RT PSIS
LHSLIFTR	V	SA	1	LOWER	HORZ	STAB	LIFT	RIGHT	PSI(VLOW HZ STAB LIFT RT PSIV

Figure 39 - UPCGEN sample listing of sorted data by category.

CARD NO.	COLS.	CARD FIELD CONTENTS
1→n	2-18	Starting time.
	20-36	Stopping time. NOTE: The format for both starting and stopping time is DAY HR MIN SEC XXX-XX-XX-XX.XXXX
	40-43	Number of records (right-justified).
	45-46	Tape file number (right-justified).
	48-50	Starting data word number (right-justified).
	52-54	Stopping data word number (right-justified). NOTE: If any of the above data fields are zero or blank the following procedures are assumed. 1) If the stopping time is zero or blank, then the number of records specified in columns 40-43 are listed. 2) If the number of records is also zero or blank, then twenty records will be listed. 3) If the file number is zero or blank, then a file number of "1" is assumed. 4) If the starting and stopping data word numbers are blank, then all data words within each frame is listed. *NOTE: Any number of data cards may be present for listing different portions of the data residing on the input tape.

Figure 40 - Description of card image for TPDMP program.

```
 RSRA 545 7203 FLT05
WORDS   1 TO  42 FILE NO,   1

100-15-36-23, 990  001511 000447 003730 003641 003703 003646 003637 003620 004023 003754 004011 003735 004061 004027 003602 003517
                   002624 002567 003575 003525 004077 003761 003655 003576 003652 003613 003737 003635 003543 003451 003717 003617
                   003580 003515 003612 003507 003613 003545 003671 003612 003671 003575 001511

100-15-36-23,1690  001516 000455 003734 003646 003712 003660 003642 003623 004032 003764 004020 003744 004065 004030 003607 003526
                   002631 002573 003577 003540 004076 003764 003654 003575 003654 003614 003740 003640 003543 003451 003714 003617
                   003591 003515 003610 003505 003613 003543 003673 003611 003702 003600 001516

100-15-36-23,2404  001517 000453 003723 003640 003703 003647 003635 003615 004025 003755 004010 003734 004054 004023 003601 003520
                   002623 002565 003577 003543 004070 003757 003647 003575 003656 003613 003740 003636 003541 003446 003701 003616
                   003556 003510 003605 003502 003603 003542 003665 003607 003665 003574 001517

100-15-36-23,3105  001513 000453 003726 003637 003701 003647 003640 003617 004023 003754 004010 003731 004060 004025 003603 003514
                   002627 002566 003576 003547 004072 003756 003655 003574 003654 003613 003740 003633 003544 003447 003713 003620
                   003563 003513 003610 003503 003610 003542 003670 003610 003674 003575 001513

100-15-36-23,3810  001517 000456 003725 003634 003703 003647 003640 003615 004021 003755 004007 003727 004060 004017 003601 003517
                   002624 002565 003574 003550 004071 003755 003647 003574 003654 003614 003735 003636 003541 003443 003704 003616
                   003527 003512 003611 003506 003611 003544 003671 003611 003672 003600 001517

100-15-36-23,4510  001517 000452 003722 003634 003701 003645 003634 003614 004020 003752 004007 003731 004054 004021 003601 003516
                   002621 002561 003572 003543 004063 003754 003646 003570 003651 003613 003735 003627 003541 003444 003714 003615
                   003527 003511 003606 003504 003607 003543 003673 003614 003672 003600 001517

100-15-36-23,5220  001515 000453 003741 003652 003712 003661 003645 003624 004035 003767 004020 003746 004071 004033 003610 003527
                   002633 002573 003601 003553 004103 003765 003657 003576 003651 003613 003735 003631 003540 003444 003710 003612
                   003554 003507 003601 003501 003610 003542 003667 003610 003672 003575 001515

100-15-36-23,5930  001515 000451 003726 003636 003702 003651 003640 003616 004023 003754 004007 003734 004062 004023 003603 003523
                   002625 002566 003574 003547 004066 003754 003643 003570 003651 003611 003732 003630 003537 003443 003705 003616
                   003525 003507 003607 003505 003607 003544 003670 003610 003673 003601 001515

100-15-36-23,6630  001515 000452 003720 003633 003677 003646 003637 003614 004020 003753 004004 003731 004057 004022 003577 003517
                   002617 002557 003576 003512 004072 003757 003646 003571 003652 003614 003736 003632 003536 003445 003710 003616
                   003527 003513 003606 003501 003604 003541 003664 003605 003663 003571 001515

100-15-36-23,7340  001523 000456 003730 003643 003703 003652 003641 003621 004027 003760 004013 003745 004064 004027 003601 003524
                   002624 002566 003574 003530 004070 003757 003653 003573 003651 003610 003732 003631 003537 003445 003716 003613
                   003554 003507 003604 003502 003606 003541 003672 003605 003675 003574 001523

100-15-36-23,8043  001515 000454 003730 003642 003705 003651 003637 003617 004030 003760 004013 003735 004061 004026 003602 003522
                   002625 002566 003573 003540 004074 003756 003650 003576 003653 003611 003735 003634 003541 003451 003676 003617
```

Figure 41 - Sample listing of TPDMP program.

```
*** HEADER RECORD INFORMATION FOR FLIGHT= FLT=5***
MAXIMUM PASS NUMBER= 13
MAXIMUM NUMBER OF RUNS= 47
DATE OF FLIGHT= 041078
FLIGHT TAPE NUMBER= 51-A8
AIRCRAFT= /C545     R     PILOT= INE          COPILOT= STOFER          OBSERVER=
ONCE PER FLIGHT DATA VALUES=
  ETP= 72G-3          ESGW=  0.2711E 05     ESCG=   302.1
  KPR=  0.9000         HPC=   1216.          MRC=   203.0
    R=   31.00          RT=   5.300           GT=   6.123
  GTS=   3030.       FTBOOM=  0.         FHBOOM=   0.
```

Figure 42a. - Sample listing of TABLE PRINT program. (Header Table)

*** PASS INFORMATION FOR FLIGHT= FLT-5***

PASS NO	1	2	3	4	5	6	7	8	9	10	11	12	13
NPAR	30	0	24	40	36	40	28	12	34	29	36	0	29
SRDIN	1M		5M	5M	5M	5M	5M	1T	80X	80X	80X		20X
SRDATA	1M		5M	5M	5M	5M	5M	1T	20X	20X	5X		20X
ISTATC	2	0	2	2	2	2	2	2	2	2	2	0	3
DATA MAP													
1	1	0	1	1	1	1	1	1	1	1	1	0	0
2	3	0	3	3	3	3	3	2	3	3	2	0	0
3	5	0	5	5	5	5	5	4	6	6	3	0	0
4	7	0	7	7	7	7	7	6	8	8	4	0	0
5	9	0	9	9	9	9	9	8	10	10	5	0	0
6	11	0	11	11	11	11	11	10	13	13	6	0	0
7	13	0	13	13	13	13	13	12	16	16	7	0	0
8	15	0	15	15	15	15	15	14	19	19	8	0	0
9	17	0	17	17	17	17	17	16	22	22	9	0	0
10	19	0	19	19	19	19	19	18	25	25	10	0	0
11	22	0	22	22	22	22	22	21	29	29	11	0	0
12	24	0	24	24	24	24	24	23	32	32	12	0	0
13	26	0	26	26	26	26	26	25	35	35	13	0	0
14	28	0	28	28	28	28	28	27	37	37	14	0	0
15	30	0	30	30	30	30	30	29	40	40	15	0	1
16	35	0	55	55	55	55	55	59	68	68	22	0	106
17	39	0	75	75	75	75	75	84	91	91	28	0	194
18	40	0	80	80	80	80	80	90	96	96	30	0	213
19	42	0	87	87	87	87	87	99	104	104	32	0	242
20	44	0	95	95	95	95	95	109	113	113	35	0	276
21	46	0	102	102	102	102	102	118	121	121	37	0	307
22	50	0	122	122	122	122	122	142	143	143	43	0	392
23	53	0	133	133	133	133	133	156	156	156	47	0	440
24	57	0	150	150	150	150	150	177	175	175	52	0	514
25	59	0	157	157	157	157	157	186	183	183	54	0	545
26	62	0	171	171	171	171	171	204	199	199	58	0	605
27	66	0	190	190	190	190	190	227	221	221	64	0	688
28	70	0	206	206	206	206	206	247	239	239	69	0	756

Figure 42b. - Sample listing of TABLE PRINT program. (Pass Table)

*** SENSOR INFORMATION FOR FLIGHT= FLT=5PASS= 4***

I	MNEMONIC	PP	IPIN	IPDF	CAL	EUCV1	EUCV2	CONCVL	A	B	CRPT	PSTAT
1	MRGBQCE	V	3	1	X	14370.0	0.	620.000	103.394	0.	3	1
2	MRGBQCE	S	4	2	X	14370.0	0.	0.	103.394	-26412.5	3	1
3	MRGBQCF	V	5	3	X	14656.0	0.	620.000	107.995	0.	3	1
4	MRGBQCF	S	6	4	X	14656.0	0.	0.	107.995	-27902.3	3	1
5	LRLL	V	7	5	X	-7000.00	0.	0.	-33.3404	0.	1	1
6	LRLL	S	8	6	X	-7000.00	0.	0.	-33.3404	8552.57	1	1
7	TRTHRN	V	9	7	X	2822.00	0.	500.000	21.6472	0.	3	1
8	TRTHRN	S	10	8	X	2822.00	0.	0.	21.6472	-5661.29	3	1
9	RLLL	V	11	9	X	7000.00	0.	0.	33.4756	0.	1	1
10	RLLL	S	12	10	X	7000.00	0.	0.	33.4756	-8847.16	1	1
11	MRLIFTA	V	13	11	X	14250.0	0.	2150.00	106.676	0.	3	1
12	MRLIFTA	S	14	12	X	14250.0	0.	0.	106.676	-27369.4	3	1
13	MRLIFTB	V	15	13	X	14491.0	0.	2150.00	105.370	0.	3	1
14	MRLIFTB	S	16	14	X	14491.0	0.	0.	105.370	-27258.6	3	1
15	MRLIFTC	V	17	15	X	14043.0	0.	2150.00	101.499	0.	3	1
16	MRLIFTC	S	18	16	X	14043.0	0.	0.	101.499	-26606.8	3	1
17	MRLIFTD	V	19	17	X	14116.0	0.	2150.00	102.055	0.	3	1
18	MRLIFTD	S	20	18	X	14116.0	0.	0.	102.055	-26222.7	3	1
19	MRDRAG	V	21	19	X	4586.00	0.	1160.00	32.5289	0.	3	1
20	MRDRAG	S	22	20	X	4586.00	0.	0.	32.5289	-8495.11	3	1
21	WINGJ	V	23	21	X	16708.0	0.	0.	158.614	0.	3	1
22	WINGJ	S	24	22	X	16708.0	0.	0.	158.614	-40573.2	3	1
23	WINGL	V	25	23	X	7130.00	0.	0.	70.4227	0.	3	1
24	WINGL	S	26	24	X	7130.00	0.	0.	70.4227	-17771.6	3	1
25	WINGH	V	27	25	X	13998.0	0.	0.	139.188	0.	3	1
26	WINGH	S	28	26	X	13998.0	0.	0.	139.188	-36188.0	3	1
27	WINGI	V	29	27	X	14735.0	0.	0.	138.989	0.	3	1
28	WINGI	S	30	28	X	14735.0	0.	0.	138.989	-35536.2	3	1
29	VNGTPRT	V	31	29	X	13.9100	0.	0.	0.741501E-01	0.	1	1
30	VNGTPRT	S	32	30	X	13.9100	0.	0.	0.741501E-01	-19.4413	1	1
31	LHSLIFTL	V	33	31	X	2089.80	0.	0.	15.3898	0.	1	1
32	LHSLIFTL	S	34	32	X	2089.80	0.	0.	15.3898	-4070.59	1	1
33	WINGK	V	35	33	X	16590.0	0.	0.	154.248	0.	3	1
34	WINGK	S	36	34	X	16590.0	0.	0.	154.248	-39185.7	3	1
35	WINGM	V	37	35	X	7026.00	0.	0.	66.5122	0.	3	1
36	WINGM	S	38	36	X	7026.00	0.	0.	66.5122	-16856.8	3	1
37	LHSLIFTR	V	39	37	X	2089.80	0.	0.	15.5479	0.	1	1
38	LHSLIFTR	S	40	38	X	2089.80	0.	0.	15.5479	-3909.92	1	1
39	LOWGTPRT	V	41	39	X	-13.4800	0.	0.	-0.703989E-01	0.	1	1
40	LOWGTPRT	S	42	40	X	-13.4800	0.	0.	-0.703989E-01	17.7484	1	1

Figure 42c. - Sample listing of TABLE PRINT program. (Sensor Table)

EVENT RECORD INFORMATION FOR FLIGHT= FLT=5***

I	EVENT	TYPE	TIME1
1	XCAL PADS 1	X	54478,014
2	RCAL PADS 1	R	54487,846
3	ZCAL PADS 1	Z	54499,978
4	XCAL PADS 2	X	54509,534
5	RCAL PADS 2	R	54518,848
6	ZCAL PADS 2	Z	54529,855
7	LEAD STOP,SIDESLIP=+20	X	54540,559
8	LAG STOP,SIDESLIP= -20	X	54552,401
9	TR OUTBD,TF34 MAX,A=20	X	54563,309
10	TR INBD,TF34 G,I,,A=-20	X	54573,958
11	LT=LT-LO=FWD,FLAPS UP	X	54589,063
12	LT=LT-HI=FWD,FLAPS UP	X	54599,870
13	RT=RT-LO=AFT,FLAPS DWN	X	54612,219
14	RT=RT-HI=AFT,FLAPS DWN	X	54620,968
15	L/O AND CLIMB AT 70 KTS	T	54632,088
16	RT CLIMBING TURN,70 KTS	T	54715,118
17	LF 70 KTS, AUX G,I,	S	54785,588
18	LF 80 KTS, AUX G,I,	S	54803,837
19	LF 90 KTS, AUX G,I,	S	54830,100
20	LF 100 KTS, AUX G,I,	S	54859,681
21	RT TURN 80 KTS, 15/30DG	T	54887,018
22	LF 60 KTS, AUX G,I,	T	54955,252
23	LT TURN 80KTS, 15/30DEG	T	54995,746
24	LF 80 KTS, GEAR UP	T	55055,434
25	PPD 500 FPM, 80KTS, GU	T	55082,986
26	CL 500 FPM, 80KTS, GU	T	55132,620
27	PPD 500 FPM,LT/RT TN,GU	T	55199,294
28	APPROACH AND LANDING	T	55254,785
29	LT=LT-LO=FWD, FLAPS UP	X	55427,814
30	LT=LT-HI=FWD, FLAPS UP	X	55436,938
31	RT=RT-LO=AFT, FLAPS DWN	X	55447,807
32	RT=RT-HI=AFT, FLAPS DWN	X	55458,073
33	LEAD STOP,SIDESLIP=+20	X	55467,273
34	LAG STOP, SIDESLIP=-20	X	55476,322
35	TR OUTBD,TF34 MAX,A= 20	X	55486,534
36	TR INBD, TF34 G,I,A=-20	X	55497,872
37	XCAL PADS 1	X	55509,501
38	RCAL PADS 1	R	55521,049
39	ZCAL PADS 1	Z	55531,514

Figure 42d. - Sample listing of TABLE PRINT program. (Event Table)

*** CALRUN INFORMATION FOR FLIGHT= FLT=5***

ENTRY	PRE=Z/LO	PRE=R/HI	PRE=XDUC	POST=Z/LO	POST=R/HI	POST=XDUC
1	4	5	4	40	41	40
2	3	2	1	39	38	37
3	1	2	1	37	38	37
4	1	3	1	37	39	37
5	6	5	4	42	41	40
6	12	13	0	30	31	0
7	13	12	0	31	30	0
8	3	2	11	39	38	29
9	11	14	0	29	32	0
10	7	8	0	33	34	0
11	9	10	0	35	36	0
12	10	9	0	36	35	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0

Figure 42e. - Sample listing of TABLE PRINT program. (Calrun Table)

APPENDIX A

The Honeywell 600 Job Control Language for the EASE System

The EASE software system was originally designed for operation on the Honeywell 600 computer system. Many of the design features have been defined for optimal use of this computer equipment using DSS 180 disk packs. These disk packs can contain approximately 145 million bits or 4.032 million 36-bit data words of usable storage. The operation of the EASE system has been configured to allow sufficient storage for a complete flight of data on one DSS 180 disk pack.

The Raw Data File (RDF) for a flight consists of several physical files: the Table File, the Statistics File, and the Data Files -- one for each pass of data. One DSS 180 disk pack is designed for storing a basic flight data configuration for the RSRA project. Figure A-1 is a summary of the file configuration for the RDF. Such a file scheme is sufficient for most flights.

The structure of the file configuration is defined through the Honeywell 600 File Management System[1]. A physical space is defined through the File Management System by a catalogue file structure. Presently, there are two DSS 180 disk packs assigned. Each is configured with the same physical size for the files in the RDF so that data from two flights can be processed at one time. The distinction between data files configured on one disk pack from those on the second disk pack is by the catalogue file structure. This structure definition begins from the assigned user master catalogue name RSRATEST. Sub-catalogue names are assigned under RSRATEST for each disk pack.

These sub-catalogue names are FLTA and FLTB. Under the sub-catalogue FLTA, is another catalogue name residing on the disk pack called PCKA. Similarly, under FLTB is a catalogue called PCKB. File names are then assigned to each physical file required by the RDF. The disk pack with a sub-catalogue structure of FLTA/PCKA is assigned a label name of RSRA1 and the disk pack structured as FLTB/PCKB is assigned a label name of RSRA2. Hence, data to be stored or accessed using the disk pack RSRA1 is defined through the catalogue string RSRATEST/FLTA/PCKA, and for disk pack RSRA2 the catalogue string is RSRATEST/FLTB/PCKB.

The names assigned to each of the physical files of the RDF are the same for both RSRA1 and RSRA2. These file names are stated in figure A-1. All of these file names are associated with a respective file code within each program of the EASE system. For example, the Table File, named TBLFL, is assigned a file code of 07. This association is defined through the job control language for the Honeywell 600 system. It is through this

---

[1] DB54, File Management Supervisor, Honeywell 600 System Reference Manual, March 1973.

means that the program being run is identified, all user supplied and system routines are loaded, and all peripheral allocations are made. The following is a brief description in the order of input, of the job control language cards required by each program of the EASE system[2]. The title of each card described is contained on the job control card beginning in column 8.

SNUMB. This is the first card of any job run and is inserted by the computer operations personnel. It contains a unique number for the job being executed.

IDENT. The second card is a program identification card. It contains the assigned program number for the software.

USERID. This card informs the operating system what user master catalogue is being accessed along with the appropriate password identification.

MSG1. This card contains textual information that is written on the operator's console. It is used here to inform the operator which removable disk pack is to be placed on-line.

OPTION. This card is used here to automatically allocate the card reader, card punch, and line printer to the normal FORTRAN defined file codes.

EXECUTE. This control card instructs the operating system to load the program and any required system library routines and executes the program.

LIMITS. This control card contains four pieces of information required to load and execute the program. First is the amount of CPU time in decimal hours required to execute the program. Second is the amount of memory in computer words to load all user and system routines. Third is the overlay area required in computer words. And last is the total amount of lines of output to be generated on the line printer. All of these values are basically limits that instruct the operating system not to exceed.

TAPE, TAPE9, FILE, PRMFL, SYSOUT. These are the peripheral allocation cards. Each one allocates a particular type of input/output device to be used by the program. The "TAPE" card allocates a 7-track tape handler, the "TAPE9" card allocates a 9-track tape handler, the "FILE" card allocates a scratch disk area, the "PRMFL" card allocates a particular assigned permanent file area, and the "SYSOUT" card assigns the printer in an off-line mode. Each of these cards contain further information to associate a named file code form within the software to these devices.

ENDJOB. This is the last card in the job run and instructs the operating system that no further activities are to be processed for this job run.

The program and input data decks are also included as part of a job run. The program deck is placed after the "OPTION" card and the input data cards are placed immediately preceding the "ENDJOB" card.

---

[2] BS19, Control Cards Reference Manual, Honeywell 600 System Reference Manual, February 1973.

Figures A-2 through A-13 contain the job control language cards required to execute each of the programs in the EASE system. These figures are examples using the RSRA1 disk pack for the RDF. These examples can be applied to another assigned disk pack simply by requesting the appropriate disk pack on the "MSG1" card and defining the correct catalogue file string on the "PRMFL" cards. For example, to use the RSRA2 disk pack in the SETUP program, the text of the "MSG1" card would be changed to read: 1, PLEASE MOUNT DISC PACK RSRA2. The "PRMFL" card would be changed to read 07, R/W,R, RSRATEST$XXXX/FLTB/PCKB/TBLFL.

Within the information contained in each of the "USERID" and "PRMFL" control cards, the file defined as "XXXX" must contain the correct password assigned to the user master catalogue, RSRATEST. This information shall be given to authorized users of the system. Also, on the "TAPE" and "TAPE 9" control cards allocating input tapes, the field designated as "NNNN" is to be replaced by the assigned magnetic tape number.

It is not necessary to run each program independently. For example, the SETUP, SCAN, and ABTASK programs could be stacked to execute as one job run with each of the three programs as separate activities. To do this, only one set of job control cards for SNUMB, IDENT, USERID, MSG1, and ENDJOB are required. That is the first four job control cards are required only once and can be eliminated from the beginning of the deck setup for the SCAN and ABTASK. Also the ENDJOB is only required after the last activity so this card can be eliminated from the end of the SETUP and SCAN program. Figure A-14 lists the deck setup for running these three programs as one job run. This can be done for any of the programs in the EASE system. However, care should be taken so that the order of activities follows a logical operational procedure for processing the data.

RDF File	Data File Sample Rate	No. Samples Per Sec.	File Name Mnemonic	File Code	Size in Links*
Table File			TBLFL	07	7
Statistics File			STATS	09	149
Pass 1 Data File	1/M	Approx. 3	PASS 1	11	25
Pass 2 Data File	1/M	Approx. 3	PASS 2	12	25
Pass 3 Data File	5/M	Approx. 17	PASS 3	13	72
Pass 4 Data File	5/M	Approx. 17	PASS 4	14	72
Pass 5 Data File	5/M	Approx. 17	PASS 5	15	72
Pass 6 Data File	5/M	Approx. 17	PASS 6	16	72
Pass 7 Data File	5/M	Approx. 17	PASS 7	17	72
Pass 8 Data File	1/T	Approx. 17	PASS 8	18	72
Pass 9 Data File	20X	20	PASS 9	19	72
Pass 10 Data File	20X	20	PASS 10	20	72
Pass 11 Data File	5X	5	PASS 11	21	21
Pass 12 Data File	5X	5	PASS 12	22	21
Pass 13 Derived Par. Data File	20X	20	PASS 13	23	226
			Total Size of RDF		1050

*1 Link = 3840 words

Figure A-1. - File configuration for RSRA raw data file (RDF)

```
Col.     Col.           Col.
 1        8              16
 $       SNUMB
 $       IDENT          112320,EASE
 $       MSG1           1,PLEASE MOUNT DISC PACK RSRA1
 $       USERID         RSRATEST$XXXX
 $       OPTION         FORTRAN
(Program decks)
 $       EXECUTE
 $       LIMITS         08,19K,,2K
 $       SYSOUT         08
 $       PRMFL          07,R/Q,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $       FILE           01,AIR,3R
(Data cards)
 $       ENDJOB
*** EOF
```

Figure A-2. - Setup routine deck setup using disk pack RSRA1.

```
Col.     Col.           Col.
 1        8              16
 $       SNUMB
 $       IDENT          112320,EASE
 $       MSG1           1,PLEASE MOUNT DISC PACK RARA1
 $       USERID         RSRATEST$XXXX
(Program deck)
 $       EXECUTE
 $       LIMITS         50,24K,,2K
 $       TAPE9          03,X1D,,NNNN
 $       PRMFL          07,R/W,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $       PRMFL          09,R/W,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $       PRMFL          11,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS1
 $       PRMFL          12,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS2
 $       PRMFL          13,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS3
 $       PRMFL          14,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS4
 $       PRMFL          15,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS5
 $       PRMFL          16,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS6
 $       PRMFL          17,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS7
 $       PRMFL          18,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS8
 $       PRMFL          19,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS9
 $       PRMFL          20,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS10
 $       PRMFL          21,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS11
 $       PRMFL          22,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS12
(Data cards)
 $       ENDJOB
***EOF
```

Figure A-3. - Scan routine deck setup using disk pack RSRA1.

```
Col.            Col.                Col.
 1               8                   16
 $              SNUMB
 $              IDENT               112320,EASE
 $              MSG1                1,PLEASE MOUNT DISC PACK RARA1
 $              USERID              RSRATEST$XXXX
 $              OPTION              FORTRAN
(Program deck)'
 $              EXECUTE
 $              LIMITS              10,16K,2K
 $              PRMFL               07,R/W,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $              PRMFL               09,R,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $              ENDJOB
 ***EOF
```

Figure A-4. - Abtask deck setup using disk pack RSRA1.

```
Col.            Col.                Col.
 1               8                   16
 $              SNUMB
 $              IDENT               112320,EASE
 $              MSG1                1,PLEASE MOUNT DISC PACK RSRA1
 $              USERID              RSRATEST$XXXX
 $              OPTION              FORTRAN

(Program deck)
 $              EXECUTE
 $              LIMITS              20,16K,,10K
 $              PRMFL               07,R,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $              PRMFL               09,R,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $              PRMFL               35,R,R,RSRATEST$XXXX/UPCFL
(Data cards)

 $              ENDJOB
 ***EOF
```

Figure A-5. - Statprint deck setup using disk pack RSRA1.

```
Col.    Col.    Col.
 1       8       16
 $      SNUMB
 $      IDENT           112320,EASE
 $      MSG1            1,PLEASE MOUNT DISC PACK RSRA1
 $      USERID          RSRATEST$XXXX
 $      OPTION          FORTRAN

(Program deck)

 $      EXECUTE
 $      LIMITS          50,23K,,2K
 $      PRMFL           07,R/W,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $      PRMFL           09,R/W,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $      PRMFL           19,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS9
 $      PRMFL           20,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS10
 $      PRMFL           23,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS13

(Data cards)

 $      ENDJOB

 ***EOF
```

Figure A-6. - Dptask deck setup using disk pack RSRA1.

```
Col.    Col.    Col.
 1       8       16
 $      SNUMB
 $      IDENT           112320,EASE
 $      MSG1            1,PLEASE MOUNT DISC PACK RSRA
 $      USERID          RSRATEST$XXXX
 $      OPTION          FORTRAN
(Program deck)
 $      EXECUTE
 $      LIMITS          20,17K,,5K
 $      PRMFL           07,R,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $      PRMFL           09,R,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $      PRMFL           35,R,R,RSRATEST$XXXX/UPCFL
(Data cards)
 $      ENDJOB
 ***EOF
```

Figure A-7. - Datask program deck setup using disk pack RSRA1.

```
Col.      Col.              Col.
 1         8                 16
 $        SNUMB
 $        IDENT             112320,EASE
 $        MSG1              1,PLEASE MOUNT DISC PACK RSRA1
 $        USERID            RSRATEST$XXXX
 $        OPTION            FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS            20,18K,,5K
 $        PRMFL             07,R,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $        PRMFL             09,R,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $        PRMFL             11,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS1
 $        PRMFL             12,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS2
 $        PRMFL             13,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS3
 $        PRMFL             14,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS4
 $        PRMFL             15,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS5
 $        PRMFL             16,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS6
 $        PRMFL             17,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS7
 $        PRMFL             18,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS8
(Data cards)
***EOF
```

Figure A-8. - Cycle count deck setup using disc pack RSRA1.

```
Col.      Col.              Col.
 1         8                 16
 $        SNUMB
 $        IDENT             112320,EASE
 $        MSG1              1,PLEASE MOUNT DISC PACK RSRA1
 $        USERID            RSRATEST$XXXX
 $        OPTION            FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS            20,22K,5K
 $        PRMFL             07,R,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $        PRMFL             09,R,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $        PRMFL             19,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS9
 $        PRMFL             21,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS11
 $        PRMFL             22,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS12
 $        PRMFL             23,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS13
 $        TAPE              50,x50D,,,,556-BPI-PLOT-TAPE
 $        SYSOUT            59
(Data cards)
 $        ENDJOB
***EOF
```

Figure A-9. - Time history deck setup using disk pack RSRA1.

```
Col.      Col.         Col.
 1         8            16
 $        SNUMB
 $        IDENT        112320,EASE
 $        MSG1         1,PLEASE MOUNT DISC PACK RSRA1
 $        USERID       RSRATEST$XXXX
 $        OPTION       FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS       20,22K,5K
 $        PRMFL        07,R,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $        PRMFL        24,R,R,RSRATEST$XXXX/FLTA/PCKA/PASS14
 $        PRMFL        35,R,R,RSRATEST$XXXX/UPCFL
(Data cards)
 $        ENDJOB
 ***EOF
```

Figure A-10. - Harmonic analysis deck setup using disk pack RSRA1.

```
Col.      Col.         Col.
 1         8            16
 $        SNUMB
 $        IDENT        112320,EASE
 $        USERID       RSRATEST$XXXX
 $        OPTION       FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS       15,15K,5K
 $        PRMFL        35,R/W,R,RSRATEST$XXXX/UPCFL
 $        FILE         36,x36R,15R
(Data cards)
 $        ENDJOB
 ***EOF
```

Figure A-11. - Upcgen program deck setup.

```
Col.      Col.          Col.
 1         8             16
 $        SNUMB
 $        IDENT         111230,EASE
 $        OPTION        FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS        25,16K,,10K
 $        SYSOUT        06
 $        TAPE9         01,X1D,,NNNN,,INPUT-TAPE
(Data cards)
 $        ENDJOB
```

Figure A-12. - Tpdmp program deck setup.

```
Col.      Col.          Col.
 1         8             16
 $        SNUMB
 $        IDENT         112320,EASE
 $        MSG1          1,PLEASE MOUNT DISC PACK RSRA1
 $        USERID        RSRATEST$XXXX
 $        OPTION        FORTRAN
(Program deck)
 $        EXECUTE
 $        LIMITS        08,16K,,1K
 $        PRMFL         07,R,R,RSRATESTSXXXX/FLTA/PCKA/TBLFL
 $        ENDJOB
```

Figure A-13. - Table print program deck setup.

```
Col.          Col.           Col.
 1             8              16
 $            SNUMB
 $            IDENT          112320,EASE
 $            MSG1           1,PLEASE MOUNT DISC PACK RSRA1
 $            USERID         RSRATEST$XXXX
 $            OPTION         FORTRAN
(Setup routine program deck)
 $            EXECUTE
 $            LIMITS         08,19K,,2K
 $            SYSOUT         08
 $            PRMFL          07,R/Q,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $            FILE           01,AIR,3R
(Data cards)
 $            OPTION         FORTRAN
(Scan routine program deck)
 $            EXECUTE
 $            LIMITS         50,24K,,2K
 $            TAPE9          03,X1D,,NNNN
 $            PRMFL          07,R/W,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $            PRMFL          09,R/W,R,RSRATEST$XXXX/FLTA/PCKA/STATS
 $            PRMFL          11,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS1
 $            PRMFL          12,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS2
 $            PRMFL          13,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS3
 $            PRMFL          14,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS4
 $            PRMFL          15,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS5
 $            PRMFL          16,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS6
 $            PRMFL          17,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS7
 $            PRMFL          18,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS8
 $            PRMFL          19,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS9
 $            PRMFL          20,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS10
 $            PRMFL          21,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS11
 $            PRMFL          22,R/W,R,RSRATEST$XXXX/FLTA/PCKA/PASS12
(Data cards)
 $            OPTION         FORTRAN
(Abtask routine program deck)
 $            EXECUTE
 $            LIMITS         10,16K,2K
 $            PRMFL          07,R/W,R,RSRATEST$XXXX/FLTA/PCKA/TBLFL
 $            ENDJOB
 ***EOF
```

Figure A-14. - Deck setup to execute setup, scan, and abtask.

APPENDIX B

TELEVENT II

Real Time Digital Tape Format Description

ANSI/IBM Compatible

Handler: 9 track

Density: 800 bpi

Parity: Odd

Computer Word Size: 16 bits (msb = bit #16; lsb = #1)

Three types of records are recorded: (1)-Header Record; (2)-Data Description Record; (3)-Data Record.

Header Record

Record Size: 45 words

The header record contains alphanumeric information in EBCDIC and is identified by the first word being equal to zero. The information contained in the header record is obtained during the primary module setup of each data link. One header record is generated for each header card encountered.

Word	Contents
1	000000
2	Number of words/record (45)
3-4	4 char. link ID in EBCDIC
5	Reel Sequence Number
6-45	80 EBCDIC alphanumeric characters

Data Description Record

Record Size: 10 words

The data description record contains the information necessary to describe the data records for a specific link, format, and source. This type of record is identified by the first word containing a -1 in 2's complement form (177777_8). One data description record is present for each type of data record recorded.

Word	Contents
1	-1 (2's complement)
2	Number of words/record (10)
3-4	4 char. link ID in EBCDIC
5 bits 16-13	Zero
bits 12-11	Source ID
	0 - Analog tape
	1 - PCM Simulator
	2 - Manual
bits 10-5	6 least significant bits of first EDCDIC char. of format ID. (alpha)

Word	Contents
bits 4-1	4 least significant bits of second EBCDIC character of format ID. (numeric)
6 bit 16	= 0 for "Software Merged" time = 1 for "Hardware Merged" time
bits 15-8	Zero
bits 7-5	Time identification bits* = 0 - No time is recorded < 4 - Number of "Normal" time words recorded (T_n) > 4 - Bits 6 and 5 indicate the number of "Merged" time words recorded (T_m) = 4 - Time is recorded in "Both" mode.
bits 4-1	Handler number
7	Number of frames/physical record. (F)
8	Number of words/frame. (W)
9	Number of sync words recorded for PCM data links. (S)
10	2 EBCDIC blanks.

Data Record

Record size: Variable

Data records are identified by the first word being greater than zero.

The record size can be calculated by;

$$B = F(Tm + S + W) + Tn + 5$$

where;

B = Physical record size

F = Number of frames/physical record

W = Number of words/frame

S = Number of sync words/frame (PCM only)

Tn = Number of "normal" time words recorded

Tm = Number of "merged" time words recorded

F, W, S, Tn and TM are obtained from the Data Description Record for the specified source, link, and format.

The format of all data records is as follows:

Word	Contents
1	Same as word 5 of corresponding Data Description Record
2	Number of words/physical record (B)
3-4	4 char. link ID in EBCDIC
5	Block count of data records for this source, link, and format.
6 – (B)	→ Real time digital data including any time words and frame sync words

Time Recording Modes

The system has 3 words of time that can be recorded in a variety of ways. The format of these 3 time words is as follows:

Time Word 1	Tenths of Sec.	Hundredths of Sec.	Thousandths of Sec.	Undefined
	16 13	12 9	8 5	4 1

Time Word 2	1	Tens of Min.	Units of Min.	1	Tens of Sec.	Units of Sec.
	16	15 13	12 9	8	7 5	4 1

Time Word 3	Hund. of Days	Tens of Days	Units of Days	Tens of Hr.	Units of Hr.
	16 15	14 11	10 7	6 5	4 1

A. Normal Time

When Normal Time is indicated in bits 7-5 of word 6 of the corresponding Data Description Record, only 1 time tag is supplied for the entire physical record irrespective of the number of frames contained therein. The time tag represents the GMT of when the last data word of the buffer was received. The indicated number of Normal Time words are placed at the end of the physical record with the least significant time word being recorded first.

Tn	Time Words Recorded
1	1
2	1, 2
3	1, 2, 3

B. Merged Time

When bits 7-5 of word 6 of the corresponding Data Description Record indicated 1, 2, or 3 Merged Time words are being recorded, each frame of data within the physical record is time tagged with TM time words as follows:

Tm	Time Words Recorded
1	1
2	1, 2
3	1, 2, 3

The least significant time word is recorded first in each group of time words. Merged Time can be obtained via either hardware or software as indicated in bit 16 of word 6 of the corresponding Data Description Record. For Hardware Merged time, Tm time words are recorded for each frame immediately following the last word of each frame or immediately following the last sync word of each frame if any sync words are present. (Sync words appear only with PCM data where the sync is not being stripped).

For Software Merged time, Tm number of time words are recorded for each frame with all of the time words for all of the frames in a physical record appearing at the end of the buffer beginning immediately after the last word (or sync word) of the last frame.

The time tagging for PCM data corresponds to the last bit of the sync pattern. For the A/D link (AMQ), the time tag is obtained after the last word of the frame has been sampled.

C. Both

When bits 7-5 of word 6 of the corresponding Data Description Record = 100_2, time is recorded in what is referred to as the Both mode. In this mode, 3 words of Normal Time are recorded at the end of the buffer along with 1 word of Merged Time. The time tagging for the two types of time are the same as those previously described.

Frame Sync Pattern

When data from one of the PCM links is being digitized, the operator can request to "strip" the frame sync pattern. Under this condition, no sync words appear on the digital tape and word 9 of the corresponding Data Description Record = 0. When the frame sync pattern is not being stripped, (S) number of sync words as indicated in word 9 of the corresponding Data Description Record appear after the last word of each frame. The number of words/ frame (W) does not include PCM sync pattern words.

APPENDIX C

Derived Parameters

There are certain elements of aircraft performance that cannot be directly instrumented but are functions of one or more instrumented elements of the aircraft. Such elements cannot be processed directly but must be derived based upon their defined algorithms. These elements are processed through the routine "DPTASK" for their point by point derivations as well as the calculation of their statistics.

This appendix lists the parameters that are currently derived through "DPTASK" along with their algorithms. Figure C-1 contains the list of flight constants used in the equations. Figure C-2 lists the definition of the input variables used and Figure C-3 lists the definition of the variables being derived.

The algorithms currently defined in "DPTASK" are as follows:

$$V_{IT} = 1479.K_1 \tag{C 1}$$

where

$$K_1 = \sqrt{(1 + .06805\ V_{IP})^{.2857} - 1.} \tag{C 2}$$

$$V_{CAS} = \text{Linear Interpolation Table Lookup f } (V_{IT}) \tag{C 3}$$

$$V_{EI} = (V_{CAS})(\sqrt{DELTA})(\frac{K_2}{K_1}) \tag{C 4}$$

where

$$DELTA = (1.0 - .000006875H)^{5.256} \tag{C 5}$$

$$K_2 = \sqrt{(1. + \frac{144.\ V_{IP}}{2116.2\ DELTA})^{.2857} - 1.} \tag{C 6}$$

$$F_{AT} = ITAT - .00013177\ KPR(V_{EI}\sqrt{\frac{.0023769}{DENS}})^2 \tag{C 7}$$

where

$$DENS = \frac{2116.2\ DELTA - .378277\ PRESSV}{3089.7\ (ITAT + 273.16)} \tag{C 8}$$

and

$$PRESSV = RELHUM\ [2.685 + .013232(ITAT + 17.78)^{2.245}] \tag{C 9}$$

$$V_T = V_{CAS}\sqrt{\rho_o/\rho} \tag{C10}$$

where

$$\rho_o = .0023769 \tag{C11}$$

and

$$\rho = \frac{2116.2\ DELTA - .378277\ PRESSV_1}{3089.7\ (FAT + 273.16)} \tag{C12}$$

and

$$PRESSV_1 = RELHUM\ [2.685 + 0.13232\ (FAT + 17.78)^{2.245}] \tag{C13}$$

$$TAT = FAT + .000131714\ V_T^2 \tag{C14}$$

$$H_D = \frac{1. - (\frac{\rho}{\rho_o})^{.235}}{.000006875} \qquad (C15)$$

$$\Omega R = \frac{(2\pi)(R)(NR)(MRC)}{(60)(100)} \qquad (C16)$$

$$\Omega TR = \frac{(2\pi)(R_T)(NR)(MRC)(GT)}{(60)(100)} \qquad (C17)$$

$$\mu = \frac{1.6889\ V_T}{\Omega R} \qquad (C18)$$

$$MACH_A = \frac{\Omega R + 1.6889\ V_T}{65.7689\ \sqrt{FAT + 273.16}} \qquad (C19)$$

$$SHP_1 = \frac{(NF_1)(Q_1)(HPC)}{10000.} \qquad (C20)$$

$$SHP_2 = \frac{(NF_2)(Q_2)(HPC)}{10000.} \qquad (C21)$$

$$HP_{TOTAL} = SHP_1 + SHP_2 \qquad (C22)$$

$$CP_{ENG\ TOT} = \frac{550\ HPT}{\rho\ \pi\ R^2 (\Omega R)^3} \qquad (C23)$$

$$HP_{MR} = \frac{(NR)(MRQ)(MRC)}{(5252)(100)} \qquad (C24)$$

$$CP_{MAIN} = \frac{550\ HP_{MR}}{\rho\ \pi\ R^2\ (\Omega\ R)^3} \qquad (C25)$$

$$HP_{TR} = \frac{(NR)(TRQ)(GTS)}{(5252)(100)} \qquad (C26)$$

$$CP_{TAIL} = \frac{550\ HP_{TR}}{\rho\ \pi\ R^2 (\Omega\ R)^3} \qquad (C27)$$

$$HP_S = HP_{MR} + HP_{TR} \qquad (C28)$$

$$CP_{TOT} = \frac{550\ HP_S}{\rho\ \pi\ R^2 (\Omega\ R)^3} \qquad (C29)$$

$$YAW = HEAD - HEAD_1 \qquad (C30)$$

where

$HEAD_1$ is the value of the first sample in data burst.

$$F_X = ISOLB + .2588\ (LIFTD - LIFTA) \tag{C31}$$

$$F_Y + ISOLA - ISOLC + .2588\ (LIFTB - LIFTC) \tag{C32}$$

$$F_Z = .9659\ (LIFTA + LIFTB + LIFTC + LIFTD) \tag{C33}$$

$$M_X = 57.88\ (ISOLA - ISOLC) - 31.98\ (LIFTB + LIFTC) \tag{C34}$$

$$M_Y = -53.\ ISOLB + 32.639\ LIFTA - 2.656\ LIFTB - 37.9515\ LIFTD - 2.656\ LIFTC \tag{C35}$$

$$M_Z = -17.75\ ISOLA - 23.25\ ISOLC + .7118\ LIFTB - LIFTC) - 82.\ QLINK \tag{C36}$$

$$AR = -.0000024A^3 + .0008353A^2 - .2677A + 114.5 \tag{C37}$$

SYMBOL	DEFINITION	VALUE
KPR	Temperature probe recovery factor	.9
RELHUM	Relative humidity	0.
HPC	T58 Engine chaff horsepower @ 100%	1215.69
MRC	Main rotor RPM @ 100% NR	203.
R	Main rotor radius ft	31.
R_T	Tail rotor radius ft	5.3
GT	Gear ratio between tail rotor and main rotor	6.123
GTS	Tail rotor shaft RPM @ 100% NR	3030.

Figure C-1. Flight Constants for Derived Parameters

SYMBOL	MNEMONIC	DEFINITION
ITAT	ITATBOOM	Indicated total air temperature (BOOM) (deg C)
V_{IP}	VIPBOOM	Indicated airspeed (BOOM) (PSID)
H	HBOOM	Pressure altitude (BOOM) (ft)
HEAD	HEAD180	Compass heading ($\pm$ 180 deg)
Q_1	NO1QPCT	No. 1 engine torque %
Q_2	NO2QPCT	No. 2 engine torque %
NF_1	NO1NFPCT	No. 1 engine power turbine speed %
NF_2	NO2NFPCT	No. 2 engine power turbine speed %
MRQ	MRQ1	Main rotor shaft torque ft-lb
TRQ	TRQ	Tail rotor shaft torque ft-lb
NR	NR	Main rotor speed %
LIFTA	MRLIFTA	Main rotor LIFT A (forward) (lb)
LIFTB	MRLIFTB	Main rotor LIFT B (right hand) (lb)
LIFTC	MRLIFTC	Main rotor LIFT C (left hand) (lb)
LIFTD	MRLIFTD	Main rotor LIFT D (aft) (lb)
ISOLA	ISAFAPR	Forward lefthand latitude ISOLATOR
ISOLB	ISOFAPR	Forward longitude ISOLATOR
ISOLC	ISAFBPR	Aft right hand latitude ISOLATOR
QLINK	XMSNSGT	Transit torque link
A	AILPOSR	Right aileron position

Figure C-2. Input Variables for Derived Parameters

SYMBOL	MNEMONIC	DEFINITION
V_{IT}	VITBOOM	True instrumented indicated airspeed (BOOM) (KNOTS)
V_{CAS}	VCASBOOM	Calibrated airspeed (BOOM) (KNOTS)
V_{EI}	VEIBOOM	Equivalent airspeed (BOOM) (KNOTS)
FAT	FATBOOM	Free air temperature (BOOM) (DEGREES C)
V_T	VTBOOM	True airspeed (BOOM) (KNOTS)
TAT	TATBOOM	Total air temperature (BOOM) (DEGREES C)
H_D	HDBOOM	Density altitude (BOOM) (FEET)
Ω_R	OMEGAR	Main rotor tip speed (ft/sec)
Ω_{TR}	OMEGATR	Tail rotor tip speed (ft/sec)
μ	MU	Main rotor tip speed ratio
$MACH_A$	MACHA	Advanced blade tip mach number
SHP_1	NO1SHP	No. 1 engine shaft horsepower
SHP_2	NO2SHP	No. 2 engine shaft horsepower
HP_{TOTAL}	HPT	Total engine horsepower
$CP_{ENG\ TOT}$	CPTE	Total engine power coefficient
HP_{MR}	HPMR	Main rotor horsepower
CP_{MAIN}	CPM	Main rotor power coefficient
HP_{TR}	HPTR	Tail rotor horsepower
CP_{TAIL}	CPTAIL	Tail rotor power coefficient
HP_S	HPS	Total strain gage power
CP_{TOT}	CPT	Total power coefficient
YAW	YAWFTRM	Yaw attitude (DEGREE)
F_X	FX	Main rotor longitudinal force (lb)
F_Y	FY	Main rotor latidudinal force (lb)
F_Z	FZ	Main rotor vertical force (lb)
M_X	MX	Main rotor roll moment (in-lb)
M_Y	MY	Main rotor pitch moment (in-lb)
M_Z	MZ	Main rotor yaw moment (in-lb)
AR	AILPOSP	Right aileron position (%)

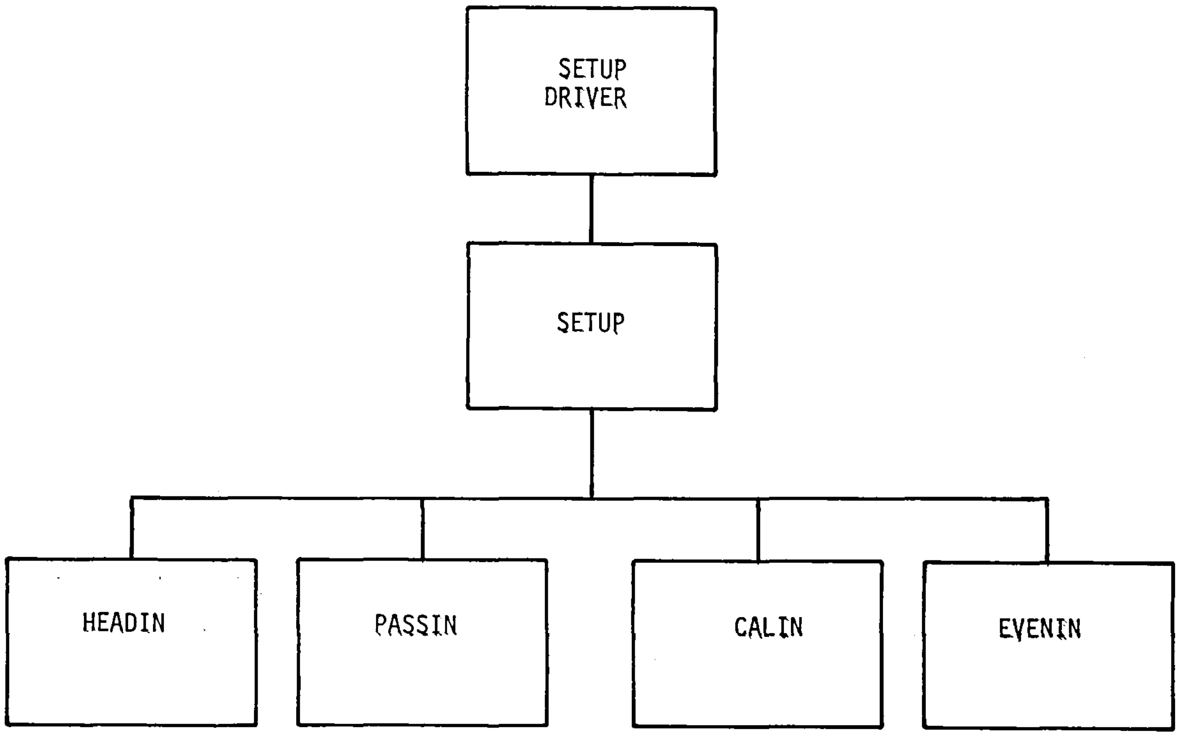
Figure C-3. Derived Parameter Variable Definitions

APPENDIX D

Program Flowcharts

This appendix describes the programming logic used in each independent software task of the EASE system and provides a basic understanding of the methods used to develop each task in the overall operational requirements. A collection of the flowcharts and a listing of each subroutine documentation cards are included here.

A hierarchy chart of each program used in the EASE system precedes the set of subroutine documentation and flowcharts applicable to each respective program.



HIERARCHY CHART for SETUP PROGRAM

*******************MAIN SETUP DRIVER************************************

PROGRAM IDENTIFICATION

PROGRAM NAME ---- MAIN SETUP DRIVER
PROGRAM NUMBER ------- 112320
AUTHOR ------- TERRY D. SOMMERS

COMPUTER ------ HW-625/635
MEMORY ------ ALL FILES OPEN (20K)
PERIPHERALS ----- CARD READER,PRINTER
LANGUAGE ------ HW 6000 FORTRAN/FORTY

PURPOSE

TO CALL APPROPRIATE TASK TO BE GENERATED.

METHOD

CHECK INPUT CARD FOR CORRECT TASK NAME AND OPTION NAME AND CALL APPROPRIATE TASK.

INPUT/OUTPUT

CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
TNAME - INPUT ARGUMENT CONTAINING TASK NAME
OPT - INPUT AND OUTPUT ARGUMENT CONTAINING OPTION NAME
NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF ERRORS COUNTED FROM BAD INPUTS.
JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF CARDS INPUT

CALLING SEQUENCE

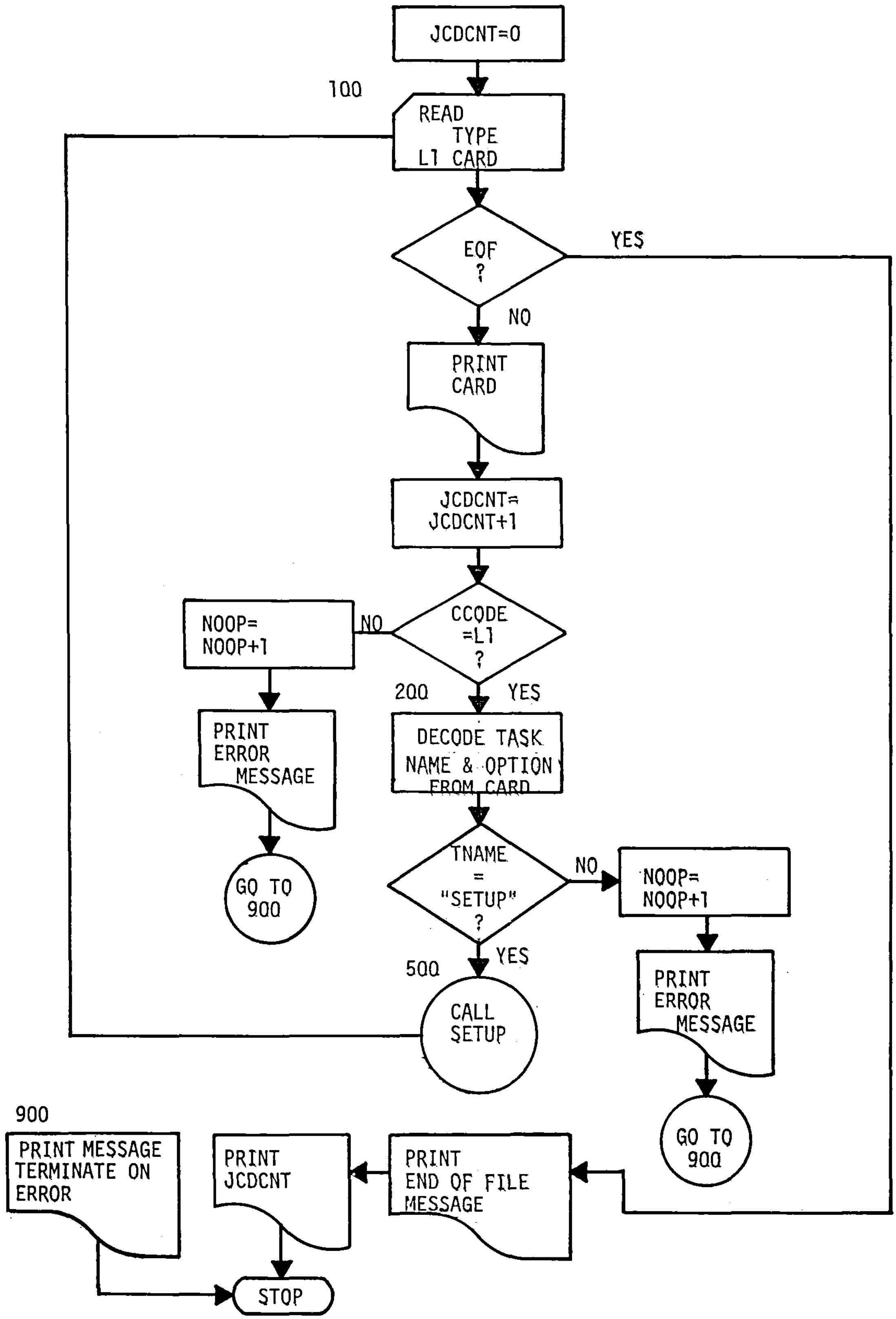
CALL SETUP

COMMON AREAS

DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT

SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.

****


JCDCNT=0
100
READ TYPE L1 CARD
EOF ?
YES
NO
PRINT CARD
JCDCNT= JCDCNT+1
CCODE =L1 ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 900
200
YES
DECODE TASK NAME & OPTION FROM CARD
TNAME = "SETUP" ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 900
500
YES
CALL SETUP
900
PRINT MESSAGE TERMINATE ON ERROR
PRINT JCDCNT
PRINT END OF FILE MESSAGE
STOP

```
********************SUBROUTINE SETUP**********************************

 PROGRAM IDENTIFICATION

     PROGRAM NAME ----        SUBROUTINE SETUP
     PROGRAM NUMBER ------- 112320
     AUTHOR -------    TERRY D. SOMMERS

     COMPUTER ------          HW-625/635
     MEMORY ------
     PERIPHERALS ----------- CARD READER,PRINTER,DISC,TAPE
     LANGUAGE ------   HW 6000 FORTRAN/FORTY

 PURPOSE

     TO INITIALIZE OR UPDATE A TABLE FILE WHICH WILL
        BE USED BY OTHER RSRA ROUTINES

 METHOD

     CHECK OPTION FOR "INITIALIZE" OR "UPDATE" AND EITHER
        ZERO OLD TABLES OR READ CURRENT TABLES FROM DISC
        INTO MEMORY. READ A CARD AND CHECK CARD CODE TO
        DETERMINE ANY OR ALL TABLE ROUTINES TO BE USED.
        WRITE NEW OR UPDATED TABLES TO DISC FILE 07 AND
        RETURN TO DRIVER ROUTINE.  TABLES WILL ONLY BE SAVED IF
        NOOP VALUE IS 0.

 INPUT/OUTPUT

     CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
     CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
     TNAME -  INPUT ARGUMENT CONTAINING TASK NAME
     OPT - INPUT AND OUTPUT ARGUMENT CONTAINING OPTION NAME
     NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF
            ERRORS COUNTED FROM BAD INPUTS.
     JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF
              CARDS INPUT
     THEAD - INPUT OR OUTPUT ARGUMENT CONTAINING HEADER TABLE
             INFORMATION
     TPASS - INPUT OR OUTPUT ARGUMENT CONTAINING PASS TABLE
             INFORMATION
     TSENSE - INPUT OR OUTPUT ARGUMENT CONTAINING SENSOR TABLE
              INFORMATION
     TEVENT - INPUT OR OUTPUT ARGUMENT CONTAINING EVENT TABLE
              INFORMATION
     ICALNO - INPUT OR OUTPUT ARGUMENT CONTAINING CALIBRATION
              TABLE INFORMATION
     SCTAB - INPUT OR OUTPUT ARGUMENT CONTAINING SENSOR TABLE
             INFORMATION FOR SCRATCH TAPE FILE 01
     TABREC - ARGUMENT CONTAINING A TABLE RECORD  600 WORDS/REC
     NTBL - ARGUMENT CONTAINING NAME OF TABLE TO BE
            INITIALIZED OR UPDATED
     FILE 07 - RANDOM DISC FILE FOR TABLES
     FILE 01 - RANDOM SCRATCH TAPE FILE FOR SENSOR TABLES ONLY
                         520 WORDS/REC.
```

CALLING SEQUENCE

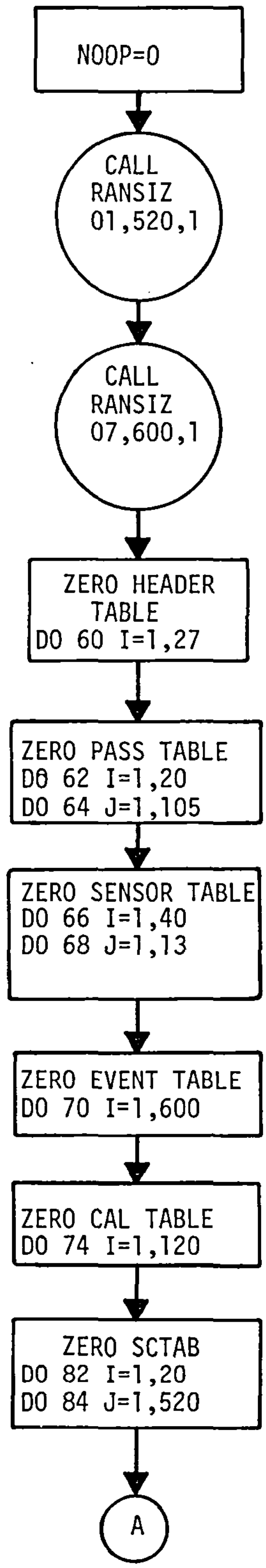
```
CALL HEADIN
CALL PASSIN
CALL EVENIN
CALL CALIN
```

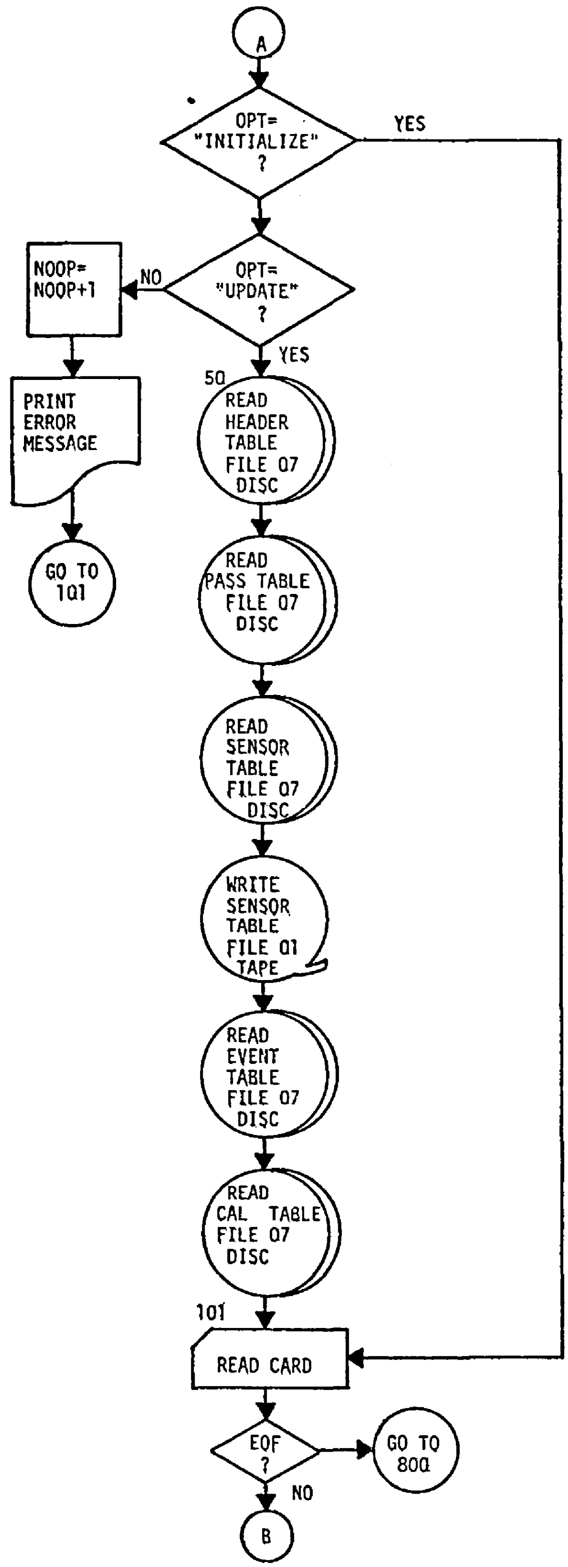
COMMON AREAS

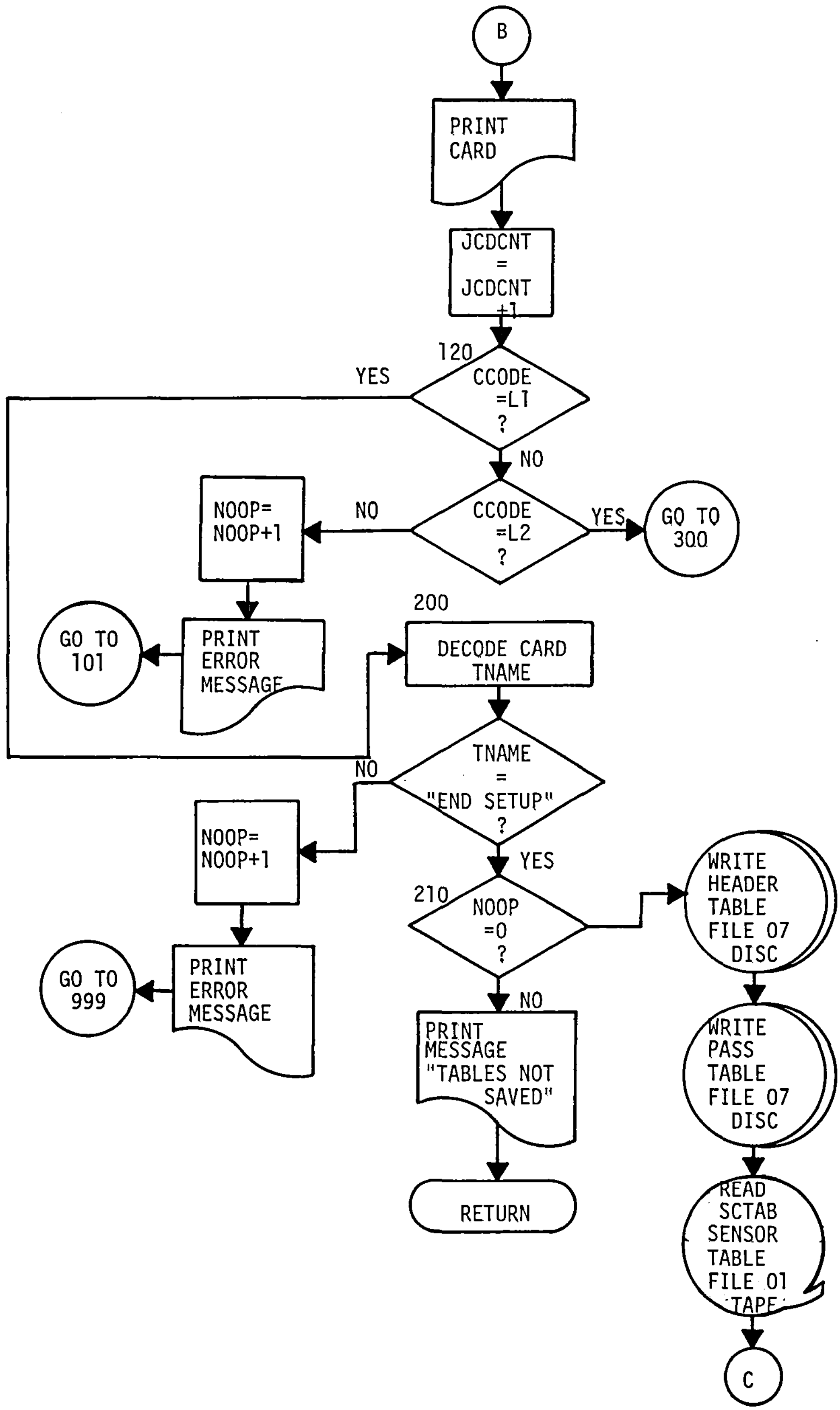
```
DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT
 TABL1 - THEAD
 TABL2 - TPASS
 TABL3 - TSENSE
 TABL4 - TEVENT
 TABL5 - ICALNO
```

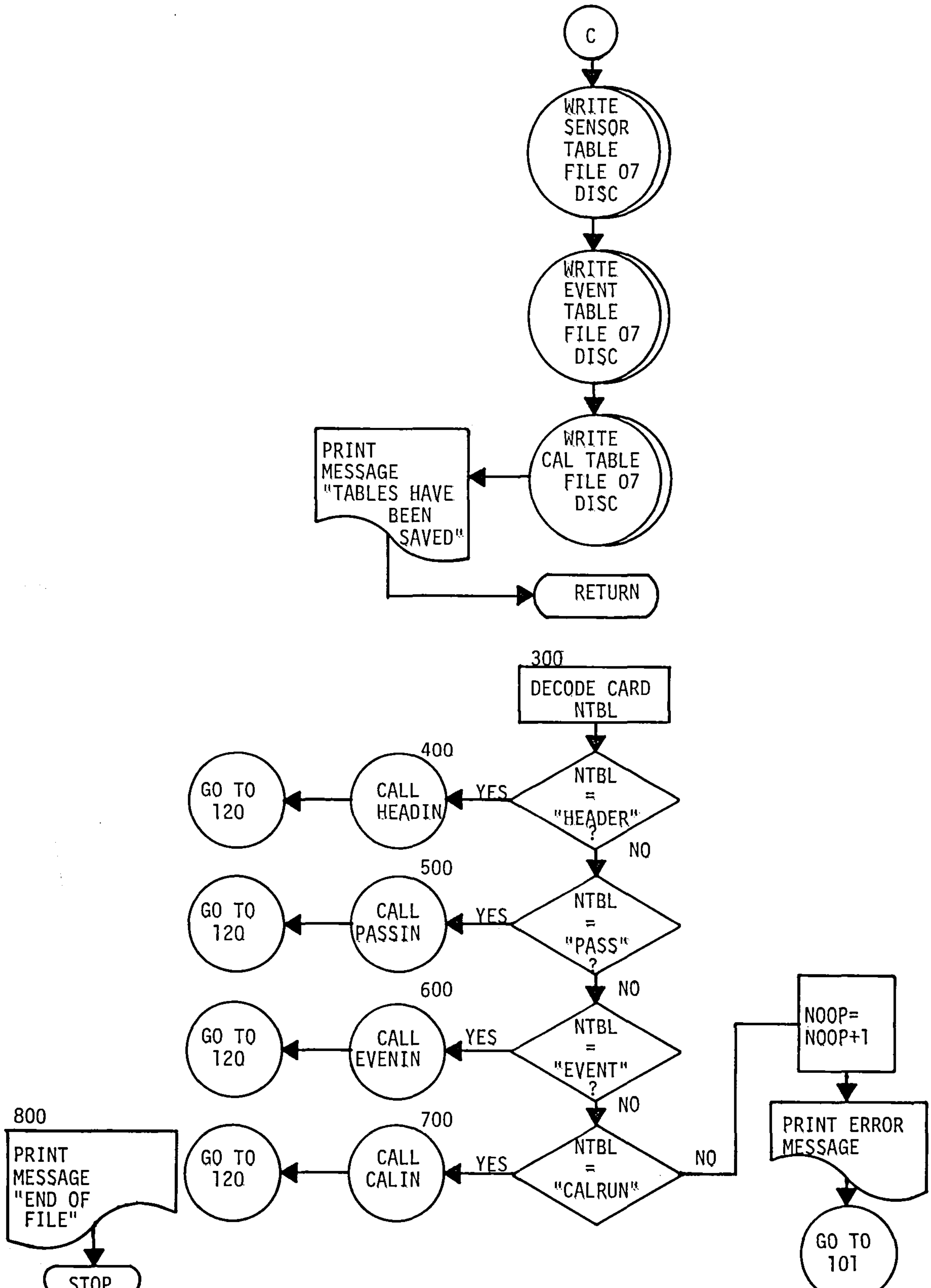
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.

****


NOOP=0
CALL RANSIZ 01,520,1
CALL RANSIZ 07,600,1
ZERO HEADER TABLE
DO 60 I=1,27
ZERO PASS TABLE
D0 62 I=1,20
DO 64 J=1,105
ZERO SENSOR TABLE
DO 66 I=1,40
DO 68 J=1,13
ZERO EVENT TABLE
DO 70 I=1,600
ZERO CAL TABLE
DO 74 I=1,120
ZERO SCTAB
DO 82 I=1,20
DO 84 J=1,520
A


A
OPT=
"INITIALIZE"
?
YES
OPT=
"UPDATE"
?
NO
NOOP=
NOOP+1
PRINT
ERROR
MESSAGE
GO TO
101
YES
50
READ
HEADER
TABLE
FILE 07
DISC
READ
PASS TABLE
FILE 07
DISC
READ
SENSOR
TABLE
FILE 07
DISC
WRITE
SENSOR
TABLE
FILE 01
TAPE
READ
EVENT
TABLE
FILE 07
DISC
READ
CAL TABLE
FILE 07
DISC
101
READ CARD
EOF
?
GO TO
800
NO
B

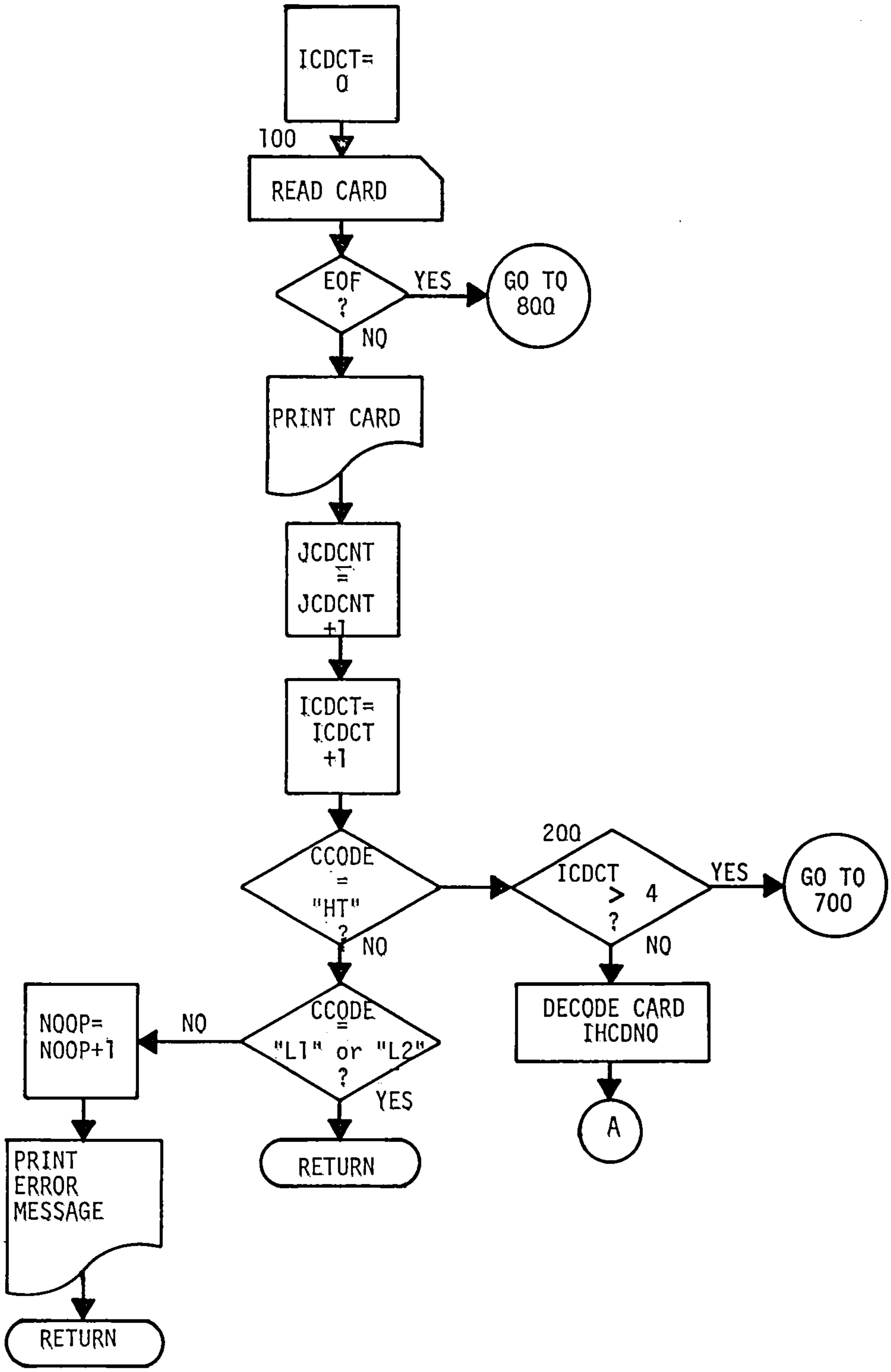

B
PRINT CARD
JCDCNT = JCDCNT +1
120
CCODE =L1 ?
YES
NO
CCODE =L2 ?
NO
YES
GO TO 300
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 101
200
DECODE CARD TNAME
TNAME = "END SETUP" ?
NO
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 999
210
NOOP =0 ?
NO
PRINT MESSAGE "TABLES NOT SAVED"
RETURN
WRITE HEADER TABLE FILE 07 DISC
WRITE PASS TABLE FILE 07 DISC
READ SCTAB SENSOR TABLE FILE 01 TAPE
C

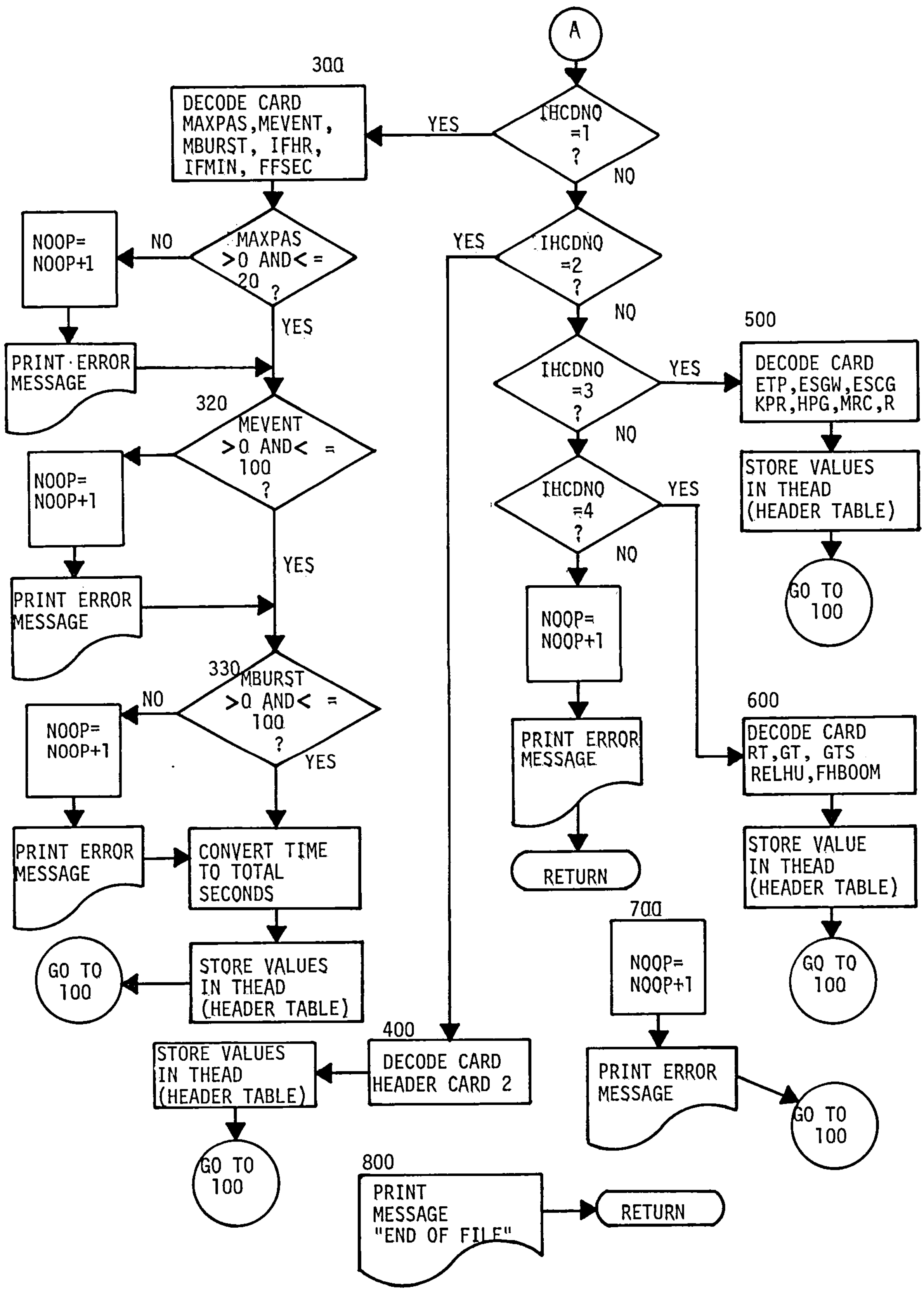


C
WRITE SENSOR TABLE FILE 07 DISC
WRITE EVENT TABLE FILE 07 DISC
WRITE CAL TABLE FILE 07 DISC
PRINT MESSAGE "TABLES HAVE BEEN SAVED"
RETURN
300
DECODE CARD NTBL
NTBL = "HEADER" ?
YES
NO
400
CALL HEADIN
GO TO 120
NTBL = "PASS" ?
500
CALL PASSIN
GO TO 120
NTBL = "EVENT" ?
600
CALL EVENIN
GO TO 120
NTBL = "CALRUN"
700
CALL CALIN
GO TO 120
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 101
800
PRINT MESSAGE "END OF FILE"
STOP

```
********************SUBROUTINE HEADIN*********************************

 PROGRAM IDENTIFICATION

     PROGRAM NAME ----        SUBROUTINE HEADIN
     PROGRAM NUMBER ------- 112320
     AUTHOR -------    TERRY D. SOMMERS

     COMPUTER ------          HW-625/635
     MEMORY ------
     PERIPHERALS -----        CARD READER,PRINTER
     LANGUAGE ------   HW 6000 FORTRAN/FORTY

 PURPOSE

     TO BUILD OR UPDATE HEADER TABLE

 METHOD

     CHECK HEADER CARD NUMBER AND DECODE ACCORDINGLY.  EDIT
        CARD PARAMETERS AND STORE INFORMATION IN HEADER COMMON
        AREA TO BE WRITTEN IN SETUP ROUTINE.

 INPUT/OUTPUT

     CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
     CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
     NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF
            ERRORS COUNTED FROM BAD INPUTS.
     JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF
              CARDS INPUT
     THEAD - HEADER TABLE NAME OR ARRAY
     ICDCT - COUNTER FOR NUMBER OF HEADER CARDS READ
     IHCDNO - INPUT ARGUMENT CONTAINING HEADER CARD NUMBER
  HEADER CARD NO. 1
     MAXPAS - MAXIMUM NUMBER OF PASSES DURING FLIGHT
     MEVENT - MAXIMUM NUMBER OF EVENTS DURING FLIGHT
     MBURST - MAXIMUM NUMBER OF BURSTS FOR 6130 TAPE DURING FLIGHT
     FTIME - BEGINNING TIME OF FLIGHT
  HEADER CARD NO. 2
     FLIGHT - FLIGHT INFORMATION
     DATE - DATE OF FLIGHT (DDMMYY)
     TAPENO - TAPE NUMBER OF 6130 TAPE
     AIRCR1 + AIRCR2 - AIRCRAFT NAME
     PILOT1 + PILOT2 - PILOT NAME
     COPIL1 + COPIL2 - COPILOT NAME
     OBSER1 + OBSER2 - OBSERVER NAME
  HEADER CARD NO. 3
        ETP - ENGINEERING TEST PLAN
        ESGW - ENGINEERING GROSS WEIGHT
        ESCG - ENGINEERING CENTER GRAVITY
        KPR(0.9) - AIRSPEED PROBE RECOVERY FACTOR
        HPC(1215.69) - T-58 ENGINE CHAFF(HP AT 100 PERCENT)
        MRC(203) - MAIN ROTOR (RPM AT 100 PERCENT)
        R(31) - MAIN ROTOR RADIUS(FEET)
```

```
 HEADER CARD NO. 4
       RT(5.3) - TAIL ROTOR RADIUS(FEET)
       GT(6.123) - TAIL ROTOR TO MAIN ROTOR(RPM RATIO)
       GTS(3030) - TAIL ROTOR SHAFT(RPM)
       RELHU(0) - RELATIVE HUMIDITY(UNUSED)
       FHBOOM - UNUSED

COMMON AREAS

    DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT
     TABL1 - THEAD

       SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS
     AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.

************************************************************************
```


ICDCT=
0
100
READ CARD
EOF
?
YES
GO TO
800
NO
PRINT CARD
JCDCNT
=
JCDCNT
+1
ICDCT=
ICDCT
+1
CCODE
=
"HT"
?
NO
200
ICDCT
> 4
?
YES
GO TO
700
NO
DECODE CARD
IHCDNO
A
NOOP=
NOOP+1
NO
CCODE
=
"L1" or "L2"
?
YES
RETURN
PRINT
ERROR
MESSAGE
RETURN


A
IHCDNO =1 ?
YES
NO
300
DECODE CARD MAXPAS,MEVENT, MBURST, IFHR, IFMIN, FFSEC
MAXPAS >0 AND< = 20 ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
320
MEVENT >0 AND< = 100 ?
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
330
MBURST >0 AND< = 100 ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
CONVERT TIME TO TOTAL SECONDS
STORE VALUES IN THEAD (HEADER TABLE)
GO TO 100
IHCDNO =2 ?
YES
NO
400
DECODE CARD HEADER CARD 2
STORE VALUES IN THEAD (HEADER TABLE)
GO TO 100
IHCDNO =3 ?
YES
NO
500
DECODE CARD ETP,ESGW,ESCG KPR,HPG,MRC,R
STORE VALUES IN THEAD (HEADER TABLE)
GO TO 100
IHCDNO =4 ?
YES
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
RETURN
600
DECODE CARD RT,GT, GTS RELHU,FHBOOM
STORE VALUE IN THEAD (HEADER TABLE)
GO TO 100
700
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 100
800
PRINT MESSAGE "END OF FILE"
RETURN

********************SUBROUTINE PASSIN*************************************

PROGRAM IDENTIFICATION

PROGRAM NAME ---- SUBROUTINE PASSIN
PROGRAM NUMBER ------- 112320
AUTHOR ------- TERRY D. SOMMERS

COMPUTER ------ HW-625/635
MEMORY ------
PERIPHERALS -------- CARD READER, PRINTER, TAPE
LANGUAGE ------ HW 6000 FORTRAN/FORTY

PURPOSE

TO BUILD OR UPDATE PASS TABLES AND SENSOR TABLES AND TO SEARCH TABLES FOR DUPLICATION OF SENSOR NAME, PREPROCESSING CODE AND SAMPLE RATE

METHOD

DECODE INFORMATION IN PASS CARD AND STORE IN PASS TABLES. CHECK FOR SENSOR CARDS AND DECODE. EDIT CARD PARAMETERS FOR CORRECT INFORMATION. STORE VALUES IN SENSOR TABLES. WRITE SCRATCH FILE 01. READ SCRATCH FILE AND SEARCH FOR DUPLICATION OF SENSOR NAME, PREPROCESSING CODE AND SAMPLE RATE.

INPUT/OUTPUT

CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF ERRORS COUNTED FROM BAD INPUTS.
JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF CARDS INPUT
IPSNO - PASS NUMBER
IRDIN - INTEGRAL VALUE OF 6130 TAPE SAMPLE RATE
RDIN - 6130 SAMPLE RATE TYPE(X,M,T ,)
IRDATA - INTEGRAL VALUE OF DATA FILE SAMPLE RATE
RDATA - DATA FILE SAMPLE RATE TYPE(X,M,T,)
ISTATC - STATUS CODE FOR DATA AVAILABILITY
0 = NOT AVAILABLE
1 = AVAILABLE
2 = AB TASK RUN
3 = DERIVED PARAMETER PASS
MAXPAR - ACTUAL NUMBER OF PARAMETERS FOR PASS
TPASS - PASS TABLE ARRAY
SRDIN - COMBINED INTEGRAL VALUE AND TYPE FOR SAMPLE RATE
SRDATA - COMBINED INTEGRAL VALUE AND TYPE FOR SAMPLE RATE
SCTAB - SENSOR TABLE ARRAY FOR SCRATCH FILE
TSENSE - SENSOR TABLE ARRAY
IDRATE - INCREMENT VALUE OF 6130 SAMPLE RATE
SNC1 - FIRST CHARACTER OF SENSOR NAME

ST CARD

SNAM1 + SNAM2 - SENSOR NAME

PPC - PREPROCESSING CODE

V = VIBRATORY

S = STEADY

D = DIRECT

IPARIN - 6130 TAPE PARAMETER LOCATION

IPARDF - DATA FILE PARAMETER LOCATION

ITRACK - 6130 TAPE TRACK NUMBER

ICHAN - 6130 TAPE CHANNEL NUMBER

CALTYP - CALIBRATION TYPE(Z,X,P,1,2)

EUCV1 - FACTORY CAL "A" TERM

EUCV2 - FACTORY CAL "B" TERM

FILNO - NOT USED

CONLEV - CONCERN LEVEL VALUE

COMMON AREAS

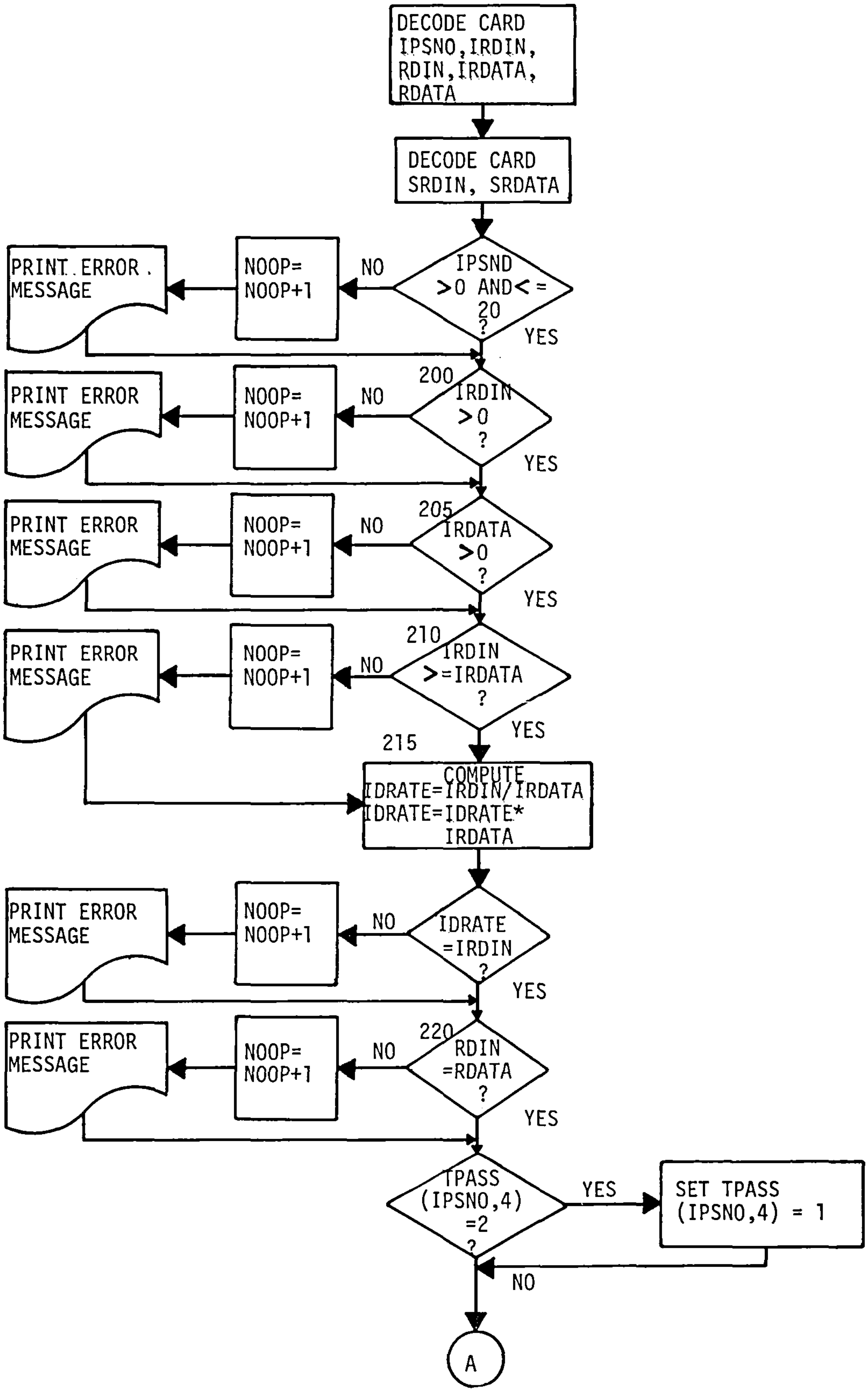
DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT

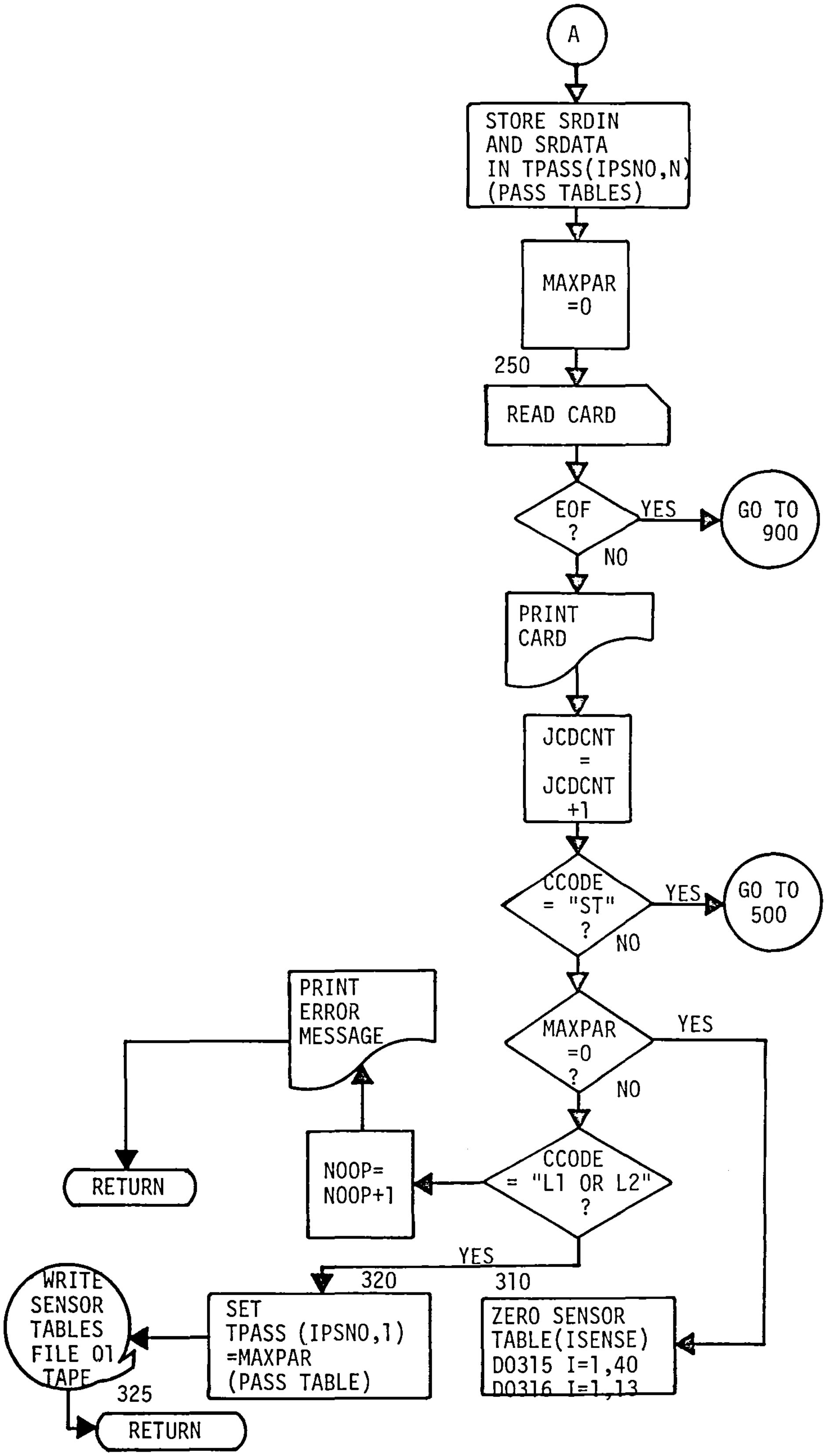
TABL2 - TPASS

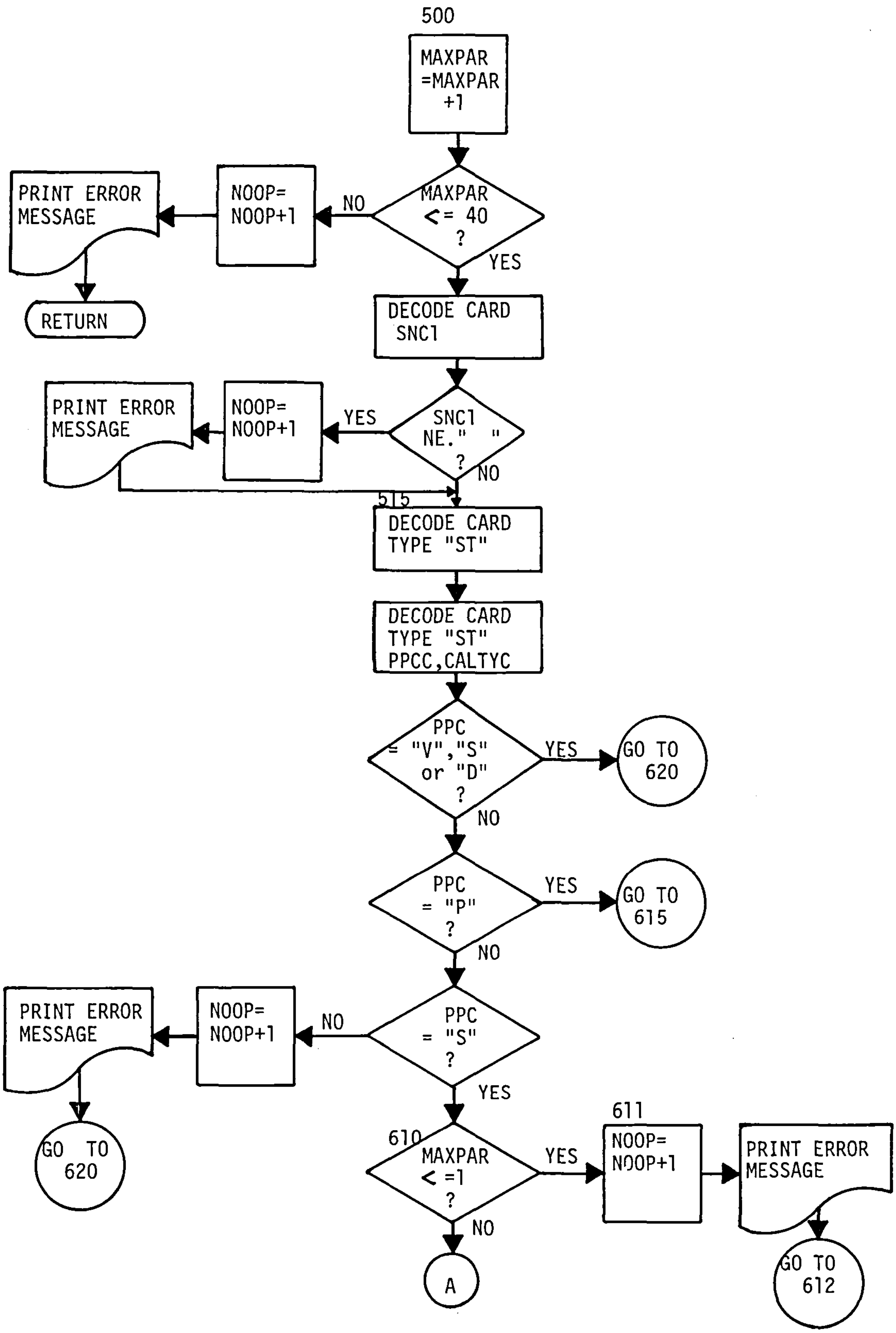
TABL3 - TSENSE

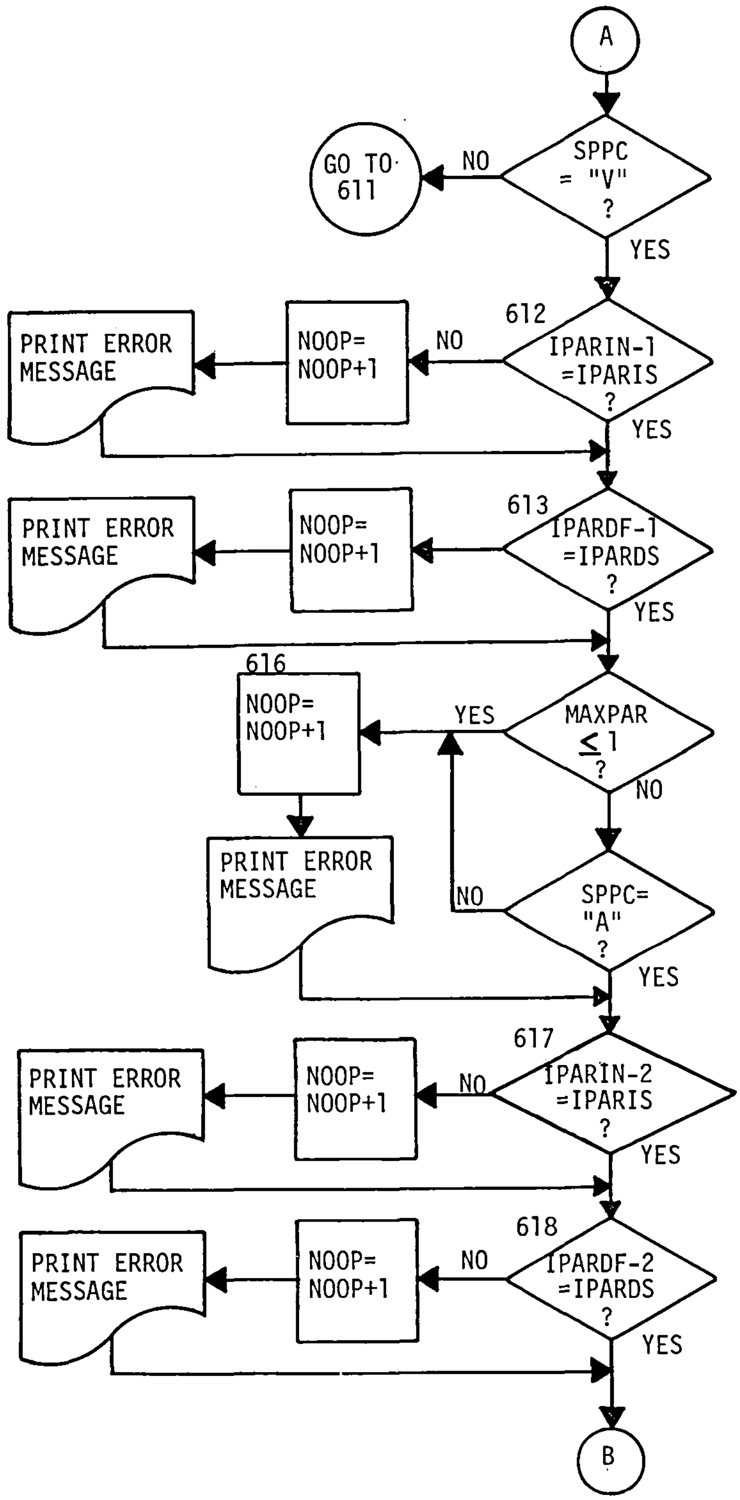
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.

****

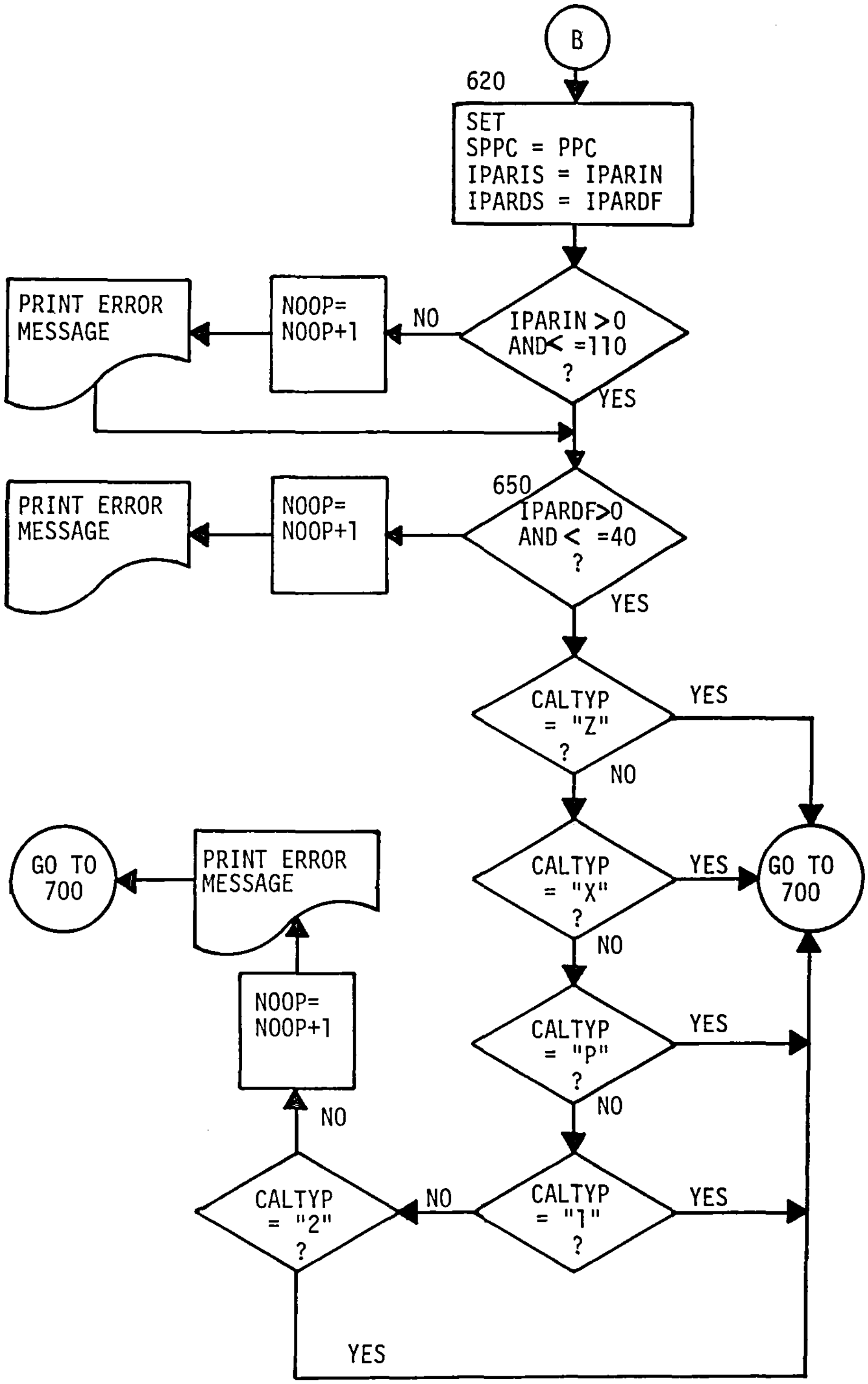

DECODE CARD IPSNO,IRDIN, RDIN,IRDATA, RDATA
DECODE CARD SRDIN, SRDATA
IPSND >0 AND<= 20 ?
NO
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
200
IRDIN >0 ?
205
IRDATA >0 ?
210
IRDIN >=IRDATA ?
215
COMPUTE IDRATE=IRDIN/IRDATA IDRATE=IDRATE* IRDATA
IDRATE =IRDIN ?
220
RDIN =RDATA ?
TPASS (IPSNO,4) =2 ?
SET TPASS (IPSNO,4) = 1
A

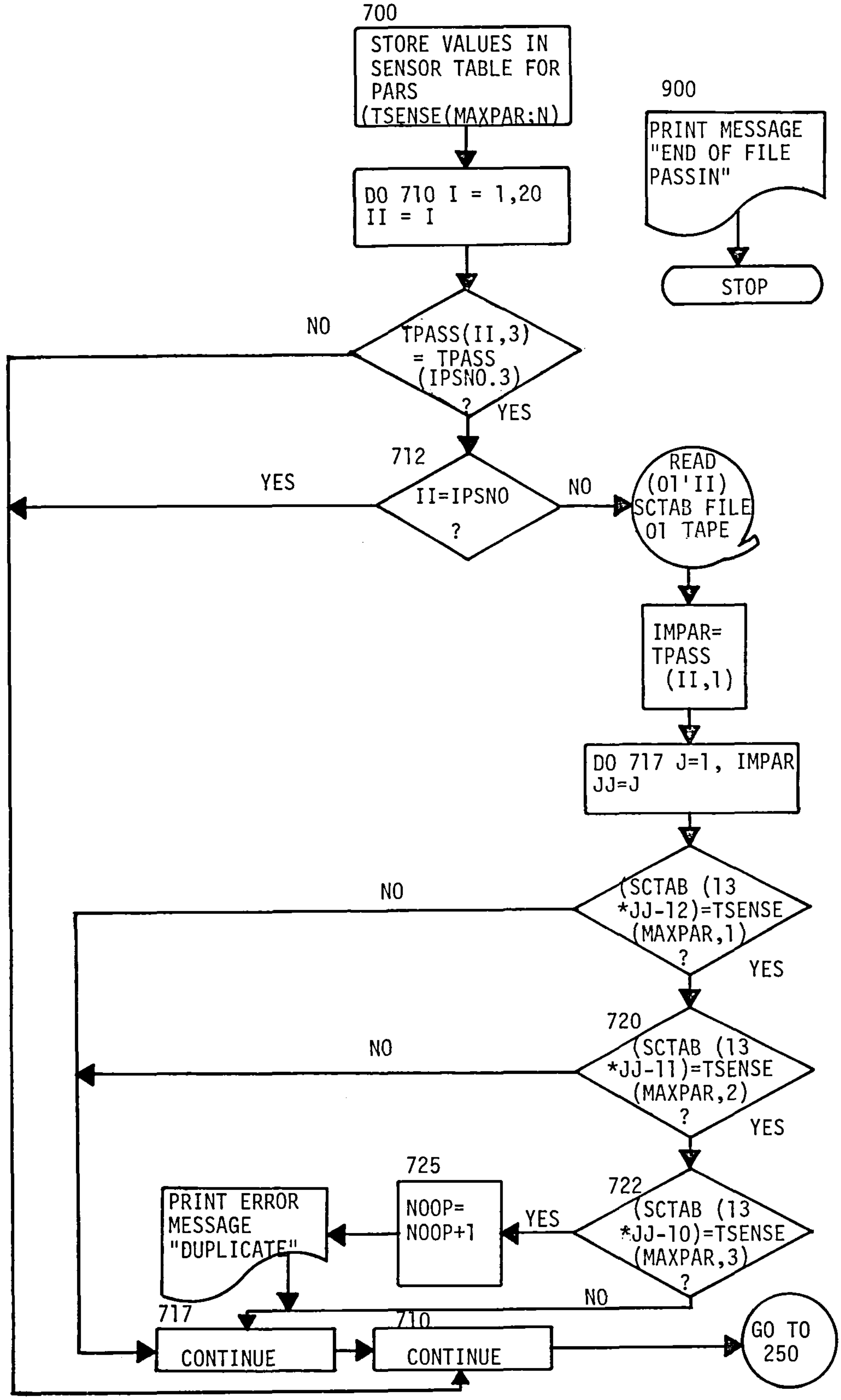

A
STORE SRDIN AND SRDATA IN TPASS(IPSNO,N) (PASS TABLES)
MAXPAR =0
250
READ CARD
EOF ?
YES
GO TO 900
NO
PRINT CARD
JCDCNT = JCDCNT +1
CCODE = "ST" ?
YES
GO TO 500
NO
MAXPAR =0 ?
YES
NO
PRINT ERROR MESSAGE
RETURN
NOOP= NOOP+1
CCODE = "L1 OR L2" ?
YES
320
310
WRITE SENSOR TABLES FILE 01 TAPE
SET TPASS (IPSNO,1) =MAXPAR (PASS TABLE)
ZERO SENSOR TABLE(ISENSE) DO315 I=1,40 DO316 I=1,13
325
RETURN


500
MAXPAR =MAXPAR +1
MAXPAR <= 40 ?
NO
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
RETURN
DECODE CARD SNC1
SNC1 NE." " ?
YES
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
515
DECODE CARD TYPE "ST"
DECODE CARD TYPE "ST" PPCC,CALTYC
PPC = "V","S" or "D" ?
YES
GO TO 620
NO
PPC = "P" ?
YES
GO TO 615
NO
PPC = "S" ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 620
YES
610
MAXPAR < =1 ?
YES
611
NOOP= NOOP+1
PRINT ERROR MESSAGE
GO TO 612
NO
A



A
SPPC = "V" ?
NO
GO TO 611
YES
612
IPARIN-1 =IPARIS ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
613
IPARDF-1 =IPARDS ?
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
616
MAXPAR ≤ 1 ?
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
NO
SPPC= "A" ?
NO
YES
617
IPARIN-2 =IPARIS ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
618
IPARDF-2 =IPARDS ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
B


B
620
SET
SPPC = PPC
IPARIS = IPARIN
IPARDS = IPARDF
IPARIN >0
AND< =110
?
NO
YES
NOOP=
NOOP+1
PRINT ERROR
MESSAGE
650
IPARDF>0
AND< =40
?
YES
NOOP=
NOOP+1
PRINT ERROR
MESSAGE
CALTYP
= "Z"
?
YES
NO
CALTYP
= "X"
?
YES
NO
GO TO
700
CALTYP
= "P"
?
YES
NO
CALTYP
= "1"
?
YES
NO
CALTYP
= "2"
?
NO
YES
NOOP=
NOOP+1
PRINT ERROR
MESSAGE
GO TO
700


700
STORE VALUES IN SENSOR TABLE FOR PARS (TSENSE(MAXPAR;N)
900
PRINT MESSAGE "END OF FILE PASSIN"
STOP
DO 710 I = 1,20
II = I
NO
TPASS(II,3) = TPASS (IPSNO.3) ?
YES
712
YES
II=IPSNO ?
NO
READ (01'II) SCTAB FILE 01 TAPE
IMPAR= TPASS (II,1)
DO 717 J=1, IMPAR
JJ=J
NO
(SCTAB (13 *JJ-12)=TSENSE (MAXPAR,1) ?
YES
720
NO
(SCTAB (13 *JJ-11)=TSENSE (MAXPAR,2) ?
YES
725
722
PRINT ERROR MESSAGE "DUPLICATE"
NOOP= NOOP+1
YES
(SCTAB (13 *JJ-10)=TSENSE (MAXPAR,3) ?
NO
717
710
CONTINUE
CONTINUE
GO TO 250

```
********************SUBROUTINE CALIN********************************
```

PROGRAM IDENTIFICATION

```
PROGRAM NAME ----        SUBROUTINE CALIN
PROGRAM NUMBER ------- 112320
AUTHOR -------    TERRY D. SOMMERS

COMPUTER ------          HW-625/635
MEMORY ------
PERIPHERALS ------       CARD READER,PRINTER,TAPE
LANGUAGE ------   HW 6000 FORTRAN/FORTY
```

PURPOSE

TO BUILD OR UPDATE CALIBRATION TABLES. TO SCAN SENSOR TABLES FOR MATCH OF SENSOR NAMES.

METHOD

CHECK CARD FOR CALIBRATION TYPE, DECODE CARD AND EDIT PARAMETERS. SCAN SENSOR TABLES FOR MATCH OF SENSOR NAMES SET CAL POINTER TO CORRECT VALUE. STORE CALIBRATION TABLE IN ARRAY TO BE WRITTEN IN SETUP ROUTINE. WRITE CORRECTED SENSOR TABLES ON SCRATCH FILE 01.

INPUT/OUTPUT

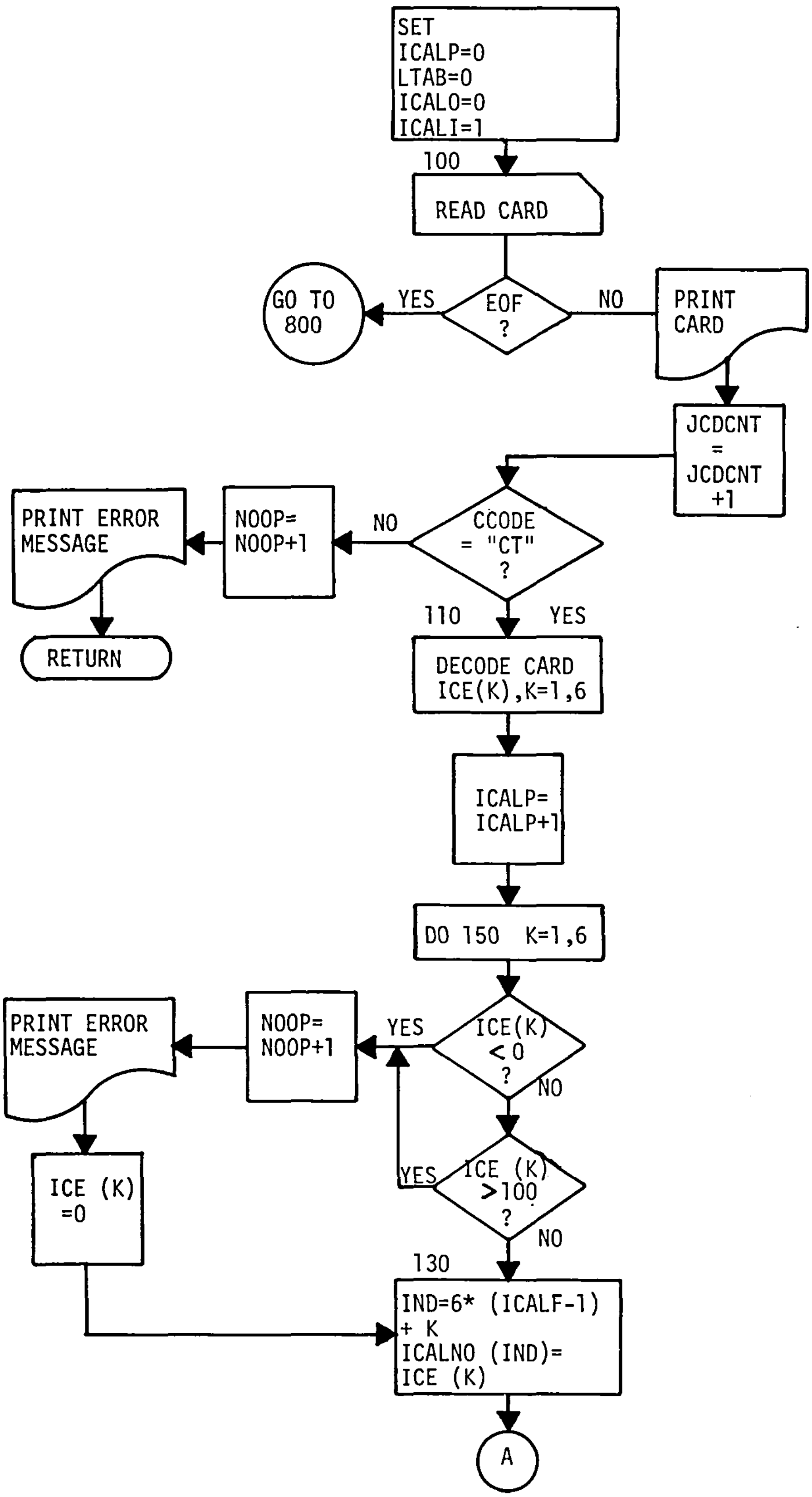
```
CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF
       ERRORS COUNTED FROM BAD INPUTS.
JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF
         CARDS INPUT
         CARDS INPUT
LTAB - COUNTER FOR INDEXING
ICALP - COUNTER FOR INDEXING
ICAL0 - CALIBRATION POINTER 0
ICAL1 - CALIBRATION POINTER 1
ICALNO - CALIBRATION TABLE ARRAY
NAMES - SENSOR NAME
ICE - PRE OR POST CAL EVENT NUMBER
SENTAB - SENSOR TABLE ARRAY (SCRATCH FILE 01)
CALTC - CALIBRATION TYPE
```

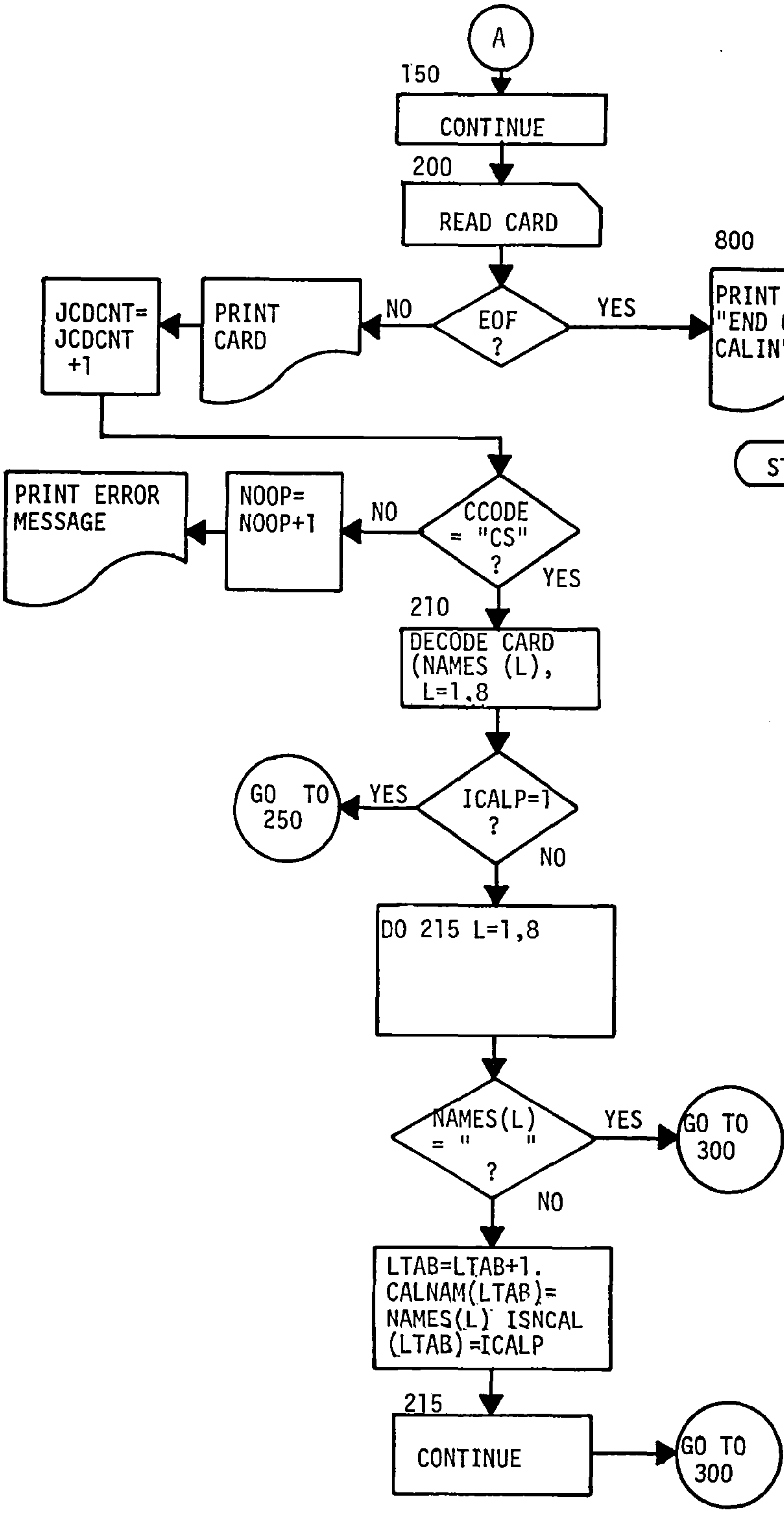
COMMON AREAS

```
DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT
TABL1 - THEAD
TABL2 - TPASS
TABL5 - ICALNO
```

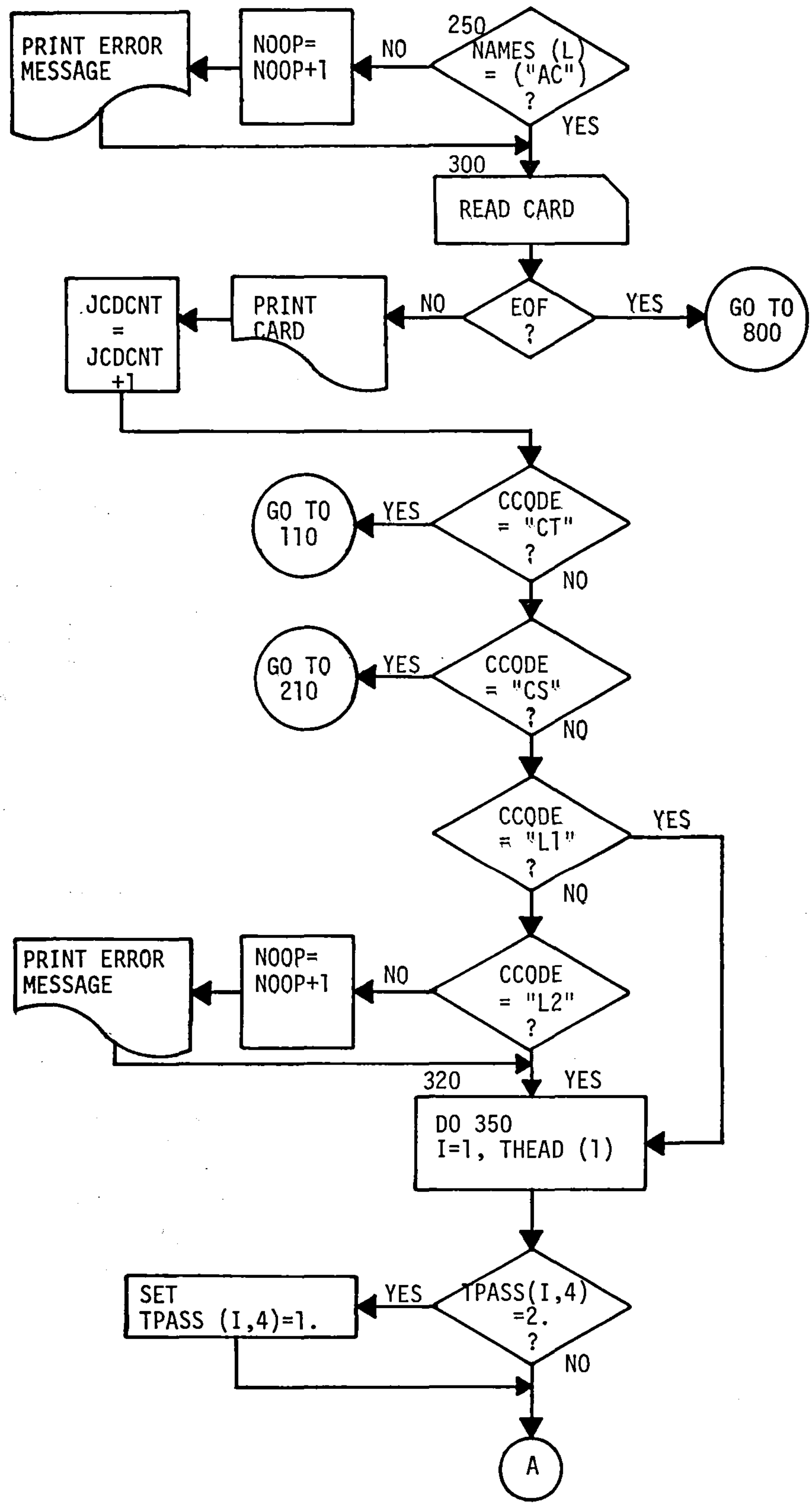
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.

```
*******************************************************************
```

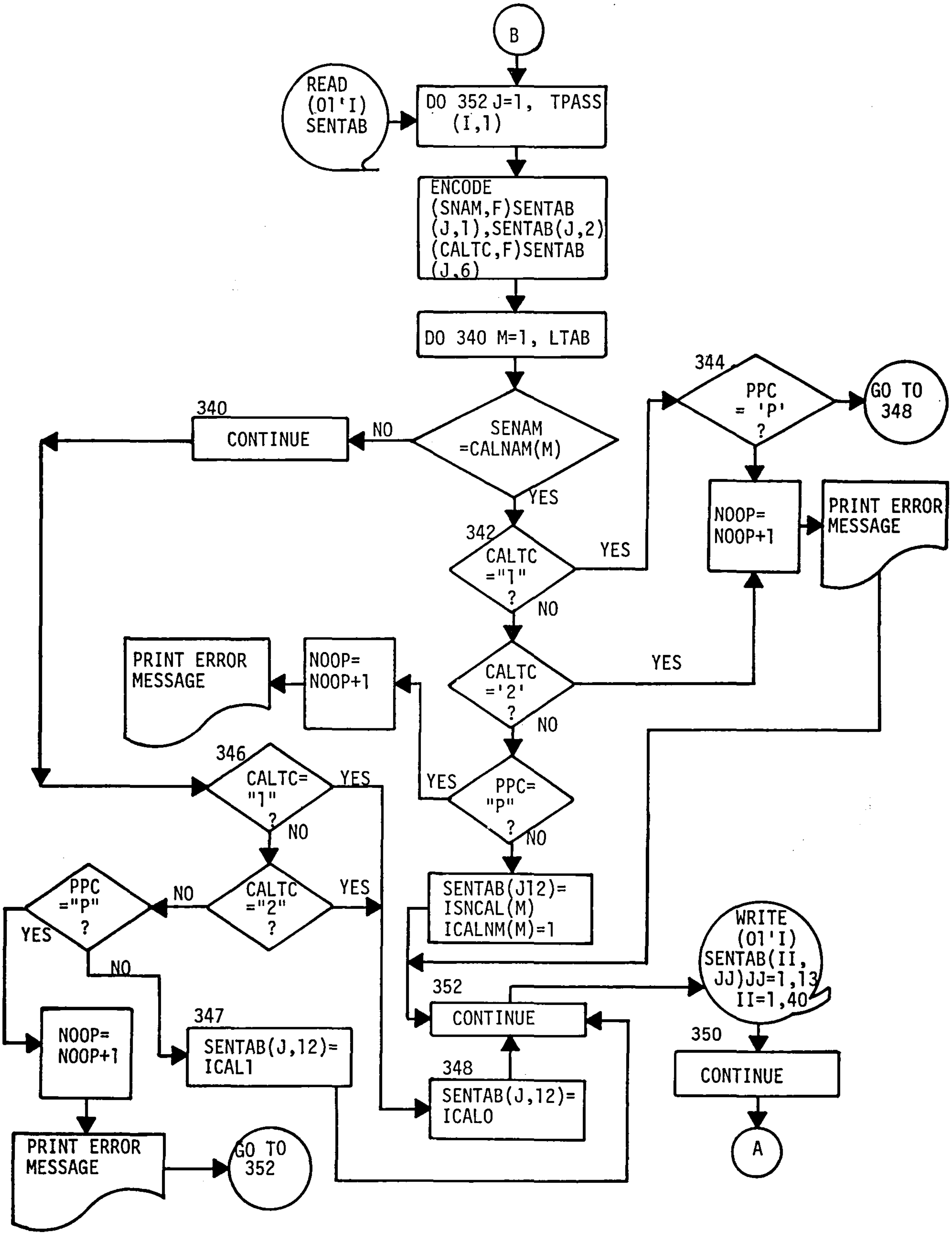

SET
ICALP=0
LTAB=0
ICALO=0
ICALI=1
100
READ CARD
GO TO 800
YES
EOF ?
NO
PRINT CARD
JCDCNT = JCDCNT +1
PRINT ERROR MESSAGE
NOOP= NOOP+1
NO
CCODE = "CT" ?
RETURN
110
YES
DECODE CARD ICE(K),K=1,6
ICALP= ICALP+1
DO 150 K=1,6
PRINT ERROR MESSAGE
NOOP= NOOP+1
YES
ICE(K) < 0 ?
NO
ICE (K) =0
YES
ICE (K) >100 ?
NO
130
IND=6* (ICALF-1) + K
ICALNO (IND)= ICE (K)
A

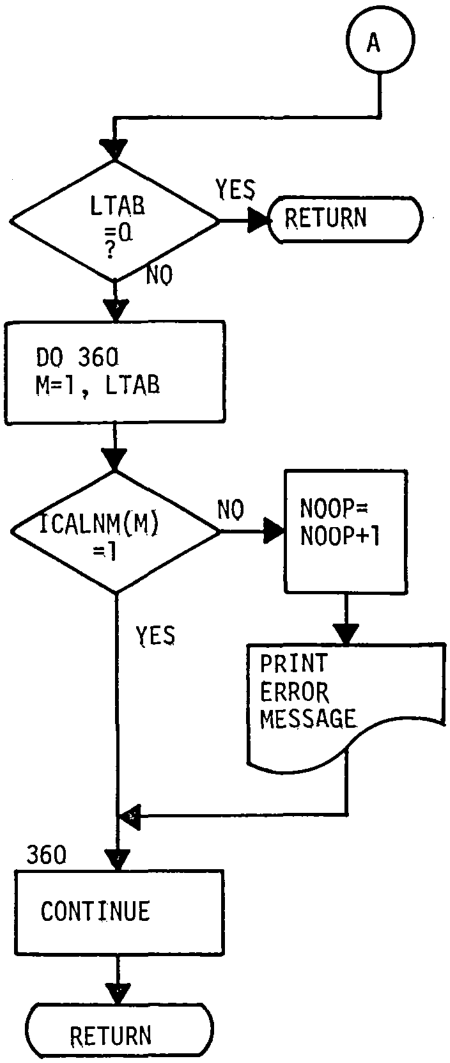


A
150
CONTINUE
200
READ CARD
EOF ?
NO
YES
800
PRINT MESSAGE "END OF FILE CALIN"
STOP
PRINT CARD
JCDCNT= JCDCNT +1
CCODE = "CS" ?
NO
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
210
DECODE CARD (NAMES (L), L=1,8
ICALP=1 ?
YES
NO
GO TO 250
DO 215 L=1,8
NAMES(L) = " " ?
YES
NO
GO TO 300
LTAB=LTAB+1. CALNAM(LTAB)= NAMES(L) ISNCAL (LTAB) =ICALP
215
CONTINUE
GO TO 300

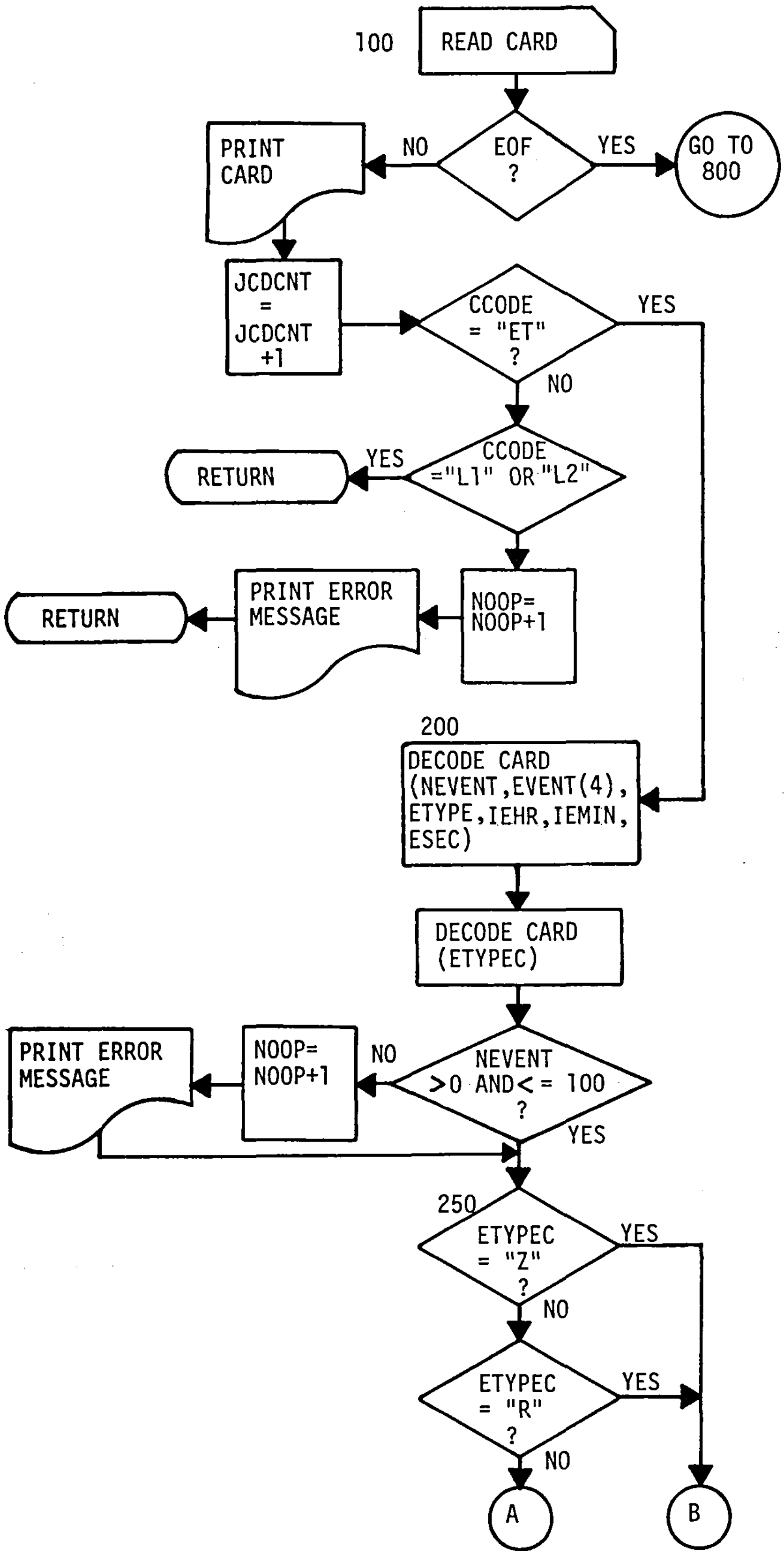


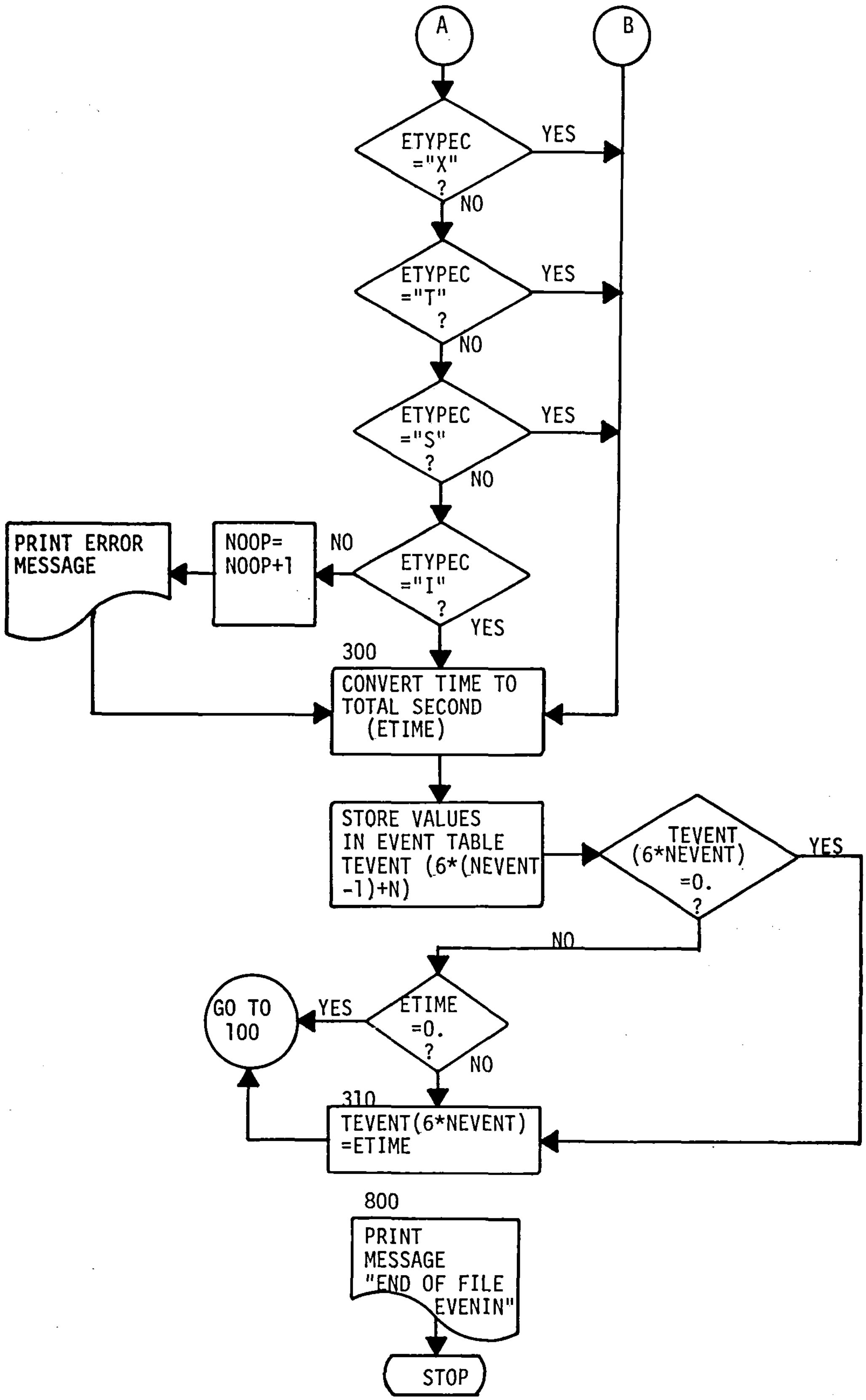
250
NAMES (L) = ("AC") ?
NO
YES
NOOP= NOOP+1
PRINT ERROR MESSAGE
300
READ CARD
EOF ?
YES
GO TO 800
NO
PRINT CARD
JCDCNT = JCDCNT +1
CCODE = "CT" ?
YES
GO TO 110
NO
CCODE = "CS" ?
YES
GO TO 210
NO
CCODE = "L1" ?
YES
NO
CCODE = "L2" ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
320
YES
DO 350 I=1, THEAD (1)
TPASS(I,4) =2. ?
YES
SET TPASS (I,4)=1.
NO
A

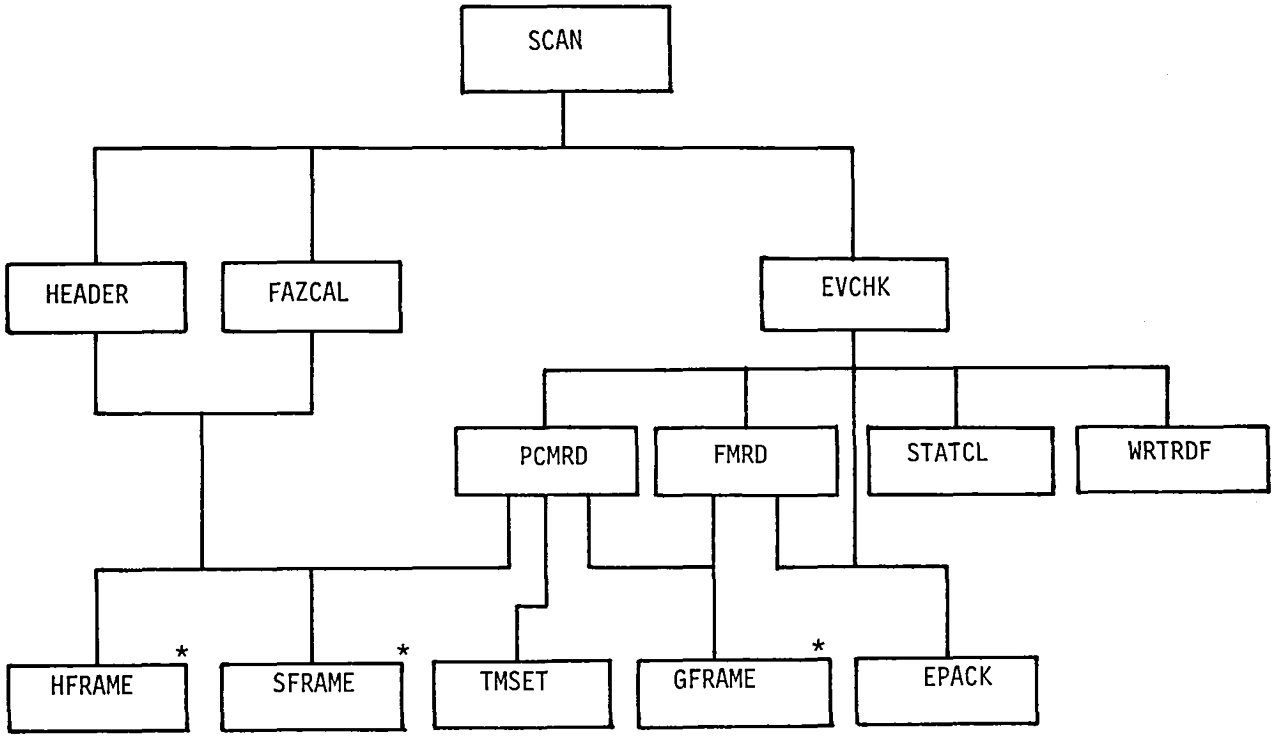

B
READ (01'I) SENTAB
DO 352 J=1, TPASS (I,1)
ENCODE (SNAM,F)SENTAB (J,1),SENTAB(J,2) (CALTC,F)SENTAB (J,6)
DO 340 M=1, LTAB
340
CONTINUE
NO
SENAM =CALNAM(M)
YES
344
PPC = 'P' ?
GO TO 348
NOOP= NOOP+1
PRINT ERROR MESSAGE
342
CALTC ="1" ?
YES
NO
CALTC ='2' ?
YES
NO
PRINT ERROR MESSAGE
NOOP= NOOP+1
346
CALTC= "1" ?
YES
NO
YES
PPC= "P" ?
NO
SENTAB(J12)= ISNCAL(M) ICALNM(M)=1
PPC ="P" ?
NO
CALTC ="2" ?
YES
YES
NO
WRITE (01'I) SENTAB(II, JJ)JJ=1,13 II=1,40
352
CONTINUE
350
CONTINUE
347
NOOP= NOOP+1
SENTAB(J,12)= ICAL1
348
SENTAB(J,12)= ICALO
A
PRINT ERROR MESSAGE
GO TO 352


A
LTAB =0 ?
YES
RETURN
NO
DO 360 M=1, LTAB
ICALNM(M) =1
NO
NOOP= NOOP+1
YES
PRINT ERROR MESSAGE
360
CONTINUE
RETURN

```
********************SUBROUTINE EVENIN************************************

PROGRAM IDENTIFICATION

    PROGRAM NAME ----       SUBROUTINE EVENIN
    PROGRAM NUMBER ------- 112320
    AUTHOR -------    TERRY D. SOMMERS

    COMPUTER ------         HW-625/635
    MEMORY ------
    PERIPHERALS ------      CARD READER,PRINTER
    LANGUAGE ------   HW 6000 FORTRAN/FORTY

PURPOSE

    TO BUILD OR UPDATE EVENT TABLES

METHOD

    READ EVENT CARDS, DECODE AND EDIT PARAMETERS FOR ERRORS.
    STORE VALUES IN EVENT TABLE TO BE WRITTEN IN SETUP ROUTINE.

INPUT/OUTPUT

    CCODE - INPUT AND OUTPUT ARGUMENT CONTAINING CARD CODE
    CARD - INPUT AND OUTPUT ARGUMENT CONTAINING CHARACTER INFORMATION
    NOOP - INPUT AND OUTPUT ARGUMENT CONTAINING NUMBER OF
           ERRORS COUNTED FROM BAD INPUTS.
    JCDCNT - INPUT AND OUTPUT ARGUMENT CONTAINING TOTAL NUMBER OF
             CARDS INPUT
    ET CARD
       NEVENT - EVENT NUMBER
       EVENT1 + EVENT2 + EVENT3 + EVENT4 - EVENT NAME
       ETYPE - EVENT TYPE (Z,R,X,T,S,I)
       ETIME - TIME OF EVENT
    TEVENT - EVENT TABLE ARRAY

COMMON AREAS

    DATIN - CARD,OPT,TNAME,CCODE,NOOP,JCDCNT
    TABL4 - TEVENT

       SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS
     AND TABLE FILE FORMATS OR FOR ANY OTHER SPECIFIC INFORMATION.
```


100
READ CARD
EOF ?
YES
GO TO 800
NO
PRINT CARD
JCDCNT = JCDCNT +1
CCODE = "ET" ?
YES
NO
CCODE ="L1" OR "L2"
YES
RETURN
NOOP= NOOP+1
PRINT ERROR MESSAGE
RETURN
200
DECODE CARD (NEVENT,EVENT(4), ETYPE,IEHR,IEMIN, ESEC)
DECODE CARD (ETYPEC)
NEVENT >0 AND< = 100 ?
NO
NOOP= NOOP+1
PRINT ERROR MESSAGE
YES
250
ETYPEC = "Z" ?
YES
NO
ETYPEC = "R" ?
YES
NO
A
B


A
B
ETYPEC ="X" ?
YES
NO
ETYPEC ="T" ?
YES
NO
ETYPEC ="S" ?
YES
NO
PRINT ERROR MESSAGE
NOOP= NOOP+1
NO
ETYPEC ="I" ?
YES
300
CONVERT TIME TO TOTAL SECOND (ETIME)
STORE VALUES IN EVENT TABLE TEVENT (6*(NEVENT -1)+N)
TEVENT (6*NEVENT) =0. ?
YES
NO
GO TO 100
YES
ETIME =0. ?
NO
310
TEVENT(6*NEVENT) =ETIME
800
PRINT MESSAGE "END OF FILE EVENIN"
STOP



*Program Modules SFRAME, HFRAME, GFRAME documentation in NASA, WFC Program Library

HIERARCHY CHART for SCAN PROGRAM

********SCAN - PROGRAM NUMBER 1.1.2326

*******NASA WALLOPS VERSION OF 08/01/77

*******LANGUAGE - FORTRAN IV

*******MACHINE - HW 625/635

*******PROGRAM AUTHOR - JEANETTE WESSELLS

*******PURPOSE -

THIS IS THE MAIN PROGRAM WHICH READS PCM AND FM DIGITIZED INPUT DATA IN TELEVENT FORMAT, COMPUTES BASIC STATISTICS, AND STORES DATA AND STATISTICS IN RANDOM FILES ACCEPTABLE TO OTHER EASE PROGRAMS.

*******METHOD -

THIS PROGRAM ACCEPTS EITHER PCM OR FM TAPES CONTAINING DATA RECORDED BY RSRA AIRCRAFT FLIGHT.

THE DIGITIZATION OF PCM DATA TAPE IS BY STANDARD PROCEEDURES. FM DATA IS DIGITIZED BY USE OF A PEAK - STRESS CONVERTER WHICH TAKES A CHANNEL OF DATA AND, DEPENDING ON THE SPECIFIED SAMPLING RATE, OUTPUTS A CHANNEL OF PEAK VALUES AND A CHANNEL OF VALLEY VALUES WITHIN THAT SAMPLING RATE. THESE TWO CHANNELS ARE COMBINED TO GIVE VIBRATORY AND STEADY COMPONENTS OF THE ORIGINAL DATA. THE BASIC STATISTICS ARE COMPUTED FOR THESE COMPONENTS.

PCM DATA IS GROUPED ACCORDING TO PASSES. A PASS IS A SET OF DATA WITH NO MORE THAN 40 CHANNELS, EACH CHANNEL REPRESENTING A SPECIFIC PARAMETER, COVERING THE ENTIRE FLIGHT FOP THESE CHANNELS AND SAMPLED AT THE SAME RATE.

A PASS IS FURTHER DIVIDED INTO BURSTS REPRESENTING DATA FOR A PARTICULAR FLIGHT MANEUVER. DATA BURSTS CAN BE DISTINGUISHED BY EITHER RUN TONE SIGNAL OR A GIVEN TIME INTERVAL. NORMAL MODE IS BY RUN TONE. CARD INPUT IS REGUIRED WITH BURST NUMBER AND START AND STOP TIMES FOR TIME INTERVAL. STATISTICS ARE COMPUTED AND STORED FOR EACH CHANNEL OF DATA BY THESE DATA BURSTS.

EACH PASS AND FLIGHT MANEUVER IS DEFINED IN THE SETUP ROUTINE. GIVEN THE PASS NUMBER, FILE NUMBER, AND FILE CODE DESIGNATOR, SETUP WILL SUPPLY NECCESSARY PARAMETER CONFIGURATION OF THE INPUT DATA.

THE ENTIRE PASS OF DATA IS THEN PROCESSED THROUGH THIS ROUTINE.

*******SYSTEMS INPUT FILES -

FILE 03 = INPUT TAPE

FILE 04 = INPUT TAPE, IF MULTI-REEL IS NECCESSARY

FILE 05 = CARD READER

FILE 07 = TABLE FILE

*******SYSTEMS OUTPUT FILES -

FILE 06 = PRINTER

FILE 07 = TABLE FILE

FILE 09 = STATISTICS FILE

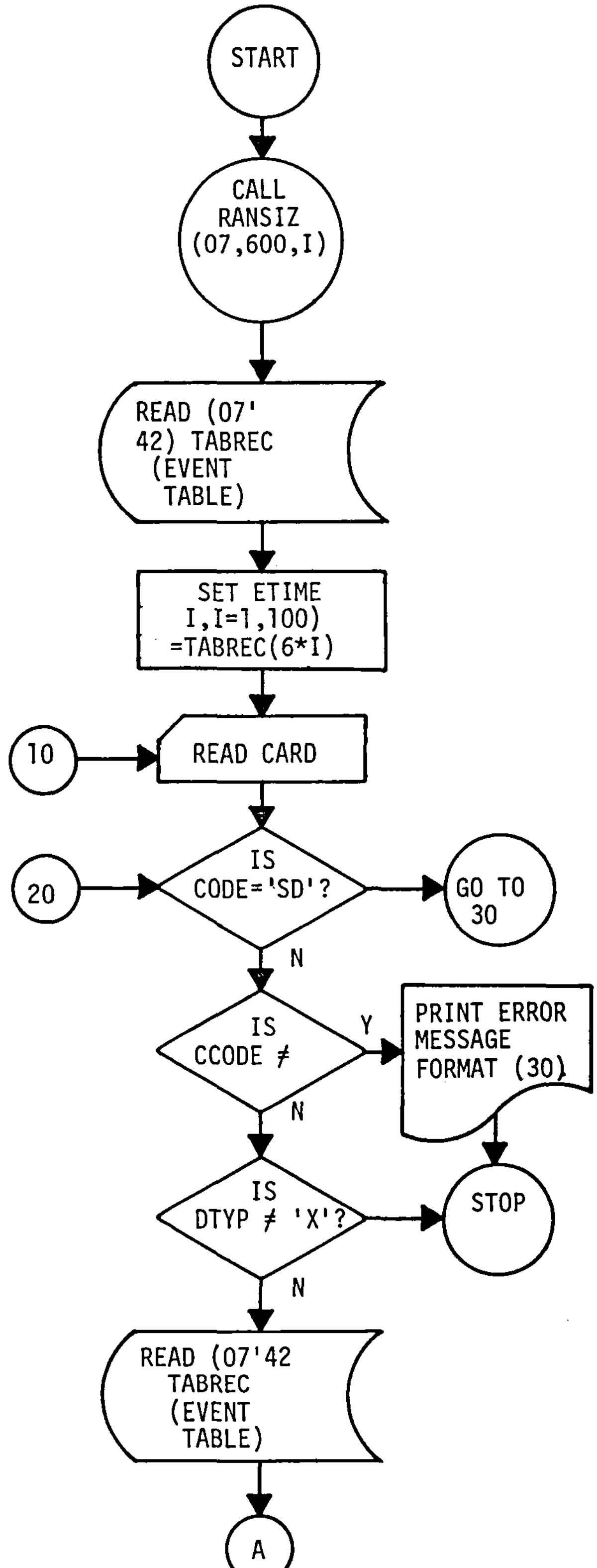
FILES 11,12,...31 = PASS DATA FILES

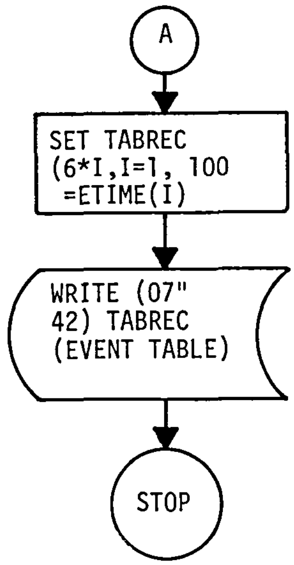
*******INPUT -

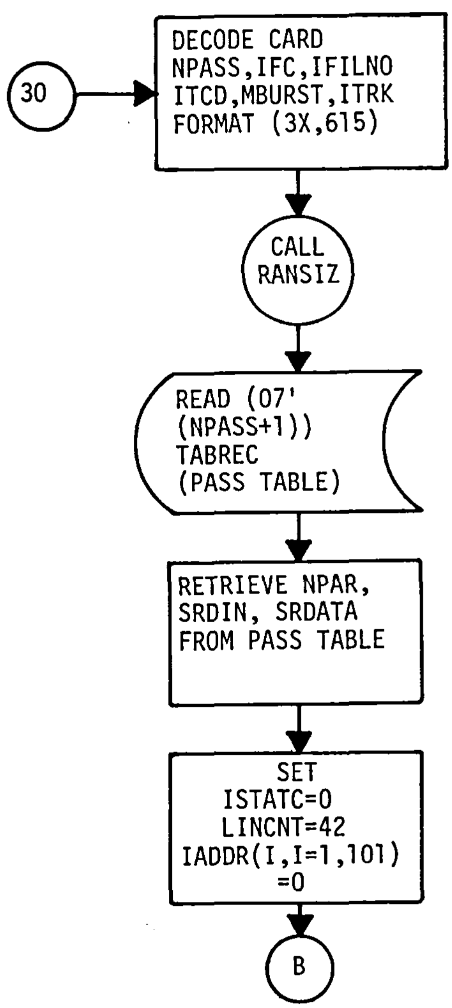
CARD	-78 CHARACTER CARD IMAGE
CCODE	-2 CHARACTER CARD CODE FOUND IN FIRST 2 COLUMNS OF -INPUT CARD -- IDENTIFIES INPUT DATA WITHIN "CARD"
ETIME	-BEGINNING TIME (SEC) OF MANEUVER UP TO 100 ENTRIES -ARRAY IS EXTRACTED FROM EVENT TABLE
NPASS	-PASS NUMBER BEING PROCESSED (CARD INPUT)
IFC	-FILE CODE OF INPUT TAPE (CARD INPUT)
ITCD	-CODE DESIGNATING WHICH TIME WORD TO USE (CARD INPUT) -=0,AIRBOURNE TIME -=1,GROUND STATION TIME
IFILNO	-LOGICAL FILE WHERE PASS = NPASS TO BE FOUND ON -INPUT TAPE (CARD INPUT)
MBURST	-MAXIMUM NUMBER OF DATA BURSTS FOUND ON INPUT TAPE -FOR THIS PASS (CARD INPUT)
JPARIN	-WORD NUMBER OF SENSOR DATA WITHIN THE DATA FRAME -MAXIMUM 40 WORDS (FROM SENSOR TABLE)
JPARDF	-WORD NUMBER OF SENSOR DATA WITHIN THE DATA FILE -FRAME (FROM SENSOR TABLE)
PPC	-PRE-PROCESSING CODE OF SENSOR (FROM SENSOR TABLE)
LASTFL	-COUNTER FOR CURRENT LOGICAL FILE BEING PROCESSED
TABREC	-600-WORD ARRAY OF TABLE DATA READ FROM FILE 07

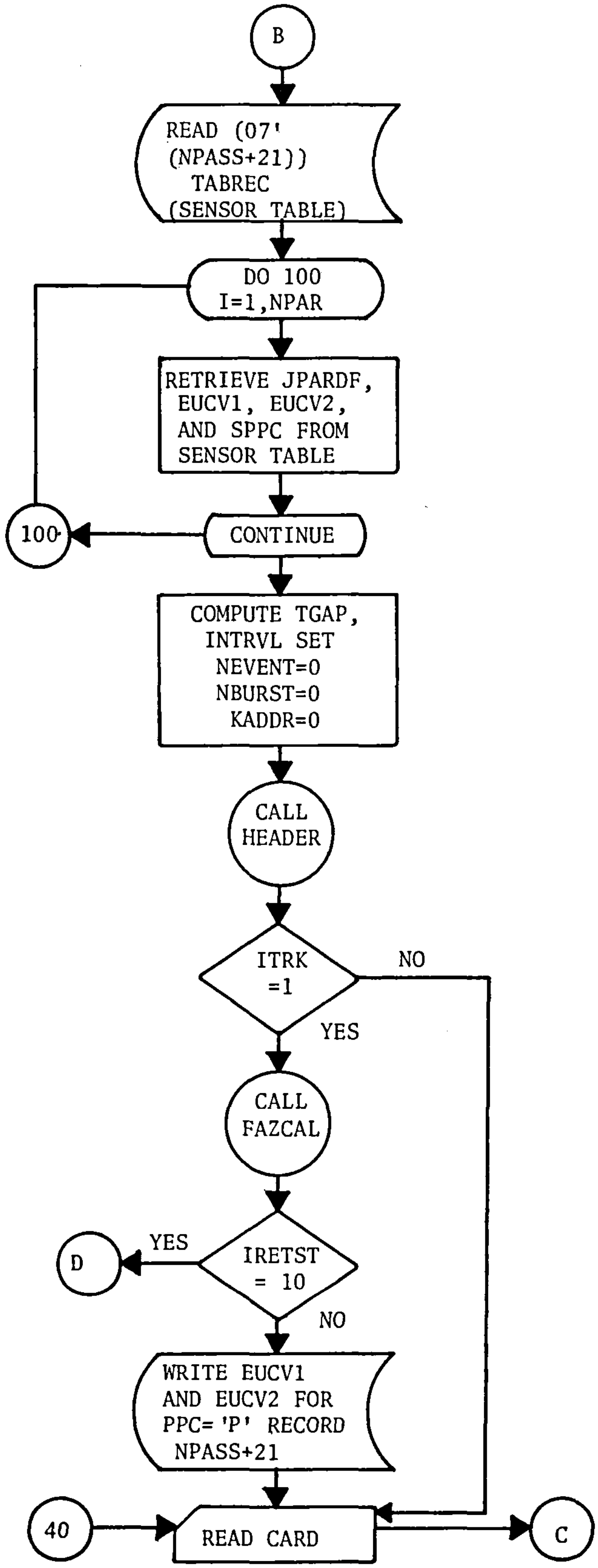
```
IDATA     -640-WORD RECORD ARRAY FOR DATA IMAGE OF PACKED
          -DATA ON DATA FILE

NEVENT    -MANEUVER NUMBER

IBURST    -BURST NUMBER (CARD INPUT)

ICOND     -CODE TO DESIGNATE HOW "IBURST" IS TO BE HANDLED
          -(CARD INPUT)
          -=0,NORMAL CONDITION - BURST INTERVAL DETERMINED
          -   BY RUN TONE
          -=1,BURST INTERVAL DETERMINED BY RUN TONE IS NOT
          -   TO BE PROCESSED
          -=2,BURST INTERVAL IS DETERMINED BY TIME
          -   NOTE< TIME VALUES TO AND TF ARE REQUIRED
          -=3,NO DATA BURST EXISTS ON INPUT TAPE FOR A VALID
          -   MANEUVER -- MUST WRITE A RECORD OF NULLS FOR
          -   STATISTICS AND INCREMENT NEVENT

TO        -BEGINNING TIME OF DATA BURST (SEC) (CARD INPUT)

TF        -ENDING TIME OF DATA BURST (SEC) (CARD INPUT)

KADDR     -CURRENT ADDRESS OF DATA FILE RECORD FOR NPASS

IRETST    -TAPE STATUS AS RETURNED FROM GFRAME
          -10,END OF FILE ON TAPE

NCOND     -VALUE FOR ICOND FOR CURRENT BURST = "NBURST"

INTRVL    -RATIO OF INPUT SAMPLE RATE VS. DATA FILE SAMPLE
          -RATE


*******OUTPUT -

NPASS     -PASS NUMBER BEING PROCESSED

SRDIN     -INPUT SAMPLE RATE

SRDATA    -DATA FILE SAMPLE RATE

NEVENT    -MANEUVER NUMBER

IADDR     -STORED ADDRESS ARRAY OF MANEUVER NUMBER

KADDR     -CURRENT ADDRESS OF DATA FILE RECORD FOR NPASS

ETO       -BURST START TIME

ETF       -BURST STOP TIME

NFRM      -NUMBER OF FRAMES OF DATA WITHIN BURST

NLOS      -NUMBER OF LOSS OF SYNS WITHIN BURST
```

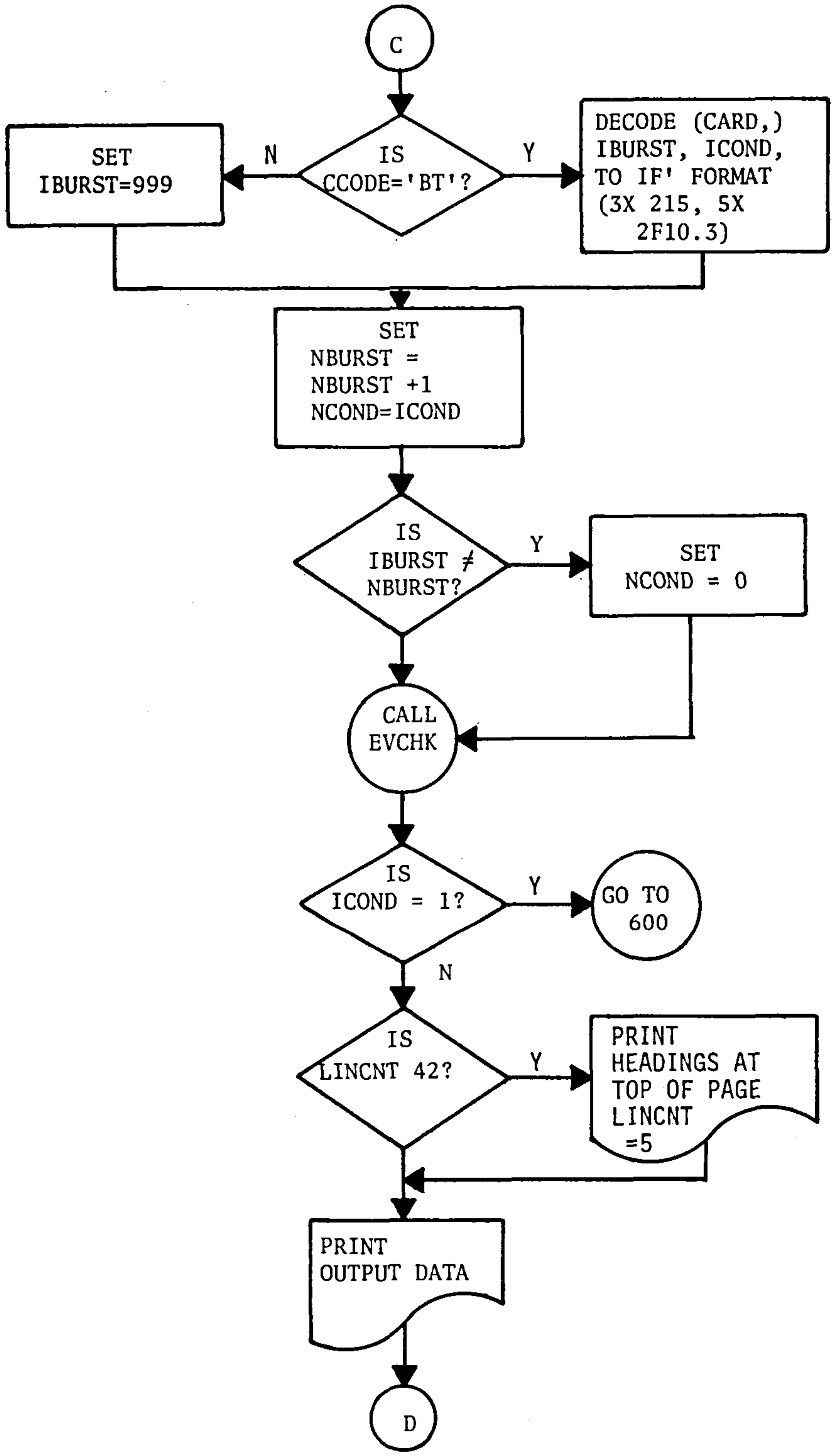
```
*******RESTRICTIONS -
       LIMIT OF 20 PASSES PER FLIGHT
       LIMIT OF 40 PARAMETERS PER PASS
       LIMIT OF 100 MANEUVERS
       INPUT TAPES ARE STANDARD TELEVENT FORMAT (RESTRICTED BY
       GFRAME MODULE)


*******SUBPROGRAMS REQUIRED -
       HEADER
             SFRAME
             HFRAME
       EVCHK
             PCMRD
                  GFRAME
                  SFRAME
                  TMSET
             FMRD
                  GFRAME
                  EPACK
             STATCL
             WRTRDF
             EPACK
```

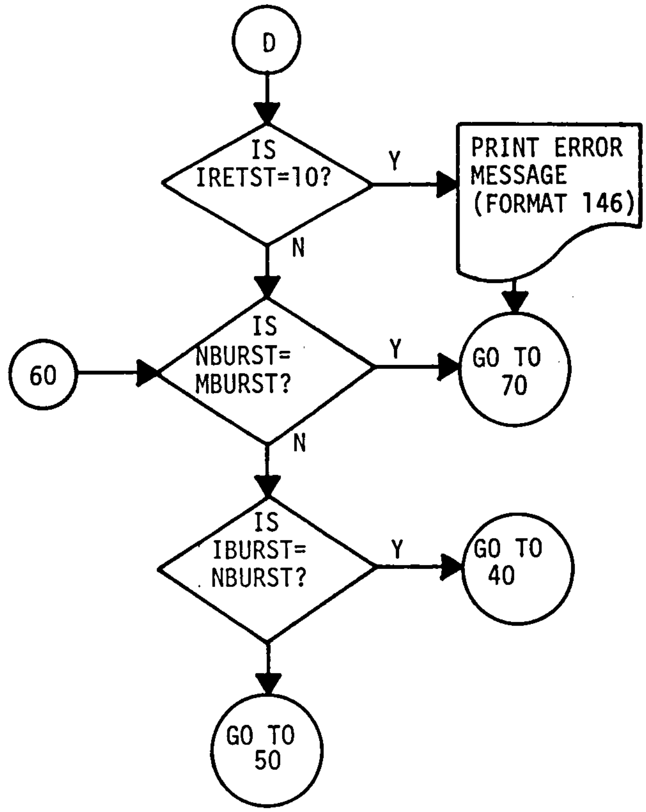

START
CALL RANSIZ (07,600,I)
READ (07'42) TABREC (EVENT TABLE)
SET ETIME I,I=1,100) =TABREC(6*I)
10
READ CARD
20
IS CODE='SD'?
GO TO 30
N
IS CCODE ≠
Y
PRINT ERROR MESSAGE FORMAT (30)
N
IS DTYP ≠ 'X'?
STOP
N
READ (07'42 TABREC (EVENT TABLE)
A

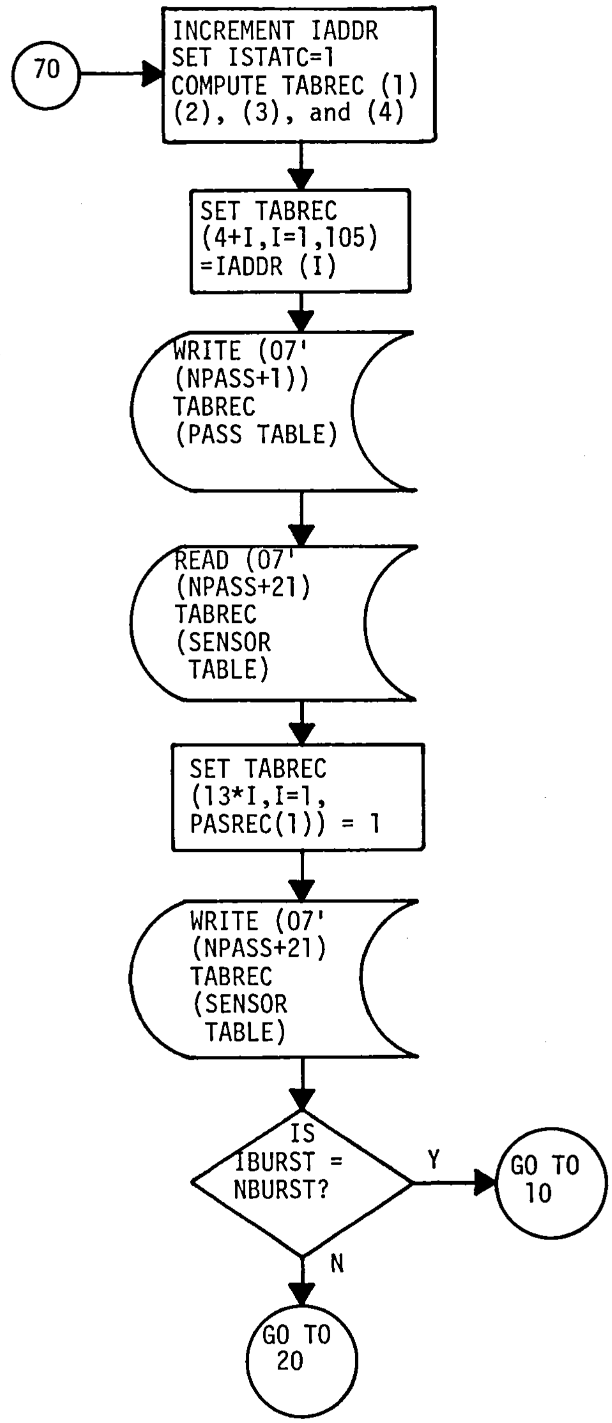

A
SET TABREC
(6*I,I=1, 100
=ETIME(I)
WRITE (07"
42) TABREC
(EVENT TABLE)
STOP


30
DECODE CARD
NPASS,IFC,IFILNO
ITCD,MBURST,ITRK
FORMAT (3X,615)
CALL
RANSIZ
READ (07'
(NPASS+1))
TABREC
(PASS TABLE)
RETRIEVE NPAR,
SRDIN, SRDATA
FROM PASS TABLE
SET
ISTATC=0
LINCNT=42
IADDR(I,I=1,101)
=0
B

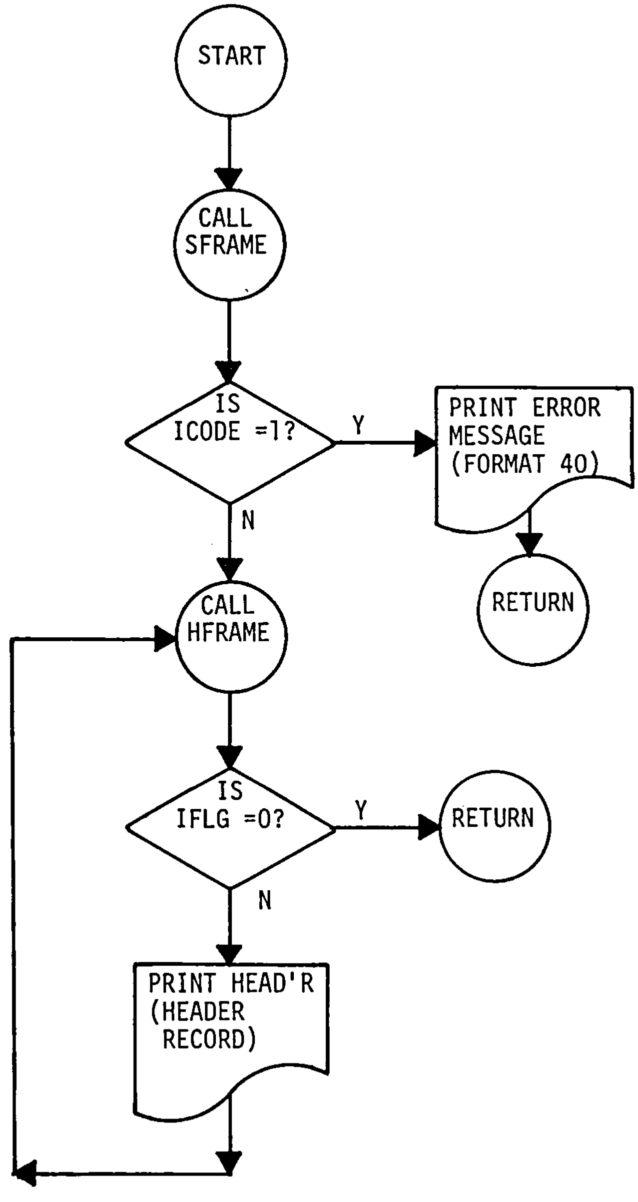

B
READ (07'
(NPASS+21))
TABREC
(SENSOR TABLE)
DO 100
I=1,NPAR
RETRIEVE JPARDF,
EUCV1, EUCV2,
AND SPPC FROM
SENSOR TABLE
100
CONTINUE
COMPUTE TGAP,
INTRVL SET
NEVENT=0
NBURST=0
KADDR=0
CALL
HEADER
ITRK
=1
NO
YES
CALL
FAZCAL
D
YES
IRETST
= 10
NO
WRITE EUCV1
AND EUCV2 FOR
PPC= 'P' RECORD
NPASS+21
40
READ CARD
C


C
SET IBURST=999
N
IS CCODE='BT'?
Y
DECODE (CARD,) IBURST, ICOND, TO IF' FORMAT (3X 215, 5X 2F10.3)
SET NBURST = NBURST +1 NCOND=ICOND
IS IBURST ≠ NBURST?
Y
SET NCOND = 0
CALL EVCHK
IS ICOND = 1?
Y
GO TO 600
N
IS LINCNT 42?
Y
PRINT HEADINGS AT TOP OF PAGE LINCNT =5
PRINT OUTPUT DATA
D

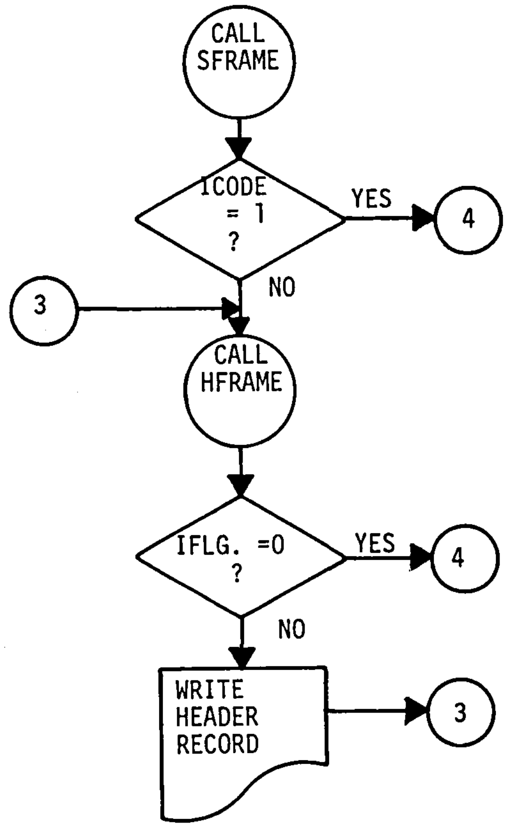

D
IS IRETST=10?
Y
N
PRINT ERROR MESSAGE (FORMAT 146)
60
IS NBURST= MBURST?
Y
N
GO TO 70
IS IBURST= NBURST?
Y
GO TO 40
GO TO 50

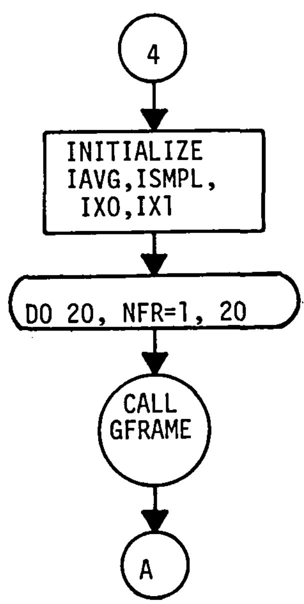

70
INCREMENT IADDR
SET ISTATC=1
COMPUTE TABREC (1)
(2), (3), and (4)
SET TABREC
(4+I,I=1,105)
=IADDR (I)
WRITE (07'
(NPASS+1))
TABREC
(PASS TABLE)
READ (07'
(NPASS+21)
TABREC
(SENSOR
TABLE)
SET TABREC
(13*I,I=1,
PASREC(1)) = 1
WRITE (07'
(NPASS+21)
TABREC
(SENSOR
TABLE)
IS
IBURST =
NBURST?
Y
GO TO
10
N
GO TO
20

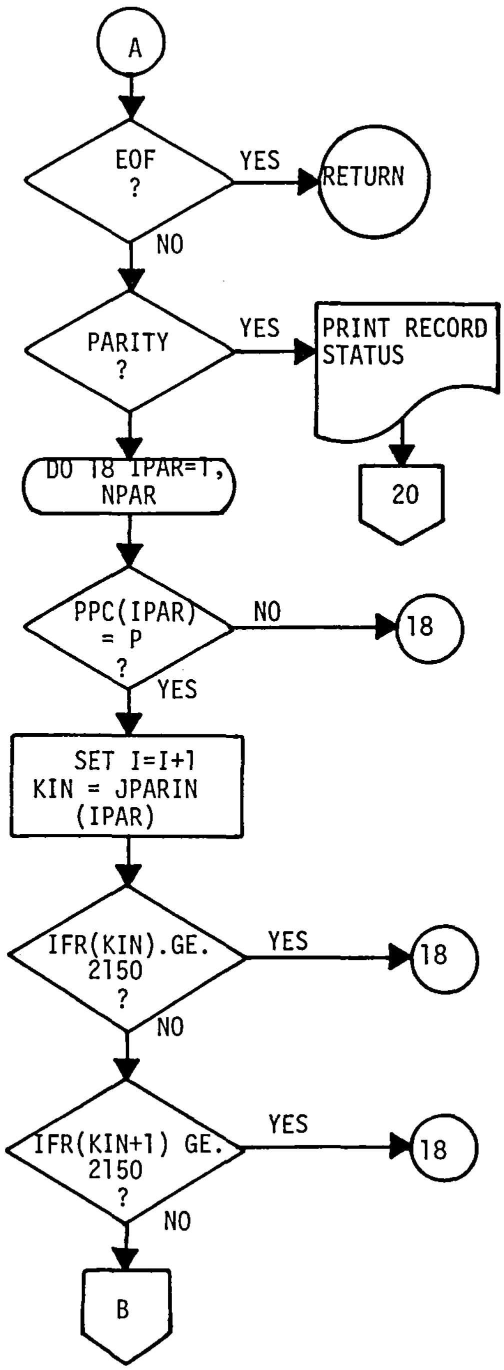
```
HEAD    HEADER-INITIALIZE INPUT 6130-TAPE
**********************SUBROUTINE HEADER********************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       INITIALIZE INPUT TAPE AND READ HEADER RECORD


*******METHOD -
       INITIALIZE TAPE BY CALLING SFRAME AND READ THE HEADER RECORD
       BY CALLING HFRAME.


*******INPUT -

       IFCD       -FILE CODE OF INPUT TAPE

       IFL        -LOGICAL FILE WHERE PASS = NPASS TO BE FOUND ON
                  -INPUT TAPE


*******OUTPUT -

       HEADR      -HEADER RECORD TO BE PRINTED


*******RESTRICTIONS -
       NONE


*******SUBPROGRAMS REQUIRED -
       SFRAME
       HFRAME
```


START
CALL SFRAME
IS ICODE =1?
Y
PRINT ERROR MESSAGE (FORMAT 40)
RETURN
N
CALL HFRAME
IS IFLG =0?
Y
RETURN
N
PRINT HEAD'R (HEADER RECORD)

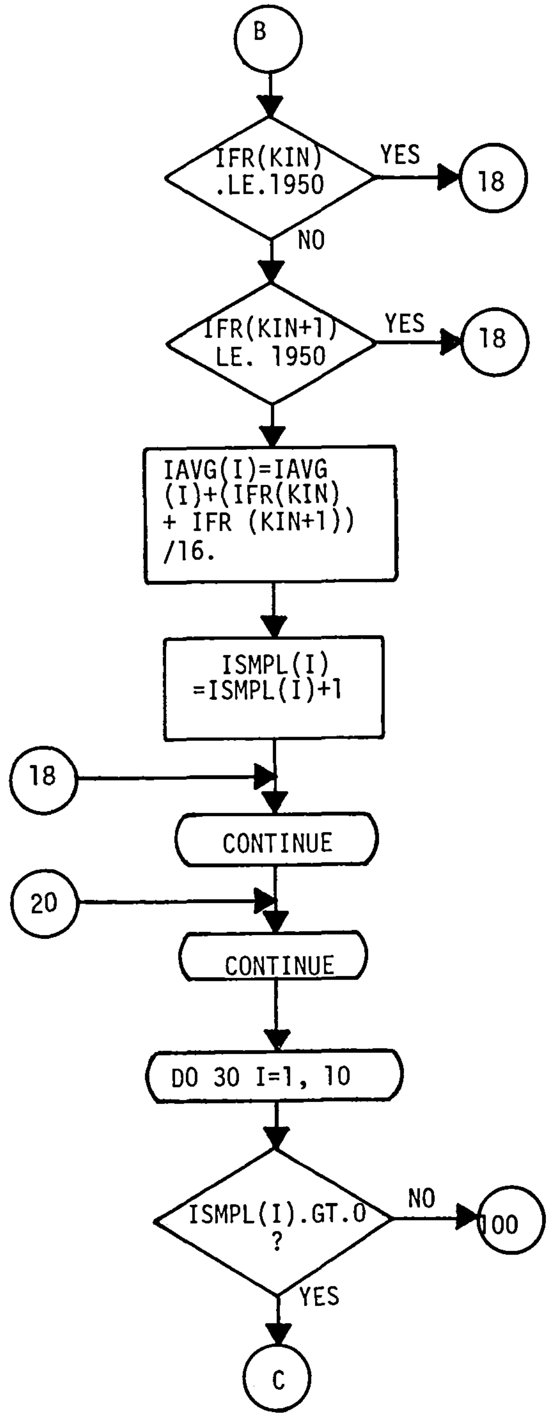
```
FAZCAL
******************************SUBROUTINE FAZCAL**********************


******************NASA WALLOPS VERSION OF MARCH 15,1978


******************LANGUAGE - FORTRAN IV


******************MACHINE - HONEYWELL 635


******************PURPOSE -
                  THIS ROUTINE PROCESSES PHASE CALIBRATION DATA.


******************METHOD -
                  THE FM TAPE MUST BE OF STANDARD TELEVENT FORMAT
                  BECAUSE DATA IS DECODED BY GFRAME MODULE.
                  THE STATUS OF EACH RECORD IS CHECKED FOR POSSIBLE
                  PARITY ON READ, AND END OF FILE DESIGNATOR.
                  ZERO DEGREE PHASE CALIBRATION IS COMPUTED AND
                  STORED IN ARRAY IAVG.
                  AVERAGE VALUE FOR ZERO DEGREES IS CALCULATED
                  AND STORED IN ARRAY IX0.
                  360 DEGREE PHASE CALIBRATION IS COMPUTED AND STORED
                  IN ARRAY IAVG.
                  AVERAGE VALUE FOR 360 DEGREES IS COMPUTED AND STORED
                  IN ARRAY IX1.
                  EUCV1 AND EUCV2 TERMS ARE COMPUTED AND STORED IN
                  ARRAYS EUCV1 AND EUCV2.


******************OUTPUT -
                  FILE NUMBER 06
                  CONSISTS OF -
                                - SENSOR NUMBER
                                - ZERO DEGREE AVERAGE
                                - 360 DEGREE AVERAGE
                                - EUCV1 (DEFLECTION IN ENG. UNITS)
                                - EUCV2 (OFFSET IN ENG. UNITS)
                                -IRETST - READ STATUS OF FRAME
                                        - 1,2,OR 3 GOOD READ
                                        - 10= END OF FILE


******************RESTRICTIONS - FM TAPE MUST BE STANDARD TELEVENT
                                FORMAT.


******************SUBPROGRAMS REQUIRED - GFRAME,HFRAME,SFRAME
```

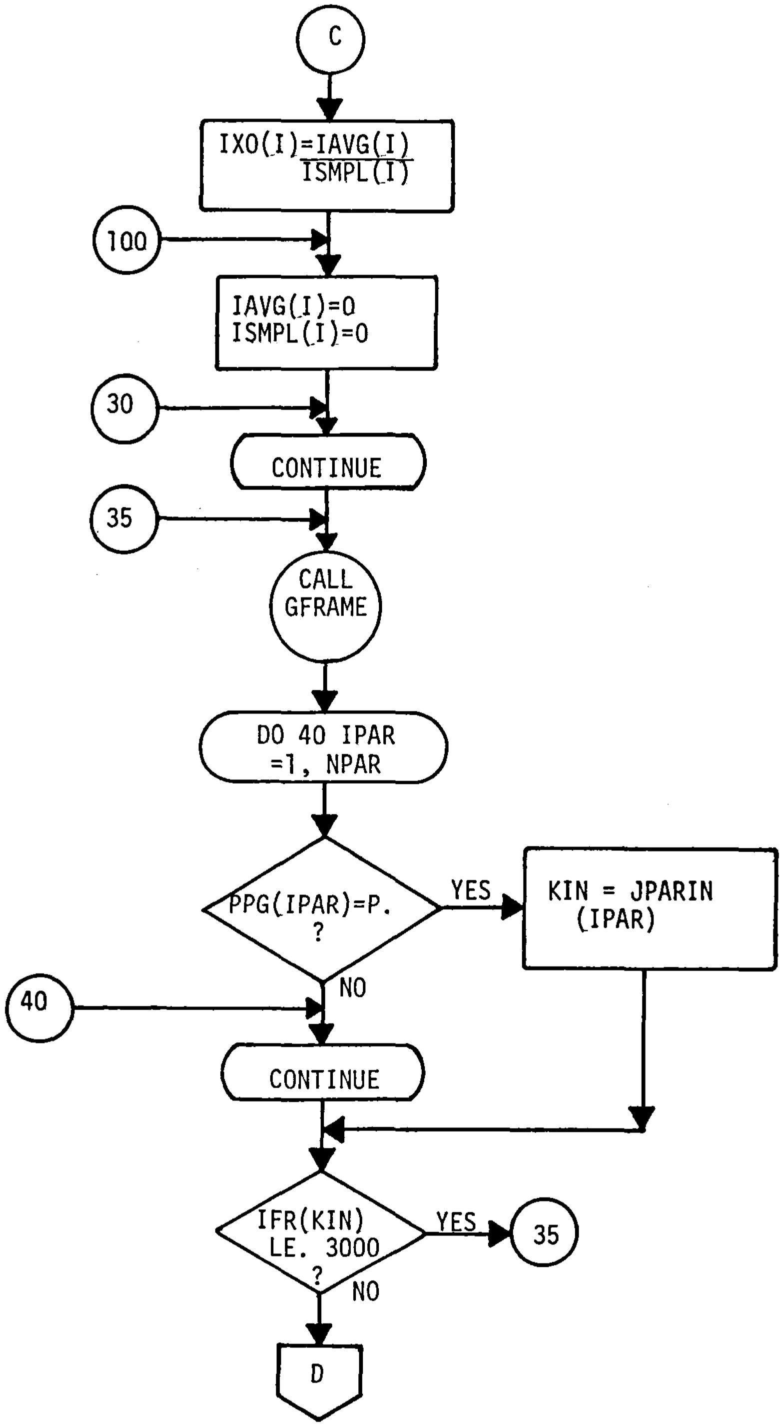

CALL SFRAME
ICODE = 1 ?
YES
4
NO
3
CALL HFRAME
IFLG. =0 ?
YES
4
NO
WRITE HEADER RECORD
3

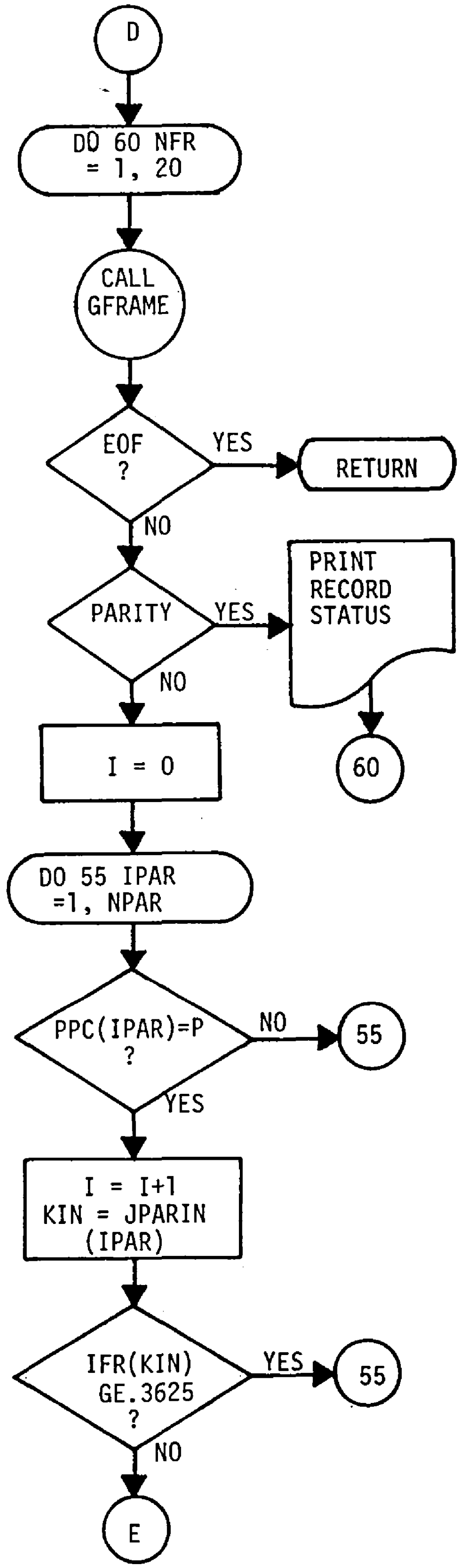

4
INITIALIZE IAVG,ISMPL, IX0,IX1
DO 20, NFR=1, 20
CALL GFRAME
A

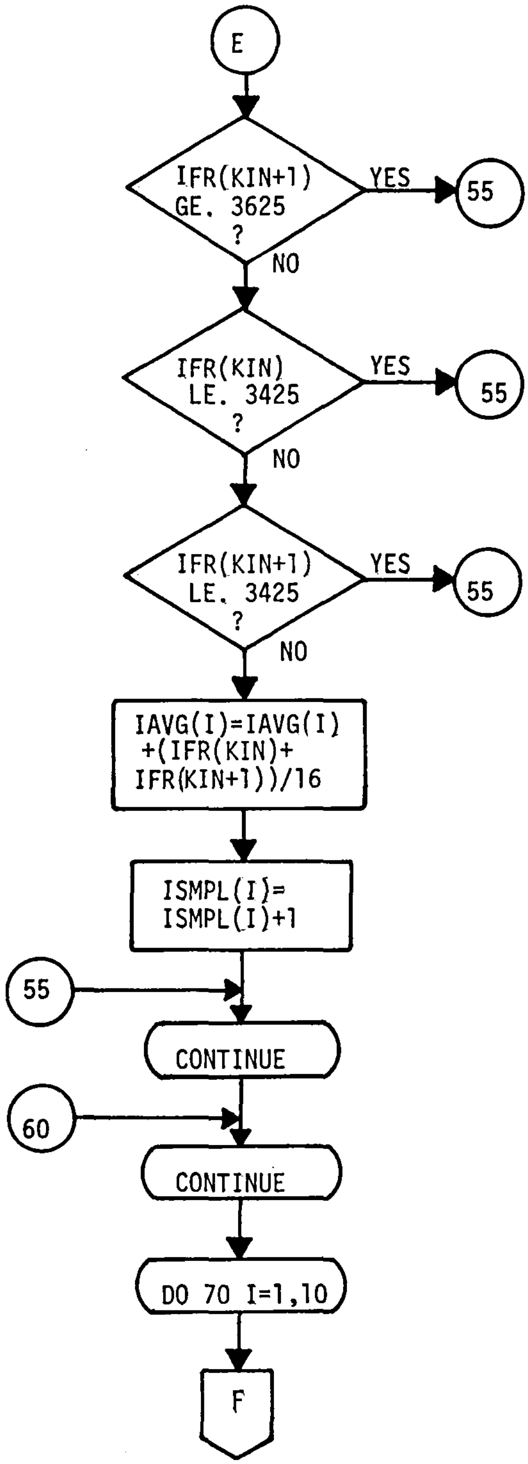


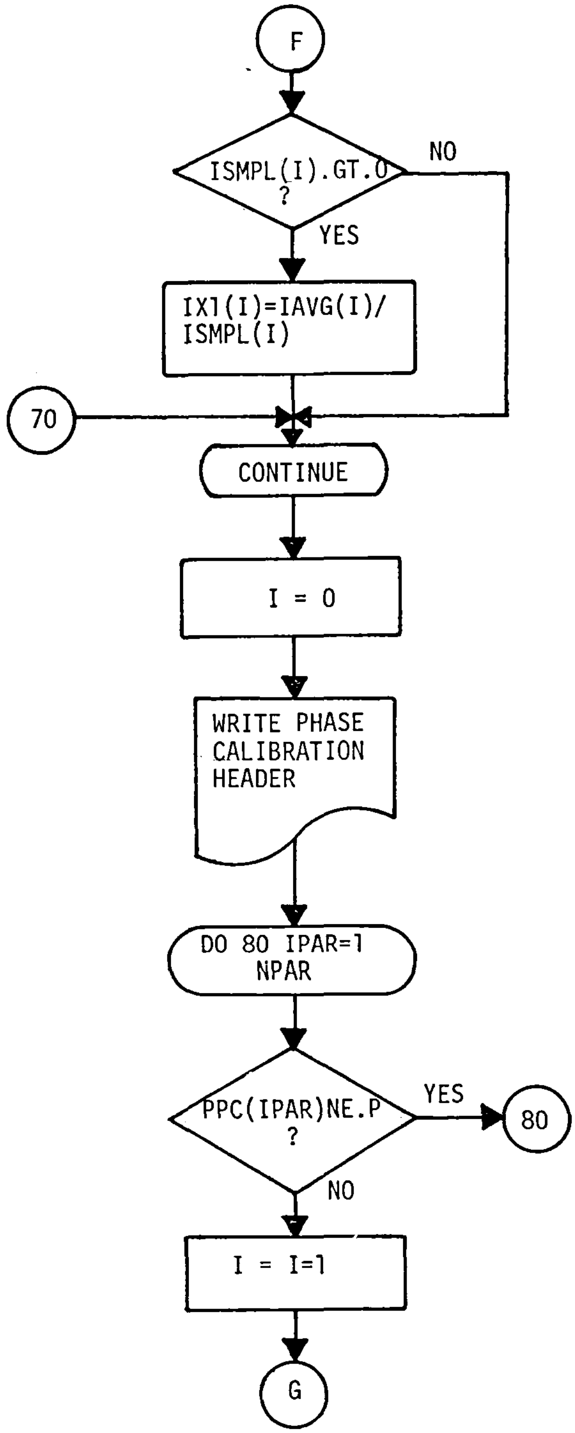
A
EOF ?
YES
RETURN
NO
PARITY ?
YES
PRINT RECORD STATUS
20
DO 18 IPAR=1, NPAR
PPC(IPAR) = P ?
NO
18
YES
SET I=I+1
KIN = JPARIN (IPAR)
IFR(KIN).GE. 2150 ?
YES
18
NO
IFR(KIN+1) GE. 2150 ?
YES
18
NO
B

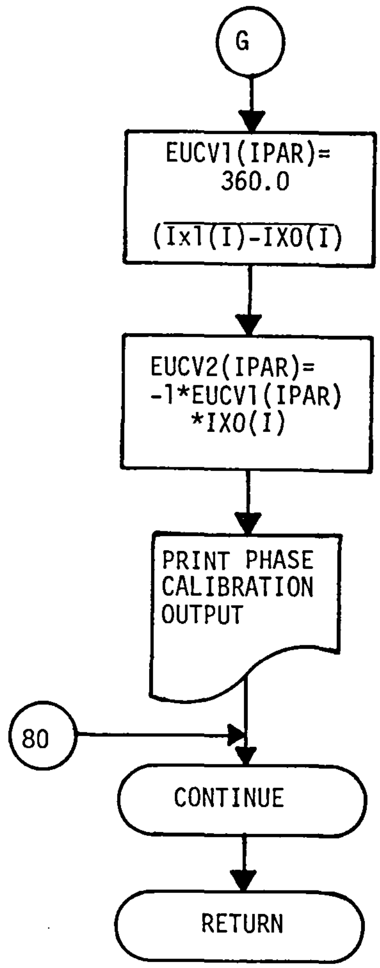

B
IFR(KIN)
.LE.1950
YES
18
NO
IFR(KIN+1)
LE. 1950
YES
18
IAVG(I)=IAVG
(I)+(IFR(KIN)
+ IFR (KIN+1))
/16.
ISMPL(I)
=ISMPL(I)+1
18
CONTINUE
20
CONTINUE
DO 30 I=1, 10
ISMPL(I).GT.0
?
NO
100
YES
C


C
IXO(I)=IAVG(I)/ISMPL(I)
100
IAVG(I)=0
ISMPL(I)=0
30
CONTINUE
35
CALL GFRAME
DO 40 IPAR =1, NPAR
PPG(IPAR)=P. ?
YES
KIN = JPARIN (IPAR)
NO
40
CONTINUE
IFR(KIN) LE. 3000 ?
YES
35
NO
D


D
DO 60 NFR = 1, 20
CALL GFRAME
EOF ?
YES
RETURN
NO
PARITY
YES
PRINT RECORD STATUS
60
NO
I = 0
DO 55 IPAR =1, NPAR
PPC(IPAR)=P ?
NO
55
YES
I = I+1 KIN = JPARIN (IPAR)
IFR(KIN) GE.3625 ?
YES
55
NO
E


E
IFR(KIN+1)
GE. 3625
?
YES
55
NO
IFR(KIN)
LE. 3425
?
YES
55
NO
IFR(KIN+1)
LE. 3425
?
YES
55
NO
IAVG(I)=IAVG(I)
+(IFR(KIN)+
IFR(KIN+1))/16
ISMPL(I)=
ISMPL(I)+1
55
CONTINUE
60
CONTINUE
DO 70 I=1,10
F

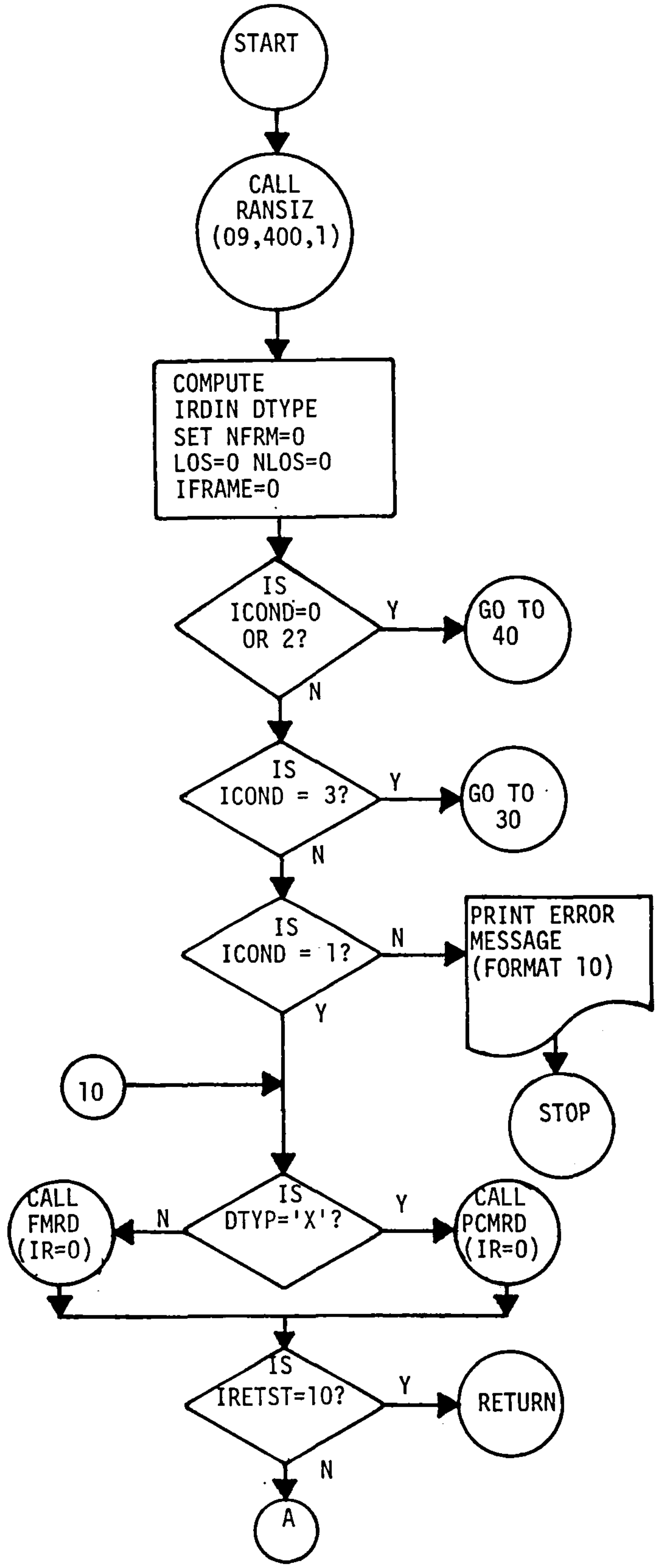

F
ISMPL(I).GT.0
?
NO
YES
IX1(I)=IAVG(I)/
ISMPL(I)
70
CONTINUE
I = 0
WRITE PHASE
CALIBRATION
HEADER
DO 80 IPAR=1
NPAR
PPC(IPAR)NE.P
?
YES
80
NO
I = I=1
G

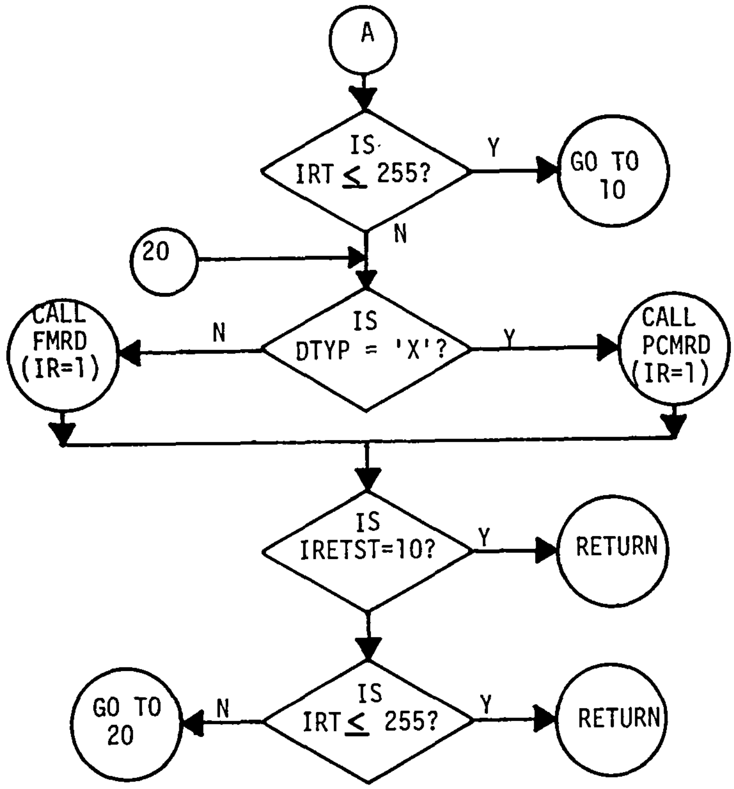

G
EUCV1(IPAR)= 360.0 / (Ix1(I)-IX0(I)
EUCV2(IPAR)= -1*EUCV1(IPAR) *IX0(I)
PRINT PHASE CALIBRATION OUTPUT
80
CONTINUE
RETURN

EVCH   EVCHK-CORDINATE EVENTS

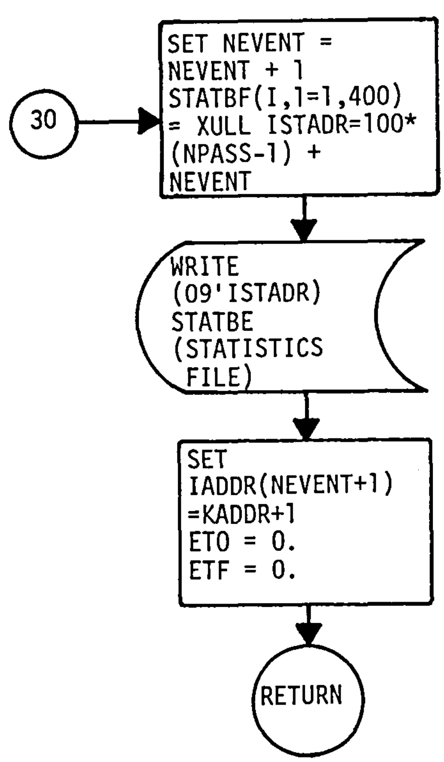
```
***********************SUBROUTINE EVCHK**********************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       PROCESS A BURST OF DATA


*******METHOD -
       BURST IS HANDLED ACCORDING TO ICOND FROM SCAN ROUTINE.  UNDER
       ICOND = 1 BURST IS READ UNTIL RUN TONE GOES OFF AND DATA IS
       IGNORED.  IF ICOND = 3 DATA RECORD AND STATISTICS ARE
       REPRESENTED BY NULL VALUES.  IF ICOND = 0 OR 2 DATA IS STORED
       AND THEIR STATISTICS ARE TO BE COMPUTED.

       METHOD OF PROCESSING DATA DEPENDS ON TYPE OF DATA BEING
       PROCESSED (PCM OR FM).

       PCM DATA MUST BE CHECKED FOR LOSS OF FRAME SYNC AND IT CAN BE
       PROCESSED AT A SAMPLE RATE LOWER THAN THE ORIGINAL.

       FM DATA HAS NO LOSS OF SYNC AND DATA IS PROCESSED AT THE
       SAMPLE RATE AS RECEIVED.  VIBRATORY AND STEADY COMPONENTS
       MUST BE COMPUTED FROM INPUT PEAK AND VALLEY DATA IN FMRD.
       PROCESSING OF DATA INCLUDES STORING DATA IN PACKED FORMAT
       THROUGH WRTRDF ROUTINE.  DATA IS PASSED TO STATCL ROUTINE FOR
       COMPILE AND COMPUTATION OF STATISTICS.  AT THE END OF EACH
       DATA BURST FINAL STATISTICS ARE COMPUTED AND WRITTEN ONTO THE
       STATISTICS FILE.


*******INPUT -

       NPASS       -PASS NUMBER BEING PROCESSED

       ICOND       -(SEE SCAN)

       IBURST      -CARD INPUT BURST NUMBER

       NEVENT      -MANEUVER NUMBER

       TO          -BEGINNING TIME OF DATA BURST (SEC)

       TF          -ENDING TIME OF DATA BURST (SEC)

       ITCD        -(SEE SCAN)
```

```
TGAP          -RATIO OF 1. VS. INPUT SAMPLE RATE

INTRVL        -RATIO OF INPUT SAMPLE RATE VS. DATA FILE SAMPLE
              -RATE

IFCD          -FILE CODE OF INPUT TAPE

DTYP          -CODE TO DETERMINE IF CURRENT PASS OF DATA IS
              -PCM (DTYP = "X") OR FM (OTHER) (CHARACTER PORTION
              -OF INPUT SAMPLE RATE)


*******OUTPUT -

KADDR         -CURRENT ADDRESS OF DATA FILE RECORD FOR NPASS
              -(OUTPUT FROM WRTRDF)

IRETST        -TAPE STATUS AS RETURNED FROM GFRAME (SEE PCMRD
              -AND FMRD)

ETO           -BURST START TIME (OUTPUT FROM WRTRDF)

ETF           -BURST STOP TIME (OUTPUT FROM WRTRDF)

NFRM          -NUMBER OF FRAMES OF DATA WITHIN BURST (OUTPUT
              -FROM WRTRDF)

NLOS          -NUMBER OF LOSS OF SYNCS WITHIN BURST

MODE          -(SEE STATCL AND WRTRDF)

LOS           -LOSS OF SYNC DETECTOR CODE
              -=0,NO LOSS OF SYNC IN BURST
              -=1,LOSS OF SYNC OCCURRED IN BURST


*******RESTRICTIONS -
      SAME AS IN SCAN


*******SUBPROGRAMS REQUIRED -
      PCMRD
            GFRAME
            SFRAME
            TMSET
      FMRD
            GFRAME
            EPACK
      STATCL
      WRTRDF
```

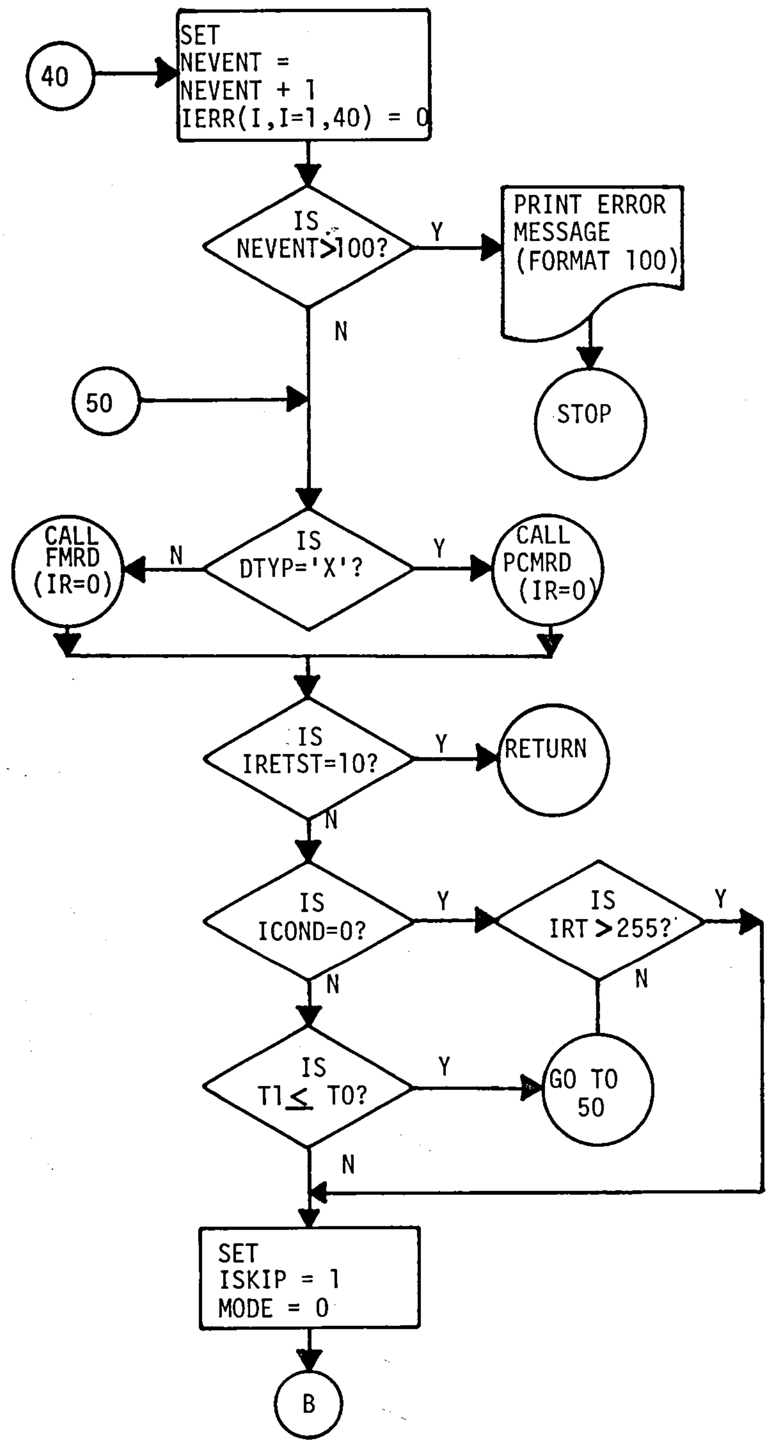

START
CALL RANSIZ (09,400,1)
COMPUTE
IRDIN DTYPE
SET NFRM=0
LOS=0 NLOS=0
IFRAME=0
IS ICOND=0 OR 2?
Y
GO TO 40
N
IS ICOND = 3?
Y
GO TO 30
N
IS ICOND = 1?
N
PRINT ERROR MESSAGE (FORMAT 10)
STOP
Y
10
IS DTYP='X'?
N
CALL FMRD (IR=0)
Y
CALL PCMRD (IR=0)
IS IRETST=10?
Y
RETURN
N
A

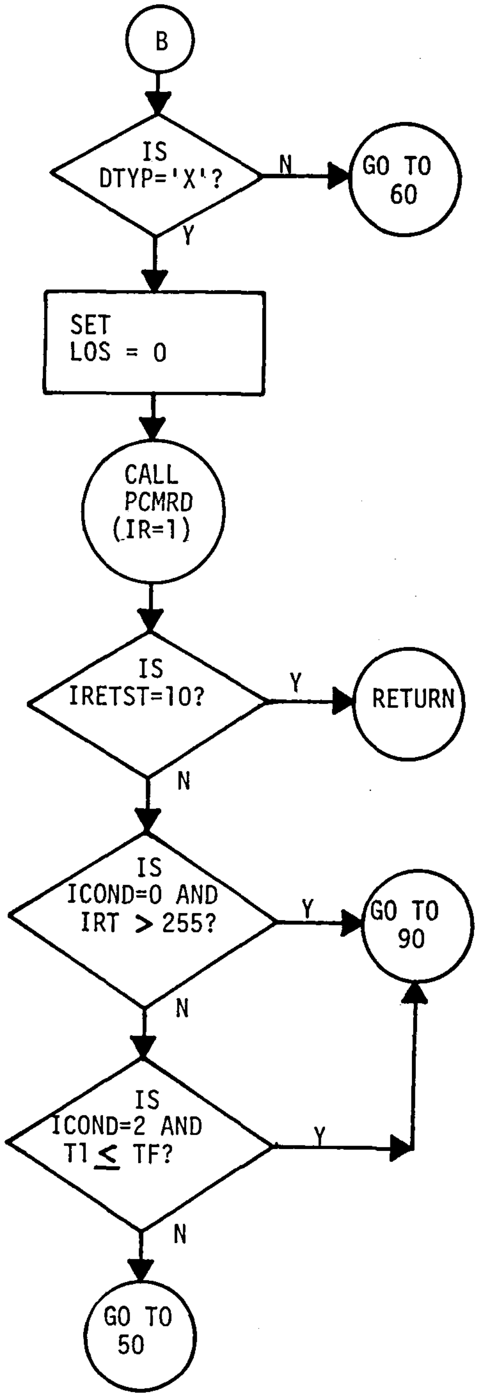


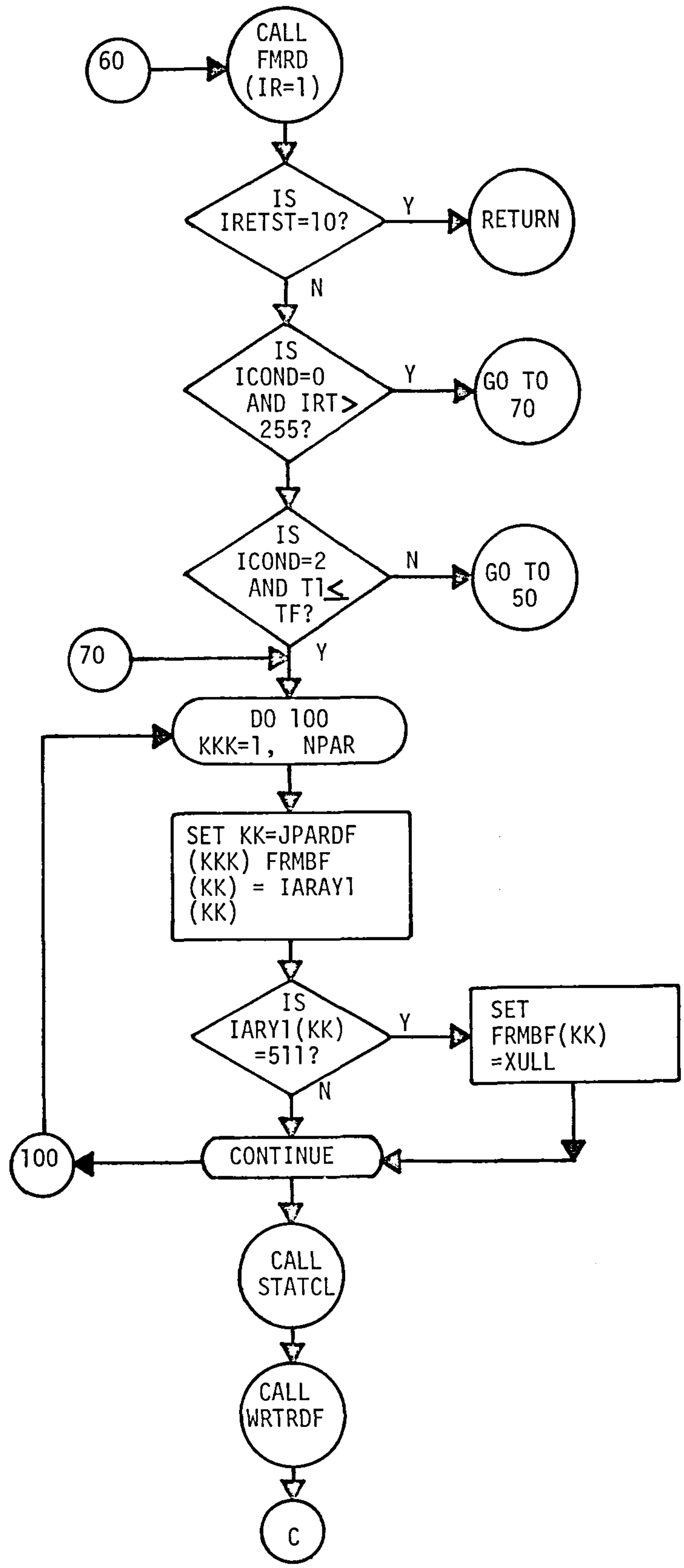
A
IS IRT ≤ 255?
Y
GO TO 10
N
20
IS DTYP = 'X'?
N
CALL FMRD (IR=1)
Y
CALL PCMRD (IR=1)
IS IRETST=10?
Y
RETURN
IS IRT ≤ 255?
N
GO TO 20
Y
RETURN

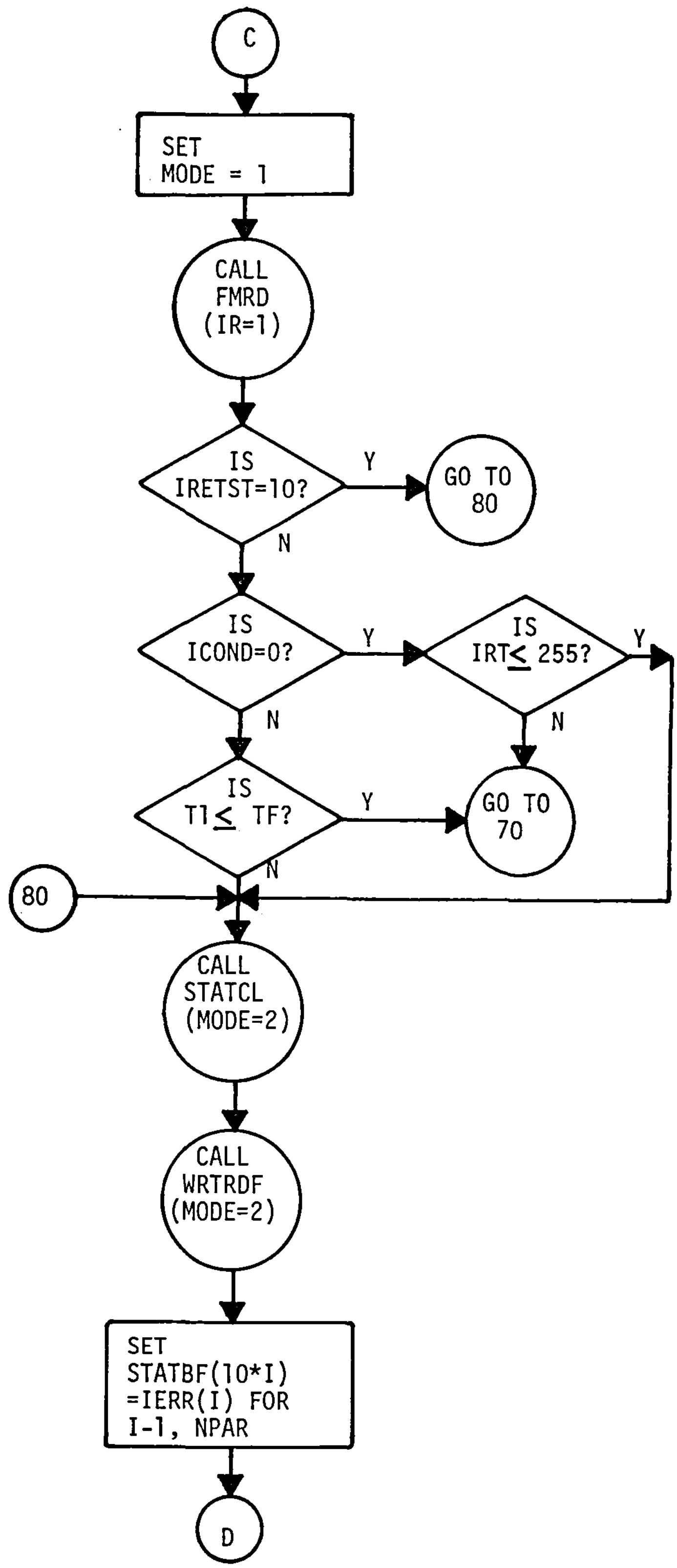


30
SET NEVENT = NEVENT + 1 STATBF(I,1=1,400) = XULL ISTADR=100* (NPASS-1) + NEVENT
WRITE (09'ISTADR) STATBE (STATISTICS FILE)
SET IADDR(NEVENT+1) =KADDR+1 ETO = 0. ETF = 0.
RETURN

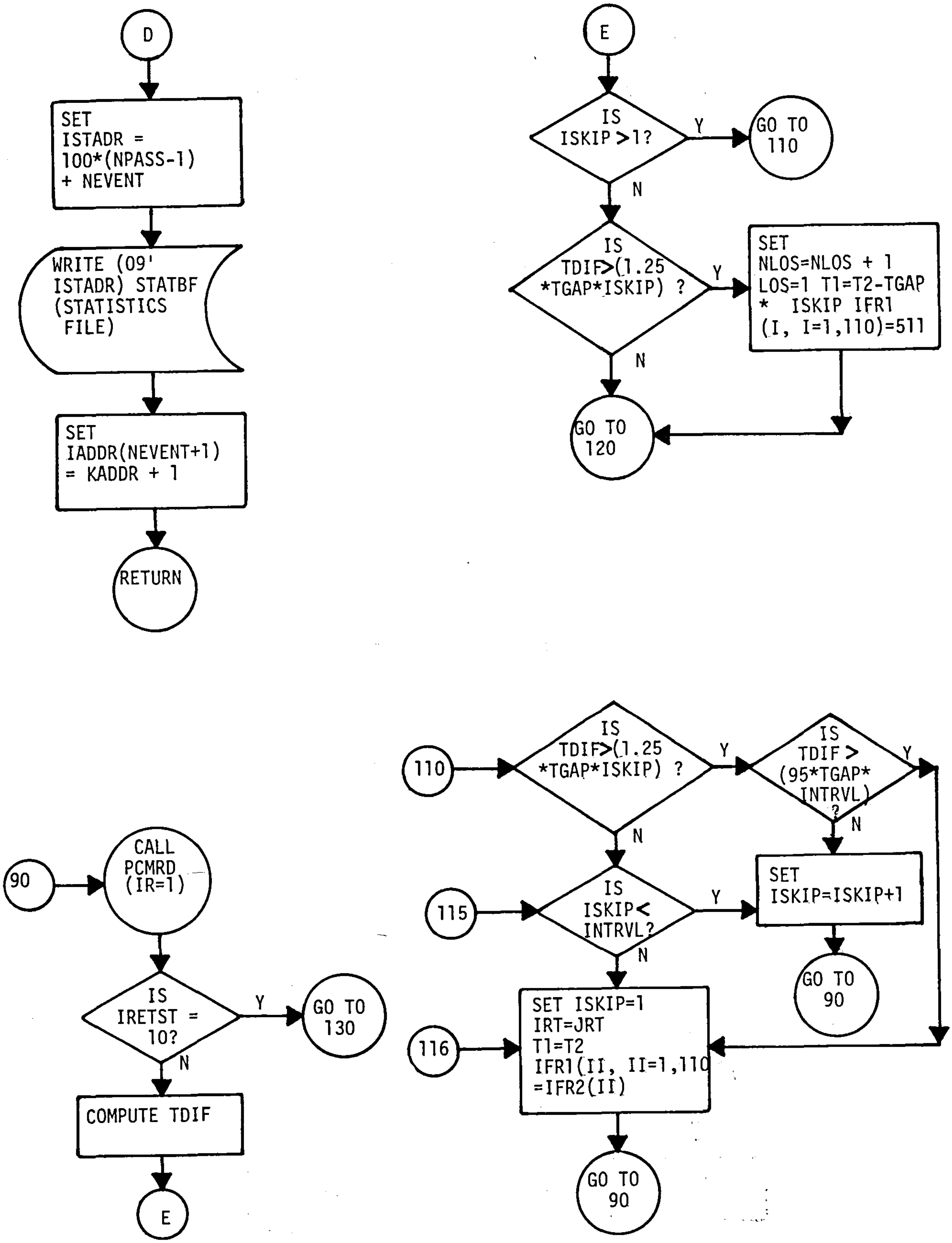

40
SET
NEVENT =
NEVENT + 1
IERR(I,I=1,40) = 0
IS
NEVENT>100?
Y
PRINT ERROR
MESSAGE
(FORMAT 100)
STOP
N
50
IS
DTYP='X'?
N
CALL
FMRD
(IR=0)
Y
CALL
PCMRD
(IR=0)
IS
IRETST=10?
Y
RETURN
N
IS
ICOND=0?
Y
IS
IRT > 255?
Y
N
N
IS
T1 ≤ T0?
Y
GO TO
50
N
SET
ISKIP = 1
MODE = 0
B

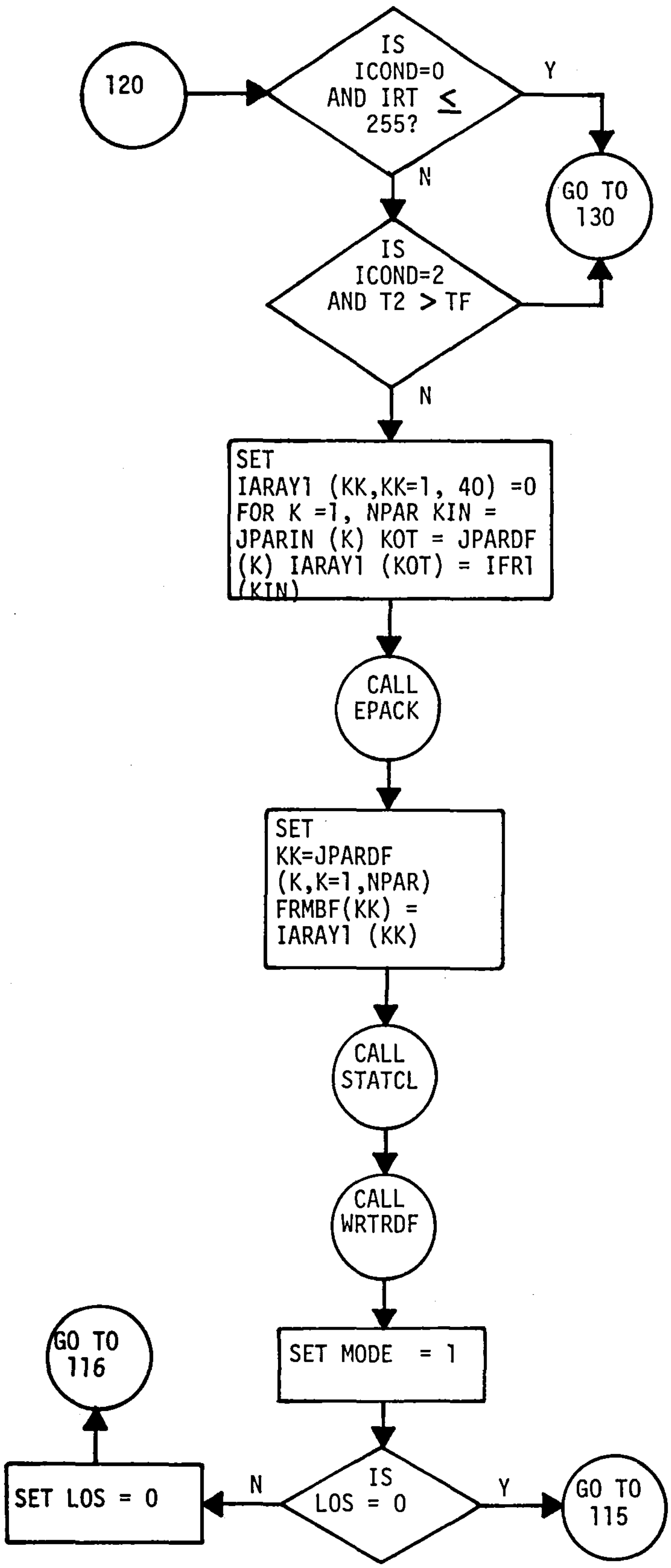

B
IS DTYP='X'?
N
GO TO 60
Y
SET LOS = 0
CALL PCMRD (IR=1)
IS IRETST=10?
Y
RETURN
N
IS ICOND=0 AND IRT > 255?
Y
GO TO 90
N
IS ICOND=2 AND T1 ≤ TF?
Y
N
GO TO 50

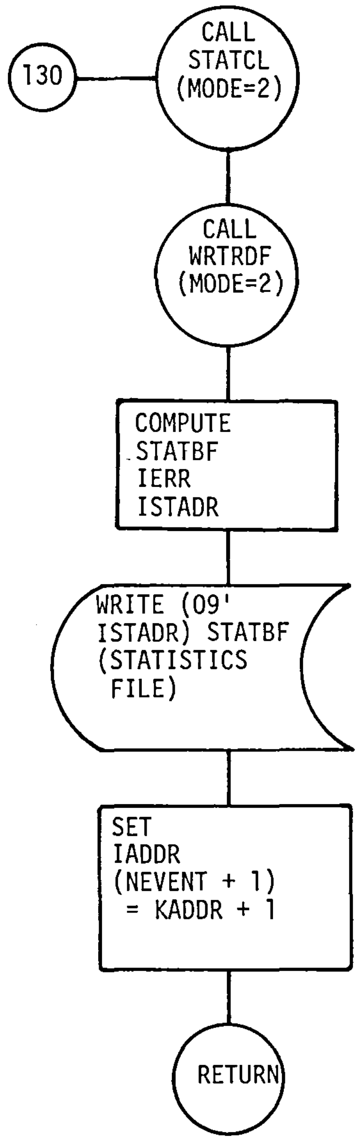

60
CALL FMRD (IR=1)
IS IRETST=10?
Y
RETURN
N
IS ICOND=0 AND IRT> 255?
Y
GO TO 70
IS ICOND=2 AND T1≤ TF?
N
GO TO 50
70
Y
DO 100 KKK=1, NPAR
SET KK=JPARDF (KKK) FRMBF (KK) = IARAY1 (KK)
IS IARY1(KK) =511?
Y
SET FRMBF(KK) =XULL
N
100
CONTINUE
CALL STATCL
CALL WRTRDF
C



C
SET
MODE = 1
CALL
FMRD
(IR=1)
IS
IRETST=10?
Y
GO TO
80
N
IS
ICOND=0?
Y
IS
IRT≤ 255?
Y
N
N
IS
T1≤ TF?
Y
GO TO
70
N
80
CALL
STATCL
(MODE=2)
CALL
WRTRDF
(MODE=2)
SET
STATBF(10*I)
=IERR(I) FOR
I-1, NPAR
D

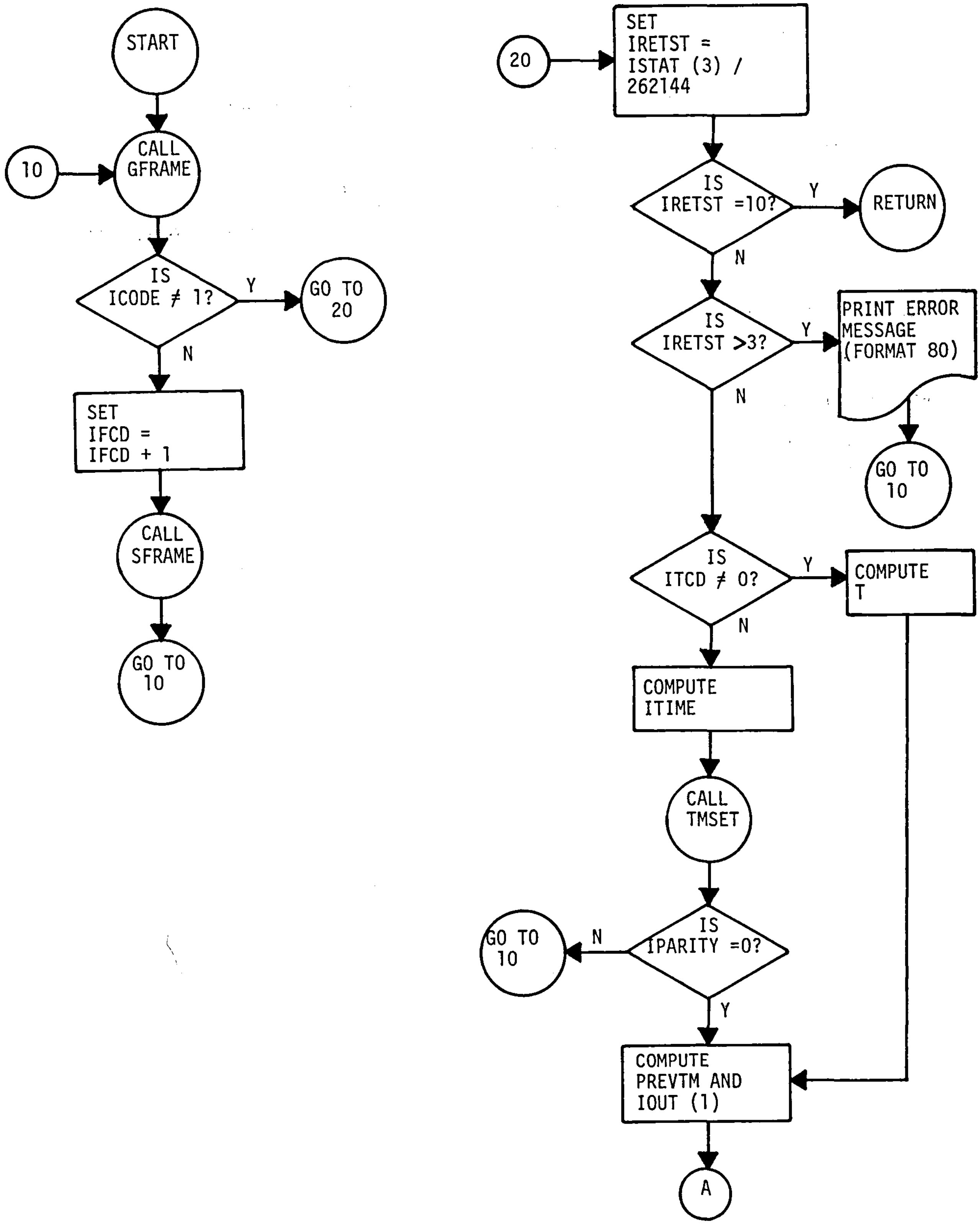

D
SET ISTADR = 100*(NPASS-1) + NEVENT
WRITE (09' ISTADR) STATBF (STATISTICS FILE)
SET IADDR(NEVENT+1) = KADDR + 1
RETURN
E
IS ISKIP >1?
Y
N
GO TO 110
IS TDIF>(1.25 *TGAP*ISKIP) ?
Y
N
SET NLOS=NLOS + 1 LOS=1 T1=T2-TGAP * ISKIP IFR1 (I, I=1,110)=511
GO TO 120
90
CALL PCMRD (IR=1)
IS IRETST = 10?
Y
N
GO TO 130
COMPUTE TDIF
E
110
IS TDIF>(1.25 *TGAP*ISKIP) ?
Y
N
IS TDIF > (95*TGAP* INTRVL) ?
Y
N
SET ISKIP=ISKIP+1
GO TO 90
115
IS ISKIP< INTRVL?
Y
N
116
SET ISKIP=1 IRT=JRT T1=T2 IFR1(II, II=1,110) =IFR2(II)
GO TO 90

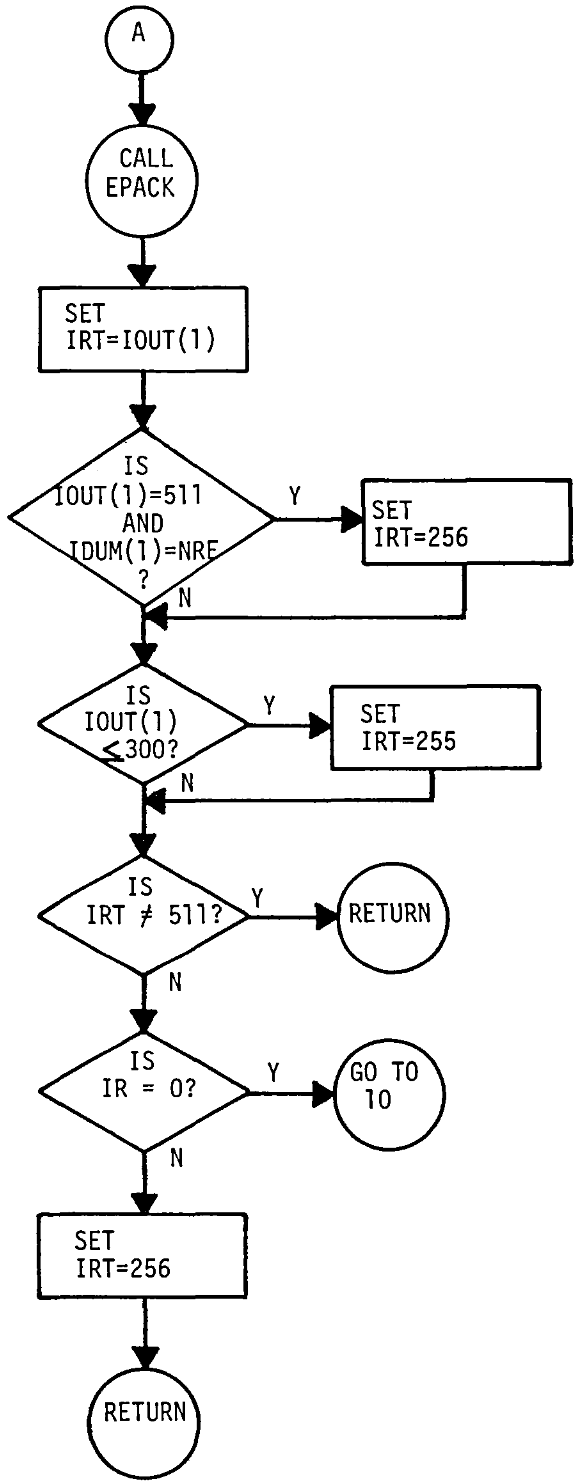

120
IS ICOND=0 AND IRT ≤ 255?
Y
N
GO TO 130
IS ICOND=2 AND T2 > TF
N
SET IARAY1 (KK,KK=1, 40) =0 FOR K =1, NPAR KIN = JPARIN (K) KOT = JPARDF (K) IARAY1 (KOT) = IFR1 (KIN)
CALL EPACK
SET KK=JPARDF (K,K=1,NPAR) FRMBF(KK) = IARAY1 (KK)
CALL STATCL
CALL WRTRDF
SET MODE = 1
GO TO 116
SET LOS = 0
N
IS LOS = 0
Y
GO TO 115


130
CALL STATCL (MODE=2)
CALL WRTRDF (MODE=2)
COMPUTE STATBF IERR ISTADR
WRITE (09' ISTADR) STATBF (STATISTICS FILE)
SET IADDR (NEVENT + 1) = KADDR + 1
RETURN

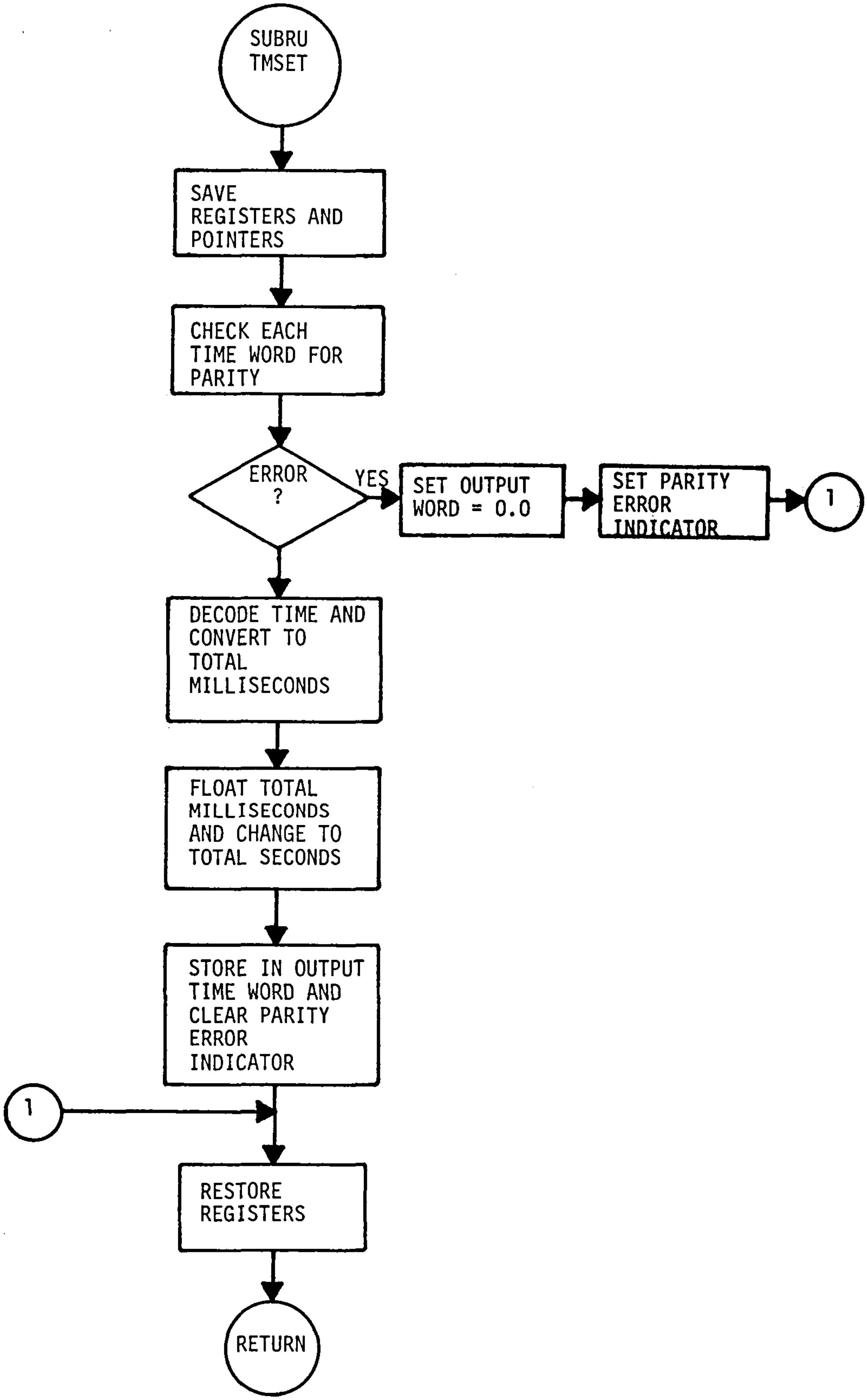
```
PCMR    PCMRD-READ PCM DATA
**********************SUBROUTINE PCMRD********************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       THIS ROUTINE READS THE INPUT PCM TAPE OR TAPES AND RETURNS
       REQUIRED DATA.


*******METHOD -
       THE PCM TAPE MUST BE OF STANDARD TELEVENT FORMAT BECAUSE DATA
       IS DECODED BY GFRAME MODULE.  IF TAPE IS MULTI-REEL,
       SEQUENTIAL TAPE IS INITIALIZED BY SFRAME BEFORE DECODED BY
       GFRAME.

       THE STATUS OF EACH RECORD IS CHECKED FOR POSSIBLE PARITY ON
       READ AND END OF FILE DESIGNATOR.

       GROUND STATION TIME IS COMPUTED FROM 5 WORDS OF TIME (ITIME)
       RETURNED BY GFRAME.  AIRBOURNE TIME IS COMPUTED FROM WORDS 2
       THROUGH 6 OF THE DATA ARRAY (IFR) RETURNED BY GFRAME.

       THE RUN TONE IS FOUND IN IFR(110) AND ALTERED SO THAT A VALUE
       OF 256 REPRESENTS RUN TONE SIGNAL BEING ON.


*******INPUT -

       ITCD         -CODE DESIGNATING WHICH TIME WORD TO USE
                    -=0,AIRBOURNE TIME
                    -=1,GROUND STATION TIME

       IR           -CODE DESIGNATING STATUS OF BURST
                    -=0,SEARCHING FOR BEGINNING OF NEXT BURST
                    -=1,WITHIN A DATA BURST (RUN TONE IS ON OR STOP TIME
                    -   OF BURST HAS NOT BEEN REACHED)

       IFCD         -FILE CODE OF INPUT TAPE


*******OUTPUT -

       T            -TIME (GROUND STATION OR AIRBOURNE)

       IRT          -RUN TONE
                    -=2560R MORE,RUN TONE IS ON
                    -=LESS THAN 256, RUN TONE IS OFF
                    -=511, FULL SCALE
```

```
IRETST        -READ STATUS OF FRAME
              -=1,2, OR 3,GOOD READ
              -=10,END OF FILE ON TAPE
              -=OTHER, ERROR ENCOUNTERED, DATA IGNORED

IFR           -DECODED DATA ARRAY

ICODE         -CODE TO DESIGNATE END OF TAPE
              -=0,END OF TAPE NOT REACHED
              -=1,END OF TAPE ENCOUNTERED, INITIALIZE SEQUENTIAL
              -   TAPE

*******RESTRICTIONS -
       NONE

*******SUBPROGRAMS REQUIRED -
       GFRAME
       SFRAME
       TMSET
```


START
10
CALL GFRAME
IS ICODE ≠ 1?
Y
GO TO 20
N
SET IFCD = IFCD + 1
CALL SFRAME
GO TO 10
20
SET IRETST = ISTAT (3) / 262144
IS IRETST =10?
Y
RETURN
N
IS IRETST >3?
Y
PRINT ERROR MESSAGE (FORMAT 80)
GO TO 10
N
IS ITCD ≠ 0?
Y
COMPUTE T
N
COMPUTE ITIME
CALL TMSET
IS IPARITY =0?
N
GO TO 10
Y
COMPUTE PREVTM AND IOUT (1)
A


A
CALL EPACK
SET IRT=IOUT(1)
IS IOUT(1)=511 AND IDUM(1)=NRE ?
Y
SET IRT=256
N
IS IOUT(1) ≤300?
Y
SET IRT=255
N
IS IRT ≠ 511?
Y
RETURN
N
IS IR = 0?
Y
GO TO 10
N
SET IRT=256
RETURN

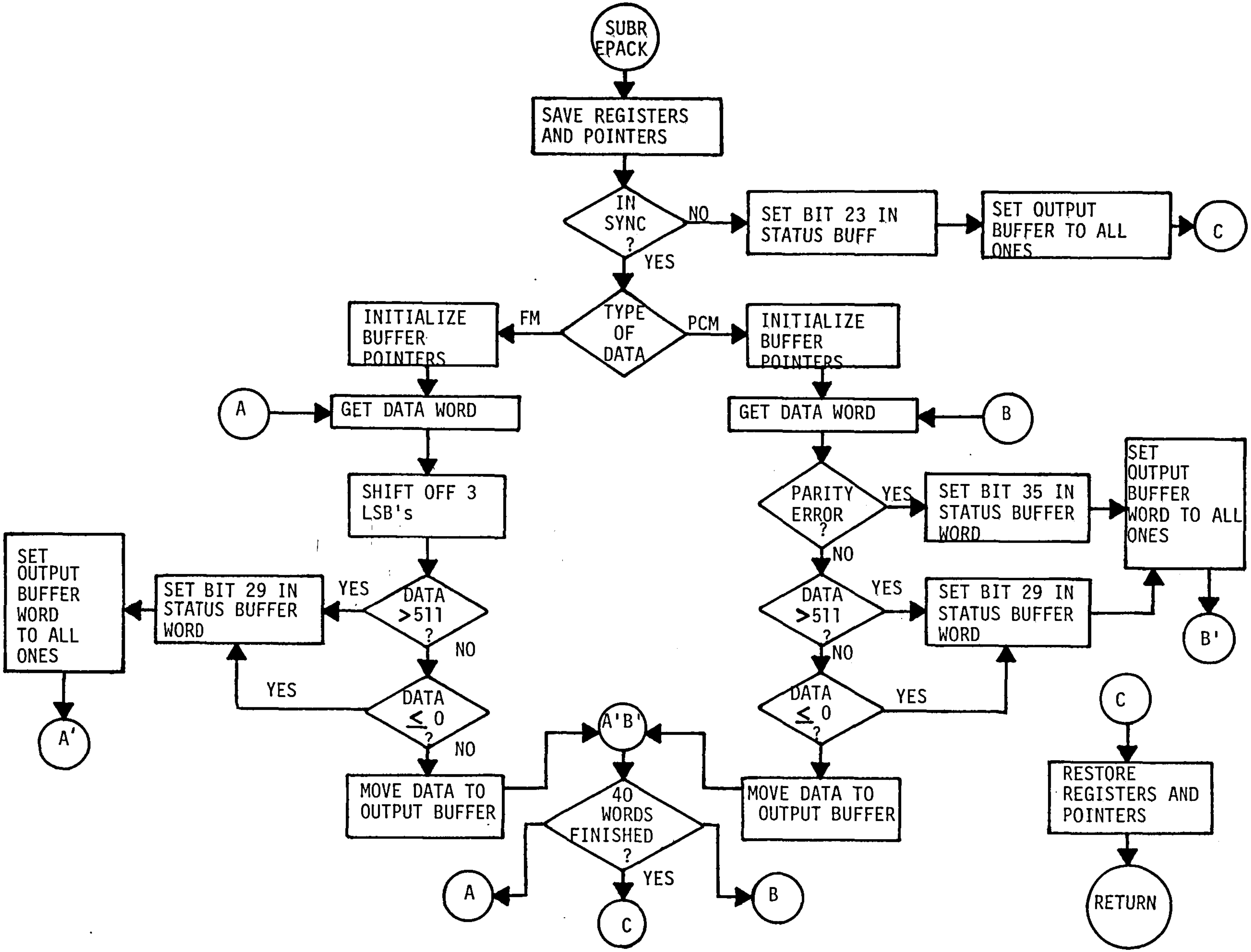
```
***************************SUBROUTINE TMSET****************************

 PROGRAM IDENTIFICATION
     PROGRAM NAME - TMSET
     PROGRAM NO.  - 1.1.2320
     AUTHOR       - DAVID L. DAVIS

     COMPUTER     - HW 625/635
     MEMORY       - 101 WORDS
     PERIPHERALS  - NONE
     LANGUAGE     - GMAP

 PURPOSE

     TMSET DECODES AND PARITY CHECKS
     THE AIRBORNE TIME, AND RETURNS
     IT AS A FLOATING POINT VALUE
     IN TOTAL SECONDS.

 METHOD

     TIME IS CONVERTED TO TOTAL
     MILLISECONDS, FLOATED,AND
     DIVIDED BY 1000 TO GIVE TOTAL SECONDS.
 INPUT/OUTPUT

     CALLING SEQUENCE<

      CALL TMSET(A,B,C) WHERE

        A = 5 WORD ARRAY OF INPUT TIME
        B = OUTPUT TIME IN SECONDS (FLOATING POINT).
        C = PARITY CHECK FLAG

        IF C = #0# PARITY OK
           C = #1# PARITY ERROR
```


SUBRU TMSET
SAVE REGISTERS AND POINTERS
CHECK EACH TIME WORD FOR PARITY
ERROR ?
YES
SET OUTPUT WORD = 0.0
SET PARITY ERROR INDICATOR
1
DECODE TIME AND CONVERT TO TOTAL MILLISECONDS
FLOAT TOTAL MILLISECONDS AND CHANGE TO TOTAL SECONDS
STORE IN OUTPUT TIME WORD AND CLEAR PARITY ERROR INDICATOR
1
RESTORE REGISTERS
RETURN

***************************SUBROUTINE EPACK*****************************

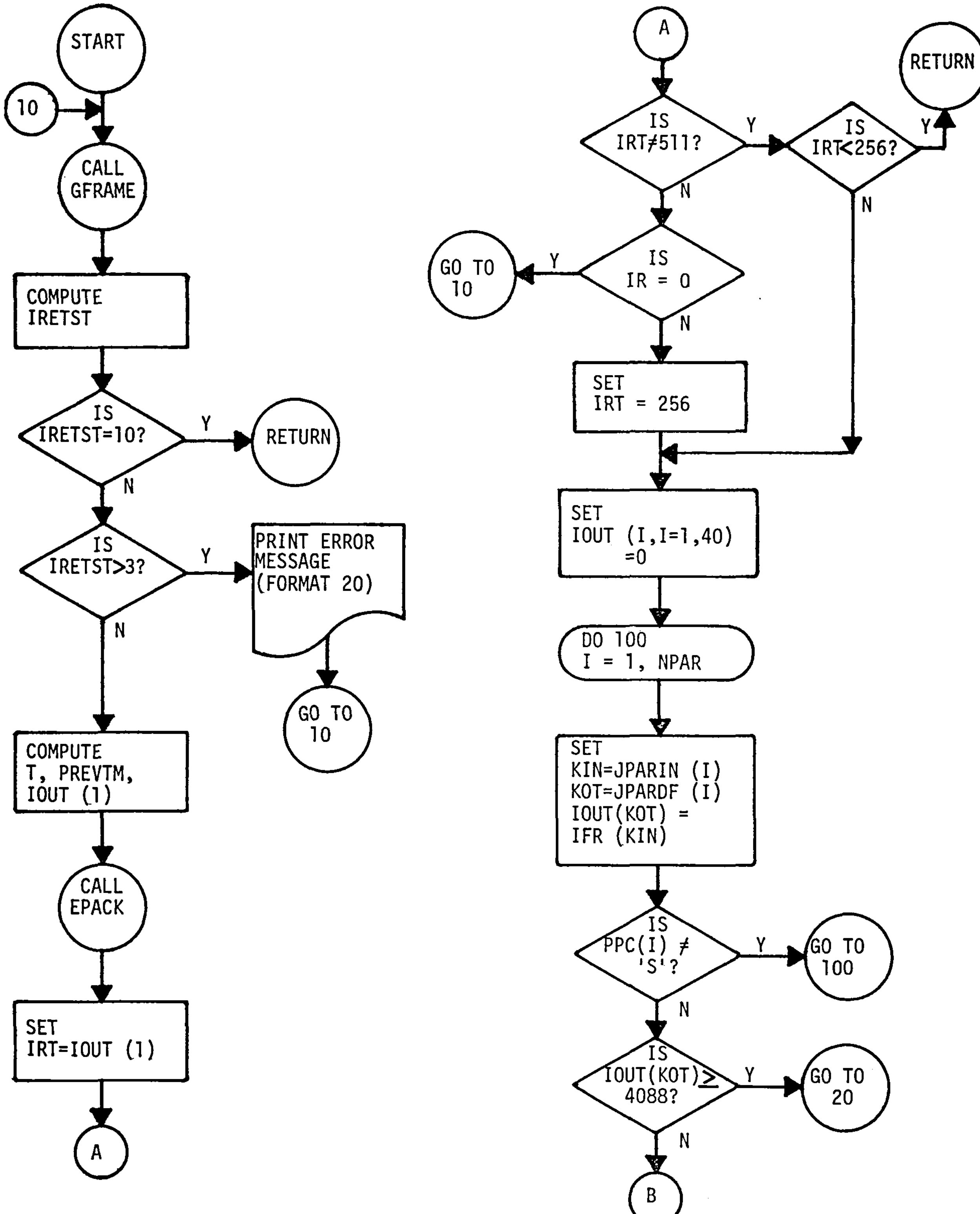
```
PROGRAM IDENTIFICATION
     PROGRAM NAME - EPACK
     PROGRAM NO.  - 1.1.2320
     AUTHOR       - DAVID L. DAVIS

     COMPUTER     - HW 625/635
     MEMORY       - 105 WORDS
     PERIPHERALS  - NONE
     LANGUAGE     - GMAP

PURPOSE

     EPACK EDITS AND PACKS A 40 WORD
     INPUT ARRAY INTO A 10 WORD OUTPUT
     ARRAY.

METHOD
METH
METHPCM DATA<
      SYNC FLAG IS CHECKED. IF OUT OF SYNC
      BIT 23 IN ALL WORDS OF THE STATUS
      BUFFER IS SET TO A ONE (#1#), AND
      THE OUTPUT BUFFER IS SET TO MAXIMUM
      DATA VALUE. IF THE DATA IS IN SYNC,
      PARITY IS THEN CHECKED. WHEN A PARITY
      ERROR IS DETECTED, BIT 35 IN THE
      CORRESPONDING STATUS WORD IS SET AND
      MAXIMUM DATA VALUE IS INSERTED IN
      THE PROPER POSITION OF THE OUTPUT
      BUFFER. IF NO SYNC OR PARITY ERRORS
      ARE DETECTED THE DATA RANGE IS CHECKED.
      FOR DATA BETWEEN 0 AND 511 (EXCLUSIVE)
      THE DATA VALUE IS MOVED TO THE PROPER
      WHEN DATA IS NOT WITHIN THIS RANGE,
      POSITION IN THE 10 WORD OUTPUT BUFFER.
      WHEN DATA IS NOT WITHIN THIS RANGE,
      BIT 29 IN THE CORRESPONDING STATUS
      WORD IS SET AND THE DATA BITS ARE SET
      TO ONES (#1#) IN BOTH THE INPUT AND
      OUTPUT BUFFERS.

     FM DATA<
      SINCE THERE IS NO PARITY BIT WITH FM
      DATA NO PARITY CHECK IS PERFORMED.
      HOWEVER, ALL OTHER CHECKS ARE THE SAME
      AS FOR PCM DATA.
INPUT/OUTPUT
```

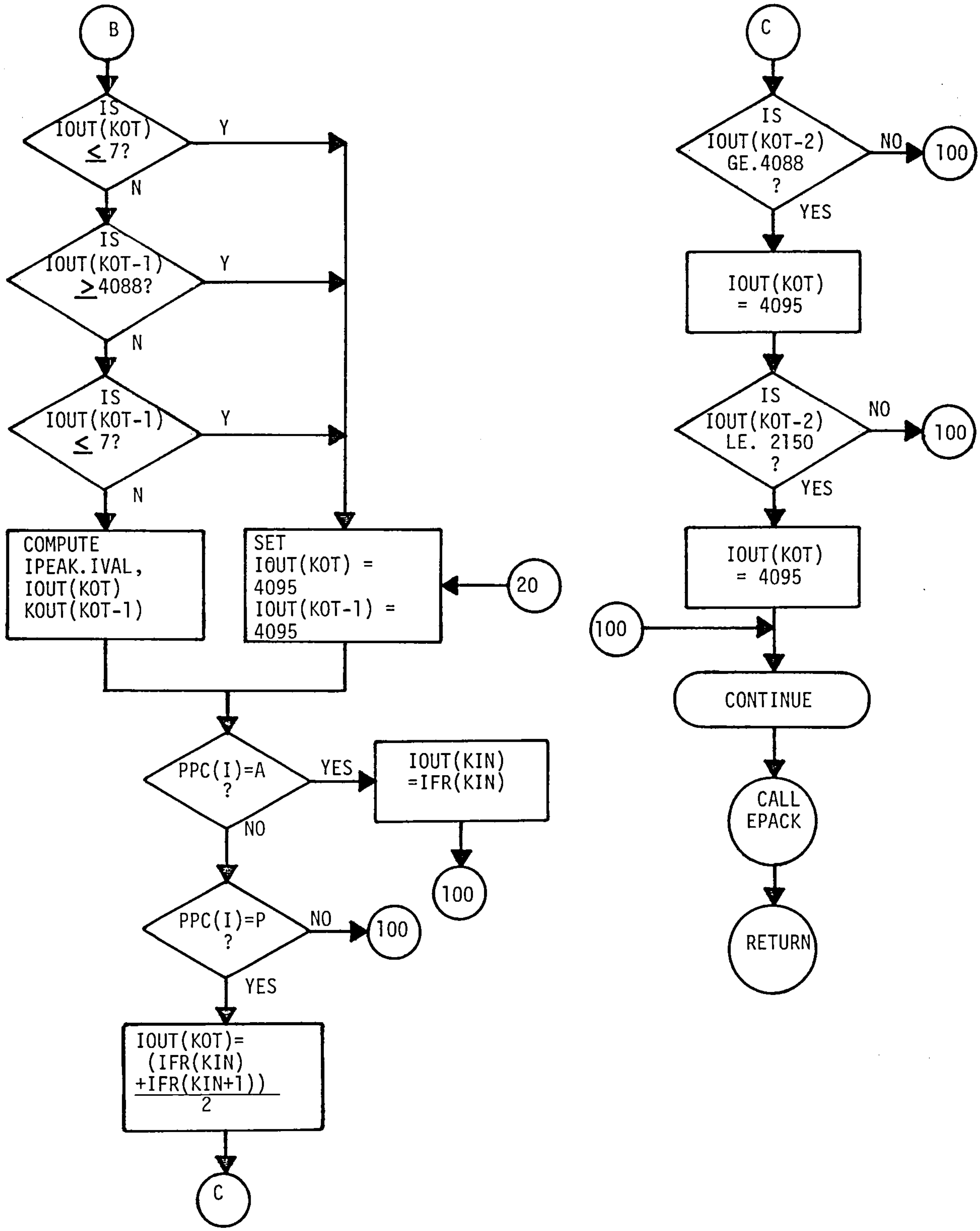
```
CALLING SEQUENCE<

 CALL EPACK(A,B,C,D,E) WHERE

   A = FIRST WORD ADDRESS OF INPUT BUFFER
   B = FIRST WORD ADDRESS OF STATUS BUFFER
   C = FIRST WORD ADDRESS OF OUTPUT BUFFER
   D = DATA SYNC FLAG
   E = DATA TYPE

INPUT BUFFER - 40 WORD ARRAY WITH DATA IN
               BITS 27 THRU 35 OF EACH WORD.


STATUS BUFFER - 40 WORD ARRAY WITH BITS
                23, 29, AND 35 USEDTO
                DESCRIBE THE CONDITION OF
                THE CORRESPONDING INPUT
                BUFFER WORD.

                     BIT 23 = I - LOSS OF SYNC
                     BIT 29 = 1 - DATA RANGE ERROR
                     BIT 35 = 1 - PARITY ERROR
OUTPUT BUFFER - 10 WORD ARRAY WITH 4
                9 BIT DATA VALUES PER
                WORD.

DATA SYNC FLAG - 1 = LOSS OF SYNC
                 0 = SYNC

DATA TYPE FLAG - 1 = FM DATA
                 0 = PCM DATA
```


SUBR EPACK
SAVE REGISTERS AND POINTERS
IN SYNC ?
NO
SET BIT 23 IN STATUS BUFF
SET OUTPUT BUFFER TO ALL ONES
C
YES
TYPE OF DATA
FM
PCM
INITIALIZE BUFFER POINTERS
INITIALIZE BUFFER POINTERS
A
GET DATA WORD
GET DATA WORD
B
SHIFT OFF 3 LSB's
PARITY ERROR ?
YES
SET BIT 35 IN STATUS BUFFER WORD
SET OUTPUT BUFFER WORD TO ALL ONES
NO
SET OUTPUT BUFFER WORD TO ALL ONES
SET BIT 29 IN STATUS BUFFER WORD
YES
DATA >511 ?
NO
DATA >511 ?
YES
SET BIT 29 IN STATUS BUFFER WORD
B'
NO
YES
DATA ≤ 0 ?
NO
DATA ≤ 0 ?
YES
A'
C
A'B'
MOVE DATA TO OUTPUT BUFFER
MOVE DATA TO OUTPUT BUFFER
40 WORDS FINISHED ?
YES
RESTORE REGISTERS AND POINTERS
A
C
B
RETURN

```
FMRD    FMRD-READ FM DATA
**********************SUBROUTINE FMRD**********************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       THIS ROUTINE READS THE INPUT FM TAPE AND RETURNS REQUIRED
       DATA.


*******METHOD -
       THE FM TAPE MUST BE OF STANDARD TELEVENT FORMAT BECAUSE DATA
       IS DECODED BY GFRAME MODULE.

       THE STATUS OF EACH RECORD IS CHECKED FOR POSSIBLE PARITY ON
       READ.AND END OF FILE DESIGNATOR.

       AIRBOURNE TIME IS THE ONLY TIME RECORDED AND IS COMPUTED FROM
       5 WORDS OF TIME (ITIME).RETURNED BY GFRAME.

       THE RUN TONE SIGNAL IS FOUND IN IFR (1) OF THE DATA ARRAY.

       VIBRATORY AND STEADY COMPONENTS ARE COMPUTED FROM INPUT PEAK
       AND VALLEY DATA.


*******INPUT -

       IR          -CODE DESIGNATING STATUS OF BURST
                   -=0,SEARCHING FOR BEGINNING OF NEXT BURST
                   -=1,WITHIN A DATA BURST (RUN TONE IS ON OR STOP
                   -   TIME OF BURST HAS NOT BEEN REACHED)


*******OUTPUT -

       T           -TIME (AIRBOURNE TIME)

       IRT         -RUN TONE
                   -=511,FULL SCALE
                   -=256 OR MORE, RUN TONE IS ON
                   -=LESS THAN 256, RUN TONE IS OFF

       IRETST      -READ STATUS OF FRAME
                   -=1,2, OR 3,GOOD READ
                   -=10,END OF FILE ON TAPE
                   -=OTHER, ERROR ENCOUNTERED, DATA IGNORED
```

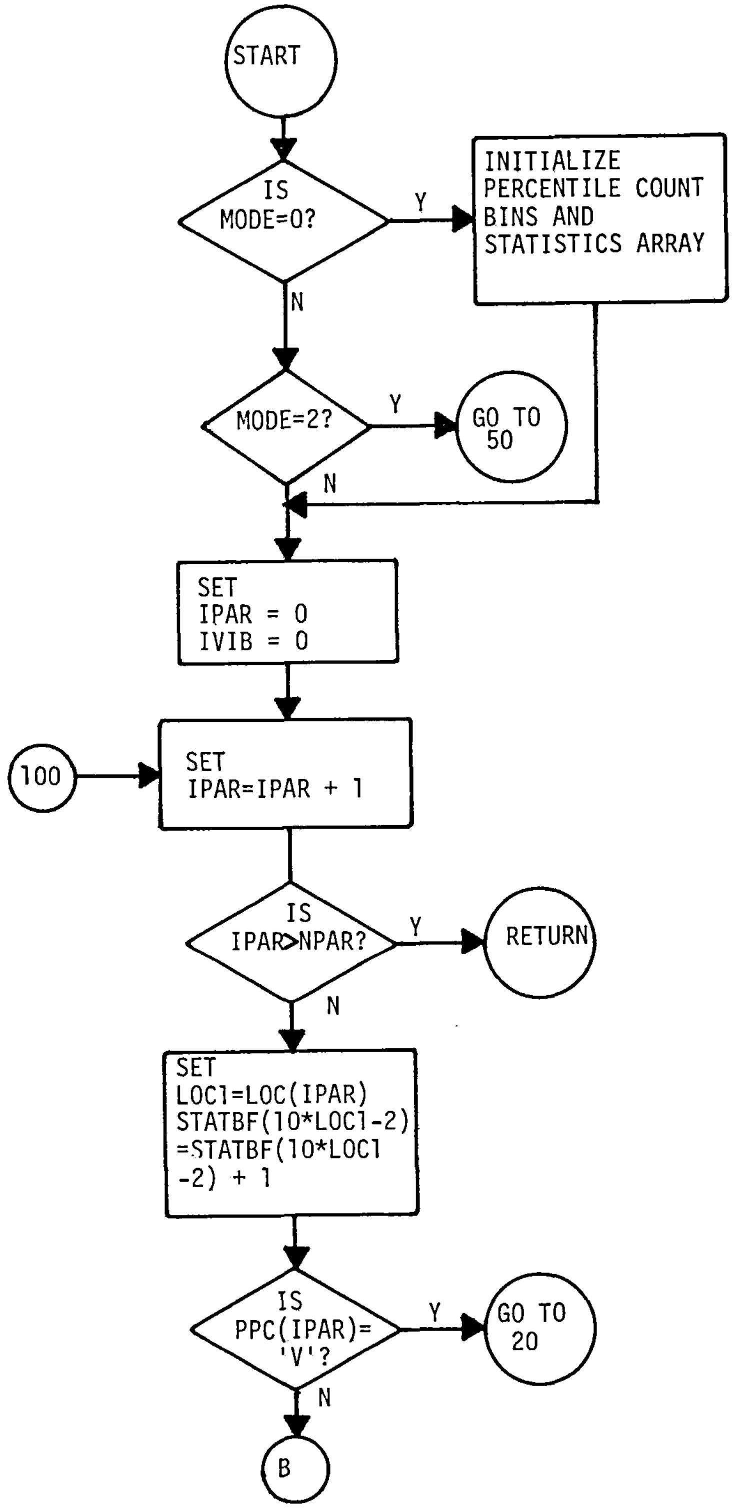
```
        IOUT        -DECODED DATA ARRAY CONTAINING VIBRATORY AND
                    -STEADY COMPONENTS

        IERR        -ERROR RETURN FROM EPACK

        IPCK        -PACKED DATA ARRAY FROM EPACK

*******RESTRICTIONS -
        NONE

*******SUBPROGRAMS REQUIRED -
        GFRAME
        EPACK
```

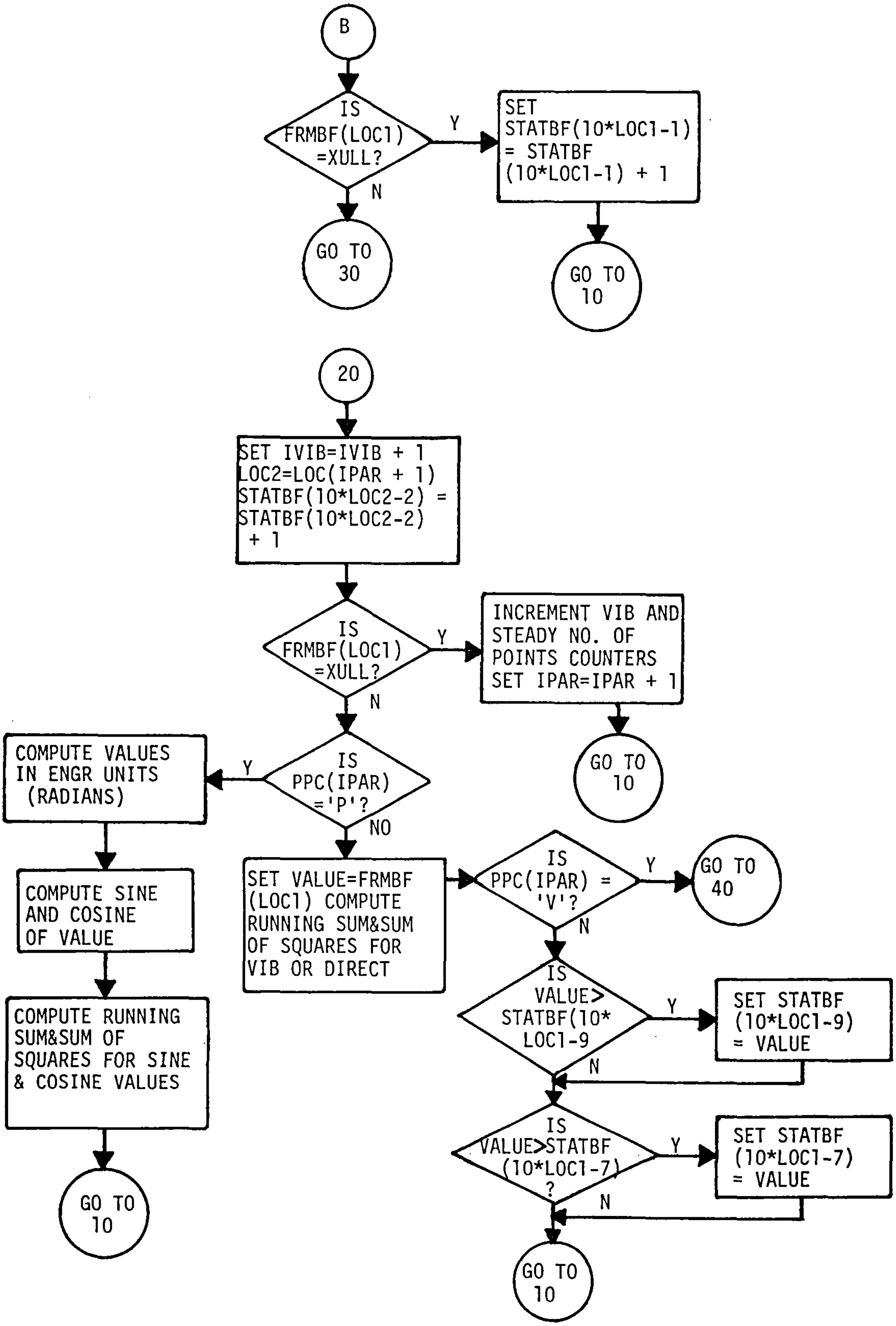

START
10
CALL GFRAME
COMPUTE IRETST
IS IRETST=10?
Y
RETURN
N
IS IRETST>3?
Y
PRINT ERROR MESSAGE (FORMAT 20)
GO TO 10
N
COMPUTE T, PREVTM, IOUT (1)
CALL EPACK
SET IRT=IOUT (1)
A
A
IS IRT≠511?
Y
IS IRT<256?
Y
RETURN
N
N
IS IR = 0
Y
GO TO 10
N
SET IRT = 256
SET IOUT (I,I=1,40) =0
DO 100 I = 1, NPAR
SET KIN=JPARIN (I) KOT=JPARDF (I) IOUT(KOT) = IFR (KIN)
IS PPC(I) ≠ 'S'?
Y
GO TO 100
N
IS IOUT(KOT)≥ 4088?
Y
GO TO 20
N
B

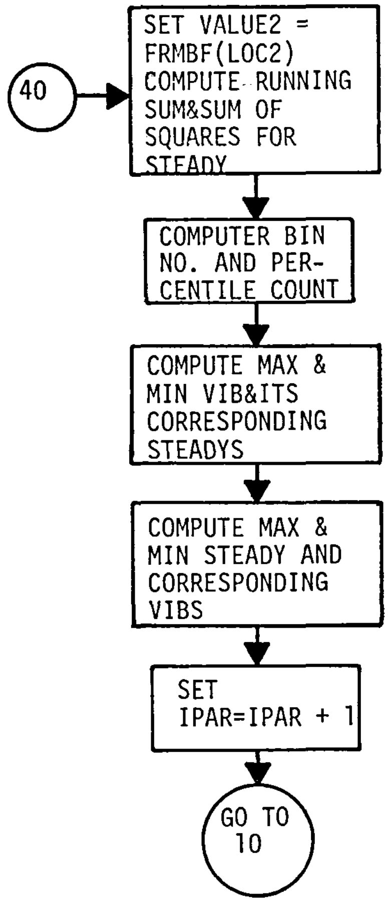

B
IS
IOUT(KOT)
≤ 7?
Y
N
IS
IOUT(KOT-1)
≥ 4088?
Y
N
IS
IOUT(KOT-1)
≤ 7?
Y
N
COMPUTE
IPEAK.IVAL,
IOUT(KOT)
KOUT(KOT-1)
SET
IOUT(KOT) =
4095
IOUT(KOT-1) =
4095
20
PPC(I)=A
?
YES
IOUT(KIN)
=IFR(KIN)
100
NO
PPC(I)=P
?
NO
100
YES
IOUT(KOT)=
(IFR(KIN)
+IFR(KIN+1))
2
C
C
IS
IOUT(KOT-2)
GE.4088
?
NO
100
YES
IOUT(KOT)
= 4095
IS
IOUT(KOT-2)
LE. 2150
?
NO
100
YES
IOUT(KOT)
= 4095
100
CONTINUE
CALL
EPACK
RETURN

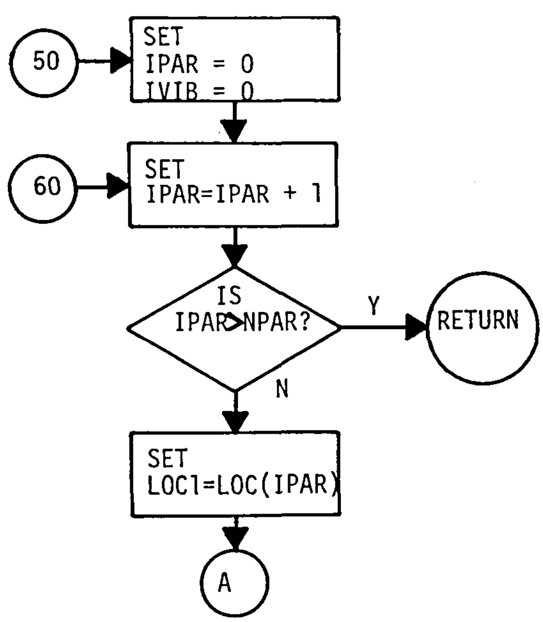
```
STAT    STATCL-COMPILE STATISTICS
***********************SUBROUTINE STATCL*******************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       THIS ROUTINE COMPILES AND COMPUTES STATISTICS AND OUTPUTS ONTO
       THE STATISTICS FILE.


*******METHOD -
       STATISTICS ARE ACCUMULATED DURING THE PROCESSING OF A DATA
       BURST.  SUCH STATS ARE RUNNING SUM, RUNNING SUM OF SQUARES,
       MAX, MIN, AND BIN COUNT FOR 95 PERCENTILE OF VIBRATORY. AND
       AT THE COMPLETION OF EACH BURST, STATS ARE COMPUTED, STORED
       IN AN ARRAY AND OUTPUT ON STATISTICS FILE IN EVCHK.
       IF THERE IS ONLY ONE DATA POINT OR NO DATA POINTS FOR A BURST,
       NULLS ARE STORED IN THE STATISTICS FOR THAT BURST.


*******INPUT -

       MODE       -CODE TO DESIGNATE DATA STATUS
                  -=0,INITIALIZE STATISTICS ARRAY (FIRST FRAME OF
                  -   BURST)
                  -=1,STATISTICS BEING ACCUMULATED
                  -=2,FINALIZE STATISTICS (LAST FRAME OF BURST HAS
                  -   BEEN STORED)

       FRMBF      -FLOATING POINT EQUIVALENT OF DATA ARRAY

       NPAR       -NUMBER OF PARAMETERS IN PASS

       PPC        -PRE-PROCESSING CODE OF SENSOR

       LOC        -40-WORD ARRAY OF LOCATIONS WITHIN FRMBF FOR EACH
                  -PARAMETER

       STATBF     -400-WORD STATISTICS BUFFER WITH 10 WORDS FOR EACH
                  -OF 40 PARAMETERS FOR BURST

       STATBF(1)  -MAXIMUM

       STATBF(2)  -CORRESPONDING VIBRATORY FOR MAXIMUM STEADY OR
                  -CORRESPONDING STEADY FOR MAXIMUM VIBRATORY

       STATBF(3)  -MINIMUM
```

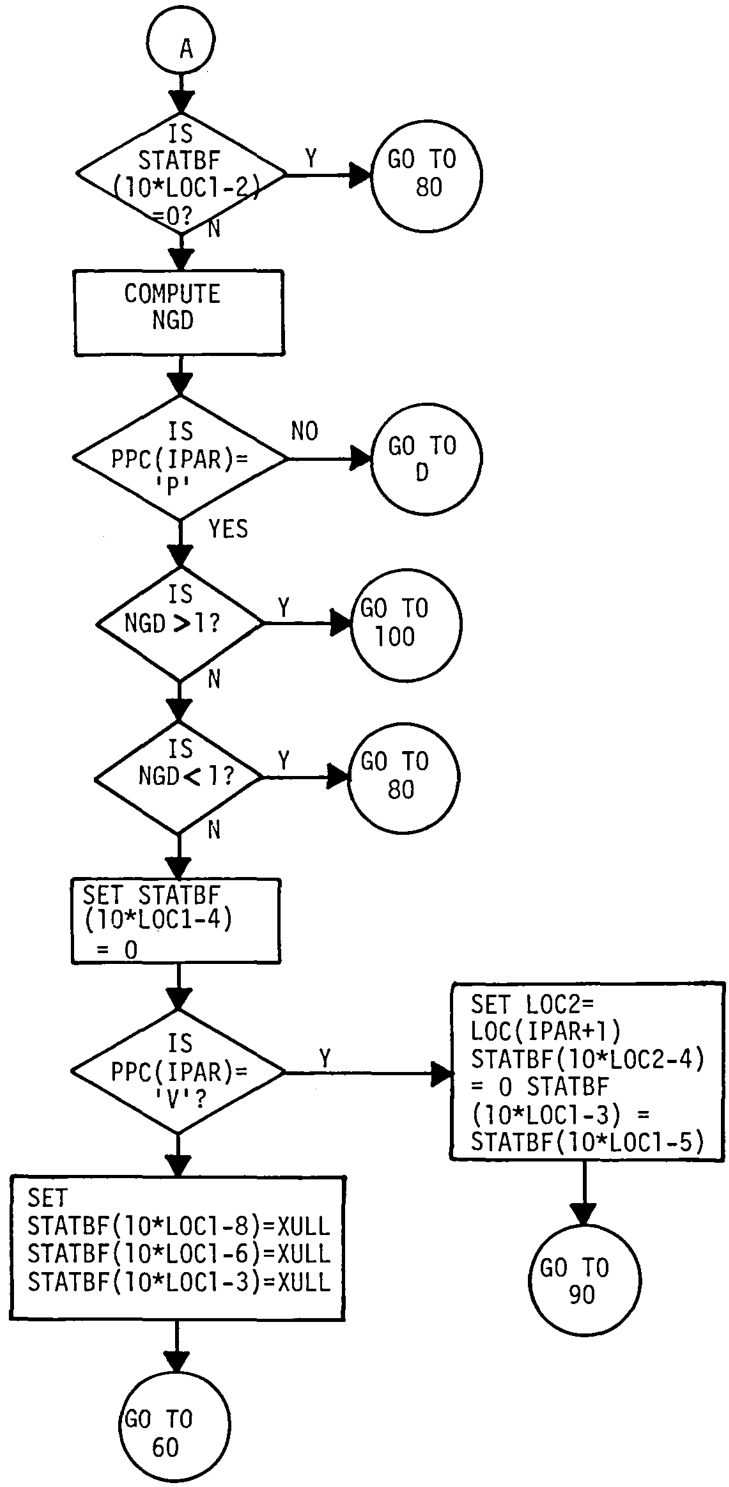
```
        STATBF(4)   -CORRESPONDING VIBRATORY FOR MINIMUM STEADY OR
                    -CORRESPONDING STEADY FOR MINIMUM VIBRATORY

        STATBF(5)   -MEAN

        STATBF(6)   -STANDARD DEVIATION

        STATBF(7)   -95 PERCENTILE

        STATBF(8)   -NUMBER OF POINTS IN BURST

        STATBF(9)   -NUMBER OF ERROR POINTS

        STATBF(10)  -ERROR CODE (OUTPUT FROM EPACK)

*******RESTRICTIONS -
        NONE

*******SUBPROGRAMS REQUIRED -
        NONE
```

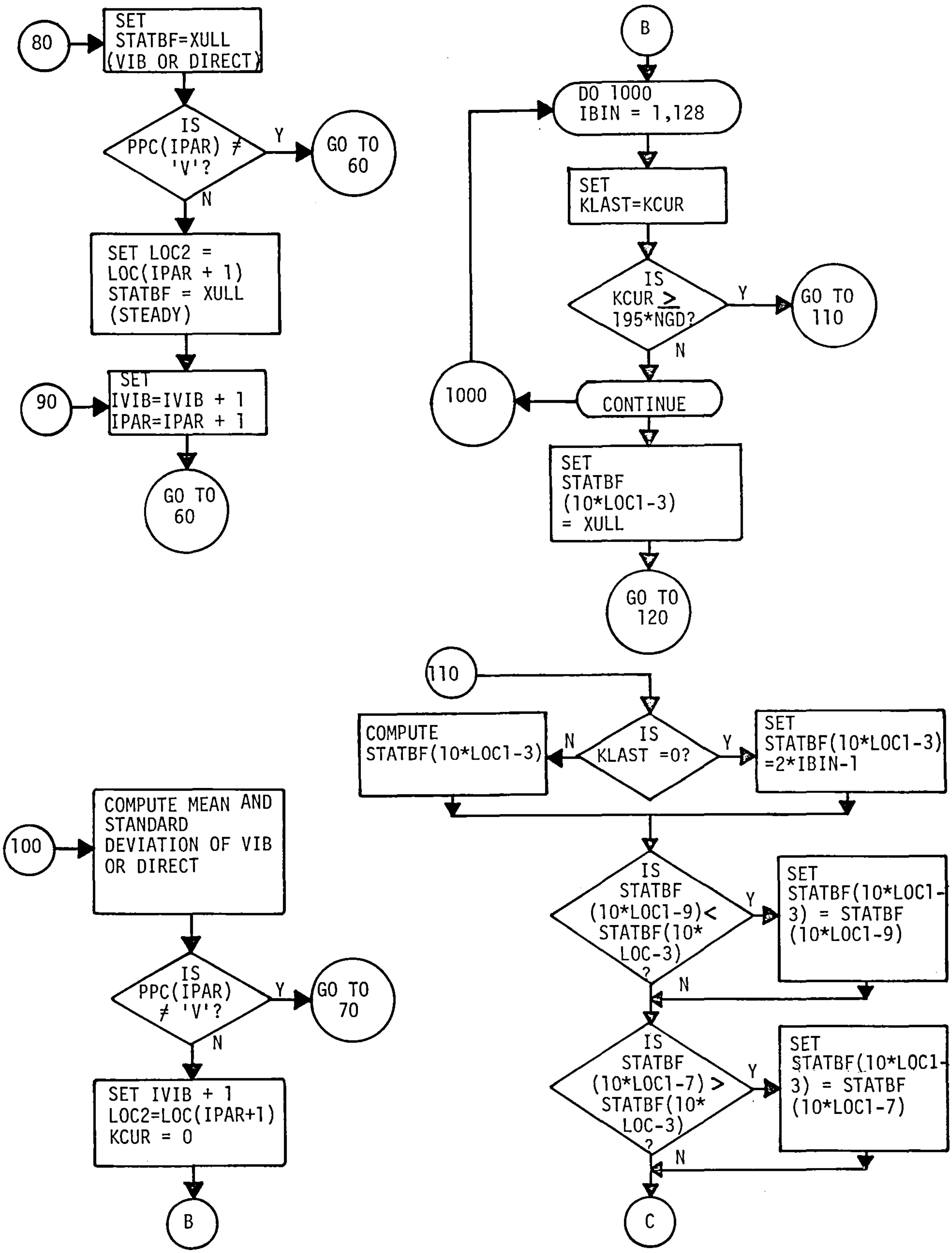

START
IS
MODE=0?
Y
INITIALIZE
PERCENTILE COUNT
BINS AND
STATISTICS ARRAY
N
MODE=2?
Y
GO TO
50
N
SET
IPAR = 0
IVIB = 0
100
SET
IPAR=IPAR + 1
IS
IPAR>NPAR?
Y
RETURN
N
SET
LOC1=LOC(IPAR)
STATBF(10*LOC1-2)
=STATBF(10*LOC1
-2) + 1
IS
PPC(IPAR)=
'V'?
Y
GO TO
20
N
B

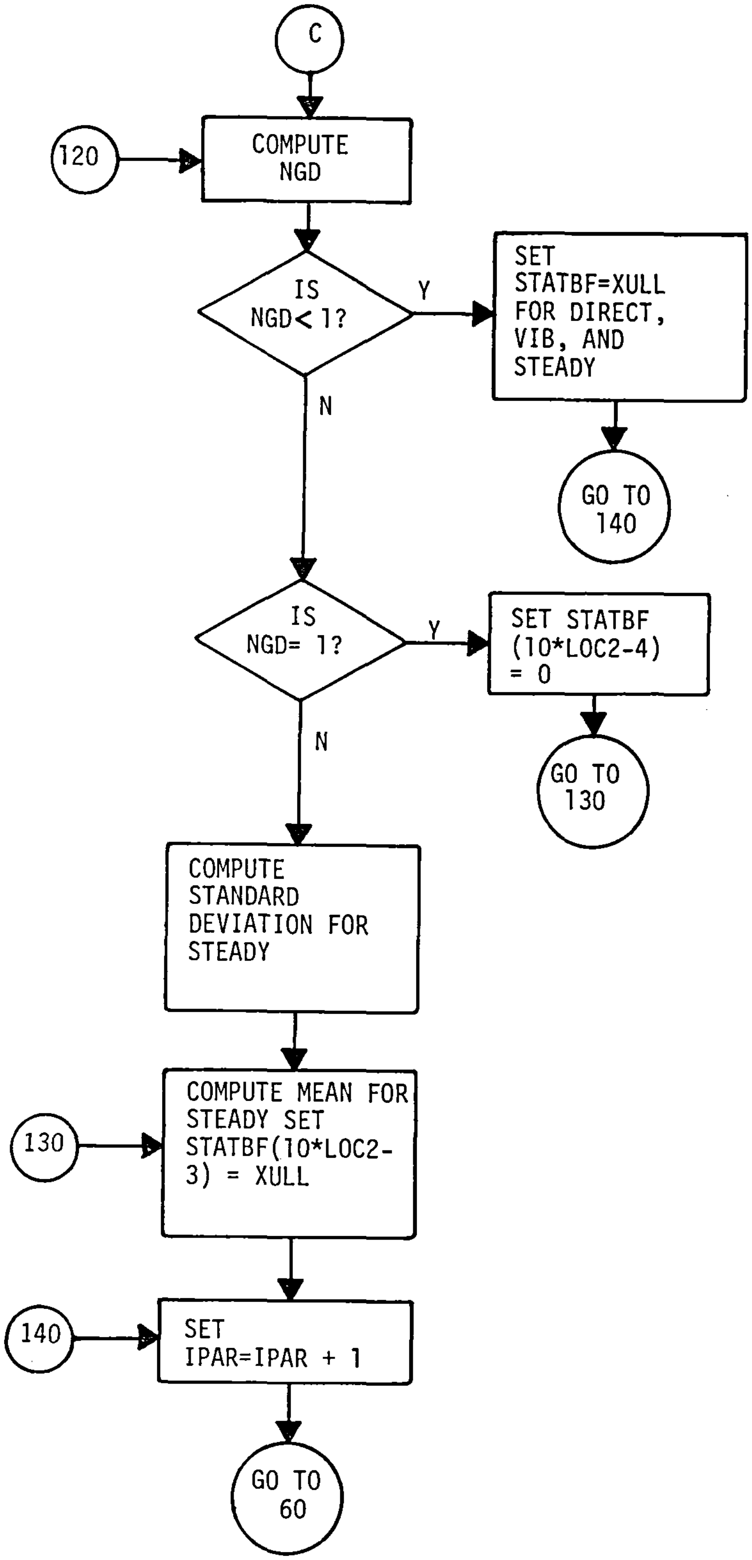

B
IS FRMBF(LOC1) =XULL?
Y
N
SET STATBF(10*LOC1-1) = STATBF (10*LOC1-1) + 1
GO TO 30
GO TO 10
20
SET IVIB=IVIB + 1 LOC2=LOC(IPAR + 1) STATBF(10*LOC2-2) = STATBF(10*LOC2-2) + 1
IS FRMBF(LOC1) =XULL?
Y
N
INCREMENT VIB AND STEADY NO. OF POINTS COUNTERS SET IPAR=IPAR + 1
GO TO 10
COMPUTE VALUES IN ENGR UNITS (RADIANS)
Y
IS PPC(IPAR) ='P'?
NO
COMPUTE SINE AND COSINE OF VALUE
SET VALUE=FRMBF (LOC1) COMPUTE RUNNING SUM&SUM OF SQUARES FOR VIB OR DIRECT
IS PPC(IPAR) = 'V'?
Y
N
GO TO 40
COMPUTE RUNNING SUM&SUM OF SQUARES FOR SINE & COSINE VALUES
IS VALUE> STATBF(10* LOC1-9
Y
N
SET STATBF (10*LOC1-9) = VALUE
IS VALUE>STATBF (10*LOC1-7) ?
Y
N
SET STATBF (10*LOC1-7) = VALUE
GO TO 10
GO TO 10

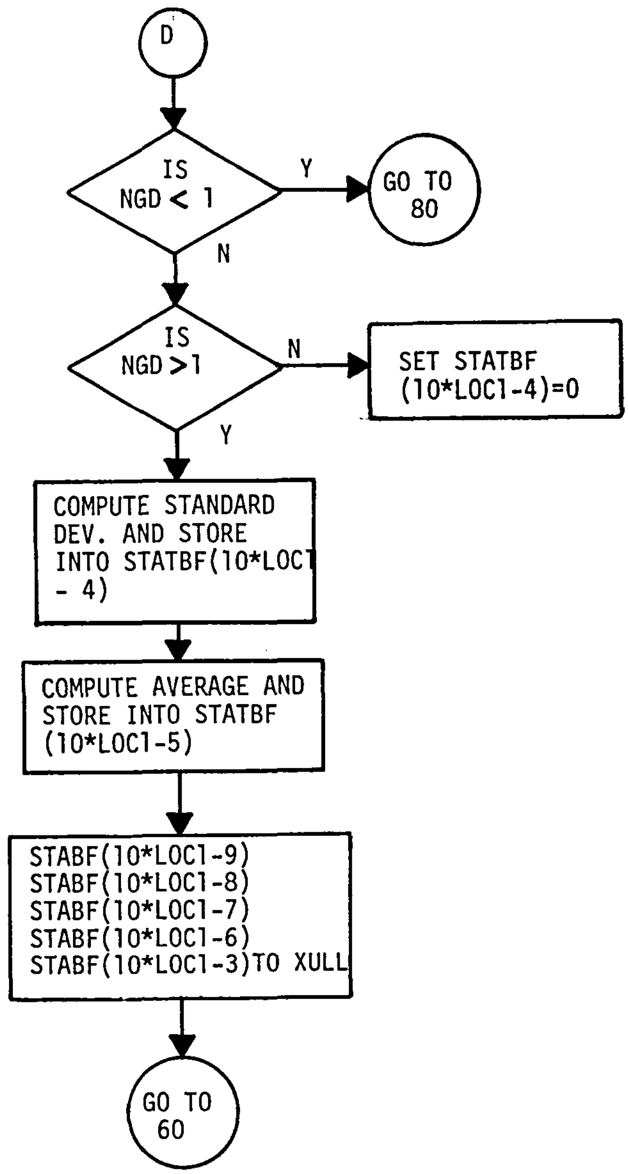

40
SET VALUE2 = FRMBF(LOC2) COMPUTE RUNNING SUM&SUM OF SQUARES FOR STEADY
COMPUTER BIN NO. AND PER-CENTILE COUNT
COMPUTE MAX & MIN VIB&ITS CORRESPONDING STEADYS
COMPUTE MAX & MIN STEADY AND CORRESPONDING VIBS
SET IPAR=IPAR + 1
GO TO 10


50
SET IPAR = 0 IVIB = 0
60
SET IPAR=IPAR + 1
IS IPAR>NPAR?
Y
RETURN
N
SET LOC1=LOC(IPAR)
A


A
IS STATBF (10*LOC1-2) =0?
Y
GO TO 80
N
COMPUTE NGD
IS PPC(IPAR)= 'P'
NO
GO TO D
YES
IS NGD >1?
Y
GO TO 100
N
IS NGD< 1?
Y
GO TO 80
N
SET STATBF (10*LOC1-4) = 0
IS PPC(IPAR)= 'V'?
Y
SET LOC2= LOC(IPAR+1) STATBF(10*LOC2-4) = 0 STATBF (10*LOC1-3) = STATBF(10*LOC1-5)
GO TO 90
SET STATBF(10*LOC1-8)=XULL STATBF(10*LOC1-6)=XULL STATBF(10*LOC1-3)=XULL
GO TO 60

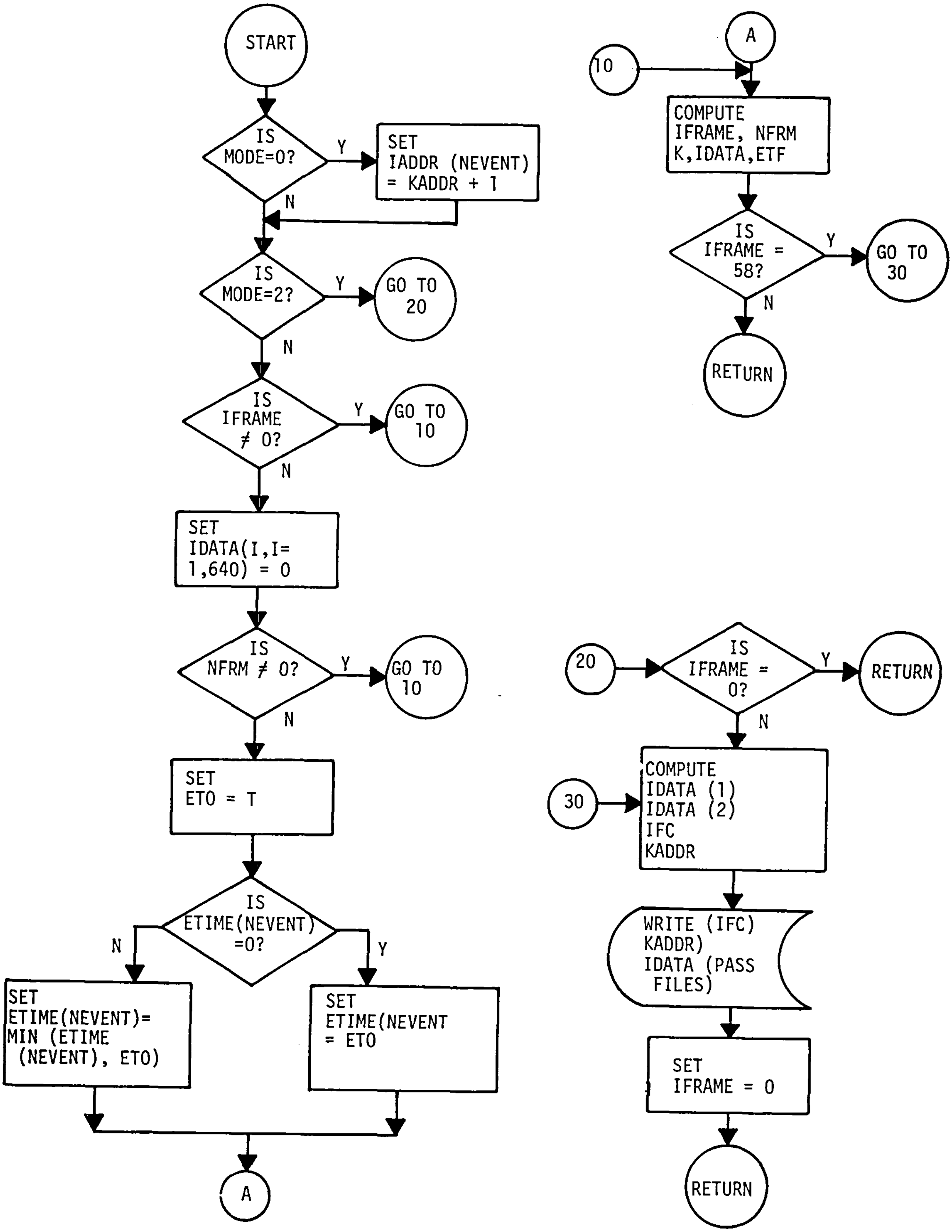

80
SET STATBF=XULL (VIB OR DIRECT)
IS PPC(IPAR) ≠ 'V'?
Y
N
GO TO 60
SET LOC2 = LOC(IPAR + 1) STATBF = XULL (STEADY)
90
SET IVIB=IVIB + 1 IPAR=IPAR + 1
GO TO 60
100
COMPUTE MEAN AND STANDARD DEVIATION OF VIB OR DIRECT
IS PPC(IPAR) ≠ 'V'?
GO TO 70
SET IVIB + 1 LOC2=LOC(IPAR+1) KCUR = 0
B
B
DO 1000 IBIN = 1,128
SET KLAST=KCUR
IS KCUR ≥ 195*NGD?
GO TO 110
1000
CONTINUE
SET STATBF (10*LOC1-3) = XULL
GO TO 120
110
IS KLAST =0?
COMPUTE STATBF(10*LOC1-3)
SET STATBF(10*LOC1-3) =2*IBIN-1
IS STATBF (10*LOC1-9)< STATBF(10* LOC-3) ?
SET STATBF(10*LOC1-3) = STATBF (10*LOC1-9)
IS STATBF (10*LOC1-7) > STATBF(10* LOC-3) ?
SET STATBF(10*LOC1-3) = STATBF (10*LOC1-7)
C

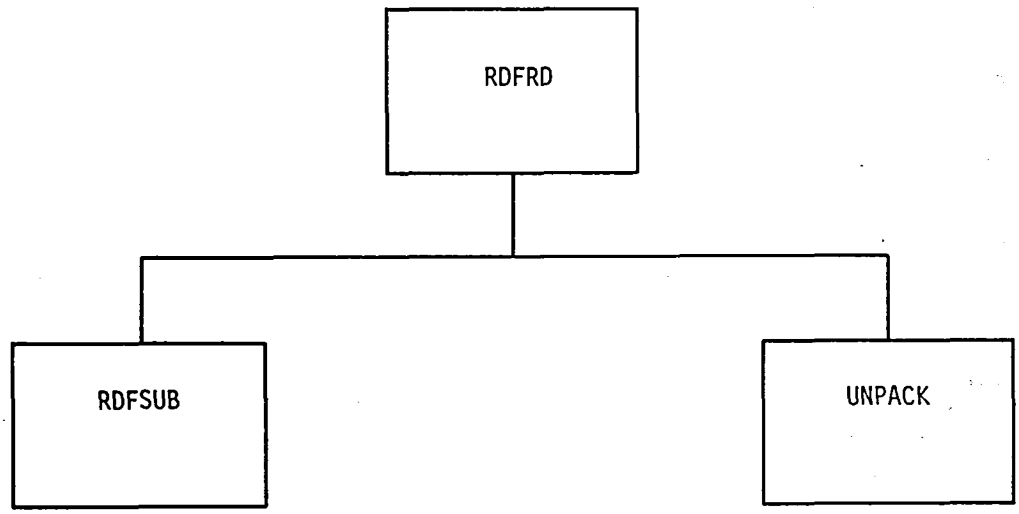

C
120
COMPUTE NGD
IS NGD< 1?
Y
SET STATBF=XULL FOR DIRECT, VIB, AND STEADY
GO TO 140
N
IS NGD= 1?
Y
SET STATBF (10*LOC2-4) = 0
GO TO 130
N
COMPUTE STANDARD DEVIATION FOR STEADY
130
COMPUTE MEAN FOR STEADY SET STATBF(10*LOC2-3) = XULL
140
SET IPAR=IPAR + 1
GO TO 60


D
IS
NGD < 1
Y
GO TO
80
N
IS
NGD >1
N
SET STATBF
(10*LOC1-4)=0
Y
COMPUTE STANDARD
DEV. AND STORE
INTO STATBF(10*LOC1
- 4)
COMPUTE AVERAGE AND
STORE INTO STATBF
(10*LOC1-5)
STABF(10*LOC1-9)
STABF(10*LOC1-8)
STABF(10*LOC1-7)
STABF(10*LOC1-6)
STABF(10*LOC1-3)TO XULL
GO TO
60

```
WRTR    WRTRDF-OUTPUT DATA ON TAPE
**********************SUBROUTINE WRTRDF******************************


*******NASA WALLOPS VERSION OF 08/01/77


*******LANGUAGE - FORTRAN IV


*******MACHINE - HW 625/635


*******PURPOSE -
       THIS ROUTINE OUTPUTS PASS DATA RECORDS ONTO PASS FILES.


*******METHOD -
       TIME, PACKED DATA, EVENT NUMBER, AND NUMBER OF FRAMES FOR EACH
       PASS ARE STORED INTO AN ARRAY AND OUTPUT ONTO A PASS FILE FOR
       EACH PASS.


*******INPUT -

       T          -TIME

       IPCK       -PACKED DATA ARRAY FROM EPACK

       NGVENT     -MANEUVER NUMBER

       NPASS      -PASS NUMBER BEING PROCESSED

       MODE       -CODE TO DESIGNATE DATA STATUS
                  -=0,INITIALIZE BEGINNING OF DATA
                  -=1,DATA BEING ACCUMULATED
                  -=2,FINALIZE DATA


*******OUTPUT -

       IFRAME     -COUNTER OF CURRENT FRAME POSITION ON OUTPUT DATA
                  -FILE RECORD

       KADDR      -CURRENT ADDRESS OF DATA FILE RECORD FOR NPASS

       NFRM       -NUMBER OF FRAMES OF DATA FOR BURST

       ETO        -START TIME OF BURST

       ETF        -STOP TIME OF BURST

       IDATA      -OUTPUT RECORD CONTAINING TIME, PACKED DATA, EVENT
                  -NUMBER, AND NUMBER OF FRAMES FOR PASS
```

```
*******RESTRICTIONS -
        NONE

*******SUBPROGRAMS REQUIRED -
        NONE
```

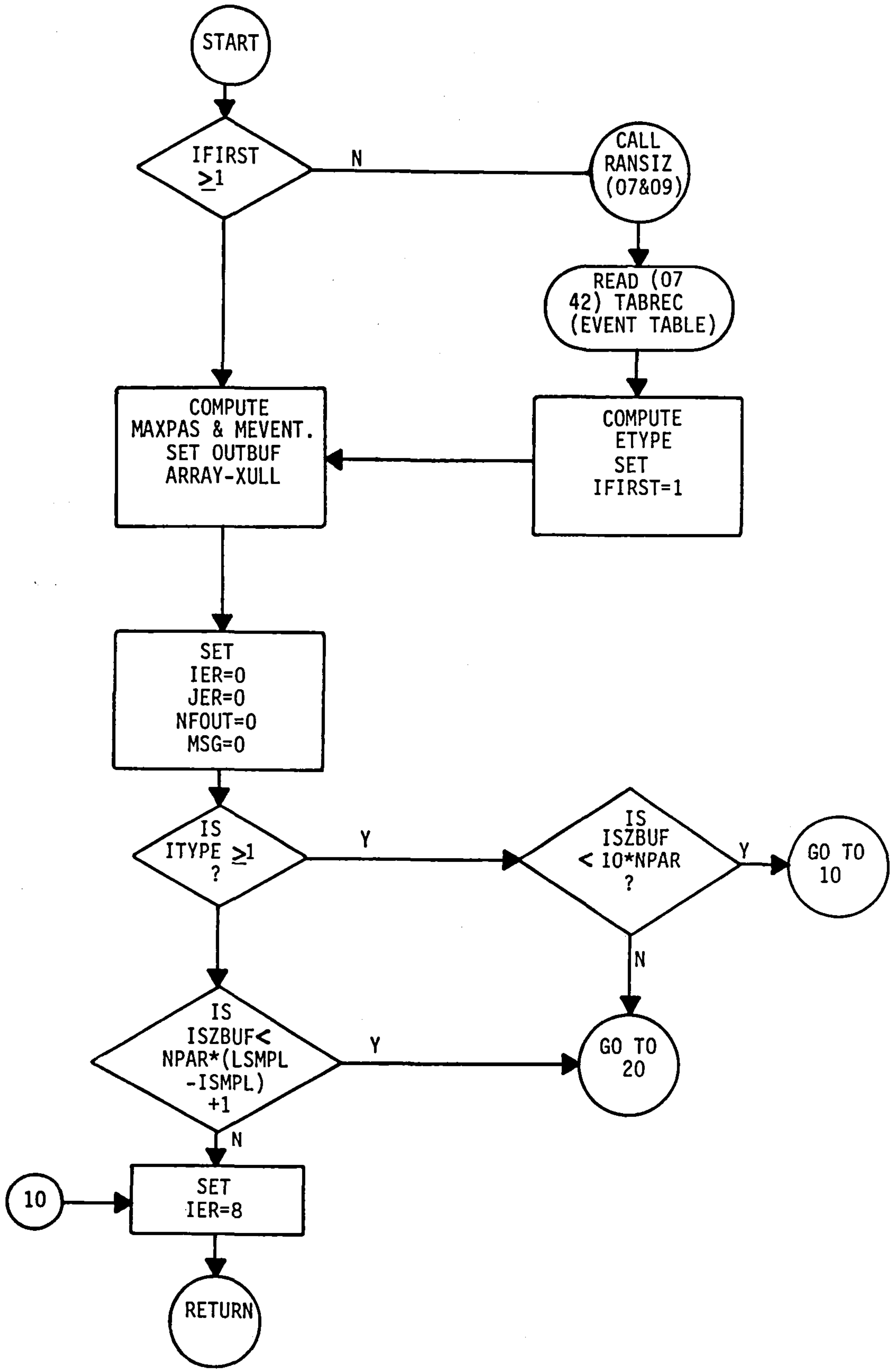

START
IS MODE=0?
Y
N
SET IADDR (NEVENT) = KADDR + 1
IS MODE=2?
GO TO 20
IS IFRAME ≠ 0?
GO TO 10
SET IDATA(I,I= 1,640) = 0
IS NFRM ≠ 0?
GO TO 10
SET ETO = T
IS ETIME(NEVENT) =0?
SET ETIME(NEVENT)= MIN (ETIME (NEVENT), ETO)
SET ETIME(NEVENT = ETO
A
10
A
COMPUTE IFRAME, NFRM K,IDATA,ETF
IS IFRAME = 58?
GO TO 30
RETURN
20
IS IFRAME = 0?
RETURN
30
COMPUTE IDATA (1) IDATA (2) IFC KADDR
WRITE (IFC) KADDR) IDATA (PASS FILES)
SET IFRAME = 0
RETURN

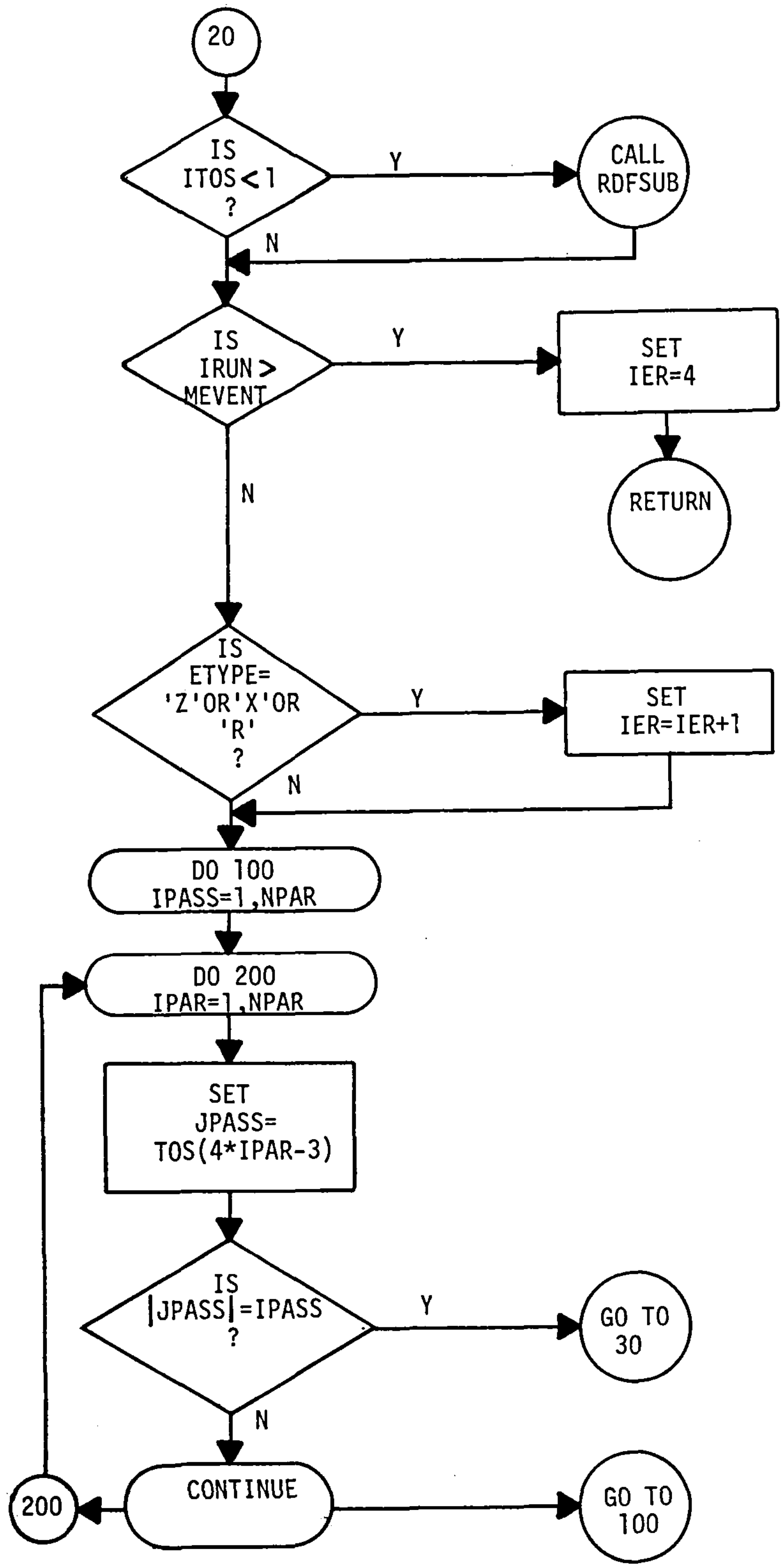


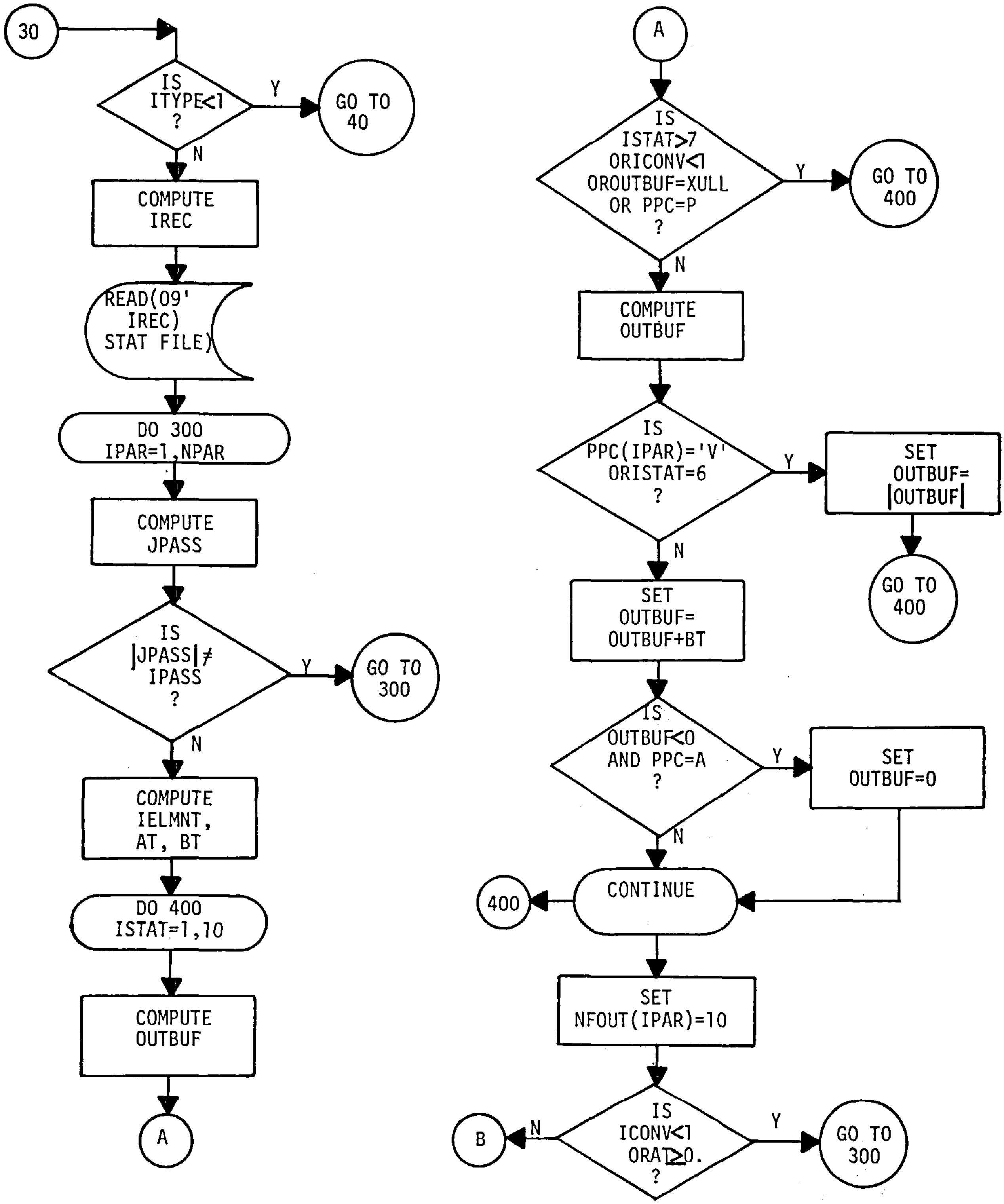
HIERARCHY CHART For RDFRD SUB-PROGRAM

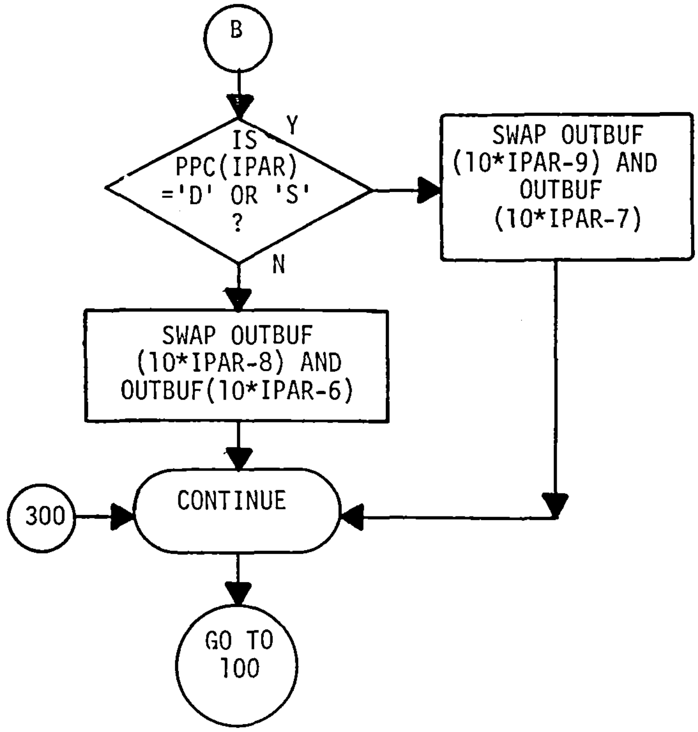
```
**********************SUBROUTINE RDFRD********************************

********NASA WALLOPS VERSION OF 08/01/77

********LANGUAGE - FORTRAN Y

********MACHINE - HW 625/635

********PURPOSE -
        THIS ROUTINE ACCESSES INFORMATION FROM RAW DATA FILES AND
        RAW STATISTICS FILES AS PRODUCED THROUGH SCAN PROGRAM.

********METHOD -
        THIS ROUTINE WILL EXTRACT DATA FOR A GIVEN SET OF PARAMETERS
        ACCORDING TO INPUT ARGUMENTS TO THIS ROUTINE AND RETURN THEM
        TO THE CALLING ROUTINE.

********INPUT -

        ITOS      -=0,WHEN INITIAL CALL WITH GIVEN SET OF PARAMETER
                      SPECS WHICH MUST BE USED TO SEARCH SENSOR TABLE
                  -=1,WHEN PREVIOUS CALL WITH SAME SET OF PARAMETER
                      SPECS HAS BEEN USED TO DEFINE TABLE OF
                      SUBSCRITS ARGUMENT (TOS) THUS AVERTING SEARCH
                      OF SENSOR TABLE

        ITYPE     -=0,DATA SAMPLES ARE BEING REQUESTED
                  -=1,STATISTICS ARE BEING REQUESTED
                  -=1,STATISTICS ARE BEING REQUESTED

        ICONV     -=0,RESULTS ARE TO BE IN RAW UNITS
                  -=1,RESULTS ARE TO BE IN ENGINEERING UNITS

        IRUN      -RUN (EVENT) NUMBER REQUESTED

        NPAR      -NUMBER OF PARAMETERS REQUESTED
        NPAR      -NUMBER OF PARAMETERS REQUESTED

        NAMES     -PARAMETER MNEMONICS

        PPC       -PARAMETER PRE-PROCESSING CODES

        SR        -PARAMETER SAMPLE RATE

        ISMPL     -BEGINNING SAMPLE NUMBER REQUESTED, IF DATA

        LSMPL     -ENDING SAMPLE NUMBER REQUESTED, IF DATA

        ISZBUF    -SIZE OF OUTBUF
```

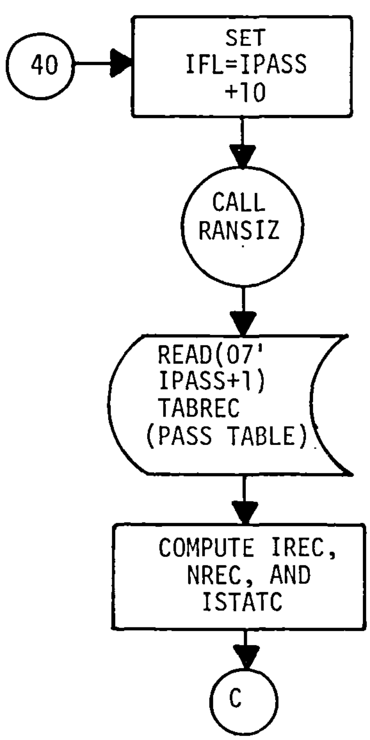
```
        TOS        -TABLE OF SUBSCRIPTS, 4 WORDS PER PARAMETER, BUILT
                   -WITHIN ROUTINE WHEN ITOS = 0 FOR POSSIBLE
                   -SUBSEQUENT CALLS WITH SAME SET OF PARAMETER SPECS
        TOS(1)     -PASS NUMBER
        TOS(2)     -ELEMENT NUMBER
        TOS(3)     -A TERM
        TOS(4)     -B TERM

********OUTPUT -

        OUTBUF     -OUTPUT BUFFER FOR DATA OR STATISTICS

        ISZBUF     -SIZE OF OUTBUF

        NFOUT      -OUTPUT ARRAY OF ACTUAL NUMBER OF FRAMES RETURNED
                   -FOR EACH PARAMETER

        MSG        -ARRAY OF OUTPUT PARAMETER MESSAGES WHEN ASKING
                   -FOR DATA
                   -=0,NO ERRORS
                   -=1,LESS DATA RETURNED THAN REQUESTED BECAUSE
                   -   ENDING SAMPLE REQUESTED IS GREATER THAN LAST
                   -   SAMPLE AVAILABLE
                   -=2,BEGINNING SAMPLE REQUESTED IS GREATER THAN LAST
                   -   SAMPLE AVAILABLE

        IER        -OUTPUT ERROR CODE DETERMINED AS THE SUM OF THE
                   -FOLLOWING INDIVIDUAL CODES
                   -=0,NO ERRORS
                   -=1,CAL RUN REQUESTED
                   -=2,PARAMETER LEVEL ERRORS EXIST
                   -=4,NO SUCH RUN
                   -=8,BUFFER SIZE ERROR

********RESTRICTIONS -
        NONE

********SUBPROGRAMS REQUIRED -
        RDFSUB
        UNPACK
```

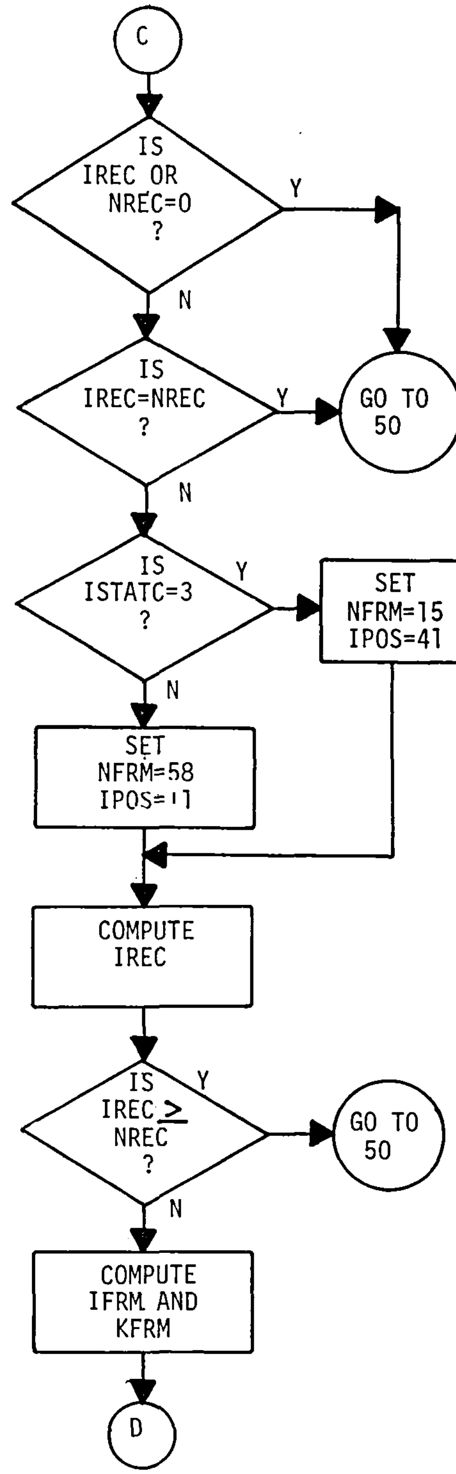

START
IFIRST ≥1
N
CALL RANSIZ (07&09)
READ (07 42) TABREC (EVENT TABLE)
COMPUTE ETYPE SET IFIRST=1
COMPUTE MAXPAS & MEVENT. SET OUTBUF ARRAY-XULL
SET IER=0 JER=0 NFOUT=0 MSG=0
IS ITYPE ≥1 ?
Y
IS ISZBUF < 10*NPAR ?
Y
GO TO 10
N
GO TO 20
IS ISZBUF< NPAR*(LSMPL -ISMPL) +1
Y
N
10
SET IER=8
RETURN

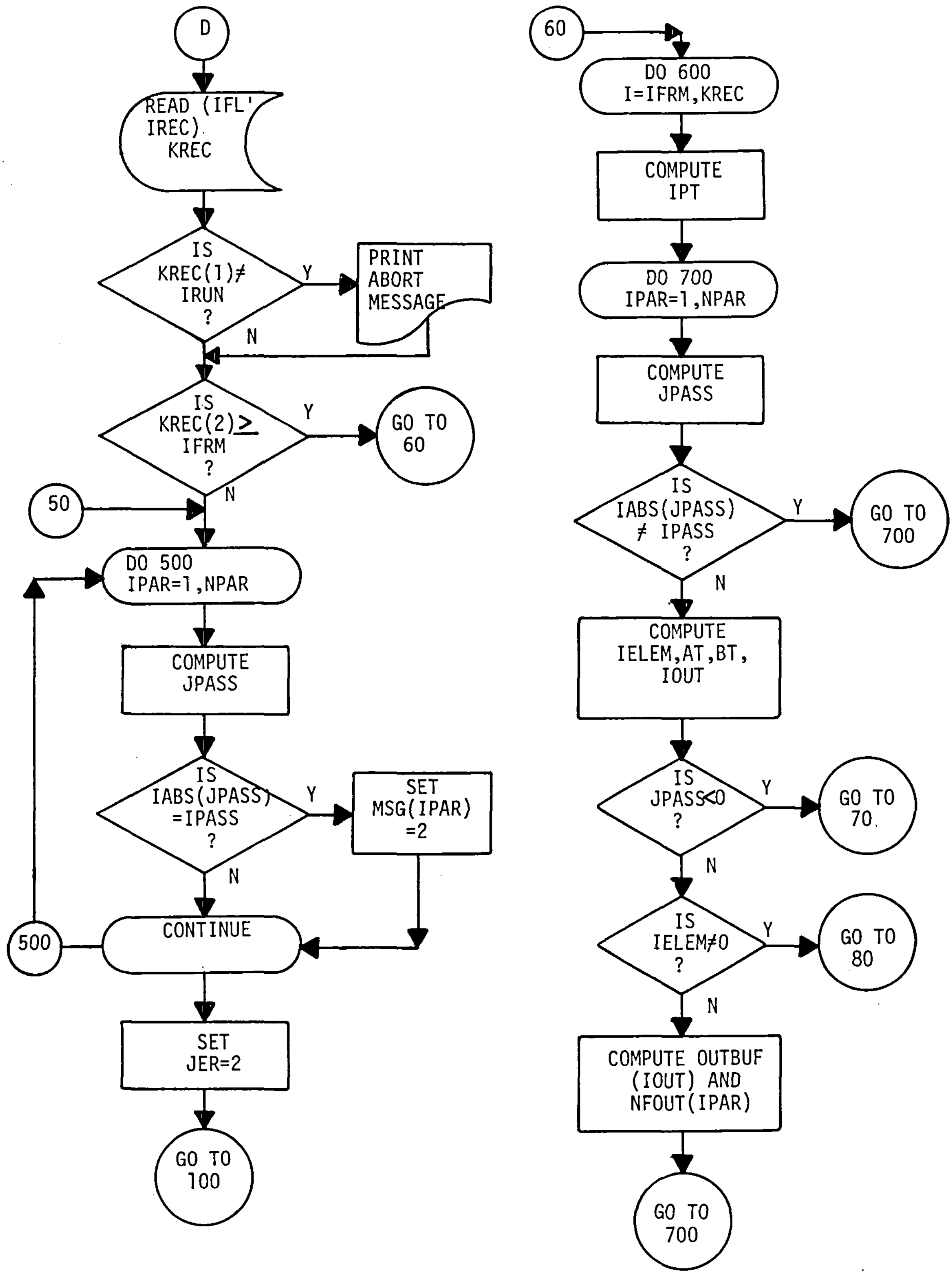

20
IS ITOS<1 ?
Y
CALL RDFSUB
N
IS IRUN > MEVENT
Y
SET IER=4
RETURN
N
IS ETYPE= 'Z'OR'X'OR 'R' ?
Y
SET IER=IER+1
N
DO 100 IPASS=1,NPAR
DO 200 IPAR=1,NPAR
SET JPASS= TOS(4*IPAR-3)
IS |JPASS|=IPASS ?
Y
GO TO 30
N
200
CONTINUE
GO TO 100

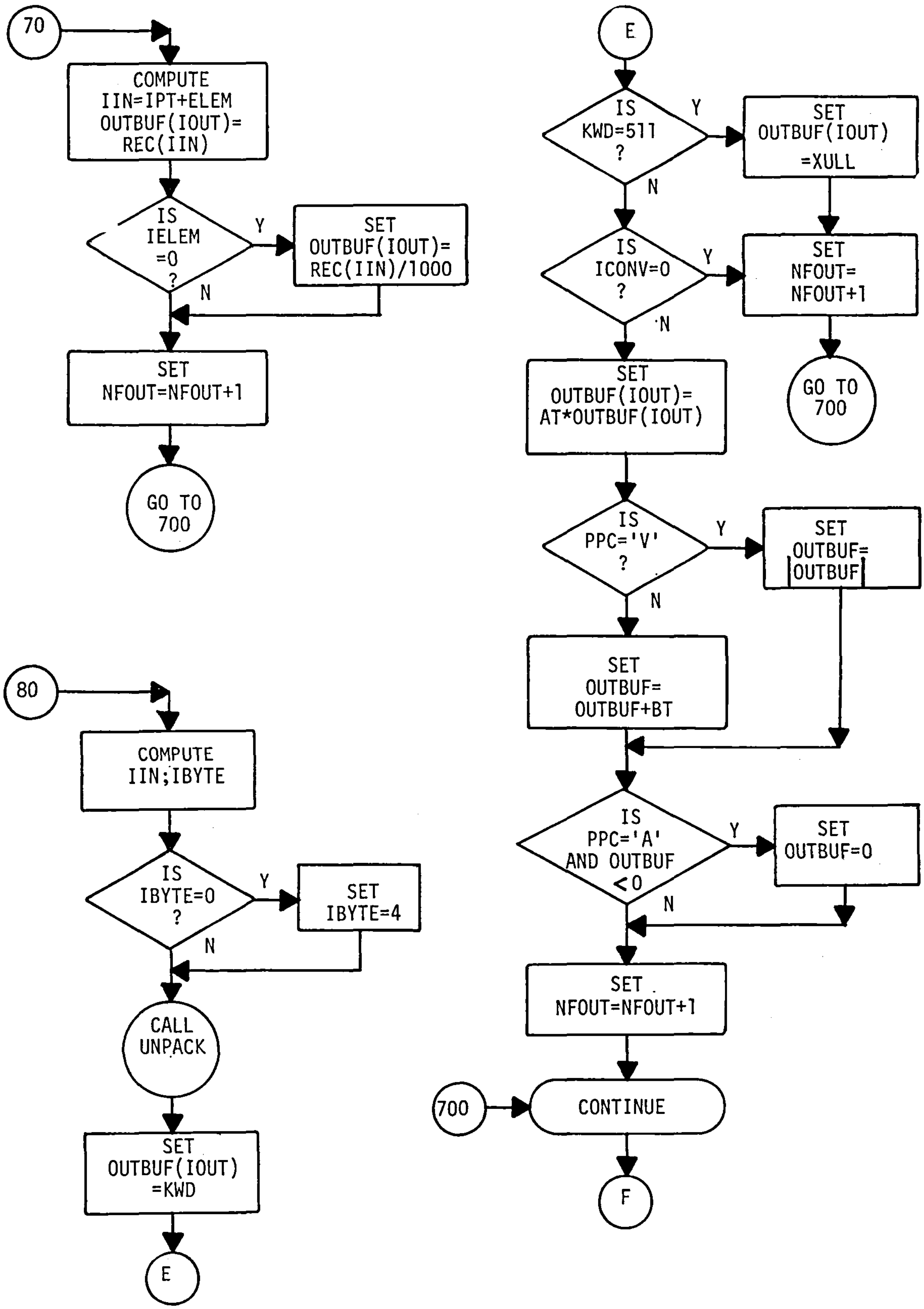

30
IS ITYPE<1 ?
Y
GO TO 40
N
COMPUTE IREC
READ(09' IREC) STAT FILE)
DO 300 IPAR=1,NPAR
COMPUTE JPASS
IS |JPASS|≠ IPASS ?
Y
GO TO 300
N
COMPUTE IELMNT, AT, BT
DO 400 ISTAT=1,10
COMPUTE OUTBUF
A
A
IS ISTAT>7 ORICONV<1 OROUTBUF=XULL OR PPC=P ?
Y
GO TO 400
N
COMPUTE OUTBUF
IS PPC(IPAR)='V' ORISTAT=6 ?
Y
SET OUTBUF= |OUTBUF|
GO TO 400
N
SET OUTBUF= OUTBUF+BT
IS OUTBUF<0 AND PPC=A ?
Y
SET OUTBUF=0
N
400
CONTINUE
SET NFOUT(IPAR)=10
B
N
IS ICONV<1 ORAT≥0. ?
Y
GO TO 300


B
IS PPC(IPAR) ='D' OR 'S' ?
Y
N
SWAP OUTBUF (10*IPAR-9) AND OUTBUF (10*IPAR-7)
SWAP OUTBUF (10*IPAR-8) AND OUTBUF(10*IPAR-6)
300
CONTINUE
GO TO 100

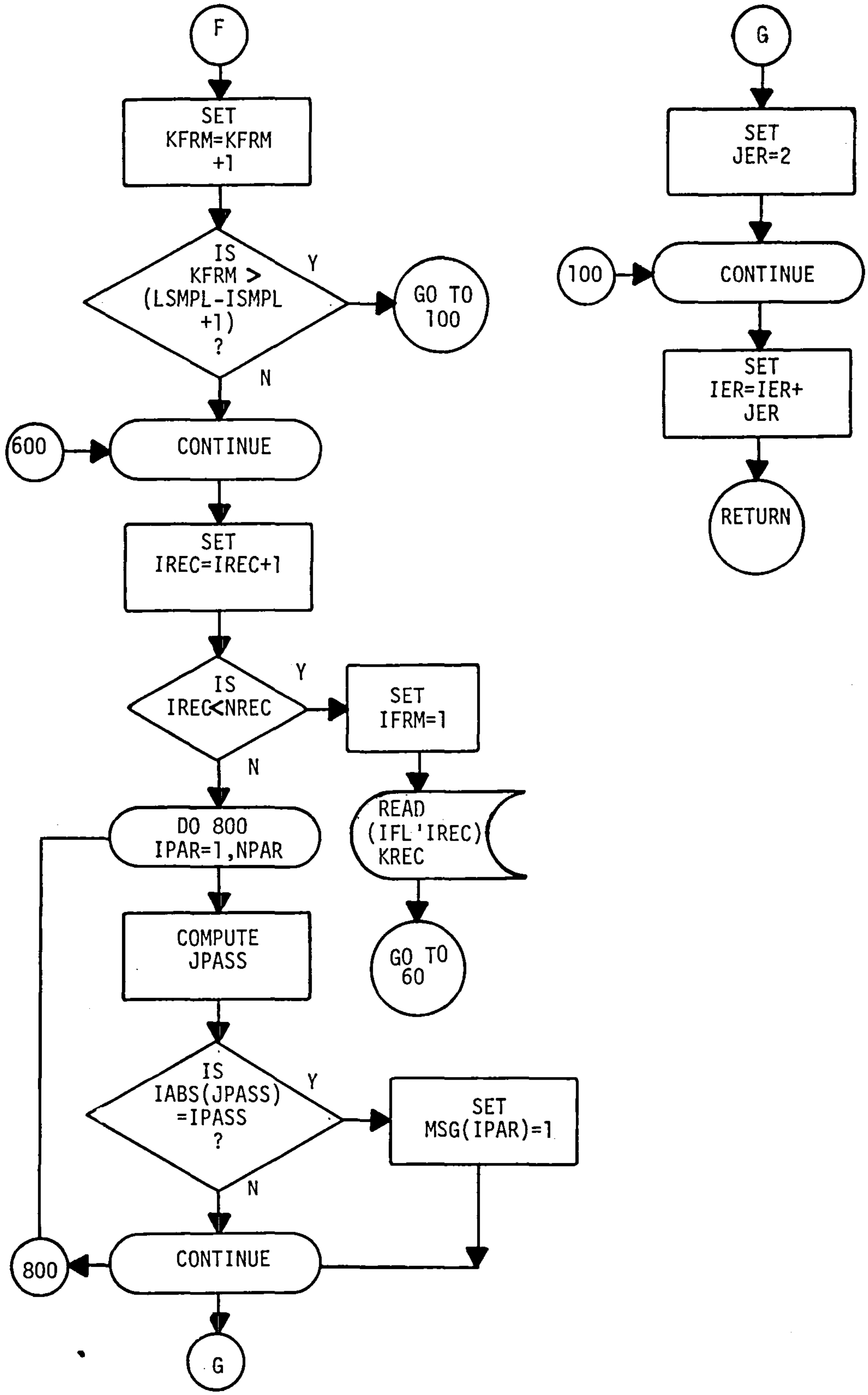

40
SET IFL=IPASS +10
CALL RANSIZ
READ(07' IPASS+1) TABREC (PASS TABLE)
COMPUTE IREC, NREC, AND ISTATC
C


C
IS IREC OR NREC=0 ?
Y
N
IS IREC=NREC ?
Y
GO TO 50
N
IS ISTATC=3 ?
Y
SET NFRM=15 IPOS=41
N
SET NFRM=58 IPOS=11
COMPUTE IREC
IS IREC ≥ NREC ?
Y
GO TO 50
N
COMPUTE IFRM AND KFRM
D

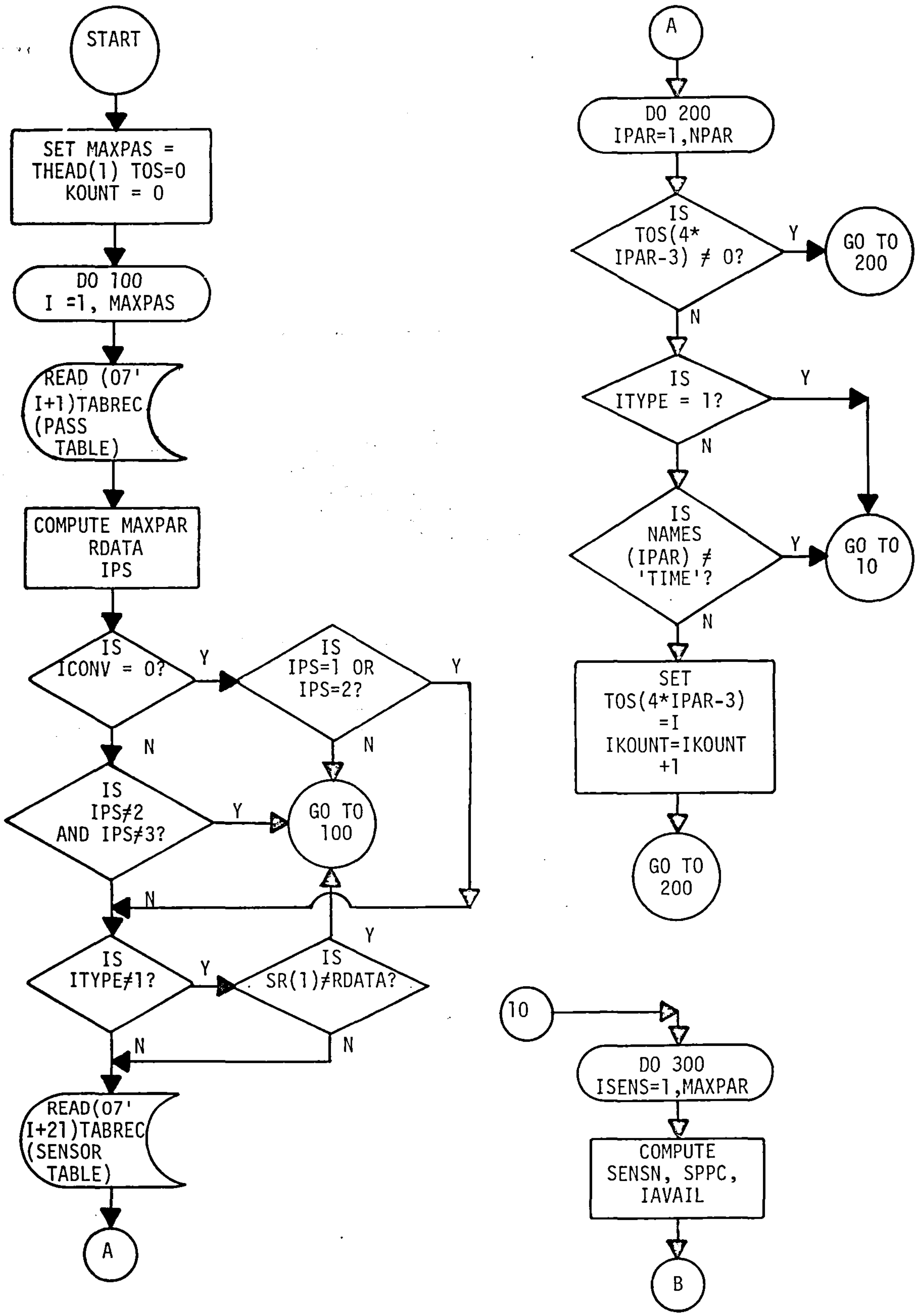

D
READ (IFL' IREC) KREC
IS KREC(1)≠ IRUN ?
Y
PRINT ABORT MESSAGE
N
IS KREC(2)≥ IFRM ?
Y
GO TO 60
50
N
DO 500 IPAR=1,NPAR
COMPUTE JPASS
IS IABS(JPASS) =IPASS ?
Y
SET MSG(IPAR) =2
N
500
CONTINUE
SET JER=2
GO TO 100
60
DO 600 I=IFRM,KREC
COMPUTE IPT
DO 700 IPAR=1,NPAR
COMPUTE JPASS
IS IABS(JPASS) ≠ IPASS ?
Y
GO TO 700
N
COMPUTE IELEM,AT,BT, IOUT
IS JPASS<0 ?
Y
GO TO 70
N
IS IELEM≠0 ?
Y
GO TO 80
N
COMPUTE OUTBUF (IOUT) AND NFOUT(IPAR)
GO TO 700

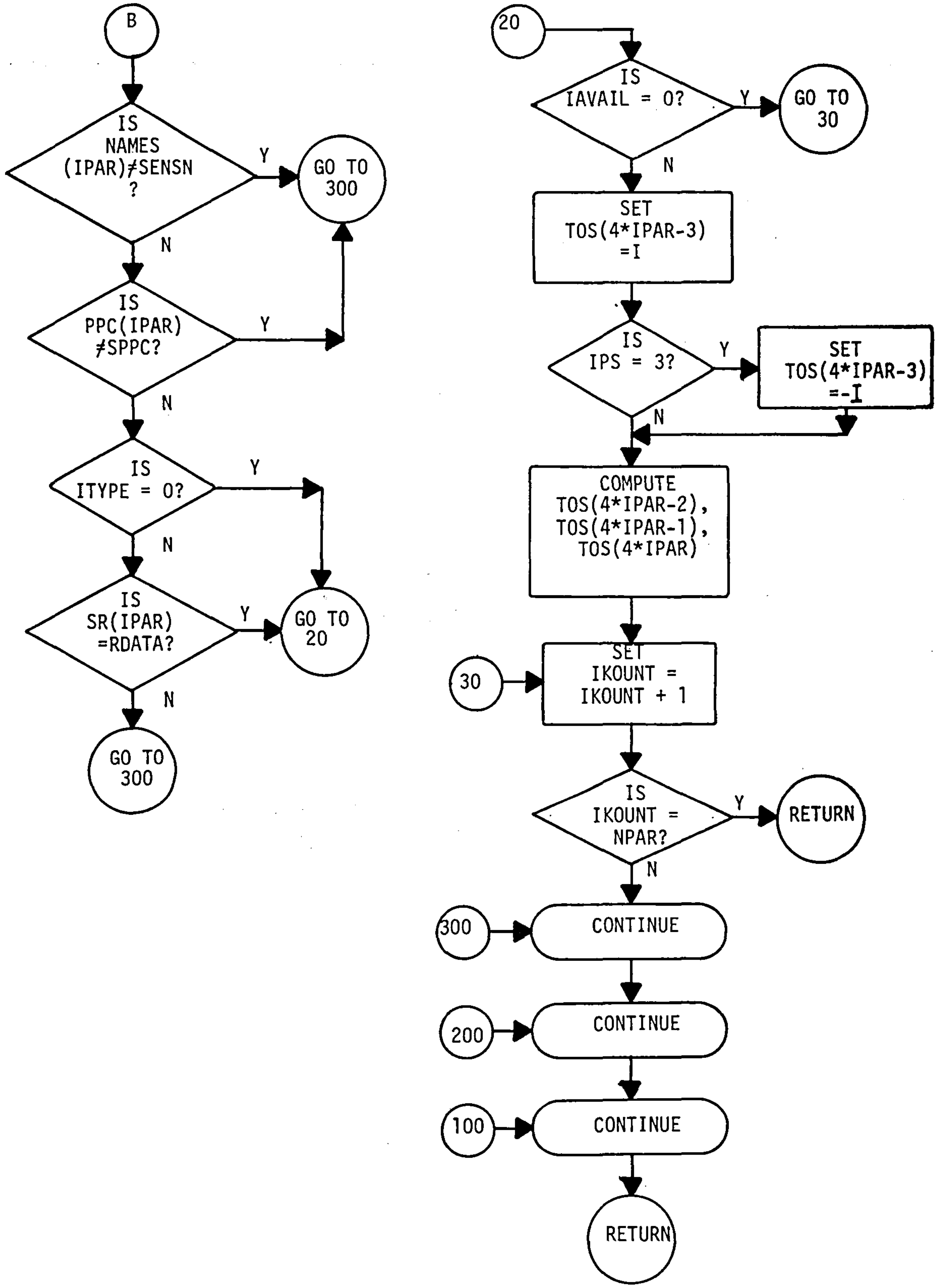


70
COMPUTE IIN=IPT+ELEM OUTBUF(IOUT)= REC(IIN)
IS IELEM =0 ?
Y
N
SET OUTBUF(IOUT)= REC(IIN)/1000
SET NFOUT=NFOUT+1
GO TO 700
80
COMPUTE IIN;IBYTE
IS IBYTE=0 ?
Y
N
SET IBYTE=4
CALL UNPACK
SET OUTBUF(IOUT) =KWD
E
E
IS KWD=511 ?
Y
N
SET OUTBUF(IOUT) =XULL
IS ICONV=0 ?
Y
N
SET NFOUT= NFOUT+1
GO TO 700
SET OUTBUF(IOUT)= AT*OUTBUF(IOUT)
IS PPC='V' ?
Y
N
SET OUTBUF= |OUTBUF|
SET OUTBUF= OUTBUF+BT
IS PPC='A' AND OUTBUF <0
Y
N
SET OUTBUF=0
SET NFOUT=NFOUT+1
700
CONTINUE
F


F
SET KFRM=KFRM +1
IS KFRM > (LSMPL-ISMPL +1) ?
Y
N
GO TO 100
600
CONTINUE
SET IREC=IREC+1
IS IREC<NREC
Y
N
SET IFRM=1
READ (IFL'IREC) KREC
GO TO 60
DO 800 IPAR=1,NPAR
COMPUTE JPASS
IS IABS(JPASS) =IPASS ?
Y
N
SET MSG(IPAR)=1
800
CONTINUE
G
G
SET JER=2
100
CONTINUE
SET IER=IER+ JER
RETURN

```
RDFS*******************SUBROUTINE RDFSUB********************************

********NASA WALLOPS VERSION OF 08/01/77

********LANGUAGE - FORTRAN Y

********MACHINE - HW 625/635

********PURPOSE -
        BUILD A TABLE OF SUBSCRIPTS TO BE USED IN RDFRD.

********METHOD -
        GIVEN A SET OF NAMES, PPC, AND SR THE PASS AND SENSOR TABLES
        ARE READ TO FIND WHICH PASS AND PARAMETER NUMBERS CORRESPOND.
        THESE ARE OUTPUT TO THE SUBSCRIPT ARRAY (TOS) IN CONJUNCTION
        WITH THE ENGINEERING UNIT CONVERSION FACTORS.  IF A SEARCH
        OF THE PASS AND SENSOR TABLES DOES NOT FIND A MATCH, TOS
        VALUES ARE RETURNED AS ZEROES.

********INPUT -

        ITYPE     -=0,DATA SAMPLES ARE REQUESTED
                  -=1,STATISTICS ARE REQUESTED

        ICONV     -=0,RESULTS ARE TO BE IN RAW UNITS
                  -=1,RESULTS ARE TO BE IN ENGINEERING UNITS

        NPAR      -NUMBER OF PARAMETERS REQUESTED

        NAMES     -PARAMETER MNEMONIC

        PPC       -PARAMETER PRE-PROCESSOR CODE

        SR        -PARAMETER SAMPLE RATE

********OUTPUT -

        TOS       -(SEE RDFRD INPUT)

********RESTRICTIONS -
        NONE

********SUBPROGRAMS REQUIRED -
        NONE
```


START
SET MAXPAS = THEAD(1) TOS=0 KOUNT = 0
DO 100 I =1, MAXPAS
READ (07' I+1)TABREC (PASS TABLE)
COMPUTE MAXPAR RDATA IPS
IS ICONV = 0?
IS IPS=1 OR IPS=2?
IS IPS≠2 AND IPS≠3?
GO TO 100
IS ITYPE≠1?
IS SR(1)≠RDATA?
READ(07' I+21)TABREC (SENSOR TABLE)
A
DO 200 IPAR=1,NPAR
IS TOS(4* IPAR-3) ≠ 0?
GO TO 200
IS ITYPE = 1?
IS NAMES (IPAR) ≠ 'TIME'?
GO TO 10
SET TOS(4*IPAR-3) =I IKOUNT=IKOUNT +1
GO TO 200
10
DO 300 ISENS=1,MAXPAR
COMPUTE SENSN, SPPC, IAVAIL
B
Y
N


B
IS NAMES (IPAR)≠SENSN ?
Y
GO TO 300
N
IS PPC(IPAR) ≠SPPC?
Y
N
IS ITYPE = 0?
Y
N
IS SR(IPAR) =RDATA?
Y
GO TO 20
N
GO TO 300
20
IS IAVAIL = 0?
Y
GO TO 30
N
SET TOS(4*IPAR-3) =I
IS IPS = 3?
Y
SET TOS(4*IPAR-3) =-I
N
COMPUTE TOS(4*IPAR-2), TOS(4*IPAR-1), TOS(4*IPAR)
30
SET IKOUNT = IKOUNT + 1
IS IKOUNT = NPAR?
Y
RETURN
N
300
CONTINUE
200
CONTINUE
100
CONTINUE
RETURN

*************************** SUBROUTINE UNPACK ***************************

PROGRAM IDENTIFICATION

```
PROGRAM NAME - UNPACK
PROGRAM NO.  - 1.1.2320
AUTHOR       - DAVID L. DAVIS

COMPUTER     - HW 625/635
MEMORY       - 70 WORDS
PERIPHERALS  - NONE
LANGUAGE     - GMAP
```

PURPOSE

TO UNPACK 40 DATA VALUES FROM
A 10 WORD ARRAY (4 DATA VALUES
PER WORD) IN A 40 WORD ARRAY
(ONE DATA VALUE PER WORD).

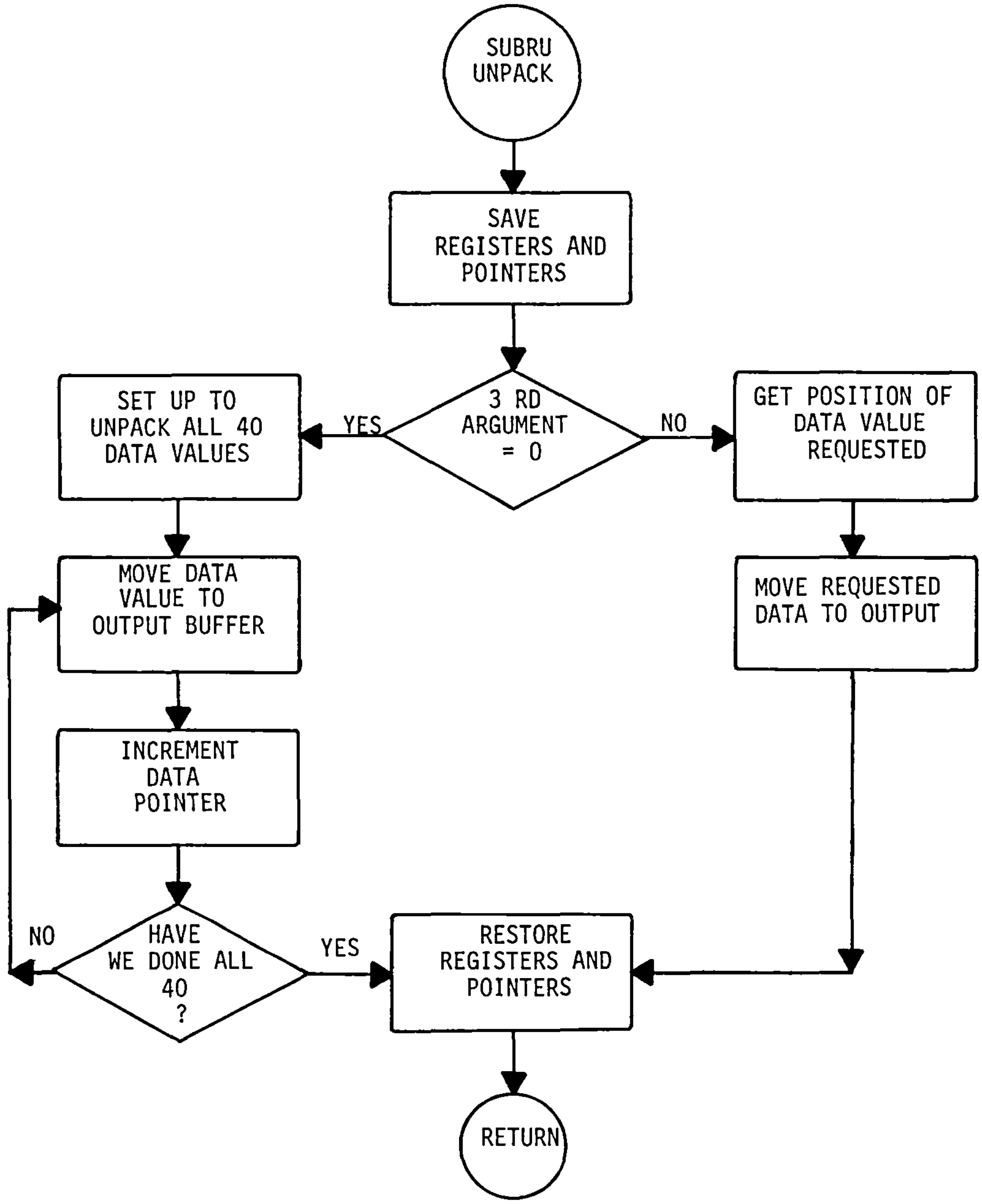
METHOD

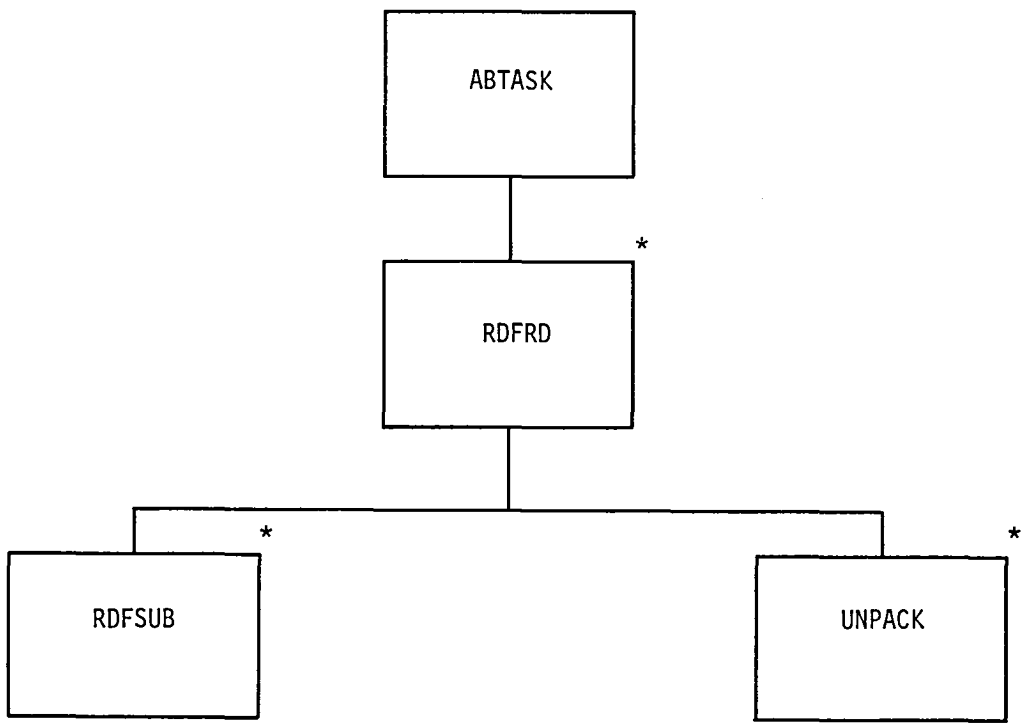
THE THIRD ARGUMENT IS EXAMINED
TO SEE IF ALL 40 DATA VALUES ARE
REQUESTED OR IF ONLY ONE IS REQUESTED.
THE SPECIFIED AMOUNT OF DATA IS THEN
MOVED TO THE OUTPUT BUFFER.

INPUT/OUTPUT

CALLING SEQUENCE<

```
 CALL UNPACK(A,B,C) WHERE

   A = FIRST WORD ADDRESS OF 10 WORD INPUT BUFFER
   B = FIRST WORD ADDRESS OF 40 WORD OUTPUT
       BUFFER IF C IS NON-ZERO, OR ONE WORD
       BUFFER IF C IS ZERO.
   C = CHARACTER POSITION INDICATOR  (1 - 4)
```


SUBRU UNPACK
SAVE REGISTERS AND POINTERS
3 RD ARGUMENT = 0
YES
NO
SET UP TO UNPACK ALL 40 DATA VALUES
GET POSITION OF DATA VALUE REQUESTED
MOVE DATA VALUE TO OUTPUT BUFFER
MOVE REQUESTED DATA TO OUTPUT
INCREMENT DATA POINTER
HAVE WE DONE ALL 40 ?
NO
YES
RESTORE REGISTERS AND POINTERS
RETURN



*See Sub-program RDFRD for Comments and Flowcharts

HIERARCHY CHART for AB PROGRAM

```
ABTASK
*************************AB TASK****************************************

********NASA WALLOPS VERSION OF AUG 19, 1977

********LANGUAGE - FORTRAN Y

********MACHINE - HONEYWELL 600 SERIES

********PURPOSE
        TO COMPUTE LINEAR COEFFICIENTS (SLOPE AND INTERCEPT)
        FOR CONVERSION OF RAW UNITS TO ENGINEERING UNITS

********METHOD

        A. HEARDER AND CALRUN TABLES ARE READ

        B. PASS AND SENSOR TABLES ARE READ FOR EACH PASS

        C. STATUS IS CHECKED TO DETERMINE IF AB IS TO BE RUN

        D. STATISTICS ARE OBTAINED FROM FILE 09 THROUGH RDFRD

        E. CALTYP IS CHECKED AND A AND B ARE CALCULATED AS FOLLOWS

           1. IF CALTYP = 1
              A = EUCV1
              B = EUCV2

           2. IF CALTYP = 2
              A = EUCV1/(EUCV2/100)*256
              B = -A*256
              UNLESS EUCV2 = 0
              A = 1
              B = 0

           1. IF CALTYP = P
              A = (EUCV1-EUCV2)/(RBAR-ZBAR)
              B = EUCV1 - A*ZBAR

           4. IF CALTYP = Z
              A = EUCV1/(RBAR-ZBAR)
              B = EUCV2 - A*ZBAR

           5. IF CALTYP = X
              A = EUCV1/(RBAR-ZBAR)
              B = (PRE*XEBAR + POST*XTBAR)/(PRE + POST)
              UNLESS POST AND PRE = 0
              B = EUCV2 - A*XBAR

        F. PERCENT FULL SCALE AND SHIFT ARE CALCULATED

        G. DATA AVAILABILITY STATUS IS CHANGED TO 2

        H. PASS AND SENSOR TABLES ARE UPDATED
```

********INPUT - FROM FILE 07

HTAB(27)	HEADER TABLE - CONTAINS DATA IDENTIFYING FLIGHT
MAXPAS	MAXIMUM NUMBER OF PASSES - FROM HTAB
CTAB(120)	CALIBRATION RUN TABLE - CONTAINS RUN NUMBERS FOR SPECIFIED MANEUVER
ICTAB(120)	INTEGER REPRESENTATION OF CTAB
PTAB(600)	PASS TABLE - CONTAINS DATA CONCERNING EACH PASS
CPTAB(600)	CHARACTER REPRESENTATION OF PTAB
MAXPAR	NUMBER OF PARAMETERS IN PASS - FROM PTAB
ISTATC	DATA AVAILABILITY STATUS CODE - FROM PTAB 0 = DATA NOT AVAILABLE 1 = DATA AVAILABLE 2 = ABTASK COMPLETED 3 = DPTASK COMPLETED
STAB(600)	SENSOR TABLE - CONTAINS SENSOR INFORMATION
CSTAB(600)	CHARACTER REPRESENTATION OF STAB
CALADR(40)	CALABRATION ADDRESS FOR EACH PARAMETER
PPC(40)	PRE-PROCESSING CODE FOR EACH PARAMETER
CALTYP(40)	CALIBRATION TYPE FOR EACH PARAMETER
EUCV1(40)	FACTORY CALIBRATED A TERM FOR EACH PARAMETER
EUCV2(40)	FACTORY CALIBRATED B TERM FOR EACH PARAMETER

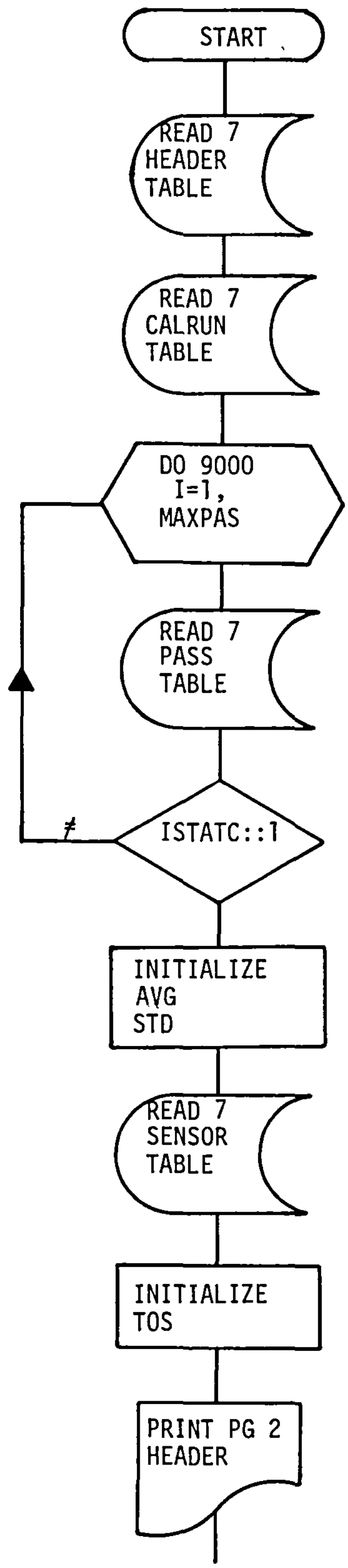
********INPUT FROM FILE 09 THROUGH SUBROUTINE RDFRD

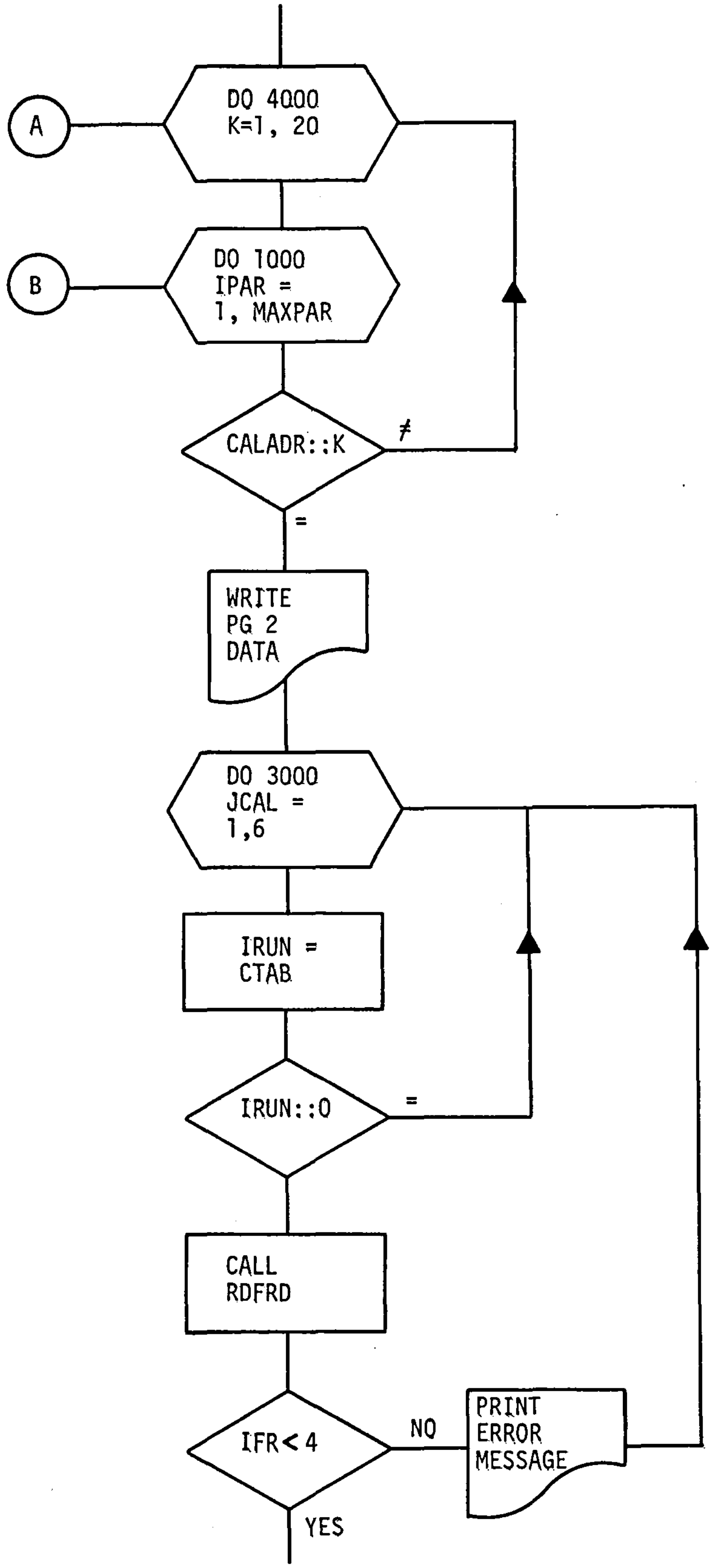
STATISTICS - AVERAGES AND STANDARD DEVIATIONS
OF RAW DATA

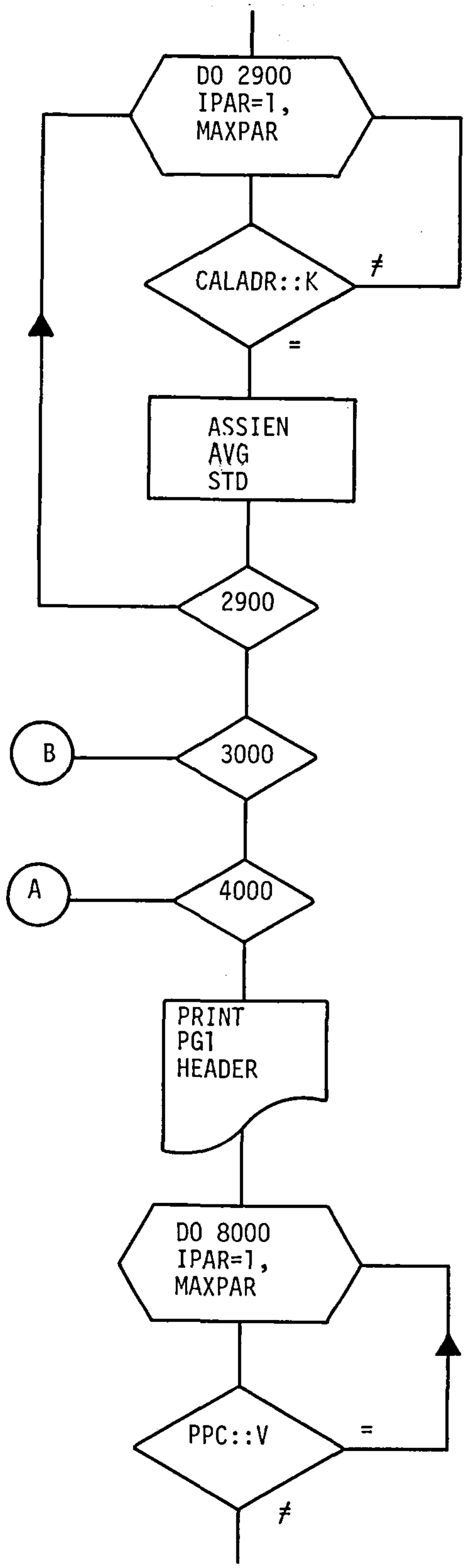
********OUTPUT - TO FILE 07

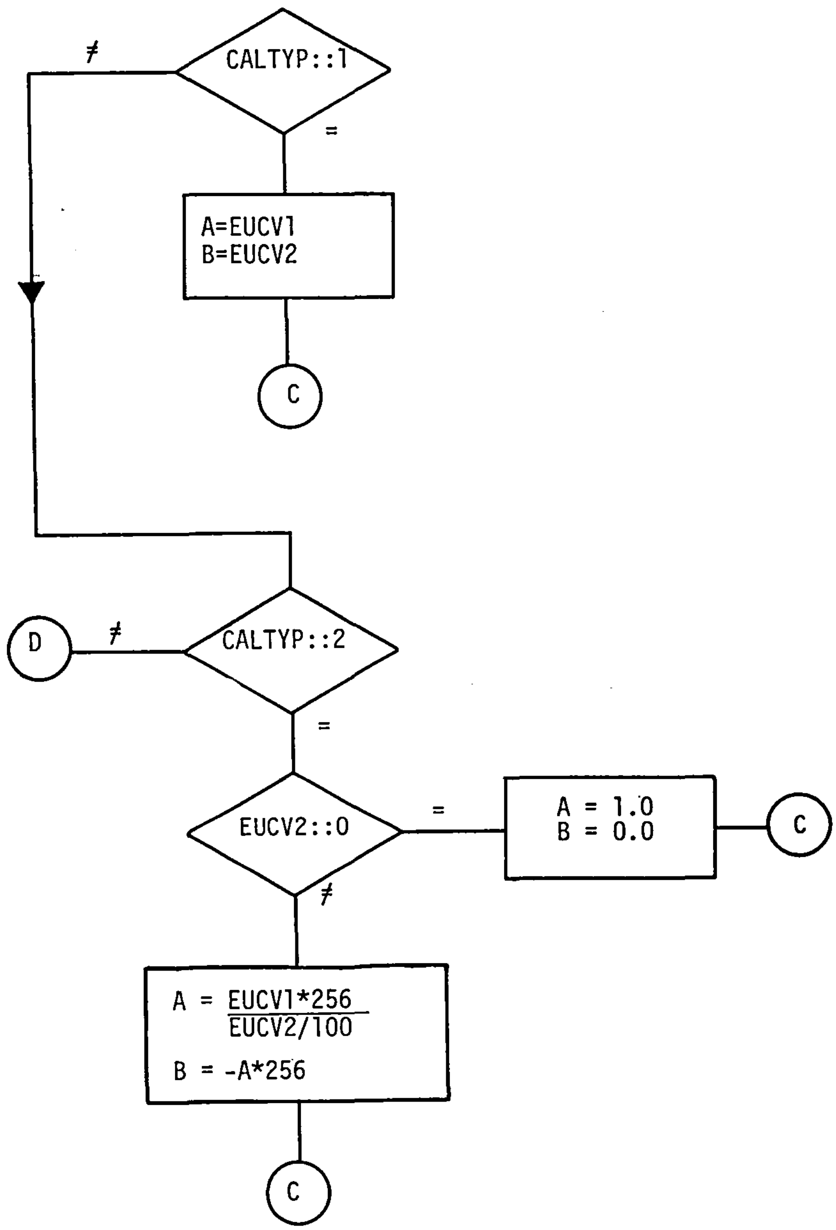
A(40)	CALCULADED A TERM - WRITTEN ON SENSOR TABLE
B(40)	CALCULATED B TERM - WRITTEN ON SENSOR TABLE
AVG(40,6)	AVERAGES FOR SPECIFIED PARAMETER
STD(40,6)	STANDARD DEVIATIONS FOR SPECIFIED PARAMETER
IPCFS	PERCENT FULL SCALE

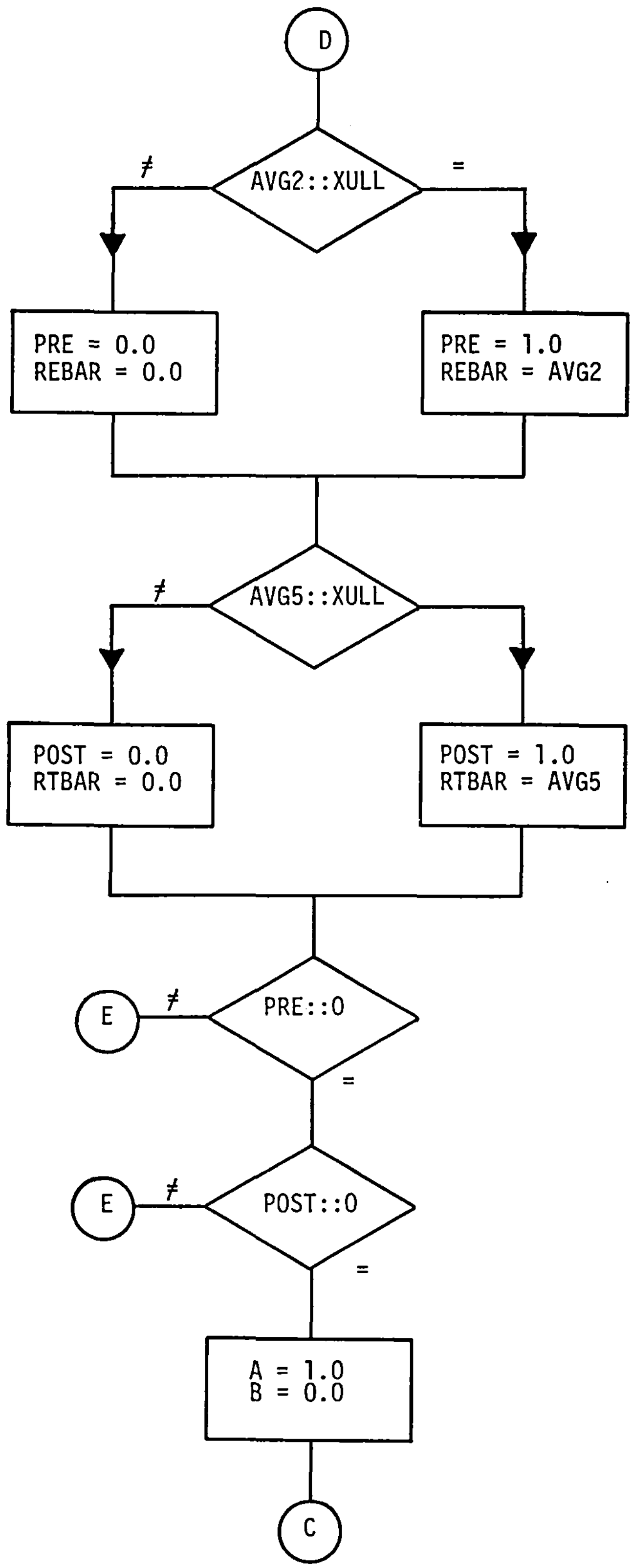
```
        ISHFT          CHANGE IN IPCFS BETWEEN PRE AND POST

********SUBROUTINES REQUIRED

        RDFRD
           RDFSUB
```

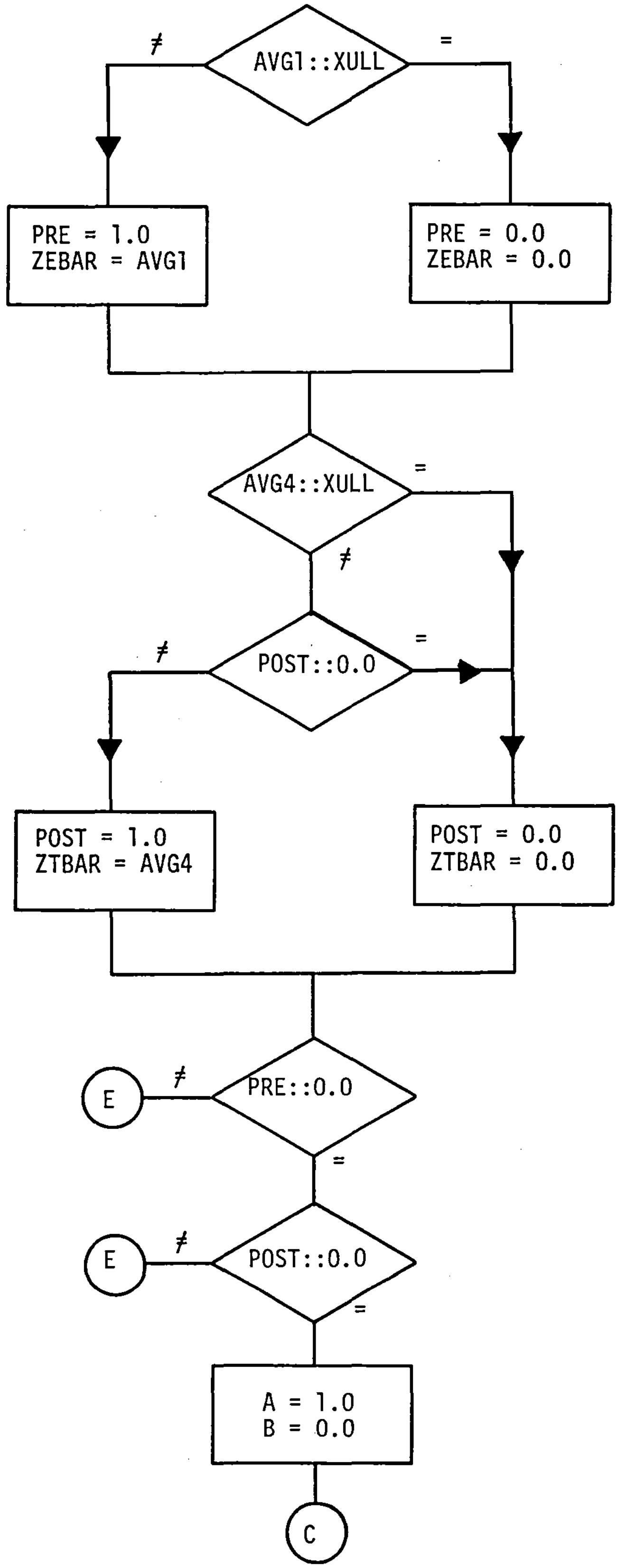

START
READ 7 HEADER TABLE
READ 7 CALRUN TABLE
DO 9000 I=1, MAXPAS
READ 7 PASS TABLE
ISTATC::1
≠
INITIALIZE AVG STD
READ 7 SENSOR TABLE
INITIALIZE TOS
PRINT PG 2 HEADER


A
DO 4000
K=1, 20
B
DO 1000
IPAR =
1, MAXPAR
CALADR::K
≠
=
WRITE
PG 2
DATA
DO 3000
JCAL =
1,6
IRUN =
CTAB
IRUN::0
=
CALL
RDFRD
IFR< 4
NO
PRINT
ERROR
MESSAGE
YES

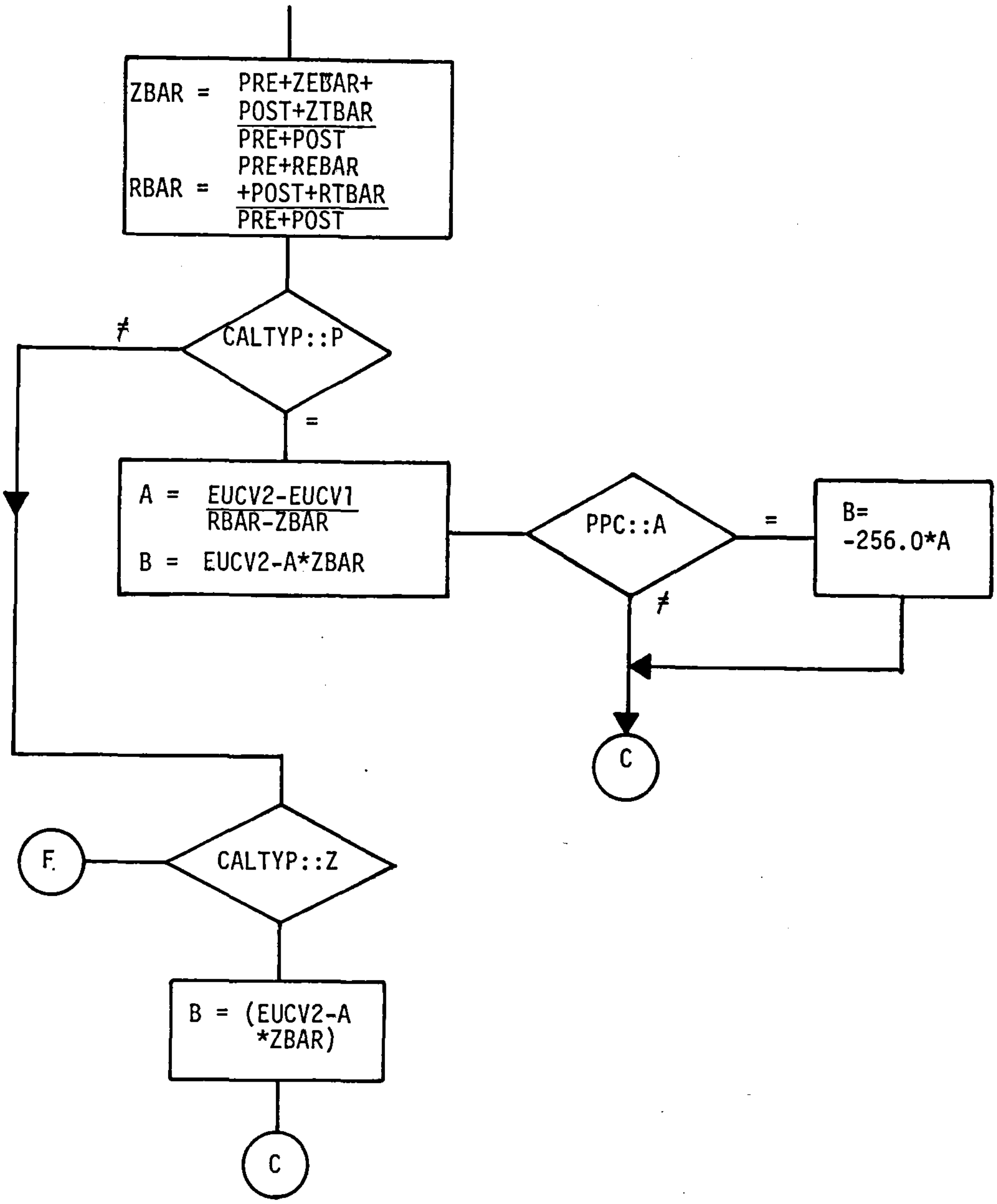

DO 2900
IPAR=1,
MAXPAR
CALADR::K
≠
=
ASSIEN
AVG
STD
2900
B
3000
A
4000
PRINT
PG1
HEADER
DO 8000
IPAR=1,
MAXPAR
PPC::V
=
≠

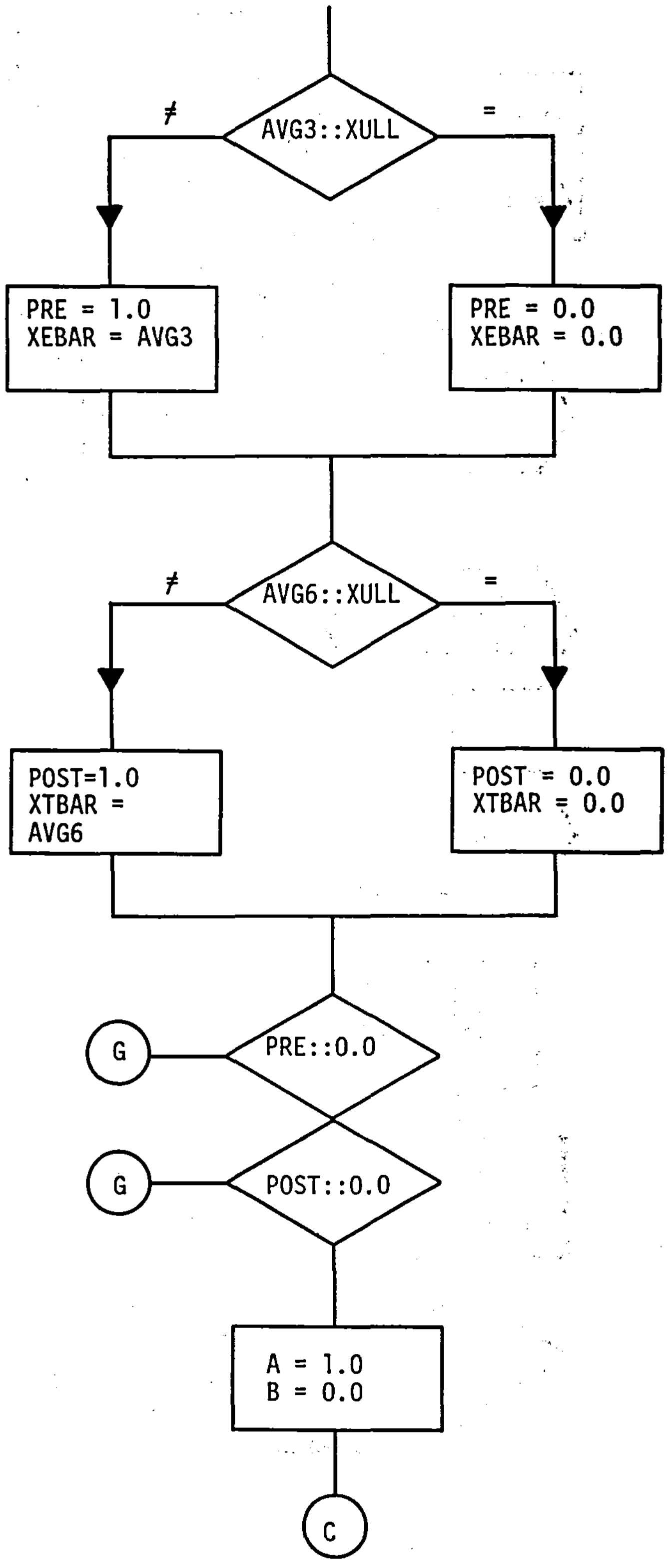

CALTYP::1
≠
=
A=EUCV1
B=EUCV2
C
D
≠
CALTYP::2
=
EUCV2::0
=
A = 1.0
B = 0.0
C
≠
A = EUCV1*256 / EUCV2/100
B = -A*256
C


D
AVG2::XULL
≠
=
PRE = 0.0
REBAR = 0.0
PRE = 1.0
REBAR = AVG2
AVG5::XULL
≠
POST = 0.0
RTBAR = 0.0
POST = 1.0
RTBAR = AVG5
E
≠
PRE::0
=
E
≠
POST::0
=
A = 1.0
B = 0.0
C

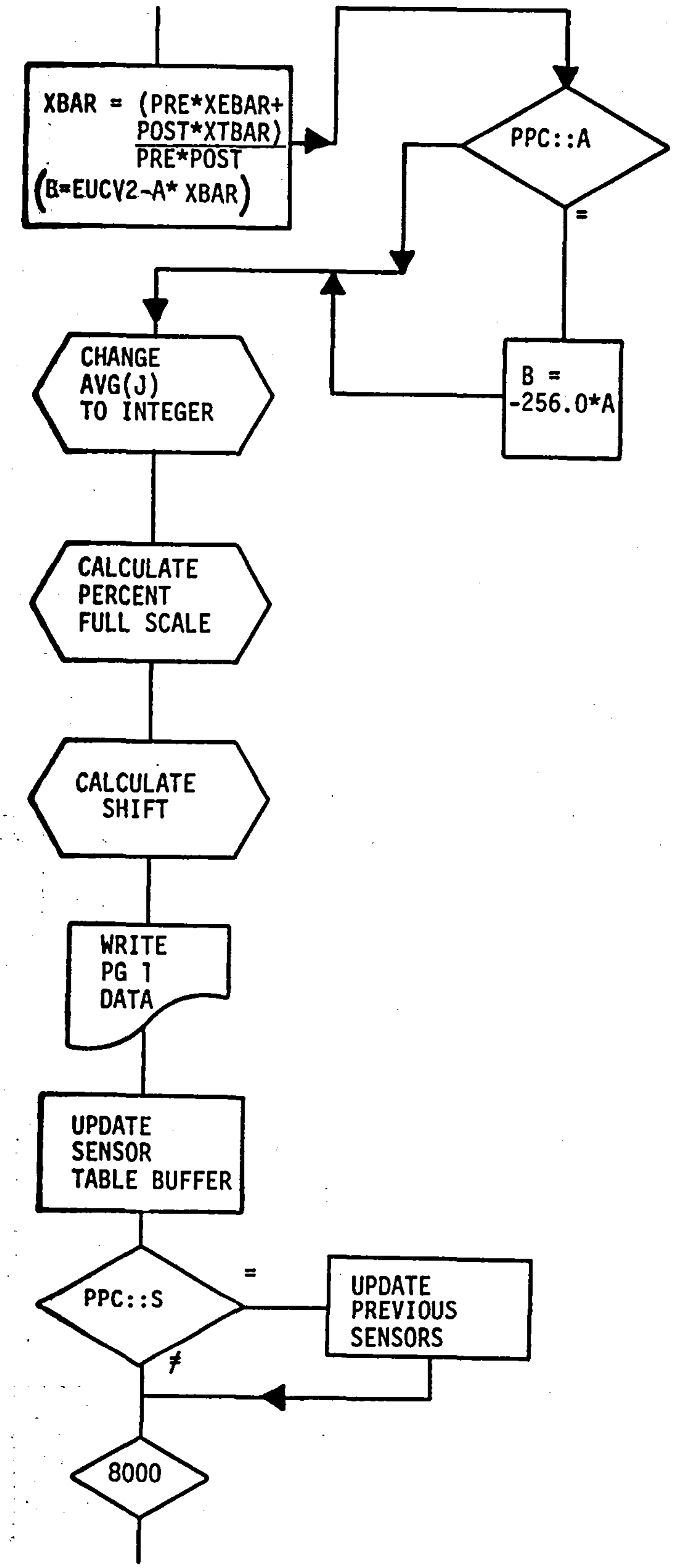


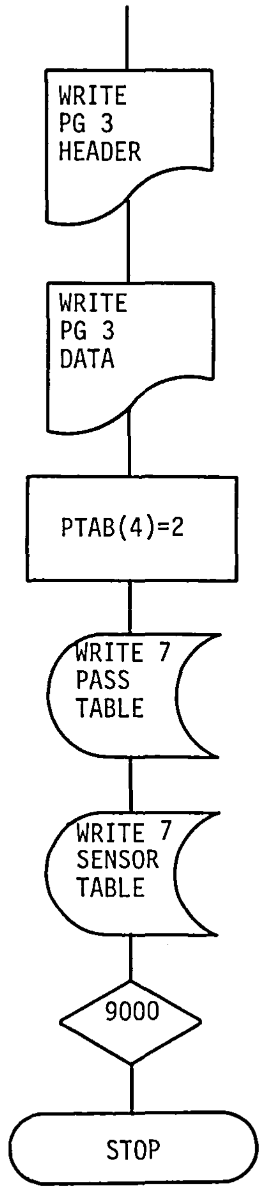
AVG1::XULL
≠
=
PRE = 1.0
ZEBAR = AVG1
PRE = 0.0
ZEBAR = 0.0
AVG4::XULL
=
≠
POST::0.0
≠
=
POST = 1.0
ZTBAR = AVG4
POST = 0.0
ZTBAR = 0.0
PRE::0.0
E
≠
=
POST::0.0
E
≠
=
A = 1.0
B = 0.0
C

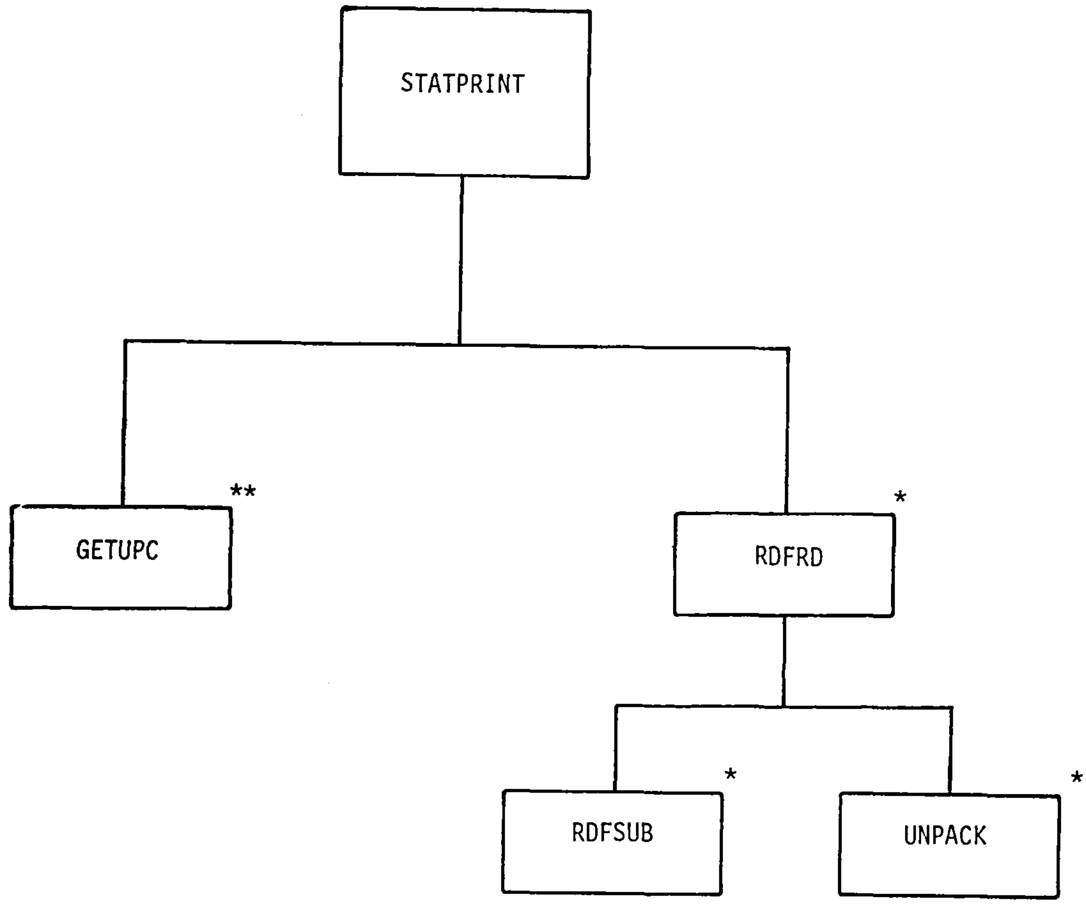

ZBAR = PRE*ZEBAR+POST*ZTBAR / PRE+POST
RBAR = PRE*REBAR+POST*RTBAR / PRE+POST
CALTYP::P
≠
=
A = EUCV2-EUCV1 / RBAR-ZBAR
B = EUCV2-A*ZBAR
PPC::A
=
≠
B= -256.0*A
C
F
CALTYP::Z
B = (EUCV2-A *ZBAR)
C



≠
AVG3::XULL
=
PRE = 1.0
XEBAR = AVG3
PRE = 0.0
XEBAR = 0.0
≠
AVG6::XULL
=
POST=1.0
XTBAR =
AVG6
POST = 0.0
XTBAR = 0.0
G
PRE::0.0
G
POST::0.0
A = 1.0
B = 0.0
C


XBAR = (PRE*XEBAR+ POST*XTBAR) / PRE*POST
(B=EUCV2-A* XBAR)
PPC::A
=
B = -256.0*A
CHANGE AVG(J) TO INTEGER
CALCULATE PERCENT FULL SCALE
CALCULATE SHIFT
WRITE PG 1 DATA
UPDATE SENSOR TABLE BUFFER
PPC::S
=
UPDATE PREVIOUS SENSORS
≠
8000


WRITE
PG 3
HEADER
WRITE
PG 3
DATA
PTAB(4)=2
WRITE 7
PASS
TABLE
WRITE 7
SENSOR
TABLE
9000
STOP



*See Sub-program RDFRD for Comments and Flowcharts

**See Program DATASK for Comments and Flowchart

HIERARCHY CHART For STATPRINT PROGRAM

STPRT STATISTICS PRINT-OUT
********STATPRINT - PROGRAM NUMBER 1.1.2327

********NASA WALLOPS VERSION OF 08/01/77

********LANGUAGE - FORTRAN Y

********MACHINE - HW 625/635

********PROGRAM AUTHOR - JEANETTE WESSELLS

********PURPOSE -

FOR ALL DATA? ALL DATA OF A GIVEN PASS? SELECTED PARAMETERS FROM ANY PASS, GENERATE STATISTICS TABULATION ON ALL BURSTS IN EITHER RAW UNITS OR ENGINEERING UNITS.

********METHOD -

STATPRINT EXTRACTS STATISTICS FROM FILE AND PRODUCES PRINTED OUTPUT OF COMPUTED MANEUVER STATISTICS INCLUDING MAXIMUM, MINIMUM, AVERAGE, STANDARD DEVIATION, NO. OF POINTS, NO. OF ERROR POINTS, AND ERROR CODE IN COLUMN FORMAT FOR EACH PARAMETER.

THE AMOUNT OF OUTPUT IS CONTROLLED BY INPUT CARDS WHICH MAY CONTAIN PASS NUMBER, MNEUMONIC NAME, PRE-PROCESSING CODE, SAMPLE RATE FOR PARAMETER, AND UNITS DESIGNATOR CODE. THESE INPUT CARDS CONTROL 3 MODES OF LOGIC<

MODE1 - IPASS, NAME, PPC ARE NOT SPECIFIED YIELDS LISTING OF ALL PARAMETERS FOR ALL AVAILABLE PASSES

MODE2 - IPASS IS SPECIFIED YIELDS LISTING OF ALL PARAMETERS FOR A GIVEN PASS

MODE3 - NAME, PPC, AND SR IS SPECIFIED YIELDS LISTING FOR ONLY THAT PARAMETER

ALL INFORMATION IS IN EITHER RAW UNITS OR ENGINEERING UNITS DEPENDING ON THE VALUE OF ICONV.

DATA IS ACCESSED THROUGH RDFRD MODULE.

********SYSTEMS INPUT FILES -

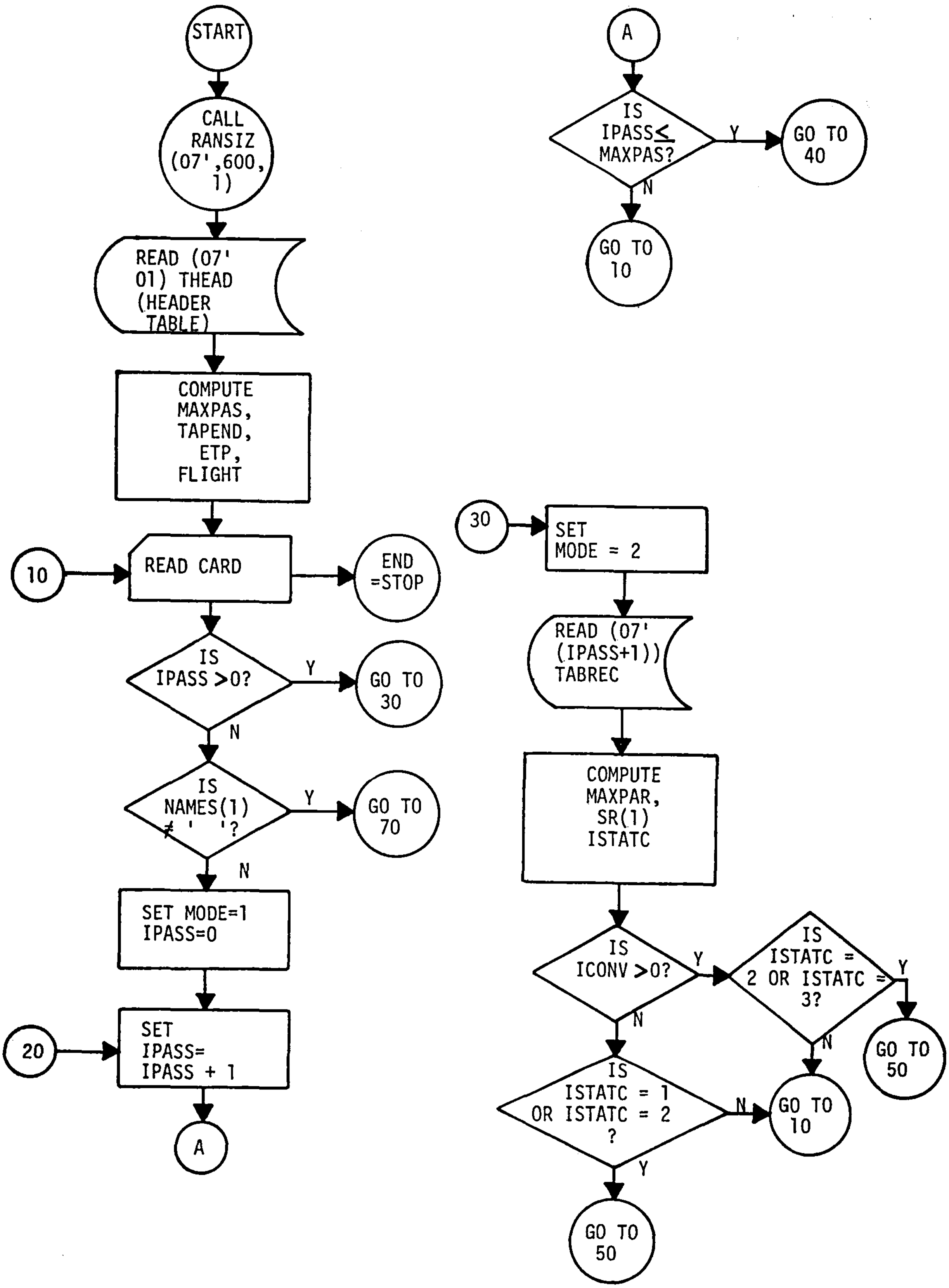
FILE 05 = CARD READER

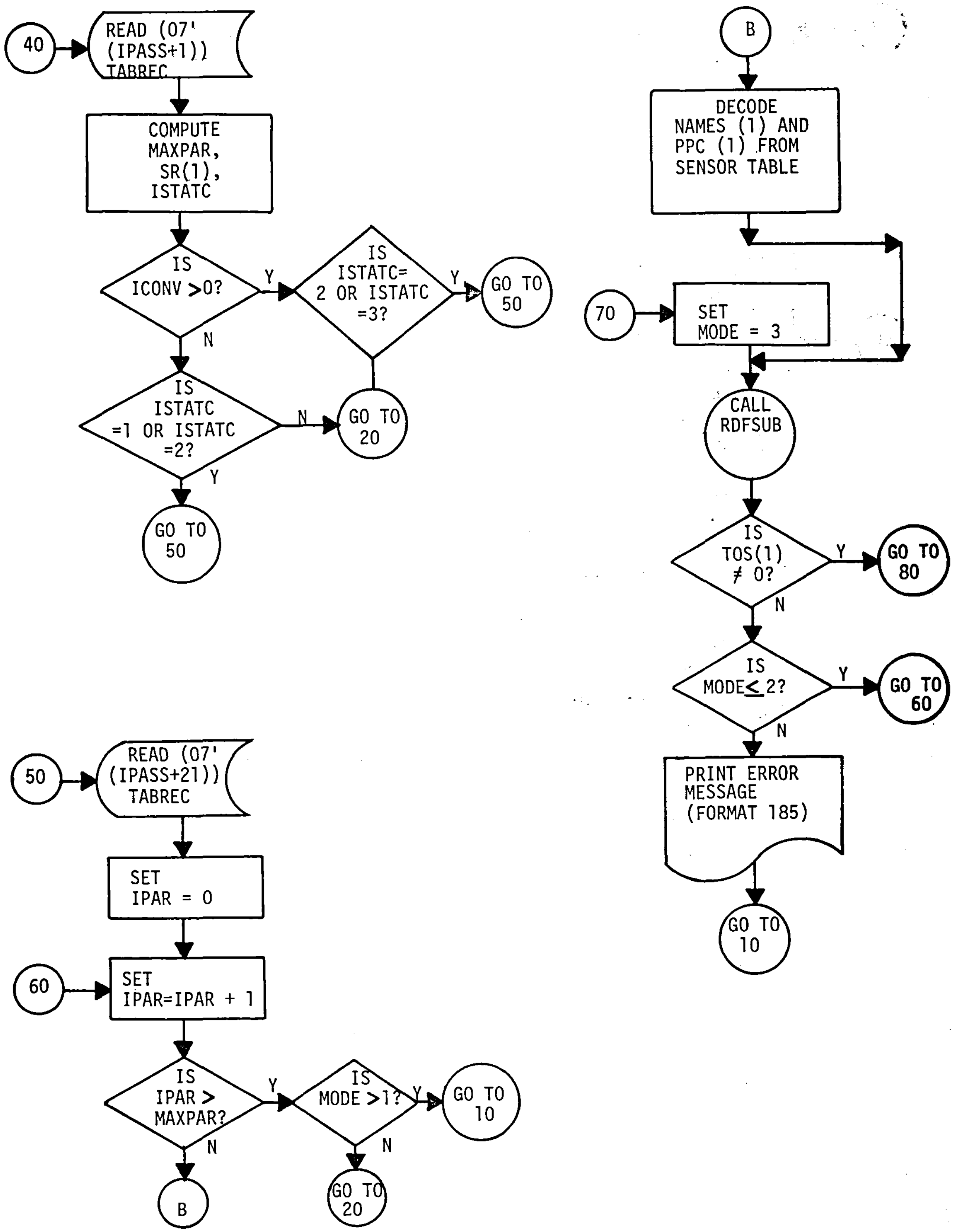
FILE 07 = TABLE FILE

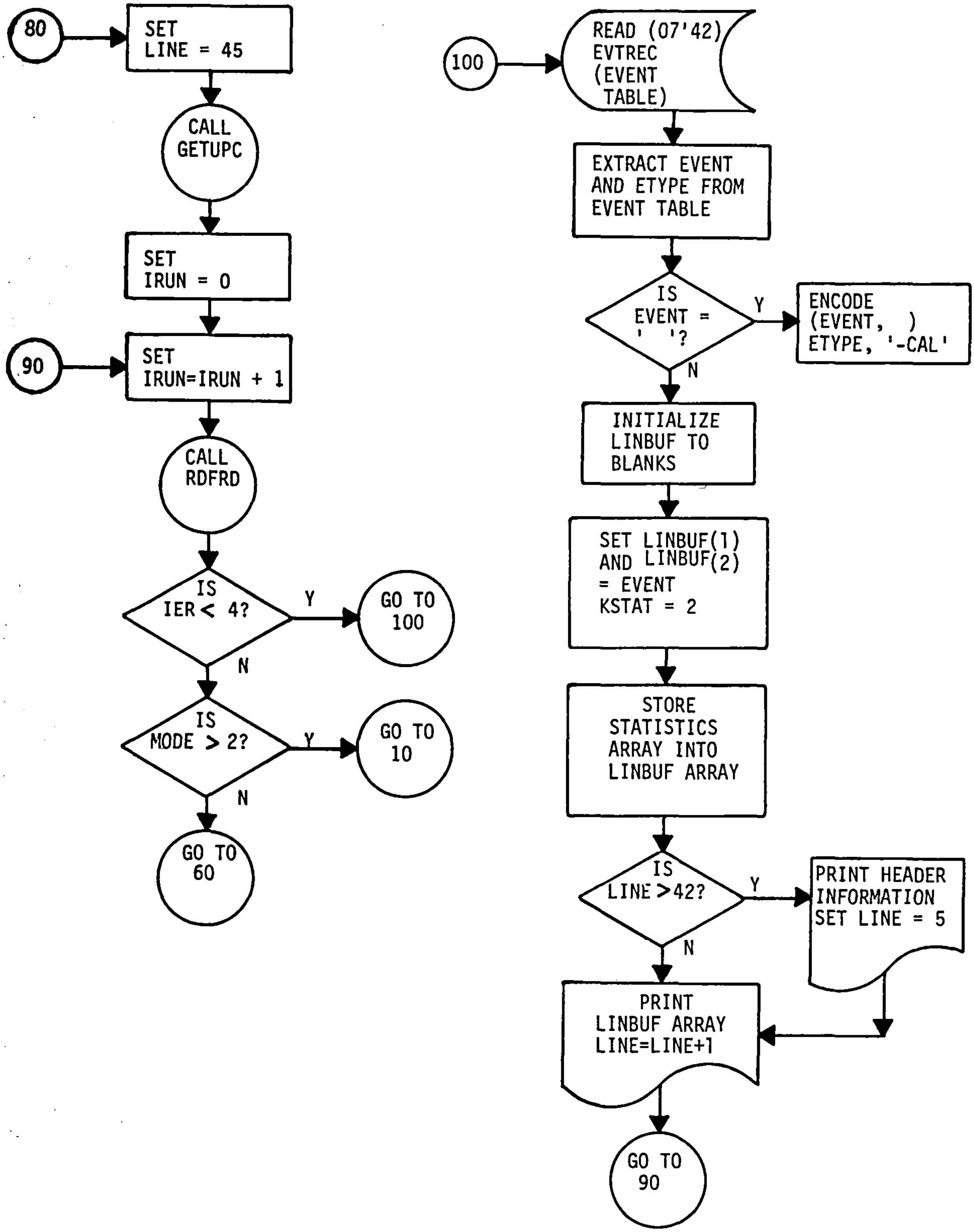
FILE 09 = STATISTICS FILE

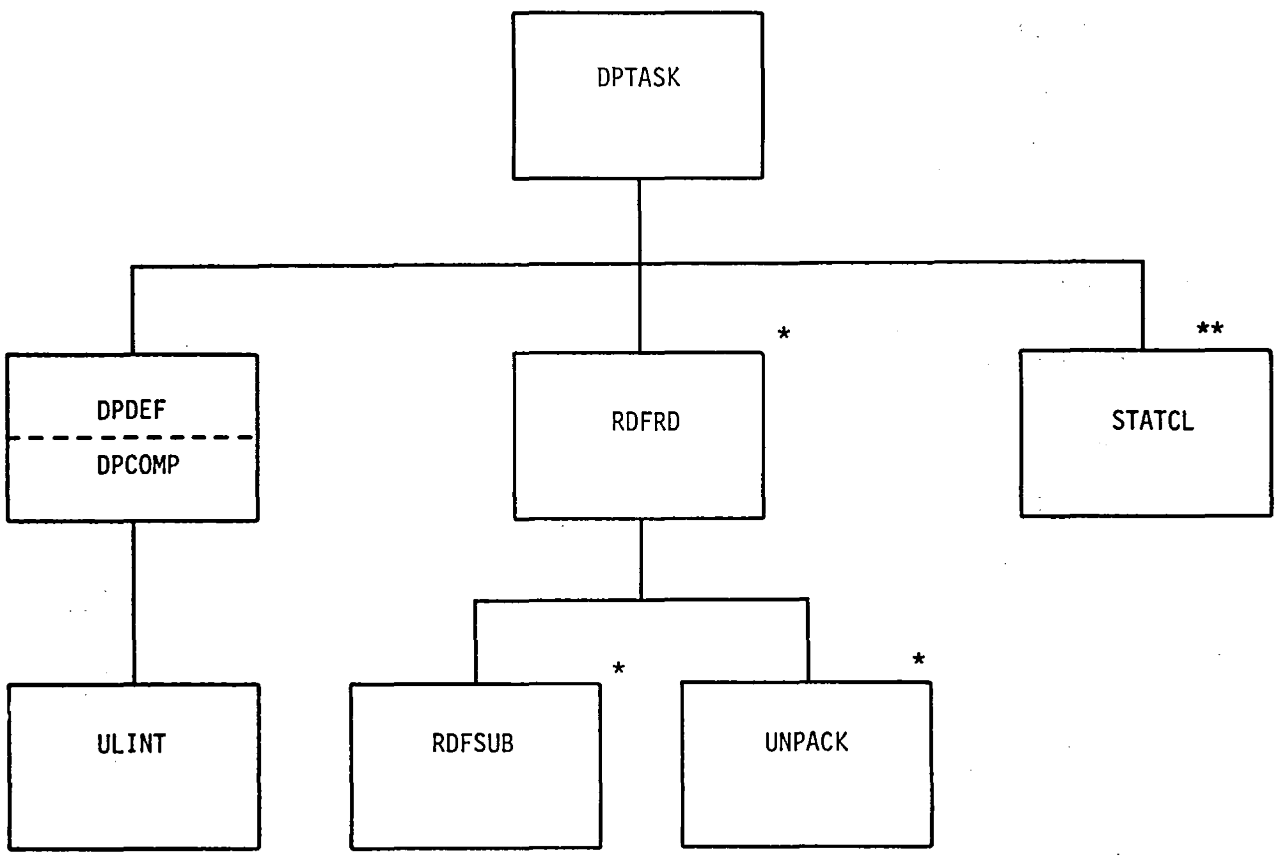
```
********SYSTEMS OUTPUT FILES -

        FILE 06 = PRINTER

        FILE 07 = TABLE FILE

********INPUT -

        IPASS       -PASS NUMBER (CARD INPUT)

        NAMES       -ARRAY OF PARAMETER MNEMONICS (CARD INPUT)

        PPC         -PARAMETER PRE-PROCESSING CODE

        SR          -PARAMETER SAMPLE RATE

        ICONV       -UNITS DESIGNATOR
                    -=0,OUTPUT RAW UNITS
                    -=1,OUTPUT ENGINEERING UNITS

********OUTPUT -

        TAPENO      -FLIGHT TAPE MNEMONIC

        FTP         -FLIGHT TEST PLAN NUMBER

        FLIGHT      -FLIGHT NUMBER

        UNIT        -OUTPUT IN RAW OR ENGINEERING UNITS

        IPASS       -PASS NUMBER

        NAMES       -PARAMETER MNEMONIC

        PPC         -PARAMETER PRE-PROCESSING CODE

        SR          -PARAMETER SAMPLE RATE

        LTITL       -LINE TITLE FOR SENSOR MNEMONIC

        IRUN        -RUN NUMBER

        LINBUF      -OUTPUT STATISTICS ARRAY
        LINBUF(1)  -MANEUVER NAME
        LINBUF(2)  -MANEUVER NAME
        LINBUF(3)  -MAXIMUM VALUE
        LINBUF(4)  -MINIMUM VALUE
        LINBUF(5)  -AVERAGE
        LINBUF(6)  -STANDARD DEVIATION
        LINBUF(7)  -95 PERCENTILE
        LINBUF(8)  -NUMBER OF POINTS
        LINBUF(9)  -NUMBER OF ERROR POINTS
        LINBUF(10)-TYPE OF ERROR CODE
                   -=000100,LOSS OF SYNC ERROR
                   -=000010,FULL SCALE ERROR
                   -=000001,PARITY ERROR
```

```
********RESTRICTIONS -
        NONE

********SUBPROGRAMS REQUIRED -
        RDFRD
             RDFSUB
             UNPACK
        GETUPC
```


START
CALL RANSIZ (07',600, 1)
READ (07' 01) THEAD (HEADER TABLE)
COMPUTE MAXPAS, TAPEND, ETP, FLIGHT
10
READ CARD
END =STOP
IS IPASS >0?
Y
GO TO 30
N
IS NAMES(1) ≠ ' '?
Y
GO TO 70
N
SET MODE=1 IPASS=0
20
SET IPASS= IPASS + 1
A
A
IS IPASS≤ MAXPAS?
Y
GO TO 40
N
GO TO 10
30
SET MODE = 2
READ (07' (IPASS+1)) TABREC
COMPUTE MAXPAR, SR(1) ISTATC
IS ICONV >0?
Y
IS ISTATC = 2 OR ISTATC = 3?
Y
GO TO 50
N
GO TO 10
N
IS ISTATC = 1 OR ISTATC = 2 ?
N
Y
GO TO 50


40
READ (07'
(IPASS+1))
TABREC
COMPUTE
MAXPAR,
SR(1),
ISTATC
IS
ICONV >0?
Y
N
IS
ISTATC=
2 OR ISTATC
=3?
Y
GO TO
50
GO TO
20
IS
ISTATC
=1 OR ISTATC
=2?
N
Y
GO TO
50
B
DECODE
NAMES (1) AND
PPC (1) FROM
SENSOR TABLE
70
SET
MODE = 3
CALL
RDFSUB
IS
TOS(1)
≠ 0?
Y
GO TO
80
N
IS
MODE≤2?
Y
GO TO
60
N
PRINT ERROR
MESSAGE
(FORMAT 185)
GO TO
10
50
READ (07'
(IPASS+21))
TABREC
SET
IPAR = 0
60
SET
IPAR=IPAR + 1
IS
IPAR >
MAXPAR?
Y
N
IS
MODE >1?
Y
GO TO
10
N
B
GO TO
20


80
SET LINE = 45
CALL GETUPC
SET IRUN = 0
90
SET IRUN=IRUN + 1
CALL RDFRD
IS IER < 4?
Y
GO TO 100
N
IS MODE > 2?
Y
GO TO 10
N
GO TO 60
100
READ (07'42) EVTREC (EVENT TABLE)
EXTRACT EVENT AND ETYPE FROM EVENT TABLE
IS EVENT = ' '?
Y
ENCODE (EVENT,) ETYPE, '-CAL'
N
INITIALIZE LINBUF TO BLANKS
SET LINBUF(1) AND LINBUF(2) = EVENT KSTAT = 2
STORE STATISTICS ARRAY INTO LINBUF ARRAY
IS LINE >42?
Y
PRINT HEADER INFORMATION SET LINE = 5
N
PRINT LINBUF ARRAY LINE=LINE+1
GO TO 90



*See Sub-program RDFRD for Comments and Flowcharts
**See Program SCAN for Comments and Flowcharts

HIERARCHY CHART for DPTASK PROGRAM

```
      SUBROUTINE DPTASK
********DPTASK******************************************************************
```

PROGRAM IDENTIFICATION

PROGRAM NAME ----- DPTASK
PROGRAM NUMBER ----- 112338
AUTHOR ----- TERRY D. SOMMERS

COMPUTER ----- HW625/635
MEMORY ----- ALL FILES OPEN(23K)
PERIPHERALS ----- CARD READER, DISC , PRINTER
LANGUAGE ----- HW6000 FORTRAN/FORTY

PURPOSE

THE PURPOSE OF THIS PROGRAM IS TO COMPUTE DERIVED PARAMETERS FROM PARAMETERS PREVIOUSLY SPECIFIED IN A FRAME OF DATA AND TO WRITE THESE DERIVED PARAMETERS ON A DATA FILE.

METHOD

A SUBROUTINE WITH TWO ENTRY POINTS IS REQUIRED.

1. CALL DPDEF - THIS IS AN INITIALIZATION CALL TO BE MADE ONCE PRIOR TO BEGINNING EACH DP PASS. ITS PROCESS IS TO DEFINE FOR THE DPCOMP ROUTINE WHAT PASS IS TO BE PROCESSED AND IN RETURN IT WILL RETURN TO THE CALLING PROGRAM A LIST OF SPECIFICATIONS FOR THE PARAMETERS THAT IT WILL BE NEEDING AND FOR THE PARAMETERS IT WILL BE GENERATING.
2. CALL DPCOMP - THIS ENTRY IS MADE FOR EACH FRAME OF DATA.

A SUBROUTINE ULINT IS ALSO REQUIRED TO PERFORM UNIVARIATE LINEAR INTERPOLATION.

INPUT/OUTPUT

IPASS - A NOMINAL PASS NO. TO BE PROCESSED. THIS IS THE ONLY INPUT ARGUMENT TO THIS ENTRY. IT IS THE NOMINAL PASS NO. AS OPPOSED TO THE ACTUAL PASS NO. IN THAT A VALUE OF 1 WILL INDICATE THE FIRST DP PASS EVEN THOUGH THE ACTUAL PASS NO. MAY BE SOMETHING OTHER THAN 1. A VALUE OF 2 WILL INDICATE THE SECOND DP PASS ETC. IF THE PASS NO. SPECIFIED DOES NOT EXIST THIS ARGUMENT WILL BE RETURNED TO THE CALLING PROGRAM AS A ZERO INDICATING THAT DP IS COMPLETE.

NOPF - THE NO. OF VALUES THAT ARE REQUIRED FROM THE ONCE PER FLIGHT SECTION OF THE HEADER TABLE.

IOPFLC - AN ARRAY OF ONCE PER FLIGHT ENTRY NOS. NOPF IN LENGTH, DEFINING WHICH ONCE PER FLIGHT VALUES WILL BE NEEDED.

NSTREQ - THE NO. OF PARAMETER STATISTICS WHICH MAY BE REQUIRED.

ISTAT - AN ARRAY WITH NSTREQ ENTRIES EACH OF WHICH WILL BE A NUMBER FROM 1 TO 10 INDICATING THE STATISTIC ELEMENT WHICH WILL BE REQUIRED.

```
STPAR - AN ARRAY WITH NSTREQ ENTRIES EACH OF WHICH IS A
        PARAMETER MNEMONIC INDICATING THE PARAMETER WHOSE
        STATISTIC WILL BE REQUIRED.
STPPC - AN ARRAY WITH NSTREQ ENTRIES EACH OF WHICH IS A PPC
        CORRESPONDING TO THE MNEMONICS OF THE ABOVE ARGUMENT.
STSR - AN ARRAY WITH NSTREQ ENTRIES EACH OF WHICH IS A
       SAMPLE RATE CORRESPONDING TO THE MNEMONIC ABOVE.
NPAREQ - THE NO. OF PARAMETERS THAT WILL BE REQUIRED AS INPUT
         ON A FRAME BY FRAME BASIS.
SR - THE SAMPLE RATE OF THE PARAMETER REQUIRED ON A FRAME
     BASIS. THIS WILL ALSO BE THE SR OF THE PARAMETERS TO
     BE COMPUTED ON A FRAME BASIS.
PARNAM - AN ARRAY OF NPAREQ LENGTH DEFINING THE MNEMONICS OF
         THE PARAMETERS THAT WILL BE NEEDED AS INPUT ON A
         FRAME BASIS.
PPC - AN ARRAY OF NPAREQ LENGTH DEFINING THE CORRESPONDING
      PRE-PROCESS CODES FOR THE ABOVE PARAMETERS.
NPOUT - THE NO. OF PARAMETERS THAT WILL BE COMPUTED FOR THIS
        PASS ON A FRAME BASIS.
PAROT - AN ARRAY OF LENGTH NPOUT CONTAINING THE MNEMONICS OF
        THE PARAMETERS THAT WILL BE GENERATED ON A FRAME BASIS.
PPCOT - AN ARRAY OF LENGTH NPOUT CONTAINING THE PRE-PROCESS
        CODES FOR THE ABOVE PARAMETERS.
ITRANS - = ZERO IF STEADY STATE CONDITION, ONE IF TRANSIENT
           CONDITION
ISTFRM - = ZERO IF THIS FRAME IS THE FIRST FRAME IN THE BURST
           = ONE IF NOT FIRST FRAME OF BURST
OPFVAL - ARRAY OF ONCE PER FLIGHT VALUES, NOPF IN LENGTH
STVAL - ARRAY OF STATISTIC VALUES, NSTREQ IN LENGTH
PARVAL - ARRAY OF INPUT FRAME VALUES, NPAREQ IN LENGTH
DPVAL - ARRAY OF DP VALUES RETURNED TO CALLING PROGRAM,
        NPOUT IN LENGTH
```

CALLING SEQUENCE

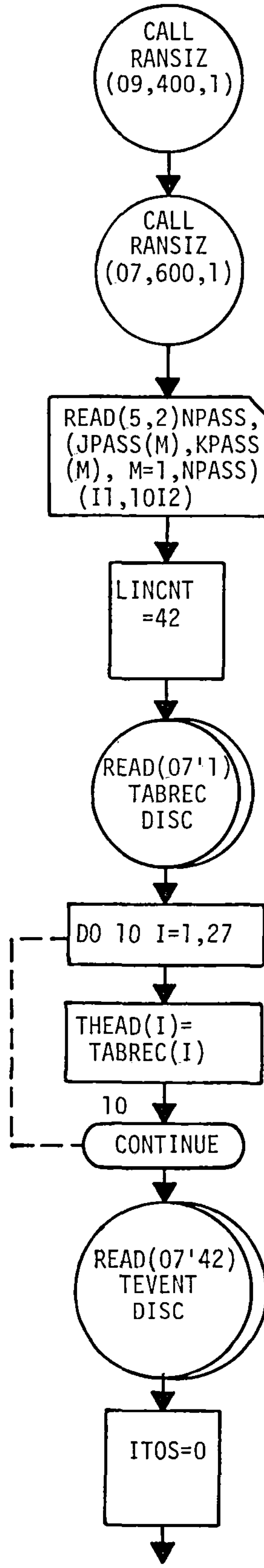
```
CALL DPDEF(IPASS,NOPF,IOPFLC,NSTREQ,ISTAT,STPAR,STPPC,STSR,
          NPAREQ,SR,PARNAM,PPC,NPOUT,PAROT,PPCOT)
CALL RDFRD(ITOST,1,1,IRUN,NSTREQ,STPAR,STPPC,STSR ,0,0,OUTBUF,
          (10*NSTREQ),TOSTAT,NFOUT,MSG,IER)
CALL DPCOMP(ITRANS ,0,OPFVAL,STVAL,PARVAL,DPVAL)
CALL RDFRD(ITOS,0,1,IRUN,(NPAREQ+1),PARNAM,PPC,SR,ISMPL,NSMPL,
          PARVAL,(15*(NPAREQ+1)),TOS,NFOUT,MSG,IER)
CALL DPCOMP(ITRANS,ISTFRM,OPFVAL,STVAL,VALIN,DPVAL)
CALL STATCL(ISTFRM ,DPVAL,NPOUT,PPCOT,EUCV1,EUCV2,LOCOT,STATBF)
CALL STATCL(2,IFRMBF ,NPOUT ,PPCOT,EUCV1,EUCV2,LOCOT,STATBF)
```

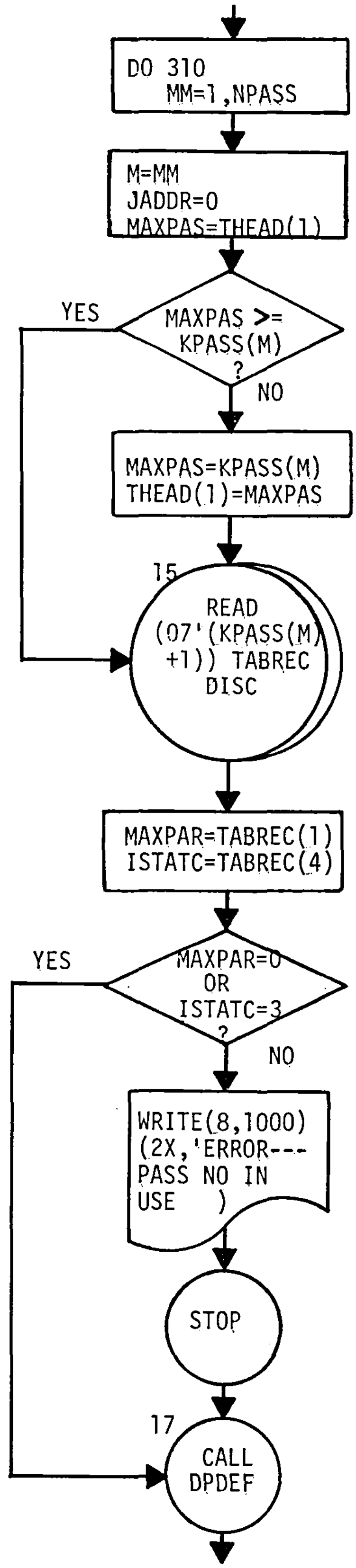
COMMON AREAS

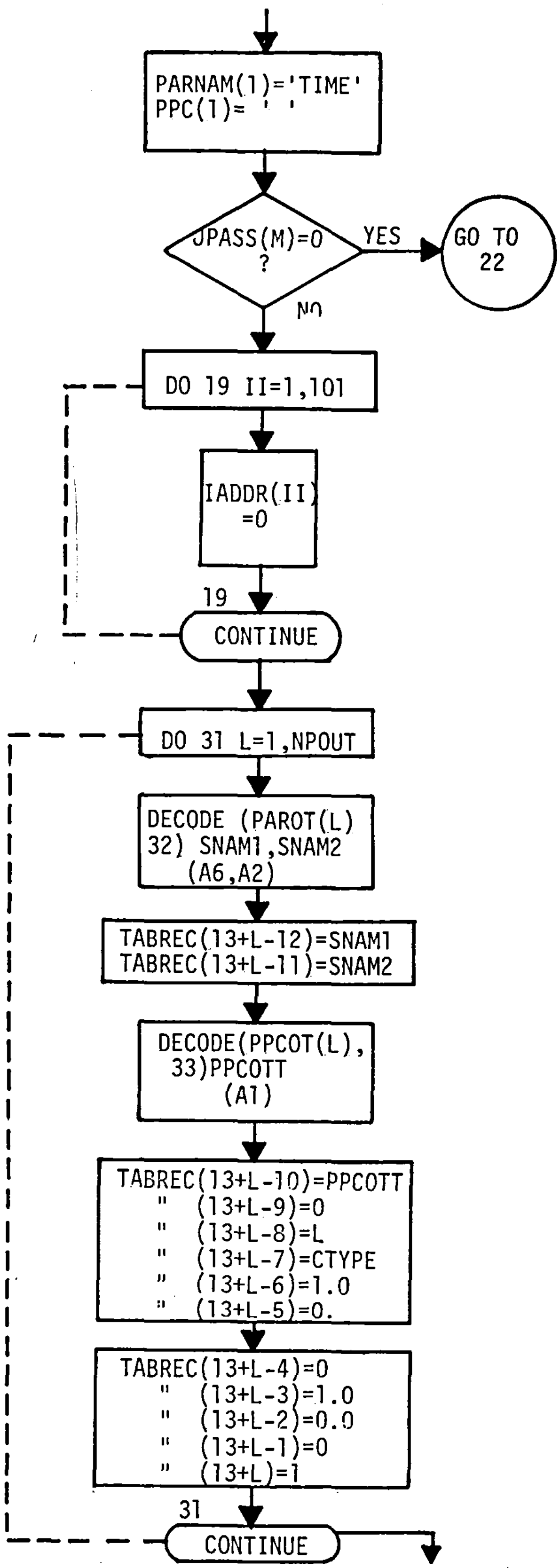
TABL1/THEAD

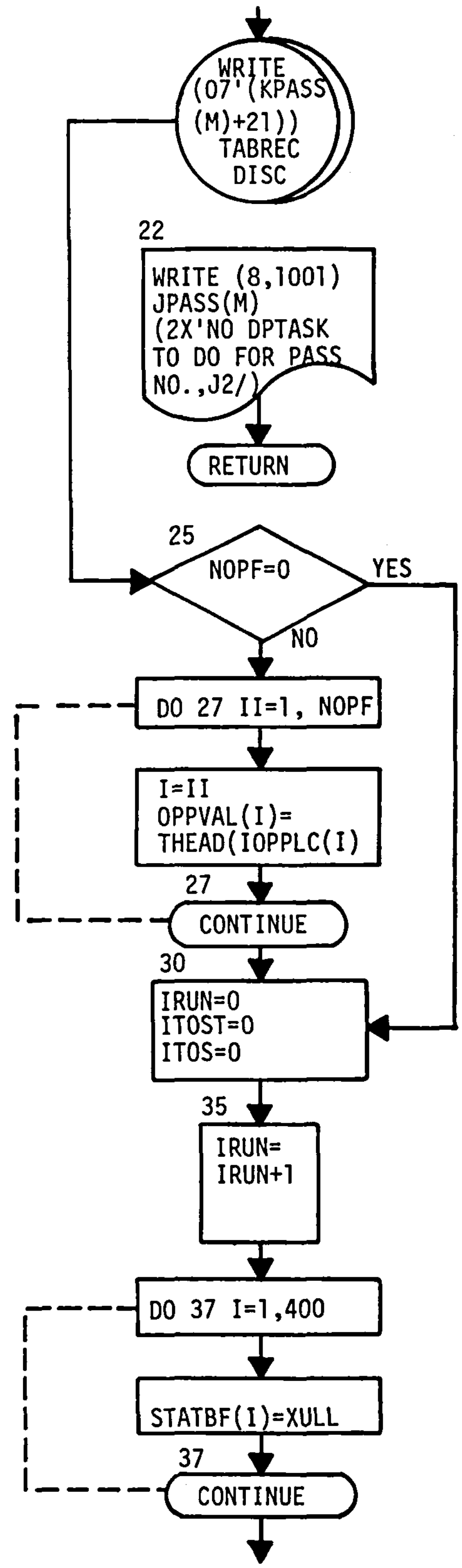
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS , FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION)

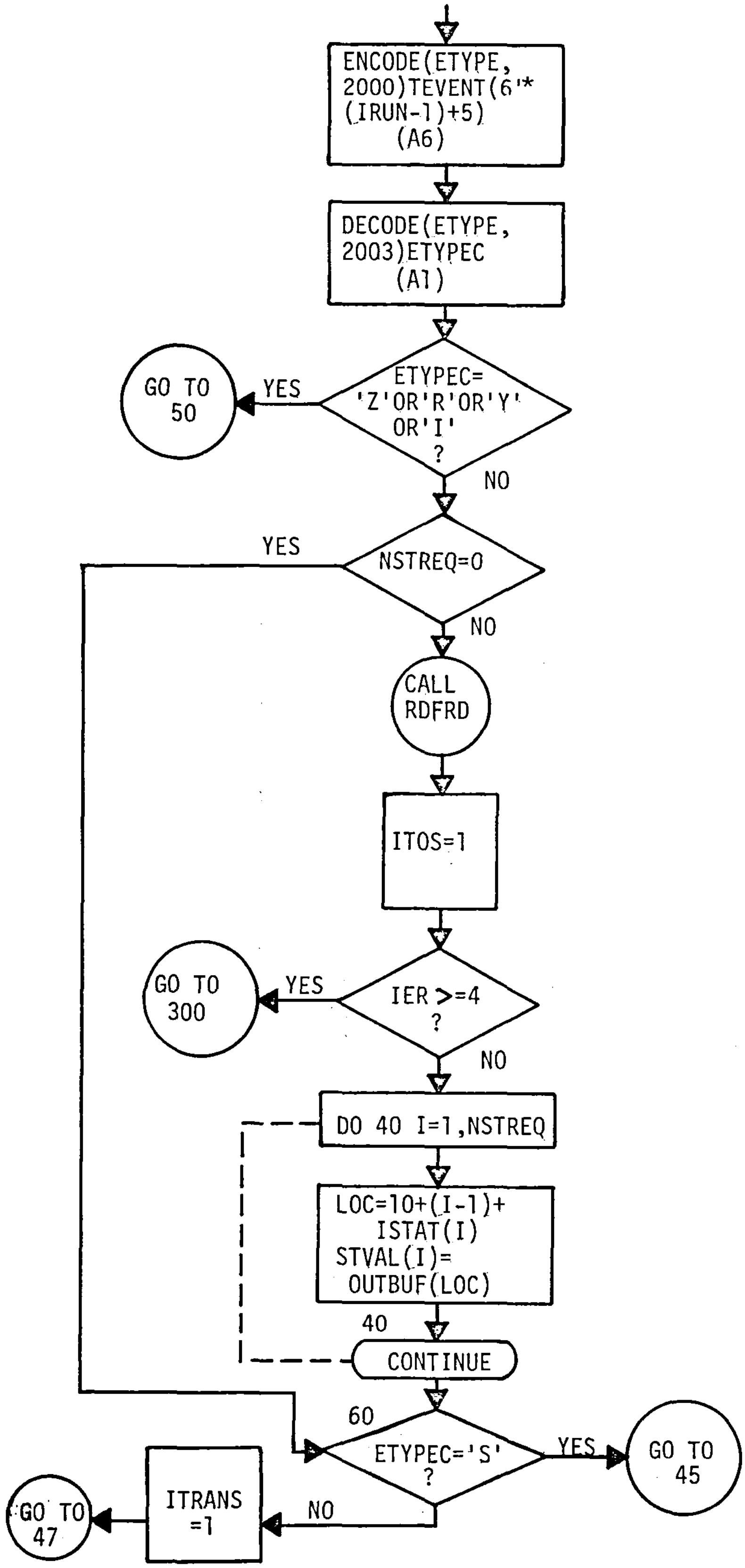
****

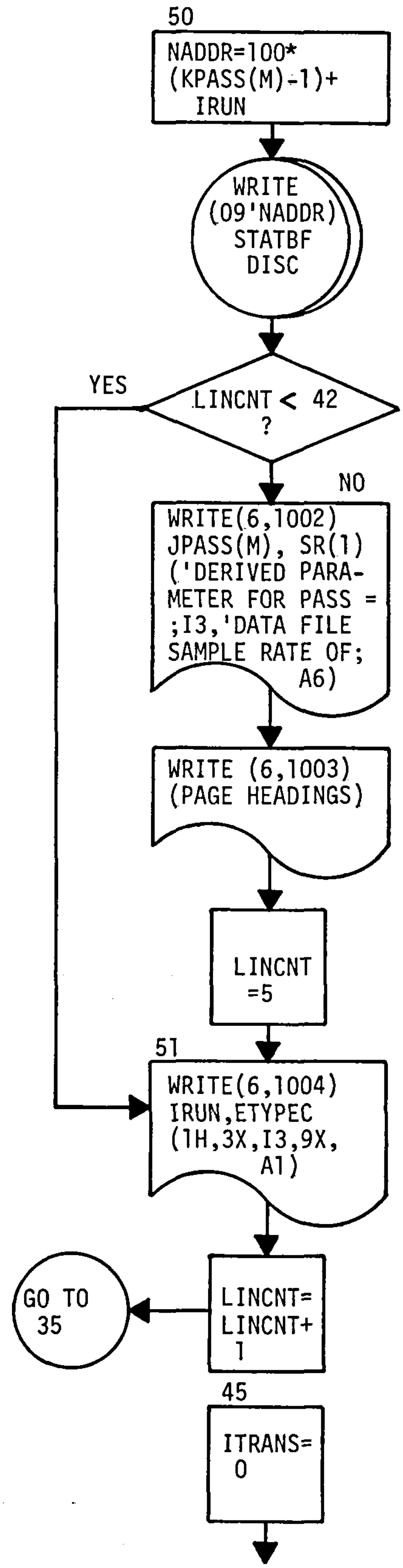

CALL
RANSIZ
(09,400,1)
CALL
RANSIZ
(07,600,1)
READ(5,2)NPASS,
(JPASS(M),KPASS
(M), M=1,NPASS)
(I1,10I2)
LINCNT
=42
READ(07'1)
TABREC
DISC
DO 10 I=1,27
THEAD(I)=
TABREC(I)
10
CONTINUE
READ(07'42)
TEVENT
DISC
ITOS=0

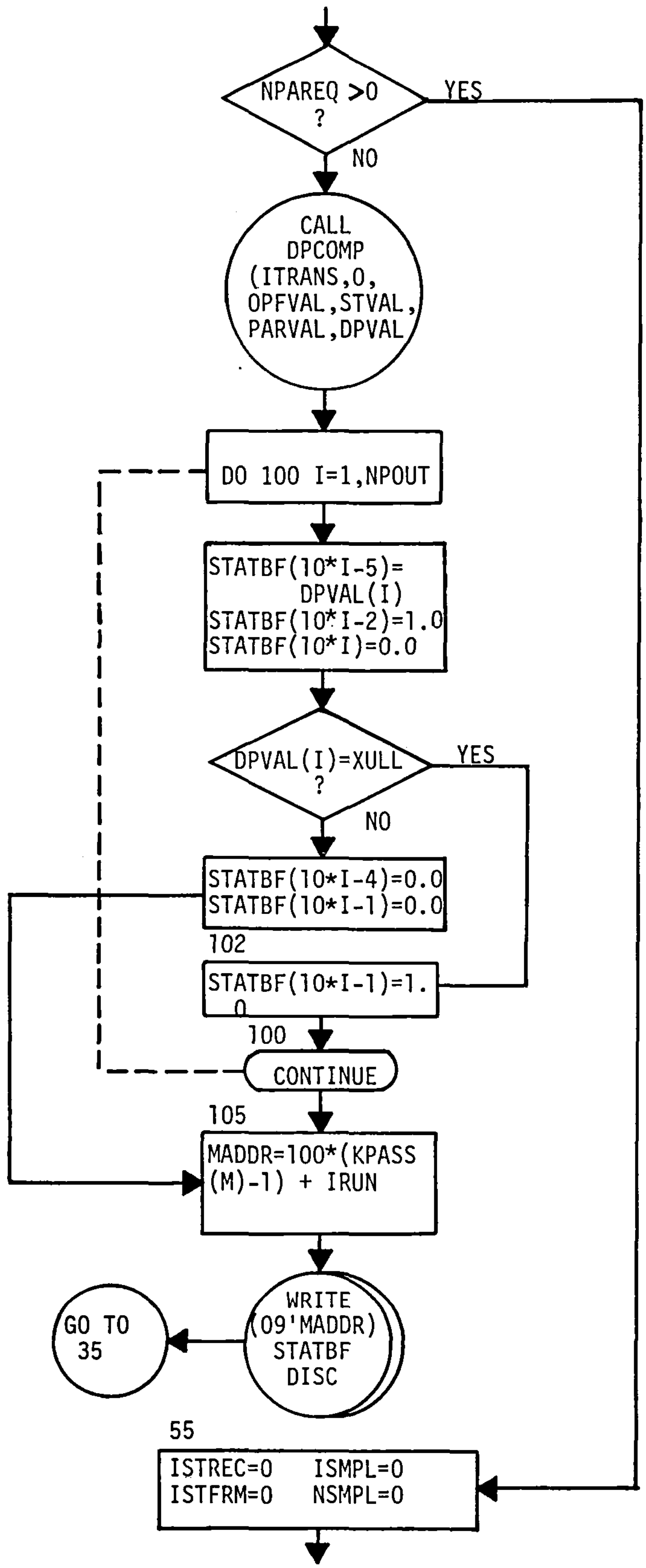

DO 310
MM=1,NPASS
M=MM
JADDR=0
MAXPAS=THEAD(1)
YES
MAXPAS >=
KPASS(M)
?
NO
MAXPAS=KPASS(M)
THEAD(1)=MAXPAS
15
READ
(07'(KPASS(M)
+1)) TABREC
DISC
MAXPAR=TABREC(1)
ISTATC=TABREC(4)
YES
MAXPAR=0
OR
ISTATC=3
?
NO
WRITE(8,1000)
(2X,'ERROR---
PASS NO IN
USE)
STOP
17
CALL
DPDEF

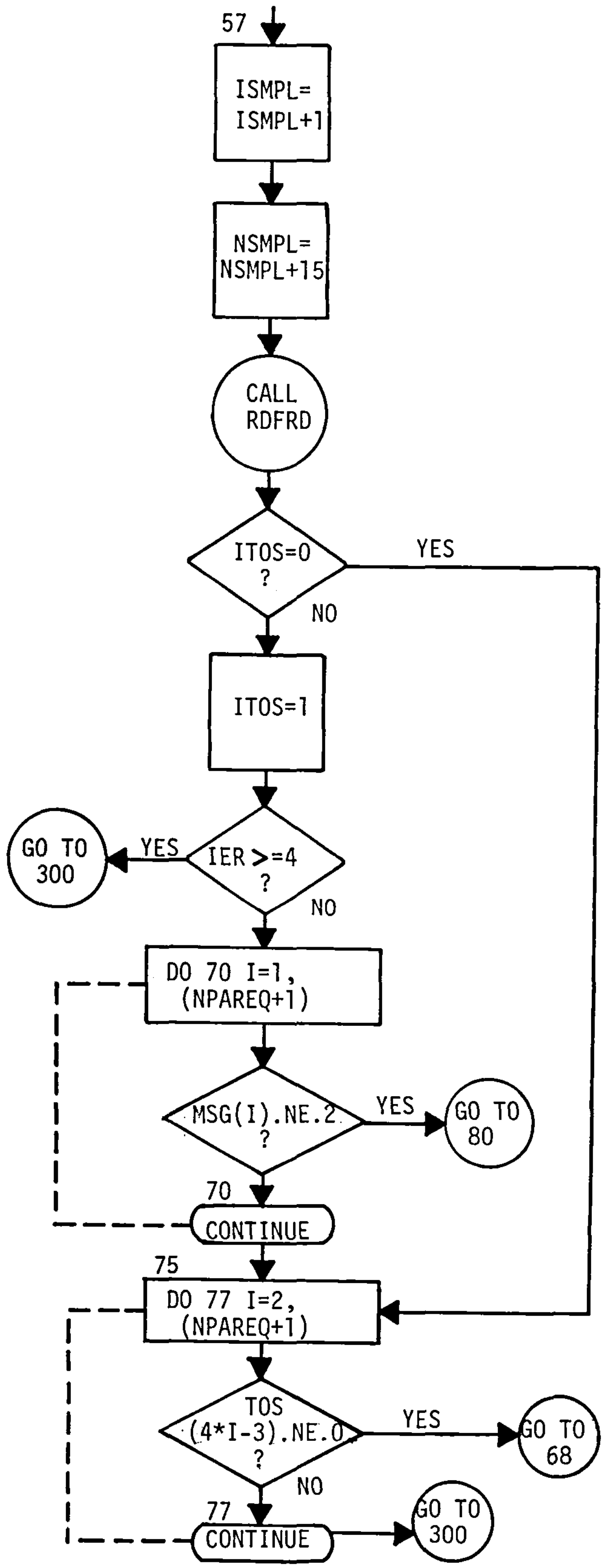

PARNAM(1)='TIME'
PPC(1)=' '
JPASS(M)=0 ?
YES
GO TO 22
NO
DO 19 II=1,101
IADDR(II) =0
19
CONTINUE
DO 31 L=1,NPOUT
DECODE (PAROT(L) 32) SNAM1,SNAM2 (A6,A2)
TABREC(13+L-12)=SNAM1
TABREC(13+L-11)=SNAM2
DECODE(PPCOT(L), 33)PPCOTT (A1)
TABREC(13+L-10)=PPCOTT
" (13+L-9)=0
" (13+L-8)=L
" (13+L-7)=CTYPE
" (13+L-6)=1.0
" (13+L-5)=0.
TABREC(13+L-4)=0
" (13+L-3)=1.0
" (13+L-2)=0.0
" (13+L-1)=0
" (13+L)=1
31
CONTINUE

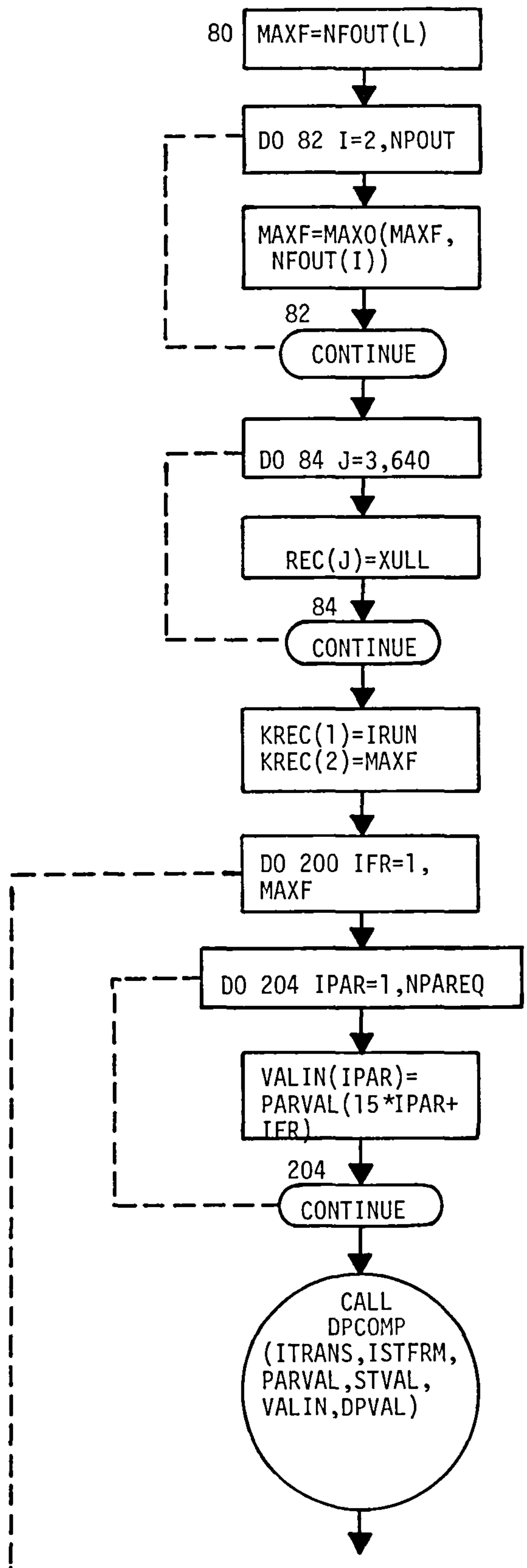

WRITE
(07'(KPASS
(M)+21))
TABREC
DISC
22
WRITE (8,1001)
JPASS(M)
(2X'NO DPTASK
TO DO FOR PASS
NO.,J2/)
RETURN
25
NOPF=0
YES
NO
DO 27 II=1, NOPF
I=II
OPPVAL(I)=
THEAD(IOPPLC(I)
27
CONTINUE
30
IRUN=0
ITOST=0
ITOS=0
35
IRUN=
IRUN+1
DO 37 I=1,400
STATBF(I)=XULL
37
CONTINUE

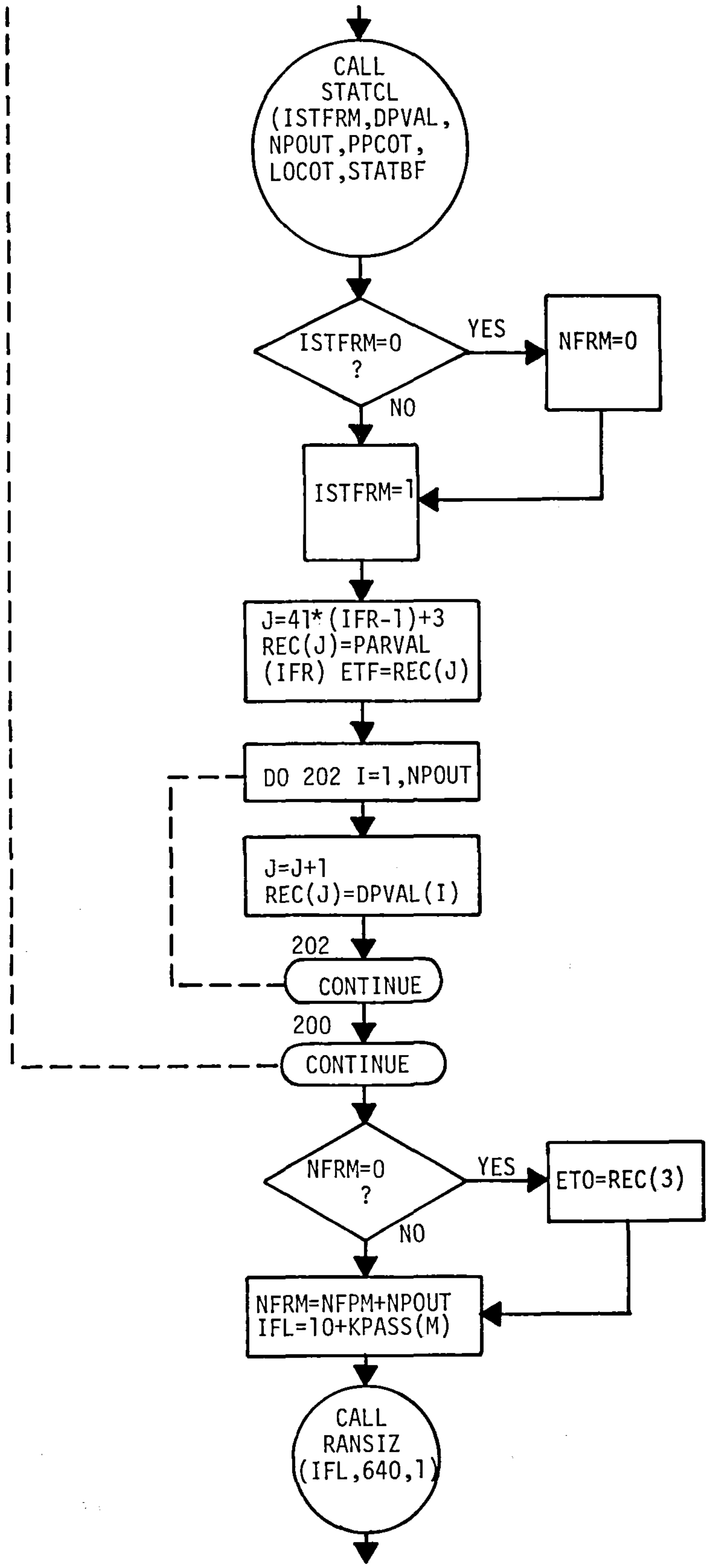

ENCODE(ETYPE, 2000)TEVENT(6'*(IRUN-1)+5) (A6)
DECODE(ETYPE, 2003)ETYPEC (A1)
ETYPEC= 'Z'OR'R'OR'Y' OR'I' ?
YES
GO TO 50
NO
NSTREQ=0
YES
NO
CALL RDFRD
ITOS=1
IER >=4 ?
YES
GO TO 300
NO
DO 40 I=1,NSTREQ
LOC=10+(I-1)+ ISTAT(I) STVAL(I)= OUTBUF(LOC)
40
CONTINUE
60
ETYPEC='S' ?
YES
GO TO 45
NO
ITRANS =1
GO TO 47

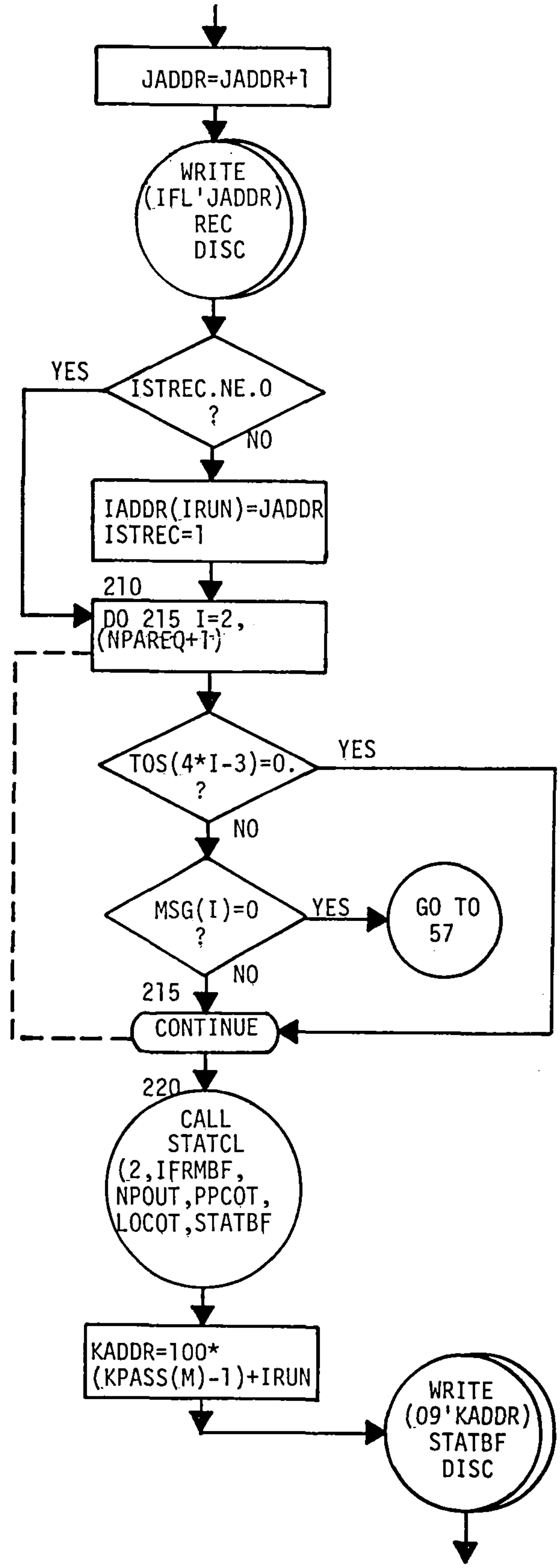

50
NADDR=100*
(KPASS(M)-1)+
IRUN
WRITE
(09'NADDR)
STATBF
DISC
YES
LINCNT < 42
?
NO
WRITE(6,1002)
JPASS(M), SR(1)
('DERIVED PARA-
METER FOR PASS =
;I3,'DATA FILE
SAMPLE RATE OF;
A6)
WRITE (6,1003)
(PAGE HEADINGS)
LINCNT
=5
51
WRITE(6,1004)
IRUN,ETYPEC
(1H,3X,I3,9X,
A1)
GO TO
35
LINCNT=
LINCNT+
1
45
ITRANS=
0

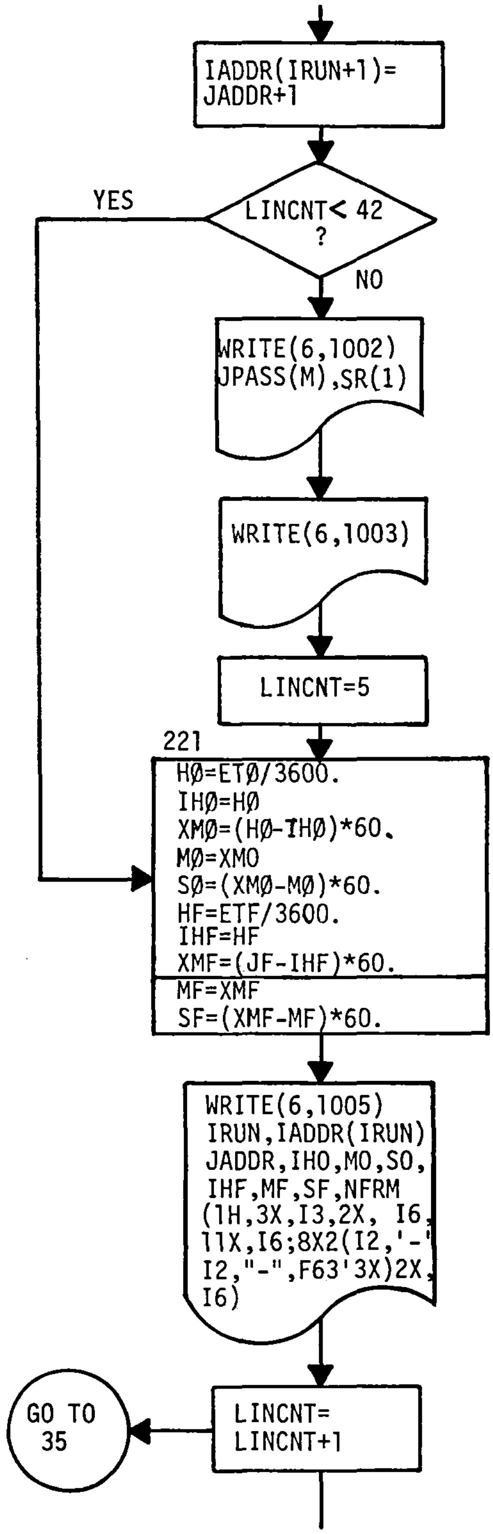

NPAREQ >0
?
YES
NO
CALL
DPCOMP
(ITRANS,0,
OPFVAL,STVAL,
PARVAL,DPVAL
DO 100 I=1,NPOUT
STATBF(10*I-5)=
DPVAL(I)
STATBF(10*I-2)=1.0
STATBF(10*I)=0.0
DPVAL(I)=XULL
?
YES
NO
STATBF(10*I-4)=0.0
STATBF(10*I-1)=0.0
102
STATBF(10*I-1)=1.
0
100
CONTINUE
105
MADDR=100*(KPASS
(M)-1) + IRUN
WRITE
(09'MADDR)
STATBF
DISC
GO TO
35
55
ISTREC=0 ISMPL=0
ISTFRM=0 NSMPL=0

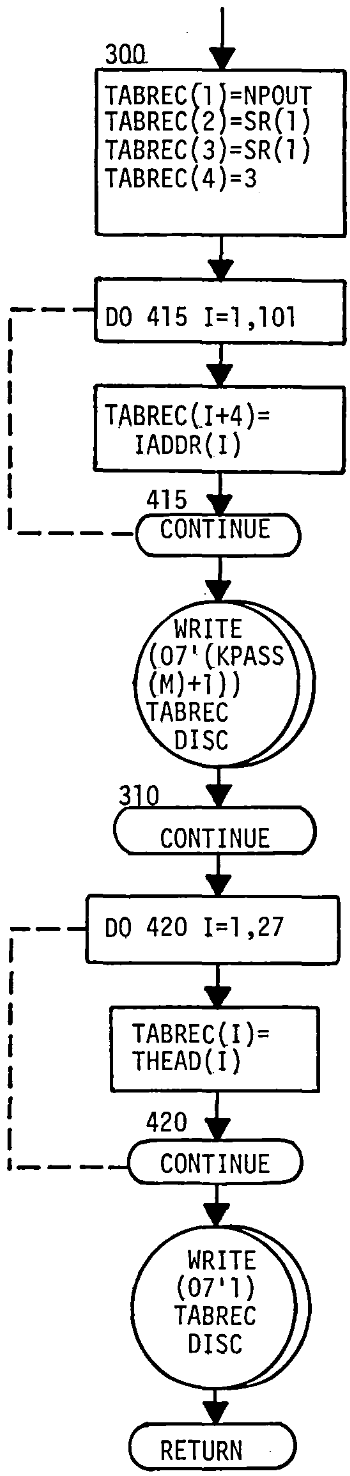

57
ISMPL=
ISMPL+1
NSMPL=
NSMPL+15
CALL
RDFRD
ITOS=0
?
YES
NO
ITOS=1
GO TO
300
YES
IER >=4
?
NO
DO 70 I=1,
(NPAREQ+1)
MSG(I).NE.2
?
YES
GO TO
80
70
CONTINUE
75
DO 77 I=2,
(NPAREQ+1)
TOS
(4*I-3).NE.0
?
YES
GO TO
68
NO
77
CONTINUE
GO TO
300


80
MAXF=NFOUT(L)
DO 82 I=2,NPOUT
MAXF=MAXO(MAXF,
NFOUT(I))
82
CONTINUE
DO 84 J=3,640
REC(J)=XULL
84
CONTINUE
KREC(1)=IRUN
KREC(2)=MAXF
DO 200 IFR=1,
MAXF
DO 204 IPAR=1,NPAREQ
VALIN(IPAR)=
PARVAL(15*IPAR+
IFR)
204
CONTINUE
CALL
DPCOMP
(ITRANS,ISTFRM,
PARVAL,STVAL,
VALIN,DPVAL)


CALL STATCL (ISTFRM,DPVAL, NPOUT,PPCOT, LOCOT,STATBF
ISTFRM=0 ?
YES
NFRM=0
NO
ISTFRM=1
J=41*(IFR-1)+3
REC(J)=PARVAL (IFR) ETF=REC(J)
DO 202 I=1,NPOUT
J=J+1
REC(J)=DPVAL(I)
202
CONTINUE
200
CONTINUE
NFRM=0 ?
YES
ETO=REC(3)
NO
NFRM=NFPM+NPOUT
IFL=10+KPASS(M)
CALL RANSIZ (IFL,640,1)


JADDR=JADDR+1
WRITE (IFL'JADDR) REC DISC
ISTREC.NE.0 ?
YES
NO
IADDR(IRUN)=JADDR
ISTREC=1
210
DO 215 I=2, (NPAREQ+1)
TOS(4*I-3)=0. ?
YES
NO
MSG(I)=0 ?
YES
GO TO 57
NO
215
CONTINUE
220
CALL STATCL (2,IFRMBF, NPOUT,PPCOT, LOCOT,STATBF
KADDR=100* (KPASS(M)-1)+IRUN
WRITE (09'KADDR) STATBF DISC


IADDR(IRUN+1)=
JADDR+1
YES
LINCNT< 42
?
NO
WRITE(6,1002)
JPASS(M),SR(1)
WRITE(6,1003)
LINCNT=5
221
HØ=ETØ/3600.
IHØ=HØ
XMØ=(HØ-IHØ)*60.
MØ=XMO
SØ=(XMØ-MØ)*60.
HF=ETF/3600.
IHF=HF
XMF=(JF-IHF)*60.
MF=XMF
SF=(XMF-MF)*60.
WRITE(6,1005)
IRUN,IADDR(IRUN)
JADDR,IHO,MO,SO,
IHF,MF,SF,NFRM
(1H,3X,I3,2X, I6,
11X,I6;8X2(I2,'-'
I2,"-",F63'3X)2X,
I6)
GO TO
35
LINCNT=
LINCNT+1


300
TABREC(1)=NPOUT
TABREC(2)=SR(1)
TABREC(3)=SR(1)
TABREC(4)=3
DO 415 I=1,101
TABREC(I+4)=
IADDR(I)
415
CONTINUE
WRITE
(07'(KPASS
(M)+1))
TABREC
DISC
310
CONTINUE
DO 420 I=1,27
TABREC(I)=
THEAD(I)
420
CONTINUE
WRITE
(07'1)
TABREC
DISC
RETURN

DPDE   SUBROUTINE DPDEF

```
      SUBROUTINE DPDEF(IPASS,NOPF,IOPFLC,NSTREQ,ISTAT,STPAR,STPPC,STSR,
     1 NPAREQ,SR,PARNAM,PPC,NPOUT,PAROT,PPCOT)
********SUBROUTINE DPDEF--ENTRY POINT DPCOMP*****************************
```

PROGRAM IDENTIFICATION

PROGRAM NAME ----- DPDEF (ENTRY DPCOMP)
PROGRAM NUMBER ----- 112338
AUTHOR ----- TERRY D. SOMMERS

COMPUTER ----- HW625/635
MEMORY -----
PERIPHERALS -----
LANGUAGE ----- HW6000 FORTRAN/FORTY

PURPOSE

THIS IS AN INITIALIZATION CALL TO BE MADE ONCE PRIOR TO BEGINNING EACH DP PASS. ITS PURPOSE IS TO DEFINE FOR THE DP COMPUTE ROUTINE WHAT PASS IS TO BE PROCESSED AND IN RETURN IT WILL RETURN TO THE CALLING PROGRAM A LIST OF SPECIFICATIONS FOR THE PARAMETERS THAT IT WILL BE NEEDING AND FOR THE PARAMETERS IT WILL BE GENERATING.

THE ENTRY TO DPCOMP IS MADE FOR EACH FRAME OF DATA.

METHOD

ASSUMPTIONS

ONCE PER FLIGHT VALUES REQUIRED

ETP - LOCATION 16
ESGW - LOCATION 17
ESCG - LOCATION 18
KPR - LOCATION 19
HPC - LOCATION 20
MRC - LOCATION 21
R - LOCATION 22
RT - LOCATION 23
GT - LOCATION 24
GTS - LOCATION 25
RELHU - LOCATION 26
FHBOOM - LOCATION 27

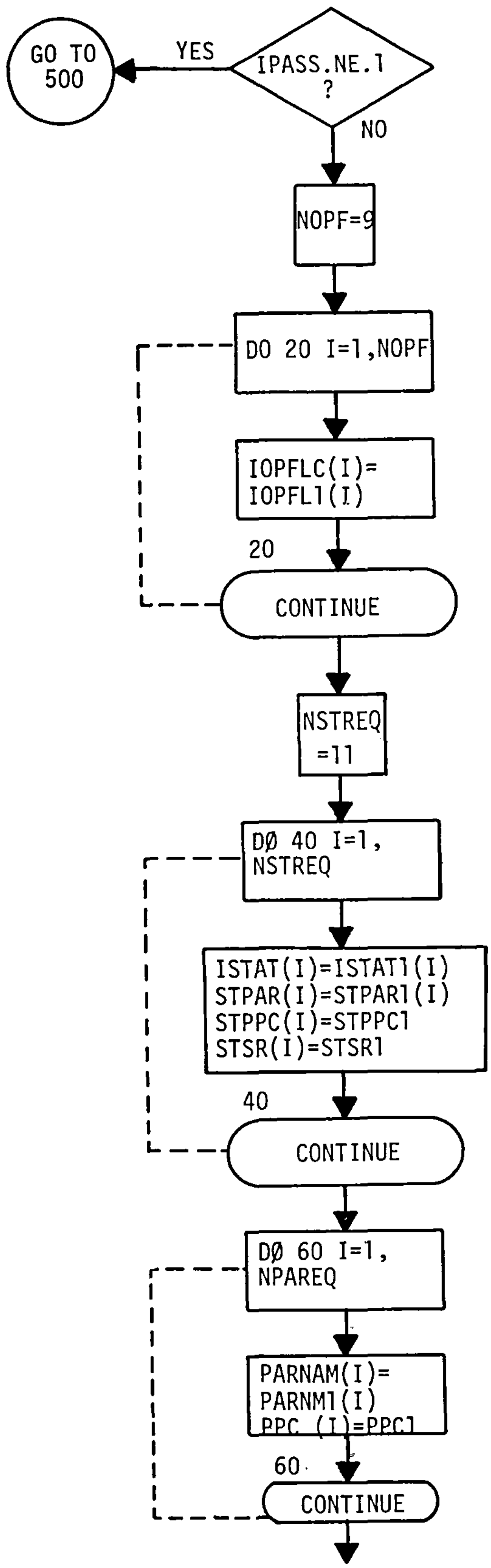
STATISTICS REQUIRED

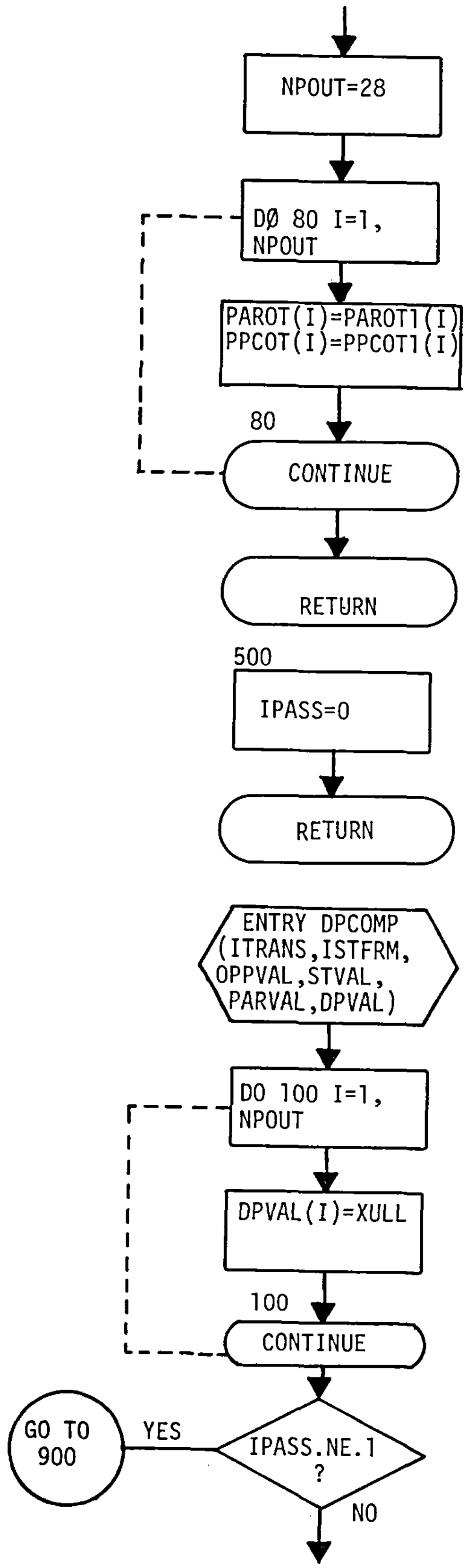
ITATBOOM	VIPBOOM	HBOOM	NO1QPCT
NO2QPCT	NO1NFPCT	NO2NFPCT	NR
MRQ1	TRQ	LOADFACT	

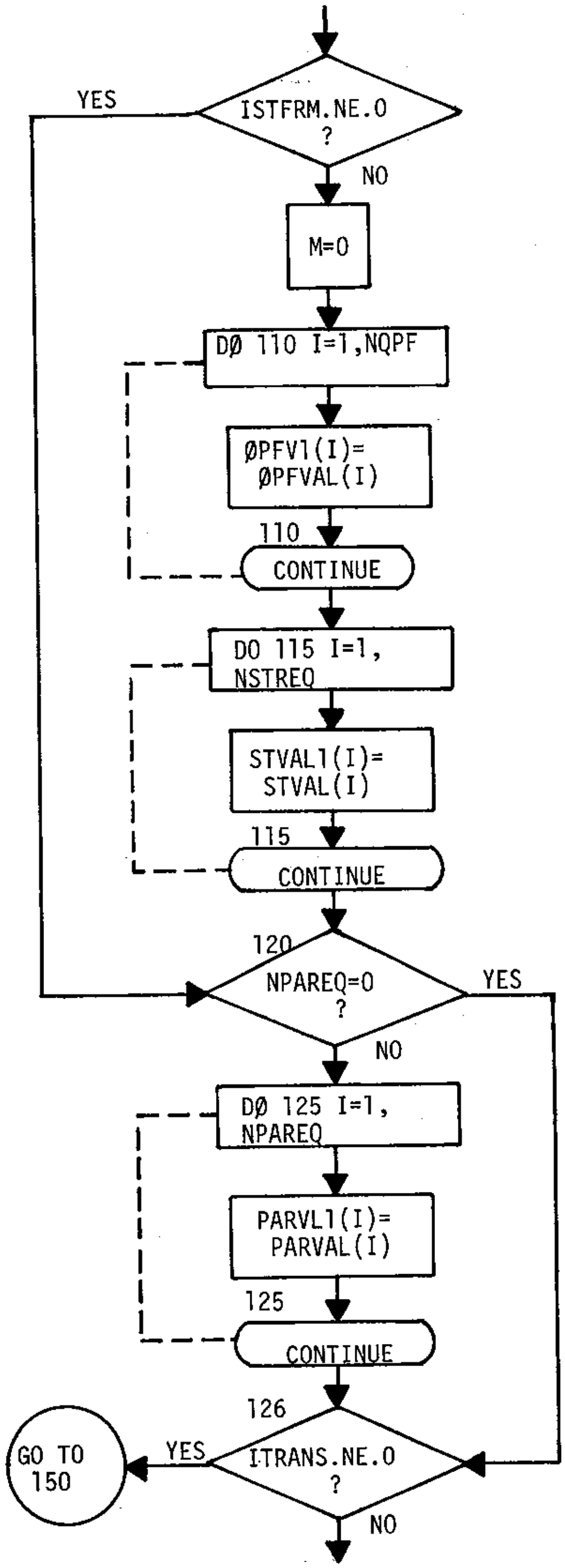
MEASUREMENTS REQUIRED AS INPUT ON A FRAME BASIS

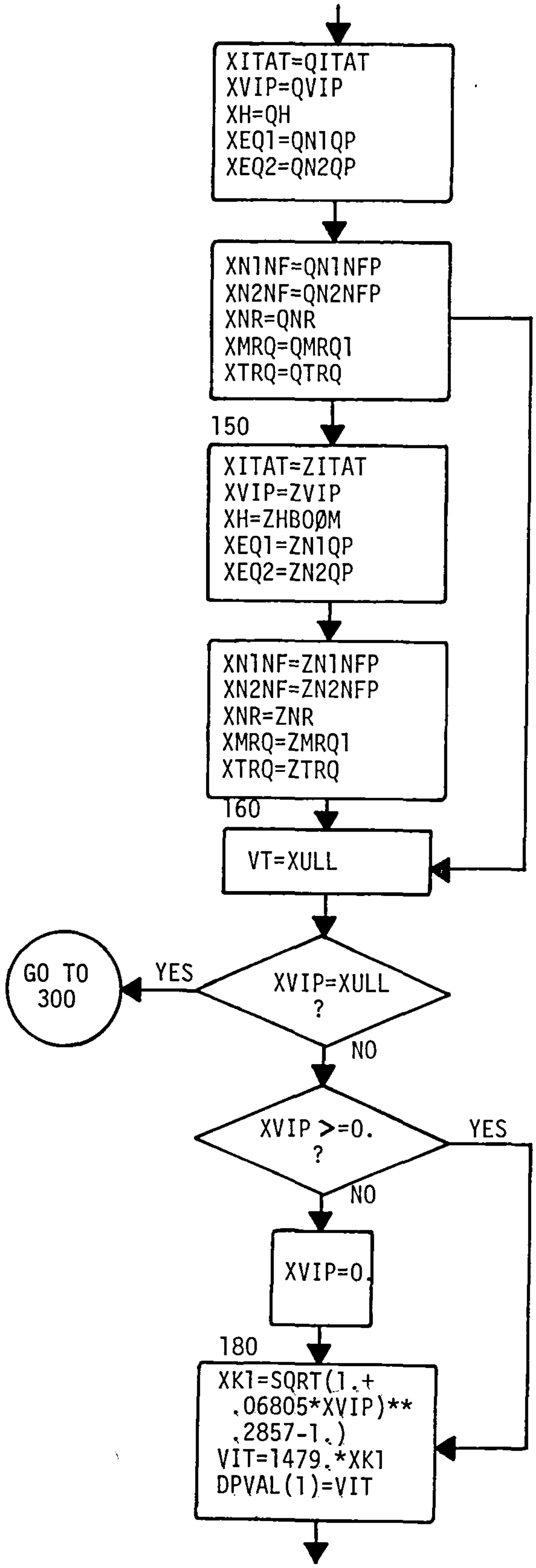
ITATBOOM	VIPBOOM	HBOOM	HEAD180	NO1QPCT
NO2QPCT	NO1NFPCT	NO2NFPCT	LOADFACT	MRQ1
TRQ	NR	TRIMPIT	MRLIFTA	MRLIFTB
MRLIFTC	MRLIFTD	ISAFAPR	ISOFAPR	ISAFBPR
XMSNSGT	AILPOSR			

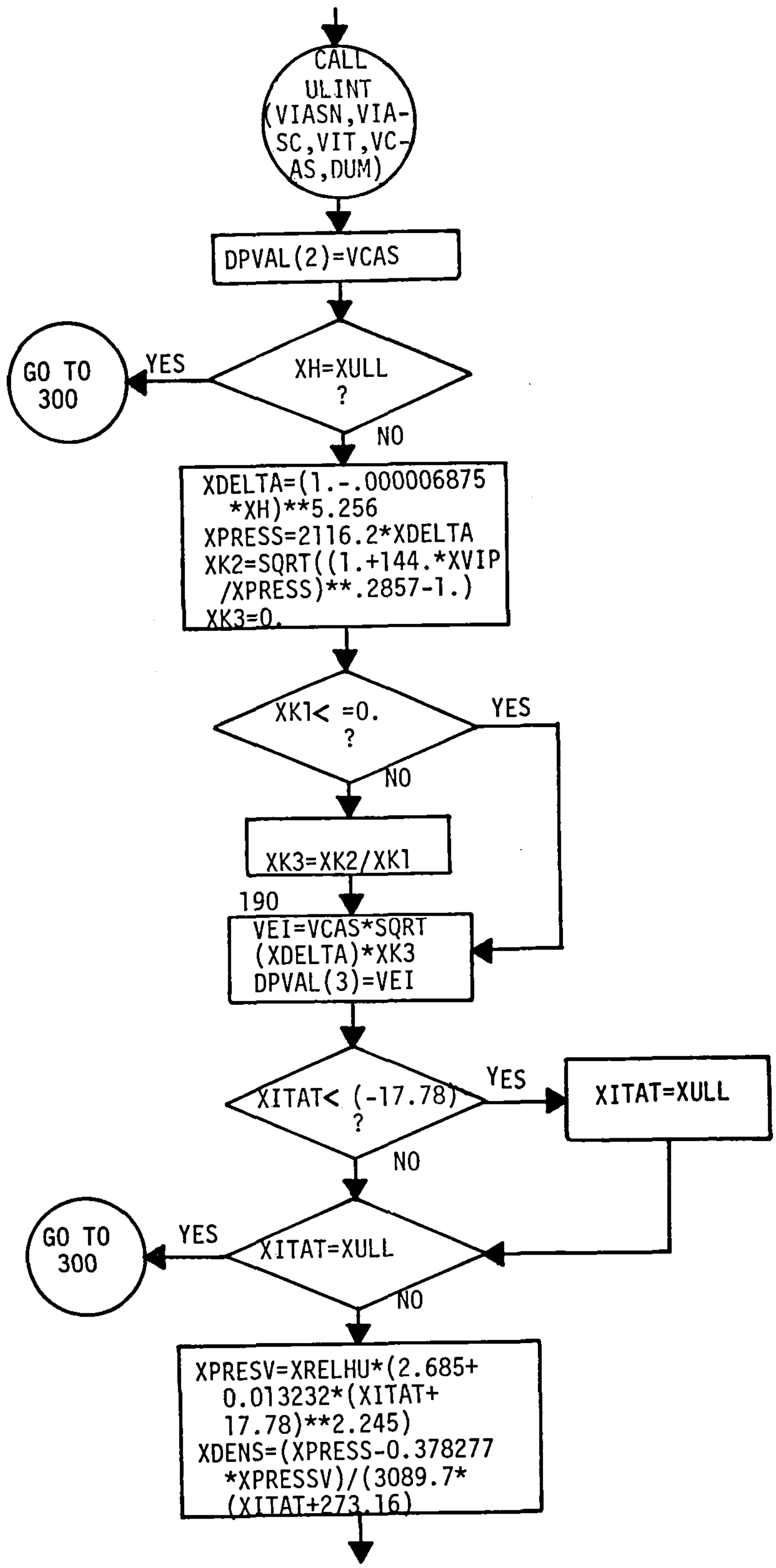
```
      PARAMETERS GENERATED FOR OUTPUT TO RDFRD
             VITBOOM         VCASBOOM  VEIBOOM   FATBOOM   VTBOOM
             TATBOOM         HDBOOM    OMEGAR    OMEGATR   MU
             MACHA           NO1SHP    NO2SHP    HPT       CPTE
             HPMR            CPM       HPTR      CPTAIL    HPS
             CPT             YAWFTRM   FX        FY        FZ
             MX              MY        MZ        AILPOSP

      VARIABLE NAMING CONVENTION

        FIRST LETTER X = TEMPORARY VARIABLE GENERATED WITHIN
                         DPCOMP BUT WHICH ARE NOT RETURNED TO
                         THE CONTROL PROGRAM AND ONCE PER FLIGHT
                         VALUES FED IN FROM THE HEADER TABLE
        FIRST LETTER Q = VARIABLES CONTAINING STATISTICS VALUES
                         FED IN FROM THE CONTROL PROGRAM
        FIRST LETTER Z = VARIABLES CONTAINING SENSOR VALUES FED IN
                         FROM THE CONTROL PROGRAM ON A FRAME BASIS

 INPUT/OUTPUT

      SEE CALLING ROUTINE FOR ARGUMENT DEFINITIONS

 CALLING SEQUENCE

    CALL ULINT(VIASN,VIASC ,VIT,VCAS,DUM)

 COMMON AREAS

     SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS ,
    FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION)

*********************************************************************
```

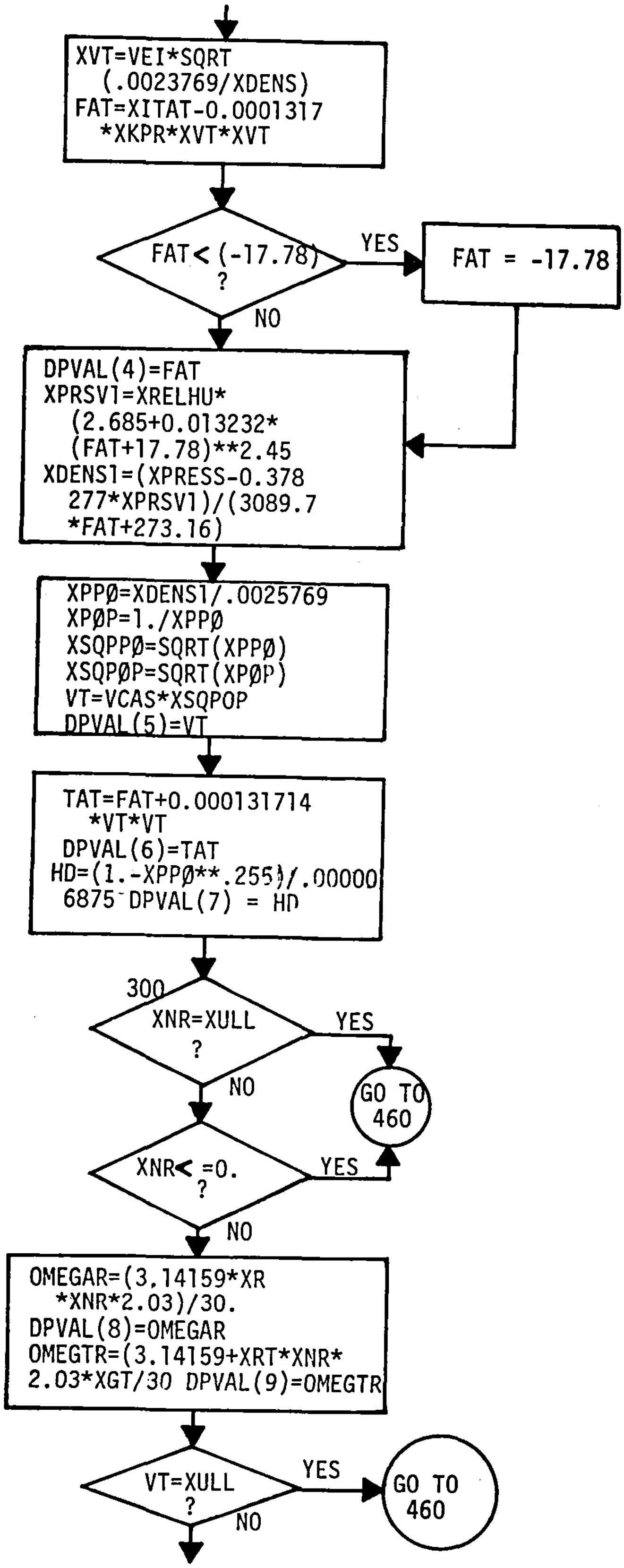

GO TO 500
YES
IPASS.NE.1 ?
NO
NOPF=9
DO 20 I=1,NOPF
IOPFLC(I)= IOPFL1(I)
20
CONTINUE
NSTREQ =11
DØ 40 I=1, NSTREQ
ISTAT(I)=ISTAT1(I)
STPAR(I)=STPAR1(I)
STPPC(I)=STPPC1
STSR(I)=STSR1
40
CONTINUE
DØ 60 I=1, NPAREQ
PARNAM(I)=
PARNM1(I)
PPC (I)=PPC1
60
CONTINUE

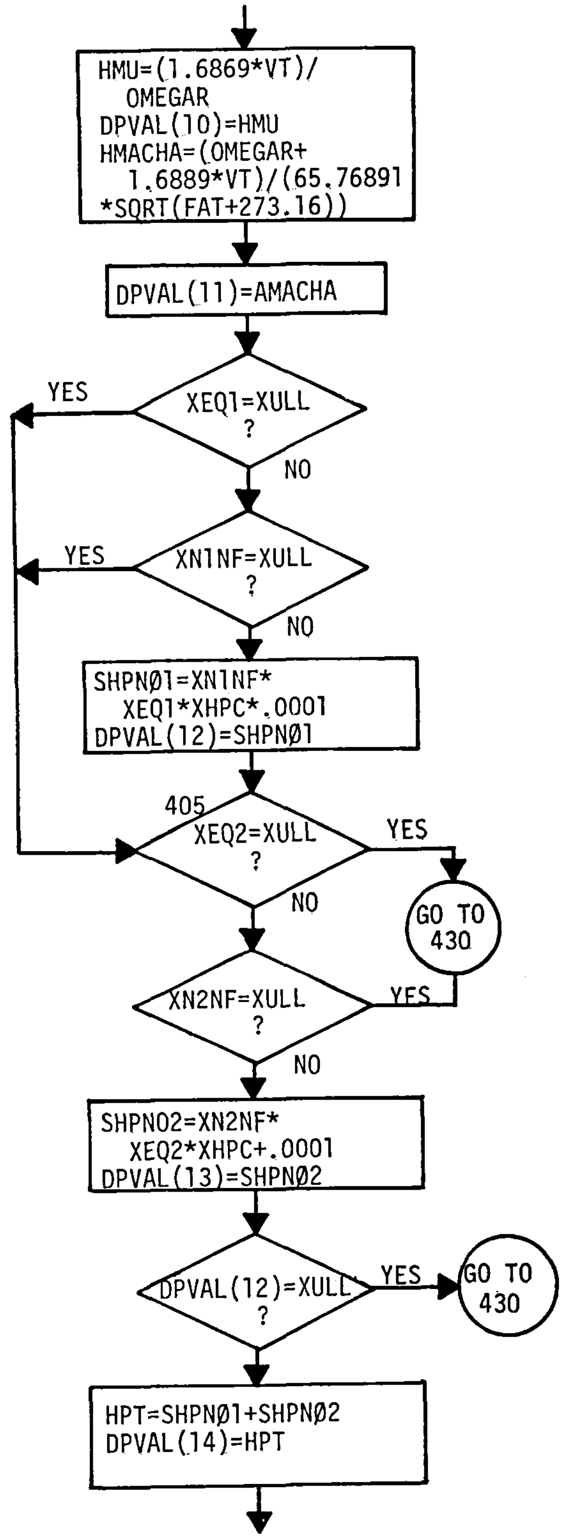

NPOUT=28
DØ 80 I=1,
NPOUT
PAROT(I)=PAROT1(I)
PPCOT(I)=PPCOT1(I)
80
CONTINUE
RETURN
500
IPASS=0
RETURN
ENTRY DPCOMP
(ITRANS,ISTFRM,
OPPVAL,STVAL,
PARVAL,DPVAL)
DO 100 I=1,
NPOUT
DPVAL(I)=XULL
100
CONTINUE
GO TO
900
YES
IPASS.NE.1
?
NO

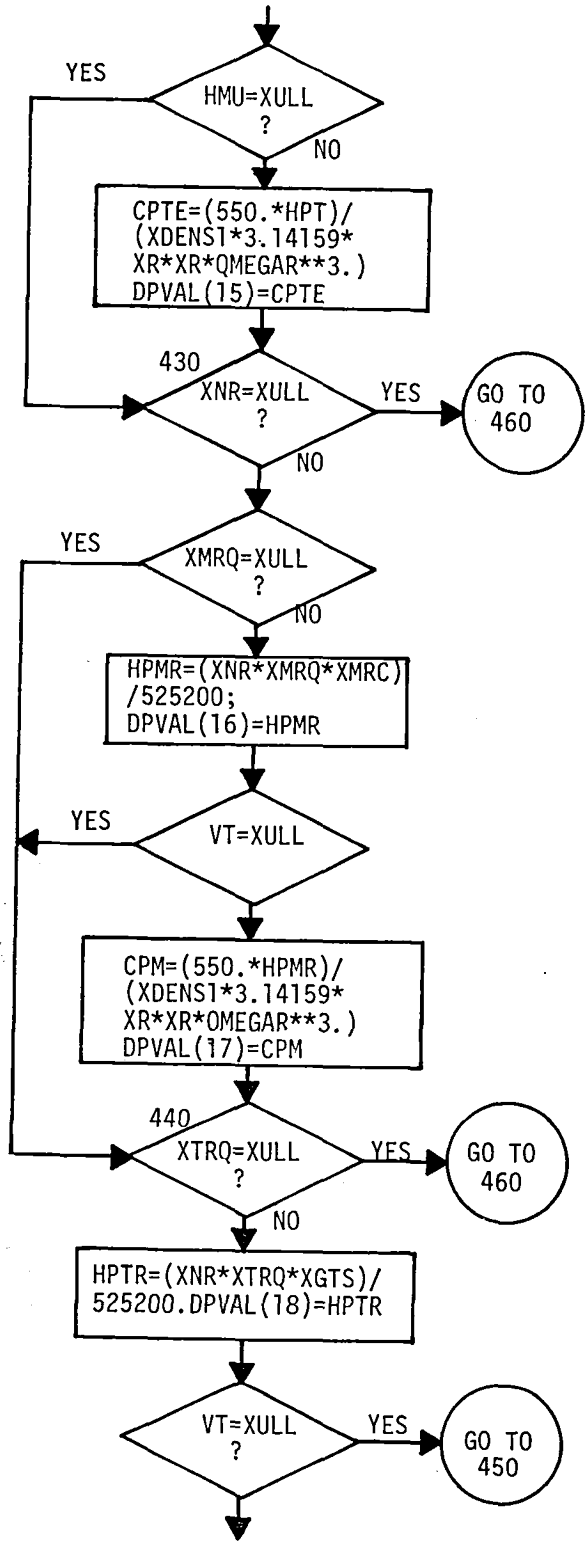

ISTFRM.NE.0
?
YES
NO
M=0
DØ 110 I=1,NQPF
ØPFV1(I)=
ØPFVAL(I)
110
CONTINUE
DO 115 I=1,
NSTREQ
STVAL1(I)=
STVAL(I)
115
CONTINUE
120
NPAREQ=0
?
YES
NO
DØ 125 I=1,
NPAREQ
PARVL1(I)=
PARVAL(I)
125
CONTINUE
126
ITRANS.NE.0
?
YES
NO
GO TO
150

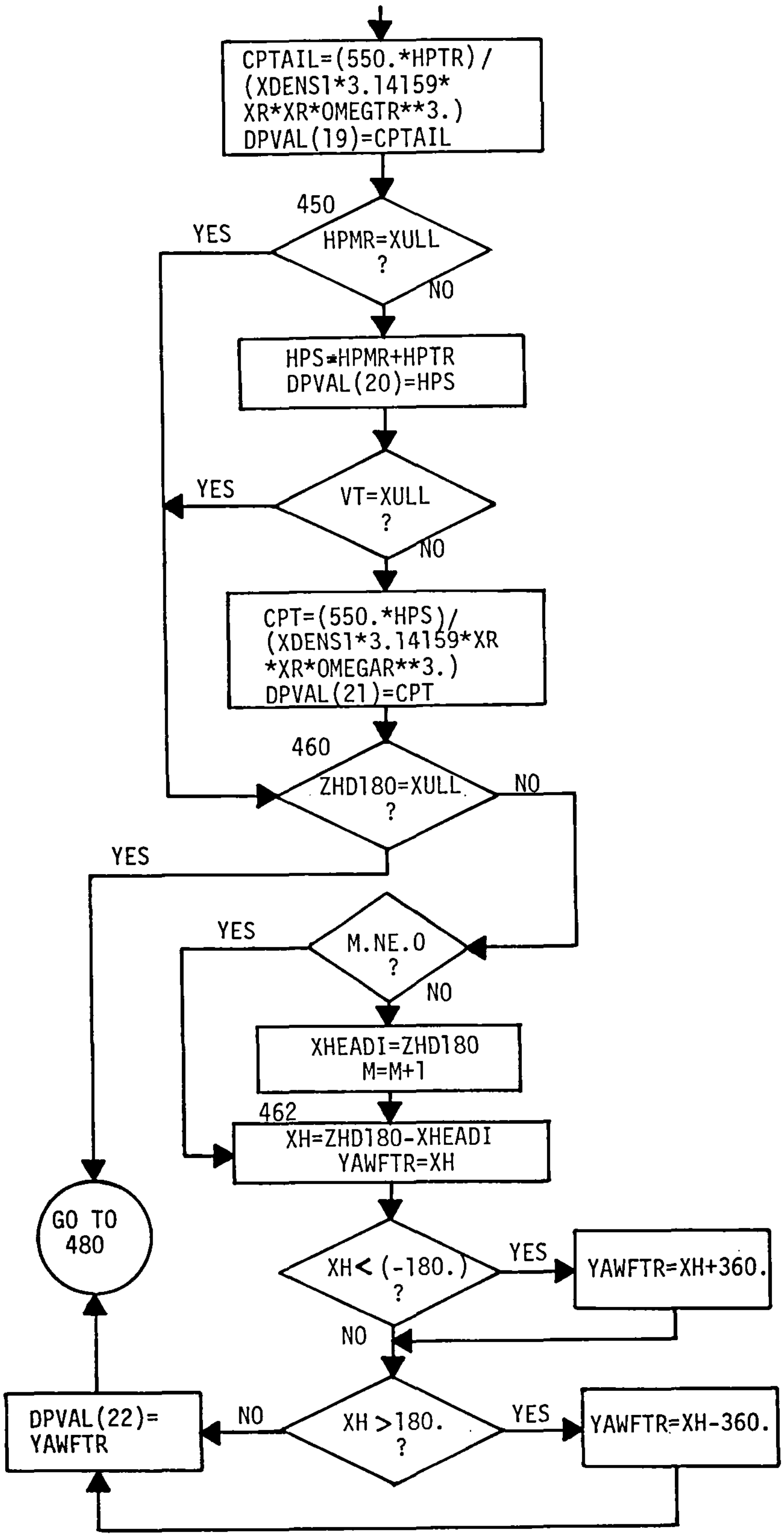

XITAT=QITAT
XVIP=QVIP
XH=QH
XEQ1=QN1QP
XEQ2=QN2QP
XN1NF=QN1NFP
XN2NF=QN2NFP
XNR=QNR
XMRQ=QMRQ1
XTRQ=QTRQ
150
XITAT=ZITAT
XVIP=ZVIP
XH=ZHBOØM
XEQ1=ZN1QP
XEQ2=ZN2QP
XN1NF=ZN1NFP
XN2NF=ZN2NFP
XNR=ZNR
XMRQ=ZMRQ1
XTRQ=ZTRQ
160
VT=XULL
GO TO 300
YES
XVIP=XULL ?
NO
XVIP >=0. ?
YES
NO
XVIP=0.
180
XK1=SQRT(1.+ .06805*XVIP)** .2857-1.)
VIT=1479.*XK1
DPVAL(1)=VIT

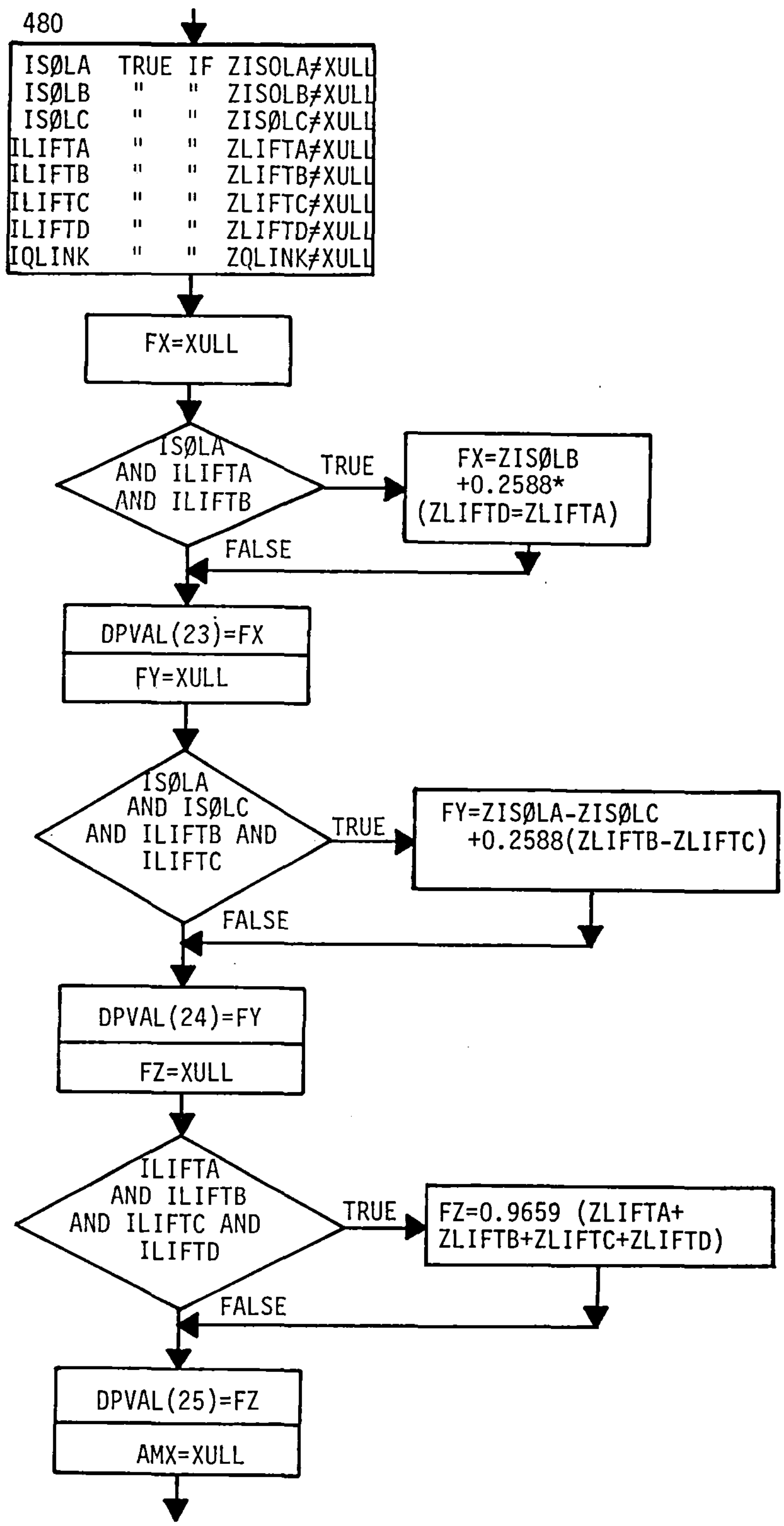

CALL ULINT (VIASN,VIA-SC,VIT,VC-AS,DUM)
DPVAL(2)=VCAS
XH=XULL ?
YES
GO TO 300
NO
XDELTA=(1.-.000006875 *XH)**5.256
XPRESS=2116.2*XDELTA
XK2=SQRT((1.+144.*XVIP /XPRESS)**.2857-1.)
XK3=0.
XK1< =0. ?
YES
NO
XK3=XK2/XK1
190
VEI=VCAS*SQRT (XDELTA)*XK3
DPVAL(3)=VEI
XITAT< (-17.78) ?
YES
XITAT=XULL
NO
XITAT=XULL
YES
GO TO 300
NO
XPRESV=XRELHU*(2.685+ 0.013232*(XITAT+ 17.78)**2.245)
XDENS=(XPRESS-0.378277 *XPRESSV)/(3089.7* (XITAT+273.16)

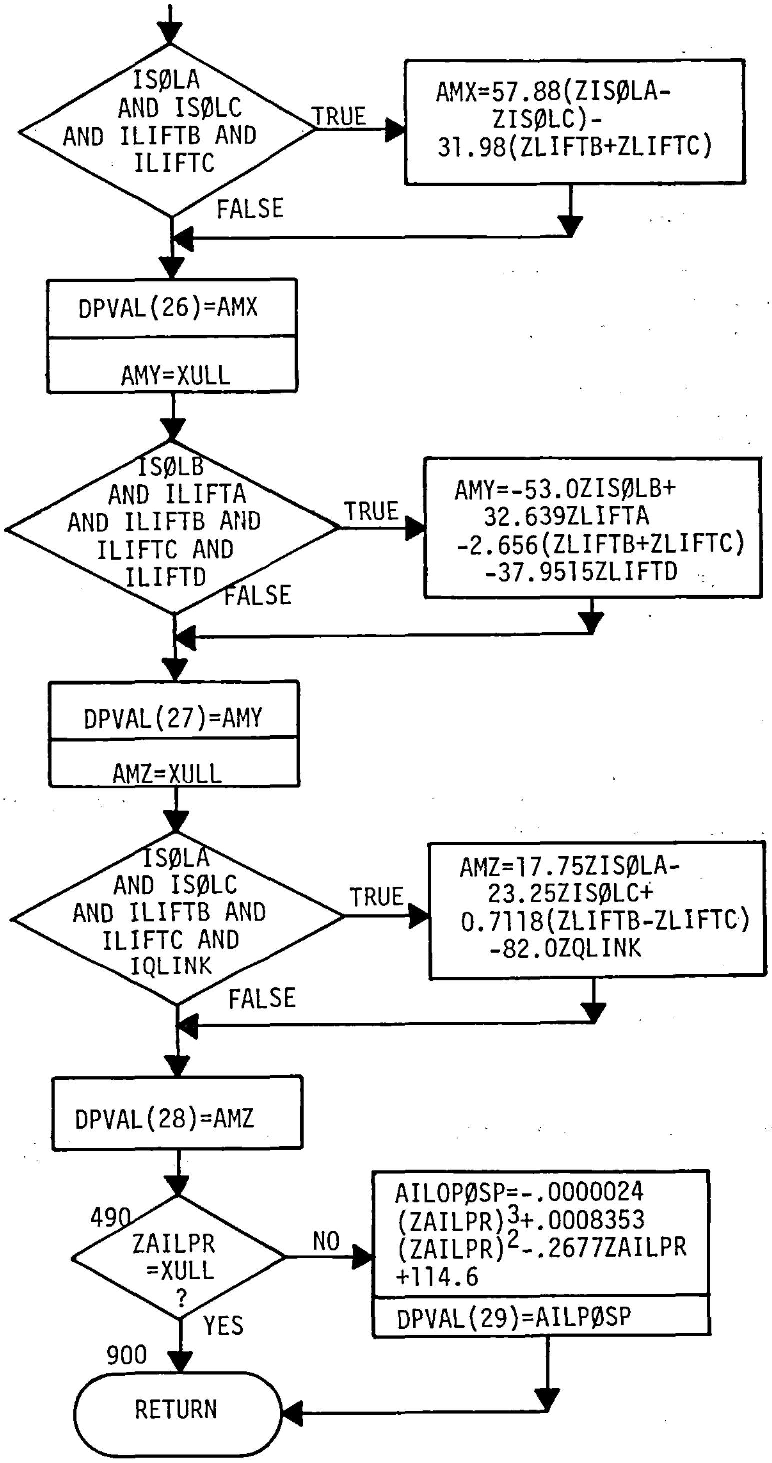

XVT=VEI*SQRT (.0023769/XDENS)
FAT=XITAT-0.0001317 *XKPR*XVT*XVT
FAT< (-17.78) ?
YES
FAT = -17.78
NO
DPVAL(4)=FAT
XPRSV1=XRELHU* (2.685+0.013232* (FAT+17.78)**2.45
XDENS1=(XPRESS-0.378 277*XPRSV1)/(3089.7 *FAT+273.16)
XPPØ=XDENS1/.0025769
XPØP=1./XPPØ
XSQPPØ=SQRT(XPPØ)
XSQPØP=SQRT(XPØP)
VT=VCAS*XSQPOP
DPVAL(5)=VT
TAT=FAT+0.000131714 *VT*VT
DPVAL(6)=TAT
HD=(1.-XPPØ**.255)/.00000 6875 DPVAL(7) = HD
300
XNR=XULL ?
YES
NO
GO TO 460
XNR< =0. ?
YES
NO
OMEGAR=(3.14159*XR *XNR*2.03)/30.
DPVAL(8)=OMEGAR
OMEGTR=(3.14159+XRT*XNR* 2.03*XGT/30 DPVAL(9)=OMEGTR
VT=XULL ?
YES
NO
GO TO 460


HMU=(1.6869*VT)/
OMEGAR
DPVAL(10)=HMU
HMACHA=(OMEGAR+
1.6889*VT)/(65.76891
*SQRT(FAT+273.16))
DPVAL(11)=AMACHA
XEQ1=XULL
?
YES
NO
XN1NF=XULL
?
YES
NO
SHPNØ1=XN1NF*
XEQ1*XHPC*.0001
DPVAL(12)=SHPNØ1
405
XEQ2=XULL
?
YES
NO
GO TO
430
XN2NF=XULL
?
YES
NO
SHPNO2=XN2NF*
XEQ2*XHPC+.0001
DPVAL(13)=SHPNØ2
DPVAL(12)=XULL
?
YES
GO TO
430
HPT=SHPNØ1+SHPNØ2
DPVAL(14)=HPT

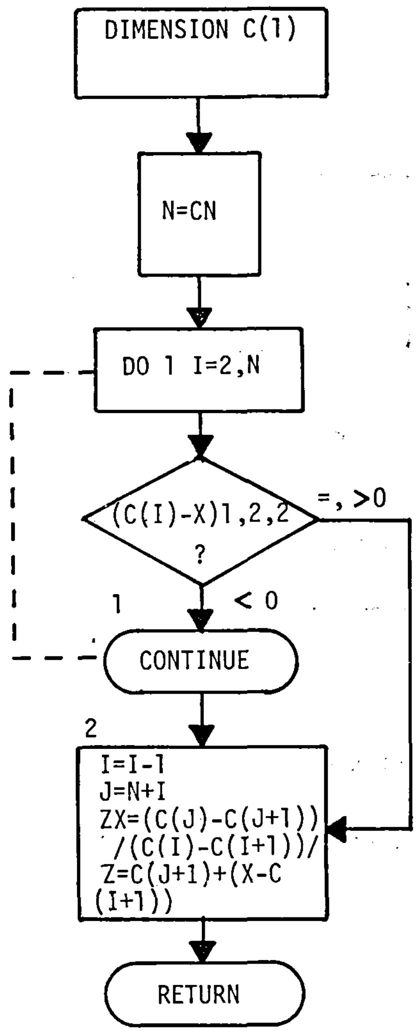

HMU=XULL ?
YES
NO
CPTE=(550.*HPT)/
(XDENS1*3.14159*
XR*XR*QMEGAR**3.)
DPVAL(15)=CPTE
430
XNR=XULL ?
YES
GO TO 460
NO
XMRQ=XULL ?
YES
NO
HPMR=(XNR*XMRQ*XMRC)
/525200;
DPVAL(16)=HPMR
VT=XULL
YES
CPM=(550.*HPMR)/
(XDENS1*3.14159*
XR*XR*OMEGAR**3.)
DPVAL(17)=CPM
440
XTRQ=XULL ?
YES
GO TO 460
NO
HPTR=(XNR*XTRQ*XGTS)/
525200.DPVAL(18)=HPTR
VT=XULL ?
YES
GO TO 450

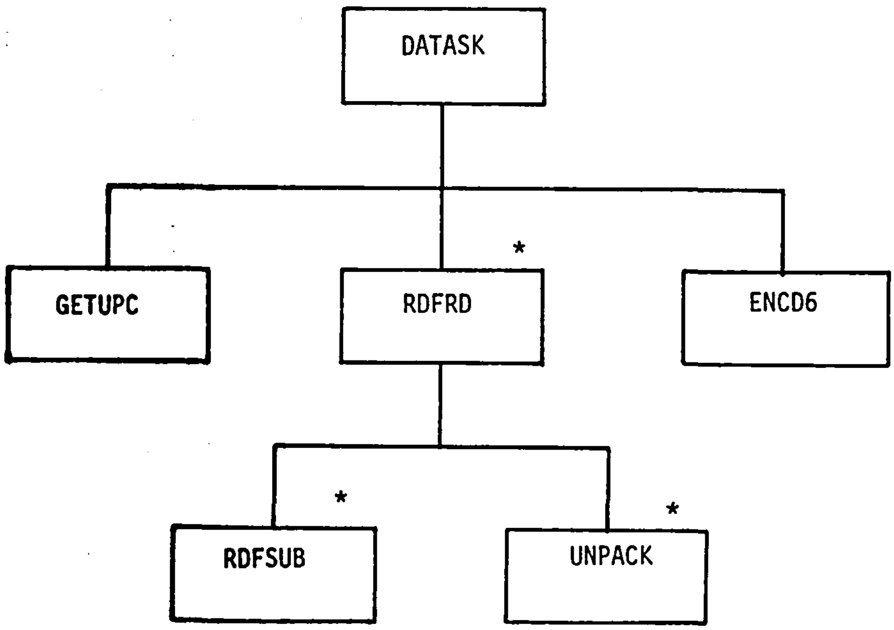

CPTAIL=(550.*HPTR)/
(XDENS1*3.14159*
XR*XR*OMEGTR**3.)
DPVAL(19)=CPTAIL
450
HPMR=XULL
?
YES
NO
HPS=HPMR+HPTR
DPVAL(20)=HPS
VT=XULL
?
YES
NO
CPT=(550.*HPS)/
(XDENS1*3.14159*XR
*XR*OMEGAR**3.)
DPVAL(21)=CPT
460
ZHD180=XULL
?
NO
YES
M.NE.0
?
YES
NO
XHEADI=ZHD180
M=M+1
462
XH=ZHD180-XHEADI
YAWFTR=XH
GO TO
480
XH< (-180.)
?
YES
YAWFTR=XH+360.
NO
XH >180.
?
YES
YAWFTR=XH-360.
NO
DPVAL(22)=
YAWFTR


480
ISØLA TRUE IF ZISOLA≠XULL
ISØLB " " ZISOLB≠XULL
ISØLC " " ZISØLC≠XULL
ILIFTA " " ZLIFTA≠XULL
ILIFTB " " ZLIFTB≠XULL
ILIFTC " " ZLIFTC≠XULL
ILIFTD " " ZLIFTD≠XULL
IQLINK " " ZQLINK≠XULL
FX=XULL
ISØLA AND ILIFTA AND ILIFTB
TRUE
FX=ZISØLB +0.2588* (ZLIFTD=ZLIFTA)
FALSE
DPVAL(23)=FX
FY=XULL
ISØLA AND ISØLC AND ILIFTB AND ILIFTC
TRUE
FY=ZISØLA-ZISØLC +0.2588(ZLIFTB-ZLIFTC)
FALSE
DPVAL(24)=FY
FZ=XULL
ILIFTA AND ILIFTB AND ILIFTC AND ILIFTD
TRUE
FZ=0.9659 (ZLIFTA+ ZLIFTB+ZLIFTC+ZLIFTD)
FALSE
DPVAL(25)=FZ
AMX=XULL


ISØLA AND ISØLC AND ILIFTB AND ILIFTC
TRUE
AMX=57.88(ZISØLA-ZISØLC)-31.98(ZLIFTB+ZLIFTC)
FALSE
DPVAL(26)=AMX
AMY=XULL
ISØLB AND ILIFTA AND ILIFTB AND ILIFTC AND ILIFTD
TRUE
AMY=-53.0ZISØLB+32.639ZLIFTA -2.656(ZLIFTB+ZLIFTC) -37.9515ZLIFTD
FALSE
DPVAL(27)=AMY
AMZ=XULL
ISØLA AND ISØLC AND ILIFTB AND ILIFTC AND IQLINK
TRUE
AMZ=17.75ZISØLA-23.25ZISØLC+0.7118(ZLIFTB-ZLIFTC) -82.0ZQLINK
FALSE
DPVAL(28)=AMZ
490
ZAILPR =XULL ?
NO
AILOPØSP=-.0000024 (ZAILPR)3+.0008353 (ZAILPR)2-.2677ZAILPR +114.6
DPVAL(29)=AILPØSP
YES
900
RETURN

```
ULIN     SUBROUTINE ULINT(CN,C,X,Z,ZX)

      SUBROUTINE ULINT(CN,C,X,Z,ZX)

********SUBROUTINE ULINT************************************************

 PROGRAM IDENTIFICATION

      PROGRAM NAME ----- ULINT
      PROGRAM NUMBER ----- 112338
      AUTHOR ----- TERRY D. SOMMERS

      COMPUTER ----- HW625/635
      MEMORY -----
      PERIPHERALS -----
      LANGUAGE ----- HW6000 FORTRAN/FORTY

 PURPOSE

       SUBROUTINE CALLED BY DPCOMP TO PERFORM UNIVARIATE LINEAR
      INTERPOLATION.

 METHOD

 INPUT/OUTPUT

      CN - NO. OF X VALUES IN C
      C - TABLE OF VALUES (CN X"S(MONOTONICALLY INCREASING)
          FOLLOWED BY CN Y"S)
      X - INPUT X VALUES
      Z - RETURNED Z VALUE
      ZX - RETURNED SLOPE

 CALLING SEQUENCE

      CALL ULINT(CN,C,X,Z,ZX)

 COMMON AREAS

        SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS ,
      FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION)
```


DIMENSION C(1)
N=CN
DO 1 I=2,N
(C(I)-X)1,2,2
?
=, >0
1
< 0
CONTINUE
2
I=I-1
J=N+I
ZX=(C(J)-C(J+1))
/(C(I)-C(I+1))/
Z=C(J+1)+(X-C
(I+1))
RETURN



*See Sub-program RDFRD for Comments and Flowcharts

HIERARCHY CHART for DATASK PROGRAM

DATK   SUBROUTINE DATASK

```
     SUBROUTINE DATASK
********DATASK*****************************************************************

 PROGRAM IDENTIFICATION

     PROGRAM NAME ----- DATASK
     PROGRAM NUMBER ----- 112320
     AUTHOR ------ TERRY D SOMMERS

     COMPUTER ------ HW-625/635
     MEMORY ------ ALL FILES OPEN(16K)
     PERIPHERALS ------ CARD READER,DISC,PRINTER
     LANGUAGE ------ HW 6000 FORTRAN/FORTY

 PURPOSE

        TO GENERATE A DATA ANALYSIS REPORT TO BE USED BY
     ENGINEERS FOR DETERMINING FLIGHT CONDITIONS AND
     DATA VALIDITY.

 METHOD

        READ CARDS CONTAINING SENSOR NAME,PRE-PROCESSING CODE
     AND SAMPLE RATE.  READ IN SPECIFIC EVENT UNLESS ALL EVENTS
     ARE TO BE USED. GET CORRESPONDING DATA FROM STATISTICS FILE 09
     AND OUTPUT VALUES IN REPORT FORM.

 INPUT/OUTPUT

     THEAD - HEADER TABLE ARRAY
     TABREC - ARRAY FOR A 600 WORD RECORD
     HEAD - A 4 X 12 ARRAY CONTAINING INFORMATION TO BE PRINTED AS
            THE PAGE HEADER.  4 INPUT CARDS PUNCHED IN COLUMNS 1-72.
     NRUN - COUNTER FOR EVENT NUMBER
     CDIM - CARD IMAGE OF GROUPED PARAMETER NAMES, PRE-PROCESSING CODE
            AND SAMPLE RATE OR EVENT NUMBERS.
     MODE - CODE SET TO INFORM PROGRAM IF ALL NON-CAL EVENTS ARE
            TO BE PROCESSED OR RUNS NAMED ON INPUT CARDS ARE TO
            BE PROCESSED.
                  MODE = 1 - PROCESS ALL NON-CAL EVENTS
                  MODE = 2 - PROCESS ONLY THOSE SPECIFIED INPUTS
     JRUN - INPUT RUN NUMBERS
     IPAGE - CONTROLS INPUT OF PARAMETER NAMES, PPC"S AND SR"S FOR
              PROGRAM LOGIC.
                  IPAGE = 0 - UNTIL FIRST OR NEXT "PAGE" CARD IS FOUND
                  IPAGE = 1 - WHEN "PAGE" CARD IS FOUND
                  IPAGE = 2 - IF PARAMETER CARDS TERMINATED BY
                              A NEW "PAGE" CARD
                  IPAGE = 3 - IF PARAMETER CARDS TERMINATED WHEN
                              18 PARAMETER CARDS HAVE BEEN READ
                  IF PARAMETER CARDS TERMINATED BY "END" CARD
     NAMES - PARAMETER MNEMONICS(NPAR IN LENGTH) MAX 18
     PPC - PRE-PROCESSING CODE(NPAR IN LENGTH) MAX 18
     SR - SAMPLE RATE(NPAR IN LENGTH) MAX 18
```

```
NPAR - NUMBER OF PARAMETERS COUNTED ON PROCESSING CARD
        INPUT OF PARAMETERS
TOS - TABLE OF SUBSCRIPTS DETERMINED FOR EACH PARAMETER
      THRU SUBROUTINE RDFSUB
             TOS(4*I-3) - PASS NO. FOR PARAMETER "I"
             TOS(4*I-2) - OFFSET POSITION OF PARAMETER IN
                          DATA FILE FRAME FOR PARAMETER "I"
             TOS(4*I-1) - COMPUTED "A" TERM FOR PARAMETER "I"
             TOS(4*I) - COMPUTED "B" TERM FOR PARAMETER "I"
JINDEX - INTEGER TO USE IN SUBROUTINE ENCD6 FOR ARRANGING
         THE OUTPUT STATISTICS INTO AN AESTHETIC FORM FOR
         PRINTING
CHBUFF - A 6 CHARACTER WORD OUTPUT FROM SUBROUTINE ENCD6
         THAT CONTAINS THE EDITED OUTPUT STATISTICS DATA
VALUE - THE PARTICULAR STATISTIC DATA VALUE SENT TO ENCD6 FOR
        EDITING INTO THE CHBUFF WORD
             FOR PPC"S = D OR S  VALUE = AVERAGE
             FOR PPC = V  VALUE = 95 PERCENT
COLTIT - ARRAY OF INFORMATION ON EACH PARAMETER NAME FOR
         PRINTING THE COLUMN TITLES OF REPORT
CTITL - COLUMN TITLES AS DEFINED IN SUBROUTINE GETUPC FOR
        EACH PARAMETER
LTITL - LINE TITLES AS DEFINED IN SUBROUTINE GETUPC FOR
        EACH PARAMETER (NOT USED IN THIS ROUTINE)
LINCNT - LINE COUNTER TO DETERMINE WHEN NEW PAGE HEADINGS
         ARE TO BE PRINTED
OUTBUF - 400 WORD BUFFER CONTAINING ONE RECORD OF STATISTICS
         DATA READ FROM THE STATISTICS FILE 09
LINBUF - ARRAY OF CHBUFF"S ARRANGED IN OUTPUT FORM
EVENT - EVENT NAME FOR EVENT = IRUN
```

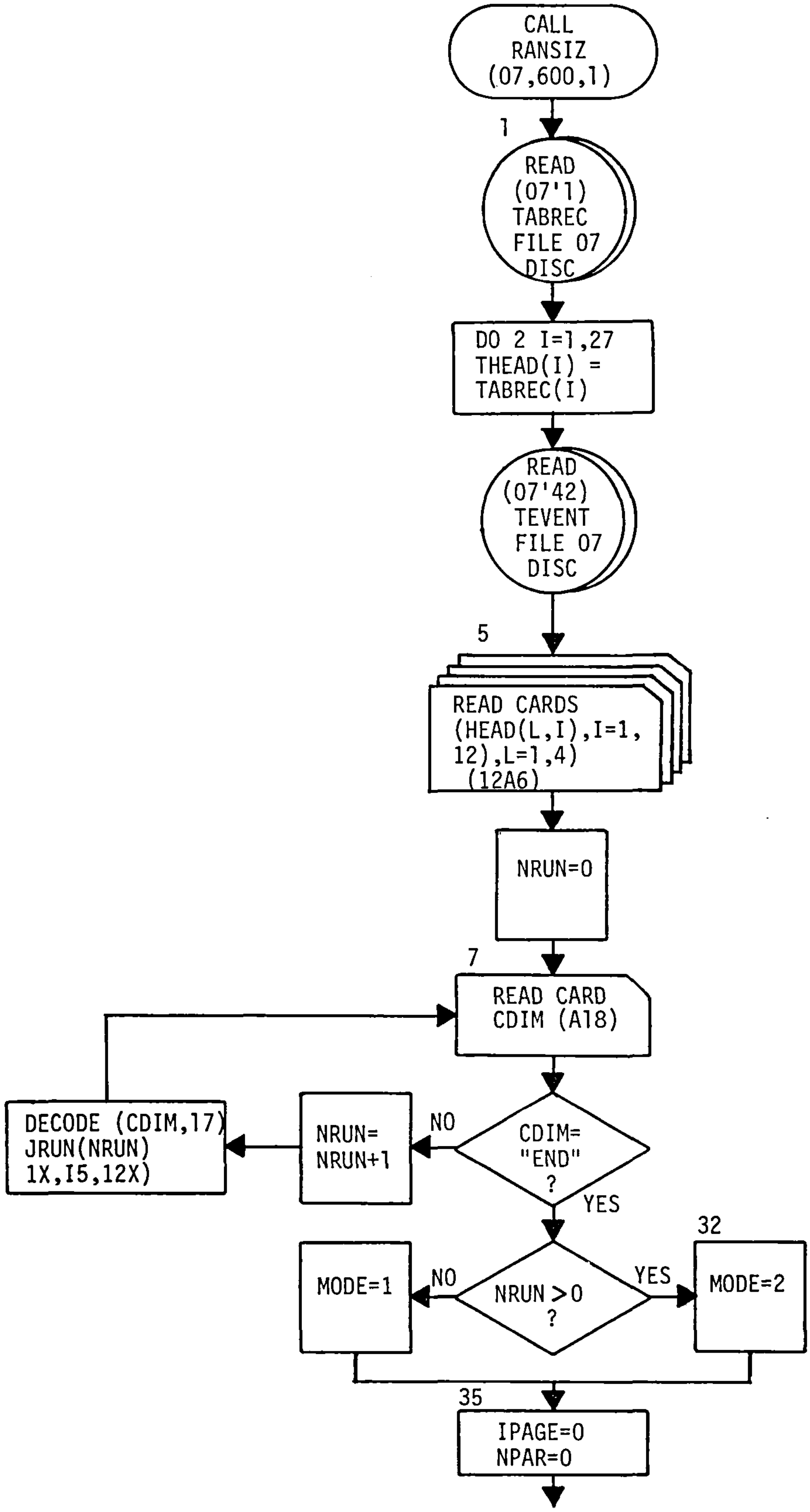
CALLING SEQUENCE

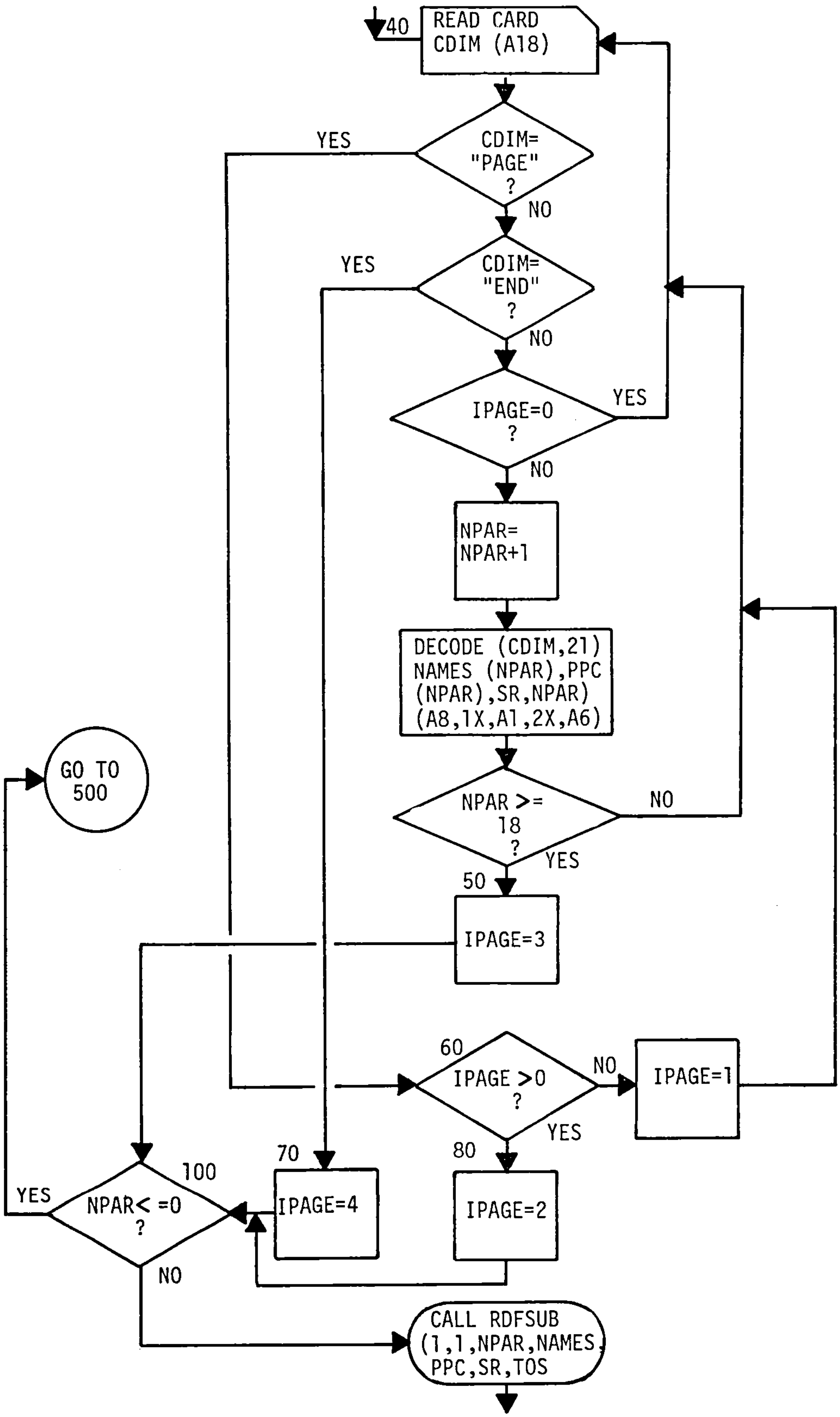
```
CALL RDFSUB(1,1,NPAR,NAMES,PPC,SR,TOS)
CALL GETUPC(NAMES(IPAR),PPC(IPAR),JINDEX(IPAR),CTITL,LTITL)
CALL RDFRD(1,1,1,IRUN,NPAR,NAMES,PPC,SR,ISMPL,LSMPL,
           OUTBUF,180,TOS,NFOUT,MSG,IER)
CALL ENCD6(VALUE,JINDEX(IPAR),CHBUFF)
```

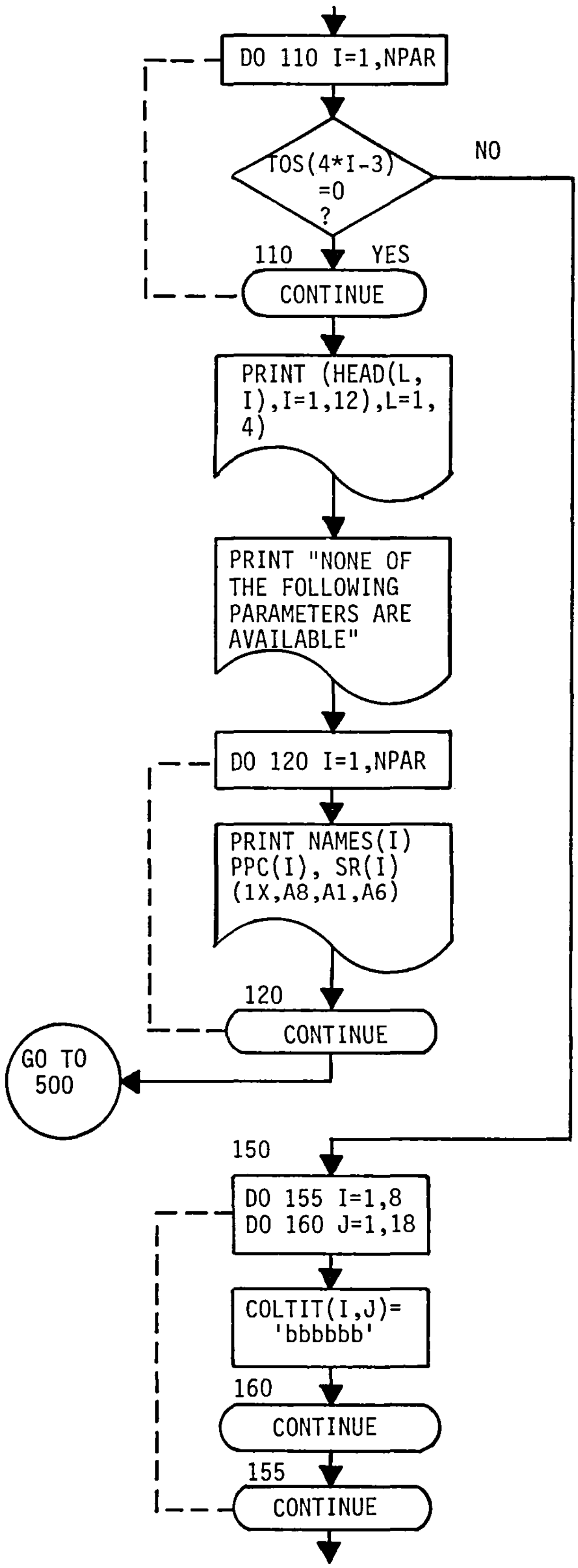
COMMON AREAS

/TABL1/THEAD

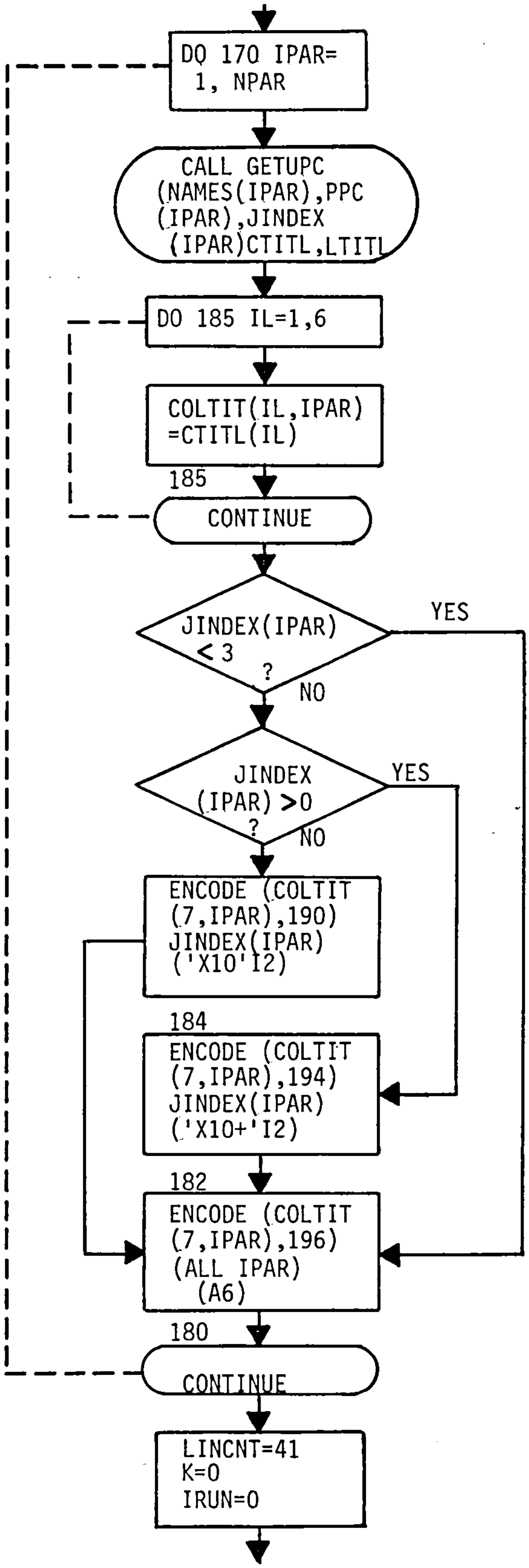
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS
FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION.


CALL
RANSIZ
(07,600,1)
1
READ
(07'1)
TABREC
FILE 07
DISC
DO 2 I=1,27
THEAD(I) =
TABREC(I)
READ
(07'42)
TEVENT
FILE 07
DISC
5
READ CARDS
(HEAD(L,I),I=1,
12),L=1,4)
(12A6)
NRUN=0
7
READ CARD
CDIM (A18)
DECODE (CDIM,17)
JRUN(NRUN)
1X,I5,12X)
NRUN=
NRUN+1
NO
CDIM=
"END"
?
YES
32
MODE=1
NO
NRUN > 0
?
YES
MODE=2
35
IPAGE=0
NPAR=0

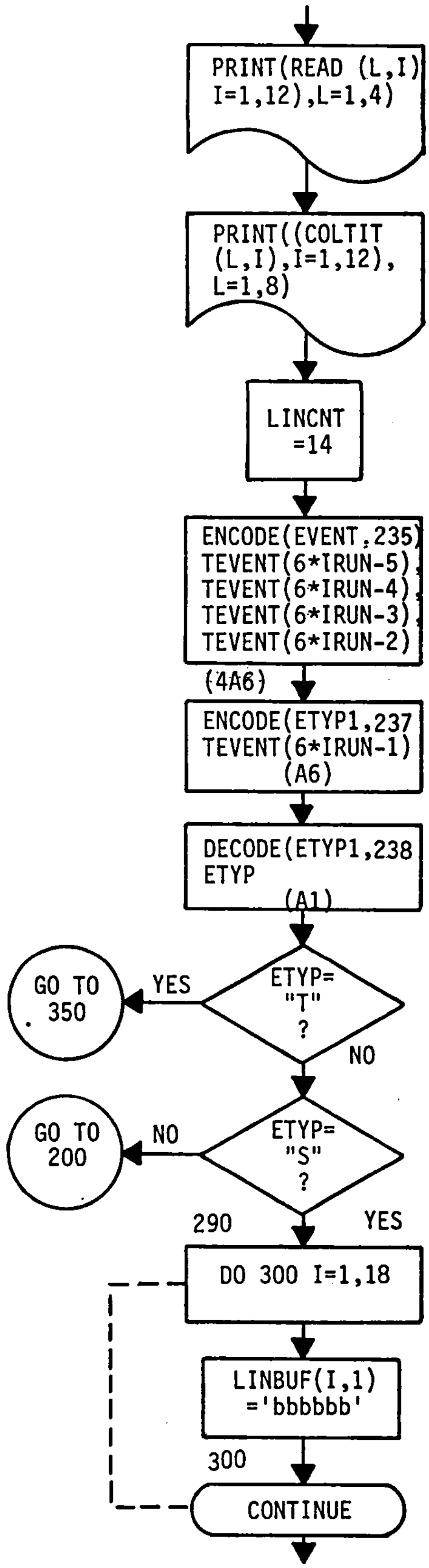

40
READ CARD
CDIM (A18)
CDIM=
"PAGE"
?
YES
NO
CDIM=
"END"
?
YES
NO
IPAGE=0
?
YES
NO
NPAR=
NPAR+1
DECODE (CDIM,21)
NAMES (NPAR),PPC
(NPAR),SR,NPAR)
(A8,1X,A1,2X,A6)
GO TO
500
NPAR >=
18
?
NO
YES
50
IPAGE=3
60
IPAGE >0
?
NO
IPAGE=1
YES
70
80
100
YES
NPAR< =0
?
IPAGE=4
IPAGE=2
NO
CALL RDFSUB
(1,1,NPAR,NAMES,
PPC,SR,TOS

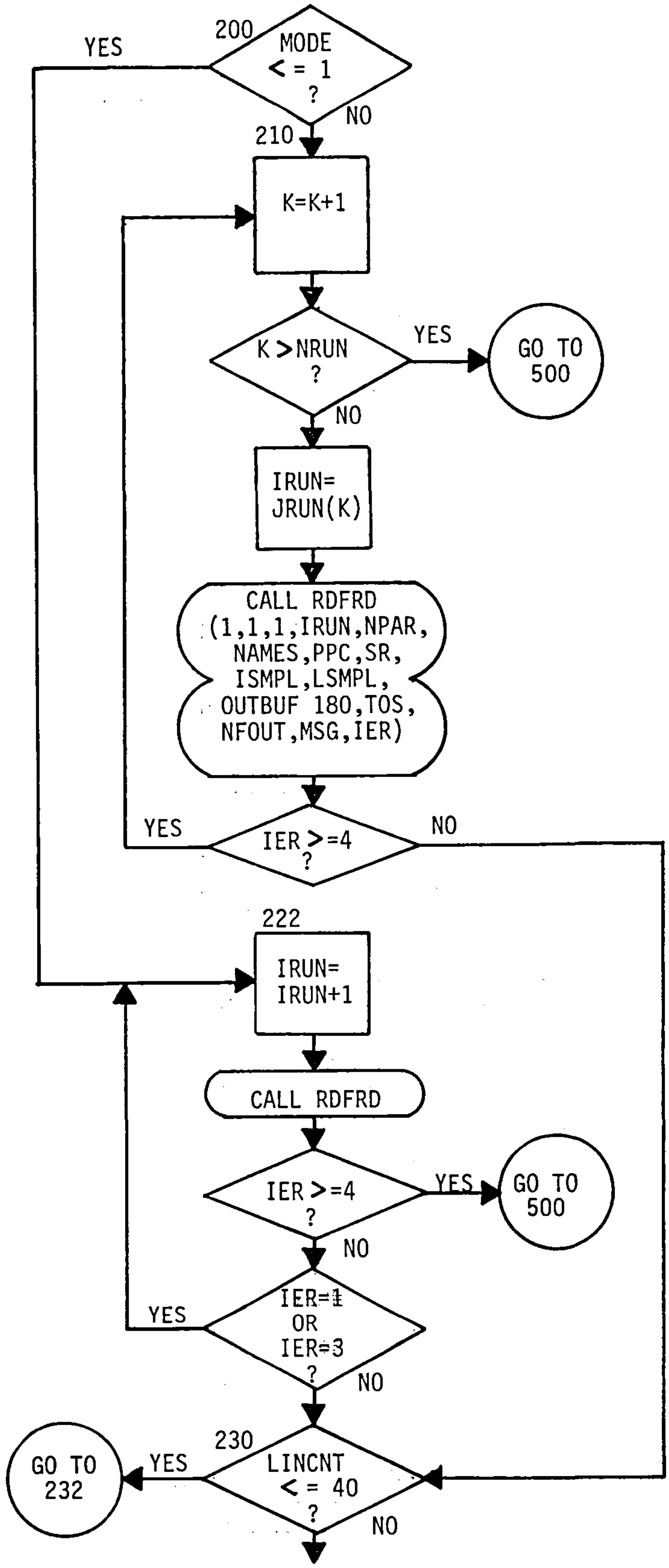


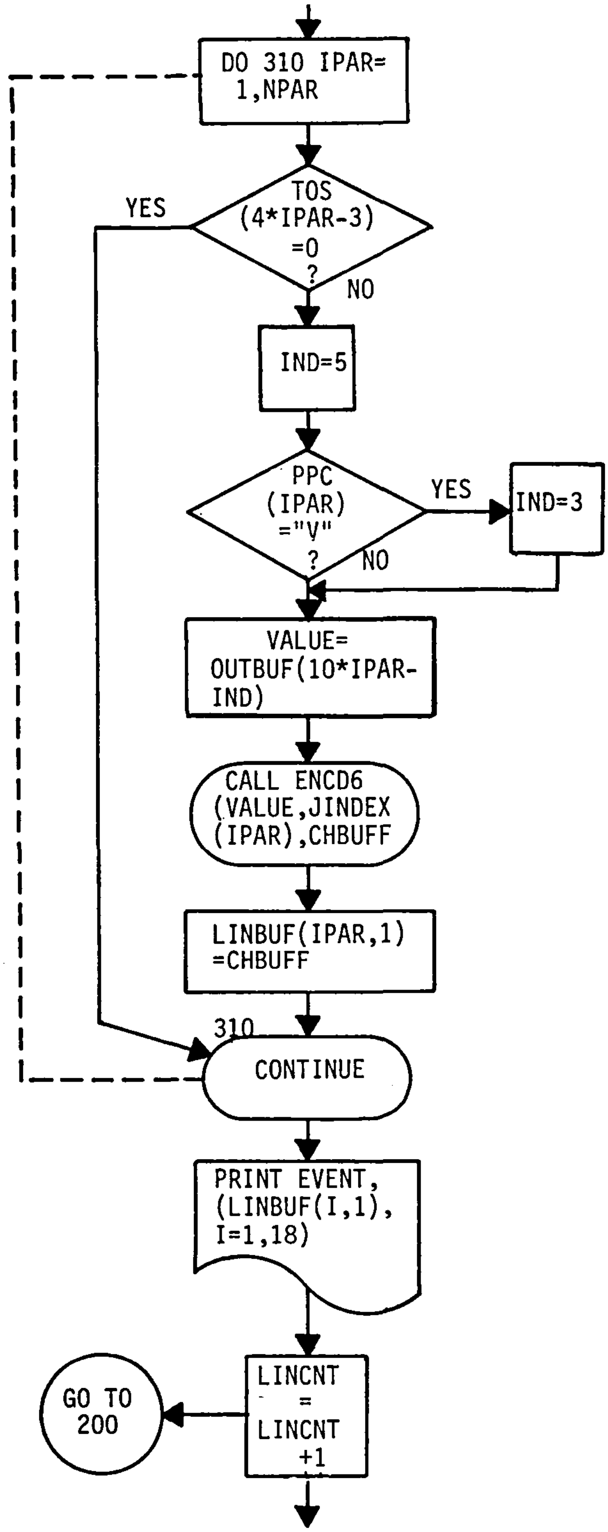
DO 110 I=1,NPAR
TOS(4*I-3) =0 ?
NO
110
YES
CONTINUE
PRINT (HEAD(L, I),I=1,12),L=1, 4)
PRINT "NONE OF THE FOLLOWING PARAMETERS ARE AVAILABLE"
DO 120 I=1,NPAR
PRINT NAMES(I) PPC(I), SR(I) (1X,A8,A1,A6)
120
CONTINUE
GO TO 500
150
DO 155 I=1,8
DO 160 J=1,18
COLTIT(I,J)= 'bbbbbb'
160
CONTINUE
155
CONTINUE

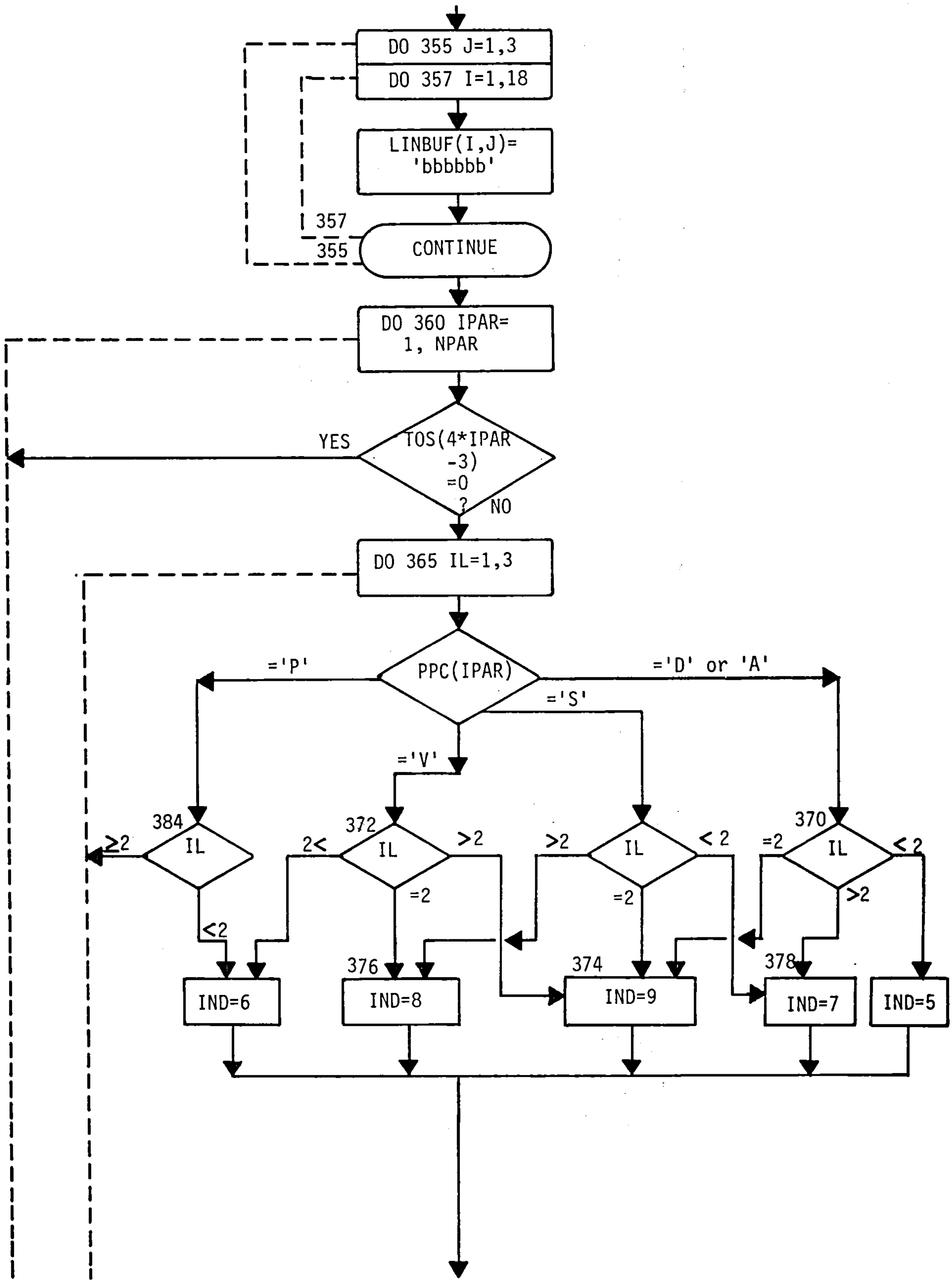


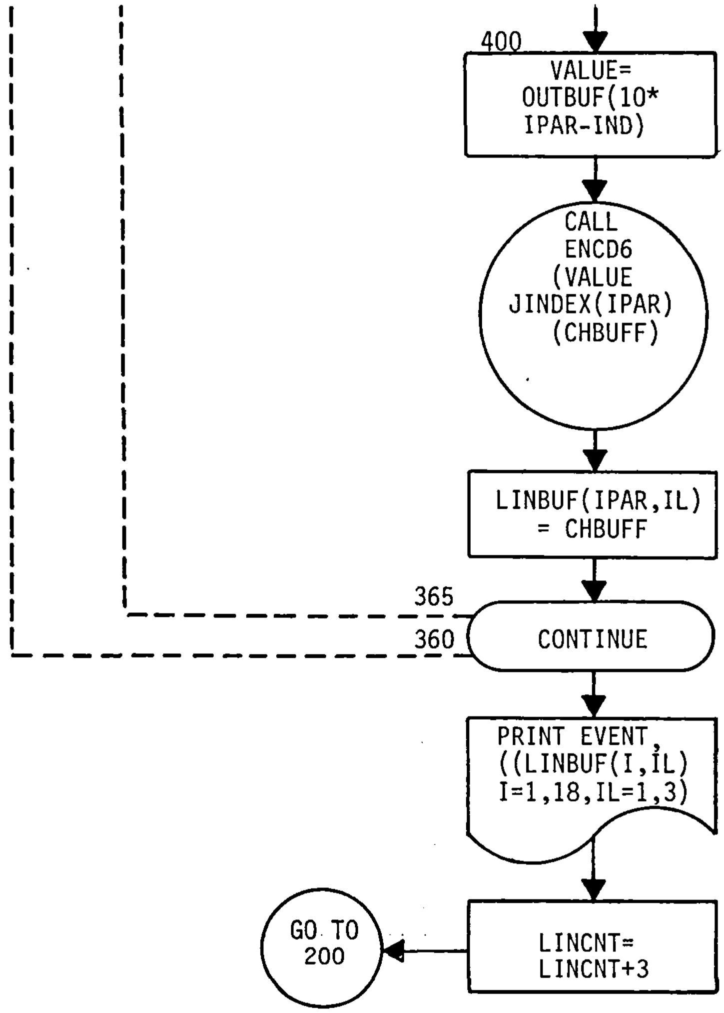
DO 170 IPAR=
1, NPAR
CALL GETUPC
(NAMES(IPAR),PPC
(IPAR),JINDEX
(IPAR)CTITL,LTITL
DO 185 IL=1,6
COLTIT(IL,IPAR)
=CTITL(IL)
185
CONTINUE
JINDEX(IPAR)
< 3
?
YES
NO
JINDEX
(IPAR) >0
?
YES
NO
ENCODE (COLTIT
(7,IPAR),190)
JINDEX(IPAR)
('X10'I2)
184
ENCODE (COLTIT
(7,IPAR),194)
JINDEX(IPAR)
('X10+'I2)
182
ENCODE (COLTIT
(7,IPAR),196)
(ALL IPAR)
(A6)
180
CONTINUE
LINCNT=41
K=0
IRUN=0

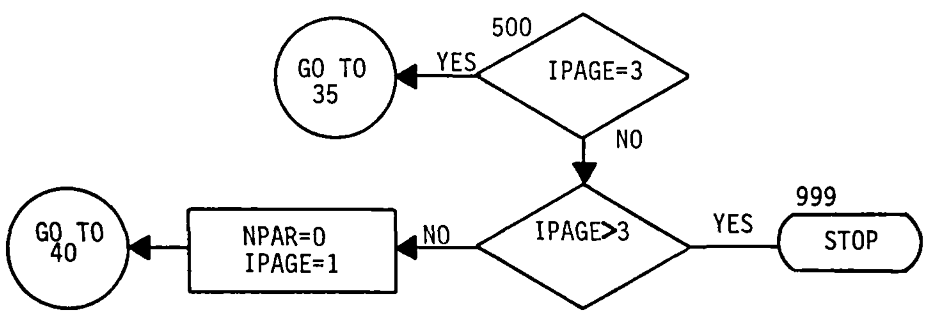

PRINT(READ (L,I)
I=1,12),L=1,4)
PRINT((COLTIT
(L,I),I=1,12),
L=1,8)
LINCNT
=14
ENCODE(EVENT,235)
TEVENT(6*IRUN-5),
TEVENT(6*IRUN-4),
TEVENT(6*IRUN-3),
TEVENT(6*IRUN-2)
(4A6)
ENCODE(ETYP1,237
TEVENT(6*IRUN-1)
(A6)
DECODE(ETYP1,238
ETYP
(A1)
GO TO
350
YES
ETYP=
"T"
?
NO
GO TO
200
NO
ETYP=
"S"
?
290
YES
DO 300 I=1,18
LINBUF(I,1)
='bbbbbb'
300
CONTINUE


200
MODE
<= 1
?
YES
NO
210
K=K+1
K >NRUN
?
YES
GO TO
500
NO
IRUN=
JRUN(K)
CALL RDFRD
(1,1,1,IRUN,NPAR,
NAMES,PPC,SR,
ISMPL,LSMPL,
OUTBUF 180,TOS,
NFOUT,MSG,IER)
IER >=4
?
YES
NO
222
IRUN=
IRUN+1
CALL RDFRD
IER >=4
?
YES
GO TO
500
NO
IER=1
OR
IER=3
?
YES
NO
230
LINCNT
<= 40
?
YES
GO TO
232
NO


DO 310 IPAR=
1,NPAR
TOS
(4*IPAR-3)
=0
?
YES
NO
IND=5
PPC
(IPAR)
="V"
?
YES
NO
IND=3
VALUE=
OUTBUF(10*IPAR-
IND)
CALL ENCD6
(VALUE,JINDEX
(IPAR),CHBUFF
LINBUF(IPAR,1)
=CHBUFF
310
CONTINUE
PRINT EVENT,
(LINBUF(I,1),
I=1,18)
LINCNT
=
LINCNT
+1
GO TO
200


DO 355 J=1,3
DO 357 I=1,18
LINBUF(I,J)=
'bbbbbb'
357
355
CONTINUE
DO 360 IPAR=
1, NPAR
YES
TOS(4*IPAR
-3)
=0
?
NO
DO 365 IL=1,3
='P'
PPC(IPAR)
='D' or 'A'
='S'
='V'
384
IL
≥2
<2
372
IL
2<
>2
=2
>2
IL
< 2
=2
370
IL
=2
< 2
>2
376
374
378
IND=6
IND=8
IND=9
IND=7
IND=5


400
VALUE=
OUTBUF(10*
IPAR-IND)
CALL
ENCD6
(VALUE
JINDEX(IPAR)
(CHBUFF)
LINBUF(IPAR,IL)
= CHBUFF
365
360
CONTINUE
PRINT EVENT,
((LINBUF(I,IL)
I=1,18,IL=1,3)
GO TO
200
LINCNT=
LINCNT+3


500
IPAGE=3
YES
GO TO
35
NO
IPAGE>3
YES
999
STOP
NO
NPAR=0
IPAGE=1
GO TO
40

GETUPC

```
      SUBROUTINE GETUPC (NAME, PPC, JINDEX, CTITL, LTITL)
************************** SUBROUTINE GETUPC **************************
```

PROGRAM IDENTIFICATION

```
PROGRAM NAME  - GETUPC
PROGRAM NO.   - 1.1.2320
AUTHOR        - DAVID L. DAVIS

COMPUTER      - HW 625/635
MEMORY        -
PERIPHERALS   - DISC SUBSYSTEM
LANGUAGE      - HW 6000 FORTRAN
```

PURPOSE

TO LOCATE ON THE UPC FILE AND RETURN TO A CALLING PROGRAN VARIOUS SPECIFICATIONS FOR A USER REQUESTED PARAMETER. IF THE REQUESTED PARAMETER CANNOT BE FOUND, ARTIFICIALLY BUILT SPECIFICATIONS WILL BE RETURNED.

METHOD

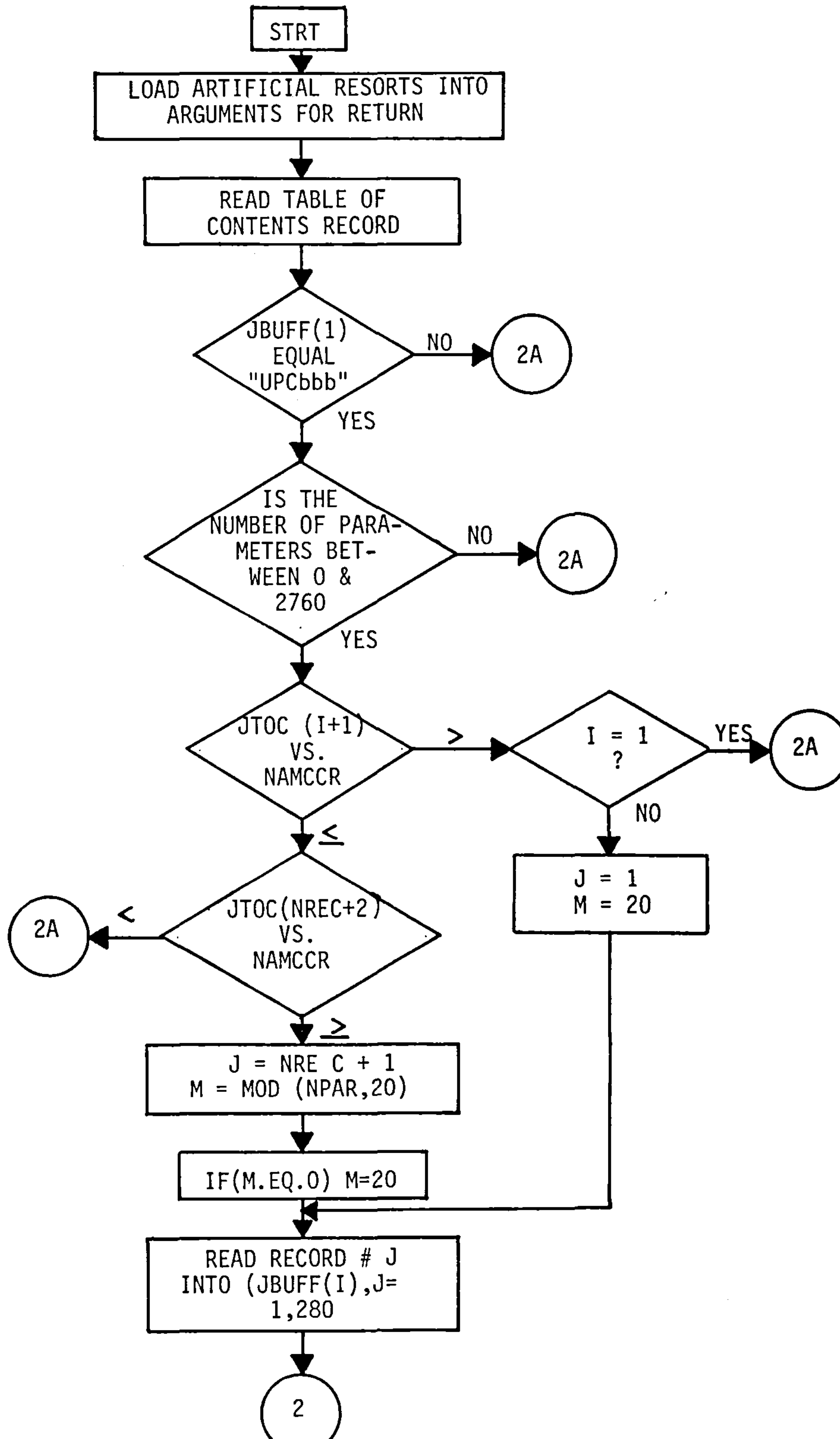
THE UPC FILE IS SEARCHED FOR A MATCH ON CONCATENATED NAME AND PRE-PROCESSING CODE. IF SUCCESSFUL THOSE RESULTS WILL BE RETURNED. IF NO MATCH IS FOUND THEN ARTIFICALLY GENERATED RESULTS WILL BE RETURNED.

INPUT/OUTPUT

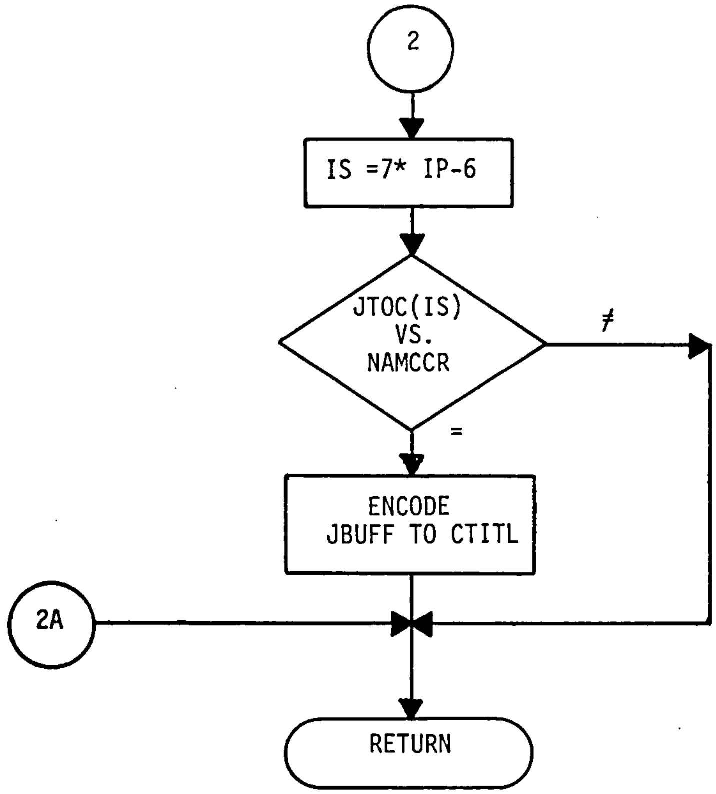
```
CALLING SEQUENCE
 CALL GETUPC(NAME,PPC,JINDEX,CTITL,LTITL) WHERE
   NAME   - INPUT ARGUMENT CONTAINING CONCATENATED NAME
   PPC    - INPUT ARGUMENT CONTAINING PRE-PRECESSING CODE
   JINDEX - OUTPUT ARGUMENT FOR INDEX INFO.
   CTITL  - OUTPUT ARGUMENT FOR COLUMN TITLE INFO.
   LTITL  - OUTPUT ARGUMENT FOR LINE TITLE INFO.
```

FILE DESCRIPTIONS

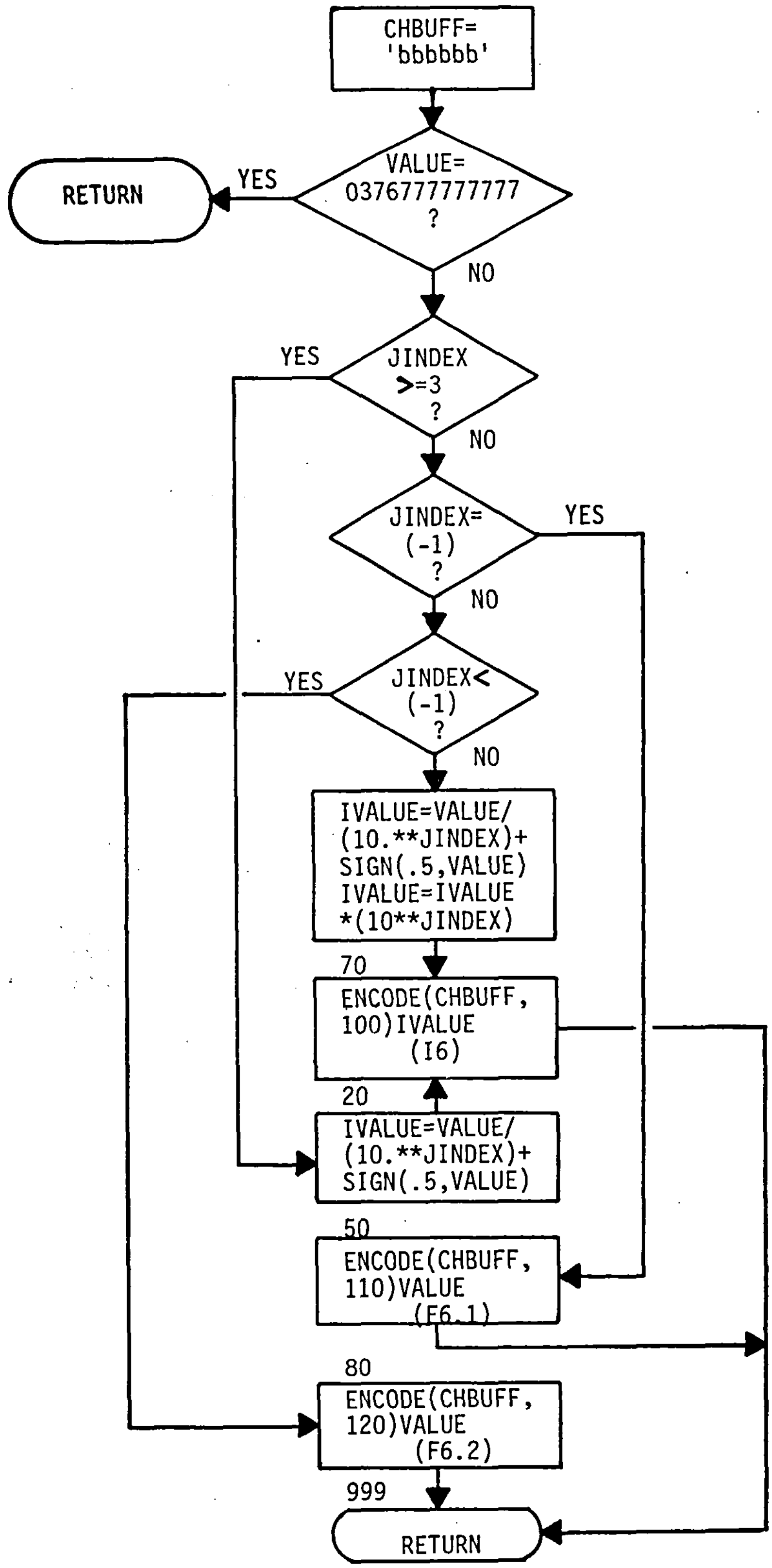
SEE DOCUMENTATION FOR ROUTINE #UPCGEN# FOR COMPLETE DESCRIPTIO OF THE #UPC# FILE.

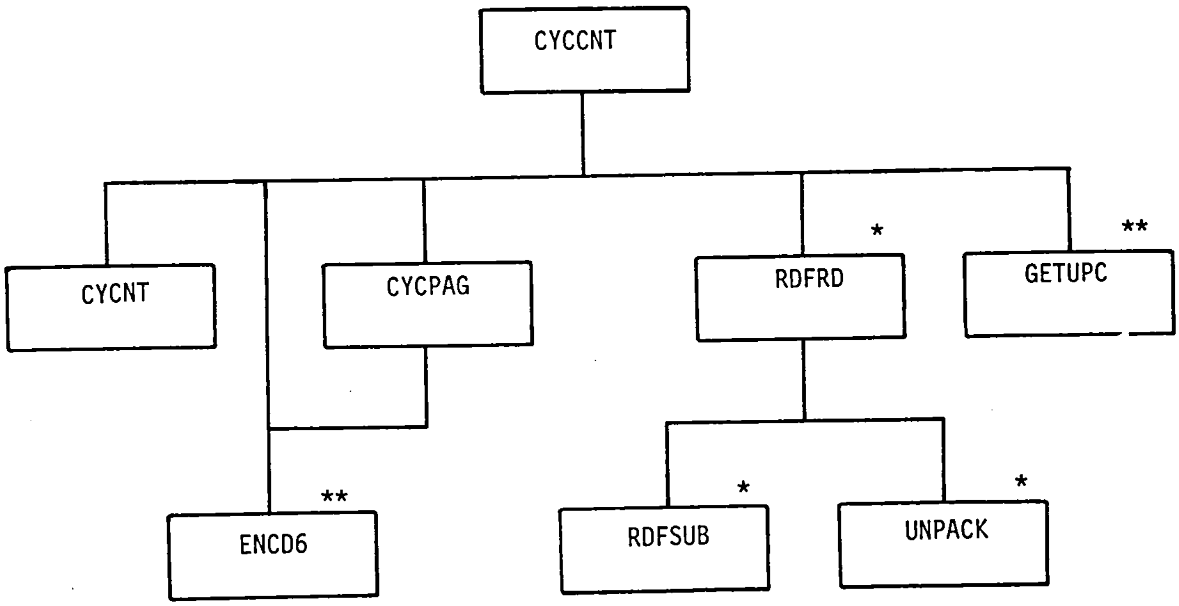


STRT
LOAD ARTIFICIAL RESORTS INTO ARGUMENTS FOR RETURN
READ TABLE OF CONTENTS RECORD
JBUFF(1) EQUAL "UPCbbb"
NO
2A
YES
IS THE NUMBER OF PARA-METERS BET-WEEN 0 & 2760
NO
2A
YES
JTOC (I+1) VS. NAMCCR
>
I = 1 ?
YES
2A
NO
J = 1
M = 20
≤
JTOC(NREC+2) VS. NAMCCR
<
2A
≥
J = NRE C + 1
M = MOD (NPAR,20)
IF(M.EQ.0) M=20
READ RECORD # J INTO (JBUFF(I),J= 1,280
2


2
IS =7* IP-6
JTOC(IS)
VS.
NAMCCR
≠
=
ENCODE
JBUFF TO CTITL
2A
RETURN

```
ENCD     SUBROUTINE ENCD6(VALUE,JINDEX,CHBUFF)

********SUBROUTINE ENCD6*************************************************

 PROGRAM IDENTIFICATION

     PROGRAM NAME ------ ENCD6
     PROGRAM NUMBER ------ 112320
     AUTHOR ------ TERRY D. SOMMERS

     COMPUTER ------- HW-625/635
     MEMORY -------
     PERIPHERALS ------
     LANGUAGE ------ HW 6000 FORTRAN/FORTY

 PURPOSE

     ROUTINE ENCODES VALUE IN 6 CHARACTER FIELD WITH FORMAT
     DETERMINED BY READOUT PRECISION CODE NUMBER, JINDEX.
     SCIENTIFIC NOTATION USED IF IABS(JINDEX) GE. 3.

 METHOD

        CHECK FOR NULL VALUE.  CHECK FOR JINDEX BEING GE. 3.
     REFORMAT VALUE ACCORDING TO JINDEX VALUE.  RETURN
     TO DATASK ROUTINE.

 INPUT/OUTPUT

     XULL - NULL VALUE EQUAL TO 0376777777777
     JINDEX - INTEGER TO USE IN SUBROUTINE ENCD6 FOR ARRANGING
              THE OUTPUT STATISTICS INTO AN AESTHETIC FORM FOR
              PRINTING
     CHBUFF - A 6 CHARACTER WORD OUTPUT FROM SUBROUTINE ENCD6
              THAT CONTAINS THE EDITED OUTPUT STATISTICS DATA
     VALUE - THE PARTICULAR STATISTIC DATA VALUE SENT TO ENCD6 FOR
             EDITING INTO THE CHBUFF WORD
                  FOR PPC"S = D OR S  VALUE = AVERAGE
                  FOR PPC = V  VALUE = 95 PERCENT

 CALLING SEQUENCE

     CALL ENCD6(VALUE,JINDEX,CHBUFF)

 COMMON AREAS

     NONE

        SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR
     SPECIFIC INFORMATION.
```


CHBUFF=
'bbbbbb'
VALUE=
0376777777777
?
YES
RETURN
NO
JINDEX
>=3
?
YES
NO
JINDEX=
(-1)
?
YES
NO
JINDEX<
(-1)
?
YES
NO
IVALUE=VALUE/
(10.**JINDEX)+
SIGN(.5,VALUE)
IVALUE=IVALUE
*(10**JINDEX)
70
ENCODE(CHBUFF,
100)IVALUE
(I6)
20
IVALUE=VALUE/
(10.**JINDEX)+
SIGN(.5,VALUE)
50
ENCODE(CHBUFF,
110)VALUE
(F6.1)
80
ENCODE(CHBUFF,
120)VALUE
(F6.2)
999
RETURN



*See Sub-program RDFRD for Comments and Flowcharts
**See Program DATASK for Comments and Flowcharts

HIERARCHY CHART for CYCLE COUNTS PROGRAM

CCNT SUBROUTINE CYCCNT

```
      SUBROUTINE CYCCNT
*************************************************************************
```

PROGRAM IDENTIFICATION

PROGRAM NAME ----- CYCCNT
PROGRAM NUMBER ----- 112336
AUTHOR ----- TERRY D. SOMMERS

COMPUTER ----- HW625/635
MEMORY ----- ALL FILES OPEN (20K)
PERIPHERALS ----- CARD READER,DISC,PRINTER
LANGUAGE ----- HW 6000 FORTRAN/FORTY

PURPOSE

TO PROCESS VIBRATORY COMPONENT OF PEAK STRESS CONVERTER FOR A SELECTED PARAMETER AND EVENT COMBINATION SO AS TO OBTAIN A COUNT OF THE NUMBER OF VIBRATORY CYCLES WHICH OCCUR IN EACH OF 8 CLASS INTERVALS. ALSO COMPUTED ARE THE ABSOLUTE MAXIMUM LEVEL AND ABSOLUTE MINIMUM LEVEL EXPERIENCED DURING THE EVENT FOR FM DATA ONLY.

METHOD

TWO METHODS OF EVENT SELECTION ARE IMPLEMENTED
1. ALL NON-CALIBRATION EVENTS
2. USER SPECIFIED EVENTS
THREE METHODS OF PARAMETER SELECTION ARE IMPLEMENTED
1. ALL AVAILABLE VIBRATORY PARAMETERS
2. ALL AVAILABLE VIBRATORY PARAMETERS ON A USER SPECIFIED PASS
3. USER SPECIFIED VIBRATORY PARAMETERS
IN ALL OF THE ABOVE THE CONCERN LEVEL GREATER THAN 0 MUST BE PRESENT ON THE SENSOR FILE OR THE PARAMETER WILL BE BYPASSED.

ASSUMPTIONS

1. VIBRATORY CANNOT EXIST WITHOUT CORRESPONDING STEADY.
2. VIBRATORY CANNOT BE A DERIVED PARAMETER.
3. STATS CANNOT EXIST WITHOUT DATA
4. ON ANY FRAME, IF VIBRATORY IS NULL, STEADY IS ALL NULL.

INPUT/OUTPUT

THEAD - HEADER TABLE ARRAY
TABREC ARRAY FOR A 600 WORD RECORD
TEVENT - EVENT ARRAY
NAME - SENSOR NAME
SR - SAMPLE RATE
PPC - PRE-PROCESSING CODE
IPASS - PASS NUMBER
MAXPAS - MAXIMUM NUMBER OF PASSES
CONLVL - CONCERN LEVEL

```
IENT - INITIAL DATA POINT
LEN - LAST DATA POINT
ISMPL - INITIAL SAMPLE
LSMPL - LAST SAMPLE
ABSMAX - ABSOLUTE MAXIMUM DATA POINT
ABSMIN - ABSOLUTE MINIMUM DATA POINT
FLTMAX - MAXIMUM DATA POINT
FLTMIN - MINIMUM DATA POINT
```

CALLING SEQUENCE

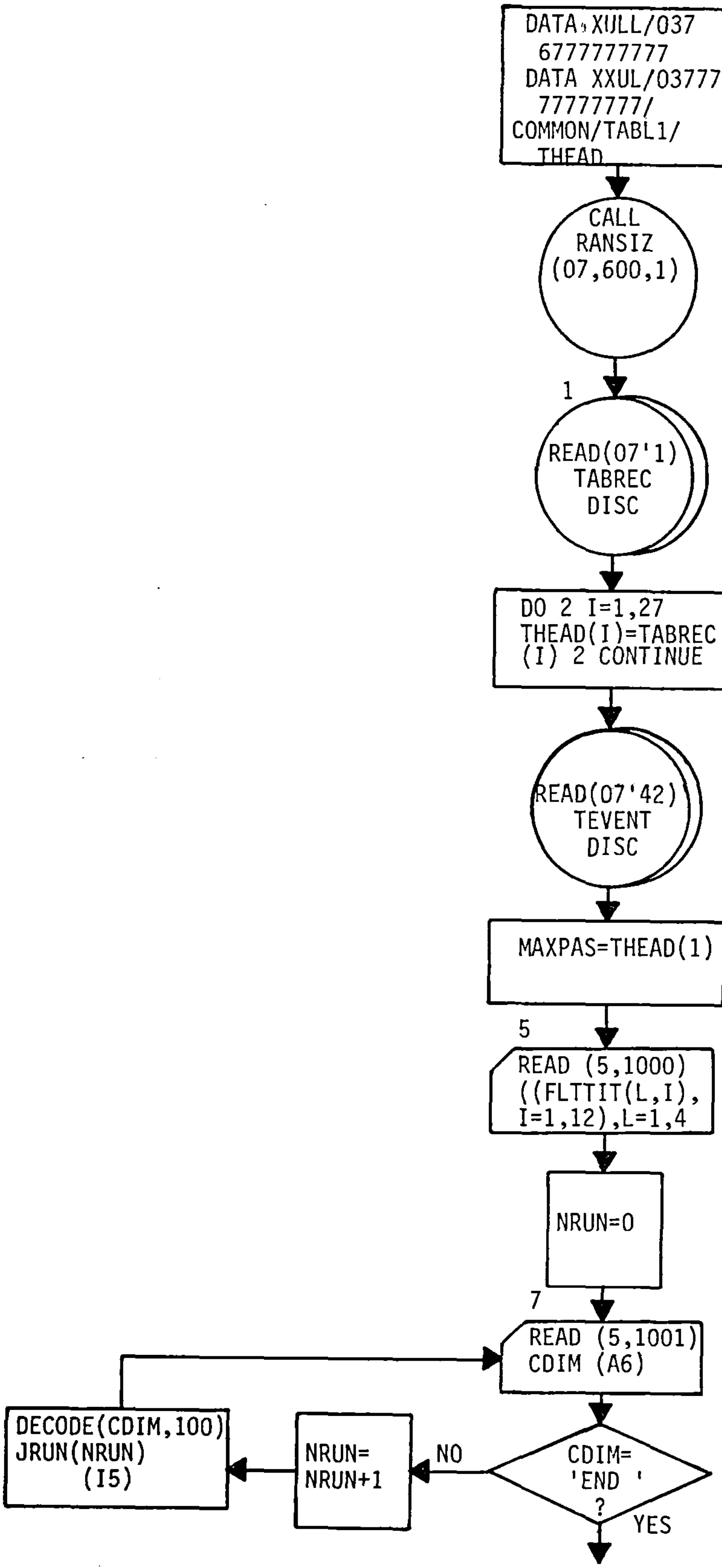
```
CALL RDFSUB(0,1,2,NAME,PPC,SR,TOS)
CALL GETUPC(NAME(1),PPC(1),JINDEX,CTITL,LTITL)
CALL RDFRD(1,1,1,IRUN,2,NAME,PPC,SR,ISMPL,LSMPL,OUTBUF,20,
          TOS,NFOUT,MSG,IER)
CALL ENCD6(OUTBUF(N),JINDEX,LINBUF(N))
CALL RDFRD(1,0,1,IRUN,2,NAME,PPC,SR,ISMPL,LSMPL,OUTBUF,2*LEN,
          TOS,NFOUT ,MSG,IER)
CALL CYCNT(IENT,NFRM,OUTBUF(1),OUTBUF(LEN+1),CONLVL,CYCINT,
          ABSMAX,ABSMIN,NGD)
CALL CYCPAG(FLTTIT,LTITL,SR(1),JINDEX,CONLVL)
```

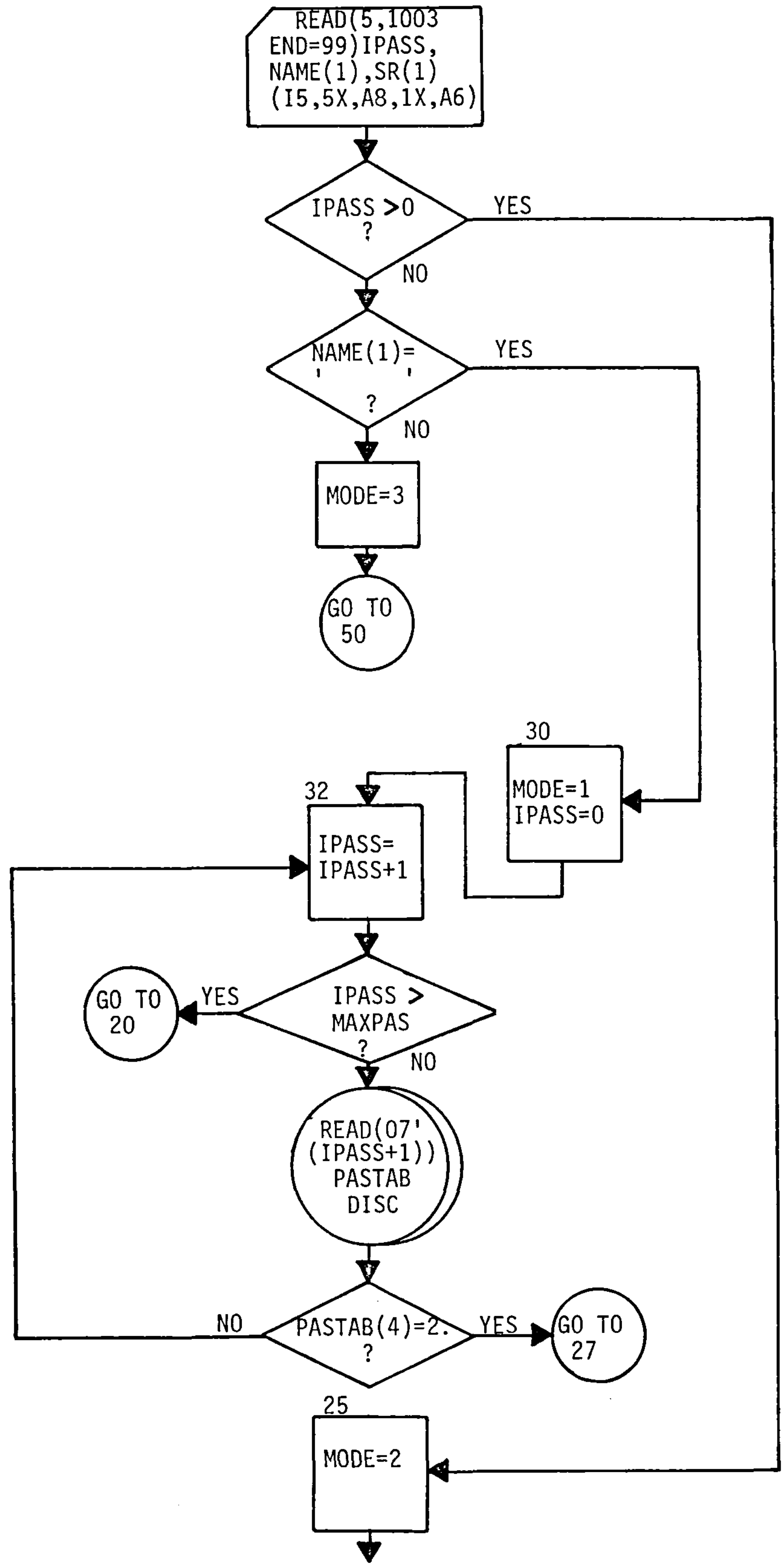
COMMON AREAS

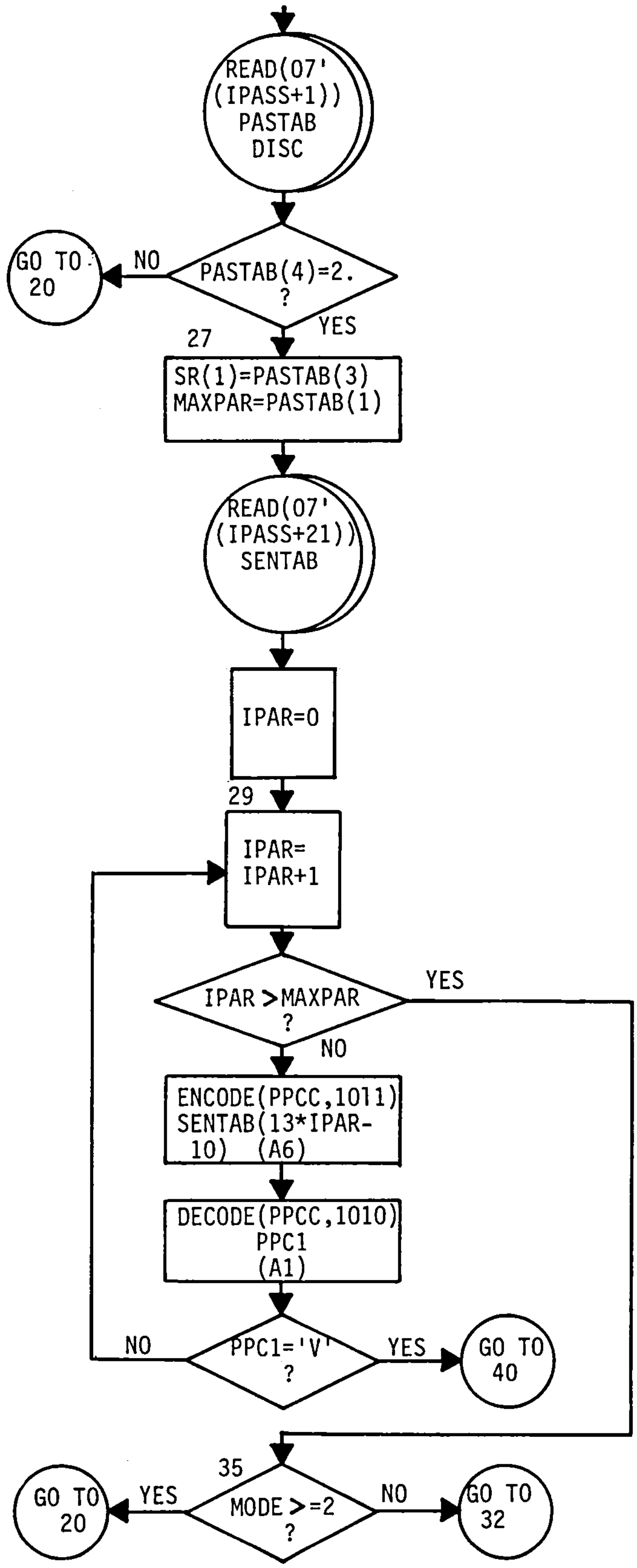
/TABL1/THEAD

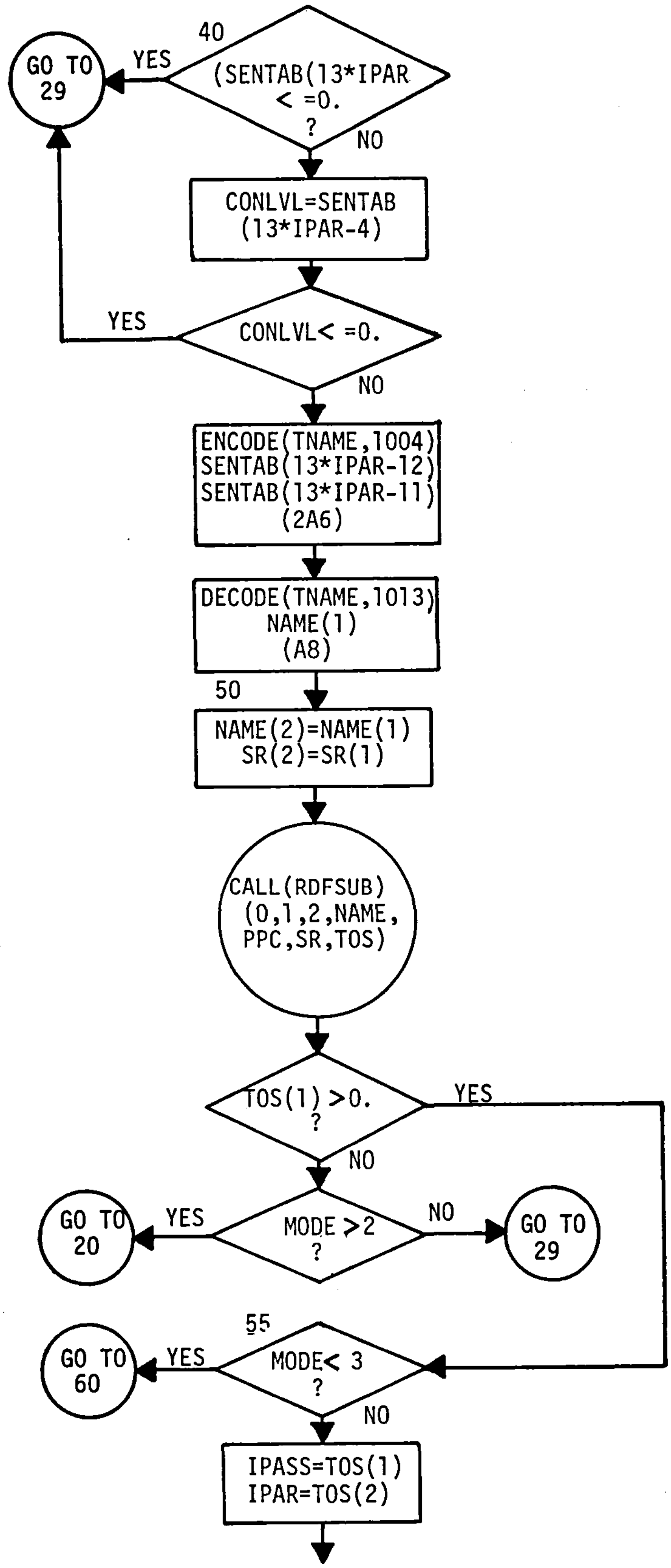
SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS, FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION.

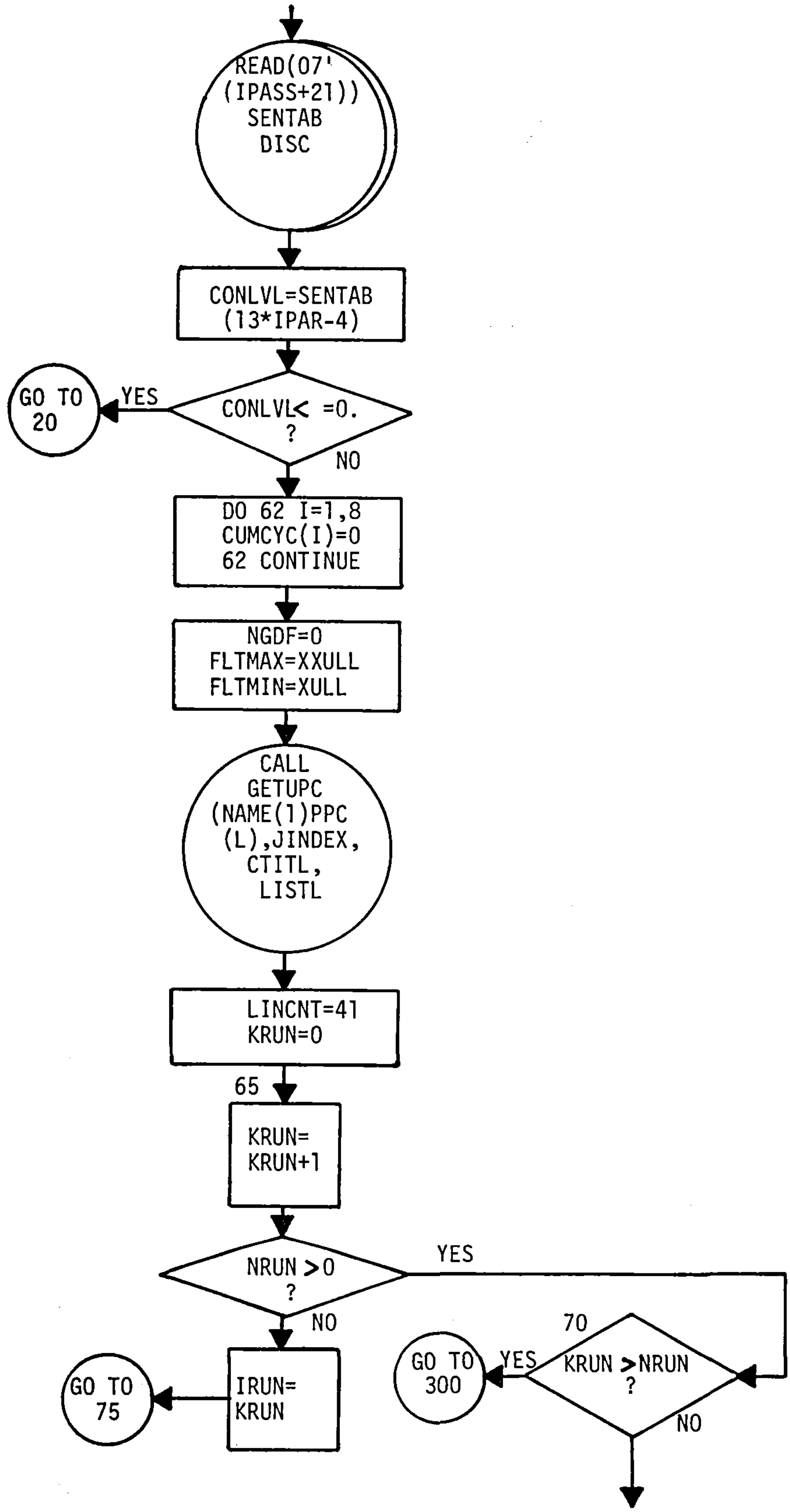
****

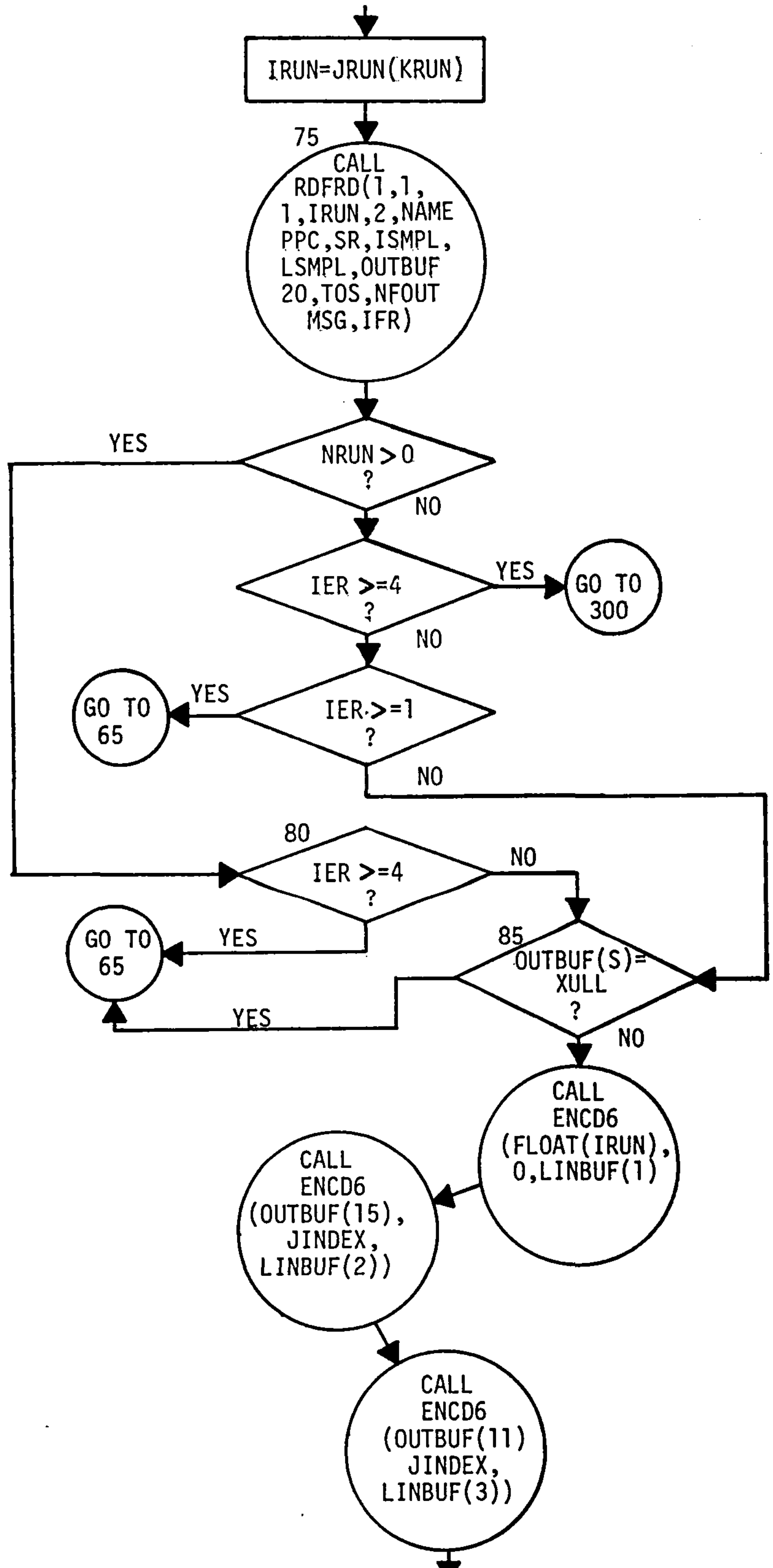

DATA XULL/037
6777777777
DATA XXUL/03777
77777777/
COMMON/TABL1/
THEAD
CALL
RANSIZ
(07,600,1)
1
READ(07'1)
TABREC
DISC
DO 2 I=1,27
THEAD(I)=TABREC
(I) 2 CONTINUE
READ(07'42)
TEVENT
DISC
MAXPAS=THEAD(1)
5
READ (5,1000)
((FLTTIT(L,I),
I=1,12),L=1,4
NRUN=0
7
READ (5,1001)
CDIM (A6)
CDIM=
'END '
?
YES
NO
NRUN=
NRUN+1
DECODE(CDIM,100)
JRUN(NRUN)
(I5)

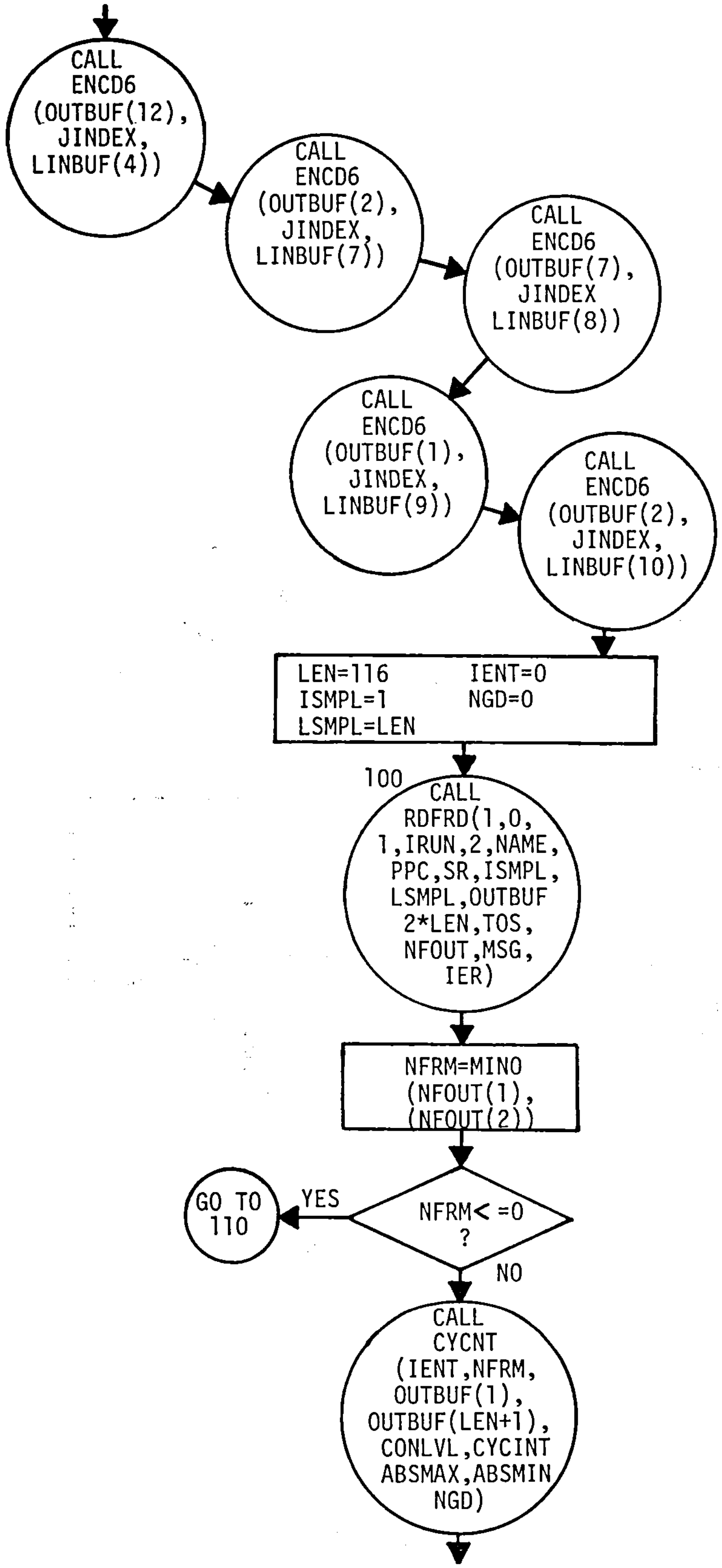

READ(5,1003
END=99)IPASS,
NAME(1),SR(1)
(I5,5X,A8,1X,A6)
IPASS >0
?
YES
NO
NAME(1)=
' '
?
YES
NO
MODE=3
GO TO
50
30
MODE=1
IPASS=0
32
IPASS=
IPASS+1
GO TO
20
YES
IPASS >
MAXPAS
?
NO
READ(07'
(IPASS+1))
PASTAB
DISC
NO
PASTAB(4)=2.
?
YES
GO TO
27
25
MODE=2

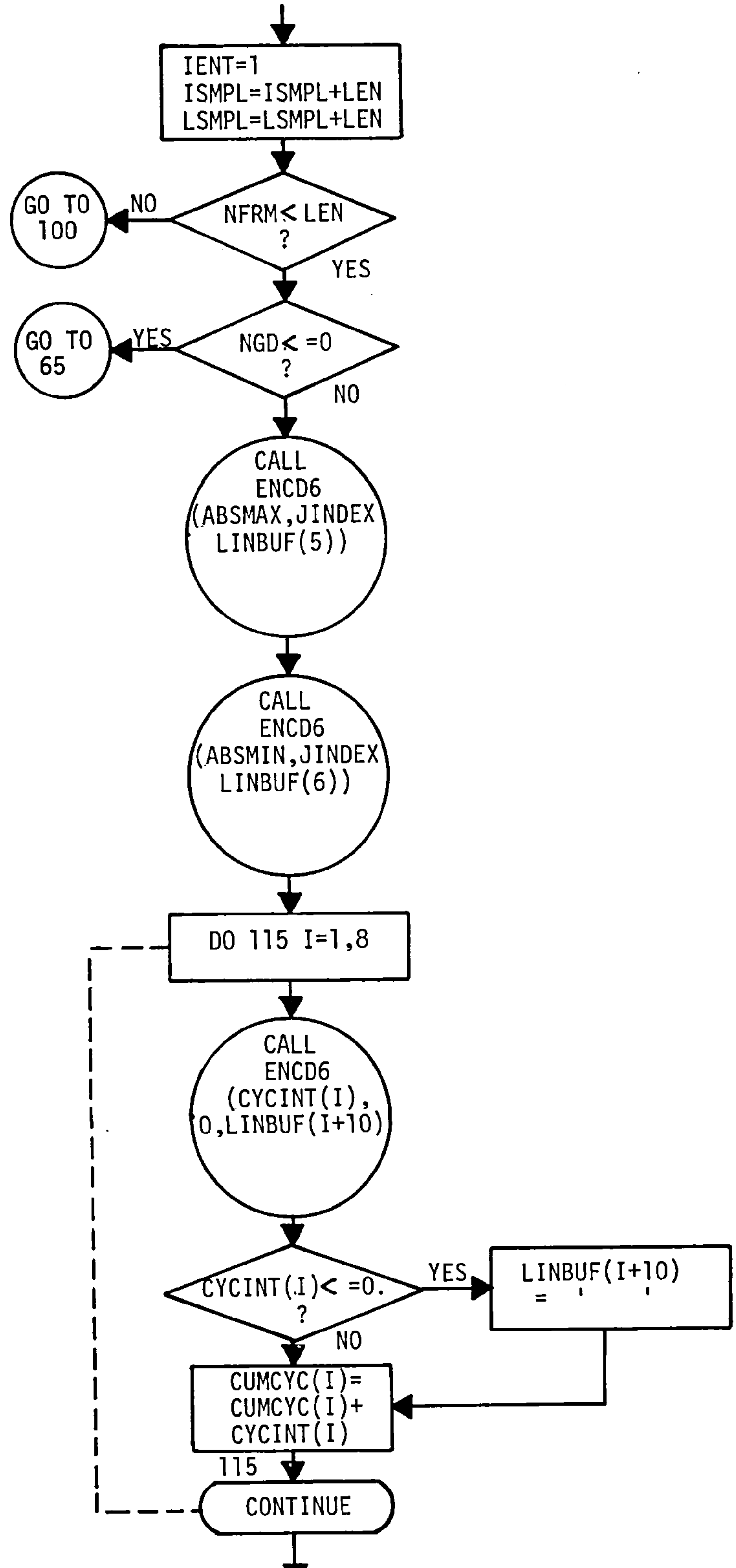

READ(07'
(IPASS+1))
PASTAB
DISC
GO TO
20
NO
PASTAB(4)=2.
?
YES
27
SR(1)=PASTAB(3)
MAXPAR=PASTAB(1)
READ(07'
(IPASS+21))
SENTAB
IPAR=0
29
IPAR=
IPAR+1
IPAR >MAXPAR
?
YES
NO
ENCODE(PPCC,1011)
SENTAB(13*IPAR-
10) (A6)
DECODE(PPCC,1010)
PPC1
(A1)
NO
PPC1='V'
?
YES
GO TO
40
35
GO TO
20
YES
MODE >=2
?
NO
GO TO
32

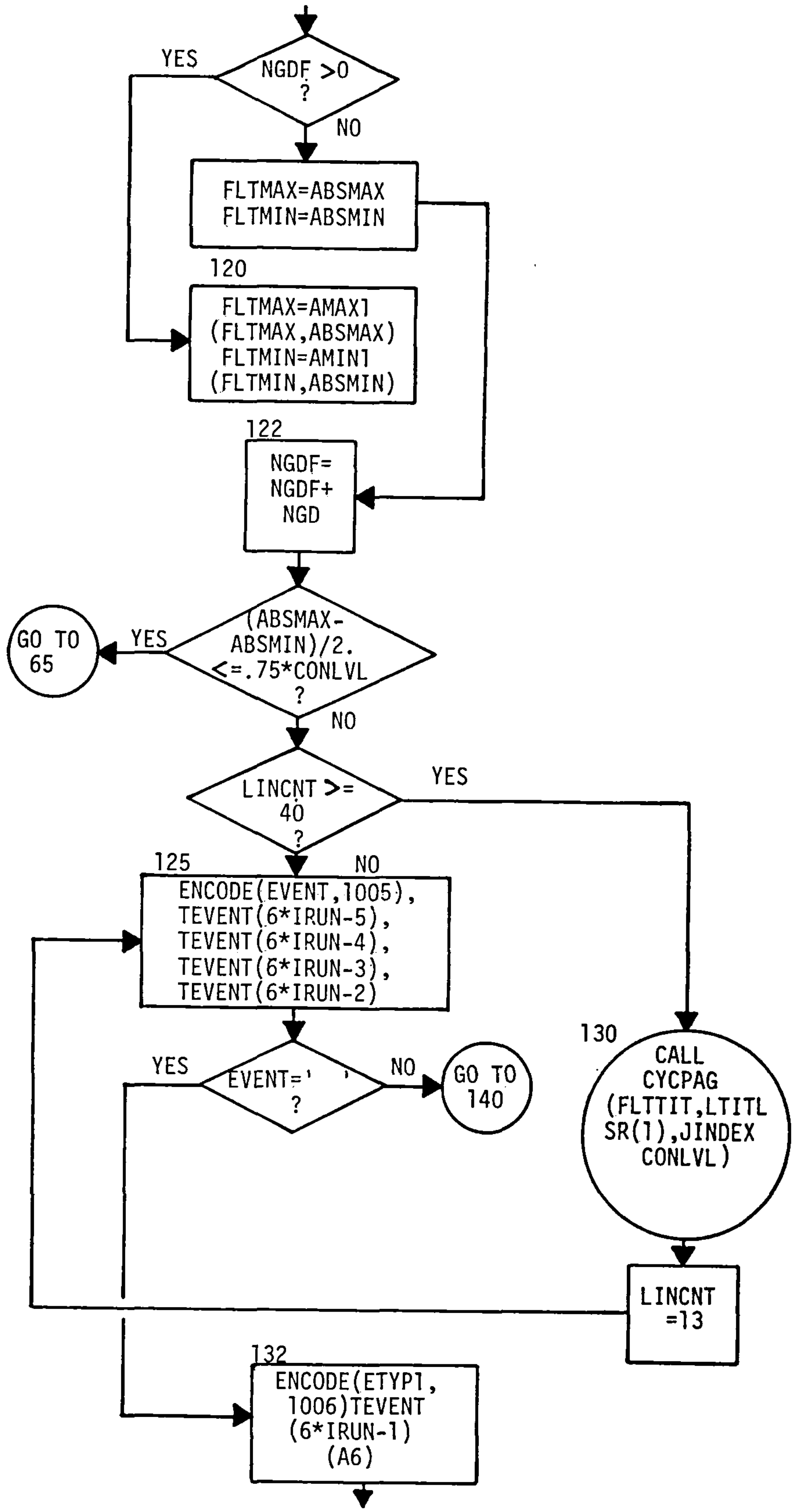

40
(SENTAB(13*IPAR
< =0.
?
YES
GO TO 29
NO
CONLVL=SENTAB (13*IPAR-4)
CONLVL< =0.
YES
NO
ENCODE(TNAME,1004)
SENTAB(13*IPAR-12)
SENTAB(13*IPAR-11)
(2A6)
DECODE(TNAME,1013)
NAME(1)
(A8)
50
NAME(2)=NAME(1)
SR(2)=SR(1)
CALL(RDFSUB)
(0,1,2,NAME,
PPC,SR,TOS)
TOS(1) >0.
?
YES
NO
MODE >2
?
YES
GO TO 20
NO
GO TO 29
55
MODE< 3
?
YES
GO TO 60
NO
IPASS=TOS(1)
IPAR=TOS(2)

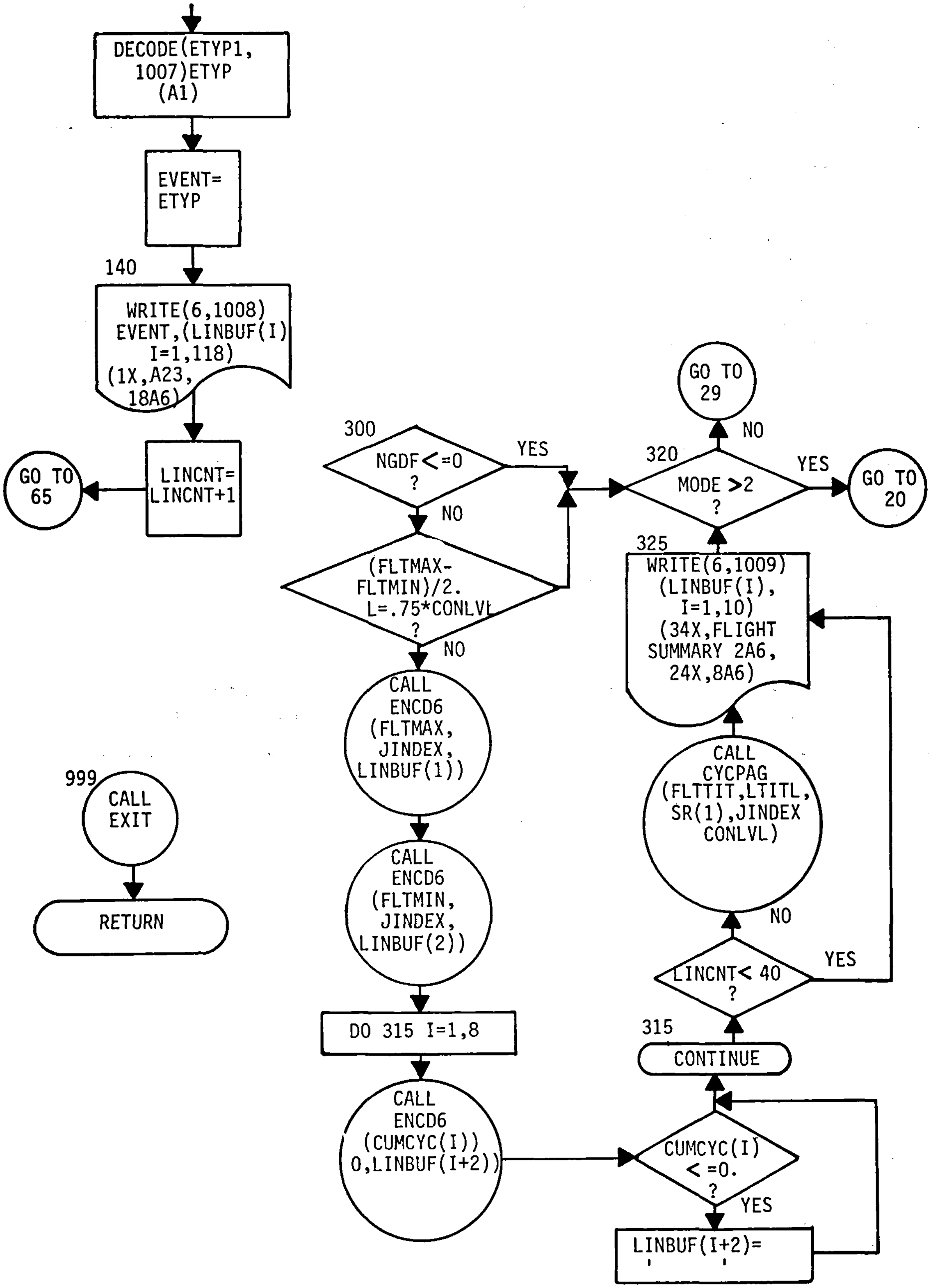

READ(07'
(IPASS+21))
SENTAB
DISC
CONLVL=SENTAB
(13*IPAR-4)
GO TO
20
YES
CONLVL< =0.
?
NO
DO 62 I=1,8
CUMCYC(I)=0
62 CONTINUE
NGDF=0
FLTMAX=XXULL
FLTMIN=XULL
CALL
GETUPC
(NAME(1)PPC
(L),JINDEX,
CTITL,
LISTL
LINCNT=41
KRUN=0
65
KRUN=
KRUN+1
NRUN >0
?
YES
NO
GO TO
75
IRUN=
KRUN
70
GO TO
300
YES
KRUN >NRUN
?
NO


IRUN=JRUN(KRUN)
75
CALL RDFRD(1,1, 1,IRUN,2,NAME PPC,SR,ISMPL, LSMPL,OUTBUF 20,TOS,NFOUT MSG,IFR)
YES
NRUN > 0 ?
NO
IER >=4 ?
YES
GO TO 300
NO
GO TO 65
YES
IER >=1 ?
NO
80
IER >=4 ?
NO
GO TO 65
YES
85
OUTBUF(S)= XULL ?
YES
NO
CALL ENCD6 (FLOAT(IRUN), 0,LINBUF(1)
CALL ENCD6 (OUTBUF(15), JINDEX, LINBUF(2))
CALL ENCD6 (OUTBUF(11) JINDEX, LINBUF(3))


CALL
ENCD6
(OUTBUF(12),
JINDEX,
LINBUF(4))
CALL
ENCD6
(OUTBUF(2),
JINDEX,
LINBUF(7))
CALL
ENCD6
(OUTBUF(7),
JINDEX
LINBUF(8))
CALL
ENCD6
(OUTBUF(1),
JINDEX,
LINBUF(9))
CALL
ENCD6
(OUTBUF(2),
JINDEX,
LINBUF(10))
LEN=116 IENT=0
ISMPL=1 NGD=0
LSMPL=LEN
100
CALL
RDFRD(1,0,
1,IRUN,2,NAME,
PPC,SR,ISMPL,
LSMPL,OUTBUF
2*LEN,TOS,
NFOUT,MSG,
IER)
NFRM=MINO
(NFOUT(1),
(NFOUT(2))
GO TO
110
YES
NFRM< =0
?
NO
CALL
CYCNT
(IENT,NFRM,
OUTBUF(1),
OUTBUF(LEN+1),
CONLVL,CYCINT
ABSMAX,ABSMIN
NGD)


IENT=1
ISMPL=ISMPL+LEN
LSMPL=LSMPL+LEN
NFRM< LEN ?
NO
YES
GO TO 100
NGD< =0 ?
YES
NO
GO TO 65
CALL ENCD6 (ABSMAX,JINDEX LINBUF(5))
CALL ENCD6 (ABSMIN,JINDEX LINBUF(6))
DO 115 I=1,8
CALL ENCD6 (CYCINT(I), 0,LINBUF(I+10)
CYCINT(I)< =0. ?
YES
NO
LINBUF(I+10) = ' '
CUMCYC(I)= CUMCYC(I)+ CYCINT(I)
115
CONTINUE


NGDF >0
?
YES
NO
FLTMAX=ABSMAX
FLTMIN=ABSMIN
120
FLTMAX=AMAX1
(FLTMAX,ABSMAX)
FLTMIN=AMIN1
(FLTMIN,ABSMIN)
122
NGDF=
NGDF+
NGD
(ABSMAX-
ABSMIN)/2.
<=.75*CONLVL
?
YES
GO TO
65
NO
LINCNT >=
40
?
YES
NO
125
ENCODE(EVENT,1005),
TEVENT(6*IRUN-5),
TEVENT(6*IRUN-4),
TEVENT(6*IRUN-3),
TEVENT(6*IRUN-2)
EVENT=' '
?
YES
NO
GO TO
140
130
CALL
CYCPAG
(FLTTIT,LTITL
SR(1),JINDEX
CONLVL)
LINCNT
=13
132
ENCODE(ETYP1,
1006)TEVENT
(6*IRUN-1)
(A6)


DECODE(ETYP1, 1007)ETYP (A1)
EVENT= ETYP
140
WRITE(6,1008) EVENT,(LINBUF(I) I=1,118) (1X,A23, 18A6)
LINCNT= LINCNT+1
GO TO 65
999
CALL EXIT
RETURN
300
NGDF<=0 ?
YES
NO
(FLTMAX- FLTMIN)/2. L=.75*CONLVL ?
NO
CALL ENCD6 (FLTMAX, JINDEX, LINBUF(1))
CALL ENCD6 (FLTMIN, JINDEX, LINBUF(2))
DO 315 I=1,8
CALL ENCD6 (CUMCYC(I)) 0,LINBUF(I+2))
GO TO 29
NO
320
MODE >2 ?
YES
GO TO 20
325
WRITE(6,1009) (LINBUF(I), I=1,10) (34X,FLIGHT SUMMARY 2A6, 24X,8A6)
CALL CYCPAG (FLTTIT,LTITL, SR(1),JINDEX CONLVL)
NO
LINCNT< 40 ?
YES
315
CONTINUE
CUMCYC(I) <=0. ?
YES
LINBUF(I+2)= ' '

CYCT   SUBROUTINE CYCNT

```
      SUBROUTINE CYCNT(IENT,NPT,DATAV,DATAS,CONLVL,CYCINT,
     1 ABSMAX,ABSMIN,NGD)
********SUBROUTINE CYCNT*************************************************
```

PROGRAM IDENTIFICATION

PROGRAM NAME ----- CYCNT
PROGRAM NUMBER ----- 112336
AUTHOR ----- TERRY D. SOMMERS

COMPUTER ----- HW625/635
MEMORY -----
PERIPHERALS -----
LANGUAGE ----- HW 6000 FORTRAN/FORTY

PURPOSE

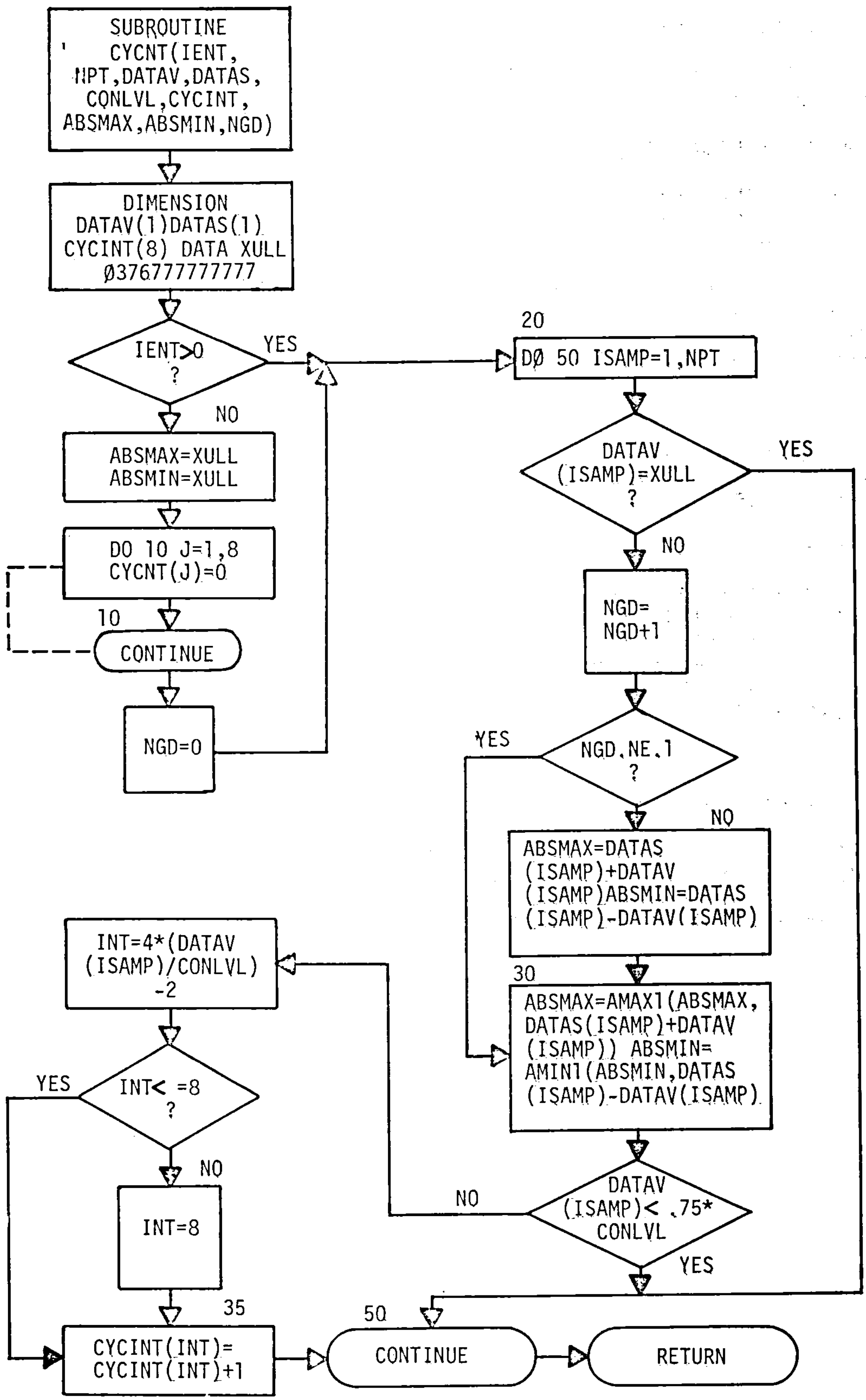
TO DETERMINE THE ABSOLUTE MAXIMUM AND MINIMUM DATA VALUES AND TO COMPUTE CONCERN LEVEL.

INPUT/OUTPUT

ABSMAX - ABSOLUTE MAXIMUM DATA VALUE
ABSMIN - ABSOLUTE MINIMUM DATA VALUE
DATAV - VIBRATORY DATA
DATAS - STEADY DATA
XULL - NULL VALUE

SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS, FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION.

```
************************************************************************
```



SUBROUTINE CYCNT(IENT, NPT,DATAV,DATAS, CONLVL,CYCINT, ABSMAX,ABSMIN,NGD)
DIMENSION DATAV(1)DATAS(1) CYCINT(8) DATA XULL Ø3767777777777
IENT>0 ?
YES
NO
ABSMAX=XULL ABSMIN=XULL
DO 10 J=1,8 CYCNT(J)=0
10
CONTINUE
NGD=0
20
DØ 50 ISAMP=1,NPT
DATAV (ISAMP)=XULL ?
YES
NO
NGD= NGD+1
YES
NGD.NE.1 ?
NO
ABSMAX=DATAS (ISAMP)+DATAV (ISAMP)ABSMIN=DATAS (ISAMP)-DATAV(ISAMP)
30
ABSMAX=AMAX1(ABSMAX, DATAS(ISAMP)+DATAV (ISAMP)) ABSMIN= AMIN1(ABSMIN,DATAS (ISAMP)-DATAV(ISAMP)
INT=4*(DATAV (ISAMP)/CONLVL) -2
YES
INT< =8 ?
NO
INT=8
NO
DATAV (ISAMP)< .75* CONLVL
YES
35
CYCINT(INT)= CYCINT(INT)+1
50
CONTINUE
RETURN

CYPC   SUBROUTINE CYCPAG

```
      SUBROUTINE CYCPAG(FLTTIT,LTITL,SR,JINDEX,CONLVL)
********SUBROUTINE CYCPAG************************************************
```

PROGRAM IDENTIFICATION

PROGRAM NAME ----- CYCPAG
PROGRAM NUMBER ----- 112336
AUTHOR ----- TERRY D. SOMMERS

COMPUTER ----- HW625/635
MEMORY -----
PERIPHERALS -----
LANGUAGE ----- HW 6000 FORTRAN/FORTY

PURPOSE

TO GENERATE REPORT OF VIBRATORY CYCLE COUNT SHOWING VIBRATORY AND STEADY DATA, PERCENT OF CONCERN LEVEL, ABSOLUTE LEVELS FOR EACH MANEUVER.

INPUT/OUTPUT

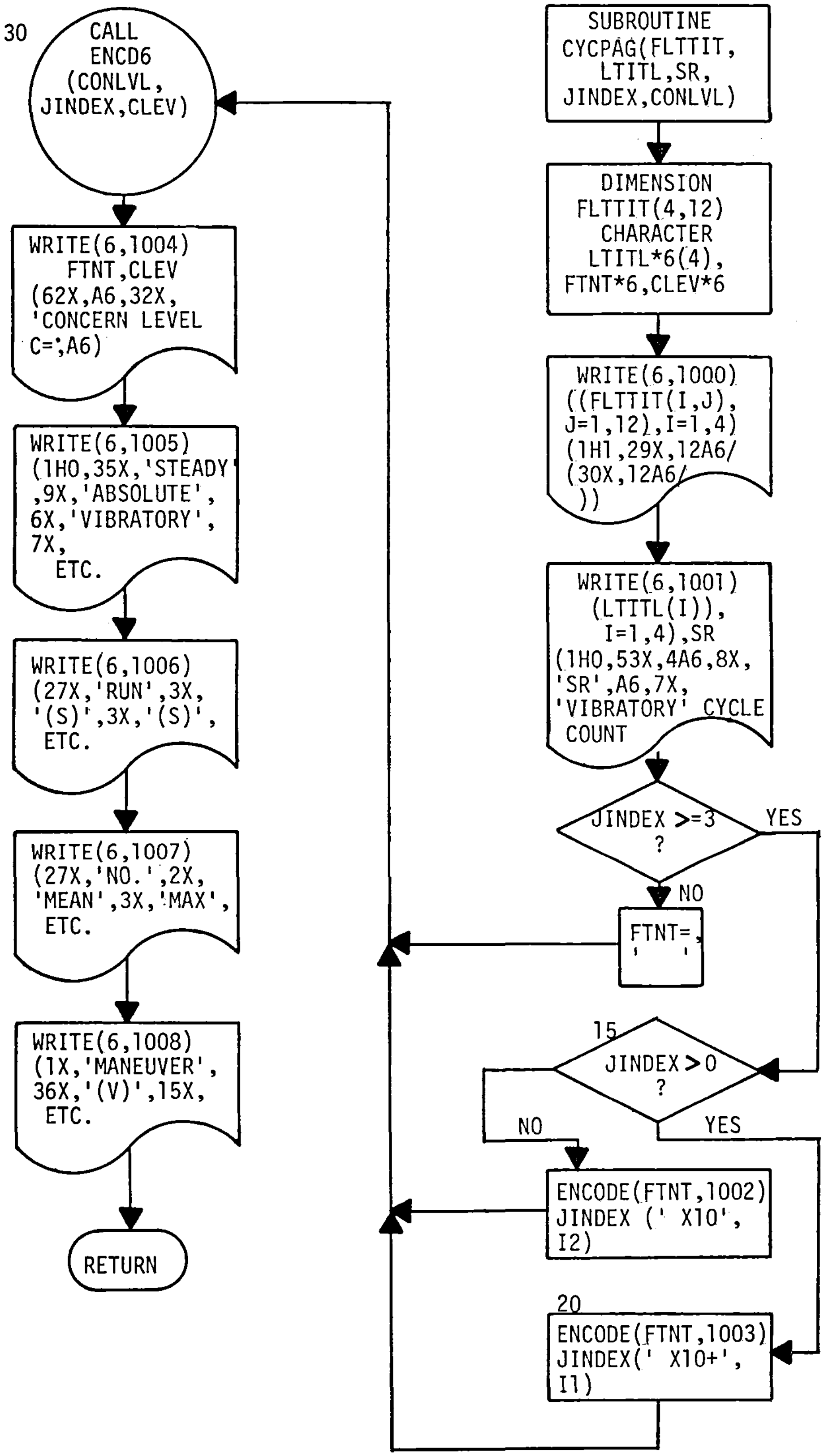
FLTTIT - FLIGHT TITLE
LTITL - LINE TITLE
JINDEX - INTEGER TO USE IN SUBROUTINE ENCD6 FOR ARRANGING THE OUTPUT STATISTICS INTO AN AESTHETIC FORM FOR PRINTING
CONLVL - CONCERN LEVEL

CALLING SEQUENCE

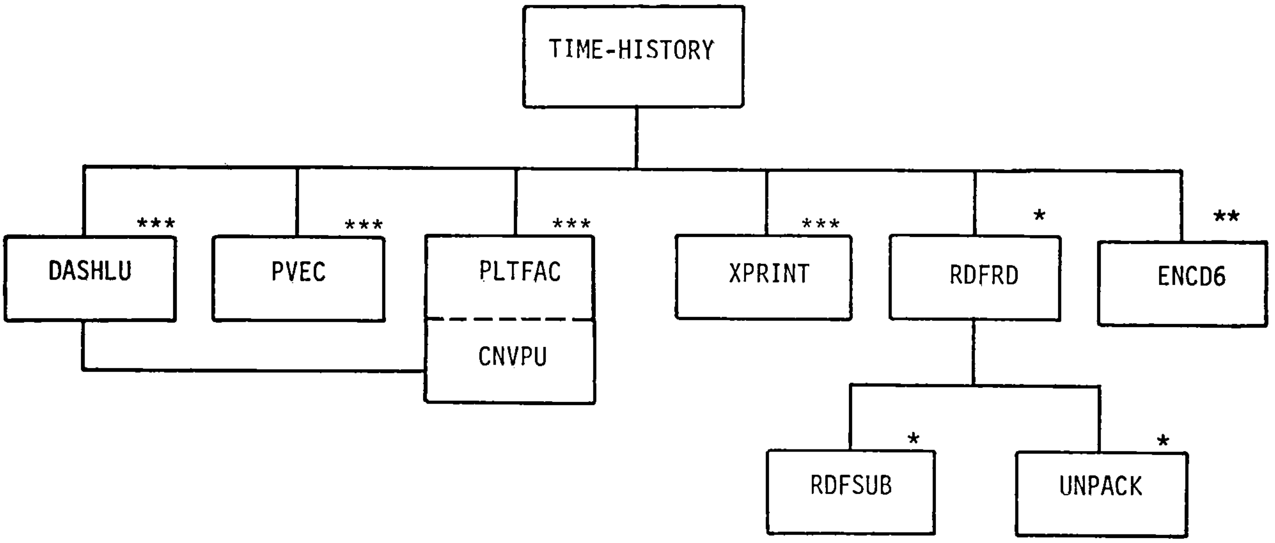
CALL ENCD6(CONLVL,JINDEX,CLEV)

SEE PROGRAM DOCUMENTATION ON FILE AT CAS FOR CARD FORMATS, FILE DESCRIPTIONS OR FOR ANY OTHER SPECIFIC INFORMATION.

****



SUBROUTINE CYCPAG(FLTTIT, LTITL,SR, JINDEX,CONLVL)
DIMENSION FLTTIT(4,12) CHARACTER LTITL*6(4), FTNT*6,CLEV*6
WRITE(6,1000) ((FLTTIT(I,J), J=1,12),I=1,4) (1H1,29X,12A6/ (30X,12A6/))
WRITE(6,1001) (LTTTL(I)), I=1,4),SR (1H0,53X,4A6,8X, 'SR',A6,7X, 'VIBRATORY' CYCLE COUNT
JINDEX >=3 ?
YES
NO
FTNT=' '
15
JINDEX > 0 ?
NO
YES
ENCODE(FTNT,1002) JINDEX (' X10', I2)
20
ENCODE(FTNT,1003) JINDEX(' X10+', I1)
30
CALL ENCD6 (CONLVL, JINDEX,CLEV)
WRITE(6,1004) FTNT,CLEV (62X,A6,32X, 'CONCERN LEVEL C=',A6)
WRITE(6,1005) (1H0,35X,'STEADY' ,9X,'ABSOLUTE', 6X,'VIBRATORY', 7X, ETC.
WRITE(6,1006) (27X,'RUN',3X, '(S)',3X,'(S)', ETC.
WRITE(6,1007) (27X,'NO.',2X, 'MEAN',3X,'MAX', ETC.
WRITE(6,1008) (1X,'MANEUVER', 36X,'(V)',15X, ETC.
RETURN



* See Sub-program RDFRD for Comments and Flowcharts

** See Program DATASK for Comments and Flowcharts

*** Comments and Flowcharts not available for these modules

HIERARCHY CHART For THPLOT PROGRAM

TIMHIS TIME-HISTORY - RSRA PLOT PROGRAM
TIMHIS NASA WALLOPS VERSION OF 8-15-77

LANGUAGE FORTRAN V (FORTY)

MACHINE - GE-625

METHOD - PROGRAM USES CALCOMP SUBROUTINES ON FILE IN SYSTEM TO COMMUNICATE WITH PLOTTER HARDWARE. TIMHIS OUTPUTS A STANDARD LABELLED TAPE AT A DENSITY OF 556BPI.

PURPOSE - TO PLOT RSRA DATA

DESCRIPTION OF TASK INPUT CARDS

GENERAL TITLE CARD

COLS.	FORMAT	DESCRIPTION
2-73	12A6	THIS TITLE IS PLACED AT THE TOP OF ALL PLOT PAGES.

PLOT ROW LABELS CARD

COLS.	FORMAT	DESCRIPTION
5	I1	RT. ADJUSTED PGF NO.(=1,9)
10	I1	RT. ADJUSTED ROW NO.(=1,4)
29-46	3A6	18 CHARACTER BOT LINE OF ROW LABEL(RITE-MOST)
11-28	3A6	18 CHARACTER TOP LINE OF ROW LABEL(LEFT-MOST)
29-46	3A6	18 CHARACTER BOT LINE OF ROW LABEL(RITE-MOST)

A CARD WITH A ZERO IN COL. 5 TERMINATES READING OF THESE CARDS
MAXIMUM NUMBER OF ROW LABEL CARDS =9 PGF"S * 4 ROWS = 36 CARDS

PARAMETER SPECIFICATION CARDS(2 CARDS/PARAMETER)

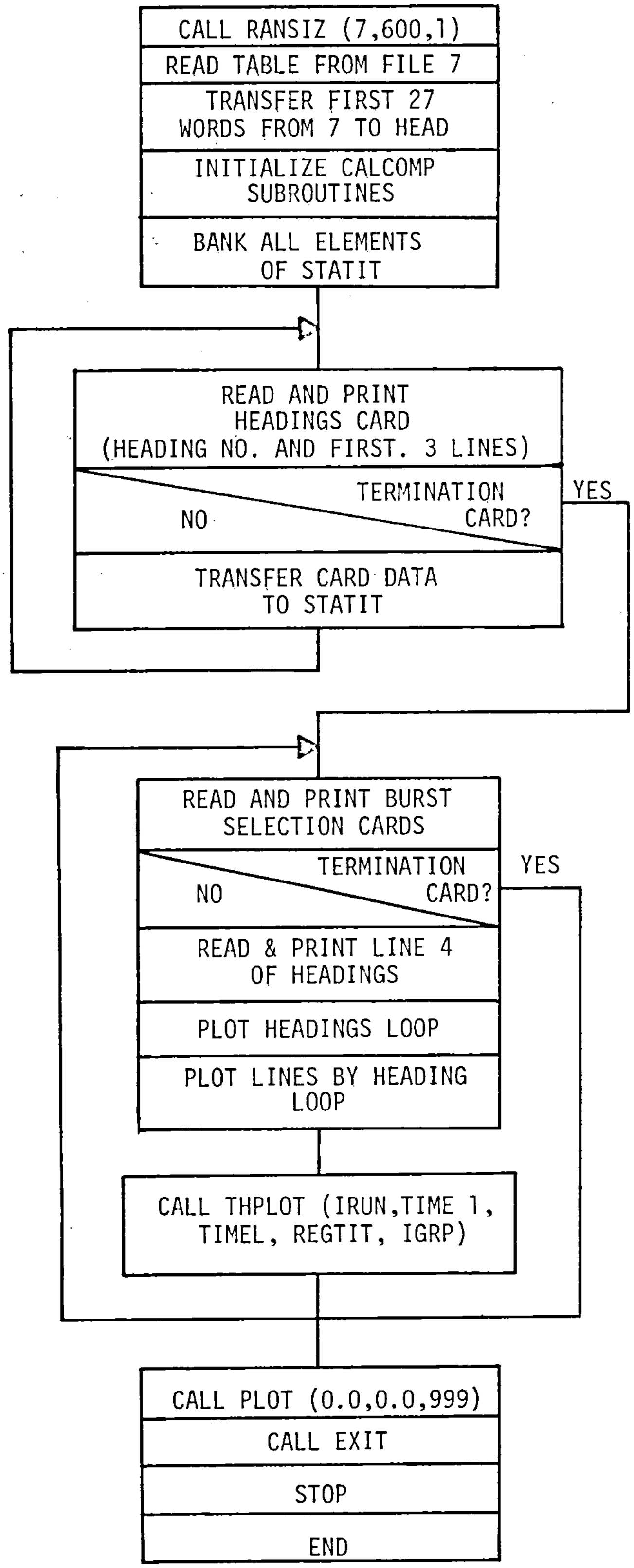
COLS.	FORMAT	DESCRIPTION
CARD NO.1		
1- 8	A8	LEFT ADJUSTED MNEMONIC OF PARAMETER
9	A1	PRE-PROCESSING CODE LETTER
10-11	BLANK	
12-14	A3	SAMPLE RATE CODE
15-18	BLANK	
19-36	3A6	18 CHARACTER AXIS LABEL (TOP LINE)
37-54	3A6	18 CHARACTER AXIS LABEL (BOT LINE)
55-58	A4	4 CHAR. DIRECTION NOTE FOR TOP OF SCALE
59-62	A4	4 CHAR. DIRECTION NOTE FOR BOT OF SCALE
63-64	I2	PARAMETERS PLOT POSITION ON PGF NO. 1
65-66	I2	PARAMETERS PLOT POSITION ON PGF NO. 2
67-68	I2	PARAMETERS PLOT POSITION ON PGF NO. 3
69-70	I2	PARAMETERS PLOT POSITION ON PGF NO. 4
71-72	I2	PARAMETERS PLOT POSITION ON PGF NO. 5
73-74	I2	PARAMETERS PLOT POSITION ON PGF NO. 6
75-76	I2	PARAMETERS PLOT POSITION ON PGF NO. 7
77-78	I2	PARAMETERS PLOT POSITION ON PGF NO. 8
79-80	I2	PARAMETERS PLOT POSITION ON PGF NO. 9

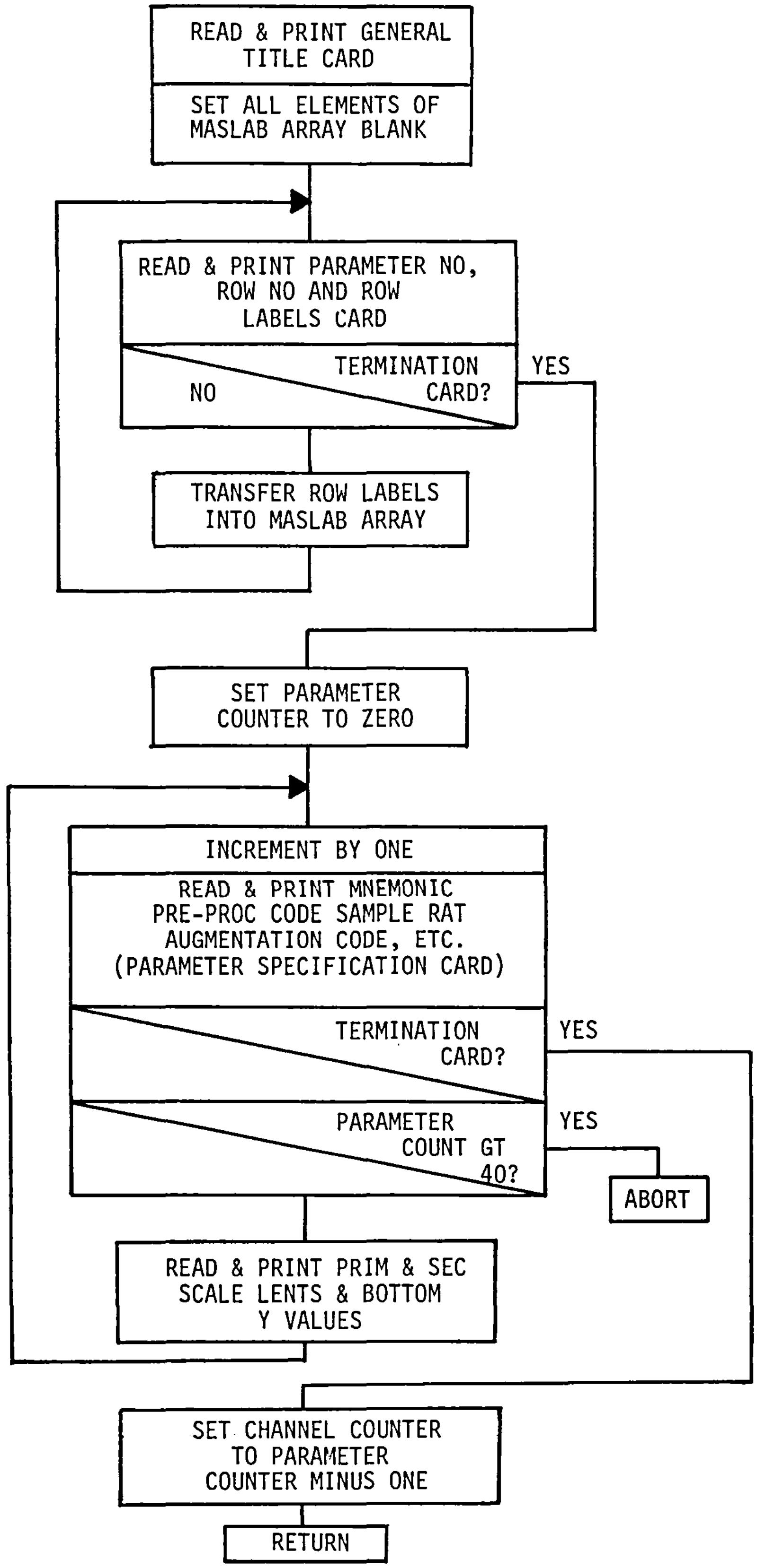
```
      PARAMETER PLOT POSITIONS ARE
      ROW     COL        POSITION NO      TWO PARAMETERS MUST NOT
       1       1            1             OCCUPY THE SAME POSITION
       1       2            2             ON A PGF.
       1       3            3
       2       1            4
       2       2            5
       2       3            6
       3       1            7
       3       2            8
       3       3            9
       4       1           10
       4       2           11
       4       3           12
CARD 2
 1-10     F10        RANGE OF PRIMARY SCALE IN E.U. (BOT TO TOP)
11-20     F10        VALUE AT BOTTOM OF PRIM. SCALE
21-30     F10        RANGE OF SECONDARY SCALE IN E. U. (BOT TO TOP)
31-40     F10        VALUE AT BOTTOM OF SECON. SCALE.
41-42      I2   *    SCALE PRECISION INDEX NO.
43-44      I2   **   OPTION FLAG TO REFERENCE(BIAS) ALL VALUES TO THE
                *          A PLUS SIGN OR NONE INDICATE INTEGER SCALING.
                           A NEG  SIGN INDICATES SCALES LT 0
                           A NEG N INDICATES N DECIMAL PLACE ACCURACY.
                **   FIRST NON-NULL VALUE. FLAG IS ON IF IT=1
                     UP TO 40 PARAMETERS MAY BE DEFINED FOR PLOTTING.
       PRIMARY SCALE WILL BE USED IF 80 PRCT. OF THE DATA CAN BE
       CONTAINED WITHIN IT. IF 20 PRCT OF THE DATA EXCEEDS THE PRIM
       SCALE,THEN THE SECOND. WILL BE USED. (20 PCNT OF DATA POINTS)
    A CARD WITH BLANKS IN COLS. 1-8 WILL TERMINATE READING OF THESE
    CARDS.


                     HEADING LABEL CARDS

COLS.   FORMAT                        DESCRIPTION
   5      I1         HEADING NO
 6-13     A8         LINE 1 CHARACTER STRING
14-21     A8         LINE 2 CHARACTER STRING
22-29     A8         LINE 3 CHARACTER STRING
                     FROM ONE TO EIGHT HEADINGS MAY BE IDENTIFIED
                     ANY MAY BE SKIPPED
    A ZERO IN COL. 5 END READING HEADING LABEL CARDS.


                     BURST SELECTION CARDS   (2 CARDS PER BURST)

COLS.   FORMAT                        DESCRIPTION
CARD 1
 1- 5     I5         BURST NO TO BE PROCESSED
 6-10     I5         PGF NO TO BE USED FOR THIS BURST
11-20     F10        START TIME OF PLOTS IN SECONDS INTO THE EVENT
21-30     F10        STOP  TIME OF PLOTS IN SECONDS INTO THE EVENT
31-78     8A6        MANEUVER TITLE (PLACED AT TOP OF PLOT)
79-80     I2         OPTION FLAG TO PRINT DATA TO BE PLOTTED(ON IF =1)
```

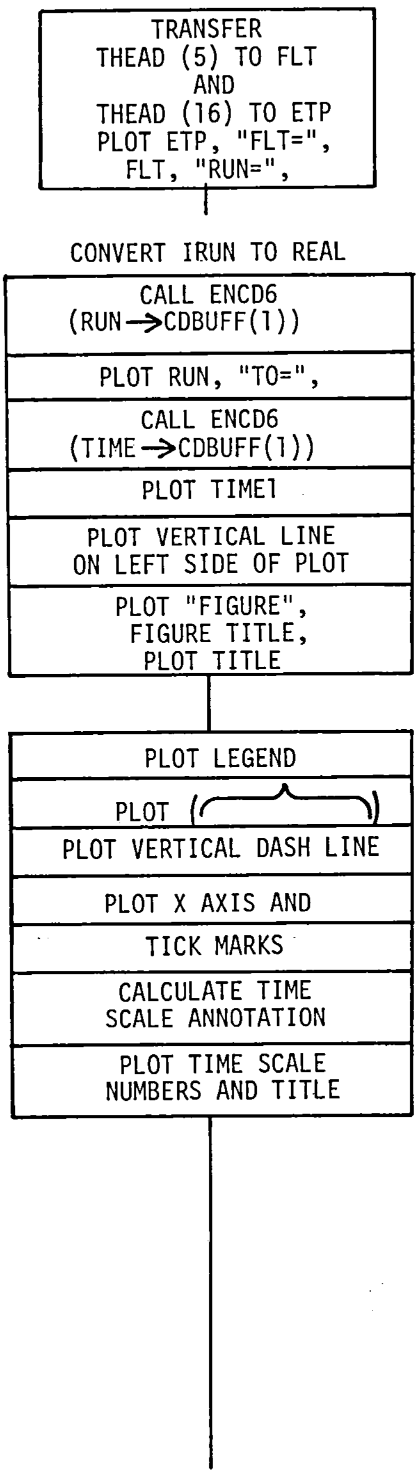
```
CARD 2
 1-10     A10       LINE 4 CHARACTER STRING HEADING 1
11-20     A10       LINE 4 CHARACTER STRING HEADING 2
21-30     A10       LINE 4 CHARACTER STRING HEADING 3
31-40     A10       LINE 4 CHARACTER STRING HEADING 4
41-50     A10       LINE 4 CHARACTER STRING HEADING 5
51-60     A10       LINE 4 CHARACTER STRING HEADING 6
61-70     A10       LINE 4 CHARACTER STRING HEADING 7
71-80     A10       LINE 4 CHARACTER STRING HEADING 8
    A CARD WITH A ZERO IN COL. 5 ENDS READING ALL CARDS.


        INPUT VARIABLES
        CARD INPUTS
 ALLPLT *6 (12)    72 CHARACTER TITLE FOR DISPLAY AT TOP OF ALL PLOTS
 BOT       (40,2) 2 POSSIBLE SCALE BOTTOM VALUES FOR ALL PARAMS DEFINED
 CDBUFF *6 (14)    LOCAL ENCODE/DECODE BUFFER FOR ARRANGING TEXTS
 IBNAME *8 (40)    MNEMONICS        OF ALL PARAMETERS DEFINED
 IBPP   *1 (40)    PRE PROCESS CODES OF ALL PARAMETERS DEFINED
 IBSR   *6 (40)    SAMPLE RATES     OF ALL PARAMETERS DEFINED
 JDELTA    (40)    FLAG TO PLOT VALUE MINUS FIRST SAMPLE
 JINDEX    (40)    PRECISION CODE FOR EACH PARAMETER DEFINED
 KGRP      (40,9) POSITION INDEX NO /PGF/PARAM
 MASLAB *6 (6,9,4)MAIN AND SUB TITLES FOR EACH ROW
 REGTIT *6 (8)     48 CHARACTER REGIME TITLE
 SCNOTE *6 (40,2) (2) 4 CHAR DIRECTION PLOT SCALE NOTES
 SLP       (40,2)  2 POSSIBLE SCALE RANGES
 YTITL  *6 (6,40) 36 CHARACTER PARAMETER TITLE   (MAIN + SUB)
 IGRP    PLOT GROUP FORMAT NO BEING EXECUTED
 IRUN    RUN NUMBER BEING PLOTTED
 TIME1   TIME INTO RUN AT WHICH DATA PRESENTATION IS TO BEGIN
 TIMEL   TIME INTO RUN AT WHICH DATA PRESENTATION IS TO END


        VARIABLES CALCULATED WITHIN PROGRAM
 ACNAME *8 (2)     MNEMONIC         OF PARAMETER BEING PROCESSED + TIME
 ACPP   *1 (2)     PRE PROCESS CODE OF PARAMETER BEING PROCESSED + TIME
 ACSR   *6 (2)     SAMPLE RATE      OF PARAMETER BEING PROCESSED + TIME
 DX        (6)     LOCAL PEN INCREMENTS IN X DIRECTION
 DY        (6)     LOCAL PEN INCREMENTS IN Y DIRECTION
 LEGEND *6 (3,2)   TEXT FOR "SOLID LINE","SHORT DASH","LONG DASH"
 OUTBUF    (232)   ACCESS BUFFER (116 FRAMES OF DATA/PARAM + TIME)
 PCTBD     (2)     0/0 OF SCALE UTILIZATION FOR EACH POSSIBLE PLOT SCALE
 PCTGD     (2)     0/0 OF SCALE UTILIZATION FOR EACH POSSIBLE PLOT SCALE
 TBASE     (3)     TIME SCALE MULTIPLIERS (1.5, 3.0, 6.0)
 TOS       (8)     TABLE OF SUBSCRIPT VECTOR USED BY RDFSUB + RDFRD
 DATFST  FIRST NON-NULL VALUE OF THE PARAMETER BEING PLOTTED
 DATMIN  MINIMUM DATA VALUE   OF THE PARAMETER BEING PLOTTED
 DATMAX  MAXIMUM DATA VALUE   OF THE PARAMETER BEING PLOTTED
 IBS     BEGINNING FRAME NO. IN SEGMENTED ACCESS OF RUN
 ICOL    COLUMN INDEX NO.
 IES     ENDING FRAME NO. IN SEGMENTED ACCESS OF RUN
 IF1ACT  FRAME NO. INTO BURST WHERE PLOT WILL BEGIN FOR THIS PARAMETER
 IFLACT  FRAME NO. INTO BURST WHERE PLOT WILL END   FOR THIS PARAMETER
 IPTR1   SUBSCRIPT OF OUTBUF FOR GIVEN FRAME OF PARAMETER BEING PLOTTED
 IPTR    SUBCCRIPT OF OUTBUF FOR GIVEN FRAME OF TIME
 IROW    ROW INDEX NO (1-4)
```

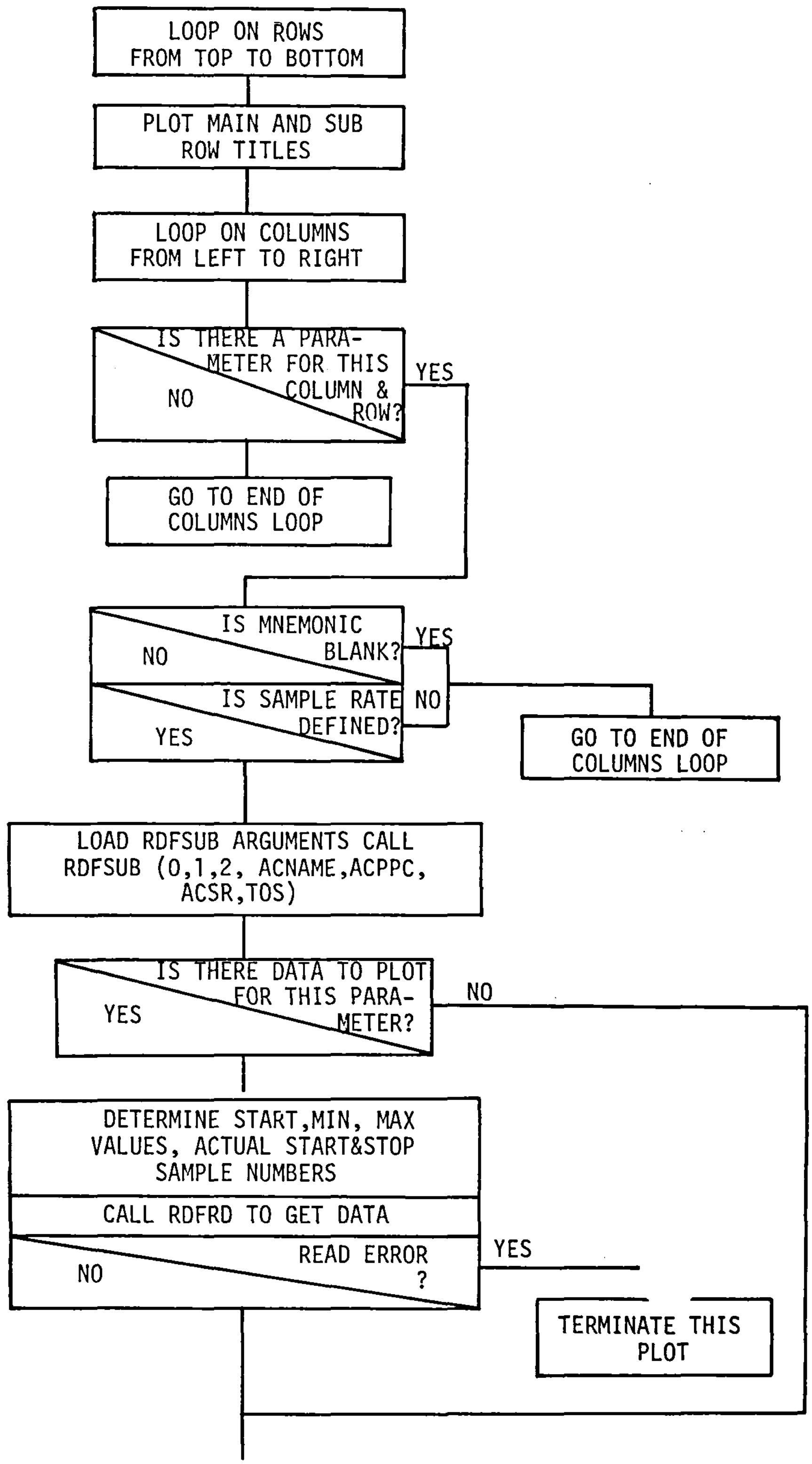
```
ITPCD   PRECISION CODE INDEX FOR TIME SCALE
LCHAN   PLOT-POSITION INDEX NO.(1-12)
LEN     THE NUMBER OF FRAMES THAT CAN BE ACCESSED AT A TIME
LL      PLOT SCALE FLAG =1,PRIMARY, =2,SECONDARY
NCHAND  NO OF PARAMETERS DEFINED IN ENTRY TO PRIMEP
NFRM    NO OF FRAMES ACTUALLY ACCESSED DURING GIVEN CALL TO RDFRD
NGD     NO OF NON-NULL FRAMES AVAILABLE FOR THIS PARAMETER
PMN     MINIMUM PARAMETER VALUE PERMITTED ON PLOT SURFACE
PMX     MAXIMUM PARAMETER VALUE PERMITTED ON PLOT SURFACE
POR     PARAMETER VALUE AT BOTTOM EDGE OF GRID (Y=0.0 CM)
SBOTX   SCALE VALUE AT BOTTOM END
SEGDN   LENGHT IN CM OF DASH LINE TO BE DRAWN
SLPACT  PLOT SCALE CHOSEN FOR PARAMETER IN ENG. UNITS/CM
STOPX   SCALE VALUE AT TOP END
SV      LOCAL VARIABLE USED IN GENERATION OF TIME SCALE
TMN     MINIMUM TIME VALUE PERMITTED ON PLOT SURFACE
TMX     MAXIMUM TIME VALUE PERMITTED ON PLOT SURFACE
TOR     TIME VALUE AT PLOT ORIGIN (X=0.0 CM)
TSLP    CHOSEN PLOT RATE FOR TIME SCALE (SECS/CM)
TSPAN   REQUESTED TIME SPAN = (TIMEL-TIME1)
TSPAN1  CHOSEN LENGHT OF TIME SCALE IN SECS
XOFFS1  X POSITION IN CM FROM ORIGIN FOR A GIVEN COLUMN Y AXIS
YOFFS1  Y POSITION


        OUTPUT
        PARAMETERS OUTPUT BY PROGRAM

OUTBUF(232),  MAY BE PRINTED ON LISTING
X,Y,XOFF1,YOFF1   PEN POSITIONS


    SUBROUTINES USED BY TIMHIS

        PLOTS - A CALCOMP ROUTINE TO INITIALIZE CALCOMP PACKAGE
             CALLING SEQUENCE
             CALL PLOTS(ARG1,ARG2,ARG3)
                  ARG1-IBUF-NAME OF OUTPUT BUFFER
                  ARG2-1000-SIZE OF IBUF
                  ARG3-50-OUTPUT TAPE FILE (STD FORMAT)

        FACTOR - A CALCOMP ROUTINE TO SCALE PLOT POSITIONS
             CALLING SEQUENCE
             CALL FACTOR(ARG)
                  ARG-0.3937-CONVERTS PEN DISPLACEMENTS FROM METRIC TO
                             INCHES FOR METRIC PAPER. INPUTS ARE METRIC

        SYMBOL-A CALCOMP ROUTINE TO PLOT ALPHANUMERIC INFORMATION
             CALLING SEQUENCE
             CALL SYMBOL(ARG1,ARG2,ARG3,ARG4,ARG5,ARG6)
                  ARG1-   - X START POSITION
                  ARG2-   - Y START POSITION
                  ARG3-   - SIZE OF CHARACTERS
                  ARG4-   - ALPHANUMERICS TO PLOT
                  ARG5-   - HORIZONTAL OR VERTICAL ALIGNMENT
                            0.0 FOR HORIZONTAL, 90.0 FOR VERTICAL
                  ARG6      NUMBER OF CHARACTERS TO BE PLOTTED
```

```
PLOT - A CALCOMP ROUTINE TO PLOT LINE DATA
     CALLING SEQUENCE
     CALL PLOT(ARG1,ARG2,ARG3)
          ARG1-  -X POSITION TO MOVE PEN TO
          ARG2-  -Y POSITION TO MOVE PEN TO
          ARG3-  -PEN POSITION CODE,2=PEN DOWM,3=PEN UP.

PLTFAC - ROUTINE TO CONVERT E. U. TO PEN POSITIONS
     CALLING SEQUENCE
     CALL PLTFAC(TSLP,SLPACT,TOR,0.0,TMX,PMN,PMX)
      SEE DESCRIPTION OF VARIABLES ABOVE

PVEC - ROUTINE TO PLOT SOLID LINE
     CALLING SEQUENCE
     CALL PVEC(OUTBUF(NFRM+1),OUTBUF(1),NFRM)
      SEE DESCRIPTION OF VARIABLES ABOVE

DASHLU - ROUTINE TO PLOT DASHED LINES
     CALLING SEQUENCE
     CALL DASHLU(OUTBUF(NFRM+1),OUTBUF(1),NFRM,SEGDN,0.075)
      SEE DESCRIPTION OF VARIABLES ABOVE

RANSIZ,RDFSUB,RDFRD,ENCD6 ARE DOCUMENTED ELSEWHERE IN THE
RSRA PACKAGE.
```

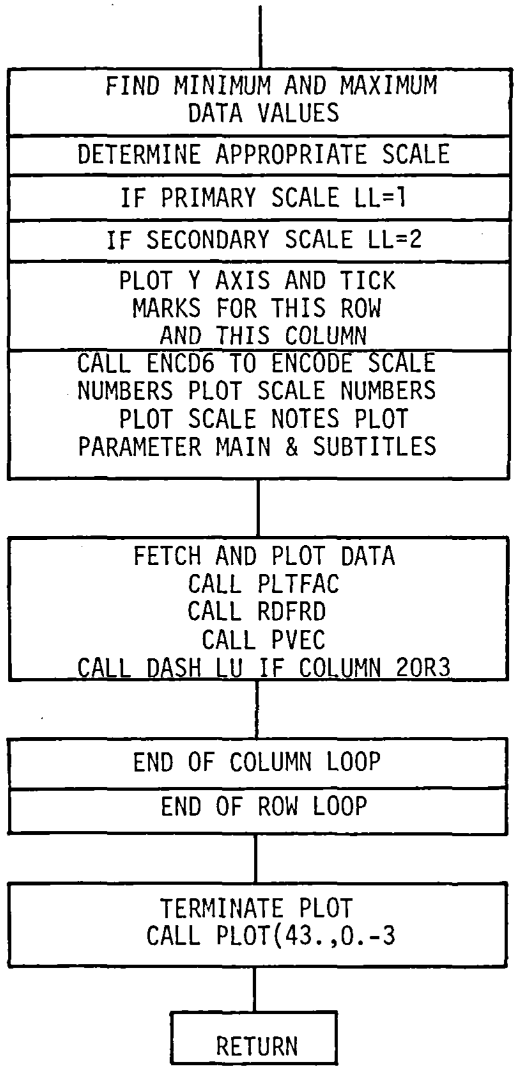

CALL RANSIZ (7,600,1)
READ TABLE FROM FILE 7
TRANSFER FIRST 27 WORDS FROM 7 TO HEAD
INITIALIZE CALCOMP SUBROUTINES
BANK ALL ELEMENTS OF STATIT
READ AND PRINT HEADINGS CARD (HEADING NO. AND FIRST. 3 LINES)
TERMINATION CARD?
YES
NO
TRANSFER CARD DATA TO STATIT
READ AND PRINT BURST SELECTION CARDS
TERMINATION CARD?
YES
NO
READ & PRINT LINE 4 OF HEADINGS
PLOT HEADINGS LOOP
PLOT LINES BY HEADING LOOP
CALL THPLOT (IRUN,TIME 1, TIMEL, REGTIT, IGRP)
CALL PLOT (0.0,0.0,999)
CALL EXIT
STOP
END

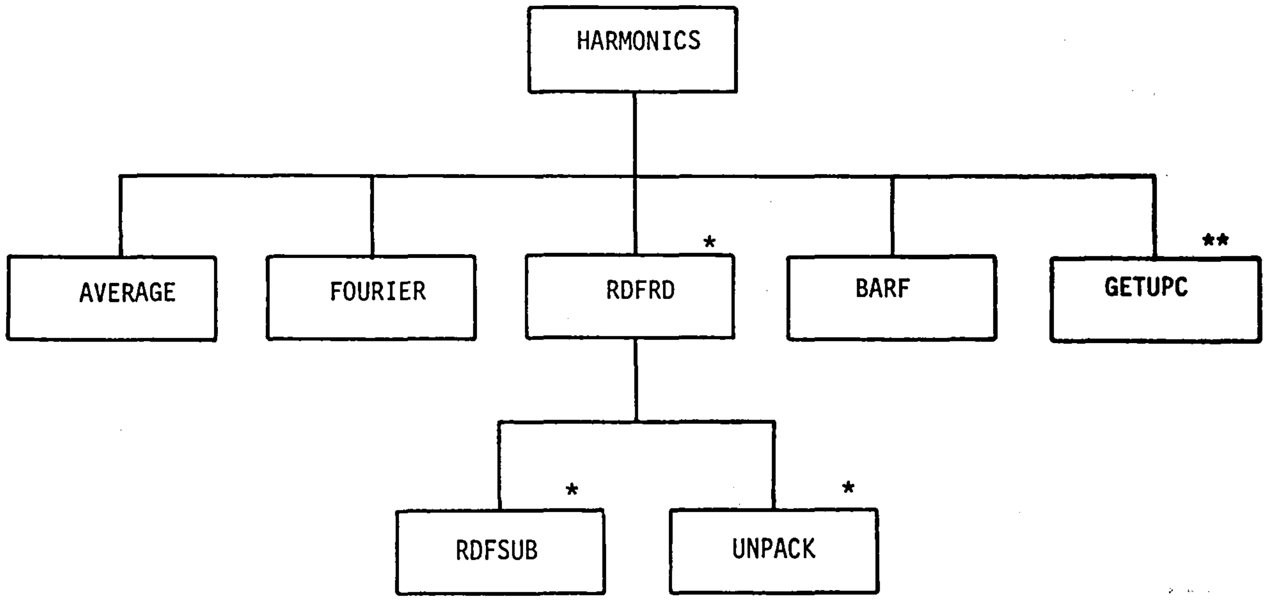

READ & PRINT GENERAL TITLE CARD
SET ALL ELEMENTS OF MASLAB ARRAY BLANK
READ & PRINT PARAMETER NO, ROW NO AND ROW LABELS CARD
TERMINATION CARD?
YES
NO
TRANSFER ROW LABELS INTO MASLAB ARRAY
SET PARAMETER COUNTER TO ZERO
INCREMENT BY ONE
READ & PRINT MNEMONIC PRE-PROC CODE SAMPLE RAT AUGMENTATION CODE, ETC. (PARAMETER SPECIFICATION CARD)
TERMINATION CARD?
YES
PARAMETER COUNT GT 40?
YES
ABORT
READ & PRINT PRIM & SEC SCALE LENTS & BOTTOM Y VALUES
SET CHANNEL COUNTER TO PARAMETER COUNTER MINUS ONE
RETURN


TRANSFER
THEAD (5) TO FLT
AND
THEAD (16) TO ETP
PLOT ETP, "FLT=",
FLT, "RUN=",
CONVERT IRUN TO REAL
CALL ENCD6
(RUN→CDBUFF(1))
PLOT RUN, "TO=",
CALL ENCD6
(TIME→CDBUFF(1))
PLOT TIME1
PLOT VERTICAL LINE
ON LEFT SIDE OF PLOT
PLOT "FIGURE",
FIGURE TITLE,
PLOT TITLE
PLOT LEGEND
PLOT
PLOT VERTICAL DASH LINE
PLOT X AXIS AND
TICK MARKS
CALCULATE TIME
SCALE ANNOTATION
PLOT TIME SCALE
NUMBERS AND TITLE



LOOP ON ROWS FROM TOP TO BOTTOM
PLOT MAIN AND SUB ROW TITLES
LOOP ON COLUMNS FROM LEFT TO RIGHT
IS THERE A PARAMETER FOR THIS COLUMN & ROW?
YES
NO
GO TO END OF COLUMNS LOOP
IS MNEMONIC BLANK?
YES
NO
IS SAMPLE RATE DEFINED?
NO
YES
GO TO END OF COLUMNS LOOP
LOAD RDFSUB ARGUMENTS CALL RDFSUB (0,1,2, ACNAME,ACPPC, ACSR,TOS)
IS THERE DATA TO PLOT FOR THIS PARAMETER?
NO
YES
DETERMINE START,MIN, MAX VALUES, ACTUAL START&STOP SAMPLE NUMBERS
CALL RDFRD TO GET DATA
READ ERROR ?
YES
NO
TERMINATE THIS PLOT


FIND MINIMUM AND MAXIMUM DATA VALUES
DETERMINE APPROPRIATE SCALE
IF PRIMARY SCALE LL=1
IF SECONDARY SCALE LL=2
PLOT Y AXIS AND TICK MARKS FOR THIS ROW AND THIS COLUMN
CALL ENCD6 TO ENCODE SCALE NUMBERS PLOT SCALE NUMBERS PLOT SCALE NOTES PLOT PARAMETER MAIN & SUBTITLES
FETCH AND PLOT DATA
CALL PLTFAC
CALL RDFRD
CALL PVEC
CALL DASH LU IF COLUMN 2OR3
END OF COLUMN LOOP
END OF ROW LOOP
TERMINATE PLOT
CALL PLOT(43.,0.-3
RETURN



* See Sub-Program RDFRD for Comments and Flowcharts

** See Program DATASK for Comments and Flowcharts

HIERARCHY CHART For HARMONIC ANALYSIS PROGRAM

************************** PROGRAM HARMONIC ANALYSIS *****************
02/22/78

**PROGRAM IDENTIFICATION

```
PROGRAM NAME = RSRA HARMONIC ANALISIS
PROGRAM NO.  =
RESEARCHER   = NORM MICHAUD (NASA WALLOPS)
PROGRAMMER   = GUS DOVI (COMPUTER SCIENCES CORP.)
ANALYST      = KEN LEWIS (SIKORSKY AIRCRAFT)

COMPUTER     = HW 625/635
MEMORY       = 14K
PERIPHERALS  = DISC SUBSYSTEM
               TAPE SUBSYSTEM
LANGUAGE     = HW 6000 FORTRAN (100 PERCENT)
NO. CARDS    = 476
```

***PURPOSE

RSRA HARMONIC ANALYSIS READS DATA FROM FILES(05);(07);(09);(35). THE DATA IS PROCESSED AND SCANNED FOR PROPER FORMAT. THE RAW DATA FILE IS READ. FROM THIS DATA THE FOURIER COEFFICIENTS ARE GENERATED AS WELL AS THE RESULTENT AMPLITUDES,PHASE ANGLES,AND STANDARD SAMPLE DEVIATION. FOURIER COEFFICIENTS ARE LIMITED TO 144.

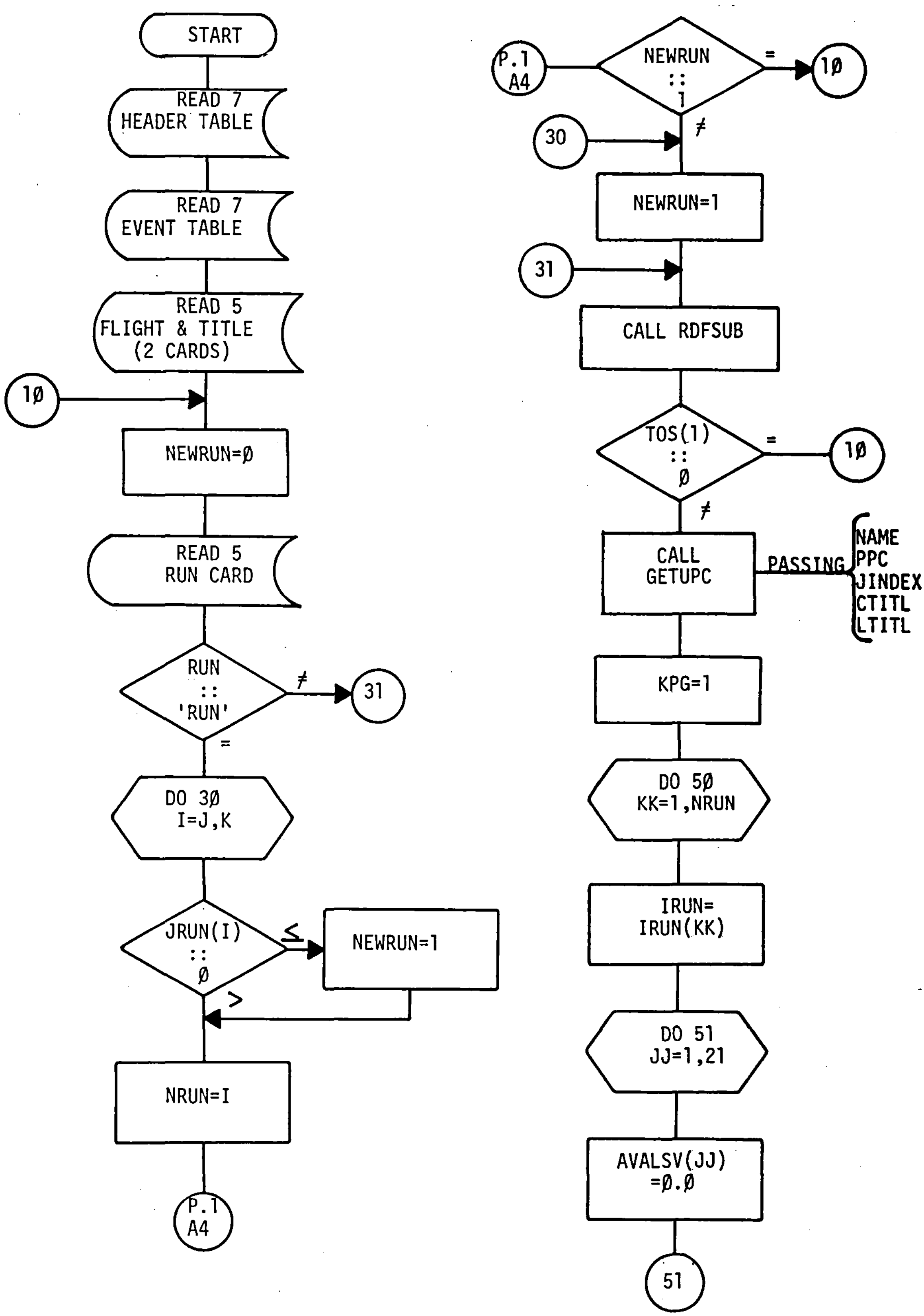
***I/O CONFIGURATION

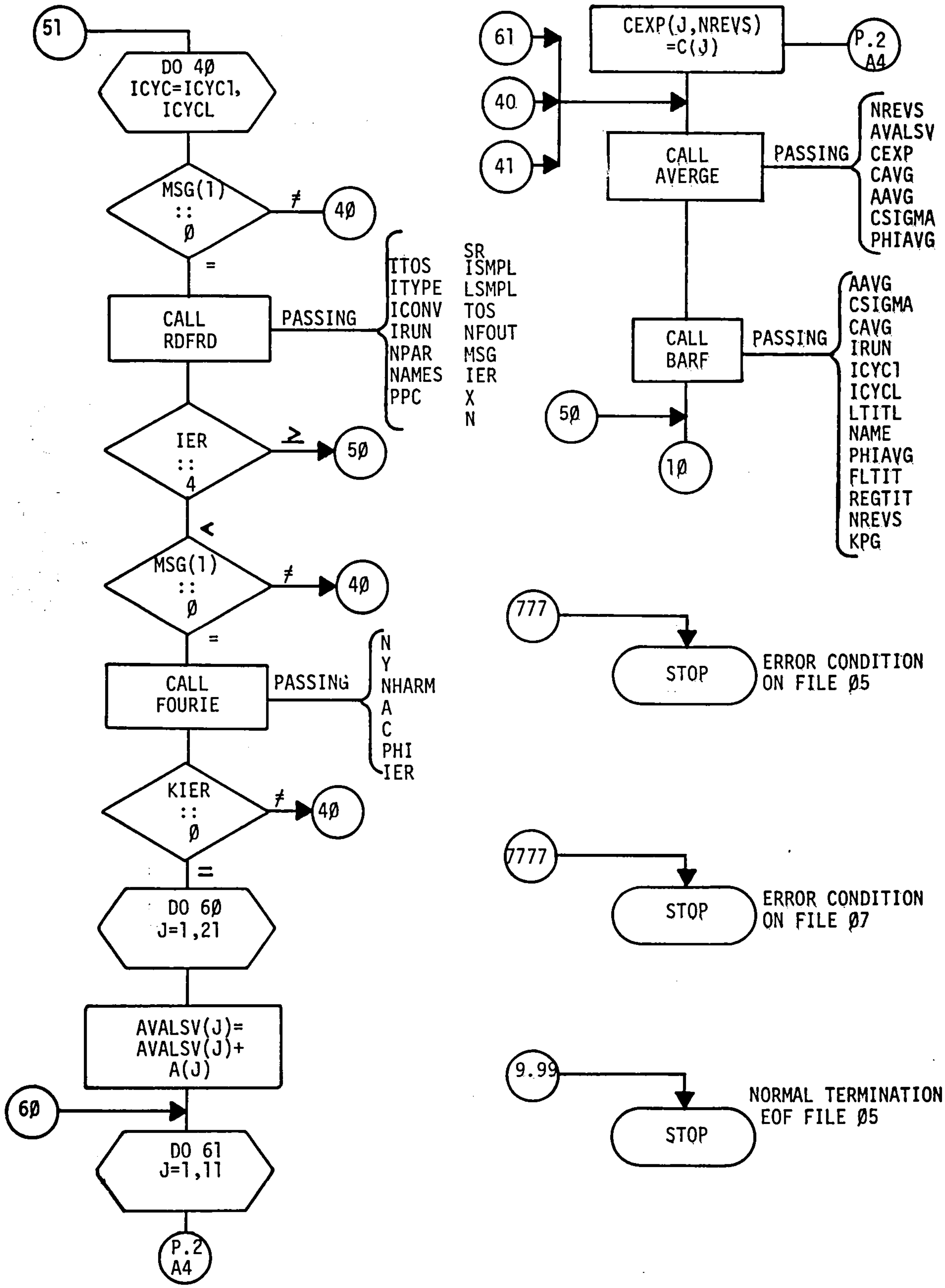
UNIT NO.	DEVICE TYPE	DESCRIPTION
05	CRD	CONTAINS TITLES AND RUN DATA.
07	DSK	RANDOM FILE CONTAINING DATA TABLES.
09	DSK	STATISTICS FILE USED BY SUBROUTINE RDFRD
35	DSK	UNIVERSAL PARAMETER CATALOG.

**SUBPROGRAMS REQUIRD

RDFSUB
GETUPC
RDFRD
FOURIE
AVERGE
BARF

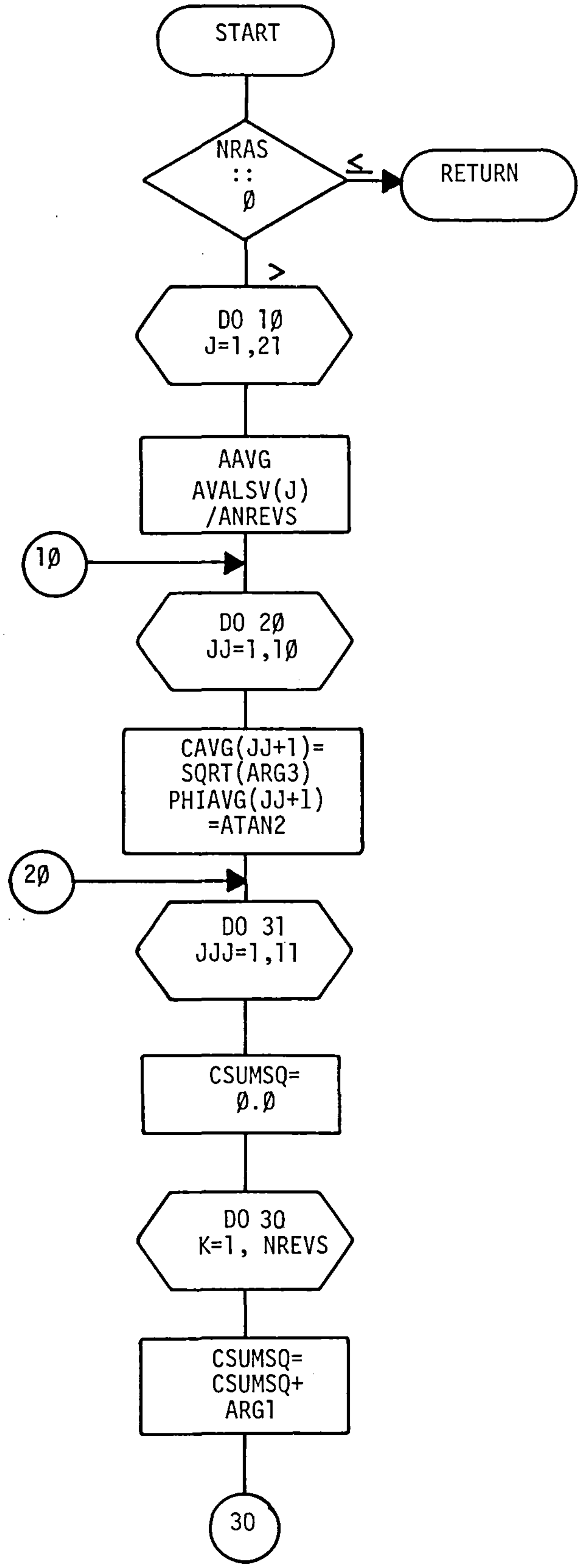
************************** PROGRAM HARMONIC ANALYSIS *****************

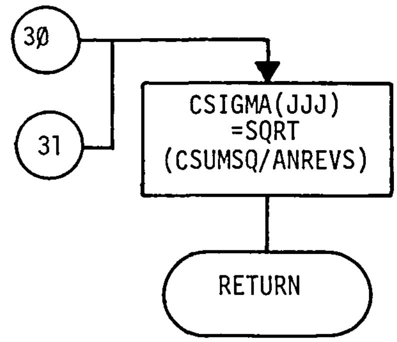

START
READ 7
HEADER TABLE
READ 7
EVENT TABLE
READ 5
FLIGHT & TITLE
(2 CARDS)
1Ø
NEWRUN=Ø
READ 5
RUN CARD
RUN
::
'RUN'
≠
31
=
DO 3Ø
I=J,K
JRUN(I)
::
Ø
≤
NEWRUN=1
>
NRUN=I
P.1
A4
P.1
A4
NEWRUN
::
1
=
1Ø
≠
30
NEWRUN=1
31
CALL RDFSUB
TOS(1)
::
Ø
=
1Ø
≠
CALL
GETUPC
PASSING
NAME
PPC
JINDEX
CTITL
LTITL
KPG=1
DO 5Ø
KK=1,NRUN
IRUN=
IRUN(KK)
DO 51
JJ=1,21
AVALSV(JJ)
=Ø.Ø
51


51
DO 4Ø
ICYC=ICYC1,
ICYCL
MSG(1)
::
Ø
≠
4Ø
=
CALL
RDFRD
PASSING
ITOS
ITYPE
ICONV
IRUN
NPAR
NAMES
PPC
SR
ISMPL
LSMPL
TOS
NFOUT
MSG
IER
X
N
IER
::
4
≥
5Ø
<
MSG(1)
::
Ø
≠
4Ø
=
CALL
FOURIE
PASSING
N
Y
NHARM
A
C
PHI
IER
KIER
::
Ø
≠
4Ø
=
DO 6Ø
J=1,21
AVALSV(J)=
AVALSV(J)+
A(J)
6Ø
DO 61
J=1,11
P.2
A4
61
40
41
CEXP(J,NREVS)
=C(J)
P.2
A4
CALL
AVERGE
PASSING
NREVS
AVALSV
CEXP
CAVG
AAVG
CSIGMA
PHIAVG
CALL
BARF
PASSING
AAVG
CSIGMA
CAVG
IRUN
ICYC1
ICYCL
LTITL
NAME
PHIAVG
FLTIT
REGTIT
NREVS
KPG
5Ø
1Ø
777
STOP
ERROR CONDITION
ON FILE Ø5
7777
STOP
ERROR CONDITION
ON FILE Ø7
9.99
STOP
NORMAL TERMINATION
EOF FILE Ø5

```
************************* SUBROUTINE AVERGE *************************
                              02/22/78

**PROGRAM IDENTIFICATION

     PROGRAM NAME = AVERGE
     PROGRAM NO.  =
     RESEARCHER   = NORM MICHAUD (NASA WALLOPS)
     PROGRAMMER   = GUS DOVI (COMPUTER SCIENCES CORP.)
     ANALYST      = KEN LEWIS (SIKORSKY AIRCRAFT)

     COMPUTER     = HW 625/635
     MEMORY       =
     PERIPHERALS  = NONE
     LANGUAGE     = HW 6000 FORTRAN (100 PERCENT)
     NO. CARDS    = 37

**PURPOSE
     TO GENERATE THE AVERAGE FOURIER COEFICIENTS, RESULTANTS, PHASE
     ANGLES AND STANDARD DEVIATIONS OVER A RANGE OF (N-1)/2 HARMONICS.
     WHERE (N) IS THE NUMBER OF HARMONICS.


**INPUT

      NREVS      = INTEGER VALUE  (0 .LT. NREVS .LE. 20)
     AVALSV      = (A) COEFFICIENTS OUTPUT FROM SUBPROGRAM FOURIE.
                   THE (A) COEFFICIENTS ARE ACCUMULATED INTO THE
                   AVALSV ARRAY.  AVALSV IS DIMENSIONED TO (21).

     CEXP        = ARRAY DIMENSIONED (11,20), CONTAINS ACCUMULATED
                   RESULTANTS OF THE A(2*JJ) AND A(2*JJ+1) COEFFICIENTS
                   OUTPUT BY SUBPROGRAM FOURIE.

**OUTPUT

     CAVG        = ARRAY DIMENSIONED (11), CONTAINS THE AVERAGE
                   RESULTANT OF THE AVERAGE (A) AND (C) COEFFICIENTS.
                   WHERE
                   CAVG(JJ+1)  = SQRT(AAVG(2*JJ)**2+AAVG(2*JJ+1)**2)

     AAVG        = ARRAY DIMENSIONED (21), CONTAINS THE AVERAGE
                   (A) COEFFICIENTS INPUT VIA ARRAY AVALSV.

     CSIGMA      = ARRAY DIMENSIONED (11), CONTAINS AVERAGE STANDARD
                   DEVIATIONS.

     PHIAVG      = ARRAY DIMENSIONED (11), CONTAINS AVERAGE PHASE ANGLES
                   DEFINED BY THE A(2*JJ) AND A(2*JJ+1) COEFFICIENTS.
```

```
**RESTRICTIONS

     NONE

**SUBPROGRAMS REQUIRED

     NONE

************************** SUBROUTINE AVERAGE *************************
```





ARGUMENTS PASSED:

NREVS
AVALSV
CEXP
CAVG
AAVG
CSIGMA
PHIAVG

```
************************** SUBROUTINE FOURIE **************************

**PROGRAM IDENTIFICATION

     PROGRAM NAME = FOURIE
     PROGRAM NO.  =
     RESEARCHER   = NORM MICHAUD (NASA WALLOPS)
     PROGRAMMER   = GUS DOVI (COMPUTER SCIENCES CORP.)
     ANALYST      = KEN LEWIS (SIKORSKY AIRCRAFT)

     COMPUTER     = HW 625/635
     MEMORY       =
     PERIPHERALS  = NONE
     LANGUAGE     = HW 6000 FORTRAN (100 PERCENT)
     NO. CARDS    = 76

**PURPOSE
     TO DETERMINE THE FOURIER COEFFICIENTS FOR A DATA VECTOR OF 144
     VALUES.  THE PHASE ANGLE IS ALSO COMPUTED.


**INPUT

     N           = NUMBER OF VALUES IN INPUT VECTOR

     Y           = ARRAY (INPUT VECTOR) DIMENSIONED (144).

     NHARM       = NHARM IS SET EQUAL TEN (10) IN THE CALLING PROGRAM.
                   NHARM IS AN INTEGER SPECIFYING THE NUMBER OF
                   (C) VALUES DESIRED.  IF NHARM = 10 (C) SHOULD BE
                   DIMENSIONED (11) AND ARRAY (A) SHOULD BE DIMENSIONED
                   (21) IN THE CALLING PROGRAM.  NOTE ((1) =(A(1)/N) AND
                   C(I) = SQRT(A(J)**2+A(J+1)**2).


**OUTPUT
     A           = ARRAY OF FOURIER COEFFICIENTS.  THE SINE TERMS ARE
                   IN THE ODD SUBSCRIPTED ELEMENTS.  THE COSINE TERMS
                   ARE IN THE EVEN SUBSCRIPTED ELEMENTS

     C           =  ARRAY OF RESULTANT VECTORS.
                    C(N) = SQRT(A(N)**2 + A(N+1)**2)

     PHI         =  ARRAY OF PHASE ANGLES FOR USE WITH (C) VECTORS.

     IER         =  0 IMPLIES NO ERRORS.
                    1 IMPLIES (Y) VECTOR CONTAINS NULL VALUES.
```

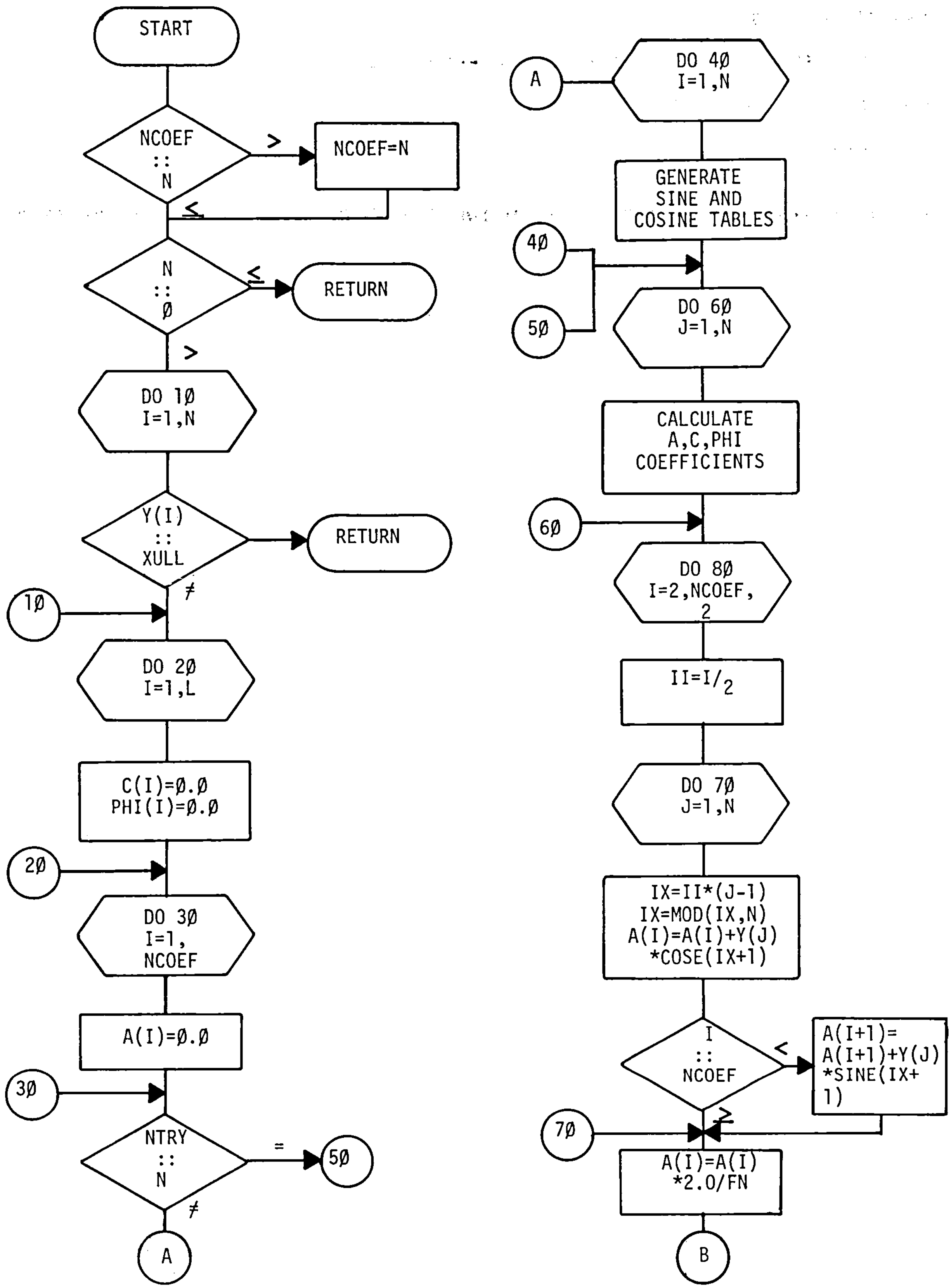
**RESTRICTIONS

Y INPUT VECTOR LIMITED TO 144 ELEMENTS.

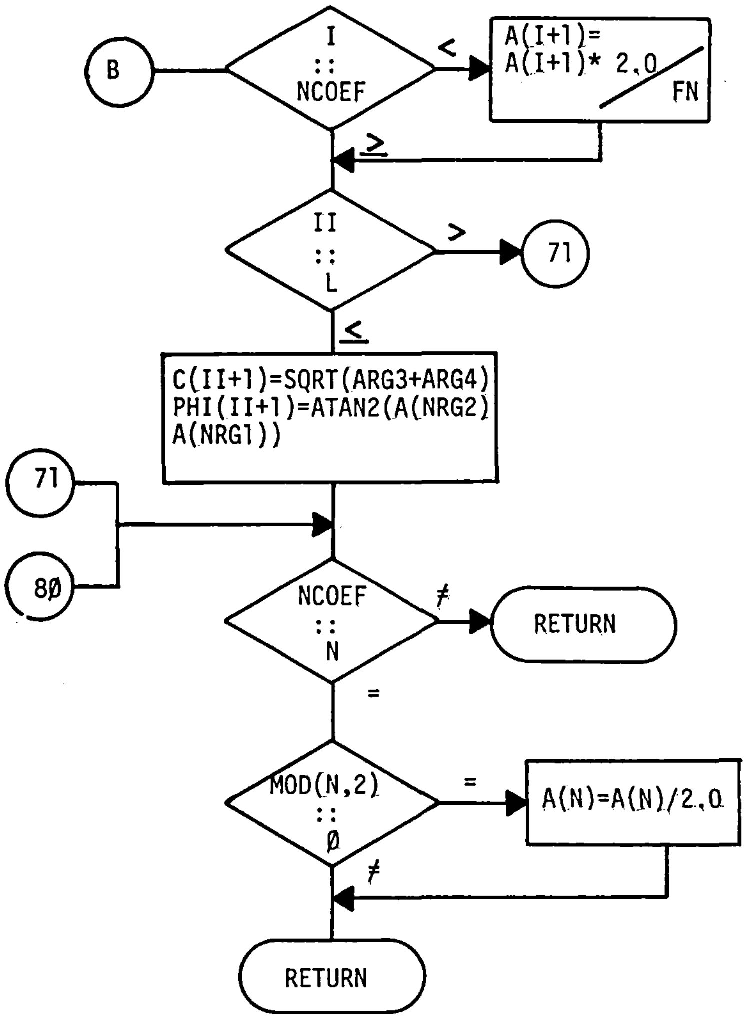
**SURPPOGRAMS REQUIRED

NONE

************************** SUBROUTINE FOURIE **************************



START
NCOEF :: N
>
NCOEF=N
≤
N :: Ø
≤
RETURN
>
DO 1Ø I=1,N
Y(I) :: XULL
RETURN
≠
1Ø
DO 2Ø I=1,L
C(I)=Ø.Ø PHI(I)=Ø.Ø
2Ø
DO 3Ø I=1, NCOEF
A(I)=Ø.Ø
3Ø
NTRY :: N
=
5Ø
≠
A
A
DO 4Ø I=1,N
GENERATE SINE AND COSINE TABLES
4Ø
5Ø
DO 6Ø J=1,N
CALCULATE A,C,PHI COEFFICIENTS
6Ø
DO 8Ø I=2,NCOEF, 2
II=I/2
DO 7Ø J=1,N
IX=II*(J-1) IX=MOD(IX,N) A(I)=A(I)+Y(J) *COSE(IX+1)
I :: NCOEF
<
A(I+1)= A(I+1)+Y(J) *SINE(IX+ 1)
≥
7Ø
A(I)=A(I) *2.0/FN
B

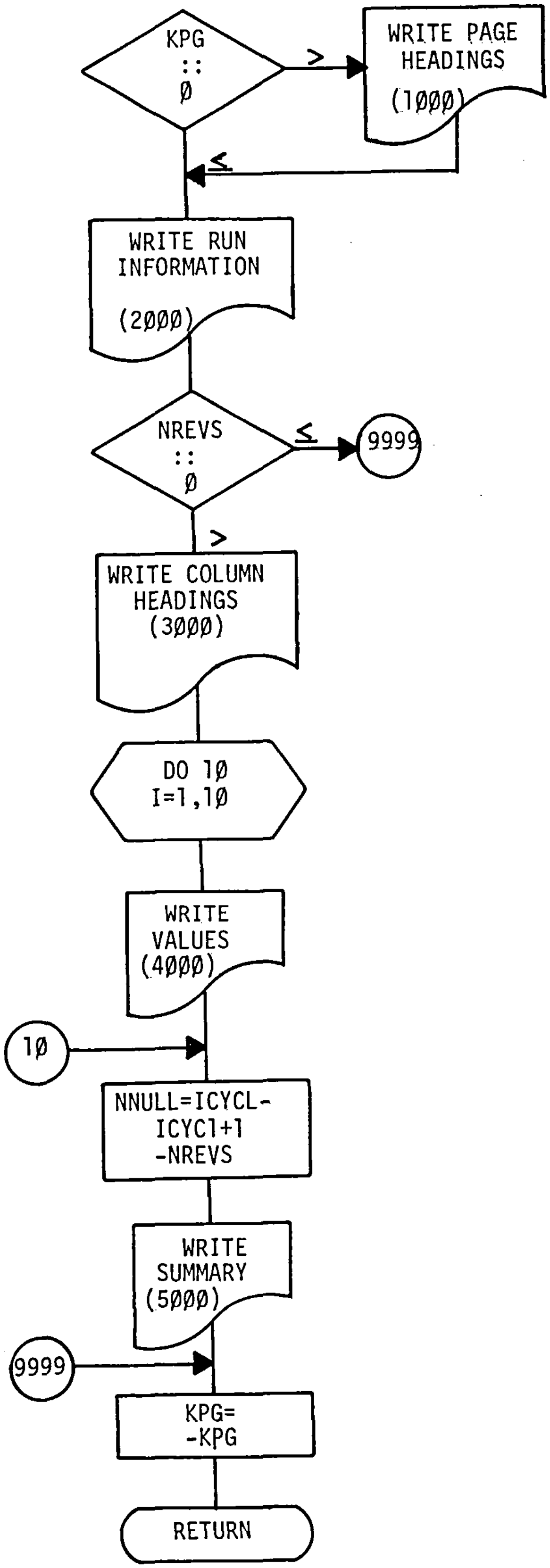


ARGUMENTS PASSED:

N
Y
NHARM
A
C
PHI
IER

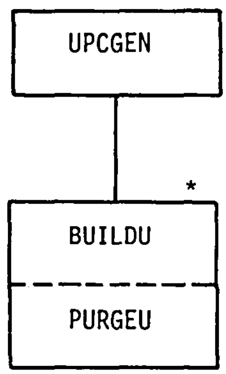
BARF

```
      SUBROUTINE BARF(AAVG,CSIGMA,CAVG,IRUN,ICYC1,ICYCL,LTITL,NAME
     X                ,PHIAVG,FLTIT,REGTIT,NREVS,KPG)
**** RESULTS OF THE HARMONIC ANALYSIS ARE GENERATED UNDER SPECIFIED
**** FORMATS.
```



ARGUMENTS PASSED:

AAVG
CSIGMA
CAVG
IRUN
ICYC1
ICYCL
LTITL
NAME
PHIAVG
FLTIT
REGTIT
NREVS
KPG



*Comments and Flowcharts not available for this module.

HIERARCHY CHART For UPCGEN PROGRAM

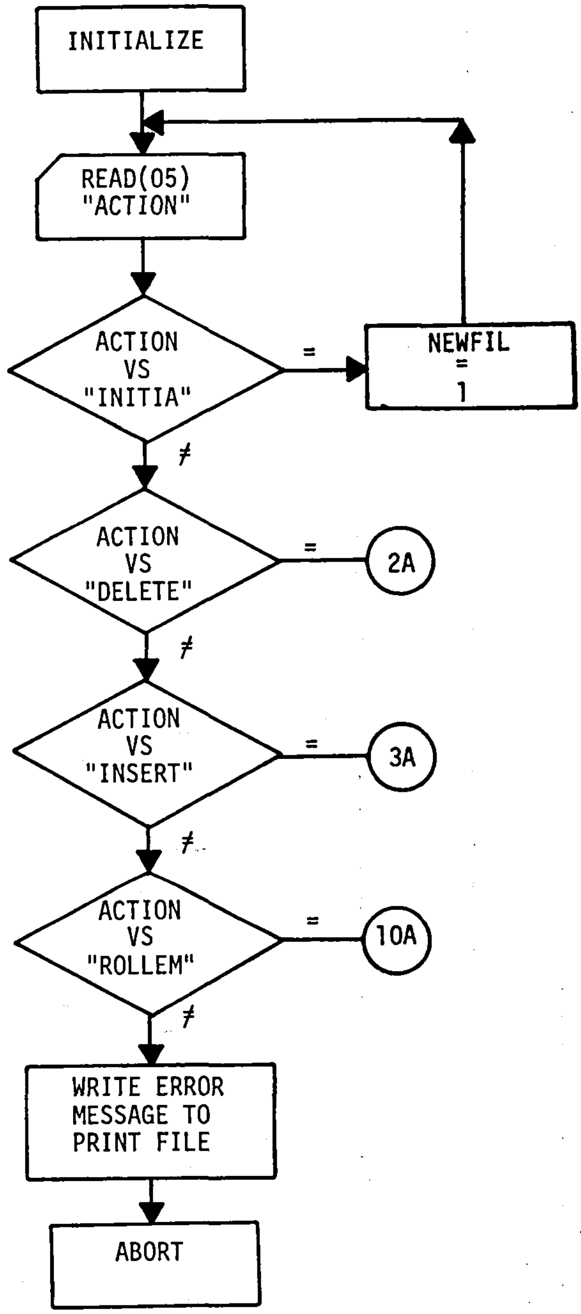
```
************************** SUBROUTINE UPCGEN **************************

  PROGRAM IDENTIFICATION

       PROGRAM NAME  - UPCGEN
       PROGRAM NO.   - 1.1.2333
       AUTHOR        - DAVID L. DAVIS

       COMPUTER      - HW 625/635
       MEMORY
       PERIPHERALS   -
       LANGUAGE      - HW 6000 FORTRAN

  PURPOSE

       CREAT AND MAINTAIN THE
       UNIVERSAL PARAMETER CATALOG
       FILE.

  METHOD

       A SET OF DATA CARDS ARE READ
       PRECEDED BY A DIRECTIVE CARD.
       THIS DIRECTIVE CARD INDICATES THE
       ACTION TO BE PERFORMED. THE ACTION
       MAY BE TO INITIALIZE THE FILE, INSERT
       PARAMETERS, OR DELETE PARAMETERS.
*******   UNIVERSAL PARAMETER CATALOG (UPC) FILE STRUCTURE    *****

       PERMANANT,RANDOM ACCESS DISC FILE

       2 TO 139 RECORDS

       280 WORDS PER RECORD

       2760 PARAMETER ENTRIES WHEN FILE = 139 RECORDS

*******   UPC RECORD DESCRIPTION   ********

       RECORD NO. 1 - TABLE OF CONTENTS

       WD 1   -  #UPC   #
       WD 2   -  INTEGER NO. OF PARAMETERS IN CATALOG
       WD 3,4-  CONCATENATED NAME AND PPC OF
          .      FIRST ENTRY IN FIRST CATALOG RECORD.
          .
          .
          .
       WD 279,280 - CONCATENATED NAME AND PPC OF
                    LAST ENTRY IN FILE.

       THE CONCATENATED NAME AND PPC OF LAST ENTRY IN FILE
       WILL IMMEDIATLY FOLLOW THE ITEM SPECIFYING THE LAST
       REQUIRED CATALOG RECORD. ITS POSITION IS THEREFORE A
       FUNCTION OF THE NO. OF PARAMETERS IN THE CATALOG AT
       A RATE OF 20 PARAMETERS PER CATALOG RECORD.
```

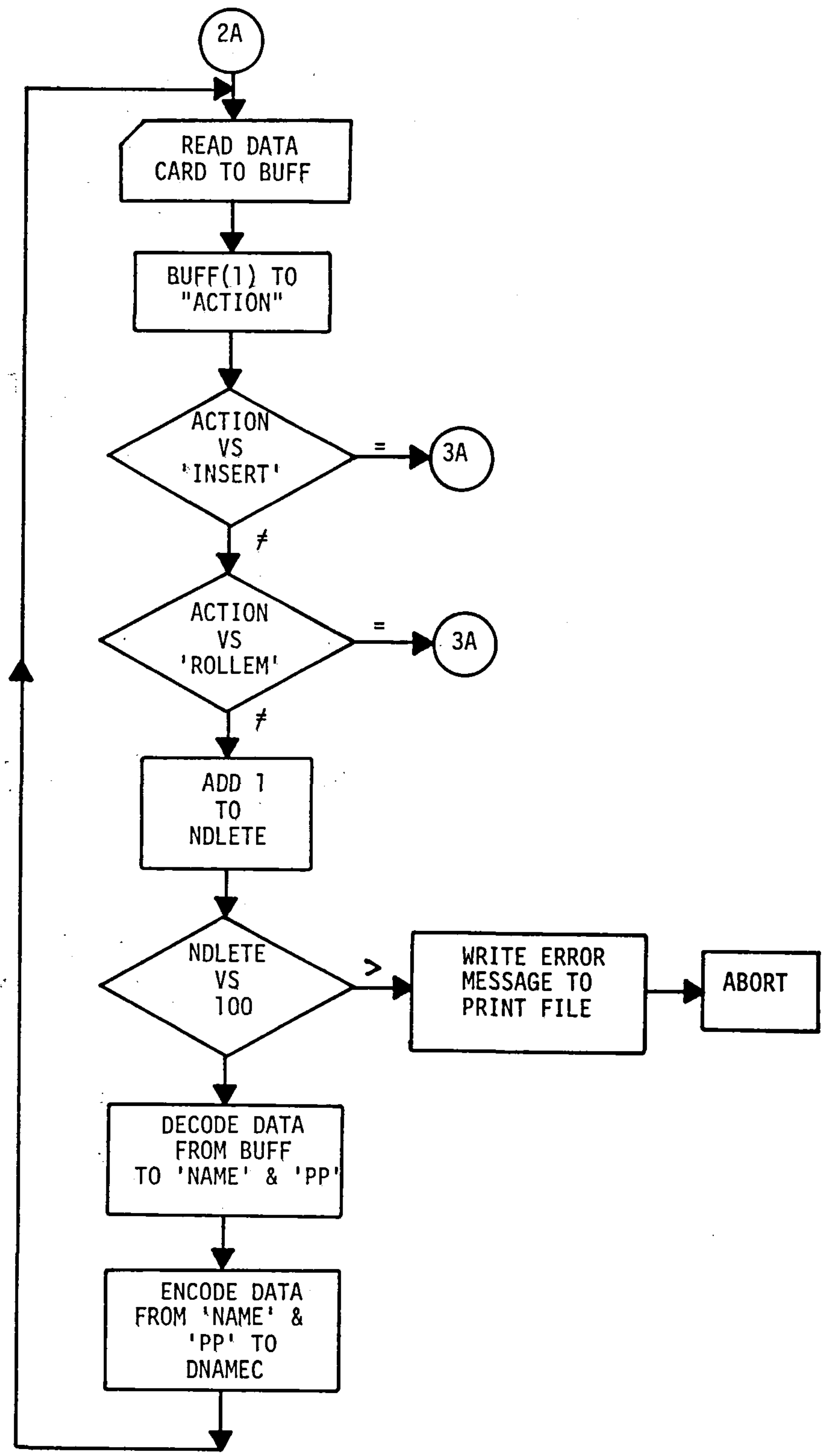
***** RECORDS NO. 2 - 139 *****

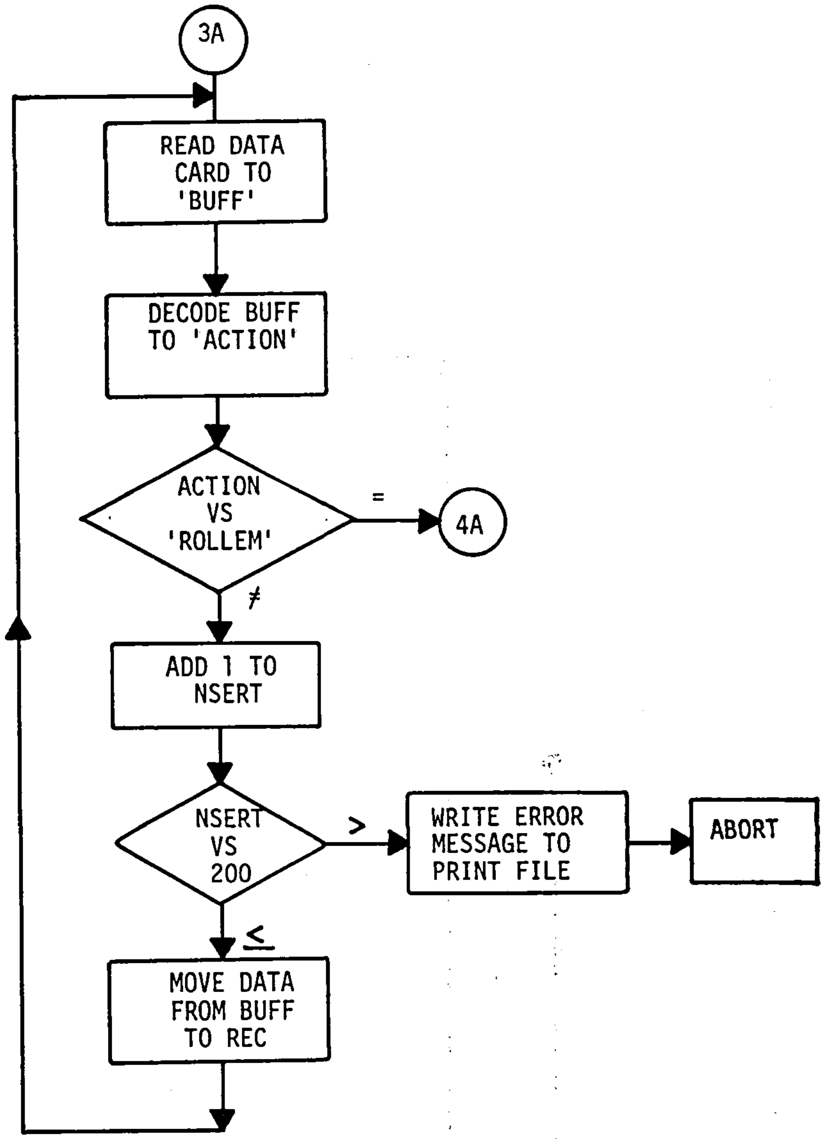
CATALOG RECORDS - 280 WORDS EACH (20 PARM) X (14 WDS/PARM)

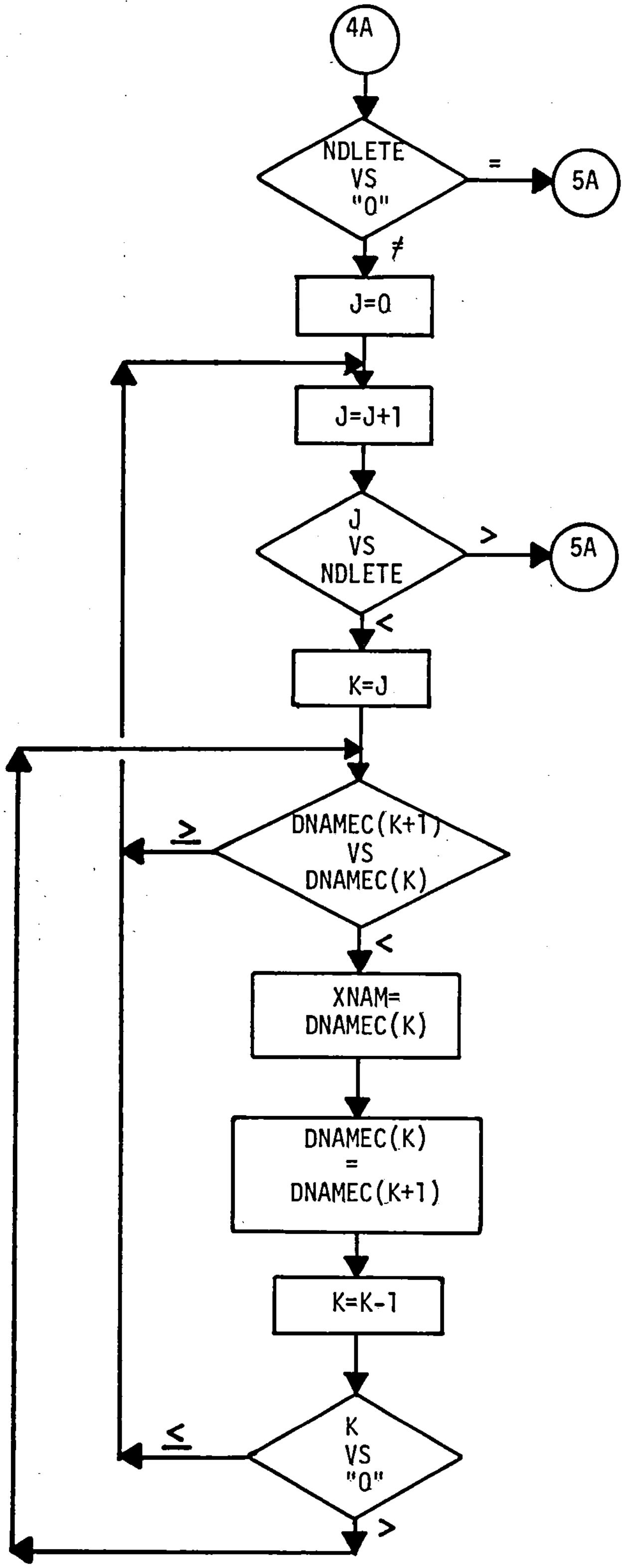
```
WD 1,2    NAME AND PPC  OF PARM NO. 1
WD 3      PARAMETER CATEGORY
WD 4      INTEGER PRECISION CODE INDEX
WD 5,10   6 WORD COLUMN TITLE
WD 11,14  4 WORD LINE TITLE

WD 15,16  NAME AND PPC  OF PARM NO. 2

 ETC
```

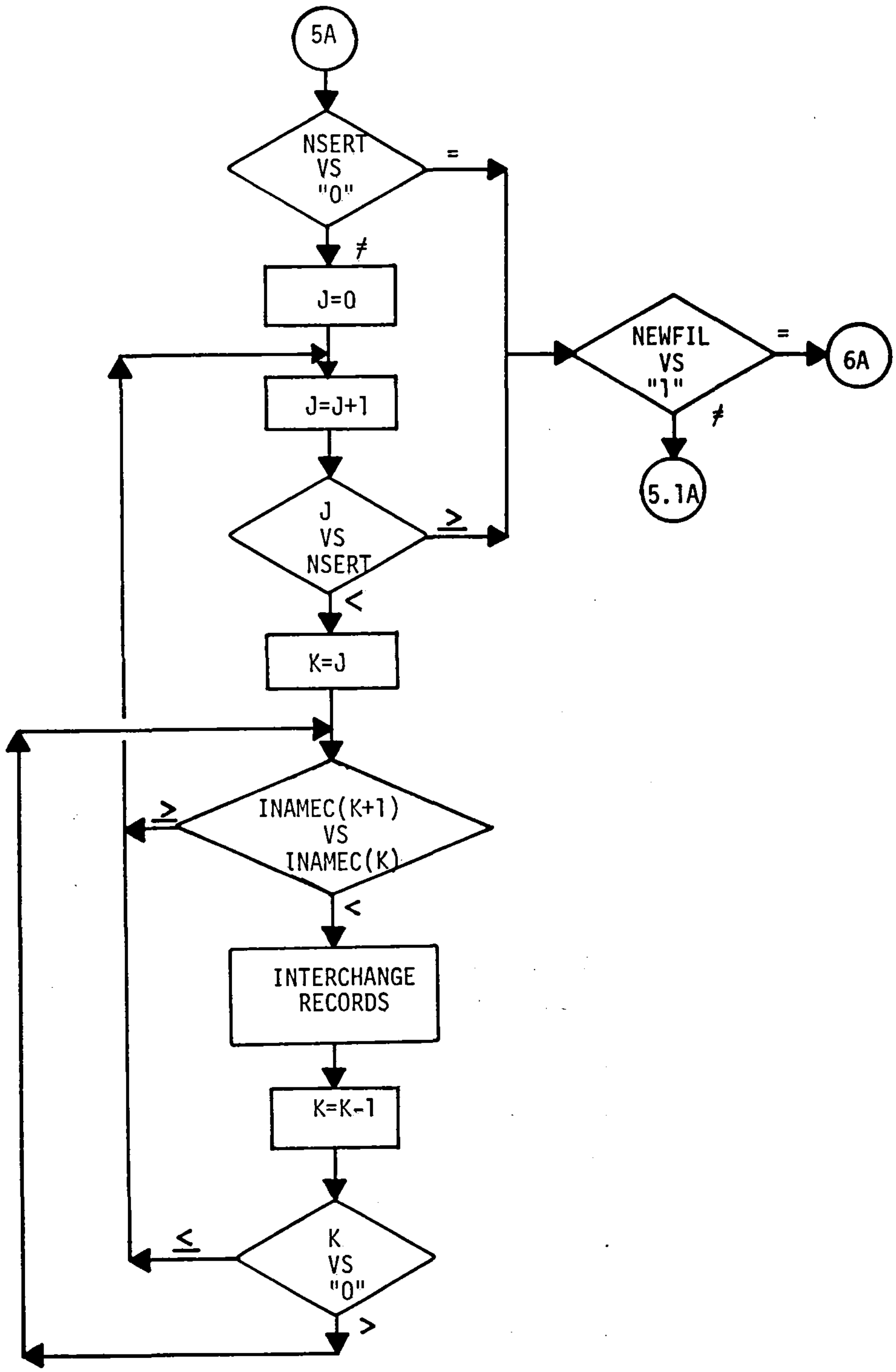
ALL PARAMETERS ARE ENTERED ALPHABETICALLY ACCORDING TO CONCATENATED NAME AND PPC.

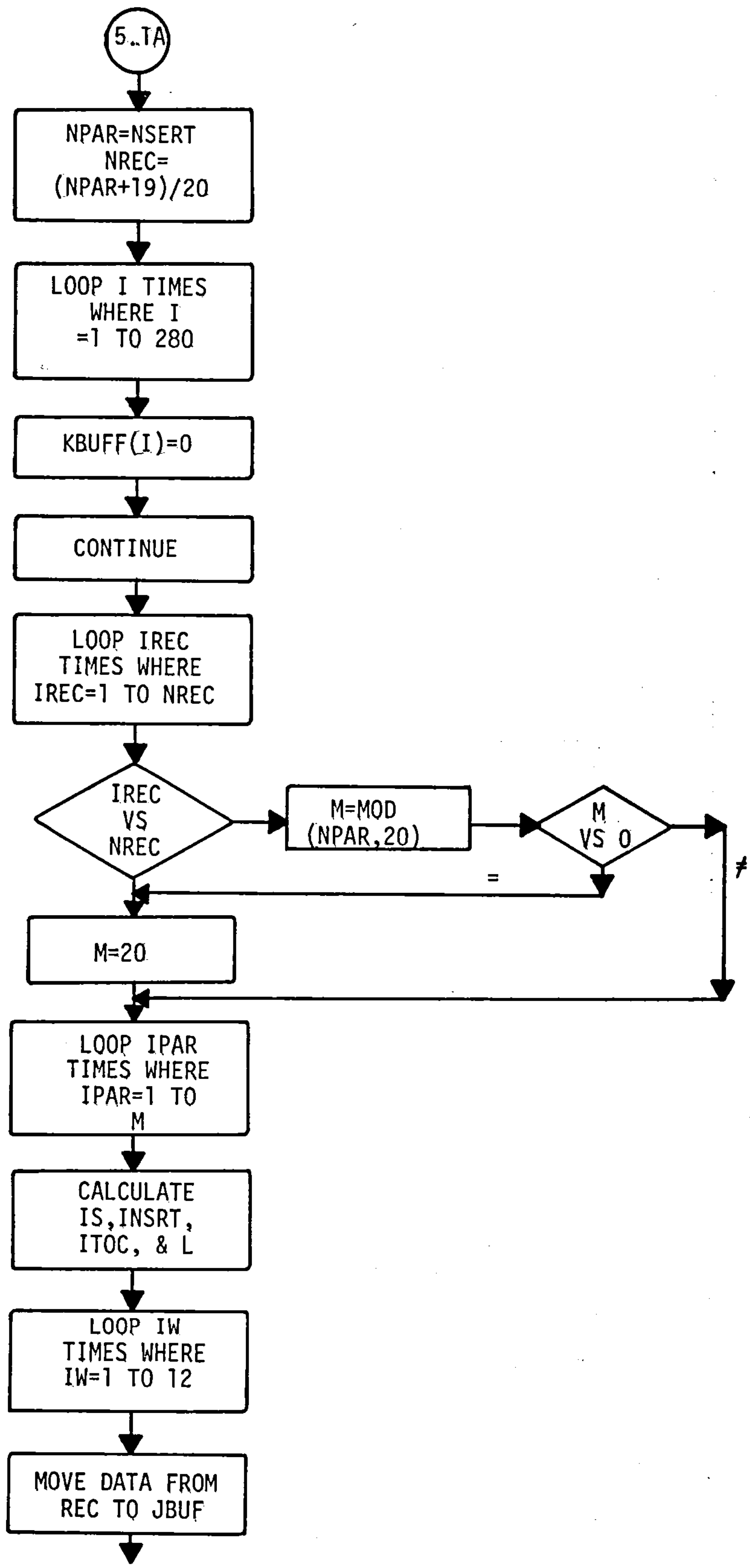

INITIALIZE
READ(05) "ACTION"
ACTION VS "INITIA"
=
NEWFIL = 1
≠
ACTION VS "DELETE"
=
2A
≠
ACTION VS "INSERT"
=
3A
≠
ACTION VS "ROLLEM"
=
10A
≠
WRITE ERROR MESSAGE TO PRINT FILE
ABORT

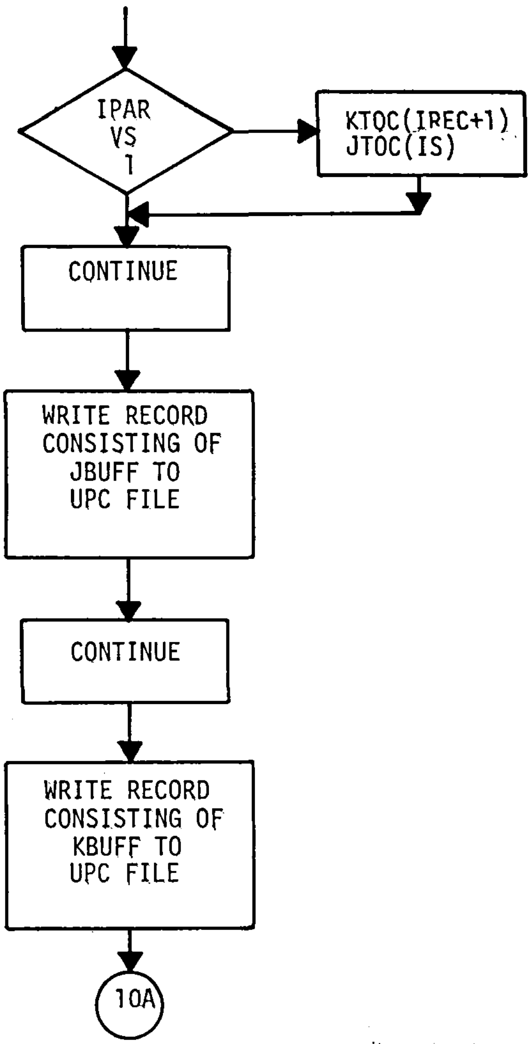

2A
READ DATA CARD TO BUFF
BUFF(1) TO "ACTION"
ACTION VS 'INSERT'
=
3A
≠
ACTION VS 'ROLLEM'
=
3A
≠
ADD 1 TO NDLETE
NDLETE VS 100
>
WRITE ERROR MESSAGE TO PRINT FILE
ABORT
DECODE DATA FROM BUFF TO 'NAME' & 'PP'
ENCODE DATA FROM 'NAME' & 'PP' TO DNAMEC

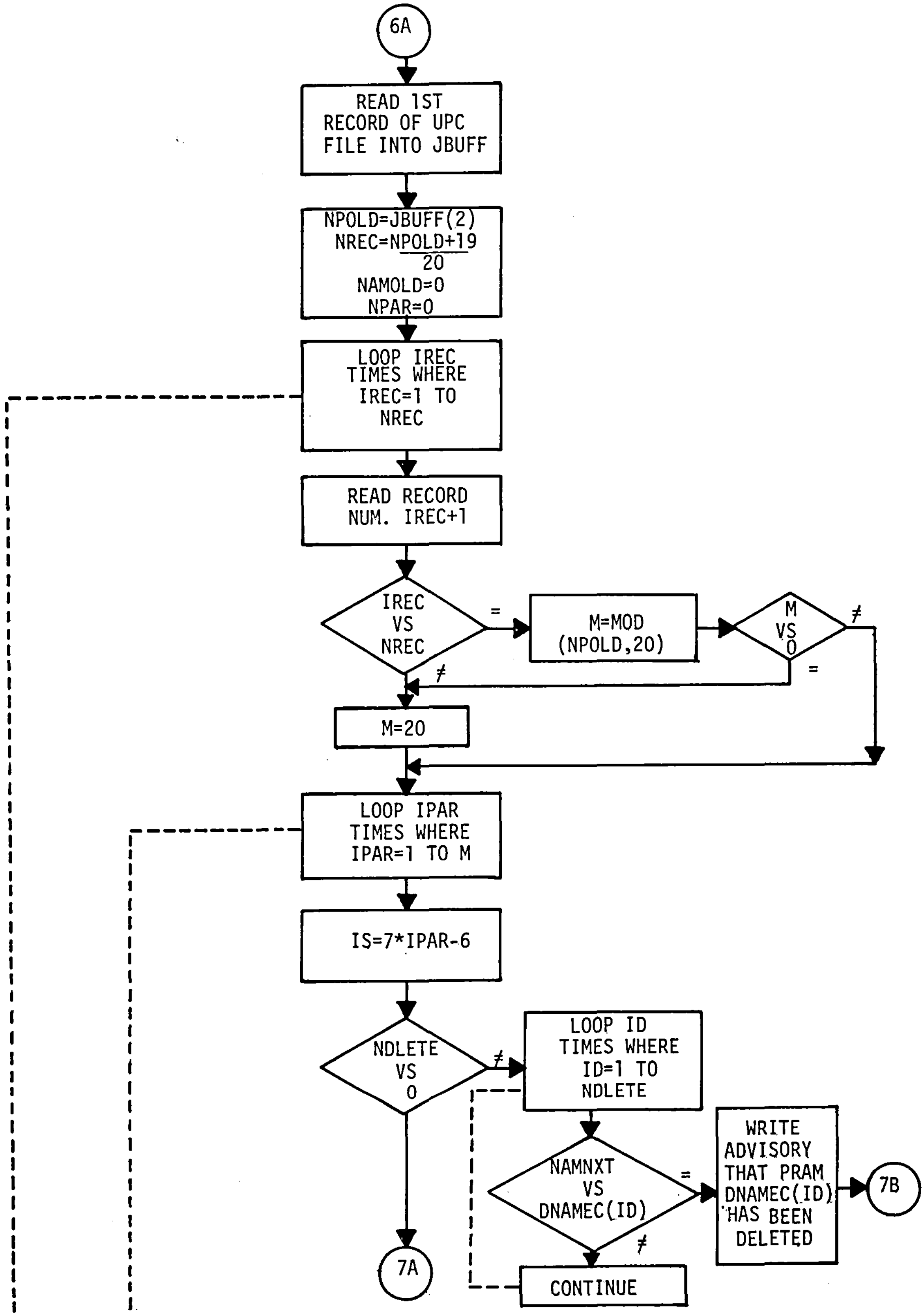

3A
READ DATA CARD TO 'BUFF'
DECODE BUFF TO 'ACTION'
ACTION VS 'ROLLEM'
=
4A
≠
ADD 1 TO NSERT
NSERT VS 200
>
WRITE ERROR MESSAGE TO PRINT FILE
ABORT
≤
MOVE DATA FROM BUFF TO REC

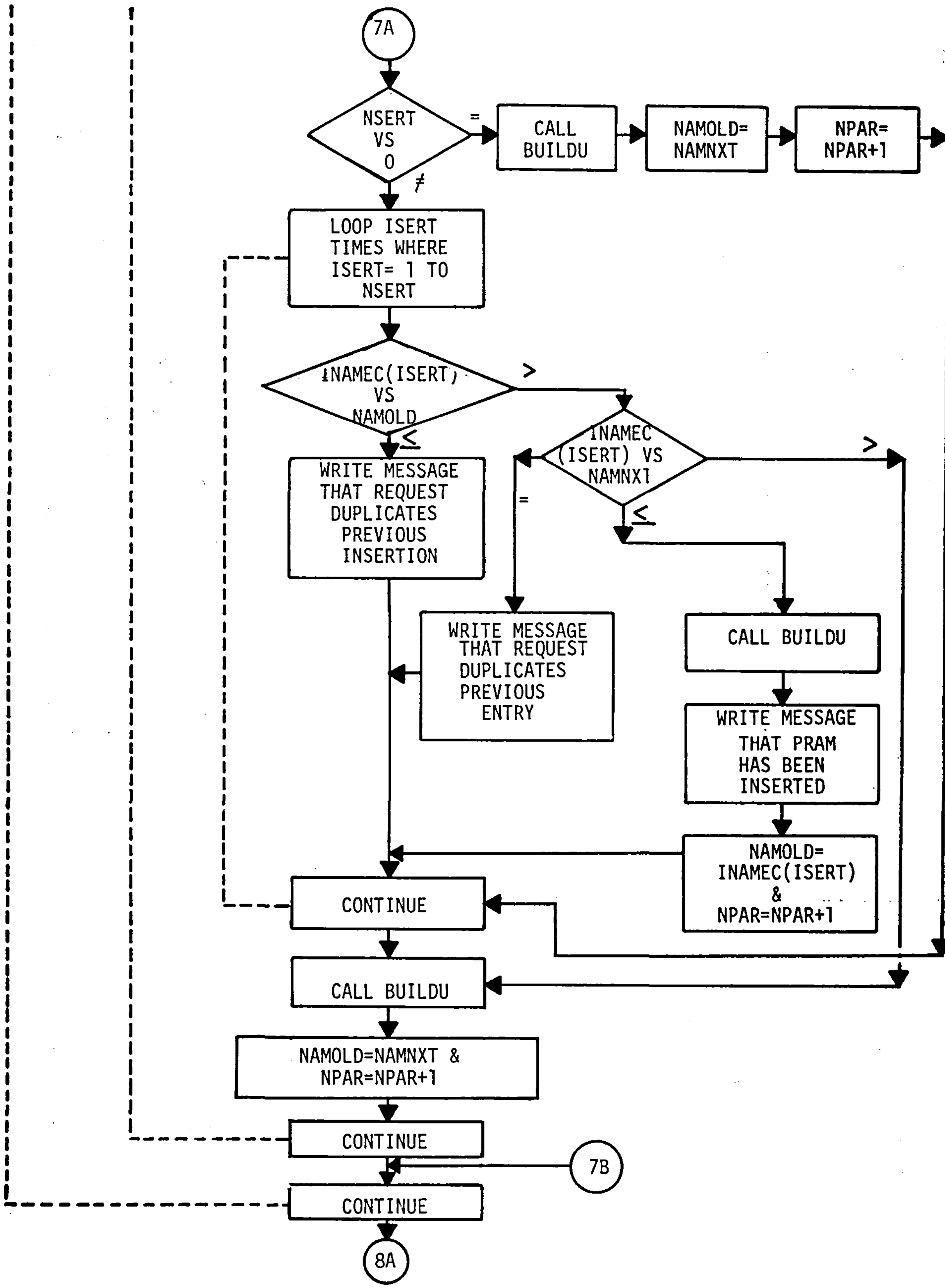

4A
NDLETE
VS
"0"
=
5A
≠
J=0
J=J+1
J
VS
NDLETE
>
5A
<
K=J
DNAMEC(K+1)
VS
DNAMEC(K)
≥
<
XNAM=
DNAMEC(K)
DNAMEC(K)
=
DNAMEC(K+1)
K=K-1
K
VS
"0"
≤
>

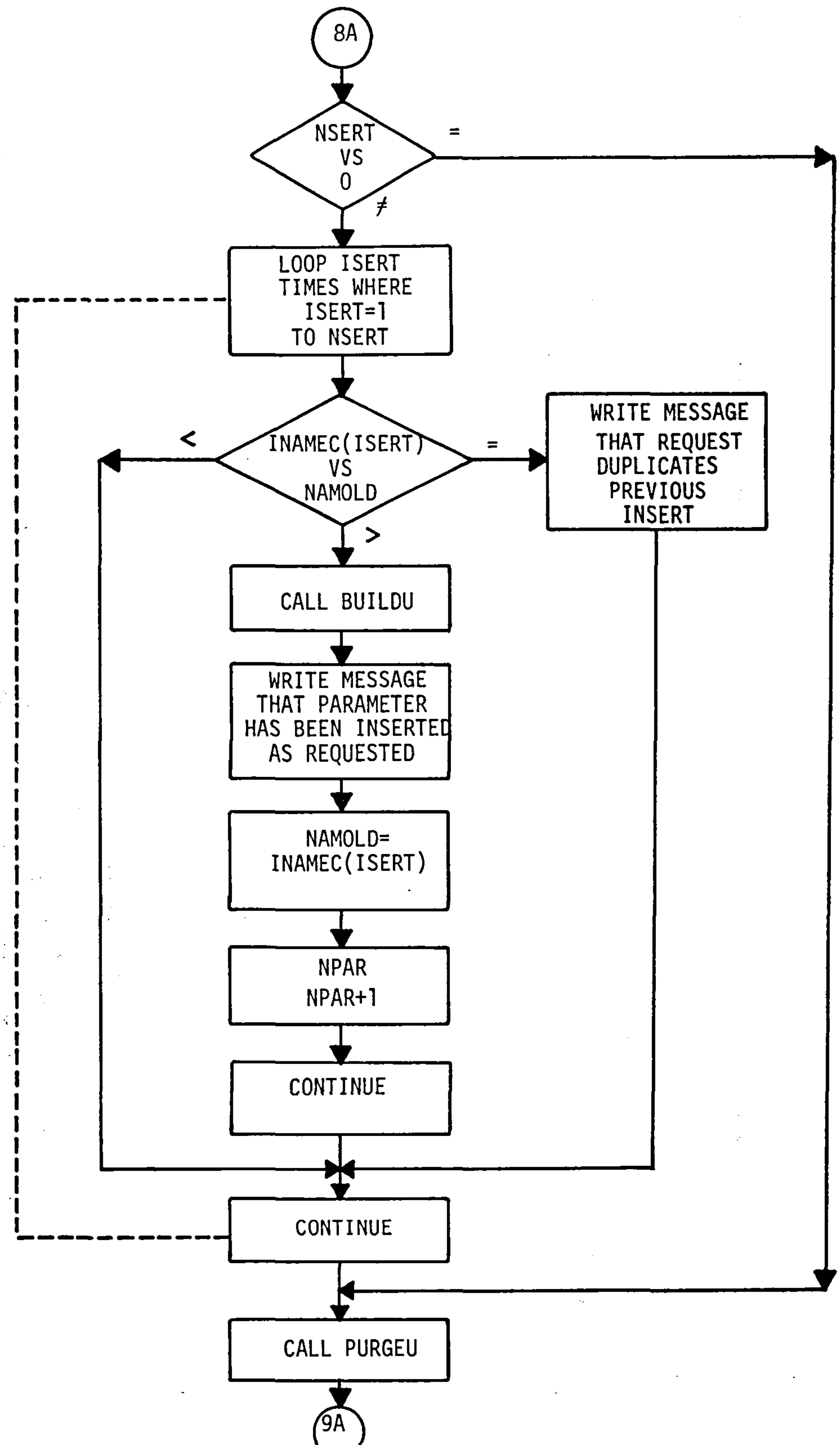

5A
NSERT
VS
"0"
=
≠
J=0
J=J+1
J
VS
NSERT
≥
<
NEWFIL
VS
"1"
=
≠
6A
5.1A
K=J
INAMEC(K+1)
VS
INAMEC(K)
≥
<
INTERCHANGE
RECORDS
K=K-1
K
VS
"0"
≤
>

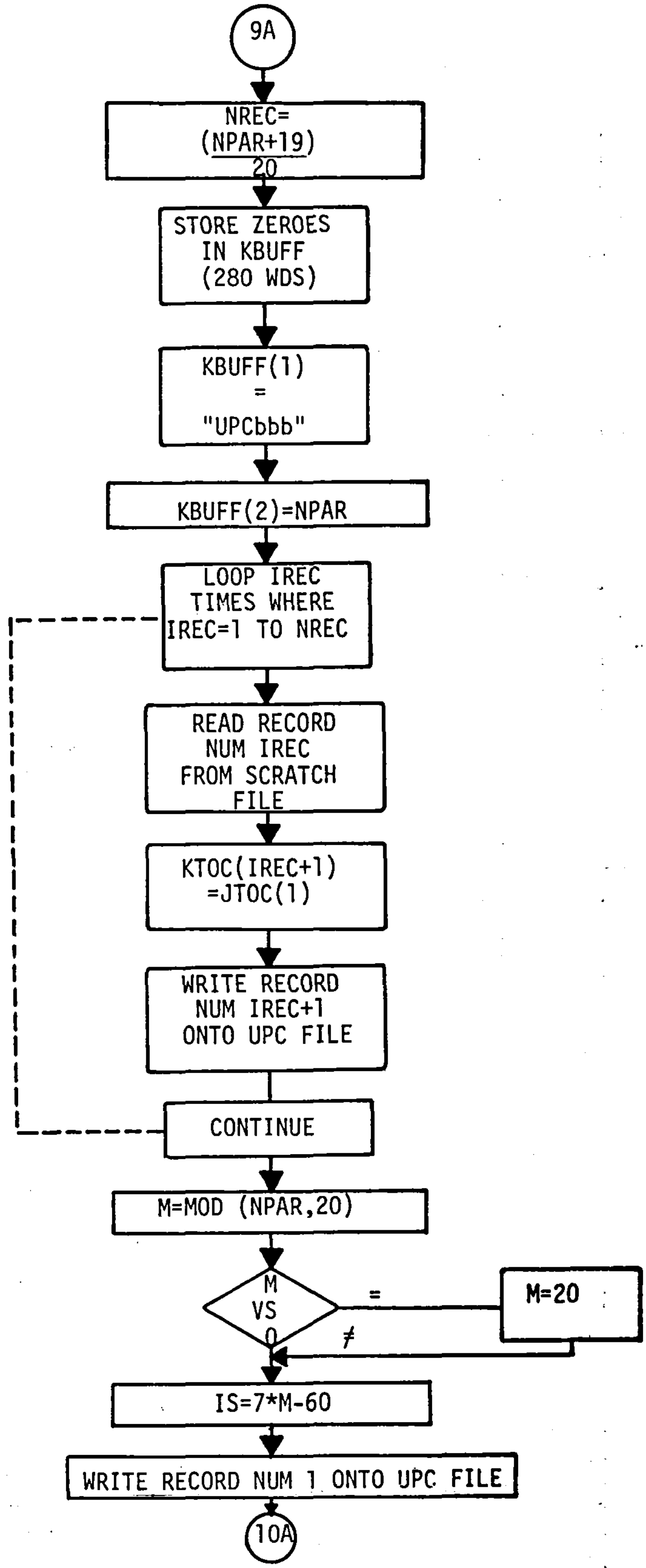

5..TA
NPAR=NSERT
NREC=
(NPAR+19)/20
LOOP I TIMES
WHERE I
=1 TO 280
KBUFF(I)=0
CONTINUE
LOOP IREC
TIMES WHERE
IREC=1 TO NREC
IREC
VS
NREC
M=MOD
(NPAR,20)
M
VS 0
=
≠
M=20
LOOP IPAR
TIMES WHERE
IPAR=1 TO
M
CALCULATE
IS,INSRT,
ITOC, & L
LOOP IW
TIMES WHERE
IW=1 TO 12
MOVE DATA FROM
REC TO JBUF


IPAR VS 1
KTOC(IREC+1)
JTOC(IS)
CONTINUE
WRITE RECORD CONSISTING OF JBUFF TO UPC FILE
CONTINUE
WRITE RECORD CONSISTING OF KBUFF TO UPC FILE
10A

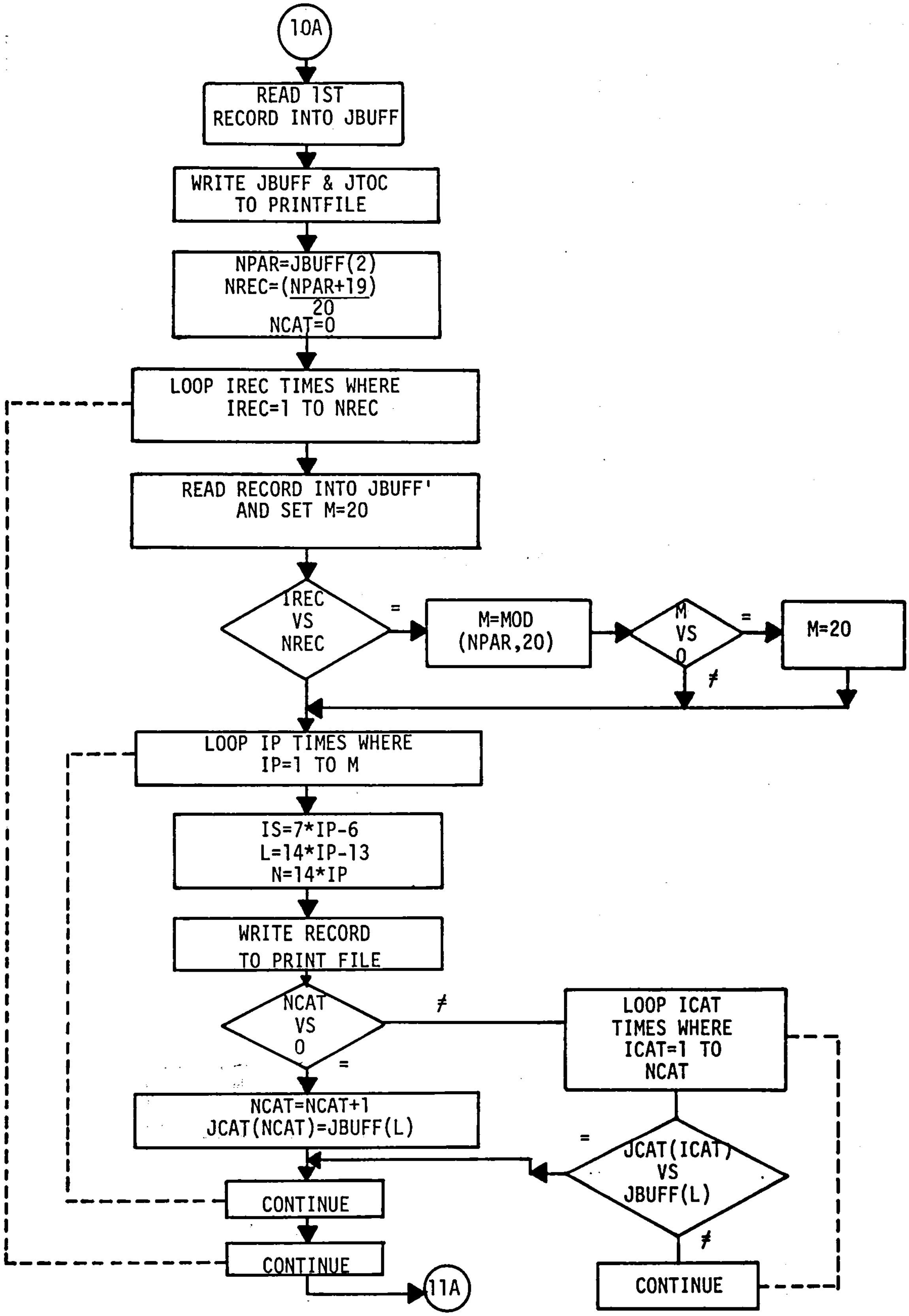

6A
READ 1ST RECORD OF UPC FILE INTO JBUFF
NPOLD=JBUFF(2)
NREC=NPOLD+19 / 20
NAMOLD=0
NPAR=0
LOOP IREC TIMES WHERE IREC=1 TO NREC
READ RECORD NUM. IREC+1
IREC VS NREC
=
≠
M=MOD (NPOLD,20)
M VS 0
≠
=
M=20
LOOP IPAR TIMES WHERE IPAR=1 TO M
IS=7*IPAR-6
NDLETE VS 0
≠
LOOP ID TIMES WHERE ID=1 TO NDLETE
NAMNXT VS DNAMEC(ID)
=
≠
WRITE ADVISORY THAT PRAM DNAMEC(ID) HAS BEEN DELETED
7B
7A
CONTINUE

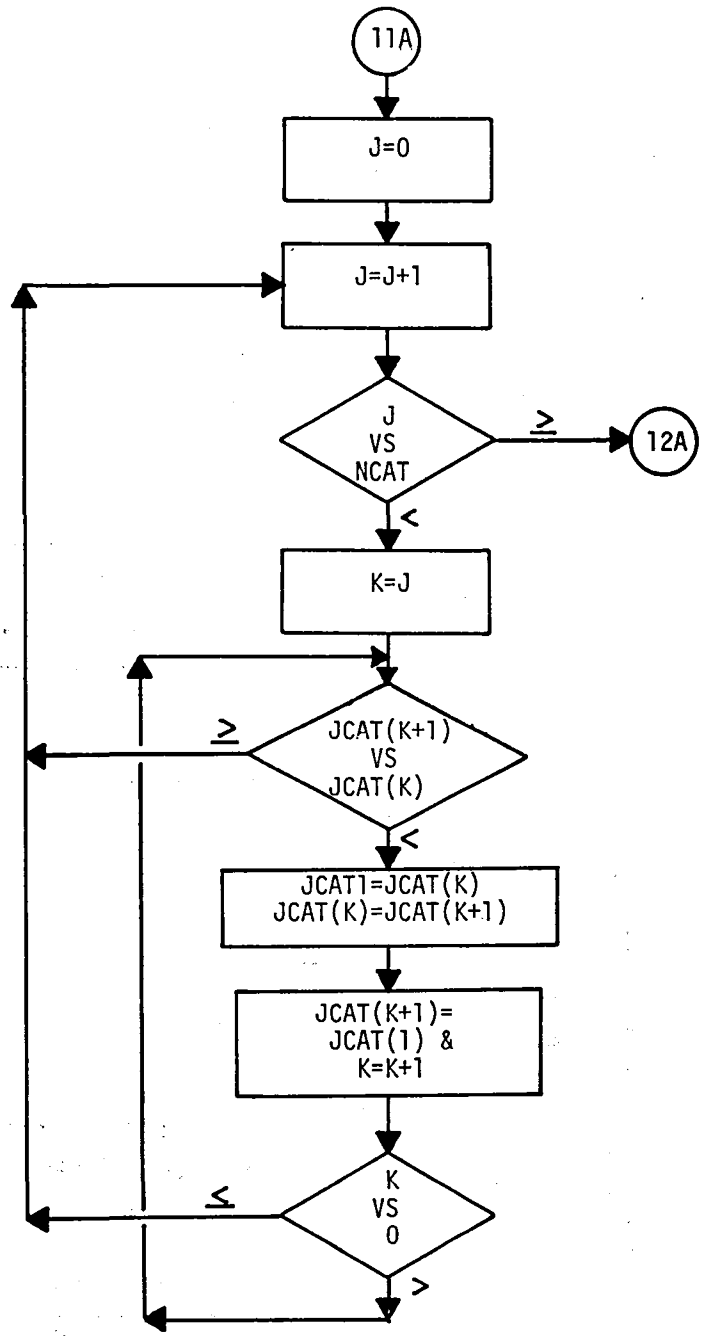

7A
NSERT VS 0
=
≠
CALL BUILDU
NAMOLD= NAMNXT
NPAR= NPAR+1
LOOP ISERT TIMES WHERE ISERT= 1 TO NSERT
INAMEC(ISERT) VS NAMOLD
>
≤
WRITE MESSAGE THAT REQUEST DUPLICATES PREVIOUS INSERTION
INAMEC (ISERT) VS NAMNX1
>
=
≤
WRITE MESSAGE THAT REQUEST DUPLICATES PREVIOUS ENTRY
CALL BUILDU
WRITE MESSAGE THAT PRAM HAS BEEN INSERTED
NAMOLD= INAMEC(ISERT) & NPAR=NPAR+1
CONTINUE
CALL BUILDU
NAMOLD=NAMNXT & NPAR=NPAR+1
CONTINUE
7B
CONTINUE
8A

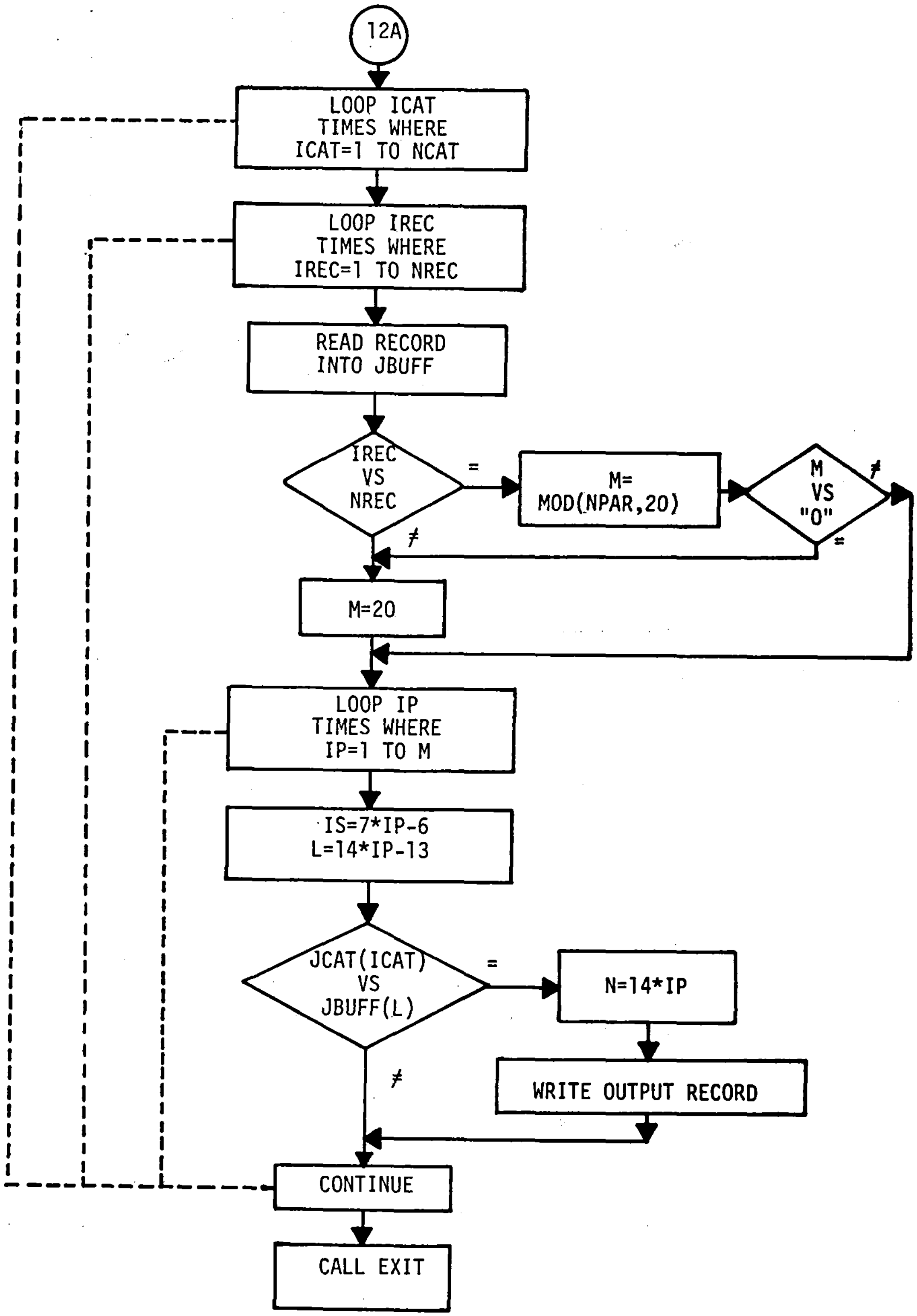

8A
NSERT VS 0
=
≠
LOOP ISERT TIMES WHERE ISERT=1 TO NSERT
INAMEC(ISERT) VS NAMOLD
<
=
>
WRITE MESSAGE THAT REQUEST DUPLICATES PREVIOUS INSERT
CALL BUILDU
WRITE MESSAGE THAT PARAMETER HAS BEEN INSERTED AS REQUESTED
NAMOLD= INAMEC(ISERT)
NPAR NPAR+1
CONTINUE
CONTINUE
CALL PURGEU
9A

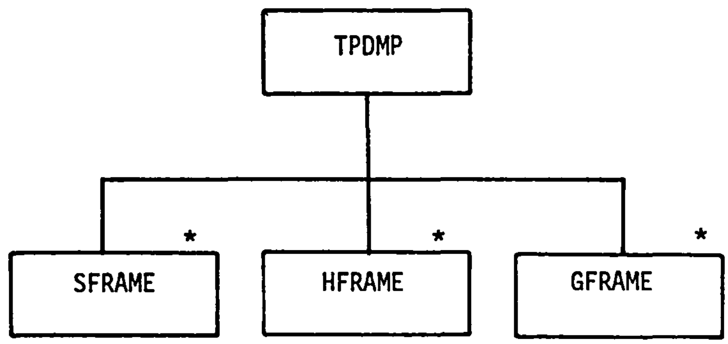


9A
NREC=
(NPAR+19)
20
STORE ZEROES
IN KBUFF
(280 WDS)
KBUFF(1)
=
"UPCbbb"
KBUFF(2)=NPAR
LOOP IREC
TIMES WHERE
IREC=1 TO NREC
READ RECORD
NUM IREC
FROM SCRATCH
FILE
KTOC(IREC+1)
=JTOC(1)
WRITE RECORD
NUM IREC+1
ONTO UPC FILE
CONTINUE
M=MOD (NPAR,20)
M
VS
0
=
≠
M=20
IS=7*M-60
WRITE RECORD NUM 1 ONTO UPC FILE
10A


10A
READ 1ST RECORD INTO JBUFF
WRITE JBUFF & JTOC TO PRINTFILE
NPAR=JBUFF(2)
NREC=(NPAR+19)/20
NCAT=0
LOOP IREC TIMES WHERE IREC=1 TO NREC
READ RECORD INTO JBUFF' AND SET M=20
IREC VS NREC
=
M=MOD (NPAR,20)
M VS 0
=
≠
M=20
LOOP IP TIMES WHERE IP=1 TO M
IS=7*IP-6
L=14*IP-13
N=14*IP
WRITE RECORD TO PRINT FILE
NCAT VS 0
≠
=
LOOP ICAT TIMES WHERE ICAT=1 TO NCAT
NCAT=NCAT+1
JCAT(NCAT)=JBUFF(L)
JCAT(ICAT) VS JBUFF(L)
=
≠
CONTINUE
CONTINUE
CONTINUE
11A


11A
J=0
J=J+1
J
VS
NCAT
≥
12A
<
K=J
JCAT(K+1)
VS
JCAT(K)
≥
<
JCAT1=JCAT(K)
JCAT(K)=JCAT(K+1)
JCAT(K+1)=
JCAT(1) &
K=K+1
K
VS
0
≤
>


12A
LOOP ICAT TIMES WHERE ICAT=1 TO NCAT
LOOP IREC TIMES WHERE IREC=1 TO NREC
READ RECORD INTO JBUFF
IREC VS NREC
=
≠
M= MOD(NPAR,20)
M VS "0"
≠
=
M=20
LOOP IP TIMES WHERE IP=1 TO M
IS=7*IP-6
L=14*IP-13
JCAT(ICAT) VS JBUFF(L)
=
≠
N=14*IP
WRITE OUTPUT RECORD
CONTINUE
CALL EXIT

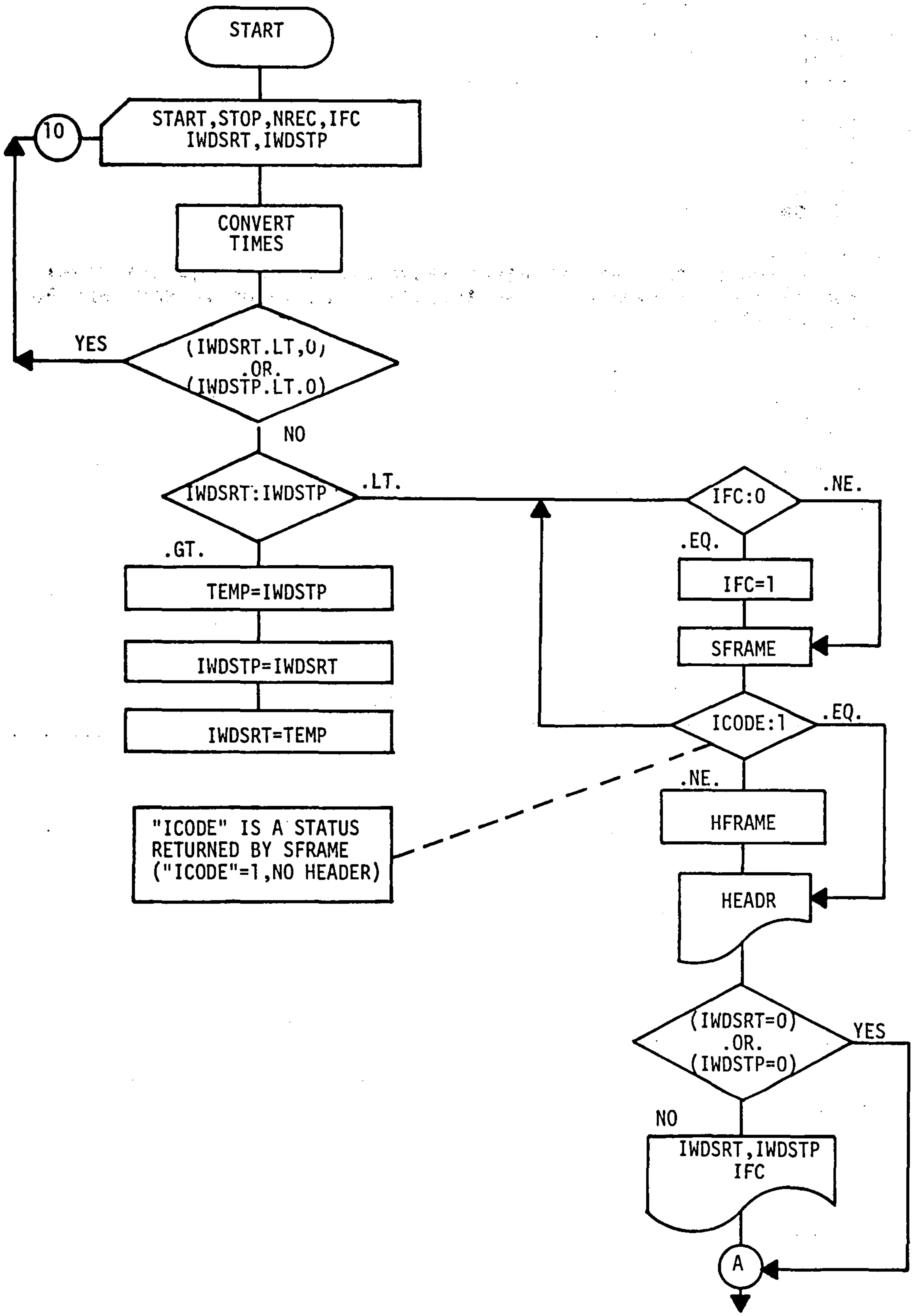


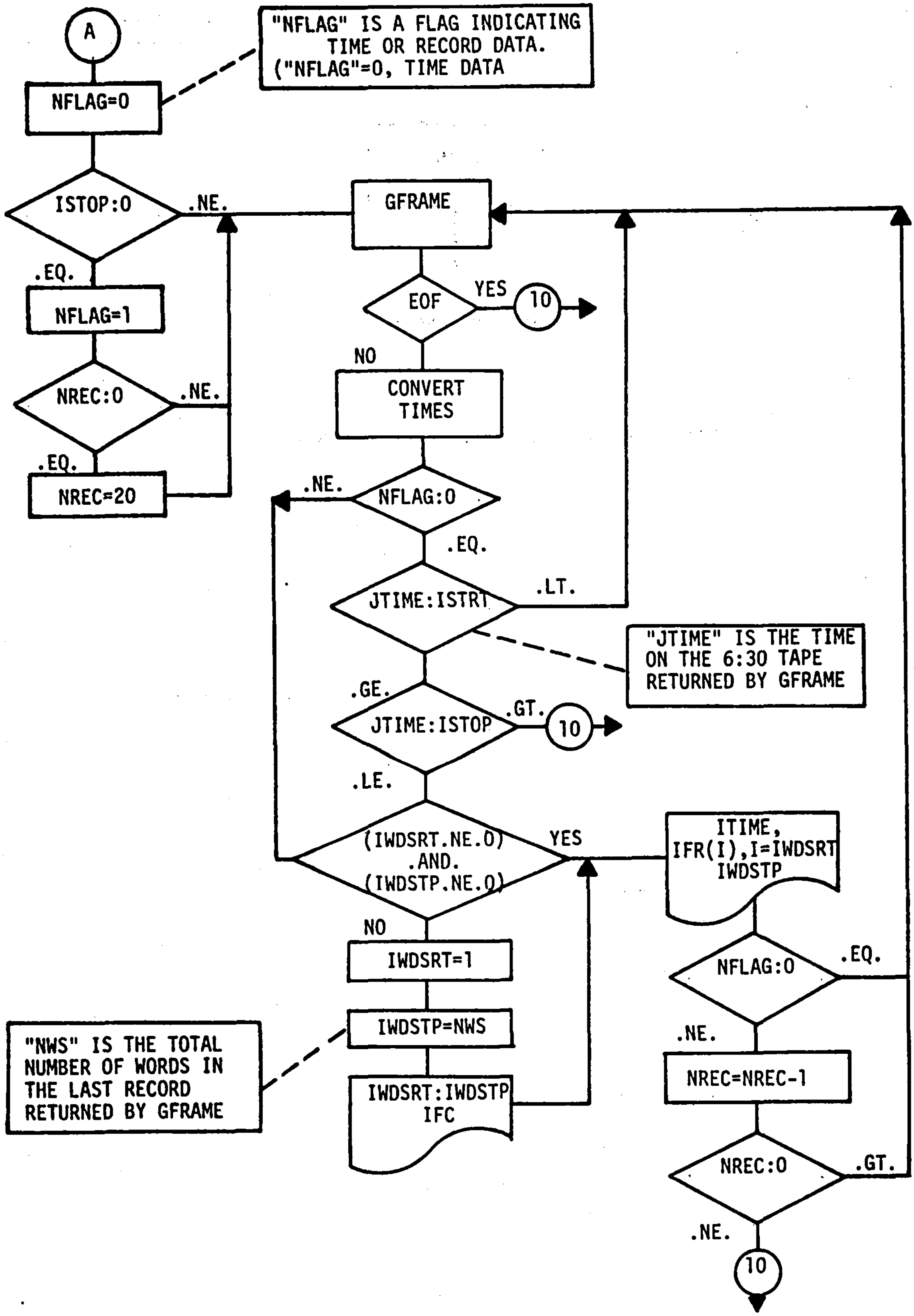
*Program Modules SFRAME, HFRAME, GFRAME documentation in NASA, WFC Program Library

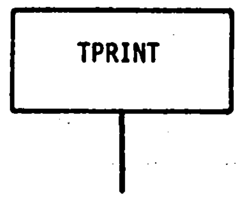
HIERARCHY CHART FOR TPDMP PROGRAM

```
TPDMP        9TRK TAPE DUMP PROGRAM
******* NASA WALLOPS VERSION OF 08/15/77

******* LANGUAGES-
        FORTRANY(MAIN ROUTINE)
        GMAP(SUBROUTINES)

******* MACHINE-
        HW625 OR HW635

******* PURPOSE-
        TO DUMP PORTIONS,SPECIFIED BY THE USER,OF AN EMR 6130 TELEVENT
        II DATA TAPE IN A USABLE FORMAT.

******* METHOD-
        FROM THE CARD READER THE FOLLOWING INFORMATION IS READ<START
        TIME,STOP TIME,NUMBER OF RECORDS,TAPE FILE,STARTING WORD AND
        STOPPING WORD. THE PORTIONS OF THE 6130 TAPE DEFINED BY THIS
        CARD ARE THEN OUTPUT SEQUENTIALLY TO THE LINE PRINTER. ANY
        NUMBER OF THESE DATA CARDS MAY BE PRESENT SINCE AN EOF ON
        FILE 05 TERMINATES PROCESSING.

******* INPUT -ON FILE 05(CARD READER)
        IDAY1 ,STARTING DAY
        IHR1  ,STARTING HOUR
        IMIN1 ,STARTING MINUTE
        ISEC1 ,STARTING SECOND
        JSEC1 ,STARTING TENTH OF A MILLISECOND
        IDAY2 ,STOPPING DAY
        IHR2  ,STOPPING HOUR
        IMIN2 ,STOPPING MINUTE
        ISEC2 ,STOPPING SECOND
        JSEC2 ,STOPPING TENTH OF A MILLISECOND
        IWDSRT,STARTING WORD
        IWDSTP,STOPPING WORD
              -ON FILE 01(6130 TAPE)
        ITIME ,FIVE WORD FRAME TIME
        IFR   ,TWO HUNDRED WORD FRAME DATA
        NWS   ,MAXIMUM NUMBER OF WORDS PER FRAME
******* OUTPUT-ON FILE 06(LINE PRINTER)
        HEADR ,EIGHTY CHARACTER TAPE HEADER
        IFC   ,TAPE FILE TO BE PROCESSED
        IWDSRT,STARTING WORD
        IWDSTP,STOPPING WORD
        ITIME ,FIVE WORD FRAME TIME
        IFR   ,TWO HUNDRED WORD FRAME DATA

******* RESTRICTIONS-
        TO OBTAIN ANY DATA AT LEAST ONE CARD MUST BE PRESENT.THIS CARD
        MAY BE BLANK IF STANDARD OPTIONS ARE WANTED.
```

```
******* SUBROUTINES REQUIRED-
        SFRAME
        HFRAME
        GFRAME
        STATS
        EBCBCD
        TRANS
        G8F36
        G16F36

******* REMARK-
        THE STANDARD OPTIONS DESCRIBED ABOVE ARE AS FOLLOWS< THE FIRST
        TWENTY RECORDS FROM FILE ONE INCLUDING ALL WORDS IN EACH RECORD
        ARE PRINTED.
```


START
START,STOP,NREC,IFC
IWDSRT,IWDSTP
10
CONVERT TIMES
YES
(IWDSRT.LT.0)
.OR.
(IWDSTP.LT.0)
NO
IWDSRT:IWDSTP
.LT.
.GT.
TEMP=IWDSTP
IWDSTP=IWDSRT
IWDSRT=TEMP
IFC:0
.NE.
.EQ.
IFC=1
SFRAME
ICODE:1
.EQ.
.NE.
HFRAME
HEADR
"ICODE" IS A STATUS
RETURNED BY SFRAME
("ICODE"=1,NO HEADER)
(IWDSRT=0)
.OR.
(IWDSTP=0)
YES
NO
IWDSRT,IWDSTP
IFC
A

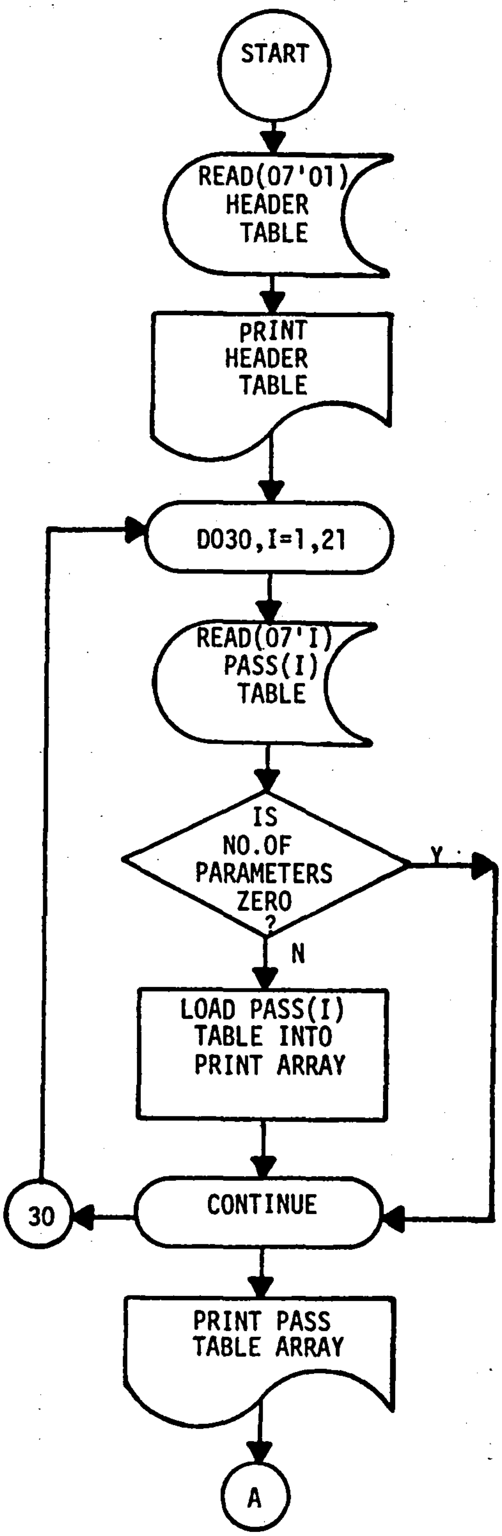

A
NFLAG=0
"NFLAG" IS A FLAG INDICATING TIME OR RECORD DATA. ("NFLAG"=0, TIME DATA
ISTOP:0
.NE.
.EQ.
NFLAG=1
NREC:0
.NE.
.EQ.
NREC=20
GFRAME
EOF
YES
10
NO
CONVERT TIMES
NFLAG:0
.NE.
.EQ.
JTIME:ISTRT
.LT.
"JTIME" IS THE TIME ON THE 6:30 TAPE RETURNED BY GFRAME
.GE.
JTIME:ISTOP
.GT.
10
.LE.
(IWDSRT.NE.0) .AND. (IWDSTP.NE.0)
YES
ITIME, IFR(I),I=IWDSRT IWDSTP
NO
IWDSRT=1
NFLAG:0
.EQ.
IWDSTP=NWS
"NWS" IS THE TOTAL NUMBER OF WORDS IN THE LAST RECORD RETURNED BY GFRAME
.NE.
NREC=NREC-1
IWDSRT:IWDSTP IFC
NREC:0
.GT.
.NE.
10

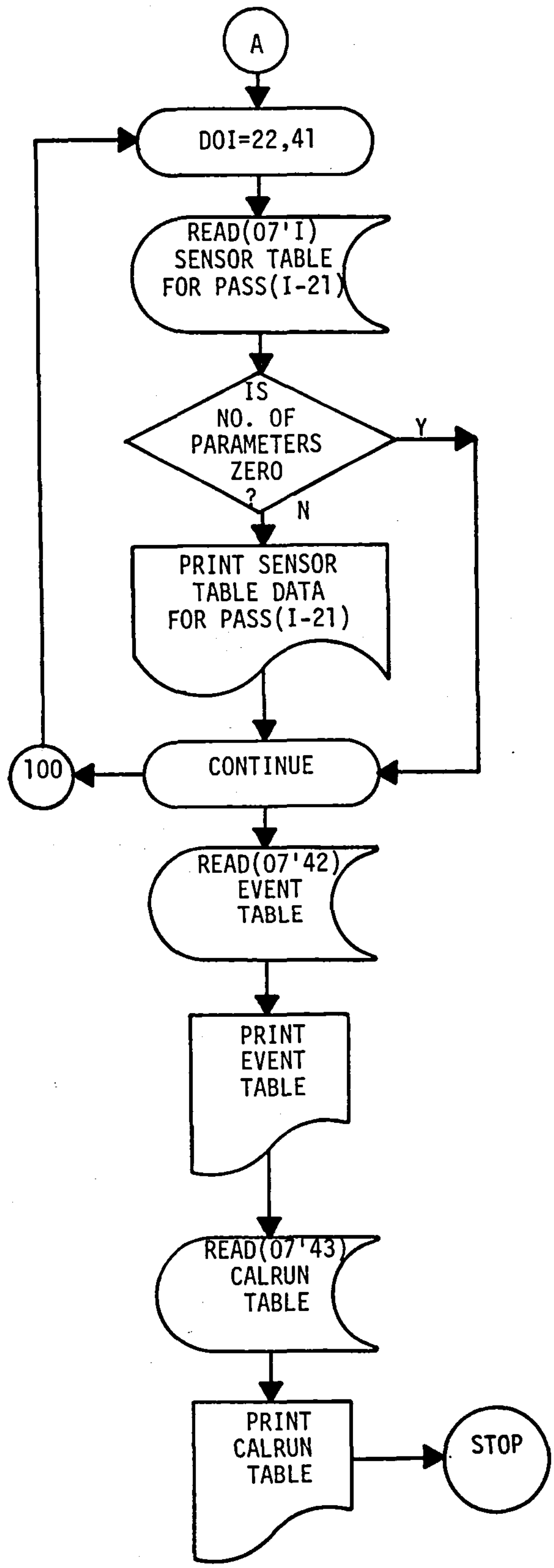


(No Subprograms)

HIERARCHY CHART For TABLE PRINT PROGRAM

```
TPRINT          TABLE PRINTOUT

******* NASA WALLOPS VERSION OF 08/15/77

******* LANGUAGE-
        FORTRANY

******* MACHINE-
        HW625 OR HW635

******* PURPOSE-
        TO MAKE AVAILIBLE IN A USABLE FORMAT THE INFORMATION CURRENTLY
        PLACED ON AN RSRA TABLE FILE

******* METHOD-
        MAKING USE OF THE SUBROUTINE RANSIZ THE DISC FILE CONTAINING
        THE TABLE WE ARE INTERESTED IN IS READ RECORD BY RECORD. AS
        EACH RECORD IS READ THE WORDS CONTAINING INFORMATION ARE
        PRINTED WITH APPROPRIATE LABELS. NO CARD INPUT IS USED.

******* INPUT -ON FILE 07(THE DISC FILE CONTAINING TABLE)
        THEAD -TWENTY SEVEN WORD HEADER RECORD
        TPASS -TWENTY RECORD,ONE HUNDRED TWENTY WORD PASS TABLE
        TSNSNE-TWENTY RECORD,FIVE HUNDRED TWENTY WORD SENSOR TABLE
        TEVNT -SIX HUNDRED WORD EVENT RECORD
        TCALR -ONE HUNDRED TWENTY WORD CALIBRATION RECORD

******* OUTPUT-ON FILE 06(THE LINE PRINTER)
        THEAD -TWENTY SEVEN WORD HEADER RECORD
        TPASS -TWENTY RECORD,ONE HUNDRED TWENTY WORD PASS TABLE
        TSNSNE-TWENTY RECORD,FIVE HUNDRED TWENTY WORD SENSOR TABLE
        TEVNT -SIX HUNDRED WORD EVENT RECORD
        TCALR -ONE HUNDRED TWENTY WORD CALIBRATION RECORD

******* RESTRICTIONS-
        NONE

******* SUBROUTINES-
        NONE
```


START
READ(07'01)
HEADER
TABLE
PRINT
HEADER
TABLE
DO30,I=1,21
READ(07'I)
PASS(I)
TABLE
IS
NO.OF
PARAMETERS
ZERO
?
Y
N
LOAD PASS(I)
TABLE INTO
PRINT ARRAY
30
CONTINUE
PRINT PASS
TABLE ARRAY
A



☆ U.S. GOVERNMENT PRINTING OFFICE: 1979-635-004/23

1. Report No. NASA RP-1043	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle PROCESSING OF ON-BOARD RECORDED DATA FOR QUICK ANALYSIS OF AIRCRAFT PERFORMANCE		5. Report Date September 1979
		6. Performing Organization Code
7. Author(s) Norman H. Michaud		8. Performing Organization Report No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Wallops Flight Center Wallops Island, VA 23337		10. Work Unit No.
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Reference Publication
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A system of independent computer programs for the processing of digitized pulse code modulated (PCM) and frequency modulated (FM) data. Information is stored in a set of random files and accessed to produce both statistical and graphical output. The software system is designed primarily to present these reports within a twenty-four hour period for quick analysis of the vehicle's performance.

17. Key Words (Suggested by Author(s)) Data processing Computer programs Helicopter	18. Distribution Statement Unclassified - unlimited STAR Category 05

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	322	$11.75

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

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business

Penalty for Private Use, $300

SPECIAL FOURTH CLASS MAIL
BOOK

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return