INSTITUTE
OF
HYDROLOGY

SOFTWARE PACKAGE FOR SIMULTANEOUS ACCESS TO PLOTTING FACILITIES ON UNIVAC 1108

by

A P J LOBLEY and R F TEMPLEMAN

ABSTRACT

A software package has been developed which allows many users to access the off-line plotting facilities simultaneously on the Institute's UNIVAC 1108 computer. The package provides for the efficient use of the UNIVAC's 7-track magnetic tape facilities, simplifies the use of the CALCOMP 936 graph plotter and attempts to maximise the throughput of graphs on the plotter. It also provides an automatic means for charging for each plotter job using an interface with the standard IH accounting routines.

REPORT NO 54

March 1978

CONTENTS

1. INTRODUCTION

A software package, known as 'The Spooling System' has been developed on the Institute of Hydrology's UNIVAC 1108 computer to allow many users simultaneous access to the Institute's off-line CALCOMP 936 plotter. The plotter's control unit reads and interprets plotting commands written on a 7-track magnetic tape by FORTRAN programs executed on the UNIVAC. The package was written to improve the usage of the UNIVAC's 7-track magnetic tape facilities which had become a major bottleneck to throughput on the system. The Spooling System also simplifies the use of the graph plotter for the user and improves the efficiency with which the plotter can be used. It also calculates a charge automatically for each graph produced and provides an easy method of accounting graph plotter usage using the standard IH accounting runs. In addition the spooling system causes an identification box to be appended to each plot which provides both the user and the Calcomp operator with relevant information.

Using the Spooling System reduces the number of graph plotter tapes required in a pool to provide an adequate service by allowing the UNIVAC operator to write many user plots to one magnetic tape. Plots are categorised into four basic types, depending on paper width and pen type. Each type is written to a separate tape. The use of multiple plot tapes reduces the workload on the 7-track magnetic tape subsystem. This is not required by each user to produce one plot but only when the UNIVAC operator's collection program is being run to gather many plots to tape. The number of magnetic tapes which the plotter operator has to handle is reduced and, because a list of the plots written to each tape is produced, the operator is given a simplified task in plotting each user plot. The list of plots on each tape includes an estimate of the length of time each plot will take thus enabling the operator to schedule the plots more efficiently and improve the average turn round time for each user plot.

As the user is permitted to write his plotter information to a disc file, not directly to tape, he may create a plotfile without any operator intervention. Hence the plotting facilities are made available to users during unmanned periods of operation. This also eliminates the delay to the user of waiting for a magnetic tape unit before being able to complete his plot. A processor is used to assign the users plotfile and also makes an entry in the system plotfile directory which contains the user's pen and paper requirements etc. Thus the user specifies his requirements within his runstream and these requirements are printed on the output from his job. The user may easily check the requirements requested in a given job from his lineprinter output. The commands which the user must insert in his runstream are simple and are used to generate a directory entry for the plot, inserting the user's pen and paper requirements, the number of copies of the plot, the FORTRAN channel number to which the plotter information will be written and any message the user may wish to convey to the plotter operator. The runstream commands also assign a disc file which will be used for the plotter information (the plotfile) and to associate this file with the user-supplied FORTRAN channel number.

The graphical output from the spooling system will contain all the information written by the user to his plotfile and also an identification plot block (Figures 1, 2 and 3) which enables the user to associate the graph with the run which created it from the time and date at which the directory entry was made, and the account and runid of the job which created the plot. The plot block also contains the cost of the plot (in pounds) and the estimated time to plot it (in minutes). The remaining entries in the plot block inform the operator of the user's pen requirements and the number of copies of the plot to expect.

Finally the spooling system calculates the cost of a plot from the number of Calcomp records contained within the user's plotfile. This ensures a consistency of charging which could never be achieved from a manual accounting system and considerably reduces the plotter operator's book-keeping duties.

The Spooling System consists of a system processor (PLDIR) which is called by the user; an operator's program (PLOTTAPE); three utility programs DIRCRE, DIRBOJ and HOWMANY and two accounting routines, ZERO and PLCHG.

PLDIR creates an entry in the plotfile directory containing the user's account code and runid, the time and date of creation of the entry, the number of copies of the plot which are required, the user's pen and paper specification and his comments to the operator. It then assigns a plotfile for the user and associates it to the user-supplied FORTRAN channel number.

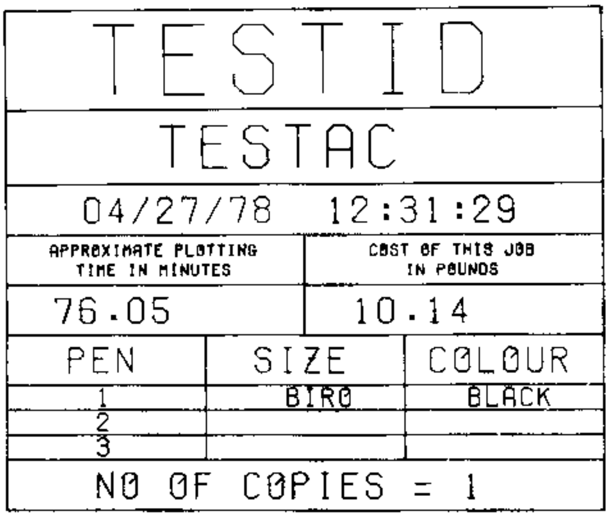
PLOTTAPE takes the contents of the system plotfile directory at any instant and transfers the files it has referenced to magnetic tapes. Each tape will contain all the plots of a given type written out in the best order as described later. PLOTTAPE generates appropriate header information on the tape and precedes each user plot by a plot block (see Figures 1, 2 and 3) which contains the information from the directory entry.

The utility programs DIRCRE, DIRBOJ and HOWMANY are used to create and initialise a plotfile directory, to rebuild a plotfile directory which contains an error and to notify the operators of the number of plots in the directory, respectively.

The accounting program ZERO is used to clear the accounting section entry in the plotfile directory and may be used when the system is installed and after each successful run of the accounting programs.

The accounting routine PLCHG is provided so that an ASCII FORTRAN accounting program may obtain the values of cost and time held in the accounting section of the plotfile directory. PLCHG is called from an accounting program (which passes an account code to PLCHG) to find the total cost and accumulated plot time used by the account since the last

FIGURE 1



TESTID

TESTAC

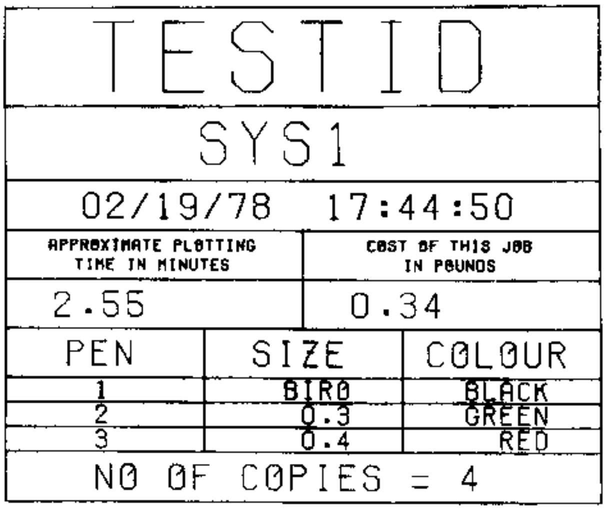
04/27/78 12:31:29

APPRØXIMATE PLØTTING TIME IN MINUTES	CØST ØF THIS JØB IN PØUNDS
76.05	10.14

PEN	SIZE	CØLOUR
1	BIRØ	BLACK
2		
3		

NØ ØF CØPIES = 1

FIGURE 2



TESTID

SYS1

02/19/78 17:44:50

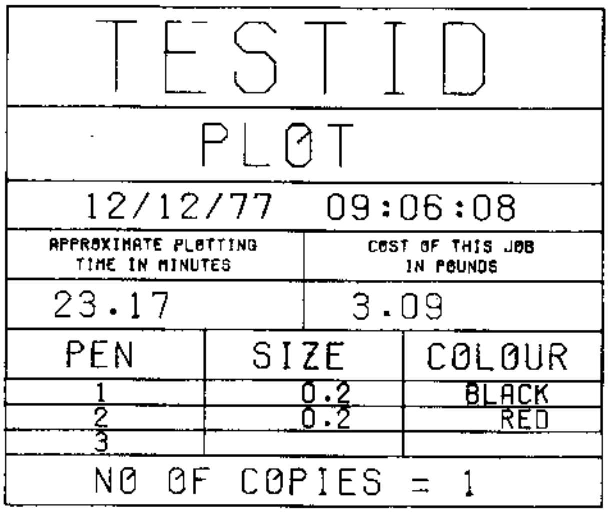
APPRØXIMATE PLØTTING TIME IN MINUTES	CØST ØF THIS JØB IN PØUNDS
2.55	0.34

PEN	SIZE	CØLOUR
1	BIRØ	BLACK
2	0.3	GREEN
3	0.4	RED

NØ ØF CØPIES = 4

USER CØMMENTS :-
PLEASE START AT LEFT MARGIN

FIGURE 3



TESTID

PLØT

12/12/77 09:06:08

APPRØXIMATE PLØTTING TIME IN MINUTES	CØST ØF THIS JØB IN PØUNDS
23.17	3.09

PEN	SIZE	CØLOUR
1	0.2	BLACK
2	0.2	RED
3		

NØ ØF CØPIES = 1

initialisation of the account file. This information is passed back to the accounting program.

2. THE USER GUIDE

2.1 Introduction

The Spooling System allows a UNIVAC user to plot graphs without using a tape unit directly. The user's plot is written to a mass storage file and an entry is made in the system plot directory. The directory entry contains all the information necessary to select pen types, colours and paper widths. The user, in his runstream, has only to place a call to the processor which creates a directory entry for each plot he wants to generate. The call to the directory entry processor is placed in the user's runstream before the execution of the program which generates the plot. When the plot directory contains a number of plots the computer operators run a batch job which examines the plot directory and copies each plot onto one of four magnetic tapes depending on the width of the paper the user requested and the type of pen (biro or ink) required for the plot. Each plot on the magnetic tape is preceded by a plot block which informs the computer operator of the pen requirements and the user of the cost of his plot and the time taken to plot the graph (see Figures 1, 2 and 3). The plot block also contains the 'runid' and account number of the job which created it, the time and date of production of the directory entry and the number of copies of the plot to be plotted. When the plot information block has been plotted the pen carriage will be positioned against the right margin stop and fifteen centimetres from the plot block in the positive X direction.

The Spooling System automatically accounts for each plot produced and charges the cost of the plot to the account number used when producing the plot. The charge is worked out according to the number of Calcomp plot records in the user's plotfile. The number of records in the plotfile is counted and a conversion factor is applied to convert this to an estimated plot time. This is produced in the header block on the user's plot and a charge factor is applied to this time to give a cost in pounds which is also produced in the header block on the user's plot (see Figures 1, 2 and 3).

2.2 Use of the Plot Directory Entry Processor (PLDIR)

2.2.1 *General Description*

The plot directory entry processor (PLDIR) constructs a directory entry for the user. The directory entry contains the user's pen requirements (pen sizes and colours), the 'runid' and account number of the job which created it, the name of the plot file which will contain the plotter instruction codes, the number of copies of the information in the plot file which the user requires to be plotted, the time and date at which the directory entry was constructed and any comment which the user wishes to plot. PLDIR will also assign a mass storage file and associate this to the FORTRAN channel number to which the user's program will write the plotter instructions. PLDIR also informs both the operator and the user, of the sequence number of the plot in the directory.

2.2.2 *The processor call*

PLDIR is called by means of the job control statement:

```
@PLDIR Ch
```

where Ch is the FORTRAN channel number to which the user program will write the plotter instructions. Ch must be two characters or less and if it is omitted then channel 8 is assumed.

2.2.3 *The pen description data card*

The pen description data card must be the first of the two possible data cards which the user may supply after the PLDIR call. If the pen description data card is not supplied the second data card must not be supplied either. The pen description data card is used to give PLDIR the user user's paper requirements and the number of copies required as well as the pen sizes and colours. The data card has the format:

```
PWID,NCP,P1S,P1C,P2S,P2C,P3S,P3C
```

where:

PWID is the paper width required and must be either 36 or 15 representing either 36 in paper or 15 in paper. Any value other than 36 or 15 in this field will cause PLDIR to error.

NCP is the number of copies of the information in the plot file that the user requires. This field must be a single digit number, any other value producing only one copy. The field may be left blank if one copy only is required.

P1S, P2S, P3S are the pen sizes required in pen 1, pen 2, pen 3 positions respectively. These fields should be either a number representing the pen width in millimetres (only one decimal place should be coded) or the word BIRO if a biro is required.

P1C, P2C, P3C are the pen colours required in pen 1, pen 2, pen 3 positions respectively. The colour fields must be selected from the available colours (See Table 2) and may be up to six characters in length.

If any of the last six fields on the pen description card is left blank then no pen will be mounted in that position. Fields may be omitted, in pairs, from the right if the pen is not required. Fields are as pen 1, a 0.3 mm green pen as pen 2 and a 0.4 mm red pen as pen 3 and with the operator instruction "PLEASE START AT LEFT MARGIN" plotted before the plot block. The user's program will write the plotter instructions to FORTRAN channel 23.

Example showing use of some defaults

```
@RUN  Prog,acc,Proj
-----
@PLDIR
15,,0.2,BLACK,0.2,RED
@EOF
-----
@XQT User Prog
-----
data
-----
@FIN
```

The plot block is shown in Figure 3. This plot would require 15 in paper and would plot one copy using 0.2 mm pens in positions 1 and 2 on the pen carriage. Pen 1 would use black and Pen 2 red ink. The plotter instructions would be written to channel 8.

2.3 Restrictions to the use of the plotter when using the Spooling System

2.3.1 The user is restricted to having one CALL PLOT(X,Y,999) in his program. The spooling system uses this call to signal the end of the plot file when copying plots from mass storage to tape and any information after the CALL PLOT(X,Y,999) will be ignored. However any number of plot files may be generated from within any one run by repeated use of the PLDIR processor.

2.3.2 The user is resticted to a maximum mass storage file size of 500 tracks for the file which will contain his plotter information. This restriction may be over-ridden, if necesary, by inserting the following job control language to supplement the call to PLDIR:

```
-----
@PLDIR ch
data card one as usual
data card two as usual
@FREE,R ch
@ASG,A ch,///NMAX
@FREE,R ch
@ASG,AX ch
@XQT ---- etc.
```

where NMAX is the maximum file size required by the user plot.

2.3.3 Not more than 9 copies of any plot may be produced from any one plot file. If more copies are required a new directory entry and plot file must be created. This may be done by the method shown separated by commas and may contain blanks which will be ignored when found.

2.2.4 *The comment data card*

The second data card contains a user's comment to the plotter operator which may be up to 48 characters in length. The specified comment will be plotted on the opposite side of the plot block from the user's plot. If the pen description data card is omitted the comment data card must also be omitted.

2.2.5 *The default actions*

If the user provides no data cards at all after a call to PLDIR the Spooling System will assume a default of one copy on 15 in paper and one black biro in pen 1 position. It is permissible to specify the FORTRAN channel number on the PLDIR call line and still use the default pen and paper specifications.

2.2.6 *Examples*

Use of the default

```
@RUN  Prog,acc,Proj
-----
@PLDIR
@XQT  user prog
-----
data
-----
@FIN
```

The above runstream will produce one copy of the plot on 15 in paper using one black biro in pen 1 position. The plotting instructions within the user'sprogram must be written to FORTRAN channel 8 (see Figure 1).

Specification of all possible parameters

```
@RUN  Prog,acc,Proj
-----
@PLDIR 23
36,4,BIRO,BLACK,0.3,GREEN,0.4,RED
PLEASE START AT LEFT MARGIN
@EOF
-----
@XQT User Prog.
-----
data
-----
@FIN
```

The plot block produced by this call is shown in Figure 2. The user would obtain four copies of the plot on 36 in paper with a black biro in the following example:

It is required to plot 15 copies of a plot produced by the user program to be found in element FILE.TEST. The runstream used to produce this would be:

```
(1)   @ASG,T A.,///500
(2)   @USE ch,A.
(3)   @XQT FILE.TEST
      -----
      data
      -----
(4)   @PLDIR ch
      PWID,9, Pen specifications
      User comment
(5)   @COPY A.,ch
(6)   PLDIR ch
      PWID,6, Pen specifications
      User comment
(7)   @COPY. A.,ch.
      ---- etc
```

Statements (1) and (2) define a temporary mass storage file which will contain the plotter instructions.

Statement (3) executes the program which writes the plotter information to file A.

Statement (4) creates a Spooling System file and directory entry from which the graphs will be plotted. This Spooling System file will be used to produce 9 of the 15 copies.

Statement (5) copies the plotter codes from the temporary file A into the Spooling System file.

Statements (6) and (7) repeat the actions of statements (4) and (5) for the remaining six copies.

3. THE OPERATOR GUIDE

3.1 Introduction

There is a disc file on the system, SYS*PLOTDIRECTOR which holds information about each file of plotter information generated by user plot jobs and accounting information for each account under which any plot jobs have been run. This will be referred to as the plot directory and the files mentioned in it have to be transferred to 7-track magnetic tape before they can actually be plotted on the Calcomp 936. When the transfer to tape is complete the plot directory is cleared ready to receive information from further plot jobs.

It is the operator's responsibility to monitor the number of plots referred to in the plot directory and run the magnetic tape generation program at suitable intervals. The number of plots in the directory can be determined by use of the utility program PLOTS which displays the number of plots in the directory on the console. Each time a user executes the PLDIR processor to create a directory entry, the number of plots in the directory is displayed on the console. At suitable intervals the operator should start the canned runstream using the ST Keyin to execute the tape generation program. This program, PLCOPY will produce up to four magnetic tape files of plotting information. PLCOPY also produces output on the lineprinter to accompany each tape identifying the tape reel number, the type of plots it contains and an entry for each user plot which enables the operator to comment on any plotting difficulties. The lineprinter output contains accounting information for each plot and an indication of the length of time the plot will take to plot.

A magnetic tape generated by the spooling system consists of an enquiry block, to ensure the correct paper width is used, alternate blocks of header information and user plots and, after the final user plot, a message block to identify the end of tape (see Figure 4).

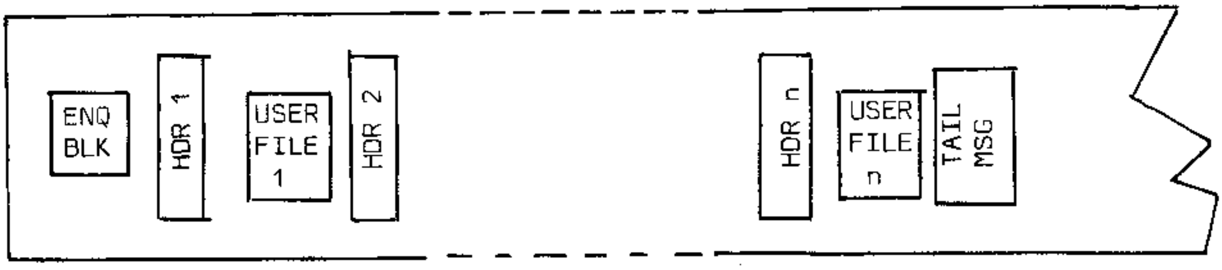


FIGURE 4

The tape should be plotted by use of the multiple plot feature on the plotter control unit. The first multiple plot consists of the enquiry block. When this has been plotted the plotter will stop thus allowing any necessary paper movement. The next multiple plot block will produce the header plot block informing the operator of any special requirements and allowing the necessary pens to be set up. This is followed by the user's plot. The sequence of header plot block followed by user's plot continues until the last user plot is complete. The last user plot is followed by a multiple plot block trailer message.

3.2 The magnetic tape generation program (PLCOPY)

3.2.1 *Magnetic tape description*

PLCOPY examines the system plot directory and arranges the plots in the directory into one of four types according to paper width and pen type.

The plot types are:

(A) All plots requiring 15 in paper and only biros (15 in Biro plots)

(B) All plots using 36 in paper and only biros (36 in Biro plots)

(C) Plots requiring 15 in paper and ink pens (15 in ink plots)

(D) Plots requiring 36 in paper and ink pens (36 in ink plots)

Each of these categories is written to a separate magnetic tape file. If no user files are present for any category, the tape for that category is not requested. Within each category each user plot is arranged in order so that the number of pen changes is reduced as far as possible. Any one category may span more than one magnetic tape but if this happens the user plot which encountered the end of the tape will be repeated at the start of the next tape. The lineprinter output should be examined to ascertain the last complete plot on each tape.

3.2.2 *The lineprinter output*

The lineprinter output can be divided into sections. Any file which has an entry in the plot directory but is either not present in the master file directory or has reserved no space on mass storage will be listed at the head of the PLCOPY output to enable the operators to answer any queries regarding a missing plot.

The remaining output consists of one section for each magnetic tape generated. Each section contains a page heading which identifies (by reel number) the tape with which it is associated and the type of user files contained on the tape. This is followed by a tabular list of all the user files present on the tape. This list contains the RUN-ID and ACCOUNT-NUMBER of each user file, the cost to the user of the plot, and the time which the plot should take to be plotted. There is a final section for each plot which is set aside for any comments the operator may wish to enter. The comment section is used by PLCOPY to hold error messages.

3.2.3 *PLCOPY messages*

Before commencing execution PLCOPY displays message (27) on the console. This may be followed by message (13), if there are no plots in the directory, or by either message (14) or message (15) for the first category of plot (15 in BIRO plots). If there are category A plots present, a tape load message will be displayed and the tape for this category of plots will be written. This will be followed by either message (14) or message (15) for category B plots (36 in Biro). Similarly, for categories C and D. If any of the four categories requires more than one tape then message (19) will be displayed and another tape load requested. If the tape change occurs in the first user file, message (18) is produced and PLCOPY terminates in error as this is the only file on this tape and it will not fit. If the tape change occurs in the trailer message, message (17) is displayed on the console but no tape change occurs. If during the process of creating any tape PLCOPY encounters a user file which is assigned to another run, PLCOPY will terminate in error and generate no output, first displaying message (24).

PLCOPY's lineprinter output contains a list of messages(25) and (26) for all directory entries which do not require plotting. The comments section of each table will contain any of the four messages(20),(21), (22) or (23) for each plot which terminated in error.

3.2.4 *Restrictions to the use of PLCOPY*

PLCOPY may be run at any time without regard to other users. If, however,any other user is using a plot file this will cause PLCOPY to display message (24) and terminate. The directory will be left intact and any files already transferred to tape will be repeated at the next run of PLCOPY. Thus if PLCOPY errors producing message (24) it may be re-run as soon as the job with the stated RUN-ID finishes. It is not necessary to plot the tapes produced by the PLCOPY which errored.

3.3 The directory enquiry program (PLOTS)

PLOTS informs the operator of the number of plots in the directory at the time of running. The operator enters a ST keyin to start a batch job called PLOTS which produces no lineprinter output but displays on the console the message:

```
'THERE ARE NNN PLOTS IN THE DIRECTORY'
```

This enables the operator to decide whether it is necessary to run the magnetic tape generation program (PLCOPY).

3.4 Accounting

The spooling system automatically accounts for each user plot by counting the number of plot records which the user's plot file contains. A factor is used to calculate the time required to plot this file from the number of plot buffers. Another factor (cost per hour)is then applied to this to give the cost of the plot.

This information is accumulated in the accounting section of the system plotfile directory on an account basis.

A subroutine (PLCHG) is provided by which the account section of the system plotfile directory may be accessed on an account number basis from an ASCII FORTRAN program; this routine returns the accumulated charge and total plot time for this account in this accounting period. The call to this routine should be:

```
     CALL PLCHG(ACC,CHG,TIM)
```

where ACC is the account number for which the information is required, stored left justified space filled in a CHARACTER*12 variable.

CHG is the cost in pounds in this accounting period.

TIM is the time in minutes accumulated in this period.

A utility program ZERO is also provided for the operators to rest the accounting information in the system plotfile directory to zero for each account when the totals for an accounting period have been produced.

The operator is provided with the program ACCHG which allows the information in the accounting section of the system plotfile directory to be charged manually to permit plots not produced by the operating system to be charged for at the same rates. ACCHG provides questions which assist the operator to perform the accounting update.

4. SYSTEM OVERVIEW

4.1 Spooling system concepts

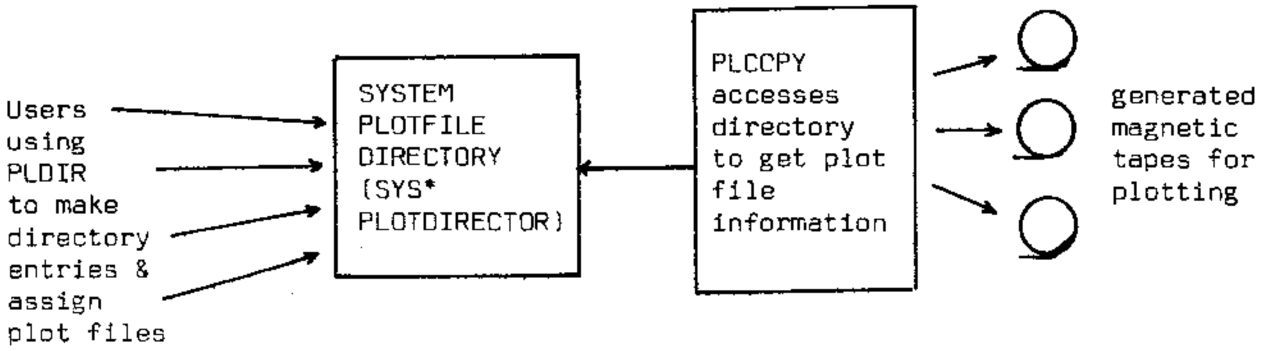

Users using PLDIR to make directory entries & assign plot files
SYSTEM PLOTFILE DIRECTORY (SYS* PLOTDIRECTOR)
PLCCPY accesses directory to get plot file information
generated magnetic tapes for plotting

FIGURE 5

The spooling system is designed around the concept of a plot directory. The user makes an entry in the system plotfile directory (a file catalogued on the system with the name SYS*PLOTDIRECTOR) by a call to the processor PLDIR,which updates the directory by adding a one sector record containing the pen and paper requirements as detailed in 4.2. PLDIR then assigns a disc file into which the user will write his plotting information.

The magnetic tape generation program (PLCOPY) references the system plotfile directory to determine what plot files exist. PLOOPY transfers this information to magnetic tape in the form of a header block giving details of the user who submitted the plot, when the plot was generated, the user's pen requirements, the cost and plot time of the plot, the number of copies to be plotted, and the user's comments.

The spooling system suite also contains three utility programs as follows:

(i) PLOTS allows a console operator to determine the number of plot file entries in the directory at any time.

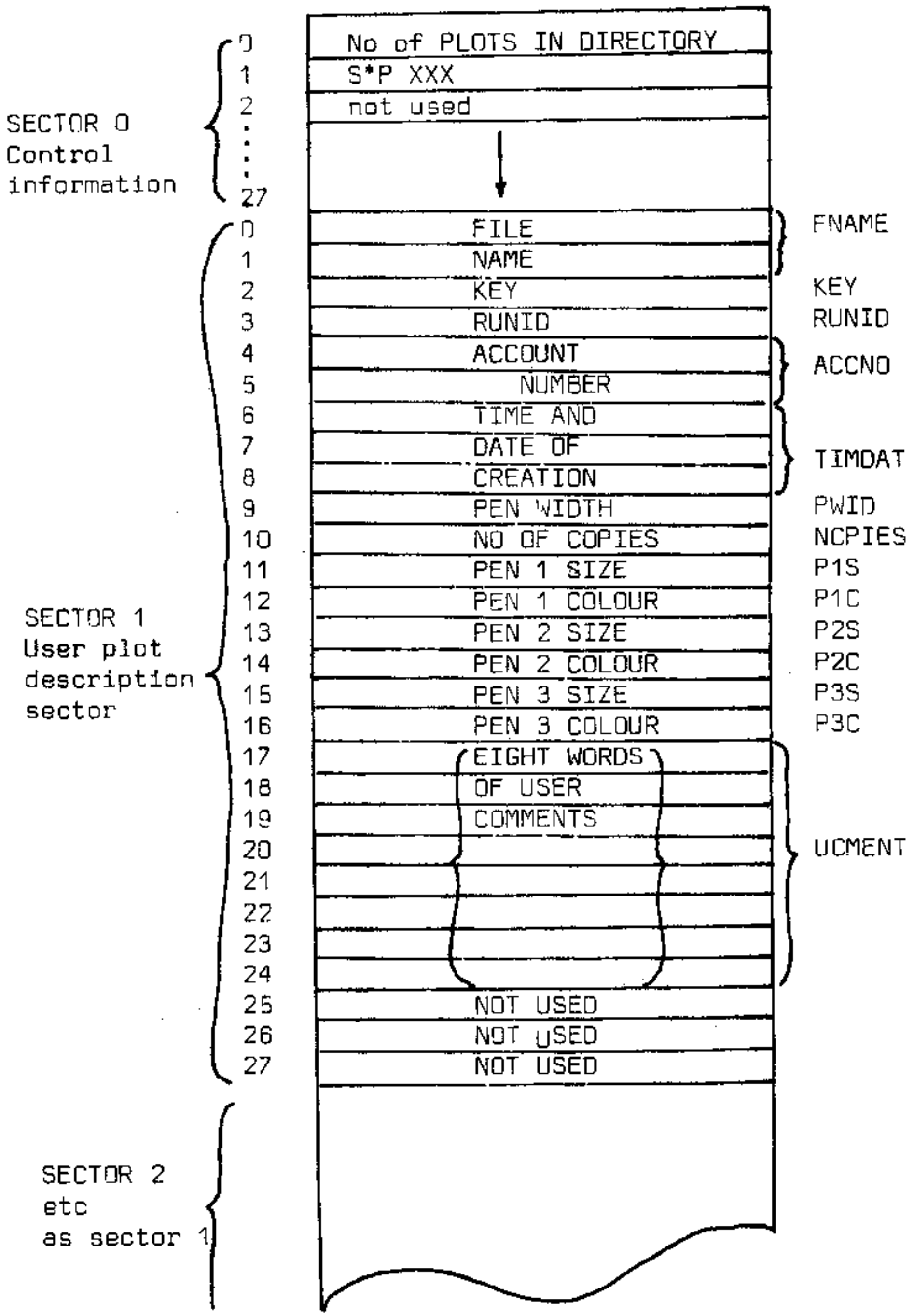
(ii) DIRCRE allows an empty directory to be built.

(iii) DIRBOJ allows a directory to be rebuilt in the event of an error.

4.2 Plot file design (SYS*PLOTDIRECTOR)

4.2.1 *User Entry Section*

FIGURE 6



The system plot file user entry section has one sector of control information which contains the number of plots currently in the directory. This number is held in word O as a binary number and in word 1 as fieldata characters in the right half word with the characters S*P in the left half.

Word O is used by the processor PLDIR to find the next available sector when a user requires a new plot file directory item.

Word 1 is used by PLDIR when generating a unique filename for the user's plotfile and also by PLOTS to generate the message informing the operator how many plots there are in the directory.

The remaining 26 words of the control sector are not used.

The rest of the plot file directory user entry section consists of 28 word records, each record describing one plot file currently waiting to be plotted. As each record is exactly one sector long the created mass storage file is written as efficiently as possible.

The first two words of the record contain the 12 fieldata characters which constitute the unique filename of the file which contains the user's plotter instructions. This is followed by a binary key which is formed from the pen sizes and colours to be used in the plot. This key is used when creating the magnetic tape to sort the plots into the most efficient plotting order.

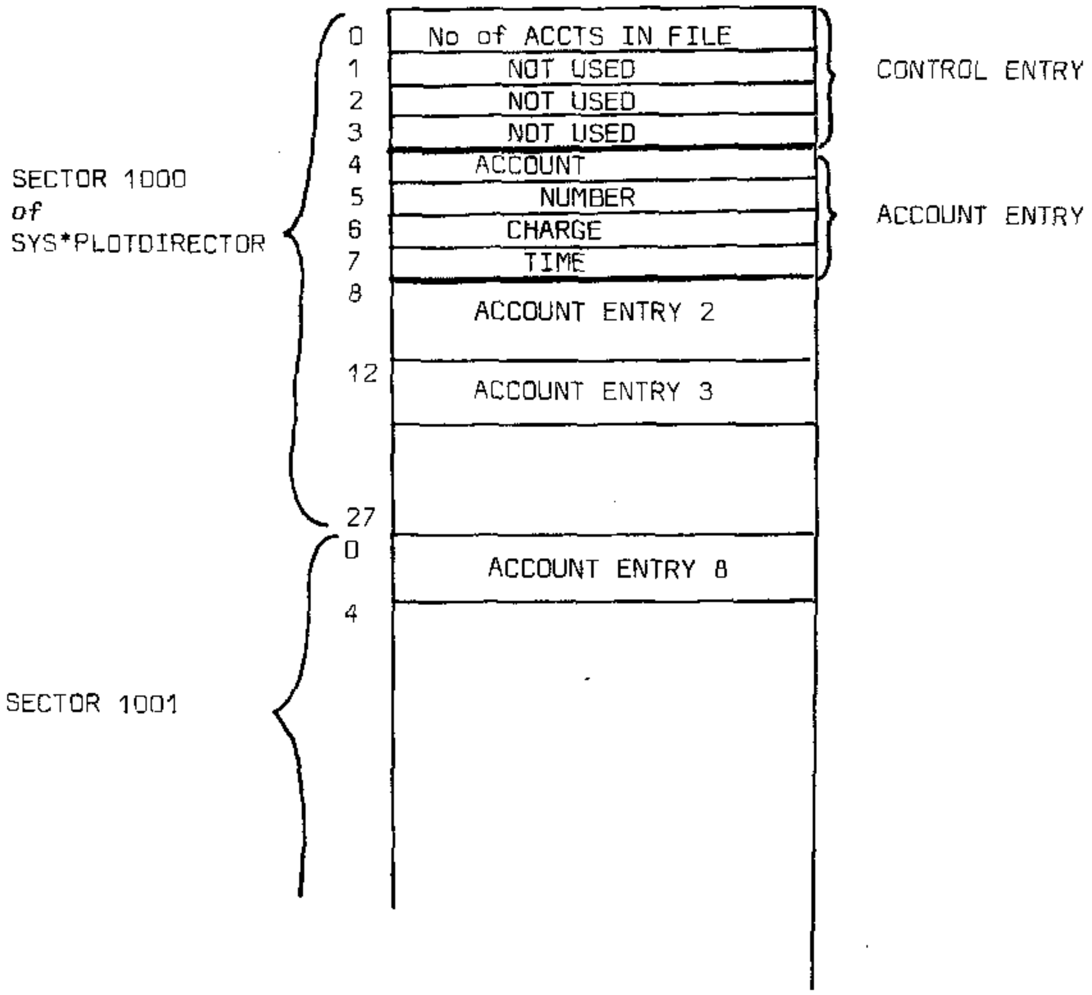
The remainder of the record is written in field data characters; the first 6 characters are the RUNID, followed by the 12 characters of the Account Number and 18 characters representing the time and date at which PLDIR was called by the user. This is followed by the user's plotting requirements which are all 6 characters long. The first of these is the paper width in inches and must be either 15 or 36; this is followed by the number of copies to be plotted which must be a number between 1 and 9 inclusive. The next 6 words represent the size and colour respectively of the pen required in pen 1, pen 2 and pen 3 positions. If a pen is not used the appropriate fields will be blank filled.

Words 17 to 24 of the sector are used for a user supplied comment which will be plotted next to the plot block. If no comment is supplied then this section of the record will be blank filled.

Words 25-27 of the directory sector are unused.

4.2.2 *Account information section*

FIGURE 7



The accounting information is written to the system plotfile directory (SYS*PLOTDIRECTOR) starting at sector address 1000. This allows enough room at the start of the file for a one sector entry for the maximum permissible number of user files in the plotfile directory. The accounting information consists of a control entry of four words in length followed by a four word entry for each account which has incurred any plotting charges during the previous accounting period. The control entry uses only the first word of the four word entry and stores in this the number of accounts currently held in the account section.

Each account entry consists of four words. The first two of these contain the account number in FIELDATA characters, held left justified, space filled in the available words. Words 3 and 4 are used to hold the total charge in the current accounting period (in pounds) and the total estimated plotting time (in minutes) in this period. These values are held as real numbers.

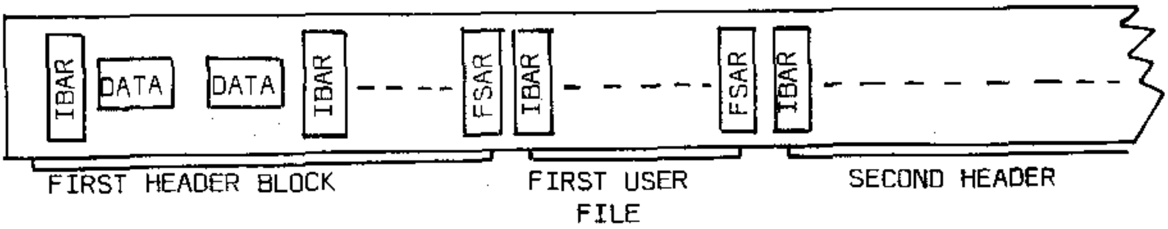
The information held in the account entry is calculated for a plot by subroutine CHGE and added to the information already in the account entry and written to the entry by WRCHG.

A utility subroutine PLCHG is provided to allow access to the accounting information by account. This subroutine must be supplied with the account number for which the information is required, left justified space filled, in a CHARACTER*12 variable and it will return the charge and plot time in the accounting period.

A utility program ZERO is also provided to enable the operator to clear the charges and plot times to zero after a successful run of the accounting program.

4.3 Plot tape construction

FIGURE 8



A plot tape consists of three types of record, these are intermediate block address records (IBAR), final search address records (FSAR) and data records. If the plotter is operated in single plot mode it will cease plotting each time an IBAR is encountered and will only recommence when the operator restarts the plotter. If the plotter is operated in multiple plot mode then each intermediate block will be plotted successively and plotting will only be interrupted by the presence of an FSAR on the tape. The plotter is usually operated in multiple plot mode and a FSAR exists only at the end of each user plot. An FSAR is generated by a CALL PLOT (X,Y,999) statement in the user's program.

An intermediate block address record is a five word record which contains two words of fixed header information which the plotter hardware will interpret while the remaining three words contain a block sequence number.

A final search address record is also five words long and consists of two words of header information which identify the record as an FSAR to the hardware. This is followed by three words of zero, which complete the record.

Data records are supplied by the user through the Calcomp plot routines and may be up to 114 words long. Data records contain the user's plot codes and are interpreted by the plotter hardware in terms of pen and drum movements.

A spooling system plot type consists of header blocks (see Figures 1, 2 and 3) and users' files placed alternately along the tape separated by FSARs to allow the operator to reset the pens to the users' requirements before each user's plot. A typical plot tape structure is shown in Figure 8 .

The users disc files from which a plot tape is created may have been written using either FORTRAN V or ASCII FORTRAN. If FORTRAN V was used each record on disc will contain three control words which will be stripped from the record when the file is being transferred to tape. ASCII FORTRAN uses an assembler subroutine to create the disc file and no control information is written to the disc file.

4.4 Site installation

The multiple graph plot magnetic tape generation package is constructed from a set of ASCII FORTRAN and assembly language routines which may be built into a suite of six absolute programs. These absolutes comprise the processor PLDIR, the tape generation program PLCOPY, and four utility programs PLOTS, DIRCRE, DIRBOJ and ZERO.

To install the spooling system at a site it is necessary to catalogue a file which contains copies of the prestored runstreams and executable versions of the programs. It is also necessary to place a copy of the processor PLDIR in the system library (SYS$*LIB$). A blank system plotfile directory (SYS*PLOTDIRECTOR) may be created by execution of the program DIRCRE (@XQT file.DIRCRE); this program must be executed in a run whose project-id is SYS because PLDIR and PLCOPY expect the plotfile directory to be called SYS*PLOTDIRECTOR. This should be followed by running the program ZERO (@XQT file.ZERO) to zero out the accounting section of the directory.

If the preceding process is carried out the spooling system installed will contain the configuration parameters used in the relocatables supplied. There are several global parameters applied to the spooling system which may be changed when generating the spooling system. These parameter values are held in a proc called PERAM which is held in element DIRPROC. The following table describes the global parameters, their meanings and values in level 5.0 of the spooling system.

TABLE 1

NAME	VALUE	MEANING OF PARAMETER
FSIZ	500	max. size in tracks as a binary number of the user's mass storage file
FSIZC	'500'	max. size in tracks as a character string of the user's mass storage file
TABLES	128	length in words of the internal sorting tables for each tape category in words ((max.no.of plots)*2)
TESTFG	0	Test flag:if = 1 turns on code to generate a parallel test system for new updates
TTAP	'8C '	The tape unit type required for the spooling tapes
LVL	'5.0'	The current level of the program suite
MAXL	114	The maximum length of a Calcomp data record
BFTIM	0.05	The coefficient relating no. of plot records to plotting time
CPHR	8.0	Charge for plotting (in pounds per hour)

5. THE DIRECTORY ENTRY PROCESSOR (PLDIR)

5.1 Introduction

PLDIR is a processor which creates an entry in the system plotfile directory and assigns a mass storage file into which the user writes his plotting information. PLDIR consists of a main program in element DIRUPD and uses subroutines KEYWD, STATUS, and TDATE.

Subroutine KEYWD is used to produce the key, held in word 2 of the user's directory entry sector (see Figure 6) in the plotfile directory. The key is derived from the pen sizes and colours and is used to minimise the number of pen changes required to plot any spooling system generated tape.

Subroutine STATUS is a general purpose subroutine which interprets the facility status bits returned from a CSF$ request and either returns control to the calling routine or creates a program error when particular status bits are set. The error status bits are defined by the calling routine which supplies a mask to STATUS.

Subroutine TDATE writes the program header line which identifies the processor and its current level. TDATE also writes the date and time at which the PLDIR call was made.

The main routine DIRUPD can be conveniently divided into three sections, firstly initialisation, then reading and interpretation of the user supplied data and finally creation of the new directory sector, updating the control sector and assignment of the user's plot file.

This section should be read in conjunction with the flow diagrams shown in Appendix B, messages in Appendix A and the program listings.

5.2 The main routine (DIRUPD)

5.2.1 *Initialisation*

The first action of DIRUPD is to call subroutine TDATE to write the program header. The time and date used are those given in registers R1 and R2 on entry to the program. The time and date calculated by TDATE are then compressed into three words and stored in the incore copy of the directory entry sector (directory buffer).

5.2.2 *Interpretation of user data*

This section of DIRUPD reads and interprets user supplied data and inserts the information into the directory buffer. The first step is to submit a read command to the operating system. If the end of file return address is used then the user is accepting the default pen and paper settings of 1 black biro in pen 1 position (set into the directory buffer initially) and control passes to the completion phase (see Section 5.2.3).

If the user has called PLDIR as a processor, the first read will return an INFOR table which should either be blank if the default channel (8) is used or will contain the user's channel number; the first read will otherwise return the contents of the pen description card.

When the return from the first read is an INFOR table, internal subroutine INFR is used to interpret the INFOR table. INFR obtains the first specification field from the table and checks it to be either blank or a valid FORTRAN channel number (less than three characters); if this is not so, error (2) will be printed. If the field is blank the default value of channel 8 is used. If the INFOR table is too long then error message (1) will be printed. The channel number from the INFOR table is written into the USE name image and into the catalogue and assign images for the user's file. INFR then performs another read to obtain the contents of the user's pen description card and unless the end of file return address is taken on the read (in which case the default pen and paper settings will be used) control passes to the completion phase (see Section 5.2.3).

When the INFOR table, if any, has been interpreted, DIRUPD prints the pen description card. This is done to produce suitable output for batch jobs where this is the only copy of the pen description card to be produced in the lineprinter output from the job. The pen descriptions are then placed in words 9 to 16 of the incore directory buffer. If an error is encountered while interpreting the pen description card or if the paper width is an illegal specification then error (3) will be printed and the processor will take an error exit. A check for the number of fields present on the pen description card is also made; if this is incorrect error message (4) is printed and the processor terminates in error. A final check is made on the pen description card to ascertain the number of copies the user requires; if this number is greater than 9 the number of copies produced is set to 1 and error message (5) is printed. This error is non-fatal and the next card is then read.

When the pen description card has been processed the final data card is read into words 17 to 24 of the incore directory buffer. If the end of file return address is taken the card is not present and it is assumed that there is not user comment. As with the pen selection card, the comment card, if present, is mirrored back to the user's runstream. if present, the comment card is checked to be less than 48 characters in length. If this check fails error message (6) is produced and the character string is truncated to 48 characters. This error is non-fatal.

5.2.3 *Completion phase*

The final section of DIRUPD completes the incore directory buffer and updates the directory, then assigns the user's plot file and associates it to the correct channel number.

The first stage in this process is to obtain the user's run-id and account number by an ER PCT$\emptyset$. An attempt is then made to assign the plot directory (SYS*PLOTDIRECTOR); if this fails the status bits are examined for exclusive assignment to another run. If this bit is set the processor will wait for 1 sec and then try again to assign the directory. Certain error stati will cause PLDIR to error after printing the status word. When a successful assignment of the plot file directory has been made, the control sector is read and this information is passed to internal subroutine FUPD.

FUPD first uses word 1 of the plot directory control sector (Figure 6) to obtain the current directory entry sequence number in characters. This is incremented by one and written back to the control sector buffer ready to update the directory. The new sequence number is also stored in the use name image preparatory to assigning the user's file. At this stage the sequence number is also used to display message (8) on the operator's console. The plotfile name is then constructed using the following algorithm which ensures that it is unique and easily recognised.

(a) The qualifier is always SYS.

(b) Character 1 of the file name is P.

(c) Characters 2 → 4 are a sequence number taken from the information in the control sector of the plot directory.

(d) Characters 5 and 6 are the paper width.

(e) Characters 7 → 12 are the run-id of the user who created the file.

This algorithm permits a maximum of 999 plots in the directory at any time. (If there are 999 plots in the directory when PLDIR is called, error message (7) will be displayed and the processor will terminate in error.) The generated user file name is stored into the use name image and into the incore directory buffer. FUPD's final action is to store the runid and user's account number from the PCT buffer into the incore directory buffer. Control is then passed back to DIRUPD.

DIRUPD next calls subroutine KEYWD to generate the pen size/colour key which is stored in the incore directory buffer. This completes the generation of the incore directory buffer. DIRUPD then increments the 'in use' sector count in word 0 of the directory control sector and writes out the control sector. This is followed by writing the newly created incore directory sector and freeing the directory for other users. The final task of DIRUPD is to assign the user's plot file; the file name is first associated to the user supplied channel number, then the file is placed in the master file directory and assigned to the run. After each CSFØ call a check is made (using subroutine STATUS) on the facility status bits; if the status is incorrect an error exit is taken after displaying the status bits in message (9).

5.3 Subroutine TDATE

Subroutine TDATE prints the call line (message (10)) on initial entry to a program. TDATE takes the date in Fieldata as supplied in R1 and spaces the characters inserting / between the month, day and year. The modified date is placed in words 6 and 7 of the call line.

TDATE then takes the date supplied in seconds past midnight (in register R2) and converts this to fieldata characters. The resultant six-character number is spaced and : is inserted between hour, minute and second. The modified time is placed in words 8 and 9 of the call line. Finally, the call line is printed and control returned to the calling program.

5.4 Subroutine STATUS

Subroutine STATUS checks the facility status bits returned by a CSFØ request against a user supplied mask (in A1) and terminates the job in error, after printing message (9), if any of the bits set in the mask are set in the returned status word.

Subroutines TDATE and STATUS are as described in Sections 5.3 and 5.4 respectively.

Subroutine NOSE is used to write the entry message (message 11) for each tape being generated. NOSE uses the common block VALS to determine which width is being used.

Subroutine TAIL is used to write the trailer message (message 12) on each tape generated. It is necessary for a final message to be written because the end of a plot tape is otherwise not obvious. TAIL uses common block VALS to determine the paper width.

Subroutine HEADER is used to write the user's plot block (see Figures 1, 2 and 3) before each user file is written to tape. HEADER uses common block VALS to determine all the values which will be placed in the header block.

Subroutine COPY copies a user's plot file from mass storage to magnetic tape. COPY will deal with mass storage files written by either FORTRAN V or ASCII FORTRAN (using assembler subroutine WRIT to create plot files).

Subroutine ORDER is used to arrange the keys from word 2 of the directory entry sectors (Figure 6) in ascending order for each tape. ORDER creates four tables, one for each tape type, which contain two word entries. The first word of each entry is the keyword while the second is the sector address of the directory sector for that file. The maximum number of files allowed in any table is configurable (see Table 1) by the label TABLES and is set to 64 files in level 5.0.

Subroutine PRHEAD writes out the heading on the operator's lineprinter output (see Figure 10). The lineprinter output is written with a new page for each magnetic tape type and a new page for every 40 plots within a tape type. Thus PRHEAD is called once for each tape type and every 40 plots within a tape type.

Subroutine LINES creates an entry in the operator's output for each user plot. This entry contains the run-id and account number, the cost and length of time the plot will take to plot, and error information pertaining to that user plot (see Figure 10).

Subroutine CHGE is used to calculate the number of records in a user's plot file and by applying the conversion factors BFTIM and CPHR (see Table 1) estimates the expected plotting time in minutes and the cost of the plot in pounds.

Subroutine WRCHG is the other accounting subroutine and is used to add the cost of a graph and the calculated plot time to the totals already accumulated in the accounting section of the plot directory.

PLCOPY can be divided conveniently into 4 functional sections:

(1) Initialisation and organisation of the plots in the directory into

The only colours recognised by KEYWD at present are shown with their key values in Table 2; any other colour not shown in the table will take a pen colour index of 076 (the largest index used). The pen size index assumes that the pen size only contains one digit or is the word BIRO. For ink pens the index is calculated by subtracting 060 from the fieldata character value of the pen size. If the pen is a biro the index is set to 016 (the largest code present for a pen) and if the field is blank (the pen position not required) the index is set to 017, thus requested pen positions have the highest key value and appear in the least significant positions in the full key.

TABLE 2. Pen colours available and their indices in the current KEYWD.

Pen Colour	Colour Index
BLANK FIELD	0
BLACK	2
BLUE	4
RED	6
GREEN	8
BROWN	10

6. THE MAGNETIC TAPE GENERATION PROGRAM (PLCOPY)

6.1 Introduction

PLCOPY is a UNIVAC operator's program used to collect all the files present in the system plot directory and copy them to magnetic tape. PLCOPY is started periodically by the computer operators when the number of plot files present in the system warrants writing some more plot tapes. PLCOPY consists of a main program held in element TAPEOUT which uses FORTRAN subroutines NOSE, TAIL and HEADER and assembler subroutines COPY, TDATE, STATUS, LINES, PRHEAD, ORDER, CHGE and WRCHG. TAPEOUT also uses the standard Calcomp plotting routines SYMBOL, NUMBER, PLOTS and PLOT and a modified version of the Calcomp routine BUFF which uses an assembler subroutine (WRITERA) to write the information to magnetic tape.

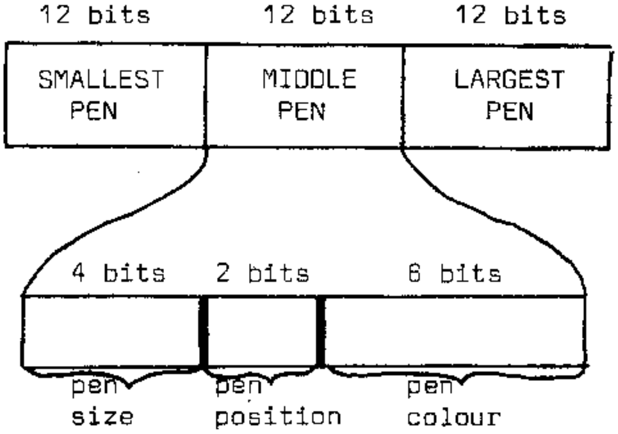
If none of the bits set in the mask are set in the status bits returned from the CSF∅ request, control is returned to the user.

5.5 Subroutine KEYWD

Subroutine KEYWD is used to generate the key which is used to sort the plots on a given tape into the order which minimises the necessary pen changes. The key ensures that pen size changes between user plots will only occur in ascending order of minimum pen size, ie any plot which used a 0.2 mm pen in any position will be plotted before all plots using only 0.3 mm or larger pens. Within this classification all pens using a pen of the same size in pen 1 position will be plotted before any plots using the same pen in pen 2 position. Also all plots using the same colour in the same position with the same pen size will be plotted sequentially. In the context of pen size a biro is classified as the largest pen size.

The key is constructed from three 12-bit fields; each field is a coded representation of the pen size, its colour and its position. The three 12-bit fields are assembled into one 36-bit key such that the field for the smallest pen is the leftmost (see Figure 9).

FIGURE 9



Each of the 12-bit fields is arranged such that the 6 least significant bits represent the pen colour index, the next 2 bits represent the position index and the 4 most significant bits represent the size index.

order in the tape type tables.

(2) Preparation of the next tape to be plotted; if the last tape has been plotted go to section 4.

(3) Transfer all the mass storage files of one type to a magnetic tape with the appropriate plot blocks; when finished return to section 2.

(4) Termination functions.

6.2 The main routine TAPEOUT

6.2.1 *Initialisation*

The first task of the initialisation section of TAPEOUT is to create and print the time and date message. This is done by a call to subroutine TDATE (see Section 5.3). TAPEOUT then completes the graph tape assign image by inserting the correct unit assign mnemonic (configured by the tag TTAP as shown in Table 1). This is followed by an attempt to assign the plot directory; if the assign fails because the directory is assigned to another run, the program will wait for one second and then try again. When the plot directory has been assigned, TAPEOUT reads the control sector and decides if there are any plots to be copied. If there are no plots in the directory, message (13) is displayed on the console and PLCOPY terminates. If there are plots in the plot directory the final task of the initialisation section of TAPEOUT is to arrange the copying of the plots in ascending order of key within tape type. This is done by a call to subroutine ORDER (See Section 6.7). Once the tape type tables have been created the initialisation section is complete.

6.2.2 *Magnetic tape preparation*

The magnetic tape preparation section must first decide if the last tape type table has been processed. If it has, the program proceeds immediately to the termination section (6.2.4) but if there are tape type tables to be processed TAPEOUT inserts the correct tape type information into the output information buffers, sets the tape type table counter for the next table and checks how many files are in this table. If there are no files of this type the operators are informed using message (14) and the next tape type tried. Having found a tape type table with some files to be transferred from mass storage to tape, TAPEOUT must assign a graph tape and inform the operators (message (15)) which type of plot the tape will contain. The tape assignment is checked using subroutine STATUS (5.4). TAPEOUT continues the preparation phase by calling subroutine NOSE to write message (11) on to the magnetic tape and then places the reel number of the graph tape currently being used into the heading message (16) for the operator's lineprinter output. Finally the heading message (16) is written to the lineprinter and the graph and sector counts are set to zero.

6.2.3 *The magnetic tape generation*

The first task in this section is to check the number of files transferred against the number of plots in this tape type table. If TAPEOUT has just completed the last plot in the table, subroutine TAIL is called to write message (12) to the graph tape, the graphtape is freed and control returns to 6.2.2 to examine the next tape type table. If there is still a plot to copy to the tape, TAPEOUT reads the appropriate directory sector, and, from this, sets up the lineprinter entry for this file using the run-id and account number in the directory sector. TAPEOUT next assigns the user's plotfile and calls subroutine CHGE to calculate the cost and the estimated plotting time for the plot. The number of copies to be plotted is obtained from the directory sector and the time and cost adjusted appropriately. TAPEOUT calls subroutine HEADER to write the header block to tape (Figures 1, 2 and 3), calls subroutine WRCHG to amend the accounting information held in the directory to include this plotfile, and writes the correct number of copies of the plot to the tape. TAPEOUT next investigates the number of lineprinter plot entries written out since the last page header. If 40 files have been written then subroutine PRHEAD is called to put out a new page header. TAPEOUT then writes the lineprinter entry for this user's plot file to the lineprinter and returns to the start of the section to search for the next plot to transfer to this tape.

It is possible that at some time during the writing of information to the magnetic tape the physical end of a tape will be encountered. If this happens TAPEOUT frees the tape and then determines which sections of code were being used when the error occurred. If TAPEOUT were using subroutine NOSE to write the initial entry message (11) to tape when the end of tape was met, PLCOPY will terminate in error because message (11) is written only once at the head of each tape. If TAPEOUT were using subroutine TAIL to write the final message to the tape, the operators are informed using message (17) that the error has occurred but PLCOPY will return to section 6.2.2 to process the next magnetic tape.

The only other possible places where an end of tape may be encountered are while using subroutine HEADER to copy a plot block or while using subroutine COPY to copy a user's file. In either of these situations the procedure is the same: TAPEOUT firstly assigns another graphtape, then checks that the file being copied was not the first file on the tape. If the file were the first on the tape, TAPEOUT assumes this file will be impossible to plot, writes message (18) to the console and terminates in error. Otherwise subroutine NOSE is called to write a new initial entry message (11). TAPEOUT then writes a new header to the lineprinter (message (16)), containing the new reel number and writes message (19) to the console. Control is then returned at the start of 6.2.3 to continue on the new tape.

6.2.4 *Termination phase*

This section is entered only from the magnetic tape preparation section 6.2.2 when the last tape type table has been processed. The only task of the termination section is to reset the system plotfile directory control sector to register that the directory is empty. This function is performed immediately before exit so that any error earlier in the program will leave the directory untouched and no user generated information will be lost.

6.3 Subroutine NOSE

Subroutine NOSE is written in ASCII FORTRAN and is called by TAPEOUT *to write the initial block (message* (11)) on to the graph plotter. The tape type (either 15 INS BIRO, 36 INS BIRO, 15 INS INK or 36 INS INK) is passed across to subroutine NOSE from TAPEOUT in fieldata characters in the first two elements of the array ARRAY in common block VALS.

6.4 Subroutine TAIL

Subroutine TAIL is written in ASCII FORTRAN and is called by TAPEOUT to draw the final plot block (message 12) on the graph plotter. The final block informs the graph plotter operator that the last user's plot file has been plotted. Subroutine TAIL determines the paper width from variable IPW in common block VALS.

6.5 Subroutine HEADER

Subroutine HEADER is written in ASCII FORTRAN and is called by TAPEOUT to draw the user's pen description box shown in Figures 1, 2 and 3 and inserts the information from the directory sector and accounting information into the box. First, the outline of the box is drawn from *the data stored in arrays X,* Y *and IPOS at compilation time and is* positioned according to the paper width passed across from TAPEOUT in common variable IPW. The user's run-id is then read from the directory sector, translated from fieldata to ASCII and written to the header block. The account number and the time and date of creation of the user's directory entry are then treated in a similar manner to the runid. HEADER calls Calcomp subroutine NUMBER to write the real numbers representing the cost of the job and the estimated plot time to the header block. These are followed by the transfer of the user's pen requirements and the number of copies of the user's plotfile which will be plotted. Finally, the user's comment, if present, will be plotted as shown in Figure 2. The information present in the user's directory sector is made available to subroutine HEADER using array ARRAY in common block VALS and the accounting information is held in common block CHGS.

6.6 Subroutine COPY

6.6.1 *Introduction*

Subroutine COPY transfers plotter codes, written to a disc file by a user's FORTRAN program, from disc to magnetic tape. The plotter codes are copied to tape as Calcomp records. Subroutine COPY can be conveniently divided into four sections.

The initialisation section distinguishes plot file types and takes appropriate actions. The second and third sections read a Calcomp record from the subroutine's internal buffer and then write it to magnetic tape respectively. The fourth section is an internal subroutine for maintaining the internal buffer.

6.6.2 *Initialisation*

The first task performed by COPY is to read a track of information from the user's plotfile into an internal buffer (using internal subroutine FETCH (6.6.5)] and to distinguish between a file written by FORTRAN V or ASCII FORTRAN programs. This is achieved by examining the first word of the file; if it is a Calcomp control word the file was written using an ASCII FORTRAN program, otherwise the program was written using FORTRAN V. This section of code then sets a flag to represent the compiler type and passes a Calcomp control word to Section 6.6.3 which will read the rest of the record.

6.6.3 *Output record construction*

Firstly a check is made that a Calcomp control image is being read; if not, a dummy final search address record (FSAR) is written to tape and control returns to TAPEOUT placing error message (20) into the operator's lineprinter output for this file. Next the Calcomp control image is used to distinguish between control records and data records.

A control record is always five words long, so if this is being constructed the remainder of the record is read and the input buffer pointer is incremented to point to the start of the next sector. A check is made to ensure that the pointer is still within the limits of the current input buffer; if not, a new buffer is read using subroutine FETCH. The next control word is then read and control passes to 6.6.4 to write the current record to magnetic tape.

A data record may contain up to 114 words of information, so first a check is made on the number of words read in this buffer; if this is greater than 114, error message (21) is sent to the operators lineprinter output and a dummy FSAR is written and control returns to TAPEOUT. The length of a data record is determined by searching for the next control word. When this has been found, COPY back-tracks over any control information and words containing zero to find the end of the last record. If the plotfile were written using a FORTRAN V program this will include the two trailing control words of each record. When reading consecutive words from the incore plotfile buffer the read routine FETCH will maintain the incore buffer.

6.6.4 *Write a Calcomp record to magnetic tape*

Firstly, the word count in the output packet is updated to reflect the number of words in the buffer which will be written. The buffer is then written to tape and the status is checked. If the end of the tape has been reached a flag is set and control returns to TAPEOUT. Any other status will cause PLCOPY to terminate in error.

If the write operation completes normally, a check is made to see if the buffer just written was the final buffer. If so, the task is complete and control returns to TAPEOUT; if not, the output buffer pointer and the word count are reset, the buffer count is incremented and control returns to 6.6.3 to construct the next record.

6.6.5 *Internal subroutine FETCH*

The standard action of this routine is to present the next word from the one-track internal buffer to Section 6.6.3 in register A1. However, if the word which has been read is the last word of the current buffer the next track of user's information is read into the buffer. Before reading the next track a check is made to confirm that the next track is within the maximum allowed file length. If this is not so, message (22) is written to the lineprinter output, a dummy FSAR is written to tape and control returns to TAPEOUT.

Having read in the next buffer, a check is made on the status of the read operation. If the status is correct the buffer address for the input packet is adjusted to point to the next track to be read and control is returned to 6.6.3; if the status is incorrect message (23) is written to the lineprinter output and control returns to TAPEOUT.

6.7 Subroutine ORDER

6.7.1 *Introduction*

Subroutine ORDER is used to work through the system plotfile directory validating each entry and then writing the key and sector address of the entry to the appropriate tape type table.

6.7.2 *File size and existence check*

A file size check is made to verify that there are still directory sectors to be processed; if not, control passes back to TAPEOUT. The directory sector address is then updated and the next sector read in. The filename found in this sector is then used to assign the user's plotfile. The status is checked and if the exclusive use facility bit is set, message (24) is sent to the console and PLCOPY terminates in error. If the plotfile does not exist, message (25) is written to the lineprinter and control returns to the start of this section to read the next sector. Sector 0 of the master file directory main item for this file is examined next and the total number of granules of all types of mass storage is calculated; if this is zero, message (26) is written to the lineprinter, the file is deleted and control returns to the start of this section to read the next sector. Finally, the file is freed and control passes to 6.7.3 to process this directory sector.

6.7.3 *File placement in the tape type table*

The first part of this code determines which tape type table should contain this plotfile. This is done by checking the required paper width and pen type. The key from the current sector is then checked against each successive entry in the correct table and the file inserted in its correct place in ascending order and control returns to 6.7.2 to read the next sector.

6.8 Subroutine PRHEAD

Subroutine PRHEAD writes the lineprinter output table header for each tape type and for each group of 40 files within type (see Figure 10). The page header contains the tape type and the graph tape reel number to which the table refers. Having written the table header PRHEAD returns control to TAPEOUT.

6.9 Subroutine LINES

Subroutine LINES is called once for each plotfile. It is used to produce an entry containing the account code, run-id and any error message pertaining to the plotfile, and to write this entry to the operator's output for a particular tape type (see Figure 10).

6.10 Subroutine CHGE

Subroutine CHGE is called once for each plotfile in the directory and calculates the cost and estimated plot time for a user's plotfile. CHGE reads the plotfile word by word (using internal subroutine FET) searching for a control sequence (IBAR or FSAR) and increments a counter when a control sequence is read. When the file has been read and the number of records calculated as an integer, the number of records is converted to floating point and multiplied by the factor BFTIM (see Table 1) to give the estimated plotting time (in minutes). This time is multiplied by the cost per minute, calculated from the supplied value of cost per hour (CPHR, (see Table 1)), to give the cost of the job and control is returned to TAPEOUT.

Internal subroutine FET is used to read the plotfile and functions in a similar manner to internal subroutine FETCH in subroutine COPY, passing back the next word in the file unless the end of the buffer is reached when a new track is read in.

If an error of any kind is encountered during execution of subroutine CHGE, the current value of cost and plot time is calculated and control returns to TAPEOUT. Interpretation of errors is left until subroutine COPY is called.

6.11 Subroutine WRCHG

Subroutine WRCHG updates the accounting section of the plotfile directory for each user plot. WRCHG may conveniently be divided into three sections:

6.11.1 *Account file analysis*

This section of code reads the first sector of the accounting information and uses the control entry to determine the number of account entries in the file. If there are no entries in the file this entry must be a new entry and control passes to 6.11.3. Otherwise, the

FIGURE 10

```
GRAPH PLOTTER SPOOLING SYSTEM NNNNNN XXXXXX PLOTS  GRAPHTAPE NO :- YYYYYY


RUNID      ACCOUNT NO      TIME(MINS)      CHARGE     COMMENTS

TEST       TESTACCT         26.62          £  3.55
PLOT       SYS1             13.20          £  1.76    03 CD PLOT TO DD
TONY       PLOT             34.95          £  4.66
ROGER      OPS               2.47          £  0.33    02     PLOT TO DD
RUNID      IH07              4.67          £  0.65
PLOTS      IH08              3.97          £  0.53
PLTONY     IP35             10.05          £  1.34
PLR03      SYS1              6.67          £  0.89    01     PLOT TO DD
PLMWV      OPS              18.07          £  2.41
PLRES      SYS1              8.32          £  1.11
PLPD       32-45            22.57          £  3.01
PLAH       OPS              14.55          £  1.94

$BRKPT PRINTE
```

account code of the plot file currently being copied to tape is compared successively with each entry in the account file. If a match is found, control passes to 6.11.2 to update an existing entry. After all entries in a sector have been compared, the next sector is read in and the comparison continues. If no match is found with any of the accounts currently present in the account file, control passes to 6.11.3 to create a new entry. All errors in subroutine WRCHG result in PLCOPY displaying message (28) on the lineprinter and terminating in error.

6.11.2 *Updating an existing entry*

When WRCHG finds a match between the account code of the plotfile being transferred to tape and the entry in the account file currently being examined, the cost and plot time information are added to the current total in the account file and the new information is written back to the account file. Control then returns to TAPEOUT.

6.11.3 *Creating a new account entry*

When the last entry in the account file has been read and no match has been found for the current plotfile, WRCHG tests the position of the last valid entry in the current sector and if it is the last entry WRCHG resets the pointers to reference a new entry at the start of the next sector. The account code of the current plotfile, the cost of the plot and the plot time are written to the new account entry and the entry is written back to the account file in its correct place. WRCHG next reads the first sector of the accounting section of the plotfile directory, increments the number of accounts present and rewrites the first sector. Control then passes to TAPEOUT.

7. UTILITY ROUTINES

7.1 Program DIRCRE

The program DIRCRE must be run when installing the spooling system at any new site. DIRCRE initialises the control information kept in sector 0 of the plotfile directory. The plotfile directory is set as an empty directory. DIRCRE should be run with the correct project-id (that which PLDIR and PLCOPY will be using to assign the directory). It is recommended that the project-id should be SYS and in any case the last letter of the project-id must be S.

7.2 Program HOWMANY

HOWMANY is an operator's utility which enables an operator to run a batch job which will display on the console the number of plotfiles currently present in the plot directory. It achieves this by reading sector 0 of the directory (Figure 6) and writing the information therein to the operator's console.

7.3 *Program DIRBOJ*

DIRJOB enables corrections to be made in the system plotfile directory in the case of directory error or corruption. The offending entries may be eliminated from the directory but extra entries may not be added. The program requires a list of filenames, consisting of those which are to be retained in the directory. The filenames supplied must appear in the order in which the directory sector entries occur in the directory, otherwise the filenames will not be found. The program first reads a filename from the runstream and then checks this name against successive names in the directory until a match is found. The contents of this sector are then written to sector 1 of the directory and the next filename is read from the runstream. The search of the directory continues from where the last find occurred. This process continues until an end of file is read when a new sector 0 of the directory is built and written to the directory.

7.4 Program ZERO

Program ZERO must be run when installing the spooling system at a new site and after every successful run of the accounting program. ZERO resets the control sector of the account section of the plotfile directory (SYS*PLOTDIRECTOR) to represent an empty account section. Before executing ZERO it is necessary to assign the plotfile directory and associate the name PLOTCHG with it by means of an @USE command. An error while executing ZERO will produce error message (29) in the output and an error termination.

7.5 Subroutine PLCHG

Subroutine PLCHG is an assembler subroutine provided to allow an ASCII FORTRAN program to access the account section of the system plotfile directory. The user program supplies PLCHG an account code as an argument held in a CHARACTER*12 variable. The routine reads the account section of the plotfile directory and returns the total cost and accumulated plot time to the user.

PLCHG first converts the user supplied account code from ASCII to FIELDATA characters so that a comparison may be made with the entries in the account file. The first sector of account information is then read and the control record is used to determine the number of account entries in the file; if there are no entries the values of time and cost are set to -1.0 and control returns to the user. Otherwise a test is made between the user supplied account and successive account entries. When the end of a sector is reached a new sector is read and the search for matching accounts continued. If a match is found the values in the account entry for total time and total cost are transferred into the user supplied variables and control returned to the user.

If the end of the account file is reached and no match is found then the values of cost and time are set to -1.0 and control returned to the user.

If an error is encountered while reading the account file, error message (30) is written and the user's program terminates in error.

7.6 Program ACCHG

This program consists of a main routine and an assembler subroutine which allows the operators to update the account section of the Plotfile directory manually.

7.6.1 *Main routine ACCHG*

This routine requires the operator to give the answers to questions concerning the plot accounts. On entry to the main routine the operator is asked question 31 to identify the account to be updated on this pass. ACCHG then calls its assembler subroutine at entry point GETUAL to retrieve the current values of time and charge for the account. These values are written in message 32. If the value of charge is zero then the operator is informed (message 33) and allowed to abort the program if a mistake has been made by an answer of N. This will produce message 34 and the program will stop.

If either the account exists or an answer to Y is given to message 33 the operator is requested to insert the value to be added to the charge (message 35). The values of charge and time are updated and the operator is asked to check this to be correct (message 36); if so, the values are rewritten to file and to the operator (message 38) using the assembler subroutine at entry PUTVAL and the operator is asked if any further accounts are to be modified (message 39). An answer of N will cause the program to stop while an answer of Y will cause the program to request the next account (message 31). If the answer to message 36 is N message 37 is written and the program returns to ask for the value to be added to this account with message 35.

7.6.2 *Subroutine ACSUB*

This subroutine has two entry points GETUAL and PUTUAL which obtain and return values to the account section of the plotfile directory respectively.

At entry point GETUAL the user supplied account number is converted to fieldata and the first sector of the account file entries is read. The total number of entries is used to ensure that the end of the file is correctly located. The fieldata account number is successively checked against the account entries and if the end of a sector is reached, a new sector is read until either the last entry is read or the required account is found. If an error is encountered in reading the account file, message 40 is displayed and the program terminates in error.

If no match was made on the account name, the position in the current buffer is saved and the disc address of the sector and a flag is set to indicate that no match was found. The values of charge and time are set to 0.0 and control returns to the user.

At entry point PUTVAL the new values of charge and time and account passed across are restored to the account code entry and this sector is rewritten to file. If this is a new entry, the first accounting sector is read and the number of entries increased by one and rewritten. Control then returns to the user. If an error is encountered in rewriting the account file, message 40 is written and the program terminates in error.

APPENDIX A

MESSAGES FROM THE PROGRAMS

A) PLDIR

(1) TOO MANY PARAMETERS ON CALL LINE

Error produced in DIRUPD by incorrect INFOR table (too long): this means the call to PLDIR was incorrect.

(2) TOO MANY CHARACTERS IN CHANNEL NO FIELD

Error produced in DIRUPD by more than 3 characters in the channel number specification field.

(3) ERROR ON PARAMETER CARD followed by a copy of the pen description card in error.

Error produced in DIRUPD when there are either more than six characters in any specification field or the paper width specification is not 15 or 36.

(4) OLD STYLE CARD PLEASE RESUBMIT IN NEW FORMAT
PWIDTH, NCOPIES, PEN1SIZE, PEN1COLOUR, ----

Error produced in DIRUPD when an incorrect number of fields is supplied on the pen description card.

(5) TOO MANY COPIES REQUESTED ONLY 1 COPY PLOTTED

Error produced in DIRUPD when more than nine copies are requested by a user (non-fatal error).

(6) TOO MANY CHARACTERS TRUNCATED TO:
the legal 48 characters are printed

Error produced in DIRUPD when the user's comment card is more than 48 characters in length (non-fatal error).

(7) TOO MANY FILES IN DIRECTORY

Error produced in DIRUPD when an attempt is made to have more than 999 files in the directory to be plotted. Inform the operators and try later.

(8) THIS IS PLOT NO NNN IN THE DIRECTORY

Message produced by each call on PLDIR to inform the user and the operators of this run's plot sequence number in the directory.

(9) FACILITY STATUS XXX XXX XXX XXX

Error produced in status by a non-zero status word returned from a CSFØ request. The printed status is in octal.

(10) SPOOLING PACKAGE LEVEL 5.0 MM/DD/YY HH:MM:SS

Introductory message produced in TDATE at the beginning of execution.

B) PLCOPY

(11) IS THIS PAPER FOR NNNNNNXXXXXX PLOTS?

This is the initial message produced by subroutine NOSE to identify the type of plots on any magnetic plot tape. NNNNNN is either 15 INS or 36 INS (the paper width), XXXXXX is either BIRO or INK (the tape type).

(12) THATS ALL FOLKS!!

This is the final message on each plot tape. It is written by subroutine TAIL to identify the end of a plot tape.

(13) NO PLOTS AT ALL TO DO

This message is produced on the operator's console when PLCOPY finds an empty directory.

(14) NO NNNNNNXXXXXX PLOTS

This message is produced on the operator's console when PLCOPY finds an empty tape type table. NNNNNN and XXXXXX have meanings as in message (11).

(15) THIS TAPE IS FOR NNNNNNXXXXXX PLOTS

This message is produced on the operator's console when PLCOPY has assigned a tape, to inform the operators what type of plot the tape will contain. NNNNNN and XXXXXX have meanings as in message (11).

(16) GRAPH PLOTTER SPOOLING SYSTEM NNNNNNXXXXXX PLOTS GRAPHTAPE NO :- YYYYYY

This message is produced on the operator's lineprinter output as a page heading for every tape type and for each 40 files within each tape type.

(17) END OF TAPE IN TRAILER MESSAGE

This message is produced on the operator's console when an end of tape status is encountered in writing the final message to the tape.

(18) FILE YYY YYY YYY YYY WILL NOT FIT ON TAPE

This message is produced on the operator's console when a user's plot file is too large to fit on a tape. This error will cause PLCOPY to terminate in error (YYYYYYYYYYYY is the filename which must be deleted from the directory before re-running PLCOPY).

(19) NEW TAPE NEEDED FOR NNNNNNXXXXXX PLOTS

This message is produced on the operator's console when an end of tape status is encountered in either HEADER or COPY and it is possible to plot the file.

(20) 01 ZZ PLOT TO DO

This message is written to the operator's lineprinter output by subroutine COPY when a CALCOMP control image is expected but not found.

ZZ = NO if this is the first record to be transferred.

ZZ = Sp Sp if some records have been transferred.

(21) 02 ZZ PLOT TO DO

This message is written to the operator's lineprinter output by subroutine COPY when the word count on a data record is too great. ZZ takes the values as stated in message (20).

(22) 04 PLOT TO DO

This message is written to the operator's lineprinter output by subroutine COPY when the user's plotfile length exceeds the maximum configured file length (FSIZE).

(23) 03 CD WW ZZ PLOT

This message is written to the operator's lineprinter output by subroutine COPY when the user's plotfile was written in error. WW is set at the octal value of the I/O error status code. ZZ is set as in message (20).

(24) PLEASE TRY AGAIN WHEN XXXXXX FINS

This message is written to the operator's console by subroutine ORDER if PLCOPY requires a user's plotfile which is currently in use. XXXXXX is the run-id of the job which is using the file.

(25) FILE YYY YYY YYY YYY WITH RUNID RRRRRR IS NOT IN THE MFD

This message is written to the operator's lineprinter output by subroutine ORDER if a directory sector entry exists although the user's plotfile does not exist. YYYYYYYYYYYY is the filename which does not exist and RRRRRR is the run-id which created the directory entry.

(26) FILE YYYYYYYYYYYY WITH RUNID RRRRRR HAS NO SPACE ASSIGNED

This message is produced on the operator's lineprinter output by subroutine ORDER if the user's plotfile exists but is completely empty. YYYYYYYYYYYY and RRRRRR take the meanings as described in message (25).

(27) PLOT SPOOLING TAPE CREATION RUN

This message is displayed on the operator's console by PLCOPY when the run starts.

(28) I/O ERROR NN IN WRITING ACCOUNT FILE

This message is displayed by WRCHG when an error is encountered while trying to read or write the account section of the system plotfile directory. It is a fatal error.

NN is set as in message (23) to the octal value of the I/O error code.

C) ACCOUNTING

(29) I/O ERROR NN IN ZEROING ACCOUNT FILE

This message is produced by Program ZERO when an error is encountered in trying to reset the account section of the plotfile directory. The error is fatal.

NN is set as in message (23).

(30) I/O ERROR NN IN READING ACCOUNT FILE

This message is produced by subroutine PLCHG when an error occurs while trying to read the account section of the plotfile directory. The error is fatal.

NN is set as in message (23).

(31) WHICH ACCOUNT DO YOU WISH TO AMMEND?

This is the first message produced by program ACCHG. It requests the operator to provide an account cade (up to 12 digits).

(32) CHARGE AND TIME FOR ACCOUNT XXXXXXXXXXXX ARE YYYYY.YY ZZZZZ.ZZ

This message will be produced by ACCHG for the account XXXXXXXXXXXX and contains values YYYYY.YY and ZZZZZ.ZZ for the current charge and time respectively.

(33) ACCOUNT NOT FOUND. INSERTING NEW ACCOUNT? Y OR N

This message will be produced by ACCHG if the account specified by the operator is not present in the directory.

(34) WHAT A MESS START AGAIN!!

This message will be produced by ACCHG if the operator answers N to question 33.

(35) INSERT VALUE TO BE ADDED TO CHARGE

This message will be produced by ACCHG after either message 32 or an answer of Y to message 33.

(36) NEW VALUES OF CHARGE AND TIME ARE YYYYY.YY ZZZZZ.ZZ
IS THIS CORRECT? ANSWER Y OR N

This message will be produced by ACCHG after message 35 to indicate the updated values.

(37) ORIGINAL VALUES OF TIME AND CHARGE RESTORED

This message is produced by ACCHG after an answer of N to message 36.

(38) NEW VALUES OF CHARGE AND TIME FOR ACCOUNT XXXXXXXXXXXX ARE:
YYYYY.YY ZZZZZ.ZZ

This message will be produced by ACCHG after an answer of Y to message 36.

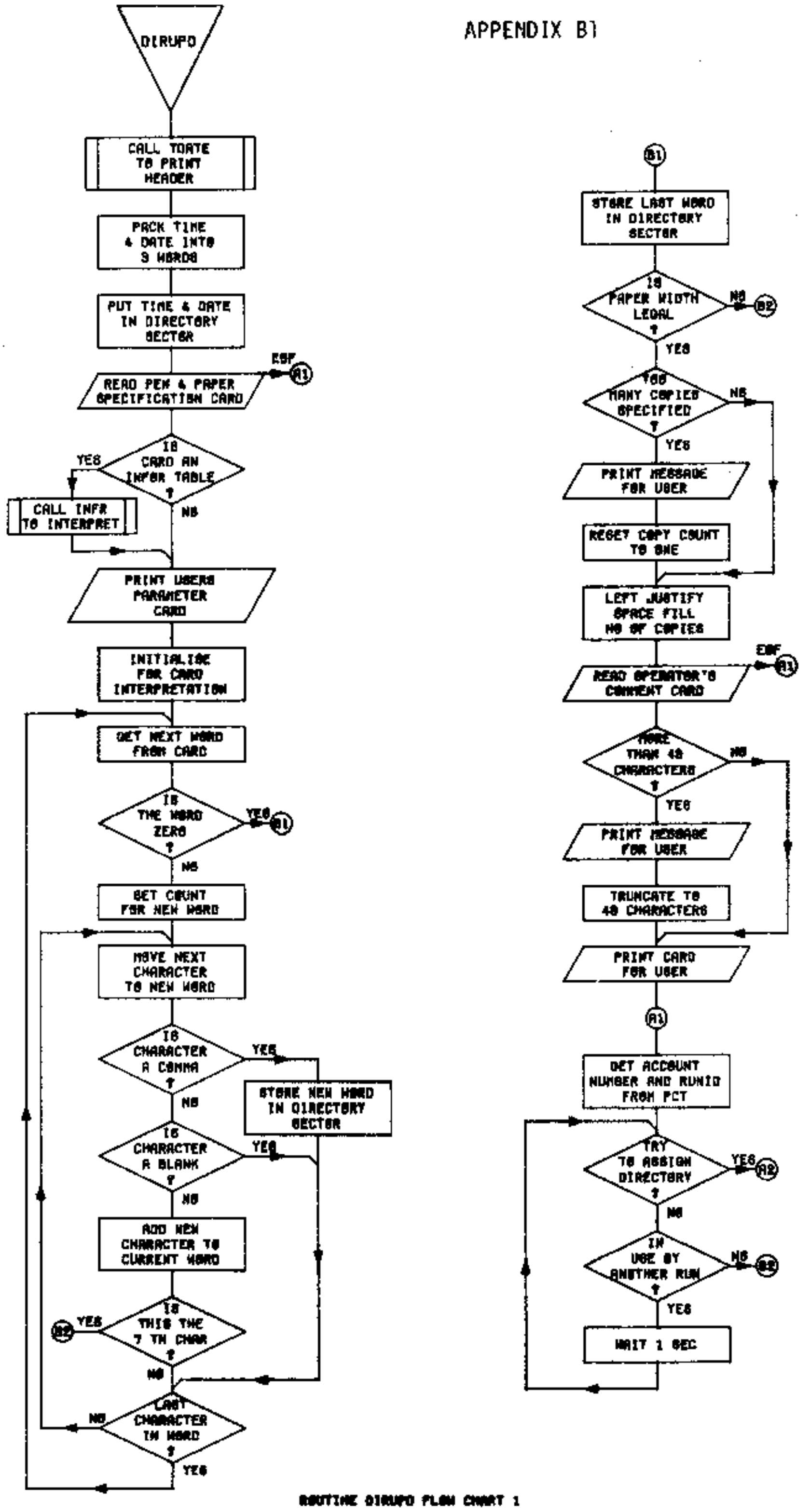
(39) ANY MORE ACCOUNTS TO UPDATE ANSWER Y OR N

This message is produced by ACCHG after message 36 to allow further updates in the same execution.

(40) I/O ERROR XX IN YYYYY.YY ACCOUNT FILE

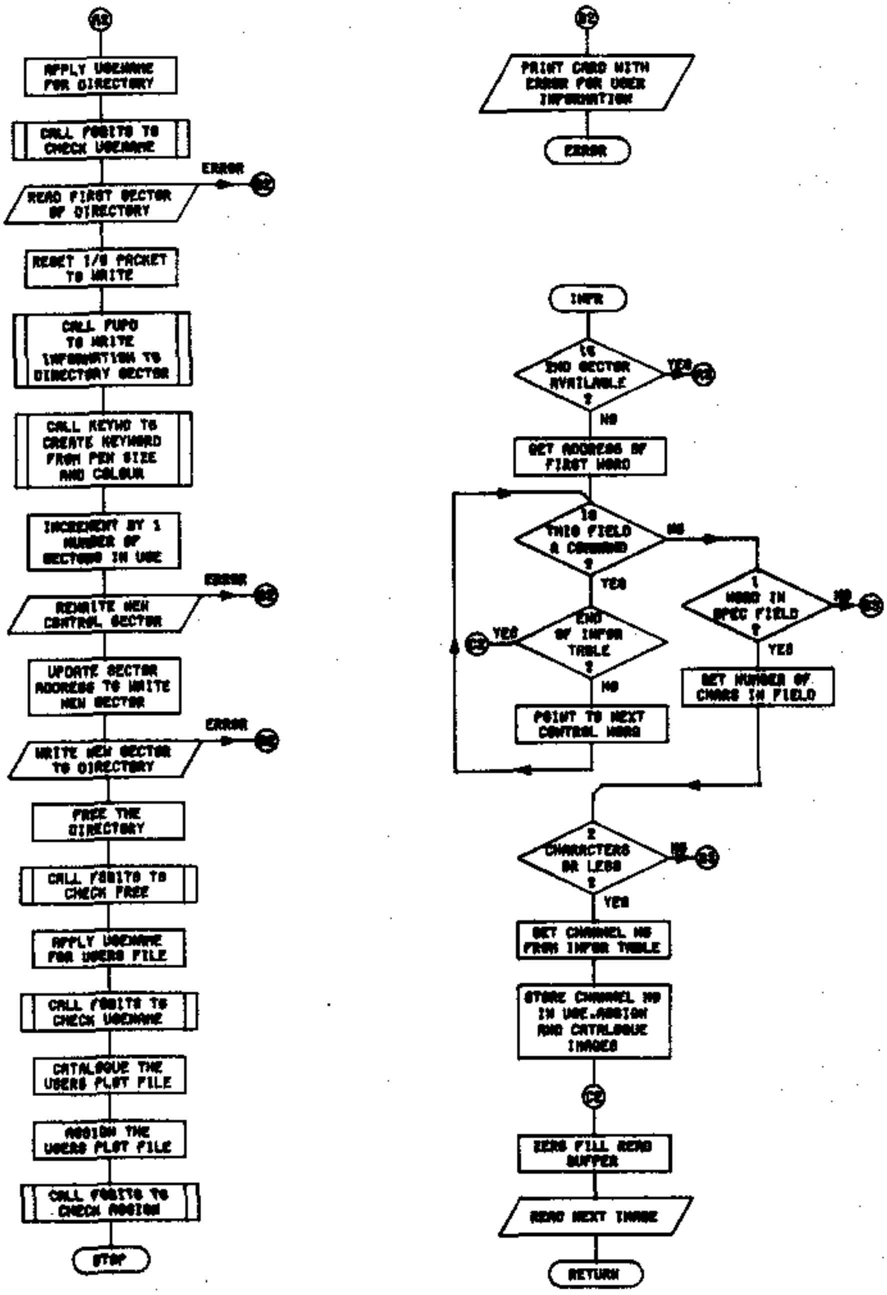
This message is produced by ACSUB when accessing the account file. XX is the I/O error code. YYYYY.YY is either READING or WRITING.

APPENDIX B1


DIRUPD
CALL TDATE TO PRINT HEADER
PACK TIME & DATE INTO 3 WORDS
PUT TIME & DATE IN DIRECTORY SECTOR
READ PEN & PAPER SPECIFICATION CARD
EOF
A1
IS CARD AN INFR TABLE ?
YES
CALL INFR TO INTERPRET
NO
PRINT USERS PARAMETER CARD
INITIALISE FOR CARD INTERPRETATION
GET NEXT WORD FROM CARD
IS THE WORD ZERO ?
YES
B1
NO
GET COUNT FOR NEW WORD
MOVE NEXT CHARACTER TO NEW WORD
IS CHARACTER A COMMA ?
YES
STORE NEW WORD IN DIRECTORY SECTOR
NO
IS CHARACTER A BLANK ?
YES
NO
ADD NEW CHARACTER TO CURRENT WORD
IS THIS THE 7 TH CHAR ?
YES
B2
NO
LAST CHARACTER IN WORD ?
NO
YES
B1
STORE LAST WORD IN DIRECTORY SECTOR
IS PAPER WIDTH LEGAL ?
NO
B2
YES
TOO MANY COPIES SPECIFIED ?
NO
YES
PRINT MESSAGE FOR USER
RESET COPY COUNT TO ONE
LEFT JUSTIFY SPACE FILL NO OF COPIES
READ OPERATOR'S COMMENT CARD
EOF
A1
MORE THAN 48 CHARACTERS ?
NO
YES
PRINT MESSAGE FOR USER
TRUNCATE TO 48 CHARACTERS
PRINT CARD FOR USER
A1
GET ACCOUNT NUMBER AND RUNID FROM PCT
TRY TO ASSIGN DIRECTORY ?
YES
A2
NO
IN USE BY ANOTHER RUN ?
NO
B2
YES
WAIT 1 SEC

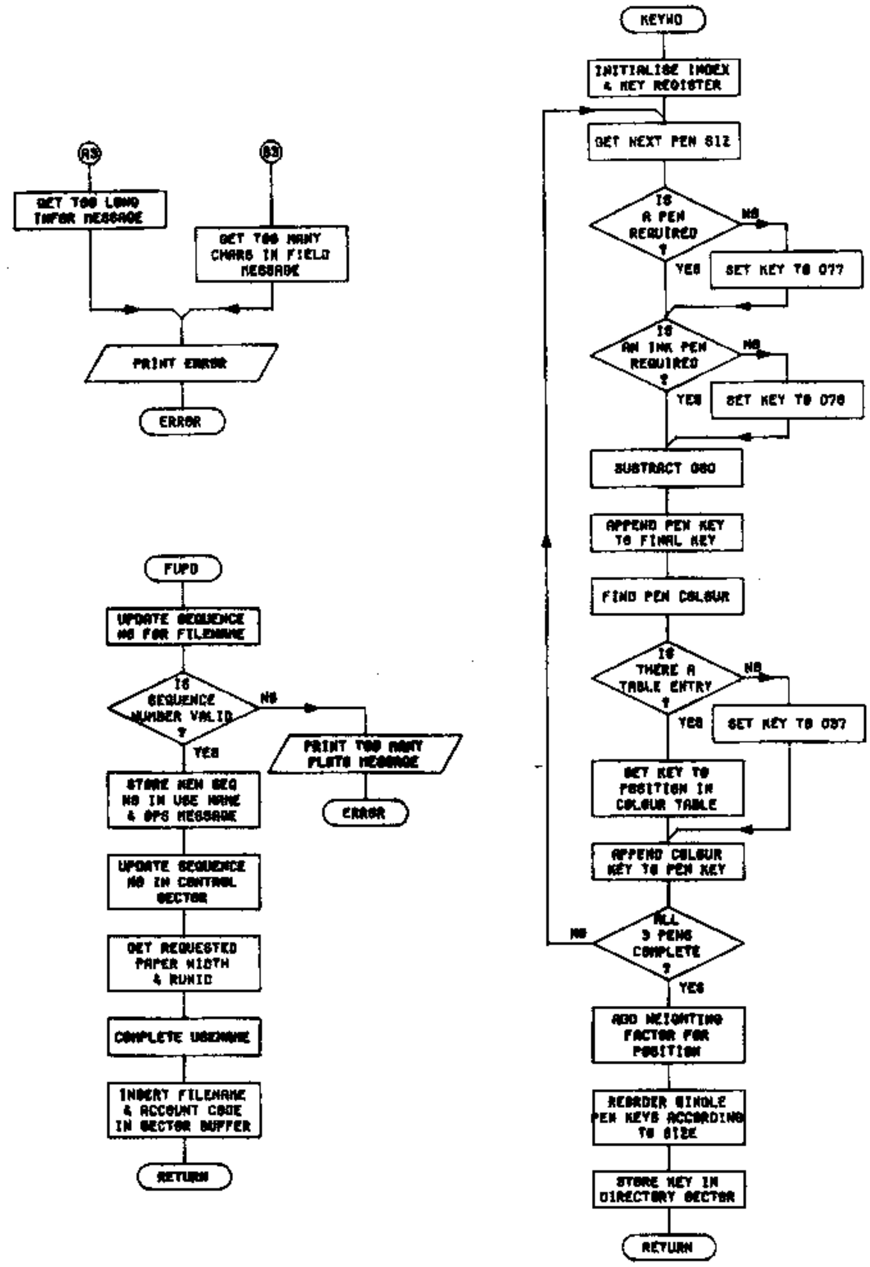
ROUTINE DIRUPD FLOW CHART 1

APPENDIX B2


A2
APPLY USENAME FOR DIRECTORY
CALL FOBITS TO CHECK USENAME
READ FIRST SECTOR OF DIRECTORY
ERROR
B2
RESET I/O PACKET TO WRITE
CALL FUPD TO WRITE INFORMATION TO DIRECTORY SECTOR
CALL KEYWD TO CREATE KEYWORD FROM PEN SIZE AND COLOUR
INCREMENT BY 1 NUMBER OF SECTORS IN USE
REWRITE NEW CONTROL SECTOR
ERROR
B2
UPDATE SECTOR ADDRESS TO WRITE NEW SECTOR
WRITE NEW SECTOR TO DIRECTORY
ERROR
B2
FREE THE DIRECTORY
CALL FOBITS TO CHECK FREE
APPLY USENAME FOR USERS FILE
CALL FOBITS TO CHECK USENAME
CATALOGUE THE USERS PLOT FILE
ASSIGN THE USERS PLOT FILE
CALL FOBITS TO CHECK ASSIGN
STOP
B2
PRINT CARD WITH ERROR FOR USER INFORMATION
ERROR
INFR
IS END SECTOR AVAILABLE ?
YES
NO
GET ADDRESS OF FIRST WORD
IS THIS FIELD A COMMAND ?
NO
YES
1 WORD IN SPEC FIELD ?
NO
YES
GET NUMBER OF CHARS IN FIELD
END OF INFOR TABLE ?
YES
C2
NO
POINT TO NEXT CONTROL WORD
2 CHARACTERS OR LESS ?
NO
YES
GET CHANNEL NO FROM INFOR TABLE
STORE CHANNEL NO IN USE.ASSIGN AND CATALOGUE IMAGES
C2
ZERO FILL READ BUFFER
READ NEXT IMAGE
RETURN

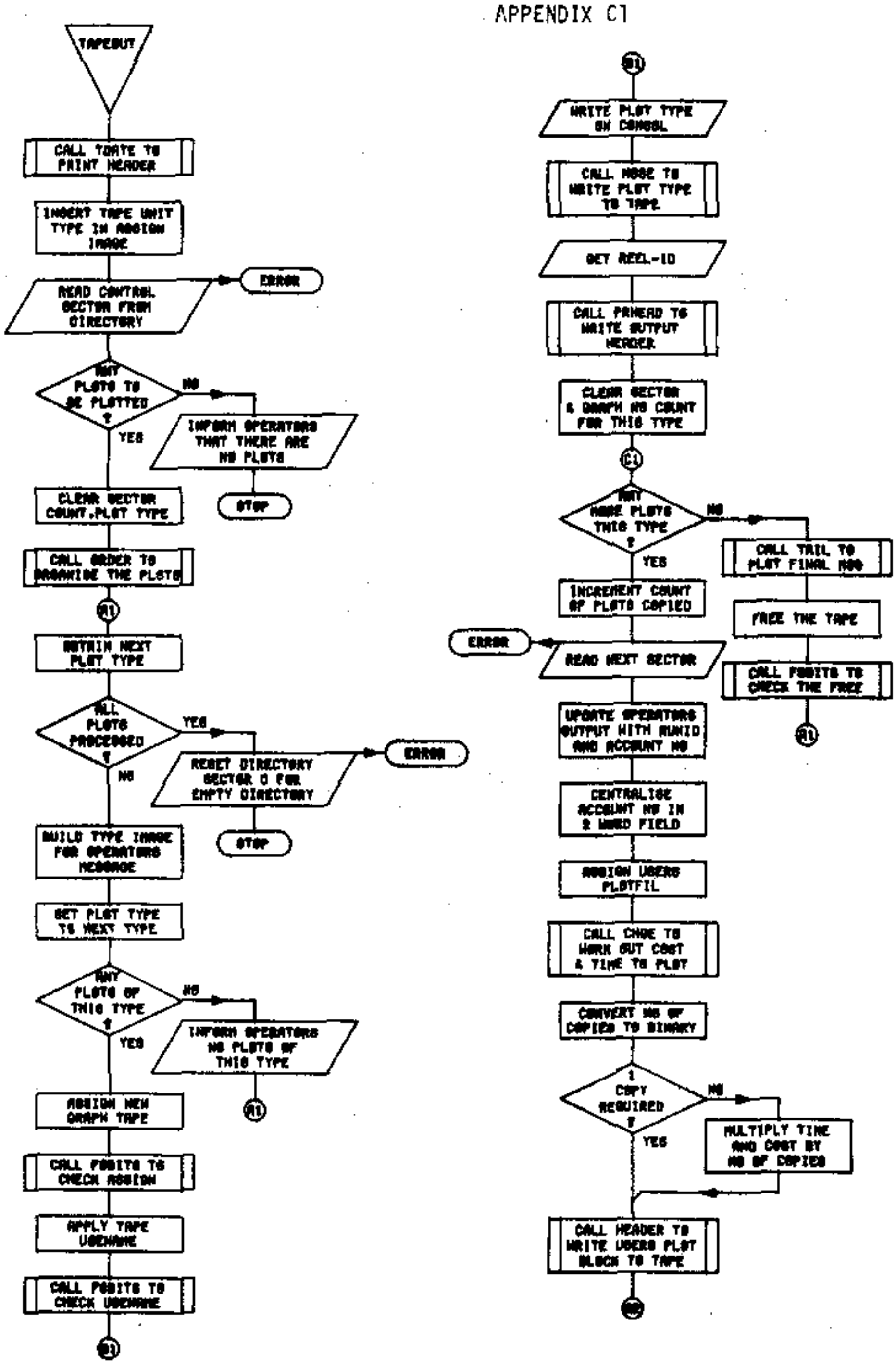
ROUTINE DISUPD FLOW CHART 2

APPENDIX B3


A3
B3
GET TOO LONG INFOR MESSAGE
GET TOO MANY CHARS IN FIELD MESSAGE
PRINT ERROR
ERROR
FUPD
UPDATE SEQUENCE NO FOR FILENAME
IS SEQUENCE NUMBER VALID ?
NO
YES
PRINT TOO MANY PLOTS MESSAGE
ERROR
STORE NEW SEQ NO IN USE NAME & OPS MESSAGE
UPDATE SEQUENCE NO IN CONTROL SECTOR
GET REQUESTED PAPER WIDTH & RUNID
COMPLETE USENAME
INSERT FILENAME & ACCOUNT CODE IN SECTOR BUFFER
RETURN
KEYNO
INITIALISE INDEX & KEY REGISTER
GET NEXT PEN SIZ
IS A PEN REQUIRED ?
NO
YES
SET KEY TO 077
IS AN INK PEN REQUIRED ?
NO
YES
SET KEY TO 070
SUBTRACT 060
APPEND PEN KEY TO FINAL KEY
FIND PEN COLOUR
IS THERE A TABLE ENTRY ?
NO
YES
SET KEY TO 037
SET KEY TO POSITION IN COLOUR TABLE
APPEND COLOUR KEY TO PEN KEY
ALL 3 PENS COMPLETE ?
NO
YES
ADD WEIGHTING FACTOR FOR POSITION
REORDER SINGLE PEN KEYS ACCORDING TO SIZE
STORE KEY IN DIRECTORY SECTOR
RETURN

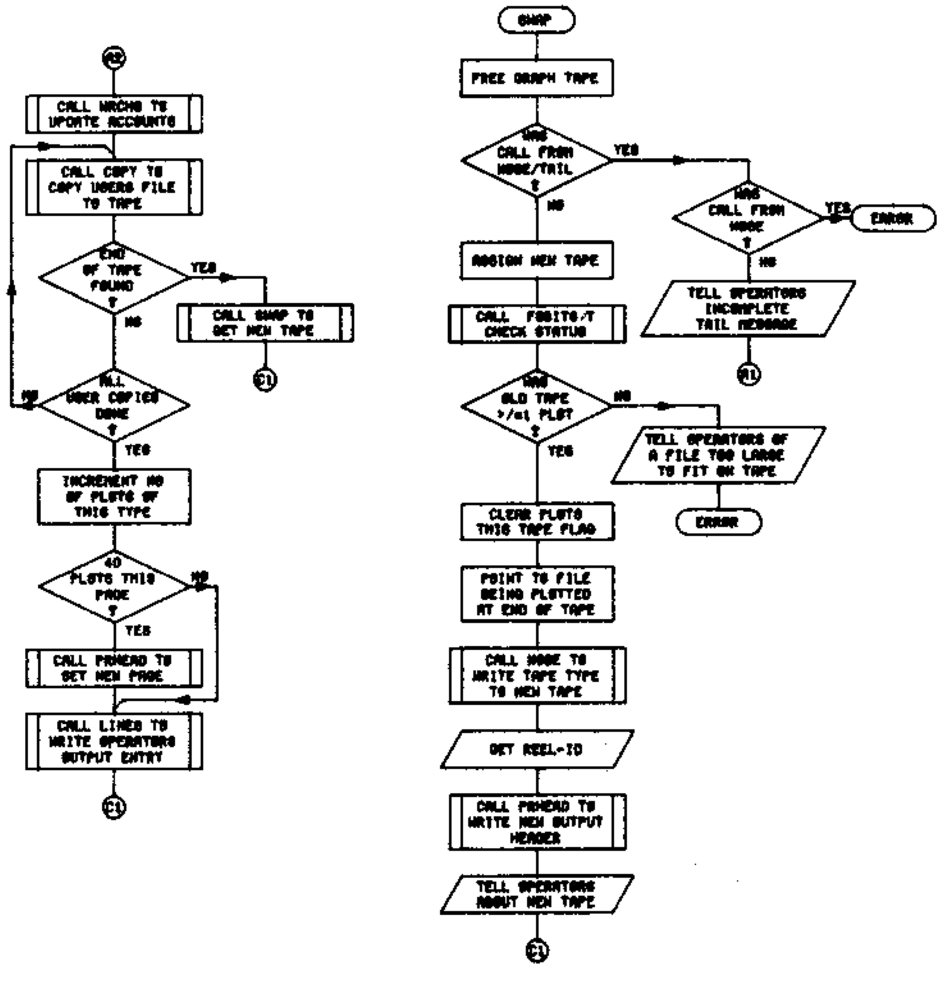
ROUTINE DIRUPD FLOW CHART 3

APPENDIX C1


TAPEOUT
CALL TDATE TO PRINT HEADER
INSERT TAPE UNIT TYPE IN ASSIGN IMAGE
READ CONTROL SECTOR FROM DIRECTORY
ERROR
ANY PLOTS TO BE PLOTTED ?
NO
YES
INFORM OPERATORS THAT THERE ARE NO PLOTS
STOP
CLEAR SECTOR COUNT,PLOT TYPE
CALL ORDER TO ORGANISE THE PLOTS
A1
OBTAIN NEXT PLOT TYPE
ALL PLOTS PROCESSED ?
YES
NO
RESET DIRECTORY SECTOR 0 FOR EMPTY DIRECTORY
ERROR
STOP
BUILD TYPE IMAGE FOR OPERATORS MESSAGE
SET PLOT TYPE TO NEXT TYPE
ANY PLOTS OF THIS TYPE ?
NO
YES
INFORM OPERATORS NO PLOTS OF THIS TYPE
A1
ASSIGN NEW GRAPH TAPE
CALL FOBITS TO CHECK ASSIGN
APPLY TAPE USENAME
CALL FOBITS TO CHECK USENAME
B1
B1
WRITE PLOT TYPE ON CONSOL
CALL NOSE TO WRITE PLOT TYPE TO TAPE
GET REEL-ID
CALL PRHEAD TO WRITE OUTPUT HEADER
CLEAR SECTOR & GRAPH NO COUNT FOR THIS TYPE
C1
ANY MORE PLOTS THIS TYPE ?
NO
YES
CALL TRIL TO PLOT FINAL NOS
FREE THE TAPE
CALL FOBITS TO CHECK THE FREE
A1
INCREMENT COUNT OF PLOTS COPIED
ERROR
READ NEXT SECTOR
UPDATE OPERATORS OUTPUT WITH RUNID AND ACCOUNT NO
CENTRALISE ACCOUNT NO IN 2 WORD FIELD
ASSIGN USERS PLOTFIL
CALL CHOE TO WORK OUT COST & TIME TO PLOT
CONVERT NO OF COPIES TO BINARY
1 COPY REQUIRED ?
NO
YES
MULTIPLY TIME AND COST BY NO OF COPIES
CALL HEADER TO WRITE USERS PLOT BLOCK TO TAPE

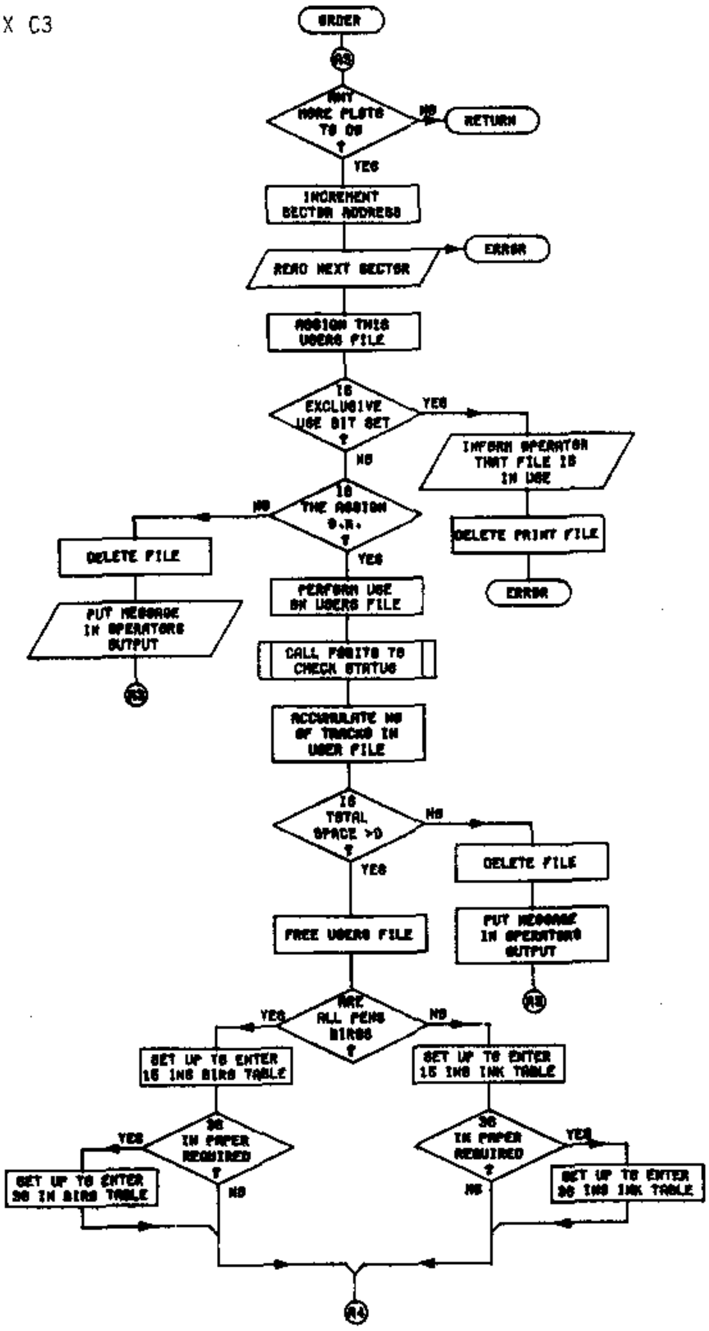
ROUTINE TAPEOUT FLOW CHART 1

APPENDIX C2


A2
CALL WRCHG TO UPDATE ACCOUNTS
CALL COPY TO COPY USERS FILE TO TAPE
END OF TAPE FOUND ?
YES
CALL SWAP TO GET NEW TAPE
C1
NO
ALL USER COPIES DONE ?
NO
YES
INCREMENT NO OF PLOTS OF THIS TYPE
40 PLOTS THIS PAGE ?
NO
YES
CALL PGHEAD TO GET NEW PAGE
CALL LINES TO WRITE OPERATORS OUTPUT ENTRY
C1
SWAP
FREE GRAPH TAPE
WAS CALL FROM HOSE/TAIL ?
YES
WAS CALL FROM HOSE ?
YES
ERROR
NO
TELL OPERATORS INCOMPLETE TAIL MESSAGE
A1
NO
ASSIGN NEW TAPE
CALL FOSITS/T CHECK STATUS
WAS OLD TAPE >/=1 PLOT ?
NO
TELL OPERATORS OF A FILE TOO LARGE TO FIT ON TAPE
ERROR
YES
CLEAR PLOTS THIS TAPE FLAG
POINT TO FILE BEING PLOTTED AT END OF TAPE
CALL HOSE TO WRITE TAPE TYPE TO NEW TAPE
GET REEL-ID
CALL PGHEAD TO WRITE NEW OUTPUT HEADER
TELL OPERATORS ABOUT NEW TAPE
C1

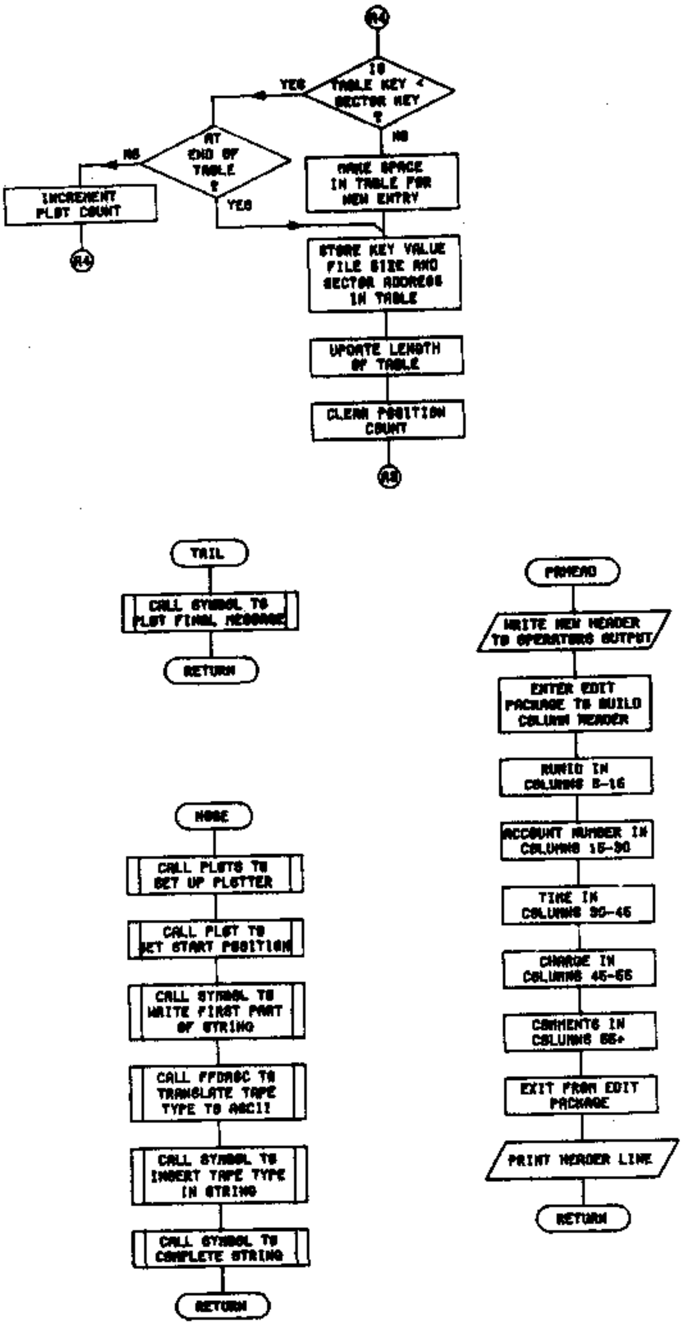
ROUTINE TAPEOUT FLOW CHART 2

APPENDIX C3


ORDER
A3
ANY MORE PLOTS TO DO ?
NO
RETURN
YES
INCREMENT SECTOR ADDRESS
READ NEXT SECTOR
ERROR
ASSIGN THIS USERS FILE
IS EXCLUSIVE USE BIT SET ?
YES
INFORM OPERATOR THAT FILE IS IN USE
DELETE PRINT FILE
ERROR
NO
IS THE ASSIGN O.K. ?
NO
DELETE FILE
PUT MESSAGE IN OPERATORS OUTPUT
A3
YES
PERFORM USE ON USERS FILE
CALL FOBITS TO CHECK STATUS
ACCUMULATE NO OF TRACKS IN USER FILE
IS TOTAL SPACE >0 ?
NO
DELETE FILE
PUT MESSAGE IN OPERATORS OUTPUT
A3
YES
FREE USERS FILE
ARE ALL PENS BIROS ?
YES
SET UP TO ENTER 15 INS BIRO TABLE
36 IN PAPER REQUIRED ?
YES
SET UP TO ENTER 36 IN BIRO TABLE
NO
NO
SET UP TO ENTER 15 INS INK TABLE
36 IN PAPER REQUIRED ?
YES
SET UP TO ENTER 36 INS INK TABLE
NO
A4

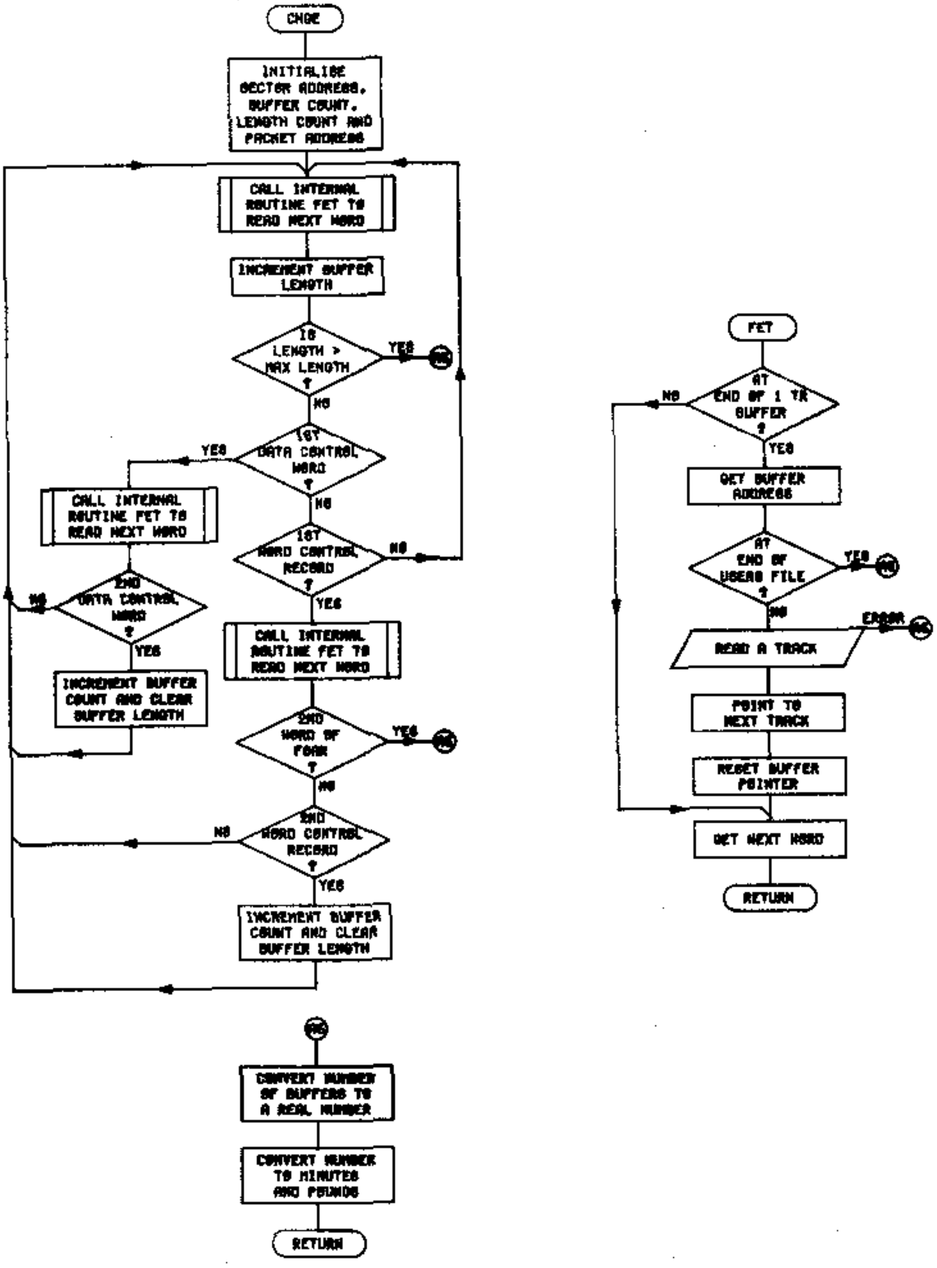
ROUTINE TAPEOUT FLOW CHART 3

APPENDIX C4


A4
IS TABLE KEY < SECTOR KEY ?
YES
NO
AT END OF TABLE ?
NO
YES
INCREMENT PLOT COUNT
A4
MAKE SPACE IN TABLE FOR NEW ENTRY
STORE KEY VALUE FILE SIZE AND SECTOR ADDRESS IN TABLE
UPDATE LENGTH OF TABLE
CLEAR POSITION COUNT
A3
TAIL
CALL SYMBOL TO PLOT FINAL MESSAGE
RETURN
PRHEAD
WRITE NEW HEADER TO OPERATORS OUTPUT
ENTER EDIT PACKAGE TO BUILD COLUMN HEADER
RUNID IN COLUMNS 8-16
ACCOUNT NUMBER IN COLUMNS 15-30
TIME IN COLUMNS 30-45
CHARGE IN COLUMNS 45-65
COMMENTS IN COLUMNS 65+
EXIT FROM EDIT PACKAGE
PRINT HEADER LINE
RETURN
NOSE
CALL PLOTS TO SET UP PLOTTER
CALL PLOT TO SET START POSITION
CALL SYMBOL TO WRITE FIRST PART OF STRING
CALL FFDASC TO TRANSLATE TAPE TYPE TO ASCII
CALL SYMBOL TO INSERT TAPE TYPE IN STRING
CALL SYMBOL TO COMPLETE STRING
RETURN

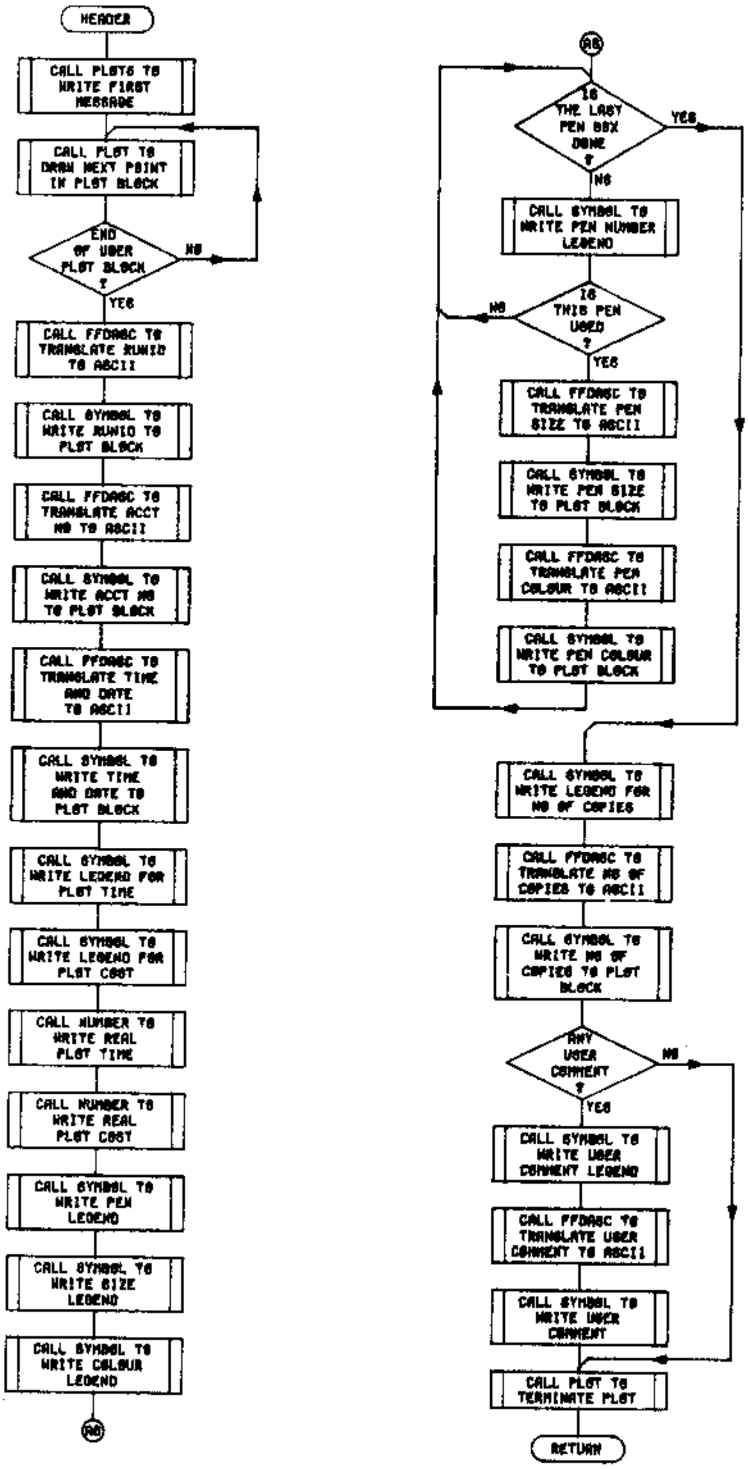
ROUTINE TAPEOUT FLOW CHART 4

APPENDIX C5


CHGE
INITIALISE SECTOR ADDRESS, BUFFER COUNT, LENGTH COUNT AND PACKET ADDRESS
CALL INTERNAL ROUTINE FET TO READ NEXT WORD
INCREMENT BUFFER LENGTH
IS LENGTH > MAX LENGTH ?
YES
NO
1ST DATA CONTROL WORD ?
YES
NO
CALL INTERNAL ROUTINE FET TO READ NEXT WORD
2ND DATA CONTROL WORD ?
NO
YES
INCREMENT BUFFER COUNT AND CLEAR BUFFER LENGTH
1ST WORD CONTROL RECORD ?
NO
YES
CALL INTERNAL ROUTINE FET TO READ NEXT WORD
2ND WORD OF FORM ?
YES
NO
2ND WORD CONTROL RECORD ?
NO
YES
INCREMENT BUFFER COUNT AND CLEAR BUFFER LENGTH
CONVERT NUMBER OF BUFFERS TO A REAL NUMBER
CONVERT NUMBER TO MINUTES AND POUNDS
RETURN
FET
AT END OF 1 TR BUFFER ?
NO
YES
GET BUFFER ADDRESS
AT END OF USERS FILE ?
YES
NO
ERROR
READ A TRACK
POINT TO NEXT TRACK
RESET BUFFER POINTER
GET NEXT WORD
RETURN

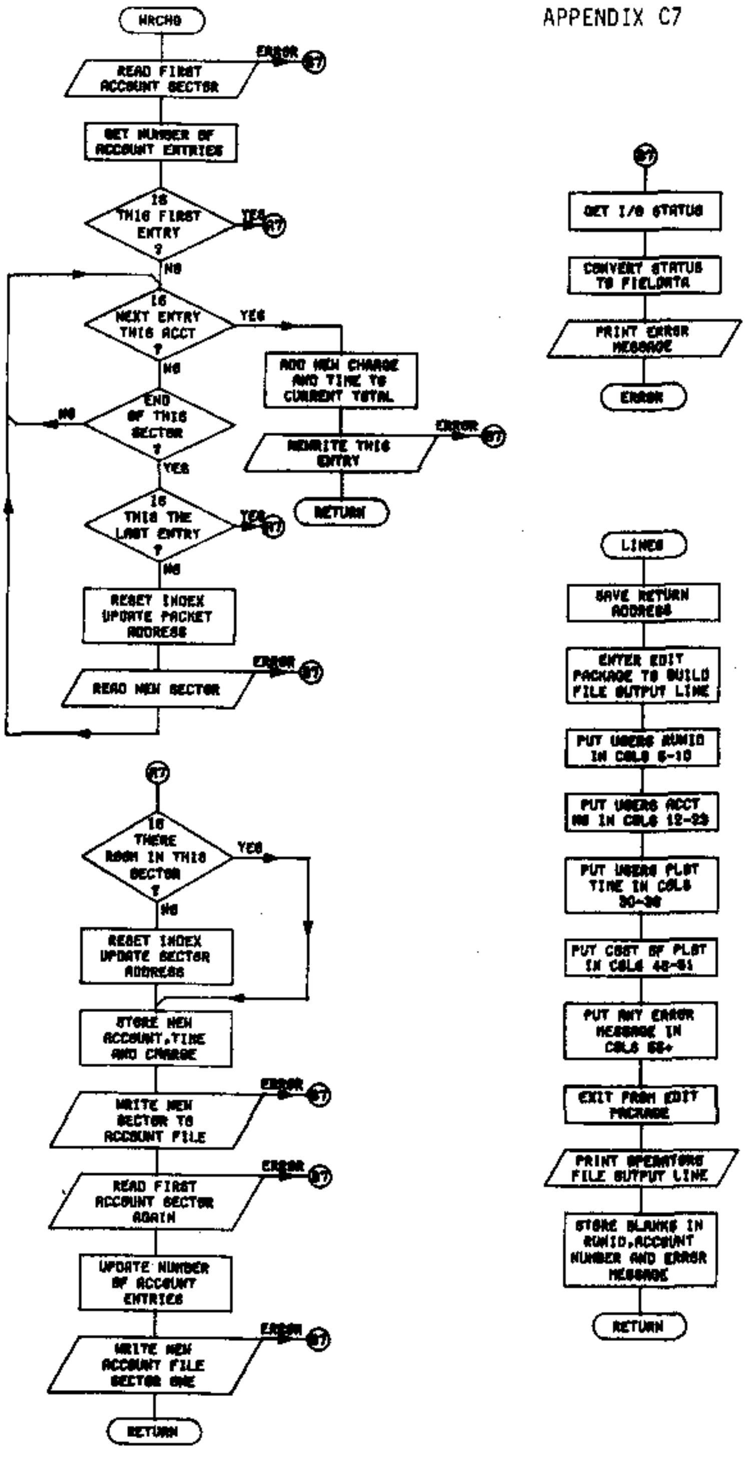
ROUTINE TAPEOUT FLOW CHART 5

APPENDIX C6


HEADER
CALL PLOTS TO WRITE FIRST MESSAGE
CALL PLOT TO DRAW NEXT POINT IN PLOT BLOCK
END OF USER PLOT BLOCK ?
NO
YES
CALL FFDASC TO TRANSLATE RUNID TO ASCII
CALL SYMBOL TO WRITE RUNID TO PLOT BLOCK
CALL FFDASC TO TRANSLATE ACCT NO TO ASCII
CALL SYMBOL TO WRITE ACCT NO TO PLOT BLOCK
CALL FFDASC TO TRANSLATE TIME AND DATE TO ASCII
CALL SYMBOL TO WRITE TIME AND DATE TO PLOT BLOCK
CALL SYMBOL TO WRITE LEGEND FOR PLOT TIME
CALL SYMBOL TO WRITE LEGEND FOR PLOT COST
CALL NUMBER TO WRITE REAL PLOT TIME
CALL NUMBER TO WRITE REAL PLOT COST
CALL SYMBOL TO WRITE PEN LEGEND
CALL SYMBOL TO WRITE SIZE LEGEND
CALL SYMBOL TO WRITE COLOUR LEGEND
A6
A6
IS THE LAST PEN BOX DONE ?
YES
NO
CALL SYMBOL TO WRITE PEN NUMBER LEGEND
IS THIS PEN USED ?
NO
YES
CALL FFDASC TO TRANSLATE PEN SIZE TO ASCII
CALL SYMBOL TO WRITE PEN SIZE TO PLOT BLOCK
CALL FFDASC TO TRANSLATE PEN COLOUR TO ASCII
CALL SYMBOL TO WRITE PEN COLOUR TO PLOT BLOCK
CALL SYMBOL TO WRITE LEGEND FOR NO OF COPIES
CALL FFDASC TO TRANSLATE NO OF COPIES TO ASCII
CALL SYMBOL TO WRITE NO OF COPIES TO PLOT BLOCK
ANY USER COMMENT ?
NO
YES
CALL SYMBOL TO WRITE USER COMMENT LEGEND
CALL FFDASC TO TRANSLATE USER COMMENT TO ASCII
CALL SYMBOL TO WRITE USER COMMENT
CALL PLOT TO TERMINATE PLOT
RETURN

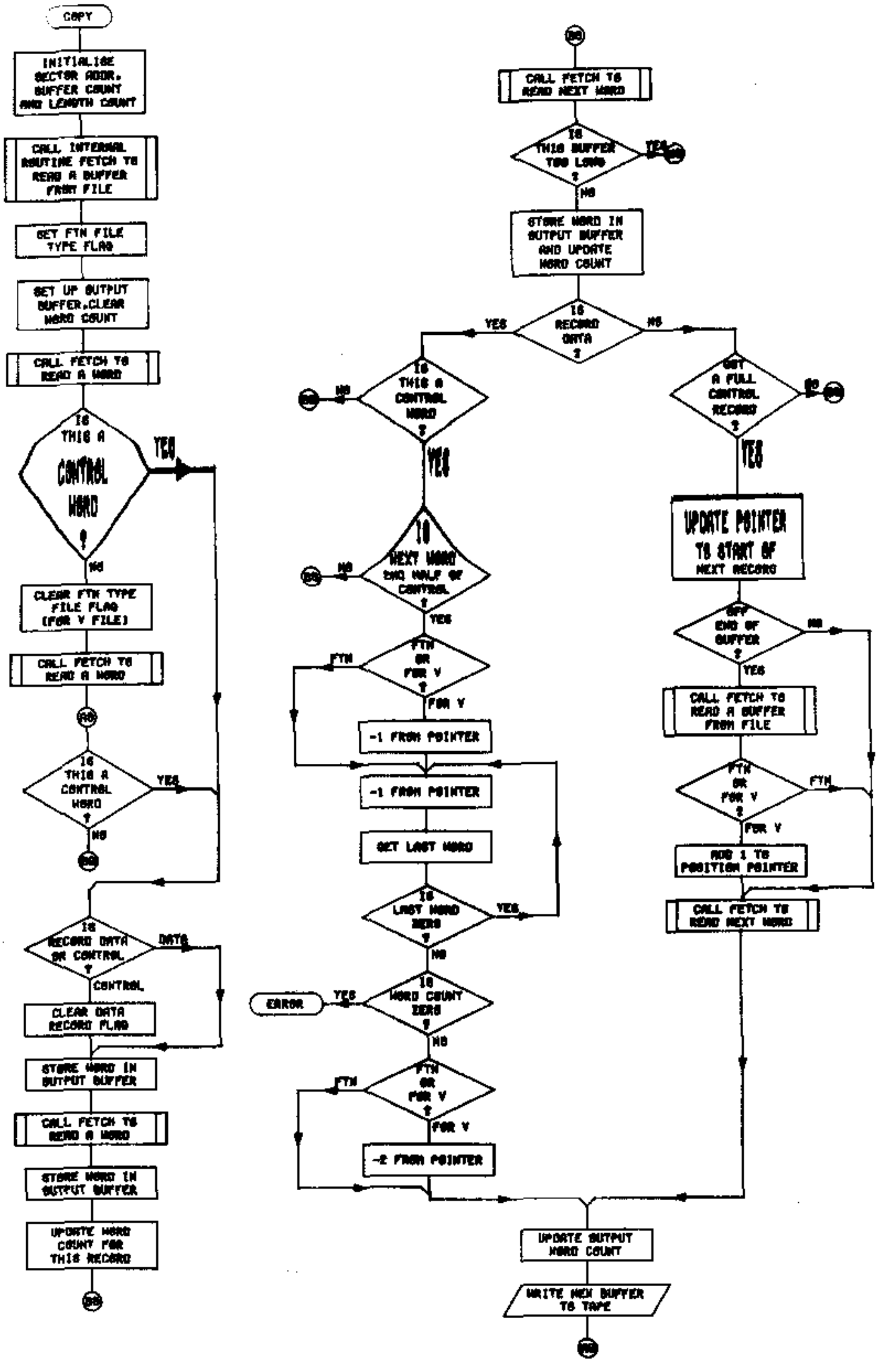
ROUTINE TAPEOUT FLOW CHART 8

APPENDIX C7


HRCHG
READ FIRST ACCOUNT SECTOR
ERROR
B7
GET NUMBER OF ACCOUNT ENTRIES
IS THIS FIRST ENTRY ?
YES
A7
NO
IS NEXT ENTRY THIS ACCT ?
YES
ADD NEW CHARGE AND TIME TO CURRENT TOTAL
REWRITE THIS ENTRY
ERROR
B7
RETURN
NO
END OF THIS SECTOR ?
NO
YES
IS THIS THE LAST ENTRY ?
YES
A7
NO
RESET INDEX UPDATE PACKET ADDRESS
READ NEW SECTOR
ERROR
B7
A7
IS THERE ROOM IN THIS SECTOR ?
YES
NO
RESET INDEX UPDATE SECTOR ADDRESS
STORE NEW ACCOUNT,TIME AND CHARGE
WRITE NEW SECTOR TO ACCOUNT FILE
ERROR
B7
READ FIRST ACCOUNT SECTOR AGAIN
ERROR
B7
UPDATE NUMBER OF ACCOUNT ENTRIES
WRITE NEW ACCOUNT FILE SECTOR ONE
ERROR
B7
RETURN
B7
GET I/O STATUS
CONVERT STATUS TO FIELDATA
PRINT ERROR MESSAGE
ERROR
LINES
SAVE RETURN ADDRESS
ENTER EDIT PACKAGE TO BUILD FILE OUTPUT LINE
PUT USERS RUNID IN COLS 6-10
PUT USERS ACCT NO IN COLS 12-23
PUT USERS PLOT TIME IN COLS 30-38
PUT COST OF PLOT IN COLS 46-51
PUT ANY ERROR MESSAGE IN COLS 58+
EXIT FROM EDIT PACKAGE
PRINT OPERATORS FILE OUTPUT LINE
STORE BLANKS IN RUNID,ACCOUNT NUMBER AND ERROR MESSAGE
RETURN

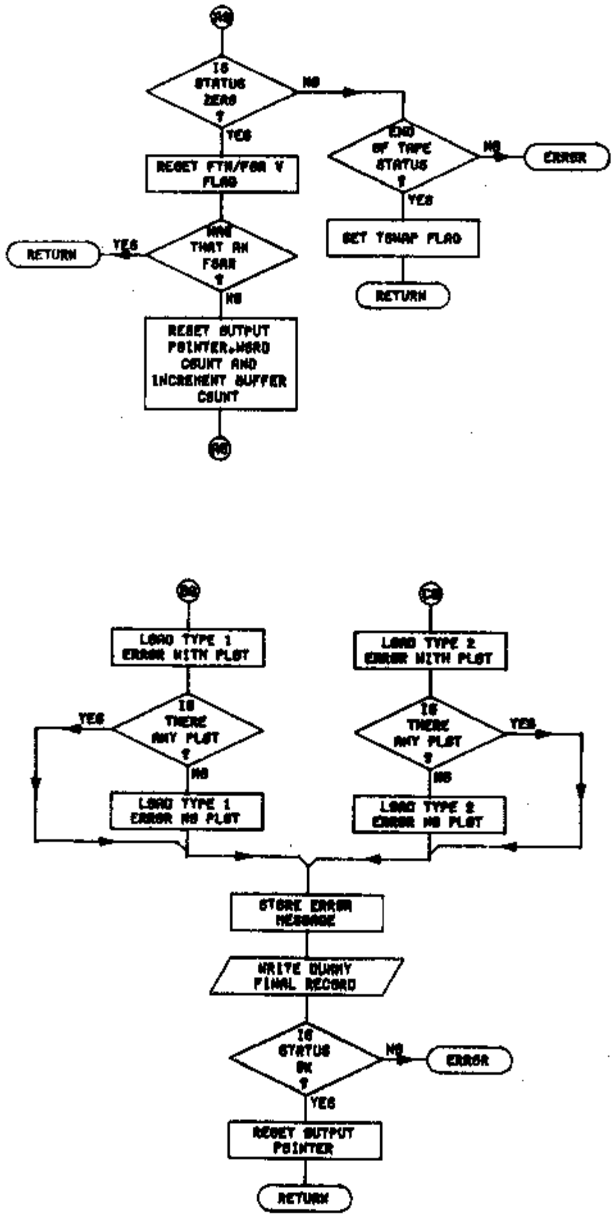
ROUTINE TAPEOUT FLOW CHART 7

APPENDIX C8


COPY
INITIALISE SECTOR ADDR. BUFFER COUNT AND LENGTH COUNT
CALL INTERNAL ROUTINE FETCH TO READ A BUFFER FROM FILE
SET FTN FILE TYPE FLAG
SET UP OUTPUT BUFFER.CLEAR WORD COUNT
CALL FETCH TO READ A WORD
IS THIS A CONTROL WORD ?
YES
NO
CLEAR FTN TYPE FILE FLAG (FOR V FILE)
CALL FETCH TO READ A WORD
IS THIS A CONTROL WORD ?
YES
NO
IS RECORD DATA OR CONTROL ?
DATA
CONTROL
CLEAR DATA RECORD FLAG
STORE WORD IN OUTPUT BUFFER
CALL FETCH TO READ A WORD
STORE WORD IN OUTPUT BUFFER
UPDATE WORD COUNT FOR THIS RECORD
CALL FETCH TO READ NEXT WORD
IS THIS BUFFER TOO LONG ?
YES
NO
STORE WORD IN OUTPUT BUFFER AND UPDATE WORD COUNT
IS RECORD DATA ?
YES
NO
IS THIS A CONTROL WORD ?
NO
YES
IS NEXT WORD 2ND HALF OF CONTROL ?
NO
YES
FTN OR FOR V ?
FTN
FOR V
-1 FROM POINTER
-1 FROM POINTER
GET LAST WORD
IS LAST WORD ZERO ?
YES
NO
IS WORD COUNT ZERO ?
YES
ERROR
NO
FTN OR FOR V ?
FTN
FOR V
-2 FROM POINTER
GOT A FULL CONTROL RECORD ?
NO
YES
UPDATE POINTER TO START OF NEXT RECORD
OFF END OF BUFFER ?
NO
YES
CALL FETCH TO READ A BUFFER FROM FILE
FTN OR FOR V ?
FTN
FOR V
ADD 1 TO POSITION POINTER
CALL FETCH TO READ NEXT WORD
UPDATE OUTPUT WORD COUNT
WRITE NEW BUFFER TO TAPE

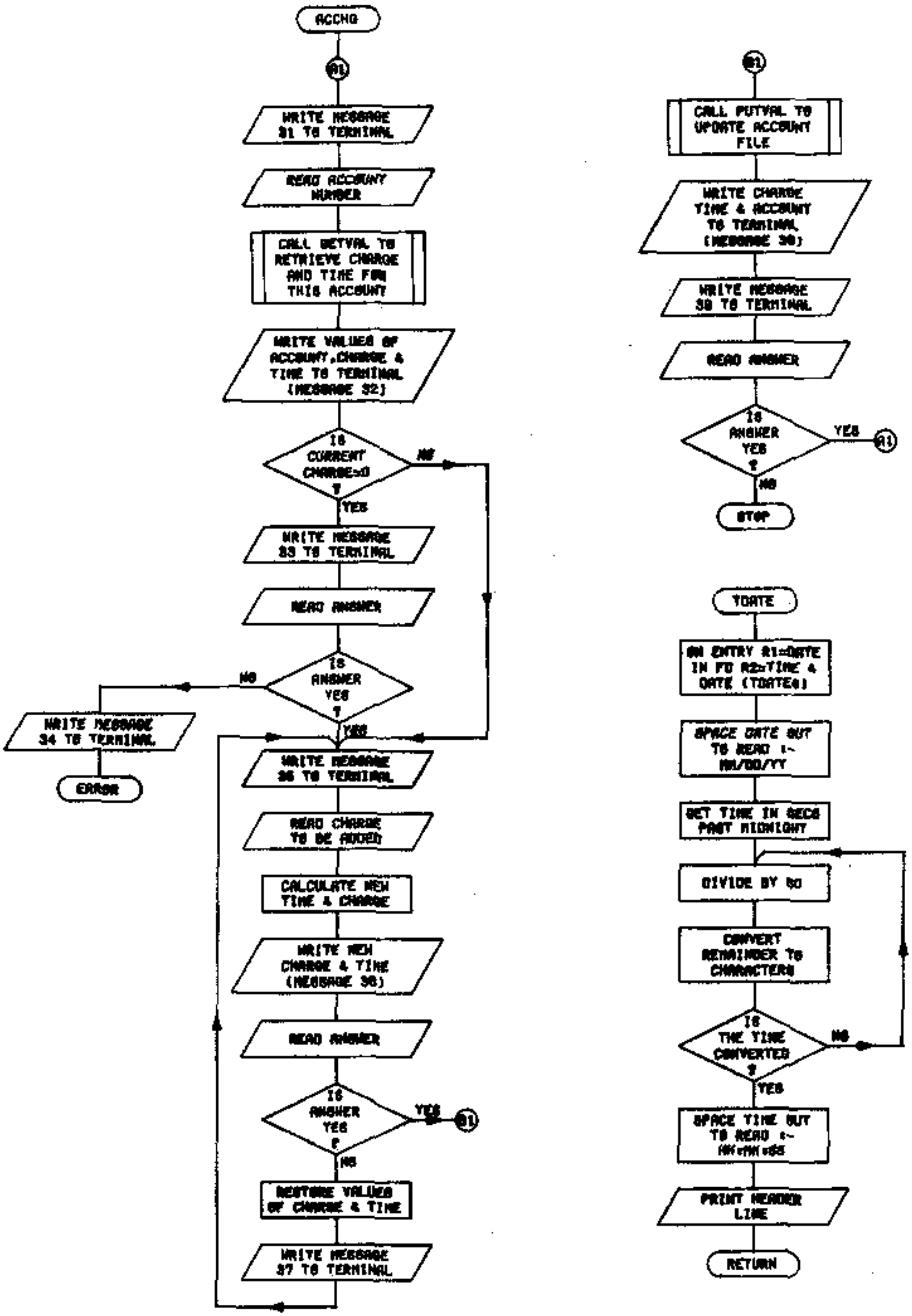
ROUTINE TAPEOUT FLOW CHART 8

APPENDIX C9



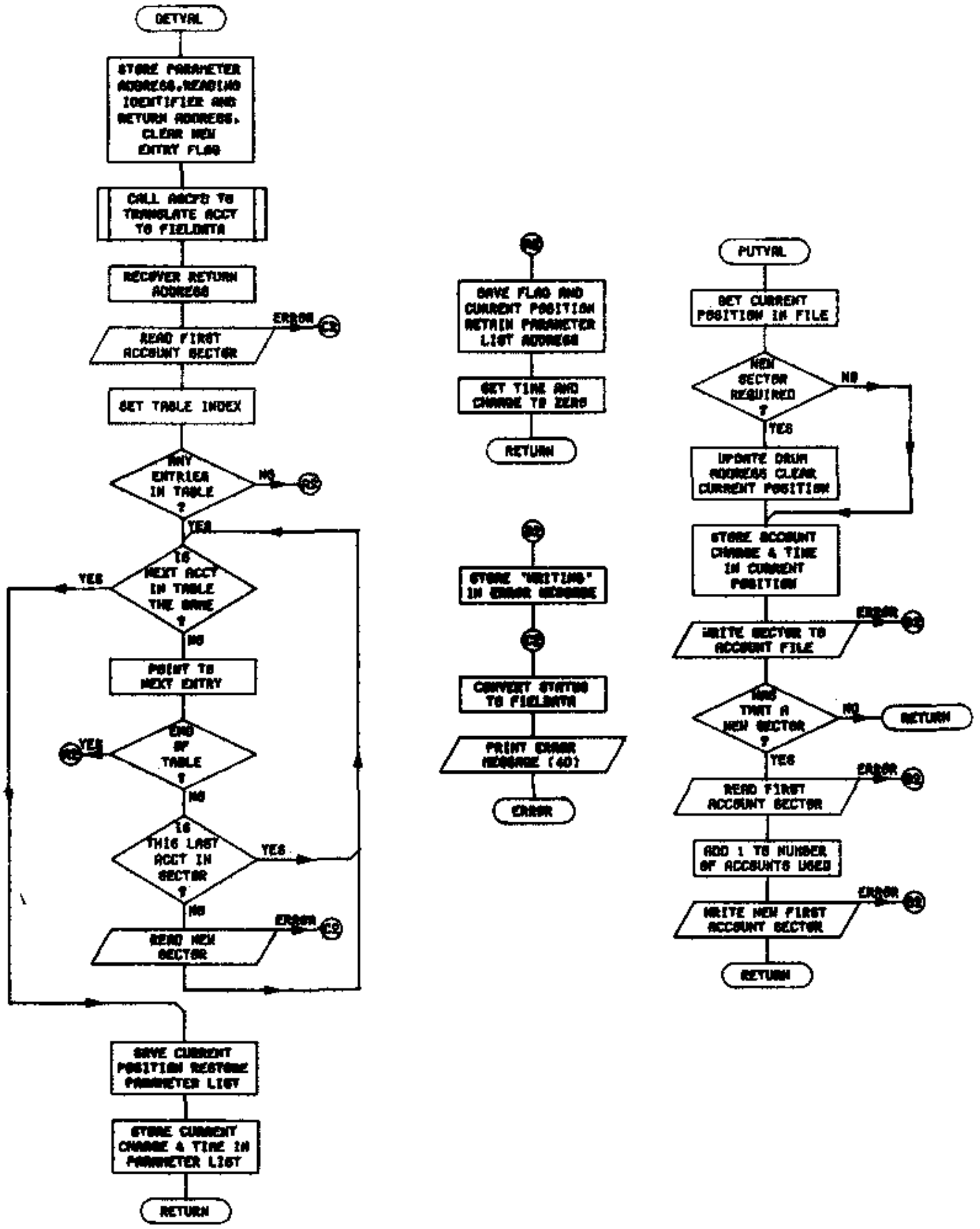
ROUTINE TAPEOUT FLOW CHART 8

APPENDIX D1


ACCNG
A1
WRITE MESSAGE 31 TO TERMINAL
READ ACCOUNT NUMBER
CALL GETVAL TO RETRIEVE CHARGE AND TIME FOR THIS ACCOUNT
WRITE VALUES OF ACCOUNT,CHARGE & TIME TO TERMINAL (MESSAGE 32)
IS CURRENT CHARGE=0 ?
NO
YES
WRITE MESSAGE 33 TO TERMINAL
READ ANSWER
IS ANSWER YES ?
NO
WRITE MESSAGE 34 TO TERMINAL
ERROR
YES
WRITE MESSAGE 35 TO TERMINAL
READ CHARGE TO BE ADDED
CALCULATE NEW TIME & CHARGE
WRITE NEW CHARGE & TIME (MESSAGE 36)
READ ANSWER
IS ANSWER YES ?
YES
B1
NO
RESTORE VALUES OF CHARGE & TIME
WRITE MESSAGE 37 TO TERMINAL
B1
CALL PUTVAL TO UPDATE ACCOUNT FILE
WRITE CHARGE TIME & ACCOUNT TO TERMINAL (MESSAGE 38)
WRITE MESSAGE 39 TO TERMINAL
READ ANSWER
IS ANSWER YES ?
YES
A1
NO
STOP
TDATE
ON ENTRY R1=DATE IN FD R2=TIME & DATE (TDATE4)
SPACE DATE OUT TO READ :- MM/DD/YY
GET TIME IN SECS PAST MIDNIGHT
DIVIDE BY 60
CONVERT REMAINDER TO CHARACTERS
IS THE TIME CONVERTED ?
NO
YES
SPACE TIME OUT TO READ :- HH:MM:SS
PRINT HEADER LINE
RETURN

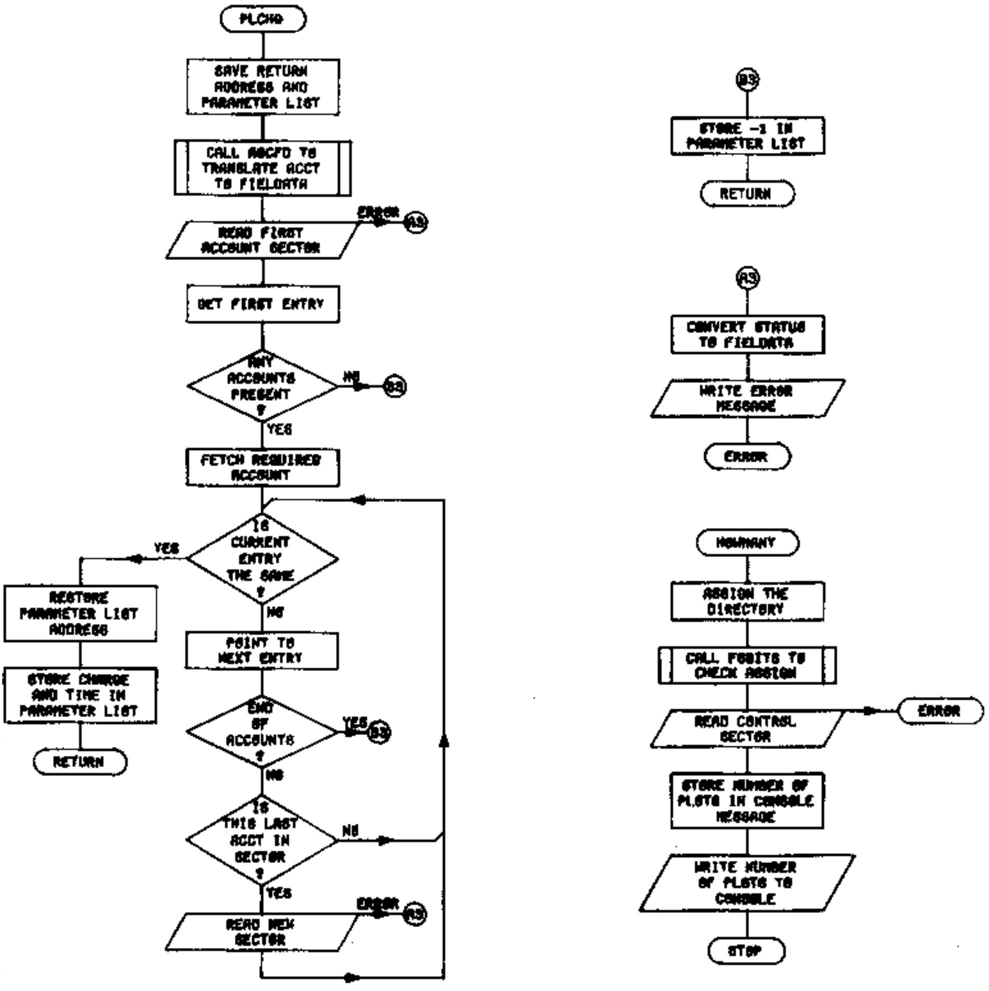
UTILITY ROUTINES FLOW CHART 1

APPENDIX D2


GETVAL
STORE PARAMETER ADDRESS, READING IDENTIFIER AND RETURN ADDRESS. CLEAR NEW ENTRY FLAG
CALL ACCFD TO TRANSLATE ACCT TO FIELDATA
RECOVER RETURN ADDRESS
READ FIRST ACCOUNT SECTOR
ERROR
SET TABLE INDEX
ANY ENTRIES IN TABLE ?
NO
YES
IS NEXT ACCT IN TABLE THE SAME ?
YES
NO
POINT TO NEXT ENTRY
END OF TABLE ?
YES
NO
IS THIS LAST ACCT IN SECTOR ?
YES
NO
READ NEW SECTOR
ERROR
SAVE CURRENT POSITION RESTORE PARAMETER LIST
STORE CURRENT CHARGE & TIME IN PARAMETER LIST
RETURN
SAVE FLAG AND CURRENT POSITION RETAIN PARAMETER LIST ADDRESS
SET TIME AND CHARGE TO ZERO
RETURN
STORE 'WRITING' IN ERROR MESSAGE
CONVERT STATUS TO FIELDATA
PRINT ERROR MESSAGE (40)
ERROR
PUTVAL
GET CURRENT POSITION IN FILE
NEW SECTOR REQUIRED ?
NO
YES
UPDATE DRUM ADDRESS CLEAR CURRENT POSITION
STORE ACCOUNT CHARGE & TIME IN CURRENT POSITION
WRITE SECTOR TO ACCOUNT FILE
ERROR
WAS THAT A NEW SECTOR ?
NO
RETURN
YES
READ FIRST ACCOUNT SECTOR
ERROR
ADD 1 TO NUMBER OF ACCOUNTS USED
WRITE NEW FIRST ACCOUNT SECTOR
ERROR
RETURN

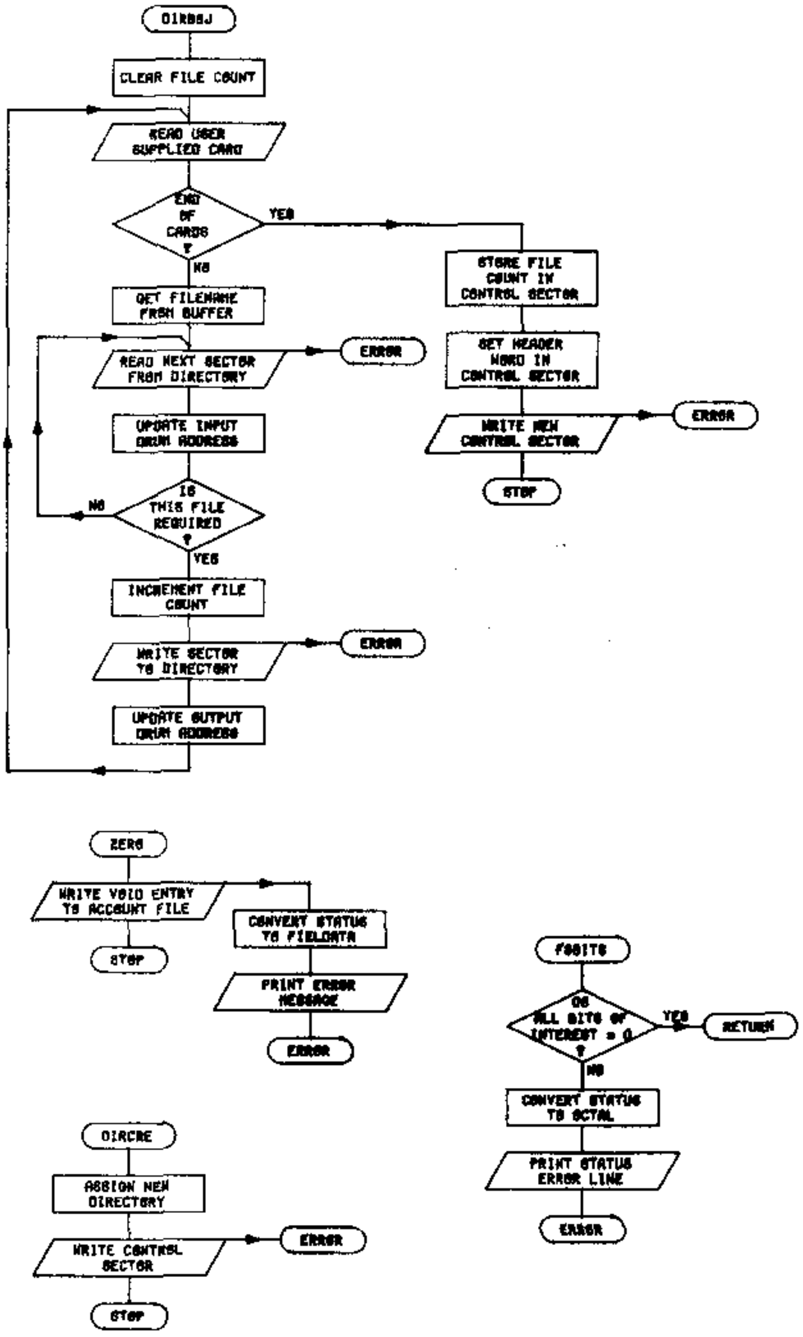
UTILITY ROUTINES FLOW CHART 2

APPENDIX D3


PLCHG
SAVE RETURN ADDRESS AND PARAMETER LIST
CALL ASCFD TO TRANSLATE ACCT TO FIELDATA
READ FIRST ACCOUNT SECTOR
ERROR
A3
GET FIRST ENTRY
ANY ACCOUNTS PRESENT ?
NO
B3
YES
FETCH REQUIRED ACCOUNT
IS CURRENT ENTRY THE SAME ?
YES
RESTORE PARAMETER LIST ADDRESS
STORE CHARGE AND TIME IN PARAMETER LIST
RETURN
NO
POINT TO NEXT ENTRY
END OF ACCOUNTS ?
YES
B3
NO
IS THIS LAST ACCT IN SECTOR ?
NO
YES
READ NEW SECTOR
ERROR
A3
B3
STORE -1 IN PARAMETER LIST
RETURN
A3
CONVERT STATUS TO FIELDATA
WRITE ERROR MESSAGE
ERROR
HOWMANY
ASSIGN THE DIRECTORY
CALL FSBITS TO CHECK ASSIGN
READ CONTROL SECTOR
ERROR
STORE NUMBER OF PLOTS IN CONSOLE MESSAGE
WRITE NUMBER OF PLOTS TO CONSOLE
STOP

UTILITY ROUTINES FLOW CHART 3

APPENDIX D4



UTILITY RØUTINES FLØW CHART 4