8. DOS6800 PROCESSORS

8.1 Introduction

The DOS6800 processors are as follows:

- Assembler — language processor
- CREDIT Translator — language processor
- CREDIT Linker — pre linkage edit processor
- Line Editor — interactive text editor
- Linkage Editor — load module builder

The processors are held in the system library and are called into execution by the control commands ASM, TRA, TLK, LED and LKE as described in section 6.12.

The Linkage Editor and Line Editor are described in the following sections. The Assembler is described in the Assembler PRM (M06). The CREDIT Translator and CREDIT Linker are described in the CREDIT PRM (M04).

8.2 The Line Editor

8.2.1 *Introduction*

The Line Editor is an interactive text editor used for updating source program modules or data files. The following types of update may be performed:

- Search text for character string and list all occurrences (command mnemonic !!LS).
- Search text for character string and replace by new character string (command mnemonic !!CH).
- Insert text from another file into the current file (command mnemonic !!JN).
- Replace character string in specified line (command mnemonic !!RE).
- Delete specified line or lines (command mnemonic !!DL).
- Insert text after specified line (command mnemonic !!IL).

Having entered the necessary updates the user may either terminate the Line Editor normally or may abort. If the Line Editor is aborted the output file is scratched.

The Line Editor is called into execution by the control command LED. Updates are normally keyed-in on the console typewriter. The updates are applied to the library file specified in the LED command and an updated output file is created. This is written to the /S file or to the temporary file specified in the LED command. The maximum line length of these file is 80 characters.

The LED command is described in detail in section 6.12. After this command has been keyed-in the Line Editor will, if the console typewriter is being used for updating, respond with the prompt L:. The user may then key-in the required update commands.

8.2.2 *Editing phases*

The Line Editor performs all updating during a single pass of the input file. For this reason the user must key-in all commands which apply to the whole file before the update pass begins. These commands are !!LS (search for character string and list all occurrences) and !!CH (search for character string and replace). !!LS commands are executed as soon as they are keyed-in because they do not update the file. !!CH commands are stored in memory until the update pass begins.

When the Line Editor begins execution it is said to be in the "definition phase". During this phase several !!LS and/or !!CH commands may be keyed-in. When a command other than !!LS or !!CH is keyed-in the Line Editor immediately starts the update pass. This is known as the "execution phase". During the execution phase the user may key-in text insertion commands (!!JN and !!IL), text replacement commands (!!RE) and text deletion commands (!!DL). Because the Line Editor carries out a single sequential update pass the line numbers specified in these commands must be in ascending order.

No !!LS or !!CH commands may be keyed-in during the execution phase. However, any !!CH commands keyed-in during the definition phase are obeyed during the execution phase as each line of the input file is scanned.

During either the definition or execution phase the Line Editor may be terminated (!!EN) or aborted (!!AB).

Note that a KPF control command must be issued after termination of the Line Editor if the temporary output file is to be retained.

8.2.3 *Update command reference*

This section describes the syntax and use of each update command. The syntax for each parameter in these commands is given in appendix A. The notation conventions are described in section 1.9.

!!CH

CHANGE STRING

!!CH

Syntax : !! CH␣ $$ character-string-1$$ character-string-2$$

Use : Search the file for character-string-1 and replace by character-string-2. Every occurrence of character-string-1 in the file will be replaced. This command may only be used in the definition phase.

!!DL

DELETE LINES

!!DL

Syntax : !!DL␣ line-number-1[,line-number-2]

Use : All lines from line-number-1 to line-number-2 inclusive are deleted. If line-number-2 is omitted only line-number-1 is deleted. The user may key-in several lines of text following the !!DL command. This text will be inserted in place of the deleted text until a line beginning with !! is encountered. All deleted lines are listed on the device with file code /2 (unchanged by any !!CH command). This command may only be used in the execution phase.

!!EN

END EDITOR

!!EN

Syntax : !!EN

Use : This command terminates updating. The remainder of the input file is copied to the output file whilst any !!CH updates are applied. The Line Editor will then hand control back to the CCI. This command may be used in either the definition or execution phase.

!!IL

INSERT LINES

!!IL

Syntax : !!IL␣ [line-number]

Use : This command enables the user to insert one or more lines of text after the specified line-number. If line-number is omitted the text is insert after the current line. Lines of text are inserted until a line beginning with !! is encountered. Inserted lines are listed on the device with file code /2. They are not changed by any !!CH command. This command may only be used in the execution phase.

!!JN

JOIN AUXILIARY FILE

!!JN

Syntax : !!JN ␣[line-number-1] ,file-name,line-number-2[,line-number-3]

Use : Text from the auxiliary file identified by file-name is inserted into the file being updated after line-number-1. The file name may refer to the file currently being edited. If line-number-1 is omitted the text is inserted after the current line. The text to be inserted is specified by line-number-2 through line-number-3 inclusive. If line-number-3 is omitted only the line specified by line-number-2 is inserted. This command may only be used in the execution phase.

!!LS

LIST LINES

!!LS

Syntax : !!LS␣$$character-string$$

Use : This command searches for all occurrences of character-string in the input file. Each line containing character-string is listed on the device with file code /2. This command may only be used during the definition phase.

!!RE

REPLACE STRING

!!RE

Syntax : !!RE␣line-number,$$character-string-1$$character-string-2$$

Use : This command replaces character-string-1 by character-string-2 in the line specified by line-number. If this replacement changes the length of the line it will be truncated or space filled on the right. Characters will not be shifted from one line to another during this process. This command may only be used during the execution phase.

8.2.4 *Error reports*

The following error reports may be printed during the execution of the Line Editor:

Message	Meaning
FILE NAME ERROR	The specified name in the update command contained an error.
FILE NAME MISSING	The specified name cannot be found on this disk.
INPUT FILE CANNOT BE ASSIGNED	This error report is followed by a report explaining the error.
/S CANNOT BE ASSIGNED	This error report is followed by a report explaining the error.
INVALID FILE CODE	The file code specified does not belong to the input device from which the update commands are input.
FILE CODE NOT ASSIGNED	The file code of the command input device must have been specified beforehand by means of ASG.
TOO MANY PARAMETERS	Too many parameters specified.
DSK INPUT ERR. UPD ABORTED	Line Editor cannot read from disk.
DSK OUTPUT ERR. UPD ABORTED	Line Editor cannot write onto disk.
UNKNOWN COMMAND. TRY AGAIN	The update command is not accepted.
I/O ERR ON LAST RECORD, TRY AGAIN	An I/O error occurred.
SEQUENCE ERR. TRY AGAIN	The line numbers in the update command are not in ascending order.
SYNTAX ERR. TRY AGAIN	The update command or the newly typed in line, contained a syntax error.
AUX INPUT CANNOT BE ASSIGNED. TRY AGAIN	The auxiliary file used in !!JN command cannot be assigned.
CMND NOT ALLOWED IN EXE MODE. TRY AGAIN	This command cannot be used in the execution phase.
TABLE O'FLOW, TRY AGAIN	The character string table is overflowing.
EOF, UPD TERMINATED	The :EOF mark has been encountered on the input source file before reaching the specified line, thus terminating the update process.

Message	Meaning
EOF IN AUXI INPUT	The :EOF mark has been encountered from the Auxiliary Input. The !!JN command is terminated but the operator continues.

When the message TRY AGAIN is printed the user has the possibility of correcting the previous command or data record from the device with file code /1. If (CR) is typed in, the input is resumed from the normal input command file.

8.3 The Linkage Editor

8.3.1 *Introduction*

The function of the Linkage Editor is to build an executable load module from a specified set of object modules.

The Linkage Editor is called into execution by the control command LKE. The input object modules are taken from the /Ø file the users /ØBJCT file and/or the system /ØBJCT file. The output load module is written to the /L file.

The LKE command is described in detail in section 6.12.

8.3.2 *Processing*

The Linkage Editor builds the load module by incorporating the relevant object modules one by one. The first object module to be incorporated is the one which contains the application program start point. This module must be held in the /Ø file and must contain as an entry point the label which is to be used as the program start point. If this label is not specified in the LKE control command the last start point found in the /Ø file will be used. This module will normally contain references to other modules. These modules may contain references to further modules, and so on. The Linkage Editor builds the load module as a hierarchy of object modules connected by entry point references.

The referenced object modules must be held in the /Ø file, the user /ØBJCT file or the system /ØBJCT file. The Linkage Editor first incorporates any referenced modules which are held in the /Ø file. If any unsatisfied references exist after this the Linkage Editor will scan the /ØBJCT file of the current userid. If there are still any unsatisfied references the Linkage Editor will scan the system /ØBJCT file and then the user /ØBJCT file again. Scanning of the user and/or system /ØBJCT files can be suppressed by including the U, S or N parameter in the LKE control command.

A number of error conditions can arise during linkage editing. These conditions and the corresponding error reports are listed below.

8.3.3 *Output listings*

An example of the listing produced by the Linkage Editor is shown below. A new page is thrown at the start of each section.
Heading information:

```
LKE U,M
DATE   /  /            TIME  00H-11M-23S-
LABEL = SAVORED      DATE = 31 08 76      PACK NBR = 000    GEERH
```

Memory map:

```
OUTP03    0008
SWRL00    0020
WRITE     0154
FUNCTN    0490
WRTLST    0718
```

Symbol table:

```
                    *** SYMBOL TABLE ***

CTLTAB 0718 R    PICTAB 0718 R    STB    0052 R    SWRL00 00F2 R
T:ADD  04AC R    T:ADDC 04A8 R    T:CMP  06A6 R    T:CMPC 06A2 R
T:CPA  069E R    T:CPAC 069A R    T:EDT  01B4 R    T:EDTZ 01B0 R
T:MOU  0602 R    T:MOVC 06FE R    T:OP1  049C R    T:OPA  0594 R
T:OPS  05D0 R    T:SUB  04A4 R    T:SUBC 04A0 R    T:WRT  0106 R
T:WRTZ 0102 R    TAWRL  0722 R    WB1    0054 R

                        ******

                START = 000C     LENGTH = 079C     REGION = 0477

:EOF
PROG ELAPSED TIME: 00H-00M-08S-700MS-
```

The heading information contains the Monitor date and time and the label, date and number of the system disk.

The listing of the memory map and symbol table is optional. It is produced if the M parameter is included in the LKE control command.

The memory map contains the name (including any comments on the IDENT line) of each object module built into the load module together with the address at which it occurs within the load module. The modules are listed in the order in which they occur within the load module.

The symbol table comprises an alphabetic list of entry point labels together with the addresses at which these labels occur within the load module. Labels which are referenced but which do not exist in the load module are included in the symbol table with **** for the address. A letter is also printed after each address indicating the type of address. These letters are:

A = Absolute
R = Relocatable
S = Internal symbol table
U = Undefined

8.3.4 *Error reports*

An error report is output on the printer whenever an error condition is detected by the Linkage Editor. Errors may be fatal or non fatal. Fatal errors are reported on the console typewriter (T), as well as the printer (P). The possible error reports are as follows:

Report	Output Unit	Abort?	Meaning
IO ERROR file ssss	T	Yes	Irrecoverable I/O error on file 'file' with status 'ssss' of the I/O unit.
BLK.COM.	P	No	Wrong optional blank common address.
BLK.DAT.name	P	No	'name' is an unknown common block name used in a Block Data Subprogram.
DBL.DEF.name	P	No	'name' is defined more than once as an entry point or as the name of a common block.
INV.LGH.name	P	No	'name' is a common block name whose length exceeds the maximum length allowed.
UNS.EXT.	P	No	There are one or more unsatisfied external references. The load module may be executable when no references are made to its externals. The externals are listed in the map.
ABS.STR.	P	No	Absolute start address (ignored).
ERR.MOD.	P	No	A link-edited module has received an error flag from assembly or compilation.
NO STRT.	P	No	Wrong (or no) start address.
INV.IDT.	T&P	Yes	Invalid IDENT record.
PRG.OVL.	T&P	Yes	Generated load module exceeds 32K words.
TBL.OVL.	T&P	Yes	Not enough space to link-edit these modules.
IDT.MIS.	T&P	Yes	IDENT record missing.
END MIS.	T&P	Yes	END cluster missing.
ERR.LKE.	T&P	No	A non-fatal error has occurred during this link-edit run.
ABS.ADR.	T&P	Yes	An absolute address was read. The Linkage Editor does not accept absolute addresses.