

C.S.I.R.O.
COMPUTING RESEARCH SECTION
CANBERRA, A.C.T.

MEMORANDUM NO. 6

THE C.S.I.R.O. ON LINE SYSTEM, USER'S GUIDE

by

T. Pearcey

C.R.S., CANBERRA

1.10.66
Edition 1.

4

[illegible]

THE C.S.I.R.O. ON-LINE SYSTEM,

USER'S GUIDE

Contents

1. INTRODUCTION
2. THE KEYBOARD DISPLAY CONSOLE
 - 2.1 Character Keys
 - 2.2 Control Keys
 - 2.3 Interrupt Key
 - 2.4 Function and Toggle Keys
3. THE DISPLAY MONITOR (DAVE)
4. THE DISPLAY LIBRARY REQUEST FORM
5. ERROR CONDITIONS
6. REQUESTS FOR DOCUMENTS
 - 6.1 EXECUTE
 - 6.2 PRINT
 - 6.3 PUNCH
 - 6.4 PLØT
 - 6.5 DELETE
 - 6.6 LØCATE
7. REPLIES AND DIAGNOSTICS TO DOCUMENT REQUESTS
8. CONTROLLING A DISPLAY PROGRAM RUN
 - 8.1 RESTART
 - 8.2 DUMP
 - 8.3 MESSAGE
9. TERMINATION OF A DISPLAY PROGRAM RUN
10. TERMINATION MESSAGES
11. DUMPS
12. STARTING A NEW RUN
13. CENTRAL RECORDING OF DISPLAY USAGE AND ACCOUNTING
14. DOCUMENT PREPARATION AND EDITING (CIDER)

15. MESSAGE Control

- 15.1 Starting a CIDER Run
- 15.2 Preparing a New Document
- 15.3 Editing an Existing Document
- 15.4 Record Size and Type
- 15.5 Tabulation Settings
- 15.6. Diagnostics
- 15.7 Request Summary
- 15.8 End of MESSAGE Control
- 15.9 Entering MESSAGE Control from FUNCTION KEY Control
- 15.10 REWIND
- 15.11 Set FØRTRAN Tabs
- 15.12 Set CØMPASS Tabs
- 15.13 FINISH
- 15.14 ABANDØN

16. FUNCTION KEY Control

- 16.1 Display Units of Information
- 16.2 Logical Records
- 16.3 Display Modes
 - 16.3.1 PAGE Mode
 - 16.3.2 RECØRD Mode
 - 16.3.3 LINE Mode
- 16.4 Changing Mode of Operation
- 16.5 Document Movement
 - 16.5.1 FØRWARD
 - 16.5.2 BACKWARD
 - 16.5.3 INSERT
 - 16.5.4 DELETE
 - 16.5.5 RESTØRE
- 16.6 Error Conditions
 - 16.6.1 Selecting an Illegal FUNCTIØN KEY
 - 16.6.2 Records Longer Than 1000 Characters

17. ON LINE PROGRAMMING (INTERP)

17.1 The Language

- 17.1.1 Variables
- 17.1.2 Variable Types
- 17.1.3 Arrays
- 17.1.4 Constants and Literals
- 17.1.5 Labels
- 17.1.6 Expressions
- 17.1.7 Functions
- 17.1.8 Declarative Statements
- 17.1.9 Replacement Statements
- 17.1.10 GO TO Statement
- 17.1.11 Conditional Statements
- 17.1.12 DO Statements
- 17.1.13 Input/Output Statements
- 17.1.14 PAUSE Statement

18. PROGRAM Mode

19. OPERATION

19.1 Checking

19.2 Execution

- 19.2.1 PAUSE
- 19.2.2 End of Executable Statements
- 19.2.3 Error Pause
- 19.2.4 READ Pause
- 19.2.5 PRINT Pause
- 19.2.6 GIVE Request
- 19.2.7 SET Request

19.3 EDIT

- 19.3.1 Scanning
- 19.3.2 INSERT
- 19.3.3 DELETE
- 19.3.4 SAVE

20. CALCULATE Mode

20.1 Result Presentation

20.2 Diagnostics

21. Termination of INTERP

APPENDIX I

A.1 Checking Phase

- A.1.1 Type 1 Diagnostics
- A.1.2 Type 2 Diagnostics

A.2 Interpretation Phase

THE C.S.I.R.O. ON-LINE SYSTEM, USER'S GUIDE

1. INTRODUCTION

This document is intended for the typical user of the C.S.I.R.O. on-line system on the Control Data 3600 at Canberra. It describes the facilities available via the six keyboard display consoles and how they may be useful to those who normally program in languages like FORTRAN. No description therefore, is given for instance of the way in which display library programs are written, checked out and incorporated into the displays program library. These details may be found in Part II of 'The DAD System, Programmers' Manual'. In the following Part I and II refer to the two parts of that manual.

The facilities naturally divide into three main components:

- (1) The Displays' Monitor, DAVE, which supervises the checking in and out of displays, the initiation and termination of display programs, the handling of all requests made by console users and I/O operations called for by display library programs.
- (2) The document* creation and editing program CIDER, allows rapid access to documents, facilities for rapid scanning of them or their alteration or creation via the keyboard.
- (3) An interpretive on-line program writing and execution system using a language basically consisting of FORTRAN-like statements.

Display library programs are special programs which are usually designed to run in intermittent bursts of a few tenths of a second duration. Each burst is initiated by a user pressing his interrupt button (see Section 2). The time taken by a user to set up his data and appreciate his results will be long compared with the duration of a burst and it is thus possible for a number of users to be operating displays at the same time without noticeably affecting the response time of the system to any user.

Servicing keyboard displays requests is carried out at a higher level of priority than the processing of the serial job stack which the central processor handles during periods between the bursts of display programs and while awaiting drum accesses by display programs. However, display programmers may themselves be interrupted to service requests for input/output activity by any other peripheral units such as card readers, punches, printers, plotters etc.

One particular feature is of importance and that is the ability of the DAVE displays monitor, to interact with the main system monitor and allow the displays to affect the job queue and other operational dynamic queues maintained by DAD. Further, CIDER allows for access to any document within the system which may then, via DAVE, be entered into the job or the output queue.

* A document is the major item of information in the DAD system. See the 'DAD System Programmers' Manual' and 'DAD for the FORTRAN User , C.R.S. Memo. No. 5'.

2.

Two aspects of the displays software design are important in this type of interactive mode of operation, first the system must always indicate the completion of a request by a change in the data displayed and second, the system must indicate operator faults and supply self-explanatory messages leading the user through his applications.

2. THE KEYBOARD DISPLAY CONSOLE

Each display console has a 14-inch cathode ray tube screen on which 63 characters chosen from the standard 64-character set (see C.R.S. Memo. No. 3) may be displayed on a 5 x 7 dot matrix format. (Underline is the only character missing and for which no key is available. It may however be provided as an output by the central processor to a display screen). A screenful or 'page', consists of 1000 characters arranged as 25 lines each of 40 characters.

Associated with the display is a keyboard consisting of a set of typewriter-like keys and a set of control keys for controlling the 'position marker' described below. In addition there are a set of 12 'function keys' and two 'toggle keys' as shown in Fig. 1.

Characters may be read from or written on the screen under computer control, or they may be entered manually using the 'typewriter' keys. In all cases the position marker, which appears as an underline, indicates the position where a character is to be written. A knurled wheel enables the display to be turned on or off and varies the brightness of the screen. No central processor time is occupied while typing into the display.

2.1 Character Keys

The character-writing keys are arranged in a typewriter fashion although only in a few cases does upper shift provide characters differing from those in lower shift as indicated in Fig. 1. Neither shift key locks.

As the character keys or space bar are depressed, characters or spaces appear on the screen at the place currently indicated by the position marker which itself moves one place to the right, or, if at the end of a line, to the left most position on the next line down or, if at the end of a page, to the top left position. The characters so written replace the characters or spaces previously occupying those positions.

This regard for line and page boundaries also applies to SKIP, SLEW, TAB, BKSP and RETURN. It also applies to computer input/output where the record length is at most 1000 characters, i.e. up to 125 words.

2.2 Control Keys

The control keys operate as follows:

- (1) CLEAR - clears all data from the screen and shifts the position marker to the top left of the page.

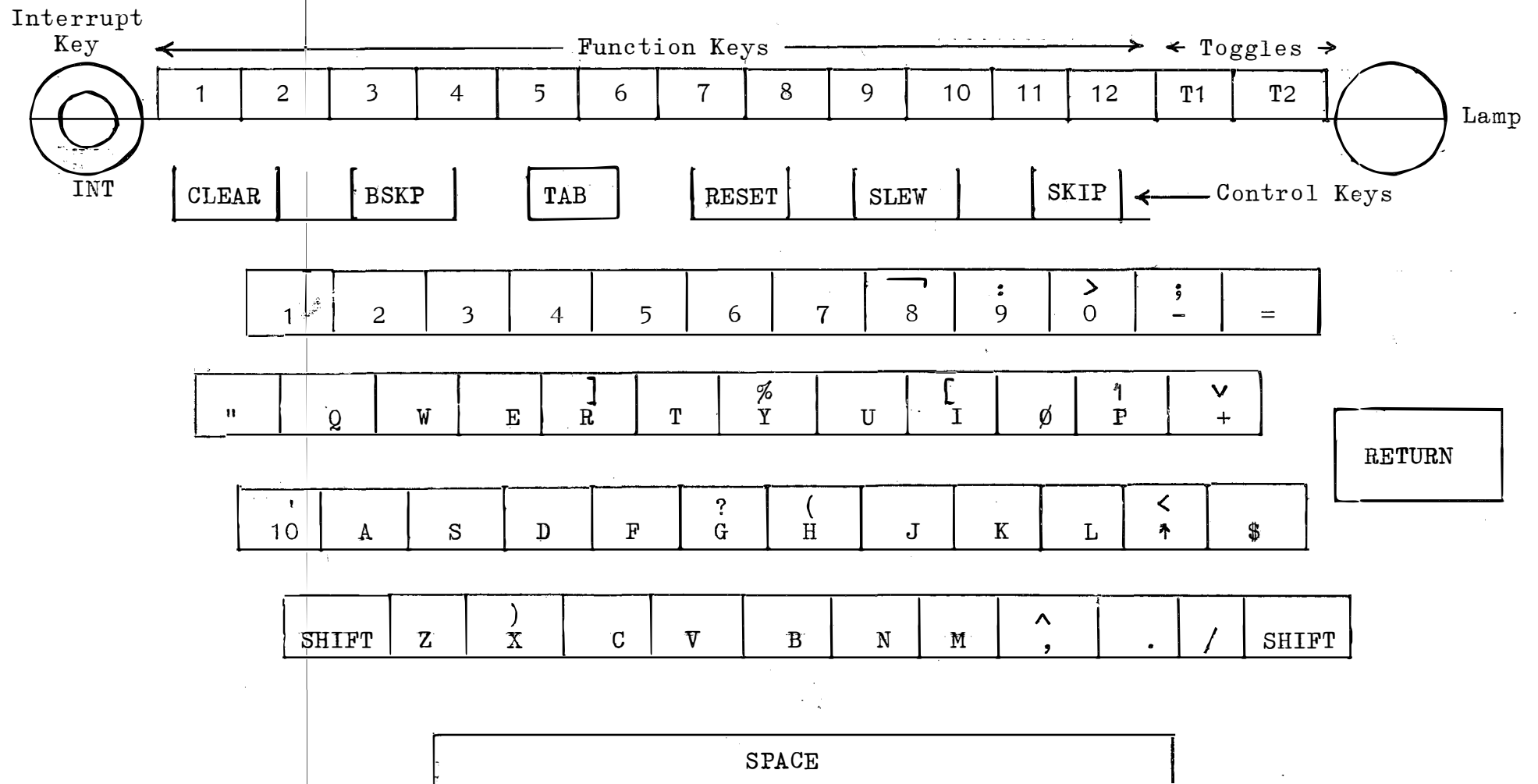


Fig. 1: The Display Keyboard

- (2) RESET - sets the marker to the top left without effecting the contents.
- (3) RETURN - sets the marker to the left of the next line. If on the last line it shifts to the top left.
- (4) SKIP - moves the marker one place to the right without effecting the character displayed. If at the end of line or page it skips to left of next line or to top left of page.
- (5) SLEW - moves the marker continuously to right, or to next line or page, without affecting displayed data.
- (6) BKSP - shifts the marker one place left, or to right of previous line or bottom right of page.
- (7) TAB - moves the marker to some preset position on the screen, ignoring line boundaries. TAB settings are not necessarily the same from line to line. The settings may be made only by a program; they cannot be set directly from the keyboard.

Generally computer programs which access displays will place tabulation settings in some way convenient to the user e.g. they may be set at positions where answers to questions may be typed.

2.3 Interrupt Key

During typing operations a display is isolated from the processor. Only when an interrupt key (INT in Fig. 1) is depressed will the processor attend to the user's request. When a request has been accepted for service and is being processed the lamp (at the top right of Fig. 1) will be lit, it will be extinguished on completion of servicing the request. The nature of the service depends upon:

- (1) which display program is being used
- (2) what is on the screen
- (3) the settings of the function keys and toggles.

2.4 Function and Toggle Keys

The position of the twelve function and two toggle keys is illustrated in Fig. 1. Only one of the function keys may be depressed at a time; depression of any one will make any other depressed key reset, although it is also possible for all keys to be reset by partially depressing the set key and releasing it. The toggle keys may be operated separately from the function keys and can each take the on or off position so that there is a total of 48 different configurations of the function and toggle keys, neglecting the special case of all function keys being off.

The settings are read by the computer when an interrupt from a display console is serviced and these configurations may be used to control the nature of the processing to occur.

Only function keys 1, 2 and 3 are used by the displays' monitor, DAVE, and no notice is taken of the toggles; these three are always reserved for DAVE.

3. THE DISPLAYS MONITOR (DAVE)

The displays monitor, DAVE, performs the functions of an interface between the displays actions and the main monitor DAD. When DAVE is operational and, irrespective of the configuration of the function and toggle keys, the user presses his interrupt button, his request will be recognised. When being serviced the light on the keyboard will come on briefly and a change will occur on the screen. If DAVE is not operational on pressing the interrupt button, the marker will be returned to the top left position and the console will be 'frozen out', the interrupt button will have no effect other than setting the marker to the top left and it will be impossible to type on the screen.

DAVE handles all requests for display program users, loads programs, provides for normal and abnormal termination, accounts for central processor and console times and arranges for provision and printing of dumps.

4. THE DISPLAY LIBRARY REQUEST FORM

When DAVE is operational it first assigns to each display a special display program and writes a 'display program request form' on the screen below which is placed a 'document request form' (see Section 6).

The display program request form is as follows:

```

DISPLAY PROGRAM REQUEST FORM
CHARGE CODE?
DSP PROGRAM?
TIME NEEDED?
EXTRA IDENT?
(OPTIONAL)

```

The user depresses the TAB key until the position marker is to the right of the respective lines. The user's 8-character charge code is placed on the second line, the TAB key is again pressed and the title of the desired display program (one in the displays' library) is entered on the next line. After Tab-ing again the amount of time required at the console (not the central processor time) in minutes (not greater than 199 minutes) is placed in the next line. After a final TAB operation the last line may receive any extra identification to distinguish it from the runs of the same type, e.g. RUN n, for the nth run of the same display program on a particular day, (any output will receive the additional identifying heading). Use of this fifth line is optional.

If, with the MESSAGE function key 3 set, the user presses the interrupt button, INT, the requested program, or its first part will be loaded and entered, and that program will identify itself.

The information entered into the display program request, including the extra identification, appears on all printed dumps taken of the consoles program area.

The Display Program Request Form may be eventually returned if the START key is depressed and one or more interrupts generated.

5. ERROR CONDITIONS

Under certain conditions the completed request form is returned to the user with an additional diagnostic message placed at the bottom of the screen. These conditions are as follows:

1. The MESSAGE key is not set when requesting a display program run. The diagnostic at the foot of the screen will be:-

FUNCTION KEY ERRØR. TRY MESSAGE KEY (3)

2. The charge code typed must be in the list of legal charge codes allowed by DAD. If it is not valid or is blank the message INVALID will appear at the right hand end of the line containing the question CHARGE CODE?
3. The time typed in must be all numeric, must be greater than zero and equal to or less than 199 minutes. If these conditions are violated the word INVALID will occur at the right hand end of the line containing TIME NEEDED?
4. The program name must be that of a program currently held by the system and is stored with the special format and charge code of display library programs. If the program is not on the drum the diagnostic:

CANNØT FIND PRØGRAM ØN DRUM

will appear on the last line of the screen.

5. The charge code, program name, and time needed must be typed in their appropriate lines. They may occur anywhere except the last 8 positions on the current line and may contain embedded blanks. Failure to observe these conditions will lead to the error conditions 2, 3 and 4 above.
6. No error is possible in typing the extra identification. The user should observe the rules in 5 above if he wants what he types to be seen and recorded.
7. The message:

SYSTEM BUSY. TRY AGAIN

may be returned. The user should repeat his interrupt until he gets service.

6.

It is possible to recover from all error conditions by interrupting again with the correct entries on the screen and the MESSAGE key depressed. The new entries will be accepted even in the presence of diagnostics.

6. REQUESTS FOR DOCUMENTS

A Document Request Form accompanies and follows the Display Program Request form on the same page and appears as follows:

```
REQUESTS FOR DRUM DOCUMENTS
EXECUTE,
PRINT,
PUNCH,
PLØT,
DELETE,
LØCATE,
```

The user may make one request by placing an asterisk to the left of the function required and entering the document name, and the data as described below, to the right of the function selected. The following are the request formats and operations on the named document.

6.1 EXECUTE

The format is:

```
*EXECUTE,c,i,t,
```

where c is the charge code of 8 characters, i is the document title of up to 8 characters (or the last 8 if more than 8 are given; c and i together form the document name and t is the total expected time, in minutes, for processing the document. All fields are obligatory. The EXECUTE request will cause the document name to be entered onto Execution List EL1 if t is less than a standard amount which is a system parameter (currently 5 minutes), (see Pt. I) otherwise it is placed onto EL2.

In due course the document is executed as a program.

The document may be retained within the system if it contains the EQUIP statement

```
*EQUIP, 60 = SV
```

If a number of editions of the named document exist, that with the highest edition number is accepted for processing.

6.2 PRINT

The format is:

*PRINT,c,i,ed,SV

where c, and i constitute the name of the document to be printed, ed is the edition number and SV is an optional code. This causes the document to be listed in the list OL1 if the length of the document is less than a certain size which is a system parameter (currently, 50 drum segments*) otherwise it is entered into OL2. It is retained in the system if SV is present, otherwise it is deleted after listing. If ed is absent the latest edition is accepted but the comma must be entered (see Pt. I).

It is to be remembered that the document will be taken as being in BCD code with an appropriate line control character at the front of each record.

6.3 PUNCH

The format is:

*PUNCH,c,i,ed,SV

where the parameters are as for PRINT. If ed is absent the comma must be present and the latest edition is adopted. The whole record is punched onto one or more cards and may be in BCD or BIN according to the setting of the mode indicator in the segment inter-record words (see Pt. I).

6.4 PLOT

The format is:

*PLOT,c,i,ed,SV

where the parameters are as for PRINT. The records must be in suitable format for plotting. Plotting will occur on the large plotter.

6.5 DELETE

The format is:

*DELETE,c,i,ed

in which c, i and ed are the document name and edition number. If ed is absent the latest edition is deleted unconditionally. This statement can be accepted only if the document is inactive i.e. not in use by an execution, under execution or being input or output. If the document is active a diagnostic appears on the bottom of the screen (see Section 7).

* a drum segment corresponds to about 12 printer lines

6.6 LOCATE

The LOCATE request enables the user to discover the status of a document. Its format is:

*LOCATE,c,i,ed

where the parameters are as for DELETE. If ed is absent the latest edition is located.

When a document is in one of the execution lists the expected waiting time before the document can reach execution is returned to the user (see Section 7). This estimate of waiting time does not include the time the current job has yet to run and does not allow for executions which, although listed earlier than the document in question, may not be capable of running due to the unavailability of subject documents.

7. REPLIES AND DIAGNOSTICS TO DOCUMENT REQUESTS

All replies and diagnostics following a document request contain a repetition of the request name in the first 8 places of the bottom line. The remainder of the line contains one of the following replies or diagnostics:

1. --DOCUMENT ØN DRUM, NØT IN USE
2. --DOCUMENT ON DRUM, IN ELn, WAIT m
(where n=1 or 2 and m is an integer)
3. --DOCUMENT ALREADY IN USE
4. --REQUEST ACCEPTED
5. --DOCUMENT NØT ØN DRUM
6. --SYSTEMS DOCUMENT. CHECK
7. --SYSTEM BUSY. TRY AGAIN

These are replies to a legally formatted request and replies 1 and 2 apply only to the LOCATE request, while 3 implies that the document is active, being executed, used by an execution or being input or output. Reply 6 occurs only when a user attempts to delete a system document e.g. a library display program. If the request is really intended the user may interrupt again to get his requested accepted.

Others are:

8. --ØNE REQUEST AT A TIME, PLEASE
9. --NØT ENØUGH INFØRMATIØN
10. --FØURTH FIELD MUST BE NUMERIC
11. --EDITIØN NUMBER TØØ LARGE
12. --TIME FIELD ZERØ ØR TØØ LARGE
13. --FIFTH FIELD CAN ØNLY BE SV

9.

These diagnostics contain no repetition of the request but only indicate a badly formatted request. 8 implies that more than one request has been marked with an asterisk, while 9 implies that the name fields are incompletely filled in. Diagnostics 10, 11 and 12 indicate the execution time or edition number are incorrectly entered or are blank, while 13 applies to the SV option.

Note that no checking of the name field can be undertaken. If it has been misspelled the name will usually be non-existent and the diagnostic 5 will appear.

8. CONTROLLING A DISPLAY PROGRAM RUN

Once the user has requested and obtained use of a particular display library program he controls all further operations himself, within the limits of the selected program, by means of function key settings type-ins and interrupts.

The user may interrupt at any time. Whilst he is receiving service the light at the right hand top corner of the keyboard will be lit and during the time the light is operating the user should not interrupt again. If he does, his interrupt is ignored.

Three of the function keys numbers 1, 2 and 3 are reserved for use by DAVE and have special names as follows:

<u>Key Number</u>	<u>Key Name</u>
1	RESTART
2	DUMP
3	MESSAGE

Any or all the remaining keys are at the disposal of, and used by, display library programs e.g. CIDER.

8.1 RESTART

If this key is down when the user interrupts, the run of the display program is immediately terminated and the display will enter the end-of-job accounting phase (see Section 10). This key is obeyed whether or not a request is already in execution. Thus, if a program gets into an endless loop, this key can be used to halt it.

Certain simple display programs use this key as their normal method of termination (see Pt. II) but more complex ones such as CIDER provide their own termination.

8.2 DUMP

On interrupting with this key down a binary dump of the display program will be taken by the monitor and stored on the drum. The user has the option, at the end of the run, to have such dumps formatted and printed (see Section 11). This key is obeyed whether or not a request is already in execution.

Execution will be resumed after the dump has been taken. Normally the DUMP key will be used only by programmers who are in process of checking out their display library programs.

8.3 MESSAGE

If a user interrupts with this key set, the contents of the screen are to be interpreted as a message. ~~Programs not~~ requiring additional function keys will operate with this key depressed throughout their run.

9. TERMINATION OF A DISPLAY PROGRAM RUN

A program may be terminated either by using the RESTART key or by requesting the program to terminate itself, or the termination may be abnormal due to expiration of time or a fault condition when diagnostics are presented with the termination message (see Section 10).

1. TIMESUP. The time requested on the program request form has run out. This is a diagnostic and shown on the screen.
2. NØ RAND. ACC. DRUM. If this diagnostic appears it will be after an initial display program request. It indicates that the monitor is unable to allocate random access drum space for the copy of the display program.
3. DRUM ØVERFLØW. This diagnostic means there is not space on the drum.

10. TERMINATION MESSAGES

At termination of a display library program a termination message is presented containing the status of the termination, accounting data in terms of central processor time occupied, number of I/O operations and a list of any dumps taken by the terminating run. If dumps have been taken this message could be of the format such as:

```

charge code,      program title,      extra ident
ABNØRMAL TERMINATION      BAD SENTRY CALL
SELECT  DUMPS TØ BE FØRMATTED AND PRINTED
DUMP n
DUMP n'
DUMP n"

```

n, n' and n" etc. are integers identifying the dumps. The user places an asterisk (or any non blank character) to the right of the dumps which are required for testing. The word ABNØRMAL may be replaced by NØRMAL in case of normal termination and no diagnostic is given.

If no dumps have been taken or after a further interruption, and after the above message has been presented, the following type of terminal message will appear:

```

charge code,      program title,      extra ident
ABNØRMAL TERMINATIØN      TIMESUP
CØNSØLE TIME      nMIN      ØPTIME      mSEC
s I/Ø ØPERATIØNS

```

where n, m and s are appropriate integers. The word ABNØRMAL may be replaced by NØRMAL and the diagnostic is omitted.

11. DUMPS

A dump may be taken at any time. More than one dump may be selected and they are formatted and printed in the order listed. Dumps are serially numbered and more than one user may call dumps in the same period of time; only the current user's dumps are presented for selection.

A dump by DAVE provides full details such as the screen, part of DAVE, DAVE tables, PIGEON HOLES and the 2000 words of display program area. Layout is as follows:

```

DØC charge code, title, edition, time, date
(page eject)
DISPLAY PRØGRAM DUMP 00Qn

```

```

Users charge code display program name
ABNØRMAL TERMINATIØN cause of termination

```

Screen

```

1 2 3 4 5 6 7 8 9 0 1 2 3 4 ... (40 positions)

```

```

...

```

```

1

```

```

2

```

```

3 [Screen picture in BCD]

```

```

.

```

```

.

```

```

.

```

```

25

```

```

DUMP TAKEN AT time* ØN date, ØN DISPLAY m

```

```

A      )
Q      )
D      )
MS,NC  )
BR      )
IM      ) 3600 working registers contents in octal
IR      )
(ZERØ) )

```

* time is recorded to the nearest second

B1,B2 }
 B3,B4 }
 B5,B6 }
 (page eject)

DAVE

011057 relative) 4 columns of successive words in normal
 011063 addresses) dump format. i.e.
 absolute starting } half word instructions, 3 letter mnemonics
 at 175 } + octal value
 addresses " } full word instructions, 4 letter mnemonics
 } + octal value
 12223 ")
 (page eject)

DAVE TABLES

012250)
 to }
 012261) 4 columns of octal values
 (page eject)

PIGEON-HOLE

012262)
}
 012336) 4 columns of octal values
 (page eject)

(Display Program Name)

012342)
 }
 012745)
 }
} 3 columns of 3 letter mnemonics + octal values
 016260) or 4 letter mnemonics + octal values
 } + 4th column of 3 words written in BCD
 (24 characters)
 (page eject)

12. STARTING A NEW RUN

When the termination message is obtained the user may interrupt with any function key set to get a fresh copy of the display program request form and he may then proceed to make fresh requests or leave the console.

13. CENTRAL RECORDING OF DISPLAY USAGE AND ACCOUNTING

Record of the start and termination of a display program run and any document request is sent to the operator's information log (ØIL, see Pt. I). Every message contains the number of the display and the time.

In case of an abnormal termination the cause is indicated. Processor and console time are also indicated.

At the end of a display program run, accounting records are passed to the accounting media (ACM). Processor time is charged with a nominal additional charge for the time the display has been in use and for I/O calls. Processes called by any request made via the document request form is charged normally by DAD (see Pt. I).

14. DOCUMENTS PREPARATION AND EDITING (CIDER)

CIDER (Consoles Input Document Editng Routine) provides facilities for writing and loading DAD documents into the document holding storage, and for retrieving documents for inspection and editing. In conjunction with DAVE it is possible to write and edit documents and have them executed, printed, punched, plotted or deleted.

When using CIDER it is possible to scan and edit a document rapidly page by page, record by record or line by line, by using some of the remaining function keys provided on the console's keyboard.

There are two methods of controlling operations on documents using CIDER. The first of these, MESSAGE control, allows the contents of the screen to be accepted as a message defining the following course of action. In the second, FUNCTION KEY control, the state of the function keys defines the nature of the following operation in association with type-ins and interruptions.

15. MESSAGE Control

DAVE always hands control to CIDER in MESSAGE control status. CIDER then presents the user with messages which require type-in replies which correspond to completing forms similar to those of DAVE. At some stage CIDER will tell the user that his console is under FUNCTION KEY control after which access to, and editing of, a document or creation of new documents may be carried out on the keyboard with only occasional use of the function keys. Very little central processor time will be used.

When CIDER displays a message requiring an answer it provides underline marks in the positions where the replies must be entered and tabulation settings are so arranged that the position marker can easily be placed at the entry position by use of the TAB key.

If, when under MESSAGE control, the interrupt button is depressed with any function key other than the MESSAGE key set the current message will be written with the following diagnostic added at the bottom of the screen.

***E ILLEGAL FUNCTION KEY SELECTED TØ
CØNTINUE SELECT MESSAGE (MESS).

The user must depress the MESSAGE key and interrupt.

15.1 Starting a CIDER Run

The first message presented by CIDER is the Document Request Form:

```

CIDER READY
ENTER REPLIES ON THE UNDERLINED SECTIONS
PREPARE OR EDIT? ENTER P OR E.      -
DOCUMENT IDENTIFICATION. (THE CHARGE CODE
IS NOT NEEDED IF DOCUMENT IS YOUR OWN).
CHARGE CODE? -----
TITLE? ----- EDITION NUMBER? --
IF EDITING, A COPY MUST BE TAKEN. ENTER
IDENTIFICATION FOR NEW VERSION.
TITLE? ----- EDITION NUMBER? --
DO YOU WANT FIXED LENGTH RECORDS? IF SO,
HOW MANY LINES PER RECORD?      --
OTHERWISE YOU MUST USE VARIABLE LENGTH
RECORDS WITH "*" AS END OF RECORD MARKER
OR ANY ALTERNATIVE YOU SUPPLY,
ALTERNATIVE MARKER?      -
DO YOU WANT TABS SET? IF SO ENTER F FOR
FORTRAN OR C FOR COMPASS.      -

```

This form is filled in and returned by an interrupt. CIDER checks the charge code for only letter and asterisks.

15.2 Preparing a New Document

If the user wishes to prepare a document a P is entered and no charge code is needed, his own will be adopted as already provided by DAVE, the title and edition only are needed. If the title is new it will be given unity edition number. Edition numbers ranging from 1 to 99 or a blank are permitted, a blank is given an edition number of unity.

CIDER reacts to this type of reply by returning the statement listed in Section 15.8 thus entering FUNCTION KEY control. Under some circumstances a diagnostic will be presented (see Section 15.6).

15.3 Editing an Existing Document

If it is desired to edit a document as indicated by entering E, and the document is not under the user's own charge code the full name and edition must be entered otherwise only the title and edition are required. This document will be copied and the copy only will be provided to the user. The copy name will have the user's charge code. The original document, if it is the user's own and if it is no longer required, should be deleted when editing is completed (see Section 6.5).

While a copy is being created the following message is shown:

```

THE DOCUMENT IS BEING COPIED
WAIT

```

and will be replaced by the Summary Message (see Section 15.7).

15.4 Record Size and Type

Documents can be manipulated using fixed or variable record lengths. The maximum size of a record is 1000 characters. Fixed length records are selected by entering the number of lines per record into CIDER's starting message (see Section 15.1). If the entry is left blank, variable length records are assumed. Normally the vertical line "|" will be adopted as the record marker but the user may specify his own marker characters in the appropriate place.

15.5 Tabulation Settings

By depressing the TAB key the position marker can be moved to preset positions. CIDER allows of two special layout formats, one for FØRTRAN having tabulation settings in the first and 7th positions from the left of each line, and one for CØMPASS with settings in the 1st, 10th and 20th positions from the left of each line according as F or C are entered respectively.

15.6 Diagnostics

Under MESSAGE control the following diagnostics may appear in reply to the document request form as follows:

1. THE ØORIGINAL DØCUMENT
charge code, title, edition
DØES NØT EXIST.
CØNTINUE? __
2. THE ØORIGINAL DØCUMENT
charge code, title, edition
IS BUSY.
CØNTINUE? __
3. The ØORIGINAL DØCUMENT
charge code, title, edition
IS ØN THE ØFFLØAD UNIT.
CØNTINUE? __
4. THE CØPY name
charge code, title, edition
ALREADY EXISTS
CØNTINUE? __
5. THE DØCUMENT
charge code, title, edition
ALREADY EXISTS.
CØNTINUE? __

These messages have the additional annotation at the bottom of the page

(TØ CØNTINUE INTERRUPT, ØTHERWISE TAB,
TYPE IN 'NØ' AND THEN INTERRUPT)

They are self explanatory. A 'YES' reply is made by simply interrupting, otherwise a NØ is entered where indicated and interrupt will cause termination of the CIDER run (see Section 15.9).

A further class of diagnostic is returned if the document request form is wrongly completed. The whole form is then returned as entered together with diagnostics placed at the bottom on the screen. Thus:

1. ***E ILLEGAL FUNCTION KEY SELECTED.
TØ CØNTINUE SELECT MESSAGE (MESS)
Set the key named and interrupt.
2. ***E, YØU MUST ENTER P ØR E FØR
PREPARE ØR EDIT.
Only P or E are legal symbols, Correct and interrupt.
3. ***E THE CHARGE CØDE IS ILLEGAL.
The charge code entered is not on the legal list
Have it properly registered by CSIRO operations staff.
4. ***E ENTER A DØCUMENT TITLE.
A title of all blanks is illegal.
5. ***E THE DØCUMENT EDITIØN NUMBER IS
ILLEGAL.IT MUST BE 1 TØ 99.
An edition number must be blank or numeric 1 to 99.
6. ***E THE CØPY EDITION NUMBER IS
ILLEGAL. IT MUST BE 1 TØ 99.
7. ***E FIXED LENGTH RECØRDS WITH AN END
ØF RECØRD MARKER IS IMPØSSIBLE.
Both fixed records and an end of record symbol
must have been selected. Remove one or the other
as needed.
8. ***E THE RECØRD LENGTH IS ILLEGAL. IT
MUST BE 1 TØ 25 LINES.
The record length must be numeric and betwween
1 and 25 lines.
9. ***E FOR CØMPASS AND FØRTRAN TABS, ØNLY
F,C ØR BLANK ARE LEGAL.
Change the illegal character. An opportunity is
given later for changing the tabulation settings.

15.7 Request Summary

If the user elects to continue after he has presented his document request form and had it accepted, a summary of his specifications for the remainder of the run is presented as follows:

SUMMARY
 THE DØCUMENT YØU HAVE SELECTED TØ
 PREPARE
 IS charge code, title, edition
 WITH VARIABLE LENGTH RECØRDS
 AND WITH
 "1" AS END ØF RECØRD MARKER.

CØNTINUE?

--

(TØ CØNTINUE INTERRUPT, ØTHERWISE TAB
 TYPE IN NØ AND THEN INTERRUPT).

Where the '1' may be any other character selected. If fixed length records have been selected the summary is displayed as follows:

SUMMARY
 THE DØCUMENT YØU HAVE SELECTED TØ
 PREPARE
 IS charge code, title, edition
 WITH FIXED LENGTH RECØRDS
 ØF n LINES PER RECØRD.
 THE CØPY IS
 charge code, title, edition
 FØRTRAN TABS WILL BE SET.

CØNTINUE?

--

(TØ CØNTINUE INTERRUPT, ØTHERWISE TAB
 TYPE IN NØ AND THEN INTERRUPT).

where n is an integer and is the number of lines selected and 'FØRTRAN' may be replaced by 'CØMPASS' or the line will be absent according to the tabs selected.

If the user elects to edit a document, 'PREPARE' in the above message is replaced by 'EDIT'. If the user is to prepare a document the next message will be the final message under MESSAGE control, (see Section 15.8).

If a copy is to be edited a diagnostic may follow. A document containing records longer than 125 words, i.e. more than 1000 characters will cause the following to be displayed:

THE ØRIGINAL DØCUMENT CØNTAINED
 n RECØRDS ØF GREATER THAN
 125 WØRDS
 THESE RECØRDS WERE TRUNCATED
 WHEN THEY WERE CØPIED.
 CØNTINUE?

--

(TØ CØNTINUE INTERRUPT, ØTHERWISE TAB
 TYPE IN NØ AND THEN INTERRUPT).

where n is an integer.

The largest record that CIDER can pass to a user is 125 words of 8 characters/word. If records exceeding that length are found only the first 125 words of each are copied. If the user elects to continue by just interrupting, CIDER displays one further message (see Section 15.8) and transfers to FUNCTION KEY control in a mode appropriate to the record type chosen by the user.

15.8 End of MESSAGE Control

When CIDER has obtained the required document satisfactorily it will display its final message under MESSAGE control:

```

                END MESSAGE MØDE.
YØU ARE NØW IN X MØDE UNDER FUNCTION
KEY CØNTRØL.
IF YØU ARE PREPARING A DØCUMENT SELECT
THE INSERT (NSRT) KEY.
IF YØU ARE EDITING A DØCUMENT SELECT
THE MØVE FØRWARD (F'WD) KEY.
YØU MAY RE-ENTER MESSAGE MØDE AT ANY
TIME BY SELECTING THE MESSAGE KEY.THE
SCREEN CØNTENTS WILL BE SAVED AND
RESTØRED WHEN YØU ARE RETURNED TØ
FUNCTION KEY CØNTRØL.EXAMPLE,SELECT MESS
TØ FINISH ØR REWIND.
THE WHITE FUNCTION KEYS ARE LABELLED
WITH MNEMONICS AS FØLLØWS:
START-DØ NØT USE.  DUMP-DØ NØT USE.
F'WD-MØVE FØRWARD.  BACK-MØVE BACKWARD.
NSRT-INSERT.      D'LT-DELETE.
R'ST-RESTØRE.
PAGE-PAGE MØDE.  (PERFØRM ØPERATIONS IN
                UNITS ØF A PAGE)
LINE-LINE MØDE.  RCD-RECØRD MØDE.

```

where X is the mode selected by CIDER on passing from MESSAGE to FUNCTION KEY control.

When preparing a document with variable length records CIDER transfers to PAGE mode. When preparing or editing fixed length records it transfers to RECØRD mode, and when editing variable length records it transfers to LINE mode. Following this the user may transfer to either of the modes by selecting the appropriate mode key and interrupting.

When preparing a document CIDER displays a page with a blank record (if variable length records have been selected the record is one line long) at the top of the page and the rest of the page is filled with blank lines starting with *EØD. When editing a document CIDER will display the first part of the document appropriately.

15.9 Entering MESSAGE Control From FUNCTION KEY Control

If the user interrupts with the MESSAGE key selected, CIDER transfers to MESSAGE control with the following message displayed:

SELECT THE REQUIRED ACTION BY ENTERING
YES IN THE CORRECT POSITION.

```

                REWIND? ---
            SET FØRTRAN TABS? ---
            SET CØMPASS TABS? ---
                FINISH? ---
        (SAVE THE DØCUMENT)
                ABANDØN? ---
        (DELETE THE DØCUMENT)

```

The user can select an action by entering YES in the appropriate place. A reply must be made or the response will be

***E YØU MUST SELECT ØNE ØF THE ABØVE ACTIONS

placed at the bottom on the screen. On interrupting the appropriate action occurs as in the following sections. If YES is placed in more than one place only the first is acted upon. If this point is reached from a display program by accidental keying, a selection of 'SET FØRTRAN TABS' and interrupt will restore the user to the previous state.

15.10 REWIND

CIDER rewinds the document to load point and then displays the message:

```

                charge code, title, edition
            REWØUND
        YØU ARE NØW IN X MØDE UNDER FUNCTION KEY CØNTRØL
        AT THE LØAD PØINT ØF YØUR DØCUMENT.
        TØ CØNTINUE SELECT THE MØVE FØRWARD(F'WD) KEY.

```

where X is the mode appropriate. If fixed length records are being used X will be RECORD mode, otherwise LINE mode.

The user should now depress the forward key and interrupt and the first part of the document will be presented.

When using variable length records it is possible to have a display on the screen, which, if moved forward, would produce a record of greater than 125 words. If an attempt is made to FINISH or REWIND in this situation CIDER will display the message

```

        FINISH REQUEST CANCELLED
        RECØRD ØVERFLØW WILL ØCCUR IF LINE N
        IS ADVANCED, TØ PRØCEED, PRESS RESTØRE
        YOU ARE UNDER FK CONTROL

```

where N is an integer giving the present line number at which overflow of the record will occur.

The user should select the RESTORE key and interrupt, then the display which was on the screen before message mode was entered will be returned and the user can proceed under function key control (section 6.2)

15.11 Set FØRTRAN Tabs

CIDER will set tabulation positions in columns 1 and 7 of the page and return the screen as it was before message control was entered (from the preceding FØRTRAN control condition). The display is again under function key control and the user should select whichever key he wishes.

15.12 Set COMPASS Tabs

CIDER will set tabulation positions at columns 1, 10 and 20 of the page and then return the screen as it was before message mode was entered. The display will then be under function key control and the user should select whichever key he wishes.

15.13 FINISH

CIDER writes onto the document whatever was on the screen before MESSAGE mode was entered, then releases the document and saves it on the drum. CIDER then displays the message:

```
JØB FINISHED
charge code, title, edition
SAVED
RESTART?
```

(TØ RESTART INTERRUPT, ØTHERWISE TAB
IN NØ AND THEN INTERRUPT)

On a 'YES' reply, by an interrupt only, the initial message is displayed (see Section 6.1.1), whilst if NØ is entered, termination of the CIDER run occurs. The display is then handed back to DAVE which displays the accounting message (see Section 10).

The behaviour on the occurrence of a document larger than 1000 characters is the same as that occurring in the case of FINISH except that REWIND replaces FINISH in the diagnostic.

15.14 ABANDØN

If ABANDØN is selected CIDER releases the document, causes it to be deleted from the system and displays the message:

```
JØB ABANDØNED
charge code, title, edition
DELETED
RESTART?
```

(TØ RESTART INTERRUPT, ØTHERWISE TAB
TYPE IN NØ AND THEN INTERRUPT)

In case of RESTART, indicated by an interrupt only, the initial CIDER message (see Section 15.1) is displayed, if case NØ is replied the display is passed to DAVE which displays the accounting record (see Section 10).

16. FUNCTION KEY Control

When inspecting, editing or writing a document using a display it is inconvenient to have to supply a keyboard message every time the user wishes a standard type of move to be made e.g. 'move forward one record', 'move backward one record' etc. Thus the function of CIDER depends on both the 'conversational' keyboard message and reply behaviour called MESSAGE CONTROL and on the response to activities provided by the user by selection of single function keys in a manner known as FUNCTION KEY control.

The user is informed when CIDER enters FUNCTION KEY control and thereafter, until it passes back to MESSAGE control, operation consists entirely of selection of a particular function key, typing directly into documents followed by depression of the interrupt button. For the direct handling of documents special functions have been associated with nine of the function keys as follows:-

<u>Key Number</u>	<u>Function</u>
3	MESSAGE
4	PAGE MODE DISPLAY
5	LINE MODE DISPLAY
6	RECORD MODE DISPLAY
7	MOVE ONE DUI FORWARD
8	MOVE ONE DUI BACKWARD
9	INSERT ONE DUI
10	DELETE ONE DUI
11	RESTORE DISPLAY

where DUI means Display Unit of Information (see Section 16.1). The MESSAGE key is that used by DAVE during the initial and final stages of use of a display console.

16.1 Display Units of Information

There are three distinct modes of operation on display units of information (DUI) and these units consist of a string of 125 words (including blank fills, (see Section 16.2)), a logical record or a single line corresponding to PAGE, RECORD and LINE modes respectively.

16.2 Logical Records

Logical records may be selected to be of fixed length or variable length and, in the latter case, any symbol may be selected as an end of record marker, although the vertical line '|' is adopted if the user does not elect to specify a different character.

Logical records are shown on the screen with their last lines blank filled except for the last character of the last line of each variable length record which, (see Section 16.3.2), will contain the end of Record mark.

16.3 Display Modes

Upon CIDER entering FUNCTION KEY control a message stating which mode is initially set is displayed (see Section 6.1.2) and will be one or other of PAGE, LINE or RECØRD modes. A change from one mode to another at any stage of operation under FUNCTION KEY control is achieved by depressing the appropriate function key and interrupting. If these key actions cause a change of mode a message is displayed to indicate the mode selected. Pressing the interrupt button again will return the previous screen contents to view suitably adjusted for the conditions of new mode if needed.

16.3.1 PAGE Mode

PAGE mode is entered by CIDER on passing to FUNCTION KEY control when variable length records are selected or when the user selects PAGE mode by depressing the PAGE key and interrupting. The response to depression of the PAGE key is the statement:

****PAGE MØDE SELECTED****

and the screen contents filled to a full page, are restored on interrupting again. The screen contents may be changed to the adjacent page ahead or that following according as the function keys FORWARD or BACKWARD are set on interrupting. Logical records each occupy an integral number of lines with blank fill and end of record marker when appropriate. A page fills every line of the display. Part of a record may be displayed at the bottom of the screen.

16.3.2 RECØRD Mode

Record mode is entered when starting to prepare or edit fixed length records, or by depressing the RECØRD key and interrupting. The screen will then show one or more complete logical records (DUIs) as may be completely fitted onto the screen. The screen will first show the message:

****RECØRD MØDE SELECTED****

on interrupting. The screen may be restored or filled anew by interrupting again. There may be blank lines at the bottom of the screen if the next non-displayed record exceeds the space available on the screen. Each record occupies an integral number of lines with blank fill on the last line, and with an end of record mark at the right hand end of the last line if variable record length is selected. If PAGE or LINE mode was previously selected all lines of information between the last end of record marker and the end of the screen will be moved backwards off the screen to give an integral number of records displayed.

16.3.3. LINE Mode

CIDER enters LINE mode if variable length records have been selected or if the LINE key is set on interrupting. In the latter case the message issued to the user is:

****LINE MØDE SELECTED****

and on re-interrupting the screen will be filled with the current document page. If the previous mode was ~~RECØRD~~ mode any blank lines at the bottom will be filled with part of the next record.

Records in this mode occupy an integral number of lines with blank filled in the last line and end of record mark if needed.

16.4 Changing Mode of Operation

When changing from one mode to another, by pressing the appropriate mode key and interrupting, while under function key control, special conditions apply to moving into record mode. Moves between PAGE and LINE modes and from RECØRD mode are straight forward, the contents of the screen are saved and a statement

****X MØDE SELECTED****

is presented, where X takes the new mode name (see Section 16.3 to 16.3.3).

RECØRD mode can be selected only if the top line of the screen is at the start of a record. If it is not, the message:

***TØPLINE ØF SCREEN NØT START ØF RECØRD,
TØ CØNTINUE SELECT PAGE ØR LINE MØDE***

will appear. Act accordingly.

If a record does start at the first line an integral number of records will be presented on interrupting and part of a record may be erased from the bottom of the screen.

16.5 Document Movement

The remaining function keys control the movement of a document for preparation or editing.

16.5.1 FØRWARD

Depression of the FØRWARD key followed by interrupt will show the first DUI of the selected document being edited or will write the first DUI from the screen into the document and will show the following DUI

If in PAGE Mode, the entire contents of the screen are replaced. In RECORD Mode, the top record is shifted out of the top of the screen the remainder shifting up by the amount vacated. The bottom is filled with as many complete logical records as may completely fit. In LINE mode, the top line is shifted out, the rest shifted up one line and the bottom line filled with the following data from the document.

When entering the head of a document the first DUI of a document being edited will have the first record at the top of the screen left justified. When reaching the end of a document *EØD will appear at the bottom of the screen.

If a new document is being written, the first screenful presented will, in fixed length record mode, contain a blank record at the top of the screen of as many lines as specified by the record length, or one line if variable records have been selected, with *EØD in the remaining lines. The *EØD's may be overwritten by the user. On moving forward the top DUI is moved upward off the screen and the user is presented with the space for the next DUI with *EØD at the left of each line except, when in RECORD mode, for those lines at the bottom into which a record cannot be fitted. If records are of suitable length the *EØD is placed on each line. Also with variable length records each *EØD has a corresponding end of record mark to the right of each line.

An attempt to move a line starting with *EØD off the top of the screen is ignored.

16.5.2 BACKWARD

When interrupted with the BACKWARD key set, one DUI moves into the top of the screen.

In RECORD mode the contents are shifted down so as to accommodate one complete new record at the top. An integral number of records may be removed from the bottom of the screen. This may cause the appearance of one or more blank lines at the bottom. In PAGE mode the whole page will be replaced by the preceding page, records being blank filled to occupy an integral number of lines and with end of record marks as needed. In case of LINE mode the screen contents are shifted down one line and the preceding line placed at the top with blank fill appropriately if the new line is part or all of a new record.

On reaching the load point of a document one or more lines each with *DØC to the left will appear at the top of the screen. In RECORD mode these will be repeated to equal the number of lines occupied by a fixed length record. With variable length records one line appears with an end of record mark to the right. Repetition of BACKWARD moves beyond the load point causes one or more additional *DØC statements to appear, the real records being progressively removed from the bottom.

This will continue on interrupting until the bottom record is one beginning with *DØC. In LINE mode the bottom line is removed, the screen contents are lowered by one line and a *DØC appears at the top; this may be repeated until all lines contain *DØC. In PAGE mode the screen contents are removed and replaced by the lines as far as load point, with *DØC appearing on each line at the top of the screen which would be beyond load point. A further interrupt in PAGE mode will set all lines with *DØC.

Any action which attempts to move a DUI with *DØC beyond the bottom of the page is ignored.

16.5.3 INSERT

On interrupting with INSERT depressed, a DUI is removed from the top of the screen and the space it occupied is left blank for the user to supply a further DUI.

When in RECØRD mode with fixed length records, one whole record is removed. The data to be inserted is written onto the screen and moved out by a FØRWARD movement and the new data is then recorded on the document. When inserting in RECØRD mode with variable length records the top record is moved forward and the remainder of the DUI moved back sufficiently to provide a blank page for the insertion. (No record can be greater than 1000 characters).

If a DUI is to be inserted before load point, LINE mode may be used with the BACKWARD facility until sufficient space at the top of the screen is available. The record may be written over the *DØC's and the DUI inserted into the document by use of the FØRWARD facility. Insertion of a DUI at the end of a document may be achieved by moving FØRWARD until *EØD is reached. As many *EØD lines as needed may be generated by the FØRWARD facility; the DUI is then written on the screen and passed out at the top of the screen for insertion by use of FØRWARD. These functions are best performed using RECØRD and LINE modes only.

16.5.4 DELETE

Deletion of a DUI may be achieved by placing the DUI to be deleted at the top of the screen, depressing the DELETE key and then interrupting. The subsequent actions are as for FØRWARD except that the topmost DUI will be removed without it passing to the document.

16.5.5 RESTORE

Upon interrupting with the RESTØRE key down the screen is returned to the state it was in immediately after the last function key interrupt. This allows for recovery by the user of his current page of data if he causes damage to his page such as would be best corrected by starting from the original page again.

16.6 Error Conditions

Possible errors while under FUNCTION KEY control apart from those mentioned in section 6.2.3.2 are:

16.6.1 Selecting an Illegal FUNCTION KEY

Apart from the situation mentioned in section 16.3.2 it is possible to select an illegal FUNCTION KEY control (e.g. key 12). If this happens CIDER saves the screen display and shows the message

```
***E  ILLEGAL FUNCTION KEY SELECTED.  TØ
      CØNTINUE SELECT RESTØRE (R'ST).
```

The user should depress the RESTØRE key and interrupt and CIDER will return the original screen display. The user can then select the correct key and interrupt.

16.6.2 Attempting to Create Records Longer Than 1000 Characters

If variable length records have been selected then when moving forward in PAGE mode it is possible to attempt to create a record longer than 1000 characters. In this circumstance CIDER will save the screen display and display the message.

```
RECØRD ØVERFLØW WILL ØCCUR IF LINE  N
IS ADVANCED, TØ PRØCEED, PRESS RESTØRE
```

where N is an integer giving the present line number at which overflow of the record will occur.

The user should select the RESTØRE key and interrupt and CIDER will return the original screen display. An end of record marker must be placed before line N before another move is attempted.

17. ON-LINE PROGRAMMING (INTERP)

The displays library program INTERP provides facilities which make it an effective tool for the writing and running of short programs, development of subprograms and subroutines as well as a powerful ad hoc numerical calculator.

~~Two basic facilities are provided. The first handles~~ numerical expressions which may include variables whose values have previously been calculated. This is called CALCULATE mode. In the second, or PROGRAM mode, programs are written in a FORTRAN-like language in which each statement is syntactically checked and executed interpretively. It is also possible to insert, delete, or modify the statements of a program. INTERP programs and associated data composed in PROGRAM mode may be stored on the drums as DAD type documents (see Section 19.3).

Both of these aspects of INTERP are operated from the display consoles. The 25 lines of the screen are divided into three areas, the first and third, comprising lines 1 to 10 and 21 to 25 respectively, being the 'dead' areas and lines 11 to 20 being the 'live' area. The live area is the input area and the dead areas display the program context. When entering INTERP from DAVE the entry marker is placed at the first position of line 11 and all successive TAB-key actions move the marker to the first and seventh positions of successive lines of the live area (this corresponds to common FORTRAN usage). A statement may extend beyond one line but not beyond the boundary of the live area. This differs from FORTRAN in that no indication of statement continuation is required.

To obtain access to INTERP the user must access DAVE and enter INTERP as the display program name and then interrupt. The MESSAGE key will have been selected. INTERP makes no use of the other function keys, but the START and DUMP keys may still be used to access DAVE.

The first message presented by INTERP is:

ENTER STATEMENTS

and is placed on the fifth line of the page.

17.1 The Language

The language used is basically FORTRAN IV except for some restrictions and additions which arise from the nature of the display medium.

17.1.1 Variables

Variables are alphanumeric names of up to 8 characters commencing with a letter. If more than 8 characters are given only the first 8 will be recognised. There must be no embedded blanks.

Not more than 64 different variables may be used in any one program in PROGRAM mode or 25 in CALCULATE mode.

17.1.2 Variable Types

Variables may be either real or integer type. No other types are allowed. Integers may take values between $-(2^{47}-1)$ and $+(2^{47}-1)$. Real values have about 10 decimal digits precision with a decimal exponent between -308 and +308. Real and integer types are implied by the initial letter, as in FORTRAN, unless otherwise declared.

17.1.3 Arrays

Arrays may have up to three subscripts. A subscript may be any valid expression. It is to be noted that the order of input and output of array elements does not follow normal FORTRAN practice. In the case of INTERP arrays are displayed row by row whereas FORTRAN outputs column by column. However, elements are stored column by column in both.

17.1.4 Constants and Literals

- (1) Decimal integer constants may be input and output and have the values between $-(2^{47}-1)$ and $+(2^{47}-1)$.
- (2) Real decimal constants can take up to 10 decimal digits with a decimal integer exponent lying in the range -308 and +308.
Real values are of the format:

$$\pm n.m_{10} \pm s$$

where n, m and s are integers and $_{10}$ is the exponent symbol to base 10. The character ' $_{10}$ ' is used in place of 'E' normally used in FORTRAN. The integers may be omitted in some cases, thus, if m is zero the decimal point may be omitted. Plus signs may be omitted. (No embedded blanks are allowed in any constant.)

- (3) Octal constants may be input only and may take up to 16 octal digits terminated by B.
- (4) BCD character strings may be input only. Strings may consist of up to 64 characters and are left justified. Such strings are enclosed in single quote marks and may not themselves include quotes.

17.1.5 Labels

Labels are integer constants up to five digits in length as in normal FORTRAN practice and are placed to the left of any program statement as needed.

17.1.6 Expressions

A well formed expression consists of a sequence of variables, literals, operators and other delimiters as in normal FORTRAN. Note that variables of both types may be mixed in expressions. Operators in arithmetic expressions differ from FORTRAN usage in only two respects. The operators and delimiters are as follows:

=	equals
+ -	arithmetical addition and subtraction
*	arithmetical multiplication
/	arithmetical division
↑	arithmetical exponentiation (instead of **)
10	arithmetical decimal exponent sign in constants (instead of E)
()	parentheses

17.1.7 Functions

A number of functions of a single variable may be evaluated by entering their mnemonic names into expressions, they are:

<u>Symbolic name</u>	<u>Meaning</u>
SQRT	square root
CUBERT	cube root
ABS	absolute value
COS	cosine
SIN	sine
EXP	exponential function
LOG	natural logarithm
TAN	tangent
COT	cotangent
ACOS	arcosine
ASIN	arcsine
ATAN	arctangent
INT	convert real to integer
FLOAT	convert integer to real

The argument follows the symbolic name as a parenthetical expression:

e.g. SQRT(A), TAN(X + Y ↑ Z)

The value of π , is also available by the name, PI, in CALCULATE mode. Thus $\text{SIN}(\text{PI}*(\text{X}+\text{Z}))$ is the same as $\text{SIN}(3.1415926...*(\text{X}+\text{Z}))$.

Values of functions are real numbers except for ABS of an integer and INT. The argument of a library function is converted to the correct type before evaluation of the function.

17.1.8 Declarative Statements

The following declarations are valid and take their normal FORTRAN meanings:

```
REAL
INTEGER
COMMON
DIMENSION
```

In any of these declarations, variables may be dimensioned with up to three subscripts which must be integer constants and less than 2047. Only blank COMMON is allowed.

17.1.9 Replacement Statements

Replacement statements are of the normal FORTRAN type and mixed mode arithmetic is allowed. However no conversions are performed on octal literals or BCD strings. Truncation will occur when the left hand side is an integer variable and the right hand side is a real expression.

17.1.10 GO TO Statement

This is as in FORTRAN. Execution will not proceed beyond this statement until the label is defined.

17.1.11 Conditional Statements

Conditional statements allow for one, two or three-way branching and contain some notational differences from FORTRAN. Thus the following substitutions apply:

>	replaces	.GT.
<	"	.LT.
^	"	.AND.
v	"	.ØR.
┐	"	.NØT.
bEQb	"	.EQ.
bNEb	"	.NE.
bLEb	"	.LE.
bGEb	"	.GE.

where b represents one or more blanks.

17.1.12 DO Statements

DO statements follow FORTRAN form and behaviour except for the following:

- (1) The DO list parameters may be any expression, positive or negative.
- (2) The indexing parameters are checked for magnitude not greater than 2047, and the increment must not be zero. If the increment size is incompatible the DO block is skipped. Execution will be incomplete until the last statement of the DO loop has been input.

17.1.13 Input/Output Statements

Only two input/output statements are allowed:

- (1) READ $V_1, V_2, \dots V_n$
- (2) PRINT $\text{exp}_1, \text{exp}_2, \dots \text{exp}_n$

where 'V' represents a variable, simple or subscripted, and 'exp' denotes an expression and includes single or subscripted variables, or literals including BCD strings and octal integers. Values called for under READ are keyed in, via the displays keyboard, into line 11 (see Section 17), in integer or real form according to variable type (reals may also be input in integer format if this is the more natural manner of representation).

There are no FORMAT statements and logical units are not referred to as all input/output is through the console.

Results are output in the 'live' area of the screen and their order is that of their declaration in the output list. The names of the variables are attached to the results.

17.1.14 PAUSE Statement

At run time the execution will halt and the statement PAUSE will appear in the 10th line of the display.

18. PROGRAM Mode

PROGRAM mode is entered when the user supplies his first statement:

PROGRAM name

starting at the 7th position on line 11 where 'name' may be any identifier. On interrupting, this statement moves to line 10 and becomes the first statement of the FORTRAN program and declares the program identification. Statements may be entered one by one interrupting after entry of each statement in the live area.

As each interrupt occurs the statements in the upper 'dead' area shift upward sufficiently to allow the next statement to fit at the bottom of that area.

19. OPERATION

There are two distinct phases, a 'checking' phase and an 'execution' phase. They may be used separately or together.

19.1 Checking

The position marker is normally placed in position 1 of line 11 after an interrupt has been serviced. The tab key will advance it to column 7 ready for entry of a FORTRAN statement.

Statements may extend beyond line 11 into succeeding lines. Statements start at position 7 and may be preceded by an integer label. Following entry of each statement the interrupt key is depressed. The statement is then checked for syntactic accuracy and if acceptable all preceding statements will be moved up to accommodate the new statement. If the statement is in error a diagnostic appears in lines 21-22. (See Appendix for examples).

19.2 Execution

Execution may start by placing the statement:

START n or START name

The message: 'name ENTERED' occurs in line 21. Another interrupt will cause execution to start at the point labelled 'n' and if n is undefined a diagnostic is provided. Label 'zero' or 'name' indicates the first executable statement or the leading statement of the program of that name respectively. Execution continues until one of a number of breakpoint conditions are attained. These are:

19.2.1 PAUSE

At this point the top ten lines shown contain the last part of the program executed up to and including the PAUSE reached. The bottom lines 21-25 contain following statements yet to be executed. At this point a GIVE request may be made (see Section 19.2.6). On entering this statement in line 11 and interrupting the requested values are provided in line 11 and the program halts. A further interrupt will cause execution to continue.

19.2.2 End of Executable Statements

Execution will halt if there are no more statements logically capable of being executed. In this case lines 1 to 10 will contain those statements last executed.

If at this stage a new statement is placed in line 11 and the interrupt button depressed that statement will be checked as in Section 19.1.

The next statement if it already exists will appear in the lower dead area. Statements may still be entered in the live area but the upper and lower dead areas will not alter until execution restarts (either by the required statement being entered or by a new START request).

19.2.3 Error Pause

If a faulty condition occurs whilst executing a statement, a diagnostic will appear on line 11 (see Appendix I) together with a reconstruction of the statement up to the point at which the error condition occurred, and execution will halt with the faulty statement in line 21. A typical diagnostic would be:

Line 11, "INTEGER OVERFLOW
 J = I + "

Line 21, J = I + ALPHA ↑ 3

meaning that in doing the + operation overflow occurred.

19.2.4 READ Pause

When a READ statement is reached execution will pause with the READ statement in line 10. If the input list is

$id_1, id_2 \dots id_n$

then the screen will have

SET id_1 =

present in line 11. The user then enters the value (or values if id_1 represents an array) and interrupts. Values for arrays of more than one dimension are input in row major order. The user is then presented with

SET id_2 =

in line 11. He enters the value and interrupts, and so on to the end of the input list. A final interrupt will cause execution to resume.

19.2.5 PRINT Pause

When a PRINT statement is entered execution will halt with that statement displayed in line 10 and the print list values will be displayed in lines 11, 12 etc. Each of these will be preceded by $VALUEn$ for the value of the n^{th} expression,

or, if the expression is simply an identifier, its value will be preceded by its name.

e.g. `PRINT 3, A, X + 3*Y`

will give `VALUE1 = 3`, `A = value of A`, `VALUE2 = value of (X+3Y)`. If a subscripted variable occurs in the print list the values of all its elements are given in the form:

"A(1) = value of A(1), value of A(2), value of A(3), etc."

and in row major order for two or three dimensional arrays.

If the array `A` cannot be completely displayed in lines 11 to 20, the system may be interrupted when the next or last part of the array will be displayed on lines 11 to 20 as:

"A(r) = value of A(r), value of A(r+1) etc."

and so on, where `A(r-1)` is the last element previously displayed in line 20. A final interrupt will cause execution to resume.

19.2.6 GIVE Request

At a pause a GIVE request may be made. This request takes the form:

`GIVE exp1, exp2, exp n`

in lines 11, 12 etc. Interruption will cause the values of the expressions to be computed and displayed in lines 11, 12 etc.

e.g. `GIVE A(1), X+5*Y`

is replaced by "A(1) = value of A(1), VALUE2 = value of (X+5*Y)". A further interrupt will cause execution to resume.

19.2.7 SET Request

At a pause a SET request may be made. This request takes the form:

`SET v1 = c1 v2 = c2 v3 = c3 etc.`

where the `v1, v2 etc.` are variables and the `c1, c2, etc.` are constants or constant lists. On interrupting the variables will be set to the defined values and the message will disappear and execution continues.

19.3 EDIT

It is possible while in PROGRAM mode to enter the request:

`EDIT i`

where i is the name of a subprogram already entered (or currently being entered) or is the DAD type identification of a drum document. If i is not the program being currently handled DAD's document list is searched for a document of the same title and under the user's charge code c (the record is thus c,i). If the required document is available ~~it is accepted for EDITing.~~ If the named document does not exist, a diagnostic

DOCUMENT name CANNOT BE FOUND

is displayed in line 21.

19.3.1 Scanning

An EDIT status remains in power until either a new EDIT statement is typed or an end-of-file (including EOD) is reached.

In EDIT status, with the MESSAGE key set, a program is inspected statement by statement starting with the leading statement which is placed in the live area. The following statement is displayed in the lower dead area, lines 21-25. After editing by the Keyboard and upon interrupting, the statement in the live area is lifted into the bottom of the upper dead area, any preceding statements being lifted to accommodate it. The statement from the lower dead area is now displayed in the live area and the next statement appears in the lower dead area.

While scanning a document it is possible to execute the statements as they move from the live area to the upper dead area by a prior START request. In this case the program will be executed statement by statement on interrupting.

19.3.2 INSERT

On setting the INSERT key (as defined by CIDER) and interrupting the statement starting in line 11 will be transferred to the upper live area and the following statements will be moved out of the live area into the lower dead area. The live area will be cleared so that a new statement may be written starting in line 11. This will be inserted on pressing the MESSAGE key and interrupting. The process must be repeated for each statement that is to be inserted.

19.3.3 DELETE

Deletion of a statement is achieved by depressing the DELETE key (as defined by CIDER) and then pressing the interrupt button, when the statement starting in line 11 will be cleared and the next statement, in line 21, is raised to line 11 and the following one is raised to occupy the space created in the lower dead area.

Deletion of a string of statements may be achieved by simply depressing the interrupt key the appropriate number of times with the DELETE key set.

19.3.4 SAVE

SAVE i

This statement will at any time cause the program currently in use to be retained on the drum and listed in DAD's document list (MDL) under the name, c,i, from where it may be accessed by an EDIT.

20. CALCULATE Mode

This mode of operation is entered if the user replies to INTERP's first message with

CALCULATE

in the eleventh line starting at the 7th position from the left. The marker will have been set to this position by INTERP, and from this point on all entries from the keyboard into the live area will be in free format in which spaces are accepted as delimiters. The message returned at this stage is:

CALCULATE MODE ENTERED
VALID STATEMENTS MUST TAKE THE FORM
VARIABLE = EXPRESSION

Statements in CALCULATE mode are now entered into the live area one by one, interrupting and computing after typing each statement.

There are three types of CALCULATE statement, one which computes the value of an variable directly from an expression involving numerical literals and pre-evaluated variables. The other two are the GIVE and SET statements as defined in 19.2.6 and 19.1.7

20.1 Result Presentation

Following the keying of the lefthand and righthand parts of a replacement statement the interrupt button must be pressed. The righthand part of the keyed statement will be calculated and the result will be placed in the first line of the display. The original replacement statement placed in the live area will be removed. All values and their accompanying variable names previously placed in the upper dead area, lines 1 to 10, are shifted down the area to accommodate new values. If there is insufficient space to the right of a line the variable and its value are placed to the left of the next line.

As the transfer of variables and values to the dead area proceeds the top 10 lines many become filled; entries may then disappear to the right of the 10th line but will be retained by INTERP so long as there are less than 25 such results. If more than 25 variables are used the diagnostic

4 TOO MANY IDENTIFIERS

will be presented on line 21 of the display.

Variables may be given new values by equating them to a new righthand part. In such a case the same variable name appearing on lines 1 to 10 will be associated with the new value.

Example:

At line 11, key in: A = 1.72532
Press INTERRUPT and
Line 1 becomes: A = 1.72532

At line 11, key in: B = 2 + 1.8*A
Press INTERRUPT and
Line 1 becomes: B = 5.10559, A = 1.72532

At line 11, key in: D = 5 + (B/A)²
Press INTERRUPT and
Line 1 becomes: D = 13.7569, B = 5.10559, A = 1.72532

At line 11, key in: INTEGER = D
Press INTERRUPT and
Line 1 becomes: INTEGER = 13, D = 13.7569, B = 5.1055
and Line 2 becomes: A = 1.72532

Note that not more than six significant figures are given in real numbers although the values are calculated to higher precision. Integers are, however, given to 15 decimal digits.

20.2 Diagnostics

Diagnostic messages may be presented. See Appendix I.

21. Termination of INTERP

Exit from an INTERP run is attained if the statement END is placed in line 11 and executed or the END termination of a INTERP program is reached. Such termination causes the message:

ENTER STATEMENTS

to be displayed in line 5. Return to DAVE is obtained by pressing the START key and interrupting.

APPENDIX IDiagnostics Produced by INTERPA.1 CHECKING PHASE

The checking phase of INTERP may produce two types of diagnostic.

A.1.1 TYPE 1 DIAGNOSTICS

This comprises a large class of possible faults of which the following examples are common:

1. parentheses not closed or nested correctly
2. two or more operators concatenated
3. DO written instead of DØ

This type of error causes the message:

```
line 21  CATEGORY IMPOSSIBLE IN CONTEXT
line 22  nnnnnB, nnnnnB
```

where the n's are octal digits. The pairs of octal numbers constitute a dictionary reference in a list of diagnosed faults. The two right hand digits of the pairs of numbers corresponds to a major category of faults and the remaining digits specify a sub-category.

Checking for syntactic errors is comprehensive and a full list of detected syntactic faults may be found in Pt. 2 of the DAD System Programmer's Manual.

A.1.2 TYPE 2 DIAGNOSTICS

Other faults of program structure cause the following diagnostics to appear:

```
line 21  **STATEMENT REJECTED**
line 22  *nn message
etc.
```

where nn are octal digits. The messages are shown in the following table where the octal integer to the left corresponds to nn.

- 0 CATEGORY IMPOSSIBLE IN CONTEXT
Implies that not all the input can be accounted for
the statement
- 1 NO SUBPROGRAM NAME GIVEN
- 2 SUBPROGRAM TOO LONG
Implies that there is overflow of a program beyond the
allowable limit
- 3 INVALID CONSTANT OR COMMENT
- 4 TOO MANY IDENTIFIERS
The user is restricted to 64 different variable names.
An array name counts as one variable name.
- 5 STRING TOO LONG
Strings may not be longer than 64 characters.
- 6 INNER DO LOOP NOT TERMINATED
e.g. DO 3 I =
 - - -
 DO 2 J =
 3 CONTINUE
- 7 LABEL REFERENCED OUTSIDE DO LP
e.g. GO TO 3
 DO 3 I =
- 10 LABEL ALREADY SET
Implies a doubly defined label
- 11 GO NOT FOLLOWED BY TO
- 12 NO PATH TO THIS STATEMENT
- 13 NO GIVE OR SET BEFORE START
Interpretation must be started before a GIVE or
SET request can be obeyed
- 14 ILLEGAL STATEMENT FOR CALCULATE
- 15 FORMAL PARAMETER NOT DECLARED
e.g. DIMENSION X(A)
- 16 MORE THAN THREE DIMENSIONS
Too many subscripts in a DIMENSION statement
- 17 ARRAYS ARE TOO LARGE
No single dimension can exceed 2047
- 20 ARRAY DECLARED TWICE
- 21 INCONSISTENCY IN DECLARATIVE
e.g. REAL X
 INTEGER X
- 22 FORMAL PARAMETER IN CMN. ARRAY
- 23 ILLEGAL USE OF SUBPROGRAM NAME
e.g. PROGRAM ALPHA
 - - - - -
 ALPHA = 1

- 24 **TØØ MANY SUBSCRIPTS**
In executable statement. Too many subscripts for array declaration
- 25 **TØØ MUCH CØMMØN**
Upper limit of CØMMØN area is 8,192 words
- 26 **TØØ MUCH LØCAL ARRAY SPACE**
7936 words may be allocated to local arrays and the condensed version of executable statements. Reduce the array sizes or the program.
- 27 **INVALID START REQUEST**
A starting label must be valid, referenced and not in a DØ loop.
- 30 **FUNCTION HAS NØ PARAMETERS**
- 31 **INVALID FØRMAL PARAMETER LIST**
e.g. FUNCTION A(B+C,...)
- 32 **ALREADY WITHIN SUBPRØGRAM**
Checks on repetition of subprogram names.
- 33 **TØØ MANY SUBPRØGRAM NAMES**
Not more than 8 subprograms can be used.
- 34 **PRØGRAM CALL WITH PARAMETERS**
e.g. PRØGRAM X(A,B,C)
- 35 **FØRMAL PARAMETER CALLED TWICE**
e.g. FUNCTION X(A,A)
- 36 **INVALID STATEMENT NUMBER**
Labels must lie between 1 and 99999
- 37 **ATTEMPT TØ ENTER DØ LØØP**
e.g. DØ 3 I = ...
 - - - -
 3 CØNTINUE
 GØ TØ 3
- 40 **INDEX VARIABLE ALREADY IN USE**
Nested DØ loops cannot use same index variable
- 41 **DØ LØØPS NESTED TØØ DEEP**
DØ loops may not be nested more than 7 deep.

A.2 INTERPRETATION PHASE

Further checking is carried out during interpretive execution and the diagnostics are as follows:

- (1) **INTEGER ØVERFLØW**
An integer may be in the range $-(2^{47}-1)$ and $+(2^{47}-1)$.
- (2) **REAL TØØ LARGE TØ CØNVERT TØ INTEGER**
- (3) **REAL ØVERFLØW**
Index too large; must be between +1023 and -1023.

- (4) NØN-STANDARD REAL
A real fractional part is not normalised to the range
.5 to $1-2^{-35}$
- (5) ILLEGAL ØPERANDS
Use of octal digits or BCD character string as both
operands in a binary operation or one of them in a
wrong operation.
- (6) LØCAL SUBSCRIPT NØT IN RANGE
- (7) REF. TØ NØN-CØMPUTED ELEMENT
- (8) INDEXING PARAMETERS GT 2047
- (9) DØ INCREMENT IS ZERØ
- (10) SET LIST TØØ LØNG
Number of values SET into an array is greater than the
dimensions of the array.
- (11) BAD ARGS FØR LIBRARY FUNCTIØN
- (12) INCØRRECT NUMBER ØF ARGUMENTS

